A NEW SEARCH AND EXTRACTION TECHNIQUE FOR MOTION CAPTURE DATA

RAFIDEI BIN MOHAMAD

A thesis submitted in fulfillment of the
requirements for the award of the degree of
Master of Science (Computer Science)

Faculty of Computer Science and Information System
Universiti Teknologi Malaysia

NOVEMBER 2008

To my Darling wife, and children

Wan Zawawiah,

Amar Muhammad and Hanifa Muhammad

ABSTRACT

Motion capture is defined as measuring the position and orientation of an object in physical space by triangulating information from multiple cameras which tracks and estimate a number of retro reflective markers over time. This information is later translated into a 3 dimensional digital representation. Motion capture is applied in a wide range of field such as biomechanics, athletic analysis/training, gait analysis, computer animations, gesture recognition, sign language, music and also fine art dance/ performance. In sport science, motion capture data is used for analyzing and perfecting the sequencing mechanics of premiere athletes, as well as monitoring the recovery progress of physical therapies. Existing indexing and extraction technique for motion capture files are based on the whole body motion, where-by motion analysis in sport generally focuses on repeated movements made by specific part of limb such as the arms and legs to measure the effects of a training program. In this research, "Silat Olahraga" movements were used as a case study to apply a new technique for searching and extraction of motion capture files based on different human body segments."Silat Olahraga" is a type of sport based on "Silat", a combative art of Malay fighting and survival. A total of sixteen "Silat Olahraga" motion samples simulating four different motion categories were collected and stored as a motion capture database. The process of identification and extraction of logically related motions scattered within the data set called content-based retrieval method was performed to return results to the user. Results from the experiments show that matching motion files were successfully extracted from the motion capture library using the new algorithm based on different human body segments.

ABSTRAK

Definisi perakam pergerakan adalah mengukur kedudukan dan orientasi sesuatu objek di dalam ruang fizikal dan kemudian merakamkan maklumat tersebut ke dalam bentuk gunaan computer. Kegunaan perakam pergerakan meliputi bidang yang luas seperti biomekanik, analisa/latihan ahli sukan, analisa gaya jalan, animasi komputer, pengecaman gerak isyarat, bahasa isyarat, muzik serta seni tarian/persembahan. Masalah yang lazimnya timbul adalah dalam penyimpanan dan kemudianya mengekstrak fail perakam pergerakan di dalam pangkalan data perkam pergerakan. Pengguna biasanya terpaksa melakukan turutan klip pergerakan secara manual atau membuat carian berdasarkan kata kunci yang memerihal perlakuan sesuatu pergerakan. Penyelidik terdahulu telah memfokuskan terhadap mengindeks dan pertanyaan pangkalan data ini berdasarkan kepada seluruh pergerakan badan. Di dalam penyelidikan ini, pergerakan Silat Olahraga telah digunakan sebagai kajian kes untuk menerapkan sebuah teknik baru di dalam pencarian dan mengekstrak fail perkam pergerakan berdasarkan segmen badan manusia yang berlainan. Silat Olahraga adalah sejenis sukan berdasarkan Silat iaitu satu bentuk seni mempertahankan dan kemandirian Melayu. Pergerakan di dalam bidang sukan biasanya tertumpu kepada gerakan yang dilakukan oleh bahagian-bahagian tertentu tubuh seperti tangan atau kaki. Sebanyak enam belas sampel pergerakan Silat Olahraga yang mewakili empat kategori pergerakan telah dirakam dan disimpan sebagai pangkalan data perakam pergerakan. Kaedah pengekstrakan berdasarkan kandungan telah diterapkan bagi memaparkan hasil pencarian. Hasil daripada uji kaji yang dijalankan menunjukan fail-fail sepadan telah berjaya diekstrak daripada pangkalan data perakam pergerakan dengan menggunakan sebuah algoritma baru berdasakan segmen tubuh manusia yang belainan.

TABLE OF CONTENTS

LIST TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AMC	**-**	Acclaim Motion Capture Data
ASF	**-**	Acclaim Skeleton File
BVH	**-**	Biovision Hierarchical data
DTW	-	Dynamic Time Warping
HTR	**-**	Hierarchical Translational Rotation
LCSS	-	Longest Common Subsequence
MCML	**-**	Motion Capture Markup Language
Mocap	**-**	Motion Capture
PCA	**-**	Principle Component Analysis
UI	-	User Interface
XML	**-**	Extensible Markup Language

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

The study of human locomotion and its applications have changed since it began form the cave drawings of the Paleolithic Era. The reason for human locomotion studies during its early days were driven by the need to survive by efficiently moving from place to place, escaping from predators and hunting for food (Thomas & Eugene, 2000). Modern-day human locomotion studies have contributed to a wide range of applications ranging from military use, sport, ergonomics, and health care. In locomotion studies, according to Susan (1991) the term *biomechanics* became accepted during the early 1970s as the internationally recognized descriptor of the field of study concerned with the mechanical analysis of living organism. In sport, human locomotion studies are made to stretch the limits of an athlete when even the smallest improvement in performance is pursued eagerly (Atha, 1984). However, the advancement of human locomotion studies remains dependent on the development of new tools for observation. According to Thomas & Eugene (2000) more recently, instrumentation and computer technology have provided opportunities for the advancement of the study of human locomotion. Atha (1984) described numerous techniques for measuring motion and mentioned the co-ordinate analyzer (*motion capture* device) as a significant advance in movement analysis.

According to Maureen (2000) *Motion capture* or also known as *mocap* was originally created for military use before it was adapted into the entertainment industry since the mid 1980's. Dyer, Martin and Zulauf (1995) define *motion capture* as measuring an objects position and orientation in physical space, then recording that information into a computer usable form. According to Kovar and Gleicher, 2003; Suddha, Shrinath and Sharat, 2005 *motion capture* is the fastest way to generate rich, realistic animation data. James *et al.* (2000) describes that *Mocap* can also be applied in several other fields such as music, fine art dance/performance, sign language, gesture recognition, rehabilitation/medicine, biomechanics, special effects for live-action films and computer animation of all types as well as in defense and athletic analysis/training.

There are basically three types of motion capture systems available such as mechanical, electromagnetic and optical based system. All three systems go through the same basic process shown in **Figure 1.1**. The first step is the input where the movement of live actors either human or animal is recorded using various method depending on the type of the motion capture system used. Next, the information is processed to identify the corresponding markers of the live actor and then transferred into virtual space using specialized computer software. Finally the output is where the information is translated into a 3D trajectory computer data that contains translation and rotation information known as *motion capture* file.

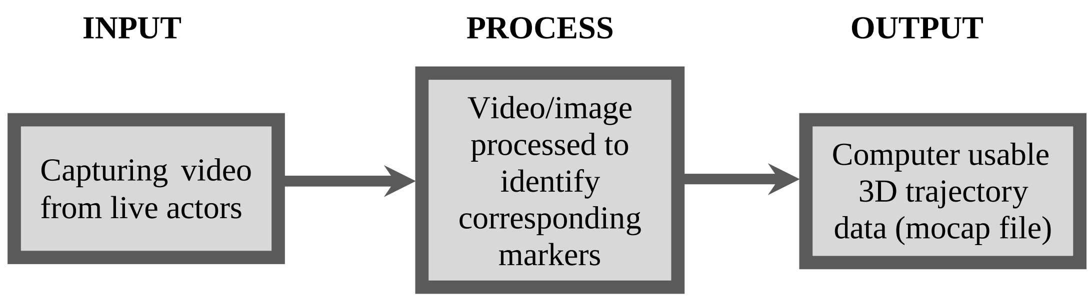


Figure 1.1 Basic Motion Capture Process.

Training or coaching sport activities such as *martial art* have been traditionally exercised by means of observations and visual assessment. This trend however is fast becoming obsolete with the rapid growth of research and developments in sports science. According to Atha (1984) visual assessment is commonly practiced because of the fact that it is low cost but not without its significant drawbacks. It is relatively slow, it has to sort what it wants from a great variety of visual information, and its usefulness is limited by the small capacity short term memory to which it has access.

Scientists at National Geographic employ state of the art mocap system to measure and map the speed, force, range and impact of *martial art* fighter techniques (Yancey, 2006). This is a practical step to take *martial art* to the next level since most of the popular self-defense techniques such as *Tae Kwon Do, Karate* and also *'Silat Olahraga'* have been adapted to become a new form of sport which raises the issue of safety and competitiveness in term of training and coaching. The goal of this research is to apply scientific method towards *'silat olahraga'* training and coaching technique.

1.2 Problem Statement

1.2.1 Even though motion capture is applied into so many fields by creating physically perfect motions, it has a few significant weaknesses. According to Hyun-Sook & Yilbyung (2004) firstly, it has low flexibility, secondly the captured data can have different data formats depending on the motion capture system which was employed and thirdly, commercially available motion capture libraries are difficult to use as they often include hundreds of examples. Shih-Pin *et al*. (2003), states that motion capture sessions are not only costly but also a labor intensive process thus, promotes the usability of the motion data.

1.2.2 In the field of animation and gaming industry, it is common that motion clips are captured to be used for a particular project or stored in a *mocap* library. These clips can either be used as the whole range of motion sequence or as part of a motion synthesis. In sport science, mocap data is used for analyzing and perfecting the sequencing mechanics of premier athletes, as well as monitoring the recovery progress of physical therapies. This simply means that a vast collection of *motion capture* files or *motion capture* library are accumulated. Currently, motion data are often stored in small clips to allow for easy hand sequencing and searches based on keywords describing the behavior (Jernej *et al.*, 2004; Tanco and Hilton, 2000). However, according to Hyun-Sook & Yilbyung (2004), Carlos (2001), Tanco and Hilton, (2000) a *motion capture* library is not good at storing and retrieving *motion capture* files as a database. This calls for an immediate need for tools that index, query, compress, annotate and organize these datasets (Feng *et al.,* 2003)

1.2.3 During a biomechanics study for the purpose of performance improvements among athletes such as the one done by Gregory *et al*. (2005), a controlled cohort repeated-measures experimental method was used to study the effects of a specific training program. This involves measuring the before and after effects of such training program and also comparisons among the athletes who participated in the experiment. With a collection of motion sequence in hand, according to Mu ller, Ro der, & Clausen (2005) one has to solve the fundamental problem of *identifying* and *extracting* suitable motion clips from the database on hand. The user usually resorts to describing the motion clips to be retrieved in various ways at different semantic levels. One possible specification could be a rough textual description such as "a kick of the right followed by a punch". Another query mode would involve a short query motion clip, the task being to retrieve all clips in the database containing parts or aspects similar to the query. Muller, Roder, & Clausen (2005) refer this problem as *content-based retrieval*. This research will attempt to apply automatic *searching* and *extraction* of matching human motion data based on statistical properties as an alternative to hand sequencing.

1.2.4 Due to the success of motion capture, many organizations developed their own motion capture file formats which were designed for specific hardware. Because of this, according to Meredith & Maddock (2005), Alexander *et al*. (2007), there are no standard motion capture file formats and according to Carlos (2001) since there are no industry-wide standard for archiving and exchanging of motion capture data, it is hard to predict how different software react with the motion capture files. This is why TRC file format was used to develop the motion search and extraction system for this research since Institut Sukan Negara and Universiti Putra Malaysia both uses Motion Analysis Corporation motion capture system that generates TRC files.

1.2.5 Most of the motion capture file search and retrieval research sited in this research including Kovar and Gliecher (2004) are concerned with character animation and motion synthesis where-by according to Bobby *et al.* (1997), in the field of animation, highly abstracted applications of *motion capture data*, analogous to puppetry, are primarily concerned with motion character, and only secondarily concerned with fidelity or accuracy where by additionally, accurate motion analysis is important to the biomechanics community.

1.3 Research Objectives

The research objectives are as follows:-

1.3.1 To study and identify a suitable *search* and *extraction* method for biomechanics motion sequence.

1.3.2 To develop and to test a motion search and extraction system for biomechanics data.

1.3.3 To study the effectiveness of the biomechanics data search and extraction system in a motion capture database.

1.4 Scope of Research

In order to achieve the best possible result from this research, we need to identify the scope in which the research will focus on. The scopes identified are concerning the software, types of data, techniques to be applied and the main topics to be covered related to the research area are as follows:-

1. This research will use the Tracked Row Column (TRC) file format for the system development and sample collection of motion capture files based on Motion Analysis Corporation motion capture System employed by Universiti Putra Malaysia where the motion data/samples were collected. The reason for using the TRC file format is explained on the problem statement 1.2.4.

2. Samples collected on basic 'silat' movements such as punching, kicking and elbowing will be manually segmented to simulate the motion library environment where motion sequences are usually named based on the behavior or the performers.

3. A prototype to apply the motion extraction technique identified from this research will be developed using Microsoft Visual Basic and Microsoft Access Database.

1.4 Research Contribution

The result of this research has generated two main contributions as far as knowledge is concerned:-

1. In this research, a content-based retrieval method for search and motion data extraction has been identified and applied for sequencing/identifying biomechanics data in a motion capture library or mocap database.

2. An automated content-based retrieval for search and motion data extraction has been developed to speed up the process of identifying and extracting suitable biomechanics data from a motion capture library for the purpose of analyzing and perfecting the sequencing mechanics of premier athletes, in relation to sports science.

1.6 Thesis Structure

The structures of the thesis are as follows. The first chapter will cover the introduction explaining briefly on *motion capture*. Also included in this chapter are the problem statement, research objectives, scopes and finally the research contributions. Chapter 2 will discuss the on the past research done on *motion capture* and other related topics to find the best methodology to address the research objectives mentioned in the first chapter. In Chapter 3 will be on the description of the methodologies of this research. Chapter 4 will explain on the implementation of the system development based on the methodology discussed on Chapter 3. Chapter 5 will cover on the test carried on the motion extraction system developed in this research and finally, Chapter 6 concludes this thesis with future directions. The general flow of the thesis is shown in *figure 1.2* below.

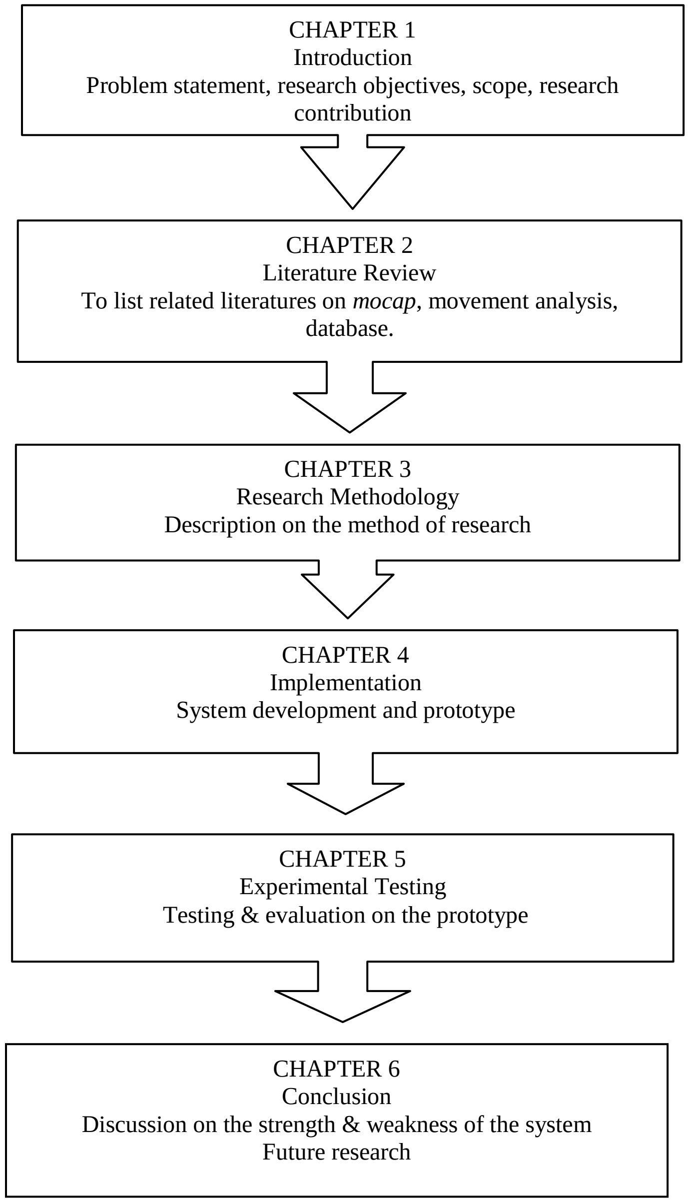


Figure 1.2 Thesis Structure

CHAPTER 2

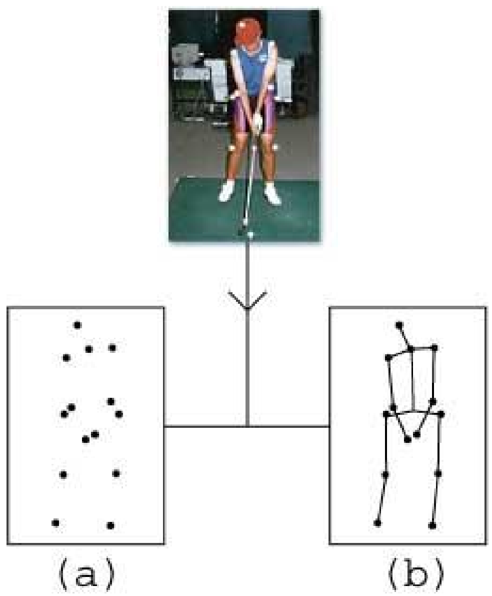
LITERATURE REVIEW

2.1 Introduction

This chapter is a discussion of the related issues on *motion capture* data *search* and *extraction*. Since *motion capture* is used in a wide range of applications, it comes to no surprise that the users are faced with common issues. In the animation and gaming industry according to Carlos (2001) the lack of an industry wide standard for archiving and exchanging *motion capture* data has contributed to the reluctance of animators to utilize this technology. This is because the same *motion capture* files react differently on different animation packages. So, before going deeper on the motion data *search* and *extraction* process, perhaps it is best to understand the basic overview *motion capture* system.

2.2 Motion Capture File Formats

Motion capture file formats can be roughly divided into two; The Tracker Format and the Skeleton Format, according to the method for processing and the system used to get the *motion capture* data. The former only has three-dimensional location values and accepts the Adaptive Optics Associates (AOA), Coordinate 3D (C3D) and Tracked Row Column (TRC) formats. The Latter has skeleton information as well as the three-dimensional location values and accepts the BVH, Biovision Data (BVA), Htr, ASF/AMC, Lamsoft Magnetic format BRD, Polhemous DAT files and Ascension ASC file formats (Hyun-Sook Chun and Yilbyung Lee, 2004).



(a)	(b)
Tracker Format Translation = Tx, Ty, Tz	Skeleton Format Hierarchical = root, Lhip, Rhip, etc Translation = Tx, Ty, Tz Rotation = Rx, Ry, Rz

Figure 2.1 Basic Motion Capture File Format

2.3 Motion Capture Data Extraction Techniques

Motion capture data are high-quality and solidified, designed for specific characters or circumstances making it difficult to edit or modified for other purposes. Its' data format vary depending on the *motion capture* system employed where each system defines its own data format to express the capture contents. Studies on *motion capture* are done basically to solve problems related to the issues mentioned above. The next section, reviews the method or approach taken by other researchers that are relevant to this study.

2.3.1 XML Based Approach

In an effort to make *motion capture* data become more versatile, Carlos (2001) applied immediate translation, indexing and archiving of *motion capture* files in a network environment by translating the file into XML document. By parsing the data provided by the user into an XML document and the developing a set of schemas for each viable output, the application would be able to use a single content file, which could serve multiple animation applications. The system handles three motion capture file formats i.e. BVA, Newtek's Lightwave and AOA. Carlos (2001) developed a structure that would hold the necessary data to reconstruct any of the input files before ASP (Microsoft's Active Server Pages) application could be used to parse out in-coming motion capture files into XML.

```
- <Mocap>
  - <General_Information>
      <NumberNodes />
      <NumberFrames />
       <SampleFrequency
         />
    </General_Information>
  - <Nodes>
    - <NodeName>
        <XTrans />
        <YTrans />
        <ZTrans />
        <XRot />
        <YRot />
        <ZRot />
        <XScale />
        <YScale />
        <ZScale />
      </NodeName>
    </Nodes>
  </Mocap>
```

Figure 2.2 Morales XML Motion Capture File Structure

2.3.2 MCML Based Approach

Hyun-Sook & Yilbyung (2004) solved the problem of *searching* and *retrieving motion capture* library by applying MCML or Motion Capture Markup Language. This is actually an extension of Carlos (2001) work based on XML to define tags to integrate different type of *motion capture* file format which includes hierarchical data format such as Acclaim Skeleton File (ASF)/Acclaim Motion Capture Data (AMC), Biovision Hierarchical data (BVH) and Hierarchical Translational Rotation (HTR). By having a standard format, the duplication of *motion capture* files can be eliminated and this creates a compact-sized motion database. A stored MCML document can be viewed either in the form of text or as a hierarchy structure in a database. Therefore, animators can examine the contents of an MCML document without loading it into main memory.

