A Few Days of A Robot's Life in the Human's World: Toward Incremental Individual Recognition

by

Lijin Aryananda

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2007

Author ....
Department of Electrical Engineering and Computer Science
February 22, 2007

Certified by...
Rodney Brooks
Professor
upervisor

Accepted by ...   .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

A Few Days of A Robot's Life in the Human's World: Toward Incremental Individual Recognition

by

Lijin Aryananda

Submitted to the Department of Electrical Engineering and Computer Science
on February 22, 2007, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis presents an integrated framework and implementation for Mertz, an expressive robotic creature for exploring the task of face recognition through natural interaction in an incremental and unsupervised fashion. The goal of this thesis is to advance toward a framework which would allow robots to incrementally "get to know" a set of familiar individuals in a natural and extendable way. This thesis is motivated by the increasingly popular goal of integrating robots in the home. In order to be effective in human-centric tasks, the robots must be able to not only recognize each family member, but also to learn about the roles of various people in the household.

In this thesis, we focus on two particular limitations of the current technology. Firstly, most of face recognition research concentrate on the supervised classification problem. Currently, one of the biggest problems in face recognition is how to generalize the system to be able to recognize new test data that vary from the training data. Thus, until this problem is solved completely, the existing supervised approaches may require multiple manual introduction and labelling sessions to include training data with enough variations. Secondly, there is typically a large gap between research prototypes and commercial products, largely due to lack of robustness and scalability to different environmental settings.

In this thesis, we propose an unsupervised approach which would allow for a more adaptive system which can incrementally update the training set with more recent data or new individuals over time. Moreover, it gives the robots a more natural *social recognition* mechanism to learn not only to recognize each person's appearance, but also to remember some relevant contextual information that the robot observed during previous interaction sessions. Therefore, this thesis focuses on integrating an unsupervised and incremental face recognition system within a physical robot which interfaces directly with humans through natural social interaction. The robot autonomously detects, tracks, and segments face images during these interactions and automatically generates a training set for its face recognition system. Moreover, in order to motivate robust solutions and address scalability issues, we chose to put the robot, Mertz, in unstructured public environments to interact with naive passersby, instead of with only the researchers within the laboratory environment.

While an unsupervised and incremental face recognition system is a crucial element toward our target goal, it is only a part of the story. A face recognition system typically receives either pre-recorded face images or a streaming video from a static camera. As illustrated an ACLU review of a commercial face recognition installation, a security application which interfaces with the latter is already very challenging. In this case, our target goal is a robot that can recognize people in a home setting. The interface between robots and humans is even more dynamic. Both the robots and the humans move around.

We present the robot implementation and its unsupervised incremental face recognition framework. We describe an algorithm for clustering local features extracted from a large set of automatically generated face data. We demonstrate the robot's capabilities and limitations in a series of experiments at a public lobby. In a final experiment, the robot interacted with a few hundred individuals in an eight day period and generated a training set of over a hundred thousand face images. We evaluate the clustering algorithm performance across a range of parameters on this automatically generated training data and also the Honda-UCSD video face database. Lastly, we present some recognition results using the self-labelled clusters.

Thesis Supervisor: Rodney Brooks
Title: Professor

Acknowledgments

I would like to express my gratitude to my advisor, Rodney Brooks, for his generous support in many different aspects. I was very fortunate to have the opportunity to work with him. His unconventional and almost rebellious thinking has really taught me to formulate and approach my research differently. He has also been a tremendous emotional support for me from the very beginning. Thank you, Rod, for waiting patiently, supporting me financially during my leave of absence, and believing that I would be able to come back to resume my study.

I would also like to thank my committee members, Cynthia Breazeal and Michael Collins. I would like to thank Cynthia for her guidance and friendship, throughout my Ph.D. career, as a lab-mate, a mentor, and eventually a committee member in my thesis. I would like to thank Michael for all of his help and advice. Thank you for taking the time to help me formulate my algorithms and improve the thesis.

The many years during the Ph.D. program are challenging, but almost nothing compared to the last stretch during the last few months. I would not have been able to survive the last few months without Eduardo, Iuliu, and Juan. Thank you, Eduardo, for helping me with my thesis draft. I would not have finished it in time without your help. Thank you, Iuliu, for going through the algorithms with me again and again, and for helping me during my last push. Thank you, Juan, for proof-reading my thesis, helping me with my presentation slides, and even ironing my shirt before the defense.

Many people have kindly helped me during the project and thesis writing. I would like to thank Ann for her warm generosity and inspiring insights; Jeff Weber for his beautiful design of Mertz; Carrick, for helping me with so many things with the robot; Philipp, for programming the complicated pie-chart plotter which really saved my result presentation; Paulina, for proof-reading my thesis and going over the clustering algorithm; Lilla, for proof-reading my thesis and always ready to help; James, for spending so many hours helping me with my thesis and slides; Mac, for helping me with the matlab plots; Marty, for discussing and giving feedback on my

results; Joaquin and Alex, for helping me through some difficult nights; Becky, for always checking on me with warm smiles and encouragement; Iris, Julian, and Justin, for helping me with various Mertz projects; Louis-Philippe and Bernd Heisele for going over my clustering results.

While the recent memory of thesis writing is more salient, I would also like to thank all of my lab-mates who have shaped my experience in the lab during the past seven years: Jessica, Aaron, Una-May, Varun, Bryan, Myung-hee, Charlie, Paul, Lorenzo, Giorgio, Artur, Howej, Scaz, Martin Martin, Maddog, and Alana. A special thank you for Kathleen, my lab-mate, my best friend, my favorite anthropologist.

Most importantly, I would like to thank my parents. Without their unconditional support, I would not have been able to achieve my goals. And without their progressive vision, I would not have aspired to my goals in the first place.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> "Learning is experiencing. Everything else is just information" (Albert Einstein).

This thesis presents an integrated framework and implementation for Mertz, an expressive robotic creature for exploring the task of face recognition through natural interaction in an incremental and unsupervised fashion. The goal of this thesis is to advance toward a framework which would allow robots to incrementally "get to know" a set of familiar individuals in a natural and extendable way.

This thesis is motivated by the increasingly popular goal of integrating robots in the home. We have now seen the Roomba take over the vacuum cleaning task in many homes. As the robotic technology further advances, we would expect to see more complex and general robotic assistants for various tasks, such as elder care, domestic chores, etc. In order to be effective in human-centric tasks, the robots must be able to not only recognize each family member, but also to learn about the roles of various people in the household – who is the elderly person, who is the young child, who is the part-time nurse caregiver.

In this thesis, we focus on two particular limitations of the current technology. Firstly, most of face recognition research concentrate on the supervised classification problem: given a set of manually labelled training data, find the correct person label for a new set of test data. The supervised approach is not ideal for two reasons.

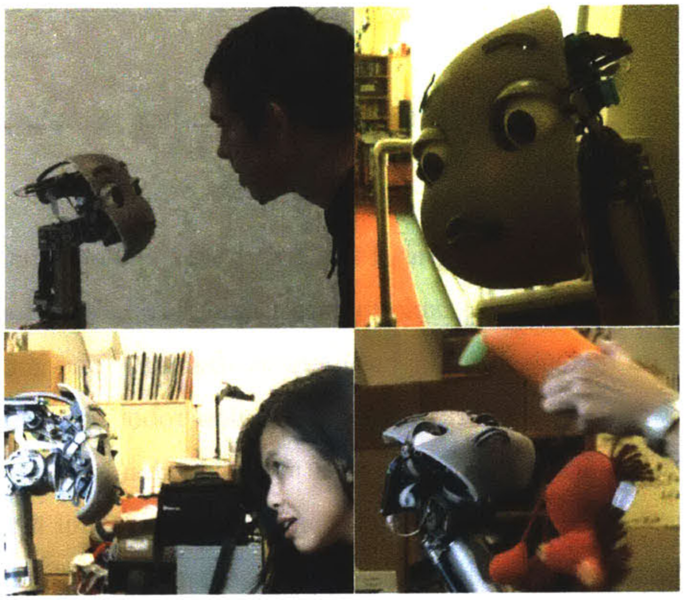


Figure 1-1: Mertz is an expressive head robot designed to explore incremental individual recognition through natural interaction. The robot's task is to learn to recognize a set of familiar individuals in an incremental and unsupervised fashion.

The first reason is a practical one. Currently, one of the biggest problems in face recognition is how to generalize the system to be able to recognize new test data that look different from the training data, due to variations in pose, facial expressions, lighting, etc. Thus, until this problem is solved completely, the existing supervised approaches may require multiple manual introduction and labelling sessions to include training data with enough variations. The second reason involves the human factor and social interface. In a study investigating long-term human-robot social interaction, the authors concluded that in order to establish long-term relationships, the robot should be able to not only identify but also "get to know" people who the robot frequently encounters [36].

The second limitation of current technology is that there is typically a large gap between research prototypes and commercial products. Despite tremendous research progress in humanoid robotics over the past decade, deployment of more complex and general robotic assistants into the home is not simply a matter of time. Lack of reliability and robustness are two of the most important obstacles in this path. Similarly, despite many advances in face recognition technology, the American Civil Liberty Union's review of deployment of a commercial face recognition surveillance system at the Palm Beach airport yielded unsatisfactory results [92].

1.1 Thesis Approach

Our target goal is a robot which can incrementally learn to recognize a set of familiar individuals in a home setting. As shown in figure 1-2, the system starts with an empty database. As the robot encounters each person, it has to decide on the person's identity. If she is a new person (i.e. does not exist in the database), the robot will generate a new class in the database, into which the robot will insert her face and voice data. Upon collecting enough data in a class, the robot subsequently trains a recognition system using its automatically generated database. After a number of encounters, the robot should be able to recognize this new person and update her data in the database appropriately.

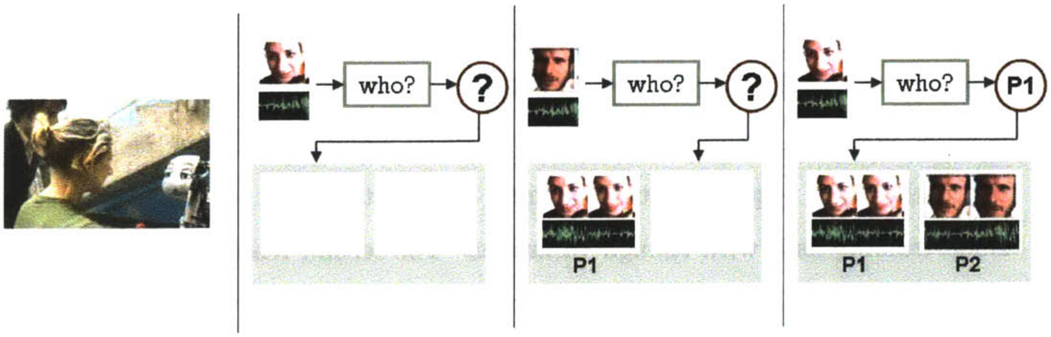


Figure 1-2: A simplified diagram of the unsupervised incremental face recognition scheme. The robot first starts with an empty database. As the robot encounters each person, it has to decide on the person's identity. If she is a new person, the robot will generate a new class in the database, into which the robot will insert her face and voice data. After a number of encounters, the robot should be able to recognize this new person and update her data in the database appropriately.

In this thesis, we propose an unsupervised approach which would allow for a more adaptive system which can incrementally update the training set with more recent data ore new individuals over time. Moreover, it gives the robots a more natural *social recognition* mechanism to learn not only to recognize each person's appearance, but also to remember some relevant contextual information that the robot observed during previous interaction sessions [58].

While an unsupervised and incremental face recognition system is a crucial element toward our target goal, it is only a part of the story. A face recognition system typically receives either pre-recorded face images or a streaming video from a static camera. As illustrated by the ACLU review, a security application which interfaces with the latter is already very challenging. In this case, our target goal is a robot that can recognize people in a home setting. The interface between robots and humans is even more dynamic. Both the robots and the humans move around.

Therefore, this thesis focuses on integrating an unsupervised and incremental face recognition system within a physical robot which interfaces directly with humans through natural social interaction. Moreover, in order to motivate robust solutions and address scalability issues, we chose to put the robot, Mertz, in unstructured public

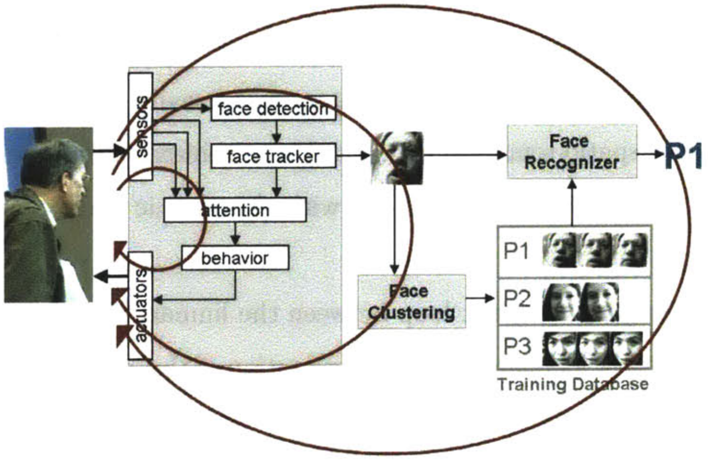


Figure 1-3: The fully integrated system from raw input to the incremental face recognition system, which we implemented in this thesis. Superimposed are three feedback loops which allow for a set of opportunites described in the text.

environments to interact with naive passersby, instead of with only the researchers within the laboratory environment. Figure 1-3 shows the fully integrated system that we implemented in this thesis, connecting raw input in real human environments to the incremental face recognition system.

1.2 Integration and Opportunities

A fully integrated system from raw input in the real world to the incremental recognition system generates a number of opportunities. As shown in figure 1-3, there are three feedback loops which allow for the following opportunities.

First, there is a small loop between the human and the robot. This corresponds to human-robot interaction. Through many hours of embodied interaction, the robot can generate a large amount of natural human-centric perceptual data. In a four day long experiment at the very early project stage, the robot collected over 90,000 face images from over 600 individuals with a wide range of natural poses and facial expressions. Moreover, the robot can take advantage of various contextual information that is freely available through its embodied experience. Mertz utilizes and learns to associate concurrent multi-modal sensory data to improve various perceptual mechanisms. Mertz's attention system also heavily relies on both multi-modal integration and spatio-temporal context. These contextual mechanisms have turned out to be particularly useful in dealing with the chaotic and noisy human's natural environment.

Second, there is a feedback loop between the human, the robot, and the learning data. This feedback loop allows for an exploration of the experiential learning concept which has been proposed by many, in various research disciplines [28, 99, 49]. As the robot autonomously decides what and when to learn, the robot can organize and influence these self-generated data in the most convenient way for its own learning task. For example, in order to allow for higher resolution face images, the robot verbally requests for people to stand closer to the robot when they are too far away.

Third, there is a big feedback loop between the human, the robot, and the incremental recognition system. This feedback loop allows for an opportunity for the robot to adapt its behavior based on the recognition output, i.e. as the robot gets to know some familiar individuals. In animal behavior research, this is widely known as *social recognition*, i.e. a process whereby animals become familiar with conspecifics and later treat them based on the nature of those previous interactions [58]. *Social recognition* capabilities have been observed in bottlenose dolphins, lambs, hens, and mantis shrimps [83, 78, 27, 24]. We do not explore this feedback loop in this thesis, however, this thesis contributes by advancing toward social recognition capabilities as a prerequisite for long-term human-robot social interaction [36].

1.3 Integration, Challenges, and Robustness

In addition to the above opportunities, a fully integrated system from raw input in the real world to the incremental recognition system, as shown in figure 1-3 also present many challenges. Error propagation through the many subsystems is one of the most unexplored challenges, since most research projects currently focus on building isolated systems for face detection, tracking, clustering, recognition, etc. Moreover, most current systems interface with data obtained from controlled experiments, by either taking pictures of subjects or asking them to move around in front of a static camera. Unless one chains together each of the subsystems in a fully integrated system and interfaces the system with real human environment, one will not see the extent of the challenges posed by the error propagation.

Inspired by biological systems which are incredibly robust despite many redundant and non-optimized components, our approach is not to optimize each subsystem. Instead, we focus on achieving a robust integrated framework where failures in a module are somehow compensated by other modules further down the line.

Moreover, our deliberate choice of an uncontrolled and challenging setup was driven by the assumption that in a dynamic and noisy environment, errors are inevitable. It would be unrealistic to assume that 100% accuracy is possible in every part of the system. We hypothesize that these errors and imperfections are in fact useful as they would motivate the rest of the system to compensate robustly.

More generally, we propose that setting a higher benchmark for robustness and generality of operating condition is likely to motivate more scalable solutions. In the conventional setup, where task performance has the highest priority, it is common to employ shortcuts so as to allow for initial progress. It is quite typical for robots to require a specific set-up, such as a particular location, set of objects, background color, or lighting condition. Such simplifications, while far from reducing the environment to a *blocks world*, naturally raise some scalability concerns. A deeper concern is that these shortcuts might actually be hampering potential progress in the longer term.

1.4 The Task Breakdown

In order to illustrate the project goal and approach more concretely, we will enumerate the robot's set of tasks. While operating in the midst of hundreds of passersby, the robot has to perform the following tasks automatically:

1. operate continuously for up to 10 hours each day;

2. attract passersby to approach the robot and engage them in spontaneous social interaction, e.g., by visual tracking and simple verbal exchanges;
3. regulate the interaction in order to generate opportunities to collect data from as many people as possible;
4. detect, segment, store, and process face images and voice samples during interaction;
5. use tracking and spatio-temporal assumptions to obtain a sequence of face images and voice samples of each individual as a starting point toward unsupervised recognition;
6. perform unsupervised clustering on these collected face sequences and construct a self-generated training database, where each class consists of face images and voice samples of each individual;
7. train a face recognition system using the self-generated training database to recognize a set of individuals the robot has previously encountered.

1.5 Thesis Scope and Criteria

Throughout this project, our design and implementation steps have been guided by the following set of criteria and scope.

Full automation and integration The robot has to be fully automatic and integrated. The robot has to autonomously interact with, segment, collect relevant data from passersby, and use it to construct a training database without any human intervention.

Unstructured environment It has to operate in real time and in different public environments, without the use of any markers to simplify the environment.

No manual data filtering or processing We do not allow any manual intervention to process and filter the data which the robot collected during the experiments. The robot automatically stores these data to be used directly by the recognition system.

Timing requirement The robot's sensors and actuators have to perform in real time. The face clustering system does not have to operate in real time.

Natural human-robot interaction The robot has to be able to interact with multiple people concurrently in the most natural way possible, with minimal constraints and instructions. In some cases, we have had to compromise performance in other parts of the system in order to enforce this requirement. For example, we have to use a desktop microphone, instead of a headset, to allow multiple people to speak to the robot at the same time. As a result, the robot's speech recognition performance is compromised. Without reliable speech understanding, the robot's verbal behavior is thus limited to simple word mimicking and some predefined sentences.

Trade-off between robustness and complexity We expect that trade-off between robustness and complexity would be inevitable. Thus, we have bypassed some optimization steps in some of the robot's subtasks. For example, the robot engages people by simply visually tracking their faces, instead of explicitly detecting their eyes for precise eye contact.

The recognition task The robot's main task is to learn to recognize familiar individuals, i.e., those who frequently interact with the robot. The robot does not have to learn to recognize every single person the robot ever encountered. Moreover, we will not evaluate each of the robot's subsystem and look for the most optimized performance. We expect that the system will make mistakes, but these mistakes must be somehow compensated by another part of the system. Instead of aiming to achieve the highest classification accuracy for each test data, we would like to explore alternative compensation methods which make sense given our setup. For example, in the future, we are interested in exploring an active learning scheme where the robot makes an inquiry to a person to check if its recognition hypothesis is correct and somehow integrate the answers into its learning system.

1.6 Thesis Contribution

This thesis makes the following contributions:

- We implemented an integrated end-to-end incremental and fully unsupervised face recognition framework within a robotic platform embedded in real human environment. This integrated systems provides automation for all stages, from data collection, segmentation, labelling, training, and recognition.

- We developed a face sequence clustering algorithm that is robust to a high level of noise in the robot's collected data, generated by inaccuracies in face detection, tracking, and segmentation in the dynamic and unstructured human environments.

- We implemented a robust robotic platform capable of generating a large amount of natural human-centric data through spontaneous interactions in public environments.

- We present an adaptive multi-modal attention system coupled with spatio-temporal learning mechanism to allow the robot to cope with the dynamic and noise in real human environment. These coupled mechanisms allow the robot to learn multi-modal co-occurence and spatio-temporal patterns based on its past sensory experience. The design of the attention system also provides a natural integration between bottom-up and top-down control, allowing simultaneous interaction and learning.

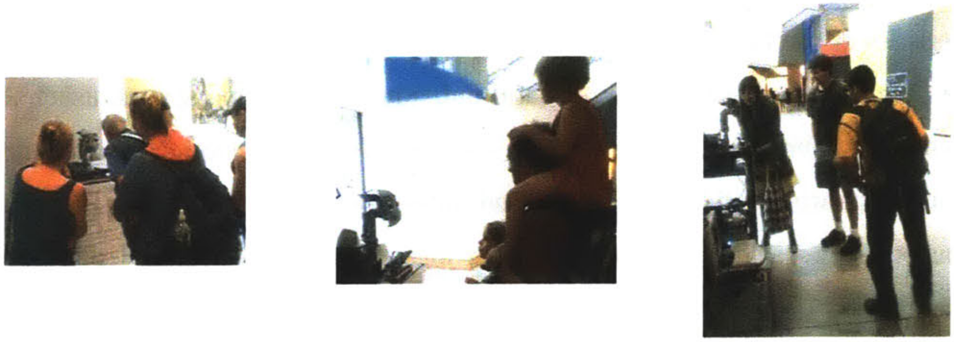


Figure 1-4: Some snapshots of Mertz in action, interacting with many passersby in the lobby of the MIT Stata Center. Mertz is typically approached by multiple people at once. It is a quite hectic environment for a robot.

1.7 Demonstration and Evaluation

We demonstrate the robot's capabilities and limitations in a series of experiments. We initially summarized a set of multi-day experiments in public spaces, conducted at the early stage of the project and interleaved with the robot development process. Lessons from these early experiments have been very valuable for our iterations of design and development throughout the project.

At the end of the project, we conducted a final experiment to evaluate the robot's overall performance from the perspective of its main task, incremental individual recognition. This experiment was held for 8 days, 2-7 hours each day, at a public lobby. Figure 1-4 shows some snapshots of the robot interacting with many passersby in the lobby of the MIT Stata Center. We describe a set of quantitative and qualitative results from each of the robot's relevant subtasks.

We first assess the robot's capabilities in finding and engaging passersby in spontaneous interaction. This involves the robot's perceptual, attention, and spatio-temporal learning systems. Toward the final goal, the robot first has to collect as many faces and voice images as possible from each person. The longer and more natural these interactions are, the more and better these data would be for further recognition. We then evaluate the face and voice data that the robot automatically collected during the experiment. We analyze the accuracy and other relevant characteristics of the face sequences collected from each person.

We then evaluate the incremental individual recognition system using the automatically generated training data. We analyze the face clustering performance across different variants of the algorithm and data-dependent parameters. For comparison purposes, we also apply the clustering algorithm on the Honda-UCSD video face database [51, 50]. Lastly, we analyze both the incremental recognition performance and the accuracy of the self-generated clusters in an evaluation of the integrated incremental and unsupervised face recognition sytem.

1.8 Thesis Outline

We first begin by discussing some background information and related work involving the issues of face recognition, social robotics, and robustness issues in robotics in chapter 2.

In chapter 3, we discuss the robot design and building process. We focus on two issues that received the most amount of consideration during the design process: robustness and social interface. We also present a series of earlier experiments conducted at different stages of the project and describe a number of valuable lessons that heavily influenced the final implementation of the robot.

In chapter 4, we provide the implementation of the robot's perceptual, attention, and behavior systems. The robot has to organize these subsystems not only to solicit spontaneous interaction, but also to regulate these interactions to generate learning opportunities and collect as much face and voice data as possible from various individuals. We then evaluate these subsystems, with respect to the target goal of collecting face and voice data from each individual through spontaneous interaction.

In chapter 5, we describe how these collected data are automatically processed by the robot's individual recognition system to incrementally learn to recognize a set of familiar individuals. We present the implementation of the unsupervised face clustering and how this solution is integrated into an incremental and fully unsupervised face recognition system. We evaluate this integrated system using the robot's collected face sequences to analyze both the incremental recognition performance and the accuracy of the self-generated clusters.

In the last chapter, we provide a conclusion and describe some possible future directions.

Chapter 2

Background and Related Work

2.1 Learning from Experience

The main inspiration of this thesis was derived from the concept of *active or* experiential learning. The emphasized role of experience in the learning process has been proposed in many different research areas, such as child development psychology, educational theories, and developmental robotics.

Piaget proposed that cognitive development in children is contingent on four factors: biological maturation, experience with the physical environment, experience with the social environment, and equilibration [77, 85]. "Experience is always necessary for intellectual development...the subject must be active...." [49]. Vygotsky developed a sociocultural theory of cognitive development, emphasizing the role the socio-cultural in the human's cognitive development [99].

Learning by "doing" is a popular theme in modern educational theories since John Dewey's argument that children must be engaged in an active experience for learning. [28]. The principle of sensory-motor coordination was inspired by John Dewey, who, as early as 1896, had pointed out the importance of sensory-motor coordination for perception. This principle implies that through coordinated interaction with the environment, an agent can structure its own sensory input. In this way, correlated sensory stimulation can be generated in different sensory channels – an important prerequisite for perceptual learning and concept development. The importance of direct experience of sensory input and actuation in the world through physical embodiment is the cornerstone of the embodied Artificial Intelligence paradigm [16, 74]. In this thesis, by learning while experiencing the world, the robot gains the opportunity to not only generate sensory input through its behavior, but also actively structure these sensory inputs to accommodate its learning task. The use of social behavior has been shown to be effective in regulating interaction and accommodating the robot's visual processing [15].

The notion that the human developmental process should play a significant role in the pursuit of Artificial Intelligence has been around for a long time. The associated idea of a child machine learning from human teachers dates back at least to Alan Turing's seminal paper "Computing Machinery and Intelligence" [95]. The interpretation and implementation of this developmental approach have varied from having human operators enter common-sense knowledge into a large database system [52] to robots that learn by imitating human teachers [84].

Developmental robotics is a very active research area that has emerged based on these developmental concepts. Many developmental approaches have been proposed and implemented in various robots [56]. Most relevant to our work are SAIL and Dav at Michigan State University, humanoid robot platforms for exploring autonomous life-long learning and development through interaction with the physical environment and human teachers [102].

2.2 Human-Robot Interaction

The long-term objective of this thesis is to advance toward incremental individual recognition as a prerequisite for long-term human-robot social interaction. Social robotics is a growing research area based on the notion that human-robot social interaction is a necessary step toward integrating robots into human's everyday lifes [3] and for some, also a crucial element in the development of robot intelligence [26, 11, 17].

[32] presents a survey of different approaches in developing socially interactive

robots. These systems vary in their goals and implementations. The following robots are mainly focused on one-on-one and shorter-term interaction in controlled environments. Kismet at MIT is an expressive active vision head robot, developed to engage people in natural and expressive face-to-face interaction [11]. The research motivation is to bootstrap from social competences to allow people to provide scaffolding to teach the robot and facilitate learning.

WE-4R at Waseda University is an emotionally expressive humanoid robot, developed to explore new mechanisms and functions for natural communication between humanoid robot and humans [62]. The robot has also been used to explore emotion-based conditional learning from the robot's experience [61]. Leonardo at MIT is an embodied humanoid robot designed to utilize social interaction as a natural interface to participate in human-robot collaboration [13]. Infanoid at National Institute of Information and Communications Technology (NICT) is an expressive humanoid robot, developed to investigate joint attention as a crucial element in the path of children's social development [47].

There have also been a number of approaches in developing social robotic platforms which can operate for longer time scales in uncontrolled environments outside the laboratory. The Nursebot at Carnegie Mellon University is a mobile platform designed and developed toward achieving a personal robotic assistant for the elderly [63]. In a two day-long experiment, the Nursebot performed various tasks to guide elderly people in an assisted living facility. Similar to our findings in dealing with uncontrolled environments, the Nursebot's speech recognition system initially encountered difficulties and had to be re-adjusted during the course of the experiment. Grace at CMU is an interactive mobile platform which has participated in the AAAI robot challenge of attending, registrating, and presenting at a conference [91].

Robovie at ATR, an interactive humanoid robot platform, has been used to study long-term interaction with children for two weeks in their classrooms [43]. Keepon at NICT, is a creature-like robot designed to perform emotional and attention exchange with human interactants, especially children [48]. Keepon was used in a year and a half long study at a day-care center to observe interaction with autistic children.

Robox, an interactive mobile robotic platform, was installed for three months at the Swiss National Exhibition Expo 2002 [89]. RUBI and QRIO at the University of California San Diego are two humanoid robots which were embedded at the Early Childhoold Education Center as part of a human-robot interaction study for at least one year on a daily basis [67]. Robovie-M, a small interactive humanoid robot, was tested at in a two-day human-robot interaction experiment at the Osaka Science Museum [88].

Most relevant to our project focus is Valerie at Carnegie Melon University, a mobile robotic platform designed to investigate long-term human-robot social interaction [36]. Valerie was installed for nine months at the entranceway to a university building. It consists of a commercial mobile platform, an expressive animated face displayed on an LCD screen mounted on a pan-tilt unit, and a speech synthesizer. It uses a SICK scanning laser range finder to detect and track people. People can interact with Valerie by either speech or keyboard input. Similar to our case, the authors report that a headset micropone is not an option and therefore the robot's speech recognition is limited especially given the noisy environment. Valerie recognizes individuals by using a magnetic card-stripe reader. People can swipe any magnetic ID cards in order to uniquely identify themselves. One of Valerie's primary interaction modes is storytelling through 2-3 minute long monologues about its own life stories.

During these nine months, people have interacted with Valerie over 16,000 times, counted by keyboard input of at lease one line of text. An average of over 88 people interacted with Valerie each day. Typical interaction sessions are just under 30 seconds. Out of 753 people who have swiped an ID card to identify themselves, only 233 have done it again during subsequent visits. Valerie encounters 7 repeat visitors on average each day. These repeat visitors tend to interact with the robot for longer periods, typicaly for a minute or longer. The authors suggest that in order to study true long-term interactions with Valerie, the robot needs to be able to identify repeat visitors automatically. Moreover, Valerie should not only identify but also get to know people who frequent the booth.

We have the common goal of extending human-robot social interaction. Moreover, Valerie's setup in the midst of passersby and public environments is similar to ours. However, Valerie has been installed and tested for a much longer period. In terms of user interface and perceptual capabilities, Mertz differs from Valerie in a number of ways. Mertz can only interact with people through visual and verbal exchange. Thus, it can rely on only noisy camera and microphone input in its interaction with people. Mertz is a mechanical robotic head and is more expressive in terms head postures. Valerie's flat-screen face was reported to have difficulties in expressing gaze direction. However, Mertz only has four degrees of freedom allocated to its facial expression, allowing a much smaller range than an animated face.

2.3 Extending Robustness and Generality

The thesis goal of extending the duration and spatial range of operation is important in that it addresses a particular limitation of current humanoid robotics research. Despite tremendous research progress in humanoid robotics over the past decade, it still is challenging to develop a robot capable of performing in a robust and reliable manner.

As accurately described by Bischoff and Graefe [8], robot reliability has not received adequate attention in most humanoid robotics research projects. One possible cause could be a false belief that when the time comes for robots to be expedited, someone else will eventually address this limitation. Moreover, robustness to a wide range of environments is crucial as the home environment is largely unstructured and each one varies from another. This flexibility is still a major challenge in robotics. The current trend in the field is to equip the robot to achieve a very specific and difficult task. The end goal is typically to demonstrate the robot performing its task for a few minutes. Humanoid robots generally have a limited average running period and are mostly demonstrated through short video clips, which provide a relatively narrow perspective on the robot's capabilities and limitations. This particular setup tends to both require and generate very sophisticated but specialized solutions. Scalability issues to other environments, other locations, and other users mostly have been put on hold for now.

Bischoff and Graefe [8] present HERMES, an autonomous humanoid service robot, designed specifically for dependability. It has been tested outside the laboratory environment for long periods of time, including an over six-month long exhibition in a museum. Although our project is exploring a different research direction, we fully concur with the underlying theme of increasing robot robustness and reliability.

Reliability is also a relevant topic in other museum tour-guide robots [22, 94, 71]. Deployment to a public venue and the need to operate on a daily basis naturally place reliability demands on these robots. Although Mertz is quite different in form and function, we are exploiting a similar demand to have the robot perform on a daily basis and interact with many people in a public venue.

2.4 Face Recognition

Research in person identification technology has recently received significant attention, due to the wide range of biometric, information security, law enforcement applications, and Human Computer Interaction (HCI). Face recognition is the most frequently explored modality and has been implemented using various approaches [105]. [68] attempted to combine face and body recognition. Speaker recognition has also been widely investigated [35]. The use of multiple modalities have been observed by ???? [21, 55, 46, 25].

There are two main branches in face recognition research: image-based and video-based recognition. Image-based recognition typically involves high-resolution face images, while video-based recognition deals with lower resolution camera input. Both approaches have been explored using many different methods ranging from Principle Component Analysis, Hidden Markov Model, to 3-dimensional morphable models [96, 53, 9].

Our work falls on the latter category. While video-based approach has its set of challenges, given the more dynamic input, it also has a number of advantages. Instead of relying on single image frame for training or recognition, we can start one step ahead by tracking and utilizing spatio-temporal context. Video-based supervised face recognition is increasingly more prevalent and has been explored using various approaches [53, 106, 51, 38].

Our implementation of the robot's face recognition system relies on the Scale Invariant Feature Transform (SIFT) method. SIFT is a feature descriptor algorithm, developed by David Lowe [54]. SIFT has been shown to provide robust matching despite scale changes, rotation, noise, and illumination variations. There have a been a number of recent supervised face recognition work which also rely on the use of SIFT features due to its powerful invariance capacity [6, 60, 100]. However, the processing of these SIFT features differ significantly among these approaches, including ours.

Most of face recognition research focus on the supervised classification problem, i.e. given a set of manually labelled training data, find the correct person label for a new set of test data. A number of researchers have been working on extending this technology to allow for unsupervised training, motivated by a range of different purposes and applications. Most of these systems, including ours, share the common feature of relying on video-based approaches. Thus, the task is to cluster face sequences obtained by tracking instead of single face images. We will now discuss the different goals and approaches of these related research.

Eickeler et al proposed an image and video indexing approach that combines face detection and recognition methods [30]. Using a neural network based face detector, extracted faces are grouped into clusters by a combination of a face recognition method using pseudo two-dimensional Hidden Markov Models and a k-means clustering algorithm. The number of clusters are specified manually. Experiments on a TV broadcast news sequence demonstrated that the system is able to discriminate between three different newscasters and an interviewed person. In contrast to this work, the number of clusters i.e. the number of individuals, is unknown in our case.

Weng et al presents an incremental learning method for video-based face recognition [103]. The system receives a video camera output as input as well as a simulated auditory sensor. Training and testing sessions are interleaved, as manually determined by the trainer. Each individual is labeled by manually entering the person's name and gender during the training session. Both cameras and subjects are static. A recognition accuracy of 95.1% has been achieved on 143 people. The issue of direct coupling between the face recognition system and sensory input is very relevant to our work, due to the requirement of an embodied setting.

Belongie et al presents a video-based face tracking method specifically designed to allow autonomous acquisition of training data for face recognition [40]. The system was tested using 500-frame webcam videos of six subjects in indoor environments with significant background clutter and distracting passersby. Subjects were asked to move to different locations to induce appearance variations. The system extracted between zero to 12 face samples for each subject and never extracted a non-face area. The described setup with background and distractions from other people is similar to ours. However, our system differs in that it allows tracking of multiple people simultaneously.

[5] presents an unsupervised clustering of face images obtained from captioned news images and a set of names which were automatically extracted from the associated caption. The system clusters both the face images together with the associated names using a modified k-means clustering process. The face part of the clustering system uses projected RGB pixels of rectified face images, which were further processed for dimensionality reduction and linear discriminant analysis. After various filtering, clustering results were reported to produce an error rate of 6.6% using 2,417 images and 26% using 19,355 images.

Raytchev and Murase propose an unsupervised multi-view video-based face recognition system, using two novel pairwise clustering algorithms and standard image-based distance measures [81]. The algorithm uses grey-scale pixels of size-normalized face images as features. The system was tested using about 600 frontal and multi-view face image sequences collected from 33 subjects using a static camera over a period of several months. The length of these video sequences range from 30-300 frames. The subjects walked in front of the camera with different speeds and occasional stops. Only one subject is present in each video sequence. Sample images show large variations in scale and orientations, but not in facial expressions. For evaluation purposes, the authors defined the following performance metric, $p = (1-(E_{AB}+E_O)/N)*100\%$, where E_{AB} is the number of sequences mistakenly grouped into cluster A although they should be in B and E_O is the number of samples gathered in clusters in which no single category occupies more than 50% of the nodes. Using this metric, the best performance rate was 91.1 % on the most difficult data set.

[66] Mou presents an unsupervised video-based incremental face recognition system. The system is fully automatic, including a face detector, tracker, and unsupervised recognizer. The recognition system uses feature encoding from FaceVACS, a commercial face recognition product. The system was first tested with a few hours of TV news video input and automatically learned 19 people. Only a qualitative description was reported that the system had no problem to recognize all the news reporter when they showed up again. The system was also tested with 20 subjects who were recorded in a span of two years. Other than the fact that one person was falsely recognized as two different people, no detailed quantitative results were provided.

In this thesis, we aim to solve the same unsupervised face recognition problem as in the last two papers. Our approach differs in that our system is integrated within an embodied interactive robot that autonomously collected training data through active interaction with the environment. Moreover, we deal with naive passersby in a more dynamic public environment, instead of manual recording of subjects or TV news video input.

Chapter 3

Robot Design and Some Early Lessons

In this chapter, we will discuss the set of criteria and strategies that we employed during the robot design process. As listed in section 1.4, the first task toward achieving the thesis goal is to build a robot that can operate for many hours and engage in spontaneous interaction with passersby in different public spaces. This translates into two major design prerequisites.

Firstly, the robot design must satisfy an adequate level of robustness to allow for long-term continuous operation and handle the complexity and noise in human's environment.

Secondly, given that natural interactive behavior from humans is a prerequisite for Mertz's learning process, the robot must be equipped with basic social competencies to solicit naive passersby to engage in a natural and spontaneous interaction with the robot. As listed in the thesis performance criteria, Mertz has to be able to interact with people in the most natural way possible, with minimum constraints. Since the robot is placed in public spaces, this means that the robot must be able to interact with multiple people simultaneously.

3.1 Design Criteria and Strategy

3.1.1 Increasing Robustness

The robot building process is a struggle of dealing with a high level of complexity with limited resources and a large set of constraints. In order to allow many hours of continuous operation, the robot must be immune against various incidents. Failures may occur at any point in the intricate dependency and interaction among the mechanical, electrical, and software systems. Each degree of freedom of the robot may fail because of inaccurate position/torque feedback, loose cables, obstruction in the joint's path, processor failures, stalled motors, error in initial calibration, power cycle, and various other sources.

Even if all predictable problems are taken into account during design time, emergent failures often arise due to unexpected features in the environment. Perceptual sensors particularly suffer from this problem. The environment is a rich source of information for the robot to learn from, but is also plagued with a vast amount of noise and uncertainty. Naturally, the more general the robot's operating condition needs to be, the more challenging it is for the robot to perform its task.

During the design process, maximum efforts must be put to minimize the risk of failures and to attain an appropriate balance in complexity and robustness. Moreover, modularity in subsystems and maximizing autonomy at each control level are crucial in order to minimize chaining of failures, leading to catastrophic accidents. We spent a lot of time and efforts in stabilizing the low-level control modules. All software programs must be developed to run for many hours and thus free of occasional bugs and memory leaks.

In addition to fault related issues, the robot must be easily transported and set up at different locations. The start-up procedure must be streamlined such that the robot can be turned on and off quickly and with minimum effort. In our past experience, such a trivial issue had generated enough hesitation in researchers to turn on the robot frequently. Lastly, we conducted a series of long exhaustive testing processes in different environmental conditions and carried out multiple design iterations to explore the full range of possible failure modes and appropriate solutions.

3.1.2 Designing A Social Interface

As social creatures, humans have the natural propensity to anticipate and generate social behaviors while interacting with others. In addition, research has indicated that humans also tend to anthropomorphize non-living objects, such as computers, and that even minimal cues evoke social responses [69]. Taking advantage of this favorable characteristic, Mertz must have the ability to produce and comprehend a set of social cues that are most overt and natural to us. Results in a human-robot interaction experiment suggest that the ability to convey expression and indicate attention are the minimal requirements for effective social interaction between humans and robots [20]. Thus, we have equipped Mertz with the capability to generate and perceive a set of social behaviors, which we will describe in more details below.

3.2 The Robotic Platform

Mertz is an active-vision head robot with thirteen degrees of freedom (DOF), using nine brushed DC motors for the head and four RC servo motors for the face elements (see figure 3-1). As a tradeoff between complexity and robustness, we attempted to minimize the total number of DOFs while maintaining sufficient expressivity.

The eight primary DOFs are dedicated to emulate each category of human eye movements, i.e. saccades, smooth pursuit, vergence, vesticular-ocular reflex, and opto-kinetic response [17]. The head pans and uses a cable-drive differential to tilt and roll. The eyes pan individually, but tilt together. The neck also tilts and rolls using two Series Elastic Actuators [80] configured to form a differential joint.

The expressive element of Mertz's design is essential for the robot's social interaction interface. The eyelids are coupled as one axis. Each of the two eyebrows and lips is independently actuated.

Mertz perceives visual input from two color digital cameras (Point Grey Dragonfly) with FireWire interfaces, chosen for their superior image quality. They produce

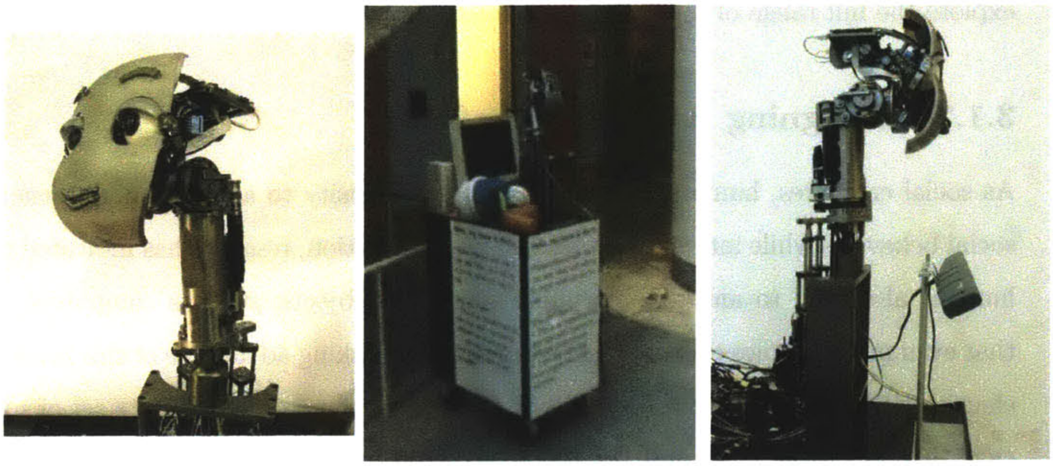


Figure 3-1: Multiple views of Mertz, an active-vision humanoid head robot with 13 degrees of freedom (DOF). The head and neck have 9 DOF. The face has actuated eyebrows and lips for generating facial expressions. The robot perceives visual input using two digital cameras and receives audio input using a desk voice-array microphone, placed approximately 7 inches in front of the robot. The robot is mounted on a portable wheeled platform that is easily moved around and can be turned on anywhere by simply plugging into a power outlet.

640 x 480, 24 bit color images at the rate of 30 frames per second. The robot receives proprioceptive feedback from both potentiometers and encoders mounted on each axis. We also equipped the robot with an Acoustic Magic desk microphone, instead of a head-set microphone, in order to allow for unconstrained interaction with multiple people simultaneously. The robot's vocalization is produced by the DECtalk phoneme-based speech synthesizer using regular PC speakers.

Lastly, the robot is mounted on a portable wheeled platform that is easily moved around and can be booted up anywhere by simply plugging into a power outlet.

3.3 Designing for Robustness

3.3.1 Mechanical Design

Mertz was mechanically designed with the goal of having the robot be able to run for many hours at a time without supervision. Drawing from lessons from previous robots, we incorporated various failure prevention and maintenance strategies, as

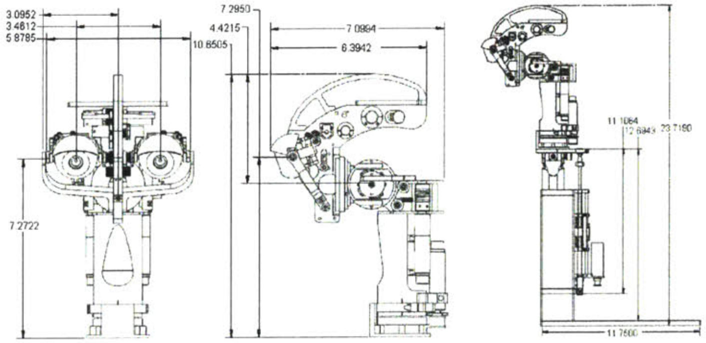


Figure 3-2: One of the mechanical design goals is to minimize the robot's size and weight. The smaller and lighter the robot is, the less torque is required from the motors to achieve the same velocity. The overall head dimension is 10.65 x 6.2 x 7.1 inches and weighs 4.25 lbs. The Series Elastic Actuators extend 11.1 inches below the neck and each one weighs 1 lb.

described below. The mechanical design of the robot was produced in collaboration with Jeff Weber.

Compact design to minimize total size, weight, and power A high-priority constraint was placed during the early phase of the design process to minimize the robot's size and weight. A smaller and lighter robot requires less torque from the motors to reach the same velocity. Also, the robot is less prone to overloading causing overheating and premature wear of the motors. The overall head size is 10.65 x 6.2 x 7.1 inches (see figure 3-2) and weighs 4.25 lbs. The Series Elastic Actuators, which bear the weight of the head at the lower neck universal joint, extend below it 11.1 inches. Mertz's compact design is kept light by incorporating nominal light alloy parts, which retaining stiffness and durability for their small size. Titanium, as an alternative to aluminum, was also used for some parts in order to minimize weight without sacrificing strength.

Force sensing for compliancy Two linear Series Elastic Actuators (SEA) [80] are used for the differential neck joint, a crucial axis responsible for supporting the

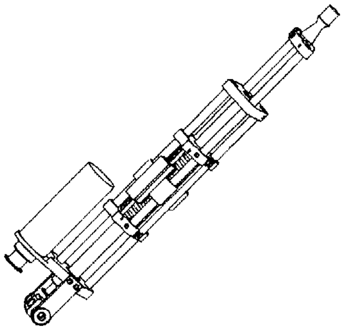


Figure 3-3: The Series Elastic Actuator (SEA) is equipped with a linear spring that is placed in series between the motor and load. A pair of SEAs are used to construct the differential neck joint, allowing easy implementation of force control. The neck axis is the biggest joint responsible for supporting the weight of the head. Thus, the compliancy and ability to maintain position in the absence of power provided by the SEA are particularly useful.

entire weight of the head. As shown in figure 3-3, each SEA is equipped with a linear spring that is placed in series with the motor, which act like a low-pass filter reducing the effects of high impact shocks, while protecting the actuator and the robot. These springs also, in isolating the drive nut on the ball screw, provide a simple method for sensing force output, which is proportional to the deflection of the spring (Hooke's Law, F = kx where k is the spring constant). This deflection is measured by a linear potentiometer between the frame of the actuator and the nut on the ball screw. Consequently, force control can be implemented which allows the joint to be compliant and to safely interact with external forces in the environment. We implemented a simple gravity compensation module to adapt the force commands for different orientations of the neck, using the method described in [70].

Additionally, the ball screw allows the SEA to maintain position of the head when motor power is turned off. Collapsing joints upon power shutdown is a vulnerable point for robots, especially large and heavy ones.

Safeguarding position-controlled axes The rest of the DOFs rely on position feedback for motion control and thus are entirely dependent on accurate position sensors. Incorrect reading or faulty sensors could lead to a serious damage to the robot, so redundant relative encoder and potentiometer are utilized in each joint. The potentiometer provides absolute position measurement and eliminates the need for calibration routines during startup. Both sensors serve as a comparison point to detect failures in the other. Each joint is also designed to be back drivable and equipped with a physical stop in order to reduce failure impacts.

Electrical cables and connectors Placement of electrical cables is frequently an afterthought in robot designs, as a broken or loose cable is one of the most common failure sources. Routing over thirty cables inside the robot without straining each other or obstructing the joints is not an easy task. Mertz's head design includes large cable passages through the center of head differential and the neck. This allows cable bundles to be neatly tucked inside the passages from the eyes all the way through to the base of the robot, thus minimizing cable displacement during joint movement. On the controller side, friction or locking connectors are used to ensure solid connections.

3.3.2 Low-Level Motor Control

Whereas our previous work has made use of off-the-shelf motion control products and PC nodes, we have implemented custom-made hardware for Mertz's sensorimotor and behavior control. Off-the-shelf products, though powerful and convenient, are limited to a set of predetermined capabilities and can be unduly complex. Customizing our hardware to more precise specifications gives us greater control and flexibility. One caveat is that it took more time to develop custom-made hardware to a reliable state.

The custom-made motor controller is built using the Motorola DSP56F807. The controller supports PWM generation, encoder support, and A/D conversion for all existing axes. The amplifier uses the LMD18200 dual H-Bridge which accommodates up to 3A continuous output as well as current sensing. The ability to sense current is crucial as it provides a way to detect failures involving stalled motors. Particular attention was given to protect the robot against power cycle or shutdown. The motors and controller use separate power sources. Thus, we added simple circuitry to prevent the motors from running out of control if the controller happens to be off or reset, which could happen depending on the controller's initial state upon power reset.

Simple PD position and velocity control were implemented on the head and eye axes. Taking advantage of the Series Elastic Actuators, we use force feedback to implement force control for the neck joint. A simple PD position control was then placed on top of the force control. Various bounds are enforced to ensure that both position/force feedback and motor output stay within reasonable values.

Each axis is equipped with a potentiometer and a digital relative encoder. This allows for a fast and automatic calibration process. Upon startup each axis is programmed to find its absolute position and then relies on the encoder for more precise position feedback. This streamlines the startup sequence to two steps which can be performed in any order: turning on the motor controller and turning on the motors. While the robot is running, the motor controller can be reset at any time, causing the robot to re-calibrate to its default initial position and resume operation. The motors can also be turned off at any time, stopping the robot, and turned back on, letting the robot pick up where it left off.

3.3.3 Modular Control

Figure 3-4 illustrates the interconnections among the robot's hardware and software modules. The rectangular units represent the hardware components and the superimposed grey patches represent the software systems implemented on the corresponding hardware module. We have arranged these subsystems in the same order as the control layers. We paid careful attention to ensure that each layer of control is independent, such that the robot is safe-guarded upon removal of higher level control while the robot is running. Motor control and behavior layers are implemented using embedded microprocessors, instead of more powerful but complex PCs, such that they can run autonomously and reliably at all times without having to worry about the many other processes running on the computers.

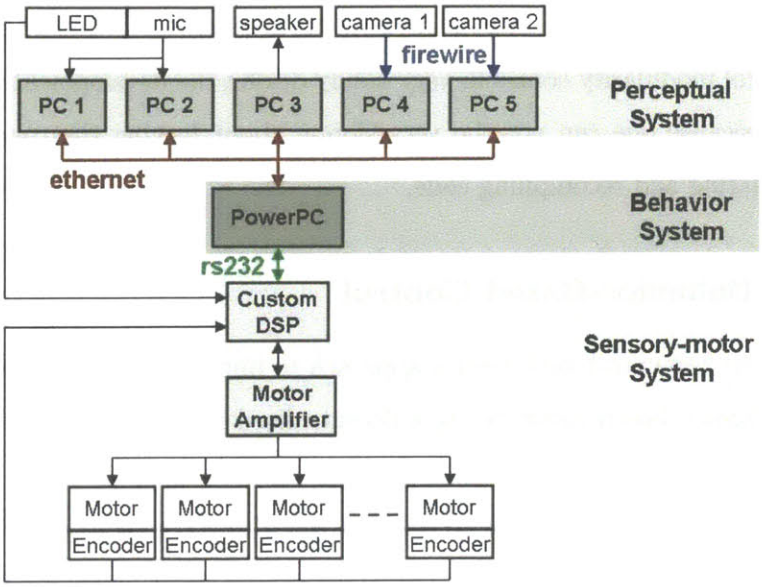


Figure 3-4: The robot's hardware and software control architecture. Rectangular units represent hardware components. The superimposed grey patches represent the software systems implemented on the corresponding hardware. We carefully designed each control layer to be modular and independent. Higher level control layers can be removed at any point without disrupting the robot's operation.

We will now go through each control layer as shown in figure 3-4 and describe how they interact and affect the robot's overall behavior. Suppose we strip all control layers, including the lowest level motor control layer. In this condition, the amplifier is guaranteed to produce zero output to the motors, until the motor controller is back up. This essentially protects the robot in the event of power loss or power cycle to the motor controller. If the motor controller is put back into the system without any other control layers, each degree of freedom will automatically calibrate into its predefined zero position and stay there. At this point, the force control for the neck joints will be active, causing both joints to be compliant to external forces. If the behavior system is now added to the configuration, it will generate random motion commands to all degrees of freedom. If the vision system is also turned back on, it will communicate to the behavior system, which will in turn send commands to have the robot respond to salient visual targets. Similarly, if the audio system is added, it will communicate to the behavior system and activate the robot's speech behavior. This control modularity comes in very handy during the development and debugging process, because one can now be very sloppy about leaving the robot's motors on while updating and recompiling code.

3.3.4 Behavior-Based Control

We used the behavior-based control approach to implement Mertz's behavior system [2]. A behavior-based controller is a decentralized network of behaviors. Each behavior independently receives sensory input and sends commands to the actuators, while communicating with each other. The overall robot's behavior is the result of an emergent and often unpredictable interaction among these behavioral processes. This decentralized approach allows for a more robust implementation, as the robot's behavior system may still work partially even if some components of the robot's system are non-functional.

The robot's behavior system was implemented in L/MARS [19]. L is a Common Lisp-based programming language specifically designed to implement behavior-based programs in the incremental and concurrent spirit of the subsumption architecture [18]. The L system has been retargetted to the POWERPC and is running on a Mac Mini computer. The MARS (Multiple Agency Reactivity System) language is embedded in L and was designed for programming multiple concurrent agents. MARS allows users to create many asynchronous parallel threads sharing a local lexical environment. Groups of these threads are called *assemblages*. Each *assemblage* can communicate to others using a set of input and output ports. As defined in the subsumption architecture, wires or connections between ports can either *suppress* or *inhibit* each other. *Assemblages* can be dynamically killed and connections among ports can be dynamally made and broken.

3.3.5 Long-Term Software Testing

As mentioned above, all software systems must be able to run continuously for many hours. Long term testing and experiments have been very helpful in identifying emergent and occasional bugs, as well as memory leaks. We conducted multiple design and testing iterations of various software components at different environmental settings to avoid overspecialized solutions.

3.4 Social Interface Design

3.4.1 Visual Appearance

As humans, like all primates, are primarily a visual creature, the robot's appearance is an important factor and should be designed to facilitate its role as a social creature. There have been some attempts to study how a humanoid robot should look like from the perspective of human robot interaction [29, 64]. However, other than a number of resulting guidelines, the search space is still enormous. We intuitively designed the robot to be somewhat human like, child-like, and friendly, as shown in figure 3-5.

3.4.2 Generating and Perceiving Social Behaviors

Results in a human-robot interaction experiment suggest that the ability to convey expression and indicate attention are the minimal requirements for effective social interaction between humans and robots [20].

We have incorporated degrees of freedom into the design of Mertz's head and face such that the robot can produce a set of social gestures. Two pan and one tilt DOF are dedicated to generate various human eye movement categories, e.g. saccades, smooth pursuit, vergence, vestibular-ocular reflex, and the opto-kinetic response. These DOF also allow the eyes to gaze in all directions. The head has three degrees of freedom to pan, tilt, and roll, yielding many possible head movements. Mertz has a pair of eyebrows, eyelids, and lips for generating a number of facial expressions. The lips also serve as a visual complement for the robot's speech synthesizer. The two-DOF neck

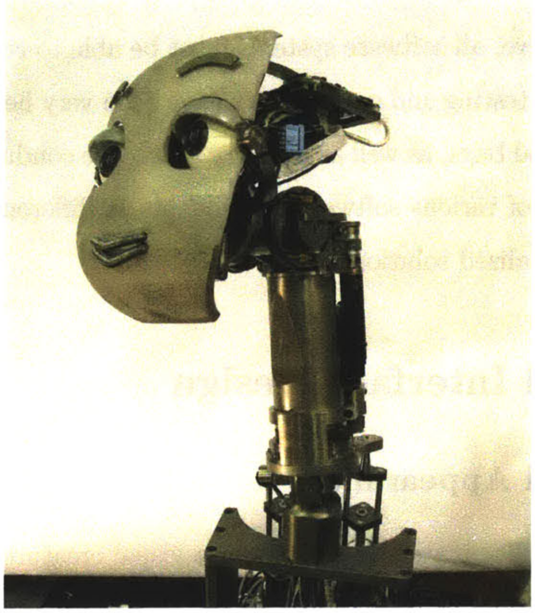


Figure 3-5: A close-up image of the Mertz robot. We intuitively designed the robot's visual appearance to facilitate its sociability. Overall, we opted for a child-like and friendly look. The robot can generate facial expressions by actuating the lips, eyelids, and eyebrows. The lips also move corresponding to the robot's speech.

adds to the robot's expressiveness by enhancing head movements as well as producing a number of overall postures. Clearly, these mechanical DOFs only provide a part of the story, as they must be controlled in conjunction with the perceptual systems. The robot's high level behavior control is described in more detail in section 4.6.

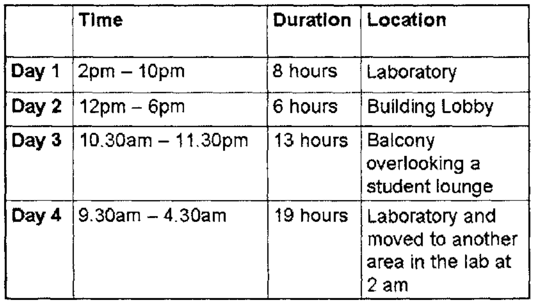
In addition to being socially expressive, the robot must also be responsive to human social cues. Toward this goal, the robot's first task is to detect the presence of humans. Once the robot locates a person's face, it then has to make eye contact and track the person. This task has been well demonstrated in many social robotic platforms [11, 13, 63, 62]. In our setup, where the robot has to deal with an unstructured environment, we found that the complexity of this particular task has increased significantly. In addition to drastic lighting and acoustical variations across different locations and times of day, the robot has to interact with a large number of people, often simultaneously. Without any specific instructions, these individuals display a wide range of behavior patterns and expectations. These complexities have triggered multiple design iterations and incremental changes in our implementation throughout the project.

In the robot's final implementation, the robot is capable of detecting people, attending to a few people at a time, and engaging in a simple verbal interaction. More details on the robot's perceptual mechanism will be covered in section 4.3.

3.5 Some Early Experiments

We conducted a number of experiments to evaluate different subsystems at various stages of the robot development. These experiments range from one to seven days long and were carried out in different locations. In this section, we will describe three of these experiments. We briefly state the setup of each experiment, illustrate the failures which occured during the experiment, and summarize a number of lessons learned.

During these experiments, the robot collected a set of numerical data to evaluate the performance of various subsystems. In addition, there was also a set of qualitative data that we observed by watching the robot from a distance. A carefully designed human subject experiment would probably generate a set of interesting quantitative data from these spontaneous human-robot interactions. However, this would require a more stringent protocol, including written permission forms, which would alter the nature of the experiment in some undesirable ways. Even though we can not present these observed lessons in numbers, we present a qualitative description of these lessons in this section, as they are very valuable in understanding the problem scope. Moreover, as we interleaved these experiments with the robot development process, many of these lessons were later incorporated into the robot's final implementation, which will be described in the next chapter.



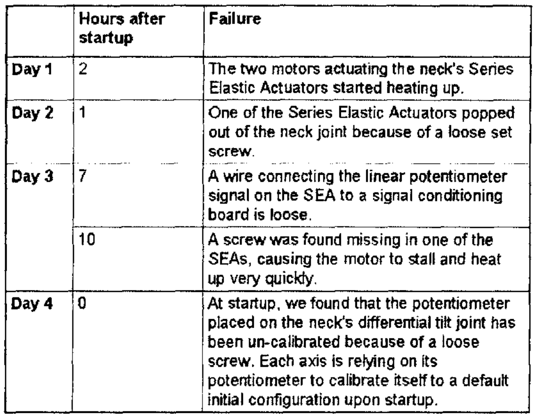
	Time	Duration	Location
Day 1	2pm – 10pm	8 hours	Laboratory
Day 2	12pm - 6pm	6 hours	Building Lobby
Day 3	10.30am – 11.30pm	13 hours	Balcony overlooking a student lounge
Day 4	9.30am – 4.30am	19 hours	Laboratory and moved to another area in the lab at 2 am

Table 3.1: Schedule, time, and location of an early experiment to evaluate the robot's reliability. We setup the robot to run at four different locations for a total of 46 hours within 4 days. At this time, the robot had a very simple visual attention system for orienting to various visual targets. Our goal was to study failure modes while the robot operated in its full range of motion.

