**MUMS:**

**a Measure of hUman Motion Similarity**

by

Francisco J. Torres Reyes

Bachelor of Computer Engineering, Universidad Autónoma de San Luis Potosí, 1999

A dissertation submitted to the Graduate Faculty of the

University of Colorado at Colorado Springs

in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2014

© Copyright by Francisco J. Torres Reyes 2014

All Rights Reserved

This dissertation for

Doctor of Philosophy Degree by

Francisco J. Torres Reyes

has been approved for the

Department of Computer Science

By

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. C. Edward Chow, Chair

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Terry Boult

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. James Carollo

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Rory Lewis

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Xiaobo Zhou

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date

Francisco J. Torres Reyes (Ph.D. in Computer Science)

MUMS: A Measure of Human Motion Similarity

Dissertation directed by Professor Edward Chow

The analysis of motion similarity, particularly human motion similarity, is needed in different areas of study: motion blending, where new motions are generated from previous ones and they are intended to be as realistic as possible; motion retrieval, where indexing, searching and retrieving a particular movement from databases of motions capture data is required; and performance analysis of dancers and athletes, where the examination of recorded dances and exercises allows to track the evolution of characteristics to be analyzed, such as strength, speed, etc.

This dissertation offers a framework for measuring human motion similarity by modeling human motion as a set of 3-dimensional curves represented as orthogonal changes of direction, and then by using a human movement notation that describes such human motion in a way that temporal and spatial analysis of human motion similarity can be achieved.

For purposes of evaluating the feasibility of this approach, a set of baseline key rehabilitation exercises has been chosen and tested using our implementation. Motion capture sessions for the key rehabilitation exercises provided the data for the experiments. FastDTW, an algorithm for measuring similarity between two temporal sequences, was used to compare the result of our implementation.

One of the main contributions of this proposal is the modeling of human motion as chain codes, or strings composed of characters from a finite alphabet. This model of human motion allows the use of string matching algorithms, sequence alignment algorithms, and statistical analysis approaches to achieve the analysis of similarity. Another contribution is the ability of spatial and temporal analysis due to the proposed model and description of the human motion. This technique takes data from a motion capture session, regardless the technique used in those sessions. The only requirement is that data must contain timed three-dimensional positions of the markers used, and information regarding the part of the body those markers were set during the motion capture session. Finally, based on the description of the key rehabilitation exercises, we suggested enhancements for LABANotation such purpose.

***Many thanks to the National Council of Science and Technology -Consejo Nacional de Ciencia y Tecnología (CONACyT), México-***

Acknowledgements

First I would like to thank my parents. It is amazing how easier is to take the burden of life when they encourage you to do so. Second, all my love and gratitude to my wife. I am so blessed that I have had her as a mate for this journey, and for the remaining ones.

I want to thank so much my advisory committee members: Dr. Boult, Dr. Carollo, and Dr. Lewis. Their patience and support was amazing.

Patricia Rea is another person I want to thank. Her patience and guidelines were invaluable. I still owe her a bottle of tequila.

I want to thank Dr. Yunyu Wang, and Susan Kanal for selecting the four key rehabilitation exercises used in this research.

Thanks to The Children’s Hospital in Denver, CO, especially Dr. James Carollo, for providing the motion capture data for those key rehabilitation exercises.

Professors and friends both from here and from Mexico, thank you for all your support.

And finally, Dr. Edward Chow and Dr. Xiaobo Zhou: I am in your debt. This was a really long journey and I know it couldn’t be a successful one without your help and guidance. From my heart, my deepest thanks.

*¡Muchas gracias, Señor!*

**Table of Contents**

[Chapter 1 1](#_Toc356372687)

[1.1 Introduction 1](#_Toc356372688)

[1.2 Human Motion Overview 2](#_Toc356372689)

[1.3 Framework for Human Motion similarity analysis 3](#_Toc356372690)

[1.4 Dissertation Outline 7](#_Toc356372691)

[Chapter 2 Proposed Framework for Measuring and Analyzing Human Motion Similarity …. 9](#_Toc356372692)

[Chapter 3 Case of Study: Rehabilitation Therapy 58](#_Toc356372696)

[3.1 Key Rehabilitation Exercises 60](#_Toc356372697)

[3.2 Experiments and Results 62](#_Toc356372698)

[Chapter 4 Comparison of Timed 3D ChainCode with FastDTW 71](#_Toc356372700)

[4.1 Dynamic Time Warping 73](#_Toc356372701)

[4.2 3D ChainCode and FastDTW Comparison Results – Sample Sequences 81](#_Toc356372702)

[4.3 3D ChainCode and FastDTW Comparison Results – Shoulder Elevation and Rotation Key Rehabilitation Excercise 96](#_Toc356372703)

[Chapter 5 Enhancing LABANotation for Rehabilitation Specification 108](#_Toc356372704)

[5.1 Proposed Enhancements Based on Mini-Squats Exercise 109](#_Toc356372705)

[5.2 Proposed Enhancements Based on Standing-Hip Abduction Exercise 111](#_Toc356372706)

[Chapter 6 Design of Human Motion Tracking System Architecture for Rehabilitation purposes …… 118](#_Toc356372707)

[6.1 Human Motion Tracking System Architecture for Rehabilitation 120](#_Toc356372708)

[6.2 Blocks and Modules Description 122](#_Toc356372709)

[6.3 Scenarios 124](#_Toc356372710)

[6.4 Suggested Enhancements to LabanDancer Software 132](#_Toc356372711)

[Chapter 7 Future Research and Conclusions 151](#_Toc356372713)

[7.2 Background 152](#_Toc356372714)

[7.3 Approaching Security from a Software Perspective 154](#_Toc356372715)

[7.4 Principles into Metrics 159](#_Toc356372716)

[7.5 Comparison 168](#_Toc356372717)

[Works Cited 221](#_Toc356372739)

Appendix A - Algorithm for mapping 3D data sets into chain codes representing orthogonal changes of direction………………………………………………………134

Appendix B - Algorithms for the analysis of human motion similarity………………..135

**List of Tables**

[Table 3.1 – Human Motion Similarity values for the four Key Rehabilitation Exercises](#_Toc356372744)

[using different step sizes 25](#_Toc356372744)

[Table 3.2 – Human motion dissimilarity values for posterior pelvic tilt exercise](#_Toc356372745)

[using different step sizes 87](#_Toc356372745)

[Table 3.3 – Results comparison between Bribiesca's work and our implementation 104](#_Toc356372746)

[Table 4.1 – Sets of sensors for each part of the body 144](#_Toc356372747)

**List of Figures**

[Figure 1.1 – Anatomical Reference Axes and Planes 14](#_Toc356372754)

[Figure 1.2 – Planar Movements 37](#_Toc356372755)

[Figure 1.3 – Motion of Objects Classification 41](#_Toc356372756)

[Figure 1.4 – Motion Capture Session 44](#_Toc356372757)

[Figure 1.5 – Human Movement Notations: a) Eshkol-Wachman, b) Benesh, c) Laban 79](#_Toc356372758)

[Figure 1.6 – Laban Effort Graph 83](#_Toc356372759)

[Figure 1.7 – LABANotation: Direction of Movement 83](#_Toc356372760)

[Figure 1.8 – LABANotation: Parts of the Body 85](#_Toc356372761)

[Figure 1.9 – LABANotation: Level of Movement 85](#_Toc356372762)

[Figure 1.10 – LABANotation: Time 97](#_Toc356372763)

[Figure 1.11 – LABANotation: Score 99](#_Toc356372764)

[Figure 1.12 – Human Motion Model: Skeleton 100](#_Toc356372765)

[Figure 1.13 – 3D Curves Generated by Markers 101](#_Toc356372766)

[Figure 1.14 – Chain Code Representation for a 2D Figure 103](#_Toc356372767)

[Figure 1.15 – Five Possible Changes of Direction for Representing 3D Curves 105](#_Toc356372768)

[Figure 1.16 – Chain Code Representation for a 3D Curve 112](#_Toc356372769)

[Figure 1.17 – Chain Code Representation is Invariant under Translation and Rotation 113](#_Toc356372770)

[Figure 2.1 – Proposed Framework for the Analysis of Human Motion Similarity 114](#_Toc356372771)

[Figure 2.2 – Chain Code Generation per Marker, per Sampling Rate 115](#_Toc356372772)

[Figure 3.1 – LABANotation: Mini-Squats 116](#_Toc356372773)

[Figure 3.2 – LABANotation: Posterior Pelvic Tilt 161](#_Toc356372774)

[Figure 3.3 – LABANotation: Shoulder Elevation with Rotation 185](#_Toc356372775)

[Figure 3.4 – LABANotation: Standing Hip Abduction 187](#_Toc356372776)

[Figure 3.5 – Human Motion Similarity for Mini-Squat Exercise 201](#_Toc356372777)

[Figure 3.6 – Human Motion Similarity for Posterior Pelvic Tilt Exercise 205](#_Toc356372778)

[Figure 3.7 – Human Motion Similarity for Standing Hip Abduction Exercise 236](#_Toc356372779)

Figure 3.8 – Accumulated Direction for a Given 3D Discrete Curve 236

Figure 3.9 – Human Motion Dissimilarity for Posterior Pelvic Tilt Exercise 236

Figure 3.10 – Human Motion Similarity for Posterior Pelvic Tilt Exercise 236

Figure 4.1 – A Warping between Two Time Series 236

Figure 4.2 – A Cost Matrix with the Minimum-Distance Warp Path traced through it 236

Figure 4.3 – Location for each Set of Sensors on the Human Body 236

Figure 4.4 – LABANotation for Shoulder Elevation and Rotation exercise 236

Figure 4.5 – Slow start vs. Fast pace, slope. Chain Codes and Similarity values 236

Figure 4.6 – Slow start vs. Fast pace, slope. Trajectory A 236

Figure 4.7 – Slow start vs. Fast pace, slope. Trajectory B 236

Figure 4.8 – Slow start vs. Fast pace, horizontal. Chain Codes and Similarity values 236

Figure 4.9 – Slow start vs. Fast pace, horizontal. Trajectory A 236

Figure 4.10 – Slow start vs. Fast pace, horizontal. Trajectory B 236

Figure 4.11 – Slow start, rotated 90o counter clockwise. Chain Code and Similarity values 236

Figure 4.12 – Slow start, rotated 90o counter clockwise. Trajectory A 236

Figure 4.13 – Slow start, rotated 90o counter clockwise. Trajectory B 236

Figure 4.14 – Slow start, rotated 180o counter clockwise. Chain Code and Similarity values 236

Figure 4.15 – Slow start, rotated 180o counter clockwise. Trajectory A 236

Figure 4.16 – Slow start, rotated 180o counter clockwise. Trajectory B 236

Figure 4.17 – Slow start, rotated 270o counter clockwise. Chain Code and Similarity values 236

Figure 4.18 – Slow start, rotated 270o counter clockwise. Trajectory A 236

Figure 4.19 – Slow start, rotated 270o counter clockwise. Trajectory B 236

Figure 4.20 – Motion Capture Session for Shoulder Elevation and Rotation Exercise 236

Figure 4.21 – Session 1 vs. Session 2 – 3D ChainCode 236

Figure 4.22 – Session 1 vs. Session 3 – 3D ChainCode 236

Figure 4.23 – Session 2 vs. Session 3 – 3D ChainCode 236

Figure 4.24 – Comparison of All Motion Sessions for Full Body and Arms - 3D ChainCode 236

Figure 4.25 - Session 1 vs. Session 2 – FastDTW 236

Figure 4.26 - Session 1 vs. Session 3 – FastDTW 236

Figure 4.27 - Session 2 vs. Session 3 – FastDTW 236

Figure 4.28 – Comparison of All Motion Sessions for Full Body and Arms - FastDTW 236

Figure 5.1 – Mini-Squat Key Rehabilitation Exercise 236

Figure 5.2 – LABANotation for ‘feet apart, equal distribution in both legs’ 236

Figure 5.3 – Proposed LABANotation for ‘feet shoulder width apart, equal weight distribution in both legs’ 236

Figure 5.4 – Standing Hip Abduction Exercise 236

Figure 5.5 – Proposed LABANotation for raising the right leg out to the side ~ 12’’ 236

Figure 6.1 – Proposed Design for HTMR System Architecture 236

Figure 6.2 – Annotated Design for HTMR Software Architecture 236

Figure 6.3 – Proposed Extra View for User Working Out 236

Figure 6.4 – Proposed Text Description for Exercises 236

# 

1.1 Introduction

Performing human motion analysis requires, by its nature, the interaction of multiple disciplines, and its range of application is extensive: in clinical gait analysis, the analysis of human motion goes from understanding the multiple mechanisms that translate muscular contractions about articulated joints into functional accomplishments [1], i.e. standing, walking, to a best appreciation of the relationships between the human motion control system and gait dynamics for the planning of treatment protocols [2], i.e. surgical intervention, orthotic prescription [3]; in sports [4], analysis techniques are used to improve performance while avoiding injuries; in virtual reality applications [5], systems that combine the use of rehabilitation techniques with virtual reality training environments have been developed; in dance, the analysis and identification of motion qualities to extract emotion from a given performance by establishing relations between effort factors -space, weight, time, flow-, and their corresponding dynamic factors -force, stiffness, inertia, gravity- have been studied [6].

To perform the analysis of human motion in such different scenarios, different approaches are used and they have been developed to meet their particular requirements: high data acquisition for sports, or real time tracking for virtual reality applications which reduces the time between data acquisition and computing the simulation so the user's experience is favored. In general those approaches include motion analysis data collection protocols [7], measurement precision [8], and data reduction models [9].

Even knowing that human body parts are not rigid structures, human body is often considered as a system of rigid links connected by joints when analyzing human motion [10]. Lengths of links are measured and positions of joints are calculated as a preprocessing step. After that, the human motion analysis begins. The calculation of forces, velocities, and accelerations for specific parts of the human body takes place and those are then compared against the expected results.

This dissertation considers the human body's motion as a generator of sets of 3D curves. The motion analysis takes those sets of 3D curves and performs a measure of similarity between them, taking time and space domains into consideration. This approach avoids the need of measuring limbs in order to calculate links and joins, and allows granular analysis based on the whole human motion or just on parts of the human body for the whole performance or on specific intervals of time.

## Human Motion overview

Human motion can be classified into gesture recognition (sign language), activity recognition (surveillance), and gait analysis (orthopedics).

The nature of human motion is highly interdisciplinary and each one of those disciplines may be interested in different aspects of the subject: biomechanics focuses more on human locomotion rather than muscle models, but if the goal is to correct motion by surgery, focusing on muscle models is essential.

This section presents a short historic review of the study of human motion, after which some methods for capturing human motion, or motion capture sessions, are described. Some works done on each of the human motion classifications are presented at the end of this section.

### *A Short Historic Review*

Aristotle (383 B.C.-321 B.C.) published, among other work, a short text *On the Gait of Animals* where he discusses some interesting questions ("why are man and bird bipeds, but fish footless, and why do animal and bird, though both bipeds, have an opposite curvature of the legs"), and presents some algebra knowledge ("when one leg is advanced it becomes the hypotenuse or a right-angled triangle. Its square then is equal to the square on the other side together with the square on the base. As the legs then are equal, the one at rest must bend either at knee, or if there were any knee less animal which walked, at some other articulation") proved by experiments ("if a man were to walk parallel to a wall in sunshine, the line described by the shadow of his head, would be no straight but zigzag, becoming lower as he bends, and higher when he stands and lifts himself up"). This text and related ones written by him (*On the Parts of Animals*, *On the Progression of Animals*) are considered the first known documents on biomechanics.

Perspective geometry emerges from optical geometry (Euclid (325 B.C.-265 B.C.), *Optics*), becoming a mathematic theory later, with Gerard Desargues (1591-1661) as its pioneer. Perspective geometry and modeling human shape is how human motion studies today are basically performed.

Giovanni Alfonso Borelli (1608-1679) applied to biology the analytical and geometrical methods developed by Galileo Galilei (1564-1642), and he is often called the "father of biomechanics" due to this reason. He "was the first to understand that the levers of the musculoskeletal system magnify motion rather than force, so that muscles must produce much larger forces than those resisting the motion". This became a basic principle for modeling human motion: bones serve as levers and muscles function according to mathematical principles.

The three laws of motion, by Isaac Newton (1642-1727), provided the foundation of modern dynamics. They were also an important contribution to the understanding of human motion.

Charles Babbage (1791-1871) provided the basic ideas about computers, ideas that have been proved to be an invaluable tool in the development of many areas, human motion included.

The Weber brothers (Ernst, Wilhelm, and Eduard), analyzed gait and muscle function in their work *The Mechanics of Human Walking Apparatus, Weber and Weber, 1894*. They predicted a "walking machine" moved by steam. Also, they were the first who studied the path of the center of mass during movement.

Etienne-Jules Marey (1830-1904) was interested in locomotion of humans and animals, which led him to the design of special cameras allowing the recording of several phases of motion in the same photograph. Later he used movies, a work that was influential in the emerging field of cinematography. Edward Muybridge (1830-1904), who was inspired by the work of Marey, is famous by his photograph of a horse showing all four hooves off the ground. For this, he set up a set of 12 cameras for recording it alongside a barn sited on what is now the Stanford University campus.

This short historic review shows the interest in the study of human motion through time. Those ideas and research were the basics for new fields of study, e.g. biomechanics, computer graphics, computer vision, biophysics, robotics, sport sciences, etc. The following sub-section presents a general idea of how human motion is described.

### Describing Human Motion

Describing human motion is a challenging task due to the number of degrees of freedom of the human body. Human motion can be described using three different types of motion:

* *Linear Motion* - or Translation. All parts of an object or the whole body move the same distance in the same direction at the same time.
* *Angular Motion* - or Rotation. All parts of an object or the whole body move in a circle about a single axis of rotation.
* *General Motion* - A combination of translation and rotation. Most human movement falls into this category.

The analysis of motion can be basically performed in three dimensions:

* *1-D* - this analysis applies to motion of a point along a line requiring only one number to describe position, i.e. considering motion of center of mass.
* *2-D* - it applies to motion on a plane requiring two numbers (x and y) to specify position and one number (θ) to specify orientation.
* *3-D* - this analysis applies to multi-planar motion requiring three numbers (x, y, and z coordinates) to specify position and three numbers (pitch, yaw, and roll) to describe orientation.

The reference frames can be *absolute* (fixed in space) or *relative* (fixed to a movable object). To find joint and segment angles, relative reference frame is used.

To be able to do the analysis of motion, there is a need of an *anatomical reference position*, which is the starting point for describing later body segment movements and measuring joint angles. This is usually accomplished by having the actor to erect standing, with his feet separated slightly and pointed forward, his harms hanging at the sides, and his palms facing forward.

To describe directions of translation and axes of rotation, there is a need of an *anatomical reference frame* (Figure 1.1), which can be relative to the whole body or individual segment. This reference frame is composed by:

* *Anatomical reference axes* - which are *longitudinal* (it goes from head to feet of the actor), *mediolateral* (it goes from right to left side of the actor), and *anteroposterior* (it goes from front to back of the actor).
* *Anatomical reference planes* - which are *saggital* (it divides the actor into right and left halves), *frontal* (it divides the actor into front and back halves), and *transverse* (it divides the actor into upper and lower halves).

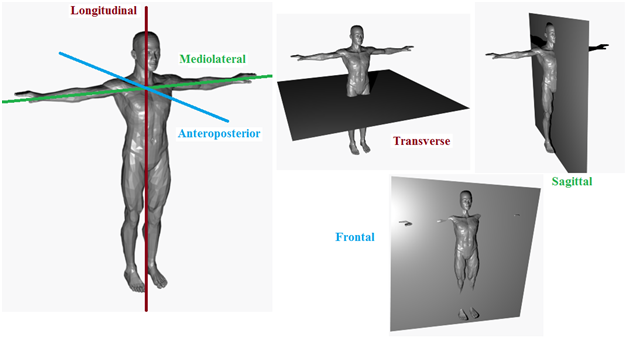


Figure 1.1 - Anatomical Reference Axes and Planes

If the motion occurs within an anatomical reference plane (Figure 1.2), the motion can be described as a *saggital motion* (Forward/Backward Up/Down), a *frontal motion* (Up/Down Right/Left), or a *transverse motion* (Right/Left Forward/Back). The importance of the planes of motion is related to muscle function:

* Muscles within a plane act as prime movers: *agonist*, or those who speed up a movement (concentric), and antagonist, or those who slow down a movement (eccentric). Some of them can also act as stabilizers (isometric).
* Muscles out of a plane act primarily as stabilizers or neutralizers by preventing unwanted motion.

