3-2008

# Statistical Approach to the Characterization and Recognition of Human Gaits

Derrick M. Chelliah

**STATISTICAL APPROACH TO THE CHARACTERIZATION AND RECOGNITION OF HUMAN GAITS**

THESIS

Derrick M. Chelliah, Captain, USAF

AFIT/GOR/ENC/08-02

**DEPARTMENT OF THE AIR FORCE**
**AIR UNIVERSITY**

# AIR FORCE INSTITUTE OF TECHNOLOGY

**Wright-Patterson Air Force Base, Ohio**

AFIT/GOR/ENC/08-02

**STATISTICAL APPROACH TO THE CHARCTERIZATION AND RECOGNITION OF HUMAN GAITS**

THESIS

Presented to the Faculty of the

Department of Mathematics and Statistics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science

Derrick M. Chelliah, B.A.

Captain, USAF

March 2008

AFIT/GOR/ENC/08-02

**STATISTICAL APPROACH TO THE CHARACTERIZATION AND RECOGNITION OF HUMAN GAITS**

Derrick M. Chelliah, B.A.

Captain, USAF

Approved:

| | |
|---|---|
| Samuel A. Wright, Maj., USAF<br>Thesis Advisor | Date |
| Dr. John O. Miller<br>Committee Member | Date |

## Acknowledgments

I would like to thank my family for their support and encouragement during the relatively long process of writing this thesis. I would also like to thank my classmates for their invaluable assistance in procuring all the resources and information sources I needed. Without both of their help it would have been difficult if not impossible to have completed this thesis.

I would also like to thank Maj. Sam Wright for being my thesis advisor and my mentor throughout my graduate studies at AFIT. I have greatly benefitted from your advice and tutelage. I would also like to extend my thanks to Dr. John O. Miller for taking the time to be on my thesis committee. Your mentorship and willingness to help me were invaluable to this endeavor. I have benefitted greatly from the advice and help of all of my instructors at AFIT.

Derrick M. Chelliah

## Table of Contents

## List of Figures

## List of Tables

AFIT/GOR/ENC/08-02

## Abstract

This thesis addresses the final portion of a complete process for human gait recognition. The thesis takes as input information that has been generated from videotaping walking individuals and converting their gaits into numerical data that measures the locations of various points on the body through time. Beginning with this data, this thesis uses a variety of mathematical and statistical methods to create identifying signatures for each individual and identify them on the basis of that signature. The end goal is to achieve under controlled laboratory conditions human gait recognition, an identification method which does not require contact or cooperation with the individual and which can be done unobserved from a distance. Various mathematical models such as the construction of classifiers utilizing Minimum Euclidean Distance, Minimum Mahalanobis Distance and Quadratic Discriminant Functions are employed on both static and dynamic characteristics in order to fully analyze gait data for the purposes of identification.

This thesis starts with previously generated numerical data from a videotaped sequence of images of a subject walking across a room that contains the positions through time of a wide variety of different markers on the individual's body. A MatLab program is initially written to convert the data into a usable format. A variety of mathematical techniques are then employed to generate several classifiers of an individual from a small set of gaits that can be used to identify their gait in any data set.

# STATISTICAL APPROACH TO THE CHARACTERIZATION AND RECOGNITION OF HUMAN GAITS

## I. Introduction

### Background

Human gait recognition is a promising new biometric with the potential for practical application in a wide variety of areas. Biometrics refers to the automatic recognition of people based on their distinctive anatomical and behavioral characteristics (Zhang, 2007: 321). Whereas biometrics encompasses techniques that can focus on a number of physical feature such as fingerprint recognition, facial recognition, palm print identification, speaker identification, iris recognition or signature verification to name a few, gait recognition merits special attention as it offers several advantages not found in other common identifiers.

The principal advantages of gait recognition as opposed to other common biometrics are that it is relatively unobtrusive to perform and difficult to obscure (Bobick, 2001: 301). Measuring an individual's palmprint generally requires the knowledge and often the cooperation of an individual being identified. Iris recognition suffers from similar drawbacks and must usually be performed at close range. Facial recognition requires close range, favorable observation conditions and can be obscured by a hood or various inexpensive means of disguise. Speaker and signature verification require the individual to engage in very specific behaviors and are not well-adapted to quickly surveying large crowds.

An effective human gait recognition technique, by contrast, would suffer from none of these limitations and greatly expand the scope of current identification techniques. An individual's gait could be effectively observed and analyzed over various distances without loss of information. Additionally, the subject being identified would require no knowledge of the process and would not have to be cooperative for the analysis to be effective. As opposed to facial recognition gait recognition is difficult to obscure, even the addition of different garments, prosthetics or other conventional cosmetic changes would not necessarily alter the natural rhythm, motion, and speed of an individual's habitual gait. Gait recognition also offers the possibility of low-resolution and night vision capabilities (Lu, 2006:249). These distinct advantages of human gait recognition offer the potential for a high degree of practical utility for the Department of Defense, Department of Homeland Security and the Air Force.

Underlying the potential for human gait recognition is the notion that an individual's gait is unique and that distinct gaits can be practically identified from a biomechanics point of view. If human gait recognition can be developed to the point where a videotaped individual could have his/her identifying characteristics automatically read from a series of images and then reliably checked against an existing database or known signature of interest, the possibilities for gait recognition applications would be extensive and varied. Currently, recording surveillance video, for instance, is a common and effective means of security in a number of public and private installations. Human gait recognition algorithms could be designed into a software program that automatically analyzes individuals in these images for identification purposes or other information of

interest. Gait recognition could be usefully employed in monitoring applications in security-sensitive environments such as banks, parking lots and airports (Lu, 2006:249). Whereas such video, which often extends over a number of days, is currently scrutinized by trained personnel, human observation can be severely limited by the quantity of information as well as practical limits on attention span and the ability to spot detail. Automatic human gait recognition offers the realistic possibility of using software to scan larger quantities of data in a wider variety of settings while yielding more valuable identification information than is currently possible with conventional identification methods.

## Problem Statement

The concept that each person possess a unique gait suggests that people can be distinguished by their manner of walking. The overall objective of this project is to develop mathematical and statistical methods to classify several individuals' gaits as part of an overarching technique that will eventually allow for the identification of more general populations by their gait signatures.

## Research Objective

There are many different steps in the identification of an individual by their gait. This thesis commences at the point in the process after the initial videotaping has been performed and the numerical information regarding positions of limbs and joints has been recorded. This initial numerical data is taken and analyzed using a variety of

mathematical and statistical methods to attempt to classify each individual uniquely. Several classifiers are built for each individual based on a small number of known gait samples and then it is experimentally determined whether these classifiers accurately identify other samples of the individual's gait. The three classifiers explored on this thesis are based on the concept of matching vectors according to Minimum Euclidean Distance, Minimum Mahalanobis Distance, or through use of Quadratic Discriminant Functions. Additionally, the three algorithms are also used on a variety of data sets including ones that only contain characteristics of a gait expressible in single numbers, termed static characteristics, gait characteristic expressible as curves over time, termed dynamic characteristics, and gait characteristics consisting of both static and dynamic characteristics. Utilizing the algorithms on these distinct databases yields insights into both the operation and effectiveness of the algorithms under a number of common conditions. These methods are then compared and contrasted with each other in order to synthesize the best method for identification. This thesis finally generates a best mathematical algorithm for identifying people on the basis of their recorded gait.

**Research Question**

The overall question this thesis considers is: Assuming that the gaits of individuals are unique, can a person be identified solely on the basis of his/her gait?

This thesis completes the process of answering this question. The specific question addressed in the course of this thesis research is: What is the best mathematical

and statistical method of identifying people by gait when the available input is the position of body markers at successive moments in time?

## Thesis Organization

Chapter II gives an overview of the extensive body of research on gait recognition. The concept of human gait recognition is defined and a brief history of research in the field is provided. The chapter continues to give summaries of a number of relevant papers regarding potential means of using mathematical and statistical models to analyze gait data and the various successes and failures that have been achieved in these endeavors.

Chapter III details the methodology used within this thesis project. The concepts of the Minimum Euclidean Distance, Minimum Mahalanobis Distance, and Quadratic Discriminant Function classifiers are explained and appropriate relevance to the current problem is shown. The specifics of the construction of specific programs that apply each of the above concepts to the current problem are given. An overview of the types of databases to be used in experimentation such as databases consisting of gait characteristics expressible as single numbers, or static characteristics, databases consisting of gait characteristics expressible as curves, or dynamic characteristics, and data bases consisting of both static and dynamic characteristics is given. Assumptions are stated about the nature of the problem and expectations regarding performance are described. Limitations of the research are also included.

Chapter IV discusses the results and analyses of the various methodologies for identifying individuals. The positive and negative aspects of the test runs and algorithm comparisons are investigated. The remainder of the chapter focuses on a detailed summary of the results of running each method of identification on a variety of sample problems.

Finally, Chapter V presents the conclusions and recommendations of the thesis. Comparisons of the various identification methodologies are condensed into an overall recommendation for the optimum method or combination of methods discussed. Advantages and disadvantages of this approach are discussed with respect to previous published efforts in the field. Possible avenues for future research are also identified. Figure 1 describes the general overview of all the stages of the human gait recognition process with the first two steps having already been performed in earlier theses. This thesis completes the process of identifying individuals on the basis of their gaits once numerical data regarding their gaits has been generated.

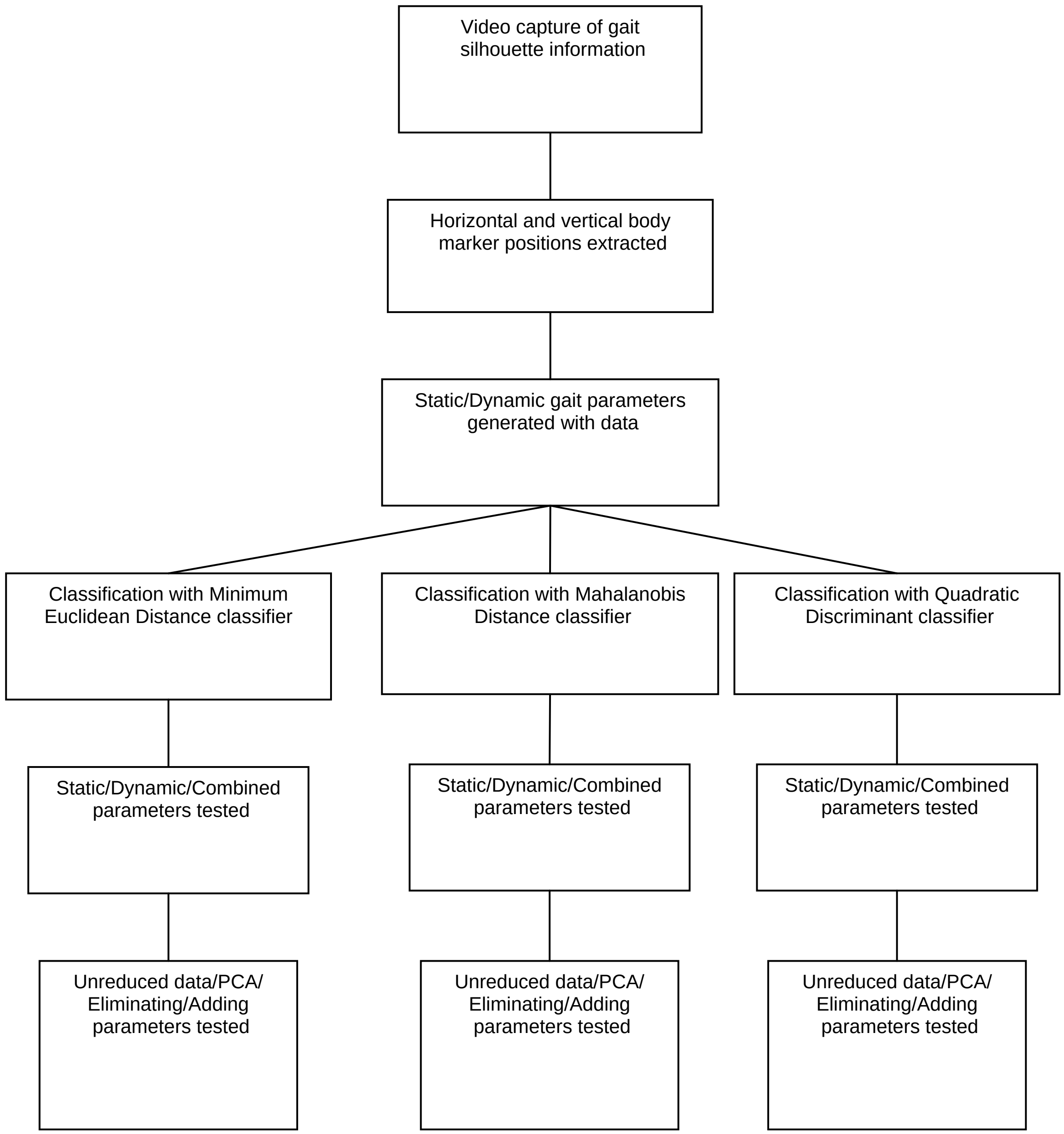


**Figure 1. Human Gait Recognition Process Outline**

# II. Literature Review

## Chapter Overview

This chapter gives a general overview of the history and background of human gait recognition as well as summarizing a number of recent relevant articles published in the field. In the Oxford Dictionary, gait is defined as a '*manner of walking, bearing or carriage as one walks,*' and human gait recognition is an attempt to recognize or describe individuals on the basis of this manner of walking or carriage (Nixon, 2006:1). A relatively unique advantage of using gait as a biometric, as opposed to retina, fingerprint, facial recognition, etc., is that it offers the ability to recognize individuals not only at a distance but also at low resolution and in environments where other biometrics have insufficient information for useful identification (Nixon, 2006: 1). In the case of fingerprinting, facial recognition, or retinal scan, identification requires close proximity to the individual and often conscious cooperation as well (Zhang, 2007:321). Gait recognition by contrast is non-contact and unobtrusive which is extremely useful when attempting to avoid the privacy issues involved in the other common biometrics (Zhang, 2007: 321). As a result of these advantages over other biometrics, human gait recognition is an area of significant interest for the Department of Defense, Department of Homeland Security, and the Air Force among other organizations. Nevertheless, gait recognition is not without its own hindrances and drawbacks. For example a variety of physical conditions such as injuries to joints, drunkenness, and pregnancy can

significantly alter the motion of an individual thereby rendering identification difficult under those circumstances (Hayfron-Acquah, 2002:632).

**Early Research**

The earliest gait recognition studies came from the field of Psychology where Kozlowski and Cutting were able to demonstrate that people could be identified solely on the basis of their gait information (Bobick, 2001: 302). This work was later extended by Stevenage, Nixon, and Vince who explored the degree to which the ability of people to identify others was affected by changes in a variety of environmental conditions (Bobick, 2001: 302).

**Medical Interest**

There has also been medical interest in gait recognition with a number of medical applications being developed. Gait recognition researchers have estimated that individual gait patterns mature by three years of age (Tingley, 2002:150). In terms of medical applications, the primary area of interest has been in the detection and diagnosis of pathological abnormalities in patients on the basis of their gaits. There have been studies of a child's gait to detect birth defects or other abnormalities. Gait studies have been used to classify the components of gait for the treatment of these patients (Nixon, 2006:1).

**Relevant Research**

The two broad classes of gait recognition techniques are model-free gait recognition and model-based gait recognition. Model-free approaches attempt to identify

individuals by only analyzing the shape or motion a subject makes as they walk; the features extracted from this shape and motion are the basis for the identifications (Bobick, 2001:302). Silhouette features, oscillations, or shape-of-motion to accumulate information all lend themselves to a model-free technique.

Model-based techniques, by contrast, match a model to either the person or his/her walk. A body-based model will match a body model to the individual in every frame of the walking sequence whereas in walking models a model of how the person moves in general is created and learned for every person (Bobick, 2001:302). The more commonly used models are volumetric and stick figure models while ribbon and blob models are relatively less used (Nixon, 2006:5). In volumetric models the person is represented with a series of spheres, compression of the spheres into two dimensions yields a ribbon model and a blob model uses amorphous blobs to represent the walking person (Nixon, 2006: 5-6). This thesis will utilize a stick model which merely notes the positions of all joints at recorded points in time and uses lines to form them into the figure of a walking person.

The first automatic gait recognition algorithm to use statistical shape analysis and is presented by Wang, Tan, Hu and Ning. This group extracts a silhouette of the individual from the background of an image sequence using a background subtraction procedure (Wang, 2003: 1120). A complex vector configuration is created from the detection of the temporal changes of the silhouettes. Next portrayed is a method in shape statistics know as Procrustes shape analysis (Wang, 2003: 1123). Instead of analyzing the dynamics of the gait Procrustes shape analysis capture the characteristics of

individual gaits using the walking action of the individual (Wang, 2003: 1120). These methods are successfully used to identify different individual gaits in the silhouettes.

A three-dimensional human body model is challenging because of the large number of free parameters but is also successfully studied by Wang, Tan, Hu and Ning. This study reduces the number of parameters by assuming walking motion is parallel to the image plane (Ning, 2002: 537). First, the posture of the human body model in the next frame is first predicted and then this prediction is matched to the next frame in the sequence. The matched frames are then optimized to find the minimization of the match error (Ning, 2002: 538). Wang, Tan, Hu, and Ning also attempted to find distinguishing characteristics in the individual gait posture vectors (Ning, 2002, 539).

A new method of spatio-temporal symmetry using the Generalised Symmetry Operator was an approach taken by Hayfron-Acquah, Nixon and Carter. The psychological view that human gait is a symmetrical pattern of motion is the inspiration for this approach (Hayfron-Acquah, 2002: 632). An individual is recognized by both body shape and motion using temporal information gained from gait recognition (Hayfron-Acquah, 2002: 632). They start by performing a symmetry extraction on the original image. First, the silhouette is extracted from the original image, the edges found, and the spatial symmetry map is detected by examining the symmetry from pairs of image points (Hayfron-Acquah, 2002: 633). Second, all the symmetry maps from an image sequence are averaged in order to derive a gait signature for recognition (Hayfron-Acquah, 2002: 634). The *k*-nearest neighbor rule is applied to produce recognition by symmetry which is a reasonable classification. A better classifier may be found in a

feature space classification or a more sophisticated classifier than the *k*-nearest neighbor (Hayfron-Acquah, 2002: 634). The results from this method corroborate previous work which indicates that human gait has symmetrical properties and is unique for an individual (Hayfron-Acquah, 2002: 632).

Another approach in gait recognition is Fourier series. Specifically, Yu has analyzed the spatio-temporal characteristic of moving silhouettes. Gait data dimensionality can be reduced and computational cost lessened through use of a set of Key Fourier descriptors (KFDs) (Yu, 2004: 282). The KFDs are derived from the discrete Fourier transform and are invariant to translation, scale, and rotation (Yu, 2004: 283). The leave-one-out cross-validation rule and nearest neighbor classifier are used to classify the data (Yu, 2004: 283). Yu discovered that a human silhouette can be recognized using only sixteen points (Yu, 2004: 285).

Tingley used Fourier series in his study of gait in young children. As opposed to identifying individual children, this study seeks only to classify the child's gait cycle as being the bounds of normal or abnormal gait (Tingley, 2002: 151). This group used eleven functions that involved hip angle, knee angle, and ankle angle to derive the coefficients that describe the Fourier curves (Tingley, 2002: 152). A linear combination using principal component analysis (PCA) is approximated from the variation of the child's gait pattern from that of normal hip angle, knee angle, and ankle angle pattern (Tingley, 2002: 153). This is sufficient to classify a child's gait pattern as either normal or abnormal.

Begg and Kamruzzaman study gait cycle changes by combining three types of machine learning approaches of gait measures; basic spatial/temporal, kinetic, and kinematic into one (Begg, 2005: 401). The gait cycles of twelve young individuals and twelve elderly individuals are compared in this study. As opposed to individual identification, the purpose of this study is only to classify an individual into an age group. This is accomplished through the use of neural networks and fuzzy clustering techniques (Begg, 2005: 402). A support vector machine (SVM) is a machine classifier that assists in the classification and regression of the data (Begg, 2005: 402). This study demonstrates that SVMs can identify the differences between the young and elderly walking gait cycles. SVMs also show the underlying data structure of the models of the young and old (Begg, 2005: 406). This proved sufficient to classify gait signatures into broad categories, though not to identify specific individuals.

Fujiyoshi, Lipton, and Kanade investigate the creation of image skeletonizations through analysis of motion of subjects in video streams. In order to achieve real-time target extraction they begin by using a background subtraction technique that is more adaptable to environmental changes (Fujiyoshi, 2004: 114). Slow dynamic changes in the environment, "once-off" independently moving false alarms, movement of environment clutter, and the moving target are all considered significant types of image motion for target detection (Fujiyoshi, 2004: 114). After the first step only removes the background, the second step is to process the target removing everything in the frame that is not the person and producing the star skeleton formation of the image (Fujiyoshi, 2004:

114). A star skeleton formation is when the extremities of an individual are joined to a center point or "centroid" by lines that form a star pattern.

Lu, Plataniotis and Venetsanopoulos pursue a different tactic with the development of a sophisticated layered deformable model (LDM) for human body pose recovery in gait analysis (Lu, 2006: 249) . The model in this study is designed to closely mimic manually labeled silhouettes of figures in motion. For a gait parallel to the observing camera, the LDM model defines the body part widths and lengths, the position and the joint angles of the human body using 22 parameters (Lu, 2006: 249). The model is designed to have four layers and allow for limb deformation. In practice, the LDM recovery algorithm is first developed for manual silhouettes in order to generate ground truth sequences (Lu: 2006: 250). It can then be extended for automatically extracted silhouettes. On a test of 10,005 frames for 285 gait sequences under a wide variety of environmental conditions, the LDM model was able to achieve an average error rate of 7% for lower limb joint angles of all the frames (Lu, 2006: 250). The LDM model of Lu, Plataniotis and Venetsanopoulos is consequently not only a sophisticated gait recognition program but also a relatively accurate one.

