2007-12-14

Human Activity Recognition and Pathological Gait Pattern Identification

Feng Niu
University of Miami, feng.niu@gmail.com

UNIVERSITY OF MIAMI

HUMAN ACTIVITY RECOGNITION AND PATHOLOGICAL GAIT PATTERN IDENTIFICATION

By

Feng Niu

A DISSERTATION

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Coral Gables, Florida

December 2007

UNIVERSITY OF MIAMI

A dissertation submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

HUMAN ACTIVITY RECOGNITION AND PATHOLOGICAL GAIT PATTERN IDENTIFICATION

Feng Niu

Approved:

____
Dr. Mohamed Abdel-Mottaleb
Associate Professor of Electrical
and Computer Engineering

____
Dr. Terri A. Scandura
Dean of Graduate School

____
Dr. Kamal Premaratne
Professor of Electrical and
Computer Engineering

____
Dr. Mei-Ling Shyu
Associate Professor of Electrical
and Computer Engineering

____
Dr. Shihab Asfour
Professor of Industrial Engineering

____
Dr. Ajay Divakaran
Research Staff of Mitsubishi Electric
Research Laboratories

NIU, FENG (Ph.D., Electrical and Computer Engineering)
Human Activity Recognition (December 2007)
and Pathological Gait Pattern Identification

Abstract of a dissertation at the University of Miami.

Dissertation supervised by Dr. Mohamed Abdel-Mottaleb.
No. of pages in text. (117)

Human activity analysis has attracted great interest from computer vision researchers due to its promising applications in many areas such as automated visual surveillance, computer-human interactions, and motion-based identification and diagnosis.

This dissertation presents work in two areas: general human activity recognition from video, and human activity analysis for the purpose of identifying pathological gait from both 3D captured data and from video.

Even though the research in human activity recognition has been going on for many years, still there are many issues that need more research. This includes the effective representation and modeling of human activities and the segmentation of sequences of continuous activities. In this thesis we present an algorithm that combines shape and motion features to represent human activities. In order to handle the activity recognition from any viewing angle we quantize the viewing direction and build a set of Hidden Markov Models (HMMs), where each model represents the activity from a given view. Finally, a voting based algorithm is used to segment and recognize a sequence of human activities from video. Our method of representing activities has good attributes and is suitable for both low resolution and high resolution video. The voting based algorithm performs the segmentation and recognition simultaneously. Experiments on two sets of video clips of different activities show that our method is effective.

Our work on identifying pathological gait is based on the assumption of gait symmetry. Previous work on gait analysis measures the symmetry of gait based on Ground Reaction Force data, stance time, swing time or step length. Since the trajectories of the body parts contain information about the whole body movement, we measure the symmetry of the gait based on the trajectories of the body parts. Two algorithms, which can work with different data sources, are presented. The first algorithm works on 3D motion-captured data and the second works on video data. Both algorithms use support vector machine (SVM) for classification. Each of the two methods has three steps: the first step is data preparation, i.e., obtaining the trajectories of the body parts; the second step is gait representation based on a measure of gait symmetry; and the last step is SVM based classification. For 3D motion-captured data, a set of features based on Discrete Fourier Transform (DFT) is used to represent the gait. We demonstrate the accuracy of the classification by a set of experiments that shows that the method for 3D motion-captured data is highly effective. For video data, a model based tracking algorithm for human body parts is developed for preparing the data. Then, a symmetry measure that works on the sequence of 2D data, i.e. sequence of video frames, is derived to represent the gait. We performed experiments on both 2D projected data and real video data to examine this algorithm. The experimental results on 2D projected data showed that the presented algorithm is promising for identifying pathological gait from video. The experimental results on the real video data are not good as the results on 2D projected data. We believe that better results could be obtained if the accuracy of the tracking algorithm is improved.

To my beloved mother, sister, and deceased father

ACKNOWLEDGMENTS

First, I would like to express my sincere gratitude to Professor Abdel-Mottaleb, who introduced me to the field of computer vision, and helped me to jump start my research career in this field. He encouraged me to pursue novel ideas and provided me with exceptional experience and knowledge.

My thanks also go to the other members of the committee, Dr. Kamal Premaratne, Dr. Mei-Ling Shyu, Dr. Shihab Asfour and Dr. Ajay Divakaran, for their valuable comments and suggestions. In particular, I want to thank Dr. Shihab Asfour for his helpful supports in the last two year of my Ph.D study and thank Dr. Ajay for his valuable supports and encouragement in the past years.

More than everyone, I indebted to my mother and older sister for their enthusiastic encouragement and unlimited supports during all stages of my life. I would like to extend my thanks to the faculty and staff members of the Electrical and Computer Engineering Department at the University of Miami for their role in my education and various resources made available to me to do my research.

In addition, I would like to thank my fellow lab members, especially Dr. Nasser Al-Ansari, Dr. Omaima Nomir, Mohammad Hossein Mahoor, Jindan Zhou, Steven Cadavic, Ali Taatian, Muhammad Rushdi, Zhouyi Xu and for their collaboration during the completion of this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Automatic visual analysis of human motion from video has been one of the most active research areas in computer vision. It usually includes detecting people, tracking, and more generally analyzing and interpreting human behaviors from image sequences. Human motion analysis has attracted great interest from computer vision researchers due to its promising applications in many areas such as automated visual surveillance, computer-human interactions, athletic performance analysis, content-based image indexing and retrieval, virtual reality, etc.

Human motion analysis has been investigated under several large research projects worldwide. For example, Video Surveillance and Monitoring (VSAM) [1], a multi-institution project funded by Defense Advanced Research Projects Agency (DARPA), tried to develop an automatic video understanding technology that enables a single human operator to monitor activities over complex areas such as battlefields and civilian scenes.

Context Aware Vision using Image-based Active Recognition (CAVIAR), funded by the EC's Information Society Technology, tries to develop the theory of context-aware visual recognition systems and build a vision system for two applications (city street surveillance and customer behavior analysis). The techniques they are interested in include integrating features, representing and recognizing objects, contexts and situations, learning instances of the representations from visual evidence, etc. In the UK, researchers have also worked on tracking vehicles and people and recognizing their interactions [2]. In addition, companies such as IBM and Microsoft also invested in research on human motion analysis [3,4].

In recent years, human motion analysis has been featured in a number of leading international journals such as CVIU (Computer Vision and Image Understanding), PAMI (IEEE Transactions on Pattern Recognition and Machine Intelligence), as well as prestigious international conferences and workshops such as ICCV (International Conference on Computer Vision), CVPR (IEEE International Conference on Computer Vision and Pattern Recognition), IWVS (IEEE International Workshop on Visual Surveillance), ICME (IEEE International Conference on Multimedia & Expo).

All the above activities demonstrate a great and growing interest in human motion analysis from the pattern recognition and computer vision community.

1.1 Potential applications

Human motion analysis has a wide range of potential applications such as visual surveillance, computer user interface, motion based diagnosis, etc.

1.1.1 Visual surveillance

Many security-sensitive areas, such as banks, department stores, parking lots, and borders, have strong need for automated surveillance systems [5, 6]. At present, surveillance cameras are already widely used in commercial establishments, while camera outputs are usually recorded in tapes or stored in some video archives. These video data are currently used only “after the fact” as a forensic tool. What is needed is a real-time analysis of surveillance data to alert security officers to a burglary in progress, or to a suspicious individual wandering around in the parking lot. Nowadays, the tracking and recognition techniques of gait [7-10] have been strongly motivated by access control applications. As well as the obvious security applications, smart surveillance has also been proposed to measure traffic flow, monitor pedestrian congestion in public spaces [11, 12], compile consumer demographics in shopping malls, etc.

1.1.2 Computer user interface

Another important application domain is advanced user interfaces in which human motion analysis is usually used to provide control and command. Generally speaking, communication among people is mainly realized by speech, so speech understanding has already been widely used in early human-machine interfaces. However, it is subject to the restrictions from environmental noise and distance. Vision is very useful to complement speech recognition and natural language understanding for more natural and intelligent communication between humans and machines. That is, some cues obtained from human behavior, such as gestures and body pose, provide useful and complementary information for a machine to understand human commands and needs [13, 14]. Hence, some abilities, such as detecting human presence and interpreting human behavior, are needed by future machines. Other applications in the user interface domain include sign-language translation, gesture driven controls, and signaling in high-noise environments such as factories and airports [15].

1.1.3 Motion based identification and diagnosis

With the development of computer-aided diagnostics, human motion analysis has attracted much attention of researchers in the medical field. There is some gait analysis based research work aimed at providing medical diagnosis and treatment support, such as [15-18]. Human gait also have been used to identify the gender, or to identify older people who usually have more potential risk of falling. Traditionally, the source data used for motion analysis are captured by some special instruments, such as ground reaction force platform and sensors attached to the human body, etc. Recently, video data based motion analysis are also researched for these kinds of applications. Particularly useful techniques include segmenting human body parts in an image, tracking the movement of joints over an image sequence, and recovering the underlying 3-D body structure for analysis.

1.1.4 Other related areas

In addition, human motion analysis shows its importance in other related areas. For instance, typical applications in virtual reality include chat-rooms, games, virtual studios, etc. As far as computer games [19] are concerned, they have been very prevalent in entertainment. Sometimes, people are surprised at the realism of virtual humans and simulated actions in computer games. In fact, this benefits greatly from computer graphics dealing with devising realistic models of human bodies and the synthesis of human movement based on knowledge of the acquisition of the human body model, the retrieval of body pose, human behavior analysis, etc.

Besides, human motion analysis also benefits model-based image coding (e.g., only encoding the pose of a tracked face in images in more detail than the uninteresting background in a videophone setting) which will bring about very low bit-rate video compression for more effective image storage and transmission.

1.2 Limitations and challenges of Human activity recognition

Automating the process of activity recognition is very challenging. Although there has been some good work in the area of gesture recognition and sign language recognition, automatically recognizing human activities is more challenging than gesture and sign language recognition. In the case of gesture and sign languages, there is a rigid syntax and predefined structure. However, in case of human activities there are no predefined vocabularies and no well-defined structures. Following are some of the issues that still need further research.

1.2.1 Segmentation

Fast and accurate motion segmentation is a significant but difficult problem. The captured images in dynamic environments are often affected by many factors such as weather, lighting, clutter, shadow, occlusion, and even camera motion. Taking only shadow for an example, it may either be connected with the detected object or disconnected from it. In the first case, the shadow distorts the object's shape, making the use of subsequent shape recognition methods less reliable. In the second case, the shadow may be classified as a totally erroneous object in the natural scene.

Nearly every system for human motion analysis starts with segmentation, so segmentation is of fundamental importance. Although current motion segmentation methods mainly focus on background subtraction, how to develop more reliable background models that are adaptive to the dynamic changes in complex environments is still a challenge.

1.2.2 Handling occlusion

At present, the majority of human motion analysis systems cannot effectively handle the problems of self-occlusion of the human body and mutual occlusions between objects, especially the detection and tracking of multiple people under congested conditions. Typically, during occlusions, only portions of each person are visible and often at very low resolution. This problem is generally intractable, and motion segmentation based on background subtraction may become unreliable. To reduce ambiguities due to occlusion, better models need be developed to cope with the correspondence problem between features and body parts. Interesting progress is being made using statistical methods [20], which essentially try to predict body pose, position, and so on, from available image information. Perhaps the most promising practical method for addressing occlusions is through the use of multiple cameras.

1.2.3 3-D modeling and tracking

2-D approaches have shown some successes in visual analysis of human motion, especially for applications that do not require high resolution (e.g., pedestrian tracking in a traffic surveillance setting). The major drawback of 2-D approaches involves the constraints on the camera viewing angles. Compared with 2-D approaches, for high resolution applications, 3-D approaches are more effective for accurate estimation in physical space, handling occlusion, and the high-level judgment between various complex human movements such as wandering around, shaking hands and dancing [21-23]. However, applying 3-D tracking requires more parameters and more computations during the matching process. In general, current research on 3-D tracking is still in its infancy. Also, vision-based 3-D tracking brings a number of challenges such as the acquisition of human models [24], handling occlusion, parameter based body modeling [25-27], etc. So 3-D modeling and tracking deserve more attention in future work.

1.2.4 Use of multiple cameras

It is obvious that future systems of human motion analysis will greatly benefit from the use of multiple cameras. The availability of information from multiple cameras can be extremely helpful because the use of multiple cameras not only expands the surveillance area, but also provides multiple viewpoints to solve occlusions effectively. Tracking with a single camera easily generates ambiguity due to occlusion or depth. However, this may be resolved by information from another view.

For multi-camera tracking systems, it is important to decide which camera or image to use at each time instant. That is, the coordination and information fusion between cameras are a significant problem.

1.2.5 Action understanding

Since the final objective of "looking at people" is to analyze and interpret human action and the interactions between people and other objects, better understanding of human behavior is the most interesting long-term open issue facing human motion analysis. For instance, the W^4 system [28] can recognize some simple events between people and objects such as carrying an object, depositing an object, and exchanging bags. However, human motion understanding still stresses tracking and recognition of some standard posture, and simple action analysis, e.g., the definition and classification of a group of typical actions (running, standing, jumping, climbing, pointing, etc). Some recent progress has been made in building the statistical models of human behaviors by using machine learning, but action recognition is just in its infancy. Some restrictions are usually imposed to decrease ambiguity during matching of feature sequences. Therefore, the difficulties of behavior understanding still lie in feature selection and machine learning. Nowadays, the approaches of state space and template matching for action recognition often choose a trade-off between computational cost and recognition accuracy, so efforts should be made to improve performance of behavior recognition. Furthermore, we should develop algorithms to extend current simple action recognition to more complex activity recognition.

1.2.6 Performance evaluation

Generally speaking, robustness, accuracy and speed are three major demands of practical human motion analysis systems [29]. For example, robustness is very important for surveillance applications that are required to work automatically and continuously. These systems should be insensitive to noise, lighting, weather, clothes, etc. It may be expected that the fewer assumptions a system imposes on its operational conditions, the better. The accuracy of a system is important for behavior recognition in surveillance or control situations. The processing speed of a system deserves more attention, especially for real time surveillance applications.

It is important to test the robustness of any system on large amount of data, a number of different users, and in various environments. Furthermore, it is an interesting direction to find more effective ideas for real-time and accurate online processing. It seems to be helpful and necessary to incorporate various data types and processing methods to improve robustness of a human motion analysis system to all possible situations.

1.3 Human activity analysis

In our work, we will address the issues of recognizing complex activities that consist of sequences of simple sub-activities. We will also address the pathological gait pattern identification.

1.3.1 General human activity recognition

Several human activity recognition methods have been proposed in the past few years. Most of them can be classified into two classes based on the features that they use for recognizing human activities from video: motion feature based methods and shape feature based methods.

Both motion-based and shape-based features have their own limitations. Motion-based features can depict the approximate direction of the motion of the body, but most motion-based features are not robust in capturing velocity. For example, motion-based features can easily discriminate walking and sitting down, but fail to discriminate between walking and slow running. On the other hand, shape-based features can capture some pose information of the body, but without motion information its capability of describing human activity is limited. Therefore, combining both features can improve the representation and the robustness of activity recognition.

Most of the work in activity recognition is view dependent and deals with recognition from one fixed view. The task of recognizing human activities from different views remains unsolved. Although some algorithms recognize human activities from different views, e.g., [30] [31], these methods either need to track different body parts from high resolution video or build 3D human model. In this thesis, we combine motion-based features with shape-based features to model human activities. We represent each activity by a set of Hidden Markov Models, where each model represents the activity viewed from a specific direction (i.e., viewing angle) to realize the view-invariance. Also, we present a voting based method to segment and recognize continuous complex human activities.

1.3.2 Pathological gait pattern identification

Pathological gait describes altered gait patterns that have been affected by deformity, muscle weakness, impaired motor control and pain. Analysis of pathological gait is very important for both the medical diagnosis and the treatment process. In order to perform a diagnostic function it is necessary to be able to distinguish pathological from normal patterns of movement. Some deviations from normal gait patterns are obvious and can be identified easily, but others have to be identified by trained doctors. Even in some cases, trained doctors have to identify those deviations with special instruments. An automatic system for identifying pathological gait can be very valuable for both clinical diagnosis and treatment. The system can be used to quantify the deviation of the gait from the normal gait and quantify the progress due to certain treatment.

For pathological gait pattern identification, it is intuitive to assess a patient's gait by measuring the symmetry of the gait. In normal individuals gait patterns with respect to time, distance and vertical force are fairly symmetrical and only deviate by a small percentage from perfect symmetry [32]. Most of the developed symmetry measures, such as those presented in [33-37], are based on stance time, swing time, step length and vertical ground reaction force. Also, most of the methods just focus on the information provided by the lower limb movement. The symmetry of the upper body movement is rarely examined. The upper body movement may be informative since it also provides information of body movement which is affected by patient’s disease. Therefore, a new gait symmetry measure based on the movement of the whole body may provide better assessment of the gait normality.

We present two algorithms for identifying pathological gait. The first algorithm uses 3D motion data. In this algorithm, a method based on Discrete Fourier Transform (DFT) is presented to measure the symmetry of two 3D trajectories. Using this method a feature vector that captures the symmetry of the whole body’s movement is obtained. Support vector machines were used for classification and the results show that the algorithm is effective.

The 3D motion-capture systems and devices that are used to capture the gait data for patients are expensive. Moreover, it usually takes much time and work to capture data (the patients are asked to take off their outerwear. Sensors and devices have to be carefully attached on patients’ bodies). Since common video cameras are inexpensive, it is advantageous to use data captured by these cameras to achieve the same goal. In this thesis, we also present a method to identify pathological gait from video data captured from a profile view. The experimental results are encouraging.

Both algorithms use a support vector machine (SVM) based classification framework as shown in figure1.1.

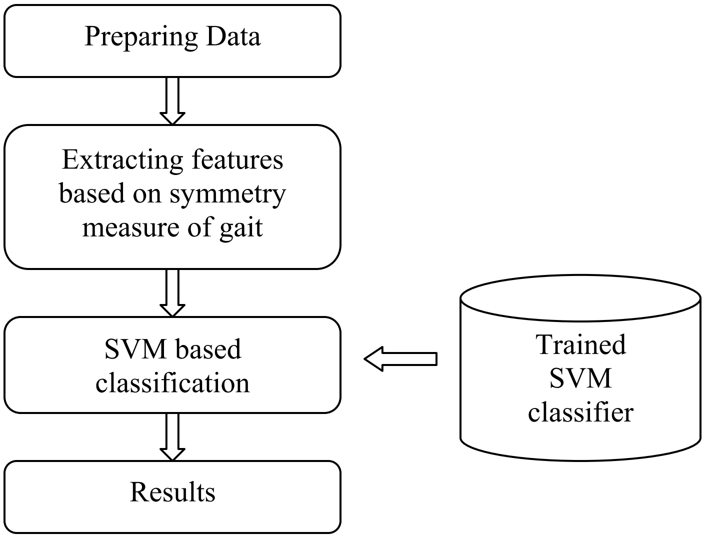


Figure 1.1 SVM based pathological gait classification framework

1.4 Contributions

The major contributions of this dissertation are as follows:

- Presenting a shape representation method based on PCA. Even though we use it to represent the silhouette of a person in our work, it could be used to describe the shape of any object in other applications.

- Developing a method that uses both shape and motion information for representing human activities. As shown later in the thesis both shape features and motion features have their own advantages and disadvantage and when used together they complement to each other. Besides, this representation is general and can be extracted from any view and is suitable for both high resolution and low resolution video.

- Developing a view independent and HMM model based method for activity recognition.

- Developing a voting scheme that works with the HMMs for segmenting and recognizing sequences of activities. In the training stage, the algorithm is only trained on single activity samples. In the recognition stage, the algorithm simultaneously segments and recognizes the activities.

- Presenting a gait symmetry measurement and SVM based framework for identifying pathological gait.

- Developing an algorithm for identifying pathological gait using 3D motion captured data. In this dissertation, we develop a DFT based gait symmetry measure using 3D trajectories of the human body parts and use it to represent the human gait for pathological gait identification.

- Presenting a 2D template-based method for tracking body parts from profile view video.

- Developing an algorithm for identifying pathological gait from video. Measuring the symmetry of two 3D trajectories based on their 2D projections is not easy. In this dissertation, we develop a symmetry measure of two 3D trajectories based on their projections on a 2D plane and use it for pathological gait identification.

1.5 Dissertation outline

In chapter 2, we review the related research work for motion segmentation, behavior understanding, and action recognition and classification. Chapter 3 describes our algorithm for general human activity recognition from video. Chapter 4 presents the system framework for pathological gait identification. Chapter 5 details an algorithm for pathological gait pattern identification using 3D motion captured data. In chapter 6, we present another algorithm for identifying pathological gait from a video sequence. In chapter 7, we summarize the thesis.

Chapter 2

Related work

In this chapter we present an overview of the previous related works. We start by reviewing the literature for general human activity recognition. In this part, we review the works related to region of interest (ROI) segmentation which usually is the first step of human activity analysis system, the works for representing and recognizing human activity and some general classification techniques. Then, we review related approaches based on gait classification for diagnostic purposes, which includes works presented in both the computer vision field and biomedical field.

2.1 General human activity recognition

2.1.1 Region of interest segmentation (ROI)

ROI Segmentation in video aims at detecting regions that correspond to moving objects such as vehicles and people in a sequence of frames. Nearly every system of vision-based human motion analysis starts with ROI Segmentation. It is an important and difficult issue in a human motion analysis system. Detecting the ROI provides a focus of attention for later processes. Usually, only those changing pixels need to be considered. However, reliable and fast segmentation is difficult because of the diverse environment changes, such as illumination and shadow. Currently, most segmentation methods use either temporal or spatial information of the images. Several approaches are discussed in the following.

- **Temporal difference**

Temporal difference based approaches use pixel-wise difference between two or three consecutive frames in an image sequence to extract moving regions, such as the algorithms presented in [1, 39]. In [39], A.J. Lipton et al. detected moving targets in video streams using temporal difference. After the absolute difference between the current and the previous frame was obtained, a threshold function was used to determine change. Using connected component analysis, the extracted moving regions were clustered into moving regions. These regions were classified into predefined categories according to image-based properties for later tracking. In [1], R.T. Collins et al. improved the algorithm by developing a hybrid algorithm for ROI segmentation by combining an adaptive background subtraction algorithm with a three-frame differencing technique. This hybrid algorithm is very fast and surprisingly effective for detecting moving objects in image sequences.

Temporal difference usually does a poor job of extracting the entire relevant feature pixels, e.g., possibly generating holes inside moving regions, even though it is adaptive to dynamic environments.

- **Optical flow**

Optical flow is used to describe coherent motion of points or features between image frames. Optical flow based ROI segmentation uses characteristics of flow vectors of moving objects over time to detect moving regions in an image sequence, such as the algorithm presented in [18, 40, 41]. In [18], D. Meyer et al. used a monotony operation which computed the displacement vector field to initialize a contour-based tracking algorithm, called active rays. Then, articulated objects were extracted for gait analysis. In [41], H.A. Rowley et al. focused on the ROI segmentation of optical flow fields of articulated objects. They added kinematic motion constraints to each pixel, and to combine ROI segmentation with estimation using EM (Expectation Maximization) computation. Most optical flow computation methods are computationally complex and very sensitive to noise, and cannot be applied to video streams in real-time without specialized hardware. More detailed discussion of optical flow can be found in Barron's work [40].

- **Background subtraction**

Background subtraction [28, 42-47] is a particularly popular method for ROI segmentation, especially for scenes with a relatively static background. It calculates the difference between the current image and a reference background image pixel-by-pixel to detect moving regions in an image. However, it is sensitive to the dynamic changes of the scene due to lighting and extraneous events. There are numerous approaches for background subtraction that differ in the type of the background model and the procedure used to update the background model. The simplest background model is a temporally averaged image that is a background approximation similar to the current static scene. Based on the observation that the median value was more robust than the mean value, in [44], Yang and Levine proposed an algorithm for constructing the background model by taking the median value of the pixel color over a series of images. The median value, as well as a threshold value, determined by using a histogram procedure based on the least median squares method, was used to create the difference image. This algorithm could handle some of the inconsistencies due to lighting changes, etc.

There are different methods for building adaptive background models in order to reduce the influence of dynamic scene changes on ROI segmentation. For instance, Karmann and Brandt in [42] and Kilger in [43], respectively, proposed an adaptive background model based on Kalman filtering to adapt temporal changes of weather and lighting. Some statistical methods to detect changing regions from the background have been represented recently. Those statistical approaches use the characteristics of individual pixels or groups of pixels to construct background models. Usually, the statistics of the background pixels can be updated dynamically during processing. Each pixel in the current image can be classified into foreground or background by comparing with the statistics of the current background model. This approach is becoming increasingly popular due to its robustness to noise, shadow and change of lighting conditions. In [46], C.R. Wren et al. presented an adaptive background mixture model for real-time tracking. In their work, they modeled each pixel as a mixture of Gaussians and used an online approximation to update it. The Gaussian distributions of the adaptive mixture models were built to evaluate if the pixels come from the background. Their algorithm resulted in a reliable, real-time outdoor tracker that can deal with lighting changes and clutter. In [28], I. Haritaoglu et al. built a statistical model by representing each pixel with three values: its minimum and maximum intensity values, and the maximum intensity difference between consecutive frames observed during the training period. The model parameters were updated periodically. The quantities that are characterized statistically are typically colors or edges. For example, in [47], S.J. McKenna et al. used an adaptive background model that combines color and gradient information, where each pixel's chromaticity was modeled by using the means and the variances, and its gradient in the x and y directions was modeled by gradient means and magnitude variances. Background subtraction was then performed to cope with shadows and unreliable color cues effectively.