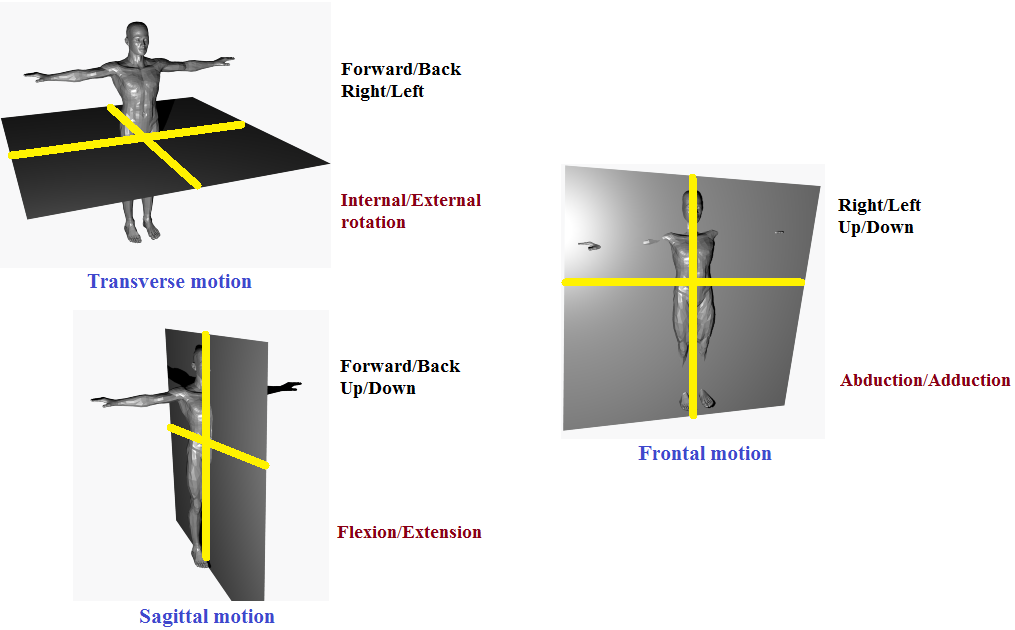


Figure 1.2 - Planar Movements

The structure of most joints in the human body allows movements in multiple planes, and the structure of the human body allows rotation about different axes at different joints. As a result, many movements do not take place in a single plane, and these movements are described as *multi-planar motion*.

Finally, the minimum number of variables needed to specify the position of the system is known as *degrees of freedom* (DF). For example, at a joint DF can be estimated by the number of axes along which segments can move relative to each other plus the number of axes about which segments can rotate relative to each other. All joints have 6 DF (3 translational and 3 rotational).

### *Capturing Human Motion*

Motion capture, or motion tracking, is a process that records movement and translates it onto a digital model which can later perform the same action as the user. In motion capture sessions, those movements (not the user's visual appearance) are sampled many times per second and this data is used to perform the analysis of human motion.

Motion capture equipment uses several technologies to produce the data that will be used to perform human motion analysis [11]. Some of them are described below:

* *Mechanical sensing* - this technology involves a direct physical linkage between the target and the environment, i.e. an articulated series of mechanical pieces interconnected with electromechanical transducers. A commercial example of such technology is Animazoo GYPSY 7, from Inition [12].
* *Inertial sensing* - technology that uses accelerometers (devices that measure the proper acceleration they experience relative to freefall. They are available to detect magnitude and direction of the acceleration as a vector quantity, and can be used to sense orientation, vibration, and shock) and gyroscopes (devices that measure or maintain orientation based on the principles of conservation of angular momentum). It was widespread used by ships, submarines, and airplanes in the 1950s, but with the advent of MEMS (microelectronic mechanical systems), inertial sensors helped to the development of inertial input devices. A commercial example of this technology is PlayStation Motion Controller [13].
* *Acoustic sensing* - technology that uses the transmission and sensing of sound waves by timing the flight duration of a brief ultrasonic pulse. A commercial example is the use of this technology for security purposes, i.e. detecting the energy produced by a breaking glass.
* *Optical sensing* - this technology has two components: light sources and optical sensors, and relies on measuring the reflected or emitted light between them. A commercial example of this technology is Vicon MX [14].

Each technology has its advantages and drawbacks (mechanical sensing: very accurate pose estimates for a single target, but with a relatively small range of motion; acoustic sensing: they offer a larger range than mechanical trackers, but can be affected by wind and require a line of sight between the emitters and receivers; etc.) and for that reason usually those technologies are mixed when obtaining data in the motion capture procedure and then motion capture equipment can record the fixed coordinates, i.e. markers, from live motion at very short intervals of time: this data can be used to create the paths of motion for the human motion analysis.

The tracking technologies used for motion capture can be summarized as follows: to sense and interpret electromagnetic fields or waves, acoustic waves, or physical forces. Specifically, the estimation of pose for those tracking systems is usually derived from electrical measurement of mechanical, inertial, acoustic, magnetic, optical, and radio frequency sensors. Each approach has its advantages and limitations, for example, electromagnetic energy decreases with distance and also subject to multipath interference; analog-to-digital converters have limited resolution and accuracy; and body worn components must be as small and lightweight as possible. For example, the MIT cricket is too bulky to wear. Using radio signal strength to estimate the distance with wireless sensors such as mica2 motes was shown to significant errors due to multipath interface.

### *Modeling Human Motion*

A body model for human motion capture has different aspects that can be distinguished:

* *The geometry or skin* - This aspect does not include any motion or animation information, but the appearance of the actor for one specific pose.
* *The skeleton* - This aspect defines the motion capabilities of the actor: the degrees of freedom of the motion model by defining a hierarchy of joints and specific bone lengths.
* *Joint positions* - this aspect defines the relation of the previous two aspects.

This model defines how a point in the skin moves with respect to joint parameters, e.g. joint angles. The calculation of such points is known in robotics as *forward kinematics*, or the calculation of a specific part if joint angles are given. The *inverse kinematics* is necessary to calculate the rotation of the joints in the skeleton for a certain part to reach a given point.

Most systems simplify the task by assuming that all the aspects of the body model are known, so the problem of motion capture is reduced to estimate joint angles per frame. The main drawback of this approach is the limited accuracy of motion, pose, and surface reconstruction.

Motion of objects can be classified into rigid and non-rigid motion. Non-rigid motion was classified first into *articulated*, *elastic*, and *fluid* motion [15]. An extended classification based on the non-rigidity of objects was proposed by [16]. This classification is shown in Figure 1.3.

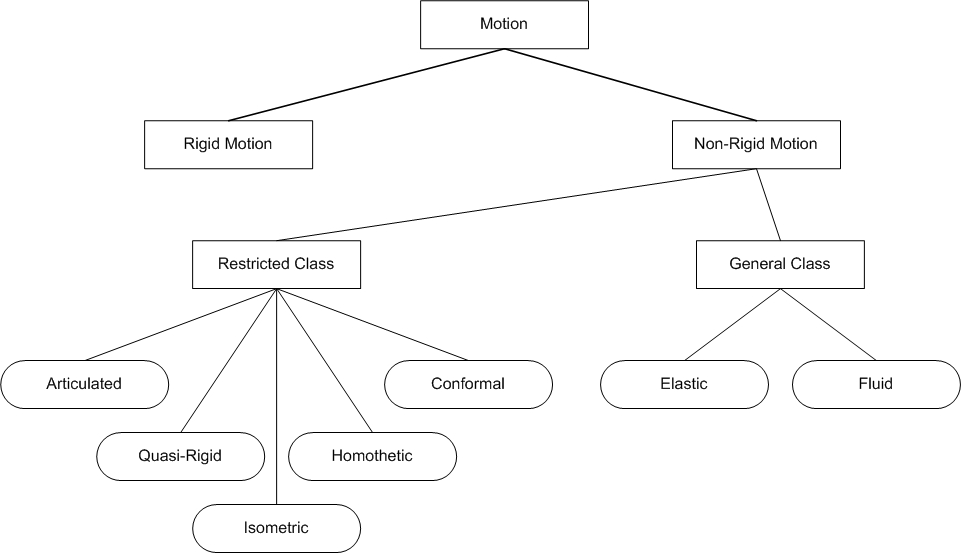


Figure 1.3 - Motion of Objects Classification

Human Motion is generally classified as *articulated motion*: the rigid parts conform to the rigid motion constrains, but the overall motion is not rigid. Conventionally, a model-based approach for articulated motion represents the human body either as a stick figure [17, 18], or as a volumetric model: 2D ribbons [19], elliptical cylinder [20], spheres [21], etc. Volumetric models represent better the details of the human body, but require more parameters for computation. Modeling human body by combining both stick figures and volumetric models [22] is another approach for modeling human motion.

## Framework for Human Motion Similarity Analysis

The interest for human motion analysis is motivated by applications that cover an extensive spectrum of topics: for medical diagnosis or athletic performance it is useful to segment the parts of the human body in an image, to track the movement of joints over an image sequence, and to recover the underlying 3D body structure; for areas such as airports where security is important, the capability to monitor human activities automatically is of great interest to the security staff; for entertainment purposes like making movies or developing videogames, the use of human-machine user interfaces that capture and analyze human motion is an important component on their process. A summary of human motion applications can be found at [23].

To help the human motion analysis, procedures are developed based on the approach taken to address such analysis. As an example, the use of a computer vision's approach to analyze the motion of the human body structure is not the same as the approach taken to track human motion using cameras, or the one used to recognize human activities from image sequences [24]. The motion analysis of the human body usually involves the extraction of low-level features including body part segmentation, joint detection and identification, etc.. Tracking human motion using cameras requires to detect the presence of humans without considering the geometric structure of the body parts; and human activity recognition is based on tracking the human through images sequences. In general, after the motion capture session is done and data is available for analysis, what remains is to find the correspondence between successive frames to understand the behavior of the studied features and, if needed, to compare such behavior with a model.

We propose a framework for human motion similarity analysis where no joint detection and identification are needed, therefore the human body is not represented as a stick figure or as volumetric model. Instead, we propose the human body to be a generator of 3D curves so the analysis of human motion is based on the shape similarity of those curves. This can be achieved due to our proposed description and model of the human motion, as explained in the rest of this section.

### *Motion Capture Session*

In a motion capture session, the movements of the actor are recorded based on the tracking system that can be optical (using passive markers, active markers, or even marker less technology) or non-optical (using inertial, magnetic, or mechanical technology). In our work we assume that the outcome of the motion capture session is a 3D dataset that includes the 3D position of each marker during the session, as shown in Figure 1.4.

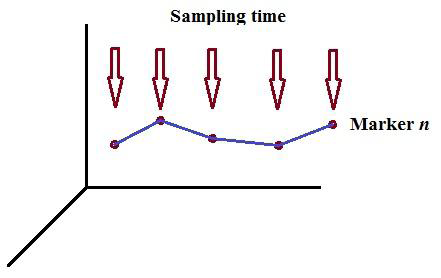


Figure 1.4 - Motion Capture Session

The technology employed to generate this 3D dataset is not part of this work, therefore the associated drawbacks for a particular technology are not addressed.

Several motion capture file formats have been developed to fulfill different needs:

* BVA & BVH - developed by Biovision. BVA is the precursor to BVH (Biovision Hierarchical Data). BVH format is mainly used as a standard representation of movements in the animation of humanoid structures.
* ASK/SDL - a variant of the BVH file format developed by Biovision. The ASK (Alias SKeleton) file only contains information concerning the skeleton. The SDL file associated to the ASK file contain the data of the movement.
* AOA - file format developed by Adaptative Optics. The file describes the captors and their position at each sampling period.
* TRC - it contains the raw data from the full body motion capture system and the data coming from the face tracker.
* CSM - it is an optical tracking format that is used by Character Studio (an animation and skinning plug-in for Autodesk 3ds Max) for importing marker data.

The 3D dataset used in this work comes in C3D format [25]. The National Institute of the Health developed this format. Some features that are included in this file format are:

* + The possibility to stock analogical data (directly coming from the measure instrument) and three-dimensional data (obtained by the information processing).
  + The possibility to stock information on the material that have been used (position marker, force captors, etc.) on the recording process (sampling rate, date, type of examination, etc.), or on the subject itself (name, age, physical parameters, etc.).
  + The possibility to add new data to the ones already recorded.
  + The file is a binary file, unlike most of other capture file format, which often are ASCII files.

We assume that 3D data is provided per marker (or any other way of capturing the motion of a given device worn by the actor during the motion capture session) on a C3D file format, and that the sampling rate is also provided within the dataset: C3D file format fulfills these requirements. Knowing what particular sub set of markers (i.e. their labels) belong to a particular part of the body (i.e. head, torso, arms, etc.) is very important in this work, as explained later in this section.

### Human Motion Description

Systems for recording and analyzing human movement have been developed throughout time for several applications: Eshkol-Wachman Movement Analysis notation (EWMN) has been utilized for the analysis of infantile reflexes on autistic children [26]; Benesh Movement Notation (BNM) for recording and restaging dance works [27], and Laban Movement Analysis (LMA) for automating motion capture segmentation [28]. Examples of these notations are shown in Figure 1.5

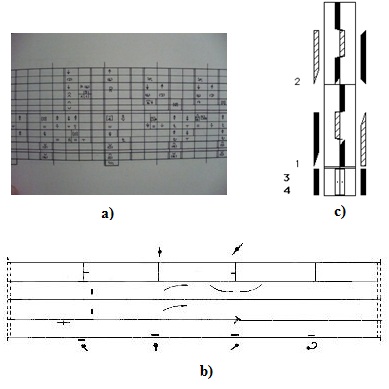


Figure 1.5 - Human Movement Notations:

a) Eshkol-Wachman, b) Benesh, c) Laban

Laban notation (Labanotation/LABANotation) was chosen as the tool to describe human motion because of its widespread use, and because it fits the objective of this proposal. A brief introduction of this notation is given below.

**Laban Movement Analysis (LMA)**

Laban is a language for interpreting, describing, visualizing and notating ways of human movement. Rudolf Laban (1879-1958) created Laban Movement Analysis (LMA). It draws on his theories of effort and shape to describe, interpret, and document human movement. Extended by the work of Irmgard Bartenieff (1890-1981), a Laban's student who applied her Laban training to the field of physical therapy [29], this system is also known as Laban/Bartenieff Movement Analysis or Laban Movement Studies and it includes Laban Movement Analysis, Anatomy and Kinesiology, Bertenieff Fundamentals, and Labanotation.

LMA includes four main categories:

* *Body* - it describes structural and physical characteristics of the human body while moving. This category describes which body parts are moving, which parts are connected, which parts are influenced by others, etc. Bartenieff developed the majority of the work of this category.
* *Effort* - or dynamics. This category describes a system for understanding the more subtle characteristics about the way a movement is done with respect to inner intention, i.e. punching someone in anger and reaching for a glass, where the strength of the movement, its control and the timing of such movement are very different. Effort has four subcategories: (Figure 1.6) Space, Weight, Time, and Flow (Figure 1.6). Laban named the combination of the first three categories the Effort Actions.
* *Shape* - the way the body changes shape during movement is analyzed through this category. All four categories are related, and Shape is often an integrating factor for combining them into meaningful movement.
* *Space* - these categories involves motion in connection with the environment, and with spatial patterns, pathways, and lines of spatial tension. Laban felt that there were ways of organizing and moving in space that were specifically harmonious in the same sense as music can be harmonious.

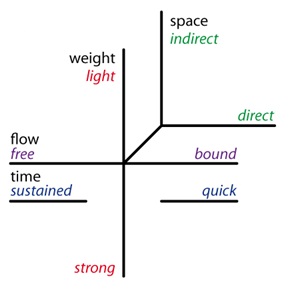


Figure 1.6 - Laban Effort Graph

LABANotation is a record of how one moves so that the same movement can be repeated. Figure 1.5c shows how LABANotation is used to specify a dance movement. This notation includes a set of symbols that are placed on a vertical staff, where its vertical dimension represents the symmetry of the body, i.e., left parts of the body and its horizontal dimension represents the time. See Figure 1.8 for the relationship between the staffs and the portions of the body they represent. The symbols need to be read from bottom to top.

In this notation, reading a symbol on a staff allows the recognition of:

* The direction of the movement
* The part of the body doing the movement
* The level of the movement
* The length of time it takes to do the movement

The shape of the symbol indicates the direction of the movement, as shown in Figure 1.7

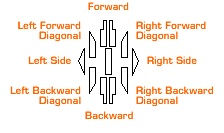


Figure 1.7 - LABANotation: Direction of Movement

The location where the symbol is placed on the staff indicates the part of the body doing the movement, as shown in Figure 1.8



Figure 1.8 - LABANotation: Parts of the Body

The dark lines on Figure 1.8 are the staff lines. The dotted lines indicate the different columns for the part of the body. Any symbol indicates either a step or a gesture: a step is a movement that takes weight, and a gesture is a movement that does not take weight. The centerline represents the centerline of the body: supports are written alongside the center line, usually the feet.

The shading of the symbol (Figure 1.9) indicates the level of the movement: for steps, low level is with a bent leg, middle level is with a straight leg, and high level is up on the toes; for gestures, middle level is with the limb parallel to the floor: the hand or the foot is on the same level as the shoulder or hip, low level is below this and high level is above.

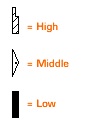


Figure 1.9 - LABANotation: Level of Movement

The length of the symbol indicates the timing of the movement, as shown in Figure 1.10

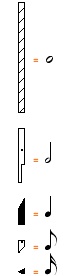


Figure 1.10 - LABANotation: Time

For dance movement, the staff is laid out in measures to match the measures of the music. Tick marks are used to indicate the beats, bar lines across the staff show the start and end of the measure. The space below the double bar lines at the bottom of the staff is for the starting position, and the double bar lines at the top of the staff indicate the end of the movement. Figure 1.11 shows an example of a score for a simple basic LABANotation. It specifies the right leg rises high in the first measure following by right arm moving to the lower right.

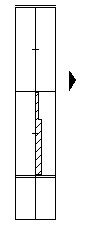


Figure 1.11 - LABANotation: Score

### *Human Motion Model*

A stick figure model [30] or a volumetric model [31] can be used to model human motion. Their goal is to construct a general model, i.e. to be used in gait analysis, which represents the human body by a skeleton, as shown in Figure 1.12 [32]. Once joints and links are calculated, kinematics and reverse kinematics are applied to analyze the human motion of the body.

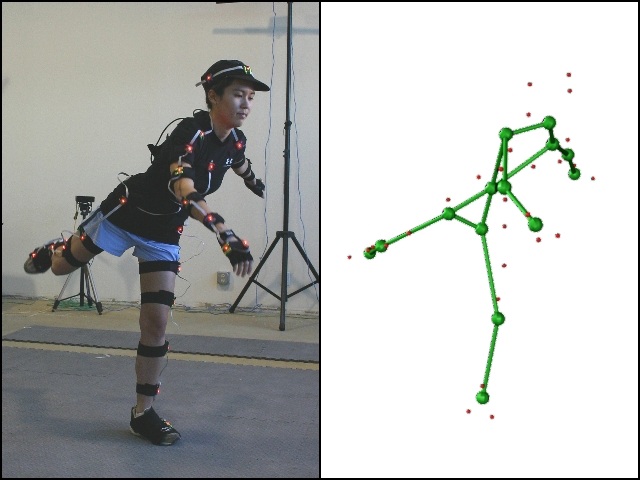


Figure 1.12 - Human Motion Model: Skeleton

We propose a model that considers the human body as a generator of 3D curves: once markers are set on the body, i.e. the red lights shown in the left side of Figure 1.12, each marker generate a 3D curve when motion is performed. The set of 3D curves represent the whole human motion for a particular performance, while the sub sets of 3D curves for the markers on a particular limb represent the motion of this particular limb. This is shown in Figure 1.13, where the performer is wearing markers on his body, not only on the arms.

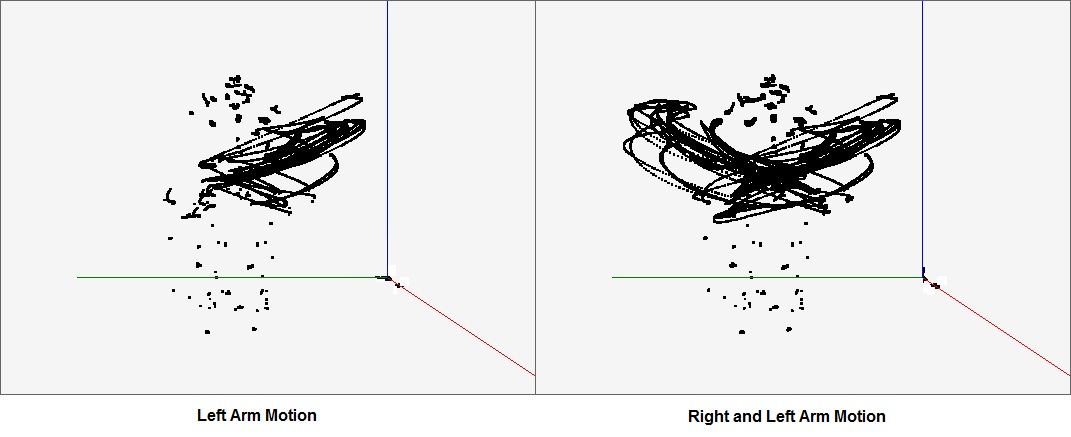


Figure 1.13 - 3D Curves Generated by Markers

Chain code methods are widely used because they preserve information and allow data reduction [33]. Also, chain codes are the standard input for numerous shape analysis algorithms [34]. A chain code representation for a 2D figure is shown in Figure 1.14. Each of the 3D curves generated by the markers in Figure 1.13 can be mapped into a chain code representation for purposes of performing the analysis of similarity.