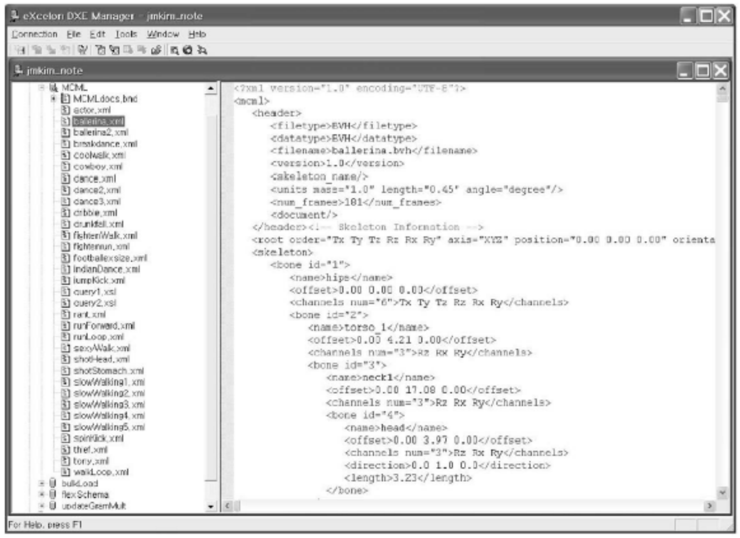


Figure 2.3 MCML Documents stored in XML Database

2.3.3 Motion Normalization Approach

According to Yan Gao *et al.* (2005), the difference between skeletons in motion data will affect the result of motion *retrieval* or motion graph. So it is desirable if each skeletal segment for all motion in motion database shares a common length. Yan Gao *et al.* (2005) proposed a method to preprocess motion data in motion database with common skeleton length. Similar to image normalization, the same principle is applied to motion normalization. The motion normalization is applied to adapt each skeletal segment of motion data in a database with common length automatically. This is done by first adjusting the root positions in an online manner and then cleaning up foot skates using Kovar and Gleicher 2003) post-process algorithm.

2.3.4 Motion Data Indexing

Shih-Pin *et al*. (2003) proposed posture feature indexing to provide a novel framework to synthesize complex human motion. For *motion index*, posture features of each frame are extracted and then mapped into a multidimensional vector. For motion synthesis, a user interface graphical stickman is devised for specifying postures of start frame and end frame respectively. Then a path finding algorithm is applied to search for smooth paths in the multidimensional *index* space. A synthesized motion can be obtained by first collecting proper motion clips along a chosen path and then polishing consecutive motion clips using constraint-based transition method.

Another *motion data indexing* approach was introduced by Guodong *et al*. (2005). Through this method, a data-driven approach for representing, compressing and *indexing human-motion* databases was used. The modeling approach is based on piecewise-linear components that are determined via a divisive clustering method. Selection of the appropriate linear model is determined automatically via a classifier using subspace of the most significant, or principle features (markers).

Michail *et al*. (2005) came up with the idea of *multidimensional time-series indexing*. The purpose was to present an efficient, compact, external memory *indexing* technique for fast discovery of similar trajectories. Their technique offers enhanced robustness, particularly for noisy data encountered often in real-world applications by applying

LCSS model for similarity measures but at the same time their *index* also supports Euclidean and DTW distance measures as well.

2.3.5 Cluster Graph Approach

Suddha, Shrinath and Sharat (2005) described a *query by example* and *cluster graph* scheme to enable search and transition for *mocap* data in their research. They view motion data as consisting of a bundle of signals and identify continuous monotonic portions (termed *fragments*) from the signals to retrieve queries. They considered *fragments* as important in their method because *fragments* enable “mining” of possibly unrelated clips and enables robust scaling. Individual clips are automatically chopped and also collects similar sub-clip sequences. All such sequences are collected into nodes in a *cluster graph*. This method allows animator to *query by example* the motion database.

2.3.6 Content-based Motion Retrieval Approach

This approach was inspired by *content-based retrieval* algorithm for retrieving multimedia data, based on the features automatically extracted from the content. Desired motions are obtained by submitting a similar sample in the form of a *motion capture* or a scripted one. Feng Liu *et al.* (2003); Kovar and Gliecher (2004); Muller, Roder and Clausen (2005) presented their methods of content-based retrieval which are discussed in detail on section 2.4.

2.4 Content-based Retrieval

According to Kovar and Gliecher (2004) the search problem is related to time sequence, which has been studied by the database community for over a decade. Given a distance metric and a query time sequence, the task is to search a database for time sequences whose distance to the query is either below a threshold ε or among the k smallest. In their research, the strategy was to find "close" matches that are numerically similar to the query and uses them as new queries to find more distant matches. Their method was developed to apply motion blending to the extracted motion that gives the user direct control over relevant motion properties.

Kovar and Gliecher (2004) uses two criteria to determine numerical similarity among two motion segments:

1. Corresponding frames should have similar skeleton poses.
2. Frame correspondences should be easy to identify. That is, related events in the motions should be clearly recognizable.

An analysis of the time alignment determines whether two motion segments satisfy the numerical similarity criteria. Given a distance function *d* for individual frames, dynamic programming can be used to compute an optimal time alignment that minimizes the total distance between matched frames.

Feng Liu *et al*. (2003) developed hierarchical tree clusters of motions by extracting key-frames of motion in a database because motion resembles video in their representation (i.e., both can be represented as posture/frames sequences) with deeper levels of the tree corresponding to joints deeper in the skeletal hierarchy. Within the hierarchical motion description, the motion of a parent joint may induce those of its children joints, whereas that of a child joint is unable to influence its parent joints. Their algorithm uses direct numerical comparison to determine similarity between two motions. According to Feng Liu *et al*. (2003) most similarity measures of motions are defined based on their corresponding key-frame sequences. A simple method is the Nearest Center (NC) algorithm.

Muller, Roder and Clausen (2005) introduced various kinds of qualitative features describing geometric relation between specified body points of a pose and show how these features induce a time segmentation of *motion capture* data stream. By incorporating spatio-temporal invariance into the geometric features and adaptive segments to adopt efficient indexing methods allowing for flexible and efficient content-based retrieval and browsing in huge motion capture database.

2.4 TRC Data Format

This section, reviews the TRC (Tracked Row Column) file format to study the file structure. The file *search* and *extraction* algorithm will be based on the position data of specified markers. As shown on **Figure 2.4** there are three main components in a TRC file format, defined as file header, data header and positional data. Please refer to **Table 2.1** for the details.

PathFileTy	4	(X/Y/Z)	d:\intern_kutpm\260405\trc\duduk_minum2.trc						
DataRate	CameraRa	NumFrame	NumMarke	Units	OrigDataR	OrigDataS	OrigNumFrames		
60	60	241	33	mm	60	1	750		
Frame#	Time	LFrontHead			RFrontHead			LBackHead	
		X1	Y1	Z1	X2	Y2	Z2	X3	Y3
1	4.317	657.5854	1242.937	-22.4315	791.6017	1261.751	-21.0561	666.4232	1190.246
2	4.333	654.7773	1242.684	-22.5112	787.9257	1262.397	-21.6838	662.9822	1189.516
3	4.35	650.5796	1242.175	-23.4569	783.4551	1262.714	-22.229	660.2784	1188.592
4	4.367	646.6044	1242.737	-22.8247	779.6567	1262.985	-22.1212	656.9877	1187.816
5	4.383	642.6852	1241.802	-24.237	776.3693	1263.074	-23.6377	653.6027	1186.791
6	4.4	639.2983	1241.813	-25.527	772.5653	1263.355	-25.2187	650.1641	1186.094
7	4.417	635.6042	1240.258	-27.8713	768.6862	1263.059	-26.7848	647.4609	1185.686
8	4.433	632.2172	1239.777	-29.9081	765.161	1263.07	-29.1351	644.6481	1185.149
9	4.45	628.5794	1239.733	-32.6419	761.9251	1262.671	-31.516	640.5477	1183.763
10	4.467	625.9575	1239.288	-35.4142	758.7261	1261.674	-34.6919	637.4854	1183.272
11	4.483	621.708	1237.773	-37.859	755.6989	1260.928	-37.4011	634.1581	1183.315
12	4.5	620.1788	1238.086	-42.5888	752.5591	1260.426	-41.7637	631.2387	1183.279
13	4.517	616.4313	1236.736	-46.5103	749.5506	1259.811	-44.7497	628.7905	1182.994
14	4.533	614.2166	1235.643	-51.0422	746.9777	1259.647	-49.6131	625.2748	1182.358
15	4.55	610.3926	1234.221	-53.9652	743.4327	1258.378	-54.0765	622.9103	1182.456
16	4.567	609.0518	1233.41	-59.9803	741.4177	1257.726	-58.2588	620.1668	1182.464

File Header

Data Header

Position Data

Figure 2.4 TRC File Sample

Table 2.1 TRC File Structure

FILE HEADER	
First row	File path
Second row	Headings for data rate, camera rate, number of frames, number of marker units, original data rate, original data start and original number of frames.
Third row	actual number/value for the 2nd row's information
DATA HEADER	
Fourth row	Headings for frame number, time and markers name
Fifth row	Markers dimension headings (x, y, z)
POSITION DATA	
Sixth row onwards	The actual position and time values

2.6 Summary

Through out this chapter, a total of six approaches on how to solve *motion capture search* and *retrieval* problem done by other researchers that is relevant to this study has been presented. The techniques discussed were the XML, MCML, Motion Normalization, Motion Data Indexing, Cluster Graph and the Content-based Retrieval approach. The reason for this is actually to make a study and comparisons as a guide to select the best method to be implemented in the *"silat olahraga*" or sport science domain.

Based on the literatures presented, *content-based retrieval* has been identified as a potential approach to be applied to this research. All the literatures reviewed in this study used the skeleton type of *motion capture* files for their system development. This is because their main goal was to assist animators in selecting suitable motion clips to be mapped onto new characters or as part of a motion synthesis. *The tracker format* is considered as raw *motion capture* files and is being used to construct skeleton type *motion capture* files. In the field of sport science, the tracker file format can either be used directly as a performance measurement or mapped onto biomechanical models to provide kinematical data for clinical gait analysis (Cerveri , Pedotti and Ferrigno, 2003). From the observations made at the *Institut Sukan Negara (ISN)* and *Pusat Pembangunan Maklumat dan Komunikasi*, *Universiti Putra Malaysia (UPM),* both owns *Motion Analysis Corporation* © motion capture system that generates TRC *motion capture* file format. All the motion files are basically stored in a *motion capture* library where the file naming is either based on the movement type or by the performer's name. This provides the perfect case studies to apply to this research based on the problem statement and research objectives presented on the first chapter.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

This chapter will discuss the design of the prototype for *motion capture search* and *extraction* system. Based on the previous chapter, the best approach to be implemented for the system development is the *content-based retrieval* technique based on Kovar and Gleicher (2004) that uses the *query by example.* The similarity measure of motion for this system is defined based on their corresponding key-frame sequence method using Nearest Center (NC) algorithm. **Figure 3.1** shows the basic flow for a biomechanics archiving system where MocapXtractPro can be implemented. The MocapXtractPro *search* and *retrieving* process, consists of three main steps. The first step is *the query* that involves a segment of motion data to find similar action or sequence while in the second step of the system automatically searches for matching files based on NC algorithm and finally in the last step, the system extracts the selected features into a database.

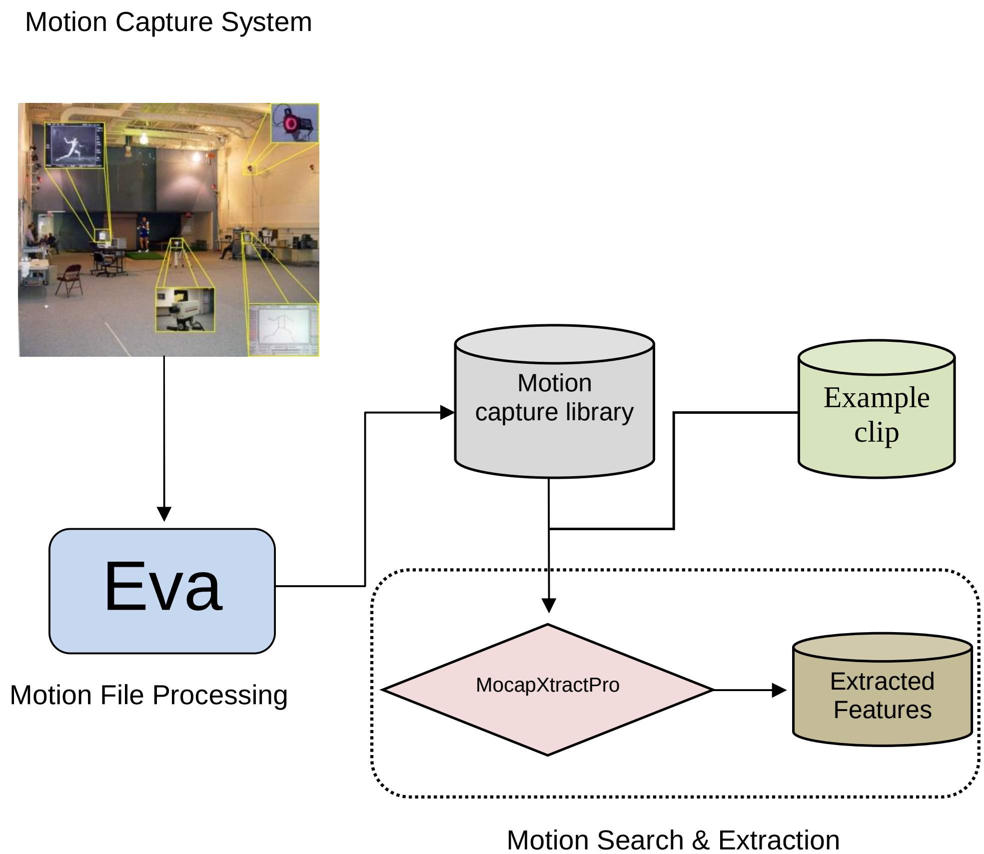


Figure 3.1 Biomechanics Archiving System Flow

3.2 MocapXtractPro System Flow

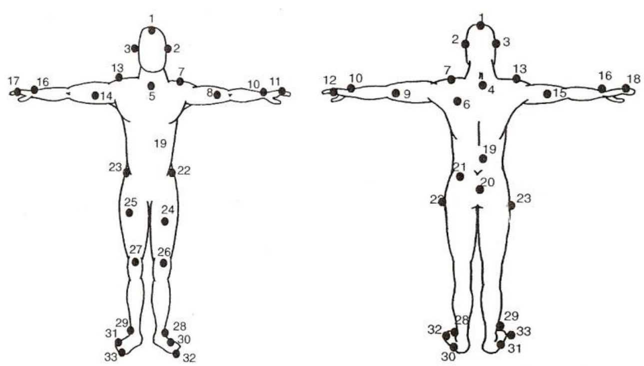
In this section, the proposed MocapXtractPro system for *search* and *retrieving* is explained. The flow starts with the *motion capture* process where the desired motion is recorded using optical based *motion capture* system. After the *motion capture* session is done, the video data is then converted into TRC *motion capture* files and stored into a *motion capture* library for archiving purpose. The

MocapXtractPro *search* and *retrieve* process begins when the user makes a query using an example motion clip to find *matching* files stored in the *motion capture* library. The system will *extract* the features based on the user's selection from the example clip and runs the algorithm to determine the center of motion for the specified clip. Next the system extracts the same features from all of the motion clips in the *motion capture* library and calculates the center of motion for each clip. The system then calculates the distance between the example clip and all the motion capture files in the motion capture library and finally extracts the matching clips according to the threshold value specified by the user. The details of the search and extraction process are discussed in **section 3.6**.

3.3 Sample Collection

To develop and to test the MocapXtractPro system, samples of *"silat olahraga"* motion is needed to build a simulated *motion capture* library or database. Every *motion capture* system has its' own set of rules for standard marker position to be placed on the performer. Universiti Putra Malaysia uses the Motion Analysis Corporation *motion capture* system and the standard marker placement is shown in **Figure 3.2**.

	Head and Neck		Shoulders and Sternum
1	Head_top	5	Sternum
2	LHead	6	LBackShoulder
3	RHead	7	LShoulder
4	Neck_rear	13	RShoulder
		19	MidBack
	Arms and Hands		**Pelvis and Hips**
8	Lbicep	20	Root
9	LElbow	21	LPelvis
10	LWrist	22	LHips
11	LThumb	23	RHips
12	LPinky		**Legs**
14	RBicep	24	LThigh
15	RElbow	25	RThigh
16	RWrist	26	LKnee
17	RThumb	27	RKnee
18	RPinky		**Feet**
		28	LAnkle
		29	Rankle
		30	LMidFoot
		31	RMidFoot
		32	LToe
		33	RToe



Front Back

Figure 3.2 Standard Marker Placements for Motion Samples

A collection of simple *"silat"* motion was collected at the *Pusat Pembangunan Maklumat dan Komunikasi**,** Universiti Putra Malaysia (UPM)* using *Motion Analysis Corporation* © *motion capture* system and the sample collection process is shown on **Figure 3.3**.

Before we can proceed with the *motion capture* session, the actors are fitted with a special suite. A total of 33 pieces of special spherical markers with reflective material are placed on strategic parts of the figure, see **Figure 3.3 (a)**. Next, the system is calibrated to determine the capture volume of the real world in relation with virtual space. This is done by placing a special cube and wand that is also placed with reflective markers. The marker placement of the cube and wand is recognized by the system which enables it to triangulate every camera's position in relation with the markers placed on the actors. The capture volume space usually depends on the number of camera used and for this particular system only allows a maximum of two actors to be recorded at one time **Figure 3.3 (b)**. The *motion capture* session starts with the actor performing the T-Pose action which is also part of the standard calibration procedure. Next, the actual *motion capture* process commences with the actors performing basic "silat olahraga" motions. The movements were done by two actors, each performing four sets of simple "silat" motion i.e. right hand punch, elbowing maneuver, right leg kick and knee attack **Figure 3.3 (c)**. With a special light mounted on the cameras, the reflective markers are made brighter than anything in the background, creating a simple black and white image with no grey scale. The actual video data is then saved with no trace of the human figure. After that the motion capture software (Eva 2.0) will then process the video data and finally produces a computer usable data known as *mocap* file **Figure 3.3 (d)**.

The motion samples generated as TRC files are then manually segmented with each sequence comprising 45 frames of motion and are then stored into a folder or a *motion capture* library. For this experiment, the samples are divided into two separate folders. The first folder called "Master TRC" contains four different motion segments. These files will be used as a *query sample* to find matching files in the second folder called “TRC”. The second folder consists of 16 (sixteen) sample motions which simulates the motion library where all the captured clips are stored. The motion clips are named after the behavior (type of motion recorded) for example RightHandPunc.trc, RightFootKick.trc, etc.

a) Actors fitted with special suit

b) System Calibration process

c) Motion Recording/Capture Process

d) TRC File Output

Figure 3.3 Sample Collection Process

3.4 Center of Motion

The crucial point in *content-based motion retrieval* is the notion of "similarity" used to compare different motions. According to Kovar and Gleicher (2004) intuitively, two motions are regarded as similar if they represent variations of the same actions. A body posed is specified by a set of marker positions, each with three spatial coordinates (*x, y, and z*). In this case, the set is composed of 33 markers representing each pose. A motion sequence is a time series of such pose. In the first stage, a Principle Marker needs to be identified. A Principle Marker is selected based on the best marker to explain the variability seen in a database. For example in a motion segment that contains right punching sequences, the right wrist marker is selected as the Principle Marker. The MocapXtractPro provides six options for the user to declare as the Principle Marker labeled as left wrist, left elbow, left toe, right wrist, right elbow and right toe.

