UNIVERSITY OF SOUTHAMPTON

FACULTY OF PHYSICAL AND APPLIED SCIENCES

Electronics and Computer Science

Model-Based 3D Gait Biometrics

by

Gunawan Ariyanto

Thesis for the degree of Doctor of Philosophy

March 2013

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL AND APPLIED SCIENCES
Electronics and Computer Science

Doctor of Philosophy

MODEL-BASED 3D GAIT BIOMETRICS

by Gunawan Ariyanto

Gait biometrics has attracted increasing interest in the computer vision and machine learning communities because of its unique advantages for recognition at distance. However, there have as yet been few gait biometric approaches which use temporal three-dimensional (3D) data. Clearly, 3D gait data conveys more information than 2D gait data and it is also the natural representation of human gait as perceived by humans. The University of Southampton has created a multi-biometric tunnel using twelve cameras to capture multiple gait images and reconstruct them into 3D volumetric gait data. Some analyses have been done using this 3D dataset mainly to solve the view dependent problem using model-free silhouette-based approaches. This thesis explores the potential of model-based methods in an indoor 3D volumetric gait dataset and presents a novel human gait features extraction algorithm based on marionette and mass-spring principles.

We have developed two different model-based approaches to extract human gait kinematics from 3D volumetric gait data. The first approach used a structural model of a human. This model contained four articulated cylinders and four joints with two degrees of rotational freedom at each joint to model the human lower legs. Human gait kinematic trajectories were extracted by fitting the gait model to the gait data. We proposed a simple yet effective model-fitting algorithm using a correlation filter and dynamic programming. To increase the fitting performance, we utilized a genetic algorithm on top of this structural model. The second approach was a novel 3D model-based approach using a marionette-based mass-spring model. To model the articulated human body, we used a stick-figure model which emulates marionette's motion and joint structure. The stick-figure model had eleven nodes representing the human joints of head, torso and lower legs. Each node was linked with at least one other node by spring. The voxel data in the next frame had a role as an attractor which able to generate forces for each node and then iteratively warp the model into the data. This process was repeated for successive frames.

Our methods can extract both structural and dynamic gait features. Some of the extracted features were inherently unique to 3D gait data such as footprint angle and pelvis rotation. Analysis on a database of 46 subjects shows an encouraging correct classification rate up to 95.1% and suggests that model-based 3D gait analysis can contribute even more in gait biometrics.

Contents

List of Figures

List of Tables

Declaration of Authorship

I, Gunawan Ariyanto , declare that the thesis entitled *Model-Based 3D Gait Biometrics* and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as: [2] and [3]

Signed:.....

Date:.....

Acknowledgements

First of all, I would like to thank my current employer, Universitas Muhammadiyah Surakarta in Indonesia, for their grateful financial support of my PhD study.

I would like to express my special thanks to my supervisor, Professor Mark S. Nixon, for his incredible and continued supports, guidance and patience during my research and towards the completion of this thesis. I owe sincere thankfulness to my examiners, Dr John N. Carter and Professor Andrew David Marshall, who provided encouraging and constructive feedbacks.

I wish to acknowledge the help from my PhD colleagues in CSPC group, especially computer vision & machine learning students, for all the discussions and helps.

Finally and specially, I would like to say huge and deepest thanks for my parents and my family. To my wife and my children who accompany me here in UK: "Sorry for all the pain I have shared with you during my doctoral study. Hope the experiences can make us grow better as a family."

To My Parents and My Family

Chapter 1

Context and Contributions

1.1 Introduction

In modern life, we often perform various authentication actions such as login into our personal computer, withdraw cash from a teller machine, or enter a country through an immigration check. The most common methods to authenticate people are through the use of identification documents, password or PIN number. Whilst these methods are popular and easily implemented, they are also prone to error. In fact, cases of identity theft, lost, forgotten, or misplaced ID and password are commonplace. For example, there are still a huge number of credit/debit card fraud losses every year. In the UK alone, the figure of card fraud losses is around £400m per year since 2007 [82]. In term of security, many intruders have been allowed into a country illegally using faked documents. The surrogate representations of identity based on credentials (passwords, or ID cards) no longer suffice for many applications including those which need a greater degree of security, automation, speed and efficiency. We need another authentication approach that cannot be misplaced, forgotten or easily forged.

In some scenarios which involve a large volume of people, such as immigration checks at airports, both speed and security are the two important considerations when implementing a system for passenger identity check. In the recent decade, many immigration agencies have deployed another layer of authentication security by using biometrics. This technology introduces automatic comparison between fingerprint, face and iris images of current subjects and their stored images in the system database. The overall average time of passenger verification process is then reduced while increasing the quality of security in the airport.

Whilst passenger numbers continue to increase, fast and mass scale biometric technology are needed soon. Many researchers have tried to tackle this matter by either improving the current biometric modalities or by starting to explore other new biometric modalities.

For example, new iris on the move technology has been developed recently to enable fast and convenience biometric authentication process [50]. In search of new biometric modality, gait biometrics has potential to satisfy many of the performance requirements.

The Oxford Dictionary defines Gait as "A person's manner of walking". Gait is one of the behavioural types of biometrics. The main advantage of gait over other biometric modalities is that it can be deployed at a distance where other biometrics are at too low resolution, or are obscured. Recording human gait is also non-invasive and easy to set up in public area. Gait is hard to disguise and it is difficult to obscure gait without impeding movement. Moreover, the sensor of camera is able to cover many targeted gait's objects at a time thus it makes gait biometrics capable to be implemented in high throughput environments.

1.2 Gait Biometrics

Human gait can be classified as a behavioural trait that is impacted by the musculo-skeletal structure of the human body [24, 70, 87]. It is a repetitive motion with a gait period as illustrated in Figure 1.1. A Gait period involves stance and swing which are phases of the legs' motion. The stance phase starts with heel strike (HS) and ends with toe-off (TO). Similarly but conversely, swing phase starts with toe off and ends with heel strike. The difference between running and walking gait is that in walking at least one foot is always in contact with the floor, whilst in running there is a time when neither foot is in contact with the floor.

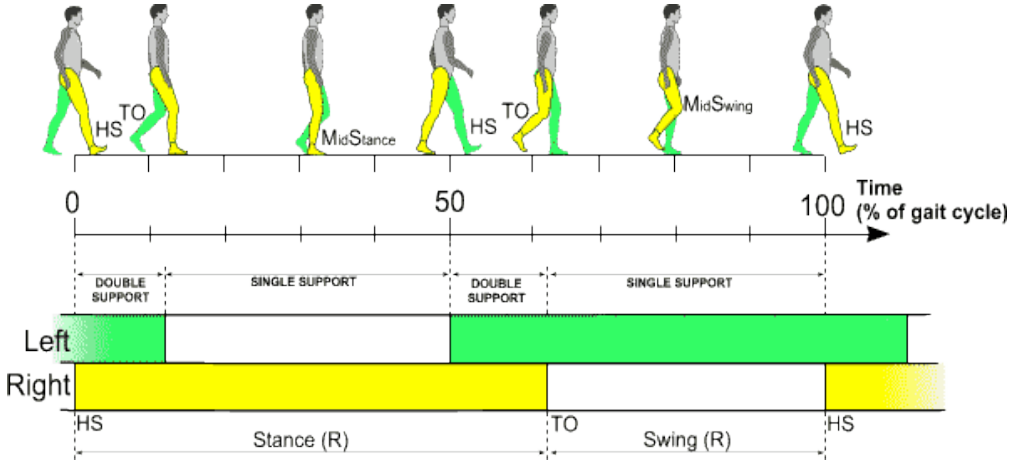


Figure 1.1: Human gait period with stance and swing phases [24]

Gait has been a long time to be a subject of study of Psychologists. Psychological studies and some classic literature have shown that humans are capable of deducing gender and recognizing known individuals based on gait. Psychological experiments were conducted using moving light displays to investigate the gait perception and gender recognition by human vision [38, 42, 51]. In literature, the great writer Shakespeare

had made several references to the individuality of gait in his work. For example in a play of The Tempest [Act 4 Scene 1], Ceres observes "Highst Queen of state, Great Juno comes; I know her by her gait". Other areas of intensive study in gait are in biomechanics and medicine. These fields usually use gait in analysis applications such as automatic diagnosis of orthopaedic patients (understanding the normal and pathological gait) or analysis and optimization of an athlete's performance. However, it has also been suggested in biomechanics that gait is unique if all gait movements are considered [58, 59]. Inspired by all those studies, computer scientists have been developing automatic computer-vision based gait recognition or gait biometrics.

Generally, biometrics refers to the automatic identification of humans by their traits. Biometrics can identify people by measuring some aspects of their physiological or behavioural characteristics. Gait biometrics means human identification based on their walking manner. Identification of people by gait is a challenging problem in recent decades and has gained significant attention from many computer vision researchers due to its unique benefits compared to other existing established biometric modalities.

Methods in gait biometrics generally can be classified as model-free or model-based approaches [63]. In model-free approaches, some methods only use the moving shape whilst others combine the shape and the motion. Model-free approaches also heavily depend on the extracted silhouette and statistical methods. On the other hand, model-based approaches can use prior information of the human structure or a known model (such as a gait motion/kinematic model, or a physical pendulum model) to emulate human gait. Model-free approaches are generally simpler in computation and so far there have been more model-free techniques than model-based ones. Even though model-based approaches are computationally more complex, they can have advantages such as immunity to noise, slight change of view and the effects of clothing.

Most research in gait biometrics has been conducted using 2D datasets and 2D approaches. Although working in 2D is simple and relatively faster in computational time, there are several limitations for most 2D gait recognition systems. One of the significant limitations is viewpoint dependence problem. The signature produced by many 2D gait analysis methods varies with the orientation of the subject relative to the camera. One of the alternative solutions to tackle these common problems is by using 3D gait dataset and 3D approaches.

Though interest in gait biometrics continues to increase, there have as yet been few approaches which use model-based algorithms with temporal 3D data. The reasons why only few studies concerned 3D gait biometrics is perhaps due to complexity and lack of publicly available 3D dataset. However, modern computing power has made it possible to investigate gait biometrics using 3D data with 3D approaches. There are several benefits of using a 3D gait dataset. Clearly, 3D gait data conveys more information than 2D data. It is also the natural representation of human gait as human has two eyes

and can sense depth to reconstruct 3D images. Moreover, 3D data are inherently view invariant as we can synthesize any view by a simple projection. It has been challenging to recognize gait at an arbitrary pose and one of the best potential solutions is by using 3D data and 3D methods. Therefore, it is important to explore the practical advantages of 3D approaches. It is also believed that 3D approaches might provide a more effective way to handle latent issues in 2D such as occlusion, noise, scale and varying view.

Although a 3D approach has many benefits, it also has some limitations. The accuracy of 3D gait reconstruction algorithms strongly depend on the quality of the extracted human silhouette. Even though more robust background modelling and subtraction algorithms have been developed, reconstruction of 3D gait data from outdoor environments remains a challenging task. In term of computation, the 3D method needs much resource which tends to be less practical for real-time applications.

In this thesis we propose to explore the potential of using 3D model-based methods in an indoor 3D volumetric gait dataset. We hypothesize that by using 3D data we can explore more unique factors in human gait. Given the 3D volumetric gait dataset, we develop the first 3D model-based method to extract gait features and perform recognition with relatively large number of subjects.

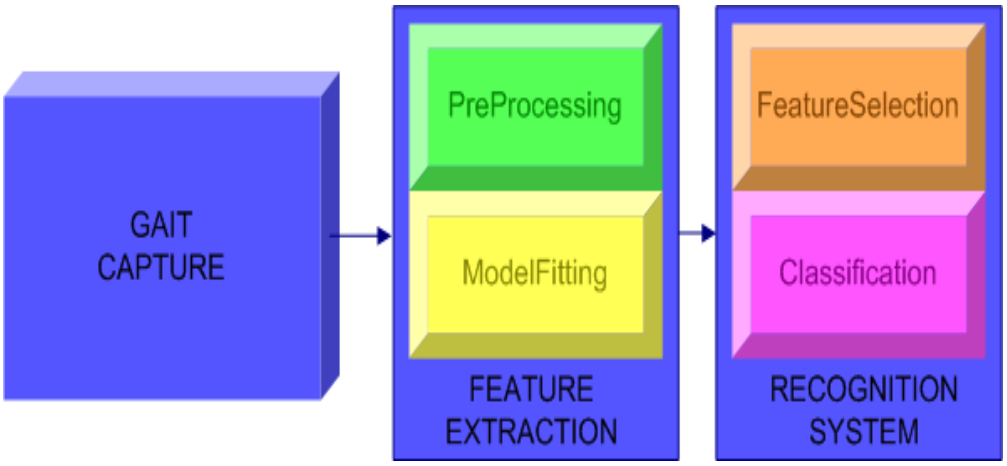


Figure 1.2: Diagram of gait biometric system

A system diagram for our gait biometric analysis is shown in Figure 1.2. There are three stages which cover gait capture, features extraction and recognition. The gait capture has mainly concerned 3D gait dataset from the Southampton multi-biometric tunnel (will be discussed in detail later in Section 2.2). Features extraction mainly concerns gait tracking and pose-estimation to estimate the human gait kinematic trajectories by fitting/warping the gait model into the gait data. Features extraction is discussed in further detail later in chapters 4, 5 and 6. For recognition, the motion kinematic information is used directly using a dynamic time warping or indirectly via a discrete Fourier transform based similarity measurement. We used a k-Nearest Neighbour method to classify the gait features and perform analysis of leave one out cross-validation (LOOCV). We

also investigated all possible combinations of available gait features to extract the best feature-subset.

1.3 Contributions

Despite the increasing literature in gait biometrics, there have been few approaches which have used 3D data and 3D models [7, 63]. A silhouette-based model-free approach has previously achieved high recognition rate on a 3D gait dataset [73], but this approach depends much on clothing or a subject's appearance. Exploring gait features in 3D data with a 3D model can increase the potential of finding significantly unique factors in human gait. The benefit of a model-based approach is that a good model allows for robust and consistent features extraction since features are obtained from human structural information. Hence model-based approaches have an ability to adapt to silhouette distortions arising from variations in camera viewpoint and clothing, or errors in segmentation. So far, only Yamauchi et al. [92] has achieved 3D model-based gait biometrics on 3D laser-range data. This work appears to be the first to conduct model-based gait biometrics analysis and evaluation directly with 3D volumetric data from visual-hull reconstruction, using images acquired by conventional cameras with relatively large number of subjects.

We have developed two different 3D model-based gait tracking approaches to extract the human gait kinematics. The first approach used a human structural model including articulated four 3D cylinders with two rotational degrees of freedom at each joint to model the human lower legs. The cylinders were designed to emulate the lower legs and to be able to extract the kinematic features. We proposed a simple yet effective model-fitting algorithm using the combination of this gait model, anthropometric data and a correlation filter. Human gait kinematic trajectories were extracted by fitting the gait model to the gait data. At each frame we used a correlation filter to generate a correlation energy map between the gait model and the data. The kinematic angles were then extracted based on the minimum value in the energy map. In order to reduce the noise, we have employed a dynamic programming algorithm. Dynamic programming was used to extract the gait kinematic trajectories by selecting the most likely (minimum total energy) path in the whole sequence. It behaved like a low-pass digital filter, removing the high frequency components, and made the extracted kinematic angles look smoother.

The second approach was a novel 3D model-based approach using a marionette and mass-spring model. To model the articulated human body, we used a stick-figure model which emulates the marionette's motion and joint structure. The stick-figure model had eleven nodes, each with three degree of translational freedom representing the head, torso and lower legs' joints. Each node was linked with at least one other node by a

spring. The voxel data in the next frame had a role as attractor which able to generate forces for each node and then iteratively warp the model into the data. This process was repeated for successive frames for one gait period.

Our model-based methods were successfully able to extract both gait static (structural) and dynamic (kinematic) features. Some of the features extracted here, such as footprint angle and pelvis rotation, are inherently unique to 3D data and hardly possible to be generated by ordinary 2D gait data. For evaluation purpose, our analysis used 46 subjects with a total of 184 video sequences. This is considerably larger than that used previously considered for 3D gait biometrics, as Yamauchi et al. used only 6 subjects.

1.4 Thesis Overview

In the next two chapters, we describe some 3D datasets established in this field and report some previous work mainly related to model-based and 3D gait analysis for biometrics. Chapters 4, 5 and 6 discuss the main algorithm and evaluation results of the proposed model-based approaches for 3D gait biometrics. Chapter 4 is about human structural model and the model-fitting process using a correlation filter and dynamic programming. While in chapter 4 we use hierarchical model-fitting process, in chapter 5 we present an improvement of the previous structural model by introducing global fitting with genetic algorithm optimization tool. Chapter 6 describes a novel model-based approach using marionette mass-spring model. We discuss how the marionette model can warp into the human voxel data in each frame. Chapter 7 shows the evaluation results and gives some analysis. We also explain the considerations of using model-based method over the model-free one in this chapter. Finally, in chapter 8 we discuss the conclusions of this work and suggest avenues to further work.

List of Publications

Papers based on this work include:

- Gunawan Ariyanto and Mark S. Nixon. Model-Based 3D Gait Biometrics. In *Proceedings of IEEE International Joint Conference on Biometrics*, October 2011.
- Gunawan Ariyanto and Mark S. Nixon. Marionette Mass-Spring Model for 3D Gait Biometrics. In *Proceedings of IEEE International Conference on Biometrics*, March 2012.

Chapter 2

Gait Datasets

2.1 2D Gait Datasets

Computer vision based gait analysis paper first appeared in 1994 by Niyogi and Adelson [64] which used human motions spatio temporal patterns analysis and described the gait signature from XYT volume. In the next decade, gait analysis and recognition became an attractive field of study. Nixon et al. [63] described the development of gait biometrics since 1990's until 2005. The techniques used in the gait biometrics in those periods were summarised here as shown in Figure 2.1.

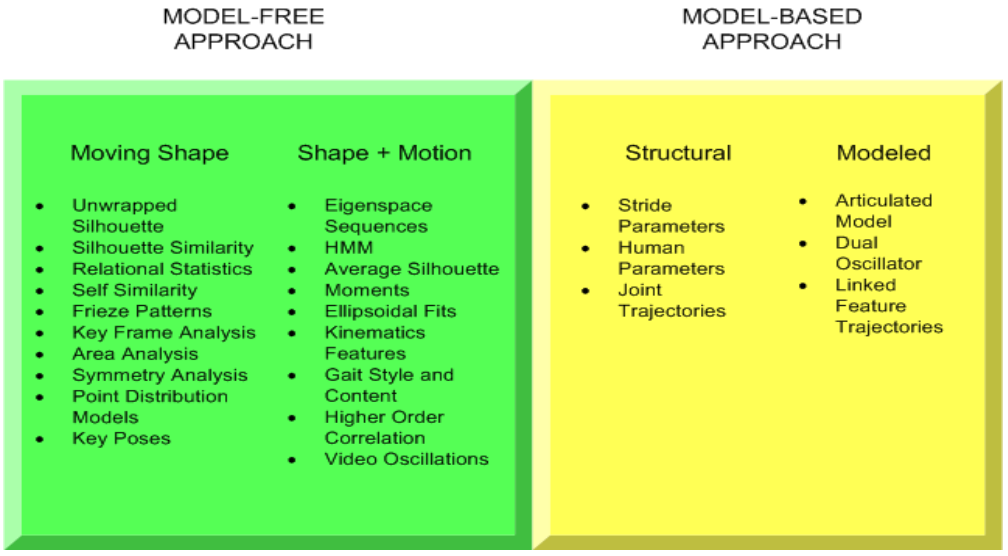


Figure 2.1: Summarized of previous research in gait biometrics from 1990's to 2005

We can classify gait datasets into three different categories: a 2D dataset, a multi-view 2D dataset and a 3D dataset. Even though theoretically we can reconstruct 3D dataset from multi-view cameras, most of the multi-view 2D datasets were recorded on cameras

without time synchronisation and none provided 3D volumetric data. All gait datasets mentioned in this section are 2D and multi-view 2D both with a relatively small number of subjects.

The earliest publicly available gait biometric databases were from University of California San Diego (UCSD) [48] and Southampton University (Soton) [16]. The UCSD dataset was recorded in an outdoor environment with 6 subjects and 42 sequences in total; each subject had 7 sequences. The early Soton gait dataset consisted of 10 subjects and 40 indoor gait samples. Subjects were required to walk frontal-parallel to the image plane. The grey level images were captured in controlled illumination by a fixed camera with a plain, static, cloth background.

There are several other gait databases available publicly for research in gait biometrics such as Soton large dataset [79], CASIA [86, 95], Georgia Tech. [8], CMU [28], MIT [46], UMD [39] and NIST/USF [71, 72]. The Soton, UMD, MIT, CMU and NIST databases were established during DARPA's Human ID at Distance programme (2000-2004). The Soton, UMD, MIT and NIST acquired subjects walking in a sagittal view: Soton was indoor and outdoor data, UMD and NIST were outdoor only, MIT was indoor data (and has been little used since). Of these, the NIST/ USF data has been analysed the most, followed by the Soton data, most likely because these were the largest datasets with over 100 subjects each. The CMU data was multi-view indoor data where subjects walked on a treadmill. The CASIA dataset developed during and after the DARPA programme concerns around 100 subjects viewed using multiple cameras.

Since the DARPA project, some databases have been evolving to be more subjects and variations. The CASIA gait dataset had created three types dataset from 2001 to 2005. The first dataset, namely dataset A, was created in 2001 with only 20 subjects. Dataset B in 2005 was much larger with 124 subjects and multi-views variant. Dataset C was also in 2005 created using infra-red camera and captured at night. The Soton database also has been changing in size and type.

In 2010, Hofmann et al. [34] published new set of gait data focusing on the variation of occlusion and carrying conditions which would frequently occur in real world applications. The camera was set up in a narrow hallway and positioned at a medium height of 1.85 meters with a perpendicular orientation to the hallway direction. The number of dataset was 35 subjects with 840 sample sequences, which means that each subject had 24 samples. Each subject was captured in six different congurations i.e. regular, hand-in-pocket, backpack, gown, static occlusion, and dynamic occlusion. Furthermore, each of the congurations was repeated two times walking right-to-left and the other two times walking left-to-right.

The latest publically available gait dataset with a relatively large number of subjects and samples comes from the University of Osaka in 2010 & 2012. Makihara et al. [49] developed a large-scale gait database comprising a treadmill dataset and then called it

the OU-ISIR gait database. This dataset is a kind of 2D multi-view dataset, focusing on walking condition on a treadmill, that includes 200 subjects with 25 views (OU-ISIR C), 34 subjects with 9 speed variations from 2 to 10 km/h (OU-ISIR A), 68 subjects with at most 32 clothes variations (OU-ISIR B), and 185 subjects with gait fluctuation of speed and cadence (OU-ISIR D). The OU-ISIR dataset also has a greater diversity in gender and age and they suggested that the dataset can be used primarily to evaluate invariant gait recognition. The researchers at Osaka also developed a database to investigate gait recognition on a large database of more than 1000 subjects [65] called the OU-ISIR Large-scale dataset though this database does not yet appear to be publicly available. The study did however reveal recognition performance similar with previous approaches and on their much larger database, on data wherein subjects were recorded in a sagittal view in a chromakey environment walking in a plane normal to the camera view (at an exhibition).

In summary, the MIT, CMU, Soton small, CASIA B, TUM-IITKGP and all OU-ISIR datasets were only recorded indoors while the others contained outdoor video sequences. In term of the gait covariate factors, some datasets had several variations such as the number of viewpoints (single/multiple), time, speed, surface, shoe, clothing and carrying conditions. CASIA B, CASIA C, Soton large, NIST/USF, OU-ISIR B, OU-ISIR C, OU-ISIR D and OU-ISIR Large-scale datasets had large number of subjects while the others only had 55 or less subjects. Table 2.1 give a resume of those 2D gait datasets which are publicly available and have been used by many researchers.

Database	Num. of subjects	Num. of sequences	Environment	Time	Variations
UCSD	6	42	Outdoor	1998	-
MIT AI	24	194	Indoor	2001	View, time
Georgia Tech.	20	188	Outdoor, indoor, magnetic tracker	2001	View, time, distance
CMU Mobo	25	600	Indoor, treadmill	2001	6 viewpoints, speed, carrying condition, incline surface
HID-UMD (Dataset 1)	25	100	Outdoor	2001	4 viewpoints
HID-UMD (Dataset 2)	55	220	Outdoor	2001	2 viewpoints
Soton Small	12	-	Indoor, green chroma-key backdrop	-	Carrying condition, clothing, shoe, view
Soton Large	115	2,128	Indoor, outdoor, treadmill	2001	View
Gait Challenge (NIST/USF)	122	1,870	Outdoor	2001	2 viewpoints, surface, shoe, carrying condition, time
CASIA (Dataset A)	20	240	Outdoor	2001	3 viewpoints
CASIA (Dataset B)	124	13,640	Indoor	2005	11 viewpoints, clothing, carrying condition
CASIA (Dataset C)	153	1,530	Outdoor, at night, thermal camera	2005	Speed, carrying condition
TUM-IITKGP	35	840	Indoor, hallway	2010	carrying conditions, occlusions
OU-ISIR Large-scale	1,023	2,070	Indoor	2010	2 viewpoints
OU-ISIR A	34	612	Indoor, treadmill	2012	9 speeds
OU-ISIR B	68	2,746	Indoor, treadmill	2012	32 clothes
OU-ISIR C	200	5,000	Indoor, treadmill	2012	25 viewpoints
OU-ISIR D	185	370	Indoor, treadmill	2012	gait fluctuation among periods

Table 2.1: Resume of publicly available 2D gait datasets

2.2 3D Gait Datasets

In the recent years, some researchers have been interested in acquiring a 3D dataset for gait recognition. Seely [76] developed a gait tunnel from which a 3D volumetric gait dataset was acquired using multiple cameras. On the other hand, Bhanu et al. [92] developed an approach to 3D gait recognition using a 3D point cloud gait dataset acquired with a 3D active range scan laser sensor. Another 3D gait dataset was recorded but using new popular depth camera, i.e. Kinect, which able to combine audio, image and depth information [33]. Those 3D gait datasets will be described further in this section.

2.2.1 Southampton multi-biometric tunnel

The current approaches to biometrics at Southampton University use a multi-modal dataset from the Southampton multi-biometric tunnel. The University of Southampton multi-biometric tunnel provides a constrained environment and is designed for use in high throughput environments such as airports. The tunnel is a walk-through environment designed for the collection of large datasets. The multi-biometric tunnel was constructed indoors using controlled lighting to reduce the effects of unwanted shadows. The system was built around a pathway enclosed by walls. The floor and walls had a non-repeated rectilinear pattern for camera calibration purposes. Figure 2.2 shows the construction of the tunnel from the entrance and subjects walk toward the far end of the tunnel.

Figure 2.2: Southampton multi-biometric tunnel from the entrance

The initial concept and the prototype version of Southampton multi-biometric tunnel was built by Middleton et al. and aimed to build a system which employed autonomous non-contact biometrics for maximizing subject throughput, and a self-contained system

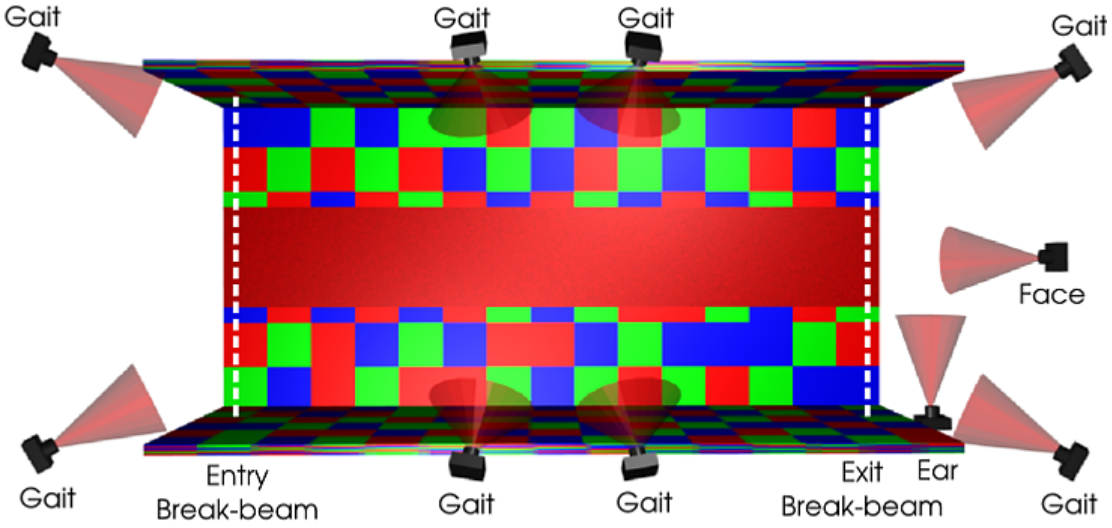


Figure 2.3: Southampton multi-biometric tunnel with 10 cameras [76]

allowing flexible deployment [55]. The tunnel was used to capture a 3D gait and a 2D face dataset and able to be used for automated collection of large amounts of non-contact data in a fast and efficient manner. By contrast, the other gait acquisition methods usually had cooperative subjects, large time capture windows, and required manual editing of the collected video data. Initially, the tunnel used eight synchronised IEEE1394 cameras at 30 fps to capture the gait data and used a single camera to capture face as shown in Figure 2.3. The gait cameras all had a resolution of 640 x 480 and capture at a rate of 30 fps, they were connected together over an IEEE1394 network employing synchronisation units to ensure accurate timing between cameras. As a subject walks through, the tunnel acquires data automatically and it was designed for the collection of very large gait datasets. Using a visual hull shape from silhouette reconstruction algorithm [44], the tunnel was able to produce the 3D volumetric gait data. The shape from silhouette reconstruction is simply the calculation of the intersection of projected silhouettes from each camera.

The current state of Southampton multi-biometric tunnel was built by Seely [74, 75, 76] and replaced the one originally constructed by Middleton. Seely investigated the drawbacks of the previous tunnel by analysing the previously collected dataset. This revealed that the correct classification rate was much lower than expected. With further investigation it was found that there were many corrupt or empty samples present in the dataset; suggesting that the reliability of the prototype system was an issue. Many of the samples in the dataset also had severe artefacts present in the reconstructed data, where the limbs of subjects were severely distorted or even completely missing. The tunnel improvement was achieved by changing the tunnel layout and adding the tunnel hardware with some other four cameras for gait and a new camera to acquire images of the subject's ear. In total, there are now 14 cameras in the current Southampton

multi-biometric tunnel as shown in Figure 2.4, 12 for gait, one for the face (in video) and one for the ear.

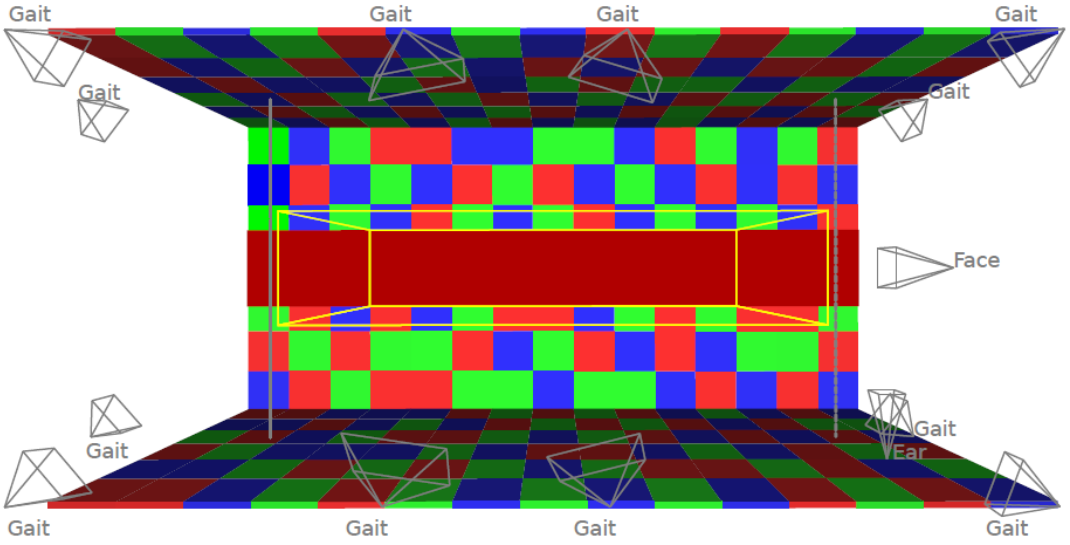


Figure 2.4: The current Southampton multi-biometric tunnel with 14 cameras [76]

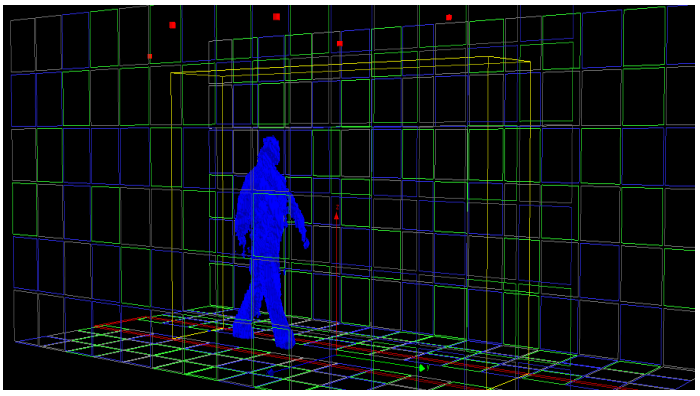
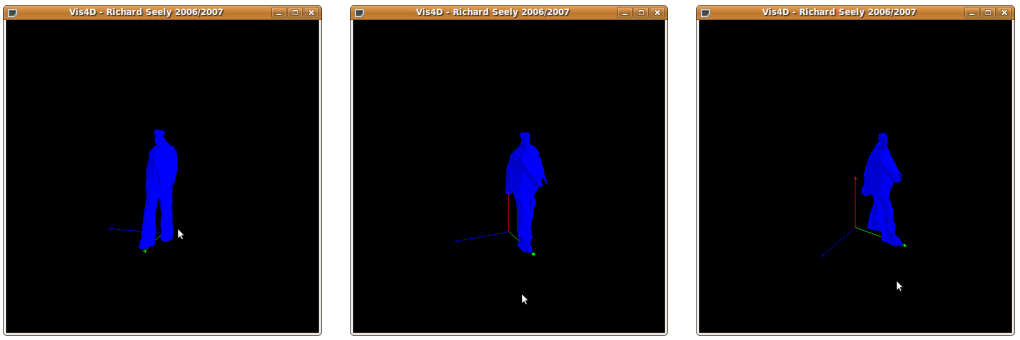


Figure 2.5: Sample data acquired from the tunnel after reconstruction [76]

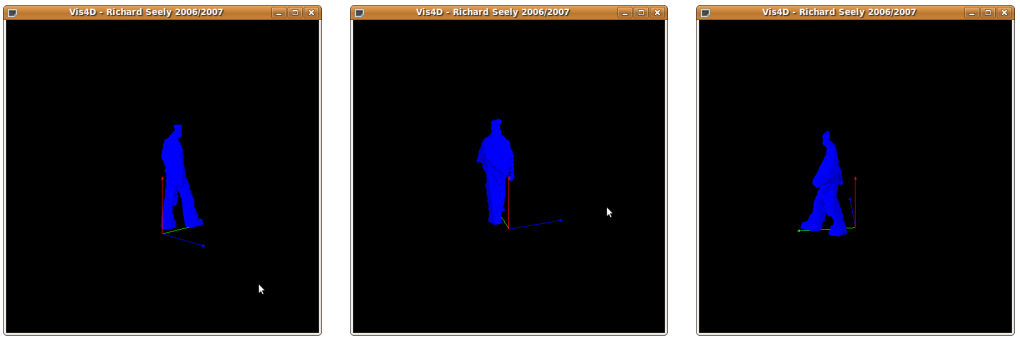
Using visual hull reconstruction [44, 53], the tunnel is able to produce 3D volume gait data. Visual hull or shape from silhouette reconstruction algorithm is simply a calculation of the intersection of projected silhouettes as described in Equation 2.1.

$$V(x,y,z) = \begin{cases} 1 & if \sum_{i=n}^{N} I_n(M_n(x,y,z)) \geq k \\ 0 & otherwise \end{cases} \tag{2.1}$$

Where V is the derived 3D volume, k is the number of cameras required for a voxel to be marked as valid and N is the total number of cameras. $M_n(x,y,z)$ is a mapping function that converts the 3D world coordinates to the coordinate system of camera n. In a conventional implementation of shape from silhouette, a voxel may only be



(a) 1 (b) 2 (c) 3



(d) 4 (e) 5 (f) 6

Figure 2.6: 3D gait dataset shown from different views

considered valid if all cameras have silhouette pixels at its location; therefore $k = N$ must be satisfied.

The 3D gait data produced from the Southampton multi-biometric tunnel is voxel-based data. Its 3D nature allows for viewpoint-invariant gait analysis. Figure 2.5 shows a frame of 3D voxel gait data produced by the tunnel. Some samples of 3D gait data projected into multiple-views are shown in Figure 2.6.

2.2.2 High resolution 3D gait dataset

In contrast to 3D volumetric data extracted from multiple passive cameras, Yamauchi et al. [92] used active range laser scanners (range-finders) to capture the whole human body data with 3 mm depth resolution. Figure 2.7 shows the construction of 3D human body measurement system used by Yamauchi et al. [93]. The 3D human body measurement was achieved using 16 compact range-finder units. Four poles were used to hold all range-finder units; each pole held four units. These poles were arranged around a human as shown in the construction image. The system can generate 3D shape reconstruction with approximately one million measurement points on the entire human body. The data rates are between 2-3 seconds for acquiring the 3D data of the entire human body.

