Final Report 2000-28

MINNESOTA DEPARTMENT OF TRANSPORTATION

Research

Pedestrian Control At Intersections: Phase IV

Minnesota Guidestar

Technical Report Documentation Page

1. Report No. MN/RC - 2000-28	2.	3. Recipients Accession No.
4. Title and Subtitle PEDESTRIAN CONTROL AT INTERSECTIONS: PHASE IV		5. Report Date February 2000 6.
7. Author(s) Osama Masoud Nikos Papanikolopoulos		8. Performing Organization Report No.
9. Performing Organization Name and Address University of Minnesota Dept. of Computer Science and Engineering Artificial Intelligence, Robotics and Vision Laboratory Minneapolis, MN 55455		10. Project/Task/Work Unit No. 11. Contract (C) or Grant (G) No. c) 74708 wo) 79
12. Sponsoring Organization Name and Address Minnesota Department of Transportation 395 John Ireland Boulevard Mail Stop 330 St. Paul, Minnesota 55155		13. Type of Report and Period Covered Final Report 1999 - 2000 14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract (Limit: 200 words)

This report presents a real-time system for pedestrian tracking in sequences of grayscale images acquired by a stationary camera.

Researchers also developed techniques for recognizing pedestrian's actions, such as running and walking, and integrated the system with a pedestrian control scheme at intersections. The proposed approach can be used to detect and track humans in a variety of applications. Furthermore, the proposed schemes also can be employed for the detection of several diverse traffic objects of interest, such as vehicles or bicycles.

The system outputs the spatio-temporal coordinates of each pedestrian during the period the pedestrian is in the scene. The system processes at three levels: raw images, blobs and pedestrians. Experimental results based on indoor and outdoor scenes demonstrated the system's robustness under many difficult situations, such as partial or full occlusions of pedestrians. In particular, this report contains the results from a field test of the system conducted in November 1999.

17. Document Analysis/Descriptors pedestrian detection and tracking action recognition pedestrian control at intersections		18. Availability Statement No restrictions. Document available from: National Technical Information Services, Springfield, Virginia 22161	
19. Security Class (this report) Unclassified	20. Security Class (this page) Unclassified	21. No. of Pages 95	22. Price

Pedestrian Control at Intersections: Phase IV
Final Report

Prepared by:
Osama Masoud
Nikolaos P. Papanikolopoulos

Artificial Intelligence, Robotics and Vision Laboratory
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455

Published by:
Minnesota Department of Transportation
Office of Research Services
First Floor
395 John Ireland Boulevard, MS 330
St Paul, MN 55155

Executive Summary

This report presents a real-time system for pedestrian tracking in sequences of grayscale images acquired by a stationary camera. We have also developed techniques for recognizing pedestrian's actions (e.g., running, walking, etc.). We have integrated the system with a pedestrian control scheme at intersections. The proposed approach can be used to detect and track humans in a variety of applications. Furthermore, the proposed schemes can also be employed for the detection of several diverse traffic objects of interest (vehicles, bicycles, etc.) The system outputs the spatio-temporal coordinates of each pedestrian during the period the pedestrian is in the scene. Processing is done at three levels: raw images, blobs, and pedestrians. Blob tracking is modeled as a graph optimization problem. Pedestrians are modeled as rectangular patches with a certain dynamic behavior. Kalman filtering is used to estimate pedestrian parameters. The system was implemented on a PC equipped with a Matrox Genesis board and was able to achieve a peak performance of over 30 frames per second. Experimental results based on indoor and outdoor scenes demonstrated the system's robustness under many difficult situations such as partial or full occlusions of pedestrians. In particular, this report contains the results from a field test of the system conducted in November 1999.

Table of Contents

List of Figures viii

List of Tables x

Chapter 1: Introduction 1
1.1 Articulated Motion 2
1.2 Application Domain 2
1.3 Tracking 4
1.4 Motion Recognition 6
1.5 Problem Definition 8

Chapter 2: Related Work 11
2.1 Whole-Body Tracking 12
2.2 Limb Tracking 14
2.2.1 2-D Limb Tracking 14
2.2.2 3-D Limb Tracking 15
2.3 Motion Recognition 18
2.3.1 Recognition Based on 2-D Tracking 18
2.3.2 Recognition Based on 3-D Tracking 19
2.3.3 Recognition Based on Motion Features 20
2.4 Hand Tracking, Hand Gesture Recognition and Lip-reading 22

Chapter 3: Motion Recognition 25
3.1 Feature Image 27
3.2 Learning and Recognition 30
3.2.1 Magnitude and Size Normalization 31
3.2.2 Principle Component Analysis 31

3.2.2.1 Overview 31
3.2.2.2 PCA for Action Recognition 33
3.2.3 Recognition 36
3.2.3.1 Distance Measure 36
3.2.3.2 Classification 37
3.3 Action Data 38
3.3.1 Data Selection 38
3.3.2 Acquisition 41
3.4 Experimental Results 42
3.4.1 Classification Experiment 42
3.4.2 Parameter Selection 48
3.4.2.1 Resolution 48
3.4.2.2 Number of Training Images 50
3.4.3 Complexity 51
3.5 Summary 52

Chapter 4: Model-based Filtering 55
4.1 Introduction 55
4.2 3-D Model Filtering 55
4.3 Experimental Results 58
4.4 Conclusion 58

Chapter 5: Active Deformable Models 63
5.1 Introduction 63
5.2 Active Deformable Models 63
5.3 Experimental Results 66
5.4 Conclusion 67

Chapter 6: Field Demonstration 69
6.1 Introduction 69

6.2 Demonstration 69
6.3 Conclusion 74

Chapter 7: Conclusions 75
7.1 Summary 75
7.2 Future Work. 76
7.2.1 Tracking 76
7.2.2 Real-time Implementation of Action Recognition 77
7.2.3 Recognition from a General View Angle. 78

Bibliography. 79

List of Figures

Figure 1.1 A moving light display with and without the human body outline. 7

Figure 2.1 General interpretation cycle.. 16

Figure 3.1 Snapshots from a motion sequence (of a person skipping) where the images have been blurred. Humans have no difficulty perceiving the action.. 26

Figure 3.2 Filter response to a step signal. 27

Figure 3.3 An example of a walking and a running motion sequence. (a) Original images. (b) Filtered images (feature images) with . . . 29

Figure 3.4 Feature image size selection. 30

Figure 3.5 Several frames from Walk, Run, Skip, and March actions.. . . . 39

Figure 3.6 Several frames from Line-Walk, Hop, Side-walk, Side-skip actions. 40

Figure 3.8 The first ten eigenvectors. 43

Figure 3.7 Eigenvectors contribution to variation in data. (a) Individual contribution. (b) Cumulative contribution.. 44

Figure 3.9 Recognition performance. 45

Figure 3.10 Misclassified actions for each subject. The numbers indicate the actions that were chosen incorrectly (1=Walk, 8=Side-skip). . . 47

Figure 3.11 Confusion plot. The area of the squares indicates the distance using the distance measure in Section 3.2.3.1. The distances are averaged over all test samples. 48

Figure 3.12 Original frame and its feature images at different resolutions. 49

Figure 3.13 Effect of resolution on classification performance.. 50

Figure 3.14 Classification performance for different values. 51

Figure 4.1 Tracking sequence of a pedestrian in different occlusion situations. 60

Figure 4.2 Tracking sequence demonstrating occlusions and pedestrian overlap. 61

Figure 4.3 Computed RMS errors for two pedestrians. 62

Figure 5.1 Snapshots showing snake tracking pedestrians. 66

Figure 6.1 Camera view of the intersection.. 71

Figure 6.2 Computer hardware used to process the incoming video. 71

Figure 6.3 The traffic light is flashing red when there are pedestrians at the intersection. 72

Figure 6.4 Another view of the intersection. 72

Figure 6.5 The boxes follow the pedestrians as they move through the intersection. 73

Figure 6.6 Another instance of pedestrian tracking. The box around the pedestrian illustrates relevant pedestrian information (position, velocity, etc.). 73

List of Tables

Table 3.1: Variation in the duration of one cycle for the data set. 38

Table 3.2: Confusion matrix. 46

Chapter 1

Introduction

A great deal of research in computer vision has dealt with the analysis of temporal image data. Video provides very rich information when compared with information obtained through other sensors. It is no surprise that humans obtain more than 70% of their sensory information through their eyes. Applications that require interaction with dynamic environments, such as robotic visual servoing, pedestrian control at intersections, obstacle detection, collision avoidance and surveillance, can benefit the most from video as a source of sensory information. Video has become even more suitable in the recent years due to the availability of high quality and continually cheaper acquisition devices (CCD, CMOS, and digital cameras). In addition, the dramatic increase in computer processing power, the increase in storage devices capacity and decrease in their cost, and the increase in bandwidth for transferring media have all contributed to making video processing a more realistic solution than it used to be.

Another reason in favor of using temporal image data is that psychophysical evidence suggests that biological visual systems are well adapted to process temporal information. The human visual system is very sensitive to motion. This is also true for many animals. Stationary objects are much harder to detect than moving objects. Many animals use this fact as a camouflage strategy.

In this project, we deal with two related, yet different, aspects of motion apparent in pedestrian control problems: tracking and recognition. Tracking attempts to identify certain spatial and dynamic quantities of motion through time. Motion recognition is a higher level process which attempts to identify the type of motion taking place. More details will be given in the problem definition section.

1.1 Articulated Motion

In computer vision, motion can be classified into rigid and non-rigid motion. Articulated motion is considered an important subset of non-rigid motion where the object of interest is composed of several rigid components connected to each other by ball and hinge joints. The human body, many animals and insects, and machinery all exhibit articulated motion. Because of the wealth of applications that involve the human body, most of the research that dealt with articulated motion was aimed at human body motion. For the sake of usefulness, the work introduced in this project also deals with human body motion although it can be generalized to any articulated motion. Because of the high articulation and flexibility of the human body, its motion can be very complex. Analysis of this motion is made even more difficult because of the effect of clothing on the appearance of the body. Another source of difficulty is that occlusions of some parts and reappearance of other parts occur very frequently as the body performs even a simple action like walking. These difficulties make the problem of dealing with human body motion a challenging one.

1.2 Application Domain

Vision-based tracking and analysis of articulated motion has the advantage of being non-intrusive. This is especially important when dealing with humans. In some applications, the only alternative would be the placement of markers on many parts of the body to track motion. Other applications may require the use of active sensors such as infrared and ultra-sonic sensors. The convenience and safety of visual sensing comes at the cost of difficulty in achieving the desired goal. Many vision-based solutions do not provide a satisfactory solution yet or in some cases, are too sensitive to small changes in the environment. However, this field is relatively recent and many research tracks have yet to be explored.

There are many applications for visual tracking and analysis of articulated motion. In *surveillance*, a human operator has been traditionally used. Automating surveillance can be highly desirable in cases where using a human operator is not feasible. Automated surveillance can be used to detect intruders to a restricted area or find suspicious activities. Parking lots, department stores and ATM machines are a few examples where such a system can be used. Simple motion detectors suffer from the problem of giving too many false positives. A human, a dog, or a moving tree due to the wind blowing will all trigger an alarm. The system needs to be intelligent enough to distinguish between a human and other moving objects. Furthermore, the system should be able to distinguish a suspicious activity from a regular one. Surveillance can also use trajectory information of pedestrians to detect and classify events and behaviors [38,44]. Automated surveillance can also be used in factories to monitor machinery as an alternative to a more expensive manual system. Breakdowns can be detected early when one of the machines starts moving in a different way than it is supposed to or stops moving completely. Early detection is very critical in situations like this.

Pedestrian *traffic* monitoring is another demanding application. In traffic control, tracking pedestrians at intersection can be used to both increase safety and optimize traffic timing. Safety can be increased by either providing extra crossing time for people who need extra time or by providing a warning signal to drivers indicating the presence of pedestrians in the crosswalk. Optimization is a direct outcome of dynamically adjusting the walk signal delay. When there are no pedestrians, the walk signal should not be part of the cycle. Pedestrian traffic monitoring can be used to count pedestrians and provide other quantities such as the different paths pedestrians take and most visited locations. This type of application is particularly useful for retailers and shopping center who can use the data to improve operating efficiency, evaluate performance, and charge hourly for retail spaces.

In each of the previous applications, a vision-based solution will require either tracking the subject (or parts of the subject) of interest or recognition of an action. These two requirements are described in more detail below along with the motivation behind the choices of the problems we addressed in this project and the solution methods we used.

1.3 Tracking

Tracking implies the recovery of trajectories. In tracking humans, there are different levels at which tracking can be performed. At the highest level, the whole body is tracked without paying attention to the details of the posture and limbs. At a lower level, the posture and limbs are tracked. At an even lower level, one or two parts of the body (such as hands) are tracked. The finest level would be tracking the fingers of a hand or facial features. The type of application determines the tracking level.

Part of the scope of this work deals with the highest level of tracking: tracking the whole body as one unit. There are several reasons for our choice of this problem. In addition to it being an important problem in some applications, it is a problem whose solution needs to be as general as possible. Applications that require whole body tracking, whether it is surveillance or traffic-related, involve mostly outdoor settings where very little environment control is possible. Changing weather and lighting conditions are some examples of inevitable factors. A satisfactory solution to this problem needs to handle these changes. In addition, controlling the subjects as in having them wear a certain color or behave in a certain way is not applicable in this case. Controlled environments may be applicable in other applications that require different tracking levels to reduce the complexity of the problem. For example, in sign language interpretation, the subject can be required to have a certain pose and wear colored gloves. The lighting conditions can also be controlled. Many solutions have been presented for tracking the whole body as a single unit but the majority imposed impractical restrictions. We chose this problem because of the lack of a general solution, in an attempt to present a solution with as few restrictions as possible.

Another motivation for this choice is that solving this problem can provide a useful tool to deal with the second problem we address in this work, namely, action recognition. The usefulness stems from the fact that locating the subject of interest will help in confining the recognition effort to that location.

In tracking humans as single units, the drawback in many of the solutions that have been presented originates from an implied assumption that there is a clean one-to-one correspondence between extracted features, or blobs, and pedestrians in the scene. Arbitrary images may not satisfy this assumption due to clutter, color of clothes, and weather conditions among other factors.

In our solution, we allow maximum flexibility by allowing this relation to be many-to-many. Our method tracks features and pedestrians at different levels and dynamically updates this relation.

1.4 Motion Recognition

Given a number of predefined actions, recognition of an articulated object motion is the problem of classifying the object motion into one of these actions. Normally, the set of actions has a meaning in a certain domain. In sign language for example, the set of actions corresponds to the set of possible words and letters that can be produced. In ballet, the actions are the step names in one of the ballet notation languages. In pedestrian applications, motion recognition can help us separate pedestrians from rollerbladers.

The study of human body motion perception by the human visual system was made possible by the use of the so-called moving light displays (MLDs) first introduced by Johansson in 1973 [42,43]. Johansson devised a method to isolate the motion cue by constructing an image sequence where the only visible features are a set of moving lights corresponding to joints of the human body. Figure 1.1 shows an example.

He found that when a subject was presented an MLD corresponding to an actor performing an activity such as walking, running, or stair climbing, the subject had no problem recognizing the activity. It took less than 200 milliseconds to identify the activity. The subjects were not able to identify humans when the lights were stationary. Cutting and Kozlowski [17,51] demonstrated that the gender of the walking person and the gait of a friend can be identified from MLDs. It was later shown that subjects can identify more complex movements such as hammering, box lifting,

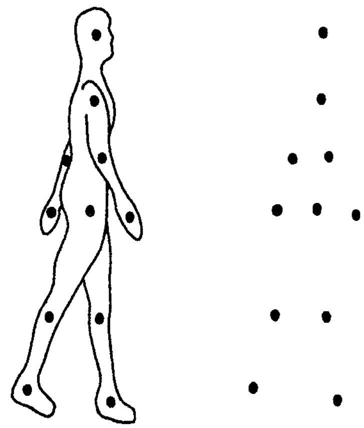


Figure 1.1 A moving light display with and without the human body outline.

ball bouncing, dancing, greeting, and boxing [21]. Two theories on how people recognize actions from MLDs have been suggested. In the first theory, people perform three-dimensional reconstruction of the object and then use that to recognize the action. In the second theory, people utilize motion information directly without performing reconstruction.

Consequently, in computer vision, the problem of motion recognition has been attempted using two approaches. The first is a reconstruction-based approach that attempts to do structure-from-motion first and then use the recovered three-dimensional information in recognition. The second is a motion-based approach that uses motion parameters directly without performing any reconstruction. The shape-from-motion problem in its general sense is a classic computer vision problem. It involves the recovery of three-dimensional motion parameters and the relative depth map. When dealing with an articulated object, this process becomes more complex due to the non-rigidity of the object. To date, the problem of articulated object shape-from-motion, which is the same as the problem of limb tracking and posture recovery described in the previous section, has not been satisfactorily solved. The vast majority of the work related to articulated motion, however, falls under the reconstruction-based category, probably more motivated by the applications that make use of the reconstruction phase alone than by psychophysics. In this work, we argue that reconstruction is not a necessary step that must precede motion recognition. We present a novel motion-based approach for motion recognition which unlike other motion-based approaches, can be generalized to recognize any articulated motion.

