

Development of Immersive and Interactive Virtual Reality Environment for Two-Player Table Tennis

by

Yingzhu LI

A thesis submitted to the University of Central Lancashire
in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

March, 2012

The work presented in this thesis was carried out in the Applied Digital Signal and Image Processing (ADSIP) Research Centre, School of Computing, Engineering and Physical Sciences, University of Central Lancashire, Preston, England.

Abstract

Although the history of Virtual Reality (VR) is only about half a century old, all kinds of technologies in the VR field are developing rapidly. VR is a computer generated simulation that replaces or augments the real world by various media. In a VR environment, participants have a perception of “presence”, which can be described by the sense of immersion and intuitive interaction. One of the major VR applications is in the field of sports, in which a life-like sports environment is simulated, and the body actions of players can be tracked and represented by using VR tracking and visualisation technology. In the entertainment field, exergaming that merges video game with physical exercise activities by employing tracking or even 3D display technology can be considered as a small scale VR.

For the research presented in this thesis, a novel realistic real-time table tennis game combining immersive, interactive and competitive features is developed. The implemented system integrates the InterSense tracking system, SwissRanger 3D camera and a three-wall rear projection stereoscopic screen. The Intersense tracking system is based on ultrasonic and inertia sensing techniques which provide fast and accurate 6-DOF (i.e. six degrees of freedom) tracking information of four trackers. Two trackers are placed on the two players’ heads to provide the players’ viewing positions. The other two trackers are held by players as the racquets. The SwissRanger 3D camera is mounted on top of the screen to capture the player’s movement in real time, thereby enhancing the realism of the VR environment. The entire virtual environment is projected onto a big rear-projection stereoscopic screen for each player. By wearing a pair of polarised glasses, the participants are able to enjoy the immersive game experience. In order to achieve a realistic effect, a detailed physics based model is developed and implemented for the prediction of ball trajectory, which takes into account of various physical phenomena such as air

resistance, gravity, Magnus effect, ball spin, and frictions. The game simulation can be run in either single-player or two-player mode. In the two-player mode, the data is transferred between the server computer and two client computers through a 1G bytes Ethernet link using the TCP/IP protocol.

The performance of the system is evaluated by a user-based study. Through the statistical analysis of both the qualitative and quantitative data obtained from the user-based experiment, the developed framework, heuristics and questionnaires were proved to be valid and reliable, and each component of the system was assessed by rating scores. Generally speaking, the system performance in terms of both technology and presence aspects is good. In addition, the statistical results prove the importance of stereoscopic display, head-tracking as well as “opponent” motion capture and display.

Contents

Chapter1 Introduction	1
1.1 Overview.....	2
1.2 Aims and Objectives.....	4
1.3 Thesis Outline.....	4
Chapter 2 Literature Review and System Design	6
2.1 Introduction.....	6
2.2 Literature Review.....	7
2.2.1 Background of Virtual Reality.....	7
2.2.2 Current Table Tennis Simulation/Exergaming.....	9
2.3 Proposed VR Environment.....	12
2.3.1 Design of VR Environment.....	12
2.3.2 Requirements of Motion Tracking System.....	15
2.3.2.1 Head and Hand Motion Tracking.....	16
2.3.2.2 Motion Tracking for Visible Opponent.....	16
2.3.3 Requirements of Immersive Display System.....	17
2.4 Conclusion.....	18
Chapter 3 Motion Tracking and Stereoscopic Display	19
3.1 Introduction.....	19
3.2 Hand and Head Motion Tracking.....	20
3.2.1 Current 6-DOF Motion Tracking Technology.....	20
3.2.2 InterSense™ Motion Tracking Technology.....	21
3.2.2.1 System Introduction.....	21
3.2.2.2 System Configuration.....	22
3.2.2.3 Operation Principle.....	23

3.2.3	6-DOF Motion Tracking Results.....	25
3.2.3.1	Motion Tracking Output.....	25
3.2.3.2	Motion Tracking Performance.....	27
3.3	Body Motion Tracking.....	29
3.3.1	Current 3D Video Motion Tracking Technology.....	29
3.3.2	Swissrange™ Motion Tracking Technology.....	30
3.3.2.1	Camera Introduction.....	30
3.3.2.2	Camera Setup and Operation Principle.....	31
3.3.3	3D Body Motion Tracking Results.....	33
3.3.3.1	Motion Tracking Output.....	33
3.3.3.2	3D Video Generation.....	34
3.3.3.3	3D Video Performance.....	35
3.4	Stereoscopic Display.....	39
3.4.1	Current Technology and Operation Principle.....	39
3.4.2	Rear-projection Polarised Stereoscopic Display System.....	41
3.4.3	Stereo Pair Rendering.....	42
3.4.3.1	Principle and Current Approaches.....	42
3.4.3.2	Stereo Pair Rendering by OpenGL.....	44
3.4.4	Display Performance.....	46
3.5	Conclusion.....	47
 Chapter 4 Virtual Reality Environment for Single-player Table Tennis..		49
4.1	Introduction.....	49
4.2	Hardware and Software Implementation.....	50
4.3	Game Environment Generation.....	51
4.3.1	Virtual Objects Generation.....	51
4.3.2	Coordinate Systems.....	53
4.4	Applying Motion Tracking Data.....	55
4.4.1	Racquet Animation based on Hand Tracking.....	55

4.4.2 Virtual Camera Setting based on Head Tracking	57
4.4.3 Frustum Setting based on Head Tracking	60
4.4.3.1 Distortion Analysis	61
4.4.3.2 Real-time Frustum Adjustment	64
4.4.3.3 Results Comparison	66
4.5 Ball Animation	69
4.5.1 Physics Model of Flying Ball	70
4.5.2 Physics Model of Collision Detection	74
4.5.2.1 Collision with Stationary Object	74
4.5.2.2 Collision with Tracked Moving Object	78
4.5.3 Physics Model of Collision Response	84
4.5.3.1 Rebounding from Stationary Surface	85
4.5.3.2 Rebounding from Moving Surface	92
4.6 Conclusion	95
Chapter 5 Virtual Reality Environment for Two-player Table Tennis	96
5.1 Introduction	96
5.2 System Implementation and Communication	97
5.3 Coordinate Systems of Two-Player Game Environment	100
5.4 3D Opponent Display	103
5.5 Game Strategy and Results	107
5.6 Conclusion	110
Chapter 6 User-based Evaluation	111
6.1 Introduction	111
6.2 Evaluation Methodology Development	112
6.2.1 Background Investigation	112
6.2.2 Development of Evaluation Heuristics	114
6.2.2.1 Existing Approaches	114

6.2.2.2	Development of Framework and Heuristics.....	116
6.3	Experiment Design.....	122
6.3.1	Aim of Experiment.....	122
6.3.2	Test Procedure.....	123
6.3.3	Questionnaire Development.....	126
6.4	Experiment Results.....	130
6.4.1	Experiment Data and Statistical Analysis.....	130
6.4.2	Reliable and Valid Measurement.....	132
6.4.2.1	Reliability.....	132
6.4.2.2	Validity.....	134
6.4.3	Data Exploring for System Evaluation.....	136
6.4.3.1	Simulator Sickness.....	136
6.4.3.2	Technological Achievement.....	136
6.4.3.3	Sense of Presence.....	139
6.4.3.4	Task Performances.....	140
6.4.3.5	Advantages of Displaying Opponent.....	144
6.5	Conclusion.....	146
 Chapter 7 Conclusion and Future Work.....		149
7.1	Conclusions.....	150
7.2	Contributions.....	152
7.3	Future Work.....	154

List of Figures

2.1 Requirements of Proposed Table Tennis Game.....	14
3.1 IS-900 System Configuration.....	22
3.2 IS-900 Equipment.....	23
3.3 IS-900 System Operation.....	24
3.4 InterSense Reference System.....	26
3.5 Screenshot of ISDEMO.....	26
3.6 Screenshot of InterSense Data Stream.....	26
3.7 SR4000 Camera.....	31
3.8 Operation Principle of SR4000 Camera.....	32
3.9 Screenshot of SR_3D_Viewer.....	33
3.10 Screenshot of SR4000 Data Stream.....	34
3.11 Screenshot of Generated 3D Video.....	35
3.12 Sample of Background Noise.....	36
3.13 Results with 0.3ms (left) and 25ms (right) Integration Times.....	38
3.14 Results with 1ms (left), 2ms (middle) and 3ms (right) Integration Times.....	38
3.15 Configuration of Rear-projection Stereoscopic Screen.....	41
3.16 Three Types of Projection (Positive/Zero/Negative Parallax)	43
3.17 Toe-in (left) and Off-axis (right) Projections.....	43
3.18 OpenGL Frustum.....	44
3.19 Frustum Setting for Right Eye.....	45
3.20 Photos of Virtual Balls and Real Table Tennis Balls	46
4.1 Hardware Configuration for Single-player Game.....	50
4.2 Software Implementation for Single-player Game.....	50
4.3 OpenGL-generated Virtual Objects.....	52

4.4 Coordinate Systems.....	53
4.5 Views of Game Scene.....	55
4.6 Views at Different Camera Locations.....	58
4.7 Views with Different Camera Rotations.....	59
4.8 Incorrect Frustum Setting with Distortion (Side View)	61
4.9 Correct Frustum Setting without Distortion (Side View)	61
4.10 Views with incorrect (left) and correct (right) Frustum Setting.....	62
4.11 Distortion Analysis for Viewpoint Displacement along x-axis.....	62
4.12 Distortion Analysis for Camera Displacement Along z-axis.....	63
4.13 Distortion Caused by the Scene Orientation (Side View)	64
4.14 Asymmetrical Frustum Setting (Top View)	65
4.15 Views with Fixed (left) and Head-tracking based (right) Frustum for a Low-position Viewpoint.....	67
4.16 Views with Fixed (left) and Head-tracking based (right) Frustum for a High-position Viewpoint.....	67
4.17 Views with Fixed (top) and Head-tracking based (bottom) Frustum for a Left-side Viewpoint.....	68
4.18 Views with Fixed (top) and Head-tracking based (bottom) Frustum for a Right-side Viewpoint.....	68
4.19 Views with Default (left) and Head-tracking based (right) Viewing Angle.....	69
4.20 Views with Default (left) and Head-tracking based (right) Viewing Angle.....	69
4.21 Forces Acting on a Flying Ball.....	71
4.22 Collision between a Ball and a Stationary Flat Surface.....	75
4.23 Colliding-Surfaces' Area of Table and Net.....	77
4.24 Two Collision Events in One Frame.....	78
4.25 Prediction of Racquet's Position.....	79
4.26 Situation of Prediction Error.....	80
4.27 Prediction of Racquet's Rotation.....	81
4.28 Two Colliding-Surfaces of a Racquet.....	83
4.29 Relative Movement of Ball to Racquet.....	84

4.30 Two Examples of Ball-Table Collision	85
4.31 Ball-Table Collision.....	87
5.1 Hardware Configuration of Two-player System.....	97
5.2 Workflow of Two-player System.....	98
5.3 Flowchart of Communication.....	99
5.4 Coordinate Systems of Two-Player Game.....	101
5.6 3D Camera Positioning.....	104
5.7 Generating 3D Surface by Triangles.....	105
5.8 Opponent's 3D Surfaces with (right) and without (left) rotation, with (bottom) and without (top) noise reduce.....	106
5.9 Player is Reading a 4-pages Tutorials.....	107
5.10 Player Plays with her Opponent.....	108
5.11 Two Players Play Against Each Other.....	109
5.12 Win and Lose in the Game.....	109
6.1 Framework of an Overall Usability Evaluation for VR Applications.....	117
6.2 Mean Scores of 18-TQ Items.....	137
6.3 Mean Scores of the Five Pretense Issues.....	139
6.4 Comparing Mean Scores among Different Groups.....	144
6.5 Mean scores of five PQ items.....	145

List of Tables

2.1 Features of Seven Table Tennis Games.....	11
3.1 IS-900 Performance Specifications.....	27
3.2 SR4000 Camera Performance Specifications.....	35
6.1 Gabbard's Taxonomy and VRUSE.....	114
6.2 Introduction of PQ, SUS and ITC-SOPI.....	115
6.3 Heuristics for Technological Achievement.....	118
6.4 Heuristics for Sense of Presence.....	120
6.5 Sub-factors in Individual Characteristics.....	121
6.6 Questionnaire Items in TQ.....	127
6.7 Questionnaire Items in TQ.....	128
6.8 Questionnaire Items in PQ.....	129
6.9 Experiment Data.....	131
6.10 Analysis of the 3 Low alpha Items.....	133
6.11 Correlations among Questionnaires.....	135
6.12 Four Questions Addressing Involvement.....	139
6.13 ANOVA Results for Stroke Analyses.....	142
6.14 ANOVA Results for Point Analyses.....	142
6.15 Comparisons of five PQ items.....	145
7.1 Features of the Developed System.....	150

Acknowledgements

I wish to express my sincere gratitude to my Director of Studies, Professor Lik-Kwan Shark for his consistent support, guidance and patience through all stages of this work, and to my Second Supervisors, Doctor Sarah Jane Hobbs, Jim Ingham and Doctor Wei Deng for their suggestion, support and encouraging.

I would like to thank the University of Central Lancashire for the support wherever possible. In particular, I would like to thank the ADSIP research centre, which provides many good facilities for my project.

A special thank to my boyfriend Xin for his support and love. It is also a pleasure to thank my research colleagues (especially to my very good friends Hong and Xiao Yin) for their friendship, and the pleasant and inspiring atmosphere they bring to me. Thanks to my best friend Mi for her good wishes from Germany.

I dedicate this thesis to my family (especially to my mum Xia and my dad You Fu) with love and gratitude.

Chapter 1

Introduction

1.1 Overview

Virtual Reality (VR) is a computer generated simulation that replaces or augments the real world by various media. VR environment aims to provide participants a perception of “presence”, which can be described by two aspects: immersion and intuitive interaction. Based on this concept, all kinds of technologies in the VR field are developing rapidly. Collaborative virtual reality (CVR) supports interaction among multiple participants, and remote CVR can be implemented based on telepresence. Augmented reality (AR) superimposes computer-generated objects into the real world. Although the history of VR is only about half a century old, its benefits have been extended to various application areas, such as simulators for aircraft cockpits and vehicles, movement analysis for sports and rehabilitation, scene representation for archaeological sites and museums, skill training for surgery operation and engineering repair, data visualization for scientific research and industry design, as well as entertainment.

In the sports field, by using realistic VR simulation, any special sport environment or dangerous situation can be simulated with all the parameters under strict control, and all the body actions of athletes can be tracked and visualised for movement analysis. These advantages are not achievable in the real world training environment. In the game field, exergame that combines sports and video game employs the VR concept to attract and motivate people to physical exercise, which benefits people’s health and the quality of life.

In this research, the proposed VR content is a realistic real-time two-player table tennis game with immersive, interactive and competitive features. The system is designed to simulate the table tennis game as realistically as possible. Since playing table tennis requires accurate spatial information of the ball and natural control of the

racquet, there are challenges to implement such a system with good tracking quality, accurate 3D effect, low response latency and realistic ball movements.