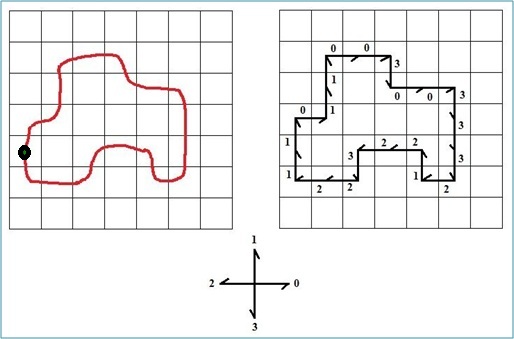
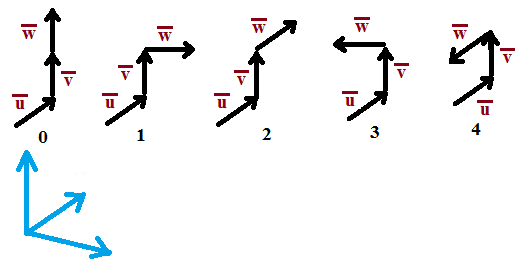


Figure 1.14 - Chain Code Representation for a 2D Figure

Starting at the point marked in the figure and following a clockwise direction, the orthogonal changes of direction generate the following chain code representation for the 2D figure: 10110030033321223221. If this figure is rotated 90 degrees to its right and the same method is applied, the chain code obtained is 03003323322210112110. This would suggest that the second figure is different from the first one, which is not the case.

E. Bribiesca [35] proposes a method for representing 3D discrete curves composed of constant straight-line segments that are invariant under rotation and translation. This method utilizes a chain code representation for the orthogonal -relative- direction changes of those constant straight line segments. It allows only five codes for the proposed four orthogonal changes of direction, plus the scenario where there is no change of direction.. The generation of those five codes, or chain elements, is based on the cross product of two orthogonal vectors as shown in Figure 1.15

Figure 1.15 - Five Possible Changes of Direction for Representing 3D Curves



Formally, if the consecutive sides of the reference angle have respective directions *u* and *v*, and the side from the vertex to be labeled has direction *w*, then the chain element is given by the following function,

**0**, if *w* = *v*;

**1**, if *w* = *u* x *v*;

**chain element (*u*, *v*, *w*)** = **2**, if *w* = *u*;

**3**, if *w* = - (*u* x *v*);

**4**, if *w* = - *u*;

where x denotes the cross product.

As an example, a chain code representation of a 3D curve is shown in Figure 1.16

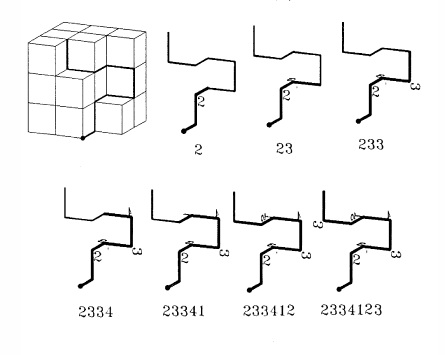


Figure 1.16 - Chain Code Representation for a 3D Curve

Due to the fact that relative direction changes are used, chain codes are invariant under translation and rotation as shown in Figure 1.17: reading it from left to right and from top to bottom, the first figure represents a 3D discrete curve and its chain code; the next three figures represent a 90, 180, and 270 degrees rotation on the axis X; the next three figures represent the same amount of rotation on the axis Y; the last three figures represent the same amount of rotation on the axis Z. Note that all chain codes are equal.

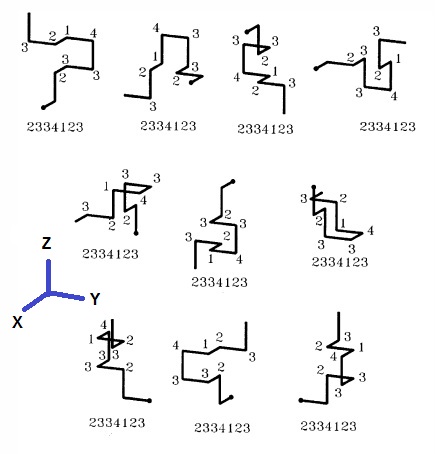


Figure 1.17 - Chain Code Representation is Invariant under Translation and Rotation

Bribiesca describes other important concepts offered by his method: inverse of a chain, independence of starting point for open and closed curves, and invariance under mirroring transformation. Those concepts are not addressed here because they are not related to our work.

### *Similarity*

Similarity is an important concept in many fields. A measure of similarity states that *it is designed to quantify the likeliness between objects so that if one assumes it is possible to group objects in such a way that an object in a group is more like the other members of the group than it is like any object outside the group, then a cluster method enables such a group structure to be discovered* [36]. However, different fields provide definitions of similarity according with their subject:

**Similar triangles (mathematics, geometry)** - triangles that have the same shape and are up to scale of one another (for a triangle, the shape is determined by its angles). Formally speaking, two triangles ∆ABC and ∆DEF are similar if either of the following conditions holds:

* Corresponding sides have lengths in the same ratio:
* ∠BAC is equal in measure to ∠EDF, and ∠ABC is equal in measure to ∠DEF. This also implies that ∠ACB is equal in measure to ∠DFE.

**String similarity (information theory, computer science)** - a string *s* is said to be a *subsequence* of string *t* if the characters of *s* appear in order within *t*, but possibly with gaps between occurrences of each character [37]. Examples of metrics to calculate the similarity between strings are:

* The Levenshtein distance (edit distance) which is defined as the minimum number of insertions, deletions, and/or substitutions required to change one string into another [38]:

kitten -> sitten (substitution of 's' for 'k')

sitten -> sittin (substitution of 'i' for 'e')

sittin -> sitting (insert 'g' at the end)

* Dice's coefficient (Dice coefficient) is a term based similarity measure (0-1) whereby the similarity measure is defined as twice the number of term common to compare entities divided by the total number of terms in both tested entities [39]. When taken as a string similarity measure, the coefficient may be calculated for two strings, *x* and *y*, using bigrams as follows:

where *nt* is the number of character bigrams found in both strings, *nx* is the number of bigrams in string *x* and *ny* is the number of bigrams in string *y*. For example, to calculate the similarity between fight and weight, the set of bigrams in each word is {fi, ig, gh, ht} {we, ei, ig, gh, ht}. The intersection of those sets has three elements ({ig, gh, ht}). Putting those values into the formula, the coefficient *s* = 0.66667.

Dekan Lin [40] proposes an information-theoretic definition of similarity that achieves universality and theoretical justification from a set of assumptions about similarity. It applies to the domain which has a probabilistic model. The similarity measure is not defined directly by a formula. It is based on the following intuitions:

* **Intuition 1** - The similarity between A and B is related to their commonality. The more commonality they share, the more similar they are.
* **Intuition 2** - The similarity between A and B is related to the differences between them. The more differences they have, the less similar they are.
* **Intuition 3** - The maximum similarity between A and B is reached when A and B are identical, no matter how much commonality they share.

The above definition of similarity is the one we use in this work.

## Contributions

One of the main contributions of this dissertation is to model human motion as chain codes, or strings composed of characters from a finite alphabet. This model of human motion allows the use of string matching algorithms, sequence alignment algorithms, and statistical analysis approaches to achieve the analysis of similarity. Another contribution is the ability of spatial and temporal analysis due to the proposed model and description of the human motion. The results in Chapters 2 and 3 was published in Proceedings of *Gait & Clinical Movement Analysis Society Conference, GCMAS 2009*, titled "Improve Rehabilitation and Patient Care with Laban Specification and Wireless Sensor Tracking.”

## Dissertation Outline

Our proposed framework for the analysis of human motion similarity is presented in Chapter 2. It utilizes the concepts and knowledge covered in Section 1.2 including what is provided by motion capture sessions, how to describe human motion, and how such motion is modeled

Chapter 3 presents our case of study by applying the framework on a set of key rehab exercises obtained in a joint research project between Taipei National University of Art, Colorado College, Children Hospital of Denver, and University of Colorado at Colorado Springs. The key rehab exercises used in the experiment are described here, and the results of this experiment using two different algorithms to measure human motion similarity are discussed: Needleman-Wunsch, and our modified version of Bribiesca’s work to analyze shape similarity among 3D curves.

In Chapter 4, we present the results of evaluating human motion similarity using our algorithm and comparing those values against the values provided by FastDTW algorithm [43].

Chapter 5 presents an enhancement of LABANotation for rehabilitation purposes.

A design of human motion tracking system architecture for rehabilitation purposes is presented in Chapter 6. Suggestions to enhance Laban Dancer software [44] for rehabilitation purposes, are also discussed in this chapter.

Chapter 7 includes future research and conclusions of this dissertation.

Appendix A contains the algorithm that maps 3D data sets into chain codes that represent orthogonal changes of directions for 3D curves.

Appendix B contains the algorithms that implement the analysis of human motion similarity, including Needleman-Wunsch and 3D curve shape dissimilarity.

# Proposed Framework for Measuring and Analyzing Human Motion Similarity

In this section, motion capture session, human motion description, human motion model, and the definition of similarity are put together to describe our proposed framework for measuring human motion similarity. The big picture of this framework is shown in Figure 2.1.

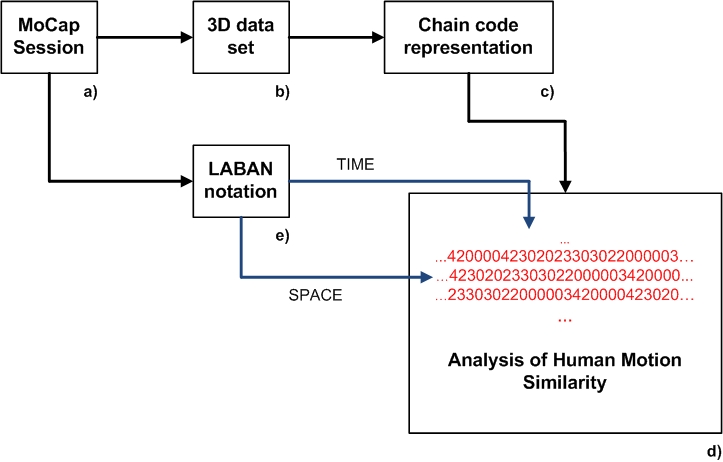


Figure 2.1 - Proposed Framework for the Analysis of Human Motion Similarity

A given rehabilitation exercise, or a piece of dance, is recorded in a motion capture session (**a**). As explained before, the technique employed in such session is outside our work. The only requirements we have is that we get the 3D positions for each marker used during the motion capture session, including the sampling rate utilized in the session. C3D file format fulfills this requirement; therefore the 3D data set obtained in (**b**) utilizes such file format. In this step, it is also assumed that we know what sets of markers were placed on what parts of the human body, i.e. markers 1-4 belong to the head, markers 10-14 belong to the right arm, etc. At his point, 1) we have the trajectory described for each marker during the motion capture session, 2) we know where those markers were placed in the actor's body, and 3) we know the sampling rate utilized for the motion capture session.

On step (**c**), the trajectories for each of the markers are mapped into chain codes, representing the 3D curves by using orthogonal changes of direction for each of those markers. The pseudo-code for the algorithm developed for this step is given in Appendix A, but it can be summarized like this: a 3D Cartesian coordinate system is used, where the size of the grid represents the constant straight line segments that form the 3D discrete curves on Bribiesca's work [35]. The first 3D position for a given marker represents the starting point A, and the next 3D position for this marker represents the ending point B, and vectors *u* and *v* (Figure 1.15) are arbitrarily set at the beginning of this algorithm. We calculate what plane of the cube -the grid- the line AB intersected, and the vertex of the cube closer to the point A is selected. The straight-line segment formed from point A to the chosen vertex represents the vector *w* in Figure 2.2, and a chain element is generated based on those three vectors. This is done for each marker's data set. An example of this algorithm is shown in Figure 1.16.

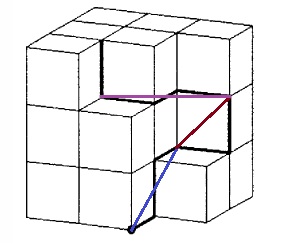


Figure 2.2 - Chain Code Generation per Marker, per Sampling Rate

The chain code representation of each marker's trajectory is our model for the human motion (**d**): the analysis of human motion similarity is done by applying our human motion similarity algorithm, as well as FastDTW algorithm.

Human motion notation (**e**) plays an important role in our proposed framework. As explained in Chapter 1.2.2, LABANotation is a 2D representation of the human movement. If we visualize this notation as a bi-dimensional matrix, its rows represent the time of the movement (TIME), while its columns represent the parts of the body (SPACE) involved on a given human motion. Intersection of rows and columns represent the motion of a human body part in a specific time of the performance. Therefore, knowing what markers belong to what parts of the body, and knowing the sampling rate of the motion capture session, the LABANotation of a given motion can rule a detailed analysis of the human motion if desired, i.e. analyze the human motion similarity for the right arm on measure 1 and 2, analyze the human motion similarity for the torso from begin to end, etc.

## Contribution

In this Chapter, we proposed a novel way of representing human motion by mapping human motion into chain codes, and by associating this representation with LABANotation to conduct the analysis of human motions both in spatial and temporal domain.

# Case of Study: Rehabilitation Therapy

A given human motion, represented as a timed 3D ChainCode, will be used as a MODEL to be compared with. Then, subsequent 3D ChainCode representations of human motion will be seen as USERS to compare with the MODEL. We assume that markers worn by the actors on both the MODEL and the USERS motion capture sessions were placed on practically the same body place. We do not expect to have an exact match, but an approximated one, as the result of the analysis of human motion similarity even if the performance was done by the same user.

As a project sponsored by Taipei National University of Art, with the participation of Colorado College, University of Colorado at Colorado Springs, and Children Hospital of Denver, a team consisting of expert on gait and motion analysis, Dr. James Carollo, Physical Therapy expert Susan Kanal, dancing and Labanotation analysis expert Professor Yunyu Wang, identified four key rehabilitation exercises: *mini-squats*, *posterior pelvic tilt*, *shoulder elevation with rotation*, and *standing hip abduction for Human Body Movement Analysis*. The selected rehab exercises are video taped with the brand new high end high resolution VICOM camera system in the Gait and Motion Analysis Lab at the Denver Children Hospital with four different camera angles and the related motion captured data are saved in C3D data format for further study. These valuable data are used as a reference data set for the design and implementation of computer animation and human motion tracking systems. They served as the basis for our experimental research.

## Key Rehabilitation Exercises

**Mini-squats: muscle group – hips**

*Description*

* Start standing with equal weight distributed between right and left legs.
* Place feet shoulder width apart.
* Keep torso upright, avoid bending at the waist.
* Slowly lower yourself by bending the ankles, knees, and hips.
* Return to standing.

*LABANotation*: Figure 3.1 shows the LABANotation for this key rehabilitation exercise.

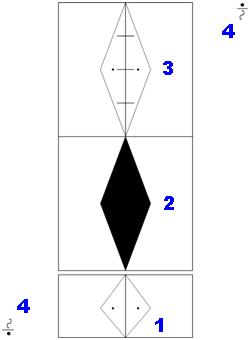


Figure 3.1 - LABANotation: Mini-Squats

*Explanation*

1. It starts with feet apart, equal weight distribution on both legs, middle level.
2. Keep the same feet position, but this time you go to low level. Do this movement in 4 times.
3. Keeping the same feet position, return to the original middle level. Do this movement in 4 times.
4. Do 2) and 3) as many times as desired.

**Posterior pelvic tilt: muscle grout – abdominal, pelvis**

*Description*

* Start laying supine/on your back.
* Bend legs up so both feet are on the supporting surface/table.
* Slowly tighten abdominal muscles to flatten your low back into the table.
* Avoid pushing through your legs.

*LABANotation*: Figure 3.2 shows the LABANotation for this key rehabilitation exercise.

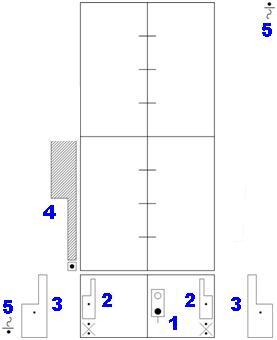


Figure 3.2 - LABANotation: Posterior Pelvic Tilt

*Explanation*

1. Start lying on your back.
2. Contract your legs to form a 90-degree angle between the upper leg and the lower leg.
3. Spread your arms toward your feet.
4. Move the pelvis upwards. Do this movement in 4 times.
5. Move back the pelvis to its original position. Keep the original position for 4 times.
6. Do 4) and 5) as many times as desired.

**Shoulder elevation with rotation: multiple muscles around the scapula and shoulder**

*Description*

* Start standing, with right hand/palm placed on the front of the left hip.
* Slowly, begin to elevate the right upper extremity to shoulder height. At the same time, rotate the right arm outward so the thumb is pointing toward the ceiling.
* Keep the elbow straight throughout.
* Lower the arm and return to the start position

*LABANotation*: Figure 3.3 shows the LABANotation for this key rehabilitation exercise.

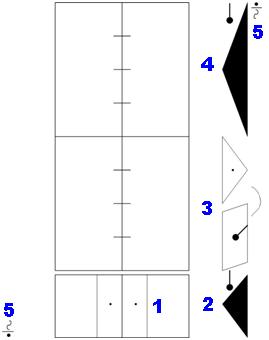


Figure 3.3 - LABANotation: Shoulder Elevation with Rotation

*Explanation*

1. Start with equal weight distribution on both legs, middle level.
2. Cross your right hand in front of you body, low level.
3. Move your right hand to the right, middle level, twisting your arm 45 degrees to the right at the same time. Do this movement in 4 times.
4. Return your right hand to the original position, crossing in front of your body. Do this movement in 4 times.
5. Do 3) and 4) as many times as desired.

**Standing hip ABD (abduction): muscle group – hips**

*Description*

* Start standing with equal weight distributed between the right and left legs
* Slowly, shift your weight to the left side
* Raise the right leg out to the side ~ 12”
* Keep the right foot facing forward
* Keep the torso upright and avoid leaning to the side

*LABANotation*: Figure 3.4 shows the LABANotation for this key rehabilitation exercise.

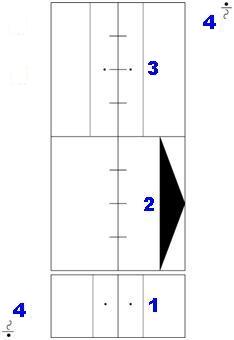


Figure 3.4 - LABANotation: Standing Hip Abduction

*Explanation*

1. Start with equal weight distribution on both legs, middle level.
2. Keep your weight on your left leg, while moving your right left to your right, raising it to the low level. Do this movement in 4 times.
3. Return your right leg to its original position, middle level. Do this movement in 4 times.
4. Do 2) and 3) as many times as desired.

3D data sets on C3D file format were collected from The Children's Hospital, Denver, in the Center for Gait and Movement Analysis. Motion data were captured with a 6 camera Vicon 512 system, using progressive scan 240Hz cameras. 3D data sets capture the human motion of these four key rehabilitation exercises, three sessions per exercise.

## Experiments and results

As stated at the end of Chapter 2, what remains is to perform the analysis of human motion similarity. Having our human motion model as sets of strings, it is straightforward to look for an algorithm that finds out how similar two strings are.

The first algorithm we have used in order to perform the analysis of human motion similarity is the Needleman-Wunsch algorithm. The second one is an approach that measures the dissimilarity between two 3D curves mapped as chain codes. These two approaches are explained in the next two sub sections.

### *Needleman-Wunsch Approach*

This is a global alignment algorithm widely used on bioinformatics to know how closely two sequences are related. It obtains the optimal global alignment between two sequences (strings of a finite alphabet), allowing gaps, by building up an optimal alignment using previous solutions for optimal alignments of smaller subsequences. It uses a scoring scheme, or a set of rules, which assigns the alignment score to any given alignment of two sequences. The scoring scheme is residue based: it consists of residue substitution scores (i.e. score for each possible residue alignment), plus penalties for gaps. The alignment score is the sum of substitution scores and gap penalties.

This algorithm is an example of dynamic programming, and in general consists of three steps:

* Initialization of the score matrix
* Calculation of scores and filling the trace back matrix
* Deducing the alignment from the trace back matrix

As an example, consider the following sequences:

Sequence 1: **G A A T T C A G T T A**

Sequence 2: **G G A T C G A**

Applying the simple scoring scheme of 1 if there is a match score, 0 if there is a mismatch score, and 0 for gap penalty, two possible optimal global sequence algorithms with a maximal global alignment score of 6 are:

**G A A T T C A G T T A**

**| | | | | |**

**G G A \_ T C \_ G \_ \_ A**

or

**G \_ A A T T C A G T T A**

**| | | | | |**

**G G \_ A \_ T C \_ G \_ \_ A**

We used the Needleman-Wunsch algorithm with values of 1 for match, 0 for mismatch, and 0 for gap penalty. The following is the algorithm in pseudo code used to analyze the human motion similarity.