- **Other methods**

In addition to the basic methods described above, there are some other approaches to ROI segmentation. In [48], N. Friedman et al. implemented a mixture of Gaussian classification model for each pixel. This model attempted to explicitly classify the pixel values into three separate predetermined distributions corresponding to background, foreground and shadow. It could also update the mixture component automatically for each class according to the likelihood of membership. Hence, slow-moving objects were well handled, and shadows were eliminated effectively. In [49], E. Stringa et al. also proposed a novel morphological algorithm for scene change detection. From a practical point of view, the statistical methods described in Section 2.1.2 are far better choices due to their adaptability in more unconstrained applications.

2.1.2 Activity representation and recognition

After successfully segmenting the moving humans from the image sequence, the problem of understanding human behavior from image sequences includes activity representation and recognition. Recognizing human activity may be simply considered as a classification problem of time varying feature data, i.e., matching an unknown test sequence with a group of labeled reference sequences representing typical human activities. It is obvious that the basic problem of human activity recognition is how to effectively represent the activity, and how to learn the reference activity sequences from training data. These are hard problems that have received increasing attention from researchers. Most of the previous methods can be classified into the following three classes based on the features they use for recognition.

- **Motion feature based methods**

Motion-based features were used in [50-57][106-108]. In [50], Sun et al. compare the use of affine motion parameters and optic flow as features for building HMMs to recognize human activities. In [51], Masoud et al. use the result of an Infinite Impulse Response (IIR) filter to measure motion and construct a feature image in which recent motion is brighter than older motion. Then, they use PCA to obtain a set of representative features. A distance measure (Minimum Distance to Average) is defined and recognition is performed by calculating the distance to some reference representing the learned activity. In [52], Hamid et al. extract spatio-temporal features such as the relative distance between two hands and their velocities and use dynamic Bayesian networks to recognize human activities such as writing, drawing and erasing on a white board. In [53], Ben-Arie et al. represent an activity by a set of velocity vectors of the major body parts (hands, legs, and torso), store the representation in a set of multi-dimensional hash tables, and use multidimensional indexing to recognize activities. In [54], they use the distribution of motion over the image space in the x and the y directions to recognize five actions (sit down, get up raising hand, nodding, and shaking hand). In [55], the features consist of two-dimensional meshes. First, optical flow fields were computed between successive frames, and each flow frame was decomposed into a spatial grid in both horizontal and vertical directions. Then, motion amplitude of each cell was accumulated to form a high-dimensional feature vector for recognition. In order to normalize the duration of motion, they assumed that human motion was periodic, so that the entire sequence could be divided into many circular processes of certain activity that were averaged into a sequence of temporal stages. Finally, they adopted the nearest neighbor algorithm for human action recognition. In [56], Yamato et al. made use of the mesh features of 2-D moving human blobs such as motion, color and texture, to identify human behavior. In the learning stage, HMMs were trained to generate symbolic patterns for each action class, and the optimization of the model parameters was achieved by the forward-backward algorithm. In the recognition process, given an image sequence, the output result of forward calculation was used to guide action identification. Moreover, in [57], a comprehensive framework using the statistical decomposition of human body dynamics at different levels of abstractions was presented to recognize human motion. In the low-level processing, the small blobs were estimated as Gaussian mixture models based on motion similarity, color similarity and spatial proximity from the previous frames. Meanwhile, the regions of various body parts were implicitly tracked over time. During the intermediate-level processing, those regions with coherent motion were fitted into simple movements represented by dynamic systems. Finally, HMMs were used as a mixture of these intermediate-level simple movements to represent complex motion. Given the input image sequence, recognition was accomplished by maximizing the posterior probability of the HMM. In [106], the authors proposed a large number of features based on velocity of subject and optical flow in the region of interest. Then, they resorted to several methods to evaluate different combinations of features based on the recognition rate achieved with the classifier. Finally, they use hierarchical Bayesian classifiers for recognition. In [107], a Burt-Adelson Pyramid approach was used to extract multi-resolution optical flow as features. Then, the author used HMMs to model and recognize human activities. In [108], an epitomic representation was presented for modeling human activities in video sequences. At first, a video sequence is divided into segments. Each segment is modeled using a linear dynamical systems based on positions and velocities of subject. Therefore, in their method, an activity is modeled as a sequence of linear dynamical systems. Then, a geodesic distance between two sequences of linear dynamical systems is defined to recognize the human activities.

- **Shape feature based methods**

Shape-based features are used in [59-61][109-114], where the body's 2D or 3D shape features are used to recognize activities. In [58], they use the angles subtended by three body components with the vertical axis as a feature vector and use the nearest neighbor classifier to recognize seven actions (walking, sitting, standing up, bending, getting up, etc) from profile views. In [59], an appearance-based, view-independent, 3D shape description is presented for classifying and identifying human posture using a support vector machine. The proposed global shape description is invariant to rotation, scale and translation and varies continuously with 3D shape variations. This shape representation is used for training support vector machines allowing the characterization of human body postures from the computed visual hull. The main advantage of the shape description is its ability to capture human shape variations allowing for the identification of body postures across multiple people, but this method needs data from multi-cameras. In [60], shape is represented by edge data obtained from canny edge detector, and key-frames are defined for each activity. Then, a shape matching algorithm is used to localize key frames of new video sequences and recognize forehand and backhand strokes in tennis video clips. [61] presents a new approach to automatically recognize human activities from video sequences acquired with a large scale view in order to monitor a wide area with a single camera. The recognition process is performed in two steps: at first the human body posture is estimated frame by frame and then the temporal sequences of the detected postures are statistically modeled. Body postures are estimated starting from the binary shapes associated to humans, selecting as features the horizontal and vertical projection histograms and using them as input to an unsupervised clustering algorithm. The Manhattan distance is used for building the clusters and for run-time classification. Statistical modeling of the detected postures is performed by Discrete Hidden Markov Models. In [109], the authors used a 50-dimensional histograms of combined shape context and edge features extracted at a variety of scales on the silhouette as feature. Then, they used discriminative Conditional Random Field (CRF) and Maximum Entropy Markov Models (MEMM) to model human activity. By comparing the results of the method based on CRF and MEMM with the results of the method based on HMM, they declared that CRFs and MEMMs outperform HMMs. In [110], body parts were extracted from pixel-level images and used to estimated pose and gestures of subjects. Then, a context-free grammar (CFG) based representation scheme was used to represent and recognize actions and interactions. In [111], the authors presented feature based on an extended radon transform of binary human silhouettes. Then, a set of HMMs based on the extracted features are trained to recognize activities. They declared that the new feature is robust to frame loss in video, disjoint silhouettes and holes in the shape, and thus achieves better performance in recognition. In [112], the authors represent the silhouette images of a person undergoing an activity as a manifold in the image space. Then they distinguished between human activities by comparing learned manifolds. Different extrapolation techniques, which are used to find the positions of novel samples on a previously learned manifold, were tested in the experiments. Those extrapolation techniques include neural networks, generalized radial basis functions and Nystrom estimator. They concluded that the Nystrom estimator is the best extrapolation technique human activity recognition using silhouettes. In [113], the authors used kernel principal component analysis (KPCA) to reduce the silhouette images and used the obtained results as features. Then, they used factorial conditional random field (FCRF) to model and recognize human activities. They concluded that the FCRF is superior to both HMM and general CRF. In [114], the authors presented a framework for view-independent human activity recognition. They used three dimensional occupancy grids, built from multiple viewpoints, in an exemplar-based HMM to model activity in training stage.

In the recognition stage, 3D models are projected to 2D images that are compared to the silhouettes extracted from the probe video.

- **Other feature based methods**

Bobick and Davis [62] present a view-based approach for the representation and recognition of action using temporal templates. They made use of the binary MEI (Motion Energy Image) and MHI (Motion History Image) to interpret human movement in an image sequence. First, a set of images in a sequence was extracted by differencing, and the set of images was accumulated in time to from MEI. Then, the MEI was enhanced into MHI， which is a scalar-valued image. Taken together, the MEI and MHI could be considered as a two-component version of a temporal template, a vector-valued image in which each component of each pixel is some function of the motion at that pixel position. Finally, these view-specific templates were matched against the stored models of views of known actions during the recognition process. Based on PCA, Chomat and Crowley [63] generated motion templates by using a set of temporal-spatial filters computed by PCA. A Bayes classifier was used to perform action selection. In [115] K. Tsuda et al., at first, built a middle level wordbook of prototypes using k-means clustering based on low level spatio-temporal features. Then, a sequential representation of human acitivities based on middle level words is obtained. Further, LPBoost classifier is used for recognition.

Recently, there are several new methods for human activity recognition. In these methods, representation of activities are not only based on the features obtained from the subject, but also other information such as the objects involved in the activity or the environments, e.g., [116,117]. In [116], the authors presented an approach to recognize activities by identifying the objects used in the scene. They used dynamic Bayesian network models which combine radio frequency identification (RFID) and video data to recognize object labels and activity. The method can be useful in some particular scenarios such as cooking, which involve a relatively small number of repeated actions but many different objects. In [117], the authors tried to recognize human activities from static images. In this work, they modeled the activity by a generative graphical model base on the "environment" and "critical objects" involved in the activities. For an unknown image, they used the trained graphical model to recognize the scene environment class and the object classes in order to recognize the activity.

2.1.3 Classification techniques

Action recognition could be considered a matching problem of time-varying data. The general analytical methods for matching time-varying data are Dynamic time warping, Hidden Markov models and Neural network.

- **Dynamic time warping**

Dynamic time warping (DTW) [64] is a template-based dynamic programming matching technique which is already widely used in speech recognition. It has many advantages, such as conceptual simplicity and robust performance. Because of that, it was used in matching human movement patterns [65,66]. In DTW technique, even if the time scale between a test pattern and a reference pattern is inconsistent, it can still successfully establish matching as long as time ordering constraints are satisfied. However, it is usually more susceptible to noise and the variations of the time interval of the movements.

- **Hidden Markov models**

Hidden Markov Models (HMMs) [67] is a more sophisticated technique for analyzing time-varying data with spatio-temporal variability. Its model structure can be summarized as a hidden Markov chain and a finite set of output probability distributions. The use of HMMs includes two stages: training and classification. In the training stage, the number of states of an HMM must be specified, and the corresponding state transformation and output probabilities are optimized in order for the generated symbols to correspond to the observed image features. In the matching stage, the probability that a particular HMM possibly generates the test symbol sequence corresponding to the observed image features is computed. HMMs are superior to DTW in processing unsegmented successive data, and are therefore extensively being applied to the matching of human activity patterns [68,69,70]. Although the HMM approach may overcome the disadvantages of the template matching approach, it usually involves complex iterative computation. Meanwhile, how to select the proper number of states and the dimensionality of the feature vector remains difficult.

- **Neural network**

Neural network (NN) [71,72] is also an interesting approach for analyzing time-varying data. As larger data sets become available, more emphasis is being placed on neural networks for representing temporal information. For example, in [71], Guo et al. used it to understand human activity pattern, and in [72] Rosenblum et al. recognized human emotion from motion using radial basis function network architecture.

- **Other methods**

In addition to the three approaches described above, the PCA (Principle Component Analysis) method [73] and some variants from HMMs and NNs such as CHMM (Coupled Hidden Markov Models) [73], VLMM (Variable-Length Markov Model) [74] and TDNN (Time-Delay Neural Network) [75], have also appeared in the literature.

2.1.4 View-invariant activity recognition

Most of the work on activity recognition is view dependent and deals with recognition from one fixed view. The task of recognizing human activities from different views is still unsolved. In [30], Rao et al. developed a view-invariant representation of action consisting of a sequence of *dynamic instants* and *interval*s, which is computed by using spatiotemporal curvature of hand trajectory. Dynamic instants represent changes in motion, such as change of speed, direction, acceleration, and curvature. Intervals represent the time-period between any two dynamic instants. In [76], Parmeswaran et al. represented each human action by a set of 3D curves which are quasi-invariant to the viewing direction. In [31], Ogale et al. presented an approach that uses training videos from multiple views to automatically create view-independent representations of actions within the framework of a probabilistic context-free grammar. This grammar is then used to parse a new single-viewpoint video sequence to deduce the sequence of actions in a view-invariant fashion.

In our work, we combine motion-based features with shape-based features to model human activities. We represent each activity by a set of Hidden Markov Models, where each model represents the activity viewed from a specific direction (i.e., viewing angle) to realize the view-invariance. We also present a voting based method to segment and recognize continuous complex human activities.

2.2 Approaches in diagnosis of pathological gait

Analysis of a person's gait during physical activities has many applications in the biomedical fields. For example, gait analysis is very helpful in assessing the potential risk of falling, which is important because injuries induced by falls have a significant impact on the rates of mortality among elders [77]. Another example is that the analysis of a patient's gait during the rehabilitation process can provide useful insights about the effect of joint replacement on a patient's walking ability and it could be used to develop biofeedback to help in patients' treatment.

In the past few years, gait-based analysis and classification have received considerable attention in the biomedical field. For example, in [78], Holzreiter and Kohle used a neural network to assess gait patterns from ground reaction forces to identify ''normal'' and ''pathological'' gait patterns. In [79], Begg, et al. extracted statistical features from minimum foot clearance data as a feature and used support vector machines to classify gait patterns of young and old people. They also claim this algorithm has potential for wider applications, such as identifying pathological gait. In [80], W. Wu et al. obtained ground reaction force (GRF) from a force platform, and used an algorithm that combines an artificial neural network and a genetic algorithm to assess patients after ankle surgery. Actually, it is intuitive to assess patients' gaits by measuring symmetries of their gaits, because in normal individuals, gait patterns with respect to time, distance and vertical force are fairly symmetrical and only deviate by a small percentage from perfect symmetry [32]. In [33], Brandstater et al. showed that for a group of persons with acute stroke, the symmetry of swing time was related to the stage of their motor recovery. In [35-37], authors used temporal-distance symmetry as an indicator of gait performance and a measure for evaluating intervention strategies. In [81], Morita et al. analyzed the relationship between symmetry of the impulse of ground reaction force (GRF) and gait speed in persons with stroke and concluded that the symmetry of GRF well reflects the degree of motor recovery in these patients. In [32], Kim et al. found that symmetry in temporal-distance measures (stance time, swing time, and step length) is accompanied by symmetry in GRF measures during gait. They quantified the relationship between the symmetry of these variables and gait speed in a group of individuals with chronic stroke.

Most of the developed symmetry measures are based on stance time, swing time, step length and vertical ground reaction force. Also, most of them just focused on the information provided by lower limb movement. The symmetry of the upper body movement is rarely examined in persons with stroke. The upper body movement may also be informative since it also provides information of body movement which is affected by patient's disease. Therefore, a new gait symmetry measure based on the movement of the whole body may provide better assessment.

Another issue is the expense of motion-capture systems and devices used to capture data for patient gait assessment in previous works. Moreover, it usually takes a long time and tedious work to capture data (The patients are asked to take off their outerwear. Sensors and devices have to be carefully attached on patients' bodies.). Since common video cameras are inexpensive, it is advantageous to use data captured by these cameras to achieve the same goal.

Recently, in [92], C. M. Kawamura et al. compared 2D gait analysis and 3D gait analysis of patients with spastic diplegic cerebral palsy, and concluded that 2D visual observations are inadequate for the quantitative assessment of pathological gaits. Even though, the question of whether 2D visual information is suitable for the qualitative assessment of gait or not still deserves further research.

In this work, we present two gait analysis algorithms, based on symmetry, for the purpose of identifying normal and pathological gait patterns. In the first algorithm, we present a discrete cosine transform method to measure the symmetry of two 3D trajectories. A feature vector based on measuring the symmetry of movement of the whole body is used to present gait. Then, a support vector machine is trained for gait classification. We demonstrate the accuracy of the classification by a set of experiments that shows that our method for gait classification based on gait symmetry is highly effective. We also show that the result of experiment using data from all the body is better than the result of experiment using data from the lower body.

The second algorithm is for the same purpose but using single video data. A symmetry measure in 2D plane is presented to represent gait from video. The same SVM based classification method is used to identify pathological gait. Experiments are executed for both 2D projections of 3D data and video data.

Chapter 3

Human activity recognition

We describe in this chapter our algorithm for general human activity recognition on a coarse level.

As mentioned in chapter 2, both motion-based and shape-based features have their own limitations in representing human activity. Motion-based features can depict the approximate moving direction of the body, but most motion-based features are not robust in capturing velocity. For example, motion-based features can easily discriminate between walking and sitting down, but fail to discriminate between walking and slow running. On the other hand, shape-based features can capture some pose information of the body, but without motion information its capability of describing human activity is limited. Since motion-based feature and shape-based features are complementary (shown in table 3.1), combining both features can enhance the robustness of activity recognition.

In this chapter, we combine motion-based features with shape-based features to represent and model human activities. We represent each activity by a set of Hidden Markov Models, where each model represents the activity viewed from a specific direction (i.e., viewing angle) to realize the view-invariance. We tested our algorithm on two sets of video clips. The first set was used in [58] and the second set is a database of 173 video clips for four activities that we collected. The results show that the algorithm is robust and capable of recognizing activities from random viewing directions. We also performed experiments to compare between the performance using either shape or motion features alone and using both features together. The experiments show that combining the two features results in better recognition performance. Also, the experiments on complex activity data show the efficiency of voting and HMM based segmenting and recognizing algorithm.

In section 3.1, we describe our activity recognition algorithm. This includes the feature extraction and the activity modeling. In section 3.2, we present the experimental results. In section 3.3 we discuss the robustness and the computational efficiency of the algorithm. Finally, we conclude the chapter in section 3.4.

Table 3.1 Advantages and disadvantages of motion and shape features

	Motion feature	Shape feature
Information	All body	Contour
Results of Background subtraction	Not sensitive	Sensitive
Variation of Velocity	Sensitive	Not sensitive
Variation of person’s figure	Little sensitive	Sensitive
Different video frame rate	Sensitive	Not sensitive

3.1 Single Activity Recognition

Our algorithm [101] consists of three steps: 1) region-of-interest extraction, 2) feature extraction, and 3) HMM-based activity recognition. The goal of the region-of-interest extraction is to separate the region that contains the human body from the background. To perform this task, we used a background subtraction algorithm. The details of this algorithm are presented in section 3.1.1. Then, the motion and the shape features are extracted from the region-of-interest, as explained in sections 3.1.2. HMM models are used for recognizing the activities as explained in section 3.1.3.

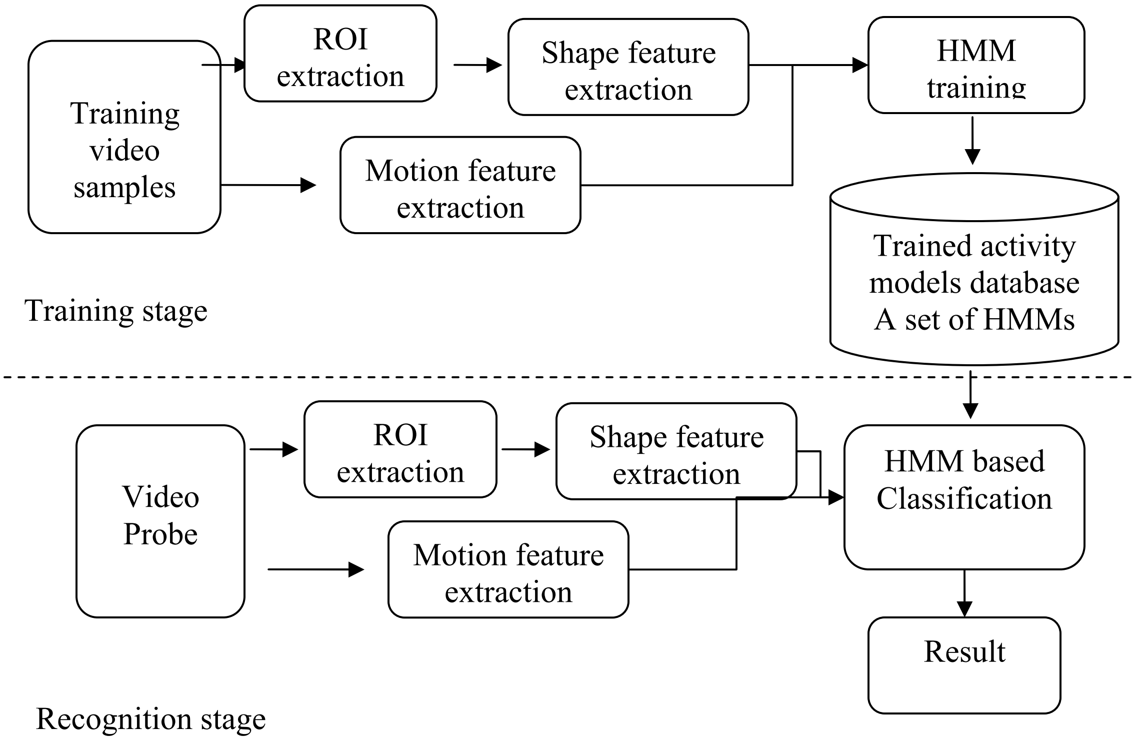


Figure 3.1 Block diagram of the system for single activity recognition

3.1.1 Region of interest (ROI) extraction

We assume that each video clip includes only one person performing a single activity. We used a simplification of the background subtraction algorithm presented in [82] to extract the ROI. In [82], each background pixel was represented by a mixture of Gaussians. The probability density is estimated by using information in the recent history frames. To accelerate the processing, we simplified the algorithm by representing the color value of each background pixel using a single Gaussian distribution instead of a mixture of Gaussians. The probability density function that a background pixel will have color value x_t at time t is estimated as

$$P(x_t) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x_t - \bar{x})^T \Sigma^{-1} (x_t - \bar{x})} \quad (1)$$

If we assume independence between the different color channels with a different kernel bandwidth σ_k^2 for the k^{th} color channel, then

$$\Sigma = \begin{pmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{pmatrix} \quad (2)$$

and the estimated probability of $x_{t,i,j}$ belonging to background is reduced to

$$P(x_{t,i,j}) = \prod_{k=1}^{3} \frac{1}{(2\pi\sigma_{i,j,k}^2)^{\frac{1}{2}}} e^{-\frac{1}{2}\frac{(x_{t,i,j,k} - \bar{x}_{i,j,k})^2}{\sigma_{i,j,k}^2}} \quad (3)$$

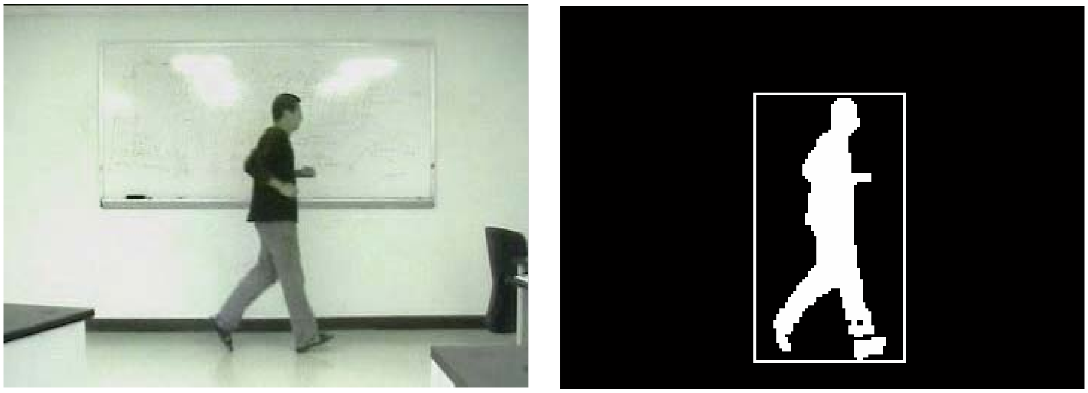
Using this probability estimate, we compute the probability $P(x_{t,i,j})$ for pixel $I(i, j)$ in frame t, i.e., the probability of belonging to the background. Then, we can use the following threshold to obtain the foreground:

$$I_t(i, j) = \begin{cases} 1 & P(x_{t,i,j}) \leq T \\ 0 & P(x_{t,i,j}) > T \end{cases} \quad (4)$$

where T is an experimentally selected threshold. In our experiments, we ran this model on a training sequence that only contained the background and selected the threshold such that it achieves an average of 2% false positive rate.

The mean, $\bar{x}_{i,j,k}$, and the standard deviation, $\sigma_{i,j,k}$, of the k^{th} color channel for pixel $I(i,j)$ can be estimated from the previous N frames (in our experiments, N is set to 10). The background parameters need to be updated continuously to adapt to changes in the scene. The update is performed in a first-in first-out manner. That is, the oldest sample is discarded and a new sample is included in the estimation of each background pixel. Here, we chose the selective update method that updates the background probability distribution by pixels that have been classified as background.

Figure 3.2.a shows a frame from a video clip that captures a running person and Figure 3.2.b shows the result of background subtraction with the ROI region inside the rectangle.