3.5.1 Experiment 1

Setup We conducted a four-day long experiment at the very early stage of the robot development. At this time, the robot only consisted of the head and neck frame. The robot's face was still in the design phase. We equipped the robot with a simple behavior system where it simply orients to salient visual targets, i.e., faces, skin color, and saturated colors. The goal was to test the robot's robustness and study failure modes while the robot operated in its full range of motion. The robot ran for 46 hours within 4 days at four different locations. We also collected raw visual data to observe variations across different times and locations.

The experiment schedule is shown in table 3.1. The shortest and longest duration are 6 and 19 hours respectively. Initially, the experiment was conducted with supervision. As the robot showed a reasonable level of reliability (with the exception of the neck joint that had to be rested every couple hours), we started leaving the robot alone and checked on it every hour. During the experiment, people were allowed to approach but not touch the robot. While unsupervised, a sign was placed near the robot to prohibit people from touching it.



	Hours after startup	Failure
Day 1	2	The two motors actuating the neck's Series Elastic Actuators started heating up.
Day 2	1	One of the Series Elastic Actuators popped out of the neck joint because of a loose set screw.
Day 3	7	A wire connecting the linear potentiometer signal on the SEA to a signal conditioning board is loose.
	10	A screw was found missing in one of the SEAs, causing the motor to stall and heat up very quickly.
Day 4	0	At startup, we found that the potentiometer placed on the neck's differential tilt joint has been un-calibrated because of a loose screw. Each axis is relying on its potentiometer to calibrate itself to a default initial configuration upon startup.

Table 3.2: List of observed failures during the experiment. All failures originate from mechanical problems in the neck joint and its Series Elastic Actuators. It is important to note that this experiment is slightly biased in finding mechanical faults, since most of the hardware and software errors are fixed during the development process.

Failure Modes. The head and eye axes are so far free of failures. Most of the failures originated from the mechanical failures on the neck SEA actuators. Table 3.2 lists each failure that occurred during the experiment.

All observed failures involve the neck joint and its Series Elastic Actuators. Loose screws seem to be particularly problematic. A probable explanation is that the neck actuators are constantly in motion. The force control loop produces output that is proportional to the linear pot signal plus some noise. A series of filters have now been put in place in order to minimize noise. In addition, the load of each SEA motor is very small causing the control output to be very sensitive to even a trivial amount of noise. A dead-band was placed in software to reduce this effect, which eliminates some but not all of the actuator's jitter. We also found that decreasing friction on the SEA's ball screw helps to reduce jitter. In addition, we put additional protection for the screws, i.e., using loctite on as many screws as possible. Lastly, it is important to note that the experiment setup is biased in finding mechanical failures, since much time was spent to make sure that the hardware and software systems are working properly during the development process.

3.5.2 Experiment 2

Setup. In this experiment, the robot ran from 11 am to 6 pm for 5 days at different spaces at the first floor of the MIT Stata Center. At this time, the robot's head and face had been completed. The goal of this experiment was to further evaluate robustness and the robot's potential for soliciting human passersby to engage in an impromptu interaction.

A written sign was placed on the robot to request people to interact with the robot. The sign explained that this was an experiment to test how well the robot operates in different environments and warned that the robot would be collecting face images and audio samples. A set of bright colored toys were placed around the robot. We monitored the robot from a distance to encourage people to freely interact with the robot, instead of approaching us for questions. Figure 3-12 shows the robot on day 5 of the experiment.

Failure Modes The robot ran without any mechanical failures for the first 3 days of the experiment. On the 4th day, we detached one of the SEAs which seemed to be exposed to more friction than the other one, tightened the screws, and re-attached it. We also had to re-calibrate the motor control software to adapt to the resulting mechanical changes. The robot continued running without any failures for the rest of the experiment. However, as we still encounter throughout the project, human error is simply difficult to avoid. Due to human error, we lost some recorded data during this experiment.

Experimental Results During the experiment, we recorded the robot's visual input and the tracker's output every second. We labelled a sequence of 14186 frames collected during a close to four hour period on day 2. Figure 3-6 shows the output of the robot's tracker during this period. For approximately 16.9% of the sequence, the robot tracked correctly segmented faces. For a small part of the sequence the robot tracked faces that either included too much background or partially cropped and tracked bright colored toys and clothing.

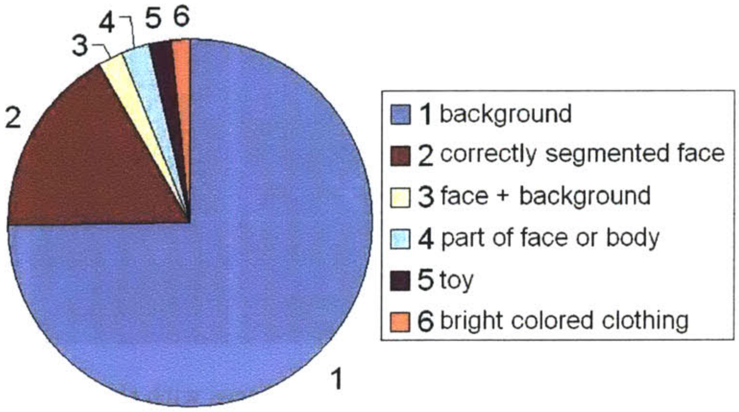
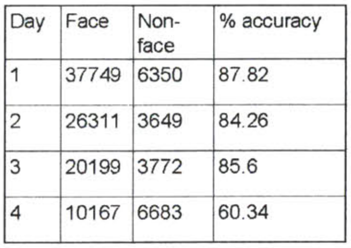


Figure 3-6: The breakdown of what the robot tracked during a sequence of 14186 frames collected on day 2.

We also collected every frame of segmented face images that were detected by the frontal face detector throughout the experiment, which amounted to a total of 114,880 images from at least 600 individuals. The robot uses the frontal face detector developed by Viola and Jones [98]. Figure 3-7 shows the breakdown of the face detector's false positive error rates during each day, excluding one day due to file loss. These results suggest that the robot is able to acquire a significant set of face images because people do interact with the robot closely enough and for long enough durations.

The robot received over 1000 utterances per day. An utterance starts when the audio input exceeds a minimum energy threshold and stops when a long enough pause is detected. We transcribed a portion of these utterances (3027 audio samples from at least 250 people) to analyze what the robot heard. Each data portion was taken from a continuous sequence of speech input spanning a number of hours in each day. Due to some file loss, we were only able to transcribe less than 300 utterances on day 5. As shown in Figure 3-8, approximately 37% of the total utterances are intelligible robot directed speech. The rest are made up of background noise, the robot's own speech, human speech that is unintelligible (cropped, foreign language, muddled, etc),



Day	Face	Non-face	% accuracy
1	37749	6350	87.82
2	26311	3649	84.26
3	20199	3772	85.6
4	10167	6683	60.34

Figure 3-7: The face detector detected 114,880 faces with the average false positive of 17.8% over the course of 4 days. On the right is a sample set of the 94,426 correctly detected faces.

and human speech directed not at the robot.

Figure 3-9 shows the number of words in each utterance from the set of intelligible human speech. One-word utterances make up 38% of all intelligible human speech and 38.64% of robot directed speech. Two-word utterances make up 17.69% of all intelligible human speech and 17.93% of robot directed speech. Approximately 83.21% of all intelligible human speech and 87.77% of robot directed speech contain less than 5 words.

We are also interested in finding out whether or not the robot may be able to acquire a lexicon of relevant words. In particular, we would like to assess whether a set of words tends to be repeated by a large number of people. Figure 3-10 illustrates the top fifteen most frequently said words during each day and a set of frequently said words that are shared by 3 days or more during the experiment.

Figure 3-11 illustrates the difference in average pitch and pitch gradient values of robot directed speech versus non-robot directed speech on each experiment day. Both female and male speakers tend to speak with higher pitch average to the robot versus to other people.

These results seem to suggest to people do in fact speak to the robot. Moreover, they tend to speak to it like they would to a young child. The frequency of one-word utterances seems to be high enough to provide the robot with a starting point

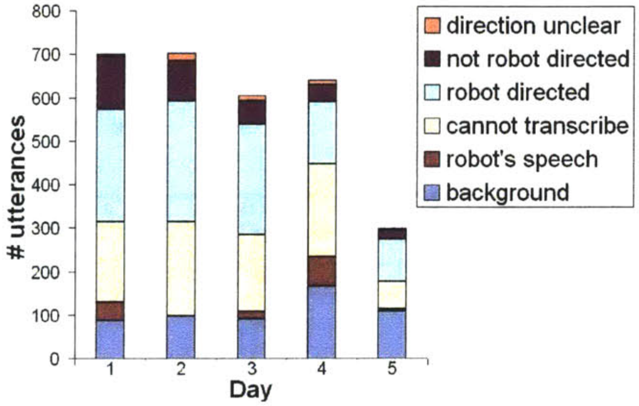


Figure 3-8: The characteristic of speech input received by the robot on each day of the experiment.

for unsupervised lexical acquisition. Lastly, a set of common words tend to be repeated throughout the experiment despite the large number of speakers and minimal constraints on the human-robot interaction.

3.5.3 Experiment 3

Setup. This experiment was done in two parts. We first conducted a 5-hour experiment inside the laboratory. We requested ten people to come and interact with the robot. A few days later, we conducted a six-hour experiment where the robot interacted with over 70 people in a public space outside the laboratory. The goal of these experiments was to evaluate the robot's multi-modal attention and spatio-temporal learning systems. The setup of the robot and experiment instruction was identical to the setup described in Experiment 2.

Failure Modes. One of the robot's computers has been problematic and finally failed during the experiment. We also discovered a software bug because of a counter which became too large and cycled back to zero. We did not encounter this error during the shorter-term testing periods. A similar error occured in data recording, where a log file storing a large amount of output from the robot's spatio-temporal

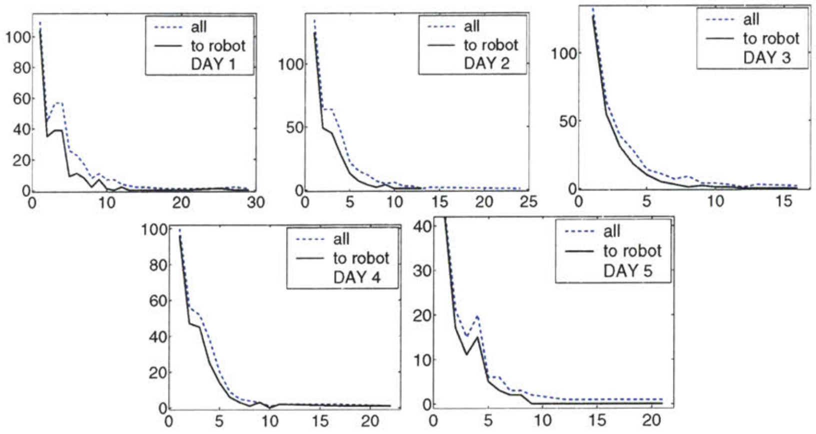


Figure 3-9: Number of words (x axis) in utterances (y axis) collected during each day of the experiment. Dashed line: all transcribed data. Solid line: robot directed speech only.

learning system grew too large and killed the program.

In a later informal experiment in the public lobby, the robot's head pan and tilt motors broke. Since the failure occured when the robot was unsupervised, we don't know the precise cause of the failures. Our hypothesis is that one of the motor's gearheads may have failed and caused a chain reaction to another joint.

3.5.4 Summary and Lessons

Based on the numerical results and visual observation of these experiments, we extracted a number of lessons which have triggered a set of incremental changes in the robot's development process.

The environment. As expected, the robot's environment is very dynamic and challenging. Mertz was approached by an average of 140 people a day. The robot was approached by one individual, small groups, and at times large groups of up to 20 people. The robot often perceives multiple faces and speech from multiple people simultaneously. Some people spoke to the robot, while some spoke to each other. Additionally, the auditory system's task is made even more difficult by the high level

	15 most frequent words
Day 1	it/it's,you,hello,I,to,is,what,hi,are,the,a,Mertz,here,your,this
Day 2	you,hello,it/it's,what,I,hi,yeah,are,the,oh,to,is,Mertz,a,your
Day 3	hello,it/it's,you,hi,Mertz,bye,to,robot,are,the,I,what,hey,how,is
Day 4	you,hello,it/it's,what,hi,Mertz,are,I,bye,this,here,how,is,robot,to
Day 5	you,hello,it/it's,hi,I,what,are,oh,how, say,the,a,at,can,is

Shared by	Common most frequent words
5 days	hello, you, it/it's, hi, what, are, I, is
4 days	to, Mertz, the, this, how, hey, what's
3 days	a, here, your, oh, can

Figure 3-10: The top 15 words on each experiment day and the common set of most frequently said words across multiple days.

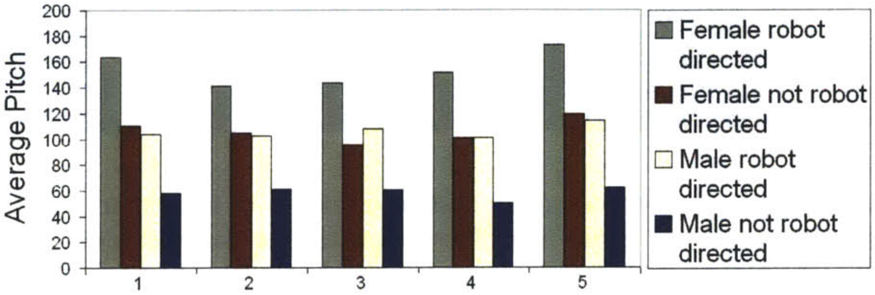


Figure 3-11: Average pitch values extracted from robot directed and non-robot directed speech on each day of the experiment.

of background noise. It is a very erratic environment for the robot's perceptual and attention system.

As we move the robot to different locations, we encounter drastic changes in the visual and acoustical input. We also continue to discover unexpected features which were absent inside the laboratory environment, but caused various difficulties for the robot's perceptual and attention system. False positive error is particularly troublesome. Detection of a face in the background or a large bright orange wall tends to dominate and steer the robot's attention system away from real salient stimuli. Variation in lighting and background noise level is also very problematic. Figure 3-13

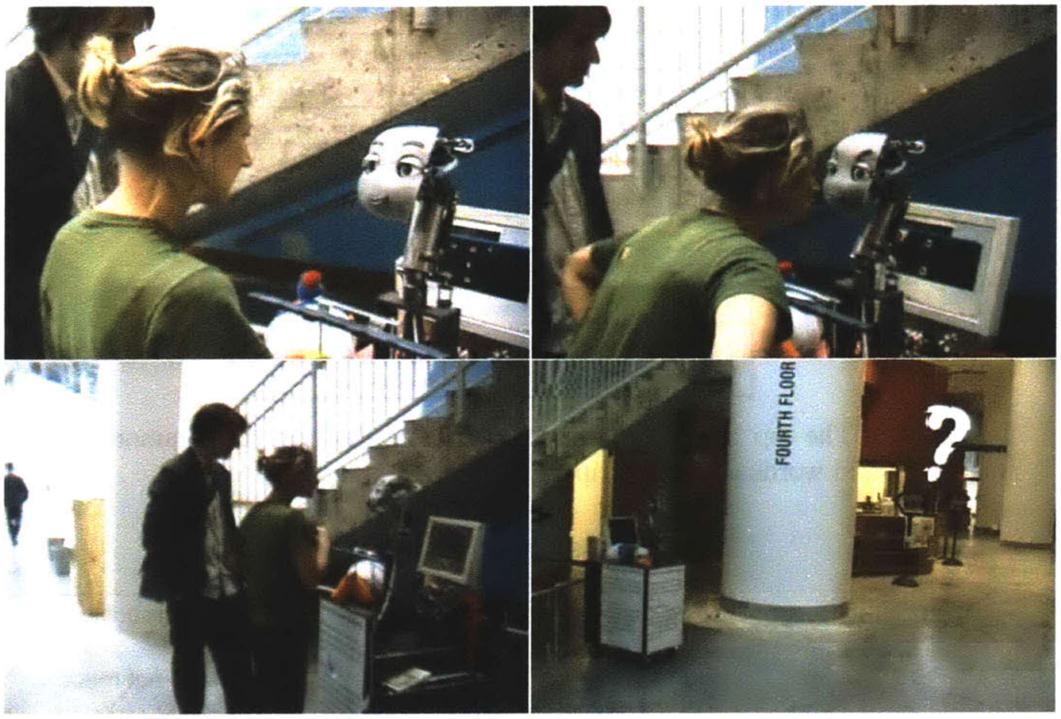


Figure 3-12: A sample interaction session on day 5 of the experiment. The robot ran continuously for 7 hours in a different public location each day. A written sign was placed on the robot: " Hello, my name is Mertz. Please interact with me. I can see, hear, and will try to mimic what you say to me." Some bright colored toys are available around the robot. On the bottom right corner is a full-view of the robot's platform in the lobby of the MIT Stata Center.

contains a set of face images collected in different locations and times of day. A fixed sound detection threshold which works well inside the laboratory is no longer effective when the robot is moved outside the laboratory. The much higher background noise causes the robot to perceive sound everywhere and overwhelms the attention system. Figure 3-14 shows the output of the robot's attention system during two experiments, inside and outside the laboratory. Each plot contains different measurements for what the robot attended to and shows how the number of sound event occurrence dominates over the visual events when the robot is moved outside the laboratory. [NOTE: ADD HERE ABOUT FALSE POSITIVE DETECTION ERRORS]

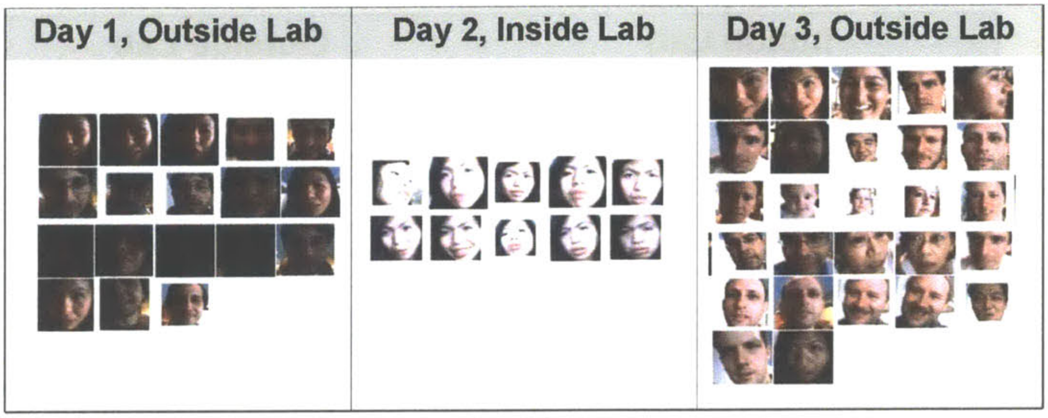


Figure 3-13: A set of face images collected in different locations and times of day.

These difficulties have led us to put a lot more efforts into the robot's attention system than we initially expected. We upgraded the robot's attention system to include an egocentric representation of the world, instead of simply relying on the retinal coordinates. We also enhanced the robot's attention system to utilize spatio-temporal context and correlate multi-modal signals to allow for a more robust integration of the noisy perceptual input. Research in computer vision and speech recognition has made a lot of advances in dealing with these environmental variations. However, we believe that errors and imperfections in the robot's various subsystems are simply inevitable. Thus, a robust integration of the robot's subsystems is a crucial element in the path toward intelligent robots.

The passersby. There is a large variation in the level of expectations and behavior patterns in the large set of naive passersby. Many people spoke naturally to the robot, but some simply stared at the robot. Some people who successfully attracted the robot's visual attention then tried to explore the robot's tracking capabilities by moving around and tilting their heads. Many people were not aware of the robot's limited visual field of view and seemed to expect that the robot should be able to see them wherever they were. When they realized that the robot was not aware of

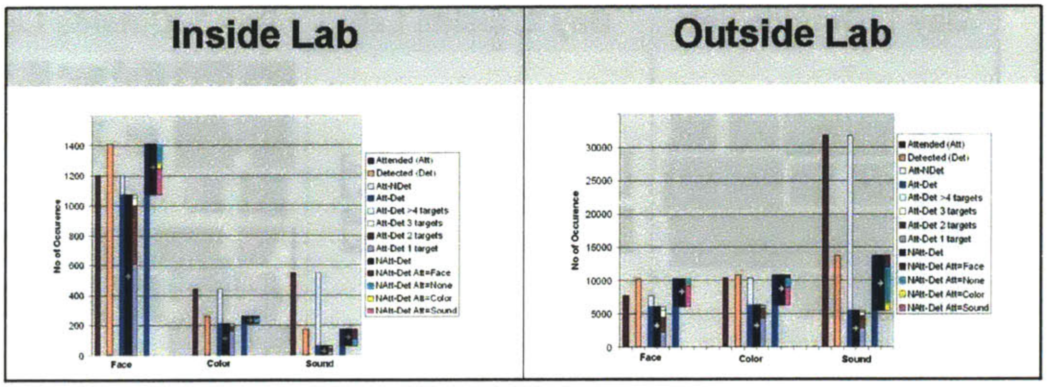


Figure 3-14: The robot's attention system output during two experiment days, inside and outside the laboratory.

their presence, many used speech or hand movements to try to attract the robot's attention. This led to a decision to give Mertz a sound localization module, which has been a tremendous addition to its ability to find people.

At this point, the robot's auditory system consisted of a phoneme recognizer. The speech synthesizer then simply mimicked each extracted phoneme sequence. This led to a lot of confusions for many people. The robot often produces unintelligible phoneme sequences due to the noisy recognizer. Even when the robot produces the correct phoneme sequence, people still had trouble understanding the robot. Many people also expect that the robot understands language and become frustrated when the robot is not responding to their sentences. For this reason, we have incorporated a more complex word recognition system into the robot.

Learning while interacting. The most important lesson that we learned during these early experiments is the difficulty of having to interact with while learning from the environment. In our setup, there is no boundary between testing and training stages. The robot's attention system has to continually decide between two conflicting choices: to switch attention to a salient input which may lead to learning targets or to maintain attention to the current learning targets. Moreover, even though the robot successfully tracked over a hundred thousand faces, the accuracy required from tracking for interaction is much lower than tracking for learning. In order to collect effective face data for recognition purposes, the robot has to be able to perform same person tracking accurately. This is a very difficult task when both the robot's cameras and people are constantly moving. Additionally, the simultaneous presence of multiple faces further increases the task complexity. We further discuss this topic in section 4.1 and present the implications, as reflected in the final implementation of the robot's attention system in section 4.4.

Chapter 4

Finding, Interacting With, and Collecting Data From People

In the previous chapter, we have demonstrated a robotic platform that is capable of operating for many hours continuously and soliciting spontaneous interaction from a large number of passersby in different public locations. We also described some early experiments and showed that there is still a large gap between the ability for superficial interaction with many passersby and our final goal of unsupervised incremental individual recognition.

In this chapter, we present the implementation details of the robot's perceptual, attention, and behavior systems. The robot has to organize its perceptual and behavior systems not only to solicit interaction, but also to regulate these interactions to generate learning opportunities. More specifically, as listed in section 1.4, the robot has to perform the following tasks automatically:

1. attract passersby to approach the robot, engage them in spontaneous social interaction, and trigger natural interactive behavior from them;

2. regulate the interaction in order to generate opportunities to collect data from as many people as possible;

3. detect, segment, store, and process face images and voice samples during interaction with people;

4. use tracking and spatio-temporal assumptions to obtain a sequence of face images and voice samples of each individual as a starting point toward unsupervised recognition.

4.1 Challenges and Approach

As we have shown in Chapter 3, Mertz was able to collect a large number of faces during some early experiments due to the extremely robust face detector, developed by Viola and Jones [98]. This of course assumes an initial condition where the robot is facing the person somewhat frontally. Even though this is not always the case, the robot still managed to track over 100,000 faces from over 600 individuals in an early 7-day long experiment, described in section 3.5. However, the further task of interacting while collecting face and voice data of each individual has generated additional load and complexity, especially on the robot's attention system.

The attention system serves as a front gate to hold back and select from an abundance of streaming sensory input. In the absence of such filtering, both the robot's controller and learning mechanism will be overwhelmed. The importance of an attention system for learning has been discovered in many research areas [104, 20]. Studies of the human's visual system suggest that selective visual attention is guided by a rapid and stimulus-driven selection process as well as by a volitional controlled top-down selection process [73]. Incorporating top-down control of attention has been explored in [34, 41, 42]. However, the top-down attention control was mostly simulated manually in most of these systems. Our initial implementation of the attention system originated from [14]. However, the requirement for operation in unstructured environments has triggered the need for many additional functionalities. Many properties of the robot's current attention system were inspired by the Sensory Ego-sphere [41].

In our setup, where there is no boundary between testing and training stages, the robot has to perform the parallel task of interacting with while collecting data and learning from the environment. This task is difficult for a number of reasons.

Firstly, the robot's attention system faces conflicting tasks, as it has to be reactive to find learning targets in the environment but also persistent to observe targets once they are found. In the human's visual attention system, this dichotomy is reflected in two separate components: the bottom-up (exogenous) and top-down (endogenous) control [79].

Secondly, attending to learn in an unconstrained social environment is a difficult task due to noisy perceptual sensors, target disappearing and reappearing, simultaneous presence of multiple people, and the target's or robot's own motion. Same person tracking in subsequent frames is an easy task for the human's visual system since we are very good in maintaining spatiotemporal continuity. Even when our heads and eyes move, we can easily determine what have moved around us. Unfortunately, for a robot active vision system, this is not the case. The robot essentially has to process each visual frame from scratch in order to re-discover the learning target from the previous frame. Tracking a person's face in order to learn to recognize the person is a somewhat convoluted problem. The robot has to follow and record the same person's face in subsequent frames, which requires some knowledge about what this person looks like, but this is exactly what the robot is trying to gather in the first place.

An additional complexity is introduced by the trade-off between timing and accuracy requirements of the interaction and learning processes. The interaction process needs fast processing to allow for timely responses, but less accuracy since the consequence of attending to the wrong thing is minimal. The data collection process is not as urgent in timing, but it needs higher accuracy. The consequence of incorrect segmentation or placing the wrong data for the wrong person is quite significant for the robot's learning process. Interestingly, this dichotomy is also reflected in the separate dorsal *where* and ventral *what* pathways in the human's visual system, for locating and identifying objects [37].

We have designed the robot's attention system to address some of the issues mentioned above, by incorporating object-based tracking and an egocentric multimodal attentional map based on the world coordinate system [1, 41]. The attention system receives each instance of object-based sensory events (face, color segment, and sound) and employs space-time-varying saliency functions, designed to provide some spatiotemporal short-term memory capacity in order to better deal with detection errors and having multiple targets that come in and out of the field of view.

In addition, inspired by the coupling between human infants' attention and learning process, we implemented a spatiotemporal perceptual learning mechanism, which incrementally adapts the attention system's saliency parameters for different types and locations of stimuli based on the robot's past sensory experiences. In the case of human infants, the attention system directs cognitive resources to significant stimuli in the environment and largely determines what infants can learn. Conversely, the infants' learning experience in the world also incrementally adapts the attention system to incorporate knowledge acquired from the environment. Coupling the robot's attention system with spatiotemporal perceptual learning allows the robot to exploit the large amount of regularities in the human environment. For example, in an indoor environment, we would typically expect tables and chairs to be on the floor, light fixtures to be on the ceiling, and people's faces to be at the average human height. Movellan et al presents an unsupervised system for face detection learning by exploiting contingency of an attention system associating audio signals and peoples' tendency to attend to the robot [23].

4.2 System Architecture and Overall Behavior

Figure 4-1 illustrates the robot's system architecture. The robot's visual system is equipped with detectors and trackers for relevant stimuli, i.e., faces, motion, and color segments. The auditory system detects, localizes, and performs various processing on sound input. Each instance of a perceptual event (face, motion, color segment, and sound) is projected onto the world coordinate system using the robot's forward kinematic model and is entered into both the multi-modal egocentric attention and spatio-temporal learning map. The attention system computes the target output and sends it to the robot's behavior system to calculate the appropriate next step. The spatio-temporal learning process incrementally updates the attention's saliency

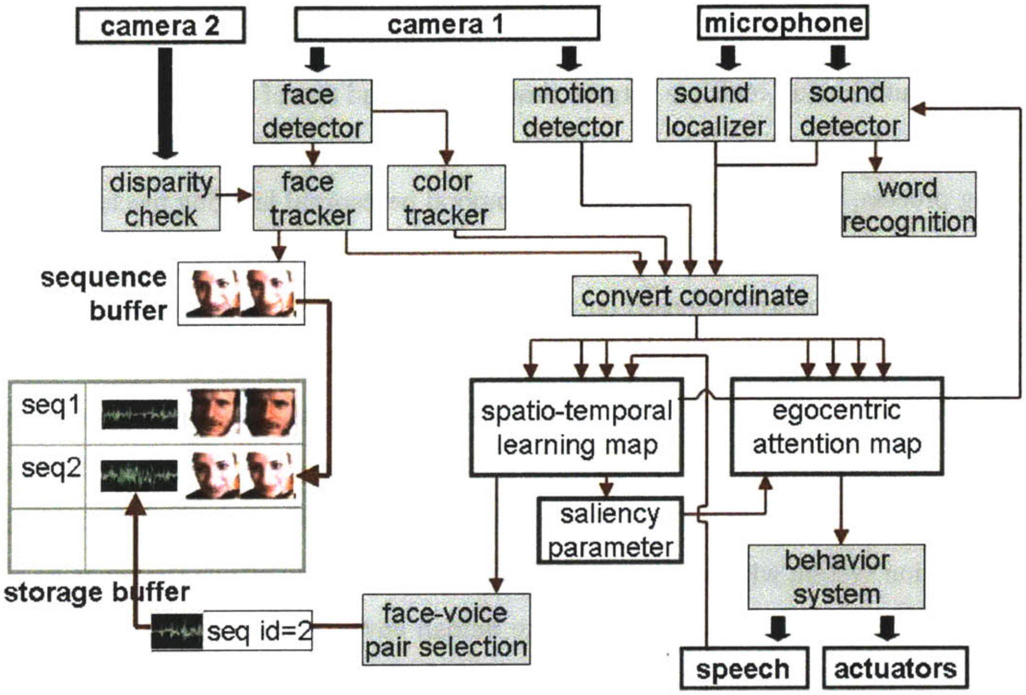


Figure 4-1: The robot's overall system architecture, consisting of a perceptual, attention, and behavior system. The robot's visual and auditory system detects and localizes various stimuli. Each instance of a perceptual event (face, motion, color segment, and sound) is projected onto the world coordinates system into both the multi-modal attention and spatio-temporal learning map. The attention system computes and sends the target output to the robot's behavior system. The spatio-temporal learning process incrementally updates the attention's saliency parameters.

parameters, which is then fed back into the attention map. In parallel, each perceptual event is also filtered, stored, and processed to automatically generate clusters of individuals' faces and voice segments for incremental person recognition.

The robot's vision system receives a 320 × 240 color frame from each camera, but processes at half that resolution to allow real-time processing. However, the system retrieves the higher resolution image when it segments and stores face images. The vision system and communication among PC nodes were implemented using YARP, an open source vision software platform developed by a collaboration effort between the MIT Humanoid Robotics Laboratory and LIRA-Lab at University of Genova. [59] YARP is a collection of libraries, providing various image processing functionalities and message based interprocess communication across multiple nodes.

4.3 Perceptual System

The robot is capable of detecting a set of percepts, i.e. face, motion, color segment, and sound. We describe the implementation details of each perceptual subsystem below and present how they are integrated in the next section.

4.3.1 Face Detection and Tracking

In order to be responsive to people's interaction attempt, MERTZ must be able to detect the presence of humans and track them while they are within its field of view. In order to detect and track faces, we are combining a set of existing face detection and feature tracking algorithms. We will not go into the implementation details, as these information are available in the publication of each original work.

We are using the frontal face detector developed by Paul Viola and Michael Jones [98]. The face detector occasionally finds a false positive face region in certain backgrounds, causing the robot to fixate on the floor, wall, or ceiling. This is especially problematic during long experiments where there was a lot of down time when no one is around. We implemented a SIFT-based feature matching module to calculate a sparse disparity measure between the face region in the images from the left and right cameras. Using an expected ratio of estimated disparity and face size, the module rules out some false faces in the background that are too far away.

Since both people and the robot tend to move around frequently, faces do not remain frontal to the cameras for very long. We are using a KLT-based tracker to complement the frontal face detector. The KLT-based tracker was obtained from Stan Birchfield's implementation [7] and enhanced to track up to five faces simultaneusly.

The robot relies on the same-person face tracking module as a stepping stone toward unsupervised face recognition. The idea is that the better and longer the robot can track a person continuously, the more likely that it will collect a *good* sequence of face images from him/her. A *good* sequence is one which contains a set

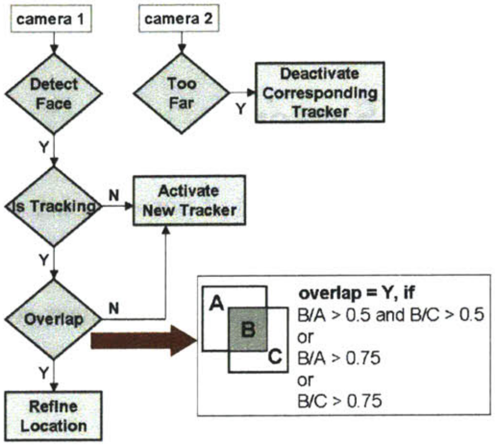


Figure 4-2: The same person face tracking algorithm. We have combined the face detector, the KLT-based tracker, and some spatio-temporal assumptions to achieve same-person tracking.

of face images of the same person with high variations in pose, facial expression, and other environmental aspects.

We have combined the face detector, the KLT-based tracker, and some spatio-temporal assumptions to achieve same-person tracking. As shown in figure 4-2 for every detected face the robot activates the KLT-based face tracker for subsequent frames. If the tracker is already active and there is an overlap in the tracked and detected region, the system will assume that they belong to the same person and refine the face location using the newly detected region. If there is no overlap, the system will activate a new tracker for the new person. The overlapping criteria is also shown in figure 4-2. If the disparity checker catches a false positive detection, the face tracker cancels the corresponding tracking target.

With this algorithm, we make some spatio-temporal assumptions that each sequence of tracked face belongs to the same individual. Of course, this can sometimes be wrong, especially in the case of simultaneous tracking of multiple people. We have observed two failure modes; where a sequence contains face images of two people and where the face image consists of mostly or completely the background region. The first failure happened in multiple occasions where a parent is holding a child. In these cases, their face proximity often confuses the tracker.

4.3.2 Color Segment Tracking

In order to enhance the robot's person tracking capabilities, we augmented the face detector by tracking the color segment found inside the detected face frame. This can be handy when the person's face is rotated too much such that neither the face detector or tracker can locate it.

The color segment tracker was developed using the CAMSHIFT (Continuously Adaptive Mean Shift Algorithm) approach [10], obtained from the OpenCV Library [82]. The tracker is initialized by the face detector and follows a similar tracking algorithm as described in figure 4-2. This tracker can only track one segment at a time, however. We also implemented an additional module to check for cases when the tracker is lost, which tend to cause the CAMSHIFT algorithm to fixate on background regions. This module performs a simple check that the color histogram intersection of the initial and tracked region is large enough [93].

4.3.3 Motion Detection

Since the robot's cameras are moving, background differencing is not sufficient for detecting motion. Thus, we have implemented an enhanced version of the motion detector. Using the same approach for detecting motion with active cameras [33], we use the KLT-based tracking approach to estimate displacement of background pixels due to robot motion at each frame [87]. Object motion is then detected by looking for an image region whose pixel displacement exhibit a high variance from the rest of the image.

In order to complement the motion detector, we implemented a simple and fast color-histogram based distance estimator to detect objects that are very close to the robot. We simply divide the image into four vertical regions and compute color histograms for each region on both cameras. We then calculated the histogram intersection between each region of the two cameras [93]. This method though simple and sparse is at times effective in detecting objects that are very close to the robot. This detection is used to allow the robot to back up and protect itself from proximate objects.

4.3.4 Auditory Perception

The robot's auditory system consists of a sound detection and localization module, some low-level sound processing, and word recognition. An energy-based sound detection module determines the presence of sound events above a threshold. This threshold value was initially empirically determined, but we quickly found that this did not work well outside the laboratory. This threshold is adaptively set using a simple mechanism described in section 4.5. Lastly, the robot inhibits its sound detection module when it is speaking.

In an early experiment, we observed that the robot's limited visual field of view really limit its capability in finding people using only visual cues. The microphone has a built in sound localizer and displays the horizontal direction of the sound source using five indicator LEDs. Thus, we now tap into these LEDs on the microphone to obtain the sound source direction. The presence and location of each sound event are immediately sent to the robot's attention system, allowing the robot to attend to stimuli from a much larger spatial range.

In parallel to this, a separate module also processes each sound input for further speech processing. The robot's speech recognition system was implemented using CMU Sphinx 2 [65]. This module uses a fixed energy threshold to segment and record sound events, because we would like to record as many sound segments as possible for evaluation purposes.

Each recorded segment is processed redundantly for both phoneme and word recognition, as well as for pitch periodicity detection. Firstly, the pitch periodicity detection is used to extract voiced frames and filter phoneme sequences from noise-related errors. The filtered phoneme sequence output is then further filtered by using TSYLB, a syllabification tool to rule out subsequences that are unlikely to occur in the English language [31]. Lastly, the final phoneme sequence is used to filter the hypothesized word list. The robot's behavior system then utilizes this final list to produce speech behaviors, as described in section 4.6.

4.4 Multi-modal Attention System

The robot's attention system consists of an attentional map. This attentional map is a 2D rectangle, which is an approximated projection of the front half of the geodesic sphere centered at the robots origin (a simplified version of the Sensory Ego-Sphere implementation [41]).

The attention system receives multi-modal events from the robot's perceptual systems (face, color segment, motion, and sound). The retinal location of each perceptual event is projected onto the attentional map's world coordinates using the robot's forward kinematic model. Based on this coordinate mapping, each perceptual event is placed on a region inside the attentional map, by altering the saliency level in the corresponding region. The location with the highest saliency value in the map becomes the next target for the robot to attend to.

Figure 4-3 shows an example of an input image and its corresponding attentional map. In this example, the robot is oriented slightly to the left. Thus, the input image occupies the left region of the robot's attentional map. The small white patch represents the detected face. The long ellipse-shaped patch corresponds to the detected sound, since the robot's sound localization module only provides a horizontal direction.

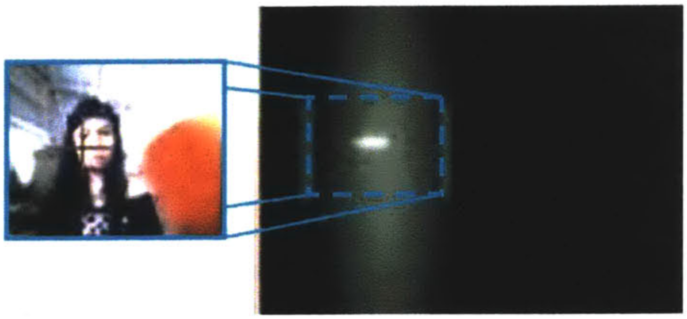


Figure 4-3: An example of an input image and the corresponding attentional map. The attention map is a is an approximated projection of the front half of the geodesic sphere centered at the robots origin The small white patch represents the detected face. The long ellipse-shaped patch corresponds to the detected sound, since the robot's sound localization module only provides a horizontal direction.

4.4.1 Implementation Details

The attentional map consists of 280x210 pixels, indexed by the azimuth and elevation angles of the robot's gaze direction which is generated by actuations of the eyes, head, and neck.

Each pixel at location x, y in the attentional map contains a set of cells $C_{x,y,n}, 0 \leq n \leq 20$.

Each cell $C_{x,y,n}$ consists of the following variables.

- Feature Type $P_{x,y,n}$. The attention system receives four types of perceptual events. $P_{x,y,n} \in$ [face, color segment, motion, sound].
- Input State $St_{x,y,n}$
- Saliency Value $V_{x,y,n}$
- Saliency Growth Rate $Rg_{x,y,n}$
- Saliency Decay Rate $Rd_{x,y,n}$
- Start Time $T0_{x,y,n}$
- Last Update Time $Tl_{x,y,n}$
- Current ID $CurrID_{x,y,n}$.
- Last ID $LastID_{x,y,n}$.
- Inside Field of View $Fov_{x,y,n}$.

The ID variables indicate when an input belongs to the same object. These information are determined differently for each perceptual type. Face IDs are provided by the same-person face tracker. Color Segment IDs are provided by the color segment tracker. Motion and Sound IDs are determined temporally, i.e. events during a brief continuous period of sound and motion are tagged with the same ID.

We begin by describing how the presence of a perceptual event activates a set of cells in the corresponding region. This activation consequently alters the saliency value $St_{x,y,n}$ in each cell $C_{x,y,n}$ over time. We then illustrate how to combine the saliency values from all active cells to produce a saliency map. Lastly, we describe how the resulting saliency map is used to produce the final attention output of the robot's next target.

We first describe how the presence of each perceptual event E activates a set of new cells and affects the Input State variable $St_{x,y,n}$ in each cell $C_{x,y,n}$.

At each time step, before the attention system processes any incoming perceptual events, the variable $St_{x,y,n}$ is reset to 0 for all cells in the attentional map.

```
FOR each perceptual event $E$ of type $p$ and ID $i$
    Assign a region $T$ for $E$ in the attentional map by converting its location from the retinal to the world coordinate
    FOR each pixel $P_{x,y}, x, y \in T$
        Activate a new cell $C_{x,y,n}$ in $P_{x,y}$
        Set $St_{x,y,n} = 1$
        Set $P_{x,y,n} = p$
        Set $Rg_{x,y,n} = AdaptRg_{x,y,p}$ which is incrementally set by the spatio-temporal learning system for type $p$
        Set $Rd_{x,y,n} = AdaptRd_{x,y,p}$ which is incrementally set by the spatio-temporal learning system for type $p$
        Set $LastID_{x,y,n} = CurrID_{x,y,n}$
        Set $CurrID_{x,y,n} = i$
        IF $x, y \in$ the robot's current field of view
            THEN Set $Fov_{x,y,n} = 1$
            ELSE Set $Fov_{x,y,n} = 0$
        ENDIF
    ENDFOR
ENDFOR
```

We now define how the changes in the Input State variable $St_{x,y,n}$ affect the $T0$ and Tl variables which consequently alter the Saliency Value $V_{x,y,n}$ in each active cell. We describe the latter alteration of the Saliency Value in the next part.

```
FOR each cell $C_{x,y,n}$ at time $t$
    IF it is active, $St_{x,y,n} == 1$
        THEN Set $Tl_{x,y,n} = t$
        IF it is a new object, $CurrID_{x,y,n} \neq LastID_{x,y,n}$
            THEN Set $T0_{x,y,n} = t + \sqrt{-2 * Rg^2_{x,y,n} * log(M_{init}/M_{max})}, M_{init} = 100, M_{max} = 200$
        ENDIF
    ENDIF
    IF has not been updated for some time, $t - Tl_{x,y,n} > 10$ msec
        THEN Set $Rd_{x,y,n} = 0.2$
    ENDIF
ENDIF
```

We now describe how the Saliency Value $V_{x,y,n}$ is updated at each time step based on the rest of the variables stored in each cell $C_{x,y,n}$. Figure 4-4 illustrates how the Saliency Value changes over time for varying values of Saliency Growth Rate ($Rg_{x,y,n}$) and Decay Rate ($Rd_{x,y,n}$). The Saliency Value initially increases using the Growth Rate until it reaches a peak value and starts decreasing using the Decay Rate.

```
FOR each active cell $C_{x,y,n}$, $St_{x,y,n} == 1$at time $t$
    IF $P_{x,y,n} \neq$ face OR $Fov_{x,y,n} == 1$
        THEN Set $t_{peak} = T0 + \sqrt{-2R_g^2 log(M_{init}/M_{max})}$
        IF $t < t_{peak}$
            THEN $V_{x,y,n} = M_{max} * \exp -((t - T0)^2)/(2 * Rg_{x,y,n}), M_max = 200$
        ELSE $V_{x,y,n} = M_{max} * \exp -((t - T0)^2)/(2 * Rd_{x,y,n})$
        ENDIF
    ENDIF
ENDFOR
```

Note that if $Fov_{x,y,n} \neq 1$ for face inputs, i.e. the face is located outside the robot's field of view, the saliency value does not change to provide short-term spatial memory.

We now combine the Saliency Value $V_{x,y,n}$ from all active cells to produce a saliency map S.

At each location x, y,

$$S_{x,y} = \sum_{n=0}^{N} V_{x,y,n} \tag{4.1}$$

Lastly, we describe how the resulting saliency map S is used to produce the final attention output O of the robot's next target. O is a coordinate $x, y = argmaxS_{x,y}$.

4.4.2 Saliency Growth and Decay Rates

Initially, both $AdaptRg_{x,y,p}$ and $AdaptRd_{x,y,p}$ are set to 30 for all locations x, y, and feature types p. As the robot gains experience in the environment, the spatio-temporal learning system incrementally updates both $AdaptRg_{x,y,p}$ and $AdaptRd_{x,y,p}$.

Figure 4-4 illustrates the saliency function for varying values of saliency growth rate (R_g) and decay rate (R_d). The idea is that if a face or color segment is detected

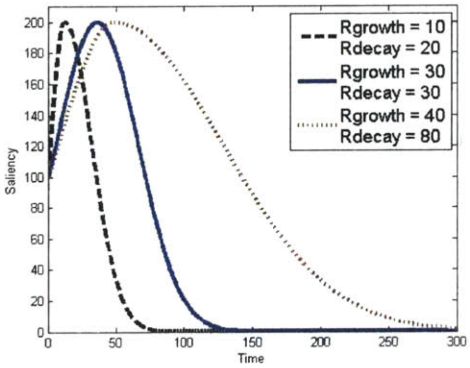


Figure 4-4: The attention's system's saliency function for varying growth and decay rates. The idea is that if a face or color segment is detected and subsequently tracked, its saliency value will initially grow and start decaying after a while. The saliency growth rate determines how good a particular stimuli is in capturing the robot's attention and the decay rate specifies how well it can maintain the robot's attention.

and subsequently tracked, its saliency value will initially grow and start decaying after a while. The saliency growth rate determines how good a particular stimuli is in capturing the robot's attention or vice versa. The decay rate specifies how well it can maintain the robot's attention. The time-varying saliency functions and interaction among these functions for multiple sensory events generate a number of advantages. Firstly, since each object has to be tracked for some time to achieve a higher saliency value, the system is more robust against short-lived false positive detection errors. It also deals better with false negative detection gaps. The combination of decay rates and egocentric map's short-term memory provides some short-term memory capabilities to allow the robot to remember objects even if they have moved outside the robot's field of view. Moreover, the emergent interaction among various saliency functions allows the attention system to integrate top-down and bottom-up control and also to naturally alternate among multiple learning targets. Lastly, the system architecture provides natural opportunities to detect various spatio-temporal and multi-modal correlation in the sensory data. The incremental adaptation of the saliency parameters based on these observed patterns allows the attention system to be more sensitive to a set of previously encountered learning target types and locations.

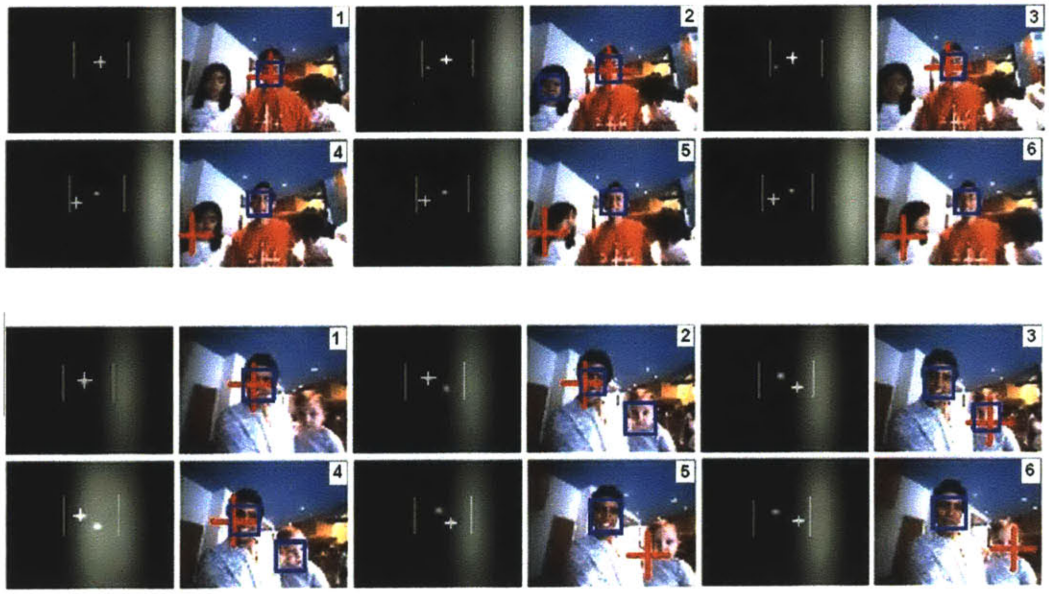


Figure 4-5: Two sample image sequences and the corresponding attentional map, illustrating the attention system's output while interacting with two people simultaneously. On each attention map (left column, the two vertical lines represent the robot's current field of view. Two people were interacting with the robot. The blue box superimposed on the image indicates detected faces. The red cross indicates the current attention target.

4.4.3 An Example

Figure 4-5 shows two sample sequences of the attentional map output. On each attention map (left column), the two vertical lines represent the robot's current field of view. Two people were interacting with the robot. The blue box superimposed on the image indicates detected faces. The red cross indicates the current attention target. Once a person's face is detected, it is represented by a white blob in the attentional map, with time-varying intensity level determined by the saliency function described above. Thus, the blob often remains in the map even if the face is no longer detected for some time, allowing the robot to still be aware of a person despite failure in detecting his or her face. In the upper sequence, the female's face was detected only in frame 2, but was still present in the map in frames 3 through 6. Similarly, in the lower sequence, the infant's face was detected in frames 2 through 4 and remains in the map for the rest of the frames. Moreover, as shown in both sequences, after attending to the first person, the attention system switches to the second person after some time due to the temporal interaction among each blob's saliency function. In both upper and lower sequences, this attention switch from the first person to the second person in frame 4 and 5 respectively.

4.5 Spatio-temporal Sensory Learning

The robot maintains a spatio-temporal learning map to correlate spatio-temporal patterns in the robot's sensory experience. Like the attention map, the system receives multi-modal events from the robot's perceptual systems (face, color segment, and sound). Note that the color segment corresponds to the color inside the detected face region. Additionally, it receives an input for each *response window* event, which is a fixed-duration period following each speech event produced by the robot. This input is used to detect when speech input occurs shortly after and thus is possibly a response to the robot's own speech. The spatiotemporal map's task is to detect spatio-temporal patterns of sensory input occurrence and correlate multi-modal inputs. The idea is that if a person is indeed present in front of the robot, concurrent presence of face, color, sound, and response are morely likely to happen. Thus, the spatiotemporal map can use this information to increase or lower confidence for the output of the perceptual system. Moreover, this information is also useful for biasing the attention system to favor certain regions where a person was just recently present or where people tend to appear.

4.5.1 Implementation Details

Figure 4-6 shows an illustration of the spatio-temporal learning map. This map is spatially equivalent to the egocentric attention map and also represents an approximated projection of the front half of the geodesic sphere centered at a robots origin , but at a lower resolution. It consists of a 2D rectangle with 70x52 pixels.

Each pixel spatially represents a 4x4 pixels corresponding region in the robot's attentional map. Each map pixel $P_{x,y}$ at location x, y in the spatio-temporal learning

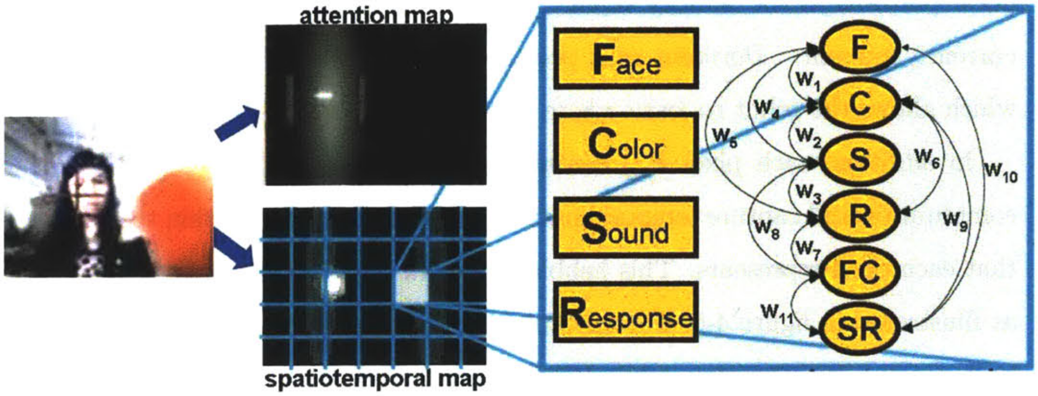


Figure 4-6: A diagram of the robot's spatio-temporal learning map. It consists of a 2D rectangle with 70x52 pixels. Each pixel spatially represents a 4x4 pixels corresponding region in the robot's attentional map. Each map pixel is a storage space, containing four cells, one for each input type (face, color, sound, response), and a hebbian network.

map contains four cells, $C_{x,y,p}$, one for each input type (face, color, sound, response) and a Hebbian network $H_{x,y}$.

Each cell $C_{x,y,p}$ has a number of states depending on its activity level $A_{x,y,p}$, as follows.

- Initially, all cells are *empty* and $A_{x,y,p} = -1$.
- When a perceptual event of type P is present at time t and location L, a set of cells $C_{x,y,P}, x, y \in L$ become *active* and $A_{x,y,P}$ is set to an initial magnitude of $M = 200$.
- Over time, the magnitude of $A_{x,y,P}$ decays based on the function $A(t) = M \exp -D * (t - T_{start}$ $.3, T_{start} =$ the time of when the last perceptual event was entered into $C_{x,y,P}$. With this decay function, if a cell has not been activated for about 2 seconds, its activity level $A_{x,y,P}$ will decay to 0 and the cell becomes *dormant* until activated again. Figure 4-7 illustrates an example of the spatio-temporal learning system's activity function when a perceptual input is entered at time t=3,5,7,10,18,20,22,30 ms.

Using this simple mechanism, the map can be used to record various spatiotemporal patterns in the sensory input. *Active* cells represent sensory events that are currently present. *Dormant* cells provide spatial history of each perceptual type, which allows the robot to learn where faces typically occur, etc.

In addition, each pixel $P_{x,y}$ contains a Hebbian network $H_{x,y}$ to perform local computation and capture temporal pattern of events occurring within the small region that each pixel represents. This hebbian network contains six nodes and six weights as illustrated in figure 4-6. For example, W_4 is strengthed when face and sound are both present in this region, while W_{10} is strengthened when both face and sound are present in this region shortly after the robot speaks.

The following are the processing steps of the hebbian learning process:

```
FOR each pixel $P_{x,y}$
    IF $C_{x,y,P}$ is *active*, i.e. $A_{x,y,P} > 0$
        THEN Activate the corresponding Hebbian node $m$ for feature type $P$, by setting the node's input $I_m = 1$
    ENDIF
    FOR each Hebbian weight $W_{ij}$, which connects node $i$ to node $j$
        Update each weight $dW_{ij} = Y0 * I_j * (I_i - (I_j * W_{ij})), Y0 = 0.01, I_i = inputofnodei$
    ENDFOR
ENDFOR
```

When combined together, these simple local cells provide spatio-temporal information about where and when things tend to co-occur. Figure 4-8 shows some examples of two-dimensional maps constructed by the spatially combined Hebbian weights from each cell. The spatio-temporal learning system currently utilizes these Hebbian maps in two ways. Firstly, it relies on W_{11} in making decisions on when to correlate pairs of face and voice samples. In particular, when it detects co-occurring face and voice

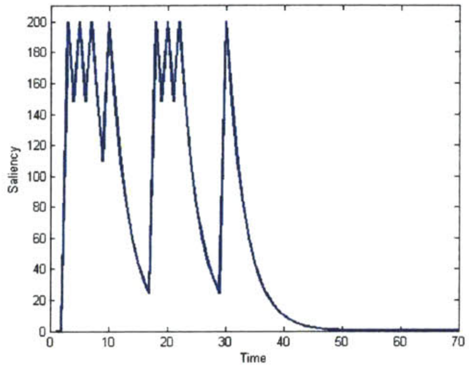


Figure 4-7: An example of the spatio-temporal learning system's activity function when an object is entered a t time t=3,5,7,10,18,20,22,30 ms.

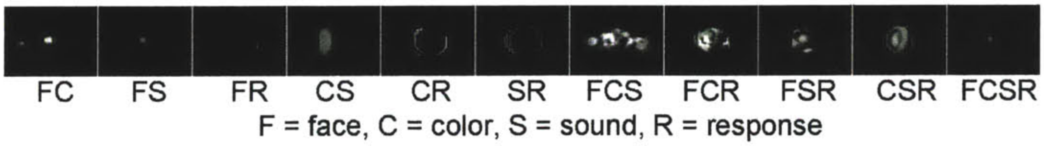


Figure 4-8: Some samples of two-dimensional maps constructed by the eleven hebbian weights(W_1-W_{11}) from left to right.

in a region, it will only correlate and store the pair if the value of W_{11} within the corresponding region is larger than a predefined threshold. In other words, it will only correlate the pair if co-occurrence of all input types has been high within this region. Secondly, it also uses the W_{11} weight map to update the $AdaptRg_{x,y,p}$ and $AdaptRd_{x,y,p}$ parameters in the attention system. Similarly, if co-occurrence of all input types has been high within a certain region, this will bias the attention system to favor this region over others.

Each cell in the map also performs a simple histogram calculation for the sound energy values occurring in two cases: when a face is present and not present. These histograms are then used to adaptively set the threshold for the sound detection module. This adaptation step is necessary since we deal with a large variations of background noise in different environments. Adaptive sound energy threshold allows the robot to disregard background noise when attending to people, simply by modelling sound energy values when faces are present.

4.6 Behavior-Based Control Architecture

We have so far covered the robot's physical actuators and sensors, low level motor controller, and perceptual systems. The behavior system integrates these lower level components into overall coherent actions and behaviors that are relevant to the robot's current perceptual inputs. The overall goal of the control architecture is to control the robot's high level behavior such that the robot is able to determine potential learning targets in the world while engaging in social interaction with human mentors. This has to be done in a timely manner as a human's social behavior is a very complex mechanism and humans are very well tuned to expect a certain degree of expressiveness and responsiveness. At any given time, the high level controller must in real time assess the current multi-modal perceptual state for presence of human social cues and potential learning targets. Simultaneously, it must also determine the most relevant behavioral response for the robot.

Figure 4-9 illustrates MERTZ's final behavior-based controller for finding, interacting with, and learning to recognize people. The robot's behavior system was implemented in L/MARS [19]. The controller has a number of behavior modules which communicate with each other using inhibiting and suppressing signals, as defined in the subsumption architecture [18].

Down Time At the lowest level, module *random-explore* simply generates random motion commands to module *explore* which sends the commands to the robot's eyes. This allows the robot to randomly explore its environment when noone is around, which is likely to happen in a long-term experiment.

Attending to Target The *attend* module consists of the multi-modal attention system described in section 4.4. It receives input from both visual and audio processing, which provide detection and tracking of faces, color segments, motion, and sound. Whenever the attention system decides that a potential learning target is present, it sends the target coordinates to the robot's eyes by inhibiting the output of module *explore*. The eyes are actuated to simply minimize the error between the

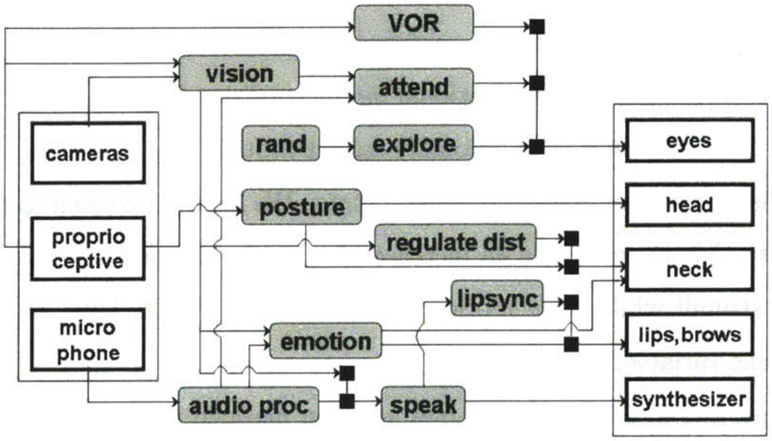


Figure 4-9: MERTZ's behavior-based controller for finding, interacting with, and learning to recognize people. The robot orients to and track salient visual targets, receives audio input, and tries to mimic what people say. The behavior system was implemented in MARS, a programming language specifically designed to implement behavior based programs in the incremental and concurrent spirit of the subsumption architecture.

target location and the center of the robot's field of view. This feedback control is done using only one of the robot's eyes, i.e., the right one.

Natural Human-Like Motion Module *VOR* monitors the eyes and head velocity and at times sends commands to the robot's eyes to compensate for the robot's moving head, inhibiting module *explore*. In parallel, module *posture* monitors the position of the robot's eyes and generates postural commands to the head and neck such that the robot is always facing the target. Lastly, module *lipsynch* receives phoneme sequences that the robot is currently uttering and commands the robot's lips to move accordingly.

Emotional Model Module *emotion* contains the robot's emotional model, implemented based on various past approaches for computational modelling of emotion [97, 12, 72]. For Mertz, the emotional model serves as an important element of the social interface. Research in believable agents suggests that the ability to generate appropriately timed and clearly expressed emotions is crucial to the agent's believable quality [57, 4]. Mertz's emotional model consists of two parameters: arousal and valence. In the absence of emotionally salient input, these variables simply decreases or increases over time toward a neutral state. We have predefined faces, motion, and proximate objects to trigger an increase of arousal. Medium-size faces are defined as positive stimuli while large motion and proximate objects have negative affects on the valence variable.

