Facial Expression Analysis by Computational Intelligence Techniques

Maja Pantic

About the front cover

ISFER logo: originally designed by Anna Wojdel of Delft Univesity of Technology and redesigned by Pyrrhos Stathis of Delft University of Technology

About the back cover

Pumpkins: data courtesy Joost Elffers, © 1997 by Joost Elffers

Maja Pantic

Facial Expression Analysis by Computational Intelligence Techniques / Maja Pantic
Delft: TU Delft, Faculteit der Informatietechnology en Systemen.
Thesis Technische Universiteit Delft. – With ref. – With summary in Dutch

Subject heading: facial expression analysis / computational intelligence techniques

Facial Expression Analysis by Computational Intelligence Techniques

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Techische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.F. Wakker,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 15 oktober 2001 om 10.30 uur

door

Maja PANTIC

informatica ingenieur
geboren te Belgrado, Joegoslavië

Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. H. Koppelaar

Toegevoegd promotor:
Drs. dr. L.J.M. Rothkrantz

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. H. Koppelaar,	Technische Universiteit Delft, promotor
Drs. dr. L.J.M. Rothkrantz,	Technische Universiteit Delft, toegevoegd promotor
Prof. dr. ir. J. Biemond,	Technische Universiteit Delft
Prof. dr. H. J. van den Herik,	Rijks Universiteit Limburg
Prof. dr. N. Frijda,	Vrije Universiteit Amsterdam
Prof. dr. K. Bowyer,	The University of South Florida
Dr. A. Amir,	IBM Almaden Research Centre

PROPOSITIONS

belonging to the Ph.D. thesis

Facial Expression Analysis by Computational Intelligence Techniques

by

Maja Pantic

1. Anyone can become angry - that is easy. But to be angry with the right person, to the right degree, at the right time, for the right purpose, and in the right way – that is not easy. This challenge recognised by Aristotle in the third century BC, remained for humans difficult enough to make our stress assessing system (quite) profitable.

   [this thesis]

2. The things we have to deal with in practical life are usually too complicated to be represented by neat, compact expressions. Especially when it comes to understanding emotions, so little is known that we cannot be sure our ideas are even aimed in the right directions. Therefore, one must not think that we developed a universal emotion recogniser. We just simplified the problem by mapping it onto an (infinite) number of (solvable) subjective opinions.

   [this thesis]

3. AI has many definitions. As explained by Randall Davis, an unifying definition is difficult to give:
   "Is the AI science or engineering, analytic or synthetic, empirical or theoretical? The answer is of course 'yes'."

   (in *AI Magazine* 19(1): 94)
   [this thesis]

4. Ontologies specify the concepts and relations within a domain of discourse. Scientists, philosophers, sociologists and linguistics have been striving for good ontologies since Empedocles described the four elements – air, earth, fire and water – in the fifth century BC. Only in the AI field we now use ontologies for classification, system modelling, human computer interface, computer reasoning, data mining, and so on, in the fields such as medicine, physics, molecular biology, electronics, etc.
   Did we forget the meaning and power of the term "abstraction" (as a counterpart of the term "detailed")?

5. In our post-agrarian society, where few of us farm or sew, the phrase "looking for a needle in a haystack" has lost it's meaning. A more fitting expression for today's culture might be "finding the right information on the Internet".

6. A quick test of intelligence: Read the following sentence and count only once the number of the letter F in it.

   ```
   FINISHED FILES ARE THE RESULT OF YEARS OF SCIENTIFIC STUDY
   COMBINED WITH THE EXPERIENCE OF YEARS.
   ```

   A person of average intelligence finds three of them. Nevertheless, there are in total six F's. There is no catch. Many people forget the OF's and that's because the human brain tends to see them as V's and not F's.
   Nevertheless, we still do consider the human visual system as the (most) reliable inspection facility of our facial-expression-analyser.

   [this thesis]

7. People are seldom interested in *how* results are achieved. Most of the times they are just interested in the results themselves. This is because we assume that everybody thinks / works in a similar way, that is, the same way as we do.

8. One must not mistake to flatter himself thinking that what he achieved is perfect. It is a praise of ignorance.

9. It is positive to remember the past and think about the future, but to live even slightly in the past or the future, is dangerous. It is a robbery of the present life: neither rescuing anything from the past nor doing something for the future.
   Carpe diem !

10. Only active people with ambition and Machiavellian way of thinking move life forward, but only passive persons due to their patience and goodness make life bearable.

11. Once upon a time in a faraway forest there was a rabbit writing something on a computer. A wolf saw him and asked what is he doing. "I am writing my Ph.D. thesis", replied the rabbit. "And what is the subject?", asked the wolf. "Rabbit, the strongest animal of the forest", said the rabbit. "It can't be", argued the wolf, "come in the bushes and I will show you". After a couple of minutes the wolf came out of the bushes, beaten almost to death. The rabbit followed him with a bear explaining "It is not the subject what matters, but the mentor".

STELLINGEN

behorend tot het proefschrift

Facial Expression Analysis by Computational Intelligence Techniques

door

Maja Pantic

1. Eenieder kan boos worden – dat is gemakkelijk. Maar, boos worden op de juiste persoon, in de juiste mate, op het juiste moment, met het juiste doel en op de juiste wijze – dat is niet gemakkelijk. Deze uitdaging die Aristoteles in de derde eeuw vC formuleerde, bleef voor mensen moeilijk genoeg om ons stress-observerend systeem (redelijk) doelmatig te houden.

   [dit proefschrift]

2. De zaken waarmee we rekening moeten houden in het dagelijkse leven zijn gewoonlijk te gecompliceerd om weer te geven met behulp van nette beknopte uitdrukkingen. Vooral als het erop aan komt om emoties te begrijpen is daar zo weinig over bekend dat we er niet zeker van kunnen zijn dat onze vermoedens daarover zelfs maar in de juiste richting gaan. Daarom moeten we niet denken dat we een universele emotie-herkenner ontwikkelden. We hebben gewoon het probleem vereenvoudigd door het op een (oneindig) aantal (onderscheidbare) subjectieve meningen af te beelden.

   [dit proefschrift]

3. AI heeft vele definities. Zoals uitgelegd door Randall Davis, is een eensluidende definitie moeilijk te geven:
   "Is the AI science or engineering, analytic or synthetic, empirical or theoretical? The answer is of course 'yes'."

   (in *AI Magazine* 19(1): 94)
   [dit proefschrift]

4. Ontologieën specificeren concepten en relaties in een vakgebied. Wetenschappers, filosofen, sociologen en linguïsten hebben gestreefd naar goede ontologieën sinds Empedocles, in de vijfde eeuw vC, de vier elementen – aarde, lucht, water en vuur – beschreef. Tegenwoordig worden slechts in het vakgebied AI ontologieën opgesteld voor classificatie, systeemmodellering, mens-machine interactie, automatisch redeneren, data mining, en toegepast in zulke disciplines als medicijnen, natuurkunde, moleculaire biologie, elektronica,

enz. Vergaten wij de betekenis en kracht van de term “abstractie” (als tegenhanger van de term “gedetailleerd”)?

5. In onze postagrarische samenleving, waarin weinigen van ons nog zaaien of oogsten, heeft de zinsnede “zoeken naar een naald in een hooiberg” zijn betekenis verloren. Een meer passende uitdrukking in de hedendaagse cultuur zou kunnen zijn “zoeken naar de juiste informatie op het Internet”.

6. Een snelle intelligentietest: Lees de volgende zin slechts eenmaal en tel daarbij het aantal malen dat de letter F erin zit.

   ```
   FINISHED FILES ARE THE RESULT OF YEARS OF SCIENTIFIC STUDY
   COMBINED WITH THE EXPERIENCE OF YEARS.
   ```

   Een persoon van gemiddelde intelligentie vindt er drie. Toch zijn er in totaal zes letters F. Er schuilt niets achter. Mensen vergeten het woordje OF omdat het menselijk brein ertoe neigt om dat te lezen met een V en niet met een F. Desalniettemin beschouwen we het menselijke visuele systeem als het (meest) betrouwbare waarnemingsinstrument van onze gezichtsuitdrukkinganalysator.
   [dit proefschrift]

7. Mensen zijn zelden geïnteresseerd in *hoe* resultaten bereikt worden. Meestal zijn zij slechts geïnteresseerd in resultaten. Dit komt omdat zij aannemen dat ieder denkt en werkt op dezelfde wijze, dat is, de wijze waarop zij het zelf doen.

8. Men moet zich niet abusievelijk vleien door te denken dat hetgeen bereikt is volmaakt is. Dit is de lof der onwetendheid.

9. Het is positief om zich het verleden te herinneren en te denken over de toekomst. Om zelfs maar enigszins in het verleden, of in de toekomst, te leven is gevaarlijk. Het is een zonde tegen leven in het heden: noch verandert het iets uit het verleden, noch voorkomt het iets in de toekomst. Carpe diem!

10. Slechts actieve mensen met ambitie en een Machiavelliaanse wijze van denken brengen leven in de brouwerij. Slechts passieve mensen maken het leven draaglijk, dank zij hun geduld en goedaardigheid.

11. Er was eens in een heel donker bos een konijn dat typte tekst met behulp van een computer. Een wolf zag hem bezig en vroeg wat hij deed. “Ik schrijf mijn proefschrift”, antwoordde het konijn. “En wat is je onderwerp?”, vroeg de wolf. “Konijnen, de sterkste dieren van het bos”, zei het konijn. “Dat kan niet”, argumenteerde de wolf, “kom maar mee in de struiken en dan zal ik je dat laten zien”. Na enkele minuten kwam de wolf uit de struiken … halfdood. Het konijn liep achter hem aan in gezelschap van een beer en legde uit “Het onderwerp doet er niet toe, wat er wel toe doet is welke mentor je hebt”.

Preface

The pre history of this thesis dates back to the period of the last six months of 1996 when I was working on my M.Sc. thesis "*Human Emotion Recognition Clips Utilised Expert System (HERCULES)*". I keep many pleasant memories of those times and remember the great ambience; I wish to thank my family and friends for making me laugh and my supervisor Leon Rothkrantz for his uttermost confidence in me. Those were the times when I started my Ph.D. research.

Not even two years after that, as everybody prophesied to me, it was time for the great "second-year Ph.D. depression". At that time, automatic reasoning about facial expressions seemed to me much easier than the reasoning about my own life. Fortunately, I met great people shortly afterwards, which turned my life into a romantic comedy instead of the previous psychological tragedy. Special thanks go to Ioannis Patras.

Much of this work would not have been possible without the dedicated support of my promotors Henk Koppelaar and Leon Rothkrantz. What I learned from Leon goes far beyond the covers of this thesis. I would like to thank him for enduring my personality, his willingness to listen to all of my stories and, most of all, for his valuable opposing to my scarce proposals and his abiding encouragement.

Many keen thoughts and "lights in the darkness" came out of discussions with certain members of the Information and Communication Theory group of the Faculty of Electrical Engineering, namely Ioannis Patras and Cor Veenman. I would like to thank them as well as Rafael Bidarra and Miodrag Djurica for their non-exhaustive pool of advice, suggestions and reassuring words about my papers and the thesis. Also, I would like to thank Mrs. Nieman for her patience and great help in editing my thesis.

My special thanks go to my mom, who was always ready to spent hours making my English readable, and to Ivana Pantić and Zoran Trajković, without whom I would not have been able to come to the Netherlands. Thank you for your invaluable support and encouragement.

Contents

To my mom

1 Introduction

Grand challenges like space exploration and weather prediction are expanding human frontiers, but the grandest challenge is exploration of how we as human beings react to the world and interact with each other. Faces are accessible "windows" into the mechanisms which govern our emotional and social lives. The technological means are now in hand to develop automated systems for monitoring facial expressions and animating artificial models. Face technology of the sort we describe, which is now feasible and achievable within a relatively short time frame, could revolutionise fields as diverse as medicine, law, communications and education.

(Ekman and Sejnowski 1993)

The human face is involved in an impressive variety of different activities. It houses the apparatuses for speech production (mouth, tongue and teeth) as well as the majority of our sensory apparatuses: eyes, ears, mouth and nose, allowing the bearer to see, hear, taste and smell. Besides these biological functions, the human face provides a number of social signals essential for interpersonal communication in our public life. The face mediates person identification, attitudinal/emotional state and lip-reading. Perceiving the focus of social attention and facial attractiveness also affect interpersonal behaviour.

While communicating, we speak and at the same time, we usually use three of the senses – we hear, see and touch/feel. Hence, human communication has two main aspects: verbal and non-verbal. While words can be seen as the atomic information units of verbal communication, phenomena like facial expressions, vocal utterances, body movements and physiological reactions could be seen as the atomic units of non-verbal communication. It is quite clear that non-verbal communicative signals are not necessary for human-human interaction; a phone call is an example. Still, considerable research in social psychology has shown that non-verbal communicative cues can be used to synchronise the dialogue, to signal comprehension or disagreement and to let the dialogue run smoother and with less interruptions (Boyle et al. 1994, Stephenson et al. 1976). As indicated by Mehrabian (1968), whether the listener feels liked or disliked depends only for 7% on the spoken word, for 38% on vocal utterances, while facial expressions determine this feeling for even 55%[1]. This and the commonly used terms "face-to-face" and "interface" all indicate that facial expressions play an important role in human face-to-face interpersonal communication.

Besides their crucial role in the non-verbal aspect of human communication, facial expressions provide information about the observed person's attitudinal/affect state, age, attractiveness and gender, as well as about his/her personality, cognitive activity and psychopathology. Recent advances in image analysis and pattern recognition open up the possibility of automatic measurement of facial signals. Automated facial expression analysis could facilitate machine perception of human facial behaviour, bring facial expressions into man-machine interaction as a new modality making interaction more natural and more efficient (see section 1.1) and make classification and quantification of facial expressions widely accessible to research in behavioural science and medicine.

This thesis addresses various problems concerning the modelling, recognition and classification of the encountered facial expression kept in a digitised static facial image. Section 1.1 discusses the scope of this research and its main goal, outlining the key problems the resolution of which characterises this research. Section 1.2 provides the outline of this thesis.

1.1 The aim of this thesis: The ISFER project

Bruce (1992), Takeuchi and Nagao (1993) and Hara and Kobayashi (1997) pointed out that human interpretation of interpersonal face-to-face communication provides an ideal model for designing a multi-modal human-computer interface (HCI). As implied by the discussion above, the main characteristics of human interpretation of interpersonal face-to-face interaction are multiplicity and multi-modality of communication channels. A channel is a communication medium, while a modality is a sense used to perceive signals from the outside world. Examples of our communication channels are: the auditory channel that carries speech, the auditory

[1] Note that the percentages estimated by Mehrabian (1968) concern the effect that verbal and non-verbal communicative signals have on whether the listener feels liked or disliked. The given percentages do not define the extent to which the overall meaning of a communicated message is transmitted verbally, respectively, non-verbally. According to Birdwhistell (1970) and van Poecke (1996), 35 to 40% of the overall meaning of a communicated message is transmitted verbally and 60 to 65% is transmitted non-verbally.

channel that carries vocal intonation, the visual channel that carries facial expressions, and the visual channel that carries body movements. The senses of sight, hearing and touch are examples of modalities. In our usual face-to-face communication, numerous channels are employed and different modalities are activated. As a result, communication becomes highly flexible and robust. Failure of one channel is recovered by another channel and a message in one channel can be explained by another channel. This is how a multi-modal HCI should be developed for facilitating robust, natural, efficient, and effective man-machine interaction.

Nevertheless, relatively few existing works combine different modalities into a single fully-automated system for human communicative reaction analysis. Examples are the works of De Silva et al. (1997, 2000) and Chen et al. (1998), who studied the effects of a combined detection of facial and vocal expressions of emotions. So far, the majority of the studies treat various human communicative signals separately (Nakatsu 1998, Pantic and Rothkrantz 2001a). Examples of the presented systems are:

- automatic speech recognition (for an extensive review of the work in this field, the reader is referred to (Juang and Furui 2000)),
- automatic emotional interpretation of human voices (for a review of the work in this field, the reader is referred to (Pantic and Rothkrantz 2001a)),
- automatic emotion recognition by physiological signals pattern recognition (proposed in (Healey and Picard 1998)),
- detection and interpretation of hand gestures (for a review of the work in this field, the reader is referred to (Pavlovic et al. 1997)),
- recognition of body movements (for extensive reviews of the work in this field, the reader is referred to (Gavrila 1999, Cerezo et al. 1999, Pentland 2000)),
- detection and interpretation of facial expressions (for a review of the work in this field, the reader is referred to (Pantic and Rothkrantz 2000d)).

Since 1992, *Automated System for Non-verbal Communication* is an ongoing project at the Knowledge Based Systems group of the Delft University of Technology (van Vark et al. 1995). The goal of this project is the development of an intelligent automated system for the analysis of human non-verbal communicative signals (Figure 1.1). The system has to provide qualitative and quantitative information on different levels about various non-verbal signals sensed while monitoring a human subject. On the lowest level the system should detect a non-verbal signal given by the observed person. On the next level the system should categorise the detected signal as a specific facial action (e.g. smile), a specific body action (e.g. shrug), a specific vocal reaction (e.g. high speech velocity), or a specific physiological reaction (e.g. sweating). On a higher level the system should give an appropriate interpretation of the recognised communicative signals (e.g. an emotional interpretation). On the highest level the system should reason about the intentions of the user and (optionally) respond in a similar, anthropomorphic manner.

Hence, the intended intelligent analyser of human non-verbal communicative signals should be able to sense, process, interpret and animate non-verbal human communicative signals. The multi-modal input to the system should consist of multisensory data. Then, the detected vocal expression, facial expression, body movement, and physiological reaction should be analysed automatically. The final result of the system should represent a data fusion of the results of these analyses

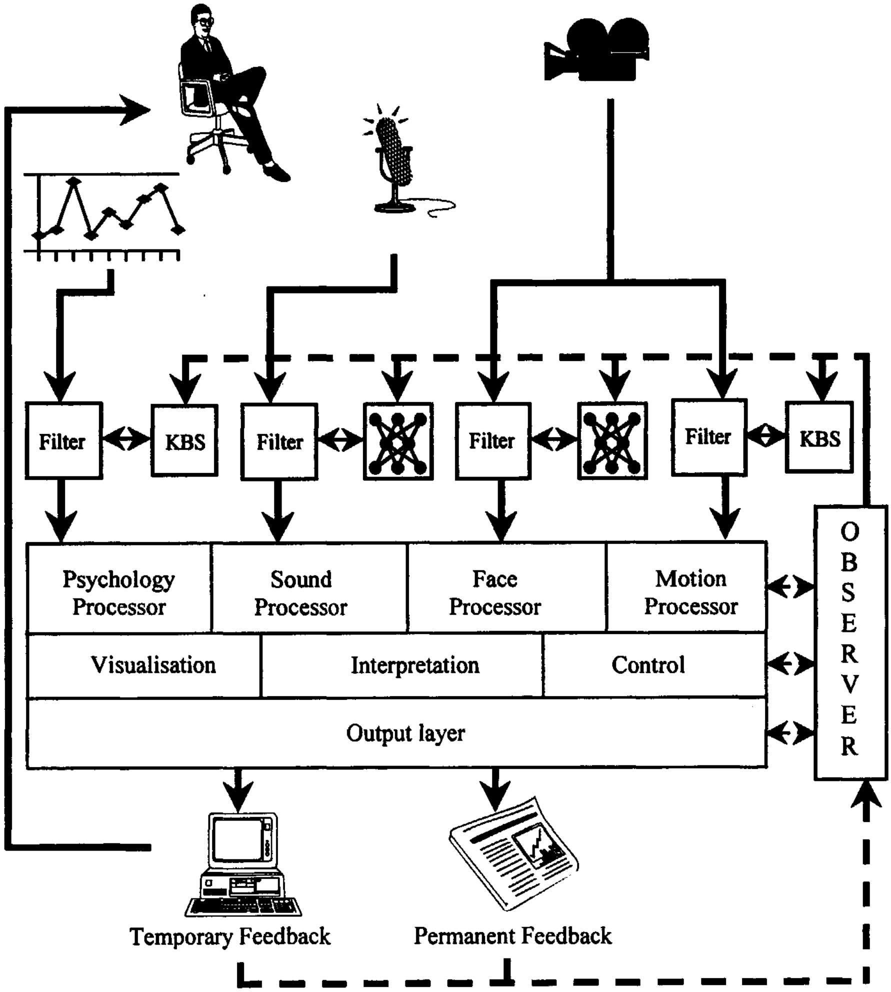


Figure 1.1: Automatic analysis of human non-verbal communicative signals

(performed, optionally, in parallel). This would form a hypothesis about the intentions of the currently monitored subject, whereupon the system should react properly in a user-friendly way (e.g. through the reactions of an animated virtual actor).

As a first step in achieving automatic analysis of human non-verbal communicative signals, automated analysis of facial expressions in digitised static facial images has been investigated. This thesis discusses the results of the research, which ensued in the development of the *Integrated System for Facial Expression Recognition* (*ISFER*). The ISFER-project was aimed at the design and implementation of a fully automated facial expression analyser that could be applied as an automated tool for behavioural investigations of the face. Since in behavioural research of the face full-face photographs of the observed subjects usually form the research material, the system was envisioned as being capable of analysing facial expressions from static facial images. In addition, the ISFER-project was aimed at the development of a system that could be (easily) enhanced to form a (front) part of an advanced multi-modal perceptive HCI.

The problem of automating facial expression analysis as defined in the ISFER-project comprises four sub-problems:

1. Automating the detection of the facial features in a digitised static facial image.
2. Automating the recognition of the encountered facial actions and their intensities.
3. Automating the affect-sensitive classification/interpretation of the observed facial actions.
4. Delimiting useful guidelines for enhancement of the system such that it can form a part of a multi-modal perceptive HCI.

1.2 Thesis overview

This study involves two research fields: facial expressions (i.e. psychological as well as computer-vision facets of the facial-expression-analysis problem domain) and Artificial Intelligence (AI). The thesis begins, therefore, with two introductory chapters. The other chapters explain the actual design and implementation of ISFER and provide sets of key challenges and opportunities which the researchers of machine perception of human behaviour face. Thus, the structure of this thesis is as follows:

Chapter 2 introduces the facial-expression-analysis issues – it provides the taxonomy of the problem domain, surveys the past work on solving these problems in an automatic way, and specifies the scope of the research pertaining to this thesis.

Chapter 3 gives an introduction to the field of AI, provides an assessment of the problem of automating facial expression analysis according to the AI paradigm and presents an overview of the AI techniques deployed in ISFER.

Chapter 4 discusses the first part of the system – the *Facial Data Extractor*. It is a framework for hybrid facial feature detection employed by ISFER for the extraction of facial expression information from input digitised static facial images.

Chapter 5 discusses the second part of the system – the *Facial Action Encoder*. This chapter pertains to the following issues: (i) modelling facial expressions, (ii) resolving the problems caused by redundant, inaccurate and partial data resulting from the Facial Data Extractor part of ISFER, and (iii) accomplishing reasoning with uncertainty about the displayed facial actions and their intensities that have produced the shown (input) facial expression.

Chapter 6 discusses the third part of the system – the *Facial Expression Classifier*. It is a learning facility of the system that achieves affect-sensitive interpretation of input facial expressions in terms of multiple quantified user-defined interpretation labels. This chapter presents the motivations for utilising a learning facility as well as the actual design and implementation of the Facial Expression Classifier part of ISFER.

Chapter 7 gives an overview of the overall performance of ISFER. Extended validation and evaluation studies suggest that the expressions' identifications and interpretations achieved by the system are satisfactory to the human observers which were involved in validation studies on ISFER.

Chapter 8 concludes the work on automating the facial expression analysis and points out some directions for future research in the field of machine perception of human behaviour. Special attention is paid to delimiting useful guidelines for development of an advanced multi-modal perceptive HCI and to discussing the usefulness and accessibility of such a HCI.

2 Facial expression analysis

Expressive people are easy to recognise but difficult to describe.
(Friedman et al. 1990)

There are marked individual differences in expressiveness. There are politicians who invigorate via their concern and passion, as well as the politicians who fail to inspire by their monotone speeches. Professors can be wearisome or eloquent, salesmen can be dull or slick. Not all of these differences are due to verbal fluency, but rather to a spirited communication which involves the use of facial expressions and body gestures. A speaker accompanies his utterances with appropriate facial expressions, which clarify what is being said; the non-verbal facial expression shows whether what is said is supposed to be important or funny or serious (Argyle 1972). Non-verbal facial cues help to establish the appropriate word meanings in any given circumstance (Friedman et al. 1990).

The human face serves not only a variety of different communicative functions in social interaction. Except information about a person's affective state, the face mediates information about personality, cognitive activity and psychopathology. In Aristotle's time, a theory has been proposed about mutual dependency between physiognomy and personality: "soft hair reveal a coward, short arms a gambler, and a smile a happy person"[1]. Today, few psychologists share the belief about the meaning of soft hair or short arms, but many believe that facial expressions are relative to emotions and psychopathology. For instance, there is a large body of psychological research that argues that emotions (at least so-called *basic emotions* – happiness, anger, sadness, disgust, surprise and fear) are universally recognised from facial expressions (Darwin 1965/1872, Izard 1971, Ekman 1980, Frijda 1986, Brown

[1] Although this theory is often attributed to Aristotle (Aristotle nd/1913), this is almost certainly not his work (Aristotle nd/1993, p. 83).

1991). Keltner and Buswell (1996) argue that even "social-moral emotions" like embarrassment and shame, which play critical roles in psychopathology, can be accurately distinguished by identifying related characteristic facial expressions. On the other hand there is a growing body of psychological research that argues that it is not emotions themselves but components of emotions which are universally linked with some facial displays like "squared" mouth or raised eyebrows (Ortony and Turner 1990, Russell 1994). Anyhow, it is certain that facial expressions play an important role in behavioural investigations, medicine and studies on social interaction.

Automating the analysis of facial expressions would therefore be highly beneficial for fields as diverse as behavioural science, medicine, monitoring, communications and education. Besides, if the goal is the design of human-like man-machine interaction, human face-to-face interpersonal communication (the verbal as well as the non-verbal aspect of it) provides an ideal model (Bruce 1992, Takeuchi et al. 1993, Schiano et al. 2000, etc.).

Although the interpretation of facial expressions is strongly situation and culture dependent (Russell and Fernandez-Dols 1997), humans detect and interpret facial expressions in a scene with little or no effort. Still, the development of an automated system that can accomplish this task is rather difficult.

There are several related issues: the detection of an image segment as a face, the extraction of the facial information, and the classification of the facial expression (e.g. in emotion categories). These issues as well as the capability of the human visual system to deal with them are discussed first (section 2.1). The capabilities of the human visual system are meant to serve as an ultimate goal and a guide for determining a set of recommendations for development of an automated facial expression analyser (section 2.2). This chapter further surveys the past work on solving these problems. Section 2.3 provides a short overview of the early works (proposed till 1995) on facial expression analysis by computer. Section 2.4 surveys recently developed systems (proposed in the period from 1996 to 2000). Finally, section 2.5 discusses the existing solutions and lists the main contributions of the work presented in this thesis to the research field of automatic facial expression analysis.

2.1 Aspects of facial expression analysis

The main goal here is to explore the issues in the design and implementation of a system that could analyse facial expressions automatically. In general, three main steps can be distinguished in tackling the problem. First, before a facial expression can be analysed, the face must be detected in a scene. The next step is devising mechanisms for extracting the facial-expression information from the observed facial image or image sequence. In the case of static images, the process of extracting the facial-expression information is referred to as *localising* the face and its features in the scene. In the case of facial image sequences, this process is referred to as *tracking* the face and its features in the scene. At this point, a clear distinction should be made between two terms, namely facial features and face model features. The *facial features* are the prominent features of the face: the eyebrows, eyes, nose, mouth and chin. The *face model features* are the features used to represent (model) the face. The face can be represented in various ways, e.g. as a whole unit (*holistic* representation), as a set of features (*analytic* representation) or as a combination of these (*hybrid* approach). The applied face representation and the kind of input images determine the choice of mechanisms for automatic extraction of facial-expression information. The final step is to define some set of categories, which we want to use for facial-expression classification and/or facial-expression interpretation, and to devise the categorisation mechanism.

Before an automated facial expression analyser can be built, one should decide what functionality the system should have. A good reference point is the functionality of the human visual system. After all, it is the best known facial expression analyser. This section discusses the three basic problems related to the process of facial expression analysis as well as the capability of the human visual system with respect to these.

Face position detection

In most works on automatic facial expression analysis, the conditions under which a facial image or image sequence is obtained are controlled. Usually, the image has the face in frontal view. Hence, the presence of a face in the scene is ensured and some global location of the face in the scene is known a priori.

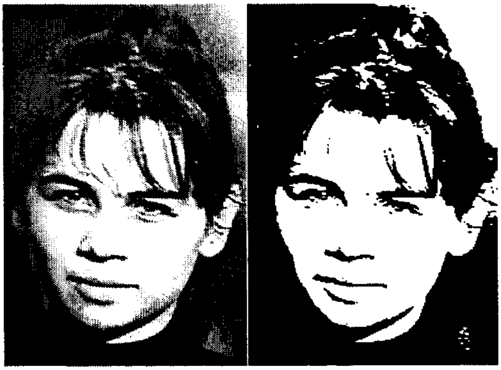


Figure 2.1: A human face at different gray levels – 256 gray levels and 2 gray levels

However, determining the exact location of the face in a digitised image is a more complex problem. First, the scale and the orientation of the face may vary from image to image. If the mug shots are taken with a fixed camera, faces in images may have different sizes and been taken at different angles due to the movements of the observed person. Thus, it is difficult to search for a fixed pattern (template) in the image. The presence of noise and occlusion makes the problem even more difficult.

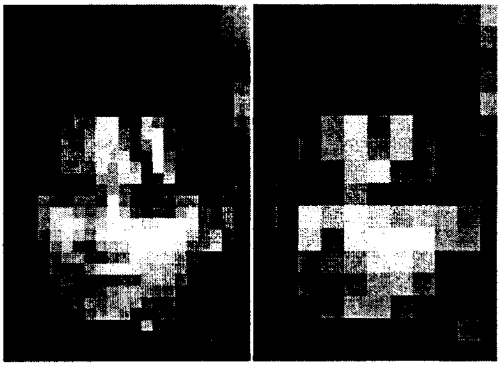


Figure 2.2: A human face at different spatial resolutions – 22x32 and 11x16 images of the same face

Humans detect a facial pattern by casual inspection of the scene. We detect faces effortlessly in a wide range of conditions, under bad lighting conditions or from a great distance. Figure 2.1 shows a face at different grey-level resolutions. In both cases, a human observer immediately notices the presence of a face. It suggests that for face detection, a 2-grey-level image is sufficient (Samal 1991). Figure 2.2 shows the same face at different spatial resolutions. The face can be detected rather easily in the 22×32 image. In the 11×16 image, however, there is a little resemblance to a face (although the face remains arguably detectable if one defocuses). It is generally believed that images of 100 to 200 pixels form a lower limit for the detection of a face by a human observer (Campbell and Green 1965). Another characteristic of the human visual system is that a face is perceived as a whole, not as a collection of facial features. When a face is partially occluded (e.g. by a hand), we perceive a whole face, as if our perceptual system fills in the missing parts. This is very difficult (if possible at all) to achieve by computer. There is also a strong perceptual bias towards seeing facial patterns. We often "see" faces in clouds, rocks and flames. An example is the result of Johansson's point-light display experiments (Bassili 1978, Bruce 1986) where the human observers were quickly aware that the white points visible on a dark monitor represent a face and its features. The presence of the features and their geometrical relationship with each other appears to be more important than the details of the features.

Facial-expression information extraction

After the presence of a face has been detected in the observed scene by an automated facial expression analyser, its next step is to extract the information about the encountered facial expression in an automatic way. If this information cannot be extracted automatically, a fully automated system cannot be developed. Both the applied face representation and the kind of input images (static images or image sequences) affect the choice of the approach to facial-expression-information extraction.

One of the fundamental issues of facial expression analysis is the representation of the visual information that the examined faces reveal (Yamada 1993). In other words, what kinds of visual properties can be used as information on facial expressions? Bassili (1978) and later Bruce (1986) conducted experiments that were similar to Johansson's point-light-display experiments and gave a clue to this problem. Bassili had the stimulus person make various facial expressions by having the face painted black with white marks put on it at random. His subjects observed only the movements of white marks through the monitor. They were quickly aware of seeing a face, and furthermore they could say what kind of facial expression the movement of the white marks represented. Bruce and Valentine (cited in (Bruce 1986)) also conducted similar experiments and found that their subjects judged facial expressions easily, but identified the person with great difficulty. These experiments suggest that certain patterns of the movements of various points on the face send expressional information independently of the information of the other cognitive domains (e.g. the person's identity, age, etc.). They also suggest that the visual properties of the face, regarding information about facial expressions, could be made clear by describing the movements of points belonging to the facial features (eyebrows, eyes, and mouth) and then by analysing the relationships between those movements. This inspired the researchers of vision-based facial-gesture analysis to attempt to determine point-based visual properties of facial expressions. This yielded various analytic face representations, in which the face is modelled as a set of facial points (e.g. Figure 2.3) or as a set of templates fitted to the facial features, such as the eyes and the mouth. In another approach to face representation (holistic approach), the face is represented as a whole unit. A 3D wire frame with a mapped texture (e.g. Figure 2.4) and a spatio-temporal model of facial-image motion are typical examples of the holistic approaches to face representation. The face can also be modelled using a so-called hybrid approach: a combination of analytic and holistic approaches to face representation. An example of this approach is the face model utilised by Thalmann et al. (1998). They use a geometric 3D wire frame with

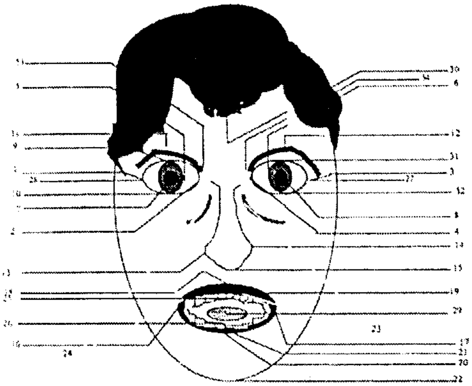


Figure 2.3: Facial "Landmarks" (Kearney and McKenzie 1993)

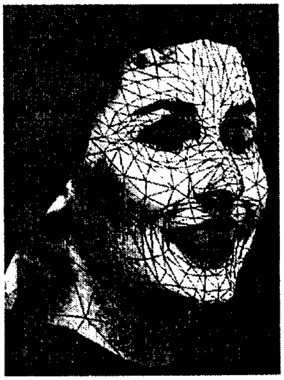


Figure 2.4: 3D mesh with texture mapped triangles proposed by Terzopoulos and Waters (1993)

the mapped texture constructed from two 2D templates containing the facial points from each orthogonal facial view (Figure 2.5).

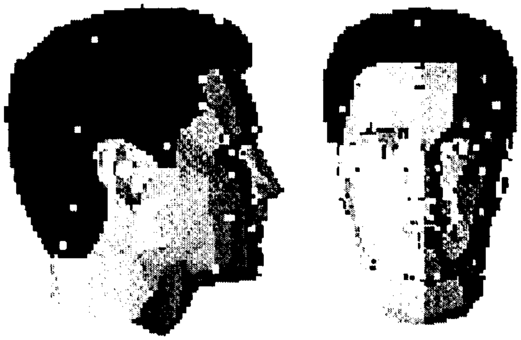


Figure 2.5: 2D templates (Thalmann et al. 1998)

Irrespectively of the kind of the face model applied, attempts must be made to keep the representation compact without losing any (or much) information about the observed facial expression. The nature of the representation affects and is affected by both the set of facial expressions and by the set of interpretation classes one deals with. The face must be represented so that a particular deformation of the face model uniquely reveals a particular facial expression. Complexity and completeness of the face representation determine the variety of expressions that can be recognised. On the other hand, the set of interpretation categories determines the set of facial expressions that should be recognised, which in its turn determines a minimal complexity required from the used face representation.

If an analytic approach has been utilised to represent the face, automatic detectors of facial features such as the eyes and mouth will be usually applied. If a holistic face model has been used, methods that fit the face model to the input image will be employed as facial-expression information extractors. Also, for each kind of input (static image or image sequence) different methods can be applied.

Several factors make facial-expression-data extraction more complex. The first is the presence of facial hair, glasses, etc., which obscure the facial features. Another problem is the variation in size and orientation of the face in input images. This disables a search for fixed patterns in the images. Finally, noise and occlusion are always present to some extent.

As indicated by Ellis (1986), human encoding of the visual stimulus (the face and its expression) may have the form of a primal sketch and may be hardwired. However, little is known in terms of the nature of internal representation of a face in the human brain.

Facial expression classification

After the face and its appearance have been perceived by an automated facial expression analyser, its next step is to "identify" the facial expression conveyed by the face. A fundamental issue in facial expression classification is defining a set of categories we want to deal with. A related issue is devising mechanisms of categorisation. Facial expressions can be classified in various ways: in terms of facial actions that cause an expression, in terms of some non-prototypic expressions such as "raised eyebrows" or in terms of some prototypic expressions such as emotional expressions.

The *Facial Action Coding System* (FACS) (Ekman and Friesen 1978) is probably the most known study on facial activity. It is a system that has been developed to facilitate objective measurement of facial activity for behavioural science investigations of the face. FACS is designed for human observers to detect independent subtle changes in facial appearance caused by contractions of the facial muscles. In a form of rules, FACS provides a linguistic description of all possible, visually detectable, facial changes in terms of 44 so-called *Action Units* (AUs). Using these rules, a trained human FACS coder decomposes a shown expression into the specific AUs that produced the expression. Automating FACS would make it widely accessible as a research tool in the behavioural science, which is furthermore the theoretical basis of multi-modal/media user interfaces. This inspired researchers in the computer-vision field to take different approaches to tackling the problem. Still, explicit attempts to automate the facial action coding so that it can be applicable to automated FACS coding are few (see Donato et al. (1999) or Bartlett et al. (1999) for a review as well as Table 2.7 of this chapter).

If facial expressions are to be classified in terms of facial actions, a method for automatic facial action coding from the input facial image (or image sequence) should be devised. The choice of method strongly depends on the utilised face representation. If a holistic face representation is used, a template-based method is usually applied. For each facial action an "ideal" deformation of the face model is learnt, which characterises a template for that facial action. The current deformation of the face model is then matched with the defined templates. In the case of an analytic face representation, a feature-based method is usually applied. Development of a successful feature-based method for automatic facial action coding is a quite complex task. A relationship between the deformations of the model-defined facial features and the set of facial actions to be recognised should be defined well. It should be made according to the rules dictated by the anatomy of the face. Two facts should be taken into consideration: some facial actions obscure some other facial actions (e.g. wrinkling of the nose obscures upward pull of the upper lip) and some facial actions can occur bilaterally as well as unilaterally (FACS, Ekman and Friesen 1978).

Most of the studies on automated expression analysis perform an emotional classification. As indicated by Fridlund et al. (1987), the most well-known and commonly used study on emotional classification of facial expressions is the cross-cultural study on the existence of *universal categories of emotional expressions*. Ekman defined six such categories, called *six basic emotions*: happiness, sadness, surprise, fear, anger and disgust (Ekman and Friesen 1975). He described each basic emotion in terms of a facial expression that uniquely characterises that emotion. In the past years, many questions arose around this study. Are the six basic emotional expressions indeed universal (Izard 1971, Ekman 1982, Ekman 1994) or do they merely emphasise verbal communication and have no relation with an actual emotional state (Fridlund 1991, Russell 1994)? Also, it is not at all certain that each facial expression that can be displayed by the face can be classified under the six basic emotion categories. Nevertheless, most of the research on vision-based facial expression analysis rely on the emotional categorisation of facial expressions defined by Ekman.

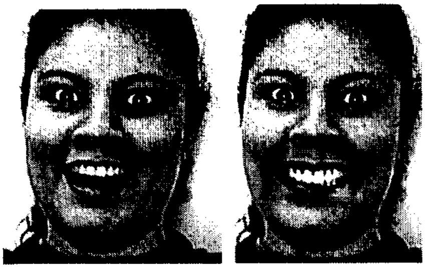


Figure 2.6: Facial expressions of blended emotions ("surprise" and "happiness")

Automating the facial expression classification in terms of emotions is difficult to handle for a number of reasons. First, Ekman's description of the six prototypic facial expressions of emotion is linguistic (and thus ambiguous). There is no uniquely defined description either in terms of facial actions or in terms of some other universally defined facial codes. Hence, the validation and the verification of the classification scheme to be used are difficult albeit crucial tasks. Second, facial expression classification into multiple emotion categories should be feasible (e.g. raised eyebrows and smiling mouth is a blend of surprise and happiness, Figure 2.6). Still, there is no psychological scrutiny on this topic. The best way of dealing with these problems is to develop a system which is independent of psychological studies and capable of adapting the facial-expression-classification mechanism according to a user-defined interpretation of facial expressions (Kearney and McKenzie 1993, Pantic and Rothkrantz 2000b).

Three other issues are related to facial expression classification in general. First, the system should be capable of analysing any subject, male or female of any age and ethnicity. In other words, the classification mechanism may not depend on physiognomic variability of the observed person. On the other hand, each person has his/her own maximal intensity of displaying a particular facial expression. Therefore, if the obtained classification is to be quantified (e.g. to achieve a quantified encoding of facial actions or a quantified emotional labelling of blended expressions), systems which can start with a generic expression classification and then adapt to a particular individual have an advantage. Second, it is important to realise that the interpretation of body language is situation dependent (Russell and Fernandez-Dols 1997). Nevertheless, the information about the context in which a facial expression is shown is very difficult to obtain in an automatic way. This issue has not been handled by currently existing systems. Finally, there is now a growing body of psychological research that argues that the timing of facial expressions is a critical factor for the interpretation of these expressions (Bassili 1978, Bruce 1986, Izard 1990). For researchers of automated vision-based expression analysis, this suggests that they should move towards a real-time whole-face analysis of facial-expression dynamics.

While human mechanisms for face detection are very robust, the same is not the case for human interpretation of facial expressions. It is often very difficult to determine the exact nature of the expression on a person's face. According to Bassili (1978), a trained observer can correctly classify facial photographs showing six basic emotions with an average of 87%. This ratio varies depending on several factors: the familiarity with the face, the familiarity with the personality of the observed person, the general experience with different types of expressions, the attention given to the face and the non-visual cues (e.g. the context in which the expression appears). It is interesting to note that the appearance of the upper face features (i.e. eyebrows and eyes) play a more important role in facial-expression interpretation than the appearance of the lower face features (Ekman 1982).

2.2 Automated facial expression analysis

Before developing an automated system for facial expression analysis, one should decide what functionality it should have. A good reference point is the best-known facial expression analyser: the human visual system. It may not be possible to incorporate all features of the human visual system into an automated system, and some features may even be undesirable, but it can certainly serve as a reference point.

The first requirement that should be met in the development of an ideal automated facial expression analyser is that all of the stages of the facial expression analysis are performed automatically:

1. face position detection,
2. facial-expression-information extraction,
3. facial expression classification.

Yet, actual implementation and integration of these stages into a system are constrained by the system's application domain. For instance, if the system is to be used as a tool for research in behavioural science, real-time performance is not an essential property of the system. On the other hand this is crucial if the system would form a part of a user-interface. Long delays make the interaction desynchronised and less efficient. Also, an explanation facility that would elucidate facial action encoding performed by the system might be useful if the system is employed to train human experts in using FACS. However, such facility is superfluous if the system is to be employed as a stress-monitoring tool or in videoconferencing. In this thesis, we are mainly concerned with two application domains of an automated facial expression analyser, namely behavioural science research and multi-modal/media user interface. In this section, an ideal automated facial expression analyser is proposed (Table 2.1) which could be employed in those application domains and has the properties of the human visual system.

As the potential applications of an automated facial expression analyser involve continuous observation of a subject over time, facial images should be acquired automatically. In order to be universal, the system should be capable of analysing subjects of both sexes, of any age and any ethnicity. No constraints should be set on the appearance of the observed subjects. The system should perform robustly despite changes in lighting conditions and distractions like glasses, changes in hair style, and facial hair like a moustache, beard or a unibrow. Like the human visual system, an ideal system would "fill in" missing parts of the observed face and "perceive" a whole face even when a part of it is occluded (e.g. by a hand). For the sake of the convenience of the observed subject, no special markers or make-up should be required for successful detection of the face and its features. In most real-life situations, complete immobility of the observed subject cannot be assumed. Hence the system should be able to deal with rigid head motions. Ideally the system should be capable of dealing with a whole range of head movements, from frontal view to profile view acquired by a fixed camera (see the discussion of Turk and Pentland (1991) on a robust system performance independently of viewing conditions). Constraints on rigid head motions can also be avoided by utilising a head-mounted camera.

An ideal system should perform a robust automatic face detection and facial-expression-information extraction from the acquired images or image sequences. Considering the state of the art in image processing techniques, inaccurate, noisy and missing data should be expected. An ideal system should be capable of dealing with these inaccuracies. In addition, the certainty of the extracted facial-expression data should be taken into account.

An ideal system should be able to analyse all visually distinguishable facial expressions. A well-defined face representation is a prerequisite for achieving this. The face representation should be such that a particular alteration of the face model uniquely reveals a particular facial expression. In general, an ideal system should be able to distinguish:

1. all possible facial expressions (a reference point is a total of 44 facial actions defined in FACS (Ekman and Friesen 1978), whose combinations form the complete set of facial expressions)
2. any bilateral or unilateral facial change,
3. facial expressions with a similar facial appearance (e.g. upward pull of the upper lip and nose wrinkling which also causes the upward pull of the upper lip).

In practice, it may not be possible to define a face model that can satisfy both reflect each and every change in facial appearance and whose features are detectable in a facial image or facial image sequence. Still, the set of distinct facial expressions that the system can distinguish should be as broad as possible.

If the system is to be used for behavioural-science-research purposes it should recognise facial expressions automatically in terms of FACS AU codes. As explained by Bartlett at al. (1999), this means that it should accomplish multiple quantified facial expression classification in terms of 44 AUs defined in FACS.

Table 2.1
Properties of an ideal facial expression analyser

	General Characteristic
1	Automatic facial-image acquisition
2	Any possible subject
3	Deals with variation in lighting
4	Deals with partially occluded faces
5	No special markers/make-up required
6	Deals with rigid head motions
7	Automatic face detection
8	Automatic facial-expression-data extraction
9	Deals with inaccurate facial-expression data
10	Automatic facial expression classification
11	Distinguishes all possible expressions
12	Deals with unilateral facial changes
13	Obeys anatomical rules (see FACS)

	Characteristic required by Behavioural science research
14	# of different AUs (from 44 in total)
15	Quantifies facial-action codes

	Characteristic required by Multi-modal/media HCI
16	Unlimited # of interpretation categories
17	Features adaptive learning facility
18	Assigns quantified interpretation labels
19	Assigns multiple interpretation labels
20	Features real-time processing

If the system is to be used as a part of an advanced multi-modal/media human-computer interface (HCI), the system should be able to interpret shown facial expressions (e.g. in terms of emotions). Since psychological researchers disagree about the existence of universal categories of facial displays of emotion, an ideal system should be able to adapt the classification mechanism according to the user's subjective interpretation of expressions, e.g. like suggested by Kearney and McKenzie (1993) and Pantic and Rothkrantz (2000b). Also it is not certain at all whether each and every facial expression that can be displayed by the face can be classified under one and only one emotion class. Think about blended emotional displays such as raised eyebrow and smiling mouth (Figure 2.6). This expression might be classified under two emotion categories defined by Ekman (1975): surprise and happiness. Yet, according to the descriptions of these prototypic expressions given by Ekman, the left-hand side facial expression shown in Figure 2.6 belongs "more" to the surprise than to the happiness class. For instance, in the left-hand side image the "percentage" of shown surprise is higher than the "percentage" of shown happiness while those percentages are approximately the same in the case of the right-hand side image. In order to obtain an accurate categorisation, an ideal expression analyser should perform quantified classification of facial expression into multiple emotion categories. Real-time performance is requisite, as already explained, for achieving fluent, tight and efficient man-machine interaction.

2.3 Early work in automatic facial expression analysis

Due to its importance for application domains of human behaviour interpretation and the human-computer interface, automatic facial expression analysis attracted the interest of many computer vision researchers. Since the mid 70s, different approaches have been proposed for automatic facial expression analysis from either static images or image sequences. In 1992, Samal and Iyengar gave an overview of the early works. Therefore, the systems for facial expression analysis proposed in the literature before 1991 are not discussed here. The reader can learn about those from the survey of Samal and Iyengar (1992). In this section a short overview of the systems for facial expression analysis proposed in the period of 1991 to 1995 is given, while the next section provides a more detailed survey of the systems developed recently, in the period of 1996 to 2000. Table 2.2 summarises the features of the "early" works with respect to the requirements posed on the design of an ideal facial expression analyser (see Table 2.1). All of the systems discussed in this section classify facial expressions in an automatic way but, none can recognise all facial expressions that can be displayed by the face, none can perform quantified facial-action coding, none can perform in real time (except the system proposed by Moses et al. (1995)), and none features an adaptive learning facility (except of the system proposed by Kearney and McKenzie (1993)). For this reason, these properties have been excluded from Table 2.2. ● stands for "yes", ✕ stands for "no" and ¯ represents a missing entry. A missing entry either means that the matter at issue has not been reported or that the matter at issue is not applicable to the system in question. An inapplicable issue, for instance, is the issue of dealing with rigid head motions and inaccurate facial data in the cases where the input data were hand measured (e.g. Kearney and McKenzie 1993).

Several systems can be classified as methods for facial expression analysis from static images. The first category of those works utilises a holistic face representation. Cottrell and Metcalfe (1991) use whole-face features, which they call *holons*, which are in fact facial pictures manually normalised and reduced to 64×64 pixels. Holons further form the input layer of a three-layer back-propagation neural network, trained to classify the input features into eight emotion categories (angry, miserable, bored, relaxed, sleepy, pleased, happy and astonished). Cottrel and Metcalfe reported that the network was much better at detecting some categories at the expense of others. Rahardja et al. (1991) also use a holistic data representation similar to the one used by Cottrell and Metcalfe, but the input images are hand-drawn faces with six different types of facial expressions (happy, sad, surprised, angry, afraid and neutral). They developed a pyramid-like feed-forward neural network for classification of hand-drawn facial expressions, which models the concept of hierarchical (multi-resolution) representation of image data. They reported that the network successfully classifies the images of the training data set but that the recognition of unknown drawings has not been evaluated and that the network performs rather poorly in classifying blurred or distorted images. Another neural-network approach to the classification of facial expressions into the six basic plus "neutral" emotion categories is proposed by Vanger et al. (1995). By manual procedures, they averaged all eye and mouth parts of 60 utilised static images of six basic prototypic expressions and created a prototype index for each emotion category. To classify a shown expression, they utilise a synergetic method to match the eye and mouth parts of the image to the prototype indexes. They claim that the recognition rate of their method is 70%. Matsuno et al. (1993) employ a holistic face model named the *Potential Net* for the recognition of four kinds of facial expressions (happy, angry, surprised and sad). A Potential Net consists of nodes, each of which is connected to four neighbour nodes through springs (see Figure 2.21). This net is set on a rectangular facial area, manually extracted from a static normalised full-face image. The net deforms by forces based on the smoothed grey-level value of the edge image, so that each node is moved to the position of facial features such as eyebrows, mouth and wrinkles. Matsuno et al. measure the movement of each node and use the displacement vectors to analyse the similarity between the vectors of the input image and the vectors of four facial-expression model nets. They tested the system on 44 unknown facial expression images of 11 people having no facial hair or glasses. The achieved successful recognition rate was 100% in the case of surprise, 90% in the case of anger and 70% in the case of sadness.

The second category of approaches to facial expression analysis from static images utilises an analytic face representation. As input to their 60×100×100×6 back-propagation neural network trained to classify facial expressions into one of the six basic emotion categories, Kobayashi and Hara (1992a) use 30 facial characteristic points (FCPs, Figure 2.7) hand-measured in facial images. They reported a correct classification ratio of 90% achieved by the trained network. They also experimented with recognising the strength of the shown facial expression (Kobayashi and Hara

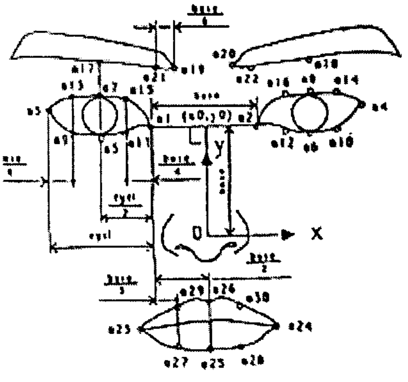


Figure 2.7: Facial Characteristic Points (Kobayashi and Hara 1992a, 1992b)

1992a) and with classifying facial expression into multiple categories (Kobayashi and Hara 1992b). They reported a correct classification ratio of 80% achieved by the 60×100×100×6 back-propagation network trained to classify facial expression in multiple emotion categories. To classify facial expressions under one of the emotion categories angry, happy and sad, Ushida et al. (1993) employ a multi-layered structure of bi-directional associative neural networks and as input the hand-measured FCPs used by Kobayashi and Hara (1992a). To reduce the quantity of input data, they take advantage of the face symmetry and use the FCPs belonging to the eyebrows, the right eye and the mouth. A shortcoming of this is that their method is not sensitive to unilateral appearance changes of the left eye. Ushida et al. reported a correct classification ratio of 79% achieved by their system. Kearney and McKenzie (1993) developed an expert system for the classification of facial expressions in one or more of the emotion categories defined by human observers. The system converts manually measured Facial Landmarks (Figure 2.3) into an intermediate facial-action-based representation, which a dynamic memory interprets further in terms of the defined emotion categories. The memory is dynamic in the sense that new emotion categories can be learned with experience. The production rules used for facial action coding are based on the rules defined for FAST (i.e. an early version of FACS, Ekman et al. 1971). Validation studies demonstrated that the facial action encoding achieved by the system is in 90% of the cases consistent with that of human experts. Those studies also suggest a correct classification ratio of 91.78% for the six basic emotion categories and 91.21% for learned categories.

Table 2.2
Properties of early approaches to automatic facial expression analysis

Reference	Properties of an ideal automated facial expression analyser (Table 2.1)														
	1	2	3	4	5	6	7	8	9	12	13	14	16	18	19
					Analysis from static facial images										
Cottrell '91	×	●	×	×	●	-	×	-	-	●	-	0	8	×	×
Rahardja '91	×	-	×	-	-	-	×	-	-	-	-	0	6	×	×
Vanger '95	×	●	×	×	●	-	×	×	-	●	●	0	7	×	×
Matsuno '93	×	×	●	×	●	-	×	×	-	●	●	0	4	×	×
Kobayashi'92	×	●	×	×	●	-	×	×	-	●	●	0	6	●	●
Ushida '93	×	●	×	×	●	-	×	×	-	×	●	0	3	×	×
Kearney '93	×	●	×	×	●	-	×	×	-	●	●	36	n	×	●
					Analysis from facial image sequences										
Mase '91	×	-	×	×	●	×	×	●	×	●	●	0	4	×	×
Yacoob '94	-	-	-	×	●	●	×	●	●	●	●	0	7	●	×
Rosenblum94	×	-	-	×	●	●	×	●	●	●	●	0	2	×	×
Moses '95	●	-	●	×	●	●	●	●	●	●	●	5	5	×	×

Legend: ● = "yes", × = "no", - = missing entry

Since there is a considerable body of psychological research that argues that the timing of facial expressions is a critical factor in the interpretation of facial expressions (Bassili 1978, Bruce 1986, Izard 1990) efforts have recently been made to analyse facial expressions by processing facial image sequences. Mase (1991) proposed a method based on optical flow data for the classification of facial expressions in one of the emotion categories: happiness, disgust, surprise and anger. He computes the optical flow of an image sequence with Horn-Schunck algorithm (Horn and Schunck 1981). He uses the means and covariances of optical flow data at evenly divided small blocks as the components of a feature vector. First, five feature vectors for each of the four emotional classes are derived from twenty labelled sample image sequences. Then the classification is performed based on the k-nearest-neighbour rule for the feature vector derived from the optical flow of the current image sequence. The method was tested on facial image sequences of one person and it does not deal with rigid head motions and changes in illumination. Mase reported a recognition ratio of 80% for this method.

Yacoob and Davis (1994a, 1994b) proposed a method for the recognition of 7 facial expressions (six basic expressions plus eye blinking) from image sequences which is also based on optical flow computation. For optical flow computation the correlation-based method proposed by Abdel-Mottaleb et al. (1993) has been used. The flow magnitudes are first thresholded to reduce the effect of small apparent motions due to noise. Then the inter-frame motion of edges (i.e. at points of high gradient values), extracted in the rectangles bounding the face regions of mouth, nose, eyes and eyebrows, is used to determine the facial action that may have occurred at the feature. A rule-based system based on the descriptions of the six basic emotional expressions (Ekman and Friesen 1975) is used to identify the "beginning", "epic" and "ending" of each facial expression. The rules are applied to the mid-level representation, given in terms of facial actions, to create a temporal map describing the evolving facial expression. On a sample of 46 image sequences of 32 subjects displaying a total of 105 various facial expressions, the system achieved an average correct recognition rate of 88% for the six basic emotion categories and 65% for eye blinking. Rosenblum et al. (1994) proposed a radial basis function network architecture that learns the correlation of facial-feature motion patterns and facial-expression emotion categories. This approach extends the work of Yacoob and Davis (1994b) and differs from the method presented in (1994a) by using a connectionist architecture instead of a rule-based system for motion patterns analysis. The three-layered network architecture proposed by Rosenblum et al. classifies facial expressions at emotion, facial-component and motion-direction level. At the emotion level, a separate network is trained for each of the six basic emotion categories. At the facial-component level, each emotion network is "broken" into sub-networks, where each sub-network specialises in a particular facial component: eyebrows, eyes or mouth. At the lowest level, the sub-networks are further decomposed so that the sub-sub-networks are sensitive to only one direction of motion (up, down, right or left) for a specific facial component for a specific emotion. Rosenblum et al. trained two emotion networks: one for the "smile" emotion category and one for the "surprise" emotion category. They tested the trained networks for retention (i.e. successful recognition of known image sequences), extrapolation (i.e. successful recognition of unknown image sequences) and rejection of image sequences that do not display the facial expression of emotion that the network was tuned for. They reported success rates of 88% for retention, 73% for extrapolation and 79% for rejection. A limitation of this method is the manual initialisation of the rectangles bounding the facial features (Rosenblum et al. 1994).

Moses et al. (1995) proposed a method for facial expression recognition based on real-time detection, tracking and classification of mouth deformation in image sequences. In their system, a valley in pixel intensity that lies between the lips describes the mouth. The valley contour is tracked using a Kalman filter designed to model the dynamics of a moving contour that is represented as a quadratic B-spline. The tracked valley contour is then used for classifying the shape of the mouth into five categories – neutral, sad, open, smile and pursed lips. A simple classification algorithm, using the linear approximation to the Bayesian classifiers, was applied for the discrimination between 5 different mouth shapes with an average correct recognition rate of 89%. Moses et al. showed that their algorithm is robust to changes in illumination, viewpoint and subject identity.

The approaches that have been explored lately also include systems for automatic analysis and synthesis of facial expressions that explicitly employ a physical model of the face. Such methods have been proposed by Terzopoulos and Waters (1993), Morishima et al. (1995), Kawakami et al. (1995), Li et al. (1993), Thalmann et al. (1995, 1998), (for a broader list of references on computer animation see also (Thalmann and Thalmann 1992)) and recently by Matsumura et al. (1997), DeCarlo et al. (1998), Eisert and Girod (1998), etc. Although the image analysis techniques in these systems are relevant to the present goals, the systems themselves are of limited use for behavioural science investigations of the face or for multi-modal/media HCI. These systems primarily concern facial expression animation and do not attempt to classify the observed facial expression either in terms of facial actions or in terms of emotion categories. For this reason these and similar methods lie outside of the scope of this thesis, the goal of which is the design of an image- and knowledge-based system for robust automatic facial-expression detection and classification.

2.4 The state of the art in automatic expression analysis

In this section, approaches to automatic facial expression analysis developed from 1996 till 2000 are surveyed in detail. The survey is divided into three parts, based on the problems discussed in section 2.1: face position detection, facial-expression-information extraction and facial expression classification. This section does not provide an exhaustive review of the past work on each of the problems related to automatic facial expression analysis. Here, recently developed systems which deal with both facial expression detection and classification are selectively discussed.

Table 2.3
Properties of the recently proposed approaches to automatic facial expression analysis

Reference	Properties of an ideal automated expression analyser (Table 2.1)													
	1	2	3	6	7	8	9	12	13	14	16	18	19	20
Analysis from static facial images														
Edwards '98	●	-	●	●	X	●	-	●	●	0	7	X	X	X
Hara '97	●	1	-	X	●	●	-	●	-	0	6	X	X	●
Hong '98	●	-	X	●	●	●	●	●	●	0	7	X	X	●
Huang '97	●	1	X	X	●	●	-	●	●	0	6	X	X	X
Lyons '99	X	●	X	-	X	X	-	●	●	0	7	X	X	X
Padget '96	X	●	X	-	X	X	-	●	●	0	7	X	X	X
Pantic 2000b	●	3	X	●	●	●	●	●	●	31	6	●	●	X
Yoneyama97	●	1	-	X	●	●	-	●	●	0	4	X	X	-
Zhang '98	X	●	X	-	X	X	-	●	●	0	7	●	●	X
Zhao '96	X	●	X	-	X	X	-	●	-	0	6	X	X	X
Analysis from facial image sequences														
Black '97	●	-	●	●	X	●	X	●	●	-	6	●	X	X
Cohn '98	●	3	X	X	X	●	X	●	●	15	-	X	X	-
Essa '97	●	●	●	-	●	●	●	●	●	2	4	X	X	●
Kimura '97	●	X	●	X	●	●	●	●	●	0	3	●	X	-
Otsuka '98	●	-	-	●	-	●	X	X	●	0	6	X	X	X
Wang '98	●	1	X	X	X	●	-	●	●	0	3	●	X	X

Legend: ● = "yes", X = "no", - = missing entry

Table 2.3 summarises the characteristics of the surveyed facial expression analysers with respect to the requirements posed on the design of an ideal facial expression analyser (Table 2.1). All of the systems discussed in this section classify facial expressions in an automatic way but, none can deal with images of partially occluded faces, none can recognise all facial expressions that can be displayed by the face, none can perform quantified facial-action coding, and none features an adaptive learning facility. For this reason, these properties have been excluded from Table 2.3. ● stands for "yes", X stands for "no" and - represents a missing entry. A missing entry either means that the matter at issue has not been reported or that the matter at issue is not applicable to the system in question. In the case of the systems where facial-expression information was manually extracted, the value of column 2 of Table 2.3 indicates that such systems can deal with the subjects of any ethnicity. In the case of an automatic facial-expression-data extraction, the value of column 2 of Table 2.3 represents the range in ethnicity of the test subjects. The number of test images, the number of subjects used to make the test images and the overall performance of the surveyed systems are summarised in Tables 2.7 to 2.9.

Face position detection

For most of the works in automatic facial expression analysis, the conditions under which an image is obtained are controlled. The camera is either mounted on a helmet-like device worn by the subject (e.g. Pantic et al. 2000b, Otsuka et al. 1998) or placed in such a way that the image has the face in frontal view. Hence, the presence of the face in the scene is ensured and some global location of the face in the scene is known a priori.

Table 2.4
Summary of the methods for automatic face position detection

	Reference	View	Method	Comments
Facial images				
Holistic approach	Huang ‘97	Frontal view	Canny edge detector PDM model fitting	No rigid movements
	Pantic 2000b	Dual view	Image histograms analysis Thresholding	Attached camera to the subject’s head
Analytic approach	Hara ‘97	Frontal view	Brightness distribution	No rigid movements Real-time process
	Yoneyama ‘97	Frontal view	-	-
	Kimura ‘97	Frontal view	Integral projection (see Wu et al. 1996) Potential Net fitting	No rigid movements
Arbitrary images				
Holistic approach	Hong ‘98	Frontal view	Steffens et al. (1998): Spatiotemporal filtering Stereo algorithm Colour detector Convex region detector Linear predictive filter	Complex background Slight head motions
	Essa ‘97	Frontal to profile view	Pentland et al. (1994): Spatiotemporal filtering Eigenfaces Eigenfeatures	Complex background Rigid head motions Faces with facial hair Faces with glasses Real-time process

In most of the real-life situations where an automated facial expression analyser is to be employed (e.g. in a multi-modal/media HCI), the location of a face in the image is not known a priori. Recently, the problem of automatic face detection in an arbitrary scene has drawn great attention. Examples are: the neural-network-based approach proposed by Rowley et al. (1998), the example-based learning approach of Sung and Poggio (1998), the colour- and invariant-moments-based method of Terrillon et al. (1998). Two of the works surveyed here deal with automatic face detection in an arbitrary scene: the method of Hong et al. (1998) and the method of Essa and Pentland (1997) (Figure 2.8).

Independently of the kind of input images (facial or arbitrary images) detection of the exact position of the face in an observed image or image sequence is approached in two ways. In the holistic approach, the face is detected as a whole unit. In the analytic approach, first some important facial features are detected (e.g. the irises and the nostrils). Then, the location of these features in correspondence with each other determines the overall location of the face. Table 2.4 provides a classification of facial expression analysers according to the kind of input images and the applied method.

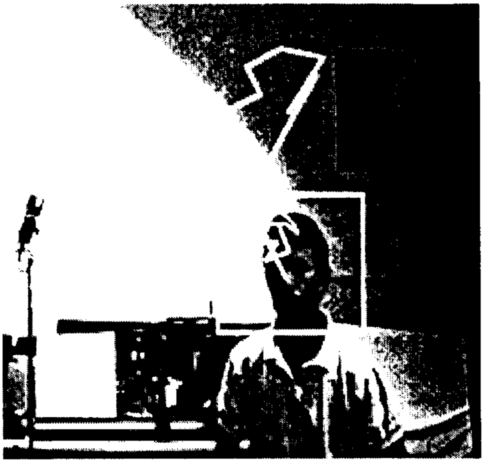


Figure 2.8: The path of the head tracked over several frames (Pentland et al. 1994)

Facial-expression-information extraction

After the presence of a face is detected in the observed scene, information about the shown facial expression should be extracted. Both the applied face representation and the kind of input images affect the choice of the approach to the facial-expression-information extraction.

Three face-representation types are mainly used in facial expression analysis: holistic, analytic and hybrid. In the holistic (template-based) approach, the template can be a 2D array of intensity values, a labelled graph or some other template that describes the properties of the face as a whole. The isodensity maps used by Kato et al. (1991) for analysis and synthesis of facial expressions form an example of holistic face representation. In the analytic (feature-based) approach, some facial points or contours of the prominent facial features (eyes, eyebrows and mouth) model the face. The relative sizes and shapes of the model features and the relative distances in between are then used for facial expression recognition. Examples of typical feature-based face models are: the Facial Landmarks proposed by Kearney et al. (1993) (Figure 2.3), the FCPs proposed by Kobayashi et al. (1992) (Figure 2.7) and deformable templates of the prominent facial features introduced by Yuille et al. (1989). A hybrid approach to face representation combines the feature-based approach and the template-based approach. In this approach, usually a set of facial feature points is used to determine an initial position of a template that models the face. A template can be a 3D wire frame, a labelled graph or a Potential Net. For example, to identify a face, Lam and Yan (1998) use an analytic-to-holistic approach which first locates 15 facial points, then sets boxes around the corresponding facial features and finally compares these with those in a database using a similarity transform. The face representations used by the surveyed automated expression analysers are described in this section and summarised in Table 2.5.

Depending on the used face representation, a template-based or a feature-based method is applied to extract expression information from an input static image or an input image sequence. Template-based methods fit a holistic face model to the input (static) image or track it in the input image sequence. Feature-based methods are used to localise the features of an analytic face model in the input image or to track them in the input image sequence. The methods utilised by the surveyed automated expression analysers are summarised in Table 2.6.

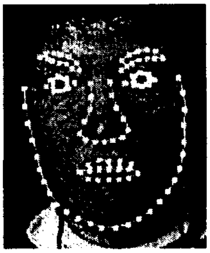


Figure 2.9: Facial landmark points (Edwards et al. 1998)

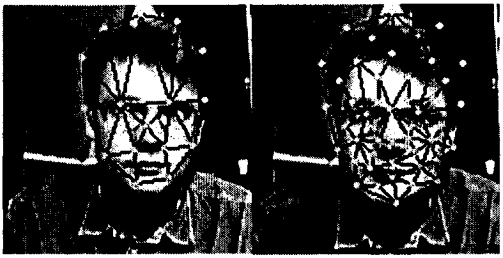


Figure 2.10: A small model graph and a dense model graph (big GFK) (Hong et al. 1998)

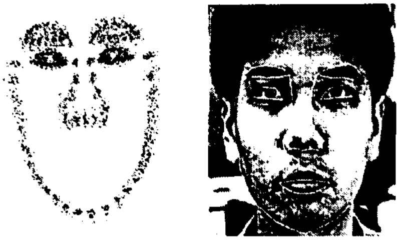


Figure 2.11-12: Aligned training set for generation of a PDM model and an example of the fitted model (Huang and Huang 1997)

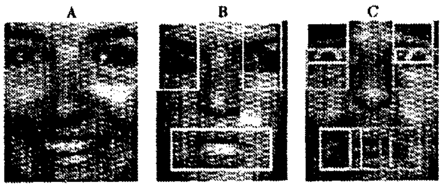


Figure 2.13: A) An example of the used images B) The feature regions from which the eigenvectors are calculated C) The 32x32 pixel blocks (Padgett and Cottrell 1996)

Table 2.5
Summary of the face models utilised by the recently proposed facial expression analysers

Ref.	Model	Figure	Comment
Holistic approach			
Edwards ‘98	Active Appearance Model (AAM) (Cootes et al. ‘98)	2.9	Manual localisation of 122 facial points
Hong ‘98	Labelled graph (GFK); each node is an array of the filter responses of a certain Gabor wavelet extracted at an image point (Lyons et al. 1998)	2.10	The “big” labelled graph (GFK) with 50 nodes containing 40-component arrays seems very suitable for detection of the facial actions; this issue is not discussed by Hong (1998)
Huang ‘97	Point Distribution Model (PDM) (Kass et al. 1987, Cootes et al. 1995)	2.11-2.12	Manual localisation of 90 facial points Mouth model does not support detection of some mouth actions
Padgett ‘96	Random block eigenvectors defined from 97 images taken from Ekman’s database (Ekman 1975)	2.13	Image format strictly constrained All used images are not real-life shots ⇒ applicability in real-life situations is not proven
Black ‘97	Optical flow (in fac. regions)	2.18	Initial regions for the head and facial features are manually selected
Otsuka ‘98	Optical flow (in fac. regions)	2.19	No modelling of the left eye
Analytic approach			
Hara ‘97	30 facial points & 13 vertical lines across them	2.15	Horizontal facial-appearance changes (e.g. frown) aren’t modelled
Pantic ‘00b	Dual-view point-based model	2.16	
Zhao ‘96	Frontal-view point-based model of 10 facial distances	2.17	Manual localisation of 10 facial distances
Cohn ‘98	Optical flow (facial points)	2.23	Manual localisation of 45 facial points Image sequence should start with an expressionless face
Hybrid approach			
Lyons ‘99	Labelled garaph of 34 nodes (see also Zhang ’98)	2.14	Image format strictly constrained Manual localisation of 34 facial points
Yon. ’97	8x10 quadratic grid		
Zhang ‘98	Labelled graph with 34 nodes	2.14	Image format strictly constrained Manual localisation of 34 facial points
Essa ‘97	Optical flow (whole face)	2.20	2D spatio-temporal representation of the facial frontal view
Kimura ‘97	Potential Net: 2D mesh which deforms governed by the elastic force of the potential field (Table 2.6)	2.21	Potential Net seems suitable for facial-action detection independently of the face rotation; this issue is not discussed by Kimura (1997)
Wang ‘98	Labelled graph with 19 nodes	2.22	Manual location of 19 facial points

Table 2.6
Summary of the methods for automatic facial-expression-data extraction

Ref.	Method	Comment
Analysis from static facial images		
Template-based methods		
Edwards '98	Multivariate multiple regression is applied to model the relationship between AAM displacement and the image difference.	Direct frontal view No facial hair / glasses Hand labelling of the images
Hong '98	Fitting a labelled graph (Figure 2.10) to an input image by utilising the method of elastic graph matching (Wiskott 1995).	No facial hair / glasses Slightly rotated faces allowed Real-time process
Huang '97	Fit the PDM (Figure 2.12) by a gradient-descent shape parameters estimation; fit three parabolas onto the mouth by gradient-based edge detector.	Direct frontal view No facial hair / glasses No variation of the background
Yon. '97	Gradient-based optical flow alg. (Horn et al. 1981) for estimating an averaged optical flow in 80 20×20 pixels regions of the grid placed in a normalised image.	Direct frontal view No facial hair / glasses Averaging the flow (drawback) Horizontal movem. not modelled
Feature-based methods		
Hara '97	Extracting the brightness distribution data along the 13 vertical facial lines (Figure 2.15) by utilising a CCD camera in monochrome mode.	Direct frontal view No facial hair / glasses Horizontal movem. not modelled Real-time process
Pantic 2000b	Multiple detectors are applied per facial feature. Model features (Figure 2.16) are then extracted from the localised contours.	Dual view images No facial hair / glasses 2 mounted cameras
Analysis from facial image sequences		
Template-based methods		
Black '97	Robust regression based on a brightness constancy assumption for image motion local models' parameters recovering. Coarse-to-fine gradient-based optical flow algorithm for estimating large motions.	Rigid head motions allowed Variations in lighting allowed The initial regions of the head and features are selected by hand
Otsuka '98	Adapted gradient-based optical flow algorithm (Black et al 1995) for estimating image motion in local areas of right eye and mouth (Figure 2.19).	No facial hair / glasses Head-mounted camera is used Left eye motion is not tracked
Essa '97	Optical flow method (Simoncelli 1993): multi-scale coarse-to-fine Kalman filter for obtaining "noise-free" 2D motion field for a normalised image (Figure 2.20).	Direct frontal view Faces with facial hair / glasses Variations in lighting allowed
Kimura '97	Fitting a Potential Net (Figure 2.21) to a normalised facial image by applying a differential and a Gaussian filter.	Direct frontal view No facial hair / glasses First frame - expressionless face

Wang '98	Fitting a labelled graph of 19 facial points (Figure 2. 22) by applying the method proposed by Buhmannet al. (1989): minimisation of the cost function based on simulated annealing procedure.	Direct frontal view No hair / glasses Hand labelling of the first frame
Feature-based methods		
Cohn '98	Hierarchical optical flow algorithm of Lucas et al. (1981) for estimating the optical flow in 13×13 pixels facial regions	Direct frontal view No facial hair / glasses First frame - expressionless face Manual normalisation Hand labelling of the first frame

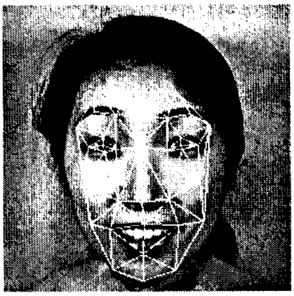


Figure 2.14: Fiducial grid of acial points (Zhang et al. 1998)

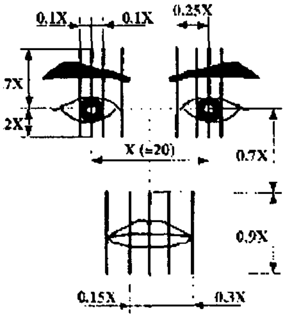


Figure 2.15: Thirteen vertical lines (Hara and Kobayashi 1997a)

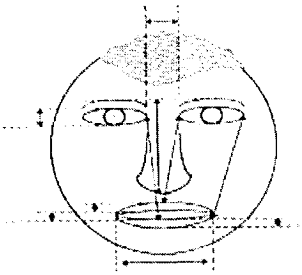


Figure 2.17: Facial distances (Zhao and Kearney 1996)

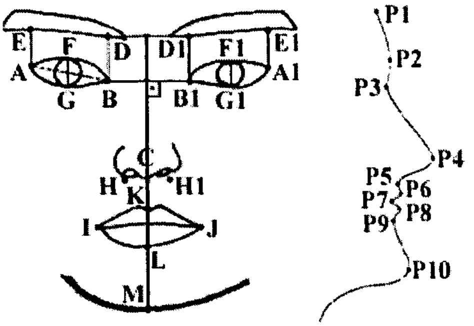


Figure 2.16: Facial landmark points of the dual-view face model (Pantic and Rothkrantz 2000b)

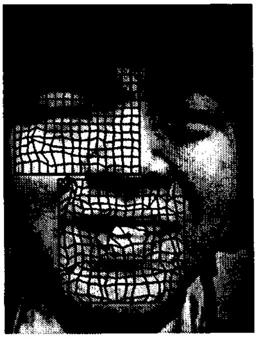


Figure 2.19: Motion-vector field epresented in the deformation of wo grids (Otsuka and Ohya 1998)

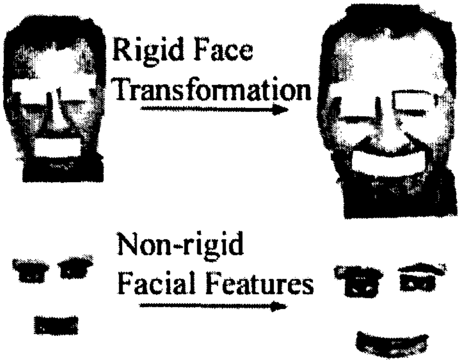


Figure 2.18: Planar model for representing rigid face motions and affine & curvature model for representing non-rigid motions of the facial features (Black and Yacoob 1997)

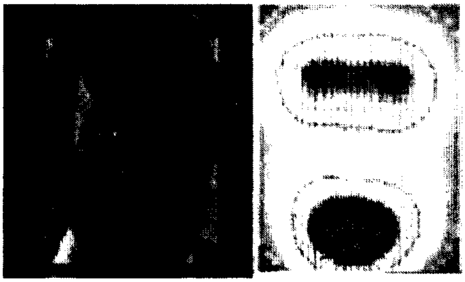


Figure 2.20: The spatio-temporal motion-energy representation of facial motion for surprise (Essa and Pentland 1997)

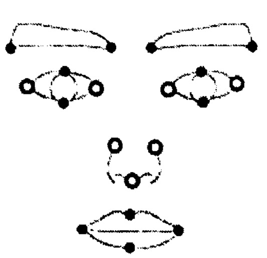


Figure 2.22: The FFPs (Wang et al. 1998)

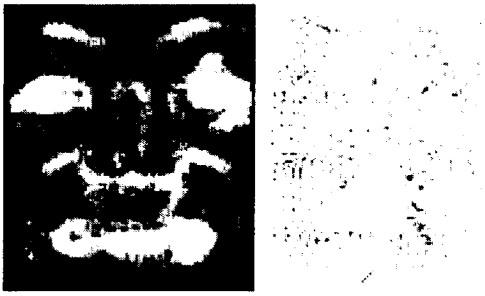


Figure 2.21: Potential Field and corresponding Potential Net (Kimura and Yachida 1997)

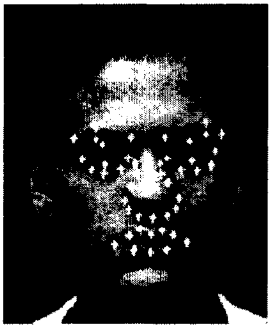


Figure 2.23: Facial landmark points (Cohn et al. 1998)

Facial expression classification

The last step of facial expression analysis is to classify (identify, interpret) the facial expression displayed by the face. A fundamental issue about the facial expression classification is to define a set of categories we want to deal with. A related issue is to devise a categorisation mechanism. As already explained in section 2.2, the actual design and implementation of an automated expression classifier is constrained by its application domain. If the system is to be used for behavioural science investigations of the face, the system should realise automatic encoding and quantification of facial actions from facial images or image sequences. If the system is to be used as an integral part of intelligent multi-modal/media HCI, the system should realise automatic quantified facial expression classification into the multiple interpretation categories defined by the user.

The surveyed facial expression analysers classify the encountered expression (i.e. the extracted facial-expression information) either as a particular facial action or a particular basic emotion. Some of the analysers perform both: encode the involved facial actions and classify these under the basic emotion categories. Independently of the used classification categories, the mechanism of classification applied by particular surveyed expression analyser is either a template-based, a neural-network-based, or a rule-based classification method. The applied methods for expression classification in terms of facial actions are summarised in Table 2.7. Table 2.8 and Table 2.9 summarise the utilised methods for facial-expression emotional classification.

If a template-based classification method is applied, the encountered facial expression is compared to the templates defined for each expression category. The best match decides the category of the shown expression. In general, it is a difficult to achieve a template-based quantified recognition of a non-prototypic facial expression (i.e. a certain combination of facial actions and their intensities). There are infinitely a lot of combinations of different facial actions and their intensities that should be modelled with a finite set of templates. The problem becomes even more difficult by the fact that everybody has his/her own maximal intensity in displaying a certain facial action.

Although the neural networks represent a "black-box" approach and could be, arguably, classified as template-based methods, the neural-network-based methods have been classified separately. The reason for doing so is that a typical neural network can perform a quantified facial-expression categorisation into multiple classes while, in general, the template-based methods cannot. In a neural-network-based classification approach, a facial expression is classified according to the categorisation process that the network "learned" during a training phase. Most of such methods utilised by the surveyed expression analysers classify facial expressions into a single category. Rècognition of non-prototypic facial expressions is feasible, however, if each neural-network output unit is associated with a weight from the interval [0,1], instead of being associated with either 0 or 1 (e.g. Kobayashi and Hara 1997, Zhao and Kearney 1996). Zhang et al. (1998), Kobayashi and Hara (1992), Ralescu and Hartani (1995), and Morishima et al. (1995) proposed different neural networks for the recognition of blended emotional expressions). As can be seen from Table 2.8, some of the expression classifiers have been classified as template-based methods although they utilise a neural network (e.g. Yonoyama et al. 1997). This has been done because the overall characteristics of these methods fit better the overall properties of the template-based expression classification approaches.

The rule-based classification methods, utilised by the surveyed expression analysers, classify the examined facial expression into the basic emotion categories based on the previously encoded facial actions (Table 2.7, Table 2.8 and Table 2.9). The expressions, which characterise the emotion categories, are first described in terms of the facial-action codes. Then the shown expression, described in terms of facial-action codes, is classified in the optimal fitting emotion category.

Table 2.7
Summary of the methods for expression classification in terms of facial actions

Ref.	Method	No. of AUs	Test cases	Comment
		Analysis from static facial images		
Rule-based methods				
Pantic 2000b	Expert System rules	30 facial actions	496 dual views 8 subjects Correct: 89%	Does not deal with minor inaccuracies No quantified AU encoding
		Analysis from facial image sequences		
Template-based methods				
Cohn ‘98	Discriminant functions	8 AUs + 7 AUs combinations	504 sequences 100 subjects Correct: 88%	One AU allowed per sequence; No quantified AU encoding
Essa ’95 & ’97	Spatio-temporal motion-energy templates	2 facial actions	22 sequences 8 subjects Correct: 100%	Faces with facial hair allowed; No quantified AU encoding
Rule-based methods				
Black ‘97	Thresholded motion parameters	-	70 sequences 40 subjects Correct: 88%	Number of facial actions that can be encoded is not known No quantified AU encoding

Table 2.8

Summary of the methods for facial-expression emotional classification from static facial images into some of the basic emotion categories as defined by Ekman (1975)

Ref.	Method	#	Test cases	Comment
Template-based classifiers				
Edwards ‘98	Mahalonobis-distance-based PCA (Hand 1981) and Linear Discriminant Analysis (LDA)	7	200 images 25 subjects Correct: 74%	Hand labelling of the images Not tested for unknown subjects Singular classification No quantified classification
Hong ‘98	Personalised galleries and Elastic graph matching (Wiskott 1995)	7	>175 images 25 subjects Correct: 81%	Processing time per image: 8 s Singular classification No quantified classification
Huang ‘97	2D emotion space (PCA) & minimum distance classifier	6	90 images 15 subjects Correct: 85%	Not tested for unknown subjects Singular classification No quantified classification
Lyons ‘99	PCA and LDA of the labelled-graph vectors	7	193 images 9 Jap females Correct: 92%	Low diversity of tested subjects Singular classification No quantified classification
Yon. ‘97	Two 14x14 Hopfield NNs with Personnaz learning (Kanter et al. 1987)	4	40 images 10 subjects Correct: 92%	Not tested for unknown subjects Singular classification No quantified classification
Neural-network based classifiers				
Hara ‘97	234x50x6 back-propagation NN	6	90 sequences 15 subjects Correct: 85%	Processing time per frame: 66.7 ms Singular classification No quantified classification
Padgett ‘96	15x10x7 back-propagation NN	7	84 Ekman’s photos Correct: 86%	Real-life mug-shots not tested Singular classification No quantified classification
Zhang ‘98	646x7x7 resilient RPROP propagation (Riedmiller et al. ‘93)	7	213 images 9 Jap females Correct: 90%	Hand labelling of the images Low diversity of tested subjects Multiple quantified classification
Zhao ‘96	10x10x3 back-propagation NN	6	94 Ekman’s photos Correct: 100%	Real-life mug shots not tested Singular classification No quantified classification
Rule-based classifiers				
Pantic 2000b	Expert system rules	6	256 images 8 subjects Correct: 91%	Multiple quantified classification

Table 2.9
Summary of the methods for facial-expression emotional classification from facial image sequences into some of the basic emotion categories as defined by Ekman (1975)

Ref.	Method	#	Test cases	Comment
Template-based classifiers				
Essa '95 & '97	Spatio-temporal motion-energy templates (see Figure 2.24)	4	30 sequences 8 subjects Correct: 98%	Faces with facial hair allowed Singular classification No quantified classification
Kimura '97	3D emotion space (PCA)	3	6 sequences 1 subject	The method tested unsuccessfully Singular quantified classification
Otsuka '98	HMM with the Baum-Welch training method	6	120 sequences 2 subjects	No description of the test results Singular classification No quantified classification
Wang '98	Averaged B-splines of feature trajectories (Press et al. 1992) for distance minimisation	3	29 sequences 8 subjects Correct: 95%	Processing time per frame: 2.5 s Hand labelling of the first frame Singular quantified classification
Rule-based classifiers				
Black '97	Rule-based approach for detection of the beginning and the end of an expression	6	70 sequences 40 subjects Correct: 88%	One expression per sequence Rules not well defined ⇒ blend anger-fear recognised as disgust Singular quantified classification

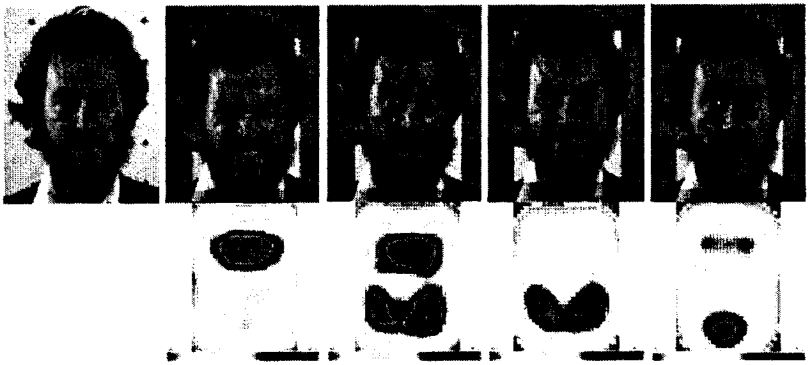


Figure 2.24: Motion energy templates for anger, disgust, smile and surprise (Essa and Pentland 1997)

2.5 Discussion

Analysis of facial expressions is an intriguing problem which humans solve with quite apparent ease. Three different but related aspects of the problem have been defined: face position detection, facial-expression-information extraction and facial expression classification. The capability of the human visual system in solving these problems has been discussed in section 2.1. It should serve as a reference point for any automatic vision-based system attempting to achieve the same functionality.

Since the mid 70s, different approaches have been proposed for the analysis of facial expressions from facial images and image sequences. In 1992, Samal and Iyengar (1992) issued an overview of the early works on solving the problems related to automatic analysis of facial expressions. Section 2.3 provided an overview of the literature on solving these problems published in the period from 1991 to 1995. The work done in the past five years on solving these problems as a whole is surveyed in section 2.4 in detail and then summarised.

A number of different facial-image-analysis approaches to facial-expression detection and classification have been explored and compared. These approaches include facial expression analysis from facial image sequences and from static facial images (Table 2.3). The investigation compared the automatic expression-information extraction using facial-motion analysis (Black and Yacoob 1997, Cohn et al. 1998, Lien et al. 1998, Essa and Pentland 1997, Otsuka and Ohya 1998), holistic spatial-pattern analysis (Edwards et al. 1998, Hong et al. 1998, Huang and Huang 1997, Yoneyama et al. 1997, Kimura and Yachida 1997, Wang et al. 1998), and analysis of facial features and their spatial arrangement (Kobayashi and Hara 1997, Pantic and Rothkrantz 2000b, Cohn et al. 1998). This investigation also compared the facial expression classification using holistic spatial analysis (Edwards et al. 1998, Hong et al. 1998, Huang and Huang 1997, Lyons et al. 1999, Yoneyama et al. 1997, Padgett and Cottrell 1996), holistic spatio-temporal analysis (Black and Yacoob 1997, Essa and Pentland 1995, Essa and Pentland 1997, Kimura and Yachida 1997, Otsuka and Ohya 1998, Wang et al. 1998), grey-level pattern analysis using local spatial filters (Lyons et al. 1999, Zhang et al. 1998), and analytic (feature-based) spatial analysis (Huang and Huang 1997, Kobayashi and Hara 1997, Zhao and Kearney 1996, Pantic and Rothkrantz 2000b). The number of surveyed systems is rather large and the reader might be interested in the results of the comparison in terms of the best performances. Nevertheless, ranking surveyed systems based on their quality has not deliberately been made. I believe that a well-defined and commonly used single database of test images (image sequences) is a necessary prerequisite for ranking the performances of the proposed systems in an objective manner. Since such a single test dataset has not been established yet, the reader is left to rank the discussed systems according to his/her own priorities and based on the overall properties of surveyed systems that have been summarised in Tables 2.3 to 2.9.

In this section, some possible directions for future research are proposed. Those originate from a comparison of the properties of surveyed facial expression analysers with the properties of an ideal analyser proposed in section 2.2.

Detection of the face and its features

Most of the currently existing systems for facial expression analysis assume that the presence of a face in the scene is ensured. However, in many instances the systems do not perform face detection in an arbitrary scene and do not utilise a head-mounted camera which ascertains the correctness of the assumption at issue. Only two surveyed systems process images acquired by a head-mounted camera (Otsuka and Ohya 1996, Pantic and Rothkrantz 2000b) and only two systems deal with automatic face detection in an arbitrary scene (Hong et al. 1998, Essa and Pentland 1997).

In addition, many approaches use strong assumptions to make the problem of facial expression analysis more tractable (see Table 2.6). Some common assumptions are:

- the images contain frontal facial view;
- the face is upright with no tilt;
- the illumination is constant;
- the light source is fixed;
- the face has no facial hair or glasses;
- the subjects are young (i.e. no permanent wrinkles) and of the same ethnicity.

In most of the real-life situations it cannot be assumed that the observed subject will remain immovable. Therefore, if a fixed camera acquires the images, the system should be capable of dealing with rigid head motions. Only three of the surveyed systems deal to some extent with rigid head motions (Black and Yacoob 1997, Edwards et al. 1998, Hong et al. 1998).

For the sake of universality, the system should be able to analyse facial expressions of any person, independently of age, ethnicity and outlook. Yet only the method proposed by Essa and Pentland (1997) deals with images of faces with facial hair and/or glasses.

For researchers of automated vision-based facial expression analysis this suggests investigation towards developing a robust method for the detection of the face and its features that will not be prone to changes in viewing and lighting conditions and distractions like glasses, facial hair or changes in hair style. Another interesting but yet not investigated ability of human visual system is "filling in" missing parts of the observed face and "perceiving" a whole face even when a part of it is occluded (e.g. by a hand).

Facial expression classification

Generally, the existing expression analysers classify the examined facial expression in merely one of the basic emotion categories proposed by Ekman and Friesen (1975). This approach to expression classification has two main limitations.

First, "pure" emotional expressions are seldom elicited. Most of the time people show blends of emotional expressions. Therefore, classification of an expression into a single emotion category is not realistic. An automated facial expression analyser should realise quantified classification into multiple emotion categories. Only two of the surveyed systems, namely those of Pantic and Rothkrantz (2000b) and Zhang et al. (1998), perform quantified facial-expression classification into multiple basic-emotion categories.

Secondly, it is not at all certain that all facial expressions that can be displayed by the face can be classified under the six basic emotion categories. So even if an expression analyser performs a quantified expression classification into multiple basic emotion categories, it would probably not be capable of interpreting each and every encountered expression. A psychological discussion on the topic can be found in Izard (1971), Fridlund (1991), Russell (1994) and Ekman (1994). Some experimental proofs can be found in the studies of Asian researchers (e.g. Huang and Huang 1997, Zhang et al. 1998), which reported that their Asian subjects have difficulties in expressing some of the basic expressions (e.g. disgust and fear).

Defining interpretation categories into which any facial expression can be classified is one of the key challenges in the design of a realistic facial expression analyser. The lack of psychological scrutiny on the topic makes the problem even harder. A way of dealing with this problem is to build a system that acquires its own expertise by learning from the user his/her interpretations of facial expressions. Kearney and McKenzie (1993) proposed a way of achieving this. This thesis proposes and elaborates another method for achieving a generally applicable facial-expression classification into the expression-interpretation categories defined by the user (see chapter 6).

If the system is to be used for behavioural science investigations of the face, it should perform automated FACS coding of input facial expressions. In other words, it should accomplish both: discern various AUs in input images and quantify those codes (Donato et al. 1999, Bartlett et al. 1999). Four surveyed systems perform facial action coding in an input image or an image sequence (Table 2.7). Yet none of these systems quantifies the facial action codes. This task is particularly difficult to accomplish for a number of reasons. First, FACS only provides five different AUs which can be assigned an intensity on a 3-level intensity scale (i.e. low, medium, and high). Further, some facial actions such as blinking, winking, and sucking the lip(s) into the mouth are either encountered or not. It is unreasonable to describe a blink as "having a higher intensity" than another blink. In addition, each person displays a particular facial action with a different maximal intensity. Therefore a system should be designed that can start with a generic facial action classification and then adapt to a particular individual to quantify the encoded facial actions for which measuring of the activation intensity is "reasonable". Also, none of the surveyed systems is capable of distinguishing all 44 facial actions defined in FACS (Ekman and Friesen 1978). This remains a key challenge for the researchers of automated FACS coding.

Another appealing but still not investigated property of the human visual system concerns assigning a higher "priority" to the upper-face features than to the lower-face features, since they play a more important role in facial expression interpretation (Ekman, 1982).

2.6 A new approach: ISFER

Facial expressions provide information about the affective state, personality, cognitive activity and psychopathology. Besides, they play a main role in non-verbal human communication (Mehrabian 1968). An automated system for facial expression analysis would therefore be highly beneficial for applications such as the behavioural science investigations of the face, stress monitoring at hazardous work places (e.g. aeroplane cockpit, nuclear power plant control room), education (e.g. medical), enhancement of communication skills, and development of advanced multi-modal human-computer interfaces (HCI).

In this chapter, various issues in tackling the problem of automating facial expression analysis have been discussed. All of them are intriguing and none has been solved in a general case. A number of conclusions have been reached:

- Most of the existing facial expression analysers assume that the input is a scale- and orientation-invariant, non-occluded portrait of the face.
- A number of surveyed analysers require manual labelling of the input images.
- Most of the existing analysers do not deal with missing or inaccurate facial data, which should be expected considering the state of the art in image processing.
- Most of the discussed methods do not detect displayed facial actions and none performs quantified facial action encoding applicable to automated FACS coding. As a result, none of the proposed facial expression analysers is an ideal automated tool for behavioural investigation of the face.
- Most of the proposed methods classify facial expressions under one of the basic emotion categories defined by Ekman and Friesen (1971). As a result, most of the discussed systems cannot classify/interpret arbitrary facial expressions.

The main goal in the development of the Integrated System for Facial Expression Recognition (ISFER) presented in this thesis was the enhancement of the state of the art in automated facial expression analysis. To wit, the aim was to develop a fully automated system for facial expression analysis that can be used for behavioural-science-research purposes and can (easily) be upgraded to form an integral part of an advanced multi-modal perceptive HCI. As a result, the research focused on a threefold:

1. Subject-independent, robust, fully automatic facial-expression-information extraction from a static (dual-view) facial image.
2. Robust, fully automatic facial expression recognition applicable to automated FACS coding; the reasoning should start by classifying facial-expression data into generic multiple facial-action categories and then adapt to a particular individual to quantify the encoded facial-action codes.
3. Automatic facial expression analysis in terms of multiple quantified interpretation labels learned from the current user.

These design requirements for the development and implementation of ISFER, emerged from the system's application domain. ISFER performs an automatic facial expression analysis from static facial images suitable for automatic measurement and for assessment of facial reactions in behavioural science. The system can be employed for facial expression analysis from either dual-view facial images acquired on-line or full-face images retrieved from an existing database of behavioural-science research material. When used on-line, the system deals with static dual-view facial images. The images are acquired using two digitised cameras mounted on the head of the observed subject. Two holders attached to a headphone-like device carry the cameras. The camera placed in front of the face at approximately 15 centimetres from the tip of the nose acquires the frontal view facial image. The camera placed on the right side of the face at approximately 15 centimetres from the centre of the right cheek obtains the profile view facial image. This camera setting ensures the presence of the face in the scene and the absence of rigid head movements. Hence, the images acquired during a single session by the utilised head-mounted camera device are scale and orientation invariant.

The actual implementation of the design requirements resulted in the system structure illustrated in Figure 2.25. ISFER consists of three major parts: a facial data extractor, facial action encoder and facial expression classifier. The theoretical background of the Artificial Intelligence techniques applied in the system is provided in chapter 3, while each part of the system is explained in a separate chapter (chapters 4 to 6). A detailed algorithmic representation of the processing of ISFER is provided in Appendix A. Validation studies on the prototype of the system suggest that the facial action encoding and the facial expression classification achieved by the system are consistent with those of human observers judging the same sample images. The validation studies are discussed in chapter 7. A set of guidelines for enhancing the system to form a part of an advanced multi-modal perceptive HCI is given in chapter 8.

The Facial Data Extractor is a framework for hybrid facial-feature detection, which for each prominent facial feature (eyebrows, eyes, nose, mouth, chin) applies multiple feature detectors to an input static facial image. The result of each detector, representing a spatial sampling of the contour of the relevant facial feature, is stored in a separate file.

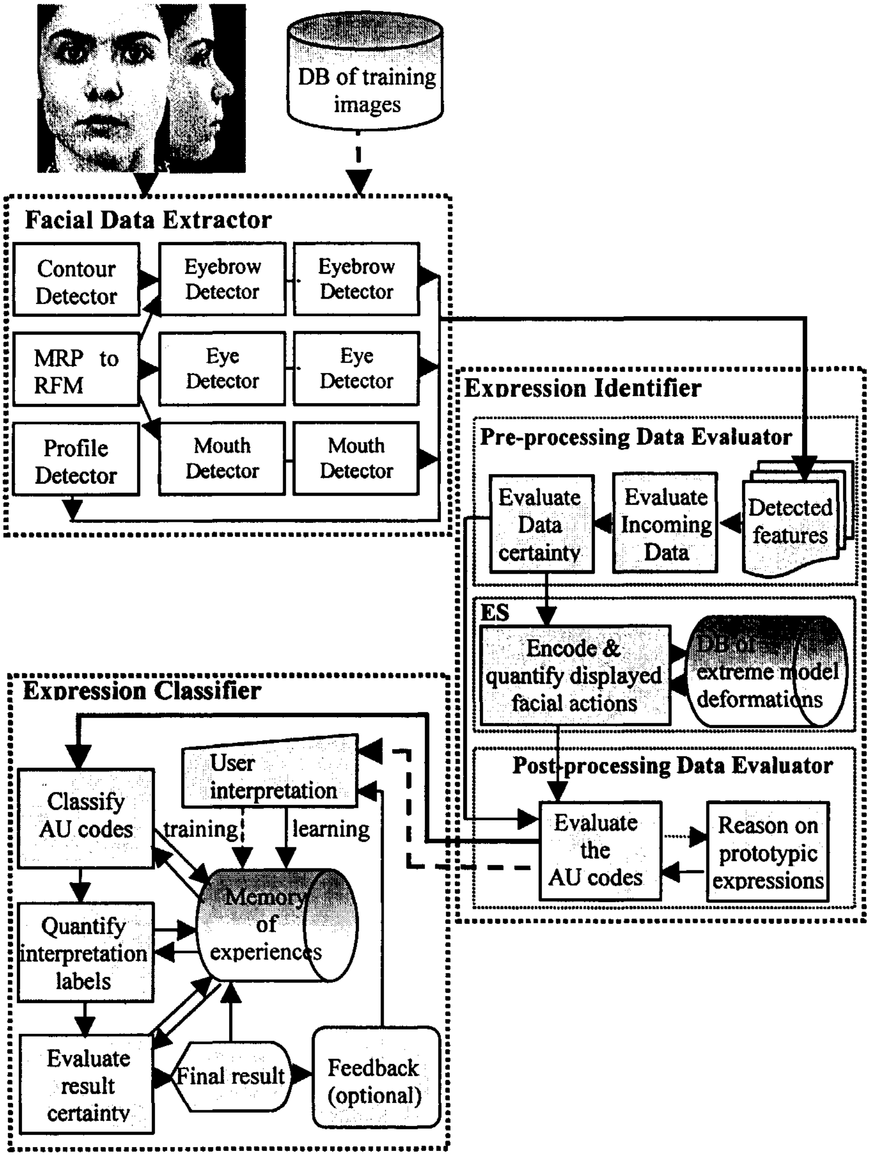


Figure 2.25: The structure of the Integrated System for Facial Expression Recognition (ISFER)

The Facial Action Encoder accepts the set of files resulting from the Facial Data Extractor and makes the best possible selection from redundantly detected facial features based on the evaluation of the certainty of the input data. The Facial Action Encoder performs quantified facial action encoding applicable to automated FACS coding and outputs a set of 32 quantified AU codes. The Facial Action Encoder has been implemented as a rule-based expert system that reasons with uncertainty while comparing the input facial-expression data with the data representing the expressionless face of the observed subject. In the first evaluation of the input data, the Facial Action Encoder deals with partial data by substituting the missing information with the pertinent data extracted from the expressionless face of the observed subject. As a result, the exact information about the examined facial expression is lost. In order to diminish this loss, a post-processor of the Facial Action Encoder is employed which (optionally) adjusts the AU-coded description of the examined expression based on a statistical prediction of the facial actions displayed.

The Facial Expression Classifier represents a memory-based expert system, founded upon the Schank's theory of human autobiographical memory organisation (Schank 1984) and instance-based learning. This part of the system expounds the encoded facial actions in terms of the interpretation labels supplied by the current user. The utilised memory of experiences (i.e. case base) is dynamic in the sense that new interpretation labels can be learned with experience.

ISFER cannot encode the full range of facial behaviour yet. From a total of 44 AUs defined in FACS, ISFER can encode 29 different AUs (i.e. 32 AU codes) from a static dual-view facial image. As a result, the system might assign an identical AU-coded description to expressions that are unlike in terms of underlying facial actions. Another system limitation is the device used for acquiring dual-view facial images; it is rather large and heavy. This is pretty inconvenient for the monitored subject. It is even more important (and hindering) that the subject should turn his/her head quite slowly for the device to remain in the same position relatively to the subject's face. Otherwise, the performed reasoning will be erroneous as the evaluation of the input data certainty done by the Facial Action Encoder is based on a comparison of the *immovable facial points* (e.g. the inner corners of the eyes, medial point of the mouth; see chapter 5) and the pertinent points extracted from an expressionless face of the observed subject. This comparison is feasible only if the images acquired during a single session are scale and orientation invariant. Those and other drawbacks of ISFER are discussed in detail in sections 4.4, 5.7, and 6.6, and then summarised in Table 8.3.

As the subsequent chapters of this thesis will point out, no system exists that performs an ideal automatic facial expression analysis from photographs or from video sequences, including ISFER presented in this thesis. Yet ISFER exhibits properties (Table 2.10) that are somewhat closer to those of an ideal facial expression analyser (summarised in Table 2.1) than those of the currently existing automated facial expression analysers, presented in the previous sections of this chapter (see Table 2.2 and Table 2.3).

Table 2.10
ISFER properties

	Characteristic			**Characteristic**	
1	Automatic image acquisition	yes	**10**	Automatic expression classification	yes
2a	Subjects of any age and outlook	no[2]	**11**	Distinguishes all pos. expressions	no
2b	Ethnicity (tested on)	8	**12**	Deals with unilateral facial changes	yes
3	Deals with variation in lighting	no[3]	**13**	Obeys anatomical rules	yes
4	Deals with partially occluded faces	no	**14**	# of different AUs (from 44 in total)	29
5	No special make-up required	yes	**15**	Quantifies facial-action codes	yes
6	Deals with rigid head motions	yes[4]	**16**	Unlimited # of interpretation categ.	yes
7	Automatic face detection	yes	**17**	Features adaptive learning facility	yes
8	Automatic facial-data extraction	yes	**18-19**	Quantified multiple classification	yes
9	Deals with inaccurate facial data	yes	**20**	Features real-time processing	no

[2] The subject may not have facial hair or wear glasses.

[3] The images are acquired under constant illumination.

[4] ISFER utilises a head-mounted camera device. Because of the constraint that the subject should move slowly, no rigid head movements are encountered.

3 Artificial Intelligence

Is the AI science or engineering, analytic or synthetic, empirical or theoretical? The answer is of course 'yes'.

(Davis 1998)

There is no strict definition of Artificial Intelligence (AI). Some definitions often referred to are the following:

- AI is a collection of techniques to handle knowledge in such way that new not explicitly programmed results can be obtained (Boullart 1992).
- AI is the science of making machines do things that require intelligence if done by men (Minsky 1986).
- AI research is the part of computer science that investigates symbolic, non algorithmic reasoning processes and representation of symbolic knowledge for use in machine intelligence (Feigenbaum 1977).

The issue here is not to select one of these definitions over another (although we all may have our individual reasons for doing so), but become aware of different conceptions that they represent and of fundamentally different assumptions they make. We may come to comprehend this diversity, one for which a complete and concise uniform description may not be possible, by learning *why* it arose. The answer lies in the vaguely defined concept of intelligence.

If AI is centrally concerned with intelligence, we ought to start by considering what kinds of behaviour characterise it. Four kinds of behaviour are commonly used to distinguish intelligent behaviour from instinct and stimulus-response behaviour, namely, prediction, adaptability, intentional action and reasoning. Yet even if we focus on just one of them – reasoning – it soon becomes clear AI provides a multitude of answers as to what we mean by reasoning. There are at least four typical notions of what constitutes intelligent reasoning:

1. Intelligent reasoning is some variety of deduction (mathematical logic view; the present-day examples of this view in AI are provided by logicists).
2. Intelligent reasoning is characteristic human behaviour (psychological view; this view has given rise to the AI area of knowledge-based systems).
3. The key to intelligent reasoning is the architecture of the human brain (biological view; this view set the basis of the AI area of artificial neural networks).
4. Intelligent reasoning obeys the axioms of probability theory (statistical view; this view initiated the AI area of reasoning with uncertainty and fuzzy logic).

Given that the notion of intelligence differs from perspective to perspective (from logical to psychological, individual to collaborative, etc.), we find that intelligence represents many things and that it is composed of many elements that have been thrown together over time. If AI is centrally concerned with intelligence (having all these facets), then we should probably define AI by a collection of definitions rather than by a single definition since a uniform definition of intelligence is not at hand.

Of course, it remains tempting to try to unify the existing definitions of AI. Perhaps Aaron Sloman (1994), who suggested that AI should be considered as the *exploration of the design space of intelligences*, has in fact done this. First, the plural – intelligences – emphasises the multiple possibilities of what might be meant by the notion of intelligent reasoning. Second, the term 'design space' suggests exploring broadly and deeply, scientifically and practically, analytically and synthetically. This is the view which allows (in a sense contradictorily) the characterisation of AI as science as well as engineering, as empirical as well as theoretical. Finally, this is the approach that encourages a currently growing body of research on applications, such as ISFER, which simultaneously make use of many different AI techniques.

A crucial decision to be taken in the development of an application based on the AI paradigm is the choice of AI technique(s) to be utilised, as the technique(s) determines all – from research methodology to be used to its chance of success. Section 3.1 is concerned with the issue of determining the appropriate paradigm for solving the problem of automating facial expression analysis in static facial images as defined by the design requirements for the development of ISFER (see section 2.6). The actual implementation of these design requirements resulted in a system consisting of three main parts (Figure 2.25): the Facial Data Extractor, the Facial Action Encoder, and the Facial Expression Classifier. The Facial Data Extractor is a hybrid approach to detecting the prominent facial features in static facial images and employs several detection schemes based on neural networking. The Facial Action Encoder is an expert system that reasons with uncertainty about the displayed facial actions and their intensity. The Facial Expression Classifier is a self-adaptive memory-based expert system that performs quantified classification of the encoded facial actions into multiple facial-expression-interpretation categories defined by the user. Finally, the interaction between different parts of the system can be viewed as a co-operation between supervisors (which may be fashionably called agents) of the lower-level processes. This makes ISFER a functionally distributed application as opposed to a spatially distributed application. Thus, ISFER simultaneously employs many different AI techniques and paradigms. This chapter presents these AI techniques. Section 3.2 is dedicated to expert systems (logical/psychological view). Section 3.3 gives an introduction to artificial neural networks (biological view). Reasoning with uncertainty (statistical/biological view) is explained in section 3.4. In this section, special attention is paid to which of the existing formalisms for handling uncertainty is most appropriate for ISFER. Section 3.5 discusses machine learning in general. Case-based reasoning (CBR), as a specific example of instance-based learning employed by the Facial Expression Classifier part of ISFER is explained in detail in section 3.6. An introduction to the distributed AI and agent-based systems (biological view) is given in section 3.7. Finally, section 3.8 discusses some general peculiarities of the AI application development process.

3.1 Assessment and scoping

Various factors must be considered before the actual development of an AI application can proceed. First of all, it is crucial to determine if the AI paradigm is an appropriate paradigm for solving the given problem. AI systems do not differ from other software and they will be successful only if there is a real demand for them. This means both that the intended AI application is the only possible solution (or at least a more efficient one) for the given problem and that future users are convinced of the need for such an application. The latter is especially important because AI applications are commonly developed through collaboration with future users, who are usually the experts providing the expertise to be built into the intended AI system. If the users/ experts are not convinced of the need for the system, they will not co-operate.

Deciding whether there is a real demand for an AI application is in fact estimating the potential for success of an AI system to be built. This assessment is the first step of an AI application's development process, known in the AI literature as the *study of feasibility* (Boullart 1992, Saborido 1992).

Study of feasibility

A number of issues determine whether the employment of the AI paradigm will result in an appropriate and successful solution to the imposed problem. The most important ones are:

1. *Can the problem be solved effectively by conventional programming?* If the answer is yes, then an AI application is most probably not the best choice. AI

systems are best suited for situations for which there is no efficient algorithmic solution.

2. *Is the domain well bounded?* It is very important to have well-defined limits with regard to what the intended AI system is expected to "know" and what its capabilities should be. If those limits are not explicitly defined then either some kind of general knowledge has to be acquired (general knowledge is difficult to acquire and this kind of knowledge has little or no realistic applications) or there is no well-defined point at which the development of the AI application ends.
3. *Is there a need and a desire for an AI application?* For instance, suppose that there are already many human experts in the field where the future AI system has to be employed. If the system merely emulates the expertise of the existing human experts and does not perform more effectively or more efficiently than they do, justifying the need for the intended system in such a domain is rather difficult. Also, if the experts/ users do not want the system, it will not be accepted even if there is a need for it.
4. *Is there at least one human expert who is willing to co-operate?* Not all experts are willing to have their knowledge examined and then "put" into a computer.
5. *Can the expert explain the knowledge so that it is understandable to the developers of the AI application?* Even if the expert is willing to co-operate, he/she might have difficulties in expressing the knowledge in explicit terms. Such difficulties are often bounded by so-called procedural knowledge. Procedural knowledge is a type of knowledge that involves an automatic response to a stimulus; for example try to explain in words the term "to ride a bicycle". Explicitly defining the portion of knowledge that is "obvious" to the expert (while the knowledge engineer knows nothing about it) forms another burden in the knowledge acquisition process. Acquisition techniques like structured interviews (for eliciting the "obvious" knowledge) and protocol analyses (for eliciting the procedural type of knowledge) could be utilised in tackling the problem. Also, recording the sessions with the expert might help in discovering gaps in the acquired knowledge.
6. *Is the problem-solving knowledge uncertain?* If the knowledge to be elicited is experience knowledge, then an AI system probably forms an appropriate paradigm since the expert's knowledge may be based on a trial-and-error approach rather than on the logic and algorithms. If logic and algorithms can solve the imposed problem, conventional techniques probably form the best paradigm to be applied.

