Chapter in "Motion-Based Recognition", Editors M. Shah and R. Jain, Kluwer Academic Publihsers, 1997.

VISUAL RECOGNITION OF ACTIVITIES, GESTURES, FACIAL EXPRESSIONS AND SPEECH: AN INTRODUCTION AND A PERSPECTIVE

MUBARAK SHAH
Computer Vision Lab
Computer Science Department
University of Central Florida
Orlando, FL 32816

AND

RAMESH JAIN
Electrical and Computer Engineering
University of California, San Diego
La Jolla, CA 92093-0407

1. Introduction

Computer vision has started migrating from the peripheral area to the core of computer science and engineering. Multimedia computing and natural human-machine interfaces are providing adequate challenges and motivation to develop techniques that will play key role in the next generation of computing systems. Recognition of objects and events is very important in multimedia systems as well as interfaces. We consider an object a *spatial* entity and an event a *temporal* entity. Visual recognition of objects and activities is one of the fastest developing area of computer vision.

Objects and events must be recognized by analyzing images. An *image* is an array of numbers representing the brightness of a scene, which depends on the camera, light source, and objects in the scene. Images look different from each other mainly due to the fact that they contain different objects. The most important visual attribute which distinguishes one object from the other is its *shape*. The shape represents the geometry of the object,

[1]The first author acknowledges the support of DoD STRICOM under Contract No. N61339-96-K-0004. The content of the information herein does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred.

which can be 2-D or 3-D. Edges, lines, curves, junctions, blobs, regions, etc. can be used to represent 2-D shape. Similarly, planes, surface patches, surface normals, cylinders, super-quadrics, etc. can be used to represent 3-D shape.

The shape plays the most dominant role in model-based *recognition*. In the most simple case of recognition, 2-D models and 2-D input are used. In this case, the computations are simple, but 2-D shape can be ambiguous; for several 3-D shapes may project to the same 2-D shape. In the the most general case of recognition, 3-D models and 3-D input are used, however, this approach is computationally expensive. In between is the approach in which 3-D models and 2-D input are used. In this case, the object's pose (3-D rotation and translation) of the model needs to be computed such that when it is projected on the image plane it exactly matches with the input.

A *sequence* of images represents how images change due to *motion* of objects, camera, or light source; or due to any of the two, or due to all three. Among all sequence of images, the most common sequences are the sequences which depict the motion of objects. The motion can be represented using difference pictures, optical flow, trajectories, spatiotemporal curvature, joint curves, muscle actuation, etc. Motion can also be represented by the deformation of shape with respect to time.

This book is about *motion-based recognition*. Motion-based recognition deals with the identification of objects or motion based on object's motion in a sequence [6]. In motion-based recognition, the motion is directly used in recognition, in contrast to the standard structure from motion (sfm) approach, where recognition follows reconstruction. Consequently, in some cases, it is not necessary to recover the structure from motion. Another important point here is that it is crucial to use a large number of frames; for it is almost impossible to extract meaningful motion characteristics using just two or three frames. There exists a distinction between *motion-based recognition* and *motion recognition*: motion-based recognition is a general approach that favors the use of motion information for the purpose of recognition, while motion recognition is one goal that can be attained with that approach.

This is an exciting research direction, which will have ever lasting effects on Computer Vision research. In the last few years, many exciting ideas have started appearing, but at disparate places in literature. Therefore, to provide the state of art in motion-based recognition at one place, we have collected key papers in this book. It consists of a collection of invited chapters by leading researchers in the world who are actively involved in this area.

The book is divided into three main parts: human activity recognition, gesture recognition and facial expression recognition, and lipreading. The

next three sections introduce each part of the book, and summarize the chapters included.