(a) (b)

Figure 3.2 a frame example from a running sequence
(a) An original frame (b) The ROI obtained after background subtraction

3.1.2 Feature Extraction

3.1.1.2 Motion Features

In [50], the experimental results demonstrated that using optical flow for activity recognition results in better performance than using the affine motion parameters. In this chapter we use optical flow to describe the motion. After applying the background subtraction and noise removal, we obtain a rectangle region of interest (ROI); one example is shown in Figure 3.2. We compute the optical flow, $o(i, j)$, for each pixel in the ROI, and normalize the optical flow values as follows:

$$\overline{o}(i,j) = o(i,j) / o_{\max} \quad (5)$$

where

$$o_{\max} = \max\{\ |o(i,j)| \,\big|\, i,j \in ROI\} \quad (6)$$

Because people usually perform the same activity with different speed every time, the motivation of normalization is to eliminate the effect of speed variation.

Then, we partition the ROI into 64 blocks, *B*(*k*), of equal sizes, where *k = 1, ..., 64.* The average optical flow vector for every block is then computed by:

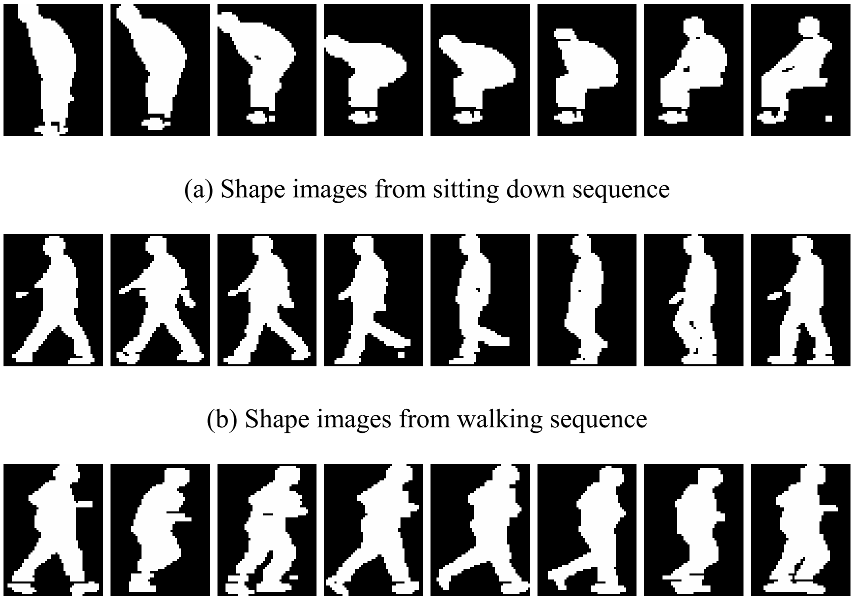
$$\overline{O}_k = \begin{bmatrix} \overline{O}_{kx} \\ \overline{O}_{ky} \end{bmatrix} = \frac{1}{n} \sum_{i,j \in B(k)} \begin{bmatrix} \overline{o}_x(i,j) \\ \overline{o}_y(i,j) \end{bmatrix} \quad (7)$$

where *n* is the number of pixels in a single block.

Then, we compose the vector $O=[\overline{O}_1,\overline{O}_2,...,\overline{O}_{64}]^T$ for every frame to represent its motion feature vector, where each element contains two components along x and y.

3.1.1.3 Shape Features

After the ROI regions are obtained from the background subtraction, their sizes are normalized to 64 by 48 pixels. Figure.3.3 shows some examples of normalized ROI regions from three frame sequences showing three different activities. Each normalized ROI image is then represented as a vector by concatenating the rows in a raster scan fashion. Thus, all ROI images are mapped to a collection of points in a large dimensional feature space, i.e., 3072 dimensions. Because human shape images have some similarity, these points are not randomly distributed in that space. To efficiently use the shape information, we use principal component analysis (PCA) to reduce the 3072 dimensional feature space to a lower dimension space. The main goal is to find those vectors that can best represent the distribution of human shape images.



(a) Shape images from sitting down sequence

(b) Shape images from walking sequence

(c) Shape images from running sequence

Figure 3.3 Examples of normalized shape images

Let $S_1, S_2, ..., S_N$ be the human shape vectors from the training set of ROIs. An average human shape vector is computed as:

$$S_{avg} = \frac{1}{N} \sum S_i \qquad (8)$$

Then, the set of orthonormal eigenvectors, $V_i, i = 1 \ldots N$, and their corresponding eigen values, $\lambda_i, i = 1 \ldots N_i$, can be computed by using the covariance matrix C.

$$C = \frac{1}{N} \sum_{i=1}^{N} \Phi_i \Phi_i^T \qquad (9)$$

where $\Phi_i = S_i - S_{avg}$. The eigenvectors, $V_i, i = 1 \ldots N$, are ranked according their associated eigenvalues, λ_i s. We choose the top M eigenvectors as the bases of the reduced space. In our experiments, M was set to 90 so that 92.8% of the energy is preserved. Figure 3.4 shows the first 24 and the last eight eigen-shape images out of the 90 obtained eigen-shape images.

Given a shape vector, S, it is projected to the new feature space by

$$f_k = V_k^T (S - S_{avg}) \text{ for } k = 1, ..., M \quad (10)$$

Here f_k is the k_{th} eigen-shape component which is the projection of S on vector V_k. Therefore, $F = [f_1, f_2, ..., f_M]^T$ is the projection of S.

The shape feature and motion feature are related for representing human activities. The relations include temporal and spatial relations. Intuitively, the temporal relation is more important for representing human activity. In this work, we just consider the temporal relation. The motion and shape feature vectors are combined together in a single feature vector

$$U_i = [\overline{O}_{1x}, \overline{O}_{2x}, ..., \overline{O}_{64x}, \overline{O}_{1y}, \overline{O}_{2y}, ..., \overline{O}_{64y}, f_1, f_2, ..., f_M]^T \quad (11)$$

where $\overline{O}_{ix}$ is the x component of $\overline{O}_i$, and $\overline{O}_{iy}$ is the y component of $\overline{O}_i$, and f_i's are the eigen shape components, where *M* is set to 90. Every video clip was then represented as a sequence, $U = \{U_{1,} U_2, ..., U_L\}$, where *L* is the number of frames in the sequence.

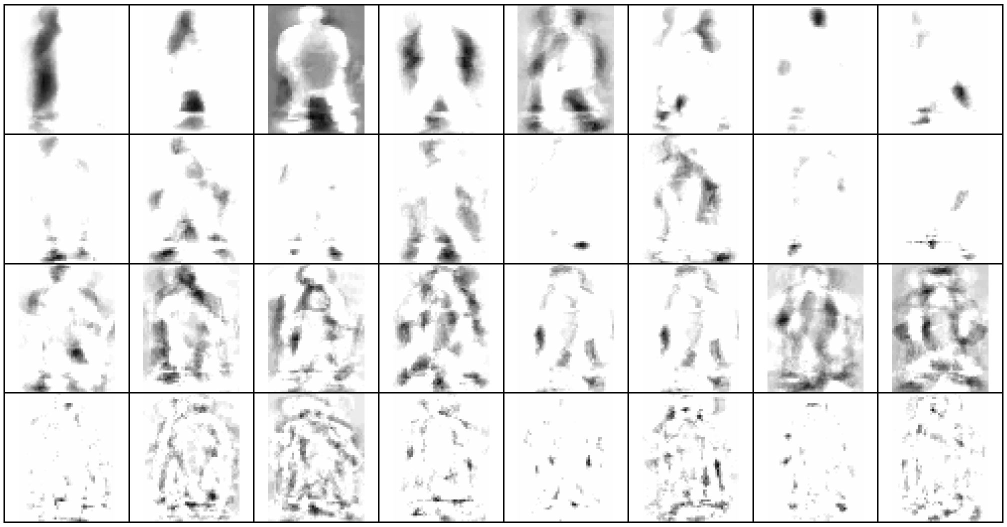


Figure 3.4 32 out of the 90 eigen-shape images

3.1.3 Activity Modeling

Hidden Markov models were successfully used for speech recognition because of their capability of recognizing spoken words independent of their duration [83][84]. In gesture recognition and human activity recognition, we can face the same situation, where the same gesture or activity can occur over different times. Some of the previous research on gesture and human activity recognition [57][85-88] used HMMs. In this chapter we use HMMs for activity modeling.

For view independent recognition of activities, we build several models for each activity, where each model represents the activity from a different viewing direction, to capture the variations arising from the changes in the view. For a given activity, *j*, through training, we obtain a set of HMMs:

$$A_j = \{A_{j1}, A_{j2}, ..., A_{jN}\} \qquad (12)$$

Each model represents the activity from a different viewing angle. All the HMMs that we used have the same fully connected topology. The number of HMMs' states was empirically determined by an experiment based on cross validation. In the experiment, we use both motion and shape features for training and testing the HMMs with states from 4 to 9. For each case, we use the leave-one-out method to train and test the HMMs. Table 3.2 shows the average recognition rate by the HMMs with different number of states.

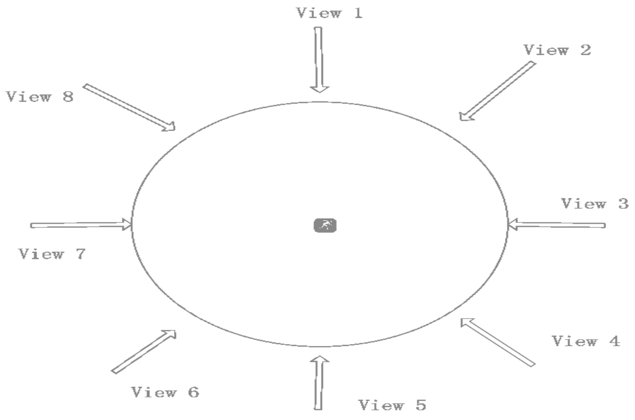


Figure 3.5 The eight views used for capturing training sequences

Table 3.2 The number of states of HMMs selection

Number of States	Average Recognition Rate(%)
4	75.7
5	69.3
6	86.1
7	85.8
8	79.7
9	66.7

Based on the results in this table, the HMMs with six states gave the best recognition results.

In our HMMs, each observation was modeled as a mixture of Gaussians. Two mixtures per feature were used in the experiments. We used the maximum-likelihood approach to classify each activity:

$$A = \underset{A_j \in all\ \ activity}{\arg\max}\ P(U \mid A_j) \qquad (13)$$

$P(U \mid A_j)$ is the conditional probability for activity *J*, and is computed by

$$P(U \mid A_j) = \max_i P(U \mid A_{ji}), \quad i = 1,\ldots,N \quad (14)$$

where U is a feature vector sequence of an unknown activity. In the training stage, we segmented the training video into short clips where each clip contained only a single activity. Then, those video sequences were classified manually into different activity classes and different views. Each HMM model was trained 10 times by using the Expectation Maximization (EM) algorithm, and the model that resulted in the highest likelihood for the training data was selected. This is because HMMs are known to produce models of varying quality, even when trained repeatedly with the same data [73].

3.2 Segmentation and Recognition of Complex Activity

In our algorithm [102], activity segmentation and recognition are combined in one process. During training, we train the HMM*s* for each single activity separately. Then, during recognition we slide a window of length N over the sequence of frame features and classify the activity represented by the sequence in the window (see Figure 3.7). For a video clip with M frames we obtain a set of results r_i, $i= 1, 2, \ldots, M\text{-}N+1,$ where resul*t* r_i is the activity assigned to window w_i. The result is used as a vote assigned to each frame in this window. We shift the window frame by frame and repeat the classification process. This will result in obtaining N results, r_j, for frame f_i, where $i\text{-}N+1 < j < i+1$. These classification results are considered as votes and we classify the activity of a frame by the activity that has maximum votes.

The result of the previous classification process produces a set of voting curves, a curve for each activity, for the sequence of frames. A low-pass filter is applied to smooth the voting curves in order to obtain the final segmentation and recognition results. Figure 3.8 shows an example of a sequence of frames from a video that contains a sequence of activities. Figure 3.9 shows the voting results (after being filtered) for the sequence of Figure 3.8. In Figure 3.9, seven curves represent votes for seven activities obtained separately for each frame.

3.3 Experimental Results

We have performed two sets of experiments. The first experiment was performed on our own database of video clips that have frames of 352x240 pixel resolution and 30 frames per second. The database contains 173 sequences of four activities (47 walking sequences, 54 running sequences, 36 standing up sequences and 36 sitting down sequences) captured from different views. Out of the 173 sequences, 128 sequences (32 for each activity) were captured at the eight views shown in Figure 3.5. The remaining 45 sequences were captured from arbitrary viewing directions. The length of each video clip is between 25 to 120 frames. To make the most use of the video clips in evaluating our approach, we used the leave-one-out cross validation method for training and testing. Therefore, each time 96 out of the 128 video clips that were captured from the eight views were used for training. The other 32 clips, one for each activity from the different eight views, along with the 45 clips captured from arbitrary views (not including the 8 views used for training) were used for testing. The average of all the results was calculated to give an overall evaluation of our algorithm.

For each activity, we trained eight HMMs, where each HMM corresponds to one of the eight views. Our selection of eight views for training was a tradeoff between the number of quantized views used for training and recognition accuracy. The greater the number of quantized views used for training, the higher the recognition rate is. To demonstrate this observation, we experimented with training using fewer views, i.e., four views and two views. The test results revealed degradation of the recognition accuracy with the decrease in the number of views used for training. For two views the average recognition accuracy was about 30% and for four views the average recognition accuracy was about 52%, while for eight views, as shown in Table 3.2, the average accuracy was about 88.6%.

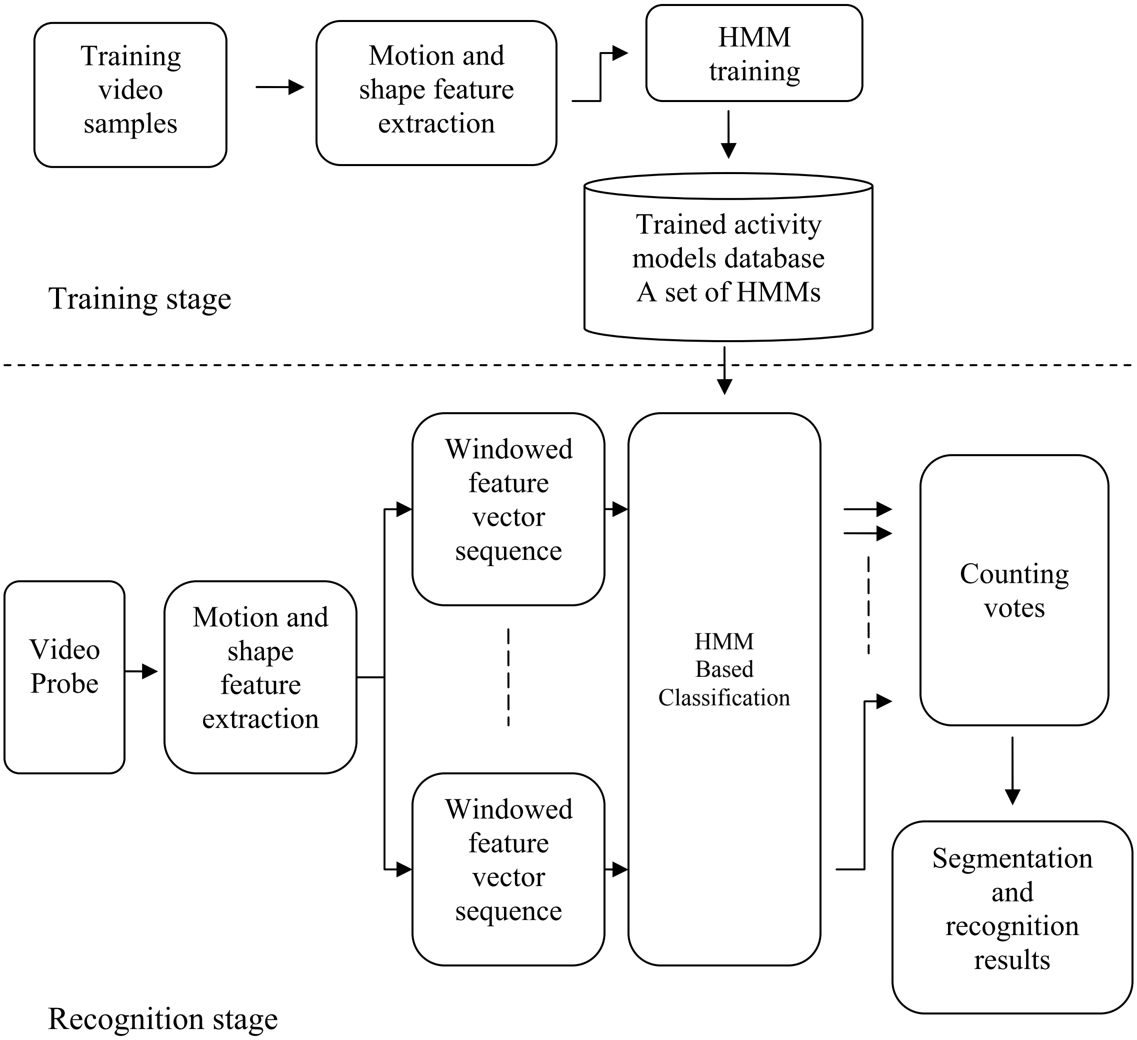

Training video samples
Motion and shape feature extraction
HMM training
Trained activity models database A set of HMMs
Training stage
Video Probe
Motion and shape feature extraction
Windowed feature vector sequence
Windowed feature vector sequence
HMM Based Classification
Counting votes
Segmentation and recognition results
Recognition stage

Figure 3.6 Block diagram of the system for complex activity recognition

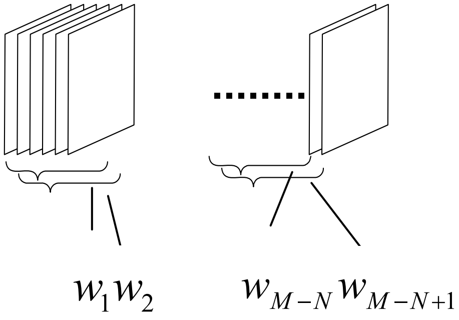


Figure 3.7 Sliding windows through the sequence of frames

Table 3.3 shows the results of activity recognition from the video clips using only motion features, only shape features, and both features together. The table shows both the average recognition rate as well as the standard deviation. Table 3.4 shows the confusion matrix of using both motion and shape features. The results in the table show the average result of four experiments, each of them with a different leave-one-out training set and corresponding test set. As shown in table 3.3, the average classification accuracy is 88.6% when using both motion and shape features, which is better than the 79.5% and the 82.1% recognition rates obtained when using either motion or shape features separately. From the experiments, we can see that the recognition rates for walking and running are lower than that of sitting down and standing up. When we checked those misclassified video clips, we found that most of the misclassifications were for walking and running activities captured from front and rear view .This is expected due to the high degree of similarity between walking and running in these views. Figure 3.10 shows four sequences for walking and running from front and rear views. In the profile view, running and walking activities are easier to distinguish by both features. From table 3.3, it is obvious that combining both motion and shape features has contributed to better results. It is important to note that in our experiments, recognition was performed on video clips that were captured from arbitrary views and were not used for training.

Table 3.3 Classification results, recognition rate and standard deviation, of using motion features, shape features, both motion and shape features

	Recognition rate (standard deviation)		
	Using motion features (%)	Using shape features (%)	Using both motion and shape features (%)
Walking	73.9(6.1)	87.0(9.3)	83.7(5.4)
Running	74.1(5.6)	71.7(7.9)	86.7(1.6)
Sitting down	93.7(7.9)	89.6(7.9)	97.9(4.1)
Standing up	89.6(7.9)	91.7(9.6)	93.8(7.9)
Average	79.5(2.2)	82.1(1.2)	88.6(1.6)

Table 3.4 Confusion matrix of using both motion and shape features

	Walking	Running	Sitting down	Standing up
Walking	83.7	16.3		
Running	13.3	86.7		
Sitting down			97.9	2.1
Standing up			6.2	93.8

The second set of experiments was performed on the database used in [58], which includes seven activities (walking, sitting, standing up, bending, getting up, squatting, rising). The video clips in this database have a frame resolution of 352x240 pixels and frame rate of 12-15 frames per second. Each sequence ranges in size from 60 to 80 frames. All the sequences were captured from the profile view and each sequence includes more than one activity. Since the data was captured only from the profile view, we just built one HMM for each activity in this experiment. We used a set of single-activity sequences for training the HMMs. Eleven continuous sequences containing 62 single-activities were used for testing. The length of the sliding window was set to eight based on the results from one training sequence. Our algorithm detected 48 out of the 55 breakpoints between the single activities. This means that the segmentation efficiency is 87.2%. The recognition rate for each activity is listed in Table 3.5.

Table 3.5 Recognition results compared with the algorithm used in [58]

Activity	Recognition rate using our algorithm	Recognition rate using algorithm presented in [58]
Walking	100%	89.66
sitting down	77.8%	76.92
standing up	71.4%	75.00
Bending	73.7%	71.42
getting up	83.3%	73.68
Squatting	77.8%	75.00
rising	77.8%	71.42
Average recog.	80.6%	76.92

3.4 Discussion

For testing the robustness of our features for the different frame rates, we did an experiment for which the results are shown in the Figure 3.11. We trained the activity models using the original video clips with 30 frames per second and then tested walking and running video clips at subsampled frame rates from the original video clips with one frame for every 1, 2, 3, 4, and 5 frames, respectively. From the results, we can see that the recognition rates degrade with the decrease in the frame rates. The degradation with the decrease in the frame rates is slow when we use only the shape features, and is fast when we use the motion features. The degradation rate when using both features is between the two. This is expected because the optical flow, i.e., the motion feature we used, is sensitive to frame rate but shape features are not. From Figure 3.11, we also see the degradation of recognition rate for walking, with the decrease of the frame rate, is smaller than the degradation for running. We think that this is because the activity cycle of running is short, i.e., one cycle of running includes few frames; when the frame rate is decreased, those sampled frames lose a great deal of information. Using both motion and shape features is more robust for different frame rates than just using motion features. Even though the experimental results demonstrate that the difference between the frame rates of the training data and the test data should not be too large for a good recognition rate, a quantized justification for the acceptable difference deserves further exploration in the future.

The computational cost of this algorithm is high. Because the training work is done offline, we disregard the computation efficiency of the training. In the recognition phase, we need to compute $N\times 8$ likelihoods (N is the number of activities); this computation can be fast enough because of the Viterbi algorithm. The main computation burden comes from the feature extraction, which includes background subtraction, optical flow computation and shape features computation. In the background subtraction step, the background is updated each frame. So the computation depends on the size of the background. The smaller the size of the background, the fewer will be the required computations. The computation of optical flow and shape features also depends on the size of the background. In our experiments, we used Matlab for implementation. The code runs at 1-2 frames per second on a 2.4 GHz Pentium processor for 352x240 color images. If we use optimized C code, we expect it to run at 15 frames per second.

The background subtraction algorithm we used is updated frame by frame and is robust to slow changes in lighting. When the environmental conditions do not change intensely, our algorithm is robust. The background subtraction results in our indoor data not being sensitive to the threshold. It does not affect the background subtraction results, i.e., it affects the shape feature to some extent, but it does not affect the motion feature much because it usually does not affect the ROI extraction much. So the final recognition results based on both motion and shape features are not so sensitive to the selection of the threshold. The results of our experiments for indoor scenes show its effectiveness. Because the algorithm we used lacks the ability to get rid of shadows, it is not suitable for outdoor scenes. There is already considerable work on background subtraction for outdoor scenes that has appeared in the past few years. In this chapter we did not focus our attention on background subtraction.

3.5 Conclusion and Future work

In this chapter, we proposed an algorithm for activity modeling and recognition from video clips captured at arbitrary views. Both motion and shape features were used to represent human activities. Based on the combined motion and shape features, a set of HMMs was built for each activity to represent the activity from different views to enable recognizing activities from arbitrary views. In our experiments, we compared the use of only motion features, only shape features and both motion and shape features in building HMMs. The experimental results show 88.6% recognition rate when using both motion and shape features, which is higher than the rate obtained when using only shape features or motion features only. We also presented a voting-based method to segment complex activities. Experimental results show that the segmentation efficiency is 87.2%. The results show that our algorithm is effective. Our future work will focus on combining human tracking with activity recognition in order to recognize activities when multiple people are present.