Each motion sequence is represented as a sequence of frames, with 30 frames per second. Each frame is represented by the translation value of the three spatial coordinates (*x, y and z*). For a simple motion, the frames form a cluster that is spread around the center of motion ***P***. For every ***P*** there are 3 dimensions representing coordinates *x*, *y* and *z* presented as follows, $\boldsymbol{P}_c = \boldsymbol{P}_x$, $\boldsymbol{P}_y$ and $\boldsymbol{P}_z$. The *center of motion* ***P*** can be defined as (Feng Liu et al., 2003):

$$\boldsymbol{P} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{P}_i$$

Each frame $\boldsymbol{P}_i$ (i = 1, 2, n) where n is the total of frame in a motion segment. The center of motion ***P*** is determined by calculating the total value of n, where n is the total number of frames in a specific motion segment ($\boldsymbol{P}_1 + \boldsymbol{P}_2 + ..., + \boldsymbol{P}_n$) divided by total number of n.

Kovar and Gleicher (2004) regarded motion as a continuous function **M**(t) that is regularly sampled into frames **M** (t_i), where each frame is a skeletal pose defined by its joint orientations and the position of the root joint. Their goal was to find other motion segments that represent variations of the same action which they refer as *matches* and then uses them as queries to find more distant matches. Their research was to find identical motions in a database for the purpose of motion blending and synthesis in the field of animation. The database in Kovar and Gleicher (2004) research consists of skeleton format motion capture files. So, they used two criteria to determine numerical similarity between two motion segments:

1. Corresponding frames should have similar skeleton poses.
2. Frame correspondence should be easy to identify. That is, related events in the motions should be recognizable.

In the case of tracker motion capture file format, the problem is still the same where-by as explained earlier that the main point is to seek “similarity”. The process of searching for “similarity” is referred as clustering problem which applies to many different applications, including data mining and knowledge discovery, data compression and vector quantization, and pattern recognition and pattern classification (Tapas *et al*., 2002). According to Tapas *et al.* (2002), one of the popular and widely studied clustering methods for points in Euclidean space is called *k-means clustering*. Given a set P of n data points in real d-dimensional space $\mathcal{R}^d$, an integer k, the problem is to determine a set of k points in $\mathcal{R}^d$, called *centers*, to minimize the mean squared Euclidean distance from each data point to its nearest center.

3.5 Motion Search

Given a query, $\mathbf{M}_q$ which is a segment of motion in the data set, the goal is to find other motion segments that are similar to $\mathbf{M}_q$ meaning, segments that represent variations of the same action. This is called *matches*. The search criteria are to find "close" matches that are numerically similar to the query. A user executes the search by providing a *query* and a distance *threshold* value that is used to determine whether two motion segments are numerically similar. Using the Euclidean distance model, the system finds a single motion $\mathbf{M}_q$ that are numerically similar to another motion $\mathbf{M}_i$. The system uses, the *center of motion* of $\mathbf{M}_q(x_q, y_q, z_q)$ and $\mathbf{M}_i$ (x_i, y_i, z_i) to calculate the distance between the two motions defined as follows (Kovar and Gleicher, 2004):

$$d = | \sqrt{(M_q - M_i)^2} |$$

The user specifies the threshold t value of the maximum distance between $\mathbf{M}_q$ and $\mathbf{M}_i$ as the criteria for his search. According to the research done by Kovar and Gleicher (2004) an analysis of the time alignment determines whether two motion segments satisfy the numerical similarity criteria. This research uses a different approach since the structure of a TRC or tracker motion capture file formats are different from a skeleton motion capture file format which has been used.

A skeleton file format uses a hierarchy file structure where every child joints are dependant of its parent joint and ultimately all joints within a skeleton structure are connected to only one root joint. The skeleton file format contains translational as well as rotational information where-by the tracker file format only holds the translational data of each individual marker. Since each marker in a tracker file format are independent, any individual or set of markers can be extracted without affecting the whole file structure. In **Figure 3.6** a visual sample of the original and extracted file of a sample *mocap* data is shown viewed from the *Alias Motion Builder Pro* software, where an experiment is done to test the file structure of the TRC (tracker format) motion capture file while **Figure 3.7** shows the result of the file extraction experiment.

3.5.1 Feature Extraction

This research uses the Principle Marker Selection approach as a basis to form a similarity analysis between two motion segments. Since the nature of a TRC file format is different from a skeleton motion capture file format, any individual marker can be identified as the feature or Principle Marker. The criteria for a Principle Marker selection is explained on **section 3.8** and the main point in this research is that the motion capture data in this research are meant for sport analysis. This is a significant difference between the research done by Kovar and Gleicher (2004) which focuses on animation and motion synthesis. Their research involves the whole skeleton structure and the motions are much more complex compared to that of a motion analysis. A motion analysis for example to study the effect of a training exercise only involves certain part of limb and on a simple motion like a golf or tennis swing. According to Conley *et al.* (2000) studies has been done to investigate injuries in related to sports on specific part of the body such as the elbow, knee, foot and ankle.

The method to compute the distance between $\mathbf{M}_q$ and $\mathbf{M}_i$ has been discussed on **section 3.5** where-by the system calculates the *center of motion* for the sample motion and all the motion capture files in the database. Then the system performs an analysis to calculate the distance between the sample motion and all the other motions. The system will match the *query* by the user based on the *threshold* value and extracts the features of all the *matching* files into a temporary database. The features extracted are actually the translational value of the matching files. This is based on the Principle Markers declared by the user. This means that at each successful extraction process, the temporary database will contain a number of translational values of the matching motion capture files and each file will consists of the three spatial coordinates (*x, y and z*) per frame. The total number of frames for this experiment is limited to 45 frames.

3.6 The Extraction Process Flow

This section will explain the proposed biomechanics data extraction system flow based on the flow chart in **Figure 3.4**. The process starts when the user initiates the query process by providing a sample file ($\mathbf{M}_q$) and a *threshold* (t) value. The system will start by computing the *center of motion* of the sample file and all the biomechanics data in the database. Using the Euclidean distance model, the next step is to calculate the distance between the sample file and all the biomechanics data in the motion capture file library. Next, the system will begin another analysis process which is to compare all the distance values of the *motion capture* files with the *threshold* value provided by the user. The system will identify the *mocap* files based on the $d \leq t$ criteria and extracts the files with the value of less than or equal to the *threshold* value provided by the user save it to a temporary database.

In order to get the optimum value of t, the user needs to run a series of random values. The best range of threshold value will depend on the type of motion involved and also how close the resemblance between motions stored in the motion library and the example motions are. The process ends after all the matching files in the database have been extracted.

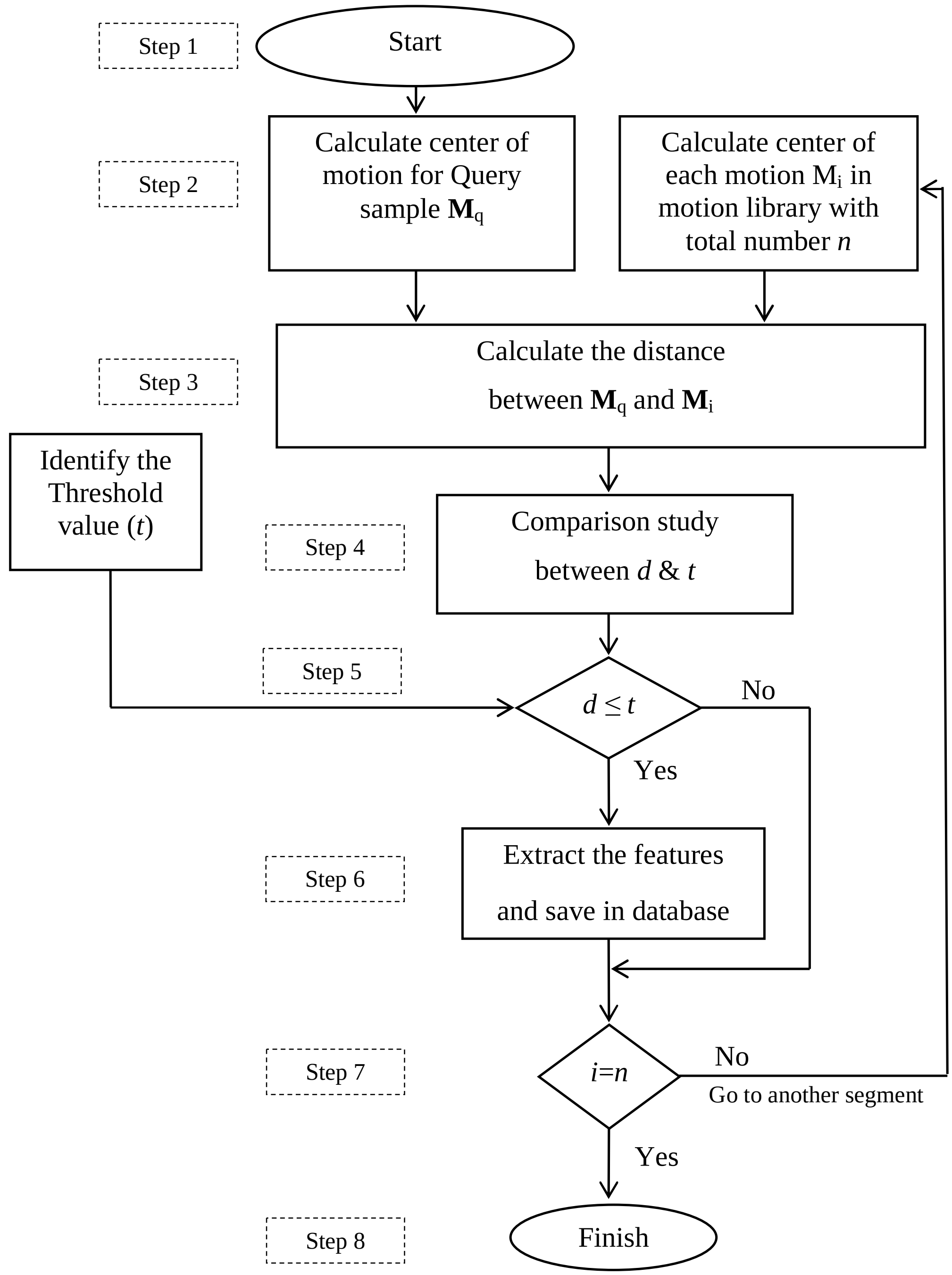


Figure 3.4 MocapXtractPro Search and Retrieval Process Flow

3.6.1 Further Extraction

By now a number of matching motion segments are found and displayed to the user. According to Kovar and Gleicher (2004) method for searching for match sequence, each match sequence is a time alignment between the query and the potential match in $\mathbf{M}_i$. In some cases, the matching motion segments returned may not be in the same set of motion segment submitted by the query. For example in this study, when a query is made using the punching example, the elbowing motion segments are also returned as the matching sequence and can be considered as redundant. This happens because the range of motion between punching and elbowing action are very close even though the *principle marker* selection is different. One way solve this problem is to lower the *threshold* value of the distance between the sample motion and the matching sequences. Another way is by refining the search criteria by the closest match. The flow for this refinement in **Figure 3.5** is basically the same as the normal search criteria except in **step 4** where only the closest match is returned.

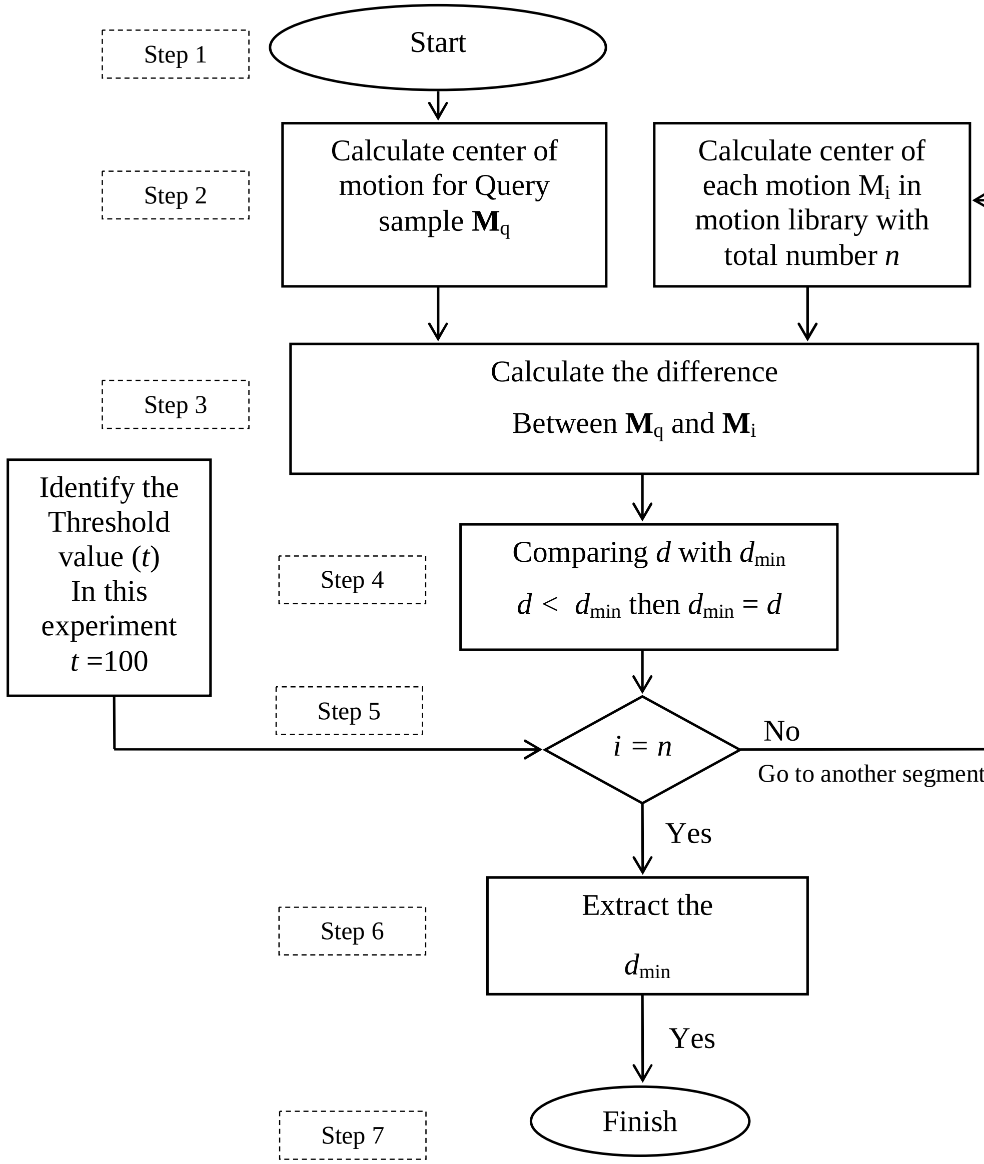
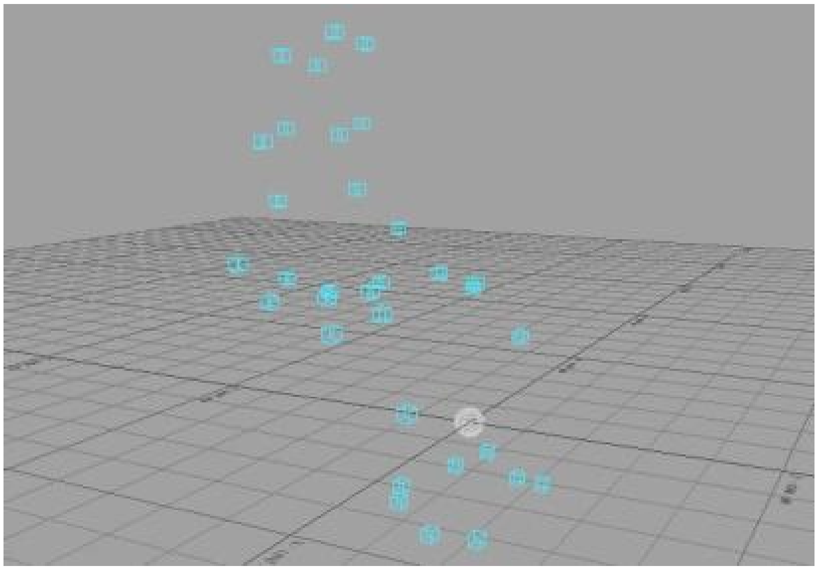


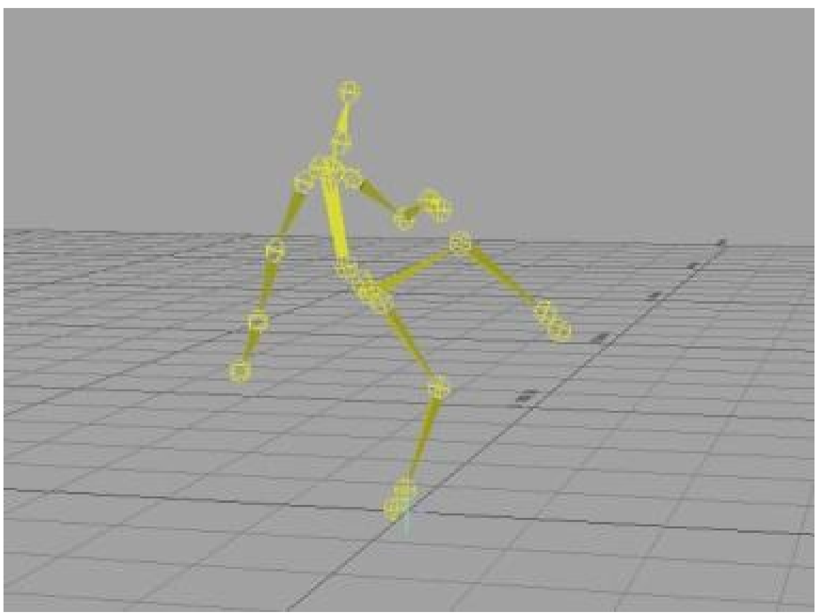
Figure 3.5 Finding Nearest Point to the Sample

3.7 TRC Data Format

This section will discuss on the TRC data format which will be used as the motion capture data format for MocapXtractPro. As mentioned earlier, TRC file format consists of three main components defined as file header, data header and positional data. MocapXtractPro extracts only the key parts of the motion data. TRC file is in the Tracker Format category which only contains the positional or translational information for a set of markers. In this case the total number of markers in the samples collected is 33. **Figure 3.6** shows the graphical representation of the two type of motion capture file format viewed from Alias Motion Builder software where **A** is the Tracker Format while **B** is the Skeleton Format. The Skeleton Format is hierarchal based where the movement of a root joint affects all the child joints in the hierarchy.



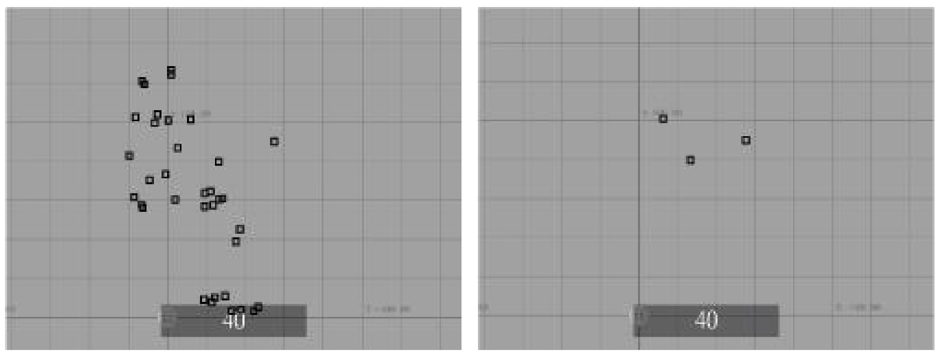
A



B

Figure 3.6 Graphical Differences between Tracker Format and the Skeleton Format

The structure of a Tracker motion capture file format is much less complex compared to the Skeleton Format. Each individual marker is independent of each other and only stores positional data without any rotational values. To prove this theory, an experiment was done by extracting a number of markers from a sample of a Tracker Format motion capture file. The result from this experiment is shown in **Figure 3.7**. The figure shows the before and after result of the extraction process. Viewed from the Alias Motion Builder software, the motion capture file is still readable even with only three markers attached.



Before After

Figure 3.7 Numbers of Markers Before and After Extraction

Table 3.1 shows the difference in terms of the numbers of markers and file size of a TRC file before and after an extraction process. The extraction was done using an earlier version of the MocapXtracPro program. As a result of the extraction process, a set of selected markers was successfully extracted thus reducing the file size of the mocap files and maintains the number of frames for the particular motion.

