HUMAN MOTION ANALYSIS VIA AXIS BASED REPRESENTATIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SEZEN ERDEM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2007

Approval of the thesis:

HUMAN MOTION ANALYSIS VIA AXIS BASED REPRESENTATIONS

submitted by **SEZEN ERDEM** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Volkan Atalay _____
Head of Department, **Computer Engineering**

Assoc. Prof. Dr. Sibel Tari _____
Supervisor, **Computer Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Volkan Atalay _____
Computer Engineering Dept., METU

Assoc. Prof. Dr. Sibel Tari _____
Computer Engineering Dept., METU

Prof. Dr. Neşe Yalabık _____
Computer Engineering Dept., METU

Assoc Prof. Dr. Halit Oğuztüzün _____
Computer Engineering Dept., METU

Barış Üçüncü (MSc.) _____
Software Engineer, ASELSAN

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Sezen Erdem

Signature :

ABSTRACT

HUMAN MOTION ANALYSIS VIA AXIS BASED REPRESENTATIONS

Erdem, Sezen

M.S., Department of Computer Engineering

Supervisor : Assoc. Prof. Dr. Sibel Tari

September 2007, 67 pages

Visual analysis of human motion is one of the active research areas in computer vision. The trend shifts from computing motion fields to understanding actions. In this thesis, an action coding scheme based on trajectories of the features calculated with respect to a part based coordinate system is presented. The part based coordinate system is formed using an axis based representation. The features are extracted from images segmented in the form of silhouettes. We present some preliminary experiments that demonstrate the potential of the method in action similarity analysis.

Keywords: motion, articulated motion, human motion, human motion analysis, axial representations, disconnected skeleton

ÖZ

EKSEN TABANLI GÖSTERİM İLE İNSAN HARAKETLERİNİN GÖRSEL ÇÖZÜMLEMESİ

Erdem, Sezen

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi : Doçent Dr. Sibel Tari

Eylül 2007, 67 sayfa

İnsan hareketlerinin görsel çözümlemesi Bilgisayar Görüşü alanının önemli konularından birisidir. Bugünlerde hareket çözümlemesi konusundaki eğilim hareket alanlarının hesaplanmasından görüntüde ne oluyorun yorumlanmasına doğru kaymaktadır. Bu tez kapsamında, parça tabanlı koordinat sistemi kullanılarak elde edilen özellik yörüngelerine dayalı bir eylem tanıma yöntemi anlatılmaktadır. Parça tabanlı koordinat sistemi eksen tabanlı bir gösterime dayalı olarak oluşturulmuştur. Hareket çözümlemesinde kullanılan özellikler siluetler kullanılarak elde edilmektedir. Bu tez kapsamında, önerilen yöntemin eylem benzerliği çözümlemesi alanında kullanılabilirliğini gösteren denemeler sunulmuştur.

Anahtar Kelimeler: hareket,eklemli hareket, insan hareketleri, insan hareketlerinin çözümlemesi, eksensel gösterim, bağlantısız iskelet.

To My Family

ACKNOWLEDGMENTS

I would like to thank to Sibel Tari for her supervision, guidance and invaluable suggestions throughout the development of this thesis.

I would also like to thank to Erkut Erdem and Aykut Erdem for their support.

I would like to thank to my family for their belief on me.

I would like to thank to Çağrı Kayadelen for his psychological support.

I am thankful to my company ASELSAN Inc. for letting and supporting my thesis.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

In physics, motion is defined as a continuous change in the position of a body relative to a reference point as measured by a particular observer in a particular frame of reference. Motion is one of the most striking events for biological visual systems. Even in very low resolutions, human vision system can detect motion and recognize the performed action. The importance of visual motion stimulated a significant body of work in computer vision.

The framework of visual analysis of motion contains three steps: detecting motion, tracking motion and generating high level descriptions for the motion (Figure 1.1).

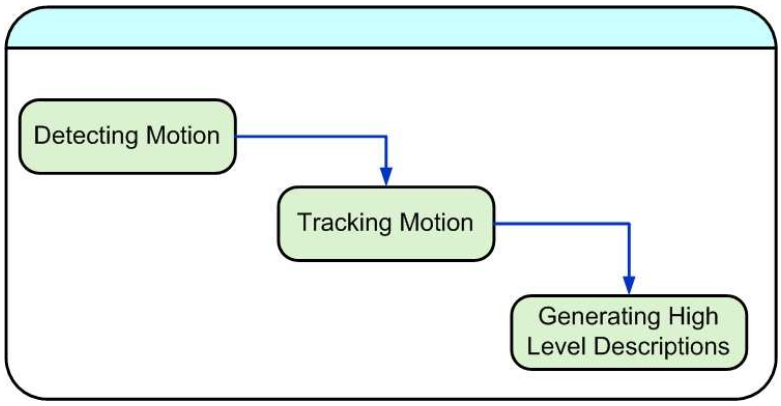


Figure 1.1: The framework of visual motion analysis

Motion detection means separating the moving objects in the scene from the rest of the image in each frame. The existing approaches use spatial or temporal data to extract motion. Various approaches are reviewed in Chapter 2.

Motion tracking is the process of locating the moving object in time on the scene.

Knowledge about the performed action is important to locate the moving object accurately. Such knowledge about the motion can be related with the kinematics of the object such as velocity and acceleration. The general idea in the approaches for motion tracking is to generate a dynamic model relating the measurements directly taken from motion data to pre-knowledge about the motion. Hence tracking problem is solved using dynamic system analysis methods which will be discussed in Chapter 2.

Generating high level descriptions for the motion is the interpretation of the spatiotemporal data. It is the step where the perception meets with the cognition. It may involve contextual data, artificial intelligence and natural languages to generate high level descriptions representing the motion. The generated descriptions can be used in action recognition to analyze and recognize motion patterns. The details of generating high level descriptions for the motion will be reviewed in Chapter 2.

Visual analysis of motion can be divided into groups as rigid and non-rigid motion according to the structure of the object of interest. (Figure 1.2)

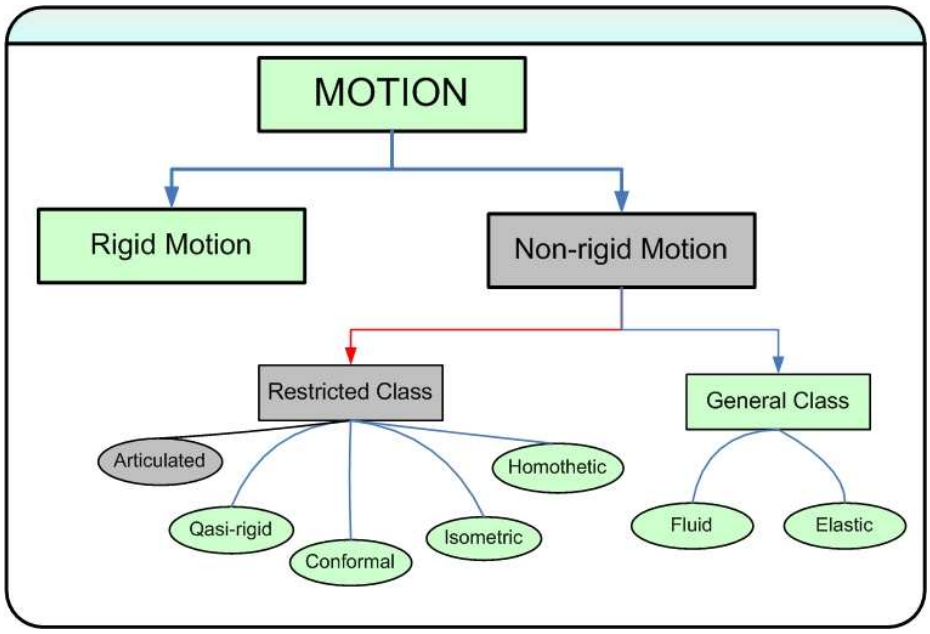


Figure 1.2: Motion classification

In rigid motion, the moving object performs an action as a whole. All of the points on the object move with the same dynamics. Therefore motion can be modeled with

single dynamic system.

In non-rigid motion, deformations and articulations are introduced. Different points on the object move with different dynamics. Therefore, non-rigid motion can not be modeled with a single dynamic system.

Articulated motion is a special case of non-rigid motion. The moving object is a composition of several rigid parts and joint points. The rigid parts can move independently at their joint points. If we fix a reference frame on the joint, the motion is rigid with respect to the local reference frame. Many of the biological systems including humans perform articulated motion.

Aggarwal et. al.[2] give the definitions of the remaining types of non-rigid motion as follows:

- Quasi-rigid Motion: It restricts the deformation to be small. A general motion is quasi-rigid when viewed in a sufficiently short interval of time.
- Isometric Motion: It is defined as motion that preserves the distances along the surface and the angles between the curves on the surface.
- Homothetic Motion: It is motion with a uniform expansion or contraction of the surface.
- Conformal Motion: It is non-rigid motion which preserves the angles between the curves on the surface, but not the distances.
- Elastic Motion: It is non-rigid motion whose only constraint is some degree of continuity or smoothness.
- Fluid Motion: It involves topological variations and turbulent deformations.

Visual analysis of human motion is one of the active research areas in computer vision. The interest on human motion analysis gained more importance with the development of wide spectrum applications such as visual surveillance, virtual reality, advanced man-machine interface, personalized training, content-based video storage etc. There is also an interest in human motion from wide variety of disciplines. In psychology, human movement is analyzed to understand human perception. Biomechanics analyze how human body functions and try to find ways of increasing movement efficiency. In choreography, high-level descriptions of human movements for

the notation of dance, ballet and theater are generated by using human motion analysis. In computer graphics, human motion analysis is used for generating realistic models for simulations, computer games and animations.

In early days of visual analysis of motion, the aim was to capture the motion in spatiotemporal video data. The need for developing intelligent systems leads to development of systems that can understand what is happening in a scene. As a result, generation of high level descriptions step gained more importance. In recent years, human motion analysis becomes a part of more general domain called as "Looking At People" which relates computer vision with artificial intelligence and natural languages. The systems should have the capability of extracting information from its environment independently rather than relying on externally supplied data. The systems should be able to understand actions taking place in the scene. The basic question changed from how things are moving to what is happening [3]. Human motion is described in terms of knowledge rather than geometric terms.

The approaches in human motion analysis can be classified into two groups based on the body structure analysis: model based and non-model based. There is a priori human body model in the model based approaches [4] [5] [6] [7] [8]. The images in the sequence are matched with the human model for motion analysis. The details of model based human motion analysis and proposed approaches in the literature are reviewed in Section 3.1. In non-model based approaches, there is no priori human body model [9] [10]. The images in the sequence are matched with each other to find the correspondences. The analysis is performed based on the data extracted from correspondences through the image sequences. The approaches in non-model based human motion analysis are reviewed in Section 3.2.

Using a priori model handles self-occlusions. However, the success of the motion analysis depends greatly on the selected priori model. The model should cover the essential part of the variations of human body poses in the sequence. In non-model based approaches, the features used in motion analysis have great importance. The features should be clear and exist in all frames of the sequence. They should be easily and accurately extractable. The evolution of the features during the sequence should reflect the performed action.

Model based and non-model based approaches for human motion analysis are

discussed in detail in Chapter 3.

After reviewing the proposed approaches and observing the trend in visual analysis of human motion, we introduce a non-model based human motion analysis scheme based on an axis-based representation of human body and change of axial features through the frames.

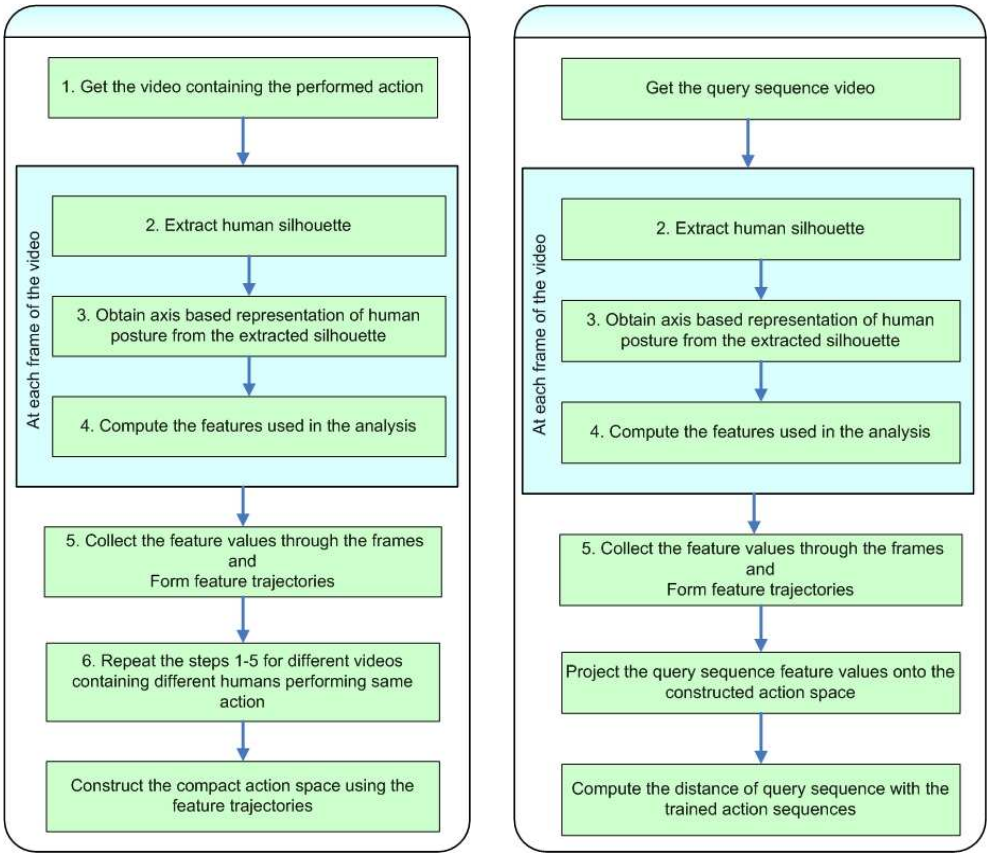
1.1 Motivation

As mentioned earlier, the methods based on using a predefined human model for analysis depend greatly on the constructed model. The model should be able to represent the variations of human posture during the course of an action. Constructing a human body model which covers an essential part of variations of postures is a difficult work. The methods which do not use a predefined model; non-model based approaches; perform motion analysis by matching the features extracted at each frame. The extracted features should be present at each frame and be easily and accurately extractable. The features should be both simple and dependable.

In this thesis, a non-model based human motion analysis scheme is proposed. Human motion analysis is based on the trajectories of features defined with respect to a part based coordinate system. Human postures are represented with a local symmetry axis representation called disconnected skeletons [1]. The symmetry branches are classified as positive and negative. A positive branch indicates an articulated part and always merges with a negative branch. Both of the branches terminate when they meet. The proposed scheme takes the advantage of this particular structure to present an action coding scheme. Trajectories of change in the features extracted from axis based representation generate patterns for performed actions. Representations for the actions are generated by using the patterns. The steps of generating representations for the actions are displayed in the Figure 1.3.

The generated patterns are used in comparing the similarity of actions. The flow diagrams of the similarity analysis are displayed in Figure 1.3.

Human body is a 3D structure. There are two possible approaches including the effects of the 3D body structure into the motion analysis. One of them is to use 3D models [11] [12] [13] [14]. The data collected from 3D model are matched with 2D data obtained from images which brings a non trivial complexity into the mo-



(a) (b)

Figure 1.3: Flow diagrams of analysis (a)Action space construction (b)Similarity analysis of actions

tion analysis. Second approach is to use special motion capture systems such as body markers to collect the data [15]. However, motion capture systems such as body markers requires special hardware and body markers are stuck on the human performing the action. Hence the motion analysis systems intervene into the action performance.

In our analysis scheme, 3D effects are neglected while obtaining motion data and analysis is performed on 2D silhouettes. However, the initially ignored 3D effects are revealed in analysis. We can capture the effects of 3D body structure such as self-occlusion and motion in the plane orthogonal to viewing plane.

In this work, we represent motion without using flow vectors or fields. We try to generate semantic descriptions for the performed actions rather than geometric

representations.