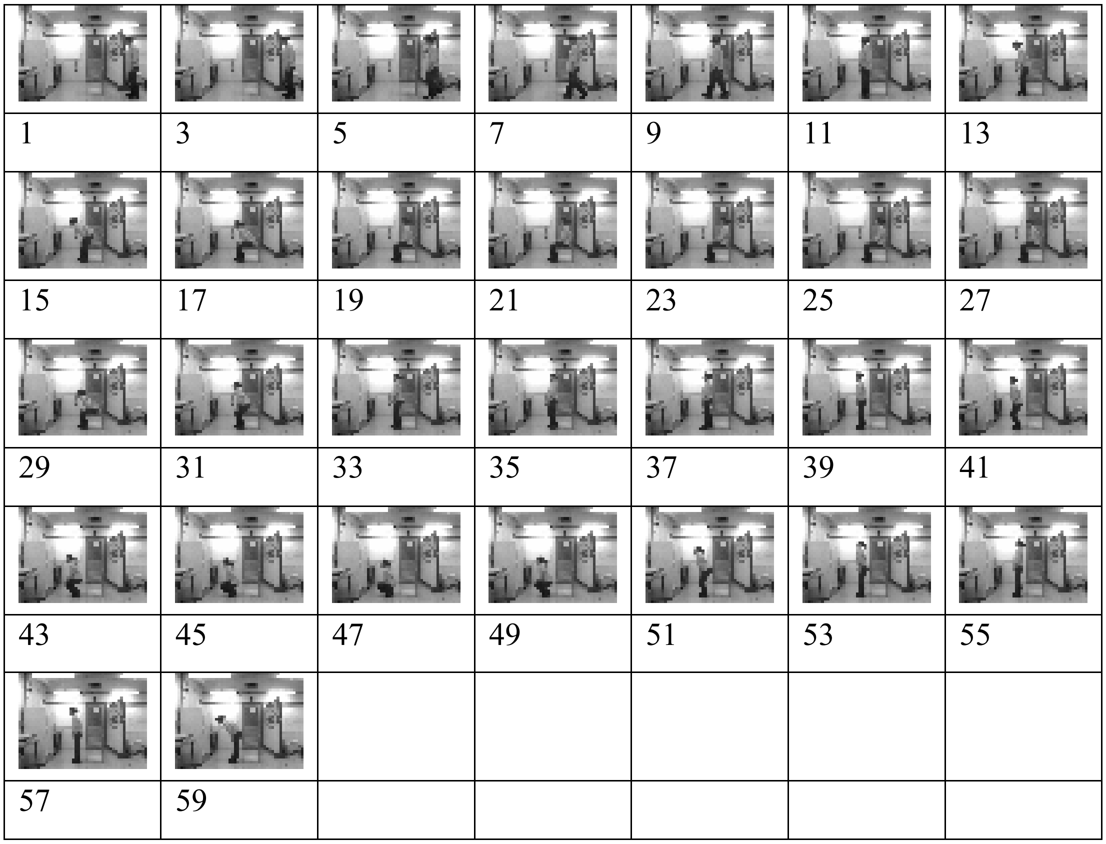


Figure 3.8 An image sequence from the database used in [58]

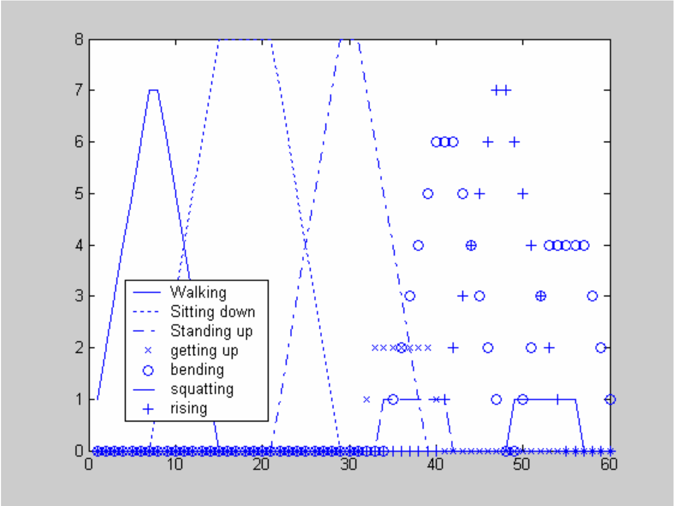
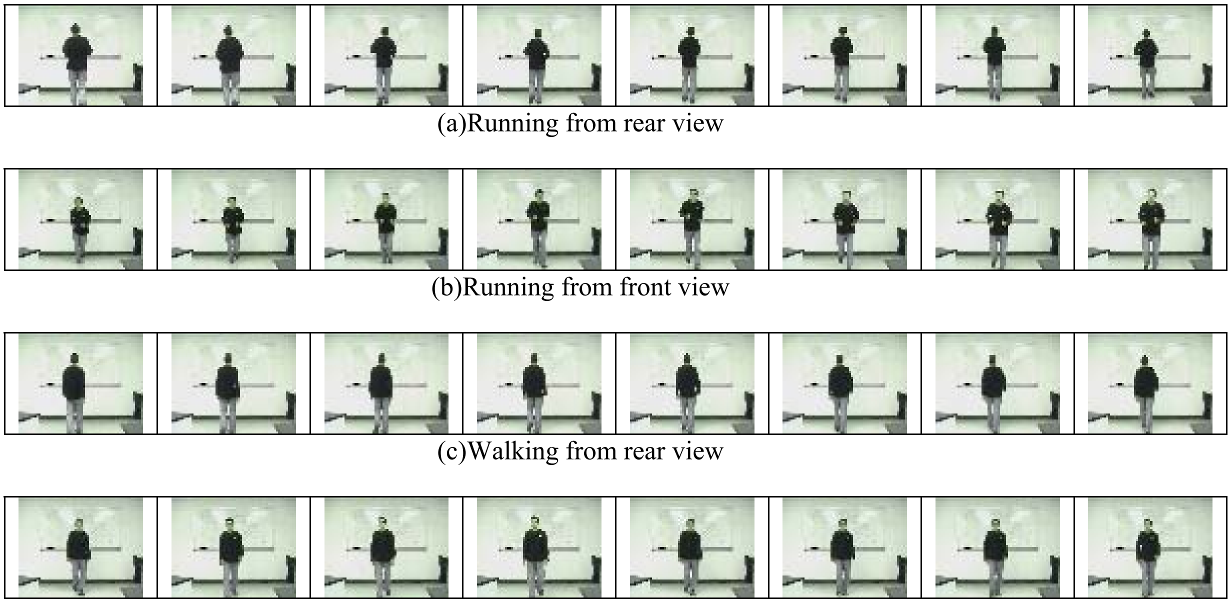


Figure 3.9 Voting results for the sequence shown in Figure 3.8



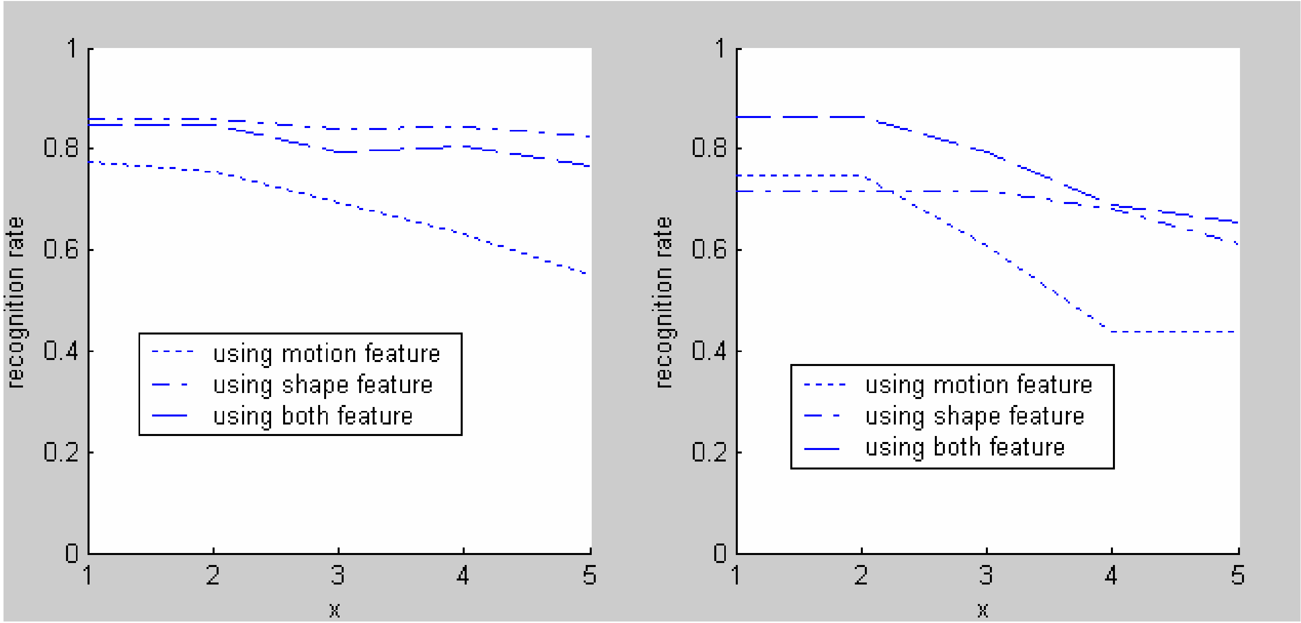
(a)Running from rear view

(b)Running from front view

(c)Walking from rear view

(d)Walking from front view

Figure 3.10 Walking and running from front and rear views



(a) Result for walking (b) Result for running

Figure 3.11 Results of testing the robustness of features using different frame rates

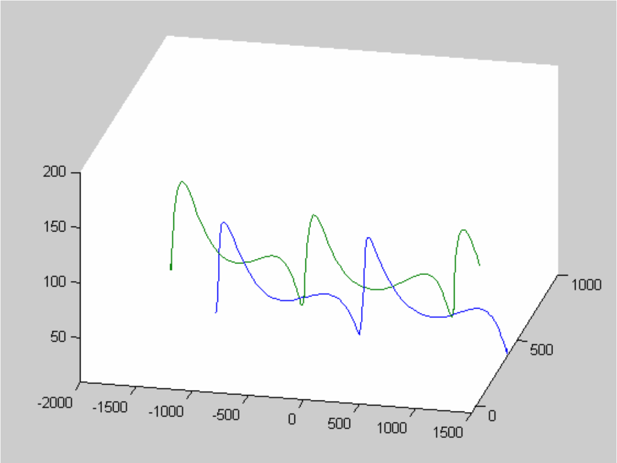
Chapter 4

System framework for pathological gait pattern identification

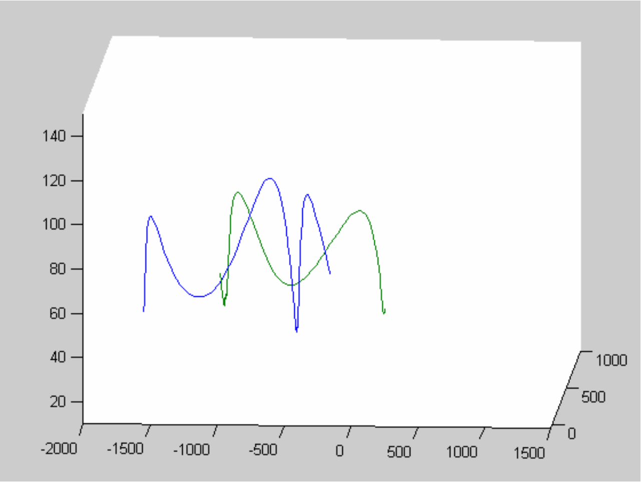
4.1 Gait representation

It is reasonable to assess a patient’s gait by measuring the symmetry of the gait, because in normal individuals the gait patterns with respect to time, distance, and vertical force are fairly symmetrical and only deviate by a small percentage from perfect symmetry [32]. Some methods based on measuring the symmetry of on stance time, swing time, step length, and vertical ground reaction force were developed, such as [33-37]. Based on observation, we find that the symmetry of gait also is reflected in the trajectories of body parts. Figure 4.1(a)(b) shows two pairs of example trajectories of toes, respectively, from a normal person and a patient with knee problem. The two trajectories (shown in figure 4.1 a) from the normal person are clearly similar, but the two trajectories (shown in figure 4.1 b) from the patient are quite different. Here, we describe the symmetry of gait as: that the trajectories of the left body parts are the translation of the trajectories of their corresponding right body parts with small rotation in 3D space.

In this work, we use symmetry measure based on trajectories of body parts as features to represent gait for classification.



(a) From a healthy subject



(b) From a pathological subject

Figure 4.1 3D trajectories of left toe and right toe

4.2 Support vector machines

The Support vector machine (SVM) is a powerful machine learning tool, especially for binary classification problems. It has been successfully used in many applications, such as facial expression classification [14] and text categorization [15]. The core idea of SVM is to find the hyper-plane which is used to separate the two classes in feature space by maximizing the margin between the two classes. Our problem of classifying gaits into normal gaits and pathological gaits may be stated as follows: we have a training data set with input features and classification output

$$\{(x_1,y_1),(x_2,y_2),\ldots,(x_N,y_N)\}$$

Where N is the total number of samples in the training set , x_i's represent the feature vectors, and y_i's have one of two values, either -1 or 1, which denote normal gait and pathological gait, respectively.

The SVM linearly separates the patterns by finding a hyper-plane in the feature space that has the largest margins from the closest feature vectors. The linearly separable case can be represented mathematically as

$$w^T x + b < 0 \text{ for } y_i = -1 \quad (8)$$

$$w^T x + b \geq 0 \text{ for } y_i = 1 \quad (9)$$

where w is the adjustable weight vector and b is the hyper-plane bias.

The equation of the boundary (the hyper-plane) is

$$w^T x + b = 0 \quad (10)$$

In SVM, the optimal values of w and b are defined when the distance to closest feature vectors are maximized. In most real life problems, the data are not linearly separable. This is overcome by mapping the data from the input feature space into another space via a nonlinear kernel function, where the data will be linearly separable in the new space. In our experiments we used three kernels: linear kernel function, Radial Basic Function (RBF) kernel, and a 2nd degree polynomial kernel.

4.3 System framework

Figure 4.2 show the SVM based system framework used to identify pathological gait.

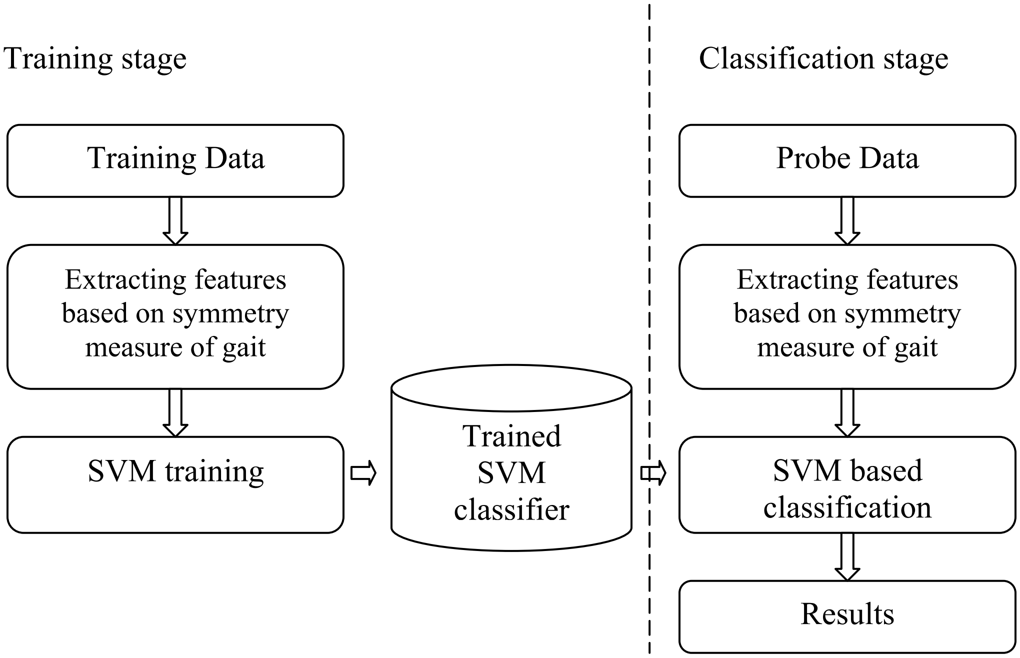


Figure 4.2 SVM based pathological gait classification framework

Chapter 5

Pathological gait pattern identification using 3D data

In this chapter, we describe a gait analysis system, based on symmetry, for the purpose of identifying normal and pathological gait patterns [104]. The main objective of the research is to develop a classification system that is capable of differentiating between gait patterns of individuals who have had total knee replacement surgery and normal healthy individuals. We represent gait by a feature vector that is obtained from 3D motion data which contain information of the whole body's movement. Then, support vector machines are trained for classification. The results show that the algorithm is effective. The rest of the chapter is organized as follows: In Section 5.1, we describe the data and the setup that was used in our work. Our algorithm, including feature extraction and the support vector machine (SVM) classifier, is described in Section 5.2. In Section 5.3, we present the experimental results. Finally, we conclude the chapter in Section 5.4.

5.1 Motion-capture Data

5.1.1 Subjects

In this study, we collected data from thirteen subjects, seven patients and six normal. All the patients had their knee replaced by either a metallic or an allograft knee. The normal subjects are healthy male students. All the subjects were informed of the procedures and signed an informed consent approved by the University of Miami Institutional Review Board.

5.1.2 Data collection setup

A motion-capturing system composed of eight M-Cam cameras (Vicon 512, Vicon Motion Systems, Lake Forest, CA)[99] that record the spatial positions of a set of markers were positioned on the body throughout the whole gait cycles. The cameras captured a volume large enough to contain two gait cycles of the subject during any trial. Dynamic calibration was performed using a set of 50mm diameter reference markers ('wand'). Figure 5.1 shows the camera positions within the construction volume.

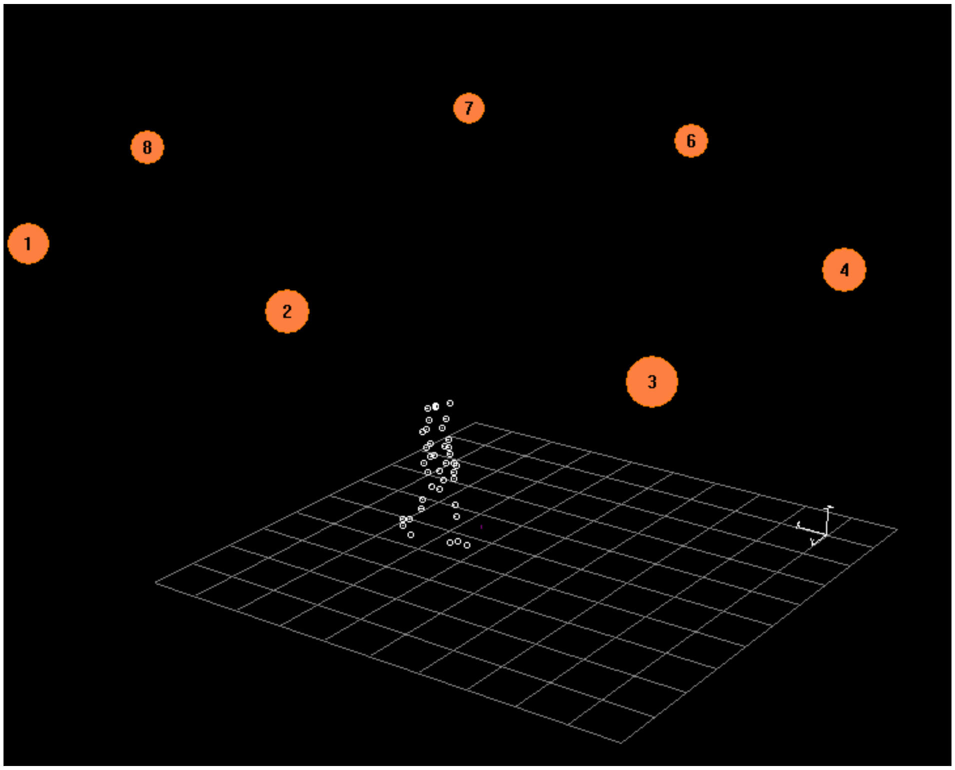


Figure 5.1 Camera positions within the reconstruction volume

5.1.3 Procedure

The data was collected in collaboration with the Medical School of the University of Miami. The subjects were acquainted with the testing procedures and were prepared for the data collection. This preparation included locating reflective markers on different body landmarks of the subjects according to the Modified Helen Hayes (MHH) model [89] (shown in figure 5.2). Although we did not use the information from the full set of body markers, there were thirty-nine 25mm in diameter reflective markers that were placed on the subject's body to capture the motion parameters. The placement of those markers is shown in Figure 5.2, where most of the markers are placed on symmetrical positions of the body, such as RSHO and LSHO, and some markers are not, such as RTIB and LTIB. Because we were interested in assessing the walking gait based on motion symmetry, we used the data from symmetrical markers for our purpose. The markers we chose are RFHD, LFHD, RBHD, LBHD, RSHO, LSHO, RELB, LELB, RWRA, LWRA, RWRB, LWRB, RASI, LASI, RPSI, LPSI, RFIN, LFIN, RKNE, LKNE, RANK, LANK, RTOE, and LTOE. We also used the data from marker CLAV in our work. The reason is explained later in this chapter.

5.2 Method

5.2.1 Feature extraction

The 3D motion-captured data consists of a set of 3D trajectories of markers placed on a subject's body. These trajectories represent the 3D spatial positions of the corresponding parts of the body across time. Therefore, the question becomes how to measure gait symmetry by comparing corresponding trajectory pairs, such as the trajectories of the markers labeled RTOE and LTOE. We represent the 3D trajectory *T*(*x, y, z, n*) as three trajectory components *X*(*n*), *Y*(*n*), *Z*(*n*), which are three orthogonal components of the trajectory *T*(*x, y, z, n*) in the *x, y,* and *z* directions, respectively. Here, *x* is the person's walking direction, *z* is the vertical axis parallel to the person's body, and *y* is the axis that is perpendicular to *x* and *z*. We measure the difference between a pair of 3D trajectories, e.g. T_L (*x, y, z*) and T_R (*x, y, z*), by measuring the difference between each of the three corresponding components, e.g., (X_L (*n*), X_R (*n*)), (Y_L (*n*), Y_R (*n*)) and (Z_L (*n*) and Z_R (*n*)).

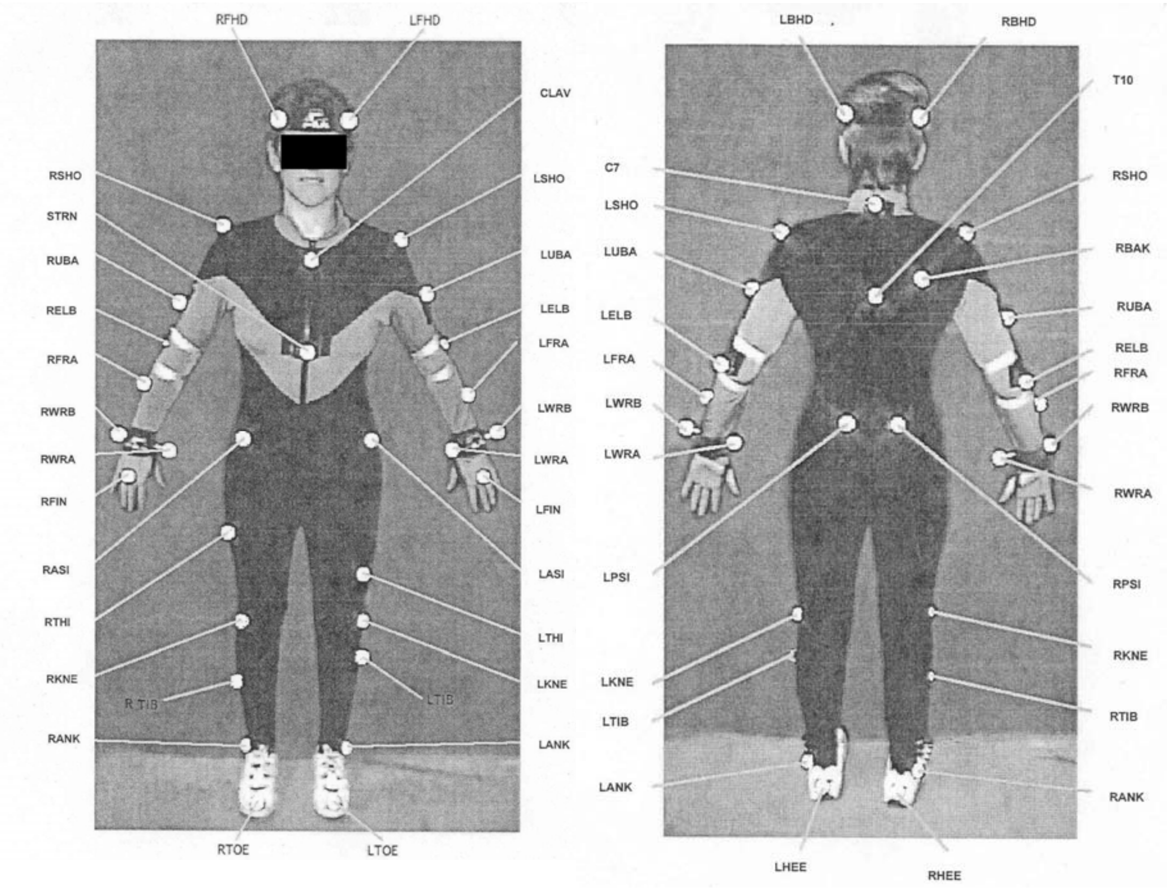


Figure 5.2 Marker set used in the current study

5.2.1.1 Normalization

The ranges for the coordinate values of the trajectory components are different. For example, the *z* components of the trajectories of the feet usually range from 20mm to 140mm in our reference coordinate system, but the *z* components of the trajectories of the head usually range from 1500mm to 1800mm, depending on the person's stature. To eliminate the effect of the difference in the range between different trajectory components on the gait symmetry assessment, we normalize those trajectory components before further processing.