Table 3.1 Result after the Extraction Process

	Original File	Extracted File
File Size	240kb	36kb
Num. of Markers	33	3
Num. of Frames	241	241

3.8 Principle Marker Selection

As discussed earlier, the *principle marker* selection by the user is based on the main criteria of a specified motion. The MocapXtractPro system needs to understand and identify the position of the *principle marker* inside a TRC file. The TRC format places its content using tab delaminated style where pieces of information is separated by space. Even though the TRC file can be viewed using the Notepad program, it is difficult to identify the position of the *principle markers* this way. Instead MocapXtractPro uses the ASCII code where each character within the content of the file is defined clearly. **Table 3.2** shows the value of all six *principle marker* options included in MocapXtractPro. The system will use the LHand or RHand marker features for distance calculations when the Wrist option is selected, the LOuterElbow and LOuterElbow for Elbow option while LToe and RToe will be read for the Toe option.

Table 3.2 Principle Marker Values in MocapXtractPro

	Left	Right
Wrist	LHand	RHand
Elbow	LOuterElbow	ROuterElbow
Toe	LToe	RToe

Figure 3.8 shows the content for a particular motion data in the TRC file which it uses for calculating the center of motion and also determine the distance value. The figure shows a screen shot of a TRC file viewed from Microsoft Excel. The content within the red dotted box are the actual features that will be extracted by the MocapXtractPro. The explanation on the features has been discussed in **section 3.5.1**.

RShoulder			ROuterElbow			RWristThumb		
X13	Y13	Z13	X14	Y14	Z14	X15	Y15	Z15
-231.03	1406.562	1078.897	-245.976	1099.623	957.4944	-280.18	851.4525	1012.842
-231.488	1405.051	1078.051	-244.032	1098.576	955.9617	-277.183	850.1851	1012.28
-231.042	1404.716	1076.69	-242.984	1098.024	953.7708	-278.837	850.0246	1010.96
-230.786	1404.767	1074.813	-243.709	1097.557	952.5363	-280.341	849.5171	1009.069
-230.909	1404.737	1073.292	-243.456	1096.785	951.2424	-282.671	849.6297	1006.97
-230.605	1404.7	1071.49	-242.188	1097.075	948.7084	-284.739	849.902	1005.448
-231.435	1403.164	1070.261	-240.478	1097.939	943.9708	-288.155	851.9878	1004.226
-232.479	1402.088	1070.274	-236.361	1099.949	934.2545	-294.175	858.0735	1005.398
-233.876	1401.29	1070.157	-231.681	1103.978	922.0927	-301.92	869.0364	1008.129
-233.458	1400.963	1068.424	-227.356	1108.265	910.8951	-310.792	884.1053	1012.172
-232.606	1398.922	1065.993	-223.215	1110.316	902.5789	-317.65	897.7108	1016.628
-231.539	1393.701	1062.409	-218.651	1106.817	896.5035	-325.184	907.1672	1020.878
-232.654	1383.287	1058.956	-211.346	1097.053	891.3649	-331.845	912.2952	1023.657
-233.924	1365.164	1057.053	-200.95	1081.057	885.6587	-337.884	915.7652	1022.977
-232.517	1340.857	1056.722	-187.357	1062.306	881.0885	-345.972	921.7078	1017.915
-231.699	1318.557	1066.059	-184.434	1049.763	877.3214	-360.206	942.8767	1013.134
-228.352	1308.358	1085.58	-210.157	1053.23	880.9996	-386.187	980.7575	1027.155
-215.358	1309.84	1113.866	-276.972	1069.171	905.5618	-422.324	1043.3	1088.98
-191.797	1321.939	1145.257	-366.596	1105.275	981.3774	-432.121	1128.912	1203.948
-164.192	1330.818	1172.95	-436.662	1169.775	1122.92	-379.065	1213.296	1345.81
-141.631	1332.278	1192.433	-423.396	1270.74	1295.429	-261.88	1281.422	1469.605
-117.19	1333.482	1203.326	-328.94	1348.468	1402.415	-114.853	1318.94	1531.631
-97.3225	1334.543	1210.553	-263.073	1354.419	1447.982	-32.33	1329.824	1545.082
-91.3045	1337.212	1223.149	-244.321	1327.93	1473.621	-3.40727	1330.633	1546.792
-86.9576	1338.477	1231.283	-243.505	1307.78	1481.154	-5.19023	1330.885	1551.562
-88.2195	1340.314	1233.523	-250.391	1306.036	1479.091	-16.9465	1330.076	1558.867
-91.0642	1343.396	1234.466	-260.607	1310.557	1474.978	-30.1284	1331.799	1562.513
-97.3978	1347.096	1231.819	-276.365	1315.033	1466.986	-46.1094	1333.953	1562.107
-108.129	1351.522	1227.488	-296.655	1314.224	1454.014	-73.4508	1337.501	1558.574
-118.877	1356.521	1222.484	-321.526	1305.547	1437.359	-107.533	1338.304	1552.999
-130.186	1362.292	1215.8	-348.207	1289.343	1414.652	-143.34	1332.11	1546.657
-143.234	1367.802	1208.196	-374.364	1266.803	1385.332	-183.904	1321.94	1531.669
-157.614	1373.856	1198.748	-396.415	1235.949	1347.594	-225.138	1306.201	1508.677
-172.812	1379.923	1186.908	-408.643	1199.571	1299.79	-264.803	1283.102	1477.961
-187.855	1383.795	1172.447	-410.549	1161.215	1242.783	-299.206	1253.638	1438.569
-203.315	1386.922	1157.012	-398.827	1128.135	1180.157	-327.516	1220.549	1392.226
-215.965	1392.401	1137.902	-372.87	1102.425	1113.866	-345.608	1183.113	1340.188
-227.618	1395.896	1120.527	-338.41	1089.391	1050.416	-352.56	1144.731	1285.759
-235.351	1398.3	1103.58	-297.761	1087.936	993.256	-349.486	1108.033	1230.935
-244.279	1399.605	1091.421	-259.099	1096.142	948.5204	-340.823	1076.943	1178.161
-251.043	1397.886	1080.242	-227.432	1107.197	916.5547	-327.088	1048.83	1133.018
-255.115	1395.997	1068.326	-203.235	1116.629	895.5328	-322.854	1033.766	1090.09
-257.249	1393.448	1058.301	-187.016	1124.085	879.1557	-308.195	1019.38	1065.899
-259.257	1390.391	1050.442	-176.028	1130.646	865.8133	-298.814	1013.309	1044.81
-261.643	1388.293	1045.121	-171.493	1135.245	856.4137	-292.602	1010.564	1027.156

Sample of content extracted by MocapXtractPro

Figure 3.8 Sample of Content Used by MocapXtractPro for its Calculation

3.9 Summary

Throughout this chapter, the methodology of this research is discussed in detail. The *content-based retrieval* system of this study is based on the method introduced by Kovar and Gleicher (2004). In term of knowledge contribution, this study has applied the *content-based retrieval* system to the sport science domain. Kovar and Gleicher (2004) made studies to solve the 3D animators problem in reusing existing motion capture clips for creating new motions of motion synthesis. This research was done to improve the analysis process to aid in coaching and improving “silat olahraga” athletes in particular or virtually any form of sport in general.

The system flow for MocapXtractPro *search* and *retrieving* system has been laid out in detail so that further discussion can be made to improve the outcome of this research. The basic design of MocapXtractPro prototype discussion in this chapter also helps to make the implementation process which will be discussed in the next chapter. The implementation of MocapXtractPro will significantly reduce the *searching* and *retrieving* time of *motion capture* files thus improving the efficiency of a biomechanics archiving system.

CHAPTER 4

IMPLEMENTATION AND RESULTS

4.1 Development of Prototype

The MocapXtractPro prototype is developed in order to test the efficiency of the *search* and *extraction* of *motion capture* data based on the proposed method of *content-base retrieval*. This is done also to meet the third objective of this research which is to test in order to measure the effectiveness of the developed system in term of the accuracy of the file extraction and its' speed. A user friendly interface is designed to get the user easy access to the motion capture library at hand.

4.2 User Interface

A user friendly User Interface is designed for the users to easily make motion capture queries. It enables the user to browse and select the sample motion segment for the queries and displays the result for the motion match. The **MocapXtractPro** and the function of the relevant buttons within the UI are shown in **Appendix C**.

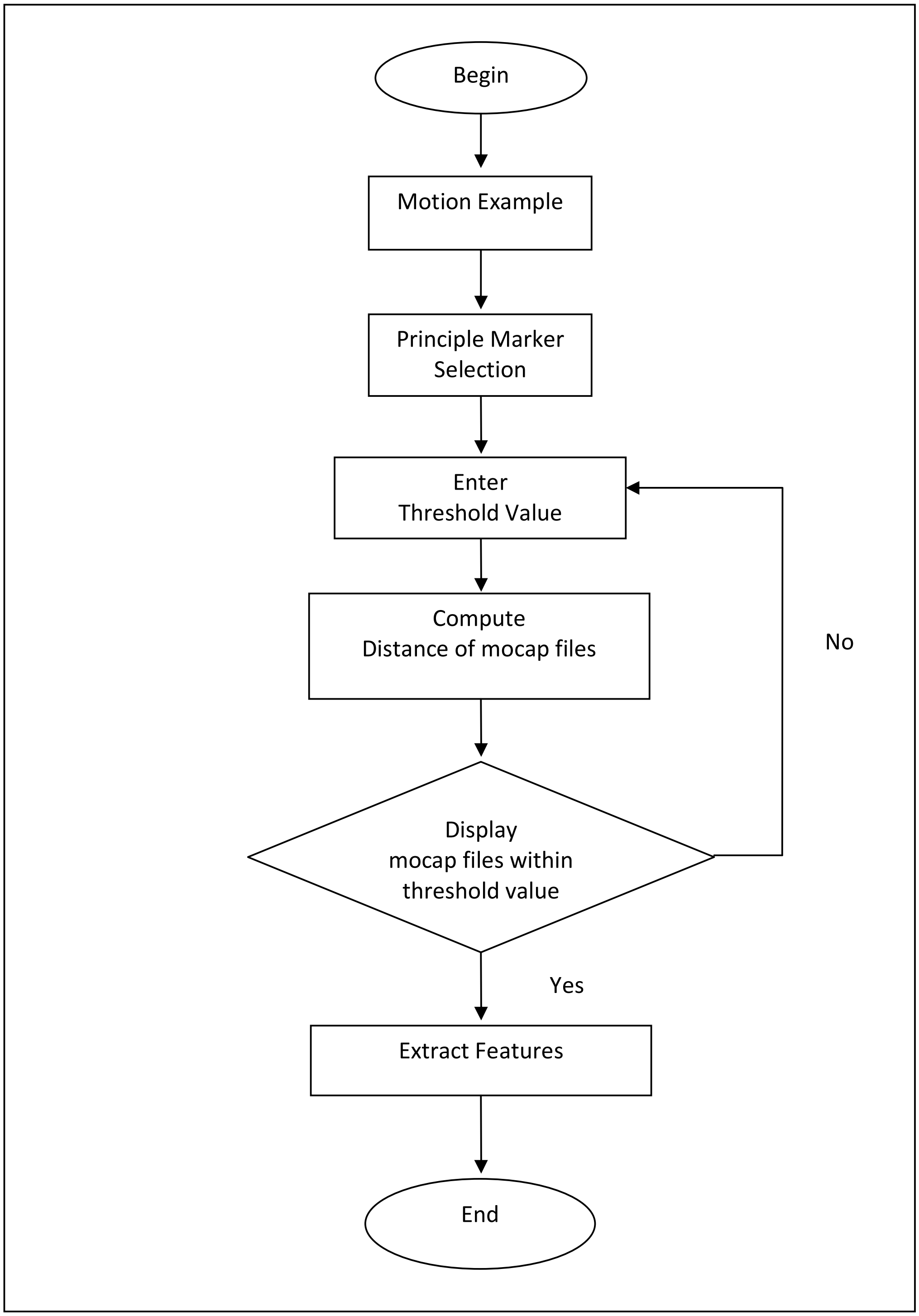


Figure 4.1 Biomechanics Data Extraction Algorithm

4.3 Implementation of the Biomechanics Data Extraction

Figure 4.1 above is the algorithm for the MocapXtractPro program and the explanations on it are as follows:

4.3.1 Motion Example

The function of *content-based retrieval* is to extract information of files based on its content. For this particular system, the extraction of files is made based on the query where an example *motion capture* file is used as a reference to find other matching motion sequence in the database.

4.3.2 Principle Marker Selection

The user needs to specify the principle marker based on the best marker to explain the variability seen in a database.

4.3.3 Enter Threshold Value

The user enters a threshold value to locate similar motion files. This works the same as a normal search function where the user can change the search criteria to get a more refined result.

4.3.4 Compute Distance of Mocap Files

The main task of the operation is to do the *center of motion* calculation and compare the distance value of all the motion segments in the database with the sample motion provided by the user.

Formula: $\boldsymbol{P} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{P}_i$

Where,

$\boldsymbol{P}$ - center of motion

n - number of frames

For example please refer to **Table 4.1**:

Table 4.1 A Simple Motion Segment Example

Motion segment 1					
Frame No.	1	2	3	4	5
Coordinate *x*	0	1	3	5	6
Coordinate *y*	1	3	3.5	4	4.5
Coordinate *z*	2	2	2.5	3	3.5

The center of motion for **motion segment 1** is

Coordinate $x = \frac{0+1+3+5+6}{5} = 15$

Coordinate $y = \frac{1+3+3.5+4+4.5}{5} = 16$

Coordinate $z = \frac{2+2+2.5+3+3.5}{5} = 13$

Center of motion for **motion segment 1** = (15, 16, 13)

The distance between all the motion files in the database with the sample motion file is calculated using Euclidean Distance model. Euclidean distance or simply 'distance' examines the root of square differences between coordinates of a pair of objects.

Formula: $d = | \sqrt{(M_q - M_i)^2} |$

Where,

M_q - input for motion sample

M_i - input for motion segment in database

For example please refer to **Table 4.2**:

Master Kick is the sample motion file and

Right Foot Kick is the motion segment in the motion capture database

Table 4.2 Center of Motion Example

Center of Motion			
	X	*y*	*z*
Master Kick	275	1162	1099
RightFootKick	291	1231	1037

$d = \sqrt{(275-291)^2+(1162-1231)^2+(1099-1037)^2}$

$= \sqrt{256 + 4761 + 3844}$

$= 94.13288$

4.3.5 Display Mocap Files Within Threshold Value

Once all the distance measurement between the sample motion and all the motion segments in the database has been computed, the system will display the motion files that matches ($d \leq t$) with the threshold value entered by the user. The user can choose to enter a new threshold value and run the distance calculation again or go to the final step.

4.3.6 Extract Features

If and when the user is satisfied with the outcome of the motion capture file search result, he can execute the final step where the system extracts the feature (coordinates of the principle marker) into a temporary database.

4.4 Implementation Summary

This chapter discussed the operational framework for development of biomechanics data *search and extraction* task. It explains the use of *content-based retrieval* method introduced by Kovar and Gleicher (2004). Also mentioned in this chapter is the use of principle marker selection which enables the user to determine the type of behavior of a particular motion to be included as another criterion in his search. The process of motion file *search* and *extraction* is also discussed in detail from the algorithm developed from this research. The implementation of the **MocapXtractPro** prototype is completed according to the system design and methodology discussed in **Chapter 3** using visual basic programming. Alas matching motion segments in TRC file format can now be extracted from the motion library. The next step now is to test the prototype for its efficiency of *searching* and *extracting* motion capture files.

4.5 Experimental testing

After the implementation of the algorithm is successfully implemented, a series of test is conducted on MocapXtractPro prototype. The test is done to determine the efficiency and accuracy of the system in retrieving matching *motion capture* files. This test is done also to prove that the method used in solving the research problem is sufficient. In all of the experiments for this study were done with the threshold ***t*** value of 100.

4.6 Preparation for Data Testing

The *motion capture* files used for this experiment were manually segmented into 16 (sixteen) individual files consisting of 45 frames of motion sequence each sampled at 30 frames per second. The motion consists of simple “silat olahraga” moves divided into four categories which includes punching, elbowing maneuver, kicking and knee attack. Each category has four different files representing four different performers. All experiments were run on a machine with 0.97GB of RAM and 3.20 GHz processor. The reasons why all the samples in the database are segmented into only 45 frames are:

i) Because all the movement recorded consists of fast and direct motions, and 45 frames are observed as the average number of frames recorded for all motions in this experiment.

ii) The main focus for this study is to evaluate the performance of the NC algorithm as a simple method to identify similar motions in a motion capture library.

From the experiments done, clusters of center of motion for all the motion clips form in respect to their principle marker selection. **Figure 4.2** shows the motion center clusters for punching and kicking action with respect of their principle marker selections. The master files or motion example is indicated by the red dots while the matching clips are indicated by the blue dots. **Figure 4.2** is a screen shot from a 3D animation software Alias Maya. The clusters shown are within the threshold ***t*** value set at 100.

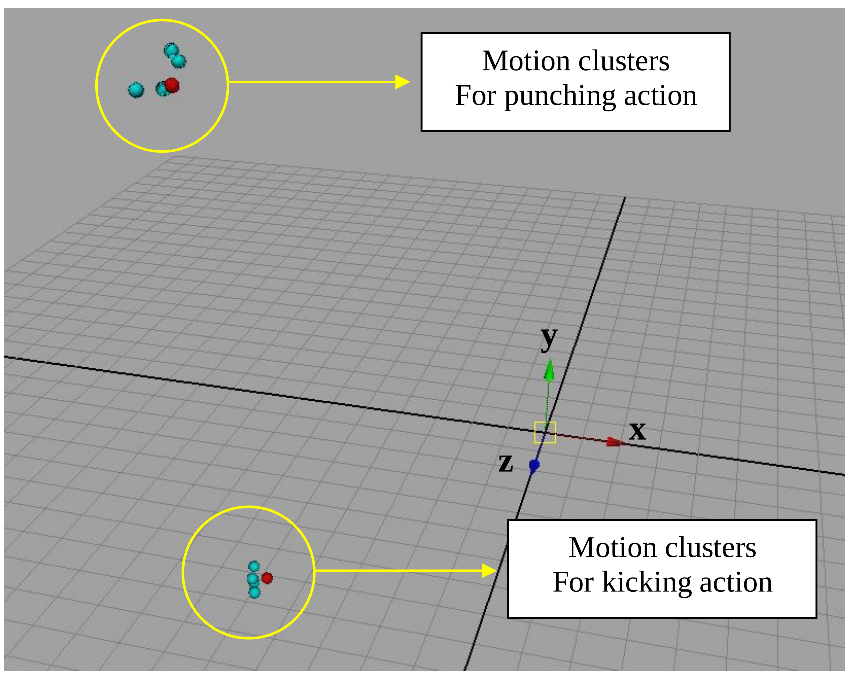


Figure 4.2 Clusters of center of motion in 3D space

4.7 The Results of Experimental Data Extraction

Table 4.3 shows the result of data extraction using the punching motion example. The extraction time is 0.25s and the system extracted 8 similar motion files. The analysis of the experiment is shown in **Figure 4.3** and **Table 4.4.** It shows the distance of extracted motion match against the example motion when maximum distance was set at 100. From this result, it clearly shows that the range of motion between punching motion and elbowing maneuver are extremely close or similar.

Based on observation, the reason for this situation is due to the selection of the *principle markers*. The placements of both the wrist and elbow markers are very close together and the nature of motion affects the result of the distance calculation. The same applies to kicking and knee attack motions. This is consistent with most human motions for example it is not easy to identify between running motions with walking motions based only on the markers coordinates. If the user decides to lower the threshold value, the result will eventually return to only of the punching motion. In **Chapter 3**, this research also suggested an enhancement of the search criteria into extracting the closest match. If the user were to do this, the result would definitely return RightHandPunch_1.trc (punching sample 1) since it is the closest to the model/example file. This might be useful if the user wants to find the same class of motion done by the same actor for instance.