**HumanMotionSimilarity**(hm1, hm2)

**For** each marker on hm1 and hm2 **Do**

seq1 = chain codes for this marker on hm1

seq2 = chain codes for this marker on hm2

**Needleman-Wunsch**(seq1, seq2, seq1\_al, seq2\_al)

/\*Intuition 3\*/

**If** seq1\_al = seq2\_al **Then**

similarity\_value = 1.0

**Else**

/\*Intuition 1 and 2\*/

seq\_length = length\_of(seq1\_al)

**For** i = 1; i <= seq\_len **Do**

**If** seq1\_al[i] = seq2\_al[i] **Then**

sim = sim + 1

**EndIf**

**EndFor**

similarity\_value = sim / seq\_length

**EndIf**

sum\_sim\_value = sum\_sim\_value + similarity\_value

counter = counter + 1

**EndFor**

sum\_sim\_value = sum\_sim\_value / counter

If chain codes are the same, the human motion similarity value is 1 for Needleman-Wunsch algorithm. This corresponds to Intuition 3 in the definition of similarity used in this proposal. If chain codes are not the same, Needleman-Wunsch algorithm will insert gaps as needed until the optimum sequence alignment is found. This corresponds to Intuitions 1 and 2.

Some of the results obtained by using 3D ChainCode implementation with different ***step size*** (grid size) values (0.2, 0.4, 0.6, 0.8, 1.0) are shown in Table 3.1. Labels used in the table are:

**MSx** = Mini-squat x

**PPx** = Posterior Pelvic Tilt x

**SEx** = Shoulder elevation with rotation x

**SHx** = Standing hip abduction x

where x represents the session number (1-3), and numbers in **bold** face represent the ***step size***

Table 3.1 - Human Motion Similarity values for the four Key Rehabilitation Exercises

using different step sizes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0.2** | MS1 | MS2 | MS3 |  | **1.0** | SE1 | SE2 | SE3 |
| MS1 | 1.0 | 0.577 | 0.640 |  | SE1 | 1.0 | 0.868 | 0.839 |
| MS2 |  | 1.0 | 0.601 |  | SE2 |  | 1.0 | 0.855 |
| MS3 |  |  | 1.0 |  | SE3 |  |  | 1.0 |
|  |  |  |  |  |  |  |  |  |
| **0.4** | PP1 | PP2 | PP3 |  | **0.8** | SH1 | SH2 | SH3 |
| PP1 | 1.0 | 0.809 | 0.739 |  | SH1 | 1.0 | 0.778 | 0.730 |
| PP2 |  | 1.0 | 0.714 |  | SH2 |  | 1.0 | 0.750 |
| PP3 |  |  | 1.0 |  | SH3 |  |  | 1.0 |

The graphs for mini-squat, posterior pelvic tilt, and standing hip abduction key rehabilitation exercises are shown in Figures 3.5, 3.6, and 3.7 respectively.



Figure 3.5 - Human Motion Similarity for Mini-Squat Exercise



Figure 3.6 - Human Motion Similarity for Posterior Pelvic Tilt Exercise



Figure 3.7 - Human Motion Similarity for Standing Hip Abduction Exercise

We have presented the use of chain code representation for 3D curves that allows the analysis of human motion similarity. Analysis of similarity among different human motions is done by searching for global alignment between chain codes using Needleman-Wunsch global alignment algorithm, and then by applying an information-theoretic definition of similarity on those alignments.

Chain code representation for 3D curves proposed in [35] was enhanced to include a representation for the lack of motion, or the periods of time where body parts are immobile during human motion.

Different values for ***step size*** were used in the analysis of similarity for the four key rehabilitation exercises. Although graphs in Fig. 3.5, 3.6, and 3.7 show an increase of human motion similarity if ***step size*** is larger, that is not the case. Larger ***step size*** generates more lack of motion characters in the chain codes even if the body parts are moving, and the ratio of sensors placed on body parts that barely move in a particular human motion add to the final value of human motion similarity. As an example, Mini-squats key rehabilitation exercise shown in Fig. 3.5 requires the movement of almost all of the markers set on the human body, whereas in Posterior pelvic tilt key rehabilitation exercise requires the movement of very few markers, i.e. those in the abdominal region. We can see that in those motions where most of the markers are immobile contribute to have a more similar human motion, which increases if ***step size*** is larger.

An approach that considers the human motion as a human body generating sets of 3D curves, and the description of human motion by using a human movement notation is described in [41]. The motion analysis takes those sets of curves and performs a measure of similarity between them, taking time and space domains into consideration.

Finally, the use of a global sequence alignment algorithm serves well in an approach where the analysis of similarity is meant for the whole human motion, but if we need a more granular analysis we need an algorithm that does not modify the chain codes in order to provide motion similarity values. Reference [42] proposes a measure of dissimilarity for 3D curves that has been adapted for the purposes of this work, and is explained in the next sub section.

### *3D Curve Shape Dissimilarity approach*

A measure of shape dissimilarity for 3D curves [42] that does not modify the chain code representation for human motion was implemented. This approach does not include code representation for lack of motion, nor for a trajectory that returns in the opposite direction.

A dissimilarity measure is a function that associates a numeric value with a pair of sequences where a lower value indicates greater similarity. Two 3D curves are more similar when they have in common more sub curves, and when those sub curves have the same orientation and position inside their 3D curves. These common sub curves could overlap, so there is a need of finding a way to choose the right sub curve among those that overlap. Finding common sub curves and detecting overlaps between them, if any, is an approach that follows the definition of similarity presented on Section 3.4

***Finding similarities***

Due to the fact that 3D curves are represented by orthogonal changes of direction, as explained in Section 3.3, every 3D curve (and sub curve) has a unique representation that is invariant under translation and rotation. This means that a sub curve that could be found in a 3D curve will have the same representation regardless its position and orientation.

The position of the sub curve is given by the index of the sub curve's first element within the 3D curve it belongs to.

The first two non-zero elements preceding the sub curve represent the two orthogonal changes of direction needed to define the first element of the common sub curve. If needed, imaginary non-zero elements can be added, i.e. if the common sub curve begins at indices 1 or 2 of the 3D curve.

Summarizing, the similarity of two 3D curves is given by finding the longest common sub curves among them, including the first *n* elements needed to define the first two orthogonal changes of direction. This represents the problem of finding all the longest common sub strings of two strings. Two strings A and B are used as an example.

Given the strings,

A = 41434

B = 222221432222214322222100200000400003100003000400031003004031034

the maximum common couples (S, T) are:

(S, T) = (4**143**,

22222**143**)

(S, T) = (4**143**,

2222214322222**143**)

(S, T) = (414**34**,

2222214322222143222221002000004000031000030004000310030040310**34**)

where the common substring P is shown in bold face.

***Finding dissimilarities***

After finding all the longest common sub curves for two 3D curves, it could be possible that two or more common sub curves overlap. A dissimilarity measure for this scenario provides a parameter to choose the best matching if overlaps are found based on:

**size**: the length of the sub curve

**position**: the sub curve starting element index

**beginning**: the number of non-zero elements needed to define the first element of the sub curve

**accumulated direction**: the final orientation of the sub curve after it has been affected by all its preceding chain elements.

Accumulated direction is composed of two direction vectors that are the reference to define the next element of the sub curve. For the first sub curve element, those two vectors are given arbitrarily but they have to be orthogonal to each other. Also, two non-zero elements are needed at the beginning of the sub curve to be used as a reference for the first element of the sub curve. The new vectors *u* and *d* are calculated in terms of the current vectors *u'* and *d'* and the current chain element as shown below:

Element 0: *u* = *u'* *d* = *d'*

Element 1: *u* = *d'* *d* = *u*' x *d'*

Element 2: *u* = *d'* *d* = *u'*

Element 3: *u* = *d'* *d* = - (*u'* x *d'*)

Element 4: *u* = *d'* *d* = - *u*

where x denotes the cross product.

The accumulated direction for a curve is provided by the final vectors *u* and *d*for each sub curve. If the vectors for sub curve 1 are equal to those of sub curve 2, a value of 0 is given. If they are not equal, a value of 1 is given. Figure 3.8 shows the accumulated direction for a given 3D curve.

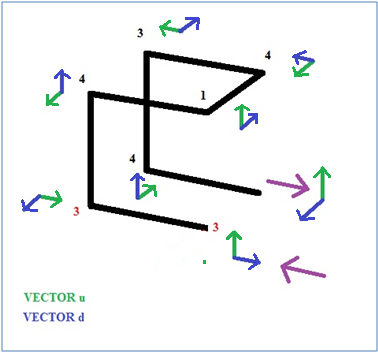


Figure 3.8 - Accumulated Direction for a Given 3D Discrete Curve

The given 3D curve shown in Fig. 3.8 is represented as 41434. Red chain elements (33) represent the non-zero elements needed to be used as the reference for the first element of the sub curve. They are also needed to obtain the first two pair of vectors *u* and *d*.

Summarizing, after finding all the maximum common couples (S, T) among two 3D curves, the following equation (Eq. 1) is used as a sorting criteria:

**Equation 1**

0, when S = T

d(S, T) = , when S ≠ T

where,

|  |  |
| --- | --- |
|  | measures the displacement of the two common sub curves within their respective curves |
|  | measures how large is the common sub curve (P) with respect to the 3D curve where it is contained |
|  | pseudo-metric of accumulated direction |
|  | measures the number of preceding elements to P in 3D curves S and T |

Lower dissimilarity values obtained in Eq. 1 mean higher ranks for the common couple in the sorted list. Continuing the example using strings A and B, the resulting sorted list is:

(S, T) = (4**143**,

22222**143**)

(S, T) = (4**143**,

2222214322222**143**)

(S, T) = (414**34**,

2222214322222143222221002000004000031000030004000310030040310**34**)

Eq. 1 can be bounded to the range [0, 1] as shown in the following equation (Eq. 2):

**Equation 2**

Sorted longest common couples could present overlaps among them. When an overlap is found among those sub curves, the one that has the lower dissimilarity *d'* is chosen. Reference [42] proposes a way to inspect the eliminated common sub curves to find out if there are still parts of them that could be matched without overlap. In our implementation, however, the common substring P with higher dissimilarity *d'(s, t)* is eliminated. This process is shown in the following pseudo code:

**setCommonCouples** (L)

while (L is not empty)

selected = first common couple in L

LL = remaining common couples in L

while (LL is not empty)

if (LL->P is a sub chain of

selected->P)

remove current common couple

from L

end if

end while

**insert** (selected, chosen)

remove selected from L

end while

return chosen

end function

where ***L*** is the list of common couples (S, T) using Eq. 1 as sorting criteria, ***insert*** is a function that inserts common couple ***selected*** in list ***chosen***, and ***chosen*** is the list with common couples (S, T) with no overlaps. The above function provides the set of longest common sub curves that represent the similarity between two 3D curves. Continuing with the example, the sorted list of common couples with no overlaps is:

(S, T) = (4**143**,

22222**143**)

(S, T) = (414**34**,

2222214322222143222221002000004000031000030004000310030040310**34**)

Eq. 3 adds up the dissimilarity values for each common couple with no overlaps in the list:

**Equation 3**

where,

|  |  |
| --- | --- |
|  | is the jth partial common sub curve found in the two 3D curves with no overlap with other partial common sub curves |
| *l* | is the total number of partial common sub curves found in the two 3D curves without overlap |

To quantify the differences between those 3D curves, the number of elements left without correspondence is computed using the following equation (Eq. 4):

**Equation 4**

where,

|  |  |
| --- | --- |
|  | is the jth sub curve that corresponds to the jth partial common sub curve |
|  | are the number of elements in Sj and Tj needed to define an orthogonal change of direction with respect to P'j |
|  | length of curve A |
|  | length of curve B |
|  | the number of elements (2 per sub curve) needed to define the first element of each sub curve |

Finally, the following equation (Eq. 5) includes the similarities and the differences between two 3D curves:

**Equation 5**

which can be bounded to the range [0, 1], as shown in the following equation (Eq. 6):

**Equation 6**

representing a measure of shape dissimilarity for 3D discrete curves.

Results obtained by using this approach with different ***step size*** values (0.2, 0.4, 0.6, 0.8, 1.0) for Posterior Pelvic Tilt exercise and Eq. 6 are shown in Figure 3.9 and in Table 3.2

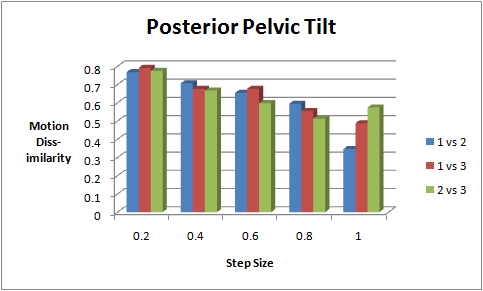


Figure 3.9 - Human Motion Dissimilarity for Posterior Pelvic Tilt Exercise

Table 3.2 - Human motion dissimilarity values for posterior pelvic tilt exercise

using different step sizes

|  |  |  |  |
| --- | --- | --- | --- |
| **0.2** | PP1 | PP2 | PP3 |
| PP1 | 0.0 | 0.767545 | 0.791231 |
| PP2 |  | 0.0 | 0.774239 |
| PP3 |  |  | 0.0 |
|  |  |  |  |
| **0.4** | PP1 | PP2 | PP3 |
| PP1 | 0.0 | 0.706760 | 0.675754 |
| PP2 |  | 0.0 | 0.666683 |
| PP3 |  |  | 0.0 |
|  |  |  |  |
| **0.6** | PP1 | PP2 | PP3 |
| PP1 | 0.0 | 0.653668 | 0.675881 |
| PP2 |  | 0.0 | 0.597629 |
| PP3 |  |  | 0.0 |
|  |  |  |  |
| **0.8** | PP1 | PP2 | PP3 |
| PP1 | 0.0 | 0.594289 | 0.554475 |
| PP2 |  | 0.0 | 0.512633 |
| PP3 |  |  | 0.0 |
|  |  |  |  |
| **1.0** | PP1 | PP2 | PP3 |
| PP1 | 0.0 | 0.345936 | 0.487258 |
| PP2 |  | 0.0 | 0.573433 |
| PP3 |  |  | 0.0 |

The analysis for human motion dissimilarity shown in Fig. 3.9 can be represented as the analysis of human motion similarity by subtracting the dissimilarity values from 1. For example, the result of comparing the *similarity* value between PP1 and PP2 with a ***step size*** of 0.2 is

1 - 0.767545 = 0.232425

These new values of similarity are shown in Figure 3.10

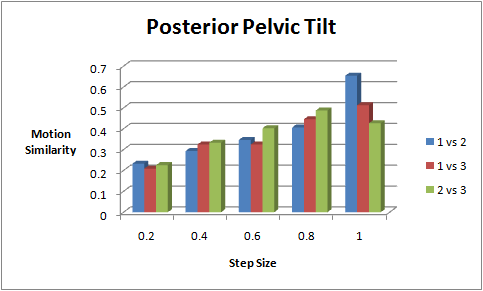


Figure 3.10 - Human Motion Similarity for Posterior Pelvic Tilt Exercise

We have presented the use of a measure of 3D curve shape dissimilarity method for the analysis of human motion similarity. Analysis of similarity among different human motions is done by modeling human motion as a set of 3D curves, and then by applying an information-theoretic definition of similarity on those alignments.

The original chain code representation [35] is intended for the analysis of closed 3D curves. This representation was enhanced to include a character (code) for the lack of motion, or the periods of time where body parts are immobile during human motion, and to include a code for trajectories that return by the same path.

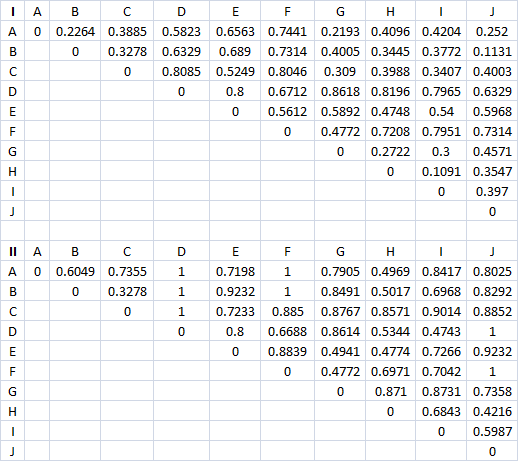
Figure 3.6 shows an analysis of human motion by using Needleman-Wunsch algorithm applied to the posterior pelvic tilt exercise. To perform this analysis, the chain code representation of 3D curves was altered (gap insertion) and codes representing lack of motion were changed during the analysis. This work uses a different approach where 3D chain code representation was changed only by eliminating the characters that represent lack of motion.

Data in Figures 3.6 and 3.10 shows a reduction in the similarity values obtained by applying a measure of 3D curve shape dissimilarity compared to applying Needleman-Wunsch algorithm. Removing codes for lack of motion in this experiment could be the reason so its inclusion in the analysis of 3D curve shape dissimilarity might be needed. Length of ***step size*** affects the obtained motion similarity measure as expected.

In Section 4.3.2 we explain our basic approach while checking for overlaps between longest common sub curves. In order to compare our approach, our implementation was tested using the 10 curves in [42]. The values obtained in that work and in our implementation are shown in Table 3.3, where the first data set comes from their work and the second data set is the result of our implementation. Criteria suggested in [42] gives a more accurate analysis for 3D curves, but this analysis is not intended for human motion. Adding chain code elements that represent aspects related to human motion, i.e. lack of motion or motion that changes its trajectory in 180 degrees, would serve better to our proposed work.

A better analysis of human motion similarity must include time and space in the equation: being able to select the limbs (set of sensors) and the periods of time for a particular human motion should generate a more accurate analysis for human motion similarity. The inclusion of Labanotation in this approach will allow the analysis of human motion similarity taking time and space into consideration.

Table 3.3 - Results comparison between Bribiesca's work and our implementation



### Contributions:

In this Chapter, we provided a compared the results obtained when performing the analysis of human motion between a well-known global alignment algorithm (Needleman-Wunsch), and our 3D ChainCode implementation. Needleman-Wunsh algorithm alters the chain code representation, so an experiment where 3D ChainCode is used with different step sizes was performed. This experiment shown the importance of removing the lack of movement when performing the analysis of human motion: limbs that are almost steady contribute to the value of human motion similarity obtained during the analysis.

It is worth to mention that our 3D ChainCode implementation is not an exact implementation of Bribiesca’s work, due to the time spent when performing the analysis of similarity between 3D curves.

# Comparison of Timed 3D ChainCode with FastDTW

4.1 Dynamic Time Warping

Dynamic time warping (DTW) is a technique that finds the optimal alignment between two time series, if one of those time series may be “warped” by stretching or shrinking it along its time axis. This warping can then be used to find corresponding regions between the two time series or to determine the similarity between them. DTW is often used in speech recognition, data mining, gesture recognition, etc.

DTW is commonly used in data mining as a distance measure between time series. An example of how one time series is “warped” to another is shown in figure 4.1

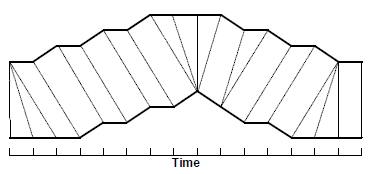


Figure 4.1 – A Warping between Two Time Series

Each vertical line connects a point in one time series to its correspondingly similar point in the other time series. If both of the time series were identical, all of the lines would be straight vertical lines because no warping would be necessary to ‘line up’ the time series. The warp path distance is a measure of the difference between the two time series after they have been warped together –Figure 4.2-, which is measured by the sum of the distances between each pair of points connected by the vertical lines in Figure 4.1 Thus, two time series that are identical except for localized stretching of the time axis will have DTW distances of zero.

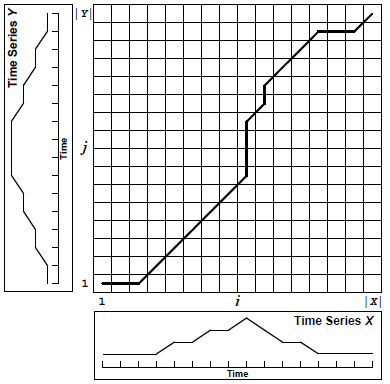


Figure 4.2 – A Cost Matrix with the Minimum-Distance Warp Path traced through it

DTW has an O(*N2*) time an space complexity that limits its usefulness to small time series. FastDTW algorithm is linear in both time and space complexity, and it avoids the brute-force dynamic programming approach of the standard DTW algorithm by using a multilevel approach. The time series are initially sampled down to a very low resolution. Once a warp path is found for the lowest resolution it is projected onto an incrementally higher resolution time series. The projected warp path is refined and projected again to yet a higher resolution. The process of refining and projecting is continued until a warp path is found for the full resolution time series.