In order to fulfil these requirements, the implemented system integrates the InterSense tracking system, SwissRanger 3D camera and a three-wall rear projection stereoscopic display system. The Intersense tracking system is based on ultrasonic and inertia sensing technologies, which is able to provide fast and accurate 6-DOF (i.e. six degree of freedom) tracking information of four trackers. In the system implemented, two trackers are placed on the two players' heads to get their viewing positions, and the other two trackers are held by players as table tennis racquets. The SR 3D camera is mounted on top of the screen to capture the player's body movement in real-time based on the time of flight (TOF) principle, thereby enhancing the realism of the VR environment. The entire VR environment is projected onto a 2.06×2.74m stereoscopic screen for each player. By wearing a pair of polarised glasses, participants are able to enjoy the immersive experience. Furthermore, the key factor to achieve a realistic simulation is the physics model used for estimating the ball's trajectory. In this research, a detailed physics model is implemented which takes into account various physical phenomena such as air resistance, gravity, Magnus effect, ball spin, and frictions.

In order to assess the performance of the implemented VR system as a table tennis game environment, a user-based evaluation was conducted. In order to organise the issues addressed by previous literatures, a framework and heuristics of VR evaluation were developed. Based on the heuristics, questionnaires were developed as well and employed to investigate participants' perspective of the system usage. Both qualitative and quantitative data were obtained from the user-based test. By using statistics analysis, both the technology achievement and the degree of presence provided by the system were evaluated. In addition, the heuristics-based questionnaires were proved to be valid and reliable.

1.2 Aims and Objectives

The overall research aim is to design and implement a realistic real-time table tennis game for both single player and two players using the VR concept, which fulfils the requirements of immersiveness, interactiveness and competitiveness.

The specific objectives of the research are:

1. To develop and implement a complete VR hardware platform by integrating the tracking device, 3D camera capturing device and visualisation device.
2. To design and implement a communication workflow that can achieve fast and accurate transfer of data between multiple hardware systems.
3. To design and implement a virtual environment of the table tennis game according to measurement of the real objects.
4. To develop and implement a proper physics model to simulate realistic physical actions of the objects (i.e. ball trajectory, collision detection and responses).
5. To analyse and evaluate the performance of the VR environment implemented and compare with the state of the art systems.

1.3 Thesis Outline

The rest of the thesis is organised as follows. In Chapter 2, the background knowledge of VR is studied. Several VR related table tennis game systems developed by other researchers are reviewed and compared. Chapter 3 focuses on discussion of two crucial parts of an interactive and immersive VR environment, which are motion tracking and stereoscopic display. In Chapter 4, a single-player table tennis game is implemented which integrates the InterSense motion tracking system and a rear-projection stereoscopic screen. Additionally, the method of virtual objects generation and coordinate transformation are given. Furthermore, the physics based

ball animation model including collision detection and response are derived in details. As an extension of Chapter 4, a two-player table tennis game environment is implemented and described in Chapter 5. The communication strategy, coordinate transformation, displaying opponent and game workflow are presented in this Chapter. Furthermore, a systematic user-based evaluation of the proposed system is given in Chapter 6. Finally, conclusions and future work are given in Chapter7.

Chapter 2

Literature Review and System Design

2.1 Introduction

This chapter starts with an introduction of the concept and state of the art techniques of virtual reality. In particular, the researches and developments in the table tennis simulation and exergame applications are reviewed and compared. This is then led to the proposal and design of an immersive, interactive and competitive VR environment for a two-player table tennis game, as well as the technical requirements of the tracking and display systems needed.

2.2 Literature Review

2.2.1 Background of Virtual Reality

VR (Stanney, 2002; Sherman and Craig, 2002) is a computer generated simulation that replaces or augments the real world by various media. Different from conventional HCI (Human-Computer Interaction), such as GUI (Graphic User Interface), VR systems aim to provide participants a perception of “presence”, which can be described by two aspects: a sensation of “being there” (Bowman, et al., 2002; Chertoff, et al., 2010; Sylaiou, et al., 2010; etc) and an illusion of “non-mediation” (Lessiter, et al., 2001; Ijsselsteijn, et al., 2000; etc.).

“Being there” indicates a “subjective experience of being in one place or environment, even when one is physically situated in another” (Witmer and Singer, 1998), which can also be described by “immersion” in both spatial and psychological terms. In order to provide an visual immersive effect, it is not sufficient to provide visual experience through a large screen, various stereoscopic display systems are available for use, such as monitor-like FishtankTM (Demiralp, et al., 2006), tabletop-like Responsive WorkbenchTM (Grey, 2002), helmet-like HMD (Head Mounted Display) (Sherman and Craig, 2002), and surrounded-screen-like CAVETM (Cruz-Neira, et al., 1993). In order to create a complete immersive experience, it is also important to combine aural cues with visual cues to aid localisation of objects. Furthermore, spatialised sound can present objects outside a user's field of view (Funkhouser and Tsingos, 2004). On the other hand, since the sense of touch is a self-evident and natural experience in everyday life, integrating haptic feedback into a VR simulation enhances the degree of immersion (Magenat-Thalmann and Bonanni, 2006).

“Non-mediation” describes an ideal state of intuitive interaction, which means life-like manipulations without awareness of operating devices. Obviously, it can not be achieved by the traditional computer paradigm with unnatural input devices, such as keyboard and mouse. In order to implement natural interaction, tracking technology (Root, et al., 2010; Rolland, et al., 2001) becomes the core of human-computer interaction in the context of VR. Based on the acquired data from motion tracking systems, such as data gloves (Lu, et al., 2011), hand-held tracking stations (Wormell and Foxlin, 2003) and full-body motion tracking systems with or without markers (Weinland, et al., 2011), participants’ behaviour information can be identified and transferred to instructions.

Nowadays, all kinds of technologies in the VR field are developing rapidly. CVR (Collaborative Virtual Reality) supports interaction among multiple participants, and remote CVR can be implemented based on telepresence. AR (Augmented Reality) superimposes computer-generated objects on the real world, and MR (Mixed Reality) is a combination of both VR and AR. Although the history of VR is only about half a century old, its benefits have been extended to numerous application areas, such as simulator for aircraft cockpits and vehicles (Wan, et al., 2011), movement analysis for sports and rehabilitation (Bideau, et al., 2010), scene representation for archaeological sites and museums (Champion, et.al., 2011; Sylaiou, et.al., 2010), skill training for surgery operations and engineering repair (Cramer, 2004), data visualisation for scientific research and art design (Henry and Polysb, 2010; Grey, 2002), as well as entertainment (Zargarpour, et al., 2010; Liarokapis, 2006).

In the field of sports training, realistic sports simulations try to preserve the real world naturalness in order to provide a life-like experience to the players, which has distinct advantages if the real training environment is dangerous, or a peculiar sport environment is required. By using VR simulation, all environment parameters are under strict control, which is not achievable for the training in the real world. The skills acquired from virtual experience by players can be then transferred to the real

world (Li and Sun, 2009). Since the coherent body actions of athletes can be tracked and represented by using VR motion tracking and visualisation technology, VR is very useful in the analysis of biomechanics, physiology, and behavioural neuroscience (Bideau, et al., 2010). For instance, a competition environment can be simulated, where the athlete competes with an avatar of her/his real opponent. During this virtual competition, the sports performance, as well as physiological and psychological feedbacks of the athlete can be recorded and analysed.

On the other hand, movement-based sports game shows more enjoyable than standard exercises, and therefore largely engages people in physical activities (Marco, et al., 2009). In the interactive game field (Zargarpour, et al., 2010), exergame (named exertion game as well) (Sinclair, et al., 2007; Mueller, et al., 2011) is a merger of video game with physical exercise activities, and it normally employs tracking technology to acquire players' body motion information, which can be considered as a small scale VR. The big success of commotional exergame launched in recent years, such as games released for Nintendo Wii™, Microsoft Kinect™ for Xbox™ 360, and Sony Play Station™ 3, has demonstrated huge interests from people in innovative interaction devices. To provide more immersive experience, the new generation of exergame employs stereoscopic display and more precise tracking technologies, such as Astrojumper (Finkelstein, et al., 2011) and Swordplay (Katzourin, et al., 2006).

2.2.2 Current Table Tennis Simulation/Exergame

Among the previous research and development carried out, one of the most realistic table tennis simulation is V-Pong (Brunnett, et al., 2006), which enables real-time interaction between a player and a computer in an immersive virtual space. V-Pong employs a big stereoscopic rear-projection display screen and a set of marker-based infrared tracking systems. By attaching markers on the player's glasses and a real racquet, as well as fixing four infrared cameras on the screen's frame, the movements

of both racquet and player viewpoint are tracked.

Another single-player virtual table tennis game is AR-Table-Tennis (Park, et al., 2006). Based on image recognition technology, a square-shape marker with black and white pattern is attached on a real racquet, which is tracked by a video camera. This game supports both HMD and monitor display.

With respect to competitive games for two players, CamBall (Woodward, et al., 2004) is a network table tennis game. CamBall enables two remote players to see and play against each other through PC (Personal Computer) displays and web cameras. In the game, each player holds a real racquet with two rectangle coloured markers attached on both sides of the racquet. With image recognition technology, the spatial information of the two racquets is computed, and transmitted to the master PC. The video of each player is displayed on the opponent's monitor.

Another networked two-player game is Haptic Battle Pong (Morris, 2004). In the game, each player is seated at a PC and grasped a mechanical device with electrical haptic feedback. This device not only tracks the hand movement, but also simulates the impact force of collision. Due to the limited moving range of the mechanical device, each player's avatar needs to be moved to left and right sides through keyboard input. The same mechanical device with haptic feedback has been applied in a bouncing ball game (Bianchi, et al., 2006) as well, which enables one player to play with a virtual wall.

A multiplayer game can also be achieved locally through wireless Bluetooth communication, such as Virtual Ping-Pong (Kim, et al., 2007) designed for two-player competition. In order to display the views from three viewing directions (i.e. one is a "top-view" of the whole scene and the other two for the two plays, respectively), a big screen is split into three parts. Similar as V-Pong, optical tracking is employed in this game with four infrared cameras fixed on the screen and markers

attached on each racquet. In addition, force feedback is simulated by triggering a direct current motor that is attached on each racquet as well.

As mentioned previously, exergame in the commercial video game industry has achieved big success. Take one of the best selling consoles, Nintendo Wii for example, its motion sensing controller is based on both inertia and infrared optical tracking technology, and it has vibration feedback. In a popular table tennis game in the Wii-Sports serial, players often stand up in front of TV and physically swing their arms.

With rapid development of smart-phones, there is a new trend of mobile phone exergame. SymBall (Hakkarainen and Woodward, et al., 2004) is an interactive table tennis game based on the Bluetooth communication, which enables two players to play against each other face-to-face by swing their mobile phones as racquet control. Any object that has simple shape and distinct colour in the surrounding real world can be appointed as a motion tracking reference, and mobile phone cameras track themselves according to this reference.

Table 2.1 compares the features of the above applications. In order to highlight these features, the advantages and disadvantages are coloured by blue and gray, respectively.

	Immersive Display	Hand Tracking	Head Tracking	Multiple Players	Other Features
V-Pong	Yes	Yes	Yes	No	High Reality
AR-Table-Tennis	Yes	Yes	No	No	-
Haptic Battle Pong	No	Yes	No	Yes	Force Feedback
Virtual Ping-Pong	No	Yes	Yes	Yes	Force Feedback
Wii-Sports (Table Tennis)	No	Yes	No	No	Vibration Feedback
SymBall	No	Yes	No	Yes	-

Table 2.1 Features of Seven Table Tennis Games

Each application has its own advantages and contributes to different areas of the research or entertainment market. However, from the view of realistic simulation, V-Pong is the best one. In order to achieve “high reality”, there are some essential factors, which are stereoscopic display with wide angle of view, precise racquet and viewpoint tracking, as well as realistic animation based on physics. In addition, haptic feedback enhances the immersive effect, and good scene design makes the game more attractive.

2.3 Proposed VR Environment

2.3.1 Design of VR Environment

The design of a VR environment can be described by the following four steps:

1. Specify a VR content
2. Specify a virtual environment, and the degree of immersion (for immersive VR environment)
3. Specify the method of interaction between participants and virtual objects, and the degree of the interaction (for interactive VR environment)
4. Specify the method of interaction among multiple participants, and the degree of the interaction (for collaborative or competitive application)

The proposed VR content is a real-time table tennis game for two players. Through the review of previous works described in Section 2.1.2, there are rooms for further development by combining immersive, interactive and competitive features. Since playing table tennis requires accurate spatial information of the ball and natural control of the racquet, there are challenges to implement such a system with good

tracking quality, accurate 3D effect, low response latency, and realistic ball movements.

To design a virtual environment, a 3D scene with virtual objects needs to be created. For a table tennis game, the basic virtual objects consist of a table tennis ball, two racquets (for two-player game), and a table tennis table with net. The scene can be as simple as a room. When playing table tennis, players need to estimate the ball's position based on its moving direction and speed, which requires an immersive representation of spatial information to give participants an illusion of locating in the virtual environment and co-locating with virtual objects. In order to achieve such an illusion, immersive visual experience needs to provide what are normally referred to as "sense of depth", "wide viewing angle" and "correct viewpoint". Obviously, a 2D display can not give depth information, and a display with a fixed viewpoint will distort the depth perception if the observer is moving. Therefore, 3D display, tracked viewpoint, and wide viewing angle are the basic requirements to display the virtual environment.

Interaction refers to the ability of modifying the VR environment by participants. In this case, players need to interact with the virtual ball by controlling their virtual racquets, which is the most essential interaction in a table tennis application. Therefore, the virtual racquets need to be tracked in real time. Furthermore, the feedback of the action is expected to be life-like. For instance, if a ball goes through a racquet, or disappears at the moment of colliding, players would have unnatural experience with respect to interaction. Therefore, a physics based ball animation model has to be established, and a robust real-time collision detection method needs to be developed as well.

Collaborative or competitive interaction can be achieved if more than one participant interacts with same virtual objects in a shared VR environment at the same time. For instance, in a table tennis game, two players are located in a same virtual room, and

try to hit the same virtual ball by controlling their own virtual racquet. Furthermore, displaying opponent can help players to estimate the ball's trajectory, as well as enhance game's attractiveness, which can be implemented by full-body (or upper-body at least) tracking technology.

Based on the four design steps, the proposed VR environment and the system requirements are summarised in Figure 2.1. In detailed requirements, the three items in light blue squares require motion tracking technology, the item highlighted by yellow refers to the need of a display system, and all items highlighted by light green can be implemented through software development.