2. Human Activity Recognition

Automatically detecting and recognizing human activities from video sequences is a very important problem in motion-based recognition. There are several possible applications of activity recognition. One possible application is in automated video surveillance and monitoring, where human visual monitoring is very expensive, and not practical. One human operator at a remote host workstation may supervise many automated video surveillance systems. This may include monitoring of sensitive sites for unusual activity, unauthorized intrusions, and triggering of significant events. Another area is detection and recognition of animal motion, with the primary purpose of discriminating it from the human motion in surveillance applications. Video games could be made more realistic using an activity recognition system, where the players control navigation by using their own body movements. A virtual dance or aerobics instructor could be developed that watches different dance movements or exercises and offers feedback on the performance. Other applications include athlete training, clinical gait analysis, military simulation, traffic monitoring, video annotations (most videos are about people) and human-computer interface.

Given a sequence of images, usually the first step in activity recognition is to detect a motion in a sequence; if the sequence represents a stationary scene, there is no point analyzing such a sequence for activity recognition. Difference pictures have been widely used for motion detection since the original paper of Jain *et al* [18]. The simple difference picture has some limitations, for instance, with covering and uncovering of image regions. Several ways to deal with this limitations has been reported in the literature. An example is to use the difference between the stationary background image and the current image, resulting only in the moving components. Here, the difficulty is how to obtain the background image. One possibility is to use the first image taken before the objects start moving in front of the camera as a background image. The other possibility is to reconstruct the background image for each pixel, by applying the median filter to all pixels gray values at a given location in a sequence; this is more general, but it is time consuming. In fact, in numerous application, the stationary background image is very easily available and can be used effectively.

The difference picture identifies the changed pixels in an image. The changed pixels need to be grouped into regions corresponding to the human body using a connected component algorithm. Also, due to the non-rigid nature of human body, there may be several small adjacent regions, which

need to be merged.

Some variations of this change detection method include: computing the difference in a small neighborhood (e.g. 5×5) around a pixel instead of a pixel by pixel difference; or computing the difference between the current and the previous two and the next two frames (accumulated difference picture), as compared to the difference between just two frames (current and previous frames).

The optical flow has been widely used in motion analysis work to extract motion information from a sequence of images. Optical flow computes displacement vectors for each pixel between frames. One problem with optical flow, in general, is that it is susceptible to the aperture problem, which, in some conditions, only allows the precise computation of the *normal flow*, i.e. the component parallel to the gradient. Therefore, several researchers in motion-based recognition have employed normal flow, instead of full flow.

Human motion occurs at a variety of scales from fine to coarse (e.g., motion of lips to motion of legs). In the second chapter of this book, Yacoob and Davis present a scale-space based optical flow method for computing motion from sequences of humans. Their method first uses a simple model of optical flow, which assumes the optical flow to be constant, and employs robust estimation. However, human motion contains an acceleration component as well. Therefore, they extend their model to include acceleration. They also extend their approach to compute optical flow using parameterized models: affine and planar models.

Both change detection and optical flow are bottom-up approaches. An alternate approach is to use some *a priori* knowledge about the object being tracked. Snakes provide means to introduce a priori knowledge [20, 32]. The user-imposed constraint forces can guide the snakes near features of interest. Recently, another approach, using active shape models, was proposed, which use the Point Distribution Model (PDM) to build models by learning patterns of variability from a training set [8]. The major difference between snakes and PDM is that in PDM the shape can deform only when it is consistent to the training sets.

In chapter three of this book, Baumberg and Hogg present a method for automatically generating deformable 2-D contour models for tracking human body using the PDM approach. The conventional PDM approach requires a hand generated set of labeled points from the training images. The PDM approach is extended to automate the process of extracting a training set and building the model automatically. The results on tracking sequences of humans with scaling, change of view, translation and rotation are shown. In this approach, a B-spline is used to represent contour, and real time tracking is performed using a Kalman filter framework .