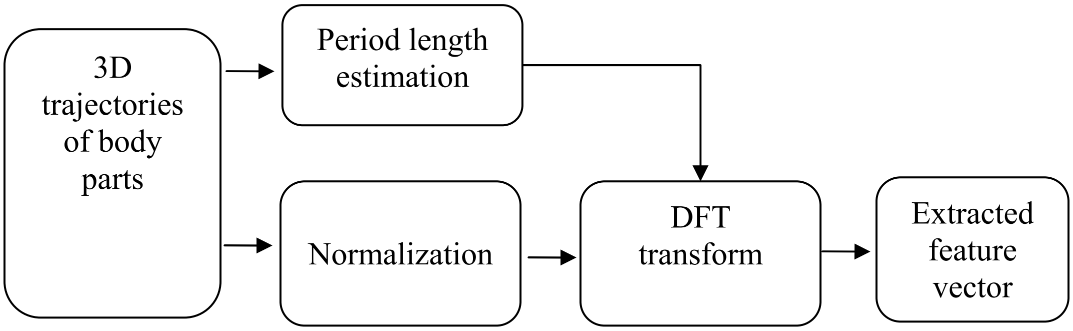


Figure 5.3 Block diagram of the feature extraction processing

.

The normalization process for the trajectory components in the *x* direction is different from the normalization of the trajectory components in both the *y* and *z* directions. Since people walk along the *x* direction, all trajectory components in the x direction continuously increase with time, i.e., the frame number *n*, which is not the case for the trajectories in the *y* and *z* directions. Therefore, all x components are not periodic as in the example shown at the top of Figure 5.4. We transform the $X_L(n)$, $X_R(n)$, i.e., the x components, in order to make it periodic as follows:

$$\begin{cases} \widetilde{X}_L(n) = X_L(n) - X_0(n) \\ \widetilde{X}_R(n) = X_R(n) - X_0(n) \end{cases} \quad (1)$$

where $X_0(n)$ is the x trajectory component of CLAV, the marker on the jugular notch where the clavicles meet the sternum. Then, we normalize $\widetilde{X}(n)$ by

$$\begin{cases} \overline{X}_L(n) = (\widetilde{X}_L(n) - \widetilde{X}_{\min})/(\widetilde{X}_{\max} - \widetilde{X}_{\min}) \\ \overline{X}_R(n) = (\widetilde{X}_R(n) - \widetilde{X}_{\min})/(\widetilde{X}_{\max} - \widetilde{X}_{\min}) \end{cases} \quad (2)$$

$\widetilde{X}_{\max}$ and $\widetilde{X}_{\min}$ are the minimal and maximal values of $\{\widetilde{X}_L(n), \widetilde{X}_R(n)\}$. The same transformation and normalization steps are applied to the x components of all the trajectories. For the components in the y and the z directions, we normalize them directly by

$$\begin{cases} \overline{Y}_L(n) = (Y_L(n) - Y_{\min})/(Y_{\max} - Y_{\min}) \\ \overline{Y}_R(n) = (Y_R(n) - Y_{\min})/(Y_{\max} - Y_{\min}) \end{cases} \quad (3)$$

$$\begin{cases} \overline{Z}_L(n) = (Z_L(n) - Z_{\min})/(Z_{\max} - Z_{\min}) \\ \overline{Z}_R(n) = (Z_R(n) - Z_{\min})/(Z_{\max} - Z_{\min}) \end{cases} \quad (4)$$

where $Y_{\min}$ and $Y_{\max}$ are the minimal and maximal values of $\{Y_L(n), Y_R(n)\}$, and similarly for Z. Figure 5.5 shows one example of the normalization of the z components of LTOE and RTOE of a normal person. Figure 5.6 shows one example of the normalization of the *z* components of LTOE and RTOE of a patient with left knee replacement. From the figures, we can see the asymmetry between the left and the right trajectories.

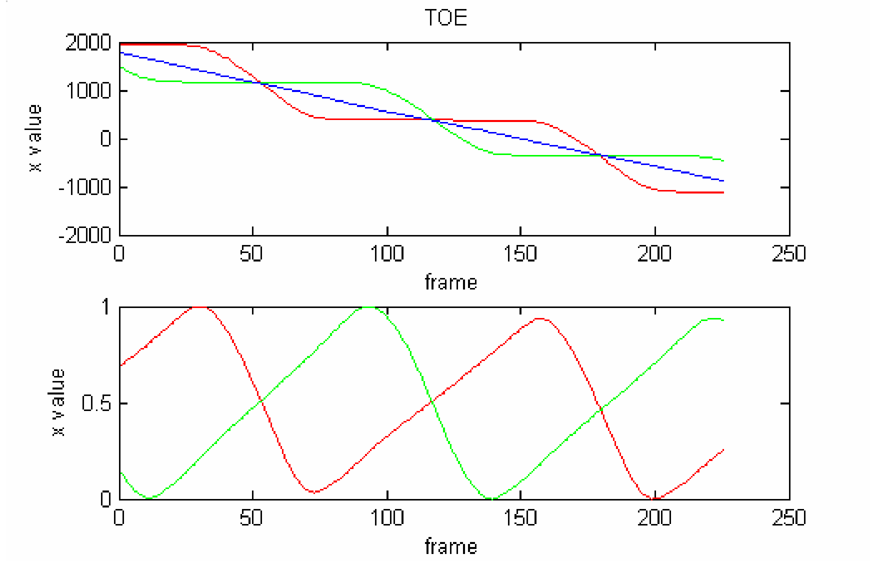


Figure 5.4 Plots of x components of LTOE and RTOE of a normal person before(upper) and after(bottom) normalization. The red denotes the left foot and the green denotes the right foot. The blue denotes the *x* trajectory component of the CLAV marker on the jugular notch. The upper figure and the lower figure show the *x* trajectory components before and after normalization.

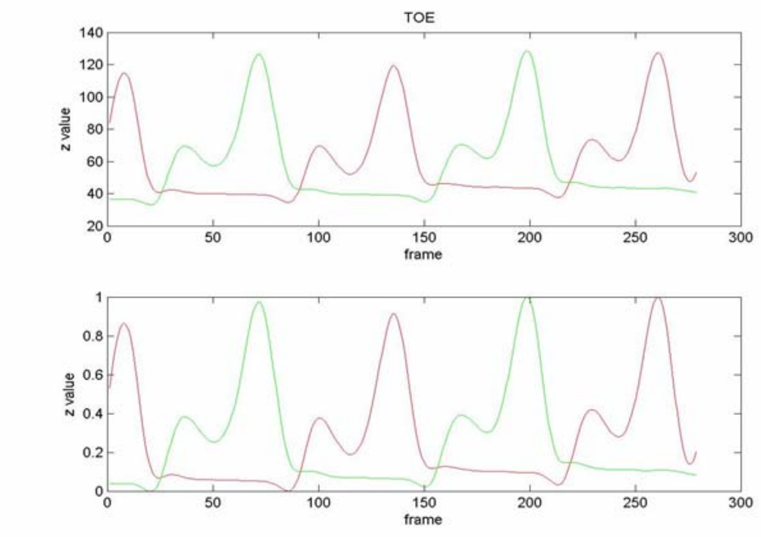


Figure 5.5 Plots of z components of a normal person before (upper) and after (bottom) normalization. The red denotes the left foot and the green denotes the right foot.

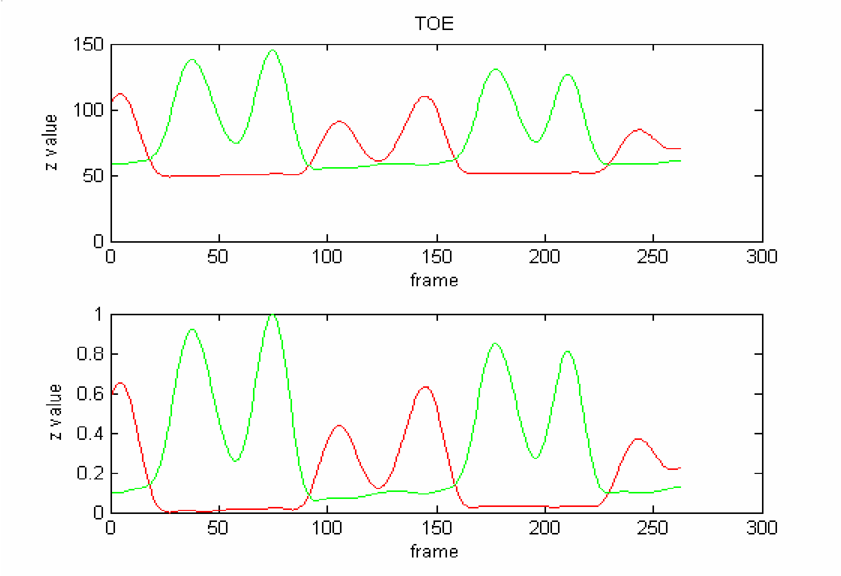


Figure 5.6 Plots of z components of LTOE and RTOE of a patient with left knee replacement before (upper) and after (bottom) normalization. The red denotes the left foot and the green denotes the right foot.

5.2.1.2 Discrete Cosine transform

There are some methods for measuring the difference between two one-dimensional trajectories, such as the method used in [13]. In this method, authors obtain a measure of difference between two trajectories through searching and finding the optimal spatial translation and temporal shift. The drawback of this algorithm is that the optimization process is computationally expensive. According to the nature of our data, most of the body trajectories are periodic because the movement of a person is periodic while walking. Therefore, we measure the difference between two one-dimensional trajectories by comparing the absolute values of their DFT coefficients without the need to consider the spatial shift between the two trajectories.

Let $f(k)$ represent a discrete time signal, and let $F(n)$ represent its discrete cosine transform. The Discrete Fourier Transform is given by

$$F(m) = \sum_{k=0}^{N-1} f(k) \exp[-\frac{2\pi i}{N} km] \qquad (5)$$

where N is the period. Assume that $F_{l,i}(m)$ and $F_{r,i}(m)$, $m = 1,\ldots,N$ are the discrete cosine transforms of a pair of trajectory components, where the component is denoted by i, in the same direction and come from a pair of markers set on two symmetrical positions of the body, l denotes the trajectory component of a left body part, r denotes the trajectory component of a right body part. The difference between the pair of trajectory components, denoted by i, can be represented as

$$d_i = [d_i^1 \quad d_i^2 \quad \cdots \quad d_i^M] \qquad (6)$$

where $M \leq N$ and

$$d_i^m = \frac{1}{N} \| F_{l,i}(m) | - | F_{r,i}(m) \| \qquad (7)$$

Therefore, we form a vector D = $(d_1, d_2, \ldots, d_L)$ to assess the symmetry of a person's gait, where L is the number of trajectory component pairs and is equal to the number of marker pairs multiplied by three.

5.2.1.3 Determining the period length for each trajectory

In order to use the DFT, we need to specify the number of samples N that constitute a period. One way is to compute N for each trajectory component pair. But actually, the movements of all body parts have the same period and equal to the period of walking cycle. It is reasonable to use the period of one walking cycle as the period N for all trajectories. There are already some methods for computing the duration of a gait cycle from video data [17][18]. Here we present a simple and robust algorithm to detect the duration of the gait cycle. In the process of walking, when the two feet are on the ground, the distance between them is the largest. Starting from this phase, the distance decreases from a maximum to zero and then increases from zero to a maximum in a half cycle. Figure 5.7 shows an example plot of the distance between two feet with respect to the video frame numbers. We compute this distance using the x components of the marker trajectories of the left foot and the right foot.

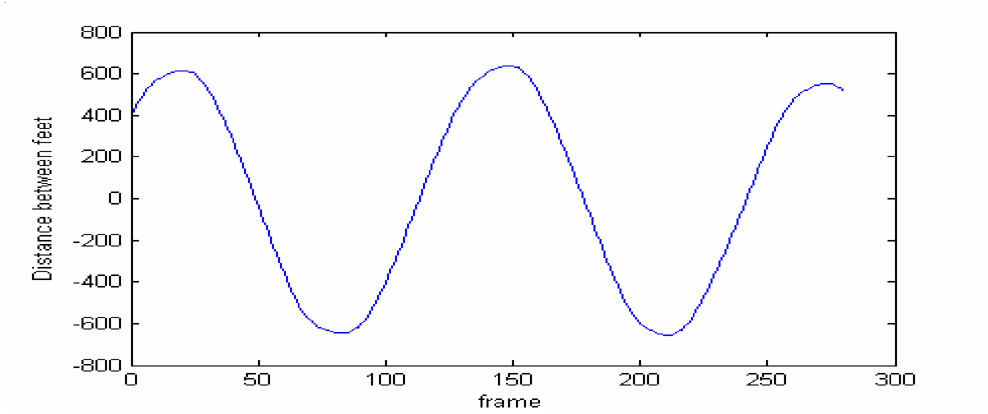


Figure 5.7 Distance between LTOE and RTOE in x direction. From this plot we can see that the period of walking is around 143 frames.

5.2.2 SVM classifier used in experiments

LIBSVM library [96] was employed in the experiments. In our experiments we used three kernels: linear kernel function, Radial Basic Function (RBF) kernel, and a 2nd degree polynomial kernel. The Parameters for the three different classifiers are list in table 5.1.

What needs to be mentioned is that before applying SVM, we linearly scaled each dimension of feature vector to the range [0, 1]. The main reason is to avoid attributes in greater numeric ranges from dominating those in smaller numeric ranges. Of course, we have to use the same method to scale testing data before testing.

Table 5.1 Parameters of three classifiers

	Degree	Gamma	Coefficient	C	epsilon	SVM algorithm
Linear	1	1	1	1	0.001	Nu-SVC
RBF	1	2	1	1	0.001	Nu-SVC
Polynomial	2	1	1	1	0.001	Nu-SVC

5.3 Experimental results

We applied our method of symmetry assessment using the SVM in order to classify normal gait and pathological gait. The experiments were performed on a database containing 72 walking sequences for 13 subjects. The database included sequences from six healthy people and seven patients who had knee replacement. We split all the data into two sets, one with 39 sequences for training and the other one with 33 sequences for testing.

Because all the patients had problems with their knees, we thought to use only data from the lower body parts. Therefore, we compared the results of classifying the subjects into subjects with normal and pathological gaits using either all the data or just the data from the lower body parts. Therefore, we performed two sets of experiments. In the first set of experiments, we used data from all the markers for training and testing, and we compared the results from three implementations of support vector machines for classification, i.e., using linear kernel function, using RBF kernel function, and using the second degree polynomial kernel. We also experimented with using a different number of the DFT coefficients, i.e., *M* described in section 3.1.2. Usually, the walking period of a person is from 125 to 160 frames under our system's frame rate, i.e., N is from 125 to 160. Most of the higher DFT coefficients of the trajectories are zero, so we just need to keep the first few coefficients. We found that in most cases, $d_i^m = 0$, when $m > 9$. So we experimented with different *M* values from 1 to 9. The results for these tests are shown in figure 5.8.

In the second set of experiments, we just used the data of the markers from the lower body parts: LTOE, RTOE, LHEE, RHEE, RANK, LANK, RKNE, LKNE, RASI, and LASI. We also used the three SVM classifiers with different kernels and experimented with different *M* values from 1 to 9. The results for these tests are shown in figure 5.9.

5.4 Discussion and Conclusions

From figure 5.8, we can see that the classification results of the three SVM classifiers, with different kernels, are close to each other. The results of using the RBF kernel and the 2nd degree polynomial kernel are a somewhat better than the result from the linear kernel. A 93.9% classification rate was obtained when $M > 4$, which means that two out of 33 testing sequences were misclassified. Table 2 shows the number of correct classifications for each class using a different number of the DFT coefficients for the SVM classifier with RBF kernel. It shows that when $M > 4$, the feature vector has a good discrimination power to classify normal gait and pathological gait.

Figure 5.9 shows the classification rate curves obtained when using only data from the markers on the lower body parts. We can see that the classification results are much worse than the results obtained when using the data from all the markers. In this set of experiments the classifiers with RBF kernel and 2nd degree polynomial kernel gave better results than the classifier with linear kernel. The experiment shows that we lose some important information when we use only data from the lower body parts.

Table 5.2 Experimental results obtained when using RBF kernel

M	Using all data		Using leg data	
	Normal (correct /total num)	Pathological (correct /total num)	Normal (correct /total num)	Pathological (correct /total num)
1	5/16	12/17	7/16	15/17
2	9/16	15/17	5/16	16/17
3	14/16	14/17	12/16	15/17
4	15/16	15/16	8/16	15/17
5	14/16	17/17	6/16	15/17
6	15/16	16/17	11/16	15/17
7	15/16	16/17	10/16	15/17
8	15/16	16/17	12/16	16/17
9	15/16	16/17	12/16	15/17

From the experimental results, we see that assessing the gait symmetry based on feature vectors obtained from the 3D trajectories is accurate in differentiating normal gait from pathological gait. We also found that SVM classifiers with RBF kernel and 2nd degree polynomial kernel perform better than the SVM classifier with linear kernel.

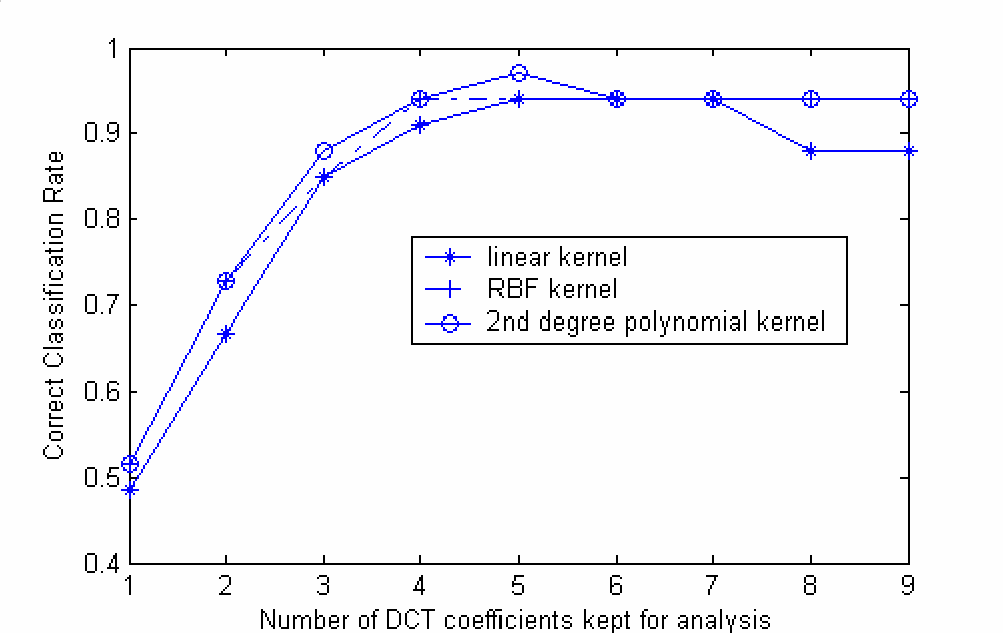


Figure 5. 8 Experimental results using data from all the markers

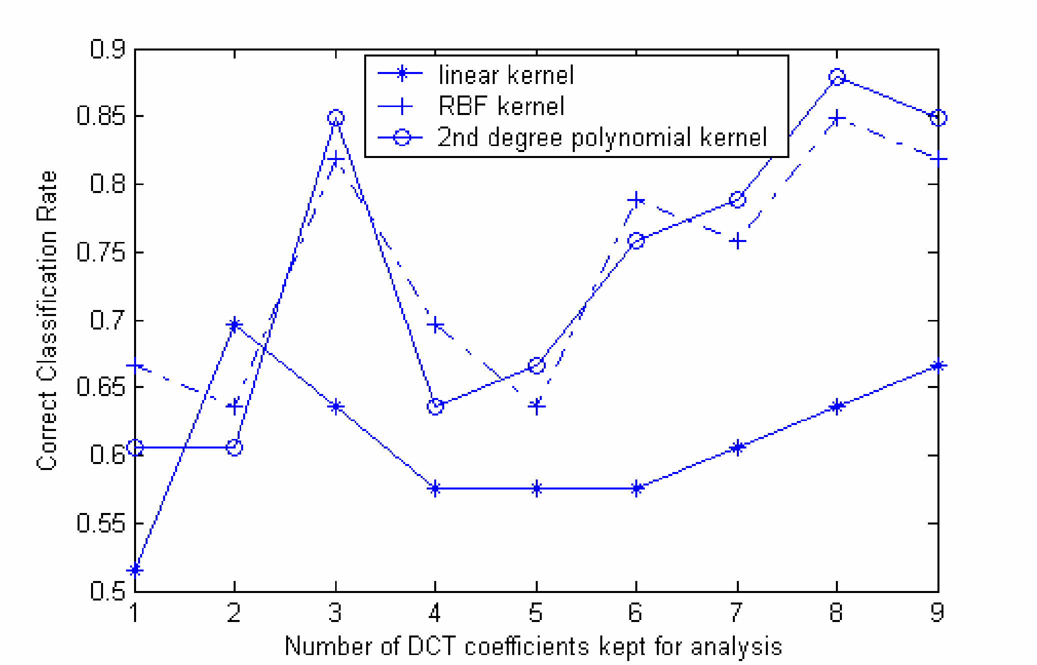


Figure 5.9 Experimental results using data from markers of the lower body parts

Chapter 6

Pathological gait pattern identification from video data

In this chapter, we present an algorithm [105] that extends our work in the previous chapter for identifying pathological gait pattern from video.

In chapter 5, we presented a DFT based symmetry measure method using 3D trajectories of body parts. But it is very difficult to extend the DFT based approach to work with video data. In order to assess symmetry from video data, there are two major issues that need to be dealt with. The first issue is extracting trajectories of the different body parts. Tracking is a particularly important issue in human motion analysis, where in our case it is important in order to extract features that can be used to assess symmetry. Even though object tracking in video streams has been a popular topic in the field of computer vision for many years and many methods have been presented [22-24][94-95], accurate tracking is still a challenging problem, especially for tracking human body parts such as hands, face, and feet. The second issue is incomplete data due to self-occlusion. It is difficult to obtain complete trajectories of hands or feet from video captured from a single profile view because of the self-occlusion that happens while the subject walks. This means that the DFT based feature presented in chapter 4 cannot be used to represent symmetry of gait. Therefore, a new method is needed to deal with the incomplete data.

A silhouette or contour is relatively easy to extract from the image. We propose a model based body part tracking algorithm to track the 2D contour. The geometric structure of the human body is represented as a 2-D contour. A set of contour models of the body during different phases of the gait of subjects was selected as models and stored in a database. The different body parts are manually labeled for all these models. Given a new contour of a subject that is extracted from a frame, a classification algorithm is used to find the most similar stored model in the database. Then, a matching algorithm is used to match the extracted contour and the most similar model to find the corresponding body parts. We use this method to find the body parts in each frame and obtain the trajectories of these body parts.

Because of self-occlusion, we will not be able to obtain the complete trajectories of all body parts. One solution is to use methods such as interpolation or Kalman filter to estimate the lost trajectory segments. But this interpolated data is very inaccurate. Here, we developed another method to deal with the incomplete data. At first, we develop a mathematical representation of the symmetry assumption of gait in the 3D space. Then, under the projective camera model, we present a measure of symmetry in the 2D image plane.

6.1 Symmetry measure in 2D projected plane

6.1.1 Symmetry representation in 3D space

According to the symmetry assumption that the movement of the left and the right parts of the body is symmetrical, we assume that the trajectories of the left body parts are the same as the translation of the trajectories of their corresponding right body parts with a small rotation in the 3D space.

Therefore, this symmetry relation can be represented as:

$$\begin{bmatrix} s_{x,1}(n) \\ s_{y,1}(n) \\ s_{z,1}(n) \end{bmatrix} = \begin{bmatrix} 1 & \alpha & -\gamma \\ -\alpha & 1 & \beta \\ \gamma & -\beta & 1 \end{bmatrix} \begin{bmatrix} s_{x,2}(n) \\ s_{y,2}(n) \\ s_{z,2}(n) \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \qquad n = 1 \ldots L \qquad (1)$$

Where ($s_{x,1}(n)$, $s_{y,1}(n)$, $s_{z,1}(n)$) and ($s_{x,2}(n)$, $s_{y,2}(n)$, $s_{z,2}(n)$) , $n = 1 \ldots L$, is a pair of trajectories that have already been registered. Registration means to establish the correspondence between the points of the two trajectories. L is the length of one trajectory. Subscripts 1 and 2 denote the trajectory corresponding to the left and the trajectory corresponding to the right side of the body, respectively.

The above equation represents the relationship between two symmetrical trajectories in the 3D space. Since our goal is to measure the symmetry of gait using video data, we need to represent the relationship between two 3D symmetrical trajectories based on their projections in 2D. In the following sections, we will first introduce the camera model which is used for projection. Then, we will introduce the representation of the symmetric trajectories in 2D and present the corresponding measure of symmetry.

6.1.2 The Camera Model

Two camera models are commonly used to represent projection from 3D to 2D: affine model and projective model. The affine model requires that the distance between the camera and the object to be large compared to the size of the object. In our case, this means that the camera should be placed far from the walking person. This will induce relatively large errors when we extract the trajectories from video. Therefore, in this work we use a projective camera model. We set the camera reference frame to coincide with the world's reference frame. The projection of a 3D point ($s_{x,}$, s_y, s_z) can be described by

$$\begin{cases} u = \dfrac{fs_x + s_z p_x}{s_z} \\ v = \dfrac{fs_x + s_z p_y}{s_z} \end{cases} \qquad (2)$$

Where f is the focal length of the camera and p_x and p_y are the offsets of the image reference frame center.