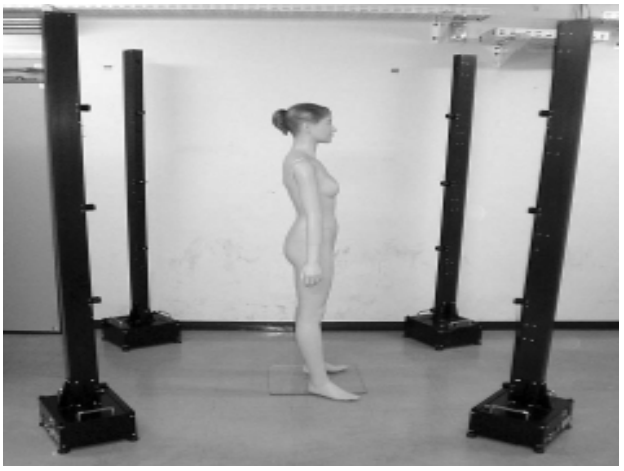


Figure 2.7: Laser rangefinder system for high resolution 3D human body reconstruction

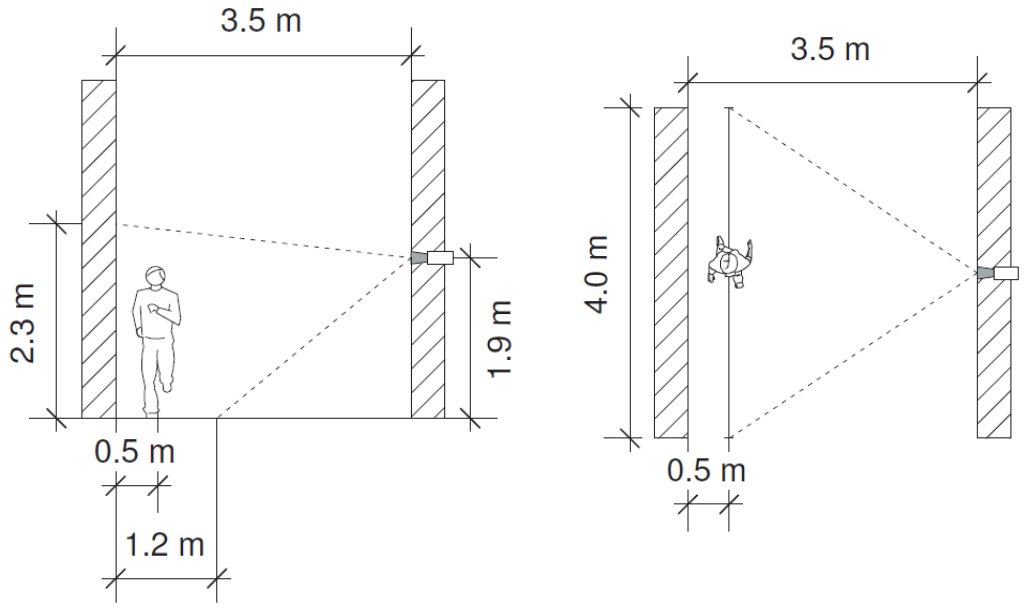
Considering the number of point cloud reconstructed from the system, the 3D gait data collected is dense and at a high resolution which has several advantages compared to the volumetric dataset from the Southampton multi-biometric tunnel. The dense and high resolution datasets have relatively little noise and it is considerably easier to perform any model-fitting in the features extraction phase. However, the dataset contained only six subjects. At that time they planned in the future to expand their dataset and make it publicly available, though this does not appear to have occurred.

2.2.3 Kinect gait dataset

Figure 2.8: Some samples of RGB and depth images from TUM-GAIT dataset [33]

Due to the recent advances in depth imaging devices, Hofmann et al. [32, 33] from Technische Universität München (TUM) published gait dataset for recognition using Microsoft kinect sensor that simultaneously contains RGB video, depth and audio. This database is called gait from audio, image, and depth database (TUM-GAID). The TUM GAID database was created to foster multimodal gait that is why it was recorded with an RGB-D sensor, as well as with a four-channel microphone array simultaneously. The dataset was collected twice in January and April 2012 and contained 305 people in three variations, i.e. backpack, coating shoes, and time. Figure 2.8 shows some samples of the dataset consisting both RGB and depth images.

TUM-GAID dataset was recorded in a place of a 3.5 m wide hallway corridor. Figure 2.9 shows the schematic of the recording site consisting of side and top-down views. A static and solid surface background was implemented in the hallway. The Kinect sensor was placed at 1.9 m high and facing downwards at an angle of roughly 13°. The walking path of the subject was perpendicular to the line of sight at a distance of roughly 3 m close to the opposite wall. Each person in the dataset typically has between 1.5 and 2.5 gait cycles in each recorded sequence with approximately 2–3 seconds.



(a) side view (b) top-down view

Figure 2.9: Schematic of the TUM-GAID dataset's recording site [33]

Chapter 3

Model-based Gait Recognition and on Using 3D Datasets

This chapter presents an analysis and a review of previous research in model-based gait recognition approaches and on using 3D datasets. Even though the model-based approaches are computationally more complex, they have many advantages as opposed to model-free silhouette based approaches. Some of the benefits are that model-based approaches can reliably handle occlusion (especially self-occlusion) and have immunity to noise, slight change of view and the effects of clothing [63].

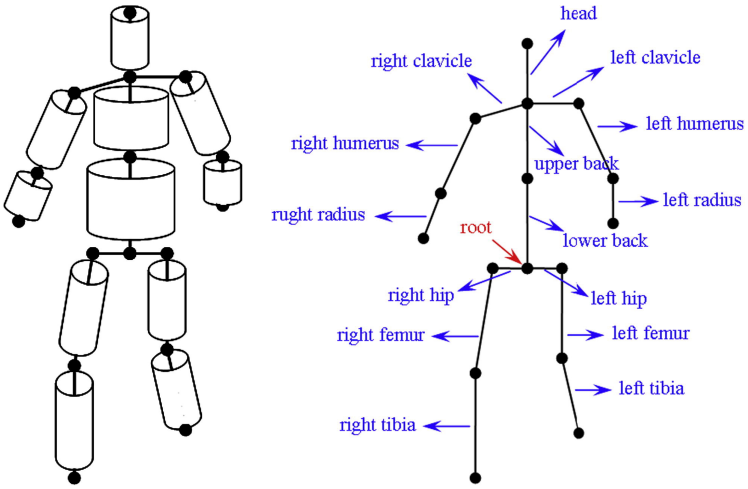
There have been few biometric studies concerning 3D gait using 3D data. Most research in gait biometrics has been conducted with 2D datasets and using 2D approaches even though there are some unique advantages of using a 3D dataset. The 3D representation of human gait can convey more information than in 2D and it is inherently view invariant as we can synthesize any view. Working in 3D can also bring more consistency regarding the occlusion and multi-interpretation problems. Although a 3D approach has many benefits, 3D approaches generally are more complex and need more computational resources. These characteristics make a 3D approach less practical for outdoor and real-time applications.

3.1 Model-based Gait Recognition Approaches

This section reports some gait analysis techniques that use model-based approaches. In model-based gait recognition, the gait model always represents the discriminatory gait characteristics either static or dynamic. This model comes with a set of parameters and a set of logical and quantitative relationships between them. The model's parameters are usually meaningful quantities such as the height of human body, stride length, or kinematic properties such as joint angle trajectories extracted from joint positions. Most

model-based gait recognition methods use kinematic information to aid the recognition process. In order to extract full kinematic information, the model has to be three-dimensional.

Model-based approaches can use prior information of the human structure (human anthropometric data) or a known motion model (such as physical pendulum model) to emulate human gait. A human body can be viewed as an articulated object, consisting of a number of body parts. Human bodies can be represented as stick figures [40, 45, 78], 2D contours, ribbons or volumetric models [35, 83] such as cylinders or tapered cones. Figure 3.1 shows images of some human body models.



(a) 3D cylinder model (b) Stick figure model

Figure 3.1: Some examples of 3D human body model

Niyogi et al. [64] and Guo et al. [29] in 1994 published their gait analysis algorithms which used human models to demonstrate that gait was suitable for recognition purposes. Niyogi et al. made a preliminary study of gait recognition in a spatio-temporal (XYT) volume. They first found the bounding contours of the walking subject, and then fitted a simplified stick model. Then a characteristic gait pattern in XYT was generated from the model parameters for recognition. Guo et al. employed a more complex ten-stick model, which was fitted to a silhouette sequence by calculating a cost field for each silhouette, then finding the set of model parameters that minimised the cost accumulated by the model.

At an early stage in the development of gait biometrics, Cunado et al. [16] reported that it was possible to perform gait recognition using a simple model approximating each leg as a single line segment joined at the hip. The angles of the lines were found for each frame using hough transform [5] and then smoothed and interpolated using polynomial splines. After performing a discrete Fourier transform on the angles, they used the frequency coefficients to achieve recognition. Later, Cunado et al. [15, 17] extended it by using an advanced model. The lower leg model was a pair of articulated pendula

and a structural model. This model was then fitted to the edge feature using a genetic-algorithm based velocity Hough transform. A genetic-algorithm was implemented to perform the search of high dimensional parameter space in the velocity Hough transform. Yam et al. [88, 89, 90, 91] also proposed a pendula model and extended the Cunado work. Their approach could recognize people whilst walking or running.

Lee and Grimson [47] proposed a simple model-based method based on human body part segmentation. The human silhouette was divided into local regions corresponding to different human body parts, and then ellipsoidal models were fitted to each region to represent the human structure. Later, Lee et al. extended the ellipse fitting approach to volumetric data to achieve view invariant gait recognition [46]. They used a multiple camera system and performed three-dimensional visual hull reconstruction. A simple three-stick model was proposed by Bobick and Johnson [8]. This model had three line segments, representing the two limbs and the torso, all connected at the center of pelvis. They used static parameters for recognition such as the distance between the head and pelvis, the pelvis and feet, and between both feet. The results of the approach were validated against ground-truth data acquired from a magnetic sensor system. In the same year, Tanawongsuwan and Bobick [81] used the time-normalized trajectories of lower limb joint angles in sagittal view as the main features for gait recognition. Rather than using marker-less method, they intentionally employed an electro-magnetic motion capture system to obtain accurate data. Abdelkader et al. [1] also proposed an approach using only static features, stride length and cadence, for recognition. To extract the proposed static features, they used an analysis of the variation in the subject's bounding box width. Davis and Taylor [19] used a similar three-stick model for gait analysis but only used basic dynamic features for recognition. They tried to obtain the stance to swing ratio and the double support time data. The feet were located by finding the principal axis of each leg and then took the furthest silhouette pixel's location along the principal axis as the foot position.

Ning et al. [61, 62] employed a model-based approach to recover joint angle's kinematic features of walking people using a human body and motion models in a particle filter framework. In recognition process, they mentioned that the correct recognition rate and equal error rate (EER) using the kinematic features are better than the results using static features extracted from model-free method based on statistical shape analysis. Wang et al. [85, 86] later extended that work by introducing more robust pose evaluation function of model-fitting and then evaluating the fusion of static and dynamic features.

Yoo et al. [94] estimated hip and knee angles from the body contour using trigonometric-polynomial interpolate functions. The gait description was derived by topological analysis guided by medical studies that selects areas from which joint angles are derived by regression analysis. Wagg and Nixon [84] extended the work of Cunado et al. by representing the head and torso by a pair of ellipses and each leg consisted of two pairs of line segments, for the thigh and the shin. They fitted the model into the data over

multiple stages of fitting due to huge computation. First the velocity of the subject was estimated, then a bounding region surrounding the subject was established and refined to consist of three primitives. The proposed method had been tested using an indoor dataset from the University of Southampton and achieved good correct classification rate over 80%. The other model-based approach was proposed by Bouchrika and Nixon [10]. They proposed a method that exploiting the subject's heel strike information in order to reduce the complexity of model-fitting.

3.2 Previous Work on Using 3D Datasets for Gait Recognition

The earliest use of the 3D gait dataset for human recognition was by Shakhnarovich et al [77]. A small dataset of 3D volumetric visual hull gait dataset was created using 12 subjects with the number of samples per subject varying between two and eight. Visual hull reconstruction was done using silhouettes generated from four video cameras. 2D canonical view silhouettes were synthesised by placing a virtual camera into the volumetric space. Two dimensional gait analysis techniques were then used to analyse the synthesised silhouettes. This approach allowed the use of 2D gait analysis techniques with view-invariant data.

Seely [75, 76] et al. developed the Southampton multi-biometric tunnel as described in the section 2.2 and created the Soton 3D gait dataset. He also successfully used the dataset for view-invariant gait recognition using model-free analysis. He employed simple 2D average silhouette-based methods using view point projection techniques to convert 3D data into 2D view-invariant data. There were three different view-point projections used: side-on, front-on and top-down projection. Using dataset of 103 subjects with 1030 samples, the results showed that using 3D gait data can lead to high accuracy (99.6%) and the best performance was achieved by using a combination of projected views [75]. Another study by Zhang et al. [96] has similar data to that of the Soton 3D dataset, but used it for human tracking rather than recognition.

Yamauchi et al [7, 92] used an active vision sensor to capture 3D high quality gait data and then fitted the 3D model into the data to obtain the kinematics information. They used a laser range sensor and collected subject poses which represent only the key frames in a gait period. The complete gait sequence was synthesized by interpolation of joint positions and their movements from the fitted body models. The experimental results showed high recognition rates, though at that time there were only six subjects and 24 samples used in the experiments.

The effect of time on gait biometric performance has been studied by Matovski [52] in 2010. Matovski used a subset of the Soton 3D gait dataset with 25 subjects and nearly

2000 samples taken over a 9 month period of time and projected the 3D data to give sagittal views. The results showed that gait can be used as a reliable biometric over time and at a distance. It was also reported that clothing drastically affects performance regardless of elapsed time.

There was other research using 3D models but with 2D multiple-view dataset. Orrite-Uruuela et al [68] proposed a model-fitting of 3D skeletal model to multi-view video data from the CMU database. They used fitting point-distribution models to the silhouette from each frame. The skeletal model was then extracted from the set of point-distribution models after the fitting process. Zhao et al [97] conducted a 3D model-fitting where multiple views were used to improve model-fitting performance. A skeletal model was initially fitted to the first frame in a sequence, with the position, orientation, body geometry and joint angles being manually chosen. Tracking was then performed on the subsequent frames to extract the variation the model's parameters during the walking action.

In the Kinect gait dataset, baseline algorithms were setup by Hofmann et al. [33]. They used four well known appearance-based methods, i.e. the Gait Energy Image (GEI), Gait Energy Image on Depth Data (depth-GEI), Gait Energy Volume (GEV) and Depth Gradient Histogram Energy Image (DGHEI). For the normal dataset, GEI achieved the best performance with a good 94.4% recognition rate. On the backpack and shoes datasets, DGHEI had a better performance than the other algorithms. It achieved 40.3% and 96.1% recognition rate of backpack and shoes, respectively. In the time/temporal dataset experiment, the performance sharply degraded. The best performance for temporal normal, temporal backpack, and temporal shoes datasets were achieved at only 50%, 6% and 9%, respectively. Even depth-GEI and GEV algorithms completely failed with 0% recognition rate for the case of temporal backpack dataset.

There have been approaches to 3D markerless human motion captures (mocap) in which the majority has aimed at gait motion characterisation and action recognition rather than for gait biometrics. In biomechanics study, Mundermann et al. [57] proposed a markerless mocap using a 3D visual-hull dataset and an articulated iterative closest points (ICP) algorithm. They also implemented soft-joint constraints in the tracking process. It was reported that the markerless mocap can accurately extract human gait motions very similar to the established marker-based mocap. Later, Corazza et al. [14] extended this work by introducing a subject specific model which is obtained through an automatic model generation algorithm that combines a space of human shapes with biomechanically consistent kinematic models and a pose-shape matching algorithm.

It is possible that the techniques in 3D markerless mocap in biomechanics could be deployed for biometrics purposes but that has been little achieved. Krzeszowski et al. [43] in 2012 collected 3D markerless human motion capture data and then used the data for gait recognition evaluation. Instead of designing a tunnel, they setup a laboratory

with four cameras where people can walk in any direction inside the room. This kind of laboratory layout need more space than a tunnel, but it has big advantage on the quality of data reconstruction due to flexible multiple camera locations and configurations. They implemented Annealed Particle Swarm Optimization (APSO) in global model-fitting to recovering pose. There were only 10 subjects evaluated for gait recognition with each subject having two walking sequences. Gait classification task was carried out using two different methods, i.e. Naive Bayes and Multi Layer Perceptron (MLP) classifiers. The identification results were about 85% at rank 1.

A common thread to the 3D approaches is the lack of use of an underlying model or model-free, thereby assuming that the data samples focus on human subject, without the discriminating capability of non-human objects. Developing a model based on human gait can address this deficiency.

Chapter 4

Robust 3D Model-Fitting Using a Structural Model

In this chapter we describe a gait tracking method performed directly in 3D space using a model-fitting approach with a structural model. The tracking process was conducted frame by frame. In each frame, the human legs' model was fitted directly in 3D space to the 3D volumetric data. Working in a 3D space generally can bring more consistency, while fitting in 2D domain is more easily affected by self-occlusion. Moreover, the 3D volumetric data can synthesize all information regarding the camera parameters and background subtraction, allowing simpler and more efficient fitting.

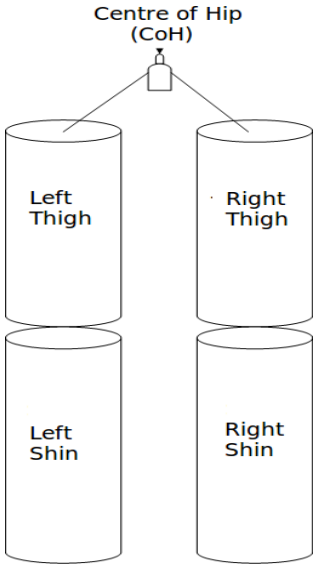
An effective model-fitting process always needs a good model and a fitting algorithm. In this work, the human lower legs were modelled using a structural model consisting of articulated cylinders. It is important to note that in gait motion we only deal with a single type of motion, i.e. walking. Hence, a simple yet effective method of model-fitting was proposed here based on using a correlation filter. In the correlation filter, the cylinder models are correlated with the data points using a Euclidean distance measurement. We also proposed a dynamic programming approach to process the gait sequence as a whole rather than frame by frame thus making it possible to filter noise and produce smooth kinematic trajectories. In this case, the dynamic programming behaved like a low-pass filter.

In order to extract gait kinematic features, we need to perform human lower legs' tracking for at least for one gait period. In the experiments we only processed and extracted the gait features for exactly one gait period on each sample because the 3D gait sequence from Southampton multi-biometric tunnel dataset contains no more than one and a half periods of walking frames.

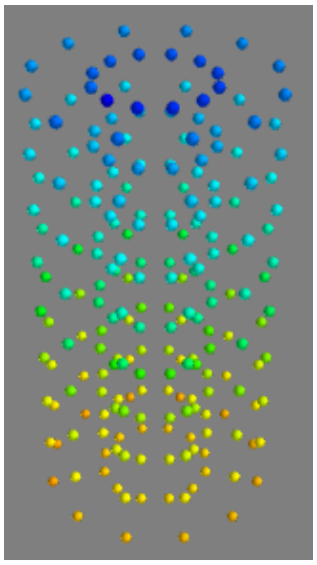
4.1 Assumptions and Prior Knowledge

As previously mentioned, the source of gait dataset used in this work were collected from the Southampton multi-biometric tunnel. Due to the properties of the dataset, the proposed methods in this thesis have the following assumptions:

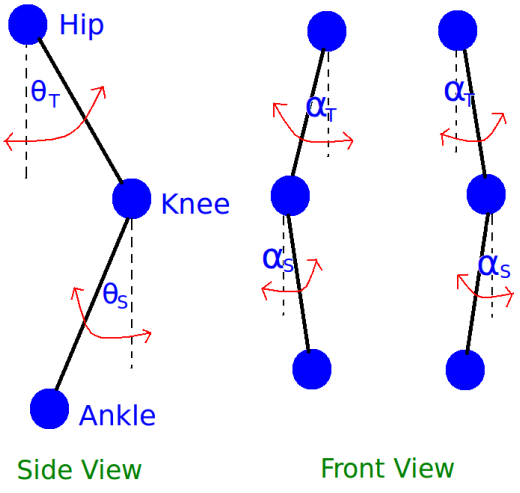
1. Only one subject walks through the tunnel at a time
2. The height's proportions of hip, knee, and ankle of the model are defined from the established human anthropometric data [87] as shown in Figure A



(a) 3D cylinder model



(b) A 3D cyclinder with point cloud inside



(c) The kinematic angles in the model

Figure 4.1: 3D structural model of human gait

4.2 Structural Gait Model

We used articulated 3D cylinders as our gait model. As shown in Figure 4.1, there were four cylinders and four joints for modelling the lower human legs. These cylinders corresponded to the thighs and the shins. Two rotational joints at each leg were used to connect the shin to the thigh (knee joint) and the thigh to the pelvis (hip joint). All joints defined here had two rotational degree of freedom (DoF). Within each cylinder in the model, we generated a regular cloud of points and then a local coordinate frame was defined with the origin point located at the top of the cylinder. The origin point also corresponded to the rotational center of the cylinder. The global coordinate system originated at the center of hip (CoH) which is in the middle of pelvis between the right and left hip joints and it was currently modelled as a single 3D static point. The CoH was static during the model-fitting process and its value/location was derived from the central mass of the human body as described in preprocessing stage later. The length of the cylinders were estimated using subject's height and human anthropometric data [26, 94]. The human anthropometric data are described in Figure A in the appendix.

The gait kinematics were extracted using the thigh and shin trajectory angles of the cylinder gait model. Using the proposed gait model, the rotation of the cylinder in transversal plane will give no effect and can be ignored. We can extract in total up to eight kinematic angles as we only consider two rotation angles in each cylinder: sagittal θ and frontal α. Figure 4.1(c) shows the rotation angles in the sagittal and the frontal plane; T, S, R and L denote the thigh, shin, right and left respectively. Thus θ_{RT} denotes the sagittal angle of the right thigh.

4.3 Preprocessing Stage

A number of preprocessing steps are required before performing any model-fitting process. In this stage, we seek to process a 3D voxel data sequence to obtain 3D bounding boxes which each bounding box has three parameters, i.e. width, height and length. The variation of the 3D bounding box parameters in the dataset sequence can be used to estimate the gait period and the subject's height. All of these gathered information were used to initialise the gait model and model-fitting process.

4.3.1 3D bounding box

Let the voxels representing a person at frame j be $V^j(x, y, z) = \{x_i, y_i, z_i\}$. The bounding box BB is the smallest volume which encloses the connected voxel data. It can be derived from the values of x (frontal/length), y (sagittal/width) and z (transverse/height) axis for a filled voxel as described in Equation 4.1. To find the value of $\min_i x_i$, we slice the

volume space into x-planes and then scan those planes starting from plane $X = 0$ to the maximum number of x-axis; or in the opposite way, starting from the maximum number to 0 for the $\max_i$ operator. The scanning proses will check the number of pixels in the plane and will stop if the pixel number is satisfied. The same proses also applied for other axis, i.e. y and z. In order to filter noise, we set up a specific threshold value when evaluating the number of pixel in each plane. We also found that using the information from the subjects centre of mass (CoM) can validate the correct bounding box and remove the noise. If operator $\min_i$ or $\max_i$ has a value far away from the CoM reference, it will be regarded as noise and then will be discarded and search another valid value.

$$BB_a = \max_i a_i - \min_i a_i \quad \text{where} \quad a \in x, y, z \tag{4.1}$$

Another approach of finding 3D bounding box and its orientation is a convex-hull algorithm [23, 67]. However this approach needs more computational time and our model-based approach did not use the orientation of the bounding box.

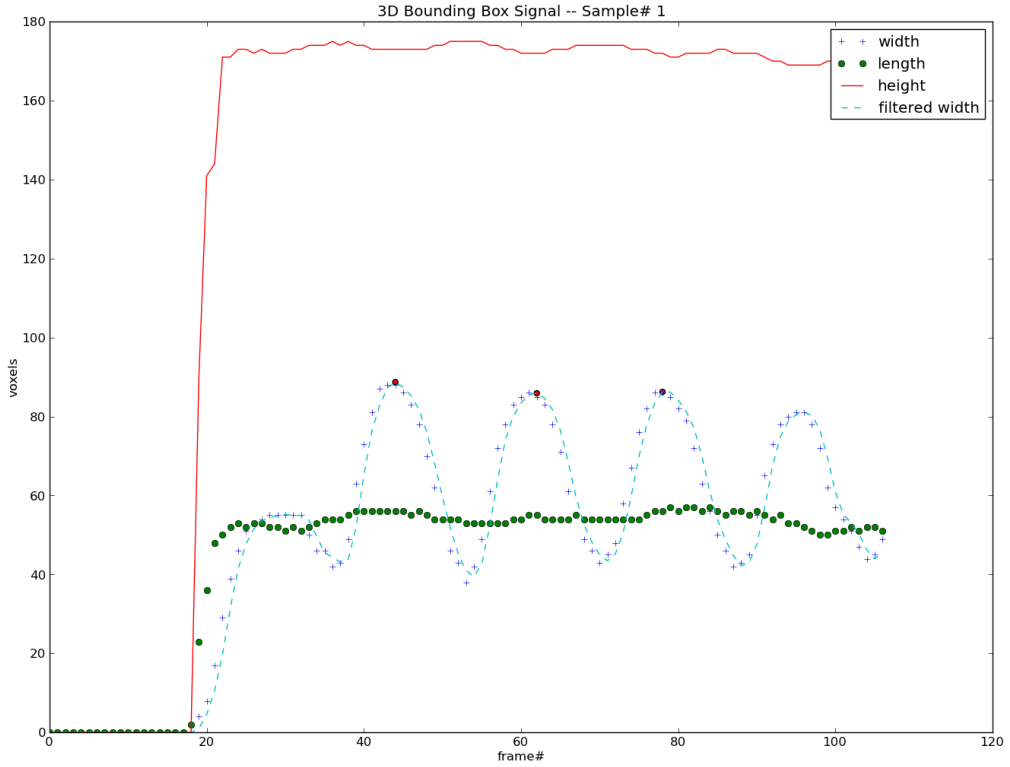


Figure 4.2: The extracted values of width, length, and height of a voxel sequence 3D bounding box

Figure 4.2 shows the size/value of typical bounding boxes of a voxel sequence extracted along the x (width), y (length) and z (height) axis respectively. The figure also shows the filtered width of the bounding boxes of the sequence that will be used for a gait

period estimation. To obtain a filtered width signal, we use a digital low pass filter with a hamming window function.

4.3.2 Gait period estimation

Using the 3D bounding box data of all frames in a sequence, we can determine the starting and ending frame in a gait period. As shown in Figure 4.2, the heel strike pose corresponds with the peak, labelled with the red circle, of the filtered width of the 3D bounding boxes. The start frame is that for which a maximum width bounding box occurs first (frame# 44). The end frame is the second next maximum width (frame# 78). The length of the gait period then can be derived from these two key frames which is 34 frames.

A 3D gait sequence in the Southampton multi-biometric gait dataset typically has around 110 frames. However, from Figure 4.2 we can see that the first 20 frames are just blank or empty voxel data. The subject usually starts at frame 40 and then disappears at frame 90. Therefore, we can obtain a full voxel data of the subject only for 50 frames which is around one and a half of the gait cycle.

Another robust approach for gait period estimation could be achieved by curve-fitting algorithm [9]. Given the raw data of width bounding boxes' signal, we can define a sinusoidal curve and then fit it into the data. The period of this curve give the value of the gait period. However, the current quality of the dataset has allowed us to use the simple method as previously described.

4.3.3 Subject height

The subject's height h is derived from the average of the transverse plane bounding box in one gait period (N frames) of the sequence as shown in Equation 4.2.

$$h = \frac{\sum_{j=1}^{N} BB_z^j}{N} \tag{4.2}$$

4.3.4 Centre of hip

In order to fit the gait model to the voxel data, we first need to find the position of the centre of the hip (CoH) in the voxel data. This CoH will be used as a stationary point of the gait model in the fitting process. We estimate the CoH (c_x, c_y, c_z) by using the mean of the voxel data and the subject height. The formula to obtain the CoH position is described in Equation 4.3 where $I(x, y, z)$ is the value/intensity of the voxel i.e. $\{0, 1\}$,

h is the subject height, and 0.53 is a proportional scale of hip position using anatomical estimates as shown in Appendix A.

$$\begin{aligned} c_x &= \frac{1}{s_x s_y s_z} \sum_{x=1}^{s_x} \sum_{y=1}^{s_y} \sum_{z=1}^{s_z} xI(x,y,z) \\ c_y &= \frac{1}{s_x s_y s_z} \sum_{x=1}^{s_x} \sum_{y=1}^{s_y} \sum_{z=1}^{s_z} yI(x,y,z) \\ c_z &= 0.53 \times h \end{aligned} \tag{4.3}$$

4.4 Model-Fitting Process and Correlation Energy Map

The aim of model-fitting process here is to generate kinematic paths from a voxel data sequence. In each frame, this process involves voxel data, the structural model using cylinder points, and a fitting algorithm. The proposed model-fitting algorithm has two stages, i.e. generating a correlation energy map and using this energy map to find an optimal trajectory in a whole sequence. This section describes how we measure the similarity between the data and the model by generating a correlation energy map. Finding an optimal trajectory will be discussed later in the next section.

The first step in our model-fitting method is a cross-correlation operation between the data and a model to determine where in the data space the model was most likely to occur. Cross-correlation is similar to convolution in that two signals are moved over each other to produce a third signal describing where they best match. In our case, the cross-correlation operation will move cylinder model around the voxel data to produce similarity analysis.

A correlation energy map $\mathbf{E}$ was defined by calculating the cross-correlation for each possible parameter value (frontal angle α and sagittal angle θ) over model points $\{m_i\}$ as described in Equation 4.5. We used the least squares Euclidean distance to measure the correlation energy between the model and the data. The range of θ and α was ± 45 and ± 7.5 degrees respectively. In order to achieve robust model fitting, we performed brute-force algorithm by evaluating all possible pose in the cylinders model.

Let $M = \{m_i\}$ is the 3D thigh/shin cylinder points of the model behaving as "filter" and $D = \{d_i\}$ is the 3D data points derived from the 3D voxel data. The rotation operation of the model has two rotational degree of freedom using sagittal θ and frontal α angles. We use a transformation matrix $\mathbf{T}(\alpha, \theta)$ as in Equation 4.4 to implement this rotation

operation. This allows us to parameterise the correlation process with α and θ.

$$\mathbf{T}(\alpha,\theta) = \begin{pmatrix} cos(\alpha) & 0 & sin(\alpha) \\ 0 & 1 & 0 \\ -sin(\alpha) & 0 & cos(\alpha) \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) \\ 0 & sin(\theta) & cos(\theta) \end{pmatrix} \tag{4.4}$$

$$\begin{aligned} \mathbf{E}[\alpha,\theta] &= \sum_{i=1}^{s_m} dist(m_i \times \mathbf{T}(\alpha,\theta), D)^2 \\ dist(m,D) &= \min_i dist(m,d_i) \end{aligned} \tag{4.5}$$

To choose the resolution of these parameters we need to consider a computational effort and the smoothing process of finding optimal trajectory in the next stage. We chose a step of 6° and 1° for θ and α. By choosing these numbers, we had 90/6 (equals to 15) possible value for θ and also 15/1 (equals to 15) possible value for α. Hence, in one cylinder of each frame we had 15 x 15 poses' resolution. If we increase the resolution twice for each angle parameter, the computation will be four times longer.

The rationale behind the chosen step numbers was also affected by the decision of using dynamic programming in the next step of model-fitting as will be explained in section 4.5. In order to obtain smooth kinematic trajectories, the difference between a pose and the next one should be limited within one or two steps. From biomechanis, we know that the pattern of thigh sagittal θ angle is similar to a sinusoidal curve. The trajectory from the bottom to the top means a half period of walking sequence either as in a swing or a stance phase. Because we had around 33 frames in a gait period and 90° range of sagittal θ angle ($\pm$45°), the minimum step we could achieve for theta angle was 90/16.5 (around 6°). For frontal α angle ($\pm$7.5°), the minimum step was 15/16.5 (around 1°).

Figure 4.3 shows a 3D surface plot from correlation energy map of sample 1 at frame# 65. The darker the blue color means the lower energy value with a better fit. From the figure we can see that a defined correlation function has a minimum value. The variations of energy map values are dominated by sagittal angle θ compared to frontal angle α. This is happen because sagittal angle has more range values, which is from -45 to +45, while the frontal angle ranges from -7.5 to 7.5. We can increase the frontal angle range, but it would make the human leg looks unrealistic.

For each cylinder, we generate the same energy map as described above. In the left and right thigh cylinders, we used the CoH as the reference point to define the origin points of the rotation. For the shin cylinders, we define the origin based on the optimal trajectories determined by the thigh. This means that we extract the thighs' trajectories

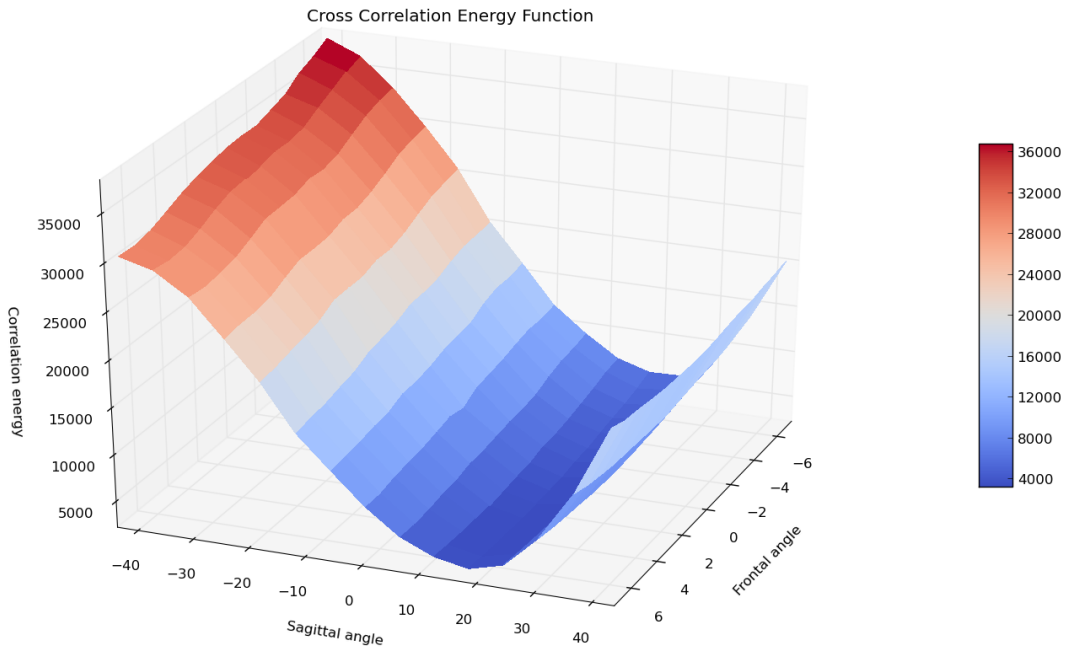


Figure 4.3: Correlation energy map in 3D plot

first before we can do the same for the shins. The detail of this process will be explained later in this chapter in section 4.6.

Appendix B gives the implementation details of this correlation algorithm in Python. We have created a plain correlation map and a multi-resolution correlation map. The multi resolution version is simply a fast version in term of computation.

4.5 Dynamic Programming for Optimal Gait Trajectory Extraction

Since gait is a spatio-temporal activity, we process the gait sequence as a whole rather than frame by frame thus making it possible to filter noise. We can phrase the problem of extracting the best gait angle trajectory as a shortest path problem. For each cylinder in each frame, we have 15 x 15 (equals to 255) possible parameters (α and θ) values. Within one gait period, there are usually around 33 voxel frames. If we try to find the best trajectories from frame# 1 to the end of gait period (frame# 33), it will require 225^{33} possibilities which is impossible to find the solution using a brute-force approach.

We choose dynamic programming to select the optimal trajectory based on the previously generated correlation energy maps. Dynamic programming is a method for solving complex problems by breaking them down into simpler sub-problems. A problem is said

to satisfy the Principle of Optimality if the sub-solutions of an optimal solution of the problem are themselves optimal solutions for their sub-problems.

Dynamic programming has been used in other applications such as in seam carving for image resizing by Avidan and Shamir [4]. A seam is a connected path of low energy pixels in an image. Avidan and Shamir employed dynamic programming to find the minimum seam in a 2D image and then remove their corresponding pixels from the image.

In our problem, a correlation energy map $\mathbf{E}[j,k]$ can be considered as a matrix with row index j corresponding to θ and column index k corresponding to α. The accumulation of this map in one gait period will form a 3D array $\mathbf{E}[i,j,k]$ where i corresponds to the frame number. We transverse the entire map and then compute the cumulative minimum energy $\mathbf{M}(i,j,k)$ for all possible connected links for each entry as shown in Equation 4.6. In this case we use 9-connected links to connect the two energy maps in adjacent frames.

$$\begin{aligned}\mathbf{M}(i,j,k) = \mathbf{E}(i,j,k) + min(&\mathbf{M}(i-1,j-1,k-1),\\ &\mathbf{M}(i-1,j-1,k), \mathbf{M}(i-1,j-1,k+1),\\ &\mathbf{M}(i-1,j,k-1), \mathbf{M}(i-1,j,k),\\ &\mathbf{M}(i-1,j,k+1), \mathbf{M}(i-1,j+1,k-1),\\ &\mathbf{M}(i-1,j+1,k), \mathbf{M}(i-1,j+1,k+1))\end{aligned} \tag{4.6}$$

At the end of this process, the minimum value of the last matrix in $\mathbf{M}$ will indicate the end of the minimal connected trajectories in one gait period. We then backtrack from this minimum cell at the last matrix in $\mathbf{M}$ to find the path of the optimal trajectory.

We consider that using dynamic programming is an efficient way to exploit the temporal information and to reduce noise. The dynamic programming approach works in both spatial and temporal domain to extract the minimal energy path and it can assure that there will be no sudden change in the extracted kinematic angles as such consistent with human motion. The implementation details of this algorithm can be seen in Python programming code at Appendix B.

4.6 Hierarchical Kinematic Features Extraction

We use a hierarchical approach of model-fitting [22] to extract the kinematic features (lower legs trajectory angles). Firstly, we fit the thigh cylinder to the data and then extract the thigh kinematics. Secondly, we use this thigh kinematic data to update the

knee joint and shin origin position. Finally, we fit the shin cylinder to determine the shin kinematics.

The hierarchical approach has an advantage of reducing the optimal pose search space and the complexity of computation. In our case, it is possible to extend the current methods using simultaneous fitting for both thigh and shin. However, we would have at least four parameters in the correlation map and it is difficult to implement the dynamic programming for finding the optimal path due to huge computation time.

