Title of thesis

REDUCING LATENCY IN A VIRTUAL REALITY-BASED TRAINING APPLICATION

I, YULITA HANUM BINTI P ISKANDAR hereby allow my thesis to be placed at the Information Resource Centre (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP.
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as

[ ] Confidential

[X] Non-confidential

If this thesis is confidential, please state the reason:

____

____

____

The contents of the thesis will remain confidential for __________ years.

Remarks on disclosure:

____

____

____

Endorsed by:

ykı

Signature of Author

Signature of Supervisor

Permanent Address:
2-4-N Taman Boundary,
Lintang Sempadan,
11400, Penang

Date : 10th July 2006

Name of Supervisor:
Associate Professor Dr. Abas Md. Said

Date : 10th July 2006

UNIVERSITI TEKNOLOGI PETRONAS

Approval by Supervisor

The undersigned certify that they have read, and recommend to The Postgraduate Studies Programme for acceptance, a thesis entitled

Reducing Latency in a Virtual Reality-based Training Application

submitted by

Yulita Hanum binti P Iskandar

for the fulfilment of the requirements for the degree of

Masters of Science in Information Technology

10th July 2006
Date

Signature :

Main Supervisor : ASSOCIATE PROFESSOR DR. ABAS MD. SAID

Date : 10th July 2006

Co-Supervisor 1 :

Co-Supervisor 2 :

UNIVERSITI TEKNOLOGI PETRONAS

Reducing Latency in a Virtual Reality-based Training Application

by

Yulita Hanum binti P Iskandar

A THESIS

SUBMITTED TO THE POSTGRADUATE STUDIES PROGRAMME

AS A REQUIREMENT FOR THE

DEGREE OF MASTERS OF SCIENCE IN INFORMATION TECHNOLOGY

Information & Communication Technology Programme

BANDAR SERI ISKANDAR,

PERAK

JULY 2006

DECLARATION

I hereby declare that the thesis is based on my original work except for quotations and citations, which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Signature : ____ylc__________

Name : YULITA HANUM BINTI P ISKANDAR

Date : 10th July 2006

ACKNOWLEDGEMENTS

Alhamdulillah be to ALLAH, the Most Gracious and Most Merciful.

One of the greatest pleasures of writing thesis is acknowledging the efforts of many people whose names may not appear on the cover, but without those hardworking, cooperation, friendship and understanding, producing this thesis could have been impossible.

Personally, I would like to express my sincere gratitude to my supervisor, Associate Professor Dr. Abas Md Said for his tireless effort in guiding and giving me valuable advice throughout this project. His seemingly endless energy and dedication to academic excellence have been a true source of inspiration to me. He also gave me unfailing encouragement and support in my learning process. Based on his experience, he had made constructive comments and suggestions to highlight my weak points in helping me in completing my projects and improved my communication skills. Without his constant guidance and strong support, I would not have produced this thesis and achieved as a researcher as well as a scholar.

A token of appreciation goes to Mr. Mohamed Nordin Zakaria, for valuable information and guidance in collecting and analyzing resources, without him it will be hard to make this project a reality.

Thank you to Dr. Kamil Mahmud, Head of Information & Communication Technology (ICT) Programme for his help in providing the facilities and equipments needed for the successful completion of my research. I wish to thank the staff of Postgraduate Studies Programme for assisting and helping me in time of need and support.

I would like to thank Dr. Shaikh Malik for reading and editing my thesis in such a short notice.

I acknowledge the perceptive comments by all colleagues and friends especially Azwandy Mohamad, Salisa Abdul Rahman and Zuraini Samsuddin who collectively create a work culture that support and nurture my learning process and personal development.

Finally yet importantly, my highest respect and gratitude is to my parents who had spared efforts and hardship to provide me with all possible opportunities for a good education during the past few years that I have been so absorbed with my studies. Also, to all people who involved directly or indirectly. Thank you.

ABSTRACT

Overall latency is the elapsed time from input of human motion to the immediate response of the input in the display. Apparently, latency is one of the most frequently cited shortcomings of current Virtual Reality (VR) applications. To compensate latency, previous prediction mechanisms insert a complex mathematical algorithm, which may not be appropriate for complex virtual training applications. More complex VR simulations most likely will impose greater computation burdens and resulted in the increase of latencies.

In order to overcome latency problem, this research is an attempt to suggest a new prediction algorithm based on heuristic that could be used to develop a more effective and general system for virtual training applications. The heuristic-based predictor provides a platform to utilize the heuristic power of human along with the algorithmic power, geometry accuracy of motion-planning programs and biomechanical laws of human. Heuristic algorithm is an important module widely used for humanoid robots and avatars in VR systems. However, to the best of the researcher's knowledge, the heuristic approach has not been used as a single prediction algorithm for compensating latency in virtual training systems.

In order to find out whether the new prediction algorithm is acceptable and possibly could reduce latency, a fast synchronization squash-game simulation was selected as a study source. This research analyzed the latencies of all subcomponents of this system and designed prediction algorithm that allows high-speed interaction.

In measuring the performance on various prediction methods, this research also makes a comparison in real tasks among 1) the heuristic-based prediction, 2) the Grey system prediction and 3) the one without prediction using different sample rates. Findings indicated that heuristic-based algorithm is an accurate prediction method to compensate latency in virtual training. Apparently, heuristic-based prediction and Grey system prediction are significantly better than the one without prediction. When heuristic-based prediction and Grey system prediction were compared, heuristic-based prediction was in fact a better predictor. Overall findings indicated that heuristic-based prediction is efficient, robust and easier to implement.

ABSTRAK

Penyelengahan merupakan antara faktor yang memberi impak negatif kepada aplikasi realiti maya. Sistem penyelengahan ini didefinasikan sebagai kekangan masa yang diambil oleh skrin komputer untuk mempaparkan pergerakan yang pantas. Bagi mengatasi masalah ini, para penyelidik menggunakan kaedah jangkaan. Satu masalah dengan kaedah jangkaan ialah ia mengandungi konsep matematik kompleks yang kurang sesuai digunakan bersama aplikasi latihan maya yang kebetulan juga kompleks. Hal ini akan menyebabkan simulasi realiti maya yang kompleks akan menambahkan beban pengiraan dan secara langsung meningkatkan tahap penyelengahan kepada sesuatu aplikasi.

Kajian ini mencadangkan satu kaedah jangkaan baru untuk mengatasi masalah penyelengahan yang dikenali sebagai heuristik diharapkan dapat membina aplikasi latihan maya yang lebih efektif. Kaedah heuristik ini adalah gabungan tabii manusia, algoritma, pelan geometri gerakan yang tepat, dan biomekanika manusia. Sungguhpun kaedah heuristik algoritma kerap digunakan dalam aplikasi realiti maya, tetapi kaedah berkenaan belum pernah diaplikasikan untuk mengatasi masalah penyelengahan.

Kajian ini menggunakan simulasi squasy, iaitu sukan yang melibatkan pergerakan yang pantas, sebagai dasar penyelidikan bagi tujuan mencuba keberkesanan kaedah heuristik. Seterusnya kajian ini menganalisa penyelengahan yang terdapat dalam setiap sub-sub komponen sistem sekaligus mencorak algoritma jangkaan bagi interaksi yang pantas.

Perbandingan keupayaan tiga kaedah jangkaan iaitu: 1) kaedah heuristik, 2) kaedah "Grey" dan 3) kaedah tanpa sebarang jangkaan dijalankan. Perbandingan ini dibuat bagi kadar sampel yang berbeza melalui tugas nyata (real tasks). Dapatan kajian ini telah membuktikan kaedah heuristik adalah kaedah yang tepat untuk mengatasi masalah penyelengahan. Kajian ini juga membuktikan kaedah heuristik dan kaedah "Grey" adalah lebih baik daripada kaedah tanpa sebarang jangkaan. Dapatan kajian juga telah menunjukkan keupayaan kaedah heuristik meningkat berbanding kaedah "Grey". Berdasarkan kajian ini, secara keseluruhan kaedah heuristik lebih efektif, lebih tegap dan lebih mudah untuk diimplementasikan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AI	: Artificial Intelligence
AMD	: Advanced Micro Devices
CAD	: Computer Aided Design
DC	: Direct Current
DESP	: Double Exponential Smoothing
EKF	: Extended Kalman Filter
GUI	: Graphical User Interface
HMD	: Head-Mounted Display
KF	: Kalman Filter
MMAE	: Multiple Model Adaptive Estimation
O-O	: Object Oriented
RAM	: Random Access Memory
UKF	: Unscented Kalman Filter
UML	: Unified Modeling Language
VA	: Virtual Assistant
VE	: Virtual Environment
VR	: Virtual Reality

GLOSSARY*

Bezier space curve	: A parametric curve generated by computing the curve's coordinates from control points weighted by terms from the Bernstein basis function.
Bounding box	: A minimum bounding volume in the form of rectangular box such that it completely contains an object.
B-spline space curve	: A parametric curves whose shape is determined by a series of control points, whose influence is determined by basis functions.
Frame rate	: The rate at which a process is modified.
Hz	: The unit of measurement to represent cycles per second.
Latency	: The time delay (lag) between activating a process and its termination.
Sample rate	: The frequency with which display is refreshed in Hz.
Stereoscopic	: A stereoscopic system provides two images with parallax information.
Tracking	: The action of locating an object's 3D position and orientation.
Virtual Environment (VE)	: A 3D data set describing an environment based upon real world or abstract objects and data.
Virtual Reality (VR)	: A generic term associated with computer systems that create a real-time visual/audio/haptic experience.

* Glossary from:
Vince, J. (1995). *Virtual Reality Systems*. England: Pearson Education Limited.

CHAPTER ONE
INTRODUCTION

1.0 Background

An authority in Virtual Reality (VR) -- Dr. Ivan Sutherland in 1968 has made a groundbreaking contribution to computer graphics and immerses interaction (Sutherland, 1969). By wearing a head-mounted display (HMD) on which wire-frame graphics were displayed, users perceived that they were occupying the same space as virtual objects. The goal of VR is to place the user in a three dimensional environment that can be directly manipulated, so the user perceives interaction with the environment rather than the computer.

Virtual environment (VE) can be emulated by using two types of VR System, which are: 1) Immersive and 2) Non-immersive (Desktop) VR (Boschker & Mulder, 2004).

In Immersive VR system user gets the feeling of being in real environment. The user can hear, touch, visualize, smell and taste with the contents of the VE by means of variety of hardware like tracker, HMD, Cyber glove, Computer Assisted VE (CAVE), Binocular Omni-Orientation Monitor (BOOM) etc (Boschker & Mulder, 2004). However, high cost and computational technology needed are two main drawbacks of this system.

Non-Immersive (Desktop) VR is a computer-generated emulation of three-dimensional environment and shows a perspective display, which can be interacted with a mouse, keyboard or 3D controller, such as the Space ball. (Boschker & Mulder, 2004). Desktop VR systems, as the name implies, are based on standard desktop computers. With Desktop VR a computer screen is normally used as the display medium. The user views the VE on the computer screen. In order to experience the VE, the user must look at the screen the whole time. 3D applications do not require CAVE-like, fully immersive environments. Often, fish tank, desktop, or dexterous types of environments are sufficient (Deering, 1996; Poston & Serra, 1996; Ware &

Franck, 1996). Compared to other VR systems, the benefits of desktop VR systems are high display resolution, eliminates complex simulation techniques and relatively low cost. Desktop VR systems are usually based on an image being displayed on a Cathode-Ray Tube (CRT) monitor and a user wearing some form of shutter glasses. This research was conducted in a Desktop VR environment using the keyboard as a real-time tracking prototype.

A lot of attention has been paid to VR in the last few years (Allison et al., 2001). The development of virtual training is a prevalent and growing phenomenon and perhaps garners a unanimous support (Fan & Xiong, 2003). Training has been considered to be one of the most natural applications areas of VR. The main reason for this is the strong need for interaction with computers and artificial worlds, in the way human beings interact with the real world. VE's do not appreciably differ from previous generations of training simulators except that the kinds of interactions are expanded through a variety of input devices and the proximity of interaction between self and objects drawn closer to the user.

To date, most VR-based training systems are situation based, where the trainees are trained for discrete decision making in special situations presented by the VR environments (Choi & Ko, 2000). The trainee is evaluated on whether or not the choices of actions are correct, rather than how the actions are performed kinesthetically. In contrast, this research discusses the application of VR to a different class of training, which is often required in sports and the arts (such as golf swing, dancing, martial arts and calligraphy).

VR has great potential for enhancing sports and fitness by creating realistic simulations and enhancing the experience of indoor exercise (Mokka et al., 2003). Sports had always fascinated humanity and also sports exercises are good examples of the motion training of dynamic movements (Ishii et al., 1999; Jie et al., 2004). Sports simulations can take many forms depending on the purpose of the simulation. Researchers are targeting many sport arts such as soccer, basketball, racquetball, ping-pong, ice hockey and also dart (Mueller et al., 2003). One might participate in this form of simulated sport to practice for improvement, to develop coordination, to develop a mental understanding of games strategies, to engage in fitness or just simply to entertain oneself. Furthermore, these highly interactive games challenge physical coordination, tactical decision-making and cooperation between players.

This research focuses on the realization of a squash simulation because squash involves fastest reaction game that demands quick responses and lightning reflexes of the players (Vuckovic et al., 2005). Besides that, the motion of the squash player is specific to squash, due to closed and small space in which the game is played and the bouncing of the squash ball from all four walls of the court (Vuckovic et al., 2005). As most of the racquet sports, squash has well defined rules and long record of research focuses on player movement.

1.1 Overview of Problem

In VR applications, one of the major issues is to provide an immersive environment, which is computer generated with realistic appearance, behavior and interaction. However, the requirement of such an ultimate display is not easily met. One of the critical problems is the perceived latency or lag (Adelstein et al., 2003; Kalawsky, 1993; Kyger & Maybeck, 1998; Wu & Ouhyoung, 2000).

Human users of virtual training are disturbed by system latency, which limits its usefulness in real world applications (Adelstein et al., 2003; Kalawsky, 1993; Kyger & Maybeck, 1998; LaViola, 2003a; Rhijn et al., 2005; Wu & Ouhyoung, 2000). Latency, in this context, refers to the delay that occurs between the movement of the user within a VEs and the result of that action being reflected by the VE. The computer needs time to read the tracker measurements, set the new camera position and perform rendering. Because of this, the picture is presented with some delay, which makes some "especially fast" tasks harder to perform.

System latency and its visible consequences are fundamental VE deficiencies that can hamper user perception and performance (Adelstein et al., 2003; Kalawsky, 1993; Kyger & Maybeck, 1998; Wu & Ouhyoung, 2000). For example, latency in estimating head orientation of only a few milliseconds can at best be distracting and at worst lead to cyber-sickness. At the same time, the illusion of a virtual world is destroyed if the objects on the screen jitter significantly while the head is not in motion – "swimming effect" mentioned by Brooks et al. (1990). Similarly, latency in other parts of a human body can make the system difficult to interact with and used effectively.

The excessive latency has been known to degrade manual performance, forcing users to slow down to preserve manipulative stability and ultimately driving them to adopt "move and wait" strategies (Adelstein et al., 2003; Ellis et al., 1999; Rhijn et al., 2005 Sheridan & Ferrell, 1963; Sheridan, 1999, Smith & Smith, 1962; Smith et al., 1962). This problem has proven to be the most limiting when trying to use virtual training for real-world applications since it is a major source in reducing the speed of target selection. Moreover, Ellis et al. (1999) showed that system latency is highly correlated with performance in a VR. Ware and Balakrishnan (1994) showed that the mean time for static target acquisition is directly proportional to latency. For example, in virtual training, when a trainee moves to an actual system with different latency characteristics, some of the techniques and strategies developed in the VE may not successfully transfer, making the VE training less successful.

Moreover, the effectiveness of training in VE is extremely dependent on the realism of the system (Kyger & Maybeck, 1998; Mokka et al., 2003; Bideou et al, 2004). In order to push the realism of these training simulations, it should combine three-dimensional design, sound and high resolution graphics at relatively smooth rendering rates. Realism in this case, is directly linked to the user's reaction itself. According to Mokka et al. (2003), only when the user "believes" or "immerse" his environment, he forget that all surrounding objects are virtual, then the user takes the environment too seriously that leads to full body activities and may sweat as he does in a real sport.

1.2 Motivation of Research

To compensate for the latency, many proposed method used prediction (Azuma & Bishop, 1994; Garrett, 2003; Kiruluta et al., 1997; LaViola, 2003a; Mazuryk & Gervautz; 1995; Rhijn et al., 2005; Wu & Ouhyoung, 2000). According to LaViola (2003c), designing and developing suitable prediction is an extremely difficult and challenging task. Certain algorithms may perform better or worse depending on the underlying tracking system's sampling rate, noise variance, types of user motion and tracking technology (e.g., magnetic, mechanical, inertial, acoustic and optical).

Various prediction algorithms have been proposed (Azuma & Bishop, 1994; Garrett, 2003; Kiruluta et al., 1997; LaViola, 2003a; Mazuryk & Gervautz; 1995; Rhijn et al., 2005; Wu & Ouhyoung, 2000). Detailed descriptions on each prediction will be discussed in Chapter Two. However, each of the prediction algorithms has their own limitations. The prediction algorithm inserts a complex mathematical algorithm, which may not be appropriate for a complex virtual training application. More complex VR simulations most likely may impose greater computation burdens and resulted in the increase of latencies drastically (Yung et al., 2000). Furthermore, according to LaViola (2003b), almost all prediction algorithms contain one or more parameters that are used for tuning to optimize performance. Therefore, a significant aspect in determining what prediction algorithms to use is in adjusting an algorithm's parameter values. These adjustments are nontrivial in the sense that an optimal parameter setting for one type of user motion may not be optimal for another.

Besides that, most of the previous work only compares the proposed method to doing no prediction at all and how to choose a suitable prediction algorithm for a given VE is a problem that has not been addressed much (LaViola, 2003b; Rhijn et al., 2005).

The majority of predictive compensation works for virtual training applications have focused on Kalman filter-based techniques, either for their primary prediction simulation (Chua et al., 2003; Rusdorf & Brunnet, 2005) or as a secondary implementation against which another technique is compared (Wu & Ouhyoung, 1995, 2000). Wu & Ouhyoung presented a comparison of Kalman filter and Grey system in 1995. Both the algorithms mentioned give reasonable results in the prediction. However, the computation complexity of Kalman filter is relatively high compared to the Grey system.

Although study has shown that the computation of Grey system is relatively low, however the algorithm still requires the calculation of partial derivatives, matrix multiplication, inverse and a more complex infrastructure thus causing implementation difficulties. The Grey system algorithm's parameters were tuned using a limited amount of motion datasets, optimizing their performance to motion data with similar characteristics. However, if the prediction needs to be applied to other types of motion, these tuned parameters may not yield accurate results.

Here lies the hub. After analyzing literature, this research proposed a prediction method based on heuristic that could be used to develop a more effective and general system for virtual training applications. According to Perkins (1981), heuristic is:

> *"A rule of thumb that often helps in solving a certain class of problems but makes no guarantees."*

Although, heuristic algorithm was an important module and applied to a wide range of humanoid robots and avatars in VR systems (Mas & Thalman, 1994; Noser & Thalman, 1997; Tsuji et al., 2004; Zordan, 2002), to the best of the researcher's knowledge, the suggested approach has not been tried yet as a single prediction algorithm for compensating latency in virtual training application. The researcher introduces a motion control paradigm that simplifies the generation of expressive movements by proceduralizing the qualitative aspects of movement. There are five fundamental principles on movement (Price, 1995):

1. Movement is a process of change.
2. The change is patterned and orderly.
3. Human movement is intentional.
4. The basic elements of human movement may be articulated and studied.

5. Movement must be approached at multiple levels if it is to be properly understood.

The movement's principles can be considered as heuristic. This is because human movement can be patterned and orderly which is also known as the rule of thumb. This is one of the reason, the researcher use motion prediction as a prediction algorithm. The heuristic algorithm implemented in this study focus on the habitual motion of the squash game.

Heuristic is sort of like a rule of thumb. Sport is a game of habits and provides the strategic thinking and planning skills. This is a basic habit that gives a player the basic control. The squash game has certain rules, tactics and pattern of movements that makes it an ideal subject for the research of heuristic.

It has been suggested that individuals may use heuristic of reasoning and rely on schemata when a quick decision is necessary. Accordingly, it is possible that decisions made by a player may be influenced by prior knowledge. Squash is a type of game with fastest reaction movements that demand quick responses and lightning reflexes as well as coordination, concentration and anticipation (Collins et al., 1990; Mckenzie, 1996, 1998; Wong & Beng, 1995; Vuckovic et al., 2005). Thus heuristic is indirectly involved in squash application concerned with the automation of intelligent behavior.

Heuristic, which was widely used in the literature, means to improve the efficiency of the subject's performance. Many heuristic planners have been created in an attempt to achieve the necessary speeds in off-line (or more ambitiously, on-line) processing (Wan et al., 2003). It is an advent of modern technique, which requires virtual training designers to develop programs that would simulate human behavior. For many applications, a specialized heuristic motion planner is often sufficient and a computationally expensive complete planner may not be needed.

Therefore, in this research, the researcher explores the benefits of heuristic over the previous predictions in human estimation. Drawing on this research, this study explores how and what the subsequent outcomes are.

1.3 Research Objectives and Scope of Study

The purpose of this research was to determine the feasibility of using heuristic-based prediction to predict motion movements in virtual training application. This is accomplished primarily with the report, but also practically aided by the developed software.

The main objectives in this research are:

1) **To develop heuristic-based prediction to minimize latency problem for dynamic motion in virtual training application.**

Movement is often described in terms of actions or what one does (Hodgins et al., 1998). However, the researcher interested in how one moves. This heuristic-based prediction algorithm should therefore be based on previous behavior patterns and applied to speculate on the future behavior of a human. A brief explanation of heuristic algorithm technique is presented in Chapter Three.

Any tracking system to be used in a VR environment has to be accurate, efficient, robust and simple to implement (LaViola, 2003c). These aspects are even more critical; any tracking deficiencies will result in unrealistically behaving objects as discussed in Section 1.1. Therefore, the first objective of this research is to develop heuristic-based prediction, which is required to be accurate, efficient, robust and simple to understand and implement. Accurate prediction is important since the researcher want to mask latency and keep images fresh. Fast prediction is an important requirement since the researcher wants to minimize any additional latency introduced into the VE. In addition, the suggested prediction algorithm do not has to predict as far into the future as slower ones to compensate for any computation overhead. Robustness is important for heuristic-based prediction to be useful in a VE, as the prediction needs to handle a variety of different motion dynamics and styles in different applications. Finally, the heuristic-based prediction should be simple to understand and implement because the prediction will be easily used in VE systems and applications.

2) To develop squash simulation which will be integrated with the prediction algorithm.

Sports training are good potential examples of the dynamic movements training. In order to find out whether the newly prediction algorithm is acceptable and could fulfill the main purpose of reducing latency in virtual training application; this research has chosen squash as a study source. In general, virtual sports are very difficult to interact in comparison with other virtual tasks because the motion speed of players is very fast and the motion area is quite large. For realization of squash simulation, the researcher needs prediction with high speed. Furthermore, a high-speed visual system with stereovision is necessary so as to display a moving player with high speed. The design and development of squash simulation will create a physically interactive game that would be both fun and useful as a virtual form of sport applications training.

Squash is usually played by two people but it can be played by four (Mckenzie, 1996; Collins et al., 1990). In this research, the researcher focused on the case of singles version of player. Squash is a type of sport game with different number of games. Each of this game is made up with number of a rallies (Mckenzie, 1996, 1998; Wong & Beng, 1995). The rallies can be characterized by constant player motion and therefore much more interesting from the viewpoint of motion analysis.

3) To experimentally validate the performance of heuristic-based prediction and demonstrate it's utility as a prediction algorithm.

The third objective was to evaluate the performance of the researcher proposed prediction with the Grey system prediction. This analysis will be a proof of the capabilities of the algorithm to show the ways in which the performance algorithms could be used. The experiments are described in Chapter Five and the results are presented in Chapter Six.

1.4 Research Hypotheses

Concluding from Section 1.2 and 1.3, the researcher formulates research hypotheses so as to examine whether:

1. Heuristic prediction is better in reducing latency than the Grey system prediction.
2. Heuristic prediction is better in reducing latency than the non-prediction.

To prove the validity of the hypothesis that was formulated by the researcher, comparison experiment between: 1) heuristic-based prediction, 2) Grey system prediction and 3) non-prediction will be conducted. Since Grey system prediction is the superior replica-table prediction available, it serves as a benchmark. The heuristic-based prediction must outperform Grey system prediction to justify its greater complexity and computational demand. A second benchmark of performance is the case of no prediction in the loop at all, representative of many current implementations of virtual training simulators. Thus heuristic-based prediction performance can be bracketed by good and unobtainable prediction and no prediction, with a direct comparison to the best results of a prediction (i.e., Grey system).

Because of this, the experiment is interested in the performance of players in both running time and numerical accuracy using the following three (3) methods:

1) Non-prediction,
2) Prediction with the Grey system prediction and
3) Prediction with the heuristic-based prediction.

1.5 Thesis Overview

This thesis is divided into seven chapters. This chapter introduces the background ideas that constitute the foundation of this research. It describes the aim of the research together with the methods used. The main purpose has been to provide information about how and why this research has been conducted.

Chapter Two contains an overall background discussion introducing the concepts and technology that forms the core of this research. This overview provides a general discussion of related research, and each chapter then includes other more specific background information where relevant.

Drawing upon the issues highlighted in Chapter One and Chapter Two, Chapter Three outlines the basis for the implementation of the current and suggested prediction algorithm. The general research methodology employed and the argument for the specific choice of research methods used to address the research's problem in this research is presented.

Chapter Four presents designs and developments of the actual squash simulation.

Chapter Five discussed on user acceptance study and experimental study conducted during the research.

Chapter Six is dedicated to an in-depth discussion and analysis of the empirical findings. Apart from highlighting the main results of the study, it will also examine the empirical findings in the light of the current literature.

The final chapter, Chapter Seven reflects on the journey of this research and draws attention to the theoretical, methodological and practical contributions of this thesis. In addition to identifying these contributions, the researcher also addresses the limitations of the study and suggests directions for further research.

CHAPTER TWO
LITERATURE REVIEW

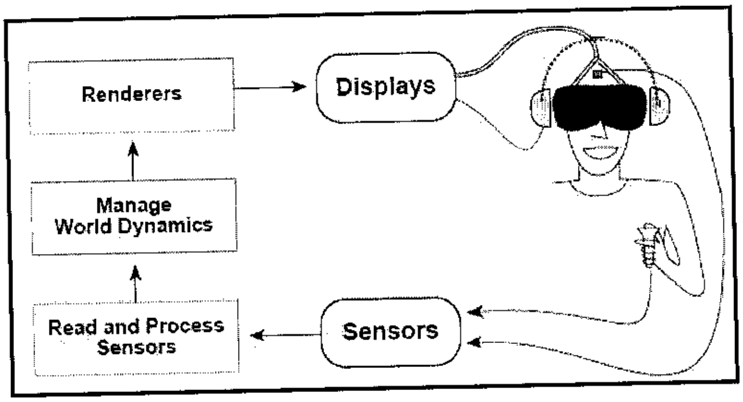
2.0 Overview

In this chapter, the researcher reviews prior work in the field of VR from which this research is drawn upon. This chapter starts by providing an introduction to the VR technology, virtual training and latency issues. The researcher then presents a brief survey of various proposed solutions to the latency problem in the current VR literature. The researcher concludes by briefly describing solution to previous prediction algorithms, which is heuristic.

2.1 VR System

There are several basic components that are required for the creation of any VE. These elements are shown in Figure 2.1 and discussed below.

Figure 2.1: Generic VR system [1]



[1] See: http://www.cg.tuwien.ac.at/research/vr/studierstube/multidim/ [December 2005]

Displays are hardware devices used to present the environment to the user. The display that is most commonly associated with VR is the head-mounted display (HMD) (Adelstein, 2003; LaViola; 2000; Wu & Ouhyoung, 1994, 1995, 1997, 2000). HMD has two video screens mounted in a helmet or mask used to project a stereographic view to the user. The computer screen, with or without 3D capabilities, is another video display. Headphones and loudspeakers are examples of audio displays. Research is also being done to create and improve haptic, kinesthetic, and various other types of displays to be included in the VR interface.

Renderers are used to compute the sensations to be presented in the displays (Deering, 1995). Using the description and positions of objects in the world and the position of the user's body, renderers create the appropriate sensory experience and drive the associated hardware.

Sensors are input devices used for determining the user's physical behaviors. Typically, position sensors are used to determine the orientation and relative position of the user's head and hand (Isdale, 1999). Other sensors include the VPL Data glove, which measures the amount of flex in the users hand joints, the "wand," a 6D joystick which might report whether or not buttons on it are pressed and where it is pointing, the standard mouse, and 6D input devices such as the Space ball (Isdale, 1999; Deering, 1995). World dynamics provide the environment with which the user interacts. Almost all systems provide some sort of manipulation of objects in the world as opposed to only enabling the user to fly through a static model.

The system elements above must be integrated into a smooth, fast-flowing process to facilitate the VE while introducing as little latency between the sensors and displays as possible.

2.2 VR Technology

Most VR configurations include a tracking system, which is a vital aspect of a VR system (Brunner et al., 2003; Isdale, 1999; Rhijn et al., 2005). The tracking system allows the user to interact with the VE. Typically a tracking technology includes a head-mounted display, head-coupled display, 3D interactive device, headphone and 3D tracker (Boschker & Mulder, 2004; Brunner et al., 2003; Isdale, 1999; Rhijn et al., 2005). Systems for tracking and motion capture for interactive computer graphics have been explored for over 30 years (Rhijn et al., 2005). Throughout these years, industries and researchers in VR have explored mechanical, magnetic, acoustic, inertial and optical tracking technologies.

As previously identified in the literature, magnetic tracking is the most prevalent for real-time applications (Isdale, 1999). It is generally used in higher-end applications including movie and games. Possessing reasonable accuracy with little or no obstruction problems, these relatively inexpensive systems track both segment position and orientation with body-mounted sensors that measure a spatially varying magnetic field. Magnetic trackers are fairly easy to use, accurate and can have reasonable latency (Brunner et al., 2003). The most common magnetic trackers are: 1) Polhemus style trackers and the 2) Flock of Birds trackers (Brunner et al., 2003; Isdale, 1999). These devices (i.e., their interfaces) transmit six degrees-of-freedom position and orientation data to the host computer at a high rate while acting in concert with competing processes in the complete VR environment[3].

Figure 2.2: Polhemus tracker [2] (left) and Flock of Birds tracker [3] (right)

Figure 2.2 shows Polhemus tracker and Flock of Birds tracker. The Polhemus trackers have sampling rates of around 120 Hz per receiver[2]. This means that the user's position and orientation is measured 120 times every second, and the accuracy of the measurements increases with the number of measurements taken per unit time (Mine, 1993). The range of these trackers is on average around 12 feet, but it can be extended to 30 feet by using a long-range transmitter[2]. The latency of the Polhemus systems, as reported by the company averages about 45 milliseconds, which means for the user's movement to be sensed, measured and communicated to the rendering computer (Isdale, 1999; Mine, 1993). In this way, the rendering computer is always 45 milliseconds behind the actual movement of the user[3]. The main problem with the Polhemus is that any conducting materials present in the environment will affect its performance. Moreover the Polhemus has a limited working range ($\sim 1m^3$) and an update rate (~16Hz) which is barely enough for interactive applications (Isdale, 1999; Mine, 1993).