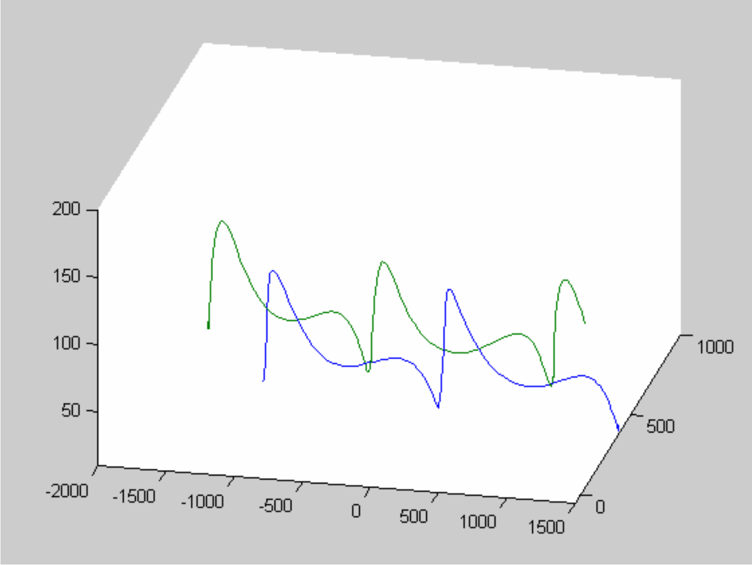


Figure 6.1 A pair of 3D trajectories of markers placed on left toe and right toe

6.1.3 Relationship between the 2D projections of Symmetrical 3D trajectories

Assume $I_1 = \begin{bmatrix} u_1(1) & u_1(2) & \dots & u_1(L) \\ v_1(1) & v_1(2) & & v_1(L) \end{bmatrix}$ and $I_2 = \begin{bmatrix} u_2(1) & u_2(2) & \dots & u_2(L) \\ v_2(1) & v_2(2) & & v_2(L) \end{bmatrix}$ are two sequences that represent the projections of a pair of 3D trajectories ($s_{x,1}(n)$, $s_{y,1}(n)$, $s_{z,1}(n)$) and ($s_{x,2}(n)$, $s_{y,2}(n)$, $s_{z,2}(n)$) , n = 1 … L. Using equations (1) and (2), the relationship between the 2D projections of two 3D symmetrical trajectories can be represented as:

$$\begin{cases} u_1(n) - p_x = f\dfrac{u_2(n) + \alpha\ v_2(n) + T_u(s_{z,2}(n))}{\gamma\ u_2(n) - \beta\ v_2(n) + T_w(s_{z,2}(n))} \\ v_1(n) - p_y = f\dfrac{-\alpha u_2(n) + v_2(n) + T_v(s_{z,2}(n))}{\gamma\ u_2(n) - \beta\ v_2(n) + T_w(s_{z,2}(n))} \end{cases} \qquad n = 1 \dots L. \qquad (3)$$

Where

$$\begin{cases} T_u(s_{z,2}(n)) = -f\gamma - f\ t_x / s_{z,2}(n) - p_x - p_y\alpha \\ T_v(s_{z,2}(n)) = -f\gamma - f\ t_y / s_{z,2}(n) + p_x\alpha - p_y \\ T_w(s_{z,2}(n)) = -f\gamma - f\ t_z / s_{z,2}(n) - p_x\gamma + p_y\beta \end{cases}$$

The details of deriving Eq. 3 are presented in the Appendix I.

When the camera is placed on the side of the walking person such that the person walks along a line parallel to the 2D image plane, the $s_{z,2}(n)$ does not change significantly and as a result the $f\ / s_{z,2}(n)$ can be approximated by a constant. Therefore, we have $T_u^{'} \approx T_u^{'}(s_{z,2}(n))$, $T_v^{'} \approx T_v^{'}(s_{z,2}(n))$ and $T_w^{'} \approx T_w^{'}(s_{z,2}(n))$ for n = 1 … L., Eq. (3) can be approximated as follows

$$\begin{cases} u_1(n) - p_x = f\,\dfrac{u_2(n) + \alpha\, v_2(n) + T_u^{'}}{\gamma\, u_2(n) - \beta\, v_2(n) + T_w^{'}} \\ v_1(n) - p_y = f\,\dfrac{-\alpha\, u_2(n) + v_2(n) + T_v^{'}}{\gamma\, u_2(n) - \beta\, v_2(n) + T_w^{'}} \end{cases} \qquad n = 1 \ldots L. \quad (4)$$

Let $\eta(n) = (\gamma\, u_2(n) - \beta\, v_2(n) + T_w^{'}) / f$. For a pair of symmetrical 2D trajectories I_1 and I_2, we would have the relationship:

$$\begin{cases} \eta(n)\cdot(u_1(n) - p_x) = u_2(n) + \alpha v_2(n) + T_u^{'} \\ \eta(n)\cdot(v_1(n) - p_y) = -\alpha u_2(n) + v_2(n) + T_v^{'} \end{cases}, \quad n = 1 \ldots L. \quad (5)$$

6.1.4 Measuring **Symmetry**

By arranging the $2L$ equations from Equation (5), the symmetry of two trajectories in the 2D plane can be represented as:

$$AX = b \qquad (6)$$

Where A is a $2L \times (L+3)$ dimensional matrix composed of the coordinates of points in both I_1 and I_2, b is a $2L \times 1$ dimensional vector composed of coordinates of points in I_2 and X is an unknown $(L+3)\times 1$ dimensional vector containing information about the translation and rotation between the two trajectories, the camera parameters and the distance between the object and the camera, i.e.,

$$A = \begin{bmatrix} u_1(1) - p_x & 0 & \cdots & 0 & -v_2(1) & 1 & 0 \\ 0 & u_1(2) - p_x & \cdots & 0 & -v_2(2) & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & u_1(L) - p_x & -v_2(L) & 1 & 0 \\ v_1(1) - p_y & 0 & \cdots & 0 & u_2(1) & 0 & 1 \\ 0 & v_1(2) - p_y & \cdots & 0 & u_2(1) & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & v_1(L) - -p_y & u_2(L) & 0 & 1 \end{bmatrix}_{2L \times (L+3)} \qquad (7)$$

$$X = \begin{bmatrix} \eta(1) \\ \eta(2) \\ \vdots \\ \eta(L) \\ \alpha \\ T_u^{'} \\ T_v^{'} \end{bmatrix}_{(L+3) \times 1} \quad \text{and} \quad b = \begin{bmatrix} u_2(1) \\ u_2(2) \\ \vdots \\ u_2(L) \\ v_2(1) \\ v_2(2) \\ \vdots \\ v_2(L) \end{bmatrix}_{2L \times 1}$$

This is a linear system. When the two trajectories are symmetrical, given the computed *A* and *b*, there is one solution for *X* when *L* > 3. But because of numerical errors and noise there may be no exact solution especially in the case where the two trajectories are not symmetrical.

Therefore, for two arbitrary trajectories I_1 and I_2, the question is how to evaluate the symmetry between I_1 and I_2 based on the computed *A* and *b*. In this work, we use the minimum residual of the linear system, i.e., *dist*, to measure the symmetry of two trajectories,

$$dist = \min_{X \in R^{L+3}} \{ \frac{1}{L} \| AX - b \| \} \qquad (8)$$

The reason for this is explained as follows:

At first, given A and b, the vector X is estimated as

$$X_{LS} = \arg\min_{X \in \mathrm{R}^{L+3}} \{\|AX - b\|\}$$

Then, we assume $\bar{I}_1 = \begin{bmatrix} \bar{u}_1(1) & \bar{u}_1(2) & \dots & \bar{u}_1(L) \\ \bar{v}_1(1) & \bar{v}_1(2) & & \bar{v}_1(L) \end{bmatrix}$ is a projected trajectory that is approximate to I_1 and the two trajectories $\bar{I}_1$ and I_2 are exactly symmetrical:

$$\begin{cases} u_1(n) = \bar{u}_1(n) + \varepsilon_n \\ v_1(n) = \bar{v}_1(n) + \sigma_n \end{cases} \quad i = 1 \ldots L \qquad (9)$$

and

$$\bar{A} X_{LS} = b$$

Where $\bar{A}$ is a $2L \times (L+3)$ dimensional matrix composed of the coordinates of points in both $\bar{I}_1$ and I_2.

Furthermore, from eq. (9), we have the relation between $\bar{A}$ and A:

$$\bar{A} = A + \Delta$$

where

$$\Delta = \begin{bmatrix} \varepsilon_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & \varepsilon_2 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \varepsilon_L & 0 & 0 & 0 \\ \sigma_1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \sigma_L & 0 & 0 & 0 \end{bmatrix}$$

Therefore, the symmetry measure, *dist,* between I_1 and I_2 can be represented as:

$$
\begin{aligned}
dist &= \min_{X \in R^{L+3}} \frac{1}{L}\{\|AX-b\|\} \\
&= \frac{1}{L}\|AX_{est}-b\| \\
&= \frac{1}{L}\|\bar{A}X_{est}+\Delta X_{est}-b\| \\
&= \frac{1}{L}\|\Delta X_{est}\| \\
&= \frac{1}{L}\sum_{i=1}^{L}\eta(i)(\varepsilon_i{}^2+\sigma_i{}^2)
\end{aligned}
\qquad (10)
$$

From (10), we can see that the measure *dist* gives a weighted sum of errors by which the projected trajectory I_1, must be perturbed in order to make it symmetrical to the projected trajectory I_2. The measure *dist* can be readily solved by the method based on QR factorization of *A* [93]. The details of computing *dist* are presented in Appendix II. Therefore, from *N* pairs of projected body part trajectories, we can obtain a vector *D* = [$dist_1$, $dist_2$, ..., $dist_N$]^T as a feature vector to represent the gait. $dist_i$ is the symmetry measure for trajectory pair *i*, where the pair consists of two corresponding trajectories coming from the left and right sides of the human body.

6.2 Trajectory registration

In Section 6.1, we assumed that the two corresponding trajectories are already registered. Therefore, before composing matrix *A* and vector *b*, we need to register the two trajectories.

Assume the points on 2D trajectories are represented as

$$\bar{s}_1(n) = \begin{bmatrix} \bar{s}_{x,1}(n) \\ \bar{s}_{y,1}(n) \end{bmatrix} \qquad n = 1 \ldots K_1 \ \text{ and } \ \bar{s}_2(j) = \begin{bmatrix} \bar{s}_{x,2}(j) \\ \bar{s}_{y,2}(j) \end{bmatrix} \qquad j = 1 \ldots K_2$$

Because the trajectories of body are periodic during walking and the phase difference between corresponding trajectories of a left and a right parts of the body is half a period, we use the method described in Section 5.2.1.3 to obtain the length of the gait's period, *N*. The point $\bar{s}_1(n)$ corresponds to $\bar{s}_2(n + N/2)$. To overcome the effect induced by the estimation error of *N*, we iteratively register a pair of trajectories, *i,* multiple times. Each registration is based on a different length of the gait's period from *N-m* to *N*+*m*. Therefore, a set of the symmetry measures for the pair of trajectories *i* is computed for the 2*m*+1 registrations. The minimal symmetry measure

$$dist_i = \min \{ dist_i^{N-m}, dist_i^{N-m+1}, \ldots, dist_i^{N+m} \} \qquad (10)$$

is chosen as the final symmetry measure for the pair of trajectories *i*.

6.3 Tracing body parts from video

To apply the algorithm to real video data, the first step is to extract the trajectories of the body parts, i.e., tracking body parts from video.

Object tracking in video streams has been a popular topic in the field of computer vision for many years and many methods have been presented. For example, Guo et al. [94] represent the human body structure of the silhouette by a stick figure model which has ten sticks articulated with six joints. They transform the problem into finding a stick figure with minimal energy in a potential field. A. O. Balan et al. [95] presented an algorithm based on an adaptive appearance model that includes an adaptive template, frame-to-frame matching and a post-process to get rid of outliers. An annealed particle filtering algorithm is employed to infer the position of the body parts. A 3-D body model is used to predict self-occlusion and improve pose estimation accuracy. When using data from four views, they obtained very good tracking results. But when the number of views decreases, the results are degraded. An important advantage of using a 3-D human model is the ability to handle occlusion and obtain more accurate data for action analysis. However, it is it is computationally expensive.

In this chapter, we present a template-based algorithm to track human body parts in a video for our application. Figure 6.2 shows the steps of this algorithm. Initially, a set of templates are built to model static states of gait from profile view. Then, the extraction of the body parts' trajectories from a video is achieved by three steps. In the first step, for each frame of the video, the region of interest (ROI), i.e., binary silhouette, is extracted and the template which is most similar to the extracted ROI is found through a matching algorithm. In the second step, the body parts' positions are estimated by using an assignment algorithm for each frame. In the last step, the trajectories of body parts are obtained from the information extracted from the whole video.

6.3.1 Building the template

The cycle of a gait can be taken as a sequence of states where each state is modeled by one or more templates. Each template includes a normalized binary silhouette image of size 192*144 and its contour image with manually labeled body parts that are denoted as “front hand,” “rear hand,” “front knee,” “rear knee,” “front foot” and “rear foot.” Figure 6.3 shows two examples. Each body part is labeled by a set of contour points as what is shown in Figure 6.3 (b) and (d). The geometric center of each set of points is considered as the position of the body part represented by the set of points. What needs to be noticed is that for some templates, the hands, knees or feet are not labeled because in these templates some parts of the body do not clearly appear in the contour and therefore cannot be labeled accurately. For example, the template shown in Figure 6.3(b) only has one labeled knee. In our experiment, 37 templates were selected and labeled manually from the training data.

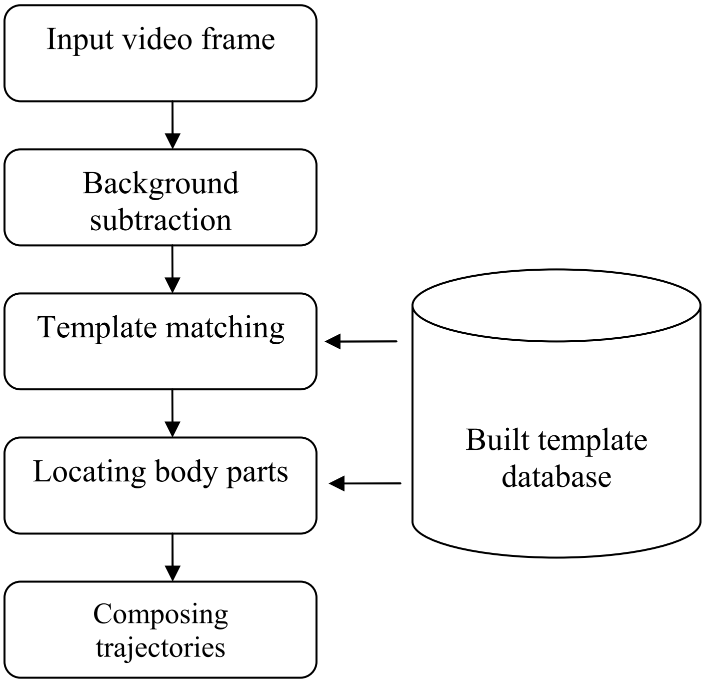
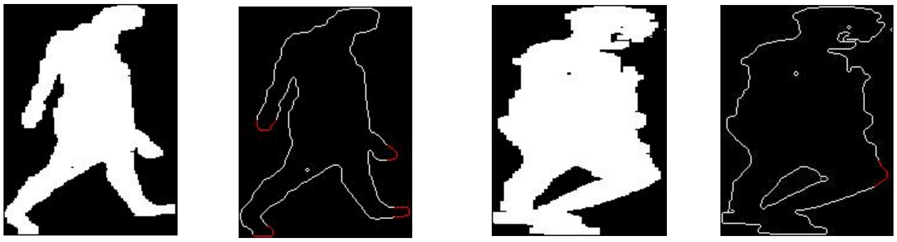


Figure 6.2 The block diagram of our algorithm to obtain trajectories



(a) **(b)** **(c)** **(d)**

Figure 6.3 Examples of templates

6.3.2 Template matching

Given a frame of video, at first, the binary region of interest (ROI) image p and its contour image p_e are extracted by using the background subtraction algorithm used in [101]. Then the template most similar to extracted ROI must be selected from the stored templates. Before matching**,** the extracted ROI image has to be normalized into the size of the stored templates. The normalization is straightforward: we enlarge/shrink the ROI image to the same height and width. The normalized ROI image is denoted as $\overline{p}$. Then the template t which is most similar to p can be selected by

$$t \ = \arg\max_{t_i} S(\overline{p}, t_i) \quad t_i \in T_s$$

Where $S(\overline{p}, t_i)$ is the ratio of the intersection and the union of the two binary silhouettes and is used as a measure of the similarity between them,

$$S(\overline{p}, t_i) = \frac{\overline{p} \cap t_i}{\overline{p} \cup t_i},$$

t_i denotes the binary silhouette template and T_s is the set of normalized binary silhouette templates. The intersection and the union are calculated in terms of the number of pixels. The value of this ratio ranges within [0*,* 1], with 0 representing no overlap between the two silhouettes, and 1 implying an identical match.

6.3.3 Locating body parts

After obtaining the matched template *t*, we locate positions of the body parts in the extracted ROI image *p* by matching p_e and t_e which are the contour images of *p* and *t*. The goal of matching is to find the "best" matching point on p_e for each point on t_e. This can be formulated as a bipartite assignment problem. We use an algorithm presented in [97-98] to find the optimal matching between the two contours. More details are presented in Appendix III. Then we obtain the sets of points that correspond to the hands, the knees and the feet, respectively, on the contour p_e. The centers of each set of points are computed as the positions of the hands, the knees and the feet in *p*. The corresponding positions in the original video frame are obtained accordingly.

6.3.4 Composing trajectories

Using the algorithm described in the two previous sections, we can obtain the positions of the hands, the knees and the feet in the frames of a video. Those positions are labeled as "front hand," "rear hand," "front knee," "rear knee," "front foot" and "rear foot." The points labeled as "front hand" cannot be used to compose a hand's trajectory, because the "front hand" may be the left hand in some frames and the right hand in other frames. We need to differentiate between the left and right hands, knees and feet. Walking is a periodic behavior that can be divided into two half cycles. In the first half cycle, one hand is in front of the body and the other hand is behind the body. In the second half cycle, they are reversed. This means that the front hand in the first half cycle is the rear hand in the second half cycle. The boundary between the first half cycle and the second half cycle can be robustly detected by templates denoted as boundary templates. By detecting the boundary between the two half cycles of the gait, we can appropriately assign the positions of the “front hand” and the “rear hand” to the left or the right trajectories. Using the same method, we can obtain the trajectories for the knees and the feet. We need to mention that the obtained trajectories are discontinuous trajectories because the hands, knees and feet are not localized for some frames of video. This is not a problem for the algorithm presented because the symmetry measure presented in Section 6.1 can deal with incomplete trajectories.

6.4 Experiments and results

We employ two set of experiments. The first experiments are performed on 2D data projected from real 3D data. Then we apply the presented method to real video data.

6.4.1 Experiment on 2D projected data

This experiment is employed on 2D projected data. We prepared the data by projecting 3D trajectories (3d motion-captured data) to 2D profile view using a projective camera model. The 2D projected data contains 72 walking sequences for 13 subjects, which includes six healthy subjects and seven patients with knee problems. For each walking sequence, there are eight pairs of trajectories of markers on the left side and the right side of the body. They are located on the head, shoulders, elbows, hands, waists, knees, ankles and feet. We partitioned the data into six subsets. Each subset contained data from an approximately equal number of "healthy" and "pathological" subjects. A linear support vector machine was used for classification. The cross-validation method was used for training and testing. The virtual camera was placed to capture the profile view of a working person. To check the sensitivity of the proposed method to the resolution of the image, we projected data onto a 2D plane when a virtual camera is placed 500cm far from the subject and at a height of 200cm. We adjusted the length of quantization step to generate the images with different resolutions, such as 0.2cm/pix, 1cm/pix, 2cm/pix, 3cm/pix, 4cm/pix, 5cm/pix, 6cm/pix, 7cm/pix, and 8cm/pix. Figure 6.4 shows the experimental results. From the figure, we can see that the algorithm obtains 84.5% recognition rate when image resolution is higher than 1cm/pix. The recognition rate decreases with the drop of image resolution.

To check the sensitivity of the proposed method to the height of the camera, we projected data onto a 2D plane when the virtual camera is placed at different heights and 500cm far from the subject. The resolutions of those generated image were kept as 1cm/pix. Figure 6.5 shows the experimental results. We can see that the algorithm achieves good recognition rates when the height is lower than 200cm and the recognition rate degrades when the height increases. This is understandable because when the height of the camera increases, the projections of the trajectories tend to converge to lines and the details are lost.

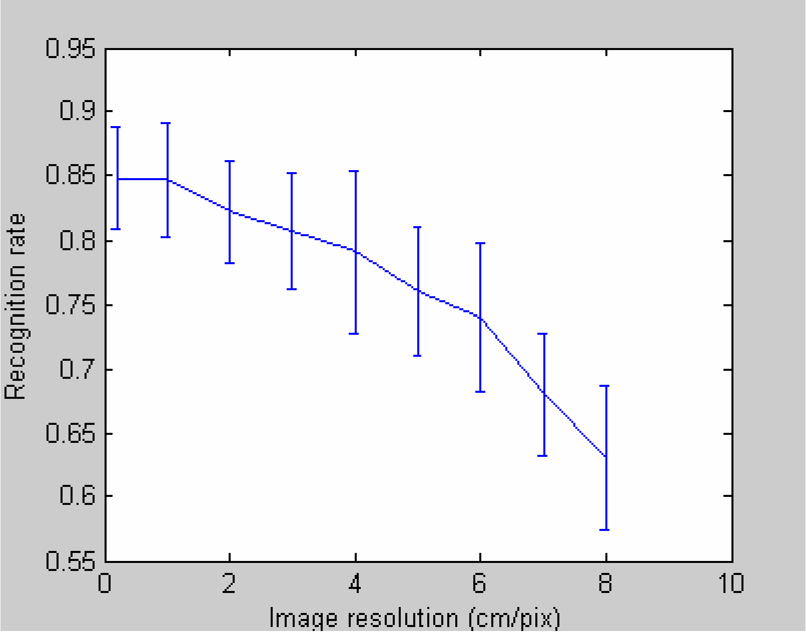


Figure 6.4 Means and standard deviations of recognition rates for different image resolutions. The camera was placed at 500cm far from subject and 200cm high. Only trajectories of hands, knees and feet were used.

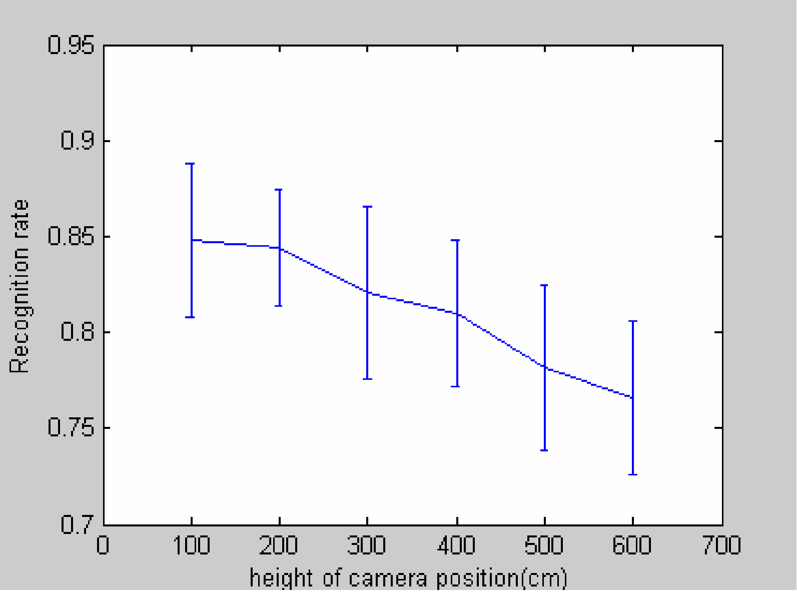


Figure 6.5 Means and standard deviations of recognition rates for different heights of the camera. Distance between the subject and the camera is set as 500cm. Only trajectories of hands, knees and feet were used.

In a real video sequence, it is hard to track the movement of each part of the entire body.

It is easy to track some of the body parts, such as the hands, feet and knees. Therefore, in order to study the feasibility of the algorithm for video data, we performed experiments using only a subset of the projected trajectories, e.g., the trajectories of the hands, feet and knees, in experiments shown in figure 6.4 and figure 6.5. We also executed experiments using trajectories of more body parts. Figure 6.6 shows the Receiver Operating Characteristic (ROC) curves for an experiment using different data. The experiment shows the best results are achieved when using all of the data and the results degrade when using data only from the hands, elbows, feet, and knees. The results from using only trajectories of the hands, feet and knees, and excluding the elbows, degrade a little bit compared to the results obtained when using trajectories of the hands, elbows, feet and knees.

To examine the sensitivity of the proposed method to errors in trajectories, we performed an experiment on the projected data with added Gaussian noise. The standard deviation is 2cm for the added noises. Their means are 2cm, 4cm, 6cm, 8cm and 10cm. Figure 6.7 shows the experimental results. We can see the recognition result degraded very fast when the mean of the errors was greater than 6 cm. And for greater than 10 cm, the features extracted from the corrupted data almost lost discrimination power.