4.7 Gait Features, Signatures and Classification

Using the proposed methods, we are able to extract five gait features, i.e. height, stride, footprint angle, centre of hip and lower leg's kinematic angles. These features can be classified as either structural or dynamical. A structural feature relates to the static measurement of the human body. The values of the dynamic features change with time. The dynamic features are an array with length corresponding to the number of frames in a gait period.

4.7.1 Structural features

We seek to extract some structural features of gait using the 3D voxel gait data. So far we are able to estimate three structural gait parameters: height, stride length and footprint angle. The footprint angle can be considered as a new gait feature because it is inherently unique to the 3D markerless gait biometric system.

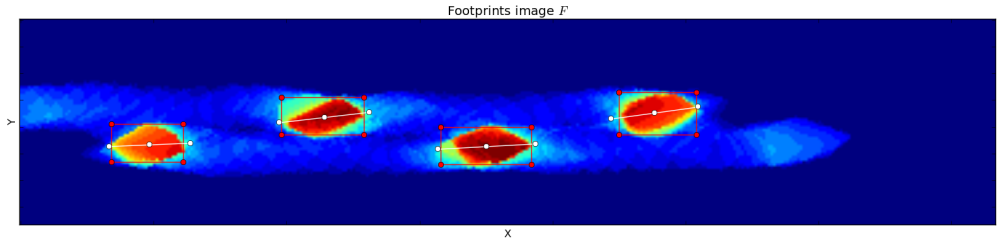
4.7.1.1 Subject height

The subject's height h is derived from the average of the transverse plane of subject bounding box in one gait period (N frames) as described in the preprocessing stage. The formula to estimate a subject height is shown in Equation 4.2.

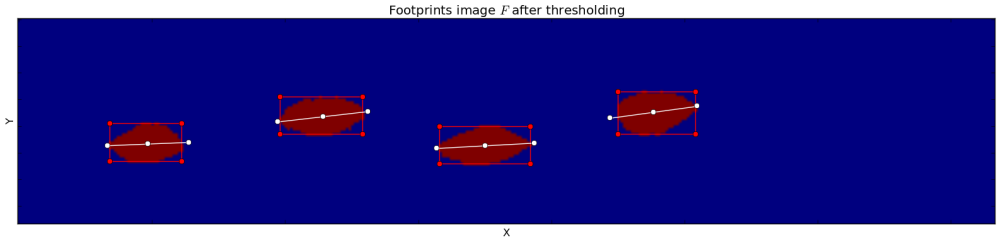
4.7.1.2 Footprint angle

We extract the footprint image using a cross-sectional plane of the voxel data. During the walking period, each heel is in a static position on the ground for around half gait period. Therefore, we project cross-sectional planes at height j ($Z = j$) around the ankle into the ground plane G^j for each frame and then accumulate it to derive the footprints image F as described in Equation 4.7 where S_j is the number of cross-sectional planes. We use Otsu thresholding [69] and closing morphological operation to obtain a clean footprint image and locate the heel strike positions.

$$\begin{aligned} G^j(x,y) &= V^j(x,y,z=j) \\ F(x,y) &= \sum_{j=1}^{S_j} G^j(x,y) \end{aligned} \tag{4.7}$$



(a) Footprint image



(b) Footprint image after thresholding

Figure 4.4: Footprints image F and its footprint angles

Figure 4.4 shows the extracted footprints image before and after thresholding operation. From the figure, we can also see footprint orientation angles imposed into the image. In order to obtain the orientation angle for each footprint, we employ a moment analysis as described in [12] in Equation 4.8. Figure 4.5 shows a single footprint image with its orientation angle β. The red color in the image means high density value while the blue one is low value. The collected footprint angle feature consists of four orientation angles and they are stored in an array labelled as [β_1,β_2,β_3,β_4].

$$\begin{aligned} M_{00} = \sum_x \sum_y F(x,y); M_{10} = \sum_x \sum_y xF(x,y) \\ M_{01} = \sum_x \sum_y yF(x,y); M_{11} = \sum_x \sum_y xyF(x,y) \\ M_{20} = \sum_x \sum_y x^2F(x,y); M_{02} = \sum_x \sum_y y^2F(x,y) \\ x_c = \frac{M_{10}}{M_{00}}; y_c = \frac{M_{01}}{M_{00}} \\ \beta = \frac{arctan\left(\frac{2\left(\frac{M_{11}}{M_{00}} - x_c y_c\right)}{\left(\frac{M_{20}}{M_{00}} - x_c^2\right) - \left(\frac{M_{02}}{M_{00}} - y_c^2\right)}\right)}{2} \end{aligned} \tag{4.8}$$

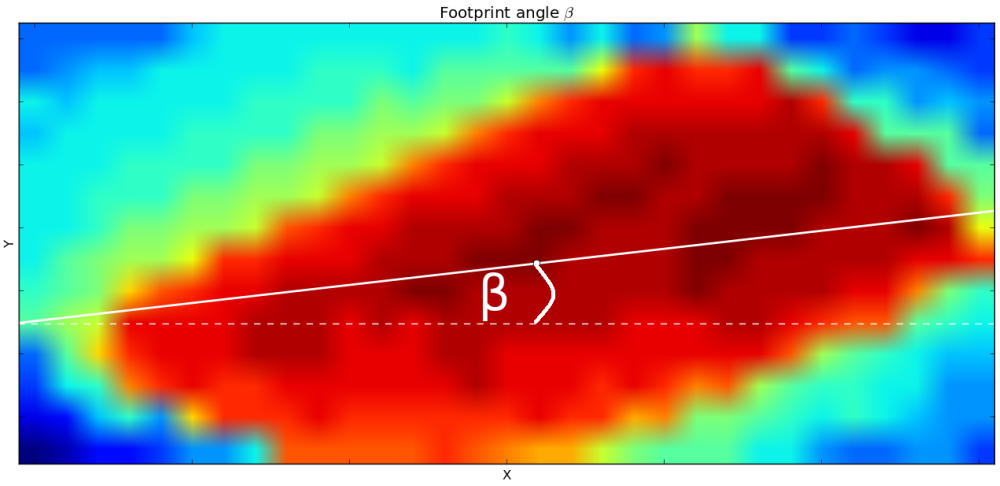


Figure 4.5: A footprint orientation angle β extracted using moment analysis

4.7.1.3 Stride length

Using the footprint information above, we are able to extract the stride length by calculating the distance between the adjacent footprints of the same foot.

4.7.2 Dynamic features

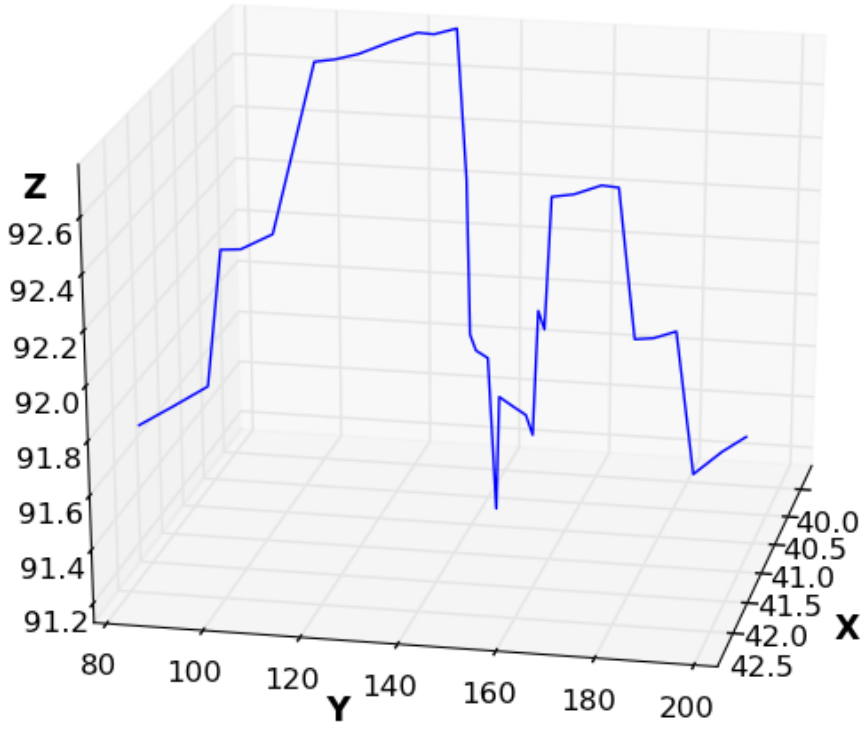
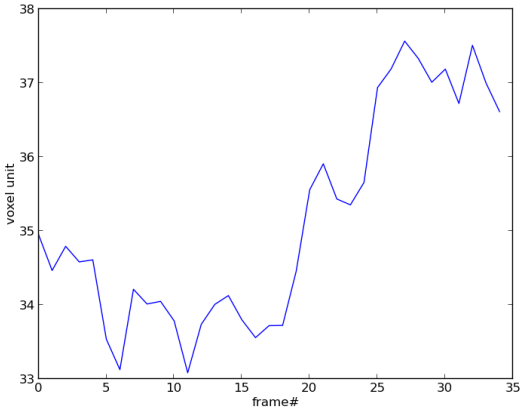
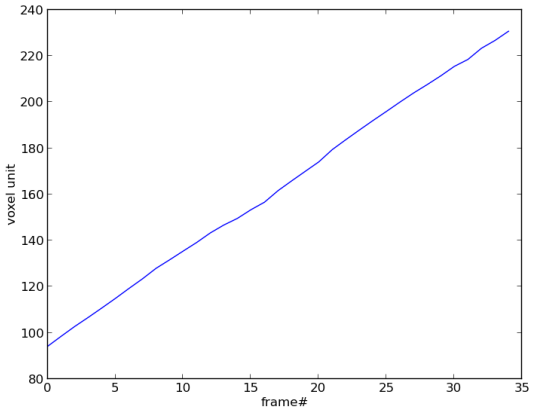


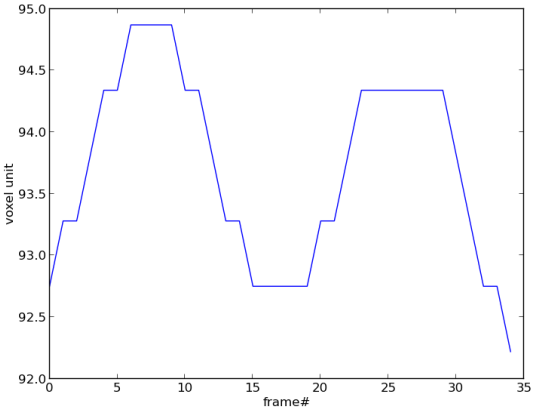
Figure 4.6: Center of hip 3D position in one period of walking sequence



(a) c_x value of CoH



(b) c_y value of CoH



(c) c_z value of CoH

Figure 4.7: Graphs of CoH in each axis

We have developed a model-fitting framework as described in the previous section to collect some dynamic gait features. So far we have collected two kinds of 3D dynamic features, the centre of hip (CoH) and the gait kinematic angles. For each frame in the

gait period, we estimate the CoH (c_x, c_y, c_z) by using the mean of the voxel data and the subject height as described in Equation 4.3. A sample of the extracted CoH is shown in Figure 4.6.

If we plot the CoH into three orthogonal axes relative to time or frame number, we can obtain new three graphs as shown in Figure 4.7. In this figure we can see that the top one is the position of c_x which can tell the direction of walk. Next in the middle is the graph of c_y that represents the distance passed during the walk. The last one at the bottom of the image is the plot of c_z which represent the height of CoH. From these images, we can see that the c_z position is very dynamic and has a kind of sinusoidal pattern.

As previously described, we use a hierarchical approach of model-fitting to extract the kinematic features. There are eight total number of kinematic features extracted so far from the four cylinder model, with each cylinder give two features, i.e. sagittal (θ) and frontal (α) trajectory angles.

4.7.3 Gait signatures

The overall gait signatures are created by selecting and combining the structural and the dynamic features. Table 4.1 shows all available gait features and their gait signature representations used in this experiment including what data type they hold and what distance metric they use for matching operation in classification.

For gait kinematics we extract the gait signature using The Discrete Fourier Transform (DFT) [17] and Dynamic Time Warping (DTW) [11]. After applying the DFT to the gait kinematic data, we will obtain new information about the frequency components of the subject's gait. If we use DFT representation then these frequency components will be used as gait signatures. However, if we use DTW then we will use the kinematic angles directly as the gait signatures. The distance metric of gait kinematic signature is various because it has many metric's functions to compare two sets of frequency components such as Euclidean, magnitude, and magnitude-phase distance functions. All distance functions will be described later in the next section.

The DFT is defined by Equation 4.9 [66]. A series x of N complex numbers is transformed into another series X of N complex numbers. The number X_i represents the i^{th} multiple of the fundamental frequency. We used the Fast Fourier Transform (FFT) with 128 points.

$$X_i = \sum_{k=0}^{N-1} x_k e^{-\frac{2\pi i}{N} ik} \qquad i = 0, \ldots, N-1 \tag{4.9}$$

Feature	**Gait Signature Representation**	**Data Type**	**Distance Metric**
Height	Height	Scalar	Euclidean
Stride	Stride	Scalar	Euclidean
Footprints	Footprints	Array	Euclidean
Centre of Hip	DTW	Array	Euclidean
Gait Kinematics	DFT & DTW	Array	Various

Table 4.1: Gait signatures

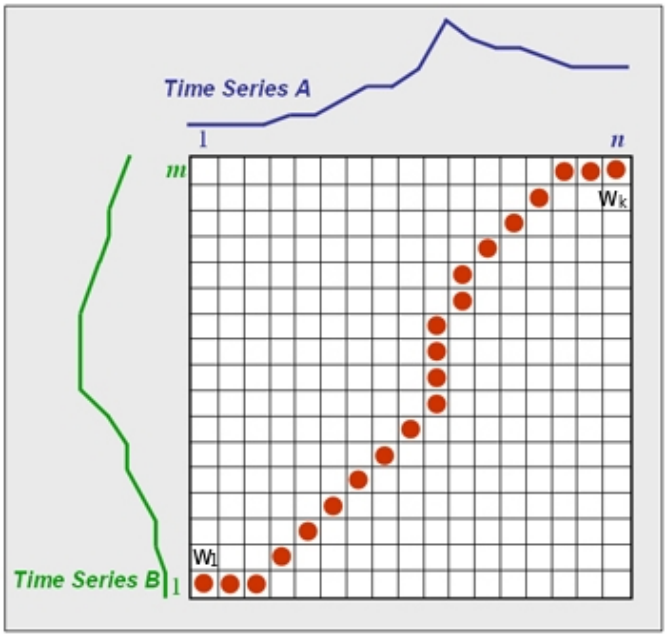


Figure 4.8: Dynamic time warping

The CoH features behave like a time series datasets with 3D values. We use Dynamic Time Warping (DTW) [6] to measure the similarity of the CoH features. DTW involves the creation of a matrix in which the distance between every possible combination of time instances is stored. This distance is calculated in terms of the feature values of the points using Euclidean distance function. Let $A(a_1, a_2, \ldots, a_n)$ and $B(b_1, b_2, \ldots, b_m)$ be two CoH features with length n and m, respectively, and $d(a_i, b_j)$ indicate the Euclidean distance between a_i and b_j. The point-to-point alignment and matching relationship between A and B can be represented by a time warping path $W =< w_1, w_2, \ldots, w_k >$. The corresponding DTW is described in Equation 4.10.

$$\mathrm{DTW}(A, B) = \min_{W} \left\{ \sum_{k} d(a_i, b_j), W \right\} \tag{4.10}$$

4.7.4 Classification and evaluation methods

We classify the subject based on its feature type:

1. Structural features: height, stride length and footprint's angles
2. Dynamic features: CoH and lower legs' kinematic angles

In this work, k-NN was used as a classifier to evaluate the recognition performance of our system. We used k-NN because it is a standard practice and widely used by other publications in gait biometrics. Seely et al. [73] who used the same 3D dataset also employed k-NN classifier. Thus, to some extent we can compare our results with the previous research. Alternative classifiers are available and can be applied which may produce better performance such as SVM, and Bayesian; However, the k-NN classifier suffices to demonstrate good performance. In the k-NN algorithm three different values of k were used: 1, 3 and 5 with leave one out cross validation (LOOCV). LOOCV works by evaluating all of the samples one by one. We pick one sample as a probe, while use the rest of the samples as a gallery. We repeated this proses for every sample in the dataset.

Some experiments had been advised to evaluate the performance of our system in three measurements i.e. recognition rate of static features, recognition rate of dynamic features, and ROC/CMC curves. Generally, all of the evaluations are basic in that more advanced approaches such as biometric fusion or more sophisticated classifiers could putatively improve performance.

Firstly, we want to evaluate the structural features performance. We evaluate the extracted structural features individually using k-NN classification. In order to evaluate the benefits of 3D features (footprint angle), we compare the performance of a combination of the height-stride and the height-footprint.

Secondly, we want to analyse the performance of our model-based system by evaluating the dynamic features which are the kinematics and CoH. For the kinematic gait signature with the DFT spectral components, we use three different distance functions of the magnitude and of the product of magnitude and phase. The distance functions between kinematic gait signatures $\mathbf{a}$ and $\mathbf{b}$ are:

- **Magnitude**. In the magnitude distance, we compare the magnitude of each frequency component.

$$d_m(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{N/2} \left(|a_i| - |b_i|\right)^2 \tag{4.11}$$

- **Magnitude Weighted Phase**. The magnitude weighted phase distance will use phase information along with magnitude. This distance function was also used by Cunado et al. [16].

$$d_{mp}(\mathbf{a},\mathbf{b}) = \sum_{i=1}^{N/2} \left(|a_i| \arg a_i - |b_i| \arg b_i\right)^2 \tag{4.12}$$

- **Euclidean**. In Euclidean distance, we compare both the real and imaginary parts directly from the coefficients.

$$d_e(\mathbf{a},\mathbf{b}) = \sum_{i=1}^{N/2} \left(\Re a_i - \Re b_i\right)^2 + \left(\Im a_i - \Im b_i\right)^2 \tag{4.13}$$

We also compared the kinematic features directly using DTW distance as described previously in the Equation 4.10. In the classification of the CoH, we projected the 3D CoH values into three orthogonal axes, x, y and z, and then evaluated its recognition rate one by one using DTW distance function.

In order to determine the maximum discriminating power of the kinematic features, we conducted a recognition test for all possible combination of the kinematic features subsets. Because we have eight kinematic features, we can have 254 possible combinations of the feature subsets. We also evaluated the overall performance of our system using several combinations of structural and dynamic features.

In the kinematic feature evaluation as described above, we have also used five different methods of data normalization. The methods are *minmax*, *decimal scaling*, *Z-score*, *median absolute deviation* and *double sigmoid* as defined in Table 4.2 and described further in the reference [30]. Let s is the data, $\acute{s}$ is the normalized data, and k is index of number of the data.

Normalization	**Equation**
minmax	$\acute{s} = \frac{s - min\{s_k\}}{max\{s_k\} - min\{s_k\}}$
Z-score	$\acute{s} = \frac{s-\mu}{\sigma}$
decimal scaling	$\acute{s} = \frac{s}{10^n};\ n = log_{10}\, max\{s_k\}$
median absolute deviation	$\acute{s} = \frac{(s - median\{s\})}{MAD};\ MAD = median(\lvert\{s_k\} - median\{s\}\rvert)$
double sigmoid	$\acute{s} = \frac{1}{1+exp(-2(\frac{s-t}{r}))}$

Table 4.2: Normalization methods [30]

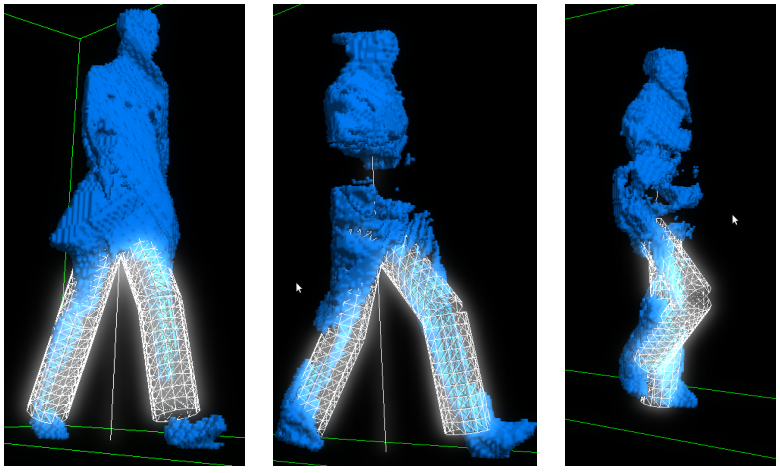
Finally, the performance of our biometric system was also evaluated using a Receiver Operating Characteristic (ROC) for verification and Cumulative Match Characteristics (CMC) for identification/recognition. CMC and ROC are common evaluation methods for biometric systems. The CMC is able to visualize the identification performance

and ROC can describe the verification capability of the biometrics system. To generate the CMC and ROC graphs, we set up an experiment using 184 sample sequences (46 subjects) as probe and gallery and only use the kinematic features. In CMC, we evaluate the identification rate with rank varying from 1 to 46 (the number of subjects). In this experiment, we also report the evaluation of the distance functions in CMC and ROC performance.

4.8 Evaluation of Performance

Our experiments were carried out on the Southampton multi-biometric 3D gait database [76]. We used a database which contains 46 different subjects; four recordings of each subject were taken, giving a total of 184 samples. The database was collected at several different sessions and its quality varies from good to very poor. From 184 sequences we found manually that 144 samples is good, 32 is poor, and 8 is very poor. What we mean by poor quality is that the voxel data are corrupted or missing mostly around the shin. The very poor quality means that many voxels are corrupted around both the thigh and the shin.

4.8.1 Model-fitting

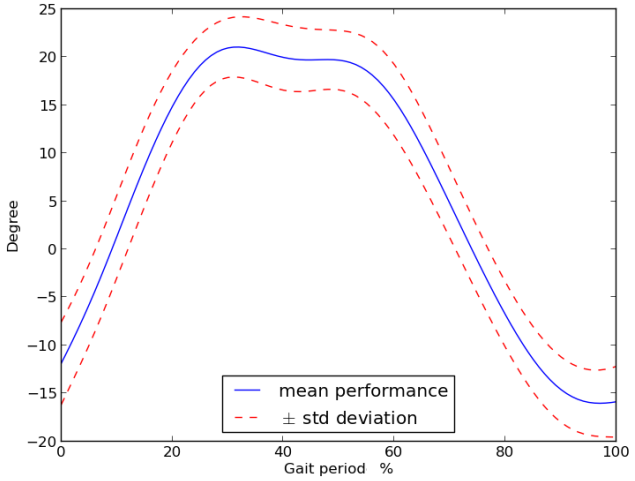


(a) Sample# 78 (b) Sample# 112 (c) Sample# 109

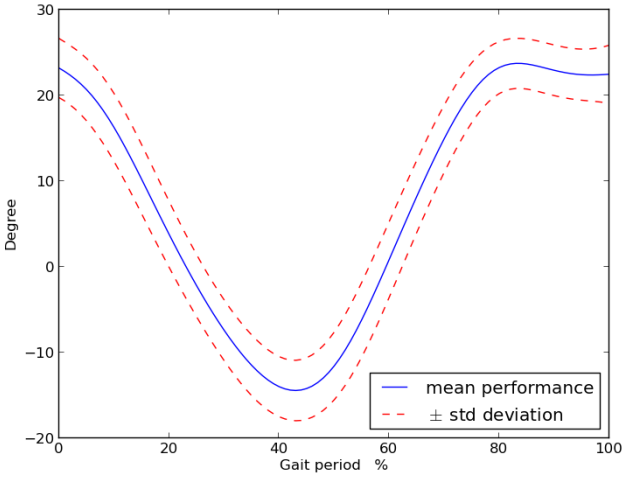
Figure 4.9: Model-fitting result

Figure 4.9 shows the model-fitting results for several frames with varying data quality. Generally, the cylinder gait model was able to fit into the voxel data with a good quality as shown in Figure 4.9(a). In Figure 4.9(b) our model-fitting system was still able to work satisfactorily even though the voxel data quality is imperfect. However in some occasions like in Figure 4.9(c), our system could differ from the optimal fitting.

We argue that Figure 4.9(c) shows limitations associated with the hierarchical fitting method and the data in that the thigh is extracted well, but the shin has less fidelity to the 3D data. However, in term of overall performance our results still showed capability of good fitting.



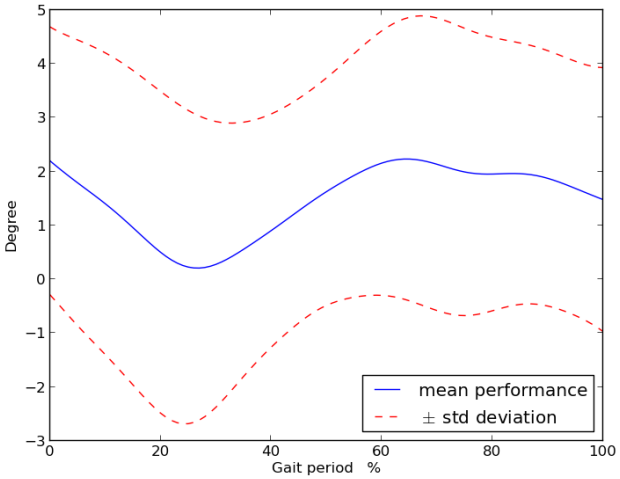
(a) Right thigh sagittal (θ_{RT}) kinematic angle



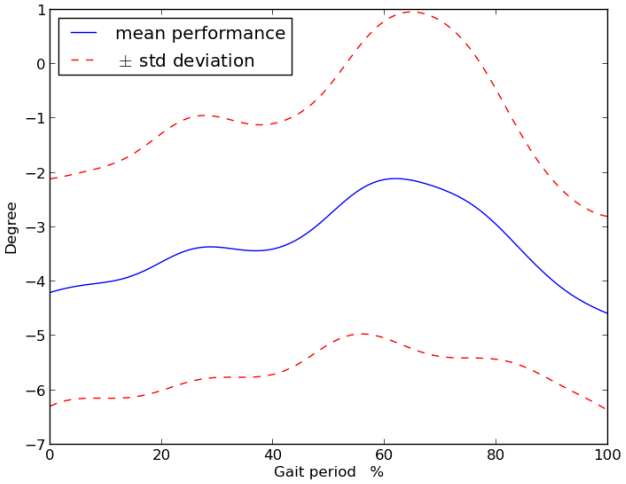
(b) Left thigh sagittal (θ_{LT}) kinematic angle

Figure 4.10: The extracted kinematics of thigh sagittal angles

We were also interested in analysing the extracted kinematic angles of the model. We evaluated the statistical profile of these angles by using the mean and the standard deviation. Due to different frame numbers in one gait cycle of the samples, firstly we normalised each kinematic angle by transforming the frame number into a percentage of cycle/period. The starting frame means 0% and the ending frame corresponds to 100%. After that we calculated all kinematic data to obtain the mean and the standard deviation values. Figure 4.10 and Figure 4.11 shows the statistical profile of the extracted kinematics of thighs in sagittal and frontal angle respectively. The blue solid curve shows the mean performance whilst the red dashed lines show the standard deviation. The



(a) Right thigh frontal (α_{RT}) kinematic angle

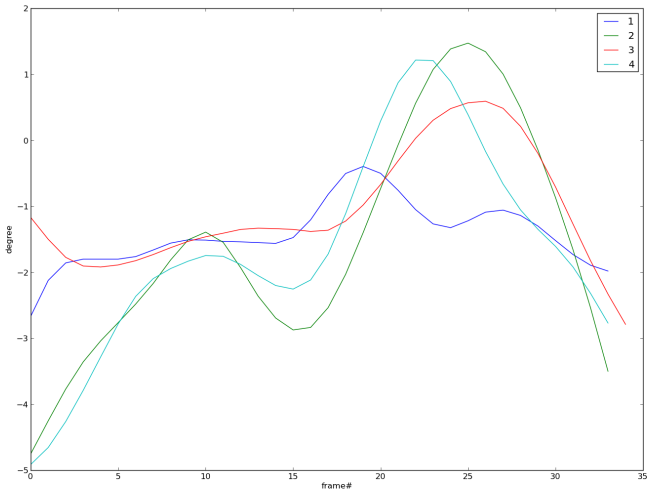


(b) Left thigh frontal (α_{LT}) kinematic angle

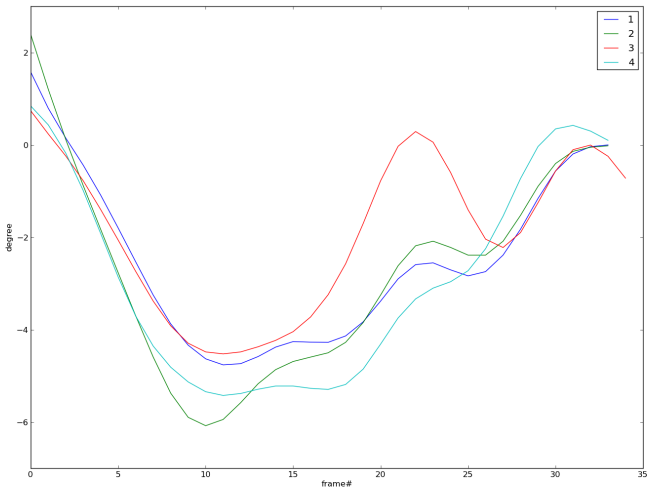
Figure 4.11: The extracted kinematics of thigh frontal angles

trend and pattern of the extracted kinematic angles confirmed the similarity with a result in biomechanics [87].

From the sagittal angles as shown in Figure 4.10, we can see the relation between left and right angles. The left angle is just 180° or half-cycle shifted from the right angle. It confirms that walking consists of swing and stance phases and each phase has approximately 50% of the cycle. Each leg also has a state either in stance or swing phase. From that image, we can also see that the deviation around peak and valley was greater than the other places. Figure 4.11 shows the frontal angles of the thighs. From that image we can find that the right thigh always has a positive frontal rotation whilst the left one has a negative one. This result means that the two thighs most likely never crossed each other in walking. The extracted kinematic angles of shin had similar profiles to the thigh's angles and they can be seen in Appendix D.



(a) Alpha right thigh (α_{RT}) kinematic angles



(b) Alpha right shin (α_{RS}) kinematic angles

Figure 4.12: Frontal angles from samples of subject# 1

Variations of frontal kinematic angles within a subject are presented in Figure 4.12. This figure shows the four samples' angles from the same subject (subject# 1). We can observe that the extracted frontal angles were similar within this subject. Only one sample depicted in red had a small different trend.

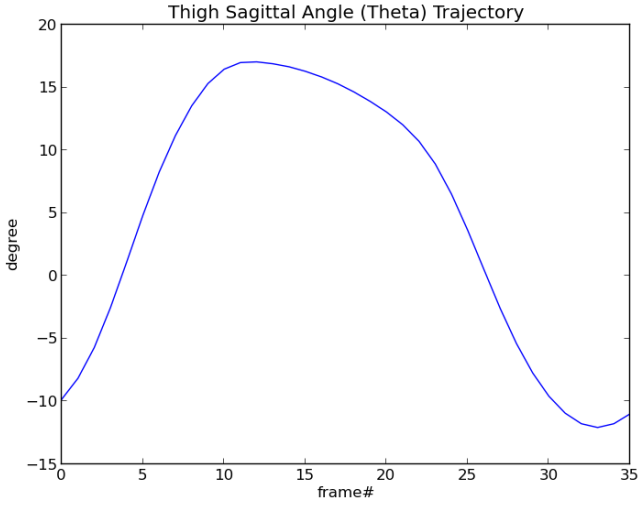
4.8.2 Recognition of structural features

Table 4.3 shows the correct classification rate using structural features. This result shows that height was the most discriminatory feature within the structural features.

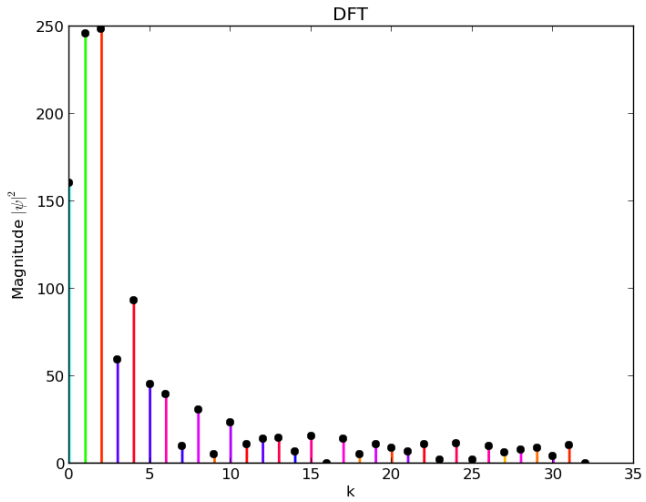
The footprint and stride performed with a low recognition rate, but they contributed to an increased recognition rate when combined with the height feature. The best recognition performance of 64.6% was achieved for k=1 using a combination of height and footprint as the chosen features. It is interesting to note that the footprint was able to increase recognition capability better than the stride.

Chosen feature(s)	**CCR (%)**		
	k=1	**k=3**	**k=5**
Height	40.2	46.7	51.6
Stride	12.5	13.0	13.0
Footprint	27.7	27.7	25.0
Height & Stride	46.1	46.7	41.8
Height & Footprint	**64.6**	64.1	61.4

Table 4.3: k-NN classification result (%) for structural features



(a) Trajectory of θ_{RT}.



(b) Frequency components of θ_{RT} trajectory.

Figure 4.13: Thigh sagittal angle trajectory and its frequency components

4.8.3 Recognition of kinematic features

In order to classify kinematic features, we employed DTW and DFT operations as described in section 4.7.3. In the DTW operation, the kinematic features are regarded as time series data so that we can compare directly the distance between them. The DFT converts the time series data into their frequency components which have magnitude and phase. Figure 4.13 shows the right thigh sagittal angle trajectory (θ_{RT}) and its frequency components. These frequency components then are used as a gait kinematic signature and will be compared each other in the classification or recognition evaluation. As can be seen from the frequency components image, the first six components are very high while the rests are relatively much smaller. Thus, our kinematic angles have more information located in low frequency domain.

Table 4.4 shows a classification result of the kinematic features. Each individual feature was classified with varying distance methods. The comparison of the distance methods shows the Euclidean distance to be best. The best recognition performance was achieved by using the α_{RT} angle which is the movement of right thigh in the frontal plane, confirming an earlier study of the potency of the front view for gait recognition [25].

Chosen feature	**Chosen distance method**			
	Mag	**Mag-Phase**	**Eucl.**	**DTW**
θ_{RT}	33.6	28.2	38.5	16.8
α_{RT}	33.1	37.5	**54.8**	36.4
θ_{LT}	34.7	29.3	32.6	16.3
α_{LT}	36.9	34.7	48.3	28.2
θ_{RS}	29.3	21.1	27.1	13.5
α_{RS}	28.8	35.8	49.4	29.3
θ_{LS}	25.5	22.8	29.3	8.6
α_{LS}	22.2	21.7	40.2	18.4

Table 4.4: Correct classification rate (%) for kinematic feature with k=1

In order to determine the best kinematic features, we evaluated all possible feature subsets. In this case, with eight kinematic angles we can have 255 combinations of kinematic features. We also evaluated those 255 feature subsets using four different distance metrics as described in section 4.7.4. Hence, in total we evaluated 255 x 4 (equals to 1020) unique feature subsets. Again, all classification experiments used leave one out cross validation.

Figure 4.14 shows the correct classification rate of all subsets indexed and sorted from 1 to 1020. The curve had a linear slope in the middle but increased more at the end. The best classification rate was around 86%. Figure 4.15 shows feature subsets' recognition rate for each distance metric. From the image, we can see that DTW distance had the

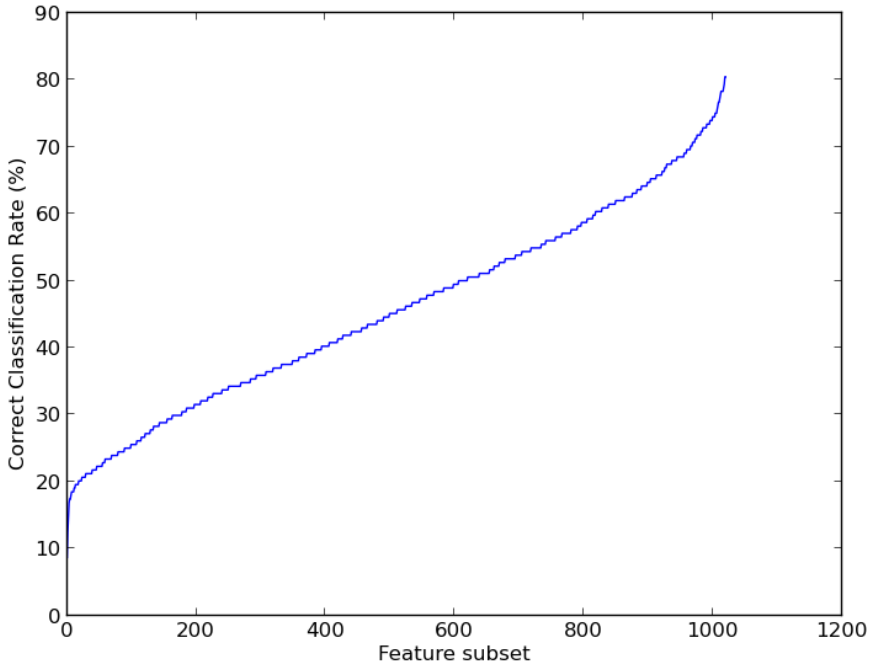


Figure 4.14: Feature subsets' recognition rate

best performance. It had bigger recognition rate than the others for more than half of the subsets.