1.2 Thesis Organization

The rest of the thesis is organized as follows:

In Chapter2, the framework of visual analysis of motion is reviewed. The steps in the framework are discussed in detail. The approaches in the literature are summarized briefly to give a general idea about the visual analysis of motion. The ideas and methods construct a basis for visual analysis of motion.

In Chapter 3, the literature on visual analysis of human motion is reviewed. Model based and non-model based approaches are discussed and the details of some example works are explained to address where our work stands in human motion analysis literature.

In Chapter 4, the proposed human motion analysis scheme is presented. The axis based representation of human body is described. The features of the representation, the information carried in the features and the usage of the extracted information in the human motion analysis are explained in detail. The experimental results of the proposed scheme are displayed.

Summary and future works are discussed in Chapter 5.

1.3 Contributions

Main contributions of this thesis are as follows:

- An overview of the methods for human motion analysis are provided. (Chapter 3)
- A non model based articulated human motion analysis scheme based on an axis based representation of human posture is presented. The method is applied to images segmented as 2D silhouettes. The method has an important difference from other silhouette based approaches [10], in the sense that, initially ignored factors such as the effects of 3D body structure are revealed in the analysis. (Chapter 4)

CHAPTER 2

MOTION

Psychophysical researches suggest that biological visual systems are very sensitive to motion. This fact leads to many researches in computer vision area dealing with visual analysis of motion in spatiotemporal image data. There are many important applications of visual analysis of motion in various areas such as visual surveillance, robotics, tracking, gait analysis, etc. In recent years, due to the increase in computational power and decrease in cost, the demand for visual analysis of motion has increased.

Although there are various approaches in visual analysis of motion, they follow a general framework (Figure 1.1). The steps of the framework are as follows:

1. Detecting Motion
2. Tracking Motion
3. Generating High-Level Descriptions of Motion

In this chapter, the details of the steps in visual analysis of motion are explained. In Section 2.1, the approaches and the methods in detecting motion are discussed. In Section 2.2, the methods and some well known algorithms for tracking motion are reviewed. In Section 2.3, ideas in generating high level descriptions for motion are mentioned and action recognition is discussed.

The approaches and the methods reviewed in this chapter construct a basis for the visual analysis of motion. The sample video data that we used in our work contain a single human performing some specific actions and scene background is static. Therefore, in our work, *Detecting Motion* and *Tracking Motion* steps of the motion analysis are quite simple and we do not directly make use of many of the methods

reviewed in *Detecting Motion* and *Tracking Motion* sections. However, the methods reviewed in this chapter can be involved into our work in the future when we deal with more complex cases such as real world scenarios. The reviews are included for the sake of completeness of visual analysis of motion and uninterested reader may skip Section 2.1 and Section 2.2 without interrupting the readability of the thesis.

2.1 Detecting Motion

Moving objects are separated from the rest of the image in each frame in detecting motion. Each pixel in each frame is classified as either background or foreground to decide on moving objects. Lighting, shadows, weather changes, occlusions and many other factors make accurate motion detection a though problem.

There are several approaches for motion detection. They can be clustered into two general categories according to the information used: temporal or spatial. Temporal information based methods make use of differences between the images in the sequence to obtain movement data. In temporal information based methods, it is assumed that the background is static. Hence the differences between the images in the sequence originate from the motion of the objects. Spatial information based methods use some special features of individual pixels or groups of pixels such as color, intensity, edges, etc. to extract the motion.

The methods used in motion detection are explained in the following part.

2.1.1 Background Subtraction

Background subtraction is a popular temporal information based approach for motion detection. Moving objects are detected by differencing the current image from a reference background image. The difference image is further processed to extract the moving objects in the frame.

The background subtraction methods in the literature differ in modeling the reference background image. The simplest one is considering a static, non-adaptive background model (Figure 2.1). However non-adaptive background subtraction approaches need manual background model initialization. The background model has to be re-initialized periodically for the success of the approach. Median of the pixel values observed in an image sequence can be used as background model [16].

(a) (b) (c)

Figure 2.1: Static, non-adaptive background modeling approach (a)Input image (b)Background model (c)Output image

Background subtraction methods are very sensitive to changes in the dynamic scenes. A good background subtraction algorithm should adapt to various levels of illumination values at different times of the day, handle weather conditions (fog, rain, snow etc.), shadows, slow moving objects, objects that become background for a time and become foreground at a later time. However most of the background subtraction based methods can not satisfy these requirements. New methods including statistical data for background image construction are examined to solve the problems faced in background subtraction based approaches.

2.1.2 Temporal Differencing

Temporal differencing is another temporal information based approach for motion detection. In temporal differencing approach, pixel-wise differences between consecutive frames (generally two or three consecutive frames) of a sequence are used. Temporal differencing approaches can capture any change in the environment and are highly adaptive to dynamic environments. However they have a poor performance in extracting the entire relevant feature pixels. Generally they generate holes in the moving objects.

Lipton, Fujiyoshi and Patil [17] used temporal differencing in their work. Moving targets in real video streams are detected by using temporal differencing. Then, the extracted moving sections are clustered by using connected component analysis.

Another work is performed in VSAM [18]. A hybrid algorithm is developed for motion segmentation by combining an adaptive background subtraction algorithm with three frame differencing technique.

2.1.3 Optical Flow

Flow vectors or fields describe the apparent motion of points or features between image frames. Optical flow based motion segmentation approaches make use of flow vectors over time to detect moving objects. Bregler [19] represents each pixel by its optical flow. The flow vectors are grouped into blobs having coherent motion.

Optical flow methods are successful to extract coherent motion. However, obtaining accurate results with optical flow is difficult. Due to the aperture problem [20], noise sensitivity, difficulties with multiple moving objects and the complexity of the operations, optical flow based methods are not commonly used in applications.

2.1.4 Statistical Methods

In real world scenarios, the background of a scene does not stay constant. Illumination changes, lighting, shadows, noise and many other environmental factors affect the state of the background. The model should be updated to reflect very recent state of the background scene.

Statistical methods are used for constructing advanced background models by using characteristics of the pixels. A dynamic background model is constructed and this model is updated with the images from the sequence. Each pixel in the current image is compared with the dynamic background model and marked as either background or foreground.

Ridder, Munkelt and Kirchner [21] model each pixel with a Kalman Filter to have a robust system to scene lighting changes. Pfinder by Wren et al. [22] uses a single Gaussian model per pixel to model background. Single Gaussian is sufficient if the pixels are on a particular surface under a particular lighting. However in real life, multiple surfaces appear on a particular pixel and the lighting conditions change.

Stauffer and Grimson [23] use mixture of Gaussians for each pixel to model background. History of recent values of each pixel is modeled with a mixture of Gaussians. Each Gaussian in the mixture has a weight. At each time step, these weights are recalculated according to the current pixel values. After the recalculation of weights, the Gaussians in the mixture are evaluated for determining most likely ones belonging to the background model. Pixels not matching any of the background Gaussians are marked as foreground. These pixels are grouped by using connected component

analysis and construct the foreground objects (Figure 2.2).

(a) (b) (c)

Figure 2.2: Mixture Of gaussian background modeling approach (a)Input image (b)Obtained background model (c)Output image

In non-parametric background modeling approach of Elgammal,Harwood and Davis [24], a sample for each pixel of the scene is stored. The method has two stages. In the first stage, the probability that a new pixel belongs to background is estimated non-parametrically. Small motions in the scene background lead to false detections. In the second stage, the false detections are suppressed. The pixels detected as foreground in the first stage are queried to check whether they belong to the background distribution of some points in their neighborhood. The objective of the constructed background model is to capture very recent information about the image sequence. This model is continuously updated to capture fast changes in the scene background (Figure 2.3).

(a) (b)

Figure 2.3: Non-parametric model for background subtraction (a)Input image (b)Output image

At the study of Haritaoglu et al. [25], the maximum and the minimum values a pixel gets during training period and the maximum intensity difference between

consecutive frames are collected for each pixel. The data construct the background model and it is updated periodically to reflect changes in the scene.

Statistical methods are becoming increasingly popular. They are robust to changes in the background and they produce better results compared with other motion detection methods.

In this section, the approaches and the methods in detecting motion step of visual analysis of motion framework are reviewed. The next step is tracking motion. The approaches in tracking motion are reviewed in the following section.

2.2 Tracking Motion

The moving object is located on the scene in motion tracking. It may be thought that tracking can be done by just finding and extracting the moving object at each frame. However, it is not the case. We should have knowledge about the performed action to locate the moving object accurately. Tracking can not be based on only the measurement data obtained directly from image data. The measurements contain errors due to noise, occlusions, shadows and etc. The knowledge about the motion should be included into the tracking process to get dependable results.

Generally, kinematics of the moving object such as velocity and acceleration are used in tracking motion. The performed motion is represented with a dynamic system model. The tracking problem is reduced to dynamic system analysis problem.

Dynamic system analysis requires two models: system model and measurement model. The system model describes the evolution of a dynamic system with time (Equation 2.1). The measurement model is used for relating the noisy measurement data to the dynamic system (Equation 2.2). The state of the dynamic system is estimated by using the noisy measurements.

System Model:

$$x_t = f(x_{t-1}, v_{t-1}) \tag{2.1}$$

where f is possibly a non-liner function of x_{t-1} and v_{t-1} is independently and identically distributed noise.

Update Model:

$$z_t = h(x_t, w_t) \tag{2.2}$$

where h is possibly a non-liner function and w_t is independently and identically distributed measurement noise.

In motion tracking, system and measurement models are generally represented in a probabilistic form. Tracking is a prediction process of the states of the dynamic system via Bayesian approach [26]. The aim is to construct the posterior probability density function of the state by using available information. The prediction is repeated as new measurements are received. This approach is called as recursive filtering [26] in statistics and convenient for dynamic system analysis.

The recursive filtering approaches contain two steps. The first step is the prediction. The state of the system is predicted using the system model. The next step is the update. The prediction is updated with the latest measurements. Tracking problem can be considered as the evolution of the states of a moving object.

The process can be described as calculating some degree of belief on the state (x_t) at time (t) based on the measurements (z_t) up to time (t) in Bayesian approach. The probability distribution function ($p(x_t, z_{1:t})$) can be obtained recursively by following prediction and update steps. The following part explains the details of the prediction process via Bayesian approach.

Assume that the posterior pdf ($p(x_{t-1}, z_{1:t-1})$) at time ($t-1$) is available. The prediction step is computed using the system model to obtain the prior distribution at time t. The prior distribution can be computed using Chapman - Kolmogorov equation (Equation 2.3).

$$p(x_t|z_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})dx_{t-1} \tag{2.3}$$

The prior distribution is updated using the measurements (z_t) via Bayes' rule and the posterior distribution ($p(x_t|z_{1:t})$) is obtained (Equation 2.4).

$$p(x_t|z_{1:t}) = \frac{p(z_t|x_t)p(x_t|z_{1:t-1})}{p(z_t|z_{t-1})} \tag{2.4}$$

The recursive estimation of the posterior density generally provides a conceptual solution to the problem instead of an analytical solution. Several different approaches are proposed for the analytical solution of the problem. The following part describes some of these approaches.

2.2.1 g-h Filters

The g-h filters are one of the dynamic system analysis methods and generally used in tracking applications on radars [27]. The g-h filters assume that the object to be tracked moves with a constant speed. The system model of the dynamic system is as Equation 2.5 and Equation 2.6.

$$x_t = x_{t-1} + v_{t-1}T \tag{2.5}$$

$$v_t = v_{t-1} \tag{2.6}$$

where x_t denotes location, v_t denotes velocity and T denotes time interval.

At time step t, the measurements (y_t) are obtained from the system. There are three cases to be considered:

- if $x_t < y_t$ then the speed should be increased
- if $x_t = y_t$ then no need for update
- if $x_t > y_t$ then the speed should be decreased

After considering the update conditions, the velocity should be updated as Equation 2.7.

$$v_t = v_t + h_t(y_t - x_t) \tag{2.7}$$

where h_t is small parameter.

The problem with the Equation 2.7 is that both updated velocity estimate after the measurement and the velocity estimate before measurement are represented with v_t. This ambiguity can be cleared with a new representation as follows:

$$\dot{v}_{t,t} = \dot{v}_{t,t-1} + h_t(y_t - \dot{x}_{t,t-1}) \tag{2.8}$$

where $\dot{v}_{t,t-1}$ represents the predicted estimate and $\dot{v}_{t,t}$ represents updated estimate. Similarly the updated position can be rewritten as Equation 2.9.

$$\dot{x}_{t,t} = \dot{x}_{t,t-1} + g_t(y_t - \dot{x}_{t,t-1}) \tag{2.9}$$

where g_t is small parameter. With this new notation, the system model of the dynamic system becomes:

$$\dot{v}_{t+1,t} = \dot{v}_{t,t} \tag{2.10}$$

$$\dot{x}_{t+1,t} = \dot{x}_{t,t} + \dot{v}_{t,t}T = \dot{x}_{t,t} + \dot{v}_{t+1,t}T \tag{2.11}$$

And update model of the dynamic system becomes:

$$\dot{v}_{t+1,t} = \dot{v}_{t,t-1} + \frac{h_t}{T}(y_t - \dot{x}_{t,t-1}) \tag{2.12}$$

$$\dot{x}_{t+1,t} = \dot{x}_{t,t-1} + T\dot{v}_{t+1,t} + g_t(y_t - \dot{x}_{t,t-1}) \tag{2.13}$$

2.2.2 g-h-k Filters

Another dynamic system analysis method is g-h-k filters. The assumption in the approach is that the moving object travels with a constant acceleration [27]. The system dynamic model is described with Equation 2.14, 2.15 and 2.16.

$$x_t = x_{t-1} + v_{t-1}T + a_{t-1}\frac{T^2}{2} \tag{2.14}$$

$$v_t = v_{t-1} + a_{t-1}T \tag{2.15}$$

$$a_t = a_{t-1} \tag{2.16}$$

where x_t denotes location, v_t denotes velocity, a_t denotes acceleration and T denotes time interval. By following the same heuristic procedure pursued in the g-h filters, the system and update models are represented as follows:

System Model

$$\dot{a}_{t,t} = \dot{a}_{t,t-1} + \frac{2k}{T^2}(y_t - \dot{x}_{t,t-1}) \tag{2.17}$$

$$\dot{v}_{t,t} = \dot{v}_{t,t-1} + \frac{h}{T}(y_t - \dot{x}_{t,t-1}) \tag{2.18}$$

$$\dot{x}_{t,t} = \dot{x}_{t,t-1} + g(y_t - \dot{x}_{t,t-1}) \tag{2.19}$$

Update Model

$$\dot{a}_{t+1,t} = \dot{a}_{t,t} \tag{2.20}$$

$$\dot{v}_{t+1,t} = \dot{v}_{t,t} + \dot{a}_{t,t}T \tag{2.21}$$

$$\dot{x}_{t+1,t} = \dot{x}_{t,t} + \dot{v}_{t,t}T + \dot{a}_{t,t}\frac{T^2}{2} \tag{2.22}$$

However in real world, targets generally do not move with either constant velocity or constant acceleration. Hence more advanced methods are introduced to deal with the nonuniformity of kinematics. The following part describes some of the advanced methods used in dynamic system analysis.

2.2.3 Kalman Filters

Kalman Filter is a recursive filter used in dynamic system state estimation. Kalman filter is based on linear dynamical systems which are discretized in the time domain [27].

The Kalman Filter has two basic assumptions:

- The posterior density always has a Gaussian distribution
- The measurement errors are independent and normally distributed

When these assumptions are assured, the Kalman Filter is the optimal solution for tracking problem. It produces the minimum mean squared error estimate of the system state.

The dynamic system equations (Equation 2.1 and 2.2) can be written as

$$X_t = F_{t-1}X_{t-1} + W_{t-1} \tag{2.23}$$

$$Z_t = H_tX_t + V_t \tag{2.24}$$

where

X_t is state vector

F_{t-1} is state transition matrix

W_{t-1} is zero mean, white gaussian error with covariance $Q(t)$

Z_t is measurement vector

H_t is transition matrix

V_t is zero mean, white gaussian error with covariance $R(t)$

F_{t-1} and H_t are matrix representation of linear functions.

The Kalman Filter estimates recursively the state vector (X_t) based on the measurements (Z_t) taken at time (t=1...n).