The Flock of Birds tracker, created by Ascension Technology Corporation is a modular tracker for simultaneously tracking the position and orientation of one or more receivers (targets) over a specified range $\pm$ 4feet[3]. Motions are tracked to accuracies of 0.5° and 0.07 inch at rates up to 144 Hz[3]. This means that it is more accurate that the Polhemus trackers but it has a smaller range. Due to the simultaneous tracking, fast update rates and minimal lag can occur even when multiple targets are tracked (Brunner et al., 2003; Isdale, 1999; Mine, 1993). The latency of the Flock of Birds tracker is comparable to the Polhemus trackers (Mine,

[2] See: http://www.macs.hw.ac.uk/~hamish/9ig2/topic39.html [December 2005]
[3] See: http://www.vrealities.com/flockofbirds.html [December 2005]

1993). One of the benefits of the Flock of Birds system is that it is able to accommodate a particular problem of magnetic trackers since The Flock of Birds system employs pulsed Direct Current (DC) magnetic fields to minimize the distorting effects of nearby metals. In particular, it is designed for head and hand tracking in VE games, sport analysis, simulations, animation and visualizations (Isdale, 1993).

2.3 VR-based Training

Current advancements in the 3D computer graphics technology, in particular, the latest appearance of powerful and affordable 3D graphics hardware, brings a fresh wave of interest to immersive VR applications. (Ponder et al., 2003). Immersive VR is able to provide a rich, interactive and engaging training context that in reality would be too dangerous, too expensive or simply too impossible to access (Deering, 1995; Ponder et al., 2003). In addition, performing tasks enhances the learning process providing learning by doing medium through a first-person experience.

Figure 2.3: VR-decision training system; concept and key elements (Source: Ponder et al., 2003)

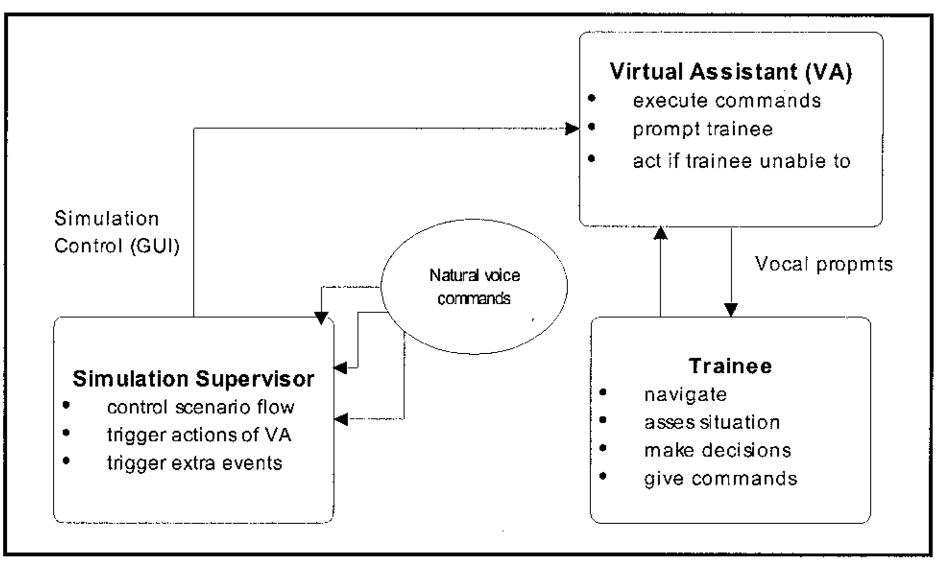


Figure 2.3 shows an immersive VR decision training system concept. This type of training can be categorized into on-the-job training and also situation-based training

(Ponder et al., 2003). The respective roles and mutual relationships between the trainee, Virtual Assistant (VA) and the simulation supervisor are presented schematically in Figure 2.3. The trainee interacts with his VA using natural voice commands and hearing respective replies and prompts from the VA. Simulation supervisor has the following tasks: direction of the scenario path, natural voice commands given by the trainee and triggering of respective actions to be executed by VA and triggering of "disturbing" virtual events.

On-job training is particularly effective in complex tasks where a great deal of independence is granted to the task performer. Unfortunately, on-job training is the most expensive method and furthermore it is limited by frequent unavailability of the training context itself (Ponder et al., 2003). In order to address the problem of on-job training, many VR solutions have been proposed in recent years. Successful cases as well as new attempts can be found in a very broad range of domains such as space and aviation (e.g. flight and space mission simulators (Lotfin & Kenny, 1995)), medicine (e.g. virtual endoscopy (Wiet et al., 1997), arthroscopic knee surgery (Gibson et al., 1996) and bone dissection surgery (Kuppersmith et al., 1997)) and military (e.g. mission training (Calvin et al., 1993), nuclear weapon disassembly (Stansfield, 1998)).

On the other end of the spectrum, VR system can help the trainee in developing his or her psychological skills required to face the reality. Such a system can be classified as situation training or in particular as decision training (Ponder et al., 2003). Using this system, a trainee is expected to lower his psychological barriers by going through a semi-real, synthetic experience (Duffy et al., 2004). According to Duffy et al. (2004), on-job training systems that stresses on "learning by doing" while a situation-based training focuses on "living through." For example firefighters training (Julien & Shaw, 2002; Perdigau et al., 2003), hostage negotiations (Ruisseau et al., 2000) and earthquakes (Tate et al., 1997).

However, this research focuses on the application of free motion training such as sports (Baek et al., 2003). The motion training can be modeled as a process of transmitting motion information from the trainer to the trainee through a series of interactions. Along this line, motion-training environments have been investigated focusing on the interactivity based on human motion and feedback type.

There are many examples of human-computer interfaces that require motion-training effort. Mueller et al. (2003) discusses social aspects of computerized sports over distance, based on a game setup with a regular soccer ball and life-size videoconference screen. Ishii et al. (1999) presented an "athletic-tangible interface," a ping-pong table that augments the sport with dynamic graphics and sound. Davis and Bobick (1998) built a virtual aerobics trainer in which the user's actions were captured and recognized as or she was instructed to "get moving" or given feedback such as "good job." In the context of martial arts and computers, Chua et al. (2003) and Yang (1999) have developed VR system to check how a trainee follows an avatar.

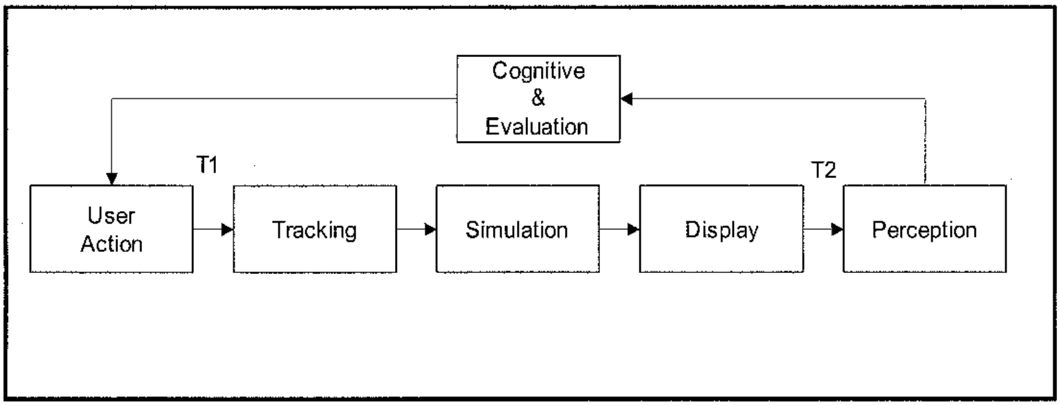
2.4 Definition of Latency

For athletes, image training should play an important role to advance their skills. For such training, applying VR techniques must be helpful for their concentration on their images.

However, in order to realize a virtual training application, researchers are faced with latency problem. High speed and low latency are probably the most important characteristics of any virtual training system (Adelstein et al., 2003; Ellis et al., 1999). Earlier research has identified latency as a critical problem for VR system. All of the latency definition found in literatures commonly defined latency as the time delay between the human input to a VE and the response of the environment (Adelstein et al., 2003, Allison et al., 2001; Azuma & Bishop, 1994; Rhijn et al., 2005; Wu & Ouhyoung, 2000). In this research, latency will be referred to as the delay between a user's action/motion and when that action is visible in the display.

Techniques used to measure latency share the requirements of capturing both a signal generated by a change in the tracker position and a signal generated by the change of the image on the display device. Ware (1994), Liang (1991) and Mine (1993) have illustrated techniques for making these measurements. Figure 2.4 shows the framework of latency as presented by Mine (1993). Based on this model, a user performs an action at time T1. Then tracking system registered actions of the user. Next, system analyzes and evaluates the data, calculates an appropriate response and updates simulation tasks. At time T2, result is fed back to the user through a display system. Latency will occur during T2-T1.

Figure 2.4: Latency framework (Source: Mine, 1993)



2.5 Sources of Latency

Major components of VE latency come in three different forms: 1) rendering latency, 2) application-dependent processing lag and 3) user input device lag.

2.5.1 Rendering Latency

Rendering latency at its most basic is the amount of time it takes for the computer screen to be updated. More explicitly, it is the time required from sending data to the rendering hardware until that data is displayed (Garrett, 2003). This rendering latency component is highly dependent on the scene complexity (for example, number of objects, number of polygons per object, background). The variance of rendering latency is generally large because of the dependence on the user perspective. The more accurate the polygon is rendered, the slower the image display becomes (Blade & Padgett, 2002). It is common for rendering hardware that makes use of double buffering and which does not use the CPU for computations to be capable of refreshing the display at 150 to 160 Hz (Blade & Padgett, 2002; Garrett, 2003). However, this causes a rendering latency of around 6.25 milliseconds (Garrett, 2003).

The stereoscopic mode is one of the key advancement that will make VR more realistic (Ware & Franck, 1996). The aim is to produce images from a location that dynamically tracks the viewing orientation of the viewers. If stereo goggles are used, then each eye's viewpoint will be refreshed at 75 or 80 Hz (Ware & Franck, 1996). This causes a rendering latency of 12.5 milliseconds (Blade & Padgett, 2002).

2.5.2 Application-dependent Latency

Application-dependent processing lag is the latency that arises from the computation of the three-dimensional model. This includes all aspects of the computer graphics such as physics, collision, rotation, interactions with other objects, file I/O, etc (Adelstein et al., 2003). This latency is entirely dependent on the complexity of the model and the speed of the computational hardware. Latency will increase dramatically when output requires substantial computation for graphic rendering. Complex VE simulation involves a lot of complicated computational burdens (Yung et al., 2000). For instance, a very complex model of a real-world object could possibly contain million of faces, all of which must be transformed by the computer during any change in the model or scene. In contrast, a very simple model consisting of a cube would have only six faces. For this reason, most VE applications are forced to use simple models in order to avoid large application dependent latencies.

2.5.3 User Input Device Latency

User input device latency is introduced from the necessary communication between the human-computer interface device and the VE application. This type of latency includes event recording, amplification, analog-to-digital conversion, communication time, etc. It is associated mostly with three-dimensional tracking devices attached to the hand or head. This latency varies with the type of tracking device used and the modes of operation of the devices (Garrett, 2003; MacKenzie & Ware, 1993; Miller et al., 2002).

At a 120Hz update rate, the tracker generates a steady burst of serial data at 8.3ms intervals (Mine, 1993; Rhijn et al., 2005). Since the optical switch can be crossed at any time within this interval, a variable latency exists range of 0-8.3ms with a uniform probability of any given latency value occurring at any given time (Mine, 1993). Attaching an electromagnetic pickup to the tracker source revealed that the tracker locates the sensor by sampling three electromagnetic bursts from the source. The interval from the middle bursts to the beginning of a serial output was measured to be 3.5ms (Mine, 1993; Garrett, 2003).

2.6 Impact of Latency

Time latency continues to be a major limitation in the VE (Ellis et al., 2000; Garrett, 2003; Rhijn et al., 2005; Wu & Ouhyoung, 2000). The types of latency discussed above are known to have an adverse affect on both user performance and comfort in VE's.

Loss of immersion (sometimes called a break in presence, or BIP) is caused when the head-mounted display does not completely block out the light from the outside world (Ellis, 1999; Garrett, 2003). A user touches an object in the real world that does not have a counterpart in the virtual world, thus leading to irregularities during movement. For instance, suppose a user wears a HMD to scan the VE from left to right. If there is latency within the VE system, the location of objects within the VE will not change at the appropriate time during the movement. In essence, the user may turn his or head, but due to latency, the world's shift is not temporally registered with the position of the head. This amount of time needs only be on the order of tens of milliseconds for the BIP to occur (LaViola, 2000).

Another important and troublesome problem with latency is the tendency for some users to exhibit symptoms that parallel symptoms of classical motion sickness both during and after the VE experience. This type of sickness, aka cybersickness, is distinct from motion sickness in that the user is often stationary but has a compelling sense of self motion through moving visual imagery (LaViola, 2000). This sickness is generally caused by some disparity between different sensory modalities within the body. The inner ear sends a signal to the brain that indicates that the head is rotating, while the visual system does not perceive the corresponding rotation of the world (in the opposite direction) (LaViola, 2000; Garrett, 2003). In a HMD system, if the latency is relatively large (over 100ms for instance) it will cause dizziness (Ellis, 2000). At the same time, the illusion of a virtual world is destroyed if the objects on the screen jitter. Significantly while the head is not in motion, the "swimming effect" mentioned by Brooks et al. (1990) will occur. There are a number of symptoms that can occur due to cybersickness and motion sickness which includes: eye strain, headache, pallor, sweating, dryness of mouth, fullness of stomach, disorientation, nausea and vomiting (LaViola, 2000; Garrett, 2003).

Previous research has shown that even low latency can have adverse effects on performance in VE. Latency to other parts of a human body (such as wrist and hands) can make the system difficult to interact with, use effectively and reduces user speed (Azuma & Bishop, 1994). Recent work by Ellis et al. (1999) is representative of the group's work investigating the impact of latency and variations in latency on user performance. They have examined overall user performance in VE and it has also shown that latency appeared to be the stronger, more statistically reliable factor in degrading the performance. If the VE is being used for accomplishing time-critical or life-critical tasks, such adaptations are not adequate (Garrett, 2003).

2.7 Past Approaches to Latency Reduction

VE effectiveness is often measured in terms of application success, e.g. task performance of training transfer, or is measured by the degree to which the VE creates a subjective illusion of presence-a sense of "being there" in the virtual world (Azuma & Bishop, 1994; Garrett, 2003; Kiruluta et al., 1997; Miners, 2001). The effectiveness of VE can be evaluated by the extent to which the system minimizes or eliminates factors that hinder user performance and/or break the user's sense of presence, e.g. latency. Many researchers have examined latency and have tried to reduce its detrimental effects. Previous efforts can be categorized as bounding latency, reducing latency, compensating for latency and achieving registration despite latency (Kiruluta et al., 1997). However none of them has been entirely successful.

The method for reducing the latency depends on the hardware tracking sensors (Mine, 1993). However, it can be also determined by the software generating the virtual world. Most current approaches for VR focuses on polygon rendering, which involves approximating the shape of each object with various numbers of polygons. This algorithm can give real-time performance for only simple virtual worlds; to handle richer worlds, improved algorithms must come from somewhere else. Use of Fitts' Law (Mackenzie & Ware, 1993) has a good potential but the application of this method is quite limited. The Fitts' Law noted that latency is very important in 3D applications; therefore experiment should be presented in 2D applications.

"Time-critical" computation is essential for VR systems to give real-time performance (Wu & Ouhyoung, 1995). It is a technique in which quality is traded for speed. This application is constantly aware of time. Hence, the most obvious way to reduce latency is to increase the speed of required computation (Kyger & Maybeck, 1998, Wu & Ouhyoung, 2000). To reduce the latency through system hardware and software reorganization, Jacoby et al. (1996) improve temporal response significantly by minimizing the round trip delay. However hardware improvements alone cannot solve the problem. This is because computation, sensor and display processing take finite time to execute and it contains minimum latency even if approachable (Ellis et al., 2000). For example, for a commercial tracking system, displaying from an "old" LCD screen, there is still about 170ms of constant refreshing delay, even though the system round trip delay has been fully eliminated (Wu & Ouhyoung, 2000). Moreover, to become more and more realistic in visualization, the complexity of the rendered environment will increase dramatically and the latency will increase accordingly. Faster computer may reduce latency, but they will not eliminate the problem (Kyger & Maybeck, 1998).

The best results up to now can be achieved by the prediction method (LaViola, 2003b). Very often in any application, motion position has to be predicted well in advance in order to overcome delays that are introduced by the image generating system. Without prediction, VR must use the current pose to compute and display new images for each frame. This naive approach can cause problems since the user might be moving during the computation and display process, resulting in stale images and a display-to-user-motion synchronization mismatch.

Figure 2.5 shows prediction framework in reducing latency. Prediction based systems use signals from 3D trackers as the inputs, process them, and output the predicted data. It takes place in the interaction cycle between tracking and evaluation stages. As latency occurs during T2-T1, T_Predict should be ideally equal to the latency of the system (T2-T1). This information would allow the computer that generates image to complete all necessary operations and store the appropriate graphics in a buffer before it is actually needed to render the scene at the predicted position, and to finish the process by the time user finish their tasks. (Kyger & Maybeck, 1998).

Figure 2.5: Prediction framework (Source: Mine, 1993)

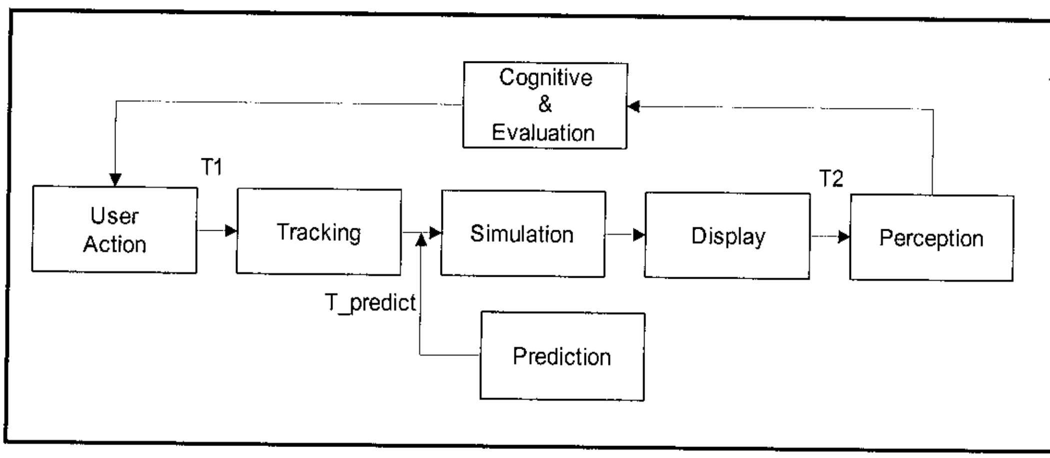


Figure 2.6: Category of prediction algorithms

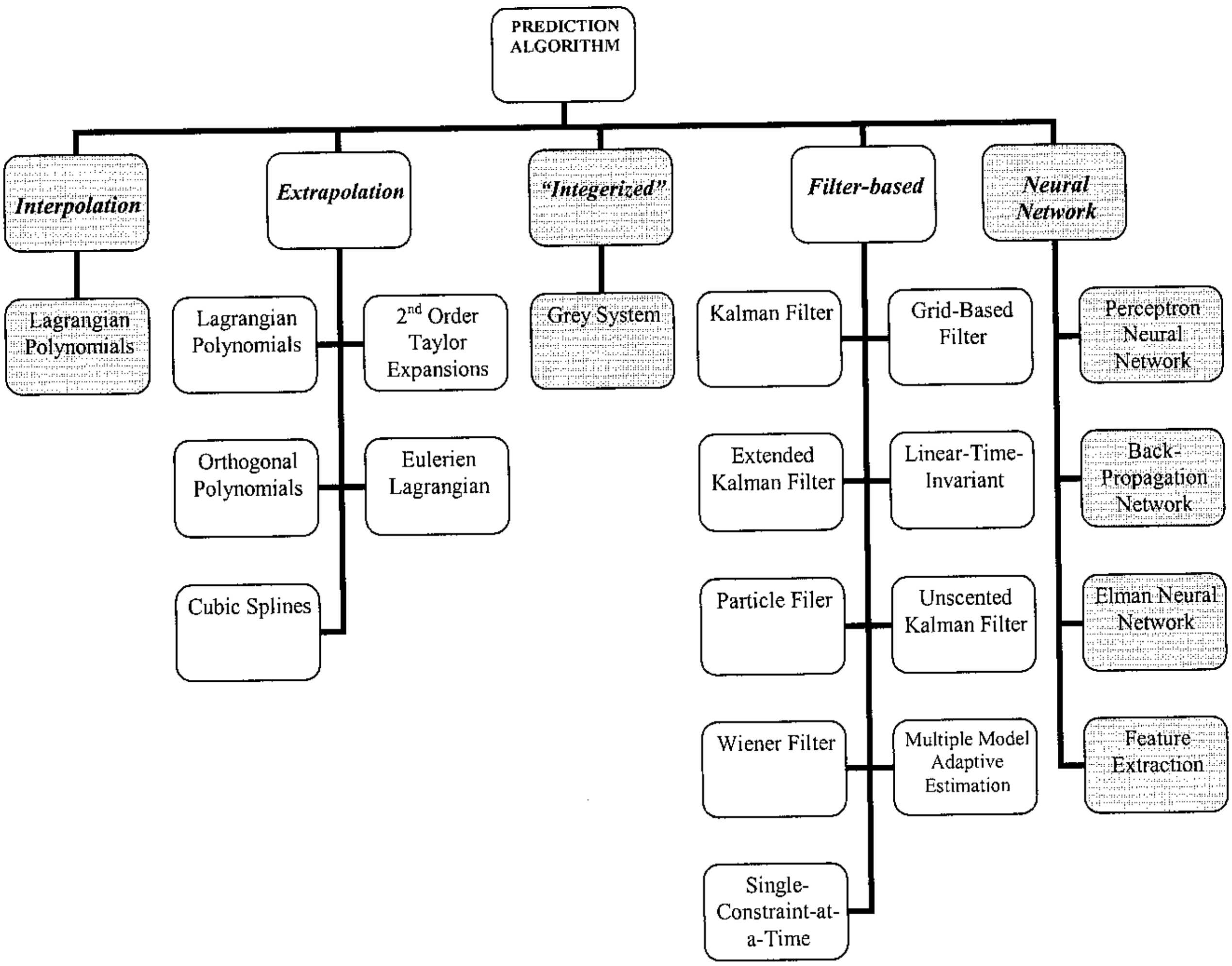


Figure 2.6 shows the different categories of prediction algorithm. Those predictions have demonstrated the ability to track human movement in VR environment. Brief description of each algorithm will be given in the next section.

2.7.1 Interpolation and Extrapolation

A common theme with the simple interpolation and extrapolation routines is that they all are based on approximating a function with polynomials using the previous *n* user poses from evolving motion sequence, and then using that approximating function to extrapolate the user's future location (Garrett, 2003; LaViola, 2003b). Currently these are four simple extrapolation routines available based on LaGrange polynomials, cubic spines, least square approximation using orthogonal polynomials and 2nd order Taylor expansions. The different prediction in this category all use different function approximation schemes.

Prediction mechanism is often no more than a crude averaging and extrapolation from the current movement in time. For example, extrapolation can be accomplished by using a Taylor polynomial approximation of the angular position curve. Such an approximation would make use of the following equation (Garret, 2003):

$$p(t_0 + n) = p(t_0) + np'(t_0) + \frac{1}{2}n(n-1)p''(t_0) \qquad (1)$$

Where t_0 is the current time, n is the number of milliseconds beyond which it will be extrapolated and $p(t)$ is the angular position at time t (Garret, 2003).

While such an algorithm can be useful and effective in certain situations, it manifests serious problem if the user moves quickly or change velocity abruptly (Allison, 2001). Even though there was compensation for most movements, the user would still experience latency effects when performing quick and abrupt direction changes. As a consequence users are forced to make only slow, deliberate motions in order to maintain the illusion of reality. Also, by using these algorithms, producing predictions of a useful range (i.e., 20-40ms) are subject to very large errors (Garrett, 2003). This is because the interpolation and extrapolation approach amplifies the noise in the original signal, especially when the temporal extrapolation distance is high.

2.7.2 "Integerized"Prediction"

Under "integerized" category, future poses are forecasts integer multiplies of the time between samples Δt (LaViola, 2003b). Such a prediction scheme presents no difficulties when predicting $i\Delta t$ steps into the future. However, modifications to the algorithms are required if i is not an integer. For general applicability, the predictions in this category are modified to handle any prediction time i by predicting $\lfloor i \rfloor \Delta t$ and

$[i]\Delta t$ steps into the future and then using extrapolation to predict user pose at the exact time.

In 1980, Professor Deng Ju-long introduced the Grey system theory which been successfully used in many fields (Deng, 1989). It was a new theory that worked on unascertained systems with partially known and partially unknown information by drawing out valuable information by generating and developing the known information (Lin & Liu, 2004). In Grey system, the behavior of information are neither deterministic (white) nor totally unknown (black), but (partially known defined as grey).

In 1994, Wu and Ouhyoung proposed prediction algorithm based on Grey system theory (Wu & Ouhyoung, 1994, 1995, 2000). This prediction is applied to the prediction of a tracker motion because the behavior of the tracker output is "grey." In real world, the behaviors of most systems are uncertain. The effects of other systems to a system under monitoring are also unclear. Thus, it is hard to exactly analyze and predict the behaviors of such a system.

Figure 2.7: A general system diagram of a Grey system prediction (Source: Wu et al., 1997)

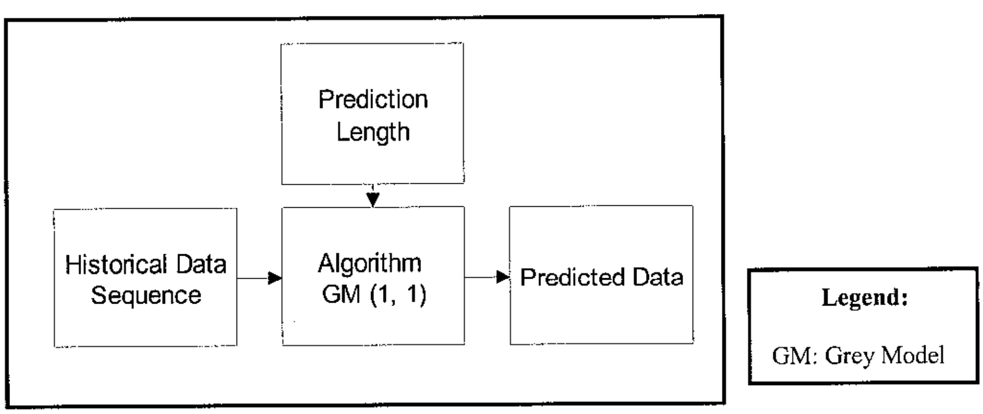
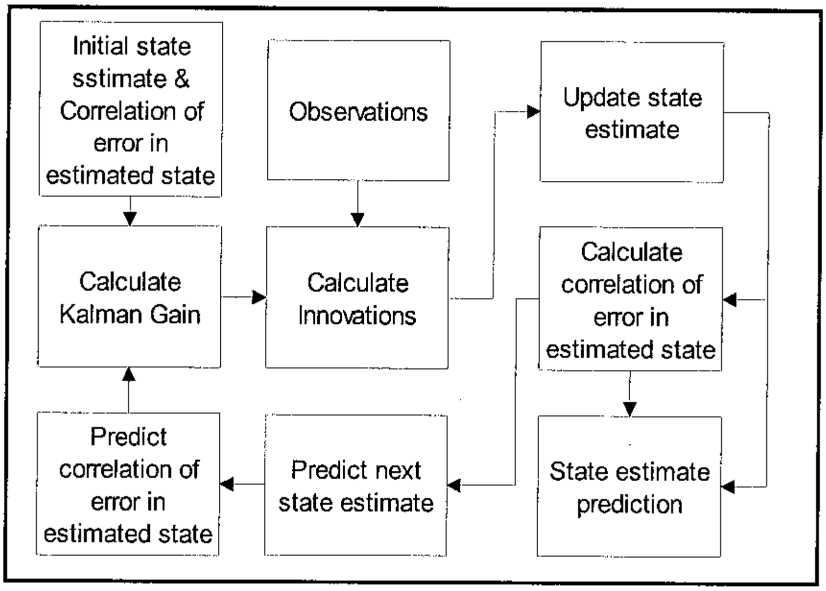


Figure 2.7 shows the system diagram of a Grey system prediction. System behavior formula can be generated by applying historical data sequence to the prediction algorithm (GM (1, 1)). The number of historical data will used as inputs. By applying

a specified prediction length to the system behavior formula, a new predicted data is formulated.

2.7.3 Filter-based Prediction

Figure 2.8: Filter Summary (Source: Miners, 2001)



Liang, Shaw and Green (1991) have developed a predictive head tracking algorithm based on Kalman filter (KF) that based on the observation that latency is mainly due to the delay in orientation data. The term filtering represents an optimal recursive (repetitive with time) procedure, which enables the estimation of signals to be determined from successive measurement in time (Miners, 2001). It is optimal because it incorporates any information that is provided to it and it is recursive because it does not need to keep all of the previously processed data in memory at each time step. Generally, these filters use an underlying process model to make an estimate of the current state of the system. The estimation correction is made using any tracker measurements available. Then, after the correction is made, the process model is used to make a prediction. These processes are summarized in Figure 2.8.

Filter-based predictions have received considerable attention in the literature and appear to be the prediction method of choice by many researchers (Azuma & Bishop, 1994; Friedmann et al., 1992; Kiruluta et al., 1997; Kyger & Maybeck, 1998; LaViola, 2003a, 2003b, 2003c). However this algorithm still has severe deficiencies. Predictive filtering introduces complex changes in the image motion and the dynamics of remaining the image slip are strongly dependent on the prediction (Azuma & Bishop, 1994; LaViola, 2003a). Other filter-based predictions, which had been developed and studied, have similar restrictions on system parameters and user motions either by the refresh/sampling rate and often come in the form of noisy movement, which would normally inhibit the effectiveness of a prediction-corrector system (Mazuryk & Gervautz, 1995; Garrett, 2003; LaViola, 2003b).

Friedmann et al. (1992) used filter-based predictions for hand motion, but their studies were limited (i.e., drumming) and were not thoroughly tested across different system parameters. This is a few cases where a predictive tracking algorithm was tested with motion coming from a body part other than the user's head.