Number	**Feature**	**Distance metric**	**CCR (%)**
1	α_{RT}	Euclidean	54.8
2	α_{RT}, α_{LT}	Euclidean	78.2
3	$\alpha_{RT}, \alpha_{LT}, \alpha_{RS}$	Euclidean	82.0
4	$\alpha_{RT}, \alpha_{LT}, \alpha_{RS}, \alpha_{LS}$	Euclidean	86.4
5	$\theta_{RT}, \alpha_{RT}, \alpha_{LT}, \alpha_{RS}, \alpha_{LS}$	DTW	84.7
6	$\alpha_{RT}, \theta_{LT}, \alpha_{LT}, \theta_{RS}, \alpha_{RS}, \alpha_{LS}$	DTW	86.4
7	$\theta_{RT}, \alpha_{RT}, \theta_{LT}, \alpha_{LT}, \theta_{RS}, \alpha_{RS}, \alpha_{LS}$	DTW	84.2
8	$\theta_{RT}, \alpha_{RT}, \theta_{LT}, \alpha_{LT}, \theta_{RS}, \alpha_{RS}, \theta_{LS}, \alpha_{LS}$	DTW	80.9

Table 4.5: Best feature subsets' recognition rate based on the number of feature

Table 4.5 describes what feature subset is the best for each number of features in term of recognition rate. From this table, we can see that frontal angles were very dominant and important. The best subsets with four or less number of features were always the frontal angles. Even for a number more than four, it always contained all four frontal angles. We also found that the Euclidean distance metric was the best when we used four or less number of features. However, DTW distance was the best when we used more than four features.

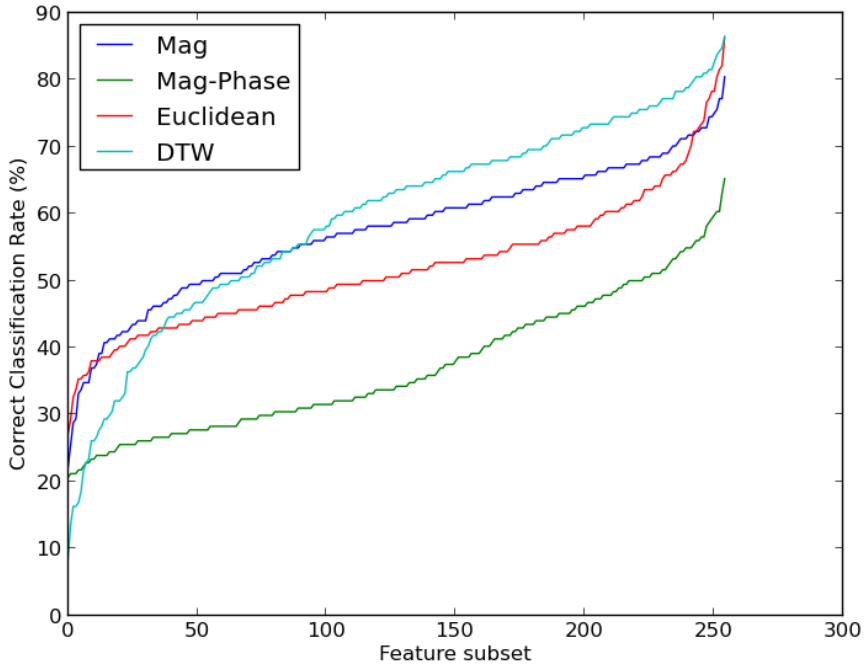


Figure 4.15: Feature subsets' recognition rate for each distance metric

Table 4.6 shows the result of three best kinematic feature subsets. From this table, we can see that the best kinematic features subset was $\{\alpha_{RT}, \theta_{LT}, \alpha_{LT}, \theta_{RS}, \alpha_{RS}$ and $\alpha_{LS}\}$ and $\{\alpha_{RT}, \alpha_{LT}, \alpha_{RS}, \alpha_{LS}\}$ with 86.4% correct classification. The best distance metric of k-NN classification was achieved by DTW and Euclidean respectively. If we refer to the previous results in Table 4.4 and Table 4.5, we can conclude that the Euclidean distance suits best for the recognition of individual or small number of kinematic feature subset, whilst the DTW distance is for a large number of kinematic feature subsets.

Features subset	**CCR (%)**	**Distance method**
$\alpha_{RT}, \theta_{LT}, \alpha_{LT}, \theta_{RS}, \alpha_{RS}, \alpha_{LS}$	86.4	DTW
$\alpha_{RT}, \alpha_{LT}, \alpha_{RS}, \alpha_{LS}$	86.4	Euclidean
$\theta_{RT}, \alpha_{RT}, \alpha_{LT}, \alpha_{RS}, \alpha_{LS}$	84.7	DTW

Table 4.6: Three best kinematic features subsets

Feature subset	**CCR (%)**	**Distance method**
$\theta_{RT}, \theta_{LT}, \theta_{RS}, \theta_{LS}$	53.2	DTW
$\alpha_{RT}, \alpha_{LT}, \alpha_{RS}, \alpha_{LS}$	86.4	Euclidean

Table 4.7: Comparison between a combination of sagittal θ and frontal α angles

We were interested in knowing which one is more important in recognition between the sagittal θ and frontal α kinematic angles. Table 4.7 shows the comparison result and it suggested that frontal side is more important in model-based gait recognition. A combination of frontal angles was able to achieve up to 86.4% recognition rate.

CoH component(s)	**CCR k=1 (%)**
x	8.1
y	6.5
z	66.8
x, y, z	31.5

Table 4.8: Classification results of CoH in GA-based system

Chosen feature(s)	**CCR (%)**		
	k=1	**k=3**	**k=5**
CoH	**66.8**	67.9	65.7
Kinematics	**86.4**	84.2	80.4
CoH & Kinematics	**95.1**	92.9	87.5

Table 4.9: k-NN classification results (%) for all dynamic features

For the CoH, we have evaluated the performance of its components (x, y, and z) and obtained that the z-component gave much better recognition performance as shown in Table 4.8. On the other hand, the combination of these components even produced a lower result. Therefore, we only used the z-component of CoH in every classification task.

Table 4.9 shows the classification results using all dynamics feature. From this table we can see that our system was able to achieve quite high recognition result using dynamic features. The best result was achieved at 95.1% CCR by a combination of CoH and kinematic features. The dynamic features clearly had much better discriminatory power than the static features as previously discussed.

We have also tried the five different methods of data normalization when combining the available features. However, the data normalization methods were not able to significantly improve the performance.

4.8.4 CMC and ROC analysis

We conducted an experiment for identification and verification analysis using CMC and ROC graphs. We decided only to evaluate the kinematic features as they have a much better recognition rate compared to the structural features.

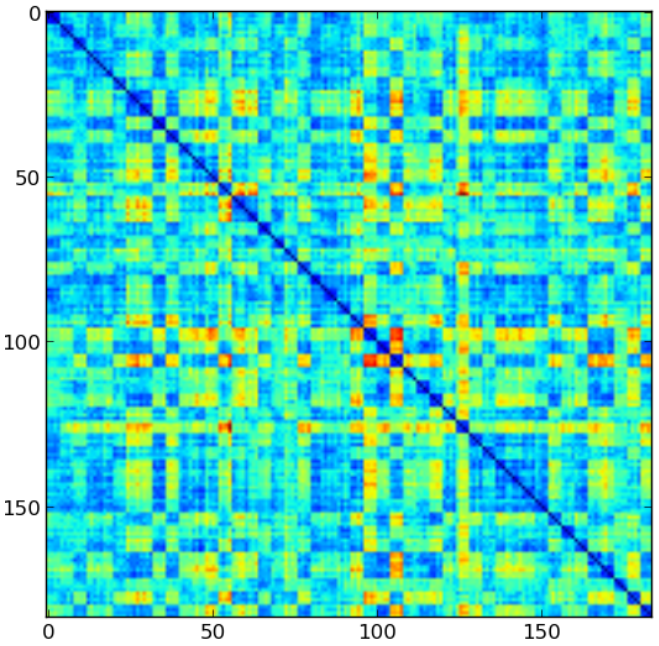


Figure 4.16: Similarity matrix of the best kinematic feature subset

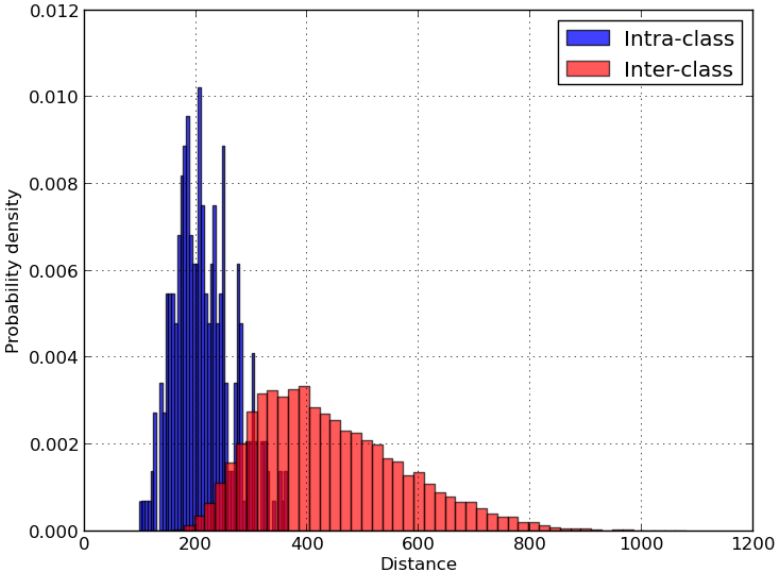
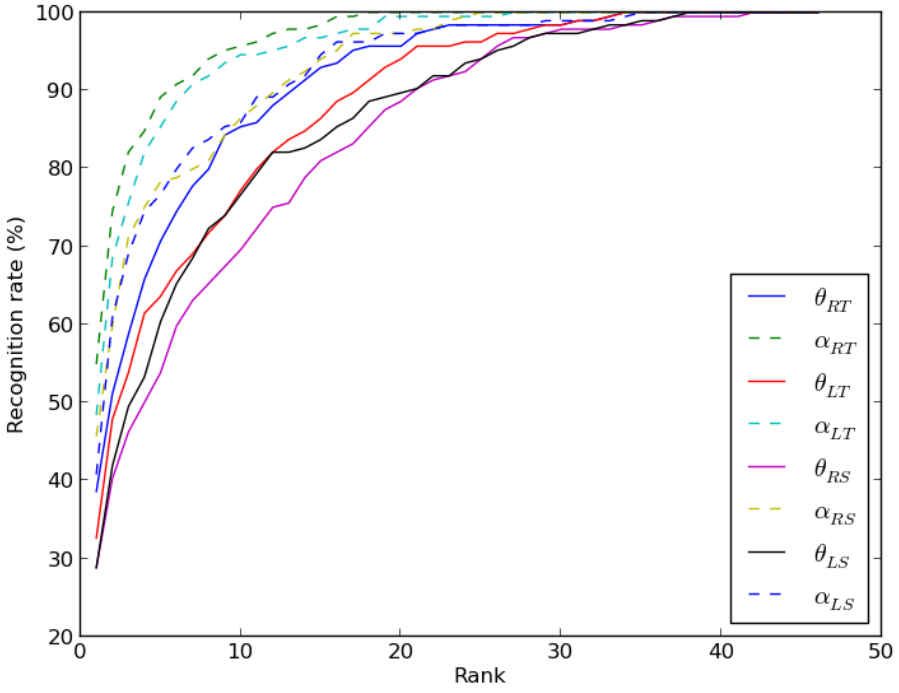


Figure 4.17: Intra/Inter-class variation for the best feature subset

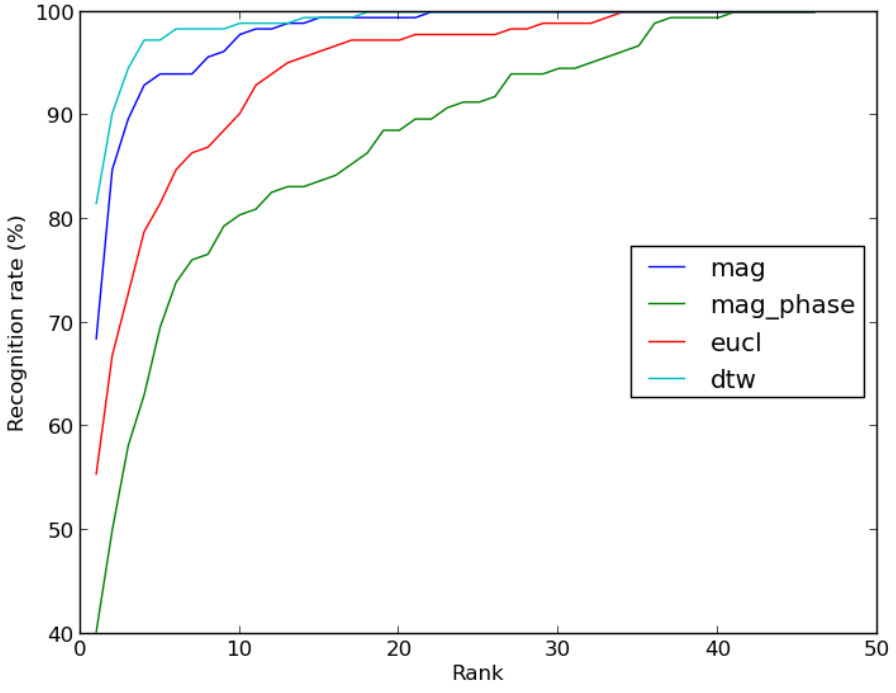
In order to compute the CMC and ROC graphs, we need to compute similarity scores between the probe and the gallery data as described in the section 4.7.4. In our experiments, we represented the similarity scores using distance functions. In our case, the smaller the distance reflects a better match.

Figure 4.16 shows the similarity matrix of recognition using best kinematic feature subset, i.e. $\{\alpha_{RT}, \theta_{LT}, \alpha_{LT}, \theta_{RS}, \alpha_{RS}, \alpha_{LS}\}$. In this image, blue represents the good matches and red is for the poor matches. We have changed all values at the diagonal of the matrix

to be maximum in order to ensure that we do not count the recognition between the same samples (which are obviously having zero distance). Ideally, the darkest blue color areas (lowest distance) are near the diagonal. The intra-class and inter-class distributions for the best feature subset are shown in Figure 4.17.



(a) CMC of all individual kinematics angle using Euclidean distance function



(b) CMC of a combination of all eight kinematic angles using four different distance functions

Figure 4.18: Cumulative match characteristics (CMC)

Figure 4.18 shows the cumulative match characteristics (CMC) graph for recognition using kinematic features. The first graph, Figure 4.18(a), shows the CMC performance for all kinematic angles individually (eight features) using Euclidean distance function. We chose Euclidean distance because it had best CCR performance in the classification task. This graph shows a large range of recognition-performance variations within the kinematic features. The figure also shows that frontal angles had better CMC performance than sagittal angles. The second graph, Figure 4.18(b), shows the CMC of the combination of all kinematic angles evaluated using four different distance metrics. We can see from Figure 4.18(b) that only DTW distance was able to achieve more than 80% CCR at rank 1. It also had more than 90% recognition rate at rank 5. At rank 10, all distance functions achieved more than 80% recognition rate. In overall performance, the better choices of distance function characterised by CMC curves were DTW, Magnitude, Euclidean, and Magnitude-Phase respectively.

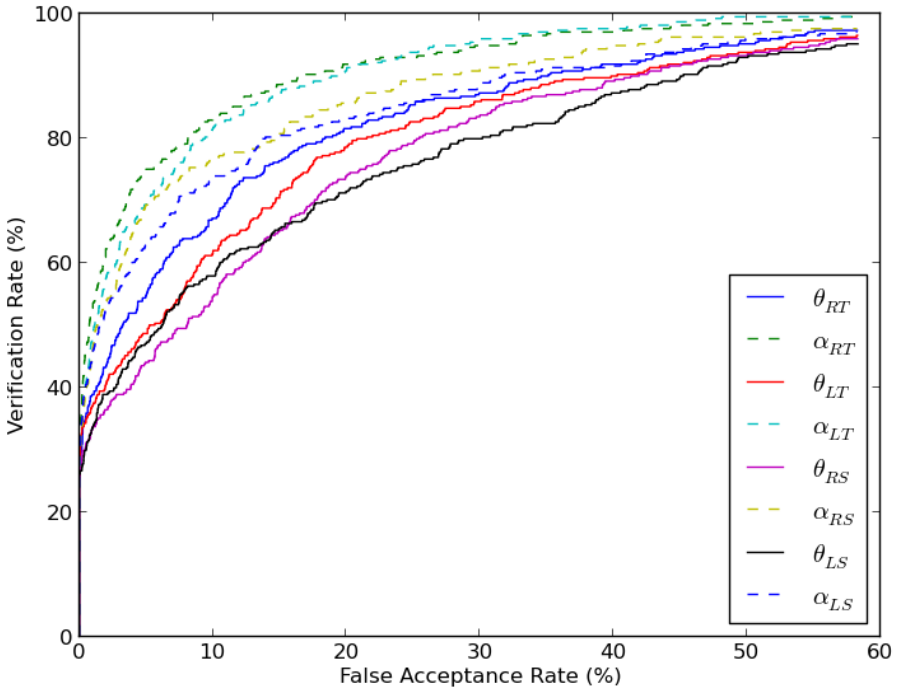
The verification evaluation is described in a ROC curve as shown in Figure 4.19. The performance of individual kinematic angles using Euclidean distance function is shown in Figure 4.19(a). This graph shows a large range of verification-performance variations within different angles and suggests that frontal angles had better performance in a verification task. We can also see that most of them had fast increasing trends within first 10% FAR and then after that they increased steadily with lower slopes. Figure 4.19(b) shows a verification performance of the combination of all kinematic angles evaluated with four different similarity distance functions. On average, they had around 70% verification rate at 10% false alarm rate. We can also see from the figure that the best verification performance was achieved by using the DTW distance function.

4.8.5 Analysis of dynamic programming contribution

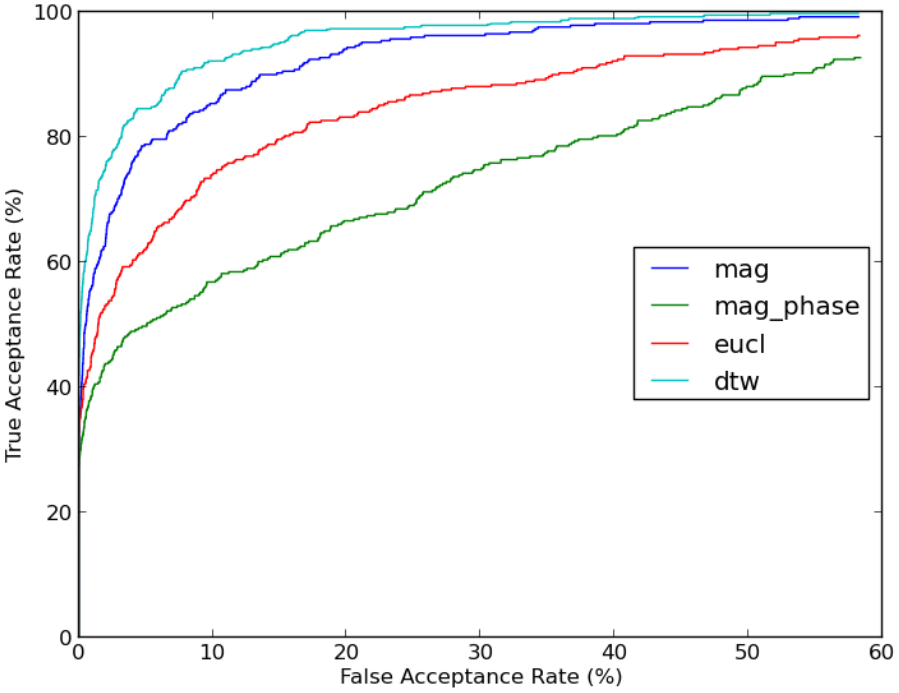
As mentioned previously, dynamic programming (DP) can guarantee the smoothness of the extracted kinematic trajectories. Beside our DP-based method, we can extract the trajectories without dynamic programming by directly choosing the angle in each frame with minimum energy map. Thus, there will be no guarantee of smooth trajectories produced. Here we seek to compare the performance of kinematic trajectories extracted with and without dynamic programming.

Method	**Best CCR (%)**		
	k=1	**k=3**	**k=5**
Using DP	86.4	84.2	80.4
Without DP	84.2	83.1	80.4

Table 4.10: Comparison of best correct classification rate (%) between using DP and without DP



(a) ROC of all individual kinematic angles using Euclidean distance function



(b) ROC of a combination of all eight kinematic angles using four different distance functions

Figure 4.19: Receiver operating characteristic (ROC)

We evaluated the performance of all trajectory angle combinations. There were 255 possible combinations and we selected the best result of k-NN classification for each value of k. Table 4.10 shows the comparison of best correct classification rate (%) between using DP and without DP. From this table, we can see that the DP based

system had performance of more than 86% CCR. On the other hand the trajectory without DP only achieved around 84%, which is lower.

4.9 Conclusions

A simple yet effective structural model has been developed here for model-fitting directly in 3D space. The structural model used a correlation filter with the Euclidean distance metric to measure the similarity between the model and the data. Each 3D point in the model sought the closest corresponding voxel data to define its distance and the total distance was simply the aggregation for all the point distance inside the model. By varying the model parameters, we generated a correlation energy map that measures the best fit between the model and the data. Dynamic programming was used to select the optimal trajectory based on the correlation energy maps.

The experimental results show that the proposed method was able to extract both structural and dynamic features. The extracted features consisted of subject's height, stride length, footprint angle, center of hip and kinematic joint angle trajectories. These extracted features had potential to become the discrimination factors for gait biometrics. The footprint angle was considered unique to 3D gait data and it has surpassed the performance of stride length feature which is commonly used in 2D data. We have also shown that kinematic features have greater discriminatory capability than the structural parameters.

The evaluation of kinematic angle trajectories explained that frontal angles have better discriminatory power than sagittal angles. It is also concluded that the Euclidean distance metric of spectral components give the best achievement for individual kinematic angle recognition. However, if we use a combination of some kinematic angles together in the recognition process, we will find that the dynamic time warping metric in classification is the best.

The proposed structural model-fitting system was able to achieve the best performance at 95.1% correct classification rate. This best recognition rate was achieved by the combination of CoH and kinematic angle features and by using dynamic time warping distance metric in the classification.

Chapter 5

Refining the Fitting Results

5.1 Introduction

Our system currently uses a structural model-fitting method based on using a correlation filter and dynamic programming. The correlation filter has a role to find the minimum distance between the model and the voxel data. The dynamic programming contributes to smoothing the extracted trajectories. This structural model method basically works as a brute-force optimization with an objective to find the maximum spatio-temporal similarity between the model and the data.

Due to the limitation of hierarchical approach in the previous model-fitting and tracking procedure, we were interested in refining the current results. We sought to do a fitting process of all body parts at the same time which means that all best angle parameters and new CoH values are sought within one single fitting process for each frame. Fitting the lower leg model most likely will generate a cross correlation map with multiple optima in high (up to 11) dimensional space.

Optimization is the process of finding the point that optimizes a function. A local optimum of a function is a point where the function value is better than or equal to the value at nearby points, but possibly worse than at a distant point. While a global minimum is a point where the function value is better than or equals to the value at all other feasible points. Due to the huge search space of the generated cross correlation map between the model and the data, we need global optimization algorithms to find a good feasible model pose.

The objective of global optimization is to find the globally best solution of (possibly nonlinear) models, in the (possible or known) presence of multiple local optima. Formally, global optimization seeks global solution(s) of a constrained optimization model. There are some well-known global optimization algorithms available such as simulated annealing (SA), particle swarm optimization (PSO), and genetic algorithm (GA).

We used a genetic algorithm as an optimization tool here to find a better fit while introducing some important constraints. A genetic algorithm tends to succeed in an environment in which there are many candidate solutions and the search space is uneven with many optimum values. When implementing the genetic algorithm, we used the current extracted trajectories from structural model for an initialization and then optimized the fitting objective function in the fitting process of all cylinders simultaneously. We also defined an evaluation function that accommodates some constraints such as enforcing plausible poses for all body parts.

5.2 Genetic Algorithm

A genetic algorithm is a global search heuristic that mimics the process of natural evolution [27, 31, 54]. A genetic algorithm works based on the following procedures iteratively:

1. Select the best-fit individuals for reproduction
2. Breed new individuals through crossover and mutation operations
3. Evaluate the individual fitness of new individuals
4. Replace least-fit population with new individuals

Using a genetic algorithm we can redefine our model-fitting process as a constrained optimization problem. We can set up some important constraints such as prohibiting extreme kinematic angles, establishing a smooth path, and ensuring plausible walking poses (forbidding inter-penetration among body parts).

5.3 Encoding the Model-fitting Problem

Before we can run the searching process using a genetic algorithm, the first stage is that we need to encode the candidate solutions of our problem into a chromosome. A chromosome is a set of parameters which define a proposed solution to the problem that the genetic algorithm is aimed to solve. Our problem is seeking the best model pose in each frame. Thus we encode the model pose with a maximum of 11 parameters P consisting of three parameters of CoH location (x, y, z) and eight angle parameters $(\theta_{RT}, \alpha_{RT}, \theta_{LT}, \alpha_{LT}, \theta_{RS}, \alpha_{RS}, \theta_{LS}, \alpha_{LS})$. Because in this stage we are refining results, we use the hierarchical structural model results as the initial poses and then try to find the optimal adjustment values using a genetic algorithm chromosome. We define a chromosome in the genetic algorithm with incremental value of the model parameters as δP.

5.4 Evaluation Function

Evaluation or a fitness function is used to evaluate how well the solution domain can tackle the problem. We have defined one main evaluation function combined with a penalty function as constraint. When calculating the evaluation function, the model pose parameters are derived from the initial value plus the chromosome value. The updating process of new model parameters is described below in Equation 5.1, where the subscript i defines the kinematic angle index, the subscript 0 denotes data from initialization and δ denotes the value from the chromosome.

$$\begin{aligned} x &= x_0 + \delta x \\ y &= y_0 + \delta y \\ z &= z_0 + \delta z \\ \theta_i &= \theta_{i0} + \delta\theta_i \\ \alpha_i &= \alpha_{i0} + \delta\alpha_i \end{aligned} \tag{5.1}$$

After determining all required model parameters, we instantiate the four cylinders of the structural model. The main evaluation function evaluates the total distance between the model and the data using using the correlation filter as described in Equation 4.5 in the previous chapter. Each point in the model will search the nearest distance voxel in the data and then be squared produce the distance score. The total score of the evaluation function is derived by aggregating the distance scores in the point cloud model.

A single constraint function has been set up here in order to ensure plausible walking poses (forbidding inter-penetration among body parts) in the solution domain. We used information from the knee and ankle joint vertices' location. By measuring the distance between left and right joints, we can define a condition when penalty score *pscore* will be added to the current evaluation score. If the joint distance $jointDist$ is less than or equal the constant radius of the cylinder then we will add the penalty score. This score will increase proportionally to how deep inter-penetration among body parts occurs as described in Equation 5.2

$$pscore = 100 + weight * (RADIUS - jointDist) \tag{5.2}$$

Due to a huge computational cost of genetic algorithm, in the implementation stage we need to speed up the execution time of the evaluation function by introducing an Euclidean distance transform of the 3D data. By performing the distance transform on the data, we have created a dictionary or template for the model correlation distance. Each point in the model will have a corresponding value in the distance transform data array.

5.5 Selection

The genetic algorithm works in a manner similar to natural selection which always chooses the fittest individual to survive and to evolve. There are many different types of selection method in genetic algorithm such as roulette wheel selection, tournament selection, and rank selection. We chose the most common type, i.e. roulette wheel selection. In roulette wheel selection, each individual is given a probability of being selected that is directly proportionate to their fitness score. In other words, the roulette wheel segments are of size proportional to each individual's relative fitness. An individual is then chosen randomly based on these probabilities. Pseudo-code for a roulette wheel selection is shown below in Algorithm 1.

Algorithm 1 Roulette wheel selection algorithm

```
1: for all members of population do
2:     sum += fitness of this individual
3: end for
4: for all members of population do
5:     probability = sum of probabilities + (fitness / sum)
6:     sum of probabilities += probability
7: end for
8: repeat
9:     number = random between 0 and 1
10:    for all members of population do
11:        if current probability < number ≤ next probability then
12:            this individual have been selected
13:        end if
14:    end for
15: until new population is full
```

5.6 Genetic Operator

Crossover and mutation are two basic operators in a genetic algorithm that influence the search performance. The type and implementation of operators depends on encoding schema and also on the problem to be tackled. There are many ways how to implement crossover and mutation in genetic algorithm.

5.6.1 Crossover

Crossover is a genetic operator that combines (mates) two chromosomes (parents) to produce a new chromosome (children). The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. It is analogous to reproduction and biological crossover, upon

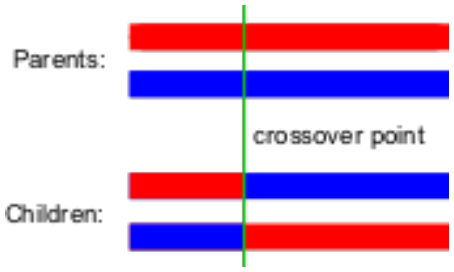


Figure 5.1: An illustration of one-point crossover technique

which genetic algorithms are based. Crossover occurs during evolution according to a user-definable crossover probability. There are many different crossover techniques in genetic algorithm such as one-point crossover, two-point crossover, and uniform crossover. In our experiments we chose the simplest crossover method, one-point crossover.

In one-point crossover, a single crossover point on both parents' organism chromosome is selected. All data beyond that point in either organism string is swapped between the two parent organisms. The resulting organisms are the children. This simple procedure can be visualized in Figure 5.1.

5.6.2 Mutation

Mutation alters one or more gene values in a chromosome from its initial state. This can result in entirely new gene values being inserted and it is analogous to biological mutation. Mutation is essential to prevent the population from stagnating at any local optima. Mutation occurs during evolution according to a user-definable mutation probability. This probability should be set low otherwise it will turn the whole search process into a primitive random search. Depending on the data type of the encoded genes, there are some options for the mutation technique to use such as flip-bit, uniform, non-uniform, and Gaussian mutation. Because we had chosen integer genes, we chose uniform mutation that replaces the value of the chosen gene with a uniform random value selected between the user-specified upper and lower bounds for that gene.

5.7 Parameters of Genetic Algorithm

There are four parameters in our genetic algorithm design, i.e. mutation probability, crossover probability, number of population and number of generation. The mutation and crossover probability were set equal to 0.02 and 0.9 respectively. These parameters would be later tuned and here the purpose is to show that the technique can operate successfully.

The number of generation and population are essential and must be chosen carefully because they can affect the quality of the performance as well as the computational time. A genetic algorithm can be very slow when it has a large population and number of generation as well as a complex evaluation function. We need to analyse prior experimental data before we are able to choose good values of population and generation numbers. Here we set a number of 80 for population and 100 for generation. The reasons behind this are explained later in the experimental section shortly after this.

5.8 Termination Criteria

There are some methods to terminate the iteration process in genetic algorithm. Generally there are two kinds of criteria i.e. threshold and convergence. Some of them are generation number, evolution time, fitness threshold, fitness convergence, and population convergence. In this thesis, we used both the population number and the population convergence to terminate the searching process.

5.9 Experiments and Results

We conducted several experiments using a genetic algorithm with focus on determining the best model parameters of model-fitting problem. Generally, a smaller distance between model and data represented by the Euclidean distance function in the model-fitting process means that better pose recovery is produced. As mentioned earlier in the data encoding Section 5.3, there are many model parameters we can choose in our chromosome. We then tried to choose several set of model parameters and then used our genetic algorithm method to minimize the distance function. The system performance was evaluated in terms of recognition capability.

5.9.1 Choosing the right generation number

As mentioned above that a good selection of the internal parameters of a genetic algorithm may reduce the optimization speed considerably as well as maintaining the quality of the solution. Before applying our designed genetic algorithm (GA-based system), we also need to define the parameters of population and generation numbers. We decided to set the population number to 80. This number of population was considered to be reliable enough for finding good acceptable solution. In order to choose the right number of generation, in the first stage we ran the GA-based model-fitting with a value of 500 for generation which is considered to be a large number of generations. This process took about 12 minutes for one sequence with 34 frames. Then we analysed the convergence results to find the optimal generation number.

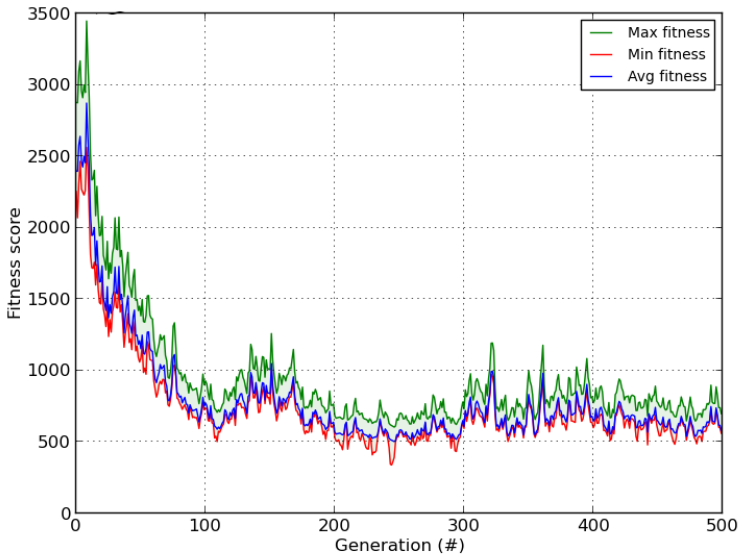


Figure 5.2: Fitness score statistics (min, average, and max)

The fitness and raw score statistics (min, average and max) evolving along these generations can be seen in Figure 5.2. Raw score is a value that relates directly to the Euclidean distance measurement between the model and data. While the fitness score is a value obtained from rescaling the raw score. It is important to recall that because we need to minimize the distance, we ran the GA-system in a minimization mode. In this optimization mode, lower fitness and raw scores are the targets. The initial minimum, averaged, and maximum fitness score was around 2100, 2500, and 2900. During the first 110 generations, the fitness score decreased quite dramatically. After that, it seems that the fitness score had varying behaviour with small margin of swing. The minimum fitness function was around 300 while the best achieved score within 110 generations was slightly higher at 500.

The other experimental data collected here were about the fitness and raw scores differences between the maximum and minimum. This data can be seen in Figure 5.3. Ideally, when the number of population grows up, the quality of population also become better. It means that the difference between the best and the worst is smaller. This figure shows that both the fitness and raw score differences had monotonic decline during the first 100 generations. In the next further generations, the different values fluctuated with small range of improvement. Based on these two experimental data, we decided to choose 100 as the best value for the parameter of generation number.

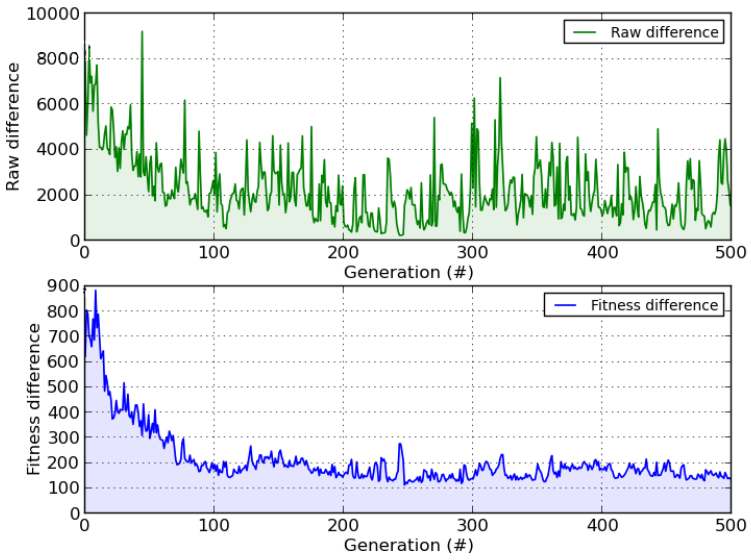


Figure 5.3: Fitness and raw score differences

5.9.2 Fitting and recognition performance

In this experiment, we used exactly the same dataset as the previous chapter on the structural model which is the gait dataset from the Southampton multi-biometric tunnel containing 46 different subjects; four recordings of each subject were taken, giving a total of 184 samples. The extracted angle trajectories from the structural model were used as an initial state in this GA-based model-fitting. The GA-based system then generated new angle trajectories aiming for better accuracy and recognition performance. It is important to note that repeated trials of the genetic algorithm on the same dataset are not guaranteed to produce exactly the same result because of the random nature of the genetic algorithm's operation.