Many motions in nature are cyclic, like walking, a heartbeat, birds fly-

ing, a pendulum swinging, etc. The presence of cyclic motion in a sequence of images can reveal a lot about the object showing that type of motion. The cyclic motion detection problem was first introduced in Computer Vision by Allmen and Dyer in 1990. Based on studies of the human visual system, Allmen and Dyer [4] and Allmen [3] argue that cyclic motion detection: (1) does not depend on prior recognition of the moving object, i.e. cycles can be detected even if the object is unknown; (2) does not depend on the absolute position of the object; (3) needs long sequences (at least two complete cycles); (4) is sensitive to different scales, i.e. cycles at different levels of a moving object can be detected. They detected cyclic motion by identifying cycles in the curvature of a spatiotemporal curve using some form of A* algorithm. Polana and Nelson, and Tsai *et al* [30] proposed methods for cyclic motion detection using the Fourier transform. Polana and Nelson (see chapter five) first compute what they call the reference curve, which essentially is a linear trajectory of a centroid, the frames are aligned with respect to this trajectory. If the object presented some periodic motion, it will create some periodic gray level signals. They use the Fourier transform to compute periodicity of those gray level signals. The periodicity measure of several gray level signals are combined using some form of non maxima suppression.

The problem with these approaches to cyclic motion is that they only deal with strictly cyclic motions; the motions that repeat but are not regular are not dealt with. More important, these methods are not view invariant, they do not allow the camera to move. In chapter four, Seitz and Dyer describe view-independent analysis of cyclic motion using affine invariance. They introduce *period trace*, which gives a set of compact descriptions of near periodic signals.

There are two classes of approaches for human activity recognition: 3-D and 2-D. In a 3-D approach, some 3-D model of the human body and human motion is used. A projection of the model from a particular pose and particular posture in a cycle of activity is compared with each input image to recognize an activity. The advantage of this approach is that since a 3-D model is used it is not ambiguous. However, it is computationally quite expensive. Hogg [15] was the first to use a 3-D model-based approach for tracking humans. Instead of using a 3-D model and 2-D input, another approach is to use a 3-D input and 3-D model. Bobick *et al* [5] use 3-D point data to recognize ballet movements.

In 2-D approaches, no model of a 3-D body is used, only 2-D motion, e.g. optical flow, is employed to compute features in a sequence of frames to recognize activities. The advantage of this approach is that it is quite simple. In this book (chapters five to seven), the approaches of Polana and Nelson, Bobick and Davis, and Goddard are basically 2-D approaches.

Besides recognizing activities from motion, the motion can also be used to recognize people by their gaits. From our own experience, it is relatively easy to recognize a friend from the way he or she walks, even though this person is at a distance so that the face features are not recognizable. Rangarajan *et al* [25] describe a method for recognizing people by their gaits. They use trajectories of joints of a human body performing walking motions. Niyogi and Adelson [1] has proposed a method based on XT-trace to discriminate people. Boyd and Little [21] has described a method for recognizing people based on the phase of the weighted centroid of the human body.

Most approaches employ only one camera to capture the activities of a person. It may happen that due to self-occlusion, some parts of a person are not visible in the image, which may result in not having enough information for recognition. Another possible problem is that due to the limited field of view of one camera, the person may move out of the field of view of the camera. In order to deal with these difficulties some researchers have advocated the use of multiple cameras [22, 12, 27, 2]. However, with the introduction of additional cameras there is additional overhead, and we need to answer the following questions: How many views should be employed? Should information from all cameras be used or from only one? How to associate the image primitives among images obtained by multiple cameras? etc.

In chapter five of this book, Polana and Nelson classify motion into three categories: *events*, *temporal textures*, and *activities*. The events consists of isolated simple motions that do not exhibit any temporal or spatial repetition. Examples of motion events are opening a door, starting a car, throwing a ball, etc. The temporal textures exhibit statistical regularity but are of indeterminate spatial and temporal extent. Examples include ripples on water, the wind in the leaves of trees, a cloth waving in the wind. Activities consists of motion patterns that are temporally periodic and possess compact spatial structure. Four features of normal flow: the mean flow magnitude divided by its standard deviation, the positive and negative curl and divergence, the non-uniformity of flow direction, and the directional difference statistics in four directions are used to recognize temporal textures. Their activity recognition approach also uses normal flow. First, cycles in sequences of frame are detected using Fourier transform of reference curves. Each image is divided into a spatial grid of $X \times Y$ divisions. Each activity cycle is divided into T time divisions, and motion is totaled in each temporal division corresponding to each spatial cell separately. The feature vector is formed from these spatiotemporal cells, and used in a nearest centroid algorithm to recognize activities.