The most extended technique for estimating the potential for success of an intended AI application is called a *checklist.* A checklist is organised in categories and criteria, where each category has several criteria. Each criterion represents a yes/no question and has a weight assigned to it according to its importance. For each category the weights of the yes answers are summed, representing the score of the category. This means that the maximal score of a category is the sum of weights of all criteria belonging to that category, which can be achieved if and only if the answer to each of the questions is affirmative. An observed future AI system has a fair potential for success if the sum of scores achieved per category is at least 50% of the maximum possible score. A commonly used checklist is the one proposed by T. Beckman (1991). This checklist is composed of six categories, each of which has its own distinct influence on the project's potential for success expressed by the maximal score that the given category can have. These categories are:

1. *Characteristics of the task to be performed.* The questions (criteria) in this category concern the complexity of the task, existence of an algorithmic solution, limitation of the knowledge domain, etc. The maximum score is 25 points. The future AI system must score at least 13 points in this category.
2. *Future system payoff.* The questions in this category concern the benefit/cost ratio. The maximum score is 20 points. For this category, the intended AI system must score at least 10 points.
3. *Customer management commitment.* This category can be crucial because even if the technical feasibility of a certain system is demonstrated and payoff is ensured, the project can still fail due to management problems or simply lack of interest. The maximum score is 20 points.
4. Knowledge engineer(s) skills and experience. The maximum score is 15 points.
5. *Domain expert(s) characteristics.* Expert(s) should be co-operative, communicative and available for an extended period of time. The design team must not only develop the system, but also make the experts feel partly in control of the project. If the experts feel that they have no say over the project, it is likely that they won't co-operate. The maximum score in this category is 10 points.
6. *User characteristics.* If the users feel threatened by the future AI system, or simply do not need it, or the interface is inadequate, they won't use it and the project will fail. The maximum score is 10 points.

The first two categories are essential for success and therefore the future system must score at least 50% for each one. The other four categories contribute to a lesser degree. Scores below 50% for any of them do not imply that the system should be discarded but indicate potential difficulties. In any case, the future system should have an overall score of at least 50% (i.e. 50 points).

Assessment and scoping in ISFER

ISFER is strongly application dependent (see also section 2.6). The main goal in its development is to achieve a fully automatic facial expression analysis which is applicable to automated facial action (FACS) coding and automated facial expression classification in observer-defined interpretation categories, so that it can be employed for behavioural science investigations of the face. Although FACS (Ekman and Friesen 1978; for a detailed explanation, the reader is referred to sections 2.1 and 5.1) represents the most prominent method for measuring facial expressions in behavioural science, a major impediment to its widespread employment is that its manual application is time consuming and that much time is required to train human experts to use it. Each minute of videotape takes approximately one hour to score and it takes 100 hours of training to achieve minimal competency on FACS. Automating FACS would not only make it widely accessible as a research tool, it would also accelerate the whole process of facial action coding and improve the precision and reliability of facial measurement. This, and the fact that automated expression classification into observer-defined interpretation categories would make it possible for behavioural scientists to define their own notions of various affective states such as stress, embarrassment, and pain, explains and justifies the need for an automated facial expression analyser like ISFER.

The behavioural-science application domain defines all the characteristics of the task that the system should carry out, the knowledge that has to be emulated by the system, and the environment (the deployment platform and human resources) in which the system is to be primarily used. To estimate the potential for success of ISFER, the checklist of T. Beckman (1991) was used and the following results were obtained:

1. Characteristics of the task to be performed: 22 from a maximum of 25 points scored (Table 3.1).
2. Future system payoff: 13 from a maximum of 20 points scored (Table 3.2).
3. Customer commitment: 11 from a maximum of 20 points scored. The ISFER project was not founded by any other organisation (potential customer) except by the Delft University of Technology where the system is also designed and developed. However, due to some preliminary discussions with the researchers of the behavioural science group of the Free University of Amsterdam, the Netherlands, this research group might be viewed as a potential customer. Yet further arrangements and agreements should be made before we can realistically expect that this research group will fully participate in a further development, testing and applying ISFER as a behavioural-science-research tool.
4. System designer skills: 15 points from a maximum of 15 points scored.
5. Domain expert characteristics: 7 points from a maximum of 10 points scored. The expertise on facial action encoding from facial images that has been built into ISFER has been acquired from the FACS manual in a straightforward manner (see chapter 5). Therefore no human expertise was necessary while developing the system. However, for validating and evaluating the performance of the system, human experts in FACS coding were necessary. The research staff of the Knowledge Based Systems Group at the Delft University of Technology, being involved in the ISFER and similar projects (and therefore motivated to achieve a rather high level of competency in FACS coding), facilitated completion of these tasks.
6. User characteristics: 8 points from a maximum of 10 points scored. Primary users of the system will be behavioural science researchers of the face. Since

there is a strong need for an automated facial expression analyser like ISFER in the field of behavioural science, it is likely that future user will be willing to use the system.

Table 3.1
Properties of the facial expression analysis task (see section 2.6) mapped on the criteria of T. Beckman checklist's first category

Criterion	Answer	Score
Task is primarily cognitive, requiring analysis, synthesis, decision making	yes	2
Task involves primarily symbolic knowledge and reasoning rather than numerical computation	yes	2
Task is complex	yes	2
Task involves chains of reasoning or multiple levels of knowledge	yes	1
Task requires judgement or reasoning about subjective factors	yes	2
Task cannot be solved using conventional computing methods	yes	1
Task involves incomplete or inaccurate data	yes	2
Task often requires explanation, justification of results, or reasoning	yes	2
Task requires classification rather than search	yes	1
Task knowledge is confined to a narrow domain	yes	1
Task knowledge is stable	yes	1
Incremental progress is possible; task can be subdivided	yes	1
Task does not require reasoning about time or space	yes/no	0
Task is not natural-language intensive	yes	1
Task requires little or no common sense or general-world knowledge	yes	1
Task does not require the system to learn from experience	no	0
The intended AI system is similar to an existing AI system	no	0
Data to be used as well as some case studies are available	yes	1
System performance can be accurately and easily measured	yes	1
	Σ	22

In summary, assessing the potential for success of ISFER according to T. Beckman's (1991) checklist resulted in an estimate of 76% from a maximum of 100% (the total score over all categories was 76 points from a maximum of 100 points). Altogether, this led to the following conclusions:

- ISFER is justified by real needs of the researchers of facial behaviour,
- ISFER has a clear scope restricted to automated facial expression analysis,
- the source of the knowledge emulated by ISFER is available,
- ISFER can be objectively validated and evaluated by domain human experts,
- ISFER has a high potential for success.

The next step in the development of an AI application is to select the AI techniques to be employed. It is usually wrong to make a definitive choice of techniques to be used at the beginning of the development process and before the problem has been analysed in detail. As obvious as this might seem, this rule is often not followed and often some AI technique is chosen prematurely only because it is a hot topic in AI or because the developer is familiar with it (Moulton 1998). The appropriate techniques to be used should be selected according to the properties of the task that the future system should carry out, while keeping in mind that the simpler the system, the easier to deploy it, to understand it, and to accept it.

Table 3.2
Properties of the ISFER in terms of system's payoff mapped on the criteria of T. Beckman checklist's second category

Criterion	Answer	Score
System would significantly increase revenues (less time spent on analysing a single photograph or on documenting categorisation of affective states)	yes	2
System would reduce costs	no	0
System would improve quality	yes	2
System would capture undocumented expertise that is in short supply	yes	2
System would distribute accessible expertise to novice users	no	0
System would require no or minimal more data entry than current system	yes	1
System would be developed using commercially available tools	yes	2
System maintenance would be low	yes	1
System would be executable on an affordable work station	yes	2
Partial completion of the system would still be useful	yes	1
System would result in benefit/cost ratio of at least 10:1	no	0
	Σ	13

3.2 Expert systems

Traditionally, tools and machines have been used by humans as passive mechanical artefacts that extended, enhanced and multiplied their physical and mental abilities – a hammer is stronger than a hand and by car one can travel faster than on foot. Over the last decades, computers have been used as sophisticated tools for enhancing human abilities such as memory and calculation. Research in artificial intelligence has been aimed at developing software to emulate the so-called intelligent capabilities of human beings such as reasoning, natural language communication and learning. With such programs, the computer's role departs from that of a mere tool, as it becomes a kind of assistant to humans. Knowledge-based systems, synonymously expert systems, form a sub-field of artificial intelligence which, for some three decades now, has investigated knowledge models and reasoning techniques that might assist a human decision maker.

Definition

Expert systems have been defined in various ways, but all the definitions share a general vein suggesting that expert systems are artificial means used to emulate the decision-making ability of a human expert (e.g. Barr and Feigenbaum 1981, Boullart 1992). Preferably the definition suggested by Jackson (1999) is used instead:

> *An expert system is a computer program that represents and reasons with knowledge of some specialist subject with a view to solving problems or giving advice.*

Yet, since the concepts like *expert*, *program*, *specialist* and *problem* are vaguely defined notions, it might be advantageous to characterise an expert system rather than to attempt to define it. There are five most important characteristics of an expert system that can be distinguished:

1. *An expert system emulates human reasoning.* This does not mean that the system is a faithful model of a human expert, but that it simulates the performance of the relevant expert's problem-solving process. It reasons using appropriate representations of human knowledge.
2. *An expert system should be capable to learn from its past experience.* Similarly to human experts, the system should be able to derive a proper solution faster when the same problem is presented more than once.
3. *An expert system applies heuristic or approximate reasoning.* It performs a task by applying rules of thumb to a symbolic representation of well-defined domain knowledge (heuristic), rather than by employing just algorithmic or statistical methods. In addition, the data and knowledge about the problem domain might be ambiguous. Similarly to human experts, an expert system should perform in uncertain environments by utilising a kind of intelligent guessing referred to as *uncertainty management* (Zadeh 1983, Kandel 1991).
4. *An expert system should be able to explain and justify obtained solutions.* As with human experts, the system's ability to explain and/or justify obtained solutions or recommendations concerns its ability to clarify its reasoning process and answers questions about the inference procedure. The user can relate to the inference process and verify that the expert system does what it is supposed to do. In addition, usually a wide range of (non-expert) users works with an expert system and its processing should therefore be rather transparent if it is to be understood by the users.
5. *An expert system must exhibit high performance in terms of speed and reliability.* In order to be a useful tool, an expert system must propose correct solutions in a reasonable time, at least as fast and correct as a human expert.

An expert system is always designed to be an "artificial" expert in a single problem domain. Similarly to a human expert, it reasons about the knowledge which is specific to that problem domain as opposed to common-sense knowledge. The point here is to have a *well-defined knowledge domain*, that is, to have well-defined limits to the problem the expert system is expected to solve. For instance, an expert system designed to diagnose infectious diseases should not be expected to give a recommendation for paediatrics cases or a weather forecast.

When the knowledge in the system originates from sources other than human experts, the more general term *knowledge-based system* should be used instead of the term expert system. However, nowadays the term expert system is often applied to any system which uses expert system technology.

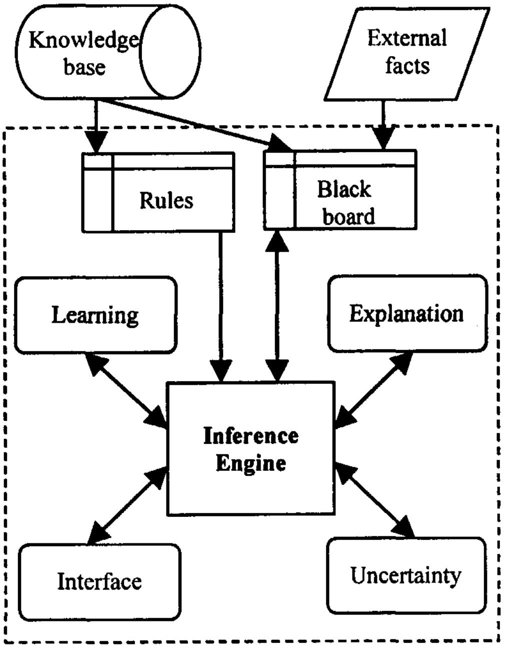


Figure 3.1: General structure of an expert system

Architecture

Expert systems have two main parts, namely, a knowledge base and an inference engine (Figure 3.1). The knowledge base contains knowledge about the problem domain, usually in the form of heuristic rules. The inference engine uses the rules to infer appropriate conclusions based on relevant portions of the knowledge base and a set of facts that form the current input to the system (stored on a so-called blackboard). While the knowledge base is always specific to the problem to be solved, the inference engine is usually generic – i.e. it could be reused in another expert system.

Yet, in order to implement the characteristics defined above, an expert system should also contain other components, shown in Figure 3.1. A learning facility, an explanation facility and an uncertainty management program should be implemented if the system is to exhibit the ability to learn, explain and reason in uncertain environments. Since the user is not necessarily familiar with the inner operations of the expert system, the expert system should be easy to use. Therefore, a user-friendly interface is a rather crucial integral part of a well-developed expert system.

Knowledge representation

In the realm of expert systems, additional data other than the raw data (facts) are usually employed as well. These data are referred to as *knowledge* and considered as a refinement of information. Knowledge can be incomplete or fuzzy and consists of collections of related facts, procedures, models, and heuristics that represent the problem-solving tactic of the human expert. Knowledge may be regarded as contextual information, organised in such a way that it can readily be applied to problem solving, perception and learning.

A knowledge base is a file which contains knowledge and facts about the domain of the problem. The first step in development of a knowledge base is referred to as *knowledge acquisition.* This is the process of acquiring knowledge from an expert (or experts), which is usually performed by the developers of the expert system. Knowledge varies widely in both content and form and it may be specific, general, exact, fuzzy, procedural or declarative. In addition, people usually have problems with articulating the ways they think and reason. Hence, interviews with an expert should be well structured so that all available knowledge about the relevant problem domain can be gathered.

Next, the acquired knowledge should be represented in the knowledge base in a form that will be not only efficient to retrieve and manipulate by the expert system but also amenable to the user. The user should be able to maintain and edit the knowledge base in a relatively straightforward manner. Over the years, numerous knowledge representation schemes have been proposed and implemented. Various classifications of knowledge representation schemes have been proposed as well. The categorisation presented here is proposed by Hughes (1991). It was chosen because it is more general than for instance the one proposed by Barr and Feigenbaum (1981) or the one of Giarratano and Riley (1990). Hughes' classification of the knowledge representation schemes is the following:

1. *Logical knowledge representation schemes* represent the knowledge base by expressions in formal logic. First-order predicate calculus (Kowalski 1979) and the programming language PROLOG are most commonly used.
2. *Network knowledge representation schemes* represent the acquired knowledge by a graph in which the nodes represent facts or concepts from the problem domain and the arcs represent relations between these facts. Semantics nets exemplify this prepositional declarative knowledge representation formalism.
3. *Structured knowledge representation schemes* form an enhancement of the network schemes (Findler 1979); the net nodes can be complex data structures consisting of named slots with attached values that may be either simple data values or pointers to other complex data structures or procedures for executing some particular task. Frames form a typical structured knowledge representation

scheme in which the knowledge is divided into a hierarchy of clusters. Due to the organisation in which lower hierarchical levels inherit the properties of higher hierarchical levels, frames resemble object-oriented structures which have a high expressive power.

4. *Procedural knowledge representation schemes* represent knowledge in the form of procedures or sets of instructions for solving a given problem. Production- or rule-based expert systems (Davis and King 1977) exemplify the procedural representation approach.

Since the Facial Action Encoder part of ISFER (Figure 2.25), has been designed and implemented as a rule-based expert system, the rule-based knowledge representation is discussed in detail. Production rules can be represented in a variety of ways which can be classified into three basic categories (Schneider et al. 1996): trees, bit matrices and relational lists. In the *tree representation* of a knowledge base, a directed graph is specified in which the nodes represent the rules and the links depict the relations between the conclusions and premises of the rules. It is impractical to utilise a tree representation if multiple conclusions are searched because the entire tree has to be searched in that case. A *bit matrix* is a ***NxN*** matrix where ***N*** is the number of rules in the knowledge base and each matrix element ***(i, j)*** is set either to 1, if the conclusion of rule ***i*** forms a part of the premise of rule ***j***, or otherwise to 0. A bit-matrix representation of the knowledge base requires quite some memory space for storage and the search of a bit matrix is quite slow. A *relational list (R-list)* is a list of 4-tuple elements, where the first two columns of each element indicate a conclusion clause which forms a premise clause of another rule, depicted by the next two columns. The R-list is a simple data structure that can easily be modified, facilitates a fast search algorithm and enables the inference process (explained below) to generate multiple conclusions.

Inference procedure

In a rule-based expert system, three different reasoning methods can be employed: forward chaining, backward chaining and direct chaining.

Forward chaining is used when the goal is not specified. The underlying concept entails the verification of the premise of a rule in order to verify that the conclusion of that rule is true. The cycle of forward-chaining inference procedure is schematically illustrated in Figure 3.2. The procedure begins with instantiation and matching of the rule premises to the facts. Each rule for which the premise part is true is placed on the agenda. The inference engine chooses the first rule on the agenda to fire (i.e. to execute the consequent part of that rule). Due to execution of the rule, new facts can be added, altered or removed from the blackboard. The process is repeated either until the solution is reached or until no new facts are present on the blackboard.

There are two types of forward chaining, namely, autonomous forward chaining and interactive forward chaining. In an autonomous forward reasoning expert system the initial data and the knowledge base are provided. The task of the inference procedure is to generate the proper decision tree, to match the data with the rules of the knowledge base, to fire the rules, and to reach the conclusion. In an interactive forward reasoning expert system the decision tree is assumed to be available. The inference engine uses the depth-first search, which guarantees that if there is a solution, that solution will be found by visiting each node along the path and asking the user to provide the data necessary to continue the search of the tree.

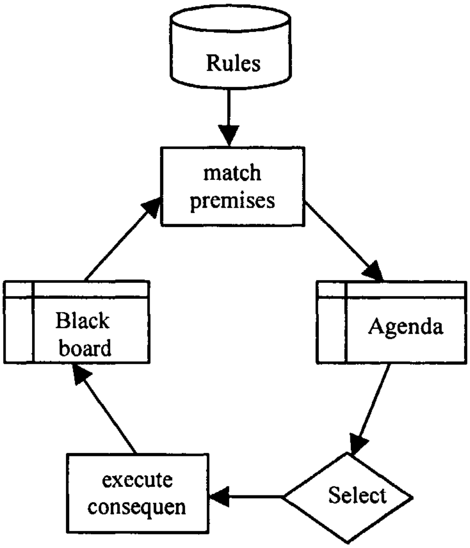


Figure 3.2: The cycle of forward chaining inference procedure

Backward chaining suggests a process of recursive deductions in which a goal (hypothesis) is defined first. Then the steps necessary to validate the goal are defined (by creating a path to the top of the decision tree) and each step along the path is validated until the goal (equal to the premise of a certain rule) is reached. Backward chaining is usually used in expert systems where the goal is known and the expert system's role is to find evidence to support the goal.

Two types of the *direct chaining* inference procedure can be distinguished: simple direct chaining and fast direct chaining. Simple direct chaining is a process of firing the rules of the knowledge base repeatedly for as long as there are rules that can be fired. A recursive process always starts with the first rule of the knowledge base and ends with trying to fire the last rule of the knowledge base, where only already fired rules are skipped. Thus, the process is directed by the order of the rules in the knowledge base.

Fast direct-chaining inference procedure is a breadth-first search algorithm that takes advantage of the R-list representation of the knowledge. A recursive process starts with the first rule of the knowledge base and then searches the R-list for a linkage between the fired rule and the rule that the process will try to fire in the next loop. If there is a rule whose premise is equal to the conclusion of the rule just fired, the process will first try to fire that particular rule in the next loop. Consequently, fast direct-chaining inference is more efficient than any inference process described above because it revisits only the rules that may potentially contribute to the reasoning process.

ISFER and other applications

Expert systems are constructed to mimic the reasoning process of human experts, to capture and preserve their knowledge and to make this knowledge more accessible. This technology has been successfully applied to a diverse range of domains such as:

- Interpretation of data, e.g. DENDRAL was designed to interpret data from mass spectrometers and to determine the structure of molecules (Feigenbaum et al. 1971), HEARSAY is a speech-understanding expert system (Erman et al. 1988), and the HERCULES expert system (Pantic et al. 1998b) was designed to interpret facial-expression data in terms of six basic emotions (Ekman and Friesen 1975).
- Diagnosis of malfunctions, e.g. MYCIN was developed to help physicians diagnose meningitis and blood infections (Shortliffe 1976) and JET X was designed to diagnose faults in military aircraft engines (Shah 1988).
- Prediction, e.g. the PROSPECTOR expert system predicts molybdenum mineral deposits after analysis of geological data (Duda et al. 1979).
- Consulting, e.g. MYCIN was used for diagnosis and therapy recommendations for infectious diseases (Shortliffe 1976), EXPLAIN was developed to assist non-experts in using a package of image processing algorithms (Tanaka and Sueda 1988) and LIA was designed to consult students about their study plans (Pantic et al 1998a).
- Configuration of complex objects, e.g. ASDEP is an expert system for power-plant design (Matin and Oxman 1988).

Generally, expert systems are useful if the built-in knowledge is well formalised, circumscribed, established, and stable, the problem domain is well-bounded and no common-sense knowledge is required, and there are acknowledged experts willing to co-operate or there is a body of well-structured detailed literature on the topic. In addition, the knowledge base of an expert system must be well structured and represented such that it facilitates easy testing and maintenance as well as high efficiency and reliability.

Generally, due to the properties of the facial-action-encoding task (section 3.1), an automated facial expression analyser can be developed as an AI application. In particular, due to the rule-based character of FACS and the overall characteristics of the task (i.e. it is a cognitive task that involves reasoning rather than numerical computation on a stable and narrow knowledge domain defined by FACS, see Table 3.1), a rule-based expert system seems to be an appropriate technique to be exploited for the development of the Facial Action Encoder part of ISFER. These matters are discussed in detail in chapter 5.

Expert systems are usually built using languages such as Prolog, Lisp and Clips (Giarratano and Riley 1990), expert system tools such as FEST (Schneider et al. 1996) or expert system shells such as S.1 and M.4 (Jackson 1999). Of course, other programming languages and development tools can be used instead, as long the intended functionality of the expert system can be achieved.

3.3 Artificial neural networks

The area of AI that deals with parallel, distributed, adaptive information processing systems that develop information processing capabilities in response to exposure to an information environment is called *neurocomputing* (Hecht-Nielsen 1990). The primary information processing structures of interest in neurocomputing are artificial neural networks (ANNs), although other classes of adaptive information processing structures are sometimes also considered (e.g. learning automata, associative memories, data-adaptive content addressable memories, simulated annealing system).

ANNs realise the connectionist paradigm of representing and processing information. The driving force behind this paradigm is the idea that information-processing systems can be built similar to the ones found in biological organisms. In rough analogy, ANNs are built out of a densely interconnected set of simple units (called *neurons*), where the connections between those units are bound with coefficients (called *weights*). The connection weights are the "memory" of the system, which can be adjusted such that the network "learns" some desired behaviour. ANNs are also often referred to as sub-symbolic information representation models since no interpretation can be usually given to individual nodes or connections (Kasabov 1996). In addition, processing of an ANN is considered as a "black-box" procedure since the "rules" for solving a given problem are not directly extractable from a trained ANN (for a survey of techniques for extracting rules from a trained ANN the reader can consult Andrews et al. (1995)).

ANNs provide a general practical problem-solving method based on learning of the desired input-output function from examples. ANN learning is robust to errors in the training data and has been successfully applied to a wide range of problems. This section presents the basic principles of artificial neural networks, but it is far from a detailed and exact presentation of all existing neural network architectures.

Definition

An artificial neural network is a parallel distributed information processing structure that can be viewed as a directed graph having the following properties (Hecht-Nielsen 1990):

- The nodes of the graph are called *processing elements* (or artificial neurons).

- The links of the graph are called *connections.*
- Each processing element can receive any number of incoming (input) connections.
- Each processing element has a single output connection that can branch (fan out) into identical copies to form multiple output connections.
- Processing elements can have *local memory.*
- Each processing element possesses a *transfer function* which can use local memory and/or input signals to produce the processing element's output signal. Transfer functions usually have a sub-function, called a *learning law,* which is responsible for adapting the input-output behaviour of the transfer function (over a period of time) in response to the input signals that impinge on the processing element. This adaptation is usually accomplished by modification of the values of variables stored in the local memory of the processing element.

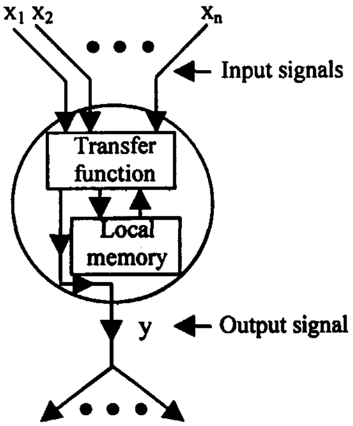


Figure 3.3: Architecture of a generic processing element

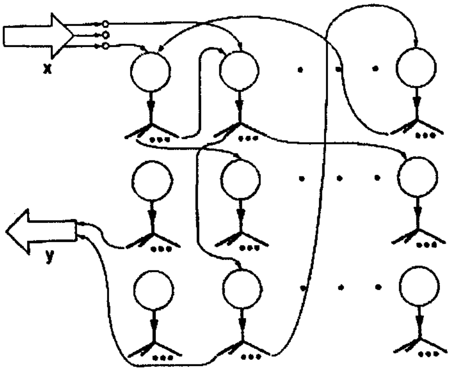


Figure 3.4: Architecture of a generic artificial neural network

Figure 3.3 shows the architecture of a neural network processing element. The transfer function receives as input the signals arriving via the incoming connections (those connections may originate from other neurons of the network or from the outside world) as well as values from local memory. Given these inputs, the transfer function outputs both the values to be stored in specified locations in local memory and the output signal of the processing element which may fan out and either form the input signals to other neurons of the network or form the output from the network to the outside world. This is schematically presented in Figure 3.4. The figure also illustrates a typical division of the network's processing elements into disjoint subsets, called *layers*. Any neural network can be configured as a collection of layers, in which all processing elements possess the same transfer function and are updated together. The input to the network can be viewed as a data array x, the output of the network is a data array y and the network can be thought of as a function $y(x)$. This observation is the basis of the mechanism used to embed neural networks into programmed computing systems.

An ANN as a computational model can be further characterised by four parameters (Kasabov 1996): type of neurons, connectionist architecture (the organisation of the connections between neurons), learning algorithm, and recall algorithm.

Artificial neurons

McCulloch and Pitts (1943) proposed the first mathematical model of a neural network processing element, i.e. an artificial neuron. It was a binary device using binary inputs and a binary output. In general, a functional model of an artificial neuron is based on the following parameters, which describe a neuron (see Figure 3.5): input values, input function, activation function, and output function. According to the type of values each of these parameters can take, different types of neurons can be identified.

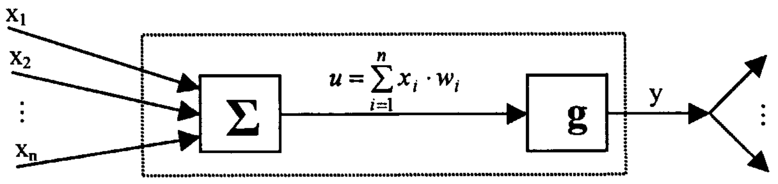


Figure 3.5: A general functional form of an artificial neuron

The input values $x_1, x_2, \ldots, x_n$ and the output value y of a neuron can be: binary {0,1}, bivalent {-1,1}, continuous [0,1], or discrete numbers in a defined interval. One of the inputs to a neuron, called *bias*, causes the transfer function (i.e. the input, activation, and output function) of the neuron to operate on the current input values and local memory values, to produce the output signal, and to (eventually) modify local memory values. Bias has a constant value of 1 and is usually represented as a separate input, say x_0, but for simplicity it is treated here just as another input clamped to a constant value.

The input function f of a neuron calculates the aggregated net input signal to the neuron $u = f(x, w)$, where x is the input vector and w is so-called *weight* vector; each component w_i of the weight vector w is a local memory variable associated with the corresponding input x_i. A typical example of input function is the summation function $u = \sum_{i=1}^{n} x_i w_i$.

The activation function s of a neuron calculates the activation level of the neuron $a = s(u)$. Four types of activation functions are most commonly used:

1. *The hard-limited threshold function*: if the net input signal to the neuron u is above a certain threshold T, the neuron becomes active, say $a = 1$.
$$a = s(u) = \begin{cases} 1, & u > T \\ 0, & u \leq T \end{cases}$$
2. *The linear threshold function*: the activation value a increases linearly with the increase of the net input signal to the neuron u, starting from a certain threshold T_1, but after a certain threshold T_2 is reached, the output becomes saturated (say to a value 1). There are different variants of this function, depending on the range of the neuronal output values.
$$a = s(u) = \begin{cases} 1, & u > T_2 \\ 1 - \dfrac{u - T_2}{T_1 - T_2}, & T_1 < u \leq T_2 \\ 0, & u \leq T_1 \end{cases}$$
3. *The Sigmoid function (S-function)*: any S-shaped non-linear transformation function which is bounded (e.g. to the interval [0,1] or [-1,1]), monotonically increasing, continuous, and smooth. Different types of sigmoid functions have been employed in practice, but the most commonly used is the logistic function $a = s(u) = 1/(1 + e^{-cu})$, where c is a constant.
4. *Gaussian function*: $a = s(u) = e^{-u^2/2}$.

The output value of a neuron can be represented by a single static potential or by a pulse.

In addition to the types of neurons described here, many other types have been developed. Examples are the RAM-based neuron (Aleksander 1989), fuzzy neuron (Yamakawa 1990), oscillatory neuron, chaotic neuron, wavelet neuron, etc. Descriptions can be found in Kasabov (1996) and Hecht-Nielsen (1990).

Connectionist architecture

Not only the type of artificial neuron used in an ANN should be described; the type of connections between the neurons in the ANN should be defined as well (i.e. the topology of the ANN should be determined too). Neurons in an ANN can be either *fully connected*, that is, each neuron is connected to each other neuron, or *partially connected* (this may mean that only connections between neurons in different layers are allowed, but in general it means that not all possible connections between all neurons of the ANN are present). In fact, we should be able to tell where a certain connection originates and terminates; a wiring diagram should be defined.

One way to construct a wiring diagram, which will represent topology of an ANN, is to number the neurons of the network from 1 to N and to delimit an interconnection matrix $\boldsymbol{M} = [\boldsymbol{m}_{ij}]$. Here, $\boldsymbol{m}_{ij} = 1$ if there is a connection going to the neuron $\boldsymbol{i}$ from the neuron j, and $\boldsymbol{m}_{ij} = 0$ if there is no such connection. This is a universally applicable approach to defining connections. Since it is difficult to construct an interconnection matrix for large networks, a geometric approach to defining connections is often used instead. This approach is based on the observation that the connections are made up of disjoint bundles of fibres, so-called fascicles, going from one geometrical region of neurons to another (Hecht-Nielsen 1990).

Two major connectionist architectures can be distinguished, according to the number of input and output sets of neurons and the layers of neurons used:

1. *Autoassociative* architecture, in which input neurons also function as output neurons.
2. *Heteroassociative* architecture, which has separate sets of input and output neurons.

According to the absence or presence of feedback connections in an ANN, two other types of connectionist architectures can be distinguished:

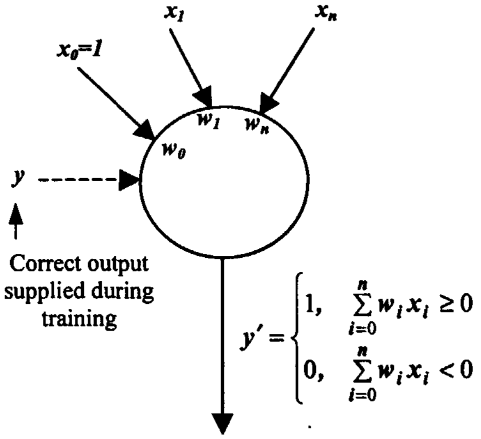
1. *Feedforward* architecture, which has no connections from output to input neurons and the ANN does not keep a record of its previous output values and the activation states of its neurons.
2. *Feedback* architecture, which contains connections from the output to the input neurons. The ANN keeps a memory of its previous states, and the next state depends not only on the input signals but on the previous states of the network as well.

Learning

The most attractive characteristic of ANNs is their ability to learn. Learning enables modification of behaviour in response to the environment. An ANN is *trained*, so that the application of a training set X of input vectors produces the desired set of output vectors Y, or the ANN learns about internal characteristics and structures of data from the set X. The training process is reflected in the change of the weights bounded to the connections between the neurons of the network. During training, the weights should gradually converge to values such that each input vector x from the training data set causes a desired output vector y produced by the network. Learning occurs if after a training example has been supplied, a change takes place in at least one synaptic weight.

The learning ability of an ANN is achieved through applying a learning (training) algorithm. Based on the way a network is trained, the ANNs can be classified into two major groups:

1. supervised trained networks and
2. unsupervised trained networks.



$$y' = \begin{cases} 1, & \sum_{i=0}^{n} w_i x_i \geq 0 \\ 0, & \sum_{i=0}^{n} w_i x_i < 0 \end{cases}$$

Figure 3.6: Perceptron

In supervised training, input-output pairs are presented to the ANN, which has to learn to associate each input vector *x* to its corresponding and desired output vector *y*. In fact the network learns to approximate a function *y = f(x)* represented by a set of training examples *(x, y)*. Typical supervised trained networks are Perceptron, Multiple-Layer Perceptron (MLP) and the Hopfield network.

The perceptron is a simplest type of ANN introduced by Rosenblatt (1958). It consists of one or more processing elements shown in Figure 3.6 (each of which is also referred to as perceptron). The perceptron has an input consisting of an *(n+1)*-dimensional vector *x*, where x_0 is a bias input permanently set to 1. It utilises a summation input function and a hard-limited threshold activation function, where the threshold *T* = 0. The correct values of the connection weights are found using the delta learning rule (Widrow and Hoff 1960):

$$w^{new} = w^{old} + (y - y')x,$$

where *w* is the weight vector, *y* is the desired output vector, *y'* is the actual output vector of the perceptron and *x* is the input vector. The perceptron can be used to recognise patterns on the input matrix (e.g. a digitised image), but it cannot solve non-linearly separable problems.

To overcome this limitation of the perceptrons, Multiple-Layer Perceptrons were introduced. An MLP consists of an input layer, at least one intermediate (*hidden*) layer and one output layer. The neurons from each layer are fully connected to the neurons from the next layer (in some particular applications this does not have to be the case; the neurons might be partially connected). The neurons in an MLP usually have continuous value inputs and outputs, a summation input function and a non-linear activation function. The simple delta rule cannot be used to train an MLP because the errors of the hidden layers are not known. The correct values of the connection weights are usually found using a so-called generalised delta learning rule, also called gradient descent learning rule or error back-propagation learning algorithm (Rumelhart et al. 1986):

$$w^{new} = w^{old} - \alpha\nabla_w E(w),$$

where w is the weight vector, $\alpha > 0$ is a small constant called the learning rate and E is the difference (i.e. error) between the desired output vector y and the actual output vector y' represented as a surface in the weight vector space. In principle, the back-propagation learning algorithm requires the use of a differentiable activation function so that the error can be back-propagated using the chain rule for differentiation. This algorithm is simple in implementation, but requires many training loops. Various other back-propagation-network learning laws have been developed. The general goal is to provide a faster descent to the bottom of the error surface. Silva and Almeda (1990) gave an overview of the investigations on the topic. The MLPs are often used for classification problems because they can learn complex decision surfaces.

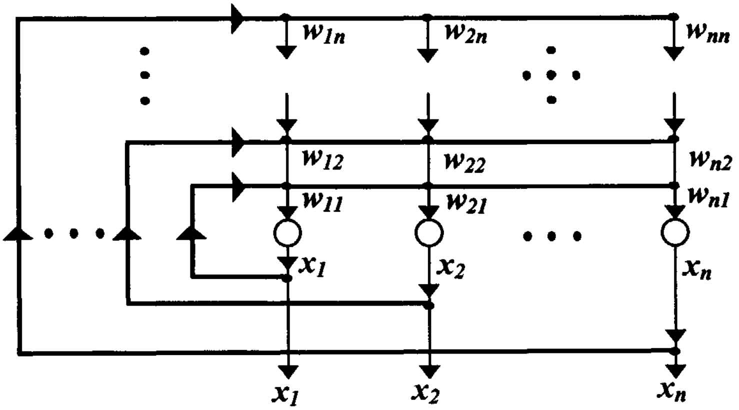


Figure 3.7: Hopfield neural network

A Hopfield network (Hopfield 1982) is a recurrent, fully connected, autoassociative network shown in Figure 3.7. Recurrent associative networks have something that other types of networks possess only to a limited extent or not at all: accrete behaviour. Typically, recurrent associative ANNs start at some initial state and then converge to one of a finite number of stable states. The neurons in a Hopfield network are characterised by a binary or bivalent input signal, binary or bivalent output signals that are wrapped around to become inputs to the network, a simple summation function and a hard-limited threshold activation function. Given an input vector x, the activation function of a neuron of a Hopfield network is:

$$x_i^{new} = \begin{cases} 1, & \sum_{j=1}^{n} w_{ij} x_j^{old} > T_i \\ x_i^{old}, & \sum_{j=1}^{n} w_{ij} x_j^{old} = T_i \\ -1, & \sum_{j=1}^{n} w_{ij} x_j^{old} < T_i \end{cases}$$

where $W = (w_{ij})$ is a symmetrical $n \times n$ weight matrix ($w_{ij} = w_{ji}$ and $w_{ii} = 0$) and T_i is the threshold defined for the i^{th} neuron. The response of a Hopfield network is dynamic: after a new input pattern has been supplied, the network calculates the outputs and feeds them back to the neurons recursively until equilibrium is reached. Equilibrium is considered to be the state of the system where the output signals do not change for two consecutive cycles, or change within a small constant.

A Hopfield network does not have a learning law associated with its transfer function; an energy function H is associated with it instead.

$$H(x) = -\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} x_i x_j + 2\sum_{i=1}^{n} T_i x_i \text{ and } \Delta H = 2*\left(x_k^{old} - x_k^{new}\right)*\left[\sum_{j=1}^{n} w_{kj} x_j^{old} - T_k\right]$$

where x is the input vector and ΔH is the change in the energy if just the state of the neuron $k \in \{1,n\}$, $j \neq k$ has changed. Whenever a neuron changes state, the energy function decreases, i.e. ΔH is negative. Starting at some initial position, the state vector of the system simply moves downhill on the energy surface of the network until it reaches a local minimum. In addition, independently of the initial state, the Hopfield network always converges to a stable state in a finite number of neuron-update steps (Hecht-Nielsen 1990). A major disadvantage of Hopfield networks is that they can rest in a local minimum state instead of a global minimum state (i.e. equilibrium). In order to overcome the local minima problem, Hinton et al. (1984) proposed a Boltzmann machine which is a discrete-time Hopfield network in which the neuron transfer function is modified such that it utilises the simulated annealing procedure.

Other supervised trained networks are the Radial Basis Function Network (RBFN), Bidirectional Associative Memory (BAM), Hamming net, MAXNET, etc. For a detailed description, readers are referred to (Zurada 1992, Kasabov 1996).

Unsupervised learning is a human ability some ANNs possess. Humans usually learn more by experience than by attending organised lectures. Similarly, ANNs based on an unsupervised training are merely given input examples, as they themselves can discover patterns and/or clusters in the presented data. A typical unsupervised trained ANN is a Kohonen network.

Figure 3.8 illustrates the basic structure of a Kohonen network (also referred to as the Kohonen layer). It consists of N neurons, each of which receives n inputs from a layer of fan-out units below. Each x_j, $j \in \{1,n\}$ input to a Kohonen processing element i has a real value weight w_{ij} assigned to it. Each processing element i calculates its input intensity $I_i = D(w_i, x)$, where $w_i = (w_{i1}, \ldots, w_{in})^T$, $x = (x_1, \ldots, x_n)^T$, and D(***u***, ***v***) is a distance measurement function. Two common choices for D(***u***, ***v***) are the Euclidean distance and the spherical arc distance. After calculating their input intensities, the Kohonen neurons "compete" to see which one has the smallest input intensity, i.e. to find out whose weight vector w_i is closest to x. Ties are resolved based on the lowest processing-element-index number. This competition can be implemented in various ways. One way is to facilitate each Kohonen neuron to compare its I_i value to those received from other processing elements to find out whose value is smaller. Once the winning Kohonen processing element is determined, its output is set to $y_i = 1$ while the outputs of other neurons emit $y_i = 0$. At this point the Kohonen learning takes place:

$$w_i^{new} = w_i^{old} + \alpha\left(x - w_i^{old}\right)y_i,$$

where $0 < \alpha \leq 1$ is a constant. The Kohonen learning law allows only the winning neuron to modify its weight and it moves the pertinent weight vector a fraction α of the way along the straight line from the old weight vector to the x vector. As new x vectors are entered into the network, the neurons' weight vectors are "drawn" to them and form a cloud near where the x vectors actually appear. A Kohonen network can be used for some statistical problems (e.g. for finding k-means), but in general it does not perform well, except in the case of linearly separable clusters of training data.

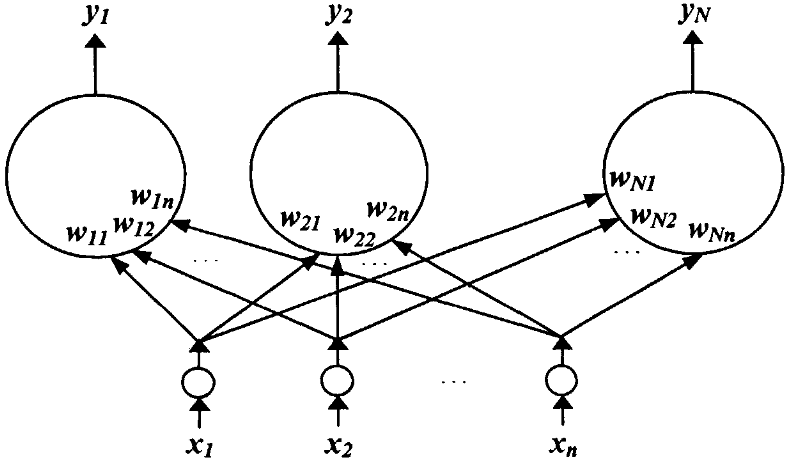


Figure 3.8: Kohonen network

Other, more complex unsupervised trained ANNs are the Cluster Discovery Network based on the adaptive resonance theory (ART) developed by Carpenter and Grossberg (1987) and Self Organising Feature Maps (SOM) developed by Kohonen (1982, 1990).

Generalisation

The recall process in the human brain is characterised by *generalisation,* that is, similar stimuli recall similar patterns of activity. In fact, the brain reacts to an unseen stimulus according to previously learned patterns. Similarly, the generalisation ability also characterises artificial neural networks. When a new input vector x' is presented to a trained ANN, a *recall procedure* is activated. The network would produce an output y' which is similar to the output y_i from the training examples, if x' is similar to the input vector x_i. In other words, the generalisation principle is that similar stimuli cause similar reactions. For a new input vector x', this principle is illustrated in Figure 3.9 as a mapping of the domain space to the solution space.

In the case of a recurrent ANN, generalisation can take several iterations of calculating consecutive states of the network. Eventually, the network goes into a state of equilibrium, when the network does not change its state during the next iterations. The example is the processing of a Hopfield network.

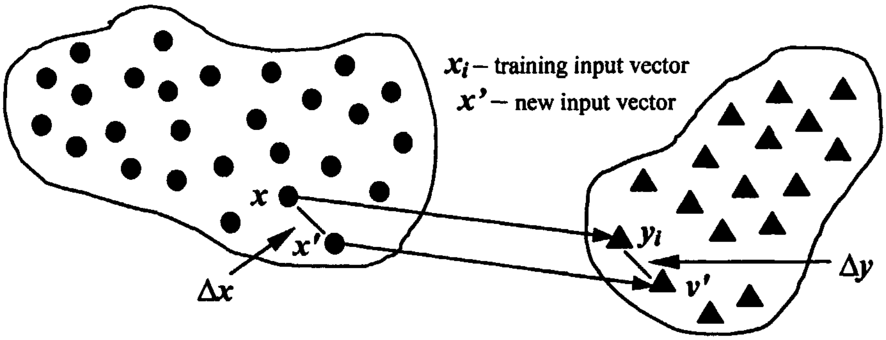


Figure 3.9: The generalisation principle of ANN

Applications

By far the largest area of activity in neurocomputing is that of applications (for an extensive review of the topic, the reader can consult (Rumelhart et al. 1994)). Numerous technologists, business analysts, scientists and mathematicians are now applying neuro-computing to a wide variety of problems. In this broad area of applications, ANNs have been commonly recognised as being well suited for solving problems in three domains:

- Sensor processing, in which the training data correspond to noisy, complex sensor data, such as inputs from cameras and microphones. For instance, two different facial feature detection schemes are integrated into the Facial Data Extractor part of ISFER which are based on a neural network processing of the facial image data (see section 4.3).

- Control or decision learning tasks for which more symbolic representations are used.
- Data analysis, in which the goal is either the development of a predictive model or a summary representation of a large body of data.

3.4 Reasoning with uncertainty

One of the characteristics of human reasoning is the ability to form useful judgements from uncertain and incomplete evidence. This ability is not only needed for everyday activities which people would normally never formalise, but also for tasks such as the medical diagnosis or securities analysis which have been subjected to formal treatment. Because the general need to form judgements from uncertain and incomplete data is so widespread, many techniques have been developed to aid or supplant people in this task. This section introduces a number of basic ideas about uncertainty measurement and inexact reasoning. It should be stressed that here the emphasis lies on methods for managing the uncertainty in expert systems in general as well as on the reason why in the case of ISFER it was necessary to experiment with a number of different formalisms. Hence, this section does not deal with the topic of uncertainty in a global manner, or in great depth. For such a prevalent discussion, the reader can consult the relevant literature (e.g. Shachter et al. 1990, Shafer and Pearl 1990, and Jackson 1999).

Sources of uncertainty

In general there are many different sources of uncertainty in knowledge-based problem solving, but most of them can be attributed to either imperfect domain knowledge or imperfect case data.

In the first case the theoretical basis of the domain may be vague and incomplete. Such domain theories typically use concepts which are not precisely defined or deal with phenomena that are imperfectly understood. The knowledge built in the reasoning mechanism of the Facial Action Encoder part of ISFER is in its entirety acquired from the FACS, which for each facial action (AU code) provides an unambiguous linguistic description of the facial change caused by the activation of that facial action. Hence, the knowledge domain of the Facial Action Encoder is narrow and well defined as opposed to vague and incomplete.

In the second case, data we deal with may be imprecise, missing, in conflict, or simply unreliable. We shall refer to data as *partial* when some answers to relevant questions about the data are not available. In the scope of ISFER, this might be the case when none of the feature detectors integrated into the Facial Data Extractor (chapter 4) succeeds to spatially sample the contour of a prominent facial feature. We shall refer to data as *redundant* when there are multiple answers to a single relevant question. In the scope of ISFER, this might be the case when several of the integrated facial feature detectors successfully perform spatial sampling of the contour of a single prominent facial feature. We shall refer to data as *approximate* when answers to relevant questions are available but these are of variable precision. The facial feature detectors integrated into ISFER employ image processing techniques as various as neural networking, template matching and fuzzy reasoning. Each of these has a different performance when applied for localising the contour of a certain facial feature and, consequently, data that should be dealt with are of variable precision.

As already mentioned, the main goal for the development of ISFER is to build an automated tool which can facilitate automated facial expression analysis in static facial images and be used for behavioural-science investigations of the face. Facilitating an automated facial action (FACS) coding from static facial images implies modelling the changes in facial expression in general, estimating the current deformations of the model based on the information extracted from the currently examined image, and mapping the estimated model deformations on the appropriate set of AU codes. In ISFER, a point-based dual-view face model (see Figure 5.2 for a somewhat adapted model illustrated in Figure 2.16) models the changes in facial expression. There are several motivations for choosing such a model (section 5.3), but the crucial one is the observation reported by Bassili (1978) and Bruce (1986). They showed that a point-based face model resembles the model used by human observers judging a displayed facial expression. Hence, a point-based face model facilitates a straightforward conversion of the rules for expression classification used by human observers (e.g. the FACS rules) into the rules of an automatic facial expression analyser. Further, the points of the deployed face model can be extracted in a straightforward manner from the contours of the facial features detected in the currently examined image by the Facial Data Extractor part of ISFER (Tables 5.4 and 5.6). In turn, the current deformations of the model can also be extracted in a straightforward manner (section 5.5). Roughly speaking, the Facial Action Encoder part of ISFER encodes the displayed facial actions and their intensities in two steps:

1. it estimates the current displacement of the model points from the relevant model points extracted in an expressionless face of the observed subject, and
2. it performs a rule-based mapping of the estimated model-points displacements and their intensity onto an appropriate set of quantified AU codes; this mapping is not bijective in the sense that a set of model-points displacements may be used for the encoding of a single AU code and a single model-point displacement may be used for the encoding of multiple AU codes.

Thus, the Facial Action Encoder reasons about the encountered facial actions and their intensity based upon the data which are extracted from the currently examined static facial image by the Facial Data Extractor and are approximate, most likely redundant and occasionally partial. In turn, despite of a well-defined knowledge domain of the FACS, employing the exact reasoning methods in the Facial Action

Encoder cannot lead towards some useful results, that is, towards the conclusions having variable certainty measures corresponding to the uncertainties embedded within the input data. To achieve this, some inexact reasoning methods should be employed.

There is broad agreement among artificial intelligence researchers that inexact methods are important in expert systems applications, but there is less agreement concerning what form these methods should take (Jackson 1999). Roughly speaking there are three principal formalisms for handling the uncertainty, namely probability theory, belief functions (also called the Dempster-Shafer theory of evidence), and fuzzy logic. The following subsections explore both the formalisms themselves as well as their suitability for the estimation and propagation of data certainty on various reasoning levels of ISFER. Given the kind of data uncertainty that may occur in the input to the Facial Action Encoder part of ISFER, a rough comparison of the three formalisms is summarised in Table 3.3.

Table 3.3
Applicability of three formalisms for estimating and propagating data certainty within ISFER

	Probability theory	Belief functions	Fuzzy logic
partial data	no/yes	no	no
redundant data	no/yes	no	no
approximate data	no/yes	no	no/yes
data dependency	no	no	yes

Probabilistic approach

The probabilistic approach to plausible reasoning on inexact data can be explained in terms of *conditional probability* (Shafer 1990). The conditional probability of ***a*** given ***d*** is the probability that ***a*** occurs if ***d*** occurs as defined by the *Bayes' Rule*, given in formula *(1)* in its simplest form and in formula *(2)* in a more general form.

$$P(a|d)=\frac{P(a\wedge d)}{P(d)}=\frac{P(d|a)P(a)}{P(d)} \tag{1}$$

$$P(a|d_1\wedge\ldots\wedge d_k)=\frac{P(d_1\wedge\ldots\wedge d_k|a)P(a)}{P(d_1\wedge\ldots\wedge d_k)} \tag{2}$$

P(a) is the *prior probability* of ***a***, that is, the probability prior to discovery of ***d***. ***P(a|d)*** is the *posterior probability*, that is, the probability once ***d*** has been discovered. The meaning of probability can be interpreted in two different ways, the subjectivists' and the objectivists' way. Subjectivists contend that the probability of an event is the degree to which someone believes that the event at issue is possible, as indicated by a person's willingness to place bets upon its occurrence. On the other hand, frequentists or objectivists contend that the probability of an event is the frequency with which it occurs.

One way in which this debate in mathematical statistics must be modified for application to AI is in its identification of subjectivity with prior- and joint-occurrences probabilities. Subjectivists and frequentists tend to agree on the objective (frequentist) character of the statistical model for the data. This model is usually only partially known, however, and the two schools have different views on how to handle this lack of knowledge. Subjectivists prefer to assess prior subjective probabilities for different possible statistical models and then use the data to update these prior probabilities to posterior probabilities. Frequentists, on the other hand, prefer to rely on the data alone to estimate the model. Nevertheless, in many problems of interest to AI, the kind of data analysed by subjectivists and frequentists is not available. Let us consider this issue for the case of facial expression recognition that is applicable to automated FACS coding and based upon the point-based face model employed by ISFER (Figure 5.2).

Estimating $P(a|d)$ given some set of facial actions A and some set of face model deformations D is not too problematic in the single model-deformation case. In that case, estimating $P(a|d)$ is limited to calculating for each facial action in A the conditional probability that the subject is displaying A given that a single model deformation in D is spotted in the observed image. Nevertheless, given m facial actions in A and n face deformations in D, $mn + m + n$ probabilities are required. This means that in the simplest case, that is, under the assumption that each facial action can be correctly encoded based merely on a single face-model deformation, 1088 probabilities are needed taking in consideration that ISFER automatically encodes 32 different facial actions (AU codes).

This is not a small number, and the situation gets considerably more complicated if a realistic system performance is to be achieved. When a single facial action is coded in an input static facial image, more than one model deformation should be taken into account since each facial action might cause several model deformations. The more general form of the Bayes' Rule given in formula *(2)* requires $(mn)^k + m + n^k$ probabilities, that is, 1049632 probabilities for even the most modest value $k = 2$ given $m = n = 32$. This is because $P(d_1|d_2 \wedge \ldots \wedge d_k)P(d_2|d_3 \wedge \ldots \wedge d_k)\ldots P(d_k)$ must be computed. Nevertheless, a simplification is possible under the assumption that the model deformations are independent on each other; in that case $P(d_i \wedge d_j) = P(d_i)P(d_j)$ and formula *(2)* does not require more probabilities than the single-model-deformation scenario.

It can therefore be concluded that the probability theory and the Bayes' Rule can provide a means for estimating the certainty of the data, which form the input to the reasoning mechanism of the Facial Action Encoder part of ISFER, only if:

- all the $P(d_j|a_i)$ are available, where d_j is a single deformation of the face model employed in ISFER and a_i is a single facial action, and
- it can be assumed that the data are independent, in which case the computation of the joint probabilities of model-deformation sets becomes feasible.

Yet none of these requirements holds in the case of automated facial action coding in static facial images based on the face model used by ISFER. Let us consider this issue in more detail and from the point of view of both the objectivists and the subjectivists.

Frequentists regard the probability $P(d_j|a_i)$ as a long-run relative frequency of that event which should be derived from an objective empirical investigation. However, in the case of facial action encoding based on the face model employed by ISFER, the necessary statistical data are difficult to obtain for a number of reasons. The existing body of literature on psychological and anatomical investigations of the face does not provide any cross-cultural statistical scrutiny on dependencies between each change in facial expression and each facial action. Hence, no source for the necessary statistical data is currently available. Nevertheless, let us assume that each conditional probability that a certain change in facial expression is visible if a certain facial action is present will be available after some period of time. In that case, yet another problem would be encountered. Even if cross-culturally and empirically defined frequencies of joint occurrences of changes in facial expression and facial actions would be available, there would still be no record at our disposal of dependencies between the facial actions and the relevant deformations of the face model employed by ISFER. In addition, a single change in facial expression is usually mapped on a set of deformations of the face model employed by ISFER. Therefore, the face-model deformations that model a single change in facial expression cannot be treated independently from each other. In turn, the problem of data dependency is encountered. Hence, a simplification of the Bayes' Rule given in formula *(2)* is not feasible. As a result, more than a million probabilities are required. Keeping track of dependencies between data, propagating probability updates and detecting occasional inconsistencies turns out to be intractable in this case. Furthermore, any facial action can be displayed at various intensity levels, causing weaker or stronger relevant face-model deformations. Hence, the problem becomes considerably more complicated since the track of dependencies between different intensity levels of the face-model deformations and the related facial actions should also be kept. Finally, the input data to the Facial Action Encoder which result from the feature detectors integrated into the Facial Data Extractor part of the system are redundant, moreover, they may be partial and of variable precision. This means that the probability $P(d_j)$ cannot merely be approximated with the long-run relative frequency with which the event d_j occurs; the overall accuracy of each detector which may be used for delimiting the deformation d_j as well as the degree to which the results of other detectors confirm this result must be taken into account. Altogether, this leads to the conclusion that a frequentists' probabilistic approach is not a convenient means for estimating and propagating data certainty within the Facial Action Encoder.

The arguments to dismiss a subjectivists' probabilistic approach are quite the same as the ones described above. This dismissal has been furthermore aided and abetted by another argument: a rigorous application of the Bayes' Rule would not have produced accurate probabilities in any case, since the used conditional probabilities would have been subjective (McCarty and Hayes 1969, Buchanan and

Shortliffe 1984). The issue here is that human beings do not appear to be reliable Bayesian reasoners. People are apt to discount prior odds and accord more weight to recently presented evidence (Kahneman and Tversky 1972), they are over-confident in their judgements (Kahneman et al. 1982), and have poor understanding of the sampling theory (Tversky and Kahneman 1990). However, any knowledge engineer probably seeks to represent an expert's knowledge of the world (imperfect though it may be), rather than to create a veridical model of the world. Nevertheless, even such an "imperfect" interpretation of probabilities turned out to be by no means a trivial task in the case of ISFER. Let me explain this by an example. A contraction of the lateral portion of the forehead muscle (FACS coded as AU2) results in an upward pull of the outer corners of the eyebrows. This means that AU2 is a proper coding of the change in facial expression in which the outer corners of the eyebrows are pulled up. However, the rule of FACS for recognition of AU2 states that AU2 should be scored in both cases: if the outer corners of both eyebrows are pulled up as well as if the outer corner of just one of the eyebrows is pulled up. In other words, the activation of AU2 might be bilateral as well as unilateral. As already mentioned above, subjectivists estimate the probability $P(d_j|a_i)$ based on the strength of a person's belief that the event at issue will indeed occur. But if we know that AU2 can be bilaterally as well as unilaterally activated, what is then a good probability estimate for the event "raised outer corner of the left eyebrow" versus the event "raised outer corner of the right eyebrow" given that AU2 is activated? Introducing multiple levels of activation intensity makes the problem even more difficult to handle: what is the strength of my belief that the outer corner of the left eyebrow would be raised for 40% given that the intensity of AU2 activation is 70%?

In summary, a rigorous application of the probability theory and the Bayes' Rule does not provide a convenient means for estimating and propagating data certainty at various reasoning levels of ISFER because:

- the assignment of probabilities to events, according either to the frequentists' view or to the subjectivists' view, requires information that is simply not available;
- the data-independence assumption cannot be made and the computation of the joint probabilities of face-model-deformation sets is therefore not feasible (i.e. "too many numbers" required); in turn, keeping track of data dependencies and accordingly updating belief values is intractable;
- keeping track of dependencies between different intensity levels of the face-model deformations and the related facial actions turns out to be intractable as well; moreover, it is not clear how one must deal with the interaction of the probabilities related to those events.

Due to these problems, a rigorous probabilistic formulation has not been adopted in tackling the problem of assessing the certainty of the data resulting from the Facial Data Extractor. However, in contrast to other formalisms for handling uncertainty, such as certainty factors and fuzzy logic, which assume that all prior probabilities ***P(d_j)*** are available, a probabilistic approach provides a means for actual calculation of these probabilities. So formalisms like certainty factors and fuzzy logic are not suitable for estimating the prior probabilities $P(d_j)$, where d_j is a face model deformation related to a facial feature that has been either inaccurately spatially sampled or redundantly detected by different detectors having variable precision. They do not facilitate dealing with redundant, approximate and partial data generated by the Facial Data Extractor. On the other hand, although a probabilistic approach is not a convenient means for propagating data certainty within ISFER, under certain considerations it might provide a means for estimating the prior probabilities of that data. These issues are discussed in the subsection concerned with the actually employed method for dealing with uncertainty in ISFER.

Certainty factors and belief functions approach

An alternative to the Bayesian reasoning with uncertainty is the use of *certainty factors* (CFs). Shortliffe and Buchanan first proposed this method within their expert system for medical diagnosis MYCIN (Shortliffe and Buchanan 1990). In terms of subjectivists' view, they defined the certainty factor $CF(h,e)$ as the change in belief that a hypothesis h is correct based on the evidence e:

$$CF(h,e)=\begin{cases} MB(h,e)=\dfrac{P(h|e)-P(h)}{1-P(h)} & P(h|e)>P(h) \\ MD(h,e)=\dfrac{P(h)-P(h|e)}{P(h)} & P(h)>P(h|e) \end{cases} \qquad (3)$$

where if e is supporting evidence such that $P(h|e) > P(h)$,where $P(h)$ is the expert's subjective probability that hypothesis h is correct, the increase of the expert's degree of belief in h will be given by the *measure of belief* (MB); if e constitutes evidence against h such that $P(h|e) < P(h)$, the increase of the expert's degree of disbelief in h will be given by the *measure of disbelief* (MD). The main uses of CFs are:

- to guide the program in its reasoning,
- to cause a hypothesis to be deemed unpromising if $CF \in [0.2, -0.2]$,
- to rank hypotheses after all the evidence has been considered.

The main criticism of certainty factors as defined by Shortliffe and Buchanan is that, in general, the CF associated with a hypothesis by MYCIN does not correspond to the probability of the hypothesis given the evidence if a simple probability model based on the Bayes' Rule is adopted (Adams 1976). Heckerman (1990) established a probabilistic semantics for the certainty-factors calculus used in MYCIN and showed that the belief changes are controlled by a monotonic transformation of likelihood ratio $P(e|h)/P(e|\text{-}h)$, a result obtained earlier by Good (1968) relative to the notion of "weight of evidence". This led further to uncovering of fundamental assumptions implicit in a certainty-factor model which turned out to be satisfied only in tree-structured networks where no evidence bears on more than one hypothesis variable. Otherwise, as shown by Adams (1976), in some circumstances the ranking of hypotheses will depart from the ranking produced by applying the probability theory.

The example he gives is the following. Let h_1 and h_2 be two hypotheses and e supportive evidence for both hypotheses. Let h_1 have a higher subjective prior probability than h_2 and let this superiority remain after the evidence is taken in consideration. Thus, let $P(h_1) \geq P(h_2)$ and $P(h_1|e) > P(h_2|e)$. Under these circumstances it should hold that $CF(h_1,e) > CF(h_2,e)$. Yet, suppose that $P(h_1) = 0.9$, $P(h_2) = 0.2$, $P(h_1|e) = 0.95$, $P(h_2|e) = 0.7$. Then the increase of belief in h_1 and the increase of belief in h_2 are given by:

$$MB(h_1, e) = \frac{0.95 - 0.9}{1 - 0.9} = 0.5 \text{, respectively } MB(h_2, e) = \frac{0.7 - 0.2}{1 - 0.2} = 0.625 .$$

Thus, $CF(h_1,e) < CF(h_2,e)$ even though $P(h_1|e) > P(h_2|e)$.

In the case of automated facial action coding based on the face model employed by ISFER, a single deformation of the model may bear on the encoding of several AU codes. For example, the distance between the left mouth corner and the inner corner of the left eye is a model deformation that bears on the encoding of three different AU codes: an upward pull (AU12), sharp upward pull (AU13), and downward pull of the mouth corners (AU15). As we have already seen, a certainty-factor model is not suitable for the cases where single evidence bears on more than one hypothesis. This forms the primary argument against the employment of certainty-factors-based inference in the Facial Action Encoder.

Another argument against applying certainty factors for estimating and propagating data certainty at various reasoning levels of ISFER is abetted by the Horvitz-Heckerman criticism of MYCIN (1986). They state that Shortliffe and Buchanan use certainty factors as measures of change in belief while certainty factors were actually elicited from experts as degrees of absolute belief. Yet it is not possible to elicit degrees of either absolute or some "temporary" belief in a certain event since that requires information which is simply not available. For instance, if we know that an upward pull of the left mouth corner bears on the encoding of AU12 as well as on the encoding of AU13, what is then a good probability estimate for the event “AU12 activated” versus the event “AU13 activated” given that the left mouth corner is raised? The situation becomes even more complicated when dependencies are introduced between different intensity levels of the face-model deformations and the related facial actions. For example, what is the degree of my belief in the event “AU12 activated with 60% of intensity” given that the left mouth corner is raised for 60%? Due to these problems a certainty-factors-based inference is not applicable in the case of automated facial action encoding based on the face model employed by ISFER.

Another alternative approach to inexact reasoning is the theory of belief functions, also called the Dempster-Shafer theory of evidence. While certainty factors represent a rather ad hoc approach to plausible reasoning, the Dempster-Shafer theory is theoretically well founded. This theory assumes that there is a fixed set, a so-called *frame of discernment*, of mutually exclusive and exhaustive elements (hypotheses, conclusions) which are the subject of reasoning. Further, there is a certain fixed amount of belief (=1) that is distributed over all subsets of the frame of discernment including the empty subset representing the environment. Newly encountered evidence causes a redistribution of the belief among the subsets. This redistribution is defined by the Dempster's Rule of Combination (Shafer and Srivastava 1990) and it is done such that the sum of distributed belief always remains 1. Finally, a *belief interval* of any given focal element A is defined as [***Bel(A)***, ***Pls(A)***], where ***Bel(A)*** is the total belief of a given set A and all its subsets, and ***Pls(A)*** is the plausibility of A defined as ***Pls(A)*** = 1-***Bel(¬A)***. The width of the belief interval is regarded as the amount of uncertainty with respect to an element (hypothesis, conclusion) given the available evidence.

An argument to dismiss the Dempster-Shafer theory as a means for estimating and propagating data certainty at various reasoning levels of ISFER is that the hypotheses in the Dempster-Shafer theory are assumed to be both exhaustive and mutually exclusive. However, neither of these assumptions can be made in the case of facial action encoding based on the face model employed by ISFER. As explained above, a single deformation of the face model may bear on the encoding of several AU codes. As a result, mutual exclusiveness cannot be assumed. In addition, from a total of 44 AUs defined in FACS, the employed face model facilitates a unique representation of merely 29 different AUs (Table 5.8). As a result, the system may describe two different facial expressions (i.e. unlike in terms of displayed facial actions) using identical AU-coded descriptions. When this drawback of the employed face model is taken in consideration, exhaustiveness cannot be assumed.

Fuzzy Logic approach

The knowledge which an expert uses to interpret some signal or to perceive a symptom as a manifestation of a particular change or disorder is usually based on relations between classes of data and classes of hypotheses, rather than on individual data and hypotheses. Many forms of problem solving involve some kind of data classification; for instance, specific sensor signals, disease symptoms, changes in facial expression are likely to be seen as the instances of more general categories (e.g. malfunction classes, classes of diseases, emotion classes). Yet such categories may not be accurately defined. Hence, class membership may be difficult to assess: a datum may exhibit some properties of the class, but not all of them, or it may exhibit the relevant properties only to a certain degree. Fuzzy set theory (Zadeh 1965) is a formalism for reasoning about such phenomena forming the basis of both fuzzy logic (Zadeh 1975) and the possibility theory (Zadeh 1978).

In the classical set theory a set is a collection of any number of definite distinguished objects that share common properties. For example, if A is the set of distances that are 40 centimetres, then for x to belong to A, x must be 40. So if a distance is 39.9 or 5 centimetres, the distance will be excluded from A. Classifying a 5 centimetres long distance and a 39.9 centimetres long distance in the same category, while at the same time classifying a 39.9 centimetres long distance and a 40 centimetres long distance into two different categories clearly demonstrates the kind of inconsistencies associated with the classical set theory. Therefore, classical sets are sometimes referred to as crisp sets. The crispness of the classical set theory poses a problem when we deal with concepts that are not accurately defined.

This kind of observations led to the development of fuzzy set theory. The term *fuzzy* was introduced by Zadeh to describe the sets whose membership criteria are vague. In contrast to the classical set theory, an element can belong to a fuzzy set, be completely excluded from a fuzzy set, or it can belong to a fuzzy set to any intermediate degree between these two extremes. The extent to which an element belongs to a given fuzzy set is called the *degree of membership*. Uncertainty about the statement that a number belongs to a given fuzzy set is not represented by the probability that the number belongs to that set, but rather by the possibility that the number belongs to the set. A so-called degree of membership of a particular number in a particular fuzzy set, generated by a so-called *membership function* represents the possibility of truth that the number belongs to the fuzzy set. As given in formula *(4)*, a fuzzy set $\boldsymbol{F}$ can be viewed as an association between numbers in which to each element $\boldsymbol{f} \in \boldsymbol{F}$ one and only one degree of membership $\boldsymbol{\chi_F(f)}$ is assigned, where $\boldsymbol{\chi_F}$ is the related membership function.

$$\boldsymbol{F : f \in F \rightarrow \chi_F(f)} \in [0,1] \qquad (4)$$

Consider the concept denoted by the word "heavy" as applied to objects. Given the vagueness of the concept, how do we characterise a set of heavy objects? In the classical set theory, we could describe a set ***HEAVY*** by characterising the set of objects heavier than 100kg as $\{\boldsymbol{x} \in \boldsymbol{HEAVY} \mid \boldsymbol{weight(x)}>100\boldsymbol{kg}\}$. However if my ambition is to select a heavy object but I am presented with a limited choice, then an object of 80kg is still a better choice than an object of only 50kg. But neither an object of 80kg nor an object of 50kg would be a member of the delimited ***HEAVY*** set and I would not be able to make any choice. So I need a set of heavy objects that could be characterised by the function from the domain of available objects. Such a set could be a fuzzy set ***HEAVY*** defined by the set of weights of the available objects {40kg, 50kg, 60kg, 70kg, 80kg, 90,kg, 100kg} and the corresponding degrees of membership {0.00, 0.05, 0.30, 0.50, 0.85, 0.95, 1.00}.

The fuzzy set ***HEAVY*** can be represented graphically as illustrated in Figure 3.10. In the first graph, ***HEAVY*** is represented as a discontinuous function which does not take account of the intermediate weight values. In order to account for an intermediate weight value, the values for the degree of membership should be interpolated. An interpolation can be achieved by utilising smoothing functions like a Sigmoid function or a π-function (given in formula *(5)*) or the piecewise-linear interpolation illustrated in Figure 3.10.

$$S(x;\alpha,\beta,\gamma)=\begin{cases} 0 & x\leq\alpha \\ 2\dfrac{(x-\alpha)^2}{(\gamma-\alpha)^2} & \alpha<x\leq\beta \\ 1-2\dfrac{(x-\gamma)^2}{(\gamma-\alpha)^2} & \beta<x<\gamma \\ 1 & x\geq\gamma \end{cases}$$

$$\pi(x;\beta,\gamma)=\begin{cases} S\left(x;\gamma-\beta,\gamma-\dfrac{\beta}{2},\gamma\right) & x\leq\gamma \\ 1-S\left(x;\gamma,\gamma+\dfrac{\beta}{2},\gamma+\beta\right) & x>\gamma \end{cases} \tag{5}$$

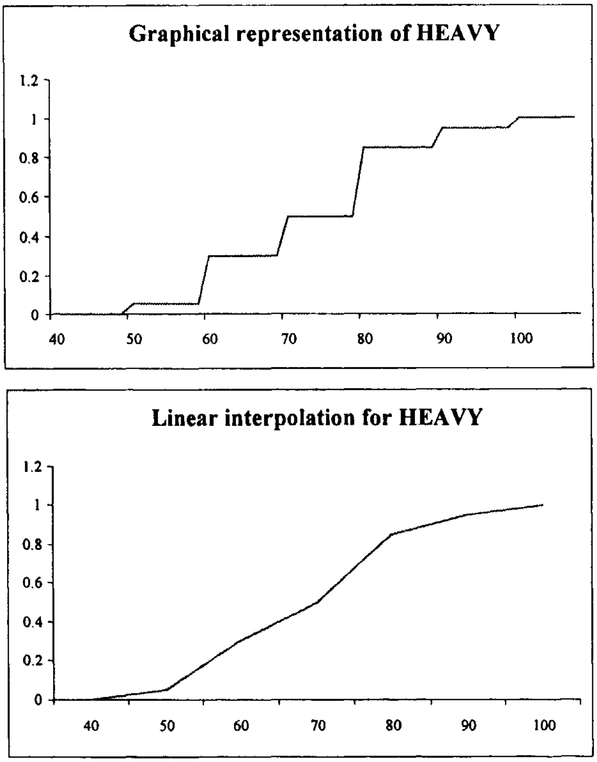


Figure 3.10: Discontinuous HEAVY and piecewise-linearly interpolated HEAVY

The main difference between probability theory and fuzzy logic is that a probability is regarded as an approximation to something more precise. There is a 50% chance that a fair coin will come down heads, but when tossed the coin will come down 100% heads or 100% tails. In the "heavy object" example, the intended meaning of ***HEAVY***(90kg) = 0.95 is not that an object of 90kg is 100% heavy or 100% not heavy but that we are 95% sure that the object is heavy. The uncertainty is inherent in the vagueness of the concept. Thus it seems reasonable to suppose that there remains a degree to which the object in question is not heavy (e.g. it is light by comparison with a concrete block of one ton).

Fuzzy logic deals with situations where the questions that we pose and the relevant knowledge that we possess both contain vague concepts. However, vagueness is not the only source of uncertainty; sometimes we are simply unsure of the facts (e.g. of the data resulting from the Facial Data Extractor part of ISFER). Possibility theory is a species of fuzzy logic for dealing with precise questions on the basis of imprecise knowledge. To illustrate the relation between fuzziness and possibility, we can use an example. A non-fuzzy statement "x is an integer in the range [0,5]" has the following meaning:

$$\begin{array}{ll} \boldsymbol{Poss}(x=\boldsymbol{u})=1 & 0\le \boldsymbol{u}\le 5 \\ \boldsymbol{Poss}(x=\boldsymbol{u})=0 & \boldsymbol{u}<0\vee \boldsymbol{u}>5 \end{array}$$

A fuzzy version of the above statement "x is a small integer in the range [0,5]" can have, for example, the following meaning:

$$\begin{array}{llll} \boldsymbol{Poss}(x=0)=1 & \boldsymbol{Poss}(x=2)=0.9 & \boldsymbol{Poss}(x=4)=0.3 & \boldsymbol{Poss}(x=\boldsymbol{u})=0 \\ \boldsymbol{Poss}(x=1)=1 & \boldsymbol{Poss}(x=3)=0.7 & \boldsymbol{Poss}(x=5)=0.2 & \boldsymbol{u}<0\vee \boldsymbol{u}>5 \end{array}$$

In the case of the fuzzy proposition, the possibility is assigned a value from the interval [0, 1], rather than it is labelled either possible or impossible, as is the case with the classical proposition. The possibility of the fuzzy proposition stated above represents, in fact, the degrees of membership {1, 1, 0.9, 0.7, 0.3, 0.2} of the corresponding elements {0, 1, 2, 3, 4, 5} of the fuzzy set ***SMALL_INTEGER***. As mentioned above, Zadeh rather refers to uncertainty whether an object belongs to a fuzzy set as the possibility that the object belongs to the set than as the probability that the object belongs to the set. The differences between possibility and probability can be summarised as:

- The probabilities have to sum to 1. The possibilities are not restricted in such way.
- A high possibility does not imply a high probability.
- A small possibility usually implies a small probability but a small probability does not imply a small possibility.

Fuzzy logic deals with negation, conjunction and disjunction in the following way:

- If $f \in F$, where F is a fuzzy set (category), $\chi_{\neg F}(f) = 1 - \chi_F(f)$.
- If $f \in F, f \in K$, and F and K are fuzzy sets, $\chi_{F \wedge K}(f) = \boldsymbol{min}(\chi_F(f), \chi_K(f))$.
- If $f \in F$ or $f \in K$, and F and K are fuzzy sets, $\chi_{F \vee K}(f) = \boldsymbol{max}(\chi_F(f), \chi_K(f))$.

The min and max operators are commutative, associative, and mutually distributive. Like the operators of standard logic, they obey the principle of *compositionality*. This means that the values of compound expressions are computed from the values of their component expressions and nothing else. This is in contrast to the laws of probability, where conditional probabilities must be taken into account when conjunction and disjunction are computed. In turn, this facilitates a way for dealing with dependant data like in ISFER. If a fuzzy logic approach is employed for propagating data certainty in ISFER, the certainty about an encoded facial action (compound expression) can be computed from the certainties about the detected face-model deformations (component expressions) that reveal the pertinent facial action. In other words, keeping track of dependencies between the face-model deformations and the related AU codes and propagating the certainty updates is tractable in that case. The actual implementation of a fuzzy logic approach within the Facial Action Encoder is explained in chapter 5.

Dealing with uncertainty in ISFER

At this point, it can be concluded that a rigorous application of any of the three principal formalisms for handling uncertainty is not suitable for estimating and propagating certainty of the data resulting from the Facial Data Extractor. The certainty factors, belief functions, or fuzzy logic do not provide a means for estimating the certainty of redundant, approximate, and/or partial data which might be generated by the Facial Data Extractor since those methods a priori assume availability of those certainties (i.e. availability of the prior probabilities $\boldsymbol{P(d_j)}$). A data independence assumption cannot be made and the track of dependencies between different intensity levels of the face-model deformations and the related AU codes must be kept. This implies the inapplicability of an approach that would be based exclusively upon the probability theory.

On the other hand, the knowledge about how to estimate the certainty of the data resulting from the Facial Data Extractor does exist. For instance, it is known that data redundancy can be exploited to compare the results of different detectors. If different detectors yield the same spatial sampling of the contour of a certain facial feature, then that datum should have a higher certainty. This can be also expressed in the following way: the larger the number of different detectors yielding the same spatial sampling of a certain facial feature's contour, the higher the certainty about that datum. Further, the Facial Data Extractor generates data of variable precision in the sense that a certain detector may be more suitable for detection of a certain feature than another. Hence, the certainty of the data which represents the spatial sampling of a facial feature obtained by a given detector can be viewed as the frequency with which that facial feature has been successfully spatially sampled by the given detector. In addition, it is also known that some of the facial points are immovable (stable) in the sense that no facial action can cause the displacement of these points (FACS, Ekman and Friesen 1978). Such points are the inner corners of the eyes, the inner corners of the nostrils, and the medial point of the mouth (for a detailed discussion about stable facial points, the reader is referred to section 5.4). Since the camera setting (two head-mounted cameras; see also sections 2.6 and 4.1) ensures scale- and orientation-invariant images acquired during a single session, the location of the stable facial points should remain the same during the entire session. Therefore, the detected displacement of these model points from the relevant model points extracted from an expressionless face of the observed subject represents in fact the degree of the detection error. Hence, the certainty of the data which represents the spatial sampling of a facial feature obtained by a given detector can be estimated based upon the error made by the given detector while localising the stable points belonging to the facial feature at issue; namely the larger the degree of the detection error, the lower the certainty about the data at issue. Furthermore, it is also known that people display some *typical* facial expressions more often than some other expressions. The typicality of a facial expression can be viewed as the frequency with which that expression occurs. In addition, the probability of an expression whose occurrence might be expected but has not been actually detected can be estimated based on the typicality of that expression. Finally, as already noted above, keeping track of dependencies between the model deformations and the related AU codes and propagating the certainty updates is tractable if a fuzzy approach is applied.

These observations are reflective on a situation in which the properties used to associate the elements in a set *V* with those in a set *U* are uncertain, but we know the process used to select the properties which play a role in the association. In other words, the above listed observations model a situation in which the knowledge of the appropriate functional form is expressed by process knowledge that could be formalised either in terms of probabilities or in terms of possibilities. For a profound discussion on models which exploit process knowledge of probabilistic type, the reader can consult the Bayesian transition matrices (Yager 1988).

Dealing with partial data

In case a certain facial feature (e.g. an eyebrow, an eye or the mouth) fails to be detected by the facial feature detectors integrated into the Facial Data Extractor, the Facial Action Encoder utilises the pertinent facial feature detected in the expressionless face of the observed person to substitute missing data (sections 2.6). Hence, exact information about the examined expression is lost. This information loss can be compensated in two different ways. The first one is to exploit a "higher-level grammar of basic emotional expressions", that is, to reason about possible but yet undetected changes in facial expression in the context of emotional expressions of which the actually detected changes in facial expression make a part. In that case, the following processing might be applied:

1. *forward reasoning*: classify the actually detected changes in facial expression (coded in terms of AU codes) under one of the six basic emotion categories (Pantic et al. 1998b, 1999b)
2. *backward reasoning*: specify the appropriate facial appearance (i.e. AU code) of the undetected feature by finding the best match between the AU-coded description of the observed facial expression and the AU-coded description of the facial expression that characterises the emotion category delimited in step 1 (Pantic and Rothkrantz 2000a).

The motivation that underlies this reasoning process is based on the expectation that a smile, for instance, is coupled with "smiling" eyes rather than with expressionless eyes. However, this approach has several limitations. First, not every facial expression that can be displayed by the face can be classified under one of the six basic emotion categories. In other words, a smile could be coupled with "smiling" eyes in an expression of joy, with wide opened eyes in an expression of pleasant surprise, or with any other expression of the eyes. In turn, this approach would in any case not produce an accurate assessment of the appearance (i.e. AU code) of an undetected facial feature since the used emotional classification of facial expressions would be singular and limited to the six basic emotion categories. Furthermore, people display some typical facial expressions more often than other ones. Hence, even if the employed classification would categorise the examined expression into multiple user-defined interpretation categories, this approach would not produce an accurate assessment of the appearance (i.e. AU code) of an undetected facial feature since the typicality of an expression would not be taken into account.

Another way of dealing with partial data resulting from the Facial Data Extractor is to apply a frequentists' probabilistic approach. Such an approach seems very suitable under the consideration that the typicality of an expression can be viewed as the frequency with which that expression occurs. Expressing the typicality of facial expressions by process knowledge of a probabilistic type is reflective on a situation in which the characteristic (typicality) used to associate the elements (expressions) of some set $\boldsymbol{V}$ with some value $\boldsymbol{U}$ (degree of typicality) is uncertain, but we know the process (calculating the frequency of each expression) that can be used to assign this value.

We have seen that the Bayes' Rule is applicable only if all the inverse conditional probabilities are available and the data-independence assumptions can be made such that the computation of the joint probabilities is feasible. When applied to partial data from the Facial Data Extractor, the Bayes' Rule given in formula *(6)* computes the conditional probability that the displayed facial expression is AU-coded as $(\boldsymbol{AU}_1+...+\boldsymbol{AU}_i+\boldsymbol{AU}_{i+1})$ given that the actually detected facial expression is AU-coded as $(\boldsymbol{AU}_1+...+\boldsymbol{AU}_i)$.

$$P(AU_1+\cdots+AU_i+AU_{i+1}|AU_1+\cdots+AU_i)=$$
$$\frac{P(AU_1+\cdots+AU_i|AU_1+\cdots+AU_i+AU_{i+1})P(AU_1+\cdots+AU_i+AU_{i+1})}{P(AU_1+\cdots+AU_i)} \qquad (6)$$

Since the activation of each and every AU which is a part of $(AU_1+...+AU_i)$ forms a prerequisite for $(AU_1+...+AU_i+AU_{i+1})$, then for any combination $(AU_1+...+AU_i)$ of activated AUs $P(AU_1+...+AU_i|AU_1+...+AU_i+AU_{i+1}) = 1$. Furthermore, it is not essential to prove the data independence since each combination of activated AUs can be regarded as a single event and a more general form of the Bayes' Rule (like the one given in formula *(2)*) is unnecessary in that case. In turn, the Bayes' Rule provides a convenient means for dealing with partial data from the Facial Data Extractor. The actual implementation of a Bayesian approach to deal with partial data resulting from the Facial Data Extractor is explained in detail in section 5.6.

Dealing with approximate data

The Facial Data Extractor generates data of variable precision in the sense that a certain detector may be more reliable in detecting a certain feature than another detector. In turn, the certainty of the data that represents the spatial sampling of a facial feature obtained by a given detector can be viewed as the reliability of the detector at issue, that is, as the frequency with which that facial feature has been successfully spatially sampled by the given detector. Expressing the reliability of the data generated by a given detector while spatially sampling a certain facial feature by process knowledge of a probabilistic type reflects a situation in which the characteristic (reliability) used to associate the elements (detected feature) of some set V with some value U (degree of reliability) is uncertain, but we know the process (calculating the frequency with which the detector at issue successfully detects the facial feature at issue) used to assign this value. Hence, the certainty of a facial feature contour localised by a given detector will be estimated as the proportion between the number of test cases for which the given detector successfully sampled the facial feature at issue and the total number of test cases.

Nevertheless, this approach introduces a set of complications. First, the notion of successful spatial sampling of a given facial feature should be defined. A localisation of the contour of the given facial feature with an average localisation error of 12 pixels per contour point should be less "successful" than a localisation with an average error of 2 pixels per contour point. In addition, a localisation with an average error of 12 pixels per contour point is not useful at all for the facial expression analysis if the model deformations that can uncover a certain facial action are approximately of the same scale. Let me assume that for a certain image resolution and for each prominent facial feature it is possible to define a maximal average localisation error with which the achieved localisation of the facial feature will still be useful for the facial expression analysis. Then, by delimiting the displacement between the currently localised stable facial points belonging to a certain feature and the pertinent points localised in an expressionless face of the observed subject, the performed localisation could be classified as either successful or unsuccessful. Still another problem remains. Independently of whether in the currently examined image a given detector successfully localises a facial feature with the maximal localisation error "allowed" for that feature or it localises it with a much smaller error, the certainty measure assigned to the pertinent generated data will be the same. Obviously, this is a serious drawback of the proposed frequentists' probabilistic approach.

Another way of dealing with approximate data from the Facial Data Extractor is to exploit the knowledge about the facial anatomy and dynamics. As already noted above, some of the facial points are stable in the sense that no facial action can cause a displacement of these points (e.g. the inner corners of the eyes, the inner corners of the nostrils, and the medial point of the mouth). Since all the images acquired during a single session are scale and orientation invariant, the location of the stable facial points should remain the same during the entire session. Hence, the certainty of a facial feature contour localised by a given detector can be estimated based on the error made by the given detector in localising the stable points belonging to that feature: the larger the error, the lower the certainty of the data. The goal is to define a mapping between the displacement of the actually detected stable facial point from the pertinent point detected in an expressionless face of the observed subject (localisation error ***Er***$\in$[0,max(displace)]) and the certainty measure, ***CM***$\in$[0,100] percent, assigned to the feature to which the stable point belongs. The mapping should not have a detrimental effect upon our ability to act. In other words, if the localisation error exists, say ***Er*** = 5, then the performed facial feature detection is still better than that the one having an error ***Er*** = 12. Furthermore, the mapping should be defined in such a way that a localisation error ***Er*** = 0 corresponds to a certainty measure ***CM*** = 100%. Hence, the intended meaning of the mapping is not that a performed feature detection is 100% accurate or 100% inaccurate but that we are sure to some degree ***CM***, based upon the measured error ***Er***, that the performed feature detection is accurate. The uncertainty about the accuracy of the performed feature detection therefore can be expressed as the possibility that this detection is accurate rather than the probability that this detection is accurate. In turn, a functional form defined in the possibility theory (e.g. a Sigmoid membership function or a Gaussian membership function) would be probably the best choice for representing the intended mapping.

For example, the graphical representation of the mapping ***CERTAINTY***(***eye_Er***) = ***CM*** may be illustrated as given in Figure 3.11 under the following assumptions:

- for a 720×576 pixels image resolution the maximal localisation error of the inner corner of the eye with which the eye detection will be still useful for further analysis is ***eye_Er*** = 5,

- for a 720×576 pixels image resolution the maximal localisation error of the inner corner of the eye with which the eye detection will be still considered as "well performed" is *eye_Er* = 3.

Section 5.4 explains for each prominent facial feature and a given image resolution how to determine the maximal average localisation error with which the localisation of a certain facial feature is still useful for the facial expression analysis. Section 5.4 also discusses in detail the choice of functional representation for the mapping between the actual localisation error made in localising a certain feature and the certainty measure assigned to that feature.

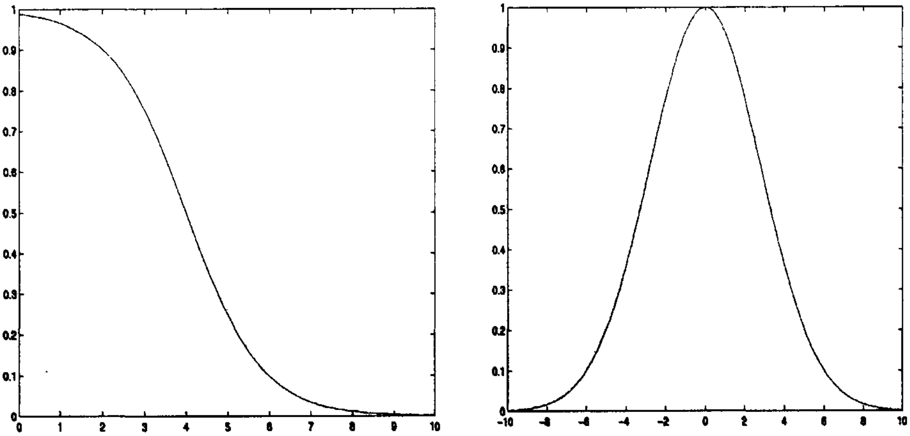


Figure 3.11: Approximating the mapping *CERTAINTY(eye_Er)=CM* with a Sigmoid membership function and a Gaussian membership function; in the case of Gaussian function, only the positive half of the x-axis will be actually exploited

Dealing with redundant data

The Facial Data Extractor generates redundant data when several of the integrated facial feature detectors successfully sample the contour of the same prominent facial feature. Data redundancy can be dealt with by choosing the best from the results of all detectors that localise the same facial feature, that is, by simply classifying those results according to the certainty measures assigned to them as explained in the previous subsection. The datum to which the highest certainty measure has been assigned will be used in a further processing of ISFER while the rest of data will be discarded.

However, as noted above, data redundancy can be exploited to compare the results of different detectors, that is, to make inter-detector accuracy checks. If different detectors yield the same spatial sampling of the contour of a certain feature, then the results of those detectors confirm each other and the resulting data should have a higher certainty. This might be also expressed in the following way: the larger the number of different detectors yielding the same spatial sampling of the contour of a certain facial feature, the higher the certainty of that data. This means that a datum d_i to which a lower certainty measure has been assigned during the process explained in the previous subsection might still be chosen over a datum d_j, where $i \neq j$, to which a higher certainty measure has been assigned if other available data confirm the datum d_i. Section 5.4 explains in detail the actual implementation of both the inter-detectors consistency checks and the procedures for selecting the best of the data redundantly extracted from an input facial image.

3.5 Machine learning

This section is concerned with the field of machine-learning and provides a short overview of a number of well-known learning paradigms. For a more profound discussion about different learning algorithms, theoretical results, and applications, the reader is referred to (Mitchell 1997).

Machine learning is inherently a multidisciplinary field. It draws on results from research fields as diverse as:

- *Artificial Intelligence*: AI forms a theoretical and methodological basis for learning symbolic representations of concepts, learning in terms of classification and pattern recognition problems, and learning by using prior knowledge together with training data as a guideline.
- *Bayesian methods*: the Bayes' theorem forms the basis for calculating probabilities of hypotheses, the basis of the naïve Bayes classifier, and the basis of algorithms for estimating values of unobserved variables.
- *Computational complexity theory*: This theory imposes the theoretical bounds on the inherent complexity of different learning tasks measured in terms of computational effort, number of training examples, number of mistakes, etc.
- *Control theory*: This theory forms the theoretical foundation of procedures that learn to control processes in order to optimise predefined objectives and to predict the next state of the process they are controlling.
- *Information theory*: Measures of entropy and optimal codes are germane and central to the issue of delimiting optimal training sequences for encoding a hypothesis.
- *Philosophy*: Philosophical argumentations like "the simplest hypothesis is the best" underlie the reasoning process of machine learning algorithms.
- *Psychology*: The view on human reasoning and problem-solving initiated many machine learning models (e.g. see the discussion on CBR in section 3.6).

- *Neurobiology*: Information processing found in biological organisms motivated ANN models of learning (section 3.3).

As delimited by the definition given by Mitchell (1997), a computer program is said to learn from experience ***E*** with respect to some class of tasks ***T*** and performance measure ***P*** if its performance at tasks in ***T***, as measured by ***P***, improves with experience ***E***. For example, the Facial Expression Classifier part of ISFER which classifies facial expressions in terms of user-defined interpretation labels (chapter 6), improves its performance *as measured by its ability to accomplish user-defined interpretations* at the class of tasks involving *classification of facial expressions*, through experience *obtained by interacting with the user on the meanings that he/she associates with different facial expressions*. In general, in a well-defined learning problem, these three features must be identified (i.e. the class of tasks ***T***, the measure of performance to be improved ***P***, and the source of experience ***E***). Once the learning problem is defined, the next step in designing a learning system is to delimit exactly:

- the type of knowledge to be learned,
- the representation of this target knowledge (i.e. the definition of target function to be learned, which when utilised will produce for any instance of a new problem as input a trace of its solution as output), and
- the learning mechanism to apply.

Different target knowledge (hypotheses space) representations are appropriate for learning different kinds of target functions. For each of these hypothesis representations, the corresponding learning algorithm takes advantage of a different underlying structure to organise the search through the hypotheses space. Therefore, deciding about the issues listed above involves searching a very large space of alternative approaches to determine the one that best fits the defined learning problem. In order to decide a machine learning algorithm which will perform best for the given problem and the given target function, it is useful to analyse the relationships between the size of the hypotheses space, the completeness of it, the number of training examples available, the prior knowledge held by the learner, and the confidence we can have that a hypothesis that is consistent with the training data will correctly generalise to unseen examples.

Though, generally, learning is considered as one of the basic facets of intelligence, not all AI techniques are capable of learning. Expert systems are an obvious example, at least in their most common form (section 3.2). The primary algorithms and approaches to machine learning are described below.

Decision trees

Decision tree learning is one of the most widely used and practical methods for inductive inference. It is a method for approximation of discrete-valued functions, in which a tree represents the learned function. A decision tree is in a nutshell a discrete value functional mapping, a classifier. Each node in the decision tree specifies a test of some attribute of the query instance, and each branch descending from that node corresponds to one of the possible values for this attribute. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute. This process is repeated for the subtree rooted at the new node as long as it takes to reach the appropriate leaf node, then returning the classification associated with this leaf. Several algorithms are available that can be used to construct a tree based on some data set. A typical example is the ID3 algorithm proposed in (Quinlan 1993). This is a greedy search algorithm that constructs the tree recursively and chooses at each step the attribute to be tested so that the separation of the data examples is optimal. This decision-tree learning method searches a complete hypothesis space (i.e. the space of all possible decision trees) and, thus, avoids difficulties of restricted hypothesis spaces (i.e. that the target function might not be present in the hypothesis space). Its inductive bias is a preference for small trees over large trees. Experiments that compare decision-tree learning and other learning methods can be found in numerous papers, for example, in (Weiss and Kapouleas 1989), (Thrun 1991) and (Dietterich et al. 1995).

Artificial neural networks

ANNs provide a general, practical method for learning real-valued, discrete-valued, and vector-valued target functions from examples. Algorithms such as backpropagation use gradient descent to tune network parameters to best fit a training set of input-output pairs. ANN learning is robust to errors in the training data and has been successfully applied to problems such as interpreting visual scenes, speech recognition, etc. (section 3.3).

Learning set of rules

One of the most expressive and human readable representations of a learned target function is a set of if-then rules that jointly define the function. One way to learn sets of rules is to learn a decision tree first, then translate the tree into an equivalent set of rules; one rule for each leaf node in the tree. A quite successful method for converting the learned tree into a set of rules is a technique called *rule post pruning* used by the C4.5 algorithm (Quinlan 1993), which represents an extension of the original ID3 algorithm.

Another way to convert a tree into a set of rules is to apply a *sequential covering algorithm* for learning sets of rules based upon the strategy of learning one rule, removing the data it covers and then iterating this process. To elaborate, given a LSR (learn-single-rule) subroutine, invoke it on all the available training examples, remove any positive examples covered by the rule it learns, then invoke it again to learn a second rule based on the remaining training examples. Thus, a sequential covering algorithm sequentially learns a set of (disjunctive) rules that together cover the full set of positive examples. Because this algorithm carries out a greedy search, so it formulises a sequence of rules without backtracking, the smallest or best set of rules that cover the training examples is not necessarily found. A prototypical sequential covering algorithm is the *general-to-specific beam search* which searches through the space of possible rules maintaining k best candidates, then generates descendents for each of these k best candidates, and again reduces the resulting set to k most promising members. This algorithm has been used by the CN2 program (Clark and Niblett 1989). Many variations on this approach have been explored (e.g. specific-to-general search like GOLEM (Muggleton 1992) and example-driven searches such as FIND-S and CANDIDATE-ELIMINATION (Mitchell 1997)).

Inductive logic programming

The previous subsection discussed algorithms for learning sets of propositional (i.e. variable-free) rules. This subsection is considered with learning rules that contain variables, in particular, learning first-order Horn theories. Inductive learning of first-order rules is also referred to as *Inductive Logic Programming* (ILP), because this process can be viewed as automatically inferring PROLOG[1] programs from examples. A variety of algorithms has been proposed for learning first-order rules. A typical example is FOIL (Quinlan 1990), which is an extension of the sequential covering algorithms to first-order representations.

Another approach to inductive logic programming is *inverse deduction*, which is based upon the simple observation that induction is just the inverse of deduction. In other words, the problem of induction is to find a hypothesis h that satisfies the constraint $(\forall\langle x_i, f(x_i)\rangle \in D)\ (B \wedge h \wedge x_i) \vdash f(x_i)$, where B is general background information, $x_1 \ldots x_n$ are descriptions of the instances in the training data D, $f(x_1) \ldots f(x_n)$ are the target values of the training instances, and expression $Z \vdash C$ is read "C follows deductively from Z". A prototypical algorithm based upon inverse deduction principle is CIGOL (Muggleton and Buntine 1988), which uses the *inverse resolution*, an operator that is the inverse of the *deductive resolution* operator introduced by Robinson (1965) and commonly used for mechanical theorem proving. For further reading on ILP, the reader can consult (Lavrac and Dzeroski 1994), (De Raedt 1996), and (Furukawa et al. 1999).

Instance-based learning

In contrast to learning methods that construct a general, explicit description of the target function when training examples are provided, instance-based learning methods simply store the training examples. Generalising beyond these examples is postponed until a new instance must be classified: given a new instance, its relations

[1] PROLOG is a general purpose, Turing-equivalent programming language in which programs are expressed as collections of Horn clauses.

to the already stored examples are examined in order to assign a target function value (the classification) for the new instance. Due to this property, instance-based learning methods are also called *lazy learning methods*, as opposed to the *eager learning methods* represented by all other learning algorithms discussed in this section. Examples of instance-based learning include nearest-neighbour learning and locally weighted regression methods. Instance-based learning also includes case-based reasoning methods that use more complex, symbolic representations for instances. Early theoretical results on nearest-neighbour learning algorithms can be found in (Cover and Hart 1967), while an overview of the topic can be found in (Mitchell 1997). A survey of methods for locally weighted regression is given in (Atkenson et al. 1997). Section 3.6 provides a detailed discussion on case-based reasoning.

A key advantage of instance-based learning as a delayed, or lazy, learning method is that instead of estimating the target function once for the entire instance space, these methods can estimate it locally and differently for each new instance to be classified. Yet, these methods are at a disadvantage because of their computation and memory/storage requirements.

Genetic algorithms

Genetic Algorithms (GA) are optimisation techniques providing an approach to learning that is based loosely on simulated evolution. One thing that distinguishes GA from other optimisation algorithms is that GA simultaneously work on large sets (populations) of possible solutions. The search for an appropriate hypothesis begins with a population of initial hypotheses. Members of the current population give rise to the next generation population by means of operations such as random mutation and crossover, which are patterned after biological evolution processes. At each iteration, the hypotheses in the current population are evaluated relative to a given measure of fitness and the most fit members of the population are selected to produce new offspring that replace the least fit members of the population. To elaborate, the learning task of GA is to find the optimal hypothesis according to the predefined fitness function.

Evolution-based computational approaches have been explored since the early days of computer science (e.g. (Box 1957)). Evolutionary programming as a method for finite-state machine evolution has been developed by Folgel et al. (1966). Genetic algorithms have been introduced by Holland (1962) and an overview of the subject can be found in (Forrest 1993) and (Mitchell 1996). GA are especially suited to tasks in which hypotheses are complex (e.g. sets of rules for robot control, sets of optimal routes, etc.) and in which the objective to be optimised may be an indirect function of the hypotheses (Goldberg 1994). A variant of GA is *genetic programming*, in which the hypotheses being manipulated are computer programs rather than bit strings[2]. Genetic programming has been demonstrated to learn programs for tasks such as simulated robot control (Koza 1992) and recognizing objects in visual scenes (Teller and Veloso 1994).

Reinforcement learning

Reinforcement learning addresses the question of how an autonomous agent (see section 3.7), which senses and acts in its environment, can learn to choose optimal actions to accomplish its goals. This generic problem covers learning tasks such as to control CAM tools and robots, to optimise operations in factories, to search Internet, to play board games, etc. In a nutshell, reinforcement learning is reward hunting. Namely, each time a given agent performs an action in its environment, a trainer may provide a reward or penalty to indicate the desirability of the resulting state; the goal of the agent is to learn an action policy that maximises the total reward it will receive from any starting state. The reinforcement learning methodology fits a problem setting known as a Markov decision process, in which the outcome of applying any action to any state depends only on this action and this state as opposed to being dependent on preceding actions or states. A prototypical reinforcement learning algorithm is Q-learning, in which the agent learns the evaluation function ***Q(s, a)*** representing the maximum expected, cumulative reward the agent can achieve by applying action ***a*** to state ***s***. Watkins (1989) introduced Q learning to acquire optimal policies when the reward and action transition functions are unknown. Some of the earliest work on reinforcement learning can be found in (Samuel 1959). Recent surveys are given by Kaelbling et al. (1996) and Sutton and Barto (1998).

Vantages and disadvantages of machine learning

The major vantage of a learning system is its ability to adapt to a changing environment. Of course, the existing machine-learning techniques are still far from enabling computers to learn nearly as well as people. Yet algorithms have been invented that are effective for certain types of learning tasks. In the late 90s, a formalised theoretical foundation of learning wasn established (Mitchell 1997) and many practical computer programs have been developed to enable different types of learning. Machine learning algorithms have proven to be of great practical value, especially in:

- Data mining problems concerning large databases that may contain valuable implicit regularities that can be discovered automatically (for an overview of this topic, the reader is referred to the special issue on Intelligent Information

[2] Though hypotheses may be represented by symbolic expressions or even computer programs, they are usually described by bit strings. The interpretation of these bit strings depends on the actual application.

Retrieval of the IEEE Intelligent Systems and Their Applications, vol. 14, no. 4, pp. 30-70).

- Poorly understood domains where humans might not have well-established, generic knowledge needed to develop effective algorithms (e.g. in learning to play games (Furnkranz 2001) or in learning to interpret human facial affect (chapter 6)).
- Domains where the program must dynamically adapt to changing conditions (e.g. see the special issue on Self-adaptive Software of the IEEE Intelligent Systems and Their Applications, vol. 14, no. 3, pp. 26-63).

However, most of the machine-learning algorithms require a special training phase whenever information is extracted (knowledge generalisation), which makes on-line adaptation (sustained learning) difficult (Aamodt 1991). Virtually all techniques discussed in this section (except instance-based learning) are not well suited for on-line learning. Hence, learning in dynamic environments is cumbersome (if possible at all) for most machine-learning methods. Another common problem is that, in general, machine-learning techniques are data oriented: they model the relationships contained in the training data set. In turn, if the employed training data set is not a representative selection from the problem domain, the resulting model may differ from actual problem domain. This limitation of machine learning methods is aided and abetted by the fact that most of them do not allow the use of a priori knowledge. Finally, machine-learning algorithms have difficulties in handling noise. Though many of them have some special provisions to prevent noise fitting, these may have a side effect of ignoring seldom occurring but possibly important features of the problem domain.

3.6 Case-based reasoning

During the 70s and 80s, one of the most visible developments in AI research was the emergence of rule-based expert systems (RBES). These programs were applied to more and more problem domains requiring extensive knowledge for very specific and rather critical tasks including hardware troubleshooting, geological exploration and medical diagnosis (section 3.2). In general, the RBES should be based upon a deep, explicit, causal model of the problem domain knowledge that enables them to reason using first principles. But whether the knowledge is shallow or deep, an explicit model of the domain must still be elicited and implemented. Hence, despite their success in many sectors, developers of RBES have met several critical problems (Schank 1984):

1. Difficult and time-consuming construction of the intended knowledge base due to complex and time-consuming expert knowledge elicitation. This is especially the case with problem domains covering a broad range of knowledge.
2. Incapability of dealing with problems that are not explicitly covered by the utilised rule base. In general, rule-based expert systems are useful if the built-in knowledge is well formalised, circumscribed, established and stable.
3. If no learning facility is built into a rule-based expert system, any addition to the existing program requires a programmer intervention.

Solutions to these problems have been sought through better elicitation techniques and tools (Brooke and Jackson 1991), improved development paradigms, knowledge modelling languages and ontologies (Wielinga et al. 1992), and advanced techniques and tools for maintaining systems (Watson et al. 1992). However, in the past decade an alternative reasoning paradigm and computational problem-solving method attracted a great deal of attention: *Case-Based Reasoning* (CBR) solves new problems by adapting previously successful solutions to similar problems. CBR draws attention because it seems to address the problems outlined above directly (Watson and Marir 1994):

- CBR does not require an explicit domain model and so elicitation becomes a task of gathering case histories.
- Implementation is reduced to identifying significant features that describe a case, an easier task than creating an explicit model.
- CBR systems can learn by acquiring new knowledge as cases. This and the application of database techniques makes the maintenance of large volumes of information easier.

The work of Roger Schank (1982, 1984) is widely held to be the origin of CBR. He proposed a different view on model-based reasoning inspired by human reasoning and memory organisation. Schank suggests that our knowledge about the world is mainly organised as memory packets holding together particular episodes from our lives that were significant enough to remember. These *memory organisation packets* (MOPs) and their elements are not isolated but interconnected by our expectations as to the normal progress of events (called *scripts* by Schank). In turn, there is a hierarchy of MOPs in which "big" MOPs share "small" MOPs. If a MOP contains a situation where some problem was successfully solved and the person finds himself in a similar situation, the previous experience is recollected and the person can try to follow the same steps in order to reach a solution. Thus, rather than following a general set of rules, reapplying previously successful solution schemes in a new but similar context solves the newly encountered problems. Using these observations about human reasoning process Schank (1984) proposed *memory-based expert systems*, which are characterised as follows:

- The utilised knowledge base is derived primarily from enumeration of specific cases or experiences. This is founded upon the observation that human experts are much more capable of recalling experiences than of articulating internal rules.
- As problems are presented to a memory-based expert system to which no specific case or rule can match exactly, the system can reason from more general similarities to come up with an answer. This is founded upon the generalisation power of human reasoning. In general, we are reminded of something by the similarity, but the retrieval can be also based on differences. Furthermore, the retrieval is almost never full breadth and is highly context dependent. The reason for not performing an exhaustive recall is not only due to the cumbersomeness of such a task but also due to the organisations of MOPs: once we focus on some MOP it is very easy to recall other MOPs related to it by some features.
- The memory of experiences utilised by the system is changed and augmented by each additional case that is presented. A cornerstone of the memory-based model of reasoning is automatic learning: the system should remember the problems that it has encountered and use that information to solve future problems. This is founded upon the capability of the human brain to merge the progress of events seamlessly into the previously developed scripts of events.

The area of AI concerned with case-based reasoning puts Schank's memory-based reasoning model in practice. In a nutshell, CBR is reasoning by remembering: previously solved problems (cases) are used to suggest solutions for novel but similar problems. Kolodner (1996) lists four assumptions about the world around us that represent the basis of the CBR approach:

1. *Regularity*: the same actions executed under the same conditions will tend to have the same or similar outcomes.
2. *Typicality*: experiences tend to repeat themselves.
3. *Consistency*: small changes in the situation require merely small changes in the interpretation and in the solution.
4. *Adaptability*: when things repeat, the differences tend to be small, and the small differences are easy to compensate for.

Figure 3.12 (Leake 1996) illustrates how the assumptions listed above are used to solve problems in CBR. Once the currently encountered problem is described in terms of previously solved problems, the most similar solved problem can be found. The solution to this problem might be directly applicable to the current problem but, usually, some adaptation is required. The adaptation will be based upon the differences between the current problem and the problem that served to retrieve the solution. Once the solution to the new problem has been verified as correct, a link between it and the description of the problem will be created and this additional problem-solution pair (case) will be used to solve new problems in the future.

Adding of new cases will improve results of a CBR system by filling the problem space more densely.

The CBR working cycle

In the problem solving illustrated in Figure 3.12 and explained above, the following steps were taken: describing the current problem, searching for a similar previously solved problem, retrieving the solution to it, adapting the solution to the current problem, verifying the solution, and storing the newly solved problem. In turn, since the newly found solution may be used for solving future problems, the process illustrated in Figure 3.12 denotes, in fact, the CBR working cycle.

According to Kolodner (1993), the CBR working cycle can be described best in terms of four processing stages:

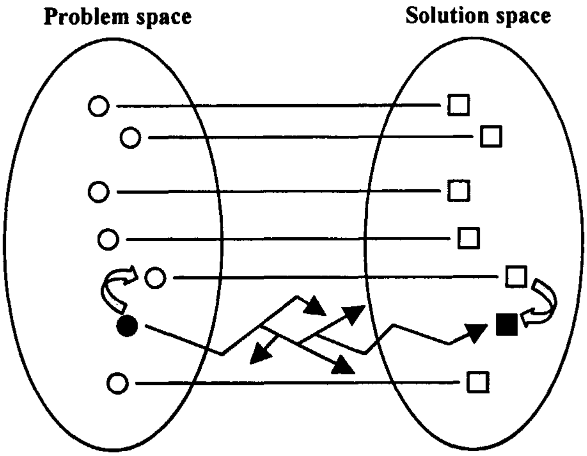


Figure 3.12: Problem solving using CBR

1. *Case retrieval*: after the problem situation has been assessed, the best matching case is searched in the case base and an approximate solution is retrieved.
2. *Case adaptation*: the retrieved solution is adapted to fit better the new problem.
3. *Solution evaluation*: the adapted solution can be evaluated either *before* the solution is applied to the problem or *after* the solution has been applied. In any case, if the accomplished result is not satisfactory, the retrieved solution must be adapted again or more cases should be retrieved.
4. *Case-base updating*: If the solution was verified as correct, the new case may be added to the case base.

Aamodt and Plaza (1994) give a slightly different scheme of the CBR working cycle comprising the *four REs* (Figure 3.13):

1. RETRIEVE the most similar case(s);
2. REUSE the case(s) to attempt to solve the current problem;
3. REVISE the proposed solution if necessary;
4. RETAIN the new solution as a part of a new case.

Although they use different terminologies, the CBR working cycles denoted above are essentially the same. A new problem is matched against the cases furnishing the case base and one or more similar cases are *retrieved*. A solution suggested by the matching cases is then *reused*. Unless the retrieved case is a close match, the solution will probably have to be *revised* (*adapted*) and tested (*evaluated*) for success, producing a new case that can be *retained* ensuing, consequently, *update of the case base*.