Table 4.3 File Extraction result

	Number of file extracted	Number of Successful match files	Number of failure %
Punch	8	4	0%
Elbow	8	4	0%
Knee	2	2	50%
Kick	4	4	0%

Table 4.4 Motion Extraction Figures

	sample 1	sample 2	sample 3	sample 4
Punching	38	64	55	49
Elbowing	59	75	68	59

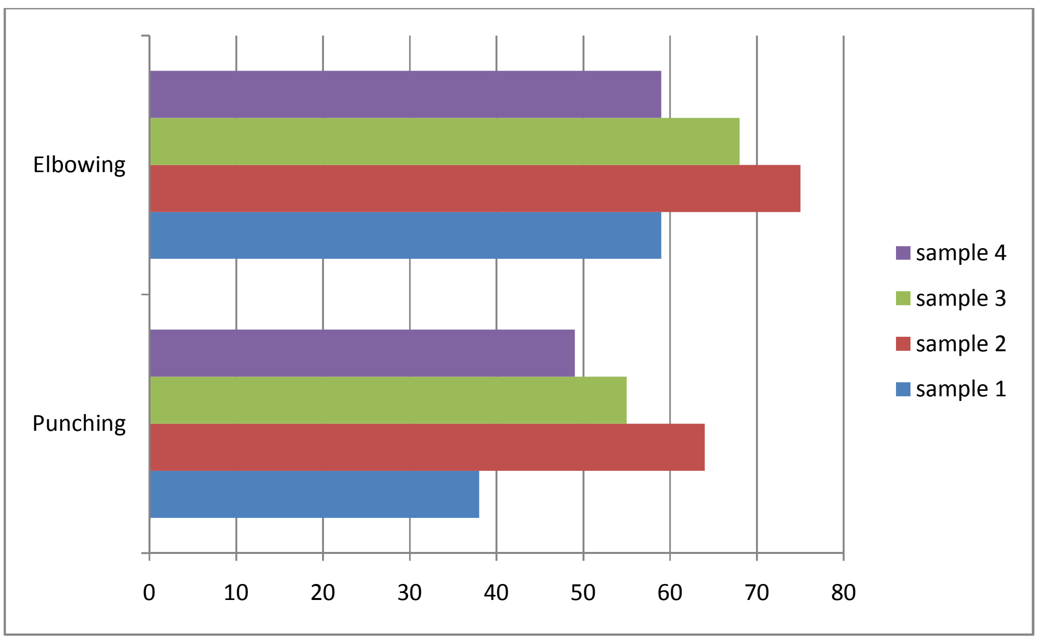


Figure 4.3 Motion Extraction Chart

The time recorded for the system to extract the matching file time are almost interactive. However, the time range of file extraction might increase if the total number of motion segments in the motion library also increases. From the result recorded on **table 4.1** shows that 100% of matching file were successfully extracted for punching, elbow and kicking motions.

4.7 Summary

From the experiment conducted on the MocapXtractPro prototype clearly this system has the potential to significantly reduce the time for searching and extracting biomechanics data in a motion library. This can help improve the efficiency of analysis on the training of “silat olahraga”. Comparison on the athlete performance can be done more efficiently with the *searching* and *extraction* significantly reduced. However from the experiment conducted, several weaknesses on the MocapXtractPro prototype have been identified. The first weakness identified is the application of the NC algorithm itself. In this experiment, all the motion samples were manually segmented into equal number of frames to simplify the data management and observations but in an actual motion library, the parameters of the motions might not be so uniformed. The motion data might consists of many more categories of performers such as in term of genders, height, weight, age etc. This would definitely yield in different result when the NC algorithm is applied to find matching files.

CHAPTER 5

CONCLUSION

5.1 Introduction

This research has successfully developed an automated *search* and *retrieval* system of *motion capture data*. The prototype of this system is called MocapXtractPro. With the development of this system, the *search* and *retrieval* process which are usually done by hand sequencing or by manually browsing through the motion library to find matching files is now automated. The system uses the *content-based retrieval* and *query by example* approach introduced by Kovar and Gleicher (2004). This research has applied the NC algorithm as a method to find a close match to the query example provided by the user. Finding a matching motion clips can be made faster for analysis purposes in sport science. In fact it can also be applied to other application such as animation. This helps the monitoring of an athletes' progress after a particular training session to improve their performance.

5.2 A Summary of Work

In this particular research, the following have been done:

i. *Content-based retrieval* has been applied in searching and retrieval of biomechanics data in the field of sport science.

ii. NC algorithm is applied to find matching motion files

iii. Conduct experiments by using "silat olahraga" motion samples representing a motion capture file library.

iv. Test and evaluate the performance of biomechanics data extraction algorithm.

v. Results analysis and discussion based on the results obtained in the experiment.

5.3 Future Works

From the experiments done and discussed in **Chapter 4**, the performance of this *search* and *retrieval* system for biomechanics data can be improved. This research assessed the NC algorithm to find similar or matching motion capture files in a database. The algorithm an method proposed proved that it is possible to automate a similar motion search. However, due to the fact that motion capture files have no industry wide standard format, a few adjustments has to be done when dealing with any different mocap format. Writing different search and extraction programs for every file format would be futile. Instead, maybe a standard motion capture file format should be introduced. With the introduction of a standard mocap file format, this might make it possible to have a consistent and optimum threshold value. Another possible future research can be done to automate the process of motion segmentation. Manual motion segmentation is time consuming and might not work for every application. This useful since motion databases are growing rapidly because of its' wide range of applications.

BIBLIOGRAPHY

Alexander Refsum Jensenius, Antonio Camurri, Nicolas Castagne, Esteban Maestre, Joseph Malloch, Douglas McGilvray, Diemo Schwartz, Matthew Wright (2007). Panel: The Need of Formats For Streaming And Storing Music-Related Movement And Gesture Data. *International Computer Music Conference (ICMC 2007)*. August 27-31, Copenhagen.

Atha J. (1984). Applied Ergonomics. *Current techniques for measuring motion*.15.4: 245 –257.

Bobby Bodenheimer, Chuck Rose, Seth Rosenthal, John Pella (1997). The Process of Motion Capture: Dealing with the Data. *Computer Animation and Simulation '97, Eurographics Animation Workshop*. Sept. 1997. Springer-Verlag, Wien. 3-18.

Carlos R. Morales (2001). Development of an XML Web based motion capture data warehousing and translation system for collaborative animation projects. *Proceedings of the 9th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*. February 5 - 9, 2001. Plzen, Czech Republic. 168.

Conley R., Tulchin, K., Harris, G., Smith, P., Humm, J., Hassani, S. *Pediatric Sports Medicine: An Evolution of Applications of Motion Analysis*. Pediatric Gait, 2000. A new Millennium in Clinical Care and Motion Analysis Technology. 116-123.

Cerveri P., Pedotti A., Ferrigno G. (2003). *Robust Recovery of Human Motion From Video Using Kalman Filters and Virtual Humans*. Human Movement Science. 22: 377-404

Dyer, S., Martin, J. and Zulauf J (1995). *Motion Capture White Paper*. Technical Report. Silicon Graphics, December 12, 1995.

Feng Liu, Yueting Zhuang, Fei Wu and Yunhe Pan (2003). *3D Motion Retrieval with Motion Index Tree*. Computer Vision and Image Understanding 92, 265-284.

Gregory D. Myer, Kevin R. Ford, Joseph P. Palumbo Timothy E. Hewett (2005). *Neuromuscular Training Improves Performance and Lower-Extremity Biomechanics in Female Athletes.* Journal of Strength and Conditioning Research, 2005, 19(1), 51–60.

Guodong Liu, Jingdan Zhang, Wei Wang, Leonard McMillan (2005). *A System for Analyzing and Indexing Human-Motion Databases*. SIGMOD 2005 June 14-16, Baltimore, Maryland, USA.

Heather Kenyon (1999). .*ILM's Seth Rosenthal on motion Capture*. ANIMATION WORLD MAGAZINE. February 1999. ISSUE 3.11

Herda L., Fua P., Plänkers R., Boulic R. & Thalmann D. (2000). Skeleton-based motion capture for robust reconstruction of human motion. *Proc. Computer Animation*, , 77-83. Retrieved July 1st, 2008 from www.scopus.com.

Hyun-Sook Chung & Yilbyung Lee (2004). *MCML: motion capture markup language for integration of heterogeneous motion capture data*. Computer Standards & Interfaces, 26. 113-130.

Ina Hunger & Jorg Thiele (2000). *Qualitative Research in Sport Science. Forum*. Qualitative Social Research. 2000. 1-2000.

James F O'Brien, Robert E. Bodenheimer Jr., Gabriel J. Brostow, Jessica K. Hodgins (2000). *Automatic Joint Parameter Estimation from Magnetic Motion Capture Data*. Graphics Interface 2000, Montréal, Québec, Canada: BibTex. 53-60.

Jernej Barbic, Alla Safonova, Jia-Yu Pan, Christos Faloutsos, Jessica K. Hodgins, Namcy S. Pollard (2004). Segmenting *Motion Capture Data into Distinct Bahaviors*. Proceedings of Graphics Interface 2004, May 17-19, London, Ontario.

Jie Yang, Qiang Huang, Zhaoqin Peng, Lige Zhang, You Shi and Xiaojun Zhao. Capturing and Analyzing of Human Motion for Designing Humanoid Motion. *Proceeding of the 2005 IEEE International Conference on Information Acquisition* June 27 – July 3, 2005, Hong Kong and Macau, China: IEEE. 332-337.

John Neter, William Wasserman, G.A Whitmore (1993). *Applied Statistic (4th Edition)*. Boston: Allyn & Bacon.

Klempous, R. (2007). *Movement identification analysis based on motion capture* Retrieved July 21, 2008 from www.scopus.com.

Kovar Lucas and Gleicher Micheal (2003). Flexible Automatic Motion Blending with Registration Curves. *Eurographics/SIGGRAPH Symposium on Computer Animation*. July 26 - 27, 2003. San Diego, California. 214 – 224.

Kovar Lucas and Gleicher Micheal (2004). *Automated Extraction and Parameterization of Motions in Large Data Sets*. ACM SIGGRAPH 2004 Papers. 559 - 568.

Lakany, H. M., Hayes, G. M., Hazlewood, M. E., & Hillman, S. J. (1999). Human walking: Tracking and analysis. *IEE Colloquium (Digest),* (103), 25-38. Retrieved August 12, 2008 from www.scopus.com.

Margaret S. Geroch (2004). *Motion Capture for the Rest of Us.* Journal of Computing Sciences in Colleges. Volume 19, Issue 3: 157 - 164

Maureen Furniss (2000). *Motion Capture: An Overview*. Animation Journal. Spring 2000: 68-82

Müller, M., Röder, T., & Clausen, M. (2005). *Efficient content-based retrieval of motion capture data*. Paper presented at the *, 24*(3) 677-685. Retrieved 25 June, 2005 from www.scopus.com.

Meredith M., & Maddock, S. (2005). *Motion Capture File Formats Explained,* Retrieved July 1st 2008 from www.scopus.com.

Michail Vlachos, Marios Hadjieleftheriou, Dimitrios Gunopulos and Eamonn Keogh (2005). *Indexing Multidimensional Time-Series*. The VLDB Journal, 15(1), 1-20. Springer-Verlag.

Niemann H., (1990). *Pattern Analysis and Understanding* (2nd Edition). Berlin: Springer-Verlag.

Ryszard Klempous, Movement Identification Analysis Based on Motion Capture. In: *Computer Aided Systems Theory – EUROCAST 2007* . Berlin / Heidelberg: Springer. 629-637; 2007.

Scott L. Delp, J. Peter Loan (2000). *A Computational Framework for Simulating and Analyzing Human and Animal Movement*. Computing in Science And Engineering. Vol. 2, No. 5: 46-55.

Shih-Pin Chao, Chih-Yi Chiu, Jui-Hsiang Chao Yil-Cheng Ruan, Shi-Nine Yang (2003). *Motion Retrieval and Synthesis Based on Posture Feature Indexing*. ICCIMA'03. 27-30 Sept. 2003: 266 – 271.

Suddha Basu, Shrinath Shanbhag and Sharat Chandran, (2005). *Search and Transisioning for Motion Captured Sequences*. VRST'05, November 7-9, Monterey, California, USA.

Susan J. Hall (1991). *Basic Biomechanics*. (3rd ed.) Boston: WCB/McGraw-Hill.

Tanco L. M., Hilton A. (2000). Realistic Synthesis of Novel Human Movements from Database of Motion. *Proceedings of the IEEE Workshop on Human Motion HUMO*. 7-8 December, 2000, Austin, Texas.137-142.

Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, Angela Y. Wu. A Local Search Approximation Algorithm for *k*-Means Clustering**.** *Proceedings of The Eighteenth Annual Symposium on Computational Geometry* . June 5-7, 2002. Barcelona, Spain. 10 – 18.

Thomas B. Moeslund and Erik Granum. *A Survey of Computer Vision-Based Human Motion Capture*. Computer Vision and Image Understanding 81 (3). 231-268.

Thomas P. Andriacchi*, Eugene J. Alexander (2000). *Studies of human locomotion: past, present and future*. Journal of Biomechanics. 33: 1217-1224.

Yancey Hall (2006). *Martial Artists' Moves Revealed in "Fight Science" Lab. National Geographic news*. Retrieved: 25, June,2006 from http://news.nationalgeographic.com/news/2006/08/060814-fight-science.html

Yan Gao, Lizhuang Ma, Zhihua Chen, Xiaomao Wu (2005). *Motion Normalization: the Preprocess of motion Data*. VRST'05, November 7-9, Monterey, California, USA, 253 - 256

APPENDIX A

A sample of TRC data

PathFileType 4 (X/Y/Z) Kaydara

DataRate CameraRate NumFrames NumMarkers Units OrigDataRate OrigDataStartFrame OrigNumFrames

30 30 45 33 mm 30 51 45

Frame# Time LFrontHead RFrontHead LBackHead

X1 Y1 Z1 X2 Y2 Z2 X3

1 1.667 -18.69157 1615.22537 1110.86693 -137.03634 1600.35156 1053.84575 35.35489

2 1.7 -13.99416 1616.12 1113.77762 -131.80346 1601.71478 1056.86386 40.15768

3 1.733 -8.38547 1616.4212 1115.6765 -125.3895 1602.36603 1058.14705 46.61876

4 1.767 1.31734 1615.97931 1116.95 -116.41677 1601.06842 1059.24195 55.69662

5	1.8	13.32421	1615.84412	1119.01123	-104.48053	1601.55136	1062.10449	66.26871
6	1.833	27.25081	1616.11618	1121.00914	-90.06212	1600.84778	1064.49211	79.40887
7	1.867	42.66023	1616.7952	1123.09258	-75.16958	1602.47299	1066.549	94.77948
8	1.9	57.45111	1619.7934	1126.41907	-59.85765	1604.0889	1068.97537	110.48658
9	1.933	69.44515	1621.7955	1129.72313	-46.42279	1605.15274	1071.9236	125.62355
10	1.967	78.37519	1624.28284	1135.10582	-37.56127	1604.92218	1077.03003	135.23796
11	2	81.40879	1625.35507	1143.26737	-33.41862	1605.31464	1083.84499	141.29726
12	2.033	81.95178	1624.61884	1155.35866	-32.29599	1603.74329	1094.10332	144.11227
13	2.067	75.67309	1621.71005	1171.01479	-36.30689	1600.0499	1106.83533	142.62744
14	2.1	69.25113	1618.311	1185.5764	-41.51485	1595.93643	1120.92713	138.80916
15	2.133	66.56529	1616.35773	1198.14453	-45.17535	1593.48663	1133.35816	134.64694
16	2.167	63.92297	1617.01248	1209.03557	-47.83008	1592.5705	1145.34172	132.10496
17	2.2	63.30836	1617.01111	1221.36864	-48.65219	1593.1218	1157.73842	131.79669
18	2.233	62.04387	1619.7908	1232.15813	-48.22303	1596.20255	1165.92484	133.59329

19	2.267	63.67223	1626.35284	1232.16553	-46.08067	1601.87439	1163.42774	137.95707
20	2.3	67.43394	1636.60492	1215.66887	-38.27477	1609.78409	1144.67491	148.43502
21	2.333	74.72768	1647.1257	1191.75659	-29.27858	1617.47955	1119.24706	159.16101
22	2.367	84.09935	1653.33771	1163.9486	-19.24614	1622.22595	1089.97216	170.73741
23	2.4	92.09909	1655.13672	1132.60941	-9.24478	1622.89246	1059.33861	180.32698
24	2.433	104.37103	1655.3241	1097.13761	-1.5923	1621.23077	1029.9791	185.63463
25	2.467	115.6548	1652.90253	1061.6732	3.27406	1618.11539	1006.11801	184.82578
26	2.5	121.18055	1647.69363	1033.77892	4.5889	1611.82846	988.48656	178.36764
27	2.533	121.49987	1640.73944	1018.94303	2.09249	1604.48105	980.55748	172.38482
28	2.567	116.58199	1632.80396	1019.06723	-3.75156	1596.7334	983.13477	164.78361
29	2.6	109.99821	1626.9989	1028.95737	-12.14679	1592.11304	992.31766	155.60986
30	2.633	101.52737	1622.89215	1041.22605	-20.76243	1587.39059	1004.75022	146.94549
31	2.667	93.31765	1620.40573	1056.54816	-28.44496	1586.19217	1019.80858	139.84802
32	2.7	88.83439	1619.95651	1072.38808	-33.59232	1587.45941	1035.44304	133.5046

33 2.733 84.38372 1621.97418 1086.89408 -38.41494
1590.65887 1049.34044 128.10482

34 2.767 78.58975 1626.3382 1099.3206 -43.96816
1597.30438 1059.91326 121.80584

35 2.8 70.26427 1632.92648 1107.13906 -51.19615 1603.5965
1066.90674 115.28029

36 2.833 60.97447 1639.90616 1112.99462 -60.04484
1609.46991 1071.92642 106.8844

37 2.867 50.65705 1645.41794 1116.71753 -68.93841
1614.41803 1075.27176 98.614

38 2.9 40.26422 1649.24728 1119.9955 -81.76486
1617.60468 1077.10381 90.23839

39 2.933 28.78468 1650.89005 1122.94411 -91.14006
1619.29733 1077.70836 80.65405

40 2.967 15.00307 1651.67358 1124.9572 -104.48226
1620.23605 1077.2741 70.34505

41 3 -0.94615 1651.02982 1126.8908 -117.74277
1618.82065 1076.23734 59.57738

42 3.033 -18.55454 1648.06671 1129.64714 -133.27274
1615.50644 1075.45754 45.92637

43 3.067 -37.87526 1640.83557 1132.14234 -151.91112
1607.69943 1075.39444 30.03749

44 3.1 -58.41559 1627.24762 1134.96071 -170.96025
1594.61823 1075.44655 11.80494

45 3.133 -58.41559 1627.24762 1134.96071 -170.96025
1594.61823 1075.44655 11.80494

		RBackHead				TopSpine				Sternum
Y3	Z3	X4	Y4	Z4	X5	Y5	Z5	X6	Y6	Z6
1536.89407	1401.54129	969.85962	887.41486	-86.51932	-63.1917	1542.5499	1304.12537	937.20154	1079.71855	-52.04662
1536.26526	1401.2883	973.94692	891.44997	-82.16957	-60.94469	1543.03009	1305.6955	939.88762	1083.92578	-48.37558
1535.38177	1398.62396	976.24069	896.59088	-76.2975	-56.73587	1541.8808	1305.53497	945.19089	1089.47464	-41.76943
1531.92474	1396.13983	979.43337	901.82831	-66.41085	-50.0373	1538.93768	1305.58289	945.04936	1095.82619	-35.34201
1529.46671	1393.17535	981.89553	905.97275	-55.55889	-41.49833	1537.49954	1305.48096	948.19931	1102.7417	-28.14654
1527.9158	1388.56812	985.07614	913.96477	-41.86662	-30.07017	1535.97198	1306.38062	951.24352	1110.25734	-19.04849
1527.5441	1385.65994	988.6261	919.05358	-27.08596	-16.47292	1535.49469	1308.47916	954.3924	1117.83974	-8.00497
1528.32672	1384.82849	992.8949	924.64386	-10.24505	-1.86383	1536.00372	1312.40021	957.74071	1125.54497	5.17725
1529.47403	1384.92981	998.38135	930.63942	2.04618	11.88733	1536.42822	1317.04681	964.61838	1133.51708	18.4091
1531.09573	1383.74039	1004.59313	936.46912	14.10999	24.34586	1536.22238	1319.40109	967.36534	1140.81116	29.44241
1533.89053	1387.04681	1013.27004	944.93637	21.2741	32.21461	1536.89331	1320.5751	975.31281	1148.35099	39.61185

1534.36493 1025.00298 24.09299 1536.19812 985.91431 49.53355
1390.12756 955.04524 32.11749 1320.93277 1156.7749

1536.02524 1040.57762 22.99344 1536.54922 998.81005 56.8986
1394.28329 964.89792 30.79227 1318.25775 1164.59069