1.5 Problem Definition

Tracking and analysis of articulated objects has many practical applications, most of which involve the human body. The desired tracking and analysis outcome depends on the specific applications. Two of these outcomes are dealt with in this project: tracking the body as a whole and recognizing actions. Therefore, we define our problem as the problem of tracking and action recognition of the human body. The problem consists of the following two sub-problems:

1. Tracking the human body as a whole: It is the problem of real-time recovery of motion trajectories in world coordinates of moving pedestrians in a scene bounded by the view field of a single stationary camera. To alleviate the recovery of world coordinates, camera parameters such as resolution, focal length, height and tilt angle are assumed to be known. No restrictions are made on the camera position, the number of pedestrians, pose, motion direction, speed, or occlusions. There is also no restriction regarding weather conditions or the existence of other possibly moving objects such as oscillating trees.

2. Action recognition: It is the problem of classifying the action performed by a human in a video sequence. No other sensory input such as three-dimensional joint locations is allowed. The domain of possible actions is provided along with samples of each action. The technique is required to be capable of generalization to any domain with any set of actions. The actions

performed may have variable durations. The same action is also allowed to have different speeds. Moreover, temporal alignment of actions is not required. Recognition should not be influenced by the actor, his/her height, shape or style in performing the actions.

Chapter 2

Related Work

Articulated motion (the motion that pedestrians exhibit) has been a subject of interest in many different fields. In robotics, researchers utilized control techniques in an attempt to build legged robots which can perform a variety of tasks such as walking and running. Biomechanics dealt with providing data that describes as close as possible human motion. In psychophysics, the perception of articulated motion by the human visual system has been studied. In computer graphics, the emphasis was on motion synthesis, or animation. Finally, in computer vision, the focus was on computational techniques for tracking and perception articulated motion.

This chapter provides a review of the body of work that has been done in computer vision in relation to articulated motion. We classify the work into three categories, which reflect the organization of this chapter:

1. Whole-body tracking: This deals with tracking the articulated body, a pedestrian, as a single unit, without the identification or tracking of the parts of the body.
2. Limb tracking: Here, the emphasis is on the individual body parts and the recovery of the structure of the body. Limb tracking techniques can be either two-dimensional or three-dimensional, depending on whether the recovered parameters are in image coordinates or in world

coordinates.

3. Motion recognition: Methods belonging to this category attempt to identify the action as one of the actions in a database. Motion recognition can rely on two- or three-dimensional tracking information. Alternatively, it can utilize motion features directly.

4. Hand Tracking, Hand Gesture Recognition and Lip-reading.

The work described in this report belongs to whole-body tracking (Section 2.1) and motion recognition using motion features (Section 2.3.3).

2.1 Whole-Body Tracking

Several attempts have been made to track pedestrians as single units. Baumberg and Hogg [5] used deformable templates to track the silhouette of a walking pedestrian. The advantage of their system is that it is able to identify the pose of the pedestrian. Tracking results were shown for one pedestrian in the scene and the system assumed that overlap and occlusions are minimal [4]. Another use of the silhouette was made by Segen and Pingali [72]. In their case, features on the pedestrian silhouette were tracked and their paths were clustered. The system ran in real-time but was not able to deal well with temporary occlusions. Occlusions and overlaps seem to be a primary source of instability for many systems. Rossi and Bozzoli [70] avoided the problem by mounting the camera vertically in their system which aimed to mainly count passing pedestrians in a corridor. Such a camera configuration, however, may not be feasible in some cases. Occlusions and overlaps occur very commonly in pedestrian scenes; and cannot be ignored by a pedestrian tracking system. The use of multiple cameras can alleviate the occlusion problem. Cai and Aggarwal [9] tracked pedestrians with multiple cameras. The system, however, did not address the occlusion problem in particular but rather how to match the pedestrian across different camera views. The switching among cameras was done manually.

Smith *et al.* [76] performed pedestrian detection in real-time. The system used several simplistic criteria to judge whether the detected object is a pedestrian or not but did not actually track pedestrians. Sullivan *et al.* [79] used deformable models to track pedestrians. Initialization and tracking was performed using segmented pedestrians from the background. In general, handling occlusions is difficult using deformable models. Shio and Sklansky [74] presented a method for segmenting people in motion with the use of correlation techniques over successive frames to recover the motion field. Iterative merging was then used to recover regions with similar motion. The method was able to deal with partial occlusions. The assumption was that pedestrians do not change direction as they move. A disadvantage of this method is the high computational cost of the correlation and the iterative merging steps. An interesting approach which was presented by Heisele *et al.* [34] is based on their earlier work on color cluster flow [33]. An image is clustered into regions of similar color. In the subsequent images, the clusters are updated using a k-means clustering algorithm. Assuming that the pedestrian legs form one cluster, a step to recognize legs enables the system to recognize and track pedestrians. This was done by checking the periodicity of the cluster shape and by feeding the gray scale images of the legs into a time delay neural network. The advantage of this approach is that it works in the case of a moving camera. Unfortunately, due to several costly steps, real-time implementation was not possible.

2.2 Limb Tracking

2.2.1 2-D Limb Tracking

Very early work by Akita [1] attempted to track body parts using a model-based approach. Body parts were recognized in the order: legs, head, arms, then trunk. The author assumes that the legs are the most stable to detect and the trunk is the least stable. Correlation was used to estimate the body parts position. When correlation fails, a set of predefined key frames are used to give a rough estimate of the position of the person. There are many assumptions in this method in addition to many simplifications such as foreground-background straightforward segmentation. Also, the motion was assumed to be known beforehand.

Leung and Yang [53] used a model that consisted of five ribbons and a trunk to label the different body parts. The features used were image edges and moving edges. Labeling was done according to some constraints such as ratios of length to width of various body parts. The choice of thresholds which may vary from one person to another is critical to this method. Chang and Huang [11] also used ribbons to track arms and legs. Labeling was simplified by imposing some assumptions. For example, the arms are assumed to be located above the legs in the scene.

Ju *et al.* [45] proposed a 2-D model-based approach to track the leg closer to the camera using connected planar patches. Their model was parameterized to accommodate for the deformations the patches undergo as the leg moves.

Niyogi and Adelson [58] tracked the legs of walking people by analyzing the X-T plane at different Y values. For a Y value around the legs area, a braided pattern appears on the X-T plane corresponding to the legs motion. Deformable contours (snakes) are used to fit the pattern. A simple model using a 5-stick figure (two sticks for each leg and one for torso) was used by simple line fitting. Gait recognition was performed by comparing the contour data of these snakes to distinguish different walkers. In their work, they assumed a rough knowledge of the heights of the head and feet and that the person is walking parallel to the image plane at a constant speed. They were able to achieve an 81% rate of classifying the walking action as belonging to one of five people.

Wren *et al.* [85] developed a real-time limb tracking system "Pfinder". It is based on blob tracking where each blob corresponds to a different body part (head, hands, feet, shirt, and pants) and has certain statistical properties. Initial labeling is done by having the person face the camera and stand in a specific pose. The system ran in real-time and handled occlusions robustly. However, the authors did not mention how sensitive the system is when body parts have the same color. The system was limited to tracking one person in the view of the camera with a static background.

2.2.2 3-D Limb Tracking

Much of the research in human body and articulated motion tracking has been in the area of 3-D tracking. The results in this area, however, are still limited. Most approaches have made use a predefined 3-D model. A model consists of primitives such as generalized cylinders [23,28,36,69] or superquadrics [26,46]. Superquadrics are generalizations of ellipsoids which have squareness parameters along the axes. The shape parameters of the 3-D model (such as limb lengths, etc.) were almost always assumed to be constant and the emphasis has been on pose recovery. The models were mainly specified by articulation points. Use of further constraints such as joint angle limits, collision, or dynamic constraints such as balance and gravity has been limited. Some methods used general articulated motion constraints such as in-plane rotation [35] and fixed-axis rotation [84]

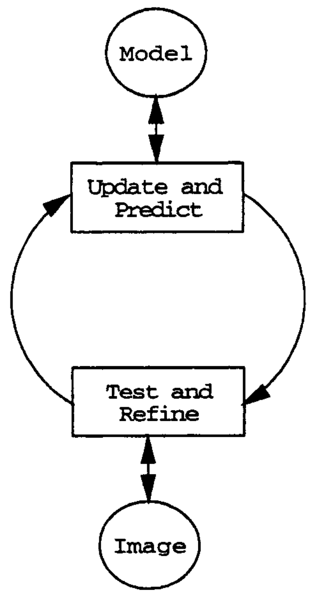


Figure 2.1 General interpretation cycle.

Some methods [23,26,36,59,60,62,69] adopted the general scheme for model-based approaches proposed by Kanade [47]. The general idea of this scheme is to have a feedback loop as depicted in Figure 2.1. The image and the model are treated as two different domains. The model supplies a set of hypothesis by predicting the future. These hypotheses are tested for validity, refined and sent back to the model domain so that the model is updated. Due to the complexity of the human body motion, a prediction process may generate a huge number of hypotheses. Testing therefore involves performing a search in a high dimensional space. If testing is not sufficiently reliable, the wrong hypothesis may be chosen. This problem was sometimes handled by either reducing the number of predicted hypotheses by restricting the movement freedom of the model [69] or making testing more reliable by assuming a easy segmentation or a clean input such as a synthesized image sequence [60]. Other approaches did not use the model feedback loop [13]. The flow of information was therefore in one direction: from the image domain to the model domain.

An alternative to this scheme using inverse kinematics was used by other methods [28,83,87,89]. This technique is often used in robotic control theory. There is a nonlinear mapping which maps the state space comprised of pose configuration and joint angles, etc. to the image space. It involves several coordinate transformations at articulation points and perspective projection. Inverse kinematics attempts to invert this mapping by linearization and gradient-based optimization to recover the changes in state parameters. Generally speaking, techniques based on inverse kinematics are computationally more efficient because of their use of gradient-based optimization. However, they are more likely to get stuck in a local minimum due to the highly nonlinear nature of the mapping. In addition, measurements may not be smooth for fast motions using standard video sampling rate.

Multiple cameras have been used by some methods [26,46,2] to reduce the effect of occlusions and much less often to utilize triangulation in 3-D information recovery. However, the majority still cannot handle significant occlusions. Azarbayejani and Pentland [2] proposed a technique to recover 3-D coordinates of the head and the two hands using a stereo setup. The features used were blobs which were obtained using the same technique as in [85]. Blobs can be considered as intermediate-level features. Other intermediate- and low-level features that were used include edges [28,26,69] or regions [83,87]. Some methods used higher-level features such as joints. The assumption was that the location of joints is provided or easily segmented [13,37,60,73,89]. This is a strong assumption since the detection of joints is generally a difficult task. Color and texture features were rarely used. Use of these features may be necessary if tracking people with loose-fitting clothes is desired.

Finally, there are other issues that still need to be addressed in 3-D tracking. One such issue is model initialization. Most methods do not handle initialization of the model and deal only with incremental pose estimation. Another issue is tracking of more than one person.

2.3 Motion Recognition

2.3.1 Recognition Based on 2-D Tracking

Goddard [27] used tracked point from Moving Light Displays (MLDs). The tracking information was assumed to be provided. The features used were relative angles and relative angular velocities of joints. Low level features were integrated to form higher level features. A connectionist model was used for recognition. Successful classification of walk, run, and skip action was achieved using this approach.

Guo *et al.* [29] classified actions into three categories: walking, running and everything else. They performed two-dimensional tracking on the silhouette of the body which is extracted using background subtraction. A stick figure is fitted to the silhouette to track the motion. Parameters extracted from the stick figure are then used for action recognition. This is done by performing Fourier transform on one period of these parameters to obtain the first few frequency components. The assumption is that one cycle of the action can be reliably segmented. Recognition is done by feeding the frequency components into a neural network. They used a number of samples both real and synthetic to achieve a 97.3% recognition rate.

Yacoob and Black [86] used 2-D tracking data in the form of parameterized models of the tracked legs. The recovered parameters over the duration of the action were then compressed using principle component analysis. Recognition was done by finding the nearest neighbor in eigenspace taking into consideration temporal shifts and differences in duration. Four actions were classified with an 82% accuracy.

Finally, Rangarajan *et al.* [67] matched motion trajectories using scale space. They used speed and direction parameters rather than locations to achieve translation and rotation invariance. The input was manually tracked points on several parts of the body performing the action. Given two speed signals, matching is performed by differencing the scale space images of the signals. Direction matching is done in the same way. In the experiments, a good match score was given for two walking actions performed by the same person, while a lower score was given when the person was different.

2.3.2 Recognition Based on 3-D Tracking

Upon successful 3-D tracking, motion recognition can make use of any or the recovered parameters such as joint coordinates and joint angles. Work done in this area has been limited to inputs of the form of MLDs obtained by placing markers on various body joints which are tracked in 3-D. Ideally, the 3-D tracking data would come from one of the 3-D limb tracking methods explained above.

Campbell and Bobick [10] used phase-space representation to model actions. In this representation, parameters are plotted without the time axis forming a series of points in space. Recognition was performed by considering two-dimensional subspaces of the phase space. This representation is useful because it provides invariance to variation in speed. Another way of dealing with differences in speed variations is by using the Dynamic Time Warping (DTW) technique [56]. In this technique, given two temporally aligned patterns of lengths M and N elements, an optimal pairing of the element of the two patterns that preserves order is generated. Gavrila and Davis [26] used DTW to distinguish three actions.

2.3.3 Recognition Based on Motion Features

Yamato *et al.* [88] used Hidden Markov Models (HMMs) to distinguish different tennis strokes. A segmentation step through background subtraction was first performed. The resulting image is then binarized with an appropriate threshold and then normalized to obtain a binary foreground image. This image is then divided into a grid of equal rectangular regions and the number of pixels in each region is counted. A feature vector is formed for every frame based on these numbers. For every action, a new HMM was learned. Recognition is done simply by selecting the HMM that was most likely to generate the given sequence of feature vectors (output symbols). The main advantage of such an approach is that it is easy to add a new action to the database. This is done by training an HMM and adding it to the set of HMMs. The approach, however, was sensitive to the shape of the person performing the stroke. Use of motion features rather than spatial features may have reduced this sensitivity.

Davis and Bobick [19] used what they called *motion-history* images (MHIs). An MHI represents motion recency where locations of more recent motions are brighter than older motions. A single MHI is used to represent an action. A pattern classification technique using seven Hu moments of the image was then used for recognition. They presented results of recognizing aerobic exercises performed by two actors, one for training and one for testing. The choice of an appropriate duration parameter used in the MHI calculation is critical. Temporal segmentation was done by trying all possible parameters. The system was able to successfully classify three different actions: sitting, arm waving and crouching. It remains to be seen how well this representation can capture long cyclic actions. For example, if two motions at different times take place in the same area, the latter will destroy the history information of the former.

Motion information extracted directly form the image sequence was also utilized by Polana and Nelson [65]. In their work, they used normal flow (the component of the flow field which is parallel to the gradient). The feature vector in their case was computed by temporally dividing the action into six divisions and finding the normal flow in each. Further more, each division is spatially partitioned into 4 by 4 cells. The summation of the magnitude of the normal flow at each cell was used to make up the feature vector. Thus, the vector was composed of 96 values. Recognition was done by finding the most similar vector in the training set using nearest centroid algorithm. The duration of the action was determined by calculating a periodicity measure [64]. This helps in correcting for temporal scale but not temporal translation. To overcome this problem, their technique matched the feature vector at every possible phase shift (six in this case). They tested their method using six different activities each performed several times by the same person and one activity performed by a toy frog. The activities were: walking, running, swinging (on a swing), skiing on a skiing machine, exercising on a machine, jumping jacks. Half the performances were used for reference and the other half was used for testing. A different person was used to supply samples for one of the activities (walking). The method worked fairly well which shows that the discriminatory power of the motion features used. More testing, especially using different actors, would be needed to demonstrate the usefulness of this method. Moreover, the actions used, with the exception of walking and running, were very different in terms of spatial scale which may have contributed to recognition success. Most of the computation time was spend in normal flow computation.

2.4 Hand Tracking, Hand Gesture Recognition and Lip-reading

The problems of hand tracking and hand gesture recognition had a particular interest motivated by demanding applications in gesture-driven control and sign language coding and interpretation. Consequently, there was an emphasis on the tracking problem [15,22,32,52,68] as well as on the recognition problem [8,12,18,20,25,39,66,77,81]. Many of the ideas mentioned in the previous sections were used here. In addition, various constraints were imposed to simplify the problem. But unlike the constraints on tracking and recognition human body motion, these are more tolerated due to the nature of the applications. For example, it is easier to require someone using a gesture-driven control application to put on colored gloves than to require people to wear a certain color in a surveillance application.

In lip-reading, the goal is to classify spoken words into one of the classes specified in a database. This problem is fundamentally different than human body motion analysis because the motion of the mouth is closer to deformable motion than to articulated motion. A few approaches addressed lip-reading. The features used were mouth opening shape [63], mouth elongation [55], tracked dots placed around the mouth [24], and optical flow [54]. Recognition was done by measuring distances among representations derived from these features. Yacoob and Black [86] used their human action recognition method (see Section 2.3.1) to perform lip-reading using optical flow features. Kirby *et al.* [49] did not perform recognition. Instead, they used principle component analysis to compress images from the sequence of the spoken word down to three coefficients per image to achieve a low transmission bandwidth.