Chen, Chen, Chen, and Lee utilize a different technique with their development of a star skeleton to motivate gait recognition efforts. Specifically, the study uses a Hidden Markov Model (HMM) based methodology for action recognition using a star skeleton as a representative descriptor of human posture (Chen, 2006: 171). The star skeleton itself is created using a fast skeletonization technique that connects the centroid of target objects to contour extremes (Chen, 2006: 172). In the study, the star skeleton is clearly

defined as a five-dimensional vector in star fashion in order to correspond to the head and four limbs which are usually the local extremes of a human shape (Chen, 2006: 174). Once the skeleton has been established, time-sequential images expressing human action are transformed into a feature vector sequence (Chen, 2006: 175). The feature vector sequence is then transformed into a symbol sequence where HMM can model the action (Chen, 2006: 175). Finally, a posture codebook that contains representative star skeletons of each action type is used to match feature vectors against in order to determine the action portrayed. In one particular simulation the algorithm was used to characterize one hundred video clips each depicting a single action (Chen, 2006: 176). The authors were able to achieve a 98% recognition rate in this instance although detection rates dropped under less ideal conditions (Chen, 2006: 176).

In their study, Bobick and Johnson approach the same problem from a significantly different point of view by attempting to identify individuals through the use of static body parameters. That is to say that recognition is not based on leg swing, joint angle or motion but rather the invariant aspects of an individual's anatomy such as height, width, etc. (Bobick, 2001: 301) . They find that it is a generally simple task to discriminate between subjects from a single viewpoint but that discrimination decreases when data across views are considered (Bobick, 2001: 302). Ground truth motion-capture data of a reference subject is utilized to establish scale factors that can transform data from different viewpoints into a common frame of reference (Bobick, 2001: 301). This study by Johnson and Bobick is significant in that it demonstrates the practical

utility of gait recognition using static body parameters which is a relatively little used technique for identification.

Liu and Sarkar employ a different technique in order to study the intricacies of human gait-based recognition. According to Liu, studies of gait are frequently confounded by errors in the extracted silhouettes which are used as a basis for most recognition algorithms (Liu, 2005: 170). The new model based silhouette reconstruction strategy designed to address this issue uses a population based HMM in concert with an eigen-stance model that corrects common errors in silhouette detection that generally result from shadows or background subtraction (Liu, 2005: 172). The combination of these techniques allows for the study of extracted silhouette errors across a large population of subjects. As opposed to pixel-level techniques for cleaning silhouettes, this methods is shown to be effective at not only removing shadows but also carried items (Liu, 2005: 172). However, the overall conclusion Liu draws from this research is that though these new techniques demonstrably improve silhouette quality they fail to improve gait recognition algorithm performance (Liu, 2005: 181). This observation supports the hypothesis that factors other than poor silhouette quality are the primary contributors to poor gait recognition algorithm performance.

Kale focuses his research on a view-based approach to recognize humans on the basis of their gait. The work examines both the width of the outer contour of the binarized silhouette of the walking person and the entire binary silhouette as both are considered to be features of interest (Kale, 2004: 1163). The first method of obtaining the observation vector from the image features is to transform the high-dimensional

image feature into a lower dimensional space by generating a frame to examplar distance (Kale, 2004: 1164). In the second method, a HMM is trained directly on the feature vector thereby avoiding the need to learn high-dimensional probability density functions (Kale, 2004: 1168). The statistical nature of the HMM-based methodology yields significantly increased robustness to both representation and recognition in human gait recognition (Kale, 2004: 1163).

**Summary**

Human gait recognition holds the promise to assist in the solution of many of the security and identifications dilemmas that are currently receiving a great deal of attention in the United States and abroad. The ability of gait recognition to meet that promise is largely dependent on the speed with which it can be performed and the solution quality that can be obtained under various time constraints, viewing angles, and environmental parameters. This performance bottleneck has been approached from two different directions. As computer speeds have increased and computational cost has decreased, the computationally complex approach of gait recognition has become more practical and attractive. Additionally, a substantial amount of research has been generated over the past several years into the best mathematical and statistical techniques that can be applied to improve recognition performance. The most noticeable distinction in these techniques is between model-based and model-free approaches. Model-free approaches tend to focus on silhouette features, oscillations, or shape-of-motion to accumulate information on the gait. Model-based approaches generally use volumetric, ribbon, blob, and stick-figure models to study the structural shape of an individual. The above articles

summarize approaches that have been taken primarily in the area of model-based gait recognition. These techniques serve as an inspiration for the model-based gait recognition techniques developed in the remainder of the thesis. The previous research also provides important benchmarks in order to judge the performance of newly developed techniques. Human gait recognition is noteworthy as a biometric that can be used at a distance without the subject's knowledge and that is difficult to obscure. For these reasons, it remains an area of considerable interest for the Department of Defense, Department of Homeland Security, and the Air Force.

# III.  Methodology

## Chapter Overview

This chapter discusses how the research was conducted.  The overall concept of the project is that an individual gait can be distinguished from other gaits on the basis of data gathered through a video recording of the walking individual.  The methodology of this thesis consequently focuses on a computer program that takes data from video recordings of walking individuals and analyzes data gathered from those videos in order to identify the individuals.  This data is obtained from two sources, firstly from a sample of 45 gaits obtained from the University of Pennsylvania and eventually from data generated with AFIT's own video recording of walking individuals once the appropriate algorithms have been completed.  This chapter first describes the nature of the source data and the means of generating individual gait statistics from it.  It then discusses the various stages of the analysis program, how parameters are chosen to explain the variation in the data and how various methods such as Minimum Euclidean Distance, Minimum Mahalanobis Distance and Quadratic Discriminant Functions compare in identifying the gaits correctly.  Finally, limitations and drawbacks of this process of gait recognition are discussed.

**Data Collection**

The gait recognition program for MatLab was developed using two different sets of data gathered from two separate research facilities. One of the sets of data currently exists from the University and Pennsylvania and is used as test data throughout this thesis. The other set is to be produced at AFIT on an as-needed basis from its own camera/video recording/algorithmic processing human gait recognition facility. Currently, however, this AFIT data exists only as a set of specifications and is mentioned primarily in the context of the eventual utility of this thesis work. Each set of data in its final form contains the same format and type of information which make possible a direct comparison of the performance of the gait recognition algorithm on the two problems. Each set of data consists primarily of a series of horizontal and vertical coordinates for points that locate various key body parts in three-dimensional space. Further, each point is tracked through time in successive video frames. The first line for a frame corresponds to the head motion sensor, the second to the ankle, etc. This ordered point format is identical in both data sets and allows the program to easily construct a model of a walking human through a series of moments using a list of points. While the end product format is the same in both cases, the laboratory circumstances and physical recording apparatus do differ to some degree.

The first set of data was collected from the University of Pennsylvania gait recognition facility by Dr. Lief Finkel. Five different subjects were asked to walk on a treadmill. Each subject was then recorded at nine different speeds from 1 mile per hour

(mph) up to 5 mph in 0.5 mph increments.  Information was recorded once per frame and consisted of the locations of thirty body markers on critical points on the subjects' bodies. These points tracked the left toe, left ankle, left knee, left heel, left index finger, left wrist (b), left elbow, left wrist (a), clavicle, right knee, right heel, right shoulder, right wrist (b), right index finger, right elbow, right wrist (a), right toe, right ankle, left shoulder, sternum, left back waist, right front waist, left front waist, right back waist, C7 bone, T10 bone, right back head, left head, left back head, right head so as to be able to yield as much critical information regarding the individual's gait as possible.  The result is consequently a list containing thirty horizontal and vertical coordinate pairs per frame with 1000-2000 frames per trial and with 45 trials overall.

The second set of data is in the process of being collected at the Air Force Institute of Technology.  Subjects were filmed walking in from a static background consisting only of a background wall and level blacktop ground below the wall.  No other objects such as vegetation or sky were viewable by the video camera.  A tripod was used to minimize vibration in the video camera.  Upon assurance of confidentiality and a comprehensive explanation of the experiment and its purpose, volunteers were acquired to provide human gait subjects.  Each volunteer made three walking passes in front of the camera at distances of five, ten, and fifteen feet from the background building respectively. The order of the passes was randomized for each volunteer and the data was conducted on two separate days, the first with the building partially sunlit and the second with the building completely shaded.

After collecting the data, a MatLab program was used to convert the video files from a detailed, colored version of a person walking in front of a wall to a much simplified video of a white silhouette moving against a black background. This was done using the change in pixel color between different frames in order to differentiate the moving pixels in a scene from the stationary ones and then recoloring both. The resulting silhouette videos were then analyzed using another program that attempts to identify specific body parts, i.e. head, hands, feet, or knees through use of silhouette location, movement speed and other extractable parameters. This allows the data to be converted into a series of horizontal and vertical coordinates representing the positions of critical body parts at various points in time. In other words, this produces data in an identical format to that of the first data set from the University of Pennsylvania. The two data sets can then be directly compared for algorithm performance in the following research stages.

**Data Preparation**

Once the data has been collected, it is then processed from its raw form as a series of horizontal and vertical coordinates into a variety of statistics and measurements for each sample that lend themselves more easily to statistical analysis. An algorithm separates the data into their respective frames and labels the appropriate lines as the corresponding motion sensor and frame. Running through the frames sequentially the algorithm then calculates the parameters that will define the sample and facilitate profiling and identification. Two varieties of parameters are generated to describe the

gait of an individual.  The parameters are divided into static parameters and dynamic parameters which are distinguished from each other in terms of how they represent gait information.

**Static Parameters**

Static parameters are parameters describing a person's gait that can be expressed as a single value.  Examples of static parameters are height, leg length, maximum stride length, or shoulder/waist ratio.  All of these parameters can be expressed as a single number, often they are invariant over time and can consequently be extracted from a single frame of data, and all eliminate complex information regarding the changing aspects of a gait over time.  There are numerous advantages to examining these single value parameters in order to identify a gait.  The most obvious advantage is simplicity of understanding and calculation.  Dynamic parameters include an enormous amount of data in order to capture the behavior of certain gait characteristics over time.  It is often significantly more difficult to work with these larger data structures and far from clear what the data intuitively represents.  Static parameters, by contrast, can be recorded using far less data storage while representing a much simpler model of an individual.  They also offer the significant benefit that they can often be extracted from a single frame making identification possible in the cases where the several seconds of unobstructed view necessary to gait dynamic parameter information are unavailable.

**Dynamic Parameters**

Dynamic parameters are parameters describing a human gait that represent aspects of that gait as they vary over time. Distinguishing them from static parameters is the fact that they are not expressed as a single value but rather as a curve expressed with values extending over an entire gait cycle. Examples of changing dynamic parameters that could be collected over time are right knee angle, foot separation, hand separation, and absolute head height. Dynamic parameters have the significant advantage over static parameters that they can represent much more complex and highly individual information regarding a subject's gait than single value quantities can. Dynamic parameters have the disadvantage they require larger quantities of data to be manipulated, require several seconds of footage at minimum to acquire, and often don't lend themselves to an intuitive understanding of importance.

For the purposes of algorithmic comparison, all dynamic parameters are standardized here to have the same size and represent the same information. All the dynamic parameters here are generated by examining each frame of a gait video and recording the value of interest for that particular frame. These collected values can be thought of as a curve extending from the first to the last frame for a video. This curve is then divided into discrete identical gait cycles for purposes of comparison. The individual curves, which frequently contain widely varying numbers of frames due to inconsistent walking speed, are then all scaled to 100 frames in length through linear extrapolation. Now that all the curves represent identical gait cycles with the identical number of frames, they are all averaged together to produce one 100 element long vector.

This 100 element long vector is then referred to as a dynamic parameter representing the value of interest over an average gait cycle for one gait sample.

**Pre-Processing Methods**

A total of fourteen static parameters and six dynamic parameters are generated for the purpose of testing classifier efficiency. The algorithms are all developed to accept data regarding all gait samples in the form of a single matrix. In the case of the static parameters, this creates a 45 by14 data matrix and in the case of the dynamic parameters it creates a 45 by 600 data matrix. The algorithms working with the combined data set either work with the concatenated 45 by 614 matrix or convert the static parameters into dynamic parameters to create a total 45 by 2000 data matrix depending on the requirements of the application.

Having generated the appropriate input matrix, however, there are a number of methods for pre-processing the data in order to potentially improve the algorithm classification accuracy. All algorithms begin by running on the unaltered or “raw” data matrices in order to have a basis for comparison. There are three other techniques that are then applied in an attempt to improve the classification accuracy.

The first technique applied to the input data matrix is principal component analysis (PCA). Principal component analysis is a projection method for viewing a high-dimensional set of quantitative data in a few dimensions for the purposes of analysis. A covariance matrix is first calculated using the input data matrix and then an eigenvalue decomposition is performed on the covariance matrix in order to produce the orthogonal

vectors referred to as principal components. The data is consequently transformed to a new coordinate system while retaining those aspects of the data that contribute most to the variance. Each successive principal component represents less of the overall variance, therefore sufficient variance can be accounted for by retaining as many principal components as necessary. A loadings matrix is also calculated which indicates to what degree the original parameters are now associated with the new principal components. The overall process allows a matrix of size 45x2000 to be represented by a transformed matrix of size 45x5 and therefore represents a significant savings in calculation time and data storage.

The next type of pre-processing method to be considered is the possibility of parameter elimination. The concept is that some of the gait parameters selected for study might not lend themselves well to differentiation between different gaits and instead of being highly individual and descriptive might represent noise or misleading correlations between, for instance, the speed of the gait rather than the individual generating the gait. In order to deal with these misleading parameters two simple branch and bound methods are constructed. The first branch and bound method focuses on eliminating parameters. The algorithm is initially performed with the complete parameter set. The second step, is a process in which the algorithm is performed while eliminating a single different parameter during each performance. Comparing the results of each of these trials, each missing exactly one parameter, the algorithm selects the parameter whose absence most improves the classification effectiveness of the algorithm. The same process is then repeated iteratively but this time starting with new set of parameters that has had the least

useful one removed. In this way, successive parameters are permanently removed from the complete set until doing so ceases to improve classification accuracy. When no further gains can be achieve through parameter elimination, the algorithm stops and outputs the remaining parameter set and algorithm classification accuracy.

The second branch and bound algorithm functions in an almost identical manner with the sole difference being that it begins with no parameters, tests all the cases where exactly one parameter is added, and then permanently keeps the most useful parameter for classifications purposes. It iteratively repeats this process by running the algorithm on successive data sets that all involve adding a different new parameter but only retaining the one that most improves classification accuracy. This process is repeated for the first, second, third parameter, etc., until adding parameters no longer increases the classification accuracy of the algorithm.

**Algorithm Design**

Having successfully created the necessary data sets and chosen a pre-processing method, the next step in the gait recognition algorithm is to generate a profile and then attempt to classify gait signatures on the basis of that profile. For the purposes of comparison, several different algorithms are used ranging from relatively simple classification schemes to more advanced forms of classification.

The first identification technique used is consequently the simplest. The Minimum Euclidean Distance classifier takes the training data and averages all the parameters for each subject. As there are five subjects each walking at nine speeds for a

total of 45 samples, five separate profiles are produced. Each of the 45 samples to be identified is then compared with each of the five profiles. Five Euclidean distances are calculated per sample where the Euclidean distance is equal to the square root of the sum of each of the parameters of one vector multiplied by the corresponding parameter of the second vector. The following equation demonstrates how Euclidean distance is numerically calculated where *d* is equal to the distance, *x* represents the classifier mean, *n* is the total number of dimensions, and *y* represents the sample to be classified.

$$d = \sqrt{((x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots(x_n - y_n)^2)} \tag{1}$$

This number gives an intuitive sense of the “closeness” of one vector to the other even in dimensions greater than the three where visualization is possible. Consequently, the lowest or Minimum Euclidean Distance indicates the profile the sample matches and the algorithm classifies it as such. A percent accuracy is calculated as simply the number of samples correctly identified divided by the total number of samples. A confusion matrix is also generated in order to demonstrate any possible patterns in any errors in order to give insight into the algorithm’s operation.

The second algorithm used to classify gaits is the Minimum Mahalanobis Distance algorithm. Similar in many ways to the algorithm that uses the Minimum Euclidean Distance of samples to a profile in order to classify gaits, the Minimum Mahalanobis Distance algorithm uses the Minimum Mahalanobis Distance between samples and a profile in order to characterize gait samples. The primary difference between the two measurements is that while the Minimum Euclidean Distance measures the distance between the two vectors, the Minimum Mahalanobis Distance divides that

distance by the standard deviation of the profile. The following equation demonstrates how the Minimum Mahalanobis Distance is numerically calculated where *d* is equal to the distance, *x* represents the classifier mean, *o* represents the variance of the classifier mean and *y* represents the sample to be classified.

$$d = \sqrt{((x_1 - y_1)^2 + (x_2 - y_2)^2 + ...(x_n - y_n)^2)/o} \qquad (2)$$

This has the effect of making close distances weigh more heavily in the calculation if the profile data has less variance. This agrees with the intuitive idea that if a profile is the average of a number of vectors then a sample being close to that average has more importance if the data used to create was more tightly clustered.

The third technique employed for the identification of human gaits based on various statistical parameters of their walk is a Quadratic Discriminant Function. To apply the Quadratic Discriminant Function, it is assumed that all the classes of gaits are Gaussian in nature. Given this assumption, a Quadratic Discriminant Function yields *n*-dimensional quadratic decision boundaries between classes. The function does so through use of the following equation where *d* represents the discriminant score, *cov* is the covariance of the gaits composing the classifier, *y* is the vector to be classified and *x* is the mean of all the gaits in the classifier.

$$d = \log(\det((\mathrm{cov}(y - x))^{-1}))/2 - (y - x)/2 * (\mathrm{cov}(y - x))^{-1} \qquad (3)$$

The Quadratic Discriminant Function is an effective classifier to the extent that the n-dimensional decision boundaries are simple, uncomplicated boundaries easily delineated by the above equation. In this case each class corresponds to the identity of

the person in a video sample. Discriminant functions also share the advantage of not being dimensionally reducing methods.

Having written the three identified gait classifier algorithms, each algorithm is then tested with the relevant parameters. Confusion matrices which detail the accuracy of each algorithm and what kind and the number of misclassifications that occur for each class are generated. Subsequently, the algorithms are then run on different sets of parameters such as only static or dynamic parameters in order to determine how the system would work under a variety of conditions. In each stage, evidence is gathered to determine whether any algorithm is distinctly superior to the others and what the limitations and possibilities of the identification system as a whole may be.

**Practical Implementation**

Once an optimum algorithm/parameter type/pre-processing method is generated using the complete run of trials and the Lachenbruch hold-out procedure a few minor changes are made to the final algorithm in order to adapt it to the practical challenge of human gait recognition at AFIT. The only modification necessary is that all the test trials make use of all forty-five gait samples to generate classifiers while the AFIT data collection system when finished will have only five gait samples with which to generate a classifier. A final test case is consequently accomplished in order to ensure that the optimum algorithm/parameter type/pre-procesing method is still effective when limited to only five gait samples with which to form a classifier.

**Summary**

This chapter discusses the overall methodology by which research is performed in this thesis in order to solve the problem of the identification of individuals solely on the basis of their gaits. This chapter examines the sample data sets to be used in the research including one that already exists and another that is currently in production but which is being created to be compatible with the algorithms that are coded in this paper. It explains how the raw data is taken and analyzed in order to extract coherent and useful data regarding distinguishable traits of individuals relating to identifying them by their gaits. This chapter also explores the various potential methods for classifying the data sets used in this thesis and how they run on a variety of sample sets and compare with each other for optimization, accuracy, and efficiency purposes. Overall, this chapter details how a practical methodology for identifying people on the basis of their gaits is designed and tested in order to work for a specific type of data set that records the position of body markers through time. This first successful laboratory trial serves as a crucial initial stage for developing more general and widely applicable human gait recognition techniques.

# IV. Results and Analysis

## Chapter Overview

The purpose of this chapter is to explain and detail the results and analysis that were produced by the research. The overall purpose human of gait recognition is to identify subjects solely on the basis of their gait. This thesis focuses on the final portion of that process. In this paper, it is assumed that the individual has already been videotaped and that numerical data regarding the positions of the various body parts through time have already been automatically extracted in some fashion. It also assumes that the data has been formatted into the standard configuration of human gait recognition data that is used by the developed MatLab programs. From this point in the gait recognition problem, the thesis focuses on the data analysis that can be performed on such a data set and applies these techniques to several sample data sets in order to provide practical examples. This chapter discusses the various types of statistics and descriptive data parameters that can be derived from the original data set and the methods of creating them. It also examines various methods of identifying an individual on the basis of these extracted data parameters and explores several different means of identification. The results of these identification techniques on the sample data sets also suggest certain advantages and disadvantages to the differing techniques which are also discussed. Finally, the chapter provides resolution to the research question, a preferred algorithm for

extracting data parameters and identifying individuals on the basis of those parameters is found, and its scalability, efficiency, and possible drawbacks are discussed.