The arousal and valence variables are then mapped to a small set of facial expression, formed by the four degrees of freedom on the robot's eyebrows and lips.

Inviting and Regulating Interaction Module *maintain-distance* at times inhibits module *posture* to move the robot's neck to maintain a comfortable distance with a target, estimated using the proximate object detection and the relative changes in salient target's size.

Module *speak* receives input from module *vision* and *audio proc.* When the word recognition system successfully segments two words from the speech input, it sends a command to *speak* to either mimick these words. When the word recognition system is overwhelmed by a long speech input, it commands *speak* to produce a randomly selected sentence from a predefined set to request for people to speak in fewer words. The visual system also sends a predefined sentence to *speak* when it encounters a number of situations. Table 4.1 lists these situations and the corresponding predefined set of sentences.

4.7 Automatic Data Storage and Processing

The robot collects and stores each face image as segmented by the same-person tracker, except for those whose area is less than 2500 pixels. These face images are organized as sequences. Each sequence contains the result of a continuous tracking session and is therefore assumed to contain face images of one individual. The robot

Table 4.1: A set of predefined sentence for regulating interaction with people.

	Condition	Predefined Sentences
1	Segmented more than five words	Please say one or two words.
		I don't understand.
		Are you speaking to me?
		Too many words
		What did you say?
		Can you repeat it please?
2	Segmented face area is less	Please come closer.
	than 2500 pixels	I cannot see you.
		You are too far away.
3	The last 3 tracked sequences	Please do not move too much.
	and contain less than 20 images	Please look at my face.
		Please face me directly.
4	The spatiotemporal system	Please speak to me.
	detects a face region	Teach me some words, please.
	but no sound input	Please teach me some colors.
		Please teach me some numbers.

then eliminates all face sequences which contain less than 8 images and performs a histogram equalization procedure on all remaining sequences. All face sequences are taken as automatically produced by the robot without any manual processing or filtering.

The robot assigns a unique index for each sequence and keeps track of the last index at the end of each day. Loss of data and overwriting because of the common programming of indices to start at zero upon startup are one of the many mundane failures we simply overlooked during the project.

As mentioned above, the robot's spatio-temporal learning system utilizes the simple hebbian network within each local cell to make decision in correlating co-occurring face and sound samples. The robot automatically stores these pair correlations, retrieves the relevant sound samples, and places them along with the correllated face sequence.

The robot then automatically processes all of the final set of face sequences and correlated voice samples to extract various features for further recognition purposes.

One computer is assigned solely for this processing so that the robot can run these computationally extensive programs online without interfering with the real-time online behavior. However, at the end of each experiment day, the operator has to pause this data processing to move the robot back into the laboratory. The operator then manually resumes the processing after the move.

4.8 Experimental Results

In this experiment, we evaluate the robot's perceptual, attention, and spatio-temporal learning systems, with respect to the target goal of collecting face and voice data from each individual through spontaneous interaction. The most important task here is to collect face sequences from each person. The more images there are in each sequence and the larger each image is , the better it is in capturing visual information of each person. We also report on these collected training data which is then used by the robot's incremental face recognition system, as described in the next chapter. We analyze the accuracy and other relevant characteristics of the face sequences collected from each person.

4.8.1 Setup

We conducted the experiment in eight days. The robot was placed in x different locations (where?) for 2-7 hours each day. The exact schedule is shown in table 4.2. Like the earlier experiments, the robot was set up and people were free to approach the robot whenever they want. A written poster and sign was placed on the robot platform to introduce the robot and explain the experiment (see ??).

Throughout this project, we have seen many interesting aspects and issues associated with carrying out experiments in public spaces. In addition to the valuable lessons that we presented in section 3.5, it is a great opportunity for the robot to engage in natural and spontaneous interaction with a large number of naive passersby and also collect a huge amount of data. However, the down side is there is no guarantee of repeatability in the individuals that the robot encountered. This would severely

Table 4.2: Experiment Schedule

Experiment	Date	Time
1	Nov 20	1-7 pm
2	Nov 21	4-7 pm
3	Nov 22	3-6 pm
4	Nov 27	1-5 pm
5	Nov 29	1-7 pm
6	Nov 30	12-7 pm
7	Dec 1	2-7 pm
8	Dec 4	2-4 pm

limit our evaluation of the robot's incremental individual recognition capabilities. We thus decided on a compromise, where we recruited fourteen voluntary subjects and requested that they come to interact with the robot on multiple days throughout the experiment. In order to minimize control, we did not impose any rules or restriction on these subjects. We simply announced where and when the robot would be running on each day of the experiment. Unfortunately, due to the lack of control and instructions, most of the voluntary subjects came to interact with the robot once and only for a very short time. The detailed recruitment protocol and written instructions provided to the experiment subjects are attached in section ??.

As mentioned above, we were not able to record the experiment externally due to some limitations involving human subject experiments. This would require a more stringent protocol, such as written permission forms, which would alter the nature of the experiment in some undesirable ways. Thus, we can only provide quantitative data based on the robot's camera and microphone input. The camera's limited field of view unfortunately severely constrains the range of events that we can capture. For example, we cannot report on the number of people who approached the robot but failed to attract the robot's attention. Thus, we complement these data whenever approriate with some qualitative results through visual observation of the experiment. Though subjective, we believe that these qualitative observations yield some interesting insights uncaptured by the cameras.

4.8.2 Finding and Interacting With People

How many people did the robot find? Table 4.3 illustrates the number of face sequences produced by the robot's same-person tracking module during each experiment day. Each face sequence contains a set of face images produced by a continuous tracking session of one person. The robot collected a total of 4250 sequences and 175168 images during the entire experiment.

Figure 4-10 shows some of these sample face sequences. Each sequence contains a set of face images, which are assumed to belong to the same person. As we can see in this figure, this is of course not always the case. Some sequences contain background, badly segmented faces, and in a few cases even faces of another person. Thus, we need to further analyze these face sequences for detection and segmentation accuracy.

For this purpose, we manually labelled each sequence produced on day 6, the longest experiment day. The 863 sequences produced on day 6 come from at least 214 people. From these 863 sequences, we could not label 58 sequences due to background inclusion, segmentation error, and bad lighting. Thus, for day 6, 93% of the collected sequences contain correctly detected face images. Some statistics on the number of sequences and images that belong to each individual is shown in table 4.4 and figure 4-11.

Based on these numbers, we can infer that the robot indeed detected and tracked a large number of people. Most people generate less than 200 images. A number of people interacted with the robot much longer and thus generated up to 3177 images. These collected face sequences are later used as training data for the robot's incremental individual recognition system, except for some that the robot excluded automatically. We will further analyze these face training data for detection and segmentation accuracy in more details in the next section. Manual labelling of these training data indicates that the robot found and tracked at least 525 people during the entire experiment.

How long was the interaction and how many people at a time? Figure 4-12 shows the number of people that the robot interacted during the seven hours on

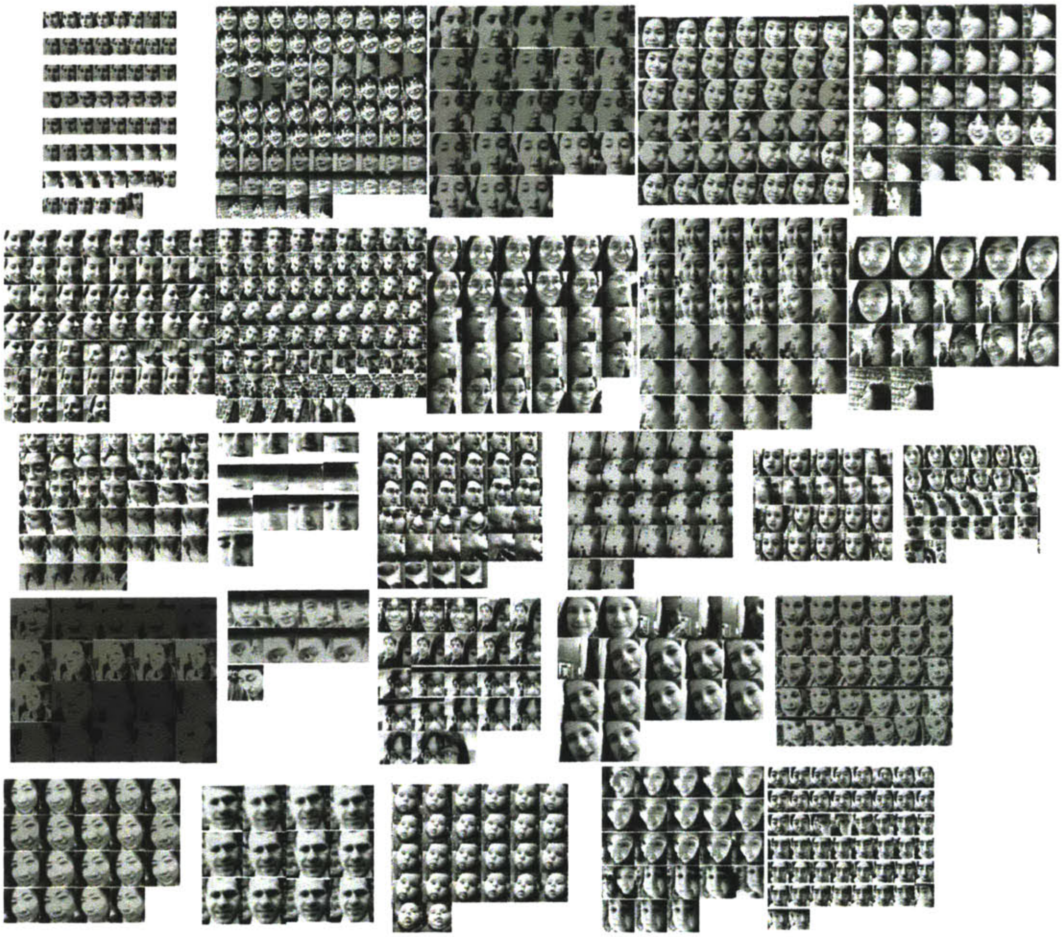


Figure 4-10: Some sample face sequences produced by the same-person tracking module. Each sequence contains a set of face images, which are assumed to belong to the same person. As we can see in this figure, this is of course not always the case. Some sequences contain background, occluded or badly segmented faces, and in a few cases even faces of another person.

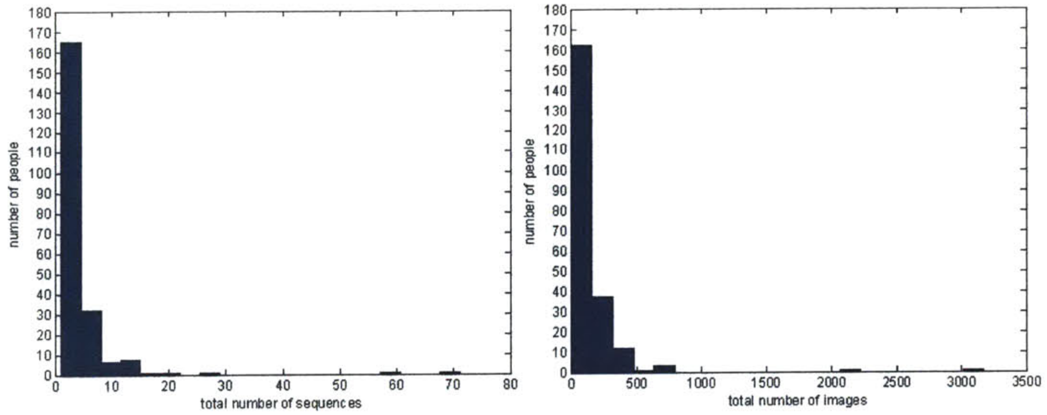


Figure 4-11: The distribution of the number of sequences and images from 214 people on day 6. On the left is the histogram of the number of sequences and on the right is the histogram of the number of images.

Table 4.3: The Face Tracking Sequence Output

Experiment	# face seq	# images	# image/seq:	min	max	ave	std
1	941	31667		1	534	33.7	48.1
2	268	12164		1	421	45.4	59.2
3	113	10023		1	926	88.7	138.9
4	459	22289		1	521	48.6	65.7
5	832	27467		1	458	33.0	48.1
6	863	27824		1	509	32.2	53.0
7	633	39622		1	487	62.6	77.6
8	141	4112		1	305	29.2	47.2
Total	4250	175168					

Table 4.4: The Same-Person Face Tracking Output on Day 6

Experiment	# face seq	# people	# seq/person:	min	max	ave	std
6	863	214		1	71	4.01	6.9
			# image/person:	min	max	ave	std
				1	3177	128.2	283.6

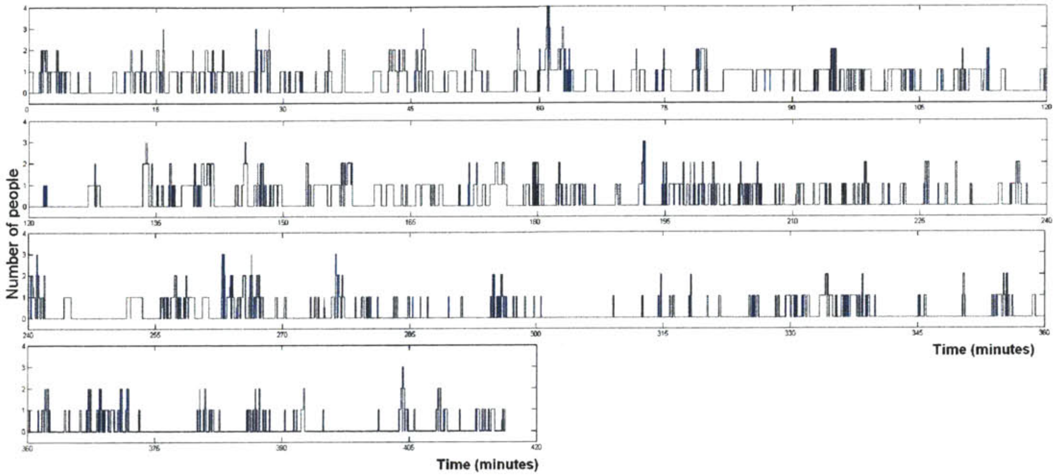


Figure 4-12: The number of people that the robot interacted with during the seven hours on day 6. This plot was calculated based on the manual labels of each sequence and the assumption that each person was still present for 15 seconds after the end of their face sequence.

day 6, according to the recorded face sequences and their timing information. This plot was calculated based on the manual labels of each sequence and the assumption that each person was still present for 20 seconds after the end of their face sequence. When actively interacting, robot interacted with more than one person concurrently for roughly 16% of the time. Table 4.5 shows the detailed breakdown of the duration of interaction segments with different numbers of people. A continuous interaction segment is defined is by the presence of a tracked face at every 10 ms interval. The robot interacted with at least one person for 32% of the seven hours. The longest duration of a continuous session with one, two, three, and four people are 510.5, 211.3, 259, 81, and 35 seconds respectively. The longest duration of down time is 510.5 seconds.

We also calculated the duration of continuous interaction sessions for each of the 214 people based on the manual labels and the timing information of each recorded face sequence. Figure 4-13 left shows the histogram of the sum of continuous session duration for these 214 people. From 214 people, 97% have a total duration of less than 119 seconds. Three people have a total duration of between 119-237 seconds. The last

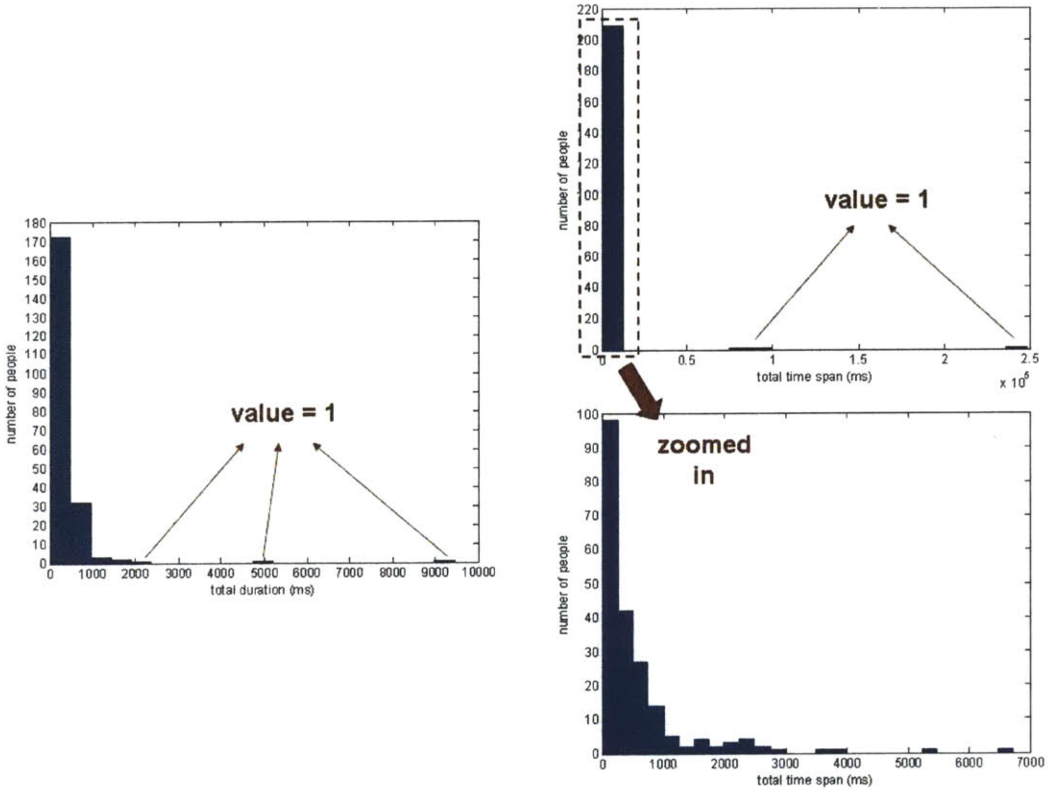


Figure 4-13: A set of histograms illustrating the distribution of session durations for the 214 people who were identified through manual labelling of all face sequences collected on day 6. Some bins of value 1 are annotated because of lack of visibility. On the left is a histogram of the sum of continuous session duration for all 214 people. On the right is the histogram of the total time span of appearance, measured from the time of first and last recorded sequence. The lower histogram is a zoomed in version of the first bin in the upper histogram.

Table 4.5: Duration of interaction sessions for different number of people on day 6

Num of people	% of 7 hours	# segments	max duration (ms)	ave duration (ms)
0	67.95%	504	5105	336.7
1	26.84%	464	2113	144.5
2	4.84%	151	259	80
3	0.34%	23	81	37.1
4	0.03%	3	35	24.3

two people have a total duration of 532 and 886 seconds, respectively. Figure 4-13 right shows the histogram of the total time span of appearance, measured from the time of first and last recorded sequence. This gives us some information for cases of individuals who interacted with the robot on multiple occasions throughout the day. From 214 people, 97.7% appeared in the total time span of less than 1244 seconds. The lower right histogram shows a more detailed breakdown for this bin. The other 3 people have a larger total time span, ranging from 8081-24240 seconds.

From these numbers, we can infer that the robot is in action and interacting with people for roughly one third of the time. About 16% of these active moments, the robot interacted with multiple people concurrently. Moreover, most people stopped by once during the day and interacted briefly with the robot. A few people stayed for a longer session with the robot. A few people also stopped by in multiple occasions during the day.

Did people verbally interact to the robot? As described in section 4.3.4, the word recognition system uses a fixed energy threshold to detect the onset of sound events. Now the threshold is set low enough such that most sound events, including background noise, are recorded for evaluation purposes. Table 4.6 shows the number of recorded samples and the number of sound samples that pass the robot's word recognition filtering mechanism described in section 4.3.4. Note that sound data recorded on day 1 were lost due to human error. The first does not contain much information as it includes almost every sound event. The latter gives a better estimate

Table 4.6: The Voice Samples Output

Experiment	# total samples	# processed
2	1641	915
3	560	323
4	1032	834
5	3274	1041
6	5148	1520
7	3492	892
8	882	282
Total	16029	5807

of the number of actual speech input that the robot received.

We manually annotated 346 sound samples which have passed the word recognition filtering system. From the 346 samples, 26 samples contain only background noise and 27 samples contain only the robot's own voice. Thus, roughly 86% of the samples contains human speech. In an earlier experiment described in section 3.5.2, we annotated all of the recorded sound files more extensively. We also differentiated between robot directed speech and non-robot directed speech based on the speech content. These annotations indicate that the robot indeed received a large amount of human speech input during the experiment.

How well did the robot correlate face and voice data? In order to complement the unsupervised face recognition system, the robot utilizes spatio-temporal context to correlate pairs of face and voice sequences from the same individual to allow for multimodal recognition. Table 4.7 illustrates the number of face and voice pairs produced by the robot's spatio-temporal correlation procedure during each experiment day. Note that a large part of the data from day 1 was unfortunately lost due to human error.

Each pair of face and voice samples contains one face sequence and one or more sound samples which were spatio-temporally correlated with the corresponding face sequence. The robot collected a total of 201 pairs during the entire experiment. We manually annotated each collected pair and report on the correlation and segmenta-

Table 4.7: The Paired Face and Voice Data

Experiment	# pairs	# voice samples	# face images
1	7	14	381
2	48	128	4944
3	17	37	3786
4	20	23	1717
5	47	52	5430
6	24	34	3457
7	32	49	4767
8	6	7	296

tion accuracy in table 4.8 and figure 4-14.

Figure 4-14 consists of three plots. The top plot shows the proportion of sound samples containing purely of background noise, robot's speech, and human speech. The middle plot shows the percentage of sound samples in each face-voice pair which correctly belong to an individual portrayed in the correlated face image. We cannot be absolutely sure that the voice and the face actually match up since we don't have a video recording of the entire experiment as ground truth. However, we at least matched up that the gender of the face and voice match up. Of these correct voice samples, we analyze the segmentation accuracy of the content. The proportion of inclusion of robot's voice and other people's voice inside these samples are shown in the lower plot. On the first four experiment days, the robot's auditory system did not inhibit its input when the robot is speaking. Thus, we can see that the proportion of robot's voice is very high during those four days.

Given the amount of noise and dynamic in the environment, this face and voice correlation task is very difficult for the robot. People tend to come in groups and they all speak at the same time. Some speak to the robot and others speak to each other in the background. These results show that the robot can perform this task with a reasonable accuracy, but it has to be very selective in its decision making, as determined by the robot's spatio-temporal learning system. Thus, out of the 5807 sound samples that the robot collected, only 344 samples were correlated with a face sequence.

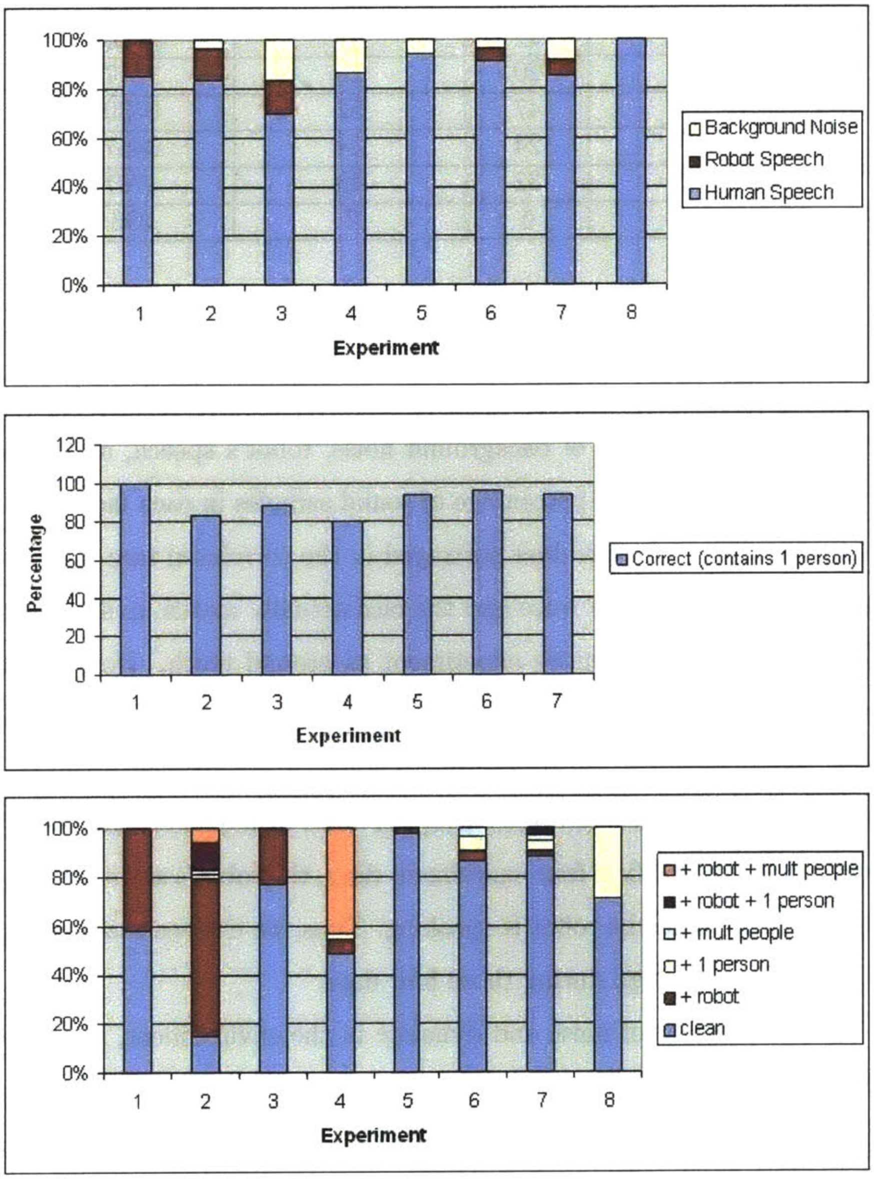


Figure 4-14: Segmentation and Correlation Accuracy of the Paired Face and Voice Data

Table 4.8: Segmentation and Correlation Accuracy of the Paired Face and Voice Data

Exp	speech	robot's	bgnd	correct	clean	+robot	+1p	+mp	+robot+1p	+robot+mp
1	14.3	0	85.7	100	58.3	4.2	0	0	0	0
2	13.3	3.1	83.6	83.3	15	64.5	1.9	1.9	11.2	5.6
3	13.5	16.2	70.3	88.2	76.9	23.1	0	0	0	0
4	0	13	87	80	85	10	5	0	0	75
5	0	5.8	94.2	93.8	98	2	0	0	0	0
6	5.9	2.9	91.2	95.8	87.1	3.2	6.5	3.2	0	0
7	6.1	8.2	85.7	93.8	88.1	2.4	4.8	2.4	2.4	0
8	0	0	100	100	71.4	0	28.6	0	0	0

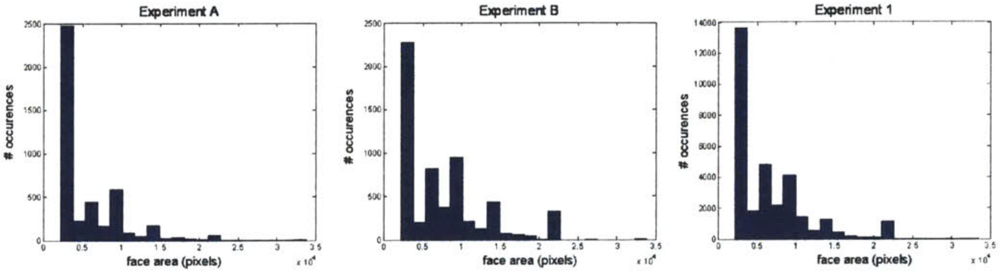


Figure 4-15: A set of histograms of the pixel area of faces collected during three experiment days. The first two were taken from an earlier experiment described in section 3.5. The third histogram was taken from day 1 of the final experiment. During these three experiment days, the robot did not make any verbal requests.

Was the robot able to influence the input data? Figure 4-15 and 4-16 shows a set of histograms of the pixel area of the faces collected on different experiment days. We will compare these histograms to show an increase of face sizes when the robot actively makes verbal requests for people to come closer when their face area is less than 2500 pixels.

Two histograms in the first figure, titled experiment A and B were taken from an earlier experiment described in section 3.5. The third histogram titled experiment 1 was taken from day 1 of the final experiment. During these three experiment days, the robot did not make any verbal requests. The second figure contains histograms

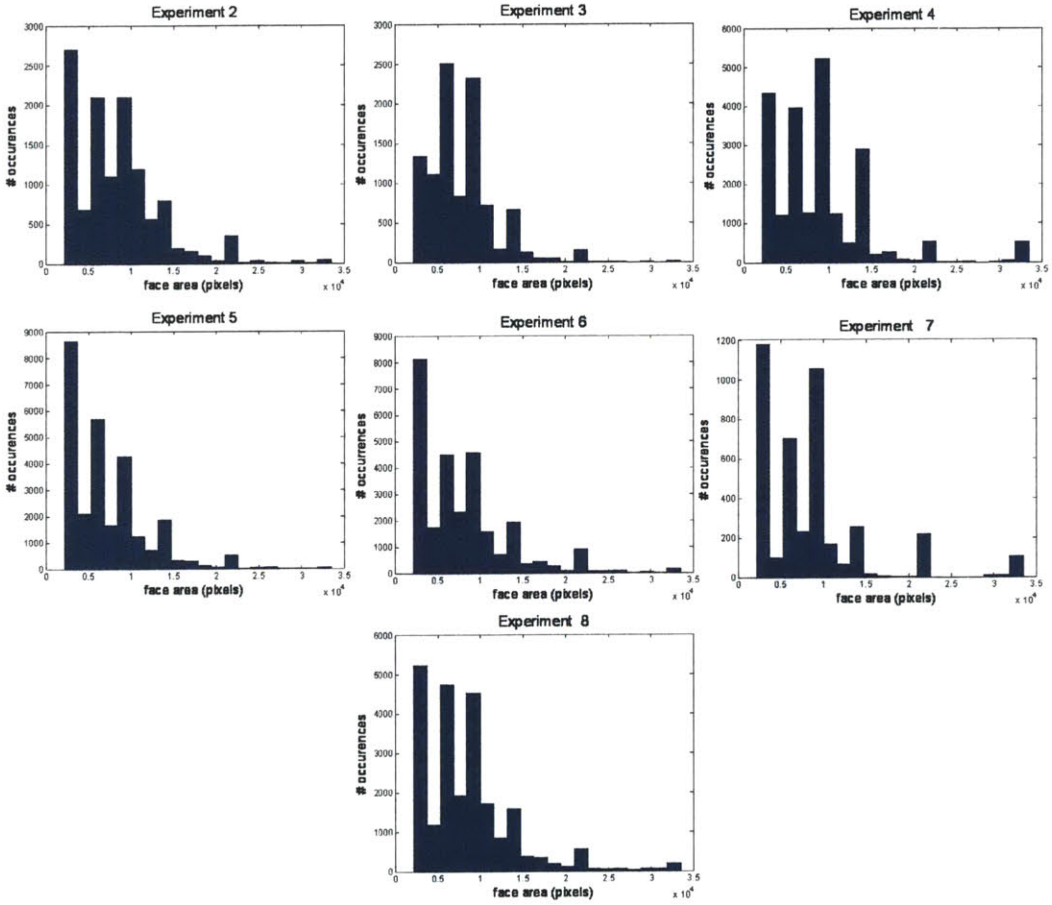


Figure 4-16: A set of histograms of the pixel area of faces collected during seven experiment days. These histograms were taken from day 2-8 of the final experiment, where the robot made verbal requests.

taken from day 2-8 of the final experiment, where the robot made verbal requests.

These results indicate that the occurence of face sizes within the range of 5000-10000 pixels indeed increases when the robot makes verbal requests for people to come closer.

What did the robot attend to? Figure 4-17-4-20 provides a few snapshots of the output of the attention system during the experiment. Each figure contains a time series of pairs of the robot's camera input image and the corresponding attention map output, ordered from left to right. Note that the attention map is an approximated projection of the front half of the geodesic sphere centered at a robots origin. As shown previously in figure 4-3, the camera input image occupies only a subregion of the attentional map. Due to the reduced resolution, the white lines which represent the camera input image borders in the attention map are not always visible. As mentioned above, white patches in the attention map represents various sensory inputs: face, sound, motion, or color segments. A white cross in the attentional map represents the current attention target, which is the region with highest saliency value.

The camera input image is sometimes superimposed with blue boxes, which represents the output of the face detector and tracker. The red cross in the camera input image represents the same current attention target (shown as a white cross in the attention map) when it happens to fall within the robot's field of view.

Figure 4-17 shows a snapshot of the attention output while the robot maintains a short-term memory for a person while interacting with another person. The robot detected two people in the second frame. It then continued to interact with one person, but maintained a short-term memory for the other person in the attentional map while he was outside the robot's field of view. After some time, the robot habituated to the first person and switched attenion to the second person.

Figure 4-18 shows a snapshot of the attention output while the robot switched attention from one person to another using sound cues. The robot interacted with a person while another person was speaking but not visible to the robot. After the robot habituated to the first person, it switched attention to the other person.

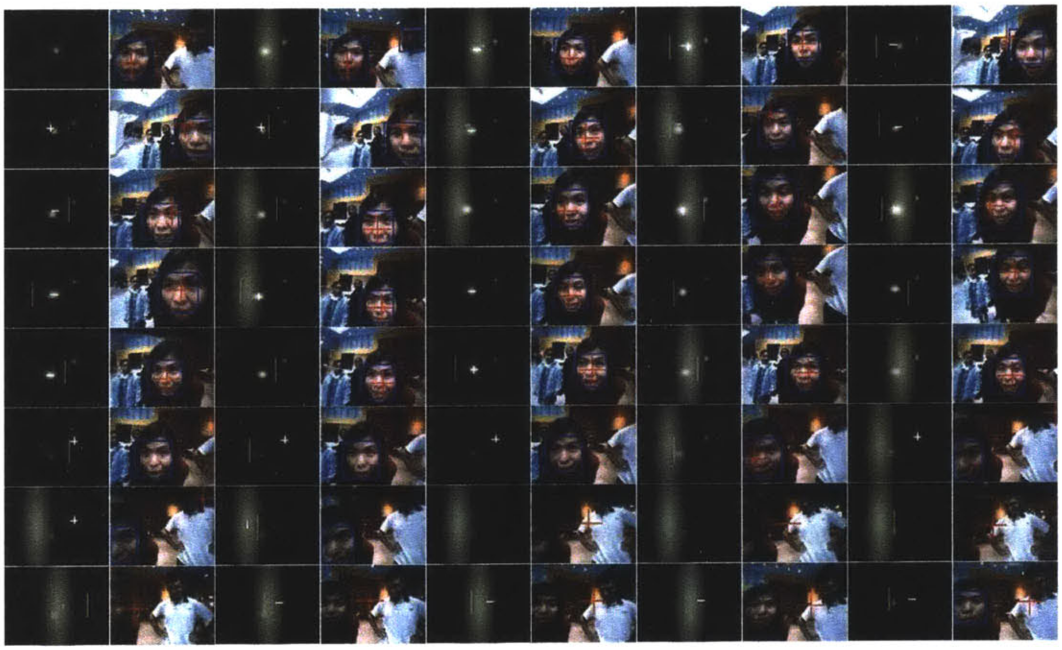


Figure 4-17: A snapshot of the attention output while the robot maintained a short-term memory for a person while interacting with another person.

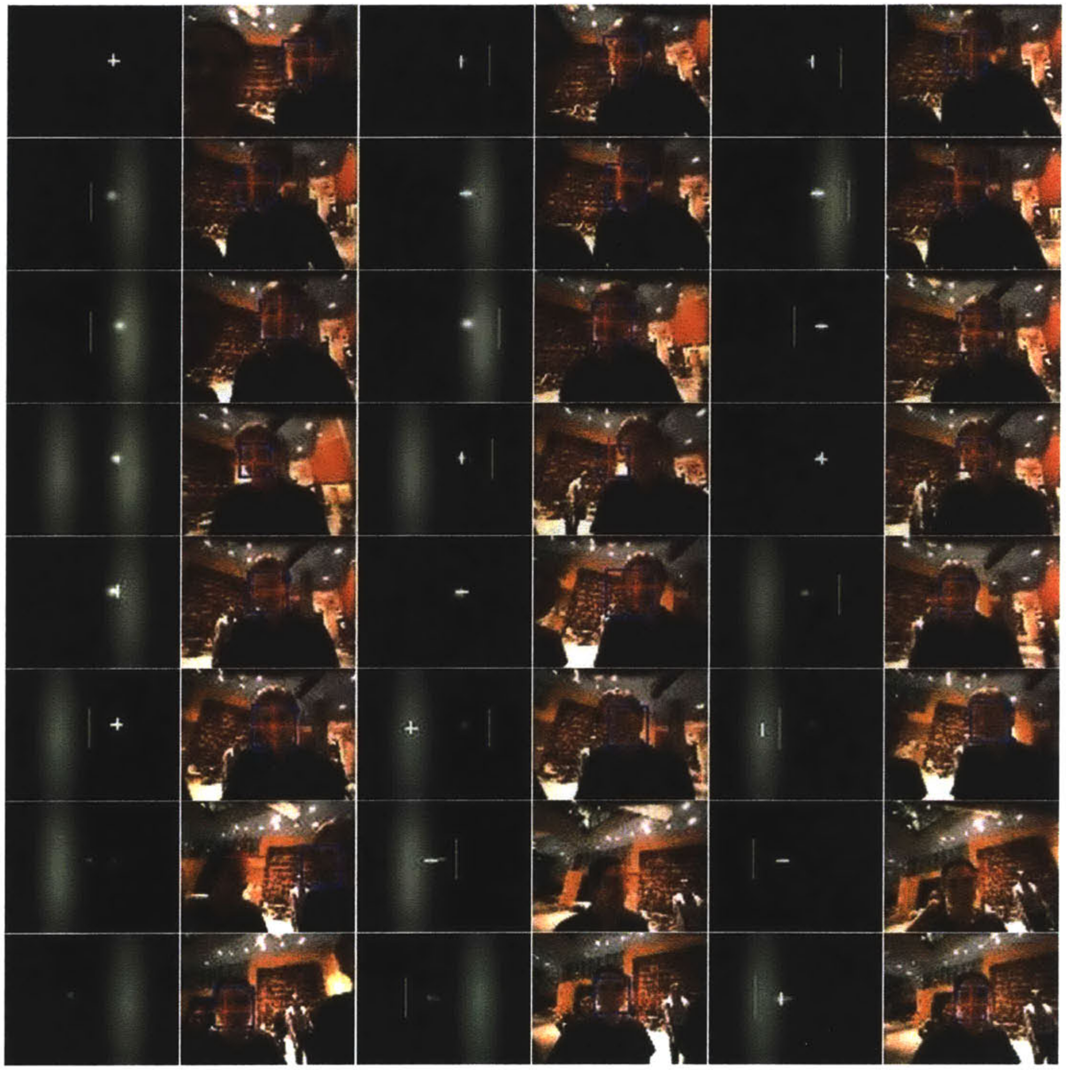


Figure 4-18: A snapshot of the attention output while the robot switched attention from one person to another using sound cues.

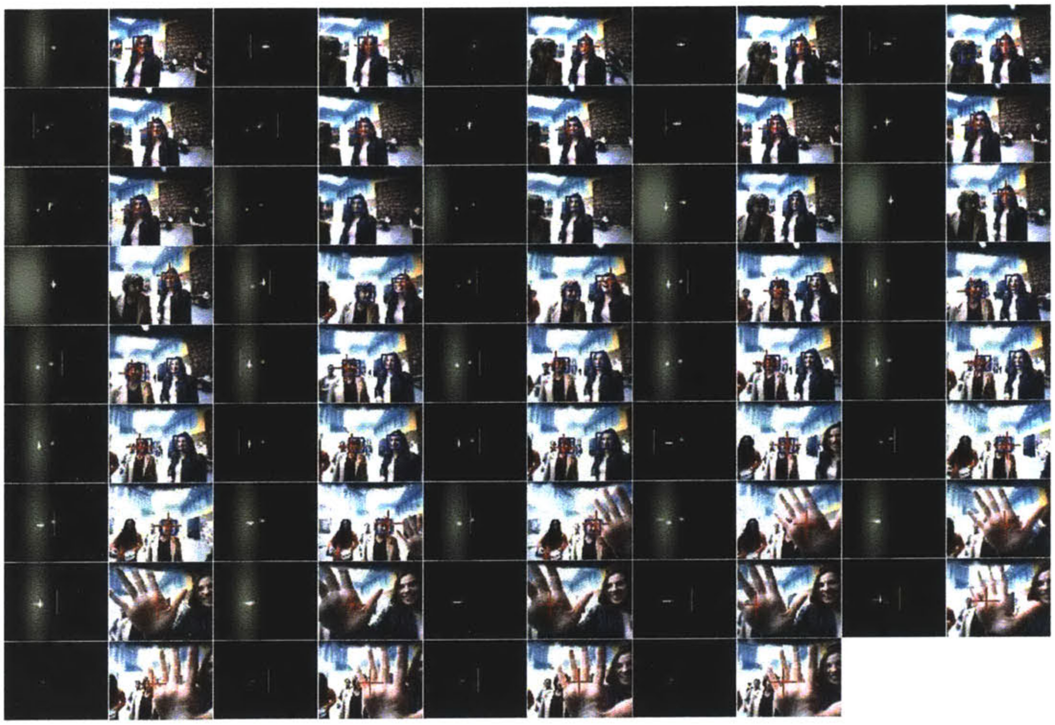


Figure 4-19: A snapshot of the attention output showing the robot interacting with two people simultaneously.

Figure 4-19 shows a snapshot of the attention output while the robot interacted with two people simultaneously. The robot first interacted with one of the two people in front of it. At some point, the robot detected both of them and switched attention to the other person after some time.

Figure 4-20 shows a snapshot of the attention output while the robot recover its tracking of a person because of the attention system's spatio-temporal persistence. The robot initially tracked a person, but was having difficulty because of false positive detection and motion in the background. At some point, the robot lost track of the person because she was too close to the robot. However, the attention system's spatio-temporal persistence allows the robot to maintain its target on the person until the face detector was able to find the person's face again.

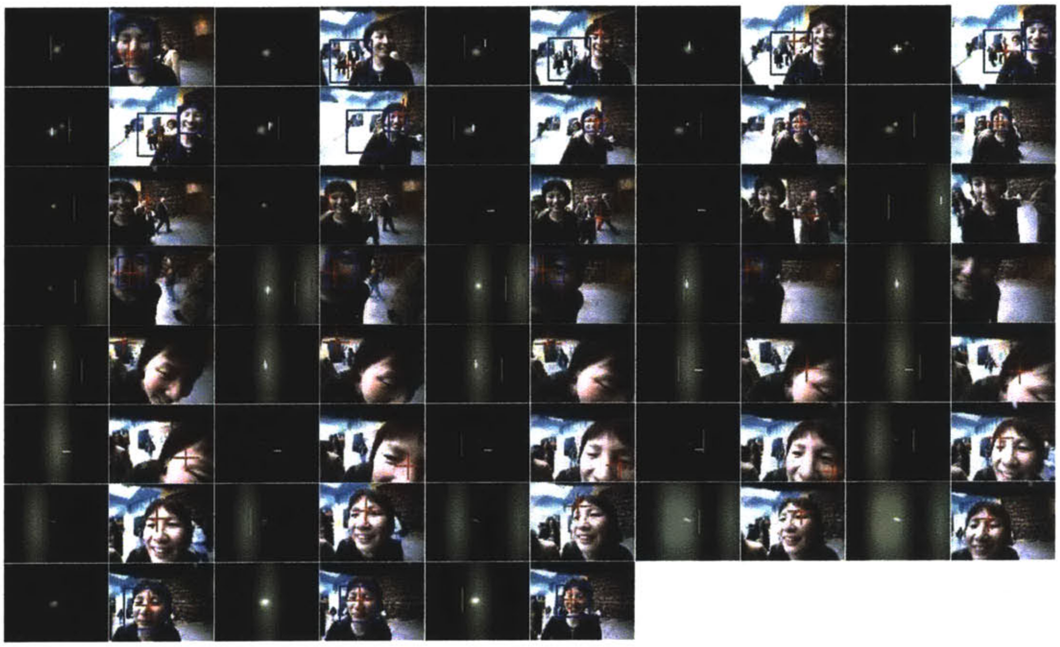


Figure 4-20: [A snapshot of the attention output showing the robot recover its tracking of a person because of the attention system's spatio-temporal persistence.

What did it learn? Figure 4-21 shows a sequence of maps consisting of the eleven hebbian weights $W_1 - W_11$ recorded on day 7 of the experiment, ordered from left to right. By the end of the experiment, the different hebbian maps provide patterns for when and where sensory inputs tend to occur. For example, the W_7 map provides the information that face, color segment, and sound tend to co-occur across the horizontal region spanning the front half of the robot, but spans a smaller vertical range. Thus, a false positive face in a ceiling is less likely to be a face. The W_8 map provides information that face, color segment, and response tend to co-occur around the middle region in front of the robot. This corresponds to the likely areas for people to verbally respond to the robot's speech.

As described above, the sound detection module also detects the onset of sound events using an adaptive energy threshold to minimize detection of and therefore allocation of attention to background noise. As shown previously in figure 3-14, variation of background noise level is particularly problematic when the robot is moved away from its familiar laboratory environment to a noisy public space. Figure 4-22 illustrates the adapted sound threshold values throughout day 6 and 7 of the experiment. The blue line corresponds to the sound energy threshold value over time. The superimposed red line corresponds to the number of faces present over the same time period. In the lower figure, the face occurrence is shown as a red dot because we do not have information to determine the exact number of faces. These plots show that the robot adaptively increases the sound energy threshold when people are present. Adapting the sound threshold according to the energy level of each interaction session protects the robot's attention system from being overwhelmed by the high level of background noise coming from all directions.

4.8.3 The Face Training Data

We have shown in the last section that the robot collected a large number of face sequences from many different individuals. The robot automatically constructs the training data set by compiling all of the collected face sequences and making some automatic exclusions, as described in section 4.7. Some samples of these face sequences

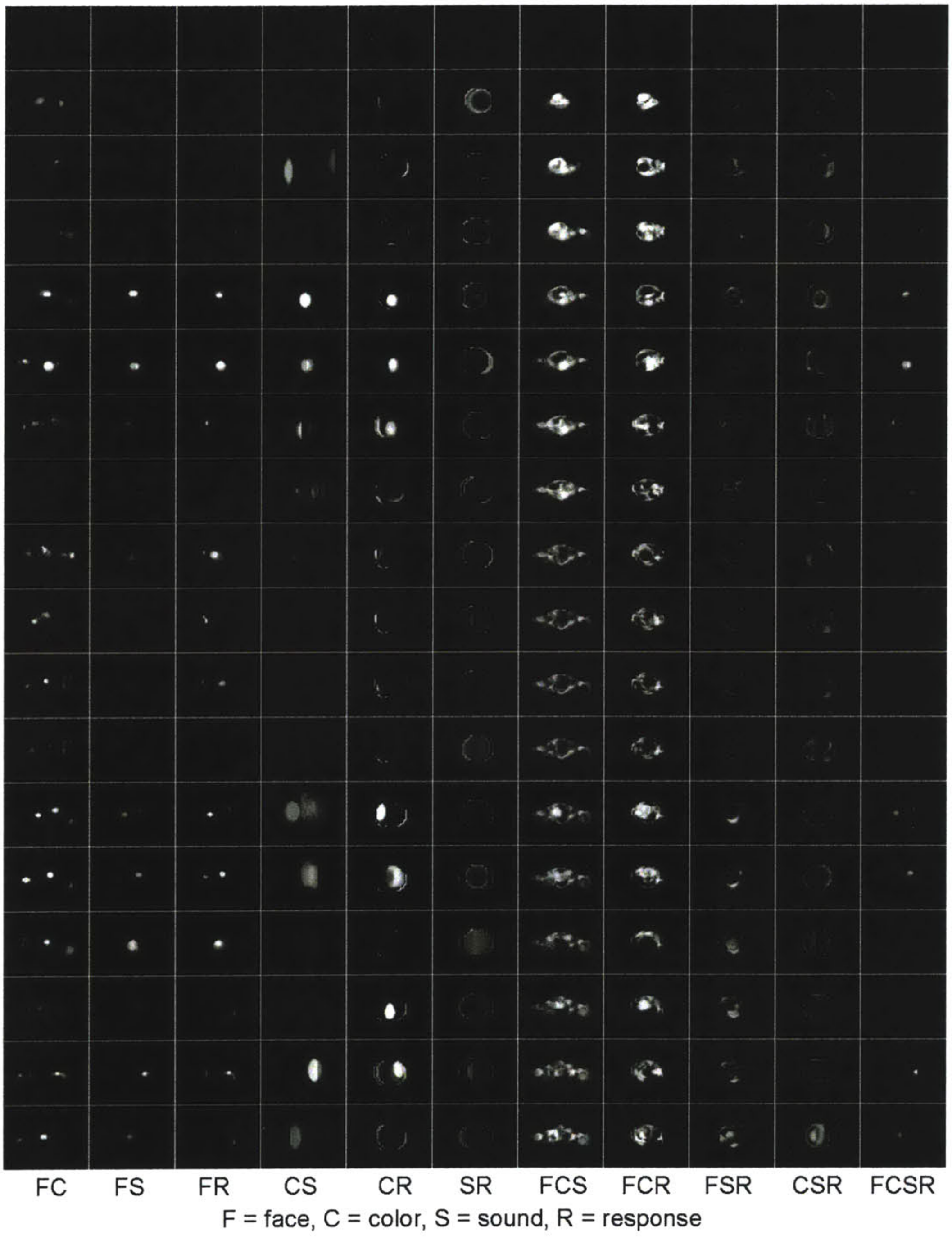


Figure 4-21: A sequence of maps consisting of the eleven hebbian weights W1 - W11 recorded on day 7 of the experiment, ordered from left to right

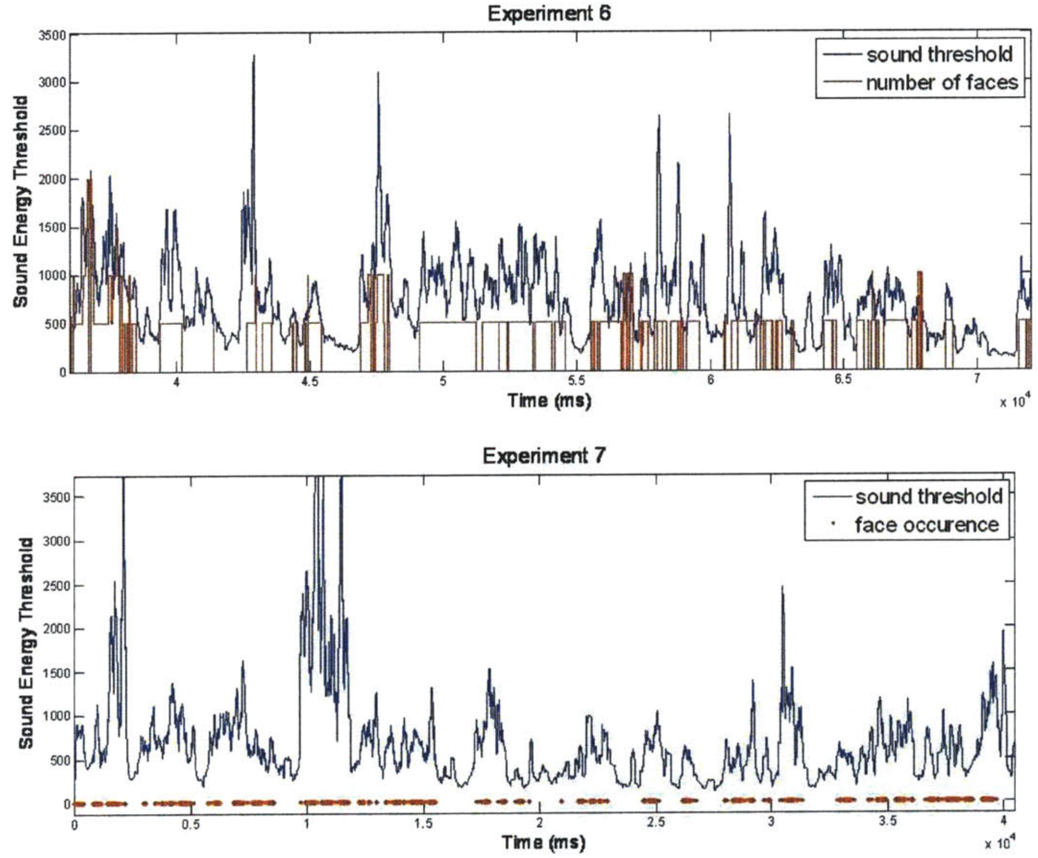


Figure 4-22: Adaptive sound energy threshold value over time on day 6 and 7. The blue line corresponds to the sound energy threshold value over time. The superimposed red line corresponds to the number of faces present over the same time period. In the lower figure, the face occurrence is shown as a red dot because we do not have information to determine the exact number of faces.

Table 4.9: The distribution of the number of images per sequence and the number of sequences and appearance days per individual

#images/seq	0-110	111-230	231-350	351-470	471-590	591-710	711-830	831-950
	1702	257	43	16	5	0	1	1
# seq/person	1-5	6-10	11-15	16-20	21-25	26-30	31	248
	448	41	11	4	2	1	1	
# days/person	1	2	3	4	5	6	7	8
	492	15	1	0	1	0	1	

are previously shown in figure 4-23.

After these automatic filtering steps, the robot's final set of face training data contains 2025 sequences. We manually labelled these sequences to obtain identification ground truth. If more than 50% of the sequence contains non-face images, the sequence is labelled as unidentified. Sequences which contain two people are labelled according to the individual who owns the majority of the face images. From these 2025 sequences, 289 are labelled as unidentified and the rest are identified as 510 individuals. Table 4.9 shows the distribution of the number of images per sequence, the number of sequences per individual, and the number of distinct appearance days of the few repeat visitors. The sequence length ranges from 9 to 926 images. The number of sequences per person is very uneven. The robot operator has the most number of sequences and repeat visits. A total of 19 people have more than 10 sequences in the database. Eighteen people interacted with the robot on more than one day during the experiment. Unfortunately, given the minimal control of our recruitment procedure, only a few volunteers actually came for multiple visits.

We also visually inspected 25% of the data set in three contiguous segments in the order of its recording time and analyzed the amount of errors and variations of the content. Figure 4-23 shows the proportion of segmentation error, occlusion, and pose variations in this annotated set. From these 567 face sequences, 8.6% consists entirely of non-face images, 28% contain at least one badly segmented face image and

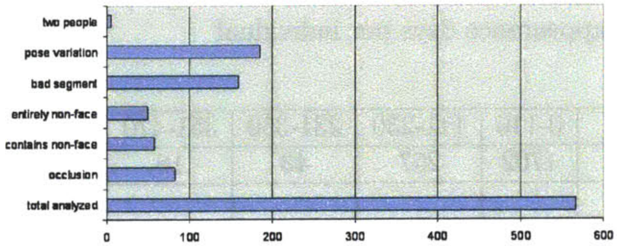


Figure 4-23: Analysis of errors and variations in the face sequence training data that the robot automatically collected during the experiment

10% contain at least one non-face or background image. These three error categories are mutually exclusive. A face image is considered to be badly segmented if less than two third of the face is included or it fills up less than half of the image. Roughly 0.9 % of the sequences contain face images of two people. No sequence in the database contains more than two people. In order to estimate the amount of noise in these sequences, we measured that 32.6% of the sequences contain large face pose variations of more than 30 degree in-plane or out-of-plane rotation. Lastly, about 14.5% of the sequences contain occluded face images. This occlusion is most frequently caused by the robot's eyelids which occasionally come down and obstruct the robot's field of view.

These results indicate that the robot collected a large face training data, containing a large number of images from many individuals. There is a lot of variation within each face sequence, which is valuable for encoding maximal information about a person's facial appearance, but challenging for recognition. As expected, there is a high level of noise in the data, induced by both tracking error and the dynamic setting the robot operates in.

Chapter 5

Unsupervised Incremental Face Recognition

In the last chapter, we have presented the implementation and experimental results of the robot's perceptual, attention, and behavior systems. The integration of these subsystems allows the robot to automatically instigate, segment and collect a large amount of face and voice data from people during natural spontaneous interaction in uncontrolled public environments. In this chapter, we describe how these data are processed by the robot's face recognition system to incrementally learn to recognize a set of familiar individuals.

An incremental and unsupervised face recognition framework essentially involves two main tasks. The first task is face clustering. Given that the system starts with an empty training set, it has to initially collect face data and form clusters of these unlabelled faces in an unsupervised way. The system then incrementally builds a labelled training set using these self-generated clusters. At this point, the system is ready for the second task of supervised face recognition based on the incrementally constructed labelled training set.

We begin by discussing a set of challenges and failure modes in realizing these tasks. We then present the implementation of the first task: unsupervised face clustering. We analyze the face clustering performance across a range of algorithmic and data-dependent parameters. We also applied the clustering algorithm on the

Honda-UCSD video face database [51, 50].

Lastly, we present an integrated system which incorporates the face clustering solution to implement an incremental and fully unsupervised face recognition system. We evaluate this integrated system using the robot's collected face sequences to analyze both the incremental recognition performance and the accuracy of the self-generated clusters.

5.1 Challenges

Automatic face recognition is a very useful but challenging task. While a lot of advances have been made, the supervised face recognition problem is still unsolved when posed with high variations in orientation, lighting, and other environmental conditions. The Face Recognition Vendor Tests (FRVT) 2002 report indicates that the current state of the art in face recognition using high-resolution images with reasonably controlled indoor lighting is 90% verification at a 1% false positive rate [75]. Recognition performance is reported to decrease as the size and time lapse of the database increase. The American Civil Liberties Union's review of face recognition surveillance tests at the Palm Beach airport describes that classification performance decreases sharply in the presence of motion, pose variations, and eyeglasses [92].

In our particular setting, embodied social interaction translates into a complex, noisy, and constantly changing environment. The system has to work directly from video stream input, instead of nicely cropped images. Both the robot's cameras and the interaction subjects are moving, leading to variations in viewing angle, distance, etc. The experiment lasts for many hours of the day in areas with large glass windows, resulting in lighting changes. Mertz typically encounters a large number of individuals each day and often has to interact with multiple people concurrently. The passersby exhibit a wide range of interaction patterns and expectations. While some people managed to face the robot frontally and allowed the robot to extract frontal face images, others were busy looking around to check out the robot's platform and thus rarely facing the robot frontally. Some people have trouble hearing the robot and thus

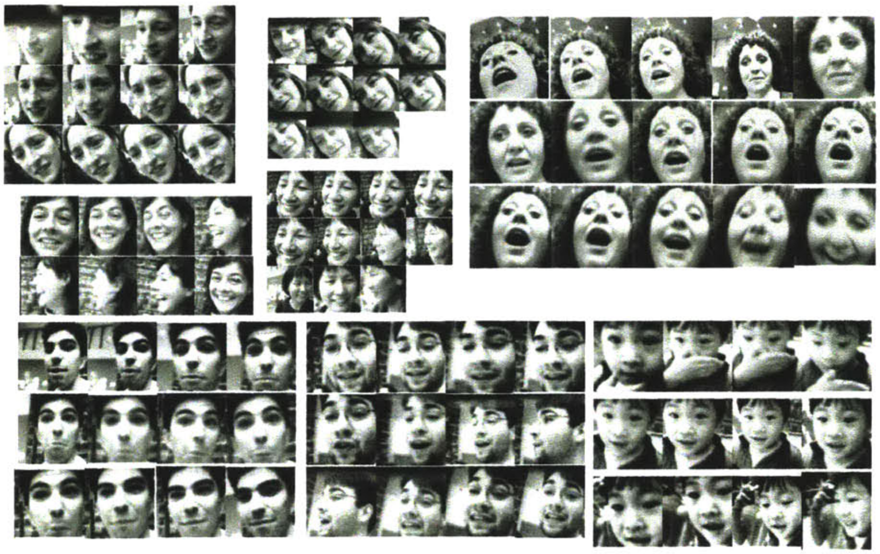


Figure 5-1: A sample of the face sequences taken from interaction sessions. Both the camera and target move around, creating a complex and noisy environment. Note the large variation in face pose, scale, and expression.

tilt their head sideways to listen to the robot, exposing only their side facial views. Figure 5-1 illustrates a sample set of faces tracked during interaction, containing large variations in scale, face pose, and facial expression. These dynamic and noisy settings lead to various errors and imprecision in the robot's collected training data, as described in section 4.8.3.

In addition to errors and imprecisions which increase the complexity of the problem, the task of extending the recognition system to allow for incremental acquisition and unsupervised labelling of training data further complicates the problem. Previous studies have shown that the intraclass distance of a person's face is sometimes larger than the interclass distance between two people's faces [90]. Thus, the task of clustering face images is challenging and cannot rely on existing distance-based clustering methods. Moreover, many existing clustering algorithms require an a priori number of clusters, which is not available in our case.

5.2 Failure Modes

We describe a set of failure modes relevant to the incremental recognition task. In our application and setup, some failures are more catastrophic than others. As the entire procedure is automated, failure in one subsytem can propagate to the rest of the system. Thus, it is important for the different subsystems to compensate for each other's failures.

The Face Data

- Partially cropped faces or inclusion of background due to inaccurate segmentation during tracking
- Inclusion of non-face images due to false positive face detection
- Inclusion of another person's face images due to to tracking error

The first two failures are certainly not ideal, but may still be compensated by the face clustering system. The last one, however, would lead to unacceptable results if they cause two people to be recognized as one person.

Unsupervised Clustering

- *Merging* – multiple individuals per cluster
- *Splitting* – multiple clusters per person
- Failure to form a cluster of an individual
- Inclusion of non-face images in some clusters in the database
- Formation of clusters containing of only non-face images

Merging and *splitting* are the two main failure modes of any clustering method. We will often refer to these failure modes by using these italicized keywords. *Merging* multiple individuals into a cluster can lead to the false recognition of two people as one person, especially if the proportion of each person in the cluster is roughly even.