The equations of the recursive process are as follows:

Innovation:

$$Y_t = Z_t - H_t\tilde{X}_t \tag{2.25}$$

Estimate :

$$\hat{X}_t = \tilde{X}_t + K_tY_t \tag{2.26}$$

Prediction :

$$\tilde{X_{t+1}} = F_t\hat{X}_t \tag{2.27}$$

Kalman Gain :

$$K_t = P_tH_t^T[H_tP_t\tilde{H_t^T} + R_t]^{-1} \tag{2.28}$$

Estimate Covariance :

$$\hat{P}_t = [I - K_tH_t]\tilde{P}_t \tag{2.29}$$

Prediction Covariance :

$$\tilde{P_{t+1}} = F_t\hat{P}_tF_t^T + Q_t \tag{2.30}$$

where P is the covariance matrix of the error of X_t. The mean and the covariance of Gaussian posterior are computed recursively.

The assumptions of Kalman Filter do not hold in many situations. It may not be possible to express the problems with the linear equations. If either the system or the update model is non-linear, the posterior distribution will not have a Gaussian distribution. Hence we can not obtain an optimal solution by Kalman Filter. The following part describes an approach based on the Kalman Filter dealing with the non-linear situations.

2.2.4 Extended Kalman Filters

Extended Kalman Filters (EKF) are based on local linearization of nonlinearity in the system and the measurement models of the dynamic system [27]. The dynamic system equations of Kalman Filter (Equation 2.23 and 2.24) can be written as

$$X_t = f_{t-1}X_{t-1} + w_{t-1} \tag{2.31}$$

$$Z_t = h_t X_t + v_t \tag{2.32}$$

where f_{t-1} and h_t are non-linear functions and v_{k-1} and w_k are mutually independent, zero mean white Gaussians. The non-linear functions f_{t-1} and h_t are approximated by using first order Taylor expansion. The Extended Kalman Filter approach assumes that local linearization of f_{t-1} and h_t are sufficient to describe nonlinearity of the dynamic system. The result of EKF is Gaussian approximation of the posterior density function. If f_{t-1} and h_t are non-linear functions which can not be approximated accurately with a Gaussian distribution, EKF approaches will fail.

By the nature of the tracking motion problem, we have to deal with nonlinearities. Using the Kalman Filter approach may not solve the problems containing multi model densities. Even using the Extended Kalman Filter approach may lead to complex computations and poor results. At this point, making use of simulation based approaches such as Sequential Monte Carlo methods may help to deal with multimodal, non-linear systems. The Sequential Monte Carlo methods are sophisticated model estimation techniques based on random sampling. The next section reviews Particle Filters which is an example of Monte Carlo methods.

2.2.5 Particle Filters

Particle Filter is one of the Sequential Monte Carlo Filters. The Sequential Monte Carlo methods are simulation based methods developed for computing posterior distributions [28]. The idea is to represent the posterior density function with a set of random samples and associated weights. The estimates are computed based on the samples and weights.

Suppose that multi-dimensional integral in Equation 2.33 is computed numerically.

$$I = \int g(x)dx \tag{2.33}$$

The Monte Carlo methods factorize $g(x)$ as Equation 2.34.

$$g(x) = f(x)\pi(x) \tag{2.34}$$

where $\pi(x)$ is a probability density function such that ($\pi(x) \geq 0$) and ($\int \pi(x)dx = 1$). The Monte Carlo estimation of integral in Equation 2.35 can be computed via drawing $N \gg 1$ samples $\{x^i; i = 1, ...N\}$ from $\pi(x)$.

$$I = \int f(x)\pi(x)dx \tag{2.35}$$

The estimate is as Equation 2.36.

$$I_N = \frac{1}{N}\sum_{i=1}^{N} f(x^i) \tag{2.36}$$

In fact, it is possible to represent the functional description of the posterior probability distribution function equivalently with the enough number of samples. In ideal case, the samples should be generated directly from $\pi(x)$. However it is not generally possible to sample from $\pi(x)$ since it is a multivariate distribution. One of the known solution to this problem is to use importance sampling. In the following part, two algorithms based on importance sampling for tracking motion is explained.

2.2.5.1 Sequential Importance Sampling (SIS) Algorithm

Sequential Importance Sampling (SIS) Algorithm is a Monte Carlo method which estimates the state of a dynamic system using the noisy measurements taken at discrete time intervals [28]. The state of the system is represented by an unknown probability distribution function. A Bayesian approach is used to estimate the state of the dynamic system. The posterior probability density function of the state is constructed by using available data which also includes the noisy measurements. As mentioned in the previous part, it is not always possible to directly sample from $\pi(x)$. To overcome this problem, the samples are taken from another distribution $q(x)$ which is similar to $\pi(x)$. The only assumption is that $q(x)$ and $\pi(x)$ have same support (Equation 2.37).

$$\pi(x) > 0 \Rightarrow q(x) > 0 \forall x \in R \tag{2.37}$$

The distribution, $q(x)$, is called importance density. Monte Carlo estimation of the posterior density of the state is still possible by weighting samples correctly.

The details of the SIS Algorithm are as follows:

Notation

x : States of the system

z : Measurements

$q(x_{0:k}|z_{1:k})$: Importance density (approximation of the posterior)

$p(x_{0:k}|z_{1:k})$: Posterior pdf

w : Weights associated to samples

Assumptions

- The proposal distribution (importance density $q(x_{0:k}|z_{1:k})$) is known
- State transition depends only on the previous state
- The proposal distribution factorizes such that

$$q(x_{0:k}|z_{1:k}) = q(x_{0:k-1}|z_{1:k-1})q(x_k|x_{0:k-1}|z_{1:k}) \tag{2.38}$$

- The measurements are independent of the observations

At each iteration step

We have N samples and their associated weights (x^i_{k-1}, w^i_{k-1}, i=1...N) approximating $p(x_{0:k-1}|z_{1:k})$ and we want to approximate $p(x_{0:k}|z_{1:k})$ with new set of samples.

- **Step 1: Resample with replacement**
  Generate N samples from the old sample set according to their associated weights. By this way, we can pick the samples having higher weight value more than once. Generate the sample set : (x^i_k; i = 1...N)
- **Step 2: Predict the generated samples by using proposal distribution**
  In this step the system model of the dynamic system is used for prediction.

$$q(x_k|x_{k-1} = x^i_k) \tag{2.39}$$

The random terms in the system model of the dynamic system introduces variations.

- **Step 3: Associate a weight w_i to each generated sample by using measurement data**

  The weight represents the probability of the sample fitting the posterior density.

$$w_i^k = p(z_{0:k}|x_{0:k} = x_{0:k}^i) \tag{2.40}$$

  There are two options for obtaining the associated weights.

  Option 1 By considering Equation 2.40, the weights can be directly estimated from the observation.

$$w_i^k = \frac{e^{E(x_k^i, x_k)}}{\sum e^{E(x_k^i, x_k)}} \tag{2.41}$$

  In Equation 2.41, Energy E() is selected such that it is lower for more probable states.

  Option 2 The associated weights (w_k^i) can be obtained by using the weights (w_{k-1}^i) obtained in $(k-1)_{th}$ iteration. The process is as follows:

$$w_k^i = \frac{p(x_{0:k}^i|z_{1:k})}{q(x_{0:k}^i|z_{1:k})} \tag{2.42}$$

  We should find $p(x_{0:k}|z_{1:k})$ term to compute the weight. By using Bayesian approach, $p(x_{0:k}|z_{1:k})$ can be expressed in terms of $p(x_k|x_{k-1})$, $p(z_k|x_k)$ and $p(x_{0:k-1}|z_{1:k-1})$ as Equation 2.43-2.46.

$$p(x_{0:k}|z_{1:k}) = \frac{p(z_k|x_{0:k}|z_{1:k-1}).p(x_{0:k}|z_{1:k-1})}{p(z_k|z_{1:k-1})} \tag{2.43}$$

$$p(x_{0:k}|z_{1:k}) = \frac{p(z_k|x_{0:k}|z_{1:k-1}).p(x_k|x_{0:k-1}|z_{1:k-1})}{p(z_k|z_{1:k-1})}.p(x_{0:k-1}|z_{1:k-1}) \tag{2.44}$$

$$p(x_{0:k}|z_{1:k}) = \frac{p(z_k|x_k).p(x_k|x_{k-1})}{p(z_k|z_{k-1})}.p(x_{0:k-1}|z_{1:k-1}) \tag{2.45}$$

  Omit normalization term

$$p(x_{0:k}|z_{1:k}) = p(z_k|x_k).p(x_k|x_{k-1}).p(x_{0:k-1}|z_{1:k-1}) \tag{2.46}$$

By replacing this formula to Equation 2.42, calculate the weights.

$$w_k^i = \frac{p(z_k|x_{0:k}^i).p(x_k^i|x_{k-1}^i).p(x_{0:k-1}^i|z_{1:k-1})}{q(x_k^i|x_{0:k-1}^i, z_k)q(x_{0:k-1}|z_{1:k})} \tag{2.47}$$

$$w_k^i = w_{k-1}^i.\frac{p(z_k|x_{0:k}^i).p(x_k^i|x_{k-1}^i)}{p(z_k|x_k^i).p(x_k^i|x_{k-1}^i)} \tag{2.48}$$

Degeneracy Problem

In perfect case, the importance density and the posterior density should be same. Unfortunately, it is not possible. This leads to problems in the process. One of the problems is so called Degeneracy Problem caused by the increase of the variance of the importance weights. After a certain number of recursive steps, all but one particle will have negligible weight [29]. In order to measure the degeneracy of the algorithm, effective sample size (N_{eff}) is introduced (Equation 2.49). If the effective sample size (N_{eff}) is small, there is a severe degeneracy.

$$N_{eff} = \frac{N_s}{1 + Var(w_k^{*i})} \tag{2.49}$$

where $w_k^{*i} = \frac{p(x_k^i|z_{1:k})}{q(x_k^i|x_{k-1}^i,z_k)}$ and $1 \leq N_{eff} \leq N$.

There are approaches for solving degeneracy problem. One of the proposed approaches is resampling. Resampling eliminates the samples having low importance weights and multiplies the samples having high importance weights. Approximate discrete representation of $p(x_k|z_{1:k})$ is resampled N times to generate a new sample set. The generated set is an identically and independently distributed and the weights are reset to $w_k^i = \frac{1}{N_s}$.

Another approach to solve the degeneracy problem is good choice of the importance density. Choosing an optimal importance density $q(x_k|x_{k-1}^i, z_k)$ minimizes the variance of w_k^{*i}. It has been shown that the optimal importance density function can be obtained as Equation 2.50 [30].

$$q(x_k|x_{k-1}^i, z_k)_{opt} = p(x_k|x_{k-1}^i, z_k) = \frac{p(z_k|x_k|x_{k-1}^i)p(x_k|x_{k-1}^i)}{p(z_k|x_{k-1}^i)} \tag{2.50}$$

This importance density is optimal because for a given x_{k-1}^i, w_k^i takes the same value whatever sample is drawn from $q(x_k|x_{k-1}^i, z_k)_{opt}$. Hence the variance of w_k^{*i} is equal to 0.

Both of the proposed solution to the degeneracy problem have some drawbacks and lead to new problems. In the resampling approach, the particles with the high importance weights are statistically selected many times which leads to loss of diversity among the particles. Choosing optimal importance density requires to sample from $p(x_k|x_{k-1}^i, z_k)$ and to evaluate the integral over the new state. Generally doing either of the operations may not be possible.

2.2.5.2 Conditional Density Propagation (CONDENSATION) Algorithm

Conditional Density Propagation (CONDENSATION) Algorithm is used for detecting and tracking the contour of the objects moving in a cluttered environment. Condensation applies factored sampling iteratively to the successive images in a sequence [31]. Since the algorithm is iteration of factored sampling over time, the output is a set of weighted samples $s_t^n, w_t^n, n = 1, ..., N$ approximating the state density $p(x_t|Z_t)$.

The steps of the algorithm are as follows (Figure 2.4):

- At the beginning of each time step, the sample set representation of the previous time step $p(x_{t-1}|Z_{t-1})$ is available as $[s_{t-1}^n, w_{t-1}^n, n = 1, ..., N]$. The prior density approximation is obtained by sampling N times from s_{t-1}^n. In this step, the samples with high importance weights can be selected more than once. The sample size is fixed with N so that the running time of the algorithm is fixed.

- Each element of the new sample set goes into predictive step. First, each element undergoes drift. The drift part of the predictive step is deterministic. Hence identical elements in the set undergo same drift. Then, diffusion part of the predictive step is executed. The diffusion part contains random features. Each element undergoes its own independent Brownian motion step.

- After the previous steps, the sample set (s_t^n) is generated without the associated weights. In measurement step, the importance weights are computed from the measurement density $p(z_t|x_t)$. Finally the sample set representation $[s_t^n, w_t^n]$ of the state density $p(x_t|Z_t)$ is obtained.

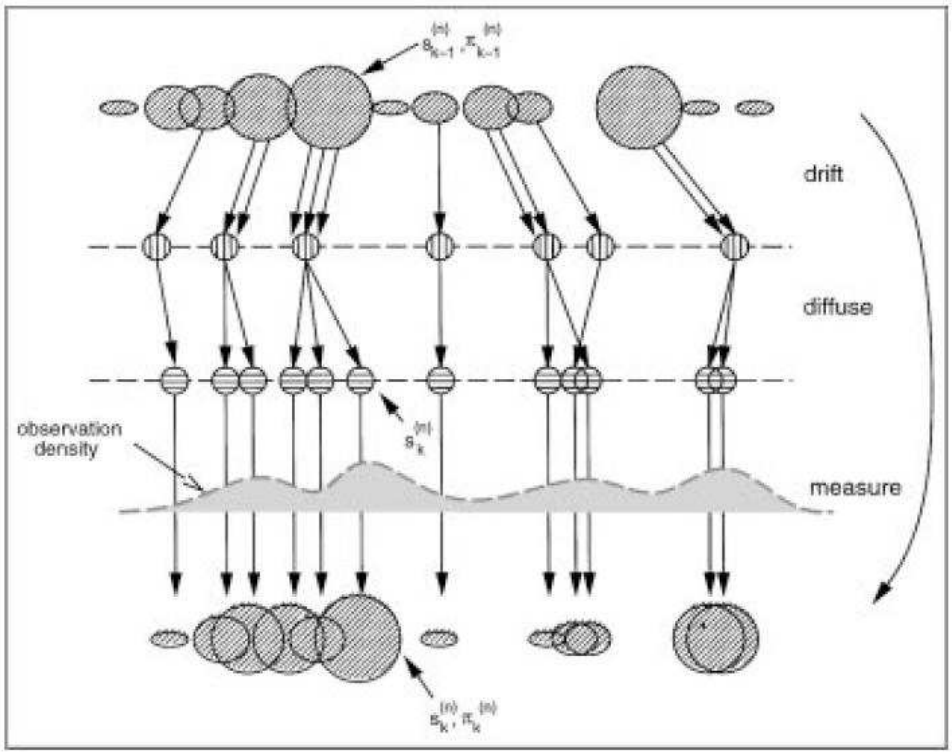


Figure 2.4: One time step in the CONDENSATION Algorithm

In this section, the approaches in the detecting motion step of the visual analysis of motion framework are discussed. Some of the methods and the algorithms are explained in details to explain the basis of the tracking motion process. In the next section, the final step of visual analysis of motion framework, Generating High Level Descriptions of Motion, is reviewed.

2.3 Generating High Level Descriptions of Motion

Final step of the framework is generating high level descriptions for motion. The previous steps (*Detecting Motion* and *Tracking Motion*) make use of only the information obtained directly from the spatiotemporal image data. Some statistical foundations

are applied on the image data. High level description generation step introduces interpretation of the extracted information. The previous steps can be considered as part of perception process while this step is the point where the cognition process begins. Artificial intelligence, natural languages and computer vision work in a collaboration to describe the motion in a higher level with the daily used words.

The idea of representing visual motion with the words in natural languages is the subject of many researches in the literature. One of them is the motion classification work of Nagel [32]. Nagel classified motion into five levels: *change*, *event*, *verb*, *episode* and *history* . *Change* is described as a discernable motion in a sequence. An *event* is a change which is considered as a primitive of more complex descriptions. *Verb* describes an activity, *episode* describes complex motions involving several actions and *history* is the extended sequence of related activities. The terms are related with story understanding. The aim is to obtain conceptual descriptions by using natural language.