**Data Generation**

The first step in generating results for the various algorithms was to create the original data set that the programs would all run on. A MatLab program was written that takes the raw data generated by Dr. Leif Finkel in his gait recognition research and calculates various useful identifying statistics for each person's walk. The original data lists a number of horizontal and vertical coordinates corresponding to the positions of body sensors placed on subjects as they walk on a treadmill. These horizontal and vertical coordinates are recorded for each frame with roughly 30 frames per second. The MatLab program looks at each individual frame for every walk and calculates the foot separation distance for every frame. Using this measure, the program determines every time the feet cross each other and the time corresponding to one crossing to the time corresponding to the second crossing from the first time constitutes one "gait cycle". Having broken up any gait into a series of gait cycles the program then calculates various statistics such as the time for one gait cycle, speed, or maximum stride length. All the statistics for every gait cycle are then averaged to create mean descriptive statistics for every person's gait. As there were forty-five tests walks, nine per person for five people, the database on which all subsequent algorithms ran constituted a matrix of forty-five vectors. The initial fourteen static statistics calculated per gait are as follows:

1. Mean time per full gait cycle (sec)

2. Mean number of frames per full gait cycle
3. Average length of a stride in the gait (mm)
4. Maximum length of a stride (mm)
5. Average height (cm)
6. Average head/waist separation (mm)
7. Average upper arm length (mm)
8. Average lower arm length (mm)
9. Average upper leg length (mm)
10. Average lower leg length (mm)
11. Average distance of the wrist from waist (mm)
12. Average angle between right leg and left leg (degrees)
13. Average knee angle (degrees)
14. Average speed (km/hr)

Dynamic parameters, in contrast, are only expressible as a long sequence of values and are created by selecting a characteristic of an individual that can be assessed from a single frame and recording the value of that parameter for every frame within every gait cycle for which there is data. Once every gait has numerical values for every desired dynamic parameter in every frame in each gait cycle, the gait cycles are separated. Each gait cycle is then scaled to the same number of frames, and all the gait cycles are averaged together on a frame-by-frame basis. Since variation of movement between gait cycles for an individual during a single walk tends to be very low, this process works quite well at generating a smooth, representative average curve for the vast

majority of characteristics of potential interest. This average curve describes dynamic parameters and is represented by a long series of numbers as opposed to the single numbers that represent static parameters.

Having defined dynamic parameters, the next question was to determine which parameters would be of significance for the analysis and identification of human gaits. Based on the those parameters deemed of interest in previous related research endeavors as well as the availability of data for this particular problem instance, a set of six distinct dynamic parameters to be recorded and analyzed was eventually chosen. The dynamic parameters used are listed below:

15. Leg separation angle (degrees)
16. Right knee angle (degrees)
17. Separation of feet (mm)
18. Head height (mm)
19. Separation between wrists (mm)
20. Foot separation divided by average wrist separation

For the remainder of the chapter, gait parameters will occasionally be referred to by their corresponding parameter number for the purposes of brevity.

Having successfully chosen the physical parameters to focus on and extracted them from the sample gait data, the next step involves attempting to identify individuals on the basis of those parameters. In this thesis, three different identification algorithms

are tested on data sets in order to determine the efficiency, accuracy, and flexibility as well as other qualities of the programs. The three functions are the Minimum Euclidean Distance function, the Minimum Mahalanobis Distance function and the Quadratic Discriminant Function. Each of the programs was written in MatLab and reads data stored in a MatLab file.

The first issue that needed to be addressed in the creation of each of the three algorithms was the proper use of the data set. The algorithms all use classifiers which require a certain amount of the data, in this case a series of gait vectors clearly labeled with the individual who produced them, in order to "train" a classifier and then use another portion, in this case gait vectors without the identity of the individual who created them, as a sample to classify. In the case of the Minimum Euclidean Distance classifier and the Minimum Mahalanobis Distance classifier, the MatLab programs are written to look at the data set, group all the gait samples for each individual together and then average them to produce a mean sample for that person that future samples can be compared to. In the case of the Quadratic Discriminant Function, the creation of a classifier is more complex and involves not only finding the mean of the gait samples but also the covariance matrices from the gait samples for each individual.

**Leave-One-Out Cross-Validation Technique**

On the basis of the percentage of the number of gaits correctly associated with a sample subject, it is possible to demonstrate the accuracy of the algorithm on the associated data set. An important concern, however, is that the classifier is not trained on

the entire data set because that would involve classifying samples with a function that had been trained on them initially. Training on all the samples and then classifying the same samples is known as the Resubstitution method and generates the Apparent Error Rate. For relatively obvious reasons though, the Apparent Error Rate tends to yield approximations of accuracy that are too high as it is prone to generating classifiers that are trained for identifying specific examples rather than being effective on more general samples of data not included in the training set. In order to deal with this problem, it was consequently necessary to generate the more reliable Accurate Error Rate for each of the algorithms.

The Actual Error Rate can be generated from a number of different perspectives so a challenge for this thesis was determining which method would be most appropriate for the problem of human gait recognition and the particular algorithms being used here. The first method of generating the Actual Error Rate considered was the Holdout Method. In the Holdout Method, the total data sample is divided into two. A classifier is trained using only about half of the data set. The other half of the data is then used only for classification purposes. The Holdout Method has the advantage of not classifying the same samples that were used to create the initial classifier. However it was found that the method required large samples in order to generate accurate estimates of misclassification. In this example, the sample data set of 45 gait samples was found not to be sufficiently large for the Holdout Method to be utilized.

The method used to generate the Actual Error Rate was consequently the Leave-One-Out Cross-Validation technique. This technique proved superior to both the

Resubstitutio method and the Holdout method in that it avoids classifying samples that were also used to train the classifier and that is effective in approximating the error rate even with relatively small sample sizes. The overall concept of the Cross-Validation technique is to only remove one sample from the entire data set and then train the classifier on all the remaining samples. Once the classifier is trained, the removed sample is identified using that specific classifier. In this thesis, one of the 45 gaits would be identified as belonging to one of the five subjects. Having then identified exactly one sample, that sample is reinserted into the original data set, a new sample is removed and a different classifier is then trained on the new remaining data set. This process is repeated for each sample in the data set until all of the samples are classified. The process of Cross-Validation consequently yields an Actual Error Rate that is less biased than the normal Apparent Error Rate. This method is used with each of the algorithms in this section, the Minimum Euclidean Distance classifier, the Minimum Mahalanobis Distance classifier and the Quadratic Discriminant Function in order to gauge accuracy.

**Algorithm Implementation**

Having successfully extracted meaningful data parameters from the original data set, chosen a set of algorithms to classify the data separately, and devised a method of accurately measuring the error rate for each algorithm, the next step was to actually develop each algorithm and test it on various sets of data parameters. The rest of this chapter will focus on the development of each of the algorithms as well as the results of their application to a variety of data parameter sets.

**Minimum Euclidean Distance/Static Parameter**

The first and simplest of the classification algorithms to be coded was the Minimum Euclidean Distance classifier. The overall concept of this classifier is relatively straightforward with respect to this particular problem and the details of its application. The general concept of the Minimum Euclidean Distance classifier is that from the training set it takes all of the parameters or vectors associated with any individual and averages then to form a basis for comparison against future samples. Once a mean has been constructed for each individual the algorithm then looks at the set of gaits to be classified which also consists of a series of vectors. To classify each individual vector of parameters corresponding to one gait, the Euclidean distance is found between that vector and each of the means calculated for each of the gait subjects. In this particular case, this yields five Euclidean distances corresponding to the five individuals who contributed gait samples. The Euclidean distance under these circumstances gives an intuitive understanding of the "closeness" of the sample vector to the subject means despite the fact that these vectors frequently exist in many more than three-dimensional space. The lowest Euclidean distance number out of the five consequently indicates to the algorithm that the corresponding subject is the one who originally generated that particular gait and the algorithm labels the sample accordingly. This process is repeated for each of the samples in the data set using the Cross-Validation method and comprises the relatively straightforward implementation of the Minimum Euclidean Distance classifier.

Using the Cross-Validation method of gauging accuracy, each algorithm including the Minimum Euclidean Distance classifier produces a contingency matrix as its final output. This contingency matrix is essentially a complete tally of how accurately the classifier identifies each of the gaits in the sample. A contingency matrix consists of a matrix that has the same number of rows and columns as the number of classes of data points to be identified as well as an additional row and column for holding the sums of the original rows and columns. The rows, not including the sum row, correspond to the classes that the data points are known to belong to. The columns correspond to the classes that the classifier assigns to each data point, not including the sum row. A completely accurate classifier will consequently produce a contingency matrix with zeros everywhere except along the diagonal and in the sum column and row. This indicates that all data points were classified in the class that they actually belong to. Values off the main diagonal are the result of misclassifications. A contingency matrix is consequently an ideal way of calculating algorithm accuracy as it is only necessary to take the number of data points on the main diagonal divided by the total number of points in order to get the Actual Error Rate. As a result, a contingency matrix is the output for each of the three classification algorithms; the Minimum Euclidean Distance classifier, Minimum Mahalanobis Distance classifier, and the Quadratic Discriminant classifier.

Once the classifier algorithm had been written and a means of determining its accuracy had been determined the next step was to test the algorithm on a number of different data sets. The first data set to test the algorithm on was the 14 parameter set consisting of the raw data parameters generated at the beginning of this chapter. Running

this data set through the Minimum Euclidean Distance classifier yielded the following contingency matrix in Figure 2.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 3 | 2 | 0 | 4 | 0 | 9 |
| Actual | 0 | 4 | 0 | 2 | 3 | 9 |
| Values | 0 | 2 | 3 | 4 | 0 | 9 |
| | 0 | 2 | 0 | 6 | 1 | 9 |
| | 0 | 3 | 0 | 4 | 2 | 9 |
| Totals | 3 | 13 | 3 | 20 | 6 | 18 |

**Figure 2. MED/Static Parameters/Unreduced data contingency matrix**

As previously discussed this contingency matrix reflects the Actual Error Rate generated by the Cross-Validation procedure. The numbers along the diagonal except for the last row indicate the number of correctly identified gaits per class. This contingency matrix indicates an overall error rate of 60%.

With a relatively high error rate the next stage in the thesis was to explore methods of improving the classification rate and analyze the problem in order to understand the underlying issues that were producing misclassifications.

The first technique employed to gain additional insight into the data was principal component analysis. Principle component analysis is a data reduction technique that transforms that data into orthogonal principal components that account for successively smaller percentages of the variance of the original data. A loadings matrix indicates which of the original data parameters corresponds to each of the principal components

and to what degree.  Examining the eigenvalues associated with each principal component indicates the percentage of the variance accounted for by each of the principal components.  Graphing the principal components one against another indicates relationships that exist between each pair and also suggests whether or not those components would be effective in discriminating between the gaits using the classifier. Specifically, if a graph of two principal components plotted against each other contains distinctly clustered groupings of subjects that suggests the principal components will be useful in distinguishing those subjects.  The absence of distinctly clustered groupings of subjects suggests that the principal components will be ineffective in discriminating between the included subjects.  In this way, principal component analysis can lend additional insight into a given data set and hopefully improve the error rate in classification.

Initially for the thesis, a MatLab program was written to calculate principal components and their associated eigenvalues.  The corresponding first six eigenvalues with respective variances for the static parameter data set are given in Table 1.

**Table 1.  Eigenvalues and Percent Variance of Principal Components**

| | PC1 | PC2 | PC3 | PC4 | PC5 | PC6 |
|---|---|---|---|---|---|---|
| Eigenvalue | 6.02 | 3.70 | 1.65 | 1.02 | 0.66 | 0.45 |
| Variance % | 44.62 | 27.42 | 12.21 | 7.54 | 4.89 | 3.32 |

One method of determining the number of principal components in the analysis is to sum up the amount of variance accounted for by the principal components until the cumulative amount reached exceeds a certain limit, often 95%.  From that criterion, it

follows that, in this case, five of the principal components should be retained. The next step involved examining the loadings matrix in order to determine which data parameters were associated with which principal component and how. Each row of a loadings matrix represents the degree to which one parameter is associated with each of the principal components. Frequently, almost all of the variance of a parameter will be captured by a single principal component. In order to visually indicate these relationships, the largest absolute value in each row of a loadings matrix is bolded in Table 2. This allows a rapid, intuitive grasp of which parameters each principal component generally represents.

**Table 2. Loadings Matrix for Static Parameter Data**

| | PC1 | PC2 | PC3 | PC4 | PC5 | PC6 |
|---|---|---|---|---|---|---|
| 1 | **-0.9371** | -0.1533 | 0.0107 | 0.1002 | -0.1559 | -0.1359 |
| 2 | **-0.906** | -0.2262 | -0.2398 | -0.0536 | 0.0386 | 0.0077 |
| 3 | **0.9052** | -0.3093 | -0.2116 | 0.1157 | -0.0969 | -0.0245 |
| 4 | **0.9268** | -0.2681 | -0.1851 | 0.0899 | -0.0857 | -0.0505 |
| 5 | -0.4047 | **-0.8344** | -0.3099 | 0.0817 | 0.143 | 0.0848 |
| 6 | -0.0513 | **-0.9269** | -0.0224 | 0.0513 | 0.2953 | 0.1504 |
| 7 | 0.0183 | **-0.7794** | -0.0376 | 0.499 | -0.3215 | -0.0691 |
| 8 | -0.0903 | 0.1406 | **-0.9182** | -0.1296 | 0.1495 | 0.2575 |
| 9 | -0.0887 | 0.4579 | **-0.7777** | -0.0086 | -0.2262 | -0.3124 |
| 10 | -0.0449 | **-0.7696** | -0.0275 | -0.3478 | 0.1955 | -0.4584 |
| 11 | -0.3069 | 0.3851 | -0.0782 | **0.7868** | 0.2847 | -0.0888 |
| 12 | **0.9591** | -0.2364 | -0.0317 | 0.0499 | -0.0784 | -0.0003 |
| 13 | **0.6983** | 0.3907 | -0.0047 | 0.0782 | 0.4538 | -0.2194 |
| 14 | **0.9676** | -0.1036 | -0.1174 | -0.0054 | -0.0179 | 0.0962 |

Based on which parameters are associated with each principal component, it can be inferred as to what each principal component represents. The first principal component is

associated with all the statistics that measure aspects of the gait such as number of frames, time, average length and maximum length of strides. The first principal component also includes parameters relating to the angle between the two legs as well as gait speed. From these observations, it is clear that the first principal component focuses on those measurements relating to the relationship between the two legs.

The second principal component has a different five measurements associated with it. The second principal component appears to contain almost all the remaining parameters dealing with the height of the individual as well as the length of the limbs. Interestingly, only two limb length parameters are not included in the second principal component and these are the lower arm length and upper leg length. These two lengths, however, are the only measurements associated with the third principal component. It is not immediately obvious what would cause these two lengths to be separated into a third principal component. The fourth principal component, by contrast, is only associated with the average separation from the waist of the wrist during the gait. As this measurement is concerns a factor qualitatively different from all the other measurements, it makes a great deal of sense for it to be associated with its own distinct principal component.

The loadings matrix overall indicates that the vast majority of the data parameters are associated primarily with the first two principal components, only one with the fourth and none primarily with the fifth principal component. Graphing the first three principal components against each other as they account for the majority of the parameters yields

The relationships between the three principal components are graphed in Figures 3, 4, and 5.

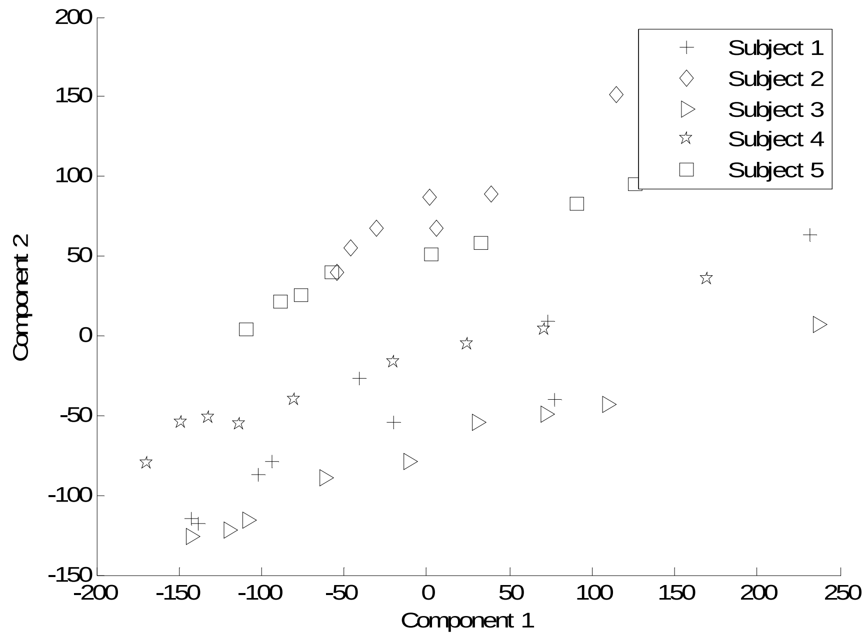


**Figure 3. Principal Component 1 vs. Principal Component 2**

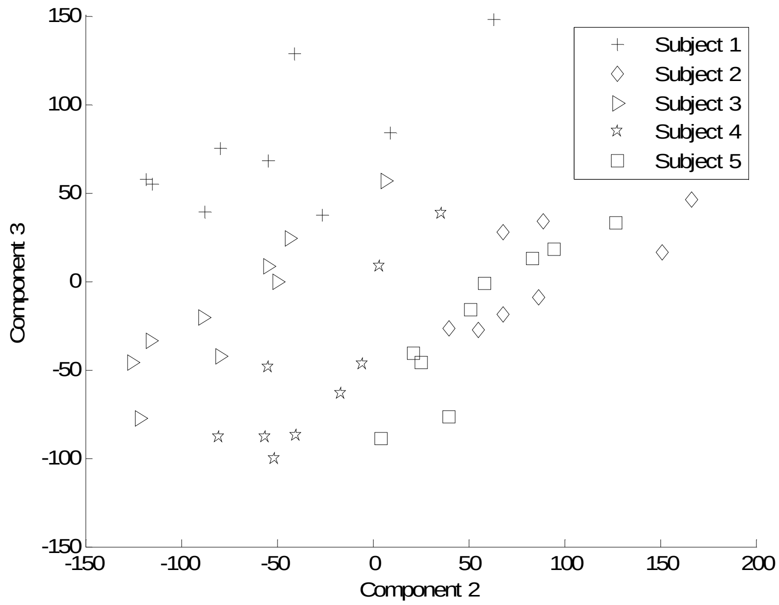


**Figure 4. Principal Component 2 vs. Principal Component 3**

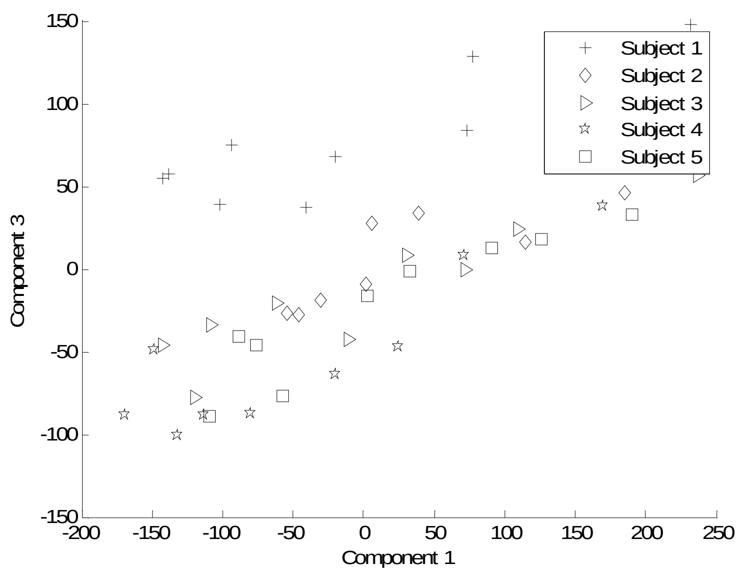


**Figure 5. Principal Component 1 vs. Principal Component 3**

From these graphs of the components against each other it is relatively clear that there is some distinct grouping between the first and second principal component with less distinct grouping involving the third component. This agrees with the correlations indicated by the loadings matrix. With this data it then became possible to run the Minimum Euclidean Distance classifier on the first five principal components of the data set. This resulted in the following contingency matrix in Figure 6.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 7 | 1 | 0 | 1 | 0 | 9 |
| Actual | 0 | 4 | 0 | 1 | 4 | 9 |
| Values | 0 | 1 | 4 | 4 | 0 | 9 |
| | 0 | 1 | 0 | 6 | 2 | 9 |
| | 0 | 3 | 0 | 4 | 2 | 9 |
| Totals | 7 | 10 | 4 | 16 | 8 | 23 |

**Figure 6. MED/Static Parameters/Principal component contingency matrix**

This has an error rate of 48.9%. It represents an improvement over looking at the raw original data but still appears to be improvable. The next phase in the thesis consequently focused on techniques for further improving the recognition rate of the classifier.