An improvement in the extracted trajectories can be seen in Figure 5.4. In these images, there were two sagittal angle trajectories extracted from thigh (hip joint) plotted relative to the frame number in x-axis. The first trajectory graph is the blue structural-based trajectory and the second one is the green GA-based trajectory. As can be seen, both trajectories had the same trend and similar peak location at a similar frame number. However, the structural model path had little variations and even some cases it showed flat values for several frames. On the other hand, the GA-based path seemed improve the accuracy of fitting by changing some flat values into better values that follow the path trend. It is important to mention here that currently we are unable to measure the absolute error metric as we do not have the ground truth values of the trajectories. However, the visual observation as available in section 4.8.1 can confirm that the

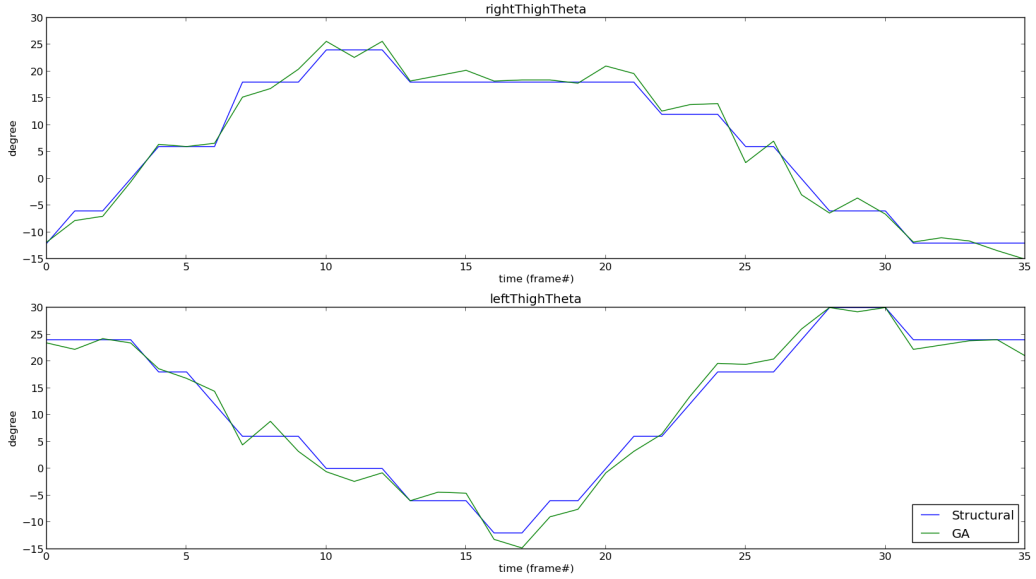


Figure 5.4: Structural and GA-based extracted trajectories

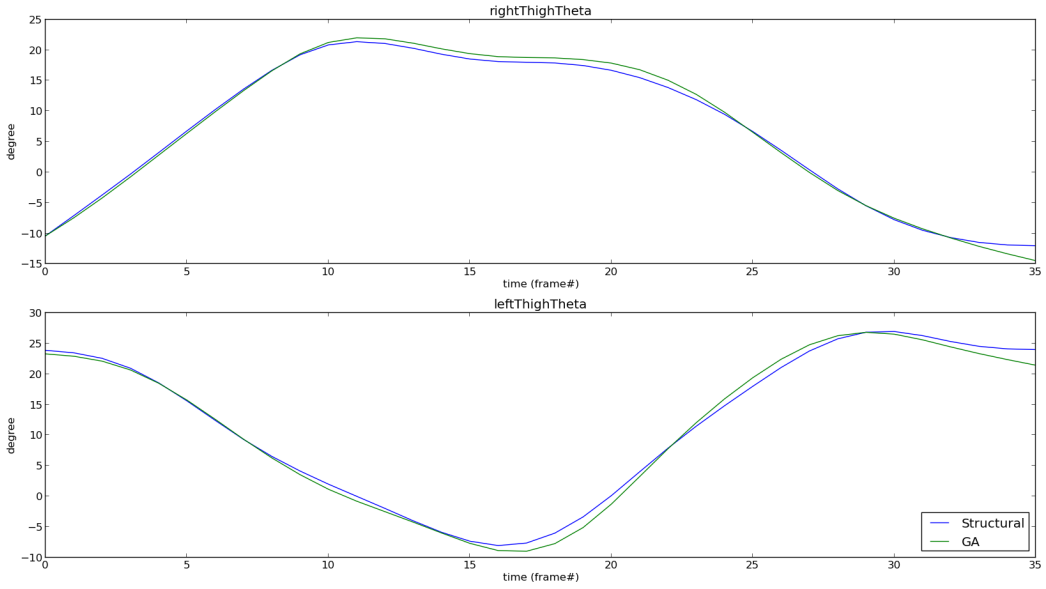


Figure 5.5: Structural and GA-based extracted trajectories after smoothing

extracted trajectories corresponded to very plausible pose and accurate enough to be justified by human eyes.

After we performed smoothing operation on these paths, we can see than the two trajectories were almost identical as seen in Figure 5.5. The smoothing operation was conducted using a Hanning window low-pass filter. The smoothing effect is to reduce noise which usually appears in high frequencies.

We evaluated the correct classification rate of the GA-based system based on the choice of

a human model parameter set. As mentioned earlier, we had several model parameters and we were able to set up any set of parameters in the encoded chromosome to be optimized. Table 5.1 gives the name of each set with its corresponding parameters. It is worthy to recall the meaning of these variables once again. The centre of hip coordinate (CoH) is represented by x, y , and z; θ and α mean the sagittal and frontal angle; T and S denote thigh and shin; R and L denote right and left.

Variables set	**Variables**
SET1	$\theta_{RT}, \alpha_{RT}, \theta_{RS}, \alpha_{RS}, \theta_{LT}, \alpha_{LT}, \theta_{LS}, \alpha_{LS}$
SET2	$x, y, z, \theta_{RT}, \alpha_{RT}, \theta_{RS}, \alpha_{RS}, \theta_{LT}, \alpha_{LT}, \theta_{LS}, \alpha_{LS}$

Table 5.1: Variable sets in GA-based evaluation

After we obtained new refined kinematic trajectories from the GA-based system, we studied classification on those features using k-NN and performed validation using LOOCV. We conducted classification on all parameter sets for each kinematic variable. We also combined all joint angle features (θ and α) in each set to achieve higher recognition rate and to see the discrimination power of the gait kinematic trajectories.

Chosen feature(s)	**Chosen distance method**			
	Mag	**Mag-Phase**	**Eucl.**	**DTW**
θ_{RT}	39.1	21.1	38.0	35.8
α_{RT}	32.6	34.7	55.9	32.6
θ_{LT}	34.7	30.4	35.8	30.9
α_{LT}	33.6	30.9	52.7	36.4
θ_{RS}	34.2	27.1	28.8	14.1
α_{RS}	26.0	27.7	42.9	28.2
θ_{LS}	30.4	33.6	33.6	21.1
α_{LS}	21.1	35.8	42.3	25.0
all angles		89.67		

Table 5.2: Correct classification rate (%) for GA-based SET1 with k=1

Table 5.2 and 5.3 shows the classification results of GA-based extracted kinematic features for SET1 and SET2 respectively. From the two tables we can see that SET1 generally performed better than SET2. In this case, adjusting the trajectories alone gave better performance rather than combining them together with the CoH.

To conduct a further evaluation, we analysed the performance of CoH recognition before and after GA-based optimisation. In the GA-based optimization, the initial value came from the structural model which is here called CoH-Structural Model. The CoH extracted from the GA is called GA-SET2 as it was produced by SET2 experiment. Table 5.4 shows the results of this experiment. From the table we can see that using GA-SET2 did not increase the recognition rate. Therefore, we can say that the original

Chosen feature(s)	**Chosen distance method**			
	Mag	**Mag-Phase**	**Eucl.**	**DTW**
θ_{RT}	35.3	20.1	34.7	30.4
α_{RT}	17.9	26.6	42.9	28.8
θ_{LT}	32.0	32.6	34.7	18.4
α_{LT}	26.0	25.0	43.4	20.6
θ_{RS}	33.1	23.9	29.8	11.9
α_{RS}	27.1	33.6	39.6	27.7
θ_{LS}	33.1	28.8	32.6	23.9
α_{LS}	25.5	30.4	43.4	28.2
all angles		88.5		

Table 5.3: Correct classification rate (%) for GA-based SET2 with k=1

CoH estimation method by averaging the voxel data has served perfectly well as a simple and good CoH estimator.

Feature(s)	**CCR (%)**	
	Structural Model	**GA-SET2**
x	8.1	6.5
y	6.5	6.5
z	66.8	50.5
x, y, z	31.5	31.5

Table 5.4: Classification results of CoH in GA-based system

One of the benefits working in 3D data is that we can explore new features. This new feature can be unique to 3D data only. In the structural or static feature, we have had footprint angle as a unique feature. For the dynamic feature, we now study the pelvis rotation. It is believed that a women's gait has greater pelvis rotation than for men whose shoulders rotate more.

We conducted an experiment to study the pelvis rotation angle. We used the same structural model, and allowed the thigh cylinders to rotate in the transversal plane. Then we used GA-based system to search the best model-fitting by encoding the pelvis rotation angle in the chromosome.

Figure 5.6 shows the result of the pelvis rotation angle (degree) in one gait period. It is clear that the results were very noisy and did not follow a simple sinusoidal pattern.

Table 5.5 shows the correct classification rate (CCR) of the pelvis rotation. As can be seen from the table, pelvis rotation had an average recognition rate compared to any structural feature. The pelvis rotation had a smaller recognition rate than the height,

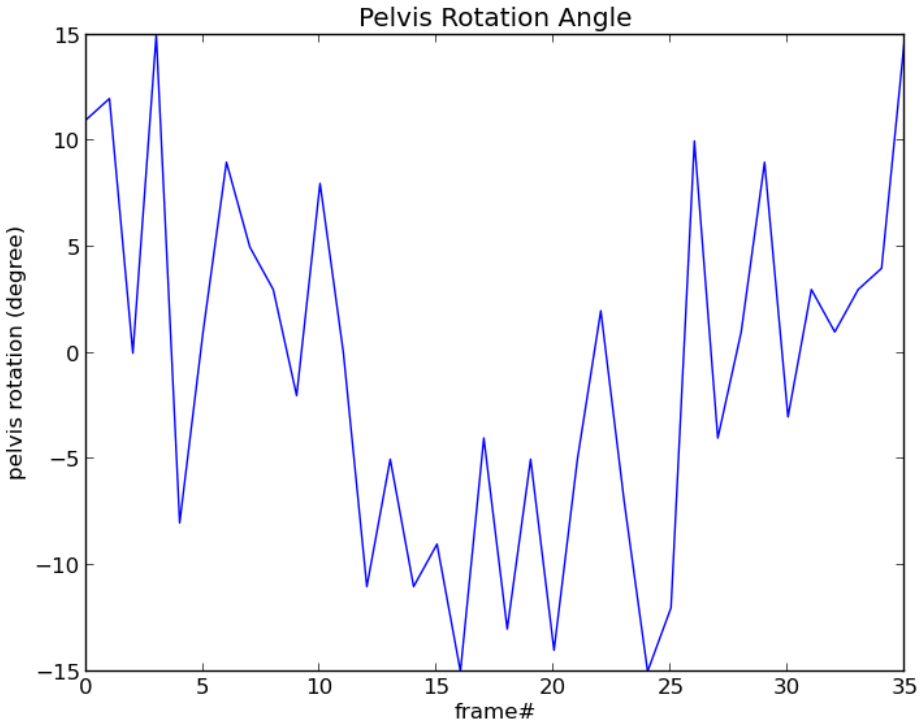


Figure 5.6: Pelvis rotation angle trajectory

CCR (%)		
k=1	**k=3**	**k=5**
13.0	10.3	9.7

Table 5.5: Correct classification rate (%) of pelvis rotation angle

but it was able to match the stride recognition rate. We argue that if we were to include a torso model and a prior motion model of pelvis rotation, it may be able to extract more accurate results.

5.9.3 Comparing the results against the structural model

We were also interested in directly comparing the results of global-fitting GA-based system with the previous hierarchical-fitting structural model. This comparative data can be used to evaluate whether the global-fitting improves the recognition performance. Table 5.6 shows the CCRs of each method. The first experiment of GA-based system used SET1 as GA-SET1 and the second experiment used SET2 parameter as GA-SET2.

From the table, we can see that compared to STRUCTURAL results, the GA-based recognition of thigh sagittal angle θ_T decreased, especially GA-SET2, but the shin parts

	Correct Classification Rate(%) k=1		
Feature(s)	**STRUCTURAL**	**GA-SET1**	**GA-SET2**
θ_{RT}	38.0	38.0	34.7
α_{RT}	51.0	55.9	42.9
θ_{LT}	36.4	35.8	34.7
α_{LT}	53.8	52.7	43.4
θ_{RS}	28.2	28.8	29.8
α_{RS}	39.6	42.9	39.6
θ_{LS}	31.5	33.6	32.6
α_{LS}	36.4	42.3	43.4
all angles	82.6	89.6	88.5

Table 5.6: Classification results of GA-based system against structural model

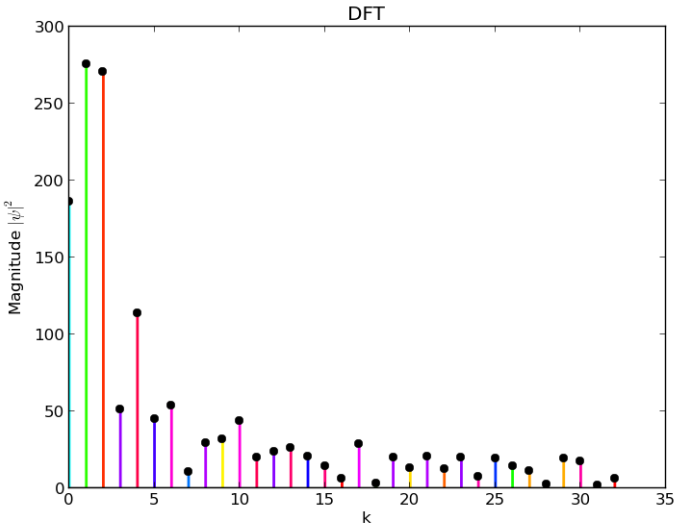
all increased. The GA-based system was able to distribute the recognition performance more evenly and the performance difference was smaller among the angles. The overall result shows that GA-based system increased the performance of structural model by around 10%.

Moreover, complete evaluation on subsets of the kinematic angles also has been done. The best subsets of kinematic features selected in each method are reported in Table 5.7. This result explains that GA-based systems did perform well in recognition performance. We notice that the best feature subset for GA-based systems both were achieved by using all eight trajectory angles, while STRUCTURAL achieved the best subset differently with some angles excluded. The best feature subset results explain that the GA-based system with its global fitting process increased the overall performance by distributing the discriminatory power to all kinematic features. It is also important to note that frontal angles α appeared to contribute more to recognition than the sagittal ones. We can see from the table that all methods had α angles of thigh and shin. Finally, we can conclude that the GA-based global fitting approach improved the recognition power of hierarchical-fitting from the structural model.

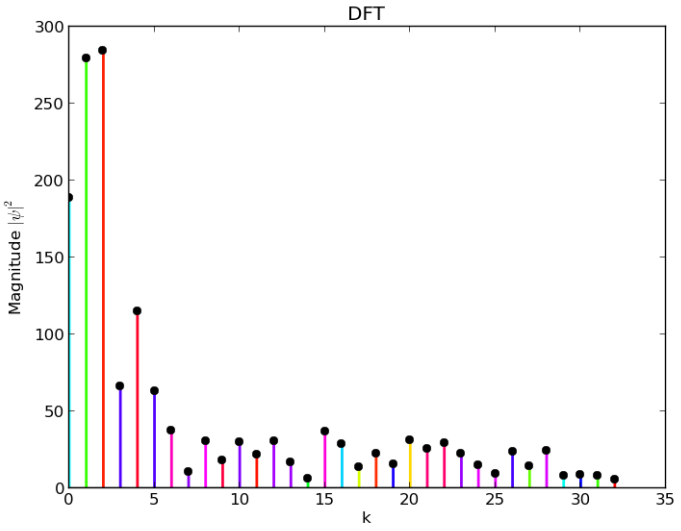
Method	**Best feature subset**	**CCR (%)**
STRUCTURAL	$\alpha_{RT}, \theta_{LT}, \alpha_{LT}, \theta_{RS}, \alpha_{RS}, \alpha_{LS}$	86.4
GA-SET1	$\theta_{RT}, \alpha_{RT}, \theta_{LT}, \alpha_{LT}, \theta_{RS}, \alpha_{RS}, \theta_{LS}, \alpha_{LS}$	89.6
GA-SET2	$\theta_{RT}, \alpha_{RT}, \theta_{LT}, \alpha_{LT}, \theta_{RS}, \alpha_{RS}, \theta_{LS}, \alpha_{LS}$	88.5

Table 5.7: The best kinematic feature subset of all methods

We were also interested to see the difference of frequency components. Figure 5.7(a) and 5.7(b) show the frequency spectrum of right thigh angle trajectories from STRUCTURAL and GA-SET1 respectively. As can be seen, all images describe similar values and trend. The most dominant and significant values are claimed by low frequency



(a) DFT magnitude coefficients of structural model path



(b) DFT magnitude coefficients of GA path

Figure 5.7: DFT magnitude coefficients

components. We can say that there are variations to the gait signature which arise from the DFT coefficients, but these are not dramatic.

We also recorded the computation-time of this algorithm to compare its complexity against the previous structural method. This kind of experiment is important to measure the potential of implementation in real-time. Table 5.8 provides the complete results. We ran our experiments on PC with Core 2 Duo Intel processor and 4GB RAM specification. On average, the structural model needed only 70 seconds to complete the model-fitting process for one sequence and took three and a half hours to complete the whole dataset.

Method	**Execution time**		
	per frame	**per sequence**	**per dataset**
Structural Model	2 seconds	1.2 minutes	3.5 hours
GA-based	12 seconds	7 minutes	20 hours

Table 5.8: Averaged computation time comparison of GA-based system against structural model

By contrast, the GA-based system took time much longer around 12 seconds to complete the global search process each frame. Within a gait sequence, on average there are around 35 frames to represent one gait period. Thus, it took around 7 minutes for one sequence and 20 hours for the gait dataset in which has 184 sequences. In conclusion, GA-based system is 6 times slower and it might be important to modify the current classic GA with a faster GA-variant. Should the increase in performance is necessary; the computational complexity of the GA is a price well worth paying.

5.10 Conclusions

We have used a genetic algorithm (GA) to refine the previous structural model results. The genetic algorithm works as a global optimization tool which make it possible to fit all model cylinders to the voxel data globally at the same time. Thus, the genetic algorithm has transformed the hierarchical fitting problem into a global one. In defining an evaluation function in the genetic algorithm, we introduced a constraint function which maintained the plausible distance of body part joint locations.

The experimental results showed that GA-based system has improved the recognition performance of kinematic features. Therefore, the global fitting approach has better result than the hierarchical one. The GA-based system tends to improve the shins' kinematic angle recognition performance while maintain the thigh's one. We can also confirm that the new CoH feature from GA-based system does not give better recognition performance. Although the GA-based system can improve the overall performance, it is also important to note that this method needs extra computational resources.

Chapter 6

Marionette and Physical Models for 3D Gait Tracking

6.1 Introduction

One of the drawbacks in the structural model is that we can only exploit the structure or shape of the model. We want to develop a new model which not only learns the structure of the human legs but also their behaviour. Motivated by the realistic appearance of a marionette, we are interested in designing a 3D gait model based on using a marionette's structure and behaviour. In some other fields such as robotics and biomechanics, human gait can be modelled by using mass-spring Newtonian physical principles. For instance, Chen [13] has successfully developed a robot for manipulating a marionette with a good degree of realistic appearance, imitating human activities. We are interested in using a 3D marionette and the mass-spring model together. The marionette represents the gait structure and the mass-spring model aims to match the human walking behaviour.

There are other approaches and methods in computer vision areas that use physical principles. In vision-based 3D human tracking and pose recovery area, some methods also use a physical model and analogy. Delamarre et al. [20] proposed a method inspired from physical phenomena using forces applied on a partially volumetric rigid model using iterative gradient descent. Luck et al. [37] used a physically-based method to compute the force and then align a volumetric model into the subsequent voxel data.

In this chapter we describe a novel 3D gait biometric approach using a marionette and mass-spring model. To model the articulated human body, we use a stick-figure which emulates the marionette's motion and joint structure. The stick-figure has 11 nodes representing the human joints of the head, torso, and lower legs. Each node is linked with at least one other node by a spring. The voxel data in the next frame behave as an attractor which is able to generate forces for each node and then iteratively warp

the model into the data. This process is repeated for successive frames for one gait period. The motion kinematics extracted from this tracking process are projected into the sagittal and the frontal planes and then used as a gait signature via the discrete Fourier transform and dynamic time warping.

6.2 Marionette Mass-spring Gait Model

6.2.1 Marionette mass-spring model

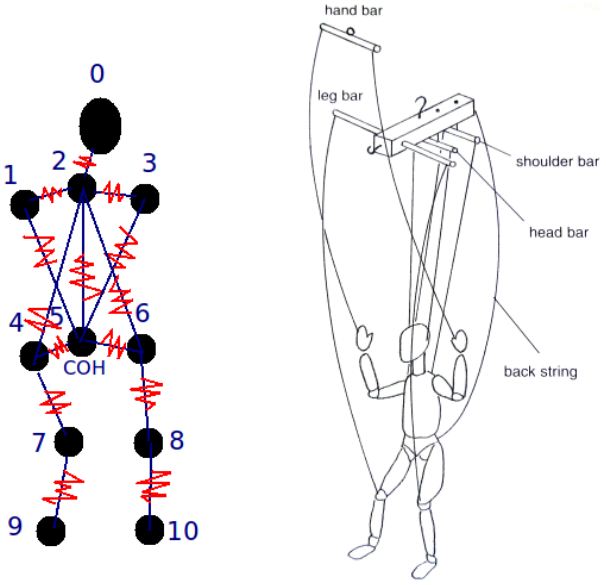


Figure 6.1: Marionette [18] and marionette mass-spring gait model

We use a marionette mass-spring model as our gait model. Figure 6.1 shows our marionette model with 11 nodes and their topology. The model topology is represented with an articulated 3D stick-figure. The human body joints and segments are modelled with nodes (masses) and springs respectively. From the figure, it is clear that we only consider the head, torso, and lower legs but omit the arms because the quality of the voxel data for the arms is poor. We are also not interested in using the arms' kinematics for recognition purposes since a subject might carry a load or wave their arms in a random manner.

As in a marionette, some body nodes of the model can be moved by application of force. We observe that manipulating the marionette's string controller is similar to applying selected forces to the model's nodes. In our model, the forces arise from the voxel points and the springs. The voxel points will behave like attractors to warp the model into the data and the springs will respond so as to maintain the model's topology.

From this marionette mass-spring model, we can extract the motion kinematics of human gait by tracking the model in the data and then measuring the rotational angle of each body segment. We are also potentially able to measure the shoulder rotation angle using the two joints at the arms.

6.2.2 Notations and model parameters

We use notations to describe the model and its parameters. Let $\mathcal{M} = (P, N)$ be the marionette model with $P = \{p_0, p_1, ..., p_{10}\}$ and $N = \{\{N_0\}, \{N_1\}, ..., \{N_{10}\}\}$ being respectively the set of nodes and the set of neighbour nodes, thereby describing the topology. The set of voxel points for each frame is defined as $A = \{a_1, a_2, ..., a_n\}$.

We use bold symbols to represent vectors such as $\mathbf{x}$. The unit vector oriented from point p to q is denoted $\mathbf{u}_{pq}$ and the Euclidean distance between them is denoted $d(p, q)$. The marionette model has a parameter k to define the spring elasticity and m for mass. In our experiment, we use the same value of k and m for all springs and masses in the model.

6.2.3 Preprocessing

Preprocessing enables automatic initialization of the marionette model (to set the initial nodes' positions). We also use prior information from the human body anthropometric data [87] as commonly used in biomechanics as shown in Appendix A. The preprocessing stage can generate robust estimates of the subject height, heel strike locations and the centre of the hip (CoH).

Firstly we seek to process 3D voxel data sequence to obtain 3D bounding boxes. The detail of how to extract 3D bounding boxes has been described in the structural model in section 4.1. This 3D bounding boxes information will be used to estimate the gait period and the subject's height. The start frame of gait period is that for which a maximum width bounding box occurs and the end frame is the second next maximum width. Height is extracted according to Equation 4.2. We also seek to extract the heel strike locations from a subject's footprint image in Figure 4.4. From the footprint image, we can segment each foot and then estimate the heel strike locations using the foot's center of mass. The heel strike information is very important since walking always uses at least one leg to support the body. The heel strikes can be used as a constraint later in the tracking process to limit the shin movement.

Center of mass (CoM) can be estimated by averaging the voxel data location. While the center of hip can use this CoM value, it need a little adjustment to change the hip height using the subject height information and the human anthropometric data. The detail of estimating CoH also has been described at chapter 4 in Equation 4.3.

6.3 Tracking System

In order to obtain the gait kinematics from the subjects, we use a tracking approach using the marionette mass-spring gait model. The tracking module will warp the model nodes' positions each time with new voxel data in the next frame. The voxel data here works as an attractor that is able to pull the nodes in the direction of motion of the data. The pulling mechanism can be seen as an analogy of the string controller in the marionette system.

6.3.1 Model initialization

With prior knowledge of the human anthropometric data, preprocessing results and the heel strike location, we can automatically initialize the marionette model straightaway in the first frame. The ankle nodes positions are placed at the footprint locations. The hip nodes positions are extracted from CoH data derived in the previous preprocessing stage. The hip nodes are extracted from CoH location derived in the previous preprocessing stage. Then, the knee nodes are interpolated between the hip and ankle nodes. Figure 6.2 shows the initialization model nodes imposed into the voxel points.

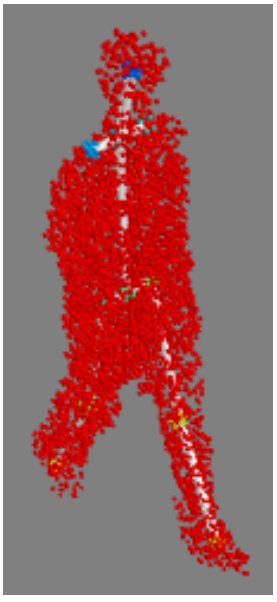


Figure 6.2: Initialization model nodes imposed into the voxel points

6.3.2 Attractor force

In order to generate an attractor force, we need information of voxel data location. The voxel points representing the location were obtained from the voxel binary data with a uniform sampling ratio $s < 1$. A lower value of s will generate a smaller number of

voxel points, hence will increase the computational speed. Based on our experiment, $0.01 < s < 0.1$ gives a good balance between performance and speed.

The voxel points will generate an attraction force on each node of the marionette model. Each node in the model is only affected by a set of attractors nearby. Generating attractor forces is similar to performing human body-parts and body-joints segmentation. To select the attractor set for each node we employ a hierarchical clustering approach with two stages [60]. The first stage is to cluster all the voxel points into the body-parts. Here, the body parts consist of the head, torso, and the two legs. The head is represented by a single point and the others are represented by segments or lines. Each voxel point in A will be assigned to one of the defined body part clusters AS based on the Euclidean distance from the point to the body part. Therefore, $A = \{AS_{Head}, AS_{Torso}, AS_{R.Leg}, AS_{L.Leg}\}$.

The second stage is a local node clustering operation to refine the results of previous body part clusters. A local clustering operation means each body segment will use its own voxel points and will not be affected by any other body segments. As we know that each body part consists of one or more nodes. Therefore, all voxel points in one body part cluster AS then will be divided again into node clusters AP_i. Optionally, we can add a dummy node in the middle of each body segment to increase the accuracy of correct correspondence between the model node and the voxel points attractors. According to Figure 6.1, we have $AS_{R.Leg} = \{AP_4, AP_{dummy}, AP_7, AP_{dummy}, AP_9\}$ where AP_{dummy} is a cluster for a dummy node.

Finally, the attractor force **FA** applied at a node p is defined by:

$$\mathbf{FA}_p = \|\mu - p\| * \mathbf{u}_{p\mu} \tag{6.1}$$

where μ is the mean of points in node cluster AP. The detail of the attractor force implementation can be seen in Appendix C.

6.3.3 Spring force

During the tracking and warping process, the spring will maintain the model topology by inserting spring forces to the nodes when the states are changing. Given two nodes p and q linked by a spring with elasticity constant k and resting length l, the spring force applied from p to q, $\mathbf{E}_{pq}$, will follow Hooke's law as:

$$\mathbf{E}_{pq} = k(d(p,q) - l_{pq}).\mathbf{u}_{pq} \tag{6.2}$$

and the total spring force for a model node p with a set of its neighbour N_p is defined as follows:

$$\mathbf{FS}_p = \sum_{q \in N_p} \mathbf{E}_{pq} \tag{6.3}$$

6.3.4 Updating the model

During the warping process, the model will be iteratively updated until reaching the stopping criteria. There are three states in the model that need updating. Those states are:

1. Acceleration $\mathbf{a}$
2. Speed $\mathbf{v}$
3. Position $\mathbf{x}$

The total force affecting a model node p is a summation of its spring force and attractor force.

$$\mathbf{F}_p = \mathbf{FS}_p + \mathbf{FA}_p \tag{6.4}$$

The acceleration a_p of node p is defined as below:

$$\mathbf{a}_p = \frac{\mathbf{F}_p}{m} \tag{6.5}$$

where m is the mass of the node. The speed is updated with:

$$\mathbf{v}^{t+1} = f\mathbf{v}^t + dT.\mathbf{a}^{t+1} \tag{6.6}$$

where dT is a constant time step, and f is the friction coefficient. Time step dT will affect the execution time needed and also the accuracy of a warping proses. Generally, the lower is better in term of the accuracy. However, a bigger value of dT can achieve very similar accuracy with much faster execution time. Based on our experiments, the tracking system can work well and faster when dT is in a range from 0.02 to 0.2.

Finally, the position of the node is updated with:

$$\mathbf{x}^{t+1} = \mathbf{x}^t + dT.\mathbf{v}^{t+1} \tag{6.7}$$

We also need to update the model rest length l state for each new frame in order to adapt the model to the new fitted pose. We consider both the original and the new

stable distance length for updating the rest length state. For two nodes p and q with the rest length l_{pq} and new distance $d(p,q)$, the updated rest length is as follows:

$$l_{pq} = \alpha.l_{pq} + (1-\alpha).d(p,q) \tag{6.8}$$

where the value of α is between 0 and 1. If α equals to 0 the rest length will totally depend on the new distance. The model may break its topology as the body segment lengths can iteratively change from the original lengths from the initialization. On the other hand, if α equals to 1 the rest length will totally depend on the initialization. Empirically, the initialization process here is very good therefore $0.5 \leq \alpha \leq 1$ can produce good result as well.

If we choose a very low value for α, the rest length l will be too sensitive to noise or any disturbance in the voxel data. Figure 6.3 shows a problem encountered when α is too small. Due to missing voxels around the knee, the knee node of the model was attracted to its closest data points in the middle of the thigh. The thigh segment adapted quickly to the data set by contracting its length and eventually warped the knee and ankle nodes inaccurately.

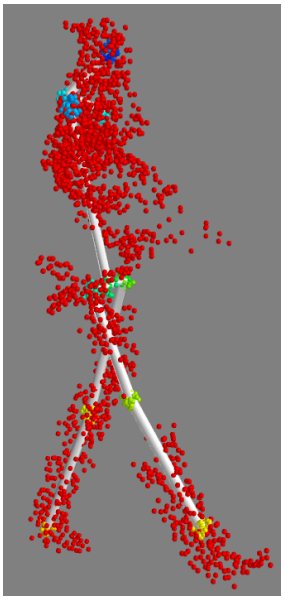


Figure 6.3: Rest length problem of the thigh due to missing voxels around the knee

6.3.5 Stability and stopping criteria

The model updating process will be terminated when the states are considered to be stable. We define a minimum threshold τ for $\mathbf{v}$ and $\mathbf{a}$ to stop the iterative process.

6.3.6 Heel strikes and CoH constraints

Based on prior knowledge, the hips (centre, left and right hips) and heel strike positions are set to be static during the iterative warping process. Therefore, the nodes corresponding to the hips and ankles will be initialised for each new frame by the data from the preprocessing result. Given the accuracy of CoH and heel strike in the preprocessing stage, these nodes need no update during the warping process.

6.4 Gait Signatures and Classification

6.4.1 Gait signatures

Gait signatures are created by selecting and combining the kinematic and structural features. We only use the lower legs motion for the kinematic features. We project the legs motion in the sagittal and frontal planes and then extract the motion angle of the thigh and shin. The kinematic feature is represented by a set of angles $\{\theta_{RT}, \alpha_{RT}, \theta_{LT}, \alpha_{LT}, \theta_{RS}, \alpha_{RS}, \theta_{LS}, \alpha_{LS}\}$ where θ, α, T, and S are sagittal angle, frontal angle, thigh, and shin respectively.

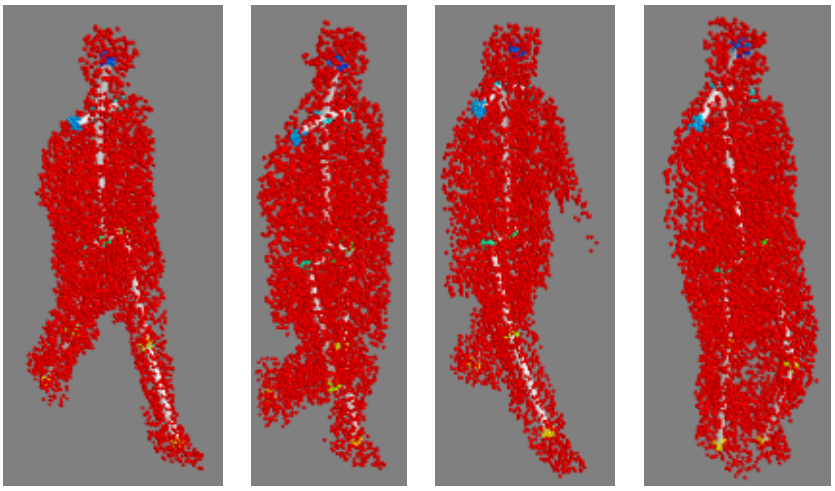
For gait kinematic features, we extract the gait signature using the Discrete Fourier Transform (DFT) and dynamic time warping (DTW). After applying the DFT to the kinematics data, we will obtain new information about the frequency components of the subject's gait. These frequency components will be used as a signature for classification. We used a Fast Fourier Transform (FFT) with 128 points. DTW can compare directly two kinematic angle trajectories. We are also able to extract the gait structural features such as height, stride length and footprint angle.

6.4.2 Classification

We classified the subject based on two different feature types:

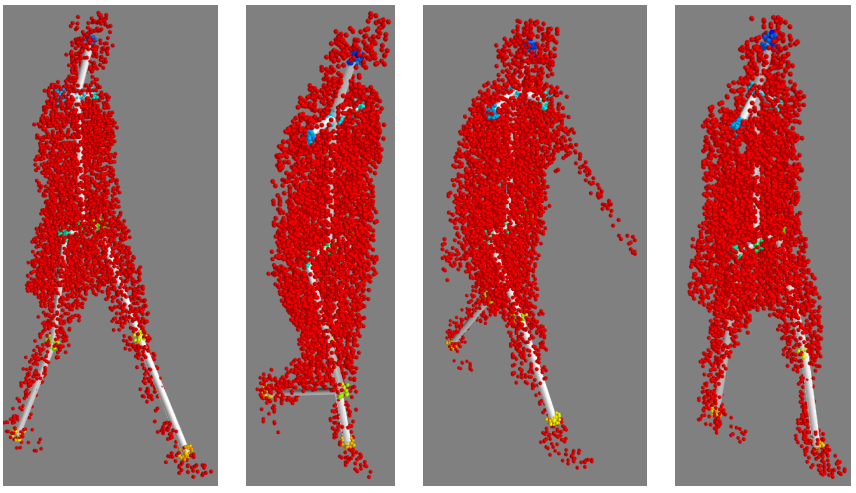
1. Structural features: height, stride length and footprint (direction and orientation)
2. Dynamic features: the frequency components and DTW of the kinematic angles

k-NN was used as a classifier with leave one out cross validation (LOOCV). Three different values of k were used: 1, 3, and 5. For the dynamic features we evaluated the dynamic time warping (DTW) distance metric and three different distances functions of the FFT components, i.e. Magnitude, Magnitude weighted phase and Euclidean.



(a) frame# 1 (b) frame# 7 (c) frame# 17 (d) frame# 28

Figure 6.4: The tracking result under normal voxel data



(a) frame# 1 (b) frame# 7 (c) frame# 17 (d) frame# 28

Figure 6.5: The tracking result under imperfect voxel data

6.5 Evaluation of Performance

We used the same dataset as in structural model evaluation which is the gait dataset from the Southampton multi-biometric tunnel containing 46 different subjects; four recordings of each subject were taken, giving a total of 184 samples.

6.5.1 Tracking system results

The constant parameter values used for our test were $s = 0.06$, $k = 20$, $m = 1$, $dT = 0.1$, $f = 0.8$, $\alpha = 0.5$, $\tau_a = 0.0001$ and $\tau_v = 0.0001$. Figure 6.4 shows the tracking result for sample frames where the model nodes are imposed into the voxel points. In this

sequence, one gait period consists of 36 frames. For complete tracking result for one period of gait can be seen in appendix A and YouTube URL: `http://www.youtube.com/watch?v=3PnbnX16now`.

The marionette model can work in slightly imperfect (partly missing) voxel data as shown in Figure 6.5. In this sequence, it has 33 frames in one gait period. Based on the result images, we can see that marionette model can warp into voxel data. To evaluate the overall marionette-model tracking performance, we need to determine the extracted kinematic angles from all sequences in the dataset.

Figure 6.6 and Figure 6.7 shows the statistical profile of the extracted kinematics of thighs in sagittal and frontal angle respectively. The blue solid curve shows the averages value whilst the red dashed line shows the standard deviations. The trend and pattern of the extracted kinematic angles in this experiment confirm the similarity with a result in biomechanics [87]. This result is quite similar to the same experimental result of structural model in Chapter 4.

6.5.2 Gait recognition analysis

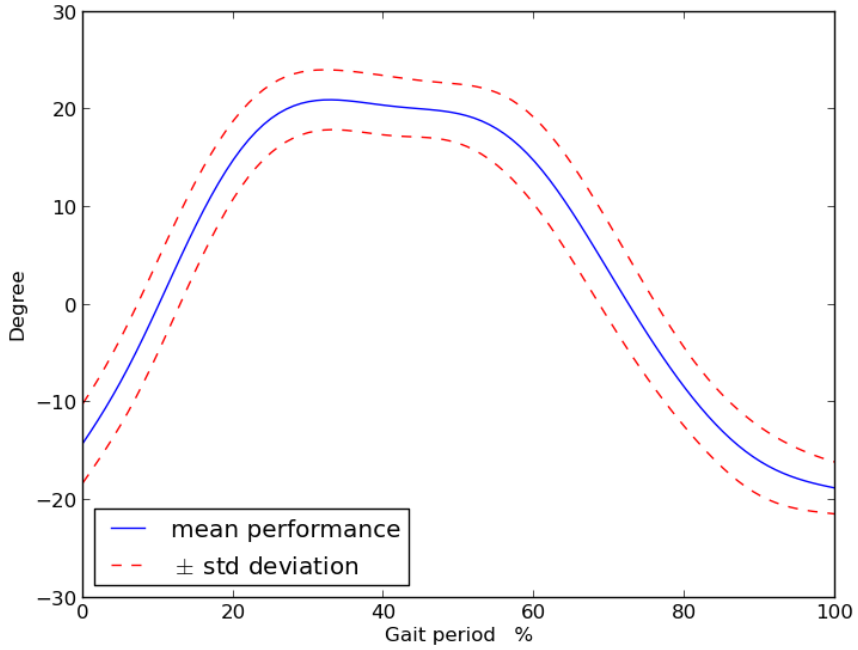
Table 6.1 shows the classification rate using structural features showing that height was the most discriminatory feature. The best performance of 64.6% was achieved for k=1 with height and footprint. It is interesting to note here that the footprint feature was able to increase recognition capability better than using the stride.