Bobick [3] introduced an alternative classification for motion. The categorization is as *movement*, *activity* and *action*. *Movement* is described as the consistent motion characterized by a definite space - time trajectory. The only required knowledge is the motion. *Activity* is the statistical sequences of the movements. *Action* relates the semantic primitives to the context of the motion.

Gonzales et. al. [5] proposed a taxonomy used in analysis of human motion. It is a combination of the taxonomies of Nagel [32] and Bobick [3]. Motion is classified into four levels: *movement*, *action*, *activity* and *situation*. *Movement* represents a change in the posture or location of human. *Movement* is not related with the context. *Action* is a temporal series of human movements (running, jumping, bending and etc.). *Activity* is defined as a sequence of one or several human actions. *Sequence* relates the contextual information with the motion. It is defined as an activity that acquires a meaning in a specific scene.

Generated descriptions can be used in recognizing, predicting or querying the actions. Action recognition is one of the most popular application area in which high level motion descriptions are widely used.

2.3.1 Action Recognition

The aim of action recognition is to analyze and recognize motion patterns and produce a high level description of actions and interactions. In order to analyze and recognize the motion patterns, typical actions should be represented as reference sequences. Then test sequences are compared with the reference sequences for recognizing action. The approaches in action recognition are classified into two groups: template matching and state space approaches.

Template matching approaches convert image sequences into static shape patterns. The patterns are compared with the stored action prototypes to recognize the performed action.

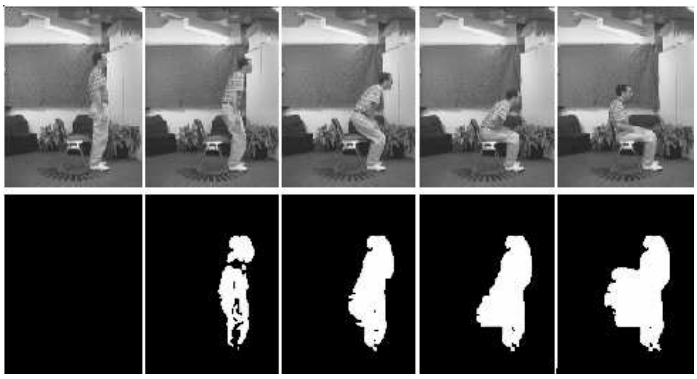
Polana and Nelson [33] utilized low level features of motion for the recognition of activities. A spatio-temporal motion magnitude template is used as a basis for activity recognition. Optical flow fields between successive frames are computed to segment and track the walking actor. The flow frames are divided into spatial grids and motion magnitude of each grid cell is accumulated to form the feature vector used in activity recognition.

The work of Bobick and Davis [34] use template matching to represent and recognize actions. In their work, Motion Energy Images (MEI) representing where motion occurs in the image sequence and Motion History Images (MHI) representing how motion in the image sequence is moving are computed (Figure 2.5).

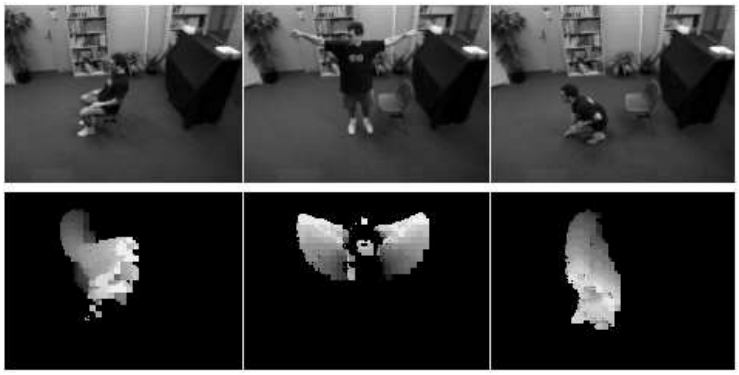
MEI and MHI are then used for constructing a template. The system is trained with image sequence samples of the actions obtained from a variety of viewing angles. Then statistical descriptions of MEI and MHI for each view/action combination are computed using moment based features. The moment descriptions of the motion is obtained. Action recognition process is done via computing Mahalanobis distance between the input moment description and the prestored known actions.

Template matching approaches have simple implementation and do not involve complex computations. However they are viewpoint dependent and sensitive to noise and variations of time intervals of movements.

State space approaches define each static posture as a state. Based on certain probabilities, transitions between the states are defined. Motion sequence is a set of state transitions between the states. State space approaches have a wide usage in the



(a)



(b)

Figure 2.5: (a)Motion energy images (b)Motion history images

analysis of temporal series. Hidden Markov Model (HMM) [35] is the most representative method used in analysis. HMM is used in construction of the states and transition rules. Yamato, Ohya and Ishii [36] introduce a human action recognition method based on HMM.

State space approaches solve some of the problems of template matching approaches. However state space approaches involve complex iterative computations. Selecting proper number of state to represent the static postures and dimensions of feature vectors describing states are difficult issues.

CHAPTER 3

HUMAN MOTION ANALYSIS

The general framework of visual analysis of motion is discussed in the previous chapter. In this chapter, we will examine human motion.

Motion can be can be divided into two groups according to body structure of the object of interest (Figure 1.2):

- **Rigid Motion:** The moving object preserves all distances and angles during the action. The deformations are neglected. The motion can be represented with a single dynamic system and all parts of the moving object perform the action according to same dynamic system.
- **Non-Rigid Motion:** Deformations, articulations and bending are introduced into the motion analysis. The parts of the moving object may perform different actions independent from each other.

Articulated motion is a special kind of non-rigid motion. It may be considered as a piecewise rigid-motion. The moving object is a composition of several rigid body parts connected at joint points. The rigid parts perform rigid motion. However, the overall motion is not rigid. Humans perform articulated motion.

Human motion analysis can also be divided into different categories according to viewing aspects such as single or multi camera capturing, monocular or stereo vision, stationary or moving camera motion analysis. The most common classification is based on body structure analysis. The works in the literature can be classified as model based and non-model based.(Figure 3.1)

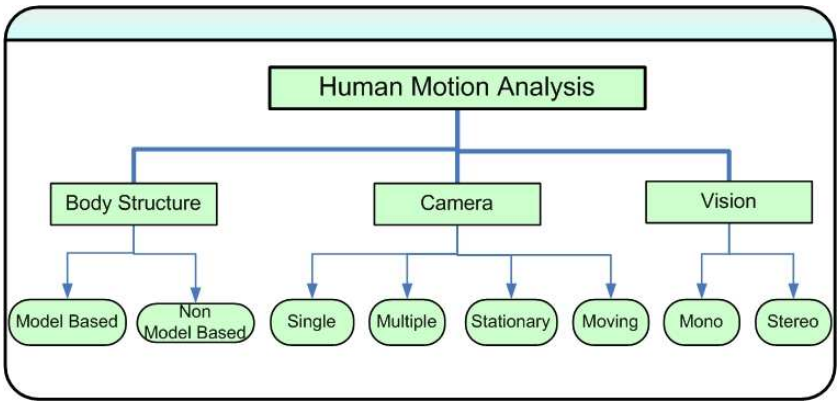


Figure 3.1: Human motion analysis classifications

The general framework for both model and non-model based approaches involves the following steps: feature extraction, feature correspondence and high level processing (Figure 3.2).

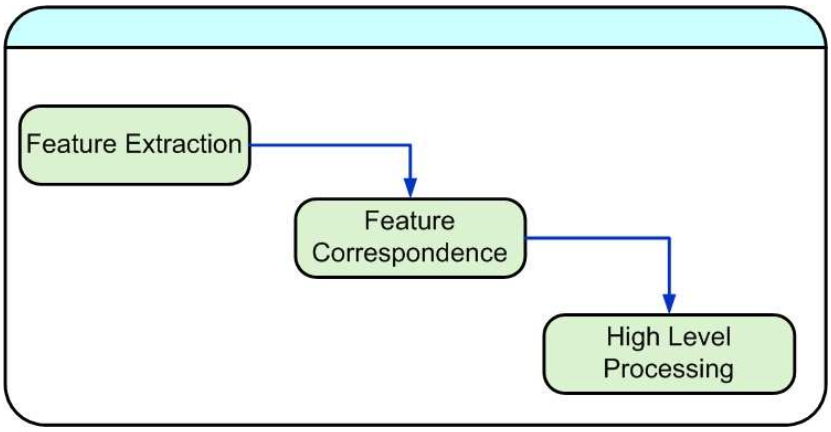


Figure 3.2: The framework of visual analysis of human motion

Model based and non-model based approaches differ only in feature correspondence process. At model based human motion analysis, there is a priori model and image sequences are matched to model. At non-model based approaches, estimation of features such as position, joint point angles, color, texture is used to find correspondence between successive frames.

In this chapter, the approaches in human motion analysis are reviewed. The work in the literature based on model and non-model based approaches are reviewed. In Section 3.1, model based approaches are explained. In Section 3.2, non-model based

approaches are examined. Finally, the application areas of human motion analysis and the sample applications are summarized in Section 3.3.

3.1 Model Based Approaches

There is a priori model to represent the observed object in model based human motion analysis approaches. The images in the sequences are matched with the priori model for feature correspondence. The models are generally represented as stick figures, 2D contours or 3D volumetric models. Type of the model used in modeling affects the features used in analysis.

The simplest representation of the human body is obtained by using stick figures. The human body is represented as line segments and joints representing the articulating features. The motion at the joint points provide the motion information of whole figure.

In ASpaces work [5] [6], Gonzales et al. represent human posture as a stick figure which is a composition of ten rigid body parts. The rigid parts are connected to each other by joints in a hierarchial manner.

In the work of Akita [4], it is assumed that the movement of the human is known a priori as a set of key frames which are stick figure representations of the human poses.

In 2-D representations, the priori model is usually a stick figure wrapped around with ribbons or blobs.

Niyogi and Adelson [7] [8] use deformable structures (contours in X-T space and surfaces in X-Y-T space) to find human silhouettes. They analyze the patterns generated by the human motion to estimate the parameters of a simple stick figure model. The generated model is used for gait recognition.

Wren et al. [22] model and track human body using a set of blobs in real-time person finder system (Pfinder)(Figure 3.3).

The blobs are described with spatial coordinates (x,y) and color components (Y,U,V). Each blob correspond to a body part of human such as head,hands and feet. Pfinder system uses a 2D contour shape analysis to identify the body parts. Tracking is accomplished by following an analysis loop. The analysis loop predicts the appearance of the human in the new image then resolves the membership of each pixel to one of

Figure 3.3: Blob representation in Pfinder.

the blobs and updates the statistical model.

In 3-D models, human body generally is represented with a skeletal structure and a flesh surrounding the skeletal structure. The skeletal structure generally is a stick figure which is a collection of segments and joints. The flesh surrounding the skeletal structure can either be surface based (polygons) or volumetric (cylinders).

Badler and O'Rourke [11] use overlapping spheres for modeling human. There is a trade-off between the accuracy of the model and the number of parameters used in modeling. In the general human motion analysis framework, it is not necessary to model the human body very accurately. The focus should be on the overall motion.

In the following part, details of *ASpaces* of Gonzales et al. [5] will be discussed as an example for model-based human motion analysis which uses features of stick figure representations of human posture for motion analysis.

3.1.1 ASpaces

In *ASpaces* [5] [6], Gonzales et al. consider human actions as a sequence of postures. Human postures are represented with stick figures. The stick figures are described in polar coordinates to handle the non-linearities of posture variations. The work is described as a human action recognition method in a generic image sequence evaluation framework. Mixture of knowledge based classifications of Nagel [32] and Bobick [3] is used in analysis and recognition of human motion (Section 2.3).

The first step of the analysis is training phase. Postures of different humans are used in building action spaces in training phase. Each human posture is represented as a stick figure with ten rigid body parts (torso, head, two primitive part for each arms, two primitive part for each leg) and joints connecting the parts. Body parts are connected by joints in a hierarchical manner (Figure 3.4).

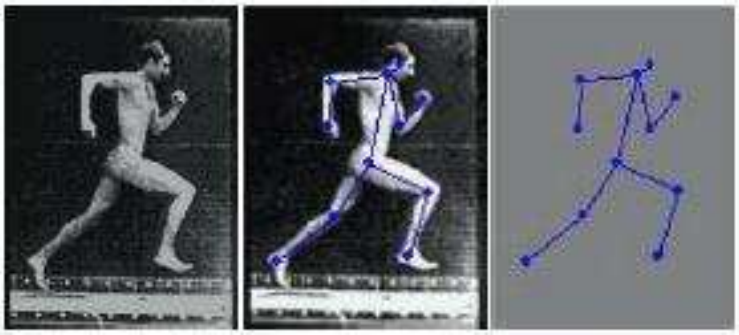


Figure 3.4: Human body model

The stick figure is described with the end points of the limbs.

$$p_s = (x_1, y_1, ..., x_{11}, y_{11})^T \tag{3.1}$$

However cartesian coordinate representation of the joints leads to problems at non-linear variations of the joints. Hence absolute angles of the limbs are computed and the stick figure is represented in polar coordinates. (Figure 3.5)(Equation 3.2, 3.3 and 3.4).

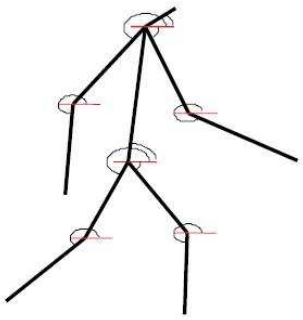


Figure 3.5: Human body model represented in terms of ten angle

$$\theta = tan^{-1}(\frac{y_i - y_j}{x_i - x_j}) \tag{3.2}$$

$$\Theta = (\theta_1, \theta_2, ..., \theta_{10})^T \tag{3.3}$$

$$x_s = (u, \Theta)^T \tag{3.4}$$

where u is hip center coordinate.

After the representation of each posture in an action is obtained, the *aSpace* is built. An action is described by a sequence of postures of different people performing same action.

$$A = x_1, ..., x_n \tag{3.5}$$

The mean posture is subtracted from each sample

$$\bar{A} = x_1 - \bar{x}, x_2 - \bar{x}, ..., x_n - \bar{x} \tag{3.6}$$

Covariance matrix of $\bar{A}$ is computed

$$\sum = \bar{A}\bar{A}^T \tag{3.7}$$

Eigenvectors (e_i) and eigenvalues (λ_i) of $\sum$ are computed. Each eigenvector corresponds to a mode of variation of the posture during the action. The smallest number of eigenvectors which best describe the large portion of the total variance of posture sequence are selected. Each posture is represented as a combination of the mean body posture and the eigenvectors. The compact representation of an action which consist of the eigenvectors (E) and the mean posture of the action ($\bar{x}$) is called as *aSpace* (Equation 3.8).

$$A = (E, \bar{x}) \tag{3.8}$$

Test sample is projected onto the action space for action recognition. The distance between the projected sample and *aSpace* is computed. Distance values are metrics for measuring the similarity of actions.

In *ASpaces*, the human posture is represented in terms of the orientations of the limbs and the coordinate of the hip. Since polar coordinates are used, the representation is invariant to translation, rotation and scaling. A nice approach is proposed based on simple stick figure representation of human posture and benefits of Point Distribution Model [35].

3.2 Non-Model Based Approaches

The previous part described the human motion analysis approaches based on using a prior human body model. There are approaches analyzing human motion without using a prior human model. Similar to model based approaches, the non-model based approaches start with representing human body using geometric structures. Stick figures, 2-D contours or 3-D volumetric structures are the most common representations.

Stick figures are the most simple representations of human body. As mentioned in the previous part, the stick figure representation is a collection of line segments and joint points. The motion trajectories of the joint points are used in analysis of motion. The idea of using motion of the joint points is first described in Moving Light Display (MLD) work of Johansson [15]. Motion is described with the positions of light displays placed on the joints of human body.

Human body can be represented as 2-D contours. Shio and Sklansky [37] use ribbons for representing the parts of human body. Ribbons are matched with each other in the sequence and ribbon sequences are built to obtain a shape description of articulated objects. Kurakake and Nevatia [9] use ribbon structures to segment people in motion.

In the following part, *Real Time Human Motion Analysis by Image Skeletonisation* of Fujiyoshi et. al. [10] is reviewed as an example for non-model based human motion analysis approach. They use a skeletal structure to represent human posture and analyze the features of the skeletal structure for motion analysis.