1536.24924 1054.46083 18.8204 1534.99725 1011.62491 58.825
1396.06125 976.91399 30.09869 1314.74121 1173.33946

1535.19562 1065.21721 14.77697 1533.40958 1024.13323 57.25089
1397.49786 990.30449 26.8292 1313.06137 1184.37844

1535.09125 1076.91155 10.32013 1532.21863 1037.76787 58.95526
1403.16574 1003.35411 26.72394 1311.00464 1194.90929

1535.6189 1089.17092 11.31135 1533.75565 1047.88178 59.99112
1409.82452 1014.80927 25.60377 1309.80179 1202.74369

1541.41068 1099.02855 13.08341 1539.12247 1057.39022 62.89884
1416.87653 1019.27307 27.87729 1312.60895 1205.40955

1551.72134 1098.44498 19.19075 1547.52228 1053.68004 66.88467
1423.60138 1015.346 34.19296 1321.99478 1202.16789

1562.15378 1085.77957 31.22169 1554.74304 1037.86278 71.79293
1425.40314 1001.69098 43.28965 1335.74432 1194.07532

1572.63962 1064.31542 43.35811 1563.81439 1014.26232 76.40171
1421.71188 979.91692 54.00511 1349.41681 1179.22417

1581.66794 1038.08159 55.24532 1570.07858 985.70091 83.34695
1415.10345 953.2737 64.99287 1362.58926 1158.54286

1583.68332 1005.76531 65.82804 1571.09955 954.4896 86.17217
1402.89581 925.04753 75.80684 1372.98599 1135.59769

1578.49152 969.22127 68.04096 1564.78409 923.66875 81.54541
1389.78043 902.55661 86.06969 1380.67291 1115.63782

1566.87073 932.53227 62.0151 1552.96173 900.57518 72.73205
1374.19037 887.25838 92.87638 1381.3028 1100.02663

1551.96594 904.99756 54.52872 1539.89655 883.92769 62.97975
1360.30548 879.0258 95.76719 1375.03647 1089.92081

1539.26575 893.25951 46.67205 1528.08182 876.4811 53.69233
1349.54544 875.15503 94.76404 1366.49323 1086.12732

1531.62033 892.38449 38.37434 1520.0174 878.13095 45.37765
1342.84286 875.47043 91.71524 1356.76117 1086.79375

1528.73306 898.66791 29.772 1518.26385 885.37857 37.65448
1339.39621 878.23159 87.89251 1346.98364 1089.01795

1526.40244 908.7999 19.9745 1514.02481 896.52886 30.35927
1339.19693 884.1552 82.75514 1336.93878 1093.20976

1527.70126 921.35422 13.53547 1516.06354 909.05457 22.66283
1341.43707 891.05843 77.12403 1327.07199 1097.00585

1531.78802 934.07128 7.33822 1521.66473 922.52251 15.16348
1350.10513 897.36382 72.75993 1323.88123 1102.70569

1536.46576 945.91759 1.38513 1527.12982 934.29177 8.96478
1358.0159 903.45337 66.90974 1324.80804 1107.91771

1543.93631 957.02286 -4.73572 1535.31631 943.73223 2.73059
1368.00018 908.88489 61.06776 1329.60052 1112.02324

1551.30554 964.53003 -10.58936 1543.02505 950.51445 -2.90066
1377.85583 912.80648 56.88686 1332.30682 1114.93263

1559.30496 970.10178 -18.66669 1549.20609 955.37407 -8.62697
1386.18744 916.22734 50.51452 1337.86835 1117.45178

1565.40695 974.20342 -27.32008 1554.93515 958.98194 -14.93106
1393.58032 918.46031 44.73524 1342.2226 1119.29093

1570.22888 977.7916 -36.91975 1558.39828 961.91971 -19.30881
1398.30475 919.58229 36.84184 1344.30084 1120.58327

1572.24411 980.73967 -45.19448 1559.72733 963.53691 -24.07391
1402.25983 919.61594 26.54849 1345.2037 1121.26221

1571.87637 984.79332 -54.3237 1560.70618 964.05823 -27.40994
1404.68307 919.13193 13.14444 1344.6106 1122.11571

1572.04315 989.62967 -65.35679 1560.26077 964.39232 -30.13297
1406.91437 917.81311 -2.19507 1342.24945 1121.25817

1570.04486 992.75658 -77.27835 1558.60855 964.3663 -33.71094
1407.96326 916.24436 -18.92349 1338.49014 1119.24294

1565.60883 996.02112 -93.17943 1552.0929 964.95415 -37.90311
1408.40027 915.0454 -37.81405 1329.27032 1116.46019

1553.51959 999.40941 -108.42887 1540.17365 962.74163 -42.50072
1403.80493 915.00473 -59.4375 1315.05463 1111.65062

1553.51959 999.40941 -108.42887 1540.17365 962.74163 -42.50072
1403.80493 915.00473 -59.4375 1315.05463 1111.65062

Chest LShoulder LOuterElbow
LWristThumb

X7 Y7 Z7 X8 Y8 Z8 X9 Y9 Z9 X10 Y10

-80.18536 1181.97487 1083.41088 98.07378 1387.39548
1029.69361 174.12387 1084.87381 889.98741 120.14959
879.87732

-78.28791 1183.95164 1087.34123 101.61772 1386.75018 1033.5669
176.27798 1082.81616 895.45403 124.72962 882.37564

-75.74628 1183.70804 1093.98743 107.19315 1385.40222
1037.35985 179.54756 1079.83521 901.18332 131.32465
884.76448

-71.99729 1184.51782 1101.99578 113.79063 1383.67737
1041.96511 184.49432 1076.66222 906.79924 140.92652
887.07054

-65.14905 1184.74609 1110.80376 122.76798 1379.9733
1044.37279 192.66609 1072.45224 913.70072 153.06477
889.37881

-56.55629 1186.31454 1119.92615 134.43685 1376.3028
1046.56578 203.83536 1064.44481 921.92429 169.13308
890.40222

-45.58062 1189.18167 1131.1454 148.73217 1371.94763
1046.88561 216.83811 1056.32469 931.22757 186.93682
892.52808

-32.76983 1194.38416 1141.74523 161.84092 1370.88593
1049.49974 232.16494 1052.24831 943.50182 206.06814
898.6367

-18.91053 1199.84436 1150.65918 174.71569 1371.35529 1052.9168
249.74198 1051.17714 961.18645 222.42455 909.10843

-5.65398 1201.52992 1157.70745 185.62229 1371.77841
1057.78191 268.17288 1048.7413 984.11278 233.72862
921.12724

2.64609 1202.96082 1164.44054 195.45134 1374.14383
1069.22478 282.07327 1047.43065 1013.03131 234.39583
932.24983

4.4455 1203.6467 1168.08548 197.6898 1379.03214 1088.16918
286.94147 1051.40343 1047.15111 223.34602 947.70569

3.85041 1201.1235 1173.4224 193.673 1381.83182
1106.87737 288.03892 1055.7106 1081.52886 202.65797
959.43939

2.04301 1198.35938 1176.94329 187.69442 1385.27802
1123.58574 286.55558 1060.51392 1120.04845 176.53103
966.7894

1.82763 1195.58144 1184.86992 183.38724 1387.23053
1139.58962 280.46873 1065.02235 1151.36757 154.35813
972.21046

2.9857 1193.57872 1191.15372 179.28526 1389.09195 1155.45899
272.14581 1067.33727 1174.34265 139.15825 971.7614

3.47241 1193.70743 1196.25366 176.43478 1390.26062
1169.50676 269.59719 1070.07751 1189.7245 132.66783
966.12862

5.16579 1197.00356 1198.36609 177.15083 1393.57269
1174.05594 275.6938 1076.48087 1196.64971 136.78203
953.10875

9.86701 1205.7576 1199.1243 184.49844 1399.66492
1168.50426 285.52752 1084.90051 1192.91382 153.69845
938.14491

18.29366 1219.49203 1200.41924 196.85589 1404.37714
1151.45241 293.79362 1091.8782 1177.77054 183.095
922.43149

29.04718 1234.92081 1197.26761 212.03304 1407.59323
1123.78433 305.05877 1093.53294 1156.02226 221.80374
910.68436

36.42585 1251.15623 1188.41118 227.0804 1406.90964
1090.04883 318.65927 1093.63785 1130.27588 263.51759
906.14716

44.14498 1268.77647 1178.95409 236.73857 1402.86209
1052.59178 328.21669 1092.05383 1105.10735 301.19236
908.09677

54.73166 1280.87494 1169.14025 241.19013 1394.9884
1017.06414 335.36488 1086.9236 1080.8799 334.26706
909.51378

63.46385 1283.73901 1162.39418 240.62431 1384.12933 987.26242
339.57581 1080.0808 1060.90348 358.0727 913.0278

70.08422 1280.77393 1158.80669 237.31632 1372.45056 968.12561
337.5259 1070.00244 1046.14647 370.72552 910.60204

72.88092 1272.57851 1155.52544 232.96028 1365.17838 960.12726
333.45284 1060.74135 1036.59592 372.67693 904.47502

72.03889 1259.2495 1153.4491 226.10584 1358.32977 957.22878
326.50833 1051.20255 1028.82668 365.23193 895.80101

70.53726 1246.36856 1153.50304 220.74412 1354.53094 957.50603
317.1435 1044.81827 1017.9718 352.20745 886.47316

67.73553 1234.04015 1154.14749 212.67607 1350.48004 959.07746
307.88019 1037.09106 1005.28404 334.96893 877.51053

65.49614 1221.44394 1155.08599 207.57221 1348.17673 964.95415
295.70597 1028.7429 992.25998 316.75713 866.90193

60.95737 1215.42221 1156.40175 203.21928 1350.21973 969.22722
278.54141 1025.2562 977.86148 295.85413 859.09233

55.12999 1212.91298 1157.16774 197.64937 1355.74783 975.68596
263.80716 1026.85181 964.35433 275.60244 855.89424

51.49033 1214.02687 1155.63195 191.43362 1365.03464 980.95413
248.16628 1033.71384 952.15744 250.71749 856.85173

46.47152 1218.32962 1155.97389 187.25004 1375.09094 985.55229
234.3125 1043.41187 941.33141 228.85632 860.72586

41.56112 1222.14066 1154.61891 181.25578 1384.67728 989.95087
223.73295 1053.80753 931.97968 209.98419 867.26547

36.65136 1225.40833 1154.33266 175.45778 1391.0379 992.07802
213.96879 1061.59584 923.80593 194.00078 873.9579

29.52712 1227.22267 1153.65639 170.37312 1399.18228 997.01042
206.2644 1067.58133 919.33708 179.4797 881.6951

19.56045 1227.09152 1152.42325 163.0793 1404.62494
1003.51647 197.73665 1077.89444 914.88556 166.34403
889.86038

7.38447 1225.76332 1151.65337 155.61099 1409.25079
1012.97844 192.78315 1085.00648 914.67179 154.74712
896.2944

-8.21738 1222.32338 1146.98326 146.29353 1411.75522
1023.60405 192.02976 1091.27052 916.70983 142.32475
900.95383

-25.19247 1216.71371 1140.17815 134.82819 1411.60553
1034.11736 191.87992 1095.11887 920.87952 132.13543
903.19313

-44.68612 1207.91222 1131.50933 121.57838 1409.70032
1047.08931 191.51814 1097.16385 926.3723 122.21534
903.74763

-65.5511 1193.38272 1119.27811 102.80501 1403.08212
1061.05828 192.06709 1098.43277 933.79296 111.02368
901.81114

-65.5511 1193.38272 1119.27811 102.80501 1403.08212
1061.05828 192.06709 1098.43277 933.79296 111.02368
901.81114

LWristPinky LHand RShoulder

Z10 X11 Y11 Z11 X12 Y12 Z12 X13 Y13 Z13

1053.00369 205.81024 851.90765 1010.54009 199.01545 797.89108
1103.25615 -231.46894 1398.19733 1018.57002

1064.86321 209.02775 853.03299 1021.61927 200.28288 802.41257
1116.56311 -227.78847 1401.05316 1020.8693

1077.3909	215.81131	854.04045	1033.83309	207.27995	805.44762	1131.21758	-221.17353	1404.87946	1024.53415
1090.85442	222.24482	854.14078	1045.11894	217.77807	809.44794	1145.68848	-212.20186	1408.0275	1031.72501
1105.32891	234.19211	854.5903	1059.26819	234.35953	812.44949	1161.75896	-200.8947	1412.55478	1039.48319
1121.34209	250.25366	855.52086	1077.11174	251.17165	816.38824	1181.26183	-186.0952	1418.58124	1048.33802
1138.53897	272.61828	857.16972	1097.30606	269.3606	823.80219	1204.14101	-168.83188	1424.81216	1057.27082
1159.39041	292.86493	862.79213	1120.46685	290.00822	835.88852	1228.38982	-151.22953	1432.3407	1064.14543
1184.497	310.78673	871.46973	1145.84793	308.19742	852.72652	1256.97167	-134.86434	1438.47992	1069.60831
1214.52149	323.23572	886.13297	1186.3839	317.4427	872.77603	1295.64385	-122.88942	1441.63528	1074.50844
1248.02117	326.92742	899.88487	1228.92494	315.91276	893.7001	1336.14022	-116.23383	1442.08054	1082.48345
1283.84415	313.51242	909.23264	1270.34142	300.85259	910.95139	1374.87099	-110.81592	1443.50067	1087.31568
1316.17958	294.00847	919.18503	1311.71807	269.91572	922.79228	1416.03829	-107.41421	1444.50989	1093.99536
1345.93941	267.81244	929.16145	1352.09465	233.41087	932.15683	1452.696	-105.44605	1443.49319	1106.1995
1369.16298	246.04296	931.11717	1378.88527	198.46752	935.18784	1480.28282	-105.97195	1441.99951	1119.86573
1386.56197	230.24458	931.02348	1399.32717	182.82738	930.88722	1497.09557	-107.2284	1439.18854	1136.3588

1396.91246 223.795 925.93658 1408.10875 175.35229 919.66293
1504.03298 -106.96267 1437.42523 1148.65113

1392.13982 229.50676 916.95511 1402.808 183.25344 905.57236
1495.66536 -106.88985 1437.48993 1154.11949

1376.41793 249.3426 910.12711 1386.07002 192.90388 880.28503
1470.88265 -105.12951 1443.07266 1151.33362

1351.79268 281.2207 904.50439 1360.06127 232.10899 859.60213
1448.26988 -101.91201 1449.97711 1140.02991

1327.20406 321.57032 902.96776 1328.98934 284.3812 848.33427
1417.5753 -96.92745 1458.56369 1119.2495

1304.99588 364.05125 907.89932 1298.56385 338.98613 847.82463
1388.18932 -87.32758 1471.1908 1091.04103

1287.82677 402.36218 915.76065 1267.61307 389.00318 851.51512
1359.86885 -76.97808 1481.14944 1061.16883

1267.5071 432.92557 919.8951 1237.7826 430.25784 860.38445
1334.75655 -69.88034 1483.8295 1038.69416

1255.91179 452.1471 920.94276 1215.35179 460.53772 866.67618
1313.24394 -67.69807 1480.71091 1025.84946

1245.21866 460.11299 916.19461 1197.69837 477.46544 867.26837
1296.46172 -69.48973 1472.46933 1019.84421

1236.4132 459.72256 907.77107 1185.89676 480.64343 862.60201
1285.04997 -73.23874 1462.41669 1019.51676

1229.92592 452.14531 897.42645 1177.84874 473.35175 852.67891
1276.08643 -77.8405 1451.64917 1023.40966

1221.50063 438.36281 883.58627 1169.22508 458.21549 840.56015
1267.69181 -82.76787 1443.66547 1030.63469

1211.36528 420.36263 868.69217 1158.42324 438.81268 826.96083
1257.69737 -87.11956 1435.05859 1042.02431

1198.5479	401.39996	854.01749	1146.17691	417.47105	813.28133	1245.58098	-91.72884	1428.35953	1054.79126
1181.96633	377.50092	840.91446	1128.74932	395.96207	801.7823	1226.25099	-95.00944	1426.67603	1067.88956
1163.08556	353.41316	833.36632	1109.54735	370.85644	791.81267	1205.36576	-97.36731	1425.68283	1076.60721
1147.50725	329.28299	830.93788	1090.32425	344.82857	785.54977	1186.60477	-105.23208	1428.50342	1082.56142
1131.66222	306.32012	833.08517	1073.70964	320.61995	784.03908	1168.52654	-113.45479	1430.77972	1087.06841
1118.68325	286.40373	837.35435	1060.99266	298.24446	785.65804	1155.77782	-121.95258	1432.92938	1091.26236
1109.17763	270.24218	842.83005	1052.19063	279.26941	790.23788	1147.49855	-130.99453	1432.72705	1091.97571
1102.76978	254.55366	848.21892	1046.91674	263.98049	796.77605	1140.42435	-141.18442	1431.34277	1091.95816
1099.89502	241.91633	853.55057	1043.58513	250.5468	803.87825	1135.62088	-152.36208	1429.5903	1088.06969
1098.6332	231.03787	860.39619	1043.51837	237.54509	809.70406	1135.25856	-165.03437	1426.29425	1082.36977
1097.82707	220.7272	864.47556	1045.23194	224.15333	814.69383	1138.43933	-180.20223	1420.86319	1072.25395
1098.07839	211.61293	866.77078	1049.38637	210.26175	817.41432	1143.37273	-196.21319	1413.16299	1060.08599
1097.53067	205.66841	867.38998	1056.45088	193.95123	819.73602	1151.01097	-212.71299	1403.58902	1045.44731
1097.01775	198.64193	866.50345	1061.86157	182.2872	818.58124	1155.07775	-229.72782	1389.8172	1028.29071

1097.01775 198.64193 866.50345 1061.86157 182.2872 818.58124
1155.07775 -229.72782 1389.8172 1028.29071

ROuterElbow RWristThumb RWristPinky

X14 Y14 Z14 X15 Y15 Z15 X16 Y16 Z16

-375.49869 1102.3407 949.17999 -341.18862 900.06607
1105.89859 -277.64202 941.97098 1120.31487

-380.39055 1109.06738 957.60338 -354.02363 912.21054
1120.72159 -287.49355 951.45638 1133.5717

-384.7929 1118.14209 968.49579 -365.01274 925.10521
1137.83425 -296.77017 961.96701 1148.43712

-388.42663 1127.7372 980.27619 -375.25223 938.43109
1155.17296 -305.78987 973.11615 1164.65325

-390.3838 1141.90834 994.54788 -384.88811 953.73878
1172.06948 -315.08057 985.29831 1179.97704

-391.55392 1159.03336 1009.41872 -395.08602 972.19337
1187.14531 -321.75213 998.72528 1193.98377

-388.89057 1179.1777 1021.02829 -404.5974 990.86372
1197.25197 -329.76185 1010.81245 1202.38892

-381.44321 1198.79929 1027.44416 -414.10358 1009.92935 1200.7383
-337.94735 1022.78976 1206.51123

-371.79962 1213.90236 1026.66382 -422.12479 1024.93454
1195.78446 -345.87868 1033.43788 1205.23125

-362.57294 1223.71017 1020.94864 -430.2594 1035.16251
1184.53507 -355.03483 1040.26939 1198.86162

-356.6119 1228.19214 1014.12926 -439.97585 1038.7748
1169.80751 -365.10223 1042.23068 1189.46121

-354.05617 1235.35049 1012.61475 -448.46905 1042.20314
1157.20574 -375.11715 1044.72961 1180.3347

-356.67965 1246.03378 1015.61913 -459.37282 1049.31702 1150.5053
-385.94044 1047.12807 1174.12575

-363.28495 1260.56862 1023.80188 -472.98221 1060.45456
1148.86818 -398.77781 1052.07909 1168.95325

-371.20933 1274.37424 1035.52605 -486.24073 1070.80856
1149.25469 -412.05467 1058.12111 1165.89562

-373.91056 1282.07611 1045.81017 -500.01545 1079.09279
1147.42638 -427.00809 1060.99045 1164.85619

-371.2743 1282.14386 1052.92107 -508.65082 1079.0316
1139.63867 -436.49845 1054.55887 1155.32524

-363.72074 1273.61839 1050.20989 -508.69633 1070.77881 1117.491
-438.14522 1042.26891 1135.25833

-352.70164 1270.34279 1037.69676 -495.98595 1060.53665
1079.68994 -427.84912 1031.67374 1099.89319

-340.74654 1271.75636 1015.55397 -475.0404 1052.23412
1034.47502 -406.55308 1026.41273 1056.00868