Chapter 3

Motion Recognition

In [19], it was demonstrated that our visual capabilities allow us to perceive actions with ease even when presented with an extremely blurred image sequence of an action. Several snapshots from a similar sequence are shown in Figure 3.1. These experiments suggest that using motion alone to recognize actions may be favorable to reconstruction-based approaches.

Our method uses motion extracted directly from the image sequence. At each frame, motion information is represented by a *feature image*. Motion information is calculated efficiently using an Infinite Impulse Response (IIR) filter. A different method, though conceptually similar, was used by [19]. Unlike [19], an action is represented by several feature images rather than just one image. Actions can be complex and repetitive making it difficult to capture motion details in one feature image. Motion features were also used by [65]. In this case several motion features were extracted throughout the action duration. The features were based on normal flow and were very small in size (a 4×4 matrix). Most of computation time was spent in normal flow computation. Our choice of IIR filtering is motivated by the efficiency of this approach. The feature image used is not limited to a small size. Higher representation resolution could give more discriminatory

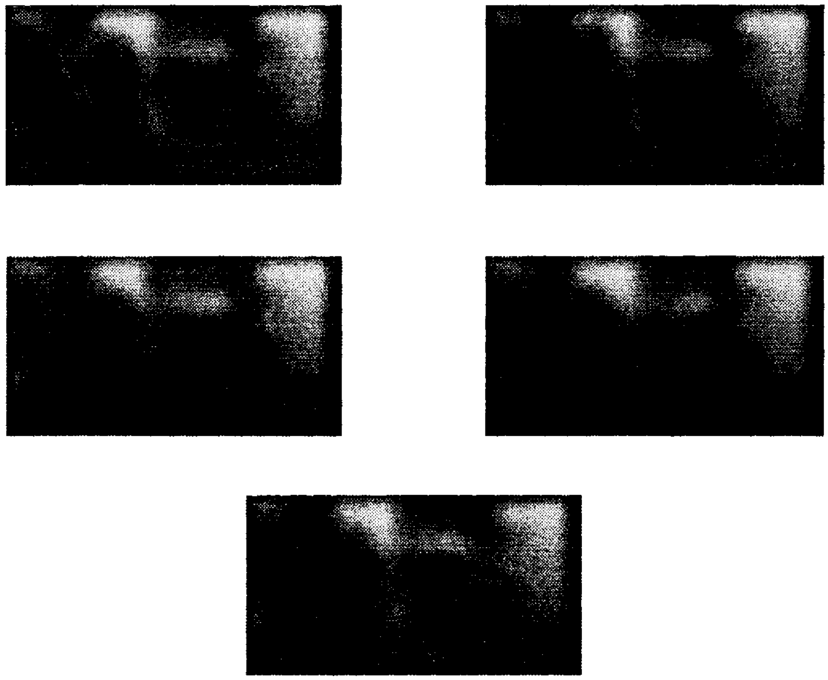


Figure 3.1 Snapshots from a motion sequence (of a person skipping) where the images have been blurred. Humans have no difficulty perceiving the action.

power when there is a similarity among actions. Dimensionality reduction using principle component analysis (PCA) is then utilized at the recognition stage.

To evaluate our approach we perform classification experiments involving eight different action classes each performed by 28 different people for a total of 232 samples, 168 of which were used as test samples.

After explaining the details of the feature image representation in Section 3.1, recognition is described in Section 3.2. The action data that was used for testing and its acquisition steps are presented in Section 3.3. Finally, results and analysis follow in Section 3.4.

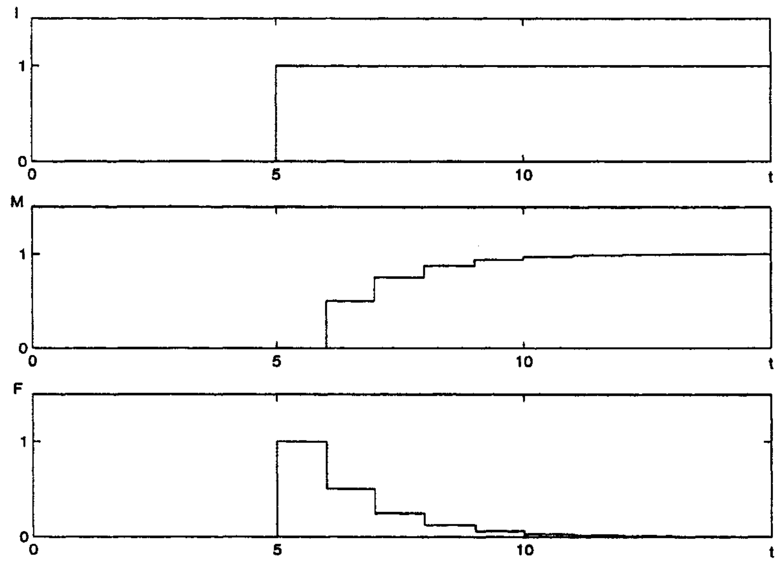


Figure 3.2 Filter response to a step signal.

3.1 Feature Image

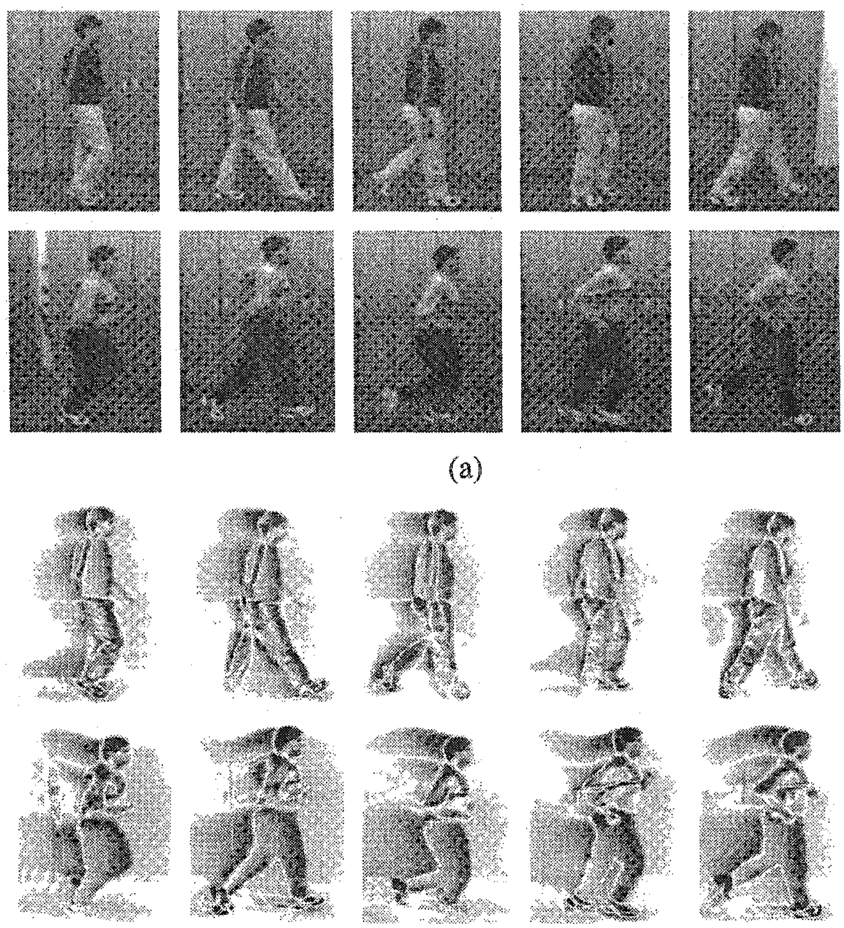
As mentioned earlier, an IIR filter is used to construct the feature image. In particular, we use the response of the filter as a measure of motion in the image. A slightly different formulation of this measurement has been used by Halevi and Weinshall [30]. The idea is to represent motion by its recency: recent motion will be brighter than older motion. This technique, also called *recursive filtering*, is simple, time-efficient and therefore, suitable for real-time applications. A weighted average at time i, M_i, is computed as

$$M_i = \alpha \times I_{i-1} + (1-\alpha) \times M_{i-1}, \tag{3.1}$$

where I_i, is the image at time i, and α is a fraction in the range 0 to 1. The feature image at time i, F_i, is computed as follows: $F_i = |M_i - I_i|$. Figure 3.2 is a plot of the filter response to a step function with α set to 0.5. F can be described as an exponential decay function similar to that of a capacitor discharge. The rate of decay is controlled by the parameter α. An α equal to 0 causes the weighted average, M, to remain constant (equal to the background) and therefore F will be equal to the foreground. An α equal to 1 causes M to be equal to the previous frame. In this case, F becomes equivalent to image differencing. Between these two extremes, the feature image captures temporal changes (features) in the sequence. Moving objects result in a fading trail behind them. In the one-dimensional case, a step edge in the image moving to the left will produce a feature image similar to that shown in Figure 1(c) (after replacing the time coordinate by a spatial coordinate). In fact, if the edge was moving one pixel at a time to the left, the feature image will be identical to Figure 1(c).

The speed and direction of motion are implicit in this representation. The spread of the trail indicates the speed while the gradient of the region indicates direction. Figure 3.3 shows several frames from a motion sequence along with the extracted motion features using this technique. Note that it is the contrast of the gray level of the moving object which controls the magnitude of F not the actual gray level value.

With the assumption that the height, h, of the person and his/her location in the image are known, feature images are sized and located accordingly. The feature image is computed in an box of dimensions $0.9h$ by $1.1h$ whose bottom is aligned with the base line and centered around the midline of the person. This is illustrated in Figure 3.4. The extra height is needed in cases of some actions that involve jumping. The width is large enough to accommodate the legs motion



(a)

(b)

Figure 3.3 An example of a walking and a running motion sequence. (a) Original images. (b) Filtered images (feature images) with $\alpha = 0.3$.

and the motion trails behind them. Details about how the person's height and location can be estimated will be explained in section 3.3.2.

The feature image values are normalized to be in the range $[0, 1]$. They are also thresholded to remove noise and insignificant changes (a threshold of 0.05 was found appropriate). Finally, a low pass filter is applied to remove additional noise.

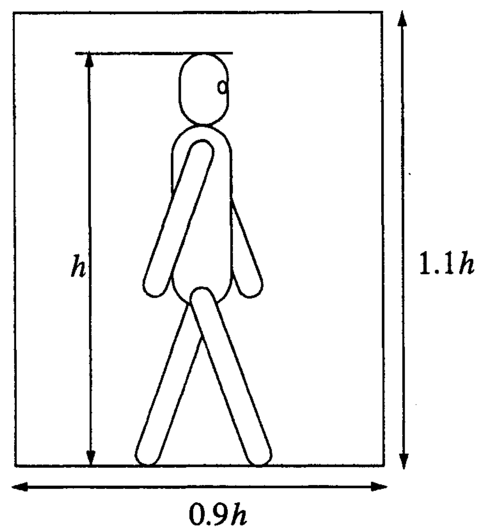


Figure 3.4 Feature image size selection.

3.2 Learning and Recognition

Our goal is to classify actions into one of several categories. We use the feature image representation calculated throughout the action duration. The idea is to compare the feature images with reference feature images of different learned actions and look for the best match. There are several issues to consider using this approach. Action duration is not necessarily fixed for the same action. Also, the method should be able to handle small speedups or slowdowns. Even if we assume actions are performed at the same speed, we cannot assume temporal alignment and therefore a frame-by-frame matching starting from the first frame should be avoided. The frame-to-frame matching process itself needs to be invariant to the actor's physical attributes such as height, size, color of clothing, etc. Moreover, since an action can be composed of a large number of frames, correlation-based methods for matching may not be appropriate due to their computationally intensive nature.

All these issues have been considered in the development of our recognition method. This section describes the details of our algorithm and addresses the issues mentioned above.

3.2.1 Magnitude and Size Normalization

As actions are represented as sequences of feature images, two types of normalization are performed on a feature image:

1. Magnitude normalization: Because of the way feature images are computed, a person wearing clothes similar to the background will produce low magnitude features. To adjust for this, we normalize the feature image by the 2-norm of the vector formed by concatenating all the values in all the feature images corresponding to the action. The values are then multiplied by the square root of the number of frames to provide invariance to action length (in number of frames).

2. Size normalization: The images are resized so that they are all of equal dimensions. Not only does this type of normalization works across different people but also it corrects for changes in scale due to distance from the camera, for instance.

3.2.2 Principle Component Analysis

3.2.2.1 Overview

Principle component analysis (PCA) is a technique which is commonly used for dimensionality reduction and for classification. Given N n-dimensional samples, $\mathbf{X} = \begin{bmatrix}\mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_N\end{bmatrix}$, $\mathbf{x}_i \in \Re^n$, a measure of scatter is the expected value of the squared between-sample distance. This scatter measure can be calculated as

$$\bar{d}_{\mathbf{X}}^2 = 2 \operatorname{tr} \Sigma_{\mathbf{X}}, \tag{3.2}$$

where $\Sigma_{\mathbf{X}}$ is the covariance matrix of $\mathbf{X}$. $\Sigma_{\mathbf{X}}$ is defined as

$$\Sigma_{\mathbf{X}} = \sum_{i=1}^{n} (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T, \tag{3.3}$$

where $\mu \in \Re^n$ is the mean of all samples. An orthonormal transformation $\Phi = [\phi_1 \phi_2 \ldots \phi_m]$ consisting of m n-dimensional vectors ϕ_i's is sought to map the original n-dimensional samples ($\mathbf{x}_i$'s) into a new set of m-dimensional vectors, where $m < n$ (dimensionality reduction). Let $\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_N \end{bmatrix}$ be the set of mapped vectors,

$$\mathbf{Y} = \Phi^T \mathbf{X}. \tag{3.4}$$

Hence, the scatter measure of $\mathbf{Y}$ is

$$\begin{aligned} \bar{d}_{\mathbf{Y}}^2 &= 2 \operatorname{tr} \Sigma_{\mathbf{Y}} = 2 \operatorname{tr} (\Phi^T \Sigma_{\mathbf{X}} \Phi) \\ &= 2 \sum_{i=1}^{m} \phi_i^T \Sigma_{\mathbf{X}} \phi_i \ . \end{aligned} \tag{3.5}$$

In PCA, it is desired to find the ϕ_i's that maximize this scatter measure. This implies that the expected value of the squared distance among the mapped vectors is as large as possible which make them appropriate for classification. It turns out that the m eigenvectors of $\Sigma_{\mathbf{X}}$ corresponding to the m largest eigenvalues are precisely those ϕ_i's. This transformation is also known as the discrete *Karhunen-Loeve (K-L) expansion*. Since ϕ_i is an eigenvector of $\Sigma_{\mathbf{X}}$,

$$\Sigma_{\mathbf{X}} \phi_i = \lambda_i \phi_i, \tag{3.6}$$

where λ_i is the eigenvalue associated with ϕ_i. Substituting equation (3.6) into equation (3.5) yields

$$\bar{d}_{\mathbf{Y}}^2 = 2\sum_{i=1}^{m} \phi_i^T \lambda_i \phi_i = 2\sum_{i=1}^{m} \lambda_i. \tag{3.7}$$

Thus, the scatter is proportional to the sum of the eigenvalues used.

Modeling and recognition using PCA is usually done in a number of steps. First, a number of samples (training data) is used to calculate the eigenvectors. Each sample is then projected in eigenspace using a selected number of eigenvectors. The resulting vector can be viewed as a point in eigenspace which has a much lower dimensionality than the original space. The vector values are sometimes referred to as coefficients because the original vector can be approximated by multiplying these coefficients by the basis (the eigenvectors). When a new sample is to be recognized, it is first projected in eigenspace in a similar manner. A classifier is then used in eigenspace to find which class in the training data the new sample belongs to. There are some variants of PCA that use other techniques such as SVD decomposition.

3.2.2.2 PCA for Action Recognition

PCA has been extensively used in the field of face recognition [6,31,50,80,82]. The use of PCA in action recognition has been limited, however. Of a particular relevance to this work is the work of Yacoob and Black [86]. In their method, the features used were based on tracking five body parts using the work of Ju *et al.* [45]. Each tracked part provided eight temporal measurements. Thus, in total, 40 temporal curves are used to represent an action. Training data is composed of these curves for every example action. Each training sample is composed by concatenating all 40 curves. The training data is then compressed using a PCA technique. An action can now be represented in terms of coefficients of a few basis vectors. Given a new action, recognition is done by a search process which involves calculating the distance between the coefficients for this action and the coefficients of every example action and choosing the minimum distance. Their method handles temporal variation (temporal shift and temporal duration) by parameterizing this search process using an affine transformation.

Our method differs in that an action is not represented by a single point in eigenspace but rather a manifold whose points correspond to the different feature images the action goes through. This moves the burden of temporal alignment and duration adjustments from searching in the measurement space to searching in eigenspace. We see two main advantages for doing this:

1. Reduction in search complexity: Because the eigenspace has a much lower dimension than the measurement space, a more exhaustive search can be afforded.
2. Increased robustness: PCA is based on linear mapping. Action measurements are inherently nonlinear and this nonlinearity increases as these measurements are aggregated across the whole action. PCA can provide better discrimination if the action is not considered as one entity but a sequence of entities.

Nayar *et al.* [57] used parameterized eigenspace manifolds to recognize objects under different pose and lighting conditions.