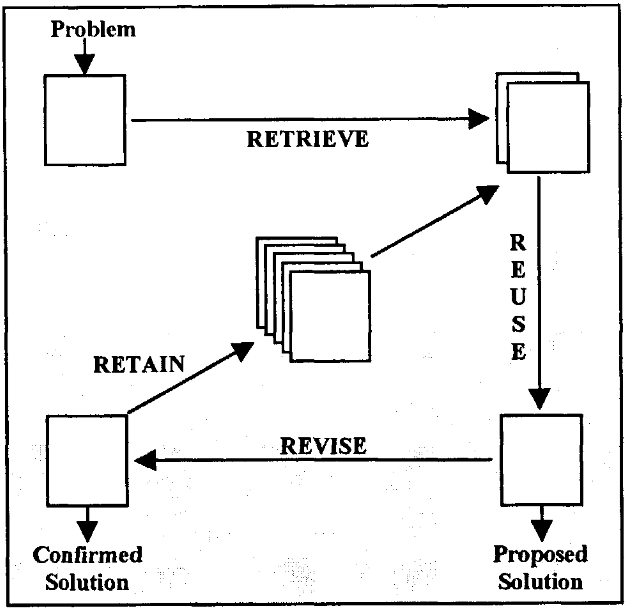


Figure 3.13: The CBR cycle

Types of knowledge in CBR

CBR systems make use of many types of knowledge about the problem domain for which they are designed. Richter (1995) identifies four knowledge containers: the vocabulary, similarity measures, adaptation knowledge, and cases themselves. The first three containers usually represent general knowledge about the problem domain. If there are any exceptions from this knowledge, they are commonly handled by appropriate cases.

Vocabulary includes the knowledge necessary for choosing the features utilised to describe the cases. Case features have to be specified so that they satisfy both: (i) being helpful in retrieving other cases, which contain useful solutions to similar problems, and (ii) being discriminative enough to prevent retrieval of too different cases, which could lead to falsê solutions and/or reduced performance. A thorough comprehension of the problem domain is necessary to be able to choose which of all problem parameters are the best as case features. In addition, either the vocabulary should be chosen such that it anticipates future expansion of the case base, or the system should be developed such that it alleviates automatic expansion of the vocabulary. Otherwise, it may be impossible to represent new problem features, which will then either be mapped to the available descriptors or be ignored, probably leading in both cases to wrong solutions.

Similarity measures include the knowledge about the similarity measure itself and the knowledge used to choose the most efficient organisation of the employed case base and the most appropriate case-retrieval method. For any given problem, there are many possible similarity measures that can be used. Hence, choosing the best among the available possibilities and implementing the chosen similarity measure efficiently exacts sound knowledge of the problem domain. This is especially important for classification problems involving complex structured cases since the value of the similarity can be used as a basis for automatic classification. As far as the organisation of the employed case base and the retrieval algorithm are concerned, a balance has to be found between case-memory models that preserve the semantic richness of cases and methods that simplify the access and retrieval of relevant cases. In general, knowledge about cases can be used to choose the organisational structure of the case base such that the cases can be accurately and efficiently retrieved.

Adaptation knowledge includes the knowledge necessary for implementing the adaptation and evaluation stages of the CBR working cycle. Generally, the adaptation stage requires knowledge about how differences in problems affect the solutions. This knowledge is usually coded in explicit rules. Yet, since for many problem domains, this is the most difficult knowledge to acquire, the adaptation is frequently left to the user of the system. This is especially the case when mistakes made by the system are expensive effecting the reliability of the system and, in turn, the user's confidence in it (Mark et al. 1996). Usually, before applying a new solution for solving a problem, its correctness has to be evaluated. The knowledge required for the evaluation stage concerns estimating the significance of differences and similarities between the situations. Thus, this type of knowledge can be viewed as an extension and refinement of the knowledge furnishing the similarity measures container.

Cases contain knowledge about solved problem instances and, in many CBR systems, this represents the knowledge that the system acquires during use. What the cases will contain is mainly determined by the chosen vocabulary. Sometimes the employed case base is initialised with carefully selected cases that provide a problem domain coverage that is as even as possible (e.g. this is the case with the Facial Expression Classifier part of ISFER, section 6.4). This is commonly the case when the necessary adaptation stage is to be kept simple, yielding manageable system maintenance. Anyhow, new cases will usually be added during use. Yet, it is often unwise to store all the solved problems as cases. Large case bases may have high memory/storage requirements, may impose long retrieval times and, in turn, may reduce the system's performance. Therefore, heuristics should be specified for determining the useful cases to be stored in the case base.

Case representation

A case is a contextualised piece of knowledge representing an experience. It contains the past lesson that is the content of the case and the context in which the lesson can be used (Kolodner 1993). In general, a case comprises a:

- *Problem description*, which depicts the state of the world when the case occurred;
- *Problem solution* which states the derived solution to that problem; and/or
- *Outcome*, which describes the state of the world after the case occurred.

Cases that comprise problems and their solutions can be used to derive solutions to new problems, whereas cases comprising problems and outcomes can be used to evaluate new situations. If such cases contain solutions in addition, they can be used to evaluate the outcome of proposed solutions and prevent potential problems (e.g. in MEDIATOR (Simpson 1985)). The more information is stored, the more useful the case can be. Yet entering all available information makes the system more complex and, in turn, more difficult to use. Due to these reasons, most of the CBR systems are limited to storing only problem descriptions and solutions.

The problem description essentially contains as much data about the problem and its context as necessary for an efficient and accurate case retrieval. Principally, it is useful to store retrieval statistics like the number of times the case was retrieved and the average match value. These statistics may be valuable for handling the case base: for prioritising cases, for pruning the case base by removing seldom used cases, and generally for maintenance of the case base.

The problem solution can be either atomic or compound. Atomic solutions are typical for CBR systems used for diagnosis or for classification in general. Compound solutions can be found for instance in CBR systems utilised for planning or design. A compound solution may be composed of a sequence of actions, an arrangement of components, etc. In the case of the Facial Expression Classifier part of ISFER, compound solutions consist of multiple, user-defined, facial-expression interpretation labels (chapter 6). The main use of a solution is to serve as a starting point for educing new solutions. Therefore, the way a solution is derived may be of equal importance as that of the solution itself.

Cases can be represented as simple feature vectors, or they can be represented using any AI representational formalism such as frames, objects, predicates, semantic nets, or rules. The choice of particular representational formalism is largely determined by the information to be stored within a case. Cases can be monolithic or compound. Individual parts of compound cases can be processed or used separately. For example, a problem can be solved by reusing partial solutions from several compound cases, like within the Facial Expression Classifier part of ISFER (chapter 6). Most representational formalisms are proprietary for the more complex cases. Nevertheless, there is a lack of consensus within the CBR community as to exactly what information should be stored within a case and, in turn, which representational formalism should be used. However, two pragmatic measures can be taken into account in deciding both the information to be stored in a case and the appropriate representational formalism: the intended functionality and the ease of acquisition of the information represented in the case (Kolodner 1993).

Finally, cases are the basis of any CBR system: a system without cases would not be a case-based system. Yet, a system using only cases and no other explicit knowledge (not even in the similarity measures) is difficult to distinguish from a nearest-neighbour classifier or a database retrieval system. In other words, such a system does not exploit the full generalisation power of CBR, resulting usually in poor system performance due to inefficient retrieval based upon case-by-case search of the whole case base.

Indexing

Within the CBR community, an explicit formal specification (i.e. ontology) of what the terms “indices” and “indexing” actually mean in terms of a CBR system has not been established yet. Kolodner (1996) identifies indexing with an accessibility problem, that is, with the whole set of issues inherent in setting up the case base and its retrieval process so that the right cases are retrieved at the right time. Thusly, case indexing involves assigning indices to cases to facilitate their retrieval. CBR researches proposed several guidelines on indexing (Watson and Marir 1994). Indexes should be:

- *predictive* of the case relevance;
- *recognisable* in the sense that it should be understandable why they are used;
- *abstract* enough to allow for widening the future use of the case base;
- *concrete* (discriminative) enough to facilitate efficient and accurate retrieval.

Both manual and automated methods are used nowadays to select indices. Choosing indices manually involves deciding the purpose of a case with respect to the aims of the user and deciding under which circumstances the case will be useful. Kolodner (1993) claims that people tend to be better at choosing the indices than automatic algorithms. Anyhow, there is an ever increasing number of automated indexing methods. For a review of these the reader is referred to (Watson and Marir 1994). For an example of an automatic indexing algorithm performing indexing cases by (case) features that tend to be predictive across the entire problem domain, the reader is referred to sections 6.4 and 6.5, which describe the processing of the Facial Expression Classifier part of ISFER.

Indices do not have to be rigid; they may change during use of the system. In fact, changing the indexes is one way of learning. Changes may be made if, for instance, a wrong case was retrieved or an entirely novel problem situation is encountered. Changes may involve changing weights (importance/priority) of the features, changing or adding features, changing or adding pointers to other cases in the case base, etc. Similarly to selecting/generating the indexes, changing the indexes can be done either manually or automatically.

Case-base organisation

Case storage is an important aspect in designing efficient CBR system, in that it should reflect the conceptual view of what is represented in the case and take into account the indexes that characterise the case. As already mentioned above, the case base should be organised into a manageable structure that supports efficient and accurate search and retrieval methods. Accurate retrieval guarantees that the best matching case will be retrieved, and efficient retrieval guarantees that cases will be retrieved fast enough for acceptable system response times. These two factors are inversely proportional: it is easy to guarantee accurate retrieval at the expense of efficiency (e.g. by matching all the cases) and easy to have fast retrieval if only a fraction of the employed case base is searched (possibly missing some examples). Hence, a good case-base organisation and a good retrieval algorithm are the ones which yield the best compromise between accuracy and efficiency of the retrieval algorithm.

In general, three main approaches to case-base organisation can be distinguished: flat organisation, clustered organisation, and hierarchical organisation. Also a combination of these methods within the same case base is possible (e.g. in Ultrasonic Rail-Inspection System proposed in (Jarmulak, 1999)).

Flat organisation is the simplest case-base organisation that yields a straightforward flat structure of the case base. Though advantageous due to its simplicity and facile case addition/deletion, a flat case-base organisation imposes, in general, case retrieval based upon a case-by-case search of the whole case base. Hence, for medium and large case bases, this leads to time-consuming retrieval, yielding an inefficient CBR system.

Clustered organisation, originating in the dynamic memory model initially proposed by Schank (1982) and refined by Kolodner (1983), is the type of case-base organisation in which cases are stored in clusters of similar cases. The grouping of cases may be based on their mutual similarity (like in the case of the dynamic memory of experiences used by the Facial Expression Classifier part of ISFER, section 6.4) or on the similarity to some prototypical cases (as proposed in (Malek et al. 1998) or in (Schmidt and Gierl 1998)). The advantage of this organisation is that the selection of the clusters to be matched is rather easy, as it is based upon the indexes and/or prototypical cases characterising the clusters. A disadvantage is that it needs a more complex algorithm for case addition/ deletion than a flat organised case base.

Hierarchical organisation, originating in the category-exemplar memory model of Porter and Bareiss (1986), is the case-base organisation that is generally obtained when cases that share the same features are grouped together. The case memory is a network structure of categories, semantic relations, cases, and index pointers. Each case is associated with a category, while the categories are inter-linked within a semantic network containing the features and intermediate states referred to by other terms. Different case features are assigned different importance in describing the membership of a case to a category. It is important to note that this importance assignment is static; if it changes, the case-base hierarchy has to be redefined. A new case is stored by searching for a matching case and by establishing the relevant feature indexes. If a case is found with only minor differences to the new case, the new case is usually not retained. In turn, a hierarchical case-base organisation facilitates fast and accurate case retrieval. However, its higher complexity implies a rather cumbersome case addition/deletion, potentially involving expensive case-base reorganisation and an inapt case-base evaluation and maintenance.

Retrieval

Given a description of a problem, a retrieval algorithm should retrieve cases that are most similar to the problem or situation currently presented to the pertinent CBR system. The retrieval algorithm relies on the indices and the organisation of the case memory to direct the search to case(s) potentially useful for solving the currently encountered problem.

The issue of choosing the best matching cases can be referred to as *analogy drawing* (Falkenehainer 1988), that is, comparing cases in order to determine the degree of similarity between them. Many retrieval algorithms have been proposed in the literature up to date: induction search (e.g. ID3, Quinlan 1979), nearest neighbour search (e.g. Kolodner 1993, Owens 1993), serial search (e.g. Navinchandra 1991), hierarchical search (e.g. Maher and Zhang 1993), parallel search (e.g. Andersen et al. 1994), etc.

The simplest form of retrieval is the 1st-nearest-neighbour search of the case base, which performs similarity matching on all the cases in the case base and returns just one best match (Mitchell 1997). It is to be expected that this method implies long retrieval times, especially in the case of large case bases. Therefore, cases are usually *preselected* prior to similarity matching. Cases can be preselected using a simpler similarity measure; commonly, this is done using the indexing structure of the case base. A typical problem with preselection concerns handling a situation where no best match has been found in the preselected set of cases; since preselection is merely approximate, there is a possibility that amongst the non-selected cases a better match can be found.

Another way of speeding up the retrieval is to employ *ranking of cases*. The simplest ranking method concerns exploiting the retrieval statistics for cases in the case base. The frequently retrieved cases can be considered as prototypical cases and probably should be matched first. Another ranking method is applicable to the clustered case-base organisation. It concerns matching the current case to the clusters' prototypes and then searching the matching clusters in the order determined by the degree of similarity between the matching clusters' prototypes and the current case.

The retrieval may result in retrieving single or *multiple best match cases*. In general, the retrieval mechanism tends to be simpler and faster if: (i) a larger number of possibly similar cases are retrieved, (ii) all of them are used to find solutions, and then (iii) the best solution is chosen. In this case, the retrieval algorithm itself may be less selective (and, therefore, simpler and faster) since the usefulness of the retrieved cases is to be determined in succeeding processing phases.

Finally, a way of speeding up the retrieval is to do it in parallel. A *parallel search* of the case base is realisable since case matching does not require exchange of much information between the parallel running processes. Thus, the speed gain scales up with the number of processing units. While the implementation of parallel retrieval is simple for flat and clustered case bases, it is rather difficult for hierarchical case bases (Jarmulak 1999). Though bringing significant speed gains, parallel retrieval is usually accompanied by an increase in implementation costs and software complexity.

Adaptation

Generally, once a matching case is retrieved, it will not correspond to exactly the same problem as the problem for which the solution is currently being sought. Consequently, the solution belonging to the retrieved case may not be optimal for the problem presently encountered and, therefore, it should be adapted. Adaptation looks for prominent differences between the retrieved case and the current case, and then (most commonly) applies a formulae or a set of rules to account for those differences when suggesting a solution. In general, there are two kinds of adaptation in CBR (Watson and Marir 1994):

1. *Structural adaptation* applies adaptation rules directly to the solution stored in cases (Kolodner 1993). If the solution comprises a single value or a collection of independent values, structural adaptation can include modifying certain parameters in the appropriate direction, interpolating between several retrieved cases, voting, etc. However, if there are interdependencies between the components of the solution, structural adaptation requires a thorough comprehension and a well-defined model of the problem domain.
2. *Derivational adaptation* reuses algorithms, methods, or rules that generated the original solution to produce a new solution to the problem currently presented to the system. Hence, derivational adaptation requires the planning sequence that begot a solution to be stored in memory along with that solution. This kind of adaptation, sometimes referred to as *reinstantiation*, can only be used for problem domains that are well understood.

An ideal set of rules must be able to generate complete solutions from scratch, and an effective and efficient CBR system may need both structural adaptation rules to adapt poorly understood solutions and derivational mechanisms to adapt solutions of cases that are well understood. However, one should be aware that complex adaptation procedures make the system more complex but not necessarily more powerful. Complex adaptation procedures make it more difficult to build and maintain CBR systems and may also reduce system reliability and, in turn, user's confidence in the system if faulty adaptations are encountered due to, for example, incompleteness of the adaptation knowledge, which is the most difficult kind of knowledge to acquire (Mark et al. 1996). Therefore, in many CBR systems, adaptation is done by the user rather than by the system. Mark et al. (1996) report that in a well-designed system, the users do not perceive "manual" adaptation as something negative.

Vantages and limitations of CBR

CBR is a lazy problem-solving method and shares many characteristics with other lazy problem-solving methods, including advantages and disadvantages. Aha (1998) defines the peculiarities of lazy problem-solving methods in terms of three Ds:

- *Defer*: lazy problem solvers simply store the presented data and generalizing beyond these data is postponed until an explicit request is made.
- *Data-driven*: lazy problem solvers respond to a given request by combining information from the stored data.
- *Discard*: lazy problem solvers dismiss any temporary (intermediate) result obtained during the problem solving process.

Unlike lazy problem solvers, eager problem-solving methods try to extract as much information as possible from the presented data and then to discard the data prior to the actual problem solving. An example of a lazy problem solver is a CBR classifier, while an ANN classifier is an example of an eager problem solver. Eager algorithms can be referred to as knowledge compilers, as opposed to lazy algorithms, which perform run-time knowledge interpretation. This is the key difference between lazy and eager problem solvers, which can also be explained by the following:

- Lazy methods can consider the current query instance x when deciding how to generalise beyond the training data (which have already been presented).
- Eager methods cannot, because their global approximation to the target function has already been choosen by the time they observe the current query instance x.

To summarise, lazy methods have the option of selecting a different hypothesis or local approximation to the target function for each presented query instance. Eager methods using the same hypothesis space are more restricted because they must choose their approximation before the presented queries are observed.

Consequently, a lazy method will generally require less computation during training, but more computation when it must generalise from training data by choosing a hypothesis based on the training examples near the currently presented query.

The benefits of CBR as a lazy problem-solving method are:

- *Ease of knowledge elicitation*: Lazy methods, in general, can utilise easily available cases or problem instances instead of rules that are difficult to extract. So, classical knowledge engineering is replaced by case acquisition and structuring (Aha 1998).
- *Absence of problem-solving bias*: Because cases are stored in a "raw" form, they can be used for multiple problem-solving purposes. This in contrast to eager methods, which can be used merely for the purpose for which the knowledge has already been compiled.
- *Incremental learning*: A CBR system can be put into operation with a minimal set of solved cases furnishing the case base. The case base will be filled with new cases as the system is used, increasing the system's problem-solving ability. Besides simple augmentation of the case base, new indexes and clusters/categories can be created and the existing ones can be changed. This in contrast to virtually all machine-learning methods (section 3.5), which require a special training period whenever information extraction (knowledge generalisation) is performed. Hence, dynamic on-line adaptation to a non-rigid environment is possible (Aha 1991, Mitchell 1997).
- *Suitability for complex and not-fully formalised solution spaces*: CBR systems can be applied to an incomplete model of problem domain; implementation involves both to identify relevant case features and to furnish, possibly a partial case base, with proper cases. In general, because they can handle them more easily, lazy approaches are often more appropriate for complex solution spaces than eager approaches, which replace the presented data with abstractions obtained by generalisation.
- *Suitability for sequential problem solving*: Sequential tasks, like these encountered in reinforcement learning problems, benefit from the storage of history in the form of a sequence of states or procedures. Such a storage is facilitated by lazy approaches.
- *Ease of explanation*: The results of a CBR system can be justified based upon the similarity of the current problem to the retrieved case(s). Because solutions generated by CBR are easily traceable to precedent cases, it is also easier to analyse failures of the system. As noted by Watson and Marir (1994), the explanations provided based upon individual and generalised cases tend to be more satisfactory than explanations generated by chains of rules.
- *Ease of maintenance*: This is particularly due to the fact that CBR systems can adapt to many changes in the problem domain and the pertinent environment, merely by acquiring new cases. This eliminates some need for maintenance; only the case base(s) needs to be maintained.

Major disadvantages of lazy problem solvers are their memory requirements and time-consuming execution due the processing necessary to answer the queries. The limitations of CBR can be summarised as follows:

- *Handling large case bases*: High memory/storage requirements and time-consuming retrieval accompany CBR systems utilising large case bases. Although the order of both is at most linear with the number of cases, these problems usually lead to increased construction costs and reduced system performance. Yet, these problems are less and less significant as the hardware components become faster and cheaper.
- *Dynamic problem domains*: CBR systems may have difficulties in handling dynamic problem domains, where they may be unable to follow a shift in the way problems are solved, since they are usually strongly biased towards what has already worked. This may result in an outdated case base.
- *Handling noisy data*: Parts of the problem situation may be irrelevant to the problem itself. Unsuccessful assessment of such noise present in a problem situation currently imposed on a CBR system may result in the same problem being unnecessarily stored numerous times in the case base because of the difference due to the noise. In turn this implies inefficient storage and retrieval of cases.
- *Fully automatic operation*: In a typical CBR system, the problem domain is usually not fully covered. Hence, some problem situations can occur for which the system has no solution. In such situations, CBR systems commonly expect input from the user.

CBR application domains

Although CBR is a relatively new AI methodology, numerous successful applications exist in the academic as well as in the commercial domain. Already in 1994, Watson and Marir reported over 100 commercially available CBR applications. The domains of these numerous CBR systems reported in the literature are the following:

- *Interpretation* as a process of evaluating situations/problems in some context (e.g. HYPO for interpretation of patent laws (Ashley 1991), KICS for interpretation of building regulations (Yang and Robertson 1994), LISSA for interpretation of non-destructive test measurements (Jarmulak 1999)).
- *Classification* as a process of explaining a number of encountered symptoms (e.g. CASEY for classification of auditory impairments (Koton 1989), CASCADE for classification of software failures (Simoudis 1992), PAKAR for causal classification of building defects (Watson and Abdulah 1994), ISFER for classification of facial expressions into user-defined interpretation categories (chapter 6)).
- *Design* as a process of satisfying a number of posed constraints (e.g. JULIA for meal planning (Hinrichs 1992), Déjà Vu for control-software production (Smyth

1996), CLAVIER for design of optimal layouts of composite airplane parts (Kolodner 1993, Mark et al. 1996), EADOCS for aircraft panels design (Netten 1997)).

- *Planning* as a process of arranging a sequence of actions in time (e.g. BOLERO for building diagnostic plans for medical patients (Lopez and Plaza 1993), TOTLEC for manufacturing planning (Costas and Kashyap 1993)).
- *Advising* as a process of resolving diagnosed problems (e.g. DECIDER for advising students (Farrel 1987), HOMER – a CAD/CAM help desk (Goker et al. 1998)).

3.7 Distributed AI

During the last decades, computing devices have been used as sophisticated tools, greatly advancing and augmenting human abilities such as memory and calculation as well as publishing and communication capabilities. Within the research areas in information technology and computer science, research in AI has aimed at developing software to emulate some intelligent capabilities of human beings such as reasoning, communication (verbal as well as non-verbal), and learning. By descending further down the tree delimiting conjugate foci of research areas in computer science, we come across distributed artificial intelligence (DAI), which is a sub-field of AI concerned with the investigation of knowledge models and communication and reasoning techniques which so-called *computational agents* might need to participate in "men-machine societies" that are composed of computers and people. In other words, DAI is concerned with situations in which several systems (e.g. persons, computers, sensors, robots, mobile vehicles, etc.) interact in order to solve a common problem; it aims to understand and model actions and knowledge in collaborative enterprises (Gasser 1991). The research in DAI can be divided into two main areas (Moulin and Chaib-Draa 1996): distributed problem solving and multi-agent systems.

Distributed problem solving considers how the task of solving a particular problem can be divided among a number of modules that cooperate in splitting and sharing the available knowledge about the problem and about its evolving solutions. Durfee et al. (1989) remarked that many applications are inherently distributed : some are spatially distributed (e.g. a system for integrating and interpreting data obtained from spatially distributed sensors), and others are functionally distributed (e.g. a group of experts with different specialisations collaborating in solving a complex problem like in ISFER, see section 2.6). For the sake of clarity, it is important to note that DAI does not address the issues related to parallel computer architectures, parallel programming languages, and distributed operating systems designed merely for their efficiency. DAI techniques have been applied to distributed interpretation, distributed planning and control, cooperating expert systems, computer-supported cooperative work, etc. (for a more detailed account of DAI applications, the reader is referred to (van Dyke Parunak 1996, Shoham 1999)).

Research in *multi-agent systems* is concerned with the behaviour of a collection of possibly pre-existing *autonomous agents* which aim at solving a given problem. As defined by Durfee et al. (1989), multi-agent systems are loosely-coupled networks of *problem solving agents* that work together to solve problems that are beyond their individual capabilities.

It is most remarkable that almost all of the definitions given in DAI concern agents and agent technology, but none of these definitions resolve what is meant by the term "agent". As remarked by Shoham (1999) and Hendler (1999), having a discussion about software agents is not easy; there is no clear and comprehensive definition of the notion. At best, what an interested party can get is a clear definition of one person's version of the concept, which is guaranteed to exclude various elements that others will swear are the essence of software agents. At worst, he/she will get an answer so general and imprecise that it has little informational content.

Two factors seem to be responsible for the use of the term agent in such an ambiguous way that no explicit formal specification (i.e. ontology) can be established: over-hype and the confusing amalgamation of quite different ideas and motivations under the agents' umbrella. The hype is inevitable – since 1995, many of the IEEE and ACM journals have devoted special issues to software agents; since 1997, many agents conferences have taken place; nowadays, special panels on agents are even organised during industry symposia; agents that sort e-mails, adaptively recommend Web pages, translate between different knowledge bases, and sometimes have an individual electronic personality are commercially available at present (e.g. Hayes-Roth et al. 1999). Notwithstanding, this hype is unfortunate; due to the popularity of the revolutionary agents, there is a general tendency to label software programs (especially the ones aimed at Internet applications) with agent-oriented terms. In turn, alongside fairly creative ideas, quite shoddy ones also seek legitimacy under the agent's umbrella. As a result, negative reactions might (and probably will) be directed indiscriminately towards all work in the area.

Agent categories

Thus, if there is no clear definition of agents, what can be done to expound what the term stands for? Shoham (1999) proposed to identify dimensions (axes) of software agency, that is, to distinguish the properties that characterise various versions of agents. The idea behind is to enable grouping of software agents that embrace roughly the same properties. Among the numerous properties agents can have, the ones commonly present in the relevant research literature are (Shoham 1999):

- *Ongoing execution*: Unlike software routines that are invoked to achieve particular tasks and then disappear, agents function continuously for a lengthy period of time.

- *Autonomy*: Agents do not require perpetual human control, supervision or feedback.
- *Environmental awareness*: Agents model the environment in which they operate and they sense and react to changes in it.
- *Adaptability*: Besides adapting to the environment in which they operate, agents adapt their behaviour over time to suit the preferences and behaviour of individual users.
- *Intelligence*: Agents exhibit intelligent behaviour facilitated by embodied techniques such as probabilistic reasoning, machine learning, and automated planning.
- *Agent awareness*: Agents may model other agents, reason about them, and interact with them using special communication languages and protocols.
- *Mobility*: Agents can migrate in a network.
- *Anthropomorphism*: Agents may exhibit human-like qualities: respond to queries about their "beliefs" or "obligations", emulate facial and vocal expression responses by displaying animations, etc.

Though the properties listed above expound some global characteristics that agents might have, categorising various agent-related works along these or similar dimensions will not result in a useful scheme. This is because most of the listed properties are vague and because, in the agent-related field, there is work that is so dissimilar to other work in goals and technology that it is misleading to even speak of some common properties. Therefore, it is perhaps more useful to first home in on the primary orientation of the work and then on its functionality in terms of problem-solving capability of the intended agent-based system.

As far as the orientation of agent-related works is considered, three strands can be distinguished (Shoham 1999):

1. *Nouvelle expert systems (NES)*: NES-motivated agent work attempts to create novel software applications, or to enhance the power of existing ones, based upon advances in AI and related fields. Prototypical examples of NES-applications deal with mining and managing vast data available online (e.g. Kushmerick 1999). The techniques in this area are fairly established, especially if we consider the area of machine learning and probabilistic techniques, which has seen significant progress in the past decade. Though important, this area intrigues computer-literate users rather than scientific researchers and is, therefore, not expected to revolutionise computing today (Pentland 2000).
2. *Exotic distributed systems (EDS)*: EDS-motivated work is aimed at: (i) building novel middleware to increase developers' productivity, creating novel infrastructures to advance the local- and/or wide-area networks, and at (ii) generating novel protocols for negotiation over the Internet to leverage electronic commerce. Typical examples of EDS-research objectives aimed at enrichment of the infrastructure are: facilitating remote programming featuring mobile code

and advancing communication languages featuring high-level semantic primitives (e.g. Labrou et al. 1999). The main focus of EDS research aimed at e-commerce is developing novel negotiation mechanisms that are network aware, i.e. that take into account that the Internet tends to change fixed pricing to dynamic pricing, having a profound impact on businesses (e.g. Jamali et al. 1999). The EDS-area of research is rather new, having been explored in the past few years. Due to its recent origin and the potential impact it might have on the way people think about computing and about "doing business", the EDS-area of research has a fair potential of biasing some standard views on the pertinent issues.

3. *Anthropomorphic design (AD)*: AD-inspired work attempts to make computers more accessible to the non-technical user by endowing them with an appearance and behaviour that are, at least superficially human-like. Typical examples of AD applications engender and incorporate animated (virtual) characters within a man-machine interactive environment (e.g. Hayes-Roth et al.1999, Kshirsagar and Thalmann 2000). As remarked by Shoham (1999), Coen (1999), and Pentland (2000), breakthroughs in AD-oriented agents could bring about the most radical change to the computing world. They could change not only how professionals practice computing, but also how mass consumers conceive of and interact with the technology. This research area of future human-computer interfaces is not concerned with menus, mice and keyboards, but with gestures, speech, affect, context and animation. In contrast to animation and natural language understanding, where the technology has advanced to a level of commercial relevance and where many companies are investing significant industrial resources (Thalmann et al. 1998, Juang and Furui 2000), other aspects of AD-type agents, in particular ones that attempt to translate and emulate human behaviour at a deeper level, are less mature and undoubtedly need many improvements of the state of the art (Tekalp 1998, Pentland 2000, Pantic and Rothkrantz 2001a, chapter 8).

Next to being characterised by its primary orientation, an agent-based system can be distinguished by its problem-solving capabilities (behaviour). An agent's problem-solving behaviour can be classified into the following categories (Moulin and Chaib-Draa 1996):

1. *Reactive behaviour*: A reactive agent reacts to changes in its environment or to messages from other agents. It is not able to reason about its intentions, that is, it is not capable of manipulating its goals. Its actions are predefined by some rigid set of rules or stereotype plans and aim, in general, either at updating the agent's fact base or at sending messages to other agents or to the environment. Expert systems, at least in their most common form (section 3.2) and if exhibiting communication capabilities in terms of receiving/sending messages to other agents within the same multi-agent system, are typical examples of reactive

agents (e.g. Facial Action Encoder and Facial Expression Classifier parts of ISFER (sections 5.2 and 6.3)).

2. *Intentional behaviour*: An intentional agent is able to reason on its intentions and beliefs, that is, it is capable of manipulating its goals, of creating new plans of actions, and of executing those plans. Intentional agents may be considered planning systems (Werner 1996): they can select their goals, reason on them by detecting and resolving goal conflicts and coincidences, select or create plans, detect conflicts between plans, execute and, if necessary, revise plans of actions. Intentional agents often resemble classic expert systems in that they encode (by using rule-based approaches now and then) domain-specific information to achieve the intended functionality. The key difference between these agents and the traditional expert systems is that the agent-based approaches generally focus on programs that provide capabilities for the user (e.g. the ability to gather information from the Web or databases, to access and download Web resources, etc.). As noted by Hendler (1999), intentional agents are like expert systems with hands and feet: they exhibit the ability to manipulate the information world on the behalf of the user. Typical examples of intentional agents are Web browsers providing an alert if some conditions hold (e.g. Rosenschein and Krulwich 1999).
3. *Social behaviour*: In addition to intentional agent capabilities, a social agent also includes and manipulates explicit models of other agents: it maintains these models by updating the pertinent goals and plans of actions, it reasons on the knowledge (intentions, expectations, reactions, etc.) incorporated in these models, and it makes its decisions and creates its plans of actions with respect to other agents' models. Social agents may be considered control agents (Hendler 1999): a social agent primarily provides control services to other agents. Such an agent's problem-solving behaviour is not tied to a particular application domain. Rather, a social agent is a program that helps other agents to function together – to find each other, perhaps to control the use of resources, and in any case to coordinate them. Examples of frameworks for agent ensembles can be found in (Jamali et al. 1999) and (Arisha et al. 1999).

Agent design

As already remarked above, DAI literature, with the emphasis on agent-oriented work, is very abundant and covers a variety of topics and experiments. Facing such abundance, a software designer may wonder how to decide if DAI technology is the best choice for the intended application, that is, if it can bring about the best relevant solutions to the recognised problems. This is a principal issue, at least if the designer's preference is effectiveness of the intended software rather than voguish employment of an agent-based approach independently of its applicability to the delimited problem domain. Alas, as noted by Shoham (1999), nowadays many designers prefer fashionable designs over effective ones. Besides, there is another reason for (mis)placing an abundance of dissimilar work under the same agent's umbrella – the lack of established evaluation criteria (which, for some, is precisely the charm). Issues like choosing a DAI-design method relative to the problem domain, designing the relevant test beds, and establishing a firm description of possible applications, have not been addressed in literature in a comprehensive way (Moulin and Chaib-Draa 1996, Shoham 1999).

Agents' vantages and disadvantages

The area of software agents offers exciting research playgrounds and presents attractive commercial opportunities. Lately, anthropomorphic-design-oriented agent work induced an upsurge of interest due to the fact that automating, monitoring, analysing and emulating human communicative behaviour is essential for the design of future human-computer interfaces (Pentland 2000, chapter 8). Similarly, the work that focused on exotic distributed systems opened up novel possibilities for enrichment of the computing infrastructure and devising Internet-based commerce and, in turn, attracted many AI researchers. Finally, the work that focused on nouvelle expert systems (with the emphasis on data retrieval, filtering and mining) already enjoys substantial interest among entrepreneurs and investors, the main reason being the exploding number of information sources available online and the growing number of potential users having online access.

Nevertheless, it should be clear by now that the field of agent technology is really not one field at all. Just as conferences on agents present abundance of papers ranging from Web routers via large-scale scheduling systems to robot programs, the scientific literature contain terminological confusions and a blend of dissimilar issues as well. Hence, agent-oriented work involves a high risk, not because of technical difficulties but because the field lacks standards among the research communities carrying out widely different work.

ISFER as a functionally distributed system

ISFER can be viewed as a multi-agent, functionally distributed system. The key idea behind discussing the architecture of ISFER as a set of agents is that of *task-level decomposition*, defined by Brooks (1991) as follows:

> *A multi-agent system can be viewed as a collection of modules, each of which has its own specific competence, operates autonomously, and is solely responsible for the sensing, modelling, computation and/or reasoning necessary to accomplish its competence. Communication among modules is reduced to a minimum and it is achieved on an information-low level. The global behaviour of such a system is not necessarily a linear composition of the behaviours of its modules, but a more complex behaviour(s) may emerge due to interaction of behaviours generated by the individual modules.*

Each part of ISFER, namely the Facial Data Extractor, the Facial Action Encoder and the Facial Expression Classifier, can be viewed as a module forming an integral part of the system as a whole and having its own competence (i.e. performing a certain task). These tasks involve: spatial sampling of the contours of the prominent facial features in an input facial image (Facial Data Extractor, chapter 4), analysis of facial expressions such that it is applicable to automated FACS coding and based on the data obtained from the Facial Data Extractor part of the system (Facial Action Encoder, chapter 5), and classification of facial expressions into multiple, quantified, user-defined interpretation labels based on the data obtained from the Facial Action Encoder part of the system (Facial Expression Classifier, chapter 6). In turn, it can be said that the communication among modules is reduced to a minimum and that a global behaviour of ISFER is a linear composition of the behaviours of its modules. The three parts of ISFER (Figure 2.25) can be further characterised as being:

- *Problem-solving* modules, where the problem domain of each is restricted by the specific task the relevant module performs. Further, the modules have a "medium" degree of heterogeneity since they differ in utilised problem-solving methods and expertise but have the same computational resource (the current input facial image).
- *Autonomous*, since they do not require the user to be in control at all times (for a further discussion on the issue, the reader is referred to sections 4.2, 5.2 and 6.3).
- *Environmentally unaware*, since they have no knowledge about the environment in which they operate and cannot react to changes in it (for a detailed discussion about context-dependent facial expression analysis, the reader is referred to chapter 8). Though ISFER is, thus, not adaptable to its environment, it is adaptable to its current user. Due to the Facial Expression Classifier part of ISFER, the system *adapts* its behaviour (i.e. the facial expression interpretation it achieves) over time to suit the preferences of the current user (chapter 6).
- *Intelligent* in the sense that they embody sophisticated techniques based on ANN-based reasoning (section 4.3), rule-based and probabilistic reasoning (chapter 5), and machine learning (chapter 6).
- *Stationary* as opposed to mobile since ISFER is a functionally distributed application as opposed to a spatially distributed application.

Thus, ISFER might be viewed as a multi-agent system. However, note that this is just one way of viewing the architecture of ISFER. As explained by Moulin and Chaib-Draa (1996), any expert system can be seen as an agent, at least as a reactive agent (as is the case with the Facial Action Encoder and the Facial Expression Classifier parts of ISFER) if not as an intentional or a social agent. The reason to discuss ISFER as a multi-agent system is not the catalyst behind the recent "agents hoopla" – the accelerating spread of the Internet – which has given rise to an explosion of applications involving intentional and social agents that can manipulate their goals and create new ones in order to search the Internet successfully. The reason lies in the fact that the current functionality of ISFER can easily be enhanced for the monitoring of a particular information source (e.g. a certain participant in a video conference, a certain patient in a group therapy, a certain student attending a course) and providing an alert if some set of conditions holds (e.g. if a certain attitudinal state like frustration or inattention is observed). By this, ISFER would represent a *consumer-based problem-solving agent* (Hendler 1999), which would be able to manipulate the incoming information on the behalf of the current user. By allowing the user to define his current interest while monitoring facial expressions, the commercial potential of ISFER would be increased – ISFER would embody an application-independent automatic tool for facial expression analysis (see also chapters 5, 6 and 8).

Currently, however, none of the three parts of ISFER is able to manipulate its goals and create new ones (e.g. according to the wishes of the user); they all achieve a predefined set of goals by selecting from the action space predefined plans that can be used to reach these goals. In other words, each part of ISFER is a reactive agent. Hence, for the sake of clarity and precision, in the remainder of this thesis the Facial Data Extractor is discussed as a framework for hybrid data extraction from an input static facial image (chapter 4), the Facial Action Encoder is discussed as a rule-based expert system that reasons on facial actions and their intensity based upon the data generated by the Facial Data Extractor (chapter 5), and the Facial Expression Classifier is discussed as a memory-based expert system that performs case-based reasoning about facial expression classification in terms of the interpretation categories defined by the current user and based upon the data from the Facial Action Encoder (chapter 6).

3.8 AI application development

The whole process of developing and maintaining a software product is called the *software life cycle* (Boullart 1992). The software life cycle is structured in several phases. All the activities of those phases that are required to define, develop, test, deliver, operate and maintain a software product form a so-called *software life cycle model*. Different models and variations of those models emphasise different aspects of software life cycle and are appropriate for a range of situations. There are three commonly used software life cycle models: Waterfall, Prototyping, and Incremental development.

Waterfall is the paradigm of the conventional software-engineering model. This is a highly structured phased model. At each phase, it must be verified that the application is built correctly, that it meets the specifications developed in the previous phases and satisfies all the requirements. Waterfall is a sequential model; revisiting a previous stage indicates a bad design. When developing an AI system like an expert system, the expert (which is often also the future user) is usually closely involved in all stages of the development. Such constant involvement results in a dynamic change of specifications and requirements, in contrast to conventional software engineering where requirements and specifications are essentially static. Hence, as a conventional engineering model, the Waterfall model is not the most suitable for the development of an AI system; because of a dynamic change of specifications, the developers will often be required to revisit prior phase, which is not intended if one uses the Waterfall model.

In the case of *prototyping*, multiple iterations have to be performed through the phases of knowledge acquisition, coding and testing until a final prototype of the intended AI system is obtained. This method has two major drawbacks. It might give a wrong impression, based merely on early prototypes, that the effort to complete the whole system will scale linearly from the effort to encode the partial prototypes. In addition, with the complexity of the system to be built, the time and costs per prototype will grow almost exponentially. Consequently, it can be concluded that prototyping can be used as a life cycle model when the problem is sufficiently small. Usually, it is used as a part of a more complex model. For example, Boehm (1988) proposes a spiral life cycle model where subsequent prototypes are developed until the full system is defined by an operational prototype and the Waterfall model is used thereafter for detailed design, coding, testing, etc. of the final system.

Incremental development represents a refinement of the waterfall model as the prototyping is integrated. In this model the product development phases of the study of feasibility, requirements analysis, and global product design, are the same as in the Waterfall model, but the phases concerning the detailed design, coding, integration, and implementation are split in successive increments of functionality. Each increment refines the functional design of the future AI system and adds new functionality to it. An important characteristic of the incremental development model is that it allows the incorporation of new user requirements as the project develops. Thus, it facilitates dynamic development of system specifications. Another crucial property of the incremental development model is related to:

1. *Breadth*: during the whole process the system as a whole has to be developed and each increment must satisfy all system specifications that are already defined.
2. *Depth*: each increment has to be deep enough so that it clearly enhances the functional design. It must be kept in mind that none of increments may exceed some "reasonable" depth. If developers focus too much on a certain part of knowledge, they will generate deep and robust solutions for that part, but this does not guarantee a deep and robust general solution.

In general, independently of the chosen development model, the development of an AI application may be split into the following phases (Boullart 1992):

1. assessment and scoping,

2. system specification,
3. refinement and implementation (coding),
4. final system integration, testing, and transfer, and
5. maintenance, enhancement and support.

As already noted in section 3.1, the primary actions of the assessment and scoping development phase are:

- appointing a so-called target group (experts, users, management) and analysing the user requirements,
- estimating the potential for success of the future expert system (i.e. performing the study of feasibility), and
- selecting the development team (i.e. project leader, knowledge engineer(s), domain expert(s), software engineer(s) and programmers, external consultants).

In small projects, the knowledge and the software engineer may be the same person, but in medium and big projects there will be enough work for two or more specialists. External consultants could be incorporated in a project to provide help in, for example, hardware or specialised topics of AI. They can also be of great help as third (objective) party by performing the validation tests. Although these considerations are not of the least importance, the crucial issue in an AI system development is to ensure both:

- that the AI paradigm is the best choice for the development of the intended application, and
- that there is a real demand for the future AI system (see section 3.1).

The actions belonging to the system specification development phase are:

- making a conceptual model of the future AI system,
- making a design model of the future system,
- selecting the appropriate AI techniques to be employed and the appropriate software tool(s) for developing the future system,
- formalising an initial prototype in the form of an offer.

The main objective of the conceptual model is to model expertise. That is, to extract the underlying process that the human expert uses to generate a solution of a problem and then to model it. The conceptual model represents an abstraction of the acquired knowledge that is independent of further implementation. The design model represents an implemented conceptual model in terms of the users' and the external requirements (resulting in a certain performance of the expert system). The design model must define the future system's:

- functional aspect – what the system has to "do",

- behavioural aspect – how the functions of the system have to be performed (methods or a combination of methods have to be selected to achieve these functions),
- structural aspect – how the modularization of the system has to be established to achieve the defined functions and behaviour of the system (i.e. with how many modules, what is the functionality of each module, what are the relations between modules, what are the relations (interfaces) with other systems and/or humans).

The next step in the development of an AI application is to select the appropriate AI techniques and paradigms to be employed. AI techniques are often computationally intensive and, therefore, an undue burden might be placed on the hardware on which the AI application should run. Hence, it is important to determine the hardware constraints like the deployment platform in an early stage of the AI application development process. Sometimes the actually available hardware may influence the choice of a specific technique, and sometimes a chosen technique will determine the hardware requirements. In any case one should be aware that both the hardware and the software tools available might change during the project and the simpler the system, the easier it is for the engineering team to develop it and for the users to accept it. To select a proper software tool for the development of the future expert system, the following tool-selection criteria have to be kept in mind:

- *Integration*: the tool must allow efficient integration of all subsystems.
- *Portability*: the tool should operate on the same platforms as the subsystems.
- *Support*: the vendor's financial viability and its commitment should be assured before purchasing a tool in order to guarantee that the problems which may arise during the usage of the tool will not prove to be not insurmountable.
- *Easy learning and use*: the tool should be easy to learn in order to assure that the future users will incur no training delays.

Finally, the results of the whole system specification phase may be formalised in a prototype. This (first) prototype should focus on a general description of the future system's functionality, behaviour and structure. It is reasonable to make such a prototype before proceeding with a more detailed design of the system because the customer management can still cancel the whole project (and in-depth development would be a complete waste of time). Accepting this prototype in the form of an offer would commit customer management to their further (full) support of the project.

The refinement and implementation phase involves the development of a detailed design of the future system and the production of the code of the intended AI application. For example, in the incremental development model, this phase involves selecting the appropriate successive increments covering the full breadth of the future system in an easy-to-integrate way. Each increment forms a proper functional extension of the design model (in other words, has the proper depth), which further has to be coded and tested. Some increments can be developed and implemented in parallel (especially those that have no knowledge component as, for example, system interfaces) but, in any case, it is essential that all increments should be fully compatible and integrable with all previous and future increments. When the last increment is implemented, the whole specification must be as complete as it should be in waterfall model; the only difference is that it is already coded. All the subsystems can now be integrated and tested as a whole.

By final system testing the developed AI system has to be validated and verified. In other words, the system should satisfy:

- all of the specified requirements (to assure that it is the right product), and
- all of the additional specifications generated during the development (to assure that it is a product of quality).

In the system-transfer phase, the following actions have to be carried out:

- acceptance of the developed system,
- release of user manuals and complementary documentation,
- training of the users, and
- performance of all necessary modifications in order to overcome the problems that could arise when the system is integrated within the customer's hardware platform.

Once the final AI system has been validated, accepted and installed in its definite environment, the system needs some degree of maintenance and enhancement in order to evolve with its environment. The more dynamic the environment, the more maintenance effort will be required. The working environment of an AI system is usually highly dynamic and therefore maintenance, enhancement and support form an important phase in an AI system's life cycle.

4 Facial expression data extraction

> *Discerning the existence and location of a face, localising its features and tracking its movements, are perceptual abilities that have not found their own place in the behavioural science literature, but duplicating these native, autonomous functions computationally is not trivial. Yet, this task is a precursor to determining the information that the face provides.*
>
> *(Yuille and Pentland 1993)*

A first step in automating facial expression analysis from digitised facial images is to investigate and decide on sensing and processing techniques that can automatically extract representations of faces and facial features from static images or image sequences. These representations should be further automatically transformed into descriptions that psychologists usually use to describe facial expressions (such as the AUs defined in FACS; see also the discussion on the issue in section 2.1).

This chapter is organised as follows. The issues of sensing and processing the facial images are explained in section 4.1. First the selection and arrangement of sensors for monitoring facial expressions is discussed. Then the detection of the presence of the face in an observed image and the detection of the facial features from static images and image sequences are discussed next. Finally, we consider these issues in the scope of ISFER. Next, the first part of ISFER, that is the Facial Data Extractor (see Figure 2.25), is presented in section 4.2. The Facial Data Extractor is a framework for *hybrid facial feature detection* from an input, static dual-view facial image. The facial expression information extracted by the system is hybrid in the sense that per facial feature, multiple feature detectors integrated into the framework are applied. The detectors integrated into the Facial Data Extractor part of ISFER are summarised in section 4.3. Finally, the benefits and the limitations of the proposed technique for automatic extraction of facial expression information from a static facial image are discussed in section 4.4. Some guidelines for enhancing the proposed method are also summarised.

4.1 Sensing and processing

Various factors must be considered when selecting and arranging sensors for monitoring facial expressions. The essential parameters are quite simple: the spatial resolution of the static images or the spatial and temporal resolution of the video images, and the camera's field of view. The sensor must provide sufficient detail to enable the discrimination of expressions of interest, and it must provide either a sufficiently wide field of view or a way for controlling camera gaze and zoom in order to ensure that the face stays in view. The *sensor data rate* is the product of field of view, spatial resolution (samples per unit angle), and temporal resolution (frame rate). While it may be desirable to use cameras with the highest possible spatial and temporal resolution and the widest possible field of view, this can place an undue burden on the computer that must analyse the resulting data. Therefore, while deciding on the appropriate facial expression monitoring sensor(s) two factors should be taken into account: the required sensor data rates strongly depend on the intended application, and they can easily exceed the limits set on processing time and/or computing devices. Hence it is highly beneficial to investigate and develop strategies for extending the field of regard (e.g. by controlling camera gaze and zoom) while maintaining a high resolution and making the most effective use of limited computing resources. Finally, one should decide on the strategies that would ensure successful detection of the existence and location of the face and extraction of the facial expression information from the images obtained by the facial monitoring sensors selected for the intended application.

Sensing and application environment

A standard NTSC (National Television System Committee) or PAL (Phase Alternate Line) video camera provides an image that, when digitised, measures approximately 720×480, respectively, 720×576 pixels. For a typical face-monitoring task it may be necessary to arrange the camera so that there are at least 100 pixels across the width of a subject's face. The field of view can then be about five times the width of the face. This camera setting should be sufficient for applications like performance monitoring or HCI, in which the subject is seated but otherwise free to move his head. On the other hand it may not be sufficient for applications like lip reading or facial action tracking, in which a high spatial resolution is necessary for the discrimination of subtle facial changes, or for applications like animation and virtual reality, in which the subject is free to walk in front of the camera and to approach and move away from the camera.

The temporal frame rate required for monitoring facial expressions depends on the types of expressions that are of interest. Some expressions, such as a smile or frown, may persist for several seconds. Others, like blink or wink, last only for a fraction of a second. A frame rate as low as one frame per second may suffice if one needs only to determine presence as opposed to temporal information. Monitoring more subtle or quick changes in facial expression may require ten or more frames per second. Lip reading, for instance, requires full NTSC or PAL frame rates (i.e. 30 or 25 frames per second).

As noted above, it would be highly beneficial to develop strategies for extending the field of regard while maintaining a high resolution and keeping the computational load low. Using a single camera which can pan and zoom under computer control to facilitate following of the moving faces is such a strategy. Still, using a controlled camera introduces other complications. Namely, a special camera mount with drive motors is required and fast image analysis should be facilitated in order to determine where to orient the camera on a moment-by-moment basis. Another strategy which can provide high-resolution images and keep data rates and computational loads low while imposing no constraints on the field of view is to use a head-mounted camera. This strategy also introduces a new set of drawbacks. For instance, a special camera mount like a helmet or a headphone device is required. Such a device can be heavy and hence inconvenient for the user as it reduces the freedom with which he/she can move around and with which he/she can turn the head.

The investigation and development of sensors and analysis techniques having the capabilities described above is the subject of research in the field of so-called *active vision*. In general terms, the objective of active camera control is to focus sensing resources on relatively small regions of the scene that contain critical information. Hence, an active vision system has to observe the scene with a wide field of view at a low spatial resolution in order to determine where to direct high spatial resolution observations. This is analogous to human vision in the fovea. The fovea is a small depression in the retina where vision is most acute; the fovea provides the resolution needed for discriminating patterns of interest, while the periphery provides broad-area monitoring for alerting and gaze control. Applying the principle of the fovea and allocating the sensing resources to both broad-area monitoring and observations of the region of interest can reduce the actual data that needs to be provided by a sensor and then processed by a factor of 1000 or more (Bajcsy 1988). This can easily mean the difference between a system that is too large to be considered at all and one that is sufficiently small to be generally used.

There are two primary areas of research in the field of active vision: the design of fast, intelligent, control processes to direct the camera and the development of special sensors based on the principle of the fovea, i.e. facilitating *foveal vision*. In addition, vision should serve a purpose (Aloimonos et al. 1987) and a vision system should operate continuously and furnish the results within a fixed delay, i.e. a vision system should achieve real-time visual computation determined by the goals of the intended application. Although active vision technology is a hot topic nowadays and quite some progress in the field has been made, this work needs to be extended for face localisation and tracking for the specific application of automatic facial expression analysis. It is certain that the field of automating facial expression analysis would highly benefit from the field of active vision; it could make automatic observing of the facial expressions more efficient, more effective, and more feasible in domains such as medicine, HCI, and in other commercial applications.

Detecting the position of the face and its features

Determining the exact location of the face in a digitised static facial image or an image sequence is by no means a trivial task. However, this task is a precursor to determining the information that the face provides. Without knowledge of where the face is, most feature-extraction algorithms (see Table 2.6) will produce many false targets and hence be useless. A robust way to locate the faces in images that is insensitive to scale, pose, self-occlusions (e.g. eye blinking), hair style, wearable accessories, facial expression, illumination and lighting conditions has been proposed by only one facial-expression-analysis research group (Essa and Pentland 1997, Pentland et al. 1994). This is still a key research topic, especially when it comes to complex environments with multiple moving objects. However, it is likely that the currently existing algorithms for head tracking (e.g. Rowley et al 1998, Terrillon et al. 1998, Smeraldi et al. 2000) will suffice for the needs of automated facial expression analysis.

Once the faces have been properly located, the knowledge about spatial features of a face can be used very effectively. The face is a good subject for computer vision research because the (global) shape of the prominent facial features (eyebrows, eyes, nose, mouth and chin), their relative arrangement, and the anatomical rules by which their appearance changes are universal, regardless of age, gender and race (Ekman and Friesen 1978). Consequently we have a priori knowledge to model the face. By using that model (see Table 2.5 for various face models) we can extract information-bearing features.

Feature extraction may be divided into at least three dimensions, represented in Figure 4.1 (each vertex of the cube is annotated with Static or Dynamic, Global (holistic) or Analytic, 2D view-based or 3D volume-based). The first consideration is dynamic

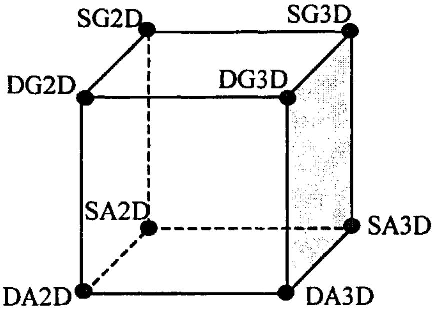


Figure 4.1: The Necker Cube of image processing

versus static features; whether or not temporal information (i.e. a sequence of images) is used? The second consideration is the grain of the features. The features may be global (i.e. holistic), spanning roughly the whole object being analysed, or they may be analytic (i.e. part-based) features, spanning only subparts of the image. These issues have been elaborated throughout section 2.1. The third consideration is view-based (i.e. 2D) versus volume-based (i.e. 3D) features. 3D features can be extracted using special sensors or active sensing and are out of the scope of this thesis, whose goal is automatic static-image-based facial expression analysis. Some considerations that have been left out of the Necker Cube include whether the sensors are active or passive and whether the features are predefined or learned by an adaptive mechanism depending on the data. Chapter 2 discusses in detail those issues as well as what has been accomplished in these traditional corners of the Necker Cube. In summary, given the nomenclature illustrated in Figure 4.1, most of the computer-vision systems for automatic facial expression analysis proposed in the literature are directed towards static, holistic or analytic, 2D feature extraction (see Table 2.2 and Table 2.3).

Feature extraction and computation of changes in facial expression is a prerequisite for automatic facial expression analysis. Hence, when building a vision-based system for automatic facial expression analysis, one should be aware that defining and segmenting facial expression information well is probably not sufficient, but in any case it is a necessary step towards a reliable, robust, fully-automatic facial expression recognition.

Sensing and processing in ISFER

The Integrated System for Facial Expression Recognition (ISFER) is strongly application dependent (see also section 2.6). The main goal for its development was to achieve fully automatic facial expression analysis, which is applicable to automated FACS coding and automated facial expression classification in observer-defined interpretation categories, so that it can be employed for behavioural science investigations of the face. Hence, this application domain defines the environment in which the system is to be used primarily. Since in behavioural science investigations of the face the research material usually consists of full-face photographs of subjects, the sensor utilised for obtaining visual data of the examined facial expressions should provide the system with static facial images of the monitored subject. In order to facilitate encoding of facial actions in images, the presence of the monitored face in images must be ensured. Also, a high spatial resolution is necessary for the discrimination of subtle facial changes. The sensor used to acquire visual data of the examined facial expression consists of two CCD (Charged Coupled Device) PAL video cameras whose digitised output is used as input to ISFER. The employed cameras are mounted on the head of the observed subject and provide high-resolution images while imposing no constraints on the field of view.

The cameras acquire images that, when digitised, measure approximately 720×576 pixels.

The cameras are mounted in the following manner. Two holders carrying the cameras are attached to a head-phone-like device, which is then mounted on the head of the monitored subject. One camera is placed in front of the face, at approximately 15 centimetres from the tip of the nose. This camera acquires a frontal-view image. The second camera, placed on the right side of the face at approximately 15 centimetres from the centre of the right cheek, acquires a profile-view image. Figure 4.2 illustrates the utilised head-mounted cameras and provides an example of an input dual-view facial image acquired by this monitoring device. The cameras move as the subject moves his/her head, ensuring both the presence of the examined face in the acquired images and the absence of rigid head movements while monitoring the non-rigid facial movements at a high spatial resolution. In other words, the images obtained during a single session are scale- and pose-invariant.

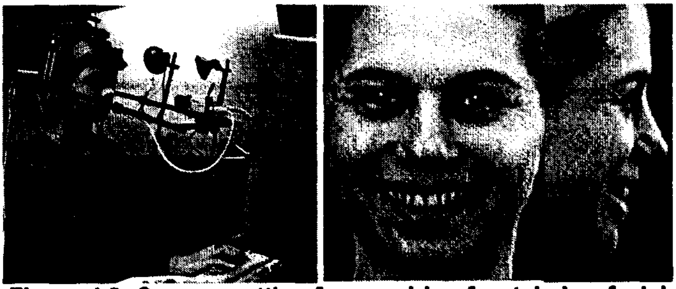


Figure 4.2: Camera setting for acquiring frontal-view facial images and an example of acquired dual-view

The utilised camera setting ensures the acquisition of visual facial-expression data that are highly suitable for the purposes of behavioural science. Nevertheless, the device described above introduces other complications. The device is heavy and therefore inconvenient for the observed subject since it reduces the freedom with which the subject can move around and with which he/she can turn the head. The subject is required to remain seated and to move the head rather slowly since a quick body movement may cause a displacement of the device and, therefore, a change of the viewing angle. On the other hand, when reasoning about the accuracy of the input data and about displayed facial actions, the system compares the currently examined facial expression and the prior recorded expressionless face of the observed subject in terms of few characteristic, immovable facial points (section 5.4). Hence, a change in the viewing angle will produce many false conclusions and thus make the result of the system useless.

The actual input to the system can either be a static frontal-view facial image obtained from a database containing behavioural science research material or obtained by the head-mounted camera placed in front of the face, or a static dual-view facial image representing combined information acquired by both head-mounted cameras. The question that had not been answered yet is the motivation for facilitating an automated facial expression analysis from static dual-view facial images given that dual views do not represent a standard image format used for behavioural investigations of the face. The reason is the increase in quality of the facial expression analysis, which emerges from the increase of the available information (Wojdel, A. et al. 1999). First, automatic extraction of a dual-view facial representation from static dual-view facial images facilitates automatic encoding of 32 different AU codes (Table 5.8) as opposed to 22 different AU codes that can be automatically encoded from a frontal-view facial image (Table 5.5) using the detectors currently integrated into the Facial Data Extractor. Second, a more accurate estimation of the certainty of the input data is facilitated since there is more data, resulting from different detectors, which can be compared (sections 5.4 and 5.6). Hence, this leads to more accurate results. Chapter 5 explains these issues in detail.

The input data (a frontal-view facial image or a dual-view facial image) is further processed by the Facial Data Extractor part of ISFER, which represents a framework for hybrid facial feature detection in facial images. Given the nomenclature illustrated in Figure 4.1, the hybrid facial feature detector explained in section 4.2 belongs to the static, analytic, 2D corner of the Necker Cube.

4.2 Framework for hybrid facial feature detection

Recent advances have been made in computer vision on the topic of automatic recognition of facial expressions in images. In chapter 2, a number of different systems for facial expression recognition in static facial images and image sequences are explored and compared. In summary, the approaches include:

- analysis of facial motion (Mase 1991, Yacoob and Davis 1994, Rosenblum et al. 1994, Black and Yacoob 1997, Cohn et al. 1998, Lien et al. 1998, Essa and Pentland 1997, Otsuka and Ohya 1998),
- holistic spatial pattern analysis (Cottrell and Metcalfe 1991, Matsuno et al. 1993, Vanger et al. 1995, Padgett and Cottrell 1996, Edwards et al. 1998, Hong et al. 1998, Huang and Huang 1997, Yoneyama et al. 1997, Kimura and Yachida 1997, Wang et al. 1998, Lyons et al. 1999),
- grey-level pattern analysis using local spatial filters (Zhang et al. 1998, Lyons et al. 1999),
- analytic spatial analysis (Rahardja et al. 1991, Kobayashi and Hara 1992a-b, Ushida et al. 1993, Kearney and McKenzie 1993, Moses et al. 1995, Zhao and

Kearney 1996, Kobayashi and Hara 1997, Huang and Huang 1997, Cohn et al. 1998), and

- image analysis based on physical models of the facial skin and musculature (Mase 1991, Terzopoulos and Waters 1993, Li and Roivainen 1993, Thalmann et al. 1995, Essa and Pentland 1997, Matsumura et al. 1997, DeCarlo et al. 1998, Eisert and Girod 1998).

Most, if not all, of these systems detect facial features in facial images by utilising detectors of a single kind (Bartlett et al. 1999, Donato et al. 1999, Pantic and Rothkrantz 2000d). In contrast, the Facial Data Extractor part of ISFER takes a "hybrid" approach to facial-expression-data extraction.

The Facial Data Extractor is a framework for *hybrid* facial feature detection in the sense that it applies facial feature detectors of different kinds. In fact, to localise the contour of a prominent facial feature (profile, eyebrow, eye, nose, mouth and chin) in an input static facial image, multiple feature detectors of different kinds are concurrently applied. The motivation for applying multiple different detectors to localise any of the prominent facial features is the increase in quality of the results obtained. This is because most of the known methods for facial feature detection in static facial images (e.g. snake fitting, template matching, local spatial filtering) are prone to changes in illumination and lighting conditions and distractions like blinking, facial hair, glasses, etc. In turn, each feature detector has circumstances under which it performs poorly and circumstances under which it performs extremely well. Introducing redundancy in facial expression data by employing a hybrid facial feature detector and then selecting the best of the acquired results yields a more accurate and complete set of detected facial features (i.e. less missing data). Hence, a hybrid facial feature detector of the kind explained above results in a more robust performance than either a single detector used for all facial features or a set of different detectors, each of which used for one facial feature.

Another main characteristic of the framework for facial feature detection employed in ISFER is that it has been built by reusing the existing knowledge. That is, each facial feature detector integrated into the Facial Data Extractor is an already existing feature detector modified to fit the purposes of localising contours of the prominent facial features. So rather than fine-tuning the existing techniques or inventing new techniques for feature localisation in static facial images, known techniques were modified and then combined. Section 4.3 provides an overview of the feature detectors integrated into the Facial Data Extractor. A detailed algorithmic representation of the processing of Facial Data Extractor part of ISFER is provided in Appendix A.

Facial features

The features extracted by the Facial Data Extractor from a static dual-view facial image are the contours of the eyebrows, eyes, nostrils, mouth, and profile, representing typical predefined features as opposed to the features learned by an adaptive mechanism depending on the data. Hence, when mapped to the Necker Cube illustrated in Figure 4.1, the features extracted by the Facial Data Extractor belong to the static 2D analytic corner as opposed to the dynamic 3D holistic corner of the Necker Cube.

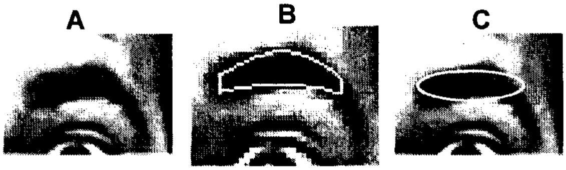


Figure 4.3: A) original image B) approximation by two 2nd degree parabolas C) approximation by an ellipse

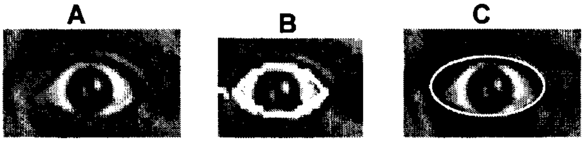


Figure 4.4: A) original image B) approximation by two 2nd degree parabolas C) approximation by an ellipse

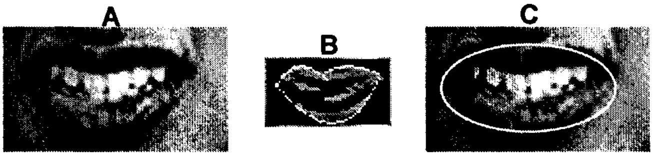


Figure 4.5: A) original image B) approximation by three 2nd degree parabolas C) approximation by an ellipse

For profile detection, a spatial approach to sampling the profile contour from a thresholded profile-view image is applied (Wojdel, J. et al. 1999). The result of the profile detector is the curvature of the profile contour function. As far as the frontal-view prominent facial features are concerned, generally the detectors integrated into the Facial Data Extractor approximate the contours of those features in terms of two 2nd degree parabolas. Arguably, the contour of any prominent facial feature to be localised from a frontal-view facial image (eyebrow, eye, mouth) can also be approximated by a single 2nd degree parabola like a circle or an ellipse. Yet, as illustrated in Figures 4.3 to 4.5, utilising two instead of a single 2nd degree parabola yields a more accurate approximation of the contours of the eyebrows, eyes, and mouth. An accurate approximation of the facial features' contours is crucial for the detection of subtle changes in facial expression and hence necessary for a robust facial action encoding from facial images, which represents one of the main goals of ISFER development. This explains the choice of approximating the contours of the eyebrows, eyes, and mouth by two instead by a single 2nd degree parabola.

Design and implementation

Since 1992 there is an ongoing research project at the Knowledge Based Systems group of Delft University of Technology whose aim is the design and implementation of an automated analyser of human non-verbal communicative signals (Figure 1.1). Most work concerned the design and development of an automated facial expression analyser and the implementation of different modules for the extraction of facial expression data from either static full-face images (Rothkrantz et al. 1998) or static profile images (Wojdel, J. et al. 1999). While the existence of various facial feature detectors facilitated the development of the framework for hybrid feature detection from static dual-view facial images, the course of the performed research imposed some additional constraints on the development of the Facial Data Extractor. First, the research is still ongoing. This means that integrating new facial feature detectors into the Facial Data Extractor should be facilitated while no constraints must be imposed by the programming language in which the detectors are implemented (the code of most of the existing modules has been written in either C or C++). Second, the framework should be a portable interactive user-friendly platform that will facilitate developers to implement, edit and test new facial feature detectors while working on either Sun Solaris, MS Windows, or Macintosh work stations, which are usually used in our group. Finally, the framework should also perform as an integral part of ISFER.

Availability of the Abstract Window Toolkit being a part of JDK (Java Development Kit), which is a platform-independent visual-interface tool builder, and JNI (Java Native Interface), which facilitates invoking native (non-Java) coded methods, made Java perfectly suitable for development of the Facial Data Extractor. The Facial Data Extractor has been implemented in Java and developed as a portable easy-to-enlarge interactive user-friendly platform that can operate stand-alone as well as part of a larger system and extract facial expression information automatically either from static frontal-view facial images or from static dual-view facial images. One might argue that the time-consuming execution of a Java-implemented application forms a serious drawback of the system. However, this is of little concern in the case of the Facial Data Extractor since the time spent by the processor on executing the code of the framework itself is very short compared to the time spent on executing the code of various modules integrated into the framework.

The modules of the Facial Data Extractor can be classified into three groups:

1. The *pre-processing group of modules*, which contains: the modules for acquiring the images from a database containing some behavioural science research material, the modules for generating digital static facial images (frontal-view or

dual-view) from the analogue signals coming from two mounted CCD PAL video cameras, and the modules for filtering the image data. The modules of the pre-processing group are explained in Table 4.1.

2. The *detection group of modules*, which contains the modules detect facial regions (face region, eyebrow-eye region, nose region and mouth region) in a frontal-view facial image. These modules are described in Table 4.2.
3. The *extraction group of modules*, which contains the modules that localise the contours of the prominent facial features in static facial images. These modules are described in detail in section 4.3.

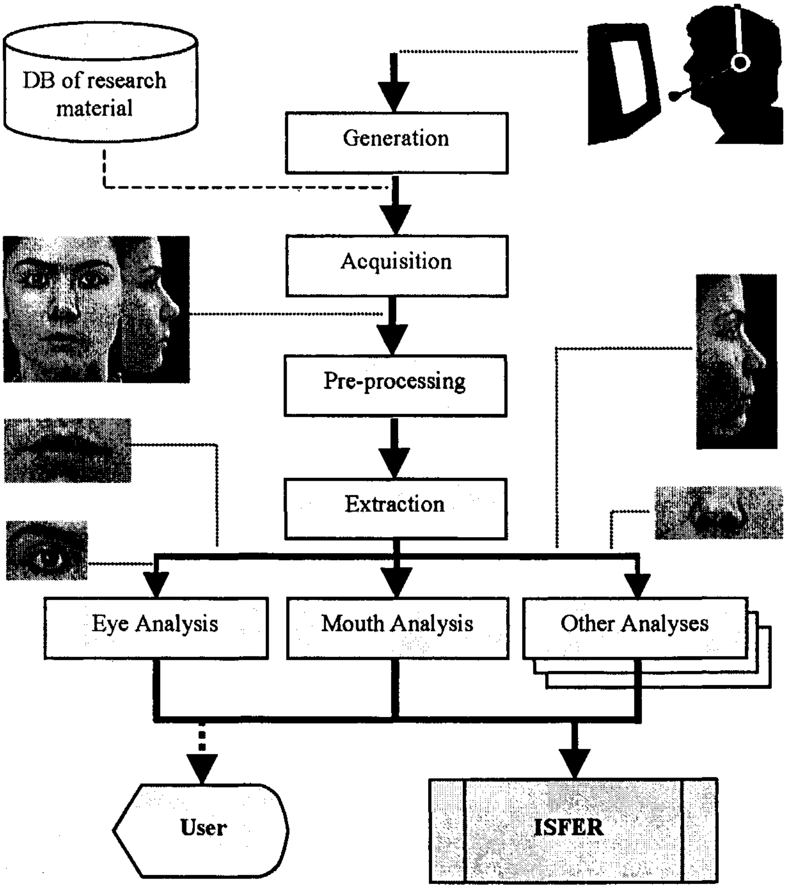


Figure 4.6: Algorithmic representation of the processing of the Facial Data Extractor

The modules of different groups interact as illustrated in Figure 4.6 (see also Appendix A). The contours of the prominent facial features detected by the Facial Data Extractor may be displayed to the user or further analysed by ISFER, depending on the operating mode in which the Facial Data Extractor is executed.

Table 4.1
The modules of the pre-processing group

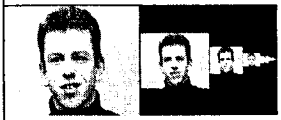
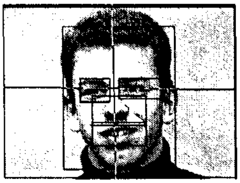
Module	Module Description
Image to Colour Implemented in Java	Conversion of Java *Image* data to a flat array of pixels
Colour to Grey Implemented in Java	Conversion of the colour picture to a grey picture
Convolution Filter Implemented in Java	Noise removal and smoothing of the image by applying linear convolution filtering with Gaussian or a Uniform filter (Glassner 1993)
Median Filter Implemented in Java	Enhancement of the continuous areas of constant brightness in the image and slight sharpening of the edges by applying non-linear Median filter (Glassner 1993)

Different modules integrated into the Facial Data Extractor can be invoked in a stand-alone mode or so-called ISFER mode. When used in the *stand-alone operating mode*, the Facial Data Extractor facilitates developers to implement, edit and test new facial feature detectors. The user is allowed to select and then connect an arbitrary number of modules in order to form a network of modules that performs a desired task; for instance, the localisation of the prominent facial features from a frontal-view facial image (Figure 4.7). At any moment the current network is displayed to the user in the form of a directed graph, where the nodes of the graph depict the modules and the branches depict the connections between the modules. Each node of the network is represented as a box containing the name of the module and the types of input and output of the module. When executed, each module can accept as input and generate as output any number of data elements. Each of these data elements has a specific type such as:

- grey-scale image, depicted in a module box as “Grey”,
- feature contour points, depicted in a module box as “FCP”,
- filed data, depicted in a module box as “FD”.

For each of the data elements a module accepts as input, there is a specific area in the module box labelled with the data type of the given data element. At this area, called the *in-connector*, a connection from another network module can end. Similarly, for each of the data elements a module generates as its output, there is a specific area (*out-connector*) in the module box which is labelled with the data type of the given data element and at which connections to other network modules can start.

Table 4.2
The modules of the detection group

Module	Module Description
Grey to MRP implemented in C (De Bondt 1995)	Creation of the layers of the Multi Resolution Pyramid by calculating the half of the current image resolution (rounded to higher integer value) and averaging squares of 2x2 to one, which half the image in both directions. The routine is performed recursively until both image sizes equal 1 (see figure below).
MRP to RFM implemented in C (De Bondt, 1995)  	The module reads the given Multi Resolution Pyramid and locates, on the given layer (the default layer is 2), the Raw Feature Map that represents a rough approximation of the locations of the facial features. First the head is located by applying sequentially the analysis of the vertical histogram (showing the colour differences between the successive rows, pixelwise) and then the horizontal histogram (showing the colour differences between the successive columns, pixelwise). The peaks of the vertical histogram of the head box correspond with the borders between the hair and the forehead, the eyes, the nostrils, the mouth and the boundary between the chin and the neck. The horizontal line going through the eyes goes through the local maximum of the second peak. The x co-ordinate of the vertical line going between the eyes and across the nose is chosen as the absolute minimum of the contrast differences found along the horizontal line going through the eyes. The box bounding the left eye is first defined to have the same size as the upper left face quadrant (defined by the horizontal and the vertical line) and to lie so that the horizontal line divides it in two. By performing the analysis of the vertical and the horizontal histogram, the box is reduced so that it contains just the local maxima of the histograms. A similar procedure is applied to define each of the boxes bounding the right eye, the nose and mouth. The initial mouth box is set around the horizontal line going through the mouth, under the horizontal line going through the nostrils and above the horizontal line representing the border between the chin and the neck. The initial nose box is set around the horizontal line going through the nostrils, under the horizontal line going through the eyes and above the horizontal line going through the mouth.
Find Head Contour Implemented in C (Rothkrantz et al. 1998)	The algorithm is based on the HSV colour model. The first step is to define the value of the parameter *Hue* ∈ [-60, 300]. Analysis of 120 full-face images of different people results in the conclusion that the *Hue* of the face colour seldom exceeds the interval of [-40, 60]. These experimental results also yield the fact that the range of *Hue* never exceeds 40 for the images

	of a single face, irrespective of changes in lighting conditions. The *Hue* is defined as [-40 < *average Hue* –20, *average Hue* +20 < 60], where the average *Hue* is calculated as the average of the *Hue* in the box containing a horizontal middle of the face. The box is defined by analysing the vertical and the horizontal histogram of the input image. The face is then extracted as the biggest object in the scene having the *Hue* in the defined range. Yang and Waibel (1996) have presented a similar method, but based on the relative RGB model.

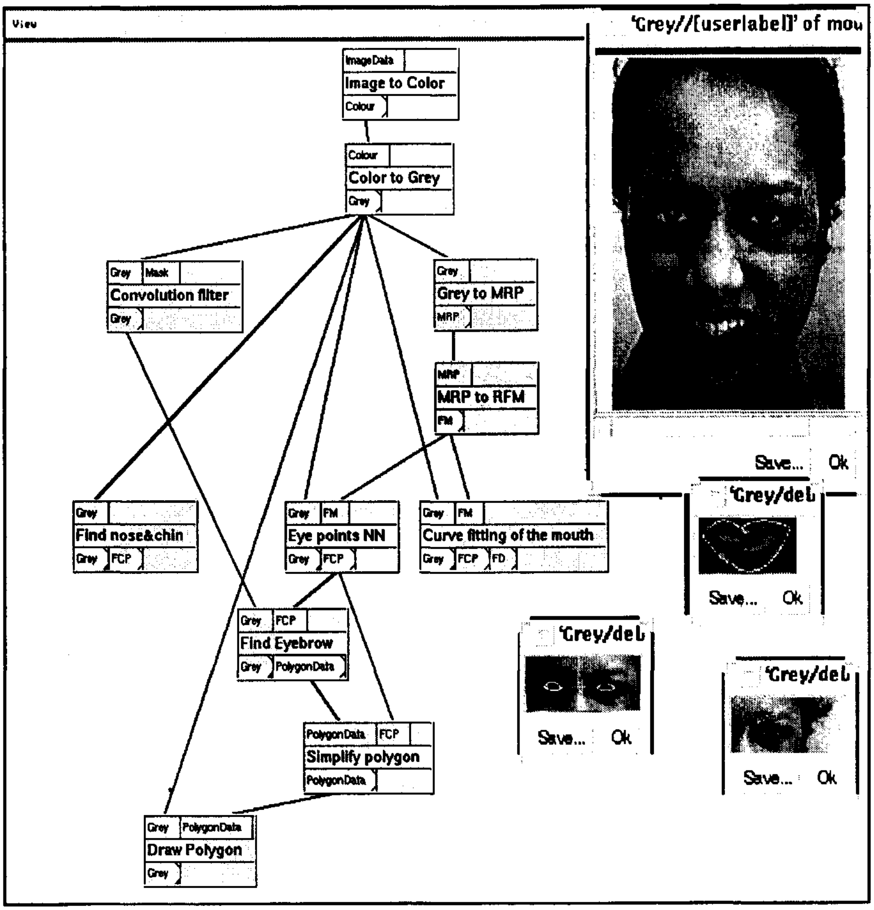


Figure 4.7: Screen shot of the stand-alone operating mode of the Facial Data Extractor

Two modules can be connected in a network when the output of one module forms the input to the other module. Each time when a connection is made, it is checked if the in-connector and the out-connector of that connection have matching data types. Only connections between matching data types are allowed. In addition, the framework prevents the user from introducing loops in the network. Once a network has been created and the input to the Facial Data Extractor has been specified, the network can be executed. The framework then searches for the modules that have all the required input specified, executes those modules, collects their output and displays and/or saves it according to the user's instructions defined in the out-connectors of the modules.

When used in the *ISFER operating mode*, the Facial Data Extractor does not interact with the user; each and every facial feature detector integrated into the framework is invoked automatically (see Appendix A). The result of each detector is stored in a separate file, that is, for each module which belongs to the extraction group of framework modules, the out-connector labelled with "FD" is clicked. Those files form further the input to the next part of ISFER, that is, to the Facial Action Encoder explained in chapter 5.

4.3 Overview of the integrated facial feature detectors

The extraction group of the modules integrated into the Facial Data Extractor contains the modules that localise the contours of the prominent facial features in static facial images. These modules can be further classified into a few categories:

- *Profile detectors*: these modules localise the profile contour in an input static profile-view image. Only one such module is currently integrated into the Facial Data Extractor.
- *Eyebrow detectors*: these modules localise the contour of an eyebrow in an input static frontal-view image. Two such modules are currently integrated into the Facial Data Extractor.
- *Eye detectors*: these modules localise the contour of an eye in an input static frontal-view image. Two such modules are currently integrated into the Facial Data Extractor.
- *Nostril detectors*: these modules localise the contours of the nostrils in an input static frontal-view image. Only one such module is currently integrated into the Facial Data Extractor.
- *Mouth detectors*: these modules localise the mouth contour in an input static frontal-view image. Two such modules are currently integrated into the Facial Data Extractor.
- *Mouth classifiers*: these modules classify an input frontal-view image into some interpretation categories (e.g. smile, sad and neutral) according to the extracted

shape of the mouth. Two such modules are currently integrated into the Facial Data Extractor.

These categories of feature detectors integrated into the framework for hybrid facial feature detection are explained in the rest of this section. However, it should be stressed that this section does not provide an exhaustive overview of each feature detector integrated into the Facial Data Extractor. Per prominent facial feature, all relative facts are provided for only one of the integrated detectors. The processing of the other detectors that localise the same facial feature is just described shortly.

Database of test images

In the course of years, a large database of static facial images of subjects displaying different facial actions and basic emotional expressions has been collected (e.g. De Bondt 1995, Profijt 1995, Pantic 1996, etc.). The full database contains over 1600 frontal views, profile views, and dual views showing hundreds of distinct facial actions, and action combinations, displayed by 25 different subjects. The subjects were college staff and students of both sexes, who ranged in age (20 to 45), and ethnicity (European, Chinese, and South American). None of the subjects had a moustache, beard or wore glasses.

The database images were collected from various sources using various techniques. Approximately one fourth of the images have been acquired by scanning the photographs used as behavioural science research material. The rest of the images have been acquired under a constant illumination by recording the faces using the mounted camera device (Figure 4.2) or a standard PAL camera placed in front of the subject. All of the collected images, when digitised (or scaled), measure approximately 720 by 576 pixels.

Profile detector

The overall properties of the profile detector integrated in the Facial Data Extractor are summarised in Table 4.3. The applied method represents a spatial approach to sampling the profile contour from a thresholded input profile-view facial image. First the *Value* of the HSV colour model is calculated and exploited for the thresholding of the input image. The tip of the nose is then found as the most right highlighted part of the binary image (Figure 4.8). The tip of the chin is found as the first distinct minimum in the vector of summed background pixels from the bottom. To solve the problem of face rotation, which may be present if the input image has been acquired from a database containing behavioural science research material, the line between the tip of the nose and the tip of the chin is used as the x-axis of the new co-ordinate system. To obtain the profile contour from the binary image, the number of background pixels is counted between the right edge of the image and the first foreground pixel. This yields a vector that represents a sampling of the profile contour curve. To remove the noise from the contour, an average procedure is performed with a three-pixel wide window, which is slid along the vector. The zero crossing of the 1st derivative of the profile function defines extremes. Usually, many extremes are found (depending on the local profile change). The list of extremes is processed in both directions from the global maximum. The decision about particular extreme rejection is made using two consecutive records in the list. This obtains the list of extremes that reflect the most distinct peaks/valleys in the profile contour.

Table 4.3
Characteristics of the Profile Detector

Presented in:	Wojdel, J. et al. 1999
Implemented:	Java
Applied to:	Profile-view facial image
Method:	Spatial sampling of a thresholded image
Tested on:	112 profile-view images
Accuracy:	High

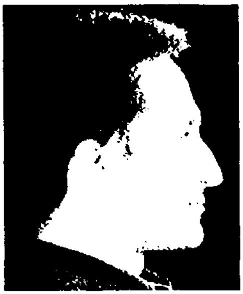


Figure 4.8: Thresholded profile-view facial image

Table 4.4
Distribution of *d(p, p_M)* for the localised profile characteristic points (PCPs) over 112 test images; numbers in the 1st column depict the number of images for which *d(p, p_M)* = 0, numbers in the 2nd column depict the number of images for which *d(p, p_M)* = 1, etc.

	0	1	2	3	4	5	6
PCP							
P1	8	14	49	27	11	3	0
P2	4	10	43	36	12	7	0
P3	5	9	48	39	9	2	0
P4	12	19	51	29	1	0	0
P5	5	23	47	34	3	0	0
P6	7	20	50	29	2	4	0
P7	3	8	45	40	10	6	0
P8	7	21	49	29	3	3	0
P9	2	10	40	37	15	8	0
P10	11	17	50	28	5	1	0

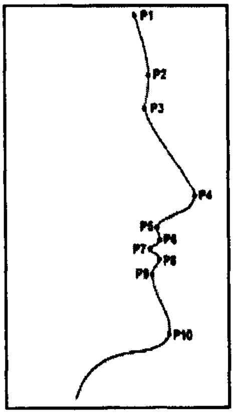


Figure 4.9: Profile Characteristic Points

The algorithm has been tested on 112 profile images representing seven basic emotional expressions shown twice by eight different subjects. The images were cut to contain just the profile and then scaled to measure approximately 240×290 pixels. Using Adobe PhotoShop and a mouse device, the profile characteristic points (Figure 4.9) were manually pointed by a human observer in all 112 images. The performance of the framework module *Find Profile Contour*, which is described above, has been evaluated by calculating the block distance (maximal difference in x and y direction) between the estimated and the manually located profile characteristic points in each test image. The performance of the algorithm is shown in Table 4.4. The localisation error for all profile characteristic points remained below 5 pixels and in most images the error was approximately 2 pixels. Most errors were caused by the difference in "definition" of the profile characteristic points in the case of manual and automatic estimation. Manually, the points were defined as the extremes of the profile contour while the automatic scheme tends to find the extremes of the curvature of the profile contour.

Eyebrow detectors

Two modules, namely *Curve Fitting of the Eyebrow* and *Chain Code Eyebrow*, localise the contours of the eyebrows. The overall properties of each are summarised in Table 4.5 and Table 4.6.

Table 4.5
Characteristics of the framework module Curve Fitting of the Eyebrow

Presented in:	Rothkrantz et al. 1998
Implemented:	C
Applied to:	Frontal-view fac. image
Method:	Contour following
Tested on:	60 frontal-view images
Accuracy:	Medium

Table 4.6
Characteristics of the framework module Chain Code Eyebrow

Presented in:	Raducanu et al. 1999
Implemented:	C
Applied to:	Frontal-view fac. image
Method:	Contour following
Tested on:	240 frontal-view images
Accuracy:	High

The framework module Curve Fitting of the Eyebrow localises the contours of the eyebrows one at a time. To localise the left eyebrow, a box containing the eye and the eyebrow is segmented from the frontal-view facial image. The box is first defined to have the same size as the upper left face quadrant (defined by the facial axes found by the detection-group module *MRP to RFM*) and to lie so that the horizontal axis divides it in two. The box is then reduced so that its boundary is defined by the contour of the face found by the detection-group module *Find Head Contour* (see Table 4.2). The eye-eyebrow region is determined by analysing the horizontal and the vertical signature (Haralick and Shapiro 1992) of the linearly filtered binarised image segment containing this box. The eyebrow region is then obtained by clipping the triangle defined by the eye points (the corners and the top of the eye found by one of the eye detectors explained in the following subsection) out of the eye-eyebrow region. Depending of the colour of the eyebrow (dark or light), the eyebrow region is thresholded. After a unique colour is assigned to each of the objects in the scene, the largest is selected and the rest of the objects are discarded. The contour-following algorithm based on 4-connected chain codes (Ritter and Wilson 1996) has been applied to localise the eyebrow contour. The processing of the module terminates by smoothing the localised contour with two simplified 2nd degree curves. A typical result of the module is shown in Figure 4.10.

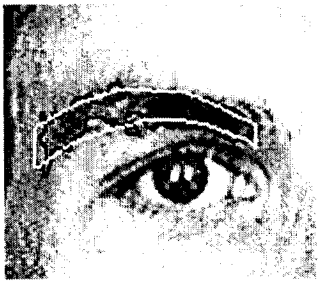


Figure 4.10: Contour of the eyebrow localised by the Curve Fitting of the Eyebrow

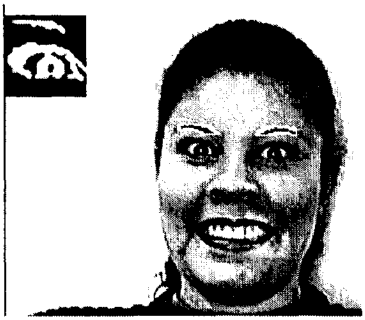


Figure 4.11: Contours of the eyebrows localised by the Chain Code Eyebrow

The module Chain Code Eyebrow localises the contours of the eyebrows simultaneously. To localise the left eyebrow, a box containing the eye and the eyebrow is segmented from the frontal-view facial image. The box is determined by the same procedure used by the module Curve Fitting of the Eyebrow and explained above. The segmented part of the image containing the box is then thresholded by applying the algorithm of minimum variance clustering (Haralick and Shapiro 1992). By analysing the horizontal and the vertical image signature of this segment, the eye-eyebrow region is located. The signatures are filtered using closing morphological filters given in formula *(1)*, where $v[n]$ is the signature on columns, $s[n]$ is the smoothed version of the signature, and $2k+1$ is the size of the applied structural element. The width of the eye-eyebrow region is set to the width between the first and the last index of the maximal value of the smoothed vertical signature. The height of the region is set to the width of the smoothed horizontal signature. The similar procedure of thresholding and segmenting is applied once again in order to define the eyebrow region. Then, the contour-following algorithm based on 4-connected chain codes is applied to localise the eyebrow contour. The very same procedure, only applied to the upper right face quadrant, is used to localise the contour of the right eyebrow. The algorithm has been tested on a set of 240 "almost" frontal-view facial images (i.e. some of the images contained limited in-plane and out-plane head rotations) picturing various facial expressions shown by 12 different subjects. The images have been scaled to measure approximately 360×290 pixels. First, all of the images were given to a human observer. Using Adobe PhotoShop and a mouse device, the observer pointed to the exact location of the four reference points: the outer and the inner corner of the left, respectively right, eyebrow. The performance of the detection scheme has been evaluated by calculating the block distance $d(p, p_M)$ between the estimated reference points $p = (x, y)$ and the manually located reference points $p_M = (x_M, y_M)$. A typical result of the module is shown in

Figure 4.11 and the performance of the algorithm is given in Table 4.7. As can be seen, the localisation error of each reference point remained below 7 pixels, which is a sufficiently low localisation error for detecting changes in the appearance of the eyebrows automatically (see also section 5.4).

$$\begin{aligned} v^e[n] &= \min(v[n+i]) \\ v^d[n] &= \max(v[n+i]) \qquad -k < i < k \qquad (1) \\ s[n] &= v^d[n] \end{aligned}$$

Table 4.7
Distribution of $d(p, p_M)$ for the reference points of the eyebrows over 240 test images; numbers in the 1st column depict the number of images for which $d(p, p_M) = 0$, numbers in the 2nd column depict the number of images for which $d(p, p_M) = 1$, etc.

	0	1	2	3	4	≥5	≥7
L out	61	19	83	14	54	9	0
L in	66	58	39	17	57	3	0
R out	40	20	51	48	46	32	3
R in	53	82	64	1	38	2	0

Eye detectors

Two modules, namely *Snake Eye* and *Eye NN*, localise the contours of the eyes. The overall properties of each are summarised in Table 4.8 and Table 4.9.

The framework module Snake Eye localises the contours of the eyes one at a time. To locate the box enclosing just one eye, the same method is used as in the Chain Code Eyebrow module. Namely, the box is first defined to have the same size as the upper left face quadrant (defined by the facial axes found by the detection-group module *MRP to RFM*) and to lie so that the horizontal axis divides it in two. The box is then reduced so that its boundary is defined by the contour of the face found by the detection-group module *Find Head Contour* (see Table 4.2). The eye-eyebrow region is then determined by analysing the horizontal and the vertical signature of the image segment containing this box. By applying this procedure again, but now on the image segment containing the defined eye-eyebrow region, a box containing just the eye region is determined. The algorithm applies further the active contour method proposed by Kass et al. (1987) with the greedy algorithm for minimising the snake's energy function proposed by Williams and Shah (1992). The method has been tested on a small number of test images (merely 45 frontal-view

Table 4.8
Characteristics of the framework module Snake Eye

Presented in:	Rothkrantz et al. 1998
Implemented:	C
Applied to:	Frontal-view fac. image
Method:	Snake fitting
Tested on:	45 frontal-view images
Accuracy:	Low

Table 4.9
Characteristics of the framework module Eye NN

Presented in:	De Jonge 1995
Implemented:	C
Applied to:	Frontal-view fac. image
Method:	Pattern recognition (NN)
Tested on:	252 frontal-view images
Accuracy:	High

images) and in 60% of the cases a successful snake fitting was obtained. This evaluation indicates that the method is not particularly useful for an application like facial action encoding from facial images, as a rather high precision of the detection schemes is necessary for detecting subtle facial changes. In addition, the processing of the module is highly time consuming (±2min on average). A result of the module is shown in Figure 4.12.

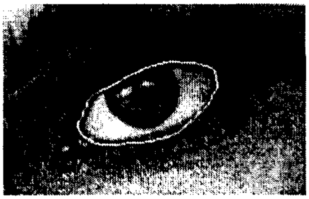


Figure 4.12: Contour of the eye localised by the Snake Eye

The module Eye NN localises the contours of the eyes simultaneously. The method represents a neural network approach to sampling the contours of the eyes from an input frontal-view facial image. Neural networks have an excellent capability to recognise specific patterns. This property is exploited here to extract graphical patterns from digitised images. The graphical pattern that is searched for is a combination of pixel values (grey values). The Eye NN module utilises an 81×4×1 back-propagation neural network with a Sigmoid transfer function. To detect the eyes in a digitised frontal-view facial image, the detector processes in two stages, coarse and fine.

For each of the eyes, a 9×9 pixels box enclosing approximately the iris of the eye is located in the coarse stage. The eye box is first segmented from the input image using the result of the MRP to RFM module. Then, a 9×9 pixels scan window is scanned over this segmented region. Each pixel of the scan window is attached to an input neuron of the neural network which has been trained to recognise the iris of the eye. The location where the highest neural response is reached is assumed to be the centre of the iris. In the next step, the scan window is set around this point. If the location where the highest neural response has been reached remains the same as in the previous step, the position of the iris is found. Otherwise, this step is repeated until the iris is found. A 9×9 pixels scan window that will be used in the fine stage of the algorithm is then set around the iris.

In the fine stage, the eye sub-features are localised. The idea behind searching the characteristic points of the eye by applying a neural network originates from the Hierarchical Perceptron Feature localisation method of Vincent et al. (1992). A difference between the two methods is the choice of the eye micro-features. The micro-features that are localised by the Eye NN module are illustrated in Figure 4.13. These

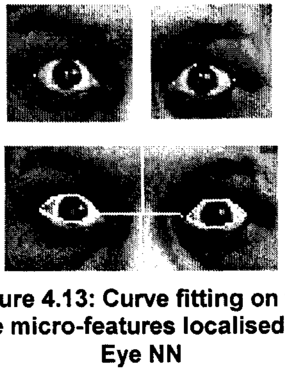


Figure 4.13: Curve fitting on the eye micro-features localised by Eye NN

Table 4.10
Distribution of $d(p, p_M)$ for the micro-features of the eyes over the training set; numbers in the 1st column depict the number of images for which $d(p, p_M) = 0$, numbers in the 2nd column depict the number of images for which $d(p, p_M) = 0.5$, etc.

	0	.5	1	.5	2	.5	3
R eye							
left	14	83	22	5	2	0	0
l-top	5	74	30	9	4	1	3
r-top	7	71	23	12	6	5	2
right	16	91	18	0	1	0	0
r-bot.	8	77	34	6	1	0	0
l-bot.	6	81	33	2	3	1	0
centre	8	65	45	8	0	0	0
L eye							
left	19	87	18	2	0	0	0
l-top	4	62	41	12	2	0	5
r-top	3	68	39	9	2	1	4
right	22	89	15	0	0	0	0
r-bot.	11	82	29	4	0	0	0
l-bot	9	81	32	2	0	2	0
centre	9	69	38	10	0	0	0

micro-features are invariant to the changes in size of the eye, in the shown facial expression, and in physiognomic characteristics of the person to whom the eye belongs. Similarly to the process of localising the iris of the eye (coarse stage of the algorithm), a 9×9 pixels search area is set for each micro-feature. Each pixel of the relevant search area is attached to an input neuron of a neural network that has been trained to recognise the relevant micro-feature. The location of the highest neural response reached represents the location of the micro-feature. A priori knowledge such as the symmetrical position of the features is used to discard false positives. In a final step of the algorithm, the border between the eyelids and the eye on which the micro-features lie is approximated by two 3rd-degree polynomials (see Figure 4.13).

For the experiments 252 frontal-view facial images of nine different persons were used. The images were scaled and clipped to measure approximately 320 by 240 pixels and the colour depth of 24 bits was reduced to 256 grey levels. The input material was divided into two groups of 126 images. Each group consists of 2×7 basic emotional expressions shown by nine different persons. One group of images has been used as a training set and the other as the test set of images. First, all images were given to a human observer. Using Adobe PhotoShop and a mouse device, the observer pointed to the exact location of the eye micro-features. Per image and for each micro-feature the training pattern has been obtained by extracting (row by row) an 81-dimensional vector of the grey levels of the pixels in a 9×9 pixels window that has been set around the micro-feature pointed out by the user. Per micro-feature an 81×4×1 back-propagation neural network was trained. Each network was trained using 126 input vectors until a small mean-squared error (<0.01) was reached for the training vectors (after approximately 1000 training epochs). Then the performance of the detection scheme was evaluated by calculating the block distance $d(p, p_M)$ between the estimated micro-feature $p = (x, y)$ and the manually located pertinent point $p_M = (x_M, y_M)$. The performance of the algorithm was measured first for the training set of images. The results of this test are shown in Table 4.10. From this table one can see that for all of the micro-features in all 126 images, the localisation error remained below 3 pixels. In most of the images the localisation error of each micro-feature is approximately 0.5 pixels. Table 4.11 expounds the performance of Eye NN module measured for the test set of images. From this table one can see that for all of the micro-features, the localisation error remains below 4 pixels and that in most of the images, the error is approximately one pixel. In fact, most of the larger localisation errors were caused by the difference in the "definition" of the eye centre in the case of manual estimation and of automatic estimation. Manually, the centre of the eye was defined as the centre of the iris, while the neural network tends to find the centre as the darkest point of the iris. Anyhow, the Eye NN module performs similarly to other eye detectors (Reinders 1997): for the pertinent resolution of test images, the average error is 0.98 pixels.

Table 4.11
Distribution of *d(p, p_M)* for the micro-features of the eyes over the testing set; numbers in the 1st column depict the number of images for which *d(p, p_M)* = 0, numbers in the 2nd column depict the number of images for which *d(p, p_M)* = 0.5, etc.

	0	.5	1	.5	2	3	4
R eye							
left	9	24	54	37	1	1	0
l-top	1	17	49	31	24	3	1
r-top	0	21	43	32	21	3	6
right	11	19	62	32	0	2	0
r-bot.	4	17	51	40	12	2	2
l-bot.	2	20	51	41	9	1	2
centre	2	42	57	24	1	0	0
L eye							
left	11	18	56	39	2	0	0
l-top	0	12	49	42	15	6	2
r-top	1	13	53	36	12	7	4
right	8	26	61	27	4	0	0
r-bot.	6	21	50	33	13	2	1
l-bot.	3	19	52	40	4	5	3
centre	2	47	59	15	3	0	0

Nostrils detector

The overall properties of the nostrils detector integrated in the Facial Data Extractor are summarised in Table 4.12. Except the contours of the nostrils, the *Find Nose & Chin* module localises the tip of the chin in a frontal-view facial image as well. The linearly filtered input image is thresholded and the seed-fill algorithm (van Dam and Foley 1995) is applied for colouring the important facial regions such as the eyes, nostrils, and mouth. The symmetry line between the important facial regions is found using an adapted version of the algorithm based on the Voronoi diagrams and presented by O'Rourke (1994). The region that is searched for the nostrils is defined by the second deepest valley of the brightness distribution along the symmetry line (a similar algorithm has been used by Hara and Kobayashi (1997a)). In this nostrils region, the small regions which are at approximately the same perpendicular distance from the symmetry line and which have the highest intensity values are considered the nostrils and approximated by two small circles. The tip of the chin is defined as the first peak after the third deepest valley (the mouth) of the brightness distribution along the symmetry line. A typical result of the module is shown in Figure 4.14.

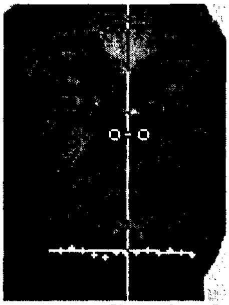


Figure 4.14: The tip of the chin and nostrils localised by Find Nose & Chin

Table 4.12
Characteristics of the framework module Find Nose & Chin

Presented in:	Rothkrantz et al. 1998
Implemented:	C
Applied to:	Frontal-view facial image
Method:	Brightness distribution analysis
Tested on:	88 frontal-view images
Accuracy:	Medium

The algorithm has been tested on a set of 88 "almost" frontal-view facial images (i.e. the images contained limited in-plane and out-plane head rotations, usually along the y-axis) picturing various facial expressions shown by 10 different subjects. The images have been scaled to measure approximately 360×290 pixels. First, all of the images were given to a human observer. Using Adobe PhotoShop and a mouse device, the observer pointed to the exact location of the three reference points: the inner corner of each nostril and the tip of the chin. Then the performance of the detection scheme was evaluated by calculating the block distance between the estimated reference points and the manually located reference points. For instance, if we denote the manually located inner corner of the left nostril by $l_M = (x_M, y_M)$ and the closest by point of the circle that represents the contour of the left nostril approximated by the Find Nose & Chin module (Figure 4.14) by $l = (x, y)$, then the performance indicator is expressed by $d(l, l_M)$. The performance of the algorithm is given in Table 4.13. From this table one can see that the localisation error of each reference point remained under 6 pixels. In fact, most of the errors have been caused by the difference in the "definition" of the reference points in manual and automatic estimation. The human observer usually marked the upper points of the nostrils as the inner corners (Figure 4.15) while the Find Nose & Chin module usually excluded those points since it approximates the contours of the nostrils by two circles (Figure 4.14). Also, the human observer usually marked the brightest point of the chin as the tip of it while the automatic scheme located this point as the border between the bright chin and the shadow under it.

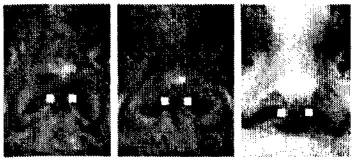


Figure 4.15: Manual localisation of the inner corners of the nostrils

Table 4.13
Distribution of $d(I, I_M)$ for the reference points of the nose and the chin over 88 test images; numbers in the 1st column depict the number of images for which $d(I, I_M) = 0$, numbers in the 2nd column depict the number of images for which $d(I, I_M) = 1$, etc.

	0	1	2	3	4	5	6
left	0	6	37	24	20	1	0
right	0	0	39	17	29	3	0
chin tip	0	9	32	4	39	2	2

Mouth detectors

Two modules, namely *Snake Mouth* and *Curve Fitting of the Mouth*, localise the mouth contour from an input frontal-view image. Their overall properties are summarised in Table 4.14 and Table 4.15.

The Snake Mouth module accepts as input a 24-bit true-colour frontal-view facial image and the result of the detection module MRP to RFM. The box enclosing the mouth, found by the MRP to RFM module, is first segmented from the input image and converted to a grey-scale image by using the original primary colours and the relation Y = 0.299R + 0.587G + 0.114B for the resulting intensity. Further, the red colour component and the green colour component are processed as independent images: the first stage of the algorithm processes the red component while its second stage processes the green component. This is because the red component does not highlight the lips while bearing the information about the mouth-through line whereas the green component bears the information about the lips, which are particularly dark in this component and with suppressed reflections. In the first stage of the algorithm, the mouth-through line is found as a distinct valley in the vertical section of intensity. The minimum of the line with the lowest horizontal integral projection of intensity, representing the centre of the mouth, is found next. A function of the vertical section of intensity through the found minimum is then

Table 4.14
Characteristics of the framework module Snake Mouth

Presented in:	Rothkrantz et al. 1998
Implemented:	C
Applied to:	Frontal-view fac. image
Method:	Colour-based snake fit.
Tested on:	55 frontal-view images
Accuracy:	Medium

Table 4.15
Characteristics of the framework module Curve Fitting of the Mouth

Presented in:	Profijt 1995
Implemented:	C
Applied to:	Frontal-view fac. image
Method:	Fit 3×2nd-degree parabola
Tested on:	280 frontal-view images
Accuracy:	High

created. The minimum of this function is found and the valley is detected by searching in both directions for edge points (zero crossings in the 2^{nd} derivation of intensity starting from a previously found minimum). The mouth-through line is further defined using an altered area-growing algorithm. The algorithm starts from the centre of the mouth and adds points that are 4-connected to the current point and whose intensity is lower than the mean intensity of the previously found valley. In its second stage, the algorithm applies the active contour method proposed by Kass et al. (1987) with the greedy algorithm for minimising the snake's energy function proposed by Williams and Shah (1992). The snake starts in the shape of ellipse whose horizontal axis is the mouth-through line, elongated on both sides for 25%. The method has been tested on a rather small number of test images (merely 55 frontal-view images) and in 85% of these cases a successful snake fitting was obtained. Although this correct recognition rate is not particularly high, the method achieves a rather precise localisation of the mouth (e.g. Figure 4.16) if the snake does not collapse and shrink into a single point. The main drawback of the method is its robustness; it is highly prone to changes in illumination and lighting conditions. The presence of a light source providing rather yellow light increases the successfulness of the algorithm with even 10%. This is because yellow light increases the contrast of the lips in the green component and conversely decreases their contrast in the red component. Another disadvantage is that the processing of the module is highly time consuming (some 2 minutes on average).

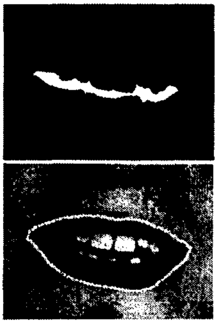


Figure 4.16: The mouth-through line and the mouth contour localised by the Snake Mouth module

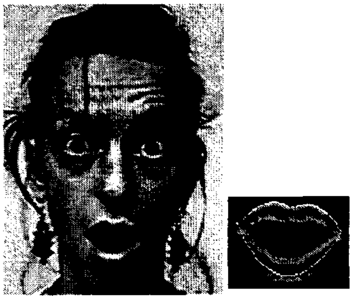


Figure 4.17: The mouth contour localised by Curve Fitting of the Mouth

The Curve Fitting of the Mouth module accepts a grey-scale frontal-view facial image and the result of the detection module MRP to RFM. The box enclosing the mouth, found by the MRP to RFM module, is segmented from the input image and then filtered with a two-dimensional Gaussian low-pass filter. In the binarised image, the lowest highlighted pixel is then selected as the starting point for following the boundary. A pixel directly connected to the current pixel, representing a zero crossing of the 2^{nd} derivative function of the mouth image, continues the mouth boundary. The points where the conjunction of the lips ends and changes in a disjunction are marked as mouth corners. A refined estimate of the mouth shape is then obtained by fitting two 2nd-degree parabolas on the upper lip and a 2nd-degree parabola on the lower lip. The 2nd-order least-square model algorithm is used to find the best relation between the points of the extracted mouth contour and the parameters of each of the parabolas. A typical result of the module is shown in Figure 4.17.

Table 4.16
Distribution of *d(p, p_M)* for the reference points of the mouth over 280 test images; numbers in the 1st column depict the number of images for which *d(p, p_M)* = 0, numbers in the 2nd column depict the number of images for which *d(p, p_M)* = 1, etc.

	0	1	2	3	4	≥5	≥8
left	8	13	57	138	54	9	1
c-top	17	77	49	97	37	3	0
right	5	19	82	126	41	5	2
c-bot.	22	27	61	116	52	2	0

The performance of the algorithm has been tested on a set of 280 frontal-view facial images of 20 different persons. The images were used in their original resolution, i.e. 720 by 576 pixels. First, all of the images were given to a human observer. Using Adobe PhotoShop and a mouse device, the observer pointed to the exact location of the four utilised reference points: the left mouth corner, centre of the upper lip, right mouth corner, and centre of the lower lip. Then the performance of the detection scheme was evaluated by calculating the block distance *d(p, p_M)* between the estimated reference points $p = (x, y)$ and the manually located reference points $p_M = (x_M, y_M)$. The performance of the algorithm is shown in Table 4.16. As can be seen, in most of the images the localisation error of each reference point is approximately 3 pixels. This proves that the detection scheme employed by the Curve Fitting of the Mouth is sufficiently accurate for an automatic detection of subtle changes in the appearance of the mouth.

Mouth classifiers

Examining children's or caricature drawings may lead to an interesting conclusion. A sad mouth or a smile can be shown using only a single drawing line that still perfectly reflects the intention of the artist (Figure 4.18). The main reason for this is that the pertinent mouth expressions affect primarily the facial position of the mouth corners. This leads further to the conclusion that an appropriate representation of such mouth expressions may be the information about the average edge intensity and direction in the corners of the mouth. If on average the edge is "going up", the mouth expression could be interpreted as "smiling". If on average the edge is "going down", the mouth

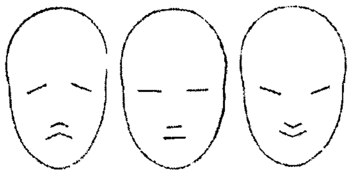


Figure 4.18. People easily detect the facial features pattern from the line drawings of faces

expression could be interpreted as "sad". This idea has been implemented in the form of an ANN classifier of "vertical" mouth expressions that applies fuzzy reasoning for edge detection (Wojdel and Rothkrantz 1998). If on average the edge is "protruding", the mouth expression could be interpreted as "stretched". If on average the edge is "shrinking", the mouth expression could be interpreted as "puckered". This idea has been implemented as a rule-based classifier of "horizontal" mouth expressions that also applies fuzzy reasoning for edge detection (Pantic et al. 2001b). Thus, both the *Vertical ANN Mouth Classifier* and the *Horizontal Rule-based Mouth Classifier* apply fuzzy reasoning for edge detection and while the Vertical ANN Mouth Classifier employs a neural-network-based classification, the Horizontal Rule-based Mouth Classifier applies a rule-based classification of the mouth expression in an input frontal-view image. The processing of only one classifier, namely the original (vertical) ANN mouth classifier, is explained below. The overall properties of each of the classifiers have been summarised in Table 4.17 and Table 4.18.

Table 4.17
Characteristics of the framework module Vertical ANN Mouth Classifier

Presented in:	Wojdel-Rothkrantz '98
Implemented:	C++
Applied to:	Frontal-view fac. image
Method:	Fuzzy reasoning and a NN-based classification
Tested on:	100 frontal-view images
Accuracy:	High

Table 4.18
Characteristics of the framework module Horizontal Rule-based Mouth Classifier

Presented in:	Pantic et al. 2001b
Implemented:	C++
Applied to:	Frontal-view fac. image
Method:	Fuzzy reasoning and rule-based classification
Tested on:	120 frontal-view images
Accuracy:	High

The processing of the module starts with segmenting the mouth region from the original input frontal-view image based on the result of the detection module MRP to RFM. Then a fuzzy reasoning for edge detection is performed based on two characteristics of the gradient, namely, that the gradient value corresponds with local steepness of the function and that the function is locally symmetrical along the gradient direction. The basic idea of fuzzy reasoning for edge detection originates with Law et al. (1994). Still, the two approaches differ: the main information in the detection scheme of the Vertical ANN Mouth Classifier is the direction of the gradient rather than its value.

The fuzzy reasoning proceeds as follows. The numerical values representing the symmetry and steepness level are first fuzzified into the labels *low*, *medium* or *high* and then passed to the reasoning part of the process. The reasoning part is based on nine rules, such as "if the steepness is high and the symmetry level is high then the edge intensity in this point is high". This part results in the labels *low*, *medium* and *high*, which depict the edge intensity in a given point. The information about the direction of the mouth symmetry axis (the axis is determined based on the result of the detection module MRP to RFM) is used to obtain the information about both, the intensity and the direction of the edge in a given point. Combining the intensity and the direction of the edge in a given point results in a vector representation of that point. The obtained vector field for the whole mouth region is then averaged and sampled in a fixed number of regions. In the case of the Vertical ANN Mouth Classifier, Wojdel and Rothkrantz (1998) used a rectangular-grid decomposition of the mouth region with 10 columns and 5 rows of average edge direction vectors. The resulting 50 vectors (100 values) are further classified by a 100×6×4×4×3 back-propagation neural network.

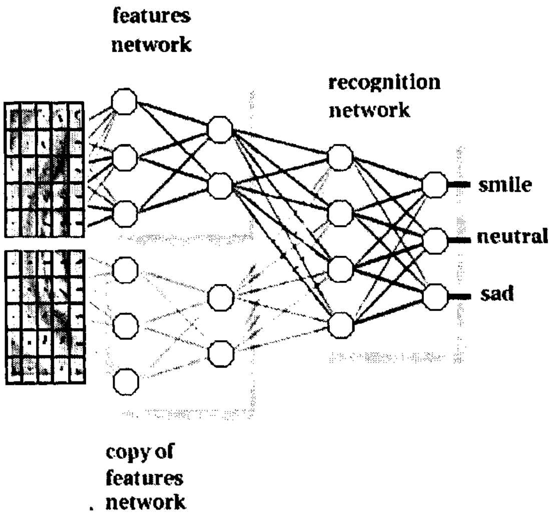


Figure 4.19: NN-architecture of the fuzzy classifier of mouth expressions

The used network layout, illustrated in Figure 4.19, reflects the vertical symmetry of the mouth. The implemented architecture contains two 50×3×2 "features" networks set in parallel (one for each side of the mouth) whose output is passed further to a 4×3 "recognition" network. The output of the network is a singular classification of the shown mouth expression – in the smile, neutral or sad categories. Both features networks should carry out the same task and therefore, they can be implemented as two copies of the same network. In that case, the error is propagated within the single network as well as from the recognition network to both features networks. This speeds up the training process and results in better generalisation properties (Wojdel and Rothkrantz 1998).