Bobick and Davis, in chapter six, first apply change detection to iden-

tify moving pixels in each image of sequence. Then MHI (Motion History Images) and MRI (Motion Recency Images) are generated. MRI basically is the union of all changed detected images, which will represent all the pixels which have changed in a whole sequence. MHI is a scalar-valued image where more recent moving pixels are brighter. In their system, MHI and MRI templates are used to recognize motion actions (18 aerobic exercises). Several moments of a region in these templates are employed in the recognition process. The templates for each exercise are generated using multiple views of a person performing the exercises. However, it is shown that during recognition only one or two views are sufficient to get reasonable results.

Next, in chapter eight, Goddard argues that the significant advances in vision algorithms come from studying extant biological systems. He presents a structured connectionist approach for recognizing activities. A scenario is used to represent movement, which is not based on 3-D, it is purely 2-D. The scenario represents a movement as a sequence of motion *events*, linked by the intervals. Input to the system is a set of trajectories of the joints of an actor performing an action. A hierarchy of models starting with the segment level is used, which include thigh, upper arm, fore arm; these segments are combined to get components like legs, arms. The components are combined into assemblies, and assemblies into objects. The system is triggered by the motion's events, which are defined as change in angular velocity of a segment, or a change in the orientation of a segment.

Finally, in the last chapter of this part of the book, chapter eight, Rohr presents a model-based approach for analyzing human movements. He uses cylinders to model the human body, and joint curves to model the motion. The joint curves were generated from the data of sixty normal people of different ages. This method is comprised of two phases. The first phase, called the initialization phase, provides an estimate for the posture and three-dimensional position of the body using a linear regression method; the second phase, starting with the estimate from the first phase, uses a Kalman filter approach to incrementally estimate the model parameters.

3. Gesture Recognition and Facial Expression Recognition

In our daily life, we often use gestures and facial expressions to communicate with each other. Gesture recognition is a very active area of research [16]. Some earlier work includes the work of Baudel *et al* [29] who used a mechanical glove to control the computer presentation; Fukomoto *et al* [14] also designed a method to guide a computer presentation, but without using any glove. Cipolla *et al* [7] used a rigid motion of a triangular region on a glove to control the rotation and scaling of an image of a model. Darrell and Pentland [9] used model views, which are automatically learned

from a sequence of images representing all possible hand positions using correlation. Gesture models are then created, for each view, correlation is performed with each image of sequence, and the correlation score is plotted. Matching is done by comparing correlation scores. They were able to recognize hello and good bye gestures, they needed time warping, and special hardware for correlation.

There are two important issues in gesture recognition. One, *what* information about the hands is used, and *how* it is extracted from images? Two, how the variable length sequences are dealt with? Some approaches use gloves or markers on hands, consequently the extraction of information from images is very easy. In other approaches, the point based features (e.g., fingertips) are extracted, which carry the motion information, but do not convey any shape information. In some other approaches, however, a blob or region corresponding to a hand is identified in each image, and some shape properties of the region are extracted, and used in recognition. In addition, some approaches also use global features using the whole image (e.g., eigen vectors).