This failure may not propagate to the recognition system however, if one person holds majority of the cluster. *Splitting* or forming multiple clusters per person is somewhat less catastrophic. In fact, we expect that this failure will frequently occur initially. As the robot interacts more with the person, we hope that these split clusters will start getting merged together. However, if the system starts building two large clusters of a person and never merges them together, this would lead to a negative consequence.

As the main goal is to recognize familiar individuals, the failure to form a cluster of a person is unacceptable of he/she has interacted with the robot frequently. However, failure to form a cluster of someone who has only shown up once or twice is perfectly acceptable.

Inclusion of non-face images in a part of or whole cluster in the database is inevitable, unless the robot has 100% face detection and tracking accuracy. This failure also may lead to negative consequences, the recognition system fails to compensate for them.

Person Recognition

- False recognition of a known person as an unknown
- False recognition of a known person as someone else

Both of these failure modes are unacceptable, given that the recognition system is our last point of contact in this thesis. In the future, we are interested in exploring an active learning scheme where the robot inquires a person to check if its recognition hypothesis is correct and somehow integrate the answers into its learning system. This would provide a way to compensate for these unacceptable recognition failures.

5.3 Unsupervised Face Clustering

Given an unlabelled training set consisting of an arbitrary number of face sequences from multiple individuals, the task of the clustering system is to cluster these sequences into classes such that each class corresponds to one individual. Figure 5-2 illustrates the steps of the face sequence clustering procedure. We first describe the overall approach that we took in our solutions. We then present each implementation step in more details below.

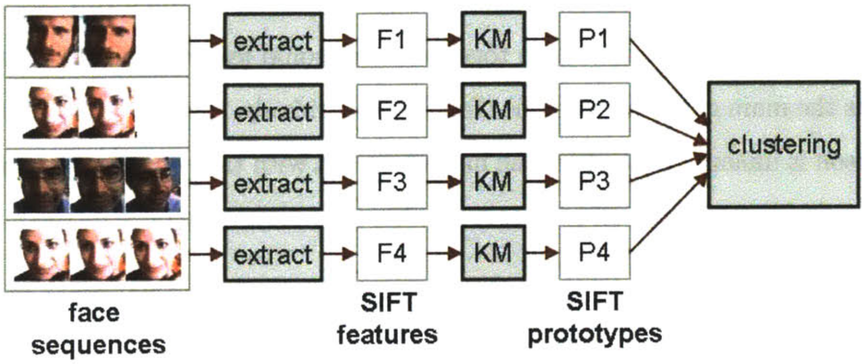


Figure 5-2: The Unsupervised Face Sequence Clustering Procedure. The input is a data set containing an arbitrary number of face sequences. Each face sequence is processed for feature extraction. Extracted features are then passed to the clustering system.

5.3.1 The Approach

The following are four properties of the solutions that we employ in our implementation of the face clustering system.

Use of face sequence We are using a video-based approach which deals with face sequences instead of images. The robot utilizes spatio-temporal context to perform same-person face tracking and obtain a sequence of face images that the robot assumes to belong to an individual. These face sequences provide a stepping stone for the unsupervised face clustering problem. Instead of clustering face images which may look very different from one another, we are clustering face sequences which contains images of varying poses and thus captures more information about a person's face. Similarly, the face recognition system also utilizes this spatio-temporal context such that it does not have to produce a hypothesis for every single image frame.

Use of local features We are using David Lowe's SIFT (Scale Invariant Feature Transform) to describe the face sequences [54]. SIFT is an algorithm for describing local features in a scale and rotation invariant way. Each SIFT feature or keypoint is represented by a 128-dimensional vector, produced by computing histograms of gradient directions in a 16x16 pixels image patch. SIFT feature descriptor is particularly suitable for our application as it has been shown to provide robust nearest-neighbor matching despite scale changes, rotation, noise, and illumination variations. Its invariance capacity is crucial in allowing our system to deal with the high pose variations and noise in the robot's collected face data.

Sparse alignment and Face Regioning We are using very sparse face alignment in our clustering solution. We simply use the face segmentation provided by the robot's face detector [98] and tracker [87]. We then blindly divide each face image into six regions, as shown in figure 5-4. The choice for a sparse alignment is an important design decision, as we want to step away from the requirement for a precise alignment, which is not yet solved for the case of multi-view faces. Many current face recognition methods are dependent on having good alignment of the face images. Therefore, the alignment accuracy of face images has been shown to cause a large impact on face recognition performance [101, 76, 86].

Clustering of local features We are using SIFT because it is a powerful local descriptor for small patches of image. However, this also means that we have to represent each data sample (face sequence) as a set of multiple SIFT keypoints extracted from different parts of the face. Thus, the task of clustering these local features in the 128-dimensional feature space is not straight-forward. We develop a clustering algorithm for local features based on the simple intuition that if two sequences belong to the same individual, there will be multiple occurrences of similar local image patches between these two sequences. In addition to providing a sparse alignment, the face regions are also utilized to enforce geometrical constraints in the clustering step. We describe the clustering algorithm in more details below.

5.3.2 A Toy Example

The clustering system's task is to cluster all face sequences in the unlabelled training set into a set of classes such that each class corresponds to each individual. In order to do this, we have to compute whether or not two sequences should be matched and put in the same class based on some distance metric. Standard distance measures do not work well in this setup since each training sample, i.e., a face sequence, is represented by a set of local features taken from different points of the image. Instead, we develop a matching algorithm for comparing two face sequences based on their local features. This matching procedure receives one face sequence as an input to be matched against a data set of an arbitrary number of sequences. It produces an output match set, which may contain an empty set if no match is found. If one or more matches are found, the output match set may contain anywhere from one up to all of the sequences in the training set.

Before we move on to the algorithm details, we first use a simple toy example to provide an intuition for the algorithm. In this example, as shown in figure 5-3, we would like to compare a face sequence S1 to a data set of two sequences, S2 and S3. Suppose we choose a feature in a region of sequence S1, highlighted by a red dot. We find ten nearest neighbors to this feature from other sequences in the data set, but only within the corresponding region, highlighted by the green dots. We then compute a list of the number of green dots found from S2 and S3, sorted from highest to lowest. In this simple example, S3 has 7 matches while S2 has 2 matches. Thus, S3 appears first on the list. The sequence matching algorithm is based on the following intuition. If a sequence is indeed a match to the input sequence, it is more likely to have more matches and thus is likely to appear higher on the sorted list. Moreover, it is more likely to appear higher on the sorted list in multiple regions of the face. In other words, if two sequences belong to the same individual, there will be multiple occurrences of similar local image patches between these two sequences.

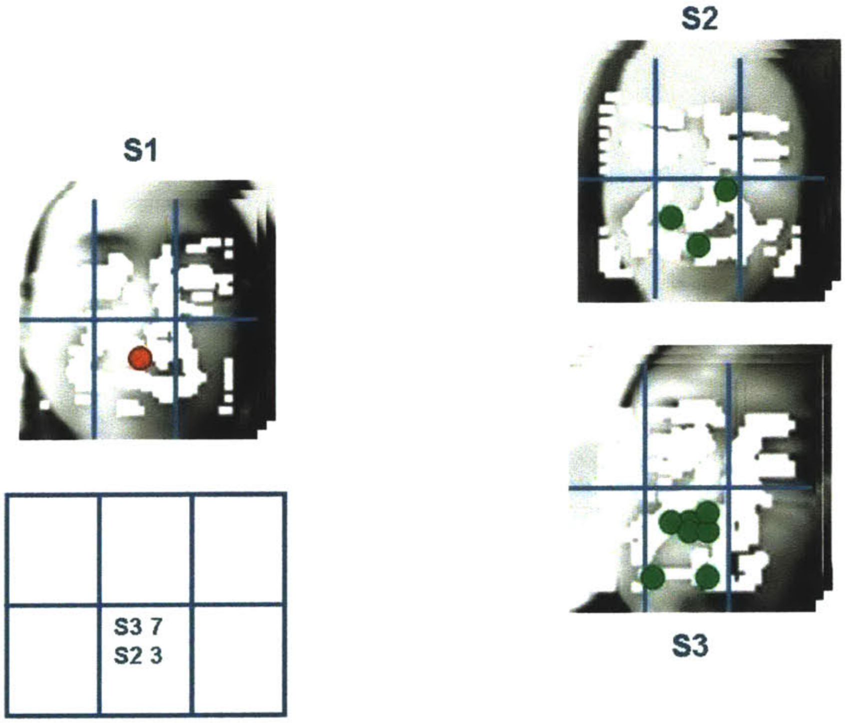


Figure 5-3: A simple toy example to provide an intuition for the sequence matching algorithm. In this example, we want to compare a face sequence S1 to a data set of two sequences, S2 and S3.

5.3.3 Implementation

We start by defining the clustering algorithm input, i.e., a data set containing an arbitrary number of sequences of face images Q.

$$Q = \{S_i | i \in [1, \dots, N]\} \tag{5.1}$$

where N is the total number of sequences.

A sequence of images S_i is defined as follows:

$$S_i = \{Im_{i,j} | j \in [1, \dots, N_i]\} \tag{5.2}$$

where N_i is the number of images in the sequence S_i.

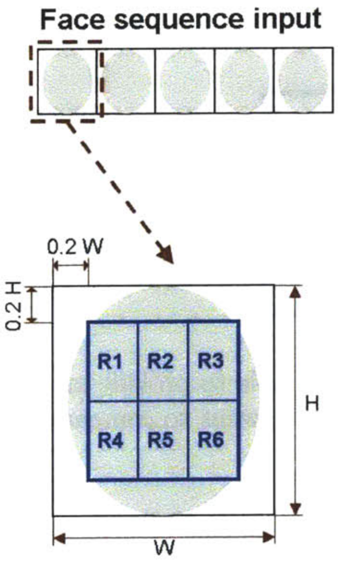


Figure 5-4: The division of face image into six regions. We initially divide each face image into six regions. This divison is done blindly without any precise alignment of face or facial features. In addition to providing a sparse alignment, the face regions are also utilized to enforce geometrical constraints in the clustering step.

Feature Extraction First, we extract a set of features from each face sequence S_i by performing the following steps.

We initially divide each face image $Im_{i,j}$ into six regions, as follows:

$$Im_{i,j} = \begin{bmatrix} R_1 & R_2 & R_3 \\ R_4 & R_5 & R_6 \end{bmatrix} \tag{5.3}$$

This division is done blindly without any precise alignment of face or facial features, as shown on figure 5-4.

For each image $Im_{i,j}$, we use the Harris corner detector to identify a set of interest points, which is a subset of all pixels in $Im_{i,j}$ [39]. We are using the Harris corner detector instead of the SIFT's keypoint finding method proposed in [54], because the latter method does not work as well with our lower resolution face images.

For each image $Im_{i,j}$, we then compute one SIFT 128-dimensional feature vector for each interest point and group the results in six batches. We define the function $\varphi(.)$ that computes SIFT feature vectors from a set of interest points in an image $Im_{i,j}$ and group each resulting feature vector into one of six batches according to which of the six regions its associated interest point comes from.

$$B_{i,j,m} = \varphi_m(Im_{i,j}) form = 1, \ldots, 6 \tag{5.4}$$

where $B_{i,j,m}$ is a set of 128-dimensional SIFT feature vectors for each region R_m, $B_{i,j,m} \in \mathbf{R}$

As shown in figure 5-5, we then combine all computed keypoints per region from all images in the sequence. Thus, in the end of this step, each sequence S_i is mapped to six batches of keypoints $A_{i,m}, m \in [1, \ldots, 6]$, as follows.

$$A_{i,k} = \bigcup_{j=1}^{N_i} B_{i,j,k} \tag{5.5}$$

Feature Prototype Generation In order to reduce the number of keypoints to be clustered, we convert them into a set of prototypes. For each batch of keypoints pro-

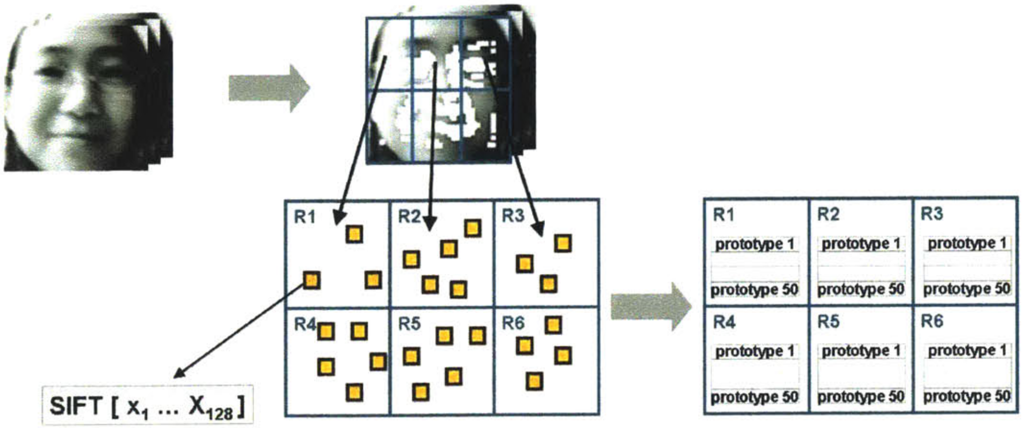


Figure 5-5: The feature extraction procedure of each face sequence. We extract SIFT features from each detected corner (using the Harris corner detector [39]) from each image in the sequence. We then combine all computed keypoints per region from all images in the sequence. Thus, in the end of this step, each sequence has six batches of keypoints, one batch for each of the six regions. Lastly, these six batches of keypoints are converted into a set of prototypes.

duced by each face sequence, we perform k-means clustering to compute 50 keypoint prototypes.

For this computation, we use KMlocal, a collection of software implementations of a number of k-means clustering algorithms [45, 44]. In particular, we use a hybrid algorithm, combining a swap-based local search heuristic and Lloyd's algorithm with a variant of simulated annealing.

Let's define the function $\Gamma_{50}(\cdot)$ that computes 50 k-means.

$$O_{i,m} = \Gamma_{50}(A_{i,m}) \qquad m \in [1, \ldots, 6] \tag{5.6}$$

Note that in practice, each $A_{i,m}$ contains at least 50 elements.

After this final step, as shown in figure 5-5, each sequence S_i is mapped to $O_{i,1}, \ldots, O_{i,6}$, where $O_{i,m}$ is defined as:

$$O_{i,m} = \left\{ C_{i,m,p} | C_{i,m,p} \in \mathbf{R}^{128}, p = 1, \ldots, 50 \right\} \tag{5.7}$$

, where each $C_{i,m,p}$ for $p = 1, \ldots, 50$ is one of the 50 prototype vectors output which are produced using k-means clustering.

Sequence Matching Now we have shown how to convert each sequence S_i in the data set Q to six sets of feature prototypes, $O_{i,1}, \ldots, O_{i,6}$.

As shown in figure 5-7, we use the extracted feature sets to compare each sequence S_i against the rest of the data set $R = \{S_x | x \neq i, x, \ldots, N\}$, using a function $\Psi(.)$ which produces an output set M_i containing matches for S_i.

$$M_i = \Psi(S_i, R) \tag{5.8}$$

M_i is a subset of $\{0, \ldots, N-1\}$.

We now define the function $\Psi(.)$, which takes in two inputs:

- a sequence input S_i which has been converted to six sets of 50 feature prototypes $C_{i,m,p}, m = 1, \ldots, 6, p = 1, \ldots, 50$.
- the rest of the data set $R = S_x | x \neq i, x \in [1, \ldots, N]$, which have been converted to $C_{x,m,p}, i = 1, \ldots, N, m = 1, \ldots, 6, p = 1, \ldots, 50$.

We describe each step of the sequence matching function $\Psi(.)$ using the following pseudo-code. This sequence matching procedure is also illustrated in figure 5-6.

Sequence Matching Algorithm

```
SequenceMatching($C_{i,m,p}$, $R$)
FOR each m
    FOR each p
        Find $K$ nearest neighbors to $C_{i,m,p}$ in region $m$ in $R$
            Define $Count_i(x, m, p)$ to be the number of nearest neighbors that
        come from region $m$ in sequence $S_x$
    ENDFOR
```

```
        Define $Count_i(x,m) = \sum_{p=1}^{50} Count_i(x,m,p)$
        Sort $Count_i(x,m)$ in descending order
        Take the top N elements in the sorted $Count_i(x,m)$
    ENDFOR
    Sequence $S_x$ matches $S_i$ iff $Count_i(x,m)$ is in the top N positions for all $m = 1, \ldots, 6$.
```

There is one missing detail in the above description. If the value of any element in the sorted $Count_i(x,m)$ list is $< C\% *$ the first element for some parameter C, this element and the rest of the elements in this sorted list of length N are excluded.

Unsupervised Face Clustering We have now shown how to compare each sequence S_i against the rest of the data set $R = \{S_x | x \neq i, x, \ldots, N\}$ to produce an output set M_i.

As shown in figure 5-7, we compare each sequence S_i against the rest of the data set $R = \{S_x | x \neq i, x, \ldots, N\}$, to produce an output set M_i containing matches for S_i.

If the output set M_i is not empty, the system will combine S_i and each element S_x in M_i into a cluster. This process is repeated for each M_i from each sequence S_i. This clustering step is performed greedily such that if any two clusters contain matching elements, the two clusters will be merged together.

5.3.4 Clustering Performance Metric

In order to evaluate the robot's unsupervised face clustering, we define a set of performance metric. Given the nature of our setup and failure modes, we feel that one single number is not sufficient to reflect both the merging and splitting errors. The clustering task is essentially a struggle between *merging* and *splitting* failures. For our purposes, *merging* multiple people into a cluster are more detrimental than *splitting* an individual's face sequences into multiple clusters. Both failures would lead to

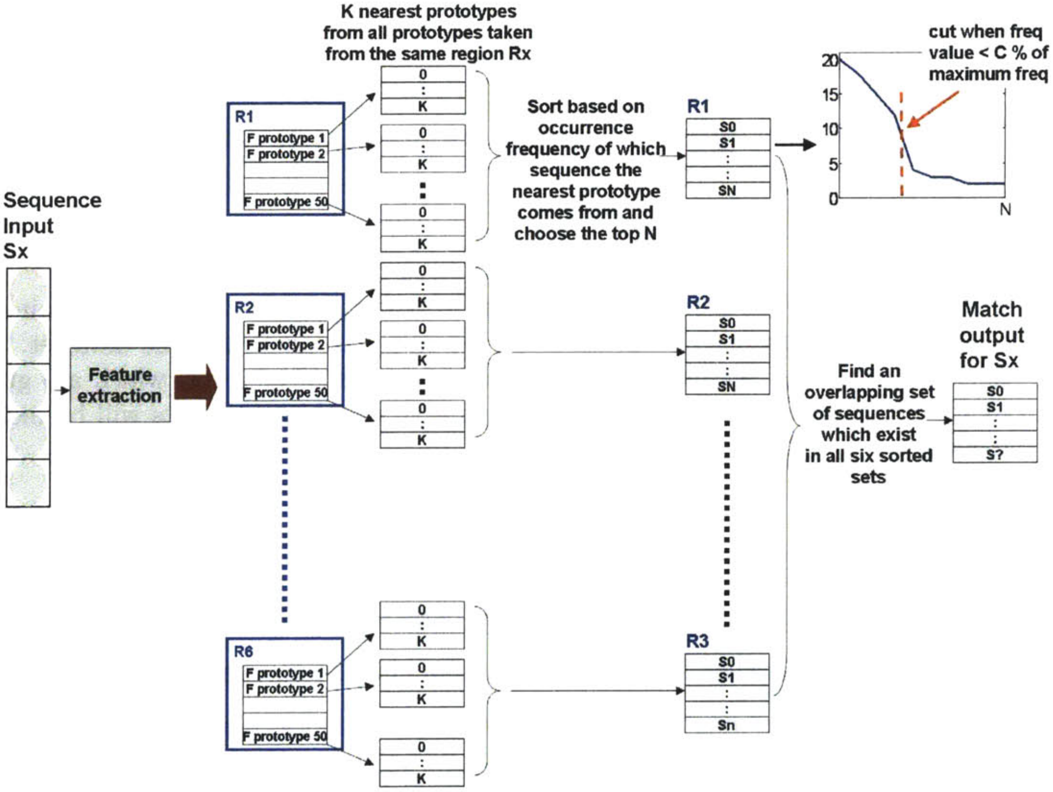


Figure 5-6: The Face Sequence Matching Algorithm. This matching procedure receives one sequence as input and produces an output match set containing one or more sequences. The intuition behind this algorithm is that if two sequences belong to the same individual, there will be multiple occurrences of similar local image patches between these two sequences.

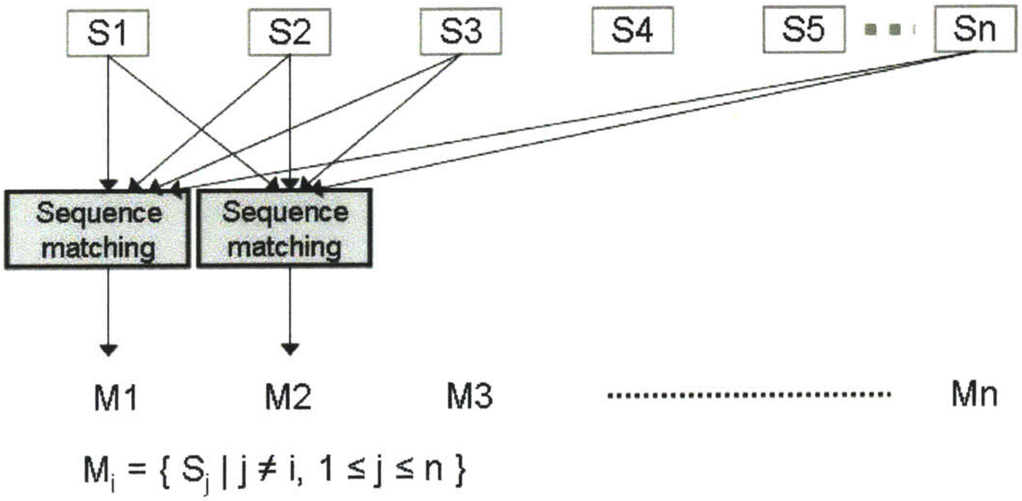


Figure 5-7: The Face Sequence Clustering Procedure. Given a data set containing t sequences, each sequence is compared to the rest of sequences in the data set. The sequence matching algorithm produces an output set of matches, which are greedily merged such that any two clusters containing matching elements will be merged together

false recognition, but in an incremental setup, the latter may get fixed if the robot acquires more data from the corresponding individual. Moreover, given the robot's greedy clustering mechanism, the *merging* failure has a compounding effect over time.

In order to reflect how the robot's clustering mechanism performs with respect to both of these failure modes, we opted for a set of metric. Given that an individual P has a set of sequences $X = S_i|i = 0, .., X_{size}$ in the training set, we define the following categories:

- *Perfect* cluster: if the system forms one cluster containing all elements in P's sequence set X .
- *Good* cluster: if the system forms one cluster containing $S_j|j = 0, .., M, M < X_{size}$ and leaves the rest as singletons. Note that the perfect cluster category is a subset of the good cluster category.
- *Split* cluster: if the system splits the elements of X into multiple clusters.

- *Merged* cluster: if the system combines sequences from one or more other individuals with any sequences inside X into a cluster. Note that the set of sequences which are labeled as non-faces are treated as if it is an individual. Thus, merging a non-face sequence into any cluster will be penalized in the same way.

We also define some additional metric for analyzing the split and merged clusters:

- *Split* degree: the proportion of the largest of the A clusters in a splitting case.
- *Merged* purity: the proportion of the number of sequences from individual I who holds the majority of the sequences in a merged cluster.
- *Merged* maximum: the maximum number of individuals merged together in a cluster from all of the merged cases.

The split degree and merged purity provide some information about the severity of a split or merged failure. A high split degree corresponds to a lower severity, as this means the clustering still successfully forms a significant cluster of an individual instead of many small ones. A high merged purity also corresponds to a lower severity, since it reflects cases where an individual still holds a significant majority of a cluster. If the merged purity value is very high, the few bad sequences may not be well represented and will not significantly affect recognition performance.

The merged maximum may provide more information that the total number of merged cases, when a large number of sequences are falsely merged together into a cluster. This would only yield one merged case, and thus its severity will not be reflected by the total number of merged cases.

Given an unlabelled training set containing face sequences from M individuals (according to ground truth), we then measure the clustering performance by the following measurements:

- *Number of people*: the number of individuals who has at least 2 sequences in the data set

- *None*: the number of individuals whose sequences did not get clustered at all
- *Perfect*: the number of perfect clusters
- *Good*: the number of good clusters, which also includes the perfect clusters
- *Split*: the number of split clusters
- *Split* degree: a distribution of the split degrees of all the split cases
- *Merged* purity: a distribution of the merged purity of all the merged cases
- *Merged* maximum: the maximum number of individuals merged together in a cluster from all the merged cases

5.3.5 Clustering Parameters and Evaluation

The sequence matching algorithm relies on three parameters, K, N, and C. K is the number of nearest prototypes used to form a sorted list of sequences with the most number of nearest prototypes. N is the maximum length of this sorted list which is then compared for overlaps with sorted lists from other face regions. C is a threshold value used to cut the sorted list in cases where some sequences dominate as the source of nearest neighbors over other sequences in the list. Thus, N becomes irrelevant when the threshold C is activated.

Note that the sequence matching algorithm does not rely directly on distance-based measures. Instead, it is very data dependent as it relies on voting among nearest neighbors and spatial configuration constraints. In order to assess the algorithm's sensitivity to various factors, we provide an analysis of the robot's clustering performance across a range of data-dependent properties and parameter values K, N, and C below.

We extract a number of data sets of different sizes from the collected training data. Each data set was formed by taking contiguous segments from the robot's final training data in order of its appearance during the experiment. Thus, each set contains a similar distribution of number of individuals and number of sequences per

Table 5.1: The data set sizes and parameter values used in the clustering algorithm evaluation.

Data set size	K	N	C
30	10	3	0%
300	30	5	30%
500	50	10	50%
700		15	70%
1000		20	
2025		30	
		40	
		50	

individual shown in table 4.9. We then perform the clustering algorithm on each data set and computed a set of metric described above. We show a subset of these results below and provide the complete set in appendix C. These results were generated using data set sizes and parameter values listed in table 5.1.

Medium to large data set sizes Figure 5-8 and 5-9 are two sample results generated from a data set of 300 and 700 sequences using different combinations of N and K as listed in table 5.1. These parameter values are shown in the lowest subplot of each figure. The C parameter is kept constant at 30%.

The top most subplot ilustrates the number of *good*, *perfect*, and *none* resulting clusters. The second subplot shows the number of total merging errors and their distribution for different merged purity values (0-25, 25-50, 50-75, 75-100%). The third subplot shows the number of merged maximum. This measure is more indicative then the total merging errors when a large number of sequences are falsely merged together. The fourth subplot shows the number of splitting errors and their distribution for different split degree values (0-25, 25-50, 50-75, 75-100%).

Figure 5-10 shows the normalized number of merging and splitting errors. These plots essentially illustrate the trade-off between merging and splitting, typically encountered in any unsupervised clustering task. Based on all of the clustering results of data sets of various sizes using different parameter values, we observe that increasing

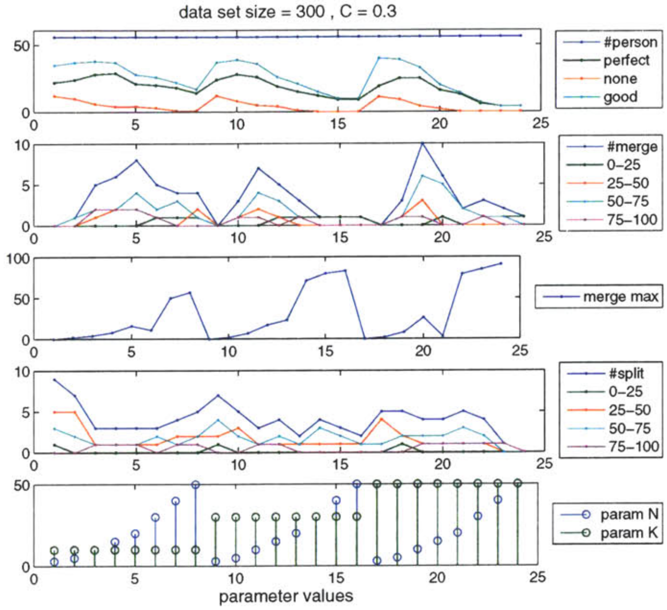


Figure 5-8: The clustering results with a data set of 300 sequences with different N and K parameter values while C is kept fixed at 30%. The top most subplot ilustrates the number of *good*, *perfect*, and *none* resulting clusters. The second subplot shows the number of total merging errors and their distribution for different merged purity values . The third subplot shows the number of merged maximum. The fourth subplot shows the number of splitting errors and their distribution for different split degree values.

the parameter K results in higher merging errors, but does not necessarily reduce the splitting errors. Thus, for the rest of this evaluation, we assume that K should be kept on the lower side.

Figure 5-11 illustrates how these trade-off curves between merging and splitting errors change as we vary our data set sizes, N, and C, while keeping K fixed at 10. Generally, turning the knob on N slides us along the split and merge trade-off. While the results are satisfying when we are in a good zone for N, we do not want to have to carefully tune parameter values each time. As N increases, merging errors increase while splitting errors decrease. However, this effect is diminished as the data set size increases. Similarly, we also observe the same diminishing effect as the parameter C

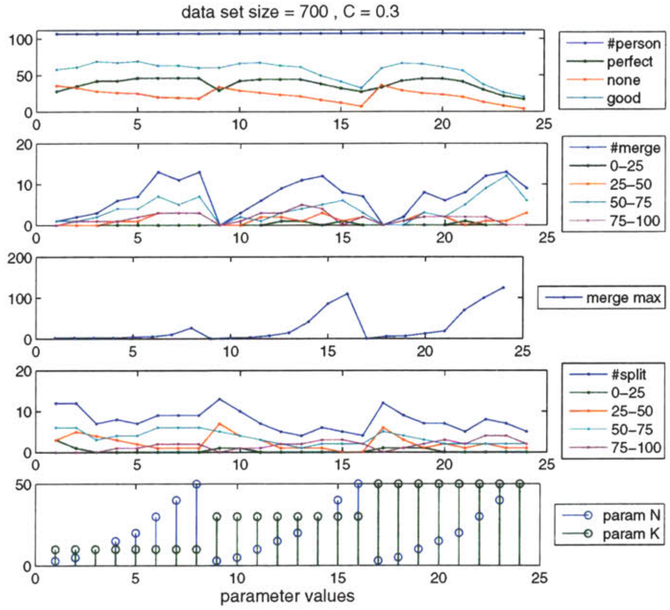


Figure 5-9: The clustering results with a data set of 700 sequences with different N and K parameter values while C is kept fixed at 30%. The top most subplot ilustrates the number of *good*, *perfect*, and *none* resulting clusters. The second subplot shows the number of total merging errors and their distribution for different merged purity values . The third subplot shows the number of merged maximum. The fourth subplot shows the number of splitting errors and their distribution for different split degree values.

increases. For larger data sets or higher C values, the merging errors still increase and the splitting errors still decrease as N increases, but not nearly as much. This means that the splitting errors will generally be slightly higher. However, it allows for a larger margin for how to specify N without sacrificing too many merging or splitting errors. We later utilize these properties in our parameter specification strategies.

Small data set sizes Figure 5-12 shows the trade-off curves between merging and splitting errors for a data set of 30 sequences, with different K, N, and C values. These curves exhibit the a similar trend as those of the larger data set. When C is low, merging errors increase while splitting errors decrease as we turn the knob on N. When C is high, the curves flatten. However, there is a difference. When C is low, the

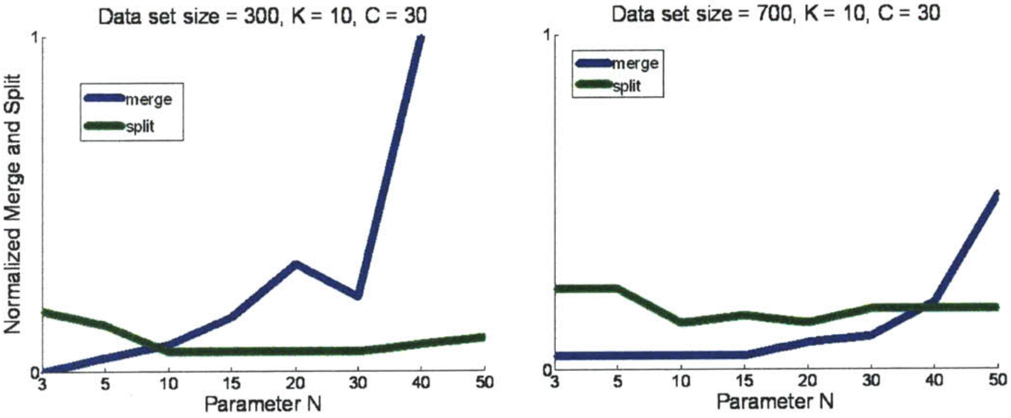


Figure 5-10: The normalized number of merging and splitting errors from the clustering results shown in figure 5-8 and 5-9. These plots essentially illustrate the trade-off between split and merge errors, typically encountered in any unsupervised clustering task.

merging error increases drastically as N increases such that there is only a very small good trade-off zone which occurs when N is very low. In terms of K, the results are consistent with our previous finding that increasing K simply magnifies the merging errors without improving the splitting errors. We incorporate these observations in our summary and parameter specification strategies later.

Different sequence distributions within the data set We have so far analyzed the clustering results on different contiguous subsets of the robot's collected training data in the order of its appearance during the experiment. Each set contains a large variation in the number of sequences per individual, ranging from one to over two hundred sequences per person. In order to assess the clustering algorithm's sensitivity to the distribution of sequences per person in the data set, we conducted the same clustering tests using data sets with different sequence distributions per person.

Figure 5-13 and 5-14 show the trade-off curves between merging and splitting errors for data sets with two different distributions of sequences per person, at different C and N values while keeping K fixed at 10.

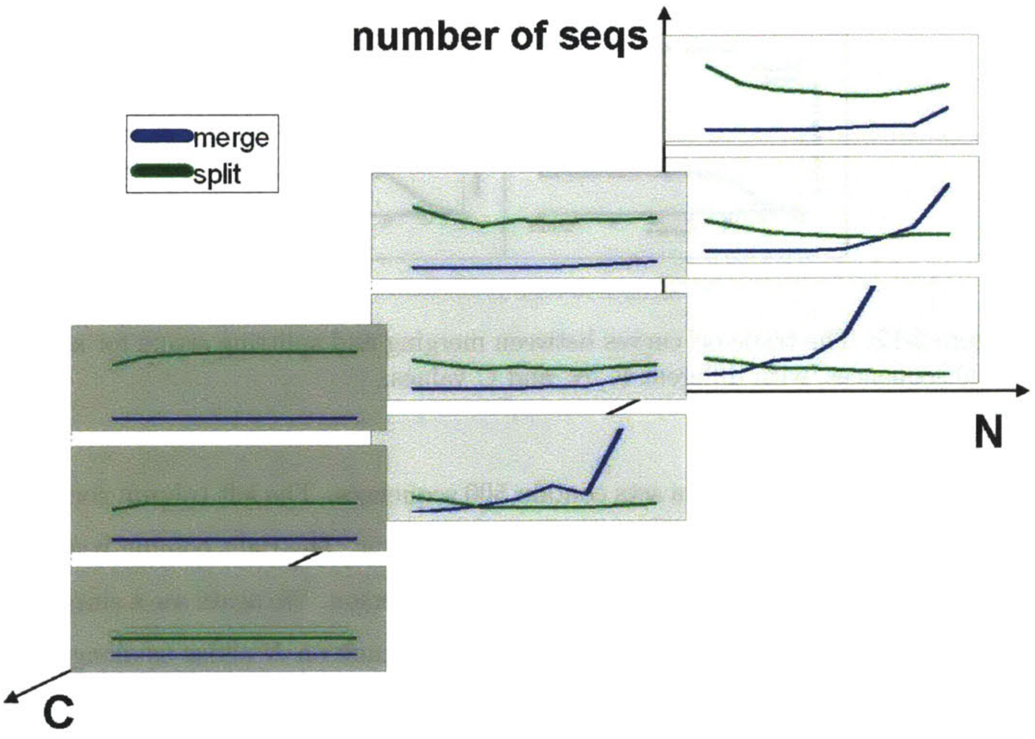


Figure 5-11: The trade-off curves between merging and splitting errors at different data set sizes and values of N and C. As N increases, merging errors increase while splitting errors decrease. However, this effect is diminished as the data set size increases. Similarly, we also observe the same diminishing effect as the parameter C increases. For larger data sets or higher C values, the merging errors still increase and the splitting errors still decrease as N increases, but not nearly as much.

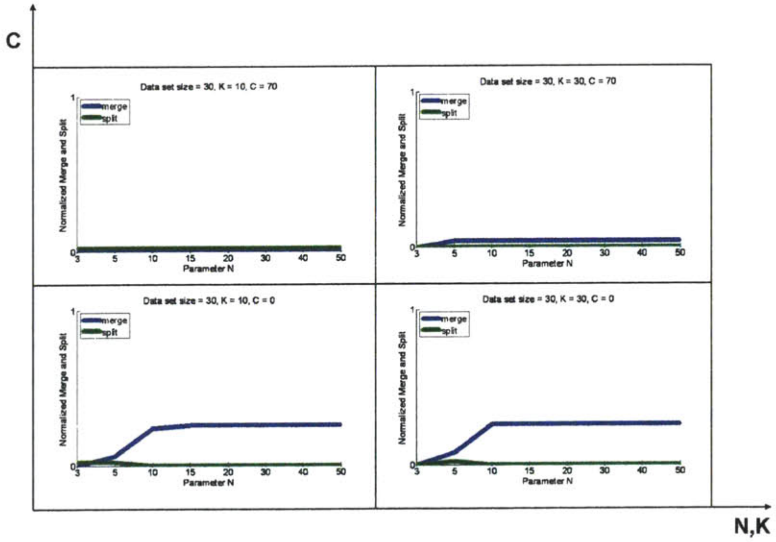
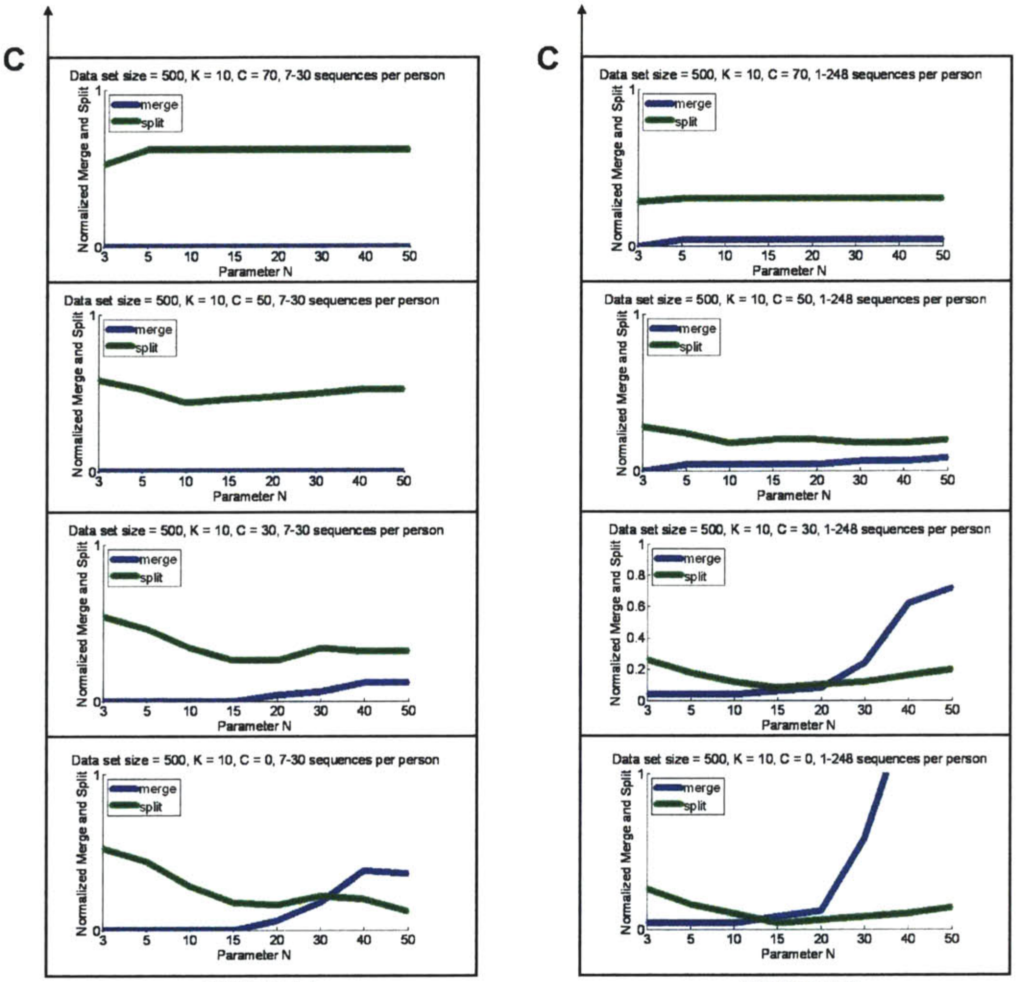


Figure 5-12: The trade-off curves between merging and splitting errors for a data set of 30 sequences, with different K, N, and C values.

In figure 5-13, both data sets contain 500 sequences. The left column corresponds to the first distribution of 7-30 sequences per person. The right column corresponds to the second distribution of 1-248 sequences per person. We again see a similar trend which we previously observed, where turning the knob on N slides us along a trade-off between merging and splitting errors. Increasing the parameter C diminishes this effect and thus flattens the curves. However, comparisons between the two distributions indicate that the first distribution yields less merging errors and more splitting errors. This is due to the fact that in the first distribution, each person has more sequences to be clustered. Thus, there is a higher chance that the clustering algorithm ends up splitting their sequences. On the other hand, in the second distribution, a large percentage of the people have only 2-3 sequences to be clustered.

In figure 5-14, we see the opposite case where the left column corresponds to the first distribution of 1-4 sequences per person. The right column corresponds to the second distribution of 1-248 sequences per person. In this case, we observe that the first distribution yields less splitting errors. In fact, sliding N to higher values increases both the merging and splitting errors.



Distribution 1
500 sequences
7-30 seq per person

Distribution 2
500 sequences
1-248 seq per person

Figure 5-13: The trade-off curves between merging and splitting errors for data sets with two different distributions of sequences per person, at different C and N values while keeping K fixed at 10. Both data sets contain 500 sequences. The left column corresponds to the first distribution of 7-30 sequences per person. The right column corresponds to the second distribution of 1-248 sequences per person.

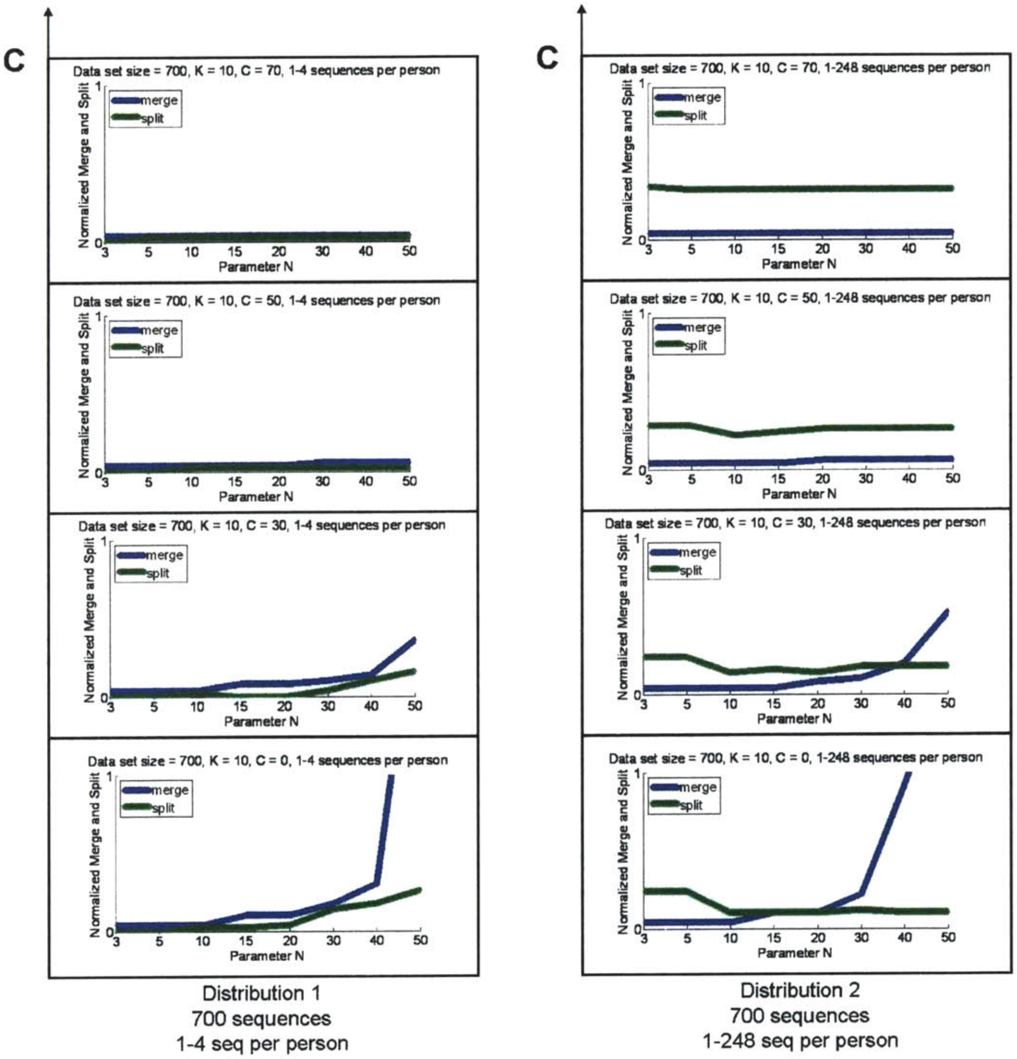


Figure 5-14: The trade-off curves between merging and splitting errors for data sets with two different distributions of sequences per person, at different C and N values while keeping K fixed at 10. Both data sets contain 700 sequences. The left column corresponds to the first distribution of 1-4 sequences per person. The right column corresponds to the second distribution of 1-248 sequences per person.

These results indicate that the distributions of sequences per person in the data set affect the shape of the trade-off curves at different values of N. Essentially, if the data set contains few face sequences per individual, there is a higher chance for merging errors to occur, especially for individuals who has only one sequence and therefore may be falsely matched to someone else's sequence. However, there is a lower chance for splitting errors since there are fewer sequences to be clustered. On the other hand, if the data set contains many face sequences per individual, there is a lower chance for merging errors to occur because there is a match for most of the face sequences. However, there is a higher chance for splitting errors since there are more sequences to be clustered per person.

Lastly, we observe that increasing the parameter C value has the same effects of flattening the trade-off curves regardless of the data set size or sequence distribution.

Sequence Matching Accuracy Figure 5-15 shows the accuracy of the sequence matching algorithm when performed on data sets of different sizes and using different parameter C values. This accuracy measure is defined as the percentage of occasions when every element in the matching output set declared by the algorithm is indeed a correct match to the input sequence.

These accuracy measures indeed confirm our previous observations. For larger data sets, decreasing the parameter C value causes a slight decrease in the accuracy rate. However, for smaller data sets, decreasing the parameter C value drastically reduces the accuracy rate, even all the way down to 0% for data sets of 30 sequences. This corresponds to a case where all sequences are falsely merged together into a single cluster.

Based on these numbers, it seems straight forward that we should use high values of C to obtain the best results. However, keep in mind that a high accuracy value reflects low merging errors, however it does not reveal anything about the splitting errors. Instead we suggest to correlate C to the data set size. We discuss these parameter specification strategies in more details below.

N of sequences	N	K	C = 70%	C = 50%	C = 30%	C = 0%
2025	30	10	98.9	97.3	96.3	93.6
1000	30	10	99.4	97.7	93.8	89.3
500	30	10	98.5	95.2	89.5	76.5
300	30	10	98.4	▨	▨	60.9
30	30	10	93.6	▨	▨	0

Figure 5-15: T he accuracy of the sequence matching algorithm when performed on data sets of different sizes and using different parameter C values. This accuracy measure is defined as the percentage of occasions when every element in the matching output set declared by the algorithm is indeed a correct match to the input sequence.

5.3.6 Summary and Parameter specification strategy

Based on these results, we make the following observations and propose a set of strategies.

- Turning the knob on the N parameter slides us along the trade-off between merging and splitting errors. As N increases, the merging error also increases while the splitting error decreases.

- Increasing parameter K causes higher merging error, but does not decrease splitting error. Thus, the value of K can be fixed at a low value.

- Parameter C and N are related in that C makes N irrelevant when the algorithm finds one or more sequences which dominate in the nearest-neighbor votes over the rest of the data set. Parameter C can be thought of as conservative measure. Increasing parameter C value allows the system to be more conservative because it essentially flatten the trade-off curves between merging and splitting. In other words, at higher values of C, the algorithm is more conservative in declaring a match and thus generate lower merging errors but higher splitting errors. This property allows us to keep N at a fixed value and vary C depending on how conservative we want the algorithm to be.

- Increasing the size of the data set has a similar effect as increasing the parameter C value. This means that when we have more data, the algoritm is less susceptible to merging errors. Our intuition for this is that as more data fills up the feature space, the algorithm's sorted list of nearest prototypes has a smaller chance in catching false positive neighbors. Thus, having a lot of data reduces the chance of a false positive match. Given these properties, when we have a lot of data, we can reduce the C value and be a lot more conservative in our clustering process.

- To summarize, we propose to keep K to be fixed at a low value and N at a middle-range value. We have observed that for a data set of 30 sequences, the merging error increases drastically as we increase N. Thus, for very small data sets, we would need to use a low value of N.

- For smaller data sets, we have to be more conservative as it is more susceptible to false matches. For larger data sets, we can be less conservative as it is more immune to false matches. Thus, we propose to correlate the parameter C to the data set size. For all of the experiments using data sets of over 300 sequences, we use K = 10 and N = 30. We also tested the clustering performance using a range of correlation function between C and the data set size, as shown in figure 5-29.

Using A Different Data Set – The Honda/UCSD Video Face Database In order to analyze the robot's clustering performance on a different dataset, we conducted a test using the Honda/UCSD Video Database [51, 50]. The database was created to provide a standard database for evaluting video-based face tracking and recognition algorithms. The database contains 72 video sequences from 20 individuals. All individuals are supposed to have more than one sequence in the database. However, in the version that we downloaded, we only have 19 people with more than one sequence and one person with one single sequence.

Each video sequence contains one person at a time, lasts for at least 15 seconds, and is recorded in an indoor environment at 15 frames per second and 640x480 resolution. In each video, the person rotates and turns his/her head in his/her own preferred order and speed. Thus, typically in about 15 seconds, the video captures a wide range of poses with significant in-plane and out-of-plane head rotations.

In our tests, we reduce the video resolution to 160x120 for face tracking and retrieve the face images to be stored at 320x240 to match the setting used in the robot's visual processing. Using the robot's face detector and tracker, we convert each video into a face sequence. We then processed and clustered these face sequences in the same way as we did in our previous test with the robot's self-generated training set.

Table 5.2 shows the different measurements for the 19 individuals in the database with more than one sequence. Since the size of the database is small (72 sequences), we knew ahead of time that it would need small parameter values. The clustering performance is in fact highest at the smallest parameter values of $N = 3, K = 10$. Out of 19 people, the clustering algorithm formed 18 *good* clusters. Nine of these are *perfect* clusters. One individual was split with a 60% split degree.

We also show how these results degrade as the K and N parameter values increase in figure 5-16. The plot structure is the same as that of figure ??. The parameter C is set at a high value of 70%, as we know that we have to be conservative with very small data sets. The clustering performance did not degrade too badly even as both N and K are increased. The algorithm still managed to find *good* clusters for at least 75% of the individuals with only one merged cluster for most the parameter values. Performance is worst at the highest values of $N = 30, K = 30$.

5.4 The Integrated Incremental and Unsupervised Face Recognition System

We incorporate the face clustering solution decribed above to implement an integrated system for unsupervised and incremental face recognition, as illustrated in figure 5-17.

Table 5.2: The batch clustering results using the Honda-UCSD face video database

person	total n seq	n seq in cluster/total n seq	split degree	merge purity
1	9	0.44	1	1
2	5	1	0.6	1
3	5	0.4	1	1
4	5	1	1	1
5	4	0.5	1	1
6	3	1	1	1
7	3	0.67	1	1
8	3	0.67	1	1
9	3	0.67	1	1
10	3	1	1	1
11	3	1	1	1
12	3	0.67	1	1
13	3	0.67	1	1
14	3	1	1	1
15	3	1	1	1
16	3	1	1	1
17	3	1	1	1
18	2	1	1	1
19	2	1	1	1

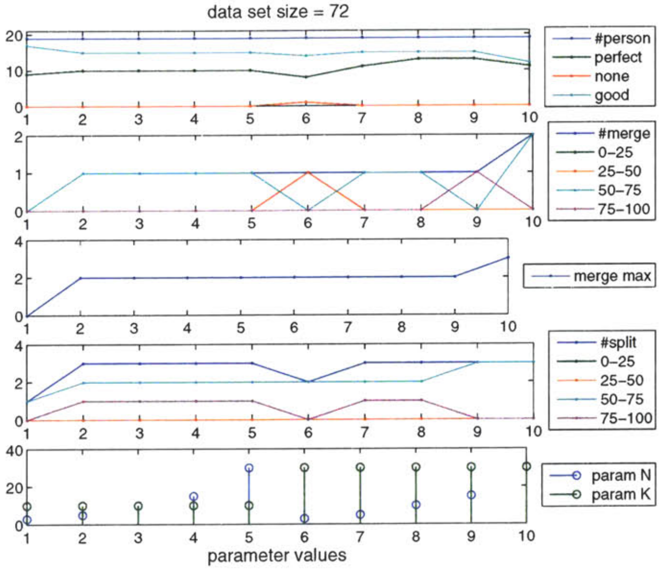


Figure 5-16: The face sequence clustering results with the Honda-UCSD video face database, using different N and K values, while keeping C fixed at 70%. The clustering performance did not degrade badly even as both N and K are increased. The algorithm still managed to find *good* clusters for at least 75% of the individuals with only one merged cluster for most the parameter values.

It consists of two separate training sets, an unlabelled one for the clustering system and a labelled one for the recognition system. Figure 5-18 illustrates the four phases of the incremental face recognition process, from left to right.

- In the first phase, both training sets are empty.
- In the second phase, the clustering system starts receiving one face sequence at a time and simply stores them until the clustering training set contains 300 unlabelled face sequences.
- At this point, we enter the third phase where multiple events occur. First, the clustering system performs a batch clustering on the stored 300 face sequences. Second, from the resulting clusters, M largest ones are automatically transferred to form labelled data in the recognition training set where each cluster corresponds to a class. Third, upon formation of these labelled training data, each

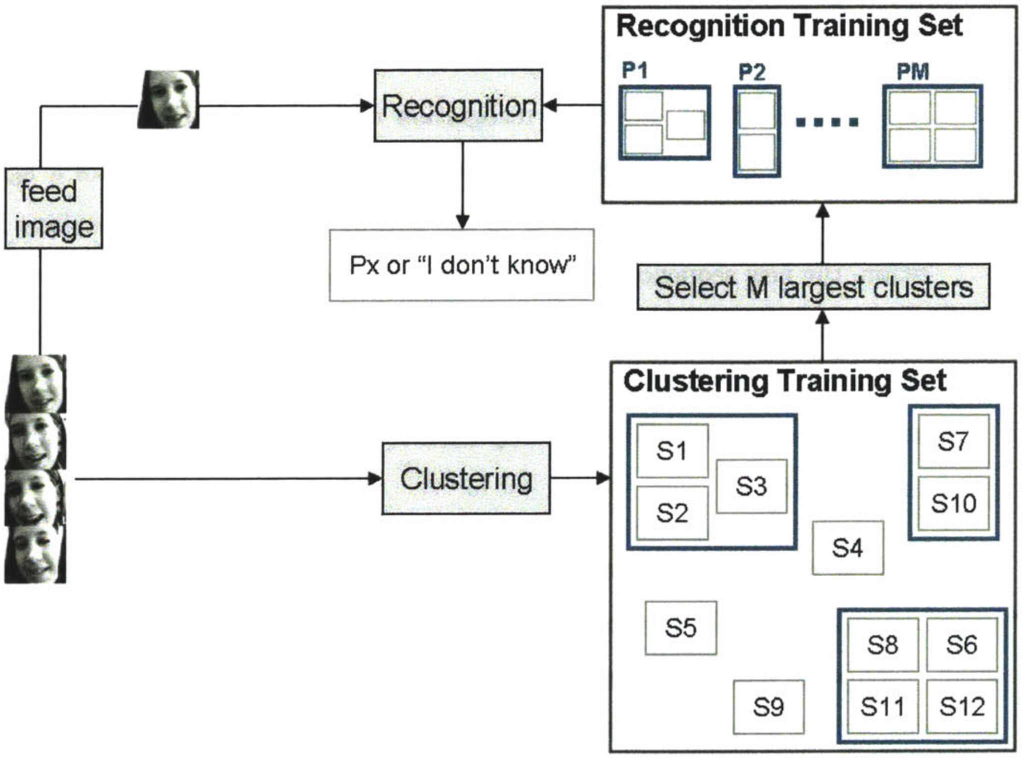


Figure 5-17: The unsupervised and incremental face recognition system. It consists of two separate training sets, an unlabelled one for the clustering system and a labelled one for the recognition system. Both training sets are initially empty. Over time, the system incrementally builds a labelled training set using self-generated clusters and then uses this training set to recognize each sequence input.

image from the sequence input is then passed to the recognition system one at a time. Based on the current labelled training set, the recognition system produces a running hypothesis based on each face image from the sequence input. The recognition system may decide on a final hypothesis after an arbitrary number of face images, that the sequence input belongs to either a particular person in the training set or an unknown person. The system then stops its processing and ignores the rest of the sequence input.

- In the fourth phase, after the recognition makes a final hypothesis, each sequence input is also passed to the clustering system, which matches it against the existing clusters. Depending on the output of the sequence matching algorithm, the new sequence input may be integrated into an existing cluster, form a new cluster, or remain as a singleton. This incremental change is subsequently reflected in the labelled training set, which will then be used to recognize the next sequence input. We then loop back to another round of recognition as in the third phase.

Note that the recognition hypothesis and the output of the sequence matching algorithm are redundant. Both essentially determine which existing cluster the sequence input belongs to, if any. However, in our implementation, we only use the sequence matching output to update our existing clusters.

5.4.1 The Supervised Variant

In theory, the supervised face recognition module can be filled by any existing supervised face recognition system. Since we are interested in evaluating our approaches, we implemented the supervised face recognition module using a variant of the sequence matching algorithm. Instead of face sequences, the face recognition receives a single face image as input. Moreover, the sequence matching algorithm relies on a k-means clustering to obtain feature prototypes which does not work in real time. Thus, the matching algorithm has to be adapted to accommodate single image input and faster processing. Figure 5-19 illustrates this adapted algorithm. It is similar

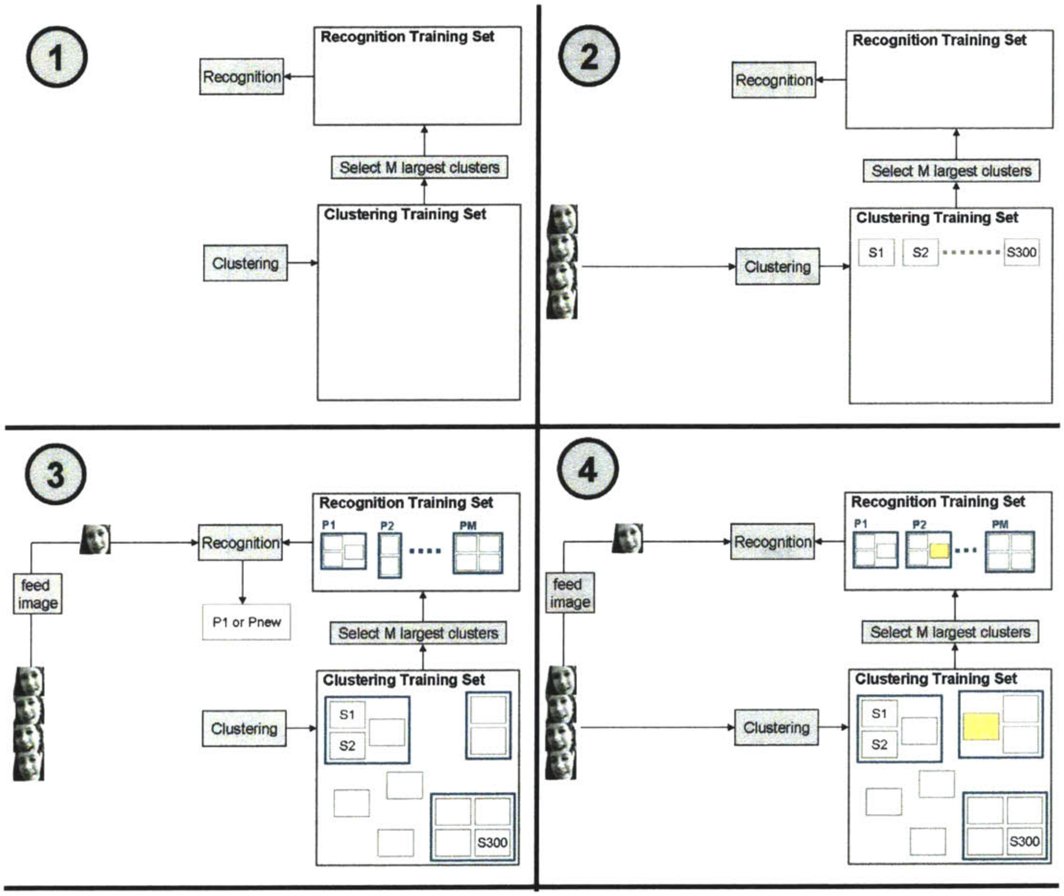


Figure 5-18: The four phases of the unsupervised and incremental face recognition process. Curing this incremental process, the system builds a labelled training set using self-generated clusters and then uses this training set to recognize each sequence input.

to the original algorithm, except that instead of matching feature prototypes of a sequence, it matches the original features directly from each input image.