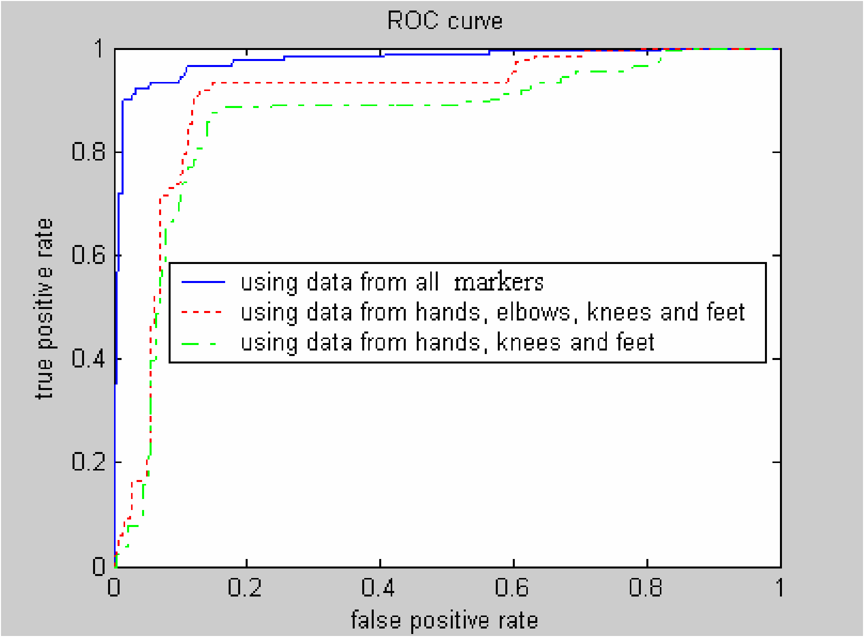


Figure 6.6 Experimental results of using different data. The height of the camera is set as 200cm, the distance between the subject and the camera is set as 500cm

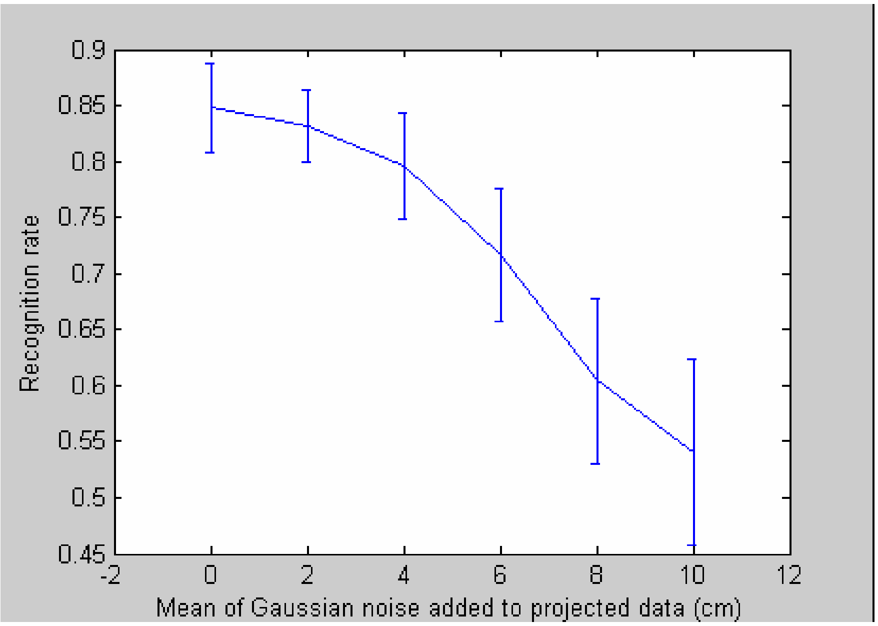


Figure 6.7 Means and standard deviations of recognition rates for projected data with different noise. Only trajectories of hands, knees and feet are used.

6.4.2 Experiments on real video data

6.4.2.1 Tracking results

To examine the accuracy of the tracking algorithm, we manually annotated the positions of body parts in four test sequences (a total of 432 frames) and computed the image distance error of the estimated joint positions. Table 1 shows the average error for each body part. The estimates for the knee are relatively better than for the hands and feet. The overall average error is 10.6 pixels, which represents approximately 6cm. Figure 6.8 shows some example frames where the body parts are estimated by using the presented algorithm.

Table 6.1 Breakdown of error in pixels after 2D tracking

Joints	Ave. 2D Error
hands	10.3
knees	9.4
feet	11.1
overall	10.6

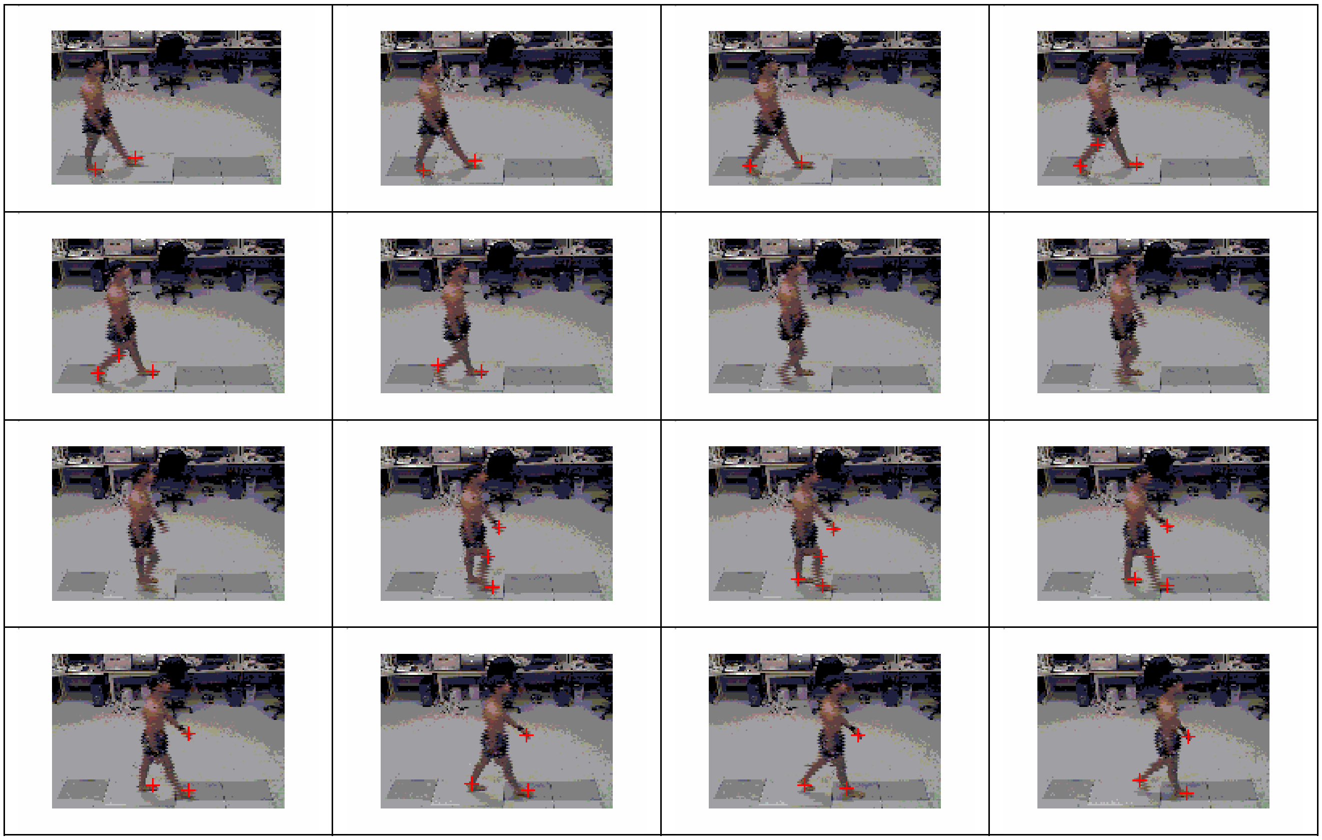


Figure 6.8 Examples of tracking results

6.4.2.2 Results of experiment on real video data

We applied the algorithm to a video database that contained a total of 72 video clips. A linear SVM was used for classification. The cross-validation method was used for training and testing. The algorithm achieved an average of 76.4% recognition rate, which is worse than the 84.2% recognition rate obtained when using the projected data. Figure 6.9 shows the ROC curves for the experiments. Section 6.4.2.1 shows that the average tracking error of the presented tracking algorithm, on our video data, was 10.6 pixels, which represents approximately 6 cm. From the experiment shown in figure 6.8, we obtained a recognition rate of 71.7% when adding the Gaussian noise (mean 6cm, standard deviation) to projected 2D data. Based on those two results, we think that the degradation in the recognition rate of video data compared to projected 2D data is due to the errors in tracking. Also, we believe a better tracking result will lead to more accurate pathological gait identification results.

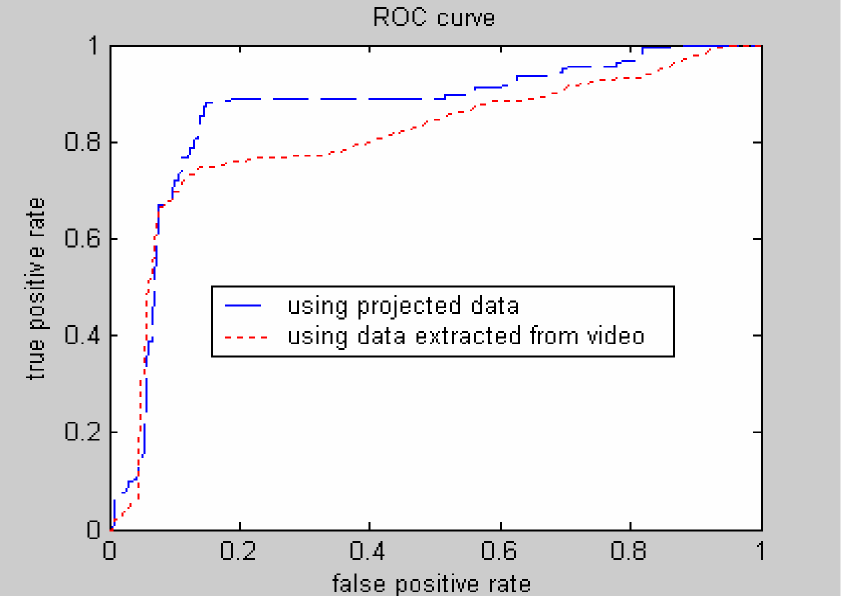


Figure 6.9 Experimental results for using hands, knee, feet data

6.5 Conclusions

In this chapter, we presented an algorithm to identify pathological gait from video data. At first a symmetry measure of two 3D trajectories based on their 2D profile view projections was presented. Then, a feature vector based on symmetry was built to represent a person’s gait. Finally, a linear SVM classifier was used to differentiate between normal and pathological gaits. We experimented with two databases. The first database consists of 2D data projected from 3D motion-captured data. We obtained an 84.5% recognition rate when using the trajectories of the hands, knees and feet and a virtual camera placed under 200cm height and 500cm far from the object. When trajectories of additional body parts were used to build the feature vector, better recognition results were obtained. The experiments on the 2D projected data also showed that the algorithm is sensitive to resolution of the image. However, in real applications, the camera should not be placed too far from the object because of the effect of the resolution of the camera. The experimental results on 2D projected data demonstrated that the presented algorithm is promising for identifying pathological gait from video.

In the second set of experiments we used a database of real video data. Because it is very difficult to track elbows and other body parts by using the tracking algorithm presented, we only used the trajectories of hands, knees and feet in the second experiment. The experimental results on the real video data achieved 76.4% recognition rate. We believe that better results can be obtained if the accuracy of the tracking algorithm is improved.

Chapter 7

Summary and contributions

In this thesis, we have provided two methodologies for general human activity recognition and pathological gait identification. In the methodology for general human activity recognition, we combine motion-based features with shape-based features to model human activities. We represent each activity by a set of Hidden Markov Models, where each model represents the activity viewed from a specific direction to realize the view-invariance. We also presented a voting based method to segment and recognize continuous complex human activities. In the methodology for pathological gait identification, we used a symmetry measure of trajectories of body parts as a feature to represent gait. Then, an SVM classifier was trained and used for classification. In particular, we developed two algorithms to identify pathological gaits using 3D motion-captured data and video data, respectively. We also presented a template-based algorithm to track human body parts from a single profile video which is used to identify pathological gait.

The major contributions of this research include:

- A shape representation method based on PCA. Even though we use it to represent the silhouette of a person in our work, it can be used to describe the shape of any object in other applications.

- A method that uses both shape and motion information for representing human activities. Because both shape feature and motion feature have their own advantages and they are complementary to each other, this representation has better attributes. Besides, this developed representation is general and can be extracted from any view and is suitable for both high resolution and low resolution video.

- A view independent and HMM model based method for activity recognition.

- A voting and HMM based algorithm for segmenting and recognizing complex activities. In training stage, the algorithm only needs single activity training samples. In recognition stage, this algorithm finishes activity segmentation and recognition in one process.

- A gait symmetry measurement and SVM based framework for identifying pathological gait.

- An algorithm for identifying pathological gait using 3D motion captured data. A DFT based gait symmetry measure using 3D trajectories of the human body parts was presented to represent the human gait for pathological gait identification.

- A 2D template based method to tracking body parts from profile view video.

- An algorithm for identifying pathological gait using video data. Measuring the symmetry of two 3D trajectories based on their 2D projections is not easy. In this dissertation, we developed a symmetry measure of two 3D trajectories based on their projections in the 2D plane and used it to represent the human gait for pathological gait identification.

Appendix I

In this Appendix we present the details of deriving the 2-D symmetry, Eq. (3), from Equations (1) and (2).

Assume

$$S_1 = \begin{bmatrix} s_{x,1}(1) & s_{x,1}(2) & \dots & s_{x,1}(L) \\ s_{y,1}(1) & s_{y,1}(2) & \dots & s_{y,1}(L) \\ s_{z,1}(1) & s_{z,1}(2) & \dots & s_{z,1}(L) \end{bmatrix} \text{ and } S_2 = \begin{bmatrix} s_{x,2}(1) & s_{x,2}(2) & \dots & s_{x,2}(L) \\ s_{y,2}(1) & s_{y,2}(2) & \dots & s_{y,2}(L) \\ s_{z,2}(1) & s_{z,2}(2) & \dots & s_{z,2}(L) \end{bmatrix}$$

They are two symmetrical trajectories in 3D space. The relationship between S_1 and S_2 can be represented as

$$\begin{bmatrix} s_{x,1}(n) \\ s_{y,1}(n) \\ s_{z,1}(n) \end{bmatrix} = \begin{bmatrix} 1 & \alpha & -\gamma \\ -\alpha & 1 & \beta \\ \gamma & -\beta & 1 \end{bmatrix} \begin{bmatrix} s_{x,2}(n) \\ s_{y,2}(n) \\ s_{z,2}(n) \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \qquad n = 1 \dots L$$

I_1 and I_2 are 2D projections of S_1 and S_2

$$I_1 = \begin{bmatrix} u_1(1) & u_1(2) & \dots & u_1(L) \\ v_1(1) & v_1(2) & & v_1(L) \end{bmatrix}, I_2 = \begin{bmatrix} u_2(1) & u_2(2) & \dots & u_2(L) \\ v_2(1) & v_2(2) & & v_2(L) \end{bmatrix}$$

where

$$\begin{cases} u_1(n) = \dfrac{fs_{x,1}(n) + s_{z,1}(n) p_x}{s_{z,1}(n)} \\ v_1(n) = \dfrac{fs_{y,1}(n) + s_{z,1}(n) p_y}{s_{z,1}(n)} \end{cases}$$

and

$$\begin{cases} u_2(n) = \dfrac{fs_{x,2}(n) + s_{z,2}(n) p_x}{s_{z,2}(n)} \\ v_2(n) = \dfrac{fs_{y,2}(n) + s_{z,2}(n) p_y}{s_{z,2}(n)} \end{cases}$$

Therefore, we have

$$\begin{cases} u_1(n) = \dfrac{fs_{x,1}(n) + s_{z,1}(n) p_x}{s_{z,1}(n)} \\ v_1(n) = \dfrac{fs_{y,1}(n) + s_{z,1}(n) p_y}{s_{z,1}(n)} \end{cases}$$

$$\Rightarrow \begin{cases} u_1(n) - p_x = \dfrac{fs_{x,1}(n)}{s_{z,1}(n)} \\ v_1(n) - p_y = \dfrac{fs_{y,1}(n)}{s_{z,1}(n)} \end{cases} \Rightarrow \begin{cases} u_1(n) - p_x = \dfrac{f \times (s_{x,2}(n) + \alpha\ s_{y,2}(n) - \gamma\ s_{z,2}(n) + t_x)}{\gamma\ s_{x,2}(n) - \beta\ s_{y,2}(n) - s_{z,2}(n) + t_z} \\ v_1(n) - p_y = \dfrac{f \times (-\alpha\ s_{x,2}(n) + s_{y,2}(n) - \beta\ s_{z,2}(n) + t_y)}{\gamma\ s_{x,2}(n) - \beta\ s_{y,2}(n) - s_{z,2}(n) + t_z} \end{cases}$$

$$\Rightarrow \begin{cases} u_1(n) - p_x = f \times \dfrac{u_2(n) + \alpha\ v_2(n) - - f\gamma - f\ t_x / s_{z,2}(n) - p_x - p_y \alpha}{\gamma\ u_2(n) - \beta\ v_2(n) - f\gamma - f\ t_z / s_{z,2}(n) - p_x \gamma + p_y \beta} \\ v_1(n) - p_y = f \times \dfrac{-\alpha u_2(n) + v_2(n) - f\gamma - f\ t_y / s_{z,2}(n) + p_x \alpha - p_y}{\gamma\ u_2(n) - \beta\ v_2(n) - f\gamma - f\ t_z / s_{z,2}(n) - p_x \gamma + p_y \beta} \end{cases}$$

So

$$\begin{cases} u_1(n) - p_x = f \dfrac{u_2(n) + \alpha\ v_2(n) + T_u(s_{z,2}(n))}{\gamma\ u_2(n) - \beta\ v_2(n) + T_w(s_{z,2}(n))} \\ v_1(n) - p_y = f \dfrac{-\alpha u_2(n) + v_2(n) + T_v(s_{z,2}(n))}{\gamma\ u_2(n) - \beta\ v_2(n) + T_w(s_{z,2}(n))} \end{cases}$$

where

$$\begin{cases} T_u(s_{z,2}(n)) = -f\gamma - f\ t_x / s_{z,2}(n) - p_x - p_y \alpha \\ T_v(s_{z,2}(n)) = -f\gamma - f\ t_y / s_{z,2}(n) + p_x \alpha - p_y \\ T_w(s_{z,2}(n)) = -f\gamma - f\ t_z / s_{z,2}(n) - p_x \gamma + p_y \beta \end{cases}$$

Appendix II

Computing minimum residual of a linear system

Let $A \in R^{m\times n} (m > n)$ and $b \in R^{m}$ be given and suppose that an orthogonal matrix $Q \in R^{m\times m}$ has been computed such that

$$Q^T A = R = \begin{array}{cc} \begin{bmatrix} R_1 \\ 0 \end{bmatrix} & \begin{array}{c} n \\ m-n \end{array} \end{array} \quad (1)$$

is upper triangular. If

$$Q^T b = \begin{array}{cc} \begin{bmatrix} c \\ d \end{bmatrix} & \begin{array}{c} n \\ m-n \end{array} \end{array} \quad (2)$$

Then

$$\begin{aligned} \|Ax-b\|_2^2 &= \|Q^T A - Q^T b\|_2^2 \\ &= \|R_1 x - c\|_2^2 + \|d\|_2^2 \end{aligned} \quad (3)$$

for any $x \in R^n$. Clearly, if rank(A) = rank(R_1) = n, then x_{LS} is defined by the upper triangular system $R_1 x_{LS} = c$. Note that the square of the minimum residual $\rho_{LS}^2 = \|d\|_2^2$.

Thus, the full rank Least Squares (LS) problem can be readily solved once we have computed (1), which we refer to as the Q-R factorization. It can be calculated in several ways. The algorithm we used is *Householder Orthogonalization*[93].

Algorithm *Householder Orthogonalization*

Given $A \in R^{m \times n} (m > n)$, the following algorithm computer an orthogonal Q such that $Q^T A = R$ is upper triangular.

- Compute the factor R

Leave result in place of A, store reflection vectors v_k for later use

```
    **For** k = 1 **to** n
    $x = A_{k:m,k}$
    $v_k = \text{sign}(x_1) \|x\|_2 e_1 + x$
    $v_k = v_k / \|v_k\|_2$
    $A_{k:m,k:n} = A_{k:m,k:n} - 2v_k (v_k^T A_{k:m,k:n})$
```

- Compute $Q^T b$

```
    **For** k = 1 **to** n
    $b_{k:m} = b_{k:m} - 2v_k (v_k^T b_{k:m})$
```

Appendix III

Matching using Shape Contexts [97]

In our approach, a shape is represented by a discrete set $P = \{p_1, ..., p_n\}$, $p_i \in R^2$, of n points sampled from the external contours on the shape.

We first perform Canny edge detection on the image to obtain a set of edge pixels on the contours of the body. We then sample some number of points (around 300 in our experiments) from these edge pixels to use as the sample points for the body.

For each point p_i on a given shape, we want to find the "best" matching point q_j on another shape. This is a correspondence problem similar to that in stereopsis. Experience there suggests that matching is easier if one uses a rich local descriptor. Rich descriptors reduce the ambiguity in matching.

Consider the set of vectors originating from a point to all other sample points on a shape. These vectors express the configuration of the entire shape relative to the reference point. Obviously, this set of n-1 vectors is a rich description, since as n gets large, the representation of the shape becomes exact.

The full set of vectors as a shape descriptor is much too detailed since shapes and their sampled representation may vary from one instance to another in a category. The *distribution* over relative positions is a more robust and compact, yet highly discriminative descriptor. For a point p_i on the shape, compute a coarse histogram h_i of the relative coordinates of the remaining n-1 points,

$$h_i(k) = \#\{q \neq p_i : (q - p_i) \in \mathrm{bin}(k)\}$$

This histogram is defined to be the *shape context* of p_i. The descriptor should be more sensitive to differences in nearby pixels, which suggests the use of a log-polar coordinate system. An example is shown in Fig. A.1(c). Note that the scale of the bins for log r is chosen adaptively, on a per shape basis. This makes the shape context feature invariant to scaling.

We use x^2 distances between shape contexts as a matching cost between sample points. We would like a correspondence between sample points on the two shapes that enforces the uniqueness of matches. The problem of matching of a test body to an exemplar body is formulated as an assignment problem (also known as the weighted bipartite matching problem). We find an optimal assignment between sample points on the test body and those on the exemplar.

To this end a bipartite graph is constructed (Figure A.2). The nodes on one side represent sample points on the test body, on the other side the sample points on the exemplar. Edge weights between nodes in this bipartite graph represent the costs of matching sample points. Similar sample points will have a low matching cost. Dissimilar ones will have a high matching cost. ε -cost outlier nodes are added to the graph to account for occluded points and noise - sample points missing from a shape can be assigned to be outliers for some small cost. Assignment problem solvers can be used to find the optimal matching between the sample points of the two bodies, such as method presented in [100].

We compare the test body to all of the exemplars from our training set. The exemplar with the lowest total matching cost is chosen as the matching result.

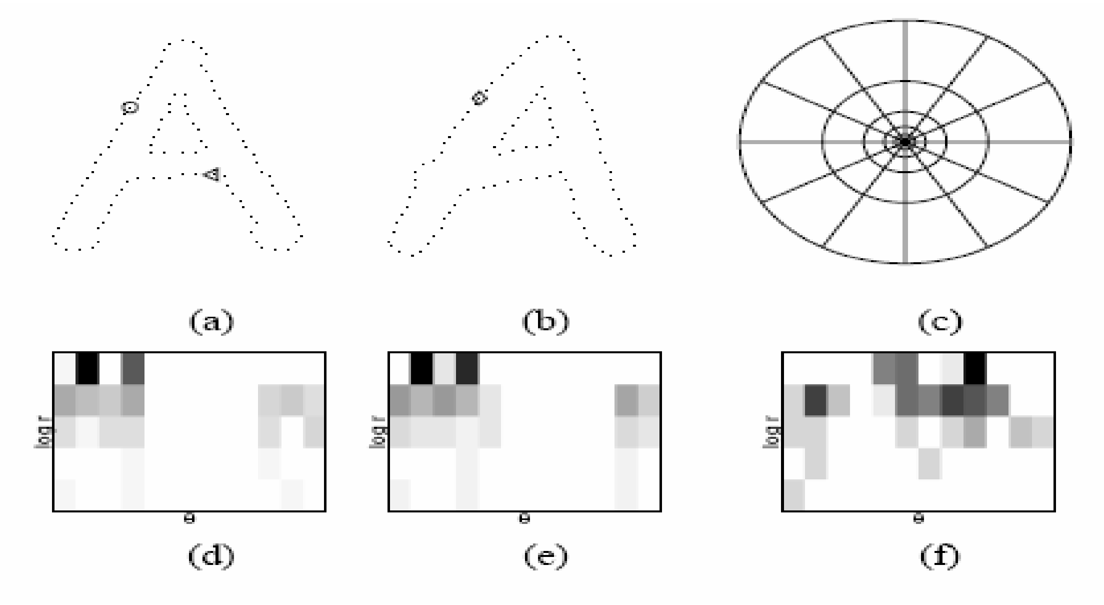


Figure A.1 Shape contexts. (a,b) Sampled edge points of two shapes. (c) Diagram of log-polar histogram bins used in computing the shape contexts. We use 5 bins for log *r* and 12 bins for θ. (d-f) Example shape contexts for reference samples marked by ○, ◇, ◁ in (a,b). Each shape context is a log-polar histogram of the coordinates of the rest of the point set measured using the reference point as the origin. (Dark=large value.) Note the visual similarity of the shape contexts for ○ and ◇, which were computed for relatively similar points on the two shapes. By contrast, the shape context for ◁ is quite different.