Chosen feature(s)	**CCR (%)**		
	k=1	**k=3**	**k=5**
Height	40.2	46.7	51.6
Stride	12.5	13.0	13.0
Footprint	27.7	27.7	25.0
Height & Stride	46.1	46.7	41.8
Height & Footprint	**64.6**	64.1	61.4

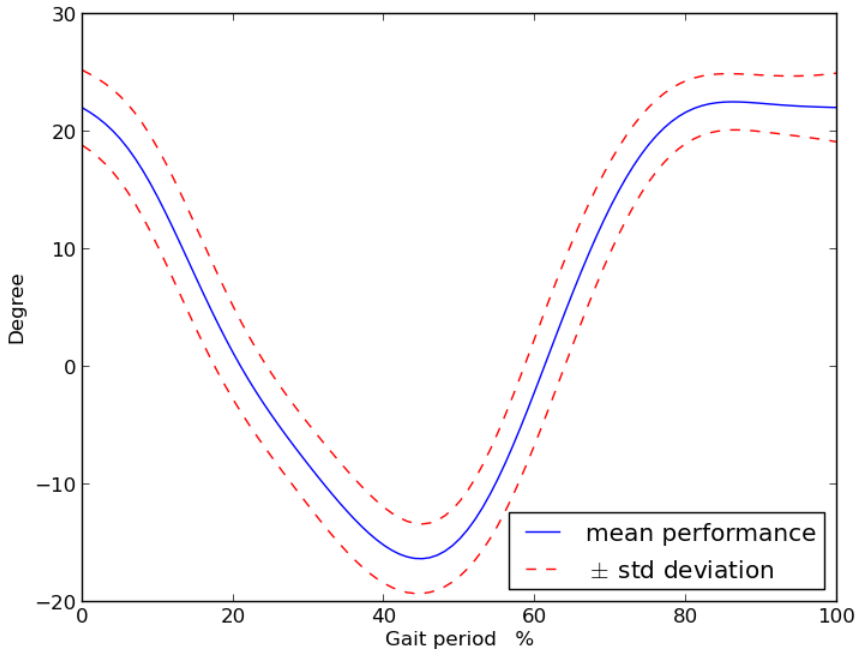
Table 6.1: k-NN classification results (%) for structural features

Classification result of kinematic features for each angle and each distance method are described in Table 6.2. The comparison of distance methods shows that the Euclidean distance metric to be best for individual kinematic angle recognition. The best recognition performance was achieved by using the angle α_{LT} which is the movement of the left thigh in the frontal plane, confirming an earlier study of the potency of the front view for gait recognition [25]. These results are very similar to the structural model results.

We also evaluated all possible feature subsets using all possible kinematic features combination. In this case, the eight kinematic angle features can generate 255 combinations



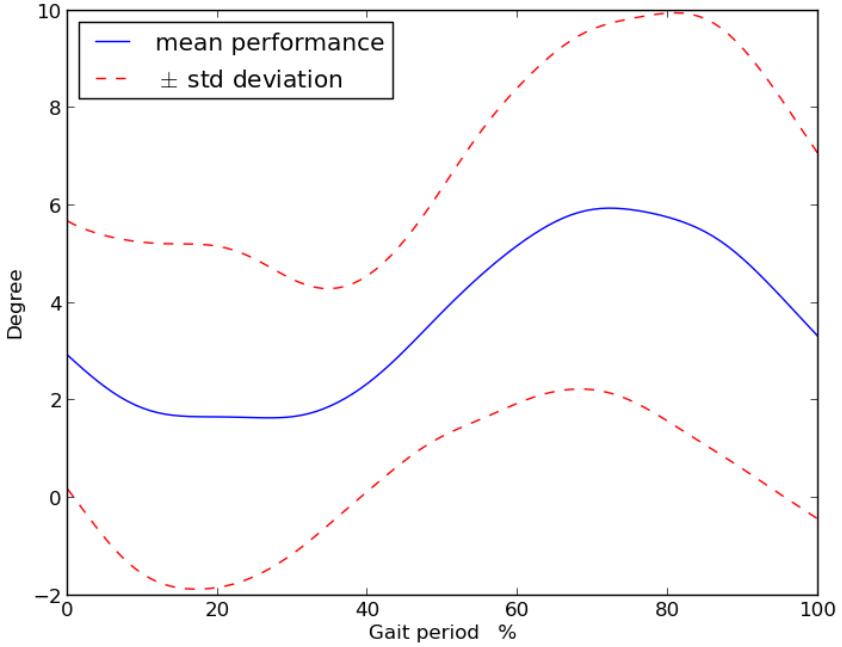
(a) Marionette kinematic trajectories - Sagittal θ_{RT} angles



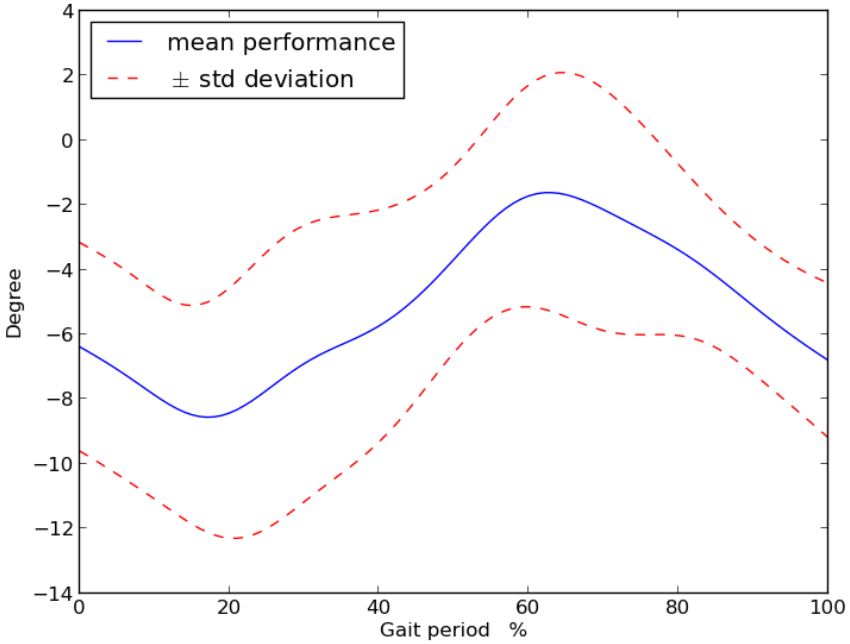
(b) Marionette kinematic trajectories - Sagittal θ_{LT} angles

Figure 6.6: The extracted kinematics of thigh sagittal angles using marionette model

of feature subsets. Again all classification experiments used leave one out cross validation. Figure 6.8 shows the feature subsets' recognition rate for each distance metric. From the image, we can see that DTW distance had the best performance. It had bigger recognition rate than the others for more than half of the subsets. Comparing this result in Figure 6.8 with the structural result in Figure 4.15, we can see that marionette model



(a) Marionette kinematic trajectories - Frontal α_{RT} angles



(b) Marionette kinematic trajectories - Frontal α_{LT} angles

Figure 6.7: The extracted kinematics of thigh frontal angles using marionette model

had better recognition rate in the beginning but then increased slowly. On the other hand, the structural result was more dynamic as it had an steady increasing trend and had better recognition rate at the end.

From Figure 6.8, we obtained the three best kinematic feature subsets as presented in Table 6.3. The best kinematic feature subset was a combination of all features

	Chosen distance method			
Chosen feature	**Mag**	**Mag-Phase**	**Eucl.**	**DTW**
θ_{RT}	38.0	29.8	44.5	34.78
α_{RT}	40.7	39.6	53.2	41.30
θ_{LT}	44.0	29.3	42.3	34.78
α_{LT}	44.5	42.3	**55.4**	39.13
θ_{RS}	38.5	27.1	35.8	23.91
α_{RS}	30.9	33.1	46.7	36.96
θ_{LS}	44.0	37.5	38.5	22.28
α_{LS}	34.7	38.0	50.0	33.15

Table 6.2: Correct classification rate (%) for marionette model kinematic feature with k=1

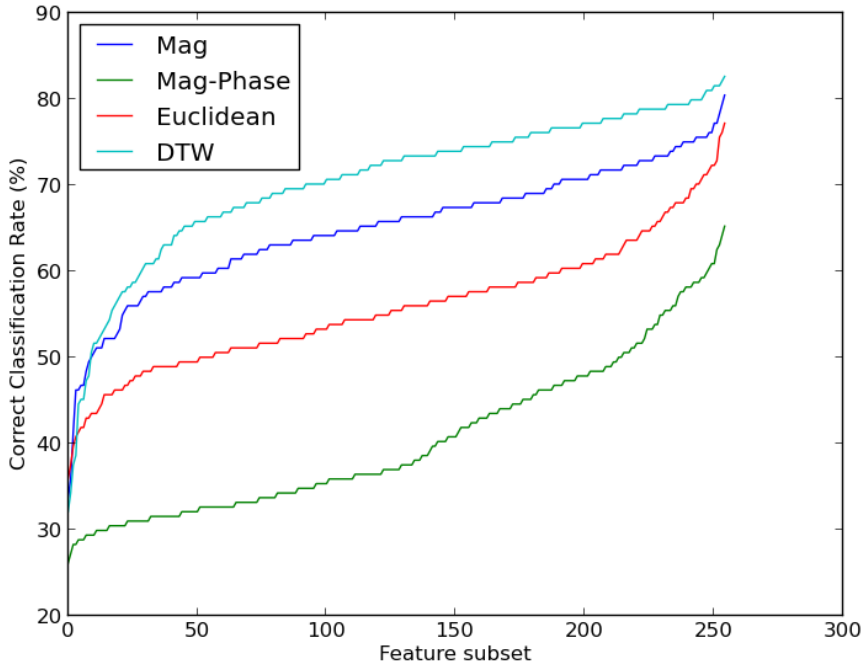


Figure 6.8: Marionette kinematic feature subsets' recognition rate for each distance metric

$\{ \theta_{RT}, \alpha_{RT}, \theta_{LT}, \alpha_{LT}, \theta_{RS}, \alpha_{RS}, \theta_{LS}, \alpha_{LS} \}$ with a 80.9% correct classification rate. This result is slightly lower than the best recognition rate of the structural model which is 86.4%. From these kinematic results, we found that good individual recognition rate of kinematic feature does not guarantee for performing better when they are combined into a single gait signature. The same situation also happened when we compare the DTW to the other distance metrics. DTW has the lowest individual recognition rate of the kinematic feature but it tends to perform better when it is used in a combined feature.

Although our best correct classification rate in marionette and structural models do not yet equal that achieved by a model-free approach [75], our system is directly related to the subject's gait and not their clothing or appearance and allows for deeper analysis of the gait mechanisms and can be related to biomechanical models of gait.

Features subset	**CCR (%)**
$\theta_{RT}, \alpha_{RT}, \theta_{LT}, \alpha_{LT}, \theta_{RS}, \alpha_{RS}, \theta_{LS}, \alpha_{LS}$	80.9
$\theta_{RT}, \alpha_{RT}, \theta_{LT}, \alpha_{LT}, \alpha_{RS}, \alpha_{LS}$	79.8
$\theta_{RT}, \alpha_{RT}, \theta_{LT}, \theta_{RS}, \alpha_{RS}, \theta_{LS}, \alpha_{LS}$	79.8

Table 6.3: Three best kinematic feature subsets

6.5.3 CMC and ROC analysis

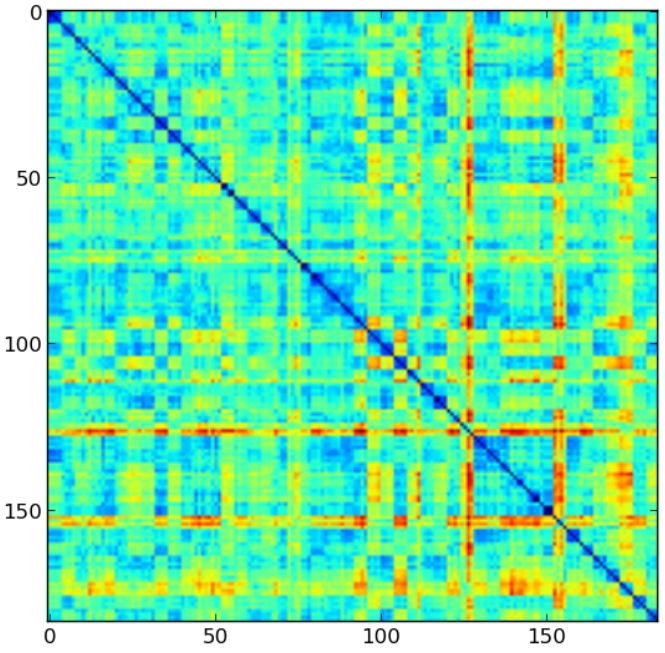


Figure 6.9: Similarity matrix of the best feature subset

We conducted an experiment for identification and verification analysis using CMC and ROC graphs. CMC and ROC are common evaluation methods for biometric systems. CMC is able to visualize the performance of identification task and ROC can describe the verification capability of the biometric systems. To generate CMC and ROC graphs, we set up an experiment using 184 sample sequences (46 subjects) as probe and gallery and only used the kinematic features. In CMC, we evaluated the identification rate with rank varying from 1 to 46. In this experiment, we also want to report the effect of the distance functions in term of CMC and ROC performance.

We decided to only evaluate the kinematic features as they have a much better recognition rate compared to the structural features. In order to generate CMC and ROC graphs, we need to compute similarity scores between the probe and the gallery data. In our experiments, we represented the similarity scores using distance functions. In our case, the smaller the distance has the better matching.

Figure 6.9 shows the similarity matrix of the best feature subset which contains all combined kinematic features, i.e. $\{\theta_{RT}, \alpha_{RT}, \theta_{LT}, \alpha_{LT}, \theta_{RS}, \alpha_{RS}, \theta_{LS}, \alpha_{LS}\}$. In this image, the blue color represents the good matches and the red one is for the bad matches. We have changed all values at the diagonal of the matrix to be the maximum in order to ensure that we do not count the recognition between the same samples (which are obviously having zero distance). Ideally, the darkest blue color areas (lowest distance) are just around the diagonal. The intra-class and inter-class distributions for the best feature subset are shown in Figure 6.10.

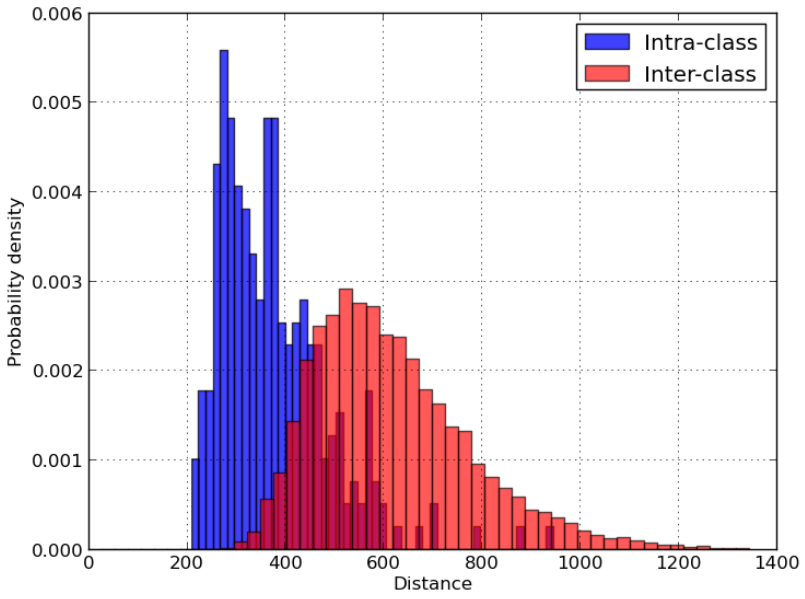
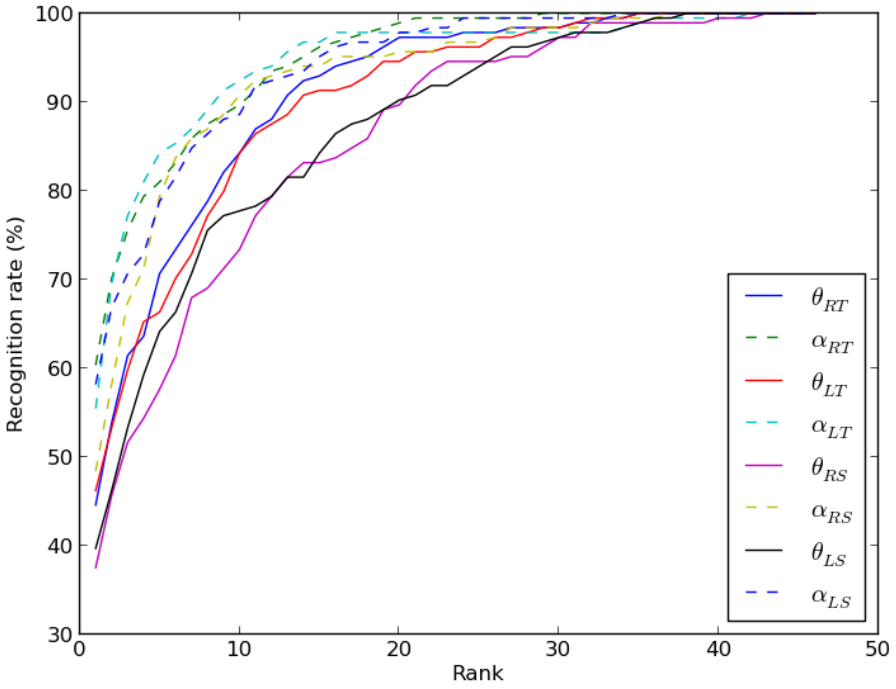
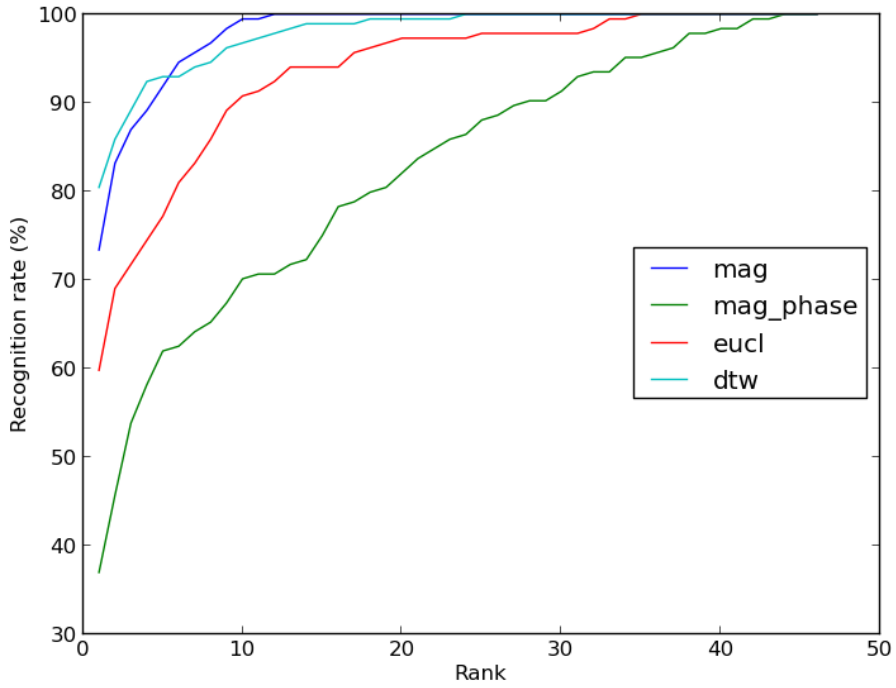


Figure 6.10: Intra/Inter-class variation for the best feature subset

Figure 6.11 shows the cumulative match characteristics (CMC) graph for recognition using kinematic features. The first graph, Figure 6.11(a), shows the performance for all individual kinematic angles using Euclidean distance. We chose Euclidean distance because this distance had given best CCR of the individual individual angle. From that graphs, we can see that the all four frontal angles had better CMC performance than the sagittal ones. The second graph, Figure 6.11(b), shows the CMC of all combined kinematic features. The best distance functions were Magnitude and DTW. The DTW distance was better at the beginning ranks but then it was lower than the Euclidean curve for ranks > 5. This result had slightly difference compared to the CMC result



(a) CMC of all individual kinematic angles using Euclidean distance function

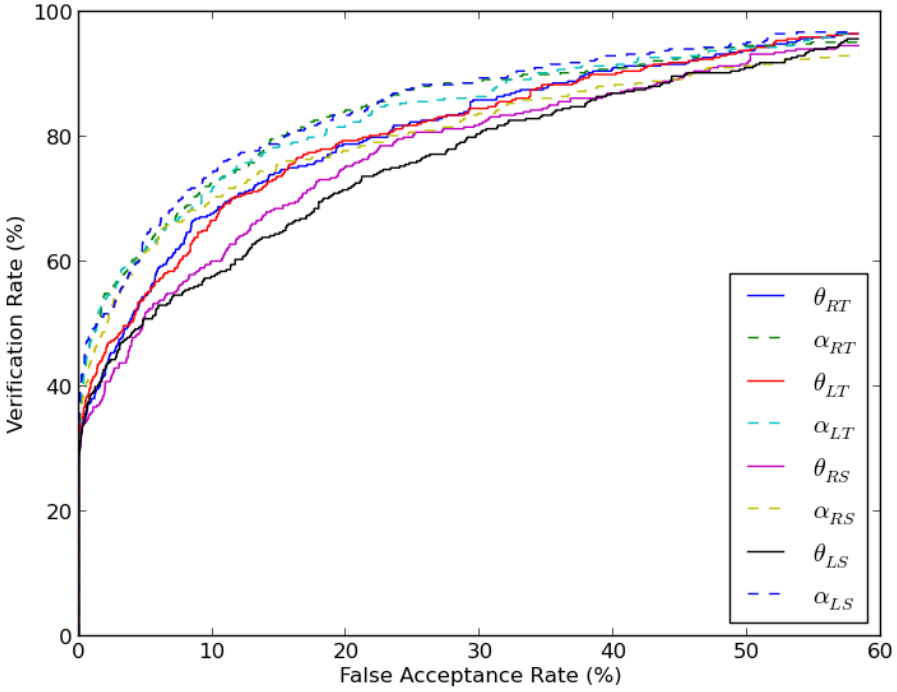


(b) CMC of a combination of all eight kinematic angles using four different distance functions

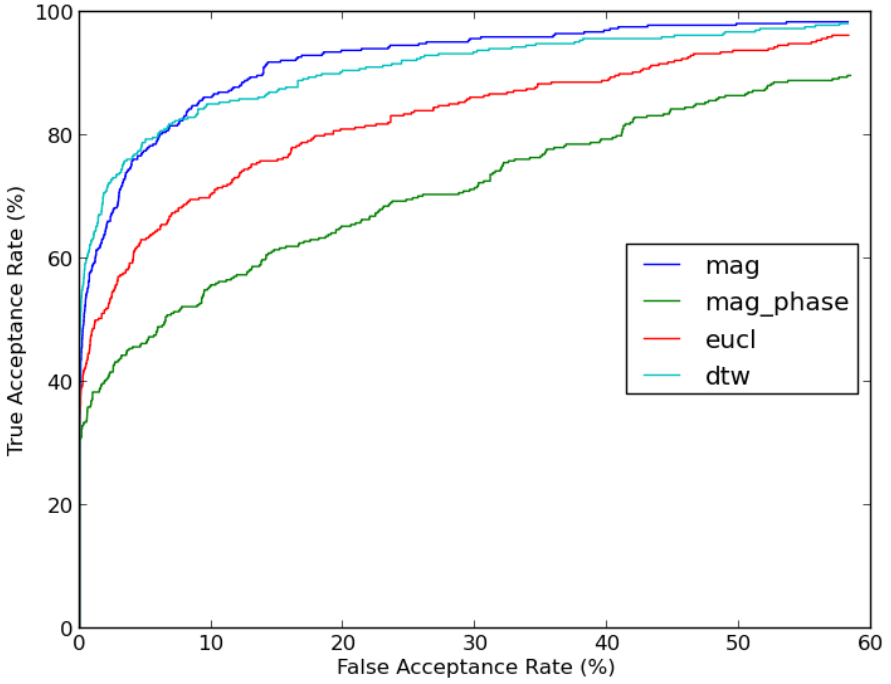
Figure 6.11: Cumulative match characteristics (CMC)

of the structural model; in the CMC of the structural model, the DTW distance was always the best.

The verification evaluation of our kinematic features is described using an ROC curve as shown in Figure 6.12. Figure 6.12(a) shows the ROC of all kinematic angles using



(a) ROC of all individual kinematic angles using Euclidean distance function



(b) ROC of a combination of all eight kinematic angles using four different distance functions

Figure 6.12: Receiver operating characteristic (ROC)

Euclidean distance. This graph shows a large range of performance variations within the kinematic features. This result was same as the ROC result of structural model in Chapter 4 where the frontal angles had better ROC than sagittal angles. Figure 6.12(b) shows the ROC curve of all combined kinematic features with four different similarity

distance functions. We can also see from the figure that the best verification performance was achieved by using Magnitude and DTW distance function. Both had quite high verification rate around 80% at very low FAR ($< 10\%$). On the other hand, the Euclidean and Magnitude-Phase distance had low verification rate.

6.6 Conclusions

We have described a novel model-based approach to 3D gait recognition using a marionette mass-spring model. The model was based on a marionette and mass spring system which represents the arrangement and interconnection of human vertices. In each frame, the model was warped to fit the voxel data of a 3D walking subject. Our initial results showed a good fit when tracking though a sequence of frames and that there was varying perspicacity in the measures derived by this model. The model was able to handle imperfect (or even missing) voxel data and the results encouraged further development along these lines. The evaluation of identification (CMC) and verification (ROC) suggested that the extracted kinematic features from marionette model have a quite good discriminating power even though it is still lower than the structural model's one. It also suggested that dynamic time warping and Magnitude distance functions are the best distance metrics when combining the kinematic features as a gait signature.

Chapter 7

Analysis and Consideration of the 3D Model-Based Approach

7.1 Intoduction

The main advantages of using model-based approaches in gait biometrics are that they can tackle some problems of occlusion, noise, low resolution, scale and rotation. Such problems can heavily degrade the recognition performance of appearance-based approaches, especially in 2D. In contrast, the model-based methods are relatively undisturbed and remain perform well.

In this chapter, experiments were planned to conduct performance analyses of both model-based and model-free approaches in handling some possible problems in the 3D voxel dataset. The first problem is about missing or corrupted volumetric data. This specific problem commonly arises due to poor cameras' calibration or imperfect background segmentation during the 3D visual-hull reconstruction. The second problem is about volume artefacts and occlusion. Calibration error and imperfect object segmentation also can introduce the volume artefacts problem in the 3D voxel data. For occlusion, it can happen especially in an environment where many subjects can interfere each other. Loose clothing and carrying conditions can also occlude the real subject appearances. Both volume artefacts and occlusion generate unwanted voxel around the real subject data.

Seely reported that the initial design of Southampton multi-biometric tunnel produced corrupted or even missing volume reconstruction that affected in the recognition performance [73]. Figure 7.1 shows a sequence of voxel data constructed in the old tunnel with significant areas of the volume missing. His investigation found that the configuration of cameras and their calibration's parameters gave big contributions for the poor reconstructed volume dataset. In the other experiments, Seely tried an outdoor environment

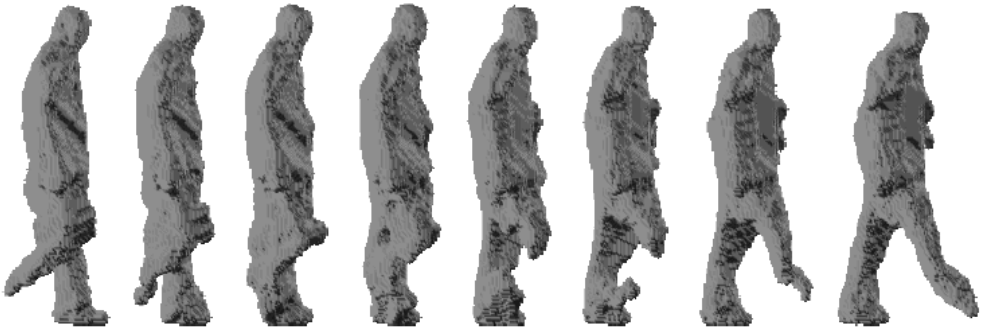


Figure 7.1: A sequence of voxel data with significant areas of the volume missing [76]

for 3D gait reconstruction and recognition. The quality of the extracted silhouettes was very bad and it had an effect of creating voxel artefacts and corrupted voxel. Eventually the artefacts and missing voxel ended up degrading the recognition performance.

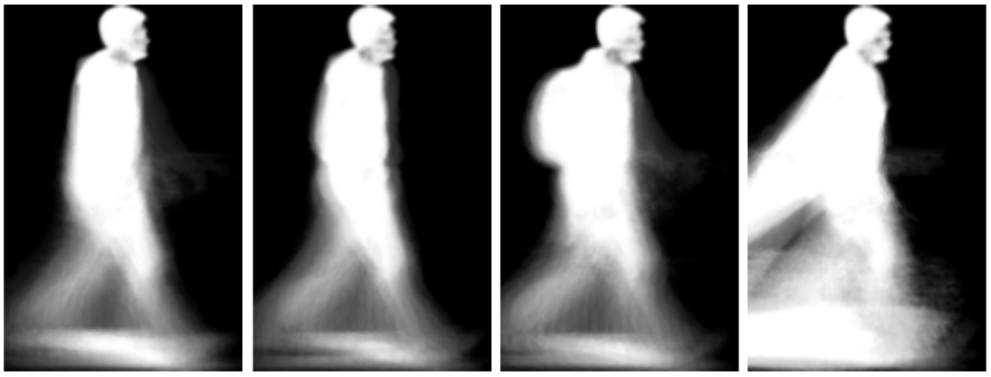


Figure 7.2: Silhouettes features with some occlusion problems in TUM-IITKGP database [34]

We are interested in analysing artefact and occlusion problems because these problems reduce performance but are unfortunately omnipresent. Especially in a real word surveillance scenario, occlusion occurs frequently. Hoffman et al. [34] presented a challenge of occlusions in gait biometrics by providing a new dataset with some varying conditions such as hand-in-pocket, backpack, gown, static occlusion, and dynamic occlusion. Figure 7.2 shows that carrying conditions and occlusion can introduce unwanted pixels into the silhouette-based extracted features.

In this work, the introduction of corrupted voxel and artefacts problems has an aim to analyse the benefits of model-based methods over model-free ones. Although the tunnel design is intended only for clutter-free and highly constrained environment, this experiment would give beneficial information for potential deploying of the tunnel in unconstrained environment or when the segmentation results are too noisy for reconstruction.

In the next sections, firstly we will evaluate a 3D model-free method using our dataset. In this experiment, we use a 3D averaged silhouettes model-free method from Seely [76] et al. since the method is simple but successfully achieved high recognition rate on 3D voxel data. After that we compare our structural model-based approach performance from chapter 4 to the performance of the model-free. We decided to use structural model-based approach in this comparison because it has similarity with an averaged silhouette method in model-free which is considered as a simple method but can achieve good recognition rate. Then we describe the excellent performance of both model-free and model-based approaches under normal condition. After that we introduce an artificial missing voxel and occlusion problems in our dataset which will affect the quality of gait features extraction and recognition performance. Finally, the performances of both methods will be analysed and compared.

7.2 Model-free Method

Here, we chose a 3D model-free method from Seely [76] et al. as the model-free baseline method. Seely conducted an evaluation on Soton 3D gait dataset using a model-free method in his PhD thesis. He employed a variant of 2D average silhouette method using a view-point projection technique to convert 3D voxel data into 2D view-invariant data. There were three different view-point projections used: side-on, front-on, and top-down projection. In the classification, Seely evaluated those three views and also a combination of them.

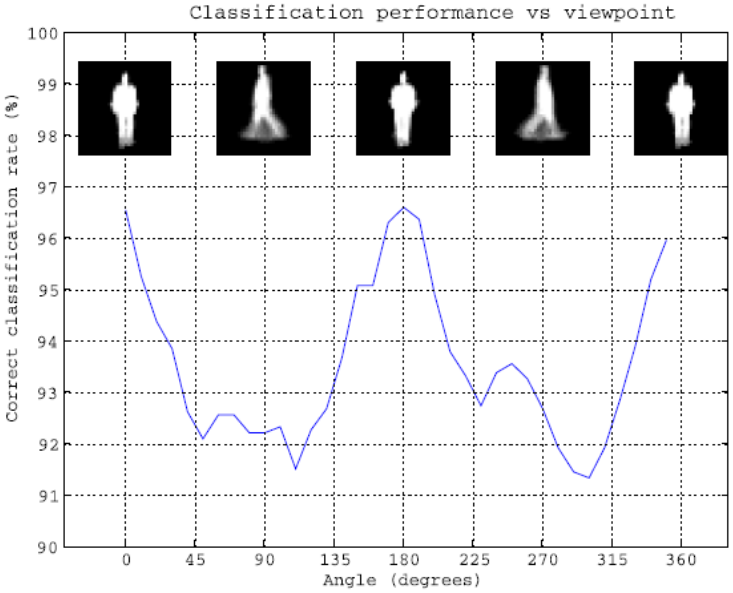


Figure 7.3: Classification performance with varying viewpoint [76]

He also conducted an experiment for recognition evaluation at any arbitrary viewpoint. Figure 7.3 shows the graph of the recognition rate in the y-axis versus viewpoint angle

in the x-axis. The results suggest that frontal viewpoint has much better performance compared to sagittal viewpoint. The other viewpoints have performance in between of these two viewpoints.

The evaluation of its ROC performance is shown in Figure 7.4. The ROC has an average of 95% verification rate at 10% false positive. The best performance was achieved using a combination of all three views which is around 95% true positive at only 5% false positive.

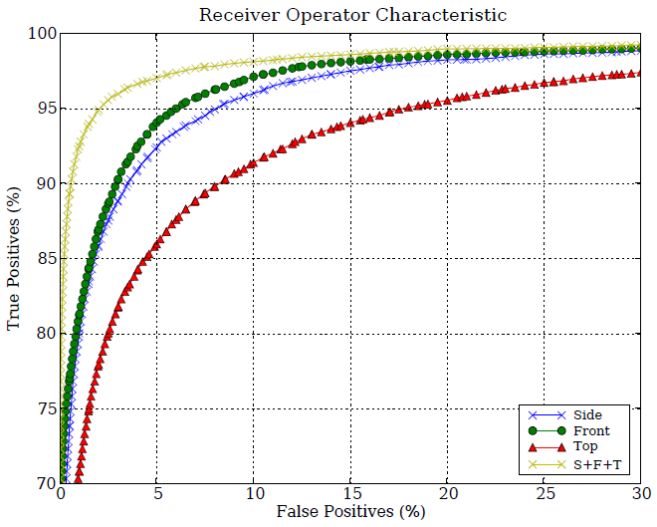


Figure 7.4: ROC of model-free [76]

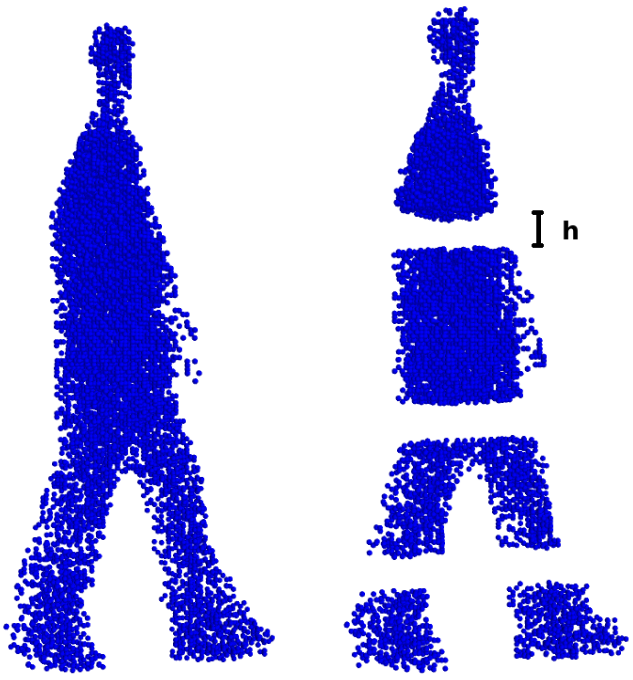
7.3 Analysis on Normal Dataset

In this experiment, we evaluated 3D structural model-based and 3D averaged silhouette model-free methods then compared and contrasted the results. We used the same dataset, which contains 46 subjects and 186 sequences. In evaluating an recognition experiment, the gallery and the probe data were from normal dataset.

Table 7.1 shows the correct classification rate (CCR) for both methods. As can be seen here, the model-free method achieved very high recognition rate especially using combinational views with 99.4% recognition rate. The performance of model-based methods was also good as it was more than 90% of correct classification rate. Both methods achieved the best classification results using k-NN when k=1. From this data we can conclude that under normal condition the model-free approach has an excellent recognition rate and perform better than the model-based method. In the next section we analyse the effect of corrupted voxel data to the recognition performance.