The DTW distance is well-defined even for series of different length, and it is parametric in a ‘ground’ distance *d*, which measures the difference between samples of the series. Usually, *d*i,j = *d*(*s*i,*q*j) = (*s*i – *q*j)2 or *d*(*s*i – *q*j) = |*s*i – *q*j|

Let *s*[1;*n*] and *q*[1:*m*] be two series of length *n* and *m*, respectively. The definition of DTW is recursive:

**DTW**(*s*[1:*n*], *q*[1:*m*])2 = *d*n,m + min{**DTW**(*s*[1:*n*, *q*[1:*m* – 1])2,

**DTW**(*s*[1:*n* – 1], *q*[1:*m*])2,

**DTW**(*s*[1:*n*-1], *q*[1:*m* – 1])2}

**DTW**(*s*[1:1], q[1:1])2 = d1,1

4.2 3D ChainCode and FastDTW Comparison Results

The results obtained from our 3D ChainCode approach for the four key rehabilitation exercises were compared against the results obtained from an implementation of FastDTW technique made by Stan Salvador and Phillip Chan [43]. The comparison of those two techniques can be summarized as follows:

* Using LABAN notation on both techniques can do spatial and temporal analysis – Our representation of human movement is a set of 3D curves represented by the 3D position of the sensors used during the motion capture session. Sets of sensors for each part of the body, i.e., head, torso, arms, abdomen, and legs, are known as shown in Figure 4.3

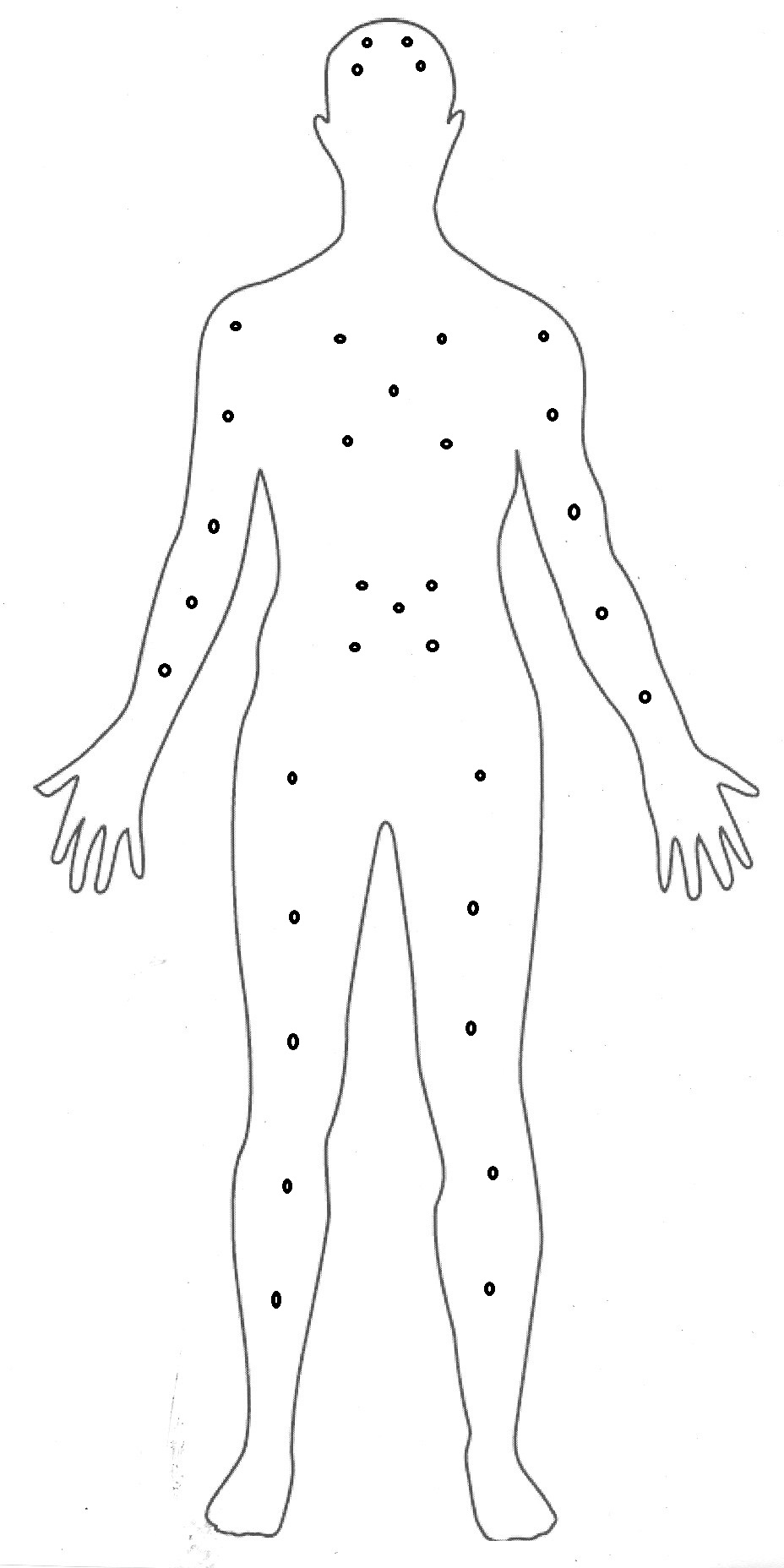


Figure 4.3 – Location for each Set of Sensors on the Human Body

The following is a table that shows where sensors were placed to capture the key rehabilitation exercises used in our experiments.

Table 4.1 – Sets of sensors for each part of the body

|  |  |
| --- | --- |
| Body Part | Sensors |
| Head | 1, 2, 3, 4 |
| Torso | 5, 6, 7, 8, 9 |
| Right arm | 10, 11, 12, 13, 14 |
| Left arm | 15, 16, 17, 18, 19 |
| Abdomen | 20, 21, 22, 23, 24 |
| Right leg | 25, 26, 27, 28, 29 |
| Left leg | 30, 31, 32, 33, 34 |

The use of those sets of sensors permit spatial analysis, i.e., using sets corresponding to right and left arms to do the analysis of them for the Shoulder Elevation and Rotation key exercise.

The LABANotation for a given exercise permits temporal analysis of it. For example, Figure 4.4 shows the LABANotation for Shoulder Elevation and Rotation Key exercise:

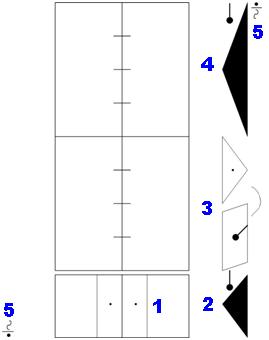


Figure 4.4 – LABANotation for Shoulder Elevation and Rotation exercise

The whole exercise can be seen as if divided in two parts: half of the time is used by the right arm, and the other half by the left one; or as if divided in eight parts: two beats to rise the right arm, two beats to lower it, and then two beats to rise the left arm followed by two beats to lower it. For simplicity, lets assume we want to analyze only the first half of the exercise: knowing how many samples were taken for this exercise, analysis of similarity can be performed by using only the first half of the sampling values for all sensors.

Finally, analysis of similarity can be performed on certain limits, and during certain periods of time, i.e., analyzing the motion of the left hand for two given sessions.

* FastDTW technique provides results faster than our 3D ChainCode approach – As an example, the run time spent to calculate 3D ChainCode similarity on sensors #9 for sessions 1 and 3 was 41 minutes and 19 seconds. Time spent by FastDTW for the same sensors, same sessions, was less than 1 second.
* 3D ChainCode results are consistent, even if one of the time series is rotated, or different limbs are compared, i.e. an arm motion vs. a leg motion – few test sequences were created to analyze the effects of rotating those sequences, and the effects of including lack of movement on them, i.e., having same 3D point values for continuous sampling rates.

Test sequences used are:

a) Slow start vs. Fast pace, a trajectory with slope – these trajectories differ only at the start and end point. In one of them, the last three sampling values for the sensor have the same 3D point. In the second trajectory, the first three sampling values for the sensor have the same 3D point. The rest of the trajectory is the same on both sequences. Figure 4.5 shows the chain code generated for these trajectories, as well as the values of similarity for both algorithms. Figures 4.6, and 4.7 show the trajectories.

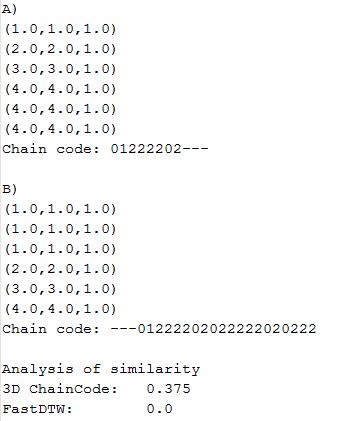


Figure 4.5 – Slow start vs. Fast pace, slope. Chain Codes and Similarity values

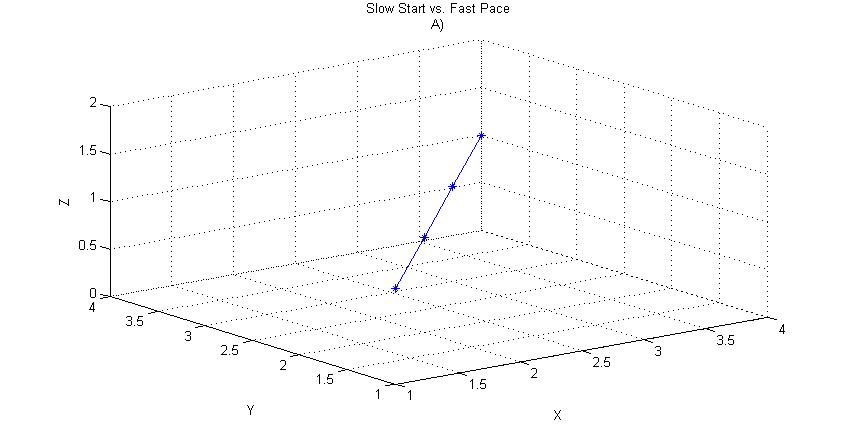


Figure 4.6 – Slow start vs. Fast pace, slope. Trajectory A

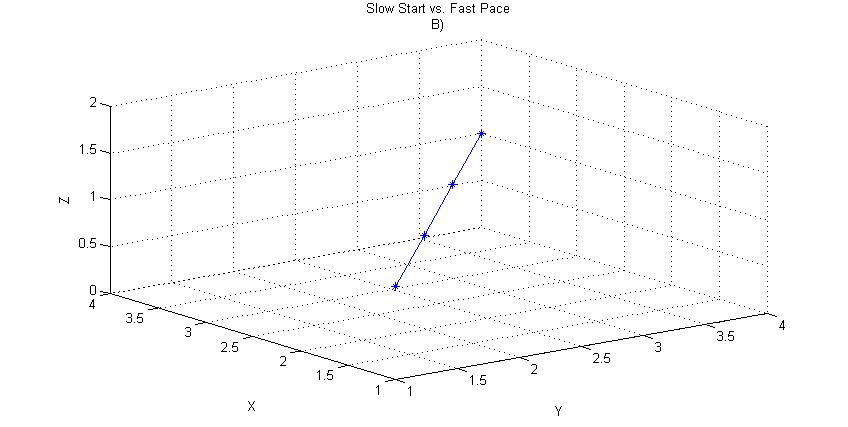


Figure 4.7 – Slow start vs. Fast pace, slope. Trajectory B

b) Slow start vs. Fast pace, a horizontal trajectory – same characteristics as the above sequence, except that this trajectory is horizontal. Figure 4.8 shows the chain code generated for these trajectories, as well as the values of similarity for both algorithms. Figures 4.9, and 4.10 show the trajectories.

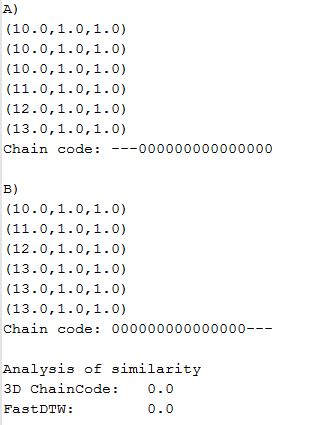


Figure 4.8 – Slow start vs. Fast pace, horizontal. Chain Codes and Similarity values

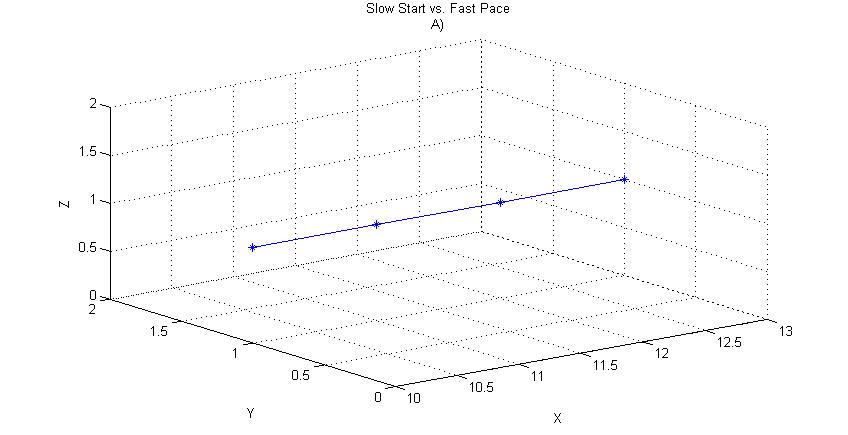


Figure 4.9 – Slow start vs. Fast pace, horizontal. Trajectory A

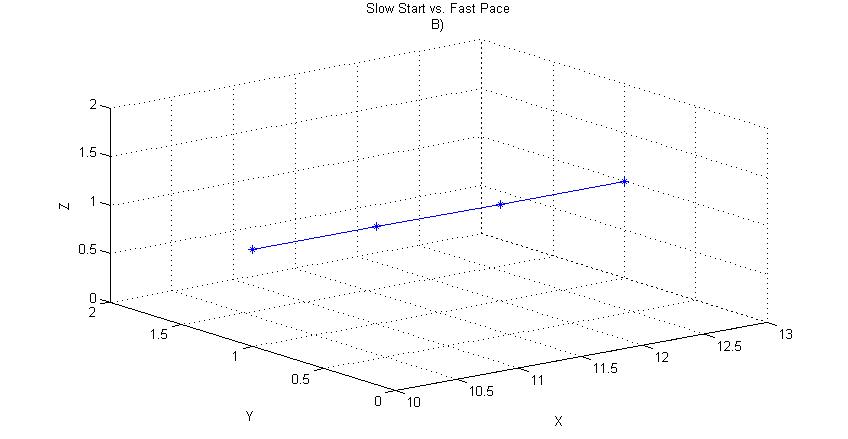


Figure 4.10 – Slow start vs. Fast pace, horizontal. Trajectory B

c) Slow start, a 90o counter clockwise rotated trajectory- in this sequence, both trajectories have the same 3D point for the first three sampling values, but one of those trajectories has been rotated 90o counter clockwise. Figure 4.11 shows the chain code generated for these trajectories, as well as the values of similarity for both algorithms. Figures 4.12, and 4.13 show the trajectories.

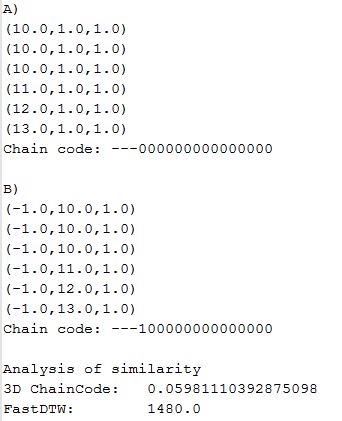


Figure 4.11 – Slow start, rotated 90o counter clockwise. Chain Code and Similarity values

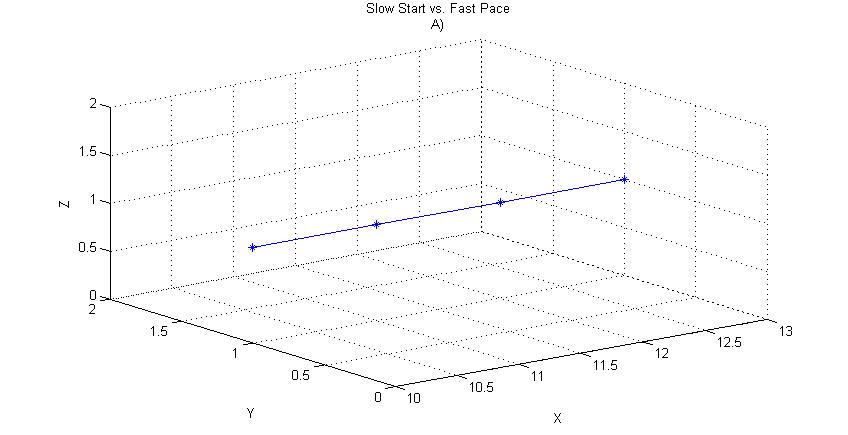


Figure 4.12 – Slow start, rotated 90o counter clockwise. Trajectory A

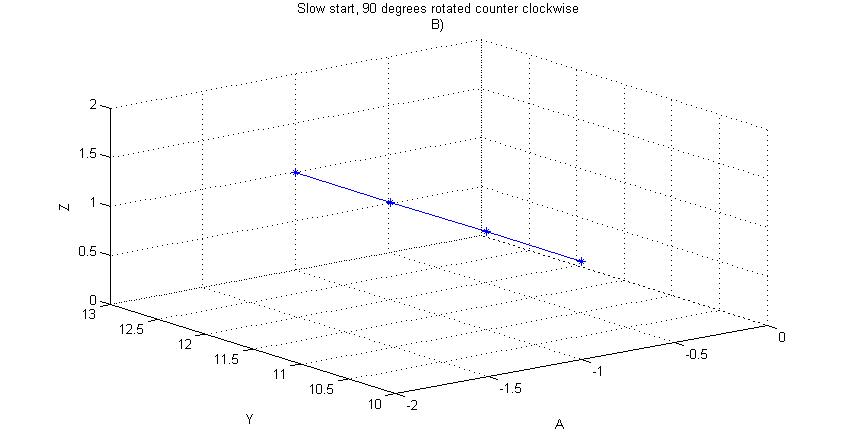


Figure 4.13 – Slow start, rotated 90o counter clockwise. Trajectory B

d) Slow start, an 180o counter clockwise rotated trajectory – same as sequences on c), but the trajectory has been rotated 180o counter clockwise. Figure 4.14 shows the chain code generated for these trajectories, as well as the values of similarity for both algorithms. Figures 4.15, and 4.16 show the trajectories.

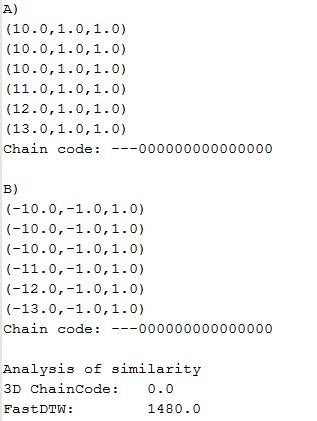


Figure 4.14 – Slow start, rotated 180o counter clockwise. Chain Code and Similarity values

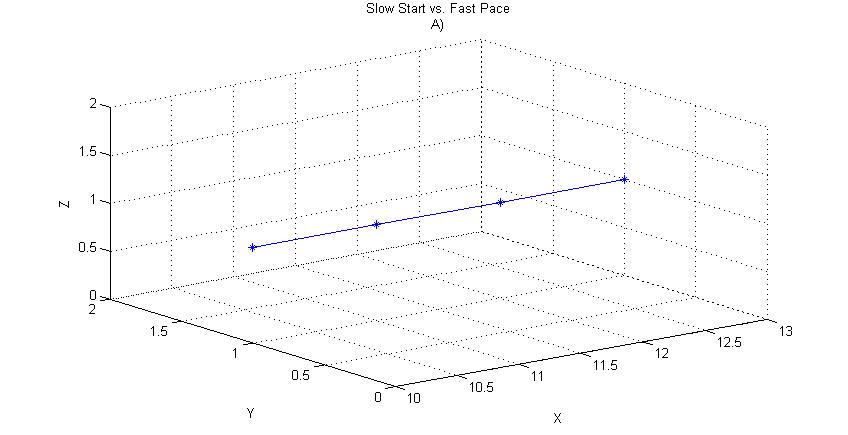


Figure 4.15 – Slow start, rotated 180o counter clockwise. Trajectory A

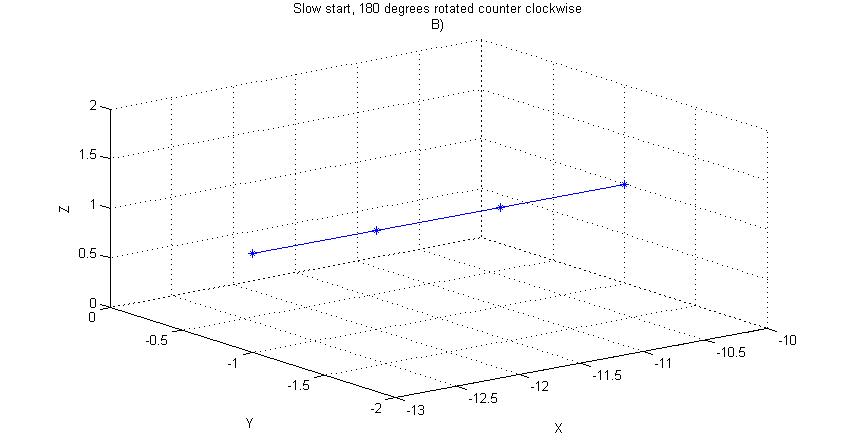


Figure 4.16 – Slow start, rotated 180o counter clockwise. Trajectory B

e) Slow start, a 270o counter clockwise rotated trajectory – same as sequences on c), but the trajectory has been rotated 270o counter clockwise. Figure 4.17 shows the chain code generated for these trajectories, as well as the values of similarity for both algorithms. Figures 4.18, and 4.19 show the trajectories.