2.7.4 Neural Network

In an effort to simulate a learning mechanism, researchers have developed neural networks (multi-layer perceptron with back-propagation learning) in the form of mathematical models (Garrett, 2003). They are software implementations of the neural dynamics and connectivity within the human brain. This prediction composed of a multitude of neurons representing simple processing elements that operate in parallel. The most commonly used neural network is the back-propagation network, which is capable of solving problems that are non-linear in nature. Two major components of neural network are an activation function and a method for updating the weights, which is knows as the learning rule for the network. The activation defines how the neuron responds to the inputs. Meanwhile the learning rule takes the inputs, outputs and expected output as the parameters (Garret, 2003).

For the output layer:

$$W_{ji} = W_{ji} + \alpha \times a_j \times g'(in_i)(T_i - 0_i) \quad (1)$$

Where W_{ji} is the weight from hidden node j to output node i, α is the learning rate, a_j is the output of the j^{th} hidden node, $g(x)$ represents the activation function, in_i represents the i^{th} input to the network, T_i represents the correct output of node i and 0_i, represents the actual node i.

For the hidden layer:

$$W_{kj} = W_{kj} + \alpha \times I_k \times g'(in_j)\sum_i (W_{ji}(T_i - O_i)g'(in_i)) \qquad (2)$$

Where W_{kj} is the weight from k to j, α is the learning rate, I_k is the input to the node k^{th}, $g(x)$ represents the activation function, in_i represents the i^{th} input to the network, $\sum_i$ is the summation across all neurons i which is the layer above the k^{th} layer, T_i represents the correct output of node i, and 0_i represents the actual output node i. Predictions under neural category are an extension or modified version of the back-propagation architecture.

2.8 Comparison of Prediction

A significant body of work exists on prediction tracking algorithms in both VR and Augmented Reality (Azuma & Bishop, 1994; LaViola, 2003a; Kyger & Maybeck, 1998; Rhijn et al., 2005). The major points of these papers discuss smoothing and prediction of tracking data.

For example, Azuma and Bishop (1994) have developed a tracking system using inertial sensors mounted on a Head Mounted Display. They use an Extended Kalman Filter (EKF) and compare two motion and measurement models, with and without inertial measurements. They had shown that since predicted output signals have more energy, prediction time should be kept below 80ms to avoid jittering. Most parameters that influence prediction performance were fixed. This study also did not take differing sample rates, prediction times and sensor noise variances into account.

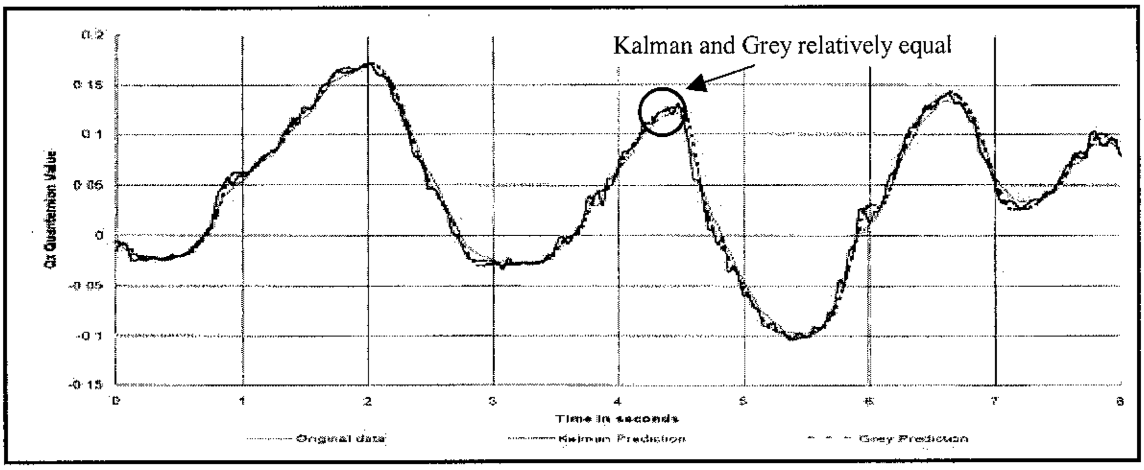
A comparison between an EKF and an Unscented Kalman filtering (UKF) for prediction was made by LaViola (2003a). The prediction time and sampling frequency was varied, no inertial measurements were included and the measurement noise was not varied. Two datasets were used: a hand and a head motion dataset.

La Viola (2003c) has compared three prediction algorithms for hand motion using two datasets: a hand and a head motion datasets. The algorithms in the comparison were KF, EKF and Double Exponential Smoothing (DESP) filter. Although he found that DESP are significantly faster than the KF and EKF, this prediction is slightly more difficult to apply in real time tracking system that produces a variety of signals with different motion dynamics.

Rhijn et al. (2005) compared different filtering methods for orientation prediction of hand movements. They concluded that there are no favourite filters. Their results were strongly depending on the input data, parameters and requirements. This means for best performance a lot of tests are needed to find the most appropriated parameters. Additionally, an adaptive algorithm is suggested, which adapts to the requirements of the application.

Although Kyger and Maybeck (1998), report that Multiple Model Adaptive Estimation (MMAE) schemes (bank of filters) yield better estimates than single prediction schemes, each filter is implemented under the assumption that the process noise terms act as independently. However, because no filter is truly an accurate description of the trajectory, time correlated errors are still introduced and degrade the performance of the filter.

Figure 2.9: A plot of Quaternion x with Kalman prediction without inertial sensor vs. Grey system (Source: Wu & Ouhyoung, 1995; Wu & Ouhyoung, 2000)



A comparison of prediction Grey system was presented in (Wu & Ouhyoung, 1995; Wu and Ouhyoung 2000). They have compared three prediction algorithms for head motion, using the same datasets. They do not include inertial measurements and performed their analysis using two prediction times. The algorithms in the comparison were an extrapolator, KF and Grey system theory-based prediction. Figure 2.9 shows plot of Quaternion (x) between Kaman prediction and Grey system. Results from KF and Grey system produces are significantly better than that without prediction on the average. Both algorithms improved performance when latency is as high as 120ms, however in terms of jittering Kalman appears to have the largest. Although the robustness of Grey system has been proved, their testing were limited to only testing with head motion sequences and no experiments have been carried out to examine its performance for other motion.

Apparently, none of the mentioned studies include comparison on which prediction algorithms are suitable for training applications.

2.9 Heuristic

As previously identified in Section 2.7, previous prediction technique insert a complex mathematical algorithm to extrapolate to a future time ahead of the current position and orientation states obtained form motion sensors or tracker measurements (Yung et al., 2000). While most of the previous predictions focus on mathematical, this research focuses on an approach known as heuristic. The connection between graphics and artificial intelligence (AI) is becoming increasingly strong, and on the one hand, it is now clear that AI research can benefit tremendously from embodiment in virtual worlds (Thalmann & Monzani, 2002). Heuristic technique is a major area of AI research (Perkins, 1981).

Heuristic research has been very active for the past few years in humanoid robotics, AI and computational geometry. Many of the heuristic approaches have been implemented for several years now and have working prototypical systems capable of planning motions in a variety of situations. This is because, simulation and analysis of human motion is a popular issue in sport applications (Jie et al., 2004; Perse et al., 2005; Hamalainen et al., 2005; Zordan, 2002). In reality human body has characteristics according to scenarios. Although heuristic is common technique for animation, VR has not been used widely in the field. Heuristic-based had been implemented in several virtual training studies (Rusdorf & Brunnet, 2005; Bideou et al., 2004; Noser & Thalman, 1997).

The study for human motion, which has been done in the robotic and computer animation fields, can be categorized into two types. The most basic researchers are based on finding the pose or transition by optimizing the particular evaluation values, such as jerk of joint angles and integration of the joint torque. They are mainly considering the arm movement and proposed models are evaluated based on the simulation result and the actual human motion. They also focused on system architecture for an interactive applications and also controlling the simulation with a motion capture. Until now the manner in which people move objects and which principles are used for timing and creating trajectories is still unclear.

The basic motion-planning problem for humanoid robotics is to move a robot from a start location to a goal location while avoiding collisions with obstacles and minimizing the path length. This is because a general-purpose robot with human-like skills has been the goal of many robotic researchers. Therefore, several systems had been developed for humanoid robots for example Mas and Thalman (1994), Cho et al. (1997), Tsuji et al. (2004) and Kagami et al. (2002).

In robot motion planning framework, the problem is to find the trajectory of the robot that would avoid all the obstacles in the environment with a certain margin of error. In general, robot motion planning around obstacles is intractable. Most algorithms combined both the space and time and apply heuristic search or dynamic programming techniques to find an optimal trajectory. To further reduce the time complexity, most algorithms consider only kinematics properties of robots when searching for optimal motion, although some theoretical results have been reported in optimal kino-dynamic planning (Mas & Thalman, 1994).

In contrast to industrial applications of robots, principles of minimization (trajectory lengths and energy costs) are not suitable for designing humanlike motions of service robots. For example, there are no laws describing the interaction of upper arm, fore arm, hand, and fingers during simple displacement of a cuboid. Characteristics of motions significantly depend on various influences like, e.g., velocity, environment, or anatomy.

Most graphics-related research topics with human motion have focused on heuristic for guiding them in designing an avatar, as in Baek et al. (2003), Zordan (2002), Bideau et al. (2004), Nakazawa et al. (2003), Popovic and Witkin (1999), Jie et al. (2004), Xia et al. (2005), and Noser and Thalman (1997). They generate realistic human motion simulation-based on the analysis of the capture human motions.

Other investigators had combined heuristic with other available prediction. KF is the most widely used approach because it has the ability to correct and predict data, but care must be taken for non-linear systems due to the mathematical complexity of the KF. For example, Rusdorf and Brunnet (2005) had developed table tennis application because this game involves very fast movements and moderate space requirements. In simulating the ball bounces off after a collision, they make use of a heuristic approach. In Rusdorf and Brunnet (2005) system they used filter for noise reduction of the tracking data. They implemented additional functionality to the system which, predict position based on velocities and accelerations and simple rule for the human body. This system was nearly realistic in behavior with respect to the ball motion including the influence of the spin. Their results tend to "jitter to a bitter" since research had proven that filter prediction contributes the largest jittering effect. The limits of this simulation with respect to correct prediction become obvious. The problem with their system was the difficulty to predict the very fast movements of professional players.

Kiruluta et al. (1997) used a magnetic tracking device to predict head positions with the goal of reducing throughput delays in a system that uses instantaneous positions to engage a selected course of action in relatively short time spans. The kinematics model was implemented in a KF and used to generate one-step-head prediction estimates for a number of head trajectories. However, under abrupt head movement conditions, large prediction errors occur.

2.10 Summary

This chapter provides a brief review of the main existing methods for reducing latency in VR system. As this research, aim is to determine the suggested prediction algorithm in virtual training application; the researcher considers that it is convenient to present the existing work in this area. The next chapter will discuss in detail design and development of the system implemented.

CHAPTER THREE
IMPLEMENTATION FRAMEWORK

3.0 Overview

This chapter provides a gentle introduction to the implementation of novel prediction algorithms using Grey system prediction and heuristic-based prediction for predicting a user's pose. It will explore the mathematical and theoretical reasoning behind the current implementation, and help clarify reasoning behind the system goals listed in Chapter One.

Based on Section 2.6, the researcher summarized that the main reasons of comparing heuristic-based prediction with the Grey system prediction is because the Grey system has a:

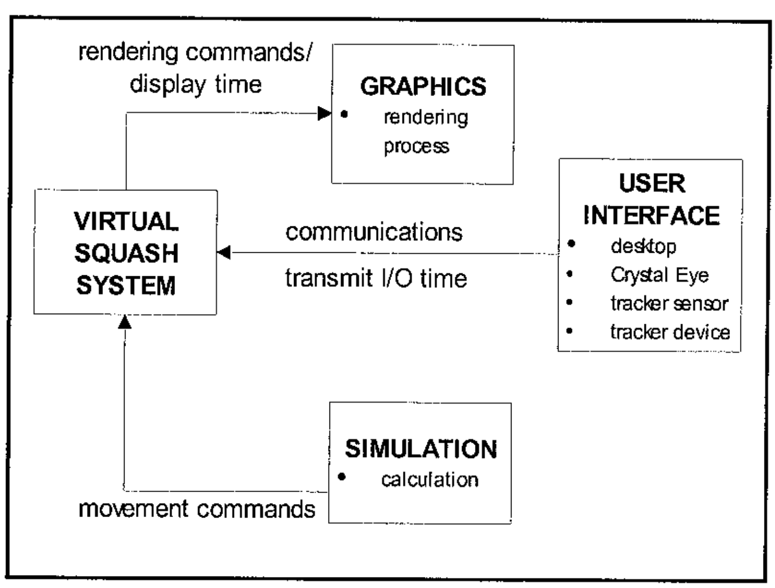
1) relatively low computation compared to other prediction,
2) lower jittering effect and
3) comparable results with Kalman filter.

3.1 Illustration of Latency

The total latency of this system was defined as the delay between the movement of the real objects (e.g. the racket) and the corresponding visible feedback at the screen. There are two fundamental issues that affects speed and latency requirements. The first concerns are the sensor values, which are the most sensitive to performance and the second, the elements that comprise total system lag. This research has chosen the keyboard for the purpose of prototyping the real-time tracker system. Since, a keyboard uses as a tracker prototype, artificial latency was added to show the impact of latency in squash simulation.

Figure 3.1 illustrates system components that contribute to the time delay between time sensors are read and the time an appropriate scene is rendered. The total latency of this system is the sum of the latencies of the sub components of the systems. The sub components are: 1) user interface, 2) simulation and 3) graphics.

Figure 3.1: Illustration of latency



The user interface system is the first element in the chain of components. The first system delay occurs within the sensor itself and is called acquisition or computation time. It consists of the desktop monitor, Crystal Eye®, tracker sensor and tracker device in order to physically realize the squash simulation. Every sensor device has its own internal time latency; for example, the Flock of Birds tracker has 18.96 ms delay (Mine, 1993). Sensors are typically hardware devices that provide raw data about the participant to be used by the renderers and the application to provide the VE. Once the sensor has determined the value, sensor must be communicated to the host processor. This is transmit or I/O time.

Depending on the communications method, the format of the data being transmitted and other factors such as flow control may add another 3.65 ms or more to total system latency (Mine, 1993). The simulation subsystem performs the calculation for the scientific phenomenon. Squash simulation involved calculations for the following:

1) position,
2) rotation,
3) fog,
4) speed,
5) direction,
6) friction,
7) air friction,
8) gravity,
9) stereoscopic,
10) light and
11) acceleration.

The graphics subsystem performs the calculations needed to render objects used in the display. Depending on how data is going to be interpreted and used, additional time may be added. Finally the processed sensor data is sent to the renderer so that the current image may be created. Although a variety of renderer and display components may contribute to the display time, the level of detail will not be considered here. Renderers are software modules that drives the display hardware to create sounds, graphics, or other physical manifestations of world entities. Entities may have a variety of attributes that may be displayed by one or more renderers. Any renderer that displays a spatial quality of an entity, such as a graphical or aural renderer uses the position attribute. Other attributes such as a graphical description or loudness may be specific to a single renderer. Renderers are responsible for driving related display hardware and software. In terms of rendering, since the current squash simulation does not have too much polygons, the rendering delay is short.

According to http://www.squashplayer.co.uk/fitness_factory.htm [May 2005]:

> *"Squash is a game, which may last from 6 minutes to 2 hours 48 minutes. One rally may last one stroke and 1.5 seconds, or 400 strokes and 10 minutes. More usually – for training purposes – rallies can be categorized into three populations, those lasting less than 5 seconds, those between 6-20 seconds, and a small but important number which last upwards of 20, but not usually longer than 60 seconds. There are approximately 7 seconds between rallies, which are played at a rate of up to 40 strokes per minute."*

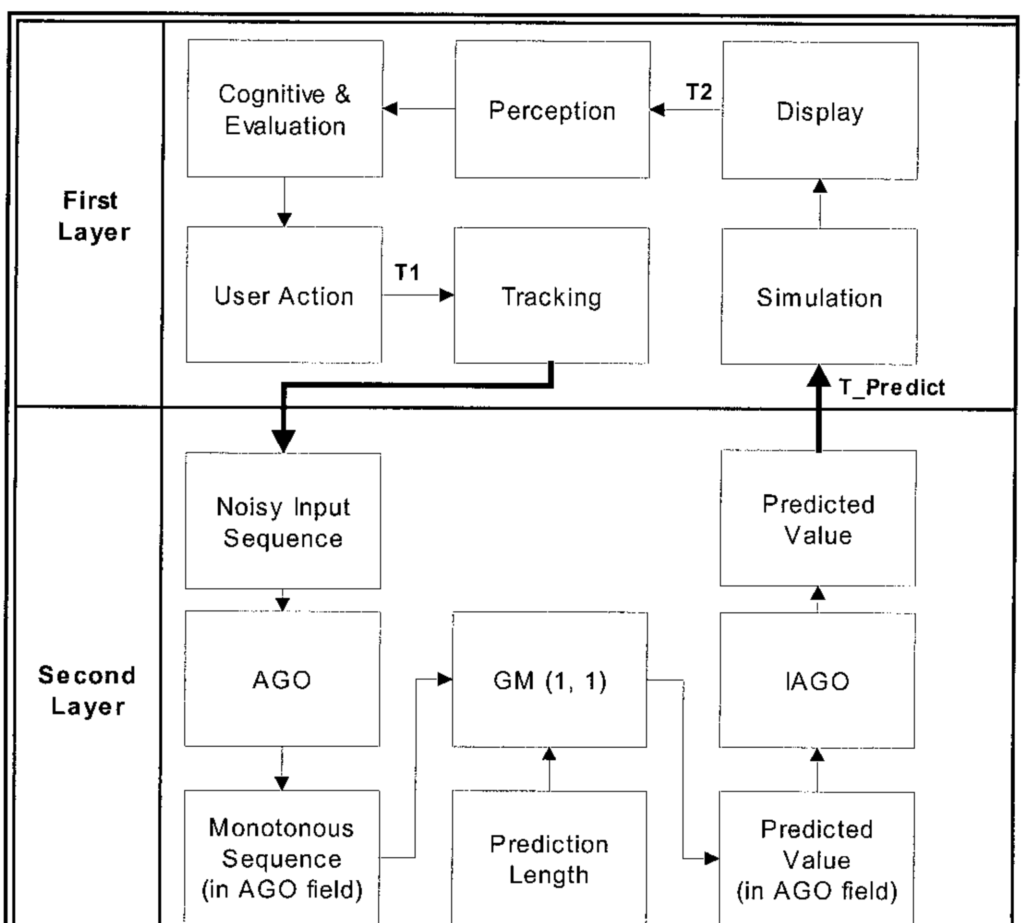
Speed and latency are probably the two most important characteristics affecting a comfortable and believable VE. This is because, speed and latency are the main qualities of the VR systems. Based on analysis of latency components and squash game itself, which involved very fast movements, this illustrates why the squash simulation needs prediction algorithms.

3.2 Main Design

Figure 3.2 and 3.4 present frameworks for comparing predictive algorithms. In both figures, two-layered frameworks are used using Mine's (1993) technique. Meanwhile the second layer shows the prediction stages, which takes place in interaction cycle between tracking and evaluation stages.

3.3 Integrate Squash Simulation with Grey System Prediction

Figure 3.2: Framework of Grey system prediction



Grey System Prediction (Source: Wu & Ouhyoung 2000)

Legend

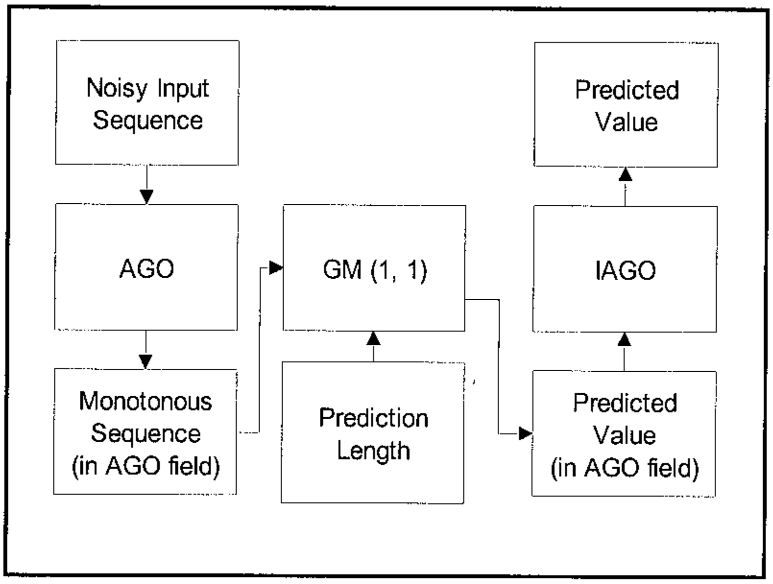
T1	: time user performs an action
T2	: time result reflect to the user
T_Predict	: prediction time
AGO	: Accumulated Generating Operation
IAGO	: Inverse Accumulated Generating Operation
GM	: Grey Model

Grey system was developed so as to make comparisons with heuristic-based prediction. Figure 3.2 shows the framework of Grey system prediction. Grey system is a set of mathematical equation that quantitatively predicts based on first-order dynamic Grey Model named as GM (1, 1). The tracker data obtained by measuring the system is lack of inter-relations and insufficient to establish a Grey Model. Since the relation among the motion of player is grey, the GM (1, 1) is applied to predict the motion of the player. Instead of forming a knowledge base, the Grey Model constructs some differential equations to characterize the controlled system behavior.

Under this prediction, the system model is established under a sequence of measured raw data, which is generated by the system with unclear characteristics. Some manipulations of the raw data are needed to get a more regular data sequence, and the newly obtained data is called the generated sequence. As with most Grey systems, the Accumulated Generating Operation (AGO) is used in the design. Grey Model and Grey Generating Space were generated from the observed sequence and applied to fit the generated sequence. As a result behavior of original system can be predicted and analyzed by using the established Grey Model. This research presents the basic of Grey system algorithm using a more algorithmic description.

3.3.1 Algorithm Outline of Grey System Prediction

Figure 3.3: Grey system algorithm outline (Source: Wu and Ouhyoung, 2000)



Legend

AGO	: Accumulated Generating Operation
IAGO	: Inverse Accumulated Generating Operation
GM	: Grey Model

Figure 3.3 shows framework of Grey system algorithm as cited by Wu and Ouhyoung (1994, 1995, 2000). The following description basically follows that of Wu and Ouhyoung.

One commonly used operation for the generated sequence is called the accumulated generating operation (AGO). Let $x^{(0)}$ be the original tracking data sequence and $x^{(i)}$ be the generated sequence for $i > 0$. The AGO is defined as:

$$x^{i} = AGOx^{(i-1)}, i > 0 \quad (1)$$

$$x^{(i)}(k) = \sum_{m=0}^{k} x^{(i-1)}(m), i > 0 \quad (2)$$

Since $x^{(0)}$ are all positives, after applying the AGO, the generated sequence $x^{(j)}$ must be a monotonically increasing sequence, and its randomness disappears. The prediction model GM can be established in an AGO domain.

With n samples and $x^{(1)} = \text{AGO}\, x^{(0)}$; $x^{(0)}$ is the original tracking data sequence. It is assumed that this method satisfy the following first-order grey differential model, GM(1,1) with a single variable:

$$x^{(o)}(k) + az^{(1)}(k) = b, k = 1,2,... \tag{3}$$

$$z^{(1)}(k) = \frac{x^{(1)}(k) + x^{(1)}(k-1)}{2}, k = 1,2,... \tag{4}$$

These are obtained from the following differential equation:

$$\frac{dx^{(1)}(t)}{dt} + a \bullet x^{(1)}(t) = b \tag{5}$$

Expand (3) with n samples in $x^{(1)}$, that is, in the AGO domain. The result as follows:

$$\begin{bmatrix} x^{(0)}(1) \\ x^{(0)}(2) \\ \vdots \\ x^{(0)}(n-1) \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}x^{(1)}(1)+x^{(1)}(0)) & 1 \\ -\frac{1}{2}x^{(1)}(2)+x^{(1)}(1)) & 1 \\ \vdots & \vdots \\ -\frac{1}{2}x^{(1)}(n-1)+x^{(1)}(n-2)) & 1 \end{bmatrix} \begin{bmatrix} a \\ \\ \\ b \end{bmatrix} \tag{6}$$

Let

$$Y = \begin{bmatrix} x^{(0)}(1) \\ x^{(0)}(2) \\ \vdots \\ x^{(0)}(n-1) \end{bmatrix} \text{and } B = \begin{bmatrix} -\frac{1}{2}x^{(1)}(1)+x^{(1)}(0)) & 1 \\ -\frac{1}{2}x^{(1)}(2)+x^{(1)}(1)) & 1 \\ \vdots & \vdots \\ -\frac{1}{2}x^{(1)}(n-1)+x^{(1)}(n-2)) & 1 \end{bmatrix}$$

Solving (6) with the minimal square approximation, a and b will be obtained from the following equation:

$$\begin{bmatrix} a \\ b \end{bmatrix} = \left[B^T B\right]^{-1} B^T Y \tag{7}$$

By solving a and b, the differential equation, prediction function $\hat{x}^{(1)}(k)$ for the grey system in the AGO domain obtained:

$$\hat{X}^{(1)}(k) = (x^{(0)}(0) - \frac{b}{a})e^{-ak} + \frac{b}{a}, \text{ for } k \geq 0 \tag{8}$$

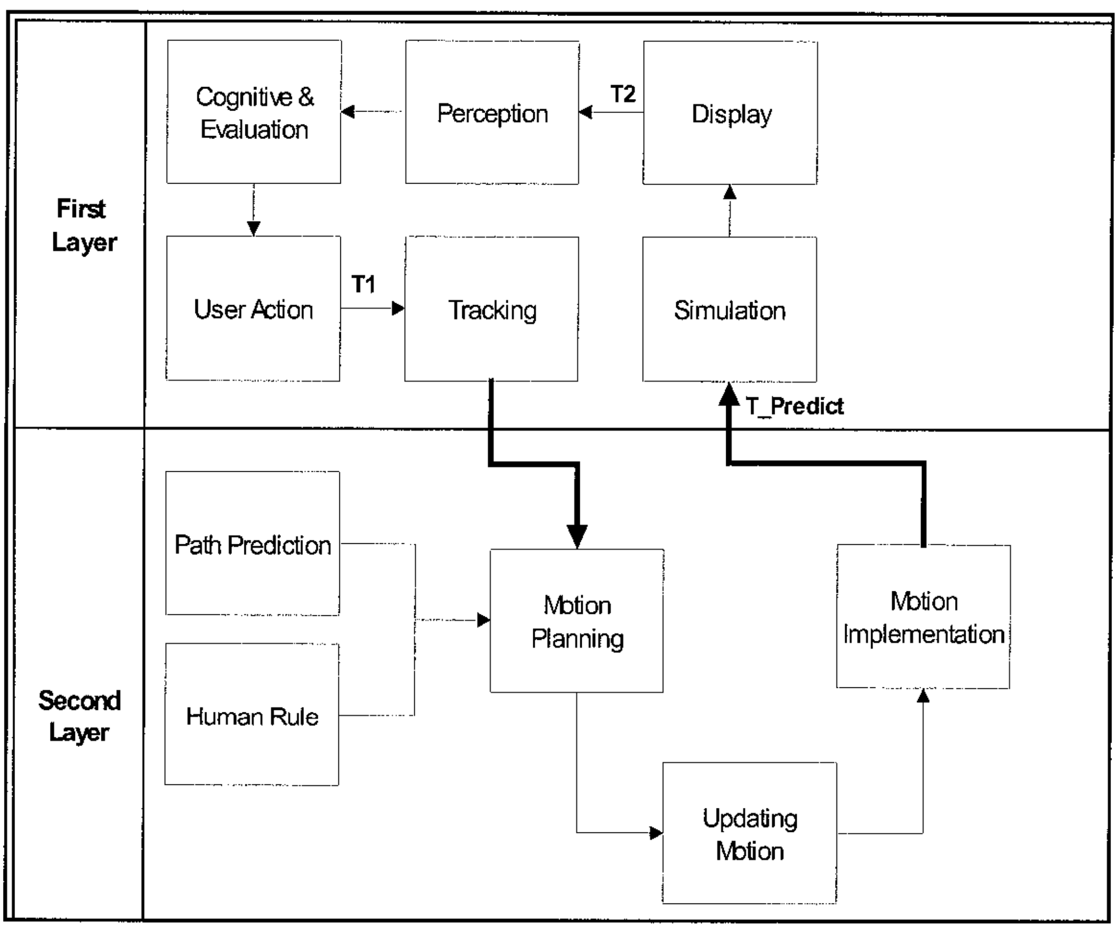
Applying prediction length k in terms of the update period to the prediction function, the predicted data in the AGO domain is acquired. After the inverse AGO (IAGO) defined in (9), the output data $\hat{x}^{(0)}(k)$ from the IAGO becomes the predicted data that is needed. The predicted value will predict the future behavior of the player as follows:

$$\begin{cases} \hat{x}^{(0)}(k) = \hat{x}^{(1)}(k) - \hat{x}^{(1)}(k-1), \text{for } k > 0 \\ \hat{x}^{(0)}(0) = \hat{x}^{(1)}(0) = x^{(0)}(0) \end{cases} \tag{9}$$

3.4 Integrate Squash Simulation with Heuristic-based Prediction

In order to convey a sense of immersion, a virtual training system must not only present a convincing visual rendering of the simulated objects, but also to manipulate them in a fast, precise and natural way (Hamalainen et al., 2005). Based on this, this research proposes to apply kinematics algorithm, which combines natural and unobtrusive interaction through gesturing with precise and general-purpose interaction. The basic heuristic-based algorithm is to move the player from a start location to a goal location, thus minimizing the path length in terms of time.

Figure 3.4: Framework of heuristic-based prediction



Legend

T1	: time user performs an action
T2	: time result reflect to the user
T_Predict	: prediction time

Figure 3.4 shows the framework of heuristic-based prediction. Motion planning incorporates path prediction with simple human rule, to develop an integrated reasoning module for richer expressiveness and robust motion. The path prediction is an analysis of human behavior and specific domain knowledge of sport application. Motion planning is use to update the player's motion which can be referred to as motion implementation or heuristic-based prediction.

The heuristic-based prediction is able to recognize the motion trajectories because:

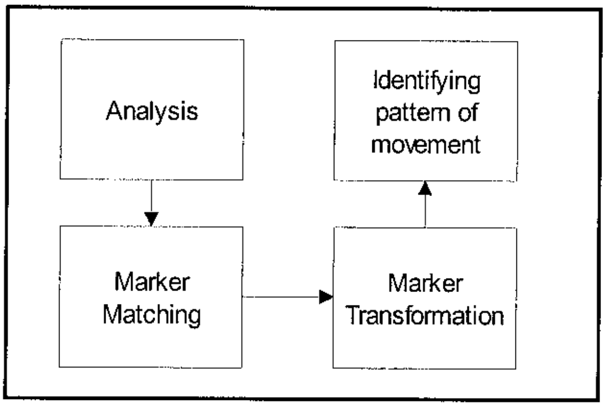
1) human motion sequence consists of some kinds of motion elements, namely primitive motions (Nakazawa et al., 2003). The primitive motions consists of a basic motion, which is common to all players. Furthermore, human motion is cannot modeled exactly by mathematics and physics method because it is very complex and controlled by brains (Xia et al., 2005). As sport is a game of habits, during sport, player typically acts in a certain habitual pattern.
2) In training a free motion, the kinesthetic aspect of an action is important (Baek et al., 2003). In developing a robust prediction algorithm, it is important to achieve good estimates of the motion kinematics. To support learning of kinesthesia, a motion training system, should provide an integrated environment. Humans have developed a set of powerful heuristics for planning motion by solving motion planning problems whenever they move (Hwang et al., 1997). For this purpose, predicted motion is implemented, as it is highly valued in sport motion training and analysis.