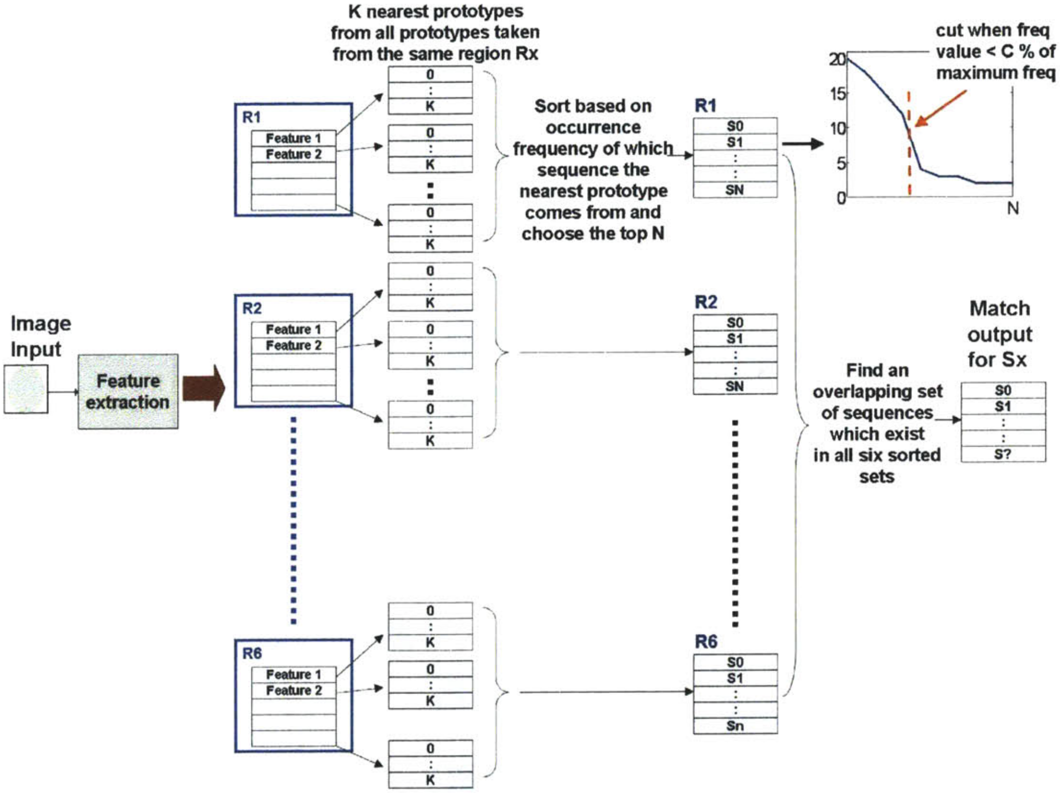

K nearest prototypes from all prototypes taken from the same region Rx
Image Input
Feature extraction
R1
Feature 1
Feature 2
R2
R6
0
:
K
Sort based on occurrence frequency of which sequence the nearest prototype comes from and choose the top N
S0
S1
SN
Sn
cut when freq value < C % of maximum freq
20
15
10
5
0
N
Find an overlapping set of sequences which exist in all six sorted sets
Match output for Sx
S?

Figure 5-19: The adapted sequence matching algorithm for supervised face recognition. This adapted version is very similar to the original algorithm, except that instead of matching feature prototypes of a sequence, it matches the original features directly from each input image.

5.4.2 Incremental Recognition Results

We conducted two tests to evaluate the integrated system. Figure 5-20 shows the incremental recognition results of each sequence input generated from the first test. In this test, the labelled recognition training set is incrementally constructed using fifteen largest self-generated clusters, containing at the minimum two sequences. The lower sub-plot shows the number of sequences in the incrementally constructed labelled training set, which increases as the robot encounters more input data over time.

For each sequence input, the system makes a recognition hypothesis that it either belongs to a specific known person Px or an unknown person who is not the the training database. The upper sub-plot shows the hypothesis accuracy of the first case. The blue and red solid lines correspond to the accummulated number of correct and incorrect hypotheses respectively over time. When the system hypothesizes that the sequence input belongs to a known person, it is correct 74.5% of the time. The slope of the number of incorrect hypotheses decreases over time as the size of the labelled training set increases. If we calculate the recognition performance after some delay, as shown by the blue and red dotted lines, the performance improves to being correct 81.8% of the time.

We calculated the recognition performance for some familiar individuals who encountered the robot on multiple days. The familiar individual whose cluster is shown in figure 5-26 came to interact with the robot on seven different days. Once the system integrates her cluster into the labelled training set, the recognition system correctly classified her 15 times and misclassified her as another person and an unknown person 2 and 6 times respectively. The familiar individual whose cluster is shown in figure D-4 interacted with the robot on two different days. Once the system integrates his cluster into the labelled training set, the recognition system correctly classified him 19 times and misclassified him as another person and an unknown person 4 and 5 times respectively.

The middle sub-plot illustrates the recognition accuracy of the second case where the system makes an unknown person hypothesis. The system is correct 74% of the time when it hypothesizes that the sequence input belongs to an unknown person who does not exist in the training database. As shown in figure 5-17, after each sequence input is processed for recognition, it subsequently goes through the clustering system and incrementally integrated into the existing self-generated clusters. Roughly 14.3% of the sequence inputs, which were falsely recognized as an unknown person, were later correctly integrated into an existing cluster, as shown by the green line.

Figure 5-21 shows the incremental recognition results from the second test. In this

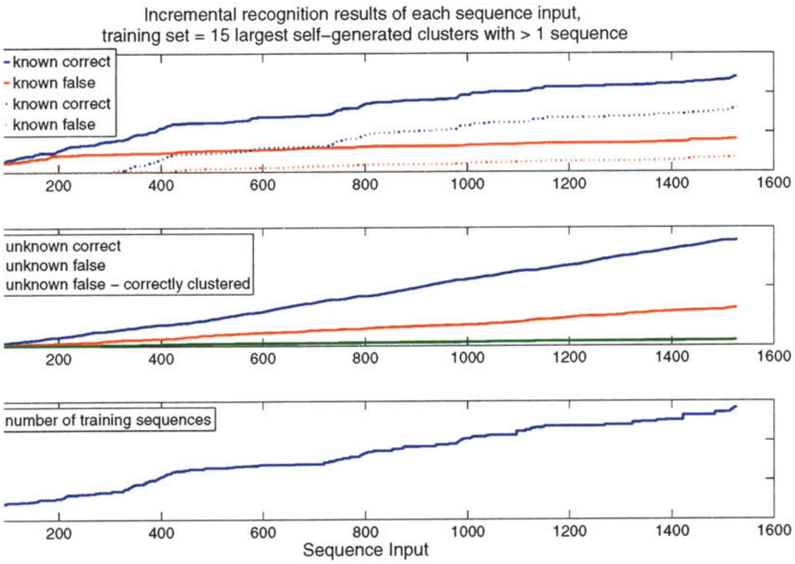


Figure 5-20: The incremental recognition results of each sequence input. The labelled recognition training set is incrementally built using fifteen largest self-generated clusters, containing at the minimum two sequences. The lower sub-plot shows the number of sequences in the incrementally constructed labelled training set, which increases as the robot encounters more input data over time. The upper sub-plot shows the recognition hypothesis accuracy. The blue and red solid lines correspond to the accummulated number of correct and incorrect hypotheses respectively over time. The middle sub-plot illustrates the recognition accuracy of the second case where the system makes an unknown person hypothesis.

test, the setting is similar except that the labelled recognition training set is incrementally built using fifteen largest self-generated clusters, containing at the minimum six sequences. At this setting, when the system hypothesizes that the sequence input belongs to a known person, it is correct 47.4% of the time. We attribute the lower recognition performance to the smaller size of the labelled training set due to the more selective process of transferring only clusters containing at least six sequences. However, if we calculate the recognition performance after some delay, as shown by the blue and red dotted lines, the performance improves to being correct 77.6% of the time. Thus, once the labelled training set is large enough, the performance is comparable to that of the previous test.

When the system hypothesizes that the sequence input belongs to an unknown person who does not exist in the training database, the performance is also comparable to that of the previous test. The system is correct 74% of the time in declaring that a person is unknown. Roughly 10.8% of the sequence inputs, which were falsely recognized as an unknown person, were later correctly integrated into an existing cluster.

To summarize, we evaluated the incremental and fully unsupervised face recognition system using the face data automatically generated by the robot during the final experiment. During this incremental clustering and recognition test, the system builds a labelled training set in a fully unsupervised way and then uses this training set to recognize each sequence input. The system hypothesizes correctly 74% of the time that the sequence input belongs to an unknown person who does not exist in the training set. After an initial learning period, when the system hypothesizes that the sequence input belongs to a specific known person, it is correct roughly 80% of the time.

5.4.3 The Self-Generated Clusters

Figure 5-22 shows the clustering results produced during the incremental recognition process. These results were generated with $N = 30$, $K = 10$, and C is correlated with the data set size using the function *inc3* shown in figure 5-29. The results are plotted

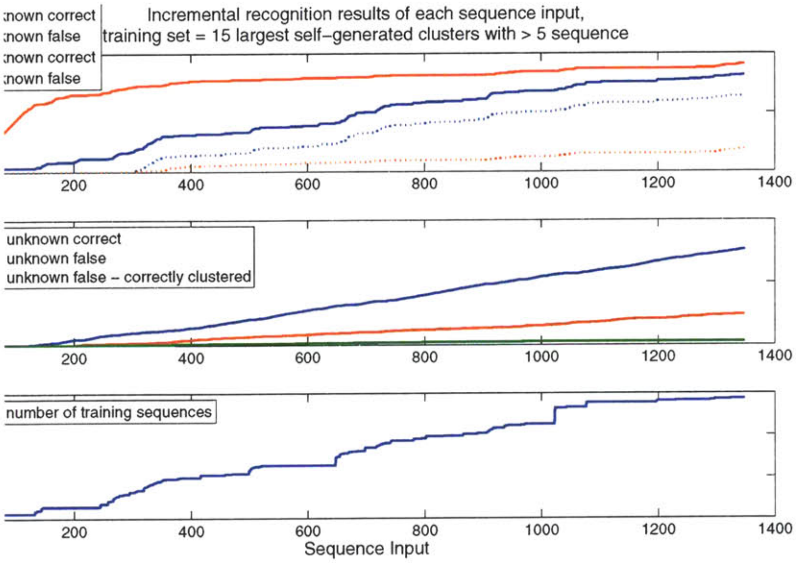


Figure 5-21: The incremental recognition results of each sequence input using a different setting. The labelled recognition training set is incrementally built using fifteen largest self-generated clusters, containing at the minimum six sequences. The lower sub-plot shows the number of sequences in the incrementally constructed labelled training set, which increases as the robot encounters more input data over time. The upper sub-plot shows the recognition hypothesis accuracy. The blue and red solid lines correspond to the accummulated number of correct and incorrect hypotheses respectively over time. The middle sub-plot illustrates the recognition accuracy of the second case where the system makes an unknown person hypothesis.

at every addition of 20 sequences, starting from the initial batch of 300 sequences. We provide some examples of these generated clusters in appendix B.

At the end of the test, the clustering algorithm found 151 *good* clusters. 75 of these are *perfect* clusters. The algorithm made 22 *merging* failures, with the largest merged cluster containing 3 individuals. In more than half of these merged cases, the merge purity value is between 75-100%. There are 26 *split* clusters. In more than half of these split cases, the split degree value is between 50-75%.

Figure 5-23 provides a visualization of these self-generated clusters. Each pie chart corresponds to an individual. The size of the pie chart circle corresponds to the number of sequences an individual have in the data set. The pie charts are ordered from individuals with the most to the least number of sequences. We exclude individuals who have only one sequence in the data set. The different shades of green regions correspond to sequences that were correctly clustered. Multiple green slices are present within a pie chart indicate that there is a splitting error. The red regions correspond to sequences that were falsely clustered, i.e. the merging errors. The gray regions correspond to unclustered sequences.

Figure 5-24 shows six different snapshots of the largest fifteen clusters taken at different times during the first incremental recognition test. These largest fifteen clusters were essentially the content of the labelled recognition training set. In the beginning, the selected clusters come from individuals who are not most familiar, i.e. only have a few sequences. At the end, the selected clusters converge to the top, i.e. the familiar individuals who have had many encounters with the robot. Note that these fifteen clusters do not actually correspond to fifteen different people, since there are some splitting cases. In particular, the first individual represented by the largest pie chart is represented by four clusters and therefore as four different classes in the labelled training set.

Figure 5-25 through 5-28 shows some examples of these largest clusters of the familiar individuals. Note that not all of the sequences are shown in these figures, due to visibility and space constraints. However, we make sure to include all of the merged errors, when present.

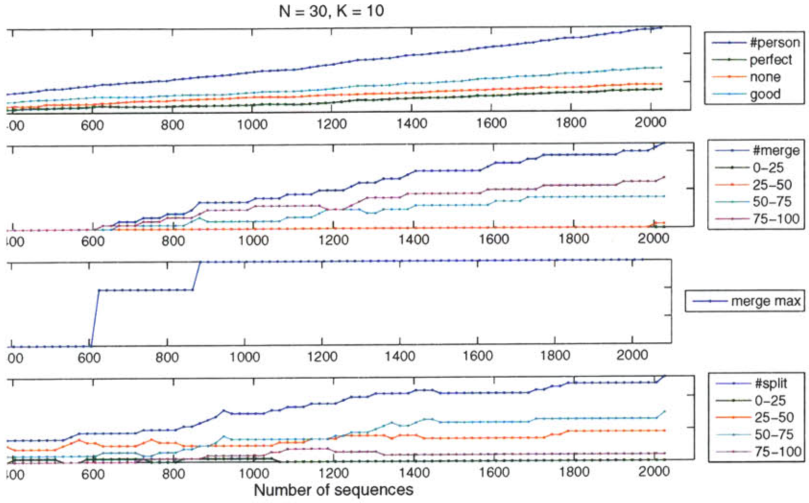


Figure 5-22: The self-generated clusters constructed during the incremental recognition process. These results were generated with $N = 30$, $K = 10$, and C is correlated with the data set size using the function *inc3* shown in figure 5-29. The results are plotted at every addition of 20 sequences, starting from the initial batch of 300 sequences.

The first one is the largest one of the four split clusters formed from the invididual represented by the largest pie chart. The second one is of a woman who came to interact with the robot on seven different days throughout the experiment. The red rectangle is placed to point out the falsely merged sequences from another individual in the cluster. The third one is a falsely merged cluster of many individuals' faces that were poorly segmented and happen to share a similar background. The fourth one is of a wall region which was falsely detected as a face by the face detector. We include more examples of these familiar individuals in appendix D.

5.4.4 Incremental Clustering Results Using Different Parameters

Figure 5-30 shows the incremental clustering results when using different correlation functions between the parameter C and data set size. We use six different correlation

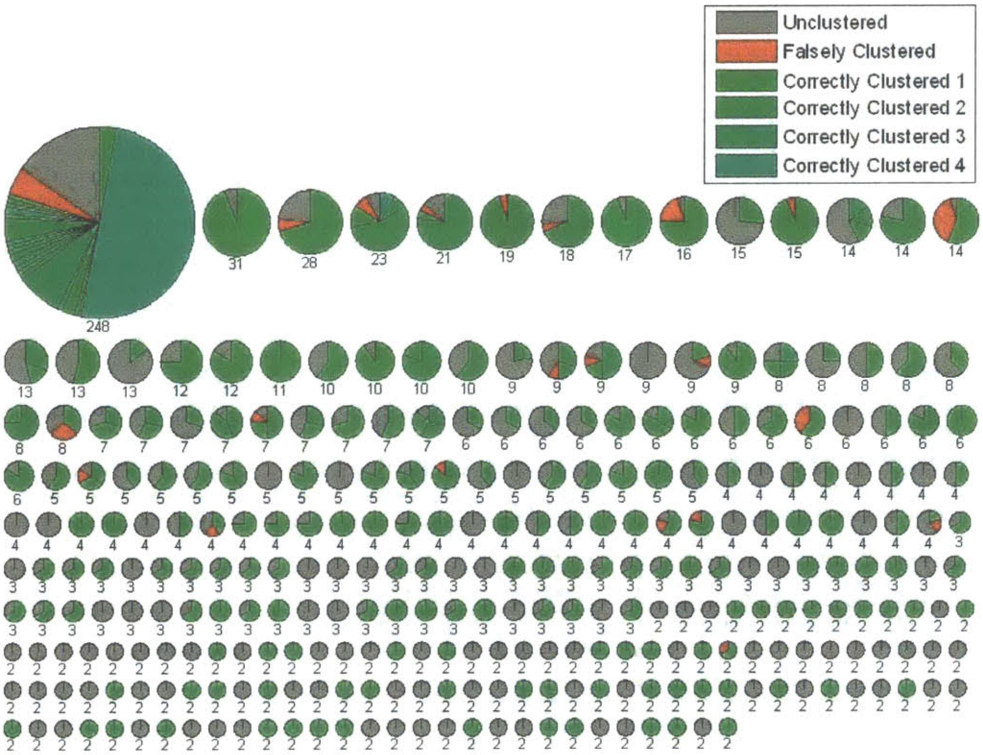


Figure 5-23: Visualization of the self-generated clusters. Each pie chart corresponds to an individual. The size of the pie chart circle corresponds to the number of sequences an individual have in the data set. The different shades of green regions correspond to sequences that were correctly clustered. Multiple green slices are present within a pie chart indicate that there is a splitting error. The red regions correspond to sequences that were falsely clustered, i.e. the merging errors. The gray regions correspond to unclustered sequences.

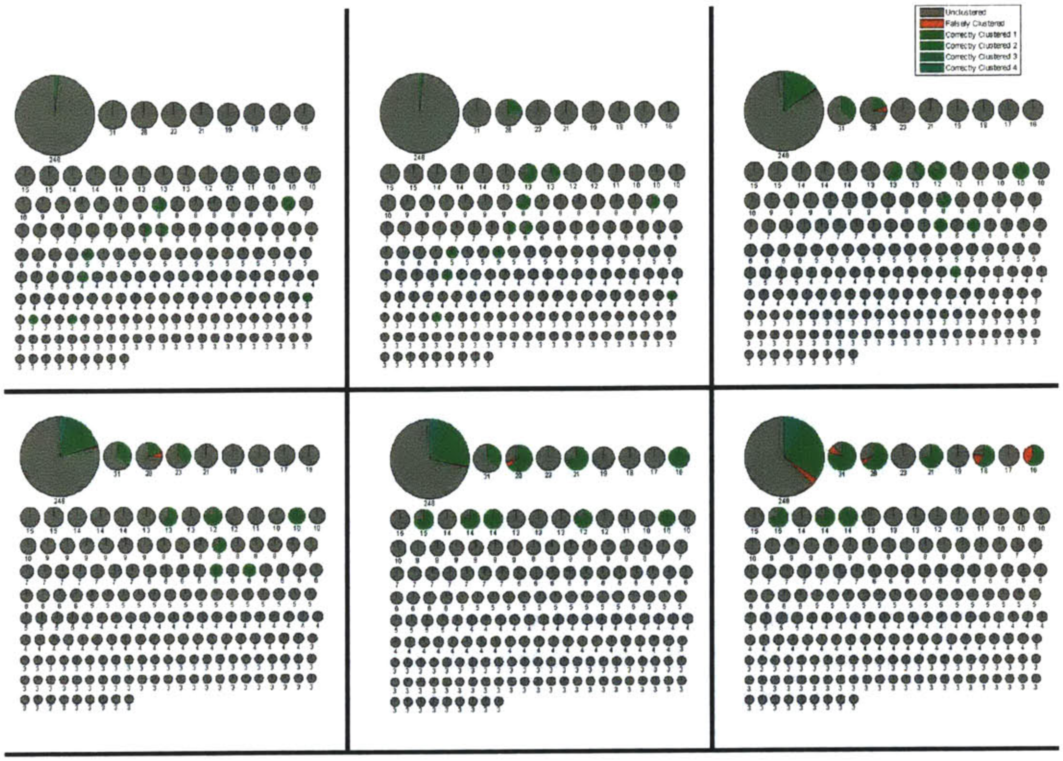


Figure 5-24: Six different snapshots of the largest fifteen self-generated clusters taken at different times during the incremental recognition process.

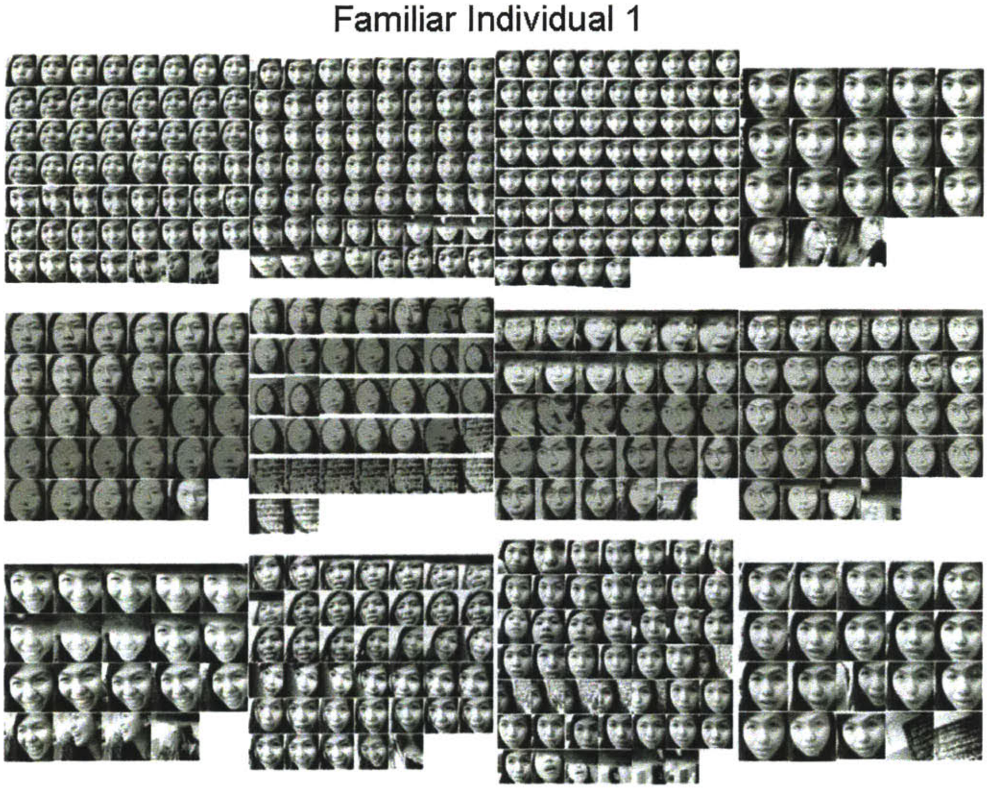


Figure 5-25: A sample cluster of a familiar individual, who was split among 4 clusters in the labelled training set. The other three clusters belonging to this individual are shown in figure ?? through ??.

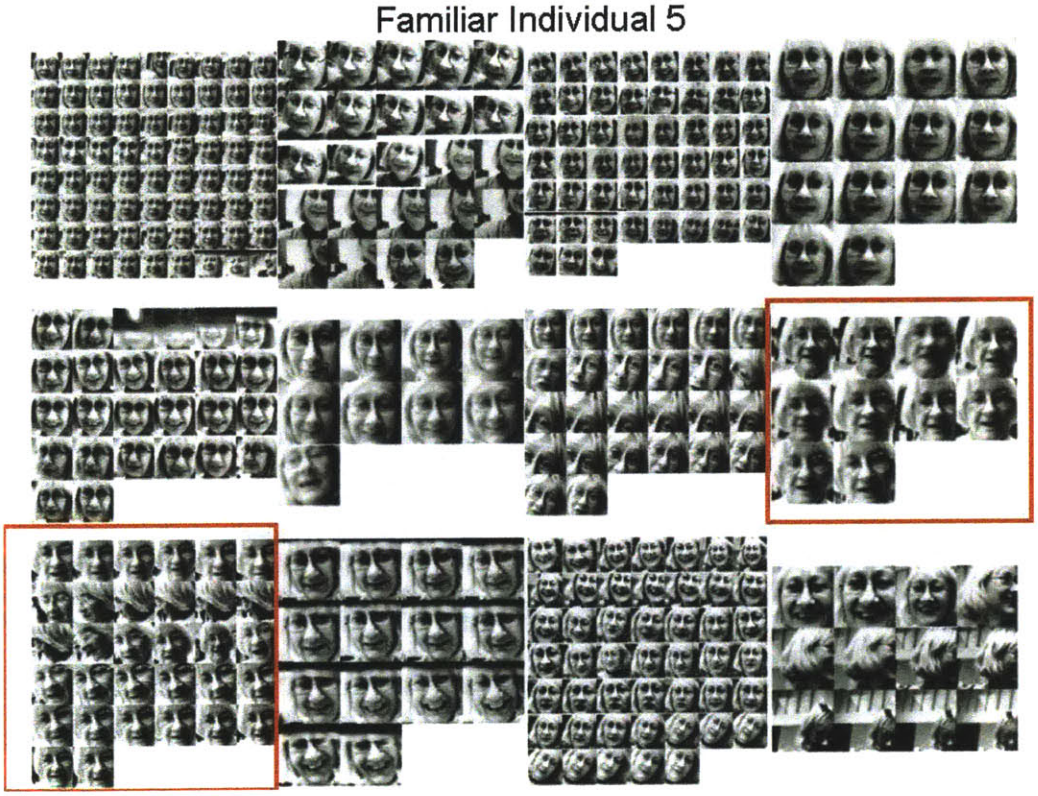


Figure 5-26: A sample cluster of a woman who came to interact with the robot on seven different days throughout the experiment. The red rectangle is placed to point out the falsely merged sequences from another individual in the cluster.

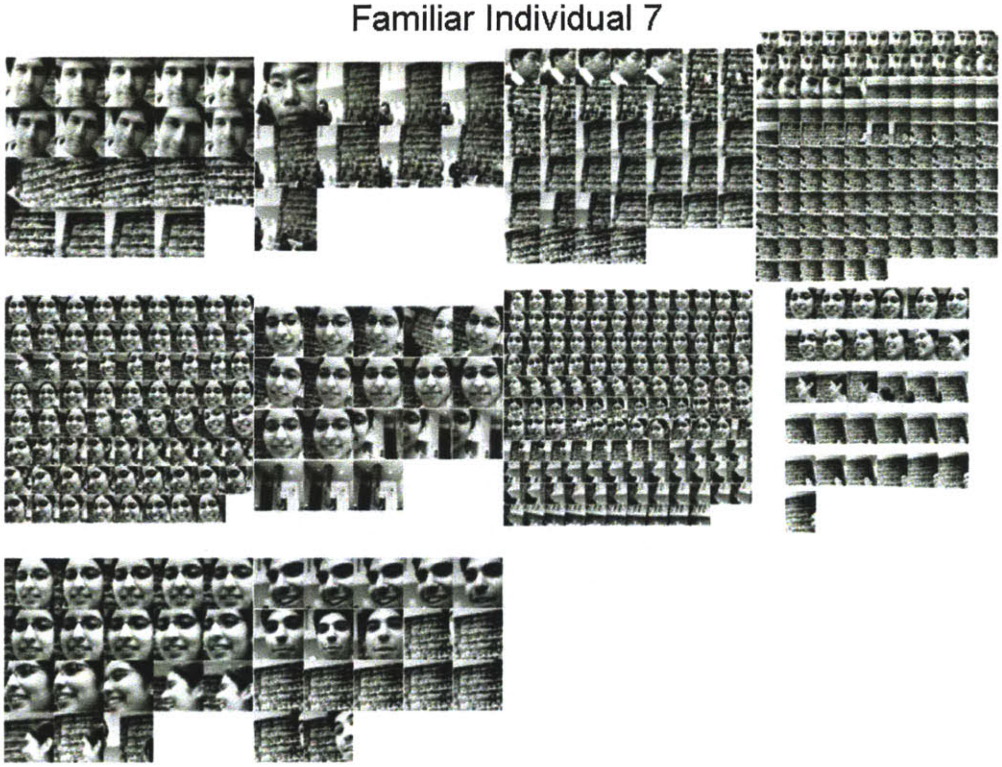


Figure 5-27: A sample cluster of a familiar individual, formed by a falsely merged cluster of many individuals' faces that were poorly segmented and happen to share a similar background.

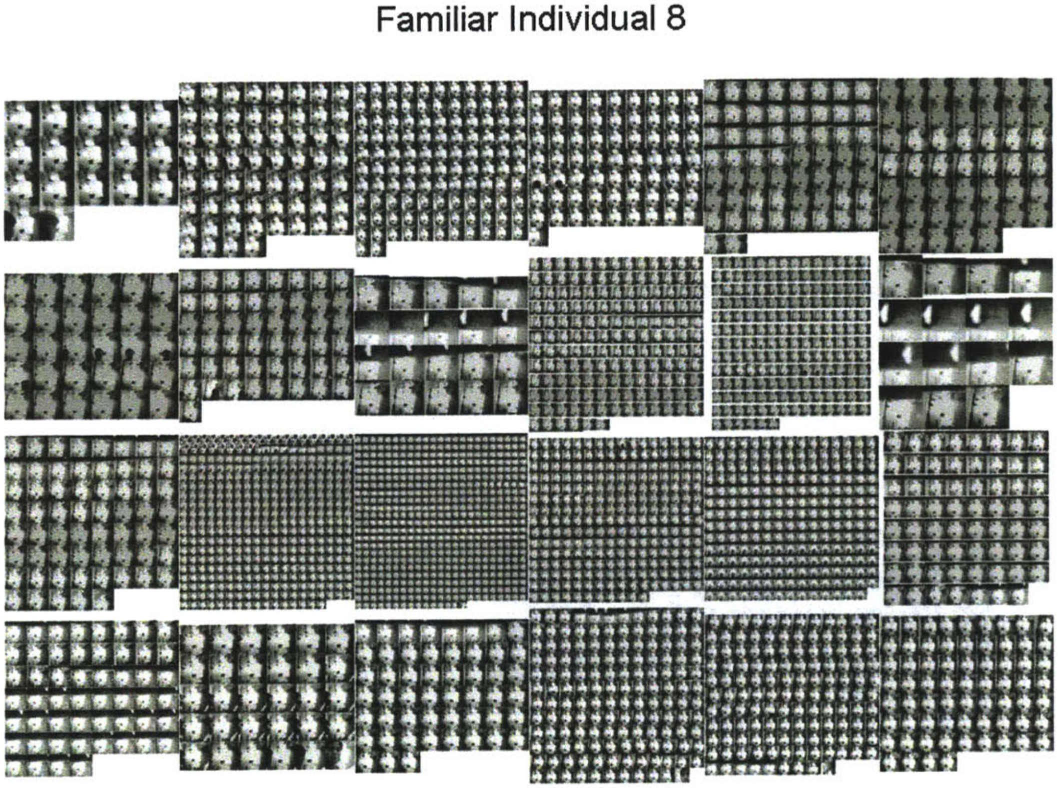


Figure 5-28: A sample cluster of a familiar individual, which is a wall region which was falsely detected as a face by the face detector.

functions, as shown in figure 5-29. As mentioned previously, the clustering system for the integrated incremental and unsupervised scheme was implemented using correlation function *inc3*.

Based on our previous clustering evaluations, we expect that the correlation function *inc1* will yield lower merging and higher splitting errors. On the other hand, we expect that the function *inc6* will yield the opposite case of higher merging and lower splitting errors. The results indeed confirm these expectations. However, the differences in performance across the six different correlation functions are not drastic. In general, the clustering system is capable of generating clusters with low errors for roughly half of the individuals in the data set.

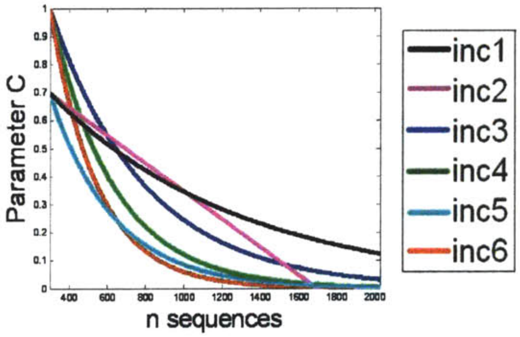
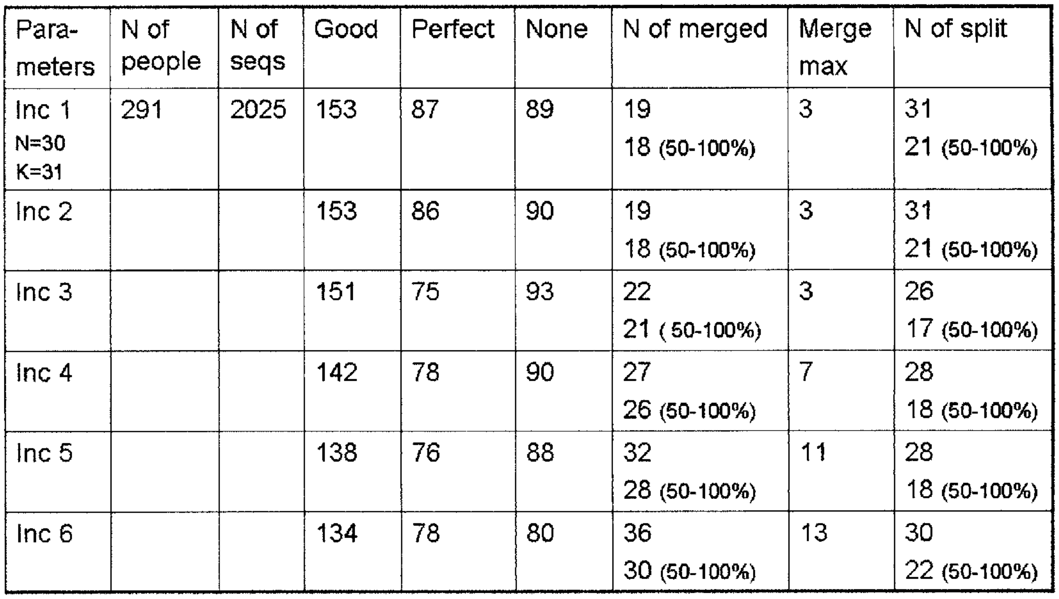


Figure 5-29: The set of correlation functions between the data set size and parameter C values used to evaluate the incremental clustering results.

5.5 Comparison to Related Work

In section 2.4, we describe a number of related research in unsupervised face recognition. For comparison purposes, we formulate our face clustering results to match the peformance metric used by Raytchev and Murase in [81] and Berg et al in [5], as shown in figure 5-31 and 5-32 respectively.

The comparison to Raycthev and Murase is more straight-forward as both approaches are video-based and thus deal with face sequences as input. The perfor-

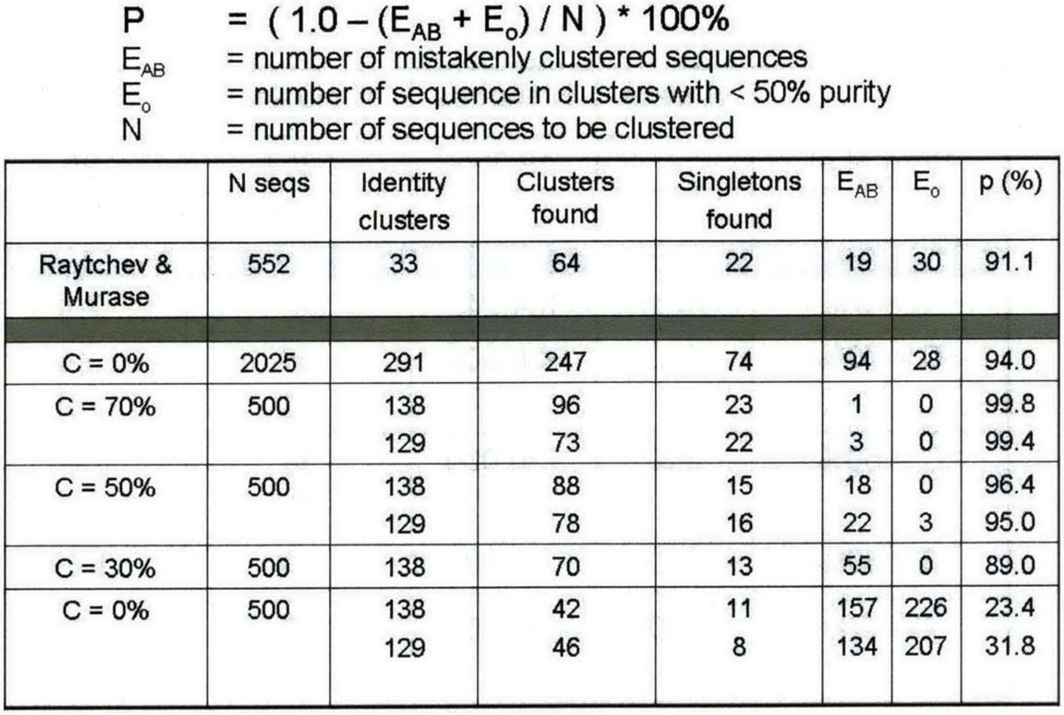


Para-meters	N of people	N of seqs	Good	Perfect	None	N of merged	Merge max	N of split
Inc 1 N=30 K=31	291	2025	153	87	89	19 18 (50-100%)	3	31 21 (50-100%)
Inc 2			153	86	90	19 18 (50-100%)	3	31 21 (50-100%)
Inc 3			151	75	93	22 21 (50-100%)	3	26 17 (50-100%)
Inc 4			142	78	90	27 26 (50-100%)	7	28 18 (50-100%)
Inc 5			138	76	88	32 28 (50-100%)	11	28 18 (50-100%)
Inc 6			134	78	80	36 30 (50-100%)	13	30 22 (50-100%)

Figure 5-30: The incremental clustering results using different correlation functions between the data set size and parameter C.

mance metric takes into account two types of errors: the number of mistakenly clustered sequences and the number of sequence in clusters with ¡ 50% purity. Using this performance metric, we present our results using data sets of 2025 and 500 sequences. For the latter, we display results using four different values of the C parameter, ranging from 0-70%, since we have observed that the smaller data set is more susceptible to a less conservative (lower) C parameter value and thus should use higher C values. For both data sets, our results are slightly better, except for when C is reduced to 30% or less.

The comparison to Berg et al is not as straight-forward, as their approach is image-based and the clustering is performed using face data along with captioned text information. The performance metric is the error rate of false cluster assignment. Our results are comparable when compared to their smaller data set and better when compared to their larger data set, except in case of when C is reduced to 0% for our data set of 500 sequences.



$P = (1.0 - (E_{AB} + E_o) / N) * 100\%$

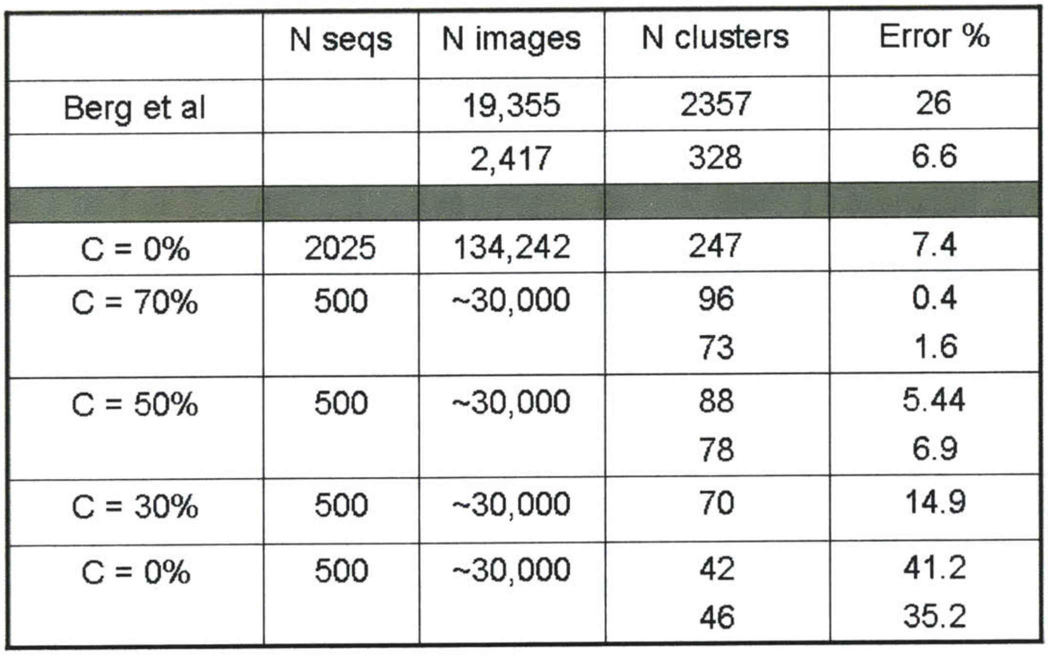
E_{AB} = number of mistakenly clustered sequences

E_o = number of sequence in clusters with < 50% purity

N = number of sequences to be clustered

	N seqs	Identity clusters	Clusters found	Singletons found	E_{AB}	E_o	p (%)
Raytchev & Murase	552	33	64	22	19	30	91.1
C = 0%	2025	291	247	74	94	28	94.0
C = 70%	500	138	96	23	1	0	99.8
		129	73	22	3	0	99.4
C = 50%	500	138	88	15	18	0	96.4
		129	78	16	22	3	95.0
C = 30%	500	138	70	13	55	0	89.0
C = 0%	500	138	42	11	157	226	23.4
		129	46	8	134	207	31.8

Figure 5-31: Comparison to Raytchev and Murase's unsupervised video-based face recognition system.



	N seqs	N images	N clusters	Error %
Berg et al		19,355	2357	26
		2,417	328	6.6
C = 0%	2025	134,242	247	7.4
C = 70%	500	~30,000	96 73	0.4 1.6
C = 50%	500	~30,000	88 78	5.44 6.9
C = 30%	500	~30,000	70	14.9
C = 0%	500	~30,000	42 46	41.2 35.2

Figure 5-32: Comparison to Berg et al's unsupervised clustering of face images and captioned text.

5.6 Discussion

We have evaluated the unsupervised face clustering system by itself. We have also evaluated an integrated system that uses the face clustering solution to incrementally build a labelled training set and use it to perform supervised recognition.

The presented solutions and results indicate promising steps for an unsupervised and incremental face recognition system. The face clustering algorithm yields good performance despite the extremely noisy data automatically collected and segmented by the robot through spontaneous interactions with many passersby in a difficult public environment. The face data contains a large number of poorly segmented faces, faces of varying poses and facial expressions, and even non-face images. Moreover, the face clustering algorithm is more robust to merging errors when more data is available. For larger data sets, the face clustering algorithm generated stable performance across a wide range of parameter settings.

The current implementation of the face clustering system and its supervised variant has not been optimized for speed. For each face sequence input, it currently takes 5-15 minutes to process, extract features, and determine where it should be placed among the existing clusters. The supervised recognition process currently takes 1-2 seconds per face image. The next research step would be to optimize the computational speed of these two systems, particularly the supervised recognition process.

The first most obvious candidate approach for optimization is in the feature representation of the face sequences. We currently use 50 SIFT feature prototypes per region to represent each face sequence. Most likely, some of these prototypes are very useful and some are probably irrelevant. Thus, pruning these feature prototypes will not only increase computational speed, but also improve the clustering algorithm. Moreover, when a set of face sequences are combined into a cluster, we currently retain all of their features to represent a class or person. In cases where a person's cluster has a large number of sequences, we prune them by selecting those which have been most frequently selected as a match throughout the incremental clustering process. This pruning process of sequences within a cluster can be performed more efficiently. The combination of pruning feature prototypes within a face sequence and pruning face sequences within a cluster would increase the computational speed of both the face clustering system and its supervised variant.

Eventually, we believe that an ultimate unsupervised recognition sytem for a home robot, that is fully robust and capable of learning to recognize people in any environmental settings, would require additional perceptual cues and contextual information. Thus, the learning system can essentially combine different sources of information to make a robust unsupervised decision. These additional information may be in the form of a multi-modal perceptual inputs, associated information such as names, or a reward signal. The use of this coupling of information was already explored in by Berg et al [5]. The robot can also actively acquire these additional information by using its social interface, e.g. by verbally asking for people's names or for confirmation of one's identity, to assist its learning task.

Chapter 6

Conclusion

We present an integrated end-to-end incremental and fully unsupervised face recognition framework within a robotic platform embedded in real human environment. The robot autonomously detects, tracks, and segments face images during spontaneous interactions with many passersby in public spaces and automatically generates a training set for its face recognition system.

We present the robot implementation and its unsupervised incremental face recognition framework. We demonstrate the robot's capabilities and limitations in a series of experiments at a public lobby. In a final experiment, the robot interacted with a few hundred individuals in an eight day period and generated a data set of over a hundred thousand face images.

We describe an algorithm for clustering local features extracted from a large set of automatically generated face data. This algorithm is based on a face sequence matching algorithm which shows robust performance despite the noisy data. We evaluate the clustering algorithm performance across a range of parameters on this automatically generated training data and also the Honda-UCSD video face database.

Using the face clustering solution, we implemented an integrated system for unsupervised and incremental face recognition. We evaluated this system using the face data automatically generated by the robot during the final experiment. During this incremental clustering and recognition test, the system builds a labelled training set in a fully unsupervised way and then uses this training set to recognize each sequence input. The system hypothesizes correctly 74% of the time that the sequence input belongs to an unknown person who does not exist in the training set. After an initial learning period, when the system hypothesizes that the sequence input belongs to a specific known person, it is correct roughly 80% of the time.

6.1 Lessons in Extending Robustness

In this thesis, we learned a number of the lessons and identified a set of challenges during the consequential process of extending the space and time in which the robot operates. Although it is still difficult for humanoid robots to operate robustly in noisy environment, the issue of robustness has not received adequate attention in most research projects 3. Since robots will ultimately have to operate beyond the scope of short video clips and end-of-project demonstrations, we believe that a better understanding of these challenges is valuable for motivating further work in various areas contributing to this interdisciplinary endeavor.

Perception has been blamed to be one of the biggest hurdles in robotics and certainly has posed many difficulties in our case. We generally found that many existing vision and speech technology are not suitable for our setting and constraints. Vision algorithms for static cameras are unusable because both cameras pan independently. The desktop microphone required for natural interaction with multiple people generates decreased performance compared to the headset typically used for speech recognition. Drastic lighting changes inside the building and conducting experiments in different locations have forced us to go through many iterations of the robot's perceptual systems. Something that works in the morning at the laboratory may no longer work in the evening or at another location. Many automatic adaptive mechanisms, such as for the camera's internal parameters to deal with lighting changes throughout the day, are now necessary.

For a robotic creature that continuously learns while living in its environment, there is no separation between the learning and testing periods. The two are blurred together and often occurring in parallel. Mertz has to continually locate learning targets and carefully observe to learn about them. These two tasks are conflicting in many ways. The perceptual system is thus divided between fast but less precise processes for the first task and slower but more accurate algorithms for the latter. Similarly, the attention system has to balance between being reactive to new salient stimuli and persistent to observe current learning target. This dichotomy is interestingly reflected in the what and where pathways of our visual cortex, as well as the endogenous and exogenous control of visual attention.

Humans' tendency to anthropomorphize generally makes the robot's task to socially interact simpler. However, requiring the robot to interact with multiple people for an extended duration has called for a more sophisticated social interface. One can imagine that a friendly robot that makes eye contact and mimics your speech can be quite entertaining, but not for too long. While the premise that social interaction can guide robot learning is promising, it also suffers from the "chicken and egg" problem in a long-term setting, i.e. in order to sustain an extended interaction, the robot also needs to be able to learn and retain memory of past events.

In all engineering disciplines, we tend to focus on maximizing task performance. Whenever people are present, Mertz's task is to detect and engage them in interaction. We learned that when the robot is on all the time, in addition to its tasks, the robot also has to deal with down time, when no one is around. All of a sudden the environment's background and false positive detection errors become a big issue. During an experiment session, the robot kept falsely detecting a face on the ceiling, stared at it all day, and ignored everything else. Lastly, as the software complexity grows, the harder it becomes to keep the entire system running for many hours. Memory leaks and infrequent bugs emerging from subsystem interactions are very difficult to track. Moreover, a robot that runs for many hours per day and learns from its experiences can easily generate hundreds of gigabytes of data. While having a lot of data is undeniably useful, figuring out how to automatically filter, store, and retrieve them in real time is an engineering feat.

6.2 Future Work

As we discussed in section 5.6, the next research step would be to optimize the computational speed of these two systems, particularly the supervised recognition process. This would allow for an online evaluation of the integrated incremental face recognition system. We have also suggested some possible optimization steps.

A natural extension to this thesis is to integrate voice recognition. The robot's spatio-temporal learning mechanism currently allows the robot to generate and segment not only face sequences, but also associated voice samples from the corresponding individual. Integration of voice recognition can assist both the clustering and recognition process. Berg et al showed that additional information given by a set of names extracted from captioned text can assist in improving the accuracy of clustering of face images from an unlabelled data set of captioned news images. Similarly, additional information given by voice recognition can be used to help the face clustering process. Moreover, during the supervised recognition part, an additional source of recognition hypothesis will be very useful.

Face, especially when segmented without any hair, provides very limited information for individual recognition. Further additional information will be in fact necessary, as we discussed in section 5.6, to achieve an ultimate incremental individual recognition that is fully robust. These additional information can come from other visual cues, multi-modal signals, associated features such as people's names, contextual information, and active learning. What we mean by active learning is to utilize the robot's social behavior to actively inquire specific individuals for information to assist its learning task. For example, the robot may ask someone if the robot has ever seen them before. Alternatively, the robot may ask someone to double check if he is in fact who the robot thinks he is.

Lastly, it would be very interesting to take the next step of closing the loop from the recognition output to the robot's behavior system. This would allow for a social recognition mechanism, where the robot can not only learn to recognize people, but also to adapt its behavior based on the robot's previous experience with specific individuals.

Appendix A

Experiment Protocol

- The content of a large poster placed in the front of the robot platform during the experiment:

  Experiment Notice

  Hello, my name is Mertz. Please interact with me. I am trying to learn to recognize different people.

  This is an experiment to study human-robot interaction. The robot is collecting data to learn from its experience.

  Please be aware that the robot is recording both visual and audio input.

- The content of a small poster placed on the robot platform during the experiment

  Hello my name is Mertz. Please interact and speak to me. I am trying to learn to recognize different people.

  I can see and hear. I may ask you to come closer because I cannot see very far.

  I don't understand any language, but I am trying to learn and repeat simple words that you say.

- The content of email sent to the recruited subjects

Thank you again for agreeing to participate in this experiment. The robot's schedule and location will posted each day at:

http://people.csail.mit.edu/lijin/schedule.html

Tomorrow (monday nov 22), the robot will be at the student street, 1st floor, from 12 - 7 pm. The first couple days will be 'test runs', so I apologize if you may find the robot not working properly or being repaired.

With the exception of the first day, the robot will generally be running from 8.30am - 7pm, at either the 1st or 4th floor of Stata. Please feel free to come on whichever days and times that work best with your schedule and travel plans. It would be great if the robot can see and talk to you on multiple occasions during the second week (from Monday Nov 27 onward).

Some written information about the experiment will be posted near the robot, in order to ensure that everyone including the passersby receives the same instructions.

If you prefer that I send you an email for each schedule update (instead of looking up online) or have any questions/comments, please let me know.

Appendix B

Sample Face Clusters

The following are a set of sample face clusters, formed by the unsupervised clustering algorithm. See Chapter 5 for the implementation and evaluation of the clustering system.

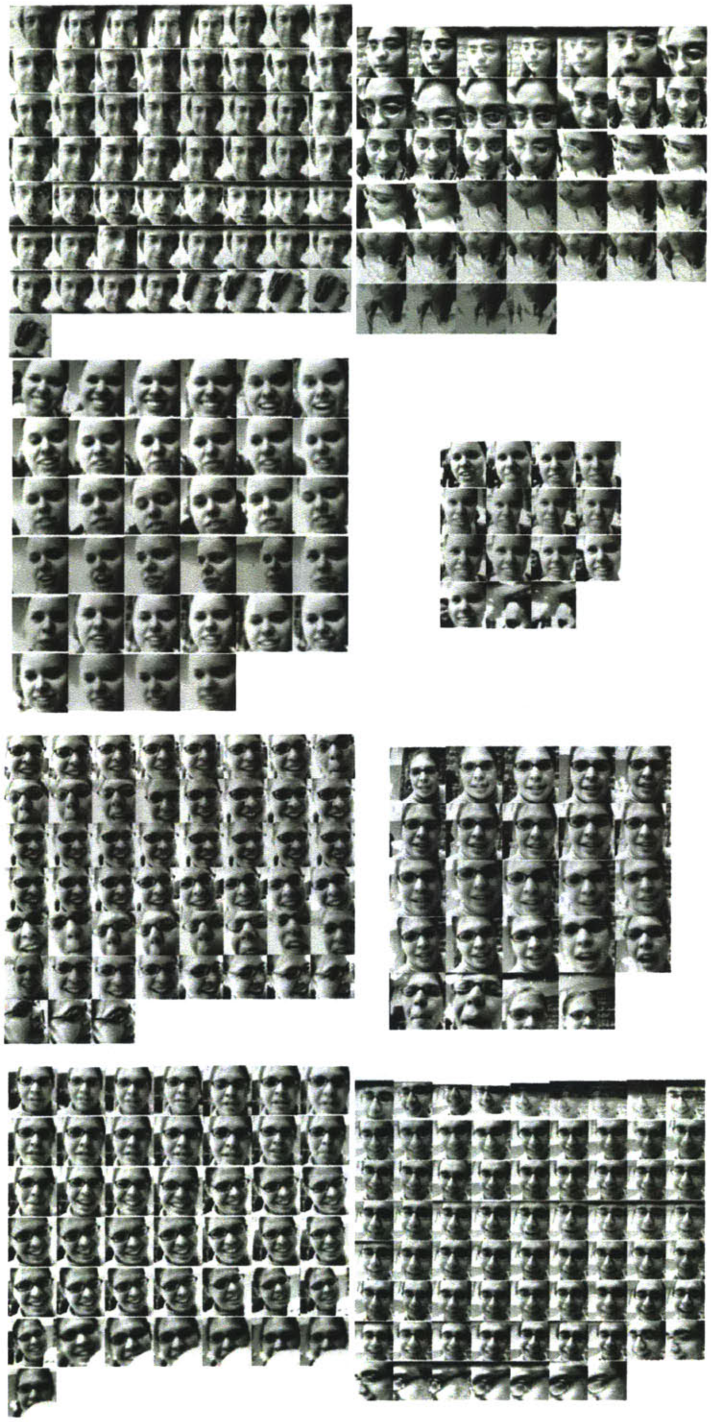


Figure B-1: An example of a falsely merged face cluster.

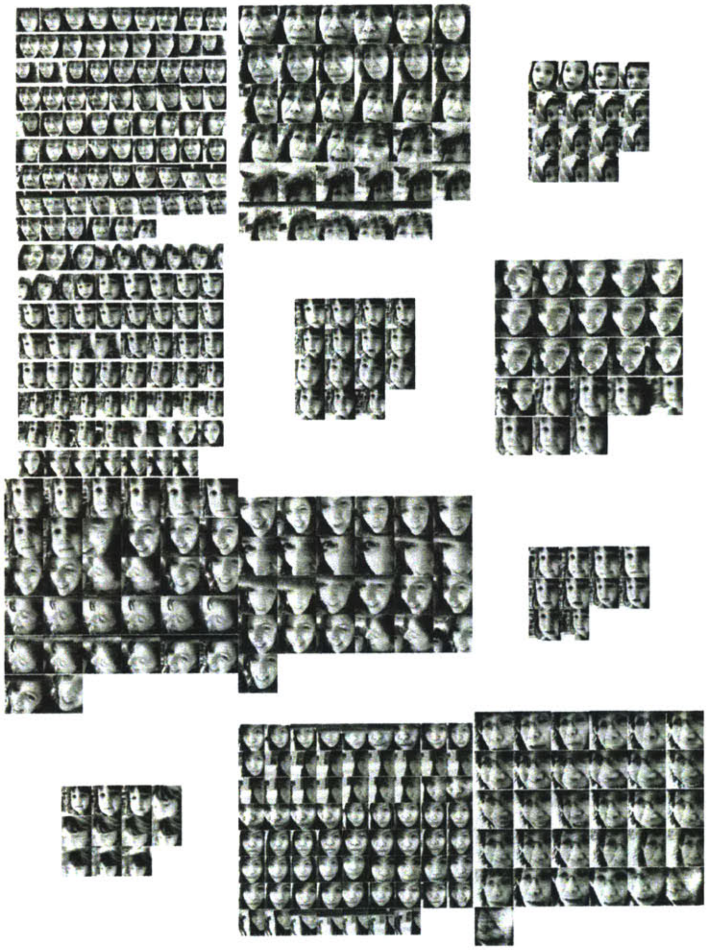


Figure B-2: An example of a falsely merged face cluster.

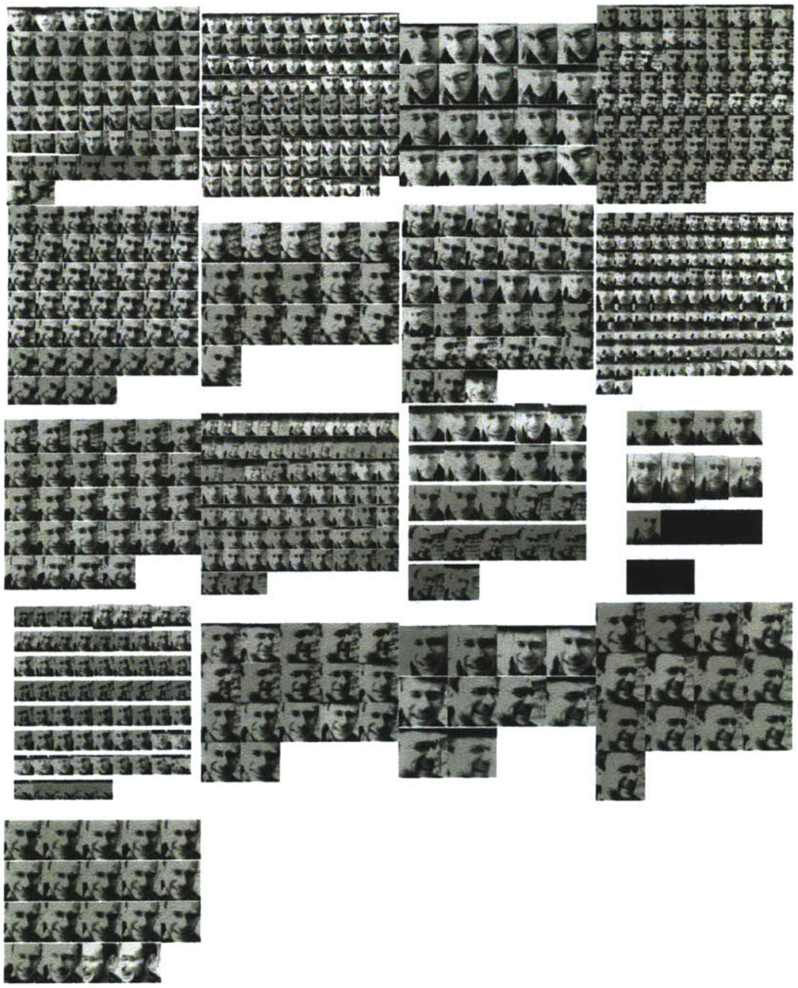


Figure B-3: An example of a good face cluster containing sequences from multiple days.

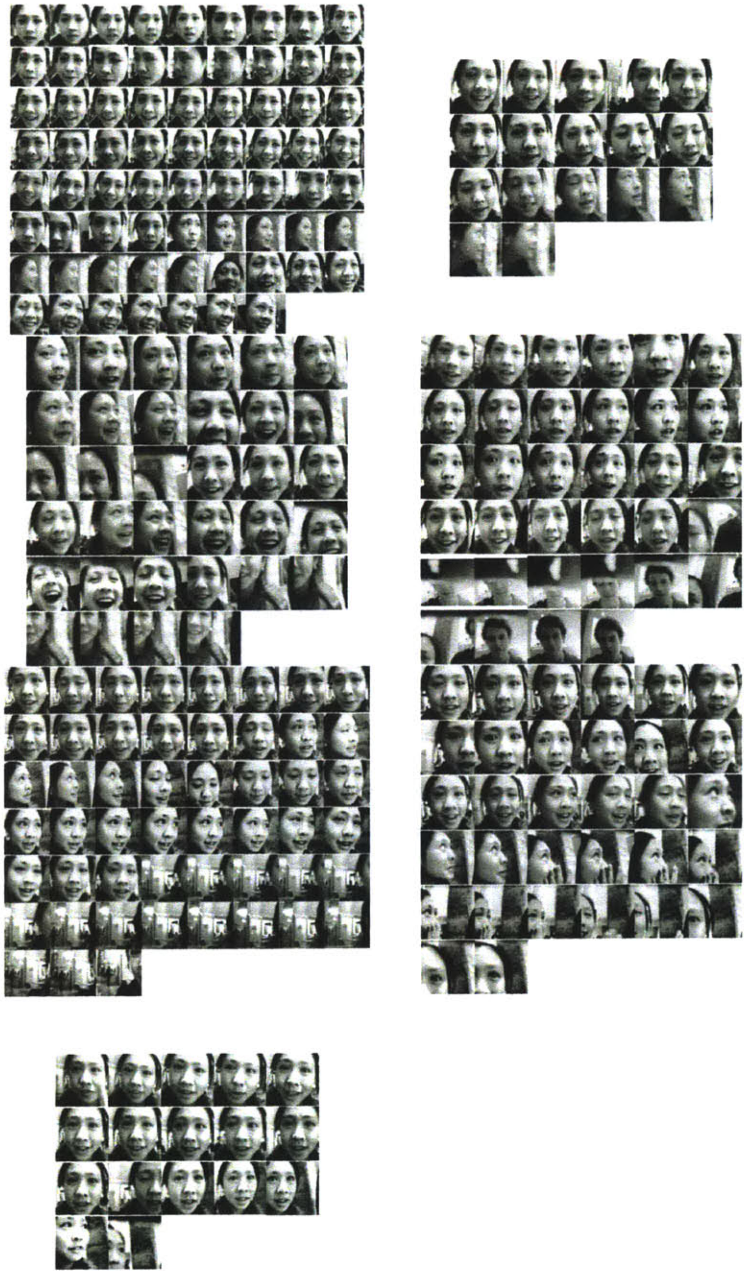


Figure B-4: An example of a good face cluster.