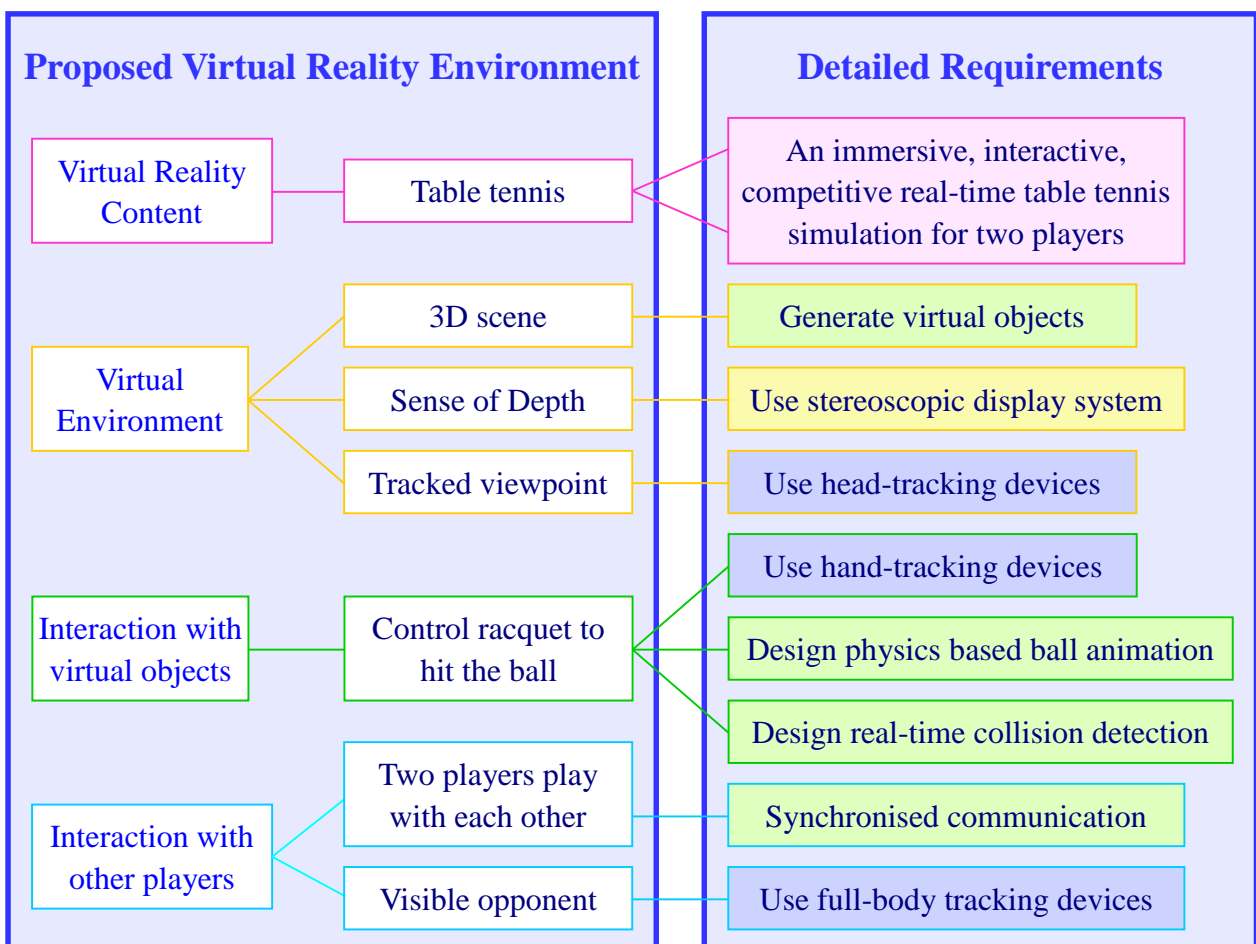


Figure 2.1 Requirements of Proposed Table Tennis Game

2.3.2 Requirements of Motion Tracking System

Motion tracking is a process of encoding motion from the real world into the digital medium in three dimensions (Root, et al., 2010). There are many types of tracking technologies, and each of them has its own advantages and limitations. Therefore, the selection should be based on what needs to be tracked. In the case of the proposed system, accurate spatial information of head and hand are required in real time, whereas a rough player posture is required in order to display the body movements as “visible opponent” in real-time. Therefore, two kinds of motion tracking approaches need to be employed.

2.3.2.1 Head and Hand Motion Tracking

When playing table tennis, the racquet is controlled in high-speed with complicated movements including real-time direction and rotation changes. Therefore, only a system capable of real-time tracking in 6-DOF (translation along x, y and z axes, and orientation by yaw, pitch and roll) with high update rate is adequate for this application. Furthermore, a tracking technique affected by line-of-sight occlusion is not a good selection in this case due to the variable hand gestures that may result in significant occlusions, especially for tracking two players at same time. Although these occlusions can be minimised by employing a large number of cameras, it incurs a high cost. Since tiny changes on racquet’s movement may result in a totally different ball trajectory, the precision of tracking is also an essential requirement. If tracking devices restrict users from playing, the effects of immersion and natural interaction will be badly reduced. Therefore, portable equipments without restricted cables are needed.

With respect to head tracking, the basic requirements are the same as hand tracking. Although the required levels of speed and precision for head tracking are not as high as for hand tracking, any visible delay or tracking error may result in uncomfortable navigation experience. On the other hand, in the situation of using fixed screen display, the rotation data obtained from head tracking are not used (will be explained in Chapter 4). Therefore, 3-DOF (translation along x, y and z axes) data needs to be collected during head tracking.

2.3.2.2 Motion Tracking for Visible Opponent

As explained in the previous section, “visible opponent” aims to help players to judge a ball’s trajectory by observing their opponent’s movements, and to increase the sense of presence perceived by players as well. A widely used solution is to capture the movements of a player’s main body joints or segments, and then generate a skeleton-based avatar based on the positions of tracked joints / segments (Herda, et al., 2001). Optical tracking technology is usually employed for this method, and the tracked joints / segments are identified by image processing methods, such as marker detection (Pintaric and Kaufmann, 2007) and feature extraction (Doshi, et al., 2008). However, line-of-sight occlusion may become serious due to the high-speed variation of players’ motion. Consequently, it becomes too complicated to interpolate occlusion parts and distinguish different body joints / segments for a real-time two-player application. Although this complexity can be reduced by increasing the number of tracking cameras, it is costly and requires a big space.

Actually, the calculation to obtain accurate spatial information of body joints or segments is not necessary. The movements of a player can be displayed in the virtual environment directly by recording a 3D video (Smolic, 2011). Combined with stereoscopic display technology, 3D video can offer a 3D depth impression of the observed scene (Smolic, 2011). This approach has the advantage of high-speed and

hence is more suitable for real-time applications. Furthermore, display of a real player in a 3D video will be more realistic compared with a computer generated avatar. Therefore, the approach of recording and playing 3D video in real time has obvious benefits in this case.

2.3.3 Requirements of Immersive Display System

A 3D display system should be capable of presenting a stereoscopic perception of 3D depth to the players. Based on the principle of binocular disparity, the basic requirement of a 3D display device is to generate two images from the two slightly different projections of the world to the left and right eye separately (Lipton, et al., 2010)

Two types of 3D display technologies are commercially widely used: stereoscopic and autostereoscopic (Dodgson, 2005). The stereoscopic technology is employed for familiar 3D films and most of VR display devices, such as HMD and CAVE. The new game console 3DS produced by Nintendo Company and most of glasses-free 3D televisions are based on autostereoscopic technology. The difference between them is on whether using special glasses or headgear to filter images for two eyes. Since a stereoscopic screen has better display quality and relatively cheaper price compared with an autostereoscopic screen of the same size and pixel resolution, it was employed for this project.

On the other hand, the display system cannot disturb the participants from playing. Therefore, fixed screen display system is more appropriate than HMD that is usually not portable. To provide a wide angle of view, the size of the screen should be big enough. Moreover, it should have good display quality and enough brightness for in-door environments.

2.4 Conclusion

The concept and state of the art techniques in the VR field are introduced firstly, especially the advantages of applying VR in the fields of sports and games. Through a review of current table tennis simulation and exergame, an immersive, interactive and competitive real-time VR environment for two-player table tennis game is proposed.

In order to implement the proposed application, the basic hardware equipment should include an immersive 3D display system, a precise 6-DOF hand and head motion tracking system, and a body movements tracking system. The detailed requirements are shown below.

Requirements of Head and Hand Tracking

1. 6-DOF for hand-tracking and 3-DOF for head-tracking
2. High precision
3. High tracking speed
4. High portability without restriction
5. Enough tracking range
6. No line-of-sight occlusion for hand tracking

Requirements for Tracking Visible Opponent

Body tracking based on 3D video

Requirements for Display System

1. Stereoscopic display
2. No disturbance to players
3. Wide viewing angle (Big size)
4. Good display quality and enough brightness

Chapter 3

Motion Tracking and Stereoscopic Display

3.1 Introduction

This chapter focuses on two crucial parts of an interactive and immersive VR environment, which are motion tracking and stereoscopic display. As introduced in Chapter 2, motion tracking is fundamental to achieve intuitive interaction, whereas stereoscopic display creates an illusion of 3D depth leading to immersive perception.

Since there are various commercial products for motion tracking and stereoscopic display based on different kinds of operation principles, a review of current approaches and the justification of the systems selected for this project are given in this chapter, and it includes a detailed presentation of each system employed, their setup in terms of hardware and software, and their performance.

3.2 Hand and Head Motion Tracking

3.2.1 Current 6-DOF Motion Tracking Technology

A variety of tracking technologies are available for capturing 6-DOF motion data for real-time applications. Since each of them has its own advantages and disadvantages (Rolland, et al., 2001; Root, et al., 2010), it is necessary to review current approaches in order to select an appropriate technology for the proposed application.

Optical tracking relied on image processing techniques can achieve a very high resolution and accuracy. Nevertheless, since orientation information of a tracked target can not be captured directly, the post-processing of 3D reconstruction is required to obtain 6-DOF data. Furthermore, optical tracking is sensitive to optical noise and suffers from line-of-sight occlusion. In order to achieve better tracking quality, the complicated calculations may be required, which results in slow tracking speed. Therefore, the performance of optical tracking is highly depended on different approaches, and it is more suitable for a case without line-of-sight occlusion. Mechanical linked tracking that has good tracking precision and speed is immune from line-of-sight occlusion. It has a distinct advantage of being easily combined with haptic force feedback devices. However, its tracking volume is normally very limited due to the limited length of mechanical arms. Although ultrasonic tracking provides wider tracking range, its tracking accuracy tends to reduce with the increase of transmitting distances. Compared with other approaches, ultrasonic tracking has relative low performance in terms of precision and speed, and it is sensitive to acoustic interferences. However, it has advantage of low cost. Similar as ultrasonic approach, the accuracy of magnetic tracking would also fail rapidly with distance. Although magnetic techniques provide relatively high speed and precise tracking, it has the drawback of suffering metallic interference and high cost. Inertial tracking is

immune from most of interferences. It has very high update rate and high resolution. However, the tracking error of an inertial device would become more and more serious over time since the measurements are relative to previous tracking results leading to reference drifts.

Since none of the above approaches is able to meet all of the five major requirements (i.e. 6-DOF tracking, high precision, high tracking speed, high portability without restriction, enough tracking range, and no line-of-sight occlusion) described in Section 2.2, hybrid tracking (Rolland, et al., 2001) was considered. Hybrid motion tracking refers to using a combination of multiple sensors based on different tracking technologies, which can overcome the weaknesses of each individual tracking method, and results in more accurate motion data and higher speed. Since a hybrid of ultrasonic and inertial sensors is relatively low cost and immune from line-of-sight occlusion, it was adopted in this project.

3.2.2 InterSense Motion Tracking Technology

3.2.2.1 System Introduction

Based on all the above considerations, the InterSense IS-900TM Motion Tracking System (Wormell and Foxlin, 2003) was employed in this project for both hand and head motion tracking. This system is able to acquire accurate 6-DOF motion data in real time based on a combination of inertial and ultrasonic tracking technologies. Such a hybrid approach can not only constantly correct the reference drifts associated with inertial tracking, but also avoid inconsistent accuracy associated with ultrasonic tracking. Furthermore, it offers tracking devices (stations) with small physical size

which could be easily carried by users, and it is immune from line-of-sight interference.

In addition, the InterSense system provides various tracking stations, which can be attached onto targets to be tracked or held by users directly. Since players can hold a hand tracking station as holding a table tennis racquet, the InterSense system is especially suitable for the case of racquet tracking. The InterSense head tracking stations are lightweight, which can be fixed on a head mounted frame or even a normal cap. Furthermore, the tracking stations employed in this project are wireless, which provide good portability and flexibility.

3.2.2.2 System Configuration

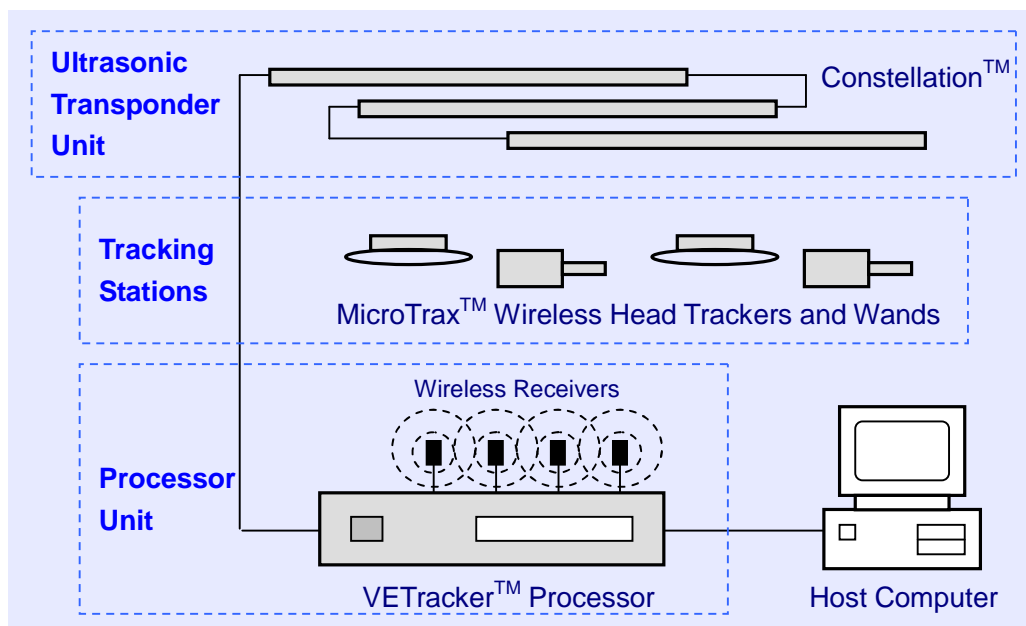


Figure 3.1 IS-900 System Configuration

The IS-900 hardware is made up of an ultrasonic transponder unit, tracking stations and a processor unit, and their setup configuration is illustrated in Figure 3.1. Constellation™ (Foxlin, et al., 1998a) is the name of a ceiling-mounted structure, which is formed by ultrasonic transponder beacons called SoniStrips™. Under this structure, these beacons are connected in series, and finally plug into a VETracker™

processor. Two kinds of tracking stations were employed in this project, which are MicroTrax™ Wireless Head Tracker and MicroTrax Wireless Wand for head and hand motion tracking, respectively. With one of each station for each player, a total of four stations are required for a two-player game. The VETracker processor is responsible for equipment control, data transfer, and tracking calculation, which is connected to a host computer via a serial link port. The photos of the MicroTrax Wireless Head Tracker and Wand, VETracker Processor, and Constellation are shown in Figure 3.2. The buttons on the wand can be used for transmitting specific commands to the processor.