3.2.1 Real Time Human Motion Analysis by Image Skeletonisation

Fujiyoshi et. al. [10] have done a work on real time motion analysis of targets; particularly humans; by using star skeletonization process which does not require a priori human model. They use cyclic motion of legs and the posture of torso to classify and recognize human movements. An adaptive background modeling is used for real time target extraction. The result image of target extraction is preprocessed to clean up the anomalies via morphological operations (Figure 3.6).

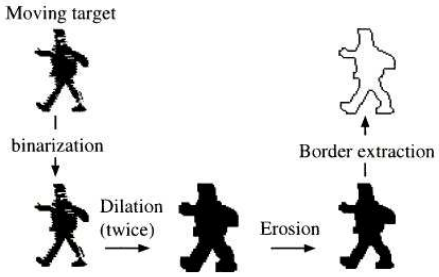


Figure 3.6: Target preprocessing

The outline of the target is extracted by using a border following algorithm. The skeletal representation of the target is extracted using local extremal points on boundary. This skeletal structure is called as *star* skeleton (Figure 3.7).

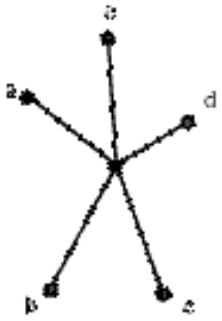


Figure 3.7: Star skeleton

The extremal points accommodate cues for motion analysis. The lower extremal points correspond to legs and cyclic motion of these points can be used in analysis of human motion such as walking or running. The angle (θ) that lower extremal point make with vertical is observed for cyclic motion detection (Figure 3.8).

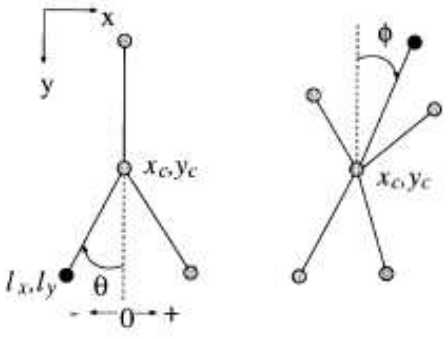


Figure 3.8: Features of star skeleton

The evaluation of angle (θ) demonstrates the cyclic nature of walking (Equation 3.9).

$$\theta = tan^{-1}(\frac{l_x - x_c}{l_y - y_c}) \tag{3.9}$$

The star skeleton representation contains another cue for determining the posture of a moving human. The inclination of torso can be approximated by the angle (ϕ) of the uppermost extremal point at the target (Figure 3.8).

In this work, the posture information is used for human motion classification. If the human is running, body tends to lean forward and the frequency of cyclic motion is higher. This metrics are used in motion classification and recognition.

3.3 Application Areas

The approaches in the human motion analysis are summarized in previous sections. The reviewed approaches have many usage in many applications in several application areas. In the following part, some of the popular application areas and sample applications are reviewed briefly.

3.3.1 Visual Surveillance

Visual surveillance applications deal with tracking the objects performing some special actions or detecting the objects performing some unusual actions. Visual surveillance applications are generally used in security sensitive areas such as battlefield management systems, borders, banks and parking lots. Video Surveillance and Monitoring (VSAM) [18], funded by Defense Advanced Research Projects Agency(DARPA), is an automatic video understanding technology that enables a single user to monitor activities in an area and alerts the user in case of unusualness. Systems for detecting burglaries or suspicious actions in parking lots are being developed. These systems involves tracking, face and gait recognition subsystems. These are called as smart surveillance systems. The smart surveillance systems detect the presence of a human in the area. The identity of the human is detected by using face recognition systems. Then, the activities of the human are analyzed and what the person is doing is determined. The visual surveillance applications are not only applied in security systems.

Grossmarkets use the systems for compiling consumer demographics. Traffic flow can be measured by using the visual surveillance systems.

3.3.2 Virtual Reality

Human motion analysis is used in creating virtual worlds. With the development of interactive spaces on the internet just like chat rooms, the systems interpreting gesture, head pose and facial expressions are becoming popular. The users will be able communicate without using text based methods.

3.3.3 Advanced User Interface

Advanced user interface applications are complementary part of speech recognition and natural language understanding systems used in human-machine interface systems. Gestures, body poses and facial expressions of a human can be used in communication. The system detects the presence of a human, then identifies who the user is and initiates a communication according to the identity using personal information and facial expressions. Advanced user interface applications can be used in sign-language translation, gesture driven control systems.

3.3.4 Motion Analysis

There are various applications involving motion analysis. One of the possible applications is context based storage and querying of video databases. Motion analysis is used in personal training systems for sports. The personalized training systems observe the skills performed by the trainee and make suggestions for improvements. Motion analysis systems are also used in choreography. High level descriptions of human movements for the notation of dance are constructed. Computer graphics area uses motion analysis to device realistic models for applications.

CHAPTER 4

HUMAN MOTION ANALYSIS VIA AXIS BASED REPRESENTATION

In the previous chapter, the approaches in human motion analysis are reviewed and application areas are discussed briefly. In this chapter, we introduce a non-model based human motion analysis scheme which is based on an axis-based representation of human body and change of axial features through the frames.

In this work, we analyze articulated motion to generate high level, knowledge based descriptions of motion. Following the terminology in knowledge based classification of motion (Section 2.3), we define *action* as temporal series of the change in the human posture. We neither assume any predefined human model, nor use any special motion capture system and represent motion without using motion fields. We only use silhouettes and articulation coordinates to represent action.

In our human motion analysis scheme, performed action is considered as a sequence of human postures. Each posture is represented with an axis based representation called disconnected skeleton [1] [38] which is essentially a collection of isolated simple local symmetry branches organized around a unique shape center (Figure 4.1). The features of local symmetry branches (the angle between symmetry branches and the length of symmetry branches) are processed to extract the motion information. Obtained information is used in analysis of human motion. The change in the angle values between the symmetry branches constitute a pattern for an action. The patterns are utilized to generate representations for actions. The generated action representations are used in similarity analysis of the actions.

In disconnected skeleton representation, symmetry branches are classified as positive or negative according to Tari, Shah, Pien method [39]. Positive symmetry branches

correspond to articulated body parts. In disconnected skeleton representation, a positive branch always merges with a negative branch. When they meet, both branches terminate (Figure 4.1). The start point and disconnection point of a negative symmetry branch stays stable under articulations. On the other hand, positive symmetry branches move freely as a result of articulated motion.

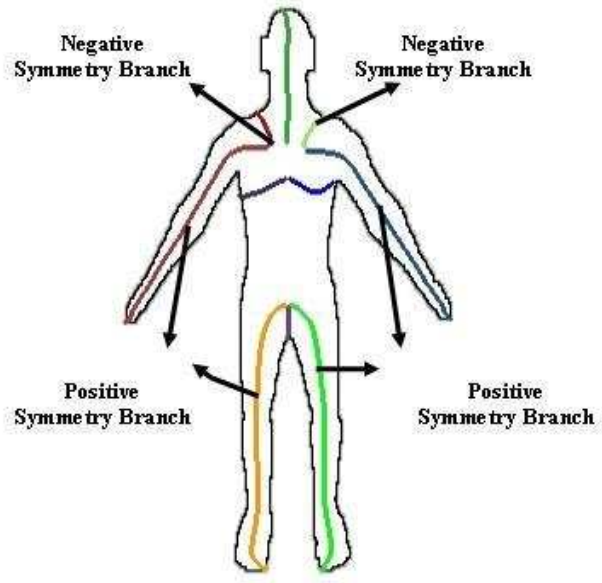


Figure 4.1: Disconnected skeleton representation of human body

In the next section, we describe the representation of a single frame in a motion sequence.

4.1 Representation of A Single Frame

In the proposed human motion analysis scheme, at each frame, disconnected skeleton representation of the posture is obtained. Recall that the angle between symmetry branches and the length of symmetry branches contain the information about the action. The features of symmetry branches of freely moving articulated parts (positive branches) are examined to extract the information. As mentioned in previous section, a positive branch merges with a negative branch and they both terminate when they meet. Furthermore, a positive (negative) branch is always neighbored by a negative (positive) branch. The angle between a positive symmetry branch and the nearest

negative branch is used in motion analysis (Figure 4.2). Details of the computation of feature values are described in the following part.

Vector, v_1, is drawn from start point of negative symmetry branch (xs,ys) to end point of negative symmetry branch (x2,y2). Similarly vector, v_2, is drawn from start point of negative symmetry branch (xs,ys) to end point of positive symmetry branch (x1,y1).

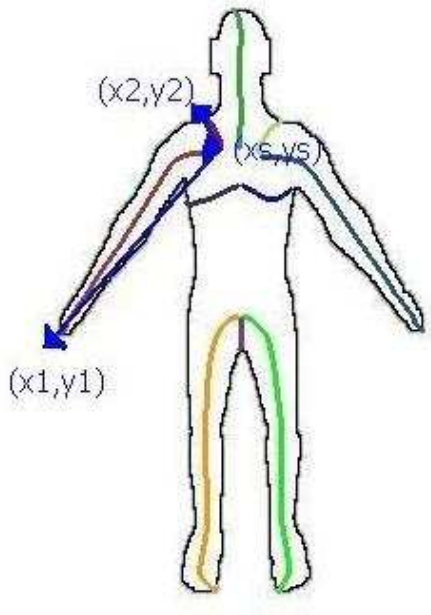


Figure 4.2: Features of symmetry branches

The angle between the symmetry branches is computed as in Equation 4.1.

$$\phi = \arccos(v1.v2) \tag{4.1}$$

where ϕ takes values between 0 and π in radians.

The length of the symmetry branch is computed as in Equation 4.2.

$$\ell = |v1| = \sqrt{((x1 - xs) * (x1 - xs)) + ((y1 - ys) * (y1 - ys))} \tag{4.2}$$

Lengths are normalized by dividing length of each symmetry branch to total length of symmetry branches. Hence the sum of the lengths of the symmetry branches is always equal to 1.

There are five positive symmetry branches in the axial representation of human body corresponding to the articulated body parts (one head, two arms, two legs) (Figure 4.3).

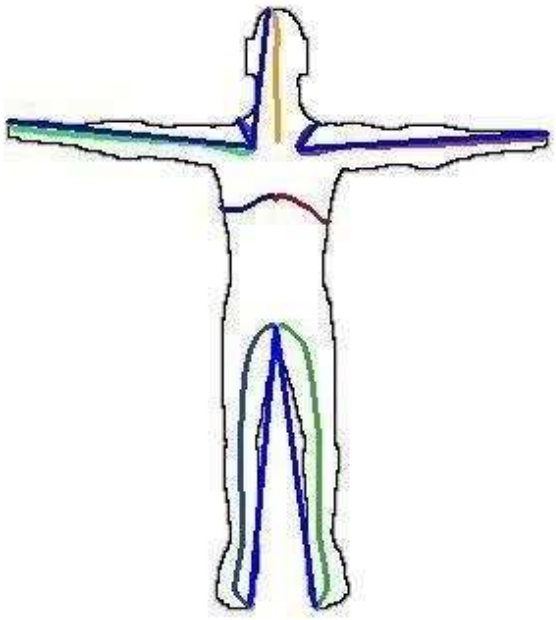


Figure 4.3: Vector representations of symmetry branches

A single frame is represented with vectors storing the values of the computed features (Equation 4.3 and 4.4).

$$\phi = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 & \phi_4 & \phi_5 \end{bmatrix} \tag{4.3}$$

$$\ell = \begin{bmatrix} \ell_1 & \ell_2 & \ell_3 & \ell_4 & \ell_5 \end{bmatrix} \tag{4.4}$$

Feature values are stored in the order of left arm, head, right arm, right leg and left leg in the vectors ϕ and ℓ.

Representation of a single frame in the proposed human motion analysis scheme is described in this section. Representation of a frame sequence is described in the following section.

4.2 Representation of Sequence of Frames

After obtaining the representation of each frame in the sequence, the values are collected to construct a trajectory showing the change of the angle during the course of an action. Vector representation of each frame are brought together and the sequence is represented as matrices θ and L (Equation 4.5). Each row of the matrices stores the representation of the human posture at a specific frame and each column stores the values of a specific feature corresponding to one of the five articulation points through the course of the performed action.

$$\theta = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} & \phi_{15} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ \phi_{N1} & \phi_{N2} & \phi_{N3} & \phi_{N4} & \phi_{N5} \end{bmatrix} \quad L = \begin{bmatrix} \ell_{11} & \ell_{12} & \ell_{13} & \ell_{14} & \ell_{15} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ \ell_{N1} & \ell_{N2} & \ell_{N3} & \ell_{N4} & \ell_{N5} \end{bmatrix} \tag{4.5}$$

where N is the number of frames in sequence.

As an example, some specific frames and trajectories of change of the feature values for the case of left arm weaving are displayed in Figure 4.4 and 4.5.

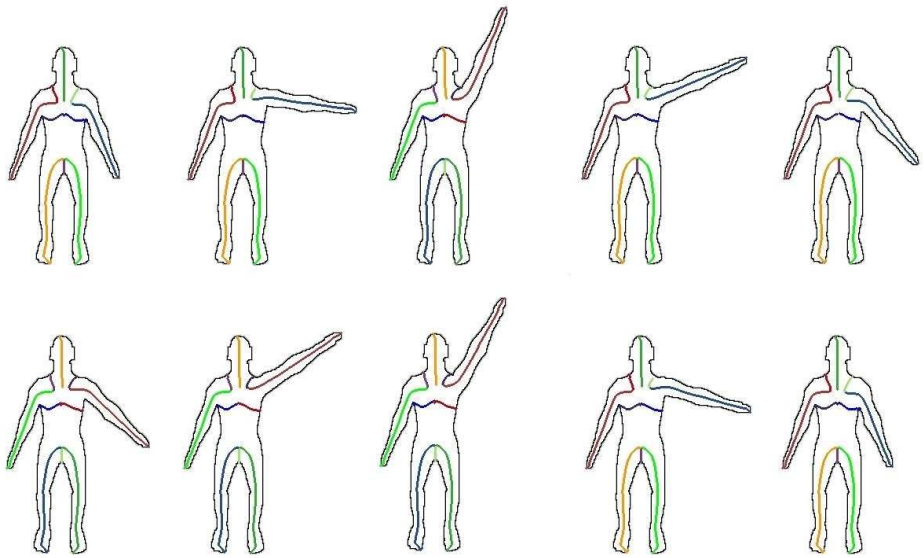


Figure 4.4: Specific frames of left arm weaving action

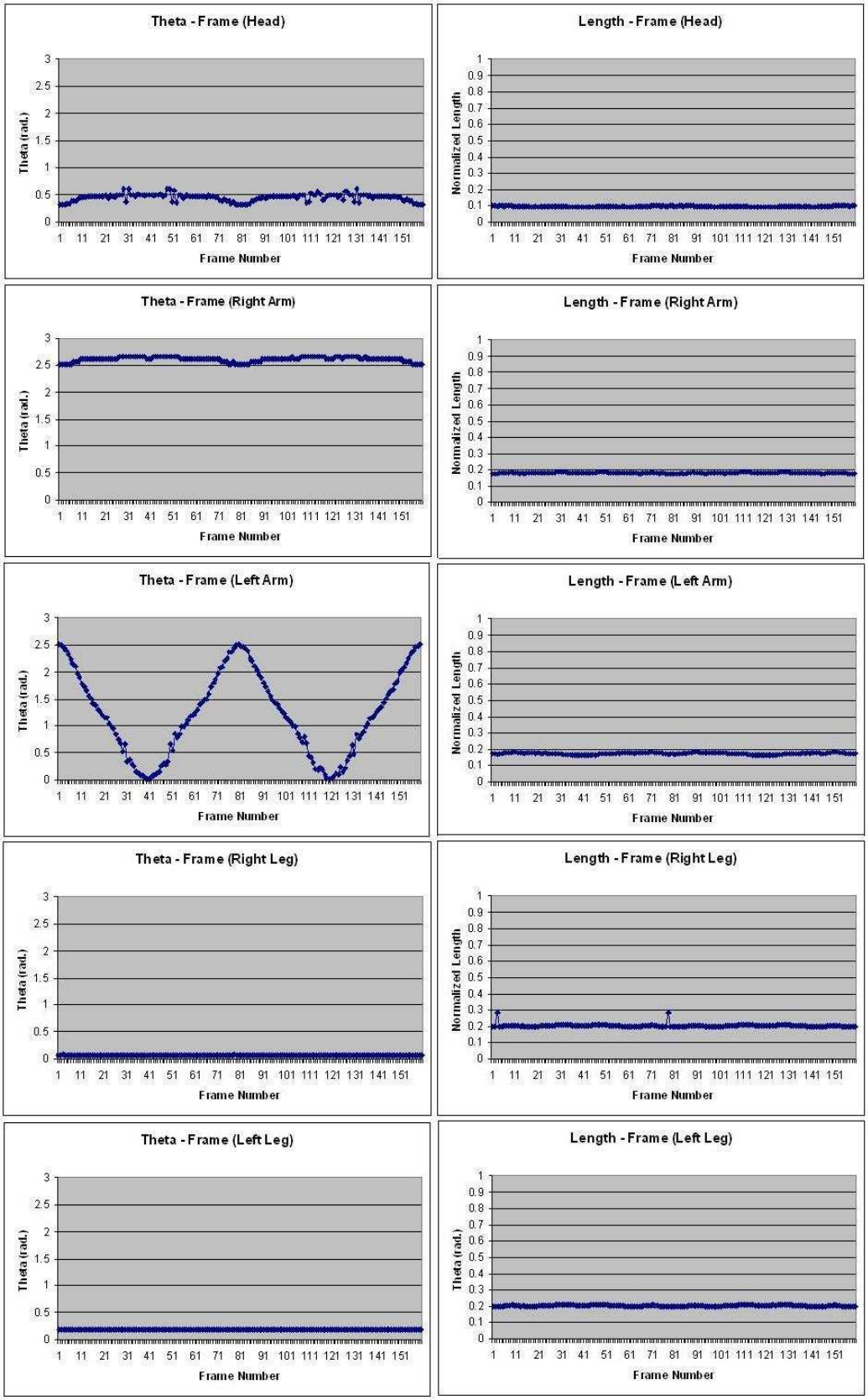


Figure 4.5: Feature trajectory graphics

In Figure 4.5 , the trajectories of the change in the angle between the symmetry branches and length of the symmetry branches corresponding to articulated body parts (positive symmetry branches) are displayed. As observed from the graphics, there is nearly no change in the feature values for the articulated parts which do not move. However, for the left arm, there is a periodic change in the values displaying the motion. There is also nearly no change in the length of the symmetry branches since there is no self-occlusion and motion in the plane orthogonal to viewing plane. The effects of self-occlusion and motion in the plane orthogonal to viewing plane are discussed in Section 4.4.