-325.88459 1272.58339 986.31233 -450.5656 1046.11069 992.4334
-383.36197 1025.76782 1020.98076

-307.83161 1276.37169 957.30576 -426.98231 1047.1936 963.56476
-358.79562 1028.43384 994.03

-293.83547 1280.38101 936.16905 -408.63453 1048.46275 952.13021
-341.2381 1032.64198 984.79813

-290.85279 1280.48065 930.08637 -401.61289 1048.41957 960.03372
-333.11325 1035.95688 992.39846

-295.09275 1273.50563 941.30028 -405.52341 1044.69604 988.72429
-335.91087 1033.87306 1018.03665

-304.33624 1261.38489 968.02239 -416.47812 1039.33884
1039.10362 -344.15844 1028.07152 1063.89321

-312.82497 1247.37709 1001.8586 -425.43217 1037.93747
1102.40013 -351.99745 1024.47464 1122.06093

-320.57945 1235.57777 1038.08357 -428.98666 1041.91284
1168.03215 -353.94569 1028.17711 1184.56131

-324.87339 1227.69615 1072.799 -422.43332 1049.74167
1231.47637 -347.58243 1038.33778 1245.35332

-327.91332 1223.46115 1104.41506 -409.52656 1058.5817
1285.94437 -333.78773 1050.24704 1294.59633

-330.8958 1219.82788 1133.41065 -392.49485 1065.07202
1331.45317 -314.14039 1058.5704 1333.38738

-330.16018 1217.3262 1156.60851 -373.7035 1068.45963
1363.44765 -295.71096 1062.37976 1360.13756

-327.75486 1213.14529 1169.26163 -355.14377 1065.82787
1381.22277 -276.18145 1065.65857 1375.32662

-326.61251 1206.16188 1175.03067 -338.3762 1059.1555
1388.37563 -259.96384 1064.69444 1380.16533

-327.47097 1198.64029 1173.68546 -323.19107 1049.90189 1386.9532
-244.8291 1060.69351 1376.10665

-330.49732 1190.56374 1166.71776 -311.05568 1038.10463 1377.6577
-236.40007 1053.25317 1364.48677

-333.49899 1181.01173 1156.27091 -303.83976 1027.20047
1363.80013 -230.14389 1043.73215 1347.73262

-337.2245 1171.15738 1142.015 -297.10045 1011.3253
1343.51753 -224.84743 1032.49069 1327.295

-338.19351 1158.52928 1123.77991 -294.27227 993.16025
1318.27339 -223.18401 1019.03336 1305.41458

-339.33216 1144.3589 1099.51538 -293.2807 972.54715
1287.56104 -221.53501 1005.37956 1279.16764

-340.85991 1130.44022 1068.43582 -294.80257 953.11539
1252.23976 -224.33449 988.67424 1247.99065

-342.38193 1114.22424 1032.1566 -296.40779 933.55492 1210.5835
-229.27107 973.16154 1211.69243

-342.29195 1100.45204 991.79306 -300.52164 913.85971
1164.02283 -235.61254 956.47964 1172.12075

-343.30234 1086.31386 948.67302 -304.05712 893.05161 1115.886
-244.86721 940.44708 1131.09795

-343.30234 1086.31386 948.67302 -304.05712 893.05161 1115.886
-244.86721 940.44708 1131.09795