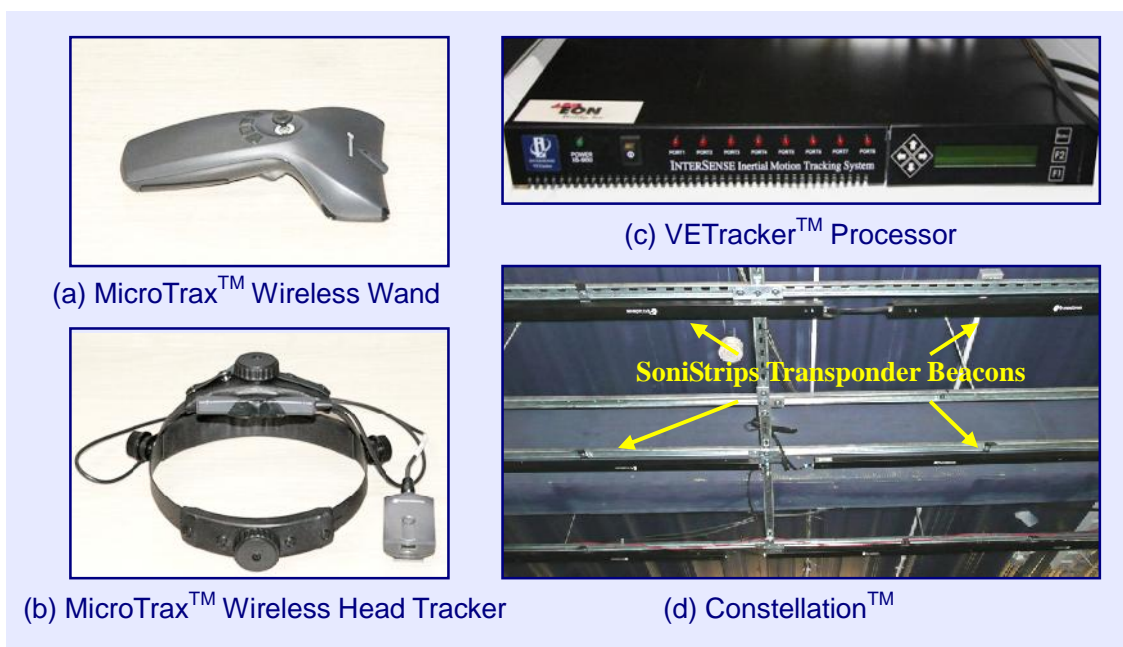


Figure 3.2 IS-900 Equipment

3.2.2.3 Operation Principle

Figure 3.3 illustrates the operation block diagram of the IS-900 system. URM (Ultrasonic Rangefinder Module) and IMU (Inertia Measurement Unit) are two major tracking components installed in each tracking station (Foxlin, et al., 1998a). While the former is mainly composed of ultrasonic receiver microphones and TOF

(Time-of-Flight) counters, the latter refers to a miniature inertial device consisted of accelerometers and gyros.

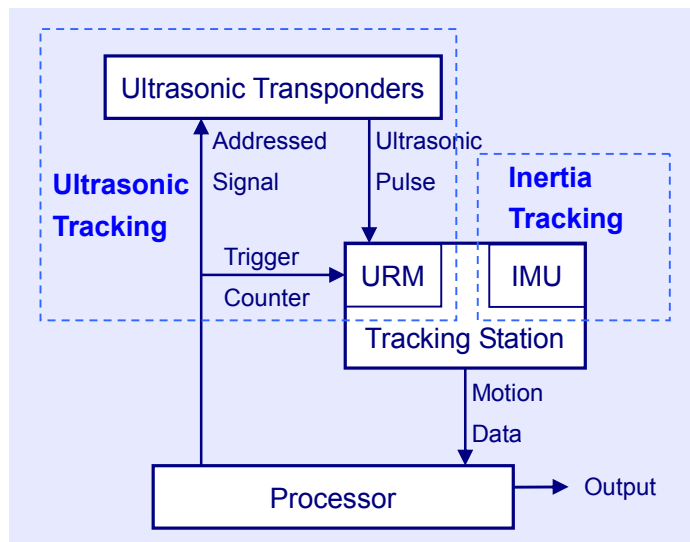


Figure 3.3 IS-900 System Operation

The ultrasonic tracking part shown in Figure 3.3 is based on the acoustic TOF principle (Rolland, et al., 2001; Root, et al., 2010), which provides a measure of distance by multiplying sound speed and sound travelling time. This technology usually has the advantage of high-speed, and the weakness of unstable accuracy due to interferences and distances. For distance measurement, an ultrasonic emitter on the transponder beacons transmits an ultrasonic pulse once it receives an addressed signal from the processor. At the same time, a TOF counter in URM is triggered by the processor as well (by wireless communication). The counter is halted once the pulse is detected by a receiver microphone in URM. Consequently, the distance between the emitter and receiver is calculated from TOF. With three TOF measurements with respect to three different emitters, the 3D spatial position of a receiver microphone can be determined. There are two receivers in a MicroTrax Wireless Head Tracker, and four in a MicroTrax Wireless Wand to prevent signal interference due to quick and complex hand motions.

The inertia tracking part shown in Figure 3.3 measures angular rate and linear acceleration by using gyroscopes and accelerometers (Foxlin, et al., 1998b; Rolland,

et al., 2001), within the IMU based on the conservation of momentum principle. Since each gyroscope or accelerometer is only able to measure along one axis, three sets of each are installed in the IMU to cover all 3 axes. With the previous motion state of a station known, the current position, orientation and velocity can be calculated based on detected angular rate and acceleration.

The IMU inner drift correction is accomplished by fusing the IMU output and the output obtained from the URM. Through the drift correction, the motion capture has consistent accuracy over entire tracking volume. As shown in Figure 3.3, accurate 6-DOF motion data are calculated and transmitted to the processor for final output. Furthermore, since shorter transmitting distance provides better tracking quality, the current detected position data are used by the processor to generate appropriate addressed signals (i.e. choosing emitters near the tracked station).

3.2.3 6-DOF Motion Tracking Results

3.2.3.1 Motion Tracking Output

A specific software tool named ISDEMO developed by InterSense Company enables system configuration and displaying output data. With the reference system of these 6-DOF data illustrated in Figure 3.4, Figure 3.5 shows the GUI of ISDEMO with the motion tracking information of a MicroTrax Wand, where a 3D model of the tracker is displayed to show its orientation.

For development of new applications with user-defined interface, the InterSense API (Application Program Interface) is also available. This API contains a dynamic link library (DLL) “isense.dll” file written in C++ that supports communication with all

types of the InterSense tracking devices. The import procedures of DLL are written in the “isense.cpp” file, and a header file “isense.h” defines data structures and function prototypes.

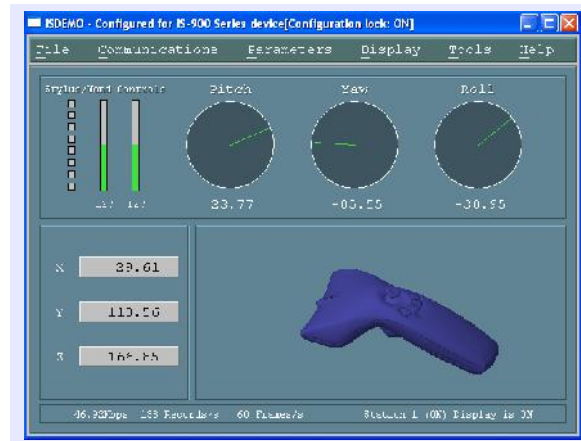
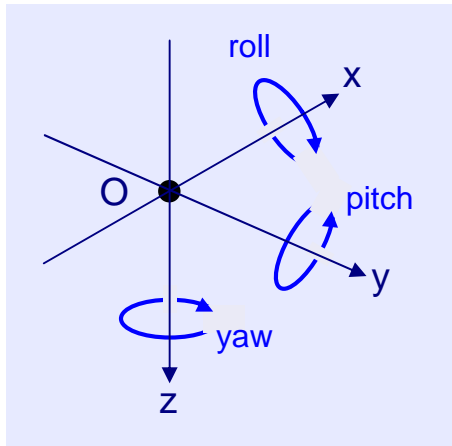


Figure 3.4 InterSense Reference System **Figure 3.5 Screenshot of ISDEMO**

A C++ sample code file used for general data stream is provided as well. Figure 3.6 shows the data output stream, which is the running result of a modified sample code file. In Figure 3.6, the first three columns indicate the tracker’s orientation in degree by Euler angles (i.e. roll, pitch, and yaw), and the following three columns show the position values in metre along each of the Cartesian coordinate axes (i.e. x, y, and z), respectively. The reference system of these rotation and translation data is demonstrated in Figure 3.4. Furthermore, the last column indicates which tracker being measured (with 0 denoting a MicroTrax wand), and the number shown in the seventh column indicate which button on the wand being pressed.

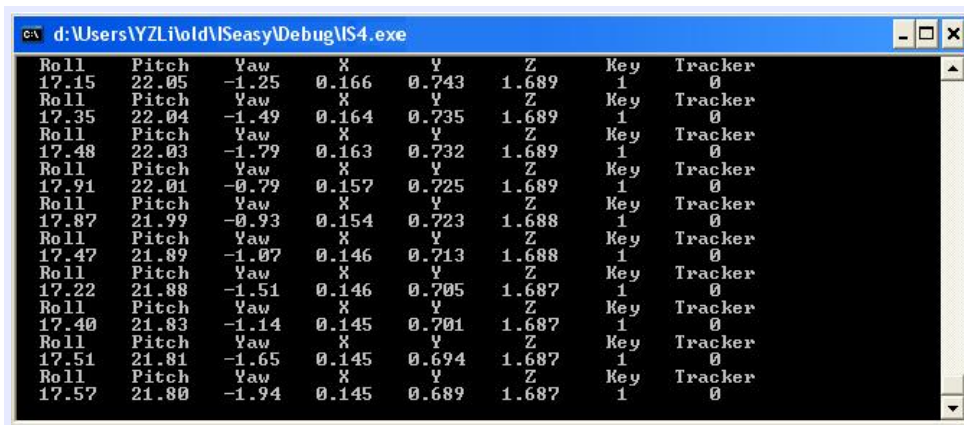


Figure 3.6 Screenshot of InterSense Data Stream

3.2.3.2 Motion Tracking Performance

The performance specifications of the InterSense IS-900 system are shown in Table 3.1, which is given by the InterSense user manual.

Main Specification	IS-900 MicroTrax Wireless Trackers
DOF	6 (X/Y/Z and Yaw/Pitch/Roll)
Update Rate	120Hz in real time
Latency	4ms typical
Resolution	translation: 1.5mm & rotation: 0.10°
Accuracy	translation: 3.0mm to 5.0mm rotation: 0.50° in Pitch & Roll, 1.00° in Yaw
Weight	Head Tracker: 40g & Wand: 70g

Table 3.1 IS-900 Performance Specifications

In evaluation of the performance specifications, ISDEMO was used. For translation performance, the tracker was moved vertically (i.e. up and down) and horizontally along the two orthogonal directions (i.e. forward and back, left and right), the position values along x, y, and z axes shown in ISDEMO were checked to see if they were changed along correct directions. To evaluate the translation accuracy, several anchor points were lined up with 10cm interval in the laboratory along each axis. With the tracker placed on these anchor points and the displayed position data in ISDEMO recorded. The distances between every two neighbouring anchor points were found to be correct at around 10cm with errors no more than 1cm. In evaluation of orientation, the tracker was rotated along different directions and the 3D model in ISDEMO was found to rotate with the tracker in the same way, and with the correct numerical angles displayed as well.

Since the quality of a received ultrasonic signal is affected by the relative position of the receiver microphones with respect to the Constellation, the tracking quality would drop rapidly if a tracker goes away from the tracking volume. A tracking quality (TQ) value is shown on the LCD screen of the processor to indicate an overall measure of

current tracking quality. The values are represented by a percentage from 0 to 100%, which correspond to a range from loss-of-tracking to tracking perfectly. A value of 80% or higher indicates good tracking quality, and a value lower than 40% indicate a relatively high probability in loss-of-tracking.

In evaluation of tracking volume and quality, a test was carried out in the laboratory with a 2m×4m Constellation. When a tracker was placed no lower than 1.5m under the Constellation, the TQ value was in a range of 78% to 98% most of time. However, the value was only about 40% if the tracker was at 2m under the Constellation, and it dropped down to 10% or even 0 rapidly if the tracker was moved down continuously. Furthermore, if the tracker was placed at 0.5m away horizontally from the Constellation's volume, the TQ value was about 40%, and it dropped down to 0 rapidly if it was placed even further. Therefore, the tracking volume with good performance is 2m×4m×1.5m, and the working volume is 3m×5m×2m.

Although a VETracker Processor is said to be capable of tracking 8 trackers at the same time, the tracking quality was found to decrease with the increasing number of trackers. Through evaluation, a good tracking quality was found to be achieved for 2 trackers. However, if there were 3 or 4 trackers working together, the TQ values were unstable. Typically, if all the four trackers were working within the good tracking volume, two of them had a TQ value around 90%, and another two were only about 60% or even lower.

Furthermore, covering of the receiver microphone was found to affect the tracking quality. The tracking quality was not affected if the handle part of a wand was grabbed by user, or a head tracker was fixed on a head mounted frame. However, if the top part of a wand or the two sides of a head tracker were covered, loss-of-tracking was found to occur due to occlusion.

In terms of real time motion capture, an update rate of each tracker at 116 to 138 times per second was observed from ISDEMO throughout the motion tracking performance evaluation.

3.3 Body Motion Tracking

3.3.1 Current 3D Video Motion Tracking Technology

A commercially widely used method for producing 3D video of motion is stereoscopic photography, which usually employs two 2D video cameras to record from two perspectives corresponding to views from left and right eyes. However, the generated video does not contain any 3D depth information, though it is usually called 3D due to its stereoscopic effect. Therefore, the viewpoint of this kind of video is fixed by the position of the two 2D video cameras and can not be changed arbitrarily (Smolic, 2011).

To realise arbitrary viewpoint display, the 3D depth information of the observed targets need to be acquired. The operation principles of current 3D cameras are mainly based on two approaches: Triangulation and TOF. Triangulation (Büttgen and Seitz, 2008) refers to geometric calculation based on computer vision methods, which can create a precise 3D depth map from two or more captured 2D images (Ogi, et al., 2003). Nevertheless, for a real-time capture with complex scenery, the complicated calculations of this approach would result in low update rate. For instance, the BumblebeeTM camera family produced by PointGreyTM Company based on the triangulation principle has frame rates of 20fps and 15fps for two and three cameras, respectively. If the rendering time is also taken into account, the update rate is too slow to generate smooth animation in real time. For TOF based on an optical

approach (Root, et al., 2010), the operation principle is similar as the acoustic TOF device described in Section 3.2.2.3, which has the advantage of high-speed and disadvantage of inconsistent accuracy. In the case of showing 3D body motion of an opponent in this project, small tracking error is acceptable as long as it does not affect the gameplay. Therefore, a 3D camera based on high-speed optical TOF technology is more appropriate for this project.

