AUTOMATIC SEGMENTATION, CLASSIFICATION, AND RE-SYNTHESIS OF HUMAN ACTIONS IN 3D SPACE

By

Seyed Ali Etemad, B.Sc.

A thesis submitted to the Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Master of Applied Science in Electrical Engineering

Ottawa-Carleton Institute of Electrical and Computer Engineering (OCIECE)

Department of Systems and Computer Engineering

Carleton University

Ottawa, Canada, K1S 5B6

August 2009

Your file Votre référence
ISBN: 978-0-494-60269-0
Our file Notre référence
ISBN: 978-0-494-60269-0

Abstract

In this research, we have analyzed and studied the concept of human motion. Various techniques such as nearest neighbour search, hidden Markov models, and artificial neural networks have been utilized to segment and classify human actions in 3D space. A comprehensive mathematical model has been proposed for describing human motion, based on which, transformation functions for style transformation and re-synthesis of motion have been derived. In the end, both the segmentation/classification and re-synthesis procedures have been applied to several action classes, each containing a number of style variations, through which we have demonstrated the significance of our proposed methods.

Acknowledgments

As I acquired a great deal of experience and knowledge through the course of this research, I would like to thank my supervisor Dr. Ali Arya for his support and encouragements. He has contributed tremendously to my academic growth and has inspired me as a teacher, a researcher, and most importantly as a human being. I would also like to thank Dr. James Green for his useful discussions and valuable lessons in the area of pattern recognition

Second, I would like to acknowledge my good friend and colleague Paul Slinger for his aid and time spent for the progress of this research.

Last but not least and from the bottom of my heart, I would like to thank my dear parents and brothers for their support. Their love and faith in me has always been the source of motivation. Without them I would not be where I am today.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1. Background

Human motion is an essential topic of study and research by physicians, biomedical engineers, and graphics and multimedia experts. Recognition and synthesis of actions are the two main categories of human action analysis. The recognition element involves detecting and identifying primary and secondary motor themes [1]. Primary themes are basic mechanical movements which form basic actions such as walking or jumping, while secondary themes are affective and stylistic variations introduced by the actor with respect to mood, gender age, style, physics, and even genetics. Synthesis of action, on the other hand, is the intent to create and/or modify primary or secondary themes of actions performed by an existing character. Motion recognition can be used in a variety of applications from human locomotion analysis to computer vision and surveillance, while

motion synthesis is a key element in character-based simulations, animation, interactive entertainment, and virtual environments.

In this research, we have tackled various issues regarding both recognition and synthesis of human motion data. Motion capture data are employed along with a variety of different tools for analysis of human motion for both fields of recognition and re-synthesis.

1.2. Motivation

Interactive virtual environments are rapidly growing for many applications from arts and entertainment to scientific simulation, education, and the service industry. Very realistic computer games with advanced character and story developments have crossed the boundary between games and movies; virtual social environments such as Second Life (http://www.secondlife.com) now host universities and embassies; and the military training programs use simulations for combat and other situations. Human characters play a significant role in most of these applications, where complex design and powerful multimedia content give the illusion of intelligent computer characters to the users; characters that perform complicated actions, engage in conversation, and display emotions. Historically, character animation is done by the traditional technique of key-framing or recently motion capture for more complicated movements. Regardless of the technique, the majority of animated behaviours in interactive or non-interactive multimedia applications are simply play-back of pre-recorded or pre-animated sequences.

This inability of the systems to generate realistic behaviours procedurally limits the applications in terms of their ability to respond to user interactions in real-time by creating appropriate content. Even in non-interactive applications like animated films, there is minimum support from intelligent animation software, while automating the process of character animation would reduce the production cost and effort, and provide animators with a more effective and systematic way to generate content.

Applications such as simulation and games require partial or full real-time recognition and synthesis while some other aspects of simulation and games along with animation and security applications will not be influenced by the real-time capabilities of the used techniques as the processing is carried out off-line. Segmentation and classification of actions and their styles are mostly employed for simulation, games and security purposes. A virtual environment capable of recognizing and adapting with the performed actions and the way of performing the action is a clear example. This can be interpreted for games, simulators, and security systems. Synthesis of motion data are highly exercised for games and animation where recording new motion capture data is either impossible or costly and time consuming. Figure 1.1 shows the concept of online/offline human motion analysis and its applications.

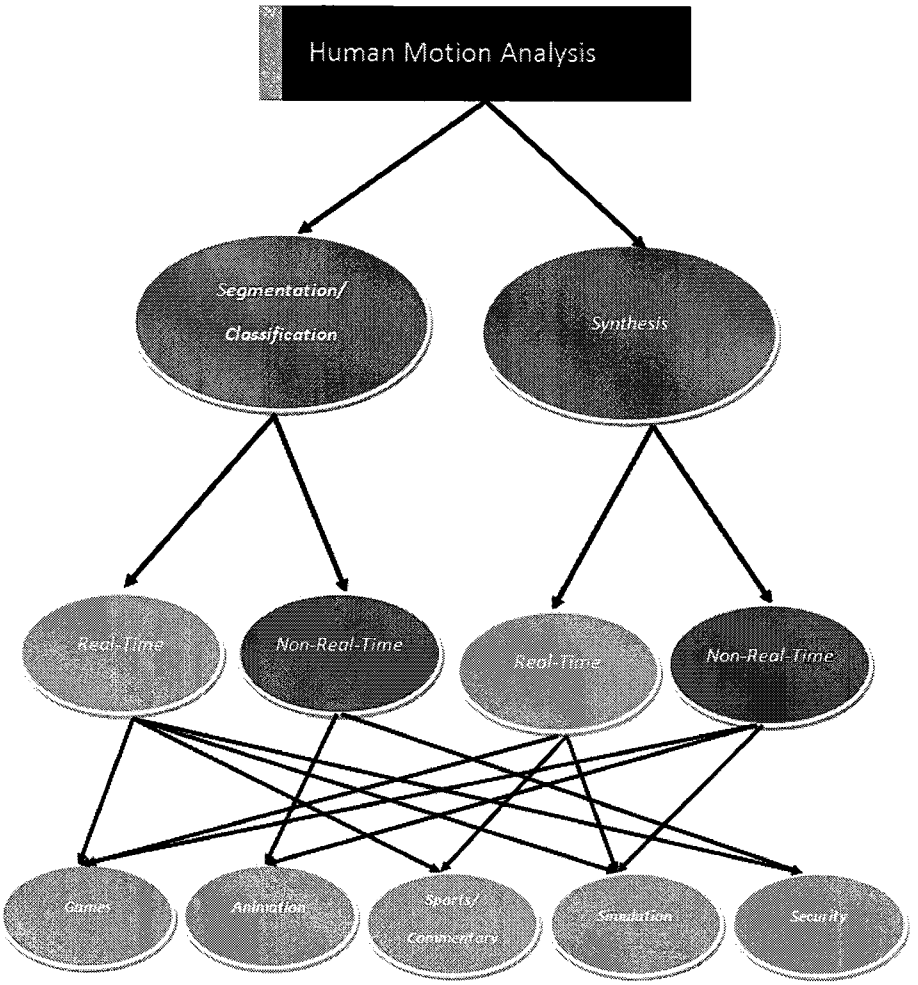

Figure 1.1. Real-time and non-real-time analysis and applications

Motion by definition is the change of the location of a body or its parts with respect to time. The displacement can occur in any defined space. One, two, or three dimensional Cartesian, polar, or any other defined systems by which the location of the object before and after the movement can be measured are valid for defining motion. Displacement, velocity and acceleration are most common in defining motion [2].

The same definition applies for human motion, yet with two constraints. Different parts of human body endure different limitations, thus, natural human motion takes these constraints into account. This is where human motion kinematics steps in, applying specific constraints on each body part with respect to the degree of freedom of the joint which the body part is connected to. The second fact which applies for human motion is a factor of intention. Natural human motion is carried out by human beings with a pre-intention and for a reason. The intention of the motion along with its effect is perceived by others as *action*. While a specific action can be carried out in different styles, the class of the action stays the same. In rare cases, however, the style of the action can affect the perception of a specific action and allow for it to be misclassified. A very *fast walk*, for instance, can easily be mistaken for *run*. These cases, however, are not very common and are usually accounted for as exceptions.

Study and analysis of human motion and kinematics is an area of research which aims at defining, recognizing, classifying and producing human motion data [3]. Various tools have been employed for recognition and classification of human motion, most of which have used visual data for the purpose. While most literature such as [4 – 24] among many others have focused on classifying the actions, recognition and classification of the styles have mainly been ignored. Synthesis of human motion data is a more recent field of research which has developed along with the growth of the digital entertainment industries such as computer games, animated movies, simulators and etc. While motion capture systems are the most practical tool for acquiring human motion data, the high price of the technology as well as the time consumption of the motion capture procedure has motivated researchers to introduce new ways of refining and

manipulating the existing motion data for re-synthesis of the desired motion sequences. Although some techniques have proven to be practical [25 – 27], the need for intelligent re-synthesis systems, lack of quantitative evaluations, and the need for human style classification techniques emphasises on the fact that there is plenty of requirement for better and novel techniques for re-synthesis/transformation of human motion data.

1.3. Problem Definition and Challenges

The goal of this research is to carry out a comprehensive analysis of human motion using motion capture data. We divide this problem into three major parts:

(1) Segmentation of basic moves in a series of motion capture data from within a sequential combination of actions and subsequently **Classification** of the targeted actions and parameters that represent personal and style-based variations (primary and secondary motor themes).

(2) An essential step prior to synthesis of motion data based on more than one existing motion sequence is to manipulate the existing sequences such that for critical features a one-to-one correspondence between the sequences is achieved. This is what we refer to as **Temporal Alignment** of actions which is a very important and fundamental step essential for further processing and blending of the action sequences.

(3) Re-synthesis of actions based on desired parameters, i.e. **Transformation** of secondary themes.

To state in simple term, the first part of the problem is to locate an action from a sequence of different motion frames along the temporal axis, and classifying the action into one of the predefined action classes based on previously learnt data for both the action class and style. The second and third phases are aimed at the process that needs to be completed for combining various primary and secondary themes, resulting in a new animation sequences.

Figure 1.2 presents an outline for human motion analysis where the three mentioned sections are covered. The first task is locating actions temporally in a sequential combination of different actions and classifying them. Temporal alignment is a key step which based on the approach towards the task, could or must be accomplished in order carry out other steps successfully. Synthesis of human motion data is presented as the third section. The dashed arrows in Figure 1.2 show that specific sections can be facilitated by the outcome of other sections. Synthesis for instance is connected to Segmentation/Classification. Although these two sections seem independent, nevertheless Synthesis can be facilitated using the models which have been used to describe human motion for Classification. Yet there is no necessity in using such models as model-free approaches can be utilized for synthesis [3]. Also classification can be carried out based on temporal alignment. In order to carry out synthesis however, it is almost essential to achieve temporal alignment among the training samples and the test sequences.

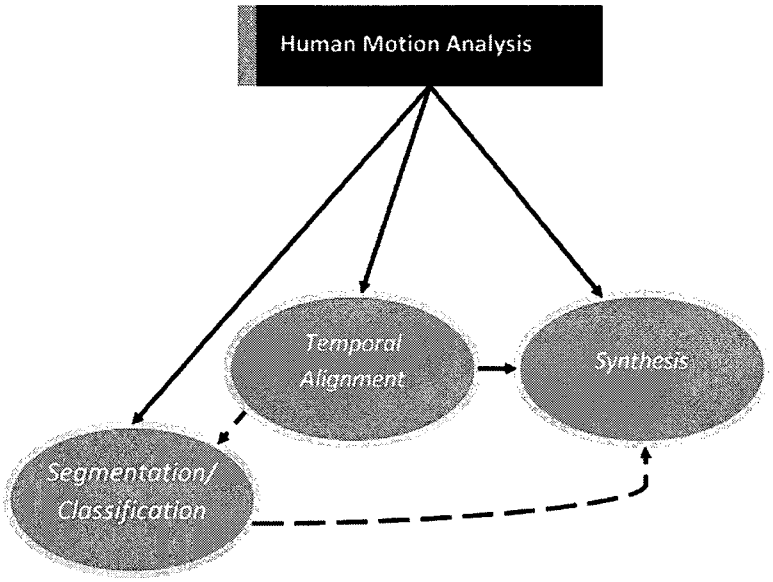

Figure 1.2. Human motion analysis

Direction of view, the way which an action is performed, background, clothing, sequential vagueness, indefinite length of actions, camera moment, and high degree of freedom in human actions [28 – 31] are all challenging tasks to conquer for an ideal automated system. While employment of motion capture data eliminates some of the above (such as background and clothing), the rest remain to be dealt with and tackled through the system and using different techniques which will be discussed through the course of this dissertation.

In this research, we have employed over 100 action sequences (nearly 4000 frames) for training the different designed systems. Each system may require a different number of training samples based on the techniques used throughout the system. Subsequent to the training process, 90 action sequences were employed for testing and evaluating each system. In the end, the overall functionality of this research can be described as an

automated system capable of segmenting and classifying human motion capture data (for specific primary and secondary themes), which can then transform certain secondary themes associated with each segmented and classified action.

1.4. Contributions

This research presents the design of a system capable of performing an inclusive study of human motion capture data, and performing the three mentioned problems in this regard which has resulted in a number of publications [32 – 35]. The research is carried out in three phases:

- Locating and segmenting specific actions temporally and classifying the located actions for class and style.
- Piecewise time warping of the sequences for achieving temporal alignment.
- Re-synthesis of actions for style transformation.

As opposed to most research in this field were actions have been selected manually and then used for classification through the system, we have provided the means for automatic segmentation of the action – phase 1. Also in phase 1 the classification of the temporally located actions have been performed using different methods and compared for determining the most accurate approach. Along with the classification of primary action themes, the secondary themes have also been subject to classification. Various

tools such as nearest neighbour search, hidden Markov models, and artificial neural networks have been employed for recognition and classification, and the experimental results have been compared and analyzed. Action class and style class have both been subject to classification, and various characteristics of the performer along with the action which he/she performs have been distinguished.

In phase 2, piecewise time warping is proposed and performed resulting in temporal alignment which is an essential step towards analysis of human motion.

In phase 3, toward re-synthesis of actions and transformation of secondary themes, we have proposed a mathematical model for describing human motion and derived the style transformation functions accordingly. We have also used artificial neural networks for this purpose. Both methods have been experimented and valuable results have been acquired. A numerical measure for evaluation of synthesized motion has also been proposed and used to validate the outcome.

We can summarize the major contributions of this research as the following:

- Providing the means for automatic segmentation and classification of actions using motion capture data.
- Classification of various actor styles alongside the action classes.
- Proposing and applying a novel technique for temporal alignment (piecewise time warping).
- Proposing a model for human motion, capable of describing both primary and secondary themes and using the same model for style transformation.
- Employing ANN (as well as the model) for style transformation.

- Employing Pearson's Correlation measure for evaluating synthesized data.

1.5. Thesis Outline

In the course of this text, the complete process of construction of the system explained earlier will be discussed. In Chapter 2 a comprehensive review of some key literature in the field of human motion analysis is carried out. Chapter 3 deals with the data type (motion capture data) used in this research. Preprocessing and noise reduction are discussed in this chapter as well.

Chapters 4 through 6 address the three main phases of this research described in Section 3 of Chapter 1. Chapter 4 tackles the problem of locating an action along the temporal axis and segmenting the sequence into different actions. This step is critical towards automation of the system where manual selection of actions would no longer be required. Classification of actions using different methods is also provided in Chapter 4. In this chapter the classification of both primary and secondary themes is discussed. Chapter 5 deals with the very critical measure of temporal alignment of actions prior to re-synthesis of actions and secondary theme transformations in Chapter 6.

Finally in Chapters 7 the experimental results are discussed and analyzed, and in Chapter 8 the concluding remarks and the potential areas and problems for future work are presented. This is followed by an overview on data acquisition and other key tools which were utilized for this research in Appendix A, B, C, and D.

Chapter 2: Related Work

2.1. Introduction

Many researchers have worked on human motion segmentation, classification and re-synthesis. In this chapter we briefly review some of these works. The focus of this research has been the use of motion-capture data, and so we have focused on researches which have employed such data. Meanwhile looking at other techniques using video and still images can provide possible ideas for tackling the problem at hand. Some of these type of literature are covered in Section 2.1 while the motion capture-based methods are covered through Section 2.2.

Similar to the course of this study, the related works which have employed motion capture data can also be divided into two major categories: segmentation/classification and synthesis. While the former have been used in areas such as surveillance and security, sport commentary tools, Human Computer Interaction (HCI), and motion-based

inputs for interactive applications, the latter focuses on applications such as animation, games, and simulation. Thus we categorize the literature accordingly into segmentation/classification and re-synthesis (transformation). In this chapter, a number of key literature in the field of human motion analysis have been reviewed for both mentioned categories based relatedness of the approaches used for tackling the problem. The literature have been selected from among a variety of research reports with the aim of providing an insight on the existing techniques for the problem at hand as well as coming about useful ideas for employing in this research.

2.2. Segmentation and Classification

Segmentation and classification of human actions have been studied by many researchers. Two most common types of input for human motion analysis are:

- Visual (image/video) input
- Motion capture data

The first category (usually video), has been subject to research more extensively since video recording technology has been around far longer than motion capture systems which is considered a recent technology. Also the first types of input are popular since there are a variety of techniques for optical feature extractions such as optical flow, histograms, analysis of different color channels, etc. However, the drawbacks to video

inputs remain, as background vagueness, clothing, lighting, and point of view tend to limit the systems at hand.

In this section, we study the literature published in both categories, while more emphasis is put on the second which this research is based on.

2.2.1. Visual (Image/Video) Input

When dealing with visual data, optical flow provides unique features which have made it very suitable for training systems for human action recognition. In [4 – 10, 36] and various other papers, the utilized data is based on optical flow vectors such as that presented in Figure 2.1, or some refined format of flow features. The Lucas-Kanade algorithm [37] for optical flow computation has shown to be most common and useful when optical data are available [5, 8, 9, 38, 39]. Figure 2.1 presents computed optical flow vectors used for human action classification.

When using optical flow features, the excessive number of correlated data must be reduced. PCA (Principal Component Analysis) [6 – 8, 36] is the most common tool for this purpose. Other techniques such as Adaboost [5] and flow histograms [4, 10, 36] have also been suggested to create stronger features. Each motion frame is usually divided into n subsections (channels) and the refining algorithm is performed on each individual channel. The outcome for each channel is then used to train the system. For example, in [10] Zhu et al. create a flow histogram for three channels representing three vertical slices for each frame. Ikizler et al. [4] divide the frames into 9 equal rectangles and

subsequently form histograms for each of the 9 channels. This type of partitioning of the frames does not take into account the fact that some channels, for instance (for scene similar to Figure 2.1) the channels corresponding to the corners of the image, are occupied by insignificant or almost no data. In the training process however, all channels are employed with equal weight and influence on the system.

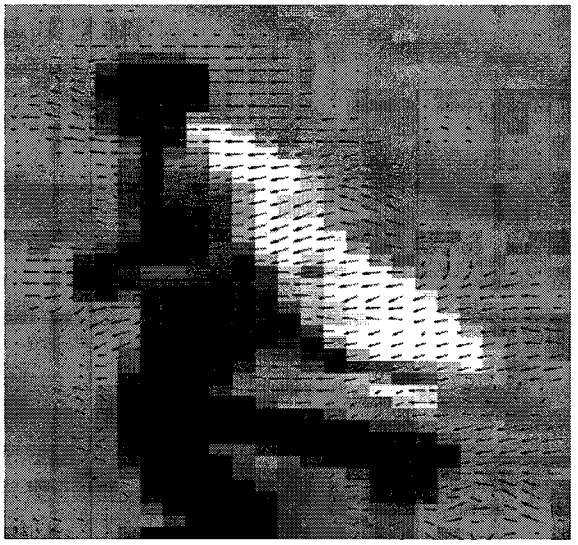

Figure 2.1. Optical flow vectors for human motion analysis reproduced directly from [4]

There are basically two problems to tackle when dealing with recognition of human actions. The first is to locate a specific action along the temporal axis. This means, in a sequential combination of actions performed by an actor, the goal is to determine when an action begins. A sliding search window may be an option for this purpose [4], yet since for each new position of the window the entire classification process must take place, it shows to be very time consuming. Most of the literature, such as [4, 5, 23, 24, 40

– 45] among many others, have avoided this problem by manually locating actions or performing one action at a time, thus focusing on the classification problem. The other main problem is to classify the selected action. Varieties of different tools have been utilized for action recognition and classification.

Hidden Markov models (HMM) are one of the most common tools [7, 8, 14 – 16]. Ahmad and Lee [7] have used optical flow along human body shape feature vector information from different angles for recognition. They have used Principle Component Analysis (PCA) for dimensionality reduction, prior to creating and representing each action using HMM for each viewing angle. Li [8] has employed optical flow features oriented by histograms for recognition. Similarly, PCA has been used in [8] for dimensionality reduction prior to training the HMM classifier. In [14] by Yamato et al, action sequences are converted to image feature vector sequences and used for training HMM which is then used for classification. Human body figure and specifically that of the arms are used by HMM [15] by Wu et al. for action classification. Li and Fukui [16] classify actions by means of HMM trained by 2D trajectories of different points on human body. They show that such data are sufficient for recognition regardless of the viewpoint.

K-Nearest Neighbor (KNN) [6, 9, 36, 46] have also been utilized for recognition of human motion when dealing with visual data where the simple nearest neighbour approach has been preferred by researchers. Ali and Shah [6] conduct a vast amount of preprocessing on optical flow vectors prior to using KNN for classification. In one of the few papers dealing with style, Wang [36] applies KNN on human silhouettes and flow vectors for determining whether the actor is performing a normal walk or not. Efros et al.

[9] have classified actions at a distance using the same classifier. Madabhushi and Aggarwal [46] have focused on the head movement for classification of actions using KNN.

Artificial Neural Networks (ANN), have also been used largely for action recognition [18 – 22]. Kornprobst et al. in [18] show that visual data used to train neural networks are an effective and efficient means for human action recognition. In [19] Babu et al. employ MHI (Motion History Image) and train neural networks for the recognition task. Self organizing neural networks have also been utilized by Kuniyoshi and Shimozaki [20, 21]. Last but not least, in [22] Theodoridis and Huosheng use a variety of different neuron/layer Multi Layer Perceptron (MLP) networks along with different training functions to classify human actions and compare the performance for different situations.

The mentioned methods along with some other tools such as support vector machines (SVM) [4, 10, 17] have been largely used and successfully classified human motion in cases where visual motion data are available.

2.2.2. Motion Capture Data

A motion capture system is a system which can digitally record and transfer motion onto a model. There are different forms of motion capture system such as optical, inertial, mechanical, and magnetic. The motion capture system which we have used in this research is an optical system. There are various forms of outputs which motion capture

systems can provide, yet the format which we have used in this research along with further description on the system itself is provided in Appendix A and Chapter 3 section 2.

In this section, we will review some key literature which have employed motion capture data for both segmentation and classification. Some of the techniques used based on this type of data are quite similar to those used for video inputs, while some others are quite different. The diversity of the techniques which are capable of utilizing motion capture data is quite larger than that of video inputs since in addition to the practical pattern classification techniques, data mining and even motion modelling play a significant role.

Lv and Levatia [23] have used the Viterbi algorithm for single view recognition of actions and propose a Pyramid Match Kernel algorithm and compute matching scores between the feature data sets. They automatically extract key poses for this purpose based on motion energy charts as shown in Figure 2.2. Their method reaches a classification accuracy of around 80%.

Zhou et al. [47] have focused on segmenting specific actions from motion capture data. They have defined the problem as an extension to practical segmentation techniques. Aligned Cluster Analysis (ACA) has been proposed by the authors for temporal segmentation of actions. Figure 2.3 illustrates the goal of this research. This problem is what we, in our research, have referred to as locating actions. Despite the relatively high accuracy achieved in their paper (approximately 97%), a strong drawback

is that the exact number of segments must be manually provided to the algorithm to avoid local minima.

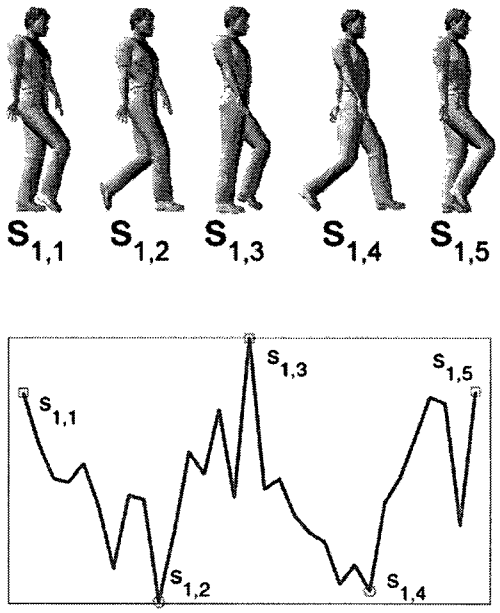

Figure 2.2. Extraction of key poses based on motion energy chart reproduced directly from [23]

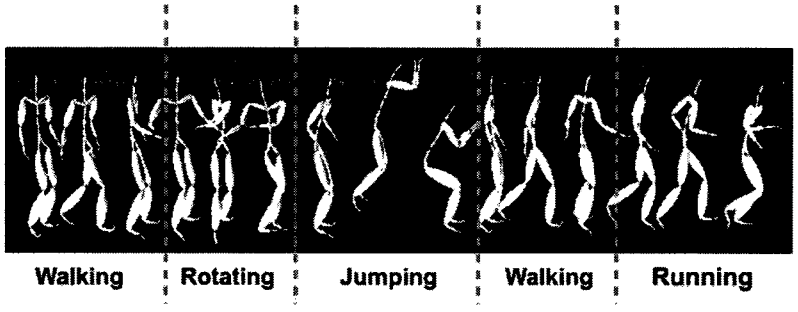

Figure 2.3. Temporal segmentation of actions reproduced directly from [47]

Ali et al. [24] model human actions using the theory of chaotic systems. The non-linear dynamic system generating the action is represented by trajectories of specific reference joints: the head, two hands and two feet. An overall classification accuracy of 89.7% is achieved through this technique. Ishiyama et al. [40] use markerless human motion capture samples and describe human motion by a low dimensional linear model. While the proposed model shows to be promising in the sense that it is robust for different body types, there is no indication as to how accurate the classification accuracy is.

Recognition and generation of motion data are carried out using HMMs by Kulic et al. [41]. In their work, different motions are organized in hierarchical tree fashion where nodes closer to the root correspond to more general motion features while further motions represent more detailed motion descriptors.

In [48] Song et al. use the concepts of maximum likelihood to build an unsupervised system applicable to both greyscale images and motion capture data for recognition of human motion. Their system employs an algorithm which uses differential entropy of the variables to locate the optimal structure of the decomposable model which is claimed to perform superior to manually constructed models. Their model however, maintains a trade-off between model complexity and accuracy and has only been tested on walking sequences and it is not clear on how it will perform with multiple action classes. A different type of probabilistic models called Switching Linear Dynamics (SLD) is employed for human action classification by Shimosaka et al. [42]. SLDs are a combination of HMMs and Linear Dynamics (LD) which perform stochastically. They also demonstrate that utilizing their proposed kernel for training support vector machines

shows to be very accurate for performing the task where 6 classes of action have been employed for this approach. Figure 2.4 presents the action classes. Nevertheless as the goal of the research was the design of specific kernals, a multi-class classifier is not designed and no results provided. It can also be noted that the 6 chosen classes of action or mostly (except for walk and run), static motions.

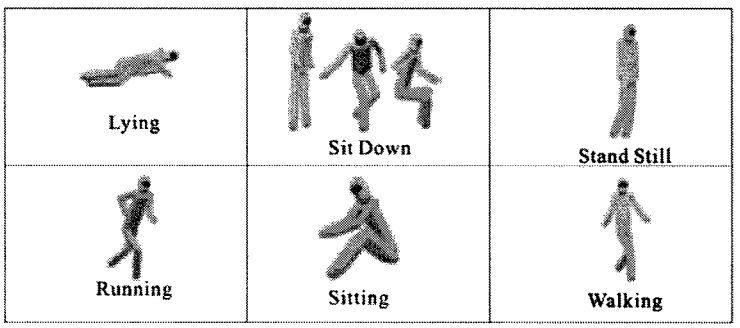

Figure 2.4. Recognized classes of action reproduced directly from [42]

Shimosaka et al. [43] have proposed an online classification procedure which is based on selection of critical motion features. Their system is based on independent and parallel classifiers, each dealing with a specific feature. Figure 2.5 illustrates the performance scheme of the system. The authors in the mentioned paper, instead of fundamental motion features have used Combinational Motion Features (CMFs) – which are a combination of motion features capable of describing the relationship between different body parts, along with SVM for classification of two action classes of walking and running achieving an accuracy of approximately 92%.

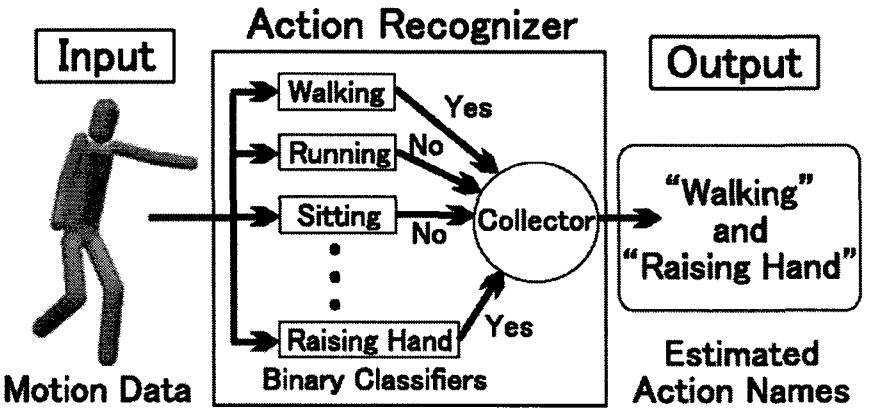

Figure 2.5. Scheme of the system reproduced directly from [43]

Mori et al. [49] have employed motion capture data to train HMMs for segmentation and segmentation of actions. In this paper the authors have used the evaluation of the starting and ending frames for segmentation and an online SVM-based classifier is used for calculating action probabilities. Then the action probabilities are analyzed by HMMs for determining whether a specific frame is to be segmented or not. As the proposed algorithm shows to be robust for segmentation, it is not clear whether classification of the segmented actions is foreseen by the algorithm and if so, how accurately the task is performed.

In [44] SVMs alone have been employed by Mori et al. for classification of actions. In this paper, the authors tend to find specific motion features using kernel parameters which minimize the generalization error of the classifier. Figure 2.6 shows an example of motion feature discovery by the system.

Using a rather different approach towards human action recognition [45] Parameswaran and Chellappa have proposed creating a 3D-invariance space for each action. Each action is characterized in this space by a curve, and test actions are probabilistically analyzed in this space with respect to the defined curves representing each action. The technique is tested for 5 different viewpoints for each action in both 2D and 3D, while the former maintains a mean accuracy of nearly 91% and the latter shows a mean accuracy of approximately 89%.