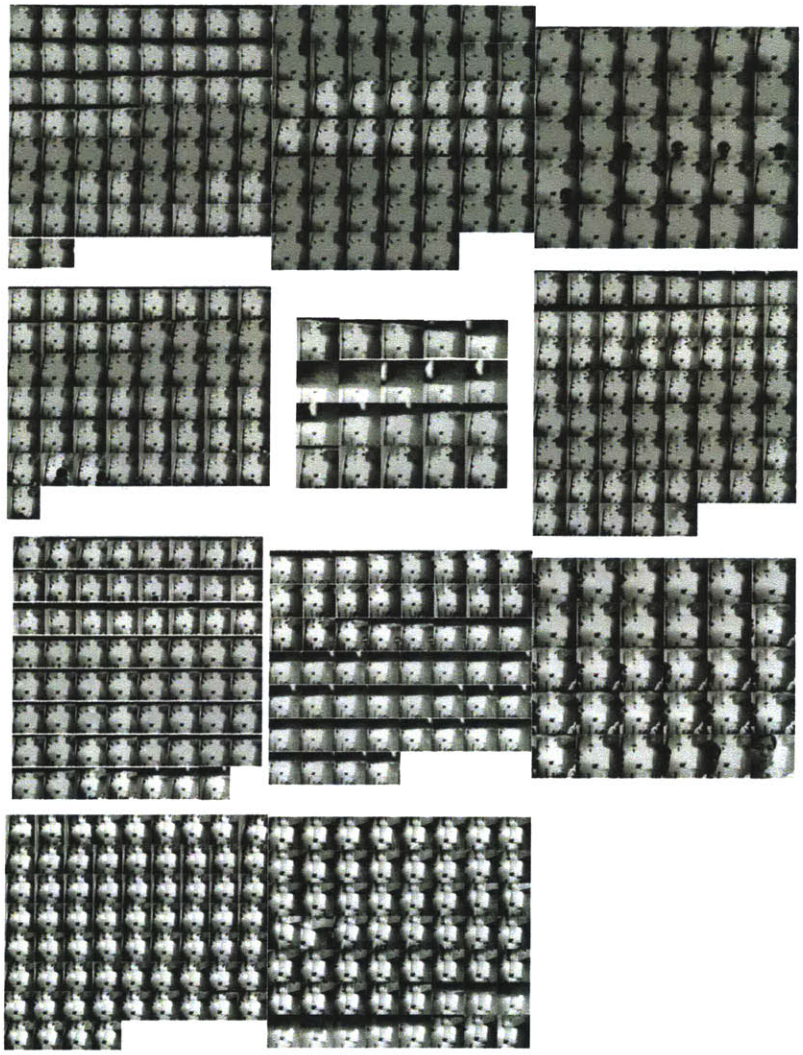


Figure B-5: An example of a non-face cluster.

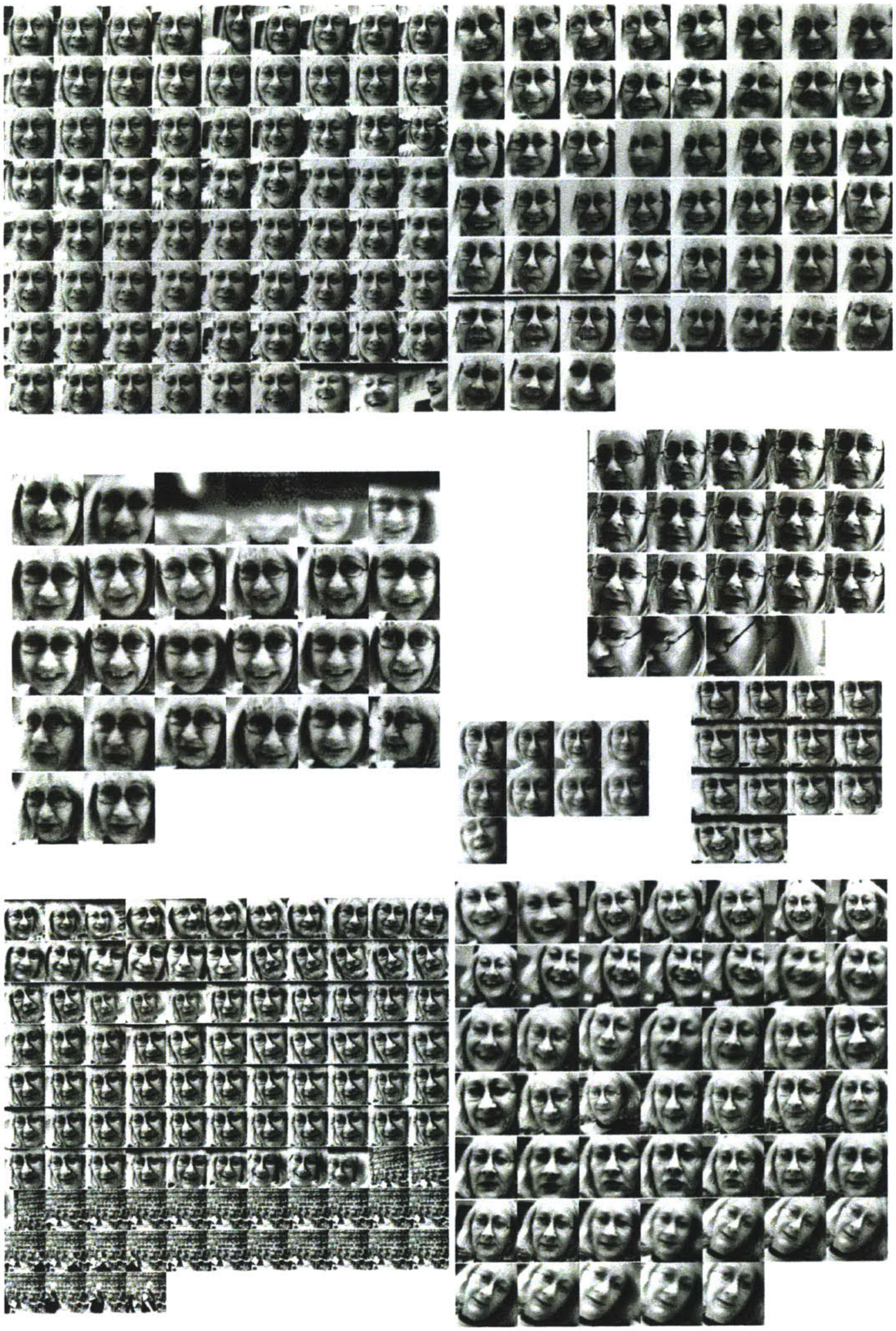


Figure B-6: An example of a good face cluster containing sequences from multiple days.

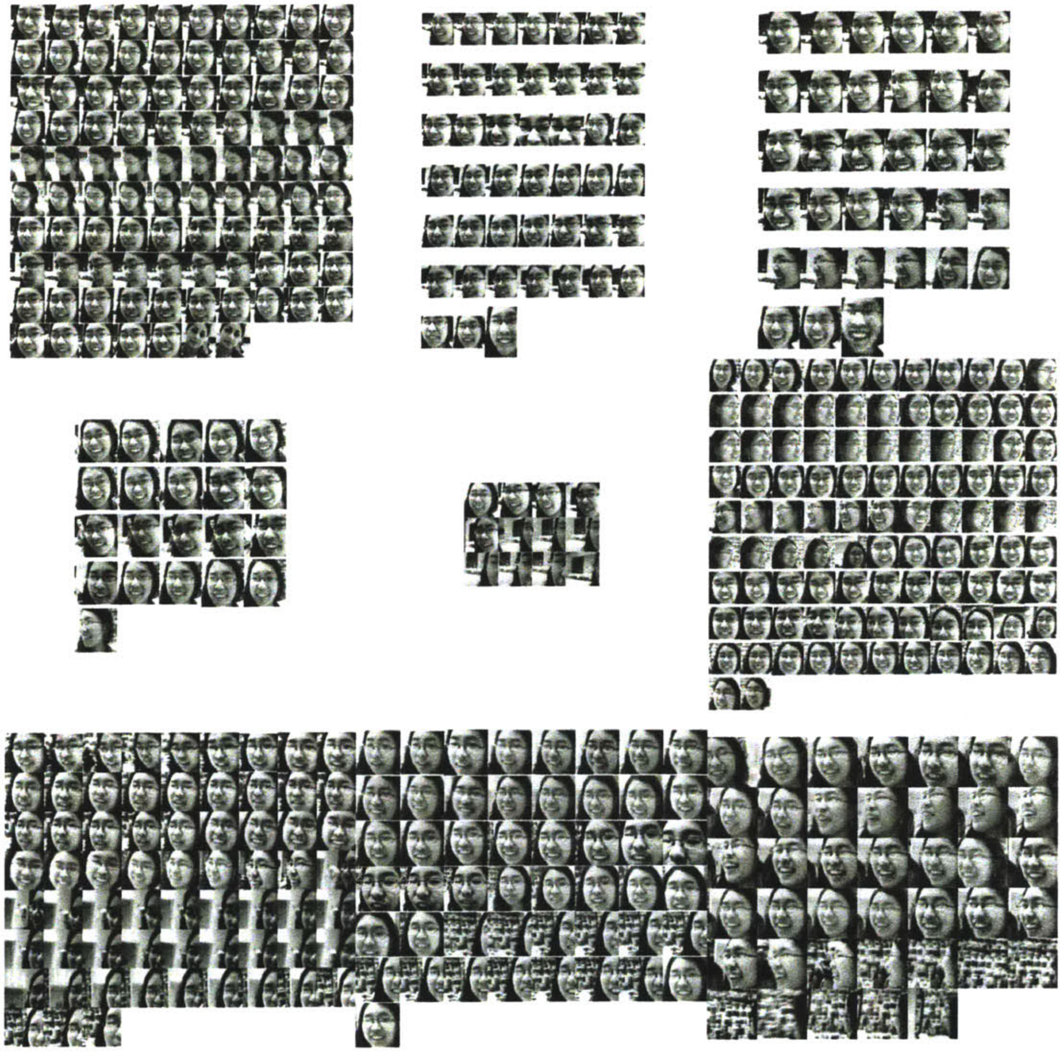


Figure B-7: An example of a good face cluster

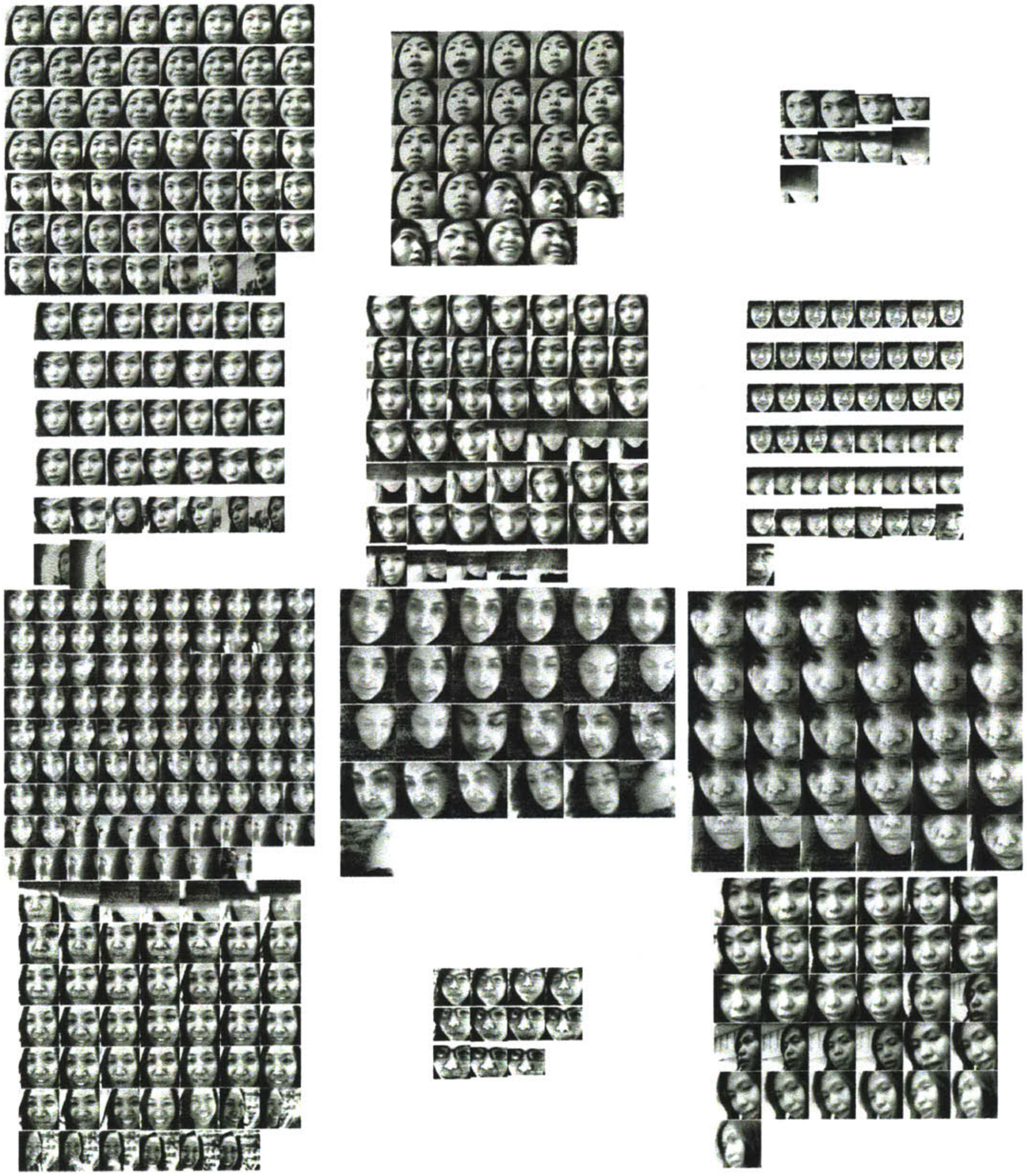


Figure B-8: An example of a falsely merged face cluster.

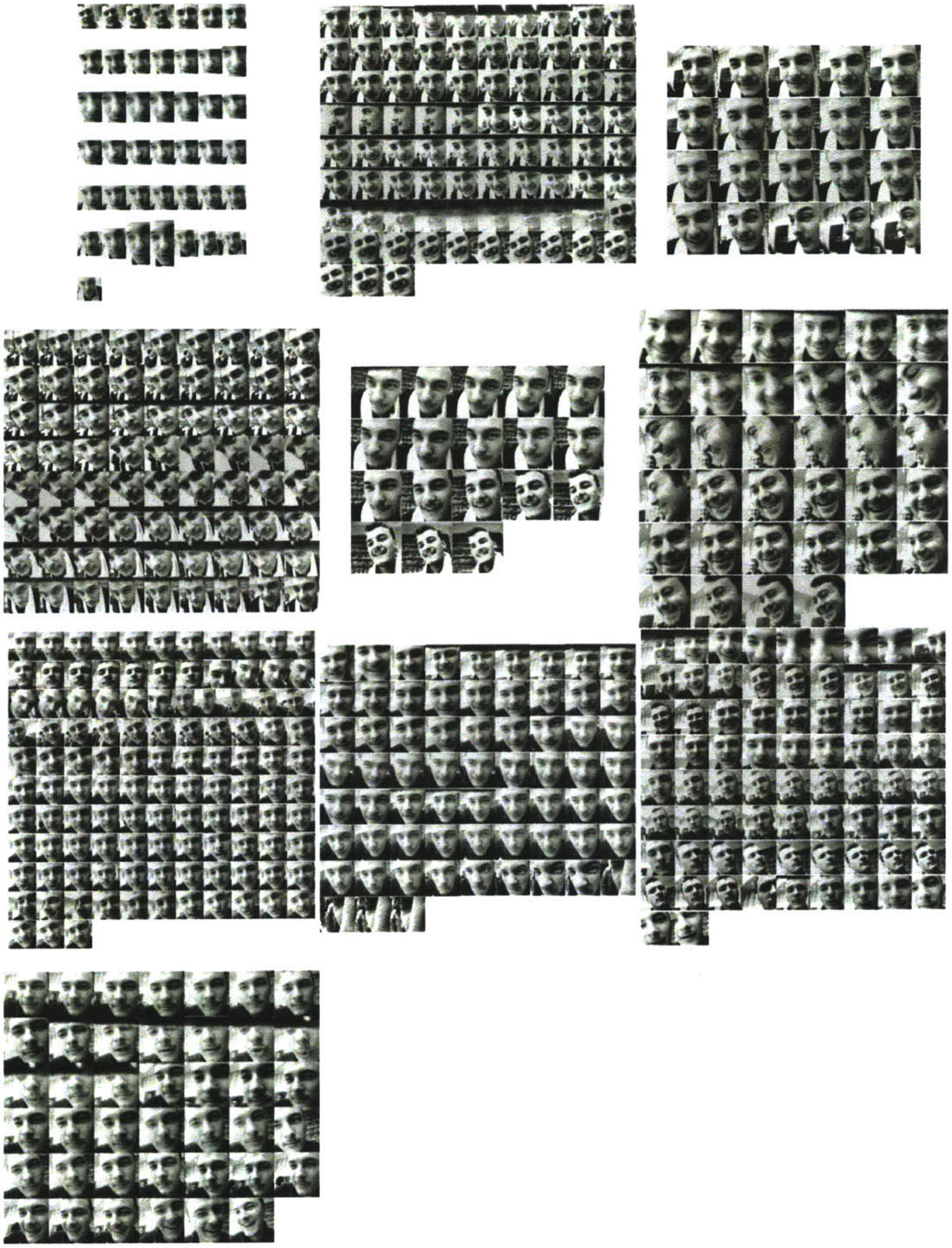


Figure B-9: An example of a good face cluster containing sequences from multiple days.

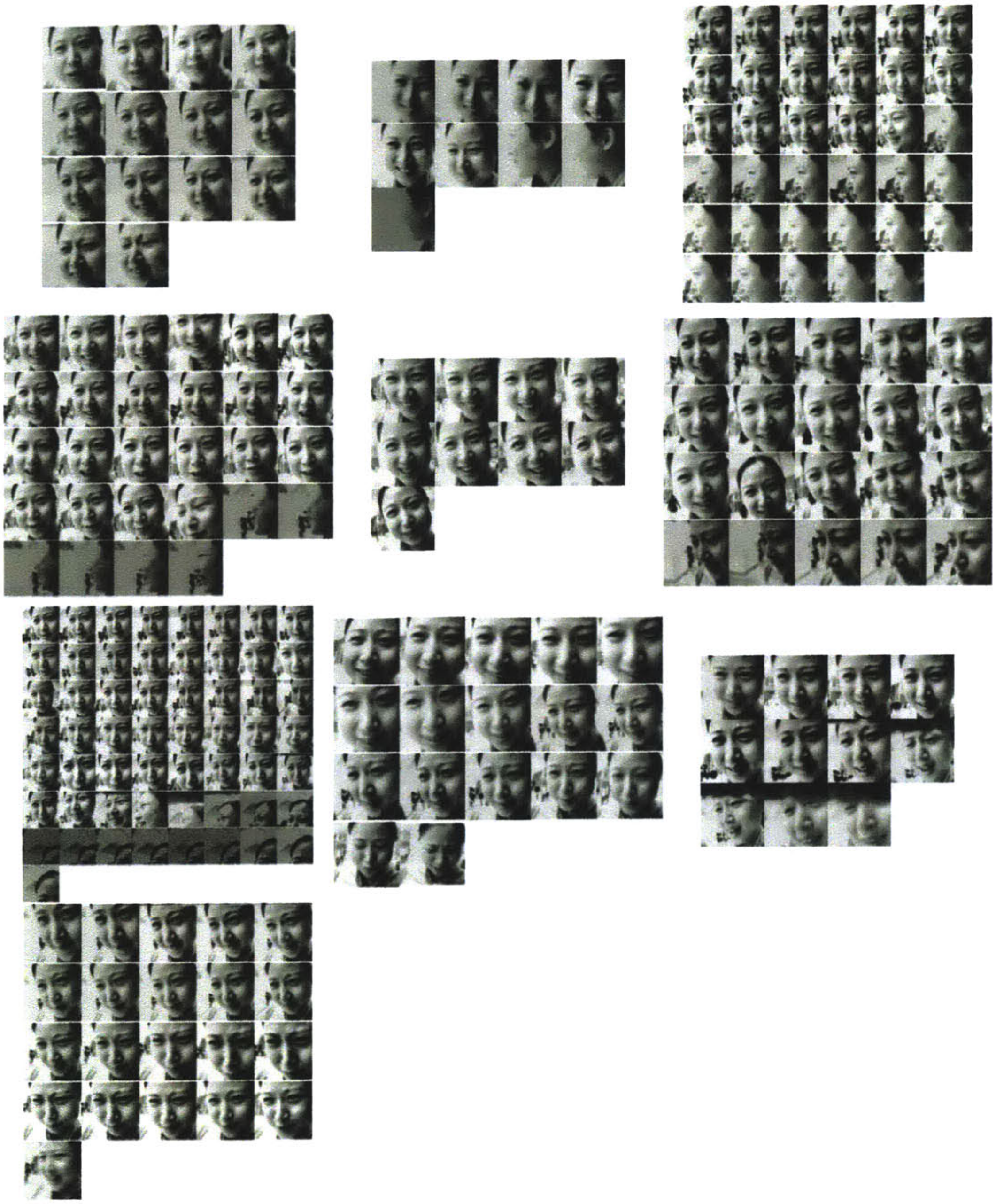


Figure B-10: An example of a good face cluster.

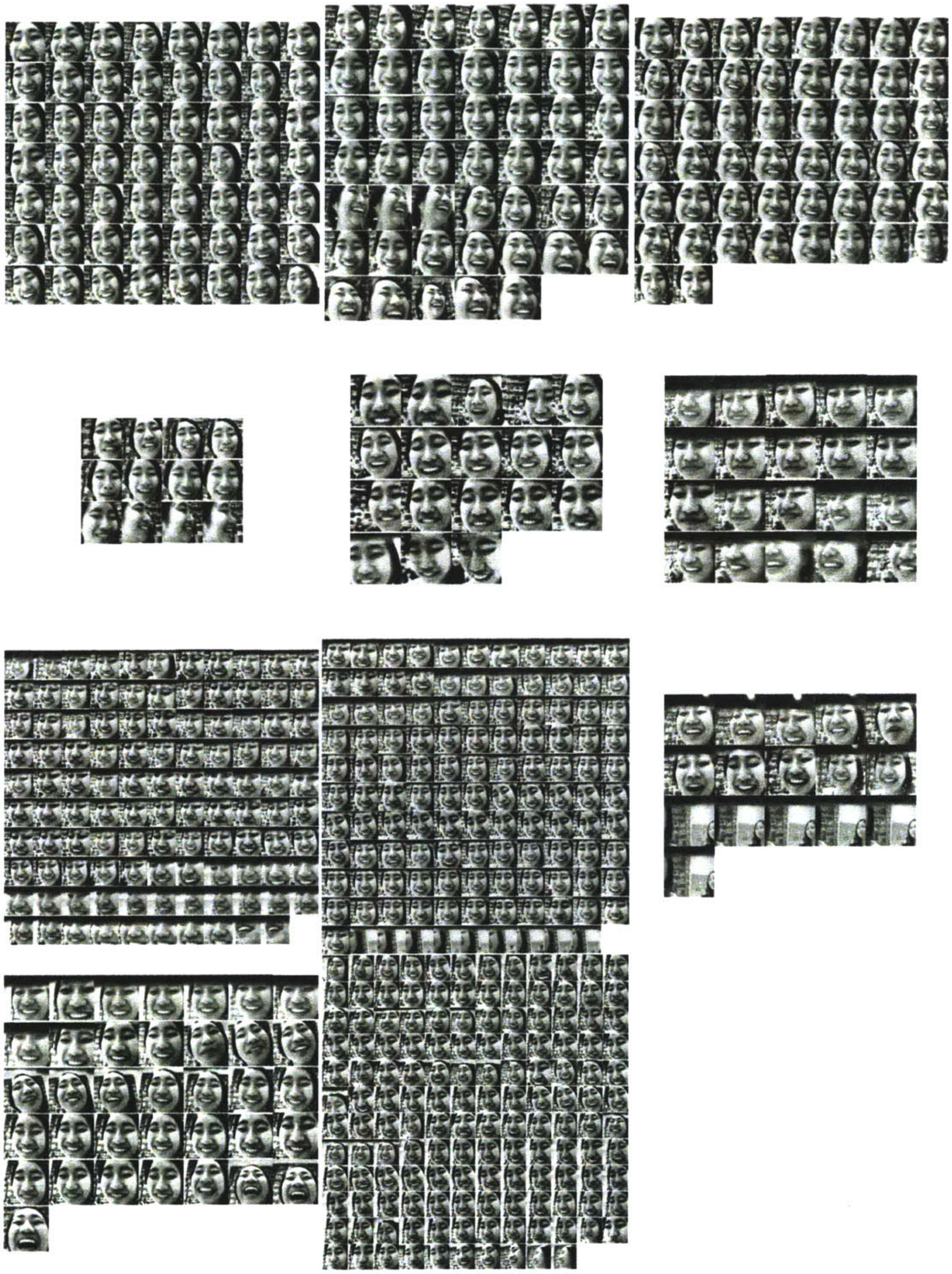


Figure B-11: An example of a good face cluster.

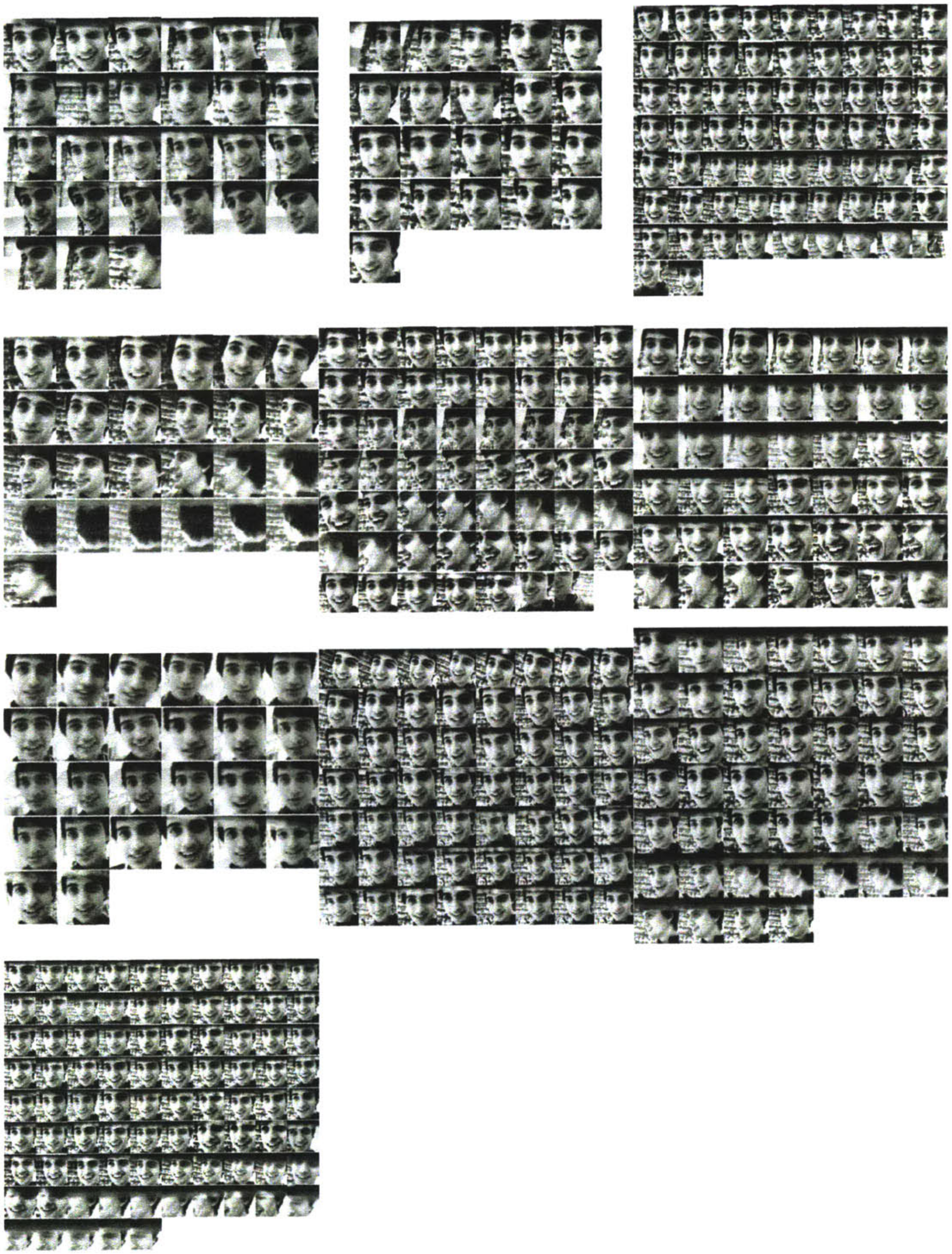


Figure B-12: An example of a good face cluster.

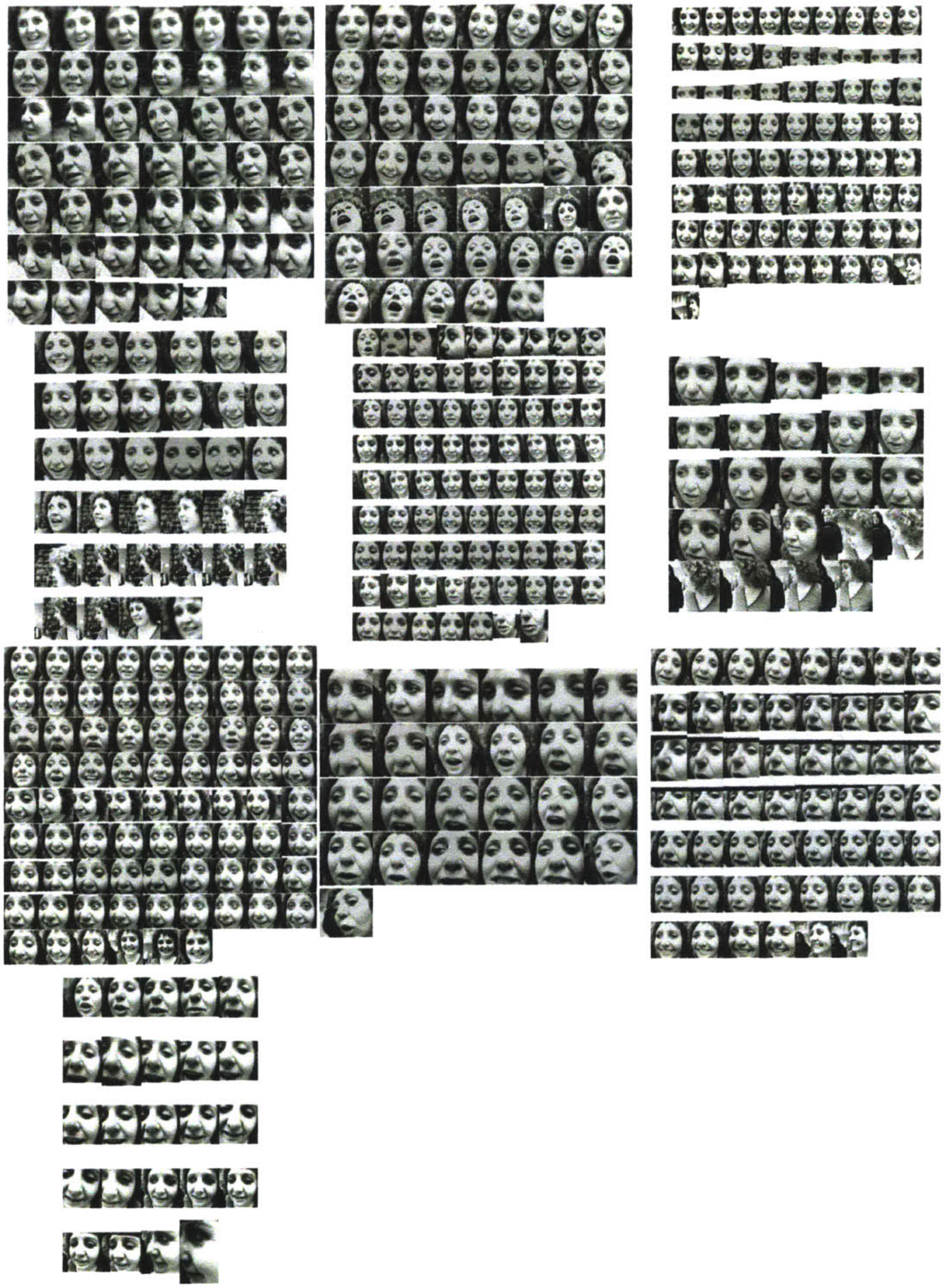


Figure B-13: An example of a good face cluster.

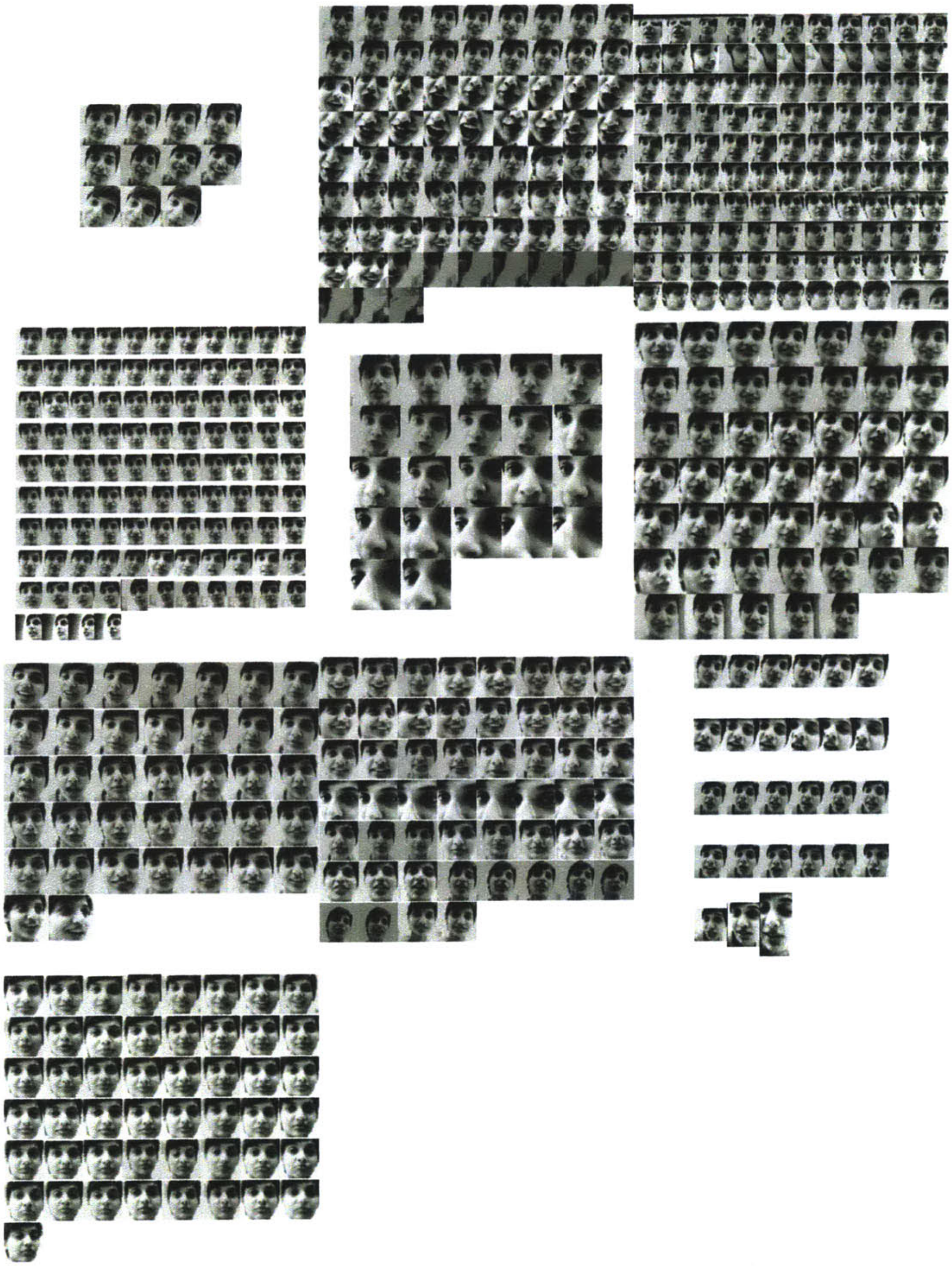


Figure B-14: An example of a good face cluster.

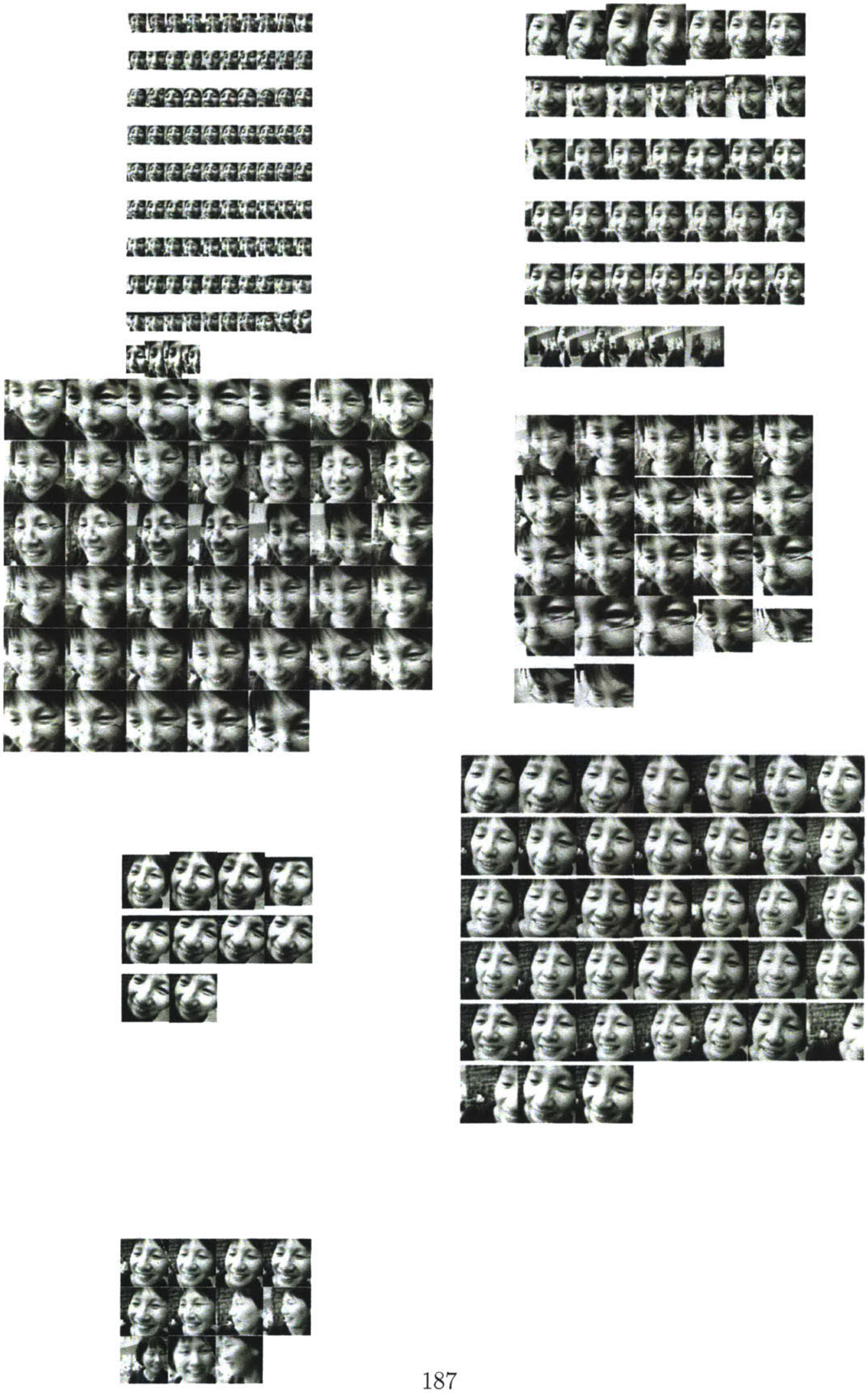


Figure B-15: An example of a good face cluster.

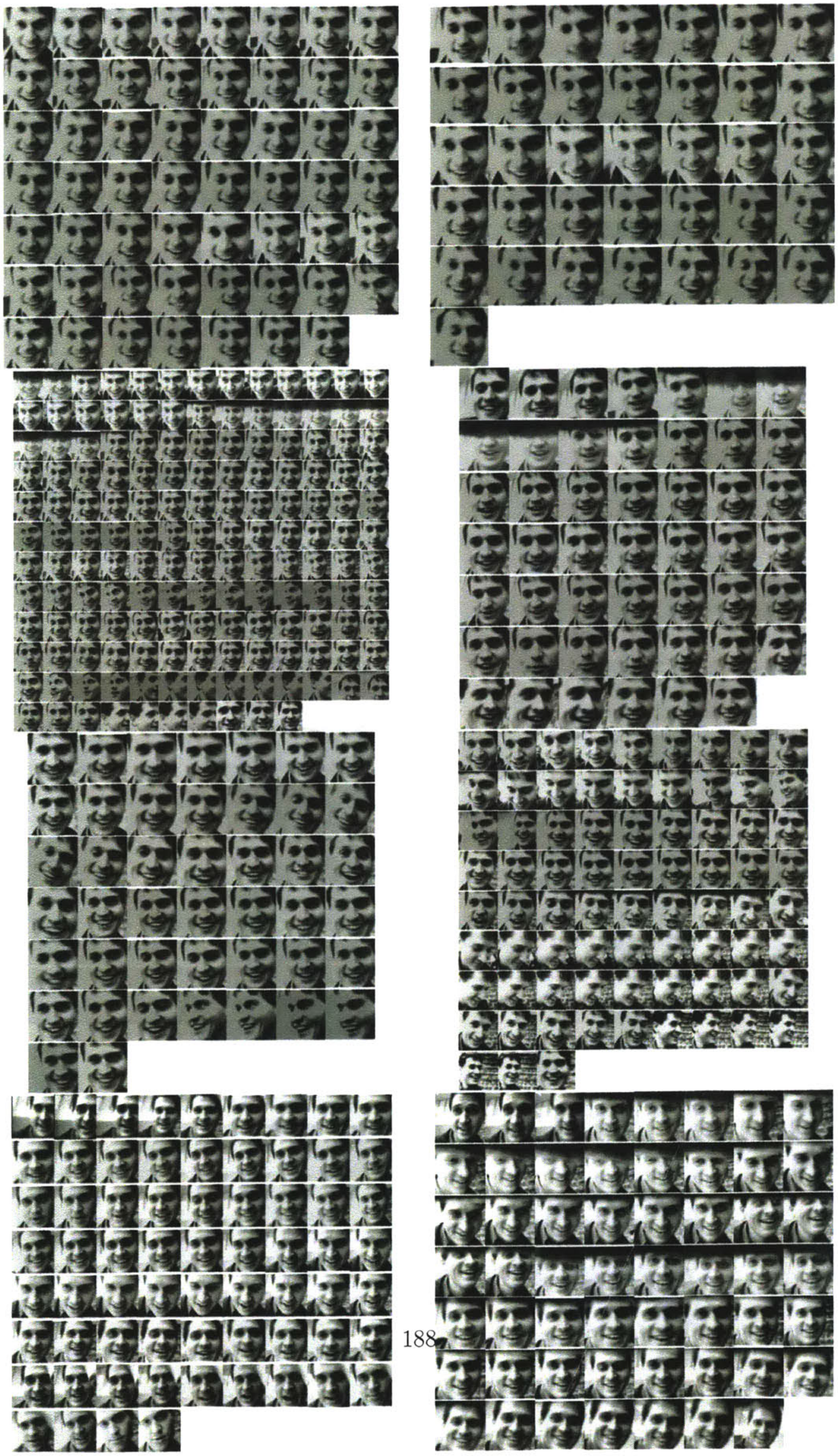


Figure B-16: An example of a good face cluster.

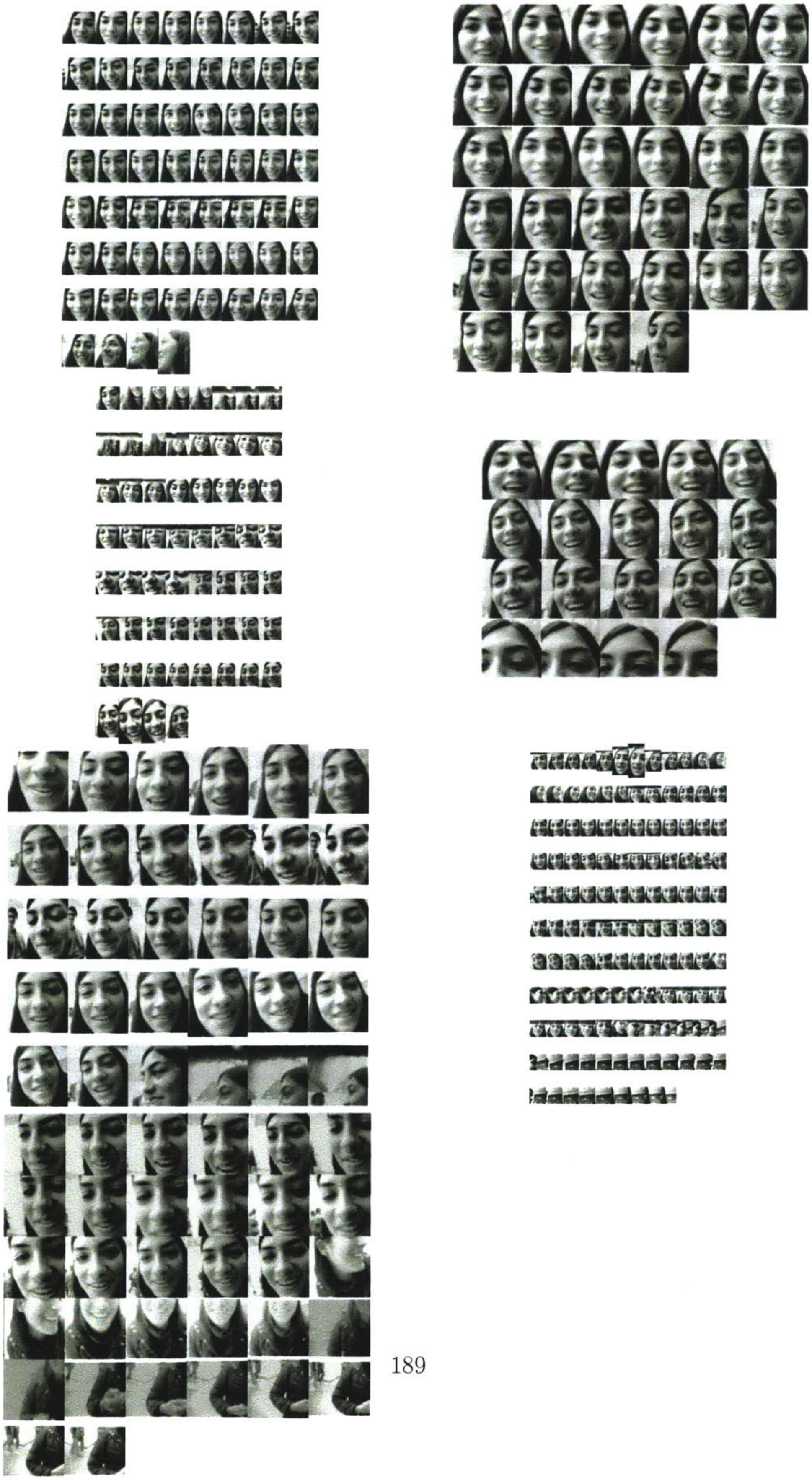


Figure B-17: An example of a good face cluster containing sequences from multiple

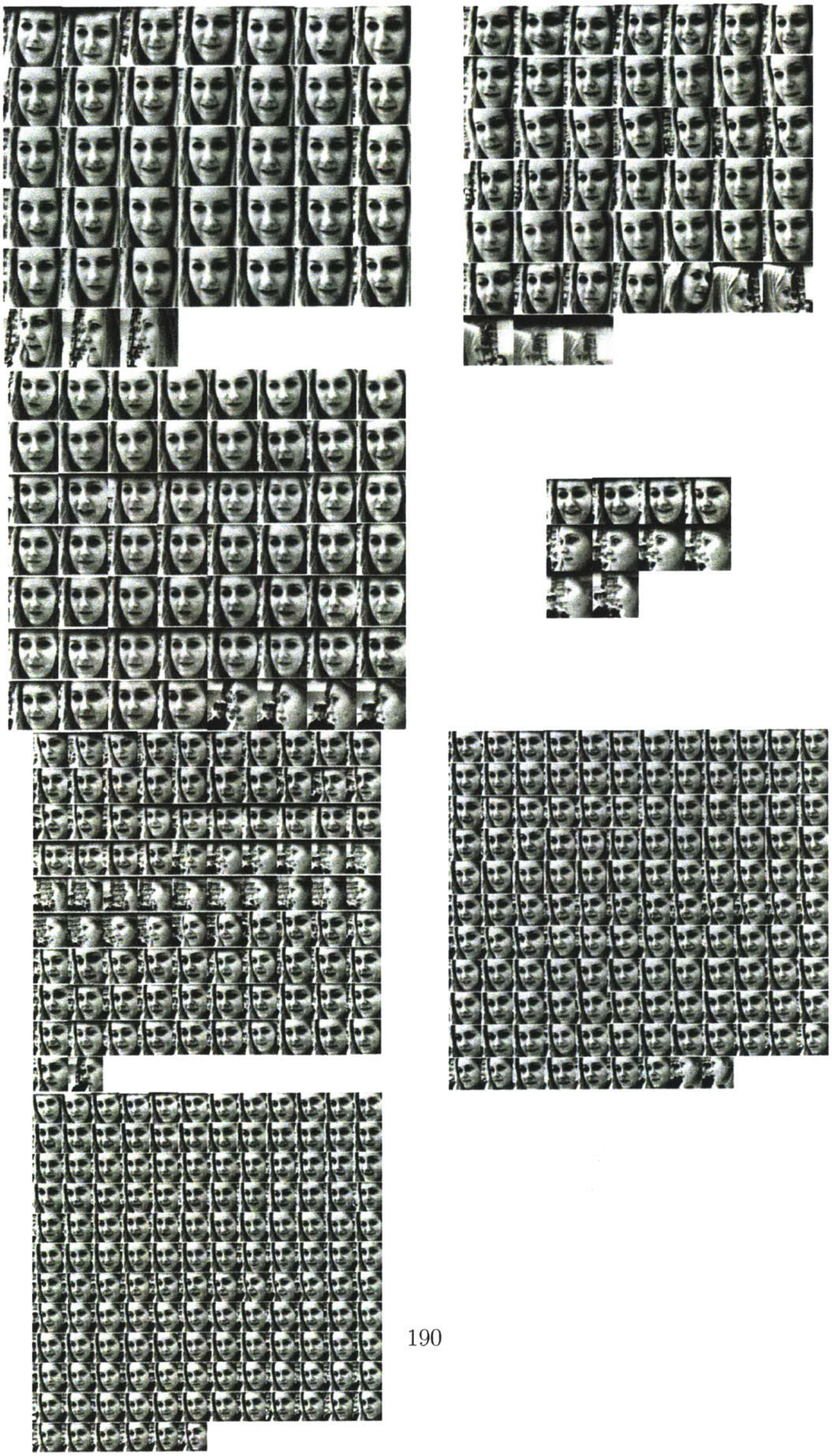


Figure B-18: An example of a good face cluster.

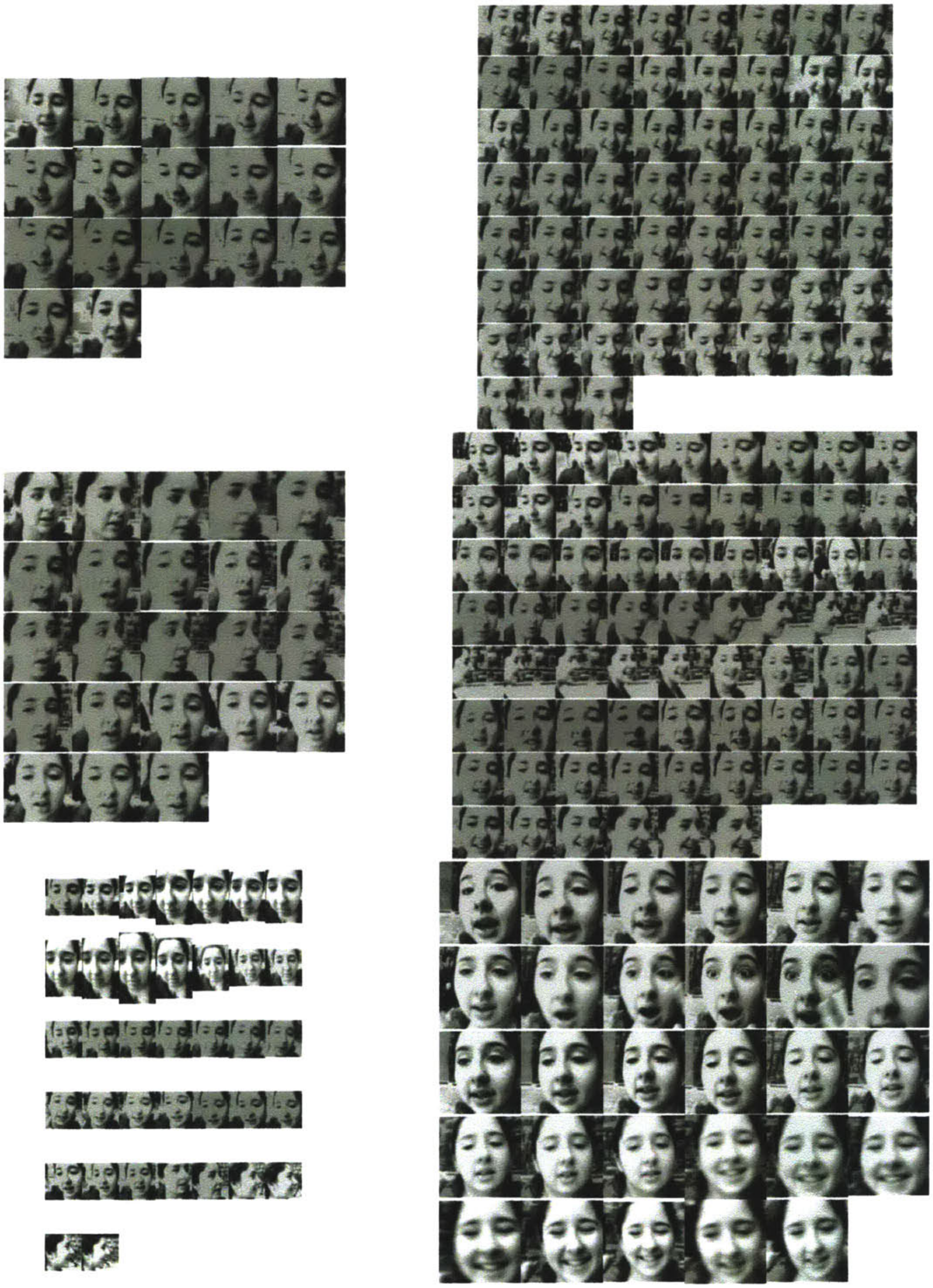


Figure B-19: An example of a good face cluster.

Appendix C

Results of Clustering Evaluation

The following is a set of clustering evaluation results. We extract a number of data sets of different sizes from the collected training data. Each data set was formed by taking contiguous segments from the robot's final training data in order of its appearance during the experiment. These results were generated using data set sizes and parameter values listed in table 5.1.

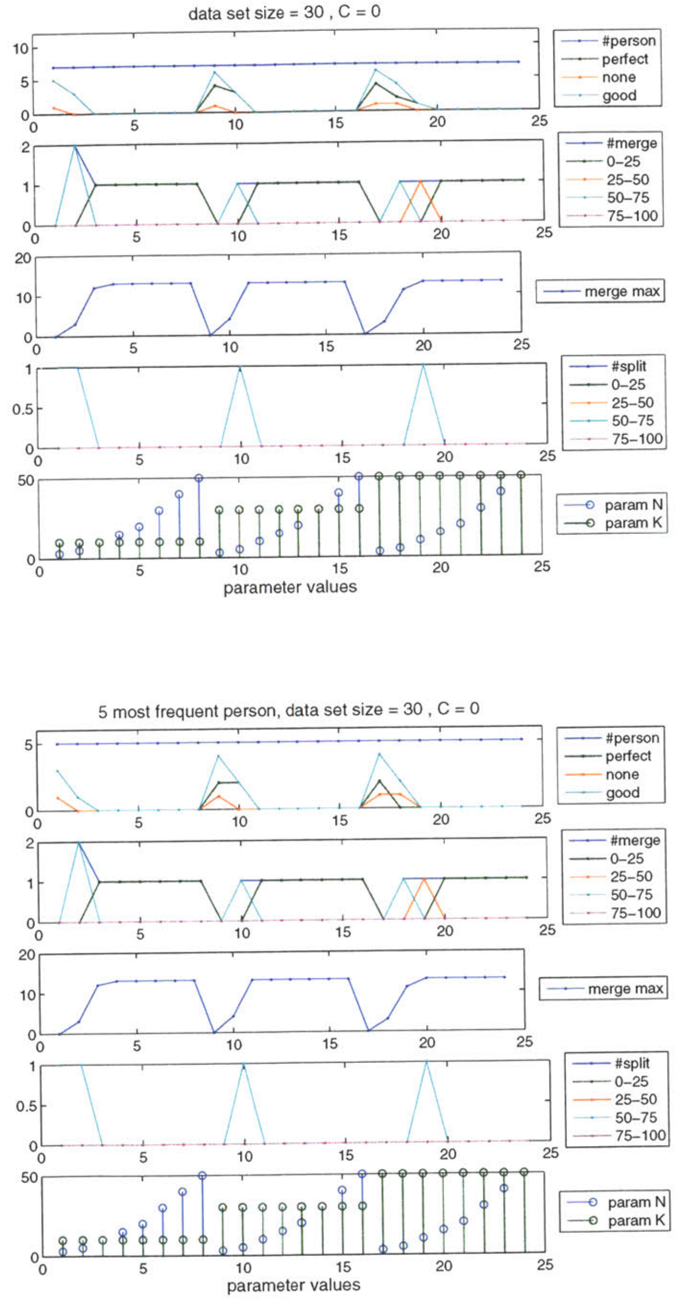


Figure C-1: Results of clustering evaluation of a data set of 30 sequences, C = 0%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set. The top most subplot ilustrates the number of *good*, *perfect*, and *none* resulting clusters. The second subplot shows the number of total merging errors and their distribution for different merged purity values . The third subplot shows the number of merged maximum. The fourth subplot shows the number of splitting errors and their distribution for different split degree values. The rest of the plots in this section follows the same structure.