Wan et al. (2003), mentioned, it can be quite difficult to capture all the motions of a movable object into analytical equation. According to Hodgins (1998), although specific methods were used to produce human-like motions, it still remains quite difficult to design a controller for all possible gestures, even for a limited task.

The heuristic chosen here is to place the foot in such a position so as to allow maximum movement of the player in the current direction. That is, it is preferable to place the foot further forward in the current direction of motion rather than only one position away from the extend of its range (where in just one more move it will become critical). If any choice of location remains after these stated rules, then the foot is to be placed towards the centre of its range to provide maximum flexibility.

3.4.1 Methodology of Heuristic-based Prediction

Figure 3.5: Flowchart of heuristic-based prediction



The methodology of proposed prediction is described in detail in the following sub-sections. The sub-sections are organized according to the flowchart in Figure 3.5.

3.4.1.1 Analysis

In analysis phase, the researcher deals with the tracking of people during sport matches. The main purpose of tracking people in squash game is to acquire data of player's position on the court. This new prediction is proposed based on:

1) theoretical and empirical material (video recordings, interviews and observations),
2) motion capture and
3) kinematical operators (provided by biomechanical analyses).

It is hope that this analysis will assists the researcher in devising better training strategies and game tactics.

For a precise comparative analysis of inter-individual and intra-individual variances of motion patterns, kinematics data of numerous motion studies had been collected. Data was collected through from game events and from self-arranged gaming sessions. A total of twenty-five (25) games have been studied during each of the competitions. As it is believed that the coach's experience is indispensable to sports simulation (Xia et al., 2005), the researcher conducted interviews and observations with squash experts. Cooperation with the squash experts enabled to obtain annotations describing the exact motion of a player. The first analysis assisted in predicting the motion sequence of the player.

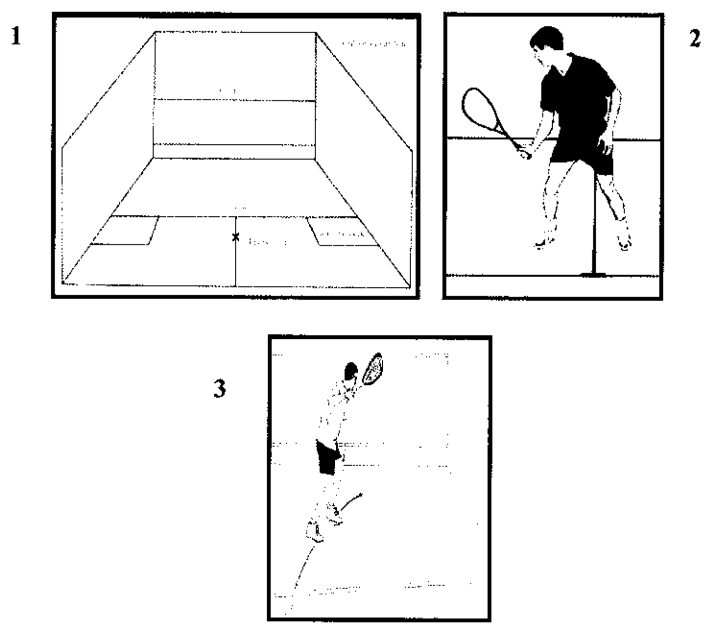
The researcher proceeds to identify those details of the player's positions on the court. Motion capture is dynamically realistic around the neighborhood of the original motion. According to (Gleicher, 1998; Zordan & Horst, 2003), motion capture is widely investigated to ensure the calculation of human like motions. By capturing the motion of many different kinds of player and thoroughly analyzing and comparing this motion, the researcher progress further towards understanding the range of motion attributes that a player is capable of.

However motion capture alone is generally geometric and did not take into account biomechanical laws and observations into account. To overcome this limitation, this algorithm also applies kinematical operators (provided by biomechanical analysis), which is able to adapt the trajectories to new orders and constraints. According to Lopez and Platen (2005) biomechanical motion analysis offers a method by which the very fast actions occur can be recorded and analyzed in detail. In order to animate this structure, specific biomechanical experiment was carried out. The biomechanical experiment conducted based on Bideou et al. (2004).

Five subjects participated in this motion capture study. Subjects played at Universiti Teknologi PETRONAS's squash court. A motion capture system (Vicon370, Oxford Metrics) composed of 4 infrared cameras were used to capture the subjects' motions at 60Hz. Reflective markers were placed over standardized anatomical landmark. The subjects were asked to perform 10 standardized hits at maximum speed into a square target. The researcher calculated the joint centers depending on the position of the external markers if missing points occur.

The primary information gained when tracking players are their positions on the court. This is why the human body movement can be modeled as a motion of a single point, where the selected point represents the gravity center of the racket.

Figure 3.6: Squash strategies: 1) T-position (Source: Collins et al., 1990), 2) Player on the T-area (Source: Yarrow, 1997) and 3) Player use a side stepping motion to move to a position beside the ball before striking it (Source: Yarrow, 1997)



Humans perform a variety of proficient movements by adjusting dynamic characteristics of their musculoskeletal system in motion (Tsuji et al., 2004). In this case the motion primitives for squash player are the position and also the style of movement. For example, a squash player starts at the basic position, which is a balanced position of the body (see Figure 3.6). Besides starting at the basic position, a player will move to and from the basic position between shots. The basic position allows the player to move off immediately in any direction with ease and the player will be able to run quickly to reach any shot.

Good court movements will bring the player a basic game together (Mckenzie, 1996; Sommers, 1991). Players will be able to position themselves for better shots. The importance of good court movement was mainly for two reasons: 1) it allows the player to cover the court more quickly and more efficiently and 2) help the player to position himself/herself. Players who does not have good movement constantly find themselves out of position, rushing madly around the court trying to keep the ball in play. During a game, a player moves in a slight J-shape position, rather than straight at the ball in order to place the player in a sideway position to play the shots.

Figure 3.7: Trajectory of the international player (left) & Trajectory of the national player (right). (Source: Vuckovic et al., 2005)

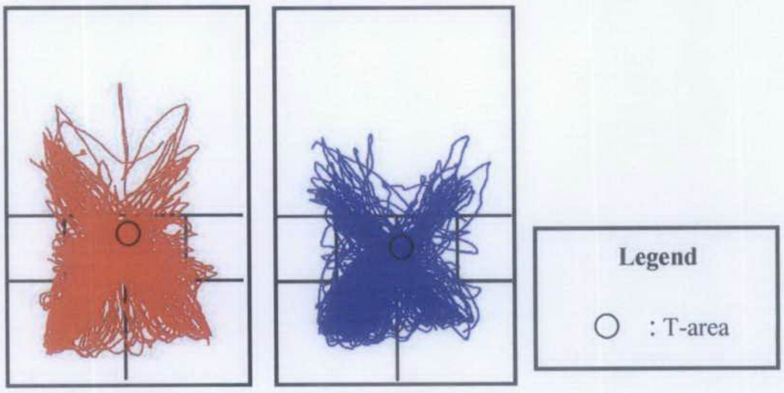
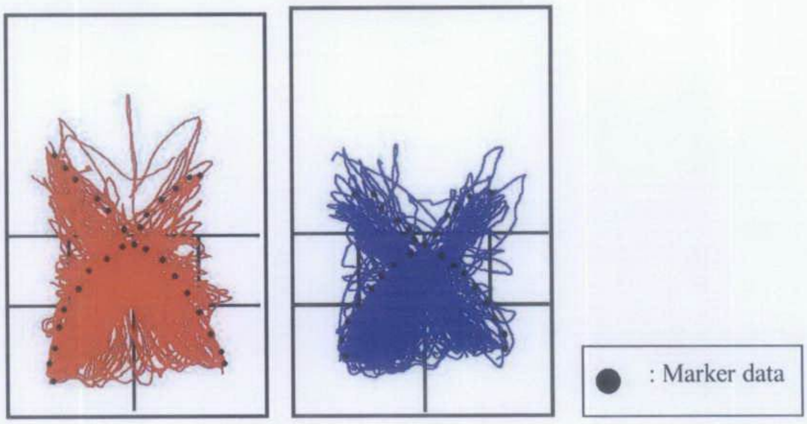


Figure 3.7 depicts typical X-shaped motion of players during a game, which suggest the similar play tactics of player of different quality rank. With an accurate stroke, the player is forced to move from the T-area to the stroke area and back, even with similar playing tactics, as it can be seen in Figure 3.7. From this trajectory, the researcher easily recognizes motion sequence of the squash player, which is the repetition of the primitive motions. When a player repeats a particular sequence of shots, the game begins to take on the form of a series of patterns of play. Squash experts have recommended several patterns in single matches for training. The patterns are based on several strategic principles that have been well tested over time

(see Figure 3.7). Most professional players follow these patterns and trajectory can be classified into one of the patterns.

3.4.1.2 Marker Matching

Figure 3.8: Marker matching



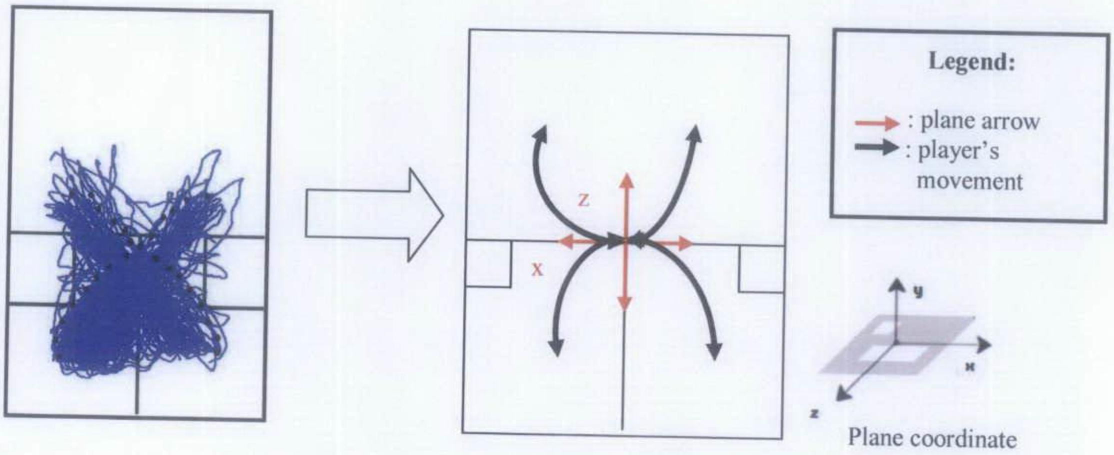
Based on analysis, the researcher summarized that movements of the sport's player are usually related to various strategic actions. The motion of the squash game comprises of different stops and poses, changes of motion direction, turns, jumps and side-steps. As previous works demonstrated that the footwork mainly influences a player's reaction, thus heuristic-based prediction focuses on the position and movement analysis.

Vuckovic et al. (2005) have shown highly and statistically significant correlation between duration and distance covered by players. According to Vuckovic (2005), average racket speed is approximately 0.725 meters per second for national players. Of course the maximum speed in a usual squash game is much smaller, especially between non-professional players. The intention of the heuristic-based prediction is to determine enough information to robustly track the position of the squash player. Real-time posture estimation is a fundamental technology for analyzing human behaviors (Ohya, 2002). For real-time determination of these parameters without having the need for a constrained environmental conditions, a physically-based mapping approach was used. It is a forward dynamic model, which had been proposed by Zordan and Horst (2003). The advantages of this method are: 1) does not need hand-crafted heuristics, 2) does not calculate intermediate body orientations or joint centers and 3) does not need to give priority to any specific markers data (Zordan & Horst, 2003).

An understanding of biomechanics and how the player moves their body is needed for planning marker placement. Specifically, the marker data guides the physical simulation movement of the player to the closet motion. Figure 3.8 illustrates the procedure of marker matching. The basic requirement for a key frame animation system is the ability to store and interpolate between values in any number of animation tracks. Although key frame animation is a standard technique, challenges are presented by: ensuring that the order of evaluation remains consistent; making certain keys do not conflict with one another; and ensuring that all portions of the squash simulation update with the modification of time. The key frame animation system used in this research must address these concerns. Because of this, time intervals during which each foot was in contact with the court were also stored. Based on the analysis of squash game, the marker position was chosen with the key frames spaced at 0.03 sec and the marker shapes in used are round blips.

3.4.1.3 Marker Transformation

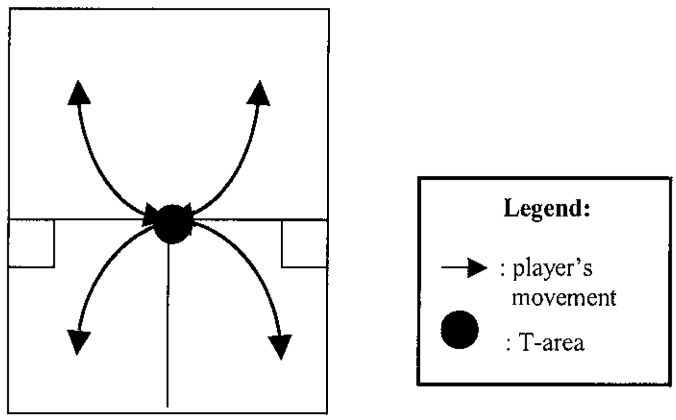
Figure 3.9: An example of marker transformation: Motion in x-z plane



In contrast to most physically-based mapping techniques that synthesize motion from scratch, motion transformation is used in this research as the underlying paradigm in generating motion sequence. In doing so, based on marker matching, a set of normalized trajectories is calculated and stored. Figure 3.9 shows pattern of squash game that resultant from marker transformation. Since the motion of most objects in nature is continuous, the resultant of motion trajectories is piecewise smooth. This is because motion of the player can be described by the direction of body links.

3.4.1.4 Identifying Pattern of Movement

Figure 3.10 Movements/trajectories of the squash player



Two of the most popular representations for graphical curves and surfaces in Computer Aided Design (CAD) and gaming technology field are the Bezier and B-spline curves and surfaces (Takayama & Kano, 1995). To preserve natural looking, i.e. physical smooth motion, the interpolation used between key frames must satisfy the continuity. This means that there are no sudden jumps in motion, velocity or acceleration. As concept of unit motions is introduced as smooth primitive motions in the coordinate system and also based on movement's pattern of the squash player, heuristic-based prediction for squash game's movements employed B-spline curve, which is usually constructed by stitching together with Bezier curve. The researcher used Bezier curve for interpolation and B-spline to get smooth movements. The B-spline curve is an extended version of the Bezier curve that consists of segments, each of which can be viewed as an individual Bezier curve. A B-spline curve is a piece wise-polynomial parametric curve that satisfies certain smoothness conditions. The Bezier curve which is curve interpolation was formally presented in (Bezier, 1970) and has since then been a very common way to display smooth curves, both in computer graphics and mathematics.

Physics says that the general position of a rigid body can be given by combining a translation with a rotation. Because of this, B-spline is described by hierarchical knot vector and the new motion was synthesized by manipulating these vectors. Figure 3.9 shows how the researcher creates a motion by using disjoint segments in one B-spline

(done through the knot vector) and interpolate the movement through Bezier. Specifying paths of movement through B-splines have several advantages. If the spline passes through an obstacle the researcher modify the curve to obtain a smooth motion around it, a very fast operation, which can be done in real time. For instance, if the researcher want an the player to move from T-position and return to T-position, the researcher first creates a straight line (B-spline) and try it for intersection with other objects in the scene, if it collides with anything the researcher manipulates the curve. Meanwhile, Bezier curve chosen for interpolation due to its rapid evaluation and sculpting simplicity.

According to all those data, heuristic-based prediction was directly developed based on the analysis made. Moreover, this algorithm takes into account some simple movement rules for the human body. For instance, it is not possible to move out of the range of their body part and to rotate at any angle. Conclusively, heuristic-based prediction implemented in this study is a consequence of the specific motion pattern for the squash's player based on the analysis and also rule of the human.

3.4.2 Algorithm Outline of Heuristic-based Prediction

Figure 3.11: Heuristic-based algorithm outline

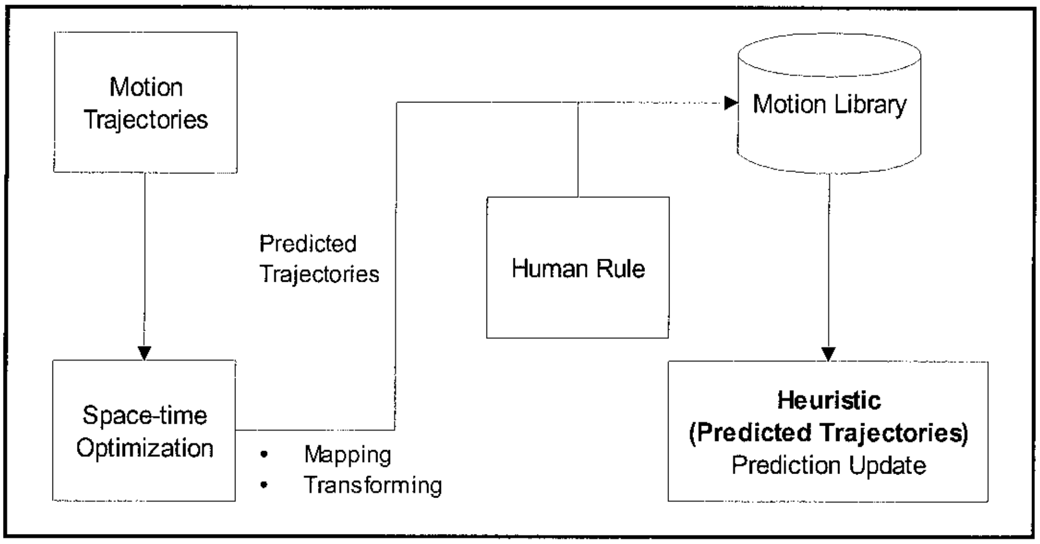


Figure 3.11 shows the algorithm outline of heuristic-based prediction. Space-time optimization model was one of the approaches, which had been developed for ergonomic human motion simulation (Park et al., 2004). The core element of this algorithm is the uses of space-time optimization, which maintains the dynamics integrity of motion as well as providing intuitive motion control. The primary problem of space-time method is that they do not guarantee a solution for motion problem of complex character such as human. The researcher circumvents this issue by developing smaller and abstracted model motivated by biomechanics research.

The first step of this algorithm is to map the original motion data. From this mapping data, the algorithm does not synthesize motion from ground zero. Instead, it transforms the mapping motion to satisfy the needs of the motion trajectories. Once the space-time motion model has been constructed from the input data, appropriate simple human rule was applied. The resulting motion has motion-like reality due to the motion analysis and motion rule. Then a motion sequence of the player can be turned into trajectories and stored into a motion library. Motion library is the base of heuristic-based prediction, which is convenient and rapid. This motion library is capable in generating all possible movements of the player, in which it also fits the need (trajectory) of the player. The counteracting motion with new footstep pattern is searched in the motion library. The one that matches the current foot-ground contact is chosen. Finally, it will be used as prediction data to predict the next position of the player during the game.

3.5 Summary

In this chapter, the researcher presents frameworks for both Grey system algorithm and heuristic-based algorithm. The researcher describes the technique of heuristic-based algorithm and contrasted it with the Grey system algorithm. The framework of heuristic-based prediction was inspired by motion planning. Heuristic-based prediction presents the method of mapping motion that takes dynamic and simple human rule into consideration. Heuristic-based prediction is designed under B-spline and Bezier motion. The framework of Grey system prediction and heuristic-based theory, which have been designed, will be integrated with squash simulation. Implementation details of the squash simulation with the integration of Grey system prediction and heuristic-based prediction will be explained n Chapter Four.

CHAPTER FOUR
DESIGN AND DEVELOPMENT

4.0 Overview

The predictions discussed in Chapter Three were implemented in a squash environment. In this chapter, the researcher puts forward a methodology for modeling a squash simulation. The researcher analyzes mainly two questions in building a VE. The questions are 1) what are the possible interactions and goals to be realized, and 2) how the environment will look like, i.e. the visual representation. As a result, the researcher describes what the possible interactions are as well as the visualization of the environment. These issues are dealt with, in Section 4.2, which in fact represents a possible methodology for describing the requirements, which are needed to create squash simulation. These requirements will particularly have an impact on the implementation design, which is also covered in this chapter.

4.1 Game Coding

To ensure the success in developing the squash simulation, middle-end hardware is required. Updated VR software is obtained and a freeware object modeling was downloaded. The graphic system was implemented in OpenGL as a basis of squash simulation. OpenGL is strictly defined as a software interface to the graphic hardware (Fan & Xiong, 2003). In essence, it is a 3D graphics and modeling library that is extremely portable and very fast that available on all major platforms. The most important advantage of squash simulation is that, full access control of all components can be obtained. This means new algorithm at any level of the library can be integrated.

The squash simulation was written in C++. This software model is used because it provides maintainability and encourages good code structure. Further implemented methods include a basic physic engine, collision detection, and components of loading and managing event-based animation system and tracking algorithm.

The development tools and others assisting equipments are listed below:

1) AMD K611 2.66Ghz,
2) 512MB of RAM,
3) Microsoft Visual Studio.net,
4) 3ds Max 7.5,
5) Adobe Photoshop.

4.2 Methodology

In this section the researcher describes the methodology for modeling a squash simulation. The methodology is a user-centered design and the researcher chose to concentrate on the interactions between the player and other entities (defined further) in the environment, and between the entities themselves. So, not solely static VEs, which only provide a representation of the information. The methodology essentially encompasses 3 steps: the requirements, the conceptual design and the implementation design. Each one of the three steps is discussed further. The methodology provides as a first step an outline of the functional requirements, i.e. which interactions the environment must realize and the requirements relating to the visual representation. Of course, there are also non-functional requirements, which are also dealt with in the methodology discussed further.

4.2.1 Requirements

In this section, the researcher gathers all kinds of requirements that needed for squash simulation. The requirements are grouped into 3 categories: functional, non-functional and environmental requirements.

4.2.1.1 Functional Requirements

In this step, the researcher gives a description of the functional requirements that the environment must fulfill. So the researcher first provides a good characterizing of the squash simulation and then the researcher look further into the functional requirements for the VE. There is a good characterizing of the type of application for VEs, proposed within two dimensions:

1. The number of subjects and entities simultaneously involved in the same world.
2. The complexity of the objects and their behaviors, ranging from static data to those responding to subject interaction, and on to dynamically changing objects without user intervention.

It is very important to give an idea of the characterizing as indicated above, because it will have a great impact on the implementation design. The two dimensions given above are typical for a VE and give an idea of the complexity of the application. Hence decisions have to be made in the implementation stage. For example, a possible characterizing would be: It is a single player, where user play the squash game. The user has the ability to interact with objects of the virtual world, e.g.: ball.

Now the researcher is going to have a closer look at the functional requirements of the VE. At this stage the researcher describes the interactions of the environment must be fulfilled. In squash simulation, the interaction is player will play the game by hit the ball.

4.2.1.2 Non-functional Requirements

Non-functional requirements are the system constraints affecting development and design. These may include: performance, portability, operating environment, reliability, etc. Further on the researcher gives a brief overview of the possible input/output devices.

There are several hardware devices for VR systems available, dependent on to which degree in creating the illusion of immersion. Following is a brief description of the input/output devices available for squash simulation.

1. Input devices are those with which the player emits information and interacts with the environment. These include kinesthetic (body movement of the user), acoustics (voice input) and/or graphics (video capturing with possible gesture recognition, etc.).
2. Output devices are those, which give the player information from the world. The output devices will simulate the player's sense. The main senses used in a VR system are: vision and hearing.

Squash simulation implemented is a desktop-VR application. This consists of the keyboard as input channels and the monitor and speakers as output channels. One of the main advantages of this approach is that it does not require a powerful graphical workstation and it is cheap. Other non-functional requirements are, for example, that the squash simulation should be cross-platform (the capability of software or hardware to run identically on different platforms).

4.2.1.3 Environmental Requirements

In this section, the researcher describes the requirements, which are inherent to the environment itself. Environmental requirements are not related to technological requirements, but merely to the 'look and feel' of the virtual environment. Visual representation is some of the issues that are important for describing the environmental requirements.

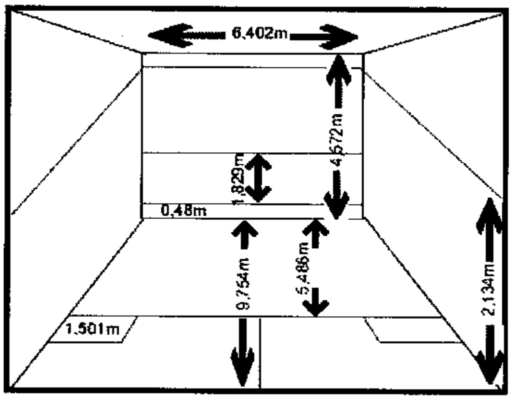
For visual representation, the researcher describes properties of the squash simulation, such as: should the VE be photorealistic or not, how should the player looks like, set of colors that may be used in the environment and audio used. These requirements have an impact on the implementation design.

To get semantic information from vision, squash simulation used light generator, which comprises certain color coding. In squash simulation, light generator allows player to differentiate between the object and the environment. By associating a certain color to an object, and giving the knowledge to player, it can localize objects by the pixel colors of its vision image.

Sound generator can be defined in order to give a realistic audio feedback. It is well known that audio feedback is an important aspect to achieve immersion in VE (Blade & Padgett, 2002). The sound generator mapping converts the movement of the player and the ball using a .wav sound module. The sound generator for the squash simulation contains the following prerecorded sounds:

1) Collision sound: ball_racket, ball_ground, ball_wall.

Figure 4.1: Real squash court [1]



The requirements for the squash court are extremely strict (Collins et al., 1990; Mckenzie, 1996, 1998). The virtual squash was designed based on real court dimensions (Figure 4.1). The court measures 9750 x 6400 millimeters. The clearance between the court plane and the ceiling should be no less than 5640 millimeters. The system is operating in a relatively unrestrained environment. However, a "bounding box" was implemented in the process function to restrict the virtual object movement within specified Cartesian co-ordinates (squash court). The range of movements applied in the simulations avoided angular discontinuities.

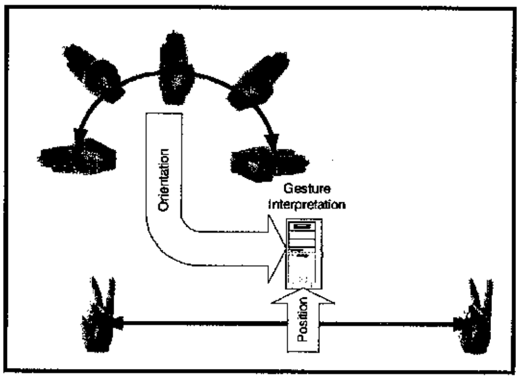
Although human movement cannot be reliably predicted, a simplified model can be used to achieve sub-optimal prediction since an accurate model for human body dynamics is a large problem on its own (Miners, 2001). This is because; the full and detailed body model would be too complex for this use. The motion of the model needs to be consistent with the law of physics. And more importantly, in order for the motion to look realistic, the entire musculoskeletal structure must be taken into account.

For this reason, racket was used to represent player's model. Figure 4.2 shows position and orientation information of the players. To play the squash simulation,

[1] See:
http://www.ffsquash.com/decouverte/ce_quil_faut_savoir/dimension/dimensions.htm [January 2005]

position, orientation and velocity of the racket must be controlled to hit an incoming ball to a particular location with a desired speed. In order to minimize the prediction error, a maximum velocity and acceleration for translation and rotation had been set. The player's constant acceleration and velocity assumption is an approximation of the actual player's dynamics. Such properties help to remove peaks during prediction system.

Figure 4.2: ***Position and Orientation Information (Source: Miners 2001)



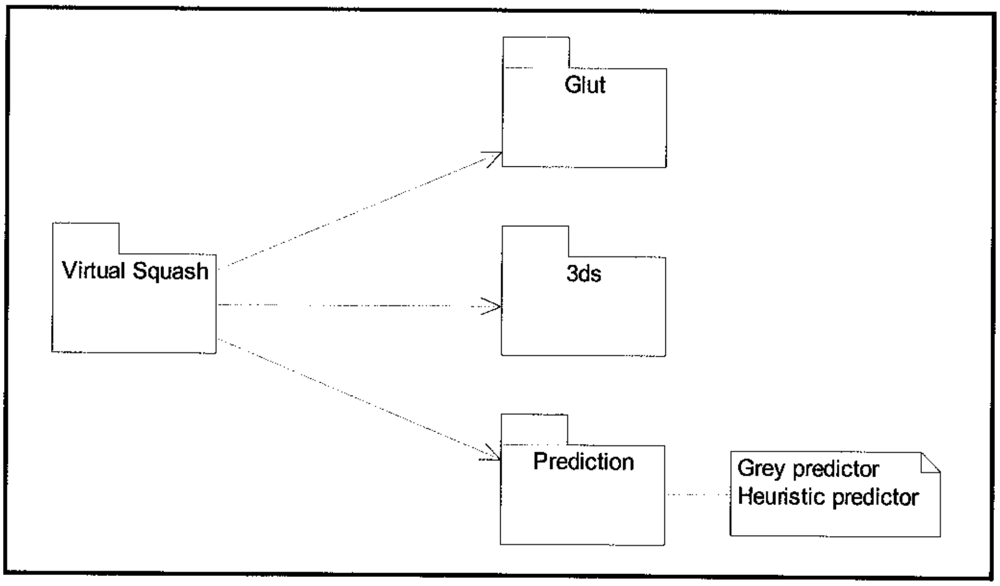
***Hand symbol indicating racket module

4.2.2 Conceptual Design

At this stage, the researcher describes which elements of the squash simulation environment, the user wants to interact with, the interaction itself between the elements of the environment and the visual representation of each one of the elements of the environment. The researcher categorizes elements of a VE into so called entities. This research uses the concept of entities as a basis for setting up and modeling our virtual world. After that the researcher apply the concepts of object-orientation to the entities, which enable Unified Modeling Language (UML) in describing squash simulation.

The UML is an object orient (O-O) technique used for analysis and design of squash simulation. O-O analysis, design and programming have been developed in response to the need for flexibility in squash simulation. O-O has already proved very effective in the development of graphical user interfaces (GUIs) (Ohya, 2002). The UML for s quash simulation was constructed using Rational Rose® software 2002. Below, the researcher gives description of what an entity is and which kind of entities appear in a virtual environment (VE).