To evaluate the method, a set of 100 frontal view facial images has been used. The images were given first to a human observer who classified the images according to the appearance of the mouth into one of the three used categories. Then, in each experiment, ten images were randomly chosen as the test set. The remaining 90 images were processed first by the fuzzy part of the algorithm and then passed to the network as the training set. In each experiment the network achieved full 100% recognition level for both the training and the test set of images. The training took 60 epochs on average. Changes of the average error of the network response during the training is illustrated in Figure 4.20. The average response error on the test set is calculated as 0.08.

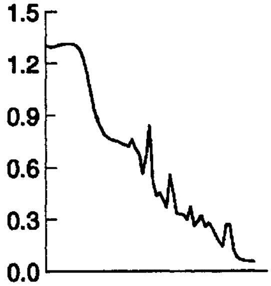


Figure 4.20: Average error in the training epoch

It has not yet been proven whether the proposed method is sufficiently sensitive. The method uses only some average properties of the image, which do not necessarily depict subtle differences between various mouth expressions. Still those fine changes in mouth appearance are crucial for a proper (emotional) interpretation of mouth expressions. The method proved quite efficient, however, as a check facility. Overall correctness of the results of the mouth detectors integrated into the Facial Data Extractor can be easily checked on the basis of the results obtained by the fuzzy classifiers. Using a simple set of rules, the output of a fuzzy classifier can be compared with the properties of the mouth contour localised by a mouth detector. Rules such as "if *smile* then the corners of the mouth are up" and "if *stretched* then the mouth is elongated" extend the fuzzy classifiers of mouth expressions, which in that way form automated tools for checking the results of the mouth detectors integrated into the Facial Data Extractor (see section 5.6).

4.4 Key challenges for future research

The utilised head-mounted camera device (Figure 4.2) facilitates the acquisition of visual facial expression data that are highly suitable for the purposes of behavioural science in the sense that a high image resolution and the presence of the face in an acquired image are ensured. However, the employed device has its drawbacks as well. The device is heavy and reduces the freedom with which the subject can move the head or move around. The subject is required to remain seated and to move the head rather slowly. A quick body movement may cause a displacement of the device and, in turn, a change of the viewing angle which will produce many false conclusions and thus make the result of the system useless. Therefore, the key challenge in the field of selecting and arranging the sensors for monitoring facial expressions for the purposes of behavioural science relates to finding strategies for employing a fixed camera while maintaining the high resolution of the acquired images and as wide a field of view as possible. As noted in section 4.1, a single camera can pan and zoom under computer control to follow faces as they move. This strategy can, in effect, provide a very wide field of view at a high resolution while keeping data rates and computational loads low. However, this strategy requires a fast image analysis in order to determine where the camera must be oriented on a moment-to-moment basis. It also requires robust analysis techniques for locating the head in the observed scene, for dealing with rigid head motions and various viewing conditions, and for dealing with changes in illumination due to changes in the subject's body position in respect to light sources. For future developers of ISFER, this suggests moving towards research of real-time purposive image processing in the field of active vision.

The design and implementation of the Facial Data Extractor as a portable, malleable, interactive, user-friendly platform for facilitating hybrid facial-feature detection in static frontal-view or dual-view facial images introduces many benefits. First of all, introducing redundant facial expression data by concurrently applying multiple facial feature detectors on a dual-view facial image ensures that a more accurate and more complete set of detected facial features is obtained than with any of the conventional facial-feature detection schemes. Second, the design and implementation of the platform enable the current and future developers to implement, edit, test and integrate new facial feature detectors independently of both the programming language used to implement new detectors and the operating system worked on.

Nevertheless, the Facial Data Extractor and the integrated facial feature detectors have their drawbacks as well. The execution of a Java-implemented application is time consuming. This is not a crucial issue for the current version of the Facial Data Extractor since the execution of some of the integrated feature detectors is a dozen times more time consuming. Although this might form a serious drawback once the currently integrated detectors are replaced with some advanced real-time facial-feature detection techniques, it is likely that the current platform will suffice for the immediate needs of the researchers at the Knowledge Based System group. On the other hand, in order to account for the time-consuming execution of the individual feature detectors integrated into the Facial Data Extractor, future developers of ISFER should investigate the implementation on parallel hardware, which could speed up the processing of the system as a whole.

A more important issue concerns the fact that the Facial Data Extractor performs facial-expression-information extraction from static images. It has been proven, however, that such extraction is sensitive to noise and changes in illumination. The key challenges in the field of detection of the facial features in static facial images are therefore the very same problems which are still considered generally unsolved in the field of image processing. For future developers of ISFER, this means that investigations towards a robust detection of the face and its features are necessary, despite the changes in viewing and lighting conditions and the distractions like glasses, birth marks, facial hair, and self-occlusions such as blinking.

Finally, integrated new modules for facial-feature tracking from image sequences would facilitate an automatic facial expression analysis in static facial images as well as in facial image sequences. If performed in real-time, this would enlarge the applicability of the system to fields as various as performance monitoring, stress monitoring, animation, and advanced HCI, in which the extraction of spatial and temporal facial expression information on a moment-by-moment basis is required. More importantly, this would facilitate the recognition of fast subtle facial changes such as blink and wink and make the system suitable for behavioural science investigations of the face from video imagery.

5 Facial action coding

Automating the process of facial signals coding would be enormously beneficial... In behavioural science, facial expression is an important variable for studies on human interaction and communication; is a focus of research on emotion, cognition, and development of infants and young children... Measurement of facial expressions is important for medicine, neurology, neurophysiology, psychiatry, political science and economics, linguistics... To automate coding of facial expressions would advance research in diverse domains.

(Golomb and Sejnowski 1993)

An automated system for facial expression analysis from digitised facial images should translate the automatically extracted facial features into a description of the encountered facial expression (see the discussion on the issue in sections 2.1 and 2.2). Usually one expects the description that is generated automatically to be identical, or at least very close, to a human's description of the displayed facial expression. But which kind of the human-like description of the shown facial expression should be actually generated generally depends on the specific application domain (and/or context) in which the intended facial expression analyser is to be employed.

The first section of this chapter gives an introduction to the Facial Action Coding System (FACS), which is probably the most prominent and commonly used method in behavioural science investigations of the face for measuring facial activity and describing facial expressions in terms of this activity. This section also summarises the potential benefits of automating facial action coding and indicates the wide variety of application areas where benefits could accrue from an automatic system like ISFER. The rest of this chapter is concerned with the second part of ISFER, that is, with the Facial Action Encoder (see Figure 2.25). The Facial Action Encoder is a rule-based expert system that converts the contours of the facial features localised automatically by the Facial Data Extractor part of the system (chapter 4) into a quantified FACS-coded description of the encountered facial expression. The architecture of the Facial Action Encoder is described in section 5.2. Three main parts of the Facial Action Encoder can be distinguished: the pre-processing data evaluator, the data analyser, and the post-processing data evaluator, each of which performs a certain task (function). The pre-processing data evaluator makes the best possible selection of the redundantly localised contours of the facial features, evaluates the certainty of the selected data, and represents those data in terms of a point-based face model. The utilised face model is discussed in section 5.3 while the processing of the pre-processing data evaluator is explained in section 5.4. The data analyser of the Facial Action Encoder performs quantified facial action coding as applied to automated FACS coding and outputs a description of the displayed facial expression in terms of 32 quantified AU codes. The processing of the data analyser is described in section 5.5 (a full list of rules used for the coding and quantification of AUs from an input dual-view facial image is provided in Appendix B). Based on a statistical prediction, the post-processing data evaluator optionally adjusts the AU-coded description of the shown facial expression which has been generated by the data analyser part of the Facial Action Encoder. The post-processing data evaluator is presented in section 5.6. Finally, the benefits and the limitations of the proposed technique for automatic facial action coding in static images are discussed in section 5.7. Some guidelines for enhancing the proposed method are also summarised. For a detailed algorithmic representation of the processing of the Facial Action Encoder, readers are referred to Appendix A.

5.1 Facial action coding

The human face is involved in an impressive variety of different activities. It houses the majority of our sensory apparatus: eyes, ears, mouth and nose, allowing the bearer to see, hear, taste and smell. Apart from these biological functions, the human face provides a number of signals essential for interpersonal communication in our social life. The face houses the speech production apparatus and is used to identify other members of the species, to regulate the conversation by gazing or nodding, to interpret what has been said by lip reading, and to understand somebody's affective state and intentions on the basis of the shown facial expression. Personality, attractiveness, age and gender can be also seen from someone's face. Thus the face is a multi-signal sender/receiver capable of tremendous flexibility and specificity. In general, the face conveys information via four kinds of signals (Ekman 1978):

1. *Static facial signals* represent relatively permanent features of the face, such as the bony structure, the soft tissue, and the overall proportions of the face. These

signals contribute to an individual's appearance and are usually exploited for person identification (Bruce et al. 1992).

2. *Slow facial signals* represent changes in the appearance of the face that occur gradually over time, such as development of permanent wrinkles and changes in skin texture. These signals can be used for assessing the age of an individual. Note that these signals might diminish the distinctness of the boundaries of the facial features and impede recognition of the rapid facial signals (see below).
3. *Artificial signals* are exogenous features of the face such as wearables like glasses and cosmetics. These signals provide additional information that can be used for gender recognition. Note that these signals might obscure facial features or, conversely, might enhance them.
4. *Rapid facial signals* represent temporal changes in neuromuscular activity that may lead to visually detectable changes in facial appearance, including blushing and tears. These signals are responsible for facial expressions.

The main consideration here is rapid facial signals. These movements of the facial muscles pull the skin, causing a temporary distortion of the shape of the facial features and of the appearance of folds, furrows, and bulges of skin. The changes in facial appearance caused by the facial muscle activity are usually brief, rarely lasting more than five seconds or less than 250ms (Ekman and Friesen 1978). The common terminology for describing the changes in facial appearance refers either to culturally dependent linguistic terms indicating a specific change in the appearance of a particular facial feature (e.g. frowned eyebrows) or to the linguistic universals describing the activity of specific facial muscles that caused the observed facial appearance changes. In these linguistic universals the muscles may be designated either by their Latin names or by Action Units (AUs) as proposed in Ekman and Friesen's Facial Action Coding System (FACS, 1978). In general, psychologists favour the usage of linguistic universals (although those might be sometimes a burden to understanding the discussed issue, they can help to avoid misunderstandings caused by usage of culturally dependent linguistic terms). For example, terms like smile, smirk, frown, sneer, etc. might (and probably will) be interpreted differently in different cultures. Further, such terms are imprecise as they may refer to a variety of different muscular actions and their intensities. In the rest of the text, the usage of linguistic universals is favoured.

There are several methods for (linguistically universal) recognition of facial changes based on the facial muscular activity (for a review of 14 such techniques see Ekman 1982b). From those, FACS is the best known and most commonly used system for facial action coding. It is being used by most (if not all) researchers who are currently working on automating facial action coding from facial static images or image sequences (for a further discussion on this issue see also chapter 2, Bartlett et al. 1999, Donato et al. 1999). Following this trend, the first stage of automatic facial expression recognition presented in this thesis concerns automatic facial action coding in static facial images applicable to automated FACS coding.

Facial Action Coding System (FACS)

As mentioned in section 2.1, FACS is a system designed for human observers to visually determine how the activation of each facial muscle (individually activated or in a combination with other facial muscles) changes the appearance of the face. The changes in facial appearance are measured with FACS in terms of Action Units (AUs) rather than in terms of muscular units due to two reasons. First, in some cases, activation of several different muscles produces a single facial appearance change (e.g. to lower the eyebrow and to draw the eyebrows together, three muscles are activated). In FACS the muscles involved in the production of a single change in facial appearance were combined into one specific AU (e.g. AU4 in the case of the drawing of the eyebrows together). Second, in some cases, the same muscle is involved in the production of various changes in facial appearance. For instance, the outer portion of the frontalis muscle can be activated independently of the activation of the inner portion of this muscle, causing merely the outer portion of the eyebrow to rise. In FACS, the activity of such muscles is separated into a few AUs, each of which represents a distinct change in facial appearance (e.g. AU1 and AU2 in the case of the raising of the inner, respectively the outer, portion of the eyebrows).

Table 5.1
List of upper face facial actions defined in FACS

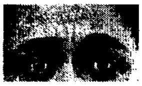
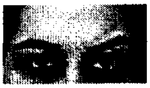
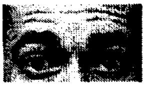
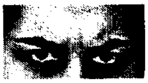
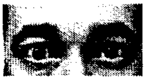
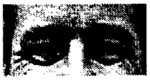
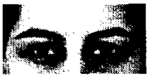
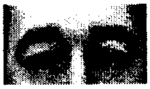
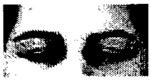
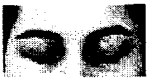
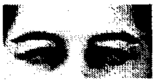
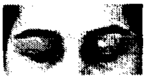
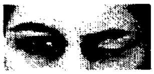
	AU1: Raised inner eyebrow		AU2: Raised outer eyebrow
	AU1 + AU2: Raised eyebrows		AU4: Lowered eyebrow Eyebrows drawn together
	AU5: Raised upper eyelid		AU6: Raised cheek Compressed eyelid
	AU7: Tightened eyelid		AU41: Drooped eyelid
	AU42: Slit – eyes are opened just a bit ("slit eyes")		AU43: Closed eyes
	AU44: Squinted eyes		AU45: Blink – same as AU43 but lasting less than ½ second
	AU46: Wink		

Table 5.2
List of lower face facial actions (AU8 to AU29) defined in FACS

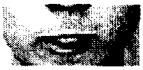
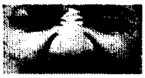
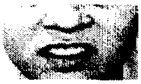
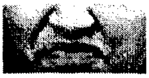
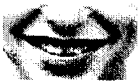
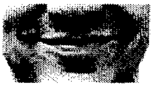
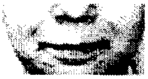
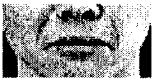
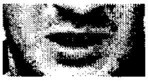
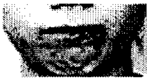
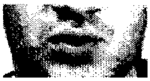
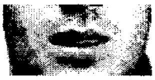
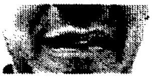
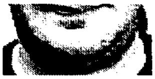
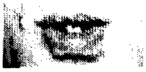
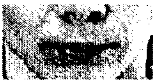
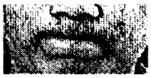
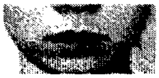
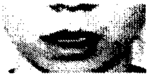
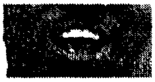
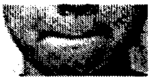
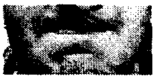
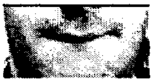
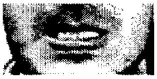
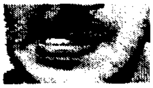
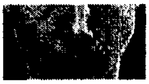
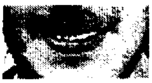
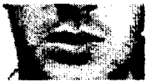
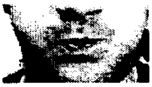
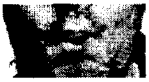
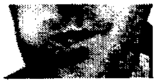
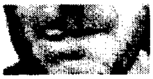
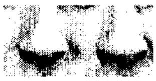
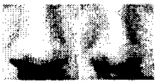
	AU8: Lips towards each other		AU9: Wrinkled nose
	AU10: Raised upper lip		AU11: Deepened nasolabial furrow
	AU12: Lip corners pulled up		AU13: Lip corners pulled up sharply
	AU14: Dimpler – mouth corners pulled inwards		AU15: Lip corners depressed
	AU16: Lower lip depressed		AU17: Chin raised
	AU18: Puckered lips		AU19: Tongue shown
	AU20: Mouth stretched horizontally		AU21: Neck tightened
	AU22: Lip funneler – as when pronouncing "flirt"		AU23: Lips tightened
	AU24: Lips pressed		AU25: Lips parted
	AU26: Jaw dropped		AU27: Mouth stretched vertically
	AU28: Lips sucked into the mouth		AU28t: Upper lip sucked into the mouth
	AU28b: Bottom lip sucked into the mouth		AU29: Jaw pushed forward

Table 5.3
List of lower face facial actions (AU30 to AU39) defined in FACS

	AU30: Jaw sideways		AU31: Jaw clenched
	AU32: Bitten lip		AU33: Blow
	AU34: Cheeks puffed out by the air forced into the mouth		AU35: Lip corners sucked into the mouth
	AU36: Bulge produced by the tongue		AU37: Wiped lips
	AU38: Nostril wings flared out (not flared and flared nostrils)		AU39: Nostril wings compressed (uncompressed and compressed nostrils)

There are 44 different AUs defined in FACS that account for the changes in facial expression (Tables 5.1 to 5.3) and 14 other actions grossly describing changes in gaze direction and head orientation. Along with the definitions of various AUs, FACS also provides the rules for AUs' encoding in a full-face photograph and in the case of five AUs, namely AU5, AU10, AU12, AU15, and AU20, it provides an option to score intensity on a 3-level scale (low, medium, high). Using these rules, a *FACS coder* (i.e. a human expert having a formal training in FACS coding of facial images) decomposes an observed facial expression into the specific AUs that produce the expression.

Although FACS is the most prominent method for measuring facial expressions in behavioural science, a major impediment to its widespread use is that its manual application is time consuming in addition to the time required to train human coders. Each minute of videotape takes approximately one hour to score and it takes 100 hours of training to achieve minimal competency on FACS. Automating FACS would not only make it widely accessible as a research tool, it would also increase the coding speed and improve the precision and reliability of facial measurements.

Applications of automated FACS

The main objective for the development of ISFER presented in this thesis is to provide an automated tool for behavioural science investigations of the face. Yet an automated system like ISFER, which outputs facial action codes, would also provide a useful tool for other basic sciences, medicine, and computer science.

Investigation of human facial reactions is crucial for research on human interaction and communication (Mehrabian 1968, Bruce 1992), research on emotions (Ekman 1982a), and for research on the development of infants and young children (Salovey and Mayer 1990, Goleman 1995, Sigman and Capps 1997). In anthropology, the cross-cultural perception and production of facial expressions is a topic of considerable interest (e.g. Ekman 1980, Matsumoto 1990, Russell 1994, Ekman 1994). For political science and economics, facial expression analysis is important in studies on negotiations and interpersonal influence (McHugo et al. 1985). In linguistics, co-articulation of spoken words and lip movements (Perkell 1986), relations between facial muscles and the soft palate in speech (van Gelder and van Gelder 1990), and lip reading, are all important for speech recognition (Adjoudani and Benoit 1995, Meier et al. 1996, Kober et al. 1997, Yang et al. 1998). All of these research fields would benefit from an automated, inexpensive, reliable and rapid facial measurement tool. In brief, such a tool would revolutionise these fields by raising the quality of research in which reliability and precision are currently nagging problems and by shortening the time necessary to conduct research that is now lengthy due to the time-consuming manual FACS coding of research material.

Many disorders reported in medical files, particularly in neurological and psychiatric diagnoses, involve aberrations in expression, perception, and interpretation of facial actions. Coding of facial actions is thus necessary to assess the effect of the encountered disorder, to understand the disorder, and to devise strategies to overcome the limitations imposed by the disorder. For instance, in the psychiatric domain, schizophrenia and psychosomatic illnesses blunt the expressiveness of patients (Steimer-Krause et al. 1990). Also, central and sensory impairments like autism and schizophrenia can also cause lack of the ability to "read" and interpret the shown facial expression (Mandal and Palchoudhury 1986, Sigman and Capps 1997). In neurology, analysis of facial expressions may provide evidence for the location and type of brain lesions. The research on the topic has proven that, for example, brainstem damage may lead to emotional lability (Hurwitz et al. 1985), and Parkinson's disease is accompanied by reduction of spontaneous facial activity (Buck and Duffy 1980). Automated methods for assessing facial responses could provide increased reliability, sensitivity, and precision needed to uncover psychiatric and neurological disorders based on facial signs displayed by patients and could lead to new insights and diagnostic methods.

Automated systems that could monitor and detect facial signals will greatly enhance the state of the art in computer technology and have great commercial potential. Such a system combined with facial expression interpretation in terms of labels like "did not understand", "does not agree", "no attention", and "approves" could be employed as a tool for monitoring human reactions during video conferences. When combined with automatic tools for speech recognition and synthesising facial expressions and speech, an automated facial action encoder facilitates development of *talking heads* and *virtual actors* 'inhabiting' a virtual world (e.g. Thalmann et al. 1995, 1998, 2000). Within a virtual environment the users are represented by personalised *avatars* and they can meet, chat, and acquire information in a more natural way than currently available tools like ICQ and Internet search engines can provide. In addition, talking heads and personalised avatars could significantly enhance international, commercial and political interactions. Finally, an automated facial action encoder forms a front-end of future 4th-generation anthropomorphic, perceptual, multi-modal human-computer interfaces (for a discussion on this topic see chapter 8 and for reviews on the topic see Pantic and Rothkrantz 2001a, Pentland 2000, Sharma et al. 1998, Waibel et al. 1995).

Other markets for automated facial action encoders combined with automatic facial expression interpretation facilities include specialised areas in professional sectors (Golomb and Sejnowski 1993). For instance, monitoring and interpretation of facial signals is important to lawyers, police, security and intelligence agents, who are often interested in issues concerning deception or attitude. Automated facial reaction monitoring could form a valuable tool in these situations, as now only informal interpretations are used. Automatic assessment of boredom, inattention, and stress, could be of high value in preventing critical situations in the hazardous (working) environments such as aircraft control cockpit, air traffic control tower, space flight operation chambers, nuclear plant operation chamber, or simply a vehicle like a car, truck, or train. An advantage of machine monitoring is that human observers need not to be present to monitor – a personalised artificial observer could provide prompts for better performance.

The different examples listed here illustrate a great potential for an automated facial action encoder. However, the reader should be aware that some, perhaps the most important applications and benefits of such an automatic tool could be the ones that nobody has imagined yet. Once this technology is available, it could enhance our common understanding of human facial behaviour, it could change some of the taboos considered as ground truth for many decades, and hence it could give rise to the emergence of a totally different set of applications. Nonetheless, the reader should be aware of the likelihood that automatic recognition of the full range of facial behaviour still lies in the relatively distant future. As discussed in chapter 2, none of the automatic facial action encoders proposed in the literature up to date is capable of performing this task. Although the automatic facial action encoder discussed in the rest of this chapter performs better than other existing facial action encoders, it is still not capable of encoding the full range of facial behaviour. The relative advantages of ISFER's facial action encoder, its disadvantages, and some general recommendations for tackling automating facial action encoding are discussed further in section 5.7.

5.2 Architecture of the Facial Action Encoder

For its utility in the various application domains outlined above, automatic facial expression recognition as applied to automated FACS coding has attracted the interest of several computer-vision researchers (see Table 2.7). Yet none of the methods reported up to date is sufficient for describing the full range of facial behaviour, that is, for encoding all 44 individual AUs. In addition, none of the systems presented in the literature up to date deals with quantification of the facial action codes. Yet, for investigations of the facial behaviour itself, such as studying the difference between a person's genuine and simulated affective state, an objective, personalised, and detailed measure of any possible facial activity is needed. Therefore, the design of the Facial Action Encoder was aimed at automating a part of the FACS scoring being as copious as possible and resulting in a highly reliable quantified AU-coded description of the examined facial expression.

Design requirements

As stated before, the main goal of the development of the Facial Action Encoder part of ISFER was to enhance the state of the art in facial expression recognition applicable to automated FACS coding. The aim was to develop a fully automated and robust tool for facial expression analysis from static facial images in terms of:

1. an extensive set A of individual AUs,
2. a level of the activation intensity $\boldsymbol{i} \in [0, 100]$ assigned in a subject-dependent manner to each encoded AU for which such a quantification is reasonable (see Table 5.9), and
3. any combination of AUs from A having any intensity level $\boldsymbol{i}$.

From an engineering point of view, the intended tool should be both efficient and effective. This means that the desirable features of the tool are the following:

- *The processing of the tool is easy to construct from the input data.* In the case of the Facial Action Encoder, the input data are the files which result from the processing of the Facial Data Extractor and contain the localised contours of the facial features. In order to achieve easy construction, the aim was to design the Facial Action Encoder as an integration of various simple processes (Figure 5.1).
- *The tool is easy to update.* In the case of the Facial Action Encoder this is of crucial importance since the intended users are psychologists and other subjects having usually no technical background. In order to achieve easy update, the aim was to design the Facial Action Encoder such that it facilitates facial action encoding from images of subjects of both sexes and any age. Otherwise, if the recognition of displayed facial actions would depend on physiognomic variability of the observed person, the user would have to update the recognition mechanism each time a novel subject is to be analysed. Hence, the tool would be

highly inefficient. On the other hand, quantifying the encoded facial actions is person dependent since everybody displays a certain facial action with a different maximal intensity. Thus, the aim was to design the Facial Action Encoder such that it performs a generic facial action classification and then adapts (in a facile way) to a particular individual in order to quantify the encoded AUs.

- *The tool is easy to use.* Once more, this is of particular importance since the potential users of the Facial Action Encoder are persons without a high level of technical knowledge. Therefore, the aim was to design the Facial Action Encoder such that it does not require the user to be in control at all times.
- *The tool itself is efficient to store.* In order to achieve this, the aim was to split the code efficiently in functions. Nevertheless, the major impediment to achieving efficient storage of the Facial Action Encoder is not the size of its code but the size of the utilised databases (Figure 5.1). Large databases are coupled with high memory/ storage requirements and long retrieval times. While the database of extreme model deformations is small (it contains 20 variables, see Table 5.9 and section 5.5 for a further discussion), the database of all encountered facial expressions may become large since it should keep records of all expressions ever encountered while monitoring a particular subject for whom it has been defined (section 5.6). Though the aim was to organise the database of all encountered facial expressions efficiently (section 5.6) – such that it supports short retrieval times – its expansion is not controlled. In consequence, depending on the actual frequency with which a particular person is subjected to the system's facial analysis and on the actual variety of that person's facial expressions, the database of all encountered facial expressions defined for that person might impose high storage requirements. Though this forms a shortcoming of the Facial Action Encoder part of ISFER (see also section 5.7), one should bear in mind that this problem becomes less and less significant as the cost of computer memory drops (e.g. Hassler 2001).

Functional design

The Facial Action Encoder part of ISFER represents a first step towards an automated personalised tool for facial expression recognition in static facial images. This step was achieved by fulfilling the design requirements listed above. The Facial Action Encoder performs five functions (Figure 5.1, Appendix A):

1. *Function F1* scavenges the incoming set of files containing the contours of the facial features localised by the detectors integrated into the Facial Data Extractor part of the system, from the files holding a *singularity* (i.e. holding a single point as the localised facial feature contour; see section 5.4).
2. *Function F2* reduces the incoming data to a set of files containing merely the relevant points of the utilised face model (see sections 5.3 and 5.4).
3. *Function F3* selects the best among the redundant data stored in the files and evaluates the certainty of the selected data (see section 5.4).

4. *Function F4* performs a person-independent facial expression recognition in terms of 32 AU codes and a person-dependent quantification of those codes (see section 5.5).
5. *Function F5* deals with partial data by performing person-dependent reasoning about the typicality of the encountered facial expression (see section 5.6).

The Facial Action Encoder is a rule-based expert system that converts an analytic description of the encountered facial expression, given in terms of the spatially sampled facial features' contours, into a quantified AU-coded description of the input expression. In general, the employed rules encode three different kinds of knowledge.

1. The knowledge about the basic anatomy of the human face: the facial point's stability and/or degree of freedom, overall facial proportions, anatomically possible changes in the facial appearance, etc.
2. The knowledge about the facial muscle activity: the relationship between a certain muscle activation and a certain facial appearance change, classification of facial changes into unilateral/bilateral, facial changes' co-occurrence rules, facial changes overruling rules, etc. This type of knowledge has been acquired from the FACS in a straightforward manner (see also section 5.3).
3. The knowledge about the currently observed subject: the extreme facial changes and the typicality of different facial expressions. This subject-oriented type of knowledge is acquired for each novel subject. The personalised data are stored further in two databases used by the Facial Action Encoder, namely the database of extreme model deformations (for a further discussion see section 5.5) and the database of all facial expressions displayed by the current subject (section 5.6).

As already mentioned in section 3.7, the Facial Action Encoder can be viewed as a problem-solving autonomous agent that is a part of a functionally distributed system. It is an "autonomous agent" since it does not require that the user is in control at all times. In fact, as soon as the initial furnishing of the database of extreme model deformations is accomplished and as long as the observed subject remains the same, the processing of the Facial Action Encoder is fully independent of the user's feedback. It is a problem-solving agent, which resembles conventional expert systems since it encodes domain-specific knowledge to achieve the intended functionality – FACS coding of static images – by using a rule-based approach (see sections 5.4 and 5.5). The kernel of the Facial Action Encoder, illustrated in Figure 5.1 as *supervisor*, performs the following agencies:

- It creates action plans depending on the quality of the incoming data.
- It carries out those plans by executing the appropriate procedures on the relevant data.
- It encodes and quantifies facial actions.

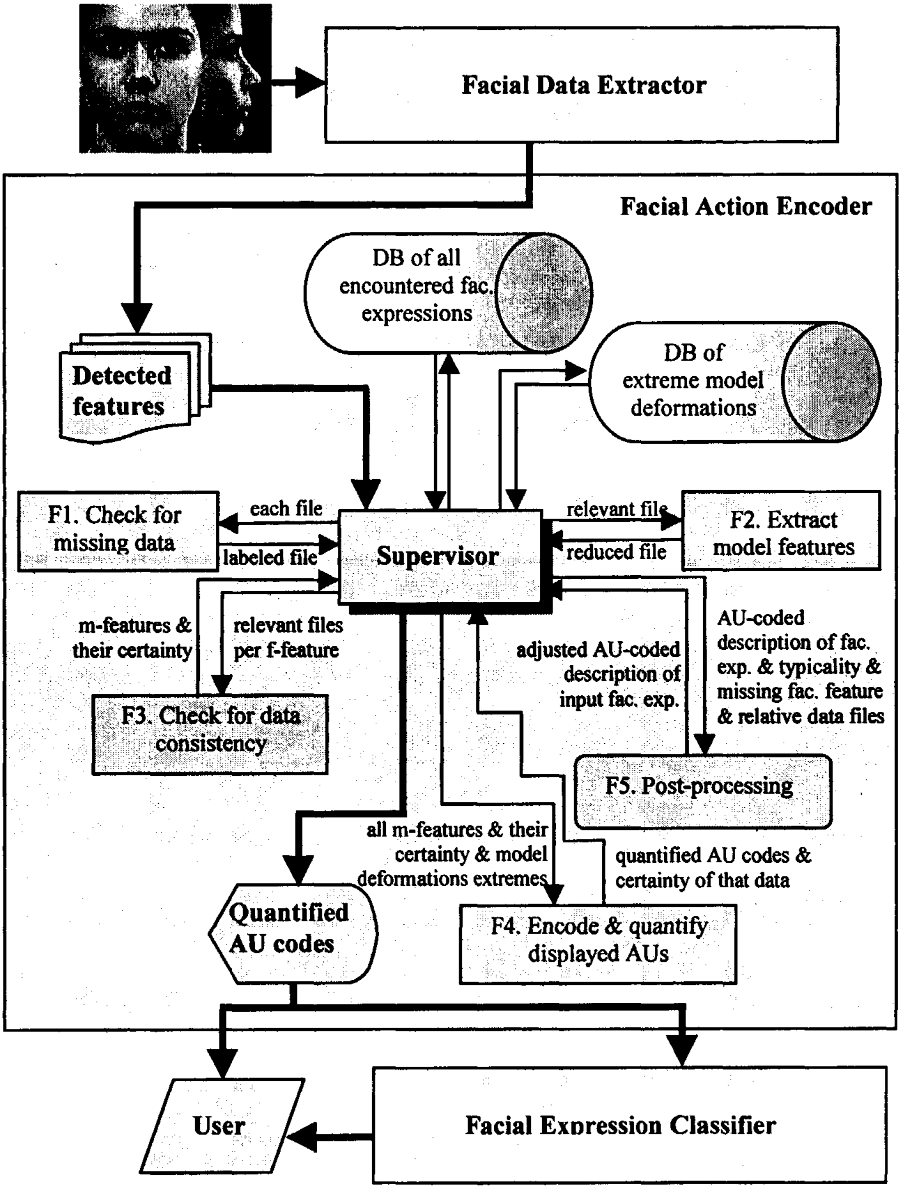

Facial Data Extractor
Facial Action Encoder
DB of all encountered fac. expressions
DB of extreme model deformations
Detected features
F1. Check for missing data
each file
labeled file
Supervisor
relevant file
reduced file
F2. Extract model features
m-features & their certainty
relevant files per f-feature
F3. Check for data consistency
adjusted AU-coded description of input fac. exp.
AU-coded description of fac. exp. & typicality & missing fac. feature & relative data files
F5. Post-processing
all m-features & their certainty & model deformations extremes
quantified AU codes & certainty of that data
Quantified AU codes
F4. Encode & quantify displayed AUs
User
Facial Exnression Classifier

Figure 5.1: Architecture of the Facial Action Encoder part of ISFER

Let us examine how the five functions, listed above and illustrated in Figure 5.1, support the problem-solving behaviour of the Facial Action Encoder. The supervisor receives the input data – a set of files containing the localised contours of the facial features – from the Facial Data Extractor part of the system. The perceived information is then categorised and reduced by *function F1* and *function F2*. The resulting information is a set of files, each of which contains the model points belonging to the detected contour of a certain facial feature. The first (predefined) goal that the supervisor will try to reach is to select per facial feature the best of the redundantly detected contours. To this end, the supervisor classifies per facial feature the reduced information: each class contains just the files related to one particular facial feature. Then the supervisor selects from the action space predefined plans that can be used to reach the active goal. For example, if a certain class contains one or more files, *function F3* will be activated to calculate the certainty of the data stored in each file and then to select the best result. But if a certain class contains no files (i.e. all files containing the detected contour of the relevant facial feature have been previously discarded by function F1), missing data will be substituted by relevant data extracted from the expressionless face of the currently observed subject. The supervisor will carry out this substitution by accessing the appropriate data stored in the *DB of all encountered facial expressions*. The second (predefined) goal that the supervisor will try to reach is to encode and quantify the displayed AUs. The supervisor again selects from the action space predefined plans that can be used to reach this goal. First, the supervisor will retrieve the appropriate data from the *DB of extreme model deformations* and activate *function F4* in order to perform reasoning with uncertainty about the activated AUs and the intensity of that activation. In the case that a currently encountered model deformation exceeds the maximal value for that model deformation stored in the DB of extreme model deformations, the supervisor will access and replace the relevant data in this database. In the case that data about a facial feature have been substituted by the relevant data extracted from the expressionless face of the currently observed subject, the supervisor activates *function F5*, which represents a statistical approach to dealing with partial data based on typicality of the displayed facial expression. Finally, the supervisor accesses once again the DB of all encountered facial expressions and updates the counter for the expression just encoded.

Thus, the Facial Action Encoder might be viewed as a problem-solving autonomous agent. However, note that this is just one way of viewing the architecture of the second part of ISFER. As explained by Moulin and Chaib-Draa (1996), any expert system can be seen as an agent, at least as a reactive agent (as in the case of the Facial Action Encoder) if not as an intentional or a social agent. The reason to discuss the Facial Action Encoder as an agent is not the catalyst behind the recent "agents hoopla" – the accelerating spread of the Internet – which has given rise to an explosion of applications involving intentional and social agents that can manipulate their goals and create new ones in order to search the Internet successfully. The reason lies in the fact that the current functionality of the Facial Action Encoder can be easily enhanced for monitoring a particular information source (e.g. a certain participant in a video conference, a certain patient in a group therapy, a certain student attending a course) and providing an alert if some set of conditions holds (e.g. if a smile is observed). By this, the Facial Action Encoder would represent a *consumer-based problem-solving agent* (Hendler 1999) which would be able to manipulate the incoming information on current behalf of the user. By allowing the user to define his current interest while monitoring facial expressions, the commercial potential of ISFER would be increased: ISFER would embody an application-independent automatic tool for facial expression analysis (see also section 3.7).

Currently, however, the Facial Action Encoder is not able to manipulate its goals and create new ones (e.g. according to the wishes of the user); it achieves a predefined set of goals by selecting from the action space predefined plans that can be used to reach these goals. In other words, it is a reactive agent as any other expert system. Hence, for the sake of clarity and precision, in the remainder of the text the Facial Action Encoder is discussed as a rule-based expert system that performs reasoning with uncertainty on quantified facial actions based on data automatically extracted from static images.

Implementation

As mentioned above, from an engineering point of view, the design requirements for the development of the Facial Action Encoder part of ISFER concern efficiency and effectiveness of the intended tool. In brief, the Facial Action Encoder should be easy to construct and to integrate into ISFER, it should be easy to use by the potential users of the system while placing no constraints on the operating system of the utilised work station, and it should be efficient to store. Hence, the aim was to design an efficient, portable, interactive user-friendly tool for facial expression recognition applicable to automated FACS coding.

Because the Facial Data Extractor part of ISFER has been implemented in Java and because of the availability of the JDK's Abstract Window Toolkit, which is a platform-independent GUI tool builder, Java was perfectly suitable for the development of the Facial Action Encoder. Similarly to the case of the Facial Data Extractor, one might argue that a time-consuming execution of Java code forms a serious drawback of the system. Yet this is of little concern since the time spent by the processor on executing the code of the Facial Action Encoder is rather short compared to the time spent on executing the code of various facial features detectors integrated into the Facial Data Extractor (chapter 4).

5.3 Modelling facial expressions

The internal representation of the visual information that an examined face might reveal is a crucial issue in facial expression analysis. Namely, the utilised face model determines the variety of facial expressions that can actually be recognised by an automated facial expression analyser. As noted in section 2.2, an ultimate goal is the development of a face model which quickly and accurately provides a person-independent unique representation for any expression. If coupled with automated FACS rules, such a universal model would facilitate automatic encoding of the full range of facial behaviour for any person.

Two basic approaches can be taken to address the problems of face modelling and visual facial-expression-information representation:

- *Volume-based approach*: This class of methods attempts to recover the 3D facial information, both geometrical (shape) and photometric (texture), from the sensed data (2D face images). The first work in developing 3D face models was done in early 70s (Parke 1972, 1974; Gillenson 1974) and the field has seen considerable activity in the last couple of years (e.g. Terzopulos and Waters 1993, Thalmann et al. 1998, Eisert and Girod 1998, Decarlo and Metaxas 1999, Malciu and Preteux 2000, Zhang and Kambhamettu 2000). These face representations have the advantage that they can be extremely accurate, reflecting the changes of the face by modelling the properties of facial tissue and muscle actions. But they have the disadvantage that they are often slow, fragile, and that usually they must be trained by hand.
- *View-based approach*: This class of methods attempts to model a face as a whole (holistic methods), as a set of facial features (analytic methods), or as a combination of these (hybrid methods), based on the 2D appearance of the face and without attempting to recover the 3D geometry of the scene. These methods can be further classified into two categories according to the temporal aspect of facial modelling, namely, into methods that model faces from static images and those that model faces from image sequences (see also Table 2.5 and Table 2.6). View-based methods have the advantage that they are typically fast and simple, and that they can be trained directly from the image data. They have the disadvantage that they cannot be used for modelling some of the subtle facial changes (e.g. wiping the lips, clenched jaw) when merely a frontal view is considered. Moreover, these methods may become unreliable when there are many different views that must be considered.

Although we agreed that automating the entire process of facial action coding would be enormously beneficial (see section 5.1), we should recognise the likelihood that the realisation of such a goal might lie in the relatively distant future. A universal face model may be too difficult to develop because it should satisfy all following requirements:

- It should uniquely reflect each and every change in facial appearance – unilateral or bilateral, similar to another change, and caused by activation of any facial muscle.
- It should be person-independent.
- It should be composed of features that can be reconstructed or straightforwardly detected in a 2D facial image or image sequence.

However, the automation of as many of the tedious and time-consuming FACS scoring parts as possible would allow trained human observers valuable time for making the most difficult judgements (on the shown affective state, mood, intention, etc.). Therefore independently of the chosen approach to modelling facial expressions (i.e. the volume-based or view-based approach), efforts should be made to develop a face model that provides a person-independent unique representation for a set of facial expressions that is as broad as possible.

Facial expression modelling in ISFER

As noted in section 2.6, the development of ISFER has been aimed at devising an automatic tool that will serve the purposes of behavioural science investigations of the face. This application domain defined all the requirements for the system's design, including the requirements for the facial expression modelling that the system achieves. In order to facilitate automated analysis of human facial behaviour, ISFER should perform person-independent modelling and unique AU-based recognition of a set of distinct facial expressions as copious as possible. Since the researchers of human facial behaviour use static full-face photographs (rather than movies) as research material, ISFER should be able to analyse facial expressions in static frontal-view facial images automatically. Finally, from an engineering point of view, the desirable features of internal data representation are easy to construct from sensed data, easy to use for the intended application and efficient to store. This means that the features of the face model used by ISFER should be easy to acquire from the contours of the prominent facial features localised by the Facial Data Extractor part of the system, easy to use for encoding and quantification of AU codes, and efficient to store.

The face model utilised by ISFER is a point-based model composed of two 2D facial views, namely the frontal view and the profile view (Figure 5.2). There is a number of motivations for this choice. As discussed in section 2.1, Bassili's (1978) and Bruce's (1986) experiments suggest that the visual properties of the face, regarding the information about facial expression, could be made clear by describing the movements (displacements) of points belonging to the prominent facial features and then by analysing the relationships between those. Hence, taking into consideration that each and every facial expression that can be displayed by the face can be FACS coded, it seems that the rules of FACS could be converted into the rules for automated FACS coding based upon a point-based face model.

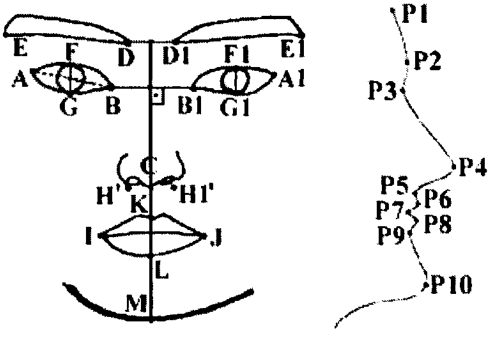


Figure 5.2: Dual-view Face Model

Furthermore, a point-based face model as well as the rules of automated FACS based upon such a model should be easy to validate. The changes in the position of the model points are directly observable. By comparing the deformation of the model and the modelled facial appearance, the validity of the model and the validity of the modelled FACS rules can be visually inspected. Finally, a point-based dual-view face model yields a more realistic representation of an observed 3D face than a single-view face model and avoids inefficiency and manual initialisation of a volume-based face model. In addition, the dual-view face model illustrated in Figure 5.2 has the following characteristics:

1. Since the camera setting ensures that scale- and orientation-invariant images are acquired (at least during a single session; see section 4.1), the model points can be extracted automatically and in a straightforward manner from the contours of the prominent facial features localised by the Facial Data Extractor (Tables 5.4 and 5.6).
2. It is possible to establish a simple and unique mapping between deformations of the frontal-view face model and 22 distinct facial appearance changes (i.e. 22 AUs; see Table 5.5). It is also possible to establish such a mapping between deformations of the profile-view model and 24 distinct facial appearance changes (i.e. 19 distinct AUs, 3 variations of AU28, and 2 variations of AU36; see Table 5.7). Finally, it is possible to combine those mappings and define a simple and unique mapping between deformations of the dual-view face model and 32 distinct facial appearance changes (i.e. 27 distinct AUs, 3 variations of AU28, and 2 variations of AU36; see Table 5.8).
3. It is possible to determine the intensity of the currently encountered AUs by determining the extent of the deviation of the face model fitted to the currently examined facial expression from the face model fitted to the neutral expression of the observed person. In other words, by quantifying the deformations of the face model, quantification of changes in facial expression can be accomplished (see section 5.5, Table 5.9, Figure 5.12).
4. The model points are efficient to store while facilitating automatic analysis of the widest range of facial behaviour reported in the literature up to date (i.e. 32 distinct facial actions and their combinations vs. 16 distinct AUs and their combinations reported by Tian et al. (2001)). In other words, the point-based dual-view face model shown in Figure 5.2 facilitates enhancement of the state of the art in automated facial expression analysis.

The point-based frontal-view face model

The frontal-view face model utilised by ISFER is composed of 21 facial points belonging to the contours of the prominent facial features, namely the contours of eyebrows, eyes, nostrils, mouth, and chin. These points are illustrated in Figure 5.2 and described in Table 5.4.

The degree of freedom of various frontal-view face model points and the manner in which those points are extracted from the facial features' contours, localised by the Facial Data Extractor in the input frontal-view static facial image, are also given in Table 5.4. The information about the degree of freedom of various frontal-view facial points has been acquired from FACS and various studies on human anatomy (e.g. McCracken, 1999). This information represents a part of the knowledge about the basic anatomy of the human face integrated into the system. For example, points B and B1 representing the inner corners of the eyes and points H and H1 representing the inner corners of the nostrils have ***0***-degree of freedom since they are *stable facial points*, meaning that no facial muscle activity can cause a physical displacement of these facial points. Similarly, the facial points having a degree of freedom ≥ ***1*** are so-called *non-stable facial points*, meaning that the activity of certain facial muscle(s) causes a physical displacement of those points. The only exceptions from this rule are points A and A1 representing the outer corners of the eyes. As far as the anatomy of the human face is concerned, these points are stable facial points since no facial muscle activity can cause their physical displacement. Consequently, points A and A1 should have ***0***-degree of freedom assigned to them. Yet, self-occlusions produced by a squint, raised lower eyelid, lowered upper eyelid, or raised cheeks, could interfere with the localisation of the position of these points and result in the detection of a horizontal inwards spatial displacement of these points (Figure 5.3). Hence, ***1***-degree of freedom was assigned to these points in order to take into account commonly encountered self-occlusions of the eyes that affect the measurement of points A and A1 and emerge as a consequence of image processing techniques utilised by the eye-contour detectors of the Facial Data Extractor.

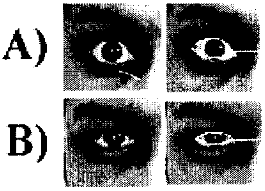


Figure 5.3: A) Original position of point A; B) Self occlusion of A produced by tightened eyes

The frontal-view face model has been generated and then validated through analysis and synthesis, respectively, of linguistic labels used to describe the visual properties of FACS AUs (Ekman and Friesen 1978). For example, the analysis of the label *upward pull of the inner portion of the eyebrow(s)*, which describes activation of AU1, caused the addition of points B, B1, D, and D1 to the model. An observed increase of the distance(s) BD (and/or B1D1) will cause trained FACS coders to conclude that AU1 is activated (see also section 7.2).

Table 5.4
Facial points of the frontal-view face model utilised by ISFER

	Characteristics	Extraction
B	Right eye inner corner, freedom degree: 0	(x_{max}, y) of right eye's contour
B1	Left eye inner corner, freedom degree: 0	(x_{min}, y) of left eye's contour
A	Right eye inner corner, freedom degree: 1 Displacement: horizontal inwards	(x_{min}, y) of right eye's contour
A1	Left eye inner corner, freedom degree: 1 Displacement: horizontal inwards	(x_{max}, y) of left eye's contour
F	Right eye top, freedom degree: 2 Displacement: vertical, horizontal inwards (only if A is displaced)	(x, y_{max}) of right eye's contour
F1	Left eye top, freedom degree: 2 Displacement: vertical, horizontal inwards (only if A1 is displaced)	(x, y_{max}) of left eye's contour
G	Right eye bottom, freedom degree: 2 Displacement: vertical, horizontal inwards (only if A is displaced)	(x, y_{min}) of right eye's contour
G1	Left eye bottom, freedom degree: 2 Displacement: vertical, horizontal inwards (only if A1 is displaced)	(x, y_{min}) of left eye's contour
D	Right eyebrow inner corner, freedom degree: 2 Displacement: vertical, horizontal inwards	(x_{max}, y_{min}) of right eyebrow's contour
D1	Left eyebrow inner corner, freedom degree: 2 Displacement: vertical, horizontal inwards	(x_{min}, y_{min}) of left eyebrow's contour
E	Right eyebrow outer corner, freedom degree: 2 Displacement: vertical up, horizontal inwards	(x_{min}, y_{min}) of right eyebrow's contour
E1	Left eyebrow outer corner, freedom degree: 2 Displacement: vertical up, horizontal inwards	(x_{max}, y_{min}) of left eyebrow's contour
H	Right nostril inner corner, freedom degree: 0	(x_{max}, y_{avg}) of right nostril's contour
H'	Right nostril outer corner, freedom degree: 1 Displacement: horizontal inwards / outwards	(x_{min}, y_{avg}) of right nostril's contour
H1	Left nostril inner corner, freedom degree: 0	(x_{min}, y_{avg}) of left nostril's contour
H1'	Left nostril outer corner, freedom degree: 1 Displacement: horizontal inwards / outwards	(x_{max}, y_{avg}) of left nostril's contour
K	Mouth top, freedom degree: 2 Displacement: any horizontal or vertical	$(x, ½(y_{max1} + y_{max2}))$ of mouth's contour
L	Mouth bottom, freedom degree: 2 Displacement: any horizontal or vertical	(x, y_{min}) of mouth's contour
I	Right mouth corner, freedom degree: 2 Displacement: any horizontal or vertical	(x_{min}, y) of mouth's contour
J	Left mouth corner, freedom degree: 2 Displacement: any horizontal or vertical	(x_{max}, y) of mouth's contour
M	Tip of the chin, freedom degree: 1 Displacement: vertical downwards	(x, y_{min}) of chin's contour

Table 5.5
Representation of AUs with the frontal-view face model using an informal pseudo code

AU	FACS description	Mapped onto the frontal-view face model
1	Raised inner eyebrow(s)	Increased BD ∨ Increased B1D1
2	Raised outer eyebrow(s)	Increased AE ∨ Increased A1E1
1+2	Raised eyebrows	Increased BD, AE, B1D1, A1E1
4	Eyebrows drawn together	Decreased DD1
5	Raised upper eyelid(s)	Increased FG ∨ Increased F1G1
6	Raised cheek(s)	AU12 or AU13 present
7	Raised lower eyelid(s)	(Absent AU9, AU12) ∧ ((FG > 0 ∧ Decreased GX) ∨ (F1G1 > 0 ∧ Decreased G1Y))[1]
8	Lips towards each other	(Absent AU9, AU12, AU13, AU15, AU17, AU18, AU20, AU23, AU24, AU35) ∧ Increased CK[2] ∧ KL > 0
12	Mouth corner(s) up	(Decreased IB ∧ Increased CI) ∨ (Decreased JB1 ∧ Increased CJ)
13	Mouth corner(s) sharply up	(Decreased IB ∧ Decreased CI) ∨ (Decreased JB1 ∧ Decreased CJ)
15	Mouth corner(s) down	Increased IB ∨ Increased JB1
18	Lips puckered	Absent AU28 ∧ Decreased IJ ∧ IJ ≥ t1 ∧ Not decreased KL
20	Mouth stretched	Increased IJ ∧ IB and JB1 remain the same
23	Lips tightened	(Absent AU28t, AU28b) ∧ Decreased KL ∧ KL > 0 ∧ Not decreased IJ ∧ Not increased IB ∧ Not increased JB1
24	Lips pressed	(Absent AU15, AU28t, AU28b, AU9+AU17, AU10+AU17, AU12+AU17, AU13+AU17) ∧ Decreased KL ∧ KL > 0 ∧ Decreased IJ ∧ IJ > t1
25	Lips parted	Increased KL ∧ Not increased CM
26	Jaw dropped	Increased CM ∧ CM ≤ t2
27	Mouth stretched	CM > t2
28	Lips sucked in	KL = 0
35	Cheeks sucked in	IJ < t1
38	Nostrils widened	(Absent AU8, AU9, AU10, AU12, AU13, AU15, AU18, AU24, AU28) ∧ Increased H'H1'
39	Nostrils compressed	(Absent AU8, AU9, AU10, AU12, AU13, AU15, AU18, AU24, AU28) ∧ Decreased H'H1'
41	Upper eyelid dropped	Absent AU7 ∧ ((FG > 0 ∧ Decreased FG ∧ Decreased FX) ∨ (F1G1 > 0 ∧ Decreased F1G1 ∧ Decreased F1Y))

[1] Point X is the centre of the left eye calculated as the intersection point between AB and FG. Point Y is the centre of the right eye calculated as the intersection point between A1B1 and F1G1.

[2] Point C is the centre of HH1.

From the total of 44 AUs defined in FACS, 22 AUs can be uniquely described using the frontal-view face model (Table 5.5). The importance of a unique representation of AU codes, and the utilised manner of achieving it in terms of the frontal-view face model, can be explained using an example. In FACS, the activation of AU38 is described as *widening of the nostrils*. However, it is also stated that activation of any of AU8, AU9, AU10, AU12, AU13, AU15, AU18, AU24, and AU28 obscures the activation of AU38. In order to obtain a unique description of AU38 activation with the utilised frontal-view face model, the following rule has been defined:

```
AU38 ⇔ distance H'H1' increased ∧ (AU8, AU9, AU10, AU12,
         AU13, AU15, AU18, AU24, AU28 are absent)
```

Similarly, all rules listed in Table 5.5 describe the relevant AUs uniquely and were acquired from FACS in a straightforward manner. These rules represent the first part of the knowledge about the facial muscle activity integrated into the system.

The point-based profile-view face model

The profile-view face model utilised by ISFER is composed of 10 face profile points. Harmon et al. (1981) have developed a similar model of the profile points for a face identification system. However, analysing an observed face in terms of displayed facial actions vs. in terms of person identification are two fundamentally different tasks. Personal characteristics such as the length of the nose are considered as unimportant data in facial action encoding while the opening of the mouth is considered as noise in face identification. The profile-view point-based face model utilised by ISFER is developed such that it is suitable for facial action encoding and facial expression interpretation and, therefore, merely resembles Harmon's model.

Table 5.6
Facial points of the profile-view face model utilised by ISFER

	Point description
P1	Top of the forehead, uppermost point of the curvature of the profile contour function
P2	Eyebrow arcade, 1st peak of the curvature of the profile contour function
P3	Root of the nose, 1st valley of the curvature of the profile contour function
P4	Tip of the nose, absolute maximum of the curvature of the profile contour function
P5	Upper jaw, 1st valley after P4 peak of the curvature of the profile contour function
P6	Upper lip, 1st peak after P4 peak of the curvature of the profile contour function
P7	Lips' joint, 1st valley after P6 peak of the curvature of the profile contour function
P8	Lower lip, 1st peak above P10 peak of the curvature of the profile contour function
P9	Lower jaw, 1st valley above P10 peak of the curvature of the profile contour function
P10	Tip of the chin, last peak of the curvature of the profile contour function

The points of the utilised profile-view face model correspond with the peaks and the valleys of the curvature of the profile contour function (Table 5.6). When locating the extremities of the curvature of the profile contour function, a priori knowledge is used to delete false positive/negative extremities (Wojdel, J. et al. 1999, section 4.3). The order of the selected extremities can be changed, however, if the tongue is visible or if either one or both lips are sucked into the mouth. In the case of a visible tongue, a valley representing the attachment of the upper lip to the tongue, a peak representing the tip of the tongue, and a valley representing the attachment of the tongue to the bottom lip, will be detectable between the points P6 and P8. In the case of the lips sucked into the mouth, only the valley of P7 will be detectable while peaks P6 and P8 will not exist. Therefore it is important to localise the profile points in a particular order. Points P1 to P5 are located first. Then, points P10 and P9 are located. After the exclusion of all extreme cases such as a visible tongue, points P8, P7 and P6 are located.

From the total of 44 AUs defined in FACS, 24 distinct facial appearance changes can be uniquely described using the profile-view face model (Table 5.7). Obtaining unique descriptions of distinct AUs in terms of the profile-view face model can be explained using an example. In FACS, the activation of AU9 as well as the activation of AU10 is described with the label *upward pull of the upper lip*. It is also stated, however, that activation of AU9 obscures the activation of AU10. On the other hand, the label *wrinkled root of the nose* describes AU9 exclusively. To obtain unique descriptions of AU9 and AU10 with the profile-view face model, the following rules have been defined:

```
AU9  ⇔ curvature between P2 and P3 increased
AU10 ⇔ distance P5P6 decreased ∧ P6 upwards ∧ P6 outwards
           ∧ curvature between P2 and P3 is not increased
```

Similarly, all rules listed in Table 5.7 describe the relevant AUs uniquely and were acquired from FACS in a straightforward manner. These rules represent the second part of the knowledge about the facial muscle activity integrated into the system.

The point-based dual-view face model

The main motivation for combining the frontal- and the profile-view face model into a dual-view face model (Figure 5.2) is the increase in quality of facial modelling caused by the increase in quantity of facial expressions that can be modelled using the combined face model. With the frontal-, profile-, and dual-view face model, activation of 22, 21 and 29 AUs, respectively, can be uniquely described. The reason is that each facial view is, in fact, more suitable for observing facial changes caused by certain AUs. For instance, changes in the appearance of the eyes, eyebrows and mouth corners can be perceived easier in the frontal view while changes in the appearance of the jaw and chin can be apprehended from the profile view more easily. The set of rules for facial action encoding based on the dual-view face model is given in Table 5.8. It is composed of the rules given in Table 5.5 and Table 5.7. This set of rules forms the knowledge about the facial muscle activity integrated into the system.

Table 5.7
Representation of AUs with the profile-view face model using an informal pseudo code

AU	FACS description	Mapped onto the profile-view face model
1	Raised inner eyebrow(s)	Absent AU1+AU2 ∧ P2 upwards ∧ Decreased P1P2
4	Eyebrows drawn together	P2 outwards
8	Lips towards each other	(Absent AU9, AU12, AU13, AU15, AU17, AU18, AU20, AU23, AU24, AU35) ∧ Increased P5P6 ∧ (P6 ∧ P8 outwards) ∧ curvature P6-P8 has [shape ∧ Increased P8P10
9	Wrinkled nose	Increased curvature P2-P3
10	Raised upper lip	P6 upwards ∧ P6 outwards ∧ Decreased P5P6 ∧ Not increased curvature P2-P3
12	Mouth corner(s) pulled up	Decreased P5P6 ∧ (P6 ∧ P8 inwards) ∧ Increased P6P8
13	Mouth corner(s) pulled sharply up	Decreased P5P6 ∧ (P6 ∧ P8 inwards) ∧ P6P8 remains the same
15	Mouth corner(s) pulled downwards	Increased P5P6 ∧ Not increased curvature P5-P6 ∧ (P6 ∧ P8 downwards) ∧ Not increased P6P8
16	Lower lip depressed	Decreased P8P10 ∧ P8 downwards ∧ P8 outwards
17	Chin raised	(Absent AU28, b, t) ∧ P10 inwards
18	Lips puckered	(P6 ∧ P8 outwards) ∧ curvature P6-P8 has not [shape
19	Tongue shown	Curvature P6-P8 contains two valleys and a peak
20	Mouth stretched	Increased P5P6 ∧ Not increased curvature P5-P6 ∧ (P6 ∧ P8 inwards) ∧ Not decreased P6P8
23	Lips tightened	(Absent AU28, b, t) ∧ P6 downwards ∧ P8 upwards ∧ (P6 ∧ P8 inwards) ∧ Not increased curvature P5-P6 ∧ Increased P5P6 ∧ Decreased P6P8 ∧ 0 < P6P8 ≥ t3
24	Lips pressed	(Absent AU15, AU28, b, t, AU9+AU17, AU10+AU17, AU12+AU17, AU13+AU17) ∧ P6 downwards ∧ P8 upwards ∧ (P6 ∧ P8 inwards) ∧ Increased P5P6 ∧ Not increased curvature P5-P6 ∧ Decreased P6P8 ∧ 0 < P6P8 < t3
25	Lips parted	Increased P6P8 ∧ Not increased P4P10
26	Jaw dropped	Increased P4P10 ∧ P4P10 ≤ t4
27	Mouth stretched	P4P10 > t4
28,t,b	Lip(s) sucked in	(Absent P6 ∧ Absent P8) ∨ (Absent P6) ∨ (Absent P8)
29	Jaw forward	Absent AU27 ∧ P10 outwards
36b	Tongue under the lower lip	Absent P9
36t	Tongue under the upper lip	Increased curvature P5-P6

Each single-view face model, when considered separately, uniquely models the appearance of facial features and when deformed does not contain any redundant information about the modelled facial expression. When two face models are combined, however, the resulting dual-view point-based face model reveals redundant information about the modelled facial expression. This redundant information is used for:

- *Control of the accuracy of the performed model-points' localisation*: For example, the distance KL measured in the frontal-view model should be equal to the distance P6P8 measured in the profile-view model. If that is not the case, then either points K and L or the profile-view model points have been localised inaccurately. Uncovering inaccurate data and dealing with these is explained in detail in section 5.4.
- *Dealing with partial data*: Several cases can be distinguished. In the case that all dual-view model points are successfully detected, the rules given in Table 5.8 are applied for facial action encoding. If the spatial sampling of the profile contour is not successfully performed, the facial action encoding is obtained according to the rules listed in Table 5.5. If merely the profile contour is successfully detected, the rules of Table 5.7 are employed for the facial action encoding. In the case of successful spatial sampling of the profile contour and partially successful localisation of the frontal-view model points, an appropriate combination of the rules given in Tables 5.5 and 5.7 will be utilised for facial action encoding. For example, if none of the detectors of the Facial Data Extractor successfully localises the mouth contour, encoding the change in the facial appearance of the mouth is based on the relevant rules given in Table 5.7. Dealing with missing data is further explained in sections 5.4 and 5.6.
- *Handling the cases where just a full-face image represents the input image*: This is in fact a special case of dealing with partial data. In this case, as noted above, the facial actions will be coded by employing the rules given in Table 5.5.

Table 5.8
Representation of AUs with the dual-view face model

Based on the frontal-view point-based face model (Table 5.5)
AU1, AU2, AU4, AU5, AU6, AU7, AU12, AU13, AU15, AU18, AU20, AU23, AU24, AU35, AU38, AU39, AU41
Based on the profile-view point-based face model (Table 5.7)
AU8, AU9, AU10, AU16, AU17, AU19, AU25, AU26, AU27, AU28, AU28t, AU28b, AU29, AU36t, AU36b

5.4 Handling ambiguous facial expression information

The input to the Facial Action Encoder part of ISFER is a set of files (Figure 5.1), each of which contains a contour of a prominent facial feature localised by a detector integrated into the Facial Data Extractor part of the system. This input data is most likely to be redundant since for each facial feature which is to be spatially sampled, several different detectors have been integrated into the Facial Data Extractor (see chapter 4). The input data might be also partial; this is the case when none of the relevant detectors can spatially sample the contour of a certain facial feature successfully. Finally, the input data is usually approximate as opposed to exact since the detectors integrated into the Facial Data Extractor are not 100% accurate and hence generate data of variable precision. In turn, the Facial Action Encoder should deal with ambiguous input information while recognising facial expressions in terms of quantified AU codes.

In general, there are three principal formalisms for handling uncertainty in an expert system: probability theory, belief functions, and fuzzy logic. In section 3.4, the suitability of these approaches for handling uncertainty within the Facial Action Encoder part of ISFER has been explored in detail. It has been shown that an appropriate method for estimating and propagating certainty of the data resulting from the Facial Data Extractor should not be based solely on either of these formalisms but should rather form a certain blend of these formalisms. This association should be further devised to reflect a situation in which the properties used to assign the elements of a set U to the elements of a set V are uncertain, but the process used to select the properties which play a role in this assignment is known and can be represented either by probabilities or by possibilities. In other words, an inexact reasoning method appropriate for handling uncertainty within the Facial Action Encoder part of ISFER should exploit the available process knowledge that can be represented either by probabilities or by possibilities. For instance, a piece of the available process knowledge that can be represented by probabilities and used for dealing with uncertain data resulting from the Facial Data Extractor concerns the following – the larger the number of detectors that spatially sample the same contour of a certain facial feature, the higher the certainty of that datum.

The rest of this section elucidates the kind of process knowledge involved in handling ambiguous facial expression information resulting from the Facial Data Extractor and exploited by the pre-processing data evaluator part of the Facial Action Encoder (Figure 2.25). For the sake of clarity and readability, the process of dealing with imperfect input data is explained further by examining how the three functions of the pre-processing data evaluator (F1, F2 and F3; Figure 5.1) support this process.

Fl and F2: Abridging the input data

A successful processing of the Facial Action Encoder part of ISFER implies that at least one of the following assumptions is true:

1. A full-face image of the currently observed person represents the input image and all points of the corresponding frontal-view face model are available.
2. A profile-view facial image of the currently observed person represents the input image and the points of the corresponding profile-view face model are available.

Therefore, the first predefined goal that the Facial Action Encoder's *supervisor* (Figure 5.1) will try to achieve is to ensure that the points of the applicable face model (frontal- or profile- or dual-view face model) are available. To this end, the supervisor first abridges the input data by activating functions F1 and F2.

If a certain detector integrated into the Facial Data Extractor part of ISFER fails to spatially sample the contour of a particular facial feature, the file forming the input to the Facial Action Encoder part of ISFER and carrying this result will be either empty or hold a singularity. The data stored in such a file do not provide any information on the currently examined facial expression and will be cast off as useless by function F1. Function F2 further reduces the amount of data furnishing the remaining input files so that each file contains merely those points of the stored facial feature's contour that correspond to certain model points. Function F2 is carried out by activating the relevant rules for extracting the model points from the localised contours of the prominent facial features (Tables 5.4 and 5.6). Since for further system processing, the whole eyebrow contour and the whole profile contour are necessary (see the rest of this section and section 5.5), function F2 is not executed for the "eyebrow" files and the "profile" file(s).

Once the input files are cleaned from superfluous data, the supervisor classifies the remaining files per prominent facial feature (profile, eyebrows, eyes, nose, mouth, chin). In the case that a certain class contains no files (i.e. all files containing the localised contour of the relevant facial feature have been discarded by function F1), the problem imposed by encountering partial data jeopardizes achievement of the first goal that the supervisor tries to reach. In the case that the system processes facial image sequences rather than static facial images, the (process) knowledge about how to estimate the spatial location of the contour of a missing facial feature will be available; i.e., the larger the number of the frames per minute of the examined video sequence, the higher the certainty that the appearance of the monitored facial features remains the same. In that case, the missing data could be substituted by the relevant data extracted from the previous frame of the examined facial image sequence. However, ISFER deals with static facial images and there is no available knowledge that can be used to estimate the spatial location of a facial feature that was not detected in an input static facial image. Yet, in order to proceed with the analysis of the available (partial) information on the currently examined facial expression, the supervisor generates per missing facial feature a file labelled as *missing* where the relevant data extracted from the expressionless face of the currently observed person are stored. The supervisor furnishes "missing" files by accessing the appropriate data stored in the personalised *DB of all encountered facial expressions* of the current subject (Figure 5.1). Of course, substituting missing data with the relevant data extracted from the neutral facial expression of the currently observed person implies that the exact information about the examined facial expression is lost. Nevertheless, in a later processing stage and by activating function F5, the Facial Action Encoder deals with the problem imposed by this information loss. Section 5.6 is dedicated to that issue.

F3: Dealing with approximate data

Once the supervisor achieves the goal of making the points of the appropriate face model available for a further processing, it will try to reach the next goal: to select per facial feature the most accurate contour from the redundantly detected contours of that feature. To this end, the supervisor selects from the action space three predefined plans that are sequentially executed, namely:

1. For each file forming a part of the abridged input data (see the previous subsection), activate function F3 to perform an *intra-file consistency check* and compute an "intermediate" certainty of the data stored in the file. This action-plan addresses the issue of dealing with approximate input data.
2. For each class of files containing the data about the spatial sampling of a particular facial feature, activate function F3 to perform an *inter-file consistency check* and compute a "final" certainty of the data stored in the files.
3. For each class of files containing the data about the spatial sampling of a particular facial feature, classify the files according to the certainty of the data stored in each file and select the ones holding the data having the highest certainty. Combined with the second action plan listed here, this action plan addresses the issue of dealing with redundant input data.

Let us examine first how function F3 supports handling of the approximate data. As already explained in section 3.4, a way of dealing with approximate data resulting from the Facial Data Extractor part of ISFER is to exploit the process knowledge that is based on the knowledge about the facial anatomy and dynamics. This involves association and employment of the following facts:

- Some facial peculiarities are stable in the sense that no facial muscle action can cause their temporary change. Those facial characteristics are the size of the eyebrows' facial area (Figure 5.4) and the facial position of the inner corners of the eyes (Figure 5.3, Figure 5.5), the medial point of the mouth (Figure 5.6), the inner corners of the nostrils (Figure 5.7), and the tip of the nose (Figure 5.9).
- All of the single-session images acquired on-line or downloaded (and then scaled) from an existing database of behavioural science research material are scale and orientation invariant (section 4.1). Hence, the measurements of the stable facial peculiarities computed from a neutral facial expression of the current subject should remain the same during the whole session with that subject.
- The certainty of an input datum, representing the spatial sampling of a certain facial feature by a particular detector, can be estimated based on the error made by that detector while detecting the stable facial characteristic particular for the given facial feature. Namely, the larger the spatial sampling error, the lower the certainty assigned to the datum in question and, if no spatial sampling error is encountered, a maximal certainty measure (say 100%) is to be assigned to the relevant datum.

Thus, function F3 models a situation in which the property (approximativeness of the input data) used to associate the element of some set *V* (spatially sampled facial feature) with some value $u \in U$ (degree of certainty) is uncertain, but we know the process used to assign this value (calculating the degree of deviation of the actually detected stable facial characteristic from the pertinent characteristic detected in the expressionless face of the observed subject). For each input file, function F3 performs an intra-file consistency check (i.e. it calculates the spatial sampling error mentioned above) and, based on this process knowledge, it expresses the approximativeness of the input data in terms of possibilities. In the case that just a full-face facial image forms the input to the system, function F3 will not evaluate the approximativeness of the "profile" file(s); it will omit this processing step.

Evaluating "eyebrow" input files

Per input file containing a spatial sampling of an eyebrow contour, function F3 assigns a certainty measure $CM \in [0,100]$ to the input data x according to the computed deviation of the currently sampled size of the eyebrow area $size_{current}$ from the pertinent $size_{neutral}$ measured in the expressionless face of the observed subject. The functional form of this mapping is further defined as $CM = S(x) * 100$, where $S(x)$ is defined as

$$sigm(|deviation(size_{current}, size_{neutral})|); size_{neutral}*0.01, size_{neutral}*0.07, size_{neutral}*0.15)$$

while $sigm(y;\ \alpha,\ \beta,\ \gamma)$ is a Sigmoid membership function defined in the possibility theory as given in formula *(1)*.

$$sigm(y;\alpha,\beta,\gamma) = \begin{cases} 1 & y \le \alpha \\ 1-2[(y-\alpha)/(\gamma-\alpha)]^2 & \alpha < y < \beta \\ 2[(y-\gamma)/(\gamma-\alpha)]^2 & \beta < y < \gamma \\ 0 & y \ge \gamma \end{cases} \qquad (1)$$

The method used for computing the size of the eyebrow area concerns counting the pixels of the input image that lay within the boundaries of the eyebrow contour sampled by an eyebrow detector. In the case that $|deviation(size_{current}, size_{neutral})|$ is 0 pixels or does not exceed 1% of $size_{neutral}$, $CM = 100$ will be assigned to the data constituting the file that carries the result of the detector in question. Since the eyebrow detection for which $|deviation(size_{current}, size_{neutral})|$ does not exceed 7% of $size_{neutral}$ is considered rather accurate (see Figure 5.4 – the difference between the two computed sizes of the right eyebrow is 3.75%), the $CM \in (50, 100)$ will be assigned to the pertinent input data. If $|deviation(size_{current}, size_{neutral})|$ exceeds 15% of $size_{neutral}$, $CM = 0$ will be assigned to the pertinent input data and handling the highly inaccurate data becomes the next problem posed to the supervisor of the Facial Action Encoder. The problem of handling highly inaccurate data is very similar to the problem of handling missing data and it has been addressed as a part of the inter-file consistency check performed by function F3 (see the next section).

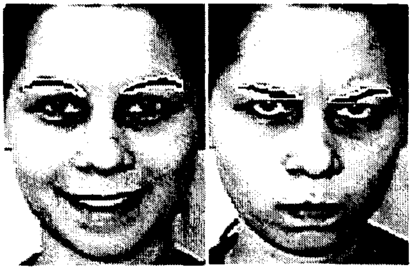


Figure 5.4: Spatial sampling of eyebrows' contours in input images of 720×576 pixels; computed sizes of the eyebrow areas: 2133 pixels (left image, right eyebrow), 2213 pixels (right image, right eyebrow), 2224 (left image, left eyebrow), 2240 (right image, left eyebrow).

Evaluating "eye" input files

Per input file containing a spatial sampling of an eye contour, function F3 assigns a certainty measure $CM \in [0,100]$ to the input data x according to the calculated deviation of the actually detected inner corner of the eye $B_{current}$ from the pertinent point $B_{neutral}$ localized in the expressionless face of the observed subject. The functional form of this mapping is defined as $CM = S(x) * 100$, where $S(x) = sigm(d(B_{current}, B_{neutral});1,4, 10)$ and $d(p_1, p_2)$ is the block distance between points p_1 and p_2 (maximal difference in x and y direction), while $sigm(y;\ \alpha,\ \beta,\ \gamma)$ is a Sigmoid membership function given in formula *(1)*.

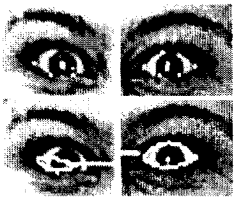


Figure 5.5: Spatial sampling of the eye contour - measured *Er*:
d($B_{current}$, $B_{neutral}$) = 14 (left image)
d($B_{current}$, $B_{neutral}$) = 1 (right image)

As already explained in section 3.4, spatial sampling of the eye contour for which $d(B_{current}, B_{neutral}) \leq 4$ is considered rather accurate given that the examined facial images measure 720×576 pixels. In the case that $d(B_{current}, B_{neutral}) \geq 10$ (see Figure 5.5), $CM = 0$ will be assigned to the input data constituting the file that carries the result of the eye detector in question.

Evaluating "mouth" input files

The approximativeness of the input data generated by a mouth detector is expressed by function F3 in terms of possibilities based on the knowledge about the facial stability of the medial point of the mouth. This knowledge originates from the anatomy of the face. Namely, independently of the action of the facial muscles (horizontal action like mouth stretching, vertical action like jaw drop, oblique action like smile, or orbital action like tightening the lips) that can affect the facial appearance of the mouth, the (imaginary) medial point M of the mouth computed according to formula *(2)* remains stable (Figure 5.6).

$$M = centre\ (M_h, M_v), \text{ where } M_h = centre\ (I, J) \text{ and } M_v = centre\ (K, L) \quad (2)$$

given that I is the right corner of the mouth, J is the left corner of the mouth, K is the top of the mouth, L is the bottom of the mouth (Table 5.4), and *centre (X, Y)* is the middle point of the line defined by the points X and Y.

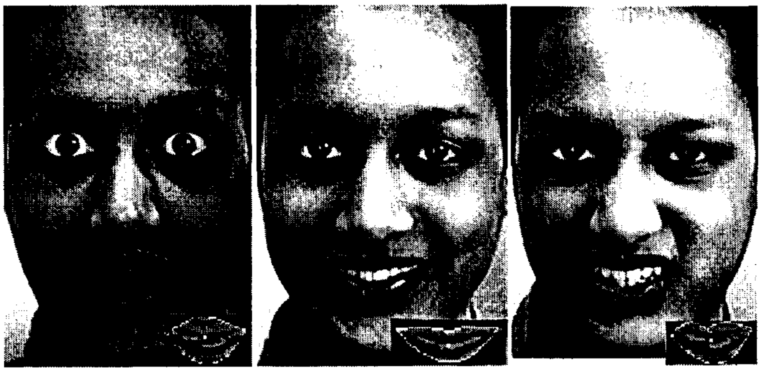


Figure 5.6: Spatial sampling of the mouth contour - measured *Er*: d($M_{current}$, $M_{neutral}$) = 6, (left image), d($M_{current}$, $M_{neutral}$) = 1 (middle image), d($M_{current}$, $M_{neutral}$) = 4 (right image)

Per input file containing a spatial sampling of the mouth contour, function F3 assigns a certainty measure $CM \in [0,100]$ to the input data x according to the calculated deviation of the actually detected medial point of the mouth $M_{current}$ from the medial point of the mouth $M_{neutral}$ localized in the expressionless face of the observed subject. This mapping is defined as $CM = S(x) * 100$, where $S(x) = sigm(d(M_{current}, M_{neutral}); 2, 7, 15)$, M is the medial point of the mouth calculated according to formula *(2)*, $d(p_1, p_2)$ is the block distance between points p_1 and p_2, and $sigm(y; \alpha, \beta, \gamma)$ is a Sigmoid membership function given in formula *(1)*.

In an input facial image of 720×576 pixels, the size of the mouth area varies between 70 and 300 pixels in the horizontal and between 20 and 200 pixels in the vertical direction (unless the lips are sucked into the mouth; handling this special case of the mouth shape is discussed in the next section of this chapter). Hence, the spatial sampling of the mouth contour for which $d(M_{current}, M_{neutral}) \leq 7$ is considered rather accurate (see Figure 5.6). In the case that $d(M_{current}, M_{neutral}) \geq 15$, $CM = 0$ will be assigned to the input data constituting the file that carries the result of the eye-detector in question.

Evaluating "nose/chin" input file(s)

For an input file containing the spatial sampling of the nostrils and the chin, function F3 assigns a certainty measure $CM \in [0,100]$ to the input data x according to the calculated deviation of the actually detected inner corners of the nostrils $H_{current}$ and $H1_{current}$ from the pertinent points $H_{neutral}$ and $H1_{neutral}$ localized in the expressionless face of the observed subject. This mapping is further defined as $\boldsymbol{CM = S(x) * 100}$, where $\boldsymbol{S(x) = avg\ (sigm(d(H_{current}, H_{neutral});}1,3,7)$, $\boldsymbol{(sigm(d(H1_{current}, H1_{neutral});}1,3,7))$, $\boldsymbol{d(p_1, p_2)}$ is the block distance between points $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$, and $\boldsymbol{sigm(y;\ \alpha, \beta, \gamma)}$ is a Sigmoid membership function given in formula *(1)*.

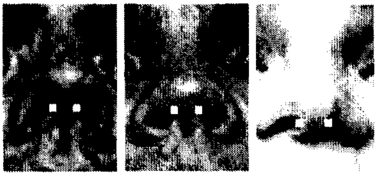


Figure 5.7: Manual localisation of the inner corners of the nostrils

When asked to point out the inner corners of the nostrils in a digitised facial image, human observers usually marked the upper points of the nostrils located close to the tip of the nose (Figure 5.7). However, since the detector *Find Nose & Chin* integrated into the Facial Data Extractor part of ISFER approximates the nostrils' contours by two circles (Figure 5.8), in the current version of the system the model points H and H1 intuitively suggested by human observers cannot be localised automatically. Points H and H1 are extracted from the detected nostrils' contours as respectively the innermost and outermost point of the relevant nostril contour (Table 5.4).

Since in an input facial image of 720×576 pixels the size of a nostril area varies between 25 and 30 pixels in either direction, the spatial sampling of a nostril contour for which $\boldsymbol{d(H_{current}, H_{neutral})} \leq 3$ (or $\boldsymbol{d(H1_{current}, H1_{neutral})} \leq 3$) is considered rather accurate (see Figure 5.8). In the case that $\boldsymbol{d(H_{current}, H_{neutral})} \geq 7$ and $\boldsymbol{d(H1_{current}, H1_{neutral})} \geq 7$, $\boldsymbol{CM} = 0$ will be assigned to the input data constituting the file that carries the result of the eye-detector in question.

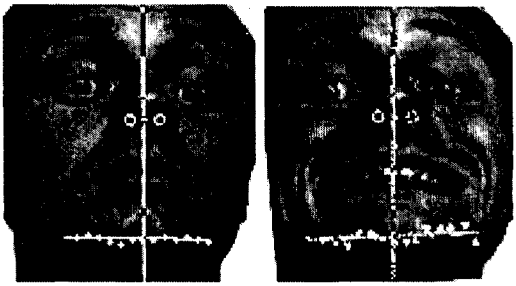


Figure 5.8: Spatial sampling of the nostrils' contours - measured *Er*: $d(H_{current}, H_{neutral}) = 1$ (left image), $d(H1_{current}, H1_{neutral}) = 1$ (left image), $d(H_{current}, H_{neutral}) = 5$ (right image), $d(H1_{current}, H1_{neutral}) = 3$ (right image)

Evaluating "profile" input file(s)

Because the tip of the nose is relatively stability with respect to the static facial signals such as the bony structure and the overall proportions of the face, this facial landmark is most commonly used for automatic person identification together with the inner corners of the eyes (Samal and Iyengar 1992). No facial muscle action can cause its temporary displacement relative to the static facial signals (Figure 5.9). This fact is used by function F3 to calculate the approximativeness of the data x furnishing the "profile" input file.

Function F3 assigns a certainty measure $CM \in [0,100]$ to the input data x according to the calculated deviation of the actually detected tip of the nose $P4_{current}$ from the relevant point $P4_{neutral}$ localized in the expressionless face of the observed subject. The functional form of this mapping is further defined as $CM = S(x) * 100$, where $S(x) = sigm(d(P4_{current}, P4_{neutral}); 1, 4, 10)$, $d(p_1, p_2)$ is the block distance between points p_1 and p_2, and $sigm(y; \alpha, \beta, \gamma)$ is a Sigmoid membership function given in formula *(1)*.

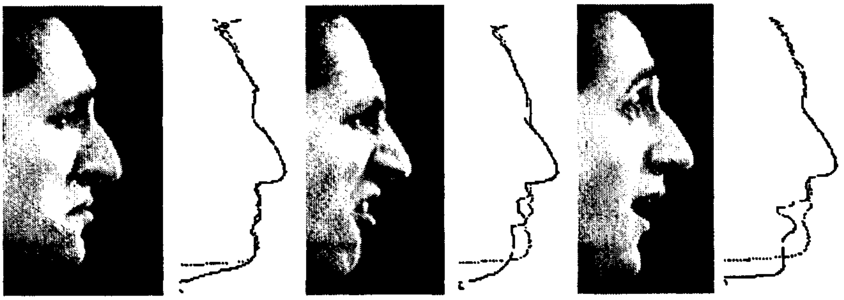


Figure 5.9: Spatial sampling of the profile contour – comparison between the profile contour spatially sampled in the neutral expression (thin line) and the profile contour detected in the examined facial expression (thick line)

A spatial sampling of the profile contour for which $d(P4_{current}, P4_{neutral}) \leq 4$ is considered rather accurate given that the examined facial images measure 720×576 pixels. In the case that $d(P4_{current}, P4_{neutral}) \geq 10$, $CM = 0$ will be assigned to the input data representing the result of the profile detector. However, the probability that this will happen is very low taking into account the test results of the profile detector integrated into the Facial Data Extractor part of ISFER (i.e. for all of the test images, the localisation error for ***P4*** remained under 5 pixels, see Table 4.4).

F3: Dealing with redundant and highly inaccurate data

The Facial Data Extractor generates redundant data when several of the integrated facial feature detectors successfully spatially sample the contour of the same prominent facial feature. As already explained, the goal that the supervisor of the Facial Action Encoder will try to reach is to select per facial feature the most accurate contour from the redundantly detected contours. To this end, the supervisor will activate function F3 as explained in the previous section. Once function F3 executes the predefined action plan that deals with approximate data it will execute the action plans that deal with redundant input data. According to these action plans an inter-file consistency check is performed and per facial feature the input file is selected containing the data having the highest certainty.

Performing the intended inter-file consistency check is, in fact, exploiting the process knowledge based on the existence of (and derived from) redundant input data. Namely, if different detectors result in a same spatial sampling of a certain feature's contour, the results of these detectors confirm each other and yield a higher confidence in the resulting datum. In other words, the larger the number of detectors performing the same spatial sampling of the contour of a certain feature, the higher the certainty about that datum.

For each class of files containing the data about the spatial sampling of a particular facial feature, function F3 performs an inter-file consistency check, calculates a final *data certainty* ***DC*** of the input data, and selects per facial feature the input datum having the highest ***DC*** assigned to it. Since the data about the spatially sampled profile contour are used for checking the consistency of the data stored in the other input files, function F3 evaluates the input "profile" file(s) first. Hereafter, the order in which function F3 evaluates the rest of the classes of input files is not of importance. If just a full-face facial image forms the input to the system, function F3 will not evaluate the "profile" file(s); it will omit this processing step. Also the inter-file consistency checks, containing a step in which the consistency of the data stored in the currently examined input file is checked against the data stored in the "profile" file, will omit this processing step.

Evaluating "profile" input file(s)

If the supervisor of the Facial Action Encoder part of ISFER has labelled an input "profile" file as "missing" (after executing functions F1 and F2) and furnished this file with the relevant data $x_{neutral}$ extracted from the neutral expression of the currently observed subject, function F3 assigns $CM = P(x_{neutral}) * 100$ to this data. $P(x_{neutral})$ is the probability of the neutral expression (i.e. the probability that no AU has been activated) calculated according to formula *(3)*. This reflects the situation where we want to calculate the probability that a specific marble (neutral expression) will be taken out of a hat full of marbles (all individual AUs and all their anatomically possible combinations that could be recognised from the profile model's deformations).

$$P(x_{neutral}) = \frac{1}{\sum_{i=1}^{n} x_i}, \qquad (3)$$

where n is the total number of visually distinguishable expressions of the prominent facial feature in question and x_i is one such expression.

Since there are 24 distinct AU codes and more than 100 different combinations of these AUs that can be uniquely encoded based on the observed deformations of the profile model (Table 5.7), the $CM \approx 0$ will be assigned to the data constituting a "profile" file labelled as "missing". Yet, due to the test results of the profile detector integrated into the Facial Data Extractor (Table 4.4), the probability that function F3 will assign $\boldsymbol{CM} = 0$ to the data furnishing an input "profile" file is very low.

Currently, a single profile detector is integrated into the Facial Data Extractor part of ISFER. Hence, function F3 cannot perform an inter-file consistency check and it assigns a final data certainty $\boldsymbol{DC} = \boldsymbol{CM}$ to the data constituting the existing "profile" input file. Yet the method with which final data certainty $\boldsymbol{DC}$ for the input "profile" file is computed, should not have a detrimental effect upon the further development of the system. If at some point in the future another profile detector were to be integrated into the system, function F3 should accomplish a proper inter-file consistency check and calculate a final data certainty accordingly. Depending on the number $\boldsymbol{j} \geq 1$ of different profile detectors integrated into the Facial Data Extractor, function F3 executes the following inter-file consistency check:

1. If $\boldsymbol{j} = 1$, then $\boldsymbol{DC_{detector1}} = \boldsymbol{CM_{detector1}}$. Terminate the execution of this algorithm and use the available "profile" file in the system's further processing.
2. If $\boldsymbol{j} = 2$, then $DC_{det\,ector1} = \begin{cases} avg\left(CM_{det\,ector1}, S\left(x_{det\,ector1}, x_{det\,ector2}\right)\right) & CM < s \\ CM_{det\,ector1} & CM \geq s \end{cases}$,

   where $\boldsymbol{s} = \boldsymbol{S}(\boldsymbol{x_{detector1}}, \boldsymbol{x_{detector2}}) = 100 * \boldsymbol{avg}(\forall \boldsymbol{i} \in [1, 10] \mid \boldsymbol{sigm}(\boldsymbol{d}(\boldsymbol{Pi} \in \boldsymbol{x_{detector1}}, \boldsymbol{Pi} \in \boldsymbol{x_{detector2}}); 0, 2, 3))$ is the measure of similarity between the results of two profile detectors in question, $\boldsymbol{Pi}$ is a profile face model point, $\boldsymbol{d(p_1, p_2)}$ is the block distance between the points $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$, and $\boldsymbol{sigm}(\boldsymbol{y}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$ is a Sigmoid membership function given in formula *(1)*. If the measure of similarity between the detectors' results is high, the confidence in the examined result is increased as expressed by the given formula. Go to step **4**.
3. If $\boldsymbol{j} = \boldsymbol{n} > 2$, then $(\forall \boldsymbol{j} \in [2, \boldsymbol{n}])$ execute step-**2** and assign the calculated $\boldsymbol{DC_{detector1}}$ to a new $\boldsymbol{CM_{detector1}}$ that is to be used in the next loop. After the termination of all loops, assign $\boldsymbol{DC_{detector1}} = \boldsymbol{CM_{detector1}}$. Repeat the process for each of $\boldsymbol{n}$ detectors. Go to step **4**.
4. If $\boldsymbol{j} \geq 2$, then select the input "profile" file having the highest $\boldsymbol{DC}$ assigned to its data. If this process results in a draw, the data resulting from the detector having the highest priority is to be selected. A priority $\boldsymbol{k} \in [1,\boldsymbol{n}]$ is to be assigned (off-line) to each "profile" detector in accordance with its test results. Terminate the execution of this algorithm and use the selected "profile" file in the system's further processing.

Evaluating "eye" input files

While estimating the approximativeness of the input data, function F3 might have assigned a certainty measure $\boldsymbol{CM} = 0$ to the data constituting an input "eye" file. If this is the case, handling this highly inaccurate data becomes the next problem posed to the supervisor of the Facial Action Encoder. The supervisor treats highly inaccurate data in the same way as missing data. That is, it labels the "eye" file in question as "missing" and fills it further with the relevant data extracted from the expressionless face of the currently observed subject.

If the supervisor has labelled an input "eye" file as "missing" and furnished this file with the relevant data $x_{neutral}$ extracted from the neutral expression of the currently observed subject, function F3 assigns $CM = P(x_{neutral}) * 100$ to this data. $P(x_{neutral})$ is the probability that no AU whose activation induces a change in the facial appearance of an eye has been activated. This probability is calculated according to formula *(3)*. Since there are 5 distinct AUs and 18 different combinations of these (allowed by the co-occurrence rules defined in FACS) which can be uniquely recognised based on the observed displacement of the frontal-view face model points modelling the eye (Table 5.5), a certainty measure $CM = 100/24$ will be assigned to the data constituting an "eye" file labelled as "missing".

Depending on the number of different eye detectors integrated into the Facial Data Extractor ($j \geq 1$), function F3 executes the same inter-file consistency check as that defined for the class of input "profile" files. The only difference lies in the definition of function S, which is defined in the case of input "eye" files as given in the following formula:

$$S(x_{detector1}, x_{detector2}) = 100*avg(\forall i \in [1,4] | sigm(d(Pi \in x_{detector1}, Pi \in x_{detector2});0,2,3)),$$

where $S(x_{detector1}, x_{detector2})$ is the measure of similarity between the results of two eye detectors in question, Pi is a frontal-view face-model point belonging to the contour of the relevant eye, $d(p_1, p_2)$ is the block distance between the points p_1 and p_2, and $sigm(y; \alpha, \beta, \gamma)$ is a Sigmoid membership function given in formula *(1)*.

Evaluating "eyebrow" input files

While estimating the approximativeness of the input data, function F3 may have assigned a certainty measure $CM = 0$ to the data constituting an input "eyebrow" file. In this case, the supervisor deals with this highly inaccurate data by labelling the "eyebrow" file in question as "missing" and supplying it with the relevant data $x_{neutral}$ extracted from the expressionless face of the currently observed subject.

If the supervisor has labelled an input "eyebrow" file as "missing", function F3 assigns $CM = P(x_{neutral}) * 100$ to the data $x_{neutral}$ constituting this file. $P(x_{neutral})$ is the probability that no AU whose activation induces a change in the facial appearance of an eyebrow has been activated. This probability is calculated according to formula *(3)*. Since there are 4 distinct AUs and 7 different combinations of these (allowed by the co-occurrence rules defined in FACS and affecting the appearance of the eyebrows) which can be uniquely recognised based on the observed relevant deformations of the dual-view face model (Table 5.5, Table 5.7), a certainty measure $CM = 100/14$ will be assigned to the data constituting an "eyebrow" file labelled as "missing".

Depending on the number of different eyebrow detectors integrated into the Facial Data Extractor ($j \geq 1$), function F3 executes the same inter-file consistency check as that defined for the class of input "profile" files, where function $\boldsymbol{S}$ is defined as:

$$S(x_{detector1}, x_{detector2}) = 100*\boldsymbol{avg}(\forall i \in [1,2] | \boldsymbol{sigm}(\boldsymbol{d}(\boldsymbol{Pi} \in x_{detector1}, \boldsymbol{Pi} \in x_{detector2}); 0,2,3)),$$

where $S(x_{detector1}, x_{detector2})$ is the measure of similarity between the results of two eyebrow detectors in question, $\boldsymbol{Pi}$ is a frontal-view face-model point belonging to the contour of the relevant eyebrow, $\boldsymbol{d(p_1, p_2)}$ is the block distance between the points $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$, and $\boldsymbol{sigm}(y;\ \alpha,\ \beta,\ \gamma)$ is a Sigmoid membership function given in formula *(1)*.

Evaluating "nose/chin" input file(s)

While estimating the approximativeness of the input data, function F3 may have assigned a certainty measure $\boldsymbol{CM} = 0$ to the data constituting an input "nose/chin" file. If this is the case, the supervisor labels the "nose/chin" file in question as "missing" and supplies it with the relevant data $x_{neutral}$ extracted from the expressionless face of the currently observed subject.

If the supervisor has labelled an input "nose/chin" file as "missing", function F3 assigns $\boldsymbol{CM} = \boldsymbol{P}(x_{neutral}) * 100$ to the data $x_{neutral}$ constituting this file. $\boldsymbol{P}(x_{neutral})$ is the probability (calculated according to formula *(3)*) that no AU whose activation induces a change in the facial appearance of the nostrils and/or of the chin has been activated. Since there are 11 distinct AU codes and 24 different combinations of these (allowed by the co-occurrence rules defined in FACS and affecting the appearance of the nostrils and/or of the chin) which can be uniquely encoded based on the observed relevant deformations of the dual-view face model (Table 5.5, Table 5.7), a certainty measure $\boldsymbol{CM} = 100/36$ will be assigned to the data constituting an "nose/chin" file labelled as "missing".

Currently, a single nose/chin detector (i.e., *Find Nose & Chin* detector, Table 4.12) is integrated into the Facial Data Extractor part of ISFER. Hence, function F3 cannot perform an inter-file consistency check and it assigns a final data certainty $\boldsymbol{DC} = \boldsymbol{CM}$ to the data constituting the existing "nose/chin" input file. Yet, if at some point in the future another nose/chin detector were to be integrated into the system, function F3 should accomplish a proper inter-file consistency check and calculate a final data certainty accordingly. Depending on the number of different nose/chin detectors integrated into the Facial Data Extractor ($j \geq 1$), function F3 executes the following inter-file consistency check:

1. If $\boldsymbol{j} = 1$, then $\boldsymbol{DC}_{detector1} = \boldsymbol{CM}_{detector1}$. Terminate the execution of this algorithm and use the available "nose/chin" file in the system's further processing.
2. If $\boldsymbol{j} = 2$, then $DC_{detector1} = \begin{cases} avg(CM_{detector1}, S(x_{detector1}, x_{detector2})) & CM < s \\ CM_{detector1} & CM \geq s \end{cases}$,

   where $s = S(x_{detector1}, x_{detector2}) = 100*\boldsymbol{avg}(\forall i \in [1,3] \mid \boldsymbol{sigm}(\boldsymbol{d}(\boldsymbol{Pi} \in x_{detector1}, \boldsymbol{Pi} \in x_{detector2});\ 0,\ 2,\ 3))$ is the measure of similarity between the results of two nose/chin detectors in question, $\boldsymbol{Pi}$ is one of the frontal-view face-model points

H', H1' or M (Figure 5.2, Table 5.4), $d(p_1, p_2)$ is the block distance between the points p_1 and p_2, and $sigm(y;\ \alpha,\ \beta,\ \gamma)$ is a Sigmoid membership function given in formula *(1)*. If the measure of similarity between the detectors' results is high, the confidence in the examined result is increased as expressed by the given formula. Go to step **4**.

3. If $j = n > 2$, then ($\forall j \in [2, n]$) execute step-**2** and assign the calculated $DC_{detector1}$ to a new $CM_{detector1}$ that is to be used in the next loop. After the termination of all loops, assign $DC_{detector1} = CM_{detector1}$. Repeat the process for each of n detectors. Go to step **4**.
4. If a dual-view facial image forms the input to the system, then $\forall j$ assign the calculated $DC_{detector1}$ to a new $CM_{detector1}$ and perform an inter-file consistency check which compares the result of the nose/chin detector with the result of the selected / available profile detector. Since a displacement of the tip of the chin is observable in the frontal-view as well as in the profile-view facial image of the current expression and given that all of the single-session images are scale and orientation invariant (section 4.1), the proportion $MC_{examined}$ / $MC_{neutral}$ = $P4P10_{examined}$ / $P4P10_{neutral}$, where point C is the centre of the line HH1 (Table 5.5), should hold. If the measure of similarity between the examined nose/chin detector's result and the selected profile detector's result is high, the confidence in the nose/chin detector's result is increased. This is expressed by the formula computing a final data certainty $DC_{detector1}$ of the examined nose/chin detector:

$$DC_{detector1} = \begin{cases} avg\left(CM_{detector1}, S\left(MC_{detector1}, P4P10_{selected\_profile}\right)\right) & CM < s \\ CM_{detector1} & CM \geq s \end{cases}, \text{ and}$$

$$S(MC_{detector1}, P4P10_{selected\_profile}) = 100 * sigm\left(\left|\frac{MC_{detector1}}{MC_{neutral}} - \frac{P4P10_{selected\_profile}}{P4P10_{neutral}}\right|; 0,2,3\right)$$

   is the measure of similarity between the examined nose/chin detector's result and the selected profile detector's result while $sigm(y;\ \alpha,\ \beta,\ \gamma)$ is a Sigmoid membership function given in formula *(1)*. Go to step **5**.
5. If $j \geq 2$, then select the input "nose/chin" file having the highest DC assigned to its data. If this process results in a draw, the data resulting from the detector having the highest priority is to be selected. A priority $k \in [1,n]$ is to be assigned (off-line) to each "nose/chin" detector in accordance with its test results. Terminate the execution of this algorithm and use the selected "nose/chin" file in the further processing.

Evaluating "mouth" input files

While estimating the approximativeness of the input data, function F3 may have assigned a certainty measure $CM = 0$ to the data constituting an input "mouth" file. If this is the case, the supervisor labels the "mouth" file in question as "missing" and supplies it with the relevant data $x_{neutral}$ extracted from the expressionless face of the currently observed subject.

If the supervisor has labelled an input "mouth" file as "missing", function F3 assigns $\boldsymbol{CM} = \boldsymbol{P(x_{neutral})} * 100$ to the data $x_{neutral}$ constituting this file. $\boldsymbol{P(x_{neutral})}$ is the probability (calculated according to formula *(3)*) that no AU whose activation induces a change in the facial appearance of the mouth has been activated. Since there are 22 distinct AUs and more than 350 different combinations of these (allowed by the co-occurrence rules defined in FACS and affecting the appearance of the mouth) which can be uniquely recognised based on the observed relevant deformations of the dual-view face model (Table 5.5, Table 5.7), a certainty measure $\boldsymbol{CM} \approx 0$, say $\boldsymbol{CM} = 0.25$ (i.e. 100/400), will be assigned to the data constituting an "mouth" file labelled as "missing".

Depending on the number of different mouth detectors integrated into the Facial Data Extractor ($j \geq 1$), function F3 executes the same inter-file consistency check as that defined for the class of input "nose/chin" files. The only two differences are:

- Function $\boldsymbol{S}$ exploited in step 2 of the algorithm is defined in the case of input "mouth" files as given in the following formula:
  $S(x_{detector1}, x_{detector2}) = 100*\boldsymbol{avg}(\forall i \in [1,4] | \boldsymbol{sigm}(\boldsymbol{d}(\boldsymbol{Pi} \in x_{detector1}, \boldsymbol{Pi} \in x_{detector2});0,2,3))$,
  where $S(x_{detector1}, x_{detector2})$ is the measure of similarity between the results of two nose/chin detectors in question, $\boldsymbol{Pi}$ is one of the frontal-view face-model points I, J, K or L (Figure 5.2, Table 5.4), $\boldsymbol{d}(p_1, p_2)$ is the block distance between the points p_1 and p_2, and $\boldsymbol{sigm}(y; \alpha, \beta, \gamma)$ is a Sigmoid membership function in *(1)*.
- The inter-file consistency check which compares the result of the examined mouth detector with the result of the selected / available profile detector (step 4 of the algorithm) is based on a comparison between the distance KL and the distance P6P8. Since a displayed opening of the mouth is observable in the frontal-view as well as in the profile-view facial image of the current expression and given that all of the single-session images are scale and orientation invariant (section 4.1), the proportion $\mathrm{KL}_{\mathrm{examined}} / \mathrm{KL}_{\mathrm{neutral}} = \mathrm{P6P8}_{\mathrm{examined}} / \mathrm{P6P8}_{\mathrm{neutral}}$ should hold (at least if AU28, AU28t, and AU28b have not been activated; otherwise one or both points P6 and P8 will be absent, see Table 5.7). If AU28, AU28t, and AU28b have not been activated and if the measure of similarity between the examined mouth detector's result and the selected profile detector's result is high, the confidence in the mouth detector's result is increased. This is expressed by the following formula, which computes a final data certainty $\boldsymbol{DC}_{detector1}$ of the examined mouth detector:

$$DC_{detector1} = \begin{cases} avg\left(CM_{detector1}, S\left(KL_{detector1}, P6P8_{selected\_profile}\right)\right) & CM < s \\ CM_{detector1} & CM \geq s \end{cases}, \text{ and}$$

$$S(KL_{detector1}, P6P8_{selected\_profile}) = 100*\boldsymbol{sigm}\left(\left|\frac{KL_{detector1}}{KL_{neutral}} - \frac{P6P8_{selected\_profile}}{P6P8_{neutral}}\right|;0,2,3\right)$$

is the measure of similarity between the examined mouth detector's result and the selected profile detector's result while $\boldsymbol{sigm}(y; \alpha, \beta, \gamma)$ is a Sigmoid membership function given in formula *(1)*.

5.5 Encoding and quantification of facial actions

The main goal of the Facial Action Encoder part of ISFER is to achieve a quantified facial action coding applicable to automated FACS coding for an input facial image. So, once the supervisor of the Facial Action Encoder has abridged and estimated the approximativeness of the input data generated by the Facial Data Extractor part of ISFER, it will try to reach the next predefined goal: to encode and quantify the displayed AUs. To this end, the supervisor selects from the action space three predefined plans that are sequentially executed:

1. Calculate the face model deformations and their certainty factors *CF* based on the data constituting the files selected by function F3.
2. Retrieve the appropriate data from the *DB of the extreme model deformations*.
3. Activate function F4 in order to perform reasoning with uncertainty about the activated AUs and the intensity of that activation.

Neutral facial expression

The information extracted from a neutral facial expression of the currently observed subject is necessary for both dealing with imperfect input data resulting from the Facial Data Extractor part of ISFER (section 5.4) and encoding and quantifying the displayed AUs based upon that data (this section). A successful processing of the Facial Action Encoder part of ISFER implies, therefore, that the information about the current person's neutral facial expression is available and stored in the "neutral expression" file of the DB of all encountered facial expressions.

In order to ensure correct extraction of the necessary data from an acquired image of the neutral expression of the currently observed subject, the Facial Data Extractor is invoked in the stand-alone operating mode (section 4.2) for the acquired image and each of the detectors belonging to the extraction group of the Facial Data Extractor's modules (section 4.3). The obtained results are visually inspected and the detectors producing the most accurate spatial sampling of the prominent facial features are selected and then connected in a new network of modules (e.g. as illustrated in Figure 4.7). The "neutral expression" file is then generated by executing the newly designed network of modules and activating the relevant rules for extracting the model points from the localised contours of the facial features (Table 5.6 and/or Table 5.4).

If the invoked detectors have failed to spatially sample a prominent facial feature, the current subject's neutral expression is to be acquired and processed once more. For example, if the detectors integrated into the Facial Data Extractor fail to spatially sample one or more facial features due to improper lighting, the monitoring device employed by ISFER (Figure 4.2) is to be biased by adjusting the current positions of the lamps mounted on the device and the current subject's neutral expression is to be acquired and processed once more.

Computing the face model deformations

Depending on the kind of the input image (i.e. frontal- or dual-view image) and using the relevant data about the neutral facial expression of the observed subject, the supervisor of the Facial Action Encoder calculates the appropriate face model deformations (frontal-view face-model deformations listed in the last column of Table 5.5 or dual-view face-model deformations listed in the last column of Table 5.5 and the last column of Table 5.7). For example, to calculate the face model deformation $\boldsymbol{BD_{deviation}}$, the supervisor applies the following formula:

$$\boldsymbol{BD_{deviation}} = \boldsymbol{BD_{examined}} - \boldsymbol{BD_{neutral}}, \text{ where } AB = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2},$$

where $\boldsymbol{B_{examined}}$ is extracted from the input "eye" file selected by function F3, $\boldsymbol{D_{examined}}$ is extracted from the input "eyebrow" file selected by function F3, $\boldsymbol{B_{neutral}}$ and $\boldsymbol{D_{neutral}}$ are extracted from "neutral expression" file stored in the DB of all encountered expressions.

The supervisor associates further a certainty factor ***CF*** with each calculated face model deformation. The certainties ***DC*** assigned by function F3 to the data constituting the selected files and used to calculate a particular face model deformation define the certainty factor ***CF*** associated with that model deformation. For instance, the certainty factor associated with the deformation $\boldsymbol{BD_{deviation}}$ will be calculated as $\boldsymbol{CF_{BD\_deviation}} = \boldsymbol{min(DC_{B\_examined}, DC_{D\_examined})}$, where $\boldsymbol{DC_{B\_examined}}$ is the data certainty assigned by function F3 to the data constituting the selected "eye" file and $\boldsymbol{DC_{D\_examined}}$ is the data certainty assigned by function F3 to the data constituting the selected "eyebrow" file.

If a dual-view facial image forms the input to the system and the certainty assigned by function F3 to the data constituting the "profile" input file is ***DC*** ≈ 0, the further processing of the system is the same as that for a frontal-view input facial image. However, considering the test results of the profile detector integrated into the Facial Data Extractor (Table 4.4), the probability that function F3 will assign ***DC*** ≈ 0 to the data constituting the "profile" file is very low. Similarly, if a dual-view facial image forms the input to the system and all "non-profile" input files have been labelled as "missing", the further processing of the system is the same as that for a profile-view facial image. This means that the supervisor will compute the profile-view face-model deformations listed in the last column of Table 5.7 and that the processing of function F4 will be solely based on the AU-recognition rules listed in Table 5.7. Finally, irrespective of the kind of an input image, if all files generated by the Facial Data Extractor part of ISFER have been labelled as "missing", the processing of the system terminates with:

```
The displayed expression is neutral (CF = 0.01%)
```

The certainty factor ***CF*** of this result calculated according to formula *(3)* represents the probability of the neutral expression, i.e., the probability that no AU has been activated. Since there are 32 distinct AUs and more than 10,000 different combinations of these that can be uniquely recognised based on the observed deformations of the dual-view face model (Table 5.8), the $CF \approx 100/10,000$ will be assigned to this result.

Database of extreme model deformations

The main goal for the development of ISFER is to achieve a fully automatic facial expression analysis. The intended analysis should further resemble automated FACS coding in digitised static input facial images so that it can serve the purpose of behavioural science investigations of the face. In other words, the system should encode facial actions in input images and quantify those codes (Donato et al. 1999, Bartlett et al. 1999).

As noted in sections 2.3 and 2.4, none of the systems for automatic facial expression analysis presented in the literature up to date quantifies the facial action codes on a 100-level intensity scale. This task is particularly difficult to accomplish for a number of reasons. First, FACS only provides five different AUs which can be assigned an intensity on a 3-level scale (section 5.1). Second, some facial actions such as blinking, winking, and sucking the lip(s) into the mouth are either encountered or not. It is not reasonable to describe a blink as having a "higher intensity" than another blink (for a full list of AUs, which can be recognised by ISFER but whose activation can be quantified merely on a 2-level intensity scale, see Table 5.9). Finally, each person displays a particular facial action with a different maximal intensity, which depends on his/her expressiveness as well as on the flexibility and strength of his/her facial muscles. Therefore, the aim was to design the Facial Action Encoder such that it facilitates a generic facial action classification (i.e. independent of the observed subject's sex, age and ethnicity) and a person-dependent quantification of the encoded AUs for which measuring of the activation intensity is "reasonable". This has been accomplished by encapsulating within the Facial Action Encoder part of ISFER the FACS's person-independent rules for AU recognition (Tables 5.5, 5.7, 5.8) and an AU codes quantification method which uses a *subject-profiled database of extreme face model deformations*.

Since the DB of extreme model deformations should facilitate person-dependent quantification of the encoded AUs, it should be subject-profiled; it should be furnished with the extreme face model deformations of the currently observed subject. Thus, each time before a session with a new subject starts, the DB of the extreme model deformations is initialised. The subject is asked to display with a maximal intensity a representative set of facial expressions, so called *individual-extreme-displays set* (IEDS), which is further processed by the system in order to measure the relevant model deformations (Table 5.9) and their certainty factors ***CF*** that will constitute the DB of extreme model deformations. The IEDS consists of 6 basic emotional expressions (fear, happiness, sadness, surprise, disgust and anger; Ekman and Friesen 1975), which are to be displayed according to the rules given in Table 5.10, and 7 maximal displays of AU8, AU18, AU23, AU24, AU27, AU39, and AU41. The 13 expressions of the IEDS are sufficient for an initial furnishing of the DB of extreme model deformations since they reveal the maximal model deformations coupled with any AU that can be recognised by ISFER and whose activation can be quantified on a 100-level intensity scale (Table 5.9).

Table 5.9
Quantification of the AU codes that can be recognised by ISFER (Table 5.8)

AU	Description	Quantification	AU	Description	Quantification
1	Raised inner eyebrow(s)	100-level scale, based on $BD_{deviation}$	20	Stretched mouth (horiz.)	100-level scale, based on $IJ_{deviation}$
2	Raised outer eyebrow(s)	100-level scale, based on $AE_{deviation}$	23	Tightened lips	100-level scale, based on $KL_{deviation}$
4	Eyebrows drawn together	100-level scale, based on $DD1_{deviation}$	24	Pressed lips	100-level scale, based on $KL_{deviation}$
5	Raised upper eyelid(s)	100-level scale, based on $FG_{deviation}$	25	Parted lips (max = $AU8_{max}$)	100-level scale, based on $P6P8_{deviation}$
6	Raised cheek(s)	100-level scale, based on $AU12_{intensity}$	26	Jaw dropped	100-level scale, based on $P4P10_{deviation}$
7	Raised lower eyelid(s)	100-level scale, based on $GX_{deviation}$	27	Stretched mouth (vertic.)	100-level scale, based on $P4P10_{deviation}$
8	Lips towards each other	100-level scale, based on $P5P6_{deviation}$	28	Sucked lips into the mouth	2-level scale
9	Wrinkled nose	100-level scale, based on $P2\text{-}P3_{deviation}$	28b	Bottom lip sucked in	2-level scale
10	Raised upper lip	100-level scale, based on $P5P6_{deviation}$	28t	Upper lip sucked in	2-level scale
12	Raised mouth corner(s)	100-level scale, based on $IB_{deviation}$	29	Jaw pushed forward	2-level scale
13	Sharp AU12 (max = $AU12_{max}$)	100-level scale, based on $IB_{deviation}$	35	Cheeks sucked in	2-level scale
15	Depressed mouth corner(s)	100-level scale, based on $IB_{deviation}$	36b	Tongue under the lower lip	2-level scale
16	Depressed lower lip	100-level scale, based on $P8P10_{deviation}$	36t	Tongue under the upper lip	2-level scale
17	Raised chin	2-level scale	38	Wide nostrils (max = $AU9_{max}$)	100-level scale, based on $H'H1'_{deviation}$
18	Puckered lips	100-level scale, based on $IJ_{deviation}$	39	Compressed nostrils	100-level scale, based on $H'H1'_{deviation}$
19	Shown tongue	2-level scale[3]	41	Dropped upper eyelid	100-level scale, based on $FX_{deviation}$

[3] The intensity of AU19 activation could be measured based on the level to which the tongue is protruding out of the mouth. However, the rule used to recognise AU19 activation is merely based on detection of two valleys and a peak between the profile points P6 and P8 (Table 5.7). Hence, there is no model deformation whose intensity (and not merely its existence) could be used to compute the intensity of AU19 activation.