3.3.2 SwissRanger Motion Tracking Technology

3.3.2.1 Camera Introduction

The SwissRanger SR4000TM 3D camera produced by MesaImagingTM Company is capable to acquire depth and shape information of objects in the measurement volume in real time (Hegde and Ye, 2008; Guomundsson, et al., 2008). Based on the depth map acquired, a 3D surface of the captured scene with objects can be generated, thereby enabling visualisation from different viewpoints.

The SR4000 camera is based on the infrared TOF technology. Comparing with acoustic TOF devices, infrared measurement has an additional advantage of acquiring grayscale 2D image sequences by light amplitude measurement (i.e. recording light intensity) at the same time of capturing 3D depth information. Therefore, the SR4000 camera can also be used to show “opponent’s appearance” by greyscale video in this project.

3.3.2.2 Camera Setup and Operation Principle

The SR4000 camera can be simply connected to a host computer running the driver software through a USB cable. Figure 3.7 shows the photo of a SR4000 camera.

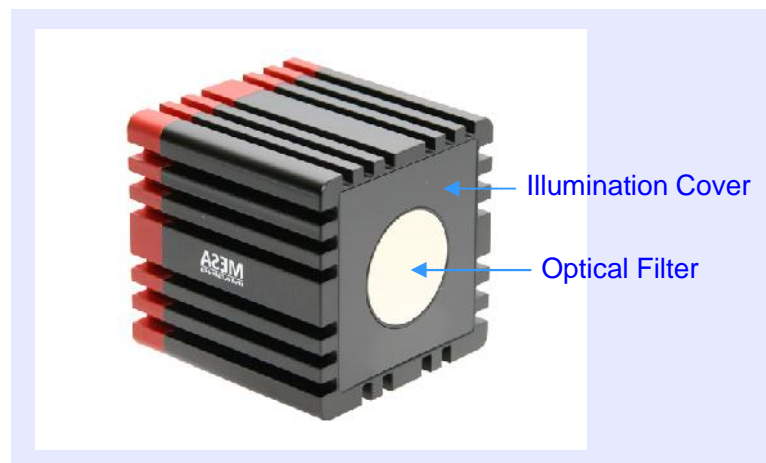


Figure 3.7 SR4000 Camera

The basic operation principle is infrared based phase-measuring TOF (Büttgen and Seitz, 2008). Different from the method of measuring the actual time of light travelling from source to sensor, it measures the phase delay between the arrived and emitted light waves. To realise the phase measurement, the light source is modulated and its amplitude is cycled in a sinusoidal pattern. Then the CCD/CMOS imaging sensor in the camera samples the incoming returned sinusoidal signal four times per modulation period to calculate the phase shift. The distance corresponding to a full cycle of the modulated signal is the maximum measuring distance. In the case of SR4000 camera, the default modulation frequency is 30MHz. Therefore, the measuring distance is limited at 5m based on Equation 3.1.

$$D_{\max} = \frac{c}{2f_{\text{mod}}} \quad (3.1)$$

where f is the modulation frequency, and c is the speed of light (about $3 \cdot 10^8$ m/s) (“ \cdot ” indicate multiply). The relationship of phase shift and light’s travelling distance is given by

$$D_{measure} = \left(\frac{c}{4ff_{mod}} \right) \{ \text{shift} \} \quad (3.2)$$

The SR4000 camera operates with 24 near-infrared (NIR) light emitting diodes (LEDs), which emit NIR rays to illuminate a target volume. The rays are reflected by the scene and imaged via an optical lens onto sensors within the camera. To collect these NIR rays without noisy lights, an optical filter is mounted on the front of the camera to allow only light with required wavelengths (i.e. near the LEDs' wavelength) to pass into the camera. As shown in Figure 3.7, the circular part is the optical filter, and the area outside the circular part is the LED illumination cover.

Figure 3.8 illustrates the imaging process. The red point shown in Figure 3.8(a) is a spatial point within the illumination volume. As a result of reflecting the NIR ray from the camera, this spatial point is captured as an image pixel in the image plane within the camera, with its value corresponding to the ray-transmitting distance calculated based on TOF. Since each pixel on the image plane has its own depth value (radial distance), a depth map is therefore generated as shown in Figure 3.8(b). The resolution of the acquired depth map is 176×144 corresponding to 43.6°×34.6° field of view.

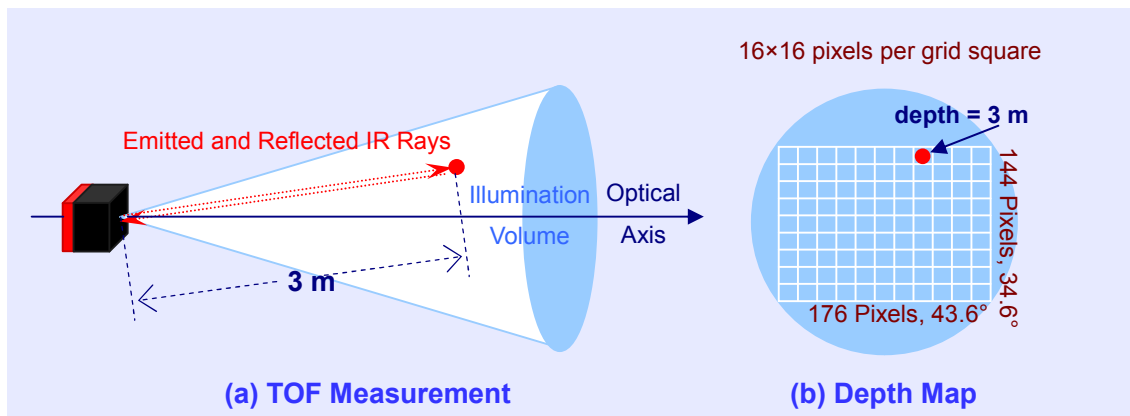


Figure 3.8 Operation Principle of SR4000 Camera

3.3.3 3D Body Motion Tracking Results

3.3.3.1 Motion Tracking Output

The Mesa Imaging Company provides a SR_3D_Viewer for continuous camera acquisition and display of the acquired 3D image sequences in real time. An example output is shown in Figure 3.9, where the depth information of the scene represented by false colours from blue (near to camera) to red (far from camera) is displayed in the main window, and a grayscale image sequence generated by the acquired intensity value of each pixel is displayed in the left panel.

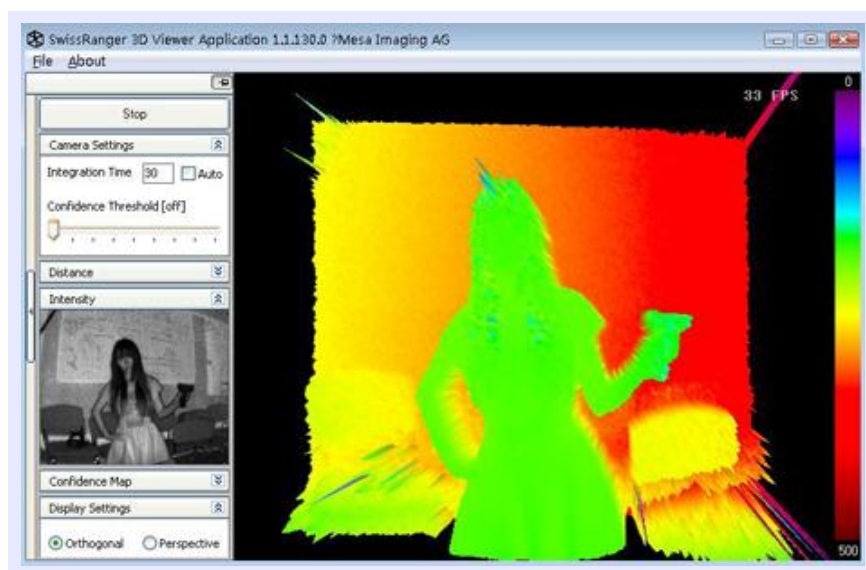
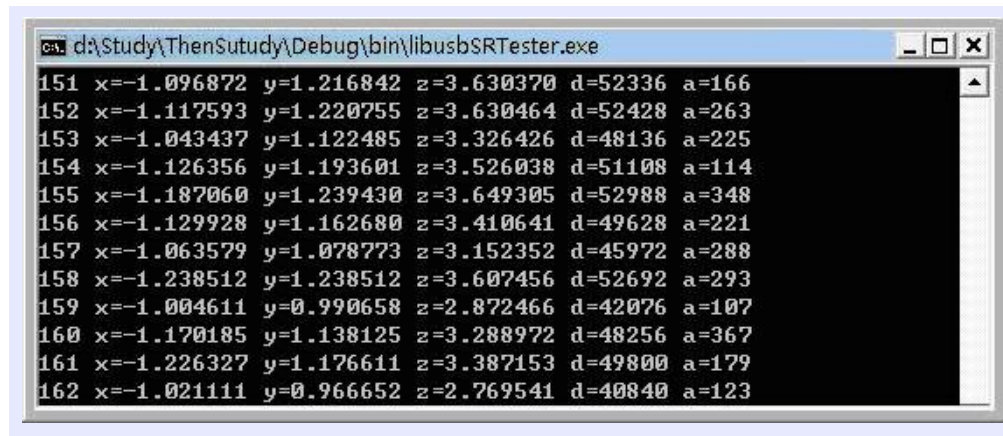


Figure 3.9 Screenshot of SR_3D_Viewer

The SwissRanger API for application development is provided as a DLL file “libusbSRTTester.dll” in Windows with a C++ header file “libusbSRTTester.h” for function declarations. This API includes extensive functions, such as median filter for noise reduction and intensity compensation for better grayscale image. The two most important functions are setting integration time (to be discussed in Section 3.3.3.3) and the coordinate transformation that converts the raw radial distance to Cartesian coordinates for each detected spatial point based on camera parameters. The output

data sequence in each frame starts from the pixel at the top-left corner, and then goes through all 144 pixels in each row, and entire 176 columns from top to down.

Modifying the sample code file written in C++ provided for data acquisition, Figure 3.10 shows the data output stream. The first column indicates the number of the detected pixel, which is followed by the position values along x, y and z axes, expressed in metres. The last two columns, "d" and "a", provide the raw radial distance and amplitude (i.e. intensity) value, respectively. Both "d" and "a" are in a range from 0 to 65535 (i.e. $2^{16}-1$).



```
cmd: d:\Study\ThenSutudy\Debug\bin\libusbSRTTester.exe
151 x=-1.096872 y=1.216842 z=3.630370 d=52336 a=166
152 x=-1.117593 y=1.220755 z=3.630464 d=52428 a=263
153 x=-1.043437 y=1.122485 z=3.326426 d=48136 a=225
154 x=-1.126356 y=1.193601 z=3.526038 d=51108 a=114
155 x=-1.187060 y=1.239430 z=3.649305 d=52988 a=348
156 x=-1.129928 y=1.162680 z=3.410641 d=49628 a=221
157 x=-1.063579 y=1.078773 z=3.152352 d=45972 a=288
158 x=-1.238512 y=1.238512 z=3.607456 d=52692 a=293
159 x=-1.004611 y=0.990658 z=2.872466 d=42076 a=107
160 x=-1.170185 y=1.138125 z=3.288972 d=48256 a=367
161 x=-1.226327 y=1.176611 z=3.387153 d=49800 a=179
162 x=-1.021111 y=0.966652 z=2.769541 d=40840 a=123
```

Figure 3.10 Screenshot of SR4000 Data Stream

3.3.3.2 3D Video Generation

The 3D surfaces were generated by OpenGL (Shreiner, et al., 2009; Hearn and Baker, 2004; Wright and Lipchak, 2004) in this project. Since OpenGL supports smooth shading that realises colour interpolation automatically when a triangle is drawn with a different colour specified for each vertex. By setting the intensity value of each pixel to the corresponding vertex, the acquired grayscale image is mapped on the created 3D surface as shown in Figure 3.11. Since the 3D surface of the scene is updated in real-time, it enables the body motion to be observed as a 3D video.

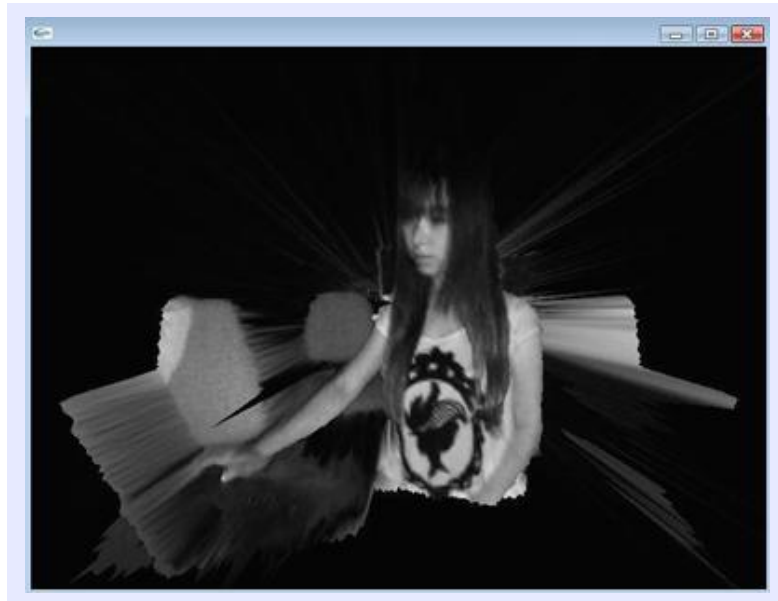


Figure 3.11 Screenshot of Generated 3D Video

3.3.3.3 3D Video Performance

Table 3.2 shows the main specifications of the SR4000 camera, which are given by the SR4000 data sheet.

Main Specification	SR4000 Camera
Pixel Array Size	176(h)×144(v)
Field of View	43.6°× 34.6°
Frame Rate	up to 54fps
Operating Range	0.3m to 5.0m
Distance Accuracy	± 1cm
Pixel Pitch	40μm
Angular Resolution	0.23°
Illumination Wavelength	850nm

Table 3.2 SR4000 Camera Performance Specifications

With motion tracking relying on the infrared TOF technology, the tracking stability is sensitive to the infrared reflection characteristics of the objects and their distances in

the illumination volume. Hence, the greater the amplitude of the reflected infrared light, the less is the noise appeared in the depth map.

To reduce noise, objects with directed reflecting materials (i.e. glossy surface, retro-reflector, or mirror in the extreme case) should be removed from the scene unless they are employed as markers. Besides, since light intensity decreases with distance, the objects located further away usually result in lower tracking quality. For objects more than 5 metres away from the camera, a back-folding phenomenon occurs with objects further away from the camera appearing as objects nearer to the camera, due to the periodicity of the signal being used for distance measurement. Figure 3.12 shows an example of such situation resulting in a noisy depth map with some extreme depth values (e.g. zero and near to zero depth as shown by the purple colour in the figure).