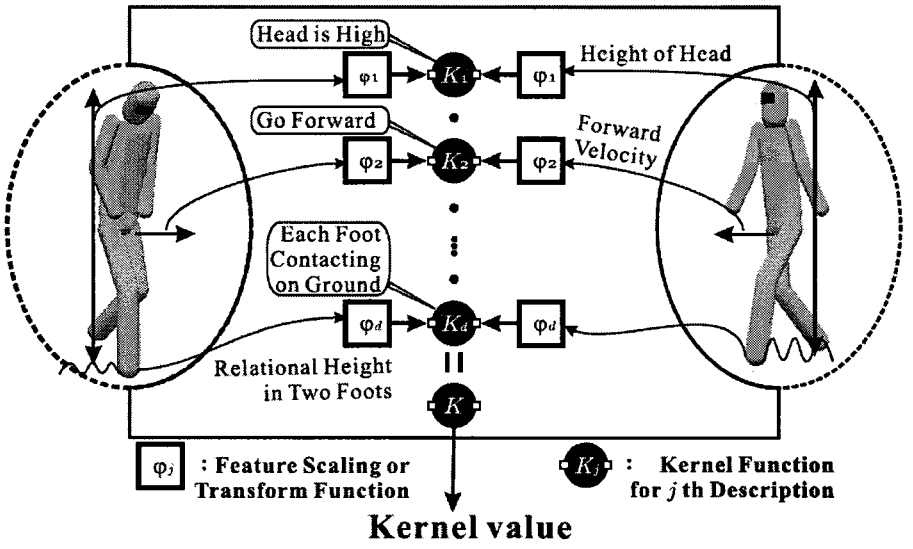

Figure 2.6. Finding motion features for Walking using kernel parameters reproduced directly from [44]

Sheikh et al. [50] have employed both motion capture data and direct video recordings, and based on the earlier factorization methods of Bregler et al. [51] and Tomasi and Kanade [52], described actions as a "combination of spatio-temporal action

basis". This technique enables them to describe the different actor styles and physiques, which are then used when classifying different actions. Figure 2.7 presents a typical action performed differently by different actors. The recognition results of this method however, are not highly accurate.

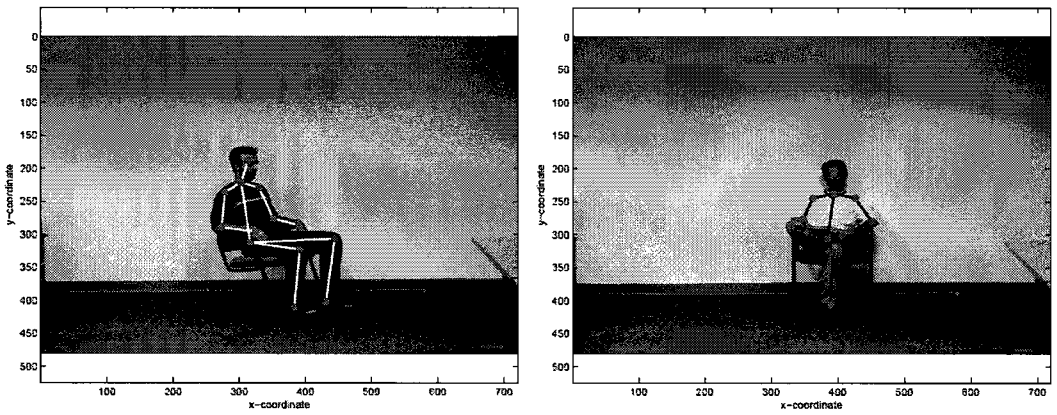

Figure 2.7. A sample of an action performed by different actors reproduced directly from [50]

Brand and Kettnaker [53] have used the concepts of entropy minimization of joint distributions to train HMMs by motion capture data. The classifiers have then been used for classifying actions from normal video sequences. The human figures are first converted to 2D silhouettes, which are then employed by the HMMs. Thus this interesting approach is applicable for ambiguous scenes as well.

Through a rather simpler approach compared to [53], Li and Fukui have trained HMMs by means of factorization of motion capture data for view-invariant classification

of human actions [54]. Despite using motion capture data which provides 4D data, they have not used the Z information during the experiments yet claiming view invariance for the system. However, they have accomplished a classification accuracy of nearly 95%.

2.3. Synthesis

In this section we review some of the important literature in the field of human motion synthesis. Although a significant amount of research has been carried out in the field of action recognition, synthesis is considered a rather new subject for research as the requirement for such studies have recently been introduced by the growth of the animation and gaming industry.

Statistical models have been one of the practical tools for human motion synthesis [55, 56]. Tanco and Hilton [55] have trained a statistical model which employs a database of motion capture data for synthesizing realistic motion sequences and using the start and end of existing keyframes, original motion data are produced. Li et al. [56] define a motion texture as a set of textons and their distribution values provided in a distribution matrix. The motion texton is modeled by a Linear Dynamic System (LDS). A maximum likelihood algorithm is designed to learn from a set of motion capture based textons. Finally, the learnt motion textures have been used to interactively edit motion sequences.

Egges et al. [57] have employed Principal Component Analysis (PCA) to synthesize human motion with the two deviations of small posture variations and change of balance.

This approach is useful in cases when an animated character is in a stop/freeze situation where in reality no motionless character exists. Liu and Papovic [58] have applied linear and angular momentum constraints to avoid computing muscle forces of the body for simple and rapid synthesis of human motion. Creating complex dynamic motion samples such as swinging and leaping have been carried out by Fang and Pollard [59] using an optimization technique applied along with a set of constraints, minimizing the objective function. Pullen and Bregler [60] have trained a system that is capable of synthesizing motion sequences based on the key frames selected by the user. Their method employs the characteristic of correlation between different joint values to create the missing frames. In the end, quadratic fit has been used to smooth the estimated values, which has resulted in more realistic looking results. Brand and Hertzmann [25] employ probabilistic models for interpolation and extrapolation of different styles for synthesis of new stylistic dance sequences using a cross-entropy optimization structure which enables their style machine to learn from various style examples. Safonova et al. [61] define an optimization problem for reducing the dimensionality of the feature space of a motion capture database, resulting in specific features. These features are then used to synthesize various motion sequences such as walk, run, jump and even several flips. This research shows that the complete feature space is not required for synthesis of human motion.

Hsu et al. [26] conduct style translations such as sideways walk and crouching walk based on a series of alignment mappings followed by space warping techniques using an LTI model. While this technique shows to be functional for the mentioned style translations, more minute style variations such as those related to gender, energy and age

have not been tested for. Figure 2.8 illustrates a weak punch transformed into an aggressive punch using their method.

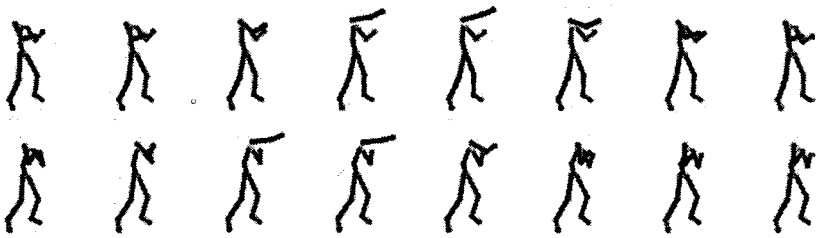

Figure 2.8. Style transformation of punching reproduced directly from [26]

Style transformations have also been studied by Shapiro et al. [62]. Motion data is first decomposed into components called style components by means of Independent Component Analysis (ICA), which is a mathematical technique for separating multivariate signals. The decomposed style components are assumed to be independent of the action class and are capable to convert, for instance, a normal walking sequence to a sneaky walk as shown in Figure 2.9. The same component can also be applied to running sequences, converting the original run into a sneaky run.

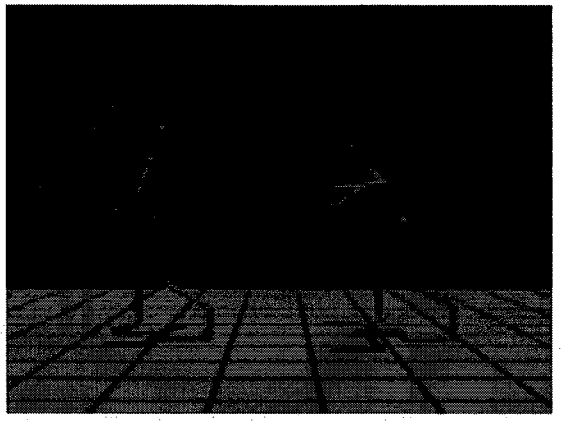

Figure 2.9. Normal walk (left) transformed to sneaky walk (right), reproduced directly from [62]

Rose et al. [27] have employed time warping as the first step towards synthesis of human motion which is an approach similar to ours. Specific kinematic constraints have been exercised along with interpolation of styles. The constraints, however, are selected manually and based on the nature of the action, as opposed to our proposed method in Chapter 5 where we tend to automate the constraints. Time warping is also studied in [63] where Hsu et al. have used a specific reference motion for defining the constraints of the time warping procedure. Although this technique proves to be useful in some cases, it is not discussed whether the reference action is capable of determining the most suitable constraints for all cases.

Paul Slinger has created an animation toolkit using Autodesk Maya which employs the generated data by our research. We discuss the toolkit in more detail in Appendix D. In regards to creating the animation toolkit, there are few animation systems that provide users with existing animation sequences that can be manipulated and updated to output a

new and unique animation sequence. One current example is a dynamic motion synthesis program, Endorphin 2.7 by Natural Motion. Endorphin uses adaptive character behaviours and physics to create their animation simulations in real-time. Although the simulations are customizable, they lack a dimension of personality, relying on physics changes to provide movement variations. Previous work in the motion capture field has been focused on overcoming the shortcomings and complexities of these systems; includes efforts to retarget motion data to new characters [64], motion adaptation techniques and library construction [65], and optimizing the motion capture pipeline for real-time applications [66]; plus, optimizing the mapping procedure of the 3D marker position data recorded by optical motion capture system to the joint trajectories together with a matching skeleton based on least-squares fitting techniques [67].

Chapter 3: Data and Preprocessing

3.1. Introduction

Data preprocessing is an essential procedure which greatly affects the success of machine learning algorithms. Various manipulations ranging from noise reduction to vectorization and clustering of data, and even dimensionality reduction are considered as preprocessing.

In this chapter we will discuss the format of the motion data used throughout the research. The acquired data by the system are scalar time-varying 3D values which are assumed to be free of noise –or containing insignificant amount of noise, due to the high accuracy of the Vicon motion capture system and based on visual evaluation of the animated version of the data using the BVHacker software. Meanwhile filtering is applied for ensuring minimum error in the numerical values. The required filtering

processes along with the other preprocessing techniques used in this research are discussed in this chapter.

3.2. Data Type

Motion capture data obtained by means of a six-camera Vicon system (more details in Appendix A) in the School of Information Technology, Carleton University, have been used for this research. We have asked various actors to perform the required basic classes of actions in each variation style, and the necessary database has been created. Figure 3.1 illustrates a caption of a motion capture session.

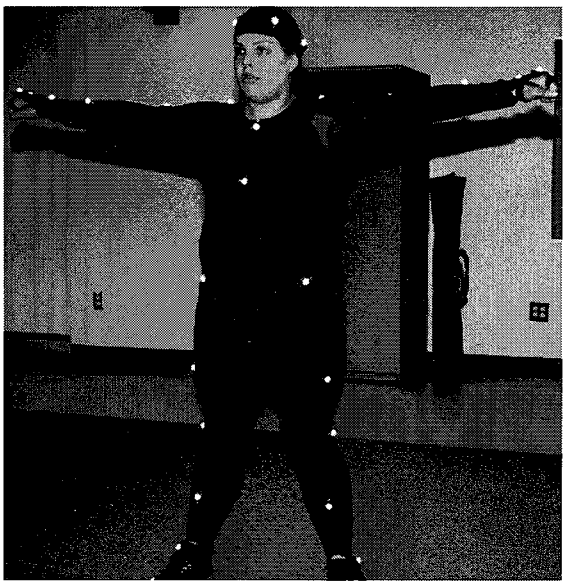

Figure 3.1. Motion capture session

The motion capture data come in the form of Equation 3.1, where $\boldsymbol{D_i}$ are the Cartesian values for the hip marker in 3D space with respect to the calibration origin and $\boldsymbol{\Theta_i}$ are the rotation angles in degrees for each marker. There are m rows, denoting m frames.

$$A = \begin{bmatrix} D_1 & , & \Theta_1 \\ D_2 & , & \Theta_2 \\ \vdots & & \vdots \\ D_m & , & \Theta_m \end{bmatrix} \tag{3.1}$$

The positioning marker is presented in Figure 3.2 by the marker connecting the two legs (marked by $\otimes$), which corresponds to the marker placed on the hip. This marker provides the Cartesian measures for locating the actor in each frame of action. Figure 3.2 also shows the axis of the joints of the left leg. Each marker on the body possesses its own frame of reference similar to those shown for the left leg.

Figure 3.3 illustrates a masculine walking sequence for 45 frames where snapshots of each 5^{th} frame have been presented.

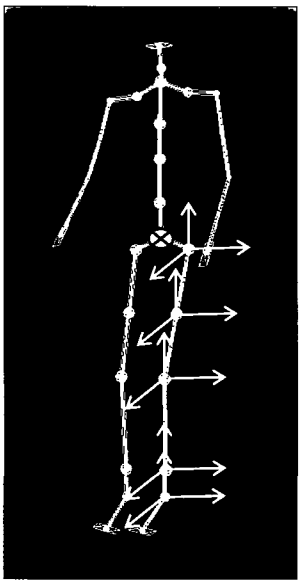

Figure 3.2. Hip positioning marker shown by ⊗ and the axis of the left leg markers are presented

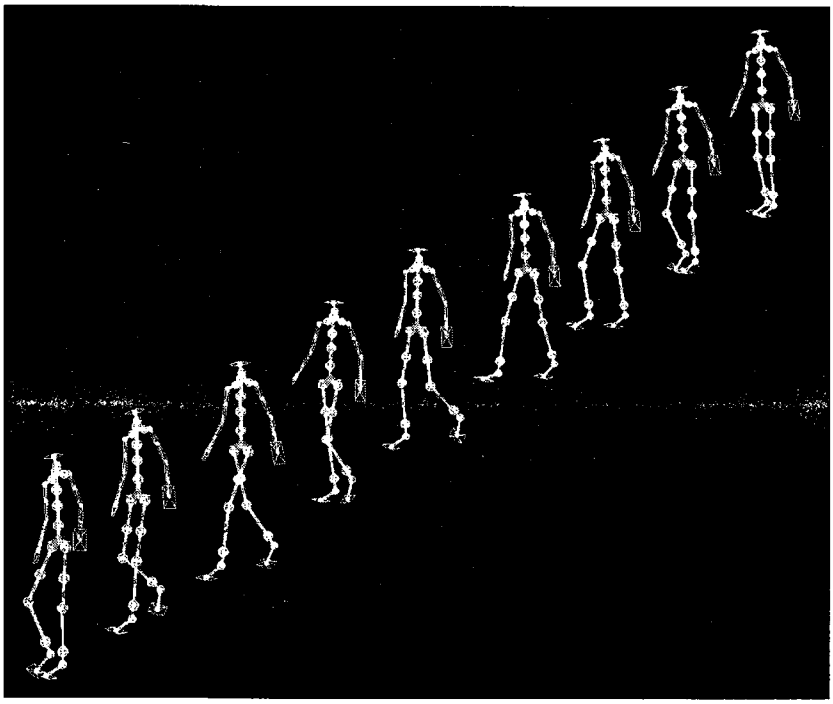

Figure 3.3. Masculine walk

The rotation values matrix for i^{th} marker for frames *1* to *m* is as follows, where $\theta_j^{x_i}$ denotes the rotation value of the *x* coordinate in space, related to i^{th} marker of the j^{th} frame:

$$\overline{\theta}^{i} = \begin{bmatrix} \theta_1^{x_i}, \theta_1^{y_i}, \theta_1^{z_i} \\ \theta_2^{x_i}, \theta_2^{y_i}, \theta_2^{z_i} \\ \vdots \quad \vdots \quad \vdots \\ \theta_m^{x_i}, \theta_m^{y_i}, \theta_m^{z_i} \end{bmatrix} \tag{3.2}$$

In Equation 3.3, the complete angular rotation matrix of *m* frames and *n* markers is presented.

$$\begin{bmatrix} \Theta_1 \\ \Theta_2 \\ \vdots \\ \Theta_m \end{bmatrix} = \begin{bmatrix} \left(\theta_1^{x_1}, \theta_1^{y_1}, \theta_1^{z_1}\right), \left(\theta_1^{x_2}, \theta_1^{y_2}, \theta_1^{z_2}\right), \ldots, \left(\theta_1^{x_n}, \theta_1^{y_n}, \theta_1^{z_n}\right) \\ \left(\theta_2^{x_1}, \theta_2^{y_1}, \theta_2^{z_1}\right), \left(\theta_2^{x_2}, \theta_2^{y_2}, \theta_2^{z_2}\right), \ldots, \left(\theta_2^{x_n}, \theta_2^{y_n}, \theta_2^{z_n}\right) \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ \left(\theta_m^{x_1}, \theta_m^{y_1}, \theta_m^{z_1}\right), \left(\theta_m^{x_2}, \theta_m^{y_2}, \theta_m^{z_2}\right), \ldots, \left(\theta_m^{x_n}, \theta_m^{y_n}, \theta_m^{z_n}\right) \end{bmatrix} \tag{3.3}$$

The related data for the hip positioning marker for frames *1* to *m* is shown by ***D***. In Equation 3.4 d_i^x represents the value of the *x* coordinate of the distance of the hip marker with respect to origin for frame *j*.

$$\begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_m \end{bmatrix} = \begin{bmatrix} d_1^x, d_1^y, d_1^z \\ d_2^x, d_2^y, d_2^z \\ \vdots \quad \vdots \quad \vdots \\ d_m^x, d_m^y, d_m^z \end{bmatrix} \tag{3.4}$$

The final form of the data is presented by Equation 3.5 where $\theta_i^{x_j}$ represents the rotation values of the x coordinate of the j^{th} marker for the i^{th} frame and d_i^x represents the position of the x coordinate of the hip marker for the i^{th} frame.

$$A = \begin{bmatrix} \left(d_1^x, d_1^y, d_1^z\right) & \left(\theta_1^{x_1}, \theta_1^{y_1}, \theta_1^{z_1}, ..., \theta_1^{x_n}, \theta_1^{y_n}, \theta_1^{z_n}\right) \\ \left(d_2^x, d_2^y, d_2^z\right) & \left(\theta_2^{x_1}, \theta_2^{y_1}, \theta_2^{z_1}, ..., \theta_2^{x_n}, \theta_2^{y_n}, \theta_2^{z_n}\right) \\ \vdots & \vdots \\ \left(d_m^x, d_m^y, d_m^z\right) & \left(\theta_m^{x_1}, \theta_m^{y_1}, \theta_m^{z_1}, ..., \theta_m^{x_n}, \theta_m^{y_n}, \theta_m^{z_n}\right) \end{bmatrix} \tag{3.5}$$

Figures 3.4 to 3.6 present the movement of the hip positioning marker in 3D Cartesian space for three action classes of walking, running, and jumping with masculine secondary theme and Figures 3.7 to 3.9 presents the orientation of the *LeftLegRoll* marker in 3D angular space in degrees for the same classes of action.

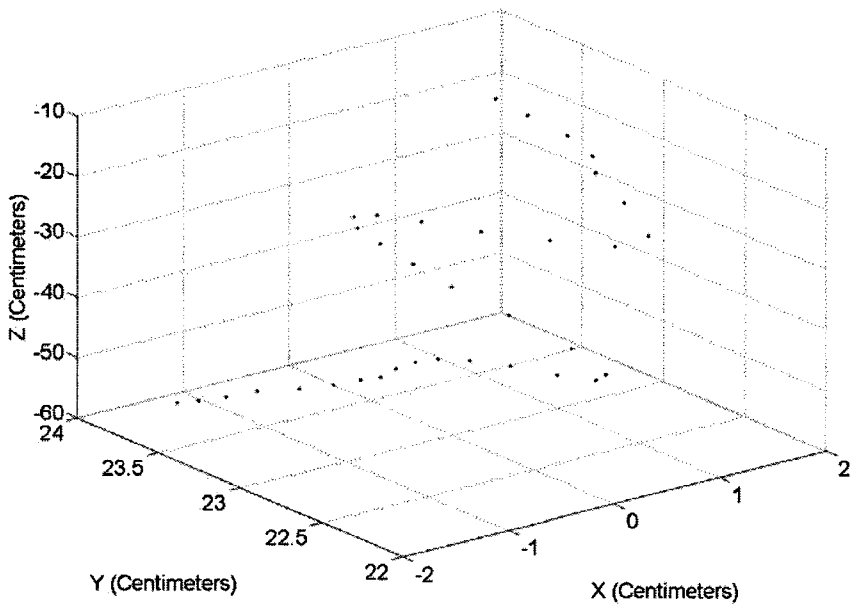

Figure 3.4. Hip marker data for masculine walk

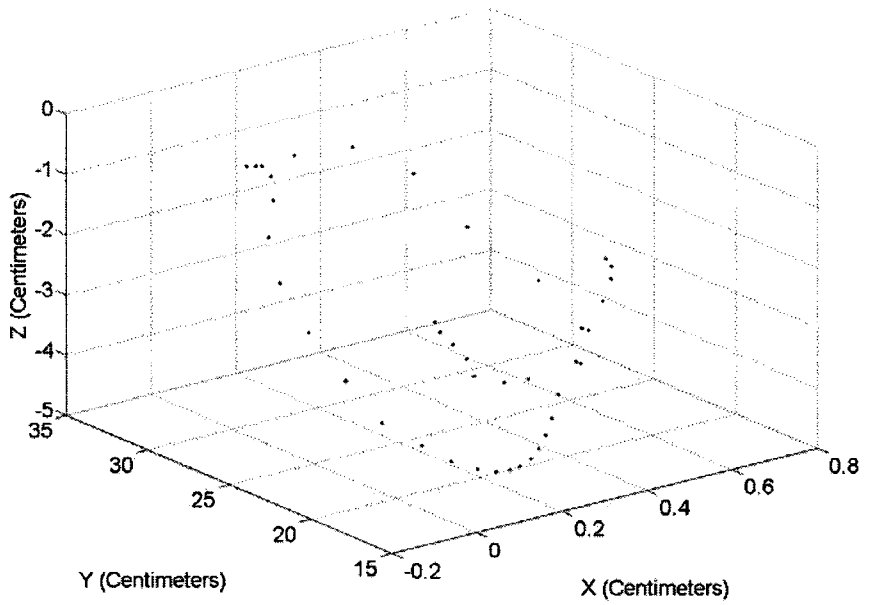

Figure 3.5. Hip marker data for masculine jump

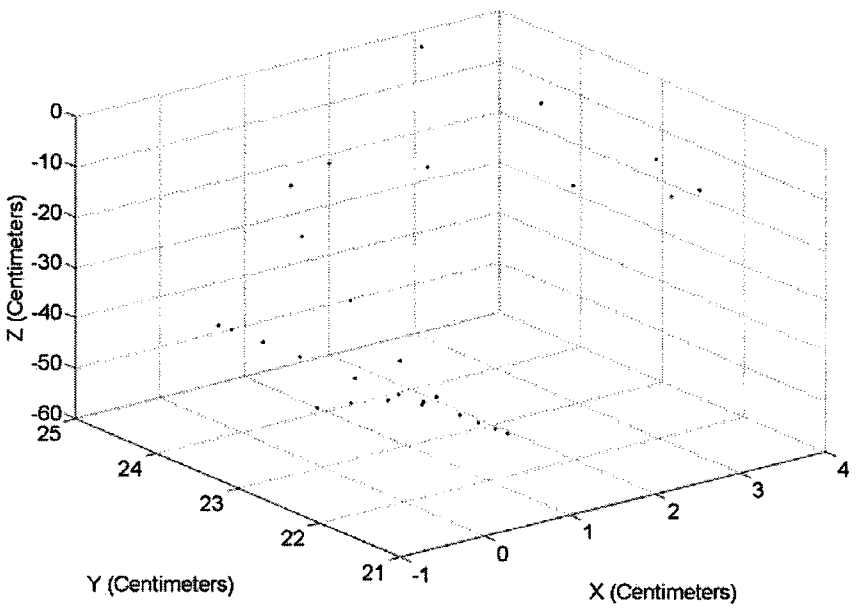

Figure 3.6. Hip marker data for masculine run

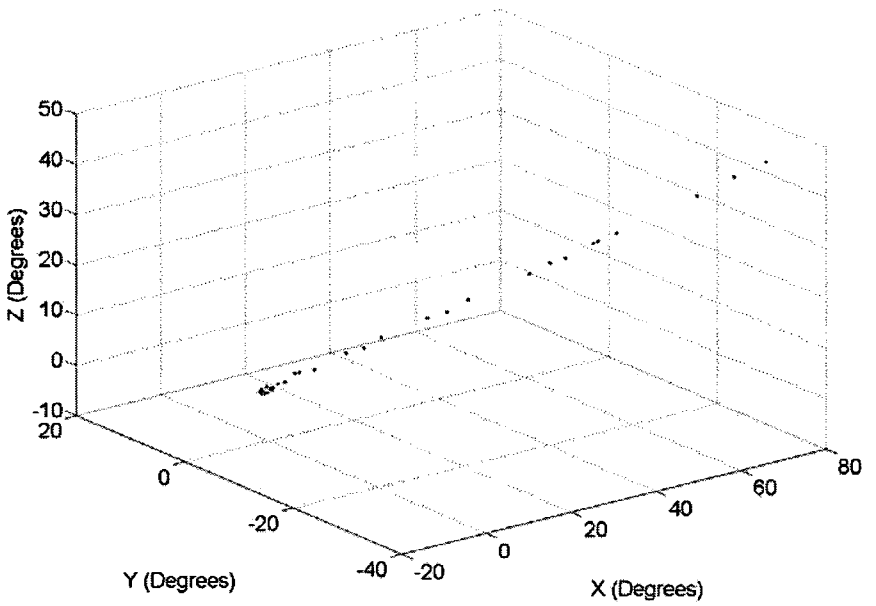

Figure 3.7. LeftLegRoll marker data for masculine walk

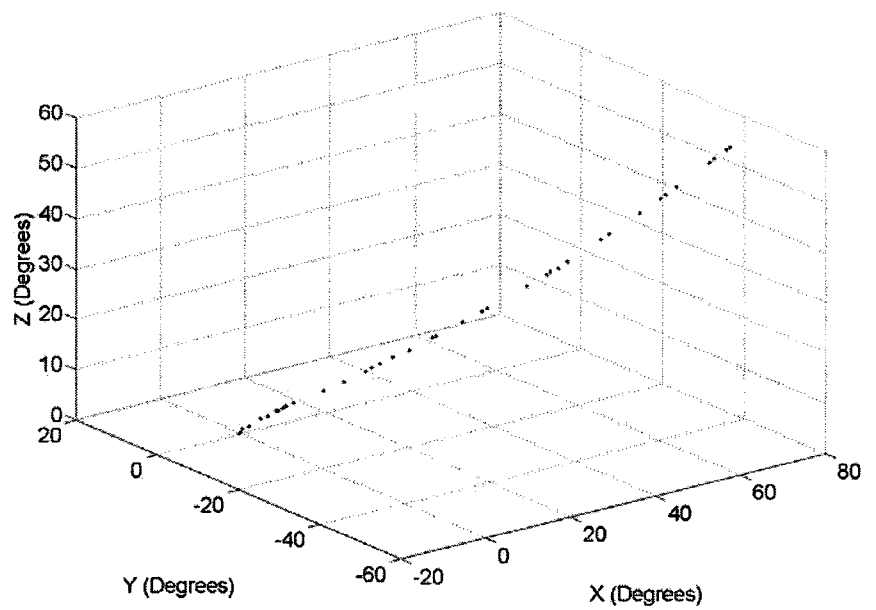

Figure 3.8. LeftLegRoll marker data for masculine jump

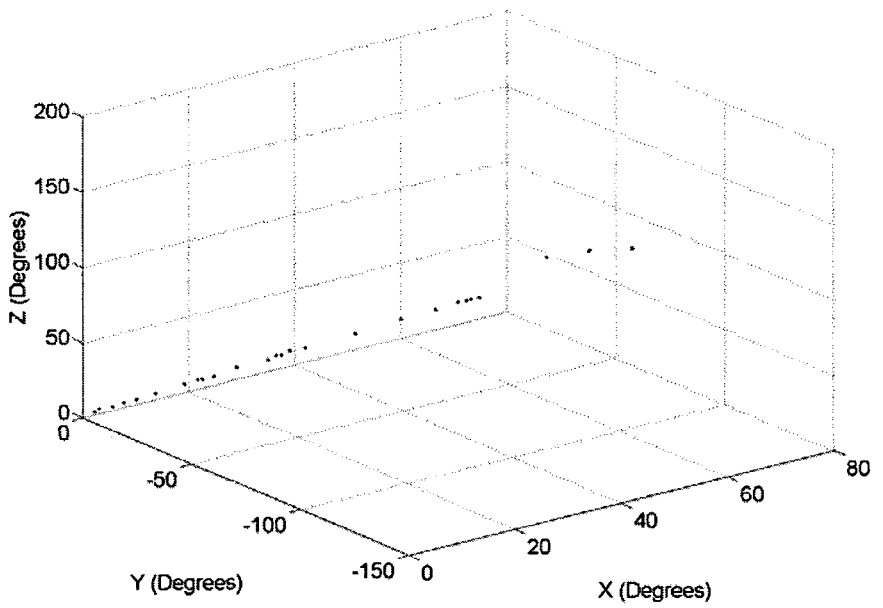

Figure 3.9. LeftLegRoll marker data for masculine run

3.3. Preprocessing of Motion Capture Data

The relative scalar values are converted to meaningful vectors by subtracting each frame from its previous one and dividing by the time difference of consecutive frames (*1/fr*). This is shown through Equations 3.6 to 3.10. $\underline{\boldsymbol{\Theta_i}}$ and $\underline{\boldsymbol{D_i}}$ represent the rotation and displacement vectors for the i^{th} frame. $\underline{\boldsymbol{A}}$ shows the angular velocity format of $\boldsymbol{A}$. Finally v_{D_i} and v_{Θ_i} are the displacement and angular velocities of the i^{th} frame.