The fact that eliminating some of the extraneous data improved the classification rate using principal components suggested a relatively common phenomenon in the data wherein poorly chosen data parameters were adding “noise” to the classifier and hurting the ability of more accurately descriptive parameters to distinguish the different subjects. Ideally, the solution to this sort of problem would be to run the classifier with all possible combinations of parameters and simply choose the combination that most accurately identifies the gaits. However, finding all possible combinations of parameters one

through fourteen and then running the classifier for each would be prohibitively time-consuming and the problem would be significantly magnified were this approach to be applied to larger data sets. The eventual solution was to develop two programs to identify the most valuable data parameters. The first program starts with all fourteen parameters, determines the accuracy of the classifier and then eliminates one parameter at a time until it finds a new combination that produces a higher accuracy. The second program starts with no parameters and adds one at time until doings so stop improving the accuracy. Eliminating parameters yields the contingency matrix graphed in Figure 6.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 0 | 9 | 0 | 0 | 0 | 9 |
| Values | 0 | 0 | 9 | 0 | 0 | 9 |
| | 0 | 0 | 0 | 9 | 0 | 9 |
| | 0 | 0 | 0 | 0 | 9 | 9 |
| Totals | 9 | 9 | 9 | 9 | 9 | 45 |

**Figure 6. MED/Static Parameters/Eliminating parameters contingency matrix**

This method produces a 100% classification rate and includes only parameters 3, 4, and 11. Adding parameters yields the contingency matrix displayed in Figure 7.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 0 | 9 | 0 | 0 | 0 | 9 |
| Values | 0 | 0 | 9 | 0 | 0 | 9 |
| | 0 | 0 | 0 | 9 | 0 | 9 |
| | 0 | 0 | 0 | 0 | 9 | 9 |
| Totals | 9 | 9 | 9 | 9 | 9 | 45 |

**Figure 7. MED/Static Parameters/Adding parameters contingency matrix**

This program also finds a 100% classification solution by retaining only parameters 6, 7, and 8.

Consequently, of the three methods of training on all the original parameters, utilizing the principal components on all parameters, or looking at only a subset of the original parameters, the method of looking at a subset of the original parameters produced significantly greater improvements in classification. The fact that this was accomplished using only three parameters suggests that many of the parameters contributed only noise or little additional information to the gait signature. Further, with an effective reduction of parameters the Minimum Euclidean Distance classifier operates with 100% accuracy on this data set.

**Minimum Mahalanobis Distance/Static Parameters**

The Minimum Mahalanobis Distance classifier was the second method used to classify the forty-five gait signatures used in this research. The coding of the Minimum Mahalanobis Distance algorithm was quite similar to that of Minimum Euclidean Distance classifier. Both algorithms rely on the calculation of a type of distance between the mean of the data parameters for all gaits from one subject and each sample gait to be identified. While the Miniumum Euclidean Distance is calculated by taking the difference between two vectors, squaring the terms, summing the terms and then taking the square root of the result (see eq.1), the Minimum Mahalanobis Distance only differs because it also divides the terms by the variance of the data set before squaring, summing, and then taking the square root of the terms (see eq. 2). The Minimum

Mahalanobis Distance algorithm was also coded to employ the Cross-Validation technique in order to avoid identifying a gait using a classifier trained on that gait.

Using the Minimum Mahalanobis Distance classifier with the original data set produced the following contingency matrix in Figure 7.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 7 | 0 | 0 | 1 | 1 | 9 |
| Actual | 1 | 7 | 0 | 1 | 0 | 9 |
| Values | 0 | 0 | 8 | 0 | 1 | 9 |
| | 0 | 0 | 6 | 3 | 0 | 9 |
| | 5 | 0 | 1 | 1 | 2 | 9 |
| Totals | 13 | 7 | 15 | 6 | 4 | 27 |

**Figure 7. Mahalanobis/Static Parameters/Unreduced data contingency matrix**

This contingency matrix indicates an error rate of 40% which is somewhat better than the error rate for the Minimum Euclidean Distance classifier. The improved classification makes sense, however, as the Minimum Mahalanobis Classifier takes additional information regarding variance into account in making its assignments.

The next testing phase involves using the Minimum Mahalanobis Distance classifier on the five principal components of the data set instead of the entire original data set. This new problem produces the following contingency matrix in Figure 8.

|  | Predicted Values |  |  |  |  | Totals |
|---|---|---|---|---|---|---|
|  | 2 | 1 | 3 | 3 | 0 | 9 |
| Actual | 0 | 8 | 0 | 0 | 1 | 9 |
| Values | 3 | 0 | 6 | 0 | 0 | 9 |
|  | 6 | 0 | 1 | 2 | 0 | 9 |
|  | 4 | 2 | 0 | 0 | 3 | 9 |
| Totals | 15 | 11 | 10 | 5 | 4 | 21 |

**Figure 8. Mahalanobis/Static Parameters/Principal components contingency matrix**

This contingency matrix also has a 53.3% error rate which is again slightly better than the performance of the Minimum Euclidean Distance classifier on the same data set. The higher error rate in comparison to the unreduced data example is presumably the result of loss of data from application of the principal components technique.

Subsequently, the Minimum Mahalanobis Distance classifier was tested using the two programs that respectively start with all parameters and then successively eliminate parameters in order to achieve the optimum accuracy and also start with no parameters and iteratively add parameters in order to achieve optimum accuracy. Eliminating parameters produced the contingency matrix in Figure 9.

|  | Predicted Values |  |  |  |  | Totals |
|---|---|---|---|---|---|---|
|  | 6 | 0 | 2 | 0 | 1 | 9 |
| Actual | 0 | 8 | 0 | 1 | 0 | 9 |
| Values | 0 | 0 | 9 | 0 | 0 | 9 |
|  | 1 | 0 | 6 | 2 | 0 | 9 |
|  | 0 | 0 | 1 | 2 | 6 | 9 |
| Totals | 7 | 8 | 18 | 5 | 7 | 31 |

**Figure 9. Mahalanobis/Static Parameters/Eliminating parameters contingency matrix**

This contingency matrix is produced by eliminating only parameter 11 and keeping all the remaining parameters thereby generating an error rate of 31.11%. This error rate is significantly worse than the results for the Minimum Euclidean Distance classifier which had an error rate of 0.0%.

Adding parameters yields the contingency matrix in Figure 10.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 0 | 8 | 0 | 0 | 1 | 9 |
| Values | 0 | 0 | 9 | 0 | 0 | 9 |
| | 0 | 0 | 0 | 8 | 1 | 9 |
| | 0 | 1 | 0 | 0 | 8 | 9 |
| Totals | 9 | 9 | 9 | 8 | 10 | 42 |

**Figure 10. Mahalanobis/Static Parameters/Adding parameters data contingency**

This contingency matrix is produced using only parameters 6 and has an error rate of 6.7% which is still worse than the Minimum Euclidean Distance classifier of 0.0%. This indicates that parameters 6 is unusually effective at discriminating between the five people in this trial.

Overall, the Minimum Mahalanobis Distance function appears to have performed less well on the PCA data and limited parameter cases than the Minimum Euclidean Distance function. This is likely due to the fact the Minimum Mahalanobis Distance attempts to take the variance of the data into consideration when labeling gait vectors. In

these cases, the variance apparently provided little information of value and may have acted as noise that obscured more accurately distinguishing aspects of the gaits.

**Quadratic Discriminant Function/Static Parameters**

The Quadratic Discriminant classifier utilizes a different method of parsing the gait samples according to the individual who produced them. Whereas the Minimum Euclidean Distance classifier and the Minimum Mahalanobis Distance classifier both look at the average of all a subject's gait parameters in the training set and then find some type of distance between that average and the gait to be identified, the Quadratic Discriminant classifier uses a much more complex algorithm to gauge similarity. The Quadratic Discriminant classifier calculates not only the mean of all the gait vectors for every training subject but also the covariance for every subject. This produces a measure of the closeness of each gait to be identified to the original training gaits for each subject. As in the previous two classifiers, the Cross-Validation technique is employed to make sure that a gait is never identified by a classifier that was also trained on that same gait.

Using the Quadratic Discriminant classifier with the original data set produced the following contingency matrix in Figure 11.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 0 | 7 | 2 | 0 | 0 | 9 |
| Actual | 0 | 1 | 0 | 0 | 8 | 9 |
| Values | 0 | 0 | 0 | 0 | 9 | 9 |
| | 2 | 1 | 6 | 0 | 0 | 9 |
| | 0 | 0 | 9 | 0 | 0 | 9 |
| Totals | 2 | 9 | 17 | 0 | 17 | 1 |

**Figure 11. QDF/Static Parameters/Unreduced data contingency matrix**

This contingency matrix indicates an error rate of 97.78% which is significantly worse than the error rate of the other two classifiers. The incorrect classification, however, is probably the result of attempting to match a quadratic function to too much data.

The next testing phase involves using the Quadratic Discriminant classifier on the five principal components of the data set instead of the entire original data set. This new problem produces the following contingency matrix in Figure 12.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 7 | 0 | 0 | 2 | 0 | 9 |
| Actual | 0 | 9 | 0 | 0 | 0 | 9 |
| Values | 0 | 0 | 9 | 0 | 0 | 9 |
| | 1 | 0 | 0 | 8 | 0 | 9 |
| | 2 | 0 | 0 | 0 | 7 | 9 |
| Totals | 10 | 9 | 9 | 10 | 7 | 40 |

**Figure 12. QDF/Static Parameters/Principal components contingency matrix**

This contingency matrix an 11.11 % error rate which is significantly better than the performance of the other two classifiers on the same data set.

Subsequently, the Quadratic Discriminant classifier was tested using the two programs that respectively start with all parameters and then successively eliminate parameters in order to achieve the optimum accuracy and also start with no parameters and iteratively add parameters in order to achieve optimum accuracy. Eliminating parameters produced the contingency matrix in Figure 13.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 0 | 9 | 0 | 0 | 0 | 9 |
| Values | 0 | 0 | 9 | 0 | 0 | 9 |
| | 0 | 0 | 0 | 9 | 0 | 9 |
| | 0 | 0 | 0 | 0 | 9 | 9 |
| Totals | 9 | 9 | 9 | 9 | 9 | 45 |

**Figure 13. QDF/Static Parameters/ Eliminating parameters contingency matrix**

This contingency matrix is produced by eliminating all parameters except 2, 6, 7, and 8 and produces an error rate of 0.0%. This error rate is the same result as for the Minimum Euclidean Distance classifier which had an error rate of 0.0%.

Adding parameters produced the contingency matrix in Figure 14.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 0 | 9 | 0 | 0 | 0 | 9 |
| Values | 0 | 0 | 9 | 0 | 0 | 9 |
| | 0 | 0 | 0 | 9 | 0 | 9 |
| | 0 | 0 | 0 | 0 | 9 | 9 |
| Totals | 9 | 9 | 9 | 9 | 9 | 45 |

**Figure 14. QDF/Static Parameters/Adding parameters contingency matrix**

This contingency matrix is produced using only parameters 6, 1, and 2 and has an error rate of 0.0% which is equal to the error rate of the Minimum Euclidean Distance classifier.

**Minimum Euclidean Distance/Dynamic parameters**

Having designed each of the three functions - the Minimum Euclidean Distance classifier, the Minimum Mahalanobis Distance classifier, and the Quadratic Discriminant classifier and tested each of them on static parameters, the next step is to modify the algorithms slightly and use them on dynamic parameters that distinguish each of the subject gaits in the sample. Static parameters refer to those parameters describing a gait that can be described using a single number. Examples of a static parameter involve the average height of a subject, the average stride length, or maximum stride length of a subject. Dynamic parameters refers to those distinguishing characteristics of a subject's gait that can only be expressed as a series of numbers. In the context of the current human gait recognition problem and this specific sample set, dynamic parameters will most frequently refer to series of data values representing curves describing the behavior of some parameter of interest as it varies through time over an entire average gait cycle.

Having identified and collected information on the characteristics, the next step was to run the new data through the already constructed classifier functions, the Minimum Euclidean Distance classifier, the Minimum Mahalanobis Distance classifier and the Quadratic Discriminant Function classifier. The purpose of this next stage in

testing was to determine whether collecting dynamic data would improve the classification rates of the existing classifiers.

Exchanging static characteristics for dynamic characteristics with the Minimum Euclidean Distance algorithm requires no significant changes to the algorithm which works in an almost identical fashion. The overall concept of this classifier remains quite straightforward. The general concept of the Minimum Euclidean Distance classifier is that all of the parameters or vectors associated with any individual are taken from the training set and averaged to form a basis for comparison against future samples. The size of the data matrix when using static parameters and having fourteen parameters yielded an average vector that was fourteen elements in length. Replacing those parameters with 100 element long parameters for dynamic parameters creates a much longer vector. Once a mean vector has been constructed for each individual the algorithm then examines the set of gaits to be classified which also consists of a series of vectors, in this case the 600 element long average dynamic parameter vector.

To classify each individual vector of parameters corresponding to one gait, the Euclidean distance is found between that vector and each of the mean vectors calculated for each of the gait subjects. In this particular case, this yields five Euclidean distances corresponding to the five individuals who contributed gait samples. The Euclidean distance under these circumstances gives an intuitive understanding of the "closeness" of the sample vector to the subject means despite the fact that these vectors frequently exist in many more than three-dimensional space. The lowest Euclidean distance of the five consequently indicates that the corresponding subject is the one who originally generated

that particular gait and the algorithm labels the sample accordingly. This is repeated for each of the samples in the data set using the Cross-Validation method and comprises the relatively straightforward concept of the Minimum Euclidean Distance classifier. The method and operation of the algorithm remain essentially unchanged from the static parameter case and as before produce a contingency matrix indicating the classification accuracy as an output.

After making minor modifications to the Minimum Euclidean Distance algorithm in order to allow it to handle the dynamic vectors, the next step was to test the algorithm on a variety of data sets. The algorithm was first tested with all six raw dynamic parameters discussed at the beginning of this section. Running this data set through the Minimum Euclidean Distance classifier yielded the following results in Figure 15.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 0 | 6 | 0 | 3 | 0 | 9 |
| Values | 0 | 0 | 8 | 0 | 1 | 9 |
| | 0 | 2 | 0 | 7 | 0 | 9 |
| | 0 | 0 | 4 | 2 | 3 | 9 |
| Totals | 9 | 8 | 12 | 12 | 4 | 33 |

**Figure 15. MED/Dynamic Parameters/Unreduced data contingency matrix**

Again, this contingency matrix reflects the Actual Error Rate generated by the Cross-Validation procedure. The numbers along the diagonal except for the last row indicate the number of correctly identified gaits per class. This contingency matrix indicates an overall error rate of 26.67%.

The next technique employed to gain additional insight into the data was principal component analysis. The corresponding first six eigenvalues for the initial dynamic parameter data set are displayed along with the percentage of variance accounted for by each principal component in Table 3.

**Table 3. Eigenvalues and Percent Variance for Dynamic Parameter Data**

| | PC1 | PC2 | PC3 | PC4 | PC5 | PC6 |
|---|---|---|---|---|---|---|
| Eigenvalue | 185.54 | 124.50 | 102.50 | 43.14 | 34.47 | 24.37 |
| Variance % | 36.06 | 24.20 | 19.92 | 8.38 | 6.70 | 4.74 |

One method of determining the number of principal components in the analysis is to sum the amount of variance accounted for by the principal components until a certain limit is reached, often 95%. Following that criterion, it follows that five of the principal components should be retained. The next step involved looking at the loadings matrix in order to determine which data parameters were associated with which principal component and the degree of the association. However, in this particular instance the fact that the original data matrix was 600 by 45 means that that the loadings matrix is 600 by 6 in length. This large matrix does indicate which gait vector is associated with each principal component, but no simple relationships are evidenced as the sheer number of data points means that each dynamic parameter will be associated with several if not all principal components. However, as the eigenvalues of the principal components demonstrate the vast majority of the variance is accounted for by the first three principal components.

The graphs of the comparison of the first three principal components against each are displayed in Figures 16, 17, and 18.

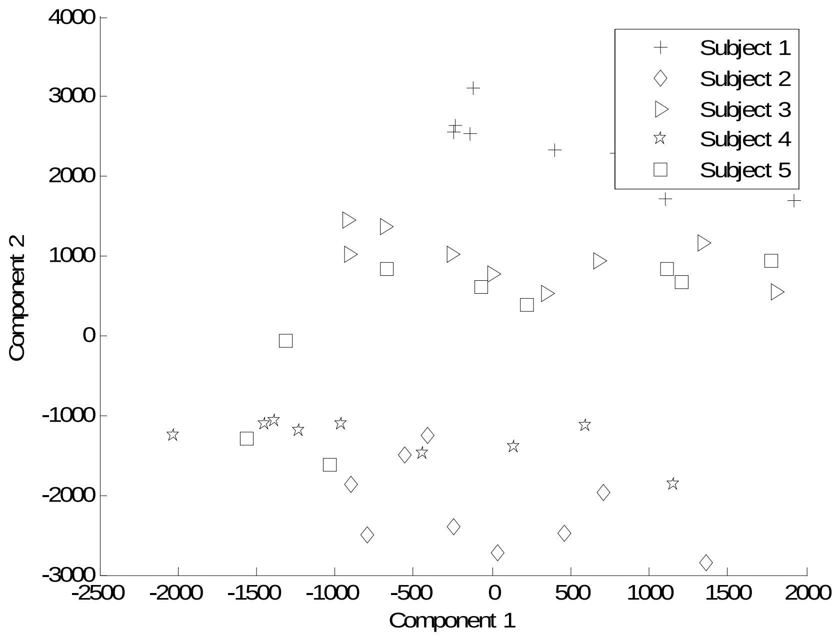


**Figure 16. Principal Component 1 vs. Principal Component 2**

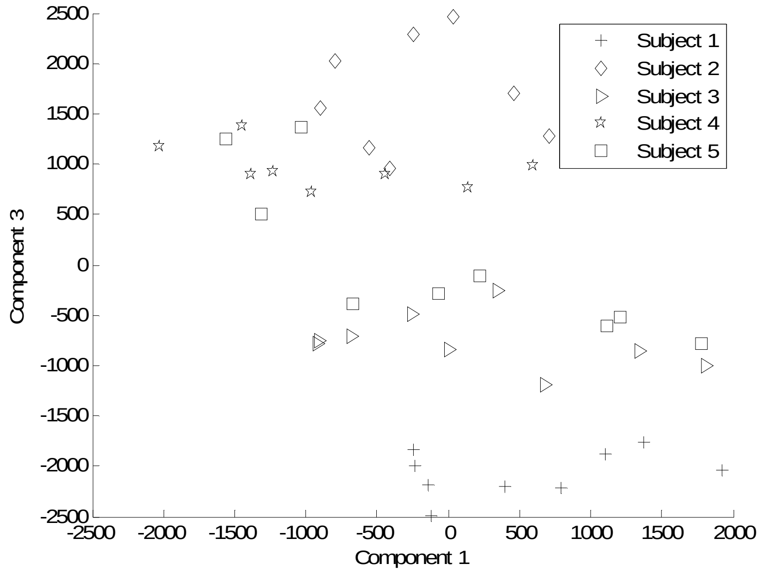


**Figure 17. Principal Component 1 vs. Principal Component 3**

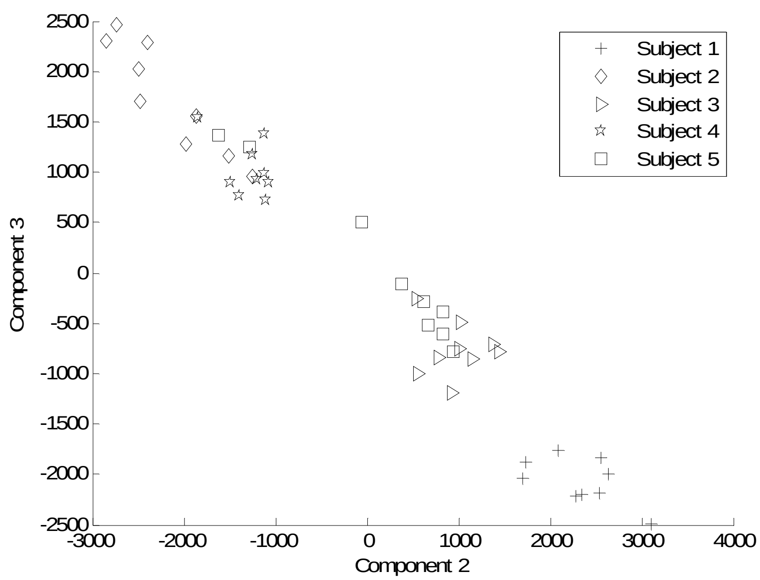


**Figure 18. Principal Component 2 vs. Principal Component 3**

Examining the distribution of the various subject data points in the above component graphs yielded a relatively accurate impression of how well the classifiers

would function on the principal components of the original data. The component graphs demonstrate distinct separation between most groups with only a small amount of overlap remaining. This indicates that the classifiers would probably be able to classify most of the subjects correctly using the above components though perhaps not perfectly.

The next step after the principal component analysis was to run the Minimum Euclidean Distance classifier on the first five principal components. The desired output was the below contingency matrix in Figure 19.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 0 | 6 | 0 | 3 | 0 | 9 |
| Values | 0 | 0 | 8 | 0 | 1 | 9 |
| | 0 | 1 | 0 | 8 | 0 | 9 |
| | 0 | 0 | 4 | 3 | 2 | 9 |
| Totals | 9 | 7 | 12 | 14 | 3 | 33 |

**Figure 19. MED/Dynamic Parameters/Principal components contingency matrix**

This contingency matrix is almost identical to that generated by classifying the unreduced data matrix. The error rate is the same at 26.67%. This agrees with the intuitive notion that if no information is lost during the data dimension reduction that the classification accuracy should not decrease significantly or at all.

Having used the Minimum Euclidean Distance classifier on both the unreduced original data matrix and the principal component matrix the next step was to examine whether eliminating any parameters entirely improved the performance of the algorithm. If removing a parameter led to an improvement in classification accuracy, it would indicate that the parameter itself was too noisy, insignificant, redundant or in some other

way undesirable to include in the final human gait recognition algorithm. To achieve this goal, two algorithms were constructed that behaved in a very similar fashion to reduce the number of parameters while measuring the effect of the reduction on the algorithm's efficiency. The first program begins with all possible parameters included and removes one at a time to see if this improves the classification accuracy. The parameter that leads to the greatest improvement in recognition when removed is removed permanently and the process is repeated until removing parameters no longer leads to performance improvements.