Table 5.10
AU-coded representation of 6 basic emotional expressions (Pantic and Rothkrantz 2000b)

Expression	AU-coded description
happiness	AU6 + AU12 + (AU25 or AU26)
sadness	AU1 + AU4 (with or without AU7) + AU15 + AU17 + (AU25 or AU26)
surprise	AU1 + AU2 + AU5 (without AU7) + AU26
disgust	AU9 (with or without AU17) + (AU25 or AU26)
fear	AU1 + AU4 + AU5 + AU7 + AU20 + (AU25 or AU26)
anger	AU2 + AU4 (with or without AU7) + AU10 + AU16 + (AU25 or AU26)

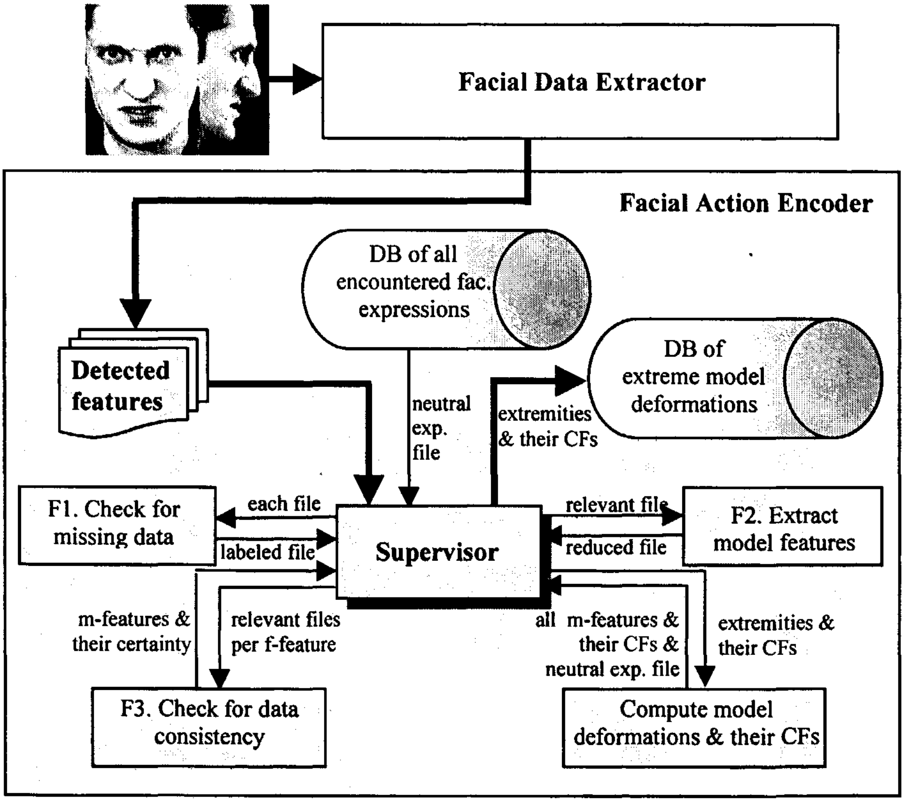

Facial Data Extractor
Facial Action Encoder
DB of all encountered fac. expressions
DB of extreme model deformations
Detected features
neutral exp. file
extremities & their CFs
F1. Check for missing data
each file
labeled file
Supervisor
relevant file
reduced file
F2. Extract model features
m-features & their certainty
relevant files per f-feature
all m-features & their CFs & neutral exp. file
extremities & their CFs
F3. Check for data consistency
Compute model deformations & their CFs

Figure 5.10: Initial furnishing of the DB of extreme model deformations

The initial furnishing of the DB of extreme model deformations is performed off-line. A set of 26 images showing two IEDS displayed by 2 different trained FACS-coders is shown to the novel subject who is further asked to display his/her own IEDS according to the provided example images. The processing of the system, algorithmically illustrated in Figure 5.10, is then invoked for each of the acquired images. The DB of extreme model deformations is altered on-line however. As noted above, each time a new facial image of the currently observed subject is entered into the system for analysis, the supervisor of the Facial Data Extractor retrieves the values constituting the DB of extreme model deformations in order to quantify the displayed AUs. If a currently computed model deformation x equals the related extremity ***x_extreme*** stored in the DB of extreme model deformations and $CF_x > CF_{x\_extreme}$, the supervisor adjusts the content of the DB accordingly. Hence, even if the initial values constituting the DB of extreme model deformations have not been measured accurately or the observed subject was reluctant or unable to display the IEDS with maximal intensity, the system is enabled to learn the subject-dependent parameters necessary for quantifying the encoded AUs.

As mentioned in section 5.2, the employment of a large database is coupled with high memory/storage requirements and long retrieval times. However, since each subject-profiled DB of extreme model deformations contains merely 20 different variables (Table 5.9), the efficiency of storage and retrieval is not an issue here.

F4: Encoding and quantifying the displayed AUs

The supervisor of the Facial Action Encoder activates function F4 in order to accomplish reasoning with uncertainty about the displayed facial actions and their intensities. Since it performs the main task of the Facial Action Encoder, function F4 forms its kernel. Broadly speaking, it encodes and quantifies the displayed facial actions based on the calculated face-model deformations and according to the mapping between 32 FACS rules and 32 face-model-based rules given in Tables 5.5, 5.7 and 5.8. Actually, function F4 infers appropriate conclusions and their certainties about the displayed AUs and their intensities based on both the internally stored face-model-based rules illustrated in Figure 5.12 (for a complete list of utilised functions, thresholds, and rules, the reader is referred to Appendix B) and a set of facts stored in a so-called blackboard and provided by the supervisor of the Facial Action Encoder (Figure 5.11).

Each of the rules utilised by function F4 encodes and quantifies a single AU based on the existence and the extent of a particular discrepancy of the spatial arrangement of the model points between the current and the neutral expression of the observed subject (Figure 5.12). As explained in section 5.3, the rules given in Tables 5.5, 5.7, 5.8, and hence in Figure 5.12 are uniquely defined in the sense that each model deformation corresponds to a unique set of AU codes. A relational list (R-list, see also section 3.2) has been utilised to represent the relations between these rules. The utilised R-list is a 4-tuple list, where the first two columns identify the conclusion clause of a certain rule that forms the premise clause of another rule identified in the next two columns of the R-list (Figure 5.12).

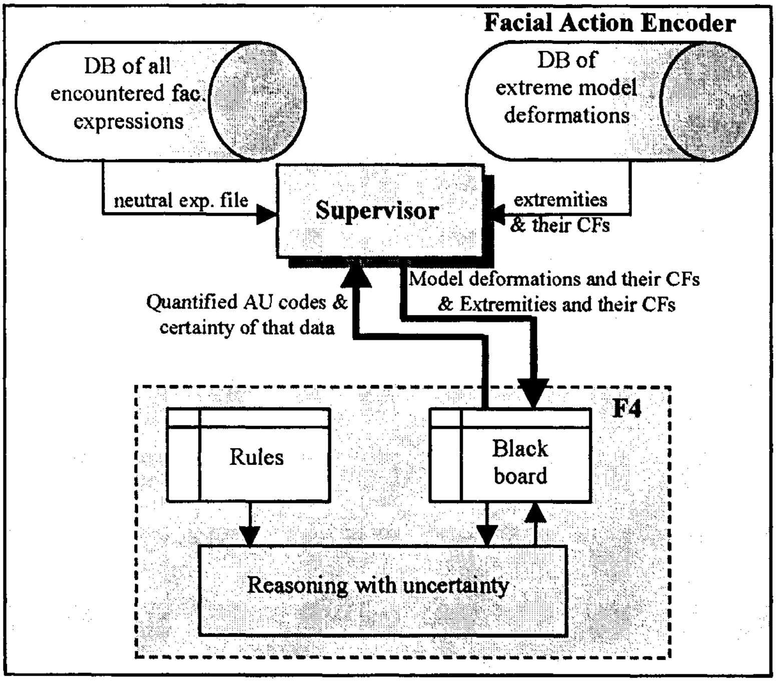

Facial Action Encoder
DB of all encountered fac. expressions
DB of extreme model deformations
neutral exp. file
Supervisor
extremities & their CFs
Model deformations and their CFs & Extremities and their CFs
Quantified AU codes & certainty of that data
F4
Rules
Black board
Reasoning with uncertainty

Figure 5.11: Function F4 of the Facial Action Encoder part of ISFER

Function F4 applies *fast direct chaining* as its inference procedure (Schneider et al. 1996; see also section 3.2). In other words, it starts with the first internally stored rule and then searches the R-list to find if the conclusion of the fired rule forms a premise of another rule that will be fired in the next loop. If such a relation does not exist, function F4 will try to fire the rule that in the internal storage comes after the rule last fired. In order to prevent function F4 from firing a rule more than once, a *list of fired rules* (*LFR*) is utilised. Thus, if a rule has fired (i.e. the certainty factor of the premise $CF_p \in [0, 100]$ of the rule is greater than or equal to a threshold *T*), the rule number is added to the LFR.

The overall certainty factor CF_p of the premise *p* of a fired rule is calculated as:

1. For the portion of the premise *p* that contains clauses *c1* and *c2* related as *c1* AND *c2*, $CF_p = min\ (CF_{c1}, CF_{c2})$.
2. For the portion of the premise *p* that contains clauses *c1* and *c2* related as *c1* OR *c2*, $CF_p = max\ (CF_{c1}, CF_{c2})$.
3. For the portion of the premise *p* that contains just clause *c*, $CF_p = CF_{c1}$.

The certainty factor $\boldsymbol{CF_c}$ of a premise clause $\boldsymbol{c}$ is calculated in the following way:

1. For a premise clause $\boldsymbol{c}$ of a kind "NOT $\boldsymbol{AU_i}$", $\boldsymbol{CF_c = CF_{cc1}}$, where $\boldsymbol{CF_{cc1}}$ is the certainty factor of the first conclusion clause of the rule encoding $\boldsymbol{AU_i}$.
2. For a premise clause $\boldsymbol{c}$ where the existence or the location of a particular profile-view face-model point $\boldsymbol{P_i}$ is examined (for the examples see Table 5.7), $\boldsymbol{CF_c = DC_{Pi}}$ where $\boldsymbol{DC_{Pi}}$ is the data certainty assigned by function F3 to the data constituting the selected "profile" file.
3. For a premise clause $\boldsymbol{c}$ where the shape of a profile-view face-model deformation $\boldsymbol{x}$ is examined (for the examples see Table 5.7), $\boldsymbol{CF_c = DC_x}$, where $\boldsymbol{DC_x}$ is the data certainty assigned by function F3 to the data constituting the selected "profile" file.
4. For a premise clause $\boldsymbol{c}$ where a model deformation $\boldsymbol{x}$ is compared to a threshold $\boldsymbol{t_i}$, $\boldsymbol{CF_c = min\ (CF_x, CF_{ti})}$, where $\boldsymbol{CF_x}$ is the certainty factor of the model deformation $\boldsymbol{x}$ and $\boldsymbol{CF_{ti}}$ is the certainty factor of the extreme model deformation related to the threshold $\boldsymbol{t_i}$ (for the examples see Figure 5.12).
5. For a premise clause $\boldsymbol{c}$ where a model deformation $\boldsymbol{x}$ is compared to zero, $\boldsymbol{CF_c = CF_x}$, where $\boldsymbol{CF_x}$ is the certainty factor of the model deformation $\boldsymbol{x}$.

If the overall certainty factor $\boldsymbol{CF_p}$ of the premise $\boldsymbol{p}$ of a rule is $\boldsymbol{CF_p \geq T}$, the rule will be fired. The value of the threshold $\boldsymbol{T}$ is set as follows:
$\boldsymbol{T = min(min(DC_{profile}), min(DC_{eye}), min(DC_{eyebrow}), min(DC_{nose/chin}), min(DC_{mouth}))}$, where $\boldsymbol{min(DC_x)}$ is the minimal data certainty that can be assigned by function F3 to data $\boldsymbol{x}$ constituting the given file (see section 5.4). The threshold $\boldsymbol{T}$ is set in such a manner because this enables the Facial Action Encoder part of ISFER to potentially encode all displayed AUs. In other words, the design of the system is such that it can encode all displayed AUs even if the reached conclusions might have low certainties (due to low certainties of the relevant input data).

The conclusion part of each of the internally stored rules, exemplified in Figure 5.12, consists of two conclusion clauses. The first conclusion clause, ***cc1***, implies that a certain $\boldsymbol{AU_i}$ has been activated, while the second conclusion clause, ***cc2***, implies that the $\boldsymbol{AU_i}$ in question has been activated with a particular intensity $\boldsymbol{I(AU_i)}$. Instead of computing an overall certainty factor $\boldsymbol{CF_{con} = min\ (CF_{cc1}, CF_{cc2})}$ of the conclusion ***con*** of a fired rule, function F4 calculates the certainty factor $\boldsymbol{CF_{cc}}$ separately for each of the conclusion clauses. Unfolding the certainties associated with each of the inferred conclusions on encoding and quantification of the displayed AUs provides the user with insight into the system's performance: it enables the user to estimate the quality of the internally stored data (the model-deformation extremities, the neutral expression data) and successfulness of the invoked detectors vs. the quality of the acquired images. One can use this knowledge to enhance the performance of the system by acquiring more accurate data and/or adjusting the monitoring device used to acquire the facial images of the observed subject.

f-on 1	***extent** (x)* = 100 * ***sigm*** (|x|; 0, ½ |$x_{extreme}$|, |$x_{extreme}$|), where x is a particular face model deformation currently computed by the supervisor and forwarded to function F4 (Figure 5.11), $x_{extreme}$ is retrieved from the DB of extreme model deformations and forwarded to function F4 by the supervisor (Figure 5.11), and ***sigm***(y; α, β, γ) is a Sigmoid membership function given in formula *(1)*.
f-on 2	$max(x,y)=\begin{cases} x & x\geq y \\ y & x<y \end{cases}$

var 1	***t1*** = $x_{extreme}$, where x = IJ$_{deviation}$ and $x_{extreme}$ encountered by maximal AU18
⋮	⋮
var 4	***t4*** = $x_{extreme}$, where x = P4P10$_{deviation}$ and $x_{extreme}$ encountered by maximal AU26 (see “surprise” in Table 5.10)

rule 1	If BD$_{deviation}$ > 0 OR B1D1$_{deviation}$ > 0 Then AU1 AND ***I*** *(*AU1*)*; ***I*** *(*AU1*)* = ***max (extent (***BD$_{deviation}$*)*, ***extent (***B1D1$_{deviation}$*))*
rule 2	If AE$_{deviation}$ > 0 OR A1E1$_{deviation}$ > 0 Then AU2 AND ***I*** *(*AU2*)*; ***I*** *(*AU2*)* = ***max (extent (***AE$_{deviation}$*)*, ***extent (***A1E1$_{deviation}$*))*
rule 3	If DD1$_{deviation}$ < 0 AND NOT AU9 Then AU4 AND ***I*** *(*AU4*)*; ***I*** *(*AU4*)* = ***extent (***DD1$_{deviation}$*)*
⋮	⋮
rule 5	If AU12 OR AU13 Then AU6 AND ***I*** *(*AU6*)*; ***I*** *(*AU6*)* = ***I*** *(*AU12 OR AU13*)*
⋮	⋮
rule 8	If P2-P3$_{deviation}$ decreased Then AU9 AND ***I*** *(*AU9*)*; ***I*** *(*AU9*)* = ***extent (***P2-P3$_{deviation}$*)*
⋮	⋮
rule 10	If (IB$_{deviation}$ < 0 AND CI$_{deviation}$ > 0) OR (JB1$_{deviation}$ < 0 AND CJ$_{deviation}$ > 0) Then AU12 AND ***I*** *(*AU12*)*; ***I*** *(*AU12*)* = ***max (extent (***IB$_{deviation}$*)*, ***extent (***JB1$_{deviation}$*))*
⋮	⋮
rule 27	If IJ$_{deviation}$ < ***t1*** Then AU35 AND ***I*** *(*AU35*)* = 100
⋮	⋮
rule 32	If ((FG > 0 AND FG$_{deviation}$ < 0 AND FX$_{deviation}$ < 0) OR (F1G1 > 0 AND F1G1$_{deviation}$ < 0 AND F1Y$_{deviation}$ < 0)) AND NOT AU7 Then AU41 AND ***I*** *(*AU41*)*; ***I*** *(*AU41*)* = ***max (extent (***FX$_{deviation}$*)*, ***extent (***F1Y$_{deviation}$*))*

Conclusion Clause		***Premise Clause***	
Rule #	*Clause #*	Rule #	*Clause #*
8	1	3	2
10	1	5	1
⋮	⋮	⋮	⋮

Figure 5.12: Functions, variables, rules (given in Tables 5.5, 5.7 and 5.8) and R-list representation of the relations among the rules utilised by function F4 (a complete list is provided in Appendix B)

The certainty factors $\boldsymbol{CF_{cc1}}$ and $\boldsymbol{CF_{cc2}}$ of the conclusion clauses ***cc1*** and ***cc2*** are calculated in the following manner:

1. $\boldsymbol{CF_{cc1} = CF_p}$, where $\boldsymbol{CF_p}$ is the overall certainty factor of the premise $\boldsymbol{p}$ of the relevant rule that has been fired.
2. $\boldsymbol{CF_{cc2} = min(CF_p, CFx_{extreme})}$, where $\boldsymbol{CF_p}$ is the overall certainty factor of the premise $\boldsymbol{p}$ of the relevant rule that has been fired, $\boldsymbol{x_{extreme}}$ is retrieved from the DB of extreme model deformations and used by function F4 to calculate the intensity $\boldsymbol{I(AU_i)}$ of the relevant $\boldsymbol{AU_i}$, and $\boldsymbol{CFx_{extreme}}$ is the certainty factor associated with the $\boldsymbol{x_{extreme}}$.

5.6 Dealing with partial data

If a certain prominent facial feature is not detected or is detected highly inaccurately by the detectors integrated into the Facial Data Extractor, the supervisor of the Facial Action Encoder generates for that facial feature a file labelled as “missing” and stores there the relevant data extracted from the neutral expression of the currently observed subject (section 5.4). The substitution of imperfect input data with the relevant data constituting the “neutral expression” file, which is stored in a personalised database of all encountered facial expressions of the currently observed person, implies that the exact information about the presently examined facial expression is lost. Dealing with this information loss, that is, handling the partial data resulting from the antecedent processing of the system, is the final goal that the supervisor of the Facial Action Encoder will try to reach before it forwards the accomplished results to the Facial Expression Classifier part of ISFER (Figure 5.1).

As explained in section 3.4, a way of dealing with partial data resulting from the preceding processing of the system is to exploit the process knowledge that is based on the knowledge about the personal patterns of facial behaviour of the currently observed subject. Since people usually display some (typical) facial expressions more often than some others, the patterns / typicality of the displayed facial behaviour can be viewed as the frequency with which that facial behaviour is shown. Yet, people have various levels of facial expressiveness and the set of someone’s typical facial expressions varies depending on the person’s personality, cultural and social background, etc. Thus, rather than having a priori (generic) rules for dealing with partial data, the appropriate (personalised) rules can be learned and the underlying statistical model of the current subject’s facial behaviour can be devised by watching the subject and recording the observed facial patterns. This reflects a situation in which the properties used to associate the elements in a set $\boldsymbol{V}$ (the input data set) with those in a set $\boldsymbol{U}$ (the AU codes set) are uncertain (due to the encountered partial data) but we know the process (recording the frequency of each expression) used to select the properties (the typicality of a facial expression) which play a role in the association (select the facial expression having the highest typicality and the AU-coded description AU_1 + ... + AU_i, where AU_i encodes the appearance of the "missing" facial feature).

Within the Facial Action Encoder, function F5 employs this process knowledge and handles the problem created by encountered partial data using the data resulting from the antecedent processing of ISFER and the data stored in the subject-profiled DB of all encountered facial expressions (Figure 5.13).

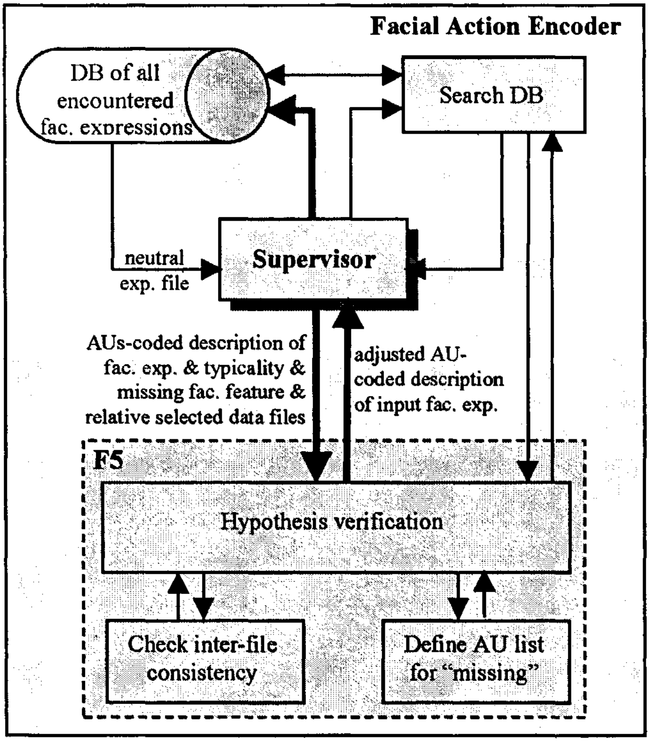

Facial Action Encoder
DB of all encountered fac. expressions
Search DB
neutral exp. file
Supervisor
AUs-coded description of fac. exp. & typicality & missing fac. feature & relative selected data files
adjusted AU-coded description of input fac. exp.
F5
Hypothesis verification
Check inter-file consistency
Define AU list for "missing"

Figure 5.13: Function F5 of the Facial Action Encoder part of ISFER

Database of all encountered facial expressions

Since the DB of all encountered facial expressions should store the patterns of facial behaviour of the currently observed subject, it should be subject-profiled. Thus, each time prior to a session with a novel subject, the DB of all encountered facial expressions is supplied merely the "neutral expression" file of the current subject (see section 5.5). If the subject has been subjected to the system's analysis before, the DB of all encountered facial expressions used in the antecedent sessions with that subject will be employed anew.

Bearing in mind that the DB of all encountered facial expressions should keep records of all expressions ever encountered while monitoring a particular subject for whom it was defined (section 5.6), efficient organisation of such a potentially large database is crucial for the accomplishment of an acceptable level of system performance (Kan 1995). In principle, a database is considered well organised if its organisation facilitates accurate and efficient retrieval of the stored data. Having accurate retrieval guarantees that the desired datum will be retrieved. Having efficient retrieval guarantees that data will be retrieved fast enough to give acceptable system response times. These two factors are inversely proportional however. Namely, it is easy to guarantee accurate retrieval at the expense of efficiency (e.g. by searching the whole database for the best matching case) and easy to have fast retrieval if only a fraction of the utilised database is searched (possibly missing the desired datum/data). In order to organise the DB of all encountered facial expressions such that it keeps these two factors in balance, that is, such that it guarantees accurate retrieval of the desired datum in a relatively short time, a clustered database organisation was adopted.

In general, the DB of all encountered facial expressions is divided into 32 clusters corresponding to the 32 AU codes that the Facial Action Encoder is able to recognise from an input dual-view facial image. Each cluster is split further into a number of partitions corresponding to the number of different AU codes forming the AU-coded descriptions of the facial expressions classified into the pertinent cluster. Of course, the actual number of clusters and accompanying partitions constituting a specific subject-profiled DB of all encountered facial expressions depends on the variety of expressions that the pertinent subject displayed during his/her sessions with ISFER. Anyway, for each facial expression displayed by the currently observed subject, there is a file (record) stored in his/her personalised DB of all encountered facial expressions constituting the AU-coded description of that facial expression and the frequency with which the subject has displayed that expression. Each of these files is stored as a part of the appropriate cluster and its appropriate partition, which is decided by the "smallest" AU code of the AU-coded description of the pertinent facial expression (e.g. AU1+AU2 will be classified into the AU1 cluster) and by the number of different AUs forming the AU-coded description of that expression (e.g. AU1+AU2 will be classified into the partition "2" of AU1 cluster) respectively. For each expression ***e*** analysed by the system during a single session, the supervisor of the Facial Action Encoder adjusts the contents of the currently employed DB of all facial expressions in the following manner (Figure 5.13):

1. If the currently examined facial expression ***e*** has been previously analysed by the system, the supervisor uses the function *Search DB* to retrieve the file ***e***-file from the database (this function decides the relevant cluster based upon the "smallest" AU code forming a part of the AU-coded description of ***e*** and the relevant partition based upon the number of different AU codes forming the AU-coded description of ***e***), enlarges the total number of ever encountered expressions N_{new}

= N_{old} + 1, calculates the new frequency $F_{e\text{-}new} = (1 + N_{old} * F_{e\text{-}old}) / N_{new}$, adjusts the ***e***-file accordingly and stores the new ***e***-file into the database.

2. If the currently examined facial expression ***e*** has not been previously analysed by the system, then the supervisor enlarges the total number of ever encountered expressions $N_{new} = N_{old} + 1$, calculates the frequency $F_e = 1 / N_{new}$, generates the *e*-file (supplies it with the AU-coded description of ***e*** and the frequency F_e), and stores it into the proper cluster (and its proper partition) of the database.

F5: Post-processing aimed at dealing with partial data

For each file labelled as "missing" and selected by function F3 (as the only choice; see section 5.4), the supervisor of the Facial Action Encoder activates function F5 in order to handle this partial data resulting from the antecedent system's processing. Depending on the facial feature *ff*, whose spatial sampling failed completely (i.e. none of the detectors integrated into the Facial Data Extractor detected the feature *ff* successfully) and whose related *ff*-file has been labelled as "missing", function F5 carries out the procedure explained here in general and illustrated in Figure 5.13:

1. Define the list of AUs ***AU-list*** holding each and every AU that codes a possible appearance of the facial feature *ff* (Table 5.11).
2. From this list exclude all AUs that are already included in the AU-coded description $d(e_{input})$ of the currently examined expression e_{input} forming the input to function F5.
3. Perform an inter-file consistency check and place a hypothesis about the appearance $d(ff)$ of the facial feature *ff* given in terms of AUs.
4. Verify the hypothesis based on the typicality of the expression $e_{adjusted}$ having the AU-coded description $d(e_{adjusted}) = d(e_{input}) + d(ff)$. The hypothesis is verified if $Fe_{adjusted} > Fe_{input}$, where $Fe_{adjusted}$ is the frequency of $e_{adjusted}$ and Fe_{input} is the frequency of e_{input} retrieved from the DB of all encountered expressions by function *Search DB*.
5. If the hypothesis has not been verified, return $d(e_{adjusted}) = d(e_{input})$ to the supervisor. Otherwise, return $d(e_{adjusted}) = d(e_{input}) + d(ff)$.

Table 5.11
Possible AU-coding of the facial feature *ff* labelled as "missing" (Table 5.8)

ff-file labelled as "missing"	Possible AU-coding
"eye" file	AU5, AU6, AU7, AU41
"eyebrow" file	AU1, AU2, AU4, AU9
"nose/chin" file	AU26, AU27, AU38, AU39
"mouth" file	AU8, AU10, AU12, AU13, AU15 to AU20, AU23 to AU28, AU28b, AU28t, AU29, AU35, AU36t, AU36b

As mentioned in section 5.4 and considering the test results of the profile detector (Table 4.4), the probability that the "profile" file will be labelled as "missing" is very low. Therefore, this case has been excluded from the processing of function F5.

Post-processing of the "eye" files

If one of the "eye" files has been labelled as "missing" while the data $\boldsymbol{x}$ constituting the other "eye" file have been assigned a data certainty $\boldsymbol{DC_x} > \boldsymbol{DC_{x\text{-}neutral}}$, where $\boldsymbol{DC_{x\text{-}neutral}}$ = 100/24 (section 5.4), the supervisor of the Facial Action Encoder will not activate function F5. The underlying reasoning is based upon the fact that the eyes are bilateral facial features and although the facial muscles affecting the facial appearance of the eyes may be activated unilaterally, affecting just one of the eyes, most of the times those activations are bilateral, affecting both eyes. Furthermore, each AU that codes the facial appearance of the eyes (Table 5.11) is scored unilaterally – if an appropriate model deformation is extracted from either of the eyes' contours, the AU is scored (Table 5.5). Hence, by activating function F5, the system's processing will assume in fact that two different AUs have been activated unilaterally, each one affecting different eye. This is highly improbable taking once more into consideration that the AUs affecting the facial appearance of the eyes are usually activated bilaterally.

If both "eye" files were labelled as "missing", function F5 executes the procedure outlined above for a general case with the following adjustments to steps 3 and 4:

- Since there is no file supplied with data that could be used to control the correctness of the hypothesis that a certain AU affecting the appearance of the eyes has been activated, no inter-file consistency check (step 3) can be performed while the "eye" files are post-processed. For each $\boldsymbol{AU_i} \in \boldsymbol{AU\text{-}list}$ and each anatomically possible combination of these, a hypothesis is placed that the AU or the AU combination in question is present. In total, there are 7 distinct hypotheses at most: AU5, AU6, AU7, AU41, AU5+AU6, AU5+AU7, AU6+AU41.
- The hypothesis verification (step 4) is performed for each hypothesis placed in step 3. If a hypothesis $\boldsymbol{h}$ is verified but not all hypotheses from step 3 have been checked, the hypothesis $\boldsymbol{h}$ and the frequency $\boldsymbol{Fe_{adjusted}}$ related to the expression $\boldsymbol{d(e_{adjusted})} = \boldsymbol{d(e_{input})} + \boldsymbol{h}$ will be stored temporarily. Once all hypotheses from step 3 have been checked, the hypothesis $\boldsymbol{h}$ stored together with the highest frequency $\boldsymbol{Fe_{adjusted}}$ is selected and $\boldsymbol{d(e_{adjusted})} = \boldsymbol{d(e_{input})} + \boldsymbol{h}$ is returned to the supervisor of the Facial Action Encoder.

Post-processing of the "eyebrow" files

The eyebrows are bilateral facial features and, similarly to the case of the eyes, the facial muscles affecting the facial appearance of the eyebrows are usually activated bilaterally. The only exception from this rule is AU2, which is often activated unilaterally. Thus, if one of the "eyebrow" files has been labelled as "missing" while the data x constituting the other "eyebrow" file have been assigned a data certainty $\boldsymbol{DC_x} > \boldsymbol{DC_{x\text{-}neutral}}$, where $\boldsymbol{DC_{x\text{-}neutral}}$ = 100/14 (section 5.4), function F5 executes the procedure outlined above for a general case with the following adjustments to steps **3** and **4**:

- Since no file consists of data that can be used to control the correctness of the hypothesis $\boldsymbol{h}$ that AU2 has been activated, no inter-file consistency check (step 3) can be performed.
- In the case that a hypothesis $\boldsymbol{h}$ is verified, the calculated $\boldsymbol{d(e_{adjusted})} = \boldsymbol{d(e_{input})} + \boldsymbol{h}$ is returned to the supervisor of the Facial Action Encoder.

If a dual-view facial image forms the input to the system and both "eyebrow" files have been labelled as "missing", function F5 executes the procedure outlined above for a general case with the following adjustments to steps **3** and **4**:

- Based on both the data constituting the "profile" file and the rules given in Table 5.7, check if any of $\boldsymbol{AU_i} \in \boldsymbol{AU\text{-}list}$ can be scored and place the hypothesis $\boldsymbol{h}$ accordingly. Except the hypothesis $\boldsymbol{h1} = \boldsymbol{h}$, place hypothesis, $\boldsymbol{h2} = \boldsymbol{h}$ + AU2.
- The hypothesis verification (step 4) is performed for both hypotheses from step **3** If the hypothesis $\boldsymbol{h1}$ is verified but the hypothesis $\boldsymbol{h2}$ has not been checked yet, the hypothesis $\boldsymbol{h1}$ and the frequency $\boldsymbol{Fe_{adjusted}}$ related to the expression $\boldsymbol{d(e_{adjusted})} = \boldsymbol{d(e_{input})} + \boldsymbol{h1}$ are stored temporarily. Once the hypothesis $\boldsymbol{h2}$ has been checked, the hypothesis $\boldsymbol{h}$ stored together with the highest frequency $\boldsymbol{Fe_{adjusted}}$ is selected and $\boldsymbol{d(e_{adjusted})} = \boldsymbol{d(e_{input})} + \boldsymbol{h}$ is returned to the supervisor.

If a frontal-view facial image forms the input to the system and both "eyebrow" files have been labelled as "missing", function F5 executes the procedure outlined above for a general case with the following adjustments to steps **3** and **4**:

- Based on in the AU-coded description $\boldsymbol{d(e_{input})}$ of the currently examined expression $\boldsymbol{e_{input}}$ that forms the input to function F5, check if AU10 has been scored. If not, remove AU9 from the $\boldsymbol{AU\text{-}list}$. The underlying reasoning is based upon the knowledge contained in FACS. Namely, AU9 and AU10 cause the same facial appearance of the mouth but AU9 obscures AU10 (section 5.3). Hence, if AU10 has been scored, AU9 might be present, but if AU9 has been scored, AU10 cannot be scored. Since no file consists of data that can be used to control the correctness of the hypothesis that a certain AU affecting the facial appearance of the eyebrows has been activated, no inter-file consistency check (step **3**) can be performed while the "eyebrow" files are post-processed. For each $\boldsymbol{AU_i} \in \boldsymbol{AU\text{-}list}$ and each anatomically possible combination of these, a hypothesis is placed that the AU or the AU combination in question is present. In total, there are 9 distinct hypotheses at most: AU1, AU2, AU4, AU9, AU1+AU2, AU1+AU4, AU2+AU4, AU2+AU9, AU1+AU2+AU4.
- The hypothesis verification (step-**4**) is performed for each hypothesis from step **3**. If a hypothesis $\boldsymbol{h}$ is verified but not all hypotheses from step **3** have been

checked, the hypothesis $\boldsymbol{h}$ and the frequency $\boldsymbol{Fe_{adjusted}}$ related to the expression $\boldsymbol{d(e_{adjusted})} = \boldsymbol{d(e_{input})} + \boldsymbol{h}$ will be stored temporarily. If the two hypotheses $\boldsymbol{h_{AU9}}$ containing AU9 are placed in step 3, their verification is postponed until all other hypotheses placed in step 3 are checked. Then, if a $\boldsymbol{h_{AU9}}$ is verified, it will be temporarily stored together with the frequency $\boldsymbol{Fe_{adjusted}}$ related to the expression $\boldsymbol{d(e_{adjusted})} = \boldsymbol{d(e_{input})}$ – AU10 + $\boldsymbol{h_{AU9}}$. Once all hypotheses from step-3 have been checked, the hypothesis $\boldsymbol{h}$ stored together with the highest frequency $\boldsymbol{Fe_{adjusted}}$ is selected and the related $\boldsymbol{d(e_{adjusted})}$ is returned to the supervisor of the Facial Action Encoder.

Post-processing of the "nose/chin" file

If a dual-view facial image forms the input to the system and the "nose/chin" file selected by function F3 has been labelled as "missing", function F5 executes the procedure outlined above for a general case with the following adjustments to steps 3 and 4:

- Based on both the data constituting the "profile" file and the rules given in Table 5.7, check if any of $\boldsymbol{AU_i} \in \boldsymbol{AU\text{-}list}$ can be scored and place the hypothesis $\boldsymbol{h}$ accordingly. Based on the AU-coded description $\boldsymbol{d(e_{input})}$ of the currently examined expression $\boldsymbol{e_{input}}$ that forms the input to function F5, check if any of AU8, AU9, AU10, AU12, AU13, AU15, AU18, AU20, AU24, AU28 has been scored. If so, remove AU38 from the ***AU-list*** (see Table 5.5). Except the hypothesis $\boldsymbol{h1} = \boldsymbol{h}$, place the hypothesis $\boldsymbol{h2} = \boldsymbol{h}$ + AU39 and, if AU38 $\in$ ***AU-list***, place the hypothesis $\boldsymbol{h2} = \boldsymbol{h}$ + AU38.
- The hypothesis verification (step 4) is performed for each hypothesis from step 3. If a hypothesis $\boldsymbol{h}$ is verified but not all hypotheses from step 3 have been checked, the hypothesis $\boldsymbol{h}$ and the frequency $\boldsymbol{Fe_{adjusted}}$ related to the expression $\boldsymbol{d(e_{adjusted})} = \boldsymbol{d(e_{input})} + \boldsymbol{h}$ will be stored temporarily. Once all hypotheses from step 3 have been checked, the hypothesis $\boldsymbol{h}$ stored together with the highest frequency $\boldsymbol{Fe_{adjusted}}$ is selected and $\boldsymbol{d(e_{adjusted})} = \boldsymbol{d(e_{input})} + \boldsymbol{h}$ is returned to the supervisor of the Facial Action Encoder.

If a frontal-view facial image forms the input to the system and the "nose/chin" file selected by function F3 has been labelled as "missing", function F5 executes the procedure outlined above for a general case with the following adjustments to steps 3 and 4:

- Based on the AU-coded description $\boldsymbol{d(e_{input})}$ of the currently examined expression $\boldsymbol{e_{input}}$ that forms the input to function F5, check if any of AU8, AU9, AU10, AU12, AU13, AU15, AU18, AU20, AU24, AU28 has been scored. If so, remove AU38 from the ***AU-list*** (see Table 5.5). Since no file consists of data that can be used to control the correctness of the hypothesis that a certain AU affecting the facial appearance of the nose and/or chin has been activated, no inter-file consistency check (step 3) can be performed. For each $\boldsymbol{AU_i} \in \boldsymbol{AU\text{-}list}$

and each anatomically possible combination of these, a hypothesis is placed that the AU or the AU combination in question is present. In total, there are 8 distinct hypotheses at most: AU26, AU27, AU38, AU39, AU26+AU38, AU26+AU39, AU27+AU38, AU27+AU39.

- The hypothesis verification (step 4) is performed for each hypothesis from step 3. If a hypothesis $\boldsymbol{h}$ is verified but not all hypotheses from step 3 have been checked, the hypothesis $\boldsymbol{h}$ and the frequency $\boldsymbol{Fe_{adjusted}}$ related to the expression $\boldsymbol{d(e_{adjusted})} = \boldsymbol{d(e_{input})} + \boldsymbol{h}$ will be stored temporarily. Once all hypotheses from step 3 have been checked, the hypothesis $\boldsymbol{h}$ stored together with the highest frequency $\boldsymbol{Fe_{adjusted}}$ is selected and $\boldsymbol{d(e_{adjusted})} = \boldsymbol{d(e_{input})} + \boldsymbol{h}$ is returned to the supervisor of the Facial Action Encoder.

Post-processing of the "mouth" file

If a dual-view facial image forms the input to the system and the "mouth" file selected by function F3 has been labelled as "missing", function F5 executes the procedure outlined above for a general case with the following adjustments to steps 3 and 4:

- Based on both the data constituting the "profile" file and the rules given in Table 5.7, check if any of $\boldsymbol{AU_i} \in$ ***AU-list*** can be scored and make the list ***AU-list1*** accordingly. Based on both the rules given in Table 5.12 and the result of the module *Vertical ANN Mouth Classifier* integrated into the Facial Data Extractor (section 4.3), check if any of $\boldsymbol{AU_i} \in$ ***AU-list*** can be scored and make the list ***AU-list2*** accordingly. Based on both the rules given in Table 5.12 and the result of the module *Horizontal Rule-based Mouth Classifier* (section 4.3), check if any of $\boldsymbol{AU_i} \in$ ***AU-list*** can be scored and make the list ***AU-list3*** accordingly. Make a new list ***AU-list*** composed of the common elements of lists ***AU-list1***, ***AU-list2*** and ***AU-list3***. For each $\boldsymbol{AU_i} \in$ ***AU-list*** and each anatomically possible combination of these, a hypothesis is placed.
- The hypothesis verification (step 4) is performed for each hypothesis from step 3. If a hypothesis $\boldsymbol{h}$ is verified but not all hypotheses from step 3 have been checked, the hypothesis $\boldsymbol{h}$ and the frequency $\boldsymbol{Fe_{adjusted}}$ related to the expression $\boldsymbol{d(e_{adjusted})} = \boldsymbol{d(e_{input})} + \boldsymbol{h}$ will be stored temporarily. Once all hypotheses from step 3 have been checked, the hypothesis $\boldsymbol{h}$ stored together with the highest frequency $\boldsymbol{Fe_{adjusted}}$ is selected and $\boldsymbol{d(e_{adjusted})} = \boldsymbol{d(e_{input})} + \boldsymbol{h}$ is returned to the supervisor of the Facial Action Encoder.

If a frontal-view facial image forms the input to the system and the "mouth" file selected by function F3 has been labelled as "missing", function F5 executes the general-case procedure with the following adjustments to steps 3 and 4:

- Based on the AU-coded description $\boldsymbol{d(e_{input})}$ of the currently examined expression $\boldsymbol{e_{input}}$ that forms the input to function F5, check if AU9 has been scored. If so, remove AU10 from the ***AU-list***. The underlying reasoning is based upon the knowledge contained in FACS (see Table 5.5). Similarly, check if any of AU26 and AU27 has been scored. If so, remove the other one as well as AU25 from the ***AU-list***. Based on both the rules given in Table 5.12 and the result of the module *Vertical ANN Mouth Classifier* (section 4.3), check if any of $\boldsymbol{AU_i} \in$ ***AU-list*** can be scored and make the list ***AU-list1*** accordingly. Based on both the rules given in Table 5.12 and the result of the module *Horizontal Rule-based Mouth Classifier* (section 4.3), check if any of $\boldsymbol{AU_i} \in$ ***AU-list*** can be scored and make the list ***AU-list2*** accordingly. Make a new list ***AU-list*** composed of the common elements of lists ***AU-list1*** and ***AU-list2***. For each $\boldsymbol{AU_i} \in$ ***AU-list*** and each anatomically possible combination of these, a hypothesis is placed.

- The hypothesis verification (step 4) is performed for each hypothesis from step 3. If a hypothesis $\boldsymbol{h}$ is verified but not all hypotheses from step 3 have been checked, the hypothesis $\boldsymbol{h}$ and the frequency $\boldsymbol{Fe_{adjusted}}$ related to the expression $\boldsymbol{d(e_{adjusted})} = \boldsymbol{d(e_{input})} + \boldsymbol{h}$ will be stored temporarily. Once all hypotheses from step 3 have been checked, the hypothesis $\boldsymbol{h}$ stored together with the highest frequency $\boldsymbol{Fe_{adjusted}}$ is selected and $\boldsymbol{d(e_{adjusted})} = \boldsymbol{d(e_{input})} + \boldsymbol{h}$ is returned to the supervisor of the Facial Action Encoder.

Table 5.12
Mapping between the results of modules *Vertical ANN Mouth Classifier* and *Horizontal Rule-based Mouth Classifier* and the individual AU codes that can be recognised by ISFER (Table 5.8)

Module's result	**Possible AU-coding**
Vertical ANN Mouth Classifier	
Smiling	AU12, AU13, AU16, AU18, AU19, AU23, AU25, AU26, AU28, AU28b, AU29, AU36b
Neutral	AU8, AU10, AU16, AU17, AU18, AU19, AU20, AU23, AU24, AU25, AU26, AU27, AU28, AU28b, AU28t, AU29, AU35, AU36b
Sad	AU10, AU15, AU16, AU17, AU19, AU23, AU25, AU26, AU28b, AU28t, AU29, AU36t
Horizontal Rule-based Mouth Classifier	
Stretched	AU10, AU12, AU13, AU15, AU16, AU17, AU19, AU20, AU23, AU25, AU26, AU27, AU28, AU28b, AU28t, AU29, AU36b, AU36t
Neutral	AU8, AU10, AU16, AU17, AU19, AU23, AU24, AU25, AU26, AU27, AU28, AU28b, AU28t, AU29, AU36b, AU36t
Puckered	AU10, AU16, AU17, AU18, AU19, AU24, AU25, AU26, AU27, AU28, AU28b, AU28t, AU29, AU35, AU36b, AU36t

Terminating the processing of the Facial Action Encoder

Prior to terminating its processing and forwarding the accomplished results to the Facial Expression Classifier (Figure 5.1), the supervisor of the Facial Action

Encoder calculates the intensities $I(AU_i)$ and the appropriate certainty factors CF_{AUi} and $CF_{I(AUi)}$ associated with each $AU_i \in d(e_{adjusted})$, where $d(e_{adjusted})$ represents the result of function F5. The following procedure has been applied:

1. $(\forall AU_i \in d(e_{adjusted}) \mid AU_i = AU_j \in d(e_{input}))$

   $$CF_{AUi} = CF_{AUj} \wedge I(AU_i) = I(AU_j) \wedge CF_{I(AUi)} = CF_{I(AUj)},$$

   where e_{input} formed the input to function F5 and CF_{AUj}, $I(AU_j)$, $CF_{I(AUj)}$ have been computed by function F4 (section 5.5).
2. $(\forall AU_i \in \mathrm{d}(\mathrm{e}_{adjusted}) \mid AU_i \neq AU_j \in d(e_{input}))$

   $$CF_{AUi} = 100 * P(AU_i + d(e_{input}) \mid d(e_{input})) \wedge$$
   $$I(AU_i) = avg(I(AU_j), \forall AU_j \in d(e_{input})) \wedge$$
   $$CF_{I(AUi)} = min(CF_{AUi}, min(CF_{I(AUj)}, \forall AU_j \in d(e_{input}))),$$

   where e_{input} formed the input to function F5, $P(AU_i + d(e_{input}) \mid d(e_{input})) = P(AU_i + d(e_{input})) / P(d(e_{input})) = typicality(d(e_{adjusted})) / (min(CF_{AUj}, \forall AU_j \in d(e_{input})) / 100)$ as already explained in section 3.2, and CF_{AUj}, $I(AU_j)$, $CF_{I(AUj)}$ have been computed by function F4 (section 5.5).

Finally, the supervisor of the Facial Action Encoder adjusts the DB of all encountered expressions as explained above and forwards the quantified AU-coded description of the currently examined facial expression as well as the certainties of that data to the last part of ISFER, namely the Facial Expression Classifier (Figure 2.25, Figure 5.1).

5.7 Discussion

Analysis of facial expressions in terms of rapid facial signals (i.e. in terms of the activity of the facial muscles causing the visible changes in facial expression) is an intriguing problem. Numerous methods exist for measuring the facial movements resulting from the action of the muscles manually (Ekman 1982b). Among those, the Facial Action Coding System (section 5.1, Ekman and Friesen 1978) is probably the most comprehensive and versatile system. In any case, it is the method most commonly used by researchers of facial behaviour (Hager 1985, Bartlett et al. 1999). Also, it is used by most researchers working on automating facial action coding from digitised images (chapter 2, Donato et al. 1999, Pantic and Rothkrantz 2000d).

While the automation of the entire process of facial action coding from digitised images would be enormously beneficial (section 5.1), we should recognise the likelihood that such a goal still lies in the relatively distant future. Yet, with the current technology it is potentially possible to automate much of the tedious and time-consuming FACS scoring parts and allow trained human observers valuable time for conducting the most difficult investigations of human facial behaviour (e.g.

on moods and intentions). The Facial Action Encoder part of ISFER, presented in this chapter, represents such an effort to automate parts of FACS scoring.

Based upon the data on spatially sampled prominent facial features coming from the Facial Data Extractor part of ISFER (chapter 4), and unlike any other system presented in the literature up to date (chapter 2, Tian et al. 2001), the Facial Action Encoder analyses facial expression fully automatically, robust and effectively in terms of:

1. an anatomically possible combination of 32 individual AU codes scored in a generic manner (i.e. independently of the physiognomy of the currently observed subject),
2. an intensity level assigned in a subject-adaptive manner to each encoded AU code,
3. a certainty measure assigned to these data based on the certainty measure assigned to the input data carrying usually ambiguous information about the currently examined facial expression captured in a static frontal-view or a dual-view facial image of the currently observed subject.

Thus, due to the Facial Action Encoder, ISFER outperforms any existing system for automatic facial expression recognition applicable to automated FACS coding in digitised facial images (see Table 2.7 and section 7.2). Yet, the processing of the Facial Action Encoder implies some drawbacks as well. First, the Facial Action Encoder performs the facial action coding in terms of 29 AUs (i.e. 32 AU codes, Table 5.8). Hence, it is not capable of encoding the full range of facial behaviour (i.e. all 44 AUs defined in FACS). Consequently, for two facial expressions that differ in terms of displayed AUs the system may generate the same AU-coded description. It is crucial, therefore, that the user is aware that the system automates merely a part of FACS scoring and that, depending on the kind of the input facial image, it can be used for detecting a limited number of specific facial actions listed in Table 5.5 and Table 5.8.

Further, the Facial Action Encoder is implemented in Java even though it is known that executing a Java-coded application is time consuming. Also, the Facial Action Encoder employs the subject-profiled database of all encountered facial expressions (Figure 5.1), which may become large since it keeps records of all expressions ever encountered while monitoring a particular subject for whom it has been defined. In principle this implies long retrieval times and, in turn, time-consuming execution of the system's code. However, since the clustered organisation of the DB of all encountered facial expressions supports efficient retrieval (section 5.6) and the execution of the Facial Action Encoder's code is incomparably faster than the execution of the code of the Facial Data Extractor part of the system, these issues do not form significant problems for the current version of ISFER. Although these problems might form serious shortcomings of ISFER once the facial feature detectors integrated into the Facial Data Extractor are replaced with a real-time detection techniques, or their current sequential processing is replaced by a parallel processing, the performance of the current version of the Facial Action Encoder is satisfactory (see also chapter 7) and it is likely that it will suffice the needs of the system as well as the needs of the potential users for some time.

As far as the subject-profiled DB of all encountered facial expressions is concerned, it might impose high storage requirements since its expansion is not controlled. Although this forms a shortcoming of the Facial Action Encoder part of ISFER, one should bear in mind that this problem becomes less and less significant as computer memory prices drop (e.g. Hassler 2001).

Another peculiarity of the Facial Action Encoder's design, which might be thought of as drawback, is the employment of thresholds (i.e. ***t1*** and ***t2*** in Table 5.5 and ***t3*** and ***t4*** in Table 5.7; see also ***var1-var4*** in Figure 5.12 and Appendix B). In general, thresholds induce a reduced flexibility of the performed reasoning, especially if used in an inference engine that should be capable of adapting to a particular individual (like the AU-quantification process of the Facial Action Encoder). Yet the thresholds employed by the reasoning mechanism of the Facial Action Encoder are not generically defined; for each novel subject, they are initialised by the relevant values stored in the related subject-profiled DB of extreme model deformations (Figure 5.12). Hence, they do not form an impediment to the intended adaptive reasoning of the Facial Action Encoder.

A more important issue concerns the fact that the Facial Action Encoder does not take into account the temporal aspect of facial expression analysis. It has been developed to perform a quantified facial action coding based on the data extracted from static facial images of the currently observed subject rather than from facial image sequences. Yet, a recently growing body of psychological research argues that timing of facial expressions is a critical factor in facial action coding. Namely, it is thought that information about the time course of a facial action has a psychological meaning relevant to the intensity and genuineness of the displayed facial action (Izard 1990, Davidson et al. 1993). From an engineering point of view, at least, associating a temporal aspect with the current spatial aspect of facial expression analysis will improve the reliability of the system's results. In that case, dealing with partial and highly inaccurate data resulting from the Facial Data Extractor could be based upon the knowledge on facial expression dynamics. The currently employed DB of all encountered facial expressions (that grows with each session performed with the pertinent subject and, therefore, can eventually slow down ISFER's processing) would not be necessary in that case. Namely, each facial action has its onset, apex, and offset, which can be recognised based on the changes in facial expression characteristic for these time markers of the facial action in question. Using the information on both the spatial and the temporal course of an encountered facial action, statistical predictions can be made on the current changes in facial expression related to the given facial action and its current time marker. In addition, the larger the number of the frames recorded per minute of the examined image sequence, the higher the certainty that the appearance of the monitored facial features remains the same.

Furthermore, including the temporal aspect of facial expression analysis will potentially facilitate encoding of a wider range of facial behaviour. By applying an optical flow computation method or some other method for estimating the motion in various facial areas (for examples see Table 2.6), brief muscle actions like blinking (AU45), winking (AU46) and wiping the lips (AU37) and the muscle actions involving conspicuous facial movements like moving the jaw sideways (AU30), producing a bulge by pushing the tongue against the cheeks (AU36l, AU36r), and possibly clenching the jaw (AU31) would be detectable in a frontal-view facial image sequences. If combined with a method for separating permanent slow facial signals like wrinkles and dimples from similar changes in facial expression caused by the activity of the muscles (e.g. by separating these features in terms of their temporal stability), a facial-motion tracking method could facilitate detection of the muscle actions like dimpling the mouth corners (AU14) and deepening the nasolabial furrow (AU11).

Moreover, if the automated expression analysis were based on a facial motion tracking method, this would potentially facilitate facial action coding in image sequences of any person, independently on his/her appearance. Artificial and natural facial signals like glasses, facial hair, birth marks and unibrow would not form untreatable sources of noise (Simoncelli 1993).

Finally, with the information on the spatial and temporal course of encountered facial actions, the problem of occlusion of the monitored facial features (e.g. by a hand), causing partial data, can be tackled. Statistical predictions about the current changes in facial appearance could be made based upon the knowledge about the facial actions encountered in the previous frame of the currently examined facial image sequence and their spatial and temporal course. This would enable the system to "fill in" missing parts of the observed face and to "perceive" a whole face even when a part of it is occluded.

In summary, by basing the intended automatic facial expression analysis upon a facial-motion tracking method and including both the spatial and temporal segmentation of the examined facial image sequences, could greatly enhance the current state of the art in automatic facial expression analysis and draw it rather close to the ideal model proposed in section 2.2. For future developers of ISFER, this means investigating a robust and reliable method for facial action encoding and quantification applicable to automated (and preferably complete) FACS-scoring, which is to be based upon automatic analysis of facial expression dynamics in dual-view facial image sequences.

6 Facial expression classification

The abilities to have, express and be aware of our true feelings, coupled with the abilities to handle feelings so that they are appropriate, to marshal the emotions in the service of a goal, to recognize emotions in others and to skilfully handle the affective arousal of others, are the abilities termed as "emotional intelligence".

(Goleman 1995)

The topic of automatic interpretation of human communicative behaviour, that is, giving machines the ability to detect, identify, and understand human interactive cues, has become a central topic in machine vision research, natural language processing research and in AI research in general. The catalyst behind this recent upsurge of interest in the research topic of *human-centred computing* is the fact that the automation of monitoring and interpretation of human communicative behaviour is essential for the design of future smart environments, perceptual user interfaces, and ubiquitous computing in general. As embedded computing devices become part of more and more aspects of our lives (office, home, car, and even clothes (Pentland 2000, Clarkson et al. 2000)), the next generation of computing and information technology requires more than engineering enhancement of the state of the art; it is a kind of an *emotional intelligence* context, that is, it is the translation and emulation of human behavioural cues what will determine the uses and usefulness of the computers of our future (Pantic and Rothkrantz 2001a).

Human face-to-face interaction consists of a complex interplay of thoughts, language and non-verbal communicative signals. If that is the intended model for future computing devices, as suggested by Thalmann et al. (1998), Pentland (2000), and Marsic et al. (2000), then the computers of the future must be equipped with techniques that enable them to sense and understand user's context, to construct theories of human mind, and to respond in an automatic and intelligent (i.e. context-dependent) way. In turn, the key technical goals of human-centred computing concern the determination of the context in which the user acts, that is, disclosing in an automatic way who the user is, where he is, what he is doing, and how he is feeling, so that the computer can act/respond accordingly. In addition, the socio-technical issues of determining how and when to interrupt the user, discovering the appropriate question to ask, and deciding in which way to respond (e.g., which words, facial expression and intonation to generate in response), have become great challenges in the design and development of next-generation computing.

In general, this thesis pertains to the specific facet of sensing the user's context; it is concerned with providing machines with the ability to detect and interpret how the user is feeling based on the sensed user's facial expression. In particular, this chapter pertains to the last problem of automating the recognition of human affective states, that is, to the problem of automating the interpretation of the displayed facial expression in terms of attitudinal states. The first section of this chapter summarises the potential benefits of automating affect-based interpretation of human facial expressions and, in turn, indicates a great diversity of application domains where benefits could accrue from an automatic system like ISFER. The psychological background of the production and interpretation of human facial affect is provided in the second section. The rest of this chapter is concerned with the third part of ISFER, that is, with the Facial Expression Classifier (see Figure 2.25). The Facial Expression Classifier is a memory-based expert system that performs case-based reasoning about the interpretation of the AU-coded description of the presently examined facial expression in terms of a quantified set of interpretation labels used by the current user. The theory of human autobiographical memory organisation (Schank 1982, 1984) and the instance-based machine learning methods (Mitchell 1997) inspired the organisation and the processing of the developed Facial Expression Classifier. The architecture of the Facial Expression Classifier is outlined in section 6.3. Section 6.4 discusses the employed dynamic memory of experiences (i.e. the utilised case base), which expounds the AU codes generated by the Facial Action Encoder (chapter 5) in terms of the interpretation labels learned from the user. The processing of the Facial Expression Classifier is explained in detail in section 6.5. For a detailed algorithmic representation of the processing of the Facial Expression Classifier, readers are referred to Appendix A. Finally, the advantages and the limitations of the proposed technique for automatic affect-based facial expression classification from static facial images are discussed in section 6.6.

6.1 Why automating affect-sensitive expression analysis?

Not all computing devices need to pay attention to users' affective states (think about a calculator or a scanner), or to have "emotional" abilities to translate the user's attitudinal states and emulate appropriate behavioural cues in response. Some machines are useful as rigid (i.e. non-adaptive) tools, and it is probably best to keep them that way. However, there are many situations, especially in ubiquitous computing, which is widely thought to be the coming of the next-generation computing and information technology (Pentland 2000), where man-machine interaction could be improved by the introduction of machines that can adapt to their users (think about computer-based advisors, virtual information desks, cars' on-board computers and navigation systems, pacemakers, etc.). The information about when the existing processing should be adapted, the importance of such an adaptation, which part of that processing/reasoning should be adapted, and how, involves information about the context in which the user acts, that is, what he works on (i.e. which part of the existing processing is invoked at the moment) and how he feels (e.g. confused, irritated, frustrated, interested, etc.). The focus of the recently initiated research area of *affective computing* (Picard 1997) lies on sensing, detecting and interpreting human affective states and devising appropriate means for handling this affective information in order to enhance interaction in man-machine interfaces.

In addition to practical concerns of ubiquitous computing such as interfaces and virtual environments, which will be perceived more natural (Nakatsu 1998), more efficacious and persuasive (Reeves and Nass 1996), and more trustworthy (Olson and Olson 2000, Cassell and Bickmore 2000) if given the ability to sense and respond appropriately to the user's affective feedback, the potential benefits of the automation of affect-sensitive monitoring of facial displays are varied and numerous. The preceding sections of this thesis have separately enumerated many benefits of the efforts to automate facial expression analysis in general. This section summarises these benefits and indicates additional research areas where benefits could accrue from an automated facial-affect-sensitive monitoring tool that were not emphasised previously.

A considerable amount of research in social psychology has shown that affective state recognition plays an important role in learning and attending to what is important (Salovey and Mayer 1990, Boyle et al. 1994). Children show signs of recognizing parents' affective expressions like approval and disapproval long before they comprehend the language. Lack of the ability to express and "read" the affective expressions is typical for central and sensory impairments like autism or schizophrenia (Sigman and Capps 1997, Steimer-Krause et al. 1990) and may provide evidence for the location and type of brain lesions (Hurwitz et al. 1985). Assessment of emotional abilities may serve not only as a marker for psychosomatic dysfunctions, but may also serve for discovering socio-behavioural disorders like delinquency (McCown et al. 1988) and as an aid in management of children who fail to elaborate or fail to heed normal social signals (Horton 1987). Hence, monitoring facial behaviour and assessing emotional abilities is important for a large number of studies in behavioural science (e.g. in the studies of emotion, cognition, and development of children), in anthropology (e.g. in the studies on cross-cultural perception and production of facial expressions), in psycho-physiology (e.g. in the studies on affect measured from physiological signals), in neurology (e.g. in the studies of dependence between brain lesions and emotional abilities impairments), and in psychiatry (e.g. in the studies on schizophrenia). A reliable, inexpensive, and rapid automated facial-affect-sensitive monitoring tool that would be widely accessible could greatly improve the research in these fields. It could raise the quality of the research in which reliability, sensitivity, and precision are currently significant problems. Because it would decrease the amount of time currently necessary for conducting research, it could create opportunities for conducting more studies of greater quality at lower costs.

In addition, automatic assessment of boredom, inattention, and stress will be highly valuable for preventing critical situations in hazardous working environments like aircraft cockpits, air traffic control towers, space flight operation chambers, nuclear power plant surveillance rooms, or simply in the vehicles like trucks, trains, and cars. An advantage of the machine facial-affect-sensitive monitoring is that human observers need not be present to perform privacy-invading monitoring; the automated tool could provide prompts for better performance based on the sensed user's affective state.

This enormous variety of commercial and basic science research areas that would reap substantial benefits from an automated facial-affect-sensitive monitoring tool is the catalyst behind the recent upsurge of interest in the research topic of automatic analysis and interpretation of human facial behaviour (for an extensive review of the existing methods for automatic facial expression analysis, the reader is referred to chapter 2).

6.2 The psychology of human facial affect

Since it would be extremely beneficial to all of the research fields enumerated above to automate the interpretation of sensed facial displays in terms of affective states, this research topic is rapidly becoming an area of intense interest in the machine vision research and AI research generally. In turn, the question of how human perception of affective states can be characterised best has become a critical issue for many researchers in affective computing. Ironically, the growing interest in affective computing comes at a time when the established perception of human facial affect is strongly being challenged in the basic research literature. Considerable recent methodological criticisms have questioned the validity of the large body of data that have been widely accepted and considered as ground truth for already a few decades.

The classic psychological research on perception of facial affect has been carried out since the beginning of 60s by psychologist Paul Ekman and colleagues (for extensive reviews of the ensued studies see (Ekman 1982, Ekman 1994, Keltner and

Ekman 2000)). A substantial body of evidence, gathered in almost four decades, primarily focuses on the human observers' interpretations of photographed facial expressions and identifies what facial configurations are associated with what *emotions*[1]. These classic psychological studies on the facial expression of emotion claim the validity of the following assertions:

1. *Existence of six basic emotions*: there are six distinct facial expressions of emotion (sadness, happiness, anger, disgust, fear and surprise; contempt was tentatively added just recently) that are universal (Ekman 1980).
2. *Universality of recognizing the six basic emotional expressions*: the six basic facial expressions represent the same emotions in every culture (Frijda 1986); human observers label the basic facial expressions in the same way, regardless of culture (Fridlund et al. 1987).
3. *Universality of expressing the six basic emotions*: there exists a universal set of emotion reaction modes including facial expressions (Mesquita and Frijda 1992); six basic emotions are expressed in much the same way in all cultures (Carlson and Hatfield 1992); the six basic facial expressions can be found in neonates and the blind as well as sighted adults, although the evidence on the blind and neonates is more limited than that for sighted adults (Ekman and Sejnowski 1993).
4. *Genetic foundation of the six basic emotional expressions*: the six basic facial expressions of emotion are genetically based or pre-programmed (Izard 1980) and emotion-specific activity in the autonomic nervous system appears to emerge when facial prototypes of emotion are produced on request, muscle by muscle (i.e. AU by AU; Ekman and Sejnowski 1993).

In the past few years, psychologist James Russell and colleagues (Russell 1994, Russell and Fernandez-Dols 1997) have strongly challenged the classic approach to the perception of facial affect, mostly on methodological grounds. Russell argues that emotion in general, and facial expressions of emotion in particular, can be best characterised in terms of a multi-dimensional affect space, rather than in terms of a small number of emotion categories such as the basic emotion categories proposed by Ekman and colleagues. More specifically, Russell claims that two dimensions – *valance* and *arousal* – are sufficient to characterise facial affect space (this is a reduction of the dimensionality of the facial affect space proposed by Scholsberg (1954), where the underlying dimensions were pleasant-unpleasant i.e. valance, attention-rejection, and the degree of activation i.e. arousal). Apart from this central question in the field of emotion, concerning whether emotions are better thought of

[1] The classic psychological studies on perception of facial affect favour the term "emotion" rather than terms like "affect", "attitude" or "communicative display", which are related less to the actual experiencing of some emotional excitement / inner feeling and more to any displayed facial communicative behaviour which may but does not have to portray an actually felt emotion (Fridlund 1991).

as distinct categories or as interrelated entities that differ along global dimensions such as valance and arousal, Russell's criticisms of classic studies address a number of additional related issues (Russell 1994).

First, Russell points out that a great deal of the data from a large number of classic studies was generated using a single corpus of rather unnatural stimuli (e.g. posed facial expressions rather than spontaneous expressions). Also, he criticises certain experimental design flaws in the previous research (e.g., failure to properly randomise stimuli, biasing procedures for practice trials, small number of trials) and argues against the common reliance on "within-subject" designs[2]. Russell's primary criticism, however, concerns the commonly used response format. Virtually all classic studies employed the forced-choice response format, in which human observers were presented with a rigid list of emotion labels and were asked to pick the one that best matches the facial expression on the stimulus image. The subjects were not given the option of saying "none of the above" or of choosing a non-emotional interpretation label. Russell's critique of the forced-choice response format is that (i) it forces the subject to choose merely one of the given emotion labels and is, therefore, insensitive to perceived differences in intensity of emotion, and (ii) if a multiple choice of the given responses or inventing a new interpretation label was allowed, the universality of facial expressions of emotion might not be demonstrated. As a result of his criticism and in order to improve the inadequate research methods relying on the forced-choice response format, Russell proposed two alternatives:

1. freely chosen interpretation labels, and
2. quantitative ratings of the freely chosen interpretation labels.

As could be expected, this open-ended approach using quantitative ratings and freely chosen interpretation labels resulted in the conclusion that each human observer uses a range of interpretation labels that are applicable to a given facial expression to different degrees (Russell, 1994). In addition to the considerable body of research in anthropology and social psychology indicating that the comprehension of a given emotion label and the ways of expressing the related affective state differ from culture to culture (Efron 1941, Matsumoto 1990, Shigeno 1998), Russell's criticisms questioned the classic consensus on the existence of six basic facial expressions of emotion. In other words, it is rather improbable that each of us will express a particular affective state by displaying the same facial expression as it is improbable that a particular facial expression will always be interpreted in the same way independently of who the observer is. Consequently, there is no psychological

[2] In a within-subject design each subject is asked to judge the entire set of stimulus posed facial images within a relatively short period of time. This invites a more direct comparison between various facial expressions than everyday encounters with facial expressions allow us. In other words, the subject might feel called on to notice the difference between two expressions and assign different labels to them.

scrutiny of universal facial expressions of affective states which can safely be adopted and employed in studies on automatic facial affect analysis and in affective computing in general. Hence, the definition of interpretation categories in which any displayed facial expression (i.e. any set of activated AUs) can be classified is a key challenge in the design of automated affect-sensitive face-monitoring tools like ISFER.

Several issues make this problem even more complex. Neither verbal (spoken words) nor non-verbal communicative signals (facial expressions, posture, body gestures, clamminess, respiration, intonation of the spoken words) reveal exact information about the affective state of the observed person; some aspects of innermost feelings may remain private, however, especially if the subject wishes them to be that way and sufficiently disguises them. Thus, even if a unified psychological scrutiny of universal facial affect expressions were to exist and facial communicative signals could always be acquired accurately (which is not feasible using the currently available sensors; chapter 4), this would still not be sufficient to determine the exact nature of the observed affective states.

The only way to reveal the (almost[3]) exact nature of facial affect expressed by a particular user is to employ a personalized facial-affect analyser trained by that user, that is, an analyser capable of adapting the facial expression classification mechanism according to the user-provided interpretations of those expressions. An additional advantage of an automated, user-profiled, facial-affect-sensitive monitoring tool is that doctrinal confrontations on emotions discussed above cannot have an effect on it.

6.3 Architecture of the Facial Expression Classifier

This section provides the reader with the details on the design and implementation of the Facial Expression Classifier part of ISFER. The section will begin by examining the problem domain of affect-based facial expression interpretation in digitised facial images. The design requirements arising from the taxonomy of the problem domain will then be summarised. A way of fulfilling these design requirements will be proposed and the types of knowledge necessary for the intended case-based reasoning approach will be discerned. Finally, a discussion concerned with the functional design and the actual implementation of the Facial Expression Classifier will be provided.

[3] As already explained above, the exact nature of someone's affective state may remain private if that person wishes it to be that way.

Problem domain

The domain of affect-based facial expression interpretation is probably the most complex area of facial image understanding. Like all other areas of machine perception of human faces (e.g. person identification or facial action encoding), it includes the problems concerned with sensing, detecting and modelling faces and facial features. However, the affect-based facial expression interpretation domain also includes the problem of *adaptability*, that is, the problem of adjusting the interpretation process in accordance with the situation in which the observed facial behaviour occurred (*context dependency*) and in accordance with the meaning that the current user associates with the displayed facial behaviour (*user dependency*). Let me explain these issues in more detail.

The problem of context dependency is closely related to the problem of *intentionality* and *reactivity*, that is, to the problem of uncovering the meaning of purposive, reactive and communicative facial behaviour. For instance, to interpret facial displays of video-conference participants requires making use of the idea that an observed person either *intends* to communicate some information to other participants or *reacts* to the present communication. A frown, for example, can help the speaker to emphasise the complexity of the discussed problem, or the listener to express disagreement with the communicated information or confusion about that information, etc. Thus, uncovering the exact meaning of a frown depends on the role of the expresser in the currently monitored situation. In other words, affect-based facial expression interpretation is context dependent. Yet acquiring information about the context in which a facial expression appears is rather difficult to accomplish in an automatic way and forms a separate research area in the field of machine vision (Pentland 2000). Although very important for the affect-based facial expression interpretation domain, the problem of context dependency has been handled neither by the existing automated systems for facial expression analysis (Pantic and Rothkrantz 2000d) nor by ISFER (the limitations of ISFER in general, and of the Facial Expression Classifier part of ISFER in particular, are summarised in section 6.6).

The problem of *user dependency* is related to the fact that facial displays are typically communicative acts whose interpretations are not cross-culturally universal (see section 6.2); what a certain facial expression means to me can be entirely different from what it means to somebody else. In other words, the interpretation of facial expressions can differ from person to person. Although much work has been done on automating facial-affect recognition in facial images and image sequences (chapter 2, Pantic and Rothkrantz 2000d), almost all of this work employs singular classification of input facial data into one of the six basic emotion categories as defined by Ekman and Friesen (1975). This approach has many drawbacks. The preceding section has separately enumerated most of the shortcomings related to the pertinent psychological research on perception of facial affect. Besides those, automated systems for facial affect recognition that classify the sensed facial expression information into one of the six basic emotion categories have a number of additional limitations that may be summarised as follows:

1. *Singular classification*: As noted by Ekman himself, pure facial expressions of six basic emotions are seldom elicited; most of the time people show blends of emotional facial displays. In Ekman's theory, the facial displays of the six basic emotions are considered to be the building blocks of facial expressions of more complex emotional states (Ekman and Friesen 1975, Ekman 1982). In turn, classification of sensed facial displays into a single basic emotion category is not realistic. An automated affect-sensitive analyser of facial expressions must accomplish, at least, a quantified facial expression classification into multiple basic emotion categories. Zhang et al. (1998) and Pantic and Rothkrantz (2000b) proposed such classifications of input facial data.
2. *Multiple classification*: As argued by Russell and colleagues, it is not at all certain that each and every facial expression which can be displayed by a human face, can be classified as a combination of the six basic emotion categories (Russell and Fernandez-Dols 1997). One can think, for instance, about the "bored" and "are you joking me?" attitudinal facial displays that, if classified into multiple basic emotion classes, would probably not retain their initial intention (Figure 6.1). In addition to the fact that a given emotion label may be comprehended differently by different persons, quantitative ratings on multiple basic emotion labels are not sufficient for a realistic interpretation of each and every displayed facial expression.

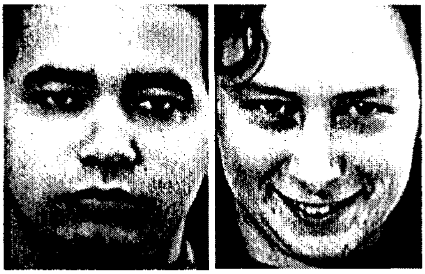


Figure 6.1: Facial displays of "bored" (left image) and "are you joking me?" (right image) attitudinal states. When classified in multiple basic emotion categories by the system of Pantic and Rothkrantz (2000b), the associated interpretations are: anger & sadness (left image) and anger & happiness (right image).

As explained in the preceding section, a way of dealing with these shortcomings of the currently existing automated affect-sensitive facial expression interpreters is to enable the intended automated interpreter to adjust the employed facial expression classification mechanism to the interpretations the current user associates with various facial displays. In other words, the solution to the problems listed above is machine learning: rather than having a priori classification rules and a priori defined interpretation categories, the rules can be potentially learned through interaction with the user and by learning the meanings he/she associates with different facial expressions.

Design requirements

In order to enhance the state of the art in automated affect-sensitive analysis of facial expressions, the design of the Facial Expression Classifier part of ISFER has been aimed at the realisation of an automated, adaptive (user-profiled), affect-sensitive, facial expression classifier such that doctrinal confrontations on emotions discussed above cannot have an effect on it. Thus, the Facial Expression Classifier part of ISFER has been envisioned as a learning facility of the system capable of performing a fully automated, robust classification of the input data generated by the Facial Action Encoder part of the system in terms of:

1. one or more *learned interpretation labels* freely defined by the current user, and
2. a *quantitative rating* on each of the scored interpretation labels.

From an engineering point of view, the intended automated classifier of input facial data should be efficient and effective. Consequently, the desirable properties of the Facial Expression Classifier have been defined as follows:

- The processing of the tool is easy to construct from the input data.
- The tool is easy to update. Similarly to the case of the Facial Action Encoder, in the case of the Facial Expression Classifier this is of crucial importance since the future users are psychologists and subjects having usually little or no technical background.
- The tool is easy to use. Once more, this is of particular importance since the potential users of ISFER are persons without a high level of technical knowledge. Here, one of the main aims is to develop a user-friendly interactive tool that does not require the user to be in control at all times.
- The tool itself is efficient to store.

Learning concept

According to the design requirements listed above, the Facial Expression Classifier part of ISFER should represent an interpreting facility of the system capable of learning from the user the appropriate interpretations of facial expressions. As explained in section 3.5, three issues crucial for the design of a learning system are: (i) to define the learning problem, (ii) to determine the appropriate target function(s) to be learned such that for any instance of a new problem as input it can produce a trace of its solution as output, and (iii) to choose the appropriate machine learning algorithm for the given learning problem and the defined target function(s).

As delimited by the definition given by Mitchell (1997), a computer program is said to learn from experience ***E*** with respect to some class of tasks ***T*** and performance measure ***P*** if its performance at tasks in ***T***, as measured by ***P***, improves with experience ***E***. Hence, to represent an adaptive interpreting facility of the system, the Facial Expression Classifier should improve its performance at the class of tasks involving *classifying facial expressions*, *as measured by its ability to accomplish user-defined interpretations*, with experience *obtained through* *interaction with the user about the meaning that he/she associates with different facial expressions.*

Once the learning problem is defined by task ***T***, performance measure ***P***, and training experience ***E***, the next choice in the design of a learning system is to determine exactly what type of knowledge will be learned and how this will be used by the performance program. In the case of the Facial Expression Classifier part of ISFER, the target function to be learned should choose the best matching user-defined interpretation label(s) for *any* input facial expression described in terms of quantified AU codes. Hence, if an input facial expression has not been previously encountered, the Facial Expression Classifier should choose one or more best matching user-defined interpretation labels that have been associated with facial expressions similar to the input facial expression. In other words, each time a new query instance (i.e. the AU-coded description of the input facial expression) is presented to the system, its relationship to the previously encountered examples (various facial expressions and the associated interpretation labels) should be examined by the system so that it can then assign a target function value for the new instance. In addition, the user should be able to define a novel interpretation label for an examined expression at any time. This implies, in fact, that the system should be capable of on-line learning, the main reason being that there are too many various facial expressions, that is, that there are too many different possible combinations of AU codes for the system to learn through training. In fact, it would be extremely difficult (if possible at all) and highly time consuming to perform such a training. Also, the user should be given the freedom to "change his mind" and redefine the interpretation associated with a certain facial expression. By this, the Facial Expression Classifier will resemble humans more closely in the way in which they become experts, namely through trial and error (i.e. experience). As defined by Aha et al. (1991) and Mitchell (1997), a technique suitable for both on-line learning and approximating the target function *after* the query instance has been observed, is instance-based learning (i.e. lazy learning as opposed to eager learning, see section 3.5).

The Facial Expression Classifier part of ISFER has therefore been designed and developed as a memory-based expert system inspired by both Schank's theory of human autobiographical memory organisation (Schank 1982) and case-based reasoning. From an engineering point of view, the memory-based design of the Facial Expression Classifier meets the design requirement outlined above in the following way:

- *The processing of the tool is easy to construct from the input data.* In the case of the Facial Expression Classifier, the input data is a quantified AU-coded description of the currently examined facial expression generated by the Facial Action Encoder part of the system. Optionally, the input data may be also the information provided by the user about a novel interpretation of the currently examined facial expression. In order to keep the construction of the processing of the intended tool simple, the Facial Expression Classifier has been developed as an integration of a *dynamic memory of experiences* and various simple processes (see Figure 6.2) that converts the quantified AU-coded description of the input facial expression into a quantified set of interpretation labels learned from the current user. The memory of experiences (i.e. case base) is dynamic since it changes and augments by each novel case that is presented (see also the next point).

- *The tool is easy to update.* To keep the update simple, the Facial Expression Classifier enables an automatic update of the dynamic memory of experiences where AU-coded descriptions of encountered facial expressions and related user-defined interpretation labels are stored (see section 6.4). Each time the user is not satisfied with interpretations provided by the system and associates a novel interpretation label with the examined facial expression, the Facial Expression Classifier automatically adapts the memory of experiences accordingly (section 6.5).
- *The tool is easy to use.* The Facial Expression Classifier does not require the user to be in control at all times, merely when he/she is not satisfied with the interpretation provided by the system for the currently examined facial expression. To enable a facile interaction between the user and the system, the aim was to provide the system with a visual user-friendly GUI.
- *The tool itself is efficient to store.* In order to achieve this, the aim was to split the code in functions efficiently (see Figure 6.2). Yet the main drawback of CBR systems is not the extensiveness of the code but the size of the used case base. Employing a large case base involves high memory/storage requirements and implies long retrieval times. Although there is no limit imposed on the size of the memory of experiences utilised by the Facial Expression Classifier, its expansion is controlled as proposed by Surma and Tyburcy (1998). Namely, if the user changes his/her mind about the interpretation that should be assigned to a certain (previously encountered) facial expression, the old (incorrect) case will be removed from the dynamic memory of experiences and the novel case will be added. Furthermore, the problems posed to the Facial Expression Classifier are compound (as opposed to monolithic) in the sense that individual parts of an encountered problem (i.e. individual AU codes) can be processed separately or in combination. In other words, a problem can be solved by reusing a single case or by reusing several cases constituting the dynamic memory of experiences. This allows the expansion of the case base to be control in the sense that only some of the encountered problems, or merely some parts of the encountered problems, have to be stored as novel cases in the dynamic memory of experiences (see sections 6.4 and 6.5 for a further discussion). In this way efficient case storage and retrieval is maintained.

Types of knowledge

Case-based reasoning (CBR) systems make use of many types of knowledge concerning the problem domain they are intended for. Richter (1995) identifies four different knowledge containers present in CBR: the vocabulary, similarity measures, adaptation knowledge, and the cases themselves. The first three containers usually hold the general knowledge about the problem domain. If there are any exceptions from this knowledge, they are typically handled by appropriate cases. According to this glossary, the Facial Expression Classifier part of ISFER comprises the following types of knowledge:

- *Vocabulary* includes, in general, the knowledge needed to choose the features used to describe the cases. The case features should be discriminative enough so that they can be used to select the cases similar to the input case and to prevent selection of too different cases that could lead to false solutions. In general, a case consists of the problem description and of the solution to that problem. Since the Facial Expression Classifier should interpret input AU codes in terms of user-defined interpretation labels, the vocabulary knowledge includes 32 different AU codes that the Facial Action Encoder is able to encode from an input facial image (see Table 5.8) and the user-defined interpretation labels. Each case is further defined as a feature vector that contains a unique set of AU codes describing the relevant facial expression and a user-defined interpretation label associated with that expression. Since the power of a CBR system lies in its ability to learn, that is, to increase its knowledge by collecting novel solved cases, the system should be able to extend the (initial) vocabulary fairly easy. The dynamic memory model which the Facial Expression Classifier utilises to organise the employed case base enables easy extension of the existing vocabulary (see section 6.4 and 6.5).
- *Similarity measures* include the knowledge that is used to choose both the case base organisation and the method of case retrieval. There are several methods for case base organisation (see section 3.6) and, in general, the knowledge about the cases actually employed can be used to choose the most appropriate one, that is, the organisation that will facilitate accurate and efficient case retrieval. Since different facial expressions can be associated with the same interpretation label, the cases constituting the case base of the Facial Expression Classifier have been organised in clusters: each cluster expounds a specific interpretation category and contains all cases that have been associated with the pertinent interpretation label. In other words, specific cases sharing the similar property (i.e. interpretation) are organised under a more general structure (i.e. a *Memory Organization Packet* (Schank 1982) forming the basic unit of a dynamic memory). In order to accomplish easy and accurate case retrieval, each cluster of the dynamic memory of experiences has been associated with certain indexes delimiting the AU codes characteristic for that cluster and within each cluster the cases have been hierarchically ordered according to the occurrence typicality of

the related set of AU codes. For a detailed discussion on these issues, the reader is referred to sections 6.4 and 6.5.

- *Adaptation knowledge* is probably the most obvious type of knowledge in a CBR system. It involves the knowledge about how differences in the problem affect the solutions. The case base can be adapted, in general, either automatically (by the system) or manually (by the user). In both cases, it is usually done by a set of rules. The case base of the Facial Expression Classifier is adapted by using a small set of hard-coded rules (i.e. choosing from a number of adaptation procedures, see section 6.5) which are directly applied to the input AU codes, interpretation by the system and user's feedback. The adaptation knowledge also includes the knowledge about how the correctness of a new solution can be evaluated. In the case of the Facial Expression Classifier, the correctness of a solution is evaluated in a straightforward manner. Namely, if the user does not provide feedback after the system has displayed a solution, the solution is considered to be correct. Conversely, if the user is not satisfied with the interpretation by the system and provides feedback, the system ensures that the entered solution does not violate the consistency of the case base and adjusts the dynamic memory of experiences accordingly. Once more, for a detailed discussion about the adaptation procedure, the reader is referred to section 6.5.
- *Cases* include knowledge about solved problem instances. The cases stored in the dynamic memory of experiences of the Facial Expression Classifier represent the knowledge that the system acquires over time. The organisation of the dynamic memory of experiences, the initial furnishing of it, and the representation of the specific cases is explained in detail in section 6.4. Manipulation, alteration and augmentation of the dynamic memory of experiences are all explained in section 6.5.

Functional design

To interpret an input facial expression successfully and to learn novel interpretation labels provided by the user automatically, the Facial Expression Classifier part of ISFER implements four functions (see Figure 6.2, section 6.5, and Appendix A) defined in accordance with the design requirements listed above:

1. *Function F1* (*retrieval*) classifies the input AU codes generated by the Facial Action Encoder part of the system according to the cases constituting the dynamic memory of experiences.
2. *Function F2* calculates a quantitative rating for each scored interpretation label.
3. *Function F3* evaluates the certainty of the resulting conclusions about interpretation of the currently examined facial expression.
4. *Function F4* (*adaptation*) adjusts the content of the dynamic memory of experiences when the user introduces a novel interpretation label for an examined facial expression (or a part of it).

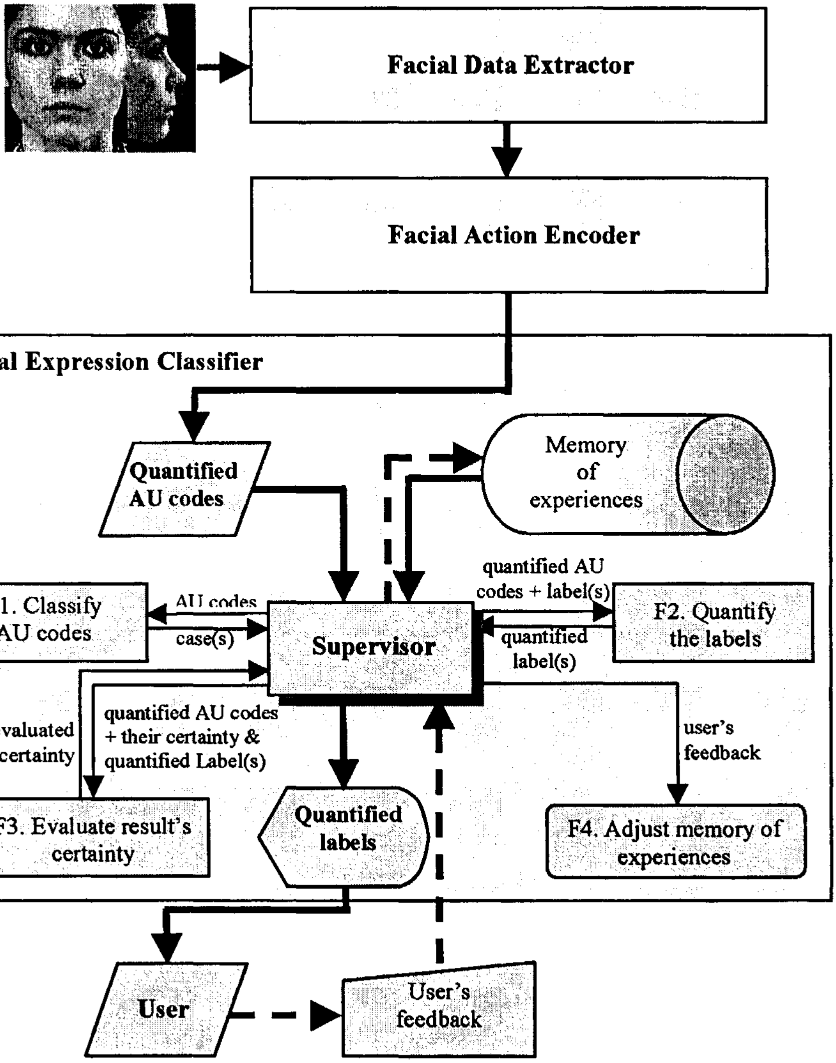


Figure 6.2: Architecture of the Facial Expression Classifier part of ISFER

As mentioned in section 3.7, the Facial Expression Classifier can be viewed as a problem-solving autonomous agent forming a part of a functionally distributed system. It is an autonomous agent since it does not require the user to be in control at all times. Namely, as soon as the initial furnishing of the dynamic memory of experiences is accomplished and as long as the user does not provide a novel interpretation label for the currently examined facial expression (i.e. as long as the Facial Expression Classifier is in *interpret mode*), the processing of the system is fully independent of the user's feedback. If the user is not satisfied with the interpretation provided by the system for the currently examined expression, the user may provide a novel interpretation label for it (i.e. the user may trigger the *learn mode* of the Facial Expression Classifier). The Facial Expression Classifier accepts the feedback provided by the user and adjusts the dynamic memory of experiences autonomously, that is, it does not require the user to be in control at all times while the dynamic memory of experiences is adapted. It is a problem-solving agent which resembles conventional CBR systems, since it uses domain-specific knowledge stored in the dynamic memory of experiences to achieve the intended functionality: facial expression interpretation in terms of interpretation labels defined by the user. The kernel of the Facial Expression Classifier, illustrated in Figure 6.2 as the *supervisor*, carries out the following agencies:

- It interprets the incoming data in terms of multiple quantified interpretation labels defined by the user.
- It adjusts the dynamic memory of experiences each time a novel case is presented.
- It creates action plans depending on the breadth of the set of cases constituting the dynamic memory of experiences.
- It carries out those plans by executing the appropriate procedures on the relevant data.

As noted above, the processing of the Facial Expression Classifier can be invoked in either an interpret mode or a learn mode. As long as the user does not provide a novel interpretation label for the currently examined facial expression, the Facial Expression Classifier runs in the interpret mode. By providing a novel interpretation label for the currently examined facial expression, the user triggers the learn mode of the system. As soon as the Facial Expression Classifier adjusts the dynamic memory of experiences to reflect the encountered novel case, it switches (automatically) back to the interpret mode. Let us examine how the four functions of the Facial Expression Classifier (Figure 6.2) support its problem-solving behaviour in each of the two execution modes.

In the interpret mode, the supervisor of the Facial Expression Classifier receives the input data – a quantified AU-coded description of the input expression and the certainty of that data – from the Facial Action Encoder part of the system. The first predefined goal that the supervisor will try to accomplish is to classify the incoming AU codes into the interpretation categories defined in the dynamic memory of experiences. To this end, the supervisor selects from the action space appropriate predefined plans. If the incoming combination of AU codes matches exactly a certain case stored in the dynamic memory of experiences, the supervisor retrieves the associated interpretation label to be used in the further processing. However, if no specific case stored in the memory of experiences matches the input combination of AU codes exactly, *function F1* will be activated to decompose the input combination of AUs into its components, each of which matches exactly a specific case stored in the dynamic memory of experiences. The second goal that the supervisor will try to achieve is to calculate a quantitative rating for each of the scored interpretation labels. To this end, the supervisor retrieves appropriate information about the relevant interpretation categories from the dynamic memory of experiences and then activates *function F2* to compute the quantitative rating for each scored interpretation label. Finally, the supervisor activates *function F3* to calculate the certainty of the obtained conclusions and displays these results to the user.

As noted above, the learn mode of the Facial Expression Classifier is triggered each time a novel case is encountered. In that case, the supervisor of the Facial Expression Classifier receives the input from the user – a novel interpretation label for the currently examined facial expression – and tries to reconstruct the dynamic memory of experiences to reflect the pertinent encountered novel event. To this end, the supervisor selects from the action space appropriate predefined plans. If no event stored in the memory of experiences matches exactly the AU-coded description of the currently examined facial expression for which a novel interpretation label has been introduced, the supervisor activates *function F4* to augment the dynamic memory of experiences with the pertinent additional case. Conversely, if the AU-coded description of the expression for which a novel interpretation label has been introduced matches a specific event stored in the dynamic memory of experiences exactly, the supervisor changes the content of the case base by removing the matching event and then activates function F4 to augment the dynamic memory of experiences with the novel user-defined event.

So, like the Facial Action Encoder part of ISFER, the Facial Expression Classifier part of ISFER might be viewed as a problem-solving autonomous agent. However, note that this is just one way of viewing the architecture of the third part of ISFER. As explained by Moulin and Chaib-Draa (1996), any expert system can be seen as an agent, at least as a reactive agent (like the Facial Action Encoder and the Facial Expression Classifier) if not as an intentional or social agent. The reason for discussing the Facial Expression Classifier part of ISFER as an agent is that its current functionality can easily be enhanced to provide an alert if some attitudinal state is encountered (e.g. boredom, frustration, stress, etc.). In that case, the Facial Expression Classifier would represent a *consumer-based problem-solving agent* (Hendler 1999) capable of manipulating the incoming information on behalf of the user. Allowing the user to define his/her current interest (i.e. a particular information source like a certain participant in a video conference or a certain patient in a group therapy as well as a particular attitudinal state of interest) would increase the commercial potential of ISFER. In that case, ISFER would embody an application-independent automatic tool for facial expression analysis (see also section 3.7).

Currently, nevertheless, the Facial Expression Classifier is not able to manipulate its goals and create new ones according to the wishes of the user; it achieves a predefined set of goals by selecting from the action space appropriate predefined plans. In other words, it is a reactive agent as any other expert system. Hence, for the sake of clarity and precision, in the remainder of the text the Facial Expression Classifier is discussed as an adaptive memory-based expert system that performs case-based reasoning with uncertainty on facial expression interpretation based on the input facial expression data, generated by the Facial Action Encoder part of ISFER, and the interpretations learned from the current user.

Implementation

As noted above, the aim was to design the Facial Expression Classifier part of ISFER such that it represents efficient and effective learning facility of the system. In brief, the Facial Expression Classifier should be designed such that it is easy to construct and to integrate into the ISFER, efficient to store, and easy to use and to update by the potential users of the system while imposing no constraints on the operating system of the utilised work-station. Hence, the aim was to design an efficient, portable, interactive, adaptive, user-friendly tool which classifies automatically facial expressions into the interpretation categories freely defined by the user.

Because both the Facial Data Extractor and the Facial Action Encoder part of ISFER have been implemented in Java and because of the availability of the JDK's Abstract Window Toolkit, which is a platform-independent GUI tool builder, Java was perfectly suitable for the development of the Facial Expression Classifier. Like by the Facial Data Extractor and the Facial Action Encoder, one might argue that the time-consuming execution of the Java code forms a serious drawback of the system. Yet this is of little concern since the time spent by the processor on executing the code of the Facial Expression Classifier is rather short compared to the time spent on executing the code of facial features detectors of the Facial Data Extractor (chapter 4). A more important issue is that CBR systems using large case bases impose high memory/storage requirements and long retrieval times (although the order of both is at most linear with the number of cases). Nevertheless, since the expansion of the utilised dynamic memory of experiences is controlled (see the third sub-section of this section), long retrieval times are of little concern in the case of ISFER.

6.4 The dynamic memory of experiences

As noted above, the case base organisation and the retrieval algorithm are interrelated. The organisation of the case base should be such that it enables accurate and efficient case retrieval. Accurate retrieval means that the best matching case will be retrieved. Efficient retrieval means that cases will be retrieved fast enough yielding acceptable system response times. These two factors are inversely proportional: it is easy to guarantee accurate retrieval at the expense of efficiency (e.g. by matching all the cases) and easy to offer fast retrieval if only a fraction of the employed case base is searched (possibly missing some examples). Hence, the accomplishment of both a good case base organisation and a good retrieval algorithm means to accomplish the best compromise between accuracy and efficiency of the retrieval algorithm. While this section explains in detail the actually utilised cases, the organisation of the employed case base and its initial furnishing, the next section discusses the employed retrieval algorithm (i.e. function F1, Figure 6.2) and the overall processing of the Facial Expression Classifier part of ISFER.

Cases

Cases are the basis of any CBR system. A case is a piece of knowledge representing an experience. Typically, a case comprises the problem description and the solution to that problem. Cases can be represented in a variety of forms using any of the existing AI representational formalisms including frames, objects, semantic nets and rules. There is a lack of consensus within the CBR community as to exactly what information should be stored for each case (Watson and Marir 1994). However, two pragmatic measures can be taken into account in deciding in which way cases should be represented: the intended CBR system's functionality and the ease of acquisition of the information represented in cases (Kolodner 1993).

The Facial Expression Classifier part of ISFER should classify the input AU codes, generated by the Facial Action Encoder part of the system, into the interpretation categories freely defined by the user. Hence, the best and the simplest way to represent a case utilised by the Facial Expression Classifier is to use a feature vector composed of the input AU codes describing the observed facial expression and the interpretation label associated with that expression. A sample case can be represented in a reader-oriented pseudo code as given in *(1)*.

```
("I don't know", AU1+AU2+AU15)            (1)
```

Organisation

As noted above, the utilised case base should be organised in a manageable structure that supports efficient case search and retrieval. A system that uses just cases and no other explicit knowledge (e.g. the relations between cases, that is, their "chunkiness") is in fact a nearest-neighbour classifier or a database retrieval system that does not exploit the full generalisation power of CBR systems. Such a flat organisation of the case base is usually inefficient since for the retrieval the whole case base has to be searched and every case matched. Thus, a balance has to be found between case memory organisations that expound and preserve the semantic richness of cases and methods that simplify the access and retrieval of relevant cases. As already noted in the preceding section, the knowledge about the actually exploited cases can be used to choose the most appropriate case base organisation.

The fact that different facial expressions might be interpreted using the same interpretation label was the reason to choose the clustered organisation as most appropriate for modelling the case base of the Facial Expression Classifier. Hence, the utilised dynamic memory of experiences comprises cases that are grouped in clusters based on their mutual similarity with respect to the interpretation assigned to them. Each cluster represents a specific interpretation category and contains all cases that have been associated with the pertinent interpretation label. For instance, the cluster forming the interpretation class "*I don't know*", used in the example *(1)* given above, can be represented in a reader-oriented pseudo code as given in *(2)*.

```
("I don't know", AU1+AU2+AU15, …, AU1+AU2+AU5+AU15+AU17+AU26)          (2)
```

In order to enable easy and accurate case retrieval, each cluster of the dynamic memory of experiences has been associated with certain indexes delimiting the AU codes characteristic for that cluster and the cases of each cluster have been hierarchically ordered according to their typicality. Let me explain these issues in more detail.

As noted in section 3.6, the term "indexing" is a term for denoting the accessibility problem (Kolodner 1996) – indexes in fact define the scheme for retrieval of cases from the case base. In order to facilitate a retrieval that will return cases most useful for solving the posed problem, the utilised indexes should be predictive of the case relevance, purposeful in the sense that it should be obvious why they are used, and discriminative but also abstract enough in the sense that they should not be uniquely applicable to just one problem situation (i.e. case). To devise the appropriate indexes with all those characteristics, the indexing vocabulary of the Facial Expression Classifier was defined such that it is synonymous to the parameters describing the problem (i.e. AU codes) and drawn from the concepts that characterise the intended reasoning task. In other words, the employed indexing method has been based upon the following:

- The main task of the Facial Expression Classifier is to interpret the input expression in terms of the user-defined interpretation labels, that is, to classify the input AU codes into the interpretation categories learned from the user.
- If each facial action (i.e. AU code) could be classified into any interpretation class, this would mean that only a single (cross-culturally universal) interpretation category would exist. However, this is not the case: as shown by psychologist James Russell (Russell 1994), human observers associate various interpretation labels (more than six basic emotion labels) with various facial expressions. In turn, each user-defined interpretation class could be characterised by some facial actions (i.e. facial expressions) that are unique to that interpretation category.

Hence, within the dynamic memory of experiences, the indexes associated with a cluster representing a certain interpretation category comprise the AU codes (or a combination of them) being characteristic for the pertinent interpretation category.

Once the retrieval method decides the interpretation class(s) useful for solving the encountered problem by parsing the input AU codes with the indexes of various clusters, it should choose the best matching case within this pre-selected cluster. In general, a method to achieve this is to search the whole pre-selected cluster and to match every case. Yet this is inefficient. To achieve efficient case retrieval, the cases most probable to represent the solution for the encountered problem should be matched first. In the case of the Facial Expression Classifier, efficient case retrieval has been achieved by introducing a hierarchical organisation of the cases within the clusters of the dynamic memory of experiences. The cases of each cluster are hierarchically ordered according to their typicality: the larger the number of times a certain case occurs, the higher its hierarchical position within the given cluster.

In summary, each cluster of the dynamic memory of experiences employed by the Facial Expression Classifier of ISFER, is associated with a certain index(es) and each case within a cluster is associated with a number representing the typicality of that case. For instance, the cluster forming the interpretation class "*I don't know*" (see examples *(1)* and *(2))* and the cluster forming an interpretation class "*surprise*" can be represented in a reader-oriented pseudo code as given in *(3)*.

```
(index(AU1+AU2+AU15), label("I don't know"),
 cases((AU1+AU2+AU15, 8), …, (AU1+AU2+AU5+AU15+AU17+AU26, 3)))
                                                                    (3)
(index(AU1+AU2, AU5, AU27), label("surprise"),
 cases((AU5, 7), (AU1+AU2, 5), …, (AU27, 1)))
```

Initial furnishing

In principle, the Facial Expression Classifier part of ISFER should capture and emulate the user's expertise in interpreting facial expressions. As with an expert system, the first step towards enabling a CBR system to accomplish the intended expert-like reasoning is to acquire and formalise the knowledge of the expert. Unlike an expert system, whose core is formed by rules, the core of a CBR system is formed by cases. Therefore, in the case of CBR systems, much of the classic knowledge engineering typical for rule-based expert systems is replaced by *case engineering* (Aha 1998), the main aim of which is to obtain the cases that will constitute the case base. In general, CBR systems are more reliable and easier to build and maintain when the number of cases is large since the adaptation rules can be simpler and evaluation of the adapted case base can be facile and less frequent. Another approach is to use a smaller number of carefully selected "golden" cases having a wide problem coverage and extensive adaptation knowledge. In both cases, the case engineer has to have records of previously solved problems in order to obtain the cases that will constitute the case base.

As explained in the preceding section, the aim in the development of the Facial Expression Classifier part of ISFER was to realise an automated, adaptive (user-profiled) interpreter of facial expressions. If a user-profiled interpretation of facial expressions is to be achieved, records of previously solved problems which would define cases to be stored in the dynamic memory of experiences to be utilised by the Facial Expression Classifier should be available for each novel user of the system. Of course, such records do not exist. Yet, even though records of problem solutions are not available, data about the problems themselves are. That is, since within the Facial Expression Classifier, the *problem* is the AU-coded description of the currently examined expression, while the *solution* is the user-defined interpretation of that expression, and since any displayed facial expression forming the input to the ISFER will be coded by the Facial Action Encoder part of the system in terms of 32 AU codes, data about the problems that should be solved are available. This is a classical case of the incomplete records problem (Mark et al. 1996). Two approaches can solve this problem of incomplete records:

1. *On-line generation from scratch*: the available data on the problems to be solved are neglected and the case base is generated from scratch through interaction with the user.
2. *Off-line initial furnishing followed by on-line adaptation*: the knowledge is acquired – start with an initial training of the system (initial furnishing of the case base) based upon the available data on the problems and then potentially enlarge/enhance the case base by interacting with and learning from the user (on-line).

As explained above, complex adaptation procedures make CBR systems more difficult to build and maintain and may also significantly reduce user's confidence in the system if faulty adaptations are encountered due to incompleteness of the adaptation knowledge, which is the most difficult kind of knowledge to acquire (Mark et al. 1996). Therefore, an iterative case-base generation starting with an off-line initial furnishing of the dynamic memory of experiences has been favoured in the design and development of the Facial Expression Classifier.

The main aim of case-base initial furnishing in general, and of the dynamic memory of experiences in particular, is to provide problem domain coverage that is as wide as possible. Since the Facial Action Encoder part of ISFER encodes each input facial expression in terms of 32 AU codes, providing the user with 32 pertinent stimulus images (i.e. each of which represents the facial expression produced by an individual facial action) and asking him/her to associate an interpretation label with each of the given images will generate cases covering the whole problem domain. Hereafter, AU-coded description of any observed facial expression forming the input to the Facial Expression Classifier could be processed as a set of singular AU codes and interpreted in terms of the interpretation labels associated with those singular AU codes. Nevertheless, people seldom produce facial expressions by activating a single AU; facial expressions are usually the result of several facial actions (Ekman and Friesen 1978). Furthermore, as suggested by Ortony and Turner (1990), some components of facial expressions (e.g. squared mouth) might be hardwired to emotional or non-emotional attitudinal states[4]. In other words, there are certain combinations of facial actions that are typically displayed by people to express attitudinal states though the meaning associated to those parts of facial expressions might differ from person to person. The theory of Ortony and Turner as well as the set of 32 AU codes that the Facial Action Encoder is actually able to encode have both influenced the choice of facial expressions (Table 6.1) recorded as the stimulus images stored in the *datùbase of training images* and used for initial off-line training of the system, that is, for initial furnishing of the dynamic memory of experiences.

The database of training images

The database of training (stimulus) images (Figures 2.25 and 6.4) has been created with the help of eight certified FACS coders (three males and five females of five different European and South American nationalities, ranged in age from 22 to 43; none of the subjects had a moustache, a beard or wear glasses.). The subjects were asked to display certain facial expressions, that is, particular combinations of facial actions given in Table 6.1. The 720×576 pixels frontal-view facial images have been acquired and then clipped to contain just the subject's face. Then each subject has been asked to assign an index of impression on the scale from 1 to 10 to each of 280 stimulus images displayed by the other seven subjects, reflecting his/her opinion about the correctness and distinctiveness/ clarity of the judged facial expression when compared to the expressionless face of the pertinent subject. The displays of the 40 facial expressions listed in Table 6.1, having the highest average index of impression, have been selected to constitute the database of training images. The chosen images are of 6 different subjects (see Figure 6.3).

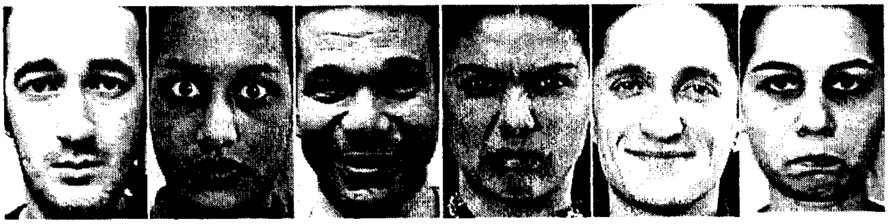


Figure 6.3: Sample stimulus images constituting the database of training images. AU-coded description of the images (from left to right): AU1, AU5, AU6+AU12(+AU26), AU9+AU17(+AU26), AU6+AU13, AU15

[4] Although Ortony and Turner (1990) indicate that some parts of facial expressions might be basic and hardwired to emotional states (e.g. a frown to an expression of anger), they emphasise that this does not entail that the emotions of which they are a (partial) expression are *basic* emotions as claimed by Ekman and colleagues (Keltner and Ekman 2000).

Table 6.1
The set of 40 facial expressions; displays of these constitute the database of training images

Expression	Description	Expression	Description
AU1	Raised inner eyebrow	AU6+AU13	From "happiness"
AU2	Raised outer eyebrow	AU15	Depressed mouth corners
AU1+AU2	From "surprise"	AU15+AU17	From "sadness"
AU4	Furrowed eyebrows	AU16+AU25	From "anger"
AU5	Raised upper eyelid(s)	AU17	Raised chin
AU7	Raised lower eyelid(s)	AU18	Puckered lips
AU1+AU4+AU5+AU7	From "fear"	AU19+AU26	Showed tongue
AU1+AU4+AU5	From "fear"	AU20	Horiz. stretched mouth
AU1+AU4+AU7	From "sadness"	AU23	Tightened lips
AU1+AU5+AU7	From "fear"	AU24	Pressed lips
AU1+AU4	From "sadness"	AU24+AU17	From "anger"
AU1+AU5	From "fear"	AU27	Vert. stretched mouth
AU1+AU7	From "sadness"	AU28+AU26	Sucked lips
AU5+AU7	From "fear"	AU28t+AU26	Sucked upper lip
AU8	Lips towards each other	AU28b+AU26	Sucked lower lip
AU9	Wrinkled nose	AU29	Jaw forward
AU9+AU17	From "disgust"	AU35+AU26	Sucked cheeks
AU10	Raised upper lip	AU36t+AU26	Tongue under upper lip
AU10+AU17	From "disgust"	AU36b+AU26	Tongue under lower lip
AU6+AU12	From "happiness"	AU41	Lowered upper eyelid(s)

Initial training of the system

To accomplish the initialisation of the dynamic memory of experiences, 40 facial images are retrieved one by one from the database of training images and shown to the user. With each of the retrieved images, an image of the neutral facial expression of the pertinent subject is also shown to the user for the comparison. The user is asked to assign an interpretation label to each of the facial expressions displayed on the stimulus images. To associate the stimulus facial expressions with the interpretation labels defined by the user, a temporary database is exploited (Figure 6.4). This database is initialised with the AU-coded descriptions of the stimulus facial expressions systematised according to the order in which the related images are shown to the user. When the user assigns an interpretation label to a particular facial expression, the pertinent AU-coded description stored in the temporary database is replaced with an appropriate 2-tuple vector representing the generated case (for an example see *(1)*). Then, the whole procedure is repeated once more in order to estimate the consistency of the performed labelling. If a certain expression has been labelled in the second round differently than in the first, the user is asked to label anew the pertinent facial expression.

Once the temporary database is supplied with the generated cases, the training phase of the system ends with indexing the generated cases and furnishing the dynamic memory of the experiences. The *Indexing* function retrieves and classifies the cases stored in the temporary database according to their attributed interpretation labels (Figure 6.4). It generates an appropriate ***n***-tuple vector for each interpretation label defined by the user (for an example see *(2)*). This partitions the cases into the interpretation categories. Finally, for each engendered interpretation category, the *Indexing* function begets a 3-tuple vector `(index(), label(), cases())` as exemplified in *(3)*, according to the following procedure:

1. *Coarse indexing*: For each previously generated ***n***-tuple vector assign the 1st element of it to the 2nd element of the new 3-tuple vector (i.e. assign the defined interpretation label to `label()`) and its last ***n-1*** elements to the 1st and the 3rd element of the new 3-tuple vector (i.e. assign all of the pertinent cases to both, `index()` and `cases()`).
2. *Fine indexing*: For each 3-tuple vector engendered in step 1 associate a counter *c* = ***0*** with each AU code or combination of AU codes belonging to `cases()`, reduce the set of combinations of AU codes belonging to `index()` by excluding each combination whose component AU codes are already included in `index()` (e.g. if AU1 belongs to the currently evaluated `index()` then exclude from it all combinations of AU codes that comprise AU1), repeat the process of `index()` reduction by excluding each combination of AU codes whose component is a combination of AU codes that already belongs to `index()` (e.g. if AU1+AU5 belongs to the reduced `index()` then exclude also from it all combinations of AU codes that comprise AU1+AU5).

Further, to keep calculation of quantitative rating on each scored interpretation label (performed by function F2 illustrated in Figure 6.2 and explained in detail in section 6.5) simple and accurate, the *Indexing* function augments the 3-tuple vectors begot by the procedure outlined above with an additional `full-expression()` element. For each interpretation category, the pertinent `full-expression()` is defined as a collection of all distinct AU codes that occur by themselves or in a combination with other AU codes within the pertinent `cases()`. Hence, a sample 4-tuple vector forming the interpretation class "*surprise*" (see example *(3)*) can be represented as given in *(4)*.

```
(index(AU1+AU2, AU5, …, AU27),
 label("surprise"),
 cases((AU1+AU2, 0), (AU5, 0), …, (AU27, 0))          (4)
 full-expression(AU1, AU2, AU5, …, AU27))
```

Eventually, the *Indexing* function generates a supplementary "*neutral*" interpretation class to account for the interpretation of an expressionless face. The pertinent 4-tuple vector can be represented as given in *(5)*. The reason for classifying AU25, AU26, AU38, and AU39 within the "*neutral*" interpretation category is that these facial actions do accompany the activation of most lower-face AUs (e.g. see Table 6.1 for the AU-coded descriptions of typical facial displays) but are too subtle to alter the impression made by the facial action they accompany (e.g. a smile caused by AU12 activation remains a smile independently of the activation of AU25 or AU26 or AU38 or AU39). In turn, since these AUs do not alter the impression made by the activation of other AUs (i.e. do not affect the interpretation of the shown facial expression), they do not form a part of another interpretation class generated by the *Indexing* function; these AU codes do not belong in the `index()`-, `cases()`- and `full-expression()` terms of any interpretation category other than the "*neutral*" category. Furthermore, the `full-expression()` term associated with the "*neutral*" category is empty since if an observed expression is interpreted as "*neutral*", it will always be interpreted as "*100% neutral*". Yet this will only be the case if the set of AU codes generated by the Facial Action Encoder part of the system is either empty (indicated as `()` in *(5)*) or contains a combination of AU25, AU26, AU38 and AU39 exclusively. For a further discussion on the case matching and retrieval performed by the Facial Expression Classifier, the reader is referred to section 6.5.