These conversions will provide us with the benefit of dealing with vectors which can be interpreted regardless of their previous states. The direction of each vector can be

calculated, and the new values are independent of the performer's physical features such as height.

$$\underline{\Theta_i} = \left(\theta_i^{x_1}, \theta_i^{y_1}, \theta_i^{z_1}, \ldots, \theta_i^{x_n}, \theta_i^{y_n}, \theta_i^{z_n}\right) - \left(\theta_{i+1}^{x_1}, \theta_{i+1}^{y_1}, \theta_{i+1}^{z_1}, \ldots, \theta_{i+1}^{x_n}, \theta_{i+1}^{y_n}, \theta_{i+1}^{z_n}\right) \tag{3.6}$$

$$\underline{D_i} = \left(d_i^x, d_i^y, d_i^z\right) - \left(d_{i+1}^x, d_{i+1}^y, d_{i+1}^z\right) \tag{3.7}$$

$$\underline{A} = \begin{bmatrix} \underline{D_1} & , & \underline{\Theta_1} \\ \underline{D_2} & , & \underline{\Theta_2} \\ \vdots & & \vdots \\ \underline{D_{m-1}} & , & \underline{\Theta_{m-1}} \end{bmatrix} \tag{3.8}$$

$$v_{D_i} = fr.\underline{D_i} \tag{3.9}$$

$$v_{\Theta_i} = fr.\underline{\Theta_i} \tag{3.10}$$

The derived relations describe the principles that hold true in the system of the motion capture data sets and will later be used in the training and test process of this research. In Equations 3.9 and 3.10, however, since the entire frame rates (*fr*) are equal and consistent throughout the research, the *fr* value simply acts as a scaling factor which can be ignored. Figures 3.10 to 3.15 illustrate the angular velocities in vector format of Figures 3.4 to 3.9 presented separately for each dimension.

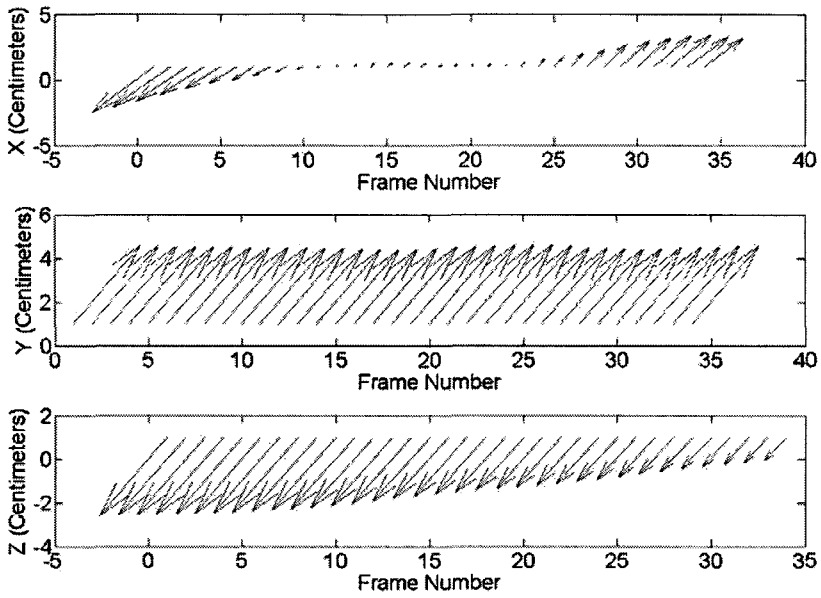

Figure 3.10. Hip marker data for masculine walk, angular velocity vector format

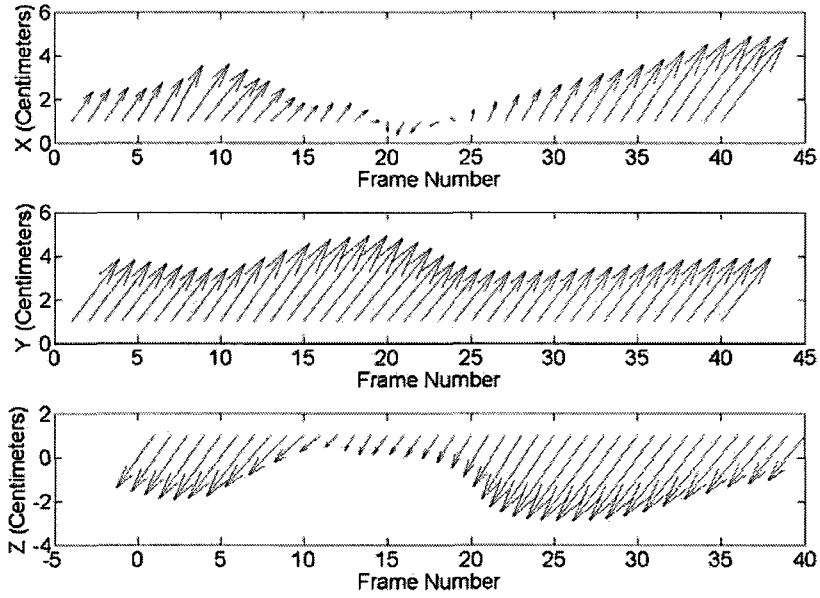

Figure 3.11. Hip marker data for masculine jump, angular velocity vector format

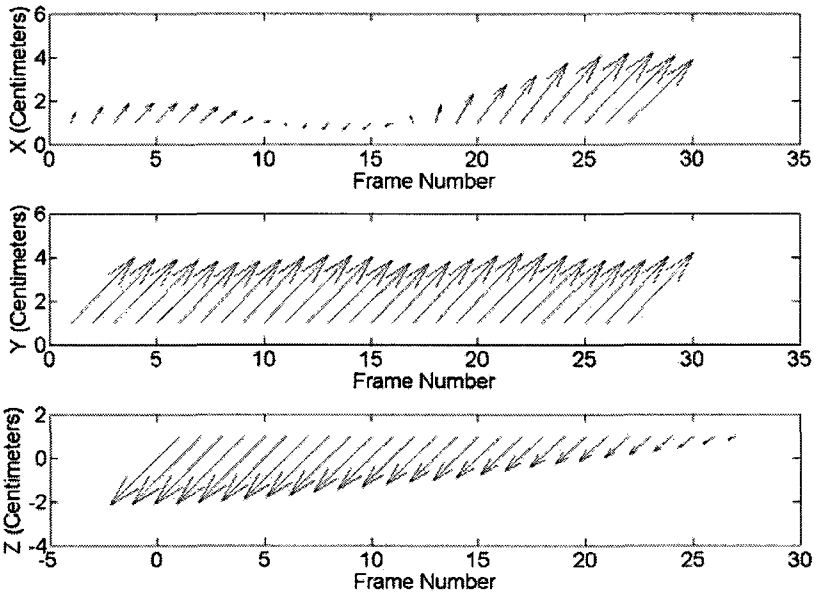

Figure 3.12. Hip marker data for masculine run, angular velocity vector format

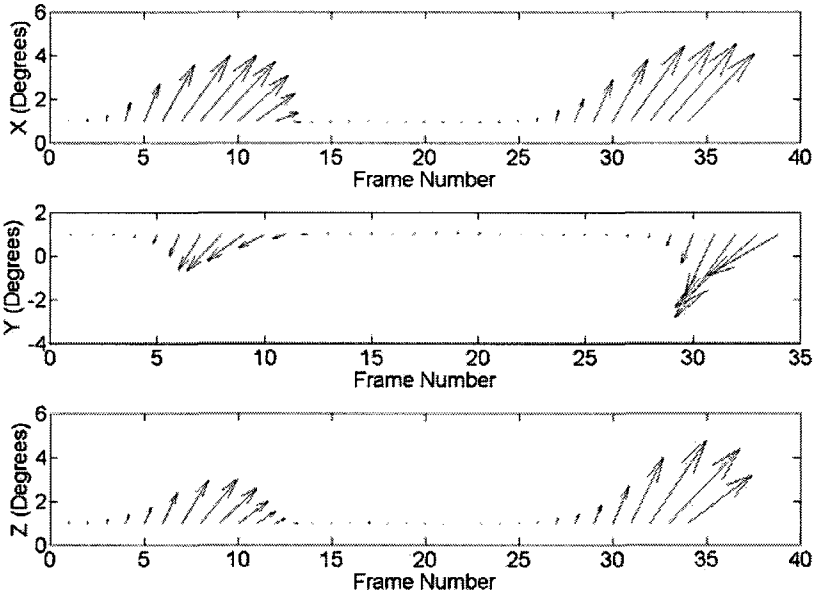

Figure 3.13. LeftLegRoll marker data for masculine walk, angular velocity vector format

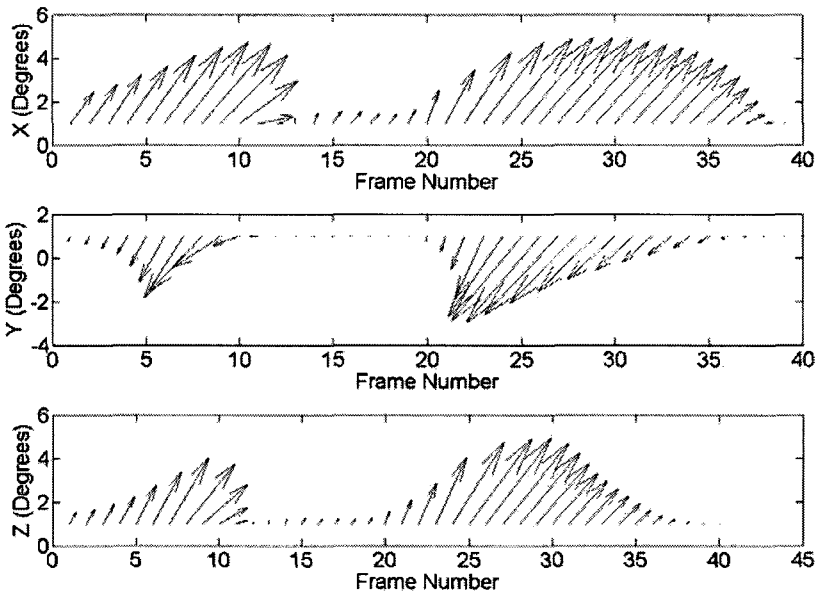

Figure 3.14. LeftLegRoll marker data for masculine jump, angular velocity vector format

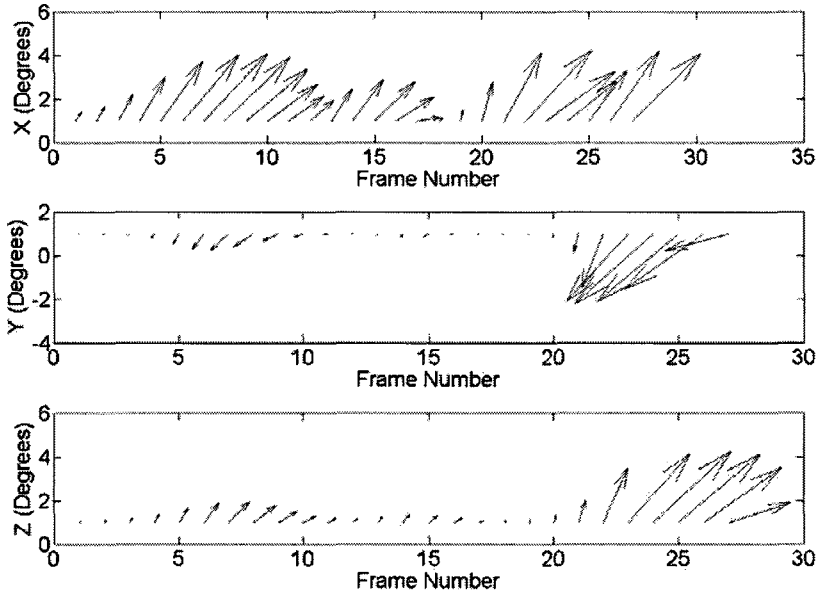

Figure 3.15. LeftLegRoll marker data for masculine run, angular velocity vector format

In Chapter 4 for the recognition process using nearest neighbour and hidden Markov models the dimensionality of the data is reduced to decrease the runtime. Also similar to [46] where entire actions have been recognized using the *head* alone, we have proposed that there is no need to use all the motion data for recognition as the task can be carried out using only some part of the body for instance just the head, or the lower part of the body (hip, legs and feet). We in this research have carried out quantization and then used the 7 markers of lower half of the body for recognition using hidden Markov models and used the scattered (non-quantized) data for the lower half of the body when using nearest neighbour classifier. The neural network technique however, employs the complete and unchanged data values (full body marker values, no quantization or velocity conversions).The preprocessing which is required to refine the data accordingly is described here.

To simplify the calculations the values were quantized by a radius of 3 centimetres/s and 3 degrees/s for $\underline{\boldsymbol{D}}$ and $\underline{\boldsymbol{\Theta}}$ values respectively. Each step based on its span is assigned a numeric value. For instance 0 is assigned to all the values within the span of 0.0 to 3.0 degrees/s. This is done for two reasons: 1) HMMs require specific output symbols and 2) Simplification of data by reducing the number of possible outputs. The 3 degrees/s range of the span was obtained by experimentally, where the mentioned range proved to be most efficient, having no significant negative impact on the accuracy while decreasing the number of output symbols thus decreasing the run-time and computational complexity.

The reason that only the 7 markers representing the lower half of the body were selected is as follows. From Equation 3.6 we can conclude Equation 3.11.

$$\underline{\theta_{i,j}} = \left(\theta_i^{x_j}, \theta_i^{y_j}, \theta_i^{z_j}\right) - \left(\theta_{i+1}^{x_j}, \theta_{i+1}^{y_j}, \theta_{i+1}^{z_j}\right) \tag{3.11}$$

Therefore the magnitude of $\underline{\theta_{i,j}}$ shown by $\left|\underline{\theta_{i,j}}\right|$ is calculated by Equation 3.12.

$$\left|\underline{\theta_{i,j}}\right| = \sqrt{\left(\theta_i^{x_j} - \theta_{i+1}^{x_j}\right)^2 + \left(\theta_i^{y_j} - \theta_{i+1}^{y_j}\right)^2 + \left(\theta_i^{z_j} - \theta_{i+1}^{z_j}\right)^2} \tag{3.12}$$

Based on Equation 3.12, the overall average marker velocity per frame for an entire sequence is provided by Equation 3.13 where N is the total number of frames and M is the number of markers.

$$\overline{\left|\underline{\Theta}\right|} = \frac{1}{N}\sum_{i=1}^{N-1}\sum_{j=1}^{M}\left|\underline{\theta_{i,j}}\right| \tag{3.13}$$

Calculating Equation 3.13 for both the lower half markers (limiting j to markers corresponding to the lower half body markers), the upper half marker (limiting j to markers corresponding to the upper half body markers), and the full body markers reveal that despite having less markers in the lower half of the body, for the action classes used in this research, roughly 60 percent of the movements of a whole human body is carried out by the lower half. This amount of information is enough to construct a system capable of recognizing actions based on the lower half only. This however is only true for the actions used in this research, and is clearly incorrect for some other action classes such as waving, punching, handshake, and etc where the significant portion of the action is carried out using the arms.

Due to high correlation between left arm and right leg, right arm and left leg, and central hip and upper body, not only omitting the upper body data will not affect the

recognition process, but will even stabilize the process by reducing the dimensionality. The reason is that in walking or running, the periodic swinging motion of the left arm is in phase with the right leg. The same situation applies for the right arm and left leg. There are extra movements in the arms while performing the mentioned actions, yet they are insignificant and more importantly, not determinant of the action class i.e. a walking sequence is still walking regardless of some arm and hand movements which might occur through the course of the action. As another result of working with the lower body data, the dimensions of the HMM is decreased, resulting in less run-time and higher accuracy.

3.4. Noise Reduction

As the last step for construction of the database, the issue of noise reduction is tackled. Through the motion capture sessions, there exist a variety of noise sources such as misplacement and movement of markers during the process, blind spot positioning of a marker for the cameras, and initial calibration of the system. In general however, most of the mentioned situations are avoidable and only a small amount of noise is present. To reduce the impact of the remaining noise, a smoothing filter in the form of a digital Low Pass Filter (LPF) is utilized in MATLAB and using the filter design toolbox. Such filters decrease the effect of high frequency noise thought to be present in the data set. The convolved LPF in time domain is illustrated by Figure 3.16.

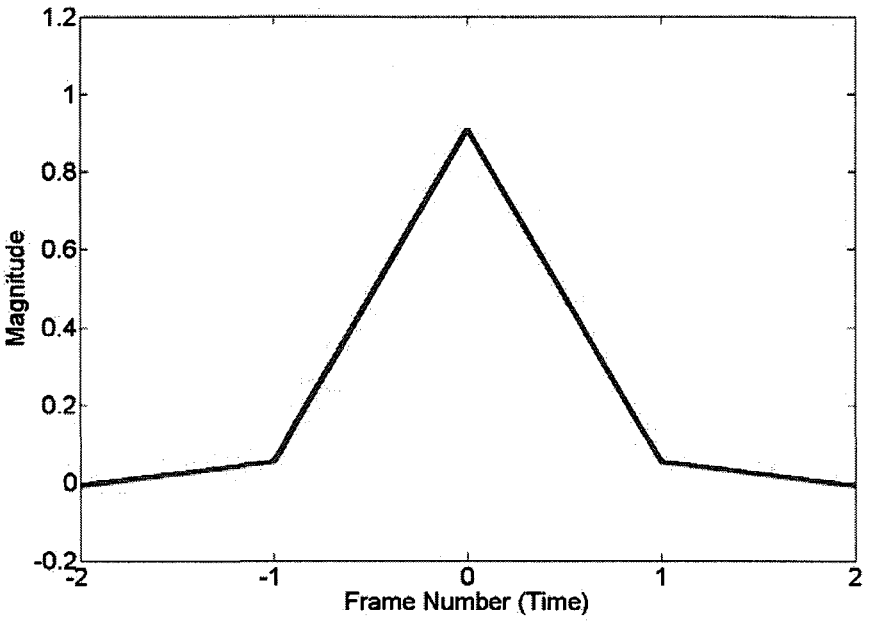

Figure 3.16. LPF in time domain

The effect of applying such filters is presented in Figure 3.17 where high frequency noise has been reduced or removed from the signal. This can especially be seen in frames 22 – 23.

The filter which has been used for noise reduction of data during the preprocessing is selected to be a weak filter. The reason for this preference is: 1) The noise elements are not too strong, 2) Low pass filters possess some drawbacks which have undesired effects on the data.

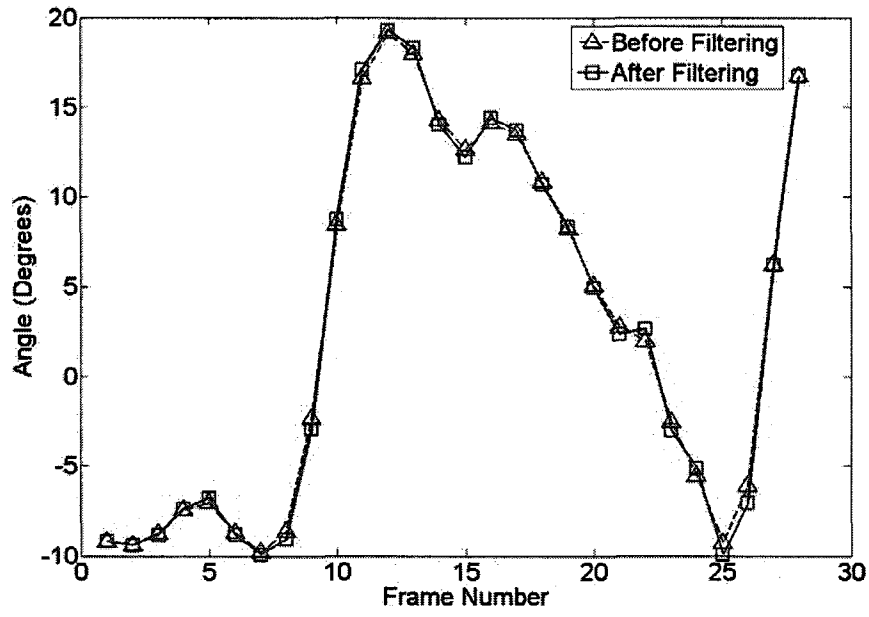

Figure 3.17. Effect of LPF on marker rotation signal

All filters contain specific drawbacks. In image processing for instance, low pass filters blur the image while high pass filters tend to sharpen the image. These effects may sometimes be desired results of the filtering process, yet in most cases, they are considered as unwanted side effects. When dealing with regular signals, low pass filters tend to introduce some phase shift to the signal. The more powerful the filter is designed, the greater the artefact will be. Thus it is important to design the filters in a way to maintain a balance between the amount of noise elimination and artefact introduced to the original signals. In this research, the window size and threshold of the filters were selected by means of experimentally. Once a filter was designed, the effects of the filter on the signal were carefully diagnosed not to have manipulated the signal significantly, yet minimizing the noise as much as possible. It is possible to calculate the phase shift for

further evaluation, yet the easier way is to re-animate the data through BVHacker (Appendix C). A significant phase shift is clearly visible through the motion of the character, usually appearing as what is called the sliding effect where a section of the body, feet for instance, are not in phase with the rest of the body and with the action being performed by the skeleton. By trial and error, we have tried to minimize the noise, while trying to introduce the least sliding artefact possible.

Chapter 4: Segmentation and Classification

4.1. Introduction

Many attempts have been made to create systems capable of classifying different actions using a variety of different classification tools [3]. While most research has focused on visual data [4 – 22, 36, 46] and features such as optical flow vectors, in this research we rely on optical motion capture data. To state in simple terms, the goal of this phase of the research is to employ motion capture data and classify different actions which the system has previously been trained with. Classifying different actions holds numerous applications in the fields of virtual environments, simulators, computer games, and sport commentary.

We break up this task into two subsections:

i) *Segmentation*, also referred to in some literature as *Locating an action*

ii) *Classification.*

The first section aims at automatically locating specific actions along the temporal axis in a sequential combination of different meaningful or meaningless actions. While most literature have ignored this section by focusing on individual actions or manually segmenting the sequences [4, 5, 23, 24, 40 – 45], we have provided the means for automatic segmentation of specific actions from a sequence of different actions. The second section aims at classifying the segmented action into one of the previously defined classes of *walking*, *running*, and *jumping.*

A very important feature of our proposed method is the fact that not only have we classified the class of action successfully, but we have also provided the means for defining and classifying the secondary theme associated with the action class. The secondary theme relates to one of the three classes of *gender*, *age*, and *energy*.

We have used three very popular tools in the field of pattern recognition: Nearest Neighbour search, Hidden Markov Models, and Artificial Neural Networks. Each tool will be described in this chapter and the results are analysed in Chapter 7.

4.2. Nearest Neighbour Classifier

By definition, nearest neighbour search is a classification technique that uses some kind of distance –in most cases Euclidean distance in the feature space to determine and classify the test samples based on the closest trained samples. In terms of functionality of this classifier, it is a binary search algorithm which is also called similarity search [68]. This classifier has previously been used in some literature for human action recognition and classification [6, 9, 36, 46].

In this research we have employed two different distance measures for nearest neighbour classifier and compared the results in Chapter 7. The first method is by means of absolute difference and the second is the very popular Euclidean distance.

For this method, a training action sequence, for instance walk, is available. A number of unknown sequences are available, each containing one or more meaningful or meaningless actions. The goal is to compare the training action sequence to all the test sequences, and recognize the most similar (nearest) sequence. Using this process we can carry out both segmentation and classification tasks. First the nearest frame of the test sample to all the starting frames of all the training samples is located. Following locating starting frame of the action, the subsequent frames are compared to the respective frames of the training samples for classification.

We name the reference action $R(\underline{\theta}_1:\underline{\theta}_v)$ where $R^i(\underline{\theta}_1:\underline{\theta}_v)$ represents the i^{th} frame of R from 1 to n and containing v arrays, each representing a coordinate of a marker. $\underline{\theta}$ is the angular velocity value obtained in Chapter 3 Section 3.The j^{th} test sample is denoted by $T_j(\underline{\theta}_1:\underline{\theta}_v)$ where $T^i_j(\underline{\theta}_1:\underline{\theta}_v)$ denotes the i^{th} frame. In this algorithm the search initiates by the

first frame. $R^1(\underline{\theta}_1:\underline{\theta}_v)$ is compared to all the frames of all the test samples, meaning if there are m test samples and the last sample contains nm frames, $R^1(\underline{\theta}_1:\underline{\theta}_v)$ is compared to $T^1{}_1(\underline{\theta}_1:\underline{\theta}_v)$ to $T^{nm}{}_m(\underline{\theta}_1:\underline{\theta}_v)$. For each test sample, the frame that shows to be the nearest neighbour (least distance) is selected as the starting point (for segmentation). The distance is measured by the absolute value of the subtraction of the two rows. All the arrays in the resulting row are summed. Once the initial frames of all the test samples are located, the following frames are subtracted respectively. The difference measure is calculated by Equation 4.1 where n is the n^{th} frame of $T(\underline{\theta}_1:\underline{\theta}_v)$ where the minimum difference for the first frame of $R(\underline{\theta}_1:\underline{\theta}_v)$ occurs.

$$e_m = \sum_{\theta=1}^{\theta=v} \sum_{i=1}^{length(R)} \left(\left| R^i(\theta) - T^{n+i-1}{}_m(\theta) \right| \right) \tag{4.1}$$

The test sample which then returns the minimum e is selected as the sequence containing an action of the same class of $R(\underline{\theta}_1:\underline{\theta}_v)$ for both primary and secondary themes.

The same process is applied when Euclidean distance is employed as the indication of distance. Instead of Equation 4.1 however, Equation 4.2 will be calculated where j represents the markers in each frame and similar to Equation 4.1 i represents the frames. This equation measures a Pseudo-Euclidean distance for our feature space.

$$e_m = \sum_{j=1}^{j=v} \sum_{i=1}^{length(R)} \sqrt{\left(R\left(\theta_i^{x_j}\right) - T\left(\theta_i^{x_j}\right)\right)^2 + \left(R\left(\theta_i^{y_j}\right) - T\left(\theta_i^{y_j}\right)\right)^2 + \left(R\left(\theta_i^{z_j}\right) - T\left(\theta_i^{z_j}\right)\right)^2} \tag{4.2}$$

In both Equation 4.1 and Equation 4.2 the R which returns the minimum e_m determines the class of action and style which the segmented section of T belongs to. Figure 4.1 illustrates a scheme of nearest neighbour search.

The numerical and experimental results of this technique are presented in Chapter 7, along with a discussion on the advantages and drawbacks, yet an obvious disadvantage of this technique is that temporal misalignment of different stages of an action can increase the error and eventually result in false classification of the actions.

As an extension to this procedure, all the test sequences can be compared to all the reference sequences. The minimum difference leads the first action to be classified. Then the classified test sequence is set aside and the procedure is carried out again until all the test samples have been classified. The results are described in more details in Chapter 7.

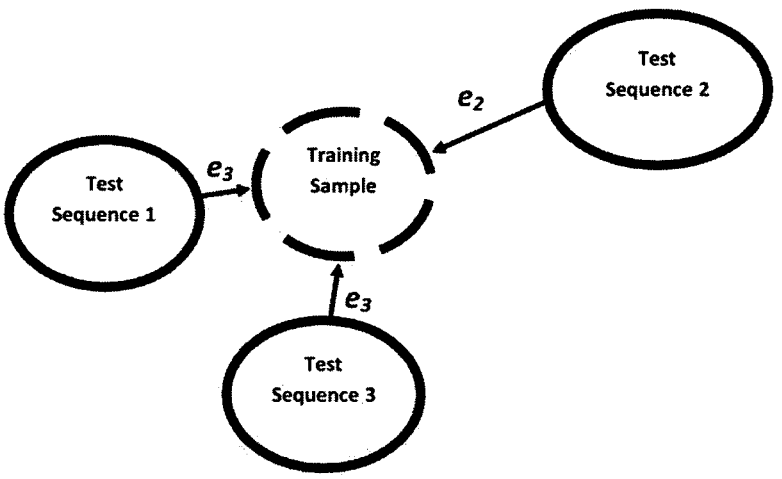

Figure 4.1. Nearest neighbour search

4.3. Hidden Markov Models

Hidden Markov Models (HMM) are statistical models often considered as the simplest type of Bayesian networks [69]. In HMMs various states are defined which are *hidden* and not visible while the outputs which depend on the states are visible, thus the overall model is considered to be visible. The states possess probability distributions over the outputs [70]. The Viterbi algorithm is one of the most popular algorithms for configuring hidden Markov models. This algorithm aims at determining the most likely sequence of hidden states within the model. First order models are usually selected for this algorithm as it satisfies the assumptions necessary to utilize this tool [71]. Since the concern is not the orientation of the hidden states and the path, the Baum-Welch algorithm is employed. The Baum-Welch method is also known as a type of Generalized Expectation Maximization (GEM or EM) algorithm, performing as a forward-backward algorithm for calculating the most likelihood of observing specific output symbols in a sequence [72]. For implementation, the Kevin Murphy HMM toolkit for MATLAB is employed (http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html).

To recognize different actions along with the associated styles, performed by different performers, hidden Markov models were employed as one of the tools. For a network with n hidden states $\{s_1,s_2,...,s_n\}$, the transition probability is $P(s_j(t+1)|s_i(t))$. Transition property is defined as the probability of the HMM being in state s_i at $(t+1)$ if it has been in state s_j at (t). Another definition in HMM is emission probability, where it is the probability of the HMM producing a certain symbol (observation) at a certain state

[73]. Providing the network with some initial probabilities, the probability of observing a sequence of symbols with a length of m is presented by Equation 4.3.

$$P(Y)=\sum_{X} P(Y \mid X).P(X) \tag{4.3}$$

Where $Y = y(0), y(1),..., y(m-1)$ is the observed sequence.

Figure 4.2 shows a basic form of an HMM network.

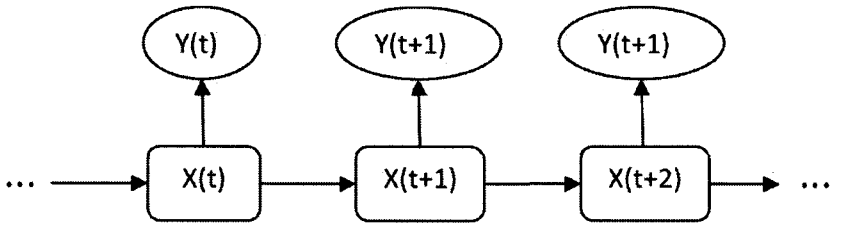

Figure 4.2. Basic HMM network

For each class of action and related variations, a different HMM is created which calculates the total emission probability for generating the sequence of feature vectors corresponding to each action. The test data will be employed to the created networks according to the *Segmentation and Classification Algorithm* presented later in Chapter 4 Section 3.2., and the HMM resulting in the most likelihood will determine the class of the test sample.

After setting up the HMM networks, the number of hidden states, as well as iterations – in order to reach maximum learning without occurrence of over learning,

must be determined. Figure 4.3 shows the learning accuracy curve for different number of states in a 45 iteration process. The highest accuracy is obtained for 17, 21, 25, 26, and 27 states. Yet the difference between a 17 state and a 27 state hidden Markov model network in this research is insignificant, and therefore to minimize the computations and runtime, 17 states is assigned to the network. In Figure 4.4 the most efficient number of learning iterations is presented, where 35 iterations show that the log likelihood does not demonstrate significant improvement beyond this point. The completed HMM network is able to recognize the three classes of walking, jumping and running for which it had been trained for, with very high accuracy.