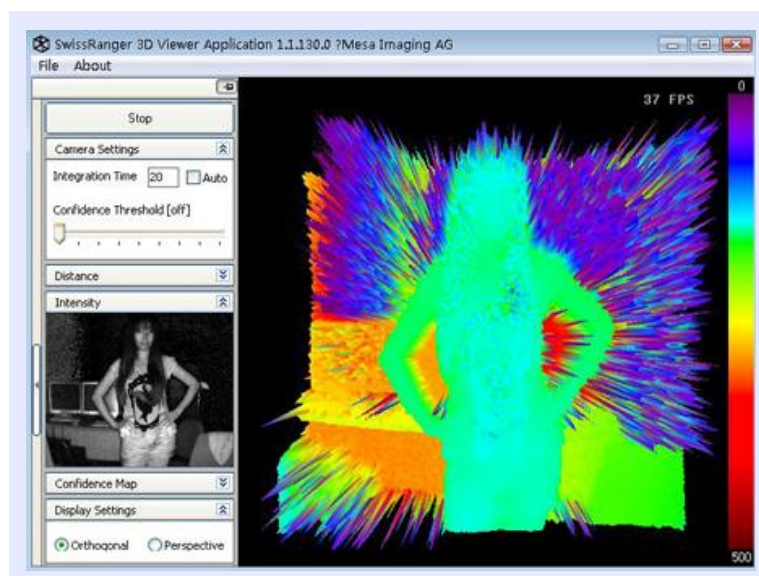


Figure 3.12 Sample of Background Noise

Furthermore, since the reflected signal is sampled at four times its period for the phase measurement, if a detected object is moving during this period, noise is introduced at the edge of the object.

The tracking quality of a SR4000 camera highly depends on the setting of an important parameter named “integration time”, which denotes how long the pixels are

allowed to collect light. Since the best tracking quality can be achieved if the greatest signal amplitude is reached without saturation, longer integration time that allows collection of a larger amount of reflected lights results in better tracking quality with lower noise level. However, longer integration time slows down the tracking speed, and increases the noise at the edge of a moved object due to the longer phase-measurement period. Therefore, a compromise between tracking quality and tracking speed, as well as the reflection characteristics and movement speed of measured objects, should be taken into account for setting an optimal integration time.

Since a phase measurement requires 4 samples that refer to 4 integration periods, the capturing frame rate can be calculated by

$$FramRate = \frac{1}{4(IntegrationTime + readoutTime)} \quad (3.3)$$

where the readout time is about 4.6ms. Therefore, to achieve a frame rate of 30fps that is the basic requirement of smooth animation, the integration time must be set to a value shorter than 3.73ms.

Various tests were conducted in the laboratory with dim ambient light to find an appropriate integration time to yield an acceptable body motion tracking quality. The distance between the camera and tracked person was about 2 metres, and the background wall was about 3 metres away from the camera. Since the intensity value increases with the integration time, it was multiplied by a coefficient to keep the brightness of the grayscale image. With the API functions of 3×3 Median Filter, Convert Gray (i.e. intensity compensation), and 5×5 Adaptive Neighbourhood Filter (implemented by the camera hardware) enabled, Figure 3.13 shows the two extreme situations with the integration times of 0.3ms (the minimum value that can be set by the software) and 25ms. While the former is seen to result in a noisy output, the latter is seen to result in a saturated output.

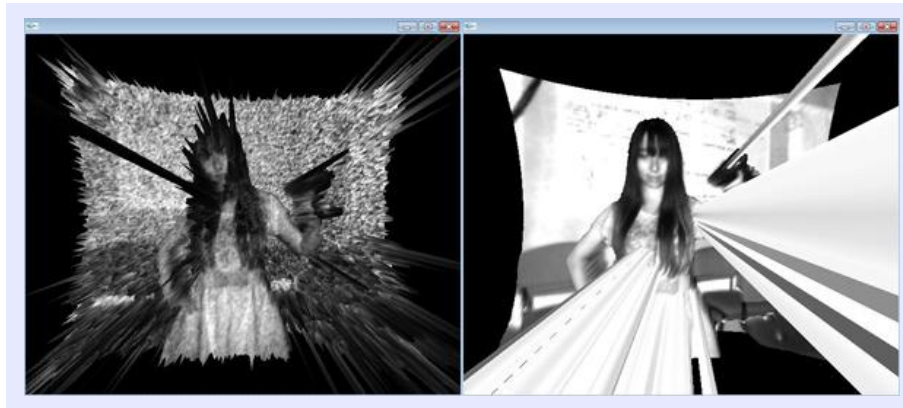


Figure 3.13 Results with 0.3ms (left) and 25ms (right) Integration Times

Through observation, if the integration time was greater than 2ms corresponding to 37.9fps frame rate, an acceptable body motion tracking quality was achieved. Less noise appeared if the integration time was increased to 3ms corresponding to a frame rate of 32.9fps. The tracking results with the integration times of 1ms, 2ms and 3ms are shown in Figure 3.14. It can be observed that most of the noises appeared near the edge of tracking volume and the head area of the tracked person due to the low reflectivity of hairs. Since both the tracking quality and tracking speed were satisfied with a 3ms integration time, and the edge noise caused by body movement was acceptable as well, the integration time was set to 3ms in this project.

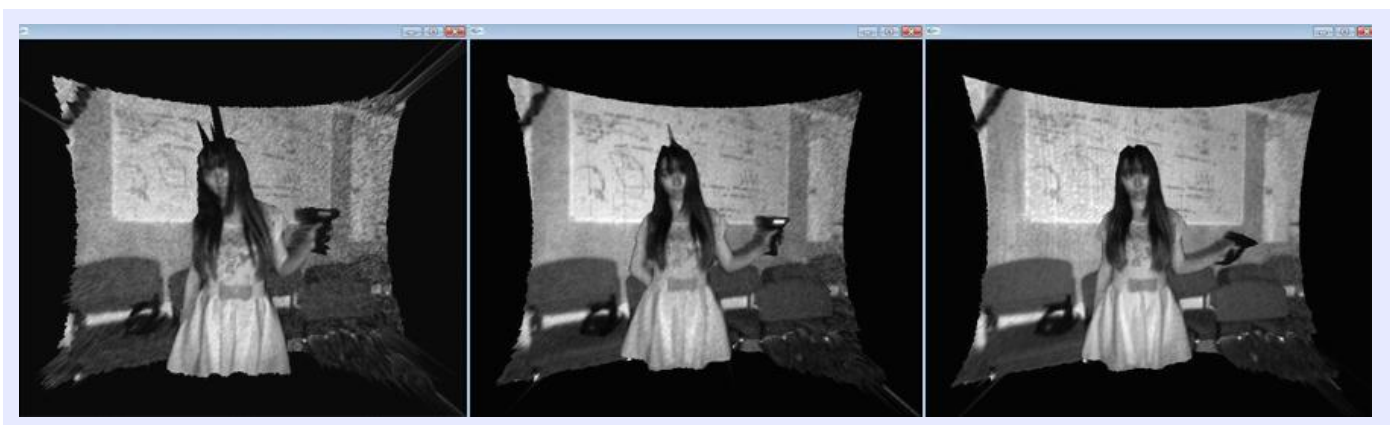


Figure 3.14 Results with 1ms (left), 2ms (middle) and 3ms (right) Integration Times

Various camera positions were tested to provide an appropriate tracking volume of body motion. When the camera was fixed 2 metres in front of the tracked person, it

was capable of covering the person's upper body in the illumination volume. Moreover, the main body of the tracked person was still visible if she/he moved no more than 0.8 metre to the left or right side from the centre of the illumination volume. Although a wider movement range can be observed if the distance between the camera and person is longer, this distance setting is limited by the space available in the laboratory as well as the configurations of the InterSense system and the rear-projection display system to be discussed in detail in the next chapter. Nevertheless, since a standard table tennis table is 1.524m in width, the tracking volume provided by the camera at 2m enables a person to be visible even when she/he moves to the corner of the virtual table.

In evaluation of correct tracking of spatial locations, several anchor points were set both in the laboratory and in the computer generated 3D space. The results showing the degree of overlap between two sets of points were used to confirm the correctness of the spatial information at the output. Furthermore, with the displayed 3D video found to mirror the body movements of the person in the illumination volume, the direction of the local x axis need to be reversed in order for it to be used to display the opponent.

3.4 Stereoscopic Display

3.4.1 Current Technology and Operation Principle

The human visual system has the characteristics of binocular disparity in order to get the depth information of observed objects. In other words, since the left and right eyes of human are separated horizontally, they are seeing different views when the same scene is projected onto the two retinas. By fusing the two views, the visual cortex of

human extracts the depth information, and a 3D impression is perceived by the brain consequently. Therefore, the basic principle of a computer based stereoscopic display is to generate two different images as a stereo pair, and each eye of the viewer is only allowed to see the corresponding image of the stereo pair (Lipton, et al., 2010). Although this principle is not complex, improper stereo pairs can easily result in bad 3D effects, or even fail to generate 3D effects. The two major technical difficulties are the capability of display hardware and the proper creation of stereo pairs.

The basic operation principle for stereoscopic display hardware is to deliver different views to the left and right eyes of observers separately. An early method with very low cost and unnatural effect named anaglyphs employs a pair of complementary colour (commonly red and cyan) to filter the two views for each eye. To produce full-colour images, modern stereoscopic display systems rely on either polarisation or shutter approach. For the polarisation approach, two views are projected and superimposed onto the same screen through different polarising filters. The polarised glasses act as polarising filters to pass only the lights that are polarised similarly, and therefore separate the two views for the left and right eyes of viewers. For the shutter approach (Froehlich, et al., 2005), since a liquid crystal (LC) layer can become dark or transparent by voltage control, the LC shutter glasses is used to alternatively enable one eye to see through the glasses in synchronisation with the refresh rate of the screen that alternatively displays the two corresponding perspectives. A more direct way to separate the two views is to isolate the field of view of each eye physically by special glasses, and render the two perspectives onto the two screens within the glasses individually. Most of HMDs employ this technology.

In this project, the anaglyphs method was rejected due to its unrealistic image colour, and HMD was not employed since the requirement of portability can not be achieved. Since all equipment for the shutter approach has to be able to process frames synchronously at double rate, it is much more expensive than the polarisation approach to get an equivalent display quality, especially for high-speed real-time

applications. Moreover, polarised glasses have advantages of non-flicker and very light weight. There are two types of widely used polarisation technology using either linear or circular polarising filter. Although both of them require viewers to keep their head upright (offset of eye plane will interfere the 3D effect), circular filters allow viewers to tilt their head slightly without confusing the left and right images. Therefore, the employed stereoscopic display system in this project adopts the polarisation technology with circular polarising filters.

3.4.2 Rear-projection Stereoscopic Display System

Two large rear-projection stereoscopic screens were employed to deliver 3D views for two players. The size of each screen is 2.74m in length and 2.06m in height, and the resolution is 1024×768 pixels. The rear-projection technique can avoid occlusion that usually occurs with front-projection, and the big size of the screen is capable to provide table tennis gaming environment in a real physical scale.

The configuration for each screen is shown in Figure 3.15. Two Epson PowerLite8800 projectors are mounted at the back of the screen with a pair of circular polarising filters placed in front of the lens. The filters are used to superimpose two different polarised images onto the same screen via a reflecting mirror. By wearing a pair of light-weight polarised glasses, each player is able to see the views with 3D depth effect. Each screen with two projectors is driven by one computer through its dual DVI graphics card output ports, and the graphic card is NVIDIA Quadro FX with 256MB memory.

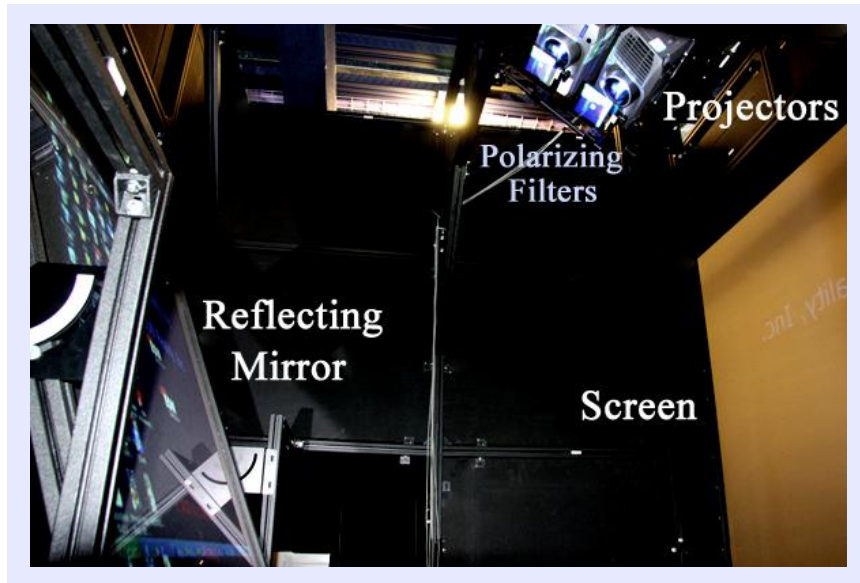


Figure 3.15 Configuration of Rear-projection Stereoscopic Screen

3.4.3 Stereo Pair Rendering

3.4.3.1 Principle and Current Approaches

To produce a depth illusion, the two different images for the left and right eyes need to be projected on a display screen. As illustrated in Figure 3.16, when a point (shown as red in colour) in the scene is projected onto a rendering plane (i.e. the big rear-projection screen in this project), its projected positions are different for the two eyes of a viewer. When an object is rendered by the positive parallax and negative parallax, it is perceived to be located within and outside the screen respectively. And for the zero parallax, the perceived position of the object is at the screen location (Lipton, et al., 2010).

To generate a stereo pair, there are two widely applied approaches, which are known as the “Toe-in” (also called Convergence) and “Off-axis” (named “Lens-shift” or “Asymmetric Frustum Parallel Projection”) methods (Lipton, et al., 2010). Both of

them are based on the geometry approximation, and employ two virtual cameras to get the different views, which are illustrated in Figure 3.17.