The important thing in the proposed human motion analysis scheme is the trajectories of the change in the feature values. Hence the obtained matrix representation is preprocessed before the analysis operations. The median of each column is subtracted from the the values in the columns. By this way, the variations of the change in the feature values are used in the analysis of motion. The ”median” is used in analysis to discard the effects of noise which causes extreme values in the trajectory.

After obtaining the representation of sequences of frames, the performed action can be analyzed using this representation. In the following sections, the approaches for the analysis of human motion based on the described representation are discussed.

4.3 Similarity Analysis of Actions

We propose a similarity analysis method for the actions. First the system is trained for specific actions. Then the system can be queried with a test action to measure the similarity of the test action with the trained actions. In this section, the details of the similarity analysis are described.

A specific action is performed by different humans having different postural characteristics and matrix representation of each performance is obtained. This part is called as training phase. In this work, it is assumed that training sequences have same number of frames to generate an accurate representation for motion.

Consequently, assuming T different training sequence, there are T matrix representation for a specific action. The action space for the motion is computed by using Principal Component Analysis [35]. Construction of action space has four steps:

1. Convert matrix representations of motion sequences into row vectors. (Equation 4.6).

$$Q = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} & \phi_{15} & \phi_{21} & \phi_{22} & ..... & \phi_{N4} & \phi_{N5} \end{bmatrix} \quad (4.6)$$

2. Mean posture for the action, Q_M, is obtained and it is subtracted from each training sequence. The matrix in Equation 4.7 is obtained.

$$D = \begin{bmatrix} Q_1 - Q_M \\ Q_2 - Q_M \\ . \\ . \\ Q_T - Q_M \end{bmatrix} \quad (4.7)$$

   where T is the number of the training sample videos.

3. Covariance matrix of D is computed (Equation 4.8).

$$C = DD^T \quad (4.8)$$

4. Eigenvalues and eigenvectors of C are calculated. Each eigenvector (e_i) has a correspondence with the variation of posture during the action performance. Corresponding eigenvalue (λ_i) is the measure of the variation. The smallest number of eigenvalues covering an essential part of the total variation are chosen. The corresponding eigenvectors and the mean posture construct a compact representation for the action.

Consequently, the compact representation of the action is obtained. The data collected in training phase and the compact action representation are used in similarity analysis of actions. We call the analysis part of the scheme as analysis phase.

Similarity analysis of a test action has four steps:

1. Axis-based descriptions of the query sequence are formed.

2. The query sequence is aligned with the trained action sequence. In this work, Dynamic Time Warping(DTW) [35] is used to compensate the difference in the

speed of the actions. Details of Dynamic Time Warping (DTW) are explained in Appendix.

3. The aligned trajectories are projected onto the action space.

4. The difference is computed. Different methods can be used for finding the distance between the sequence. In this work, we used Euclidean Distance for simplicity.

In this section, an approach for the similarity analysis of actions based on the proposed human motion representation is described. There are also other inferences about the human motion based on the examination of feature trajectories in the proposed human motion representation. In the following section, we describe some of these inferences.

4.4 Analysis of Feature Trajectories

In our motion analysis scheme, 3D effects of human body are neglected in the training phase. The information is retrieved from 2D silhouettes. However, in analysis phase, the effects of 3D structure such as self occlusion and motion in the plane orthogonal to viewing plane are revealed.

4.4.1 Self-Occlusions

Self-occlusions occur when an articulated body part is partially hidden behind another body part. In case of self occlusion, the coordinates of the start point of the corresponding symmetry branches change significantly and the angle between symmetry branches gets significantly different values from the usual. As motion continues and articulated part becomes visible (it goes far away from occluding part), the representation returns to normal and the trajectory starts to follow a usual pattern. Self-occlusions result in discontinuities and sharp changes in both angle trajectories of symmetry branches and relative length of articulated body parts.

Detecting self-occlusions reveals information about the position of the camera. In case of a dynamic camera system (the camera can be translated and rotated), the position of the camera can be changed to eliminate the self-occlusions.

In Figure 4.6, some specific frames of a video containing an action having self-occlusion are displayed. The moving arm (left arm) of the human is occluded by the torso. As the motion continues, it goes away from the torso and the occlusion disappears.

In Figure 4.7, the graphics of the trajectories of the angle and the length for articulated part are displayed.

The discontinuities in the trajectories alert self-occlusion and indicates that the position of the camera is inappropriate to catch the performed action accurately. The camera should be repositioned.

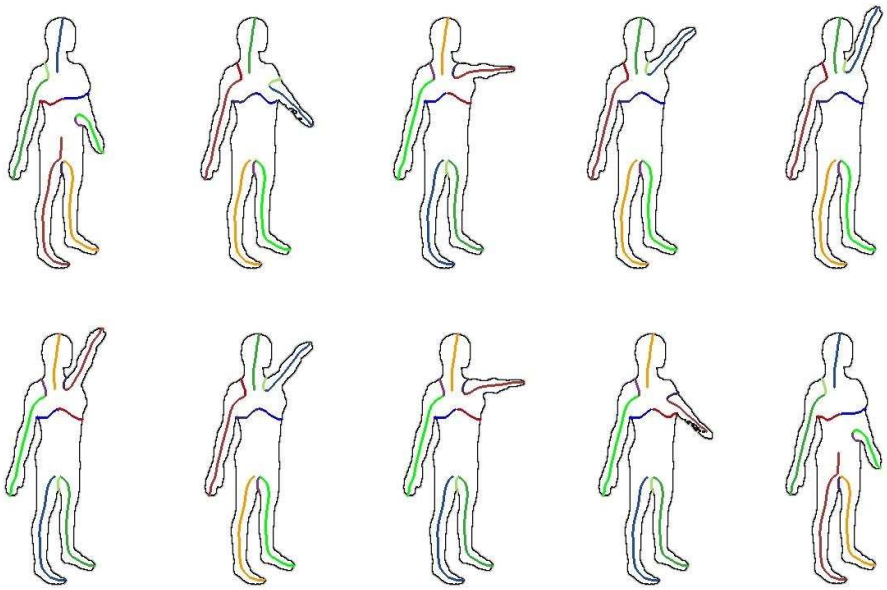
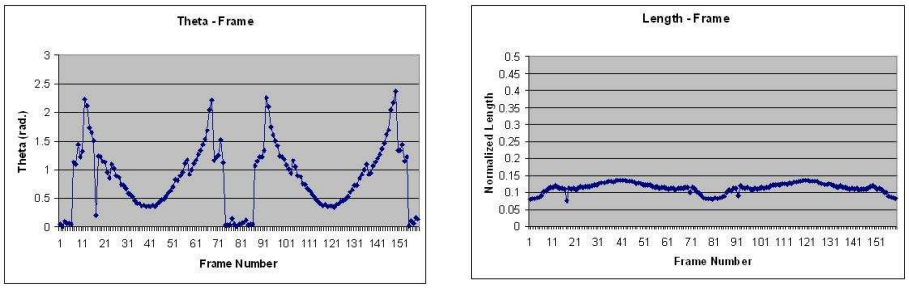


Figure 4.6: Frames of human motion having self-occlusion



(a) (b)

Figure 4.7: (a)Frame - theta (ϕ) graph of articulated body part (left arm) (b)Frame - length graph of articulated body part (left arm)

The camera is translated towards the articulated body part and rotated in the opposite direction of translation. In Figure 4.8 , frames of the new situation are displayed. In Figure 4.9, the graphics of the trajectories of the angle and the length for articulated part are displayed. There is again discontinuities in the graphics. But the number of frames that have discontinuity has decreased. This indicates that, the camera should be again repositioned to eliminate the self-occlusion.

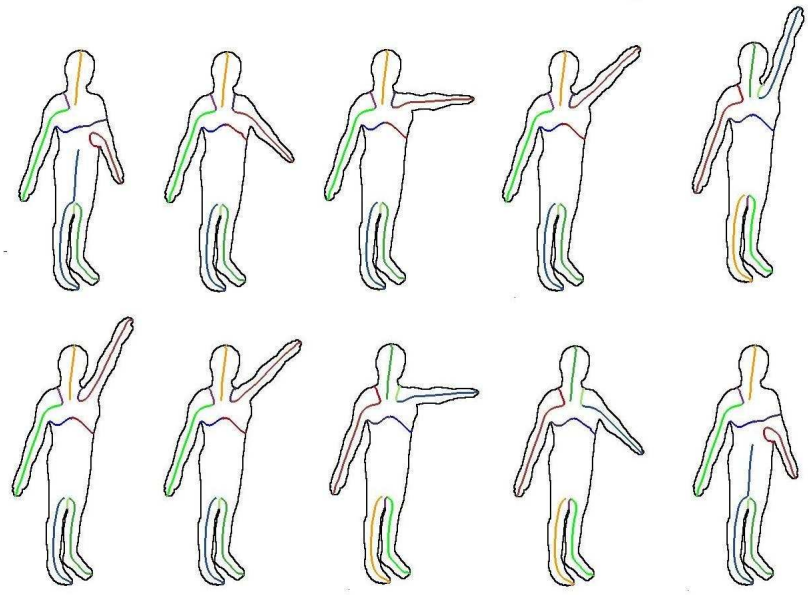
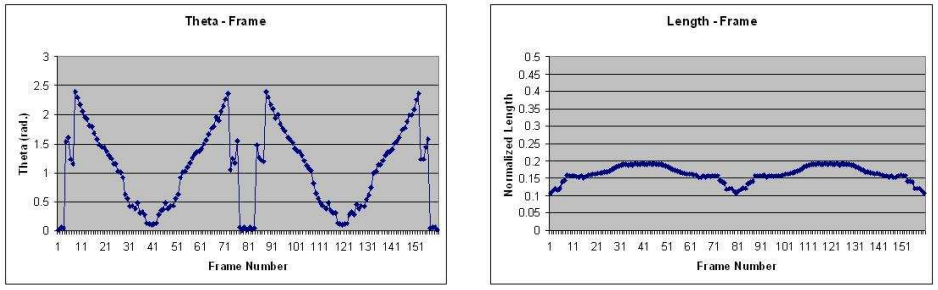


Figure 4.8: Frames of human motion having self-occlusion



(a) (b)

Figure 4.9: (a)Frame - theta (ϕ) graph of articulated body part (left arm) (b)Frame - length graph of articulated body part (left arm)

The camera is repositioned again. In Figure 4.10 , frames of the final situation are displayed. In Figure 4.11, the graphics of the trajectories of the angle and the length for articulated part are displayed. As observed from the graphics, the discontinuities has disappeared. Self-occlusion is eliminated by translating and rotating the camera.

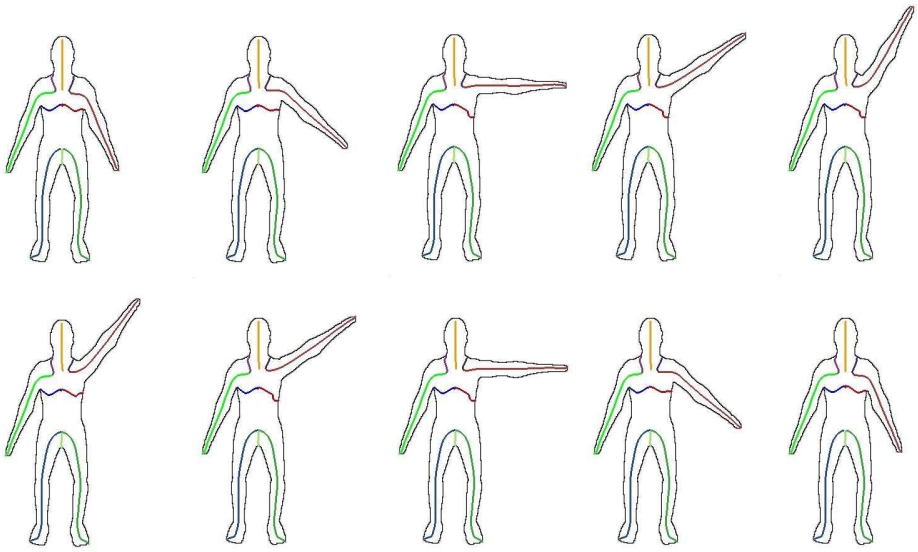
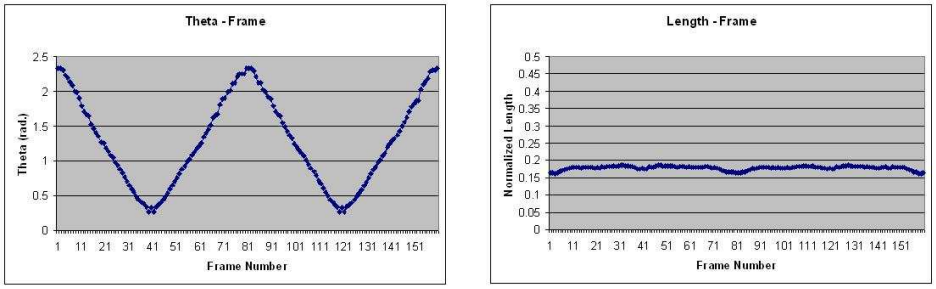


Figure 4.10: Frames of human motion (Self-occlusion is eliminated)



(a) (b)

Figure 4.11: (a)Frame - theta (ϕ) graph of articulated body part (left arm) (b)Frame - length graph of articulated body part (left arm)

4.4.2 Motion In the Plane Orthogonal to the Viewing Plane

The proposed human motion analysis scheme also carries information about the motion orthogonal to the viewing plane. The information is hidden in the change of the relative length of the articulated body parts during a video sequence. The length of a symmetry branch is simply computed as the distance between its start point and end point. As mentioned earlier, the length of articulated body part stays nearly constant when the motion is parallel to viewing plane. If the length of articulated body part changes as the angle stays stable, it implies a movement orthogonal to viewing plane. Since the articulated body part comes nearer or goes further from camera, the length changes.