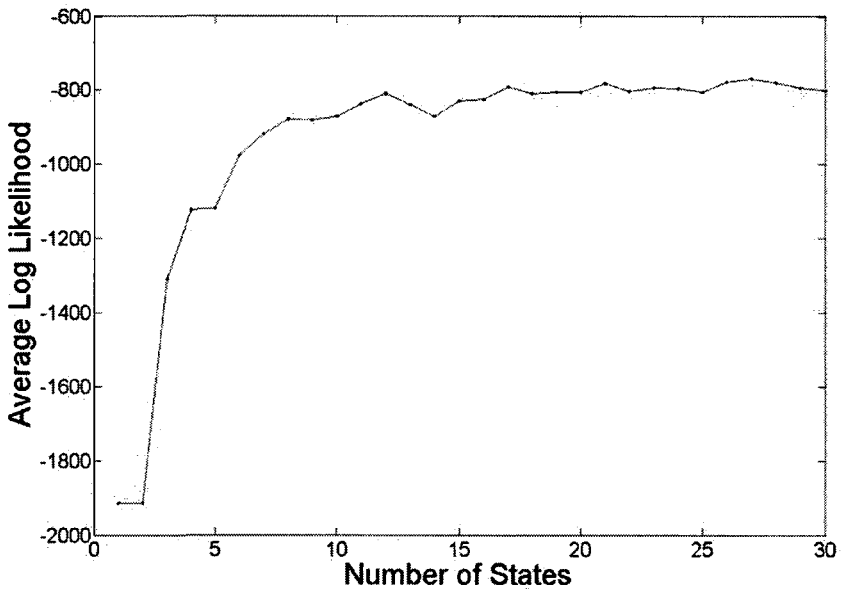

Figure 4.3. Average Log Likelihood for Different Number of States

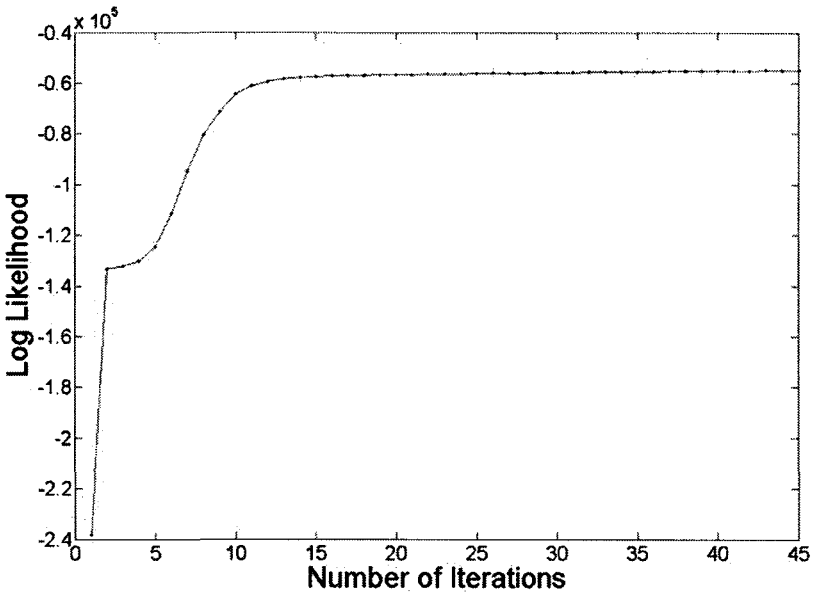

Figure 4.4. Learning Iterations for 17 states

4.3.1. Segmentation and Classification Algorithm

Once a basic HMM network is up and running, the following algorithm is proposed to recognize each movement and its related style variations. In this algorithm a fine tuning method has been employed to recognize the size of a movement cycle. Not including this section in the algorithm would result in erroneous results, since each HMM is trained using a complete cycle of each action having left intact no extra frames. The HMMs used in this research are very sensitive to irrelevant and extra frames of data. A hierarchy of hidden Markov models is presented to classify each class of action and their multiple variations of style, gender, age, and mood. Figure 4.5 shows the overall appearance and functionality of the recognition process.

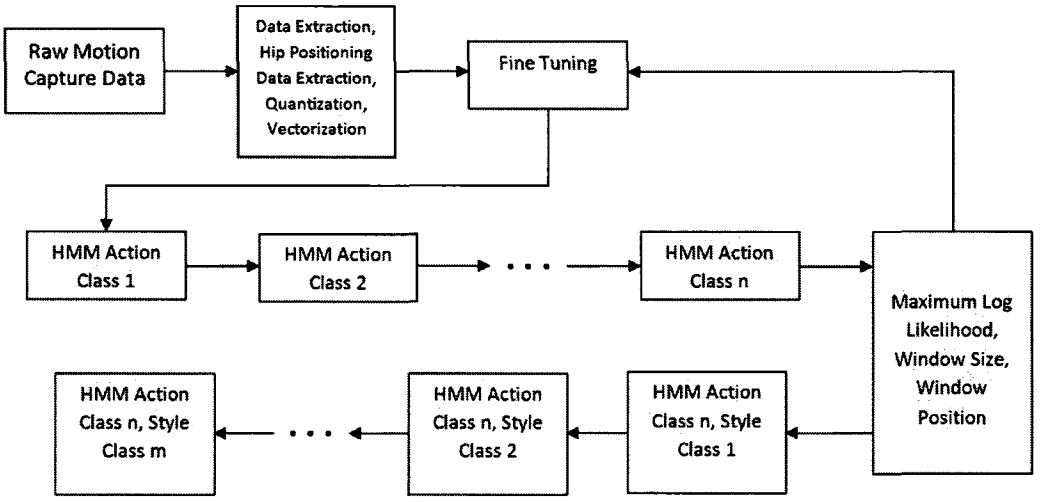

Figure 4.5. Recognition Process

From the raw motion capture data first the seven lower body markers are extracted, filtered, converted into angular velocities, and quantized according to Chapter 3. In this algorithm an HMM network is trained using the data for each class of action. HMM sub-networks for each style of action are also constructed, i.e. old, young, feminine, masculine, happy, energetic, and tired. The test data are provided to all the primary networks, classifying the action into major classes of walking, running, and jumping.

The test data is basically an unknown action with an undefined length which might be composed of various actions. As mentioned earlier, null frames in the beginning or the end of an action would significantly affect the outcome, therefore a fine-tuning method is proposed. A window with the size of the maximum number of frames used in the training process is placed at the beginning of the test data, decrementing in size, and with each decrementation, the frames within the window are provided to the first layer of HMM

networks and the log-likelihood is returned. When the window becomes as small as the shortest action used to train the HMM the decrementation stops. The window is shifted forward by one frame, the size of the window is reset to the size of the length of the largest training sample, and the process repeats. The procedure continues until the window reaches the last frame of the sequence. In each step, prior to shifting the position of the window, the log-likelihood and the location and size of the window on the action sequence are returned and saved. Once the exhaustive search is complete, the saved results are compared. The greatest log-likelihood is determined and the respective window size and position point out the location and size of the action. This whole process repeats for all the HMMs trained by the 3 action classes. The HMM which results in a higher likelihood determines the action class. The same process then repeats for classification of secondary themes.

4.4. Artificial Neural Networks

Artificial Neural Networks (ANN) are practical tools in the field of pattern recognition which are inspired by the functionality of biological neural networks for processing of data. They consist of interconnected neurons also referred to as nodes. They are structures which can model complex models and relate the inputs and outputs of a system [74]. The ability for these networks to adapt to changes in inputs and outputs as well as simple expansion and contraction in terms of number of inputs, outputs, hidden layers, and hidden nodes are some of the advantages of ANNs over other tools. Also the

low run time successive to the lengthy procedure of training the network, provides fast response for recognition and other applications.

In this research the efficient and practical feed-forward Multi Layer Perceptron (MLP) neural network is employed. MLP employs multiple layers and maps inputs and outputs, resulting in non-linear classification where required. The learning algorithm of back-propagation is adopted. Back-propagation is considered a supervised method of learning which utilizes *activation functions*.

The database consists of fourteen marker information columns in addition to the hip positioning placement data adding up to fifteen sets of columns for the entire body. Three sets of data representing each coordinate of each marker will add up to a data base of 45 columns. Each coordinate of each marker is a separate input for the neural network resulting in a 45 neuron input layer. Since re-synthesis of data is also projected to take place, the output layer would also need the same number of neurons, one neuron for each coordinate of each marker. After a number of different trials, the best configuration for the network is determined. There are some different proposed methods for setting the optimal number of hidden nodes and layers. Different number of estimates such as *somewhere between the number of inputs and outputs* [75] to *no more than twice the number of inputs* [76, 77] have been suggested. As for the number of hidden layers, especially for large number of inputs, *two hidden layers* have been proposed for more accurate results with respect to one hidden layer [78]. Some estimation methods based on the number of training samples have also been suggested [79], yet in our case, due to uncertainty in the number of frames and actions they were overlooked. Considering the

mentioned facts and after a number of different trails, the final architecture of, 45, 45 to 90, 45 to 90, and 45 neurones was adopted for layers one to four, respectively.

The approach to training the neural network is based on learning the $(i+1)^{th}$ frame based on the previous frame, i.e. the i^{th} frame. For K classes of actions and V classes of style variations per each action, K x V + K neural networks have been created and connected in what we call a neural forest as shown in Figure 4.6.

The inputs and targets for training the constructed neural forest are constructed based on $\boldsymbol{A}$. Where the i^{th} row of $\boldsymbol{A}$ for the major action class j is represented by A_i^j, the input matrix is displayed in Equation 4.4 and represented by φ^j.

$$\left(\varphi^j\right)^T \equiv \left[A_1^j \; A_2^j \cdots A_{m-1}^j\right] \tag{4.4}$$

Similarly the target matrix is constructed and represented by λ^j as shown in Equation 4.5.

$$\left(\lambda^j\right)^T \equiv \left[A_2^j \; A_3^j \cdots A_m^j\right] \tag{4.5}$$

The benefit of applying this structure is that different actions with different lengths can be analyzed and recognized. There is no need to warp the actions to produce same sized actions.

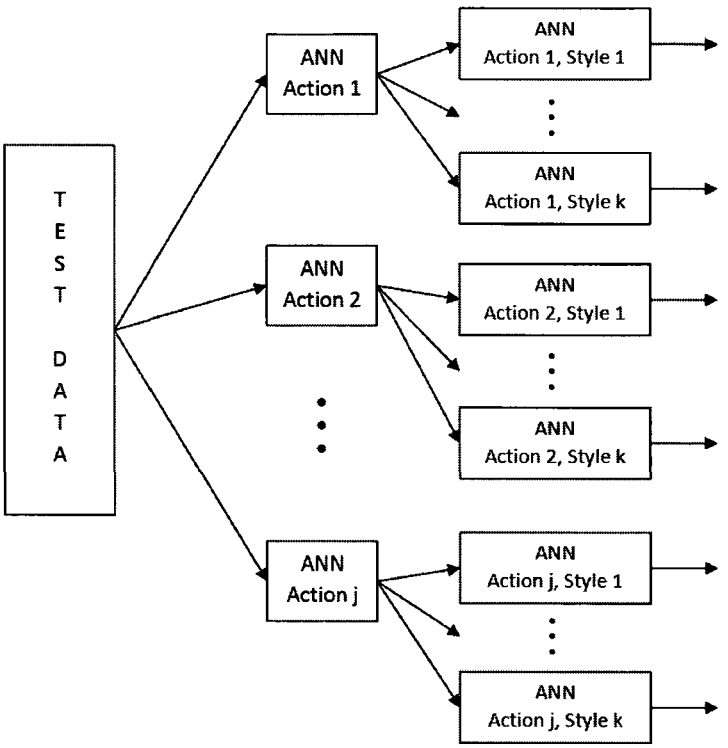

Figure 4.6. Neural Forest for Recognition and Synthesis

From 7 to 10 similar actions have been chosen for training each of the networks. The first layer of the forest is trained via all the possible secondary variations of an action, for instance masculine, feminine, young, old, energetic, and tired walk for training the walking network is used. In the second layer of the forest, a different network for each secondary style action is trained, i.e. one network for masculine walk, one network for feminine walk, and etc. The same process repeats for running and jumping. The overall input and target matrices are presented by Equation 4.6 and Equation 4.7 respectively where j is the last action to be used for training.

$$\Phi^T \equiv \begin{bmatrix} \varphi^1 & \varphi^2 & \cdots & \varphi^j \end{bmatrix} \tag{4.6}$$

$$\Gamma^T \equiv \begin{bmatrix} \lambda^1 & \lambda^2 & \cdots & \lambda^j \end{bmatrix} \tag{4.7}$$

An in depth investigation reveals that the neural networks are trained to learn each frame of an action, knowing the configuration of the previous frame. For the first stage of the neural forest, each major action class network learns all possibilities of a consecutive frame for a known frame, yet the configuration of a starting frame is different for different secondary themes of actions. A male actor stands differently from a female actor when they are to perform a straight walk, therefore the network can distinguish between the two, resulting in more sophisticated training.

Figure 4.7 provides an insight of the training and classification procedure. To measure the accuracy and significance of the system, Mean Square Error (MSE) for both the training data and new test data is utilized. As shown in Figure 4.7, the calculated output of the network is called $\theta^j{}_O[t+1]$ where the desired output is $\theta^j[t+1]$. Theta is the angular values of all coordinates of all markers of frame $[t+1]$ for action j when the input is the angular values of frame $[t]$ of the same action. The least mean square error of the network output with respect to the desired output will determine the class of actions.

The mean square error which is usually in the form of $\frac{1}{m}\sum_{j=1}^{m}\left(f_j(t)-f_{Oj}(t)\right)^2$ is slightly modified to represent 3D feature data in Equation 4.8 where M is the number of frames and N is the number of markers.

$$MSE = \frac{1}{M}\frac{1}{N}\sum_{j=1}^{M}\sum_{i=1}^{N}\left(\sqrt{\left(\theta_j^{x_i}-\theta_{Oj}^{x_i}\right)^2+\left(\theta_j^{y_i}-\theta_{Oj}^{y_i}\right)^2+\left(\theta_j^{z_i}-\theta_{Oj}^{z_i}\right)^2}\right)^2 \tag{4.8}$$

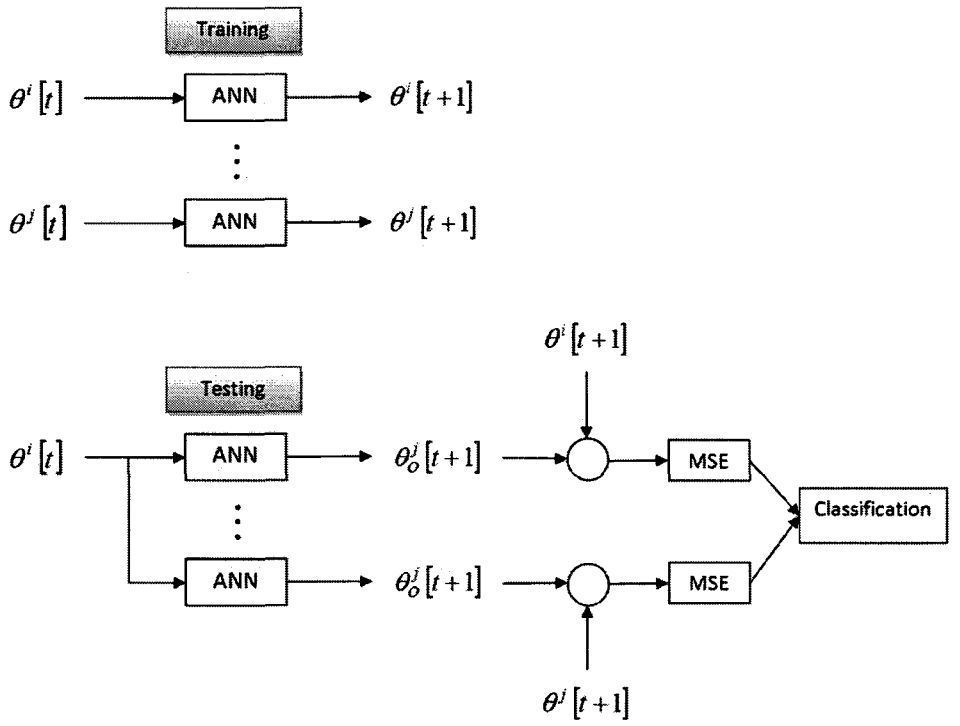

Figure 4.7. The training and classification procedure

Since in Equation 4.8, N is constant for all actions, it can be omitted from the equation. Also the equation can be simplified as shown in Equation 4.9.

$$MSE = \frac{1}{M}\sum_{j=1}^{M\times N}\left(\theta_j - \theta_{Oj}\right)^2 \tag{4.9}$$

Equation 4.9 is calculated for all M frames and a different MSE is calculated for each different network. The network that produces the least MSE will present the class of the action, for instance if the network trained with walking data performed the least MSE, it can be concluded that the tested action is a type of walking. The same process repeats for action style variations such as feminine, masculine, energetic, tired, old, and young.

4.4.2. Resilient Neural Networks

For optimization and updating the weights in neural networks, different techniques are proposed and employed. As an alternative to the Levenberg-Marquardt back-propagation training procedure which is quite popular, the very high-speed [80, 81] Resilient back-propagation (RPROP) training technique was used. From [82, 83] we can observe that RPROP is more suitable for 3D image, 3D object, and most likely in our case, 3D motion analysis. The RPROP technique which was proposed in 1993 by Riedmiller and Braum [81], holds several advantages for this research such as fast convergence and the ability to escape local minima having gone through sufficient learning epochs. The main difference between this learning scheme and other more common ones is that the adaptation of RPROP is not affected by the magnitude of the gradient and only takes into account the behaviour of the sign of the gradient. Although this property might be considered a setback for applications where more adaptability is required, in this research, it would be an asset due to uncertainty of human actions even when carried out by the same actor. In RPROP, only the sign of the gradient is considered for determining the manner for weight updates as opposed to other methods where the size of the gradient is a determining factor and this property opens more room for more uncertainty in the data.

In each stage of learning, the weight for neuron j to neuron i is updated by the amount of $\Delta w_{ji}(k)$ as shown in Equation 4.10 where $A_{ji}(k)$ is the update value and $E(k)$ is the error function [80].

$$\Delta w_{ji}(k)=\begin{cases}-A_{ji}(k) & if\ \frac{\partial E}{\partial w_{ji}}(k)>0\\ +A_{ji}(k) & if\ \frac{\partial E}{\partial w_{ji}}(k)<0\\ 0 & else\end{cases} \tag{4.10}$$

Therefore we can conclude Equation 4.11 where η is the increase factor and μ is the decrease factor, and $0<\mu<1<\eta$ holds true [80].

$$A_{ji}=\begin{cases}\eta\, A_{ji}(k-1) & if\ \frac{\partial E}{\partial w_{ji}}(k)\times\frac{\partial E}{\partial w_{ji}}(k-1)>0\\ \mu\, A_{ji}(k-1) & if\ \frac{\partial E}{\partial w_{ji}}(k)\times\frac{\partial E}{\partial w_{ji}}(k-1)<0\\ A_{ji}(k-1) & else\end{cases} \tag{4.11}$$

When dealing with human actions, during a walking cycle for instance, the actor may unintentionally take a faster step, resulting in larger angle derivatives in feature space. Therefore, this type of algorithm where the gradient magnitude influence on the weight change is eliminated, is more suitable for action recognition purposes based on the fact that RPROP is suitable for data with an amount of uncertainty. Having used the typical gradient based back-propagation techniques would have resulted in larger weight changes in the network which are inaccurate. Using this method, however, results in the system learning the pattern of the movement and ignoring some uncertainty in human actions.

Chapter 5: Temporal Alignment

5.1. Introduction

In instances where various action sequences of the same class are to be compared especially for feature selection and comparison, the actions must take place such that specific meaningful movements happen at the same time. For instance, when two takes of a jumping scene are available and subject to manipulation based on one another, the signals of one sequence must be aligned with the signals of the other sequence such that they both initiate the action at the same time, they both depart from the ground at the same time, they reach maximum height at the same time, and finally they return to the ground and finish the action at the same time. Thus an important factor for analysis and synthesis of human motion data is temporal alignment of the action sequences. Temporal alignment is critical for one-to-one correspondence of time snippets through the course of an action.

To accomplish temporal alignment, some important factors must be taken into account:

- Similar beginning of an action
- Alignment of specific (critical) features during an action
- Similar ending of an action
- Equal temporal length of an action

These aspects are not essential for segmentation and classification of actions since the proposed segmentation/classification techniques by means of HMM and ANN are capable of taking such misalignments into account. For classification using nearest neighbour search, however, temporal alignment can and does result in more accurate classification.

For synthesis of motion data and transformation of secondary themes, we have proposed a model which is capable of describing human motion, taking into account primary and secondary themes. To be able to use the model for secondary theme transformation, the action sequences are required to be temporally aligned.

Each of the four aspects of temporal alignment which were mentioned earlier must be taken into account. For similar starting and ending of action sequences of the same primary class, the feature selection is carried out manually where action sequences have been captured and segmented such that they start from a similar pose and end in similar poses as well. For the other two aspects of *alignment of specific features during an action* and *equal temporal length of an action* piecewise time warping is suggested. Also a method of alignment in frequency domain by means of frequency compensation is

experimented which as described in Chapter 5 Section 2 is not constructive as it causes a significant amount of error.

5.2. Fourier Phase Elimination

For temporal alignment of action sequences used in style transformations, since as mentioned earlier all the action sequences are manually segmented, it is only necessary to focus on manipulating the actions such that they end up with the same number of frames, and specific motion features are aligned.

The first task is carried out by means of simple stretching or compressing of the motion sequences. For any number of given motion data, the average frame number of all the sequences is first calculated. Then each individual motion matrix is stretched or compressed to reach the specified duration (frame number).

Stretching is accomplished by uniformly distributing the excessive number of required frames as empty frames among the shorter data set, extending it to the required length. The empty rows are then filled by linear interpolation (mean) of the prior and the consecutive row. As an example, an original marker signal is presented by Figure 5.1 which illustrates the data for the x^{th} coordinate of the marker placed on the right foot and the outcome of stretching the same signal is illustrated in Figure 5.2. Compressing a sequence is accomplished by simply removing the excessive number of rows, equally distributed throughout the original data set. Figure 5.3 shows the compressing process.

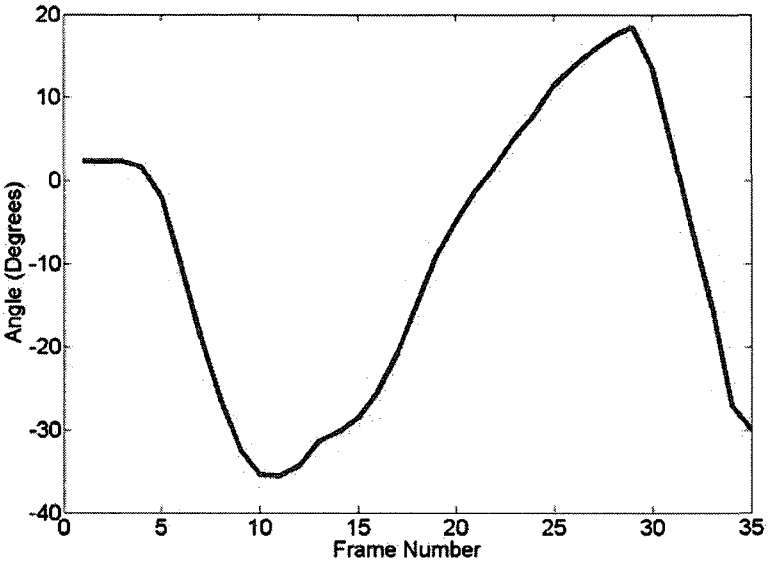

Figure 5.1. Original data signal

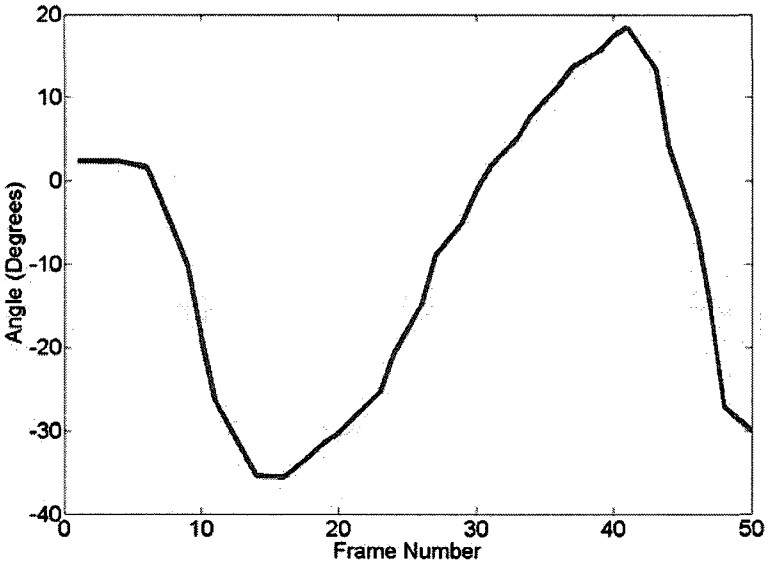

Figure 5.2. Stretched signal

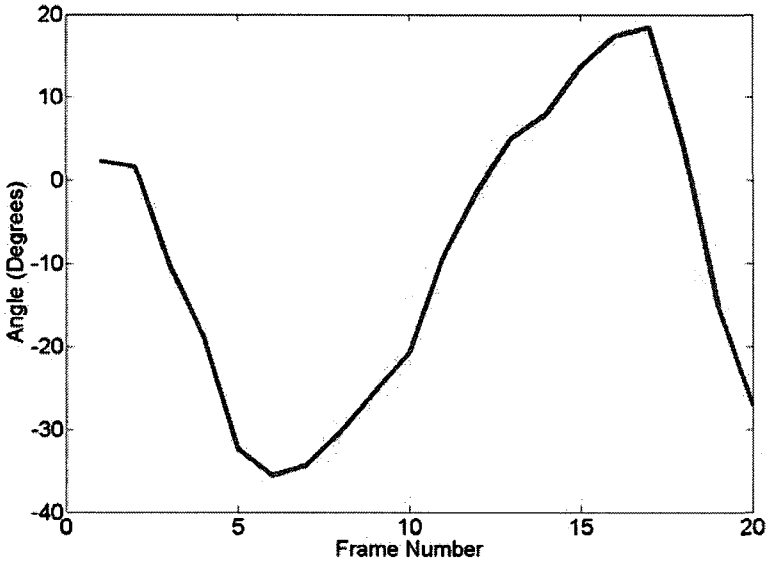

Figure 5.3. Compressed signal

Since the stretching procedure takes place in a linear fashion, the smooth curves which were apparent in the original signal are distorted in some instances. To fix this issue the low pass filter discussed in section 4.4 is employed after scaling, yet with a bit higher power by increasing the weight of the filter elements. Figures 5.4 and 5.5 illustrate the effect of noise reduction using low pass filters. This filter once convolved with the output signal, reduces the presence of break points and sharp edges. Meanwhile, the same drawbacks mentioned in Chapter 3 Section 4 apply.

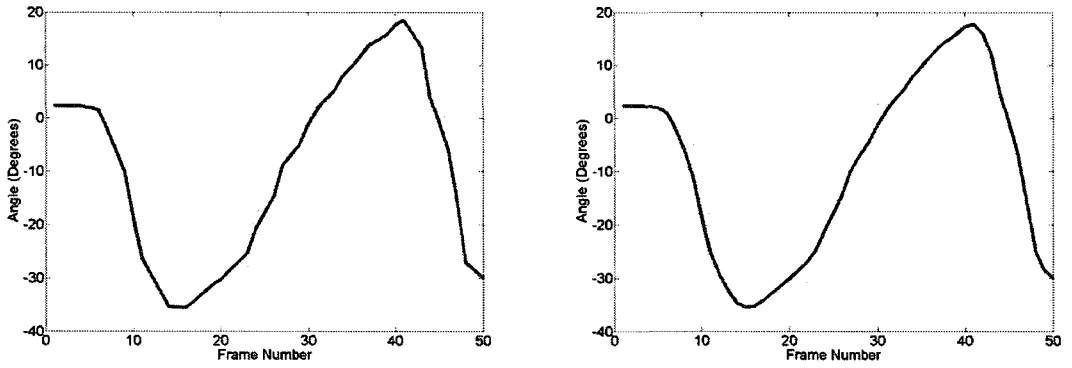

Figure 5.4. Stretched signal before (left) and after (right) filtering

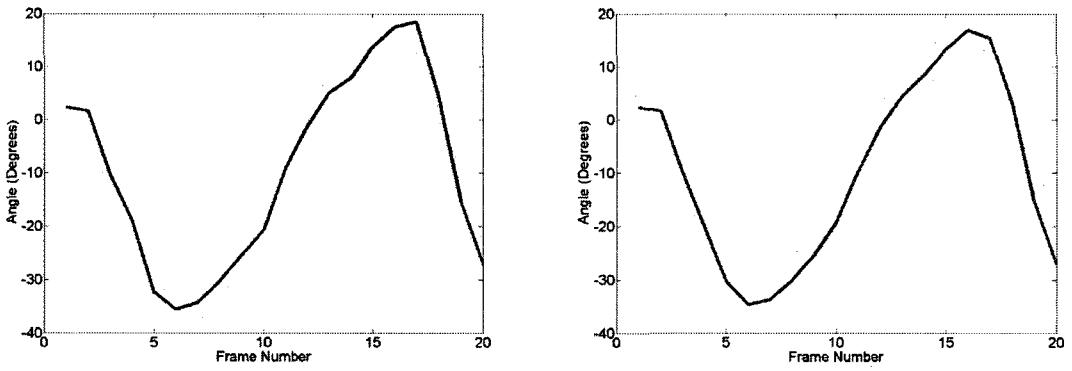

Figure 5.5. Compressed signal before (left) and after (right) filtering

The second step in temporal alignment is aimed at aligning specific motion features. Each column in a motion data matrix is considered as a discrete time signal. As each marker occupies three columns (one for each coordinate), we can either assume a 4D signal (x,y,z,t) or three 2D signals (x-t, y-t, z-t). For simplification in calculations, the latter assumption is used. It is known that any given signal in time possesses two

features: magnitude and phase. Figure 5.6 to 5.8 illustrate the magnitude and phase of the three coordinates of the right foot marker.