Figure 4.3: Architecture for a squash simulation



In UML, class collections are known as packages, and squash simulation architecture is decomposition into a smaller number of packages. Figure 4.3 shows architecture for a squash simulation. This virtual squash architecture groups all of the classes pertaining to player, including associated GUI's. Glut package and 3ds package are external component downloaded from Internet and have major contributions in the virtual squash development. Prediction includes Grey system prediction and heuristic prediction. The predictions give a major impact for the racket. Another function of virtual squash architecture would be to group all displays in a package.

Figure 4.4: Main packages of squash simulation

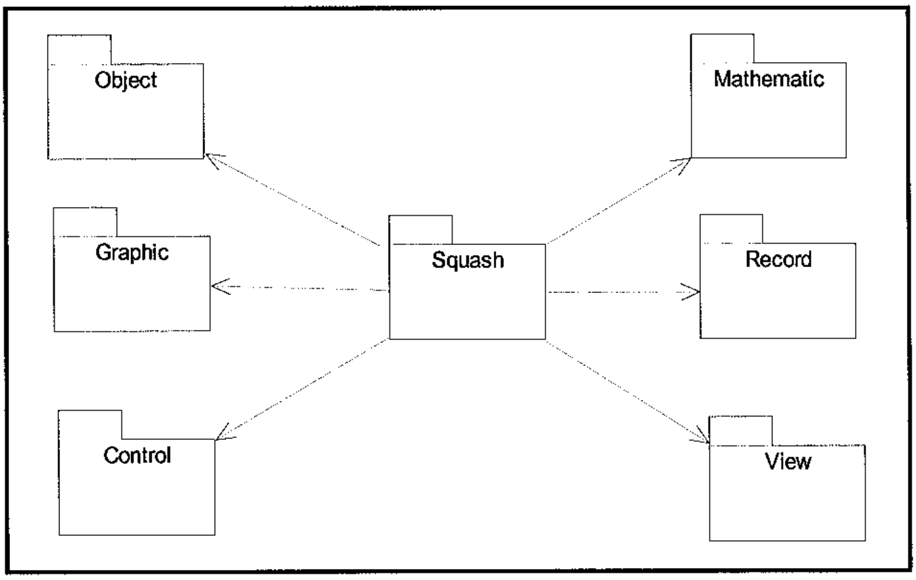


Figure 4.4 shows seven main components of virtual squash packages. It is used to graphically depict the objects and their relationship in squash simulation.

Figure 4.5: UML classes for object package

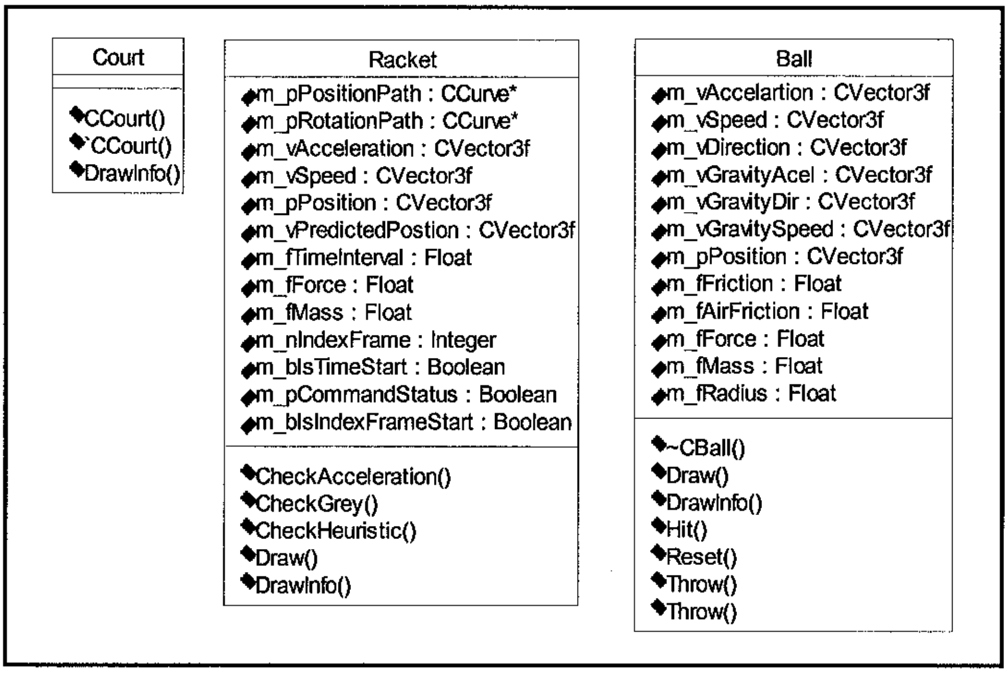


Figure 4.5 shows classes for object package. There are court, racket and ball, which are display in the squash simulation. The image for court and racket are retrieved from 3ds package while object are developed from Glut package.

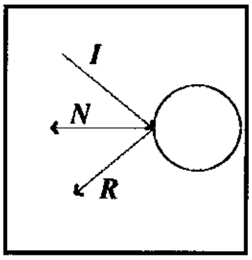
To simulate realistic movement of the ball, there are several factors that need to be considered. All of these factors determining the trajectory of the ball:

1) Collision Detection
2) Physically-based Modeling
   a. Collision response
   b. Moving under gravity

The first method of collision detection is to use bounding sphere around each object and to test whether these bounding volumes collide. After determining the collision point, physically-based modeling is taken into consideration.

The law of physics had been applied to determine responsiveness of the ball collides with the surface. When collision occurs, the ball will move in the opposite direction i.e., it bounces off. Figure 4.6 shows the collision response if the ball hits a surface.

Figure 4.6: Collision response of the ball



The new vector R is calculated as follows:

R = 2*(-I dot N)*N+1

Where:

R: is the new direction vector
I: is the old direction vector before the collision
N: is the Normal at the collision point

The angle of the new direction with the normal at the collision point is the same as the original direction vector.

The determined of the collision point and computed response is not adequate to move the ball in a realistic fashion. Therefore, physical laws also been considered for ball's movement. This is performed using Euler equations. Based on Euler equations, the velocity and position at each new time step is computed as follows:

New_Velocity = Old_Velocity + Acceleration * Time_Step
New_Position = Old_Position + New_Velocity * Time_Step

The acceleration of the ball is determined according to this equation:

Force = Mass * Acceleration

In this case, the only force the ball is gravity, which can be represented as a vector indicating as acceleration. It means that at the beginning of each time step, new_velocity been calculated and moved to test with collisions. If a collision occurs during time_step, the position of the ball increased, computed the new_velocity and move object for remaining time in order to test again for collision. This procedure gets repeated until the time_step is completed.

Figure 4.7: UML classes for graphic package

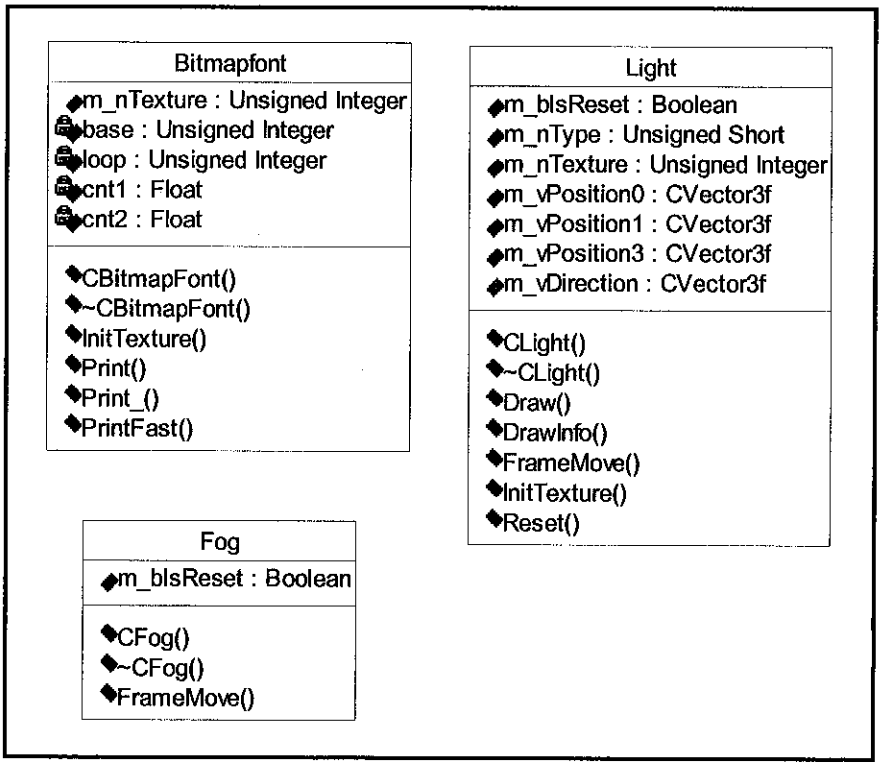


Figure 4.7 shows classes that are included in graphic package. Court has an impact on light and fog. Fog is an atmospheric effect that adds haziness to objects in squash simulation, which is usually a relation of how far away the objects in the scene are from the viewer. Meanwhile bitmap font has an effect on text functionalities display on the system.

Figure 4.8: UML classes for control package

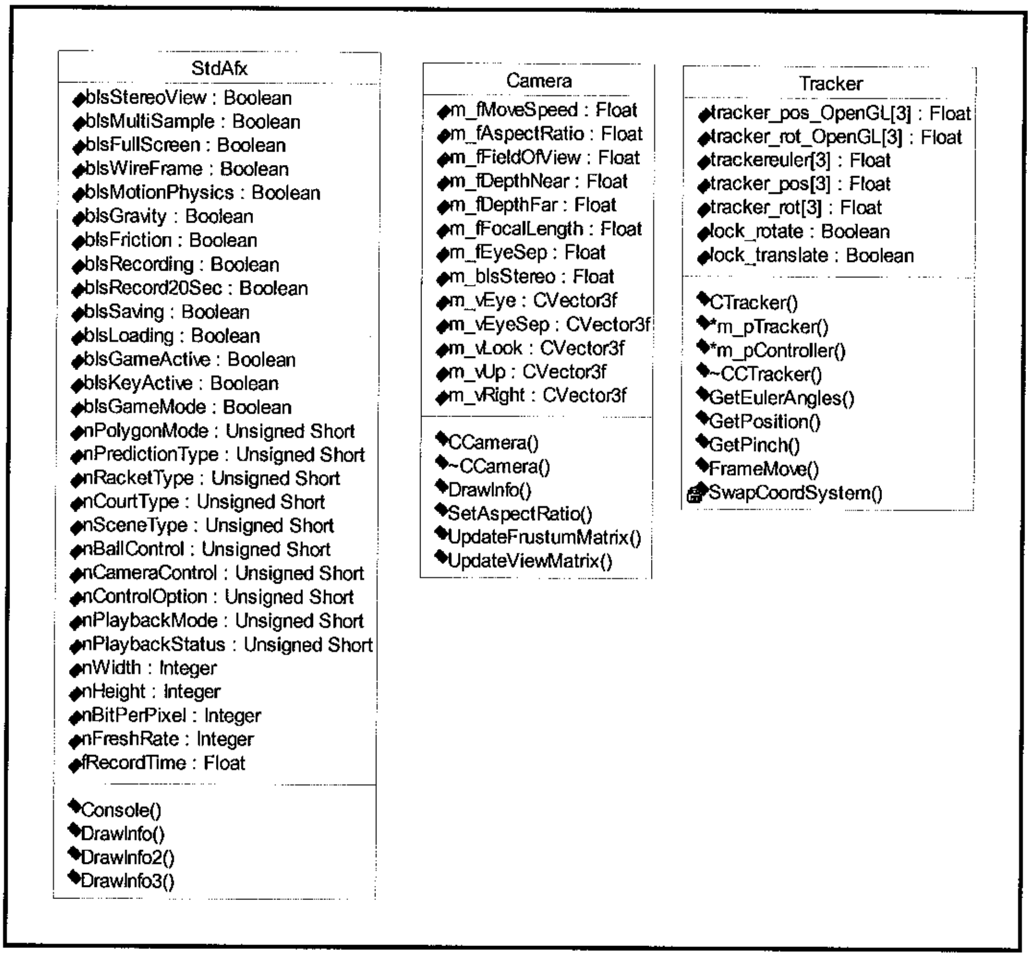


Figure 4.8 shows classes for control package. The player able to control camera and tracker. Camera is for the player to change their view. In squash simulation, player will give an option either to move their racket by using keyboard or tracker. However, for experiment and data collection, keyboard was an instrument taken for consideration. All packages in this system link to stdafx class.

Figure 4.9: UML classes for mathematic package

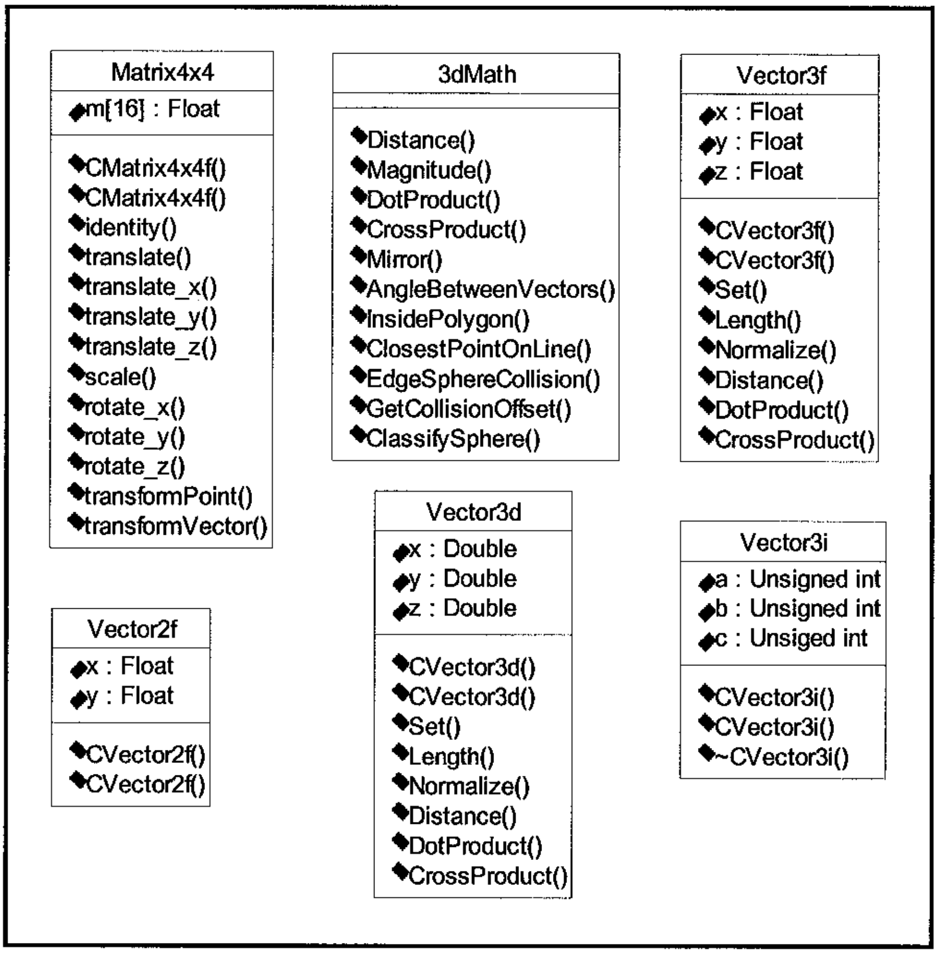


Figure 4.9 shows classes for mathematic package. All of the classes are the basic mathematic components in designing and developing graphic application.

Figure 4.10: UML classes for record package

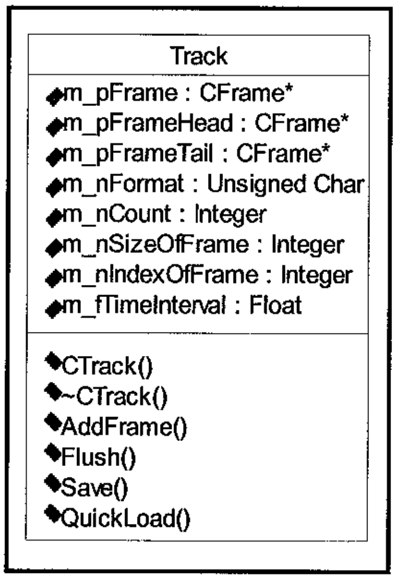


Figure 4.10 shows track class. The class used to track and record movement of the racket.

Figure 4.11: UML classes for view package

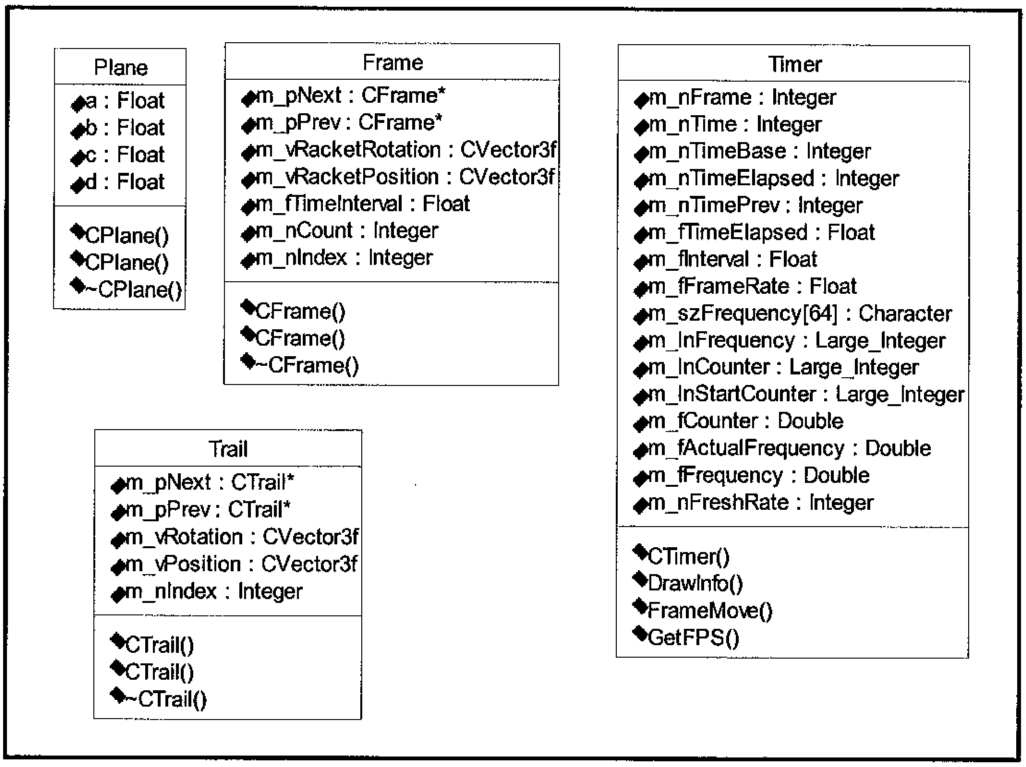


Figure 4.11 shows classes for view package. Classes under view influence the performance and display of squash simulation.

Figure 4.12: UML class diagram for squash simulation

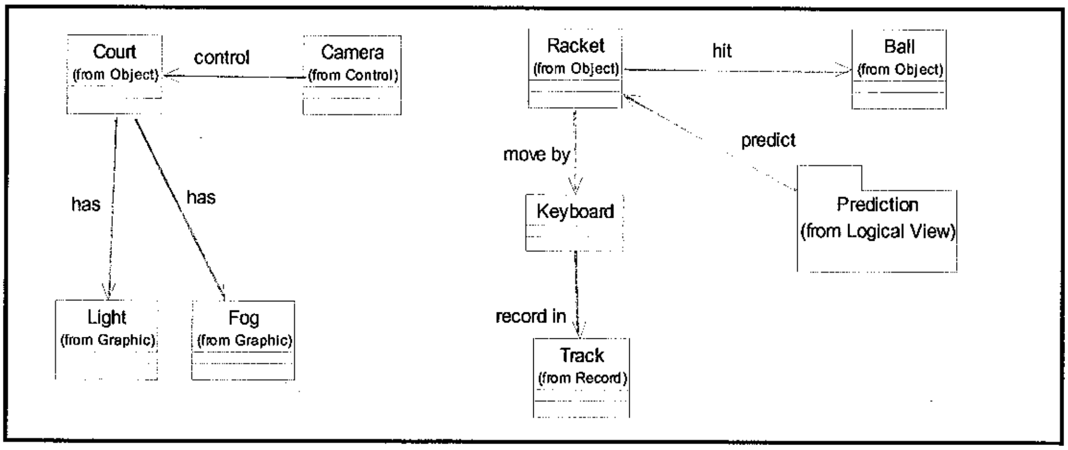


Figure 4.12 shows the relationship between entities in squash simulation. In squash simulation racket will hit the ball. Player moves the racket by using keyboard. The movement of the racket can be predicted thru prediction architecture, which consists of 1) grey system prediction and 2) heuristic-based prediction. Finally the movements of the racket recorded in track class. Meanwhile squash court controlled by the camera. Player can increase their view by camera class. Light and fog has effect on the court.

4.2.3 Implementation Design

Figure 4.13: Screen shot of enter mode

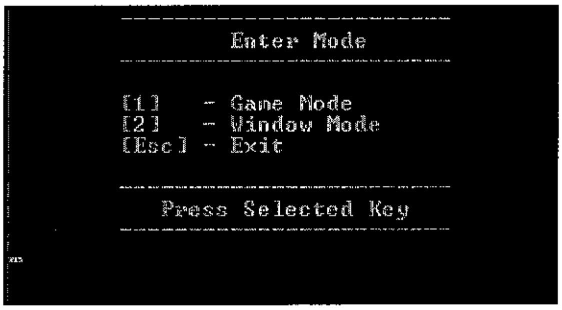


Figure 4.14: Screen shot of game mode

Figure 4.13 and 4.14 show snap shots of mode that will be chosen by the player before starting their game. For the game mode, player will have an opportunity to choose the sample rate and windows size. Each sample rate will give different performance of the display. For the game mode, the display is in stereoscopic view.

Meanwhile for the window mode, the sample rate had been set to 60Hz, which is a standard measurement. Player will enable the stereoscopic view manually. Both game mode and window mode have the same functionalities. However, during the game mode, player has advantages to choose their sample rate.

Figure 4.15: Screenshots: squash simulation scene

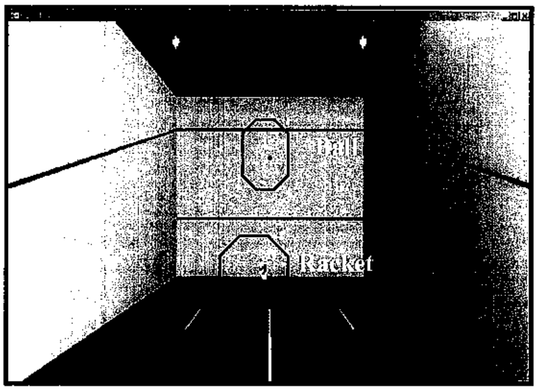


Figure 4.15 depicts the overview of the developed prototype system for virtual sports training, in which a player can play virtual squash. The player is required to hit a computer-controlled virtual ball instead of hitting an actual squash ball.

Table 4.1: Manual and functionalities of squash simulation

DESCRIPTION	It is an implementation of the squash game for a desktop VR environment.
AIM OF THE GAME	With the racket, the player hit the ball. The game is over when the player unable to hit the ball.
KEY	**FUNCTIONALITIES (The following keyboard inputs can be used)**
F1	Show settings
F2	Show data information
F5	Pause
F6	Record
F7	Record 20 second
F9	Save
ESC	Quit
F	Full screen
H	Intuitive control information

P	Reset ball
A, C, D, S, X, W	Camera control
I, J, K, L, M, N ARROW KEYS, HOME & END	Racket control
0	Turn friction on/off
1	Turn court's wire frame on/off
2	Turn Light on/off
3	Change racket
4	Change court
5	Control type
6	Turn motion physics on/off
7	Change prediction type
8	Turn stereo view on/off
9	Turn gravity on/off

Table 4.1 shows manual and functionalities of a squash simulation. The virtual model can be translated, rotated and scaled by the keyboard. The intuitive control provided, enhance the effectiveness of squash system. Hence, user can take various controls as a choice while they playing the game.

4.3 Use Case Diagram

This section describes interaction between player and squash simulation. Use case describes the system functions from the perspective of the player and in the manner and terminology in which the player understand. The creation of use case has proven to be an excellent technique in order to better understand and document system requirements. To further understanding on the system, Figure 4.17 shows the flowchart of squash simulation. The use case model diagram for the player is shown in Figure 4.16.

Figure 4.16: Squash simulation use case diagram

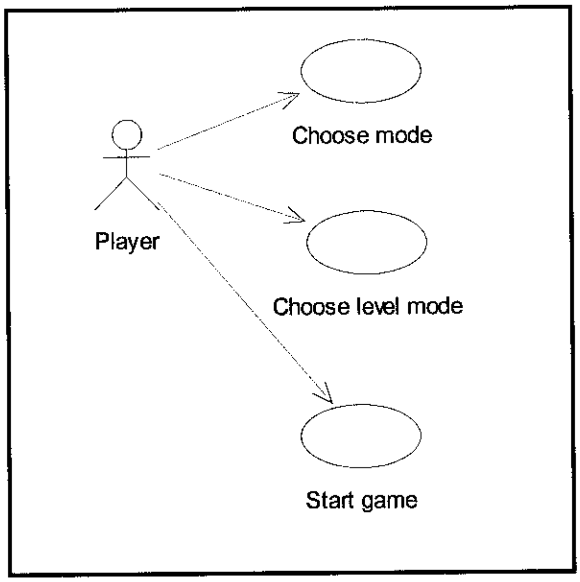
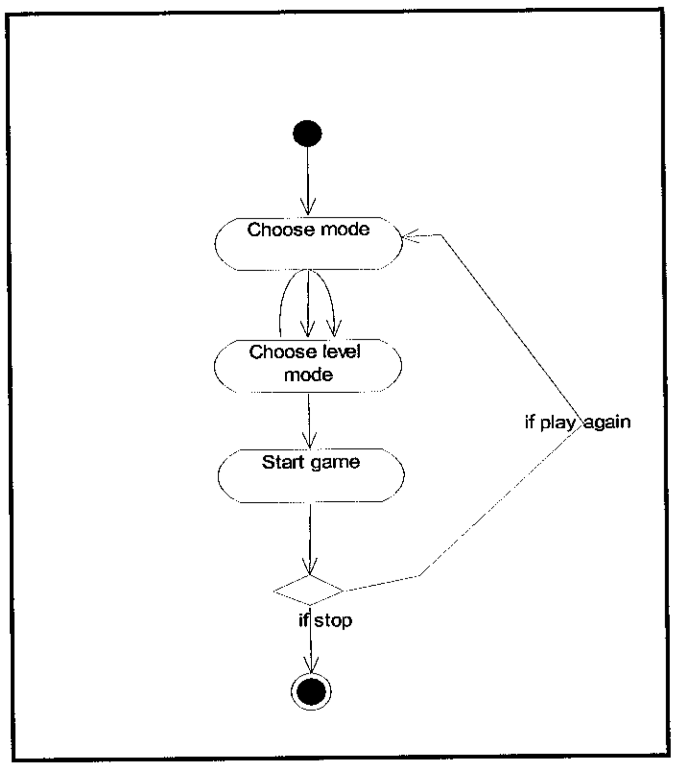


Figure 4.17: Squash simulation-activity diagram



Based on Figure 4.16 and 4.17, brief descriptions of the game process can be summarized as:

1. application starts
2. application display list of mode
3. player chooses one of the mode
4. application give chances for player to return to action number 2
5. application display two mode: 1) game mode and 2) window mode
6. if player choose game mode, then application display list of level mode
7. player choose level of mode in game mode
8. after player choose level mode and window mode
9. application start game
10. if player fail to hit the ball, then application return to action number 2

4.4 Summary

In Chapter One and Chapter Two, the research has dealt with the virtual sports training system as a useful application of VR. The extensions to the UML discussed above have been implemented and used in an application for recording and displaying user input and user data obtained from monitor and keyboard. Especially focusing on playing squash, the researcher has developed the system to be integrated with prediction algorithm. The motion area of squash simulation has shown and collision detection method for virtual objects has been described. Modeling of squash simulation can be summarized in table 4.2.

Table 4.2: Phases involved in VR application development and activities available

PHASES		PLANNING
Planning	1	Identified scene or situation of the application
	2	Perform sketches of the scene
	3	Determine objects, structure and texture of objects
	4	Confirm editing tools to be used
Analysis	1	Determine location and position of the object
	2	Select colors and determine suitability
	3	Determine types of interaction
	4	Analyze types of animations
	5	Determine method of elements of integration and method of integration
Design	1	Object design
	2	Interaction design and programming interaction
	3	Background design
Implementation	1	Integrate elements in one environment
	2	Interface setting
	3	Enhancement
Support	1	Debugging

Moreover, the performance of squash simulation will be demonstrated through experimental results, which will be explained in Chapter Five. Next chapter, the researcher will discuss how pilot study will be conducted.

CHAPTER FIVE
USER ACCEPTANCE STUDY & EXPERIMENTAL STUDY

5.0 Overview

This chapter discusses in detail user acceptance study and experimental study. User acceptance study provides an opportunity to work out any problems that arise before squash simulation is put into real experimentation. In order to verify the validity between heuristic-based prediction, Grey system prediction and non-prediction, experiment study was conducted to determine which algorithm is preferable for improving motion tracking in VR systems.

5.1 Subject

In user acceptance study and experimental study, 34 volunteer subjects were invited to participate. All of the participants invited from Information & Communication Technology Department, Universiti Teknologi PETRONAS. 17 subjects were male and 17 female. The subjects ranged in age from 20-22 years old. All the subjects reported to have at least a basic knowledge on computers. 17 of the subjects were VE novice users and 20 of the subjects had a background in squash game. 5 of the subjects reported not to play computer games at all and only 17 subjects played computer or video games more often than every other days.

It is widely accepted that differences among subject has a profound effect on task performance (Jeffries *et al.,* 1991). Identifying these differences and their implications is important if VEs are to effectively accommodate many types of subjects. Subject's experience has been shown to have a direct impact on subject skills and abilities normally associated with task performance. Subject experience also affects the manner in which subjects understand and organize task information. A subject who was new to VEs may be able to apply traditional computer experiences within the VE to improve task performance (e.g., working with virtual object). However, direct VE experience gives a subject familiarity with VE-specific issues such as field of view, suspension of belief, stereoscopic vision, and even motion sickness. Furthermore, subject who has an experience in squash game may be able to apply their skills and knowledge during the experiment. Meanwhile subjects who played computer or video games able to control keyboard quickly.