In our method, the training set consists of a actions each performed a certain number of times, s. For each of the as samples, normalized feature images are computed throughout the action duration. Let the j-th sample of action i consist of T_{ij} feature images: $F_1^{ij}, F_2^{ij}, \ldots, F_{T_{ij}}^{ij}$. A corresponding set of column vectors $\mathbf{S}_{ij} = \left[\mathbf{f}_1^{ij}\ \mathbf{f}_2^{ij}\ \ldots\ \mathbf{f}_{T_{ij}}^{ij}\right]$ is constructed where each $\mathbf{f}$ is formed by stacking the columns of the corresponding feature image. To avoid bias in the training process, a fixed number L of $\mathbf{f}$'s is used since the number of feature images T_{ij} for a particular sample depends on the action and how the action was performed. From every set of $\mathbf{f}$'s we select L evenly spaced (in time) set of vectors $\mathbf{g}_1^{ij}, \mathbf{g}_2^{ij}, \ldots, \mathbf{g}_L^{ij}$. L should be small enough to accommodate the shortest action. To ensure that the selected feature images for the samples of one action correspond to similar postures, the samples for each action are assumed to be temporally aligned. This restriction is removed in the testing phase. The grand mean, μ, of these vectors ($\mathbf{g}$'s) over all i's and j's is computed. The grand mean is subtracted from each one of the $\mathbf{g}$'s and the resultant vectors are the columns of the matrix $\mathbf{X} = \begin{bmatrix}\mathbf{x}_1 & \mathbf{x}_2 & \ldots & \mathbf{x}_N\end{bmatrix}$, where $N = asL$ is the total number of columns. The number of rows of $\mathbf{X}$ is equal to the size of the feature image. The first m eigenvectors $\Phi = [\phi_1\phi_2\ldots\phi_m]$ (corresponding to the largest m eigenvalues) are then computed as described in Section 3.2.2.1. Each sample $\mathbf{S}_{ij}$ is first updated by subtracting μ from each column vector and then projected using these eigenvectors. Let $\tilde{\mathbf{S}}_{ij} = \begin{bmatrix}\tilde{\mathbf{f}}_1^{ij} & \tilde{\mathbf{f}}_2^{ij} & \ldots & \tilde{\mathbf{f}}_{T_{ij}}^{ij}\end{bmatrix}$ be such that $\tilde{\mathbf{f}}_k^{ij} = \mathbf{f}_k^{ij} - \mu$. The projection into eigenspace is computed as

$$
\begin{aligned}
\mathbf{Y}_{ij} &= \Phi^T\tilde{\mathbf{S}}_{ij} \\
&= \begin{bmatrix}\mathbf{y}_1^{ij} & \mathbf{y}_2^{ij} & \ldots & \mathbf{y}_{T_{ij}}^{ij}\end{bmatrix}.
\end{aligned}
\tag{3.8}
$$

Each $\mathbf{y}_k^{ij}$ is an m-dimensional column feature vector which represents a point in eigenspace (the values are coefficients of the eigenvectors). $\mathbf{Y}_{ij}$ is therefore a manifold representing a sample action. We will refer to the set of all the $\mathbf{Y}$'s as the reference manifolds. Recognition will be performed by comparing the manifold of the new action to the reference manifolds as will be explained in the next section.

3.2.3 Recognition

As mentioned earlier, recognition is done by comparing the manifold of the test action in eigenspace to the reference manifolds. The manifold of the test action is computed in the same way as described above using the computed eigenvectors at the training stage. In this Section, we describe the distance measure that was used for comparison and explain how it is used for classification.

3.2.3.1 Distance Measure

The computed manifold depends on the duration and temporal shift of the action which should not have an effect on the comparison. Our distance measure can handle changes in duration and is invariant to temporal shifts. Given two manifolds $\mathbf{A} = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_l\end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_h\end{bmatrix}$, we define

$$d(\mathbf{A}, \mathbf{B}) = \frac{1}{l}\sum_{i=1}^{l} \min_{1 \le j \le h} \left\| \frac{\mathbf{a}_i}{\|\mathbf{a}_i\|} - \frac{\mathbf{b}_j}{\|\mathbf{b}_j\|} \right\| \tag{3.9}$$

as a measure of the mean minimum distance between every normalized point in $\mathbf{A}$ and every normalized point in $\mathbf{B}$. To ensure symmetry, the distance measure which we use is

$$D(\mathbf{A}, \mathbf{B}) = d(\mathbf{A}, \mathbf{B}) + d(\mathbf{B}, \mathbf{A}). \tag{3.10}$$

This distance measure is a variant of the Hausdorff metric (we use the mean of minima rather than the maximum of minima) which still preserves metric properties. The invariance to shifts is clear from the expression. In fact, $d(\ ,)$ is invariant to any permutation of points since there is no consideration for order altogether. This flexibility comes at the cost of allowing actions which are not similar, but somehow have similar feature images in a different order, to be considered similar. The likelihood of this happening, however, is quite low. This approach is similar to phase space approaches where the time axis is collapsed [10]. The temporal order in our case is not completely lost, however. The feature image representation has an implicit locally temporal order specification. This measure also handles changes in the number of points as long as the points are more or less uniformly distributed on the manifold. The normalization of points in equation (3.9) is effectively an intensity normalization of feature images.

3.2.3.2 Classification

Using the distance measure equation (3.10), three different classifiers have been considered:

1. Minimum Distance (MD): The test manifold is classified as belonging to the same action class the nearest manifold belongs to, over all reference manifolds. This requires finding the distance to every reference manifold.

2. Minimum Average Distance (MAD): The mean distance to reference manifolds belonging to each action class is calculated; and the shortest distance decides classification. This also involves finding the distance to every reference manifold.

3. Minimum Distance to Average (MDA) (also called nearest centroid): For each action, the centroid of all reference manifolds belonging to that action is computed. This is also a manifold with a number of points equal to the average number of points in each reference manifold belonging to the action. We do not interpolate to compute this manifold. Instead, the nearest points (temporally) on the reference manifolds are averaged to compute the corresponding point on the centroid manifold. A test manifold is classified as belonging to the action class with the nearest centroid. Testing involves calculating a number of distances equal to the num-

ber of action classes.

3.3 Action Data

3.3.1 Data Selection

To evaluate our recognition method, we recorded video sequences of eight actions each performed by 29 different people. Several frames from one sample of each action are shown in Figures 3.5 and 3.6. The actions are named as follows: Walk, Run, Skip, Line-walk, Hop, March, Side-walk, Side-skip. There are several reasons for our choice of this particular data set:

1. Discrimination becomes more challenging when there is a high degree of similarity among actions. Many of the actions we chose are very similar in the sense that the limbs have similar motion paths.

2. Rather than having the same person perform actions several times, we chose to have different people. This provides a more realistic data since in addition to the fact that people have different physical characteristics, they also perform actions differently both in form and speed. This would be a good test for the versatility of our approach. It can be seen from Figures 3.5 and 3.6 that people sizes as well as color of clothing are different. A few samples also had more complex backgrounds. Table 3.1 shows the variation action performance speed throughout the

Action	Minimum Duration (sec.)	Maximum Duration (sec.)
Walk	0.93	1.77
Run	0.70	0.93
Skip	1.10	1.73

Table 3.1: Variation in the duration of one cycle for the data set.

Figure 3.5 Several frames from Walk, Run, Skip, and March actions.

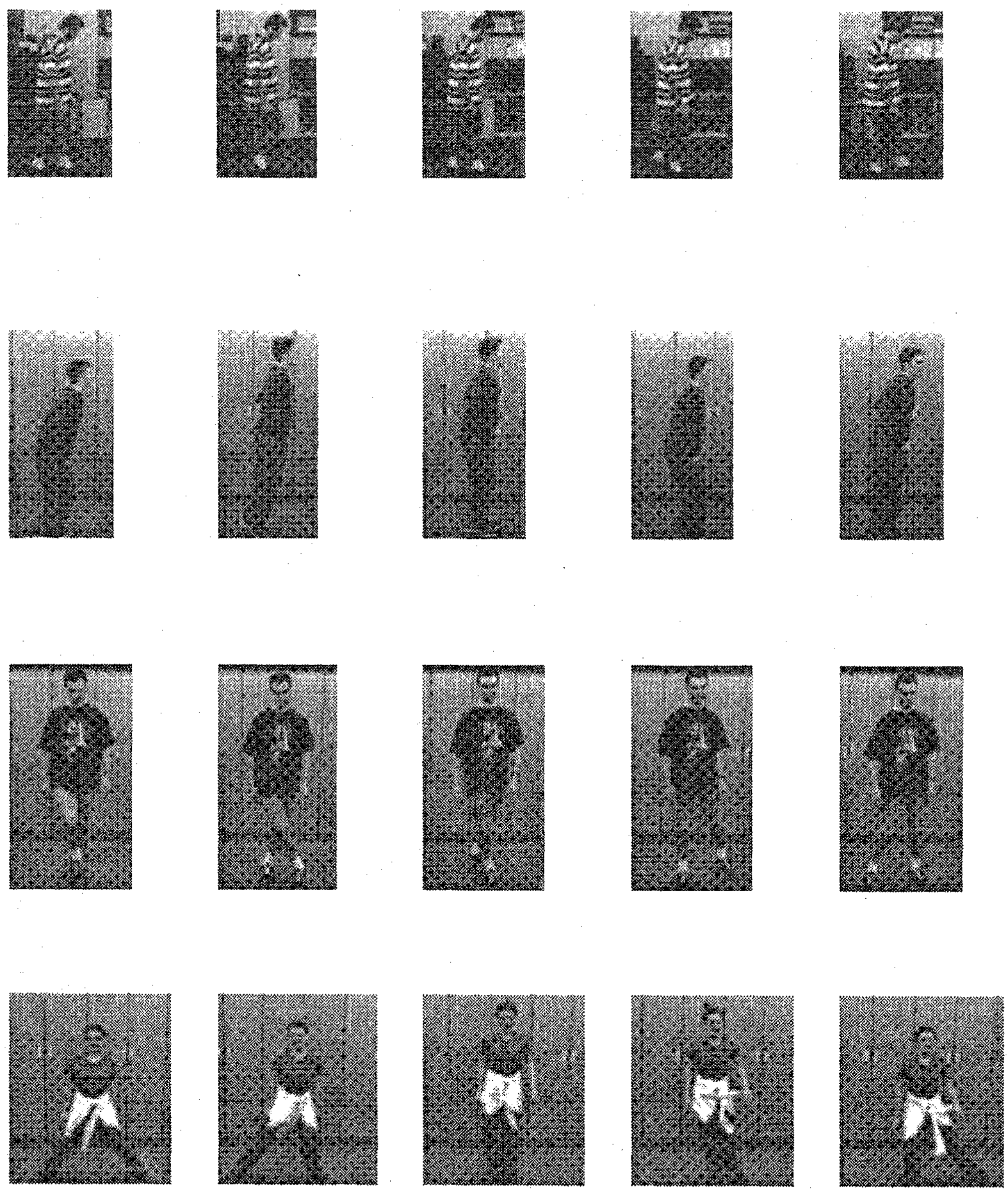


Figure 3.6 Several frames from Line-Walk, Hop, Side-walk, Side-skip actions.

Action	Minimum Duration (sec.)	Maximum Duration (sec.)
March	1.13	1.93
Line-walk	1.47	2.20
Hop	0.70	1.67
Side-walk	1.06	1.80
Side-skip	0.57	0.93

Table 3.1: Variation in the duration of one cycle for the data set.

data set. The table shows that the actions were performed at significantly varying speeds (more than double the speed in the case of Hop for instance).

3. Another consideration for a more realistic data set was that we avoided the use of a treadmill. Using a treadmill not only restricts speed but also simplifies the problem since the background is static relative to the actor.

To our knowledge, this is the largest set of action data compared to what has been reported in related research in terms of the number of subjects performing the actions multiplied by the number of actions.

3.3.2 Acquisition

The video sequences were recorded using a single stationary monochrome CCD camera mounted in such a way that the actions are performed parallel to the image plane.

In our approach, we assumed that the height (in the image plane) and location of the person performing the action are known. Recovering location is necessary to ensure that the person is in the center of the feature images. Height is used for scaling the feature images to handle differences in people's sizes and distance from the camera. To attain the recovery of these parameters, we tracked the subjects as they performed the action. Background subtraction was used to isolate the subject. A simple frame-to-frame correlation was used to precisely locate the subject horizontally in every frame. A small template corresponding to the top third of the subject's body where little shape variation is expected was used. The height was recovered by calculating the maximum blob height across the sequence. For the general case, our tracking method in the previous chapter can be used to locate the subject boundaries. Correlation can then be applied to find the exact displacement across frames. The computation of feature images as explained in Section 3.1 deals with the raw image data without any knowledge of the background. The information provided by the acquisition step is the location of the person throughout the sequence and the person's height.

3.4 Experimental Results

3.4.1 Classification Experiment

In our experiments, we used the data for eight of the 29 subjects for training (64 video sequences). This leaves a test data set of 168 video sequences performed by the remaining 21 subjects. The training instances were used to obtain the principle components. The number of selected frames (parameter L is Section 3.2.2.2) was arbitrarily set to 12. We will later show the effect of changing this parameter in further experiments. The resolution of feature images was also arbitrarily set to 25 horizontal pixels by 31 vertical pixels. Again, the effect of changing the resolution will be shown later. Decreasing the resolution has a computational advantage but reduces the amount of detail in the captured motion.

The training samples were organized in a matrix $\mathbf{X}$ as described in Section 3.2.2.2. The number

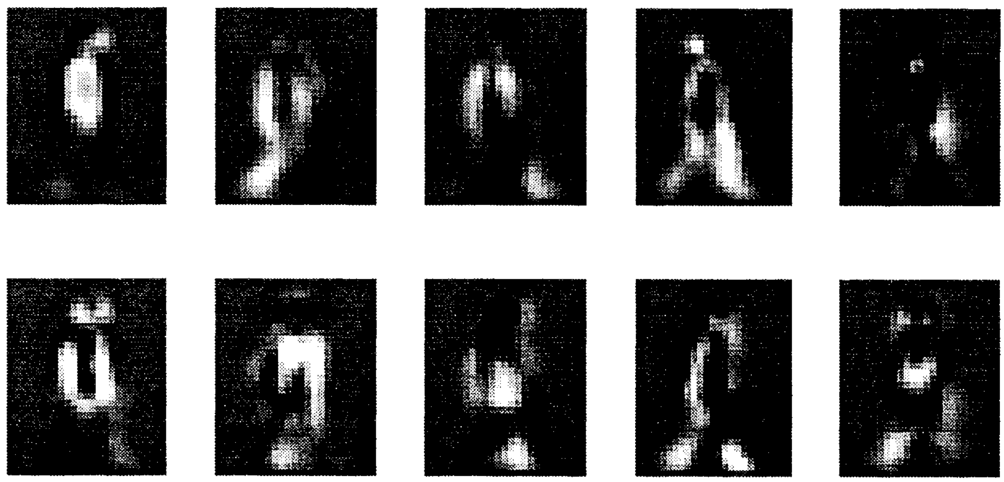


Figure 3.8 The first ten eigenvectors.

of columns is $asL = 8 \times 8 \times 12 = 768$. The number of rows is equal to the image size ($n = 25 \times 31 = 775$). The eigenvectors are then computed for the covariance matrix of $\mathbf{X}$. Most of the 775 resulting eigenvectors do not contribute much to the variation of the data. The plot $\lambda_i / \left(\sum_{k=1}^{n} \lambda_k \right)$ in Figure 3.7(a) illustrates the contribution of each eigenvector. It can be seen that from around the 50th eigenvector and on, the contribution is less than 0.5%. Figure 3.7(b) shows the cumulative contribution $\left(\sum_{k=1}^{i} \lambda_k \right) / \left(\sum_{k=1}^{n} \lambda_k \right)$. The curve increases rapidly during the first eigenvectors. The first ten eigenvectors alone capture more than 60% of the variation. The first 50 capture more than 90%. In Figure 3.8, the first ten eigenvectors are shown. The gray region corresponds to the value of 0 while the darker and brighter regions correspond to negative and