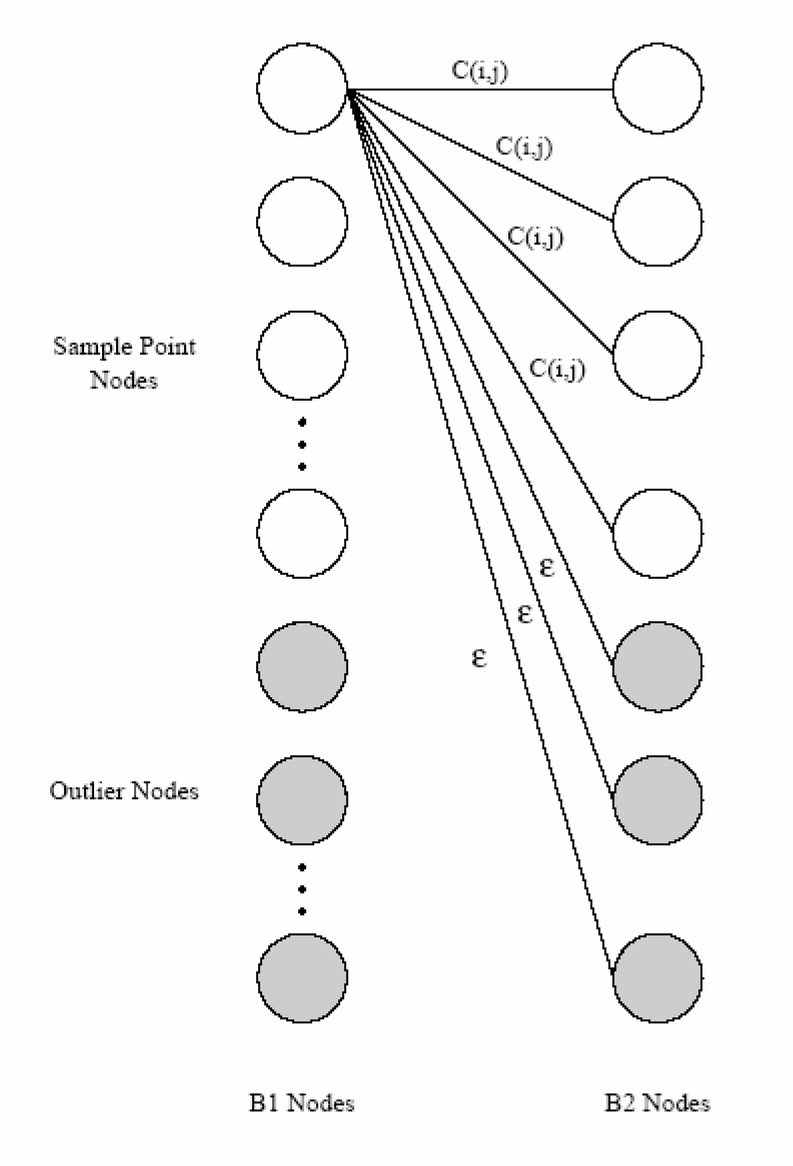


Figure A.2 The bipartite graph used to match sample points of two bodies. The bipartite graph used to match sample points of two bodies. Only the edges from the first node are shown for clarity. Each node from B1 is connected to every node from B2. In addition, ε -cost outlier nodes are added to either side. These outlier nodes allow us to deal with missing sample points between figures (arising from occlusion and noise).

References

[1]. R.T. Collins et al., “A system for video surveillance and monitoring: VSAM final report”, CMU-RI-TR-00-12, Technical Report, Carnegie Mellon University, 2000.

[2]. P. Remagnino, T. Tan and K. Baker, “Multi-agent visual surveillance of dynamic scenes”, *Image and Vision Computing*, 16 (8), pp. 529-532, 1998.

[3]. C. Maggioni and B. Kammerer, “Gesture Computer: history, design, and applications”. *Computer Vision for Human-Machine Interaction*. Cambridge Univ. Press, 1998.

[4]. W. Freeman, C. Weissman, “Television control by hand gestures”. *Proc. of Intl. Conf. on Automatic Face and Gesture Recognition*. pp. 179-183, 1995.

[5]. R.T. Collins, A.J. Lipton and T. Kanade, “Introduction to the special section on video surveillance”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22 (8), pp. 745-746, 2000.

[6]. S. Maybank and T. Tan, “Introduction to special section on visual surveillance”, *International Journal of Computer Vision*, 37 (2), pp. 173-173, 2000.

[7]. J.J. Little, J.E. Boyd, “Recognizing people by their gait: the shape of motion”, *Videre: Journal of Computer Vision Research*, The MIT Press, 1 (2), 1998.

[8]. J.D. Shutler, M.S. Nixon and C.J. Harris, “Statistical gait recognition via velocity moments”, *Proc. of IEE Colloquium on Visual Biometrics,* pp. 101-105, 2000.

[9]. P.S. Huang, C.J. Harris and M.S. Nixon, “Human gait recognition in canonical space using temporal templates”, *Proc. of IEE Vis. Image Signal Process*. 146 (2), pp. 93-100, 1999.

[10]. D. Cunado, M.S. Nixon and J.N. Carter, “Automatic gait recognition via model-based evidence gathering”, *Proc. of Workshop on Automatic Identification Advanced Technologies*, pp. 27-30, 1998, New Jersey.

[11]. B.A. Boghossian and S.A. Velastin, “Image processing system for pedestrian monitoring using neural classification of normal motion patterns”, *Measurement and Control*, 32 (9), pp. 261-264, 1999.

[12]. B.A. Boghossian and S.A. Velastin, “Motion-based machine vision techniques for the management of large crowds”, *Proc. of IEEE 6th Intl. Conf. on Electronics, Circuits and Systems*. September, 1999.

[13]. Yi Li, Songde Ma and Hanqing Lu, “Human posture recognition using multi-scale morphological method and Kalman motion estimation”, *Proc. of IEEE Intl. Conf. on Pattern Recognition*, pp. 175-177, 1998.

[14]. J. Segen and S. Kumar, “Shadow gestures: 3D hand pose estimation using a single camera”, *Proc. of IEEE CS Conf. on Computer Vision and Pattern Recognition*, pp. 479-485, 1999.

[15]. M. Turk, “Visual interaction with lifelike characters”, *Proc. of IEEE Intl. Conf. on Automatic Face and Gesture Recognition*, pp. 368-373, 1996, Killington.

[16]. H.M. Lakany, G.M. Haycs, M. Hazlewood and S.J. Hillman, “Human walking: tracking and analysis”, *Proc. of IEE Colloquium on Motion Analysis and Tracking*, pp. 5/1-5/14, 1999.

[17]. M. Köhle, D. Merkl and J. Kastner, “Clinical gait analysis by neural networks: issues and experiences”, *Proc. of IEEE Symp. on Computer-Based Medical Systems*, pp. 138-143, 1997.

[18]. D. Meyer, J. Denzler and H. Niemann, “Model based extraction of articulated objects in image sequences for gait analysis”, *Proc. of IEEE Intl. Conf. on Image Processing*, pp. 78-81, 1997.

[19]. W. Freeman et al., “Computer vision for computer games”, *Proc. of Intl. Conf. on Automatic Face and Gesture Recognition*, pp. 100-105, 1996.

[20]. G. Rigoll, S. Eickeler and S. Müller, “Person tracking in real world scenarios using statistical methods”, *Proc. of Intl. Conf. on Automatic Face and Gesture Recognition*, March 2000, France.

[21]. I.A. Kakadiaris and D. Metaxas, “Model-based estimation of 3-D human motion with occlusion based on active multi-viewpoint selection”, *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 81-87, 1996.

[22]. Q. Cai and J.K. Aggarwal, “Tracking human motion using multiple cameras”, *Proc. of 13th Intl. Conf. on Pattern Recognition*, pp. 68-72, 1996.

[23]. D. Gavrila and L. Davis, “3-D model-based tracking of humans in action: a multi-view approach”, *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 73-80, 1996.

[24]. I.A. Kakadiaris and D. Metaxas, “Three-dimensional human body model acquisition from multiple views”, *International Journal of Computer Vision*, 30 (3), pp. 191-218, 1998.

[25]. H. Sidenbladh, F. Torre and M. J. Black, “A framework for modeling the appearance of 3D articulated figures”, *Proc. of Intl. Conf. on Automatic Face and Gesture Recognition*, March 2000.

[26]. R. Plänkers, P. Fua, Articulated soft object for video-based body modeling. *Proc. of International Conference on Computer Vision*, 2001.

[27]. A. Hilton and P. Fua, “Foreword: Modeling people toward vision-based understanding of a person’s shape, appearance, and movement”, *Computer Vision and Image Understanding*, 81 (3), pp. 227-230, 1997.

[28]. I. Haritaoglu, D. Harwood and L.S. Davis, “W^4: real-time surveillance of people and their activities”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22 (8), pp. 809-830, 2000.

[29]. T.B. Moeslund and E. Granum, “A survey of computer vision-based human motion capture”, *Computer Vision and Image Understanding*, 81 (3), pp. 231-268, 2001.

[30] C. Rao and M. Shah, “View-Invariance in Action Recognition,” *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 316-313, Dec, 2001.

[31] A. S. Ogale, A. Karapurkar and Y. Aloimonos, “View-invariant modeling and recognition of human actions using grammars,”, *International Conference on Computer Vision Workshop on Dynamical Vision*, October 2005.

[32] C. Kim and J. Eng, “Symmetry in vertical ground reaction force is accompanied by symmetry in temporal but not distance variables of gait in persons with stroke,” *Gait Posture*, 18(1), pp. 23-28, 2003.

[33] M. Brandstater, H. DeBruin, C. Gowland and B. Clark, “Hemiplegic gait: analysis of temporal variables,” *Archives of Physical Medicine and Rehabilitation.*, 64, pp. 583-587, 1983.

[34]E. Titianova and I. Tarkka, “Asymmetry in walking performance and postural sway in patients with chronic unilateral cerebral infarction,” *Journal of Rehabilitation Research and Development*, 32, pp. 236-244, 1995.

[35] E. Hassid, D. Rose, J. Commisarow, M. Guttry and B. Dobkin, “Improved gait symmetry in hemiparetic stroke patients induced during body weight-supported treadmill stepping,” *Journal of Neurologic Rehabilitation*, 11, pp. 21-26, 1997.

[36] K. Silver, R. Macko, L. Forrester, A. Goldberg and G. Smith, “Effects of aerobic treadmill training on gait velocity, cadence, and gait symmetry in chronic hemiparetic stroke: a preliminary report,” *Neurorehabilitation and Neural Repair*,14, pp. 65-71, 2000.

[37] S. Tyson and H. Thornton, “The effect of a hinged ankle foot orthosis on hemiplegic gait: objective measures and users’ opinions,” *Clinical Rehabilitation*, 15, pp. 53-58, 2001.

[38] C. Wang and M.S. Brandstein, “A hybrid real-time face tracking system”, *Proc. of Intl. Conf. on Acoustics, Speech, and Signal Processing*, Seattle, WA, 1998.

[39]. A.J. Lipton, H. Fujiyoshi and R. S. Patil, “Moving target classification and tracking from real-time video”. *Proc. of IEEE Workshop on Applications of Computer Vision*, pp. 8-14, 1998.

[40]. J. Barron, D. Fleet and S. Beauchemin, “Performance of optical flow techniques”, *International Journal of Computer Vision*, 12 (1), pp. 42-77, 1994.

[41]. H.A. Rowley and J.M. Rehg, “Analyzing articulated motion using expectation-maximization”, *Proc. of Intl. Conf. on Pattern recognition*, pp. 935-941, 1997.

[42]. K.P. Karmann and A. Brandt, “Moving object recognition using an adaptive background memory”, *Time-varying Image Processing and Moving Object Recognition*, 2.Elsevier, Amsterdam, The Netherlands, 1990.

[43]. M. Kilger, “A shadow handler in a video-based real-time traffic monitoring system”, *Proc. of IEEE Workshop on Applications of Computer Vision*, pp. 1060-1066, 1992.

[44]. Y.H. Yang, M.D. Levine, The background primal sketch: an approach for tracking moving objects, Machine Vision and applications, 5 (1992) 17-34.

[45]. C.R. Wren, A. Azarbayejani, T. Darrell and A. P. Pentland, “Pfinder: real-time tracking of the human body”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19 (7), pp. 780-785, 1997.

[46]. C. Stauffer and W. Grimson, “Adaptive background mixture models for real-time tracking”, *Proc. of IEEE CS Conf. on Computer Vision and Pattern Recognition*, Vol. 2, pp. 246-252, 1999.

[47]. S.J. McKenna et al., “Tracking groups of people”, *Computer Vision and Image Understanding*, 80 (1), pp. 42-56, 2000.

[48]. N. Friedman and S. Russell, “Image segmentation in video sequences: a probabilistic approach”, *Proc. of the Thirteenth Conf. on Uncertainty in Artificial Intelligence*, Aug. 1997.

[49]. E. Stringa, “Morphological change detection algorithms for surveillance applications”, *British Machine Vision Conference*, pp. 402-411, 2000.

[50] X. Sun, C.W. Chen and B.S. Manjunath, “Probabilistic Motion Parameter Models for Activity Recognition”, *the 16th International Conference on Pattern recognition*, Volume 1, PP. 104443-104450, Quebec City, QC, Canada, August, 2002.

[51] O. Masoud and N. Papanikolopoulos, “Recognizing Human Activities,” *IEEE Conference on Advanced Video and Signal Based Surveillance*, pp. 157-162, Miami, Florida, July, 2003.

[52] R. Hamid, Y. Huang and I. Essa, “ARGMode –Activity Recognition using Graphical Models”, *Conference on Computer Vision and Pattern Recognition Workshop,* Volume 4, pp. 38-45, Madison, Wisconsin, June 16 - 22, 2003.

[53] J. Ben-Arie, Z. Wang, P. Pandit and S. Rajaram, “Human Activity Recognition using Multidimensional Indexing,” *IEEE Transactions on Pattern Analysis and Machine Intelligence Archive* Volume 24, Issue 8, PP. 1091-1104, August 2002.

[54] B. Maurin, O. Masoud, N.P. Papanikolopoulos, “Camera Surveillance of Crowded Traffic Scenes,” *Proc. of ITS America 12th Annual Meeting*, pp. 28-58, Long Beach, CA, April 2002.

[55] R. Polana and R. Nelson, Low level recognition of human motion. *Proc. of IEEE CS Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 77-82 Austin, TX, 1994.

[56] J. Yamato, J. Ohya and K. Ishii, “Recognizing human action in time-sequential images using hidden Markov model”, *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 379-385, 1992.

[57] C. Bregler, “Learning and recognizing human dynamics in video sequences”, *Proc. of IEEE CS Conf. on Computer Vision and Pattern Recognition*, pp. 568-574, 1997.

[58] A. Ali and J.K.Aggarwal, “Segmentation and Recognition of Continuous Human Activity,” *IEEE Workshop on Detection and Recognition of Events in Video*, pp. 28-35, Vancouver, Canada, July 08 - 08, 2001.

[59] I. Cohen and H. Li, “Inference of Human Postures by Classification of 3D Human Body Shape”, *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, pp. 74-81, 2003.

[60] S. Carlsson and J. Sullivan, “Action Recognition by Shape Matching to Key Frames”, *IEEE Computer Society Workshop on Models versus Exemplars in Computer Vision*, pp. 263-270, Miami, Florida, June 30, 2002.

[61] M. Leo, T. D'Orazio, I. Gnoni, P. Spagnolo and A. Distante, "Complex Human Activity Recognition for Monitoring Wide Outdoor Environments", *17th International Conference on Pattern Recognition*, Volume 4, pp. 913-916, 2004.

[62] A.F. Bobick and J. Davis, "Real-time recognition of activity using temporal templates", *Proc. of IEEE CS Workshop on Applications of Computer Vision*, pp. 39-42, 1996.

[63] O. Chomat and J.L. Crowley, "Recognizing motion using local appearance", *International Symposium on Intelligent Robotic Systems*, University of Edinburgh, 1998.

[64] C. Myers, L. Rabinier, A. Rosenberg, "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition", *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 28 (6), pp. 623-635, 1980.

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

[66] K. Takahashi, S. Seki et al., "Recognition of dexterous manipulations from time varying images", *Proc. of IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 23-28, Austin, 1994.

[67] A.B. "Poritz, Hidden Markov Models: a guided tour", *Proc. of IEEE Intl. Conf. on Acoustic Speech and Signal Processing*, pp. 7-13, 1988.

[68] L. Rabinier, "A tutorial on hidden Markov models and selected applications in speech recognition", *Proc. of IEEE*, 77 (2), pp. 257-285, 1989.

[69] T. Starner and A. Pentland, "Real-time American Sign Language recognition from video using hidden Markov models", *Proc. of Intl. Symp. on Computer Vision*, pp. 265-270, 1995.

[70] C. Vogler and D. Metaxas, "ASL recognition based on a coupling between HMMs and 3D motion analysis", *Proc. of International Conference on Computer Vision*, pp. 363-369, 1998.

[71] Y. Guo, G. Xu and S. Tsuji, "Understanding human motion patterns", *Proc. of Intl. Conf. on Pattern Recognition*, pp. 325-329, 1994.

[72] M. Rosenblum, Y. Yacoob and L. Davis, "Human emotion recognition from motion using a radial basis function network architecture", *Proc. of IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 43-49 , Austin, 1994

[73] M. Brand, N. Oliver and A. Pentland, "Coupled hidden Markov models for complex action recognition", *Proc. of IEEE CS Conf. on Computer Vision and Pattern Recognition*, pp. 994-999, 1997.

[74] A. Galata, N. Johnson and D. Hogg, "Learning variable-length Markov models of behavior", *Computer Vision and Image Understanding*, 81 (3), pp. 398-413, 2001.

[75] C-T. Lin, H-W. Nein and W-C. Lin, "A space-time delay neural network for motion recognition and its application to lip reading", *International Journal of Neural Systems*, 9 (4), pp. 311-334 , 1999.

[76] V. Parmeswaran and R. Chellappa. "View invariants for human action recognition", *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 613–619, 2003.

[77] B. Fildes, "Injuries among older people: Falls at home and pedestrian accidents", Dove publications, Melbourne, FL, 1994.

[78] S. Holzreiter and M. Kohle, "Assessment of gait patterns using neural networks", *Journal of Biomechanics*, 26, pp. 645-651, 1993.

[79] R. Begg, M. Palaniswami and B. Owen, "Support vector machines for automated gait classification", *IEEE Transactions on Biomedical Engineering,* Volume 52, Issue 5, pp. 828-838, May, 2005.

[80] W. Wu, F. Su, Y. Cheng and Y. Chou, "Potential of the genetic algorithm neural network in the assessment of gait patterns in ankle arthrodesis", *Annals of Biomedical Engineering*, 29(1), pp. 83-91, Jan, 2001.

[81] S. Morita, H. Yamamoto. and K. Furuya, "Gait analysis of hemiplegic patients by measurement of ground reaction force", *Scandinavian Journal of Rehabilitation Medicine*, 27, pp. 37-42, 1995.

[82] C. Sminchisescu, B. Triggs, "Covariance Scaled Sampling for Monocular 3d Body Tracking", *IEEE Proc. of Computer Vision and Pattern Recognition*, volume 1, pp. 547-554, December 2001.

[83] A. Elgammal, D. Harwood and L. Davis, "Non Parametric Model for Background Subtraction", *the 6th European Conference on Computer Vision*. Dublin, Ireland, June/July 2000.

[84] L. R. Rabiner, "Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proceedings of the IEEE*, 77(2), pp. 257 -- 286, 1989.

[85] A.D. Wilson, A.F. Bobick, "Recognition and Interpretation of Parametric Gesture", *Sixth International Conference on Computer Vision*, pp. 329-336, 1998.

[86] Y. Nam and K.Y. Wohn, "Recognition of Space-Time Hand-Gestures using Hidden Markov Mdel", *ACM Symposium on Virtual Reality Software and Technology*, HongKong, pp. 51-58, 1996.

[87] A. Pentland and A. Liu, "Modeling and Prediction of Human Behavior", *Intl. Conference on Human- Computer Interaction*, New Orleans, LA, Aug, 2001, pp. 229-242, 1995.

[88] P. Stoll and J. Ohya, "Applications of HMM Modeling to Recognizing Human Gestures in Image sequences for a Man-Machine Interface", *the 4th IEEE International Workshop on Robot and Human Communication*, pp. 129-134, 1995.

[89] R. Davis, S. Ounpuu, D. Tyburski and J. Gage, "A gait analysis data collection and reduction technique", *Human Movement Sciences*, 10, pp. 575-587, 1991.

[90] S. Seitz and C. Dyer, "View-Invariant Analysis of Cyclic Motion", *International Journal of Computer Vision*, volume 25 (3), pp. 231- 251, 1997.

[91] C. Rao, A. Yilmaz and M. Shah, "View-Invariant Representation and Recognition of Actions", *International Journal of Computer Vision*, volume 50(2), pp. 203–226, 2002.

[92] C. Kawamura, M. de Morais Filho, M. Barreto, S. de Paula Asa, Y. Juliano and N. Novo, "Comparison between visual and three-dimensional gait analysis in patients with spastic diplegic cerebral palsy", *Gait Posture*, 2 5(1), pp. 18-24, Jan. 2007.

[93] G.H. Golub and C. F. Van Loan, "Matrix computations", *The Johns Hopkins University Press*, Baltimore, Maryland, 1983.

[94]Y. Guo, G. Xu and S. Tsuji, "Understanding human motion patterns", *Proc. of Intl. Conf. on Pattern Recognition*, pp. 325-329, 1994.

[95] A. O. Balan, M. J. Black. "An Adaptive Appearance Model Approach for Model-based Articulated Object Tracking", *IEEE Conference on Computer Vision and Pattern Recognition*, Volume. 1, pp. 758-765, June 2006.

[96] C. C. Chang and C. J. Lin, LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[97] G. Mori, S. Belongie, and J. Malik, "Efficient shape matching using shape contexts", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume: 27, pp 1832-1837, Nov. 2005.

[98] G. Mori, and J. Malik, "Recovering 3D human body configurations using shape contexts", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume: 28, pp. 1052- 1062, July 2006.

[99] http://www.vicon.com

[100] R. Jonker and A. Volgenant, “A shortest augmenting path algorithm for dense and sparse linear assignment problems”, *Computing*, 38, pp. 325–340, 1987.

[101]F. Niu and M. Abdel-Mottaleb “View-Invariant Human Activity Recognition Based on Shape and Motion Features”, *IEEE Sixth International Symposium on Multimedia Software Engineering 2004.*

[102] F. Niu and M. Abdel-Mottaleb “HMM-based segmentation and recognition of human activity from video sequences”, *ICME,* Amsterdam, July 2005.

[103] F. Niu and M. Abdel-Mottaleb “Continuous Human Activity Recognition Based on Shape and Motion Features”, *International Journal of Robotics and Automation, issue 3, 2007.*

[104] F. Niu, M. Abdel-Mottaleb, S. Asfour and K. Abdelrahman “Identification of Normal and Pathological Gait Patterns base on Symmetry Measure”, submitted to *Pattern Recognition Letter*, 2007.

[105] F. Niu and M. Abdel-Mottaleb “Pathological gait identification using 2D data”, submitted to *Pattern Recognition*, 2007.

[106] P. C. Ribeiro and J. Santos-Victor, “Human Activity Recognition from Video: modeling, feature selection and classification architecture”, *International workshop on human activity recognition and modeling,* Oxford, UK, Sept. 2005.

[107] T. Nakata, “Recognizing Human Activities in Video by Multi-resolutional Optical Flows”, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[108] N. P. Cuntoor and R. Chellappa, “Epitomic Representation of Human Activities”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[109] C. Sminchisescu, A. Kanaujia, Z. Li and D. Metaxas, “Conditional Models for Contextual Human Motion Recognition”, *IEEE International Conference on Computer Vision*, Volume 2, 2005.

[110] M. S. Ryoo and J. K. Aggarwal, “Recognition of Composite Human Activities through Context-Free Grammar Based Representation”, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Volume 2 , pp. 1709-1718 2006.

[111] W. Ying, H. Kaiqi and T. Tieniu, “Human Activity Recognition Based on R Transform”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[112] T. Chin, L. Wang, K. Schindler and D. Suter, "Extrapolating learned manifolds for human activity recognition", *IEEE International Conference on Image Processing*, 2007.

[113] L. Wang and D. Suter, "Recognizing Human Activities from Silhouettes: Motion Subspace and Factorial Discriminative Graphical Model", *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[114] D.Weinland, E. Boyer and Remi Ronfard, "Action Recognition from Arbitrary Views using 3D Exemplars", *IEEE International Conference on Computer Vision*, 2007.

[115] S. Nowozin, G. Bakır and K. Tsuda, "Discriminative Subsequence Mining for Action Classification", *IEEE International Conference on Computer Vision*, 2007.

[116] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J.M. Rehg, "A Scalable Approach to Activity Recognition based on Object Use", *IEEE International Conference on Computer Vision*, 2007.

[117] L. Li and F. Li, "What, where and who? Classifying events by scene and object recognition", *IEEE International Conference on Computer Vision*, 2007.