5.2 Experimental Set-up

Figure 5.1: Photo of virtual squash experimental system: 1) monitor, 2) keyboard, 3) Liquid Crystal Eye Shutter Glasses and 4) subject

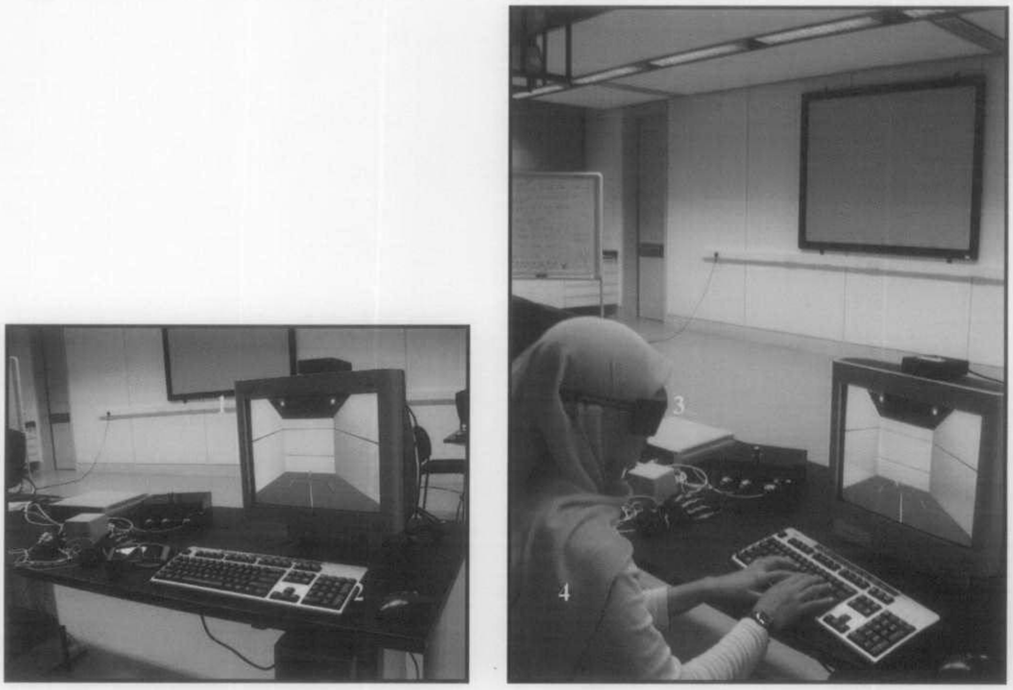


Figure 5.1 shows the snapshot of virtual squash experimental system. Both user acceptance study and experimental study conducted under this set-up. Graphic rendering is shown through a desktop monitor on an AMD K611 2.66GHz with 512 MB of main memory. The experiments ran as a windowed application. The squash simulation used Liquid Crystal Eye Shutter Glasses (Crystal Eye®) that enable stereovision and keyboard for object manipulation. The Crystal Eye® lenses are electronically controlled to close off one eye or the other at the same time as the screen shows alternatively the left or right image. This occurs 120 times per second so

that little flickering is noticed (Blade et al., 2002). The effect is that the subject perceives depth in the screen. A synchronization signal is sent from the computer to

the glasses by infrared so that no wires are needed. Within the squash simulation scenario, the player is able to interact with the virtual objects (racket and ball) in an intuitive way.

5.3 User Acceptance Study

A user acceptance study was conducted before the experimental study. The purpose of user acceptance study was to test and evaluate in order to verify that squash simulation can be deployed to the experimental study. Squash simulation was tested against performance behavior and usability applying objective and subjective measures. It has a significant influence on the degree of acceptance of the VR system by the potential subject. A carefully designed, task and user oriented human computer interface can essentially contribute to generate a high initial degree of acceptance (Sutcliffe & Kaur, 2000). In addition to usability testing, it was assessed how well the squash simulation could be used to evaluate the design of the system and its features. Nielsen (1993) identifies the following stages of a usability test:

1) preparation : setup of testbed environment,
2) introduction : explanation and sample run to get familiar with the test scenario,
3) testing : carrying-out of usability tests with test persons and
4) debriefing : fill in of questionnaires including personal comments by test persons.

5.3.1 Procedure

Subject took their place and made him/she feel comfortable. The researcher was in the same room as the subject. A welcome speech and informed consent given to subjects as an introduction. Subject was asked to answer pre-test questionnaire and presence questionnaire.

The purpose of the pre-test questionnaire is to address specific test objectives such as subject's first impression of subject interface design concept. After answering the pre-test questionnaire, pre-requisite training was given to the subject about the equipment and software that would be used for the testing. This helps the subject feel comfortable with the hardware provided. After each subject was introduced to either VE, he/she was allowed a five-minute to familiarize him or herself with the technology. After the pre-requisite training, subject was asked to read the test instructions and the list of tasks that they needed to perform during the test.

The orientation session was to inform subjects about the testing procedure. After the reading period, the researcher then gave a short introduction to explain the purpose of the test, the need of the test and what was expected from the subject. The subject was informed that the test was to evaluate the software and not the subject. The subject was also informed that the subject was being observed. Lastly the test monitor opened the question and answer session to the participants for clarification.

The subject carried out a series of tasks in the performance test. The study domain was a training task involving playing a squash game. In order to get realistic test results, the subject is test on pair basis rather than individuals. The two subjects can learn the system together. While one team member completed the task, the other team member acted as an observer recording the behavior and usability problems of their fellow team member. On completion of the task the team roles were reversed. Their discussion shows their idea and reveals their growing knowledge as well as their misunderstanding. In each condition researcher gave the subject hints when they noticed hesitations or subjects asked for help. When both team members had completed the task, each subject was asked to complete the presence assessment questionnaire based on Witmer and Singer (1998) relating to the VE they had just used.

Finally at the end of the study, each subject was questioned during a short briefing about the problems encountered during their interaction and about their experience in the squash simulation.

5.3.2 Questionnaire

Researcher used closed-ended questionnaire for user acceptance study testing. Questionnaire is an efficient data collection mechanism because the researcher knows exactly what is required and how to measure the variables of interest (Sekaran, 2003). Closed-ended questions asked the subjects to make choices among a set of alternatives give. All items used in questionnaire are Likert scale as it is commonly used for analyzing behavior scale. All items in questionnaire tapped on a 5-point scale. Research indicates that a 5-point scale is just as good as any (Sekaran, 2003).

The 5-point Likert scale that was used in this research is as follows:

5 point	= strongly agree / very superior
4 point	= agree / superior
3 point	= neither disagree nor agree / neither inferior nor superior
2 point	= disagree / inferior
1 point	= strongly disagree / very inferior

Questionnaire for pre-test consists of 7 questions, which can be categorized into three categories:

1. Category 1: make object easy to distinguish,
2. Category 2: make object easy to identify and
3. Category 3: make purpose of action clear.

Scale category listed in table 5.1

Table 5.1: Scale Category for pre-test questionnaire

Category	Items
Category 1: make object easy to distinguish	1, 2
Category 2: make object easy to identify	3, 4
Category 3: make purpose of action clear	5, 6, 7

The questions asked during user acceptance study are listed in the Appendix 1.

5.3.3 Results

The data collected on forms was analyzed. Results of the pre-test questionnaire and presence are presented. Results from questionnaire validate result that squash simulation ready for experiment.

5.3.3.1 Pre-test

Table 5.2: Results for pre-test questionnaire

Category	Percentage%
1	98.82%
2	98.82%
3	98.82%

Figure 5.2: Graphical output for pre-test questionnaire

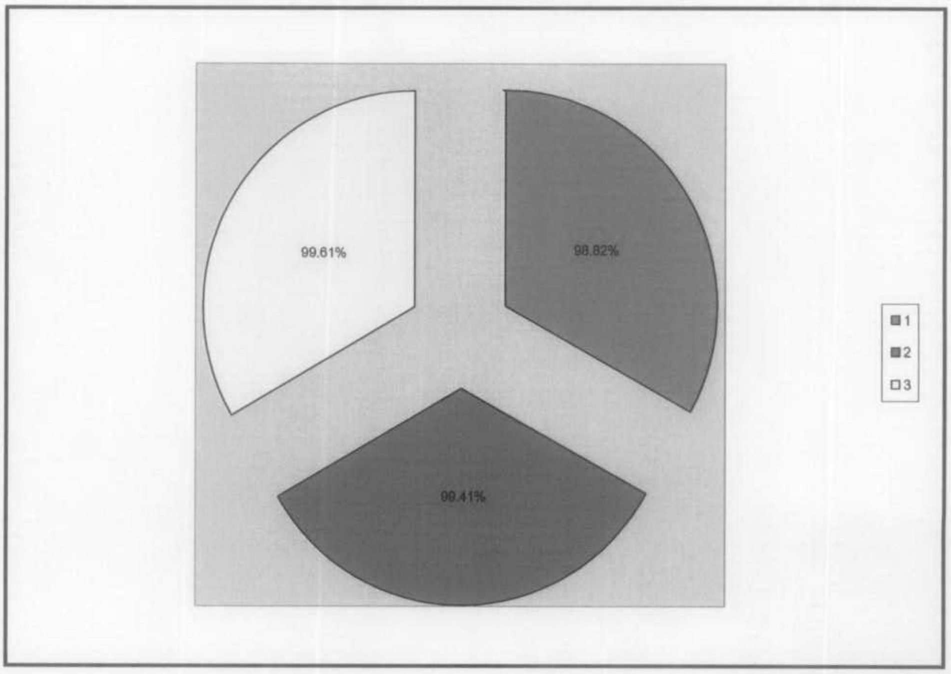


Table 5.2 and Figure 5.2 shows result from pre-test questionnaire. The results show that subject first impression on the squash simulation is satisfied. Subject can clearly identified objects in the screen and understand the purpose of each action.

5.3.3.2 Presence

Table 5.3: Statistical Analysis for presence

Number	Presence	Total	Percentage
1	Ability to control system	50	29.41%
2	Responsiveness	155	91.18%
3	Naturalness of interaction	160	94.12%
4	Naturalness of control	160	94.12%
5	Real world awareness	165	97.06%
6	Control awareness	167	98.24%
7	Sense of object movement	165	97.06%
8	Consistency with real world	165	97.06%
9	Anticipate action results	160	94.12%
10	Ability to search	166	100%
11	Sense of self movement	168	98.82%
12	Object examination	165	100%
13	Different viewpoints	170	100%
14	Object manipulation	170	100%
15	Involvement	164	96.47%
16	Action outcome delay	170	100%
17	Subject adjustment	100	58.83%
18	Subject proficiency	164	96.47%
19	Subject involvement	160	94.12%
20	Perspective	165	97.06%

Sense of presence is important since the researcher wants to maximize the subject's enjoyment and make them feel they are actually being part of the displayed environment. For a squash simulation, the researcher found that the performance of the player decreases, when audio is disabled. Specifically, when the audio is disabled player will experienced a state of confusion. The players are unable to:

1) identify the movement of the ball and
2) ensure that the ball being hit by the racket.

Table 5.3 shows total and percentage of presence questionnaire. The lowest percentage is ability to control system. Subjects were frustrated by the control and display gain on the keyboard mainly because they were used to playing the game with a much higher mouse sensitivity level. Therefore, subjects have difficulty in manipulating the keyboard to hit the ball. Overall, subject responses were positive.

Results from pre-test questionnaire and presence questionnaire fulfilled the main objective of user acceptance study. The user acceptance study showed no needs for any major changes in the test procedure. Therefore the squash simulation is ready to be used for experimental study.

5.4 Experimental Study

In this section, some basic experimental procedure and evaluation method are demonstrated to verify performance of the player with proposed prediction algorithm.

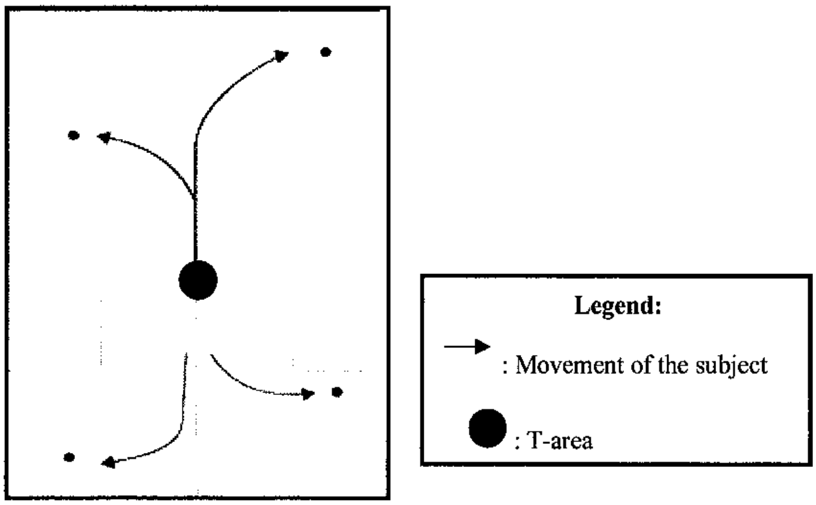
5.4.1 Test Procedure

Since the researcher wants to evaluate performance in a VR context, motion data of a typical VR tasks are needed. For this purpose, data will have to be collected from each subject and the unit analysis is the individual. Previous research has shown algorithm parameter tuning play an important role in determining prediction accuracy. For each test scenario, the previous researchers perform a parameter search routine to find the best parameter setting for that particular configuration. This approach is not ideal from a practical standpoint since it is difficult problem to know exactly what the user's motion will be in real application. After all subjects of VE are human beings. Therefore, the researcher chose to perform analysis in real tasks. Note that also, the experiments involved for doing no filtering of noise at all.

Based on the results from the user acceptance study, experiments were planned and performed. Results from user acceptance study have shown it quite difficult to hit the ball with the keyboard. Because of this, the effect of the ball on the racket is unnoticeable and can be ignored without visually changing the movement of the racket. The racket however exerts a force on the ball. The easiest way to obtain experiment's motion data is through a series of experiments, where users are asked to perform some tasks while their movements are recorded. Each collected data contains pose records consisting of a timestamp, a position vector and a unit length quaternion. However because keyboard was used as an instrument, unit length quaternion data were ignored.

The researcher administered all the 136 main tests. The test was divided into two parts: 1) introduction and 2) playing the game with testing. In the introduction, each subject was trained to be familiar with the virtual squash for about 5 minutes before the experiment. Subjects were instructed to play squash game under 3 predictions across 4 sample rates. After a short briefing, pressing a specific key signaled the start of each game. Motions were recorded while subjects move from and return to T-position on the court (Figure 5.3). A beep signaled the end of the game. While subjects play the game, the computer automatically recorded the subjects' movements. Using the test scenarios, the researcher recorded the running times for each algorithm. This study is not interested in the task of completion times and success rates.

Figure 5.3: An example of subject's movements while playing squash



5.4.2 Evaluation Method

Comparison data is needed to determine how well the heuristic-based algorithm performs over Grey system algorithm and non-prediction performed. The testing provides a number of useful features. By setting specific parameter (i.e., sampling rate), subject can run a single test, which provides global and component wise results for a number of different metrics. The researcher took a sampling of 34 tests from the non-prediction, Grey system prediction and heuristic based prediction for different sample rates.

Sampling rate may render an otherwise well-designed VE very difficult to use by introducing lag into the system. Traditional computer device sampling rates are typically 10 to 100 Hz [1]. Interactive VEs require much higher sampling rates if they are to realistically model real-world interaction. Moreover, quick user inputs motions will not be captured accurately if sampling rates are too small.

In the experiment, the subjects were tested with sampling rates of 25, 60, 80 and 140Hz giving the researcher three different test scenarios for each prediction. The sampling rates were chosen because VR tracking systems are commonly run at these rates.

In general, determining the best way to compare different prediction algorithms is a difficult problem. In order to test research hypotheses and say something about the significance of empirical results, the researcher has applied statistical hypothesis tests. There are a variety of metrics that could be used, each one providing distinct insights into prediction performance (LaViola, 2003b). The most frequently used have been the t-test together with output from validation experiments (Sekaran, 2003). The t-test used is for two samples assuming unequal variance with α=0.05 criterion for rejection of the null hypothesis. The results of these tests will determine whether or not the hypotheses are substantiated. Comparison between heuristic-based prediction and non prediction also heuristic-based prediction and Grey system prediction on several datasets used pair wise t-tests on the experimental output from all applicable datasets in order to test how the classifiers compared overall.

The researcher also chose to use two of the most common error metrics including standard mean error and percent better. All of these metrics are computed for global position as well as the individual components of the position. With the standard mean error metric, the researcher shows how a prediction is performing on an average. The percent better error metric determines how predictions are performing relative to doing no prediction at all (i.e., using the previous pose from the tracking system as the predicted pose) by count the number of times the predicted pose is closer to the true pose rather than the previous pose and dividing by the total number of predictions made. Although the supported errors metrics provide only a small sample of the

---

[1] See: www.gigascale.org/pubs/416/T-Comp_Aug.03.pdf [February 2006]

possible ways of analyzing predictions both numerically and perceptually, there are a good starting point and provide a significant amount of information about prediction accuracy.

Measurement errors and other problems are also bound to introduce an element of bias or error in this findings (Sekaran, 2003). However, the researcher would like to design a research in a manner that ensures that the research findings are as close to reality as possible. Because of this, the researcher chose confidence interval to place reliance and confidence in the results. According to Sekaran (2003), confidence refers to the probability that the estimations are correct. That is, it is not merely enough to be precise, but it is also important that the researcher confidently claim that 95% of the time our results would be true and there is only a 5% chance of being wrong.

Based on the objectives described in Chapter One, the evaluation process for prediction algorithm usually focuses on four areas: 1) accurate, 2) fast, 3) robust and 4) simple to understand and implement. Prediction algorithm is expected to fulfill all of the criteria in order to prove that it performs better than the current available predictions. If heuristic-based algorithm satisfies all of the conditions, then it is reasonable to conclude that this theory is feasible and appropriate to be used in virtual training application. This is because, higher efficiency, more robust and higher accuracy contributed by the shorter computation time.

5.5 Summary

This chapter had discussed in detail procedure of user acceptance study and experimental study. The researcher has tested both studies by using a desktop-type experimental set-up. Results from user acceptance study lead to the planning of the experimental study. A user acceptance study performed has shown the feasibility of the squash simulation. The squash simulation provides testing procedures for examining predictions across different sampling rates. All datasets are sampled at 25, 60, 80 and 140Hz. A performance analysis was conducted using heuristic-based prediction, Grey system prediction and non-prediction. The sampling rate influences the performance of prediction algorithm. In order to determine the effectiveness of the algorithm proposed, statistical analysis will be conducted. Next chapter will discuss in detail results and analysis of the experiment.

CHAPTER SIX
RESULTS AND DISCUSSION

6.0 Overview

This section is devoted to analysis and discussion of the results obtained throughout the research. The researcher submitted the experimental data for computer analysis using Microsoft Excel-a menu driven software programme. Once the data are ready for analysis, the researcher is already to test the hypotheses already developed for the study.

6.1 Hypothesizing

In particular the researcher will examine the results of hypotheses testing based on one-tailed test. In statistical terms, the hypotheses to be tested can be expressed by:

1. $H1_0$: $\mu_H = \mu_G$

   $H1_A$: $\mu_H > \mu_G$

   Where:

   μ_H = Heuristic-based prediction

   μ_G = Grey system prediction

2. $H2_0$: $\mu_H = \mu_N$

   $H2_A$: $\mu_H > \mu_N$

   Where:

   μ_H = Heuristic-based prediction

   μ_N = Non-prediction

6.2 Experimental Results

Having thus formulated the null and alternate hypotheses, the statistical analysis will be applied, which would indicate whether or not support has been found for the alternate-that is that there is a performance of heuristic-based prediction is better as hypothesized. Analysis in this chapter based on Sekaran (2003).

6.2.1 Prediction's Accuracy

Figure 6.1: Prediction accuracy

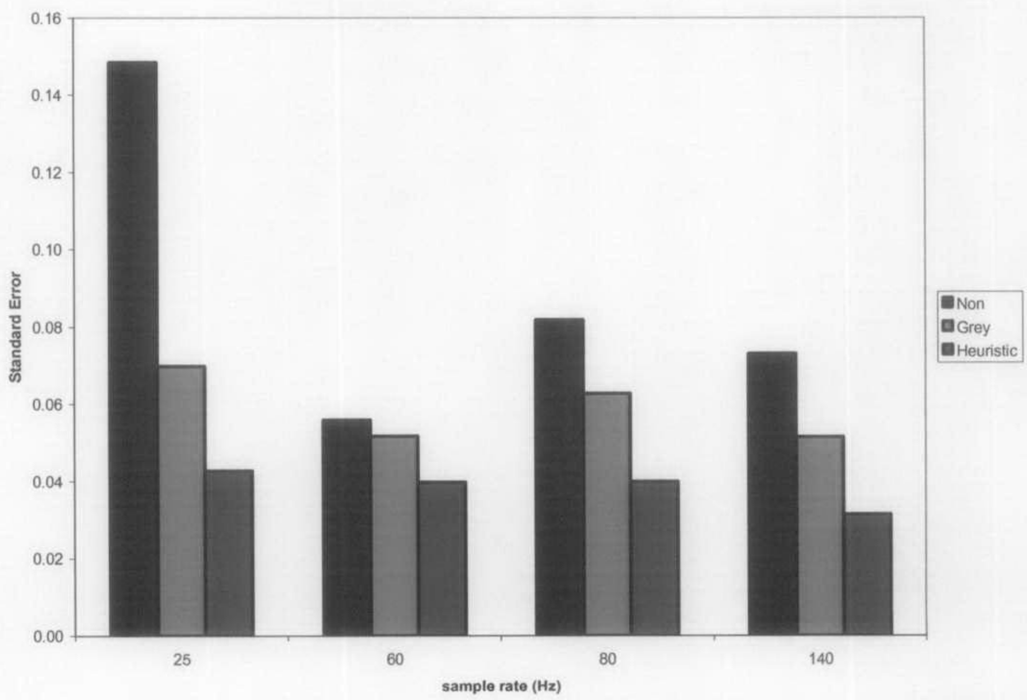


Figure 6.1 illustrates the results of the accuracy testing of each prediction in the developed squash simulation. The graph shows the relatively minute differences between the prediction accuracies for non-prediction, Grey system prediction and heuristic-based prediction. The majority of all the test runs, heuristic-based prediction performed better than non-prediction and Grey system prediction. These differences show an additional accuracy improvements obtained with the heuristic-based prediction.

Table 6.1: Statistical analysis with the Grey System Prediction and Heuristic Prediction

	Grey system prediction	Heuristic prediction
Sum	0.23	0.15
Mean	0.06	0.04
Variance	9.16667E-05	0.000025
Standard Deviation	0.01	0.005
Observations	4	4
Df	6	
t Stat	3.70	
T Critical one-tail	1.94	
P(T<=t) one-tail	0.005	
Confidence Interval (95%)	[0.052, 0.068]	[0.0351, 0.044]

Table 6.2: Statistical analysis with the Non-Prediction and Heuristic Prediction

	Non-prediction	Heuristic prediction
Sum	0.36	0.15
Mean	0.09	0.04
Variance	0.0016	0.000025
Standard Deviation	0.04	0.005
Observations	4	4
df	6	
t Stat	2.55	
T Critical one-tail	1.94	
P(T<=t) one-tail	0.02	
Confidence Interval (95%)	[0.05, 0.129]	[0.035, 0.044]

The results of the t-test done are shown in table 6.1 and 6.2. The t-test used is for two samples for means with α =0.05. In table 6.1, the t value of 3.70 is significant, so the null hypothesis is accepted. Therefore hypothesis 1 is substantiated and heuristic-based prediction is better than the Grey system prediction. Confidence interval for Grey system prediction is [0.052, 0.068] and [0.035, 0.044] for heuristic-based prediction.

As may be seen in table 6.2, the difference in the means of 0.09 and 0.04 with *t* value 2.55 for non-prediction and heuristic-based prediction is significant. Therefore, hypothesis 2 is substantiated.

Table 6.3: Performance of the Grey system prediction and Heuristic-based prediction in relation to no prediction at all

Grey system prediction (%)	Heuristic-based prediction (%)
63.88	41.66

To examine the relationship between the accuracy results for both predictions, the researcher looked at how many times better the given prediction performs than no prediction at all. This is easily calculated by taking the standard error.

Table 6.3 shows these "times better" metrics" for the Grey system better and heuristic-based prediction in relation to no prediction at all. The table shows that on average both prediction types perform better than no prediction at all for all cases, which first confirms previous results about Grey system prediction and second indicates heuristic-based prediction has better performance as the Grey system prediction. Additionally, the difference between Grey system prediction and heuristic-based prediction is larger than the 10% further indicating better performance of heuristic-based prediction.

6.2.2 Prediction's Efficiency

Figure 6.2, 6.3, 6.4, 6.5 and 6.6 show the running times performance for each subject across the different sampling rates. Results show that non-prediction takes the longest time between Grey system prediction and heuristic-based prediction. The sampling rate does have some influence on the performance of the prediction. The figures clearly shown that higher sample rate will reduce the time performance. Moreover, for each sample rate, heuristic-based prediction had the less running time.