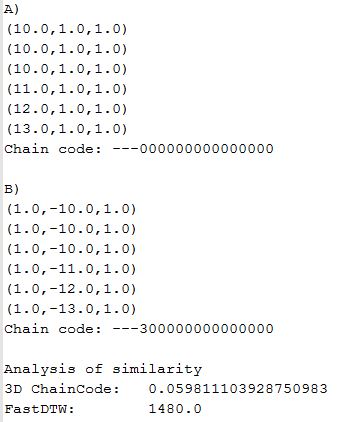


Figure 4.17 – Slow start, rotated 270o counter clockwise. Chain Code and Similarity values

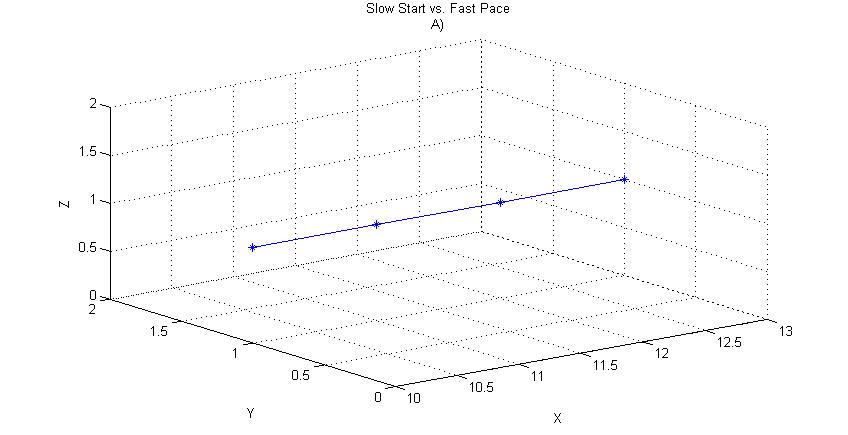


Figure 4.18 – Slow start, rotated 270o counter clockwise. Trajectory A

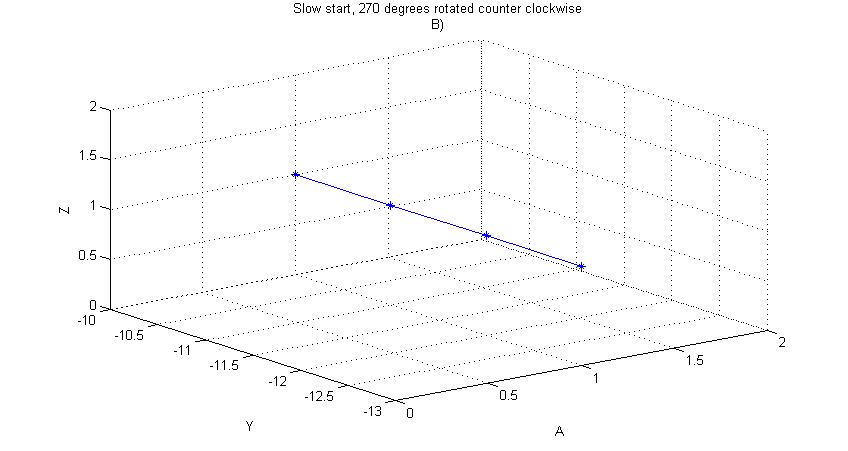


Figure 4.18 – Slow start, rotated 270o counter clockwise. Trajectory B

4.3 3D ChainCode and FastDTW Comparison Results – Shoulder Elevation and Rotation Key Rehabilitation Exercise

Shoulder Elevation and Rotation – this is one of the key rehabilitation exercises used in our experiments. A picture of the motion capture session for this exercise is shown in Figure 4.20

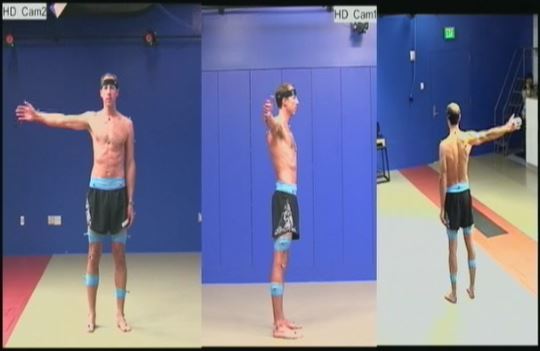


Figure 4.20 – Motion Capture Session for Shoulder Elevation and Rotation Exercise

Three sessions were captured for this exercise. The following figures show the similarity values when comparing those three sessions among each other, using 3D ChainCode algorithm. Values for the comparison of human motion similarity for the full body and for both arms are shown in the following figures. Each session was compared with each other.

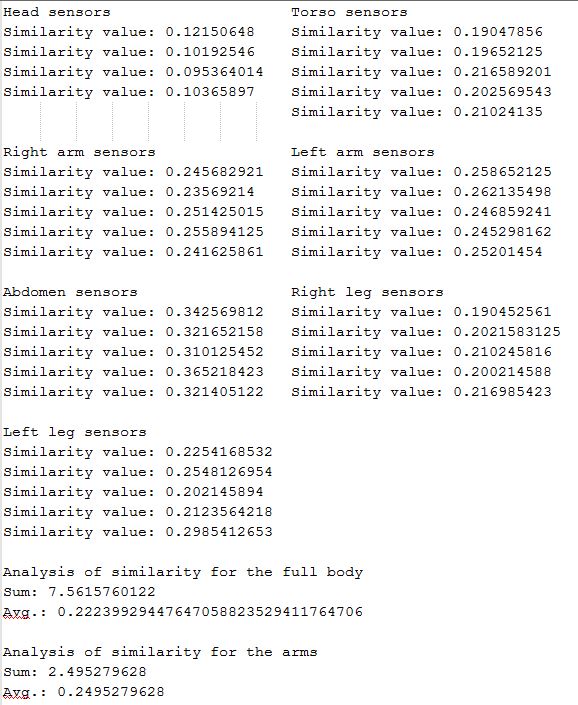


Figure 4.21 – Session 1 vs. Session 2 – 3D ChainCode

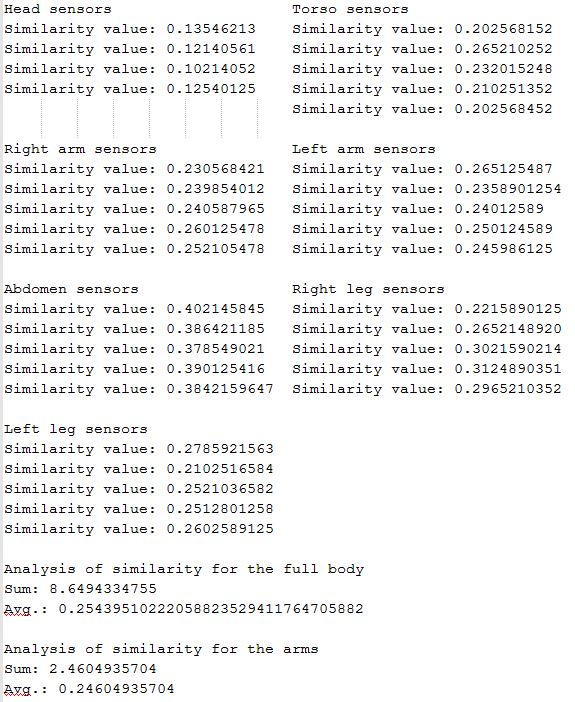


Figure 4.22 – Session 1 vs Session 3 – 3D ChainCode

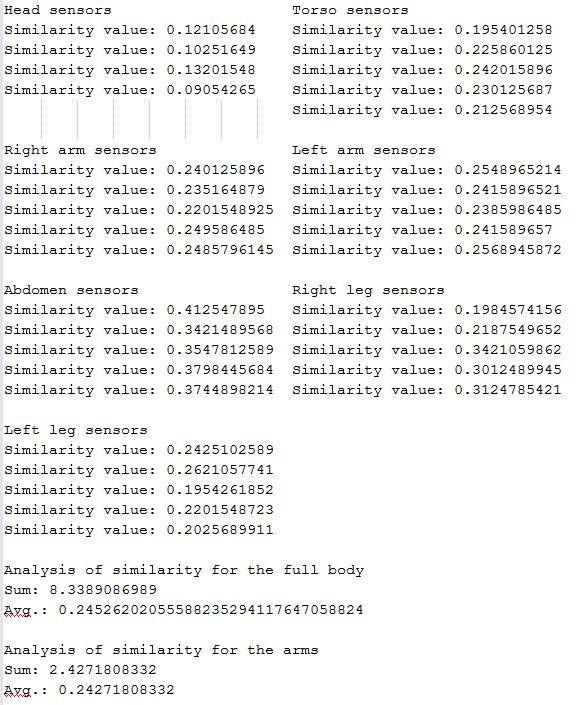


Figure 4.23 – Session 2 vs. Session 3 – 3D ChainCode

Figure 4.24 shows that when performing the analysis of similarity in the whole body, i.e., using all 34 sensors, Session 1 vs. 3 turns out to be the most similar. However, when using only the sensors for the arms, Sessions 2 vs. 3 are more similar than the rest of the comparisons.

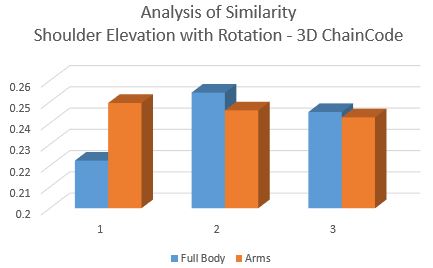


Figure 4.24 – Comparison of All Motion Sessions for Full Body and Arms – 3D ChainCode

The following figures show the similarity values when comparing those three sessions among each other, using FastDTW algorithm. Values for the comparison of human motion similarity for the full body and for both arms are shown in the following figures. Each session was compared with each other.

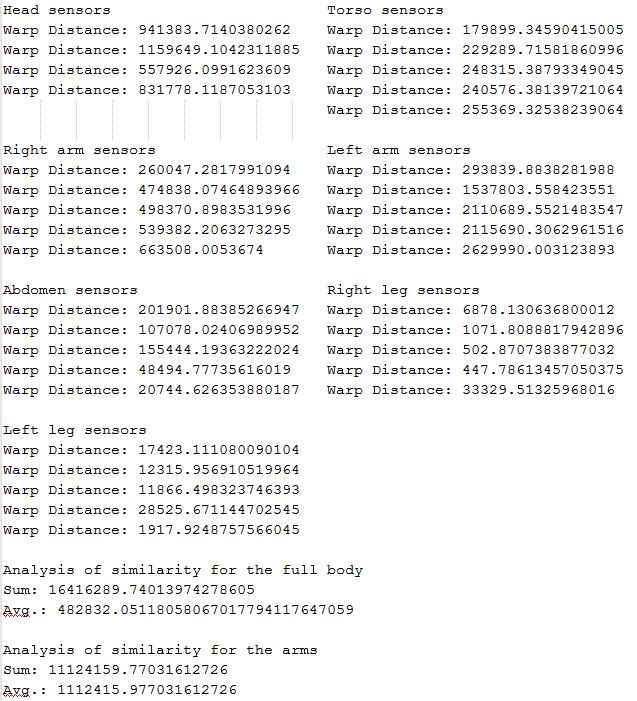


Figure 4.25 - Session 1 vs. Session 2 – FastDTW

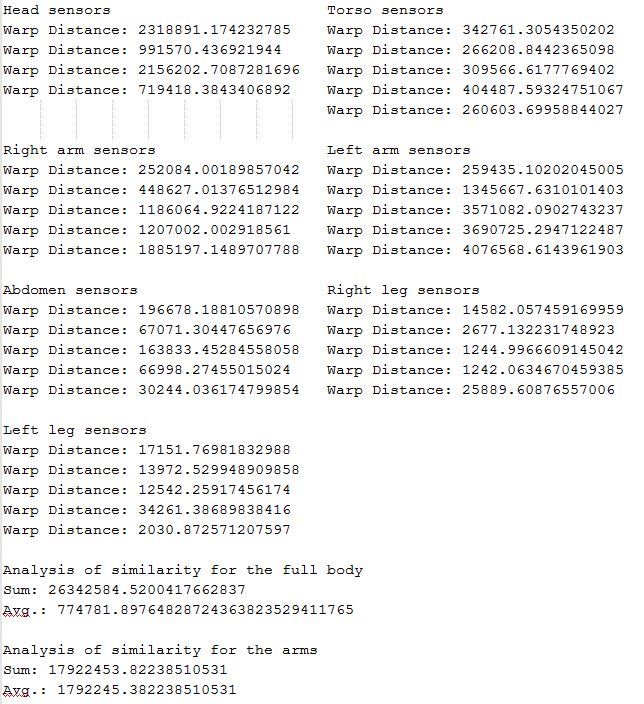


Figure 4.26 - Session 1 vs. Session 3 – FastDTW

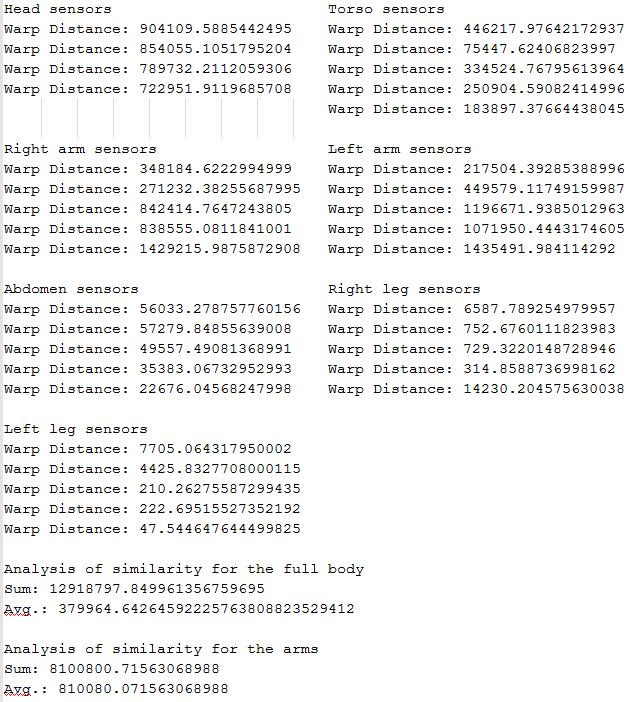


Figure 4.27 – Session 2 vs. Session 3 – FastDTW

FastDTW algorithm shows that for the analysis of both the whole body and only the arms, Session 2 vs. 3 are the more similar, as shown in Figure 4.28

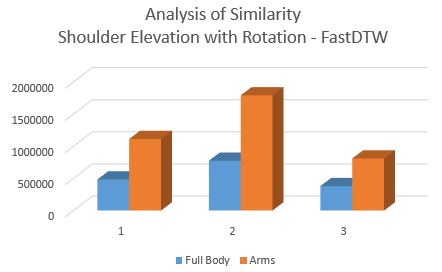


Figure 4.28 – Comparison of All Motion Sessions for Full Body and Arms - FastDTW

* Our implementation of 3D ChainCode technique cannot be used for real time analysis due to the time spent when performing the analysis of human motion. The current implementation takes time to perform the analysis of human motion similarity. Potential ways to improve execution time are reducing accuracy in terms of speed, when searching for common strings in the analysis of similarity, or using different approach when implementing the algorithm that performs the similarity step, i.e., using CUDA tools for nVidia GPUs.
* ***[This section will be updated once data is collected for the other 3 key rehabilitation exercises]***

## Contribution

In this Chapter, we have compared results from FastDTW algorithm, and our 3D ChainCode implementation. Lack of movement has been detected when generating chain code, and it has been removed so it won’t add to the similarity value when performing the analysis of human motion similarity. LABANotation was also utilized so we can compare similarity values when analyzing the whole body for this particular key rehabilitation exercise, or just the arms.

# Enhancing Laban Notation for Rehabilitation Specification

We observed that some potential requirements in the suggested key exercises cannot be fulfilled using current LABANotation.

## Proposed Enhancements Based on Mini-Squats Exercise

Figure 5.1 shows a picture of this key exercise.

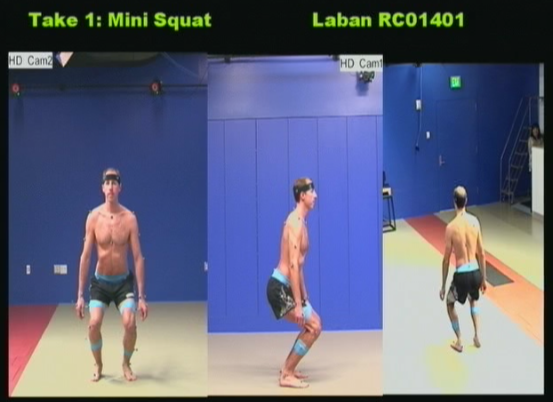


Figure 5.1 – Mini-Squat Key Rehabilitation Exercise

**Exercise a. Mini-squats: muscle group – hips**

*Description*

* Start standing with equal weight distributed between right and left legs.
* Place feet **shoulder width apart**.
* Keep torso upright, avoid bending at the waist.
* Slowly lower yourself by bending the ankles, knees, and hips.
* Return to standing.

LABANotation has ways to describe positions based on several parts of the body, previous positions, and the scenario. While this requirement could be interpreted with the following symbols shown in Figure 5.2

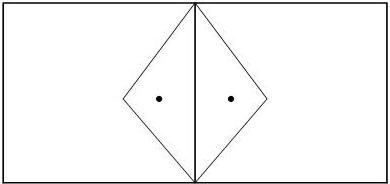


Figure 5.2 – LABANotation for ‘feet apart, equal distribution in both legs’

There is no way to prove that those symbols actually mean “*place feet shoulder width apart*”. What we propose is an enhancement to the current LABANotation set of symbols, or the use of existing symbols in LABANotation to create a new meaning. For example, in order to fulfill the requirement “*place feet shoulder width apart*”, we could use the following symbols shown in Figure 5.3

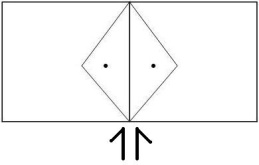


Figure 5.3 – Proposed LABANotation for ‘feet shoulder width apart, equal weight distribution in both legs’

The added symbols already exist in LABANotation, and they are used to describe both shoulders: the left and the right one. By placing ***shoulder symbols*** before the symbols “*feet apart,* equal weight distribution in both legs,” we want the meaning of such position as “*feet* ***shoulder width*** *apart,* equal weight distribution in both legs.”

## Proposed Enhancements Based on Standing-Hip Abduction Exercise

Figure 5.4 shows a picture of this key exercise.