Method	**CCR (%)**		
	k=1	**k=3**	**k=5**
Model-Free (Front)	98.9	98.3	95.6
Model-Free (Side)	98.9	98.3	93.4
Model-Free (Top)	95.1	95.1	90.7
Model-Free (F+S+T)	99.4	99.4	96.7
Model-Based (kinematics)	86.4	83.1	79.3
Model-Based (kinematics + CoH)	95.1	90.7	89.6

Table 7.1: Recognition performance under normal condition



(a) Subject from normal data (b) Subject from corrupted data

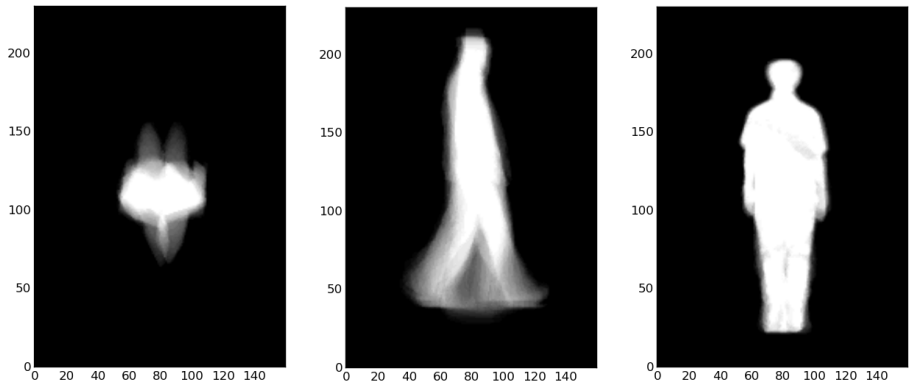
Figure 7.5: Comparison between normal and corrupted data

7.4 Analysis on Corrupted Dataset

We did an experiment of corrupted dataset by introducing missing and corrupted voxel data. We generated three horizontal 3D strips at three different vertical locations randomly. The strips have a parameter i.e. height. Based on those strips, we removed all voxel in the data which had intersection with the strips. Eventually, it ended up creating an artificial corrupted and missing dataset in which emulates the real condition of imperfect background subtraction and poor camera calibration in visual hull reconstruction. The higher the strips means the voxel are more corrupted. Figure 7.5 shows a subject in two different conditions. Figure 7.5(a) in the left image is a view of normal data and Figure 7.5(b) shows a subject data after being corrupted by 3D strips. We can

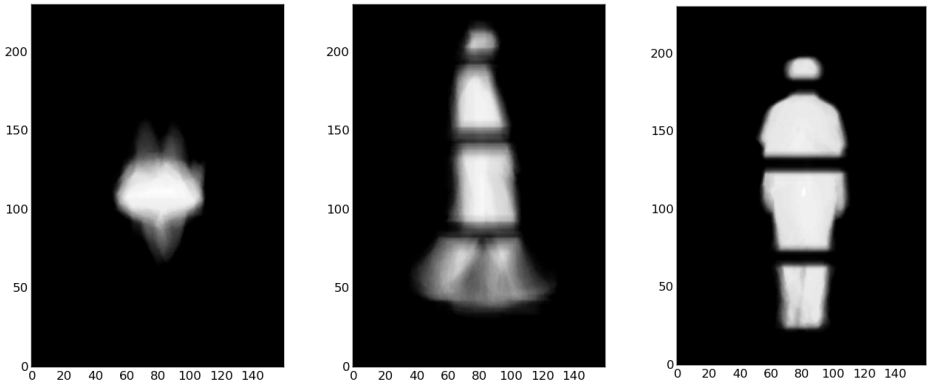
see that there are three horizontal strips of missing voxel data on it. In that image, h means the height of the strips in voxel unit.

Before we evaluate the classification performance, we are interested to see the difference in the quality of the extracted features between the normal and the corrupted dataset. The model-free method used an average silhouette as its main feature. The average silhouettes from clean normal dataset are shown in Figure 7.6. There are three images representing top, side, and frontal views. After introducing the missing voxel into the dataset, the new extracted silhouettes have some missing regions as shown in Figure 7.7. From those images we can see that there are quite noticeable differences between the features of normal and corrupted data, except for the top view averaged silhouettes. The top view silhouette relatively remains the same or unaffected by corrupted voxels since they project the voxel data vertically in the transversal plane. If we introduce the other 3D strips which vertically corrupt the data, it will affect the top view averaged silhouettes.



(a) Top side (b) Sagittal side (c) Frontal side

Figure 7.6: Averaged silhouettes from normal voxel data



(a) Top side (b) Sagittal side (c) Frontal side

Figure 7.7: Averaged silhouettes from corrupted voxel data

In the model-based method, we compare the extracted kinematic angles. Figure 7.8 shows the averaged sagittal angle of right thigh θ_{RT} from both the normal and corrupted dataset. In the model-based extracted features, it seems that there is slightly difference.

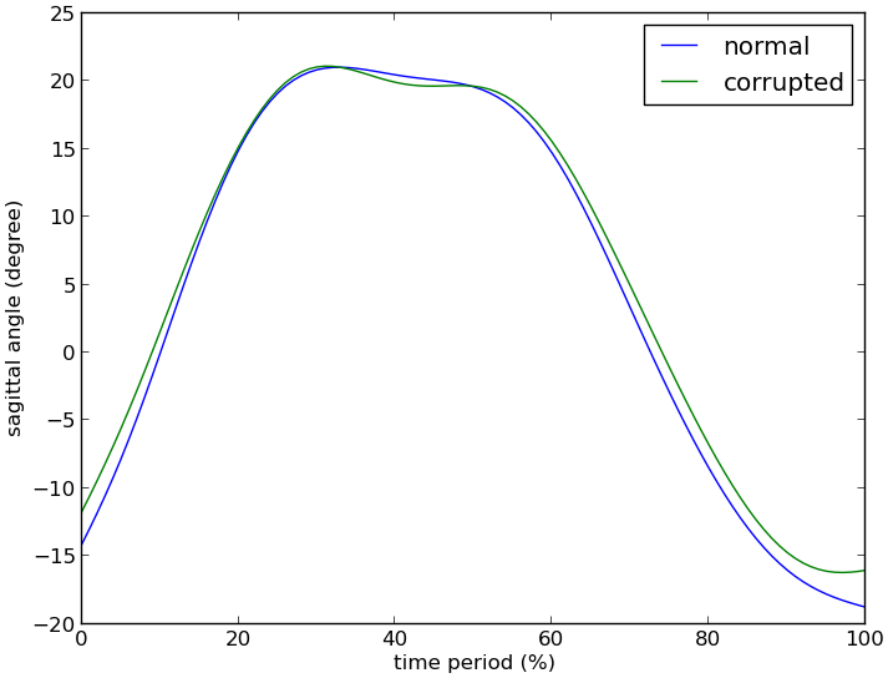


Figure 7.8: Extracted right thigh sagittal angle θ_{RT} from the normal and corrupted dataset

In this experiment, we used three different values for the 3D strips height parameter (h), i.e 5, 10 and 15 voxel units. Table 7.2 and Figure 7.9 shows the correct classification rate (%) for each method over various corrupted voxel data indicated by a height parameter h. From the results we can see that at $h = 15$, the front, side and a combination view of model-free dramatically failed to achieve recognition task. This happens because the generated silhouettes on these views were highly corrupted. On the other hand, the top view of model-free still performed very well despite the occurrence of the problem. Due to the property of the problem in the dataset that only corrupt the horizontal strips, the top projections can still manage to fill up those missing voxel holes and it is the reason why the top view model-free can still survive in this poor situation. In the model-based results, we can see from the table that the model-based performance was also degraded, but it was only slightly.

From Figure 7.9 we can see that model-free methods except the top view one, have big negative slopes especially in the very beginning of height h. On the other hand, from $h = 0$ to 5, the model-based method remain unaffected with the problem. Then slowly in the range of $h = 5$ to 15, the model-based performance slightly dropped from 92.9% to 78.8% recognition rate. In overall, the result shows that the model-based method can handle missing voxel problem better than the model free approach.

Method	CCR k=1 (%)			
	h=0	h=5	h=10	h=15
Model-Free (Front)	98.9	69.0	25.0	12.5
Model-Free (Side)	98.9	68.4	31.5	16.8
Model-Free (Top)	95.1	96.1	94.5	90.7
Model-Free (F+S+T)	99.4	84.2	35.3	18.4
Model-Based (kinematics)	86.4	84.7	76.0	65.7
Model-Based (kinematics + CoH)	95.1	92.9	88.0	78.8

Table 7.2: Recognition performance over missing voxel dataset

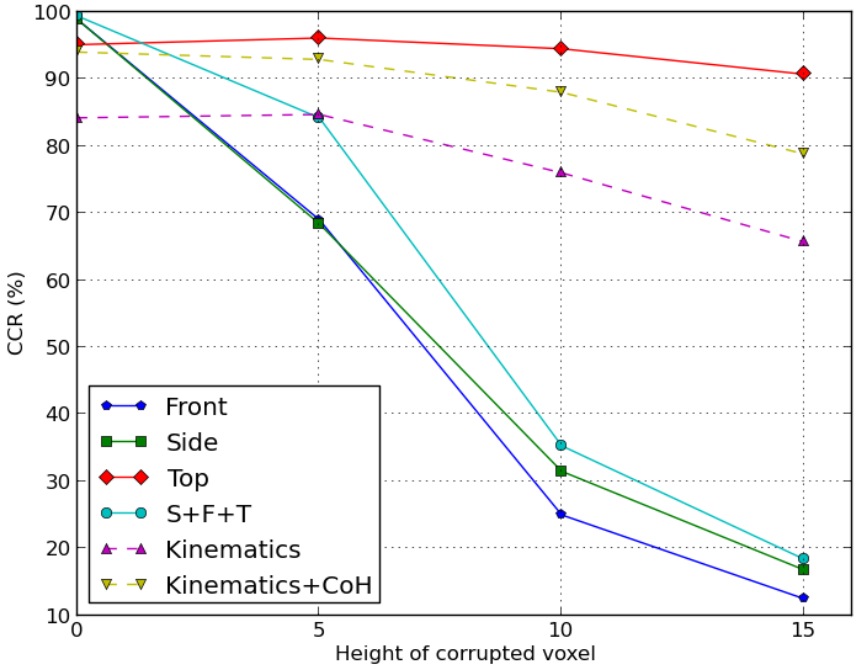
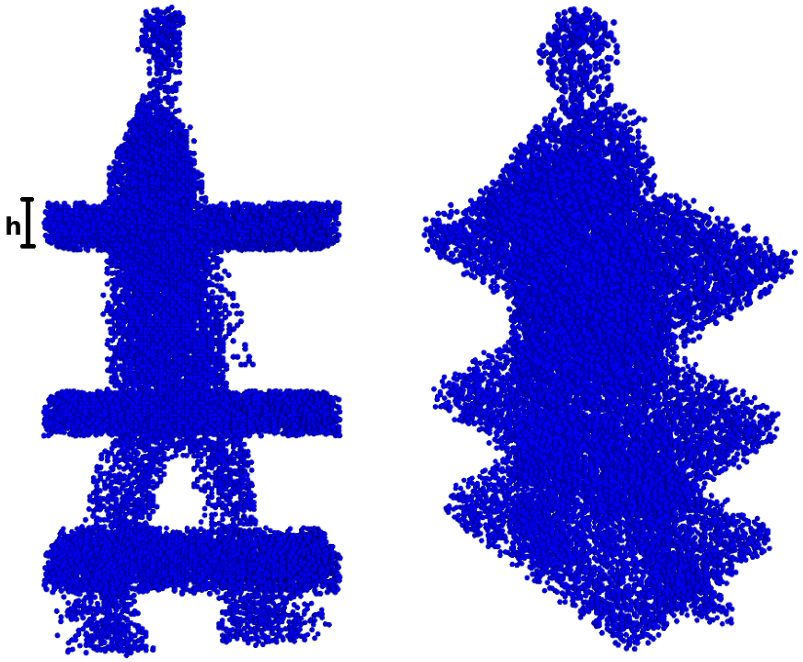


Figure 7.9: Recognition performance over missing voxel dataset

In conclusion, under missing voxel condition the model-free approach suffers heavily. However, the availability of top view data can help to maintain the original performance. It is essential to note that the top view silhouette is only available if we work with 3D data. When we work with 2D data, we are never able to extract the top view of the subject. The results also confirm that the model-based approach can handle this corrupted voxel problem well with little performance degradation.

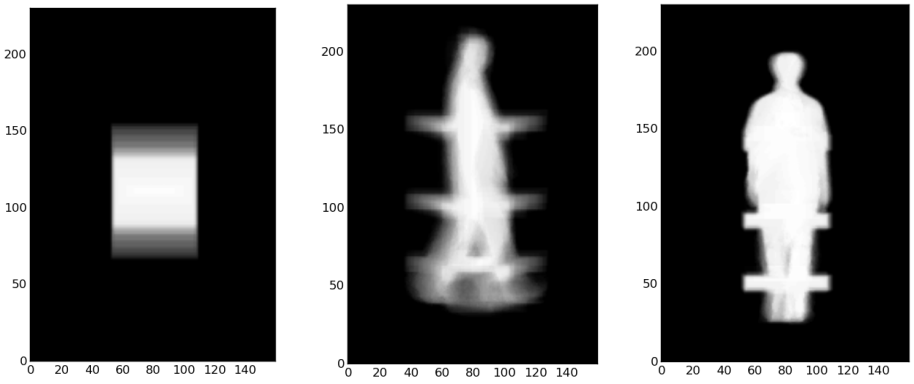


(a) Occluded subject at sagittal view (b) Occluded subject in 3D view

Figure 7.10: Voxel data with cuboids' artefacts

7.5 Analysis on Occluded Dataset

An experiment was advised to evaluate recognition performance under occlusion. We created an artificial occluded dataset by adding three 3D cuboids at three different vertical locations randomly. The cuboid' volume is defined with a height parameter while its area is always derived from subject 3D bounding box. Figure 7.10 shows an occluded data with three cuboids. From sagittal view in Figure 7.10(a), it looks like we have added three strips on it. However, in 3D view of Figure 7.10(b) we can see clearly that three cuboids have been occluded into the dataset.



(a) Top side (b) Sagittal side (c) Frontal side

Figure 7.11: Averaged silhouettes from occluded voxel data

The extracted averaged silhouettes for occluded data are shown in Figure 7.11. All views are affected and there are some areas of unwanted pixels around the real subject image. We can also see that the frontal view silhouette is also affected by occlusion problem. Its shape is no longer similar with its feature image in normal dataset,even it has changed quite dramatically to become a square silhouette.

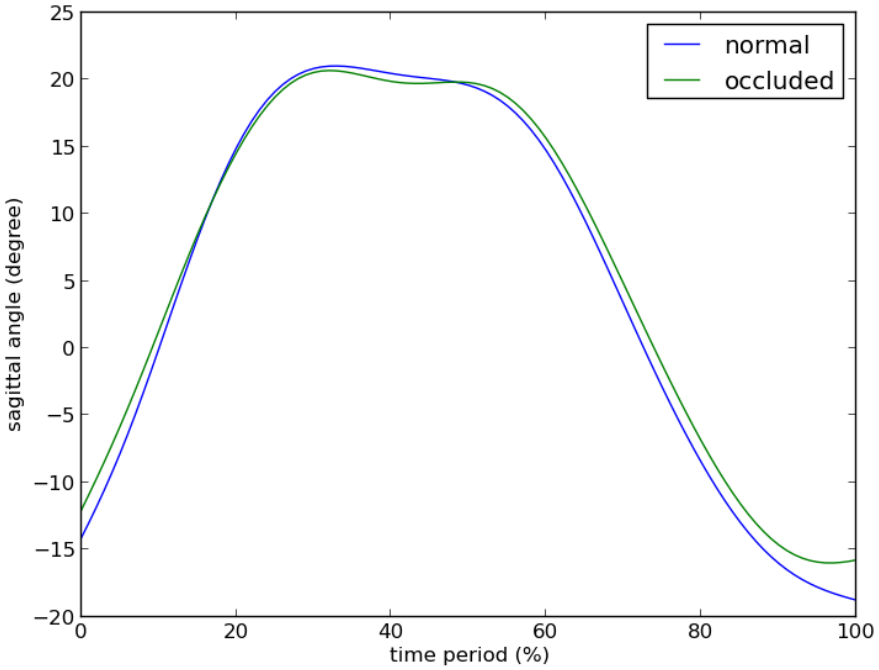


Figure 7.12: Extracted right thigh sagittal angle θ_{RT} from the normal and occluded dataset

In the model-based method, as shown in Figure 7.12, the averaged kinematic path from occluded dataset has the same trend as the normal one but it has slightly different in the starting and ending angles. The normal path has a staring angle around $-15°$ while the occluded one only just under $-10°$.

In order to analyse the effect of occluded dataset, we have evaluated three different values of height parameter, i.e 5, 10 and 15 voxel units. Table 7.3 and Figure 7.13 shows the correct classification rate (%) for each method over various occluded voxel data indicated by a height parameter h. From the results we can see that at $h = 15$, all methods were considerably disturbed and only performed below 50% of recognition rate, except for top view method that had 58.1% rate. It is interesting to notice that the top view initially dropped its performance at h=5, but then maintain its performance for the higher value of h. This can happen because we used a static top view area in the cuboids. If we employed additional parameters in the cuboid, i.e. length or width of cuboid, the top view performance will vary. In the model-based results, we can see from the figure that the model-based performance was also degraded quite heavily but have a linear and smaller negative slope which means the rate of degradations is still better than the model-free ones.

Method	**CCR k=1 (%)**			
	h=0	**h=5**	**h=10**	**h=15**
Model-Free (Front)	98.9	83.6	48.3	36.4
Model-Free (Side)	98.9	54.3	22.2	14.1
Model-Free (Top)	95.1	58.1	58.1	58.1
Model-Free (F+S+T)	99.4	86.4	45.1	26.0
Model-Based (kinematics)	86.4	59.2	43.4	32.0
Model-Based (kinematics + CoH)	95.1	78.2	61.4	41.8

Table 7.3: Recognition performance over occluded voxel dataset

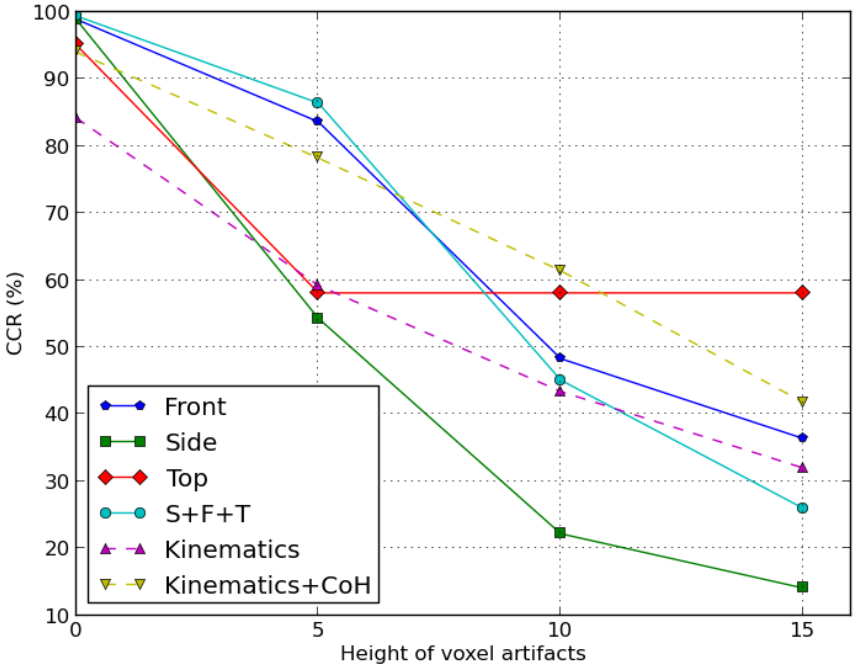


Figure 7.13: Recognition performance over occluded voxel dataset

7.6 Conclusions

After comparing the performance of 3D model-based and 3D model-free methods, we can conclude that generally both model-based and model-free methods work very well in the normal voxel dataset and to some level they can handle some problem of corrupted and occluded dataset. However, the model-based method has a better immunity to those problems indicated by its lower degradation slope. We consider and prefer to use the model-based method when the data quality is poor either being corrupted or occluded. On the other hand, the model-free is very simple method and a better choice in normal dataset as it can give very good performance.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

We have developed the first model-driven approaches to gait biometrics for use with 3D data and relatively large number of subjects. There were two 3D models (structural and marionette mass-spring) used to extract the human gait kinematics. The structural model consisted of articulated cylinders and used structural gait parameters such as height, stride and footprint angle (the latter being unique to 3D data and is not available in data wherein the subject walks in a plane normal to that of the camera). We used the structural model to fit the 3D volumetric subject data, and this provided the inclinations of the shin and thigh in sagittal and frontal planes. By Fourier analysis and dynamic time warping of the angles' trajectories, recognition capability can be achieved.

The second approach was a novel 3D marionette mass-spring model. To model the articulated human body, we used a stick-figure which emulates the marionettes' motion and joint structure. The stick-figure had eleven nodes representing the human joints of head, torso, and lower legs. Each node was linked with at least one other node by a spring. The voxel points in the next frame had a role as attractor which able to generate forces for each node and then iteratively warped the model into the data. This process was repeated for successive frames for one gait period.

We have done some experiments on these two models and evaluated the results. Both the models were able to extract static and dynamic gait features. The static gait features included height, stride length and footprint angle, whereas the dynamic features mainly covered kinematic angles of lower leg body parts. The structural model achieved up to 86.4% correct recognition rate only by using the kinematic angle trajectories. After the refining process with GA, the structural model even performed 89.6% recognition rate. The marionette model showed good fit tracking though a sequence of frames and that there was varying perspicacity in the measures derived by this model. The

model was able to handle slightly imperfect and occluded voxel data and the results encouraged further development along these lines. In term of recognition performance, the marionette model achieved slightly lower performance than the structural model. Comparing the performance and computational speed results, the structural model had better recognition performance but needed much computational time to accomplish it.

The experimental results showed that dynamic features are more important than the structural information as they had a greater recognition rate. On the structural features, the overall best recognition rate was at 64.6% and achieved by a combination of height and footprint angle. On the other hand, the dynamic features achieved up to 95.1% recognition rate. Based on the dynamic feature subsets evaluation, we can conclude that the frontal view of the human gait has a better discriminatory capabilities compared to the sagittal view.

Based on the evaluation of the corrupted and occlusion dataset, we can conclude that 3D model-based approaches have better immunity and are still able to extract the kinematic features. Whilst in model-free approaches, the degradation of 3D dataset significantly decrease their performances due to a strong dependency on the silhouette quality.

Clearly results could be improved by a more sophisticated classifier, but the basic results here confirm that discriminatory capability can be achieved by these new approaches. The approaches also reveal a selection of measures which are new to gait analysis. Naturally, the model-based approaches could be used to complement the discriminatory potential of the model-free approach.

8.2 Future Work

We have explored the potential of using 3D volumetric data in gait biometrics. In order to improve our current work in this thesis and to give more insight about the performance and contribution of 3D gait biometrics, we have three important things in mind to enhance. First is about the tunnel and the quality of its dataset. Secondly it is about the current methods of model-fitting and the last one is about the recognition process.

The University of Southampton multi-biometric tunnel is a constrained environment that is designed with airports and other high throughput environments in mind. However, the number of subject passing through the tunnel is currently limited to one subject at a time. In the future, we suggest that the system should be able to capture and reconstruct two or more subjects walking across the tunnel. Moreover this type of dataset should also be available for future research. To adapt to a multi objects dataset, our tracking system must be modified so that it will be able to achieve a good initialisation and handle multi-object tracking at the same time. While waiting for the new multi object

dataset from the tunnel, we can combine two subjects from two different sequences and then simulate them walking together at the tunnel. This new simulated data can be regarded as the multi object 3D gait dataset.

Even though the tunnel had successfully generated 3D voxel data from the tunnel, the current 3D voxel data are quite coarse due to the nature of visual hull reconstruction algorithm. To some extent, the 3D data also still have occlusion problems between the adjacent body segments and some corrupted data. Based on the results' analysis, the features extraction system tends to be likely fail when handling such poor data. The performance of any recognition system is ultimately dependant on the quality of the original source data. Therefore, it will be worthy and boost the recognition performance if we are able to improve the quality of current 3D gait dataset with more sophisticated reconstruction algorithm.

In term of reliability, performance, and efficiency of model-based gait biometric algorithms, some future work can be done to improve matters. We can modify the current system to extract some other potential gait feature such as torso periodic movement, and head pose. Those features might be able to increase the recognition performance. We can also try to use and evaluate some other distance metrics in the correlation filter. The current one was chosen with the main consideration of fast computation. There are some complex distance metrics that use both surface and voxel data [21, 61, 62]. The surface data itself can be generated from voxel data.

In order to increase the efficiency and reliability of the system in normal or especially clutter environment, we can define a specific motion model for each kinematic angle. The motion model can be extracted by using simple statistic tools from the collected and manually edited kinematic data or by defining an autoregressive moving average model. Moreover this motion model would be integrated in a framework such as the particle filter or CONDENSATION algorithm [36].

Working directly in 3D data can exploit the benefit of heel strike information. The marionette model has utilized this information but not for the structural model. As described in Chapter 4, the heel strike location can be extracted easily and accurately by aggregating footprints images. This heel strike location is assumed to be static during the stance phase of walking; therefore we can fit both thigh and shin together at the same time. The CoH and heel strike location can be treated as hard constraints which limit the possible movement of the thigh and shin respectively. Consequently, evaluating the thigh angles will automatically define the corresponding angles in the shin. Once we define the thigh angles, we can determine the shin correspondence angles. This can gain the advantages of more accurate fitting during stance phase as well as faster computational time.

Finally, in the current recognition process we use 3D data for both gallery and probe data. This process is called 3D-3D gait recognition. In the next research, it would be

interesting to see a performance of 3D-2D gait recognition system. The system will use gallery data with 3D gait features as been collected here and then use 2D gait video sequences as probe data. The 3D-2D gait recognition is very suitable with current value and it will be demanding because so many gait data sources are currently available from 2D CCTV videos. Although the 2D gait video has a problem of view dependence, there have been some successful research of tracking, estimating and extracting the human pose from monocular 2D video [41, 47, 56, 80, 81].

Appendix A

Anthropometric Measurements of the Human Body

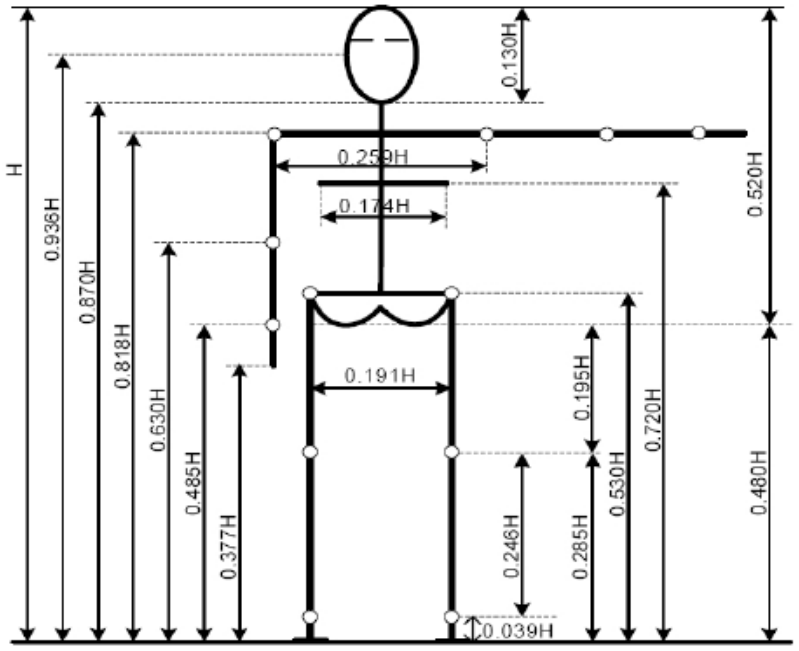


Figure A.1: Human body anatomical priors

Appendix B

Model-Fitting Source Code

Correlation Energy Map

```
def getThightCCMap(gait,multires=True):
    '''
    Get the Correlation Map for All Frames in the Sequences.
    gait    : gait class from lib.data
    res     : resolution types
    OUTPUT: CCMAP=[RTccMap LTccMap]
    '''
    c=gait.getPeriode() #1 period gait cyle
    if (gait.pointList != None):
        dataPointList = gait.pointList
    else:
        dataPointList = gait.getPointListSegmented(c)
        gait.pointList = dataPointList
    #dataPoinList= [pointRT, pointLT, point RS, pointLS]
    comData = gait.openCoMandHeight()
    hList = comData['height'][c[0]:c[1]]
    comList = comData['com'][c[0]:c[1]]
    RTCCMap=[] #RightTHIGHT CCMap
    LTCCMap=[] #LeftTHIGHT CCMap
    l=len(dataPointList)
    for i in xrange(l):
        model = HumanModel(hList[i], comList[i]) # model   : human model class from lib.model
        if(multires):
            RTCC=CCMapMultiDP(dataPointList[i][0], model.RT)
            LTCC=CCMapMultiDP(dataPointList[i][1], model.LT)
        else:
            RTCC=CCMap(dataPointList[i][0], model.RT)
            LTCC=CCMap(dataPointList[i][1], model.LT)
        RTCCMap.append(RTCC)
        LTCCMap.append(LTCC)
    return np.array(RTCCMap), np.array(LTCCMap)
```

```
def getShinCCMap(gait,thightPath,motion=None,multires=True):
    '''
    Get the Correlation Map.
    gaitData    : Class gait from lib.data
    gaitModel   : Class human model from lib.model
    motion      : motion model
    res         : resolution types
    thightPath: [RTPath, LTPath], XXPath:[theta,alpha]
    OUTPUT: CCMAP=[RSccMap LSccMap]
    '''
    c=gait.getPeriode() #1 period gait cyle
    if (gait.pointList != None):
        dataPointList = gait.pointList
    else:
        dataPointList = gait.getPointListSegmented(c)
        gait.pointList = dataPointList
    #dataPoinList= [pointRT, pointLT, point RS, pointLS]
    comData = gait.openCoMandHeight()
    hList = comData['height'][c[0]:c[1]]
    comList = comData['com'][c[0]:c[1]]
    RSCCMap=[] #RightTHIGHT CCMap
    LSCCMap=[] #LeftTHIGHT CCMap
    l=len(dataPointList)
    for i in xrange(l):
        model = HumanModel(hList[i], comList[i])
        model.initThightPose(thightPath[i])
        if(multires):
            RSCC=CCMapMultiDP(dataPointList[i][2], model.RS)
            LSCC=CCMapMultiDP(dataPointList[i][3], model.LS)
        else:
            RSCC=CCMap(dataPointList[i][2], model.RS)
            LSCC=CCMap(dataPointList[i][3], model.LS)
        RSCCMap.append(RSCC)
        LSCCMap.append(LSCC)
    return np.array(RSCCMap),np.array(LSCCMap)

def CCMap(dataPoint, model, weighted = False):
    '''
    CCMAP generates the correlation energy map for one frame.
    dataPoint: data
    model: cylinder points from our model
    weighted: put weight for the outer model's points
    '''
    tree = KDTree(dataPoint)
    theta = scipy.linspace(-45,45,15)
    alpha = scipy.linspace(-7.5,7.5,15)
    error = np.zeros([15,15]) # energy map saved here
    mWeight = model.getWeight(10) # set weight = 10
    for i in xrange(len(theta)):
        M1 = generateTransfromMatrix(theta[i],axis='x')
        for j in xrange(len(alpha)):
            M2 = generateTransfromMatrix(alpha[j],axis='y')
            model.setTransform(M1)
            model.updateTransform(M2)
            dist,_ = tree.query(model.applyTransform())
            if (weighted):
                dist = dist * mWeight
            error[i,j] = sum(dist*dist)
    return error
```

```
def CCMapMultiDP(dataPoint, model, weighted = True, motionModel=None):
    'MultiResolution Approach'
    t1s=-45     #theta1start
    t1e=45              #theta1end
    a1s=-7.5    #alpha1start
    a1e=7.5     #alpha1end
    X1=5                #resolution level 1 for theta
    X2=3                #resolution level 2 for theta
    Y1=5                #resolution level 1 for alpha
    Y2=3                #resolution level 1 for alpha
    dx1=(t1e-t1s)/X1
    dy1=(a1e-a1s)/Y1
    ERR=np.ones([X1*X2,Y1*Y2])*100000 # energy map saved here
    'Level1 (first resolution)'
    tree=KDTree(dataPoint)
    theta= scipy.linspace(t1s+(dx1/2),t1e-(dx1/2),X1)
    alpha= scipy.linspace(a1s+(dy1/2),a1e-(dy1/2),Y1)
    error=np.zeros([X1,Y1])
    mWeight = model.getWeight(1) #set weight = 1
    for i in xrange(len(theta)):
        M1=generateTransfromMatrix(theta[i],axis='x')
        for j in xrange(len(alpha)):
            M2=generateTransfromMatrix(alpha[j],axis='y')
            model.setTransform(M1)
            model.updateTransform(M2)
            dist,_=tree.query(model.applyTransform())
            if (weighted):
                dist = dist * mWeight
            error[i,j]=sum(dist*dist)
    idx=error.argmin()
    r,c=np.unravel_index(idx, error.shape)
    'Level2'
    t2s=t1s + r*dx1
    t2e=t2s+dx1
    a2s=a1s+c*dy1
    a2e=a2s+dy1
    dx2=dx1/X2
    dy2=dy1/Y2
    theta= scipy.linspace(t2s+(dx2/2),t2e-(dx2/2),X2)
    alpha= scipy.linspace(a2s+(dy2/2),a2e-(dy2/2),Y2)
    error=np.zeros([X2,Y2])
    for i in xrange(len(theta)):
        M1=generateTransfromMatrix(theta[i],axis='x')
        for j in xrange(len(alpha)):
            M2=generateTransfromMatrix(alpha[j],axis='y')
            model.setTransform(M1)
            model.updateTransform(M2)
            dist,_=tree.query(model.applyTransform())
            if (weighted):
                dist = dist * mWeight
            error[i,j]=sum(dist*dist)
    idx=error.argmin()
    r2,c2=np.unravel_index(idx, error.shape)
    ERR[r*X2:(r+1)*X2,c*Y2:(c+1)*Y2]=error
    return ERR
```

Optimal Path Using Dynamic Programming

```
def getBestPathDP(CCMap):
    'Get the optimal path based on Correlation Energy Map'
    'CCMap : Correlation Energy Map'
    CCMap=np.array(CCMap)
    'RUN DynamicProgramming to select the best path'
    Pt,D= shortestPath3D(CCMap)
    id = np.argmin(D)
    i,j= np.unravel_index(id, D.shape)
    path=Pt[:,i,j]
    theta=np.array([x[0] for x in path])
    alpha=np.array([x[1] for x in path])
    theta=theta*6-42 #Convert them into the original degree range
    alpha=alpha*1-7
    return theta,alpha


def shortestPath3D(Data, step=2):
    '''
    Data: CCMap data for all frames in the sequence
    It can choose neighbour with flexible step (defined by user)
    '''
    a,b,c=Data.shape
    pad=step*2
    D=np.ones((a,b+pad,c+pad))*100000000
    P=np.zeros((a,b,c),dtype=object) #Path Change
    for j in xrange(b):
        for k in xrange(c):
            P[a-1,j,k]=(0,0)
    D[:,step:b+step,step:c+step]=Data

    span=step*2+1
    TMP=np.zeros([span,span]) #Template
    for i in xrange(1,a):
        for j in xrange(step,b+step):
            for k in xrange(step,c+step):
                TMP=D[i-1,j-step:j+step+1,k-step:k+step+1]
                p=np.argmin(TMP)
                P[i-1,j-step,k-step]=(p/span-step,p%span-step)
                D[i,j,k]+=TMP.min()
    Pt=np.zeros((a,b,c),dtype=object) #Path
    for k in xrange(c):
        for j in xrange(b):
            p,q=j,k
            for i in xrange(a-1,-1,-1):
                p=p+P[i,p,q][0]
                q=q+P[i,p,q][1]
                Pt[i,j,k]=(p,q)
    print D[a-1,step:b+step,step:c+step].min()
    return Pt,D[a-1,step:b+step,step:c+step]
```

Appendix C

Marionette Source Code

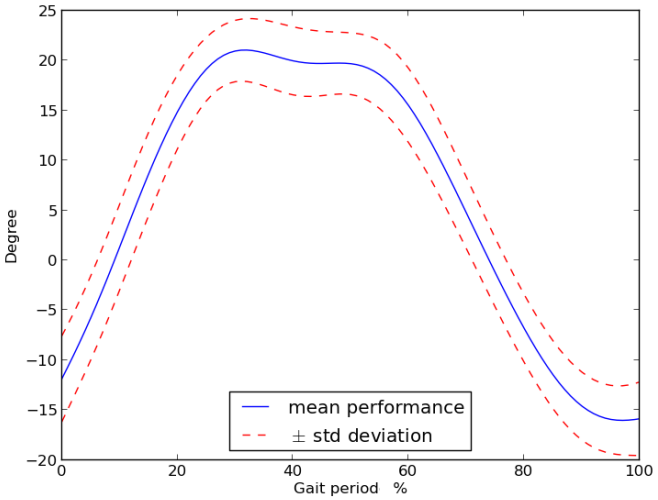
Attractor Forces and Hierarchical Clustering Method

```
def attractor_force(A, Mr, dummy = False):
    'A : Voxel points to behave like an attractor'
    'Mr : Marionette model'
    'dummy : Dummy nodes to refine the attractor_force'
    S = Mr.getModelSegments()
    N = Mr.getModelNodes()
    AS = body_segment_clustering(A, S)
    AN = node_clustering(AS, N, dummy)
    F=np.zeros([11,3])  #11 nodes in 3D
    F[0] = AN[0][0].mean(axis=0)
    F[1] = AN[1][0].mean(axis=0)
    F[2] = AN[1][1].mean(axis=0)
    F[3] = AN[1][2].mean(axis=0)
    F[4] = AN[2][0].mean(axis=0)
    F[5] = AN[1][3].mean(axis=0)
    F[6] = AN[3][0].mean(axis=0)
    F[7] = AN[2][1].mean(axis=0)
    F[8] = AN[3][1].mean(axis=0)
    F[9] = AN[2][2].mean(axis=0)
    F[10]= AN[3][2].mean(axis=0)
    return F - Mr.M     #the attractor force
```