Figure 6.2: Total time performance with 25Hz sample rate

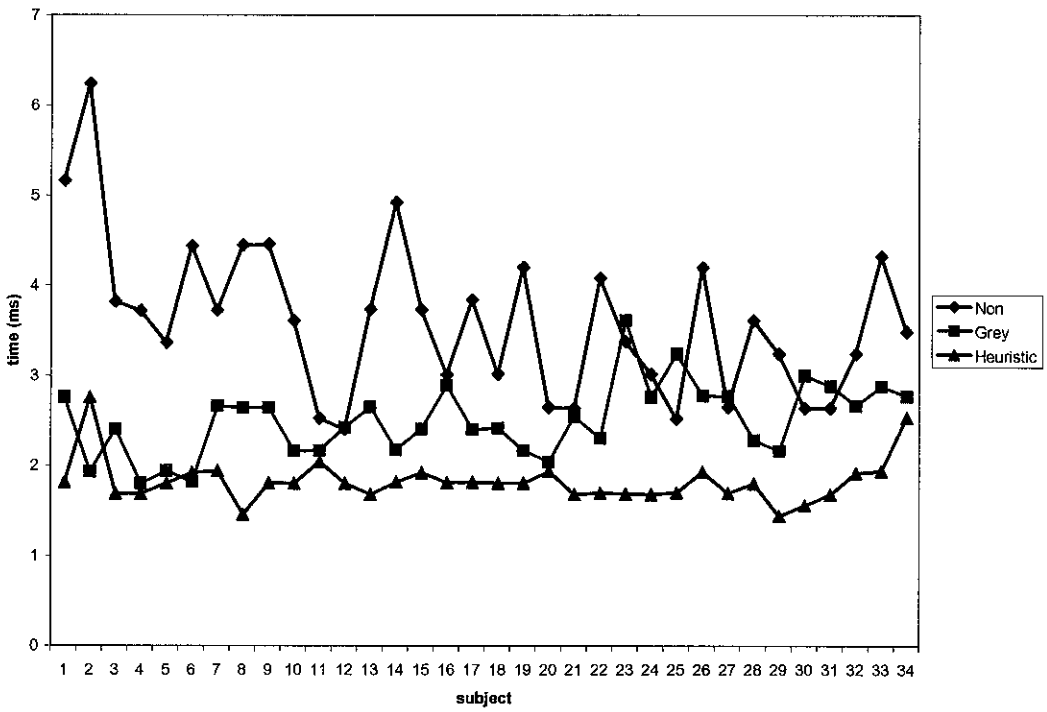


Figure 6.3: Total time performance with 60Hz sample rate

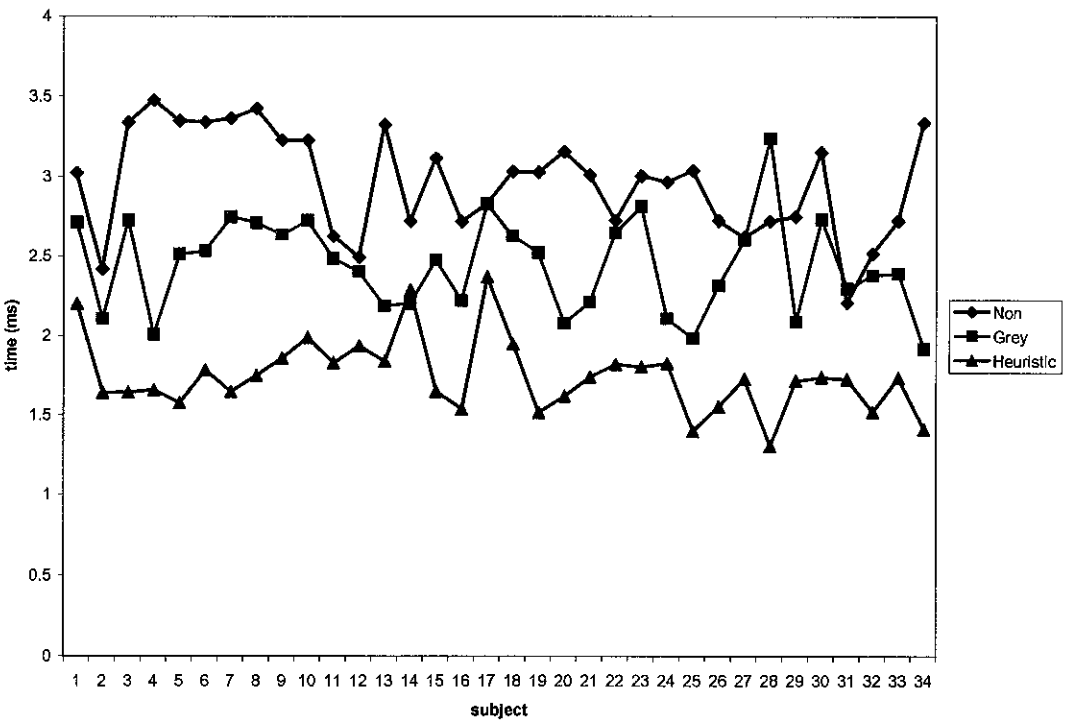


Figure 6.4: Total time performance with 80Hz sample rate

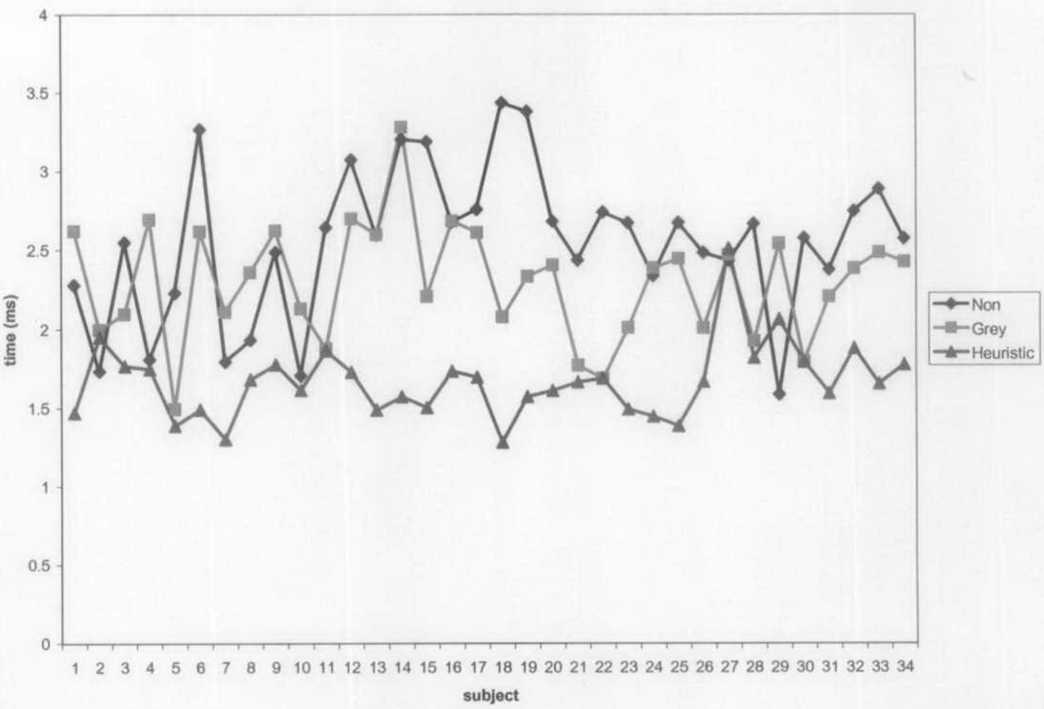


Figure 6.5: Total time performance with 140Hz sample rate

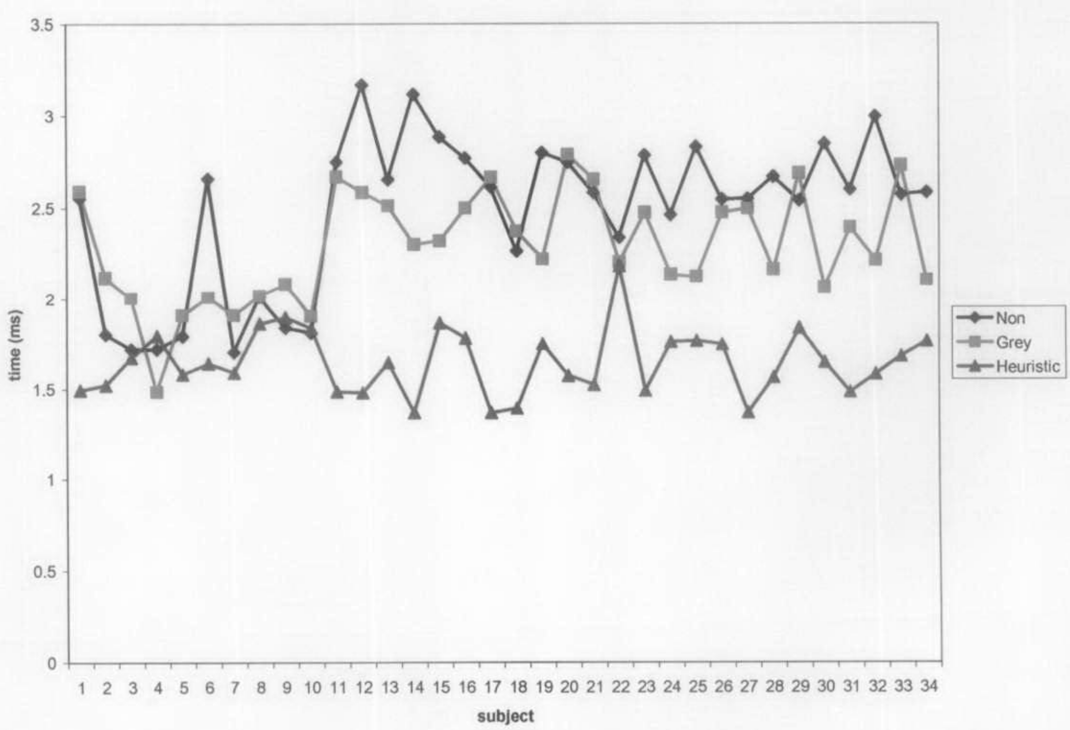
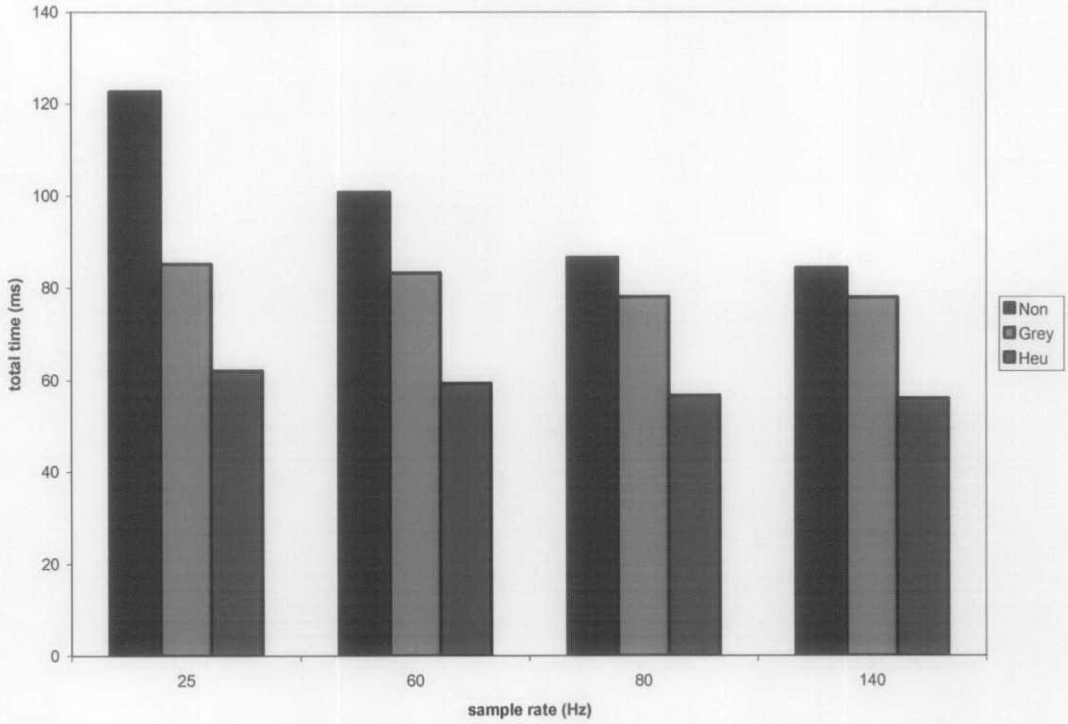


Figure 6.6: Total performance for different sample rate



In Figure 6.6, heuristic-based prediction reduced 40.65% in terms of running time performance between non-prediction and 27.83% between Grey system prediction. A t-test will indicate if the heuristic-based prediction is significantly better than the Grey non-prediction.

Table 6.4: Statistical analysis with sample rate 25Hz

	Grey system prediction	Heuristic prediction
Sum	85.12	62.02
Mean	2.50	1.82
Variance	0.17	0.06
Standard Deviation	0.41	0.25
Observations	34	34
df	66	
t Stat	8.30	
T Critical one-tail	1.67	
P(T<=t) one-tail	3.89778E-12	
Confidence Interval (95%)	[2.362, 2.637]	[1.737, 1.902]

Table 6.5: Statistical analysis with sample rate 25Hz

	Non-prediction	Heuristic prediction
Sum	122.69	62.02
Mean	3.61	1.82
Variance	0.75	0.06
Standard Deviation	0.87	0.25
Observations	34	34
df	66	
t Stat	11.60	
T Critical one-tail	1.67	
P(T<=t) one-tail	8.34642E-18	
Confidence Interval (95%)	[3.318, 3.902]	[1.737, 1.902]

Table 6.4 shows t-test results between Grey system prediction and heuristic-based prediction. The t value of 8.30 is significant, so the null hypothesis is rejected. Thus, hypothesis 1 is substantiated.

Table 6.5 shows t-test result between non-prediction and heuristic-based prediction. From the table, t value is 11.60; therefore null hypothesis can be rejected. Thus hypothesis 2 is substantiated.

Table 6.6: Statistical analysis with sample rate 60Hz

	Grey system prediction	Heuristic prediction
Sum	83.24	59.34
Mean	2.45	1.75
Variance	0.09	0.05
Standard Deviation	0.30	0.23
Observations	34	34
df	66	
t Stat	10.91	
T Critical one-tail	1.67	
P(T<=t) one-tail	1.78072E-16	
Confidence Interval (95%)	[2.352, 2.548]	[1.674, 1.826]

Table 6.7: Statistical analysis with sample rate 60Hz

	Non-prediction	Heuristic prediction
Sum	100.76	59.34
Mean	2.96	1.75
Variance	0.11	0.05
Standard Deviation	0.33	0.23
Observations	34	34
df	66	
t Stat	17.64	
T Critical one-tail	1.67	
P(T<=t) one-tail	4.09283E-27	
Confidence Interval (95%)	[2.85, 3.07]	[1.674, 1.826]

Table 6.6, t value of 10.91 is significant, so the null hypothesis is rejected. Therefore, hypothesis 1 is substantiated.

As may be seen in table 6.7, the difference in the means of 2.96 and 1.75 is significant. Thus, null hypothesis is rejected and hypothesis 2 has proven.

Table 6.8: Statistical analysis with sample rate 80Hz

	Grey system prediction	Heuristic prediction
Sum	78.02	56.65
Mean	2.29	1.67
Variance	0.13	0.05
Standard Deviation	0.37	0.23
Observations	34	34
df	66	
t Stat	8.52	
T Critical one-tail	1.67	
P(T<=t) one-tail	2.12258E-12	
Confidence Interval (95%)	[2.167, 2.413]	[1.592, 1.748]

Table 6.9: Statistical analysis with sample rate 80Hz

	Non-prediction	Heuristic prediction
Sum	86.59	56.65
Mean	2.55	1.67
Variance	0.23	0.05
Standard Deviation	0.48	0.23
Observations	34	34
df	66	
t Stat	9.69	
T Critical one-tail	1.67	
P(T<=t) one-tail	1.44629E-14	
Confidence Interval (95%)	[2.389, 2.711]	[1.592, 1.748]

Heuristic-based prediction reduces time performance of non-prediction by 34.57%. Meanwhile 27.39% time reducing while comparing with Grey system prediction. The results of the t-test are shown in Table 6.8 and 6.9.

In table 6.8, as t value is significant with value of 8.52, therefore, hypothesis 1 is substantiated.

In table 6.9, the t value is significant by 9.69. So, the null hypothesis is rejected. Thus, the hypothesis 2 is substantiated.

Table 6.10: Statistical analysis with sample rate 140Hz

	Grey system prediction	Heuristic prediction
Sum	77.88	55.99
Mean	2.29	1.65
Variance	0.09	0.03
Standard Deviation	0.30	0.18
Observations	34	34
df	66	
t Stat	10.67	
T Critical one-tail	1.67	
P(T<=t) one-tail	2.6E-16	
Confidence Interval (95%)	[2.19, 2.388]	[1.591, 1.708]

Table 6.11: Statistical analysis with sample rate 140Hz

	Non-prediction	Heuristic prediction
Sum	84.28	55.99
Mean	2.48	1.65
Variance	0.18	0.03
Standard Deviation	0.43	0.18
Observations	34	34
df	66	
t Stat	10.45	
T Critical one-tail	1.67	
P(T<=t) one-tail	6.15267E-16	
Confidence Interval (95%)	[2.336, 2.624]	[1.591, 1.708]

In table 6.10, null hypothesis rejected because t value is 10.67. Thus, hypothesis 1 is substantiated.

In table 6.11, null hypothesis is rejected as t value is 10.45. Therefore hypothesis 2 is substantiated.

Table 6.12: Statistical analysis for all sample rates

	Grey system prediction	Heuristic prediction
Sum	324.26	234.01
Mean	2.38	1.72
Variance	0.13	0.05
Standard Deviation	0.36	0.23
Observations	136	136
df	270	270
t Stat	18.18	
T Critical one-tail	1.65	
P(T<=t) one-tail	4.46873E-49	
Confidence Interval (95%)	[2.321, 2.438]	[1.681, 1.758]

Table 6.13: Statistical analysis for all sample rates

	Non-prediction	Heuristic prediction
Sum	394.32	234.01
Mean	2.90	1.72
Variance	0.51	0.05
Standard Deviation	0.72	0.23
Observations	136	136
df	270	270
t Stat	18.25	
T Critical one-tail	1.65	
P(T<=t) one-tail	2.56245E-49	
Confidence Interval (95%)	[2.781, 3.019]	[1.681, 1.758]

Table 6.12, shows results of the t-test between Grey system prediction and heuristic-based prediction. The t-test used is for two samples assuming equal variance. The *t* value of 18.18 is significant, so the null hypothesis is rejected. Thus, hypothesis 1 is substantiated. Therefore heuristic-based prediction performs better than the Grey system prediction.

As may be seen in table 6.13, the difference in the means of 2.90 and 1.72 with standard deviation of 0.72 and 0.23 for the non-prediction and heuristic-based prediction is significant. Thus, hypothesis 2 is substantiated.

6.2.3 Robustness of the Prediction

Figure 6.7: Frame rates for a different sample rate

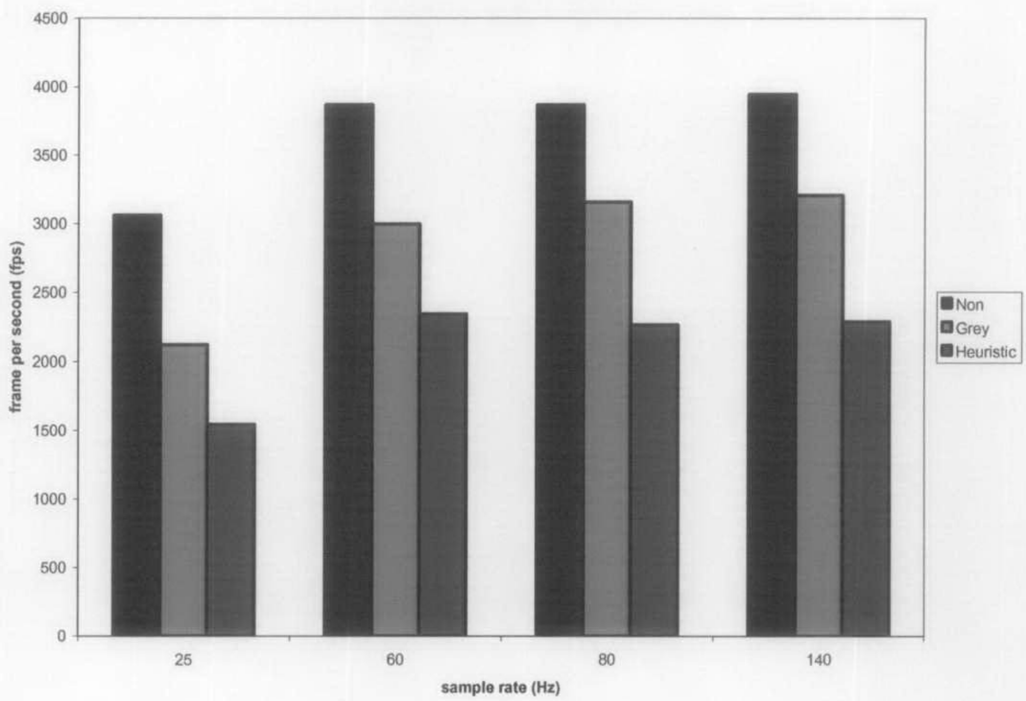


Figure 6.7 shows performance analysis of predictions, which is done on the rendering times. For this purpose, total frames per second across different predictions are calculated. A comparison of rendering types is illustrated in Figure 6.7. The performance results are highly reduced when sample rate 25Hz are applied.

Table 6.14: Statistical analysis for Frame Rate with the Grey System Prediction and Heuristic Prediction

	Grey system prediction	Heuristic prediction
Sum	11494	8445
Mean	84.51	62.10
Variance	330.10	160.47
Standard Deviation	18.17	12.67
Observations	136	136
df	270	
t Stat	11.80	
T Critical one-tail	1.65	
P(T<=t) one-tail	1.65504E-26	
Confidence Interval (95%)	[81.472, 87.548]	[59.98, 64.21]

Table 6.15: Statistical analysis for Frame Rate with the Non Prediction and Heuristic prediction

	Non-prediction	Heuristic prediction
Sum	14755	8445
Mean	108.49	62.10
Variance	487.47	160.47
Standard Deviation	22.08	12.67
Observations	136	136
df	270	
t Stat	21.26	
T Critical one-tail	1.65	
P(T<=t) one-tail	6.76229E-60	
Confidence Interval (95%)	[104.78, 112.2]	[59.98, 64.21]

In table 6.14, the lowest rendering times taken by heuristic-based prediction, showed the robustness of the proposed prediction. Null hypothesis rejected because t value is 11.80 with p value 1.65504E-26. Thus, hypothesis 1 is substantiated.

In table 6.15, null hypothesis rejected, as t value is 21.26. Therefore hypothesis 2 is substantiated.

6.3 Discussion Results

The results from this experiment provide empirical evidence showing that heuristic-based prediction significantly better than the non-prediction and Grey system prediction in the context of 34 subjects. From these results, the researcher sees that the proposed prediction has a satisfactory performance for estimating the motion of a squash player. The hypotheses results are consistent across the different sample rates and this is one of the heuristic-based prediction benefits. Sampling rate plays an important role in prediction accuracy since a higher sampling rate will improve a prediction's performance (LaViola, 2003b). The short running times of heuristic-based prediction makes it better suited to handle tracking systems with very high sampling rates. Heuristic will not has to skip tracker samples due to the prediction computation. Heuristic's simplicity makes it easier to implement than the Grey system and also make heuristic suitable for implementation on a microcontroller that could be incorporated directly into tracking hardware by assuming an accurate parameter setting model (LaViola, 2003c).

When running on an AMD K611 2.66 GHz computers in a configuration similar to the one described in Chapter Five, heuristic prediction processes at well over 140 Hz. Frame rate under 140Hz sample rate is the number of times per second that squash simulation executes its main loop of reading sensors, processing, and writing to renderers. Actual frames per second experienced by the subject are determined by the renderer used, the hardware it runs on, and the complexity of the environment.

The main purpose of the squash simulation is to provide as close a link between the sensors and the displays as possible. Any processing that must be performed before data is received from the sensors is reflected in the displays adds to latency and reduces the quality of the subject's experience. As such, only processing that leads to properly displaying to the subject his or her view of the environment should be added to the squash simulation. For example, though it may be tempting to give VE velocities and let world positions be executed in every frame by the virtual squash, this will reduce frame rate and increase latency, ultimately reducing the quality of the experience.

Although table 6.3 shows heuristic-based prediction has roughly better accuracy with the Grey system prediction, however, the t-test results in table 6.1 shows heuristic-based prediction is better than Grey system prediction. The smooth and accurate heuristic-based prediction estimation, which could in turn provide a more effective means for addressing other system latencies such as those in the rendering pipeline. The improved accuracy should also improve warping or image deflection.

Figure 6.2, 6.3, 6.4, 6.5 and 6.6 show the total results of the running time performance. From the results, heuristic-based prediction runs faster than the Grey system prediction. For a frame of reference, the researcher also timed how long it would take to simply take the previous user pose and use it as the predicted pose. These timings show that heuristic-based prediction does not take a much longer to predict a pose than simply using no prediction at all. The reason the Grey system algorithm takes significantly longer to make an estimate is because it has to handle all the complex computation. In heuristic implementation, the calculation is quite simple based on the motion library.

The robustness of heuristic-based prediction was explored by examining their restart counts. Figure 6.7 shows frame rates for a different sample rates. The performance analysis calculated based on total count for all the predictions across all the sample rates. Heuristic-based prediction consistently restarts the least frequently. These results suggest that heuristic-based prediction is the most robust algorithm compared to others.

Results have clearly indicated that heuristic-based prediction much better than Grey system prediction. Furthermore, all p values almost to 0. In terms of confidence interval, the greater the precision and confidence the researcher aim in heuristic-based prediction, the more scientific is the investigation and the more useful are the results. Length of confidence interval for heuristic-based prediction is small under all conditions. These results suggest that heuristic-based prediction has relatively solved the latency problem in virtual training application for the experimental scenario. This has proven the hypotheses:

1. $H1_0$: $\mu_H = \mu_G$

   $H1_A$: $\mu_H > \mu_G$

   Where:

   μ_H = Heuristic-based prediction

   μ_G = Grey system prediction

2. $H2_0$: $\mu_H = \mu_N$

   $H2_A$: $\mu_H > \mu_N$

   Where:

   μ_H = Heuristic-based prediction

   μ_N = Non-prediction

Furthermore, experimental results demonstrate that heuristic-based prediction is capable of achieving the high-speed performance enough to implement sports training.

6.4 Summary

In this chapter, the researcher has compared the effectiveness of three types of predictions, namely non-prediction, Grey system prediction and heuristic-based prediction. Using squash simulation, the validity of heuristic-based prediction was confirmed and the required performances such as prediction's efficiency, accuracy and robustness were evaluated. This finding has rejected research hypotheses on non-prediction and Grey system prediction perform better than the heuristic-based prediction. Heuristic-based prediction has been shown to perform better significantly where all the p values almost to 0 in virtual displays. In terms of efficiency, heuristic-based prediction has reduced the performance time by 27.83% compared to Grey system, increased accuracy by 34.78% and increased the robustness of heuristic-based prediction by 26.52%. It results in fewer lags than the Grey system prediction designed by Wu and Ouhyoung (Wu & Ouhyoung, 1994). Latency cannot be eliminated without prior information about future motion movements. The heuristic-based prediction is accurate, fast, robust, and the researcher believes it can be used to improve the performance of a wide variety of commercial and custom tracking system.

CHAPTER SEVEN
CONCLUSION & FUTURE RECOMMENDATIONS

7.0 Overview

This research examined the latency problem for virtual training applications and in particular, the problem of designing prediction algorithm. The method presented in this thesis may change in the way people think about minimizing latency for dynamic motion training in VE. A novel prediction algorithm using heuristic-based for predicting a user's motion has been proposed. The suggested algorithm used simple kinematics approach based on the habitual pattern of motion and simple human rule in order to predict player's motions.

This chapter presents the closing remarks of this research. Section 7.2 provides a recapitulation of each chapter. In the third section, the researcher discusses how this research can contribute to the development of theory, methods and practice in this field. Section 7.4 offers a discussion of the limitations of the research. This research close with suggestions on the direction for future research.

7.1 Conclusion

The results of this research indicate that the objectives of this research have been met. However, the researcher making this claim in the context of 34 subjects. The objectives of this research are:

1) To develop heuristic-based prediction to minimize latency problem for dynamic motion in virtual training application.

In Chapter Two, the researcher presented an overview of the research fields related to this research. The researcher has surveyed three fields: (1) latency issues, (2) previous prediction algorithms, and (3) heuristic. The researcher reviews the literature directly related and necessary to the development of this work.

To overcome the limitations of previous predictions as discussed in Chapter One and Chapter Two, this research proposed heuristic-based prediction. In Chapter Three, heuristic-based algorithm applied space-time optimization model, which is one of the cornerstones of the physically based motion transformation system by Popovic and Witkins (1999). Heuristic-based algorithm used space-time optimization as its core because space-time both maintains the dynamic integrity of motion and provides intuitive motion control.

Dynamic properties of motion are taken into the account by constructing the physically based space-time constraints formulation of motion. The resulting change in motion is, consequently, transformed back into the space of the input motion in order to produce the final animation. The patterns of the squash game, which had been implemented in heuristic-based theory algorithm, based on several strategies that have been well tested several times. The strategies are theoretical and empirical materials, motion capture also biomechanical experiment. The movement patterns are stored in human memory and are utilized as templates for motion planning. The researcher treats the motion data as trajectory curves in a high dimensional space and doing a novel application of a Bezier curve and B-spline, which are typically used for planar curves, to human motion data. Once the space-time motion model was constructed, appropriate simple human rule was applied. For instance, it is not possible to move out of the range of the player's body and to rotate at any angle.

2) To develop squash simulation, which will be integrated with prediction algorithm.

In Chapter Four, squash simulation was developed to verify the usefulness of heuristic-based prediction. The squash simulation developed in this research is a portable OpenGL. In any modeling of human motion, articulated structure of human body poses a difficult problem (Perse et al., 2005). The full and detailed body would be too complex for the application. Therefore squash simulation represents player as a simple motion model with constant velocity and acceleration. The UML architecture of squash simulation was analyzed and designed using Rational Rose 2002® software.

Predictions are implemented in squash simulation because very often in virtually coupled applications, a motion has to be predicted well in advance in order to overcome latency that is introduced by the image generating system. Prediction algorithm provides next predicted player position, which then improves chances of finding the player on the next frame. The potential of prediction algorithm to accurately predict future positions eases the need to shorten the processing pipeline because a fixed amount of "lead time" can be allocated to each process. For example, the positions fed to the rendering process can reflect sensor measurements one frame ahead of time so that when the image is rendered and displayed, the effect of synchrony is achieved. Consequently, unpredictable system is invisible to the user as long as they are shorter than the allotted lead-time.

The implementation of heuristic-based algorithm is a relatively straightforward effort thus potentially in reducing the complexity of designing and developing virtual training applications compared with Grey system algorithm. It can be easily used and programmed using available software.

3) To evaluate performance of heuristic-based prediction.

In Chapter Five, both user acceptance study and experimental study implemented in a desktop VR environment. The player can see the virtual court, the ball and the player itself, which represented by a simple racket in the virtual world through Crystal Eye®. The researcher has performed a user acceptance study to verify that squash simulation can be deployed to the user. The study comprised pre-test questionnaire and presence questionnaire. Based on the results from the user acceptance study, which satisfied the subjects, an experimental study was performed.

To investigate the performance of the proposed algorithm, a basic experiment was executed. The experimental study aims to show that suggested algorithm perform better than the current prediction. The experiments instructed subjects to play squash game across different sample rates for each prediction. Motions of the subjects are recorded while subjects move from and return to T-position on the court. Analysis methods involved during experimental study are t-test, confidence interval (95%), standard mean error and percent better.

In Chapter Six, performance of suggested algorithm compared with a Grey system algorithm (1994), which represents as a first benchmark and non-prediction as second benchmark had been examined through purely empirical basis and analysis. The results proved that heuristic-based prediction is consistent across different experimental cases.

In terms of efficiency, the results from the experimental scenario provide empirical evidence showing that the heuristic-based prediction was significantly faster than the Grey system prediction with p value almost to 0. Heuristic-based prediction has reduced performance time by 27.83% compared to Grey system prediction. Furthermore, the confidence interval (95%) for heuristic-based prediction is [1.681, 1.758] and Grey system prediction is [2.321, 2.438]. For the second benchmark heuristic-based prediction had reduced time performance by 40.65% compared to non-prediction.

In terms of accuracy, heuristic-based prediction has improved by 34.78% compared to Grey system prediction. Heuristic-based prediction is significant with p value 0.005 and confidence interval (95%) of [0.0351, 0.044]. In addition heuristic-based prediction has improved the accuracy by 58.33% compared to non-prediction.

It also shows that suggested prediction improved the robustness of Grey system. Heuristic-based prediction improved 26.52% robust compared to Grey system prediction. The robustness of heuristic-based prediction is significant by p value almost to 0. Meanwhile the confidence interval (95%) for heuristic-based prediction is [59.98, 64.21] and Grey system prediction is [81.472, 87.548]. Heuristic-based prediction improved 42.76% robust while comparing with non-prediction.

In addition to these results, the heuristic-based prediction explored by the researcher was conceptually easier to understand and implement than the Grey system prediction and this can be seen from the descriptions presented in Chapter Three.

7.2 Contributions of the Research

This research makes a number of research contributions and implications to the current state of this research. The researcher separates the discussion into three areas: theory, research methodology and practice.

7.2.1 Theoretical Contribution

The product of this research is to come out with a new prediction algorithm based on heuristic theory. Once this algorithm is created, it can be used as prediction and reduce latency subsequently increase the effectiveness of virtual training. Proposed prediction provides the user with the facility to utilize the heuristic power of humans along with the algorithmic power and the geometry accuracy of motion-planning programs while taking biomechanical laws into consideration. The new hypotheses developed, which are substantiated may add new knowledge and help theory building in virtual training applications.

7.2.2 Methodological Contribution

As shown in the previous Chapter Three, Four and Six, the researcher has made a contribution to each of these fields. The researcher has developed, implemented, and documented prediction algorithm, which are required to build and maintain a virtual training application.

Chapter Three has shown inherent complexity of the Grey system prediction compared to the heuristic-based prediction. The Grey system prediction requires the calculation of partial derivatives, matrix multiplication and inverse and a more complex infrastructure, while the heuristic-based prediction only need a relatively low equations. The suggested algorithm used simple equation based on Bezier and B-spline curve to reduce the computation time to a practical limit. The shorter computation time represents the efficiency of the suggested prediction algorithm. Moreover, to become more and more realistic in visualization, the complexity of the rendered environment will increase dramatically and the latency will increase accordingly. Since, computation complexity of heuristic-based is relatively low compared to others and therefore real time requirement is easily met.

Because of the simplicity in the implementation of heuristic-based, it leads to the higher accuracy and also the efficiency of this algorithm.

The heuristic-based algorithm is geometrically robust and can be applied to all motions. No adjacency information is required. There is no tuning parameter, which happens frequently in Grey system algorithm.

The simplicity of heuristic-based theory algorithm is a definite advantage and also its power over the Grey system algorithm. This simple approach is at a sweet spot in the space of possible tracking approaches that can be easily to any virtual training and provides all necessary means of interaction through gestures. As it is a simple algorithm, it will allow virtual training designers to simply understand and implement this algorithm in their system.

7.2.3 Practical Contribution

The practical contributions of this research are derived from both theory and research findings. The combination of these two major contributions has had a major impact in the VR, training and sport application. In the first chapter, the researcher listed the goals and motivations for heuristic-based prediction. In achieving these goals, the researcher has done what has never been done before. The researcher has built a complete heuristic-based algorithm. In doing this, the researcher has had to identify problems in previous prediction algorithms, analyze the patterns of movement for the squash game and compare the performance of the suggested algorithm with the current prediction algorithm. The success of this work is evidenced by the statistical analysis presented in Chapter Six. The notable contributions to the field of VR included: 1) aiding researchers visualize the pattern of movement, 2) providing tools to design the algorithm and, perhaps most importantly, proving that it is possible to build a responsive virtual training system with a rule of thumb theory.

The experimental results in Chapter Six, demonstrated that the propose algorithm can follow fast motions of a player. The significantly faster running times and simplicity of heuristic-based prediction gives heuristic-based prediction a number of important benefits. The shorter running times of heuristic-based algorithm makes heuristic-based prediction better suited to handle the tracking motion of the players. The heuristic-based prediction simplicity makes them easier to implement than the Grey system prediction and also make the algorithm suitable for implementation. These are just a few of the many possible benefits that the heuristic-based prediction can provide which the researcher is continuing to explore in future.

Based on this research, heuristic-based prediction may be a viable alternative for predictive tracking because of its simplicity and speed and has shown it should be considered when employing a predictive tracking algorithm in a virtual training application. It is hope that it may increase the realism and effectiveness of virtual training application generally and specifically for sport simulation.

7.3 Limitations of the Research

It is hard to be objective when evaluating one's own work, but no work is ever perfect. A typical property of desktop VR systems is that subjects perform their task while sitting down. Therefore, and because of the limited display area, the extent of motions is limited. The use of keyboard becomes difficult quickly when the subject cannot see the devices that reduce the interactive view control of the subjects. This is because, in contrast to common virtual training applications where the subjects stand relatively still and make only slow movements, the squash game involves fast and wide range movements.

