Computer Vision-Based Analysis of Human Daily Actions using Hidden Markovian Models

by

Trevor Robert John Beugeling
B.Sc., Simon Fraser University, 2002
B.Eng., University of Victoria, 2008

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Applied Science

in the Department of Electrical and Computer Engineering

Computer Vision-Based Analysis of Human Daily Actions using Hidden Markovian Models

by

Trevor Robert John Beugeling
B.Sc., Simon Fraser University, 2002
B.Eng., University of Victoria, 2008

Supervisory Committee

Dr. A. Branzan-Albu, Supervisor
(Department of Electrical and Computer Engineering)

Dr. M. Sima, Departmental Member
(Department of Electrical and Computer Engineering)

Dr. J. Weber, Outside Member
(Department of Software Engineering)

Supervisory Committee

Dr. A. Branzan-Albu, Supervisor
(Department of Electrical and Computer Engineering)

Dr. M. Sima, Departmental Member
(Department of Electrical and Computer Engineering)

Dr. J. Weber, Outside Member
(Department of Software Engineering)

ABSTRACT

The study of human motion from a medical standpoint has traditionally involved the use of marker-based motion tracking systems, as well as other sensory devices. This equipment is often expensive, has low-portability, and might even influence tracking results by distracting or otherwise inhibiting a subject's normal motion performance. In comparison, non-marker-based tracking methods are less costly, easier to move and set up, and requires no markers or other devices. However, previous work in silhouette-based human motion analysis is typically focused on the classification of activities or the identification of subjects, neither of which are much use to medical professionals.

We propose a merging of these two research fields. By applying silhouette-based motion tracking to the problem of motion performance analysis, we have developed a new method which can reliably and accurately model human motions and detect abnormalities. Our approach, which is based on Hidden Markovian Models with continuous observation probabilities, creates standardized models to represent common human motions. These models are then used as a basis for further analysis. We have extensively tested the proposed method with a custom designed database that takes into account speed related and subject related variations of motion performance.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank Dr. Alexandra Branzan-Albu, for her continued guidance as I complete my work on this research, and begin to look ahead to new challenges.

Special thanks to:

Jeremy Svendsen, Patrick Bonneau, Maria Girard-Martel, and Leona Wade,
for donating their time to aid in the creation of the database used in this study.

DEDICATION

To my family, for their many years of support and encouragement.

Chapter 1

Introduction

This introduction is intended to give an overview of the field of computer vision and its applications in the medical field.

Section 1.1 will provide a general overview of the uses of computer vision in medical research, providing some specific examples of its use in various applications, including user interface design, data visualization, and data processing. By reviewing the uses of computer vision in the medical field, we hope to illustrate its wide array of uses, the advantages of its use, and some of its limitations.

Section 1.2 will give a more focused view of the applications of computer vision, centered around the field of physiotherapy. As our research falls into this domain, it is important to understand the current state of the field and how our findings will expand and improve on previous knowledge.

1.1 Computer Vision in Medical Research

1.1.1 User Interface Design

The use of computer vision enables the development of advanced user interfaces which allow medical experts to interact with data without the need for physical devices such as a keyboard or mouse. We will begin by presenting two examples of user interfaces which can be used by a surgeon during a procedure, reducing the need to study such data ahead of time or rely on assistants to operate the interface for them.

Graetzel et al [12] designed a non-contact mouse which could be used by surgeons to control a user interface by means of simple hand gestures. This would allow the surgeon to perform simple interactions with a computer without endangering the

sterile environment of the operating room. However, it would have limited usefulness in procedures where both of the surgeon's hands are occupied, as controlling the interface requires the surgeon to move at least one hand away from the patient.

Nishikawa et al [23] propose another form of user interface which uses face-tracking to control the movements of a laparoscopic camera. This system allows the surgeon to keep his hands free for other tasks, and was found to have many advantages over previous voice-controlled methods. As the system is very sensitive to head motion, the surgeon is required to maintain a constant distance between their head and the monitor, where the face-tracking equipment is located. It was found that in some cases, this requirement to carefully control their head position and posture became a distraction when engaged in complex procedures.

1.1.2 Data Visualization

Many forms of visual data are collected by medical professionals in the course of examining and diagnosing health issues in patients. This includes technologies such as X-ray, computed tomography (CT), ultrasound, magnetic resonance imagery (MRI), confocal microscopy, and even "regular" photographic images and video. These technologies allow the human body to be viewed in a variety of ways, which trained professionals are able to use to detect and locate key features or abnormalities that are of interest. This has lead to a large amount of computer vision research being focused on the visualization of data.

Shinagawa et al [14] developed a method to create 3-dimensional models of objects of interest (i.e. bones or body organs) and match them to data obtained from medical imaging technologies such as CT or MRI by finding point correspondences between the medical images and the 3-D model.

Yang et al [35] investigate various techniques of generating virtual and augmented reality environments of patients. Their goal was to determine what techniques are required in order to build models which are as realistic and patient specific as possible.

1.1.3 Computer Vision aided Data Processing

Here we present examples of how computer vision algorithms can be used to assist professionals in the processing of data. Many of these implementations are only semi-automated, and still require a large amount of control from the operator. This is

normally due to medical standards, as the professional must confirm all results obtained by any computer algorithm. Instead of "replacing" the medical professional, these algorithms are instead designed to simplify their work, and increase productivity.

The treatment of brain tumors is an extremely sensitive process, which requires exposing the cancerous tissue to a high dose of radiation, while at the same time minimizing the exposure to the surrounding healthy tissue. Careful mapping of the brain is required to determine the exact 3-dimensional region which is to be treated. This work, which is normally performed manually by medical professionals, can be very tedious. Schlegel et al [10] propose an automated system to aid in the segmentation process. The system provides what it determines to be the most likely segmentation results, which the operator can then confirm or modify as they see fit.

Ahmed El-Bialy [9] presents a 3D virtual clinic, created to aid dentists in diagnosis and treatment of patients. The system uses data on the patient's soft and hard tissues to generate a 3-D model of the head. This model then be analyzed to detect skeletal or dental problems, aiding the dentist in diagnosing problems, and planning treatments. Once a treatment is decided upon, changes can be made to the model to simulate the outcome of the procedure.

Sen et al [34] also developed a 3-D virtual environment, in this case for the purpose of training health care professionals in the treatment of bone fractures. The environment simulates not only the bone(s) in question, but also the surgical tools and devices used to carry out treatment. Using a computer interface, operators (i.e. trainees) carry out the virtual surgical procedure, and are able to view the results.

Not all data processing algorithms require the direct involvement of a medical expert. Some are almost, if not completely, autonomous. Campbell et al [16] developed a method to track the behavior of stem cells by monitoring mitoses and cellular divisions. Their method employs feature extraction techniques, and training of a classifier to recognize the various behaviors of the cells.

Using computer vision for classification of behavior is not limited to microscopic cells. It can also be used to observe motion and behavior of the full human body. In the next section we explore this aspect of computer vision analysis in medicine.

1.2 Physiotherapy and Human Motion Analysis

According to Health Canada projections, by 2031 elderly persons will make up roughly 23% of the population [6]. Such an increase represents a possible doubling of the elderly population over a 25 year period starting in 2005. This will almost certainly be accompanied by an increase in age-related injuries and other associated problems. With these issues in mind, a great deal of research is now focused on health issues involving elderly people.

One such area of research is the development of *aging-in-place strategies*, which aim to preserve the health and quality of life of the elderly. Mihailidis et al [20] have developed computer vision-based assistive technologies useful in monitoring the activities of daily living.

Traditionally, this area of research has focused on the classification of activities, or identification of the subjects performing them. Such research is focused mainly in the context of security and surveillance which, according to Moeslund et al [21, 22], represents the majority of research in the area of human motion analysis using video sequences.

These research areas are often validated using ground truth data produced using basic human reasoning. Cutting and Kozlowski [8] have shown that typical humans are easily able to recognize human activities or identify familiar people from their gait. However, when attempting to evaluate the quality of human motion, even for basic daily activities, an accurate analysis can only be performed by a professional such as a physiotherapist or kinesiologist. The forms of analysis typically used involve numeric scores such as the Berg Balance Scale (BBS) [4], which consists of a 5 point ordinal scale for 14 basic activities. Such scoring methods cannot be used for validation of a vision-based technique.

One example of well-studied incidental abnormal human motion is falling. Research has been conducted by many groups regarding the visual monitoring of falls. Cuchiarra et al [7] developed a system which detects falls by tracking abrupt changes in posture. Another system proposed by Mckenna and Carif [19] learns normal human activity patterns and uses that information to detect falls. While fall detectors are an extremely useful tool for safety monitoring, they are limited to merely detecting falls, offering no aid in the prevention of such occurrences.

This has lead to the design of prevention-oriented visual motion analysis framework which will be able to predict, as well as detect, fall events. The relevance in

this direction of research is supported by medical studies [32, 37], which demonstrate that falls are preventable if gait and balance abnormalities are detected early and properly treated. In the area of vision-based abnormal gait analysis, Bauchage et al [3] developed a method for detecting uncoordinated gait. Their approach is able to detect a quasi-random gait caused by dizziness, but cannot easily be generalized to detect and analyze other types of motion abnormalities. A technique for abnormal gait classification using flow histograms and eigenspace transformation is proposed by Wang [39]. This approach works well for classifying *a priori* defined gait abnormalities. However, it is heavily oriented towards supervised classification, since the analysis of gait changes (i.e. how different a walking style is from normal) relies only on empirical distance thresholding in the feature space.

Chapter 2

Related Work

Computer vision-based human motion analysis is a broad field of research, covering many applications. In this section, we will give an overview of several prominent areas of study. First we will discuss the distinction between silhouette data and feature points data

2.1 Human motion analysis: Silhouette vs feature points data

Analysis of silhouette data focuses on the study of whole-body motion, without decomposing it into motions of individual body parts. It is particularly suitable for the analysis of "stationary" activities such as sit-to-stand where anatomical joint positions and inverse kinematics are difficult to retrieve due to loosing clothing, self-occlusion, and body shape variability.

In contrast, feature points data is obtained by tracking body parts throughout a video sequence. This type of research compliments silhouette-based approaches by retrieving information about the relative motion of body parts. Feature-based analysis is of particular use when studying locomotion activities, where the motion of anatomical joints/body parts is easily retrievable from spatiotemporal cues.

Feature points data analysis is applicable to both marker- and non-marker-based motion tracking, examples of which can be found in the following two sections. Although silhouette-based analysis is normally applied to non-marker-based motion tracking, its use in marker-based tracking analysis is not out of the question. Normally, the use of a silhouette with marker-based tracking takes the form of an "avatar"

used to give a more realistic representation of the subject in a virtual environment, as opposed to separate dots (i.e. markers) floating in space. This avatar, though useful for visualization purposes, is not normally used for any form analysis.

Our research uses a silhouette-based analysis, focusing on the development of standardized models to represent "normal" motion execution. These models will then be used as a baseline for detection and analysis of irregularities. Such irregularities include sway, temporary loss of balance, motion asymmetry, variations in speed magnitude, temporary stops in motion, and limited range of motion.

2.2 Marker-based Analysis

Marker-based tracking is a popular form of human motion analysis due to its high degree of accuracy and ability to track motions ranging from very subtle to highly complex. Marker-based tracking systems, such as those developed by Vicon, require subjects to wear numerous markers at key points on the body (i.e. joints and other points of interest) which are then located and tracked using multi-camera systems. The advantages of such systems are that they are able to pin-point marker locations to within millimeters, and can do so with a very high frame rate (120 frames per second at full 16 megapixel resolution, and up to 2000 frames per second at lower resolutions[1]). Marker-based tracking is very common in medical research where precise measurements are desirable, but it is also widely used for other applications such as generating realistic action sequences for virtual environments such as sports games or computer-generated image (CGI) characters in movies. Some notable examples include the animation of CGI characters in movies such as *Lord of the Rings* (Gollum) and numerous characters in the recent film *Avatar*.

Wang et al [42] employ a Vicon motion capture system combined with force plates to model the kinetic and kinematic parameters in the foot during normal gait. By placing infrared reflective markers on the foot and having the subject walk on force plates, they are able to not only model the articulation of the foot segments, but also measure the forces exerted on the foot by the floor surface. Jain et al [1] also conduct gait related studies using a Vicon system, focusing their efforts on the ankle region of the foot.

Studies performed on human upper limb joints by Yang et al [43] investigate joint

[1]Figures obtained from http://www.vicon.com/products/t160.html

Figure 2.1: Andy Serkis (left) wearing a motion capture suit with markers. Motion data is used to control the computer generated character of Gollum from the movie The *Two Towers*. Image taken from:
*http : //mayersononanimation.blogspot.com/*2007_06_01_*archive.html*

angles for the arm and shoulder during a "pointing" action in order to model typical motion patterns. Upper body kinematics is also the focus of study for Perry et al [28], who analyzed the motion of subjects in wheelchairs as they propelled themselves forward.

Cho et al [25] make use of marker-based tracking (Vicon 370 MA with 6 infrared cameras) and force plates to monitor the movements of normally developed children and children suffering from spastic cerebral palsy (CP) in order to compare the quality of motions performed. The motion performed by the subjects is that of sit-to-stand, one of the 14 actions performed in the Berg Balance Scale method for motion assessment. This action was chosen because it is a commonly performed activity which is often difficult for children with CP to execute. Motion abnormalities can be measured in CP patients and used as a baseline for future comparisons as they receive therapy for their condition. By analyzing the movements and joint angles of normal subjects, they were able to categorize the motion into 5 phases, which were then used as a basis of comparison with children suffering from CP. This work has very close similarities to our research, in that it aims to study and quantify the quality of performance of given actions, and does so with similar target subject groups. The main difference in this approach is that it is a marker-based approach, while our work attempts to use non-marker-based tracking to achieve the same goal.

When looking toward non-medical applications, there are still many areas of research in which the use of marker-based tracking is widespread. In the field of robotics, the actions of synthetic body parts (i.e. humanoid robotic limbs) can be controlled remotely through sensors attached to a human's corresponding body part. As the

human controller moves, the markers are tracked in real time, and those readings are sent to the robotic counterpart, which then performs the same movements. Some systems, such as that proposed by Billard et al [5], make use of both marker and non-marker cues in order to determine the action to be performed by the robot. In their research, a humanoid robot with a 2-camera visual system "observes" a human controller performing an action, and then attempts to imitate that action as accurately as possible. This action consists of either moving three colored blocks on a surface in front of the robot, or "tracing" a letter of the alphabet in the air with their hand. Sensors on the arms & shoulders of the human controller allows the system to measure the distance from the shoulder to an object of interest (colored block) being moved by the controller. This measurement allows the system to decide which arm is being used to use the block, so that an accurate imitation can be performed. When moving colored blocks, the blocks were also tracked using the stereo vision system, in addition to the sensor tracking of the controller's arm, in order to identify which block was to be moved, and the type of movement to be performed.

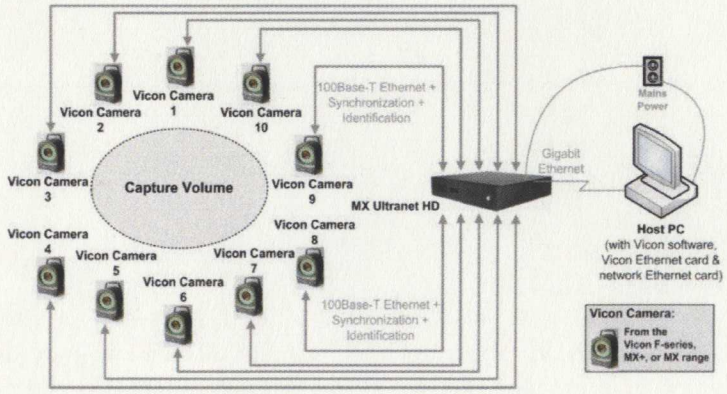


Figure 2.2: System diagram for a 10-camera Vicon motion tracking system. Image obtained from: *http* : *//www.psy.mq.edu.au/me2/index.php/about/*

2.3 Non-marker-based Analysis

Many avenues of research are focused around the development of marker-less analysis techniques. Such analysis techniques are required in situations where the use of markers is either impractical or impossible. For example, monitoring subjects in uncontrolled scenarios such as public places, or tracking body part motion when

portions of the body are occluded by clothing (i.e. long skirts) rendering the use of markers on those body parts useless.