The second algorithm begins with no parameters and looks at classification using each of the possible parameters alone. The parameter that does the best job is kept permanently and the process is repeated, adding parameters iteratively until doing so no longer leads to improvements in classification. Eliminating parameters produced the contingency matrix displayed in Figure 20.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 0 | 6 | 0 | 3 | 0 | 9 |
| Values | 0 | 0 | 8 | 0 | 1 | 9 |
| | 0 | 1 | 0 | 8 | 0 | 9 |
| | 0 | 0 | 4 | 2 | 3 | 9 |
| Totals | 9 | 7 | 12 | 13 | 4 | 34 |

**Figure 20. MED/Dynamic Parameters/Eliminating parameters contingency matrix**

The parameter removal program determines that all parameters 15-20 should be retained except for parameter 17 and has an overall error rate of 24.44%. This is a slight

improvement over retaining all parameters. Adding parameters produced the contingency matrix in Figure 21.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 0 | 7 | 0 | 2 | 0 | 9 |
| Values | 0 | 0 | 8 | 0 | 1 | 9 |
| | 0 | 2 | 0 | 7 | 0 | 9 |
| | 0 | 1 | 4 | 1 | 3 | 9 |
| Totals | 9 | 10 | 12 | 10 | 4 | 34 |

**Figure 21. MED/Dynamic Parameters/Adding parameters contingency matrix**

The parameter addition programs determines that only parameters 16 and 18 are useful yet produces exactly the same overall error rate at 24.44%. This is identical to the solution found by eliminating parameters.

**Minimum Mahalanobis Distance/Dynamic parameters**

The Minimum Mahalanobis Distance classifier was the second method used to classify the forty-five gait signatures according to the individual who had produced them. The difference in this instance was that the Minimum Mahalanobis Distance classifier was operating on the dynamic parameters as opposed to the static ones. When dealing with dynamic parameters, this means that the Euclidean distance is divided by the variance of all the concatenated dynamic parameter vectors, or the variance in this case of a 45 by 600 matrix.

Using the Minimum Mahalanobis Distance classifier with the original data set produced the following contingency matrix in Figure 22.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 0 | 1 | 3 | 1 | 4 | 9 |
| Actual | 1 | 4 | 2 | 1 | 1 | 9 |
| Values | 0 | 3 | 2 | 1 | 3 | 9 |
| | 1 | 2 | 5 | 1 | 0 | 9 |
| | 1 | 1 | 3 | 0 | 4 | 9 |
| Totals | 3 | 11 | 15 | 4 | 12 | 11 |

**Figure 22. Mahalanobis/Dynamic Parameters/Unreduced data contingency matrix**

This contingency matrix indicates an error rate of 75.56% suggesting that including the variance in the gait recognition algorithm for this data set yields more noise than useful identifying data.

The next testing phase involves using the Minimum Mahalanobis Distance classifier on the five principal components of the data set instead of the entire original data set. This new problem produces the following contingency matrix in Figure 23.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 2 | 0 | 0 | 4 | 3 | 9 |
| Actual | 0 | 0 | 0 | 2 | 7 | 9 |
| Values | 0 | 3 | 1 | 0 | 5 | 9 |
| | 0 | 0 | 0 | 4 | 5 | 9 |
| | 0 | 0 | 1 | 4 | 4 | 9 |
| Totals | 2 | 3 | 2 | 14 | 24 | 11 |

**Figure 23. Mahalanobis/Dynamic Parameters/Principal component contingency**

This contingency matrix also has a 75.56% error rate which is significantly worse than the performance of the Minimum Euclidean Distance classifier on the same data set

suggesting that the addition of variance to the gait recognition algorithm is confounding the identification attempts rather than yielding useful distinguishing information.

Subsequently, the Minimum Mahalanobis Distance classifier was tested using the two programs that respectively start with all parameters and then successively eliminate parameters in order to achieve the optimum accuracy and also start with no parameters and iteratively add parameters in order to achieve optimum accuracy. Eliminating parameters produced the contingency matrix displayed in Figure 24.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 1 | 1 | 4 | 0 | 3 | 9 |
| Actual | 0 | 6 | 3 | 0 | 0 | 9 |
| Values | 0 | 2 | 3 | 1 | 3 | 9 |
| | 1 | 1 | 6 | 1 | 0 | 9 |
| | 0 | 1 | 5 | 0 | 3 | 9 |
| Totals | 2 | 11 | 21 | 2 | 9 | 14 |

**Figure 24. Mahalanobis/Dynamic Parameters/Eliminating parameters contingency**

This contingency matrix is produced by eliminating parameter 18 and produces an error rate of 68.89%. This error rate is significantly worse than the results for the Minimum Euclidean Distance classifier which had an error rate of 24.44%.

Adding parameters produced the contingency matrix in Figure 25.

| | | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|---|
| | | 5 | 0 | 0 | 0 | 4 | 9 |
| Actual | | 0 | 6 | 0 | 0 | 3 | 9 |
| Values | | 0 | 0 | 1 | 0 | 8 | 9 |
| | | 0 | 1 | 0 | 2 | 6 | 9 |
| | | 0 | 0 | 2 | 1 | 6 | 9 |
| Totals | | 5 | 7 | 3 | 3 | 27 | 20 |

**Figure 25. Mahalanobis/Dynamic Parameters/Adding parameters contingency matrix**

This contingency matrix is produced using only parameters 17 and 18 and has an error rate of 55.56% which is still worse than the Minimum Euclidean Distance classifier of 24.44%.

### Quadratic Discriminant Function/Dynamic parameters

The Quadratic Discriminant Classifier utilizes a different method of parsing the gait samples according to the individual who produced them. It assumes that the distribution of the gait parameter data is Gaussian in nature and attempts to sketch quadratic boundaries between the groups through higher dimensional space. The Quadratic Discriminant classifier calculates not only the mean of all the gait vectors for every training subject but also the covariance for every subject in order to create these quadratic curves. As in the previous two classifiers, the Cross-Validation technique is employed to make sure that a gait is never identified by a classifier that was also trained on that same gait.

Using the Quadratic Discriminant classifier with the original data set produced the following contingency matrix in Figure 26.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 9 | 0 | 0 | 0 | 0 | 9 |
| Values | 9 | 0 | 0 | 0 | 0 | 9 |
| | 9 | 0 | 0 | 0 | 0 | 9 |
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Totals | 45 | 0 | 0 | 0 | 0 | 9 |

**Figure 26. QDF/Dynamic Parameters/Unreduced data contingency matrix**

This contingency matrix indicates an error rate of 80.00% which is exactly the amount of error predicted if the classifier is no better than random chance. The incorrect classification is presumably the result of attempting to match a quadratic function to such a large quantity of frequently non-quadratic data.

The next testing phase involves using the Quadratic Discriminant classifier on the five principal components of the data set instead of the entire original data set. This new problem produces the following contingency matrix in Figure 27.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 0 | 7 | 0 | 2 | 0 | 9 |
| Values | 1 | 0 | 6 | 0 | 2 | 9 |
| | 0 | 1 | 0 | 6 | 2 | 9 |
| | 0 | 0 | 1 | 2 | 6 | 9 |
| Totals | 10 | 8 | 7 | 10 | 10 | 34 |

**Figure 27. QDF/Dynamic Parameters/Principal components contingency matrix**

This contingency matrix has a 24.44 % error rate which is significantly better than the performance of the other two classifiers on the same data set.

Subsequently, the Quadratic Discriminant classifier was tested using the two programs that respectively start with all parameters and then successively eliminate parameters in order to achieve the optimum accuracy and also start with no parameters and iteratively add parameters in order to achieve optimum accuracy. Eliminating parameters produced the contingency matrix in Figure 28.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 9 | 0 | 0 | 0 | 0 | 9 |
| Values | 9 | 0 | 0 | 0 | 0 | 9 |
| | 9 | 0 | 0 | 0 | 0 | 9 |
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Totals | 45 | 0 | 0 | 0 | 0 | 9 |

**Figure 28. QDF/Dynamic Parameters/Eliminating parameters contingency matrix**

This contingency matrix eliminates no parameters for an error rate of 80.00%. This error rate is worse than the Minimum Euclidean Distance classifier which had an error rate of 24.44%. Adding parameters produced the contingency matrix in Figure 29.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 9 | 0 | 0 | 0 | 0 | 9 |
| Values | 9 | 0 | 0 | 0 | 0 | 9 |
| | 9 | 0 | 0 | 0 | 0 | 9 |
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Totals | 45 | 0 | 0 | 0 | 0 | 9 |

**Figure 29. QDF/Dynamic Parameters/Adding parameters contingency matrix**

This contingency matrix is produced using only parameters 15 and parameter 19 and has an error rate of 80.00% which is significantly worse than the error rate of the Minimum Euclidean Distance classifier which was 24.44%.

The overall inability of the Quadratic Discriminant Function to classify correctly the dynamic parameter data suggests that one or more of the underlying assumptions for its use are being violated here.  To operate correctly, the Quadratic Discriminant Function assumes that the data is Gaussian in nature and can be divided using relatively simple curves in higher-dimensional space.  However, the QDF is often implied in cases where it is only assumed that these assumptions hold to a reasonable degree.  In thise case, that assumption appears to be invalid and indicates that one or both of the underlying requirements do not hold for this case.

**Minimum Euclidean Distance/Static and Dynamic parameters**

Having designed each of the three functions; the Minimum Euclidean Distance classifier, the Minimum Mahalanobis Distance classifier, and the Quadratic Discriminate classifier, and tested each, the next step is to modify the algorithms slightly and use them with both static and dynamic parameters for each of the subject gaits in the sample.  The parameters used will be the same as the parameters previously defined in the chapter except all 20 parameters will be utilized simultaneously.

After making minor modifications to the Minimum Euclidean Distance algorithm in order to allow it to handle both the static and dynamic vectors, the next step was to test the algorithm on a variety of data sets.  The first data set to test the algorithm on was the

20 parameter set consisting of the unreduced raw static and dynamic parameters generated at the beginning of this section. Running this data set through the Minimum Euclidean Distance classifier yielded the following results in Figure 30.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 3 | 1 | 0 | 4 | 1 | 9 |
| Actual | 0 | 4 | 0 | 2 | 3 | 9 |
| Values | 0 | 1 | 4 | 4 | 0 | 9 |
| | 0 | 1 | 0 | 6 | 2 | 9 |
| | 0 | 3 | 0 | 4 | 2 | 9 |
| Totals | 3 | 10 | 4 | 20 | 8 | 19 |

**Figure 30. MED/Static and Dynamic/Unreduced data contingency matrix**

As previously discussed this contingency matrix reflects the Actual Error Rate generated by the Cross-Validation procedure. The numbers along the diagonal except for the last row indicate the number of correctly identified gaits per class. This contingency matrix indicates an overall error rate of 57.78%.

The next technique employed to gain additional insight into the data was Principal Component Analysis. The corresponding first six eigenvalues for the initial static and dynamic parameter data set are displayed in Table 4.

**Table 4. Eigenvalues and Percent Variance for Combined Parameter Data**

| | PC1 | PC2 | PC3 | PC4 | PC5 | PC6 |
|---|---|---|---|---|---|---|
| Eigenvalue | 772.77 | 429.48 | 224.56 | 147.35 | 111.65 | 74.84 |
| Variance % | 43.89 | 24.39 | 12.75 | 8.37 | 6.34 | 4.25 |

One method of determining the number of principal components in the analysis is to sum the amount of variance accounted for by the principal components until the proportion of variance exceeds a certain limit, often 95%.  Following that criterion, it follows that five of the principal components should be retained.  The next step involved looking at the loadings matrix in order to determine which data parameters were associated with which principal component and how.  However, in this particular instance the fact that the original data matrix was 600 by 45 means that the loadings matrix is 600 by 6 in size which is too large to usefully graph, visualize or provide intuitive insight into parameter allocation.

When the first three principal components are graphed against each other the relationship between the principal components is clearly displayed in Figures 31, 32, and 33.

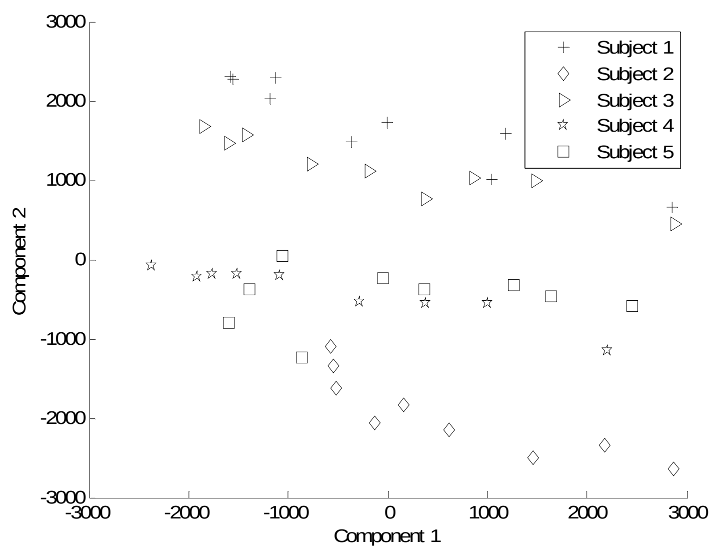


**Figure 31.  Principal Component 1 vs. Principal Component 2**

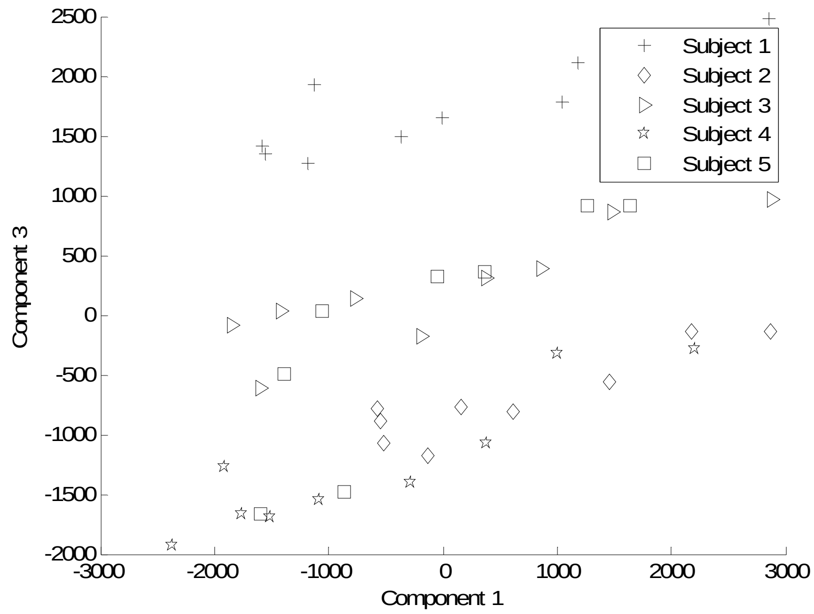


**Figure 32. Principal Component 1 vs. Principal Component 3**

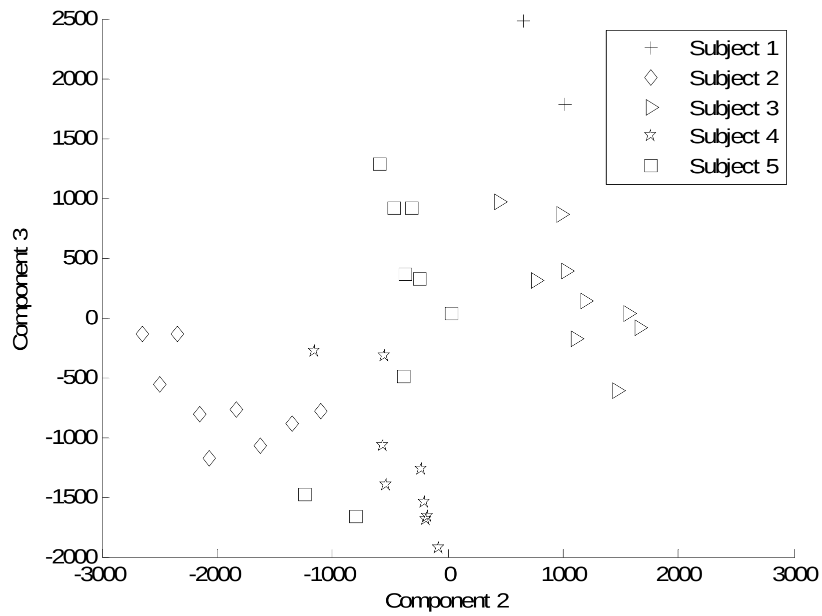


**Figure 33. Principal Component 2 vs. Principal Component 3**

Examining the distribution of the various subject data points in the above principal component graphs yielded a relatively accurate impression of how well the classifiers would function on the principal components of the original data. The component graphs demonstrate distinct separation between most groups with only a small amount of overlap remaining. This indicates that the classifiers would probably be able to classify most of the subjects correctly using the above components though perhaps not perfectly.

The next step after the principal component analysis was to run the Minimum Euclidean Distance classifier on the first five principal components. The desired output was the contingency matrix in Figure 34.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 0 | 6 | 0 | 3 | 0 | 9 |
| Values | 1 | 0 | 8 | 0 | 0 | 9 |
| | 0 | 1 | 0 | 7 | 1 | 9 |
| | 0 | 1 | 0 | 3 | 5 | 9 |
| Totals | 10 | 8 | 8 | 13 | 6 | 35 |

**Figure 34. MED/Static and Dynamic/Principal components contingency matrix**

This contingency matrix is significantly better than that generated by classifying the unreduced data matrix. The error rate is much lower at 22.22%. This agrees with the intuitive notion that if no information is lost during the data dimension reduction that the classification accuracy should not decrease significantly or at all.

Having used the Minimum Euclidean Distance classifier on both the unreduced original data matrix and the principal component matrix the next phase of the thesis was

to examine whether eliminating any parameters entirely improved the performance of the algorithm. Running the two previously discussed parameter reduction programs yielded the following two contingency matrices and classification accuracies. Eliminating parameters produced the contingency matrix in Figure 35.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 0 | 8 | 0 | 0 | 1 | 9 |
| Values | 0 | 0 | 9 | 0 | 0 | 9 |
| | 0 | 0 | 0 | 9 | 0 | 9 |
| | 0 | 0 | 0 | 0 | 9 | 9 |
| Totals | 9 | 8 | 9 | 9 | 10 | 44 |

**Figure 35. MED/Static and Dynamic/Eliminating parameters contingency matrix**

The parameter removal program determines that parameters 2 through 4 and 17 through 19 should be eliminated while retaining the rest and has an overall error rate of 2.22%. This is a slight improvement over retaining all parameters. Adding parameters yielded the contingency matrix in Figure 36.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 0 | 9 | 0 | 0 | 0 | 9 |
| Values | 0 | 0 | 9 | 0 | 0 | 9 |
| | 0 | 0 | 0 | 9 | 0 | 9 |
| | 0 | 0 | 0 | 0 | 9 | 9 |
| Totals | 9 | 9 | 9 | 9 | 9 | 45 |

**Figure 36. MED/Static and Dynamic/Adding parameters contingency matrix**

The parameter addition programs determines that only parameters 2, 6 and 7 are useful yet produces a much better overall error rate of 0.00%. This is an improvement over the solution found by eliminating parameters.

**Minimum Mahalanobis Distance/Static and Dynamic parameters**

The Minimum Mahalanobis Distance classifier was the second method used to classify the forty-five gait signatures according to the individual who had produced them. The difference in this instance was that the Minimum Mahalanobis Distance classifier was operating on both static and dynamic parameters. Using the Minimum Mahalanobis Distance classifier with the original data set produced the following contingency matrix in Figure 37.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 1 | 0 | 0 | 1 | 7 | 9 |
| Actual | 1 | 8 | 0 | 0 | 0 | 9 |
| Values | 0 | 0 | 3 | 6 | 0 | 9 |
| | 1 | 0 | 3 | 5 | 0 | 9 |
| | 2 | 1 | 0 | 2 | 4 | 9 |
| Totals | 5 | 9 | 6 | 14 | 11 | 21 |

**Figure 37. Mahalanobis/Static and Dynamic/Unreduced data contingency matrix**

This contingency matrix indicates an error rate of 53.33% which is significantly better than the error rate for random chance.

The next testing phase involves using the Minimum Mahalanobis Distance classifier on the five principal components of the data set instead of the entire original data set. This new problem produces the following contingency matrix in Figure 38.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 0 | 8 | 0 | 1 | 0 | 9 |
| Values | 1 | 0 | 7 | 0 | 1 | 9 |
| | 0 | 4 | 0 | 5 | 0 | 9 |
| | 0 | 1 | 3 | 2 | 3 | 9 |
| Totals | 10 | 13 | 10 | 8 | 4 | 32 |

**Figure 38. Mahalanobis/Static and Dynamic/Principal components**

This contingency matrix has a 28.89% error rate which is significantly worse than the performance of the Minimum Euclidean Distance classifier on the same data set.

Subsequently, the Minimum Mahalanobis Distance classifier was tested using the two programs that respectively start with all parameters and then successively eliminate parameters in order to achieve the optimum accuracy and also start with no parameters and iteratively add parameters in order to achieve optimum accuracy. Eliminating parameters produced the contingency matrix in Figure 39.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 7 | 0 | 0 | 2 | 0 | 9 |
| Actual | 0 | 7 | 0 | 2 | 0 | 9 |
| Values | 0 | 0 | 6 | 2 | 1 | 9 |
| | 5 | 0 | 2 | 2 | 0 | 9 |
| | 0 | 2 | 1 | 1 | 5 | 9 |
| Totals | 12 | 9 | 9 | 9 | 6 | 27 |

**Figure 39. Mahalanobis/Static and Dynamic/Eliminating parameters**

This contingency matrix is produced by eliminating parameter 8 and 13 and produces an error rate of 40.00%. This error rate is significantly worse than the results for the Minimum Euclidean Distance classifier which had an error rate of 2.22%.