Most approaches to gesture recognition are 2-D. In these approaches, only 2-D image motion and 2-D region properties are used. Some 3-D gesture recognition approaches have also been proposed in which the 3-D motion and the 3-D shape of the whole hand, or fingers are computed and used in recognizing gestures. For example, Regh and Kanade [26] describe a model-based hand tracking system called *DigitEyes*. This system uses stereo cameras and special real-time image processing hardware to recover the state of a hand model with 27 spatial degrees of freedom. Kang and Ikeuchi [19] describe a framework for determining 3-D hand grasps. An intensity image is used for the identification and localization of the fingers using curvature analysis, and a range image is used for 3-D cylindrical fitting of the fingers. Davis and Shah [10] first identify the fingers of the hand and fit a 3-D generalized cylinder to the third phalangeal segment of each finger. Then six 3-D motion parameters (translation and rotation) are calculated for each model corresponding to the 2-D movement of the fingers in the image plane.

A gesture can be considered as a trajectory in a feature space. For example, a motion trajectory is a sequence of locations (x_i, y_i), for $i = 1 \ldots n$, where n is the number of frames in a sequence. A motion trajectory can thus be considered as a vector valued function, that is, at each time we have two values (x, y). However, a single valued function is better suited for computations, and therefore parameterization of trajectories is necessary. A trajectory can be parameterized in several ways; for instance $\phi - S$ curve, speed and direction, velocities v_x and v_y, and spatiotemporal curvature. The first parameterization completely ignores time; two very different

trajectories might have the same $\phi - S$ curves. The remaining parameterization are time dependent. Trajectories representing the same gesture or action may be of different length. The trajectories can be temporally aligned by non-linear time-warping. The difficulty with time warping methodology is that it is computationally extensive.

An alternate approach to time warping is to model a gesture as a Finite State Machine (FSM). Davis and Shah [11] identify four main phases in a generic gesture, and use a FSM to model these phases. The user is constrained to the following four *phases* for making a gesture. (1) Keep hand still (fixed) in start position until motion to gesture begins. (2) Move fingers smoothly as hand moves to gesture position. (3) Keep hand in gesture position for desired duration of gesture command. (4) Move fingers smoothly as hand moves back to start position.

Hidden Markov Models (HMM) have been known in the literature for a long time [24]. HMMs can be employed to build a stochastic model of a time-varying observation sequence by removing the time dependency. A HMM consists of a set of states, a set of output symbols, state transition probabilities, output symbol probabilities, and initial state probabilities [6]. The model works as follows. Sequences are used to train HMMs. Matching of an unknown sequence with a model is done through the calculation of the probability that an HMM could generate the particular unknown sequence. The HMM giving the highest probability is the one that most likely generated that sequence [17].

The FSM essentially is a simplified version of HMM with state transition probabilities equal to zero or one. The important difference is that FSM was generated by the user using the conceptual four phases of a generic gesture. However, a large number of training sequences are used to automatically generate HMMs.

In chapter nine, Bobick and Wilson use a time collapsing technique to achieve time invariance. They start with trajectories in a time-augmented configuration space and compute the principal curve using least squares fit of near by points. Trajectories and principal curve are slightly compressed in time, and a new principal curve is computed. The process is repeated until time is reduced to zero. Next, the sample points of the prototype curve are clustered. Each cluster is assigned a state. Bobick and Wilson define a gesture as a sequence of states. A dynamic programming algorithm is used for recognizing gestures.

Their approach is very similar to the HMM approach. One important difference is their use of the time-collapsing technique for converting trajectories. HMMs need a large set of training samples. However, Bobick and Wilson claim rapid training with very few samples.

Starner and Pentland, in chapter ten, present a method for recognizing

American Sign Language consisting of a 40 word vocabulary involving 500 sentences. They use tracking based on color (in one case the user had to wear colored gloves, in other case the color of skin is used). A blob corresponding to each hand is identified, and eight features (centroid, angle of orientation and eccentricity of bounding ellipse) are used in the HMM based approach.

The problem of recognizing facial expressions from video sequences is a challenging one. Since Ekman and Frisen's work [13] on *Facial Action Coding System* or FACS, there has been a lot of interest in facial expression recognition in Psychology and Computer Vision. For facial expression recognition, there are also two classes of approaches: 2-D and 3-D.