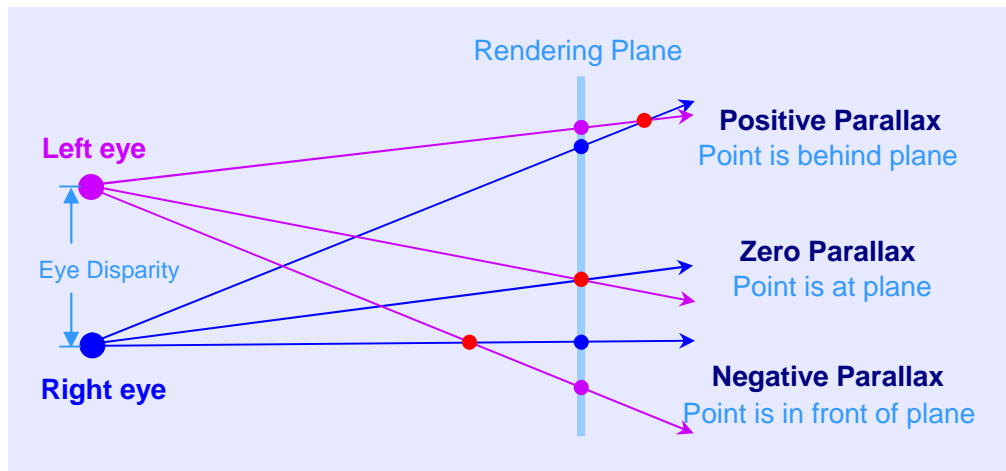


Figure 3.16 Three Types of Projection (Positive/Zero/Negative Parallax)

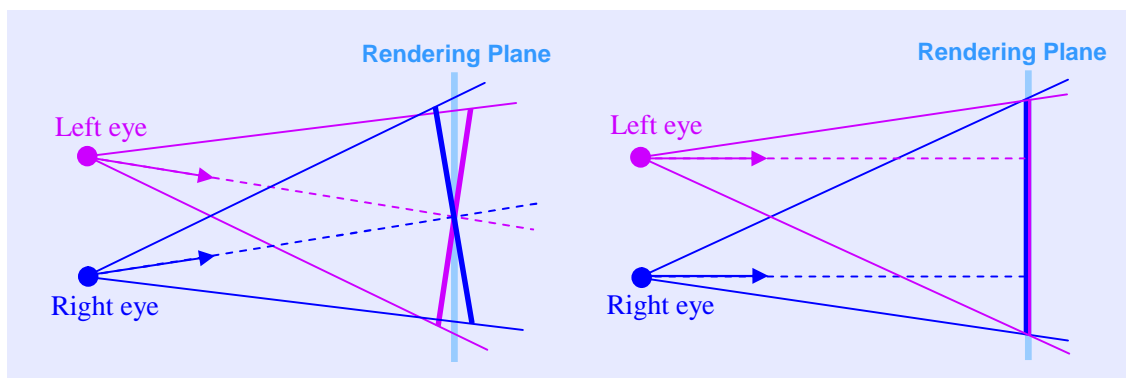


Figure 3.17 Toe-in (left) and Off-axis (right) Projections

For the Toe-in projection, two virtual cameras have symmetric apertures and point at a single focal point. This method can be easily realised by setting two virtual cameras with different orientations, but it suffers from the trapezoidal distortion that increases with the reduction of the focal length due to the rendering plane not parallel to the viewing plane. The Off-axis projection uses the asymmetric frustums and parallel projections of the two cameras, which is immune from the trapezoidal distortion. Although the Off-axis method matches the rendering geometry to the viewing geometry much better than the Toe-in projection, the parameter setting for the Off-axis method is very important and incorrect parameter values will lead to very

serious distortion. Since OpenGL supports the asymmetric frustum projection, the Off-axis method was adopted in this project.

3.4.3.2 Stereo Pair Rendering by OpenGL

OpenGL uses a frustum of pyramid to assign which parts of the scene need to be rendered based on perspective projection (Shreiner, et al., 2009; Hearn and Baker, 2004; Wright and Lipchak, 2004). Perspective projection has a foreshortening effect that means all objects on the rendering plane are “the farther the smaller”. It is similar to how real world is projected onto humans’ retinas. Usually, the frustum is symmetrical, and the default origin of the OpenGL coordinate system is at the apex of the pyramid as shown in Figure 3.18. To set up a frustum, the parameters, such as *ViewAngle*, *Aspect*, d_{near} , and d_{far} , need to be assigned correctly based on the requirements of the display system and application.

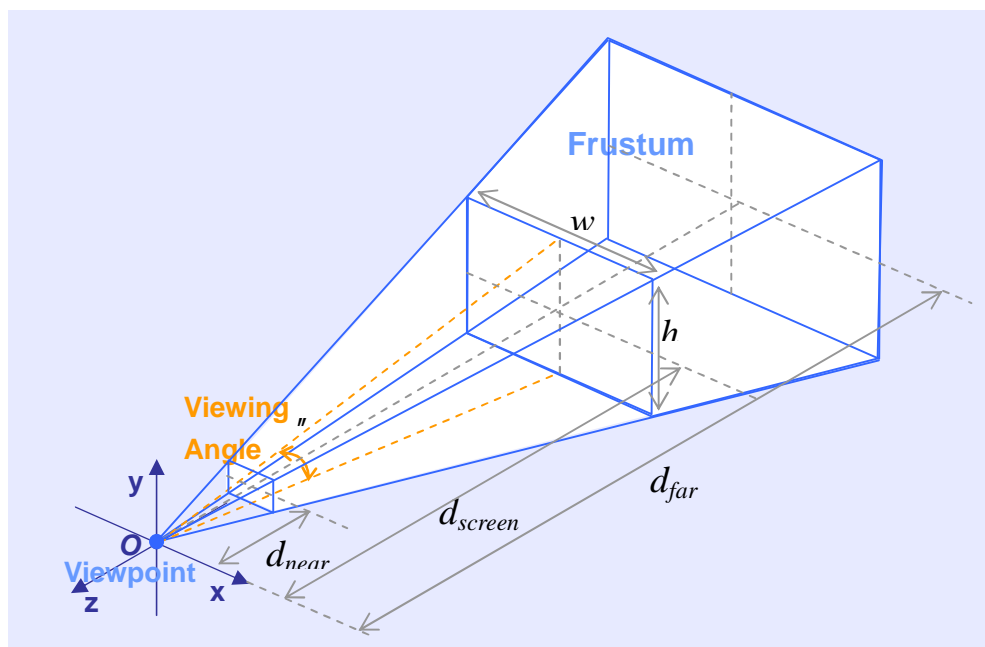


Figure 3.18 OpenGL Frustum

For common OpenGL rendering without the stereoscopic effect, a front buffer and a back buffer are swapped to allow the display data to be updated in order to provide a

smooth display. To generate a stereoscopic display that requires rendering of two independent images for the two eyes in each frame, OpenGL supports the quad-buffered rendering technology by providing left/right and front/back buffers for image storage.

To realise the Off-axis projection, two asymmetrical frustums need to be set for the two eyes' view individually. Figure 3.19 illustrates the frustum setting for the right eye, where the origin of the symmetrical frustum is first translated from the tracked viewpoint (i.e. middle of two eyes) to the right side by half of the intraocular distance, $IOD/2$, and then the frustum is shifted to match the rendering plane (i.e. screen). The resultant asymmetrical frustum for the right eye is illustrated in the top view figure by a pale blue area. An important parameter d_{shift} that describes the degree of asymmetry is given by

$$d_{shift} = \left(\frac{IOD}{2} \right) \left(\frac{d_{near}}{d_{screen}} \right) \quad (3.4)$$

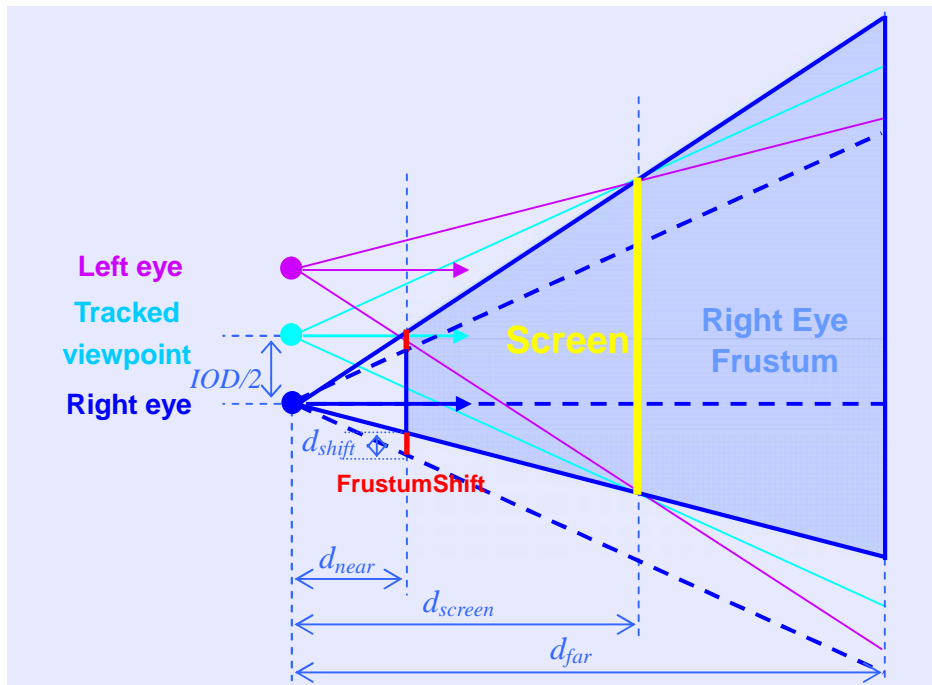


Figure 3.19 Frustum Setting for Right Eye

The asymmetrical frustum for left eye can be generated in the same way, and the origin of the OpenGL coordinate system is still at the tracked viewpoint.

3.4.4 Display Performance

In evaluation of the 3D display, three virtual white balls with a diameter of 0.04m (the diameter of a standard table tennis ball) were created in a virtual space with black background generated by OpenGL. The distances between the balls and the origin of the OpenGL coordinate system along z axis were set to 0.5m, 1.5m and 2.5m, respectively. With a parameter setting of $d_{screen}=1.5m$, the user's viewpoint should be located 1.5m away from the middle of screen. By setting the parameters of $IOD=0.06m$ and $d_{near}=0.2m$, which results in $d_{shift}=0.004m$, two asymmetrical frustums were generated. With using quad-buffered rendering, three balls were displayed on a rear-projection stereoscopic screen.

Figure 3.20 shows three photos when the three balls were displayed individually. In order to demonstrate “real scale” effect of the virtual ball, a real table tennis ball was also in the picture (on the left side of the virtual ball). The camera that used to take these photos was located at the position of the viewpoint (1.5m away from the middle of screen).

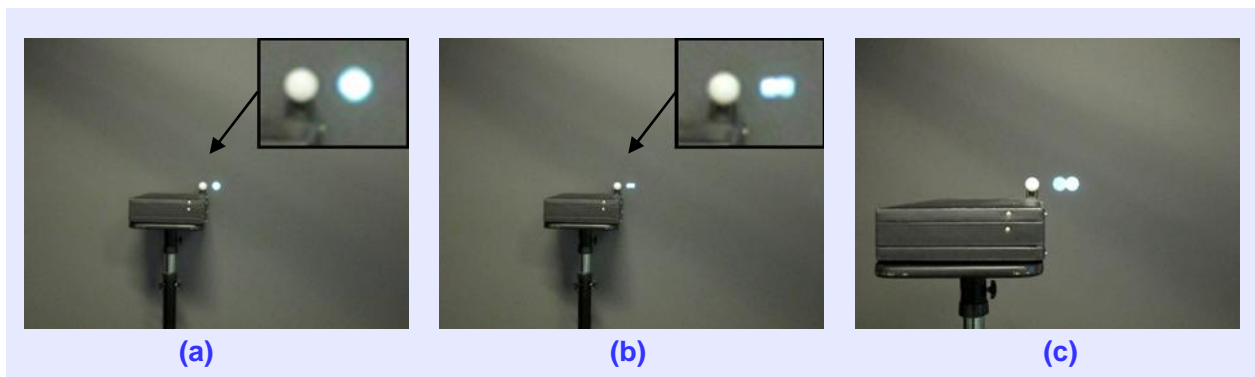


Figure 3.20 Photos of virtual balls and real table tennis balls

In Figure 3.20(a), the distance between the virtual ball and the viewpoint was 1.5m, which means the ball was at the screen's location. Therefore, the ball was displayed clearly due to zero parallax (explained in Figure 3.16). In this photo, a real table tennis was close to the screen. It can be observed that the virtual ball and the real ball

have very similar size. If the virtual ball is located 2.5m away from the viewpoint, and the real ball is still close to the screen, the result is shown in Figure 3.20(b). In this case, the size of the virtual ball was smaller than the real ball due to perspective projection. Because of positive parallax (explained in Figure 3.16), the projected positions of the ball for left and right eyes were different. Therefore, the virtual ball on the screen was blurred without wearing 3D glasses. In Figure 3.20(c), both the virtual and the real ball were 0.5m away from the viewpoint, they looked have a same size. Due to negative parallax (explained in Figure 3.16), the virtual ball was blurred as well. By wearing 3D glasses, the virtual ball was displayed clearly and looked “in front of the screen”.

By wearing 3D glasses, an illusion of 3D was perceived by viewer, where the balls in all three cases were seen clearly with the same size at different depths (this effect cannot be shown by 2D pictures in this thesis). The positions of the three balls were perceived as “on the screen” (case of Figure 3.20(a)), “behind the screen” (case of Figure 3.20(b)) and “in front of the screen” (case of Figure 3.20(c)) respectively, which correspond correctly to zero parallax, positive parallax and negative parallax. Furthermore, the big size of the screen with its good display quality and enough brightness were seen to increase the viewer’s perception of immersion, thereby justifying the use of it for this project.

3.5 Conclusion

This chapter focuses on two crucial parts of an interactive and immersive VR environment, which are motion tracking and stereoscopic display. As introduced in Chapter 2, motion tracking is fundamental to achieve intuitive interaction, whereas stereoscopic display creates an illusion of 3D depth leading to immersive perception.

Since there are various commercial products for motion tracking and stereoscopic display based on different kinds of operation principles, a review of current approaches and the justification of the systems selected for this project are given in this chapter. To summarise, the InterSense IS-900 Motion Tracking System is capable of real-time capture of accurate 6-DOF motion data of multiple trackers with a reasonably good quality. Hence it is used for tracking the pose of the heads and racquets of the players. The SR4000 camera is capable of real-time capture of 3D depth information with satisfactory tracking quality. The 3D information can be integrated to the game environment, thereby enabling stereoscopic viewing from different viewing positions.

Chapter 4

Virtual Reality Environment for Single-player Table Tennis

4.1 Introduction

This chapter presents implementation of a single-player table tennis game, with an InterSense motion tracking system and a rear-projection stereoscopic screen employed as system input and output, as well as a host computer running an application program that is responsible of motion data processing, game environment and animation generation, as well as stereoscopic rendering.

Since the hardware setup has been introduced in Chapter 3, this chapter focuses on the processing software which mainly includes virtual objects generation, coordinate systems unification, as well as applying motion data to setup viewpoint and to represent racquet's movements. In order to demonstrate the differences between a conventional game and a VR simulation in terms of viewpoint change, the displaying results with and without real-time frustum setting are compared and discussed. In addition, a physics-based ball animation model including real-time collision detection and response was developed, which is more realistic than the models used by most of real-time table tennis simulation/exergame.

4.2 Hardware and Software Implementation

The block diagram of hardware configuration for a single-player game is illustrated in Figure 4.1. The communication between a VETracker processor and a host computer is through a serial link port, and two projectors used for stereoscopic display are driven by the host computer as well through its graphic-card output ports. The operation principles of both the InterSense motion tracking system and the rear-projection stereoscopic display system have been introduced in Chapter 3.