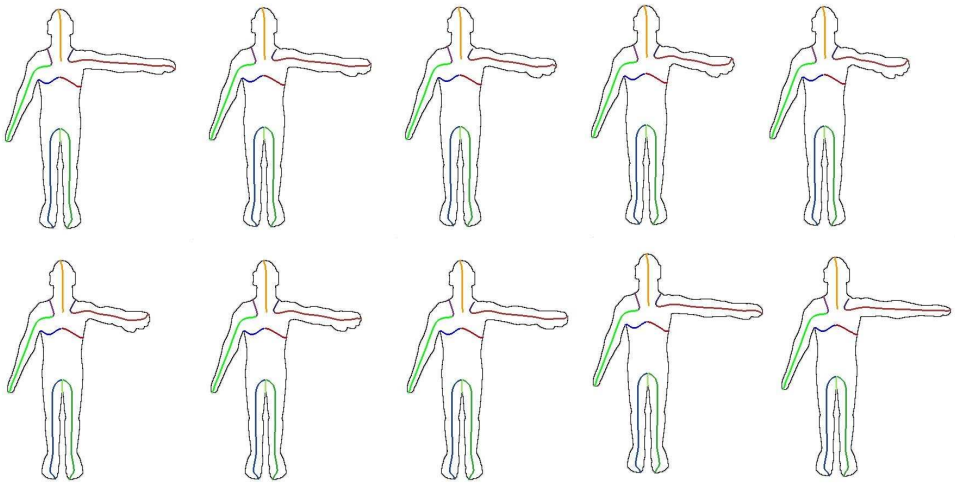
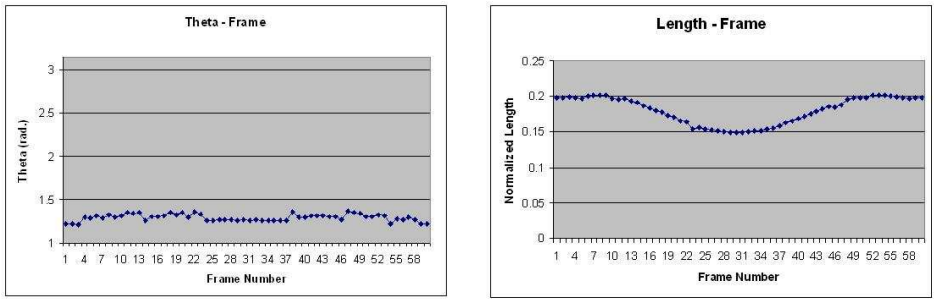


Figure 4.12: Frames of human motion orthogonal to viewing plane



(a) (b)

Figure 4.13: (a)Frame - theta (ϕ) graph of articulated body part (left arm) (b)Frame - length graph of articulated body part (left arm)

In Figure 4.12, we display some specific frames of a video of a human weaving his left arm towards the camera. Hence the movement is on the plane orthogonal to the viewing plane. In Figure 4.13, the trajectories of angle and length are displayed. The length changes through the motion while the angle stays nearly constant.

4.5 Connection to Related Works

As mentioned in Chapter 3, there are various approaches in the literature related with visual analysis of human motion. The proposed approach in this thesis can be considered as an example for non-model based approaches. In Chapter 3, two of the works in the literature are reviewed in details; *ASpace* [5] and *Real Time Human Motion Analysis by Image Skeletonisation* [10]. They are example works for model based and non-model based approaches accordingly. These works are important to show where our work stands in the literature.

In these works, human body is represented with skeletal structures and the features of the skeletal representations are used in human motion analysis. In our work, we also represent human body with an axis based skeletal structure and the features extracted from the skeletal structure are used in human motion analysis.

Our approach differs from *ASpace* work in baseline of the analysis. *ASpace* is a model based approach. A human body model is fit on the human posture at each frame and the features are computed using the model.

Real Time Human Motion Analysis by Image Skeletonisation work bases the analysis on the cyclic nature of the movement in the extreme point of the star skeleton representation. Articulated body parts are not considered while analyzing the motion. In our approach, we also focus on the change of the features through the action performance and analyze the nature of the change to extract information about the performed action. The articulated parts of the human body have great importance in our analysis. The analysis is based on the movement of articulated body parts.

Also our approach differs in the skeletal structure used in the representation of human body in the analysis. In *ASpace*, human body is represented with a stick figure composed of rigid body parts connected each other at joints. In *Real Time Human Motion Analysis by Image Skeletonisation*, a special structure called as "star skeleton" is used in analysis. We use disconnected skeletons [1] [38] to represent human body

which is more flexible structure compared with well known skeletal structures.

The proposed approach in this thesis also differs from other silhouette based approaches in the sense that initially ignored 3D effects of human body such as self-occlusion and motion in the plane orthogonal to viewing plane are revealed in the analysis.

4.6 Experimental Results

In this section, we will demonstrate the results of the experiments. In the experiments, nine different human beings having different postural characteristics are used for training our system. The training video sequences are all 160 frames length.

4.6.1 Similarity Analysis

The system is trained for five different actions in similarity analysis experiments. Then the similarity of the training actions with each other is computed and confusion matrix of the performed actions are constructed (Table 4.1). We present disconnected skeleton representation of some specific frames of action sequence in Figure 4.14 - 4.18.

The performed five actions can be described briefly as follows:

- Action #1 - Left Arm Weaving: The human weaves his left arm from down to up. There is no bending in the action. Some specific frames of the performed action are displayed in Figure 4.14.

- Action #2 - Right Arm Weaving: The human weaves his right arm from down to up. Some specific frames of the performed action are displayed in Figure 4.15.

- Action #3 - Both Arm Weaving: The human weaves his both arms from down to up. There is no bending in the action. Some specific frames of the performed action are displayed in Figure 4.16.

- Action #4 - Both Arm Weaving and Bending: The human weaves his left arm from down to up. After a point, the human bends the elbows. Some specific frames of the performed action are displayed in Figure 4.17.

- Action #5 - Left Leg Weaving: The human weaves his left leg from down to up. Some specific frames of the performed action are displayed in Figure 4.18.

Table 4.1: Confusion Matrix of Performed Actions

	Action #1	Action #2	Action #3	Action #4	Action #5	Figures
Action #1	1.34	12.76	7.91	11.99	9.93	Figure 4.14
Action #2	13.06	1.81	9.17	10.73	9.80	Figure 4.15
Action #3	9.08	9.24	2.67	9.65	13.22	Figure 4.16
Action #4	11.45	12.15	9.64	1.86	10.74	Figure 4.17
Action #5	9.37	9.50	11.89	10.71	1.30	Figure 4.18

The distance is a metric of similarity of the actions. As observed from the confusion matrix, the distances between similar actions are smaller.

Another action different from the five training actions is compared with the training sequences to decide to which action the performed action is more similar. In this action, human weaves both his right arm and left leg simultaneously. Some specific frames of the action performance are displayed in Figure 4.19. The results of the distance computation are displayed in Table 4.2. As observed from the results in the table, the distances between the performed action and Action #2 (Right Arm Weaving Action) and Action #5 (Left Leg Weaving Action) are smaller when compared with the other distances indicating that the performed action is more similar to these two actions compared with the other training actions.

Table 4.2: Test Action Similarity Analysis Results

	Action #1	Action #2	Action #3	Action #4	Action #5
Test Action	12.98	5.34	10.04	12.56	8.43