In chapter eleven, Black *et al* present a method for recognizing facial expression using 2-D motion. In their method, the rectangular windows corresponding to different parts of the face (eyes, brows, mouth) are identified, and optical flow is computed. The relative motion of parts of the face is used to recognize the expressions. Therefore, absolute motion of the face is first estimated to stabilize the motion of parts of the face in a warped sequence. The authors employ a parameterized model of optical flow using eight parameters for each patch. The eight parameters have the qualitative interpretation of the image motion in terms of translation, curl, deformation, divergence, and curvature. The approach then is extended to compute articulated motion. In articulated motion, each patch is connected to only one preceding patch and one following patch, for example a thigh patch may be connected to a preceding torso patch and following calf patch.

Essa and Pentland, in chapter twelve, present a 3-D approach for facial expression recognition. They employ a 3-D dynamic muscle based model of a face. Simoncelli's multi-scale, coarse-to-fine Kalman filter based method is used to compute optical flow. Using this optical flow, the velocities of each node of the face model are computed. Next, using a physically-based modeling technique, the forces that caused the motion are computed. Finally, a control theoretic approach is employed to obtain the muscle actuation.

The authors present two method for facial expressions. In the first one, they use peak actuation of each of 34 muscles between the application and release phases of the expression as the feature vectors. In another approach, they use spatio-temporal motion energy templates. In both methods they get a 98% recognition rate for five expressions: smile, surprise, rise eyebrows, anger, and disgust.

4. Lipreading

Automatic Speech Recognition (ASR) has been a hot research topic for a long time. Currently, there are a variety of ASR systems which are speaker independent, which recognize continuous speech, and which perform quite

well. However, the recognition rate is not 100% yet. On the other hand, speech recognition by humans is extremely accurate. For, in addition to speech, we use lip movement, facial expression and sometimes gestures to supplement our speech. Starting from the original work of Petajan [23] on visual lipreading, there has been growing interest in visual lipreading (see [28], which provides the state of art in lipreading.). To some extent, it has been demonstrated that combined audio and video speech recognition outperforms any single modality (audio or video), particularly in noisy environments like offices, bars, parties, etc.

The problem of visual lipreading have proved to be more complex than speech recognition using sound. There are several reasons for this. First, the visual image contains more data than a sound sample. An image is two dimensional, and typically a 100×100 image captures the mouth region, which is $10,000$ bytes of data compared to one byte of data per sample for the speech signal. With a reasonable frame rate, at least 15 to 20 frames are needed to capture a single utterance, making this data even larger. Second, images are very sensitive to the motion of the speaker and his or her head movement, as compared to sound signal. Third, images are sensitive to lighting conditions, and there is not much contrast between the lips and the rest of the face.

In lipreading, there are two important issues: feature extraction and feature matching for recognition. The most simple approach is to use gray levels as a feature vector. As it has been shown in the context of face recognition using static images [31], the gray levels, surprisingly, perform quite well for the lipreading problem (see chapter fifteen). Another possible approach is to use the lip contour as a feature vector. The difficulty in this connection is the reliable extraction of a contour from a mouth image. Finally, some region properties of the mouth region can also be used as a feature vector.

Recognition and training strategies have smoothly migrated from speech recognition to the lipreading domain. Two popular methodologies are: Hidden Markov Models (HMM) and Neural Networks (NN). Both HMM and NN need a large training set, which in some cases is quite difficult to obtain. The advantage of HMM is that it is able to deal with variable length sequences. It is common to have sequences of the same utterance which are not of the same length.

In chapter thirteen, Bregler and Omohundro present a method for tracking lip contour, for the lipreading problem. One obvious way to do this is to use snakes. In the original snakes paper [20], lip tracking was demonstrated. However, the problem was that one of the authors had to put on lipstick in order to create an artificial contrast between lips and rest of the face to be useful in snakes. Otherwise, snakes get caught into local minima.