Wactlar et al [11] present a marker-less method for analyzing dining activities of elderly subjects in a nursing home. Their approach combines face detection and motion tracking algorithms to model the movements of a subject's hands relative to their head. This motion data is then used as input to a 4-state HMM which classifies the action being performed as eating events or non-eating events. Hands move towards the head can be taken as the start of an eating event, while hands moving away from the head can signal the end of an eating event. Periods of non-motion or unspecific motion are classified as pauses in eating activities. The intended goal of such analysis is to aid caregivers in gauging the activity levels of their patients.

Wilson and Bobick [40] proposed a method of recognizing hand gestures under variation. Their approach is based on parametric Hidden Markov Models (pHMMs), which allow for explicit modeling of the variation of the gesture. The recognition of hand gestures determines where in the modeled space of variation the input lies. The parameter in pHMM encodes the spatial variability of the hand gesture. For instance, for size-showing gestures, the parameter is scalar-valued; for direction-showing gestures, the parameter must have two dimensions. Our research aims at generalizing pHMMs for whole-body atomic actions (i.e. sit-to-stand, stand-to-sit, bending, reaching, etc.) instead of hand gestures. This generalization process is not trivial, since the degrees of freedom of postural variability are far more numerous than those of simple hand gestures. Moreover, the analysis of variability of whole-body actions is relevant for a diversity of health applications (i.e. rehabilitation, sports, etc.).

Veeraraghavan et al [38] developed system to compare two sequences of deformable shapes (specifically, human silhouettes during walking activities) using parametric and non-parametric methods. In the parametric approach, they employ autoregressive modeling and moving averages to measure shape deformations. Their non-parametric approach uses a modified version of Dynamic Time-Warping (DTW). The goal of their system is to analyze gait activities of subjects for the purposes of subject identification. Identification of subjects using gait is a popular area of research. Chellappa et al [36] also use human silhouette data of human gait activities, creating exemplar frames and training HMMs for each subject.

Hastie et al [24] present a framework for learning and tracking cyclic human motions. The specific motion explored in their study is human gait, but rather than identifying the subject, their system attempts to track the motion and, using

Principal Component Analysis (PCA), fit it to a learned 3D model which can account for missing information in the input video (i.e. occluded body parts).

Liu and Sarkar [18] present a system for identifying subjects based on analysis of normalized gait sequences. Their method for normalizing gait sequences involves the use of pHMMs trained using a database of gait sequences from subjects that are to be recognized. Their system uses background subtraction to create binary silhouettes which are then rescaled to a standard height and horizontally aligned. Body regions are manually labeled into regions of interest (left/right leg, left/right arm, and torso). A set of exemplar frames are created for each sequence and used to estimate the parameters of a HMM for each subject. Input sequences are then tested against all models and assigned a subject identity based on which model returns the greatest likelihood of a match.

Many researchers are focused on the development of systems which can track a subject and identify multiple performed actions. Pentland et al [41] create and train HMMs for a pre-defined set of subjects and motions to be recognized. This collection of models then forms the basis of a classifier which takes an input video of a subject action, and attempts to determine both the subject performing the action, and the type of action being performed. Their system was tested using a selection of simple hand gestures as performed by different subjects. Metaxas et al [33] use a combination of Conditional Random Fields (CRF) and Maximum Entropy Markov Models (MEMM) to classify human motions. Their research focuses on the identification of whole-body motions, such as walking, running, bending, dancing, etc.

Robertson and Reid [31] propose a method for tracking subjects, and identifying individual actions. Then, by combining multiple actions in sequence and considering subject location, add context to an action sequence and create a "text commentary" which records a continuous log of the subject's actions with timestamps indicating when those actions occurred. Their method employs HMMs, optical flow (for relative displacement of subject), PCA, and motion tracking (to track subject location in a scene). By combining motion analysis with knowledge of a scene, context can be given to an action which is reflected in the text commentary. For example, a subject standing still at a bus stop can be considered to be "waiting for a bus", while a subject standing still in an alleyway can be considered to be "loitering". Testing was performed with a number of subjects in street environments to identify activities such as walking/running down a street, crossing a street, etc. Testing was also performed on a scene involving a tennis match, with the text commentary generating a summary

of all actions performed by each player (i.e. serves, motion within each player's zone, and plays on the ball).

While most research is based on the recognition of actions involving only a single subject, some researchers focus on recognizing activities which involve interactions between multiple subjects. Liu and Chua [17] devised a system which tracks motions of multiple subjects. Using a modified version of HMMs, referred to as Observation Decomposed HMMs (ODHMM), they divide larger observations of subjects into sub-observations. This is done to allow for the modeling of actions involving a changing number of subjects. Through this process, individual subjects are tracked independently of each other until they are seen to "interact", then their actions are tracked together until the interaction ceases. The system was tested using simple scenarios of people walking, such as individuals meeting up and walking together, as well as more complex interactions, such as one subject snatching another subject's belongings and fleeing as the victim gives chase.

2.4 Design Challenges

There are many databases of human motion which have been developed over the years. Some with computer animation applications in mind, such as the MOCAP data set, which is commercially available through Credo Interactive Inc. and features approximately 500 motion sequences covering a wide variety of human activities (i.e. walking, running, and slipping on a banana peel). However, although recorded from real human subjects, the data set consists only of synthetic avatars performing the actions. The activities are also not varied with the context of physiotherapy in mind; there is no set of motions commonly used for clinical testing, with the necessary variations in speed and performance quality required for our research.

The University of South Florida (USF) database [26] is a highly detailed set of sequences. Totaling around 300GB of data, the database consists of around 452 sequences taken from 74 subjects. Unfortunately, the database is made up of gait sequences only. No other actions are considered. The University of Maryland (UMD) data set [15] is another collection, again consisting only of gait sequences, recorded in an outdoor environment in order to make the sequences as realistic as possible. Sequences were recorded for 44 subjects with variations in time of recording (training and test sequences recorded on different days), clothing, gender, age, ethnicity, etc.

The Carnegie Mellon University (CMU) database [13] consists of a wider variety

of activities, including various combinations of activities (i.e. walking and running together, running with a leap, etc.), and also variation in motion speed (i.e. fast walk versus slow walk). As with the MOCAP database, sequences are not recorded with the context of physiotherapy in mind, and so do not possess any variations in performance quality of the motions.

Given the drawbacks of all available data sets that were found, the decision was made to create an original database for this research project. This database would be recorded in the Simbioses Laboratory at the University of Victoria, and would consist of human subjects performing activities similar to those used by physiotherapists in the diagnosing of motor problems. Variations of these motions would also be recorded, with the intention of simulating possible abnormalities that might occur in a clinical environment.

Chapter 3

Proposed Algorithm

Our research is focused on the development of computer vision-based human motion analysis in the context of aging-in-place and rehabilitation of elderly people. Specifically, we aim to analyze human activities in order to evaluate the quality of the motion performance. The intended application of this analysis is to develop a system which will aid physicians in detecting and measuring the severity of injuries and other physical limitations that lead to abnormalities in motion performance.

The primary goals of our work have been to:

1. Create a database of various common human motions executed in a variety of ways, including differences in speed and quality of performance.

2. Develop a method of creating a standardized representation for each motion studied, thus creating a baseline for further analysis.

3. Analyze motion sequences using the standardized representations, and attempt to detect and quantify any motion abnormalities that might be present.

The focus of our work has been mostly centered around goals 1 & 2. The third goal provides a means of testing the motion representations, in order to evaluate their usefulness and their limitations.

The creation of an original human motion database was essential, as no similar database could be found which possessed the necessary range of motions coupled with the required variation in motion performance quality.

The most important contribution of this work is our method of creating standardized representations for a given motion, based on examples gathered from a variety

of subjects. These representations model both the posture and behavior of subjects as they progress through the given motion. Once this baseline model for a motion has been established, it then becomes possible to compare new motion sequences to that model, in order to detect any deviations that might be present.

The method used to create these motion models is adapted from that proposed by Liu and Sarkar [18] for the analysis of human gait. The technique uses processed silhouette data to create a set of exemplar frames which represent different stages in the motion, and then train HMMs to model how a human subject progresses through those stages. Liu and Sarkar were interested only in the study of human gait, and its application in subject identification. For this reason, they developed a separate HMM for each subject, so as to better differentiate between them when a motion sequence was analyzed. In contrast, our work expands this technique to include multiple motions, which we will analyze not to determine subject identity, but instead to find motion irregularities.

3.1 Research Questions

The shift in focus from subject identification to motion quality evaluation opens up new possibilities for the ways in which the motion representation models can be trained. A separate model must be developed for each motion. However, once a motion is selected, there are still two key factors to consider when training the model. It is with these considerations in mind that we formulated our main research questions.

1. Is the standardized model dependent on the trained subject(s)?

2. Is the standardized model dependent on the trained speed(s)?

The first question deals with the consideration of whether or not the standardized model produced is greatly influenced by the inclusion of only a single subject, or multiple subjects, in the training set.

The second question is similar, but instead explores the possible effects of using only motion sequences performed at one speed for training, as opposed to using motion sequences with a variety of speeds.

Both of these questions are important, as they will greatly influence how future models must be trained in order to obtain useful results. Each question will be

explored independently, keeping other variables constant so as to minimize external effects. Standardized representation models will be created for each case considered above, and identical input sequences will be given to each model so that the results can be compared. The results of this testing will be discussed further in Section 4.3.

3.2 Algorithm Overview

The process of Human Motion Analysis consists of two main stages: Data acquisition, and data processing. Although the data processing stage is the main focus of this research, we will still provide an overview of the acquisition techniques used and the motivations behind their use.

3.2.1 Acquisition & Pre-Processing

Human motion sequences were recorded in a laboratory environment where all aspects of recording could be controlled including: background, objects in the scene (props), and lighting. Some aspects of the subject were also controlled, such as the style and color of clothing worn. Sequences were recorded with a single subject performing three separate tasks under various conditions. For full details of the sequences recorded and controls on the recording environment, see Section 4.1.1.

Once a human motion video has been recorded, the pre-processing steps described below are used to segment the subject from the background. This produces a binary image consisting of the subject's silhouette contour which, after some additional processing, is used as input to a HMM.

3.2.2 Background Subtraction

A static background subtraction algorithm is used to separate the human subject from the surrounding environment. This technique was chosen over more common adaptive background techniques due to the nature of the various motions that were performed by the subjects. In cases where portions of the subject's body remain stationary, adaptive background subtraction would lead to those portions of the body being mislabeled as background. A static background subtraction algorithm requires that a separate video B be recorded which consists of *only* the background. This video is then compared to the input video I with the subject present, and any corresponding regions that are determined to be similar are labeled as background.

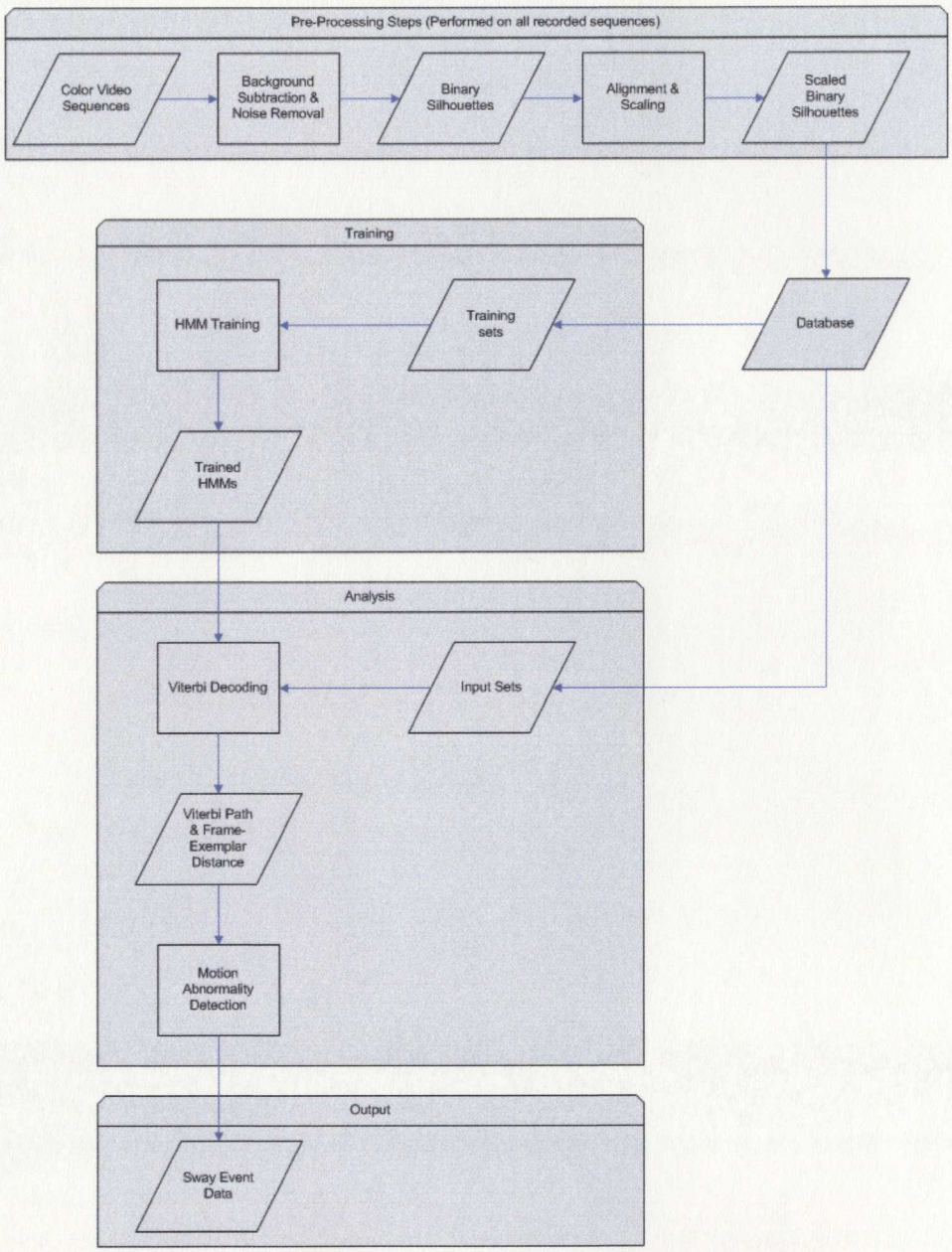


Figure 3.1: Proposed system for creation of standardized motion models and motion abnormality detection.

(a) Frame 0 (b) Frame 25 (c) Frame 50 (d) Frame 75

Figure 3.2: Sample video frames, prior to background subtraction.

A required assumption of static background subtraction is that the subject is the only thing that differs between the two videos. All other props or objects in the scene (ie: chairs) must remain stationary, otherwise they will be labeled as foreground objects. This assumption is valid for all motions considered in this research, as well as many other motions that could be studied. In cases where it is necessary for a background object to move and then once again become part of the background, a more suitable background subtraction method would need to be selected.