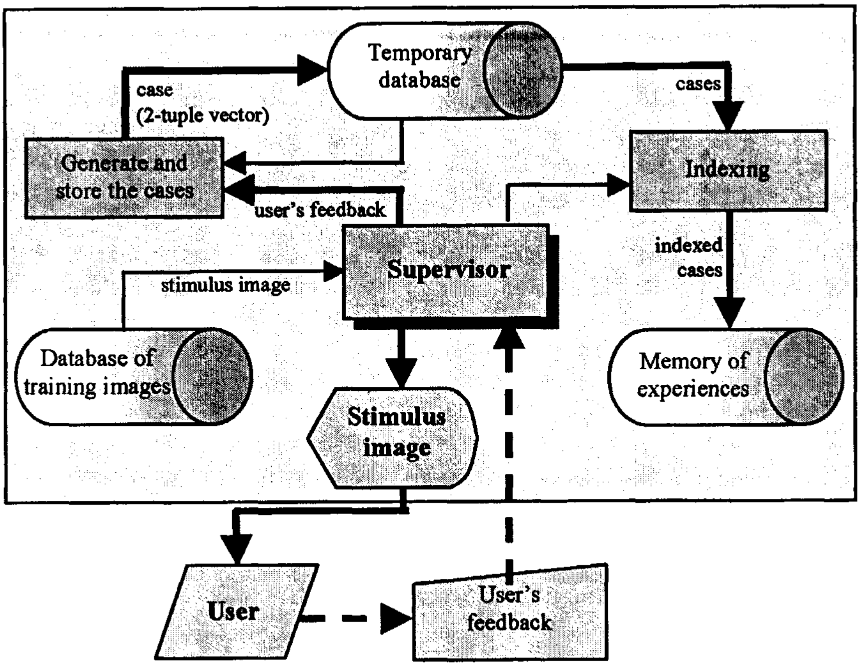


Figure 6.4: The training phase of the Facial Expression Classifier part of ISFER

```
(index((), AU25, AU26, AU38, AU39),
 label("neutral"),
 cases(((), 0), ..., (AU39, 0))                        (5)
 full-expression())
```

Eventually, the dynamic memory of experiences is rendered with the 4-tuple vectors begot by the *Indexing* function. In turn, the cases (facial expressions) constituting the dynamic memory of experiences are organised into *expression pools* according to their thematic similarity (i.e. interpretation) and into *pool hierarchies* according to their typicality (i.e. occurring frequency).

6.5 Quantified user-profiled expression interpretation

This section elucidates the processing of the Facial Expression Classifier part of ISFER. It performs both: (i) interpretation of the input quantified AU codes and the certainty of that data in terms of multiple, quantified, user-defined interpretation labels and the certainty of these conclusions, and (ii) adaptation of the utilised dynamic memory of experiences according to user-provided feedback on the meaning that he/she associates with various facial expressions. The section will first explain the process of interpreting the input data (i.e. the interpret mode of the Facial Expression Classifier) by examining how the first three functions of the Facial Expression Classifier (F1, F2, F3; Figure 6.2) exploit the knowledge stored in the dynamic memory of experiences and support the interpretation process. Then, the process of altering and augmenting the dynamic memory of experiences (i.e. learn mode of the Facial Expression Classifier) will be explained by exploring how function F4 (Figure 6.2) supports this adaptation process.

F1: Retrieval

As already explained in section 3.6, retrieval is the primary process in a CBR system. The goal of retrieval is, given the description of the current problem, to retrieve the most similar case or cases from the existing case base. The simplest form of retrieval consists of applying the first nearest-neighbour algorithm, that is, of matching all cases of the case base and returning just one best match (e.g. as proposed in (Pantic and Rothkrantz 2000c)). Yet, as already explained, this method is usually too slow, especially in the case of a large case base. A pre-selection of cases is therefore usually made based upon the indexing structure of the utilised case base and/or some ranking method based upon retrieval statistics for cases constituting the case base. As illustrated in Figure 6.5, if the retrieval function F1 of the Facial Expression Classifier is used, a pre-selection of cases is based upon the clustered organisation and indexing structure of the dynamic memory of experiences as well as upon the hierarchical organisation of cases within the clusters according to their typicality.

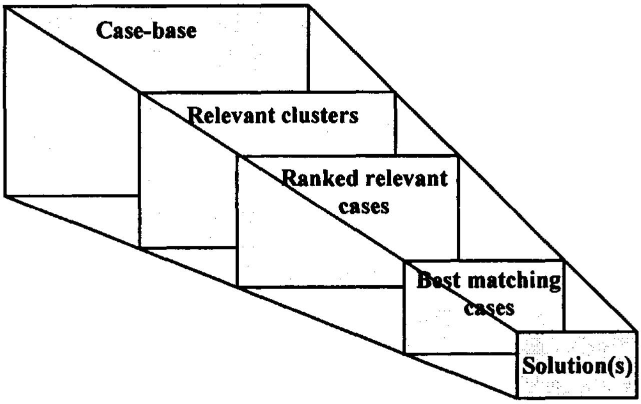


Figure 6.5: Schematic representation of retrieval function F1

Once the supervisor of the Facial Expression Classifier receives the input generated by the Facial Action Encoder part of ISFER (i.e. quantified AU codes and the certainty of that data), it activates function F1 in order to retrieve the case(s) constituting the dynamic memory of experiences that match best the input problem. The processing of F1 executes, in fact, the following procedure (Figure 6.5):

1. *Initialise five variables*: a list of AU codes, the ***AU-list***, which holds each and every input AU code; a list of relevant clusters, the ***cluster-list***, which will hold `label()` terms of those clusters; a list of ranked relevant cases, the ***case-list***, which will hold `cases()` terms of the relevant clusters; a list of best matching cases, the ***best-cases-list***, which represents a reduced ***case-list***; a list of solutions, the ***solution-list***, which will hold the final output of function F1. For a detailed explanation of different terms defining an expression pool of the dynamic memory of experiences, the reader is referred to the preceding section of this chapter. Go to step **2.**
2. *Determine the relevant clusters* by parsing the AU-coded description of the currently examined facial expression (***AU-list***) with the AU codes belonging to the `index()` terms of expression pools constituting the dynamic memory of experiences. Match the AU codes of the ***AU-list*** with (i) the combinations of AU codes belonging to the `index()` terms and then with (ii) the individual AU codes belonging to the `index()` terms. Each time a match is established, exclude the matched AU code(s) from the ***AU-list***, add the `label()` term of the relevant cluster to the ***cluster-list***, add all cases belonging to the `cases()` term of the relevant cluster to the ***case-list***, and add a separator at the end of the ***case-list***. Terminate the matching procedure when the ***AU-list*** is empty. Go to step **3.**

3. *Determine the best matching cases* by comparing each and every case of the ***case-list*** with the AU codes belonging to the initial ***AU-list*** defined in step **1**. If the currently examined case of the ***case-list*** is composed of AU codes that belong to the ***AU-list***, add the pertinent case to the ***best-cases-list***; otherwise, continue by examining the next case of the ***case-list***. Each time a separator is encountered, copy it as it is to the ***best-cases-list***. Terminate this step when there are no more cases of the ***case-list*** to be examined. Go to step **4**.
4. *Specify a monolithic solution* by contemplating the combination of all AU codes forming the initial ***AU-list*** defined in step **1** as a monolithic problem and match it with each and every case of the ***best-cases-list*** starting with the cases having the highest rank. If the combination of all AU codes belonging to the ***AU-list*** matches exactly a certain case of the ***best-cases-list***, count backwards the number ***n*** of the separators between the pertinent case and the beginning of the ***best-cases-list***, extract the ***n+1***th `label()` term from the ***cluster-list***, add it to the ***solution-list*** together with the matched case, and go to step **6**. If no case in the ***best-cases-list*** matches the combination of all AU codes belonging to the ***AU-list*** exactly, go to step **5**.
5. *Specify a compound solution* by contemplating the combination of all AU codes forming the initial ***AU-list*** defined in step **1** as a compound problem and match its sub-problems to the cases of the ***best-cases-list*** starting with the case representing the longest combination of ***k*** different AU codes and having the highest rank. Each time a match is found, exclude the matched AU codes from the ***AU-list***, count backwards the number ***n*** of the separators between the pertinent case and the beginning of the ***best-cases-list***, and extract the ***n+1***th `label()` term from the ***cluster-list***. If this `label()` term does not already belong to the ***solution-list***, add it to that list together with the matched case; otherwise, associate the matched case with the pertinent `label()` term already part of the ***solution-list***. Reduce ***k*** for 1 each time when there are no more cases in the ***best-cases-list*** to be examined that represent a combination of ***k*** different AU codes. When the ***AU-list*** is empty, go to step **6**.
6. *Terminate the execution of this algorithm*: (i) for each case *(a1+...+aN, j)*, which forms a part of the ***solution-list***, enlarge the retrieval statistics ***j*** for 1, and (ii) redefine the solution-list by grouping together, per `label()` term, the AU codes of the cases that are associated with the pertinent `label()` term (e.g. if the ***solution-list*** is ***("surprise", (AU1+AU2, 21), (AU27, 18), "happiness", (AU6+AU12, 34))***, then the redefined ***solution-list*** will be ***(("surprise", AU1, AU2, AU27), ("happiness", AU6, AU12))***.

A successful termination of this algorithm, resulting in an interpretation of the currently examined facial expression in terms of user-defined interpretation labels, is ensured. This is because the dynamic memory of experiences is initialised as explained in the preceding section. Namely, the facial images stored in the database of training images (Table 6.1) display each and every facial action that the Facial

Action Encoder part of the system is able to encode from an input facial image (except of AU25, AU26, AU38, and AU39 which are subsequently added to form the “*neutral*” expression pool of the dynamic memory of experiences; see section 6.4). As a result, the dynamic memory of experiences is initially endowed (within both, the `index()` and the `cases()` terms of the defined facial expression pools) with each and every micro-event that can possibly be encountered either as a monolithic problem or as a part of a compound problem. Hence, a successful execution of the pre-selection procedure of function F1 (step 1 to step 3 of the algorithm given above) is ensured. In turn, this yields a successful retrieval of the best matching case(s) for solving the problem currently presented to the system (step 4 or step 5 of the algorithm given above).

F2: Quantification

Once the supervisor of the Facial Expression Classifier (Figure 6.2) has achieved the goal of interpreting the input AU-coded description of the currently examined facial expression in terms of user-defined interpretation labels it will try to assign a quantitative rating to each of the scored interpretation labels. In sections 6.2 and 6.3 the importance of facilitating a quantified expression classification into multiple user-defined interpretation categories has been already emphasised. This issue is elucidated here, once more, by means of an example (for a similar example the reader is referred to section 2.2, Figure 2.6). Consider the blended facial-affect displays shown in Figure 6.6. Each of the two facial expressions might be classified in two basic emotion categories as defined by Ekman (1982): disgust and sadness in the case of the left-hand-side image and disgust and happiness in the case of the right-hand-side image. Nevertheless, according to the descriptions of these prototypic expressions of emotion as given by Ekman, the left-hand-side facial display belongs “more” to sadness than to disgust while the right-hand-side expression “equally” belongs to disgust and happiness (in accordance with the percentage of the displayed facial actions classified in one of the emotional categories as given in Pantic and Rothkrantz (2000b)). This gives us a hint that the interpretation of facial expressions would be more accurate if given in terms of quantified interpretation labels (even though it is vague which computational method would be the most appropriate for achieving such an interpretation). As discussed in sections 6.2 and 6.3, the psychologist James Russell and colleagues came to the same conclusion (Russell and Fernandez-Dols 1997). In order to quantitatively rate each user-defined interpretation label scored

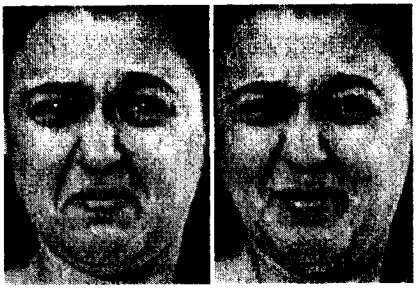


Figure 6.6: Facial displays of blended emotions: disgust-sadness and disgust-happiness

previously by means of function F1, the supervisor of the Facial Expression Classifier activates function F2.

In general, function F2 assigns an intensity level to each of the scored interpretation labels based upon the assumption that each AU code forming a component of a case being classified in the pertinent interpretation category (i.e. expression pool) has the same influence on quantitative rating (intensity) which is to be associated with the label ensuing that interpretation category. Thus, the ratio of the AU codes belonging to both the input AU-coded description and the `full-expression()` term of the pertinent facial expression pool to the total number of AU codes belonging to the `full-expression()` term might decide the quantification issue. For instance, if the `full-expression()` term of a scored interpretation category "*x*" contains five distinct AU codes and three of those, say ***a1***, ***a2***, ***a3***, belong to the input set of AU codes and have been classified in the "*x*" interpretation category (i.e. the result of function F1 is ***solution-list(("x", a1, a2, a3), ...)***), then a quantitative rating 60% will be associated with the interpretation label "*x*". Yet this ratio-based quantification method has a number of drawbacks. Two major ones are:

- *Co-occurrence rules implied by facial anatomy might be ignored.* Suppose that the facial displays of AU9 (wrinkled nose) and AU10 (raised upper lip) have been both interpreted by the user as "*disgust*". In that case the pertinent `full-expression()` term will contain both AU9 and AU10. Suppose further that no other facial-action display has been interpreted as "*disgust*" and that the AU-coded description forming the input to the Facial Expression Classifier contains just AU9. The ratio-based quantification method outlined above will then result in "*50% disgust*". Nevertheless, since the activation of AU9 obscures the activation of AU10 (FACS, Ekman and Friesen 1978) the correct interpretation should be "*100% disgust*".
- *The intensity of the activation of the facial actions is not taken into account.* Suppose that the user merely interpreted the facial display of AU12+AU6 (raised mouth corners and raised cheeks) as "*happiness*" and that the input AU-coded description generated by the Facial Action Encoder part of the system contains 20% AU12 and 20% AU6. Since the ratio-based quantification method outlined above does not take into account the intensity of the activation of facial actions, it will generate the result "*100% happiness*". Yet this would only be correct if the input were 100% AU12 and 100% AU6.

Function F2 applies therefore an adjusted ratio-based quantification method that addresses the two problems explained above. For each ***("x", a1, ..., aN)*** forming a part of the ***solution-list*** representing the output of function F1, and each ***(q1, ..., qN)*** holding the intensities associated to ***a1, ..., aN*** by the Facial Action Encoder, the processing of F2 executes the following procedure:

1. *Initialise variables*: a list of AU codes, the ***AU-list***, which holds ***a1, ..., aN***; a list of intensities, the ***Q-list***, which holds ***q1, ..., qN*** associated with ***a1, ..., aN***; a list

of AU codes, the ***full-list***, which holds all AU codes belonging to the `full-expression()` term of the "*x*" interpretation class; a real variable, the ***x-intensity***, which will hold the quantitative rating on the "*x*" interpretation label and form the output of function F2. Go to step **2**.

2. Reduce the ***full-list*** for each AU code belonging to the ***AU-list*** by applying the co-occurrences rules defined in FACS and listed in Table 6.2. For example, since the activation of AU7 cannot be encountered together with activation of AU6, AU9 and/ or AU41, if AU7 belongs to the ***AU-list*** and any of AU6, AU9, and/or AU41 belongs to the ***full-list***, AU6, AU9, and/or AU41 should be excluded from the full-list. Go to step **3**.
3. *Calculate **x-intensity*** as:

$$\textbf{\textit{x-intensity}} = \frac{1}{m}q1 + \cdots + \frac{1}{m}qN$$

where ***m*** is the number of different AU codes belonging to the ***full-list*** reduced in step **2** and ***q1, ..., qN*** belong to the ***Q-list***. Hence, for example, if AU12 and AU6 form the reduced ***full-list*** and the ***AU-list*** holds (AU6, AU12) while the ***Q-list*** holds (20, 20), then the ***x-intensity*** = 20.

Table 6.2
Co-occurrences rules: if the facial action listed in the 1st / 3rd column is activated, none of the facial actions listed in the corresponding 2nd / 4th column can be scored. For details on various AUs and their model-based representation within ISFER, see Tables 5.5 and 5.7.

	Absent AUs		Absent AUs
1	9	23	8, 16, 18, 19, 24, 25, 27, 28, 28b, 28t, 35
5	41	24	8, 9+17, 10+17, 12+17, 13+17, 15, 16, 18, 19, 20, 23, 25, 27, 28, 28b, 28t, 35, 38, 39
6	7	25	19, 23, 24, 26, 27, 28, 28b, 28t, 35, 36b, 36t
7	6, 41, 9	26	25, 27
8	9, 10, 12, 13, 15, 17, 18, 20, 23, 24, 28, 28b, 28t, 35, 38, 39	27	23, 24, 25, 26, 28, 28b, 28t, 29, 35,
9	1, 7, 8, 10, 38, 39	28	8, 10, 16, 17, 23, 24, 25, 27, 28b, 28t, 35, 36b, 36t, 38, 39
10	8, 9, 12, 13, 28, 28b, 28t, 38, 39	28b	8, 10, 16, 17, 23, 24, 25, 27, 28, 28t, 35, 36b, 36t, 38, 39
12	8, 10, 13, 15, 20, 38, 39	28t	8, 10, 16, 17, 23, 24, 25, 27, 28, 28b, 35, 36b, 36t, 38, 39
13	8, 10, 12, 15, 20, 38, 39	29	27
15	8, 12, 13, 20, 24, 38, 39	35	8, 16, 18, 23, 24, 25, 27, 28, 28b, 28t
16	18, 23, 24, 28, 28b, 28t, 35	36b	19, 25, 28, 28b, 28t
17	8, 28, 28b, 28t	36t	19, 25, 28, 28b, 28t
18	8, 16, 20, 23, 24, 35, 38, 39	38	8, 9, 10, 12, 13, 15, 18, 24, 28, 28b, 28t, 39
19	23, 24, 25, 36b, 36t	39	8, 9, 10, 12, 13, 15, 18, 24, 28, 28b, 28t, 38
20	8, 12, 13, 15, 18, 24	41	5, 7

F3: Evaluating the certainty of the conclusions

The last goal that the supervisor will try to reach while the Facial Expression Classifier is in the interpret mode is to calculate the certainty factors CF_x and $CF_{I(x)}$ associated with each interpretation label "*x*" (output of function F1) and its intensity *I(x)* (output of function F2). To this end, the supervisor activates function F3.

For each *("x", c1, ..., cM)* forming a part of the ***solution-list*** representing the output of function F1, where *(c1, ..., cM)* are the matched cases, which are associated with the interpretation label *"x"* and contain AUs *(a1, ..., aN)* (see the processing of function F1 explained above), and for each $(CF_{a1}, \ldots, CF_{aN})$ holding the certainty factors associated by the Facial Action Encoder to ***a1, ..., aN***, function F2 calculates the certainty factor $CF_x = C*100$ according to the following formula:

$$\mathbf{C} = \mathrm{P}(\text{"x"}|c1\wedge\ldots\wedge cM) = \frac{P(c1\wedge\ldots\wedge cM|\text{"}x\text{"})P(\text{"}x\text{"})}{P(c1\wedge\ldots\wedge cM)} = \frac{{}^{M}\!/_{K} * typicality(\text{"}x\text{"})}{min(CF_{ai},\ldots,CF_{aN})/100}$$

where K is the total number of cases c_i belonging to the `cases()` term associated with the interpretation category "*x*" and ***typicality***("*x*") = *S*("*x*") / ***T***, where *S*("*x*") = $\sum_{i=1}^{K} rank(c_i)$, ***rank***(c_i) is the retrieval statistic for the case c_i, and $T = \sum S(\text{"}x\text{"})$ for all expression pools "*x*" defined in the dynamic memory of experiences.

For each *(q1, ..., qN)* holding the intensities associated by the Facial Action Encoder to ***a1, ..., aN***, and for each $(CF_{q1}, \ldots, CF_{qN})$ holding the certainty factors associated by the Facial Action Encoder to ***q1, ..., qN***, function F2 calculates the certainty factor $CF_{I(x)}$ according to the following formula:

$$CF_{I(x)} = \min(CF_{q1}, \ldots, CF_{qN})$$

F4: Adaptation

As noted in section 6.3, the processing of the Facial Expression Classifier can be invoked in either an interpret mode or a learn mode. As long as the user does not provide a novel interpretation label for the currently examined expression, the Facial Expression Classifier runs in the interpret mode. By providing a novel interpretation label for the currently examined facial expression (or a part of it), the user triggers the learn mode of the system. In that case, the supervisor will receive the pertinent user's feedback (denoted with dotted lines in Figure 6.2) and will try to reconstruct the dynamic memory of experiences to reflect the pertinent encountered novel case.

As already discussed in section 3.6, the utilised case base can be adapted to the current user's needs/wishes automatically (by the CBR system itself) or manually (by the user). Adaptation makes the CBR systems more complex, but not necessarily more powerful; adaptation may reduce system reliability, especially when mistakes made by the system are expensive (Mark et al. 1996). Therefore, in many instances no attempt is made towards automatic adaptation; the user carries out the adaptation instead. Yet, even if no automatic adaptation takes place, there are always some situations where the current problem is so similar to the retrieved case that the system can automatically use the old solution and modify it in the appropriate direction (Maes 1994). In other words, generally the utilised case base can be adapted in a *hybrid way*, combining automatic and manual adaptation. Although Maes does not employ the term "hybrid adaptation", he identifies three situations that reflect incisively such an adaptation:

1. If the similarity between the retrieved case and the current problem can be expressed using a single value ***S*** and if ***S*** is above a "do-it" threshold ***T1***, the system performs an automatic adaptation.
2. If the similarity value ***S*** is above a "tell-me" threshold ***T2*** and ***S*** < ***T1***, then based on the retrieved case, the system suggests a possible appropriate adaptation to the user who can accept it or adapt it further.
3. If ***S*** < ***T2***, then the system informs the user that it does not know the solution.

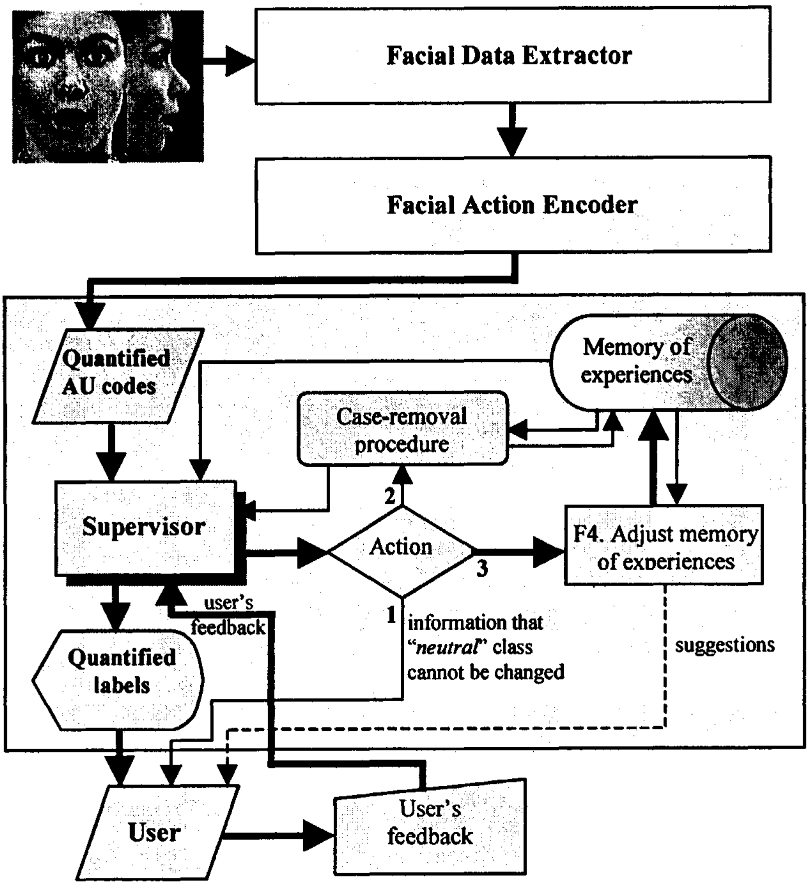


Figure 6.7: The learn mode of the Facial Expression Classifier part of ISFER

Each time the user is not satisfied with the facial expression interpretation provided by the system and renders his/her feedback on the issue, the supervisor of the Facial Expression Classifier applies a hybrid adaptation procedure in order to reconstruct the dynamic memory of experiences according to the wishes of the user. This procedure is, in fact, a 3D action space from which the supervisor selects one of the predefined plans in order to reach the active goal of adapting the dynamic memory of experiences according to the currently provided user's feedback (Figure 6.7). These plans are:

1. If the AU-coded description ***a1+...+aN*** of the facial expression for which a novel interpretation label has been introduced consists merely of AU25, AU26, AU38 and AU39, belonging to the "*neutral*" interpretation category (see the preceding section), the user is informed that he/she cannot introduce a novel interpretation category for the pertinent facial expression. An explanation is provided too, describing that these AUs accompany the activation of most lower-face AUs (e.g. see Table 6.1 for the AU-coded descriptions of typical facial displays) but are too subtle to alter the overall impression made by the facial action they accompany.
2. If the AU-coded description ***a1+...+aN*** (from which the AU codes belonging to the "*neutral*" interpretation category have been excluded) of the expression for which a novel interpretation label has been introduced matches exactly a specific case stored in the dynamic memory of experiences, the supervisor changes the content of the case base by removing the matching event and then activates function F4 to augment the dynamic memory of experiences with the novel case.
3. If no case stored in the dynamic memory of experiences matches exactly the AU-coded description ***a1+...+aN*** (from which the AU codes belonging to the "*neutral*" interpretation category have been excluded) of the currently examined expression (or a part of it) for which a novel interpretation label has been introduced, the supervisor activates function F4 to augment the memory of experiences with the pertinent additional case.

To remove from the memory of experiences the case ***(a1+...+aN, k)*** that matches exactly the AU-coded description ***a1+...+aN*** (from which the AU codes belonging to the "*neutral*" interpretation category have been excluded) of the expression for which a novel interpretation label has been introduced, the supervisor executes the following procedure:

1. *Initialise variables*: the ***case***, holding the case ***(a1+...+aN, k)***; the ***label***, holding the `label()` term of the cluster "*c*" to which the ***case*** belongs; the ***case-list***, holding the elements of the `cases()` term of the cluster "*c*"; the ***index-list***, being equal to the ***case-list***; an empty ***full-list*** that will hold the AU codes which will determine the `full-expression()` term of the redefined cluster "*c*". Go to step 2.
2. *Redefine cluster "c"*: Remove the ***case*** from the ***case-list*** and the ***index-list*** and define the `cases()` term with the elements of the reduced ***case-list***. Define the

`label()` term as the ***label***. Reduce the ***index-list*** by excluding each combination of AU codes whose component AU codes are already included in the ***index-list*** (e.g. if AU1 belongs to the ***index-list***, exclude from it all combinations of AU codes that comprise AU1); repeat the process of the ***index-list*** reduction by excluding each combination of AU codes whose component is a combination of AU codes that already belongs to the ***index-list*** (e.g. if AU1+AU5 belongs to the already reduced ***index-list***, exclude from it all combinations of AU codes that comprise AU1+AU5). Define the `index()` term by the elements of the ***index-list***. Define the ***full-list*** as a collection of all distinct AU codes that occur independently or in a combination with other AU codes in the ***case-list***. Define the `full-expression()` term by the elements of the ***full-list***. Terminate the execution of this procedure.

The supervisor activates function F4 as soon as it has ascertained that the dynamic memory of experiences does not contain a case that matches exactly the AU-coded description ***a1+...+aN*** (from which the AU codes belonging to the "*neutral*" interpretation category have been excluded) of the expression for which a novel interpretation label has been introduced. For each input ***("x", a1, ..., aN)***, representing the interpretation label "*x*" associated by the user with the expression coded in terms of ***a1, ..., aN***, the processing of F4 executes the following procedure:

1. *Coarse initialisation of variables*. Define the following variables: a list of AU codes, the ***AU-list***, which holds ***a1, ..., aN***; a list of clusters, the ***cluster-list***, which holds `label()` terms of all clusters existing in the dynamic memory of experiences; a list of AU codes, the ***index-list***, which holds all individual AU codes and combinations of AU codes belonging to the `index()` terms of each and every cluster (except the "*neutral*" cluster) belonging to the ***cluster-list***; and an empty list, the ***full-list***, which will hold all AU codes forming a part of the `cases()` term associated with the expression pool "*x*". Go to step **2**.
2. *Fine initialisation of variables*. Compare each element of the ***cluster-list*** with the input label "*x*". If the expression pool "*x*" already exists in the dynamic memory of experiences: add the new case ***(a1+...+aN, 1)*** to the `cases()` term associated with the expression pool "*x*", supply the ***full-list*** with all individual AU codes forming a part of the pertinent `cases()` term, adjust the `full-expression()` term associated with the expression pool "*x*" so that it contains the generated ***full-list***, go to step 3. If the expression pool "*x*" does not exist in the dynamic memory of experiences, generate a new expression pool "*x*": define the `index()` term as an empty list, define the `label()` term as `label("x")`, define the `cases()` term as `cases(`*(a1+...+aN, 1)*`)`, define the `full-expression()` term as `full-expression(`*a1,...,aN*`)`, go to step **3**.
3. *Redefine the* `index()` *term in an automatic way*. Reduce the ***index-list*** so that it contains either individual AU codes belonging to the ***AU-list*** or combinations of those. Reorganise the ***index-list*** according to the complexity of its elements: the

longer the combination of AU codes, the higher its rank in the ***index-list***. Represent the combination ***a1+...+aN*** of the AU codes belonging to the ***AU-list*** as ***j+k+l***, where ***j***=[] $\vee$ (***j*** $\in$ ***index-list*** $\wedge$ ***j*** is of the highest possible rank within the ***index-list***), ***k***=[] $\vee$ (***k*** $\in$ ***index-list*** $\wedge$ ***k*** is of the highest possible rank within the ***index-list***), ***l***=[] $\vee$ (***l*** $\in$ ***index-list*** $\wedge$ ***l*** is of the highest possible rank within the ***index-list***). Merely one of three different situations can be encountered (for a detailed explanation of this issue, the reader is referred to the discussion following the description of this procedure): ***j***=***k***=***l***=[], ***j*** $\in$ ***index-list*** $\wedge$ ***k*** $\in$ ***index-list*** $\wedge$ ***l***=[], or ***j*** $\in$ ***index-list*** $\wedge$ ***k*** $\in$ ***index-list*** $\wedge$ ***l*** = ***l1+...+lM*** where ($\forall$***i***$\in$[1,M], ***li*** $\neq$ [] $\wedge$ ***li***$\in$ ***index-list***). In the case that the first or the second situation is encountered, extend the `index()` term with the combination ***a1+...+aN*** of the AU codes belonging to the ***AU-list***, terminate the processing of this procedure. If the third situation is encountered, go to step **4**.

4. *Redefine the* `index()` *term based on the user's feedback.* Provide the user with a list of suggestions for possible appropriate adaptation of the `index()` term associated with the interpretation category "*x*". The list of suggestions should contain all possible combinations of at least two from the three terms defined in step **3**: ***j*** $\neq$ [], ***k*** $\neq$ [], ***l*** = ***l1+...+lM*** where $\forall$***i***$\in$[1,M], ***li*** $\neq$ [] (i.e. ***j+k***, ***j+l1***, ..., ***j+lM***, ***k+l1***, ..., ***k+lM***, ***l1+l2***, ..., ***j+k+l1***, ..., ***j+k+lM***, ..., ***j+k+l***). The list of suggestions is accompanied with the request to the user to select the one that best characterises the introduced interpretation label "*x*". Extend the `index()` term with the user-selected combination ***a1+...+aP***, ***P*** $\leq$ ***N*** of the AU codes belonging to the ***AU-list***, extend the `cases()` term with the case ***(a1+...+aP, 0)***, and terminate the processing of this procedure.

In order to understand why only one of the three situations defined in step **3** of function F4 can be encountered, the reader should keep in mind the case-removal procedure explained above and the initial furnishing process of the dynamic memory of experiences (section 6.4). Namely, the dynamic memory of experiences is initially endowed (within both the `index()` and the `cases()` terms of the defined expression pools) with each and every micro-event that can possibly be encountered either as a monolithic problem or as a part of a compound problem. In turn, a situation ***j***=***k***=***l***=[] will be encountered if and only if the case-removal procedure explained above has been executed prior to the execution of function F4. Encountering a situation ***j*** $\in$ ***index-list*** $\wedge$ ***k*** = [] $\wedge$ ***l*** = [] would mean that there is no case ***j*** in the case base (otherwise it will be removed by the case-removal procedure from the pertinent `cases()` term and the `index()` term) while, at the same time, ***j*** forms a part of an `index()` term. This is contradictory to the *Indexing* function defined for the initial furnishing of the dynamic memory (section 6.4) as well as with step 4 of function F4 and, therefore, such a situation cannot be encountered. A situation ***j*** $\in$ ***index-list*** $\wedge$ ***k*** $\in$ ***index-list*** $\wedge$ ***l***=[] will be encountered if the AU-coded description ***a1+...+aN*** of the expression for which a novel interpretation label has been introduced can be represented as $j+k = a1+...+aN$, where j and k are two indexes belonging to the `index()` terms of the dynamic memory of experiences. Finally, a situation $j \in$ ***index-list*** $\wedge\, k \in$ ***index-list*** $\wedge\, l = l1+...+lM$ where ($\forall i \in [1,M]$, $li \neq [] \wedge li \in$ ***index-list***) will be encountered if the AU-coded description $a1+...+aN$ of the expression for which a novel interpretation label has been introduced cannot be represented as a combination of two indexes belonging to the `index()` terms, but as a combination of three or more indexes belonging to the `index()` terms.

As explained in section 6.4, in order to facilitate accurate retrieval, the indexes that characterise an expression pool, should be uniquely defined for that pool. After the dynamic memory of experiences is initialised, the indexes defined by an `index()` term of an existing interpretation category are unique for that category. The indexing performed in step 2 of the case-removal procedure as well as that of step 3 and step 4 of function F4 keep the indexes uniquely defined across the existing interpretation categories (i.e. each existing category is uniquely characterised by the indexes defined in the pertinent `index()` term). In addition to being uniquely defined across the expression pools partitioning the dynamic memory of experiences, the indexes should be also kept simple; the "longer" the combination of AU codes forming an index, the less facial expressions will be presented to the system of which it will form a part. This issue can be best elucidated by means of an example. Suppose that the AU-coded description of the facial expression for which a novel interpretation label "*x*" has been introduced is $a1+...+a7$ which, when decomposed in terms of indexes, can be represented as $j + k + l + m = (a2+a5+a7) + (a1+a6) + a3 + a4$. Suppose further that function F4 has two versions: (i) step 4 does not exist and the interpretation category "*x*" is automatically characterised by $a1+...+a7$, and (ii) step 4 exists and the user selects $a3+a4$ to characterise the interpretation label "*x*". If the AU-coded description of the expression to be interpreted next by the Facial Expression Classifier is a combination of $a1, ..., a7$, containing at least $a3$ and $a4$ and not more than six different AU codes, then: (i) in the case of the first version of function F4, the system processing will result in a similar (incorrect) interpretation as was the case with the expression $a1+...+a7$, and (ii) in the case of the second version of function F4, the processing of the system will result in the (correct) label "*x*" combined further with some other interpretation label(s) begotten while interpreting expression $a1+...+a7$. In summary, step 4 of function F4 ensures that the generated indexes are kept as simple as possible.

As noted in section 6.3, the adaptation knowledge also includes the knowledge about how the correctness of a novel solution can be evaluated. Within the Facial Expression Classifier, the correctness of a new solution introduced by the user is automatically ensured. Step 3 and step 4 of function F4 ascertain this facet of the Facial Expression Classifier. Namely, whenever the user triggers the learn mode of the system, the dynamic memory of experiences is adjusted according to the user's feedback either (i) by executing step 3 of function F4, which keeps the case base consistent, or (ii) by executing step 4 of function F4, which constrains the user to select one of the proposed solutions, each of which does not violate the consistency of the case base. Hence, an additional evaluation of the correctness/consistency of a novel solution introduced by the user is unnecessary in the case of the Facial Expression Classifier.

6.6 Discussion and key challenges for future research

Darwin's (1965/1872) pioneering studies started a more than a century-long debate about whether observers can accurately judge the emotion shown in a facial expression. This issue is related to the question of whether specific expressions actually correspond to particular emotions. Over the decades, clearer conceptualisation of this problem delimited numerous critical issues that are still debated in basic psychological research. The most intriguing of these, forming the core of the two disparate studies on the perception of human facial affect (i.e. Paul Ekman and colleagues vs. James Russell and colleagues), can be summarised as follows:

- Do rapid facial signals (i.e. facial expressions) convey messages about genuinely felt emotions or are they merely a part of socially constructed communicative behaviour (e.g. emblems: symbolic communicators such as the wink, illustrators: communicators highlighting speech, regulators: conversational mediators such as nods and smiles)?
- Are there a number of facial expressions that are prototypic for a number of emotions; is each of those expressions universally associated with a single emotional meaning and, given this emotional labelling, is the pertinent expression universally produced? Are variations allowed? If so, are we still talking about prototypical (basic) emotion expressions and what variations in a prototypical expression are universally perceived as belonging to the same basic emotion? How much of a known prototypical facial expression should be displayed for observers to assign a basic emotional label? Is it then the prototypical expression itself or its parts that are associated with certain (basic) emotional label?
- How are blends of different emotion expressions within the same facial expression perceived and judged?
- Are there differences among specific cultures in the perception and production of basic emotion labels? If so, are prototypic emotion expressions prototypic at all; in other words, is the claim that they are universally perceived and produced valid?
- How do the observer's personal characteristics affect the perception of emotional facial expressions? Is the perception of emotion expressions universal, independently of the observer's sex, age, familiarity with the expresser,

extroversion vs. introversion? If not, even for a single observer, can it be claimed that the expressions postulated to be prototypical for a number of emotions are universally perceived?

- How do the expresser's personal characteristics affect the production of facial affect expressions? Are there differences among people in production of emotional facial expressions (e.g. due to expressiveness, extroversion, sex or age)? Once more, if this is the case, even for a single expresser, are the prototypic emotion expressions prototypic at all, that is, is the claim that they are universally produced valid?

Unlike any other automated system for facial affect recognition presented in the literature up to date (see chapter 2 and section 6.3), ISFER, due to its Facial Expression Classifier part, is entirely independent on the debate amongst psychologists on the perception of facial expressions of emotion. Instead of favouring one of the relevant psychological studies of the opposed camps and struggle to validate their findings, the design of the Facial Expression Classifier part of ISFER is based upon the machine learning concept. Namely, rather than adopting debated a priori rules for facial affect recognition from facial images, the system learns the appropriate rules by interacting with the user on the meaning he/she associates with facial expressions. Besides that it facilitates the user to freely define at any time his/her own facial expression interpretation labels, ISFER achieves a fully automatic and robust interpretation of an input facial expression in terms of:

1. multiple user-defined interpretation labels,
2. a quantitative rating associated with each scored interpretation label, and
3. a certainty measure which is associated with each scored interpretation label and its quantitative rating separately based on the certainty of the input expression data propagated through the system.

In comparison to the existing explicit attempts to automate facial affect recognition (see chapter 2 and/or Pantic and Rothkrantz 2001a), the Facial Expression Classifier part of ISFER is fundamentally different by the use of the CBR. While, throughout this chapter it was proven that CBR is a suitable method for a user-profiled recognition of facial affect from static facial images, the reader might be reluctant to consider it efficient as well. Namely, as already explained in section 3.6, CBR systems have a number of typical disadvantages implying reduced efficiency of such systems. Let us consider each of those typical CBR drawbacks in the scope of ISFER:

1. *High storage requirements and long retrieval times due to a large case base.* The expansion of the dynamic memory of experiences utilised by the Facial Expression Classifier is controlled, as explained in section 6.3. This and the case retrieval utilised by the Facial Expression Classifier part of ISFER, which is based upon the clustered organisation of the dynamic memory of experiences (Figure 6.5), enable efficient case storage and retrieval.

2. *Cumbersome to handle dynamic domains.* CBR systems generally have difficulties in handling dynamic problem domains since they are usually strongly biased towards what already has worked. This may result in an outdated case base. Yet, as noted in section 6.3 and explained in section 6.5, the knowledge of the problem domain (i.e. the user-profiled interpretation of facial expressions) implemented within the Facial Expression Classifier is good enough to allow the usage of adaptation rules that can generate novel solutions from scratch: if the user changes his/her mind about the interpretation that should be assigned to a certain (previously encountered) facial expression, the old (incorrect) case will be removed from the dynamic memory of experiences and the novel case will be added (Figure 6.7). Thus, the Facial Expression Classifier handles effectively and efficiently the dynamic problem domain of user-profiled interpretation of observed facial expressions.
3. *Difficulties in handling noise.* Unsuccessful assessment of noise present in a problem situation currently posed to a CBR system may result in the same problem being unnecessarily stored numerous times in the case base because of the differences due to the noise. In turn this implies inefficient storage and retrieval of cases. In the case of ISFER, the Facial Action Encoder part of the system deals with inaccurate, partial and redundant data generated by the Facial Data Extractor part of the system (sections 5.4 and 5.6) and associates a certainty measure to each of its conclusions based upon the noise encountered in the input data. In turn, the data that the Facial Expression Classifier part of ISFER employs for case storage and retrieval (i.e. AU codes describing the input facial expression) may be considered noise-free since the noise present in a problem situation currently posed to the Facial Expression Classifier is encoded separately, within the certainty measures accompanying each of the input AU codes. Hence, though noise is a typical problem of CBR systems, in the case of ISFER, noisy input problems have no effect on the successfulness of case storage and retrieval performed by the Facial Expression Classifier. At least, this is the case as far as a repetitive storage of a case is concerned. Yet the adequateness of the case-base adaptation can be affected by noisy input. Namely, the correctness of the data generated by the Facial Data Extractor and the Facial Action Encoder affects the correctness of the conclusions obtained by the Facial Expression Classifier (function F3, described in section 6.5, addresses the issue of propagating input data uncertainty through the system). In turn, if the user only considers the affect-based interpretation of the input facial expression with which he/she does not agree, and does not take into account that the pertinent interpretation might be merely a result of inadequately handled noise present in the input image, he/she may trigger the learn mode of the system and change the system's facial affect interpretation mechanism unnecessarily. Nevertheless, the system *does* provide the user with information that can be used to decide whether an unapproved interpretation of the input facial expression was generated due to the noise present in the pertinent input image or due to an inadequately trained

interpretation mechanism. The certainties associated with the quantified AU codes and the quantified interpretation labels provide this information. Hence, if the user takes this information into account prior to triggering the learn mode of the system, the case base of the Facial Expression Classifier can be adapted adequately.

4. *Cumbersome to achieve fully automatic operation.* In a typical CBR system, the problem domain is usually not fully covered. As a result, some problem situations can occur for which the system has no solution. In such cases, a typical CBR system usually expects input from the user. However, this is not the case with the Facial Expression Classifier part of ISFER. As explained in section 6.4, the process of initial furnishing of the dynamic memory of experiences generates cases covering the whole problem domain (Table 6.1, Figure 6.4). Once the dynamic memory of experiences is initialised, the Facial Expression Classifier could operate fully automatically. Of course, in that case, the system would not enhance its expertise using user-profiled interpretation of facial expressions, which forms its primary goal. Yet as far as fully automated operation of the system is concerned as an issue on its own, the process of initial furnishing of the dynamic memory of experiences facilitates fully autonomous processing of the Facial Expression Classifier.

In summary, the Facial Expression Classifier part of ISFER is unique in the field of machine perception of human facial affect because: (i) it is independent of the related debate in basic psychological research, (ii) it allows the user complete freedom in associating various meanings with various facial expressions, (iii) it facilitates the interpretation of blends of differently interpreted sub-expressions forming a single facial display, (iv) it incorporates the effect that the intensity of facial-muscle actions has on the quantitative rating of the interpretation label associated with the facial expression caused by those facial actions, (v) it provides a measure of confidence for the obtained interpretation, and (vi) it exploits CBR rather than eager learning methods typical for the existing automated systems for facial-affect recognition from facial images (chapter 2). Though it greatly enhances the state of the art in machine perception of human facial affect, the Facial Expression Classifier has a number of drawbacks representing, in some way, a set of challenges and opportunities facing the researchers in this area.

The Facial Expression Classifier classifies the expression data generated by the Facial Action Encoder part of the system. Since the Facial Action Encoder codes an input facial expression in terms of 29 AUs (i.e. 32 AU codes; Table 5.8), from 44 AUs defined in FACS (Ekman and Friesen 1978), it is not capable of encoding the full range of human facial behaviour. Consequently, because it has inherited this drawback from the Facial Action Encoder, the Facial Expression Classifier is not capable of interpreting the full range of human facial behaviour. As a result, for two facial expressions that differ in terms of displayed AUs, the system may generate the same AU-coded description and the same user-profiled interpretation. It is crucial, therefore, that the user is aware that the system automates only a part of FACS scoring, yielding the user-profiled interpretation of just a limited range of human facial behaviour. On the other hand, it should be emphasised that this is an inherited limitation; if the Facial Action Encoder were to automate the entire FACS scoring, the Facial Expression Classifier would yield user-profiled interpretations of the full range of human facial behaviour.

As far as the applied methodology is concerned, the shortcoming concerns the within-subject design of the process of initial furnishing of the dynamic memory of experiences. Namely, during the training phase of the system (Figure 6.4, section 6.4), the user is asked to judge, within a relatively short period of time, the entire set of stimulus posed facial images stored in the database of training images. As explained in section 6.2, this method invites a more direct comparison between various facial expressions than usual everyday encounters with facial expressions allow. In turn, the subject might feel called upon to notice the difference between two expressions and assign different labels to them. Yet, the decision to employ a within-subject design of the process of initial furnishing of the dynamic memory of experiences, which implies methodological criticisms outlined here, has been a conscious one. Namely, to solve the problem of incomplete records that could be used to define the cases to be stored in the dynamic memory of experiences, two approaches have been distinguished (section 6.4): (i) on-line case-base generation from scratch, which involves complex adaptation procedures and may make the development and maintenance of the CBR system cumbersome, and (ii) iterative case-base generation, which starts with an initial case-base furnishing which implies always a within-subject design of this process. Since both outlined approaches have drawbacks, the one involving less potential problems and less cumbersome development of the intended CBR system has been chosen.

Another peculiarity of the design of the Facial Expression Classifier, which might be thought of as an additional limitation of the system, concerns the process of quantitative rating of a scored user-defined interpretation label. Namely, the determination of the intensity level to be assigned to a particular interpretation label is based upon the assumption that each AU code forming a component of a case being classified in the pertinent interpretation category has the same influence on the quantitative rating of that interpretation label. On the other hand, as reported by Ekman (1982), the upper-face features might play a more important role in the recognition of facial affect than the lower-face features (see also chapter 7). If this is the case, then the upper-face AUs forming a part of a case being classified in a certain interpretation category should have a grater influence on the quantitative rating of the pertinent interpretation label than the lower-face AUs forming another part of the case in question. Yet, thus far, no functional anatomical study has been published that has shed any light on the specific neural mechanisms for interpreting facial expressions. Hence, whether the changes in the upper-face features are more important for the recognition of facial affect than those in the lower-face features, and what exactly the impact is of these changes on the perception of facial affect must first be determined by basic research, before the pertinent findings can be exploited within an automated system for facial-affect analysis.

A more important issue concerns the fact that the Facial Expression Classifier does not take into account the situation in which the facial expression to be interpreted occurs. On the other hand, as explained in section 6.3, the actual meaning of someone's facial behaviour depends to a large extent on this contextual information (Russell and Fernandez-Dols 1997). The problem of context sensing, as applied to the problem of automating facial affect analysis, can be divided into three sub-problems: (i) who the observed person is, (ii) where he is, and (iii) what he is doing. In other words, in order to emulate the user's performance in interpreting someone's facial expressions in terms of attitudinal/affective states, the computer must know:

1. *The identity of the currently monitored subject.* Based on this information, the system can compute in a person-dependent manner the intensity levels of the displayed facial actions. In the current version of ISFER, presented in this thesis, subject identification is achieved "manually". Namely, each time before a session with a new subject starts, the user executes an appropriate procedure that generates the database of extreme face-model deformations (section 5.5). If the current subject has not been analysed by the system before, the database of extreme model deformations is generated from scratch. Conversely, if the subject is known, his/her already existing database of extreme model deformations is utilised again.
2. *The overall situation in which the observed person acts.* The interpretation of the same facial behaviour could be different for two differing subject's environments. For instance, wide-open eyes can mean surprise if shown by a student monitored while attending a lecture, or frustrated fear if shown by an operator monitored in a nuclear power plant. The Facial Expression Classifier, presented in this section, does not take into account the environment in which the monitored subject acts while interpreting his/ her facial expressions.
3. *The current task the monitored subject is involved with.* At the finest level of context-dependent interpretation of facial behaviour, the variations in meaning of displayed facial expressions related to the specificity of the task that the monitored person currently performs should be accounted for. For instance, wide-open eyes can mean surprise if shown by an nuclear-power-plant operator while checking his e-mail, or fear if shown by the same operator but while checking for the reason of a just sounded alarm. The Facial Expression Classifier does not take into account the task-related reactions of the observed subject while interpreting his/her facial behaviour.

In summary, although automatic context sensing is crucial for an accurate automatic facial affect analysis, the affect-sensitive monitoring of human facial behaviour that ISFER performs is context-free. This is also the case for virtually all work done in the field of machine perception of human facial affect. Although it was initially thought that this research topic would be the hardest to solve, machine-performed context sensing has been proven remarkably tractable. For a discussion on advances and challenges in general-purpose context-sensing research, the reader is referred to (Pentland 2000). Yet, due to the complexity of this wide-ranging problem and the general luck of researchers having expertise in all necessary purpose-oriented computer vision techniques (such as person identification, facial expression analysis, image segmentation aimed at detecting the environment, gaze tracking, multiple-person tracking, etc.), the problem of context-dependent interpretation of human facial behaviour forms probably the most significant challenge for researchers of machine perception of human facial affect. Furthermore, if we take into consideration that, on one hand, speech affects the mouth and in turn the displayed facial expression (i.e. speech can be viewed as a kind of noise in machine perception of human facial affect) and that, on the other hand, facial expressions may give further meaning to the spoken words and that the spoken words may explain why an expression is displayed, the problem of untangling context-dependent meanings of human audio-visual communicative signals forms perhaps the most significant challenge for researchers of ubiquitous computing in general. For a further discussion on this topic, the reader is referred to chapter 8.

Finally, as already discussed in chapters 4 and 5, ISFER does not perform a temporal analysis of facial expressions. It has been developed to classify an input facial expression in terms of both multiple quantified facial action codes and multiple quantified user-defined interpretation labels from a static facial image rather than from a facial image sequence of the observed subject. The preceding sections of this thesis have separately enumerated many potential benefits that could accrue from efforts to account for the temporal aspect of facial expression analysis. This section indicates two additional facets of system's performance that could be advanced by including the time dimension of facial expression analysis into the automatic facial-affect-sensitive monitoring performed by ISFER. Those are:

1. *Enhanced effectiveness*: The inclusion of the temporal aspect of facial expression analysis will potentially enable the (user-profiled) interpretation of a wider range of facial affect behaviour. Let us consider the psycho-physiological states like the hypertension, stress, pain and frustration. They are all characterised by certain alterations of facial expressions (Schachter 1957, Vaughan and Lanzetta 1980, Prkachin and Mercer 1989). Yet temporal dynamics of the related facial expressions (a certain pattern of facial expressions observed over a time scale) rather than their configuration aspects encountered in a time instance make those states recognisable. Besides, a currently growing body of psychological research argues that timing of facial expressions is critical in the interpretation of any facial display (Bassili 1978, Bruce 1986, Izard 1990). For researchers of machine perception of human facial affect in general, and for future developers of ISFER in particular, this suggests investigation towards the design and development of an efficient machine-learning method which could enable the intended user-profiled facial-affect-sensitive monitoring tool to continuously enhance its

expertise through interaction with the user on the meaning that he/she associates with different spatio-temporal patterns of facial expressions.

2. *Enhanced efficiency*: With the current state of the art in processing of signals obtained by face-monitoring sensors (Pantic and Rothkrantz 2000d), noisy and missing data should be expected. As explained in chapters 2 and 5, an automated facial expression analyser (which is aimed at facial affect recognition or not) should be able to deal with these imperfect data and to generate its conclusions so that the certainty associated with them varies in accordance with the certainty of the input data. In the case of an automated facial-affect-sensitive monitoring tool such as ISFER, this can be achieved by considering the time-instance vs. the time-scale dimension of facial affect analysis. Namely, there is a certain grammar of facial affect behaviour, a pattern in the occurrence of affect expressions. Hence, only a certain subclass of these affect expressions with respect to the currently observed expression (time instance) and previously encountered affect expressions (time scale) is plausible. If the current input data reveal this statistically predicted facial affect behaviour, the certainty associated with that data should be "high" and the certainty of the drawn conclusion is to be computed accordingly. However, such a temporal analysis involves untangling the grammar of human facial behaviour, which is a rather unexplored topic even in the psychological and sociological research areas and certainly in the area of AI research. The issues that make this problem extremely difficult to solve in a general case concern the dependency of human behaviour upon personality, cultural and social vicinity, current mood, and the context in which the observed facial behaviour occurs. Yet, in an automated person-dependent face-monitoring tool such as ISFER, which interprets the observed facial behaviour in terms of interpretation labels defined by the current user, the grammar of the monitored subject's facial behaviour could be learned by "watching" that subject and interacting with the current user on his/her interpretation of the observed spatio-temporal patterns of the subject's facial behaviour. In turn, based upon the learned grammar of someone's facial affect behaviour, statistical predictions could be made about the facial expression of that person that is likely to be displayed next and could be further utilised to enhance system performance (i.e. to deal with noisy input data and partial occlusions of the subject's face and to enhance the process of computing the certainty measures to be associated with the system's results).

In summary, if automated facial expression analysis and facial affect recognition were based upon context-dependent (i.e. person-dependent, application-dependent, and task-sensitive) spatio-temporal analyses of facial image sequences, this would greatly advance the performance of ISFER as well as the state of the art in the field of machine perception of human facial expressions in general (see chapter 8 for a further discussion on this topic).

7 System evaluation

The more rigorously design and code inspection are performed, the better the quality of the final system.

(Kan 1995)

The first step in evaluating the performance of an automated system is to obtain a set of relevant test data. Because there are numerous areas where benefits could accrue from the automation of facial expression analysis (sections 5.1 and 6.1, Golomb and Sejnowski 1993), giving machines the ability to detect, track, and interpret human facial expressions attracted the interest of many researchers and became one of the hot topics in machine vision and AI research. Hence, it could seem that an accessible test database of images of faces would be readily available. Nevertheless, as already remarked by many researchers (e.g. Bowyer and Phillips 1998, Pentland 2000, Pantic and Rothkrantz 2000d, Cowie et al. 2001), no database of images exists that is shared by diverse computer-vision-research communities. In general, only isolated pieces of such a facial database exist, each of which has been made and exploited by a particular facial research community. An example is the FACS Dictionary (Friesen and Ekman 1987), which has been developed and used by the Human Interaction Lab of the San Francisco University of California (our group at the Delft University of Technology was not given permission to use it). In consequence, similar to other facial research communities, the students and research staff members working on the automation of facial expression analysis at the Knowledge Based Systems department of the Delft University of Technology collected their own database of static facial images (e.g. De Bondt 1995, Profijt 1995, Pantic 1996, Rothkrantz et al. 1998, etc.). This rather large database of static facial images (see section 7.1) has been used for evaluating the performance of ISFER.

The second issue in evaluating the performance of an automated system is that of validation. In general, validation studies address the question of whether the developed system does what it should do while complying with the pre-defined set of requirements (Saborido 1992). Validation studies on ISFER have been aimed at testing both the rules of the Facial Action Encoder part of the system and the dynamic memory components of the Facial Expression Classifier part of the system. More specifically, validation studies on ISFER addressed the question of whether its interpretations were acceptable to human observers judging the same facial images. Section 7.2 describes validation studies on the Facial Action Encoder part of ISFER. Section 7.3 is concerned with both the recall and the learning function of the Facial Expression Classifier part of ISFER. In both cases, the conclusions of ISFER were compared with those of human experts belonging to college personnel. Both qualitative and quantitative assessments using standard statistical techniques were carried out. Section 7.4 summarises the results of these studies, which support the claim that ISFER has an acceptable level of expertise.

Finally, this chapter focuses on the quality of ISFER in terms of user satisfaction with the system. However, a pragmatic examination of ISFER's usability cannot be performed since ISFER has not been actually deployed yet. What can be done, nevertheless, is to give an estimate of user satisfaction. Section 7.5 provides such an assessment.

7.1 Facial database

As already mentioned above and in spite of repeated references to the need for a readily accessible database of facial information that could be shared by diverse facial research communities all over the world, no such a common database has been established yet. The glaring lack of such a resource forms the major impediment to comparing, resolving, and extending the issues concerned with automatic facial expression analysis and understanding. This lack of a common testing resource slowed down not only the progress in applying computers to analyse human facial behaviour but also overall cooperation and collaboration among investigators of this research topic. The benefits that could accrue from a commonly used database of images of faces (both static and motion images) are numerous:

- Avoiding redundant collection of facial expression exemplars can reduce research costs: investigators can use one another's exemplars.
- Having a centralised repository for retrieval and exchange of imagery can improve research efficiency.
- Maintaining various test results obtained for a reference set of images and hence providing a basis for benchmarks of research efforts can increase research quality. This would also reduce the currently existing abundance of reports presenting rather insignificant achievements.

However, although it would be extremely beneficial to establish a common database of facial research efforts, no universally accessible database supplied with facial images and the related test results has been founded up to date. In consequence, virtually all active facial research communities, including the Knowledge Based Systems group at Delft University of Technology, have developed their own imagery databases that are used as benchmarks for their research efforts.

Since 1992, when the Knowledge Based Systems group started different projects on the automation of facial expression analysis, numerous diverse images of faces have been made and collected. As already mentioned in section 4.3, the full database contains currently over 1600 frontal, profile, and dual views of 25 different faces expressing hundreds of facial actions and their combinations. Since the images have been made for multiple needs of students and scientific staff members working in this research field, and without a clear idea to construct a common database of the face, technical standards and considerations for database images were never resolved. In other words, criteria for image resolution, colour, compression methods, and distribution mechanisms were never defined. Hence, the images, having various resolutions and colours, were scattered over various physical and virtual locations. In 1998, an attempt was made to organise the existing repositories of facial imagery and to create a centralised database of still images of faces (Schouwen 1998, Vollering 1998).

As far as the technical considerations for the database of static facial images are concerned, the following criteria have been defined:

- *Resolution*: the images should have standard PAL camera resolution, that is, when digitised, images should measure approximately 720×576 pixels.
- *Colour*: the images should be true-colour (24-bit) images or, if converted to grey-scale images, the colour depth of 24 bits should be reduced to 256 grey levels.
- *DB structure*: the images were divided into one of the three database clusters:
  - ➢ Portraits of faces that meet the above-given standards for resolution and colour (no in-plane or out-plane head rotations are present; see Figure 7.1). This part of the database contains approximately 400 images and includes those scanned from the photographs used as behavioural science research material.
  - ➢ Dual-view facial images (i.e. combined portraits and profiles of faces) that meet the above-given standards for resolution and colour and have been obtained by the mounted camera device shown in Figure 4.2 (see Figure 7.2 for examples of dual-view images). There are approximately 600 of those in the database.
  - ➢ Miscellaneous images of faces that meet multiple needs of researchers working on the topic but either do not meet the above-given standards for resolution and colour or cannot be classified into one of the previous

classes of images (e.g. "almost" frontal view images of faces where limited in-plane and out-plane head rotations are present, see Figure 7.3). This part of the database contains some 500 images and includes small clusters containing the images used for a specific research purpose (e.g. for testing a particular facial feature detector; see section 4.3).

- *Distribution*: the database is installed on our group's main server and can be easily accessed. An easy access has been achieved by a relaxed level of security that allows any student or group member, having a valid account on the group's server, a quick access to the database. This also frees the administrator of time-consuming identity checks for the database itself.
- *Security*: while individual researchers may add their own images to the database, the security status of such additions has not been determined. Though for all additions, especially those to Portraits and Dual-Views partitions of the database, it should be automatically checked (e.g. by case-based merging) whether they match the specified technical formats, no provision has been made for such a

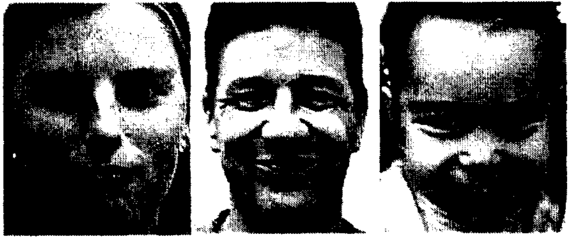


Figure 7.1: Examples of images belonging to the Portraits DB partition

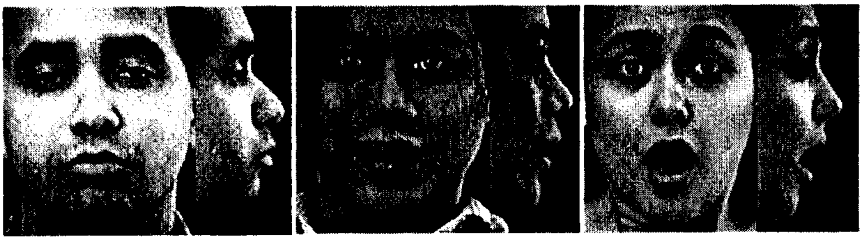


Figure 7.2: Examples of images belonging to the Dual-Views DB partition

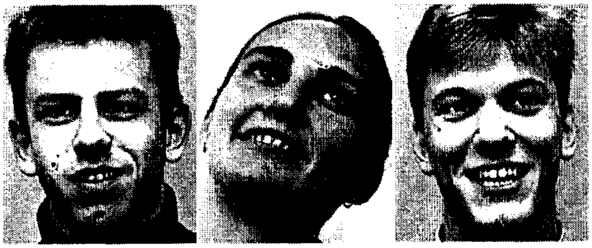


Figure 7.3: Examples of images belonging to the Miscellaneous DB partition

secure extension of the database. However, providing novel investigators with detailed instructions about the technical criteria for imagery inclusion in the database seems to be sufficient for keeping the database well organised.

The database images represent a number of demographic variables including ethnic background, gender, and age, and provide, in principle, a basis for generality of research findings. Overall, the subjects were students and college personnel (in total 25 different persons) of both sexes, young but still ranging in age from 20 to 45, and of either European, Asian, or South American ethnic background. In order to avoid effects of the unique properties of particular people, each DB partition images has been supplied with images of several individuals (e.g. the Dual-Views DB partition contains images of 8 different subjects of both sexes who differ in age and ethnicity).

Not only the issues of defining the technical requirements and ensuring demographic variability of database images are relevant for establishing a readily usable and malleable repository of research material: *metadata* should be associated with each image. Those data concern: the facial activity captured in an image (e.g. given in terms of AUs scored by a human FACS coder), the distinction between posed and spontaneous action (each may result in a different interpretation, see section 8.3), the circumstances under which an image was obtained (important for untangling the problem of context-dependency, see sections 6.6 and 8.3), etc. Though some metadata associated with some images exist in written documents (e.g. AU coding by human FACS coders), no effort has been made yet to determine, associate, and compile metadata for each and every database image. This forms an interesting playground for future facial researchers at the Knowledge Based Systems group that could prove to be extremely useful (see also sections 6.6 and 8.3).

7.2 Validation studies on the Facial Action Encoder

The aim of the validation studies on the Facial Action Encoder was to establish whether this part of the system is of good quality based on the correctness and reliability of the results it generates. First, the correctness of the applied rules for facial action coding was tested (qualitative validation of the rule base of the Facial Action Encoder). Then, the reliability of the results generated by the Facial Action Encoder part of ISFER was tested. The aim was to obtain an estimation of the measure of agreement between the facial actions encoded by the Facial Action Encoder and those encoded by human observers judging the same images (quantitative validation of the rule base utilised by the Facial Action Encoder). Both the interpretation and quantification of the displayed facial actions have been considered.

Qualitative validation of the rule base

As already explained in section 5.3, the Facial Action Encoder part of ISFER employs three different sets of rules for facial action coding in input images. If a portrait of the monitored face represents the input to the system, the rules describing AU codes in terms of the utilised frontal-view face model are applied (Figure 5.2, Table 5.5). In the case that a dual view of the monitored face forms the input to the system, the rules describing AU codes in terms of the utilised dual-view face model are applied (Figure 5.2, Table 5.8). Finally, in the case that merely the profile contour is successfully detected in an input dual-view facial image, facial action coding is performed based upon the rules describing AU codes in terms of the utilised profile-view face model (Figure 5.2, Table 5.7). Since the set of rules describing AU codes in terms of the utilised dual-view face model is a combination of the other two sets of rules, qualitative validation of the rule base used by the Facial Action Encoder considered testing of 22 rules listed in Table 5.5 and 24 rules listed in Table 5.7. Both sets of rules were tested using five experts (i.e. certified FACS coders) and 92 dual-view images. The aim was to estimate the correctness of the employed rules based upon the measure of agreement between human observers while judging facial expressions produced according to those rules.

First, two experts (say ***E1*** and ***E2***) were asked to produce 46 facial expressions: 22 expressions of separate AU activations displayed according to 22 rules describing AU codes in terms of the frontal-view face model (Table 5.5). and 24 expressions of separate AU activations displayed according to 24 rules describing AU codes in terms of the profile-view face model (Table 5.7). Both experts were asked to produce per rule only the changes in facial expression described in the pertinent rule and to "leave" the appearance of other facial features unchanged. Dual views were recorded and the acquired 94 images (per subject: 46 facial expressions of separate AU activations + a neutral expression) were given for evaluation to other two experts (say ***A*** and ***B***).

The employed questionnaires for scoring AUs have been made differently for each AU. Overall, each questionnaire was divided into two sections corresponding to the upper (eyebrows and eyes; 7 possible AU codes in total) and the lower facial features (nose, mouth, and chin; 25 possible AU codes in total). Depending on the AU (upper- or lower-face AU) for which the questionnaire was made, one of those sections was blackened. Also, depending on the rule according to which the relevant AU is produced, other parts of the pertinent questionnaire were blackened too. For instance, in the questionnaires for AU8, places for scoring AU9, AU12, AU13, AU15, AU17, AU18, AU20, AU23, AU24, AU35 were blackened since the rules for recognition of AU8 state that AU8 cannot be scored if any of these AUs is scored (see Tables 5.5 and 5.7).

Table 7.1
Comparison of human judgements of 44 stimulus images produced by experts *E1* and *E2* according to 22 rules given in Table 5.5 to those that would be produced by the utilised rule base. For 7 upper-face AUs displayed by either *E1* or *E2*, there are 14 possible agreements / disagreements (*aa* / *da*). For 15 lower-face AUs displayed by either *E1* or *E2*, there are 30 possible agreements / disagreements (*aa* / *da*).

	E1 up. AUs	*E1* low. AUs	*E2* up. AUs	*E2* low. AUs	Σ:
aa	13	28	14	30	85
da	1	2	0	0	3
Σ:	14	30	14	30	88

Table 7.2
Comparison of human judgements of 48 stimulus images produced by experts *E1* and *E2* according to 24 rules given in Table 5.7 to those that would be produced by the utilised rule base. For 2 upper-face AUs displayed by either *E1* or *E2*, there are 4 possible agreements / disagreements (*aa* / *da*). For 22 lower-face AUs displayed by either *E1* or *E2*, there are 44 possible agreements / disagreements (*aa* / *da*).

	E1 up. AUs	*E1* low. AUs	*E2* up. AUs	*E2* low. AUs	Σ:
aa	4	41	4	43	92
da	0	3	0	1	4
Σ:	4	44	4	44	96

For each stimulus image, experts ***A*** and ***B*** were asked to encode the displayed facial action by comparing the stimulus image to the neutral facial expression of the relevant subject and then to fill in the questionnaire provided for that image by selecting one of the possible (not-blackened) AU codes. Per stimulus image, the number of agreements and disagreements has been counted next. For example, if a stimulus image representing activation of AU36b was judged by expert ***A*** to be AU24 while expert ***B*** selected AU36b in the questionnaire, then for that image the agreement was ***aa*** = 1 and the disagreement was ***da*** = 1. The number of agreements and disagreements for the upper- and the lower-face AUs over 44 images, produced by experts ***E1*** and ***E2*** according to 22 rules given in Table 5.5, is summarised in Table 7.1. A similar summarisation is provided in Table 7.2 for 48 images produced by experts ***E1*** and ***E2*** according to 24 rules given in Table 5.7.

From Tables 7.1 and 7.2 it is apparent that in virtually all cases, the images showing the activation of a certain AU produced by expert ***E2*** according to the tested rules were labelled with the same AU code by experts ***A*** and ***B***. To confirm this finding, another expert (say ***C***, different from ***E1***, ***E2***, ***A***, and ***B***) was asked to evaluate 46 images of separate AU activation produced by expert *E2* according to the tested rules given in Tables 5.5 and 5.7. In the case of this novel human expert *C*, testing the correctness of the rules utilised by the Facial Action Encoder proceeded along the same lines as before: the judgments of expert *C* were compared to those that would be produced by the utilised rule base. In this test, the agreement was 100%. In other words, expert *C* labelled each stimulus image, depicting a certain AU produced according to the tested rules, with the prtinent AU code.

Quantitative validation of the rule base: Interpretation

In order to test the reliability of the results generated by the Facial Action Encoder part of ISFER, a quantitative validation was carried out. The aim was to estimate the measure of agreement between the human judgements and the interpretations produced by ISFER as to the depicted facial actions (AU codes) in test images. In this validation test, the reliability of the intensity level associated with each of the scored AU codes has not been considered. With respect to the depicted level of intensity assigned to a scored AU code, the reliability of the conclusions generated by the Facial Action Encoder part of ISFER was tested separately (see the following sub-section).

Table 7.3
Comparison of judgements given by two experts for five prominent features over 560 stimulus images. For *n* AUs of a certain feature (*n*=3 for eyebrows, nose; *n*=4 for eyes, chin; *n*=18 for mouth) there are 560x*n* possible agreements / disagreements (*aa* / *da*).

	eyebrows	eyes	nose	mouth	chin	Σ:
aa	1631	2195	1546	9474	2150	16996
da	49	45	134	606	90	924
Σ:	1680	2240	1680	10080	2240	17920

First, two experts (i.e. certified FACS coders, say *E1* and *E2*) were asked to evaluate 560 dual-view images of faces constituting the Dual-Views part of the facial database (section 7.1). The questionnaire for scoring AUs was divided into five sections corresponding to five prominent facial features: eyebrows (3), eyes (4), nose (3), mouth (18), and chin (4). The numbers in parentheses represent the number of AU codes that the Facial Action Encoder part of ISFER is able to encode for the given feature in an input dual-view facial image (see also Table 5.8). Per stimulus image and per section of the questionnaire, the number of agreements and disagreements was counted next. For example, if for a stimulus image expert *E1* selected AU2 while expert *E2* selected AU1 en AU2 in the eyebrows section of the questionnaire, then for that image the agreement about AUs affecting the eyebrows was ***aa*** = 2 (i.e. the experts agreed that AU2 is activated and AU4 is not activated) and the disagreement was ***da*** = 1 (i.e. the experts disagreed about the activation of AU1). The number of agreements and disagreements of the two experts about the depicted AU codes for each facial feature over 560 images of facial expressions (representing various activations of one or more AUs) is given in Table 7.3. It was found that for 454 images, the experts agreed about the displayed AUs. The original set of 560 images has been reduced to a set of 454 test images accordingly.

Table 7.4
A comparison of ISFER conclusions and human judgements for five prominent facial features in 454 test dual views. For *n* AUs of a certain feature (see in the text or in Table 7.3 for value of *n* per facial feature) there are 454x*n* possible agreements / disagreements (*aa* / *da*).

	eyebrows	eyes	nose	mouth	chin	Σ:
aa	1331	1795	1351	8122	1797	14396
da	31	21	11	50	19	132
Σ:	1362	1816	1362	8172	1816	14528

The test of ISFER performance in facial action coding proceeded along the same lines as explained before: the human judgements of 454 test images were compared to those produced by the system. Overall results of this comparison are shown in Table 7.4. Yet, Table 7.4 does not show the actual recognition rates achieved by the system. The reason is that each of the test images could represent the activation of one or more AUs, which the system could recognise correctly, partially, or incorrectly. Table 7.5 summarises the system performance in facial action coding of 454 test dual-view images of faces given in the following terms:

- *Correct* denotes that the AU codes recognised by the system were completely identical to the AU codes scored by human observers judging the same images.
- *Partially correct* denotes that AU-coded description obtained by the system is similar but not identical to the one given by human observers when interpreting the same image (e.g. some AU codes may be missing or may be recognised in addition to those recognised by human observers).
- *Incorrect* denotes that none of the AU codes discerned by human observers in a given image were recognised by the system.
- *Recognition rate* has been calculated as the ratio between the number of correctly recognised test images and the total number of test images. If more than one AU of a particular feature was misrecognised in a test image, the pertinent image was counted once for the given feature. If several AUs of different features were misrecognised in a test image, that image was counted for each of the pertinent features. To calculate the percentage of agreement (i.e. the recognition rates), human FACS coders typically use the ratio between the number of correctly recognised AUs and the total number of AUs shown in the stimulus image being

judged. However, it is more appropriate to calculate the recognition rates based on the number of test images when one evaluates the performance of an automated system. This is because the system may score AUs which were not scored by human observers; such errors would not be taken into account if the recognition rates were measured based upon the number of correctly scored AUs and the total number of AUs shown in the analysed images.

As can be seen from Table 7.5, in 86% of 454 test cases ISFER coded the analysed facial expression using the same AU codes as the human observers. If we consider only the images in which the AUs were encoded with higher certainty factors CF (say CF>30; there are in total 423 such images), agreement between the system conclusions and human judgements of the same images is even 91%. When compared to the performances of other automated systems for facial action coding, similar recognition results are found but for much smaller sets of AUs to be recognised in smaller sets of test images (Tian et al. 2001). From other AU-recognition systems, the best performance has been achieved by the AFA system (Tian et al. 2001), which achieves average recognition rate of 88% when encoding 16 AUs and their combinations over 226 test samples.

Table 7.5
ISFER performance in facial action coding of 454 test dual-view facial images measured for AUs per facial feature, for upper- and lower-face AUs, and overall.

	upper-face AUs		lower-face AUs		
	eyebrows	eyes	nose	mouth	chin
Correct	433	437	443	423	436
Partially correct	21	17	10	28	17
Incorrect	0	0	1	3	1
Recognition rate	**95.4%**	**96.3%**	**97.6%**	**93.2%**	**96.0%**
Correct	422		413		
Partially correct	32		37		
Incorrect	0		4		
Recognition rate	**93.0%**		**91.0%**		
Correct	392				
Partially correct	58				
Incorrect	4				
Recognition rate	**86.3%**				

As far as misidentifications produced by ISFER are concerned, most of them arise from confusions between similar AUs (AU1 and AU2, AU6 and AU7, AU18 and AU35) and from subtle activations that remained unnoticed by human observers (e.g. AU26, AU38, and AU39). The reason for the confusion between AU1 and AU2 (i.e. recognising AU1 in addition to AU2, which was detected by human judges) is that activation of AU2, which raises the outer portion of the eyebrow(s), tends to pull the inner portion of the brow (AU1) as well. Although human observers also confuse AU6 and AU7 often (Tian et al. 2001), in the case of ISFER, the reason for the confusion between AU6 and AU7 are the utilised rules for recognition of these AUs. Namely, if AU12 is present, AU6 will be scored (see Table 5.5) although this does not necessarily match the actually shown expression (Figure 7.4). Similarly, the confusion between AU18 and AU35 is caused due to the utilised rules for encoding these AUs in facial images. Since inward pull of the cheeks is not detected by the system, only the width of the mouth distinguishes AU18 from AU35, causing misidentification of a weak AU35 (see Table 5.5). The reason for all mistaken identifications of AU26 and most of the mistaken identifications of AU38 and AU39 are the subtle activations of these AUs, which remained unnoticed by the human observers. Actually, in those cases, ISFER coded the input images correctly, unlike the human observers. Yet such cases were addressed as misidentification.

Therefore, comparing an automated system's performance to that of human judges is not enough. Human observers sometimes disagree in their judgements of facial actions pictured in an analysed image (e.g. Tables 7.1 and 7.2). They occasionally make mistakes and if the tested automated system does not produce the same mistakes, its performance measure is reduced. To estimate the performance of an automated system precisely, it is necessary to compare it to a validated standard. Behavioural scientists as well as few researchers in the field of automatic facial action coding use so-called *Gold Standard Faces* (GSF - Ekman and Friesen 1984, Friesen and Ekman 1987) as a benchmark for comparison. Nevertheless, though the photographed expressions of GSF are described in detail in terms of displayed facial actions, judgements of these expressions were made by human observers based merely upon the visual cues present in the judged photographs. A more accurate means for recognising facial activity such as measures of muscular electrical activity have not been used to double-check human visual judgements of GSF. Though it is understandable why muscular electrical activities were not measured for GSF (i.e. the subjects must be wired and that, in turn, results in visual occlusions of photographed facial expressions), the lack of such a double-check involves the risk that some subtle facial actions in GSF, invisible to the naked eye, remained undetected (e.g. AU26 present due to slightly parted teeth behind closed mouth). On the other hand (as in the case of ISFER and

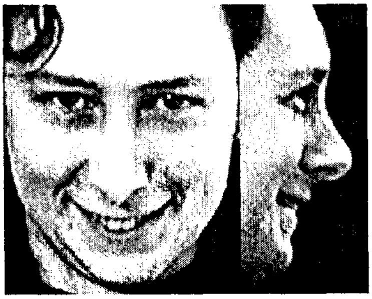


Figure 7.4: Facial expression of AU7+AU12 activation

AFA systems), due to their sensitivity to small differences in spatial samples of facial features, automated facial action coders may outperform human judges and detect subtle facial changes that were invisible to them. However, any such performance would be counted as a failure of the automated system evaluated according to a standard such as GSF. Though it is apparent that a better, readily accessible, standard set of facial images encoded in terms of displayed facial actions is necessary for measuring performances of AU recognition systems, no effort in establishing such a universally usable database of test images has yet been reported (see section 7.1). In consequence, the accuracy of recognition results of automated AU encoders cannot be measured or compared to each other. Therefore, the performance of any automated system for AU recognition including ISFER can only be estimated by comparing the system's conclusions about a set of non-validated facial images to sometimes erroneous human judgements of the same images.

Quantitative validation of the rule base: Quantification

If no standard set of FACS coded facial images can be used as ground truth (benchmark) for validating the performance of automated systems in AU recognition, one cannot expect a standard image database coded in terms of quantified AU codes that is readily accessible for validating automated tools for quantification of encoded AUs. Establishing such a facial database that could be used as benchmark for validating the performance of automated systems in quantified AU encoding is by no means a trivial task. In addition to the problem of detecting subtle facial actions that may be invisible to the naked eye of a human observer, there are a number of related issues (see also section 5.5). Since FACS only provides five different AUs which can be assigned an intensity on a 3-level scale, there are no standardised rules for displaying and scoring various AUs on a 100-level scale. Moreover, some AUs such as the blink (AU45) are either encountered or not, so they can only be scored on a 2-level scale (Table 5.9). But the crucial issue is that each person displays a particular AU with a different maximal intensity. In consequence, a standard facial-image database must contain not only images of faces described in terms of quantified AU codes, but also records containing maximal activation for each AU per subject. Because of the complexity of this task and the general lack of automated systems for AU recognition (Table 2.7) that could benefit from a standard database of FACS-coded facial expressions, no effort has been made to build such a standard resource for validating the performance of automated facial action coders.

Hence, the validation of ISFER's performance in quantified facial action coding from dual-view images of faces proceeded along the same lines as those explained before, namely, by comparing human judgements of test images to those produced by the system. Though this obtained merely an approximate estimation of the system's performance, it addressed the question of whether ISFER's conclusions about the intensity levels of scored AUs were acceptable to human beings judging the same images of faces.

Originally, the validation test for estimating ISFER's performance in quantification of encoded AUs was envisioned as to provide several human experts with the 454 FACS-coded images used in the previous validation test (see Table 7.5 in the preceding sub-section) and to compute the inter-observer agreement about the level of intensity of each AU shown in the test images. Yet two difficulties were encountered with this approach. First, for only 100 images, it took a single human observer over 5 hours to assign levels of intensity to the depicted AUs. Second, when this human observer was given the same images once again, the agreement between the first and the second judgement of those images was rather low. A possible reason for this is that the observer was only given the neutral expression image of the currently analysed subject for comparison; images of that subject's maximal displays of various AUs (i.e. his/her individual extreme displays set - IEDS; see section 5.5) were not provided. Therefore, another test procedure was defined:

- *Test images*: from the set of 454 FACS-coded dual-view images of faces used in the previous validation test, 91 images, which the system coded correctly in that test, were selected. The selection was made so that each of 24 AU codes, which ISFER can encode and quantify on the 100-level scale (see Table 5.9), was represented by at least 7 different images of this set. Overall, the set contained images of 4 different faces.
- *Questionnaire*: the employed questionnaires for eliciting intensities of face actions were made separately for each test image. The questionnaire for a given test image was divided into sections corresponding to the AUs scored in that image. For each section of the questionnaire, human judges were asked to indicate an appropriate intensity level on a scale from 1 to 10 by comparing the test image to the neutral facial expression and the IEDS set of the pertinent subject.
- *Procedure*: seven experts (i.e. certified FACS coders) were asked to fill out the questionnaires corresponding to 91 test images. The scores were averaged and multiplied by 10 (in order to obtain values that are comparable to the intensity levels that the system assigns to the encoded AU codes). The set of test images was divided into 4 partitions corresponding to four subjects. For each partition of the test set, the pertinent subject's IEDS was used to initialise the database of extreme model deformations (see section 5.5) and the performance of the system in quantifying encoded AUs was compared to the averaged human judgements of the same images.

Table 7.6 summarises (for each of the five prominent facial features, for upper- and lower-face features, and overall) the average disagreement between the conclusions of the human observers and those produced by ISFER about the intensity levels of AUs depicted in 91 test dual-view facial images. It is interesting to note that the smaller the size of the judged facial feature (in respect to the size of the whole face), the greater the disagreement between the judgements of the human experts and those of the system as to the depicted intensity level for AUs of that feature. There are two possible reasons. Since this finding applies just to the eyes and the eyebrows and not to the smallest of the facial features, the nostrils, one reason may be that human beings assign a higher priority to the upper-face features (as remarked by Ekman (1982) as well) while paying less attention to the actual degree of facial changes apparent in those features. Another, simplistic reason is that the human eye is rather insensitive to small changes in the upper-face features.

Table 7.6
ISFER performance in quantifying AU codes measured in terms of disagreement between the scores of the human experts and those of the system for 91 test dual-view facial images. Disagreement about the depicted intensity level of each AU of the test set is given for AUs of the five facial features, for the upper- and lower-face AUs, and overall.

	upper-face AUs		lower-face AUs		
	eyebrows	eyes	nose	mouth	chin
Average disagreement	**24.2%**	**27.7%**	**18.1%**	**17.0%**	**11.3%**
	26.0%		**15.5%**		
	20.8%				

Since ISFER is the only automated system for quantified AU recognition reported up to date (chapter 2 and/or Pantic and Rothkrantz 2000d), its performance in quantified facial-action coding could not be compared to the performance of some other system. In addition to the general lack of a standard image database for the face, this makes the evaluation of the generalisability of ISFER impossible.

7.3 Validation studies on the Facial Expression Classifier

The aim of the validation studies on the Facial Expression Classifier was to test the components of the dynamic memory of experiences and the pertinent recall (retrieval), learning (adaptation), and quantification functions (sections 6.4 and 6.5). A qualitative study was performed to test if the dynamic-memory clusters are accessed by their cases (i.e. the cases that were previously classified in the pertinent clusters) correctly, when these are used as input to the system. Quantitative studies on Facial Expression Classifier were carried out to test if the interpretation labels learned from the user and quantification of those, output by a trained ISFER, were acceptable to the pertinent user.

Qualitative validation of learning and recall

In this test, the learning and recall capabilities of the Facial Expression Classifier were investigated to ensure that input facial actions and the related interpretation label were learned and retrieved in a subsequent processing correctly. In other words, the aim was to ensure that the CBR architecture of the Facial Expression Classifier part of ISFER was implemented correctly.

Table 7.7
The interpretation categories defined by a human lay expert for 40 facial expressions of the database of training images

Expression	Interpretation	Expression	Interpretation
AU1	Disappointed	AU6+AU13	Ironic
AU2	Angry	AU15	“I don't know”
AU1+AU2	Surprised	AU15+AU17	“I don't know”
AU4	Angry	AU16+AU25	Angry
AU5	“Please don't”	AU17	“I don't know”
AU7	Thinking (problem)	AU18	Thinking (problem)
AU1+AU4+AU5+AU7	“Please don't”	AU19+AU26	Funny
AU1+AU4+AU5	“Please don't”	AU20	“I don't know”
AU1+AU4+AU7	Disappointed	AU23	Thinking (problem)
AU1+AU5+AU7	“Please don't”	AU24	Angry
AU1+AU4	Disappointed	AU24+AU17	Angry
AU1+AU5	“Please don't”	AU27	Surprised
AU1+AU7	Disappointed	AU28+AU26	Thinking (problem)
AU5+AU7	“Please don't”	AU28t+AU26	Thinking (problem)
AU8	Angry	AU28b+AU26	Thinking (problem)
AU9	“What a slimy thing”	AU29	Funny
AU9+AU17	“What a slimy thing”	AU35+AU26	Thinking (problem)
AU10	“What a slimy thing”	AU36t+AU26	Funny
AU10+AU17	“What a slimy thing”	AU36b+AU26	Thinking (problem)
AU6+AU12	Happy	AU41	Sleepy

To validate these functions, that is, to determine whether the "correct" cluster is selected for a known case, a "lay expert" (i.e. someone without formal training in emotion signals recognition) drawn from college scientific staff was asked to train the system by using the database of 40 training images (see Figure 6.3 and Table 6.1). Then, from the set of 454 FACS-coded dual-view facial images used to validated ISFER's performance in AU recognition (section 7.2), 11 images, which were correctly FACS-coded by the system in that test, were selected. The images corresponded to the 11 user-defined interpretation categories (Table 7.7): AU1+AU4 (Disappointed), AU4 (Angry), AU1+ AU2 (Surprised), AU1+AU4+AU5 ("Please don't"), AU7+AU23 (Thinking (problem)), AU9+AU26 ("What a slimy thing"), AU6+AU12+AU26 (Happy), AU6+AU13 (Ironic), AU15+AU17 ("I don't know"), AU19+AU26 (Funny), and AU41 (Sleepy). For all 11 test images, ISFER retrieved the correct interpretations. Thus, the implemented learning and retrieval (of known cases) appeared to be satisfactory.

Quantitative validation of recall

A more stringent validation test of recall capabilities of the Facial Expression Classifier addressed the question of whether ISFER returns the same interpretation labels learned from the user as the pertinent user when presented with an arbitrary set of dual-view facial images. This test was carried out with the same lay expert who trained the system in the previous test. In this test, we used the set of 392 dual-view facial images that were correctly FACS-coded by the system in the validation test used to measure ISFER's performance in AU recognition (section 7.2).

The questionnaire for eliciting affective/attitudinal states was divided into 12 sections corresponding to the neutral category and 11 categories defined by the lay expert while training the system (Table 7.7). Per stimulus image, the expert was asked to interpret the displayed facial expression by comparing it to the neutral facial expression of the relevant subject and then to fill out the questionnaire by selecting one of 12 affective/attitudinal categories. The lay expert labelled 27 images inconsistently, using different labels than when training the system. Hence, the recall function of the system was evaluated using the other (392 - 27 =) 365 dual-view images of faces.

Table 7.8 provides a summary of the interpretation results achieved by the system for 365 test images for each of the interpretation labels learned from the lay expert. The *Samples* column denotes the number of test images that the expert classified in the pertinent interpretation category. The numbers in parentheses represent the number of test images that the expert labelled differently than when training the system. *Correct* denotes that the interpretation produced by the system was identical to that given by the lay expert. *P(artially) Correct* denotes that the conclusion reached by the system contains one or more interpretation labels in addition to the one given by the expert. *Incorrect* denotes that the system generated none of the interpretation labels given by the expert for the pertinent sample image. *Recognition rate* denotes the ratio between the number of correctly interpreted test images and the total number of images used for testing.

On average for 74.2% of the test cases, ISFER correctly interpreted the input facial expressions in terms of interpretation labels defined by the lay expert (Table 7.8). Given that human observers, having a formal training in emotion signals recognition, detect six basic emotional facial expressions with an accuracy ranging from 70% to 98% (Bassili 1978), it is significant that ISFER matched this accuracy when detecting 12 facial affects defined by a human expert and after being trained on only 40 cases. Most of the system's misidentifications are due to the utilised testing procedure. Namely, the lay expert was asked to assign a single interpretation label to each and every sample facial expression, while ISFER resulted in multiple interpretation labels for some of the sample expressions (see the numbers in the *P(artially) Correct* column of Table 7.8). Hence, a more precise estimation of ISFER's recall capability evaluated in this test, that is, an indication whether the "correct" cluster of the dynamic memory is selected for an unknown case, is the ratio between *Correct + P(artially) Correct* interpreted images and the total number of images used for testing. Per interpretation category defined by the lay expert this ratio is 94.8% on average: Neutral (94.4%), Disappointed (86.2%), Angry (92.6%), Surprised (100%), "Please don't" (90.9%), Thinking (problem) (97.6%), "What a slimy thing" (100%), Happy (100%), Ironic (83.3%), "I don't know" (90.9%), Funny (94.7%), and Sleepy (100%). If we bear in mind that the lay expert interpreted virtually each of the facial expressions that was incorrectly classified by the system differently than she did for its component expressions in the training phase, it can be concluded that ISFER's capability (in retrieving correct interpretations for known cases) is satisfactory.

Table 7.8
ISFER's recall capability measured for 365 test images for each interpretation category defined by the lay expert

	Samples	*Correct*	*P. Correct*	*Incorrect*	***Recognition Rate***
Neutral	36	34	0	2	94.4%
Disappointed	29 (4)	16	9	4	55.2%
Angry	27 (7)	17	8	2	63.0%
Surprised	46 (1)	32	14	0	69.6%
Please don't	33	23	7	3	69.7%
Thinking (problem)	41 (8)	33	7	1	80.5%
What a slimy thing	38	33	5	0	86.8%
Happy	40	31	9	0	77.5%
Ironic	18 (2)	11	4	3	61.1%
I don't know	33 (5)	22	8	3	66.7%
Funny	19	14	4	1	73.7%
Sleepy	5	5	0	0	100%
Total:	**365 (27)**	**271**	**75**	**19**	**74.2%**

Quantitative validation of learning and quantification

The learning (adaptation) function of the Facial Expression Classifier part of ISFER was tackled next. The addressed question was: How acceptable are the interpretations given by ISFER, after the system is trained on a larger number of cases? The following procedure was designed to discover this.

The same lay expert used to train the system in the previous tests was used for this test as well. The same set of 392 dual-view facial images that the system FACS-coded correctly in the test aimed at measuring the performance of the system in AU recognition (section 7.2) was used once more. The set was divided into two sets of images: a training set (196 images) and a test set (196 images). The 27 images that were excluded from the previous test due to the expert's inconsistent labelling of these (Table 7.8) were included in the training set of images. The images that were interpreted by ISFER as *Partially Correct* and *Incorrect* in the previous test were divided between the training and test set of images. For the training set of dual-view images, the lay expert was asked to trigger the learn mode of the Facial Expression Classifier whenever the interpretation produced by the system was not satisfactory. In addition to the 11 categories defined when training the system on the database of training images (Table 7.7), the lay expert defined another three interpretation categories: bored, monkey face, and delighted. Then, the lay expert judged the acceptability of the interpretations returned by the system over the test set of images (Table 7.9). *Agree* denotes that the expert approved of the interpretations achieved by the system. *Partially Agree* denotes that the expert approved of at least half of the generated interpretation labels (e.g. Figure 7.5). *Disagree* denotes that the expert disapproved of more than half of the interpretation labels scored by the system (e.g. Figure 7.5). *Recognition rate* denotes the ratio between the number of test images the generated interpretations of which were approved of in some measure (*Agree* and *Partially Agree*) and the total number of images used for testing.

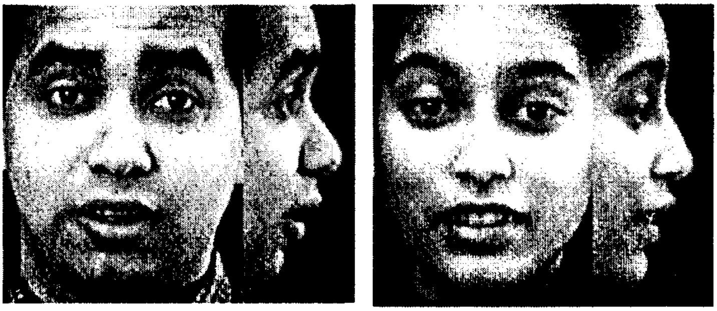


Figure 7.5: Examples of images for which the lay expert partially agreed (left-hand-side image) and disagreed (right-hand-side image) with the interpretation given by the system (i.e. for the left-hand-side image: Surprised + Thinking (problem); for the right-hand-side image: Delighted).

As shown in Table 7.9, in 97% of 196 test cases the lay expert approved of the interpretations (affective/attitudinal labels learned from that expert) generated by the system to some degree. Similar recognition results were also reported for other automated facial affect analysers (see Tables 2.8 and 2.9) but for much smaller sets of affective states to be recognised (i.e. the set of six basic emotions or a subset of it). This finding and the fact that ISFER represents a user-adaptive system that enhances its expertise with each presented case led to the conclusion that the ISFER's performance in interpreting facial expressions in terms of user-defined affective/attitudinal labels is acceptable and will become even more satisfactory as the longer the system is used.

Table 7.9
Approbation of interpretation labels generated by the system for 196 test images

	Agree	*Partially Agree*	*Disagree*
# samples	163	27	6
Rate	83.2%	13.8%	3.1%
Recognition rate	**96.9%**		

Finally, the quantification function of the Facial Expression Classifier part of ISFER was tackled. This test addressed the question of whether the intensity levels of the scored interpretation labels that ISFER had assigned were acceptable to the lay expert (the same used for the previous tests) judging the same images of faces. To this end, the lay expert was asked to evaluate the acceptability of the intensity levels assigned to the interpretation labels returned by the system for 163 test images which she approved of in the previous test (see Table 7.9). In summary, the lay expert approved of 81% (i.e. 198) of the intensity levels assigned by ISFER to the scored 244 interpretation labels. In other words, in 81% of the cases the expert agreed for ±20% with the intensity level assigned by the system to a scored interpretation label. Since the quantification of interpretation labels relies on the quantification of the scored AUs (section 6.5), this 20% were meant to account for the average disagreement between human judgements and those produced by ISFER about the shown AU-intensity level (Table 7.6).

Most of the disagreements concerned the interpretation of facial expressions affecting the upper-face features (i.e. eyes and eyebrows). Namely, the lay expert argued that the intensity levels of shown affective/attitudinal states should be higher than those assigned by the system. Something similar applied to the quantification of scored AU codes: the AU-intensity levels assigned by the human experts and those assigned by the system disagreed more when these concerned the upper-face features than the lower-face features (Table 7.6). These findings clearly indicate that humans assign a higher priority to the upper-face features than to the lower-face features when interpreting facial expressions (remarked by Ekman (1982) as well). However, to confirm these findings and implement them into the ISFER's reasoning process, more extensive field trials and more elaborate quantitative studies on the issue are necessary.

7.4 Discussion on system validation

The basis of the validation studies on ISFER was a comparison between the facial expression analyses carried out by the system and by humans of the same dual-view images of faces. The criterion involved in this approach is the degree to which human observers agree among themselves upon the AUs shown in an analysed image. After all, one cannot expect that the results of ISFER, or those of any other automated facial expression analyser, agree with those of humans to a greater extent. The preceding sections proved this in an experimental way. The agreement among human judges about the AU codes shown in 560 facial images was 81% (i.e. 454 images, Table 7.3, section 7.2). For 86% (392 images) of these cases ISFER obtained AU-coded descriptions of analysed facial expressions that were identical (Table 7.5). The average disagreement between the AU-intensity levels generated by ISFER and those given by human judges, who evaluated 392 images that were correctly FACS coded by the system, was 20% (Table 7.6). For 97% of test cases (196 images correctly FACS coded by ISFER), the system generated the interpretations (affective/attitudinal labels learned from one lay expert) that were approved to a certain extent by the lay expert (Table 7.9). Finally, in 81% of the cases, the lay expert agreed for ±20% with the intensity levels assigned by the system to the interpretation labels previously approved by that lay expert. Hence, human experts are expected to agree for ±20% with all of the conclusions produced by ISFER in merely 67.8% of the test cases[1].

Given that the validation is only as sound as the abilities of the human experts involved and that it turned out that the consensus of the involved experts is only moderate, assessing the performance of ISFER by comparing only the results of ISFER to those of human judges is not enough. Two reasons may be given why the involved experts varied somewhat in their interpretations of the facial actions present in the judged images and why inconsistencies were observed in human judgments of signalled affective states in the same images. Firstly, the recognition of facial actions is maybe a task that commonly yields different results when executed by humans (in spite of the fact that the originators of FACS theory (Ekman and Friesen 1978) claim otherwise) while the facial affect analysis is perhaps a task that always yields inconsistencies, even between the judgements made by a single human observer. The other reason may be that the experts used for validating the performance of ISFER happened to be a discordant group of persons who judged the stimulus image while being insufficiently concentrated on that task. However,

[1] From 454 test images, 392 images (86.3% of 454 images) would be correctly FACS coded, 380 images (96.9% of 392 images) would be correctly FACS coded and interpreted in terms of learned affective/ attitudinal states, and 308 images (81% of 380 images), that is, 67.8% of original 454 images, would be correctly interpreted in terms of quantified AU codes and multiple quantified user-defined interpretation labels.

resolving the actual cause of the only moderate consensus of the used experts is intractable since there is no normative test for examining the human capability in analysing facial expressions. In consequence, it can be only concluded that human observers *may* disagree in their interpretations of the facial actions shown in the judged images (e.g. see Tables 7.1 and 7.2) and that they *may* be inconsistent when judging facial expressions in terms of signalled affective/ attitudinal states (section 7.3). Hence, if human judges make mistakes occasionally and if the applied approach to assessing the system's performance is to compare ISFER's results to those of human judges then, if ISFER does not produce the same mistakes as humans do, its measure of performance is reduced.

To obtain a more precise estimation of ISFER's performance, it is necessary to compare it to a validated standard. However, as already explained in sections 7.1 and 7.2, there is no readily accessible standard set of facial images coded in terms of displayed facial actions that could be exploited for measuring the performance of AU recognition systems. Though many researchers in the field pointed out the need for a standard database of facial information, no effort in establishing such a universally usable benchmark for efforts in automating FACS coding has yet been reported. Since ISFER is the only automated system that analyses facial expressions in terms of multiple quantified AU codes and multiple quantified user-defined interpretation labels reported up to date, one cannot expect a standard image database readily accessible for validating automated systems for recognition of quantified AU codes and multiple quantified user-defined interpretation labels. In consequence, a precise estimation of ISFER's performance cannot be obtained. A measure of its performance can be established only approximately by comparing its conclusions generated for a set of non-validated facial images to sometimes erroneous human judgements of these images. Exactly such an approximate measurement of ISFER's performance was presented in this chapter.

Based upon the validation studies explained in section 7.2, it can be concluded that ISFER's performance in quantified facial action encoding from dual-view images of faces exemplifies an acceptable level of expertise. At least as far as ISFER's performance in AU recognition is concerned, the achieved results are similar to those reported for other automated FACS coders. ISFER achieved an average recognition rate of 86.3% by encoding 32 AU codes and their combinations in 446 test samples, while other automated FACS coders have (in the best case) an average recognition rate of 88% by encoding 16 AU codes and their combinations in 226 test samples (Tian et al. 2001). Since ISFER is the only automated system that quantifies the encoded AU codes reported up to date (Pantic and Rothkrantz 2000d), its pertinent performance could not be compared to the performance of some other system. As far as the human experts used to validate the performance of the system are concerned, the performance of the ISFER in scoring AUs intensity levels with an average error of 20% seemed acceptable to them.

Based upon the validation studies explained in section 7.3, it can be concluded that ISFER is capable of mapping the scored AU codes to quantified interpretation labels learned from the user. Since ISFER is the only automated system that analyses facial affect in terms of multiple quantified interpretation labels learned from the user reported up to date (Pantic and Rothkrantz 2001a), ISFER is not comparable with other automated facial affect analysers. If we make such a comparison notwithstanding, then it can be concluded that ISFER performs at least as good as any other facial affect analyser. ISFER achieved an acceptable recognition of 15 user-defined affective/attitudinal states and their combinations in 96.9% of 196 test samples, while other automated facial affect analysers have (in the best case) an average recognition rate of 98% for 4 basic emotions over 30 test samples (Table 2.9). Since ISFER is a user-adaptive system that learns its expertise incrementally from the user with whom it interacts during daily use, once ISFER is actually deployed, it can be expected that its performance will also increase incrementally to become satisfactory for the user.

Though rather acceptable, ISFER's performance can be improved in several aspects. Most of these are discussed in detail in sections 4.4, 5.7, and 6.6, and then summarised in section 8.4. Therefore, merely a brief discussion of those issues is provided here:

- ISFER cannot handle distractions like occlusions (e.g. by a hand), glasses, and facial hair. Hence, its analysis is limited to non-occluded faces without a beard, moustache, and glasses. Also, ISFER cannot deal with rigid head movements; the analysed images have to be scale and orientation invariant with respect to the image of the expressionless face of the currently observed subject, as if they were acquired by the head-mounted-camera device illustrated in Figure 4.2. Otherwise, the performed reasoning will be erroneous as the evaluation of the input data certainty done by the Facial Action Encoder is based on a comparison of the immovable facial points to the pertinent points extracted from the expressionless face of the observed subject (section 5.4). Finally, ISFER cannot encode the full range of facial behaviour (i.e. of all 44 FACS AUs); it performs facial action coding in static dual-view facial images in terms of 29 AUs (i.e. 32 AU codes, Table 5.8). These limitations can be handled, at least partially, by accommodating the analysis of facial image sequences rather than the analysis of static images of faces (see also section 8.5).
- From the validation studies on ISFER, it is apparent that humans assign a higher priority to the upper-face features than to the lower-face features when interpreting facial expressions (remarked by Ekman (1982) as well). However, to confirm these findings and to implement them into the ISFER's reasoning process, more extensive field trials and more elaborate quantitative studies on the issue are necessary.
- ISFER adopts an event approach: facial expressions of affective states are treated as context-free autobiographical events. This is a very constrained type of event where the context is limited to the accompanying facial actions displayed by a particular subject in a time instance. Nevertheless, the interpretation of a

monitored subject's facial behaviour does not only depend on that subject as far as his/her maximal displays of particular facial actions and his/her typical facial expressions are concerned. It also depends on the subject's typical temporal course of facial behaviour given the environmental constraints in which the subject acts. Due to the complexity of this problem, handling ISFER's context insensitivity is probably the most significant challenge facing future developers of ISFER.

7.5 Discussion on system usability

Looking at software engineering from a historical perspective, in the 1960s, information technology reached a level sufficient to meet institutional needs and began to link software with daily operations of institutions. In the 1970s and 1980s, hardware costs began and continued to decline so that information technology became quite customary in a wide range of institutions and low-cost applications became widely implemented. In the 1990s, the era of ubiquitous computing began (Schneiderman 1995, Pentland 2000), casting a new light on the future of information technology and setting new requirements for software products. Nowadays, by the ever-increasing dependence of society on computing (computing devices are already almost everywhere – in offices, homes, cars, and even clothes), the main demand for software products is not high productivity but rather high quality. In this era, quality is no longer merely an advantage in the market; it has become a necessary condition.

Speaking of market terms, the users' satisfaction plays a crucial role in the evolution of information technology. A software product's *fitness for use*, that is, whether the product meets the users' requirements in a satisfactory manner (Juran and Gryna 1970), determines the product's survival at the marketplace. In general, nevertheless, computing technology has not reached the ultimate goal of being satisfactorily usable and universally accessible. As reported by Schneiderman (2000, 2001), an average user wastes approximately 5.1 hours per week while trying to use computers. In consequence, a common experience of computer users is dissatisfaction. In order to change this and to achieve the goal of having satisfactorily usable and universally accessible software products, multiple human-computer interface design breakthroughs are necessary (chapter 8), but thorough usability (user satisfaction) tests are essential.

Standards and guidelines for testing the usability of software products (i.e. to measure user satisfaction) are varied and numerous (Kan 1995). For example, for products developed by IBM, customer satisfaction is measured in terms of CUPRIMDSO criteria (capability, usability, performance, reliability, installability, maintainability, documentation, service, and overall), while for Hewlett-Packard products, FURPS criteria are used (functionality, usability, reliability, performance, and service). Either of those two sets or any other similar set of criteria for measuring user satisfaction can be used for the examination of user satisfaction with ISFER. However, a precise estimation of users' satisfaction with ISFER cannot be obtained at this point since ISFER is not actually deployed yet. What can be done, nevertheless, is to establish an approximate estimation of future user satisfaction with ISFER. If that estimation indicates that future users of the system won't be satisfied, the system has to be improved before it is released.

The rest of this section deals with assessing future user satisfaction with ISFER based upon the validation studies discussed in the preceding sections and the FURPS criteria (except for serviceability, which cannot be assessed since the system has not been actually deployed yet).

Functionality

Simply speaking, the functions of a system have to meet the expectations of users and to be correct if users are to be satisfied about the functionality of the system. The functional requirements imposed on ISFER are (section 2.6):

- Automatic quantified FACS coding in static facial images: the processing should start with a generic (subject-independent) facial data extraction in input facial images, proceed with a generic classification of extracted data into multiple AU categories, and end with adapting to the currently monitored individual in order to obtain subject-dependent quantification of the encoded AU codes.
- Automatic facial affect analysis from input images in terms of multiple quantified facial expression interpretation labels learned from the user.

Since ISFER's analysis of an input facial image results in multiple quantified AU codes and multiple quantified user-defined interpretation labels describing the facial expression captured in the input image, the system does what is expected. But the question whether the generated description of the input facial expression is correct is more difficult to answer. Namely, if "correct" denotes that the description of an analysed facial expression generated by the system is identical to that given by the user when analysing the same expression, then based on the validation studies on ISFER (sections 7.2 and 7.3) one could conclude that the performance of ISFER is not correct. In consequence (if "correct" means "identical"), one could expect that future users of ISFER will not be satisfied with its functionality. However, if "correct" denotes that the description of an analysed facial expression produced by the system is acceptable to the user and most of the time identical to that obtained by him/her when judging the same expression, then based upon the validations studies on ISFER, it can be concluded that future users will be satisfied with the functionality of ISFER. To improve future user satisfaction with the functionality of ISFER, the aspects of the system performance listed above (section 7.4) should be improved.

Usability

If a system fits the intended user audience, their tasks, and their environment, the users will be satisfied about the system usability. There are four fundamental questions that can act as useful guides in usability testing (Schneiderman 1993):

1. Are the individual differences between users considered?
2. Are the social implications of the intended system on the user audience considered?
3. Did users participate in the actual design process?
4. How does the intended system empower users?

The main goal for the development of ISFER was to achieve a fully automatic facial expression analysis, which is applicable to automated FACS coding and automated facial expression classification in observer-defined interpretation categories, such that it serves the purposes of behavioural science investigations of the face (sections 1.1 and 2.6). Hence, the intended user audience concerns behavioural scientists who, in general, have little or no experience with computing and may interpret the same facial expressions in terms of different affective labels. Since in order to use ISFER, users need no specific knowledge of computing and quite a simple training, it is expected that the system will be easy to use by users having little or no computing skills (see also sections 4.2, 5.2, and 6.3). As explained in chapter 6, the Facial Expression Classifier part of ISFER has been developed so that it learns its expertise by interacting with the user and performs facial expression interpretation in terms of the user-defined affective/attitudinal labels. Besides the fact that ISFER is a Java-implemented tool (i.e. a portable, platform-independent application), this makes ISFER fit for the intended user audience, their tasks, and their environment.

Yet the system is not universally usable since its usage is constrained by the kind of input images. One may think that the problem is that the system was not developed for analysing facial image sequences. However, since each frame of a video sequence can be analysed as a static image, ISFER is generally capable of analysing facial expressions in image sequences of faces. The problem is that if ISFER is to work correctly and achieve the peak performance, the input images must be acquired under the same viewing conditions as the neutral facial expression of the currently monitored person. In other words, a neutral facial expression of the observed subject has to be available (see the reasoning of the Facial Action Encoder explained in chapter 5) and the analysed images have to be scale and orientation invariant with respect to the image of the expressionless face, as if they were acquired by the mounted camera device illustrated in Figure 4.2.

As far as the social implications of ISFER for its users are considered, there are two relevant issues: invasion of privacy and displacement of human experts. The general goal of automating facial expression analysis is to redesign user interfaces to computers so that the machines become aware of the people that interact with them and respond in a better way based upon the observed "mood", which might be caused by the experienced interaction with computers (see also chapter 8). Yet computing technology is still perceived as threatening, anxiety-producing, and cold by many people, who would be terrified by the vision of computers being able to monitor their affective states. Especially if one bears in mind that such perceptually aware machines could be tightly networked together, affect-sensitive monitoring could be seen as the realisation of Orwell's dark vision of a government that can monitor and control your every move (Orwell 1949). This is a very serious issue that can lead to the disuse of software like ISFER due to users' unwillingness to use such programs and even, at an extreme, to outlawing of computers using cameras for monitoring human behaviour. Hence, privacy issues should be taken very seriously. However, it is important to note that it is not the cameras and the reasoning about human behaviour that are the problem here, but the networking that makes it relatively easy for outsiders to monitor people's behaviour. This suggests a method for addressing the privacy problem in the case of ISFER: to avoid, simply, concentration of the information. So, instead of collecting images of the observed faces and the associated interpretations (which in a short time can exceed the limits of the memory capacity as well), just the information that will improve the system's performance in the future can be kept: newly learned interpretation labels and typical facial expressions and extreme displays of the observed subjects.

A potential displacement of the human experts due to the employment of ISFER is not an issue actually. ISFER is developed as a tool that can help behavioural scientists: it automates much of their tedious and time-consuming tasks involving FACS scoring and stores their professional opinions about facial expressions of human affective/attitudinal states in an easy to retrieve way. Hence, ISFER empowers its users by performing a time-consuming and boring part of their jobs, freeing their valuable time to make the most difficult judgements on complex phenomena of human behaviour.

ISFER has not been developed in collaboration with actual (future) users. Hence, it is not known how the users will react to the system and if its design will be considered satisfactory. Nevertheless, since the system is very easy to use, it is expected that there will be no major complains about its design. In addition, the usability of the system can be improved by providing help files on the system's usage, which will also free the developers of ISFER from training the future users.

Reliability

Reliability, as defined in the discipline of quality engineering (Kan 1995), refers to the consistency of the results acquired using the same input data on the same part of software product or the product as a whole. Hence, the reliability of a system can be expressed in terms of deviation between the results obtained in repeated trials. In this sense, ISFER is reliable: for the same image, it will produce the same conclusions (at least if it has not been trained further in between the repeated trials and the previous trial has not been counted for calculating the typicality of the pertinent facial expression). However, if the reliability is measured in terms of deviation between the results generated by the system and those given by human observer judging the same image, then ISFER is not 100% reliable. Yet, as explained in sections 7.2, 7.3 and 7.4, the performance of ISFER has been perceived as satisfactory by human experts used in the validation studies on ISFER.

Performance

In quality engineering terms, the system's performance is as good as it approximates the performance of a system that provides the computational solution for the given problem in the shortest time, using the least memory space, and requiring the smallest number of processors (Kan 1995). Since ISFER is the only automated system that performs facial expression analysis from dual-view static facial images in terms of multiple quantified AU codes and multiple quantified interpretation labels learned from the user proposed in the literature up to date, there is no system with which ISFER can be compared. Hence, it is difficult to define ISFER's performance according to the definition given in the discipline of quality engineering.