In Bregler and Omohundro's method, the training lip images are initially labeled by the snakes algorithm; sometimes snakes select the boundary of an incorrect neighboring object, like the nose instead of the mouth, which are removed by hand. Next, using these lip contours a nonlinear manifold of all possible lip configurations is learned. During lip tracking, the contour from the learned manifold is found which maximizes the image gradient along the contour in the image. They report experiments with a four word vocabulary using only visual information with a 95% recognition rate. They also report the results on the database of six speakers on sequences of 3-8 letter names using audio only, and combined audio and video. The best recognition rate of about 55% is obtained with combined audio and video input in the presence of crosstalk.

Goldschen *et al*, in chapter fourteen, present an ambitious project for recognizing continuous speech using lipreading. Even though, their system gets only 25% recognition rate, it highlights several important issues in lipreading. They use fifteen features from the oral-cavity shadow images of the speaker. These images were taken by specialized apparatus, consequently detection of the oral-cavity images was not difficult. Their system uses HMMs for recognition, as does Bregler and Omohumdro's. The HMM's were trained to recognize a set of sentences using visemes, trisemes (triplets of visemes), and generalized trisemes (clustered trisemes).

Finally, in the last chapter of this book, chapter fifteen, Nan *et al* present a non-HMM method for lipreading. In their method, the feature vector consists of gray levels of all pixels in all images in a sequence. Several such vectors corresponding to the training sequences are used to compute eigenvectors (eigensequence), for each spoken letter. The recognition of an unknown sequence representing a spoken letter is performed by computing the ratio of energy of projection of the sequence on the model eigenspace and the energy of the sequence. The region around mouth area in each image is extracted by using a simple correlation method. A mouth template from the previous image is run through the current image, and correlation computed. The window around the location with the highest correlation is taken as the mouth region. They employ time warping to get equal length sequences. The recognition rate (85% for ten letter vocabulary) is good, even though their method is computationally expensive due to warping.

5. Conclusion

The papers presented in this book are representative of the directions being explored in dynamic vision that are very relevant to many applications in video analysis, video databases, and different advanced human-computer interfaces. Progress in these areas is essential to design computing environ-

ments needed in the next generation systems. On the other hand, techniques developed in these areas are in turn forcing computer vision researchers to look at computer vision as a dynamic system in which information exists at different abstraction levels. Clearly, we are at the beginning of an exciting research field. Many difficult problems have not even been articulated. We hope that this book will help researchers find relevant material at one place and encourage new researchers to explore some of the exciting and challenging directions presented in this book.