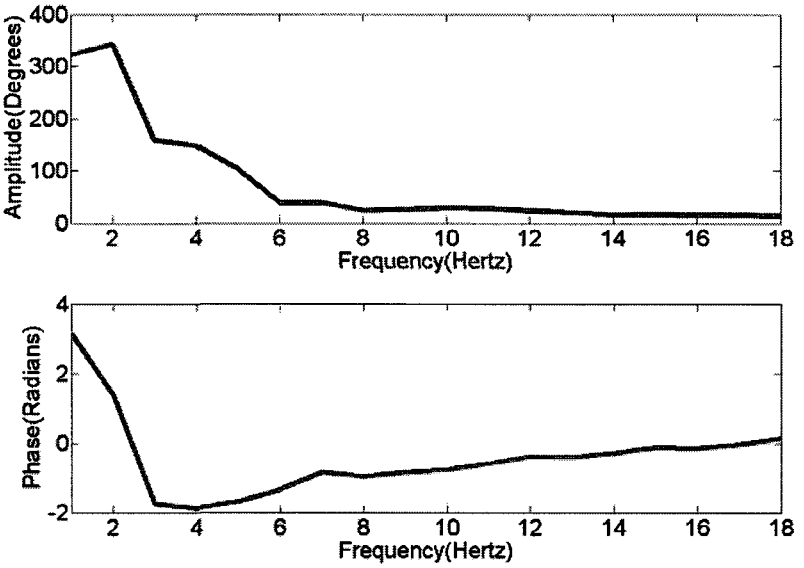

Figure 5.6. Frequency domain, x coordinate right foot marker

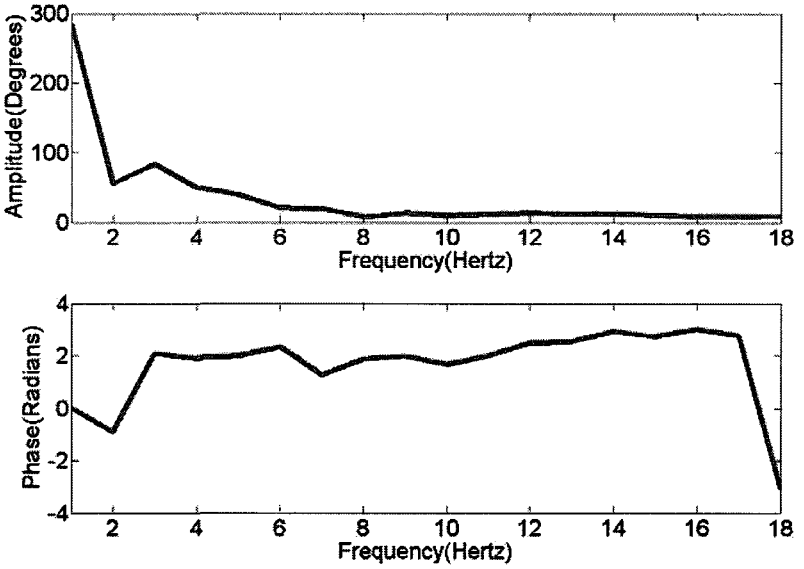

Figure 5.7. Frequency domain, y coordinate right foot marker

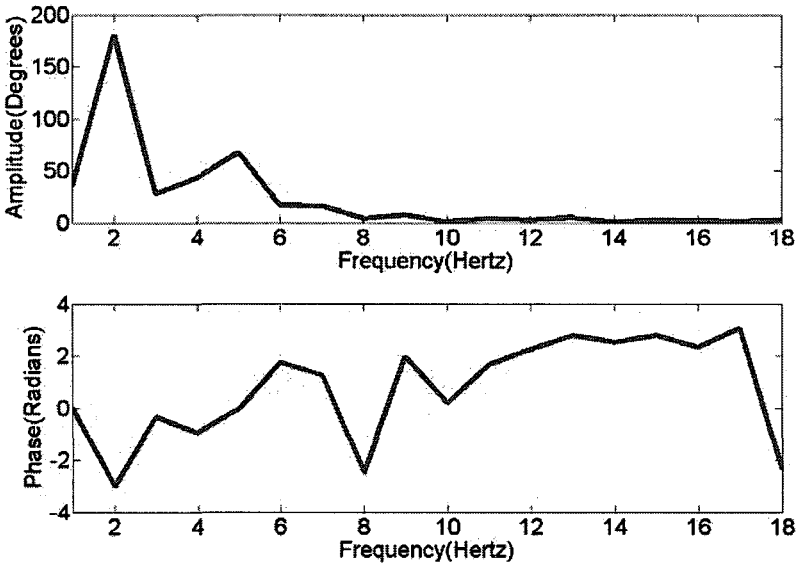

Figure 5.8. Frequency domain, z coordinate right foot marker

It may be possible that by equalizing the phase of the corresponding signals (respective coordinates of same markers), or by some other phase manipulation procedure the feature alignment be accomplished. In practice, however, two problems exist:

- Aliasing occurs
- Changing the phase of signals, only shifts the signal in time

Aliasing is an artefact in signal processing which will result in faulty reconstruction of signal, meaning that the inverse Fourier transform of the signal in frequency domain is not representative of the original signal. This happens when the signal frequency is less than half of the sampling frequency presented by the Nyquist rate Equation 5.1.

$$f_s/2 > f \tag{5.1}$$

This happens through the sampling process during the capture session. Some of the information required to reconstruct the time domain signal from the frequency domain is lost which results in an incorrect reconstructed signal. Figure 5.9 illustrates the reconstructed signal and the negative effect of aliasing.

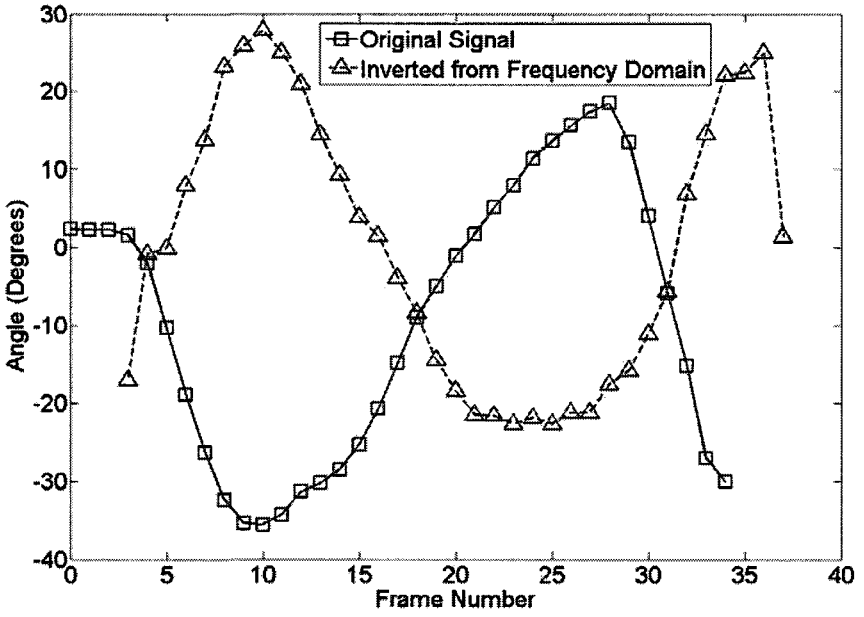

Figure 5.9. Occurrence of aliasing when reconstructing the signal

The second issue is that manipulating the phase of the signal only results in a time shift and does not aid in the alignment of the critical features since shifting the signals with respect to one another results in misalignment in the starting and ending instances of the actions signals.

Based on the mentioned facts, the temporal alignment of specific features cannot be accomplished by phase manipulation, and must be carried out in time domain. In the next section, the following two problems will be addressed:

- Locating the critical features
- Aligning the features while stretching/compressing the signals

5.3. Piecewise Time Warping

Based on the earlier discussion in the previous section, two problems remain unsolved. The first is to locate the critical features which will be used to align the signals based on which the most correspondence between the signals is aimed to be accomplished. The second is to align the features. Resizing (stretch or compress) of the signals was achieved in the previous section, yet our proposition for alignment of the features is to carry out the *resizing* and *alignment* simultaneously. Here we propose piecewise time warping which aims at conducting both tasks together.

The captured motion data matrix $\boldsymbol{A}$ can be of various lengths. While some actions have been performed faster or slower due to actor preferences, some actions are by nature faster or slower than others. For instance, a 2-step walk cycle takes more time compared to a 2-step run cycle. In order to manipulate such data matrices, the temporal length of each action should be warped such that it matches the temporal length of relative actions which we intend to blend with. As for the number of columns in the matrices, they do not

require any manipulation since the exact same number and orientation of markers have been used for all capture sessions.

The system is trained based on three sets of data. The first set is the base animation containing the same secondary theme as the test set, while the second is the captured sequence containing the desired secondary theme (target theme). Together they are called the training data. The training data along with the test set are used together for the warping process. The system is then tested for new sets of action sequences called the test data. The training data along with the test sequence are temporally warped such that they are all equal in length. A straightforward approach is simple scaling; that is stretching or compressing the motion sequences. Yet the sequences must be aligned appropriately. To tackle this problem, the warping of the matrices is carried out such that a key feature in the sequence is aligned for the three datasets.

To conduct the very crucial piecewise time warping, the signals are divided into two sections, the corresponding first sections are warped together, and the second parts are warped together as well. This is carried out in order for different corresponding sections of the action sequences to be aligned correctly. Thus, selecting the proper feature for which the signals are sliced at is very critical.

Different methods were tested for finding the critical features, used to determine the boundaries for the piece-wise time warping. The first method is by manually selecting the features. For walk and run sequences, the feature instance was selected as the occurrence of a foot touching the ground. This means in a sequence where walking starts with the right foot initiating the walk and ending in the same situation, the instance when the left

foot touches the ground is labelled as the feature. For the jump sequence, the feature is defined as the instance where the actor is at the peak of his/her jump (when the inclining motion stops and returning to the ground initiates). Figures 5.10 and 5.11 illustrate the piecewise warping conducted using this technique.

It is noticeable from Figure 5.11 that the warped signals are stretched to last the exact amount of time (46 frames) which is the average of each of the individual signals in Figure 5.10. Yet misalignment in the global minimum peaks has occurred. Also the alignment of the signals between frame 20 and 30 is discarded which is an undesired artefact. Thus selecting the features manually is not recommended.

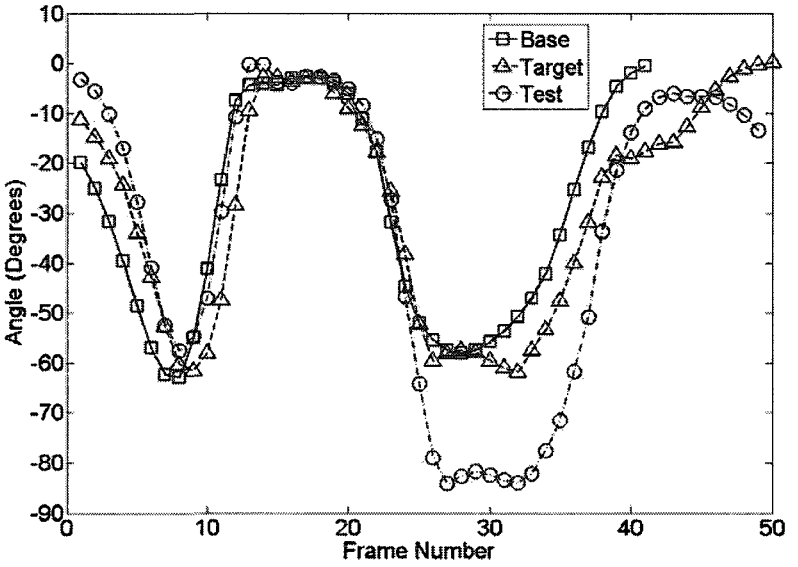

Figure 5.10. Signals from base, target, and test matrices before time warping

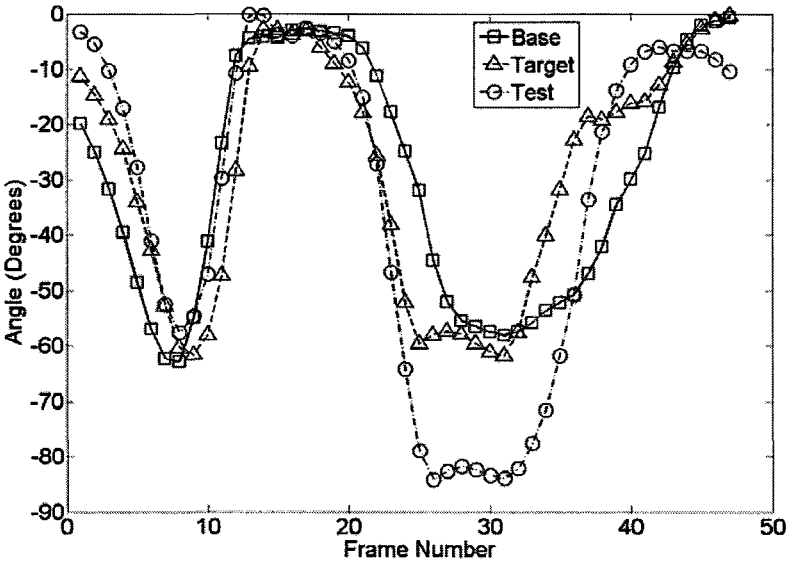

Figure 5.11. Signals from base, target, and test matrices after time warping using manual feature selection

The second method is by means of statistical analysis. In Chapter 3, the following was presented:

$$\begin{bmatrix} \Theta_1 \\ \Theta_2 \\ \vdots \\ \Theta_m \end{bmatrix} = \begin{bmatrix} \left(\theta_1^{x_1}, \theta_1^{y_1}, \theta_1^{z_1}\right), \left(\theta_1^{x_2}, \theta_1^{y_2}, \theta_1^{z_2}\right), \ldots, \left(\theta_1^{x_n}, \theta_1^{y_n}, \theta_1^{z_n}\right) \\ \left(\theta_2^{x_1}, \theta_2^{y_1}, \theta_2^{z_1}\right), \left(\theta_2^{x_2}, \theta_2^{y_2}, \theta_2^{z_2}\right), \ldots, \left(\theta_2^{x_n}, \theta_2^{y_n}, \theta_2^{z_n}\right) \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ \left(\theta_m^{x_1}, \theta_m^{y_1}, \theta_m^{z_1}\right), \left(\theta_m^{x_2}, \theta_m^{y_2}, \theta_m^{z_2}\right), \ldots, \left(\theta_m^{x_n}, \theta_m^{y_n}, \theta_m^{z_n}\right) \end{bmatrix} \tag{5.2}$$

We name the calculated matrix in Equation 5.2 by Θ. Based on Equation 5.2, we can derive Equation 5.3 where N is the number of markers.

$$\underline{\Theta}=\begin{bmatrix} v_1^1, v_1^2, \cdots, v_1^{3n} \\ v_2^1, v_2^2, \cdots, v_2^{3n} \\ \vdots \\ v_m^1, v_m^2, \cdots, v_m^{3n} \end{bmatrix}=\begin{bmatrix} \left(\theta_1^{x_1}-\theta_2^{x_1}, \theta_1^{y_1}-\theta_2^{y_1}, \theta_1^{z_1}-\theta_2^{x_n}, \ldots, \theta_1^{x_n}-\theta_2^{x_n}, \theta_1^{y_n}-\theta_2^{y_n}, \theta_1^{z_n}-\theta_2^{z_n}\right) \\ \left(\theta_2^{x_1}-\theta_3^{x_1}, \theta_2^{y_1}-\theta_3^{y_1}, \theta_2^{z_1}-\theta_3^{z_1}, \ldots, \theta_2^{x_n}-\theta_3^{x_n}, \theta_2^{y_n}-\theta_3^{y_n}, \theta_2^{z_n}-\theta_3^{z_n}\right) \\ \vdots \\ \left(\theta_{m-1}^{x_1}-\theta_m^{x_1}, \theta_{m-1}^{y_1}-\theta_m^{y_1}, \theta_{m-1}^{z_1}-\theta_m^{z_1}, \ldots \theta_{m-1}^{x_n}-\theta_m^{x_n}, \theta_{m-1}^{y_n}-\theta_m^{y_n}, \theta_{m-1}^{z_n}-\theta_m^{z_n}\right) \end{bmatrix} \quad (5.3)$$

The vector Φ is then calculated by Equation 5.4.

$$\Phi=\begin{bmatrix} \sum_{i=1}^{3n} v_1^i \\ \sum_{i=1}^{3n} v_2^i \\ \vdots \\ \sum_{i=1}^{3n} v_m^i \end{bmatrix} \quad (5.4)$$

The **Φ** vector determines the net velocity of all markers for each frame. The total velocity of all markers is a very determinant and informative value. The maximum or minimum arrays of **Φ** verify the instances where the actor has reached his/her maximum angular velocity of the action. The largest column value in **Φ** determines the critical feature, which is where the maximum net angular velocity is observed. Assuming that during the course of an action, the markers accelerate from zero angular velocity to this value and then accelerate back to zero, time warping is carried out for each section of the action. The result is actions of the same length and temporally aligned such that sections of the action with positive angular acceleration and negative angular acceleration are aligned. Both the maximum and minimum values were tested. The minimum net angular velocity features determine the instance when the body is in such formation that even though it might be moving, the net angular velocity is zero. Figures 5.12 and 5.13 show the effect of selecting the features based on this method.

It can be noticed that in Figure 5.12 and Figure 5.13, a similar stretching as Figure 5.11 has occurred and the output signals are all 46 frames. Yet the signals are aligned more properly. The global maximum and minimum points are aligned with higher accuracy as they take place in the same time compared to Figure 5.11, and frames 20 to 30 have not been misaligned. Comparing Figure 5.12 and Figure 5.13, we can observe that Figure 5.13 shows more alignment for the case of global minimum. This especially can be observed for frames 20 to 30 where the three signals are more aligned compared to the other methods. The same result was acquired when tested for signals of other joints, therefore in this research the minimum angular velocity features were adopted for time warping of actions.

Animating the new warped signals confirms the fact that the conducted operations on the datasets have not manipulated the data significantly and that the primary or the secondary themes are not altered, thus the groundwork for computing the conversion matrices for re-synthesis of motion capture data has been laid successfully.

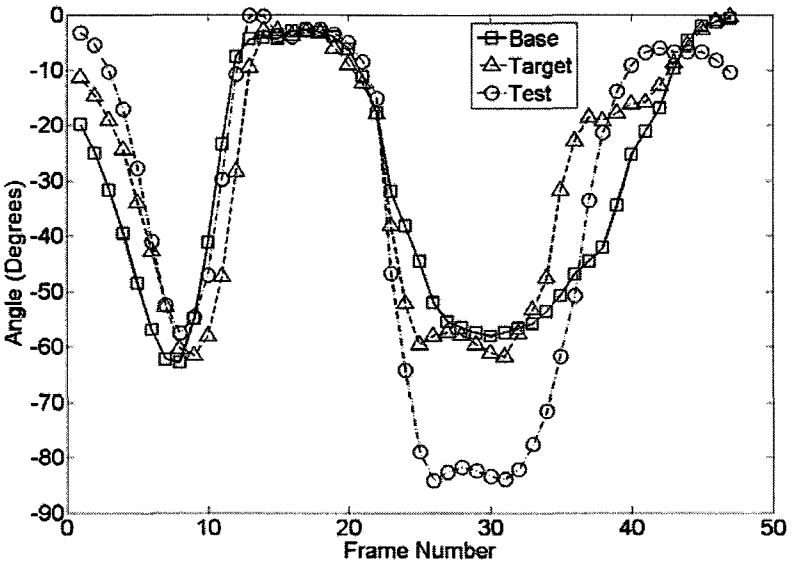

Figure 5.12. Signals from base, target, and test matrices after time warping using maximum velocity features

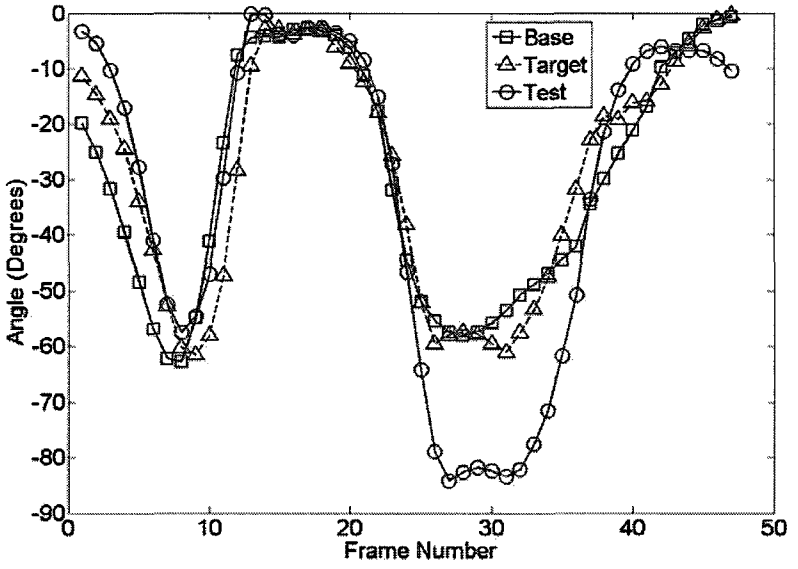

Figure 5.13. Signals from base, target, and test matrices after time warping using minimum velocity features

Chapter 6: Re-Synthesis

6.1. Introduction

The final step of this research is the re-synthesis of motion data. The goal of re-synthesis is to manipulate motion capture action sequences to accomplish a desired sequence. In simple terms, the goal of this chapter is to convert the secondary themes of actions from one style to another, while maintaining the primary themes. For instance Figure 6.1 and Figure 6.2 illustrate a *tired walk* and an *energetic walk* sequence respectively where the goal is to convert the data for Figure 6.1 to the data for Figure 6.2. An impression must be given to viewers that in the second sequence, walking is being performed with more energy. This is visible through more movement in the body especially the arms. An important fact in transformations is that the action class (in this case *walking*) remains the same.

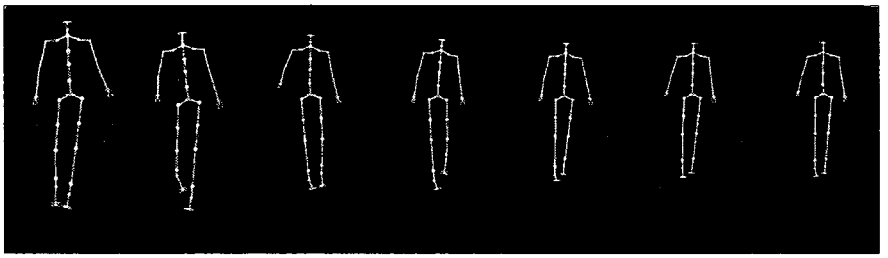

Figure 6.1. Tired walk sequence

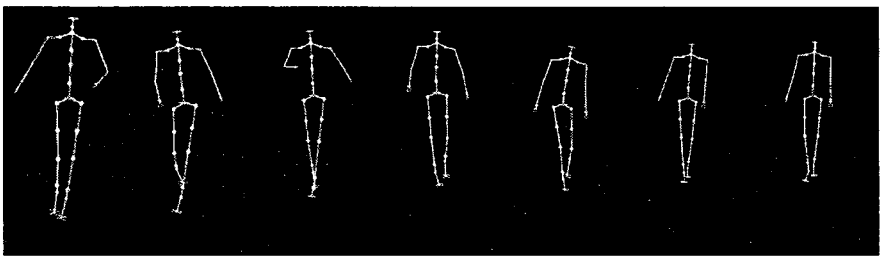

Figure 6.2. Energetic walk sequence

The possible secondary themes which are target to transformation in this research are those related to age, gender, and energy as we will be focusing on converting masculine to feminine, young to old, and low energy to high energy actions.

There may be various categories of approaches towards solving this problem. In this research, we have focused on two diverse approaches. The first is by means of artificially intelligent systems and algorithms. These solutions would provide slight or even no insight on the actual governing rules of human motion and relations between the primary and secondary themes. These systems simply learn the data and are later able to reproduce similar data regardless of the meaning of the data. The second method is by means of defining mathematical models which describe human motion. Successive to

defining the proper model, mathematical computations are employed to derive the desired rules of motion and relations between primary and secondary themes.

As mentioned in Chapter 2, some research has been carried out in the field of re-synthesis and action style transformations, yet the problem is far from solved. In this chapter, we have taken both approaches towards solving this problem. Artificial neural networks are employed as the artificially intelligent system. A mathematical model is also proposed. This model is then employed for derivation of transformation functions capable of converting the secondary themes, leaving the primary theme untouched.

6.2. Artificial Neural Networks

The goal in this section of the research is to convert the different styles of specific classes of actions i.e. secondary themes using an A.I based technique. Neural networks are selected as the tool for this purpose. Neural networks possess unique abilities which have made them quite suitable for numerical processing and meaningful number generation. Neural networks learn the relationship between the provided inputs and outputs by updating the weights within the neurons. Thus when provided with new data, the possible outcome is projected. Here we tend to utilize this property for synthesis of motion data.

The same neural network which was proposed for the recognition task in Chapter 4 is used for transformation of secondary themes. To review the process, in Chapter 4, a separate neural network was configured for each secondary theme of each action class,

i.e. a total of 18 neural networks for 3 action classes and 6 styles for each class. This will make the method unique in the sense that both classification and re-synthesis is carried out using the same system.

In contrast with most action synthesis systems where they are statistical, the proposed system here is A.I. based. Once the training of the system presented in Chapter 4 Section 4 is complete, the same system is employed for re-synthesis and secondary variation transformations. As explained earlier, in the second cascade of ANNs presented in Figure 3.5 are trained based on Figure 3.6. The overview of this section of the system is an ANN trained for each style of a particular action. It is expected that test samples be provided to this row of ANNs and the outcomes be quantitively compared to the output data which is constructed based on the input. However, the second row of ANNs is constructed and trained to predict the successive frame of an action given the previous frame data, and therefore if any initial configuration is provided as the input of any ANN block of the second row, the projected successive frame is calculated. If the frames of action class 1 (e.g. walking) of style 1 (e.g. feminine) be presented as the inputs to the ANN for action class 1 and style 2 (e.g. masculine), the output frames are the projected outputs for a style 2 action regardless of the nature and style of the input.

We represent the input matrix in Equation 6.1 by $\varphi^{j,k}$ and the output in Equation 6.2 by $\varphi^{j,k\to k'}$ where j is the action class, k is the original style class, and k' is the target style class of action.

$$\left(\varphi^{j,k}\right)^T \equiv \left[A_1^{j,k}\ A_2^{j,k}\ \cdots\ A_m^{j,k}\right] \tag{6.1}$$

$$\left(\varphi^{j,k\to k'}\right)^T \equiv \left[A_2^{j,k'}\ A_3^{j,k'}\ \cdots\ A_{m+1}^{j,k'}\right] \tag{6.2}$$

It should be noted that as illustrated in Equation 6.2, the style-converted action retains a single frame shift in time yet the length of the transformed action remains unchanged and the single frame shift is insignificant.

6.3. Human Motion Model

Human action and motion in general is a combination of an action class and a set of stylistic variations adjoined to it. We call the action class which are the dominant signals throughout the data sets, primary themes, and the stylistic variations as secondary motor themes. Based on this definition, any action can be modeled by Equation 6.3 where *Y[k,r]* is the action sequence with the primary theme *k* and secondary theme *r* as it is observed, *P[k]* is the primary motor theme of the same action class, and *S[r]* is the secondary theme of class *r*. In Equation 6.3 *w[r]* is the weight applied to the secondary themes, and *e* represents the noise available in the system.

$$Y[k,r(1:f)] = P[k] + \sum_{r=1}^{f}\left(w[r]\cdot S[r]\right) + e \tag{6.3}$$

The model Equation 6.3 is defined such that a combination of different styles can be applied to the same primary class of action. For instance, for action class jump, the secondary theme young-feminine can be defined. A total of *f* different *S* functions are foreseen in the model. Nevertheless, while training the system, all samples are designed

and assumed to hold only one secondary theme for simplification. Each secondary theme function is learnt separately, and ultimately, they can be merged to form a multi-style secondary theme.

The goal of this research is to create the *S[r]* values, while trying to minimize *e*. In most cases, the *S[r]* signals are not powerful enough to influence the perception of *k*, i.e. the action class. But rarely, this scenario might take place. For instance if *r* relates to class of energy and speed related themes, *S[r]* along with a notable *w[r]* applied to a primary theme of walking can cause confusion as to whether *Y[k,r]* was originally walking with a strong *S[r]* and *w[r]*, or whether *k* determines the action class of running for *Y[k,r]*, and *S[r]* and *w[r]* have been insignificant. This dilemma is addressed in Chapter 7, yet further research is ongoing to tackle this issue.

With the assumption that in Equation 6.3, the *k* and *r* values are unchanged and the same model is still valid, two action sequences ($Y_1[k,r_1]$ and $Y_2[k,r_2]$) of the same class ($P[k]$) and different secondary themes ($S[r_1]$ and $S[r_2]$) can be modeled by Equations 6.4 and 7.5 respectively.

$$Y_1[k,r_1] = P[k] + w[r_1] \cdot S[r_1] + e_1 \tag{6.4}$$

$$Y_2[k,r_2] = P[k] + w[r_2] \cdot S[r_2] + e_2 \tag{6.5}$$

Differentiating among Y_1 and Y_2 generates ΔY presented by Equation 6.6.

$$\Delta Y = \left(w[r_2] \cdot S[r_2] - w[r_1] \cdot S[r_1]\right) + \left(e_2 - e_1\right) \tag{6.6}$$

Assuming equal weights for the secondary themes and that only one secondary theme is available for each sequence, the differentiated secondary themes provide the desired transformation matrix between the two secondary themes, $\Gamma[r_1,r_2]$, which is defined by Equation 6.7 and computed by Equation 6.8.

$$\Gamma[r_1, r_2] \equiv w[r_2] \cdot S[r_2] - w[r_1] \cdot S[r_1] \tag{6.7}$$

$$\Gamma[r_1, r_2] = (Y[k, r_2] - Y[k, r_2]) + (e_1 - e_2) \tag{6.8}$$

In simple terms, for an action sequence with only one secondary theme, the transformation matrix is derived by differentiating the training and the target sequences, along with noise elimination techniques.

It can be examined that the transformation function T defined by Equation 6.9 can be applied to action sequence $Y[k,r(1:f)]$ for converting the secondary theme r_i to r_j. It should be noted that this function will only produce reasonable results once temporal alignment is conceived (based on discussions in Chapter 5 or by any other means).

$$T_{r_i,r_j}[Y[k, r(1:f)]] = Y[k, r(1:f)] + w \cdot \Gamma(r_i, r_j) \tag{6.9}$$

To eliminate the term $(e_1 - e_2)$ from Equation 6.8, low pass filtering is utilized. The affect of the noise reduction process is presented in Figure 6.3.