However, what is meant by the term "performance" by the majority of people in information and computing technology is the processing speed of the system (capability), the accuracy in performing the required tasks (functionality), and whether its usage is satisfactory (usability). From this point of view and based on the discussions provided in this section and the validation studies explained in sections 7.2 and 7.3, ISFER's performance is satisfactory for the purposes of behavioural science investigators. Yet to confirm this finding by measuring the performance of ISFER in terms of methods and metrics commonly used by behavioural science investigators (which differs from those used to evaluate the performance of an automated system as explained in section 7.2), more extensive trials and more elaborate quantitative validation studies on ISFER are required. These concern mainly the evaluation of the performance of the Facial Action Encoder part of the system according the validation studies used by psychologists (i.e. collecting / acquiring stimulus images that are objectively FACS-coded[2], computing the number of agreements and disagreements about AUs displayed in these images for all possible pairs involving human observers and ISFER, and calculating the significance of these findings by taking into account chances for agreements at random). On the other hand, the execution of the system code is rather time consuming yet. This makes the system unsuitable for the purposes of perceptual human-computer interfaces where real-time performance is necessary since the delays make the interaction desynchronised and unnatural (see also section 2.2 and the discussion in chapter 8). As mentioned above, an improvement of the aspects

[2] As explained in sections 7.1 and 7.2, acquiring the images that are objectively FACS-coded is a difficult problem in its own right.

discussed in section 7.4 will improve ISFER's performance and, in turn, the prospect of future user satisfaction with the system.

8 HCI: The future

Multimedia is an art-world term, often credited to designers Charles and Ray Eames, that describes the fusion of media such as painting, sculpture, photography, music and video. Within the world of computers it is used broadly to describe almost any combination of media, ranging from simple text through to the Eames' vision. This diversity raises questions about the future of multimedia interfaces. In the natural world such diversity is elegantly explained by Darwin's theory of evolution through survival of the fittest.

(Preece and Schneiderman 1995)

Predicting the future of human-computer interfaces (HCIs) is a difficult task, but one important source of help is the accumulated information about the preferences and limitations of humans interacting with computers. Principles can be drawn upon, which may explain why some interfaces survive and others become extinct. Of course, in the case of any technology, market forces determine eventually which novel designs survive. Yet, since profit is the main drive behind all market decisions, clients' preferences have crucial relevance in the evolution of technology. As far as the interfaces are concerned, rigid designs that do not allow users to undo their actions, do not protect against errors, provide help at all times except at the right moment, and all in all make users frustrated, are likely to become quickly extinct due to their poor usability (Nielsen 1995). On the other hand, designs that include adequate attention to individual differences among users, support for a wide range of hardware/software and network access, design for reliability and safety, provision of access to the elderly or handicapped, and appropriate user controlled adaptation, are the kind of HCI designs that are likely to become the trend in computing technology (Schneiderman 2000). To elaborate, as computers become ever more ubiquitous in society (they organise our affairs, help us to work and express our ideas, find information and services we need, help us drive a car, entertain us), reliable multi-modal HCIs that are able to adjust to both individual and environmental differences are the future of man-machine interfaces (Marsic et al. 2000, Pentland 2000).

The key idea behind pursuing such a fully adaptive multi-modal HCI is facilitating natural, human-like, man-machine interaction. People communicate by using a variety of modalities such as sight, sound, and (optionally) touch, and by displaying a wide range of communicative signals such as spoken words, facial expressions, gestures, and vocal intonations, which may (and usually do) vary from situation to situation (environment) as well as from person to person. Therefore, the HCI systems that can interpret and emulate this variety of human communicative signals and account for contextual differences (i.e. who the user is, where he is, what he is doing, and how he is feeling) promise flexibilities and functionalities that transcend the traditional mouse and keyboard.

In general, this chapter deals with natural multi-modal HCI systems that are widely thought to become the "fourth generation" of computing and information technology (Waibel et al. 1995, Nakatsu 1998, Coen 1999, Marsic et al. 2000, Pentland 2000, Clarkson et al. 2000, etc.). In particular, this chapter examines the state of the art in the automation of multi-modal affect-sensitive monitoring, which is a prerequisite for the development of a natural multi-modal HCI of the kind described above. The first section of this chapter renders a brief history of human-computer interfaces. It is meant to serve as a guide to determining recommendations for the design and development of a new generation of HCI systems based upon the preferences and limitations of humans interacting with computers. Section 8.2 summarises research directions that could lead to very exciting improvements of man-machine interfaces. The rest of this chapter is concerned with the topic of automatic, multi-modal, affect-sensitive monitoring, that is, with enabling computers to detect, identify and understand how the user is feeling based upon his/her communicative cues sensed by the computer. Section 8.3 provides the taxonomy of the pertinent problem domain. The issues addressed in this section are: which modalities should be accommodated by an automated affect-sensitive monitoring tool to perceive which interactive signals, should the signals observed by various modalities be analysed in an isolated way or integrated at a fundamental stage of the tool's processing, is the time scale of sensed data important and how can it be employed in the intended tool's processing, etc. The discussion will imply that the sensing, detection and interpretation of facial and vocal non-verbal human communicative signals is essential for the realisation of a sophisticated affect-sensitive monitoring tool. Section 8.4 examines the state of the art in the automation of human affect analysis. It enumerates the advantages and limitations of ISFER, which is an automated system for human facial-affect analysis, and surveys the past work done in affect-sensitive monitoring of human vocal reactions. Based upon this overview of the current state of the art, section 8.5 summarises the challenges and opportunities facing researchers in the field of automated affect-sensitive monitoring of human interactive cues. Finally, section 8.6 discusses the impact that the proposed affect-sensitive monitoring tools (forming a front-end for future HCI systems described above) could have on the potential users and the society in general. This is a serious issue since it determines the uses, usefulness, and trustworthiness – essentially, the survival – of the proposed affective HCI systems.

8.1 The evolution of human-computer interfaces

Around 1980, at the dawn of the personal computer age, many chaotic and rigid user interfaces were produced that turned the users into frustrated victims of machines they could not control. Typical examples of useless interfaces at that time could display a five-minute video without a stop button and generate choice sequences that could not be reversed or cancelled. As high-resolution displays and fast chips emerged, video and audio processing as well as animations flourished (particularly for video games), introducing increased interface complexity accompanied by the users' need of better and more direct ways for controlling the wide range of possible operations. This gave rise to a new generation of user interfaces, in which direct manipulation became the dominant form of interacting and WYSIWYG (what you see is what you get) became a guiding principle. The aim was: (i) to make operations visible, incremental, rapidly manageable by means of a keyboard, and reversible, as well as (ii) to prevent user errors by effective designs. During the late 80s and early 90s, direct-manipulation interfaces were enhanced with embedded menus in text and graphics, mice, and various joysticks as the devices of choice. The relatively recent emergence of high-precision touch screens marked another enhancement of direct-manipulation interfaces.

As remarked by Preece and Schneiderman (1995) and Pentland (2000), the mid 90s can be viewed as the dawn of ubiquitous computing that shed a new light on the future of computing and gave rise to novel requirements that useful user-interfaces should fulfil. The growing availability of World Wide Web access with embedded menus providing links across the world led to an unusually rapid growth of Web servers and applications. First, as the emphasis has been on surfing the Internet, many WD (Web Development) firms emerged creating various Internet search engines, Web browsers and applications for data mining and retrieval. Then, the emphasis shifted to electronic text-based communication (such as e-mail and chat facilities), to tools for the development of Web pages and, eventually, to e-commerce. This produced novel possibilities for "doing business" and, as the number of users having on-line access grew steadily, an exploding number of WD firms competed for survival. The necessity of delivering new products in an ever-decreasing time frame affected, consequently, the quality of the issued products and interfaces. This Internet hype also blurred the essence of some paradigms, such as software agents since an increasing number of vaguely related applications needed legitimacy and sought it under the umbrella of the "agents"(section 3.7).

Though it was unfortunate at one hand and accompanied by numerous shoddy Web-oriented applications, on the other hand this Internet hype initiated rapidly accelerating progress in facilitating accessibility, speed, and reduction of error and failure rates. Moreover, it changed our view on computing and commerce (Shoham 1999). Above all, it clearly forecasted the type of working environments and information-communication spaces we are about to use in our everyday activities. Even nowadays a steadily growing minority of people exploits computers for work and use the Internet to communicate with each other, to shop, to seek out new information, and to entertain themselves (e.g. role-playing games allow people to become part of an interesting story as heroes in virtual worlds). This clearly indicates that in the future, with the aid of computers, we will carry out our daily tasks (think about the abundance of computers and intranets in offices and the popularity of video conferences, cars' on-board computers, remote education, e-commerce, etc.), we'll communicate and entertain ourselves in cyberspace across distance, cultures and time. Of course, the specifics of such virtual cyber worlds and of pertinent interfaces, which should facilitate easy and natural communication within those worlds and with the variety of embedded computing devices, are far from settled. Yet, it is clear that before this new generation of ubiquitous computing can be widely deployed, the users should experience it as being universally usable (i.e. having satisfactory performance and being universally accessible).

8.2 Rethinking human-computer interaction

The designers of older technologies such as postal services, telephones, and television, have reached the ultimate goal of having products and services that are universally usable, but developers of computing technology cannot claim the same. Schneiderman (2000, 2001) reports an average of 5.1 hours per week wasted by the users while trying to use computers. Consequently, despite visible progress in accessibility, increase of speed, and reduction of the error and failure rates, the primary experience of many computer users is dissatisfaction or even frustration. Common problems include incompatibility (e.g. of file formats, applications' versions, screen sizes) and low speed (e.g. due to varying network bandwidths and processor speeds). Although these issues are not of the least importance, the crucial problem, which is primarily responsible for the users' dissatisfaction, is incomprehensibility of many currently available software packages and Internet services. Confusing menu choices, disorganised structures of windows, shoddy engines for information mining and retrieval, incomprehensible error massages, and usually unnatural rigid (non-adaptive) interaction, are troubling to novices as well as to more experienced users and impose a significant barrier to elderly users and users with disabilities. Obviously, interface design breakthroughs are necessary if computing technology is to achieve the ultimate goal of being universally usable.

If we take into account the above state of the art in human-computer interfaces as well as the prediction that embedded computing devices will be ubiquitously present in society in near future (Nakatsu 1998, Pentland 2000, etc.), the key challenges facing researchers in the area of HCIs and related fields can be summarised as follows:

- supporting the technological variety,
- enhancing comprehensiveness of the graphical design,
- allowing natural multi-modal interaction,
- developing context-sensitive HCI systems, and
- facilitating (optionally) anthropomorphic response.

Although this list may not be complete, it summarises important issues that are rather insufficiently addressed by the current initiatives. Research devoted to these challenges will have a broad range of benefits for novel, intermittent, and frequent users by enhancing the usability of HCI systems. Hence, at the very least, these issues (explained in more detail in the rest of this section) are among the most exiting and economically important topics in HCI-systems research and in information technology in general (Sharma et al 1998, Pentland 2000, Schneiderman 2001).

Support of technology variety

In order to address the problem of incompatibility, the designers of HCI should deal with the change in pace of technology and the variety of equipment that users employ. According to Moore's Law, processor speeds double every 18 months. Since many users do not change the configuration of their computers at the same pace, this means that there are at least hundreds of different processor speeds currently in use. In turn, HCI designers who wish to take advantage of new technologies risk excluding users with older machines. Improvements of other hardware components such as RAM, hard disk space, and screen size, threaten to limit access as well. Network access variety (some users still use 14K dialup modems while others use 10M cable modems) imposes similar problems. Finally, continually changing software represents an additional concern. As application programs and operating systems evolve, users of current software may find their programs become obsolete because newer versions fail to preserve file format compatibility. As far as the problem of evolving software is concerned, the recent spread of Java applications is a promising step since Java supports platform independence. Yet, the execution of Java programs is time consuming, potentially limiting users' satisfaction.

Undoubtedly, it is necessary to accommodate varying processor speeds, hardware components, input devices, and network access speeds, and to design software platforms that promote evolution while ensuring compatibility and bi-directional file conversion, in order to deal with the currently nagging problem of incompatibility (Schneiderman 2000). Since limiting the technological progress is usually an unsatisfactory solution (if possible at all), a strategy that could lead to very exciting results and systems is to make user interfaces malleable by making them adaptable.

Comprehensiveness of graphical design

A second challenge to enhancing the usability of human-computer interfaces is to make them easily comprehensible. As far as the fundamental goals of GUI design are concerned, the aim is to omit (or at least minimise) extensive technical terminology, irreversible actions, user-uncontrollable actions, unstructured screen layouts, confusing menu choices, incomprehensible error massages and unexpected crashes.

A more sophisticated and appealing challenge is to account for differences in computing skills and experience of potential users. As reported by Schneiderman (2000), some users need only a few minutes of orientation to understand the novelties and begin to use new tools successfully, others need more time since they are not familiar with the application domain or not accustomed to the specifics of the encountered interface or, simply, not familiar with computing anyway. In turn, it is necessary to provide HCI systems with lucid instructions for use and error-prevention mechanisms, in general, as well as with effective tutorials for novices, constructive help files for intermittent users, and compact presentations for experts.

A similar challenge in making HCI systems universally usable is to accommodate users having some impairment or disability. To reach this goal it is necessary to accommodate a user-controlled font size and contrast (crucial for partially sighted and elderly users), alternative access for physically disabled users (e.g. plug-ins for a disability-customised interface), and easily comprehensible layouts (preferably fully graphical) specially developed for users with mild learning/memory disabilities.

Allowing natural multi-modal communication

A third challenge in making HCI systems universally usable is to establish human/ computer interaction that captures attributes of human/human communication and approaches its naturalness. As already mentioned above, people favour the sensory dimensions of sight, sound, and touch as primary channels of communication because they are elementary constituents of usual face-to-face interpersonal interaction. This is why one long-term goal in human/computer interaction research concerns the integration of these "natural" modalities that humans employ to interact with each other into HCI systems (Sharma et al. 1998).

As remarked by Schneiderman (2000, 2001), many HCI systems available today assume users' proficiency in computing; they are, for a casual user, often cumbersome and obtrusive, lacking the adaptability necessary to accommodate users with various levels of computing skill and experience. Furthermore, virtually all "classic" HCI systems tend to confine the user to a less natural, uni-modal means of interaction (e.g. a mouse movement, pressing of a key, speech input, or hand motion). For example, to manipulate a virtual object with a typical HCI system, the user is usually required to select the object by employing mouse motion, then point with the mouse at a control panel to change the object's colour. On the other hand, in a more natural setup, the user would point at the object with his finger and say: "Make it red". Integration of more than one "natural" modality into an interface would potentially overcome the current limitations of HCI systems: it would ease the need for specialised training and ease the information- and command-flow bottleneck between the user and the computer. Besides, recent data shows that a multi-modal HCI can be an effective means for reducing uncertainty of uni-modally sensed data (such as speech or hand motion), thereby improving robustness (Oviatt 2000). Although the incorporation of all features of human/ human interaction (i.e. an intricate interplay of thoughts, language, and non-verbal communicative displays) into human/computer interaction may be very complex and difficult to achieve, equipping HCI systems with a multi-modal setup so that they can approach naturalness, flexibility and robustness of human/human communication will give them the potential to:

- transcend the traditional, cumbersome and rigid mouse/keyboard interaction, and
- yield a more effective and efficient information- and command-flow between the user and the computer system and, by that,
- approach universal usability.

With this motivation, automatic speech recognition and spoken-language processing have been topics of research for decades (Juang and Furui 2000). Some other techniques like automatic gesture recognition, analysis of facial and vocal expressions, eye tracking, and analysis of physiological reactions have only recently matured enough to be used more effectively in freeing computer users from the classic keyboards and mice (Roy and Pentland 1997, Yang et al. 1998, Marsic et al. 2000). However, most of the available relevant studies address merely the issues of sensing and interpreting a single human communicative channel (either facial expressions, or vocal intonations, or hand gestures etc.); the role of these modalities in a multi-modal HCI system is still being explored. The basic questions relevant for multi-modal HCI that should be answered are:

- which modalities should be integrated, and
- when and how should these multiple modalities be integrated.

Modalities for more natural HCI

Natural face-to-face interpersonal interaction is perceived through five basic senses and expressed through the production of various communicative signals. We speak about, point at, and look at objects all at the same time. We also listen to the intonation of the spoken words and look at facial expressions and body movements of the speaker to find clues about the discussed subject, the importance the speaker assigns to the discussed notions, his/her feelings about those notions, etc. Based on a person's respiration and clamminess we judge the nervousness and even the personality and health state of the speaker. Yet, when it comes to human/computer interaction, HCI systems tend to confine us to less natural means of communication

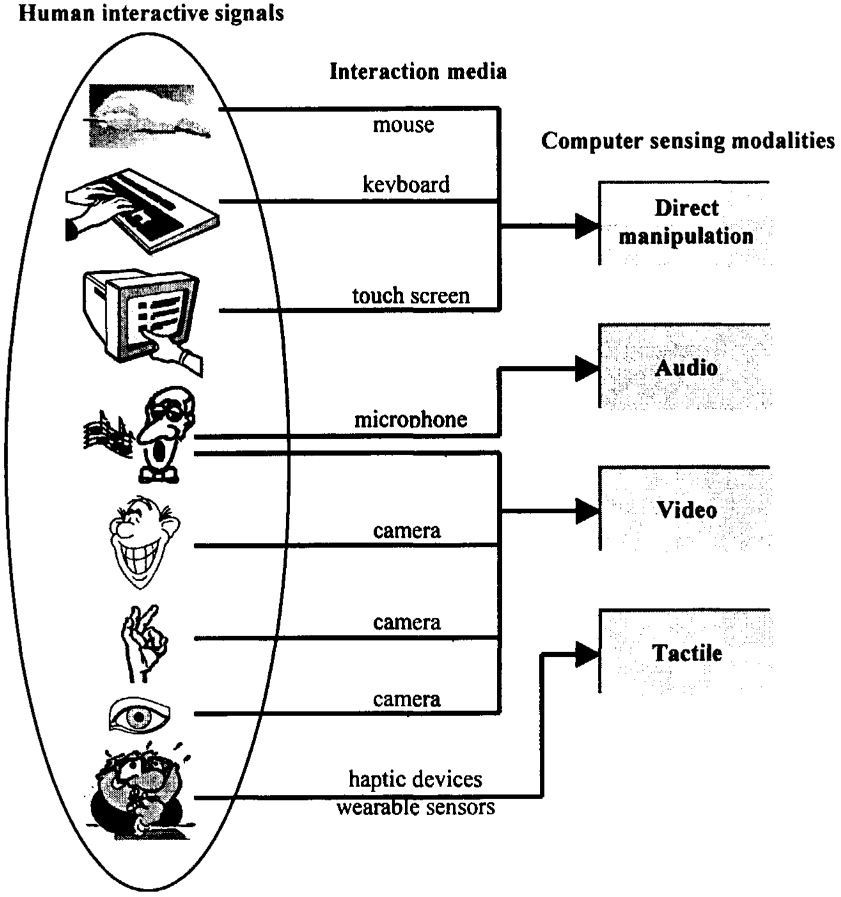


Figure 8.1: Sensing of different human communicative signals (carried by various interaction media) by multiple modalities for HCI

by allowing usage of only one interface device at a time (typing, clicking a mouse button, speaking, touching the touch screen, etc.). To make human/computer interaction as natural as possible, it is necessary to enable computers to interpret all natural human communicative signals (Sharma et al. 1998, Marsic et al. 2000). Hence, computers should interpret human speech and lip movements, hand and body gestures, eye gaze, facial expressions, vocal intonations, and various physiological reactions (Figure 8.1).

Speech is the most natural form of communication among humans and, as machines become ever more widespread, the need to allow natural-speech-based communication between a human and a machine gains significant interest from the computer scientists. The field of automatic speech recognition has witnessed a number of substantial advances in the past two decades, spurred on by advances in signal processing, software and hardware (Juang and Furui 2000). Nevertheless, the current automatic speech recognition technology is still not robust, especially outside controlled environments, under noisy conditions and with multiple speakers (Pentland 2000). One possible solution to this problem involves using microphone arrays and noise-cancellation techniques. However these tend to work only for the environments for which they are designed (Sharma et al. 1998). The human capability to "hear" in noisy environments by means of lip reading initiated research on whether combined audio and video sensing and processing can provide a better solution (Stork and Hennecke 1996). Except of supplementing acoustic speech signals, lip movements as visible speech signals can also provide cues to whether a person is speaking or not (Wojdel and Rothkrantz 2000). This is of particular importance to HCI systems placed in noisy environments as well as to affect-sensitive HCI systems, for which it is crucial to distinguish the movements of the lips due to speech articulation from those representing signs of affective/attitudinal states (section 6.6). However, a number of impediments make robust and widely applicable joint audio-visual speech reading difficult to achieve (Chen 2001): (i) the speech-required lip movements vary for different languages; (ii) optical features to be tracked may be speaker dependent; (iii) though number of solutions have been proposed for the problem of *co-articulation* within words and between words (i.e. acoustic and optical perceiving of a single phone varies with adjacent phones), the issue is still considered hindering; and (iv) there is a lack of consensus on how to combine audio and visual input (see the discussion below). Due to the complexity of the problem and general lack of researchers having expertise in both domains, accomplishing robust automatic audio-visual speech reading still forms a significant research challenge.

Historically, hand movements have been exploited most for HCI (Myers 1996). This is largely due to the dexterity of the human hand, which allows highly accurate selection and manipulation of objects and devices with the help of visual feedback. Numerous interface devices have been based upon hand movement: the keyboard, mouse, joystick, trackball, magnetic wand, touch screen, glove-based device. Also, in the last decade, tremendous progress has been made in the field of visual sensing, detection, tracking and interpretation of human hand gestures (from simple pointing through manipulative gestures to more complex symbolic gestures such as those in sign languages). Several exhaustive surveys on this topic have been published recently: glove-based gestural HCI (Sturman and Zeltzer 1994), hand gesture visual recognition and interpretation (Pavlovic et al. 1997), human modelling techniques (Cerezo et al. 1999), and tracking of human body and hand (Gavrila 1999, Pentland 2000). The current progress in automatic visual analysis of hand gestures opened up possibilities for enhancing the state of the art in HCI – devices like the mouse and the joystick could be replaced by allowing a more natural interaction based on finger-pointing-based commands. However, visual sensing of hand gestures for HCI suffers difficulties from both a theoretical and a practical standpoint. Which interpretation should be assigned to which visually detected hand gestures is still a subject of debate, particularly when it is desirable not to put restrictions on the complexity of the hand movements to be monitored for more natural HCI. The main problem here is that the interpretation of body gestures is context dependent, that is, culture, person, situation, and task dependent (Efron 1941, Russell and Fernandez-Dols 1997). One source of help for this problem is machine learning: instead of having a priori generic rules for human body gesture interpretation, we can potentially learn appropriate context-sensitive rules by watching the user in the environment in which the intended gesture analyser is to be deployed (see also the discussion about context-sensitive HCI systems given in the subsequent section). From a practical standpoint, visual sensing involves the processing of huge amounts of data in real time, which might put undue demands on the required processing speed. Yet this problem becomes less and less significant as the computer hardware gets faster and computer memory prices drop (Hassler 2001). A more critical practical issue, common for all visual sensing including gaze, lip-movement, and facial-gesture tracking, concerns: scale, pose and occlusion. Namely, in most real-life situations it cannot be assumed that the subject will remain immovable; rigid head and body motions can be expected causing changes in the viewing angle and in the visibility and illumination of the tracked facial and body features. Though interesting progress in addressing these issues has been made in machine vision research (for a more detailed discussion, the reader is referred to section 8.5), these problems in general and context-sensitive understanding of human body gestures in particular pose significant research challenges (Pavlovic et al 1997, Pentland 2000).

Except lip movements and hand gestures, the *sight* modality also includes visual sensing of gaze and facial expressions. Where the user is looking can provide a clue to the intended meaning of a particular action (i.e. task discovery). For example, if several windows are currently open and the user issues a spoken command "zoom in", the direction of the user's gaze can be employed to detect the right window on which to zoom in. Also, gaze tracking can be employed for controlling a display, for instance, to scroll by looking to the left, right, up or down (Stiefelhagen and Yang 1997). Numerous eye-tracking systems have been proposed in the literature up to date (Morimoto et al 2000). Though the presence of pupil-brightness-response variations for different subjects imposes a problematic phenomenon, real-time gaze tracking may be considered, in principle, a solved issue that can be effectively used to free computer users from the classic keyboard and mouse (Morimoto et al 2000). Yet the same cannot be said for the techniques available for automatic face detection, tracking and interpretation. Similarly to the case of visual sensing and interpretation of hand gestures, automatic visual analysis of facial expressions suffers difficulties from both a theoretical and a practical standpoint. From a theoretical standpoint, the main impediment to accomplishing universally usable automatic facial expression analysis is the lack of consensus on the human perception of facial gestures (section 6.2). In other words, which interpretation should be assigned to which visually detected facial expression is still an issue of debate, particularly when it is desirable not to put restrictions on the variety of facial expressions (attitudinal/emotional, emblematic, manipulative, illustrative, or interaction-regulative) to be monitored for more natural HCI. The crucial issue here is that of context dependency: it is very difficult to anticipate someone's facial expression due to the fact that facial expression interpretation is generally situation and person dependent (chapter 6). From a practical standpoint, to accomplish robust visual sensing of facial expressions, the intended monitoring tool should be able to detect faces and facial features in arbitrary scenes under various lighting and viewing conditions and independently of distractions like glasses and facial hair. If we take into consideration the current state of the art in automatic facial expression analysis (chapter 2 and/or Pantic and Rothkrantz 2000d, section 8.4), the accomplishment of robust, automatic, context-sensitive facial expression analysis in real time still lies in a relatively distant future (section 8.5).

Finally, except *sound* and *sight* modalities, a *tactile* computer-sensing modality for more natural HCI has been explored recently with increasing interest. Especially since haptic devices are now commercially available (e.g. Rutgers force-feedback tactile glove designed for interaction with virtual environments; Burdea 1996), the sensory dimension of touch has become a potentially realistic solution to a variety of interaction design challenges (Marsic et al. 2000, Oakley et al. 2000). Computer sensing of touch and force is particularly important for building a proper feel of "realism" in virtual reality: force-feedback capability is essential for grasping, moving, and placing virtual objects (Engel et al. 1994). Not only force sensing (Bergamasco 1995, Oakley et al 2000) has been used to enhance HCI; also sensing of brain electrical activity (Putnam and Knapp 1993, Nasman et al. 1997), muscular electrical activity (Suryanarayanan and Reddy 1997, Picard 1997), and other physiological human reactions like respiration, temperature, and heart rate (Picard and Healey 1997). The key idea behind this recent interest in introducing tactile modality in HCI is threefold: it might form a means of making HCI accessible for physically disabled users (Lusted and Knapp 1996), a means for reducing visual overload in the conventional desktop (Oakley et al. 2000), and a means for sensing affective states of the user (Healey and Picard 1998, Vyzas and Picard 1999). Yet many theoretical and practical open problems are still to be addressed. For instance, the application of haptic technology might have a profound impact on the users' fatigue if done improperly (Oakley et al. 2000), currently available wearable physiological sensors imply wiring the subject, which is usually experienced as uncomfortable, skin-sensors are very fragile (Djurica 2001) and the accuracy of measurements is commonly affected by hand washing and the amount of gel used (Cacioppo et al. 2000).

Integrating multiple modalities for more natural HCI

The performance of the intended multi-modal HCI is not only greatly influenced by the different types of modalities to be integrated; the abstraction level at which the pertinent multiple modalities are to be integrated/fused and the technique which is to be applied to carry out multi-sensory data fusion are clearly of the utmost importance as well. In general, there is a lack of consensus within the multi-modal-HCI research community as to how exactly multi-sensory data fusion is to be carried out (Figure 8.2). Yet, if the goal is to ensure that HCI approaches the naturalness of human/human communication, three pragmatic issues can be considered when making the decision about how to integrate data from individual computer-sensing modalities into more complex multi-modal decisions:

1. How are the modalities combined in natural human/human interaction?
2. Does this hold in HCI?
3. Are there any existing data-fusion techniques that support the desired coupling of multiple modalities?

Insight into how the modalities of sight, sound and touch are combined in natural human/human interaction can be gained from neurological studies on fusion of sensory neurons (Bower 1974, Stein and Meredith 1993). Three concepts relevant to multi-modal fusion can be distinguished:

1. *1+1 > 2*: The response of multi-sensory neurons is stronger for multiple weak input sensory signals than for a single strong signal.
2. *Context dependency*: The fusion of sensory neurons is modulated according to the signals received from the cerebral cortex: depending on the sensed context, different combinations of sensory signals are made.
3. *Handling of discordances*: Based upon the sensed context, sensory discordances (i.e. sensor malfunctioning) are either handled by fusing the sensory observations without any regard for individual discordances (e.g. when a fast response is necessary), or by attempting to recalibrate discordant sensors (e.g. by taking a second look), or by suppressing discordant sensors (e.g. when one sensory observation is contradictory to another).

Hence, humans simultaneously employ the tightly coupled modalities of sight, sound and touch (McNeill 1992). As a result, analysis of the perceived information is highly robust and flexible; undetected or noisy information from one channel is recovered or explained by the information available from other channels (e.g. in noisy environments we "hear" what has been said by means of lip reading). Several studies confirmed that this tight coupling of different modalities persists when the modalities are used for HCI (e.g. Oviatt et al. 1997, Nakatsu 1998, Chen et al. 1998, Chen 2001).

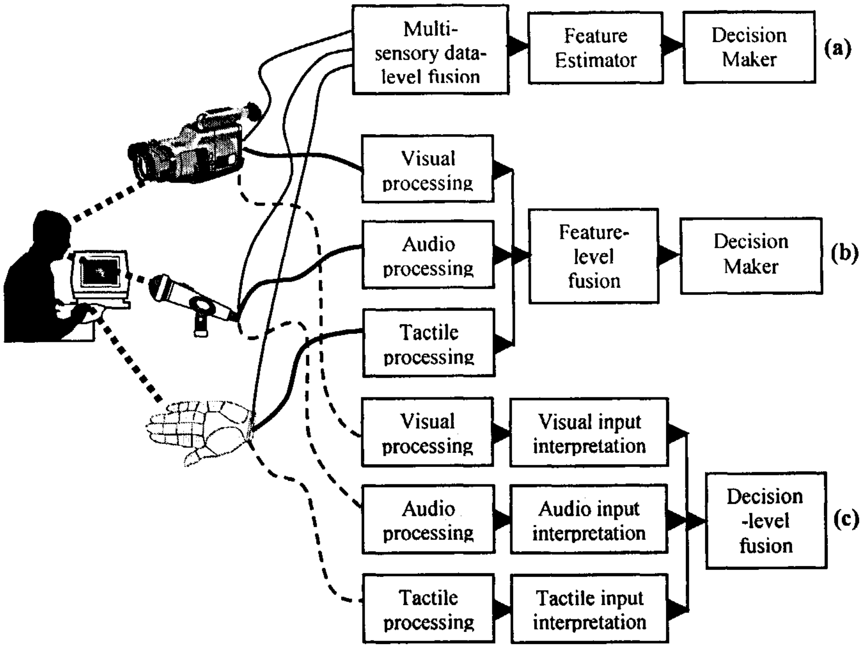


Figure 8.2: Fusion of multiple sensing modalities: (a) data-level fusion integrates raw sensory data; (b) feature-level fusion combines features from individual modalities; (c) decision-level fusion combines data from different modalities at the end of the analysis

A question remains, nevertheless, as to whether such a tight coupling of multiple modalities can be achieved using the theoretical and computational apparatus developed in the field of sensory data fusion (Hall and Llinas 1997, Dasarathy 1997). As illustrated in Figure 8.2, fusion of multi-sensory information can be accomplished at the three levels: data, feature, and decision level. Data-level fusion involves integration of raw sensory observations and can be accomplished only when the observations are of the same type. Since the monitored human interactive signals are of different nature and are observed using different types of sensors (Figure 8.1), data-level fusion is, in principle, not applicable to multi-modal HCI. Feature-level fusion assumes that each stream of sensory information is first analysed for features and then the detected features are fused. Feature-level fusion retains less detailed information than data-level fusion, but it is also less prone to noise and sensor failures, and, most importantly, it is the most appropriate type of fusion for tightly coupled and synchronised modalities. Though many feature-level techniques like Kalman fusion, ANN-based fusion, and HMM-based fusion have been proposed (Stork and Hennecke 1996, Dasarathy 1997), decision-level (i.e. interpretation-level) fusion is the most frequently applied approach to multi-modal integration (Sharma et al. 1998, Marsic et al. 2000, Pentland 2000). Yet, it is almost certainly incorrect to use a decision-level fusion since people display audio, visual, and tactile communicative signals in a complementary and redundant manner.

In order to accomplish a multi-modal analysis of human interactive signals acquired by multiple sensors, which will resemble human processing of such information, the input signals cannot be considered mutually independent and cannot be combined only at the end of the intended analysis. In turn, the input data should be processed in the joint feature space. In practice, however, there are two major difficulties: a huge joint feature space (resulting in a heavy computational burden) and different feature formats and timing. A way to deal with these problems and to achieve a tightly coupled multi-sensory fusion is to apply a Bayesian inference method, as presented in (Pan et al. 1999). However, due to the complexity of the phenomena and general lack of researchers having expertise in all domains (audio, visual, and tactile processing), untangling the problem of joint audio-visual-tactile analysis of human interactive displays is still a significant challenge facing researchers of multi-modal HCI systems.

Context-sensitive HCI systems

Another challenge in fashioning universally usable HCI systems is to make them context sensitive. The key idea is to account for individual differences of the users and for the overall situation in which the user acts. To achieve this, HCI systems should be able to specify in an automatic way:

1. Who the user is?
2. Where the user is?
3. What the user's current task is (what he is doing and what he intends to do)?
4. How the user is feeling?

Based upon the user's identity and the knowledge about his/her environment and current task, it would be possible to retrieve information about the importance of that task in the given environment, the user's skills in performing that task, and the user's overall preferences. Together with the sensed user's affective state, this information could be employed to define the following:

- *What form should the instruction manual have?* For example, for a novel user a lucid tutorial could be provided, to an average user constructive help files could be offered, for an expert user compact notes could prove to be sufficient. Yet the kind of help files provided to the user should not be rigid; it should be determined based upon the overall user's preferences and his/her current affective state. Namely, users might prefer compact help files even if they are novices and especially if they are in a hurry.

- *When should the user be interrupted?* This is of particular importance for the systems that are designed such that they rely on the user's feedback at all times or offer assistance each time a certain task is to be performed or particular conditions are encountered. The user's current affective state and the importance of his/her current task given his/her environment could be exploited to time the interrupts conveniently. For example, if the user is hurriedly writing e-mails, interrupting him to correct a syntax mistake can be postponed till the moment he tries to send the e-mail.
- *When, in which part, and in what way should the system be adjusted?* The sensed user's affective state could be exploited to time the adjustment of the system, the information about the user's current task might form the target of the adjustment, and the complete information about the sensed context could be used to determine the adjustment properly. For instance, suppose that the user always browses through a particular application in the same way in order to come to a specific window and displays irritation each time the system starts the pertinent application by displaying its very first window. In that case, a proper adjustment might be to mark the window where the user commonly stops browsing and to start the pertinent application with that specific window (browsing through the preceding windows of the application does not have to be apparent to the user). Yet, suppose that the user always becomes frustrated if a certain person enters the office. In that case, no adjustment should be made since the user's affective state is not caused by HCI but by an external (and for HCI) irrelevant event.

As embedded computing devices become ever more omnipresent in society, sophisticated HCI systems that can adjust to individual differences of potential users and adapt to an overall situation delimited by the task and by environmental constraints promise flexibilities and functionalities necessary for the coming era of ubiquitous computing. Though the specifics of these user-centred context-sensitive HCI systems are still far from delineated, it is clear that before this new HCI systems can be deployed, they should be equipped with context-sensing technology and machine-learning techniques that would allow them to adapt to both the overall situation and the individual user. Although it was initially thought that visual context sensing would be the research problem that would be the hardest to solve, tremendous progress has been made in the last decade. Several exhaustive surveys on various related topics have been published recently:

- Person identification (Samal and Iyengar 1992, Adini et al. 1997) and bi-modal speaker verification (Chen and Rao 1998),
- Person detection and tracking (Gavrila 1999, Collins et al. 2000),

- Detecting environmental cues (Collins et al. 2000, Strobel et al. 2001, Patras 2001),
- Affect-sensitive monitoring (Pantic and Rothkrantz 2000d, 2001a, Cowie et al. 2001).

Table 8.1
Multi-modal HCI research vs. Context-sensitive HCI research: similarities and differences

DIMENSION	MULTI-MODAL HCI	CONTEXT-SENSITIVE HCI
Purpose	Allowing a more natural HCI according to the human/ human interaction model Ultimate research goal: Achieving universally usable and accessible HCI systems	Allowing adaptive HCI that, similarly to the human adaptation capability, can adjust to both the overall situation and the individual user Ultimate research goal: Achieving universally usable and accessible HCI systems
Goal	Introducing computer-sensing modalities of sight, sound and touch into HCI	Context sensing: person identification, task detection, environment recognition, person's affective state interpretation
Phenomena measured (techniques)	Speech recognition, lip reading, gaze detection, hand and body gestures detection / interpretation, facial expressions detection / interpretation, physiological reactions detection / interpretation	Face detection / identification, hand and body gestures detection / interpretation, speech recognition, gaze detection, environment monitoring (object tracking / detection), facial expression detection / interpretation, vocal expression detection / interpretation, physiological reactions detection / interpretation
Main problem areas	*Audio input*: noisy environments with multiple speakers are difficult to handle *Visual input*: occlusions and variations in scale, pose, and illumination are difficult to handle *Tactile input*: inaccurate measurements due to fragile skin sensors are difficult to handle *Multi-modal input*: processing of multi-sensory data in joint feature spaces is difficult to achieve *Interpretation*: Context dependency (user, task, and environment dependency) of human behaviour is difficult to handle in a general case since there are no generic rules of human behaviour; machine-learning techniques should be employed to make HCI context adaptable, but the specifics of this adaptation process are difficult to delimit due to the complexity of the relevant socio-technical issues	

However, many theoretical and practical open problems are still to be addressed in the research field of context-sensitive HCI. Since the research on context-sensitive HCI deals with similar issues as the research on multi-modal HCI systems, the theoretical and practical challenges facing researchers in these fields are very much alike (Table 8.1). Both research fields aim at accomplishing a form of human/computer interaction that approaches the flexibility of human/human interaction. Moreover, both seek to achieve robust multi-modal sensing and processing of human behavioural cues. Finally, perhaps the most significant challenge facing researchers in both fields is to achieve a joint audio-visual-tactile context-sensitive interpretation of human behaviour.

While we may agree that an automated context-sensitive multi-modal interpretation of human behavioural cues would be enormously beneficial for the development of universally usable HCI systems, we also should recognise the likelihood that such a goal still lies in the relatively distant future due to the state of the art in sensing technology and signal-processing techniques (see the preceding section) as well as due to numerous untangled socio-psychological aspects of human behaviour (see section 8.3). Still, at the very least, multi-modal context-sensitive HCI systems are among the most exiting and economically important research areas in computer science (Pentland 2000).

Anthropomorphic response

Anthropomorphically designed HCI systems (AD-HCI) attempt to make computers more accessible to the non-technical user by endowing them with looks and behaviours that are, at least superficially, human-like. Typical AD-HCI systems deal with engendering and incorporating animated and virtual characters within man-machine virtual interactive environments (e.g. Figure 8.3). A virtual environment (e.g. an internet-based virtual insurance agency) is typically inhabited by a clone (*avatar*), representing a real person, and by virtual autonomous *actors* (e.g. an animated insurance broker) (Kshirsagar and Thalmann 2000). In order to accomplish a multi-modal natural interaction between these entities, mimicking of the avatar and autonomy of the actors have to be considered. The mimicked features are usually speech (spoken words and intonation), facial expressions and body gestures (Thalmann et al. 1998). The autonomy of an actor might be achieved by enabling it to manipulate its own goals and to generate its responses (displayed in terms of both verbal and non-verbal human-like interactive signals) based upon intentions and affective states displayed by the avatar. Thus, except for the animation issue, the research field of AD-HCI is concerned with the very same topics (i.e. gesture, speech, affect, and context) as the research fields of multi-modal and context-sensitive HCI systems.

As remarked by Shoham (1999), Coen (1999), and Pentland (2000), breakthroughs in AD-oriented HCI systems could bring about the most radical change in the computing world. They could change not only how professionals practice computing, but also how mass consumers conceive of and interact with the technology. However, while the technology in animation has advanced to a level of commercial relevance (Thalmann et al. 1998), other aspects of AD-HCI, in particular ones concerned with the interpretation and emulation of human behaviour at a deeper level, are less mature and need many improvements (preceding section, Tekalp 1998, Pentland 2000, Pantic and Rothkrantz 2001a). Critical issues and great challenges in the design and development of the AD-oriented HCI systems are the problems related to untangling the context in which the user acts, deciding the proper context-dependent question to ask, choosing a suitable moment to pose a question, determining whether to interrupt the user at all, and eliciting a proper response (e.g. which words, facial expression, and intonation to use).

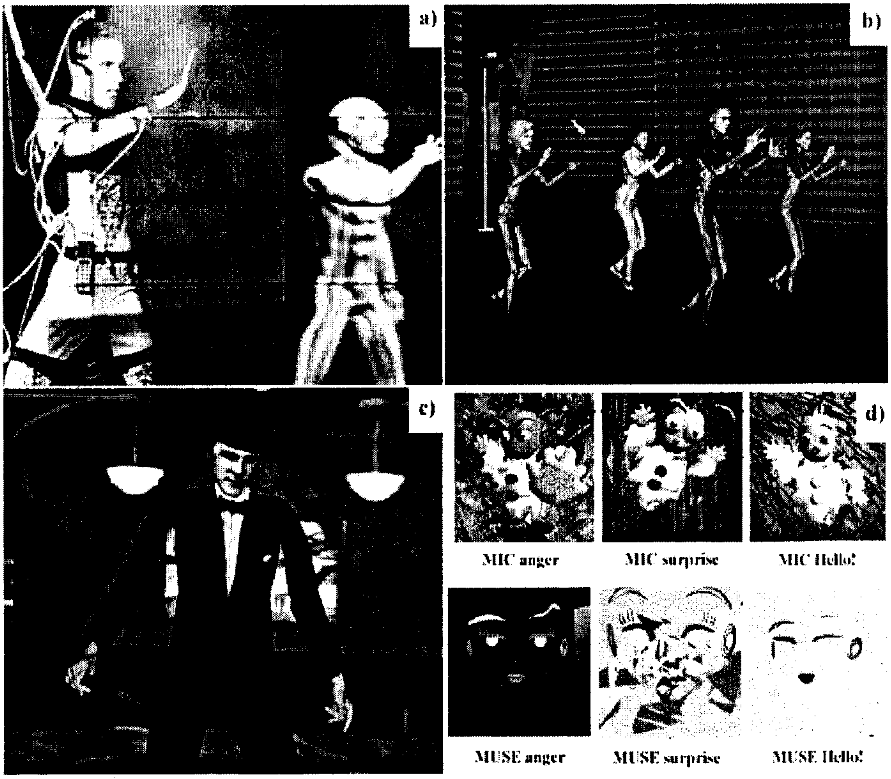


Figure 8.3: Virtual characters: (a) A dancer and his avatar; (b) Cyber-dance of virtual actors; (c) A virtual real-estate agent; (d) Virtual emotion-sensing characters MIC and MUSE. (a)-(c): Thalmann and Moccozet 1998. (d): Tosa and Nakatsu 1996.

In addition, another issue should be considered while discussing AD-oriented HCI. As remarked by Turkle (1984) and Schneiderman (1993), if the boundary between computers and people blurs, this may undermine a child's emerging sense of self. Attributing intelligence, independent activity, or free will to computers may lead children to believe that they are autonomous, undercutting their responsibility for mistakes and for poor treatment of friends, teachers, or parents. Therefore it is important to make AD-types of HCI systems to be transparent, to emphasize the fact that machines do what we design/program them to do and that any emerging result is the product of our own effort. The key idea here is to design virtual characters which portray computers as tools and not as invisible persons. Well-chosen words, for instance, can make a great difference (e.g. "Would you like some help?" instead of "Can I help you?"). For more detailed guidelines on design of "transparent" HCI, the reader is referred to Schneiderman (1993).

When should we expect the "fourth generation" HCI?

Obviously, as remarked in the preceding sections, HCI design breakthroughs are necessary if the computing technology is to achieve the ultimate goal of being universally usable. Yet, during the last decade, many research problems initially thought to be intractable (e.g. sensing and detecting of human communicative displays, affect-sensitive interpretation of those, and context sensing in general), have been proven manageable and have even spawned several thriving commercial enterprises (Pentland 2000). Still, *when* this new generation of universally usable and accessible HCI systems will actually be deployed remains an open question. A way of answering this question is to consider two pragmatic issues:

- *To what extent does the development of more advanced technology constrain the actual deployment of a certain HCI design?* For example, while the technological means are now in hand to develop comprehensive GUI systems which accommodate the variety of commonly used equipment and ensure compatibility and bi-directional file conversion, the same is not the case for multi-modal, context-sensitive, and anthropomorphically designed HCI systems. As mentioned above, before these novel HCI systems can be widely deployed, they must be equipped with sensing technology that allows robust and accurate sensing and detection of multi-modal human interactive signals and their context-dependent interpretation and emulation.
- *To what extent does the realisation of certain HCI systems constrain the realisation of other HCI systems?* As explained above, multi-modal HCI systems and context-sensitive HCI systems are mutually dependent (Table 8.1): while robust and accurate sensing and detection of human verbal and non-verbal interactive displays must be achieved to realise context-sensitive HCI, context-dependent interpretation of sensed human behavioural cues must be achieved to realise the integration of human natural modalities of sight, sound, and touch, into HCI systems. Further, the realisation of multi-modal context-sensitive HCI

constrains the actual deployment of AD-HCI: context-sensitive monitoring and interpretation of human verbal and non-verbal communicative signals is a prerequisite for achieving proper, context-dependent responses by virtual actors. Finally, multi-modal HCI, context-sensitive HCI, and anthropomorphic HCI must be comprehensive while ensuring compatibility, bi-directional file conversion, and support for the variety of equipment.

Based on these observations, the coming of the next generation of universally usable HCI systems, might be represented in terms of the time-scale and the required advanced technology by Figure 8.4.

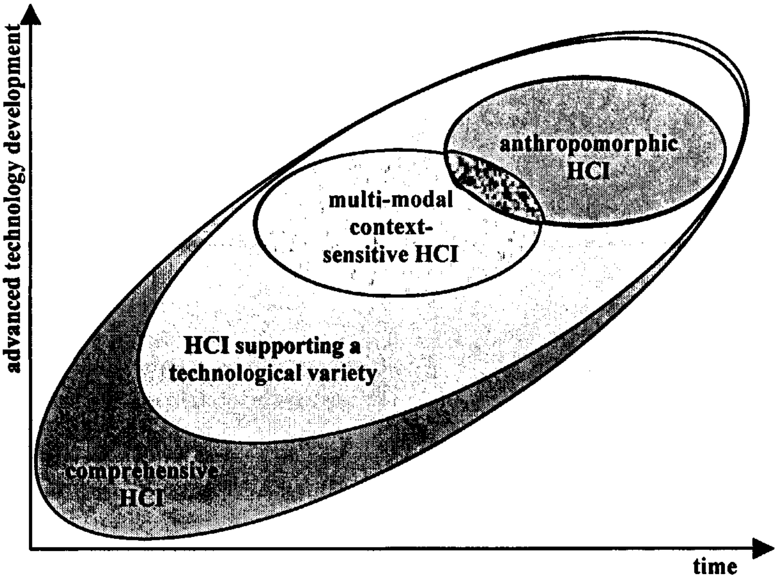

advanced technology development
anthropomorphic HCI
multi-modal context-sensitive HCI
HCI supporting a technological variety
comprehensive HCI
time

Figure 8.4: The coming of the new generation HCI is constrained by the technological development and by the mutual dependencies between different kinds of HCI systems

8.3 Affect-sensitive HCI: The problem domain

One of the key challenges in enhancing the usability of human-computer interfaces is to devise human/computer interaction that retains attributes of human/human interaction and approaches its naturalness. Hence, one long-term goal in HCI research is (section 8.2):

- to integrate human natural modalities of sight, sound and touch into HCI systems (Sharma et al. 1998, Marsic et al. 2000), and
- to make HCI systems sensitive and adaptable to the context (i.e. to the current user, his preferences, affective state, task, and to the overall environmental constraints) in which they operate (Pentland 2000).

While the preceding section discussed the problem of fashioning multi-modal context-sensitive HCI in general, the rest of this chapter is concerned with only one aspect of this problem, namely, with providing machines with the ability to detect and interpret user's affective states.

Because there are numerous areas where benefits could accrue from the automation of affect-sensitive monitoring of human communicative displays, tackling this problem has attracted the interest of many AI researchers. Besides enhancing the usability of HCI systems by enabling them to sense and respond appropriately to users' affective feedback (Picard 1997), automatic affect-sensitive monitoring tools could simplify and improve the research in areas as diverse as behavioural science, anthropology, medicine, psycho-physiology and political sciences (section 6.1). The automatic assessment of attitudinal states like boredom, inattention, and stress would be valuable for preventing critical situations in hazardous working environments like aircraft cockpits, nuclear power plant surveillance rooms, air traffic control towers, or simply in the vehicles like trucks, trains, and cars. An advantage of affect-sensitive monitoring done by computer is that it compromises people's privacy less than monitoring by human observers; an automated tool could provide prompts for better performance based on the sensed user's affective state. Besides, a computer-based monitoring can be more accurate than that carried out by human observers since computers can be equipped with sensory modalities that humans lack (e.g. the EEG and EMG).

However, while there is a general agreement that the automation of multi-modal affect-sensitive monitoring of human interactive cues would be enormously beneficial, tackling this problem is not an easy task. The main problem areas concern the following:

1. *What is an affective state?* This question is related to psychological, linguistic, and physiological issues pertaining to the nature of affective states and the way affective states are to be described by automated human-affect analysers.
2. *What kinds of evidence warrant conclusions about affective states?* In other words, which human communicative signals convey messages about an affective arousal? This issue shapes the choice of different modalities to be integrated into automated affect-sensitive monitoring tools.
3. *How can various kinds of evidence be combined to generate conclusions about affective states?* This question is related to neurological issues of human sensory-information fusion, which shapes the way multi-sensory data is to be combined within automated affect-sensitive monitoring tools.

This section examines basic issues in these problem areas and provides a taxonomy of the pertinent problem domain. The section will begin by examining the body of research literature on the human perception of affective states, which is large, but disunited. This lack of consensus among basic researchers on a set of basic emotions that can be universally recognised implies that the selection of a list of affective states to be recognised by an intended automated affect-sensitive monitoring tool requires pragmatic choices. The section will then explain the capability of the human sensory system in the detection and understanding of the other party's affective state. It is meant to serve as an ultimate goal in the development of automated multi-modal affect-sensitive monitoring tools and as a basis for addressing two main issues relevant to such HCI tools: *which* modalities should be integrated and *how* should those modalities be combined.

Psychological issues

Since an automated analyser of human affective states would be extremely beneficial, the question of how the human perception of affective states can be characterised best has become an important concern for many researchers in affective computing. Ironically, as already remarked in section 6.2, the growing interest in affective computing comes at a time when the established wisdom on human affective states is being strongly challenged in the basic research literature (for detailed summaries on issues debated in the basic research on emotion, readers are referred to Cornelius (1996) and Cowie et al. (2001)).

On the one hand, classic psychological research claims the existence of six basic expressions of emotions that are universally displayed and recognized: anger, happiness, sadness, surprise, disgust, fear (Darwin 1965/1872, Bezooijen 1984, Keltner and Ekman 2000). This implies that, apart from verbal communicative signals (spoken words), which are person dependant (Furnas et al. 1987), non-verbal communicative signals (i.e. facial expression, vocal intonations, body gestures, and physiological reactions) involved in these basic emotions are displayed and recognized cross-culturally.

On the other hand, there is now a growing body of psychological research that strongly challenges the classical theory on emotion. The psychologist James Russell argues that emotion in general can best be characterized in terms of a multi-dimensional affect space, rather than in terms of a small number of emotion categories (Russell 1994, Russell and Fernandez-Dols 1997). He also criticises experimental design flaws applied in classic studies (e.g. using a single corpus of unnatural stimuli and within-subject designs; see section 6.2). Besides Russell (1991), other social constructivists like Averill (1986) argue that emotions are socially constructed ways of interpreting and responding to particular classes of situations and that they do not explain the genuine feeling (affect). In addition, some psychologists imply that attitude is a kind of affect (Fishbein and Ajazen 1975), while others consider affect as a component of attitude (Pratakanis et al. 1989).

Further, while classic studies claim that the basic emotions (whatever that may mean) are hardwired in the sense that there are some specific neural structures corresponding to different emotions, alternative studies like the study of Ortony and Turner (1990) suggest that it is not emotions but some components of emotions which are universally linked with certain communicative displays like facial expressions. Except for this lack of consensus on the nature of emotion, there is no agreement on how affective states should be labelled/named. The key issue here, which stands in contradiction to the classic studies' emphasis on emotions as a product of evolution, is that of culture dependency: the comprehension of a given emotion label and the expression of the related emotion are culture dependent (Matsumoto 1990, Wierzbicka 1993, Shigeno 1998, Cacioppo et al. 2000).

In summary, the available body of basic research literature is excessively fragmented and does not provide firm conclusions that could be safely assumed and employed in studies on affective computing. Due to this unresolved debate concerning the standard emphasis on emotions as a product of evolution and evidence that they are culture dependent, there is no agreement on a set of basic emotions that are displayed and recognised uniformly across different cultures. In other words, it is not certain that each of us will express a particular affective state by modulating the same communicative signals in the same way, nor is it certain that a particular modulation of interactive cues will be interpreted always in the same way independently the observer. The immediate implication is that pragmatic choices (e.g. application- and user-dependent choices) must be made regarding the selection of affective states to be recognised and the appropriate recognition mechanism to be employed by an automated human-affect analyser.

As already suggested in chapter 6 for the case of the automated human facial affect analyser, a way in which this problem can be handled is to apply machine learning: rather than using a priori generic rules for affective state recognition, we can potentially learn the rules by interacting with the user about his/her interpretations of the observed affective displays in the given environment/application domain. In other words, a promising strategy is to build a personalised, context-sensitive analyser of human affective states capable of adapting the employed communicative-signal classification mechanism according to the sensed context and the user's wishes.

Human performance

Affective arousal modulates all verbal and non-verbal communicative signals. As shown by Furnas et al. (1987), it is very difficult to anticipate a person's word choice and the associated intent: even in highly constrained situations, different people choose different words to mean exactly the same thing. On the other hand, in usual interpersonal face-to-face interaction, people detect and interpret non-verbal communicative signals in terms of affective states expressed by their communicator with little or no effort (Ekman and Friesen 1969). Although the correct recognition of someone's affective state depends on many factors (the attention given to the speaker and the familiarity with the speaker's personality, face, vocal intonation, etc.), humans recognise affect with apparent ease.

The human sensory system does not only use multi-modal analysis of multiple communication channels to interpret face-to-face communication, but also to recognise another party's affective states. A channel is a communication medium (e.g. the auditory channel that carries vocal intonations and the visual channel that carries facial expressions) while a modality is a sense used to perceive signals from the outside world (e.g. the senses of sight and hearing). The interpretation of another party's affective states in usual face-to-face interaction involves simultaneous usage of many channels and combined activation of various modalities. Hence, the analysis of the communicator's attitudinal states becomes highly flexible and robust. Failure of one channel is recovered by another channel and a message in one channel can be explained by that in another channel (e.g. a mouth expression that might be interpreted as a smile will be seen as a display of sadness if at the same time we can see tears and hear sobbing).

The abilities of the human sensory system define, in some way, the expectations for automated affect-sensitive HCI tools. Although it may not be possible to incorporate all features of the human sensory system into an automated system (due to the complexity of the phenomena, which involves an intricate interplay of knowledge, thoughts, language, and non-verbal behavioural cues), the affect-recognition capability of the human sensory system can serve as the ultimate goal and a guide for determining design recommendations for an automatic affect-sensitive HCI tool.

Taxonomy: modalities, their fusion, temporal information, and context dependency

The taxonomy of the automatic affect-sensitive-monitoring problem domain can be devised by considering the following issues:

- Which channels of information, corresponding to which human interactive channels, should be integrated into automated human-affect analysers?
- How should the data carried by multiple channels be fused to achieve a human-like performance in recognising affective states shown by the monitored subject?
- How should the temporal aspects of the information carried by multiple channels be handled?
- How can automated human-affect analysers be made more sensitive to the context in which they operate? The relevant issue here is that the interpretation of human communicative displays is situation and person dependent.

Though one could expect that automated human-affect analysers should include all human communicative modalities (sight, sound, and touch) and should analyse all non-verbal interactive signals (facial expressions, vocal expressions, body gestures, and physiological reactions), the reported research does not confirm this finding. The visual channel that carries facial expressions and the auditory channel that carries vocal intonations are widely thought of as most important in human recognition of affective states (Cowie et al. 2001). As already mentioned in chapter 1, according to Mehrabian (1968), whether the listener feels liked or disliked depends for 38% on vocal utterances and for even 55% on facial expressions. This indicates that, while judging someone's affective state, people rely less on body gestures and physiological reactions displayed by the observed person; they rely mainly on his/her facial expressions and vocal intonations. As far as body gestures are concerned, as much as 90% of body gestures are associated exclusively with speech (Church and Meadow 1986, McNeill 1992), so body gestures play a secondary role in human recognition of affective states. As far as physiological signals are concerned, people commonly neglect these since they cannot sense them at all times. Namely, in order to detect someone's clamminess or heart rate, the observer should be in a physical contact (touch) with the observed person. Nevertheless, the research in psycho-physiology has produced firm evidence that the affective arousal has a range of somatic and physiological correlates such as pupillary diameter, heart rate, skin clamminess, temperature, respiration velocity, etc. (Cacioppo et al. 2000). However, the integration of tactile channels carrying physiological reactions of the monitored subject into an automated human-affect analyser requires wiring the subject, which is usually perceived of as uncomfortable and unpleasant. Though the recent advent of non-intrusive sensors and wearable computers (Mann 1997, Clarkson et al. 2000) opened up possibilities for less invasive physiological sensing, yet another problem persists: currently available skin sensors are very fragile (Djurica 2001) and the accuracy of the measurements is commonly affected by hand washing and the amount of gel used (Cacioppo et al. 2000). Hence, automated affect-sensitive monitoring tools should at least combine modalities for perceiving facial and vocal expressions of attitudinal states. Optionally, if provided with robust sensory equipment, they could include the modality for perceiving affective physiological reactions as well (Figure 8.5).

People employ the interactive channels in a complementary and redundant manner. Automatic human-affect analysers should work in the same way: observation channels must be considered together and information carried by multiple channels cannot be combined only at the end of the intended analysis. As explained in section 8.2, multi-sensory information should be fused at the feature level (Figures 8.2 and 8.5) and not at the decision level as the existing systems for human-affect analysis usually do (section 8.5 and/or Pantic and Rothkrantz 2001a).

Furthermore, each observation channel, in general, carries information at a wide range of time scales. At the longest scale are *static and semi-permanent signals* like bony structure, fat deposits, metabolism, and phonetic peculiarities like accent. Those signals provide a number of social cues essential for interpersonal communication in our everyday life. They mediate person identification, gender, attractiveness, and provide clues on a person's origin and health. At shorter time scales are *rapid behavioural signals* that represent temporal changes in neuromuscular and physiological activity that can last from a few milliseconds (e.g. a blink) to minutes (e.g. the respiration rate) or hours (e.g. sitting). Among the types of messages communicated by rapid behavioural signals are:

- affective states (e.g. joy, stress, inattention, frustration, etc),
- emblems (i.e. culture-specific symbolic communicators such as a wink or thumbs up),
- manipulators (i.e. manipulative actions used to act on objects in the environment or self-manipulative movements like scratching and lip biting),
- illustrators (i.e. actions accompanying and highlighting speech such as finger pointing and raised eyebrows), and
- regulators (i.e. conversational mediators such as the exchange of a look, palm pointing, head nods and smiles).

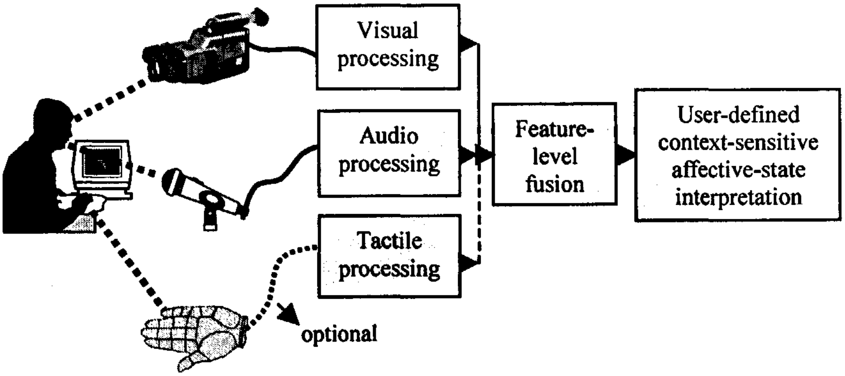


Figure 8.5: Architecture of an ideal automated affect-sensitive HCI tool

While rapid behavioural signals can generally be recognised from 40ms video frames and 10ms audio frames, the ability to discriminate subtle affective expressions requires a comparison over time. Namely, changes in human behaviour observed in a time instance may be misinterpreted if the temporal pattern of those changes is not taken into account. For example, a rapid frown of the eyebrows, denoting a difficulty with understanding discussed matters, could be misinterpreted for an expression of anger if the temporal pattern of behavioural changes, indicating attentiveness to the discussed subject, is not taken into account. In addition, performing both time-instance and time-scale analysis of the information carried by multiple observation channels can be extremely useful for handling sensory discordances and handling ambiguous input information in general. To wit, there is a certain grammar of neuromuscular actions and physiological reactions: only a certain subclass of these actions/reactions with respect to the currently encountered action/reaction (time instance) and the previously observed actions/reactions (time scale) is plausible. Thus, by considering the previously observed affective states (time scale) and the current information carried by functioning observation channels (time instance), a statistical prediction might be derived about the current affective state and the pertinent information, which have been lost due to malfunctioning or inaccuracy of a particular sensor.

Rapid behavioural signals can be subdivided further into the following classes:

- reflex actions under the control of afferent input (e.g. backward pull of the hand from a source of heat, scratching, squinting the eyes when facing the sun, etc.),
- rudimentary reflex-like (impulsive) actions that appear to be controlled by innate motor programs and might accompany affective expressions (e.g. wide-open eyes by encountering an unexpected situation) and less differentiated information processing (e.g. orienting in space),
- adaptable, versatile, culturally and individually variable spontaneous actions that appear to be mediated by learned motor programs and form a firm part of affective expressions (e.g. smile of greeting or at a joke, raised eyebrows in wonder, etc.), and
- malleable, culturally and individually variable intentional actions that are controlled by learned motor programs and form a part of affective expressions (e.g. uttering a spoken message by raising the eyebrows, shaking hands to get acquainted with someone, tapping the shoulder of a friend in welcome, etc.).

Thus, some classes of rapid behavioural signals demand relatively little of a person's information processing capacity and are free of deliberate control for their evocation, while others demand a lot of processing capacity, are consciously controlled and are governed by complex and culturally specific rules for interpersonal communication. Yet, while some rapid behavioural signals belong exclusively to one class (e.g. scratching), others may belong to any of the classes (e.g. squinted eyes). It is crucial to determine to which class a shown behavioural signal belongs since this influences the interpretation of the observed signal. For example, squinted eyes may be interpreted as sensitivity of the eyes if this action is a reflex, as an expression of hate if this action is displayed unintentionally, and as an illustrator of friendly anger on friendly teasing if this action is displayed intentionally. To determine the class of an observed rapid behavioural signal and to interpret it in terms of affective/attitudinal states, one must know the context in which the observed signal has been displayed. In other words, it is necessary to know the attitudinal interpretation of the observed behavioural signals that the expresser himself associates with those signals in the given situation (i.e. expresser's current environment and task; see also the discussion on context-sensitive HCI given in section 8.2).

In summary, ideal automated human affect analysers will perform (Figure 8.5):

- fully automatic, robust (despite auditory noise, the frailness of skin sensors, occlusions and changes in viewing and lighting conditions), generic (independent of variability in subjects' physiognomy, sex, age and ethnicity) time-instance and time-scale analyses of facial expressions, vocal intonations, and physiological signals, and

- user- and context-dependent affect-discriminative interpretation of the sensed data, previously combined by applying a multi-sensory feature-level data fusion.

8.4 Affect-sensitive HCI: The state of the art

This section will survey the state of the art in the affect-sensitive-monitoring problem domain. Rather than presenting an exhaustive survey, this section focuses on the efforts recently proposed in the literature that had the greatest impact on the community (as measured by, e.g., coverage of the problem domain, citations and received testing).

Relatively few of the existing works combine different modalities into a single system for human affective state analysis. Examples are the works of Chen et al. (1998) and De Silva & Ng (2000), who studied the effects of a combined detection of facial and vocal expressions of affective states. Other existing studies treat various human communicative signals separately and present approaches to automatic uni-modal human-affect analysis.

After careful literature research, only a single work aimed at automatic analysis of affective physiological signals was found: the work presented in (Healey and Picard 1998) and in (Vyzas and Picard 1999). In this work automatic recognition of 8 user-defined affective states (neutral, anger, hate, grief, platonic love, romantic love, joy and reverence) is reported. Data have been collected over a period of 32 days from an actress intentionally expressing 8 affective states during daily sessions (data obtained in 20 days have been used for further experiments). Five physiological signals have been recorded: EMG from jaw (coding the muscular tension of the jaw), blood volume pressure (BVP), skin conductivity, respiration, and hart rate calculated from the BVP. A total of 40 features has been used: 30 statistical features (for each raw signal they calculated 6 statistical features such as mean and standard deviation) and 10 features like mean slope of the skin conductivity, heart rate change, and power spectral density characteristics of the respiration signal. For emotional classification, the authors used an algorithm that combines the Sequential Floating Forward Search and the Fisher Projection that achieves an average correct recognition rate of 81.25%. As reported in (Vyzas and Picard 1999), the features extracted from the raw physiological signals were highly dependent on the day the signals were recorded. Also the signals have been recorded in short (3 minutes) sessions (Healey and Picard 1998).

For these reasons, this survey is divided into two parts. The first is dedicated to the work done on the automation of facial affect analysis. Since an exhaustive overview of the past work on this topic has been provided in chapter 2, this section only discusses the main issues relative to the automation of facial affect analysis. The emphasis will lie on enumerating advantages and disadvantages of ISFER, which represents the automated facial-affect analyser presented in this thesis. The second part of this section explores and compares automatic systems for affective-state recognition from audio input.

Automatic facial affect analysis

Facial expressions are our primary means for communicating emotion. In addition, human face-to-face interaction is inherently natural and substantial evidence suggests that this may also be true for human-computer interactions (section 8.2, Marsic et al. 2000, Schiano et al. 2000). These findings, together with recent advances in image analysis and pattern recognition, produced an upsurge of interest in the automatic recognition of facial affect. For exhaustive surveys, the reader is referred to: Samal and Iyengar (1992) for an overview of early works, Donato et al. (1999) for a review of techniques for detecting micro facial actions (AUs), chapter 2 or Pantic and Rothkrantz (2000d) for a survey of current efforts.

The problem of affect-sensitive monitoring of facial expressions includes three sub-problem areas: finding faces, detecting facial features, and classifying these data into some affect classes.

The problem of finding faces can be viewed as a segmentation problem (in machine vision) or as a detection problem (in pattern recognition). Possible strategies for face detection vary a lot, depending on the type of input images. The existing systems for facial expression analysis process either *facial* image sequences or static *facial* images. In other words, current studies assume, in general, that the presence of a face in the scene is ensured. Posed portraits of faces (uniform background and good illumination) constitute input data processed by the majority of the current systems. Yet, in many instances, the systems do not utilize a camera mounted on the subject's head, as proposed in this thesis or by Otsuka and Ohya (1998), which ascertains correctness of that assumption. Except the works reported by Essa and Pentland (1997) and Hong et al. (1998), presently existing systems do not perform automatic face detection in an arbitrary scene.

The problem of facial feature extraction from input images may be divided into at least four dimensions: (i) are the features extracted in an automatic way, (ii) is temporal information (image sequence) used, (iii) are the features holistic (templates spanning the whole face) or analytic (spanning subparts of the face), and (iv) are the features view based (2D) or volume based (3D). Given this glossary, most of the proposed approaches to facial affect analysis in facial images are directed towards automatic, static, analytic, 2D facial feature extraction. Still, many of the proposed systems do not extract facial information in an automatic way (e.g. the system proposed by Chen et al. (1998)). Though the techniques for facial affect classification employed by these systems are relevant to the present goals, the systems themselves are of limited use for affect-sensitive monitoring as analyses of human communicative signals should be fully automatic and preferably achieved in real time (see also section 2.2). The approaches to automatic facial-data extraction utilised by the existing systems include analyses of:

- facial motion (e.g. systems proposed by Essa and Pentland (1997), Otsuka and Ohya (1998), De Silva and Ng (2000), etc.),
- holistic spatial pattern (e.g. the system proposed by Hong et al. (1998)),
- analytic spatial pattern (e.g. ISFER proposed in this thesis).

In many instances strong assumptions are made to make the problem of facial feature detection more tractable (e.g. images contain portraits of faces with no facial hair or glasses, the illumination is constant, the subjects are young and of the same ethnicity). Few of the existing systems deal with rigid head motions (an example is the system proposed by Hong et al. (1998)) and only the method proposed by Essa and Pentland (1997) deals with the images of faces with facial hair and glasses. None of the automated facial-affect analysers proposed in the literature up to date "fills in" missing parts of the observed face, that is, none "perceives" a whole face when a part of it is occluded (e.g. by a hand or some other object). Also, though the conclusions generated by an automated facial expression analyser are affected by input data certainty, except for ISFER (see Table 8.2 for a summary of the system's advantages), none of the existing systems for automatic facial expression analysis calculates output data certainty based upon input data certainty.

Finally, automated facial expression analysers should terminate their execution by translating the extracted features into a description of a facial expression. This description should be identical, or at least very close, to a human's description of the examined facial expression. Yet, the actual interpretation of the examined expressions is constrained by the application domain of the system. For instance, an automated tool for facial expression analysis developed for the purposes of behavioural science investigations will probably code the facial actions in input images. On the other hand, an automated facial affect analyser should provide a description of the displayed facial affect. Once more, exactly which affective/ attitudinal states the system should recognise will depend on both its current user and the application domain. If the intended application is, for example, monitoring of a nuclear-power-plant operator, then the facial affect analyser to be deployed will probably be aimed at discerning stress and inattention. Also, if the current user distinguishes frustration, stress, and panic as variations of the generic category "stress", then the facial affect analyser should adapt to these interpretation categories. Overall, none of the currently existing automated facial expression analysers can classify facial expressions into all 44 facial action categories defined in FACS (Ekman and Friesen 1978), that is, none is capable of encoding the full range of facial behaviour. Besides the system proposed by Cohn et al. (1998) that recognises eight AUs and seven combinations of AUs, ISFER is the only system capable of automatic recognition of 29 AUs (i.e. 32 AUs codes, Table 5.8). Further, except for ISFER, none of the reported automatic facial expression analysers quantifies the facial action codes. Finally, except of ISFER, which classifies facial expressions into multiple quantified user-defined interpretation classes, the existing facial expression analysers perform facial expression classification into a number of the six basic emotion categories defined by Ekman and Friesen (1975).

Table 8.2
The advantages of ISFER as an automated human affect analyser

Advantage	Comment
Fully automated	The whole processing, from image acquisition through facial feature extraction to facial expression classification into multiple quantified facial action (AU) categories and user-defined interpretation classes, proceeds in an automatic way.
Uses head-mounted camera	The required presence of the face in an examined image is ensured; undesirable rigid head movements cannot be encountered (images acquired during a single session are scale and pose invariant).
Handles noisy data	The Facial Action Encoder part of ISFER deals with redundant, inaccurate, and partial data generated by the Facial Data Extractor part of ISFER. Input data certainty is propagated through the system and output data certainty is calculated accordingly (similar performance has not been reported in the literature up to date).
Distinguishes 29 AUs	The largest coverage of distinct facial action categories reported in the literature up to date.
Quantifies AU codes	The Facial Action Encoder part of ISFER assigns an intensity level to each encoded AU code in a subject-adaptive manner (similar performance has not been reported in the literature up to date).
User adaptable (see also the preceding point)	The Facial Expression Classifier is the CBR-implemented learning facility of ISFER that classifies input data into user-defined interpretation categories. The user is allowed complete freedom in associating various meanings with various expressions (similar performance has not been reported in the literature up to date).
Independent upon psychological debates	See the preceding point (similar performance has not been reported in the literature up to date).
Results in multiple quantified interpretation labels	The Facial Expression Classifier part of ISFER assigns an intensity level to each scored interpretation label by calculating the effect the intensity of facial muscle actions exhibits on the quantitative rating of the interpretation label associated with the facial expression caused by those facial actions (similar performance has not been reported in the literature up to date). This facilitates interpretation of blended facial expressions.
Portable, easy to enlarge, interactive, user-friendly	ISFER is a Java-implemented automated facial expression analyser.

Table 8.3
The disadvantages of ISFER as an automated human affect analyser

Disadvantage	Comment
Uni-modal	Insensitive to human affective displays that can be perceived by sound and touch modalities and, consequently, unable to handle sensory malfunctioning.
Uses head-mounted camera	The device (Figure 4.2) is heavy (uncomfortable) and reduces the freedom of movement; the user is asked to move slowly since a quick head movement may cause a displacement of the device ⇒ undesirable changes in viewing conditions ⇒ false conclusions. Yet, the current advance of sensing technology (Perry 2001) makes this problem less and less significant.
Does not deal with occlusions	Does not "fill in" missing parts of the observed face, that is, does not "perceive" a whole face when a part of it is occluded (e.g. by a hand). Also, ISFER cannot handle distractions like glasses and facial hair (beard, moustache).
Distinguishes 29 AUs	Does not perform encoding of the full range of facial behaviour (i.e. of all 44 AUs defined in FACS).
Does not perform a temporal analysis	Performs analysis of static imagery. While each frame of a video sequence can be analysed as a static image and then combined in order to perform time-scale analysis, ISFER is not capable of combining the information from several static images. Hence, it performs time-instance interpretation of facial behaviour ⇒ false conclusions may be derived due to insensitivity to the temporal course of monitored facial behaviour.
Context-insensitive spatial analysis	Insensitive to environmental constraints and temporal course of monitored facial behaviour ⇒ unable to devise subject-dependent context-sensitive grammar of facial behaviour ⇒ cannot distinguish intentionally displayed facial expressions ⇒ false conclusions may be generated.
Time-consuming processing	Due to time-consuming execution of some facial feature detectors integrated into the Facial Data Extractor part of ISFER as well as due to time-consuming execution of Java code.

The classification techniques used by the existing systems include:

- template-based classification in static images (e.g. Hong et al. (1998)),
- template-based classification in image sequences (e.g. the systems proposed by Essa and Pentland (1997), Otsuka and Ohya (1998), etc.),
- ANN-based classification in static images (e.g. Zhang et al. (1998)),
- rule-based classification in static images (e.g. Chen et al. (1998)),
- CBR-based classification in static images (ISFER proposed in this thesis)
- rule-based classification in image sequences (e.g. De Silva and Ng (2000)).

Given that humans detect six basic emotional expressions with an accuracy ranging from 70% to 98% (Bassili 1978), it is rather significant that the automated systems achieve an accuracy of 74% to 98% when detecting 3-7 emotions deliberately expressed by 8-40 subjects (chapter 2 and/or Pantic and Rothkrantz 2000d). Nevertheless, none of the automated facial-affect analysers proposed in the literature up to date performs a context-sensitive temporal analysis at longer time scales. Inter-frame analyses are only used to handle the problem of partial data, that is, to substitute missing data by the relevant data extracted from a preceding frame. The reported research did not yet contain a method for devising a grammar of context-dependent facial behaviour, which could be used to handle the problem of observed facial-expression intentionality that is crucial for generating a more accurate affective interpretation of facial changes observed in a time instance and over a time scale (see section 8.3 for a detailed discussion).

Given this overview of the state of the art in the facial-affect-sensitive monitoring problem domain, the advantages and limitations of the research results expounded in this thesis are summarised in Tables 8.2 and 8.3. More detailed discussions on the advantages and disadvantages of ISFER are available in sections 4.4, 5.7, and 6.6.

Automatic vocal affect analysis

In contrast to spoken language processing, which has witnessed significant advances in the last decade (Juang and Furnai 2000), the processing of "emotional" speech has not been widely explored by the auditory research community. However, recent data show that the accuracy of automated speech recognition, which is about 80-90% for neutrally spoken words, tends to drop to 50-60% if it concerns emotional speech (Steeneken and Hansen 1999). Although such findings triggered some efforts at automating vocal affect analysis, most researchers in this field have focussed on synthesis of emotional speech (Murray and Arnott 1996).

The problem of vocal affect analysis includes two sub-problem areas: specifying auditory features to be estimated from the input audio signal, and classifying those data into some affect classes.

The research in psychology/psycholinguistics provides an immense body of results on acoustic and prosodic features which encode the affective state of a speaker (e.g. Frick 1985, Schrer and Banse 1996). These studies point to the *pitch* as the main vocal cue for affective-state recognition in speech. Most of the works on automating affect-sensitive analysis of vocal expressions presented in the literature up to date use this finding and estimate the pitch of the input audio signal. Other acoustic and prosodic features used in the existing works are:

- *intensity* (i.e. the vocal energy, power) (Tosa and Nakatsu 1996, Amir and Ron 1998, Li and Zhao 1998, Chen et al. 1998, Petrushin 2000, Kang et al. 2000, Nakatsu et al. 2000, Polzin 2000),

- *slope* (Dellaert et al. 1996, Tosa and Nakatsu 1996, Li and Zhao 1998, Petrushin 2000, Polzin 2000),
- *temporal features* like the speaking rate (Dellaert et al. 1996, Tosa and Nakatsu 1996, Amir and Ron 1998, Petrushin 2000),
- *derivate features* such as the smoothed pitch contour and its derivatives (Dellaert et al. 1996),
- *phonetic features* like the signal's LPC (linear predictive coding) parameters (Tosa and Nakatsu 1996),
- *supra-segmental features* such as the intensity and pitch over the duration of a syllable, word or sentence (Li and Zhao 1998, Polzin 2000).

Virtually all of the existing work on automatic vocal affect analysis performs singular classification of input audio signals into a few "basic" emotion categories such as anger, happiness, sadness/grief, fear, disgust, surprise, affection and irony (Cowie et al. 2001). Utilised classification techniques include:

- K-nearest neighbours (Dellaert et al. 1996, Petrushin 2000, Kang et al. 2000),
- Hidden Markov Models (De Silva & Ng 2000, Kang et al. 2000, Polzin 2000),
- Gaussian mix density models (Li and Zhao 1998),
- Rule-based approach (Chen et al. 1998),
- Fuzzy membership indexing (Amir and Ron 1998),
- Maximum-Likelihood Bayes (Kang et al. 2000),
- Artificial Neural Networks (Tosa and Nakatsu 1996, Li and Zhao 1998, Petrushin 2000, Nakatsu et al. 2000).

In general, people can recognize emotion in a neutral-content speech with an accuracy of 55-70% when choosing from among six basic affective states (Bezooijen 1984). Automated vocal-affect analysers match this accuracy when recognizing 4-8 emotions deliberately expressed by 2-100 subjects recorded while pronouncing sentences having a length of 1-12 words.

Nevertheless, in many instances strong assumptions are made to make the problem of automating vocal-expression analysis more tractable (e.g. the recordings are noise free; the recorded sentences are short, delimited by pauses, and carefully pronounced to express the required affective state; subjects are non-smoking professional or non-professional actors). Only one of the existing automated vocal affect analysers, i.e. (Petrushin 2000), has been tested on "almost" real-world data composed of short telephone messages spoken by 18 non-professional actors expressing mainly neutral and angry vocal affects (recognition rates reported are 73-77%). Overall, the testing data sets are small (5-50 sentences spoken by few subjects) containing exaggerated vocal expressions of affective states. Hence, the state of the art in automatic affective state recognition from speech is similar to that of speech recognition several decades ago when computers could classify the carefully articulated digits spoken with pauses in between, but could not accurately detect these digits if they were spoken in a way not previously encountered and forming a part of a longer continuous conversation.

8.5 Affect-sensitive HCI: Challenges and opportunities

The limitations of the existing affect-sensitive monitoring tools are probably the best place to start a discussion about the challenges and opportunities that researches of affective computing face. The issue that strikes and surprises me most is that, though the recent advances in video and audio processing make automatic *bi-modal* affect-sensitive monitoring a remarkably tractable problem and though all agreed that solving this problem would be extremely useful, merely two efforts (i.e. Chen et al. 1998, De Silva and Ng 2000) aimed at an actual implementation of such a bi-modal tool have been reported in the literature up to date. There is no record of a research endeavour towards inclusion of all non-verbal modalities into a single system for affect-sensitive monitoring of human behaviour. Besides the problem of achieving a deeper integration of detached visual, auditory and tactile research communities, there are a number of related issues.

Visual input

As already remarked in section 8.3, the acquisition of video input for an affect-sensitive monitoring HCI tool concerns, at least, the detection of a monitored subject's face in the observed scene (if not of the upper part of body as well). The problematic issue here, typical for all visual sensing including gaze, lip-movement, body and facial gesture tracking, is that of scale, pose and occlusion. Namely, in most real-life situations it cannot be assumed that the subject will remain immovable; rigid head and body motions can be expected, causing changes in the viewing angle and in the visibility and illumination of the tracked facial and body features. Although highly time-consuming, the scale problem can be solved by forming a multi-resolution representation of input image/frame and performing the same detection procedure for different resolutions (e.g. Viennet and Soulie 1992). Pose and occlusion are more difficult problems, initially thought to be intractable or at least the hardest to solve. However, interesting progress is being made in machine vision research. The focus of active vision field on *foveal purposeful vision* is the development of special sensors, which serve a specified purpose and are based on the principle of the human-eye fovea in the sense that they can pan and zoom on relatively small regions of the scene that contain critical information. Further, statistical methods have been developed that essentially try to predict the pose of monitored objects from whatever image information is available. Finally, methods for the monitored object's representations at several orientations, employing data acquired by multiple cameras, are currently thought to provide the most promising solution to the problems of pose and occlusion. For an extensive review of the methods for video surveillance, the reader is referred to Collins et al. (2000).

Besides these standard visual-processing problems, there is another issue which is typical for facial image processing: the "universality" of the employed technique for detection of the face and its features. Namely, the employed detection method must not be prone to the physiognomic variability and the current looks of monitored subjects. As explained in section 8.3, an ideal automated affect-sensitive monitoring tool should perform generic analyses of the sensed facial information, independently of possibly present static facial signals such as wrinkles and artificial facial signals like glasses and make-up. Essa and Pentland (1997) proposed such a method.

Audio input

As already remarked in section 8.4, virtually all of the work done on automating vocal affect analysis assumes a fixed listening position, a closely placed microphone, non-smoking subjects, and noise-free recordings of short sentences that are delimited by pauses and carefully pronounced to express the required affective state. Such a clean audio input is not realistic, nevertheless, especially not in unconstrained environments characteristic for most HCI applications and ubiquitous computing in general. A way of enhancing the state of the art in vocal affect analysis is to explore existing methods for human language processing and to employ the most prominent pattern-recognition methods that minimise the classification error rate. Excellent reviews of the existing methods for spoken language processing can be found in (Juang and Furui 2000).

Another intriguing issue is the kind of features that should be adopted in order to achieve robust vocal affect recognition from speech. The psycholinguistic research results are consistent, in general (Cowie et al. 2001), on some speech correlates of a few "basic" emotions like anger (pitch increase in mean, median, and range, intensity raised), fear (pitch increase in mean and range, intensity normal), happiness (pitch increase in mean and range, intensity increased), and sadness (pitch decrease in mean and range, intensity decreased). Yet, there are many contradictory reports on some other speech correlates of the very same "basic" emotions. For example, there is a disagreement on duration facets of anger, happiness, and fear – some report a longer duration (e.g. Williams and Stevens 1972, Bezooijen 1984) and some report a faster speech rate (e.g. Fonagy 1978, Coleman and Williams 1979). Furthermore, while some adhere to the standpoint that the features should be solely prosodic and different from the phonetic features used for speech recognition, others adhere to the standpoint that prosodic and phonetic features are tightly combined when uttering speech (i.e. it is impossible for us to express and recognize vocal affects by considering only prosodic features). The latter is experimentally proven: observers who did not speak the Sinhala language correctly recognised six different emotions in Sinhala spoken speech with an average of 32.3% (De Silva et al. 1997).

Another interesting observation is that the information encoded in the speech signal becomes far more meaningful if the pitch and intensity can be observed over the duration of a syllable, word, or phrase (Izzo 1998, Polzin 2000). Although quite sophisticated techniques are available to locate pauses between phrases (e.g. McKinley and Whipple 1997), the detection of boundaries between words and phonemes still forms a significant challenge in speech processing (Cowie et al. 2001). For researchers of automatic vocal affect analysis this summary of currently nagging problems suggests investigation of a robust, speaker-independent, temporal analysis of phonetic and prosodic characteristics of spontaneous affective speech.

Multi-modal input

The goal of automatic multi-modal monitoring of human affective states is to achieve generic time-instance and time-scale analyses of audio, visual, and tactile human communicative signals. An ideal human-affect analyser (Figure 8.5) should generate a reliable result based on multiple input signals acquired by different sensors. Let us consider these issues in more detail.

If we consider the state of the art in audio, visual, and tactile processing, noisy and partial input data should be expected. An (ideal) affect-sensitive monitoring tool should be able to deal with these imperfect data and generate its conclusion so that the certainty associated with it varies in accordance with the input data (section 8.3). A way of achieving this is to consider the time-instance vs. time-scale dimension of human paralanguage. Namely, as already explained in section 8.3, there is a certain grammar of neuromuscular actions and physiological reactions: only a certain subclass of these actions/reactions with respect to the currently encountered action/reaction (time instance) and the previously observed actions/reactions (time scale) is plausible. If the current input data affirm these statistically predicted actions/reactions, the certainty associated with that data should be "high" and the certainty of the drawn conclusion is to be computed accordingly. Nevertheless, such a temporal analysis involves untangling of the grammar of human behaviour, which is a rather unexplored topic even in the psychological and sociological research areas. The issue that makes this problem even more difficult to solve in a general case is the dependency of a person's behaviour on his/her personality, cultural and social vicinity, current mood, and the context (situation) in which the observed behavioural cues were encountered. One source of help for these problems is machine learning: rather than using a priori rules to interpret human behaviour, we can potentially learn application-, user-, and context-dependent rules by watching the user's behaviour in the sensed context. Though context sensing and the time needed to learn appropriate rules are significant problems in their own right, many benefits could accrue from such an adaptive affect-sensitive HCI tool (section 8.2, Pentland 2000).

Another typical issue of multi-modal data processing is that the multi-sensory data are processed separately and only combined at the end (section 8.2, Sharma et al 1998). The system proposed by de Silva and Ng (2000) is an example. Nevertheless, this is almost certainly incorrect; people display audio, visual, and tactile communicative signals in a complementary and redundant manner. Chen et al. (1998) have proven this experimentally for the case of audio and visual input. In order to achieve a multi-modal analysis of multiple signals acquired by different sensors, which will resemble human recognition of affective states, the input signals cannot be considered mutually independent and cannot be combined at the end of the intended analysis (section 8.3). The input data should be processed in a joint feature space. In practice, however, there are two major difficulties: the size of the required joint feature space, which is usually huge and results in a heavy computational burden, and different feature formats and timing. A way to deal with these problems and to achieve tightly-coupled multi-sensory data fusion is to apply a Bayesian inference method as proposed by Pan et al. (1999). Yet, the complexity of the phenomena and a general lack of researchers having expertise in all domains (audio, visual, and tactile processing), make the problem of joint audio-visual-tactile human-affect analysis a significant challenge facing researchers of multi-modal human-affect analysis and multi-modal HCI in general (section 8.2).

Affect-sensitive interpretation of multi-modal input data

Currently existing methods aimed at the automation of human-affect analysis are not context sensitive. Yet, the interpretation of human communicative signals is strongly situation dependent (Russell 1994, Russell and Fernandez-Dols 1997). Although it was initially thought that it would be the research topic that would be the hardest to solve, context sensing, that is, answering questions like who is the user, where is he, and what is he doing, has been proven remarkably tractable (section 8.2, Pentland 2000). However, the complexity of this wide-ranging problem and a general lack of researchers having the necessary expertise, make the problem of context-sensitive human-affect analysis perhaps the most significant research challenge.

Another issue concerns the actual interpretation of human communicative signals in terms of affective/attitudinal states. Almost all of the existing work employs singular classification of input data into one of the six basic emotion categories (section 8.4). This approach has many limitations. As explained in sections 6.2 and 8.3, the theory on the existence of six universal emotion categories is nowadays strongly challenged in the psychological research area. Further, as noted by the inventor of this theory himself, pure expressions of basic emotions are seldom elicited (Ekman 1982); most of the time people show blends of emotional displays. Hence, the classification of human communicative signals into a single basic-emotion category is not realistic. Affect-sensitive analysers of sensed human communicative signals must at least realise quantified classification into multiple emotion categories, for example, as proposed by Pantic and Rothkrantz (2000b) or by Zhang et al. (1998) for automatic facial affect analysis. Yet not all human communicative displays can be classified as a combination of the six basic emotion categories. Think for instance about the frustration, stress, boredom, or "I don't know" attitudinal states. Besides, it has been shown that the comprehension of a given emotion label and the ways of expressing the related affective state differ from culture to culture (sections 6.2 and 8.3). Hence, the definition of interpretation categories in which any set of displayed human communicative signals can be classified is a key challenge in the design of realistic affect-sensitive monitoring tools. The lack of psychological scrutiny on the topic makes this problem even harder. One source of help for this problem is (again) machine learning: instead of integrating rigid generic rules for the interpretation of human communicative behaviour into the intended tool, the system can potentially learn its own expertise by allowing the user to define his own interpretation categories (e.g. as proposed in chapter 6 for an automated facial affect analyser). As already remarked, such an adaptive (user-, application-, and context-profiled) affect-sensitive HCI tool would be an ideal tool for understanding human behaviour that could greatly enhance the state of the art in HCI.

ISFER: Future enhancements

The limitations of ISFER presented in this thesis (Table 8.3) are probably the best place to start a discussion about the challenges that future developers of the system face.

As already discussed, ISFER does not perform a temporal analysis of human facial expressions. It has been developed to perform quantified facial action coding and facial expression classification into multiple quantified user-defined interpretation labels from static facial images rather than from facial image sequences of the observed subject. As remarked in section 8.4, in a case like ISFER's, the problem of performing a temporal analysis of an input facial image sequence can be tackled by analysing each frame of a video sequence as a static image and then combining the acquired information in order to perform a time-scale analysis. Yet, to accomplish this, ISFER must first be enabled to combine the information from several static images. Currently, however, ISFER performs a time-instance interpretation of facial behaviour, that is, it analyses automatically facial expressions in a way that is insensitive to temporal course of monitored facial behaviour. To include the time dimension in the facial expression analysis is the first challenge that future developers of ISFER face. This could enhance many facets of the current performance of the system (Table 8.4, sections 5.7 and 6.6).

ISFER is an automatic, uni-modal, user-adaptable, affect-sensitive analyser of human communicative behaviour. It interprets affective states shown by the currently observed subject (in terms of multiple quantified interpretation labels learned from the user) based only on the sensed facial expressions. In other words, ISFER is insensitive to a subject's affective vocal and physiological displays. While the integration of the touch modality into ISFER could introduce additional drawbacks to the system (due to uncomfortable wiring of the monitored subject and the frailness of the currently existing skin sensors), the incorporation of a sound modality into ISFER could introduce many benefits (Table 8.5). Enabling ISFER to sense, recognise and interpret the subject's facial and vocal affect (i.e. turning it into a bi-modal user-adaptable affect-sensitive monitoring tool) is the second challenge that developers of future ISAR (Integrated System for Affect Recognition) face.

Although automatic context sensing is crucial if a more accurate automatic facial affect analysis is to be achieved (Table 8.6), affect-sensitive monitoring of human facial behaviour performed by ISFER is context insensitive. Consequently, ISFER cannot adapt to the currently monitored subject automatically (i.e. it cannot identify the current subject automatically), it interprets shown facial expressions independent of the situation, and it cannot distinguish between intentionally and unintentionally displayed facial cues. In turn, turning ISFER into a fully automated bi-modal user-adaptable context-sensitive human-affect analyser is the third challenge that developers of future ISAR face. Yet, the complexity of both the problem of context sensing (section 8.2, Pentland 2000) and the problem of untangling user and context dependent meanings of joint audio-visual human communicative signals, sets the actual realisation of ISAR in relatively distant future.

Table 8.4
Benefits that could accrue if a temporal aspect is associated with the current spatial aspect of the facial expression analysis performed by ISFER

Benefit	Comment
Relaxing the constraints on the subject's looks	If the automated expression analysis is based on a facial motion tracking method, facial hair, birthmarks, unibrow, and artificial facial signals like glasses would not form a cumbrous source of noise (Simoncelli 1993).
Encoding a wider range of facial actions (current range given in Table 5.8)	If a method for estimating the motion in various facial areas is applied, brief muscle actions like blinking (AU45), winking (AU46) and wiping the lips (AU37) and the muscle actions that involve conspicuous facial movements like moving the jaw sideways (AU30), producing a bulge by pushing the tongue into the cheeks (AU36l, AU36r), and clenching the jaw (AU31) will be detectable.
Encoding a wider range of facial affects	Psycho-physiological states like hypertension, stress, pain and frustration are all characterised by a certain alteration in facial expressions, but it is rather the temporal dynamics of expressions (a certain pattern of expressions observed over a time scale) than the configuration aspects of an expression encountered in a time instance that make those states recognisable.
Dealing with occlusions and inaccurate (noisy) input data in general	If the information about the spatial and the temporal course of an encountered facial action and the temporal course of monitored affective behaviour is available, statistical predictions could be made on the current facial appearance changes (which were uncertainly discerned due to imperfect input data) based upon: the affective state expected to be displayed, the pertinent facial actions and their current time markers (onset, apex, offset).

Table 8.5
Benefits that could accrue from turning ISFER into an automated bi-modal user-adaptable affect-sensitive HCI tool for monitoring human facial and vocal communicative displays

Benefit	Comment
Encoding a wider range of affective states	The recognition of psycho-physiological states like hypertension, stress, and frustration, as well as of moods and traits is more tractable if the temporal dynamics of both facial and vocal cues are considered in combination (Oatley and Jenkins 1996).
Increased reliability	Ambiguities about the observed facial affect can be resolved based upon the sensed vocal correlates and vice versa. If the data sensed by the two modalities affirm each other and the statistically predicted affective state (based upon the knowledge on the temporal courses of the current subject's behaviour), the confidence in data, and in turn in conclusions, can be increased.
Enabling sensory discordances to be handled	Malfunctioning of a sensor would not form an untreatable problem: the analysis of the subject's affective states could proceed based upon the information perceived by the other modality.

Table 8.6
Benefits that could accrue from making ISFER more sensitive to context in which it acts

Benefit	Comment
Automatically adaptable to the current subject	If automatic person identification could be achieved, it would enable person-adaptable quantification of the displayed facial actions and automatic retrieval of person-dependent grammar of facial behaviour. Then, if the monitored subject and the current user are the same person, the system could adapt automatically to interpretations learned from the current user.
Automatically adaptable to the environment of the monitored subject	Automatic detection and tracking of environmental cues would enable better interpretation of sensed facial displays, which is strongly situation dependent (e.g. a frown could be interpreted as a speech utterance if shown by a speaker or as confusion if shown by a listener).
Automatically adaptable to the monitored subject's task	E.g., automatic detection and tracking of the monitored subject's gaze could facilitate detection of his/her current task and, in turn, task-dependent interpretation of his/her facial displays. E.g., wide-open eyes will mean surprise if shown while checking e-mail and fear if shown while checking the reason of a just sounded alarm.

8.6 Conclusions

The automation of user-profiled multi-modal context- and affect-sensitive monitoring and interpretation of human behavioural cues is likely to become the single most widespread research topic of the AI research community in general (Pentland 2000). The reason behind it is that untangling the problems related to this research topic is prerequisite for the design of the next generation perceptual interfaces and for ubiquitous computing in general. Yet, currently existing methods aimed at the automation of human-affect analysis are:

- uni-modal, except for the systems proposed by Chen et al. (1998) and De Silva and Ng (2000) that perform a joint audio-visual affect analysis,
- context insensitive, and
- user inadaptable, except for ISFER proposed in this thesis (Table 8.2).

In summary, although the fields of machine vision, audio processing, and affective computing witnessed rather significant advances in the past few years, the realisation of a fully automated, robust, multi-modal, adaptive, affect-sensitive analyser of human communicative cues still lies in a rather distant future. Besides the problems involved in the integration of multiple sensors and pertinent modalities according to the model of the human sensory system and the lack of a better understanding of individual- and context- dependent human behaviour in general, there are two additional related issues.

The first issue that jeopardises a future wide deployment of adaptive affect-sensitive multi-modal monitoring HCI tools proposed in this chapter concerns the efficiency of such HCI tools. Namely, since it is generally thought that embedded computing devices will be everywhere in the future, it will be inefficient if the user should train each of those devices separately. The computers of our future must know enough about the people and the environment in which they act to be capable of acting appropriately with a minimum of explicit instruction (Pentland 2000). A long-term way of achieving this is:

- to develop multi-modal affect-sensitive tools, as proposed here, which will be able to monitor human behaviour and to adapt to the current user (i.e. to who he is and to what the grammar of his behavioural actions/reactions is), to his context (i.e. to where he is and to what he is doing at this point), and to the application domain (e.g. if stress is observed by a nuclear power plant operator while he reads his e-mail is not cause for an alarm), then
- to make those self-adaptive tools commercially available to the users who will profile them in the context in which the tools are to be used, and finally
- to withdraw the trained systems after some time and combine the stored knowledge in order to derive generic statistical rules/models for interpretation of human behaviour in the given context/environment.

Though the unwillingness of people to participate in such a privacy-intruding large-scale project is a significant problem in its own right, this approach could resolve many intriguing questions. The most important is that this could resolve the social impact of interaction in electronic media, that is, the effects of computing and information technology on: our interpersonal interaction, overall related human behaviour, and our cultural and social vicinity.

Another issue that might jeopardise a future wide deployment of adaptive affect-sensitive multi-modal monitoring HCI tools proposed in this chapter concerns the design of such HCI tools. Computer technology and especially affect-sensitive monitoring tools might be perceived as "big brother is watching you" tools (see also section 7.5). As remarked by Schneiderman (1993b), a large proportion of the population would in fact be terrified by the vision of the universal use of computers in the coming era of ubiquitous computing. Therefore, the actual deployment of adaptive affect-sensitive multi-modal HCI tools proposed in this chapter will only be attainable if the design of those tools *will not*:

- invade the user's privacy (the proposed tools' capacity to monitor and concentrate information about human behaviour must not be misused),
- cause the user to worry about being unemployed (air-traffic or production controllers do not want machines that could replace them entirely, but that could help them in performing their job faster and potentially more accurately),
- reduce the user's professional responsibility (insisting on the "intelligent" capabilities of computing devices could have negative effects like blaming machines for our own poor performance or seeing machines as infallible devices instead of tools that can merely empower us by retrieving and processing information faster and more reliably).

In summary, if machines were given the ability to interpret human behaviour without explicit instruction, they would represent the coming of universally usable and accessible HCI systems and the means for determining the impact the information technology has on our social behaviour. However, we also should recognise that the realisation of this goal is still likely to lie in the relatively distant future and this goal will be attainable only if we properly design the HCI envisioned here such that it is trustworthy.

Appendix A: Data Flow Diagrams

This Appendix provides the reader with a detailed algorithmic representation of the processing of the Integrated System for Facial Expression Recognition (ISFER) presented in this thesis. It revisits the main goals for the design and development of ISFER and expounds Data Flow Diagrams (DFDs) of various parts of the system.

To recapitulate, the main goal in the design and development of ISFER was the enhancement of the state of the art in automated facial expression analysis (chapter 2); the aim was to develop a fully automated system for facial expression analysis that achieves the following:

1. Subject-independent, robust, fully automatic facial-expression-information extraction from a static (dual-view) facial image. The first part of ISFER, the Facial Data Extractor explained in detail in chapter 4, achieves this. The Facial Data Extractor is a framework for hybrid facial-feature detection, which applies multiple feature detectors to an input static facial image. The result of each detector, representing a spatial sampling of the contour of the relevant facial feature (one of the eyebrows, eyes, nose, mouth, chin and profile), is stored in a separate file. An algorithmic representation of the processing of the Facial Data Extractor is given in Figure A.2.
2. Robust, fully automatic facial expression recognition that is applicable to automated FACS coding and results in both generic multiple facial-action codes and observed-subject-dependent quantification of these. The second part of ISFER, the Facial Action Encoder explained in detail in chapter 5, achieves this. The Facial Action Encoder is a rule-based expert system that reasons with uncertainty about 32 AU-codes and their intensity levels shown in the currently examined facial image. To this end, it compares the input facial-expression data which are generated by the Facial Data Extractor and may be redundant, partial, and/or approximate with the data representing the expressionless face of the currently observed subject. An algorithmic representation of the processing of the Facial Action Encoder is given in Figure A.3.

3. Automatic facial expression analysis in terms of multiple quantified interpretation labels learned from the current user. The third part of ISFER, the Facial Expression Classifier explained in detail in chapter 6, achieves this. The Facial Expression Classifier is a memory-based expert system based upon Schank's theory on human autobiographical memory organisation (Schank 1984) and case-based reasoning (section 3.6). An algorithmic representation of the processing of the Facial Expression Classifier is given in Figures A.4 and A.5.