3.2.3 Noise Removal

Each frame of the segmented video is then processed to remove noise. The occurrence of noise happens in three ways: scattered pixels mis-labeled as foreground, scattered pixels mis-labeled as background, and large pixel "blobs" which result from objects other than the subject being labeled as foreground. The first type of noise is dealt with using a series of erosions followed by dilations. The second type of noise is dealt with using a single dilation followed by an erosion. The third type of noise is more problematic, and was present when a small shift in the chair surface occurred during "sit-to-stand" motion sequences. In this case, the segmented portion of the chair would appear as a region of foreground-labeled pixels very close (usually directly connected to) the subject's silhouette. To correct for this, multiple erosion operations were performed in an attempt to "disconnect" the region from the subject's silhouette. The image is then searched to and all foreground regions found. Any regions with an area below a chosen threshold are re-labeled as background. The threshold value is, through observation, chosen to be greater then any mis-labeled regions, while also being lower than the area of the subject's silhouette. After all regions have been considered, multiple dilation operations were performed to reverse the original erosions.

(a) Original frame (b) Binary silhouette (with noise) (c) Binary silhouette (with noise removed)

Figure 3.3: Background subtraction and noise removal.

3.2.4 Alignment & Scaling

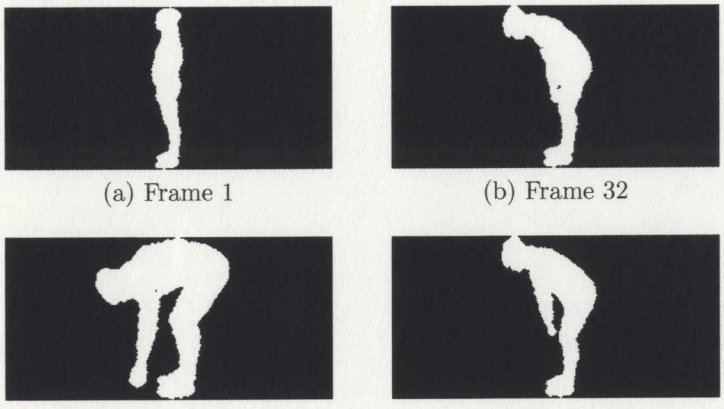
In preparation for the processing steps required for the algorithm in both the training and the Viterbi decoding of the HMM (See Sections 3.3.2 and 3.3.3), silhouettes from all video frames must be properly scaled and aligned.

In order to standardize the size of every silhouette, each video frame is cropped horizontally and vertically so as to contain only the bounding box of the silhouette. This cropped image is then scaled to a pre-determined height of 128 pixels. The width of the image is scaled in proportion to the height in order to maintain the aspect ratio. Black space is then added to the left and right edges of the image in order to increase the width of the image to 257 pixels. The black space is added in such a way as to keep the center of mass of the silhouette centered horizontally in the image.

These binary silhouettes, once aligned and scaled, formed the feature space of the Hidden Markov Models which are detailed in the following sections. Each silhouette image requires the storage of a 128x257 array of 1 bit values, with the number of images dependent on the number of frames in the video being processed.

3.3 Hidden Markov Models

In this section, we will discuss hidden Markov models (HMMs) in detail. We will begin with a general overview giving a definition of HMMs using this research as an example case, then move on to an in-depth description of the precise HMM implementation



(a) Frame 1 (b) Frame 32

(c) Frame 45 (d) Frame 55

Figure 3.4: Rescaling and horizontal Alignment.

used in this study.

3.3.1 HMMs: An Overview

To understand what a hidden Markov model is, it is best to start by first introducing the idea of a "regular" Markov model. A complete explanation and tutorial on Markov models is presented in [29], we will cover the relevant information here. A Markov model is a statistical model parameterized by a set of states, emissions (also commonly referred to as observations), and probabilities distributions associated with each, referred to as the transition and emission variables respectively. In the case of the regular Markov model, both the state and observation values are visible. A simple example would be a Markov model relating different weather states, and the possible observations associated with each weather state. For this example, three possible "states" would be: Sunny, Cloudy, and Rainy. The emission values could be an observation of whether a specific person either is, or is not, carrying an umbrella on a given day. In this case, both the states (weather condition) and emissions (presence or absence of umbrella) are visible to an outside observer. Given a set of probabilities defining the transition from one weather state to another, and a set of probabilities defining how likely the person is to carry their umbrella under each weather condition, it is possible to model the behavior of the system.

A hidden Markov model is a special case wherein only the emission values are visible, while the states producing those emissions are "hidden". To use the above example, this would equate to an observer who is locked in a building with no windows

and is unable to observe the weather conditions outside, instead only able to observe if the person walking past their room is carrying an umbrella or not. Given the associated transmission/emission probability variables, and a sequence of emission values, it is possible to predict the sequence of "hidden" state values which produced the given emission sequence.

Stated more formally, a hidden Markov model is specified by:

A set of states, $S = \{s_1, s_2, \ldots, s_N\}$

The prior probabilities, $\pi_i = P(q_1 = s_i)$ which define the probability of each state, s_i being the first state in a state sequence, Q, of arbitrary length. Note that in a regular Markov model, Q is visible, while in a hidden Markov model, it is not.

The transition probabilities which define the probability of the model moving from state i to state j. It is commonly referred to as the Transition matrix, as the probabilities are arranged into an NxN matrix.

The emission probabilities which define the probability of an observation (emission) x, given the current state of the model, s_i. There are two types of emissions to be considered:

- *discrete observations*, $x_n \in \{v_1, \ldots, v_K\}$. In this case, the emission probabilities take the form $b_{i,k} = P(x_n = v_k | q_n = s_i)$, the probability of observing v_k given the current state s_i. The probabilities are organized into an NxK matrix, referred to as the emission matrix.
- *continuous observations*, $x_n \in \mathbb{R}^D$: A set of probability density functions (pdfs) over the observation space for the model being in state s_i.

The (hidden) state sequence, $Q = \{q_1, q_2, \ldots, q_L\}, q_l \in S$**:** The sequence of states through which the model progresses.

The observation sequence, $X = \{x_1, x_2, \ldots, x_L\}$**:** The sequence of observations which are produced as the model progresses through the state sequence.

The probability of a state sequence Q being produced by a HMM with parameters Θ is defined as:

$$P(Q|\Theta) = \pi_{q_1} \cdot \prod_{n=1}^{N-1} a_{q_n,q_{n+1}} = \pi_{q_1} \cdot a_{q_1,q_2} \cdot a_{q_2,q_3} \cdot \ldots \cdot a_{q_{N-1},q_N} \tag{3.1}$$

The likelihood of an observed sequence X given a state sequence Q is:

$$P(X|Q,\Theta) = \prod_{n=1}^{N} P(x_n|q_n,\Theta) = b_{q_1,x_1} \cdot b_{q_2,x_2} \cdot \ldots \cdot b_{q_N,x_N} \tag{3.2}$$

The joint likelihood of an observed sequence X and a state path Q (i.e. the probability of X and Q occurring simultaneously) is:

$$P(X,Q|\Theta) = P(X|Q,\Theta) \cdot P(Q|\Theta) \tag{3.3}$$

The likelihood of an observation sequence X being produced by a HMM with parameters Θ:

$$P(X|\Theta) = \sum_{\text{all } Q} P(X,Q|\Theta) \tag{3.4}$$

Traditionally, HMMs have been used as a classification tool to identify the motion being performed, or to identify which subject is performing the motion. In this research, hidden Markov models are used as a tool to temporally normalize and analyze the motion of human subjects, in an attempt to gauge the quality of the motion performed. This is a new approach for the application of HMMs to human motion analysis.

In our HMM implementation, the set of states consists of a collection of "exemplar" images. These images represent the various stages of the motion from start to finish. They can be thought of as "snapshots" of an average subject performing the motion, taken at different points in the motion's progression. The number of states, N, defines the temporal resolution of the model, and is set by the user before the model is created. Several values of N were used in testing (See Section 4.2).

The prior probabilities are always taken to be $\mathrm{P}(q_1 = s_1) = 1$. This is a required condition of the Viterbi algorithm, and is an easy assumption to enforce when recording human motions. Each recording is begun with the subject in the "starting position" for the given motion (ie: Recordings are not begun with the subject being already "part way" through a motion). This allows for every state sequence to start at state s_1, as required.

Transition probabilities represent the probably that, between frames i and i+1 of an input video, the model moves from state s_j to s_k. The emission probabilities are more complicated, as a continuous probably model is used as opposed to a discrete probability model like that of the above example. Emission values are represented by the input silhouette images created in pre-processing, with each image denoting one emission value. Emission probabilities are calculated as N-dimensional vectors derived from the distance between the given image and each of the N exemplar frames.

The use of HMMs in this research has two primary goals. First, by pairing every frame of an input video with its appropriate motion state, it becomes possible to compare motion sequences independently of the speed at which those motions were performed. This is extremely important, as it does not make sense to compare a frame 10% of the way through a motion with a frame that is 60% of the way through a motion. Second, the HMM can be used to detect and identify possible abnormalities in an input motion video.

The use of HMMs allows us to create (through training) a baseline representation of a particular motion. Motion videos can then be compared to this baseline to produce corresponding state sequences. A state sequence for a single input video can then be analyzed for abnormalities such as sways. Comparing two different state sequences would allow us to measure relative differences in the speed of the motion. We can also compare frames from equivalent states of the motion in an attempt to identify differences in posture or movement style of the subjects.

3.3.2 Training

Before training begins, a training set must be selected. Different training sets were used depending on what aspect of the research was being explored. For instance, when testing subject independence of the model (See Section 4.3.1), several training sets were used, each involving videos from a single subject, as well as other training sets which involved a mixture of subjects. The size of the training set is arbitrary, containing any number of videos, each having their own length. The only restriction imposed on the training set is that every video consists of the same motion being performed by the given subject. For further details on the training set composition, see section 4.1.2.

The number of states for the model must also be selected. Once a value for the number of states has been decided, it must be kept constant throughout training the

training and analysis. To test with a different number of states would require that the model be retrained. Three values of N were tested: 10, 20, and 30.

The training algorithm used to estimate the HMM parameters is adapted from that used by Sarkar et al. [18]. It employs an iterative clustering to the video frames from a specified training set, and uses the results to create "average silhouettes" referred to as state exemplars. It then calculates transition and emission probabilities. This training algorithm implementation was chosen for its ability to accommodate the specific feature space being used (large arrays of binary values).

The Initial clustering is created by dividing each training sequence into N equally sized groups of frames. For each frame cluster, a centroid frame is calculated by taking the average of each frame in the cluster. The clusters are then updated by taking the endframes of each cluster group which are adjacent to other clusters, and calculating the distance between the endframe and its cluster centroid, as well as the distance between the endframe and the adjacent cluster's centroid.

The similarity, $S(\mathbf{f}_i, \mathbf{f}_j)$, of two binary silhouettes is measured by taking the ratio of pixels in their intersection to the pixels in their union.

$$S(\mathbf{f}_i, \mathbf{f}_j) = \frac{{\mathbf{f}_i}^T \mathbf{f}_j}{{\mathbf{f}_i}^T \mathbf{f}_i + {\mathbf{f}_j}^T \mathbf{f}_j - {\mathbf{f}_i}^T \mathbf{f}_j} \tag{3.5}$$

The distance between the two silhouettes is then defined as one minus the similarity:

$$D(\mathbf{f}_i, \mathbf{f}_j) = 1 - S(\mathbf{f}_i, \mathbf{f}_j) \tag{3.6}$$

These values are commonly referred to as the Tanimoto similarity (3.5), and Tanimoto distance (3.6).

If the distance to the adjacent centroid is smaller, then the endframe is moved to that cluster. This process is repeated once for each cluster of each training video, then the centroid frames are re-calculated. This process of calculating centroids and updating the cluster groups is repeated until an update completes in which no frames move to a different cluster (usually occurring in 25-35 iterations). The final set of cluster centroids to be calculated become the state exemplars, and are used to represent the "average" silhouette for the given state.

The average "frame-to-exemplar" distance is then calculated for each exemplar:

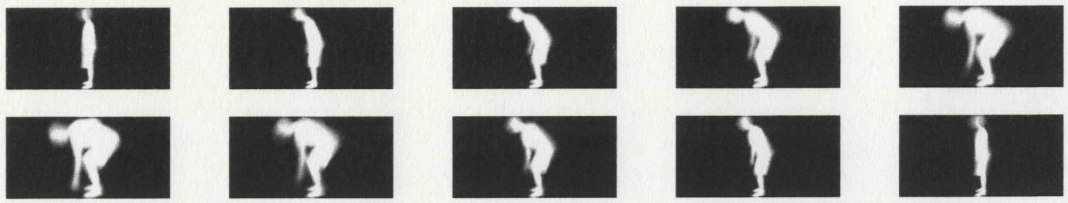


Figure 3.5: Exemplars for a 10-state HMM modeling a "pickup" motion.

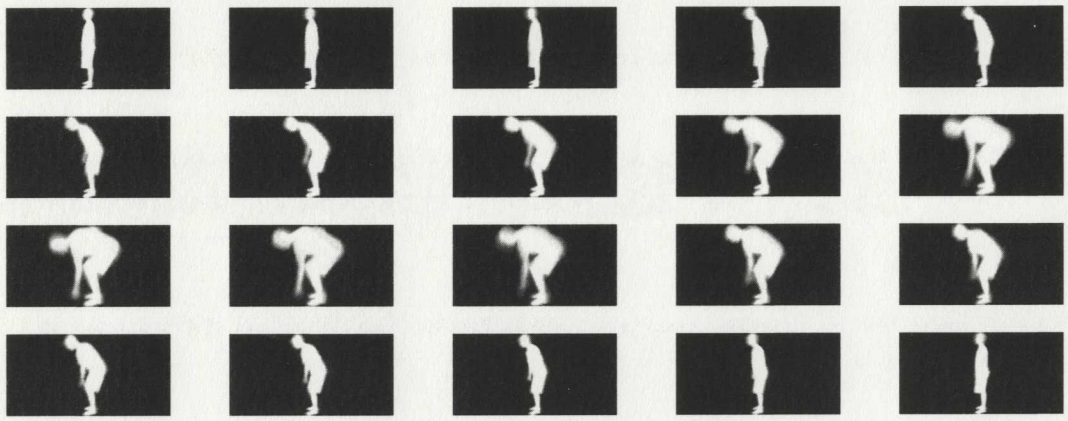


Figure 3.6: Exemplars for a 20-state HMM modeling a "pickup" motion.