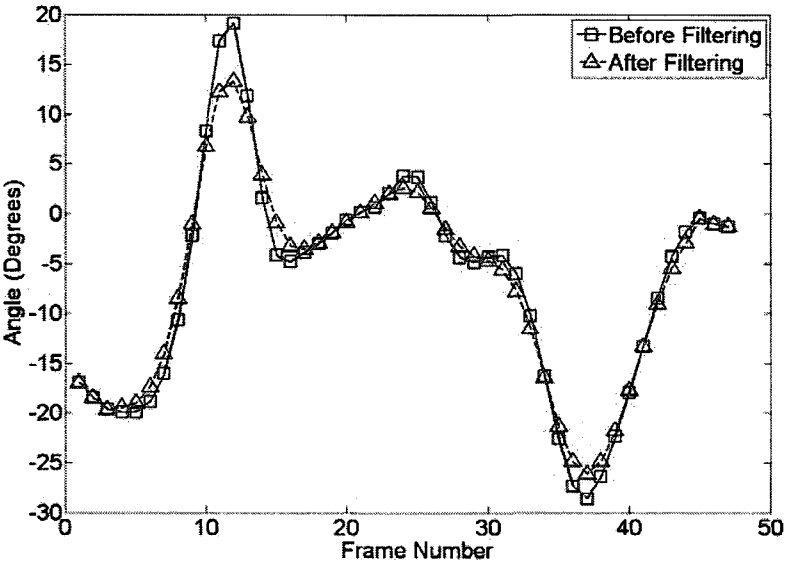

Figure 6.3. The effect of low-pass filtering

The conversion signals are a result of differentiating between two sequences of the same primary theme with different secondary themes. Due to the frame-to-frame approach to differentiation, this process is very sensitive to noise or any artefact introduced through the warping process. As a result, local and global maximum and minimums are produced. Animating the final raw outcome illustrates various artefacts for different joints. Undesired motion such as tremor and rigidity in some body parts are the direct affect of the proposed techniques. Applying a low-pass filter results in smoothing of the conversion data and the output signals. As illustrated in Figure 6.3, filtering eliminates some sharp local maximum and minimums. Disappearance of the tremor symptoms of the animations prior to filtering the data reaffirms the necessity of post-processing the data.

Another technique was also employed for transforming the secondary themes. With the assumption that one secondary theme is present in the sequence, the interpolation procedure derived by Equations 6.10 and 6.11 will also result in *Y[k,r]* which is a sequence with the original primary action class *P[k]*.

$$Y[k,r] = \frac{\left(w_1 \cdot Y_1[k,r_1] + w_2 \cdot Y_2[k,r_2]\right)}{w_1 + w_2} \tag{6.10}$$

$$Y[k,r] = P[k] + \frac{\left(w_1 \cdot w[r_1] \cdot S[r_1] + w_2 \cdot w[r_2] \cdot S[r_2]\right)}{w_1 + w_2} + e \tag{6.11}$$

In Equation 6.11 the second term presents an interpolation among the two secondary themes while the third term *e* is the interpolated noise signal. Also the first term *P[k]* indicates that the primary theme of the action has remained unchanged. It is very important that the secondary themes which are to be interpolated be of the same nature. For instance, they must be all related to age, or gender, or energy. Interpolating between secondary themes of different categories is meaningless as it is not logical to interpolate, for instance, between a young theme and a low energy theme. Nevertheless interpolation between a young theme and an old theme is likely to produce a mid-aged theme for the primary action class *k*. The same low pass filtering process is carried out to eliminate the noise from the output data.

The two techniques presented above have been implemented and the results are discussed and compared in Chapter 7. It is important to note the fact that the conversion data does not need to include all the joint values. For instance the markers placed on the head do not have a significant contribution on the secondary themes. A selected handful

of markers, thought to be influential to the secondary themes are altered and the rest are left unchanged, since including all the markers will only increase the run time and system error. The markers which have been selected for manipulation are on the spine, arms, and legs.

6.4. Numerical Evaluation of Outputs

Various literature have addressed the problem of motion data synthesis [55 – 63, 25 – 27]. Yet the lack of a quantitative evaluation of the results is clearly visible. To evaluate the outcome, the animation of the output frames can be observed. This technique however, is inaccurate in terms of measuring the system performance accuracy and is basically a qualitative approach. Since we are dealing with recorded rotation values of markers in 3D feature space the Pearson product-moment correlation coefficient (PCC) is selected for evaluation of the synthesized data. PCC which is also known as the sample correlation coefficient calculates the correlation between two sets of measured samples of x^1 and x^2 as shown by Equation 6.12 where n is the number of samples and ρ is the correlation coefficient, ranging from -1 to +1. A coefficient of 0 means absolutely no correlation and +1 shows perfect correlation. Similarly, -1 indicates perfect inverse correlation.

$$\rho = \frac{n\sum x_i^1 x_i^2 - \sum x_i^1 \sum x_i^2}{\sqrt{n\sum (x_i^1)^2 - (\sum x_i^1)^2}\sqrt{n\sum (x_i^2)^2 - (\sum x_i^2)^2}} \tag{6.12}$$

The goal in the case of this research is to employ Equation 6.12 and determine whether $\varphi^{j,k\rightarrow k'}$ is of style class k or k', i.e., transformation has been successful or not. The way to go about this would be to calculate average of Equation 6.12 for $\varphi^{j,k\rightarrow k'}$ and all the actions used to train the k' ANN and compare the results with the average of Equation 6.12 for $\varphi^{j,k\rightarrow k'}$ and all the actions used to train the k ANN. The k and k' style actions used to train the ANNs however are actions of different lengths, therefore a time warping technique similar to the one explained in Chapter 5 Section 3 is employed to accomplish actions of the same length of $\varphi^{j,k\rightarrow k'}$. It should be noted that time warping is not carried out for conversion purposes and is simply applied to measure the correlation between converted k-to-k' action and the two existing style classes of k and k'. Once the time warping is complete, PCC is measured for each marker data of $\varphi^{j,k\rightarrow k'}$ and the respective marker data of all the k and k' actions where the average of each calculation is represented as $\rho(tr)$ and $\rho(in)$ respectively. If $\rho(tr)$ shows a higher (numerically greater) correlation than $\rho(in)$ this would mean that the transformation of style k-to-k' has been carried out successfully.

This evaluation method is used when synthesis of data is carried out, either by means of ANN or transformation model. In both cases, the output data is in the form of motion capture data. Once the data have been produced, they need to be verified as to which primary and secondary theme class they belong to. We have used PCC in Chapter 7 Section 4 for valuation of synthesized data.

Chapter 7: Experimental Results and Discussions

7.1. Introduction

In this chapter the results of our proposed methods for all three tasks of segmentation, classification, and re-synthesis (style transformation) are presented and discussed. For each section described in previous chapters, and using each of the presented methods, the quantitative results are computed and explained in this chapter.

This research is composed of three major fields. The first section aims at locating and segmenting specific actions based on trained action classes using two different techniques. Nearest neighbour search and HMM are the two techniques used for segmentation of actions. The second field is classifying the segmented sequences.

Nearest neighbour search and HMM which were used for segmentation of actions, are also utilized for classifying the located actions. ANN is another tool used for classifying the segmented actions. The third section of this research is transformation of secondary themes. This is also referred to as re-synthesis of human motion data. Two different methods have been proposed in this regard, the first being based on the artificially intelligent technique of ANN and the second is based on describing human motion through a mathematical model and putting that model into use for transforming the secondary themes. All these techniques have been carried out, the results are mentioned in this chapter, and the necessary discussions are provided.

Later in Chapter 8 we will compare the results of the different methods for each of the three tasks and come to a conclusion on the most suitable technique for segmentation, classification, and re-synthesis of human motion data provided by optical motion capture systems.

7.2. Segmentation

Based on the descriptions in Chapter 4, the first step for analysis of human motion data is to locate and segment different actions from within sequential combination of actions. For instance if a sequence contains several meaningful actions or some meaningful actions and some random data with similar dimensions to motion capture data, the goal is to locate the meaningful ones, and provide them to the classification system for further evaluation.

To test our proposed techniques, 18 sequences were selected. Each sequence contains one of the possible classes of action, taking into account the secondary themes of action. The possible actions within each sequence used in this research are provided in Table 7.1.

Table 7.1. Utilized primary and secondary theme classes

Primary Theme	Secondary Theme	
Walk	*Feminine*	*Masculine*
	Tired	*Energetic*
	Young	*Old*
Run	*Feminine*	*Masculine*
	Tired	*Energetic*
	Young	*Old*
Jump	*Feminine*	*Masculine*
	Tired	*Energetic*
	Young	*Old*

Nine of the eighteen sequences contain one class of the mentioned actions and a number of meaningless frames before and after the intended action class while another nine set of the sequences contain a meaningful action (which the system is not trained for) before or after the intended action. Five different samples for each of the 18 classes were captured and tested using the proposed methods summing up to a total of 90 test samples. Usually 2/3 of the whole dataset are used for training a system and the

remaining 1/3 are utilized for testing and evaluation, therefore 90 samples (5 number of samples per each class), are sufficient for testing our systems.

Here based on the methods mentioned in Chapter 4, we locate the specific meaningful actions using nearest neighbour search and HMM. The acquired results (Table 7.2 and Table 7.3) based on these methods are compared to manual segmentation of the actions for error calculation. The error is calculated by the difference of the true starting frame number and the frame number recognized by the system as the starting frame of an action.

7.2.1. Nearest Neighbour

In Chapter 4, two different methods were proposed for segmentation of actions, the first being nearest neighbour search. For nearest neighbour search, the following two different methods of distance calculation were proposed:

- Absolute difference
- Pseudo-Euclidean distance

The descriptions of these two methods were provided in Chapter 4 Section 2. While Euclidean distance has shown to be the more popular method for numerical pattern classification [84], absolute difference is also tested for further evaluation of nearest neighbour technique.

Table 7.2 presents the segmentation results of 90 test samples. The training samples are selected manually and nearest neighbour is applied based on what was explained in Chapter 4. The error for each of the distance calculation methods is provided in the table for comparison. It is clearly visible that Pseudo-Euclidean angular distance provides finer segmentation results compared to absolute angular difference. A possible reason is that absolute difference takes each array as a separate value while the Pseudo-Euclidean distance takes into account the fact that each three components form the data related to one marker resulting in a more meaningful outcome.

Based on Table 7.2 we can also conclude that locating an action sequence from within a sequence containing excessive meaningless data is less complicated compared to the cases where other action sequences have been added to the original action. This is simply because the added action class may contain similar frames to the target action which might confuse the locating algorithm. This trend holds true for most of the test samples except for few where the error has increased by a small amount.

Another conclusion that can be made from the table is that while absolute difference computation provides a wider range of results, i.e. greater variance, the Pseudo-Euclidean distance measurement shows to be more reliable as the results obtained by this measurement are quite close to one another.

The overall standard deviation (SD) for the results using absolute difference and Pseudo-Euclidean distance are 3.62 and 2.06 respectively. When employing absolute difference, the Old Run segmentation error rate falls outside 2*SD of the mean error and is not reliable. The rest of the results all fall within the range.

Table 7.2. Segmentation using nearest neighbour

Sample	Action Class	Action Style	Additional Frames	Mean Frame Difference Absolute Difference	Mean Frame Difference Pseudo-Euclidean Distance
1-5	Walk	Feminine	Meaningless	8.6	6.0
6-10	Walk	Masculine	Punch	13.2	6.8
11-15	Walk	Energetic	Meaningless	8.0	7.6
16-20	Walk	Tired	Kick	7.8	5.6
21-25	Walk	Young	Meaningless	5.8	0.8
26-30	Walk	Old	Pickup	9.6	9.4
			Mean Error	8.83	6.03
31-35	Run	Feminine	Meaningless	7.8	2.0
36-40	Run	Masculine	Punch	12.6	3.2
41-45	Run	Energetic	Meaningless	9.2	9.8
46-50	Run	Tired	Kick	10.0	8.2
51-55	Run	Young	Meaningless	13.4	4.6
56-60	Run	Old	Pickup	18.2	9.0
			Mean Error	11.87	6.13
61-65	Jump	Feminine	Meaningless	4.0	5.8
66-70	Jump	Masculine	Punch	4.2	6.0
71-75	Jump	Energetic	Meaningless	6.8	0.2
76-80	Jump	Tired	Kick	6.0	5.2
80-85	Jump	Young	Meaningless	11.8	8.4
86-90	Jump	Old	Pickup	11.2	10.8
			Mean Error	7.33	6.07
			SD:	3.62	2.06
			Overall Mean Error	9.34	6.08

7.2.2. HMM

The second proposed method was by employing HMM. The same HMM which is later used for classification of actions was utilized for segmentation of the actions. This is done by selecting an adjustable sliding window which scans through the motion sequence, returning the window with the most likelihood as the segmented action. The benefits of this segmentation technique compared to the nearest neighbour search are:

- Both segmentation and classification are carried out simultaneously.
- If more than one meaningful action of the trained classes are embedded in the sequence, it can locate them all whereas our proposed algorithm for nearest neighbour can only locate one action.
- The size of the action to be located is dynamic.

Table 7.3 presents the results of segmentation using this technique. Based on the description in Chapter 4, a network of hidden Markov models is configured, trained, and the sliding window is put to action. While the results of both segmentation and classification are obtained simultaneously, here the focus is on the segmentation only and we ignore the classification results for now.

The table clearly illustrates the fact that segmentation using HMM is carried out with more accuracy compared to that using nearest neighbour search. While the performance quality does not follow a similar trend for meaningful and meaningless added frames, this method is superior to nearest neighbour based on both accuracy and ability to dynamically locate the closing frame as well as the starting frame of actions. It can also

be verified that the accuracy of locating the closing frames is slightly higher than that of starting frames.

The SDs of Table 7.3 are 1.20 and 0.85 for starting frames and closing frames respectively. For locating the starting frames, all the data fall within the range of 2*SD of the mean error. The same situation occurs for closing frames, except for the Tired Run which the error value is exactly on the 2*SD mark.

Table 7.3. Segmentation using HMM

Sample	Action Class	Action Style	Additional Frames	Starting Frame Difference	Closing Frame Difference
1-5	Walk	Feminine	Meaningless	3.2	2.0
6-10	Walk	Masculine	Punch	3.0	0.4
11-15	Walk	Energetic	Meaningless	2.4	2.2
16-20	Walk	Tired	Kick	0.6	1.0
21-25	Walk	Young	Meaningless	0.0	1.8
26-30	Walk	Old	Pickup	1.2	1.2
			Mean Error	1.73	1.43
31-35	Run	Feminine	Meaningless	1.2	2.2
36-40	Run	Masculine	Punch	2.8	1.8
41-45	Run	Energetic	Meaningless	1.2	0.4
46-50	Run	Tired	Kick	2.0	3.2
51-55	Run	Young	Meaningless	4.4	1.2
56-60	Run	Old	Pickup	1.8	2.6
			Mean Error	2.23	1.90
61-65	Jump	Feminine	Meaningless	3.8	1.8
66-70	Jump	Masculine	Punch	2.2	2.6
71-75	Jump	Energetic	Meaningless	2.2	2.4
76-80	Jump	Tired	Kick	0.4	1.0
80-85	Jump	Young	Meaningless	2.4	0.2
86-90	Jump	Old	Pickup	0.8	1.0
			Mean Error	1.96	1.50
			SD:	1.20	0.85
			Overall Mean Error	1.97	1.61

7.3. Classification

The second step for human motion analysis is classification of segmented actions. The actions which have been previously located within a motion sequence in (Section 2 of this chapter) are not directly provided to the classification subsystems for classification, since any amount of error in locating and segmenting the actions would result in unreal results of the classification system. The goal here is to perform a correct evaluation of the proposed classification techniques, therefore, manually segmented actions are used for a correct evaluation.

In this section, 5 samples for each of the 18 action sequences are provided for evaluation. These 90 samples have not been used in the training process of the classification subsystem and are new to the system. All the primary and secondary theme combinations are included in the samples.

Three techniques are proposed for classification of actions, based on the description provided in Chapter 4. The methods are:

- Nearest neighbour
- HMM
- ANN

While nearest neighbour search is considered as one of the simplest means for classification, HMM is considered a probabilistic technique. ANN is also employed for classification of actions. In this section the results for each method are discussed and

analyzed. The accuracy is measured by the number of correct classifications divided by the total number of samples available for classification (for each class).

7.3.1. Nearest Neighbour

Nearest neighbour is the first technique which we used for classification of actions. This is done by calculating the numerical distance between the test action sequence and the training action set. The training sample which returns the least distance, determines both the primary and secondary class of the action. The two different distance measures used for locating the actions (absolute difference and Pseudo-Euclidean distance) are employed here for classification.

The drawback of the proposed method in Chapter 4 is the fact that the length of the test sample must be exactly equal to all the training samples for acquiring the best results, which is an impossible requirement. Thus, the accuracy rate in some cases drops to as low as 20% which is considered a significant decrease.

Table 7.4 presents a sample of some of the classified actions as well as the results of action and style classification using nearest neighbour search for all the 90 samples. It is observed that the accuracy of the action classification is considerably higher than that of style classification. This is not unexpected since the secondary theme features happen to be quite minute and insignificant compared to the primary theme features.

Another pattern which can be concluded based on Table 7.4 is that similar to action segmentation, when using nearest neighbour search, Pseudo-Euclidean distance shows a higher accuracy when compared to absolute difference.

The SDs of primary theme classification using absolute difference, secondary theme classification using absolute difference, primary theme classification using Pseudo-Euclidean distance, and secondary theme classification using Pseudo-Euclidean distance are 15.04%, 15.17%, 16.80%, and 18.43%, respectively. All the results in Table 7.4 fall within 2*SD of the mean accuracy of the respective techniques.

Table 7.4. Samples and results of classification using nearest neighbour search

Sample	Action Class	Action Style	Accuracy: Primary Theme Absolute Difference	Accuracy: Secondary Theme Absolute Difference	Accuracy: Primary Theme Pseudo-Euclidean Distance	Accuracy: Secondary Theme Pseudo-Euclidean Distance
1-5	Walk	Feminine	60%	40%	80%	80%
6-10	Walk	Masculine	60%	40%	80%	20%
11-15	Walk	Energetic	40%	20%	80%	60%
16-20	Walk	Tired	80%	20%	60%	60%
21-25	Walk	Young	60%	20%	60%	40%
26-30	Walk	Old	60%	40%	80%	40%
	Mean Accuracy:		60.00%	30.00%	73.33%	50.00%
31-35	Run	Feminine	60%	60%	80%	60%
36-40	Run	Masculine	40%	40%	60%	40%
41-45	Run	Energetic	60%	20%	60%	40%
46-50	Run	Tired	60%	20%	80%	20%
51-55	Run	Young	60%	20%	60%	60%
56-60	Run	Old	60%	40%	60%	60%
	Mean Accuracy:		56.67%	33.33%	66.67%	46.67%
61-65	Jump	Feminine	60%	60%	100%	60%
66-70	Jump	Masculine	80%	40%	100%	60%
71-75	Jump	Energetic	80%	40%	100%	80%
76-80	Jump	Tired	80%	60%	100%	20%
81-85	Jump	Young	100%	40%	100%	60%
86-90	Jump	Old	80%	60%	100%	60%
	Mean Accuracy:		80.00%	50.00%	100.00%	63.33%
	SD:		15.04%	15.17%	16.80%	18.43%
Overall Mean Accuracy:			65.56%	37.77%	80.00%	53.33%

7.3.2. HMM

Hidden Markov models are very powerful tools for pattern classification especially when dealing with data which are variable with time. Based on the proposed algorithm in Chapter 4, a classification system is configured. The action sequences which are used to test the system are the same samples used to test the nearest neighbour classifier. Each test sample is run through each of the 18 HMM networks and is classified relatively.

Table 7.5 shows the results for classification of actions using HMM. The overall action (primary theme) classification accuracy shows a slight improvement compared to the nearest neighbour search when the Pseudo-Euclidean distance was computed. The style (secondary theme) classification, is also more accurate than the nearest neighbour technique. Similar to the previous section, the style (secondary theme) classification is carried out with lower accuracy compared to the primary theme classification.

With regards to the accuracy of different primary and secondary themes being classified, the performance of the system shows to be superior for Jump when the primary theme is to be classified, while Walk and Jump show similar performances and higher than that of Run when being classified for secondary themes.

The SDs for classification of primary and secondary themes are 18.75% and 12.78% respectively. For recognition of the primary themes using HMM, the Old Run falls outside the 2*SD of the mean accuracy. The rest of the results for both themes however, remain within the desired range.

Table 7.5. Classification using HMM

Sample	Action Class	Action Style	Accuracy: Primary Theme	Accuracy: Secondary Theme
1-5	Walk	Feminine	100%	80%
6-10	Walk	Masculine	80%	60%
11-15	Walk	Energetic	80%	80%
16-20	Walk	Tired	60%	60%
21-25	Walk	Young	80%	60%
26-30	Walk	Old	80%	60%
	Mean Accuracy:		80.00%	66.67%
31-35	Run	Feminine	60%	40%
36-40	Run	Masculine	60%	60%
41-45	Run	Energetic	60%	60%
46-50	Run	Tired	80%	40%
51-55	Run	Young	80%	60%
56-60	Run	Old	40%	40%
	Mean Accuracy:		63.33%	50.00%
61-65	Jump	Feminine	100%	60%
66-70	Jump	Masculine	100%	80%
71-75	Jump	Energetic	100%	80%
76-80	Jump	Tired	100%	60%
81-85	Jump	Young	100%	60%
86-90	Jump	Old	100%	60%
	Mean Accuracy:		100.00%	66.67%
	SD:		18.75%	12.78%
	Overall mean Accuracy		81.11%	61.11%

7.3.3. ANN

The third and last proposed method for classification of actions is ANN. A network of 21 ANNs in the form of MLP for each action and different styles are constructed and trained using the Resilient Back-Propagation (RPROP) technique (3 networks for primary action classes and 18 for secondary classes).

Based on the discussions in Chapter 4, two hidden layers with 45 to 90 neurons in each hidden layer were provided for each network. Various numbers of experiments proved the final orientation of 45, 70, 50, 45 neurons for layers one to four to be most effective. For different ANN blocks, slight changes were made on the number of hidden neurons based on the number of frames of actions used to train the respective ANN. As mentioned in Chapter 4, the RPROP training method was adopted for the forest. The training was then carried out in all the neural networks using the RPROP training technique as well as Batch training with weight & bias learning rules, Powell-Beale conjugate gradient back-propagation, Gradient descent back-propagation, Scaled conjugate gradient back-propagation, and Levenberg-Marquardt back-propagation, where none proved to be more precise for this application compared to RPROP. The RPROP, however, displayed the most number of local error maxima during training, which could result in erroneous results if the training stops within one of the maxima. Providing the networks with large number of training epochs significantly reduces the probability of being trapped in local error maxima. Figure 7.1 illustrates the training curve by number of training epochs vs. performance error for walking composed of all the styles. Due to the large size of ANN inputs, outputs, and the ANNs themselves, up to 25000 training epochs were completed. Also different training functions were tested for different layers

where Log-Sigmoid, Tan-Sigmoid, Tan-Sigmoid, and Pure-Linear were adopted for layers one to four, respectively.

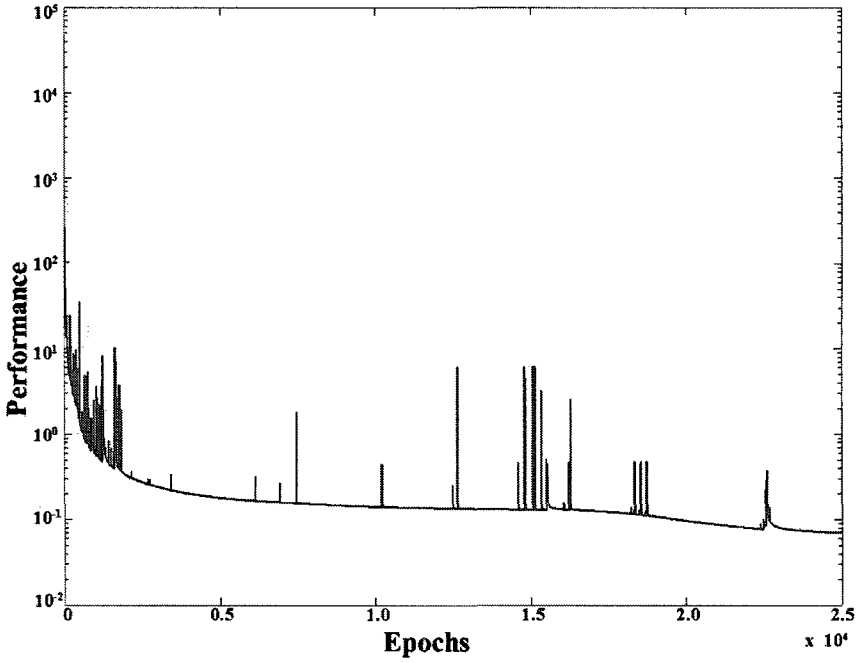

Figure 7.1. Walking ANN training process

Figure 7.2 presents the training curve when a network is trained using samples of only one specific secondary theme for a primary action class. It is visible that fewer local error maxima occur compared to Figure 7.1. The reason is that learning different styles of an action is quite confusing for the ANN. Once the network learns a specific style of an action, upon introducing a new style to be learnt, all the weights have to be re-adjusted accordingly. The weights must be altered in a way that would be representative of both the previous class and the new class, resulting in higher error rate. The extra local maxima which appear in Figure 7.1 could be a direct result of using new styles during the

training procedure. Due to this unpredicted event, the test data are only presented to the second row of neural networks which have been trained by only one style per action class. This will definitely increase the run-time. The run-time after employing this technique will be doubled since each test sample must be analyzed by 18 networks instead of 9 (3 for classification of primary theme and 6 for classification of secondary theme), yet in the same time the classification performance will be boosted as well.

Once the data have been analyzed by the system, MSE has been used to classify the output data as explained in Chapter 4 Section 4. Table 7.6 presents the results for classification of 90 samples (similar to previous sections). In this table, jumping has been classified with greater accuracy for both primary and secondary themes. Also generally, classification of the primary themes has been more accurate with respect to that of secondary themes.

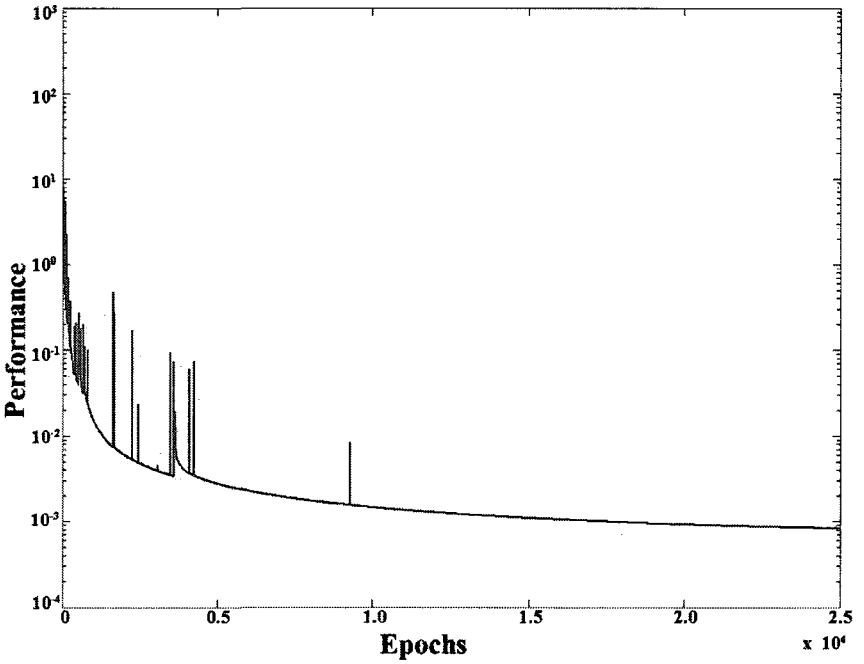

Figure 7.2. Feminine Walk ANN training process

The SDs for Table 7.6 are 12.15% and 13.53% for classification of primary and secondary themes respectively. In a similar case to HMM, Old Run, when intended to be classified for the primary theme, falls outside the 2*SD of the mean accuracy.

Table 7.6. Classification using ANN

Sample	Action Class	Action Style	Accuracy: Primary Theme	Accuracy: Secondary Theme
1-5	Walk	Feminine	80%	60%
6-10	Walk	Masculine	80%	80%
11-15	Walk	Energetic	100%	80%
16-20	Walk	Tired	80%	80%
21-25	Walk	Young	80%	80%
26-30	Walk	Old	80%	60%
	Mean Accuracy:		83.33%	73.33%
31-35	Run	Feminine	80%	80%
36-40	Run	Masculine	80%	80%
41-45	Run	Energetic	80%	60%
46-50	Run	Tired	80%	80%
51-55	Run	Young	100%	100%
56-60	Run	Old	60%	60%
	Mean Accuracy:		80.00%	76.67%
61-65	Jump	Feminine	100%	80%
66-70	Jump	Masculine	100%	100%
71-75	Jump	Energetic	100%	100%
76-80	Jump	Tired	100%	60%
81-85	Jump	Young	100%	80%
86-90	Jump	Old	100%	80%
	Mean Accuracy:		100.00%	83.33%
	SD:		12.15%	13.53%
Overall mean Accuracy			87.78%	77.78%

7.4. Re-synthesis

The final task in this research is transformation of secondary themes. Based on Chapter 4 Section 4 and Chapter 6, two rather diverse approaches were taken towards tackling this problem:

- ANN
- Motion model

Each approach has its own advantages which will be discussed later. For testing each approach, the three primary themes of walking, running, and jumping are available. For each class of action, we chose the secondary themes of *masculine*, *young*, and *tired* as the base themes, setting our goal to converting them to *feminine*, *old*, and *energetic*. Five different samples were tested for each transformation.

After the data have been re-synthesized with the aim of transforming the secondary theme, the fabricated data must be evaluated to confirm or reject a successful secondary theme transformation. Two different methods are chosen for evaluation:

- PCC (Pearson's Correlation Coefficient)
- User study (Questionnaire)

The first method is described in Chapter 6 Section 4 (Equation 6.12). For the second method we asked 5 engineering graduate male students to carefully watch each action sequence before and after the transformation and either confirm or reject a successful transformation. These people were clarified to either approve or rule out a transformation between the styles. The transformation was not required to be perfect for acceptance by

the participants. The user study here was carried out by BVHacker and for a preliminary evaluation of the output data alongside the numerical evaluation by means of PCC. The accuracy is determined by the number of successful transformations divided by the total number of samples tested for this purpose (for each class).

7.4.1. ANN

After constructing and training the set of 18 neural networks, which were used for classification, 9 of them are again used here for testing the capability of the system for re-synthesizing motion capture data with the desired secondary theme. Since our goal is to create walking, running, and jumping sequences containing the styles of feminine, old, and energetic, the following networks are used for transformation:

Feminine walk, old walk, energetic walk, Feminine run, old run, energetic run, feminine jump, old jump, energetic jump.

As explained earlier, we have assumed that each network is specialized for a specific style of a specific action, thus capable of predicting the consecutive frame given a known frame, hence upon providing for instance the feminine-walk network with the frames of a masculine walk, it would try to predict the consecutive frames in feminine form regardless of the nature of the provided action. Stacking all the predicted frames generates the data matrix. Figures 7.3 and 7.4 illustrate the exaggerated masculine walk and the output for transformation using a feminine walk network respectively. It can be

seen that the gesture of the arms in Figure 7.3 which is raised for demonstrating the masculine style, has been altered by being lowered and more close to the body in Figure 7.4.