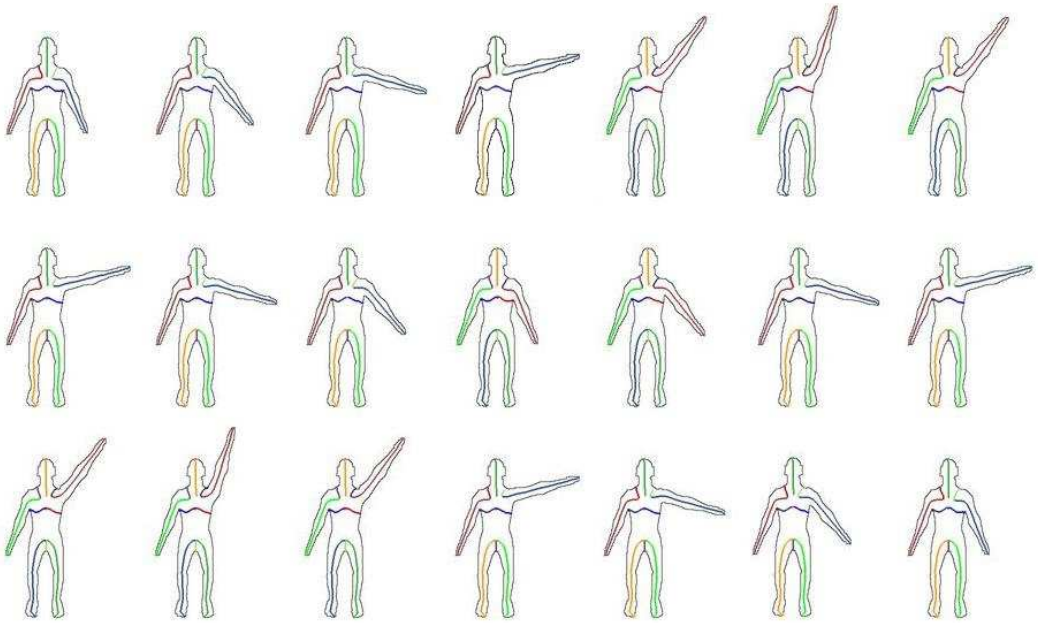


Figure 4.14: Frames of left arm weaving action performance

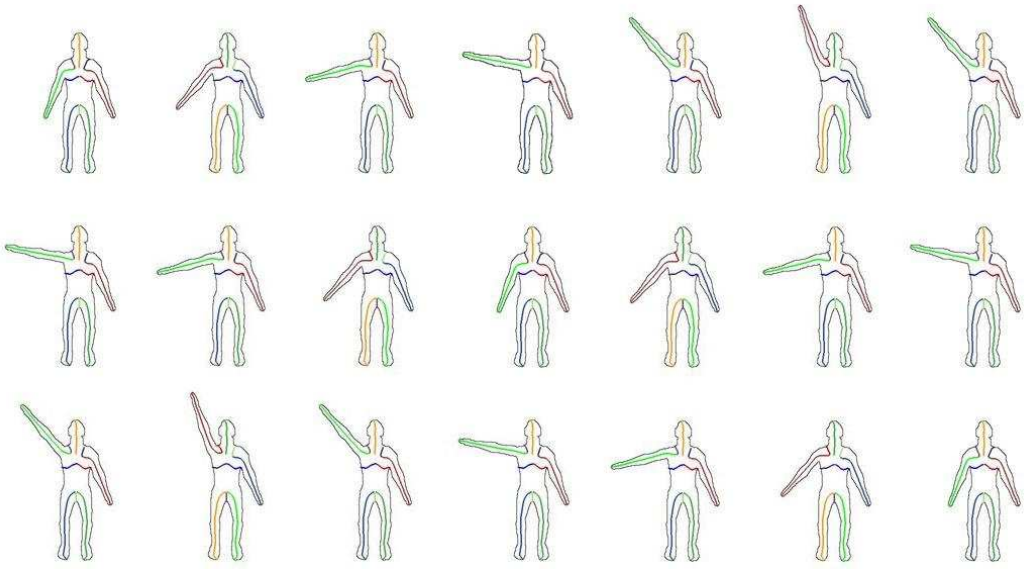


Figure 4.15: Frames of right arm weaving action performance

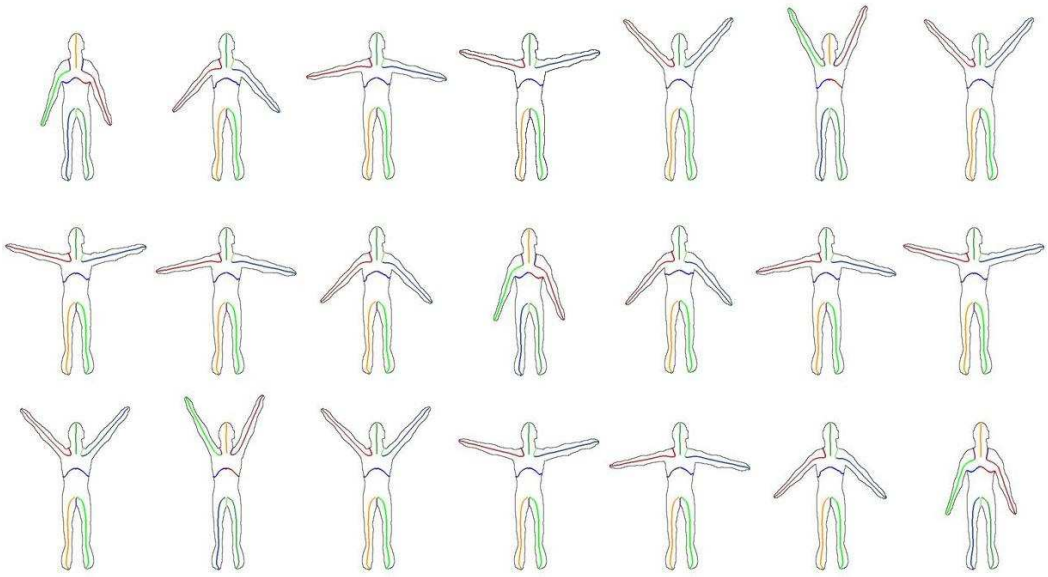


Figure 4.16: Frames of both arm weaving action performance

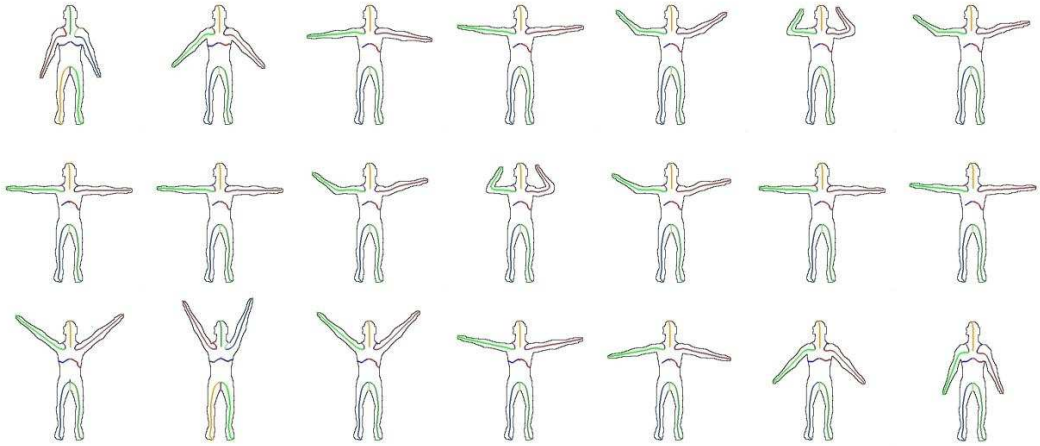


Figure 4.17: Frames of bended both arm weaving action performance

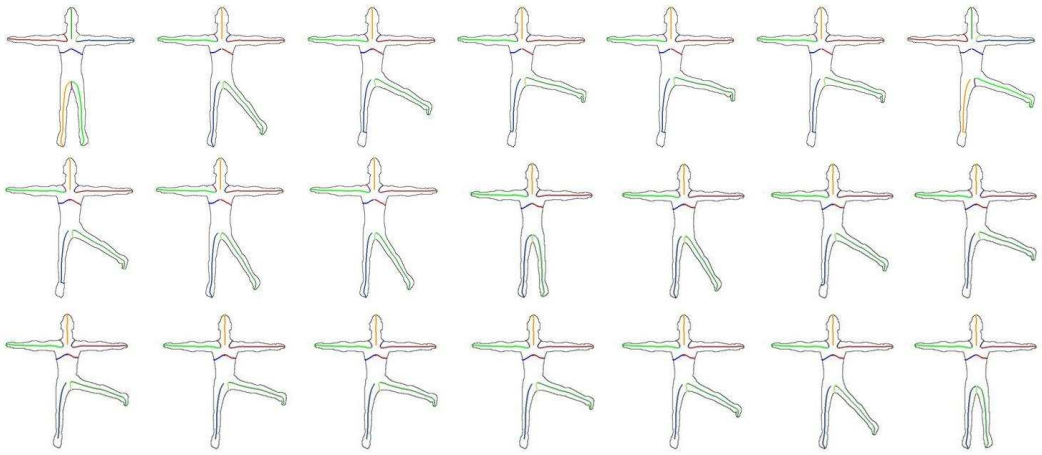


Figure 4.18: Frames of leg weaving action performance

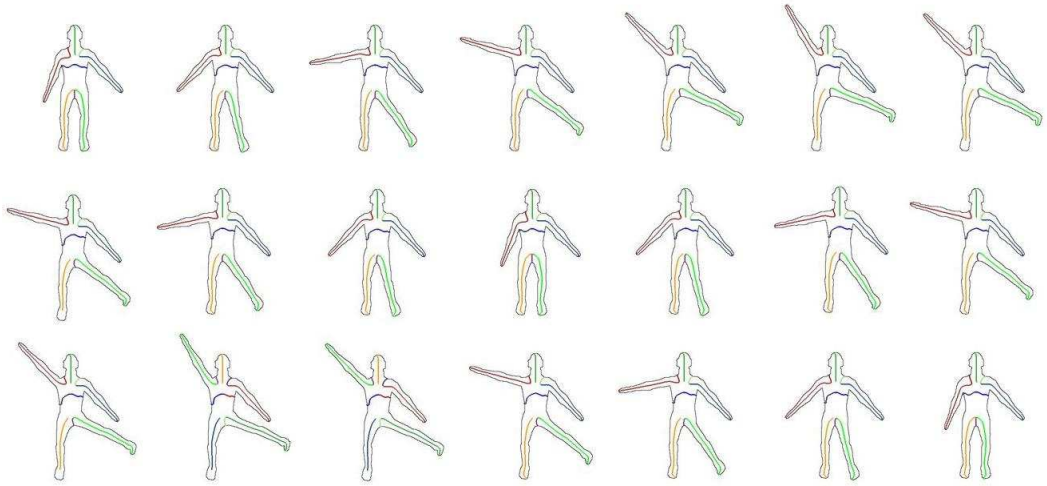


Figure 4.19: Frames of test action performance

4.6.2 Self Occlusion Detection

In this experiment, the human weaves his left arm. The camera is shifted a bit right and rotated about 50 degree to the left. Due to the position of the camera, the left arm is self occluded by the torso at some specific frames. In Figure 4.6, some specific frames of the action sequence are displayed. In Figure 4.7, the trajectories of the features are displayed. The disconnected parts in the Frame - Theta graphics (Figure 4.7.a) correspond to the frames having self occlusion (frames: 1-20, 65-95 and 145-160). As it can be observed in Frame-Length graph (Figure 4.7.b), the length of the corresponding symmetry branch at the corresponding frames changes abruptly. As

mentioned earlier, it should stay stable in ordinary cases.

Self-occlusion indicates that the viewing plane should be considered to grab the action accurately. The position and rotation of the camera should be changed. The camera is shifted a bit left and rotated about 20 degrees to the right. The frames of the performed action in this new viewing position are displayed in Figure 4.8 and the trajectories of the features corresponding to the left arm are displayed in Figure 4.9.

As observed from the trajectory graphics, self-occlusion has not disappeared but number of the frames having self-occlusion has decreased (frames: 1-10, 75-85 and 155-160). Hence the camera is again shifted to left and rotated 30 degrees to the right.

The frames in this new situation are displayed in Figure 4.10 and the trajectories of the features corresponding to the left arm are displayed in 4.11. This time, self-occlusion disappeared.

4.6.3 Motion Orthogonal to the Viewing Plane

In this experiment, the human weaves his left arm towards the camera. Hence, the motion is on the plane orthogonal to viewing plane. In Figure 4.12, some specific frames of the action sequence are displayed. In Figure 4.13, the trajectories of the features are displayed. The angle value stays almost stable. There are some variations due to noise and the small changes in the start or end point location of the symmetry branches. However, there is a continuous and severe change in the length. The length of the symmetry branch corresponding to the moving arm get shorter as the arm comes nearer to camera and then returns back to its original length. The change in the length indicates the motion in the plane orthogonal to viewing plane.

CHAPTER 5

SUMMARY and DISCUSSIONS

In this thesis, a potential use of disconnected skeletons in motion analysis is demonstrated. A non-model based human motion analysis scheme is proposed. The particular structure of disconnected skeletal representation accommodates features revealing the motion information. The trajectory of features defined with respect to a part based coordinates are utilized to describe action. An action coding scheme based on the descriptions obtained from analysis of the trajectories is presented. Trajectories are aligned via Dynamic Time Warping (DTW) and compact action models are generated via Principal Component Analysis. Generated models are used in action recognition. The scheme works on 2D silhouettes and ignore 3D effects during skeleton extraction step. However, initially ignored 3D effects such as self-occlusions or motion in the plane orthogonal to the viewing plane can be detected during the analysis step.

It may be argued that if we miss a feature at some frames, the trajectory will have gaps and the gaps will lead to defects in the analysis. The problem can be solved by using prediction algorithms such as Kalman Filters, simulation based approaches (Chapter 2). The trajectory can be fitted into a functional model which will be used to estimate the missing values.

The intend of the thesis is to study the usage of disconnected skeleton representation in human motion analysis. We explored the features of the disconnected skeleton structure and analyzed the behaviors of the features. We obtained good results revealing many cues about the human motion. The methods and the results are discussed in the thesis. The future work will be to generate high level descriptions for the performed actions and introduce contextual data to produce a more accurate

descriptions for real life events. Another future work may be developing a template matching based action recognition method based on the similarity analysis discussed in Section 4.3.

We experimentally demonstrate the results of the proposed approach in action similarity analysis, self-occlusion detection and detecting motion in the plane orthogonal to viewing plane in Chapter 4. The proposed method takes the advantages of axis based representation human body to generate a motion analysis scheme.

REFERENCES

[1] C. Aslan and S. Tari. An axis based representation. *IEEE International Conference on Computer Vision*, 2005.

[2] W. Liao J. Aggarwal, Q. Cai and B. Sabata. Articulated and elastic non-rigid motion: A review. *Proc. of IEEE Computer Society Workshop on Motion of Non-Rigid and Articulated Objects*, pages 16–22, 1994.

[3] A. Bobick. Movement, activity and action: The role of knowledge in the perception of motion. *Royal Society Workshop on Knowledge-based Vision in Man and Machine*, 1997.

[4] K. Akita. Image sequence analysis of real world human motion. *Pattern Recognition*, 17(1):73–83, 1984.

[5] F. X. Roca J. Gonzàlez, J. Varona and J. J. Villanueva. aspaces: Action spaces for recognition and synthesis of human actions. *Articulated Motion and Deformable Objects (AMDO)*, 2492:189–200, 2002.

[6] F. X. Roca J. Gonzàlez, J. Varona and J. J. V illanueva. Analysis of human walking based on aspaces. In *Articulated Motion and Deformable Objects (AMDO)*, pages 177–188, 2004.

[7] S. Niyogi and E. Adelson. Analyzing and recognizing walking figures in xyt. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 469–474, 1994.

[8] S. Niyogi and E. Adelson. Analyzing gait with spatiotemporal surfaces. *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 64–69, 1994.

[9] S. Kurakake and R. Nevatia. Description and tracking of moving articulated objects. *International Conference on Pattern Recognition*, 1:491495, 1992.

[10] H. Fujiyoshi and A. J. Lipton. Real-time human motion analysis by image skeletonization. In *Proceedings of the 4th IEEE Workshop on Applications of Computer Vision (WACV'98)*, page 15, 1998.

[11] J. O'Rourke and N. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 522–536, 1980.

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

[13] D. Hogg. Model based vision: A program to see a walking person. *Image and Vision Computing*, pages 5–20, 1983.

[14] K. Rohr. Towards model-based recognition of human movements in image sequences. *Computer Vision, Graphics and Image Processing: Image Understanding*, pages 94–115, 1994.

[15] G. Johansson. Visual motion perception. *Science American*, 232(6):7588, 1975.

[16] Yee-Hong Yang and Martin D. Levine. The background primal sketch: An approach for tracking moving objects. *Machine Vision and Applications*, 5(1):17–34, 1992.

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

[18] A. Lipton R. Collins and T. Kanade. A system for video surveillance and monitoring. *Proceedings of the American Nuclear Society (ANS) Eighth International Topical Meeting on Robotics and Remote Systems*, 1999.

[19] C. Bregler. Learning and recognizing human dynamics in video sequences. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 568–574, 1997.

[20] B.K.P. Horn. *Robot Vision*. The M.I.T. Press, 1986.

[21] O. Munkelt C. Ridder and H. Kirchner. Adaptive background estimation and foreground detection using kalman-filtering. *Proceedings of International Conference on Recent Advances in Mechatronics*, 1:193–199, 1995.

[22] T. Darrell C. Wren, A. Azarbayejani and A. Pentland. Pfinder: Realtime tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[23] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, 2:246–252, 1999.

[24] D. Harwood A. Elgammal and L. Davis. Non-parametric model for background subtraction. *IEEE International Conference on Computer Vision*, pages 751–767, 2000.

[25] D. Harwood I. Haritaoglu and L. Davis. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[26] W. Mendenhall and T.L. Sincich. *Statistics for Engineering and the Sciences*. Prentice Hall, 4th edition, 1995.

[27] E. Brookner. *Tracking and Kalman Filtering Made Easy*. John Wiley, 1998.

[28] N. Gordon B. Ristic, S. Arulampalam. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House Radar Library, 2004.

[29] N. Gordon M. S. Arulampalam, S. Maskell and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[30] C. Andrieu A. Doucet and S. Godsill. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[31] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[32] H.H. Nagel. From image sequences towards conceptual descriptions. *Image Vision Computing*, 6(2):59–74, 1988.

[33] R. Polana and R. Nelson. Low level recognition of human motion or how to get your man without finding his body parts. In *Proceedings of IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 77–82, 1994.

[34] A.F. Bobick and J.W. Davis. The representation and recognition of action using temporal templates. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 928–934, 1997.

[35] V. Hlavac M. Sonka and R. Boyle. *Image Processing, Analysis, and Machine Vision*. PWS Publishing, 2nd edition, 1999.

[36] J. Ohya J. Yamato and K. Ishii. Recognizing human action in time sequential images using hidden markov model. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 379–385, 1997.

[37] A. Shio and J. Sklansky. Segmentation of people in motion. *IEEE Workshop on Visual Motion*, 2:325332, 1991.

[38] C. Aslan. Disconnected skeletons for shape recognition. Master's thesis, Middle East Technical University, 2005.

[39] Shah J. Tari Z.S.G. and Pien H. Extraction of shape skeletons from grayscale images. *Computer Vision and Image Understanding*, 66(2):133–146, 1997.

[40] E. Keogh and M. Pazzani. Derivative dynamic time warping. *First SIAM International Conference on Data Mining (SDM'2001)*, 2001.

APPENDIX A

METHODS

A.1 Principal Component Analysis

Principal Component Analysis (PCA) is one of the corner stone approach in data analysis. PCA is a technique to reduce a multi-dimensional data set to a lower dimension while retaining as much as possible of variations present in the data set. Relevant information from confusing data is simply extracted and complex data set is reduced to a lower dimension. It transforms a number of correlated variable into smaller number of uncorrelated variables. These variables are called as principal components. First principal component accounts for the greatest variance in the data, the second one accounts for the second greatest variance and so on. Data can be compressed using obtained components. Number of dimension is reduced without loosing much information. PCA has wide usage in many application areas as motion analysis, face recognition, pattern finding, image compression. The steps followed in PCA are as follows.

1. **Collect Data:** Multi dimensional data to be used in principal component analysis are collected.
2. **Subtract Mean:** Mean of collected data is computed and it is subtracted from each sample. This step produces a data set having zero-mean.
3. **Calculate Covariance Matrix:** Covariance matrix for mean-subtracted data is calculated to measure how much the dimensions vary from the mean with respect to each other.
4. **Calculate Eigenvalues and Eigenvectors of Covariance Matrix:** Eigenvalues and eigenvectors of covariance matrix is computed Eigenvectors identifies the

patterns in the data. The lines characterizing data is extracted by finding eigenvectors.

5. **Choosing Principal Components:** Eigenvalues covering large portion of the total variance in data is selected as principal components. The eigenvalues are ordered from highest to lowest. The highest eigenvalue is called as first principal component, second one is called as second principal component and so on. Components having small eigenvalue can be ignored. Of course, there is a loss of information by ignoring them however it is not significant. The final data set has less dimension than the original.

A.2 Dynamic Time Warping

Comparing two or more sequences, finding the distance between them and measuring the extent of difference has a vital importance in many application area. However comparing the raw data of sequences may lead to incorrect results. The sequences may have approximately same overall distribution but may not line up in a reference axis. In order to find the similarity between the sequences, they should be warped and aligned with each other. Dynamic Time Warping (DTW) is one of the methods used for warping sequences. DTW is an algorithm for measuring similarity between two sequences which may vary in time or speed. It has a wide usage area as gesture recognition, data mining, robotics, speech processing, etc. Details of the algorithm is as follows.

Suppose that there exists two sequences Q and R which have M and N elements respectively.

$$Q = q_1, q_2, q_3, ..., q_M \tag{A.1}$$

$$R = r_1, r_2, r_3, ..., r_N \tag{A.2}$$

DTW constructs an M-by-N distance matrix. The (i^{th}, j^{th}) elements of distance matrix contains the distance $d(q_i, r_j)$ between the points of sequences. The aim is to find a warping path W defining the mapping between the sequences.

$$W = w_1, w_2, w_3, ..., w_K \tag{A.3}$$

The warping path should satisfy some constraints.

Boundary Condition : Warping path starts and finish in diagonally opposite corner cells of distance matrix. So $w_1 = (1, 1)$ and $w_K = (M, N)$

Continuity : Allowable steps in warping path contains only adjacent cells. So for $w_k(a, b)$, $w_{k-1}(\hat{a}, \hat{b})$ is such that $a - \hat{a} \leq 1$ and $b - \hat{b} \leq 1$.

Monotonicity : Points in W is monotonically spaced in time. So for $w_k(a, b)$, $w_{k-1}(\hat{a}, \hat{b})$ is such that $a - \hat{a} \geq 0$ and $b - \hat{b} \geq 0$.

Under these constraints, the aim is to find the path which minimizes the warping cost.

$$DTW(Q, R) = min\frac{\sqrt{\sum_{k=1}^{K} w_k}}{K} \tag{A.4}$$

The solution to finding minimum distance is obtained easily by using dynamic programming. The algorithm is as follows.

```
int DTWDistance(char q[1..m], char r[1..n], int d[1..m,1..n]) {
    declare int DTW[0..n,0..m]
    declare int i, j, cost
    for i := 1 to n
        DTW[0,i] := infinity
    for i := 1 to m
        DTW[i,0] := infinity
    DTW[0,0] := 0
    for i := 1 to m
        for j := 1 to n
            cost:= d[q[i],r[j]]
            DTW[i,j] := minimum(DTW[i-1,j  ] + cost,    // insertion
                                DTW[i  ,j-1] + cost,    // deletion
                                DTW[i-1,j-1] + cost)    // match
    return DTW[m,n]
}
```

Dynamic time warping is an useful and easy-to-implement method for warping sequences. By using dynamic programming, the cost of algorithm is reduced. However there are some weakness in dynamic time warping. The algorithm only considers the difference in time axis for sequences. However the sequences may also differ in Y axis. The sequences may have different means. This problem is eliminated by using offset translation methods. The sequences may have different scaling. It is overcome by applying amplitudes scaling. These problems are related with global differences affecting the entire sequences. However local differences between the sequences also generate problems. A valley in a sequence may be deeper than in other sequences. DTW algorithm try to explain this difference in terms of time axis. This problem may also lead to matching a point which is part of raising trend in one sequence to a point in other sequence which is part of falling trend. Derivative Dynamic Time Warping Algorithm (DDTW) [40] overcomes the described problems. The steps followed in DDTW are same as in DTW. The distance measure between sequences is not Euclidean. The measure is square of the difference of the estimated derivatives of q_i and r_j. Any of the approaches of derivative computation of a sequence can be used for DDTW. Keogh and Pazzani use simply following estimate

$$D[q] = \frac{(q_i - q_{i-1}) + ((q_{i+1} - q_i)/2)}{2} \tag{A.5}$$

This is simply the average of the slope of the line through the point in question and its left neighbor, and the slope of the line through the left neighbor and the right neighbor [40].