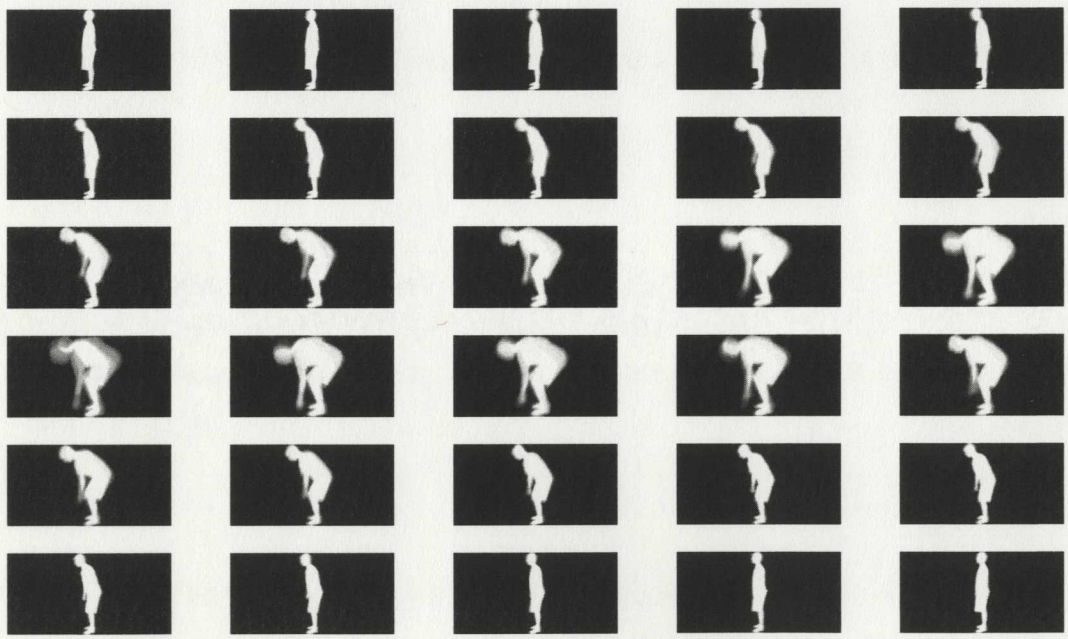


Figure 3.7: Exemplars for a 30-state HMM modeling a "pickup" motion.

$$\mu_j = \frac{\sum_{\mathbf{f}_i \in E_j} D(\mathbf{f}_i, \bar{E}_j)}{|E_j|} \tag{3.7}$$

Continuous probability emission vectors are calculated for each of the training video frames. Every frame is compared to each of the N exemplars, and the Tanimoto distance is calculated. The probability vector is defined using an exponential parameterization of this distance, and the corresponding μ_j.

$$b_j(\mathbf{f}_t) = \frac{1}{\mu_j} e^{-\frac{D(\mathbf{f}_t, \bar{E}_j)}{\mu_j}} \tag{3.8}$$

The final step of training is to generate the transition probability matrix, A. An initial estimate of A is generated using the exemplars.

$$a^{[0]}(i,j) = \frac{\sum_k \#\text{ of } \mathbf{f}_{t+1}^k \text{ in } E_j \text{ given } \mathbf{f}_t^k \text{ in } E_i}{|E_i|} \tag{3.9}$$

It is then updated using Levinson's method, a process for training using multiple observation sequences which is based on the iterative Baum-Welch algorithm [27].

$$a^{[n+1]}(i,j) = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k - 1} \alpha_t^k(i) a^{[n]}(i,j) b_j(\mathbf{f}_{t+1}^k) \beta_{t+1}^k(j)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k - 1} \alpha_t^k(i) \beta_t^k(i)} \tag{3.10}$$

where $P_k = P(\mathbf{F}^k|\lambda)$, the likelihood of kth observation. P_k, α^k, and β^k are calculated as follows:

$$\alpha_t^k(j) = P(\mathbf{f}_1^k, \ldots, \mathbf{f}_t^k, q_t = j|\lambda) \tag{3.11}$$

$$= b_j(\mathbf{f}_t^k)/N_s; \text{ for } t = 1 \text{ and } 1 \leq j \leq N_s \tag{3.12}$$

$$= \sum_{i=1}^{N_s} \alpha_{t-1}^k(i) a^{[n]}(i,j) b_j(\mathbf{f}_t^k); \text{ for } 2 \leq t \leq T_k - 1 \text{ and } 1 \leq j \leq N_s \tag{3.13}$$

$$P_k = P(\mathbf{F^k}|\lambda) = \sum_{j=1}^{N_s} \alpha_{T_k}(j) \tag{3.14}$$

$$\beta_t^k(i) = P(\mathbf{f_{t+1}^k}, \ldots, \mathbf{f_{T_k}^k} | q_t = i, \lambda) \tag{3.15}$$

$$= 1; \text{ for } t = T_k \text{ and } 1 \leq i \leq N_s \tag{3.16}$$

$$= \sum_{j=1}^{N_s} a^{[n]}(i,j) b_j(\mathbf{f_{t+1}^k}) \beta_{t+1}^k(j); \text{ for } t = T_k - 1, \ldots, 1 \text{ and } 1 \leq i \leq N_s \tag{3.17}$$

After convergence, a final modification is made to the transition probabilities in order to account for the possibility that the subject might reverse their direction of motion mid-sequence, as is sometimes the case when a "sway" occurs. This modification adds a small amount to the probability of state i transitioning to state i-1, and (in order to retain the sum of each row equaling 1) removes the same amount from the probability of the state remaining the same.

3.3.3 Viterbi Decoding

In order to analyze an input video, it must be converted into a sequence of continuous probability emission vectors, $X = \{x_1, x_2, \ldots, x_L\}$, similar to those generated for the training videos. The process is the same, with one vector, x_l, per video frame being produced.

Viterbi decoding is used to compute the output of the HMM [30]. Given the HMM parameters (Priors, Transition probabilities, and Emission probabilities), and a sequence of input emissions (X), the Viterbi algorithm finds the "most likely state sequence" along which the model could progress in order to produce the given emissions. This output state sequence, $Q = \{q_1, q_2, \ldots, q_L\}$, is stored as a series of integer values which range between 1 and N, identifying the most likely state, s_n, for frame l.

3.4 Detection of Motion Abnormalities

Once the Viterbi path, Q, has been generated for a given input sequence, the path can be analyzed to detect motion abnormalities (i.e. sways).

The analysis begins by grouping the sequence $Q = \{q_1, q_2, \ldots, q_L\}$ into K ordered, non-overlapping, partitions consisting of adjacent q_i which are of equal value:

$$D = \{D_1, D_2, \ldots, D_K\} = \Big\{\{q_{1,1}, \ldots, q_{1,d_1}\}, \{q_{2,1}, \ldots, q_{2,d_2}\}, \ldots, \{q_{K,1}, \ldots, q_{K,d_K}\}\Big\} \tag{3.18}$$

such that $q_{k,j} = g_k$ for $1 \leq j \leq d_k, 1 \leq k \leq K$, and no adjacent D_k have equal g_k.

We detect sway events by considering the ordered list of $\{g_k, d_k\}$ pairs for $1 \leq k \leq K$, where g_k represents the state value assigned to a partition, and d_k the size of the partition (i.e. the number of frames that the subject remained in state g_k before transitioning to another).

A sway event is defined by an ordered sequence of g_i such that $g_i > g_{i+1}, \ldots, g_{i+j}$ where either $i + j = K$ (implying that g_{i+j} is the state value of the final partition), or $g_{i+j+1} \geq g_i$ (indicating the end of the sway event). When this occurs, the following information is gathered about the event:

sway location $= g_i$; The state at which the sway began.

sway size $=$ **min**$\{g_{i+1}, \ldots, g_{i+j}\}$; The largest reversal of motion observed during the sway event.

sway duration $= \sum_{t=i+1}^{i+j} d_t$; The number of frames over which the sway event occurred.

Tables 4.3 and 4.4 in section 4.2 give a detailed example of a Viterbi path, and its corresponding partitioning, which is used to calculate the details of a detected sway event.

This information is of value to physicians, as it indicates at what point in the motion the event occurred, and provides a measurement of the severity and duration of the abnormality. Such knowledge will aid with the diagnosis of health problems, and planning of specific treatments. As treatment progresses, further analysis can be done to see if the abnormalities diminish, thus allowing the physician to track a patient's recovery progress.

Chapter 4

Experimental Results

4.1 Database

4.1.1 Sequence Acquisition

Human motion sequences were recorded using a single camera connected via firewire to a Windows desktop PC. Using the Computer Vision Acquisition Laboratory at the University of Victoria, a controlled environment was created for the recording of video sequences. Heavy curtains and diffused overhead lights were used to maintain uniform light levels. One wall of the Acquisition Lab was painted with black non-reflective paint to reduce the appearance of light specularities and shadows. The carpet and all props used in the recording area were covered with black sheets, also for the purpose of reducing the appearance of shadows.

With a recording environment which was colored entirely black it was important, for background subtraction purposes (See Section 3.2.2 for more details), for the human subject to wear colored clothing so as to be easily distinguishable. Loose or "baggy" clothing was not used, as such clothing could adversely effect the segmented contour for the given motion being recorded. For instance, during a recording for the "stepping" motion, a long skirt worn by the subject might occlude both legs, leading to a segmented contour which would appear much different than if the same subject had performed the same motion while wearing close-fitting pants. Long hair posed a similar problem, in that it would drape down during a "pickup" motion, greatly effecting the segmented contour. With these considerations in mind, participants in the study were asked to wear colorful, close-fitting clothing, remove any jewelery (a possible source of light specularities), and to wear long hair "up". In cases where the

subject's hair color was dark, a red toque was worn to cover the hair completely.

The subject was then asked to perform one of the three chosen motions:

Pickup From an upright standing position, bending down to pick up a small object from the floor.

Stepping From an upright standing position, lifting a foot and placing it on a raised step directly in front of ones' self, then lowering the foot to return to their original position.

Sit-to-Stand From a sitting position, "getting up" to stand in an upright position.

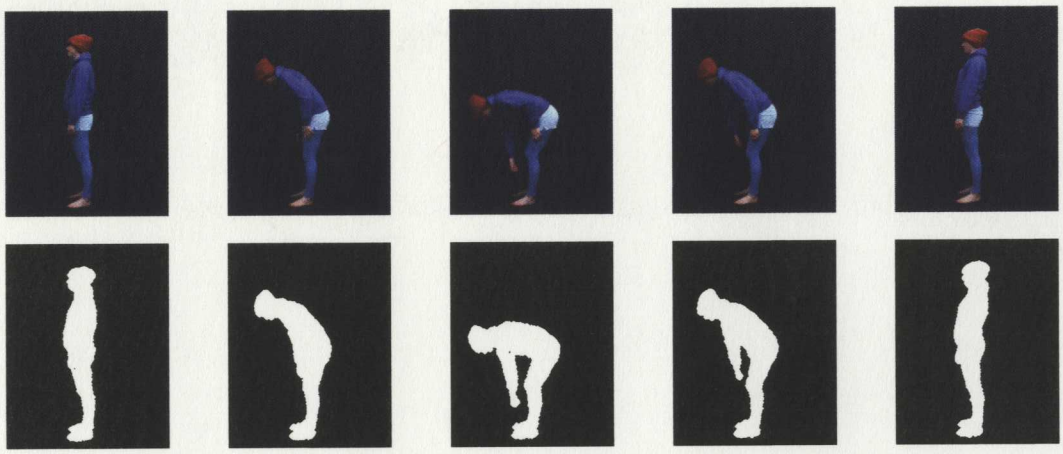


Figure 4.1: Pickup motion. (Top) Original frames. (Bottom) Binary silhouettes.

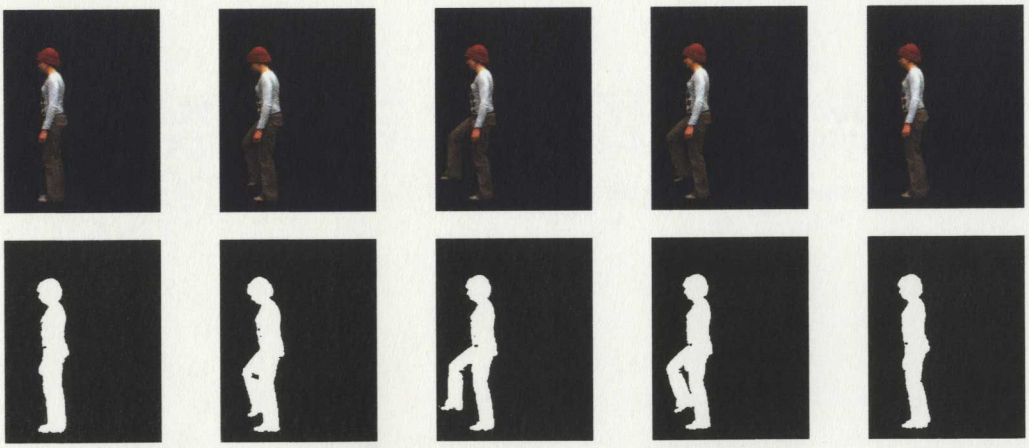


Figure 4.2: Stepping motion. (Top) Original frames. (Bottom) Binary silhouettes.

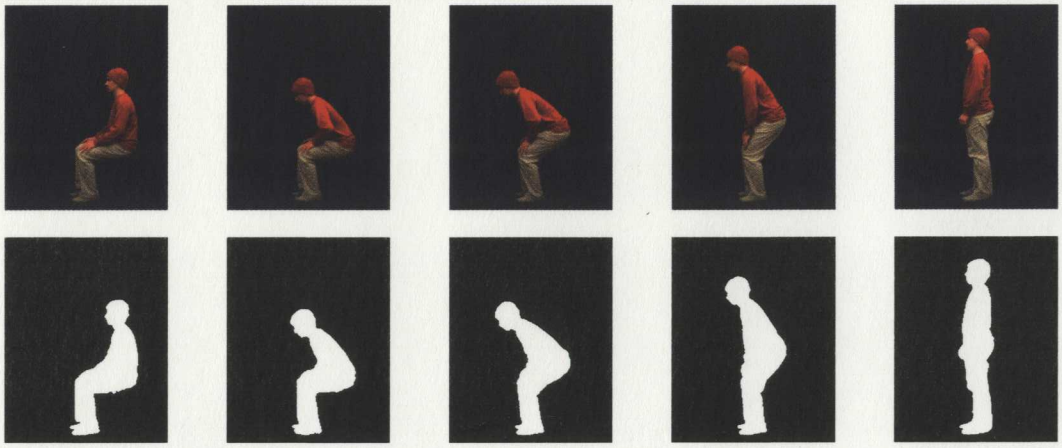


Figure 4.3: Sit-to-Stand motion. (Top) Original frames. (Bottom) Binary silhouettes.

These actions were performed with the following variations:

Normal The subject was asked to perform the motion in a "natural" way, with a speed and manner that felt most appropriate to the subject.

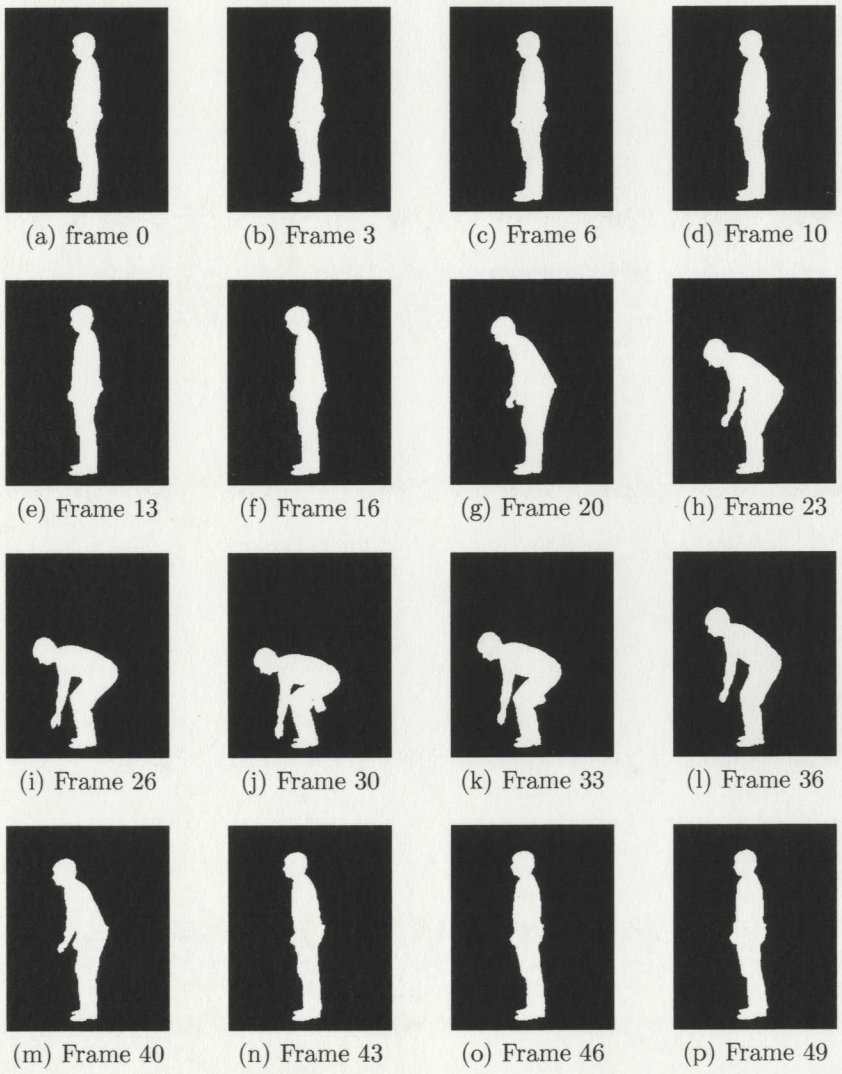
Slow The subject was asked to perform the motion more slowly, taking roughly twice the "normal" amount of time to complete it.