Table 7.7 presents the outcome for testing 5 different samples for each transformation. Both methods of evaluation have been tested. We can see that in general the 5 people who participated in evaluating favoured the transformation since more average positive votes were assigned to the transformation with respect to PCC (Equation 6.12).

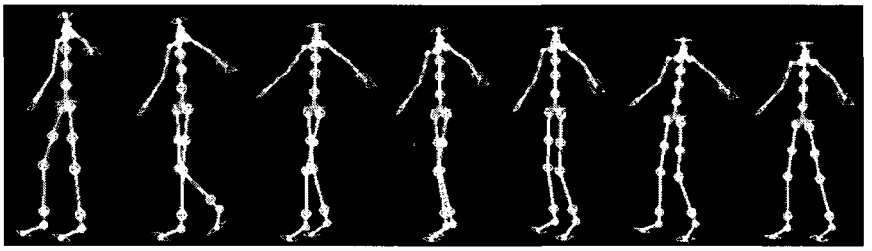

Figure 7.3. Masculine walk

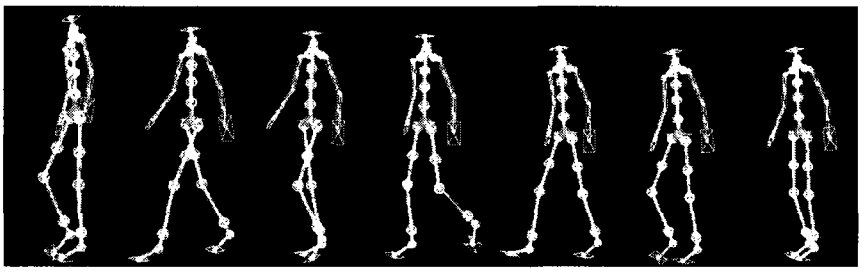

Figure 7.4. Transformation to feminine walk using ANN

When the results of the system here are evaluated using PCC, the SD is 10.0% and when evaluation is carried out by means of user study, the SD is 3.53%. Despite the apparent difference between the range of variations of the results using the two evaluators, all the results are located inside the 2*SD range of the mean.

Table 7.7. Style transformation using ANN

Sample	Action Class	Initial Secondary Theme	Intended Secondary Theme	PCC	User Study
1-5	Walk	Masculine	Feminine	80%	88%
6-10	Walk	Young	Old	60%	92%
11-15	Walk	Tired	Energetic	80%	84%
			Mean Accuracy:	73.33%	88.00%
16-20	Run	Masculine	Feminine	80%	84%
21-25	Run	Young	Old	80%	88%
26-30	Run	Tired	Energetic	80%	88%
			Mean Accuracy:	80.00%	86.67%
31-35	Jump	Masculine	Feminine	80%	84%
36-40	Jump	Young	Old	60%	84%
41-45	Jump	Tired	Energetic	60%	80%
			Mean Accuracy:	66.67%	82.67%
			SD:	10.0%	3.53%
Overall mean Accuracy				73.33%	85.87%

7.4.2. Motion Model

As discussed in Chapter 5 various techniques were used for determining the feature instances for the piecewise time warping of the data. The minimum velocity-based features proved to be most suitable and are employed in the system. Successive to warping the action sequences, the transformation of the secondary theme take place using Equations 6.4 to 6.11.

Interpolation between the sequences was implemented to compare with our approach. The weights of 0.5 and 0.5 for w_1 and w_2 were selected for each of the sequences to be interpolated. Figure 7.5 shows an original masculine jump while Figure 7.6 presents an original feminine jumping sequence successive to the time warping procedure. Figure 7.7 illustrates the interpolated output of the two actions. It is noticed that the action is femininized to some extent; yet creating a 100% feminine jump is only possible through setting the weight of the masculine jump to zero which results in a warped version of the training feminine sequence. Figure 7.8 illustrates the output of the system based on computing the transformation function. In this sequence, the legs are femininized similar to the interpolation output, yet with few variations. The arms however are adjusted with more significance. Overall, based on visual evaluation, the output is no worse, if not better, than the practical interpolation. The huge advantage remains, however, the fact that using the proposed model, we have not eliminated the base action which the secondary theme is desired to be added upon. The same affect can be seen for other primary and secondary themes and the advantage holds valid.

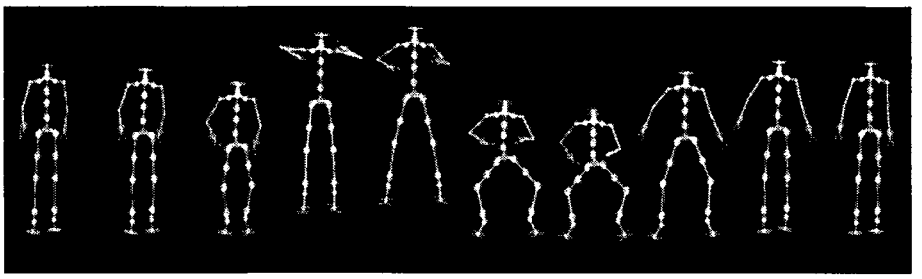

Figure 7.5. Original masculine jump.

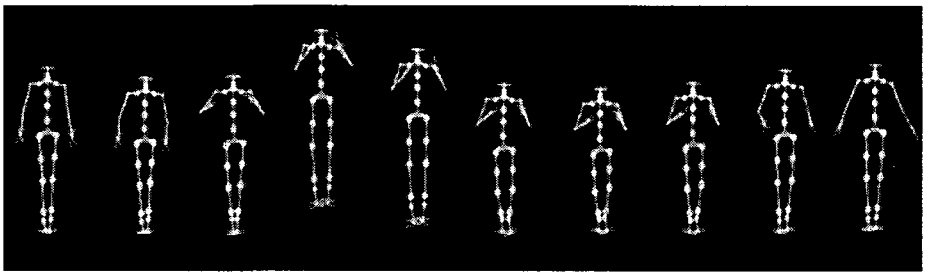

Figure 7.6. Original Feminine jump.

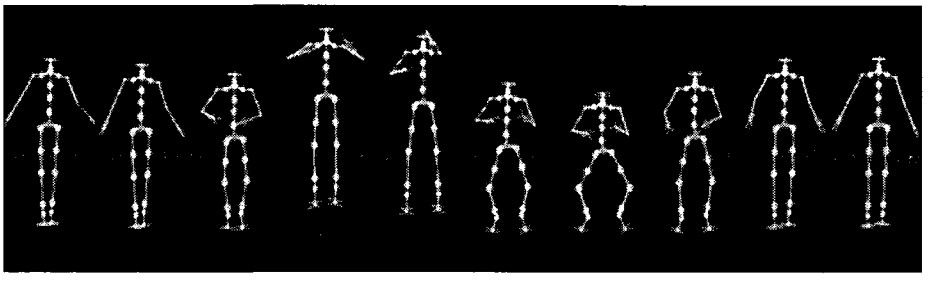

Figure 7.7. Interpolation output.

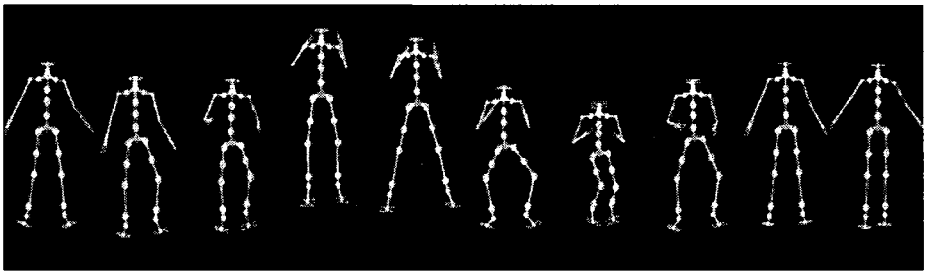

Figure 7.8. The output using transformation function.

A rather detailed investigation for the proposed transformation technique shows that the secondary themes have been detached and added to the test samples with high precision. Figure 7.9 (left) shows the original masculine walk before conversion and Figure 7.9 (right) shows the same frame after the conversion.

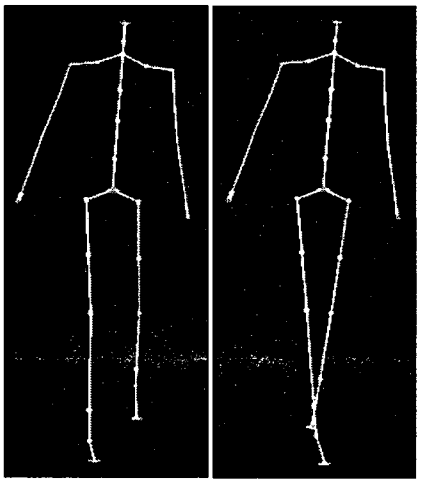

Figure 7.9. Original masculine walk (left), converted to feminine walk (right).

The Figure clearly shows the transformation of masculine walk to feminine walk using the toolkit. While in masculine walk, the legs are placed rather apart, in feminine walk, the legs are placed sequentially in front of one another. Also the movement of the hip is limited in masculine walk as opposed to the feminine where more movement is visible in that area. Similar manipulations are visible for other styles and other classes of action, which shows the significance of our proposed model and transformation function.

Similar manipulations are visible for other styles and other classes of action such as that illustrated in Figure 7.10 which presents the conversion of low energy (tired) run (left) to a high energy run (right) using our proposed model. The configuration of the legs clearly shows the successful conversion of the two styles.

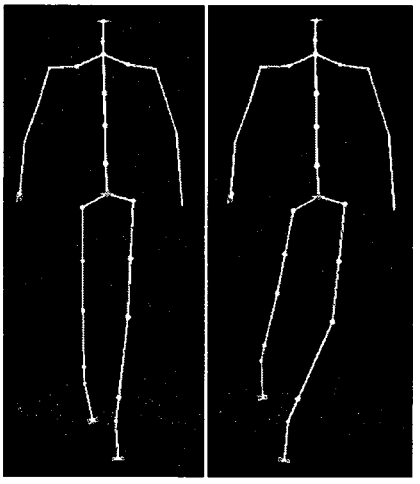

Figure 7.10. Original low energy run (left), converted to energetic run (right).

Table 7.8 presents the results for style transformation using the motion model. A similar trend to Table 7.7 regarding PCC and the user study is observed where the user study shows a higher rate of approval for the transformed animations. The data in Table 7.8 have a SD of 8.82% for the PCC evaluator and 2.0% for evaluation based on user study. Similar to Table 7.7, the range of changes for the PCC is greater than that of user study. None of the entries in Table 7.8 fall outside the 2*SD range of the mean. When evaluating by means of user study, however, Masculine-to-Feminine Walk and Young-to-Old Jump, are on the exact 2*SD mark.

Table 7.8. Style transformation using motion model

Sample	Action Class	Initial Secondary Theme	Intended Secondary Theme	PCC	User Study
1-5	Walk	Masculine	Feminine	80%	92%
6-10	Walk	Young	Old	60%	88%
11-15	Walk	Tired	Energetic	80%	88%
			Mean Accuracy:	73.33%	89.33%
16-20	Run	Masculine	Feminine	60%	88%
21-25	Run	Young	Old	80%	88%
26-30	Run	Tired	Energetic	80%	88%
			Mean Accuracy:	73.33%	88.00%
31-35	Jump	Masculine	Feminine	80%	88%
36-40	Jump	Young	Old	80%	84%
41-45	Jump	Tired	Energetic	80%	88%
			Mean Accuracy:	80.00%	86.67%
			SD:	8.82%	2.0%
Overall mean Accuracy				75.55%	88.00%

7.5. Discussions

In this section we will compare the different techniques used for each section of this research and discuss the pros and cons of each methods based on the desired goals. The first task of this research was segmentation of actions. Figure 7.11 shows that nearest neighbour search operates with a considerably higher error rate compared to HMM. Although the Pseudo-Euclidean distance measure provides significantly higher accuracy compared to the absolute difference measure, it is still quite erroneous compared to HMM. The HMM maintains a better performance for locating both the starting and closing frames of actions, especially the latter.

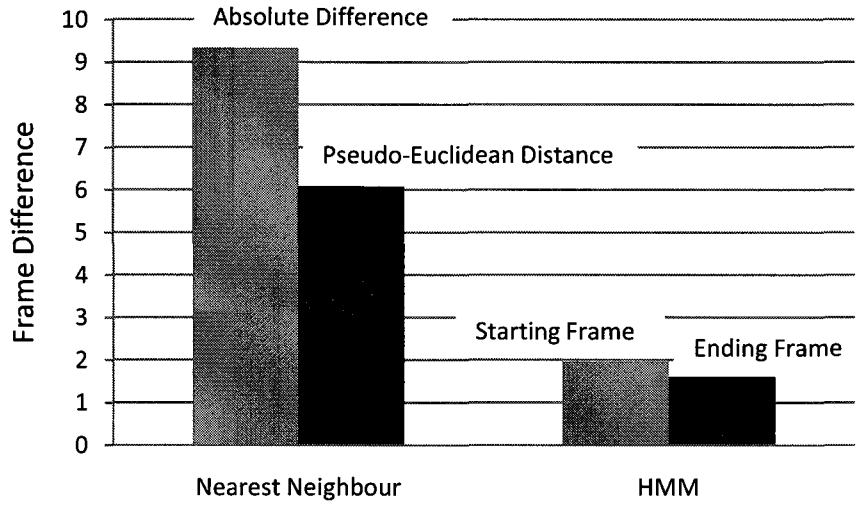

Figure 7.11. Overall segmentation performance

The second task was classification of both primary and secondary themes of actions. Figure 7.12 illustrates the mean classification success of each of the classifiers. It can be concluded that generally the rate of classification of primary themes is higher than secondary themes for all classifiers. The nearest neighbour search with the absolute difference measure provides the poorest results for both primary and secondary themes. Nearest neighbour with the Pseudo-Euclidean distance, provides almost the same performance as the HMM, yet the secondary theme is classified slightly better with the HMM. ANN on the other hand shows the best classification performance for both themes. The leap in the performance for the secondary theme classifications is especially visible. One reason could be the fact that in the ANN system, the data for the entire body were employed as opposed to the HMM where only the lower half of the body was used to train the system. This could affect the classification result for secondary theme classifications since some of the stylistic variations are performed by the upper portion of the body. In addition, the difference in performance for classification of the two themes is minimized using ANN.

The focus of the final section of this research was on re-synthesis of motion capture data with the aim of transforming the secondary theme. Figure 7.13 presents the overall performance of the two transformation techniques, evaluated by both PCC and users. The difference in the two methods has not been significant, while clearly the user study approved the process more than PCC. In general, the motion model seems more promising than ANN.

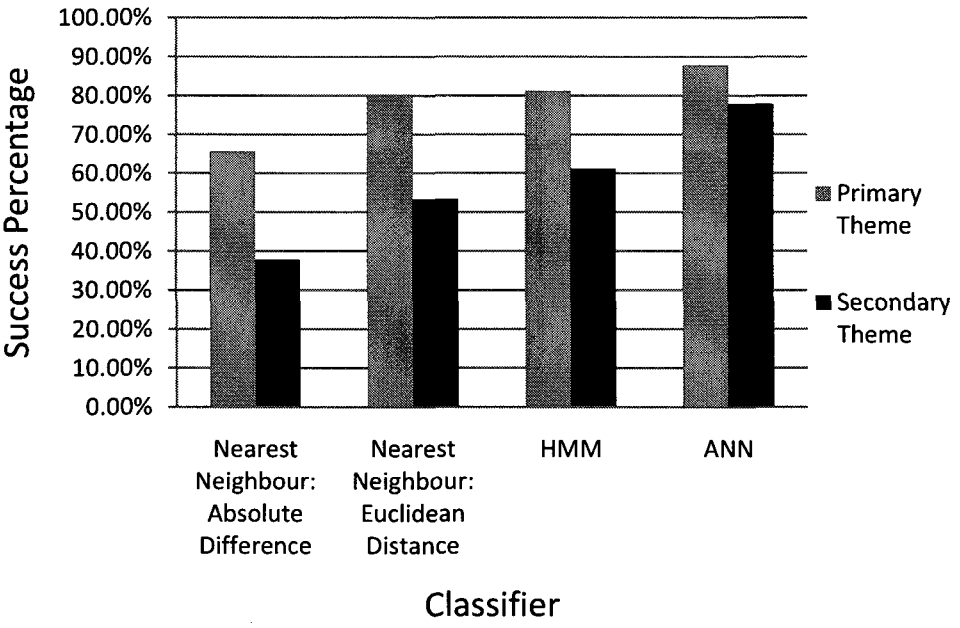

Figure 7.12. Overall classification performance

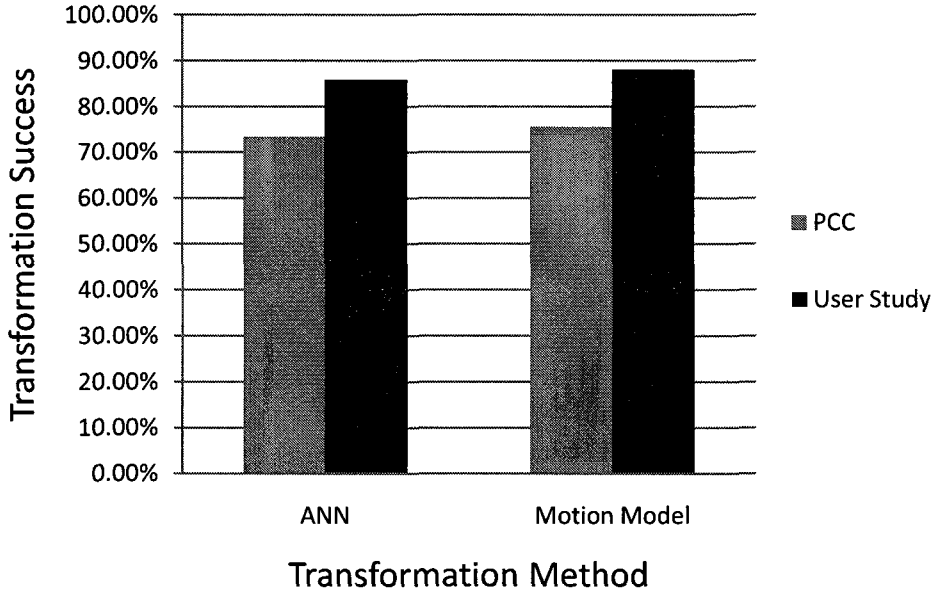

Figure 7.13. Overall transformation performance

Based on the mentioned discussions, each technique holds specific characteristics which may be suitable for different applications. While the ANN holds the best classification performance, it is also capable of synthesizing the data needless of any alterations to the system. The very long runtime and inability to perform segmentation are the drawbacks of the ANN based system.

The HMM is capable of performing both segmentation and classification tasks simultaneously. The classification performance is inferior to ANN, yet it is able to detect motion in combinational sequences, finding both the starting and ending frames, while the ANN is unable to do so. HMM carries out the segmentation task with significantly higher accuracy with respect to the nearest neighbour search. The training time of the HMM is also less than that of ANN.

Nearest neighbour search has been the least precise tool among all. Nevertheless it is capable of segmenting and classifying the actions at the same time. The only superiority of this method over the others is the runtime where they proved to be significantly faster than the rest.

For re-synthesis of motion capture data, the motion model showed to be slightly better and considerably faster than ANN. Yet the ANN is able to classify and synthesize motion at the same time.

7.6. Runtime

The scope of this research is to segment, classify and re-synthesize motion capture data, and real-time processing is not considered essential. MATLAB, on the other hand, is not intended for real-time processing as most of its libraries and functions are far from real-time. Most procedures of this research are not suitable for real-time purposes due to use of exhaustive methods. The very bulky type of data also has negative effects on the runtime.

Hidden Markov models and neural networks, especially, take quite a long time to train prior to being used for the intended tasks. In terms of runtime, successive to the training procedures, ANNs showed to be faster than HMMs. If we take the training time into account, the speeds of the systems created and run on a PC with 2.8 GHz dual core processor with 3.00 gigabytes of RAM, come in the following order: nearest neighbour search (less than 30 minutes), HMM (near 2 hours), and finally ANN (6-7 hours). These training and runtime values, however, can be significantly reduced if implemented in a different language such as C which is significantly faster than MATLAB for processing of data.

For re-synthesis of motion and secondary theme transformations, the mathematical model proved to be significantly faster than the ANN method. The complete re-synthesis section using the model, including the piecewise time warping, takes few seconds, while the ANN takes hours to train and provide the output.

Chapter 8: Conclusion

8.1. Concluding Remarks

Through the course of this work, various techniques were tested for segmentation, classification, and re-synthesis of human motion capture data. The data were captured using the Vicon motion capture system. Preprocessing was carried out as an essential step towards utilizing the data. For each task, the required systems were introduced and created, and various tests were carried out to evaluate each phase. Various tools such as MATLAB and BVHacker have been employed for different tasks. While all the computations were carried out in MATLAB, BVHacker was used for visualising the synthesized output data.

The major contributions of this research are:

- Segmentation and classification of different actions using motion capture data by means of various tools (nearest neighbour search, HMM, ANN).

- Classification of various actor styles (secondary themes).
- Introducing and applying the novel piecewise time warping for temporal alignment.
- Developing a model for human motion capable of describing both primary and secondary themes. Transformation functions for manipulating secondary themes have been derived.
- Transforming secondary themes by means of an intelligent method (ANN) as well as the motion model.
- Proposing and testing a technique for numerical evaluation of synthesized data i.e. style transformation output.

The focus of the first section of this research has been on locating and segmenting the actions by means of nearest neighbour search (absolute difference and Pseudo-Euclidean distance calculation), and HMM. Using the HMM classifier the closing frame of the actions was also located.

The second task of this research has been classification of individual actions for both primary and secondary themes. Three different classifiers were employed: nearest neighbour search (absolute difference and Pseudo-Euclidean distance calculation), HMM, and ANN.

The final task of this research has been re-synthesis of motion capture data with the aim of transforming the secondary themes of the actions. Two different approaches were taken for this aim, first by ANN and second by deriving a mathematical model for human motion and calculating transformation matrices. Piece-wise time warping was proposed

as an essential step towards utilizing the model. An attempt was made through this research to present a mathematical means of evaluating the transformation outcome. Pearson's correlation was used for this aim. Also due to uncertainty in the ability of PCC to correctly evaluate the transformations, a user study was performed using 5 participants for determining whether the process has been successful or not.

The following are a recap of specific findings for segmentation, classification and re-synthesis of human motion capture data, acquired through the course of this dissertation:

- For segmentation of motion, HMM proved to be more accurate than nearest neighbour search. The HMM itself proved more precise for locating the closing frames of actions compared to the starting frames.
- Pseudo-Euclidean distance is more reliable means (compared to absolute difference) for measuring distance when dealing with motion capture data.
- RPROP training is a suitable method for training ANN for the aim of recognition and re-synthesis of human motion capture data.
- ANN trained using RPROP performs more accurate than HMM and nearest neighbour search for classification.
- For re-synthesis of motion capture data, ANNs are constructive tools.
- Time warping is an important step towards reaching temporal alignment which itself is essential prior to performing specific operations where one-to-one correspondence among two different sequences are required. Piecewise time warping proved to be very suitable for achieving temporal alignment.
- For re-synthesis of motion capture data, the proposed motion model shows to be very promising and is capable of defining human motion.

- PCC is a measure which can provide a reasonable evaluation on the success of style transformations.

8.2. Future Work

Through this research, segmentation and classification of 3 main actions with 6 style variations per each action class was carried out with a relatively high accuracy and precision. Further primary and secondary themes are to be tested. Actions such as idling, strafing left and right, crouching, and a collapsing or death sequence which are popular in the field of animation and game design are projected to be included. Other secondary themes such as weight and mood need to be considered.

Optimization of the proposed systems is a task which is very likely to enhance the outcome. Some different techniques however may provide more accurate results. Implementation of new techniques for segmentation and classification of motion capture data is a possible area for further research. These techniques may include the following along with plenty others:

- Fuzzy based systems
- Genetic algorithms
- Support vector machines

By implementing the above techniques and even some other methods, further classifiers with different characteristics in terms of action segmentation, action classification, style classification, re-synthesis and run-time may be accomplished.

For re-synthesis of data with new transformed styles, and for the piecewise time warping procedure room for further development of the system is available. For instance the following can be carried out and studied:

- Employing expert knowledge: With the help of physicians and experts, specific features can be detected, tracked, and manipulated to convert the styles of an action. Different tools can be used for this method yet it is projected that fuzzy interpretations are required.
- Applying physical constraints for re-synthesis of motion data: The synthesis process does not currently take into account the physical constraints that come into play during human motion. Considering these constraints can result in more realistic motion sequences.
- Selection of multiple features when performing the piecewise time warping: The method used in this research was based on the fact that each signal was cut into two sections. Each section was warped and the two were re-joined to form a new signal. This can be done with more than one feature. The signals would be cut in multiple sections, each snippet can then be warped accordingly, and all the sections will then be re-joined.
- Non-linear expanding and compressing of the sectioned pieces: In this research linear interpolation was performed for expansion and excluding equally distributed excess samples was employed for compression. This procedure can be

carried out differently. For instance, non-linear interpolation can be a legitimate substitute for the linear kind.

- Non-linear model for human motion: The proposed model in this research was in the form of a linear additive description for primary and secondary themes. Non-linear, differential, or even other types of models relating the two types of themes can be possible alternatives for describing human motion. It should be noted that defining different models would accordingly require different transformation functions for transforming the themes.
- Combination of secondary themes: Another field in which this research seems promising is the combination of secondary themes. It is usually necessary to produce combined styles such as old-angry, or young-feminine and etc. Different methods for combining the transformations for each style such as weighted average, MIN/MAX, and rule-based operators may be required.

The software used for visualizing the synthesized data was BVHacker (described in Appendix B). This requires the transfer of data from MATLAB (the processing software which was employed in this research) to BVHacker. For speeding up the process and for more convenience, implementing a similar platform in MATLAB seems interesting and handy. As a result, the processed and synthesized data will directly be animated in MATLAB.

Another field which requires further development is the means of evaluation of synthesis and style-transformed data. So far a numerical technique (explained in Chapter 6 Section 4) along with a user based evaluation (the questionnaire) have been employed. Other numerical means of evaluating the style change can be tested and studied such as

different correlation factors. Specific features determinant of each style can also be defined and detected through an action for verifying a successful transformation of styles. Also different and further improved questionnaires used by animation experts can provide more accurate evaluation of the results.

Finally the runtime is an area which requires further effort. While real-time processing was outside the scope of this research, heading for real-time systems capable of performing similar tasks may be an interesting topic, worthy of further research. While a real-time system may require an entirely new approach and new software, minimizing the runtime of the current systems is foreseen. One method to go about this is by means of a real-time animation or gaming engines such as XNA (http://msdn.microsoft.com/en-us/xna/default.aspx). A real-time toolkit can be created in XNA and the matrices which have been developed by the proposed procedures in this research can be transferred and employed by the toolkit for creating state of the art real-time procedural animation. This is currently being carried out by a team under the supervision of Dr. Arya.

Appendix A

To record the movement of human motion in the form of digital data is often referred to as motion capture which dates back to 1970's [85]. The recorded motion can be used for on the spot or afterwards playback to be used in computer animation applications. The Vicon MX40 motion capture system provides all the input data. Vicon MX40 is a marker-based optical motion capture system which applies basic light reflection laws for precise recording of motion. Marker-based motion capture systems can be categorized as optical and non-optical. The optical are governed by the basic laws of light reflection while the non-optical can be inertial [86], mechanical [87], or magnetic [88] based systems.

Optical marker-based motion capture systems make use of at least 6 infrared cameras. Each camera includes a projector which is usually positioned as concentric circles around the lens. The camera frame rate usually exceeds 120 fps and the resolution is as high as 16 megapixels. As shown in Figure A.1, concentric circles of red LEDs surround the lens of the camera. These LEDs emit red light which will be reflected by the markers and collected by all cameras.

Markers are small white reflective spheres which are positioned on the special suit which the actor wears. The markers must be placed very carefully and on critical joints and muscles. The exact orientation of the markers for accurate and complete data capture is provided by the system manual. Usually more than 14 markers are used. These markers reflect the red light and the reflected light is to be imaged by the cameras. In theory only 2 cameras are sufficient for 3D representation of a scene, yet for higher accuracy and

better results, 6 or more cameras are used. It is possible that markers be hidden by body parts throughout an action; therefore using 6 cameras increases the possibility that all markers are visible by at least 2 cameras all the time. Most systems employ 8-12 cameras. The actors must be careful not to have any reflective objects on, although it is almost certain that all the body (except the face) are covered by the black stretch suit, gloves, shoes, and hat. Figure A.2 illustrates the orientation of the markers on the special capture suit and a scene of capture session is provided in Figure A.3. The data from the light which is reflected by the markers and captured by the cameras are collected by a single computer which all cameras are connected to.