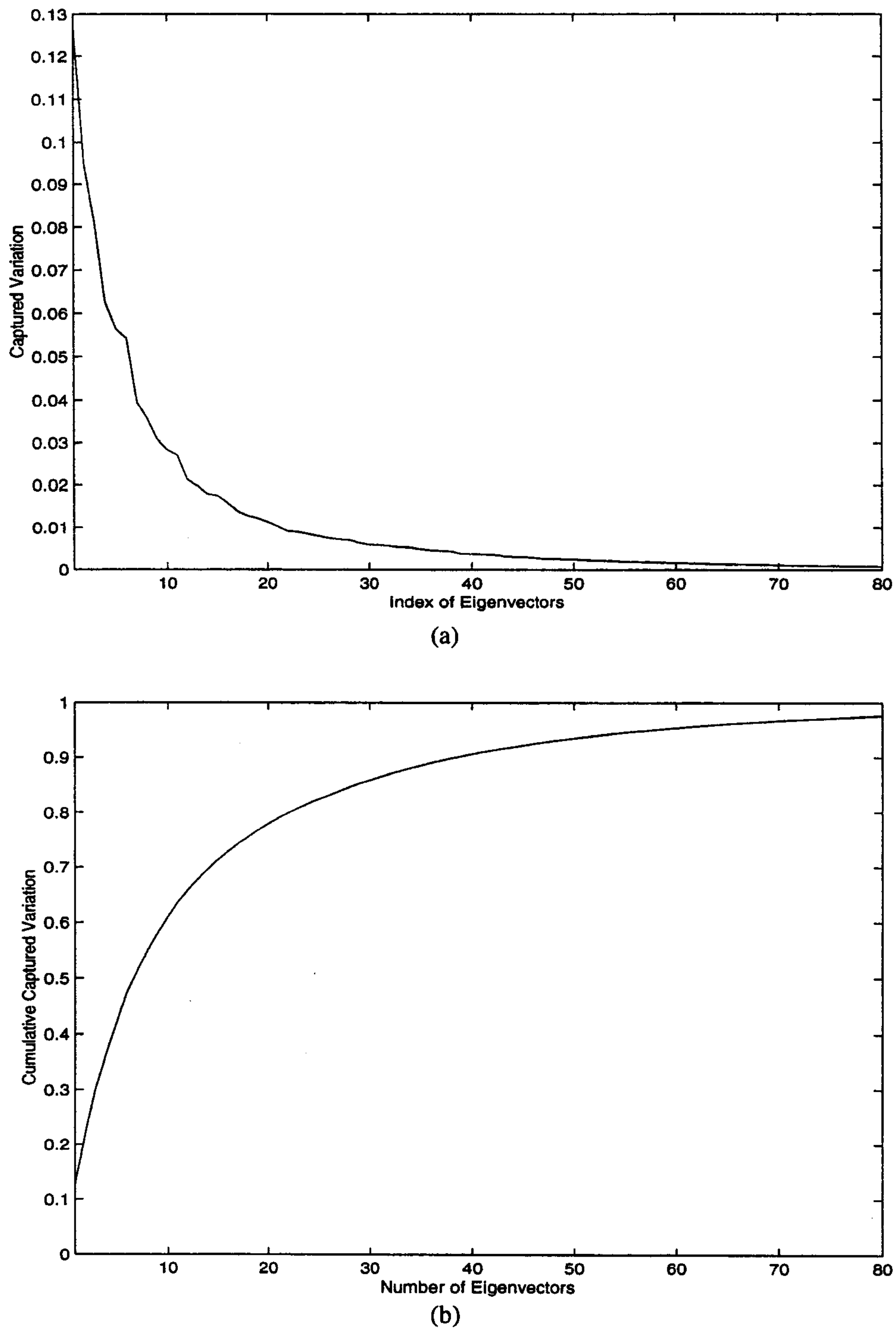


Figure 3.7 Eigenvectors contribution to variation in data. (a) Individual contribution. (b) Cumulative contribution.

positive values, respectively. It can be seen from the figure that different eigenvectors are tuned to

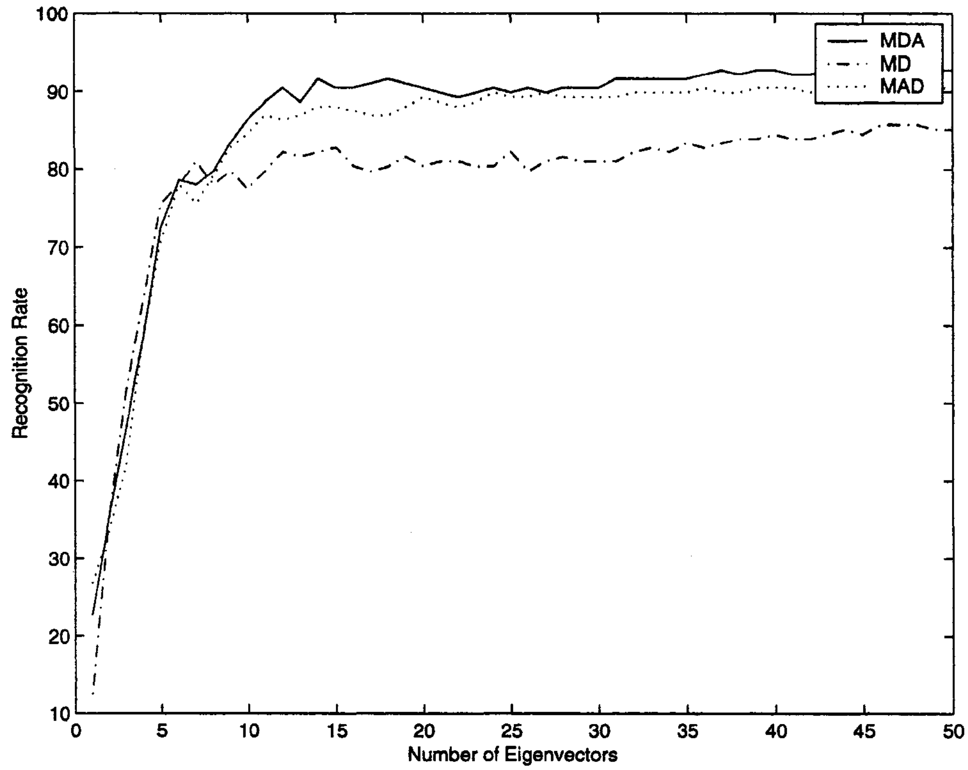


Figure 3.9 Recognition performance.

specific regions in the feature image.

In our experiments, the choice of m (the number of eigenvectors to be used) was varied from 1 to 50. Using a small m is computationally more efficient but may result in a low recognition rate. As m increases, the recognition rate is expected to improve and approach a certain level. Recognition was done on the 168 test sequences as described in Section 3.2.3 using all three classifiers (MD, MAD, MDA). Recognition rate was computed as the ratio of number of samples classified correctly to the total number samples. Figure 3.9 displays the recognition performance for the different classifiers as a function of m. It can be seen that the recognition rate rises rapidly during the first few values of m. At $m = 14$, the rate using MDA reaches over 91.6%. At $m = 50$, the rate is over 92.8% for MDA. MAD performance is slightly lower while MD is about 10% below. One explanation for this behavior is that some clusters are close to each other so that a point, which may be classified correctly using MDA, can be misclassified using MD. In later experiments, only the MDA classifier will be shown.

Yacoob and Black [86] reported a recognition rate of 82% but they had only four action classes.

Table 3.2 shows the confusion matrix for $m = 50$. Most actions had a perfect or near perfect

Action	Walk	Run	Skip	March	Line-walk	Hop	Side-walk	Side-skip
Walk	20	0	0	0	1	0	0	0
Run	1	20	0	0	0	0	0	0
Skip	2	0	15	2	0	2	0	0
March	1	0	1	19	0	0	0	0
Line-walk	0	0	0	0	21	0	0	0
Hop	0	0	0	0	0	21	0	0
Side-walk	0	0	0	0	1	0	19	1
Side-skip	0	0	0	0	0	0	0	21

Table 3.2: Confusion matrix.

classification except for the Skip action. Although the Skip action was classified correctly about 70% of the time, it was mistaken with Walk, March, and Hop actions numerous times. The 12 misclassified actions are shown in Figure 3.10. One person (number 15) had two action misclassified while the remaining people had at most one misclassification. When the correct action class was allowed to be within the first two choices, the number of misclassified actions becomes five. All these five actions (mostly Skip actions) were either executed erroneously or had a very low

	Walk	Run	Skip	March	Line-walk	Hop	Side-walk	Side-skip
1				1				
2							8	
3			1					
4								
5	5							
6								
7			1					
8								
9			6					
10								
11			6					
12							5	
13								
14								
15		1	4					
16								
17								
18				3				
19								
20								
21			4					

Figure 3.10 Misclassified actions for each subject. The numbers indicate the actions that were chosen incorrectly (1=Walk, 8=Side-skip).

color contrast.

To give an indication of the quality of classification, Figure 3.11 shows a confusion plot which represents the distance among test and reference actions averaged across all subject. The larger the box size, the smaller the distance it represents. The diagonal in the figure stands out and very few other boxes come near the sizes of the boxes at the diagonal. However, it can be seen that there is mutual closeness in matching between Walk and Skip actions (a Walk action is close to a Skip action and vise-versa). This was expected due to the high degree of similarity between these two actions.

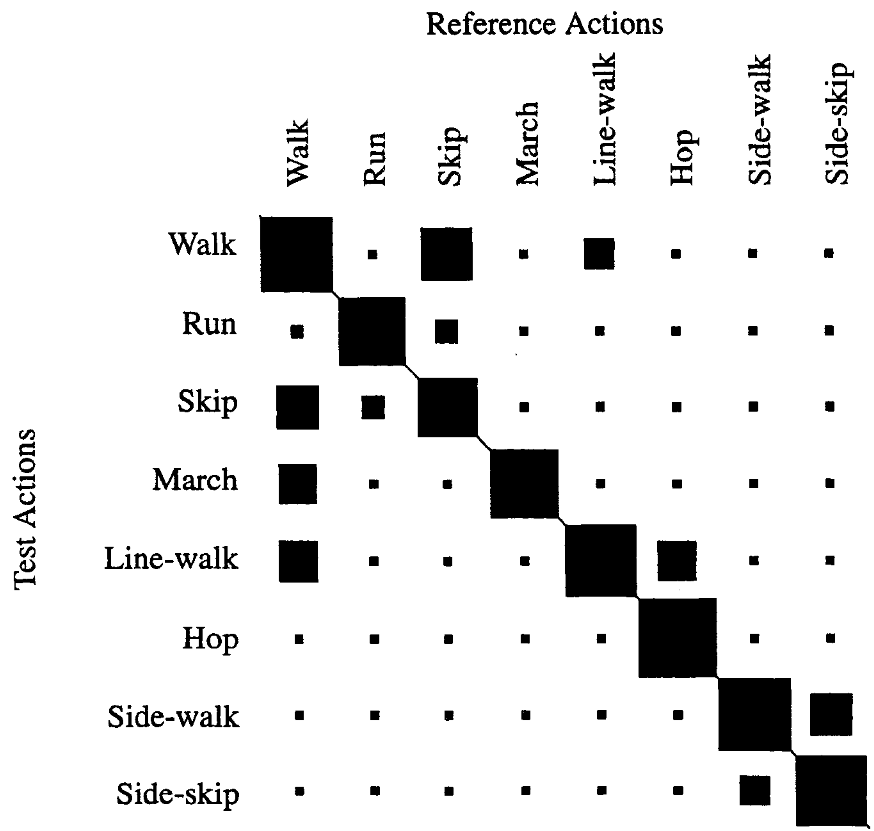


Figure 3.11 Confusion plot. The area of the squares indicates the distance using the distance measure in Section 3.2.3.1. The distances are averaged over all test samples.

3.4.2 Parameter Selection

3.4.2.1 Resolution

The resolution of feature images decides the amount of motion detail captured. In size normalization of feature images, a certain resolution must be chosen (See "Magnitude and Size Normalization" on page 31.) Figure 3.12 shows an example feature image and feature images normalized at different resolutions. The classification experiment was run with different resolutions to see if there is a resolution beyond which little or no improvement in performance is gained. Such a

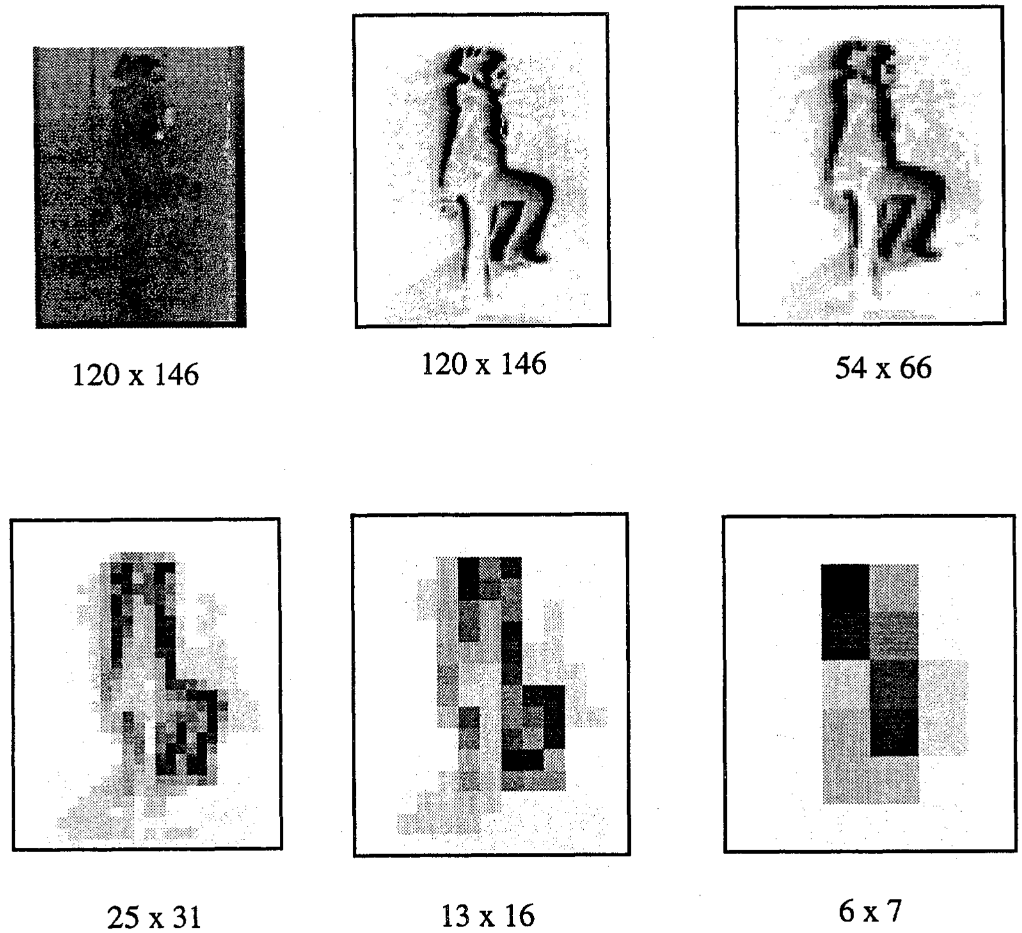


Figure 3.12 Original frame and its feature images at different resolutions.

reduced resolution has computational benefits. It also give an indication of the smallest "good" resolution which can be used to decide the maximum distance from the camera the action can take place (assuming the camera parameters are known). In Figure 3.13, the classification performance is shown for different resolution. It can be seen from the figure that increasing the resolution beyond 25×31 does not have any gain in performance.

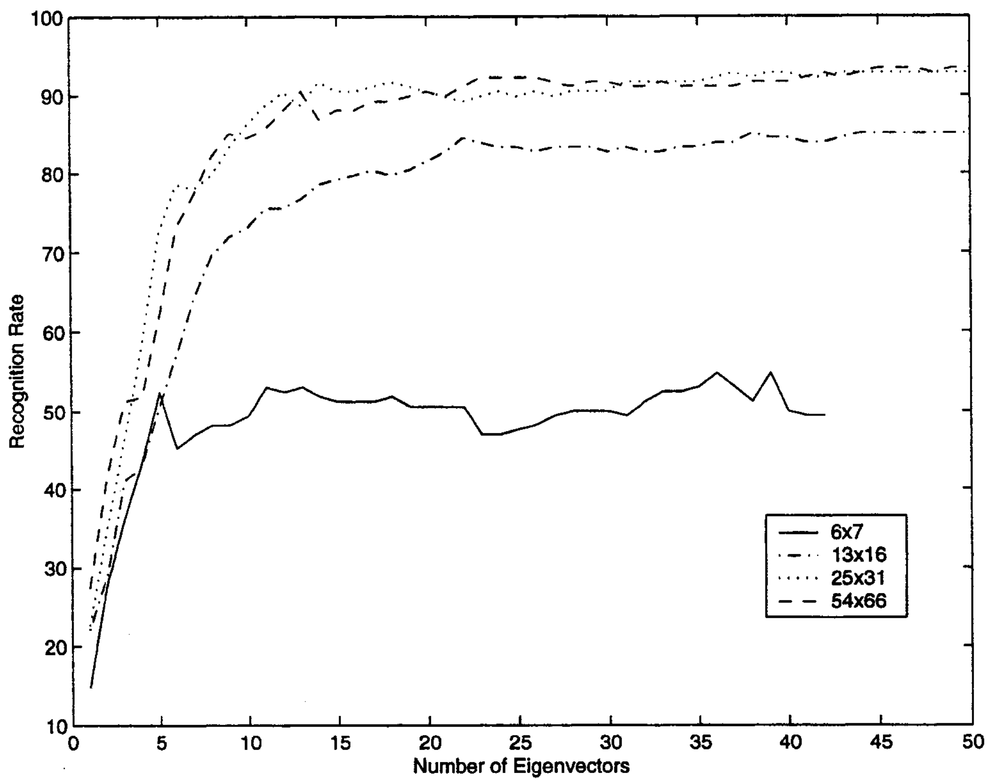


Figure 3.13 Effect of resolution on classification performance.

3.4.2.2 Number of Training Images

Parameter L is used in the training process to select the same amount of feature images from every training action sequence (See "PCA for Action Recognition" on page 33.) Here we see the effect of choosing different values for L on performance. Figure 3.14 shows the classification results for the values: 1, 2, 3, 4, 6, 12, 18, and 24. Values of 3 and above seem to have identical performance. This suggests that three feature images from an action sequence capture most of the variation in the different postures.