Adding parameters produced the contingency matrix in Figure 40.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 8 | 0 | 0 | 1 | 0 | 9 |
| Actual | 0 | 9 | 0 | 0 | 0 | 9 |
| Values | 0 | 0 | 9 | 0 | 0 | 9 |
| | 0 | 0 | 0 | 9 | 0 | 9 |
| | 0 | 0 | 0 | 2 | 7 | 9 |
| Totals | 8 | 9 | 9 | 12 | 7 | 42 |

**Figure 40. Mahalanobis/Static and Dynamic/Adding parameters contingency matrix**

This contingency matrix is produced using only parameters 2, 6 and 18 and has an error rate of 6.67% which is still worse than the Minimum Euclidean Distance classifier of 0.00%.

### Quadratic Discriminant Function/Static and Dynamic parameters

The Quadratic Discriminant classifier was next applied to the problem of looking at both the static and dynamic parameters combined. For the unreduced data, the fourteen static variables were reduced to one column each and then concatenated with the six 100 element long dynamic parameters to produce a 45 by 614 data matrix. This technique was used as opposed to the 45 by 2000 data matrix used for the MED and Minimum Mahalanobis Distance algorithm because it was found that MatLab had

difficultly taking the covariance of matrices that were 2000 elements in length. Using the Quadratic Discriminant classifier with this original data set produced the following contingency matrix in Figure 41.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 9 | 0 | 0 | 0 | 0 | 9 |
| Values | 9 | 0 | 0 | 0 | 0 | 9 |
| | 9 | 0 | 0 | 0 | 0 | 9 |
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Totals | 45 | 0 | 0 | 0 | 0 | 9 |

**Figure 41. QDF/Static and Dynamic/Unreduced data contingency matrix**

This contingency matrix indicates an error rate of 80.00% which is exactly the amount of error predicted if the classifier is no better than random chance. The incorrect classification, however, is probably the result of attempting to match a quadratic function to too much data.

The next testing phase involves using the Quadratic Discriminant classifier on the five principal components of the data set instead of the entire original data set. This new problem produced the following contingency matrix in Figure 42.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 0 | 8 | 0 | 0 | 1 | 9 |
| Values | 0 | 0 | 9 | 0 | 0 | 9 |
| | 0 | 0 | 0 | 9 | 0 | 9 |
| | 0 | 0 | 0 | 1 | 8 | 9 |
| Totals | 9 | 8 | 9 | 10 | 9 | 43 |

**Figure 42. QDF/Static and Dynamic/Principal components contingency matrix**

This contingency matrix has a 4.44 % error rate which is significantly better than the performance of the other two classifiers on the same data set.

Subsequently, the Quadratic Discriminant classifier was tested using the two programs that respectively start with all parameters and then successively eliminate parameters in order to achieve the optimum accuracy and also start with no parameters and iteratively add parameters in order to achieve optimum accuracy. Eliminating parameters produced the contingency matrix in Figure 43.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 0 | 9 | 0 | 0 | 0 | 9 |
| Values | 0 | 0 | 9 | 0 | 0 | 9 |
| | 0 | 0 | 0 | 9 | 0 | 9 |
| | 0 | 0 | 0 | 0 | 9 | 9 |
| Totals | 9 | 9 | 9 | 9 | 9 | 45 |

**Figure 43. QDF/Static and Dynamic/Eliminating parameters contingency matrix**

This eliminates all parameters except 2, 6, 7, and 8 for an error rate of 0.00%. This error rate is slightly better than the Minimum Euclidean Distance classifier which had an error rate of 2.22%. Adding parameters produced the contingency matrix in Figure 44.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 9 | 0 | 0 | 0 | 0 | 9 |
| Actual | 0 | 9 | 0 | 0 | 0 | 9 |
| Values | 0 | 0 | 9 | 0 | 0 | 9 |
| | 0 | 0 | 0 | 9 | 0 | 9 |
| | 0 | 0 | 0 | 0 | 9 | 9 |
| Totals | 9 | 9 | 9 | 9 | 9 | 45 |

**Figure 44. QDF/Static and Dynamic/Eliminating parameters contingency matrix**

This contingency matrix is produced using only parameters 1, 2 and 6 and has an error rate of 0.0% which is equal to the error rate of the Minimum Euclidean Distance classifier which was 0.00%.

**Analysis of Run Results**

Having successfully run all combinations of algorithms (Minimum Euclidean Distance, Minimum Mahalanobis Distance, and Quadratic Discriminant Function) with all combinations of parameters (static, dynamic, and combined) and all methods of data pre-processing (unreduced data, principle components, adding parameters, eliminating parameters) it is now possible to graph all the results in a single chart which will then be divided into subsections for further analysis. The overall results summary is displayed graphically in Figure 45 and numerically in Table 5.

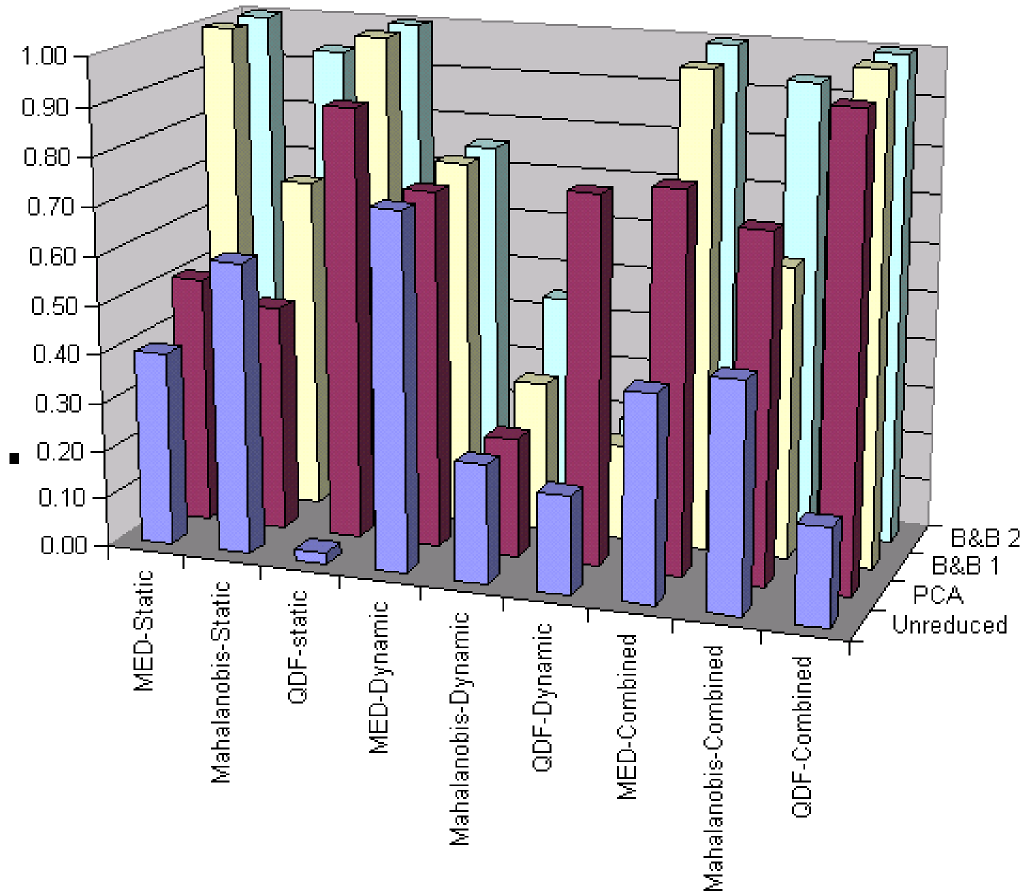


**Figure 45. Classification Accuracies for All Combinations**

**Table 5. All Algorithm/Parameter Combinations**

| | Unreduced | PCA | B&B 1 | B&B 2 |
|---|---|---|---|---|
| MED-Static | 0.40 | 0.51 | 1.00 | 1.00 |
| Mahalanobis-Static | 0.60 | 0.47 | 0.69 | 0.93 |
| QDF-static | 0.02 | 0.89 | 1.00 | 1.00 |
| MED-Dynamic | 0.73 | 0.73 | 0.76 | 0.76 |
| Mahalanobis-Dynamic | 0.24 | 0.24 | 0.31 | 0.44 |
| QDF-Dynamic | 0.20 | 0.76 | 0.20 | 0.20 |
| MED-Combined | 0.42 | 0.78 | 0.98 | 1.00 |
| Mahalanobis-Combined | 0.47 | 0.71 | 0.60 | 0.93 |
| QDF-Combined | 0.20 | 0.96 | 1.00 | 1.00 |

From the collective graph of run results, it is already possible to make several immediate observations regarding algorithm accuracy. It is clear that numerous algorithm/parameter/pre-processing combinations produce completely accurate identification. For this data set then other factors such as algorithm run-time, robustness, and scalability will be considered in order to determine a best algorithm as numerous combinations otherwise satisfy the criteria of completely accurate identification. Further testing on other data sets would also yield additional insight into the performance of the algorithm/parameter/pre-processing combinations that could lead to a best combination. For the purpose of this thesis, alternate factors will be used to make the discrimination.

Another interesting observation is the existence of clear trends in the performance of the various algorithms. In all but two cases the Minimum Euclidean Distance classifier substantially outperformed the Minimum Mahalanobis Distance classifier. In the other two cases the Minimum Mahalanobis Distance classifier outperformed the Minimum Euclidean Distance classifier only marginally. This suggests that the Minimum Euclidean Distance classifier is generally “dominant” over the Minimum Mahalanobis Distance classifier which is to say that it is almost always more accurate or roughly equivalent for this test problem.

In terms of which data parameters, static or dynamic, to consider, the dynamic parameter data set appears to fare poorly overall. From the results graph is clear that there are a few results which are significantly worse than the other combinations and the two most obvious of these are derived from dynamic parameter data sets.

There are several possible reasons for the poor performance of the Minimum Mahalanobis Distance classifier and dynamic parameter data sets which will be explored further. Implications of the run results can be garnered by examining each of the types of parameter data set individually. The results for all algorithms run on static parameter data are listed in Figure 46.

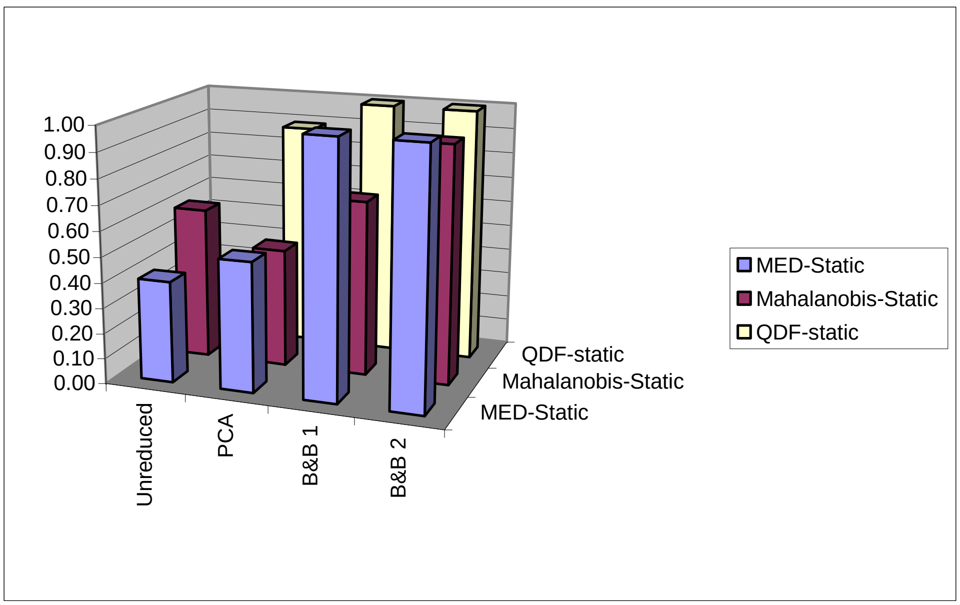


**Figure 46. Classification Accuracies with Static Parameters**

From the static parameter graph it can be inferred that pre-processing Quadratic Discriminant Function data makes a great difference. When the data is reduced to five columns using principal component analysis the classification accuracy increases dramatically. Otherwise, it can be seen that MED classifiers and QDF classifiers offer the possibility of completely accurate classification when only the most useful parameters are retained. The results for all algorithms run on dynamic parameters are listed in Figure 47.

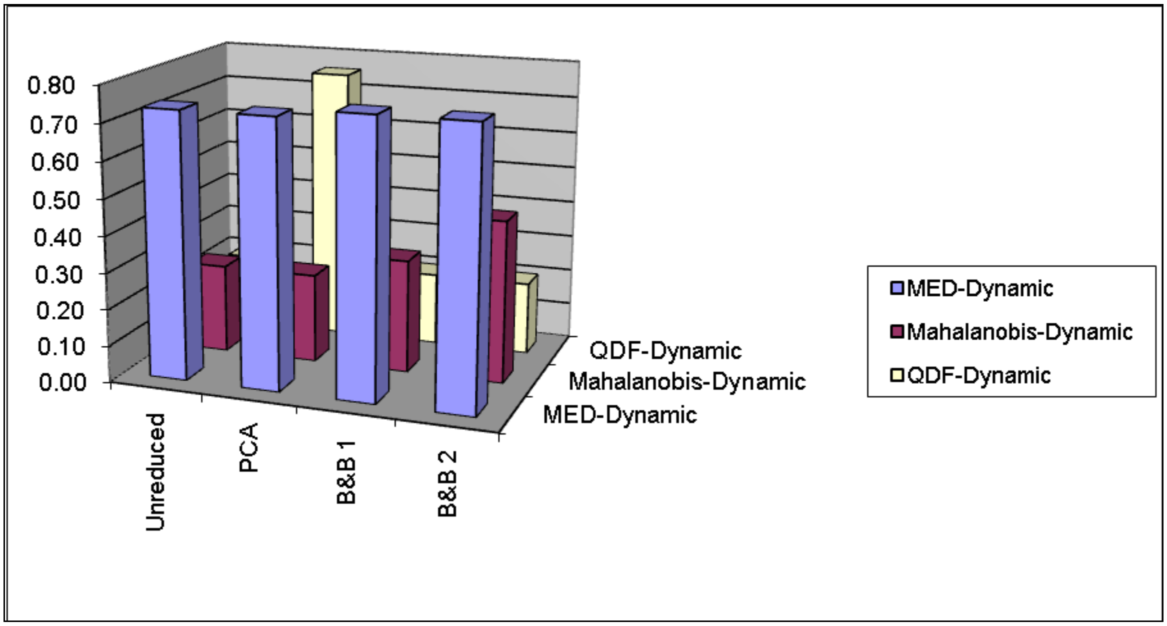


**Figure 47. Classification Accuracies with Dynamic Parameters**

It is readily obvious in the dynamic parameter data set that none of the algorithms comes close to complete identification accuracy. A possible reason for this is that the types of parameters that represent data than can only be expressed in curves are significantly more complex than static parameters which poses a challenge for the classification programs while also allowing for the possibility of significantly more "noise" in the data. It is again clear that the MED classifier is one of the better algorithms and that the QDF classifier is much more accurate when working with the data reduced by principal components analysis, but while using only dynamic parameters the MED algorithm remains the most accurate while the QDF drops off sharply in accuracy. The Minimum Mahalanobis Distance algorithm is even more ineffective under these

circumstances. The results of all algorithms run on combined parameters are listed in Figure 48.

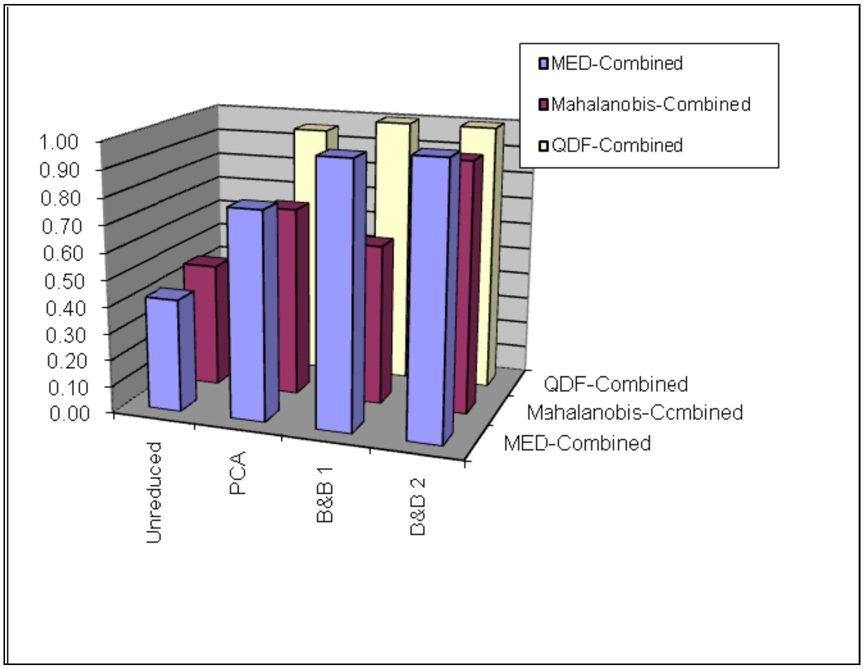


**Figure 48. Classification Accuracies with Static and Dynamic Parameters**

Finally, looking at both the static and dynamic parameters combined there can be seen patterns almost exactly similar to the case where only static parameters were considered. This is significant in that dynamic parameters add substantially more data than the static parameters but appear to have a generally negative effect on classification accuracy as the algorithms perform worse when considering both static and dynamic parameters than when working with solely static parameters. Given that the addition of dynamic parameters can easily increase the size of a data matrix by a factor of one hundred, the lack of increased effectiveness clearly implies that working solely with static parameters is more efficient.

While further testing in a wider variety of scenarios with a greater number of data sets would be highly useful, the current range of tests on the limited combinations of algorithms, types of parameters, and types of data pre-processing are nonetheless instructive in terms of general algorithm behavior and crucial observations. The above observations all support the idea that the best means of human gait recognition considered by this thesis is to consider only static parameters, use the Minimum Euclidean Distance function to classify gaits, and use the simplified branch and bound method in order to add or eliminate only the most useful parameters. Supporting these contentions is a wide variety of very clear trends in the data.

First the Minimum Mahalanobis Distance classifier almost invariably underperforms the Minimum Euclidean Distance classifier. This strongly argues against its general usage.

Next, the Quadratic Discriminant Function has a great deal of difficulty dealing with the larger data matrices created by the dynamic parameters and static/dynamic parameters since it must calculate covariances of matrices which are over 360,000 numbers large. Combined with the fact that the Quadratic Discriminant Function performs roughly equally to and sometimes much worse than the Minimum Euclidean Distance argues against the general usage of the Quadratic Discriminant Function.

Addition of the dynamic parameters causes recognition rates to decrease sharply and this is even more pronounced when the static parameters are removed as well, strongly implying that the static parameters are the more valuable indicators of gait identity.

Lastly, adding or eliminating parameters based on whether or not they improve identification accuracy can by definition only improve the method and should always be performed. Performing these branch and bound algorithms always either improved classification accuracies or left them unaffected.

The overall trends in the data of the myriad algorithm/data parameter type/pre-processing methods strongly suggest that a method that looks at static parameters, adds only those parameters that are most useful and then uses the Minimum Euclidean Distance to classify those gaits is the most efficient and effective method of classifying gaits of all of the combinations studied.

**Practical Implementation**

Having identified a best algorithm, parameter type, and pre-processing method combination through the large number of trials composing the main body of the thesis, the final step in the process is to implement that best process in a more realistic human gait recognition scenario. All the gait recognition problems to this point, regardless of the particular algorithm, have used all of the forty-five sample gaits per data set in order to build a classifier and then employed the Lachenbruch hold-out procedure in order to classify all of the same forty-five gaits and generate an Actual Error Rate (AER) for the algorithm. This is a highly useful process for testing algorithm efficiency in that it employs all the available very limited data in order to both build classifiers and to find gaits to classify. Since it is constructed from a very general point of view, the concept is that it will give a better impression of the effectiveness of an algorithm over a wide

variety of gait problems than it might if it were tailored to a more specific and highly individual gait recognition problem for a specific situation. The notable drawback of this method is that frequently forty-five gaits are not available to build the classifier in order to identify a gait sample when it is found. For the specific gait recognition setup at AFIT that this algorithm is being designed for, it is expected that only five gaits will be available with which to build a classifier. The final portion of the process of designing an optimum algorithm for gait recognition at AFIT is then to verify that the optimum algorithm/parameter type/pre-processing method established during the large number of general trials remains effective when performed on the specific, practical example of interest to this thesis.

The Limited Minimum Euclidean Distance function (LMED) is consequently an algorithm almost identical to the Minimum Euclidean Distance algorithm except that instead of employing the Lachenbruch hold-out procedure it simply builds classifier using one medium speed gait from each of the five subjects and classifies the remaining forty gaits on the basis of those five gaits. As the intention is only to establish that the classification method continues to work on the specific example, the only test case examined is the one determined to be optimum in the previous trials.