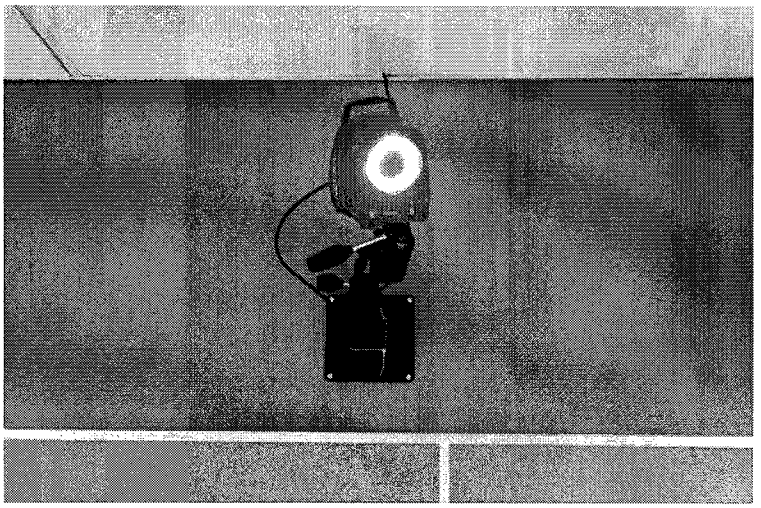

Figure A.1. Vicon Motion Capture Camera

The specifications of the camera and lens of the current Vicon motion capture system at Carleton University are as follows (http://mocap.csit.carleton.ca/):

- Sensor Type: CMOS

- Sensor Resolution: 2352 (W) x 1728 (H) = 4,064,256 pixels ~ 4 x 10^6 pixels
- Sensor Depth: 10bits/pixel greyscale; 1024 grey levels
- Sensor Dimensions: 16.46mm (W) x 12.10 (H); 20.43 mm diagonal
- Pixel Size: 7 microns x 7 microns
- Focal Length: 12.5 mm
- 1" C-Mount View:
    - Field of View Angle (Horizontal) 54.22 degrees
    - Field of View Angle (Vertical) 42.0 degrees
- Full Frame:
    - Field of View Angle (Horizontal) 67.04 degrees
    - Field of View Angle (Vertical) 51.65 degrees

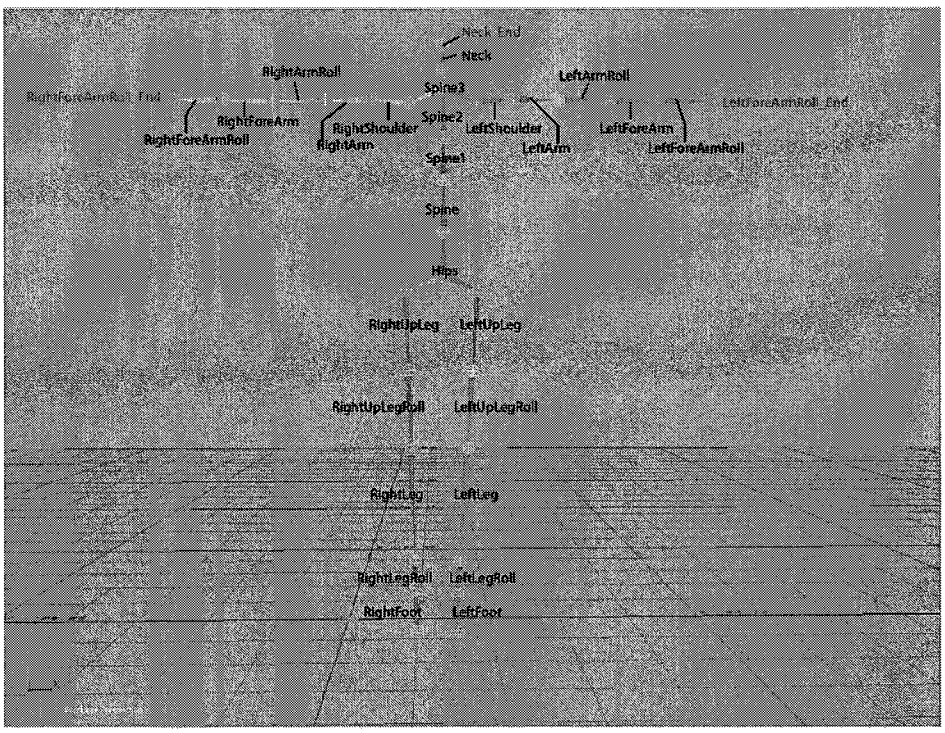

Figure A.2. Orientation of the markers

Figure A.3. Motion Capture Session

Prior to initiating a capture session, the system must carefully be calibrated. The intention of the calibration is to define the origin as well as the imaginary boundaries of the capture area. Also like any other optical system, when cameras are used, sufficient calibration is required to adjust the system based on the orientation and positioning of the cameras with respect to each other and with respect to the target object.

The Output data provided by the system is in the form of .V (skeletal file) and .VSK (motion data). These files cannot be directly used for numerical processing for the intentions of this research. These two files are imported into Autodesk MotionBuider to produce an .FBX file, which is then applied to an existing character template (in Motionbuilder) which can then exported as the .BVH format. The .BVH files contain two sections:

- Header
- Motion Data

The header section of .BVH files describes the formation of the skeleton and the order in which the markers are read through the motion section of the file, as well as the frame rate at which the recording has taken place. The motion data contains the rotation values of each coordinate of each joint at each frame. This data is used for analysis and processing, and the systems provided in this research are built based on these data. Further explanation about the motion data is provided in Chapter 3.

Appendix B

All the processing of the data is carried out in MATLAB. MATLAB is an extremely powerful tool for numerical processing of data. Developed in 1970's in the University of New Mexico it was initially created for students to avoid learning Fortran Finally in 1984 it was re-written in C and commercialized (http://www.mathworks.com/company/newsletters/news_notes/clevescorner/dec04.html). Today, various commercial and user-created libraries and programs are available for MATLAB.

MATLAB R2007a was used for almost all the processing in this research. A comprehensive library containing the required functions was created and employed. Figure B.1 illustrates the main environment of this software.

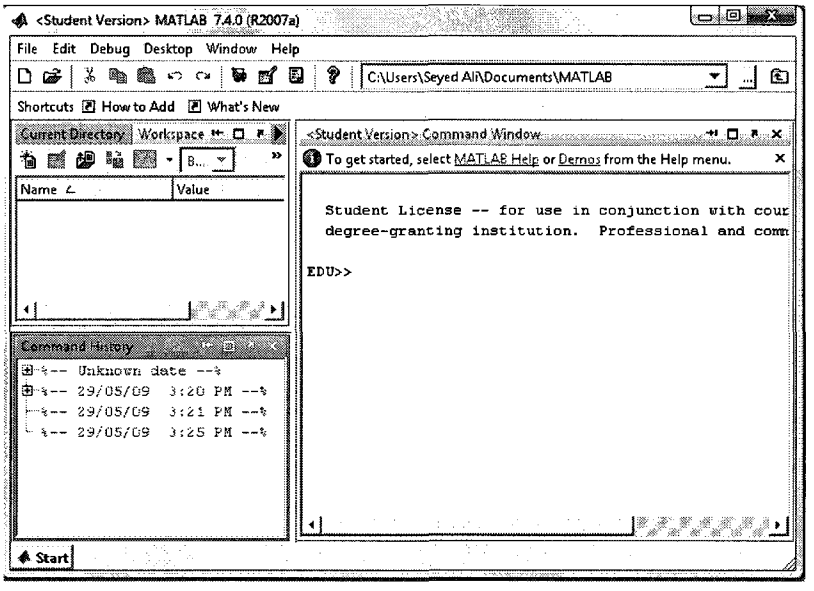

Figure B.1. MATLAB environment

While MATLAB possesses many advantages such as predefined libraries and functions, debugging options, great computational power and accuracy, and user-friendliness, the only drawback remains the computational speed. In cases dealing with recognition and classification (Chapter 4), the processing of the data happens to be quite lengthy and cumbersome. Synthesis of the data (Chapter 4 and Chapter 6), however, is significantly faster than the former two tasks. Synthesis of motion capture data is performed in two different ways which is described in Chapter 6. The technique which uses neural networks is far from real-time, while the second method (model) is carried out almost in real-time.

Appendix C

BVHacker (http://davedub.co.uk/bvhacker/) is a small open source program which is used for visualizing .BVH format files. As illustrated in Figure C.1 this program provides a variety of different options in a very user-friendly environment. Options such as cropping, knitting, changing the frame rate, zooming in and out, converting the data into translation values instead of the original rotation values, manipulating different frames, and adjusting the original skeleton are all made available by this software.

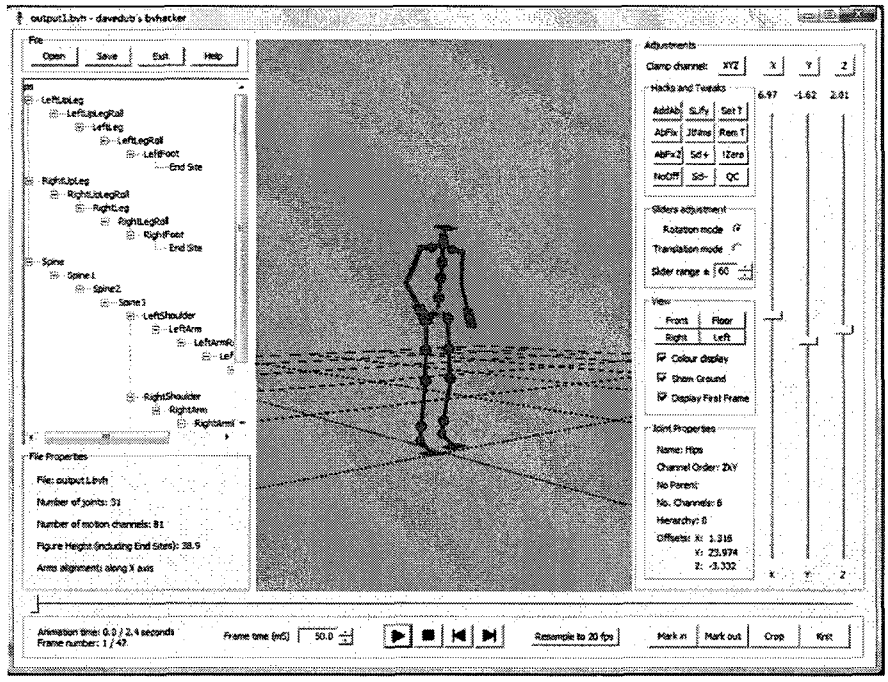

Figure C.1. BVHacker environment

We have used BVHacker for two purposes in our research. The first application of this software is cropping down large .BVH files into smaller sequences. In order to simplify the capture sessions, instead of various capture sessions, few long files have been captured, which are then cropped and segmented using this software to separate smaller files used in the system. The other application of this software is to visualize the output of our processed data without having to import the resulting .BVH files into Maya which is very time consuming. The output .BVH files are simply read and visualized by BVHacker for qualitative visual evaluation of our results.

Appendix D

Successive to processing of the data in MATLAB and initial qualitative evaluation of the results by BVHacker, an animation toolkit has been created in Autodesk Maya by another graduate student (Paul Slinger) as an independent but related project for applying the processing results to different skeletons and generating procedural animation.

Autodesk Maya is a node-based 3D modeling and animation software suite. Maya provides a scripting language called Maya Embedded Language (MEL) that allows direct access to the wide range of animation tools. This accessibility has made Maya (and MEL) an ideal implementation platform for procedural animation methods.

The toolkit creates a simple Graphical User Interface (GUI) that provides users with a range of animation controls. Organized sequentially, the controls are presented in four steps of our procedural pipeline:

- Step 1 allows the user to select which existing character set in the scene file, if there is more than one, to apply the new procedural animation. The dropdown menu produces a list of all the existing character sets in the scene graph and choosing an option will select the desired character hierarchy to be animated.
- Step 2 provides a utility to keyframe each joint for every frame, but can be optional depending on whether these keys already exist for the given character set. Set keys are required in order for new joint rotation values to be retained.
- Step 3 holds the primary procedures of the animation toolkit. There are two separate dropdown menus that can be used to select the type of animation (walk,

run, or jump) and a corresponding animation variation (Masculine to Feminine, Young to Old, or Tired to Energetic). After selecting the two options from the dropdown menu, the user can use the slider control to apply a weight value between the two extremes of the variation. The automatic update button takes the weight percentage of the variation on the right-hand of the slider to the variation on the left-hand of the slider and applies it to the joint rotations character set, producing a new animation.

- Step 4, provides further control to the user by allowing them to apply any additional manual adjustment values to the newly animated character. The user can apply new rotation values and weighting for a single specific joint (or for all joints) at each individual frame (or all frames) by using the available controls.

In addition to the four primary steps, the user is also presented with visual feedback and animation playback controls. These include visual feedback to the user regarding the current frame being processed and ultimately the total elapsed time for the finished process and the standard Maya playback (Play/Stop, Skip to the End/Beginning and Step Forward/Back a frame). Figure D.1 is a screenshot of the Animation Toolkit GUI in its initial start-up state.

Using the Animation Toolkit, a user is able to select an animation type and apply a range of variations in order to derive a new animation sequence. As an example, if a user requires a walk sequence with a more masculine distinction, the toolkit is able to facilitate this action. By selecting the Walk type and the "Masculine ↔ Feminine" variation from the two dropdown menus in Step 3, as well as a minimum weight value (i.e. 0.0), the resulting animation is a masculine walk cycle. Adjusting the weighting to a

maximum value (i.e. 1.0) results in the output of a feminine walk cycle. A user is able to select weight values between the minimum and maximum values that result in a unique mix between the two extremes. Selecting a weight value in the middle (i.e. 0.5) produces neither, a gender-neutral mix between the two variations.

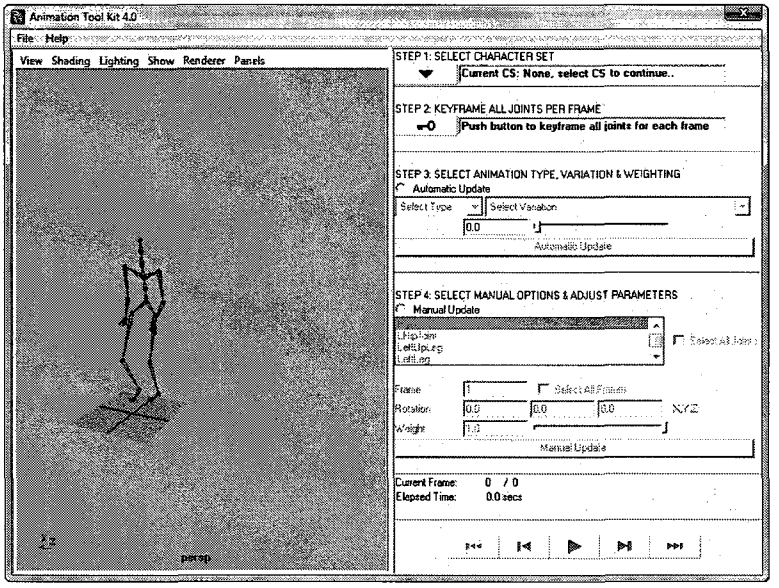

Figure D.1. Animation Toolkit GUI

The evaluation of the toolkit involves (1) allowing users (animators) to interact with the GUI and provide feedback and (2) comparing the procedurally generated animation with motion captured ones performing the same actions. Preliminary results show that the toolkit can be successful in creating a range of animated behaviours with an acceptable similarity to motion capture data methods.

Future extensions to the animation toolkit could include an increase in complexity and number of animation types and variations included in the database to include more uncommon animation types, an expansion of the additional manual controls (to include translation, blending, and other animation options), and translating our existing method from Maya to a designated gaming framework such as Microsoft's XNA for runtime execution.

References

[1] A. Hutchinson, Labanotation. Dance Books, 1996.

[2] D. W. Murray and B. F. Buxton, Experiments in the machine interpretation of visual motion. Cambridge: MIT Press, 1990.

[3] D. A. Forsyth, O. Arikan, L. Ikemoto, J. O'Brien, and D. Ramanan, Computational Studies of Human Motion: part 1, Tracking and Motion Synthesis, Foundation and trends in computer graphics and vision, Now Publishers, vol 1, no 2/3, pp. 77 – 254, 2005.

[4] N. Ikizler, G. R. Cinbis, and P. Duygulu, "Human action recognition with line and flow histograms", *19th IEEE International Conference on Pattern Recognition*, 2008, pp. 1 – 4.

[5] A. Fathi and G. Mori, "Action recognition by learning mid-level motion features", in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1 – 8.

[6] S. Ali and M. Shah, "Human Action Recognition in Videos Using Kinematic Features and Multiple Instance Learning", this paper appears in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Accepted for future publication, ISSN: 0162-8828.

[7] M. Ahmad and S. W. Lee, "Human Action Recognition Using Multi-View Image Sequences Features", in *7th IEEE International Conference on Automatic Face and Gesture Recognition*, 2006, pp. 523 – 528.

[8] X. Li, "HMM based action recognition using oriented histograms of optical flow field" in *IEEE Electronics Letters*, Volume 43, Issue 10, 2007, pp. 560 – 561.

[9] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, "Recognizing Action at a Distance", *Proceedings of the 9th IEEE International Conference on Computer Vision*, Vol. 2, 2003, pp. 726 – 733.

[10] G. Zhu, C. Xu, Q. Huang, and W. Gao, "Action Recognition in Broadcast Tennis Video", *18th International Conference on Pattern Recognition*, Vol. 1, 2006, pp. 251 – 254.

[11] T. Zhao and R. Nevatia, "3D tracking of human locomotion: a tracking as recognition approach", *Proceedings of the 16th International Conference on Pattern Recognition*, Vol. 1, 2002, pp. 546 – 551.

[12] H. L. Zhu, P. Y. Du, and J. Xiang, "3D Motion Recognition based on Ensemble Learning", *8th International Workshop on Image Analysis for Multimedia Interactive Services*, 2007, pp. 28 – 28.

[13] S. Zhang, M. H. Ang, W. Xiao, and C. K. Tham, "Detection of activities for daily life surveillance: Eating and drinking", *10th International Conference on e-health Networking, Applications and Services*, 2008, pp. 171 – 176.

[14] J. Yamato, J. Ohya, K. Ishii, "Recognizing human action in time-sequential images using hidden Markov model", *Proceedings 1992 IEEE Conference on Computer Vision and Pattern Recognition*, 1992, pp. 379 – 385.

[15] Y. C. Wu, H. S. Chen, W. J. Tsai, S. Y. Lee, and J. Y. Yu, "Human action recognition based on layered-HMM", *2008 IEEE International Conference on Multimedia and Expo*, 2008, pp. 1453 – 1456.

[16] X. Li and K. Fukui, "View Invariant Human Action Recognition Based on Factorization and HMMs", *IEICE Transactions on Information and Systems*, Vol. E91-D, Issue 7, 2008, pp. 1848 – 1854.

[17] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing Human Actions: A Local SVM Approach", *Proceedings of the 17th International Conference on Pattern Recognition*, Vol. 3, 2004, pp. 32 – 36.

[18] P. Kornprobst, T. Vieille, and I. K. Dimo, "Could early visual processes be sufficient to label motions?", *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks*, Vol. 3, 2005, pp. 1687 – 1692.

[19] R. V. Babu and K. R. Ramakrishnan "Recognition of human actions using motion history information extracted from the compressed video", *Image and Vision Computing*, Vol. 22, Issue 8, 2004, pp. 597 – 607.

[20] Y. Kuniyoshi and M. Shimozaki, "A self-organizing neural model for context-based action recognition", *First International IEEE EMBS Conference on Neural Engineering*, 2003, pp. 442 – 445.

[21]M. Shimozaki and Y. Kuniyoshi, "Integration of spatial and temporal contexts for action recognition by self organizing neural networks", *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 3*, 2003, pp. 2385 – 2391.

[22] T. Theodoridis and H. Huosheng, "Action classification of 3D human models using dynamic ANNs for mobile robot surveillance", *2007 International IEEE Conference on Robotics and Biomimetics*, 2007, pp. 371 – 376.

[23] F. Lv and R. Nevatia, "Single View Human Action Recognition using Key Pose Matching and Viterbi Path Searching", *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1 – 8, June 2007.

[24] S. Ali, A. Basharat, and M. Shah, "Chaotic Invariants for Human Action Recognition", *11th IEEE International Conference on Computer Vision*, pp. 1 – 8, Oct. 2007.

[25] M. Brand and A. Hertzmann, "Style Machines", *Proceedings of the 27th International Annual Conference on Computer Graphics and Interactive Techniques*, pp. 183 – 192, 2000.

[26] E. Hsu, K. Pulli, and J. Popovic, "Style Translation for Human Motion", *ACM SIGGRAPH '05*, pp. 1082 – 1089, 2005.

[27] C. Rose, M. F. Cohen, and B. Bodenheimer, "Verbs and Adverbs: Multidimensional Motion Interpolation", *IEEE Computer Graphics and Applications*, Vol. 18, Issue 5, pp. 32 – 40, Sept./Oct. 1998.

[28] C. Thurau and V. Hlavac, "Pose primitive based human action recognition in videos or still images", *IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1 – 8.

[29] S. Cherla, K. Kulkarni, A. Kale, and V. Ramasubramanian, "Towards fast, view-invariant human action recognition", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, June 2008, pp. 1 – 8.

[30] D. Y. Chen, S. W. Shih, and H. Y .M. Liao, "Human Action Recognition Using 2-D Spatio-Temporal Templates", *IEEE International Conference on Multimedia and Expo*, July 2007, pp. 667 – 670.

[31] V. M. Zatsiorsky, Kinematics of Human Motion. Champaign, Illinois: Human Kinetics Publishers, 1998.

[32] S. A. Etemad and A. Arya, "3D Human Action Recognition and Style Transformations Using Resilient Back-propagation Neural Networks", submitted to *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS 2009)*.

[33] S. A. Etemad, P. Payeur, and A. Arya, "Automatic Temporal Location and Classification of Human Actions based on Optical Features", Accepted for presentation in the *2nd IEEE International Conference on Image and Signal Processing (CISP'09)*.

[34] P. Slinger, S. A. Etemad, and A. Arya, "Intelligent Toolkit for Procedural Animation of Human Behaviours", *2009 ACM Future Play*, Vancouver, Canada.

[35] S. A. Etemad and A. Arya, "Recognition and Synthesis of 3D Human Motion with Personalized Variations", *Proceedings of 2009 IEEE International Conference on Multimedia Computing and Systems*, Ouarzazate, 2009.

[36] L. Wang, "Abnormal Walking Gait Analysis Using Silhouette-Masked Flow Histograms", *18th International Conference on Pattern Recognition*, Vol. 3, 2006, pp. 473 – 476.

[37] T. Kanade and B. Lucas, "An iterative image registration technique with an application to stereo vision", *1981 International Joint Conferences on Artificial Intelligence*, pp. 674 – 679, 1981.

[38] Y. Kong, X. Zhang, Q. Wei, W. Hu, and Y. Jia, "Group Action Recognition in Soccer Videos", *19th International Conference on Pattern Recognition*, 2008, pp. 1 – 4, Dec. 2008.

[39] P. Natarajan and R. Nevatia, "View and Scale Invariant Action Recognition Using Multiview Shape-Flow Models", *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1 – 8, June 2008.

[40] R. Ishiyama, H. Ikeda, and S. Sakamoto, "A Compact Model of Human Postures Extracting Common Motion from Individual Samples", *Proceedings of the 18th International Conference on Pattern Recognition*, Vol. 1, pp. 187 – 190, 2006.

[41] D. Kulic, W. Takano, and Y. Nakamura, "Incremental on-line hierarchical clustering of whole body motion patterns", *16th IEEE International Symposium on Robot and Human interactive Communication*, pp. 1016 – 1021, Aug. 2007.

[42] M. Shimosaka, T. Mori, T. Harada, and T. Sato, "Marginalized Bags of Vectors Kernels on Switching Linear Dynamics for Online Action Recognition", *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 3072 – 3077, April 2005.

[43] M. Shimosaka, T. Nishimura, Y. Nejigane, T. Mori, and Tomomasa Sato, "Fast Online Action Recognition with Boosted Combinational Motion Features", *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5851 – 5858, Oct. 2006.

[44] T. Mori, M. Shimosaka, and T. Sato, "SVM-Based Human Action Recognition and Its Remarkable Motion Features Discovery Algorithm", Springer Tracts in Advanced Robotics, Springer Berlin/Heidelberg, Vol. 21/2006, pp. 15 – 25, 2006.

[45] V. Parameswaran and R. Chellappa, "View Invariants for Human Action Recognition", *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 613 – 619, June 2003.

[46] A. Madabhushi and J. K. Aggarwal, "Using head movement to recognize activity", *Proceedings of the 15th International Conference on Pattern Recognition*, Vol. 4, pp. 698 – 701, Sept. 2000.

[47] F. Zhou, F. De la Torre, and J. K. Hodgins, "Aligned Cluster Analysis for Temporal Segmentation of Human Motion", *8th IEEE International Conference on Automatic Face & Gesture Recognition*, pp. 1 – 7, Sept. 2008.

[48] Y. Song, L. Goncalves, and P. Perona, "Learning Probabilistic Structure for Human Motion Detection", *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 771 – 777, 2001.

[49] T. Mori, Y. Nejigane, M. Shimosaka, Y. Segawa, T. Harada, and T. Sato, "Online Recognition and Segmentation for Time-Series Motion with HMM and Conceptual Relation of Actions", *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3864 – 3870, Aug. 2005.

[50] Y. Sheikh, M. Sheikh, and M. Shah, "Exploring the space of a human action", *10th IEEE International Conference on Computer Vision*, Vol. 1, pp. 144 – 149, Oct. 2005.

[51] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering Non-Rigid 3D Shape from Image Streams", *Proceedings of the 2000 IEEE Conference of Computer Vision and Pattern Recognition*, Vol. 2,pp. 690 – 696, June 2000.

[52] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams under Orthography", *International Journal of Computer Vision*, Vol. 9, Issue 2, pp. 137 – 154, Nov. 1992.

[53] M. Brand and V. Kettnaker, "Discovery and segmentation of activities in video", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, Issue 8, pp. 844 – 851, 2000.

[54] X. Li and K. Fukui, "View Invariant Human Action Recognition Based on Factorization and HMMs", *IEICE - Transactions on Information and Systems*, Vol. E91-D, Issue 7, pp. 1848 – 1854, July 2008.

[55] L. M. Tanco and A. Hilton, "Realistic synthesis of novel human movements from a database of motion capture examples", *Proceedings of Workshop on Human Motion*, pp. 137 – 142, July 2000.

[56] Y. Li, T. Wang, and H. Y. Shum, "Motion Texture: A Two-Level Statistical Model for Character Motion Synthesis", *ACM Transactions on Graphics*, Vol. 21, Issue 3, pp. 465 – 472, July 2002.

[57] A. Egges, T. Molet, and N. Magnenat-Thalmann, "Personalised Real-time Idle Motion Synthesis", *Proceedings of the 12th Pacific Conference on Computer Graphics and Applications*, pp. 121 – 130, Oct. 2004.

[58] C. K. Liu and Z. Popovic, "Synthesis of Complex Dynamic Character Motion from Simple Animations", *Proceedings of the 29th International Annual Conference on Computer Graphics and Interactive Techniques*, pp. 408 – 416, 2002.

[59] A. C. Fang and N. S. Pollard, "Efficient Synthesis of Physically Valid Human Motion", *ACM Transactions on Graphics*, Vol. 22, Issue 3, pp. 417 – 426, July 2003.

[60] K. Pullen and C. Bregler, "Motion Capture Assisted Animation: Texturing and Synthesis", *ACM Transactions on Graphics*, Vol. 21, Issue 3, pp. 501 – 508, July 2002.

[61] A. Safonova, J. K. Hodgins and N. S. Pollard, "Synthesizing Physically Realistic Human Motion in Low-Dimensional, Behavior-Specfic Spaces", *ACM Transactions on Graphics*, Vol. 23, Issue 3, pp. 514 – 521, Aug. 2004.

[62] A. Shapiro, Y. Cao, and P. Faloutsos, "Style Components", *Proceedings of Graphics Interface*, pp. 33 – 39, June 2006.

[63] E. Hsu, M. da Silva, J. Popovic, "Guided Time Warping for Motion Editing", *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 45 – 52, 2007.

[64] V. B. Zordan and N. C. Van Der Horst, "Mapping Optical Motion Capture Data to Skeletal Motion Using a Physical Model", *Proceedings of the 2003 Eurographics/SIGGRAPH Symposium on Computer Animation*, pp. 245 – 250, 2003.

[65] W. Geng and G. Yu, "Reuse of Motion Capture Data in Animation: A Review", *The 2003 International Conference on Computational Science and Applications*, LNCS 2669, pp. 620 – 629, Jan. 2003.

[66] S. Cooper, A. Hertzmann, and Z. Popovic, "Active Learning for Real-Time Motion Controllers", *ACM Transactions on Graphics*, Vol. 26, No. 3, Article 5, July 2007.

[67] G. Wen, Z. Wang, S. Xia, and D. Zhu, "From Motion Capture Data to Character Animation", *Proceedings of the ACM Symposium on Virtual Reality Software & Technology*, pp. 165 – 168, Nov. 2006.

[68] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions", *Journal of the ACM*, Vol. 45, Issue 6, pp. 891 – 923, Nov. 1998.

[69] Z. Ghahramani, "An introduction to hidden Markov models and Bayesian networks", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 15, Issue 1, pp. 9 – 42, 2001.

[70] L. R. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition", *Proceedings of the IEEE*, Vol. 77, Issue 2, pp. 257 – 286, Feb. 1989.

[71] G. D. Jr. Forney, "The viterbi algorithm", *Proceedings of the IEEE*, Vol. 61, Issue 3, pp. 268 – 278, March 1973.

[72] M. Elmezain and A. Al-Hamadi, "Gesture Recognition for Alphabets from Hand Motion Trajectory Using Hidden Markov Models", *2007 IEEE International Symposium on Signal Processing and Information Technology*, pp. 1192 – 1197, Dec. 2007.

[73] L. J. Luotsinen, H. Fernlund, and L. Boloni, "Teamwork recognition of embodied agents with hidden Markov models", *2007 IEEE International Conference on Intelligent Computer Communication and Processing*, pp. 33 – 40, Sept. 2007.

[74] C. M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1996.

[75] A. Blum, Neural Networks in C++: an object-oriented framework for building connectionist systems, NY: Wiley, 1992.

[76] K. Swingler, Applying Neural Networks: A Practical Guide, London: Academic Press, 1996.

[77] M. J. A. Berry and G. Linoff, Data Mining Techniques, NY: John Wiley & Sons, 1997.

[78] D. Chester, "Why two hidden layers are better than one", *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 265 – 268, Jan. 1990.

[79] N. Wanas, G. Auda, M. S. Kamel, and F. Karray, "On the optimal number of hidden nodes in a neural network", *IEEE Canadian Conference on Electrical and Computer Engineering*, Vol. 2, pp. 918 – 921, May 1998.

[80] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the RPROP algorithm", *1993 IEEE International Conference on Neural Networks*, Vol. 1, pp. 586 – 591, March 1993.

[81] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the RPROP algorithm", *1993 IEEE International Conference on Neural Networks*, Vol. 1, pp. 586 – 591, March/April 1993.

[82] Y. K. Ham and R. H. Park, "3D Object Recognition in range images using hidden Markov models and neural networks", *Pattern Recognition*, Vol. 32, pp. 729 – 742, 1999.

[83] E. Besdok, "Neurovision with Resilient Neural Networks", Advances in Visual Information Systems, Springer Berlin/Heidelberg, Vol. 4781/2007, pp. 438 – 444, Nov. 2007.

[84] H. Wang, "Nearest neighbors by neighborhood counting", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, Issue 6, pp. 942 – 953, June 2006.

[85] D. J. Sturman, "A Brief History of Motion Capture for Computer Character Animation", *Character Motion Systems, SIGGRAPH 94*, Course 9. (http://www.siggraph.org/education/materials/HyperGraph/animation/character_animation/motion_capture/history1.htm)

[86] T. Cloete and C. Scheffer, "Benchmarking of a full-body inertial motion capture system for clinical gait analysis", *2008 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4579 – 4582, Aug. 2008.

[87] T. Harada, T. Mori, and T. Sato, "Human posture probability density estimation based on actual motion measurement and eigenpostures", *2004 IEEE International Conference on Systems, Man and Cybernetics*, Vol. 2, pp. 1595 – 1600, Oct. 2004.

[88] S. Yabukami, H. Kikuchi, M. Yamaguchi, K. I. Arai, K. Takahashi, A. Itagaki, and N. Wako, "Motion capture system of magnetic markers using three-axial magnetic field sensor", *IEEE Transactions on Magnetics*, Vol. 36, Issue 5, Part 1, pp. 3646 – 3648, Sept. 2000.