Fast The subject was asked to perform the motion quickly, taking roughly half of the "normal" amount of time to complete it.

Variable The subject was asked to vary the speed at which they performed the action, as they were performing it. The point(s) in the motion where they changed speed, and whether those changes were to "speed-up", or "slow-down" were varied with each recording.

Hesitate The subject was asked to perform the motion at a normal speed, but also to "hesitate" at some point during the motion. This would appear as a pause in the motion, after which the subject would continue normally.

Sway The subject was asked to perform the motion at a normal speed, but also to "sway" at some point during the motion. This sway could manifest in two ways: As a "wobble" in the subject's body (similar to a temporary loss of balance), or as a small reversal in the motion (i.e. rocking back and forth).



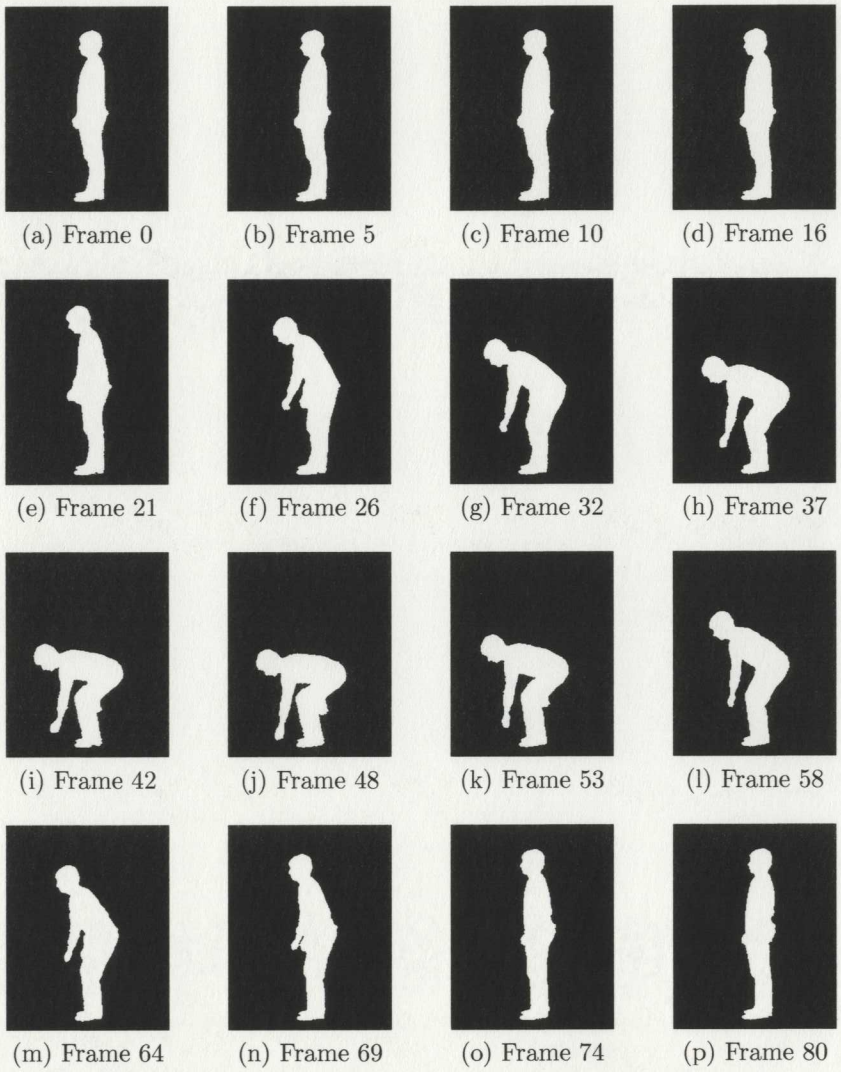
(a) frame 0 (b) Frame 3 (c) Frame 6 (d) Frame 10

(e) Frame 13 (f) Frame 16 (g) Frame 20 (h) Frame 23

(i) Frame 26 (j) Frame 30 (k) Frame 33 (l) Frame 36

(m) Frame 40 (n) Frame 43 (o) Frame 46 (p) Frame 49

Figure 4.4: Sample frames for motion performed at "fast" speed.



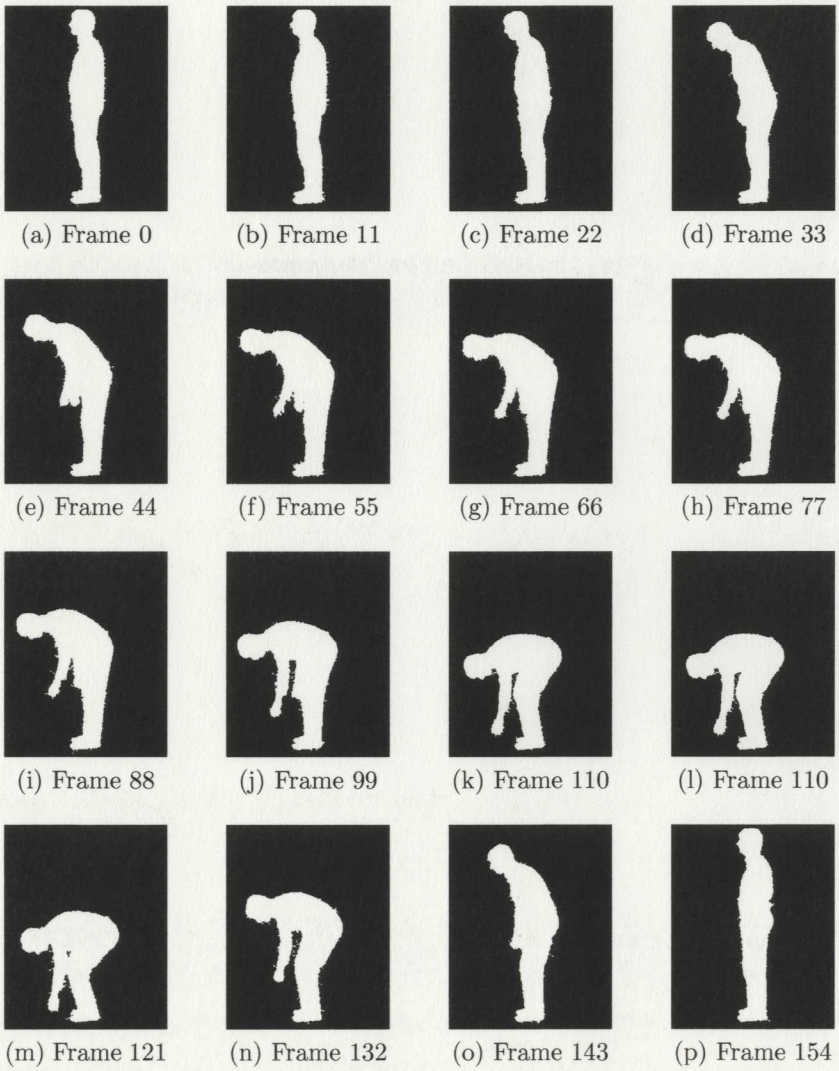
(a) Frame 0 (b) Frame 5 (c) Frame 10 (d) Frame 16

(e) Frame 21 (f) Frame 26 (g) Frame 32 (h) Frame 37

(i) Frame 42 (j) Frame 48 (k) Frame 53 (l) Frame 58

(m) Frame 64 (n) Frame 69 (o) Frame 74 (p) Frame 80

Figure 4.5: Sample frames for motion performed at "slow" speed.



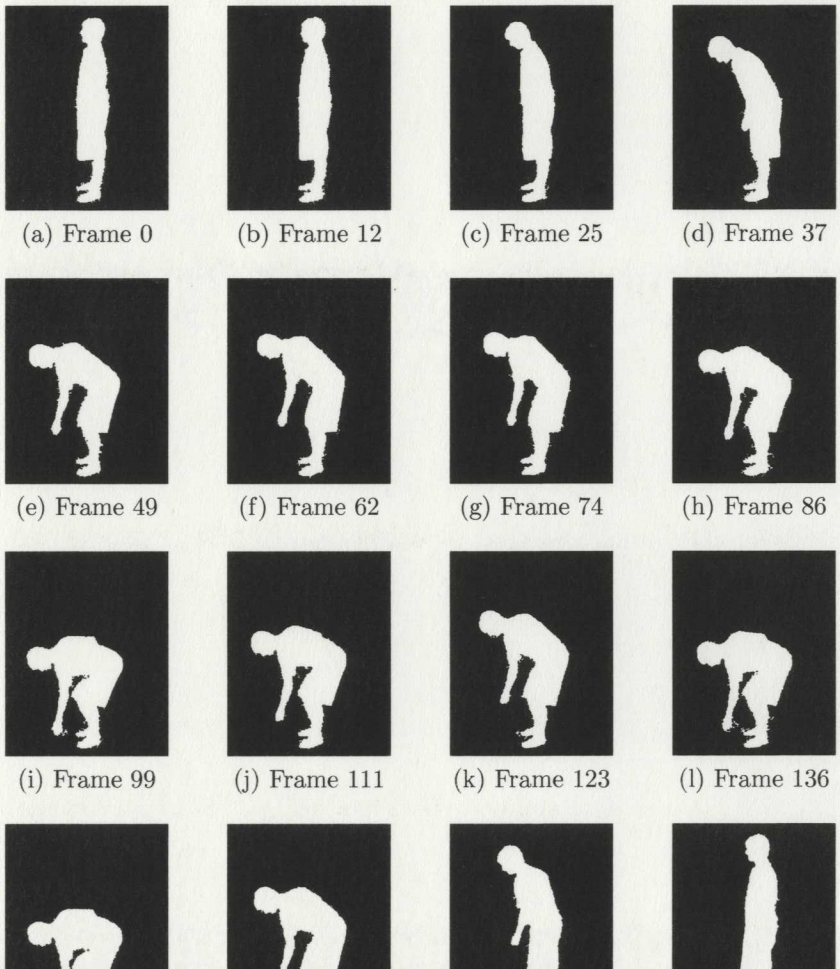
(a) Frame 0 (b) Frame 11 (c) Frame 22 (d) Frame 33

(e) Frame 44 (f) Frame 55 (g) Frame 66 (h) Frame 77

(i) Frame 88 (j) Frame 99 (k) Frame 110 (l) Frame 110

(m) Frame 121 (n) Frame 132 (o) Frame 143 (p) Frame 154

Figure 4.6: Sample frames for motion performed with a "hesitation".



(a) Frame 0 (b) Frame 12 (c) Frame 25 (d) Frame 37

(e) Frame 49 (f) Frame 62 (g) Frame 74 (h) Frame 86

(i) Frame 99 (j) Frame 111 (k) Frame 123 (l) Frame 136

(m) Frame 148 (n) Frame 160 (o) Frame 173 (p) Frame 185

Figure 4.7: Sample frames for motion performed with a "sway".

Several videos were recorded of each subject, performing each motion, for each variation. Additional "background" videos were recorded at the start of each recording session. These videos were 100 frames in length, and consisted of the environment with no subject present. These background recordings were repeated periodically during each subjects' recording session, in order to assure correct background subtraction in the event that a prop was moved slightly by the subject.

Camera settings (brightness, gain, etc.) were kept fixed during all recordings. Videos were recorded at 25 frames per second with a resolution of 640x480 pixels, and were saved to the PC frame-by-frame as uncompressed 24-bit color bitmap files.

4.1.2 Database Composition

Five human subjects (3 male, 2 female) were used to record motion sequences. Subjects ranged between 26 and 29 years of age, and no subjects possessed any mental or physical disabilities. All subjects were of average build, with the heights of the male subjects being 6ft 3in, 6ft, and 5ft 6in, while the heights of the female subjects were 5ft 7in and 5ft.

Using the methods described in the previous section, numerous videos were recorded for each subject, motion, and variation. To avoid possible problems with testing different numbers of HMM states, only motion sequences made up of 30 or more frames were selected for use in the database. This choice is based on the fact that 30 is the highest number of HMM states which was tested. A combined total of 1417 motion sequences were obtained, composing roughly one hour of video, and requiring approximately 52GB of storage space.

4.1.3 Training and Analysis Sets

Several variables (number of subjects, speed variations) were considered when training HMM parameters, and several variations were used for the creation of training sets. These variations are detailed below:

Single subject, normal speed Training set consisted motion sequences from the same subject, and all performed at "normal" speed.

Single subject, mixed speed Training set consisted motion sequences from the same subject. Sequences were selected for each of the speed variations: Normal, fast, slow, and variable.

Motion	Variation	Subject #				
		1	2	3	4	5
Pickup:	Normal	32	31	50	44	23
	Fast	25	22	43	23	36
	Slow	10	14	20	10	13
	Variable	19	13	11	12	17
	Hesitate	13	8	5	10	N/A
	Sway	8	6	7	10	N/A
Stepping:	Normal	26	33	41	29	25
	Fast	22	24	40	23	42
	Slow	9	11	17	10	16
	Variable	11	13	20	12	16
	Hesitate	12	9	9	9	N/A
	Sway	11	9	10	9	N/A
Sit-to-Stand:	Normal	14	20	29	21	14
	Fast	14	14	11	12	20
	Slow	11	17	21	15	23
	Variable	10	8	11	9	15
	Hesitate	7	6	5	6	N/A
	Sway	6	6	7	7	N/A

Table 4.1: Number of motion sequences in database for each combination of: subject, motion, and variation.

Multi-subject, normal speed Training set consisted motion sequences from every subject, all performed at "normal" speed.

Multi-subject, mixed speed Training set consisted motion sequences from every subject. Sequences were selected for each of the speed variations: Normal, fast, slow, and variable.

For each of the above cases, a training set was selected for each subject (where appropriate), and each motion (pickup, stepping, sit-to-stand). This lead to the creation of 36 separate training sets. Each set was then used to train three HMMs consisting of $N = 10$, 20, and 30 states (for a total of 108 trained HMMs).

Every motion sequence, whether included in a training set or not, was then taken as input sequences and analyzed using HMMs trained using either the corresponding single-subject training sets, or the multi-subject training sets for the given motion. The analysis was done for both normal and mixed speed training sets, as well as for all three cases for the value of N. The resulting data from each analysis was saved

a file as described in section 3.4. Further details of these results are discussed in the next section.

Training and analysis was performed on a PC with a dual-core 3.0GHz CPU (1333MHz FSB), 8GB of DDR2 RAM (800MHz), SATA2 1.5TB 7200rpm hard disk drive, Windows 7 Professional operating system, and running MatLab version 2010a.

The number of clustering iterations and computation time required for training varied greatly depending on the type and size of the training set. Smaller, less varied training sets such as "Single subject, normal speed" sets normally required 15 to 25 clustering iterations, and from 3-6 minutes of computation time. Larger, more varied training sets such as "Multi-subject, mixed speed" sets usually required 25 to 35 clustering iterations, and as much as 2 to 4 hours of computation time. The time required to compute training parameters for all 108 HMM conditions was approximately 3 days.

Once the HMMs were trained, analysis of input sequences was much faster, taking an average of 20 seconds per sequence. However, with roughly 1400 input sequences in the database, each being analyzed under 12 different conditions (single or multi subject, normal or mixed speed, and three different values of N), the number of trials was very large (approximately 17000). The computation time of all of the analysis trials combined was approximately 4 days.