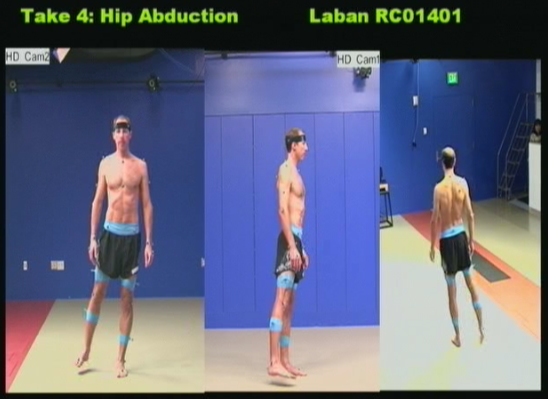


Figure 5.4 – Standing Hip Abduction Exercise

**Exercise d. Standing hip ABD (abduction): muscle group – hips**

*Description*

* Start standing with equal weight distributed between the right and left legs
* Slowly, shift your weight to the left side
* Raise the right leg out to the side **~ 12”**
* Keep the right foot facing forward
* Keep the torso upright and avoid leaning to the side

LABANotation has ways to describe three different levels (low, middle, high), and symbols to modify certain movements, like do a step forward, a big step forward, or a small step forward. It has no symbols for absolute metrics, like *raising the right leg out to the side ~ 12”*. We propose that by adding the symbol from where the absolute metric will be done, and the value, we could fulfill this particular requirement, as shown in Figure 5.5

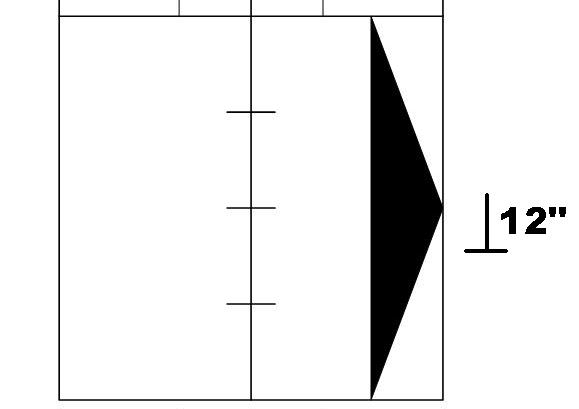


Figure 5.5 – Proposed LABANotation for raising the right leg out to the side ~ 12’’

Finally, we think that effort needs to be captured but this is hard to describe using LABANotation. Dr. James Carollo mentioned that although some exercises need to be precise about the effort, most of them do not have this requirement.

## Conclusion

In this Chapter, we suggested a new meaning for a combination of existing symbols in LABANotation, i.e. ‘stand on your feet together’ with ‘shoulder’ symbols mean ‘feet shoulder width apart. Also, we propose to add new symbols where absolute metric is needed, i.e. 12’’ on the side of the symbol for raising the leg out to the side, so it means ‘rise the leg out to the side, aprox. 12 inch’

# Design of Human Motion Tracking System Architecture for Rehabilitation Purposes

In this chapter, we propose the design for a Human Motion Tracking system architecture for Rehabilitation purposes (HMTR). The diagram for this design is shown in Figure 6.1 It is worth to mention that, as stated before, our current implementation can’t perform analysis of human motion in real time, then our current implementation could not be used in this particular design.

## Human Motion Tracking System Architecture for Rehabilitation

In this section, a system architecture design for rehabilitation exercises specification, animation, recording, and analysis, is presented.

* User interacts with the system through a GUI interface. GUI interface interacts with modules in *LABAN Simulation Block* and *User Simulation Block* to execute simulation.
* User selects the input file containing the desired LABANotation (*Read Input File Module*), and selects the desired timing for the simulation, along with the desired camera view (*Animation Settings Module*).
* User wears the sensors and goes into an initialization process (*User’s Initialization Module*) to test the functionality of those sensors.
* User is ready to follow the LABAN simulation (*LABAN Simulation Module and User Simulation Module*). User can verify how he is performing for required exercises (*Read Sensors Module, Mapping Module, Sampling Module, and Messages Generator Module*).
* User can store his performance (*Output File Module*).
* User performance can be sent to his rehab physician via Internet (*Output Module*), and get feedback from him in “real time” (*Input Module*). Exercises based on LABANotation can be shared using these modules.
* Administration of LABAN files (*LABAN Files Module*) and System generated files (*User Files Module*) is going to be handled by a System’s database.

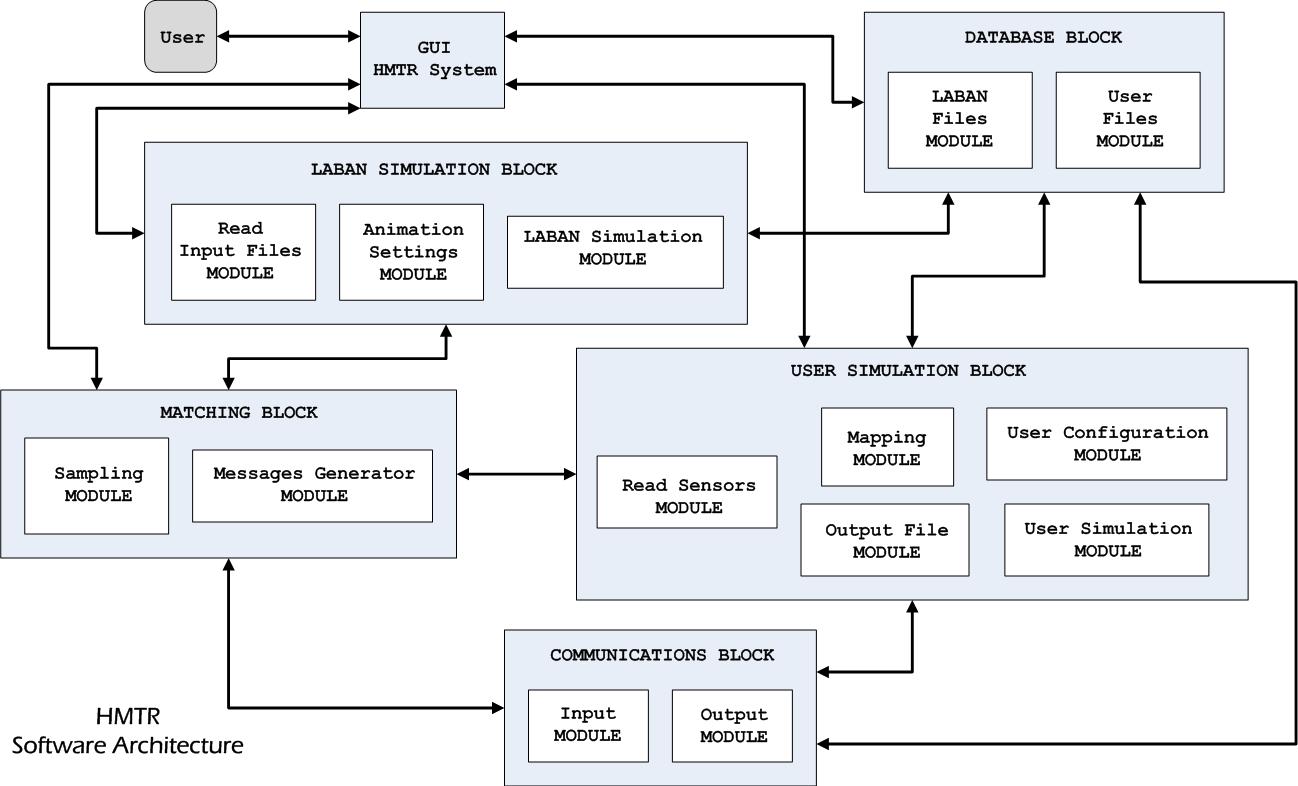


Figure 6.1 – Proposed Design for HTMR System Architecture

6.2 - **Blocks and Modules Description**

* **GUI HMTR SYSTEM**

System Graphical User Interface (GUI) will allow interaction between user and the system. Standard menus for handling files -i.e. reading, saving, etc.-, and setting parameters –i.e. camera angles, timing for simulation, etc.-, will be available. Human motion simulations based on the LABANotation and the user will be shown.

* **LABAN SIMULATION BLOCK**

This block deals with the simulation of the LABANotation.

*Modules*:

1. ***Read Input Files*** – This module will read files created by LABAN Writer software
2. ***Animation Settings*** – This will store the parameters set by the user, like camera angles and timing for simulation.
3. ***LABAN Simulation*** – Module where the simulation is being computed and shown.

* **DATA BASE BLOCK**

This block deals with the administration of LABAN and User files.

*Modules*:

1. ***LABAN Files*** – To administrate LABAN files.
2. ***User Files*** – To administrate the files generated by the HTMR system, i.e. user files.

* **USER SIMULATION BLOCK**

This block deals with the simulation of user’s movement/exercising.

*Modules*:

1. ***User Configuration*** – This will allow testing the correct functionality of the sensors.
2. ***Read Sensors*** – Module to read the data from the sensors into the system.
3. ***Mapping*** – This will translate data read from sensors into timed 3D chain codes in the system.
4. ***Output File*** – To save user motion into a file.
5. ***User Simulation*** – Module where the simulation of user’s movements are computed and shown.

* **MATCHING BLOCK**

This block will do the matching between LABAN simulation and User’s movements.

*Modules*:

1. ***Sampling*** – This module will sample from both LABAN Simulation Block and User Simulation Block, in order to compare data from both blocks in time and space.
2. ***Messages Generator*** – Based on the previous module this module will generate a message for the user, stating the current performance of the user.

* **COMMUNICATIONS BLOCK**

This block will deal with sending/receiving data through Internet.

*Modules*:

1. ***Input*** – This module is for reading data into the system.
2. ***Output*** – This module is for sending data to Internet.

6.3 **SCENARIOS**

We present three different scenarios in order to show the flow of data in the proposed HMTR System, we use a modified diagram, shown on Figure 6.2, to explain those three scenarios.

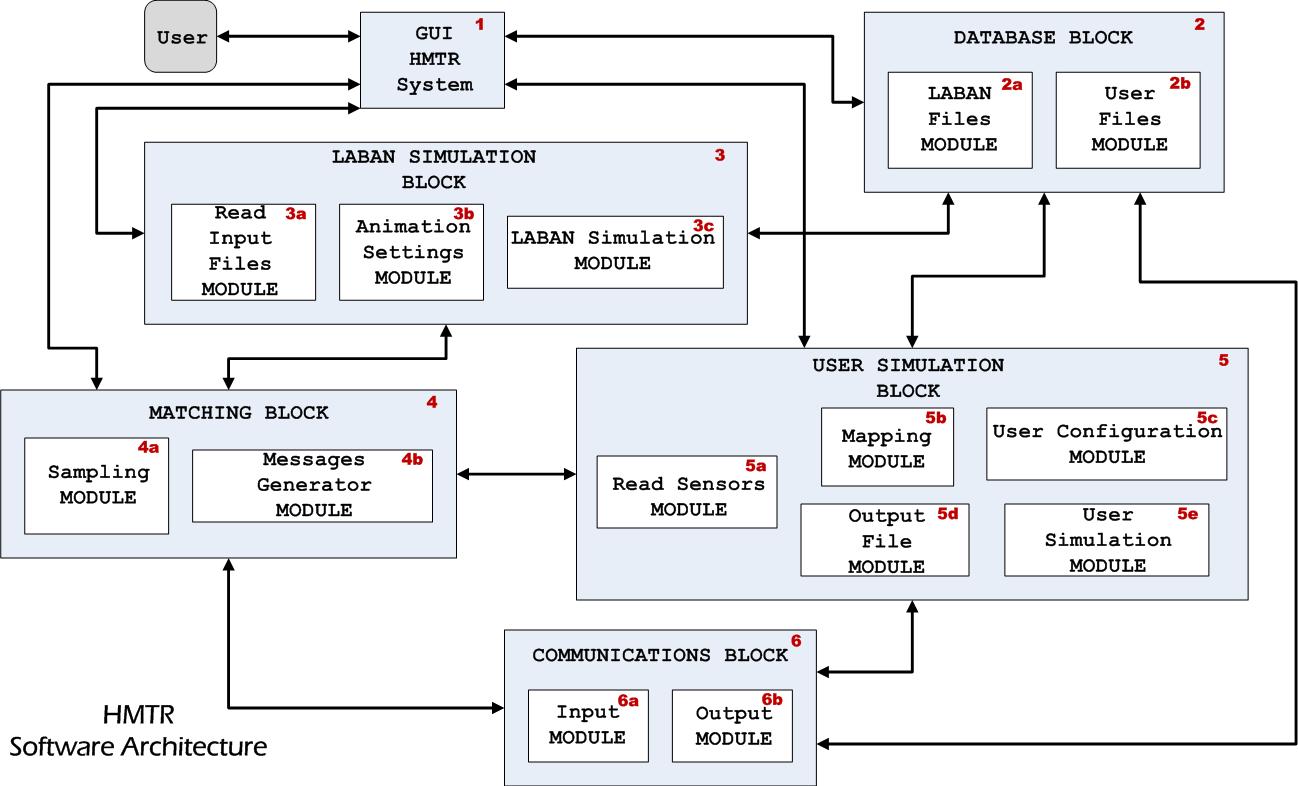


Figure 6.2 – Annotated Design for HTMR Software Architecture

* **Scenario ONE: USER PERFORMING THE EXERCISE**

1. ***Loading exercise from input file based on LABANotation***

***User*** interacts with ***GUI HMTR System (1)*** to select the desired exercise. ***GUI HMTR System*** ***(1)*** requests to load the exercise from ***DATABASE Block (2)***, which uses its ***LABAN Files Module (2a)***. ***LABAN Files Module (2a)*** sends the information to ***LABAN Simulation Block (3)*** where it is read by ***Read Input Files Module (3a)***.

1. ***Configuring HMTR System for Model Simulation***

***User*** interacts with ***GUI HMTR System (1)*** to enter required settings for model simulation, i.e. type of figure he wants, camera to be used, timing for the simulation, naming the file where performance is going to be saved, etc. ***GUI HMTR System (1)*** interacts with ***LABAN Simulation Block (3)*** who uses its ***Animation Settings Module (3b)*** to accomplish this process.

1. ***Configuring sensors for use***

***User*** wear sensors and interacts with ***GUI HMTR System (1)*** for the initialization process. ***GUI HMTR System (1)*** interacts with ***User Simulation Block (5)*** to get this process done. ***Read Sensors Module (5a)*** reads from the sensors, ***Mapping Module (5b)*** will map that input data into 3D ChainCode system data, which is sent to ***User Configuration Module (5c)*** as part of this process. Once that process is done, ***User Configuration Module (5c)*** will sends a signal to ***GUI HMTR System (1)*** so ***User*** knows that he is ready to perform the exercise.

1. ***Matching User’s performance vs. Model***

Once ***User*** is ready to do the exercise, he interacts with ***GUI HMTR System (1)*** to run the simulation. ***GUI HMTR System (1)*** triggers ***LABAN Simulation Module (3c)***, for Model simulation, and ***User Simulation Module (5e)***, for User simulation. ***LABAN Simulation Module (3c)*** starts Model simulation, and ***User Simulation Module (5e)*** interacts with ***Read Sensors Module (5a)*** and ***Mapping Module (5b)*** to read data from ***User***. ***Sampling Module (4a)*** begins to sample both ***LABAN Simulation Module (3c)*** and ***User Simulation Module (5e)*** to perform the analysis of similarity. Messages regarding user performances are generated by ***Messages Generator Module (4b)***, and then sent to ***GUI HMTR System (1)*** so they can be shown to ***User***.

* **Scenario TWO: GETTING A NEW EXERCISE FROM USER’S DOCTOR AND PERFORMING THE EXERCISE**

1. ***Getting the new exercise from Doctor***

***User*** interacts with ***GUI HMTR System (1)*** to get the new exercise from his Doctor. Here we assume ***User*** already knows that such notification has been sent. ***GUI HMTR System (1)*** interacts with ***Communications Block (6)***, and this block establishes a communication with the Doctor to get the new exercise using ***Input Module (6a)***.

1. ***Loading exercise from input file based on LABANotation***

***User*** interacts with ***GUI HMTR System (1)*** to select the desired exercise. ***GUI HMTR System*** ***(1)*** requests to load the exercise from ***DATABASE Block (2)***, which uses its ***LABAN Files Module (2a)***. ***LABAN Files Module (2a)*** sends the information to ***LABAN Simulation Block (3)*** where it is read by ***Read Input Files Module (3a)***.

1. ***Configuring HMTR System for Model Simulation***

***User*** interacts with ***GUI HMTR System (1)*** to enter required settings for model simulation, i.e. type of figure he wants, camera to be used, timing for the simulation, naming the file where performance is going to be saved, etc. ***GUI HMTR System (1)*** interacts with ***LABAN Simulation Block (3)*** who uses its ***Animation Settings Module (3b)*** to accomplish this process.

1. ***Configuring sensors for use***

***User*** wear sensors and interacts with ***GUI HMTR System (1)*** for the initialization process. ***GUI HMTR System (1)*** interacts with ***User Simulation Block (5)*** to get this process done. ***Read Sensors Module (5a)*** reads from the sensors, ***Mapping Module (5b)*** will map that input data into 3D ChainCode system data, which is sent to ***User Configuration Module (5c)*** as part of this process. Once that process is done, ***User Configuration Module (5c)*** will sends a signal to ***GUI HMTR System (1)*** so ***User*** knows that he is ready to perform the exercise.

1. ***Matching User’s performance vs. Model***

Once ***User*** is ready to do the exercise, he interacts with ***GUI HMTR System (1)*** to run the simulation. ***GUI HMTR System (1)*** triggers ***LABAN Simulation Module (3c)***, for Model simulation, and ***User Simulation Module (5e)***, for User simulation. ***LABAN Simulation Module (3c)*** starts Model simulation, and ***User Simulation Module (5e)*** interacts with ***Read Sensors Module (5a)*** and ***Mapping Module (5b)*** to read data from ***User***. ***Sampling Module (4a)*** begins to sample both ***LABAN Simulation Module (3c)*** and ***User Simulation Module (5e)*** to perform the analysis of similarity. Messages regarding user performances are generated by ***Messages Generator Module (4b)***, and then sent to ***GUI HMTR System (1)*** so they can be shown to ***User***.

* **Scenario THREE: EXERCISING AND SENDING PERFORMANCE TO DOCTOR**

1. ***Loading exercise from input file based on LABANotation***

***User*** interacts with ***GUI HMTR System (1)*** to select the desired exercise. ***GUI HMTR System*** ***(1)*** requests to load the exercise from ***DATABASE Block (2)***, which uses its ***LABAN Files Module (2a)***. ***LABAN Files Module (2a)*** sends the information to ***LABAN Simulation Block (3)*** where it is read by ***Read Input Files Module (3a)***.

1. ***Configuring HMTR System for Model Simulation***

***User*** interacts with ***GUI HMTR System (1)*** to enter required settings for model simulation, i.e. type of figure he wants, camera to be used, timing for the simulation, naming the file where performance is going to be saved, etc. ***GUI HMTR System (1)*** interacts with ***LABAN Simulation Block (3)*** who uses its ***Animation Settings Module (3b)*** to accomplish this process.

1. ***Configuring sensors for use***

***User*** wear sensors and interacts with ***GUI HMTR System (1)*** for the initialization process. ***GUI HMTR System (1)*** interacts with ***User Simulation Block (5)*** to get this process done. ***Read Sensors Module (5a)*** reads from the sensors, ***Mapping Module (5b)*** will map that input data into 3D ChainCode system data, which is sent to ***User Configuration Module (5c)*** as part of this process. Once that process is done, ***User Configuration Module (5c)*** will sends a signal to ***GUI HMTR System (1)*** so ***User*** knows that he is ready to perform the exercise.

1. ***Matching User’s performance vs. Model***

Once ***User*** is ready to do the exercise, he interacts with ***GUI HMTR System (1)*** to run the simulation. ***GUI HMTR System (1)*** triggers ***LABAN Simulation Module (3c)***, for Model simulation, and ***User Simulation Module (5e)***, for User simulation. ***LABAN Simulation Module (3c)*** starts Model simulation, and ***User Simulation Module (5e)*** interacts with ***Read Sensors Module (5a)*** and ***Mapping Module (5b)*** to read data from ***User***. ***Sampling Module (4a)*** begins to sample both ***LABAN Simulation Module (3c)*** and ***User Simulation Module (5e)*** to perform the analysis of similarity. Messages regarding user performances are generated by ***Messages Generator Module (4b)***, and then sent to ***GUI HMTR System (1)*** so they can be shown to ***User***.

Additionally, we assume ***User*** selects “*saving performance into a file*” option. By choosing that option, performance ***User*** is saved during simulation using ***Output File Module (5d)*** of ***User Simulation Block (5)***.

1. ***Sending data to user’s Doctor***

After exercise is done, ***Output File Module (5d)*** from ***User Simulation Block (5)*** will send the user’s file to ***Database Block (2)***. ***User*** will interact with ***GUI HMTR System (1)*** to select the file to be sent to his Doctor. ***GUI HMTR System (1)*** will interact with ***Database Block (2)***, and this will use ***User Files Module (2b)*** to send the file to ***Communications Block (6)***. Finally, ***Communications Block (6)*** will send the file to user’s Doctor using its ***Output Module (6b)***.