Legend for the symbols used in the Figures A.1 – A.5

Symbol	Stand for	Symbol	Stand for	Symbol	Stand for
□	Process	▢	Alternate Process	❐	End data destination
⌭	Direct Access Storage	○	Data-flow separator into several directions	⊕	Or
⬚	Optional Alternate Process	⬚	Optional Internal Storage	⟠	Sort
⧖	Collate	△	Extract	▽	Store

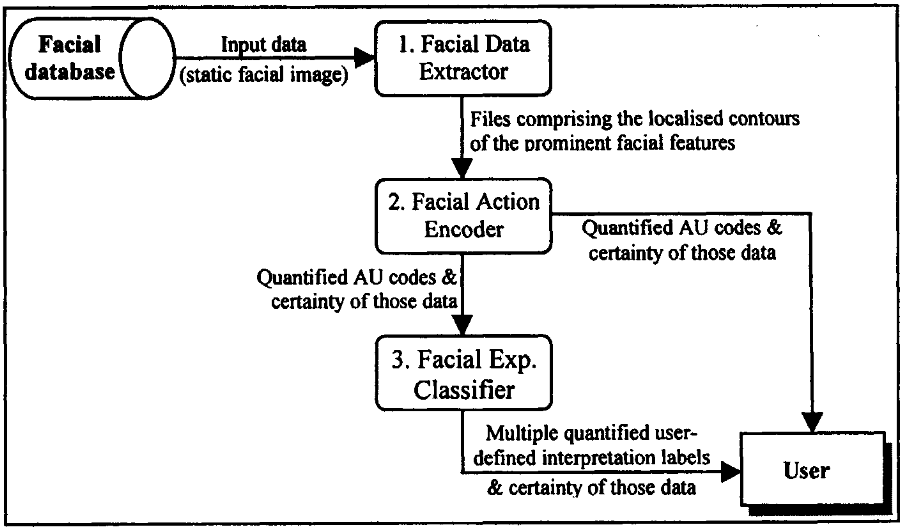

Facial database
Input data
(static facial image)
1. Facial Data Extractor
Files comprising the localised contours of the prominent facial features
2. Facial Action Encoder
Quantified AU codes & certainty of those data
Quantified AU codes & certainty of those data
3. Facial Exp. Classifier
Multiple quantified user-defined interpretation labels
& certainty of those data
User

Figure A.1: DFD of ISFER

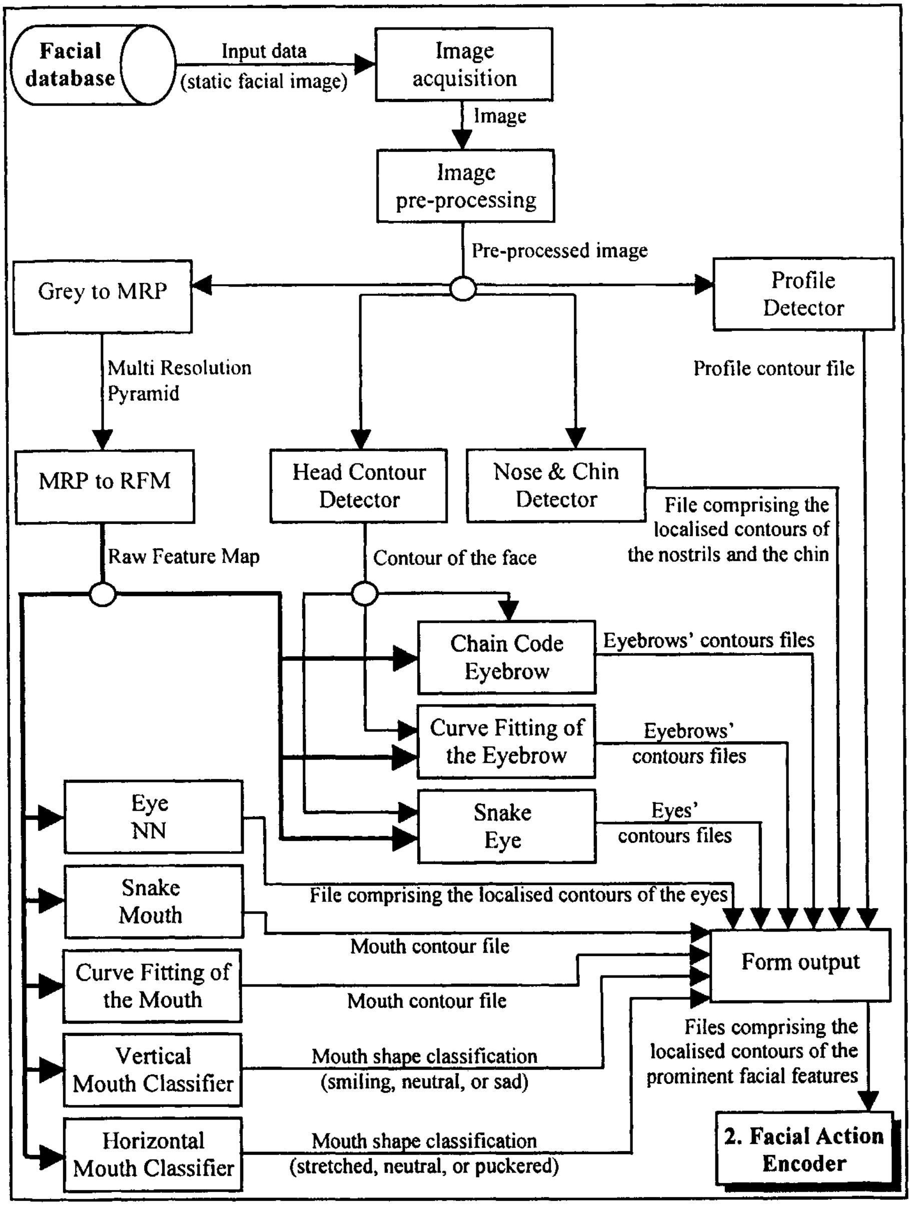


Figure A.2: DFD of the Facial Data Extractor

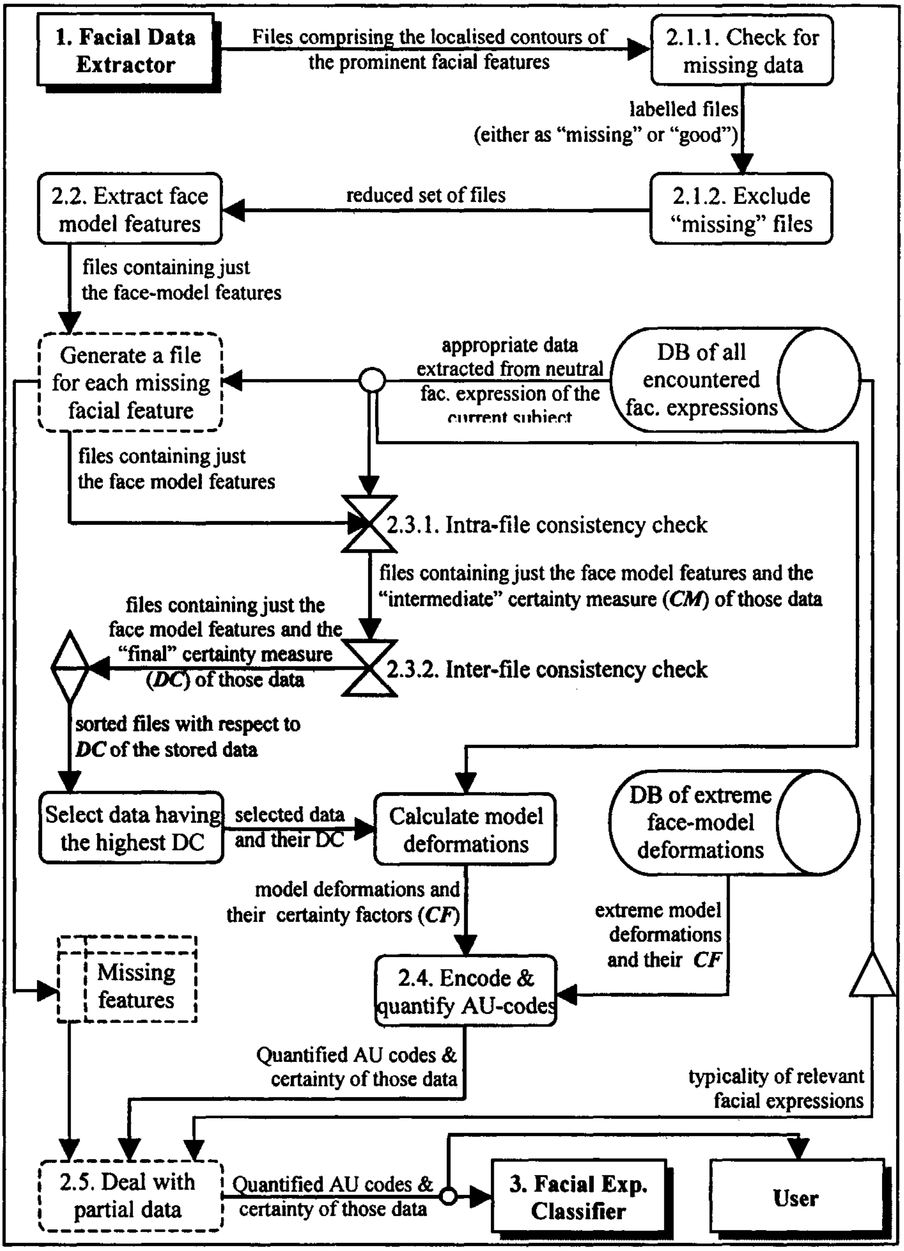

1. Facial Data Extractor
Files comprising the localised contours of the prominent facial features
2.1.1. Check for missing data
labelled files (either as "missing" or "good")
2.1.2. Exclude "missing" files
reduced set of files
2.2. Extract face model features
files containing just the face-model features
Generate a file for each missing facial feature
appropriate data extracted from neutral fac. expression of the current subject
DB of all encountered fac. expressions
files containing just the face model features
2.3.1. Intra-file consistency check
files containing just the face model features and the "intermediate" certainty measure (CM) of those data
files containing just the face model features and the "final" certainty measure (DC) of those data
2.3.2. Inter-file consistency check
sorted files with respect to DC of the stored data
Select data having the highest DC
selected data and their DC
Calculate model deformations
DB of extreme face-model deformations
model deformations and their certainty factors (CF)
extreme model deformations and their CF
Missing features
2.4. Encode & quantify AU-codes
Quantified AU codes & certainty of those data
typicality of relevant facial expressions
2.5. Deal with partial data
Quantified AU codes & certainty of those data
3. Facial Exp. Classifier
User

Figure A.3: DFD of the processing of the supervisor of the Facial Action Encoder part of ISFER

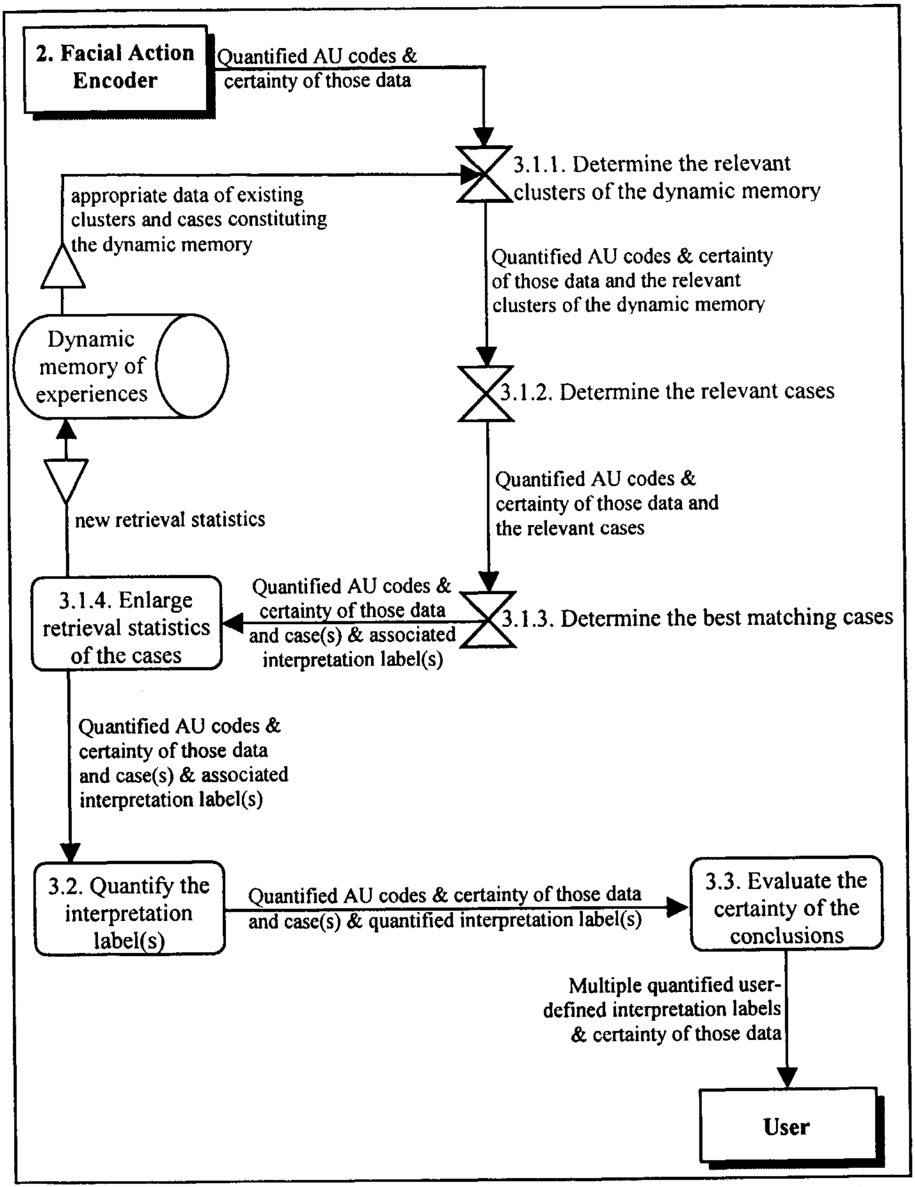

2. Facial Action Encoder
Quantified AU codes & certainty of those data
3.1.1. Determine the relevant clusters of the dynamic memory
appropriate data of existing clusters and cases constituting the dynamic memory
Quantified AU codes & certainty of those data and the relevant clusters of the dynamic memory
Dynamic memory of experiences
3.1.2. Determine the relevant cases
new retrieval statistics
Quantified AU codes & certainty of those data and the relevant cases
3.1.4. Enlarge retrieval statistics of the cases
Quantified AU codes & certainty of those data and case(s) & associated interpretation label(s)
3.1.3. Determine the best matching cases
Quantified AU codes & certainty of those data and case(s) & associated interpretation label(s)
3.2. Quantify the interpretation label(s)
Quantified AU codes & certainty of those data and case(s) & quantified interpretation label(s)
3.3. Evaluate the certainty of the conclusions
Multiple quantified user-defined interpretation labels & certainty of those data
User

Figure A.4: DFD of the processing of the supervisor in the interpret mode of the Facial Expression Classifier part of ISFER

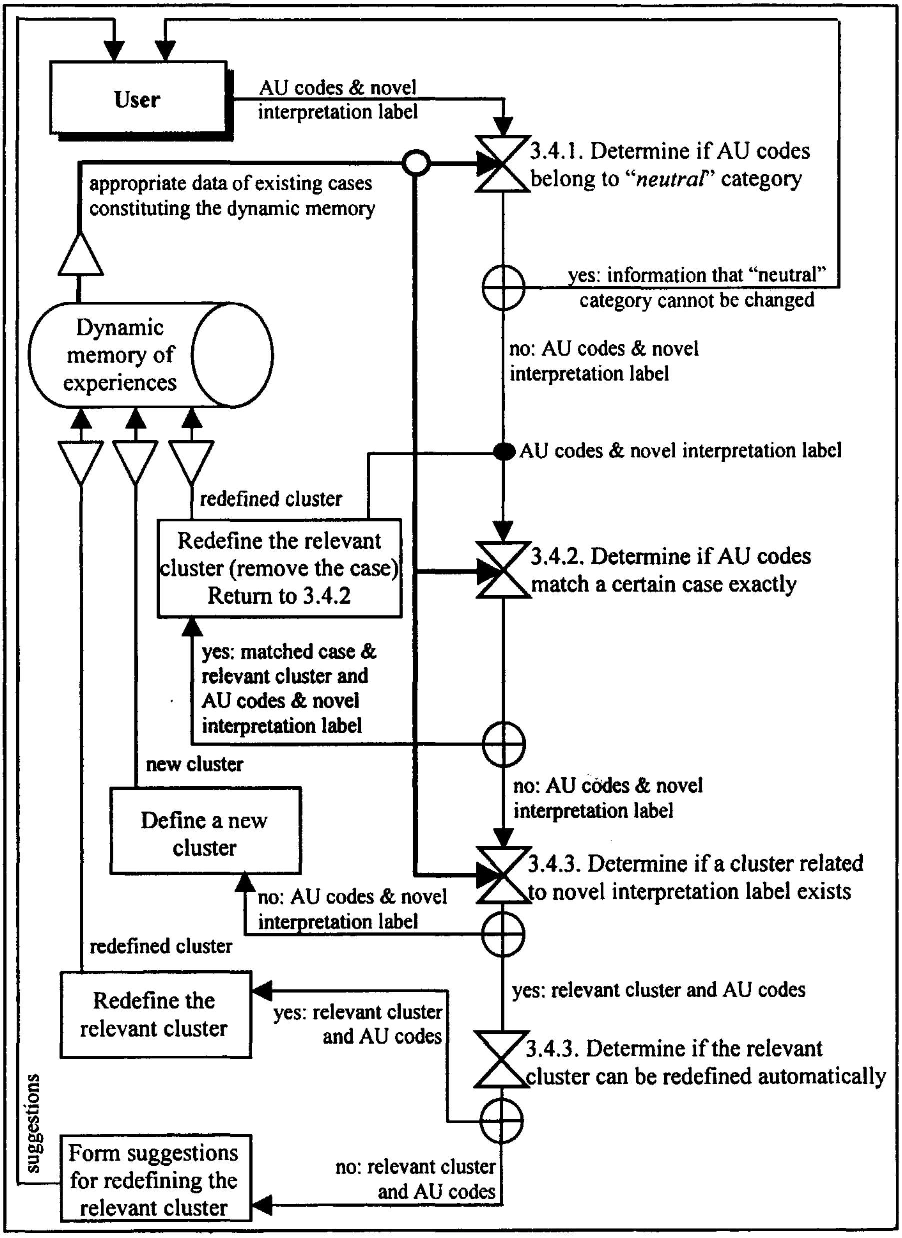

User
AU codes & novel interpretation label
3.4.1. Determine if AU codes belong to "neutral" category
appropriate data of existing cases constituting the dynamic memory
yes: information that "neutral" category cannot be changed
Dynamic memory of experiences
no: AU codes & novel interpretation label
AU codes & novel interpretation label
redefined cluster
Redefine the relevant cluster (remove the case) Return to 3.4.2
3.4.2. Determine if AU codes match a certain case exactly
yes: matched case & relevant cluster and AU codes & novel interpretation label
new cluster
no: AU codes & novel interpretation label
Define a new cluster
3.4.3. Determine if a cluster related to novel interpretation label exists
no: AU codes & novel interpretation label
redefined cluster
yes: relevant cluster and AU codes
Redefine the relevant cluster
yes: relevant cluster and AU codes
3.4.3. Determine if the relevant cluster can be redefined automatically
suggestions
Form suggestions for redefining the relevant cluster
no: relevant cluster and AU codes

Figure A.5: DFD of the processing of the supervisor in the learn mode of the Facial Expression Classifier part of ISFER

Appendix B: The AU-coding rules

As explained in chapter 5, the Facial Action Encoder part of ISFER encodes and quantifies the facial actions captured in a dual-view facial image of the currently observed subject based on the calculated dual-view face-model deformations and according to the mapping between 32 FACS rules and 32 dual-view face-model-based rules given in Tables 5.5, 5.7 and 5.8. A complete list of the utilised functions, variables and rules (defined in Tables 5.5, 5.7 and 5.8) and a R-list representation of the relations among these rules (i.e. a complete Figure 5.12) are provided here.

Functions

f-on 1	$\boldsymbol{sigm}(y;\alpha,\beta,\gamma)=\begin{cases} 1 & y\le\alpha \\ 1-2[(y-\alpha)/(\gamma-\alpha)]^2 & \alpha<y<\beta \\ 2[(y-\gamma)/(\gamma-\alpha)]^2 & \beta<y<\gamma \\ 0 & y\ge\gamma \end{cases}$
f-on 2	***extent (x)*** = 100 * ***sigm*** $(\lvert x\rvert;\ 0,\ ½\ \lvert x_{extreme}\rvert,\ \lvert x_{extreme}\rvert)$, where x is a particular current face model deformation computed by the supervisor of the Facial Action Encoder part of ISFER, $x_{extreme}$ is retrieved from the DB of extreme model deformations of the currently observed subject, and $\boldsymbol{sigm}(y;\ \alpha,\ \beta,\ \gamma)$ is a Sigmoid membership function (i.e. f-on 1).
f-on 3	$\boldsymbol{max}(x,y)=\begin{cases} x & x\ge y \\ y & x<y \end{cases}$
f-on 4	$\boldsymbol{min}(x,y)=\begin{cases} y & x\ge y \\ x & x<y \end{cases}$
f-on 5	***in/out(P)*** = $x_{Pneutral}$ - $x_{Pexamined}$, where $x_{Pneutral}$ is the x-coordinate of point P retrieved from the "neutral expression" file stored in the DB of extreme model deformations of the currently observed subject and $x_{Pexamined}$ is the x-coordinate of currently localised point P.

f-on 6	***up/down(P)*** = $\boldsymbol{y_{Pneutral}}$ - $\boldsymbol{y_{Pexamined}}$, where $\boldsymbol{y_{Pneutral}}$ is the y-coordinate of point P retrieved from the "neutral expression" file and $\boldsymbol{y_{Pexamined}}$ is the y-coordinate of currently localised point P.
f-on 7	***curvature(P1-P2)***$_{\textit{deviation}}$ = ***max(***$\boldsymbol{x_{Pi\text{-}neutral}}$ | ***Pi-neutral*** ∈ curvature P1-P2$_{\text{neutral}}$) - ***max(***$\boldsymbol{x_{Pj\text{-}examined}}$ | ***Pj-examined*** ∈ curvature P1-P2$_{\text{examined}}$), where $\boldsymbol{x_{Pi\text{-}neutral}}$ is the x-coordinate of point ***Pi-neutral*** belonging to the curvature P1-P2$_{\text{neutral}}$ extracted from the "neutral expression" file stored in the DB of extreme model deformations of the currently observed subject and $\boldsymbol{x_{Pj\text{-}examined}}$ is the x-coordinate of point ***Pj-examined*** belonging to the currently localised curvature P1-P2$_{\text{examined}}$.

Variables (Thresholds)

var 1	***t1*** = $\boldsymbol{x_{extreme}}$, where ***x*** = IJ$_{\text{deviation}}$ and $\boldsymbol{x_{extreme}}$ encountered by maximal AU18
var 2	***ε1*** = 0.1 * ***min(***|$\boldsymbol{x1_{extreme}}$|, |$\boldsymbol{x2_{extreme}}$|***)***, where ***x1***=***x2***=IB$_{\text{deviation}}$, $\boldsymbol{x1_{extreme}}$ encountered by maximal AU12 and $\boldsymbol{x2_{extreme}}$ encountered by maximal AU15
var 3	***ε2*** = 0.1 * ***min(***|$\boldsymbol{x1_{extreme}}$|, |$\boldsymbol{x2_{extreme}}$|***)***, where ***x1***=***x2***=JB1$_{\text{deviation}}$, $\boldsymbol{x1_{extreme}}$ encountered by maximal AU12 and $\boldsymbol{x2_{extreme}}$ encountered by maximal AU15
var 4	***t4*** = $\boldsymbol{x_{extreme}}$, where ***x*** = P4P10$_{\text{deviation}}$ and $\boldsymbol{x_{extreme}}$ encountered by maximal AU26

Dual-view face-model-based rules

rule 1	If BD$_{\text{deviation}}$ > 0 OR B1D1$_{\text{deviation}}$ > 0 Then AU1 AND ***I (***AU1***)***; ***I (***AU1***)*** = ***max (extent (***BD$_{\text{deviation}}$***), extent (***B1D1$_{\text{deviation}}$***))***
rule 2	If AE$_{\text{deviation}}$ > 0 OR A1E1$_{\text{deviation}}$ > 0 Then AU2 AND ***I (***AU2***)***; ***I (***AU2***)*** = ***max (extent (***AE$_{\text{deviation}}$***), extent (***A1E1$_{\text{deviation}}$***))***
rule 3	If DD1$_{\text{deviation}}$ < 0 AND NOT AU9 Then AU4 AND ***I (***AU4***)***; ***I (***AU4***)*** = ***extent (***DD1$_{\text{deviation}}$***)***
rule 4	If FG$_{\text{deviation}}$ > 0 OR F1G1$_{\text{deviation}}$ > 0 Then AU5 AND ***I (***AU5***)***; ***I (***AU5***)*** = ***max (extent (***FG$_{\text{deviation}}$***), extent (***F1G1$_{\text{deviation}}$***))***
rule 5	If AU12 OR AU13 Then AU6 AND ***I (***AU6***)***; ***I (***AU6***)*** = ***I (***AU12 OR AU13***)***
rule 6	If ((FG > 0 AND GX$_{\text{deviation}}$ < 0) OR (F1G1 > 0 AND G1Y$_{\text{deviation}}$ < 0)) AND NOT (AU9 OR AU12) Then AU7 AND ***I (***AU7***)***; ***I (***AU7***)*** = ***max (extent (***GX$_{\text{deviation}}$***), extent (***G1Y$_{\text{deviation}}$***))***
rule 7	If P5P6$_{\text{deviation}}$ > 0 AND ***in/out***(P6) < 0 AND ***in/out***(P8) < 0 AND (P6-P8 has [shape) AND P8P10$_{\text{deviation}}$ > 0 AND NOT (AU9 OR AU12 OR AU13 OR AU15 OR AU17 OR AU18 OR AU20 OR AU23 OR AU24 OR AU35) Then AU8 AND ***I (***AU8***)***; ***I (***AU8***)*** = ***extent (***P5P6$_{\text{deviation}}$***)***
rule 8	If ***curvature***(P2-P3)$_{\text{deviation}}$ < 0 Then AU9 AND ***I (***AU9***)***; ***I (***AU9***)*** = ***extent (***P2-P3$_{\text{deviation}}$***)***
rule 9	If ***in/out***(P6) < 0 AND ***up/down***(P6) < 0 AND P5P6$_{\text{deviation}}$ < 0 AND ***curvature*** (P2-P3)$_{\text{deviation}}$ ≥ 0 Then AU10 AND ***I (***AU10***)***; ***I (***AU10***)*** = ***extent (***P5P6$_{\text{deviation}}$***)***
rule 10	If (IB$_{\text{deviation}}$ < 0 AND CI$_{\text{deviation}}$ > 0) OR (JB1$_{\text{deviation}}$ < 0 AND CJ$_{\text{deviation}}$ > 0) Then AU12 AND ***I (***AU12***)***; ***I (***AU12***)*** = ***max (extent (***IB$_{\text{deviation}}$***), extent (***JB1$_{\text{deviation}}$***))***

rule 11	If ($IB_{\text{deviation}} < 0$ AND $CI_{\text{deviation}} < 0$) OR ($JB1_{\text{deviation}} < 0$ AND $CJ_{\text{deviation}} < 0$) Then AU13 AND ***I*** *(*AU13*)*; ***I*** *(*AU13*)* = ***max*** *(* ***extent*** *(*$IB_{\text{deviation}}$*)*, ***extent*** *(*$JB1_{\text{deviation}}$*))*
rule 12	If $IB_{\text{deviation}} > 0$ OR $JB1_{\text{deviation}} > 0$ Then AU15 AND ***I*** *(*AU12*)*; ***I*** *(*AU15*)* = ***max*** *(* ***extent*** *(*$IB_{\text{deviation}}$*)*, ***extent*** *(*$JB1_{\text{deviation}}$*))*
rule 13	If $P8P10_{\text{deviation}} < 0$ AND ***in/out***(P8) < 0 AND ***up/down***(P8) > 0 Thcn AU16 AND ***I*** *(*AU16*)*; ***I*** *(*AU16*)* = ***extent*** *(*$P8P10_{\text{deviation}}$*)*
rule 14	If ***in/out***(P10) > 0 AND NOT (AU28 OR AU28t OR AU28b) Then AU17 AND ***I*** *(*AU17*)* = 100
rule 15	If $IJ_{\text{deviation}} < 0$ AND $IJ_{\text{deviation}} \geq$ ***t1*** AND $KL_{\text{deviation}} \geq 0$ Then AU18 AND ***I*** *(*AU18*)*; ***I*** *(*AU18*)* = ***extent*** *(*$IJ_{\text{deviation}}$*)*
rule 16	If (P6-P8 contains two valleys and a peak) Then AU19 AND ***I*** *(*AU19*)* = 100
rule 17	If $IJ_{\text{deviation}} > 0$ AND $\lvert IB_{\text{deviation}}\rvert < \varepsilon 1$ AND $\lvert JB1_{\text{deviation}}\rvert < \varepsilon 2$ Then AU20 AND ***I*** *(*AU20*)*; ***I*** *(*AU20*)* = ***extent*** *(*$IJ_{\text{deviation}}$*)*
rule 18	If KL > 0 AND $KL_{\text{deviation}} < 0$ AND $IJ_{\text{deviation}} \geq 0$ AND $IB_{\text{deviation}} \leq 0$ AND $JB1_{\text{deviation}} \leq 0$ AND NOT (AU28t OR AU28b) Then AU23 AND ***I*** *(*AU23*)*; ***I*** *(*AU23*)* = ***extent*** *(*$KL_{\text{deviation}}$*)*
rule 19	If KL > 0 AND $KL_{\text{deviation}} < 0$ AND $IJ_{\text{deviation}} < 0$ AND $IJ_{\text{deviation}} >$ ***t1*** AND NOT (AU9 OR AU10 OR AU12 OR AU13 OR AU15 OR AU17 OR AU28t OR AU28b) Then AU24 AND ***I*** *(*AU24*)*; ***I*** *(*AU24*)* = ***extent*** *(*$KL_{\text{deviation}}$*)*
rule 20	If $P6P8_{\text{deviation}} > 0$ AND $P4P10_{\text{deviation}} \leq 0$ Then AU25 AND ***I*** *(*AU25*)*; ***I*** *(*AU25*)* = ***extent*** *(*$P6P8_{\text{deviation}}$*)*
rule 21	If $P4P10_{\text{deviation}} > 0$ AND $P4P10_{\text{deviation}} \leq$ ***t4*** Then AU26 AND ***I*** *(*AU26*)*; ***I*** *(*AU26*)* = ***extent*** *(*$P4P10_{\text{deviation}}$*)*
rule 22	If $P4P10_{\text{deviation}} >$ ***t4*** Then AU27 AND ***I*** *(*AU27*)*; ***I*** *(*AU27*)* = ***extent*** *(*$P4P10_{\text{deviation}}$*)*
rule 23	If (P6 is absent) AND (P8 is absent) Then AU28 AND ***I*** *(*AU28*)* = 100
rule 24	If (P6 is absent) Then AU28t AND ***I*** *(*AU28t*)* = 100
rule 25	If (P8 is absent) Then AU28b AND ***I*** *(*AU28b*)* = 100
rule 26	If ***in/out***(P10) < 0 AND NOT AU27 Then AU29 AND ***I*** *(*AU29*)* = 100
rule 27	If $IJ_{\text{deviation}} <$ ***t1*** Then AU35 AND ***I*** *(*AU35*)* = 100
rule 28	If (P9 is absent) Then AU36b AND ***I*** *(*AU36b*)* = 100
rule 29	If ***curvature***$(P5\text{-}P6)_{\text{deviation}} < 0$ Then AU36t AND ***I*** *(*AU36t*)* = 100
rule 30	If $H'H1'_{\text{deviation}} > 0$ AND NOT (AU8 OR AU9 OR AU10 OR AU12 OR AU13 OR AU15 OR AU18 OR AU24 OR AU28) Then AU38 AND ***I*** *(*AU38*)*; ***I*** *(*AU38*)* = ***extent*** *(*$H'H1'_{\text{deviation}}$*)*
rule 31	If $H'H1'_{\text{deviation}} < 0$ AND NOT (AU8 OR AU9 OR AU10 OR AU12 OR AU13 OR AU15 OR AU18 OR AU24 OR AU28) Then AU39 AND ***I*** *(*AU39*)*; ***I*** *(*AU39*)* = ***extent*** *(*$H'H1'_{\text{deviation}}$*)*
rule 32	If ((FG > 0 AND $FG_{\text{deviation}} < 0$ AND $FX_{\text{deviation}} < 0$) OR (F1G1 > 0 AND $F1G1_{\text{deviation}} < 0$ AND $F1Y_{\text{deviation}} < 0$)) AND NOT AU7 Then AU41 AND ***I*** *(*AU41*)*; ***I*** *(*AU41*)* = ***max*** *(* ***extent*** *(*$FX_{\text{deviation}}$*)*, ***extent*** *(*$F1Y_{\text{deviation}}$*))*

R-list representation of the relations among the dual-view face-model-based rules

Conclusion Clause		***Premise Clause***	
Rule #	*Clause #*	Rule #	*Clause #*
8	1	3	2
10	1	5	1
11	1	5	2
8	1	6	3
10	1	6	4
8	1	7	6
10	1	7	7
11	1	7	8
12	1	7	9
14	1	7	10
15	1	7	11
17	1	7	12
18	1	7	13
19	1	7	14
27	1	7	15
23	1	14	2
24	1	14	3
25	1	14	4
24	1	18	6
25	1	18	7
8	1	19	5
9	1	19	6
10	1	19	7
11	1	19	8
12	1	19	9
14	1	19	10
24	1	19	11
25	1	19	12
22	1	26	2
7	1	30	2
8	1	30	3
9	1	30	4
10	1	30	5
11	1	30	6
12	1	30	7
15	1	30	8
19	1	30	9
23	1	30	10
7	1	31	2
8	1	31	3
9	1	31	4
10	1	31	5
11	1	31	6
12	1	31	7

15	1	31	8
19	1	31	9
23	1	31	10
6	1	32	3

Bibliography[☼]

Aamodt, A. (1991) *A knowledge-intensive, integrated approach to problem solving and sustained learning.* PhD thesis, Norwegian Institute of Technology, Trondheim, Norway

Aamodt, A. and Plaza, E. (1994) CBR: foundational issues, methodological variations and system approaches. In: *AI Communications* 7(1): 39-59

Abdel-Mottaleb, M., Chellappa, R. and Rosenfeld, A. (1993) Binocular motion stereo using MAP estimation. In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 321-327

Adams, J.B. (1976) A probability model of medical reasoning and the MYCIN model. In: *Mathematical Biosciences* **32**: 177-186

[☼] References are listed alphabetically, as in the following examples: books and technical reports [1], articles in readings [2], articles in journals [3], articles in conference proceedings [4] or [5].

[1] Smith, J. (2001) *How to write a thesis.* Academic Press, Delft, NL

[2] Davis, B. (2001) Processing the bibliography. In: *Everything about writing a Ph.D. thesis*, Smith, J. (Ed.), pp. 1-7, Academic Press, Delft, NL

[3] Smith, J. (2001) Classifying bibliography. In: *IEEE Transactions on Copyright Transfer* 3(7):1-27

[4] Smith, J. (2001) Literature research. In: *Proceedings of the European Conference on Tools in Education*, pp. 13-18

[5] Smith, J. (2001) Proceedings with volume numbers. In: *Proceedings of the International Conference on Standards* **3**: 182-185

Adini, Y., Moses, Y. and Ullman, S. (1997) Face Recognition: The problem of compensating for changes in illumination direction. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(7): 721-732

Adjoudani, A. and Benoit, C. (1995) Audio-visual speech recognition compared across two architectures. In: *Proceedings of Eurospeech'95* **2**: 1563-1566

Aha, D. W., Kibler, D. and Albert, M.K. (1991) Instance-based learning algorithms. In: *Machine Learning* **6**: 37-66

Aha, D.W. (1998) The omnipresence of case-based reasoning in science and application. *Knowledge-Based Systems* **11**(5-6): 261-273

Aleksander, I. (Ed.) (1989) *Neural computing architectures – the design of brain-like machines*. Cambridge University Press, Cambridge, USA

Aloimonos, J.Y., Weiss, I. and Bandopadhay, A. (1987) Active Vision. In: *International Journal on Computer Vision*: 333-356

Amir, N. and Ron, S. (1998) Towards automatic classification of emotions in speech. In: *Proceedings of the International Conference on Spoken Language Processing*, pp. 555-558

Andersen, W.A., Evett, M.P., Hendler, J.A. and Kettler, B.P. (1994) Massively Parallel Matching of Knowledge Structures. In: *Massively Parallel Artificial Intelligence*, Kitano, H. and Hendler, J. (Eds.), pp. 52-73, AAAI Press/The MIT Press, Cambridge, USA

Andrews, R., Diederich, J. and Tickle, A.B. (1995) Survey and critique of techniques for extracting rules from trained artificial neural networks. In: *Knowledge-Based Systems* **8**(6): 373-389

Argyle, M (1972) Non-verbal communication in human social interaction. In: *Non-verbal communication*, Hinde, R. (Ed.), Cambridge University Press, Cambridge, UK

Arisha, K.A., Ozcan, F., Ross, R. and Subrahmanian, V.S. (1999) Impact: A platform for collaborating agents. In: *IEEE Intelligent Systems and Their Applications* **14**(2): 64-72

Aristotle (nd/1913) Physiognomonica. In: *The works of Aristotle*, Ross, W.D. (Ed.), pp. 805-813, Clarendon, Oxford, UK

Aristotle (nd/1993) Physiognomics. In: *Aristotle – Minor Works*, Hett, W.S. (Ed.), pp. 83-137, Harvard University Press, Cambridge, USA

Ashley, K.D. (1991) Reasoning with Cases and Hypotheticals in HYPO. In: *International Journal of Man-Machine Studies* **34**(6): 753-796

Atkeson, C.G., Moore, A.W. and Schaal, S. (1997) Locally Weighted Learning. In: *Artificial Intelligence Review* **11**: 11-73

Averill, J.R. (1986) Acquisition of emotions in adulthood. In: *The Social Construction of Emotions*, Harre, R. (Ed.), pp. 100, Blackwell, Oxford, UK

Bajcsy, R. (1988) Active Perception. In: *IEEE Proceedings* **76**(8): 996-1006

Barlow, M. and Rose, P. (Eds.)(2000) *Section on Multimodal Speech, Proceedings of the Australian International Conference on Speech Science and Technology*, pp. 86-111, Australian Speech Science and Technology Association Press, Canberra, AUS

Barr, A. and Feigenbaum, E.A. (1981) *The Handbook of Artificial Intelligence, Vol. I*. Morgan Kaufmann, Los Altos, USA

Bartlett, M.S., Hager, J.C., Ekman, P. and Sejnowski, T.J. (1999) Measuring facial expressions by computer image analysis. In: *Psychophysiology* **36**: 253-263

Bassili, J.N. (1978) Facial motion in the perception of faces and of emotional expression. In: *Journal of Experimental Psychology* **4**: 373-379

Beckman, T.J. (1991) Selecting expert system applications. In: *AI Expert*: 42-48

Bergamasco, M. (1995) Haptic interfaces: The study of force and tactile feedback systems. In: *Proceedings of the IEEE International Workshop on Robot and Human Communication*, pp. 15-20

Bezooijen, R.V. (1984) *Characteristics and recognizability of vocal expression of emotions*. Floris, Dordrecht, NL

Birdwhistell, R.L. (1970) *Kinesics and Context: Essays on Body Motion Communication*. University of Pennsylvania Press, Philadelphia, USA

Black, M.J., and Yacoob, Y. (1995) Tracking and recognising rigid and non-rigid facial motions using local parametric models of image motions. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 374-381

Black, M.J. and Yacoob, Y. (1997) Recognising facial expressions in image sequences using local parameterised models of image motion. In: *International Journal of Computer Vision* **25**(1): 23-48

Boehm, B.W. (1988) *A Spiral Model of Software Development and Enhancement.* Prentice Hall, Englewood Cliffs, USA

Boullart, L. (1992) A Gentle Introduction to Artificial Intelligence. In: *Application of artificial intelligence in process control*, Boullart, L., Krijgsman, A. and Vingerhoeds, R.A. (Eds.), pp. 5-40, Pergamon Press, Oxford, UK

Bower, T.G.R. (1974) The evolution of the sensory system. In: *Perception: Essays in Honour of James J. Gibson*, MacLeod, R.B. and Pick, H.L. (Eds.), pp. 141-153, Cornell Univ. Press, Ithaca, USA

Bowyer, K.W. and Phillips, P.J. (1998) *Empirical evaluation techniques in compute vision.* IEEE Computer Society Press, Los Alamitos, USA

Box, G. (1957) Evolutionary operation: A method for increasing industrial productivity. In: *Journal of the Royal Statistical Society* 6(2): 81-101

Boyle, E., Anderson, A.H. and Newlands, A. (1994) The effects of visibility on dialogue and performance in a co-operative problem solving task. In: *Language and Speech* **37**(1): 1-20

Brooke, S. and Jackson, C. (1991) Advances in elicitation by exception. In: *Proceedings of the SGES International Workshop on Knowledge Based Systems Methodologies*, pp. 70-78

Brown, D.E. (1991) *Human universals*. Temple University Press, Philadelphia, USA

Bruce, V. (1986) *Recognising faces*. Lawrence Erlbaum Associates, Hove, UK

Bruce, V. (1992) What the human face tells the human mind: Some challenges for the robot-human interface. In: *Proceedings of the IEEE Workshop on Robot and Human Communication*, pp. 44-51

Bruce, V., Burton, A.M. and Craw, I. (1992) Modelling face recognition. In: *Phil. Trans. Roy. Soc. London* **B335**: 121-128

Buchanan, B.G. and Shortliffe, E.H. (Eds.) (1984) *Rule-Based Expert Systems.* Addison-Wesley, Reading, USA

Buck, R. and Duffy, R. (1980) Nonverbal communication of affect in brain-damaged patients. In: *Cortex* **16**: 351-362

Buhmann, J., Lange, J. and von der Malsburg, C. (1989) Distortion invariant object recognition by matching hierarchically labelled graphs. In: *Proceedings of the International Joint Conference on Neural Networks*, pp. 155-159

Burdea, G. (1996) *Force and Touch Feedback for Virtual Reality*. John Wiley & Sons, New York, USA

Cacioppo, J. T., Tassinary, L. G. and Berntson, G. G. (Eds.) (2000). *Handbook of psychophysiology, 2nd edition*. Cambridge University Press, New York, USA

Campbell, F.W. and Green, D.G. (1965) Optical and retinal factors affecting visual resolution. In: *Journal of Physiology* **181**: 576-593

Carlson, J.G. and Hatfield, E. (1992) *Psychology of emotion*. Harcourt Brace Jovanovich, Fort Worth, USA

Carpenter, G.A. and Grossberg, S. (1987) A massively parallel architecture for a self-organising neural pattern recognition machine. In: *Computer Vision, Graphics, and Image Processing* **37**: 54-115

Cassell, J. and Bickmore, T. (2000) External manifestations of trustworthiness in the interface. In: *Communications of the ACM* **43**(12): 50-56

Cerezo, E., Pina, A. and Seron, F. (1999) Motion and behavior modelling: State of art and new trends. In: *The Visual Computer* **15**: 124-146

Chen, L.S., Huang, T.S., Miyasato, T. and Nakatsu, R. (1998) Multimodal human emotion/expression recognition. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 366-371

Chen, T. and Rao, R.R. (1998) Audio-visual integration in multimodal communication. In: *Proceedings of the IEEE* **86**(5): 837-852

Chen, T. (2001) Audiovisual speech processing. In: *IEEE Signal Processing Magazine* **18**(1): 9-21

Church, R.B. and Meadow, S.G. (1986) The mismatch between gesture and speech as an index of transitional knowledge. In: *Cognition* **23**: 43-71

Clark, P. and Niblett, R. (1989) The CN2 induction algorithm. In: *Machine Learning* **3**: 261-284

Clarkson, B., Mase, K. and Pentland, A. (2000) *Recognizing user's context from wearable sensors: Baseline system.* Technical Report (TR #519), Massachusetts Institute of Technology, Cambridge, USA

Coen, M.H. (1999) The future of human-computer interaction, or how I learned to stop worrying and love my intelligent room. In: *IEEE Intelligent Systems and Their Applications* **14**(2): 8-10

Cohn, J.F., Zlochower, A.J., Lien, J.J. and Kanade, T. (1998) Feature-point tracking by optical flow discriminates subtle differences in facial expression. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 396-401

Coleman, R. and Williams, R. (1979) Identification of emotional states using perceptual and acoustic analyses. In: *Care of the Professional Voice 1*, Lawrence, V. and Weinberg, B. (Eds.), The Voice Foundation, New York, USA

Collins, R.T., Lipton, A.J. and Kanade, T. (Eds.) (2000) Special Section on Video Surveillance. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(8): 745-887

Cootes, T.F., Taylor, C.J., Cooper, D.H. and Graham, J. (1995) Active shape models – training and application. In: *Computer Vision Image Understanding* **61**(1): 38-59

Cootes, T.F., Edwards, G.J. and Taylor, C.J. (1998) Active Appearance Models. In: *Proceedings of the European Conference on Computer Vision* **2**: 484-498

Cornelius, R. (1996) *The Science of Emotion.* Prentice Hall, Englewood Cliffs, USA

Costas, T. and Kashyap, R.L. (1993) Case-based reasoning and learning in manufacturing with TOTLEC planner. In: *IEEE Transactions on Systems, Man and Cybernetics* **23**(4): 1010-1022

Cottrell, G.W. and Metcalfe, J. (1991) EMPATH: Face, Emotion, Gender Recognition Using Holons. In: *Advances in Neural Information Processing Systems* **3**: 564-571

Cover, T. and Hart, P. (1967) Nearest neighbour pattern classification. In: *IEEE Transactions on Information Theory* **13**: 21-27

Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W. and Taylor, J.G. (2001) Emotion recognition in human-computer interaction. In: *IEEE Signal Processing Magazine* **18**(1): 32-80

Dariush, B., Kang, S.B. and Waters, K. (1998) Spatio-temporal analysis of face profiles: detection, segmentation and registration. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 248-253

Darwin, C. (1965/1872) *The expression of the emotions in man and animals.* University of Chicago Press, Chicago, USA (original work published 1872)

Dasarathy, B.V. (1997) Sensor fusion potential exploitation – Innovative architectures and illustrative approaches. In: *Proceedings of the IEEE* **85**(1): 24-38

Davidson, R., Allman, J., Cacioppo, J., Ekman, P., Friesen, W., Hager, J.C. and Phillips, M. (1993) Basic Science for Understanding Facial Expression. In: *Final Report to NSF of the Planning Workshop on Facial Expression Understanding*, Ekman, P., Huang, T.S., Sejnowski, T.J. and Hager, J.C. (Eds.), pp. 32-38, Technical Report, Human Interaction Laboratory, University of California, San Francisco, USA

Davis, R. and King, J. (1977) An overview of production systems. In: *Machine Intelligence* **8**: 300-332

Davis, R (1998) What are Intelligence? And Why? In: *AI Magazine* **19**(1): 91-110

De Bondt, P.C. (1995) *A modular approach for facial analysis*. M.Sc. thesis, Delft University of Technology, Faculty of Technical Mathematics and Informatics, Department of Knowledge Based Systems, Delft, NL

De Carlo, D., Metaxas, D. and Stone, M. (1998) An anthropometric face model using variational techniques. In: *Proceedings of SIGGRAPH*, pp. 67-74

De Carlo, D. and Metaxas, D. (1999) Combining information using hard constraints. In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition '99*, pp. 132-138

De Jonge, D. (1995) *Toward automatic facial feature extraction and recognition of facial expressions.* M.Sc. thesis, Delft University of Technology, Faculty of Technical Mathematics and Informatics, Department of Knowledge Based Systems, Delft, NL

De Raedt, L. (Ed.) (1996) *Advances in Inductive Logic Programming*. IOS Press, Amsterdam, NL

De Silva, L.C., Miyasato, T. and Nakatsu, R. (1997) Facial emotion recognition using multimodal information. In: *Proceedings of the Information, Communication and Signal Processing Conference*, pp. 397-401

De Silva, L.C. and Ng, P.C. (2000) Bimodal emotion recognition. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 332-335

Dellaert, F., Polzin, T. and Waibel, A. (1996) Recognizing emotion in speech. In: *Proceedings of the International Conference on Spoken Language Processing*, pp. 1970-1973

Dietterich, T., Hild, H. and Bakiri, G. (1995) A comparison of ID3 and backpropagation for English text-to-speech mapping. In: *Machine Learning* **18**(1): 51-80

Djurica, M. (2001) *Design of Low Power Analog to Digital Converters*. PhD thesis, Delft University of Technology, Delft, NL

Donato, G., Bartlett, M.S., Hager, J.C., Ekman, P. and Sejnowski, T.J. (1999) Classifying Facial Actions. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(10): 974-989

Duda, R., Gasching, H. and Hart, P. (1979) Model Design in the PROSPECTOR Consultant System for Mineral Exploration. In: *Expert Systems in the Micro-Electronic Age*, Michie, D. (Ed.), pp. 153-167, Edinburgh University Press, Edinburgh, UK

Durfee, E.H., Lesser, V.R. and Corkill, D.D. (1989) Trends in cooperative distributed problem solving. In: *IEEE Transactions on Knowledge Data Engineering* **11**(1): 63-83

Edwards, G.J., Cootes, T.F. and Taylor, C.J. (1998) Face recognition using active appearance models. In: *Proceedings of the European Conference on Computer Vision* **2**: 581-695

Efron, D. (1941) *Gesture and environment*. King's Crown Press, New York, USA

Eisert, P. and Girod, B. (1998) Analysing facial expressions for virtual conferencing. In: *IEEE Computer Graphics and Applications* **18**(5): 70-78

Ekman, P. and Friesen, W.V. (1969) The repertoire of nonverbal behavioral categories – origins, usage, and coding. In: *Semiotica* **1**: 49-98

Ekman, P., Friesen, W.V. and Tomkins, S. (1971) Facial affect scoring technique: A first validity study. In: *Semiotica* **3**: 37-58

Ekman, P. and Friesen, W.V. (1975) *Unmasking the Face.* Prentice Hall, New Jersey, USA

Ekman, P. and Friesen, W.V. (1978) *Facial Action Coding System (FACS): Manual.* Consulting Psychologists Press, Palo Alto, USA

Ekman, P. (1978) Facial signs: Facts, fantasies, and possibilities. In: *Sight, Sound, and Sense*, Sebeok, T. (Ed.), Indiana University Press, Bloomington, USA

Ekman, P. (1980) *The face of man: Expressions of universal emotions in a New Guinea village.* Garland STPM Press, New York, USA

Ekman, P. (1982) *Emotion in the Human Face.* Cambridge University Press, New York, USA

Ekman, P. (1982b) Methods for measuring facial action. In: *Handbook of methods in Non-verbal behaviour research*, Scherer, K.R. and Ekman, P. (Eds.), pp. 45-90. Cambridge University Press, Cambridge, USA

Ekman, P. and Friesen, W.V. (1984) *Unmasking the face: A guide to recognising emotions from facial cues.* Prentice Hall, Englewood Cliffs, USA

Ekman, P. and Sejnowski, T.J. (1993) Executive Summary. In: *Final Report to NSF of the Planning Workshop on Facial Expression Understanding*, Ekman, P., Huang, T.S., Sejnowski, T.J. and Hager, J.C. (Eds.), pp. 3-5, Technical Report, Human Interaction Laboratory, University of California, San Francisco, USA

Ekman, P. (1994) Strong Evidence for Universals in Facial Expressions: A Reply to Russell's Mistaken Critique. In: *Psychological Bulletin* **115**(2): 268-287

Ellis, H.D. (1986) Process underlying face recognition. In: *The Neuropsychology of Face Perception and Facial Expression*, Bruyer, R. (Ed.), pp. 1-27, Lawrence Erlbaum Associates, New Jersey, USA

Engel, F.L., Goossens, P. and Haakma, R. (1994) Improved efficiency through I- and E-feedback: A trackball with contextual force feedback. In: *International Journal on Human-Computer Studies* **41**: 949-974

Erman, L.D., Hayes-Roth, F., Lesser, V.R. and Reddy, D.R. (1988) The HEARSAY-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty. In: *Computing Surveys* **12**(2): 213-253 (also in: *Blackboard Systems*, Engelmore, R. and Morgan, T. (Eds.), pp. 31-86, Addison-Wesley, Reading, USA)

Essa, I.A. and Pentland, A.P. (1995) Facial expression recognition using visually extracted facial action parameters. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 35-40

Essa, I.A. and Pentland, A.P. (1997) Coding, Analysis, Interpretation and Recognition of Facial Expressions. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(7): 757-763

Falkenhainer, B., (1988) *Learning from physical analogies: A study in analogy and the explanation process*. PhD thesis, University of Illinois, Urbana Champaign, USA

Farrel, R. (1987) Intelligent case selection and presentation. In: *Proceedings of the IEEE/ AAAI International Joint Conference on Artificial Intelligence*, pp. 174-176

Feigenbaum, E.A., Buchanan, G. and Lederberg J. (1971) Generality and Problem Solving: A Case Study Using the DENDRAL Program. In: *Machine Intelligence* **6**: 165-190

Feigenbaum, E.A. (1977) *The art of artificial intelligence: I. Themes and case studies of knowledge engineering*. Computer Science Technical Report (CS-TR-77-621), Stanford University, Palo Alto, USA

Findler, N.V. (Ed.) (1979) *Associative Networks*. Academic Press, New York, USA

Fishbein, M. and Ajazen, I. (1975) *Belief, Attitude, Intention and Behaviour: An Introduction to Theory and Research*. Addison-Wesley, Reading, USA

Folgel, L.J., Owens, A.J. and Walsh, M.J. (1966) *Artificial intelligence through simulated evolution*. John Wiley & Sons, New York, USA

Fonagy, I. (1978) Emotions, voice and music. In: *Language and Speech* **21**: 34-49

Forrest, S. (1993) Genetic algorithms: Principles of natural selection applied to computation. In: *Science* **261**: 872-878

Frick, R. (1985) Communicating emotion: The role of prosodic features. In: *Psychological Bulletin* 97(3): 412-429

Fridlund, A.J., Ekman, P. and Oster, H. (1987) Facial expressions of emotions: Review literature 1970-1983. In: *Nonverbal behaviour and communication*, Siegman, A.W. and Feldstein, S. (Eds.), pp. 143-224, Lawrence Erlbaum Associates, New Jersey, USA

Fridlund, A.J. (1991) Evolution and facial action in reflex, social motive, and paralanguage. In: *Biological Psychology* **32**: 3-100

Friedman, H.S., Prince, L.M., Riggio, R.E. and Dimatteo, M.R. (1990) Understanding and assessing nonverbal expressiveness: The affective communication test. In: *Journal of Personality and Social Psychology* **39**: 333-351

Friesen, W.V. and Ekman, P. (1987) *Dictionary – Interpretation of FACS Scoring.* Unpublished manuscript, Human Interaction Lab, University of California, San Francisco, USA

Frijda, N.H. (1986) *The emotions.* Cambridge University Press, Cambridge, UK

Furnas, G., Landauer, T., Gomes, L. and Dumais, S. (1987) The vocabulary problem in human-system communication. In: *Communications of the ACM* **30**(11): 964-972

Furnkranz, J. (2001) Machine Learning in Games: A Survey. In: *Machines that Learn to Play Games*, Furnkranz, J. and Kubat, M. (Eds.), pp. 11-59, Nova Scientific Publishers, Huntington, USA

Furukawa, K., Michie, D. and Muggleton, S. (1999) *Machine Intelligence 15: machine intelligence and inductive learning.* Oxford University Press, Oxford, UK

Gasser, L. (1991) Social conceptions of knowledge and action: DAI foundations and open systems semantics. In: *Artificial Intelligence* **47**: 107-138

Gavrila, D. (1999) The visual analysis of human movement: A survey. In: *Computer Vision and Image Understanding* 73(1): 82-98

Giarratano, J. and Riley, G. (1990) *Expert Systems – Principles and Programming.* PWS-KENT Publishing Company, Boston, USA

Gillenson, M.L. (1974) *The interactive generation of facial images on a CRT using a heuristic strategy*. Ohio State University, CG Research Group, Columbus, USA

Glassner, A.S. (1993) *Graphics Gems*. Academic Press, London, UK

Goker, M., Roth-Berghofer, T., Bergmann, R., Pantleon, T., Traphoner, R., Wess, S. and Wilke, W. (1998) The development of HOMER – a case-based CAD/CAM help-desk support tool. In: *Proceedings of the European Workshop on Case-Based Reasoning*, pp. 346-357

Goldberg, D. (1994) Genetic and evolutionary algorithms age. In: *Communications of the ACM* **37**(3): 113-119

Goleman, D. (1995) *Emotional Intelligence*. Bantam Books, New York, USA

Golomb, B. and Sejnowski, T.J. (1993) Benefits from efforts to understand the face. In: *Final Report to NSF of the Planning Workshop on Facial Expression Understanding*, Ekman, P., Huang, T.S., Sejnowski, T.J. and Hager, J.C. (Eds.), pp. 57-61, Technical Report, Human Interaction Laboratory, University of California, San Francisco, USA

Good, I.J. (1968) Corroboration, explanation, evolving probability, simplicity and a sharpened razor. In: *British Journal of Philosophical Science* **19**: 123-143

Hager, J.C. (1985) A comparison of units for visually measuring facial action. In: *Behaviour research methods, instruments and computers* **17**: 450-468

Hall, D.L. and Llinas, J. (1997) An introduction to multisensor data fusion. In: *Proceedings of the IEEE* **85**(1): 6-23

Hand, D.J. (1981) *Discrimination and Classification*. John Wiley and Sons, New York, USA

Hara, F. and Kobayashi, H. (1997a) Facial interaction between animated 3D face robot and human beings. In: *Proceedings of the IEEE International Conference on System, Man and Cybernetics*, pp. 3732-3737

Hara, F. and Kobayashi, H. (1997b) State of the art in component development for interactive communication with humans. In: *Advanced Robotics* **11**(6): 585-604

Haralick, R.M. and Shapiro, L.G. (1992) *Computer and Robot Vision*. Addison-Wesley, Reading, USA

Hassler, S. (Ed.) (2001) Always On: Living in a Networked World. Special Issue on Technology 2001: Analysis and Forecast. In: *IEEE Spectrum* **38**(1): 3-138

Hayes-Roth, B., van Gent, R., Reynolds, R. Johnson, M.V. and Wescourt, K. (1999) Web guides. In: *IEEE Intelligent Systems and Their Applications* **14**(2): 23-27

Healey, J. and Picard, R.W. (1998) Digital processing of affective signals. In: *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing*, pp. 3749-3752

Hecht-Nielsen, R. (1990) *Neurocomputing*. Addison Wesley, Reading, USA

Heckerman, D. (1990) Probabilistic interpretations for MYCIN's certainty factors. In: *Readings in Uncertain Reasoning*, Shafer, G. and Pearl, J. (Eds.), pp. 298-312, Morgan Kaufmann, San Mateo, USA

Hendler, J. (1999) Making sense out of agents. In: *IEEE Intelligent Systems and Their Applications* **14**(2): 32-37

Herpers, R., Witta, L., Bruske, J. and Sommer, G. (1997) Dynamic sell structures for the evaluation of key points in facial images. In: *International Journal of Neural Systems* **8**(1): 27-39

Hinrichs, T.R. (1992) *Problem Solving in Open Worlds*. Lawrence Erlbaum, Northvale, USA

Hinton, G., Sejnowski, T.J. and Ackley, D.H. (1984) *Boltzmann machines: constraint satisfaction networks that learn*. Technical report (# CMU-CS-84-119), Carnegie Mellon University, Pittsburgh, USA

Holland, J.H. (1962) Outline for a logical theory of adaptive systems. In: *Journal of the Association for Computing Machinery* **3**: 297-314

Hong, H., Neven, H. and von der Malsburg, C. (1998) Online facial expression recognition based on personalised galleries. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 354-359

Hopfield, J.J. (1982) Neural networks and physical systems with emergent collective computational abilities. In: *Proceedings of the National Academy of Sciences* **79**: 2554-2558

Horn, B.K.P. and Schunck, B.G. (1981) Determining optical flow. In: *Artificial Intelligence* **17**(1-3): 185-203

Horton, S.V. (1987) Reduction of disruptive mealtime behaviour by facial screening: A case study of a mentally retarded girl. In: *Behaviour Modification* **11**(1): 53-64

Horvitz, E. and Heckerman, D. (1986) The inconsistent use of measures of certainty in artificial intelligence research. In: *Uncertainty in Artificial Intelligence*, Kanal, L.N. and Lemmer, J.F. (Eds.), pp. 137-151. North Holland, Amsterdam, NL

Huang, C.L. and Huang, Y.M. (1997) Facial expression recognition using model-based feature extraction and action parameters classification. In: *Journal of Visual Communication and Image Representation* **8**(3): 278-290

Hughes, J.G. (1991) *Object-oriented Databases*. Prentice Hall, Hertfordshire, UK

Hurwitz, T.A., Wada, J.A., Koska, B.D. and Strauss, E.H. (1985) Cerebral organisation of affect suggested by temporal lobe seizures. In: *Neurology* **35**(9): 1335-1337

Izard, C.E. (1971) *The face of emotion*. Appleton Century Crofts, New York, USA

Izard, C.E. (1980) Cross-cultural perspectives on emotion and emotion communication. In: *Handbook of cross-cultural psychology – Basic processes (vol. 3)*, Triandis, H. and Lonner, W. (Eds.), pp. 185-222, Allyn & Bacon, Boston, USA

Izard, C.E. (1990) Facial expressions and the regulations of emotions. In: *Journal on Personality and Social Psychology* **58**(3): 487-498

Izzo, G. (1998) *Multiresolution techniques and emotional speech*. Technical Report on PHYSTA Project, NTUA Image Processing Laboratory, Athens, GR

Jackson, P. (1999) *Introduction to Expert Systems*. Addison Wesley Longman Limited, Harlow, UK

Jamali, N., Thati, P. and Agha, G.A. (1999) An actor-based architecture for customizing and controlling agent ensembles. In: *IEEE Intelligent Systems and Their Applications* **14**(2): 38-44

Jarmulak, J. (1999) *Case-Based Reasoning for NDT Data Interpretation*. PhD thesis, Delft University of Technology, Delft, NL

Juang, B.H. and Furui, S. (Eds.) (2000) Special Issue on Spoken Language Processing. In: *Proceedings of the IEEE* **88**(8): 1139-1366

Juran, J.M. and Gryna, F.M. (1970) *Quality Planning and Analysis: From Product Development Trough Use*. McGraw-Hill, New York, USA

Kaelbling, L.P., Littman, M.L. and Moore, A.W. (1996) Reinforcement learning: A survey. In: *Journal of AI Research* **4**: 237-285

Kahneman, D. and .Tversky, A. (1972) Subjective probability: a judgement of representativeness. In: *Cognitive Psychology* **3**: 430-454

Kahneman, D., Slovic, P. and Tversky, A. (Eds.) (1982) *Judgment under Uncertainty: Heuristics and Biases*. Cambridge University Press, Cambridge, USA

Kan, S.H. (1995) *Metrics and Models in Software Quality Engineering*. Addison-Wesley, Reading, USA

Kandel, A. (1991) *Fuzzy Expert Systems*. CRC Press, Boca Raton, USA

Kang, B.S., Han, C.H., Lee, S.T., Young, D.H. and Lee, C. (2000) Speaker dependent emotion recognition using speech signals. In: *Proceedings of the International Conference on Spoken Language Processing*, pp. 383-386

Kanter, I. and Sompolinsky, H. (1987) Associative recall of memory without errors. In: *Physical Review* **35**(1): 380-392

Kasabov, N.K. (1996) *Foundations of neural networks, fuzzy systems, and knowledge engineering*. MIT Press, Cambridge, USA

Kass, M., Witkin, A. and Terzopoulos, D. (1987) Snake: active contour model. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 259-269

Kato, M., So, I., Hishinuma, Y., Nakamura, O. and Minami, T. (1991) Description and synthesis of facial expressions based on isodensity maps. In: *Visual Computing*, Kunii, T.L. (Ed.), pp. 39-56, Springer-Verlag, Tokyo, Japan

Kawakami, F., Okura, M., Yamada, H., Harashima, H. and Morishima, S. (1995) 3D emotion space for interactive communication. In: *Proceedings of the International Conference on Computer Science*, pp. 471-478

Kearney, G.D. and McKenzie, S. (1993) Machine Interpretation of Emotion: Design of a Memory-Based Expert System for Interpreting Facial Expressions in Terms of Signalled Emotions (JANUS). In: *Cognitive Science* **17**(4): 589-622

Keltner, D. and Buswell, B.N. (1996) Evidence for the distinctness of embarrassment, shame, and guilt: A study of recalled antecedents and facial expressions of emotion. In: *Cognition and Emotion* **10**(2): 155-171

Keltner, D. and Ekman, P. (2000) Facial Expression of Emotion. In: *Handbook of Emotions*, Lewis, M. and Haviland-Jones, J.M. (Eds.), pp. 236-249, Guilford Press, New York, USA

Kimura, S. and Yachida, M. (1997) Facial expression recognition and its degree estimation. In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 295-300

Kobayashi, H. and Hara, F. (1992a) Recognition of six basic facial expressions and their strength by a neural network. In: *Proceedings of the International Workshop on Robot and Human Communication*, pp. 381-386

Kobayashi, H. and Hara, F. (1992b) Recognition of mixed facial expressions by a neural network. In: *Proceedings of the International Workshop on Robot and Human Communication*, pp. 387-391

Kober, R., Harz, U. and Schiffers (1997) Fusion of visual and acoustic signals for command-word recognition. In: *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing*, pp. 1495-1498

Kohonen, T. (1982) Self-organised formation of topologically correct feature maps. In: *Biological Cybernetics* **43**: 59-69

Kohonen, T. (1990) The self-organising map. In: *Proceedings of the IEEE* **78**: 1464-1497

Kolodner, J. (1983) Maintaining organisation in a dynamic long-term memory. In: *Cognitive Science* 7(4): 243-280

Kolodner, J. (1993) *Case-Based Reasoning*. Mogan Kauffman, San Mateo, USA

Kolodner, J. (1996) Making the implicit explicit: Clarifying the principles of case-based reasoning. In: *Case-Based Reasoning: Experiences, Lessons & Future Directions*, Leake, D.B. (Ed.), pp. 349-370, AAAI Press, Menlo Park, USA

Koton, P. (1989) *Using experience in learning and problem solving.* PhD thesis, Massachusetts Institute of Technology, Cambridge, USA

Kowalski, R.A. (1979) *Logic for Problem Solving.* North-Holland, Amsterdam, NL

Koza, J. (1992) *Genetic Programming: On the programming of computers by means of natural selection.* MIT Press, Cambridge, USA

Kshirsagar, S. and Thalmann, N.M. (2000) Multimedia communication with virtual humans. In: *Proceedings of EUROMEDIA*, pp. 3-10. SCS International, Ghent, Belgium

Kushmerick, N. (1999) Gleaning the Web. In: *IEEE Intelligent Systems and Their Applications* **14**(2): 20-22

Labrou, Y., Finin, T. and Peng, Y. (1999) Agent communication languages: The current landscape. In: *IEEE Intelligent Systems and Their Applications* **14**(2): 45-52

Lam, K.M. and Yan, H. (1998) An analytic-to-holistic approach for face recognition based on a single frontal view. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(7): 673-686

Lavrac, N. and Dzeroski, S. (1994) *Inductive logic programming: Techniques and applications.* Ellis Horwood, Chichester, UK

Law, T., Itoh, H. and Seki, H. (1994) Image filtering, Edge Detection and Edge Tracing using Fuzzy Reasoning. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18** (5): 481-491

Leake, D.B. (1996) CBR in context: The present and future. In: *Case-Based Reasoning: Experiences, Lessons & Future Directions*, Leake, D.B. (Ed.), pp. 3-30, AAAI Press, Menlo Park, USA

Li, H. and Roivainen, P. (1993) 3D motion estimation in model-based facial image coding. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**(6): 545-555

Li, Y. and Zhao, Y. (1998) Recognizing emotions in speech using short-term and long-term features. In: *Proceedings of the International Conference on Spoken Language Processing*, pp. 2255-2258

Lien, J.J., Kanade, T., Cohn, J.F. and Li, C.C. (1998) Automated facial expression recognition based on FACS action units. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 390-395

Lopez, B. and Plaza, E. (1993) Case-based planning for medical diagnosis. In: *ISMIS-93: Lecture Notes in Artificial Intelligence* **689**: 96-105

Lucas, B.D. and Kanade, T. (1981) An iterative image registration technique with an application to stereo vision. In: *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 674-680

Lusted, H.S. and Knapp, R.B. (1996) Controlling computers with neural signal. In: *Scientific American* **275**(4): 82-87

Lyons, M.J., Akamatsu, S., Kamachi, M. and Gyoba, J. (1998) Coding facial expressions with Gabor wavelets. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 200-205

Lyons, M.J., Budynek, J. and Akamatsu, S. (1999) Automatic classification of single facial images. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(12): 1357-1362

Maes, P. (1994) Agents that reduce work and information overload. In: *Communications of the ACM* **37**(7): 31-40

Maher, M.L. and Zhang, D.M. (1993) CADSYN: A Case-Based Design Process Model. In: *Artificial Intelligence in Engineering, Design, and Manufacturing* 7(2): 97-110

Malciu, M. and Preteux, F. (2000) A robust model-based approach for 3D head tracking in video sequences. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 169-174

Malek, M., Toitgans, M.P., Wybo, J.L. and Vincent, M. (1998) An operator support system based on case-based reasoning for the plastic moulding injection process. In: *Proceedings of the European Workshop on Case-Based Reasoning*, pp. 402-413

Mandal, M.K. and Palchoudhury, S. (1986) Choice of facial affect and psychopathology: A discriminatory analysis. In: *Journal of Social Behaviour and Personality* **1**(2): 299-302

Mann, S. (1997) Wearable computing: A first step toward personal imaging. In: *Computer* **30**(2): 25-32

Mark, W., Simoudis, E. and Hinkle, D. (1996) Case-based reasoning: Expectations and results. In: *Case-Based Reasoning: Experiences, Lessons & Future Directions*, Leake, D.B. (Ed.), pp. 269-294, AAAI Press, Menlo Park, USA

Marsic, I., Medl, A. and Flanagan, J. (2000) Natural communication with information systems. In: *Proceedings of the IEEE* **88**(8): 1354-1366

Mase, K. (1991) Recognition of facial expression from optical flow. In: *IEICE Transactions* **E74**(10): 3474-3483

Matsumoto, D. (1990) Cultural similarities and differences in display rules. In: *Motivation and Emotion* **14**: 195-214

Matsumura, K., Nakamura, Y. and Matsui, K. (1997) Mathematical representation and image generation of human faces by metamorphosis. In: *Electronics and Communications in Japan, Part 3* **80**(1): 36-46

Matsuno, K., Lee, C.W. and Tsjui, S. (1993) Recognition of facial expression with potential net. In: *Proceedings of the Asian Conference on Computer Vision*, pp. 504-507

McCarty, J. and Hayes, P. (1969) Some philosophical problems from the standpoint of artificial intelligence. In: *Machine Intelligence 4*, Meltzer, B. and Michie, D. (Eds.), pp. 463-502. Edinburgh University Press, Edinburgh, UK

McCown, W.G., Johnson, J.L. and Austin, S.H. (1988) Patterns of facial affect recognition errors in delinquent adolescent males. In: *Journal of Social Behaviour and Personality* 3(3): 215-224

McCracken, T.O., Ed. (1999) *New Atlas of Human Anatomy*. Lustre Press Ltd. and Roli Books Ltd., New Delhi, India

McHugo, G.J., Lanzetta, J.T., Sullivan, D.G., Masters, R.D. and Englis, B.G. (1985) Emotional reactions to a political leader's expressive displays. In: *Journal of Personality and Social Psychology* **49**: 1513-1529

McKinley, B.L. and Whipple, G.H. (1997) Model based speech pause detection. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1179 - 1182

McNeill, D. (1992) *Hand and Mind: What Gestures Reveal About Thought*. Chicago University Press, Chicago, USA

Mehrabian, A. (1968) Communication without words. In: *Psychology Today* **2**(4): 53-56

Meier, U., Hurst, W. and Duchnowski, P. (1996) Adaptive bimodal sensor fusion for automatic speech-reading. In: *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing*, pp. 833-836

Mesquita, B. and Frijda, N.H. (1992) Cultural variations in emotions: A review. In: *Psychological Bulletin* **112**: 179-204

Minsky, M. (1986) *The Society of Mind*. Simon and Schuster, New York, USA

Mitchell, T.M. (1996) *An introduction to genetic algorithms*. MIT Press, Cambridge, USA

Mitchell, T.M. (1997) *Machine Learning*. McGraw-Hill Companies Inc., Singapore

Moccozet, L. and Thalmann, N.M. (1997) Dirichlet free form deformations and their application to hand simulation. In: *Proceedings of the International Conference on Computer Animation*, pp. 93-102

Morimoto, C.H., Koons, D., Amir, A. and Flickner, M. (2000) Pupil detection and tracking using multiple light sources. In: *Image and Vision Computing Journal* **18**(4): 331-335

Morishima, S. and Harashima, H. (1993) Emotion Space for Analysis and Synthesis of Facial Expression. In: *Proceedings of the International Workshop on Robot and Human Communication*, pp. 188-193

Morishima, S., Kawakami, F., Yamada, H. and Harashima, H. (1995) A Modelling of Facial Expression and Emotion for Recognition and Synthesis. In: *Symbiosis of Human and Artifact*, Anzai, Y., Ogawa, K. and Mori, H. (Eds.), pp. 251-256, Elsevier Science BV, Amsterdam, NL

Moses, Y., Reynard, D. and Blake, A. (1995) Determining facial expressions in real time. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 332-337

Moulin, B. and Chaib-Draa, B. (1996) An Overview of Distributed Artificial Intelligence. In: *Foundations of Distributed Artificial Intelligence*, O'Hare, G.M.P. and Jennings, N.R. (Eds.), pp. 3-55, John Wiley & Sons, New York, USA

Moulton, M. (1998) Success comes from experiencing failure. In: *Proceedings of the International Conference of the British Computer Society Specialist Group on Expert Systems*, pp. 263-274

Muggleton, S. (Ed.) (1992) *Inductive logic programming*. Academic Press, London, UK

Murray, I.R. and Arnott, J.L. (1996) Synthesizing emotion in speech: Is it time to get excited? In: *Proceedings of the International Conference on Spoken Language Processing*, pp. 1816-1819

Myers, B.A. (1996) *A brief history of human computer interaction technology*. Technical report (CMU-CS-TR-96-163), Carnegie-Mellon University, Pittsburgh, USA

Nakatsu, R. (1998) Toward the creation of a new medium for the multimedia era. In: *Proceedings of the IEEE* **86**(5): 825-836

Nakatsu, R., Nicholson, J. and Tosa, N. (2000) Emotion recognition and its application to computer agents with spontaneous interactive capabilities. In: *Knowledge-Based Systems* **13** (7-8): 497-504

Nasman, V.T., Calhoun, G.L. and McMillan, G.R. (1997) Brain-actuated control and HMDS. In: *Head Mounted Displays*, Melzer, J. and Moffitt, K. (Eds.), pp. 285-312, McGraw-Hill, New York, USA

Navinchandra, D. (1991) Exploration and innovation in design: Towards a computational model. Springer-Verlag, New York, USA

Netten, B.D. (1997) *Knowledge Based Conceptual Design: An Application to Fibre Reinforced Composite Sandwich Panels*. PhD thesis, Delft University of Technology, Delft, NL

Nielsen, J. (1995) *Multimedia and Hypertext: The Internet and Beyond*. Academic Press, Cambridge, USA

O'Hare, G.M.P. and Jennings, N.R. (Eds.) (1996) *Foundations of Distributed Artificial Intelligence*. John Wiley & Sons, New York, USA

O'Rourke, J. (1994) *Computational Geometry in* C. Cambridge University Press, Cambridge, UK

Oakley, I., McGee, M.R., Brewster, S. and Gray, P. (2000) Putting the feel in 'look and feel'. In: *Proceedings of the ACM International Conference on Computer-Human Interaction*, pp. 415-422

Oatley, K. and Jenkins, J.M. (1996) *Understanding Emotions*. Blackwell, Oxford, UK

Olson, J.S. and Olson, G.M. (2000) Trust in e-commerce. In: *Communications of the ACM* **43**(12): 41-44

Ortony, A. and Turner, T.J. (1990) What is basic about basic emotions? In: *Psychological Review* **74**: 315-341

Orwell G. (1949) 1984. Secker & Warburg, London, UK

Otsuka, T. and Ohya, J. (1996) Recognition of facial expressions using HMM with continuous output probabilities. In: *Proceedings of the International Workshop on Robot and Human Communication*, pp. 323-328

Otsuka, T. and Ohya, J. (1998) Spotting segments displaying facial expression from image sequences using HMM. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 442-447

Oviatt, S., DeAngeli, A. and Kuhn, K. (1997) Integration and synchronisation of input modes during multimodal human-computer interaction. In: *Proceedings of the ACM international Conference on Computer Human Interaction*, pp. 415-422

Oviatt, S. (2000) Taming recognition errors with a multimodal interface. In: *Communications of the ACM* **43**(9): 45-51

Owens, C. (1993) Integrating feature extraction and memory search. In: *Machine Learning* **10**(3): 311-340

Padgett, C. and Cottrell, G.W. (1996) Representing face images for emotion classification. In: *Proceedings of the International Conference on Advances in Neural Information Processing Systems*, pp. 894-900

Pan, H., Liang, Z.P., Anastasio, T.J. and Huang, T.S. (1999) Exploiting the dependencies in information fusion. In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* **2**: 407-412

Pantic, M. (1996) *Human Emotion Recognition Clips Utilised Expert System (HERCULES).* M.Sc. thesis, Delft University of Technology, Faculty of Technical Mathematics and Informatics, Department of Knowledge Based Systems, Delft, NL

Pantic, M., Djordjevic, A., Rothkrantz, L.J.M. and Koppelaar, H. (1998a) Computer Assisted Study Planning. In: *Proceedings of EUROMEDIA*, pp. 259-263. SCS International, Ghent, Belgium

Pantic, M., Rothkrantz, L.J.M. and Koppelaar, H. (1998b) Automation of Non-Verbal Communication of Facial Expressions. In: *Proceedings of EUROMEDIA*, pp. 86-93. SCS International, Ghent, Belgium

Pantic, M. and Rothkrantz, L.J.M. (1999a) Ambiguous Data in Automated Facial Expression Recognition. In: *Proceedings of the 5th Annual Conference of ASCI*, pp. 125-132. ASCI Press, Delft, NL

Pantic, M. and Rothkrantz, L.J.M. (1999b) An Expert System for Multiple Emotional Classification of Facial Expression. In: *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, pp. 113-120

Pantic, M. and Rothkrantz, L.J.M. (2000a) An Expert System for Recognition of Facial Actions and Their Intensity. In: *Proceedings of the 12th AAAI International Conference on Innovative Applications of Artificial Intelligence*, pp. 1026-1033

Pantic, M. and Rothkrantz, L.J.M. (2000b) Expert System for Automatic Analysis of Facial Expressions. In: *Image and Vision Computing Journal* **18**(11): 881-905

Pantic, M. and Rothkrantz, L.J.M. (2000c) Self-adaptive Expert System for Facial Expression Analysis. In: *Proceedings of the IEEE International Conference on System, Man and Cybernetics*, pp. 73-79

Pantic, M. and Rothkrantz, L.J.M. (2000d) Automatic Analysis of Facial Expressions: the State of the Art. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(12): 1424-1445

Pantic, M. and Rothkrantz, L.J.M. (2001a) Affect-sensitive Multi-modal Monitoring in Ubiquitous Computing: Advances and Challenges. In: *Proceedings of the*

IEEE / AAAI International Conference on Enterprise Information Systems, pp. 466-474

Pantic, M., Tomc, M. and Rothkrantz, L.J.M. (2001b) A hybrid approach to mouth features detection. In: *Proceedings of the IEEE International Conference on System, Man and Cybernetics*, to appear

Parke, F.I. (1972) Computer generated animation of faces. In: *Proceedings of the ACM National Conference* **1**, pp. 451-457

Parke, F.I. (1974) *A parametric model for human faces*. Technical report (# UTEC-CSc-75-047), University of Utah, Salt Lake City, USA

Patras, I. (2001) *Object-based video segmentation with region labelling*. PhD thesis, Delft University of Technology, Delft, NL

Pavlovic, V.I., Sharma, R. and Huang, T.S. (1997) Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(7): 677-695

Pentland, A.P., Moghaddam, B. and Starner, T. (1994) View-based and modular eigenspaces for face recognition. In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 84-91

Pentland, A. (2000) Looking at people: Sensing for ubiquitous and wearable computing. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(1): 107-119

Perry, T. (2001) Service takes over in the networked world. In: *IEEE Spectrum* **38**(1): 102-110

Petrushin, V.A. (2000) Emotion recognition in speech signal: Experimental study, development, and application. In: *Proceedings of the International Conference on Spoken Language Processing*, pp. 222-225

Picard, R.W. (1997) *Affective computing*. MIT Press, Cambridge, Massachusetts, USA

Picard, R. and Healey, J. (1997) *Affective Wearables*. Technical Report (# 467), Media Laboratory, Massachusetts Institute of Technology, Cambridge, USA

Polzin, T.S. (2000) *Detecting verbal and non-verbal cues in the communications of emotions*. PhD thesis, Carnegie Mellon University, Pittsburgh, USA

Porter, B.W. and Bareiss, E.R. (1986) PROTOS: Experiment in knowledge acquisition for heuristic classification tasks. In: *Proceedings of the 1st International Meeting on Advances in Learning*, pp. 159-174

Pratakanis, A.R., Breckler, S.J. and Greenwald, A.G. (Eds.) (1989) *Attitude Structure and Function*. Erlbaum, Hillsdale, USA

Preece, J. and Schneiderman, B. (1995) Survival of the fittest: Evolution of multimedia user interfaces. In: *ACM Computing Surveys* **27**(4): 557-559

Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1992) *Numerical Recipes in C*. Cambridge University Press, New York, USA

Prkachin, K.M. and Mercer, S.R. (1989) Pain expression in patients with shoulder pathology: validity properties and relationship to sickness impact. In: *Pain* **39**: 257-265

Profijt, W.F. (1995) *Automated assessment of emotions*. M.Sc. thesis, Delft University of Technology, Faculty of Technical Mathematics and Informatics, Department of Knowledge Based Systems, Delft, NL

Putnam, W. and Knapp, R.B. (1993) Real-time computer control using pattern recognition of the electromyogram. In: *Proceedings of the IEEE International Conference on Engineering in Medicine and Biology Society* **15**: 1236-1237

Quinlan, J.R. (1979) *Induction over large databases*. Technical Report (TR-HPP-79-14), Computer Science Department, Stanford University, Stanford, USA

Quinlan, J.R. (1993) *Programs for Machine Learning*. Morgan Kaufmann, San Mateo, USA

Raducanu, B., Pantic, M., Rothkrantz, L.J.M. and Grana, M. (1999) Automatic Eyebrow Tracking Using Boundary Chain Code. In: *Proceedings of the 5th Annual Conference of ASCI*, pp. 137-143. ASCI Press, Delft, NL

Rahardja, A., Sowmya, A. and Wilson, W.H. (1991) A neural network approach to component versus holistic recognition of facial expressions in images. In: *SPIE vol. 1607 Intelligent robots and computer vision X: algorithms and techniques*, pp. 62-70

Ralescu, A. and Hartani, R. (1995) Some issues in fuzzy and linguistic modelling. In: *Proceedings of the International Conference on Fuzzy Systems*, pp. 1903-1910

Reeves, B. and Nass, C.I. (1996) *The Media Equation: How people treat computers, television, and new media like real people and places.* Cambridge University Press, New York, USA

Reinders, M.J.T. (1997) Eye Tracking by Template Matching using an Automatic Codebook Generation Scheme. In: *Proceedings of the 3rd Annual Conference of ASCI*, pp. 85-91. ASCI Press, Delft, NL

Richter, M.M. (1995) On the notion of similarity in case-based reasoning. In: *Mathematical and Statistical Methods in Artificial Intelligence*, della Riccia, G., Kruse, R., Viertl, R. (Eds.), pp. 171-184, Springer-Verlag, Heidelberg, Germany

Riedmiller, M. and Braun, H. (1993) A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In: *Proceedings of the International Conference on Neural Networks*, pp. 586-591

Ritter, G.X. and Wilson, J.N. (1996) *Handbook of Computer Vision Algorithms in Image Algebra.* CRC Press, Boca Raton, USA

Rosenblatt, F. (1958) The Perceptron: A probabilistic model for information storage and organisation in the brain. In: *Psychological Review* **65**: 386-408

Rosenblum, M., Yacoob, Y. and Davis, L. (1994) Human emotion recognition from motion using a radial basis function network architecture. In: *Proceedings of the IEEE Workshop on Motion of Non-rigid and Articulated Objects*, pp. 43-49

Rosenschein, J.S. and Krulwich, B. (1999) AgentSoft. In: *IEEE Intelligent Systems and Their Applications* **14**(2): 2-3

Rothkrantz, L.J.M., van Schouwen, M., Ververs, F. and Vollering, J. (1998) A Multimedial Workbench for Facial Expression Analysis. In: *Proceedings of EUROMEDIA*, pp. 94-101. SCS International, Ghent, Belgium

Rowley, H.A., Baluja, S. and Kanade, T. (1998) Neural network based face detection. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(1): 23-38

Roy, D. and Pentland, A. (1997) *Multimodal Adaptive Interfaces.* Technical Report (# TR-97-438), Perceptual Computing Group, Massachusetts Institute of Technology, Cambridge, USA

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) Learning internal representations by error propagation. In: *Parallel Distributed Processing:*

Explorations in the Microstructure of Cognition I, Rumelhart, D.E. and McClelland, J.L. (Eds.), pp. 318-362, MIT Press, Cambridge, USA

Rumelhart, D., Widrow, B. and Lehr, M. (1994) The basic ideas in neural networks. In: *Communications of the ACM* **37**(3): 87-92

Russell, J. (1991) Rejoinder to Ekman, O'Sullivan and Matsumoto. In: *Motivation and Emotion* **15**: 177-184

Russell, J.A. (1994) Is there universal recognition of emotion from facial expression? In: *Psychological Bulletin* **115**(1): 102-141

Russell, J.A. and Fernandez-Dols, J.M. (Eds.) (1997) *The Psychology of Facial Expression.* Cambridge University Press, New York, USA

Saborido, M.T. (1992) *An introduction to expert system development.* Cataluna University of Technology Press, Barcelona, Spain

Salovey, P. and Mayer, J.D. (1990) Emotional intelligence. In: *Imagination, Cognition and Personality* **9**(3): 185-211

Samal, A. (1991) Minimum resolution for human face detection and identification. In: *SPIE Vol. 1453 Human Vision, Visual Processing, and Digital Display II*, pp. 81-89

Samal, A. and Iyengar, P.A. (1992) Automatic recognition and analysis of human faces and facial expressions: A survey. In: *Pattern Recognition* **25**(1): 65-77

Samuel, A.L. (1959) Some studies in machine learning using the game of checkers. In: *IBM Journal of Research and Development* **3**: 211-229

Schachter, J. (1957) Pain, fear and anger in hypertensives and normotensives: psycho-physiological study. In: *Psychosomatic Medicine* **19**: 17-29

Schank, R.C. (1982) *Dynamic memory: A theory of reminding and learning in computers and people.* Cambridge University Press, Cambridge, UK

Schank, R.C. (1984) *Memory-based expert systems.* Technical Report (# AFOSR. TR. 84-0814), Yale University, New Haven, USA

Schiano, D.J., Ehrlich, S.M., Rahardja, K. and Sheridan, K. (2000) Face to interface: Facial affect in human and machine. In: *Proceedings of the ACM International Conference on Computer Human Interaction*, pp. 193-200

Schmidt, R. and Gierl, L. (1998) Experiences with prototype designs and retrieval methods in medical case-based reasoning systems. In: *Proceedings of the European Workshop on Case-Based Reasoning*, pp. 370-381

Schneider, M., Kandel, A., Langholz, G. and Chew, G. (1996) *Fuzzy Expert System Tools*. John Wiley and Sons, Chichester, UK

Schneiderman, B. (1993) A nonanthropomorphic style guide: Overcoming the humpty dumpty syndrome. In: *Sparks of Innovation in Human-Computer Interaction*, Schneiderman, B. (Ed.), §7.1, Ablex Publishers, Norwood, USA

Schneiderman, B. (1993b) Human values and the future of technology: A declaration of responsibility. In: *Sparks of Innovation in Human-Computer Interaction*, Schneiderman, B. (Ed.), §7.2, Ablex Publishers, Norwood, USA

Schneiderman, B. (2000) Universal Usability. In: *Communications of the ACM* **43**(5): 85-91

Schneiderman, B. (2001) CUU: Bridging the Digital Divide with Universal Usability. In: *ACM Interactions* **8**(2): 11-15

Scholsberg, H. (1954) Three dimensions of emotion. In: *The Psychological Review* **61**(2): 81-88

Schouwen, M. (1998) *A Workbench for Automatic Recognition of Facial Expressions*. M.Sc. thesis, Delft University of Technology, Faculty of Technical Mathematics and Informatics, Department of Knowledge Based Systems, Delft, NL

Schrer, K.R. and Banse, R. (1996) Acoustic profiles in vocal emotion expression. In: *Personality and Social Psychology* **70**: 614-636

Shachter, R.D., Levitt, T.S. Kanal, L.N. and Lemmer, J.F. (Eds.) (1990) *Uncertainty in Artificial Intelligence 4*. North Holland, Amsterdam, NL

Shafer, G. (1990) The Meaning of Probability. In: *Readings in Uncertain Reasoning*, Shafer, G. and Pearl, J. (Eds.), pp. 7-13. Morgan Kaufmann, San Mateo, USA

Shafer, G. and Pearl, J. (Eds.) (1990) *Readings in Uncertain Reasoning*. Morgan Kaufmann, San Mateo, USA

Shafer, G. and Srivastava, R. (1990) The Bayesian and belief-functions formalisms – A general perspective for auditing. In: *Readings in Uncertain Reasoning*, Shafer, G. and Pearl, J. (Eds.), pp. 482-521, Morgan Kaufmann, San Mateo, USA

Shah, R.P. (1988) JET-X: Jet Engine Troubleshooting Expert System. In: *Proceedings of the IEEE International Workshop on Artificial Intelligence for Industrial Applications*, pp. 135-139

Sharma, R., Pavlovic, V.I. and Huang, T.S. (1998) Toward multimodal human-computer interface. In: *Proceedings of the IEEE* **86**(5): 853-869

Shigeno, S. (1998) Cultural similarities and differences in recognition of audio-visual speech stimuli. In: *Proceeding of the International Conference on Spoken Language Processing*, pp. 281-284

Shoham, Y. (1999) What we talk about when we talk about software agents. In: *IEEE Intelligent Systems and Their Applications* **14**(2): 28-31

Shortliffe, E.H. (1976) *Computer-Based Medical Consultation: MYCIN*. Elsevier, New York, USA

Shortliffe, E.H. and Buchanan, B.G. (1990) A Model of Inexact Reasoning in Medicine. In: *Readings in Uncertain Reasoning*, Shafer, G. and Pearl, J. (Eds.), pp. 259-273, Morgan Kaufmann, San Mateo, USA

Sigman, M. and Capps, L. (1997) *Children with Autism: A Development Perspective*. Harvard University Press, Cambridge, USA

Silva, F.M. and Almeda, L.B. (1990) Accelerating Backpropagation. In: *Advanced Neural Computer*s, Eckmiller, R. (Ed.), pp. 151-158, Elsevier North Holland, Amsterdam, NL

Simoncelli, E.P. (1993) *Distributed representation and analysis of visual motion*. PhD thesis, Massachusetts Institute of Technology, Cambridge, USA

Simoudis, E. (1992) Using case-based retrieval for customer technical support. In: *IEEE Expert* 7(5): 7-13

Simpson, R.L. (1985) *A Computer Model of Case-Based Reasoning in Problem Solving: An Investigation in the Domain of Dispute Mediation*. Technical Report (# TR-GIT-ICS-85/18), Georgia Institute of Technology, Atlanta, USA

Sloman, A. (1994) Explorations in Design Space. In: *Proceedings of the 11th European Conference on Artificial Intelligence*, pp. 578-582

Smeraldi, F., Carmona, O. and Bigun, J. (2000) Saccadic search with Gabor features applied to eye detection and real-time head tracking. In: *Image and Vision Computing Journal* **18**(4): 323-329

Smyth, B. (1996) *Case-Based Design*. PhD thesis, Trinity College, Dublin, Ireland

Soborido, M.T. (1992) *An Introduction to Expert System Development*. Cataluna University of Technology, Barcelona, Spain

Steeneken, H.J.M. and Hansen, J.H.L. (1999) Speech under stress conditions: Overview of the effect on speech production and on system performance. In: *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing*, pp. 2079-2082

Steffens, J., Elagin, E. and Neven, H. (1998) PersonSpotter – fast and robust system for human detection, tracking and recognition. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 516-521

Steimer-Krause, E., Krause, R. and Wagner, G. (1990) Interaction regulations used by schizophrenics and psychosomatic patients: Studies of facial behaviour in dyadic interactions. In: *Psychiatry* **53**: 209-228

Stein, B. and Meredith, M.A. (1993) *The Merging of Senses*. MIT Press, Cambridge, USA

Stephenson, G.M., Ayling, K. and Rutter, D.R. (1976) The role of visual communication in social exchange. In: *Britain Journal of Social Clinical Psychology* **15**: 113-120

Stiefelhagen, R. and Yang, J. (1997) Gaze tracking for multimodal human-computer interaction. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2617 -2620

Stork, D.G. and Hennecke, M.E. (Eds.) (1996) *Speech-reading by Man and Machine: Data, Models and Systems*. NATO/Springer-Verlag, New York, USA

Strobel, N., Spors, S. and Rabenstein, R. (2001) Joint audio-video object localization and tracking. In: *IEEE Signal Processing Magazine* **18**(1): 22-31

Sturman, D.J. and Zeltzer, D. (1994) A survey of glove-based input. In: *IEEE Computer Graphics Applications Magazine* **14**(1): 30-39

Sung, K.K., and Poggio, T. (1998) Example-based learning for view-based human face detection. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(1): 39-51

Surma, J. and Tyburcy, J. (1998) A study on competence-preserving case replacing strategies in Case-Based Reasoning. In: *Proceedings of the European Workshop on Case-Based Reasoning – Advances in Case-Based Reasoning*, pp. 233-238

Suryanarayanan, S. and Reddy, N.R. (1997) EMG-based interface for position tracking and control in VR environments and teleoperation. In: *Presence: Teleoperators and Virtual Environment* **6**(3): 282-291

Sutton, R.S. and Barto, A.G. (1998) Reinforcement learning: An introduction. MIT Press, Cambridge, USA

Takeuchi, A. and Nagao, K. (1993) Communicative Facial Displays as a New Conversational Modality. In: *Proceedings of the ACM INTERCHI*, pp. 187-193

Tanaka, T. and Sueda, N. (1988) Knowledge Acquisition in Image Processing Expert System EXPLAIN. In: *Proceedings of the IEEE International Workshop on Artificial Intelligence for Industrial Applications*, pp. 267-272

Tekalp, A.M. (Ed.) (1998) *Special Issue on Multimedia Signal Processing, Proceedings of the IEEE* **86**(5): 751-1014

Teller, A. and Veloso, M. (1994) PADO: A new learning architecture for object recognition. In: *Symbolic Visual Learning*, Ikeuchi, K. and Veloso, M. (Eds.), pp. 81-116, Oxford University Press, Oxford, UK

Terrillon, J.C., David, M. and Akamatsu, S. (1998) Automatic detection of human faces in natural scene images by use of a skin colour model of invariant moments. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 112-117

Terzopoulos, D. and Waters, K. (1993) Analysis and Synthesis of Facial Image Sequences Using Physical and Anatomical Models. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**(6): 569-579

Thalmann, N.M. and Thalmann, D. (1992) Six hundred indexed references on computer animation. In: *Journal of Visualisation and Computer Animation* **3**: 147-174

Thalmann, N.M., Kalra, P. and Pandzic, I.S. (1995) Direct face-to-face communication between real and virtual humans. In: *International Journal on Information Technology* **1**(2): 145-157

Thalmann, N.M., Kalra, P. and Escher, M. (1998) Face to Virtual Face. In: *Proceedings of the IEEE* **86**(5): 870-883

Thalmann, N.M. and Moccozet, L. (1998) Virtual Humans on Stage. In: *Virtual Worlds: Synthetic Universes, Digital Life and Complexity*, Heudin, J.C. (Ed.), pp. 95-126, Perseus books, Reading, USA

Thalmann, N.M. and Kshirsagar, S. (2000) Multimedia communication with virtual humans. In: *Proceedings of Euromedia*, pp. 3-10, SCS Press, Gent, Belgium

Thrun, S.B. (1991) The Monk's problems: A performance comparison of different learning algorithms. Technical Report (TR-CMU-CS-91-197), Carnegie Mellon University, Pittsburgh, USA

Tian, Y., Kanade, T. and Cohn, J.F. (2001) Recognising action units for facial expression analysis. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(2): 97-115

Tosa, N. and Nakatsu, R. (1996) Life-like Communication Agent – Emotion Sensing Character MIC and Feeling Session Character MUSE. In: *Proceedings of the International Conference on Multimedia Computing and Systems*, pp. 12-19

Turkle, S. (1984) *The Second Self.* Simon and Schuster, New York, USA

Tversky, A. and Kahneman, D. (1990) Judgement under uncertainty: heuristics and biases. In: *Readings in Uncertain Reasoning*, Shafer, G. and Pearl, J. (Eds.), pp. 32-39, Morgan Kaufmann, Los Altos, USA

Ushida, H., Takagi, T. and Yamaguchi, T. (1993) Recognition of facial expressions using conceptual fuzzy sets. In: *Proceedings of the International Conference on Fuzzy Systems 1*, pp. 594-599

van Dam, A. and Foley, J.D. (1995) *Computer graphics – algorithms and principles.* Addison-Wesley, Reading, USA van Dyke Parunak, H. (1996) Applications of Distributed Artificial Intelligence in Industry. In: *Foundations of Distributed Artificial Intelligence*, O'Hare, G.M.P. and Jennings, N.R. (Eds.), pp. 139-167, John Wiley & Sons, New York, USA

van Gelder, R.S. and van Gelder, L. (1990) Facial expression and speech: Neuroanatomical considerations. In: *International Journal of Psychology* **25**(2): 141-155

van Poecke, L. (1996) *Nonverbal Communication*. Garant-Uitgevers, Apeldoorn, NL

van Vark, R.J., Rothkrantz, L.J.M. and Kerckhoffs, E.J.H. (1995) Prototypes of Multimedia Stress Assessment. In: *Proceedings of the MediaComm*, pp. 108-112. SCS International, Ghent, Belgium

Vanger, P., Honlinger, R. and Haken, H. (1995) Applications of synergetics in decoding facial expression of emotion. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 24-29

Vaughn, K.B. and Lanzetta, J.T. (1980) Vicarious instigation and conditioning of facial expressive and autonomic responses to a model's expressive display of pain. In: *Journal of Personality and Social Psychology* **38**: 909-923

Viennet, E. and Soulie, F.F. (1992) Multiresolution scene segmentation by MLPs. In: *Proceedings of the International Joint IEEE / AAAI Conference on Neural Networks* **3**: 55-59

Vincent, J.M., Myers, D.J. and Hutchinson, R.A. (1992) Image feature location in multi-resolution images using a hierarchy of multi-layer perceptrons. In: *Neural Networks for Vision, Speech and Natural Language*, Lingard, R., Myers, D. and Nightingale, C. (Eds.), pp. 13-29, Chapman Hall, London, UK

Vollering, J. (1998) *A Workbench for Automatic Recognition of Facial Expressions*. M.Sc. thesis, Delft University of Technology, Faculty of Technical Mathematics and Informatics, Department of Knowledge Based Systems, Delft, NL

Vyzas, E. and Picard, R. (1999) Offline and online recognition of emotion expression from physiological data. In: *Proceedings of International Conference on Autonomous Agents - Workshop on Emotion-Based Agent Architectures*, pp. 135-142

Waibel, A., Vo, M.T., Duchnowski, P. and Manke, S. (1995) Multimodal interfaces. In: *Artificial Intelligence Review Journal* **10**(3-4): 299-319

Wang, M., Iwai, Y. and Yachida, M. (1998) Expression recognition from time-sequential facial images by use of expression change model. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition,* pp. 324-329

Watkins, C. (1989) *Learning from delayed rewards.* PhD thesis, King's College, Cambridge, UK

Watson, I., Basden, A. and Brandon, P.S. (1992) The client centred approach: expert system maintenance. In: *Expert Systems* 9(4): 189-196

Watson, I. and Abdulah, S. (1994) Developing case-based reasoning system: a case study in diagnosing building defects. In: *Proceedings of the IEE Colloquium on Case-Based Reasoning: Prospects and Applications 1,* pp. 1-3

Watson, I. and Marir, F. (1994) Case-base reasoning: A review. In: *The Knowledge Engineering Review* 9(4): 327-354

Weiss, S. and Kapouleas, I. (1989) An empirical comparison of pattern recognition, neural nets and machine learning classification methods. In: *Proceedings of the IEEE/ AAAI International Joint Conference on Artificial Intelligence*, pp. 781-787

Widrow, B. and Hoff, M.E. (1960) Adaptive switching circuits. In: *1960 IRE WESCON Convention Record* **4**: 96-104

Wielinga, B.J., Schreiber, A.T. and Breuker, J.A. (1992) KADS: A modelling approach to knowledge engineering. In: *Knowledge Acquisition* **4**(1): 5-53

Wierzbicka, A. (1993) Reading human faces. In: *Pragmatics and Cognition* **1**(1): 1-23

Williams, C.E. and Stevens, K.N. (1972) Emotions and speech: Some acoustic correlates. In: *Journal of Acoustic Society of America* **52**: 1238-1250

Williams, D.J. and Shah, M. (1992) A fast algorithm for active contours and curvature estimation. In: *Computer Vision and Image Processing: Image Understanding* **55**(1): 14-26

Wiskott, L. (1995) Labelled graphs and dynamic link matching for face recognition and scene analysis. In: *Reihe der Physik* **53**, Verlag Harri Deutsch, Thun, Frankfurt a. Main, Germany

Wojdel, A., Wojdel, J. and Rothkrantz, L.J.M. (1999) Dual-view Recognition of Emotional Facial Expressions. In: *Proceedings of the 5th Annual Conference of ASCI*, pp. 191-198. ASCI Press, Delft, NL

Wojdel, J. and Rothkrantz, L.J.M. (1998) Mixed Fuzzy-system and Artificial Neural Network Approach to the Automated Recognition of Mouth Expressions. In: *Proceedings of the 8th International Conference on Artificial Neural Networks*, pp. 833-838

Wojdel, J., Wojdel, A. and Rothkrantz, L.J.M. (1999) Analysis of Facial Expressions Based on Silhouettes. In: *Proceedings of the 5th Annual Conference of ASCI*, pp. 199-206. ASCI Press, Delft, NL

Wojdel, J. and Rothkrantz, L.J.M. (2000) Visually based speech onset/offset detection. In: *Proceedings of Euromedia*, pp. 156-160. SCS International, Ghent, Belgium

Wu, H., Yokoyama, T., Pramadihanto, D. and Yachida, M. (1996) Face and facial feature extraction from colour image. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 345-350

Yacoob, Y. and Davis, L. (1994a) Recognising facial expressions by spatio-temporal analysis. In: *Proceedings of the IEEE International Conference on Pattern Recognition 1*, pp. 747-749

Yacoob, Y. and Davis, L. (1994b) Computing spatio-temporal representations of human faces. In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 70-75

Yager, R.R. (1988) Uncertain associational relations: compatibility and transition relations in reasoning. In: *Lecture Notes in Economics and Mathematical Systems* **310** (Combining Fuzzy Imprecision with Probabilistic Uncertainty in Decision Making), Kacprzyk, J. and Fedrizzi, M. (Eds.), pp. 152-167, Springer Verlag, New York, USA

Yamada, H. (1993) Visual Information for Categorising Facial Expressions of Emotions. In: *Applied Cognitive Psychology* 7: 257-270

Yamakawa, T. (1990) Pattern recognition hardware system employing a fuzzy neuron. In: *Proceedings of the International Conference on Fuzzy Logic and Neural Networks*, pp. 943-948

Yang, J. and Waibel, A. (1996) A real-time face tracker. In: *Workshop on Applications of Computer Vision*, pp. 142-147

Yang, J., Stiefelhagen, R., Meier, U. and Waibel, A. (1998) Visual tracking for multimodal human computer interaction. In: *Proceedings of the ACM International Conference on Computer Human Interaction*, pp. 140-147

Yang, S. and Robertson, D. (1994) A case-based reasoning system for regulatory information. In: *Proceedings of the IEE Colloquium on Case-Based Reasoning: Prospects and Applications 3*, pp. 1-3

Yoneyama, M., Iwano, Y., Ohtake, A. and Shirai, K. (1997) Facial expressions recognition using discrete Hopfield neural networks. In: *Proceedings of the IEEE International Conference on Image Processing* **3**: 117-120

Yuille, A.L., Cohen, D.S. and Hallinan, P.W. (1989) Feature extraction from faces using deformable templates. In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 104-109

Yuille, A.L., and Pentland, A. (1993) Breakout Group on Sensing and Processing. In: *Final Report to NSF of the Planning Workshop on Facial Expression Understanding*, Ekman, P., Huang, T.S., Sejnowski, T.J. and Hager, J.C. (Eds.), pp. 38-47, Technical Report, Human Interaction Laboratory, University of California, San Francisco, USA

Zadeh, L.A. (1965) Fuzzy sets. In: *Information and Control* **8**: 338-353

Zadeh, L.A. (1975) Fuzzy logic and approximate reasoning. In: *Synthese* **30**: 407-428

Zadeh, L.A. (1978) Fuzzy sets as a basis for a theory of possibility. In: *Fuzzy Sets and Systems* **1**: 3-28

Zadeh, L.A. (1983) The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems. In: *Fuzzy Sets and Systems* **11**: 199-227

Zhang, Y. and Kambhamettu, C. (2000) Robust 3D head tracking under partial occlusion. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 176-182

Zhang, Z., Lyons, M., Schuster, M. and Akamatsu, S. (1998) Comparison between geometry-based and Gabor wavelets-based facial expression recognition using multi-layer perceptron. In: *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 454-459

Zhao, J. and Kearney, G. (1996) Classifying facial emotions by backpropagation neural networks with fuzzy inputs. In: *Proceedings of the International Conference on Neural Information Processing* **1**: 454-457

Zurada, J. (1992) *Introduction to Artificial Neural Systems*. West Publishing Company, St Paul, USA

Summary

Facial Expression Analysis by Computational Intelligence Techniques

Maja Pantic

The automatic interpretation of human communicative behaviour is concerned with giving machines the ability to detect, identify, and understand human interactive cues (speech, vocal utterances, facial expressions, body gestures). This research topic has become central in machine vision research, natural language processing research and in AI research in general. The reason behind this recent upsurge of interest in the research topic of *human-centred computing* is that the automation of the analysis of human communicative behaviour is essential for the design of future smart environments, perceptual user interfaces, and ubiquitous computing in general. The key technical goals concern determining the context in which the user acts (i.e. disclosing in an automatic way where the user is, what he is doing, and how he is feeling) and redesigning user interfaces to computers so that machines become "aware" of people that interact with them and become capable of acting properly in a context-sensitive manner.

This thesis pertains to one of these issues: providing machines with the ability to analyse a user's *facial expressions* and interpret them in terms of displayed *affective/ attitudinal states*.

Humans detect and interpret faces and facial expressions in a scene with little or no effort. Nevertheless, the development of an automated system that accomplishes this task is rather difficult. There are several related problems: finding face(s) in an input image, detecting facial features, and classifying the observed facial expression (e.g.

according to the shown affective state). In this thesis the past attempts to tackle these problems are surveyed and the state of the art of research activities in the field of machine monitoring of facial signals is summarised (chapter 2).

Overall, the existing automated facial expression analysers classify input facial expressions either in terms of facial actions (i.e. Action Units (AUs) defined in the FACS system introduced by Ekman and Friesen (1978)) or in terms of affective (emotional) states. As far as the existing automated systems for AU recognition are considered, they are capable of encoding small sets of AUs and their combinations (in the best case 16 AUs) and none is capable of quantified facial action encoding from input facial images. As far as the existing automated facial affect analysers are concerned, they classify facial information into the six basic emotion categories (Ekman and Friesen 1975) even though this classic study on six basic emotions is nowadays strongly challenged in the basic research literature.

In contrast to this incapability of current systems, the *main contribution* of this thesis is that it proposes a novel method for automatic facial expression analysis of static facial images in terms of 32 quantified AU codes (chapter 5) and multiple quantified (facial affect) interpretation labels learned from the current user (chapter 6). To wit, the Integrated System for Facial Expression Recognition (ISFER), proposed in this thesis, represents a step further in the automation of FACS coding of facial imagery and a step further in the creation of a universally usable automated tool for facial affect analysis which is independent of the psychological debates on emotion. To our best knowledge, ISFER represents the first attempt (proposed in the literature up to date) to automate:

- Encoding of 32 AU codes and their combinations from an input facial image.
- Quantification of the encoded AU codes (in terms of the intensity of AUs' activation) in a subject-dependent manner.
- User-adaptive classification of facial data in (facial affect) interpretation categories.
- Quantification of the scored interpretation labels based upon the activation intensity levels associated with the scored AUs that produced the pertinent facial expression.
- Generation of the conclusions so that the associated certainties vary in accordance with the input data certainties.

The proposed method for automatic, (observed-subject- and user-) adaptive, affect-sensitive, facial expression analysis from static images of faces brings together two fundamentally diverse *technologies*: psychologically and anatomically based FACS and computational intelligence. ISFER employs:

- Various *image processing techniques* such as active contours, fuzzy edge detection, and artificial neural networks, in order to spatially sample the contours of the prominent facial features (including eyebrows, eyes, nostrils, mouth, chin, and profile) from an input static image of the face (chapter 4).

- A rule-based expert system that performs *reasoning with uncertainty* about the AUs and their activation intensity shown in the currently examined facial image (chapter 5).
- A memory-based expert system that applies *case-based reasoning* while expounding the encoded quantified AU codes in terms of interpretation labels *learned* from the current user (chapter 6).

Validation studies on ISFER (chapter 7) suggest that the conclusions achieved by the system are generally consistent with those of human observers who judged the same images. Those studies also indicate that ISFER's performance is satisfactory acceptable for the purposes of behavioural investigations of the face. The execution of the system's code is rather time consuming, however, which makes the system still unsuitable for the purposes of perceptual human-computer interfaces, where real-time performance is necessary since delays make the interaction desynchronised and unnatural. Nevertheless, the integration of image processing techniques that perform in real time could make ISFER a suitable tool that could be employed for monitoring and interpretation of facial signals in most of the application domains mentioned above.

Though acceptable, ISFER's *performance can be improved* at several points. The two most important ones are:

- ISFER cannot handle distractions like glasses and facial hair and therefore its analysis is limited to faces without a beard, moustache, and glasses. Also, ISFER cannot encode the full range of facial behaviour (i.e. of all 44 FACS AUs). These two limitations can be handled, at least partially, by accommodating the analysis of facial image sequences rather than the analysis of static images of faces.
- ISFER adopts an event approach: facial expressions of affective states are treated as context-free autobiographical events learned from the current user. This is a very constrained type of event where the context is limited to the accompanying facial actions displayed in a time instance by a particular subject (to whom the system adapts in order to quantify the displayed facial actions). However, the interpretation of a monitored subject's facial behaviour also depends on the subject's typical temporal course of facial behaviour given the environmental constraints in which the subject acts. Due to the complexity of this problem, handling ISFER's context insensitivity is probably the most significant challenge facing future developers of ISFER.

Samenvatting

Analyse van Gezichtsuitdrukkingen door Technieken uit de Computationele Intelligentie

Maja Pantic

Automatische interpretatie van menselijk communicatief gedrag zal machines in staat stellen om gedrags- en gemoedsuitingen te detecteren, te identificeren, en te begrijpen (bijv. spraak, stemuitingen, gezichtsuitdrukkingen en lichaamstaal zoals gebaren). Dit onderwerp is centraal komen te staan in onderzoek naar natuurlijke taalverwerking, automatische beeldverwerking en in AI onderzoek in het algemeen. De katalysator achter deze recente interesse in onderzoek naar *informatietechnologie waarin de mens centraal staat* (Engels: HCC) is dat geautomatiseerde analyse van menselijk communicatief gedrag essentieel is voor ontwerp van gebruikersinterfaces met waarnemingsvermogen en voor toekomstige intelligente omgevingen, kortom: voor alomtegenwoordige informatietechnologie in het algemeen. De twee technische hoofddoelen in HCC zijn: bepalen van de context waarin de gebruiker manoeuvreert (d.w.z. ontsluiten op een automatische wijze waar de gebruiker is, wat hij doet, en hoe hij zich voelt) en herontwerp van gebruikersinterfaces voor computers zodanig dat machines zich bewust worden van mensen waarmee zij communiceren en in staat zijn om op geëigende contextgevoelige wijze te reageren.
Dit proefschrift gaat over een van deze zaken van HCC: om machines in staat te stellen om gebruikers' *gezichtsuitdrukkingen* te analyseren en te interpreteren in termen van vertoonde *gemoedstoestanden.*

Mensen detecteren en interpreteren gezichten en gezichtsuitdrukkingen in beeld zonder (veel) moeite. Desalniettemin is ontwikkeling van een automatisch systeem, dat deze taak uitvoert, tamelijk moeilijk. Er zijn meerdere aanpalende problemen: het vinden van gezichten in een inputbeeld, detecteren van gezichtskenmerken en classificeren van de geobserveerde gezichtsuitdrukking (bijv. in overeenstemming met de uitgedrukte gemoedstoestand). Dit proefschrift geeft een overzicht van de elders geëntameerde aanpakken van deze problemen en vat thans samen de stand van zaken op het gebied van automatische analyse van gezichtssignalen (Hoofdstuk 2).

Bestaande automaten voor gezichtsuitdrukking analyse classificeren inputbeelden van het gezicht in termen van:

- of gelaatsmimiek (spieractivatie),
- of gemoedstoestand (emotie).

Bestaande automaten voor herkenning van gelaatsmimiek zijn beperkt tot coderen van kleine hoeveelheden (in het beste geval 16, wel of niet gecombineerde) AU codes ("Action Unit" (AU) codes zijn gedefinieerd in het FACS systeem dat door Ekman en Friesen in 1978 geïntroduceerd was). Tevens is geen enkele van deze systemen in staat tot coderen van de intensiteit (sterkte) van de waargenomen AU codes.

Bestaande automaten voor herkenning van gemoedstoestanden in beelden scheiden meestal zes emotionele basiscategorieën (zoals geïntroduceerd door Ekman en Friesen in 1975), ongeacht het feit dat deze theorie thans sterk bekritiseert is in de psychologische literatuur.

In tegenstelling tot bovengenoemde tekortkomingen (van bestaande automaten voor gezichtsuitdrukking analyse) is de *voornaamste bijdrage van dit proefschrift* een nieuwe methode voor de automatische analyse van gelaatsuitdrukkingen in stilstaande beelden van gezichten in termen van beide: 32 AU codes en hun sterkten (Hoofdstuk 5) en gebruiker-afhankelijke gemoedsinterpretaties en hun sterkten (Hoofdstuk 6). Met andere woorden: automatische analyse van gezichtsuitdrukkingen m.b.v. ISFER, zoals beschreven in dit proefschrift, is een stap voorwaarts in zowel de automatisering van FACS codering van stilstaande beelden van het gezicht als een stap voorwaarts in het scheppen van een universeel bruikbare gezichtsuitdrukking-analyse automaat (d.w.z. die werkt ongeacht de bestaande polemieken (in de psychologie) over emoties). Naar ons beste weten, vormt ISFER het eerste prototype van een automaat voor het:

- coderen van 32 AU codes en hun combinaties vanuit een stilstaand duaal aanzicht (zowel van voren als van opzij) van het gezicht,
- coderen van de sterkte van elke waargenomen spieractivatie (AU code) op een proefpersoon-afhankelijke manier,
- classificeren van gezichtsinformatie in gemoedsinterpretatie categorieën die door de huidige gebruiker gedefinieerd zijn,

- coderen van de uitingssterkte van elke afzonderlijke gemoedstoestand op basis van de al berekende sterkten van de waargenomen AU codes,
- voortbrengen van conclusies met bijbehorende zekerheidsfactoren (afhankelijk van zekerheid over input gezichtsinformatie).

De voorgestelde methode voor gelaatsmimiek- en gemoedsanalyse van stilstaande duale aanzichten van het gezicht, die zich automatisch voegt naar de waargenomen proefpersoon en de huidige gebruiker, brengt twee principieel verschillende *technologieën* tezamen: FACS (psychologisch en anatomisch gefundeerd) en CI (computationele intelligentie). Namelijk, ISFER realiseert synergie tussen:

- Diverse *beeldverwerkingtechnieken* zoals kunstmatige neurale netwerken, actieve contouren, en fuzzy randherkenning, om contouren van de gezichtskenmerken (wenkbrauwen, ogen, neusgaten, mond, kin, en zijaanzicht) te kunnen lokaliseren vanuit een input stilstaand duaal aanzicht van het gezicht (Hoofdstuk 4).
- Een op regels uitgerust expert systeem dat op basis van de gedetecteerde contouren van de gezichtskenmerken *redeneert met onzekerheid* over getoonde AUs en de mate waarin deze geactiveerd zijn (Hoofdstuk 5).
- Een op dynamisch geheugen uitgerust expert systeem dat op basis van de ontsloten AU codes en hun uitingssterkten *redeneert over gevallen die door het systeem toereikend opgelost waren* (case-based reasoning) om gebruikers-afhankelijke gemoedsinterpretaties en hun sterkten voort te brengen (Hoofdstuk 6).

De gepleegde *validatie* van ISFER (Hoofdstuk 7) suggereert dat, in het algemeen, de conclusies van het systeem consistent zijn met die van menselijke waarnemers van dezelfde stimulus gezichtsbeelden. De desbetreffende studies concluderen dat ISFER als adaptief onderzoeksinstrument voor experimentele gedragskunde toereikend is. Omdat de werking van de systeemsoftware nogal veel tijd vergt, is het systeem nog niet effectief toepasbaar voor mens-machine interfaces met waarnemingsvermogen. Daarvoor is real-time verwerking van de inputinformatie noodzakelijk, omdat vertragingen in de gegevensverwerking ongesynchroniseerde en daardoor minder natuurlijke interactie geeft. Desalniettemin is toepassing van ISFER haalbaar in al de bovengenoemde toepassingsgebieden, zodra de huidige in het systeem geïntegreerde beeldverwerkingtechnieken (voor lokaliseren van contouren van gezichtskenmerken) door soortgelijke technieken voor real-time beeldverwerking vervangen zijn.

Hoewel haalbaarheid van ISFER als adaptief onderzoeksinstrument voor gedragskunde aangetoond is, *kunnen meerdere aspecten van het systeem verbeterd worden*. De twee meest belangrijke zijn:

- Tot op heden is ISFER's analyse van gelaatsuitdrukkingen beperkt tot verwerking en analyse van stilstaande beelden van gezichten zonder baard, snor of bril. Tevens, is het systeem niet in staat om de volle reikwijdte van gezichtsgedrag (d.w.z. al 44 in het FACS gedefinieerde AUs) te coderen. Deze twee beperkingen kunnen, tenminste gedeeltelijk, worden behandeld door in plaats van stilstaande gezichtsbeelden, bewegende gezichtsbeelden te analyseren.
- ISFER is zodanig gebouwd dat gemoedstoestanden contextonafhankelijke autobiografische gebeurtenissen zijn die van de huidige gebruiker geleerd worden. Met andere woorden, wordt er bij het analyseren van het gezichtsgedrag van de geobserveerde proefpersoon alleen de huidig getoonde gelaatsuitdrukking in beschouwing genomen. De daarvoor getoonde gezichtsuitdrukkingen, evenals de omstandigheden waaronder de huidige gelaatsuitdrukking is getoond, worden buiten beschouwing gelaten. Desalniettemin hangt de interpretatie van gezichtsgedrag af van de typische dynamiek van gezichtsuitdrukkingen van de huidige geobserveerde proefpersoon, evenals van de situatie en de omgeving waarin de desbetreffende persoon manoeuvreert. Vanwege de complexiteit van dit probleem is ISFER's context-ongevoeligheid waarschijnlijk de grootste uitdaging voor vervolgontwikkeling van ISFER.

Curriculum Vitae

PERSONAL

Name:	Maja Pantic
Born:	April the 13th 1970 in Belgrade, Yugoslavia

EDUCATION

9/84 – 6/88 Matura in the Mathematical Gymnasium of Belgrade, Yugoslavia

9/88 – 6/92 Faculty of Mathematical Sciences at Belgrade University, Yugoslavia

9/93 – 12/96 M.Sc. (cum laude) in Technical Informatics at Delft University of Technology, the Netherlands. Thesis title: "Human Emotion Recognition Clips Utilised Expert System".

8/97 – 10/01 Ph.D. in Technical Informatics at Delft University of Technology, the Netherlands. The Ph.D. research resulted in this thesis.

WORK EXPERIENCE

7/95 – 10/95 Junior system developer at Gist-brocades in Delft, the Netherlands.

2/97 – 10/01 Research Assistant at Delft University of Technology, Faculty of Information Technology and Systems, Dept. MediaMatics, Knowledge Based Systems Group, the Netherlands.