Running the Limited Minimum Euclidean Distance function on the unreduced static parameter data produced the contingency matrix depicted in Figure 49.

| | Predicted Values | | | | | Totals |
|---|---|---|---|---|---|---|
| | 8 | 0 | 0 | 0 | 0 | 8 |
| Actual | 0 | 8 | 0 | 0 | 0 | 8 |
| Values | 0 | 0 | 8 | 0 | 0 | 8 |
| | 1 | 0 | 0 | 7 | 0 | 8 |
| | 0 | 0 | 0 | 0 | 8 | 8 |
| Totals | 9 | 8 | 8 | 7 | 8 | 39 |

**Figure 49. LMED/Static/Adding parameter data contingency matrix**

As expected, running this algorithm/parameter/pre-processing technique on the more limited, practical test case where only five gaits are available to generate a classifier also produces an Actual Error Rate of only 2.5% using parameters 2, 5, 6, and 7.

### Research Questions Answered

From Chapter I, the overall research question for this study is: If the human gait is unique to every individual, can a person be identified by their gait? This thesis considered the research question:

> What is the best mathematical and statistical method of identifying people by gait when the available input is the position of body markers at successive points in time?

This question was answered in the process of the research in this thesis. The optimum combination of algorithms, types of parameters, and data pre-processing methods was determined to be the Minimum Euclidean Distance method working with static parameters and using a simplified branch and bound method to only include the

most helpful parameters in the analysis. This was demonstrated by examining the clear data trends from a wide variety of test situations and combinations of methodologies employed throughout this chapter.

**Summary**

This chapter explores the results and analysis of the thesis research into the possibility of human gait recognition in a controlled environment. The thesis begins at the point in the process after body marker location data has already been collected and proceeds to extract salient identifying characteristics from each sample gait of each individual. Having successfully generated a wide variety of descriptive parameters for each gait, the thesis then proceeds to generate and test three different algorithms that attempt to classify the same gaits using three distinctive methods. In order to further explore the issue of optimization of human gait recognition under controlled conditions, a number of other options are examined for each algorithm including the type of parameters to be examined (static or dynamic) as well as a variety of methods of pre-processing the data before it is input into the classification algorithms. After all possible combinations of algorithms/parameter types/pre-processing types are run and the results tabulated with descriptive contingency matrices, the overall results are used to create a series of graphs regarding classification accuracy under a variety of conditions. The graphs are then analyzed in order to generate a number of general trends and interesting observations that are used to understand the behavior of the underlying algorithms. Having successfully developed a number of general propositions regarding the strengths

and weaknesses of the various algorithms, it is then possible to derive an overall best algorithm/parameter type/pre-processing method combination based on the tests run in the course of this thesis research. The chapter ends with the answer to the research question posed: a best mathematical and statistical method of identifying people by gait when the available input is the position of body markers at successive points in time has been selected from all the possible options considered during this thesis.

# V. Conclusions and Recommendations

## Chapter Overview

This chapter explores the conclusions and recommendations reached through this thesis research. This thesis completes the overall goal of identifying individuals on the basis of their gaits beginning at the point where accurate body marker location data over time has already been generated. Detailed in this chapter are both the conclusions derived from this research as well as the significance of the research. The chapter ends with a number of suggested recommendations for future research.

## Conclusions of Research

The thesis has the overall goal of achieving accurate and efficient identification of humans based on their gait under controlled laboratory conditions. The thesis begins with a series of recorded points through time from several individuals who have walked on treadmills while having the positions of markers on their bodies recorded over time. Given this initial raw data, the thesis begins with the development of a MatLab program that reads the raw data files into a usable format of data matrix. It then takes these matrices and calculates a number of statistics from them. These statistics are broken into the categories of static and dynamic. Static statistics refer to gait properties expressible as a single value such as average height while dynamic statistics refer to gait properties that are only expressible as a curve, or a series of values over time, such as average knee angle over one gait cycle.

The next step in the thesis is to create a number of classification algorithms that each employ a different classification principle in order to identify the sample gaits now described by statistics. Once these algorithms have been coded, they are tested on a variety of data sets including data sets consisting solely of static parameters, sets consisting solely of dynamic parameters, and sets consisting of both. In order to add an additional element of complexity and possibility for discovery to the setup, four different methods of pre-processing the data are considered including leaving the data untouched, calculating principal components, as well as adding or eliminating parameters in order to only retain the most useful for classification.

Running each of the scenario configurations and analyzing the results yields numerous identification methods where 100% identification accuracy is achieved. This conclusively answers the overall research question as to whether human gait recognition is possible given the type of data and conditions of the experimentation. The thesis is further able to discriminate between the various methods based on their overall performance throughout all of the tests and thereby generate a number of general conclusions and observations regarding algorithm performance as well as an overall optimum classification algorithm. This answers the second research question in that it identifies a best algorithm/parameter-type/pre-processing method for the human gait recognition problem out of all the considered methods.

**Significance of Research**

This thesis marks a crucial step in that it conclusively establishes a completely automated system for achieving effective human gait recognition with the specific type of

data format available.  This is a non-trivial goal in that there is no definitive or preferred method of achieving human gait recognition in the current scientific community.  As such, the number of mathematical models, the aspects of gait and environmental conditions they concentrate on, and their resulting success rates are many and varied. Success in the endeavor under the controlled conditions available at AFIT and with known accurate body marker position data is a crucial first step intended to serve as a platform for future advances in which human gait recognition will be attempted in increasingly general and robust environments.  The knowledge regarding successes and failures, effective and ineffective methods, as well as the practical familiarity with human gait and the previous research into its characterization is intended to be highly transferrable to more complicated and difficult human gait recognition problems. Compared with the numerous approaches to human gait recognition, this thesis establishes a reliable, practical, and effective solution under the conditions of the provided data which can serve as a useful platform for increasingly complex identification methods.

Human gait recognition itself is a topic that is receiving increasing attention both at home and abroad.  A biometric identification technique that can be performed at a distance without the knowledge of the subject and that is difficult to obstruct is of significant interest to the Department of Defense, Department of Homeland Security, and the United States Air Force because of its many applications to the increasingly unpredictable nature of modern conflict.  This thesis increases the number and variety of

identification methods available and is therefore significant in increasing U. S. capabilities in this arena.

## Recommendations for Future Research

Though this research does produce several methods for completely accurate human gait recognition it does so only under certain relatively ideal circumstances. Future research would be useful in removing many of the simplifying conditions of the laboratory environment of this thesis.

It would be highly useful to test the algorithm on more varied and larger databases. It is reasonable to assume that larger databases containing more varied gaits would prove proportionately more difficult to classify. Additionally, data is currently collected under relatively ideal conditions using a single person walking completely parallel to a stationary camera against a stationary background. Relaxing any of those idealizing constraints would make the person substantially harder to identify and would constitute a rich area for further inquiry. Were both of these limitations solved, a next practical step would be to deploy the cameras to public areas such as banks and malls where theoretically without the constraints of database size and ideal camera angle, a practical amount of automatic human gait identification capability could be realized. The ability to track a single individual from camera to camera solely on the basis of his/her gait would be a highly useful tool in the surveillance of a bank, city or other public area. Any or all of these ideas for moving the concept of human gait recognition out of the laboratory and into practical everyday use would be very fruitful avenues for future research.

**Summary**

This chapter provides summaries of conclusions and recommendations for this thesis. The overarching goal is to achieve practical human gait recognition. This thesis successfully devises a method of solving this problem while comparing and contrasting it to other similar methods. The research demonstrates that the recognition of humans solely on the basis of their gaits is possible within a laboratory environment. Given any database of the gaits of individuals, the algorithm designed here can be used to distinguish the individuals to some reasonable degree. This methodology should serve as a platform for human gait recognition using larger and more diverse databases, in different contexts and environments, and using more diverse individuals. The method provides an important benchmark for a practical level of classification accuracy against which increasingly complex methodologies can be compared. There is a significant amount of research remaining in the field of human gait recognition in order to make the techniques more robust and practical. This chapter also summarizes recommendations for avenues that research might usefully take.

## Appendix

## MATLAB Routines

```
Input Read- The UPenn data initially existed in a set of uniquely
formatted 45 text files.  The combination of the inputread3, worktext15
and loadtext 5 files is specifically tailored to turn these 45 files
into usable MatLab data.Use of this human gait recognition algorithm on
a different data set willnecessarily involve altering one or all three
of these files.  The outputvariable created contains the static data,
graphoutput contains the dynamic data.



function [output,graphoutput]=inputread3
%Derrick Chelliah 28 Feb 2008

output=[]

x=zeros(50,100,7)

[output(1,:) x(1,:,:)]=worktext15(1,'SandyWalk1_0.emf')

[output(2,:) x(2,:,:)]=worktext15(1,'SandyWalk1_5.emf')

[output(3,:) x(3,:,:)]=worktext15(1,'SandyWalk2_0.emf')

[output(4,:) x(4,:,:)]=worktext15(1,'SandyWalk2_5.emf')

[output(5,:) x(5,:,:)]=worktext15(1,'SandyWalk3_0.emf')

[output(6,:) x(6,:,:)]=worktext15(1,'SandyWalk3_5.emf')

[output(7,:) x(7,:,:)]=worktext15(1,'SandyWalk4_0.emf')

[output(8,:) x(8,:,:)]=worktext15(1,'SandyWalk4_5.emf')

[output(9,:) x(9,:,:)]=worktext15(1,'SandyWalk5_0.emf')

[output(11,:) x(11,:,:)]=worktext15(2,'JesseWalk1_0.emf')

[output(12,:) x(12,:,:)]=worktext15(2,'JesseWalk1_5.emf')

[output(13,:) x(13,:,:)]=worktext15(2,'JesseWalk2_0.emf')

[output(14,:) x(14,:,:)]=worktext15(2,'JesseWalk2_5.emf')

[output(15,:) x(15,:,:)]=worktext15(2,'JesseWalk3_0.emf')

[output(16,:) x(16,:,:)]=worktext15(2,'JesseWalk3_5.emf')
```

```
[output(17,:) x(17,:,:)]=worktext15(2,'JesseWalk4_0.emf')
[output(18,:) x(18,:,:)]=worktext15(2,'JesseWalk4_5.emf')
[output(19,:) x(19,:,:)]=worktext15(2,'JesseWalk5_0.emf')
[output(21,:) x(21,:,:)]=worktext15(3,'MaciejWalk1_0.emf')
[output(22,:) x(22,:,:)]=worktext15(3,'MaciejWalk1_5.emf')
[output(23,:) x(23,:,:)]=worktext15(3,'MaciejWalk2_0.emf')
[output(24,:) x(24,:,:)]=worktext15(3,'MaciejWalk2_5.emf')
[output(25,:) x(25,:,:)]=worktext15(3,'MaciejWalk3_0.emf')
[output(26,:) x(26,:,:)]=worktext15(3,'MaciejWalk3_5.emf')
[output(27,:) x(27,:,:)]=worktext15(3,'MaciejWalk4_0.emf')
[output(28,:) x(28,:,:)]=worktext15(3,'MaciejWalk4_5.emf')
[output(29,:) x(29,:,:)]=worktext15(3,'MaciejWalk5_0.emf')
[output(31,:) x(31,:,:)]=worktext15(4,'RobertWalk1_0.emf')
[output(32,:) x(32,:,:)]=worktext15(4,'RobertWalk1_5.emf')
[output(33,:) x(33,:,:)]=worktext15(4,'RobertWalk2_0.emf')
[output(34,:) x(34,:,:)]=worktext15(4,'RobertWalk2_5.emf')
[output(35,:) x(35,:,:)]=worktext15(4,'RobertWalk3_0.emf')
[output(36,:) x(36,:,:)]=worktext15(4,'RobertWalk3_5.emf')
[output(37,:) x(37,:,:)]=worktext15(4,'RobertWalk4_0.emf')
[output(38,:) x(38,:,:)]=worktext15(4,'RobertWalk4_5.emf')
[output(39,:) x(39,:,:)]=worktext15(4,'RobertWalk5_0.emf')
[output(41,:) x(41,:,:)]=worktext15(5,'SusanaWalk1_0.emf')
[output(42,:) x(42,:,:)]=worktext15(5,'SusanaWalk1_5.emf')
[output(43,:) x(43,:,:)]=worktext15(5,'SusanaWalk2_0.emf')
[output(44,:) x(44,:,:)]=worktext15(5,'SusanaWalk2_5.emf')
[output(45,:) x(45,:,:)]=worktext15(5,'SusanaWalk3_0.emf')
```

```
[output(46,:) x(46,:,:)]=worktext15(5,'SusanaWalk3_5.emf')

[output(47,:) x(47,:,:)]=worktext15(5,'SusanaWalk4_0.emf')

[output(48,:) x(48,:,:)]=worktext15(5,'SusanaWalk4_5.emf')

[output(49,:) x(49,:,:)]=worktext15(5,'SusanaWalk5_0.emf')

graphoutput=x;

end
```

Workable Text-Loads data from a single person at a single speed as a matrix.  Runs through the matrix one frame at a time while recording static and dynamic statistics and marking the discrete gait cycles.

```
function [output,x]=worktext15(ident, string)
%Derrick Chelliah 28 Feb 2008
%Command Line: [output,x]=worktext15(1,'SandyWalk3_0.emf')

[matrix,d]=loadtext5(string);


output=[];
currentmatrix=[];
newmatrix=[];
stats=[];
pairedstats=[];
height=[];
times=[];
pairedtimes=[];
stridelengths=[];
headwaist=[];
leftrightfoot=[];
leftrightwaist=[];
leftrightshoulder=[];

k=1;
p=2;
stridelength=0;
framecount=1;
u=[0 0];

gaitframe=zeros(1,100);
legframe=zeros(100,100);
kneeframe=zeros(100,100);
footsep=zeros(100,100);
headbounce=zeros(100,100);
wristsep=zeros(100,100);

%Builds initial matrix of gait data from first frame

currentmatrix=[matrix(1,:);matrix(2,:);matrix(3,:);matrix(4,:);matrix(5
,:);...
matrix(6,:);matrix(7,:);matrix(8,:);matrix(9,:);matrix(10,:);...
matrix(11,:);matrix(12,:);matrix(13,:);matrix(14,:);matrix(15,:);...
matrix(16,:);matrix(17,:);matrix(18,:);matrix(19,:);matrix(20,:);...
matrix(21,:);matrix(22,:);matrix(23,:);matrix(24,:);matrix(25,:);...
matrix(26,:);matrix(27,:);matrix(28,:);matrix(29,:);matrix(30,:)];

%Calculates distance between feet
length=matrix(2,4)-matrix(18,4);
```

```
for j=1:(d-1)
%Builds subsequent gait data matrices from subsequent frames

newmatrix=[matrix(1+30*j,:);matrix(2+30*j,:);matrix(3+30*j,:);matrix(4+
30*j,:);matrix(5+30*j,:);...
matrix(6+30*j,:);matrix(7+30*j,:);matrix(8+30*j,:);matrix(9+30*j,:);mat
rix(10+30*j,:);...
matrix(11+30*j,:);matrix(12+30*j,:);matrix(13+30*j,:);matrix(14+30*j,:)
;matrix(15+30*j,:);...
matrix(16+30*j,:);matrix(17+30*j,:);matrix(18+30*j,:);matrix(19+30*j,:)
;matrix(20+30*j,:);...
matrix(21+30*j,:);matrix(22+30*j,:);matrix(23+30*j,:);matrix(24+30*j,:)
;matrix(25+30*j,:);...
matrix(26+30*j,:);matrix(27+30*j,:);matrix(28+30*j,:);matrix(29+30*j,:)
;matrix(30+30*j,:)];


 if (abs(length)>stridelength)
     stridelength=abs(length);
 end

%Calculates salient statistics per frame
height(j)=(newmatrix(30,5)-newmatrix(11,5));

newlength=newmatrix(4,4)-newmatrix(11,4);

waist=[newmatrix(22,4) newmatrix(22,5)];

leftleg=[newmatrix(2,4)-waist(1) newmatrix(2,5)-waist(2)];
rightleg=[newmatrix(11,4)-waist(1) newmatrix(11,5)-waist(2)];

ll=dist(u,leftleg');
ul=dist(u,rightleg');

legangle(j)=(acos((leftleg*rightleg')/(ll*ul)))*180/pi;

%Records data in legframe matrix
legframe(framecount,k)=legframe(framecount,k)+legangle(j);

upperleg=[newmatrix(22,4)-newmatrix(10,4) newmatrix(22,5)-
newmatrix(10,5)];
lowerleg=[newmatrix(11,4)-newmatrix(10,4) newmatrix(11,5)-
newmatrix(10,5)];

uleg=dist(u, upperleg');
lleg=dist(u, lowerleg');

kneeangle(j)=180-(acos((upperleg*lowerleg')/(uleg*lleg)))*180/pi;

kneeframe(framecount,k)=kneeframe(framecount,k)+kneeangle(j);

footsep(framecount,k)=footsep(framecount,k)+abs(newlength);
```

```
headbounce(framecount,k)=headbounce(framecount,k)+newmatrix(28,4);

wristsep(framecount,k)=wristsep(framecount,k)+abs(newmatrix(8,4)-
newmatrix(16,4));

gaitframe(k)=gaitframe(k)+1;

framecount=framecount+1;

%prod(j) determines if feet have passed each other, if they have then
one
%value will be positive the other negative and prod(j) will be negative

prod(j)=length*newlength;


if (prod(j)<0)
    %When prod(j) is negative feet have passed each other, every second
    %time this happens, a gait cycle has been completed
    if p==2
        framecount=1;
        k=k+1;
        p=0;
    end

    p=p+1;

    %Gait cycle parameters being recorded
    stats(k)=newmatrix(1,1);
    times(k)=newmatrix(1,2);

    headwaist(k)=newmatrix(30,5)-newmatrix(22,5);

    rightshoulder=[newmatrix(12,4) newmatrix(12,5)];
    rightelbow=[newmatrix(15,4) newmatrix(15,5)];
    rightwrista=[newmatrix(16,4) newmatrix(16,5)];
    rightfrontwaist=[newmatrix(22,4) newmatrix(22,5)];
    rightknee=[newmatrix(10,4) newmatrix(10,5)];
    rightheel=[newmatrix(11,4) newmatrix(11,5)];

    rupperarm(k)=dist(rightshoulder, rightelbow');
    rlowerarm(k)=dist(rightwrista, rightelbow');
    rupperleg(k)=dist(rightfrontwaist, rightknee');
    rlowerleg(k)=dist(rightknee, rightheel');

    rightswing=dist(rightwrista, rightfrontwaist');


    stridelengths(k)=stridelength;
    stridelength=0;
```

```
end

length=newlength;

end


f=floor(size(stats)/2);


%Stats are paired to account for two foot crossings being equal to one
%cycle
for s=1:f(2)
    pairedstats(s)=stats(2*s-1);
    pairedtimes(s)=times(2*s-1);
end

cycleframes=diff(pairedstats);
cycletimes=diff(pairedtimes);


%avframes converts matrix of gait cycles into one average vector for
the
%parameter
avlegframe=avframes(legframe,gaitframe,k);
avkneeframe=avframes(kneeframe,gaitframe,k);
avfootsep=avframes(footsep,gaitframe,k);
avheadbounce=avframes(headbounce,gaitframe,k);
avwristsep=avframes(wristsep,gaitframe,k);


disp('The identification # of the subject is:')
disp(ident)
output(1)=ident;
variance(1)=ident;

disp('The mean time per full cycles:')
disp(mean(cycletimes))
output(2)=mean(cycletimes);
variance(2)=var(cycletimes);

disp('The mean # of frames per full cycles:')
disp(mean(cycleframes))
output(3)=mean(cycleframes);
variance(3)=var(cycleframes);

disp('The average stridelength:')
disp(mean(stridelengths))
output(4)=mean(stridelengths);
variance(4)=var(stridelengths);

disp('The maximum stridelength:')
disp(max(stridelengths))
output(5)=max(stridelengths);
```

```
disp('The average height in cm:')
disp(mean(height)/10)
output(6)=(mean(height))/10;
variance(6)=var(height)

disp('The average head/waist distance:')
disp(mean(headwaist))
output(7)=mean(headwaist);
variance(7)=var(headwaist);

disp('The average upper arm length:')
disp(mean(rupperarm))
output(11)=mean(rupperarm);
variance(11)=var(rupperarm);

disp('The average lower arm length:')
disp(mean(rlowerarm))
output(12)=mean(rlowerarm);
variance(12)=var(rlowerarm);

disp('The average upper leg length:')
disp(mean(rupperleg))
output(13)=mean(rupperleg);
variance(13)=var(rupperleg);

disp('The average lower leg length:')
disp(mean(rlowerleg))
output(14)=mean(rlowerleg);
variance(14)=var(rlowerleg);

disp('The average distance of wrist from waist:')
disp(mean(rightswing))
output(15)=mean(rightswing);
variance(15)=var(rightswing);

disp('The average angle between right leg and left leg:')
disp(mean(legangle))
output(16)=mean(legangle);
variance(16)=var(legangle);

disp('The average knee angle:')
disp(mean(kneeangle))
output(17)=mean(kneeangle);
variance(17)=var(kneeangle)

disp('The average speed (km/hr):')
disp(((sum(stridelengths)/1000000)/(max(times)/7200)))
output(18)=(sum(stridelengths)/1000000)/(max(times)/7200);

%Uses fillincurve to extend the length of all average parameter vectors
to
%a length of 100 elements then records dynamic data in x matrix
```

```
avlegframe=fillincurve(avlegframe,105);
x=zeros(1,100,3)
x(1,:,1)=avlegframe;

avkneeframe=fillincurve(avkneeframe,105);
x(1,:,2)=avkneeframe;

avfootsep=fillincurve(avfootsep,105);
x(1,:,3)=avfootsep;

avheadbounce=fillincurve(avheadbounce,105);
x(1,:,4)=avheadbounce;

avwristsep=fillincurve(avwristsep,105);
x(1,:,5)=avwristsep;


x(1,:,6)=x(1,:,3)./x(1,:,5);
end
```