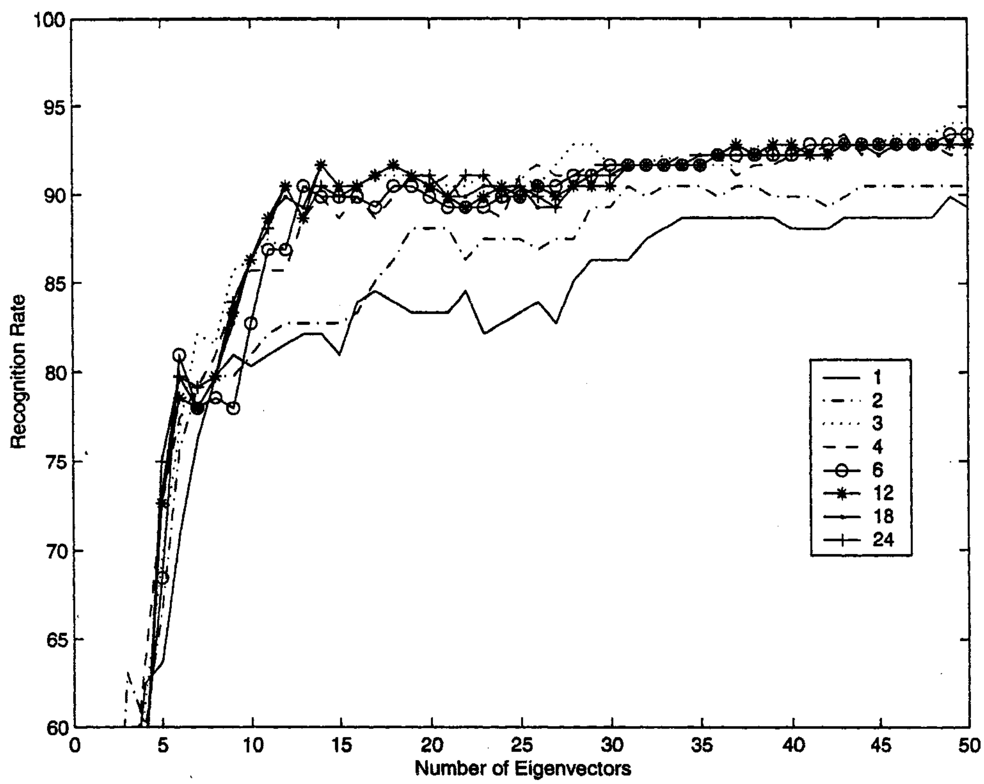


Figure 3.14 Classification performance for different L values.

3.4.3 Complexity

Testing an action involves computing feature images, projecting them in eigenspace, and comparing the resulting manifold with the reference manifolds. Computing feature images require low level image processing steps (addition and scaling of images) which can be done efficiently. Let n be the number of pixels in the scaled feature image according to the selected resolution. Using m eigenvectors, projecting a feature requires an inner product operation with each eigenvector and thus, a complexity of $O(mn)$. If the action has l frames, the time needed to compute the manifold is $O(lmn)$. Manifold comparison involves calculating the distance between every point on the action manifold and every point on every reference manifold. Assuming there are a action classes with s samples of each. If the average length of the reference actions is T, there will be $asTl$ distance calculations in the case of MD and MAD and aTl calculations in the case of MDA. Calculating a distance between two points in an m-dimensional eigenspace is $O(m)$. Therefore, recognizing an action using MD or MAD is $O(asTlm)$ while in the case of MDA, it is only $O(aTlm)$. In our experiments, $a = 8$, $s = 8$, $T \approx 37$, $m = 50$, and $n = 25 \times 31 = 775$.

The total complexity for MDA is therefore, $O(lmn) + O(aTlm)$, or $O(l)$ since the remaining variables are constant. This demonstrates the efficiency of this method and its suitability for a real-time implementation. On-line implementation is also possible where the distance measure is updated upon receiving new frames, requiring a small number of comparisons per frame. This allows an incremental recognition such that certainty increases as more frames are available. The choice of the implementation approach depends on the application at hand. We left the actual real-time implementation as a future work.

3.5 Summary

This chapter describes a motion recognition approach. The approach is based on low level motion features which can be efficiently computed using an IIR filter. Once computed, motion features at every frame which we call feature images are compressed using PCA to form points in eigenspace. An action sequence is thus mapped to a manifold in eigenspace. A distance measure was defined to test the similarity between two manifolds. Recognition is performed by calculating the distances to some reference manifolds representing the learned actions.

Experimental results for a large data set (168 test sequences) were presented and recognition rates of over 92.8% have been achieved. Complexity was also analyzed. The results demonstrate the promise and efficiency of the proposed approach.

Chapter 4

Model-based Filtering

4.1 Introduction

In cluttered image sequences, certain scenarios may pose problems for the pedestrian tracking system. We are particularly concerned with problems due to occlusions, similarity of a pedestrian color to objects behind the pedestrian, and false objects such as trees moved by the wind. Model-based filtering in addition to some other heuristic measures help resolve many of these problems. In this task we explored model-based filtering and provided solutions to the problems mentioned above. The filtering technique is explained in this report followed by experimental results showing its usefulness.

4.2 3-D Model Filtering

Pedestrians in the scene move in world coordinates. Therefore, to be able to estimate pedestrian parameters most accurately, we choose to do filtering in the 3-D world coordinates. The disadvantage of this approach is that camera calibration as well as knowledge about the scene geometry becomes necessary. Fortunately, the system is not very sensitive to the calibration parameters and therefore, coarse calibration was used. Knowledge about the scene geometry is limited to identifying flat surfaces where pedestrians walk and elevation differences among them. This is done manually but can be also done through a simple user interface. Initial steps for the creation of a calibration user-interface have been undertaken.

Filtering was done with the aid of the extended Kalman filter. Given a sampling interval δt, the discrete-time dynamic system for the pedestrian model can be described by the following equation:

$$\mathbf{x}_{t+1} = \mathbf{F}\mathbf{x}_t + \mathbf{v}_t, \tag{4.1}$$

where $\mathbf{x} = \begin{bmatrix} x & \dot{x} & y & \dot{y} \end{bmatrix}^T$ is the state vector consisting of the pedestrian location, (x, y) and velocity, $(\dot{x}, \dot{y})$, $\mathbf{F}$ is the transition matrix of the system given by $\begin{bmatrix} 1 & \delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}$, and $\mathbf{v}_t$ is a sequence of zero-mean, white, Gaussian process noise with covariance matrix $\mathbf{Q}$, x is the ground distance between the center line of the camera and the left side of the patch, and y is the ground distance between the camera optical center and the patch. $\mathbf{Q}$ is equal to $\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \end{bmatrix} q$, where

$\mathbf{A} = \begin{bmatrix} \frac{(\delta t)^3}{3} & \frac{(\delta t)^2}{2} \\ \frac{(\delta t)^2}{2} & \delta t \end{bmatrix}$ and q represents the variance of the acceleration.

The prediction and estimation equations become

$$
\begin{aligned}
\hat{\mathbf{x}}_{t+1} &= \mathbf{F}\mathbf{x}_t, \\
\hat{\mathbf{P}}_{t+1} &= \mathbf{F}\mathbf{P}_t\mathbf{F}^T + \mathbf{Q}, \text{ and}
\end{aligned}
\tag{4.2}
$$

$$
\begin{aligned}
\mathbf{K}_{t+1} &= \hat{\mathbf{P}}_{t+1}\mathbf{H}^T(\mathbf{H}\hat{\mathbf{P}}_{t+1}\mathbf{H}^T + \mathbf{R}_t)^{-1}, \\
\mathbf{x}_{t+1} &= \hat{\mathbf{x}}_{t+1} + \mathbf{K}_{t+1}(\mathbf{z}_{t+1} - \mathbf{h}(\hat{\mathbf{x}}_{t+1})), \\
\mathbf{P}_{t+1} &= (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{H})\hat{\mathbf{P}}_{t+1}, \text{ respectively.}
\end{aligned}
\tag{4.3}
$$

Here, $\hat{\mathbf{x}}$ and $\hat{\mathbf{P}}$ are the predicted state vector and state error covariance matrix, respectively. $\mathbf{x}$ and $\mathbf{P}$ are the previously estimated state vector and state error covariance matrix. $\mathbf{K}_{t+1}$ is the Kalman gain at $t+1$, $\mathbf{z}$ is the measurement, $\mathbf{R}_t$ is the covariance matrix for the measurement error given, and $\mathbf{H}$ be the Jacobian of the nonlinear measurement function $\mathbf{h}$ which is an inverse perspective projection function.

This filtering process estimates the pedestrian parameters (location and speed) in 3-D and thus, can produce reliable results in cases of occlusion or disappearance of some pedestrian features for any reason. The problem with false objects is handled using a number of heuristics:

1. If the blobs generated are too small, they are automatically eliminated.
2. A blob is considered at the pedestrian level only if it is tracked successfully over a certain number of frames. This eliminates blobs that appear then disappear momentarily (such as most blobs that appear due to tree motion back and forth).
3. The system can be given information about the scene. In particular, the locations where pedestrians can be expected to appear. Thus, the system will not instantiate pedestrian boxes except at these locations.

4.3 Experimental Results

The system was tested on a few cluttered sequences, one of which is shown in Figure 4.1. This sequence was taped during a snow storm. One can see the effect of the wind on the tree which resulted in false blobs in the difference images (frames 36, 66, 104, 140). The system was able to deal with this and did not generate any pedestrian boxes for the tree. The figure also demonstrates the successful tracking in case of minor pedestrian occlusion.

Figure 4.2 shows another tracking sequence which involves missing pedestrian blob information due to more severe occlusion (frame 32) and similarity of color to the background (frame 44). It also shows what happens when two pedestrian walk past each other. Kalman filtering is essential here because it provides good prediction when there is little or no data. The blob-pedestrian relationship refinement procedures guarantee that the pedestrian will be related to the correct blobs when data is available again.

Figure 4.3 shows the computed RMS errors for two pedestrians and shows the convergence of state parameters.

4.4 Conclusion

In this task we addressed some frequent problems that appear especially in cluttered image sequences. We used the extended Kalman filter and a number of heuristics in an attempt to overcome these problems. The results we obtained were promising and demonstrated the success of our techniques.

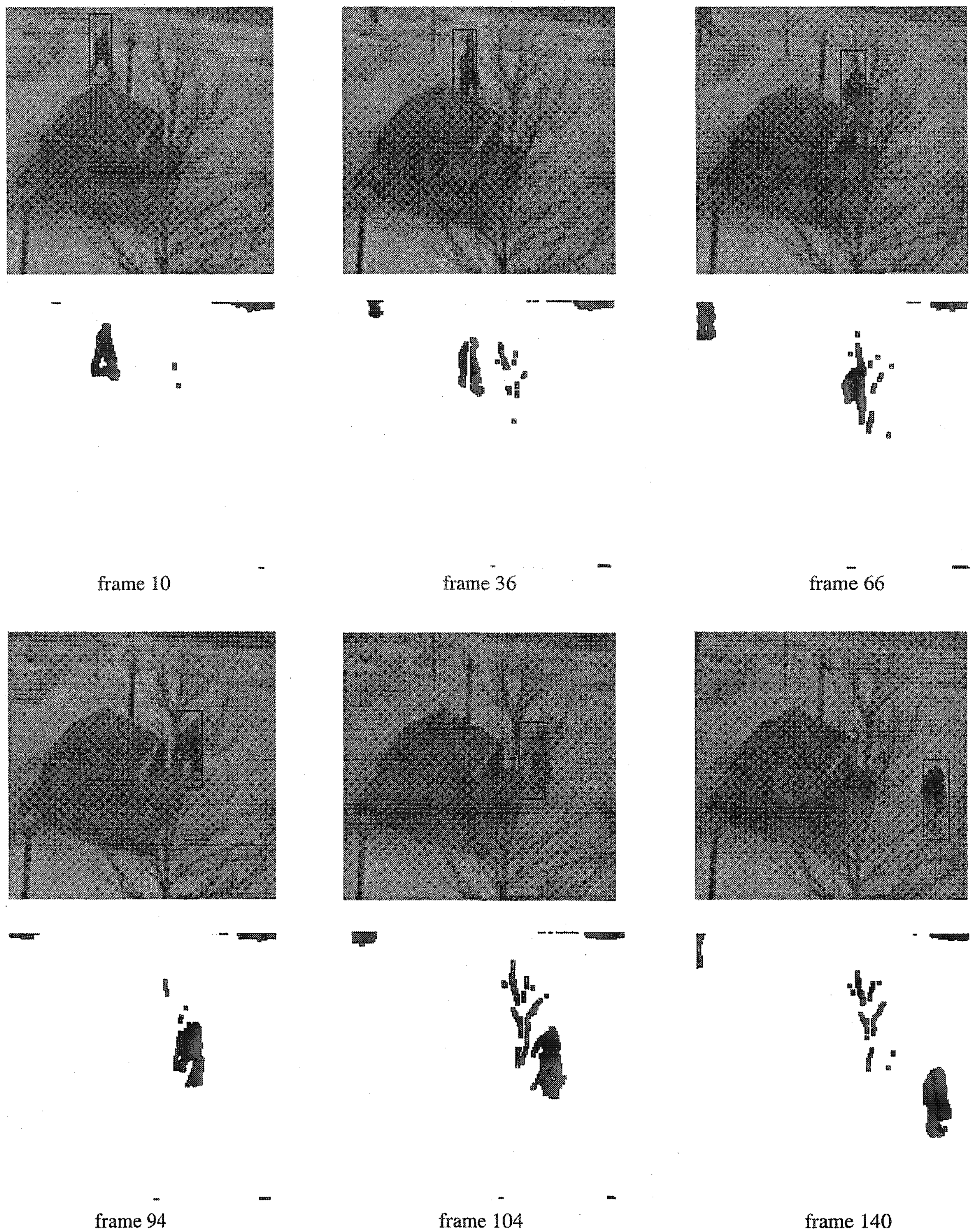


Figure 4.1 Tracking sequence of a pedestrian in different occlusion situations.

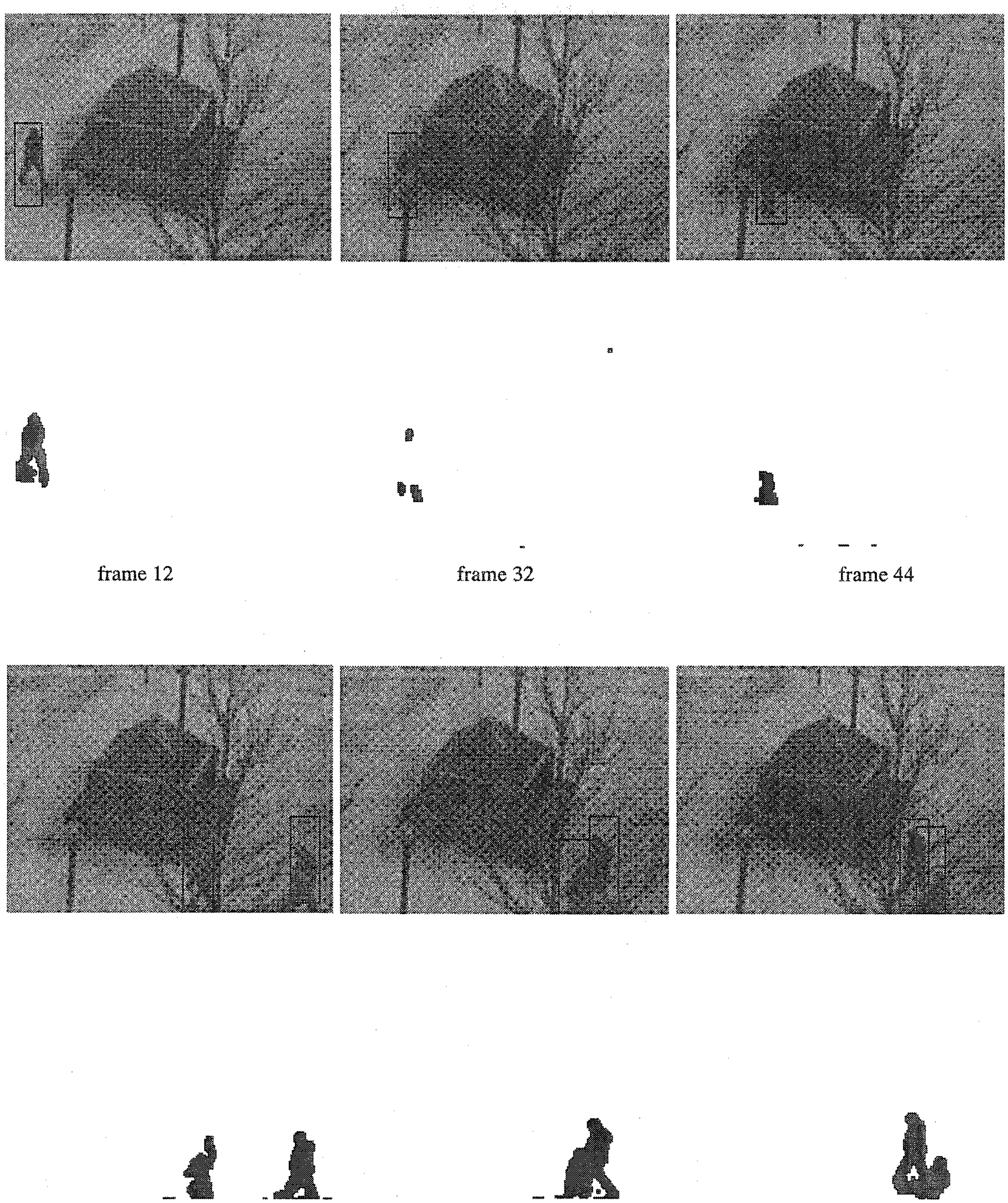


Figure 4.2 Tracking sequence demonstrating occlusions and pedestrian overlap.