References

1. Adelson, E.H. and Niyogi, S. A. Analyzing and recognizing walking figures in XYT. In *IEEE CVPR-94*, pages 469–474, 1994.
2. A. Katkere, S. Moezzi, D. Kuramura, P. Kelly, and R. Jain. Towards video-based immersive environments. *Multimedia Systems Journal*, Spring 1997.
3. M. C. Allmen. *Image Sequence Description Using Spatiotemporal Flow Curves: Toward Motion-Based Recognition.* PhD thesis, University of Wisconsin–Madison, 1991.
4. M. C. Allmen and C. R. Dyer. Cyclic Motion Detection Using Spatiotemporal Surfaces and Curves. In *Proc. 10th Int. Conf. Pattern Recognition*, pages 365–370, 1990.
5. Bobick, A. F. and Campbell, L.W. Recognition of human body motion using phase space constraints. In *IEEE ICCV-95*, pages 624–630, 1995.
6. Cédras, C., and Shah, M. Motion-based recognition: A survey. *Image and Vision Computing*, 13(2):129–155, March 1995.
7. Cipolla, R.,Okamoto, Y. and Kuno, Y. Robust structure from motion using motion parallax. In *IEEE Proceedings on International Conference on Computer Vision*, 1993.
8. Cootes, T. J., taylor, C.J., and Graham, J. Active shape models-their training and application. *Computer Vision, and Image Understanding*, 61:38–59, 1995.
9. Darrell, T., and Pentland, A. Space-time gestures. In *CVPR*, pages 335–340. IEEE, 1993.
10. Davis, J., and Shah, M. Three-dimensional gesture recognition. In *Asilomar Conference on Signals, Systems, And Computers*, 1994.
11. Davis, J., and Shah, M. Visual gesture recognition. *IEE Proceedings Vision, Image and Signal Processing*, 141(2):101–106, 1994.
12. Davis, L. S. and Gavrila, D. M. 3-d model-based tracking of human upper body movement: A multi-view approach. In *IEEE CVPR-96*, pages 73–80, 1996.
13. Ekman, P. and Friesen, W. A.. *Facial Action Coding System.* Consulting Psychologist Press, 1978.
14. Fukumoto, M., Mase, K., and Suenaga, Y. Real-time detection of pointing actions for a glove-free interface. In *IAPR Workshop on Machine Vision Applications*, pages 473–476, December 1992.
15. D. C. Hogg. *Interpreting Images of a Known Moving Object.* PhD thesis, University of Sussex, 1984.
16. Huang, T., Pavlovic, V. Hand gesture modeling, analysis, and synthesis. In *Proc. International Workshop on Automatic Face and Gesture Recognition*, pages 73–79, 1995.
17. J. Schlenzig, E. Hunter and R. Jain. Recursive identification of gesture inputs using hidden markov models. In *Proc. IEEE Workshop on Applications of Computer Vision*, pages 187–194, 1994.
18. Jain, R.C., Militzer, D., and H.-H. Separating non-stationary from stationary scene

components in a sequence of real world tv-images. In *IJCAI-77*, pages 612–618, 1977.
19. Kang, S.B., and Ikeuchi, K. Toward automatic robot instruction from perception – recognizing a grasp from observation. *IEEE Transactions of Robotics and Automation*, 9:432–443, August 1993.
20. Kass, M., Witkin, A., and Terzopoulos, D. Snakes: Active contour models. In *Proceedings of First International Conference on Computer Vision*, pages 259–269, London, 1987.
21. Little, J. and Boyd, J. Describing motion for recognition. *Int. Symposium on Computer Vision-95*, pages 235–240, November 1995.
22. Metaxas, D. and Kakadiaris, I.A. Model-based estimation of 3d human motion with occlusion based on active multi-viewpoint selection. In *IEEE CVPR-96*, pages 81–87, 1996.
23. Petajan, E. *Automatic Liprading to Enhance Speech Recognition*. PhD thesis, University of Illinois, 1984.
24. L.R. Rabiner and B.H. Juang. An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, pages 4–16, January 1986.
25. Rangarajan, K., Allen, Bill, and Shah, M. . Matching motion trajectories. *Pattern Recognition*, 26:595–610, July, 1993.
26. Rehg, J., and Kanade, T. Visual tracking of high dof articulated structures: an application to human hand tracking. In *ECCV*, pages 35–46, May 1994.
27. R. Jain and K. Wakimoto. Multiple perspective interactive video. In *Proceedings of the International Conference on Multimedia Computing and Systems*, pages 202–211. Computer Society Press, May 15-18 1995.
28. Stork, D. and Hennecke, M. *Speechreading by humans and machines*. Springer, 1996.
29. Baudel T. and Beaudouin-Lafon M. Charade: Remote control of objects using free-hand gestures. *CACM*, pages 28–35, July 1993.
30. Tsai, Ping-Sing, Keiter, K., Kasparis, T., and Shah, M. Cyclic motion detection. *Pattern Recognition*, 27(12), 1994.
31. Turk, M., and Pentland, A. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, pages 71–86, 1991.
32. Williams, D. and Shah, M. Greedy algorithm for active contour and curvature estimation. *Computer Vision, Graphics, and Image Processing*, pages 14–26, January, 1992.