4.2 HMM Analysis Results

Viterbi paths were generated for each video in the database by comparing each to various HMMs trained under different conditions. Two examples of the produced Viterbi paths are provided here, one of a normal input motion sequence with no abnormalities, and one of a motion in which a sway event occurs.

Table 4.2 presents the Viterbi path generated for a "sit-to-stand" motion performed at normal speed. The path indicates a relatively smooth transition from state to state, with no detectable abnormalities.

Table 4.3 gives the Viterbi path for a "sit-to-stand" motion in which a sway event occurs. The sway event begins at state 9, at which point the motion is reversed back to state 5, and then returns to state 9 before continuing normally. By referring to table 4.4 and recalling the analysis steps defined in section 3.4, we calculate the location of the sway event to be 9, its size to be 4, and its duration to be 30.

4.3 Testing Research Questions

In order to test the two research questions posed in section 3.1, a comparison was done between the Viterbi paths generated for given input sequences by HMMs trained under different conditions. For a specific input sequence, the paths produced for each set of conditions were analyzed to determine the percentage of path elements that differed, the average magnitude of the difference, and the maximum magnitude of the difference. Details of each test, and the results obtained, are discussed in the following two sections.

4.3.1 Subject Dependence

To test subject dependence of the trained HMMs, a set of 10 motion sequences were selected for a given subject, in which the subject performed a motion at "normal" speed. These 10 sequences were then used as inputs for HMMs trained under two different conditions: The first using a "single subject, normal speed" training set, and the second using a "multi-subject, normal speed" training set. The corresponding results for each of the 10 inputs were then compared as described above.

This comparison was done for each of the 5 subjects, and for each of the three studied motions, for a total of 15 trials. The entire testing process is then conducted three times, to test HMMs with $N = 10$, 20, and 30 states.

Although we present results here for all three values of N, our discussion will focus on the 20 state case unless otherwise stated. The effects of varying the number of states will be discussed fully in section 4.4.1.

Frame index:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Assigned State:	1	1	2	3	3	3	3	3	3	3	3	3	3	3	3
Frame index:	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Assigned State:	4	5	5	5	6	7	7	8	9	9	9	9	9	9	9
Frame index:	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
Assigned State:	9	9	9	9	9	9	9	9	9	9	10	11	12	13	14
Frame index:	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
Assigned State:	14	14	15	15	15	15	16	17	18	18	18	18	18	18	18
Frame index:	61	62	63	64	65	66									
Assigned State:	18	18	18	18	18	18									

Table 4.2: Viterbi Path for a "sit-to-stand" motion.

Frame index:	1	2	3	4	5	6	7	8	9	10
Assigned State:	1	2	3	3	3	3	3	3	3	3
Frame index:	11	12	13	14	15	16	17	18	19	20
Assigned State:	3	3	3	3	3	3	3	3	3	3
Frame index:	21	22	23	24	25	26	27	28	29	30
Assigned State:	3	4	5	6	7	8	9	9	9	9
Frame index:	31	32	33	34	35	36	37	38	39	40
Assigned State:	9	9	9	9	9	9	9	9	9	9
Frame index:	41	42	43	44	45	46	47	48	49	50
Assigned State:	9	9	9	9	9	9	9	9	9	9
Frame index:	51	52	53	54	55	56	57	58	59	60
Assigned State:	8	7	6	5	5	5	5	5	5	5
Frame index:	61	62	63	64	65	66	67	68	69	70
Assigned State:	5	5	5	5	5	5	5	5	5	5
Frame index:	71	72	73	74	75	76	77	78	79	80
Assigned State:	5	5	5	5	5	5	5	6	7	8
Frame index:	81	82	83	84	85	86	87	88	89	90
Assigned State:	9	9	9	9	9	9	9	10	11	11
Frame index:	91	92	93	94	95	96	97	98	99	100
Assigned State:	11	11	12	12	12	12	12	12	12	13
Frame index:	101	102	103	104	105	106	107	108	109	110
Assigned State:	14	14	14	15	15	15	15	15	16	17
Frame index:	111	112	113	114	115	116	117	118	119	120
Assigned State:	18	18	18	18	18	18	18	18	18	18

Table 4.3: Viterbi Path for a "sit-to-stand" motion with a sway event.

k	1	2	3	4	5	6	7	8	9	10
g_k	1	2	3	4	5	6	7	8	9	8
d_k	1	1	19	1	1	1	1	1	24	1
k	11	12	13	14	15	16	17	18	19	20
g_k	7	6	5	6	7	8	9	10	11	12
d_k	1	1	24	1	1	1	7	1	4	7
k	21	22	23	24	25	26				
g_k	13	14	15	16	17	18				
d_k	1	3	5	1	1	10				

Table 4.4: Ordered pairs of $\{g_k, d_k\}$ for $1 \leq k \leq K$ where in this case $K = 26$, as defined in section 3.4.

Motion	Subject	% Diff	Average Mag.	Max Mag.
Pickup	1	38	1.4	2
	2	23	1	1
	3	13	1	1
	4	45	1	1
	5	31	1	1
Stepping	1	67	3	3
	2	50	2	2
	3	37	2.7	3
	4	61	3.7	4
	5	92	7.3	8
Sit-to-Stand	1	52	1.2	3
	2	38	2.6	3
	3	80	2	2
	4	54	2	2
	5	75	2	2

Table 4.5: Subject dependence results. ($N = 10$)

Motion	Subject	% Diff	Average Mag.	Max Mag.
Pickup	1	74	2	2
	2	65	2	2
	3	63	2.2	3
	4	72	2	2
	5	71	2.1	3
Stepping	1	85	8.3	16
	2	80	3.3	4
	3	57	4.8	5
	4	80	5.4	9
	5	74	2.1	3
Sit-to-Stand	1	64	2.2	3
	2	75	3	3
	3	92	3.5	4
	4	53	3.2	4
	5	85	4.5	5

Table 4.6: Subject dependence results. ($N = 20$)

Motion	Subject	% Diff	Average Mag.	Max Mag.
Pickup	1	70	3	3
	2	83	3.6	4
	3	75	2	2
	4	70	2.9	3
	5	73	3.8	4
Stepping	1	91	4	5
	2	88	6.2	12
	3	74	4.1	5
	4	77	5.7	6
	5	74	4.8	5
Sit-to-Stand	1	75	3	4
	2	76	3.9	6
	3	93	4	5
	4	51	3.5	5
	5	88	5.7	7

Table 4.7: Subject dependence results. ($N = 30$)

By observing the results listed in table 4.6 we can see that for most trials a large number of path elements are effected by the change in training sets, as seen by the average % of differing path elements ranging between 53% and 92%. However, when we examine the average magnitude of the difference in state values assigned to a given path element, we see that the differences are often quite small (i.e. between 2 and 4 state values).

In one trial (subject 1, stepping motion) a large aberration in the results was observed. Upon closer examination of the sequences in question, it was hypothesized that the source of this large discrepancy was due to the clothing worn by the subject, and the effects that this clothing had on the subject's silhouettes for the given motion. In this case, the subject was wearing knee-length, somewhat loose-fitting shorts. When performing the "stepping" motion, this article of clothing had the effect of occluding the upper half of the subjects legs, thus influencing the results when matching these silhouettes with those of subjects whose legs were not occluded in this fashion.

Excluding this exceptional case, in only one other trial did the magnitude of the difference in assigned state values rise above 5. From these results we can surmise that there is an observable effect on the standardized motion model due to the subjects included in the training set, but that this effect will not have a large impact on

analysis results.

A similar, though less severe, aberration was observed in one of the N=30 trials (subject 2, stepping motion). The deviation occurred in only 2 of the 10 input sequences for that trial, which is why the effect on the data was less pronounced than the above case, in which all of the input sequences were effected due to the subject's clothing. No obvious cause was found for the deviation in results for this trial, leading us to believe that they are simply exceptional error cases.

4.3.2 Speed Dependence

Testing the speed dependence of the trained HMMs is done in a similar fashion to the subject dependence tests outlined above. In this case the same set of 10 sequences per subject were selected (i.e. motion performed at "normal" speed), and used as inputs for HMMs trained using "single subject, normal speed" and "single subject, mixed speed" training sets. Again, the test was performed for each subject, and each motion.

For this test, we see a larger range in the average % of differing path elements, starting as low as 33% and going up to 96%. However, just as will the subject dependence tests, we observe that the magnitude of the difference is again relatively

Motion	Subject	% Diff	Average Mag.	Max Mag.
Pickup	1	44	1.3	2
	2	10	1	1
	3	32	1	1
	4	24	1	1
	5	11	1	1
Stepping	1	50	3	3
	2	62	1.3	2
	3	10	1.2	2
	4	51	3	3
	5	93	7.3	8
Sit-to-Stand	1	38	1.1	2
	2	48	1	1
	3	44	2	2
	4	45	1.1	2
	5	25	1.1	2

Table 4.8: Speed dependence results. ($N = 10$)

Motion	Subject	% Diff	Average Mag.	Max Mag.
Pickup	1	96	2	2
	2	39	1.3	2
	3	53	2	2
	4	65	3	3
	5	33	1.2	2
Stepping	1	84	10.7	18
	2	64	2.4	3
	3	34	2.7	5
	4	75	2.2	3
	5	88	3.3	4
Sit-to-Stand	1	48	2.3	4
	2	70	2	2
	3	86	3	3
	4	46	2.2	4
	5	63	2.1	3

Table 4.9: Speed dependence results. ($N = 20$)

Motion	Subject	% Diff	Average Mag.	Max Mag.
Pickup	1	82	3	3
	2	59	1.2	2
	3	63	3.8	4
	4	80	3.4	4
	5	62	2	2
Stepping	1	46	3	4
	2	82	4	6
	3	59	1.2	2
	4	68	2	2
	5	86	4.1	6
Sit-to-Stand	1	75	2.9	5
	2	64	2.6	6
	3	91	2.7	3
	4	46	2.3	4
	5	68	4.1	6

Table 4.10: Speed dependence results. ($N = 30$)

small in most cases.

The same aberration was detected in the trail for the stepping motion of subject 1. The magnitude of the aberration was roughly the same, and is assumed to have the same cause, as described in the previous section.

If we again exclude this single exceptional case, we observe that in no other trials does the magnitude of the difference in assigned state values rise above 5. Thus we draw the same conclusions as with the subject dependence testing, and say that although a difference in the model is found when including various motion speeds in the training set, the effects of this difference will likely have little impact on analysis results.

4.4 Additional Testing

In addition to exploring the research questions discussed above, some other aspects of the motion models, and the methods of analysis were considered. The first thing we will look at is the effect of changing the number of states in the trained HMMs.

4.4.1 Testing State Variations

Liu and Sarkar [18] tested the effects of varying the number of states in their gait analysis problem. Based on an analysis of the Akaike Information Criterion (AIC) [2], they concluded that the use of HMMs with 20 states would give optimal results. This decision, combined with the similarities between their analysis method and our own, prompted us to select similar values for the number of states.

In sections 4.3.1 and 4.3.2 we present results for $N = 10$, 20, and 30. By examining the results for identical trails tested under three different values of N, we can observe the effects of increasing or decreasing the number of states.

Comparing tables 4.5 and 4.6 we see that in the 10 state case a large number of trials have a lower % difference, as well as lower average difference magnitude and maximum difference magnitude. This is due to a lower "motion resolution" resulting from the reduced number of states. Having such a low number of states means that there are fewer state exemplars with which to assign frames. This means that a larger range of the motion will represented by a single exemplar, and thus some details of the motion may be lost.

Comparing tables 4.6 and 4.7 we observe that in general there is again an increase in the % difference, average difference magnitude, and maximum difference magnitude. The increase in the % difference would indicate that the increased motion resolution provided by a 30-state model allows for the detection of more subtle differences in the input sequences. However, the changes are not as pronounced as the changes between the 10 and 20-state cases, which suggests that there is a diminishing return as the value of N is increased beyond an optimal value. The changes in the average difference magnitude and maximum difference magnitude are also not as pronounced in this comparison, and can again be attributed to the incrased motion resolution, as an equivalent motion interval with be represented by a greater number of states in a 30-state model as opposed to a 20-state model.

When comparing the results for the speed dependence tests, our findings are the same as above. This is expected, as changes in the value of N should have similar results on trial tests regardless of which training set was used to create the HMM.

4.4.2 Detecting Motion Abnormalities

With the creation of standardized models, it becomes possible to analyze the results for specific input sequences in an attempt to detect motion abnormalities. Using the methods discussed in section 3.4, we attempted to detect sway events in a set of Viterbi paths. All input sequences had been manually labeled to indicate whether or not they possessed a sway event. These ground truth values were compared to the detection results of the algorithm, and the findings are given in tables 4.11,4.12, and 4.13.

From this data we can see that our proposed algorithm is able to detect sway events with an error rate between 0 and 13% for the case of $N = 20$ states. These results are promising, and indicate that through the use of standardized motion models it is possible to detect motion adnormalities with a high degree of accuracy.

When the number of states is decreased to $N = 10$ or increased to $N = 30$, we observe that there is very little change in the detection accuracy compared to the $N = 20$ case. This indicates that the number of states in the model has little or no effect on the detection of sway events.

It should be noted that these tests do not evaluate our algorithm's ability to provide quantitative information on the sway events when they are detected. This is because of the lack of ground truth data with which to compare the algorithm's

Motion	Subject	Correct Positive Detections	Correct Negative Detections	False Positive Detections	False Negative Detections	% Error
Pickup	1	14	198	0	2	1
	2	4	175	1	8	5
	3	2	251	7	12	7
	4	15	196	2	5	3
	5	0	178	0	0	0
Stepping	1	8	85	75	14	49
	2	8	180	0	10	5
	3	2	248	6	18	9
	4	6	165	1	12	7
	5	0	198	0	0	0
Sit-to-Stand	1	5	112	0	7	6
	2	6	140	0	6	4
	3	10	146	8	4	7
	4	8	126	0	6	4
	5	0	144	0	0	0

Table 4.11: Sway detection accuracy. ($N = 10$)

Motion	Subject	Correct Positive Detections	Correct Negative Detections	False Positive Detections	False Negative Detections	% Error
Pickup	1	16	198	0	0	0
	2	4	176	0	8	4
	3	10	248	10	4	5
	4	18	197	1	2	1
	5	0	178	0	0	0
Stepping	1	1	158	2	21	13
	2	4	172	8	14	11
	3	1	250	4	19	8
	4	6	166	0	12	7
	5	0	198	0	0	0
Sit-to-Stand	1	5	112	0	7	6
	2	6	140	0	6	4
	3	10	146	8	4	7
	4	10	126	0	4	3
	5	0	144	0	0	0

Table 4.12: Sway detection accuracy. ($N = 20$)

Motion	Subject	Correct Positive Detections	Correct Negative Detections	False Positive Detections	False Negative Detections	% Error
Pickup	1	15	198	0	1	1
	2	2	176	0	10	5
	3	11	250	8	3	4
	4	16	197	1	4	2
	5	0	178	0	0	0
Stepping	1	0	158	2	22	13
	2	1	180	0	17	9
	3	0	253	1	20	8
	4	5	166	0	13	7
	5	0	198	0	0	0
Sit-to-Stand	1	4	112	0	8	6
	2	6	140	0	6	4
	3	10	146	8	4	7
	4	8	126	0	6	4
	5	0	144	0	0	0

Table 4.13: Sway detection accuracy. ($N = 30$)

performance. Although it is relatively simple for a human operator to recognize the occurrence of a sway event, it is very difficult for them to provide correct and accurate measurements on the event based solely on visual data.

4.5 Validation & Discussion

In this section we discuss some of the limitations of our approach, compare our algorithm to other methods of analysis previously explored and used, and present a method of visualizing the analysis results in a way that makes the resulting data easy to interpret.