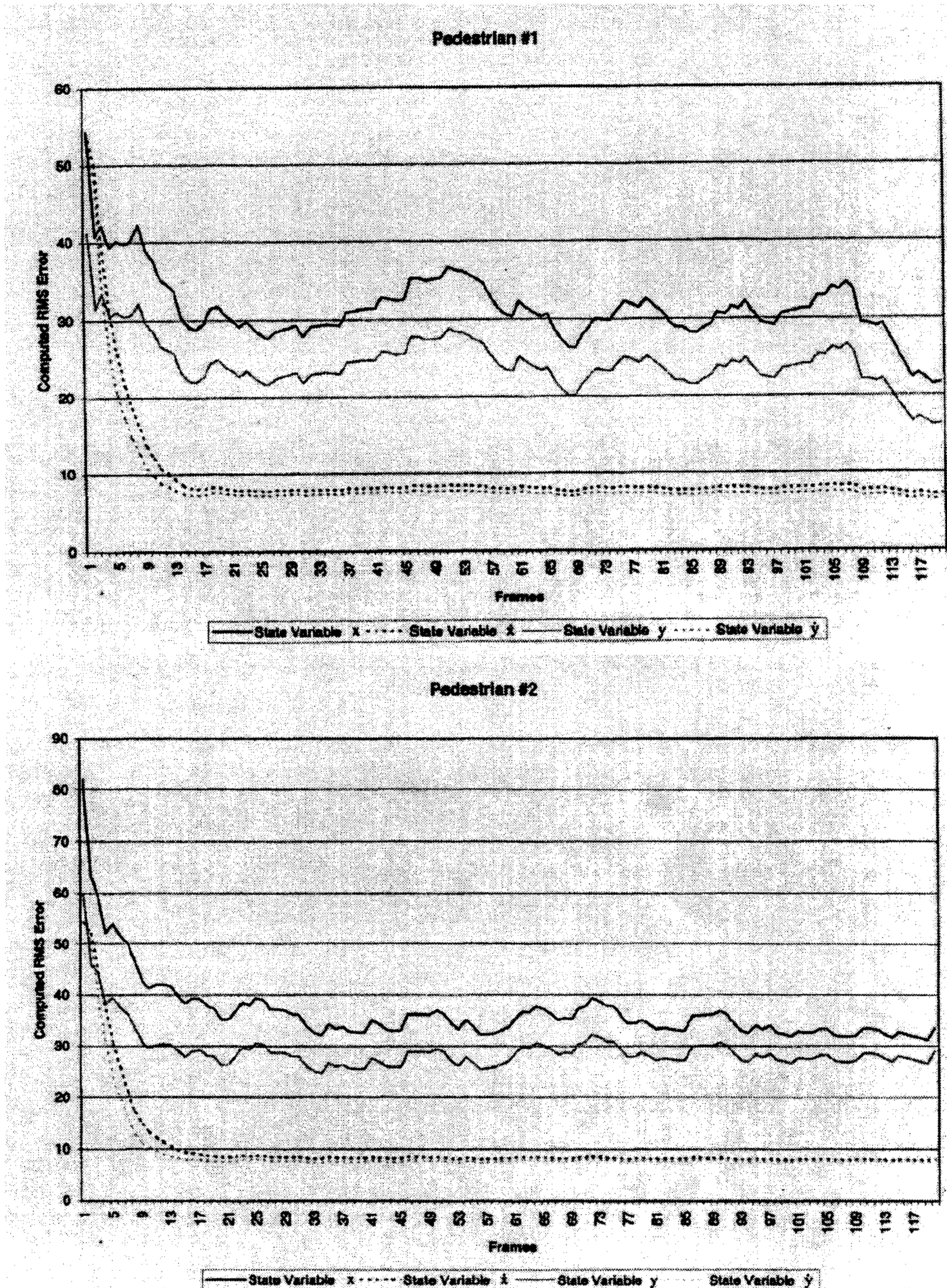


Figure 4.3 Computed RMS errors for two pedestrians.

Chapter 5

Active Deformable Models

5.1 Introduction

Active deformable models have been a popular technique in vision-based tracking of non-rigid bodies. The human body as a good example of a non-rigid body is therefore ideal for this application. Deformable models get attracted to the contour of the tracked object and change their shape according to the object contour's shape. Since deformable models take the shape of the tracked object, using them on pedestrians can provide valuable shape information. For example, a shape recognition technique can be used to distinguish pedestrians from other moving objects. This is useful in cluttered scenes where moving objects need not be predominantly pedestrians. In this chapter, we test the use of active deformable models in tracking pedestrians. The next section describes the technique that we used. Then, experimental results are presented. Finally, the conclusion is given.

5.2 Active Deformable Models

Our method uses active deformable models (i.e. snakes) to extract contour information from image measures. The snakes deform based upon statistics derived from the image data, resulting in a contour model for objects in the workspace. An advantage of snakes in general is that these models can track partially-occluded objects or semi-rigid targets [16][78].

The traditional deformable model was first proposed by Kass et al. [48]. It is a parametric curve S of the form

$$(S(u) = (x(u), y(u))', u \in [0, 1]), \tag{5.1}$$

where x and y are the coordinates of the curve. The curve is placed onto a potential field derived from the following energy equation:

$$E = \frac{\alpha}{2}\oint\left|\frac{\partial S(u)}{\partial u}\right|^2 du + \frac{\beta}{2}\oint\left|\frac{\partial^2 S(u)}{\partial u^2}\right|^2 du + \rho\oint P(I) du \tag{5.2}$$

where α, β, and ρ are weights. The first term corresponds to the internal force from tension, the second term corresponds to the internal force from curvature, $P(I)$ is the potential induced by the image (edges, corners, or dark spots on the image) along the curve. The energy along the length of the curve is minimized by allowing the model to change shape and position.

Other formulations may include an external energy term (e.g., by employing user-selected points as attractors). The problem with both of these formulations is that in the absence of image energy, these models collapse to a point. Pressure snakes (balloons) [14] have been developed to alleviate this problem by adding an internal pressure term P_r to force the model to expand.

$$E = \frac{\alpha}{2}\oint\left|\frac{\partial S(u)}{\partial u}\right|^2 du + \frac{\beta}{2}\oint\left|\frac{\partial^2 S(u)}{\partial u^2}\right|^2 du + \rho\oint P_o(I) du + P_r \tag{5.3}$$

Unfortunately, the constant pressure term introduces new problems with the model. For instance, the initial placement of the snake had to be entirely within the target. This leeds to dynamic pressure models.

Several forms of dynamic pressure models were proposed by Ivins and Porrill [40]. The pressure models are based upon first order statistics and utilize a seed region of the image to identify positive vs. negative pressure regions. That is, image regions that are statistically similar to the seed region yield positive pressure while image regions that are some number of standard deviations away from the seed mean will yield negative pressure. When a portion of the contour is in a positive region, it will expand away from the center of the contour. When the contour portion is in a negative region, it will contract toward the center. It follows that the minimum energy of the contour lies on the pressure boundary between positive and negative.

For our particular method, we use a dynamic statistical pressure snake that does not require an image energy term. We use an energy function with only internal energy and the dynamic statistical pressure model. The pressure model is given by

$$P_r(S) = \left(\frac{S}{u}\right)^{\perp}\left(1 - \frac{|I(S) - \mu|}{\sigma k}\right), \tag{5.4}$$

where S is the curve, μ and σ are the mean and standard deviation of the seed region, and k is a user specified parameter. There are issues with this model (and snakes in general) related to the automation of seed selection and the automation of the selection of the k parameter.

The snake model that we use is re-parameterized by automatically adding and deleting control points to S. In addition, we set the tension weight at least as high as the other snake parameters. This causes control points to form an even and relatively dense distribution around the object con-

Figure 5.1 Snapshots showing snake tracking pedestrians.

tour.

5.3 Experimental Results

We used several pedestrian sequences to test the snake behavior on pedestrians. The images were close-ups of pedestrian to increase the detail level of pedestrian contours. As can be seen in Figure 5.1, the snake converged to the pedestrian contour in every case. The snake is a concise representation of shape containing only a few parameters and can be used to do shape recognition.

The current implementation uses only one snake for testing purposes. Future implementations will have multiple snakes and will allow merging and splitting of snakes. Also, snakes by nature are sensitive to low contrast between foreground and background. In our experiments, we used subjects with enough contrast. One way to solve this problem is to use a prior snake model which is a parameterization of a pedestrian contour. This allows the snake to assume the presence of contour features despite low contrast in certain regions. The current processing was performed on raw intensity data. Using difference images with an appropriate threshold may also help alleviate this problem.

5.4 Conclusion

In this work, we applied active deformable models (snakes) to pedestrian tracking. These models were able to take the shape of the pedestrian contour and track it over time. The results are promising and demonstrate that snakes can be a useful tracking tool which can be integrated into our pedestrian tracking system for added robustness as well as recognition power.

Chapter 6

Field Demonstration

6.1 Introduction

This chapter summarizes our findings from a real demonstration of the pedestrian detection/tracking system at an intersection near the University of Minnesota. The results were very promising and the system was able to track pedestrians as they crossed the street in the middle of vehicles moving at a direction perpendicular to the pedestrians' direction of motion.

6.2 Demonstration

We demonstrated our pedestrian detection and tracking system on November 3 and 4, 1999 at an intersection next to the Harvard Ramp at the University of Minnesota. We connected our system to a traffic light and tried to flash the red light when there was a pedestrian at the crossing. This test was conducted to help us learn how the proposed system handles various simple and complex pedestrian scenarios, including different walking speeds, partial and full obstructions, and pedestrians meeting and passing each other. The accuracy of the system was around 90 percent. We had some problems with the moving shadows of the surrounding buildings and with pedestrians who crossed the street further away from the intersection.

Figure 6.1 Camera view of the intersection.

Figure 6.2 Computer hardware used to process the incoming video.

Figure 6.3 The traffic light is flashing red when there are pedestrians at the intersection.

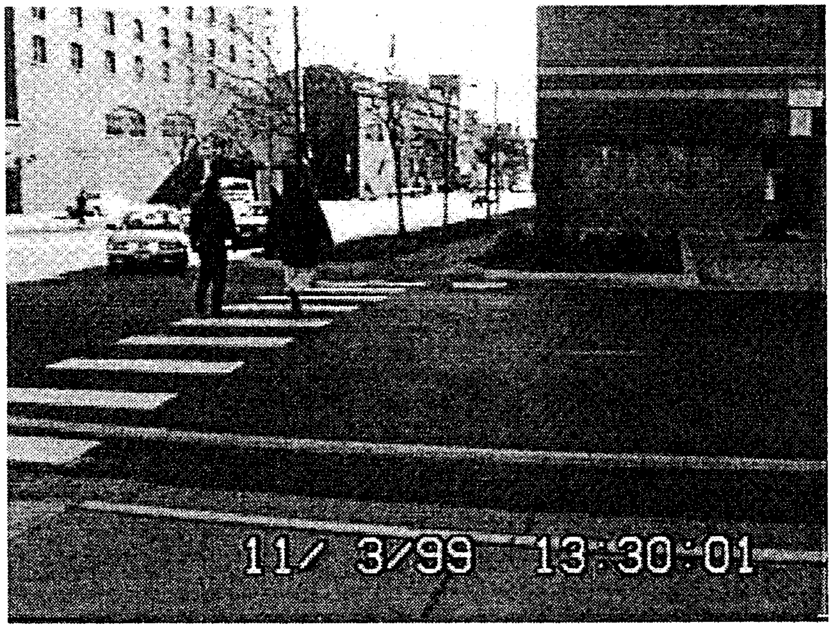


Figure 6.4 Another view of the intersection.

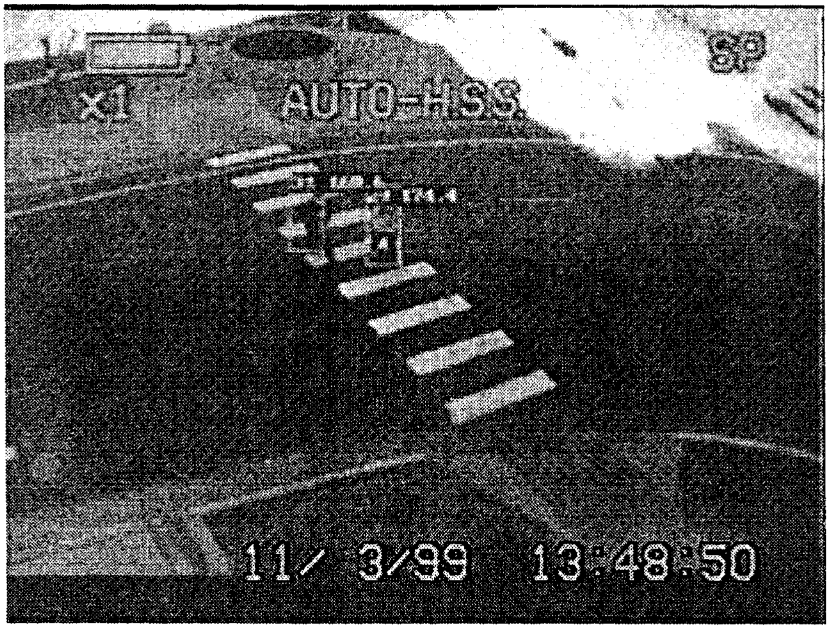


Figure 6.5 The boxes follow the pedestrians as they move through the intersection.

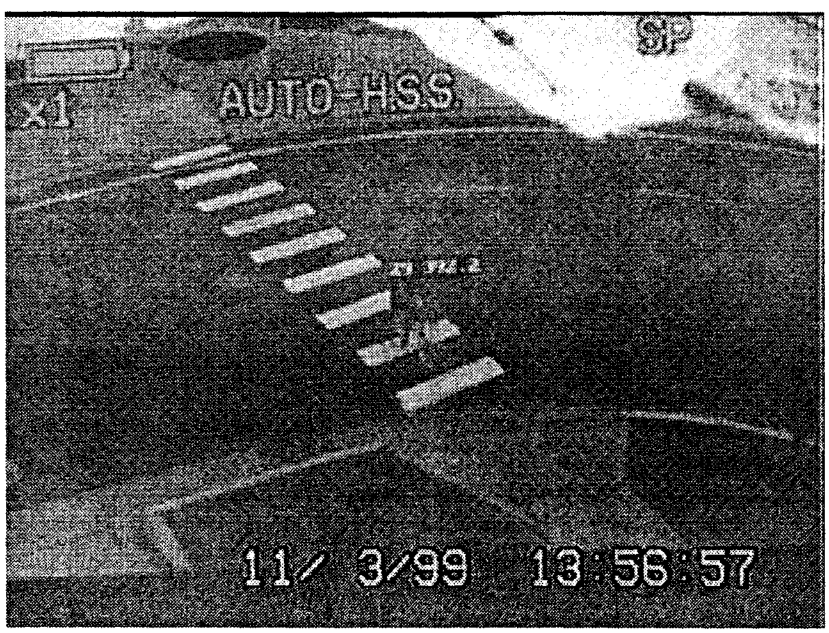


Figure 6.6 Another instance of pedestrian tracking. The box around the pedestrian illustrates relevant pedestrian information (position, velocity, etc.).

6.3 Conclusion

We demonstrated our approach to a real intersection with very promising results. The next step is to have a long term evaluation in order to test the approach in various traffic and weather patterns. There is also a need to detect and handle shadows effectively. Another leasson learnt is the importance of camera placement. It is significant for the camera to view the whole pedestrian crossing.

Chapter 7

Conclusions

7.1 Summary

In this report, we presented solutions to the human detection and tracking problem. In particular, we looked at the human tracking problem as tracking of articulated motion. We tracked the body as a whole without identifying individual limb motion. Our goal was to address certain shortcomings in previous solutions to this problem, the main shortcoming behind their over-constrained nature. Some of the constraints are: restriction to indoor environments, little or no occlusions, fixed lighting and weather conditions, limits on the number of tracked objects, restriction on camera placement, using stereo cameras, and constrained motion. Our solution which was presented as a real-time pedestrian tracking system is capable of working under many difficult circumstances. The method relies on an underlying robust tracking of blobs. The tracked blobs are used to instantiate a pedestrian model. The model has both a spatial as well as temporal specifications which, although simple, were found adequate in modeling the pedestrian shape and motion. Extensive experimentation using long video sequences demonstrated the robustness of our system and its generality. The system was also adapts well to tracking rigid motion. This was demonstrated by an application to track vehicles across lanes at a weaving section.

The second problem is recognition of articulated motion (in order to identify intentions of pedestrians). The goal here was to show that the recovery of three-dimensional properties of the object or even two-dimensional tracking of the object parts are not necessary steps that must precede action recognition. The approach we devised uses motion features only. Unlike other similar approaches, the motion features are used in such a way to represent complex and long actions as well as to distinguish different actions with many similarities. Motion features are computed very efficiently and subsequently projected into a lower dimension space where the matching is performed. Each action is represented as a manifold in the lower dimension space and matching is done by comparing these manifolds. To demonstrate the effectiveness of this approach, it was used on a large data set of similar actions each performed by many different actors. Classification results were very accurate in comparison to reported action classification results. The results show that this approach can handle many challenges such as variations is performers physical attributes, color of clothing, and stylistic attributes. Moreover, we have shown that the method is efficient and suitable for real-time implementation.