APPENDIX B

Source code for MocapXtractPro

```
Option Explicit

'*******************************************************************
**********************
' modTest
' Description:
' This module contains the main test function that run through all
the tests.
' It also contains the test functions for each function for the
component.
'*******************************************************************
**********************

'*******************************************************************
**********************
' Enum and constant section
' (add all enums and constants here)
'*******************************************************************
**********************
Private Const mstrMOD_NAME As String = "modProcess"

Private Const DB_PATH As String = "Data.mdb"
Private Const TBL_MATCH As String = "tblFile"
Private Const FLD_FILENAME As String = "FileName"
Private Const FLD_FRAMENO As String = "FrameNo"
Private Const FLD_X As String = "X"
```

```
Private Const FLD_Y As String = "Y"
Private Const FLD_Z As String = "Z"
Private Const FLD_DSMEASURE As String = "DistanceMeasurement"

Public Const TYPE_CREATE = 1
Public Const TYPE_INSERT = 2


' Constants for input file delimiters and escape character
Public Const INFILE_ESC = "\"

Public Const DIR_FILES As String = "TRC"
Public Const LINE_HEADER As Integer = 4
Public Const LINE_DATA As Integer = 6
Public Const DEFAULT_SPACE As Integer = 2

Public Enum OPT_VALUES
   OPT_LWRIST = 1
   OPT_RWRIST = 2
   OPT_LELBOW = 3
   OPT_RELBOW = 4
   OPT_LTOE = 5
   OPT_RTOE = 6
End Enum

Public Enum SRCH_CRITERIA
   SRCH_INWORDS = 0
   SRCH_SPACE = 1
   SRCH_START_POS = 2
'   SRCH_POS = 3
   SRCH_START_X_POS = 3
   SRCH_START_Y_POS = 4
   SRCH_START_Z_POS = 5
   SRCH_X_POS = 6
   SRCH_Y_POS = 7
   SRCH_Z_POS = 8
End Enum

Private Type TRCInfo
   sngXValue As Single
   sngYValue As Single
```

```
   sngZValue As Single
End Type

Private Type TRCsInfo
   strFileName As String
   sngMeanX As Single
   sngMeanY As Single
   sngMeanZ As Single
   lngTotalFrame As Long
   boolIsMatch As Boolean
   atypTRCs() As TRCInfo
End Type

Public Type TblFld
   Name As String
   Type As String
End Type

'*******************************************************************
*********************
' Variable declaration section
' (add all variables declaration here)
'*******************************************************************
*********************
Private mtypMasterTRC As TRCsInfo
Private matypTRC() As TRCsInfo

'*******************************************************************
*********************
' Function : SearchMatchingFile
'            - returns:
' Purpose  :

' Modified :
' Notes    :
'*******************************************************************
*********************
Public Function SearchMatchingFile(ByVal strMasterFilePath, ByVal
strOption As String) _
                                    As Long
   Const FUNC_NAME As String = "SearchMatchingFile"
```

```
   On Error GoTo ErrHandler

   Dim strMatchedItem As String

   Call MapItems(strOption, strMatchedItem)

   If strMatchedItem = "" Then
      MsgBox "No matched item", vbExclamation
      Exit Function
   End If

   If Dir(strMasterFilePath) = "" Then
      MsgBox "File: " & strMasterFilePath & " not found",
vbExclamation
      Exit Function
   End If

   Call AnalystMasterTRC(strMasterFilePath, strMatchedItem)
   Call AnalystTRCs(strMatchedItem)

   Call AnalystProcess

   Exit Function
ErrHandler:
   SearchMatchingFile = -1
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Function

'*******************************************************************
*********************
' Function : AnalystMasterTRC
'            - returns
' Purpose  :
' Modified :
' Notes    :
'*******************************************************************
*********************
Private Function AnalystMasterTRC(ByVal strPath As String, ByVal
strItem As String) As Long
   Const FUNC_NAME As String = "AnalystMasterTRC"
   Const PROCESS_TYPE As String = "Master"
```

```
   On Error GoTo ErrHandler

   mtypMasterTRC.strFileName = Mid(strPath, InStrRev(strPath,
INFILE_ESC) + 1)

   Call ProcessFile(strPath, strItem, PROCESS_TYPE)

   Exit Function
ErrHandler:
   AnalystMasterTRC = -1
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Function

'*******************************************************************
*********************
' Function : AnalystTRCs
'            - returns
' Purpose  :
' Modified :
' Notes    :
'*******************************************************************
*********************
Private Function AnalystTRCs(ByVal strItem As String) As Long
   Const FUNC_NAME As String = "AnalystTRCs"
   Const PROCESS_TYPE As String = "TRC"
   On Error GoTo ErrHandler

   Dim objFSO As FileSystemObject
   Dim objFiles As Files
   Dim objFile As File
   Dim objFolder As Folder

   Dim strPath As String

   Set objFSO = New FileSystemObject

   On Error Resume Next
   Set objFolder = objFSO.GetFolder(App.Path & "\" & DIR_FILES)
   On Error GoTo ErrHandler

   If objFolder Is Nothing Then
```

```
      MsgBox "Default TRC path not found", vbExclamation
      Exit Function
   End If

   On Error Resume Next
   Set objFiles = objFolder.Files
   On Error GoTo ErrHandler

   If objFiles Is Nothing Then
      MsgBox "No TRC file", vbExclamation
      Exit Function
   End If

   For Each objFile In objFiles
      If Not objFile Is Nothing Then
         strPath = objFile.Path
         Call UpdateTRCFileNameProperty(strPath)
         Call ProcessFile(strPath, strItem, PROCESS_TYPE)
      End If
   Next objFile

   Exit Function
ErrHandler:
   AnalystTRCs = -1
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Function

'*******************************************************************
**********************
' Function : ProcessFile
'            - returns
' Purpose  :
' Modified :
' Notes    :
'*******************************************************************
**********************
Private Function ProcessFile(ByVal strPath As String, ByVal strItem
As String, _
                              Optional ByVal strType As String) As
Long
   Const FUNC_NAME As String = "ProcessFile"
```

```
   On Error GoTo ErrHandler

   Dim objFSO As FileSystemObject
   Dim txtStream As TextStream
   Dim strLine As String
   Dim lngMatchPos As Long
   Dim lngCounter As Long

   Set objFSO = New FileSystemObject
   Set txtStream =
objFSO.GetFile(strPath).OpenAsTextStream(ForReading)

   lngCounter = 0
   If Not txtStream Is Nothing Then
      Do While txtStream.AtEndOfStream = False
         lngCounter = lngCounter + 1
         strLine = txtStream.ReadLine

         If lngCounter = LINE_HEADER Then
            Call ReadHeader(strLine, strItem, lngMatchPos)
         ElseIf lngCounter >= LINE_DATA Then
            Call ReadData(strLine, lngMatchPos, strType)
         End If

         DoEvents
      Loop
   End If

   Call txtStream.Close

   Exit Function
ErrHandler:
   ProcessFile = -1
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description

End Function

'*******************************************************************
*********************
' Function : ReadHeader
'            - returns
```

```
' Purpose  :
' Modified :
' Notes    :
'*******************************************************************
*********************
Private Function ReadHeader(ByVal strContent As String, ByVal
strItem As String, _
                              ByRef lngFieldPos As Long) As Long
   Const FUNC_NAME As String = "ReadHeader"
   On Error GoTo ErrHandler

   Dim strData As String
   Dim strTemp As String
   Dim lngMatchPos As Long
   Dim lngSearchType As Long
   Dim i As Long

   strItem = "RBackHead"

   If Len(strContent) = 0 Then Exit Function

   strData = ""
   lngMatchPos = 0
   For i = 0 To Len(strContent)
      strTemp = Mid(strContent, i + 1, 1)
      ' 0 means still in a word
      If lngSearchType = SRCH_INWORDS Then
         If Not IsWhiteSpace(strTemp) Then
            strData = strData & strTemp
         Else
            If StrComp(Trim(strData), Trim(strItem), vbTextCompare)
= 0 Then
               lngFieldPos = lngMatchPos

               Debug.Print "match " & strData & " at pos:" &
lngMatchPos
               Exit Function
            End If
            lngMatchPos = lngMatchPos + 1
            strData = ""
            lngSearchType = SRCH_SPACE
```

```
         End If
      ' otherwise, means found whitespace
      Else
         If Not IsWhiteSpace(strTemp) Then
            strData = strData & strTemp
            lngSearchType = SRCH_INWORDS
         End If
      End If
   Next i

   Exit Function
ErrHandler:
   ReadHeader = -1
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Function

'*******************************************************************
*********************
' Function : ReadData
'            - returns
' Purpose  :
' Modified :
' Notes    :
'*******************************************************************
*********************
Private Function ReadData(ByVal strContent As String, ByVal
lngFieldPos As Long, _
                           ByVal strType As String) As Long
   Const FUNC_NAME As String = "ReadData"
   On Error GoTo ErrHandler

   Dim strData As String
   Dim strTemp As String
   Dim lngPos As Long
   Dim lngSearchType As Long
   Dim i As Long

   If Len(strContent) = 0 Then Exit Function

   lngSearchType = SRCH_START_POS
   strData = ""
```

```
   lngPos = 0
   For i = 0 To Len(strContent)
      strTemp = Mid(strContent, i + 1, 1)
      If lngSearchType = SRCH_START_POS Then
         If IsWhiteSpace(strTemp) Then
            lngPos = lngPos + 1
            If DEFAULT_SPACE = lngPos Then
               If lngFieldPos = lngPos Then
                  lngSearchType = SRCH_X_POS
               Else
                  lngSearchType = SRCH_START_X_POS
               End If
            End If
         End If
      ElseIf lngSearchType = SRCH_START_X_POS Then
         If IsWhiteSpace(strTemp) Then

            lngSearchType = SRCH_START_Y_POS
         End If
      ElseIf lngSearchType = SRCH_START_Y_POS Then
         If IsWhiteSpace(strTemp) Then

            lngSearchType = SRCH_START_Z_POS
         End If
      ElseIf lngSearchType = SRCH_START_Z_POS Then
         If IsWhiteSpace(strTemp) Then
            lngPos = lngPos + 1
            If lngFieldPos = lngPos Then
               lngSearchType = SRCH_X_POS
            Else
               lngSearchType = SRCH_START_X_POS
            End If
         End If
      ElseIf lngSearchType = SRCH_X_POS Then
         If Not IsWhiteSpace(strTemp) Then
            strData = strData & strTemp
         Else
'             Debug.Print "X:" & strData
            Call UpdateProperty(strType, strData, "X")
            lngSearchType = SRCH_Y_POS
            strData = ""
```

```
         End If
      ElseIf lngSearchType = SRCH_Y_POS Then
         If Not IsWhiteSpace(strTemp) Then
            strData = strData & strTemp
         Else
'            Debug.Print "Y:" & strData
            Call UpdateProperty(strType, strData, "Y")
            lngSearchType = SRCH_Z_POS
            strData = ""
         End If
      ElseIf lngSearchType = SRCH_Z_POS Then
         If Not IsWhiteSpace(strTemp) Then
            strData = strData & strTemp
         Else
'            Debug.Print "Z:" & strData
            Call UpdateProperty(strType, strData, "Z")
            Exit Function
         End If
      End If
   Next i

   Exit Function
ErrHandler:
   ReadData = -1
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Function

'*******************************************************************
*********************
' Function : UpdateProperty
'            - returns
' Purpose  :

' Modified :
' Notes    :
'*******************************************************************
*********************
Private Function UpdateProperty(ByVal strType As String, ByVal
strData As String, _
                                 ByVal strField As String) As Long
   Const FUNC_NAME As String = "UpdateProperty"
```

```
   On Error GoTo ErrHandler

   Dim sngValue As Single

   If UCase(strType) = UCase("Master") Then
      If IsNumeric(strData) Then
         sngValue = CSng(strData)
         Call UpdateMasterProperty(sngValue, strField)
      End If
   ElseIf UCase(strType) = UCase("TRC") Then
      If IsNumeric(strData) Then
         sngValue = CSng(strData)
         Call UpdateTRCProperty(sngValue, strField)
      End If
   End If

   Exit Function
ErrHandler:
   UpdateProperty = -1
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Function

'********************************************************************
*********************
' Function : UpdateMasterProperty
'           - returns
' Purpose  :
' Modified :
' Notes    :
'********************************************************************
*********************
Private Function UpdateMasterProperty(ByVal sngValue As Single, _
                                       ByVal strField As String) As
Long
   Const FUNC_NAME As String = "UpdateMasterProperty"
   On Error GoTo ErrHandler

   Dim lngUBound As Long

   With mtypMasterTRC
      On Error Resume Next
```

```
      lngUBound = UBound(.atypTRCs)
      If Err.Number <> 0 Then
         lngUBound = -1
      End If
      On Error GoTo ErrHandler

      If UCase("X") = UCase(strField) Then
         ReDim Preserve .atypTRCs(0 To (lngUBound + 1))

         .lngTotalFrame = .lngTotalFrame + 1
         With .atypTRCs(lngUBound + 1)
            .sngXValue = sngValue
         End With
      ElseIf UCase("Y") = UCase(strField) Then
         With .atypTRCs(lngUBound)
            .sngYValue = sngValue
         End With
      ElseIf UCase("Z") = UCase(strField) Then
         With .atypTRCs(lngUBound)
            .sngZValue = sngValue
         End With
      End If
   End With

   Exit Function
ErrHandler:
   UpdateMasterProperty = -1
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Function

'********************************************************************
**********************
' Function : UpdateTRCProperty
'            - returns
' Purpose  :
' Modified :
' Notes    :
'********************************************************************
**********************
Private Function UpdateTRCProperty(ByVal sngValue As Single, _
```

```
                                    ByVal strField As String) As
Long
   Const FUNC_NAME As String = "UpdateTRCProperty"
   On Error GoTo ErrHandler

   Dim lngUBound As Long
   Dim lngInnerUBound As Long

   On Error Resume Next
   lngUBound = UBound(matypTRC)
   If Err.Number <> 0 Then Exit Function
   On Error GoTo ErrHandler

   With matypTRC(lngUBound)
      On Error Resume Next
      lngInnerUBound = UBound(.atypTRCs)
      If Err.Number <> 0 Then
         lngInnerUBound = -1
      End If
      On Error GoTo ErrHandler

      If UCase("X") = UCase(strField) Then
         ReDim Preserve .atypTRCs(0 To (lngInnerUBound + 1))

         .lngTotalFrame = .lngTotalFrame + 1
         With .atypTRCs(lngInnerUBound + 1)
            .sngXValue = sngValue
         End With
      ElseIf UCase("Y") = UCase(strField) Then
         With .atypTRCs(lngInnerUBound)
            .sngYValue = sngValue
         End With
      ElseIf UCase("Z") = UCase(strField) Then
         With .atypTRCs(lngInnerUBound)
            .sngZValue = sngValue
         End With
      End If
   End With

   Exit Function
ErrHandler:
```

```
   UpdateTRCProperty = -1
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Function

'*******************************************************************
*********************
' Function : UpdateTRCFileNameProperty
' Purpose  :
' Modified :
' Notes    :
'*******************************************************************
*********************
Private Function UpdateTRCFileNameProperty(ByVal strPath As String)
As Long
   Const FUNC_NAME As String = "UpdateTRCFileNameProperty"
   On Error GoTo ErrHandler

   Dim lngUBound As Long

   On Error Resume Next
   lngUBound = UBound(matypTRC)
   If Err.Number <> 0 Then
      lngUBound = 0
      ReDim matypTRC(0 To lngUBound)
      Err.Clear
   Else
      lngUBound = lngUBound + 1
      ReDim Preserve matypTRC(0 To lngUBound)
   End If
   On Error GoTo ErrHandler

   With matypTRC(lngUBound)
      .strFileName = Mid(strPath, InStrRev(strPath, INFILE_ESC) + 1)
   End With

   Exit Function
ErrHandler:
   UpdateTRCFileNameProperty = -1
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Function
```

```
'*****************************************************************
*********************
' Function : AnalystProcess
'            - returns
' Purpose  :

' Modified :
' Notes    :
'*****************************************************************
*********************
Private Function AnalystProcess() As Long
   Const FUNC_NAME As String = "AnalystProcess"
   On Error GoTo ErrHandler

   Call CalcMasterMean
   Call CalcTRCMean

   Call CalcPCA
   Call ProduceMatchFiles

   Exit Function
ErrHandler:
   AnalystProcess = -1
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Function

'*****************************************************************
*********************
' Function : CalcMasterMean
'            - returns
' Purpose  :
' Modified :
' Notes    :
'*****************************************************************
*********************
Private Function CalcMasterMean() As Long
   Const FUNC_NAME As String = "CalcMasterMean"
   On Error GoTo ErrHandler

   Dim sngX As Single
   Dim sngY As Single
```

```
   Dim sngZ As Single
   Dim lngUBound As Long
   Dim lngCounter As Long
   Dim i As Long

   With mtypMasterTRC
      On Error Resume Next
      lngUBound = UBound(.atypTRCs)
      If Err.Number <> 0 Then
         MsgBox "Master file no data"
         Exit Function
      End If
      On Error GoTo ErrHandler

      For i = LBound(.atypTRCs) To UBound(.atypTRCs)
         With .atypTRCs(i)
            sngX = sngX + .sngXValue
            sngY = sngY + .sngYValue
            sngZ = sngZ + .sngZValue
            lngCounter = lngCounter + 1
         End With
      Next i

      If lngCounter > 0 Then
         .sngMeanX = sngX / lngCounter
         .sngMeanY = sngY / lngCounter
         .sngMeanZ = sngZ / lngCounter
      End If

   End With

   Exit Function
ErrHandler:
   CalcMasterMean = -1
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Function

'*******************************************************************
*********************
' Function : CalcTRCMean
'             - returns
```

```
' Purpose  :
' Modified :
' Notes    :
'*******************************************************************
*********************
Private Function CalcTRCMean() As Long
   Const FUNC_NAME As String = "CalcTRCMean"
   On Error GoTo ErrHandler

   Dim sngX As Single
   Dim sngY As Single
   Dim sngZ As Single
   Dim lngUBound As Long
   Dim lngInnerUBound As Long
   Dim lngCounter As Long
   Dim i As Long
   Dim j As Long

   On Error Resume Next
   lngUBound = UBound(matypTRC)
   If Err.Number <> 0 Then
      MsgBox "Error with no data"
      Exit Function
   End If
   On Error GoTo ErrHandler

   For i = LBound(matypTRC) To UBound(matypTRC)
      With matypTRC(i)
         On Error Resume Next
         lngInnerUBound = UBound(.atypTRCs)
         If Err.Number = 0 Then
            On Error GoTo ErrHandler
            sngX = 0
            sngY = 0
            sngZ = 0
            lngCounter = 0
            For j = LBound(.atypTRCs) To UBound(.atypTRCs)
               With .atypTRCs(j)
                  sngX = sngX + .sngXValue
                  sngY = sngY + .sngYValue
                  sngZ = sngZ + .sngZValue
```

```
                lngCounter = lngCounter + 1
             End With
          Next j

          If lngCounter > 0 Then
             .sngMeanX = sngX / lngCounter
             .sngMeanY = sngY / lngCounter
             .sngMeanZ = sngZ / lngCounter
          End If
       Else
          Err.Clear
          On Error GoTo ErrHandler
       End If
    End With
 Next i

 Exit Function
ErrHandler:
   CalcTRCMean = -1
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Function

'*******************************************************************
*********************
' Function : CalcPCA
'            - returns
' Purpose  :
' Modified :
' Notes    :
'*******************************************************************
*********************
Private Function CalcPCA() As Long
   Const FUNC_NAME As String = "CalcPCA"
   On Error GoTo ErrHandler

   Dim sngMeanX As Single
   Dim sngMeanY As Single
   Dim sngMeanZ As Single
   Dim sngRes As Single
   Dim lngUBound As Long
   Dim i As Long
```

```
   With mtypMasterTRC
      sngMeanX = .sngMeanX
      sngMeanY = .sngMeanY
      sngMeanZ = .sngMeanZ
   End With

'    Debug.Print "Mean Master(x,y,z): " & sngMeanX & ", " & sngMeanY
& ", " & sngMeanZ

   On Error Resume Next
   lngUBound = UBound(matypTRC)
   If Err.Number <> 0 Then
      MsgBox "Error with no data"
      Exit Function
   End If
   On Error GoTo ErrHandler

   For i = LBound(matypTRC) To UBound(matypTRC)
      sngRes = 0
      With matypTRC(i)
         sngRes = CalcDistanceMeasurement(sngMeanX, sngMeanY,
sngMeanZ, .sngMeanX, .sngMeanY, .sngMeanZ, 100)

'          Debug.Print "(" & .strFileName & "): " & .sngMeanX & ", "
& .sngMeanY & ", " & .sngMeanZ
'          Debug.Print "DS: " & sngRes

         If (sngRes > 0 And sngRes <= 1) Then
            .boolIsMatch = True
         Else
            .boolIsMatch = False
         End If
      End With
   Next i

   Exit Function
ErrHandler:
   CalcPCA = -1
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Function
```

```
'*****************************************************************
*********************
' Function : CalcDistanceMeasurement
'            - returns
' Purpose  :
' Modified :
' Notes    :
'*****************************************************************
*********************
Private Function CalcDistanceMeasurement(ByVal sngXRef As Single,
ByVal sngYRef As Single, _
                                          ByVal sngZRef As Single,
ByVal sngX As Single, _
                                          ByVal sngY As Single,
ByVal sngZ As Single, _
                                          Optional ByVal lngDivider
As Long = 0) As Single
   Const FUNC_NAME As String = "CalcDistanceMeasurement"
   On Error GoTo ErrHandler

   Dim sngRes As Single

   sngRes = Sqr(((sngXRef - sngX) ^ 2) + ((sngYRef - sngY) ^ 2) +
((sngZRef - sngZ) ^ 2))

   If lngDivider > 0 Then
      CalcDistanceMeasurement = sngRes / lngDivider
   Else
      CalcDistanceMeasurement = sngRes
   End If

   Exit Function
ErrHandler:
   CalcDistanceMeasurement = -1
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Function

'*****************************************************************
*********************
' Function : ProduceMatchFiles
```

```
'            - returns
' Purpose  :
' Modified :
' Notes    :
'*******************************************************************
*********************
Private Function ProduceMatchFiles() As Long
   Const FUNC_NAME As String = "ProduceMatchFiles"
   On Error GoTo ErrHandler

   Dim lngUBound As Long
   Dim i As Long

   On Error Resume Next
   lngUBound = UBound(matypTRC)
   If Err.Number <> 0 Then Exit Function
   On Error GoTo ErrHandler

   For i = LBound(matypTRC) To UBound(matypTRC)
      With matypTRC(i)
         If .boolIsMatch Then
            frmMain.lstFile.AddItem (.strFileName)
         End If
      End With
   Next i

   Exit Function
ErrHandler:
   ProduceMatchFiles = -1
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Function

'*******************************************************************
*********************
' Function : ExportFile
' Purpose  :
' Modified :
' Notes    :
'*******************************************************************
*********************
```

```
Public Function ExportFile(Optional ByVal strExportPath As String)
As Long
   Const FUNC_NAME As String = "ExportFile"
   On Error GoTo ErrHandler

   Dim cnObj As ADODB.Connection
   Dim atypTblFld() As TblFld
   Dim strDBPath As String
   Dim strTblFld As String

   If strExportPath = "" Then
      strDBPath = App.Path & "\" & DB_PATH
   Else
      strDBPath = strExportPath & "\" & DB_PATH
   End If

   Set cnObj = CreateObject("ADODB.Connection")

   cnObj.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
              "Data Source = " & strDBPath & ";"

   If TableExists(cnObj, TBL_MATCH) Then
      cnObj.Execute "DROP TABLE " & TBL_MATCH
   End If

   Call TableFieldGet(TBL_MATCH, atypTblFld())
   strTblFld = StringConvert(atypTblFld(), TYPE_CREATE)

   Call CreateTable(cnObj, TBL_MATCH, strTblFld)

   Call InsertDataM(cnObj, TBL_MATCH)

   Exit Function

ErrHandler:
   If Err.Number = -2147467259 Then
      MsgBox "Database do not exist"
      Exit Function
   End If
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
   ExportFile = -1
```

```
End Function

'*****************************************************************
*********************
' Function : InsertData
' Purpose  :
' Modified :
' Notes    :
'*****************************************************************
*********************
Private Function InsertData(ByVal cnObj As ADODB.Connection, _
                             ByVal strTblName As String) As Long
   Const FUNC_NAME As String = "InsertData"
   On Error GoTo ErrHandler

   Dim strSQL As String
   Dim strFileName As String
   Dim sngX As Single, sngY As Single, sngZ As Single
   Dim sngRes As Single
   Dim lngUBound As Long
   Dim lngInnerUBound As Long
   Dim lngCount As Long
   Dim i As Long, j As Long

   On Error Resume Next
   lngUBound = UBound(matypTRC)
   If Err.Number <> 0 Then Exit Function
   On Error GoTo ErrHandler

   For i = LBound(matypTRC) To UBound(matypTRC)
      With matypTRC(i)
         If .boolIsMatch Then
            On Error Resume Next
            lngInnerUBound = UBound(.atypTRCs)
            If Err.Number <> 0 Then Exit Function
            On Error GoTo ErrHandler

            sngX = .sngMeanX
            sngY = .sngMeanY
            sngZ = .sngMeanZ
            strFileName = .strFileName
```

```
            lngCount = 0
            For j = LBound(.atypTRCs) To UBound(.atypTRCs)
               With .atypTRCs(j)
                  lngCount = lngCount + 1
                  sngRes = CalcDistanceMeasurement(sngX, sngY, sngZ,
_
                                           .sngXValue, .sngYValue,
.sngZValue, 100)

                  strSQL = "INSERT INTO " & strTblName & _
                           " (" & FLD_FILENAME & ", " & FLD_FRAMENO
& ", " & FLD_X & "," & _
                           FLD_Y & ", " & FLD_Z & ", " &
FLD_DSMEASURE & ") Values " & _
                           "('" & strFileName & "', " & lngCount &
", " & .sngXValue & _
                           ", " & .sngYValue & ", " & .sngZValue &
", " & sngRes & ")"
                  cnObj.Execute strSQL, , adCmdText
               End With
               DoEvents
            Next j
         End If
      End With
   Next i

   Exit Function

ErrHandler:
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
   InsertData = -1
End Function

'*******************************************************************
**********************
' Function : InsertDataM
' Purpose  :
' Modified :
' Notes    :
'*******************************************************************
**********************
```

```
Private Function InsertDataM(ByVal cnObj As ADODB.Connection, _
                              ByVal strTblName As String) As Long
   Const FUNC_NAME As String = "InsertDataM"
   On Error GoTo ErrHandler

   Dim strSQL As String
   Dim strFileName As String
   Dim sngX As Single, sngY As Single, sngZ As Single
   Dim sngRes As Single
   Dim lngUBound As Long
   Dim lngInnerUBound As Long
   Dim lngMasterUBound As Long
   Dim lngMasterInnerUBound As Long
   Dim lngCount As Long
   Dim i As Long, j As Long

   On Error Resume Next
   lngMasterUBound = UBound(mtypMasterTRC.atypTRCs)
   If Err.Number <> 0 Then Exit Function

   lngUBound = UBound(matypTRC)
   If Err.Number <> 0 Then Exit Function
   On Error GoTo ErrHandler

   For i = LBound(matypTRC) To UBound(matypTRC)
      With matypTRC(i)
         If .boolIsMatch Then
            strFileName = .strFileName

            On Error Resume Next
            lngInnerUBound = UBound(.atypTRCs)
            If Err.Number <> 0 Then Exit Function
            On Error GoTo ErrHandler

            lngCount = 0
            For j = LBound(.atypTRCs) To UBound(.atypTRCs)
               With .atypTRCs(j)
                  lngCount = lngCount + 1

                  With mtypMasterTRC
                     sngX = .atypTRCs(j).sngXValue
```

```
                  sngY = .atypTRCs(j).sngYValue
                  sngZ = .atypTRCs(j).sngZValue
               End With

               sngRes = CalcDistanceMeasurement(sngX, sngY, sngZ,
_
                                    .sngXValue, .sngYValue,
.sngZValue, 100)

               strSQL = "INSERT INTO " & strTblName & _
                        " (" & FLD_FILENAME & ", " & FLD_FRAMENO
& ", " & FLD_X & "," & _
                        FLD_Y & ", " & FLD_Z & ", " &
FLD_DSMEASURE & ") Values " & _
                        "('" & strFileName & "', " & lngCount &
", " & .sngXValue & _
                        ", " & .sngYValue & ", " & .sngZValue &
", " & sngRes & ")"
               cnObj.Execute strSQL, , adCmdText
            End With
            DoEvents
         Next j
      End If
   End With
Next i

Exit Function

ErrHandler:
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
   InsertDataM = -1
End Function

'*******************************************************************
**********************
' Function : TableExists
' Purpose  :
' Modified :
' Notes    :
'*******************************************************************
**********************
```

```
Private Function TableExists(ByVal cnObj As ADODB.Connection, _
                              ByVal strTblName As String) As Boolean
   Const FUNC_NAME As String = "TableExists"
   On Error GoTo ErrHandler

   Dim rsObj As ADODB.Recordset

   Set rsObj = CreateObject("ADODB.Recordset")

   Set rsObj = cnObj.OpenSchema(adSchemaTables, _
                  Array(Empty, Empty, strTblName, Empty))

   Do Until rsObj.EOF
      If rsObj!table_name = strTblName Then
         TableExists = True
         rsObj.Close
         Set rsObj = Nothing
         Exit Function
      End If
      rsObj.MoveNext
   Loop

   rsObj.Close
   Set rsObj = Nothing

   TableExists = False
   Exit Function

ErrHandler:
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
   TableExists = False
End Function

Public Sub TableFieldGet(ByVal strTableType As String, ByRef
atypTblFld() As TblFld, _
                         Optional strInput As String)
   Const FUNC_NAME As String = "TableFieldGet"
   On Local Error GoTo ErrHandler

   Dim vntFixFldNm As Variant
   Dim vntFixFldType As Variant
```

```
   Dim intIndex As Integer

   Select Case strTableType
      Case TBL_MATCH
         vntFixFldNm = Array(FLD_FILENAME, FLD_FRAMENO, FLD_X, _
                             FLD_Y, FLD_Z, FLD_DSMEASURE)

         vntFixFldType = Array("Text (50)", "Integer", _
                              "Single", "Single", "Single",
"Single")
   End Select

   ReDim atypTblFld(UBound(vntFixFldNm))

   For intIndex = LBound(vntFixFldNm) To UBound(vntFixFldNm)
      atypTblFld(intIndex).Name = vntFixFldNm(intIndex)
      atypTblFld(intIndex).Type = vntFixFldType(intIndex)
      DoEvents
   Next intIndex

   Exit Sub

ErrHandler:
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Sub

'*******************************************************************
*********************
' Function : StringConvert
' Purpose  :
' Modified :
' Notes    :
'*******************************************************************
*********************
Public Function StringConvert(ByRef atypTblFld() As TblFld, ByVal
strType As String)
   Const FUNC_NAME As String = "StringConvert"
   On Error GoTo ErrHandler

   Dim intIndex As Integer
   Dim strFld As String
```

```
   Select Case strType
      Case TYPE_CREATE
         For intIndex = LBound(atypTblFld) To UBound(atypTblFld)
            strFld = strFld & "[" & _
                     atypTblFld(intIndex).Name & _
                     "] " & atypTblFld(intIndex).Type
            If intIndex < UBound(atypTblFld) Then
               strFld = strFld + ", "
            End If
            DoEvents
         Next intIndex
         StringConvert = strFld

      Case TYPE_INSERT
         For intIndex = LBound(atypTblFld) To UBound(atypTblFld)
            strFld = strFld & "[" & _
                     atypTblFld(intIndex).Name & "]"
            If intIndex < UBound(atypTblFld) Then
               strFld = strFld + ", "
            End If
            DoEvents
         Next intIndex
         StringConvert = strFld
   End Select

   Exit Function

ErrHandler:
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Function

'*********************************************************************
*********************
' Function : CreateTable
' Purpose  :
' Modified :
' Notes    :
'*********************************************************************
*********************
```

```
Private Sub CreateTable(ByVal cnObj As ADODB.Connection, ByVal
strTblNm As String, _
                        ByVal strTblFld As String)
   Const FUNC_NAME As String = "CreateTable"
   On Local Error GoTo ErrHandler

   cnObj.Execute "CREATE TABLE " & strTblNm & _
                  "(" & strTblFld & ");", , adCmdText
   Exit Sub

ErrHandler:
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Sub

'*******************************************************************
*********************
' Function : IsWhiteSpace
'            - strByte
'              A byte to check if it is a white space
'            - returns
'              True is it is a white space, False otherwise
' Purpose  : To check if a character is white space
' Modified :
' Notes    : - Space, and all characters before &H20 are considered
white space
'            - If more than 1 character is passed in, only the first
character
'              is checked
'*******************************************************************
*********************
Private Function IsWhiteSpace(ByVal strByte As String) As Boolean
   Const FUNC_NAME As String = "IsWhiteSpace"
   On Error GoTo ErrHandler

   Dim lngChar As Long

   lngChar = Asc(strByte)
   If lngChar <= &H20 Then IsWhiteSpace = True

   Exit Function
```

```
ErrHandler:
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
   IsWhiteSpace = False
End Function

'*******************************************************************
*********************
' Function : ClearMasterProperty
' Purpose  :
' Modified :
' Notes    :
'*******************************************************************
*********************
Public Sub ClearMasterProperty()
   Const FUNC_NAME As String = "ClearMasterProperty"
   On Error GoTo ErrHandler

   With mtypMasterTRC
      .lngTotalFrame = 0
      .sngMeanX = 0
      .sngMeanY = 0
      .sngMeanZ = 0
      .strFileName = ""
      Erase .atypTRCs
   End With

   Exit Sub
ErrHandler:
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Sub

'*******************************************************************
*********************
' Function : ClearTRCProperty
' Purpose  :
' Modified :
' Notes    :
'*******************************************************************
*********************
Public Sub ClearTRCProperty()
   Const FUNC_NAME As String = "ClearTRCProperty"
```

```
   On Error GoTo ErrHandler

   Erase matypTRC

   Exit Sub
ErrHandler:
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
End Sub

'*******************************************************************
*********************
' Function : MapItems
'            - returns
' Purpose  :
' Modified :
' Notes    :
'*******************************************************************
*********************
Public Function MapItems(ByVal enuOption As OPT_VALUES, ByRef
strMatchedItem As String) As Long
   Const FUNC_NAME As String = "MapItems"

   On Error GoTo ErrorHandler

   Select Case enuOption
      Case OPT_LWRIST
         strMatchedItem = "LHand"
      Case OPT_RWRIST
         strMatchedItem = "RHand"
      Case OPT_LELBOW
         strMatchedItem = "LOuterElbow"
      Case OPT_RELBOW
         strMatchedItem = "ROuterElbow"
      Case OPT_LTOE
         strMatchedItem = "LToe"
      Case OPT_RTOE
         strMatchedItem = "RToe"
      Case Else
         strMatchedItem = ""
   End Select
```

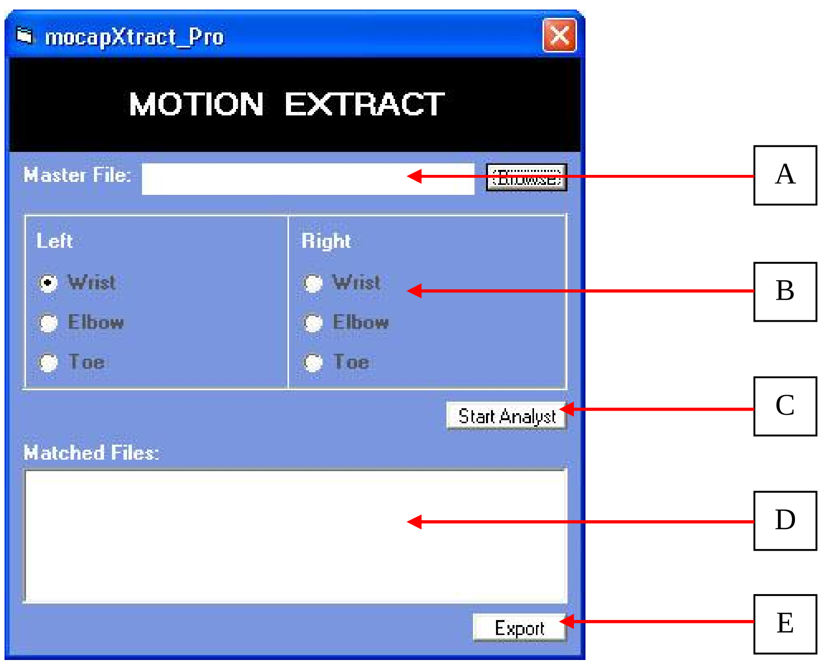
```
   Exit Function

ErrorHandler:
   ' Message box is fine for test function
   MsgBox mstrMOD_NAME & ":" & FUNC_NAME & ":" & Err.Description
   MapItems = -1

End Function
```

APPENDIX C

MocapXtractPro User Interface



A	***Control Dialog***	- *Input location for the example motion segment*
B	**Option Button**	- Principle marker selection
C	**Calculation Button**	- Initiates the search and extraction process
D	**Motion Match Display**	- The matching motion capture file are displayed here**.**

E	**Export Button** Microsoft	- This button exports the extracted data into Access Database.

Appendix D

MocapXtractPro UI During Motion Extraction