4.5.1 Limitations

An inherent limitation of single-camera human motion tracking is that movement perpendicular to the plane of view of the camera (i.e. motion directly towards, or directly away from, the camera lens) is difficult to detect. As a result of this, the motions analyzed using this approach must take place in the plane of view of the camera, or else there will be a loss of information which could lead to missed detection of abnormalities.

One method of compensating for this would be to expand this approach to a multi-camera system, where the motion of the subject can be viewed from two or more different viewpoints (i.e. front and side views).

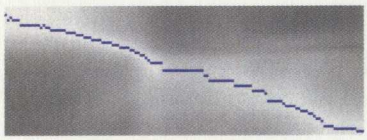
4.5.2 Comparison to Other Analysis Methods

The use of similarity matrices is common in computer vision for comparing one video sequence to another. Early in our research the idea of using similarity matrices for tracking of motion progression and detecting abnormalities was explored. It was later abandoned in favor of using HMMs, which were found to be more robust for modeling complex motions. Here we will discuss how similarity matrix-based analysis would work, and demonstrate its limitations with respect to the use of HMMs.

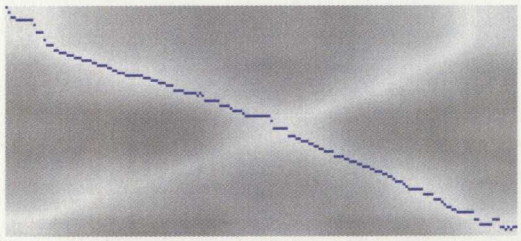
An L_1xL_2 similarity matrix is constructed for two video sequences (with lengths L_1 and L_2) by comparing each frame of the first video to every frame of the second, and calculating a "similarity score". The frames of one video are indexed along the rows of the matrix, and the frames of the second video are indexed along the columns.

The similarity score, as its name suggests, is a measure of how similar one frame is to another, as determined by a suitable formula. Cross-correlation is a commonly used measure to determine a similarity score between two binary silhouettes.

Often a similarity matrix is displayed visually by scaling and discretizing the similarity scores to values which can be formatted into a grayscale image (in MatLab the matrix can be converted to 8-bit unsigned integer values). In this way, a white element indicates the highest possible similarity between the paired frames, and black indicates the lowest possible similarity, with all other elements being expressed as shades of gray with a brightness relative to their corresponding similarity score.



(a) Sit-to-Stand



(b) Pickup

Figure 4.8: Similarity matrix comparing motions performed at "fast" speed (row indexed) and "slow" speed (column indexed). The blue path indicates matched frames calculated to represent equivalent points in the motions progression.

In figure 4.8b we see the characteristic "X" pattern in the high similarity values. This results whenever a motion has an inherent reversal of direction, as is the case with both the "pickup" and "stepping" motions. With a "pickup" motion, frames observed while the subject is bending downward look virtually identical to frames observed as the subject rises back upward, they simply appear in the reverse order.

Given two points A and B in the matrix, it is possible to calculate a "least cost" path between those points. This can be viewed as a more simplified approach to that of a HMM finding a "most likely state path". However, the least cost path calculated by a similarity matrix does not take any temporal dependence into account, as it lacks the information about transition probabilities that a HMM possesses. This can lead to incorrect or inaccurate results, as seen in figures 4.9.

In this case, the "true" path of equivalence lies on the main diagonal of the similarity matrix. The path finding algorithm in this example was a modified algorithm for determining the least-cost path, adjusted to also take into account optical-flow information gathered from the input video frames, and use that information to its decision on frame equivalence. Due to very high similarity scores in regions of the similarity matrix other than the main diagonal, as well as optical-flow information gathered, the algorithm incorrectly follows a different path.

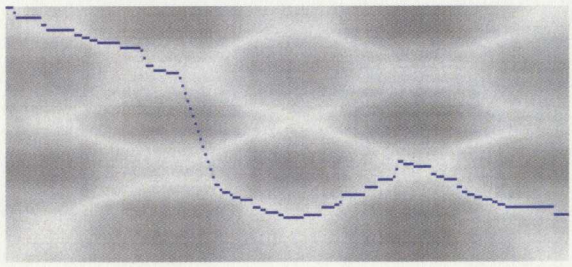


Figure 4.9: Similarity matrix for "double-step" motions (lifting one foot onto a pedestal, lowering it, then lifting the other foot onto the pedestal, and lowering it), performed at "fast" speed (row indexed) and "slow" speed (column indexed). The blue path indicates matched frames calculated to represent equivalent points in the motions progression.

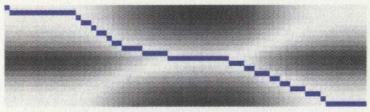
4.5.3 Visualizations

Although a similarity matrix is limited in its ability to model a motion, under the right conditions it can provide a useful means of visualizing the results obtained from HMM analysis.

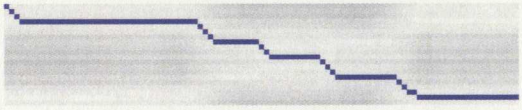
Instead of generating a similarity matrix in the usual way, by comparing frames from two distinct video sequence, we compare the frames of a video sequence (or more accurately, the processed binary silhouettes obtained from that video) with the state exemplars of the HMM that was used in the analysis. The Tanimoto similarity (Equation 3.5) is used to calculate the similarity scores. This results in an NxL similarity matrix which can be converted to an image format and displayed. State values are indexed along the rows of the matrix, and input video frames are indexed along the columns.

It is then possible to overlay the Viterbi path onto this image by coloring the appropriate matrix elements.

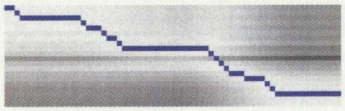
By visualizing the analysis data in this way, it is easier to interpret the Viterbi path and how it is obtained through the HMM analysis. Motion abnormalities can also be interpreted visually, as seen in figures 4.11 and 4.12.



(a) Pickup motion



(b) Stepping motion



(c) Sit-to-Stand motion

Figure 4.10: Similarity matrix visualization for the analysis of motions using a 20-state HMM. State values are indexed along the rows, while input frames are indexed along the columns. The Viterbi paths are displayed in blue.

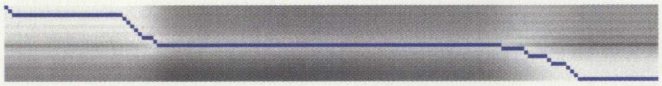


Figure 4.11: Similarity matrix visualization for the analysis of a "sit-to-stand" motion using a 20-state HMM. State values are indexed along the rows, while input frames are indexed along the columns. The Viterbi path is displayed in blue. This case demonstrates the visualization of a temporary stop in the motion.

Figure 4.12: Similarity matrix visualization for the analysis of a "sit-to-stand" motion involving a sway event using a 20-state HMM. State values are indexed along the rows, while input frames are indexed along the columns. The Viterbi path is displayed in blue.

Chapter 5

Conclusion

5.1 Summary

In this study, we have proposed an original method for marker-less human motion analysis with the goal of analyzing and evaluating motion performance. Building on previous research in the field of gait analysis, our work expands on the idea of creating standardized motion models to include several motions commonly used by physiotherapists in the diagnosis of injuries and other physical problems.

Our approach involves the creation of a standardized model for each motion, which is then used as a basis of comparison for input motion sequences. The comparison results were then analyzed further, in an attempt to detect motion abnormalities and quantitatively measure them.

Using a marker-less based approach to motion analysis has several advantages. It is much less costly than expensive marker-based motion tracking systems. It is more portable, requiring only a single camera instead of the 6 to 12 cameras required by Vicon and other similar tracking systems, which must be synchronized and calibrated before data acquisition can take place. Marker-less tracking is also less distracting and intrusive to the subject, as there is no need to wear numerous markers or other specialized sensory equipment. This is of particular importance when considering subjects who are elderly and/or physically impaired, as the distraction of wearing markers or other equipment may cause undo stress, or further limit their mobility, influencing analysis results.

Testing of our algorithm required the creation of a specialized database of human motion video sequences. Several motions were needed, and each of those motions had

to be recorded under a variety of conditions, including differences in speed, and the occurrence of motion abnormalities.

The computational efficiency of the algorithm was found to be quite good, with the analysis of input sequences averaging roughly 20 seconds per trial. Abnormalities in the form of motion sways are detected with an accuracy of 87-100%. These results are promising, and indicate that our work could lead to the development of new diagnostic tools for clinical use, as the system evolves to detect a wider range of motion abnormalities.

5.2 Future Work

There are many avenues of research still left to explore in this area. Although initial results are very promising, testing and validation should be expanded to further evaluate the system's effectiveness and discover any unforeseen limitations.

Expanding the motion sequence database to include additional motion abnormalities is vital to assessing the usefulness of the system. An increased level of collaboration with medical professionals would help to ensure that the motions and abnormalities included in the database are of the most use. Expanding the population of subjects to include more variation in age, body type, physical fitness, physical injuries or impairments, etc. would allow for realistic training and testing of the standardized motion models.

Further validation of experimental results will be possible if marker-based tracking is employed during recording of database sequences. By acquiring motion tracking data using both our system, and a marker-based approach such as a Vicon system, we will be better able to compare our results to other analysis methods.

Appendix A

Additional Information

A.1 Code

A.1.1 Background Subtraction

```
inputs: - Background video
        - input video



set [M,N] = [height, width] of Background video frames

for( each pixel location (i,j) s.t. 1<i<M  1<j<N )
{
  calculate average and standard deviation of pixel (i,j) for each image layer
                        (hue,saturation,value) of every Background video frame
}

let ColorThresh = 2 and BrightnessThresh = 0.6 //these parameters are
                                               //adjusted for each sequence
                                               //if background subtraction
                                               //results are not satisfactory.

for(each frame F of the input video)
```

```
{
  for(every pixel (i,j) of frame F)
  {
    let H = Hue of pixel (i,j)
    let S = Saturation of pixel (i,j)
    let V = Value of pixel (i,j)

    if((H < Hue_ave(i,j)+ColorThresh*Hue_std(i,j)) and
       (H > Hue_ave(i,j)-ColorThresh*Hue_std(i,j)) and
       (S < Sat_ave(i,j)+ColorThresh*Sat_std(i,j)) and
       (S > Sat_ave(i,j)-ColorThresh*Sat_std(i,j)))
    {
       //colors within range, check brightness

       if( (V < Val_ave(i,j)+BrightnessThresh*Val_std(i,j))and
           (V > Val_ave(i,j)-BrightnessThresh*Val_std(i,j)))
       {
          //brightness in range -> set as background

          NewImage(RowNum,ColNum) = 0;
       }
       elseif( (V < Val_ave(i,j)-BrightnessThresh*Val_std(i,j)) )
       {
          //colors in range, but brightness lower
          //-> background in shadow -> set as background.

          NewImage(i,j) = 0;
       }
       else
       {
          //colors in range, but brightness higher -> set as foreground.

          NewImage(i,j) = 1;

       }
```

```
    }
    else
    {
       //colors don't match -> set as foreground

       NewImage(i,j) = 1;
    }
  }
}

return(background subtracted frames)  //binary images
```

A.1.2 Noise Removal

```
inputs: - Background subtracted video frames

let ComponentThreshold = 250 //used for deleted erroneous pixel regions

for(every frame F)
{
  let SE = [0 1 0 ; 0 1 0 ; 0 0 0]  // structuring element
  erode image using SE
  dilate image using SE

  let SE = [0 0 0 ; 0 1 1 ; 0 0 0]
  erode image using SE
  dilate image using SE

  let SE = [0 1 0 ; 1 1 1 ; 0 1 0]
  dilate image using SE
  erode image using SE
```

```
  //the above code eliminates most salt & pepper noise
  //now we must remove large erroneous pixel groups

  erode image 5 times using SE

  determine all connected components in image //MatLab function bwlabel()

  for(every connected component)
  {
    calculate area of component
    if(area < ComponentThreshold)
    {
      set all pixels in component to 0 //label component as background
    }
  }

  dilate image 5 times using SE //reverse previous erosions
}

return(binary silhouette images) //with noise removed
```

A.1.3 Rescaling & Alignment

```
inputs: - Binary silhouette images

for(each Binary silhouette image)
{
  determine minimum width & height for a bounding box to contain silhouette

  crop image to include only the bounding box region of the image

  rescale image to be of height 128 pixels     //MatLab function imresize()
```

```
    //maintains aspect ratio
  calculate COM of silhouette

  append columns of black pixels (background) to left and right edges of
       image until image width = 257, keeping the silhouette COM
       centered in the image
}

return(rescaled and aligned silhouette images)
```

A.1.4 HMM Training

```
inputs: - Set of K sequences (consisting of rescaled & aligned binary silhouettes)
to be used for training
- N, the number of states to be used in the trained HMM

//create initial clustering
for(each sequence in training set)
{
  divide sequence into an ordered list of N equally sized groups of frame
   silhouettes
}

//update clustering iteratively
while(clusters not converged)
{
  //calculate centroids
  for(each group, G_n, in clustering)  //  1 < n < N
  {
    let S = the set of all frames in group G_n of every training sequence
    let group centroid, C_n = average(all silhouettes in S)
  }
```

```
  //update clusters
  for(each training sequence)
  {
    for(each group, G_n, in clustering)
    {
      compare endframes of group to the closest adjacent group's centroid
               using Tanimoto distance measure
      if(endframe is closer to adjacent group's centroid than its current
           group centroid)
      {
        move endframe to the adjacent cluster group
      }
    }
  }
  if(no frames moved from one group to another)
  {
    clusters have converged
  }
  else
  {
    clusters not converged
  }
}

let the set of Exemplars = the most recent set of cluster group centroids, C_n


//calculate Mu values according to equation 3.7
for( j = 1 to N)
{
    Exemplar = E_j;
    SumOfDistances = 0;
    SizeOfCluster = 0;
    for(k = 1 to K)
    {
```

```
        retrieve silhouettes for group G_j of sequence k
        for(i = 1 to size(G_j))
        {
            Distance = TanimotoDistance(Frame,Exemplar);
            SumOfDistances = SumOfDistances + Distance;
            SizeOfCluster = SizeOfCluster +1;
        }
    }

    Mu(j) = SumOfDistances/SizeOfCluster;
    ClusterSizes(j) = SizeOfCluster;  //for later use
}


//calculate emmission values
for(k = 1 to K)
{
    for( j = 1 to N)
    {
        t = 1;
        //get current Exemplar
        Exemplar = E_j
        for(ClusterNum = 1 to N)
        {

            //get frames for current cluster
            retrieve silhouettes for group G_j of sequence k
            for i =1:length(FrameIndeces)
            {
                Distance = TanimotoDistance(Frame,Exemplar);
                Emissions{k,j,t} = (1/Mu(j))*exp(-1*(Distance/Mu(j)));
                t = t+1;
            }
        }
    }
```

```
}


//calculate estimate for transition matrix
A = zeros(N,N);
for(i = 1 to N)
{
    for(j = 1 to N)
    {
        Sum = 0;
        ClusterSize = 0;
        for(k = 1 to K)
        {
            //Get cluster groups
            Ei = silhouette group i for sequence k
            Ej = silhouette group j for sequence k

            for(index = 1 to length(Ei))
            {
                f_t = Ei(index);
                if(ismember((f_t + 1), Ej))
                {
                    Sum = Sum + 1;
                }
            }

            ClusterSize = ClusterSize + ClusterSizes(i);

        }
        A(i,j) = Sum / ClusterSize;
    }
}


//update transition probabilities
```

```
while(Transitions not converged)
{
    //generate forward / backward probabilities and P_k ( =P(k) )

    //Calculate Alpha
    for(k = 1 to K)
    {
        //calculate t=1 case: Alpha{k,1,j} = b_j(f_1)/NumStates
        for(j = 1 to N)
        {
            Alpha{k,1,j} = Emissions{k,j,1};
        }

        //calc remaining cases
        for(t = 2 to length of sequence k)
        {
            for(j = 1 to N)
            {
                Sum = 0;
                for(i = 1 to N)
                {
                    Value = Alpha{k,t-1,i}*A(i,j)*Emissions{k,j,t};
                    Sum = Sum + Value;
                }
                Alpha{k,t,j} = Sum; //assign value
            }
        }

    }

    //Calculate P_k

    for(k = 1 to K)
    {
        Sum = 0;
```

```
    Tk = length(training sequence k)
    for(j = 1:N)
    {
        Sum = Sum + Alpha{k,Tk,j}; //Sum T_k column of alpha

    }
    P(k) = Sum;  //assign value
}

//Calculate Beta
for(k = 1 to K)
{
    //calc t = T_k case
    Tk = TrainingLengths(k);
    for(i =1:N)
    {
        Beta{k,Tk,i} = 1;
    }

    //Calc remaining cases
    for(t = (Tk-1) down to 1)
    {
        for(i = 1:N)
        {
            Sum = 0;
            for(j = 1 to N)
            {
                Value = A(i,j)*Emissions{k,j,t+1}*Beta{k,t+1,j};
                Sum = Sum + Value;
            }
            Beta{k,t,i} = Sum; //assign value
        }
    }
}
```

```
//Calculate Next iteration of 'A'
//equation (7) of referenced paper

for(i = 1 to N)
{
    for(j = 1 to N)
    {
        Numerator = 0;
        Denominator = 0;
        for(k = 1 to K)
        {
            //calculate Numerator
            Pk_inv = 1/P(k);
            Tk = length(training sequence k)
            N_Sum = 0;
            D_Sum = 0;
            for(t = 1 to (Tk-1))
            {
                //numerator
                N_Value = Alpha{k,t,i}*A(i,j)*Emissions{k,j,t+1}*Beta{k,t+1,j}
                N_Sum = N_Sum + N_Value;
                //denominator
                D_Value = Alpha{k,t,i}*Beta{k,t,i};
                D_Sum = D_Sum + D_Value;
            }
            Numerator = Numerator + (Pk_inv*N_Sum);
            Denominator = Denominator + (Pk_inv*D_Sum);
        }
        Next_A(i,j) = Numerator/Denominator; //assign value
    }
}

//determine Convergence
```

```
    A_Difference = abs(A-Next_A);
    Max_Diff = max(max(A_Difference));
    if(Max_Diff < ConvergenceLimit)
        //convergence reached
        TransitionConverged = 1;
    end;

    //Update 'A'
    A = Next_A;

}


//modify Transition Matrix to allow for motion reversals
for(i = 2 to N)
{
    ModifyAmount = A(i-1,i)/1000;
    A(i,i-1) = A(i,i-1) + ModifyAmount; //raise Prob(move back) slightly
    A(i,i) = A(i,i) - ModifyAmount; //lower Prob(no change) by same
    if(A(i,i) <= 0)
    {
        A(i,i) = 0.0001; //Do not let stationary probability go to zero
    }
}
```