7.2 Future Work

7.2.1 Tracking

There are several issues that still need to be addressed. Spatial interpretation of blobs is one such issue. In the current system, the only spatial attribute of blobs taken into consideration is the blob area. The shape of the blob can give a good clue on its contents. Although blobs obtained from difference images can be sufficient to decide the location of pedestrians in many cases, the intensity information may be useful to resolve certain ambiguities. The use of such information in the form of statistical distribution of intensities may add to the robustness of the current system and is well worth pursuing.

Although the system is robust in many circumstances, large shadows can get tracked as pedestrians. The problem of shadows is challenging since they cannot be easily distinguished from other moving objects. Detecting shadows is one area of possible improvement. In the past, this has been attempted by considering statistical shadow properties or by using prior knowledge about the location of the light source.

In its current state, our method assumes that all the objects in the scene are pedestrians. This means that if another object (such as a vehicle) comes into the scene, it will be tracked as a pedestrian (or a group of pedestrians). The problem is currently handled by limiting pedestrian tracking to areas to areas where there are no other large moving objects. Action recognition can be useful in this situation to classify moving object as pedestrians based on the action being performed (most likely walking or running).

7.2.2 Real-time Implementation of Action Recognition

We have shown that the proposed method is efficient and appropriate for real-time implementation. This can provide a convenient way of further testing as well as training for other actions. A real-time implementation can also be used in conjunction with the pedestrian tracking system in a real world surveillance application.

7.2.3 Recognition from a General View Angle

Our experiments considered actions performed parallel to the camera plane. This is an undesired restriction in some applications. Recognition performance has not been tested for other viewing angles. Although size scaling should compensate for shrinking in width, we expect it to handle only small angle variations since the motion features will change dramatically for larger angles. This has been a problem for most action recognition methods which are not based on three-dimensional tracking. It is normally handled by considering actions performed at different viewing angles as separate actions, thus increasing the size of the learned action database. This is still an option in our case which is worth trying.

Bibliography

[1] K. Akita, "Image sequence analysis of real world human motion," *Pattern Recognition*, vol. 17, no. 1, pp. 73-83, 1984.

[2] A. Azarbayejani and A. Pentland, "Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features," in *Proc. of International Conference on Pattern Recognition*, Vienna, 1996.

[3] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic Press, 1988.

[4] A. Baumberg and D. Hogg, "Learning flexible models from image sequences," in *Proc. of European Conference on Computer Vision*, vol. 1, pp. 229-308, May 1994.

[5] A. Baumberg and D. Hogg, "An efficient method for contour tracking using active shape models," in *Proc. of IEEE Workshop on Motion of Non-rigid and Articulated Objects*, pp. 194-199, IEEE Computer Society Press, Nov. 1994.

[6] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," IEEE Transactions on Pattern Recognition and Machine Intelligence, vol. 19, no. 7, pp. 771-720, 1997.

[7] A. Bobick, J. Davis, S. Intille, F. Baid, L. Campbell, Y. Ivanov, C. Pinhanez, A. Schutte, and A. Wilson, "KIDSROOM: Action recognition in an interactive story environment," MIT Media Lab Perceptual Computing Group Technical Report No. 398, MIT, December 1996.

[8] A. Bobick and A. Wilson, "A state-based technique for the summarization and recognition of gesture," in *Proc. of International Conference on Computer Vision*, pp. 382-388, Cambridge, 1995.

[9] Q. Cai and J. K. Aggarwal, "Tracking human motion using multiple cameras," in *Proc. of the 13th International Conference on Pattern Recognition*, pp. 68-72, 1996.

[10] L. Campbell and A. Bobick, "Recognition of human body motion using phase space constraints," in *Proc. of International Conference on Computer Vision*, pp. 624-630, Cambridge, 1995.

[11] I.-C. Chang and C.-L. Huang, "Ribbon-based motion analysis of human body movements,"

in *Proc. of International Conference on Pattern Recognition*, pp. 436-440, Vienna, 1996.

[12] C. Charayaplan and A. Marble, "Image processing system for interpreting motion in American Sign Language," *Journal of Biomedical Engineering*, vol. 14, no. 15, pp. 419-425, 1992.

[13] Z. Chen and H. J. Lee, "Knowledge–guided visual perception of 3–d human gait from a single image sequence," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 22, no. 2, pp. 336-342, 1992.

[14] Cohen, L. and Note, D.: "On Active Contour Models and Balloons," *CVGIP (image understanding)*, Vol 53, #2, pp211-218, 1991

[15] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models - their training and applications," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38-59, 1995.

[16] Couvignou, P., Papanikolopoulos, N., Sullivan, M., and Khosla, P., "The use of active deformable models in model-based robotic visual servoing," *Journal of Intelligent and Robotic Systems: Theory and Applications*, 17(2):195-221, 1996.

[17] J. E. Cutting and L. T. Kozlowski, "Recognizing friends by their walk: Gait perception without familiarity cues," *Bull. Psychonometric Soc.*, vol. 9, no. 5, pp. 353-356, 1977.

[18] T. Darrel and A. Pentland, "Space-time gestures," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 335-340, New York, 1993.

[19] J. W. Davis and A. F. Bobick, "The representation and recognition of human movement using temporal templates," in *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 928-934, 1997.

[20] J. Davis and M. Shah, "Gesture recognition," Technical Report CS-TR-93-11, University of Central Florida, 1993.

[21] W. H. Dittrich, "Action categories and the perception of biological motion," Perception, vol. 22, pp. 15-22, 1993.

[22] B. Dorner, "Hand shape identification and tracking for sign language interpretation," in *Proc. of Looking at People, International Joint Conference on Artificial Intelligence*, Chambery, 1993.

[23] A. Downton and H. Drouet, "Model-based image analysis for unconstrained human upper-body motion," in *Proc. IEE International Conference on Image Processing and its Applications*, pp.274-277, 1992.

[24] K. E. Finn and A. A. Montgomery, "Automatic optically-based recognition of speech," Pattern Recognition Letters, vol. 8, pp. 159-164, 1988.

[25] W. Freeman, K. Tanaka, J. Ohta, and K. Kyuma, "Computer vision for computer games," in *Proc. of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 100-105, Killington, 1996.

[26] D. M. Gavrila and L. S. Davis, "3-D model-based tracking of humans in action: a multi-view approach," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 73-80, San Francisco, 1996.

[27] N. Goddard, "Incremental model-based discrimination of articulated movement direct from motion features," in *Proc. of IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 89-94, Austin, 1994.

[28] L. Goncalves, E. Di Benardo, E. Ursella, and P. Perona, "Monocular tracking of the human arm in 3-D," in *Proc. of International Conference on Computer Vision*, pp. 764-770, Cambridge, 1995.

[29] Y. Guo, G. Xu, and S. Tsuji, "Understanding human motion patterns," in *Proc. of the 12th IAPR International Conference on Pattern Recognition*, pp. 325-329, 1994.

[30] G. Halevi and D. Weinshall, "Motion of disturbances: detection and tracking of multi-body non-rigid motion," in *Proc. of IEEE Conference Computer Vision and Pattern Recognition*, pp. 897-902, June 1997, Puerto Rico.

[31] P. Hallinan, "A low-dimensional representation of human faces for arbitrary lighting conditions," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 995-999, 1994.

[32] T. Heap and D. Hogg, "Towards 3-D hand tracking using a deformable model," in *Proc. of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 140-145, Killington, 1996.

[33] B. Heisele, U. Kressel, and W. Ritter, "Tracking non-rigid, moving objects based on color cluster flow," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*,

pp. 257-260, San Juan, 1997.

[34] B. Heisele and C. Wohler, "Motion-based recognition of pedestrians," in *Proc. of the Fourteenth International Conference on Pattern Recognition*, vol. 2, pp. 1325-1330, Brisbane, Australia, Aug. 1998.

[35] D. Hoffman and B. Flinchbaugh, "The interpretation of biological motion," *Biological Cybernetics*, vol. 42, pp. 195-204, 1982.

[36] D. Hogg, "Model based vision: A paradigm to see a walking person," *Image and Vision Computing*, vol. 1, no. 1, pp. 5-20, 1983.

[37] R. Holt, A. Netravali, T. Huang, and R. Qian, "Determining articulated motion from perspective views: A decomposition approach," in *Proc. of IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp.126-137, Austin, 1994.

[38] R. Hosie, S. Venkatesh, and G. West, "Detecting deviations from known paths and speeds in a surveillance situation," in *Proc. of the Fourth International Conference on Control, Automation, Robotics and Vision*, pp. 3-6, Singapore, Dec 1996.

[39] E. Hunter, J. Schlenzig, and R. Jain, "Posture estimation in reduced-model gesture input systems," in *Proc. of International Workshop on Automatic Face and Gesture Recognition*, pp. 290-295, Zurich, 1995.

[40] Ivins, J. and Porrill, J., "Active region models for segmenting medical images," *Proceedings of the IEEE International Conference on Image Processing*, 227-231, 1994.

[41] R. Jain, D. Militzer, and H.H. Nagel, "Separating non-stationary from stationary scene components in a sequence of real-world TV images," in *Proc. of the Int. Joint Conf. on Artificial Intelligence,* pp. 612-618, 1977.

[42] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception and Psychophysics*, vol. 14, no. 2, pp. 201-211, June 1973.

[43] G. Johansson, "Visual motion perception," *Sci. Amer.*, vol. 232, pp. 75-88, June 1976.

[44] N. Johnson, and D. Hogg, "Learning the distribution of object trajectories for event recognition," *Image and Vision Computing*, vol. 14, no. 8, pp. 609-615, Aug. 1996.

[45] S. Ju, M. Black, and Y. Yacoob, "Cardboard people: A parameterized model of articulated image motion," in *Proc. of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 38-44, Killington, 1996.

[46] I. A. Kakadiaris and D. Metaxas, "3D human body model acquisition from multiple views," in *Proc. of the Fifth International Conference on Computer Vision*, pp. 618-623, Boston, MA, Jun. 1995.

[47] T. Kanade, "Region segmentation: signal vs. semantics," *Computer Graphics and Image Processing*, vol. 13, no. 4, pp. 279-297, August 1980.

[48] Kass, M., Witkin, A., and Terzopoulos, D., "Snakes: active contour models," *Proceedings of the First International Conference on Computer Vision*, 259-268, 1987.

[49] M. Kirby, F. Weisser, and G. Dangelmayr, "A model problem in the representation of digital image sequences," *Pattern Recognition*, vol. 26, no. 1, pp. 63-73, 1993.

[50] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve procedure for the characterization of human faces," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 1, pp. 103-108, 1990.

[51] L. T. Kozlowski and J. E. Cutting, "Recognizing the sex of a walker from dynamic point–light displays," *Perception and Psychophysics*, vol. 21, no. 6, pp. 575-580, 1977.

[52] J. Kuch and T. Huang, "Vision-based hand modeling and tracking for virtual teleconferencing and telecollaboration," in *Proc. of International Conference on Computer Vision*, pp. 666-671, Cambridge, 1995.

[53] M. Leung and Y. Yang, "First sight: a human body outline labeling system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 4, pp. 359-377, 1995.

[54] G. A. Martin and M. Shah, "Lipreading using optical flow," in *Proc. National Conference on Undergraduate Research*, March 1992.t

[55] K. Mase and A. Pentland, "Lip reading: automatic visual recognition of spoken words," Technical Report 117, M.I.T. Media Lab Vision Science, 1989.

[56] C. Myers, L. Rabinier, and A. Rosenberg, "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition," IEEE Transactions on ASSP, vol. 28, no. 6, pp. 623-635, 1980.

[57] S. K. Nayar, S. A. Nene, and H. Murase, "Real-time 100 object recognition system," in *Proc. of IEEE Conference on Robotics and Automation*, vol. 3, pp. 2321-2325, Minneapolis, 1996.

[58] S. Niyogi and E. Adelson, "Analyzing and recognizing walking figures in XYT," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp.469-474, 1994.

[59] J. Ohya and F. Kishino, "Human posture estimation from multiple images using genetic algorithm," in *Proc. of International Conference on Pattern Recognition*, pp. 750-753 (A), 1994.

[60] J. O'Rourke and N. I. Badler, "Model–based image analysis of human motion using constraint propagation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 2, no. 6, pp. 522-536, November 1980.

[61] J.R. Parker, "Gray level thresholding in badly illuminated images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, pp. 813-819, 1991.

[62] F. Perales and J. Torres, "A system for human motion matching between synthetic and real images based on a biomechanic graphical model," in *Proc. of IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 83-88, Austin, 1994.

[63] E. D. Petajan, B. Bischoff, D. Bodoff, and N. M. Brooke, "An improved automatic lipreading system to enhance speech recognition," in *SIGCHI '88: Human Factors in Computing Systems*, pp. 19-25, October 1988.

[64] R. Polana and R. Nelson, "Detecting activities," *Journal of Visual Communication and Image Representation*, vol. 5, no. 2, pp. 172-180, 1994.

[65] R. Polana and R. Nelson, "Detection and recognition of periodic, nonrigid motion," *International Journal of Computer Vision*, vol. 23, no. 3, pp. 261-282, 1997.

[66] F. Quek, "Eyes in the interface," Image and Vision Computing, vol. 13, no. 6, pp. 511-525, 1995.

[67] K. Rangarajan, W. Allen, and M. Shah, "Matching motion trajectories using scale space," *Pattern Recognition*, vol. 26, no. 4, pp. 595-610, 1993.

[68] J. Rehg and T. Kanade, "Visual tracking of high DOF articulated structures: an application to human hand tracking," in *Proc. of European Conference on Computer Vision*, pp. 34-46,

Stockholm, 1994.

[69] K. Rohr, "Towards model-based recognition of human movements in image sequences," *CVGIP: Image Understanding*, vol. 59, pp. 94-115, Jan. 1994.

[70] M. Rossi and A. Bozzoli, "Tracking and counting moving people," in *Proc. of Second IEEE International Conference on Image Processing*, pp. 212-216, 1994.

[71] P.K. Sahoo, S. Soltani and A.K.C. Wong, "A survey of thresholding techniques," *Computer Vision Graphics Image Processing*, vol. 41, pp. 233-260, 1988.

[72] J. Segen and S. Pingali, "A camera-based system for tracking people in real time," in *Proc. of the 13th International Conference on Pattern Recognition*, pp. 63-67, 1996.

[73] T. Shakunaga, "Pose estimation of jointed structures," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 566-572, 1991.

[74] A. Shio and J. Sklansky, "Segmentation of people in motion," in *Proc. of IEEE Workshop on Visual Motion*, pp. 325-332, 1991.

[75] K. Skiestad and R. Jain, "Illumination independent change detection for real world image sequences," *Computer Vision Graphics and Image Processing,* vol. 46, pp. 387-399, 1989.

[76] C. Smith, C. Richards, S. A. Brandt, and N. P. Papanikolopoulos, "Visual tracking for intelligent vehicle-highway systems," *IEEE Trans. on Vehicular Technology*, vol. 45, no. 4, pp. 744-759, Nov. 1996.

[77] T. Starner and A. Pentland, "Real-time American Sign Language recognition from video using hidden Markov models," in *International Symposium on Computer Vision*, pp. 265-270, Coral Gables, 1995.

[78] Sullivan, M. and Papanikolopoulos, N., "Using active-deformable models to track deformable objects in robotic visual servoing experiments," *Proceedings of the IEEE International Conference on Robotics and Automation*, 2929-2934, 1996.

[79] M. J. Sullivan, C. A. Richards, C. E. Smith, O. Masoud, and N. P. Papanikolopoulos, "Pedestrian tracking from a stationary camera using active deformable models," in *Proc. of the Intelligent Vehicles '95*, pp. 90-95, September 1995.

[80] D. L. Swets and J. Weng, "Using discriminant eigenfeatures for image retrieval," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 18, no. 8, pp. 831-836, 1996.

[81] S. Tamura and S. Kawasaki, "Recognition of sign language motion images," Pattern Recognition, vol. 21, no. 4, pp. 343-353, 1988.

[82] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 13, no. 1, pp. 71-86, 1991.

[83] J. Wang, G. Lorette, and P. Bouthemy, "Analysis of human motion: A model-based approach," in *Proc. 7th Scandinavian Conference on Image Analysis*, Aalborg, 1991.

[84] J. A. Webb and J. K. Aggarwal, "Structure from motion of rigid and jointed objects. Artificial Intelligence," vol. 19, pp. 107-130, 1982.

[85] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: real-time tracking of the human body," in *Proc. of the Second International Conference on Automatic Face and Gesture Recognition*, pp. 51-56, Oct 1996.

[86] Y. Yacoob and M. J. Black, "Parameterized modeling and recognition of activities," *Journal of Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 232-247.

[87] M. Yamamoto and K. Koshikawa, "Human motion analysis based on a robot arm model," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 664-665, Maui, 1991.

[88] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time sequential images using Hidden Markov Model," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 379-385, 1992.

[89] J. Zhao, *Moving Posture Reconstruction from Perspective Projections of Jointed Figure Motion*, Ph.D. thesis, University of Pennsylvania, 1993.