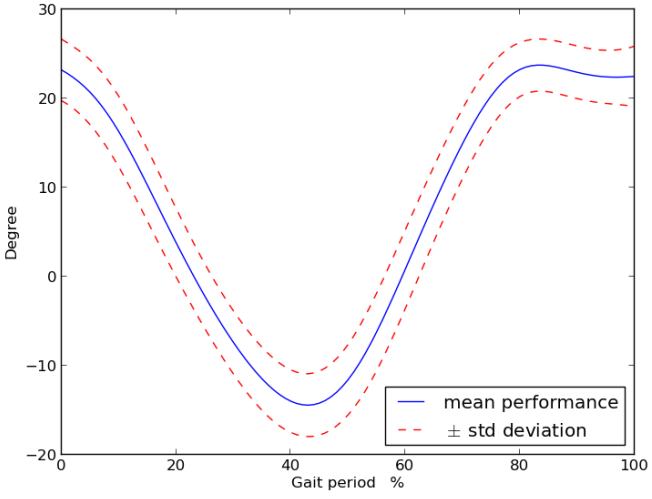
```
def body_segment_clustering(A,M):
    'Hierarchical clustering, stage 1. Separate voxel based on body parts'
    'A voxel points from data'
    'M body-part model'
    'Output AS: a set of voxel points for each body-part model'
    D=[]
    for m in M:
        if len(m)== 1:
            D.append(fc.eucl_dist_array(m[0], A))   #point (head model)
        else:
            D.append(fc.plinedist4(A, m[0], m[1]))  #line (other body parts model)
    D=np.array(D)
    idx = D.argmin(axis=0)
    AS = []
    for i in xrange (len(M)):
        AS.append(A[idx==i]) #this is all points that close to  model M[i]
    return [AS[0],AS[1],np.append(AS[2], AS[3], axis=0),np.append(AS[4], AS[5], axis=0)]

def node_clustering(AS,MN,dummy=False):
    'Hierarchical clustering, stage 2. Separate voxel based on body joints (Nodes)'
    'AS voxel points from cluster1 output'
    'MN node-model'
    AN=[]

    if (dummy): #IMPLEMENT The Dummy Nodes!
        Dm = get_dummy(MN)
        for A,M,d in zip(AS,MN,Dm):
            D = segment_attractor(A, M, d)
            AN.append(D)
    else:
        for A,M in zip(AS,MN):
            D = segment_attractor(A, M)
            AN.append(D)

    return AN

def get_dummy(MN):
    d0  = None
    d1  = (MN[1][1]+MN[1][3])/2.0
    d21 = (MN[2][0]+MN[2][1])/2.0
    d22 = (MN[2][1]+MN[2][2])/2.0
    d31 = (MN[3][0]+MN[3][1])/2.0
    d32 = (MN[3][1]+MN[3][2])/2.0
    Dm = [d0,[d1],[d21,d22],[d31,d32]]#dummy nodes
    return Dm
```

Appendix D

Extracted Kinematic angles

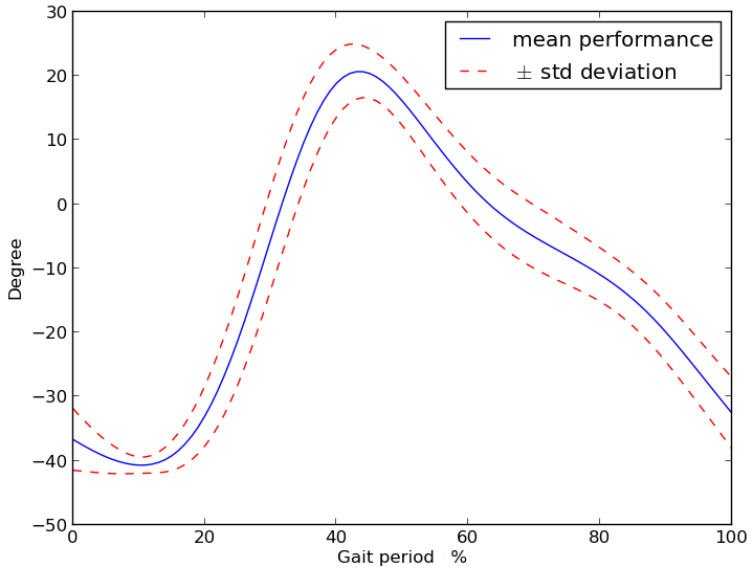


(a) Right thigh sagittal (θ_{RT}) kinematic angle

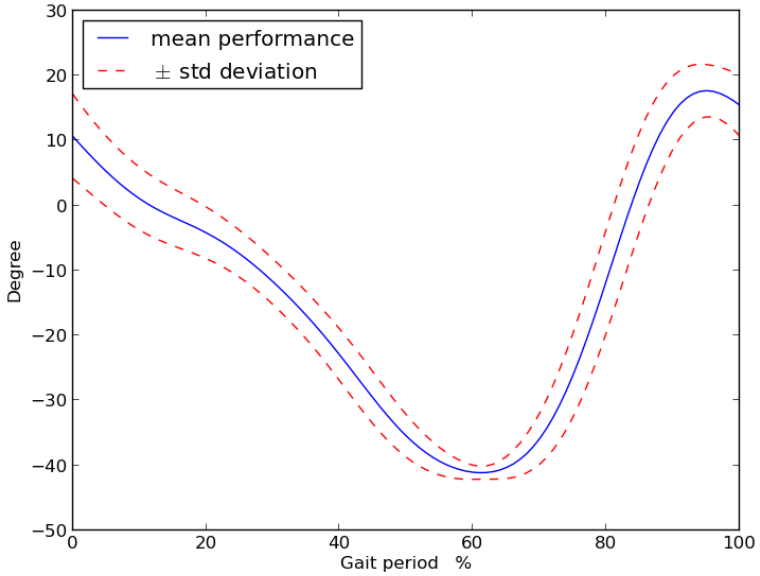


(b) Left thigh sagittal (θ_{LT}) kinematic angle

Figure D.1: The extracted kinematics of thigh sagittal angles

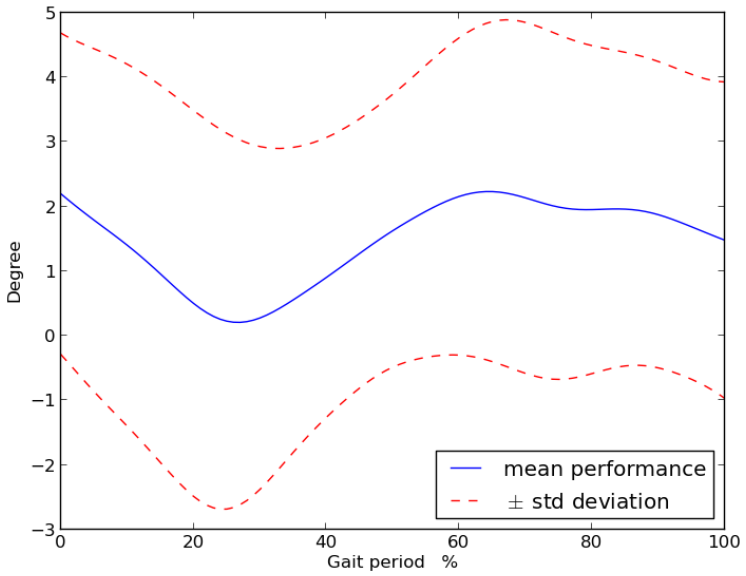


(a) Right shin sagittal (θ_{RS}) kinematic angle

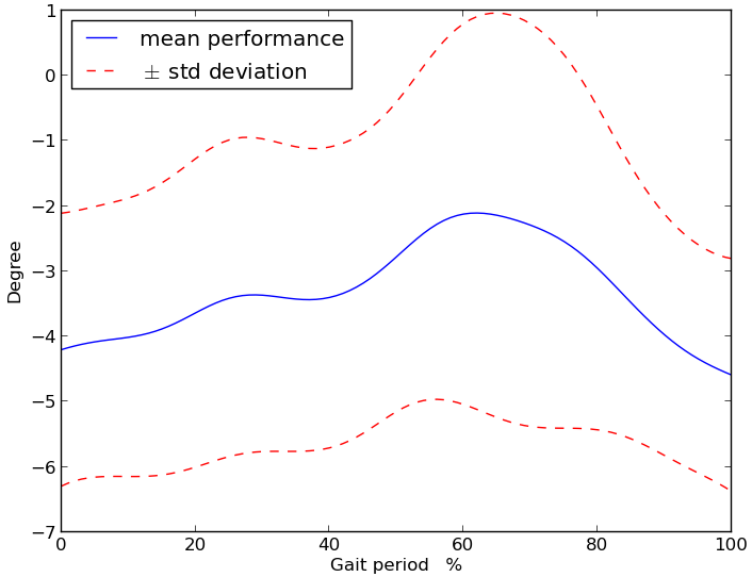


(b) Left shin sagittal (θ_{LS}) kinematic angle

Figure D.2: The extracted kinematics of shin sagittal angles

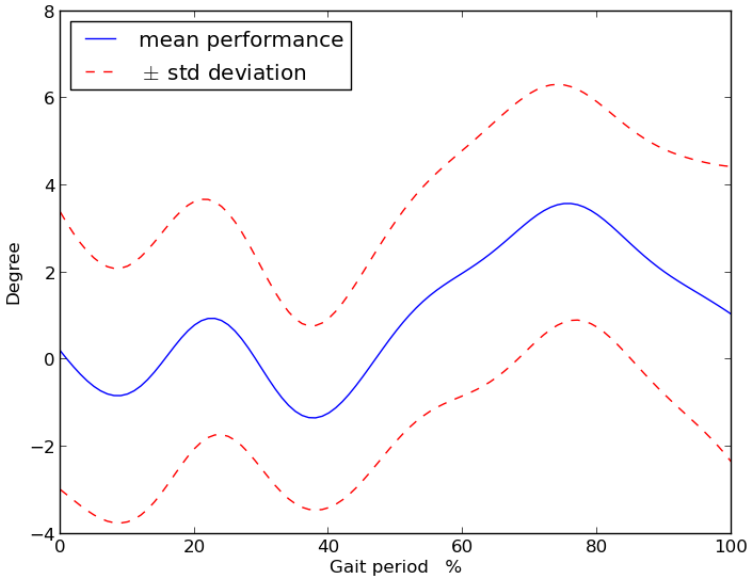


(a) Right thigh frontal (α_{RT}) kinematic angle

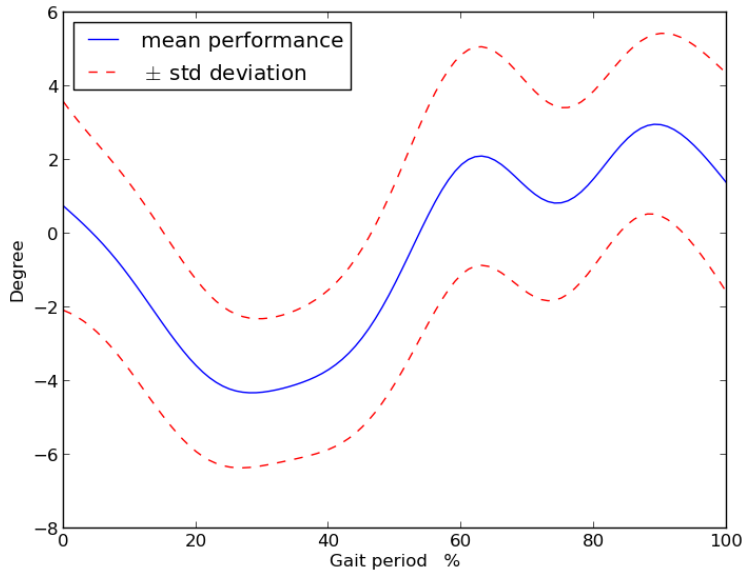


(b) Left thigh frontal (α_{LT}) kinematic angle

Figure D.3: The extracted kinematics of thigh frontal angles



(a) Right shin frontal (α_{RS}) kinematic angle

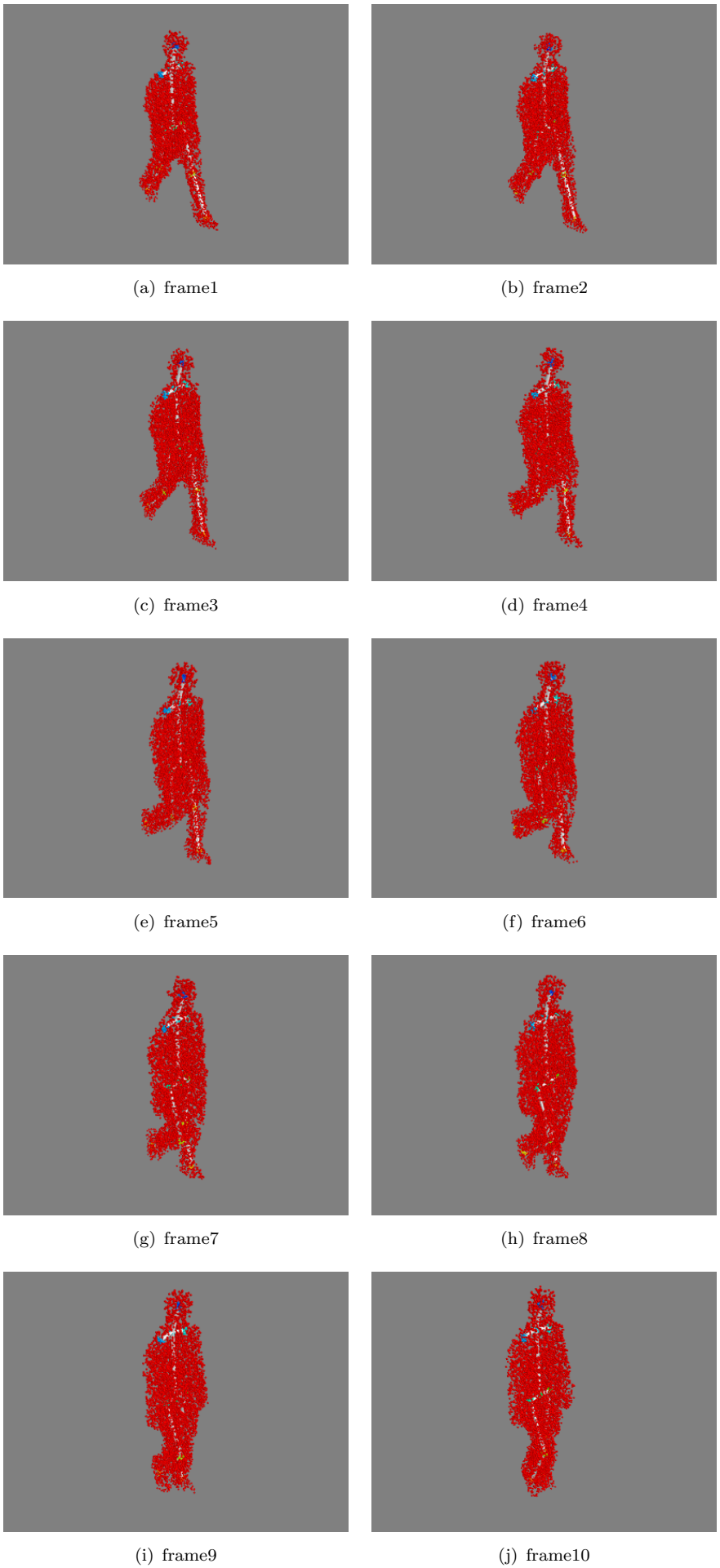


(b) Left shin frontal (α_{LS}) kinematic angle

Figure D.4: The extracted kinematics of shin frontal angles

Appendix E

Marionette Mass-Spring Tracking Results



(a) frame1 (b) frame2

(c) frame3 (d) frame4

(e) frame5 (f) frame6

(g) frame7 (h) frame8

(i) frame9 (j) frame10

Figure E.1: Marionette Model Tracking Results from Frame# 1 to 10

(a) frame11 (b) frame12

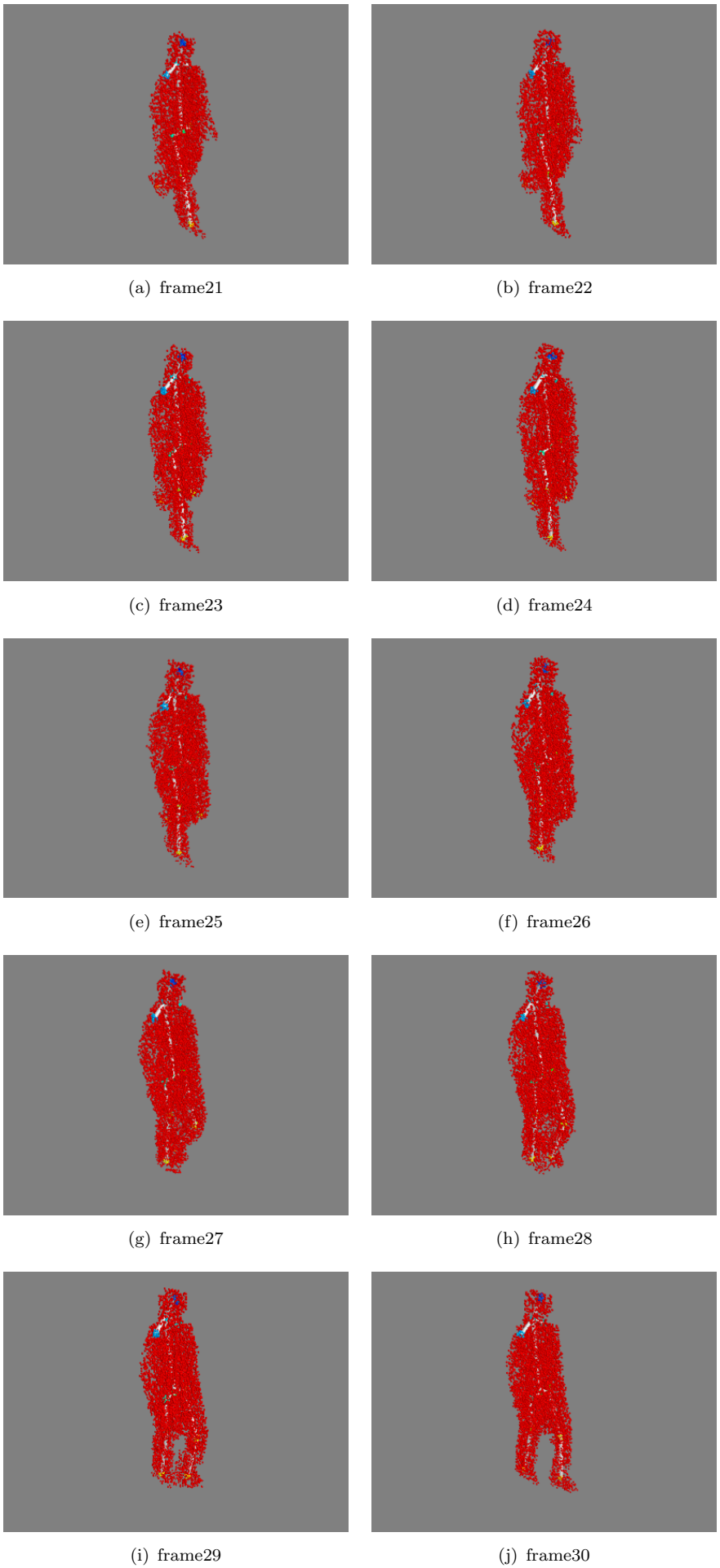
(c) frame13 (d) frame14

(e) frame15 (f) frame16

(g) frame17 (h) frame18

(i) frame19 (j) frame20

Figure E.2: Marionette Model Tracking Results from Frame# 11 to 20



(a) frame21 (b) frame22

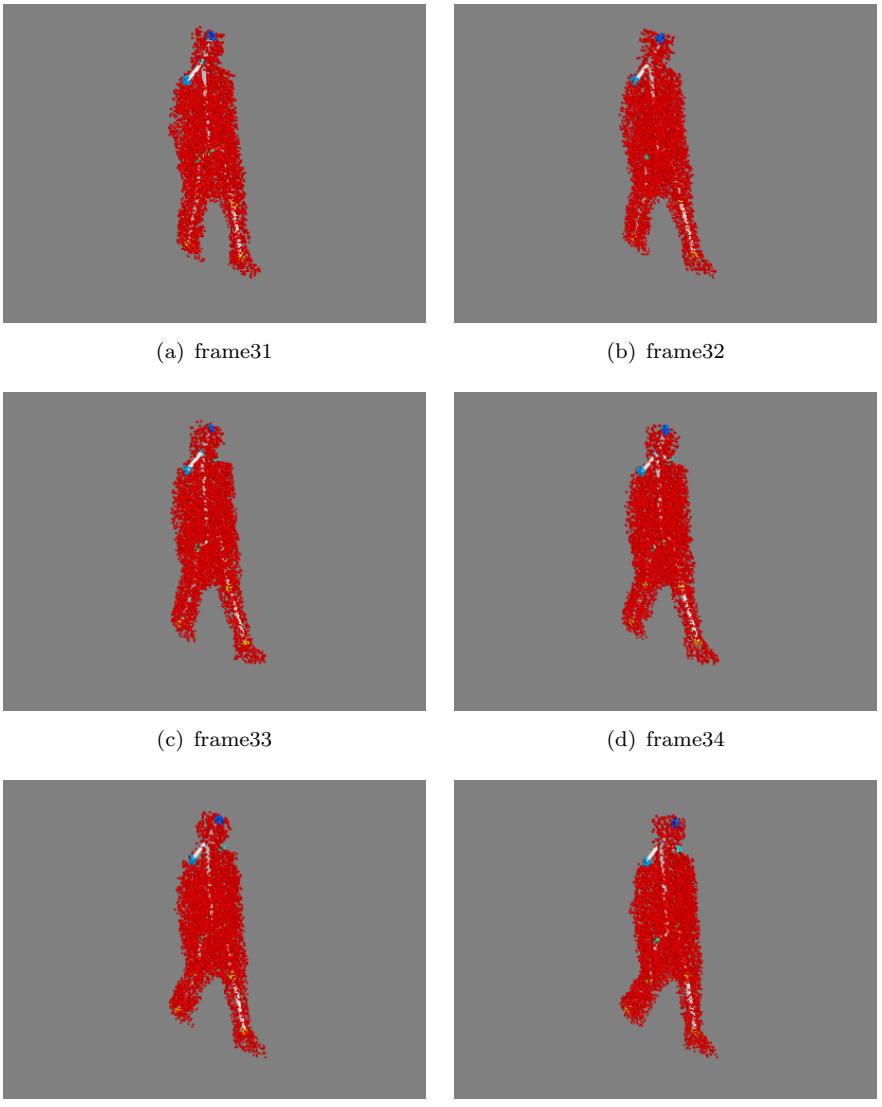
(c) frame23 (d) frame24

(e) frame25 (f) frame26

(g) frame27 (h) frame28

(i) frame29 (j) frame30

Figure E.3: Marionette Model Tracking Results from Frame# 21 to 30



(a) frame31 (b) frame32

(c) frame33 (d) frame34

(e) frame35 (f) frame36

Figure E.4: Marionette Model Tracking Results from Frame# 31 to 36

Bibliography

[1] C. Ben Abdelkader, R. Cutler, and L. Davis. Stride and cadence as a biometric in automatic person identification and verification. In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, page 372, Washington, DC, USA, 2002. IEEE Computer Society.

[2] G. Ariyanto and M. S. Nixon. Model-based 3D gait biometrics. In *International Joint Conference on Biometrics*, pages 1–7, Washington, DC, USA, October 2011.

[3] G. Ariyanto and M. S. Nixon. Marionette mass-spring model for 3D gait biometrics. In *5th IAPR International Conference on Biometrics*, pages 354–359, April 2012.

[4] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. *ACM Transaction on Graphics*, 26(3), July 2007.

[5] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111 – 122, 1981.

[6] D. J. Berndt and J. Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. In *KDD Workshop*, pages 359–370, 1994.

[7] B. Bhanu and J. Han. *Human Recognition at a Distance in Video.* Advances in Pattern Recognition. Springer, 1st edition, 2010.

[8] A. F. Bobick and A. Y. Johnson. Gait recognition using static, activity-specific parameters. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages 423–430, 2001.

[9] R. T. Boston. Techniques for orientation independent gait analysis, October 2008.

[10] I. Bouchrika and M. S. Nixon. Model-based feature extraction for gait analysis and recognition. In *Mirage: Computer Vision / Computer Graphics Collaboration Techniques and Applications*, volume 4418, pages 150–160, March 2007.

[11] N. V. Boulgouris, K. N. Plataniotis, and D. Hatzinakos. Gait recognition using dynamic time warping. In *Multimedia Signal Processing, 2004 IEEE 6th Workshop on*, pages 263–266, September 2004.

[12] G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, (Q2), 1998.

[13] I-M. Chen, R. Tay, S. Xing, and S. H. Yeo. Marionette: From traditional manipulation to robotic manipulation. In *International Symposium on History of Machines and Mechanisms*, pages 119–133. Springer Netherlands, 2004.

[14] S. Corazza, L. Mndermann, E. Gambaretto, G. Ferrigno, and T. P. Andriacchi. Markerless motion capture through visual hull, articulated icp and subject specific model generation. *International Journal of Computer Vision*, 87:156–169, 2010.

[15] D. Cunado, J. M. Nash, M. S. Nixon, and J. N. Carter. Gait extraction and description by evidence-gathering. In *Proceedings of the Second International Conference on Audio- and Video-Based Biometric Person Authentication AVBPA99*, pages 43–48. IAPR, March 1999.

[16] D. Cunado, M. S. Nixon, and J. N. Carter. Using gait as a biometric, via phase-weighted magnitude spectra. In J. Bigun, G. Chollet, and G. Borgefors, editors, *Proceedings of 1st Int. Conf. on Audio- and Video-Based Biometric Person Authentication*, pages 95—102. Springer Verlag, March 1997.

[17] D. Cunado, M. S. Nixon, and J. N. Carter. Automatic extraction and description of human gait models for recognition purposes. *Computer Vision and Image Understanding*, 90(1):1 – 41, 2003.

[18] D. Currell. *Making and Manipulating Marionettes*. The Crowood Press, 2004.

[19] J. W. Davis and S. R. Taylor. Analysis and recognition of walking movements. In *Proceedings. 16th International Conference on Pattern Recognition*, volume 1, pages 315–318, 2002.

[20] Q. Delamarre and O. Faugeras. 3D articulated models and multiview tracking with physical forces. *Computer Vision and Image Understanding*, 81(3):328 – 357, 2001.

[21] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *Int. J. Comput. Vision*, 61(2):185–205, February 2005.

[22] S.L. Dockstader, M.J. Berg, and A.M. Tekalp. Stochastic kinematic modeling and feature extraction for gait analysis. *IEEE Transactions on Image Processing*, 12(8):962–976, August 2003.

[23] H. Freeman and R. Shapira. Determining the minimum-area encasing rectangle for an arbitrary closed curve. *Commun. ACM*, 18(7):409–413, July 1975.

[24] University of Glasgow. Definition and description of the gait cycle. www.gla.ac.uk/ibls/us/fab/tutorial/anatomy/hfgait.html, November 2012.

[25] M. Goffredo, J. N. Carter, and M. S. Nixon. Front-view gait recognition. In *IEEE Second International Conference on Biometrics: Theory, Applications and Systems (BTAS 08)*, October 2008.

[26] M. Goffredo, R. D. Seely, J. N. Carter, and M. S. Nixon. Markerless view independent gait analysis with self-camera calibration. In *IEEE International Conference on Automatic Face and Gesture Recognition 2008*, September 2008.

[27] D. Goldberg. *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley Pub. Co, Reading, Mass, 1989.

[28] R. Gross and J. Shi. The CMU motion of body (MoBo) database. Technical report, 2001.

[29] Y. Guo, G. Xu, and S. Tsuji. Understanding human motion patterns. In *Pattern Recognition, 1994. Vol. 2 - Conference B: Computer Vision Image Processing., Proceedings of the 12th IAPR International. Conference on*, volume 2, pages 325–329, oct 1994.

[30] J. Han. *Data mining : concepts and techniques.* Elsevier/Morgan Kaufmann, Amsterdam Boston, 2012.

[31] R. Haupt. *Practical genetic algorithms.* John Wiley, Hoboken, N.J, 2004.

[32] M. Hofmann, S. Bachmann, and G. Rigoll. 2.5D gait biometrics using the Depth Gradient Histogram Energy Image. In *IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS), 2012*, pages 399–403, September.

[33] M. Hofmann, J. Geiger, S. Bachmann, B. Schuller, and G. Rigoll. The TUM Gait from Audio, Image and Depth (GAID) Database: Multimodal Recognition of Subjects and Traits. *to appear in Journal of Visual Communication and Image Representation, Special Issue on Visual Understanding and Applications with RGB-D Cameras*, 2013.

[34] M. Hofmann, S. Sural, and G. Rigoll. Gait recognition in the presence of occlusion : a new dataset and baseline algorithms. In *Proc. of International Conference on Computer Graphics, Visualization and Computer Vision, Plzen, Czech Republic*, pages 99–104, February 2011.

[35] D. Hogg. Model-based vision: A program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

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

[37] P. L. Jason and E. S. Daniel. Realtime markerless motion tracking using linked kinematic chains. In *JCIS*, pages 849–854, 2002.

[38] G. Johansson. Visual perception of biological motion and a model for its analysis. *Attention, Perception, and Psychophysics*, 14:201–211, 1973.

[39] A. Kale, A. N. Rajagopalan, N. Cuntoor, and V. Kruger. Gait-based recognition of humans using continuous HMMs. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 336 –341, May 2002.

[40] I. A. Karaulova, P. M. Hall, and A. D. Marshall. A hierarchical model of dynamics for tracking people with a single video camera. In *British Machine Vision Conference*, pages 352–361, 2000.

[41] I. A. Karaulova, P. M. Hall, and A. D. Marshall. Tracking people in three dimensions using a hierarchical model of dynamics. *Image and Vision Computing*, 20(910):691 – 700, 2002.

[42] L. Kozlowski and J. Cutting. Recognizing the sex of a walker from a dynamic point-light display. *Attention, Perception, and Psychophysics*, 21:575–580, 1977.

[43] T. Krzeszowski, B. Kwolek, A. Michalczuk, A. witoski, and H. Josiski. View independent human gait recognition using markerless 3D human motion capture. In Leonard Bolc, Ryszard Tadeusiewicz, LeszekJ. Chmielewski, and Konrad Wojciechowski, editors, *Computer Vision and Graphics*, volume 7594 of *Lecture Notes in Computer Science*, pages 491–500. Springer Berlin Heidelberg, 2012.

[44] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis Machine Intelligent*, 16(2):150–162, February 1994.

[45] H-J. Lee and Z. Chen. Determination of 3D human body postures from a single view. *Computer Vision, Graphics, and Image Processing*, 30(2):148 – 168, 1985.

[46] L. Lee. Gait analysis for classification. Technical report, Massachusetts Institute of Technology, June 2003.

[47] L. Lee and W. E. L. Grimson. Gait analysis for recognition and classification. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 148 –155, May 2002.

[48] J. Little and J. E. Boyd. Recognizing people by their gait: The shape of motion. *Videre*, 1:1–32, 1998.

[49] Y. Makihara, H. Mannami, A. Tsuji, Md.A. Hossain, K. Sugiura, A. Mori, and Y. Yagi. The OU-ISIR gait database comprising the treadmill dataset. *IPSJ Transactions on Computer Vision and Applications*, 4:53–62, 2012.

[50] J. R. Matey, O. Naroditsky, K. Hanna, R. Kolczynski, D. J. LoIacono, S. Mangru, M. Tinker, T. M. Zappia, and W. Y. Zhao. Iris on the move: Acquisition of images for iris recognition in less constrained environments. *Proceedings of the IEEE*, 94(11):1936 –1947, November 2006.

[51] G. Mather and L. Murdoch. Gender discrimination in biological motion displays based on dynamic cues. *Proceedings: Biological Sciences*, 258(1353):pp. 273–279, 1994.

[52] D. Matovski, M. S. Nixon, S. Mahmoodi, and J. N. Carter. The effect of time on the performance of gait biometrics. In *IEEE Fourth Conference on Biometrics: Theory, Applications and Systems*, September 2010.

[53] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 369–374, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[54] Z. Michalewicz. *Genetic algorithms + data structures = evolution programs.* Springer-Verlag, Berlin New York, 1996.

[55] L. Middleton, D. K. Wagg, A. I. Bazin, J. N. Carter, and M. S. Nixon. A smart environment for biometric capture. In *IEEE Conference on Automation Science and Engineering*, 2006.

[56] Th.B. Moeslund, A. Hilton, V. Kruger, and L. Sigal. *Visual Analysis of Humans: Looking at People.* Springer, 2011.

[57] L. Mundermann, S. Corazza, and T.P. Andriacchi. Accurately measuring human movement using articulated icp with soft-joint constraints and a repository of articulated models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2007.

[58] M. P. Murray. Gait as a total pattern of movement. *Am. J. Physical Medicine*, 46(1):290–329, 1967.

[59] M. P. Murray, A. B. Drought, and R. C. Kory. Walking patterns of normal men. *J Bone Joint Surg Am*, 46(2):335–360, March 1964.

[60] F. Murtagh. A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, 26(4):354–359, 1983.

[61] H. Ning, T. Tan, L. Wang, and W. Hu. People tracking based on motion model and motion constraints with automatic initialization. *Pattern Recognition*, 37(7):1423 – 1440, 2004.

[62] H. Ning, L. Wang, W. Hu, and T. Tan. Articulated model based people tracking using motion models. In *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*, pages 383–388, 2002.

[63] M. S. Nixon, T. N. Tan, and R. Chellappa. *Human Identification Based on Gait.* Springer, 2006.

[64] S.A. Niyogi and E.H. Adelson. Analyzing and recognizing walking figures in XYT. pages 469–474, 1994.

[65] M. Okumura, H. Iwama, Y. Makihara, and Y. Yagi. Performance evaluation of vision-based gait recognition using a very large-scale gait database. In *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, pages 1–6, September 2010.

[66] A. Oppenheim. *Discrete-time signal processing.* Pearson, Upper Saddle River, 2010.

[67] J. O'Rourke. Finding minimal enclosing boxes. *International Journal of Computer and Information Sciences*, 14:183–199, 1985.

[68] C. Orrite-Urunuela, J. M. del Rincon, J. E. Herrero-Jaraba, and G. Rogez. 2D silhouette and 3D skeletal models for human detection and tracking. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 4, pages 244–247, August 2004.

[69] N. Otsu. A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9:62–66, March 1979. minimize inter class variance.

[70] J. Perry. *Gait analysis : normal and pathological function.* SLACK, Thorofare, NJ, 1992.

[71] P. J. Phillips, S. Sarkar, I. Robledo, P. Grother, and K. Bowyer. The gait identification challenge problem: data sets and baseline algorithm. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 1, pages 385–388, 2002.

[72] S. Sarkar, P. J. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. W. Bowyer. The HumanID gait challenge problem: data sets, performance, and analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(2):162–177, February 2005.

[73] R. D. Seely. *On a Three-Dimensional Gait Recognition System.* PhD thesis, Electronics and Computer Science, University of Southampton, July 2010.

[74] R. D. Seely, J. N. Carter, and M. S. Nixon. Spatio-temporal 3D gait recognition. In *3D Video - Analysis, Display and Applications.* The Royal Academy of Engineering, February 2008.

[75] R. D. Seely, M. Goffredo, J. N. Carter, and M. S. Nixon. View invariant gait recognition. *Handbook of Remote Biometrics: for Surveillance and Security*, April 2009.

[76] R. D. Seely, S. Samangooei, L. Middleton, J. N. Carter, and M. S. Nixon. The university of southampton multi-biometric tunnel and introducing a novel 3D gait dataset. In *Biometrics: Theory, Applications and Systems*. IEEE, September 2008.

[77] G. Shakhnarovich, L. Lee, and T. Darrell. Integrated face and gait recognition from multiple views. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:439, 2001.

[78] S. Shen, M. Tong, H. Deng, Y. Liu, X. Wu, K. Wakabayashi, and H. Koike. Model based human motion tracking using probability evolutionary algorithm. *Pattern Recognition Letters*, 29(13):1877–1886, 2008.

[79] J. Shutler, M. Grant, M. S. Nixon, and J. N. Carter. On a large sequence-based human gait database. In *Fourth International Conference on Recent Advances in Soft Computing*, pages 66–72, November 2002.

[80] N. Spencer and J.N. Carter. Towards pose invariant gait reconstruction. *IEEE International Conference On Image Processing*, 3:261–264, September 2005.

[81] R. Tanawongsuwan and A. Bobick. Gait recognition from time-normalized joint-angle trajectories in the walking plane. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages 726–731, 2001.

[82] Financial Fraud Action UK. Fraud the facts 2012. www.financialfraudaction.org.uk, 2012.

[83] S. Wachter and H.-H. Nagel. Tracking persons in monocular image sequences. *Computer Vision and Image Understanding*, 74(3):174–192, 1999.

[84] D. K. Wagg and M. S. Nixon. On automated model-based extraction and analysis of gait. In *Proceedings of the Sixth IEEE international conference on Automatic face and gesture recognition*, FGR' 04, pages 11–16, Washington, DC, USA, 2004. IEEE Computer Society.

[85] L. Wang, H. Ning, T. Tan, and W. Hu. Fusion of static and dynamic body biometrics for gait recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(2):149 – 158, February 2004.

[86] L. Wang, T. Tan, W. Hu, and H. Ning. Automatic gait recognition based on statistical shape analysis. *IEEE Transactions on Image Processing*, 12(9):1120–1131, September 2003.

[87] D.A. Winter. *The Biomechanics and Motor Control of Human Movement.* John Wiley and Sons, 2nd edition, 1990.

[88] C. Y. Yam and M. S. Nixon. Model-based gait recognition. In *Enclycopedia of Biometrics*, pages 633–639. Springer, 2009.

[89] C. Y. Yam, M. S. Nixon, and J. N. Carter. Extended model-based automatic gait recognition of walking and running. In *Proceedings of 3rd International Conference on Audio- and Video-Based Biometric Person Authentication, AVBPA 2001*, pages 278–283, June 2001.

[90] C. Y. Yam, M. S. Nixon, and J. N. Carter. Gait recognition by walking and running: A model-based approach. In *Proceedings Asian Conference on Computer Vision, ACCV 2002*, pages 1–6, January 2002.

[91] C. Y. Yam, M. S. Nixon, and J. N. Carter. Automated person recognition by walking and running via model-based approaches. *Pattern Recognition*, 37(5):1057–1072, 2004.

[92] K. Yamauchi, B. Bhanu, and H. Saito. Recognition of walking humans in 3D: Initial results. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 45–52, June 2009.

[93] K. Yamauchi and Y. Sato. 3D human body measurement by multiple range images. In *18th International Conference on Pattern Recognition*, volume 4, pages 833–836, 2006.

[94] J. H. Yoo, M. S. Nixon, and C. J. Harris. Model-driven statistical analysis of human gait motion. In *International Conference on Image Processing*, volume 1, pages 285–288, September 2002.

[95] S. Yu, D. Tan, and T. Tan. A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition. In *18th International Conference on Pattern Recognition*, volume 4, pages 441–444, August 2006.

[96] Z. Zhang, H.S. Seah, C.K. Quah, A. Ong, and K. Jabbar. A multiple camera system with real-time volume reconstruction for articulated skeleton pose tracking. In *Proceedings of the 17th international conference on Advances in multimedia modeling - Volume Part I*, MMM'11, pages 182–192, Berlin, Heidelberg, 2011. Springer-Verlag.

[97] G. Zhao, G. Liu, H. Li, and M. Pietikainen. 3D gait recognition using multiple cameras. In *7th International Conference on Automatic Face and Gesture Recognition*, pages 529–534, April 2006.