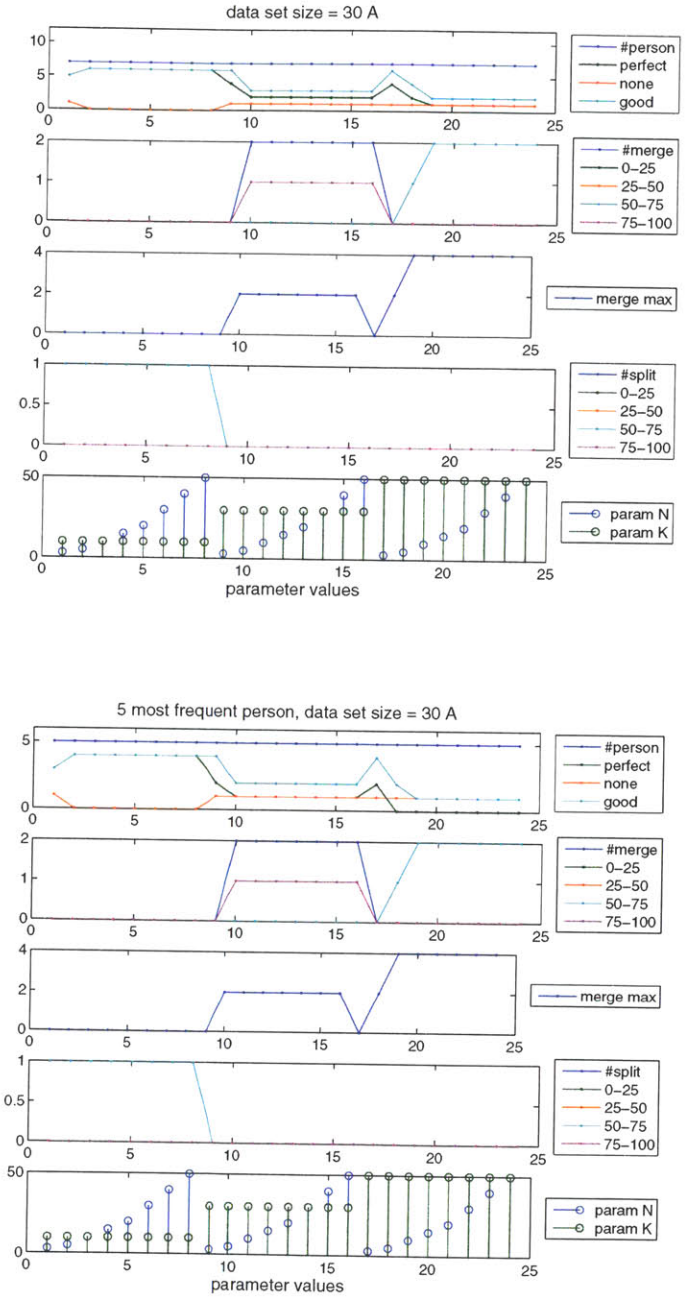


Figure C-2: Results of clustering evaluation of a data set of 30 sequences, C = 70%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

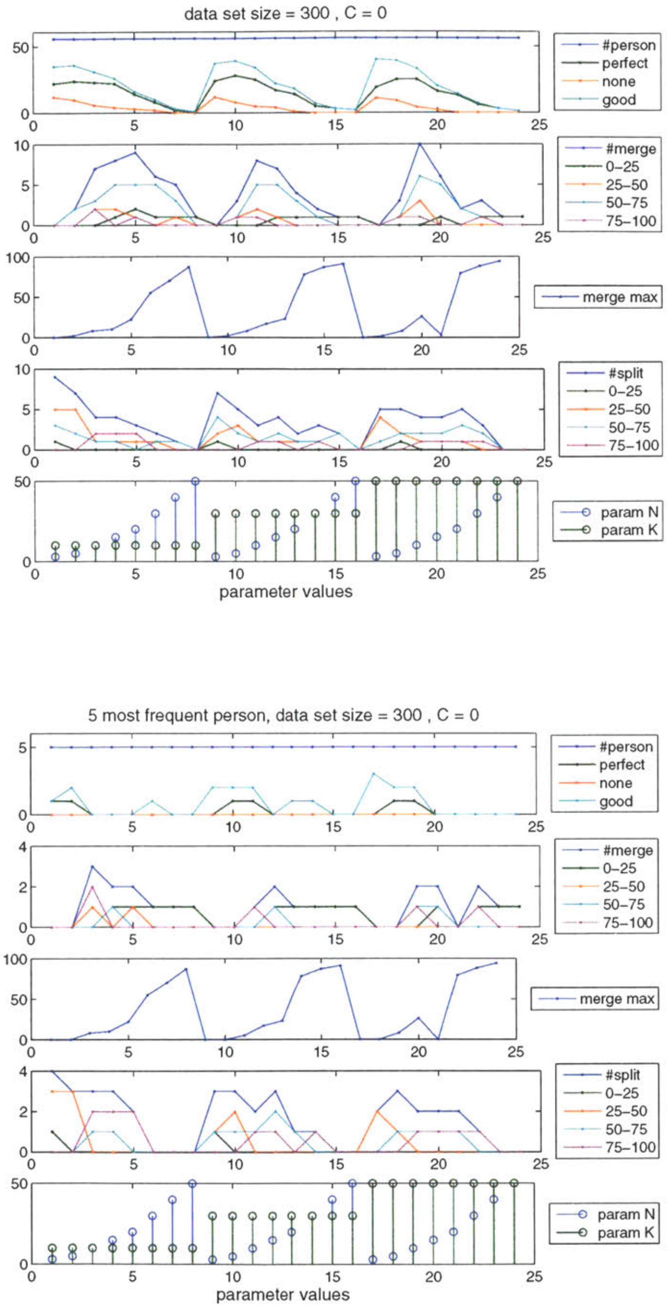


Figure C-3: Results of clustering evaluation of a data set of 300 sequences, C = 0%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

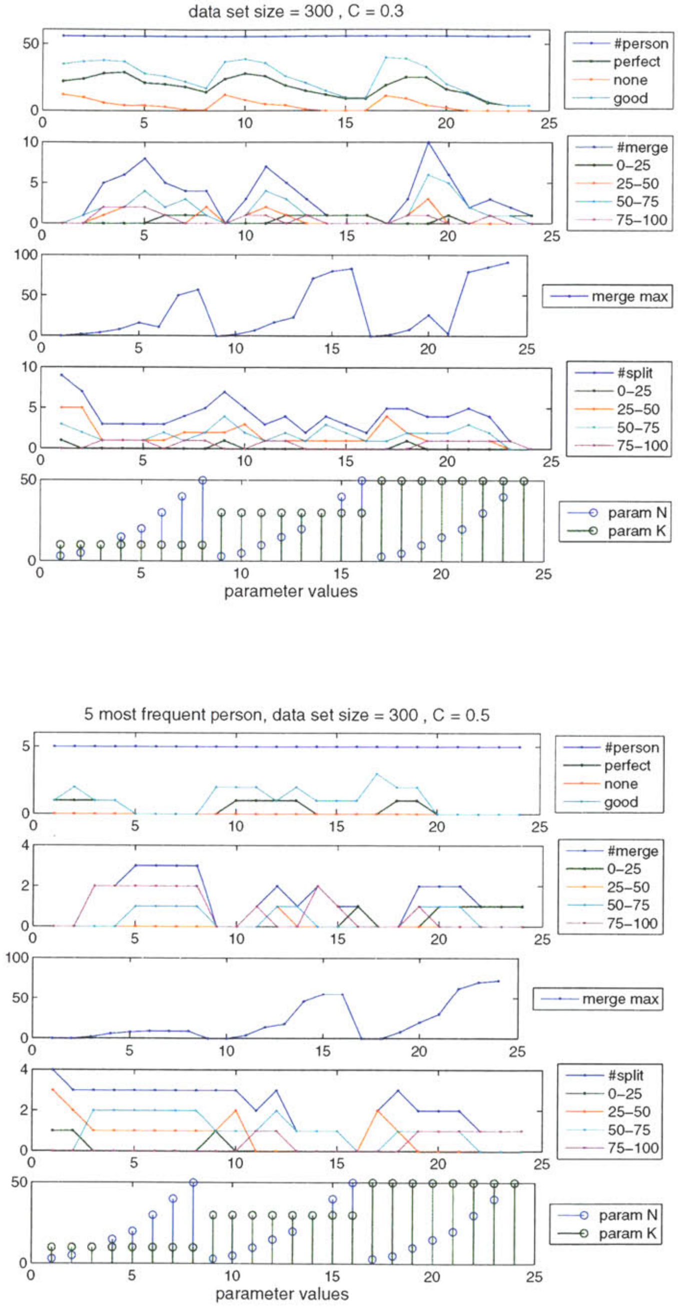
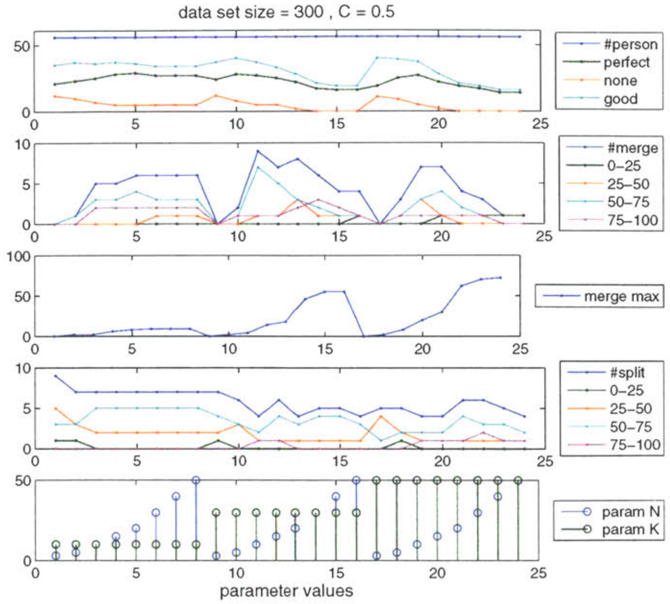


Figure C-4: Results of clustering evaluation of a data set of 300 sequences, C = 30%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.



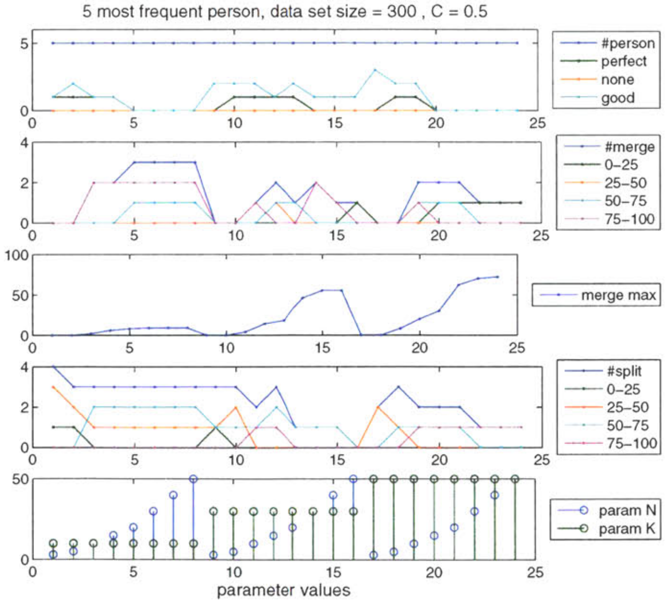


Figure C-5: Results of clustering evaluation of a data set of 300 sequences, C = 50%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

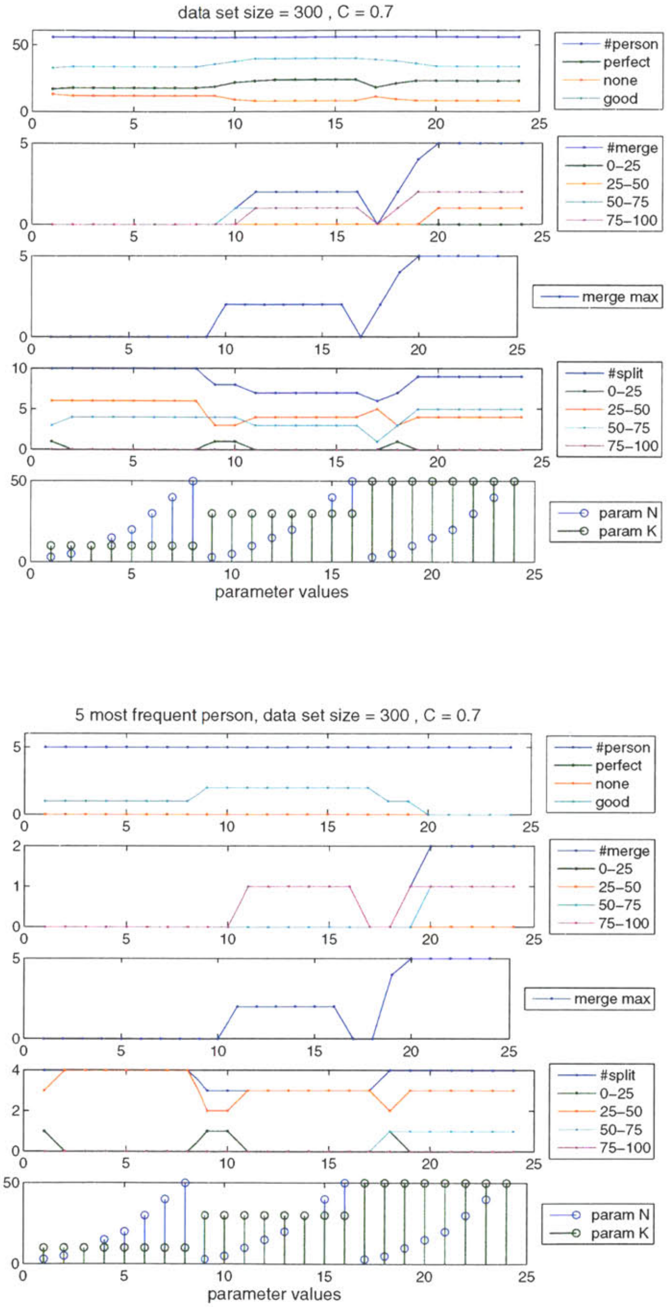
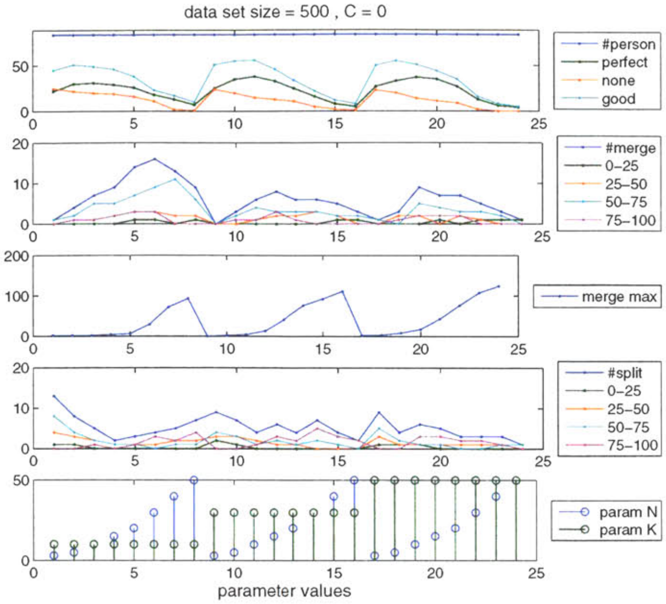


Figure C-6: Results of clustering evaluation of a data set of 300 sequences, C = 70%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.



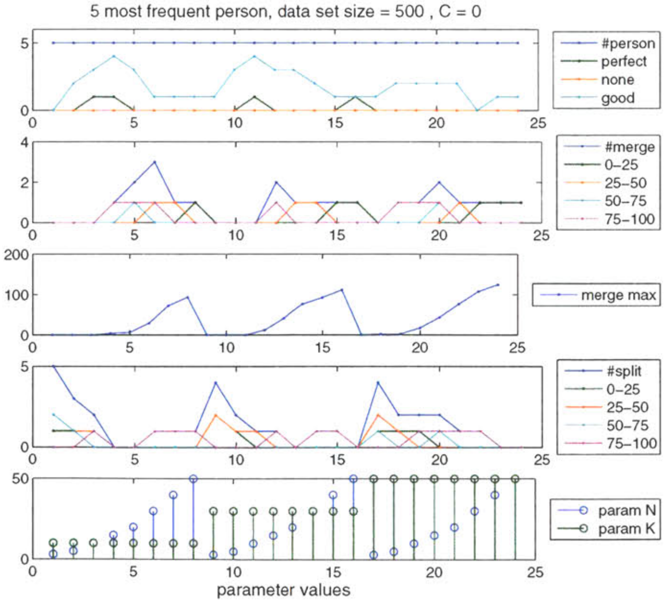


Figure C-7: Results of clustering evaluation of a data set of 500 sequences, C = 0%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

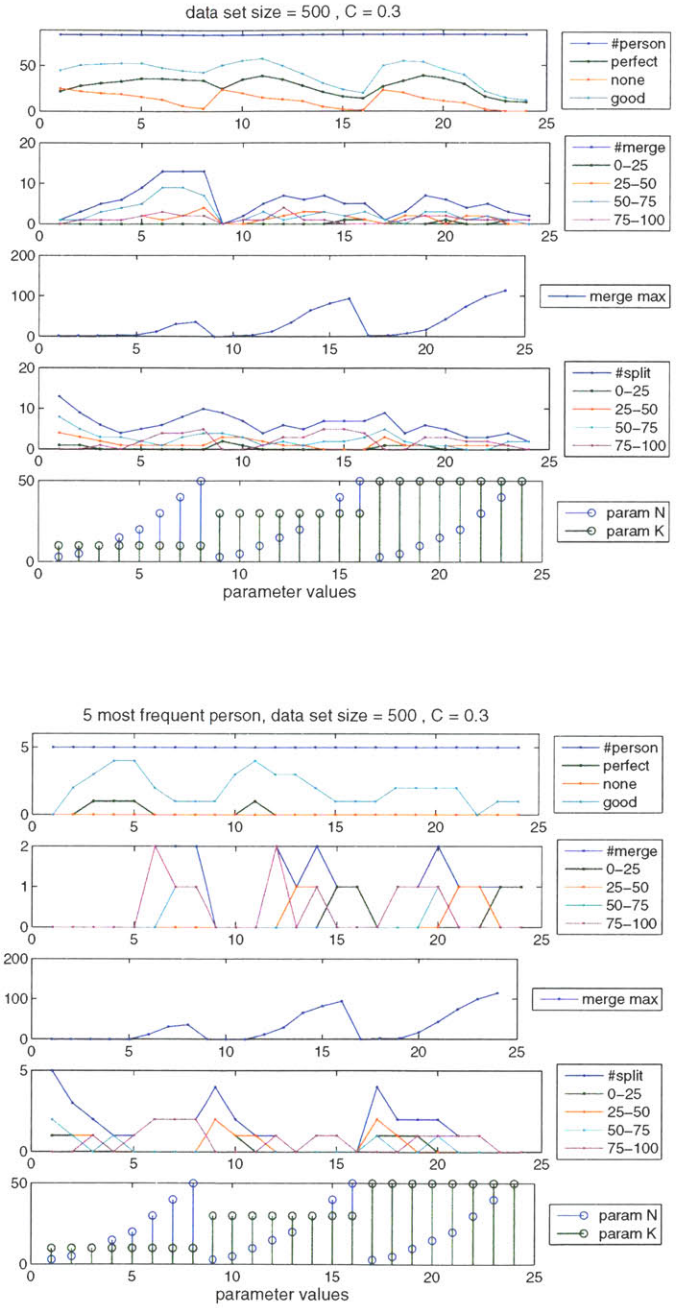


Figure C-8: Results of clustering evaluation of a data set of 500 sequences, C = 30%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

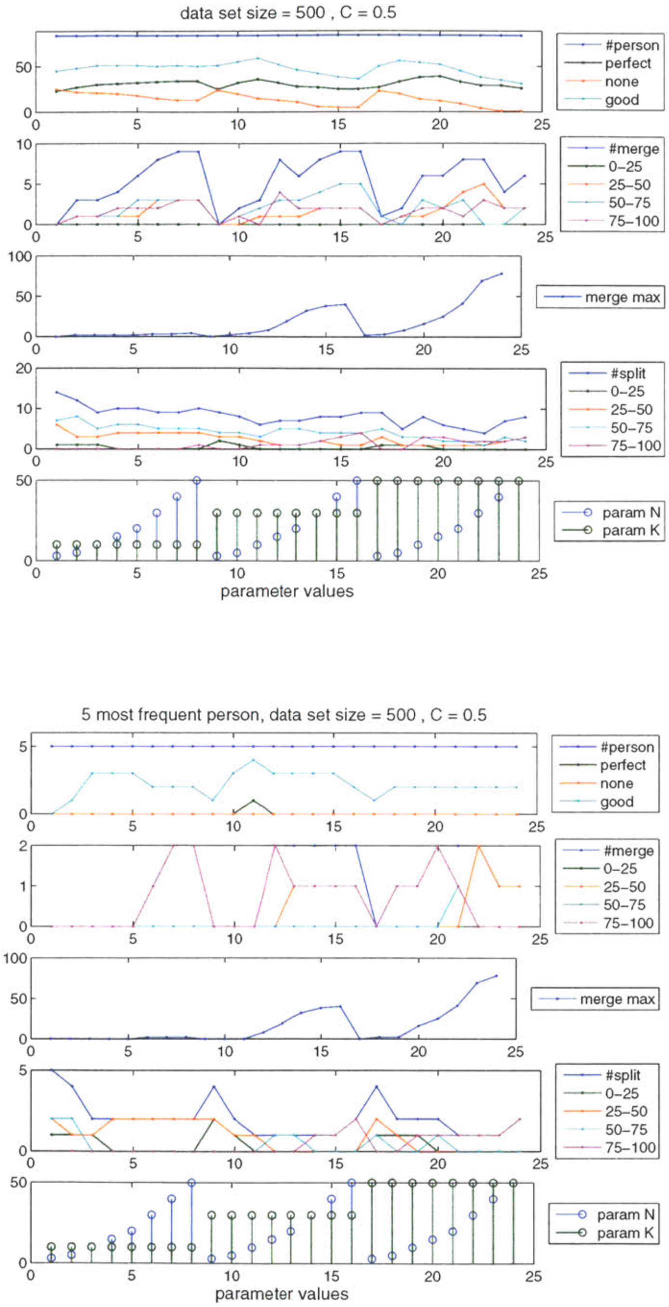


Figure C-9: Results of clustering evaluation of a data set of 500 sequences, C = 50%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

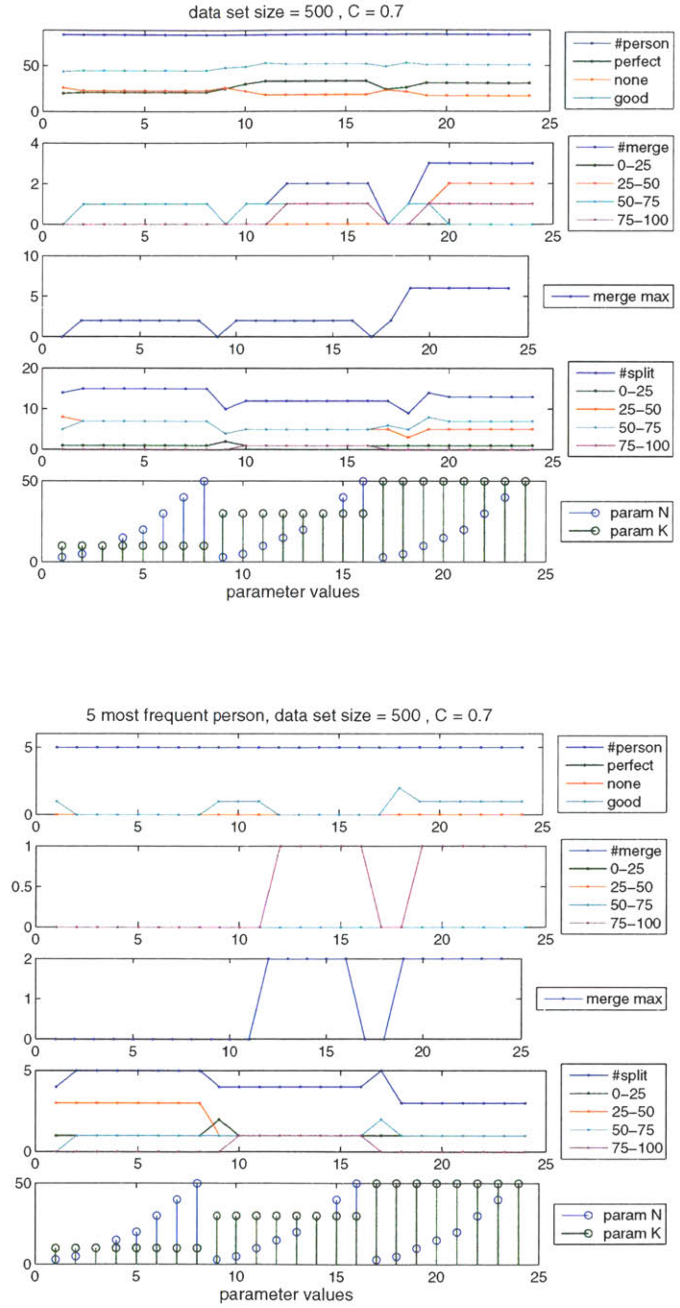


Figure C-10: Results of clustering evaluation of a data set of 500 sequences, C = 70%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

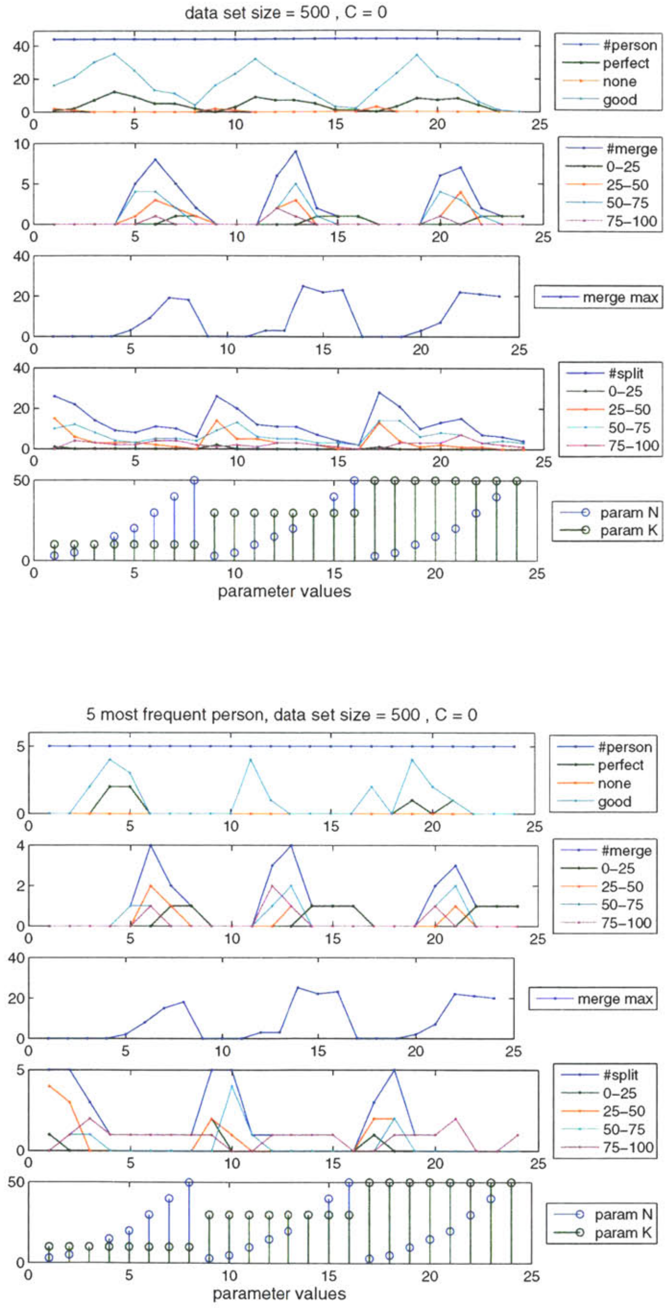


Figure C-11: Results of clustering evaluation of a data set of 500 sequences, C = 0%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

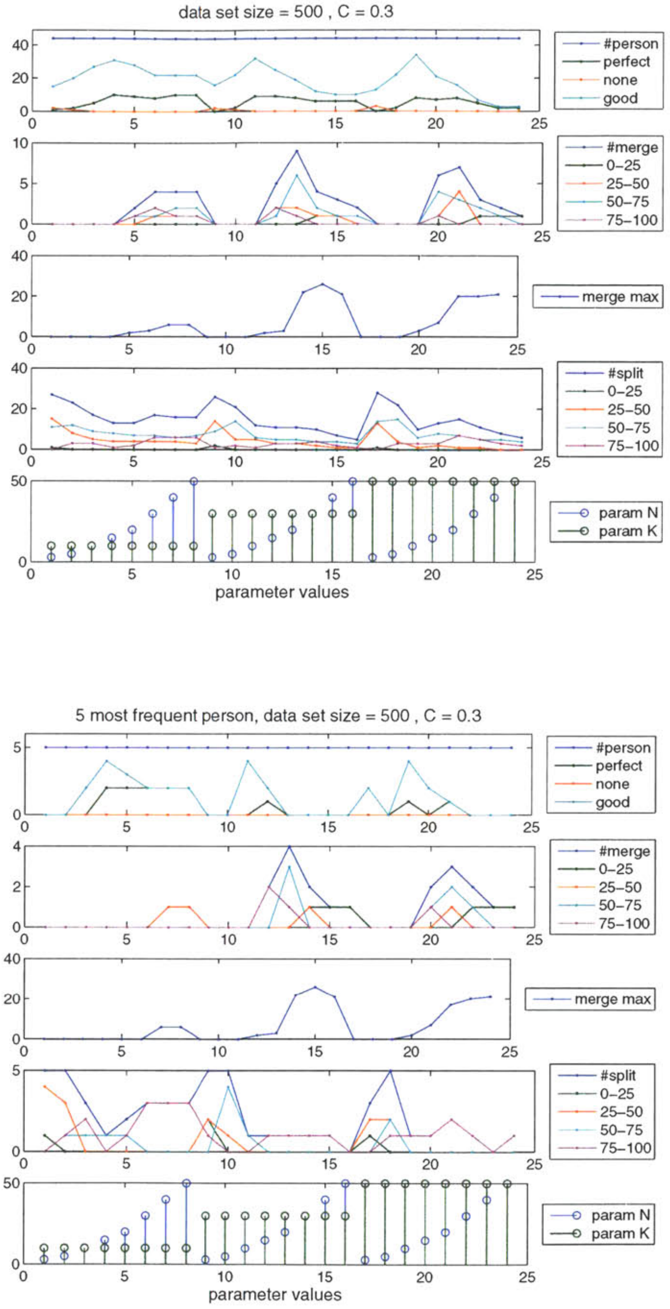
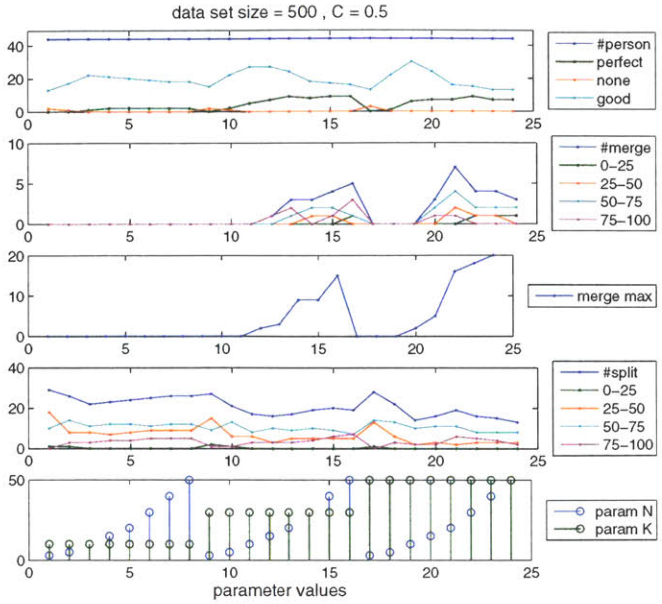


Figure C-12: Results of clustering evaluation of a data set of 500 sequences, C = 30%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.



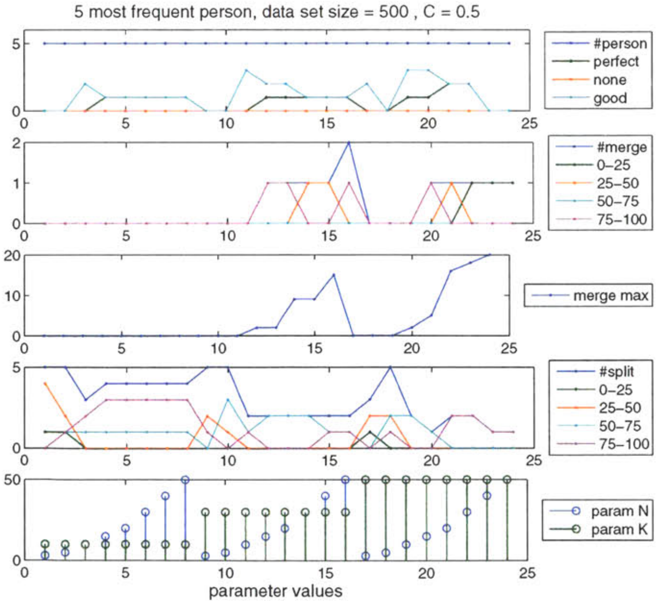
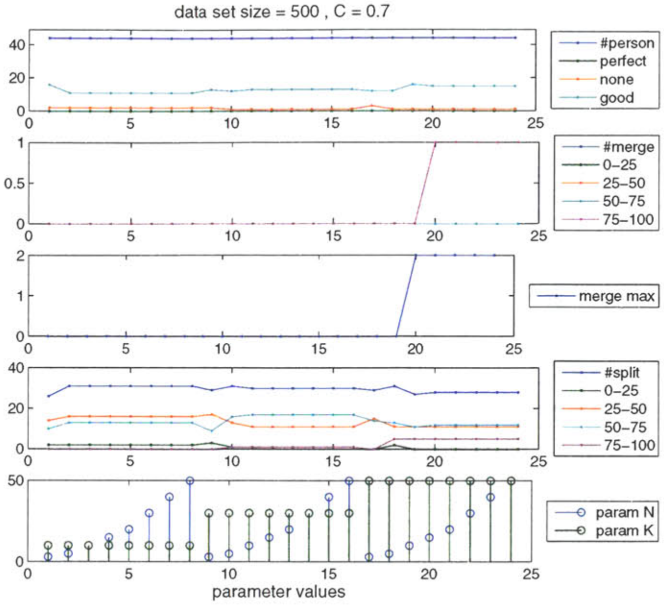


Figure C-13: Results of clustering evaluation of a data set of 500 sequences, C = 50%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.



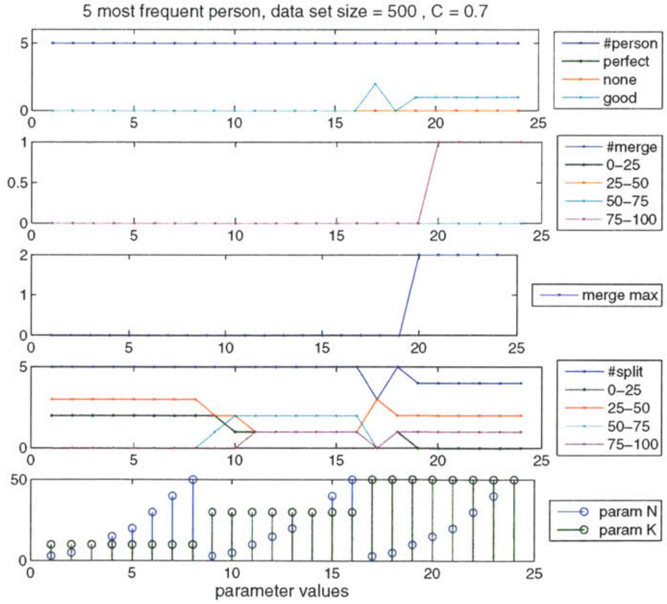


Figure C-14: Results of clustering evaluation of a data set of 500 sequences, C = 70%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

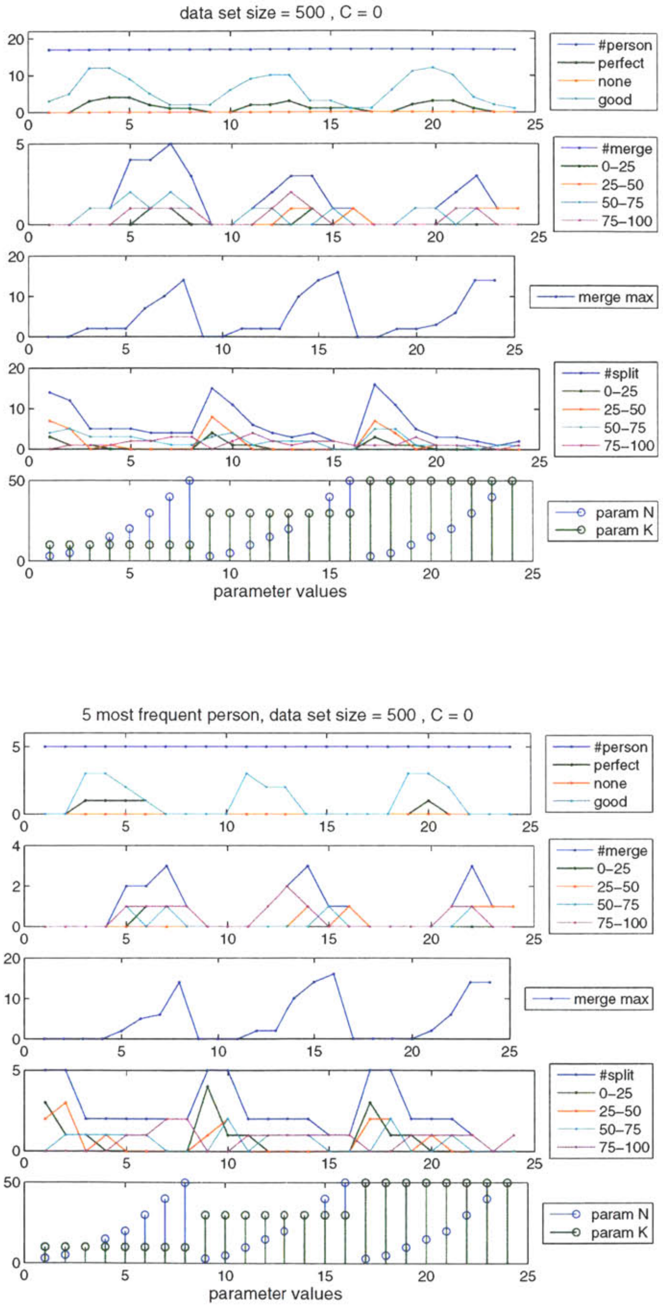


Figure C-15: Results of clustering evaluation of a data set of 500 sequences, C = 0%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

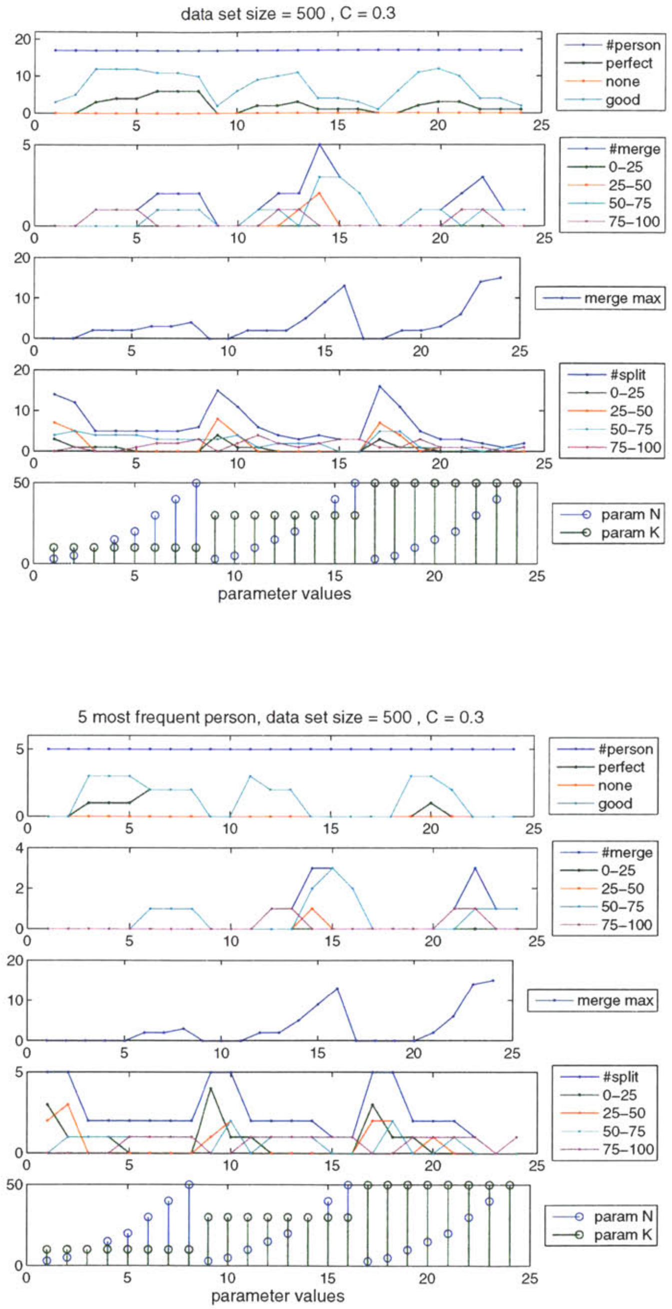


Figure C-16: Results of clustering evaluation of a data set of 500 sequences, C = 30%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

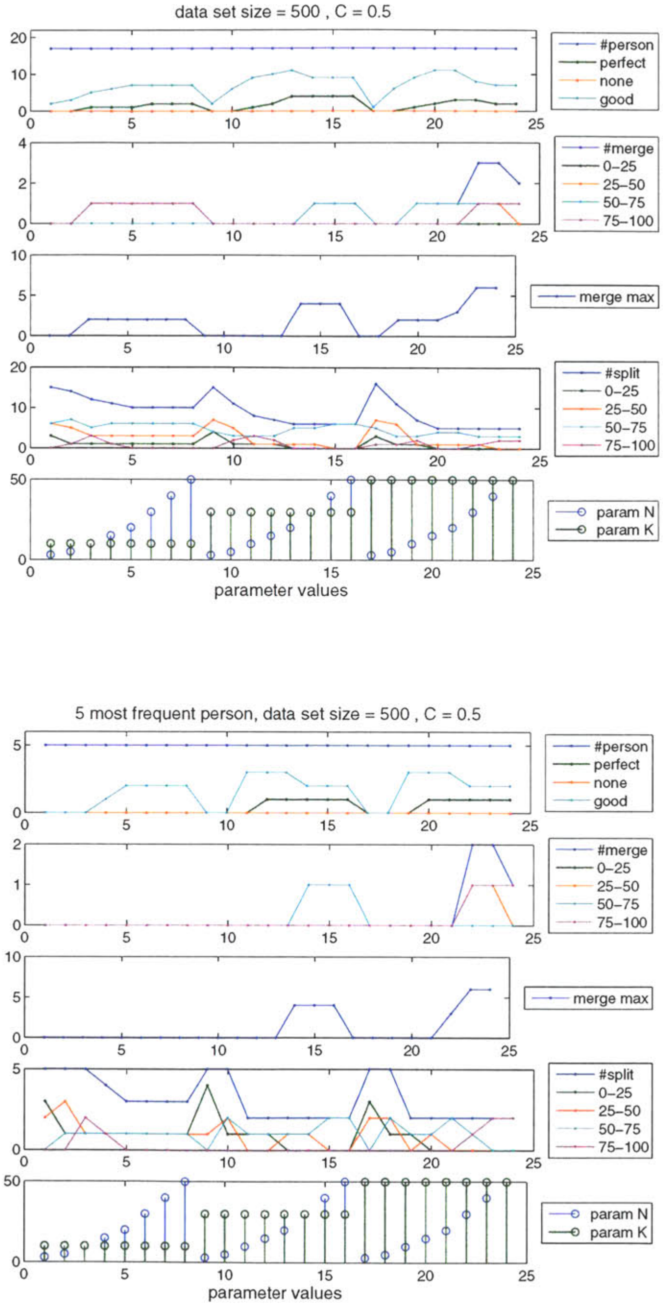


Figure C-17: Results of clustering evaluation of a data set of 500 sequences, C = 50%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

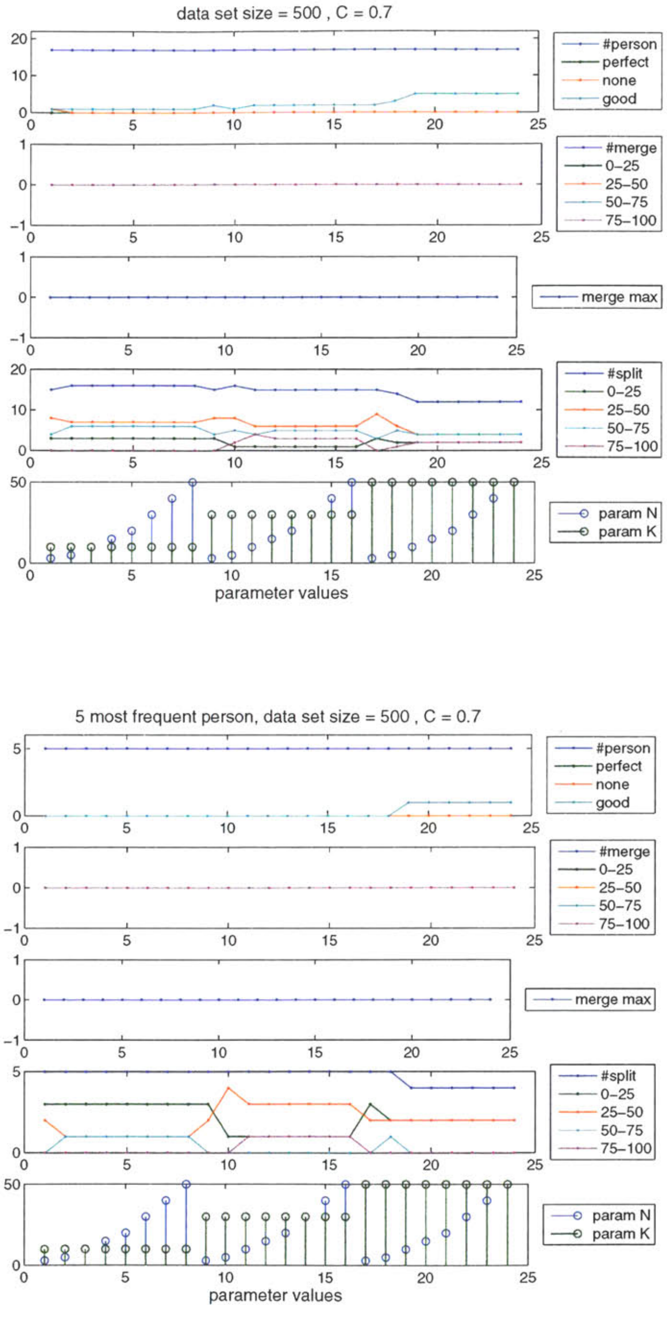


Figure C-18: Results of clustering evaluation of a data set of 500 sequences, C = 70%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

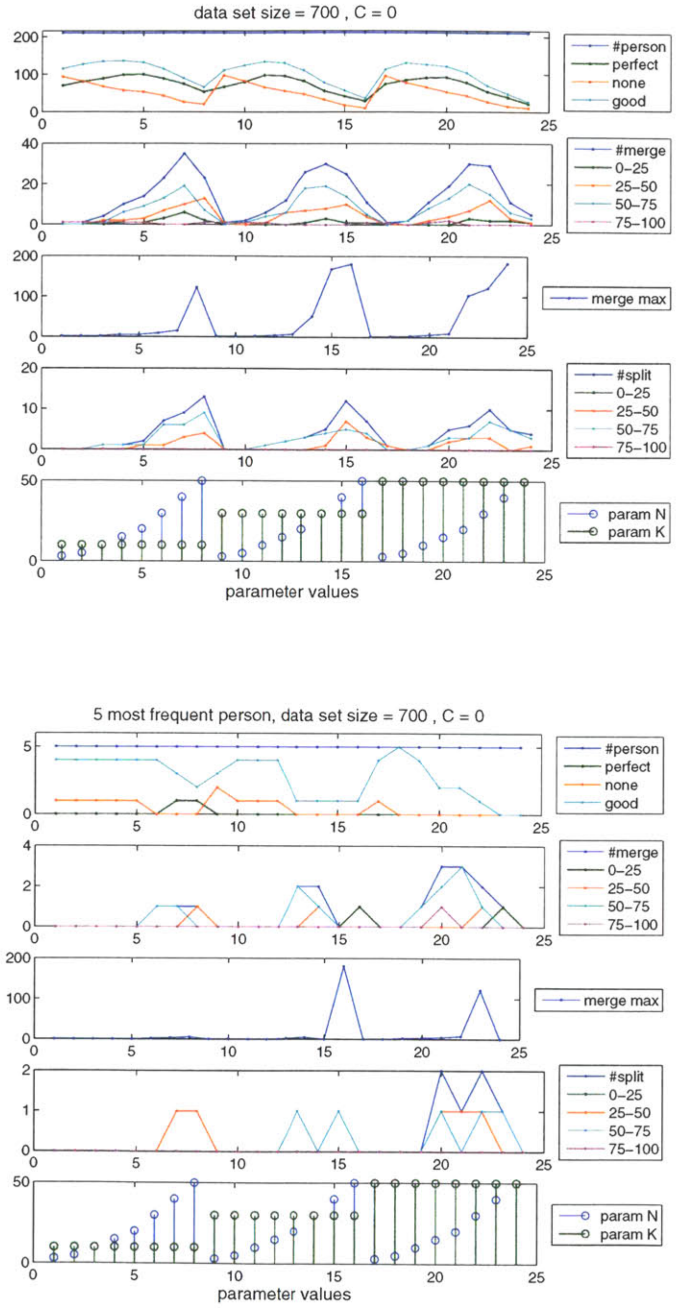


Figure C-19: Results of clustering evaluation of a data set of 700 sequences, C = 0%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

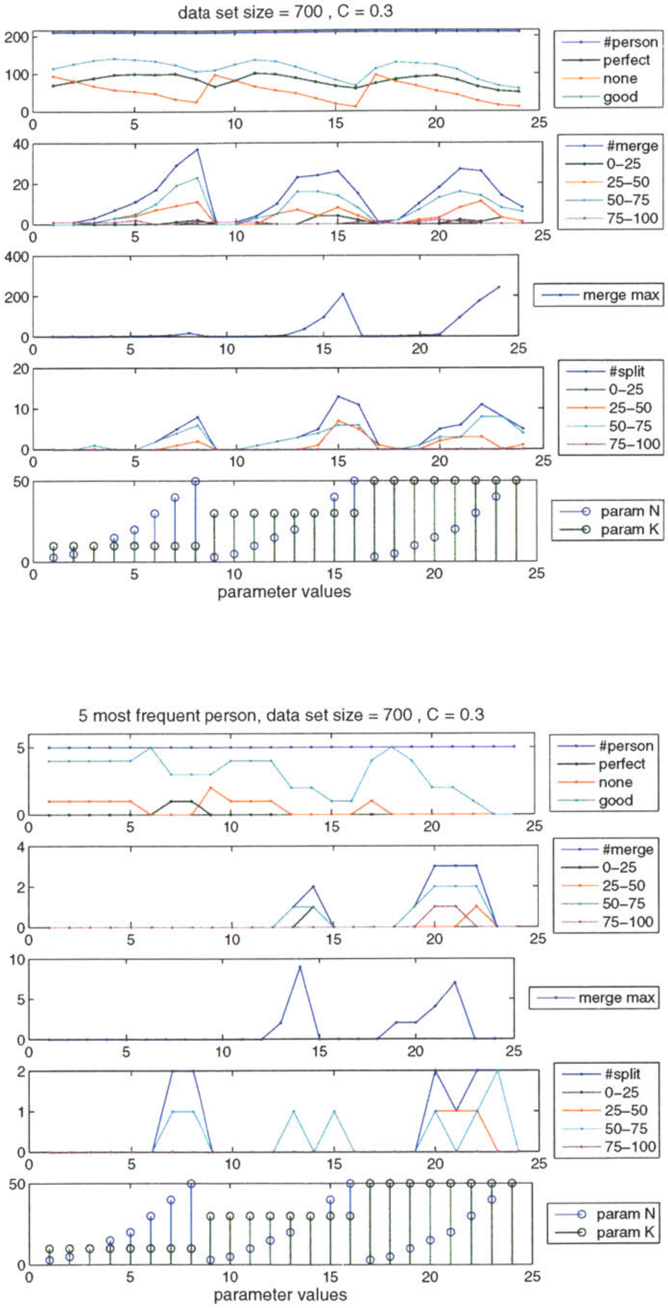
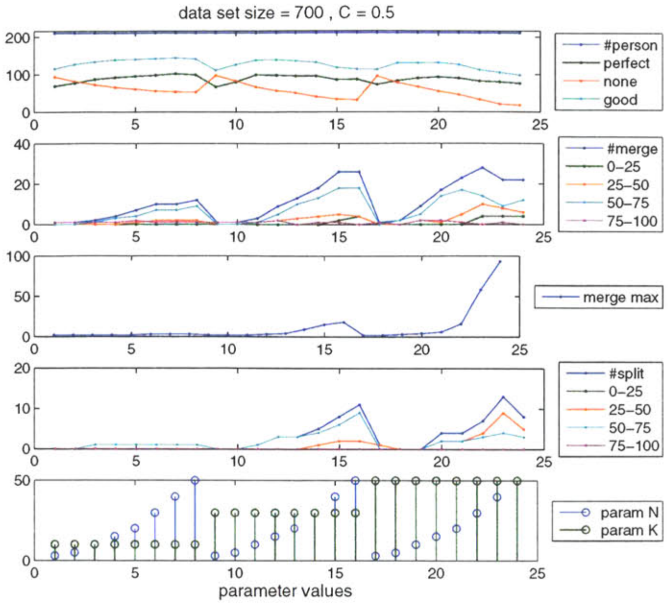


Figure C-20: Results of clustering evaluation of a data set of 700 sequences, C = 30%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.



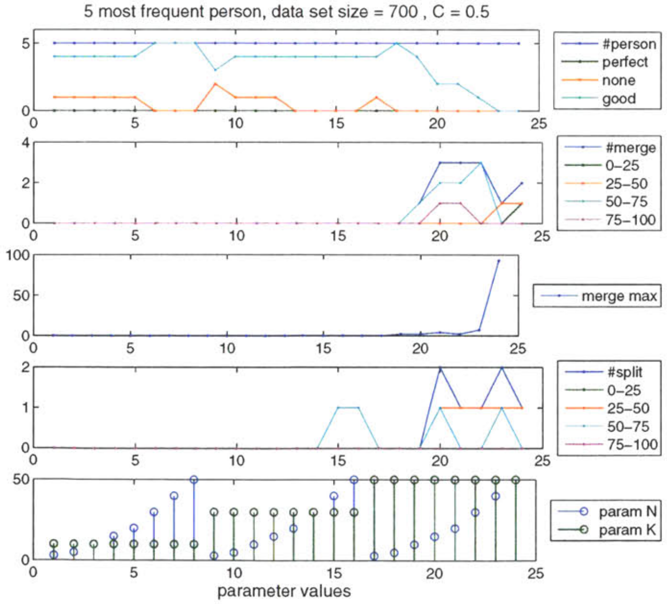


Figure C-21: Results of clustering evaluation of a data set of 700 sequences, C = 50%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

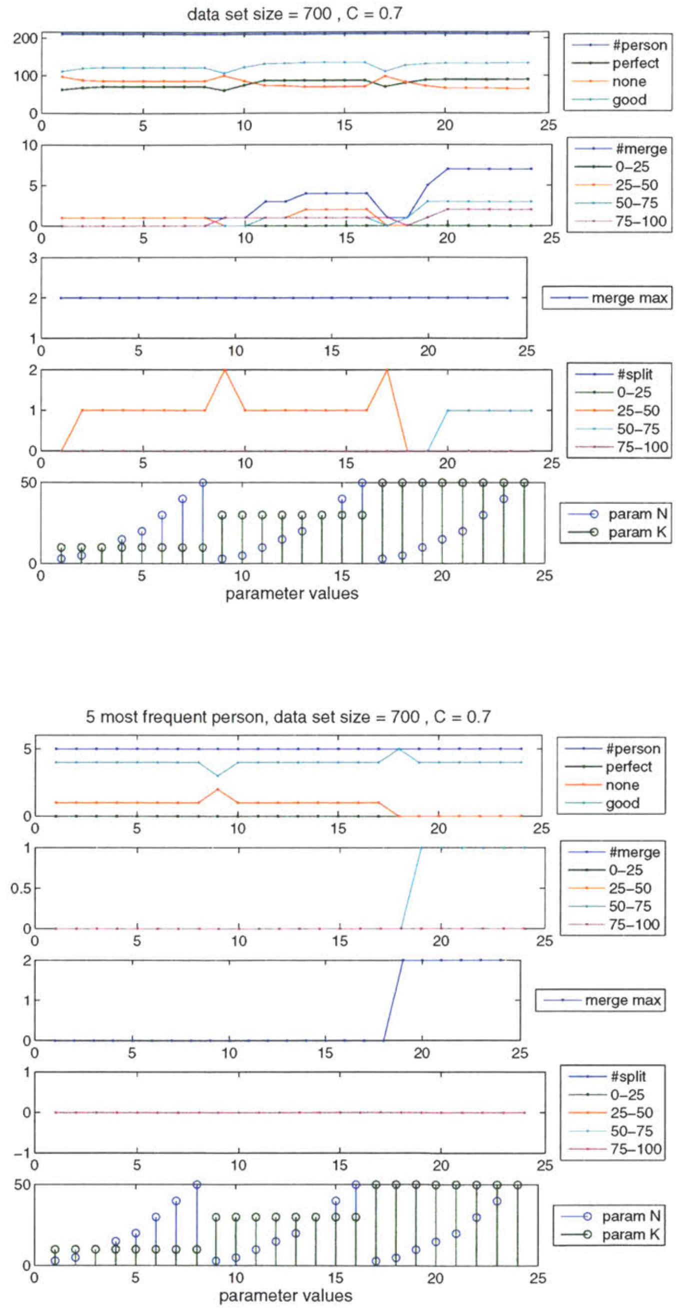


Figure C-22: Results of clustering evaluation of a data set of 700 sequences, C = 70%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

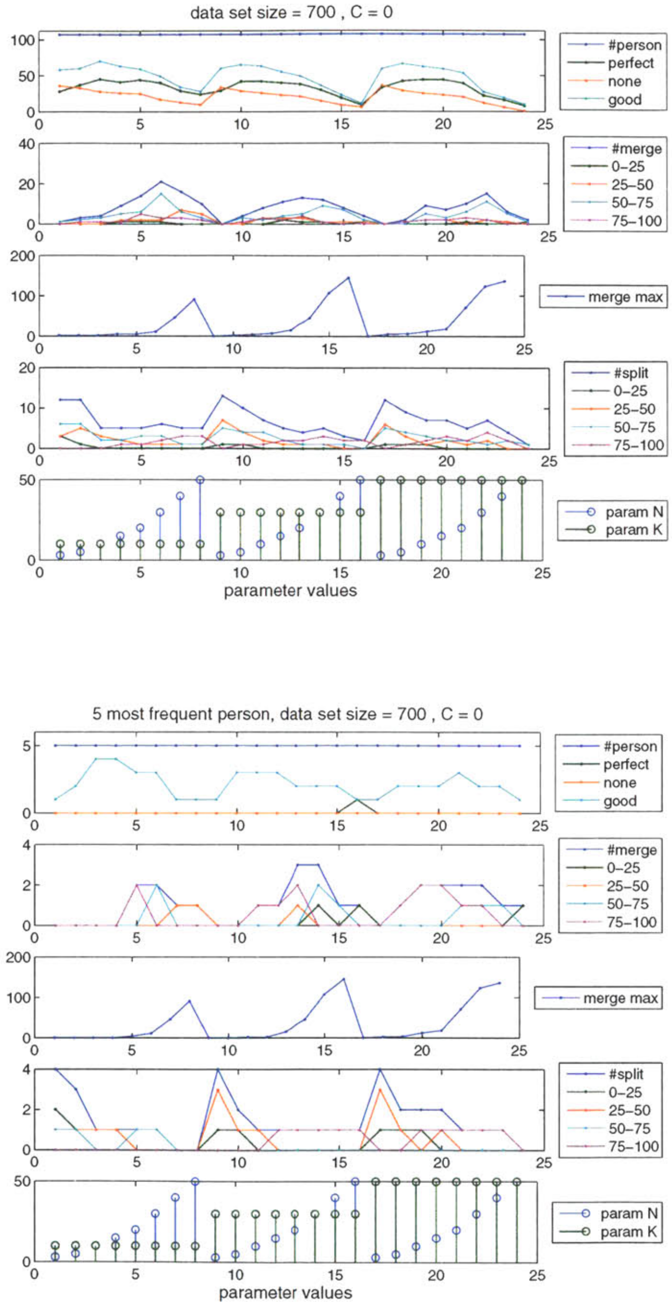


Figure C-23: Results of clustering evaluation of a data set of 700 sequences, C = 0%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

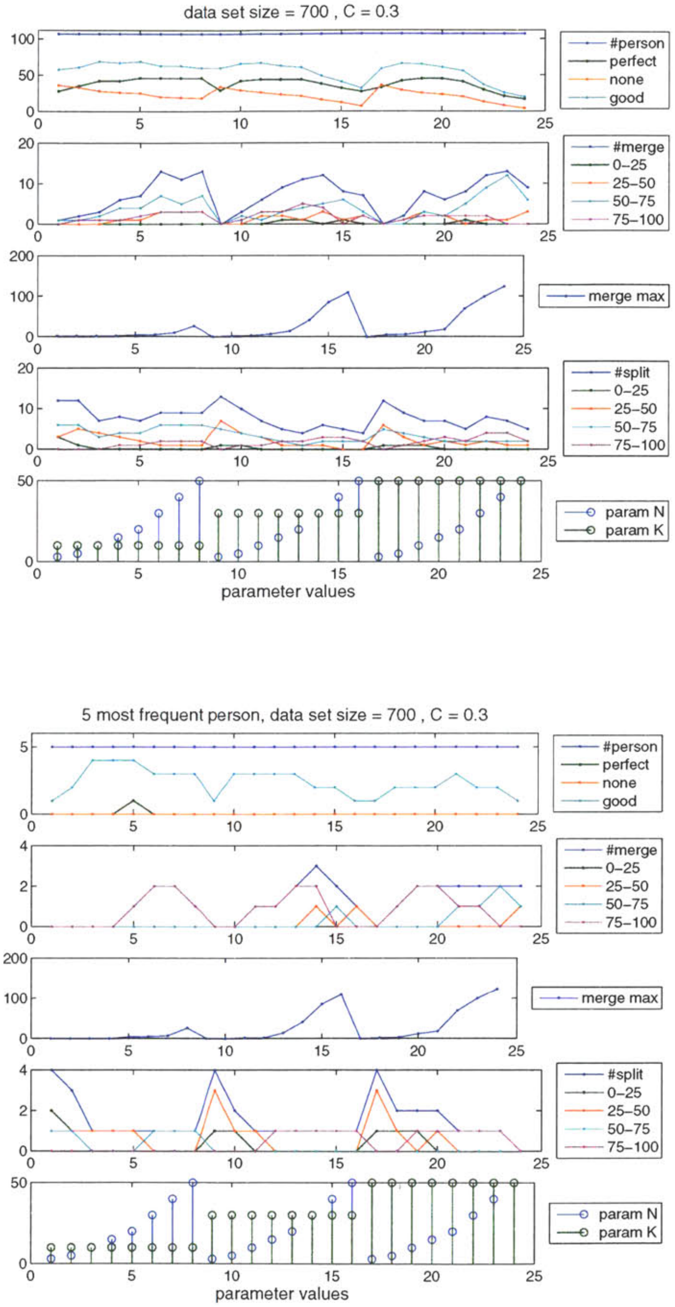
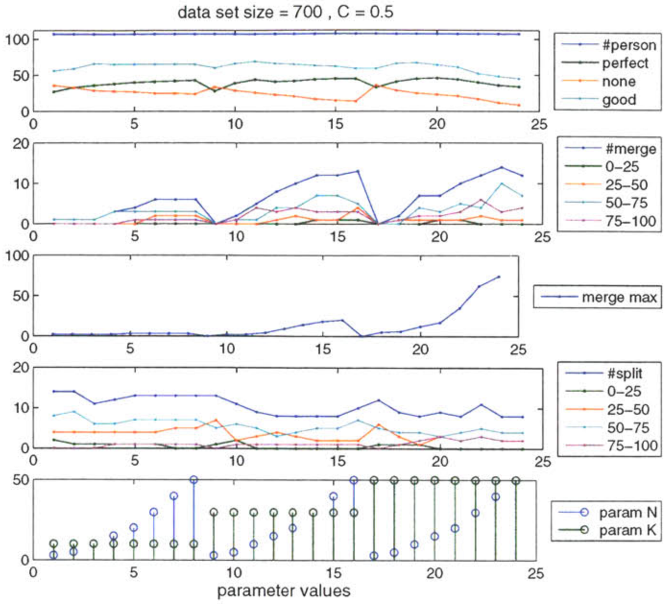


Figure C-24: Results of clustering evaluation of a data set of 700 sequences, C = 30%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.