Loading Text- Loads a text file of data for a single person at a single speed. Outputs data in a matrix with d equal to number of frames.

```
function[matrix,d]=loadtext5(string)
%Derrick Chelliah
%Command Line:  x=loadtext5('SandyWalk1_0.emf')

matrix=[];

fid = fopen(string, 'r');  %Opens file to be read

for i=1:3
    tline=fgetl(fid);   %Skips three lines description
end

q=fscanf(fid, '%c %c %c %c %c %c %c %c', [8 1]); %Skips 8 characters
d= fscanf(fid, '%g', [1 1])  % Reads number of frames

for i=1:16
    tline=fgetl(fid);       %Skips 16 lines
end


for i=1:d

z= fscanf(fid, '%s', [1 1]);    %Skips one string
r= fscanf(fid, '%g %g', [2 1]);  %Reads frame # and time in seconds

a = fscanf(fid, '%g %g %g %g', [4 30]); %Reads 4x30 matrix of
coordinates
a = a';

column=ones(30,1);

matrix=[matrix;r(1)*column r(2)*column a];  %Appends 6x30 matrix

end

fclose(fid);    %Closes file

end
```

Average Frames-Avframes takes as the first value a matrix that consists of some number of gait measurements with each row representing a different gait cycle.  gaitframe is a vector that lists the number of frames in each gait. k refers to the number of gaits. avframes takes several gait cycles, extends them to the same number of frames and then averages them to create one average gait cycle

```
function [avframe]=avframes(matrix,gaitframe,k)
% Derrick Chelliah 28 Feb 2008

maxframe=max(gaitframe); %Finds maximum number of frames of all gaits
newframes=[];
newlabel=[];

for c=1:(k-1)     %This creates matrix newframes where all gaits have
the same number of frames
    for t=1:gaitframe(c)
        newlabel(t)=round(t*maxframe/gaitframe(c));
        newframes(newlabel(t),c)=matrix(t,c);
    end
end

rowsum=0;
rowcount=0;

for t=1:maxframe   %Averages all gaits together
    rowsum=sum(newframes(t,:));

    for c=1:(k-1)
        if newframes(t,c)>0
            rowcount=rowcount+1;
        end
    end

    %Divides by number of rows summed to create an average
    avframe(t)=rowsum/rowcount;

    rowcount=0;
end
end
```

Linear Interpolation-This function takes a vector, removes all the zeros and then stretches that vector to 100 elements in length through linear interpolation. It is necessary because the avframes function creates average vectors for various gait statistics that all have different lengths. The fillincurve function standardizes all the lengths to "maxframe"

```
function [curve]=fillincurve(vector,maxframe)
%[curve]=fillincurve([0 0 0 0 0 0 0 0 1 20 4])
%Derrick Chelliah 28 Feb 2008


newlabel=[];
newvector=[];

framenum=length(vector);


%Initially removes all zeros from vector
for t=1:framenum
    if vector(framenum+1-t)==0
        vector(framenum+1-t)=[];
    end
end

framenum=length(vector);

%Creates longer vector with original values equally spaced
for t=1:framenum
        newlabel(t)=round(t*maxframe/framenum);
        newvector(newlabel(t))=vector(t);
end

k=length(newlabel);


%Fills in all zeros by interpolating between every two consecutive
points
for j=1:(k-1)

firstframe=newlabel(j);
nextframe=newlabel(j+1);
interveningframe=newlabel(j+1)-newlabel(j)-1;
slope=(newvector(newlabel(j+1))-newvector(newlabel(j)))/(newlabel(j+1)-
newlabel(j));

for i=1:interveningframe
    newvector(newlabel(j)+i)=newvector(newlabel(j))+i*slope;
end

end
```

```
%Removes first five elements which often remain zero

newvector(1)=[];
newvector(1)=[];
newvector(1)=[];
newvector(1)=[];
newvector(1)=[];

curve=newvector;
```

Overarching Calling Function-This function merely sets up and calls one of the four human gait recognition functions, either MED, Mahalanobis, Quadratic Discriminant or LMED by altering line 23. The file dynamicvar.mat contains the variable outputcombined which contains statistics generate by the inputread3 file

```
function [accuracy]=callingfunction(values)
%Derrick Chelliah 28 Feb 08

load('dynamicvar.mat');

[n,m]=size(values);

newoutput=[];


for i=1:m

    newoutput=[newoutput outputcombined(:,:,values(i))];

end

label=[1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2  3 3 3 3 3 3 3 3 3  4 4 4 4
4 ...
    4 4 4 4  5 5 5 5 5 5 5 5 5]';

[accuracy]=LMED(newoutput,label,5,newoutput)

end
```

Minimum Euclidean Distance-Function takes as input a matrix of parameters and uses the Lachenbruch hold-out procedure to find the Minimum Euclidean Distance between each gait and the average of the gaits in each test class in order to assign an identity to each gait

```
function [totalaccuracy]=med(x,label,classnum,y)
%Derrick Chelliah 28 Feb 2008



[nc,n]=size(x);

meddis=zeros(nc,1);     %Assigns initial MED distance for all lines as
1000
%assign=1000*ones(nc,1);
for k=1:nc
    assign(k,1)=10000000;
    assign(k,2)=k;
end

original=x;
labeloriginal=label;

for z=1:nc

x(z,:)=0;
label(z)=0;

means=zeros(classnum,n);
length=zeros(classnum,1);

for i=1:classnum    %Calculates means for each class as well as # per
class
for k=1:nc
    if label(k)==i
        means(i,:)=means(i,:)+x(k,:);
        length(i)=length(i)+1;
    end
end
end




for i=1:classnum
    means(i,:)=means(i,:)./length(i);
end






%for i=1:nc      %Calculates med distance between each line and class
mean and assigns lowest med and associated class for each line
    for j=1:classnum
```

```
        meddis(z)=sqrt(sum((y(z,:)-means(j,:)).^2));
        if meddis(z)<assign(z,1)
            assign(z,1)=meddis(z);
            assign(z,3)=j;
        end
    end


x=original;
label=labeloriginal;
end


accuracy=zeros(classnum,1);     %Calculates # of lines appropriately
labelled

for i=1:nc
    if assign(i,3)==label(i)
        accuracy(label(i))=accuracy(label(i))+1;
    end
end


for i=1:classnum        %Calculates percent accuracy per class
    accuracy(i)=accuracy(i)/9;
end

totalaccuracy=1-sum(accuracy)/classnum;

conting=zeros(6,6);

for i=1:nc
        conting(label(i),assign(i,3))=conting(label(i),assign(i,3))+1;
end

for i=1:5
    conting(i,6)=sum(conting(i,:));
    conting(6,i)=sum(conting(:,i));
end

conting(6,6)=sum(diag(conting));

conting
```

```
Minimum Mahalanobis Distance-%Function takes as input a matrix of
parameters and uses the Lachenbruch hold-out procedure to find the
Mahalnobis distance between each gait and the average of the gaits in
each test class in order to assign an identity to each gait

function [totalaccuracy]=Mahalanobis(x,label,classnum,y)
%Derrick Chelliah 28 Feb 2008

[nc,n]=size(x);

meddis=zeros(nc,1);     %Assigns initial med distance for all lines as
1000

for k=1:nc
    assign(k,1)=10000000;
    assign(k,2)=k;
    assign(k,3)=0;
end

original=x;
labeloriginal=label;

for z=1:nc

x(z,:)=0;
label(z)=0;

%Calculates means and variance for each class
for i=1:classnum
    indx=find(label==i);
    means(i,:)=mean(x(indx,:));
    varmean(i,:)=var(x(indx,:),0,1);
    [a,b]=size(x(indx,:));
    length(i)=a;
end

%Calculates Mahalanobis distance
   for j=1:classnum
        meddis(z)=sqrt(sum((y(z,:)-means(j,:))./varmean(j,:))^2);
        if meddis(z)<assign(z,1)
            assign(z,1)=meddis(z);
            assign(z,3)=j;
        end
    end

x=original;
label=labeloriginal;
end

accuracy=zeros(classnum,1);     %Calculates # of lines appropriately
labelled

for i=1:nc
    if assign(i,3)==label(i)
        accuracy(label(i))=accuracy(label(i))+1;
```

```
    end
end


for i=1:classnum        %Calculates percent accuracy per class
    accuracy(i)=accuracy(i)/9;
end

totalaccuracy=1-sum(accuracy)/classnum;

conting=zeros(6,6);

for i=1:nc
        conting(label(i),assign(i,3))=conting(label(i),assign(i,3))+1;
end

for i=1:5
    conting(i,6)=sum(conting(i,:));
    conting(6,i)=sum(conting(:,i));
end

conting(6,6)=sum(diag(conting));

conting
```

Quadratic Discriminant Function-Function takes as input a matrix of parameters and uses the Lachenbruch hold-out procedure to find the Quadratic Discriminant Function distance between each gait and the average of the gaits in each test class in order to assign an identity to each gait

```
function [accur]=quadtest(output,label,classnum,sample)
%Derrick Chelliah 28 Feb 2008


[n,m]=size(output);
q=0;

%Converts static parameters to one column if converted to 100 columns

if (output(:,1)==output(:,m))
    output(:,1)'
    output(:,m)'
    output=output(:,1);
    q=5;
end

x=output;

[nc,n]=size(x);

origx=x;
origsample=sample;
origlabel=label;
assign=ones(nc,1)*-10^1000000;

for i=1:nc

x=origx;
label=origlabel;
x(i,:)=0;
label(i)=0;

%Calculates means for classes
for z=1:5
    indx=find(label==z);
    x_mean(z,:)=mean(x(indx,:));
     [a,b]=size(x(indx,:));
    length(z)=a;
end

%Calculates covariances for each class
indx=find(label==1);
 covariance(:,:,1)=cov(x(indx,:));

 indx=find(label==2);
 covariance(:,:,2)=cov(x(indx,:));
```

```
 indx=find(label==3);
 covariance(:,:,3)=cov(x(indx,:));

 indx=find(label==4);
 covariance(:,:,4)=cov(x(indx,:));

 indx=find(label==5);
 covariance(:,:,5)=cov(x(indx,:));

 %Calculates discriminant function scores for each gait
 for j=1:5
        p(i,j)=0.5*log(det(inv(covariance(:,:,j))))-0.5*(sample(i,:)...
            -x_mean(j,:))*inv(covariance(:,:,j))*(sample(i,:)-
x_mean(j,:))';

        %Assigns label based on lowest discriminant score
        if p(i,j)>assign(i,1)
            assign(i,1)=p(i,j);
            assign(i,2)=j;
        end
    end
end

label=origlabel;

conting=zeros(6,6);

%Calculates contingency matrix on the basis
for i=1:nc
        conting(label(i),assign(i,2))=conting(label(i),assign(i,2))+1;
end

for i=1:5
    conting(i,6)=sum(conting(i,:));
    conting(6,i)=sum(conting(:,i));
end

conting(6,6)=sum(diag(conting));

x_mean;

conting

accur=1-(conting(6,6)/45)
```

Brand and Bound 1 algorithm-This function starts with parameter listed in initialvalues and runs them algorithm referred to by calling function using them. It then attempts to remove parameters one at a time to see if this improves classification accuracy. This process is repeated until elimination of parameters no longer improves classification accuracy.

```
function [newaccuracy, initialvalues]=variablefunctionmed
%Derrick Chelliah 28 Feb 2008

initialvalues=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20];

newaccuracy=callingfunction(initialvalues);
initialaccuracy=1;


while newaccuracy<initialaccuracy
    initialaccuracy=newaccuracy;
    for i=1:length(initialvalues)
        values=initialvalues;
        values(i)=[];
        accuracy(i)=callingfunction(values);
    end

    [newaccuracy variable]=min(accuracy);

    if newaccuracy<initialaccuracy
    initialvalues(variable)=[];
    end
end


end
```

Branch and Bound 2 algorithm-This function starts with parameters 1 through m and incrementally tests them individually with the algorithm currently referred to by callingfunction.  The most effective classifying parameter is kept and then the process is repeated to choose the second parameter.  This continues until addition of more parameters does not improve classification accuracy

```
function [curraccur, values]=variablefunctionmed2
%Derrick Chelliah 28 Feb 2008

accurval=[];

m=14

for i=1:m
    values=i;
    accurval(i)=callingfunction(values);
end


   [c,k]=min(accurval);


values=k;
curraccur=c;


improvement=1;

while improvement>0

    improvement=0;

    for j=1:m
    tempvalues=[values j];
    tempval(j)=callingfunction(tempvalues);
    if tempval(j)<curraccur;
        values=tempvalues;
        improvement=curraccur-tempval(j);
        curraccur=tempval(j);

    end
    end
end
end
```

Principal Component Correlation-Calculates principal components for whatever data matrix is entered into line 16.  Also plots first three components against each other

```
function [y,z,loadings]=princompcorrel
%Derrick Chelliah 28 Feb 2008

load('dynamicvar.mat');

graphoutput(46,:)=0;
graphoutput(46,:)=[];


outputcombined(46,:)=0;
outputcombined(46,:)=[];

x=outputcombined

origx=x;

meanx=mean(x);

[n,m]=size(x);

meanx=ones(n,1)*meanx;

x=x-meanx;

correl=corrcoef(x);

display('PCA output:')
[v,d]=eigs(correl);


display('Loadings matrix calculated:')

D=-(correl-triu(correl)-tril(correl));

loadings=(D^-0.5)*v*(d^0.5)

z=diag(d)'
z=z./sum(z)


display('Component scores calculated:')

y=x*v;

component1=y(:,1);
component2=y(:,2);
```

```
component3=y(:,3);
component4=y(:,4);


figure

scatter(component1(1:9),component2(1:9),'b+')
hold
scatter(component1(10:18),component2(10:18),'kd')
scatter(component1(19:27),component2(19:27),'m>')
scatter(component1(28:36),component2(28:36),'gp')
scatter(component1(37:45),component2(37:45),'cs')
legend('Subject 1','Subject 2','Subject 3','Subject 4','Subject 5');
xlabel('Component 1')
ylabel('Component 2')

figure

scatter(component1(1:9),component3(1:9),'b+')
hold
scatter(component1(10:18),component3(10:18),'kd')
scatter(component1(19:27),component3(19:27),'m>')
scatter(component1(28:36),component3(28:36),'gp')
scatter(component1(37:45),component3(37:45),'cs')
legend('Subject 1','Subject 2','Subject 3','Subject 4','Subject 5');
xlabel('Component 1')
ylabel('Component 3')


figure

scatter(component2(1:9),component3(1:9),'b+')
hold
scatter(component2(10:18),component3(10:18),'kd')
scatter(component2(19:27),component3(19:27),'m>')
scatter(component2(28:36),component3(28:36),'gp')
scatter(component2(37:45),component3(37:45),'cs')
legend('Subject 1','Subject 2','Subject 3','Subject 4','Subject 5');
xlabel('Component 2')
ylabel('Component 3')

end
```

## Bibliography


Begg, R., and Kamruzzaman, J. (2005). A machine learning approach for automated recognition of movement patterns using basic, kinetic and kinematic gait data. *Journal of Biomechanics, 38*(3), 401-408.

Bobick, A. F., and Johnson, A. Y. (2001). A multi-view method for gait recognition using static body parameters. *Electrical and Computer Engineering*, 301-311.

Chen, H., Chen, H., Chen, Y., and Lee, S. (2006). Human action recognition using star skeleton. *4th ACM International Workshop on Video Surveillance and Sensor Networks, VSSN'06, Co-Located with the 2006 ACM International Multimedia Conference,* 171-178.

Fujiyoshi, H., Lipton, A. J., and Kanade, T. (2004). Real-time human motion analysis by image skeletonization. *IEICE Transactions on Information and Systems, E87-D*(1), 113-120.

Hayfron-Acquah, J. B., Nixon, M. S., and Carter, J. N. (2002). Human identification by spatio-temporal symmetry. *Proceedings - International Conference on Pattern Recognition, 16*(1), 632-635.

Kale, A., Sundaresan, A., Rajagopalan, A. N., Cuntoor, N. P., Roy-Chowdhury, A. K., Krüger, V., et al. (2004). Identification of humans using gait. *IEEE Transactions on Image Processing, 13*(9), 1163-1173.

Krzanowski, W. J. (1988). *Principles of multivariate analysis*. New York: Oxford University Press.

Kshirsagar, A. M. (1972). *Multivariate analysis*. New York: Marcel Dekker, Inc.

Liu, Z., and Sarkar, S. (2005). Effect of silhouette quality on hard problems in gait recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 35*(2), 170-183.

Lu, H., Plataniotis, K. N., and Venetsanopoulos, A. N. (2006). A layered deformable model for gait analysis. *FGR 2006: 7th International Conference on Automatic Face and Gesture Recognition, 2006* 249-256.

Ning, H., Wang, L., Hu, W., and Tan, T. (2002). Model-based tracking of human walking in monocular image sequences. *2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering, 1,* 537-540.

Nixon, M. S., Carter, J. N., Nash, J. M., Huang, P. S., Cunado, D., and Stevenage, S. V. (2006). *Automatic gait recognition*. Southampton.

Samler, J. J. (2006). *Statistical approach to background subtraction for production of high-quality silhouettes.* Air Force Institute of Technology (AU), Wright Patterson AFB OH, (ADA456804)(122068637).

Tingley, M., Wilson, C., Biden, E., and Knight, W. R. (2002). An index to quantify normality of gait in young children. *Gait & Posture, 16*(2), 149-158.

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

Yu, S., Wang, L., Hu, W., and Tan, T. (2004). Gait analysis for human identification in frequency domain. *Proceedings - Third International Conference on Image and Graphics*, 282-285.

Zhang, R., Vogler, C., and Metaxas, D. (2007). Human gait recognition at sagittal plane. *Image and Vision Computing, 25*(3), 321-330.

## Vita

Captain Derrick M. Chelliah was born in Lower Merion, Pennsylvania. He graduated from Mount Olive High School, Flanders, New Jersey in May 1997. From there he went to Tempe, Arizona to attend Arizona State University where he graduated with a Bachelor of Arts in Mathematics in May 2001.

Captain Chelliah entered active duty in August 2003 when he attended the Air Force Officer Training School where he received his Commission in October 2003. His first assignment was to Hanscom Air Force Base, Air Force Materiel Command, Air Force Research Labs, Sensors where he served as the Radar Research Analyst. Upon completion of this assignment Capt Chelliah began his graduate studies at the Air Force Institute of Technology (AFIT), pursuing a Master of Science in Operations Research.

Upon completing AFIT, Capt Chelliah was assigned to Tyndall Air Force Base, Air Combat Command, 83rd Fighter Weapons Squadron.

| REPORT DOCUMENTATION PAGE | | | | *Form Approved* *OMB No. 074-0188* |
|---|---|---|---|---|

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

**PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

| **1. REPORT DATE** *(DD-MM-YYYY)* | **2. REPORT TYPE** | **3. DATES COVERED** *(From – To)* |
|---|---|---|
| 03-24-2008 | **Master's Thesis** | Jun 2008 – Mar 2008 |

| | |
|---|---|
| **4. TITLE AND SUBTITLE** STATISTICAL APPROACH TO THE CHARACTERIZATION AND RECOGNITION OF HUMAN GAITS | **5a. CONTRACT NUMBER** |
| | **5b. GRANT NUMBER** |
| | **5c. PROGRAM ELEMENT NUMBER** |
| **6. AUTHOR(S)** Chelliah, Derrick, M., Captain, USAF | **5d. PROJECT NUMBER** |
| | **5e. TASK NUMBER** |
| | **5f. WORK UNIT NUMBER** |
| **7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)** Air Force Institute of Technology<br>Graduate School of Engineering and Management (AFIT/EN)<br>2950 Hobson Street, Building 642<br>WPAFB OH 45433-7765 | **8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GOR/ENC/08-02 |
| **9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)** Intentionally left blank | **10. SPONSOR/MONITOR'S ACRONYM(S)** |
| | **11. SPONSOR/MONITOR'S REPORT NUMBER(S)** |

**12. DISTRIBUTION/AVAILABILITY STATEMENT**

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**

Human gait recognition is a newly emergent biometric technique that offers the capability of identifying people at a distance. As opposed to other forms of biometric identification, gait recognition has the advantage of being unobtrusive and not requiring the cooperation of the subject. This research develops a variety of mathematical algorithms that seek to provide a practical solution to the problem of human gait recognition. Here, a database of recorded gaits is used with algorithms that initially build gait signatures to characterize the distinct walking features of known subjects. Having constructed the gait signatures, techniques employing Quadratic Discriminant functions, the Mahalanobis distance, and the Euclidean distance are utilized in order to identify unknown gaits through their mathematical relationships to existing gait signatures. The performances of the developed algorithms on the sample database are compared with each other to determine the advantages and disadvantages of the methodologies for achieving human gait recognition.

**15. SUBJECT TERMS**

Human Gait Recognition, Human Gait, Biometric, Biometric Recognition

| **16. SECURITY CLASSIFICATION OF:** | | | **17. LIMITATION OF ABSTRACT** | **18. NUMBER OF PAGES** | **19a. NAME OF RESPONSIBLE PERSON** Samuel A. Wright, Capt, USAF (ENC) |
|---|---|---|---|---|---|
| **a. REPORT** | **b. ABSTRACT** | **c. THIS PAGE** | | | **19b. TELEPHONE NUMBER** *(Include area code)* 937-255-3636 x4549; e-mail: samuel.wright@afit.edu |
| **U** | **U** | **U** | **UU** | 138 | |

**Standard Form 298 (Rev. 8-98)**
Prescribed by ANSI Std. Z39-18