* 1. **SUGGESTED ENHANCEMENTS TO THE LabanDancer SOFTWARE**
* If we want to utilize this software for purposes of performing the analysis of human motion using our 3D ChainCode algorithm, some modifications need to be done. For example, adding an extra view that shows user working out, as shown in Figure 6.3

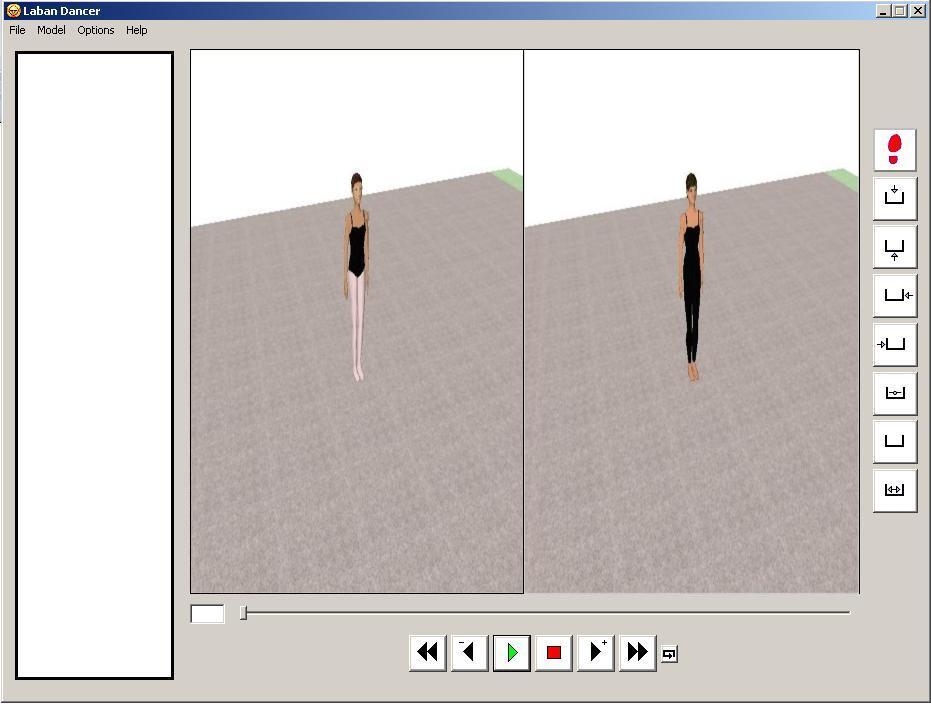


Figure 6.3 – Proposed Extra View for User Working Out

With this extra view, the user could see, side by side, the simulation using LABANotation along with his own performance.

* Another modification could be showing a text description for the current exercise, and having the LABANotation as an optional feature. Figure 6.4 shows this modification

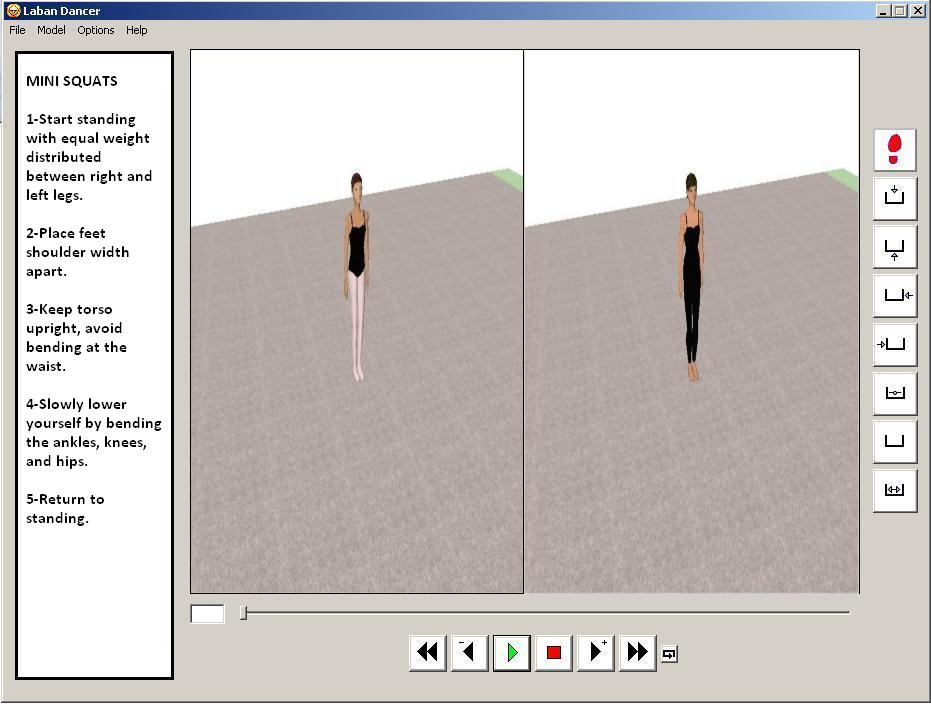


Figure 6.4 – Proposed Text Description for Exercises

* We propose to have the camera closer to the figure for rehabilitation purposes. Since this software is developed mainly for dancing performance using LABANotation, the camera tries to capture as much room as possible. For rehabilitation purposes, it is better to have a closer look to the body and limbs so executing the exercise is easier to the user.
* Since we are trying to enhance this software for rehabilitation purposes, messages given in real time regarding the performance of the user compared to the LABANotation is a good feature to have.

## Contributions

In this Chapter, we provided a system architecture design for a human motion tracking system for rehabilitation purposes. A description of the system functionality, as well as three different scenarios to show the flow of data in the system were discussed in this Chapter.

Finally, suggested enhancements for LabanDancer software [44] are discussed. The implementation of these suggestions should permit the use of this software for rehabilitation purposes as well.

# Future Research and Conclusion

In this research, several topics were explored associated with ways to analyze similarity between human motions. After some research, it was obvious that we needed to establish the human model to be used for the analysis of similarity, as well as having a valid definition of the similarity concept, one that could be applied to our research.

Although there are classical approaches to model the human body to analyze its motion, the author visualized human motion as sets of three-dimensional curves generated by the sensors in a motion capture session. More research shown that three-dimensional shapes could be analyzed to find their similarity, when those shapes were described as chain codes. This lead the author to propose a model for human motion composed by sets of three-dimensional curves that were mapped into chain codes as part of the analysis of similarity process.

But the analysis of human motion similarity, especially for rehabilitation purposes, most of the times is focused on parts of the body, and even on certain periods of time. This lead the author to find a tool that could serve as a ruler for the analysis of human motion both for spatial and time domain. LABANotation served for this purpose.

## Known Concerns and Solutions

In this section some of the concerns and issues that were encountered before and during the research process are detailed, as well as the solutions.

1. Data was needed to test our implementation. Taipei University of Arts sponsored a project where experts on gait and motion analysis, Dr. Yunyu Wang, and Dr. James Carollo respectively, identified four key exercises. Three different sessions for each key rehabilitation exercise were performed in the Center for Gait and Movement Analysis (GCMA) in Denver, CO, where motion capture sessions provided the data used on this research.
2. A complete understanding of the Measure of Shape Dissimilarity for 3D curves paper proposed by Dr. Ernesto Bribiesca was difficult. There were some gaps in the paper that were obvious when trying to implement our 3D ChainCode version of it. Emails were sent to Dr. Bribiesca in order to clarify some of the questions about the algorithm.
3. The author realized that, if implementing the algorithm as proposed by Dr. Bribiesca, run time execution of it would be huge when using data from key rehabilitation exercise. We decided to favor speed in detriment of accuracy, by not getting all maximum common couples when implementing the algorithm. It is worth to mention that our current implementation takes approximately 30 minutes to perform the analysis of human motion similarity on a pair of sensors.
4. Once obtaining results with our 3D ChainCode implementation, we needed something widely used to compare with. Dr. Terry Boult suggested the use of a DTW algorithm.

## Evaluation of Success Criteria

The following is a self-evaluation of the success criteria of this research work:

1. A proprietary tool was developed, allowing the evaluation of human motion similarity on real data for key rehabilitation exercises.
2. Research into human motion similarity and chain code representation of three-dimensional curves lead the author to consider the importance of lack of movement while analyzing motion similarity. The paper where we based our own implementation of 3D ChainCode does not consider this because the analysis is for static shapes, not three-dimensioal curves representing motion. The author was able to detect this lack of movement, and to include its value during the analysis of similarity, even though this was done by just removing such lack of movement from the analysis.
3. After implementing our 3D ChainCode tool, and performing the analysis of similarity on the read data for the key rehabilitation exercise, it was noticed that our current implementation will not serve for real time analysis of similarity, due to the time spent when calculating such similarity.

## Contributions

This research contributes to the exploration of ways to analyze human motion similarity.

1. A new model to represent human motion.
2. The use of LABANotation to help in the analysis of human motion on spatial and temporal domains.
3. The enhancement of current LABANotation when used for rehabilitation purposes.
4. A proprietary tool to perform the analysis of human motion similarity on motion capture sessions.
5. The design of a Human Motion Tracking system architecture for Rehabilitation purposes.
6. Proposed enhancements for LabanDancer software so it can be used for rehabilitation purposes.

## Future Research

There are a few areas of research that warrant further investigation. First, to expand the use of this proposed framework into another areas, i.e. dancing, video games, etc. The author expects very few modifications to this framework in order to fit into those areas.

Secondly, there is still work to do to reduce the time taken to perform the analysis in our current implementation so it can be done in real time. nVIDIA CUDA (Compute Unified Device Architecture) seems to be a way to accomplish such reduction of run time. Some works have been done using these Graphical Processor Units (GPU), i.e. a work to accelerate the similarity searching of DNA and protein molecules trough parallel alignments of their sequences was implemented on GPU, and the results of efficiency test were compared to other known implementations [45]. According to the authors, results show that it is possible to search bioinformatics databases accurately within a reasonable time.

Obtaining more real data, with exercises focused on showing that some motions are more similar than other to a given one, could help in the search of finding the meaning for the similarity value. Our dissertation provides a way to identify how similar is a given human motion compared to another one, knowing only that a value close to 0 means that those motions are very similar, and a value close to 1 means that they are very different. Understanding the meaning of values such as 0.5, 0.7, etc., could provide a better feedback when analyzing the performance of a rehabilitation patient through time. Since our work proposes to model human motion as sets of three-dimensional curves, wrapping those curves inside a tube with radius *r* could be a starting point to try to assign meaningful values to the similarity value: if user’s three-dimensional curves touches the surface of the tube, or even if it goes out of it, it means that the user needs to adjust his exercise.

## Conclusion

Analysis of human motion is needed in different areas of study. For purposes of rehabilitation, a framework that provides feedback regarding the similarity of a patient’s exercise compared to what the patient should do, is of great value. The introduction of this framework allows the analysis of human motion similarity that can be done per limb, for periods of time, or for a combination of them.

# References

[1] [Online]. Available:

http://www.thechildrenshospital.org/conditions/tests/gait/index.aspx

[2] A. S. Arnold, S. L. Delp, "Computer modeling of gait abnormalities in cerebral palsy: application to treatment planning," *Theoretical Issues in Ergonimocs Science, Vol. 6, No. 3-4, May-August 2005, 305-312.*

[3] Clinical Gait Analysis and its Role in Treatment Decision-Making; Roy B. Davis, III, PhD, Sylvia Õunpuu, MSc, Peter A. DeLuca, MD, Mark J. Romness, MD. [Online]. Available:

http://www.medscape.com/viewarticle/440148

[4] [Online]. Available:

http://thesportsmedicineclinic.com/tsmc/services/gaitana.asp

[5] [Online]. Available:

http://www.cs.odu.edu/~jrcrouch/nigel/VRGaitRehab.html

[6] Antonio Camurri, Ingrid Lagerlöf, Bualtier Volpe, "Recognizing Emotion from Dance Movement: comparison of spectator recognition and automated techniques," *Int. J. Human-Computer Studies*, 59, 2003.

[7] R. B. Davis, J. R. Davids, G. E. Gorton, M. Aiona, N. Scarborough, D. Oeffinger, C. Tylkowsky, A. Bagley, "A Minimum Standardized Gait Analysis Protocol: Development and Implementation by the Shriners Motion Analysis Laboratory Network (SMALnet)," *IEEE, 2000.*

[8] R. A. States, E. Pappas, "Precision and repeteability of the Optotrak 3020 motion measurement system," *Journal of Medical Engineering & Technology, Vol. 30, No. 1, January-February 2006, 11-16.*

[9] Sykvue Gybet, Pierre-François Marteau, "Analysis of Human Motion, Based on Reduction of Multidimensional Captured Data - Application to Hand Gesture Compression, Segmentation and Synthesis," *AMDO 2008, LNCS 5098, pp. 72-814, 2008.*

[10] Vladimir M. Zatsiorsky, "Kinematics of Human Motion," *Human Kinetics, 1st Edition, 1997.*

[11] Greg Welch, Eric Foxlin, "Motion Tracking: No Silver Bullet, but a Respectable Arsenal," *IEEE 2002.*

[12] [Online]. Available:

http://www.inition.co.uk/inition/dispatcher.php?URL\_=product\_mocaptrack\_animazoo\_gypsy7&SubCatID\_=20&model=products&action=get&tab=summary

[13] [Online]. Available:

http://us.playstation.com/ps3/playstation-move/

[14] [Online]. Available:

http://www.vicon.com/products/viconmx.html

[15] T. S. Huang, "Modeling, analysis, and visualization of nonrigid object motion," *Proc. of 10th ICPR, pages 361-364, 1990.*

[16] C. Kambhamettu, D. B. Goldgof, D. Tersopoulus, and T. S. Huang, "Nonrigid motion analysis," *In Handbook of PRIP: Computer Vision, 2, 1994.*

[17] Z. Chen and H. J. Lee, "Knowledged-guided visual perception of 3D human gait from a single image sequence," *IEEE Trans. on Systems, Man, and Cybernetics, 22(2):336-342, 1992.*

[18] A. G. Bharatkumar, K. E. Daigle, M. G. Pandy, Q. Cai, and J. K. Aggarwal, "Lower limb kinematics of human walking with the medial axis transformation," *Proc. of IEEE Computer Society Workship on Motion of Non-Rigid and Articulated Objects, 1994.*

[19] M. K. Leung and Y. H. Yang, "An empirical approach to human body motion analysis," *Tech. report 94-1, University of Saskatchewan, Saskatchewan, Canada, 1994.*

[20] K. Rohr, "Towards model-based recognition of human movements in image sequences," *CVGIP: Image Understanding, 59(1):94-115, 1994.*

[21] J. O'Rourke and N. I. Badler, "Model-based image analysis of human motion using constraint propagation," *IEEE Trans. on PAMI, 2:522-536, 1980.*

[22] K. Akita, "Image sequence analysis of real work human motion," *Pattern Recognition, 17(1):73-83, 1984.*

[23] D. M. Gavrila, "The Visual Analysis of Human Movement: A Survey," *Computer Vision and Image Understanding, Vol. 73, No. 1, January, pp. 82-98, 1999.*

[24] J. K. Aggarwal, Q. Cai, "Human Motion Analysis: A Review," *Computer Vision and Image Understanding, Vol. 73, No. 3, March, pp. 428-440, 1999.*

[25] [Online]. Available:

http://www.c3d.org

[26] P. Teitelbaum, O. B. Teitelbaum, J. Fryman, R. Maurer, "Infantile Reflexes Gone Astray in Autism".

[27] The Benesh Institute & Benesh Movement Notation [Online]. Available:

http://www.rad.org.uk/article.asp?id=114

[28] Durell Bouchard, "Automated Motion Capture Segmentation using Laban Movement Analysis," *PhD Dissertation, University of Pennsylvania, 2008.*

[29] [Online]. Available:

http://labanforanimators.wordpress.com/2008/09/01/irmgard-bartenieff/

[30] Alpha Agape Gopalai, S, M. N. Arosha Senanayake, "2D Human Motion Regeneration with Stick Figure Animation Using Accelerometers," *World Academy of Science, Engineering and Technology, 39, 2008.*

[31] Juan Carlos Niebles, Bohyung Han, Li Fei-Fei, "Efficient Extraction of Human Motion Volumes by Tracking," *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference, 2010.*

[32] [Online]. Available:

https://buffy.eecs.berkeley.edu/PHP/resabs/resabs.php?f\_year=2005&f\_submit=chapgrp&f\_chapter=14

[33] Qiang Wu, Fatima Aziz Merchant, Kenneth R. Castleman, "Microscopic Image processing, Page 189," *Academic Press 2008.*

[34] Herbert Freeman, "Computer Processing of Line-Drawing Images," *Computing Surveys, Vol. 6, No. 1, March 1974.*

[35] Ernesto Bribiesca, "A chain code for representing 3D curves," *Pattern Recognition 33 (2000) 755-765.*

[36] C.J. van Rijsbergen, "Information Retrieval," *Butterworth-Heinemann; 2nd edition (March 1979).*

[37] University of Maryland, [Online]. Available:

http://www.cs.umd.edu/Outreach/hsContest98/questions/node5.html

[38] National Institute of Standards and Technology, [Online]. Available:

http://www.itl.nist.gov/div897/sqg/dads/HTML/Levenshtein.html

[39] Natural Language Processing Group, Department of Computer Science, The University of Sheffield, [Online]. Available:

http://www.dcs.shef.ac.uk/~sam/stringmetrics.html#dice

[40] Dekang Lin, "An Information-Theoretic Definition of Similarity," Department of Computer Science, Univ. of Manitoba, Canada.

[41] Francisco Torres, J. Knight, Yunyu Wang, James Carollo, Edward Chow, Semwal Sudhansu, "Improve Rehabilitation and Patient Care with Laban Specification and Wireless Sensor Tracking," *Gait & Clinical Movement Analysis Society Conference, GCMAS 2009*.

[42] Ernesto Bribiesca, Wendy Aguilar, "A Measure of Shape Dissimilarity for 3D Curves," *Int. J. Contemp. Math. Sciences, Vol. 1, 2006, no. 15, 727-751*.

[43] Stan Salvador, Phillip Chan, “FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space,” *KDD Workshop on Mining Temporal and Sequential Data, pp. 70-80, 2004.*

[44] Lars Wilke, Tom Calvert, Rhonda Ryman, Ilene Fox, “*From Dance Notation to Human Animation: The LabanDancer Project: Motion Caption and Retrieval*,” Journal Computer Animation and Virtual Worlds, CASA 2005, Vol. 16, Issue 3-4, pages 201-211

[45] Robert Pawlowski, Bozena Malysiak-Mrozek, Stanislaw Kozielski, Dariusz Mrozek, “*Fast and Accurate Similarity Searching of Biopolymer Sequences with GPU and CUDA*”, ICA3PP 2011, Part I, LNCS 7016, pp. 230-243

# Appendix A – Algorithm for mapping 3D data sets into chain codes representing orthogonal changes of direction

**GenerateChainCode**(list3Dpoints)

Create orthogonal vectors *u* and *v*

**While** there are 3D points in list3Dpoints

A = this 3D point

B = next 3D point

**If** distance between A and B < ***step size*** **Then**

chainCode = chainCode + ***'.'***

**While** there are 3D points in list3Dpoints

B = next 3D point

**If** distance between A and B > ***step size*** **Then**

break loop

**EndIf**

chainCode = chainCode + ***'.'***

**EndWhile**

**If** no 3D points in list3Dpoints **Then**

break loop

**EndIf**

**Else**

**While** intersection of AB in any of the current cube's planes

Create vector *w*

**If** *w* = *v* **Then**

chainCode = chainCode + ***'0'***

**Else If** *w* = *u* x *v* **Then**

chainCode = chainCode + ***'1'***

**Else If** *w* = *u* **Then**

chainCode = chainCode + ***'2'***

**Else If** *w* = - (*u* x *v*) **Then**

chainCode = chainCode + ***'3'***

**Else If** *w* = - *u* **Then**

chainCode = chainCode + ***'4'***

**EndIf**

*u* = *v*

*v* = *w*

A = point of intersection of line AB and the plane

**EndWhile**

**EndIf**

**EndWhile**

**Appendix B -** **Algorithms for the analysis of human motion similarity**

B.1 Needleman-Wunsch

**HumanMotionSimilarity**(hm1, hm2)

**For** each marker on hm1 and hm2 **Do**

seq1 = chain codes for this marker on hm1

seq2 = chain codes for this marker on hm2

**Needleman-Wunsch**(seq1, seq2, seq1\_al, seq2\_al)

/\*Intuition 3\*/

**If** seq1\_al = seq2\_al **Then**

similarity\_value = 1.0

**Else**

/\*Intuition 1 and 2\*/

seq\_length = length\_of(seq1\_al)

**For** i = 1; i <= seq\_len **Do**

**If** seq1\_al[i] = seq2\_al[i] **Then**

sim = sim + 1

**EndIf**

**EndFor**

similarity\_value = sim / seq\_length

**EndIf**

sum\_sim\_value = sum\_sim\_value + similarity\_value

counter = counter + 1

**EndFor**

sum\_sim\_value = sum\_sim\_value / counter

B.2 3D curve shape dissimilarity

**HumanMotionSimilarity**(hm1, hm2)

SR1 = ReadMotion(hm1)

SR2 = ReadMotion(hm2)

**For** each marker on hm1 and hm2 **Do**

A = ReadChainCode()

B = ReadChainCode()

maxCommCouples.Clear()

maxCommCouples = findMaxCommCouples(A, B)

maxCommCouples = MCC.chooseMaxCommCouples(maxCommCouples,

A.Length, B.Length)

D = **Equation 5**

Dprime = **Equation 6**

accSim += Dprime

nSensors = nSensors + 1

**EndFor**

Similarity = accSim / nSensors;