7.4 Future Recommendations

Throughout the thesis several ideas about future improvements have appeared. Primarily these ideas have emerged when analyzing the results from the evaluation study, but some have also appeared during implementation. A number of suggestions can also be directly derived from the original limitations inherent in the project.

7.4.1 Development of Immersive Virtual Training Application

As subjects have the difficulty in control the keyboard, an immersive VR will be developed, which uses a data glove (or data suit) to track movements of the user's body and then duplicate them in the VE. In order to improve 'reality' force information is important. At the impact on the ball, the player may feel the reaction force exerted on the racket by the ball using the force display device as the racket.

7.4.2 Extended Validation Process

Although the empirical results present strong evidence to support the research hypothesis, the researcher would like to obtain more evidence both empirically and analytically. Empirically, the researcher wishes to continue to run the experiments testing different datasets from other tracking systems, applications and interaction techniques. Adding more error metrics, such as screen error, more signal analysis tool for quantifying motion sequences, will provide more functionality to the testing and analysis application. The researcher hypothesizes that these results may extend to other domains, such as robotics and applications involving animation. Future work can validate this hypothesis and further experimentations will aid in showing the robustness of the heuristic-based prediction.

Analytically the researcher would like to apply complete squash strategies to the heuristic-based prediction. This work would provide a much stronger theoretical basis for the validity and accuracy of the heuristic-based algorithm in the context of motion prediction.

7.4.3 Further Enhancement on Player Modeling

Though results had proven that heuristic-based algorithm is effective, it doesn't consider on skeleton estimation also joint angle of human. Further developments will concern on representing player in the Virtual Environment application as a "photo-realistic" avatar in spite of a simple object. This is because in virtual training applications, it requires more and more flexible virtual humans with individualities.

Virtual human simulations are becoming each time more popular. This avatar is the replica of the human source and is extremely lifelike since the movements are driven by either prescript human animation data or actual human input.

7.4.4 Development of a Multiplayer Virtual Training Platform

A larger application of virtual training should be developed in order to determine the successful use of the suggested algorithm in a big application. As far the researcher knows, not much research work has been performed in the realm of the real-time multiplayer games. The researcher plans to realize a squash simulation with multiplayer support on a single screen. Multiplayer games where several persons interact simultaneously receive much interest since competing with human counterparts is typically considered as much more interesting and challenging than playing just against a computer. Reasons for applying multiplayer games are because humans are often play intelligently, spontaneously and intuitively. Therefore it is important to investigate the impact of suggested algorithm can have on the performance of the multiplayer games.

7.5 Summary

It is the researcher's hope that this findings will add to the body of knowledge in this area of research. Also it hopes that suggested heuristic-based prediction may be able to form a good basis for further investigation by other researchers.

REFERENCES

ADELSTEIN, B.D., LEE, T.G., & ELLIS, S.R. (2003). Head tracking latency in virtual environments psychophysics and a model. In *Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting*, 2083-2087.

ALLISON, R.S., HARRIS, L.R., JENKIN, M., JASIOBEDZKA, U., & ZACHER, J.E. (2001). Tolerance of temporal delay in virtual environments. In *Proceedings of the IEEE Virtual Reality*. 247-253.

AZUMA, R., & BISHOP, G. (1994). Improving static and dynamic registration in a see through HMD. In *SIGGRAPH'94*, pp.197-204.

BAEK, S., LEE, S., & KIM, G.J. (2003). Motion retargeting and evaluation for VR – based training of free motions. In *The Visual Computer*, 222-242.

BEZIER, P. (1970). Emploi des Machines a Commande Numerique. In Masson et Cie, Paris.

BIDEOU, B., MULTON, F., & KULPA, R. (2004). Virtual Reality Applied to Sports: Do Handball Goalkeepers React Realistically to Simulated Synthetic Opponents? In *ACM SIGGRAPH*, 210-216.

BLADE, R., & PADGETT, M. (2002). Virtual environments standards and terminology. *Handbook of Virtual Environments: Design, Implementations and Applications*, 15-27.

BOSCHKER, B.R., & MULDER, J.D. (2004). Lateral head tracking in desktop virtual reality. In *Eurographics Symposium on Virtual Environments*.

BROOKS JR, F.P., OUHYOUNG, M., BATTER, J.J., & KILLPATRICK, P.J. (1990). Project GROPE-Haptic Displays for scientific visualization. In *Computer Graphics 24*, 177-185.

BRUNNER, S., BIMBER, O., & MADER, S. (2003). *Report on tracking technology* [Online], Available www.ntb.ch/SEFI/TWI/Conf99.html [October 2004].

CALVIN, J., DICKENS, A., GAINES, R., METZGER, P., MILLER, D., & OWEN, D. (1993). The SIMNET virtual world architecture. In *Proceedings of the IEEE Virtual Reality Annual International Symposium*, 450-455.

CHOI, K.J., & KO, H.S. (2000). Online motion retargeting. In *The Journal of Visualization and Computer Animation*, 11(5): 223-235.

CHUA, P.T., CRIVELLA, R., DALY, B., HU, N., SCHAAF, R., VENTURA, D., CAMILL, T., HODGINS, J., & PAUSCH, R. (2003). Training for physical tasks in virtual environments: Tai Chi. In *IEEE Virtual Reality*, 87-94.

COLLINS, D., CHAPMAN, C., LOOPY, A.D., & THOMAS, P. (1990). *Get ready for squash: a complete training programme*. Great Britain: The Crowood Press.

DAVIS, J.W., & BOBICK, A.F. (1998). Virtual PAT: a virtual personal aerobics trainer. In *Workshop on Perceptual User Interface*.

DEERING, M. (1996). High resolution virtual reality. In E.E. Catmull, (Ed.), *Computer Graphics SIGGRAPH '92 Proceedings*, volume 26,195– 202.

DENG, J. (1989). Introduction to grey system theory. *The Journal of Grey System*, 1:1-24.

DUFFR, V.G., Ng, P.N.P., & RAMAKRISHNAN. A., (2004). Impact of a simulated accident in virtual training on decision making performance. In *International Journal of Industrial Ergonomics*, 34 (4) 335-348 (June 24, 2004).

ELLIS, S. R., YOUNG, M. J. EHRLICH, S. M., & ADELSTEIN, B. D. (1999). Discrimination of changes of rendering latency during voluntary hand movement. In *Proc. of the 43th Annual Meeting Human Factors and Ergonomics Society*, 1182-1186.

FAN, J., & XIONG, L. (2003). Intelligent driving training simulation system-based on virtual reality system. In *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, 27-30.

FRIEDMANN, MARTIN, STARNER, T., & PENTLAND, A. (1992). Device synchronization using and optimal linear filter. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics*, 57-62.

GARRETT, A.L. (2003). *Reduction of latency in Virtual Environment Applications.* Published dissertation, Jacksonville State University.

GIBSON, S., SAMOSKY, J., MOR, A., FYOCK, C., GRIMSON, E., KANADE, T., KIKINIS, R., LAUER, H., McKENZIE, N., NAKA JIMA, S., OHKAMI, H., OSBORNE, R., & SAWADA, A. (1996). Simulating arthroscopic knee surgery using volumetric object representations, real-time volume rendering and haptic feedback. Technical Report 96-19.

GLEICHER, M. (1998). Retargeting motion to new characters. In COHEN, M.F (Ed.), *SIGGRAPH 1998, ACM Press/ ACM SIGGRAPH,* 33-42.

HAMALAINEN, P., ILMONEN, T., HOYSNIEMI, J., LINDHOLM, M., & NYKANEN, A. (2005). Martial arts in artificial reality. In *CHI 2005*, 781-790.

HODGINS, J.K., O'BRIEN F., & TUMBLIN, J. (1998). Perception of human motion with different geometric models. In *IEEE Transactions on Visualization and Computer Graphics*, 4(4):307—316.

HWANG, Y.K., CHO, K.R., LEE, S., PARK, S.M., & KANG, S. (1997). Human computer cooperation in interactive motion planning. In *ICAR'97*, July 7-9, 571-576.

ISDALE, J. (1999). *Motion Tracking* [Online], Available http://vr.isdale.com/vrTechReviews/MotionTracker_Oct1999.htm [September 2004].

ISHII, H., WISNESKI, C., ORBANES, J., CHUN, B., & PARADISO, J. (1999). Ping-Pong plus: design of an athletic-tangible interface for computer-supported co-operative play. In *CHI'99. ACM Press*, 394-401.

JACOBY, R., ADELSTEIN, B.D., & ELLIS, S.R. (1996). Improved temporal response in virtual environment hardware and software. In *Proceedings of the SPIE Conference on Stereoscopic Displays and Applications VII*, 271-284.

JEFFRIES, R., MILLER, J.R., WHARTON, C., & UYEDA, K.M. (1991). User interface evaluation in the real world: a comparison of four techniques. In *Proceedings of the ACM CHI'91 Conference*, 119-124.

JIE, Q.X., QI, Q.Z., HONG, X.S., & DONG, W.Y. (2004). A virtual-real comparison technique used on sport simulation and analysis. In *ICSP'04 Proceedings*, 1296-1300.

JULIEN, T.S., & SHAW, C. (2002). Firefighter training virtual environment. In *SIGGRAPH Conference Abstracts and Applications*, 183.

KAGAMI, S., NISHIWAKI, K., KUFFNER JR. J.J., KUNIYOSHI, Y., INABA, M., & INOUE, H. (2002). Online 3D vision, motion planning and bipedal locomotion control coupling system of humanoid robot: H7. In *Proceedings of the 2002 IEEE/RSJ international Conference on Intelligent Robots and systems*, volume 3, 2557-2562.

KALAWSKY, R.S. (1993).*The science of virtual reality and virtual environments.* Wokingham, England: Addison-Wesley.

KIRULUTA, A., EIZENMAN, M., & PASUPATHY, S. (1997). Predictive head movement tracking using a Kalman filter. In *IEEE Transactions on Systems, Man and Cybernatics,* volume 27 (Z), 326-331.

KUPPERSMITH, R., JOHNSTON, R., MOREAU, D., LOTFIN, R., & JENKINS, H. (1997). Building a virtual reality temporal bone dissection simulator. In *Medicine Meets Virtual Reality, Studies in Health Technology and Informatics 39*, 180-186.

KYGER, D.W., & MAYBECK, P.S. (1998). Reducing Lag in Virtual Displays Using Multiple Model Adaptive Estimation. In *IEEE Transaction on Aerospace and Electronic Systems*, volume 34 (4), 1237-1248.

LAVIOLA, J. (2000). A discussion of cybersickness in virtual environments. In *ACM SIGCHI Bulletin*, vol 32 issue 1.

LAVIOLA JR, J.J. (2003a). A comparison of unscented and extended kalman filtering for estimating quaternion methods. In *AM Control Conf,* 2435-2440.

LAVIOLA JR, J.J. (2003b). A testbed for studying and choosing predictive tracking algorithms in virtual environments. In *Eurographics Workshop on Virtual Environments.*

LAVIOLA JR, J.J. (2003c). Double exponential smoothing: an alternative to kalman filter-based predictive tracking. In *Eurographics Workshop on Virtual* Environments.

LIN, Y., & LIU, S. (2004). A historical introduction to grey systems theory. In *IEEE International Conference on Systems, Man and Cybernetics.*

LIANG, J., SHAW, C., & GREEN, M. (1991). On temporal-spatial realism in the virtual reality environment. In *Proceedings 4th Annual ACM Symposium on User Interface Software and Technology (UIST) 1991*, pp. 19-25.

LOPEZ, C., & PLATEN, P. (2005). Motion analysis in handball-a review. In *4th International Symposium on Image and Signal Processing and Analysis*, 345-346.

LOTFIN, R., & KENNY, P. (1995). Training the Hubble space telescope flight team. In *IEEE Computer Graphics and Applications 15*, 31-37.

MCKENZIE, I. (1996). *The skill of the game: squash*. Great Britain: The Crowood Press.

MACKENZIE, I. (1998). *The squash workshop: a complete game guide*. Hong Kong: The Crowood Press.

MACKENZIE, I.S & WARE. (1993). Lag as a determinant of human processing in interactive systems. In *INTERCHI'93 ACM Press*, 488-493.

MAS, R., & THALMANN, D. (1994). A hand control and automatic grasping system for synthetic actors. In *Proceedings of the Eurographics'94*.

MAZURYK, T., & GERVAUTZ, M. (1995). *Two-step prediction and image deflection for exact head tracking in virtual environments.* In *EUROGRAPHICS'95*.

MEEHAN, M., RAZZAQUE, S., WHITTON, M.C., & BROOKS JR, F.P. (2003). Effect of latency on presence in stressful virtual environments. In *Proceedings of the IEEE Virtual Reality*, 141-148.

MINE, M.R. (1993). *Characterization of end-to-end delays in head-mounted display systems*. Technical. Report TR93-001, Chapel Hill, NC, University of North Carolina.

MINERS, B. (2001). *Kalman filtering and prediction for hand tracking, an advanced DSP project* [Online], Available: http://pami.uwaterloo.ca/pub/bminers/kalmanhand.pdf [September 2004].

MOKKA, S., VAATANEN, A., & VALKKYEN, P. (2003). Fitness Computer Games with a Bodily User Interface. In *Second International Conference on Entertainment Computing* (Ed.), ACM International Conference Proceeding Series, 1-3.

MUELLER, F., AGAMANOLIS, S., & PICARD, R. (2003). Exertion interfaces: sports over a distance for social bonding and fun. In *CHI'03, ACM Press*, 561-568.

NAKAZAWA, A., NAKAOKA, S., SHIRATORI, T., & IKEUCHI, K. (2003). Analysis and synthesis of human dance motions. In *Sixth International Conference on Humans and Computers (HS2003)*, 83-88.

NIELSEN, J. (1993). *Usability engineering*. London: Morgan Kauffman.

NOSER, H., & THALMAN, D. (1997). Sensor Based Synthetic Actors in a Tennis Game Simulation. In *Computer Graphics International'97* (Ed.), IEEE Computer Society Press, 1997, 189-198.

OHYA, J. (2002). Analysis of human behaviors by computer vision based approaches. In *2002 IEEE International Conference on Multimedia and Expo (ICME2002)*, 913-916.

PARK, W., CHAFFIN, D.B., & MARTIN, B. J. (2004). Toward memory-based human motion simulation: development and validation of a motion modification algorithm. In *IEEE Transactions on Systems, Man and Cybernetics*, May, 376-386.

PERDIGAU, E., TORGUET, P., SANZA, C., & JESSEL, J.P. (2003). A distributed virtual storytelling system for fighters training. In *ICVS'03*, 227-230.

PERKINS, D, N. (1981). *The Mind's Best Work.* Cambridge, MA: Harvard University Press.

PERSE, M., PERS, J., KRISTAN, M., KOVACIC, S., VUCKOVIC, G., 2005. Physics-based modeling of human motion using Kalman filter and collision avoidance algorithm. In *4th International Symposium on Image and Signal Processing and Analysis*, 328-333.

PONDER, M., HERBELIN, B., MOLET, T., SCHERTENLIEB, S., ULICNY, B., PAPAGIANNAKIS. G., THALMANN, N.M., & THALMANN, D. (2003). Immersive VR decision training: telling interactive stories featuring advanced virtual human simulation technologies. In *The Eurographics Association 2003*, 97-106.

POPOVIC, Z., & WITKIN, A. (1999). Physically based motion transformation. In *Proceedings of the SIGGRAPH 99, ACM SIGGRAPH*, 11-20.

POSTON, T., & SERRA, L. (1996). Dextrous virtual work. In *CACM,* 39(5):37–45.

Price, H. (1995). *Times' arrow & archimedes' point: new directions for the physics.* Oxford University Press of Time.

RHIJN, A.V., ROBERT, V.L., & MULDER, J.D. (2005). An analysis orientation prediction and filtering methods for virtual reality and augmented reality. In *Proceeding of IEEE VR2005*, Mar, 2005, Germany, 67-74.

RUISSEAU, J.Y., TODESCHINI, L., & GORZERINO, P. (2000). Learning and training simulator virtual reality development for parachute descent. In *Proceedings of the IEA 2000/HFES 2000 Congress, 527-529.*

RUSDORF, S., & BRUNNET, G. (2005). Real-time tracking of high-speed movements in the context of a table tennis application. In *VRST'05*, (pp92-200). ACM-Press, New York.

SEKARAN, U. (2003). *Research methods for business*. Carbondale: John Wiley & Sons.

SHERIDAN, T.B. (1999). Musings on tele-presence and virtual presence. In *Presence: Tele-operators and Virtual Environments 1*, 212-227.

SHERIDAN, T.B., & FERREL, W.R. (1963). Remote manipulative control with transmission delay. In *IEEE Transactions on Human Factors in Electronics 4 (1)*, 25-29.

SMITH, K.U., & SMITH, W.M. (1962). *Perception and action*. Saunders: Philadelphia, 247-277.

SMITH, W.M., McCRARY, J.R., & SMITH, K.U. (1962). Delayed visual feedback and behavior. *Science*, 132, 103-1014.

SOMMERS, E. (1991). *Squash: technique, tactics, training*. Hong Kong: The Crowood Press.

STANSFIELD, S. (1998). Application of VR to nuclear safeguards. In *Joint ESRADA/INMM Workshop on Science and Modern Technologies for Safeguards.*

SUTCLIFFE, A. G., & KAUR, K. D, (2000). Evaluating the usability of virtual reality user interfaces. In *Behavior and Information Technology*, 415-426.

SUTHERLAND, I. (1969). A head mounted three dimensional display. In *Fall Joint Computer Conferences, AFIPS Conference Proceedings*, 33, 757-764.

TAKAYAMA, K., & KANO, H. (1995). A new approach to synthesizing free motions of robotic manipulators based on a concept of unit motions. In *IEEE Transactions on Systems, Man and Cybernetics*, 453-463.

TATE, D.L., & SIBERT, L. (1997). Virtual environments for shipboard fire fighting training. In *Proceedings of the IEEE virtual reality*, 61-66.

THALMANN, D., & MONZANI, J.S. (2002). Behavioral animation of virtual humans: what kind of laws and rules? In *Computer Animation 2002 (CA 2002).*

TSUJI, T., TAKEDA, Y., & TANAKA, Y. (2004). Analysis of mechanical impedance in human arm movements using a using a virtual tennis system. In *Biological Cybernetics*, 295-305.

VUCKOVIC, G., DEZMAN, B., PERS.J., & KOVAVIC. S. (2005) Motion analysis of the international and national rank squash players. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium*, 334 – 338.

WAN, T.R., CHEN, H., & EARNSHAW, R. (2003). Real-time path planning for navigation in unknown environment. In *Proceedings of the Theory and Practice of Computer Graphics (TPCG'03).*

WARE, C., & BALAKRISHNAN, R. (1994). Reaching for objects in VR displays: lag and frame rate. In *ACM Transactions on Computer-Human Interaction.*

WARE, C., FRANCK, G. (1996). Evaluating stereo and motion cues for visualizing information nets in three dimensions. In *ACM Transactions on Graphics*, 15(2):121–140.

WIET, G., YAGEL, R., STREDNEY, D., SCHMALBROCK, P., SESSANNA, D., KURZION, Y., ROSENBERG, L., LEVIN, M., & MARTIN, K. (1997). A volumetric approach to virtual simulation of functional endoscopic sinus surgery. In *Medicine Meets Virtual Reality, Studies in Health Technology and Informatics 39.*

WITMER, B. G., & SINGER, M. J. (1999). Measuring presence in virtual environments: A presence questionnaire. In *Presence 7*, 225-240.

WONG, C.F., & BENG, H.T.H. (1995). *Taking up squash.* Shah Alam: Penerbit Fajar Bakti Sdn. Bhd.

WU, J.R., DUH, C.D., & OUHYOUNG, M. (1997). Head motion and latency compensation on localization of 3D sound in virtual reality. In *ACM VRST'97.* Lausanne: Switzerland.

WU, J.R., & OUHYOUNG, M. (1994). Reducing the latency in head mounted displays by a novel prediction method using grey system theory, In *Computer Graphics Forum (EuroGraphics'94)*, 503-512.

WU, J.R., & OUHYOUNG, M. (1995). A 3D tracking experiment on latency and its methods. In *UIST'95*, 41-49.

WU, J.R., & OUHYOUNG, M. (2000). On latency compensation and its effect on head motion trajectories in virtual environments. In *The Visual Computer*, 79-90.

XIA, S., QIU, X., & WANG, Z. (2005). A novel framework for athlete training-based on interactive motion editing and silhouette analysis. In *VRST'05*, 56-58.

YANG, U. (1999). Just follow me: an immersive VR-based motion training system. In *International Conference on Virtual Systems and Multimedia*.

YARROW, P. (1997). *Squash: steps to success*. USA: Human Kinetics Publisher.

YUNG, J.Y., ADELSTEIN, B.D., & ELLIS, S. (2000). Discriminability of prediction artifacts in a time-delayed virtual environment. In *IEA 2000/HFES 2000 Congress*.

ZORDAN, V.B., & HODGINS, J.K. (2002). Motion-capture driven simulations that hit and react. In *ACM SIGGRAPH Symposium on Computer Animation*, 89-96.

ZORDAN, V.B., & HORST, N.C.V.D. (2003). Mapping optical motion capture data to skeletal motion using a physical model. In *The Eurographics Association*, 245-250.

APPENDIX 1

Pre-test Questionnaire

Using the preceding Likert scale, state the extent to which you agree with the following statements.

1. ***Object can easily be distinguished from all likely viewpoints.***

Strongly disagree	Disagree	Neither disagree nor agree	Agree	Strongly agree

2. ***Important parts of objects can be easily distinguished from the object image.***

Strongly disagree	Disagree	Neither disagree nor agree	Agree	Strongly agree

3. ***Prominent features of an object been represented.***

Strongly disagree	Disagree	Neither disagree nor agree	Agree	Strongly agree

4. ***Object size matches the Virtual Environment.***

Strongly disagree	Disagree	Neither disagree nor agree	Agree	Strongly agree

5. ***Background has an adequate and proper lighting.***

Strongly disagree	Disagree	Neither disagree nor agree	Agree	Strongly agree

6. ***Purpose of action is satisfied.***

Strongly disagree	Disagree	Neither disagree nor agree	Agree	Strongly agree

7. ***Action being made and results should match the user's expectations.***

Strongly disagree	Disagree	Neither disagree nor agree	Agree	Strongly agree

****

The questions in the survey may not be all embracing and comprehensive and may not therefore have afforded you an opportunity to report some things you may want to say. Please make any additional comments needed, in the space provided.

Presence Questionnaire

Using the Likert scale, please indicate your response to each of the items that follow, by circling the scale that best describes your feeling.

1. How well were you able to control the system?

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

2. How responsive was the environment to actions that you initiated (or performed)?

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

3. How natural did your interactions with the environment seem?

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

4. How natural was the mechanism, which controlled movement through the environment?

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

5. How aware were you of events occurring in the real world around you?

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

6. How aware were you of your display and control devices?

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

*7. **How compelling was your sense of objects moving through space?***

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

8. How much did your experiences in the virtual environment seem consistent with your real world experiences?

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

9. Were you able to anticipate what would happen next in response to the actions that you performed?

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

10. How completely were you able to actively survey or search the environment using vision?

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

11. How compelling was your sense of moving around inside the virtual environment?

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

12. How closely were you able to examine objects?

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

13. How well could you examine objects from multiple viewpoints?

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

14. How well could you move or manipulate objects in the virtual environment?

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

15. How involved were you in the virtual environment experience?

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

16. How much delay did you experience between your actions and expected outcomes?

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

17. How quickly did you adjust to the virtual environment experience?

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

18. How proficient in moving and interacting with the virtual environment did you feel at the end of the experience?

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

19. Were you involved in the experimental task to the extent that you lost track of time?

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

20. How effective was the sense of perspective (depth of field)?

Very inferior	Inferior	Neither inferior nor superior	Superior	Very superior

****

I sincerely appreciate your time and cooperation. Please check to make that you have not skipped any questions inadvertently.

Thank you!

APPENDIX 2

SAMPLE DATA FOR TIMES PER SECOND

	25			60		
Subject	**Non**	**Grey**	**Heu**	**Non**	**Grey**	**Heu**
1	5.164	2.76	1.815	3.021	2.716	2.202
2	6.24	1.929	2.76	2.417	2.106	1.641
3	3.82	2.403	1.685	3.337	2.73	1.643
4	3.72	1.8	1.686	3.477	2.008	1.658
5	3.363	1.935	1.8	3.347	2.513	1.578
6	4.44	1.814	1.92	3.34	2.534	1.783
7	3.724	2.659	1.939	3.365	2.75	1.649
8	4.448	2.644	1.456	3.423	2.715	1.751
9	4.457	2.641	1.803	3.228	2.638	1.861
10	3.606	2.16	1.8	3.228	2.728	1.989
11	2.524	2.164	2.04	2.629	2.486	1.829
12	2.4	2.422	1.8	2.494	2.405	1.938
13	3.732	2.649	1.68	3.326	2.188	1.84
14	4.92	2.172	1.816	2.725	2.204	2.2882
15	3.731	2.403	1.92	3.115	2.477	1.651
16	3.008	2.888	1.809	2.722	2.223	1.539
17	3.84	2.4	1.815	2.832	2.83	2.371
18	3.017	2.415	1.803	3.032	2.63	1.95
19	4.206	2.163	1.803	3.028	2.524	1.518
20	2.651	2.04	1.935	3.157	2.081	2
21	2.64	2.542	1.68	3.01	2.215	1.743
22	4.082	2.302	1.696	2.728	2.652	1.822
23	3.377	3.614	1.685	3.006	2.818	1.806
24	3.018	2.763	1.68	2.966	2.11	1.827
25	2.52	3.24	1.701	3.039	1.984	1.399
26	4.201	2.779	1.93	2.729	2.316	1.555
27	2.651	2.773	1.694	2.623	2.605	1.73
28	3.614	2.284	1.8	2.725	3.239	1.306
29	3.243	2.164	1.44	2.751	2.09	1.717
30	2.64	3	1.56	3.153	2.737	1.74
31	2.64	2.886	1.681	2.208	2.299	1.727
32	3.24	2.663	1.92	2.516	2.381	1.521
33	4.326	2.88	1.935	2.726	2.392	1.737
34	3.488	2.77	2.537	3.336	1.918	1.412
TOTAL	**122.691**	**85.121**	**62.024**	**100.759**	**83.242**	**59.3422**

80		
Non	Grey	Heu
2.278	2.618	1.47
1.733	1.994	1.95
2.547	2.095	1.765
1.807	2.691	1.747
2.229	1.492	1.388
3.267	2.617	1.49
1.794	2.109	1.303
1.931	2.358	1.682
2.486	2.622	1.775
1.706	2.13	1.614
2.642	1.879	1.862
3.073	2.697	1.727
2.593	2.598	1.488
3.203	3.281	1.571
3.19	2.206	1.503
2.678	2.683	1.734
2.757	2.61	1.695
3.436	2.077	1.283
3.38	2.334	1.573
2.681	2.404	1.609
2.436	1.769	1.665
2.739	1.688	1.687
2.671	2.007	1.492
2.338	2.386	1.447
2.674	2.443	1.385
2.481	2.005	1.667
2.432	2.468	2.508
2.666	1.922	1.817
1.586	2.541	2.062
2.576	1.794	1.794
2.375	2.204	1.593
2.746	2.385	1.878
2.888	2.484	1.655
2.573	2.424	1.775
86.592	**78.015**	**56.654**

140		
Non	Grey	Heu
2.554	2.591	1.495
1.802	2.12	1.522
1.721	2.005	1.674
1.721	1.485	1.796
1.792	1.91	1.582
2.658	2.009	1.643
1.704	1.909	1.59
2.012	2.019	1.863
1.838	2.084	1.897
1.809	1.905	1.836
2.75	2.669	1.485
3.168	2.585	1.479
2.654	2.512	1.646
3.116	2.301	1.368
2.884	2.322	1.865
2.77	2.5	1.782
2.613	2.665	1.368
2.265	2.374	1.392
2.8	2.219	1.749
2.747	2.788	1.572
2.581	2.655	1.522
2.338	2.204	2.184
2.785	2.473	1.488
2.461	2.134	1.76
2.831	2.123	1.765
2.546	2.475	1.746
2.548	2.497	1.367
2.67	2.162	1.563
2.539	2.687	1.837
2.85	2.061	1.645
2.6	2.393	1.481
2.995	2.214	1.581
2.571	2.729	1.679
2.584	2.103	1.764
84.277	**77.882**	**55.986**

APPENDIX 3

SAMPLE DATA FOR FRAMES PER SECOND

	25			60		
Subject	**Non**	**Grey**	**Heu**	**Non**	**Grey**	**Heu**
1	129	69	45	117	101	80
2	156	48	69	93	81	60
3	95	60	42	128	100	60
4	93	45	42	133	76	60
5	84	48	45	129	92	70
6	111	45	48	129	93	64
7	93	66	48	129	75	60
8	111	66	36	132	100	75
9	111	66	45	125	86	67
10	90	54	45	125	102	73
11	63	54	51	101	96	82
12	60	60	45	96	93	75
13	93	66	42	128	84	71
14	123	54	45	104	85	88
15	93	60	48	120	95	62
16	75	72	45	105	86	59
17	96	60	45	109	109	90
18	75	60	45	116	101	75
19	105	54	45	116	98	58
20	66	51	48	121	80	62
21	66	63	42	116	81	67
22	102	57	42	105	97	70
23	84	90	42	116	70	69
24	75	69	42	113	75	70
25	63	81	42	117	72	67
26	105	69	48	105	106	59
27	66	69	42	101	100	66
28	90	57	45	101	76	82
29	81	54	36	104	76	66
30	66	75	39	121	70	67
31	66	72	42	85	89	66
32	81	66	48	97	90	58
33	108	72	48	105	92	67
34	87	69	63	129	73	80
TOTAL	**3062**	**2121**	**1545**	**3871**	**3000**	**2345**

80		
Non	**Grey**	**Heu**
120	107	61
85	88	61
80	82	68
80	61	85
85	78	64
125	83	63
80	78	65
95	83	78
74	80	77
85	77	71
130	111	60
150	107	59
107	97	62
126	97	55
135	98	78
129	97	73
104	111	56
105	99	56
130	92	67
130	117	65
121	111	58
110	91	88
130	104	60
114	88	71
129	88	73
119	103	67
120	104	56
125	54	64
120	112	76
130	79	67
120	99	56
140	93	63
120	112	69
120	81	72
3873	**3162**	**2264**

140		
Non	**Grey**	**Heu**
105	101	60
80	82	75
120	87	73
85	111	72
105	92	57
132	108	61
72	88	52
77	98	69
110	76	72
80	87	65
125	107	76
145	112	66
104	110	61
150	84	64
150	93	61
125	141	67
131	109	70
139	86	52
159	89	65
125	99	62
115	73	68
130	70	68
125	83	61
110	99	59
124	102	56
114	82	68
115	95	102
107	81	69
75	98	84
120	73	73
110	92	65
130	98	78
135	103	68
120	102	72
3949	**3211**	**2291**