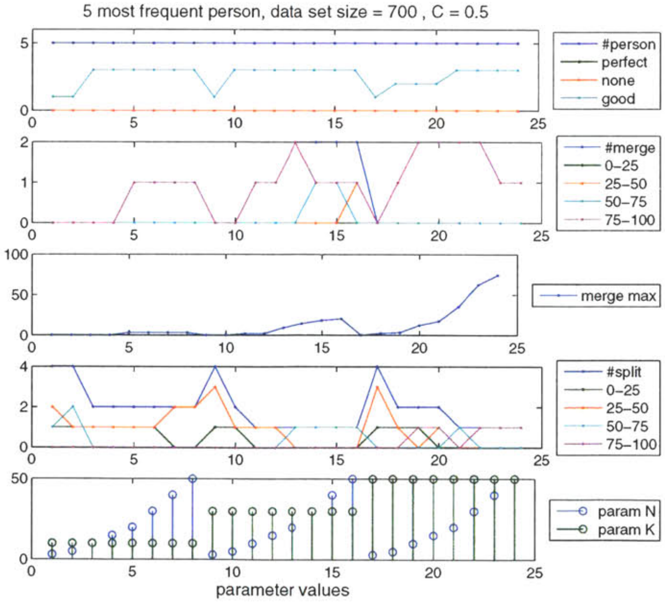


Figure C-25: Results of clustering evaluation of a data set of 700 sequences, C = 50%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

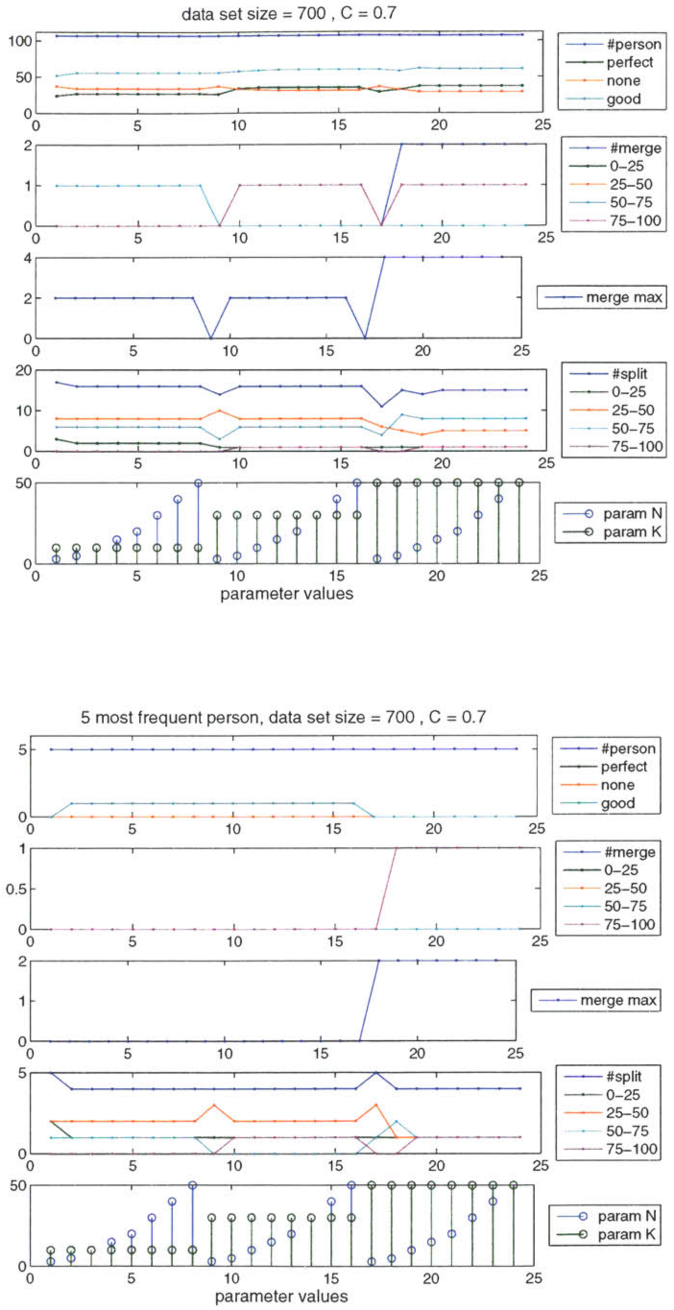


Figure C-26: Results of clustering evaluation of a data set of 700 sequences, C = 70%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

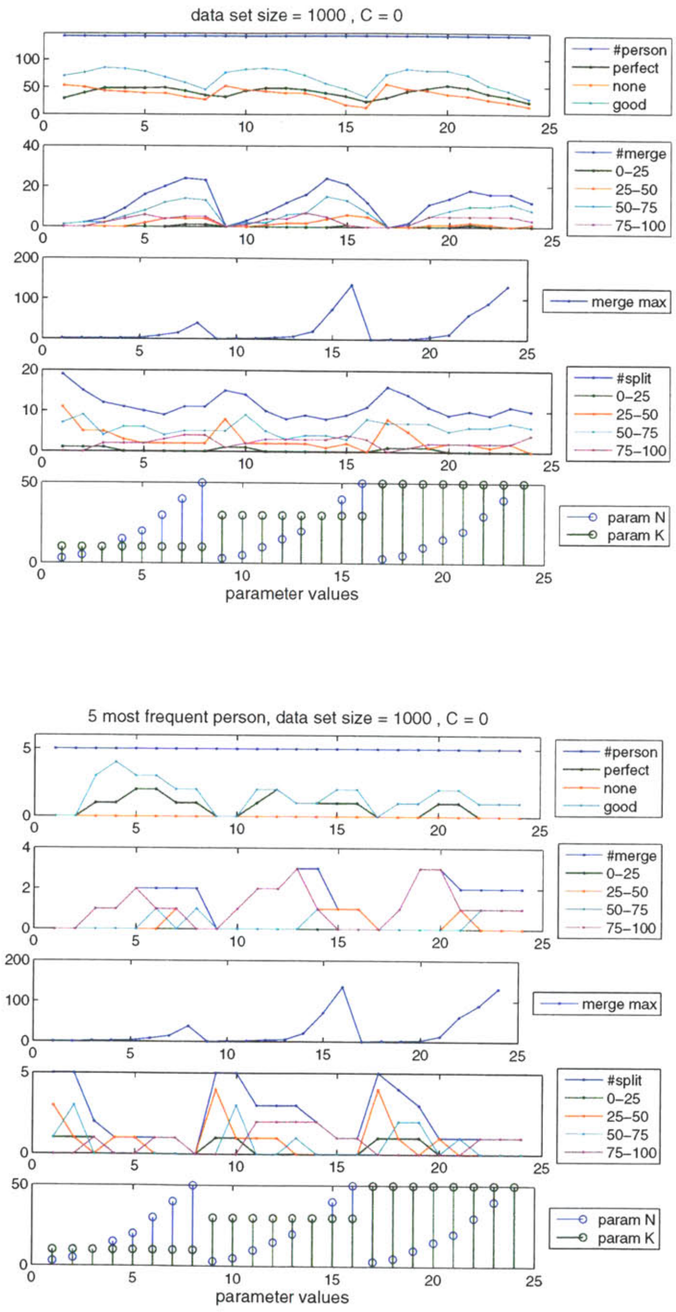
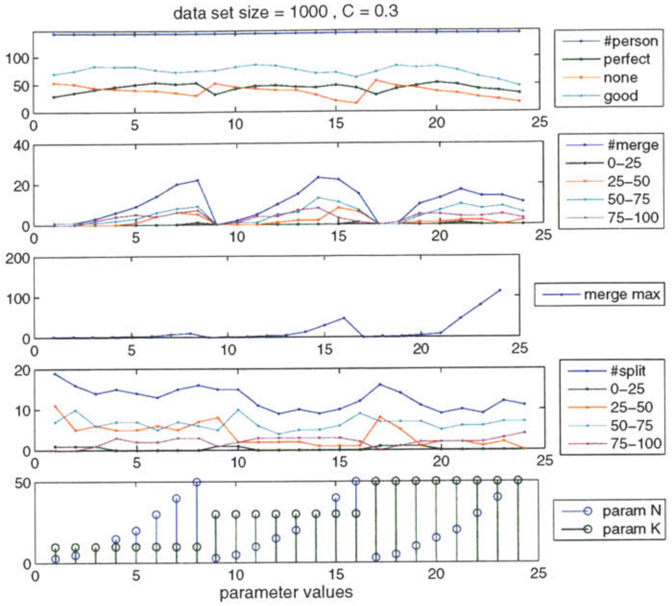


Figure C-27: Results of clustering evaluation of a data set of 1000 sequences, C = 0%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.



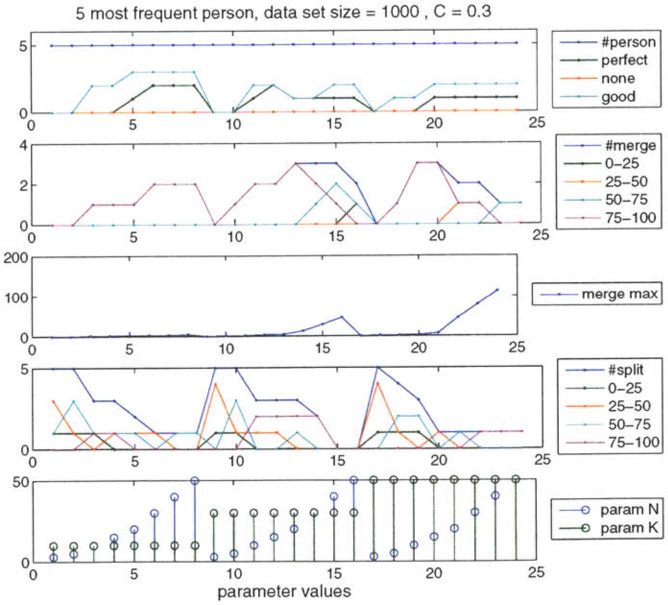


Figure C-28: Results of clustering evaluation of a data set of 1000 sequences, C = 30%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

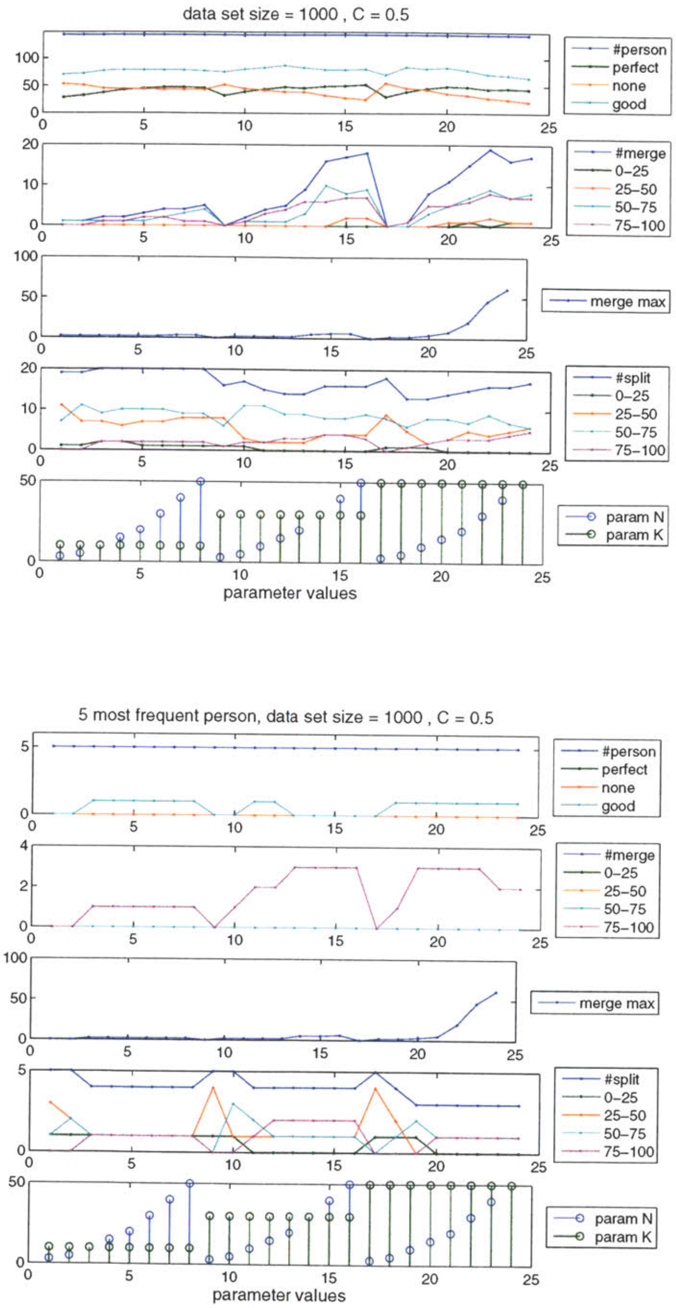


Figure C-29: Results of clustering evaluation of a data set of 1000 sequences, C = 50%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

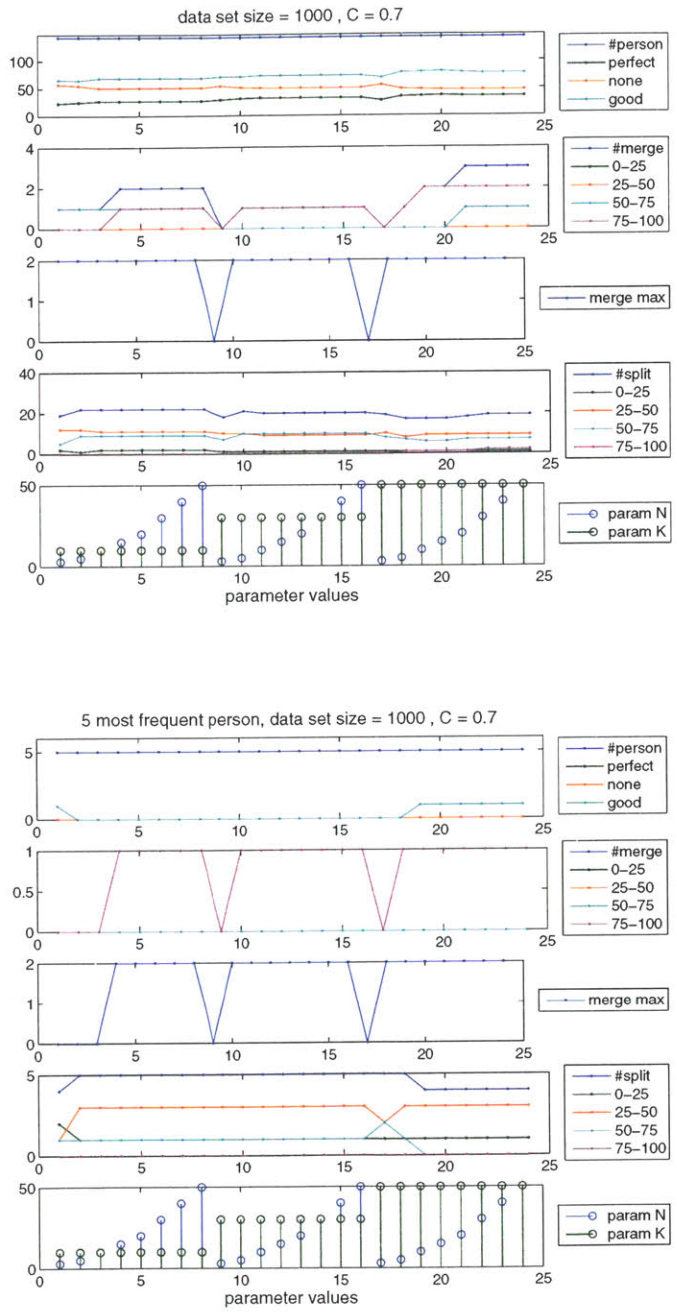
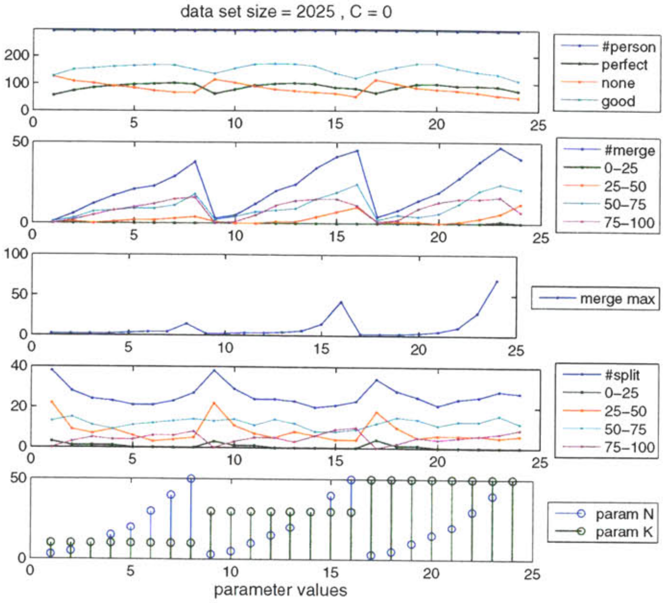


Figure C-30: Results of clustering evaluation of a data set of 1000 sequences, C = 70%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.



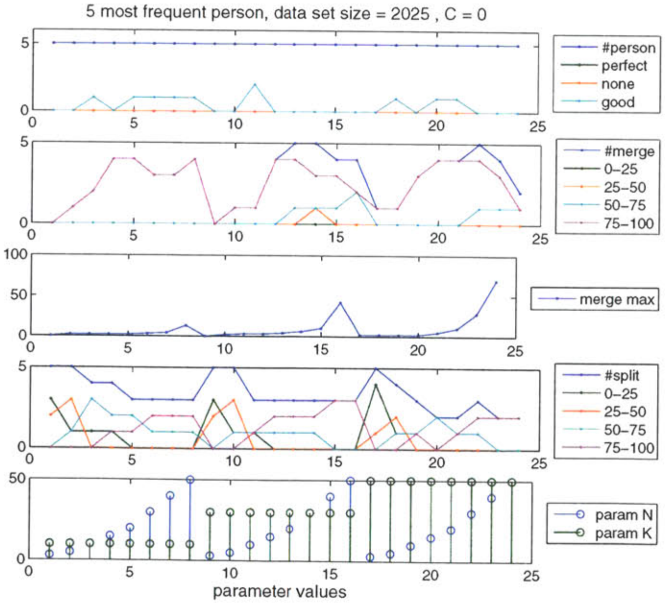


Figure C-31: Results of clustering evaluation of a data set of 2025 sequences, C = 0%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

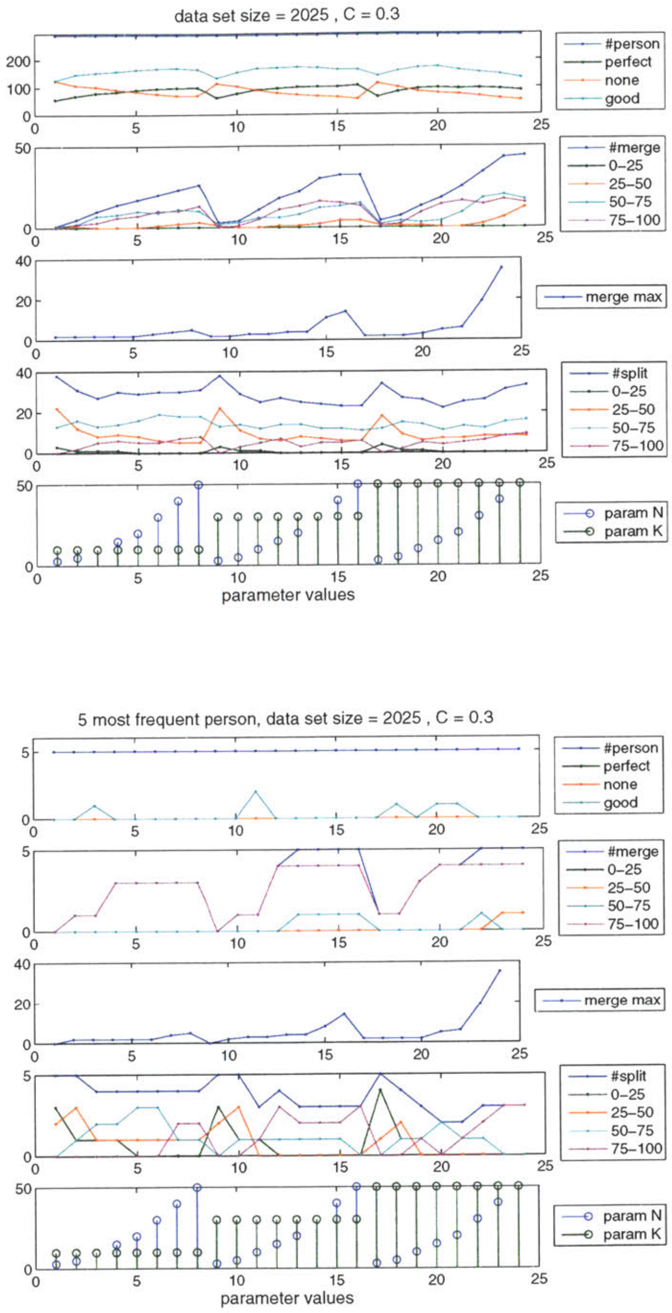


Figure C-32: Results of clustering evaluation of a data set of 2025 sequences, C = 30%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

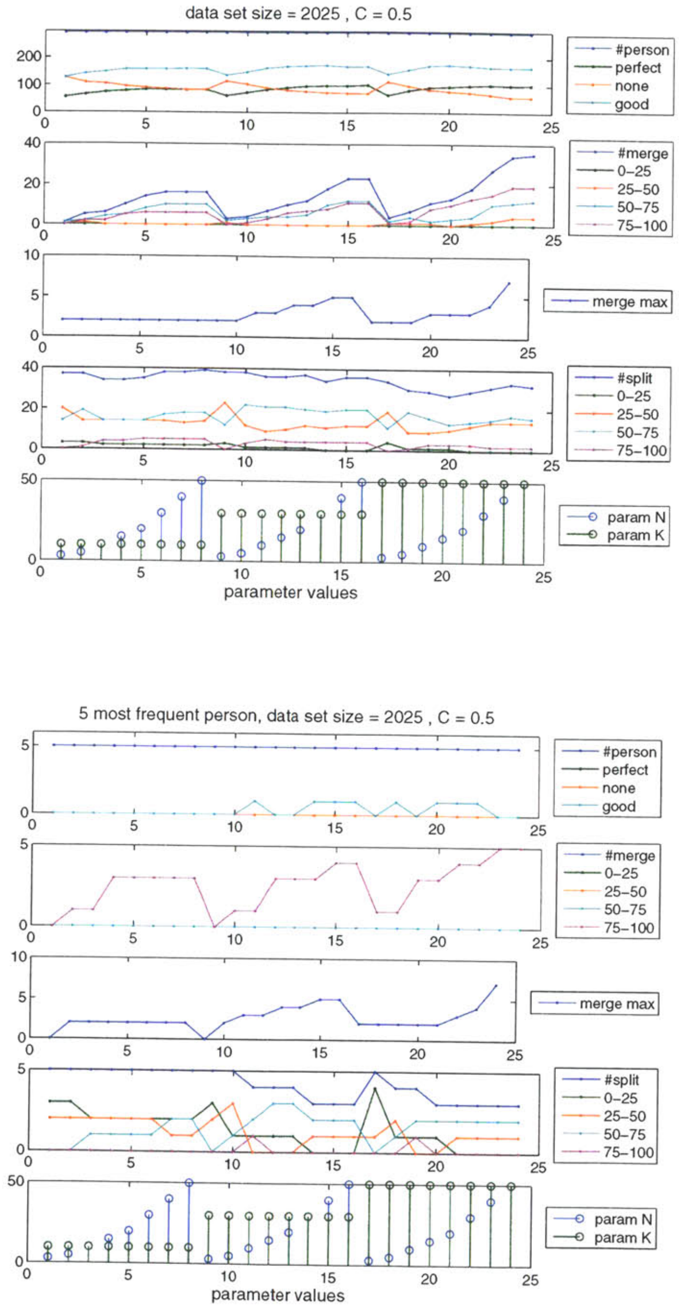


Figure C-33: Results of clustering evaluation of a data set of 2025 sequences, C = 50%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

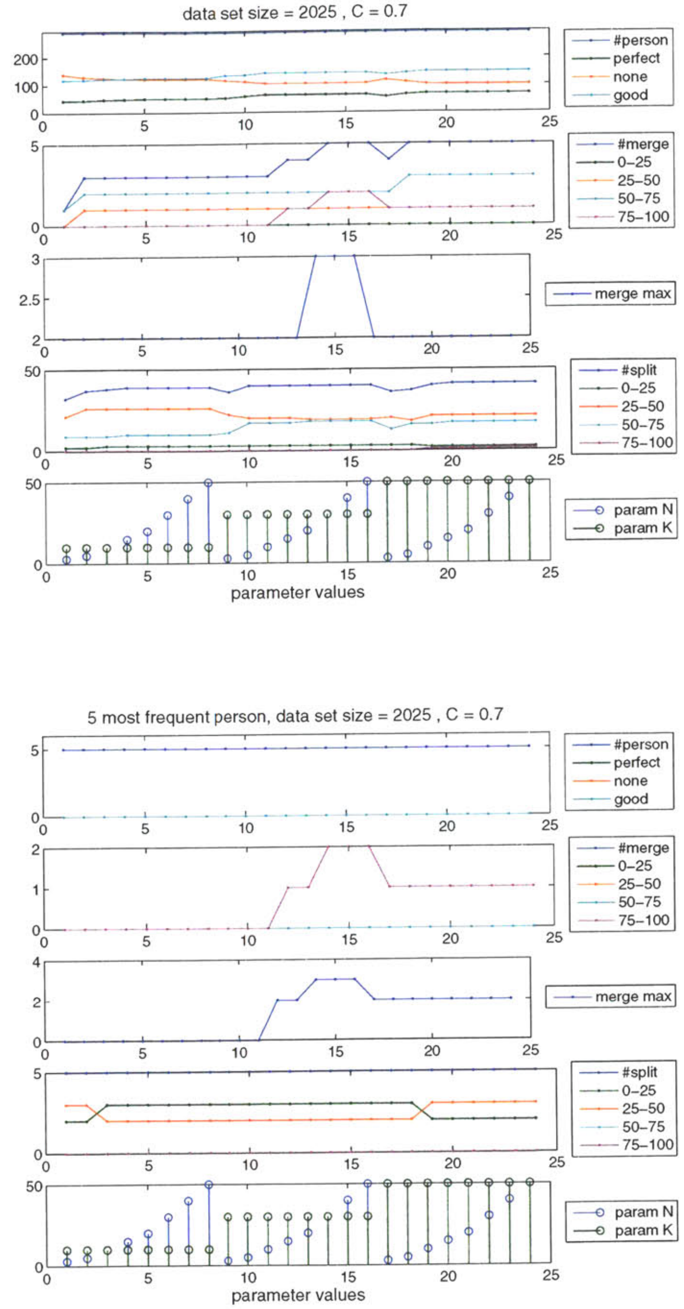


Figure C-34: Results of clustering evaluation of a data set of 2025 sequences, C = 70%. The upper figure shows the results from the entire data set and the lower figure shows a subset of the results from 5 individuals who have the most number of sequences in the data set.

Appendix D

Sample Clusters of the Familiar Individuals

The following is a set of sample clusters of familiar individuals which were constructed during the incremental recognition process, as described in section 5.4. Some sample clusters have already been shown earlier in section 5.4.3. Note that not all of the sequences are shown in these figures, due to visibility and space constraints. However, we make sure to include all of the merged errors, when present.

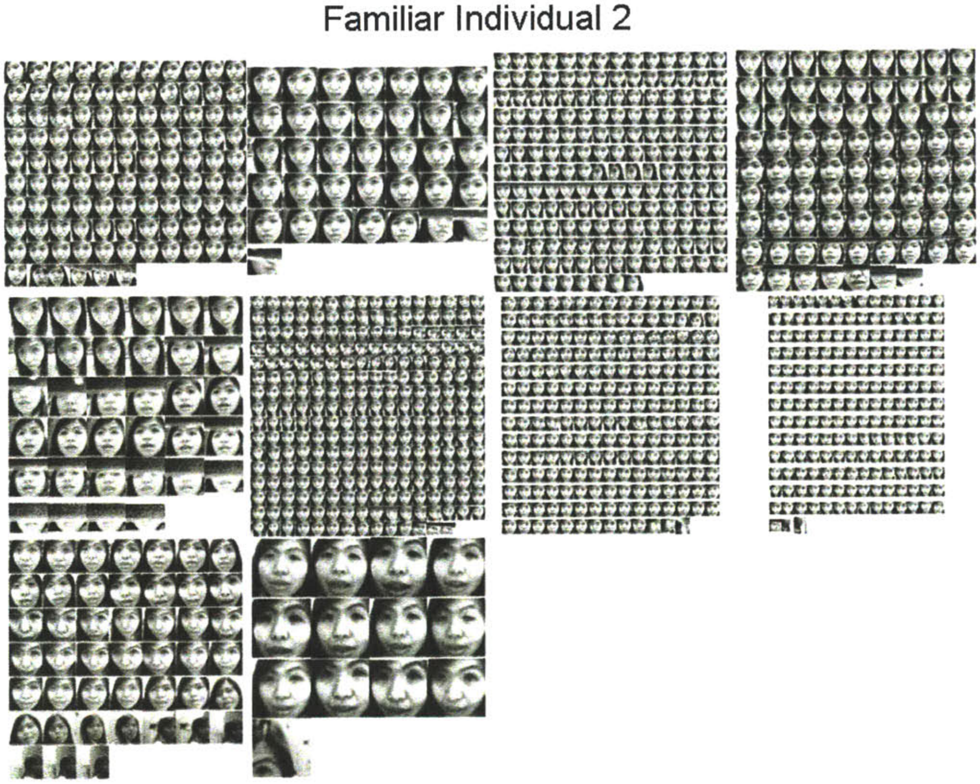


Figure D-1: The generated cluster of familiar individual 2. Individual 2 is the same person as individual 1, shown in section 5.4.3

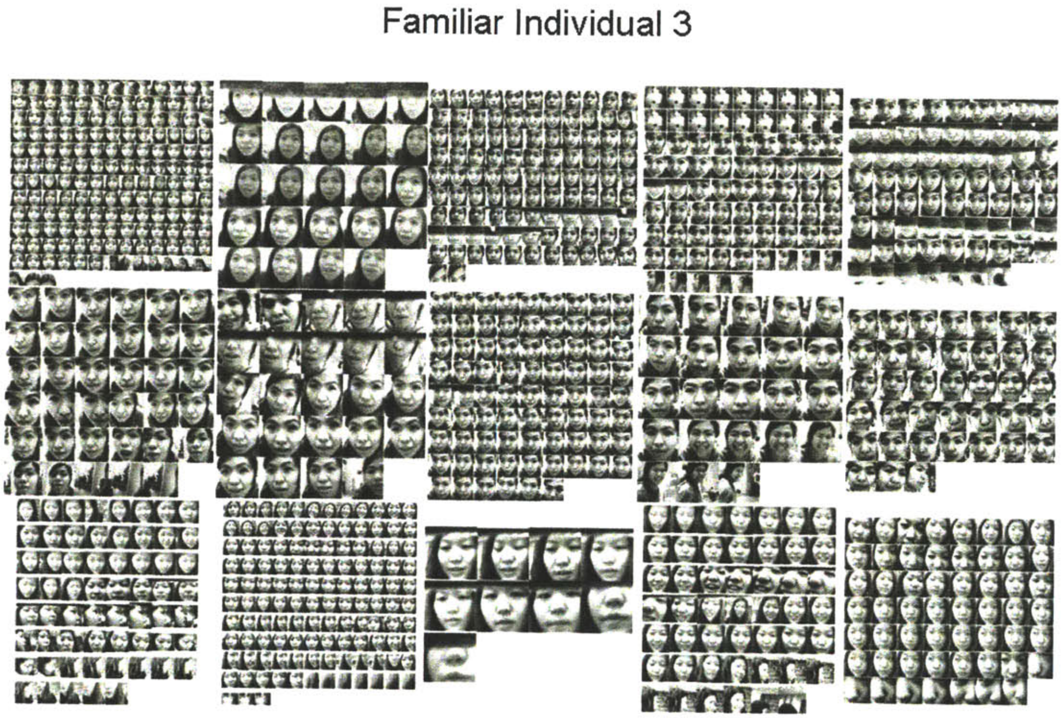


Figure D-2: The generated cluster of familiar individual 3. Individual 3 is the same person as individual 1, shown in section 5.4.3. Note that the five sequences in the last row are falsely merged from another person.

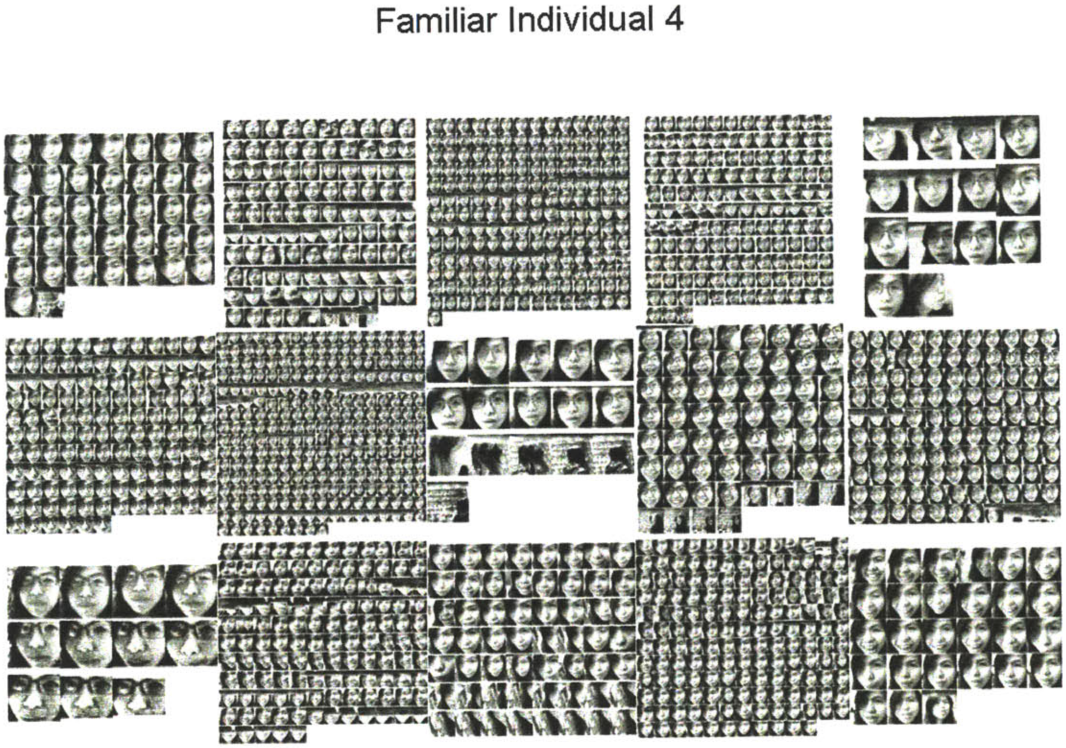


Figure D-3: The generated cluster of familiar individual 4. Individual 4 is the same person as individual 1, shown in section 5.4.3.

Familiar Individual 6

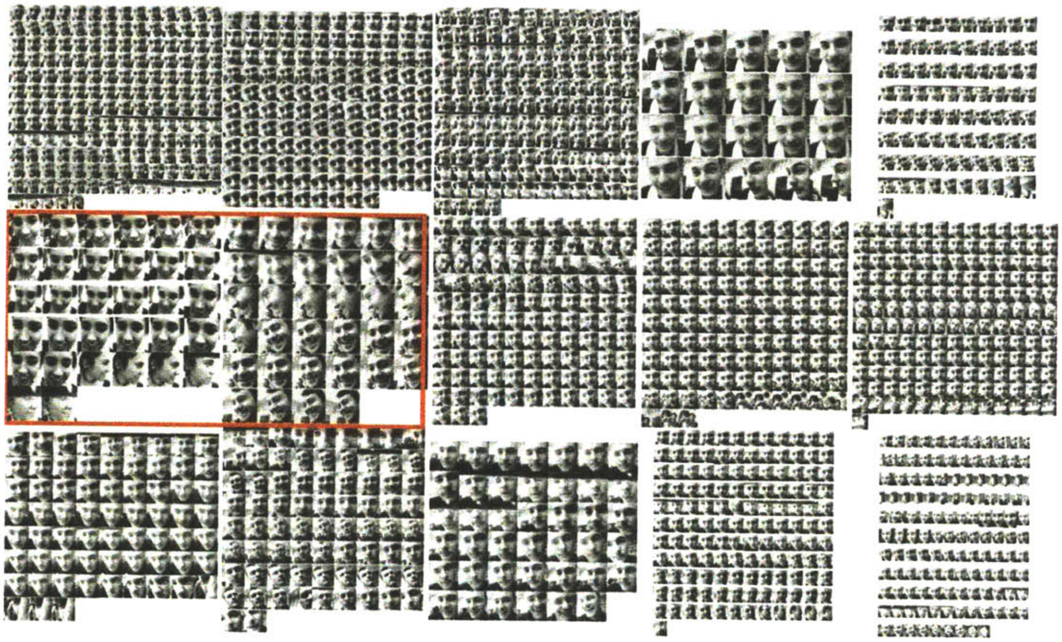


Figure D-4: The generated cluster of familiar individual 6. Individual 6 came to interact with the robot on two different days during the experiment. The red rectangle is placed to point out the falsely merged sequences from another individual in the cluster.

Familiar Individual 9

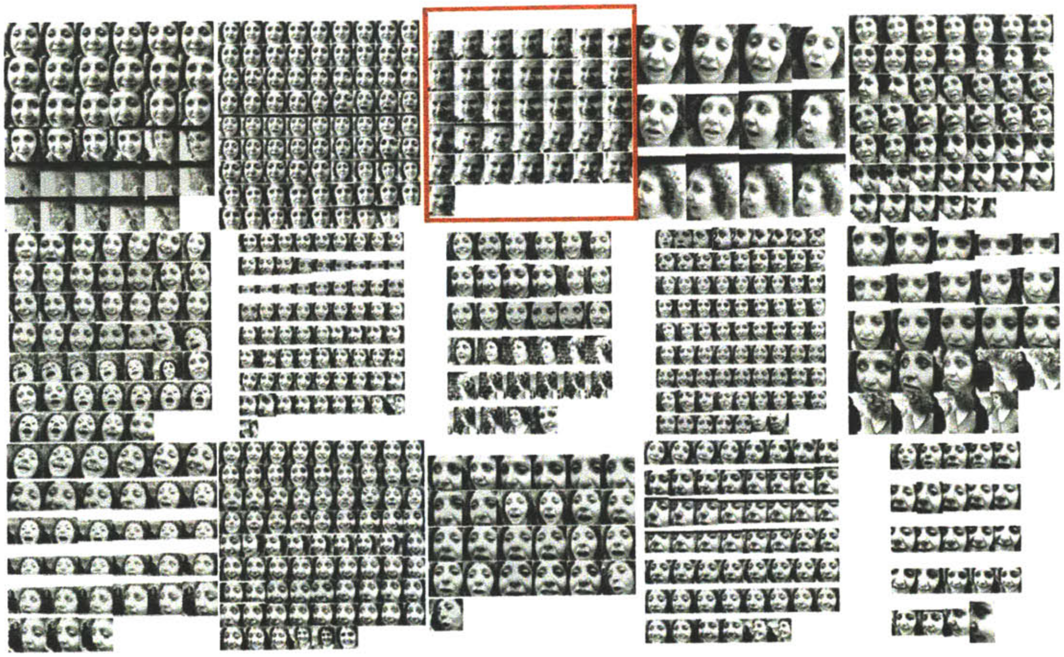


Figure D-5: The generated cluster of familiar individual 9. The red rectangle is placed to point out the falsely merged sequences from another individual in the cluster.

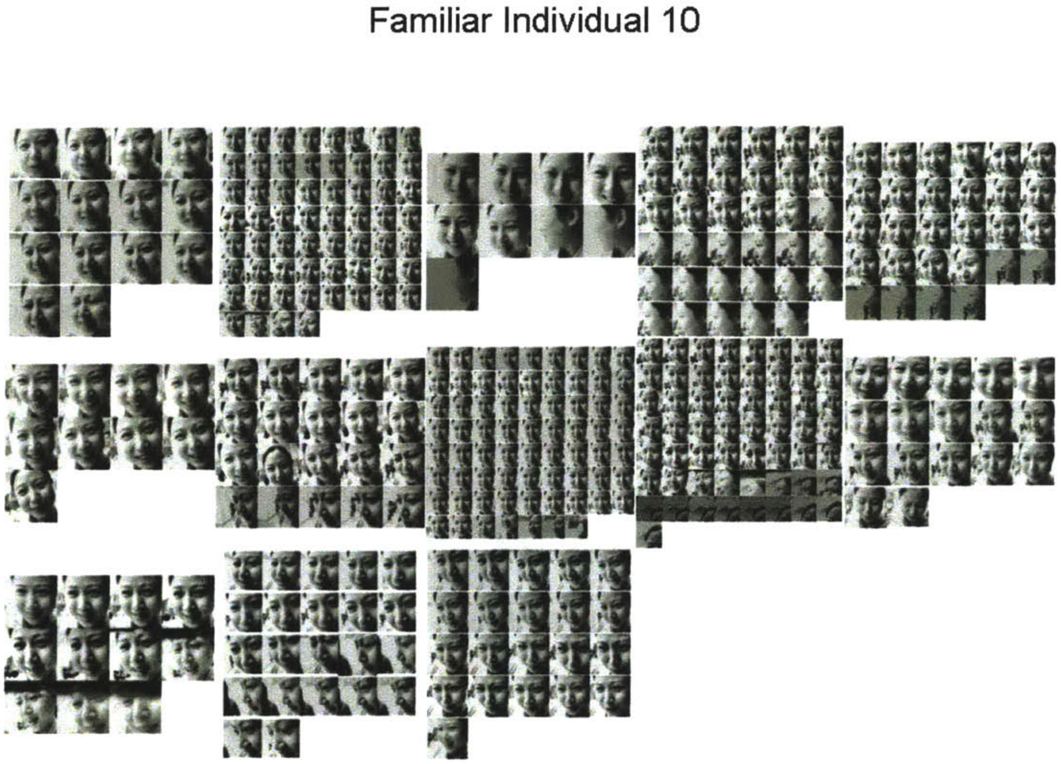


Figure D-6: The generated cluster of familiar individual 10.

Bibliography

[1] J.S. Albus. Outline for a theory of intelligence. *IEEE Transactions on Sysems, Man, and Cybernetics, vol. 21, no. 3*, 1991.

[2] R. Arkin. *Behavior-based Robotics.* MIT Press, Cambridge, Massachusetts, 1998.

[3] C. Bartneck and J. Forlizzi. Shaping human-robot interaction: understanding the social aspects of intelligent robotic products. *Conference on Human Factors in Computing Systems*, 2004.

[4] J. Bates. The role of emotion in believable agents. *Communications of the ACM, Special Issue on Agents*, 1994.

[5] T. Berg, A. Berg, J. Edwards, M. Maire, R. White, Y. Teh, E. Learned-Miller, and D. Forsyth. Names and faces in the news. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, 2004.

[6] M. Bicego, A.Lagorio, E. Grosso, and M. Tistarelli. On the use of sift features for face authentication. *Proc. of IEEE Int Workshop on Biometrics*, 2006.

[7] S. Birchfield. Klt: An implementation of the kanade-lucas-tomasi feature tracker. http://www.ces.clemson.edu/ stb/klt/.

[8] R. Bischoff and V. Graefe. Design principles for dependable robotic assistants. *International Journal of Humanoid Robotics, vol. 1, no. 1 95 125*, 2004.

[9] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(9), 2003.

[10] G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal, Q2:1-15*, 1998.

[11] C. Breazeal. Sociable machines: Expressive social exchange between humans and robots. *Sc.D. dissertation, Department of EECS, MIT*, 2000.

[12] C. Breazeal. Emotion and sociable humanoid robots. *International Journal of Human-Computer Studies, 59, pp.119-155*, 2003.

[13] C. Breazeal, A. Brooks, J. Gray, G. Hoffman, C. Kidd, H. Lee, J. Lieberman, A. Lockerd, and D. Mulanda. Humanoid robots as cooperative partners for people. *International Journal of Humanoid Robotics*, 2004.

[14] C. Breazeal and B. Scassellati. A context-dependent attention system for a social robot. *Proceedints of the Sixteenth International Joint Conference on Artificial Intelligence*, 1999.

[15] Cynthia Breazeal, Aaron Edsinger, Paul Fitzpatrick, Brian Scassellati, and Paulina Varchavskaia. Social constraints on animate vision. *IEEE Intelligent Systems*, 15(4):32–37, 2000.

[16] R. Brooks. Intelligence without representation. *Artificial Intelligence Journal 47:139-160*, 1991.

[17] R. Brooks, C. Breazeal, M. Marjanovic, B. Scassellati, and M. Williamson. The cog project: Building a humanoid robot. *Computation for Metaphors, Analogy and Agents, Vol. 1562 of Springer Lecture Notes in Artificial Intelligence, Springer-Verlag*, 1998.

[18] R.A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation 2:1:14–23*, 1986.

[19] R.A. Brooks and C. Rosenberg. L -a common lisp for embedded systems. *Association of Lisp Users Meeting and Workshop*, 1995.

[20] A. Bruce, I. Nourbakhsh, and R. Simmons. The role of expressiveness and attention in human-robot interaction. *Proc. AAAI Fall Symp. Emotional and Intel. II: The Tangled Knot of Soc. Cognition*, 2001.

[21] R. Brunelli and D. Falavigna. Person identification using multiple cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, No. 10*, 1995.

[22] W. Burgard, D. Fox, D. Hdhnel, G. Lakemeyer, D. Schulz, W. Steiner, S. Thrun, and A.B. Cremers. Real robots for the real world – the rhino museum tour-guide project. *Proc. of the AAAI Spring Symposium on Integrating Robotics Research, Taking the Next Leap, Stanford, CA*, 1998.

[23] N. Butko, I. Fasel, and J. Movellan. Learning about humans during the first 6 minutes of life. *Proceedings of the 5th Internation Conference on Development and Learning*, 2006.

[24] R. Caldwell. Recognition, signalling and reduced aggression between former mates in a stomatopod. *Animal Behavior 44,11-19*, 1992.

[25] T. Choudhury, B. Clarkson, T. Jebara, and A. Pentland. Multimodal person recognition using unconstrained audio and video. *Proceedings of the Second Conference on Audio- and Video-based Biometric Person Authentication*, 1999.

[26] K. Dautenhahn. Getting to know each other - artificial social intelligence for autonomous robots. *Robotics and Autonomous Systems 16:333-356*, 1995.

[27] M. S. Dawkins. Distance and social recognition in hens: Implications of the use of photographs as social stimuli. *Behavior 133:9-10,663-680*, 1996.

[28] J. Dewey. *Experience and education.* New York: Macmillan, 1938.

[29] C. DiSalvo, F. Gemperle, J. Forlizzi, and S. Kiesler. All robots are not created equal: The design and perception of humanoid robot heads. *Conference Proceedings of Designing Interactive Systems, London, England*, 2002.

[30] S. Eickeler, F. Wallhoff, U. Iurgel, and G. Rigoll. Content-based indexing of images and videos using face detection and recognition methods. *IEEE Int. Conference on Acoustics, Speech, and Signal Processing*, 2001.

[31] W. Fisher. Program tsylb (version 2 revision 1.1). NIST, 7 August 1996.

[32] T. Fong, I. Nourbakhsh, and K. Dautenhahn. A survey of socially interactive robots. *Robotics and Autonomous Systems 42:143-166*, 2003.

[33] G.L. Foresti and C. Micheloni. Real-time video-surveillance by an active camera. *Ottavo Convegno Associazione Italiana Intelligenza Artificiale (AI*IA) - Workshop sulla Percezione e Visione nelle Macchine, Universita di Siena, September 11-13*, 2002.

[34] Simone Frintrop, Gerriet Backer, and Erich Rome. Selecting what is important: Training visual attention. *Proceedings of the 28th German Conference on Artificial Intelligence*, 2005.

[35] S. Furui. An overview of speaker recognition technology. *ESCA Workshop on Automatic Speaker Recognition Identification Verification*, 1994.

[36] R. Gockley, A. Bruce, J. Forlizzi, M. Michalowski, A. Mundell, S. Rosenthal, B. Selner, R. Simmons, K. Snipes, A. Schultz, and J. Wang. Designing robots for long-term social interaction. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

[37] M. Goodale and A. Milner. Separate pathways for perception and action. *Trends in Neuroscience 15: 20-25*, 1992.

[38] D. Gorodnichy. Video-based framework for face recognition in video. *Proc. of the Second Canadian Conference on Computer and Robot Vision*, 2005.

[39] C. Harris and M. Stephens. A combined corner and edge detector. *Alvey Vision Conference, pp 147-152*, 1988.

[40] R. Hewitt and S. Belongie. Active learning in face recognition: Using tracking to build a face model. *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop*, 2006.

[41] R.A. Peters II, K.E. Hambuchen, K. Kawamura, and D.M. Wilkes. The sensory egosphere as a short-term memory for humanoids. *Proc. 2nd IEEE-RAS International Conference on Humanoid Robots, pp 451-459*, 2001.

[42] L. Itti. Models of bottom-up and top-down visual attention. *Ph.D. Thesis, California Institute of Technology*, 2000.

[43] T. Kanda, T. Hirano, D. Eaton, and H. Ishiguro. Interactive robots as social partners and peer tutors for children: A field trial. *Human Computer Interaction, Vol. 19, No. 1-2, pp. 61-84*, 2004.

[44] T. Kanungo, D. M. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu. Efficient algorithms for k-means clustering. http://www.cs.umd.edu/ mount/Projects/KMeans/.

[45] T. Kanungo, D. M. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Analysis and Machine Intelligence, 24:881-892*, 2002.

[46] J. Kittler, Y. Li, J. Matas, and M. Ramos Sanchez. Combining evidence in multimodal personal identity recognition systems. *International Conferrence on Audio and Video-based Biometric Person Authentication*, 1997.

[47] H. Kozima. Infanoid: A babybot that explores the social environment. *Socially Intelligent Agents: Creating Relationships with Computers and Robots, Amsterdam: Kluwer Academic Publishers, pp.157-164*, 2002.

[48] H. Kozima, C. Nakagawa, and Y. Yasuda. Interactive robots for communication-care: A case-study in autism therapy. *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication*, 2005.

[49] E. Labinowicz. *The Piaget Primer: Thinking, Learning, Teaching.* Addison-Wesley, 1980.

[50] K.C. Lee, J. Ho, M.H. Yang, and D. Kriegman. Video-based face recognition using probabilistic appearance manifolds. *IEEE Conf. On Computer Vision and Pattern Recognition*, 1:313–320, 2003.

[51] K.C. Lee, J. Ho, M.H. Yang, and D. Kriegman. Visual tracking and recognition using probabilistic appearance manifolds. *Computer Vision and Image Understanding*, 99:303–331, 2005.

[52] D.B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM 38, no. 11, November*, 1995.

[53] X. Liu and T. Chen. Video-based face recognition using adaptive hidden markov models. *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[54] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60, 2, pp. 91-110*, 2004.

[55] J. Luettin. *Visual Speech and Speaker Recognition.* PhD thesis, University of Sheffield, 1997.

[56] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini. Developmental robotics: A survey. *Connection Science, vol.00 no.0:1-40*, 2004.

[57] S. Macskassy and H. Hirsh. Towards personable agents: A survey. *1998.*

[58] J. Mateo. Recognition systems and biological organization: The perception component of social recognition. *Annales Zoologici Fennici, 41, 729-745*, 2004.

[59] G. Metta, P. Fitzpatrick, and L. Natale. Yarp: Yet another robot platform. *International Journal of Advanced Robotics Systems, special issue on Software Development and Integration in Robotics, Volume 3(1)*, 2006.

[60] A. S. Mian, M. Bennamoun, and R. Owens. Face recognition using 2d and 3d multimodal local features. *International Symposium on Visual Computing*, 2006.

[61] H. Miwa, T. Okuchi, K. Itoh, H. Takanobu, and A. Takanishi. A new mental model for humanoid robots for human friendly communication introduction of learning system, mood vector and second order equations of emotion. *Robotics and Automation*, 2003.

[62] H. Miwa, T. Okuchi, H. Takanobu, and A. Takanishi. Development of a new human-like head robot we-4r. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.2443-2448*, 2002.

[63] M. Montemerlo, J. Pineau, N. Roy, S. Thrun, and V. Verma. Experiences with a mobile robotic guide for the elderly. *Proceedings of the AAAI National Conference on Artificial Intelligence*, 2002.

[64] M. Mori. The buddha in the robot. *Charles E. Tuttle Co.*, 1982.

[65] R. Mosur. Sphinx-ii user guide.

[66] D. Mou. *Autonomous Face Recognition*. PhD thesis, University of Ulm, 2005.

[67] J. Movellan, F. Tanaka, B. Fortenberry, and K. Aisaka. The rubi/qrio project, origins, principles, and first steps. *International Journal of Human-Computer Studies, 59, pp.119-155.*, 2005.

[68] C. Nakajima, M. Pontil, B. Heisele, and T. Poggio. People recognition in image sequences by supervised learning. *A.I. Memo No. 1688, C.B.C.L. Paper No. 188. MIT*, 2000.

[69] C. Nass and S. Brave. Emotion in human-computer interaction. *in The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications, Lawrence Erlbaum Associates*, 2002.

[70] L. Natale. *Linking Action to Perception in a Humanoid Robot: A Developmental Approach to Grasping.* PhD thesis, University of Genova, 2004.

[71] I. Nourbakhsh, C. Kunz, and T. Willeke. The mobot museum robot installations: A five year experiment. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas*, 2003.

[72] T. Ogata, Y. Matsuyama, T. Komiya, M. Ida, K. Noda, and S. Sugano. Development of emotional communication robot, wamoeba-2r-experimental evaluation of the emotional communication between robots and humans. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000.

[73] S. Palmer. Vision science: Photons to phenomenology. *MIT Press*, 1999.

[74] R. Pfeifer and J. Bongard. *How the Body Shapes the Way We Think.* MIT Press, 2006.

[75] P. Phillips, P. Grother, R. Micheals, D. Blackburn, E. Tabassi, and J. Bone. Frvt 2002 evaluation report. 2003.

[76] P. Phillips, H. Moon, S. Rizvi, and P. Rauss. The feret evaluation methodology for face recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22 no. 10, pp 1090-1104*, 2000.

[77] J. Piaget. *The Origins of Intelligence in Children.* Routledge and Kegan Paul, 1953.

[78] R. Porter, L. Desire, R. Bon, and P. Orgeur. The role of familiarity in the development of social recognition by lambs. *Behavior 138:2,207-219*, 2004.

[79] M. I. Posner. Orienting of attention. *Quarterly Journal of Experimental Psychology, 32:3-25*, 1980.

[80] J.E. Pratt. Virtual model control of a biped walking robot. *Tech. Rep. AITR-1581, MIT Artificial Intelligence Laboratory, Cambridge, MA, USA*, 1995.

[81] B. Raytchev and H. Murase. Unsupervised face recognition by associative chaining. *Pattern Recognition, Vol. 36, No. 1, pp. 245-257*, 2003.

[82] Intel Research. Open source computer vision library. http://www.intel.com/technology/computing/opencv/.

[83] L. Sayigh, P. Tyack, R. Wells, A. Solow, M. Scott, and A. Irvine. Individual recognition in wild bottlenose dolphins: a field test using playback experiments. *Animal Behavior 57(1):41-50*, 1999.

[84] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences 3:233-242*, 1999.

[85] M. Schwebel. The role of experience in cognitive development. Presented at the 5th Invitational Interdisciplinary Seminar , University of Southern California, 1975.

[86] S. Shan, Y. Chang, and W. Gao. Curse of misalignment in face recognition: Problem and a novel misalignment learning solution. *IEEE International Conference on Automatic Face and Gesture Recognition*, 2004.

[87] J. Shi and C. Tomasi. Good features to track. *IEEE Conference on Computer Vision and Pattern Recognition, pages 593-600*, 1994.

[88] M. Shiomi, T. Kanda, H. Ishiguro, and N. Hagita. Interactive humanoid robots for a science museum. *ACM 1st Annual Conference on Human-Robot Interaction*, 2006.

[89] R. Siegwart, K. Arras, S. Bouabdallah, D. Burnier, G. Froidevaux, X. Greppin, B. Jensen, A. Lorotte, L. Mayor, M. Meisser, R. Philippsen, R. Piguet,

G. Ramel, G. Terrien, and N. Tomatis. Robox at expo.02: A large-scale installation of personal robots. *Robotics and Autonomous Systems 42(3-4): 203-222*, 2003.

[90] T. Sim and S. Zhang. Exploring face space. *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop*, 2004.

[91] R. Simmons, D. Goldberg, A. Goode, M. Montemerlo, N. Roy, B. Sellner, C. Urmson, A. Schultz, M. Abramson, W. Adams, A. Atrash, M. Bugajska, M. Coblenz, M. MacMahon, D. Perzanowski, I. Horswill, R. Zubek, D. Kortenkamp, B. Wolfe, T. Milam, and B. Maxwell. Grace: An autonomous robot for the aaai robot challenge. *AI Magazine, 24(2)*, 2003.

[92] J. Stanley and B. Steinhardt. Drawing a blank : The failure of face recognition in tampa. An ACLU Special Report, 2002.

[93] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision, 7:1*, 1991.

[94] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Halnel, C. Rosenbert, N. Roy, J. Schultze, and D. Schulz. Minerva: A second-generation museum tour-guide robot. *Proceedings of IEEE International Conference on Robotics and Automation, vol 3, pages 1999-2005*, 1999.

[95] A.M. Turing. Computing machinery and intelligence. *Mind, 59, 433-460*, 1950.

[96] M. Turk and A. Pentland. Face recognition using eigenfaces. *Proceedings IEEE Conference Computer Vision and Pattern Recognition*, 1991.

[97] J. Velasquez. When robots weep: Emotional memories and decision-making. *Proceedings of the Fifteenth National Conference on Artificial Intelligence. Madison, WI*, 1998.

[98] P. Viola and M. Jones. Robust real-time object detection. *Technical Report Series, CRL2001/01. Cambridge Research Laboratory*, 2001.

[99] L. S. Vygotsky. Mind in society. *Cambridge, MA: Harvard Univ Press*, 1978.

[100] C. Wallraven, A. Schwaninger, and H. H. Blthoff. Learning from humans: Computational modeling of face recognition. *Network: Computation in Neural Systems 16(4), 401-418*, 2005.

[101] P. Wang, M. Green, Q. Ji, and J. Wayman. Automatic eye detection and its validation. *IEEE Workshop on Face Recognition Grand Challenge Experiments*, 2005.

[102] J. Weng. Developmental robotics: Theory and experiments. *International Journal of Humanoid Robotics, vol. 1, no. 2*, 2004.

[103] J. Weng, C. Evans, and W. Hwang. An incremental learning method for face recognition under continuous video stream. *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 2000.

[104] C. Yu and D.H. Ballard. Exploring the role of attention in modeling embodied language acquisition. *Proceedings of the Fifth International Conference on Cognitive Modeling*, 2003.

[105] W. Zhao, R. Chellappa, A. Rosenfeld, and P. Phillips. Face recognition: A literature survey. *ACM Computing Surveys, Volume 35, Issue 4, pp. 399-458, December*, 2003.

[106] S. Zhou, V. Krueger, and R. Chellappa. Probabilistic recognition of human faces from video. *Computer Vision and Image Understanding, 91:214-245*, 2003.