A.1.5 Viterbi Decoding

The Viterbi algorithm is a well known method for deocding HMM state sequences.

The following code was used for calculating the Viterbi path, it can be obtained from http://www.igi.tugraz.at/lehre/CI/tutorials/MixtGaussian/MixtGaussian.pdf

```
function [path, loglik] = viterbi_path(prior, transmat, obslik)
% VITERBI Find the most-probable (Viterbi) path through the HMM state trellis.
% path = viterbi(prior, transmat, obslik)
%
% Inputs:
% prior(i) = Pr(Q(1) = i)
% transmat(i,j) = Pr(Q(t+1)=j | Q(t)=i)
% obslik(i,t) = Pr(y(t) | Q(t)=i)
%
% Outputs:
% path(t) = q(t), where q1 ... qT is the argmax of the above expression.


% delta(j,t) = prob. of the best sequence of length t-1 and then going to state j,
% and O(1:t)
% psi(j,t) = the best predecessor state, given that we ended up in state j at t

scaled = 1;

T = size(obslik, 2);
prior = prior(:);
Q = length(prior);

delta = zeros(Q,T);
psi = zeros(Q,T);
path = zeros(1,T);
scale = ones(1,T);


t=1;
```

```
delta(:,t) = prior .* obslik(:,t);
if scaled
  [delta(:,t), n] = normalise(delta(:,t));
  scale(t) = 1/n;
end
psi(:,t) = 0; % arbitrary value, since there is no predecessor to t=1
for t=2:T
  for j=1:Q
    [delta(j,t), psi(j,t)] = max(delta(:,t-1) .* transmat(:,j));
    delta(j,t) = delta(j,t) * obslik(j,t);
  end
  if scaled
    [delta(:,t), n] = normalise(delta(:,t));
    scale(t) = 1/n;
  end
end
[p, path(T)] = max(delta(:,T));
for t=T-1:-1:1
  path(t) = psi(path(t+1),t+1);
end

% loglik = log p.
% If scaled==0, p = prob_path(best_path)
% If scaled==1, p = Pr(replace sum with max and proceed as in the scaled forwards
%    algo)
% loglik computed by the two methods will be different, but the best path will be
%    the same.

if scaled
  loglik = -sum(log(scale));
  %loglik = prob_path(prior, transmat, obslik, path);
else
  loglik = log(p);
end
```

A.1.6 Abnormality Detection

```
function Sways = AnalyzeHMMData(Vit_Path)


//analyze State Path


DurationIndex = 1;
CurrState = 1;
Index = 1;
Duration = 0;
while(Index <= length(Vit_Path))
{
    if(Vit_Path(Index) == CurrState)
    {
        //state unchanged
        Duration = Duration + 1;
        Index = Index + 1;
    }
    else
    {
        //state changed
        StateDurationMap(DurationIndex,1) = CurrState;
        StateDurationMap(DurationIndex,2) = Duration;

        DurationIndex = DurationIndex + 1;
        CurrState = Vit_Path(Index);
        Duration = 1;
        Index = Index + 1;
    }
}
```

```
StateDurationMap(DurationIndex,1) = CurrState;
StateDurationMap(DurationIndex,2) = Duration;

//StateDurationMap

StateDurations = StateDurationMap(:,2); //durations only


//scan for sways

Sway_index = 1;
CurrState = StateDurationMap(1,1);
StepNum = 1;
while(StepNum <= NumSteps)
{
    if(StateDurationMap(StepNum,1) >= CurrState)
    {
        //no sway
        CurrState = StateDurationMap(StepNum,1);
        StepNum = StepNum + 1;
    }
    else
    {
        //reversal - start of sway event

        StartState = CurrState;
        LowestState = StateDurationMap(StepNum,1);
        SwayDuration = StateDurationMap(StepNum,2);


        StepNum = StepNum + 1;
        if(StepNum <= NumSteps)
        {
            CurrState = StateDurationMap(StepNum,1);
        }
```

```
        while((CurrState < StartState)and(StepNum <= NumSteps))
        {
            //still in sway event -> update Lowest & duration

            if(CurrState < LowestState)
            {
                LowestState = CurrState;
            }
            SwayDuration = SwayDuration + StateDurationMap(StepNum,2);
            StepNum = StepNum + 1;
            if(StepNum <= NumSteps)
            {
                CurrState = StateDurationMap(StepNum,1);
            }

        }
        Sways(Sway_index,:)=[StartState,SwayDuration,(StartState-LowestState)];
        Sway_index = Sway_index + 1;

    }

}
```

Bibliography

[1] E. W Abel, A. Unger, R. Fletche, and A.S Jain. Development of clinical measurement of the axes of rotation of the ankle and subtalar joints. *Proceedings of the Second Joint EMBS/BMES Conference*, pages 2455–2456, Oct 2002.

[2] H. Akaike. Information theory as an extension of the maximum likelihood principle. *Proceedings of hte Second International Symposium on Information Theory*, pages 267–281, 1973.

[3] C. Bauchage, J.K. Tsotsos, , and F.E. Bunn. Detecting abnormal gait. *Proceedings of the 2nd IEEE Canadian Conference on Computer and Robot Vision*, pages 282–288, 2005.

[4] K. Berg, S. Wood-Dauphinee, and J.I. Williams. The balance scale: reliability assessment for elderly residents and patients with an acute stroke. *Scandinavian Journal of Rehabilitation Medicine*, 27:27–36, 1995.

[5] Aude Billard, Yann Epars, Sylvain Calinon, Stefan Schaal, and Gordon Cheng. Discovering optimal imitation strategies. *Robotics and Autonomous Systems*, 47:69–77, 2004.

[6] Statistics Canada. Population projections for canada, provinces and territories. http://www.statcan.gc.ca/ads-annonces/91-520-x/index-eng.htm.

[7] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani. Probabilistic posture classification for human behaviour analysis. *IEEE Transactions on Systems, Man, Cybernetics*, 35(1):42–54, 2005.

[8] J. Cutting and L. Kozlowski. Recognizing friends by their walk: gait perception without familiarity cues. *Bulletin of the Psychonomic Society*, 9:353–356, 1977.

[9] Ahmed El-Bialy. Towards a complete computer dental treatment system. *Proceedings of the International Conference on Biomedical Engineering*, 2008.

[10] P.J. Elliot, J.M. Knapman, and W. Schlegel. Interactive image segmentation for radiation treatment planning. *IBM Systems Journal*, 31(4):620–634, 1992.

[11] Jiang Gao, Alexander G. Hauptmann, Ashok Bharucha, and Howard D. Wactlar. Dining activity analysis using a hidden markov model. *17th International Conference on Pattern Recognition*, Aug 2004.

[12] C. Graetzel, T. Fong, S. Grange, and C. Baur. A non-contact mouse for surgeon-computer interaction. *Technology and Health Care*, 12(3), 2004.

[13] R. Gross and J. Shi. The cmu motion of body (mobo) database. *Technical Report CMU-RI-TR-01-18*, June 2001.

[14] Gregoire Guetat, Matthieu Maitre, Laur'ene Joly, Sen-Lin Lai, Tzumin Lee, and Yoshihisa Shinagawa. Automatic 3-d grayscale volume matching and shape analysis. *IEEE Transactions on Information Technology in Biomedicine*, 10(2):362–376, Apr 2006.

[15] A. Kale, A. Sundaresan, A.N. Rajagopalan, N.P. Cuntoor, A.K. Roy-Chowdhury, V. Kruger, and R. Chellappa. Identification of humans using gait. *IEEE Transactions on Image Processing*, 13:1163–1173, Sept 2004.

[16] Kang Li, Eric D. Miller, Mei Chen, Takeo Kanade, Lee E. Weiss1, and Phil G. Campbell. Computer vision tracking of stemness. *International Symposium on Biomedical Imaging*, pages 847–850, 2008.

[17] Xiaohui Liu and Chin-Seng Chua. Multi-agent activity recognition using observation decomposed hidden markov models. *Image and Vision Computing*, 24:166–175, 2006.

[18] Zongyi Liu and Sudeep Sarkar. Improved gait recognition by gait dynamics normalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):863–876, June 2006.

[19] S. J. Mckenna and H.N. Charif. Summarising contextual activity and detecting unusual inactivity in a supportive home environment. *Pattern Analysis and Applications*, 7:386–401, 2005.

[20] A. Mihailidis, B. Carmichael, and J. Boger. The use of computer vision to support aging-in-place, safety, and independence in the home. *IEEE Transactions on Information Technology in Biomedicine*, 8(3):238–247, 2004.

[21] T.B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81:231–268, 2001.

[22] T.B. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based motion capture and analysis. *Computer Vision and Image Understanding*, 104:90–126, 2006.

[23] A. Nishikawa, T. Hosoi, and K. Koara et al. Face mouse: a novel human-machine interface for controlling the position of a laparoscope. *IEEE Trans. on Robotics and Automation*, 19(5):825–844, 2003.

[24] D. Ormoneit, H. Sidenbladh, M. J. Black, and T. Hastie. Learning and tracking cyclic human motion. *Proceedings of Neural Information Processing Systems*, 2000.

[25] Eun Sook Park, Chang il Park, Hong Jae Lee, Deog Young Kim, Don Shin Lee, and Sung-Rae Cho. The characteristics of sit-to-stand transfer in young children with spastic cerebral palsy based on kinematic and kinetic data. *Gait and Posture*, 17:43–49, 2003.

[26] P.J. Phillips, S. Sarkar, I. Robledo, P. Grother, and K. Bowyer. The gait identification challenge problem: Data sets and baseline algorithm. *Proceedings of the International Conference on Pattern Recognition*, 2002.

[27] L. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition.* Prentice Hall, 1993.

[28] Sreesha S. Rao, Emest L. Bontrager, JoAnne K. Gronley, Craig J. Newsam, and Jacquelin Perry. Three-dimensional kinematics of wheelchair propulsion. *IEEE Transactions on Rehabilitation Engineering*, 4(3):152–160, Sept 1996.

[29] Barbara Resch. Hidden markov models: A tutorial for the course *Computational Intelligence.* http://speech.tifr.res.in/tutorials/hmmTutExamplesAlgo.pdf.

[30] Barbara Resch. Mixtures of gaussians: A tutorial for the course *Computational Intelligence*. http://www.igi.tugraz.at/lehre/CI/tutorials/MixtGaussian/MixtGaussian.pdf.

[31] Neil Robertson and Ian Reid. A general method for human activity recognition in video. *Computer Vision and Image Processing*, 104:232–248, 2006.

[32] L. Rubenstein and P. Trueblood. Gait and balance assessment in older persons. *Annals of Long-Term Care: Clinical Care and Aging*, 12(2):39–45, 2004.

[33] Cristian Sminchisescu, Atul Kanaujia, Zhiguo Li, and Dimitris Metaxas. Conditional models for contextual human motion recognition. *International Conference on Computer Vision*, 2005.

[34] Alexei Sourin, Olga Sourina, and Howe Tet Sen. Virtual orthopedic surgery training. *IEEE Computer Graphics and Applications*, pages 6–9, 2000.

[35] Danail Stoyanov, Mohamed ElHelw, Benny P Lo, Adrian Chung, Fernando Bello, and Guang-Zhong Yang. Current issues of photorealistic rendering for virtual and augmented reality in minimally invasive surgery. *Proceedings of the Seventh International Conference on Information Visualization*, 2003.

[36] Aravind Sundaresan, Amit RoyChowdhury, and Rama Chellappa. A hidden markov model based framework for recognition of humans from gait sequences. *IEEE International Conference on Image Processing*, 2003.

[37] M. Tinetti. Preventing falls in elderly persons. *New England Journal of Medicine*, 348(1):29–42, 2003.

[38] Ashok Veeraraghavan, Amit K. Roy-Chowdhury, and Rama Chellappa. Matching shape sequences in video with applications in human movement analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1896–1909, Dec 2005.

[39] L. Wang. Abnormal gait analysis using silhouette-masked flow histograms. *Proceedings of the IEEE International Conference on Pattern Recognition*, Aug 2006.

[40] A. Wilson and A. Bobick. Using hidden markov models to model and recognize gesture under variation. *International Journal on Pattern Recognition and Ar-*

tificial Intelligence (special issue on Hidden Markov Models in computer vision, 2000.

[41] Christopher R. Wren, Brain P. Clarkson, and Alex P. Pentland. Understanding purposeful human motion. *4th IEEE International Conference on Automatic Face and Gesture Recognition*, 2000.

[42] Sumanth Yalamanchili, Rami Abboud, and Weijie Wang. A model to calculate the joint movements and forces in the foot. *The Ninth International Conference on Electronic Measurement & Instruments*, 4:532–536, 2009.

[43] N. F. Yang, D. W. Jin, M. Zhang, C.H. Huang, and R.C. Wang. A function description for the human upper limb pointing movements performance. *Proceedings of the 23rd Annual EMBS International Conference*, pages 1236–1239, Oct 2001.