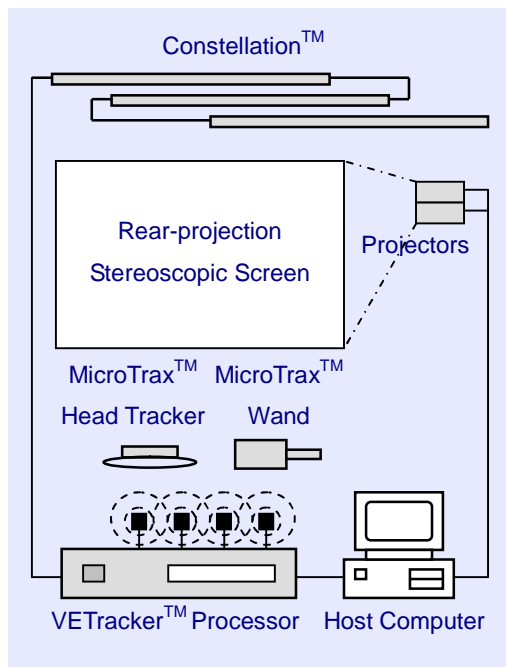


Figure 4.1 Hardware Configuration for Single-player Game

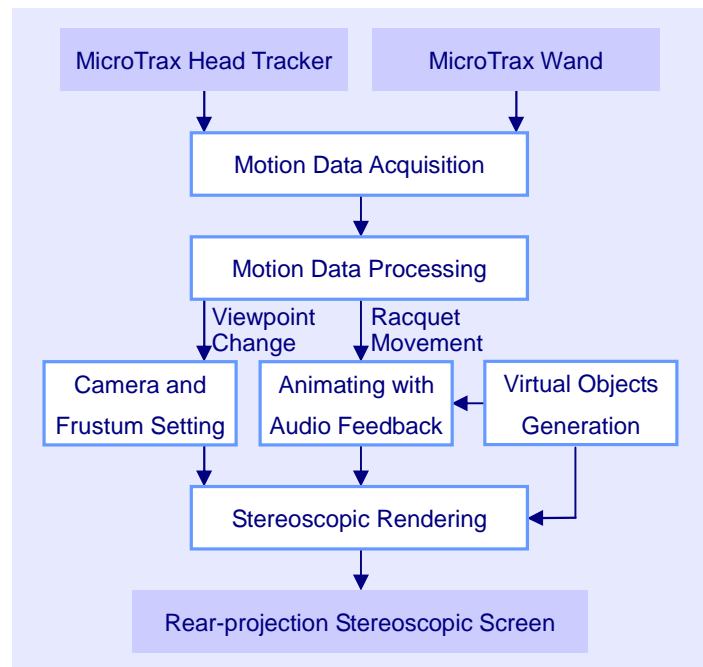


Figure 4.2 Software Implementation for Single-player Game

The software implemented using the C++ programming language is based on the software modules illustrated in Figure 4.2. First of all, the Motion Data Acquisition module for acquiring 6-DOF data of both the MicroTrax Wand and MicroTrax Head Tracker is implemented based on the InterSense API. The spatial coordinates of player's head and hand are transformed by the Motion Data Processing module to a common coordinate system. In order to provide a correct field of view, the position of the virtual camera and the frustum's angle and shift values are adjusted in real-time

according to the head-tracking result. Since the racquet's position and orientation are tracked, the collisions between the racquet and ball can be detected. By using a physics-based ball animation model, the ball's position in each frame is calculated. In conjunction with the Virtual Objects Generation module, animations of the ball and racquet, as well as a sound of collision are generated in the Animating with Audio Feedback module. All the static objects (i.e. table and room) are created by the Virtual Objects Generation module as well, and saved in a display-list. To achieve immersive effect, a stereo pair is generated in each frame by the Stereoscopic Rendering module based on the "Off-axis" method.

4.3 Game Environment Generation

4.3.1 Virtual Objects Generation

The virtual space with virtual objects was generated by OpenGL, which is a software interface to graphics hardware. Although it does not provide high-level commands for describing 3D models, it has an advantage of high-speed polygon-rendering and is available for various hardware platforms (Shreiner, et al., 2009; Hearn and Baker, 2004; Wright and Lipchak, 2004). The virtual table tennis ball, racquets, table and room were constructed by using basic geometric primitives (i.e. points, lines and polygons) as shown in Figure 4.3. Their shapes and sizes were designed according to the specifications of a standard table tennis game.

Ball

The virtual ball is an orange sphere model with a diameter of 0.04m (ITTF, 2009).

Racquet

Since there is no standard size or shape of a racquet, a round shape virtual racquet (two for two-player game) with 0.15m diameter and 0.01m thickness was created. The red and black surfaces represented the two rubber-layers on each racket, and the wood handle is pale yellow in colour that has a length of 0.075m.

Table

The virtual table consists of a rectangular board, a net and legs. The board has a height of 0.76m, and a size of 1.525m in width and 2.74m in length (ITTF, 2004). It is blue in colour with white boundary lines. The net is represented by crossed grey lines, and the height of it is set to 0.15m.

Room

The virtual room is a rectangular cuboid with 4m in width, 6m in length, and 3m in height. The wood tiles on the floor and ceiling, as well as bricks and pictures on the wall are texture mapped.



Figure 4.3 OpenGL-generated Virtual Objects

4.3.2 Coordinate Systems

In order to represent a tracked virtual racquet with correct spatial information and to project the 3D scene onto a display screen in a proper way, the local coordinate systems used for the InterSense tracking data and the OpenGL-generated virtual space should be unified in the world coordinate system. To achieve a better immersive effect, the virtual world should merge with the real world to a certain extent (e.g. through the stereoscopic display, the virtual floor can be observed on the same plane of the real floor). Therefore, the relationship between the real world and the virtual world should be established.

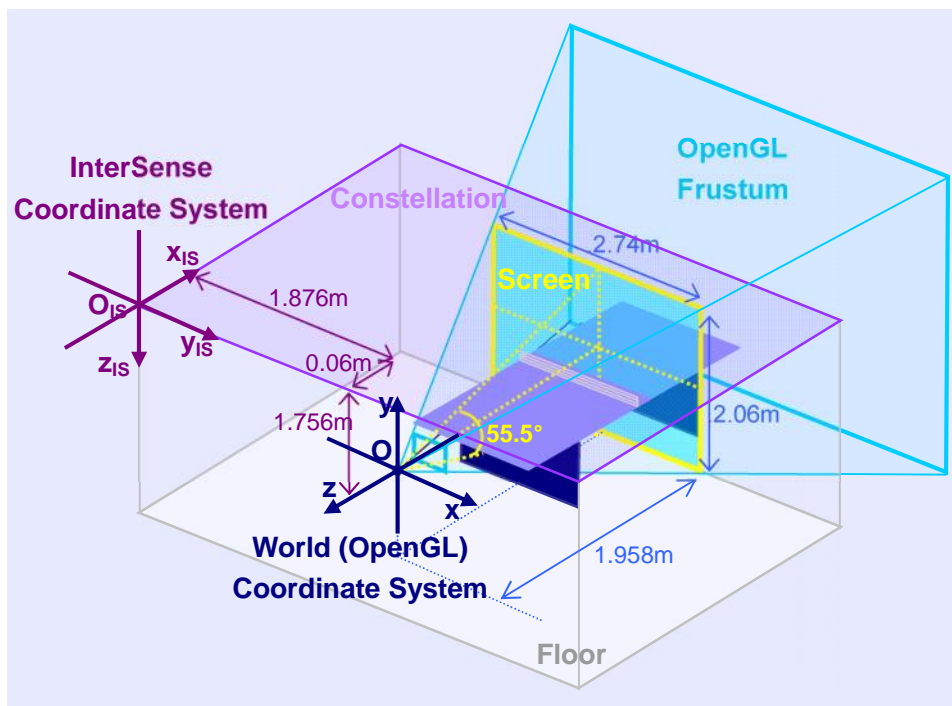


Figure 4.4 Coordinate Systems

World Coordinate System

Figure 4.4 illustrates the world coordinate system defined in the laboratory, which is a right-handed Cartesian coordinate system. From a player's view (i.e. facing the screen), the x -axis points to the right side horizontally and the y -axis points vertically upward (with the x - y plane paralleled to the screen plane). The z -axis points away

from the centre of the screen plane (i.e. pointing towards the player). The origin of the world coordinate system is at 1.958m away from the screen along the z-axis, giving a 70° (horizontal) \times 55.5° (vertical) field of view.

OpenGL Coordinate System

The coordinate system used by OpenGL has the same origin and orientation as the world coordinate system. If the virtual camera is located at the origin of the virtual scene, the OpenGL frustum is as that shown in Figure 4.4.

InterSense Coordinate System

The origin of the InterSense coordinate system is located at the corner of the ceiling-mounted Constellation. As shown in Figure 4.4, the distances between its origin and the origin of the world coordinate system along the x, y and z axes are 0.06 m, 1.876 m and 1.756 m, respectively. For the orientation, the x-axis points towards the screen plane, the y-axis has the same direction as the x-axis of the world coordinate system, and the z-axis points in the direction of gravity.

Coordinate Systems of Virtual Objects

Each virtual object has its own local coordinate system, and is implemented by transforming the position and orientation of its local coordinate system. For instance, the origin of the table's local coordinate system is at the centre of the table's upper surface, by setting this origin at a distance of 0.76m (i.e. table's height) away from ground, the table lies on the floor.

Figure 4.5 shows the result when the virtual camera was at the origin of the world coordinate system, with the z-axis pointed towards it.

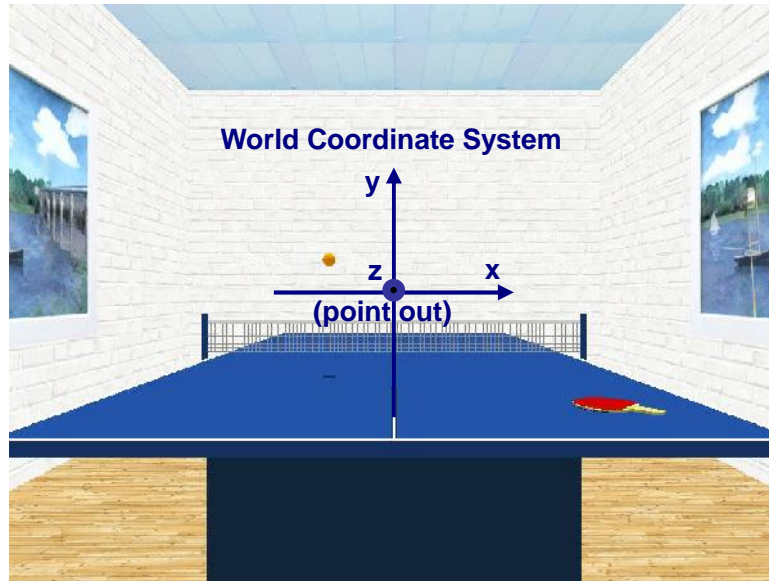


Figure 4.5 Views of Game Scene

4.4 Applying Motion Tracking Data

4.4.1 Racquet Animation based on Hand Tracking

To draw a racquet constructed by polygons, the positions of all the vertices on the virtual racquet need to be specified. Since the coordinates of these vertices refer to the racquet's local coordinate system, their corresponding coordinates in the world coordinate system can be calculated by using the method of geometric coordinate transformation (Watt and Policarpo, 2000).

On the other hand, in order to animate this racquet according to the movement of player's hand, the racquet's local coordinate system is translated and rotated in real-time based on the output data of motion tracking. The translation parameters indicate the origin of the racquet, while the rotation parameters are employed to calculate a local racquet's rotation.

Let $\vec{v} = [x_R, y_R, z_R, 1]^T$ represent a 3D vertex on the virtual racquet in the racquet's local coordinate system. With the racquet's movement, the corresponding homogeneous coordinate of \vec{v} in the world coordinate system, labelled by $\vec{v}' = [x, y, z, 1]^T$, can be calculated by Equation 4.1.

$$\vec{v}' = T^{IS \rightarrow W} K T^{Rac \rightarrow IS} \vec{v} \quad (4.1)$$

where $T^{Rac \rightarrow IS}$ is a rotation matrix used to align the directions of the racquet's local coordinate axes to the InterSense coordinate axes, which is given by

$$T^{Rac \rightarrow IS} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

K is a composite transformation matrix that firstly rotate the racquet locally, followed by translating the racquet according to the tracking data acquired by the InterSense system

$$K = \begin{bmatrix} C_p C_y & S_r S_p C_y - C_r S_y & C_r S_p C_y + S_r S_y & T_x \\ C_p S_y & S_r S_p S_y + C_r C_y & C_r S_p S_y - S_r C_y & T_y \\ -S_p & C_p S_r & C_p C_r & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

where C and S indicate cosine and sine, respectively. Subscripts y , p and r denote the rotation angles along yaw, pitch and roll directions. T_x , T_y and T_z are the translations along the x, y, and z axes, respectively.

To implement the transformation from the InterSense coordinate system to the world coordinate system, $T^{IS \rightarrow W}$ is given by

$$T^{IS \rightarrow W} = \begin{bmatrix} 0 & 1 & 0 & -1.876 \\ 0 & 0 & -1 & 1.756 \\ -1 & 0 & 0 & 0.06 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

By using a composite transformation matrix, combining (4.2) to (4.4), (4.1) becomes

$$\vec{v}' = \begin{bmatrix} S_r S_p S_y + C_r C_y & S_r C_y - C_r S_p S_y & -C_p S_y & T_y - 1.876 \\ -C_p S_r & C_p C_r & -S_p & -T_z + 1.756 \\ C_r S_y - S_r S_p S_y & C_r S_p S_y + S_r S_y & C_p C_y & -T_x - 0.06 \\ 0 & 0 & 0 & 1 \end{bmatrix} \vec{v} \quad (4.5)$$

4.4.2 Virtual Camera Setting based on Head Tracking

To enable a viewer to observe virtual objects from different points of view, the virtual camera in the 3D space should be moved and rotated in real-time according to the viewpoint and gaze direction of the tracked person. A universal approach is to set the virtual camera as an animated object (Watt and Policarpo, 2000), which is widely employed in first-person computer games, and HMD-based VR applications.

In this project, viewpoint is defined at the middle position of a player's two eyes, which can be computed from the position of an InterSense head tracker. In the initial design, a player's gaze direction was approximated by her/his head orientation obtained from head-tracking as well. However, the scene orientation change was not included in the final system, and the reason is explained in the next section.

Since the captured position of the head tracker, \vec{p}_{head} , is in the InterSense coordinate system, the viewpoint in the world coordinate system, \vec{p}_{eye} , needs to be calculated by a transformation matrix shown in Equation 4.6. The distance between the head tracker and eyes' position along the vertical axis labelled by d is set to 0.15m.

$$\vec{p}_{eye} = \begin{bmatrix} 0 & 1 & 0 & -1.876 \\ 0 & 0 & -1 & 1.756 - d \\ -1 & 0 & 0 & 0.06 \\ 0 & 0 & 0 & 1 \end{bmatrix} \vec{p}_{head} \quad (4.6)$$

Figure 4.6 shows the results when the head tracker was at different positions with a forward viewing direction without camera roll. Figure 4.6(a) and (b) are the views with low (0.75m in height) and high (1.8m in height) eyes' positions. When the camera was at the right and left side (half width of the table), the results are shown in Figure 4.6(c) and (d). Figure 4.6(e) and (f) refer to the views if the viewer stepped 1m forward and backward, respectively.

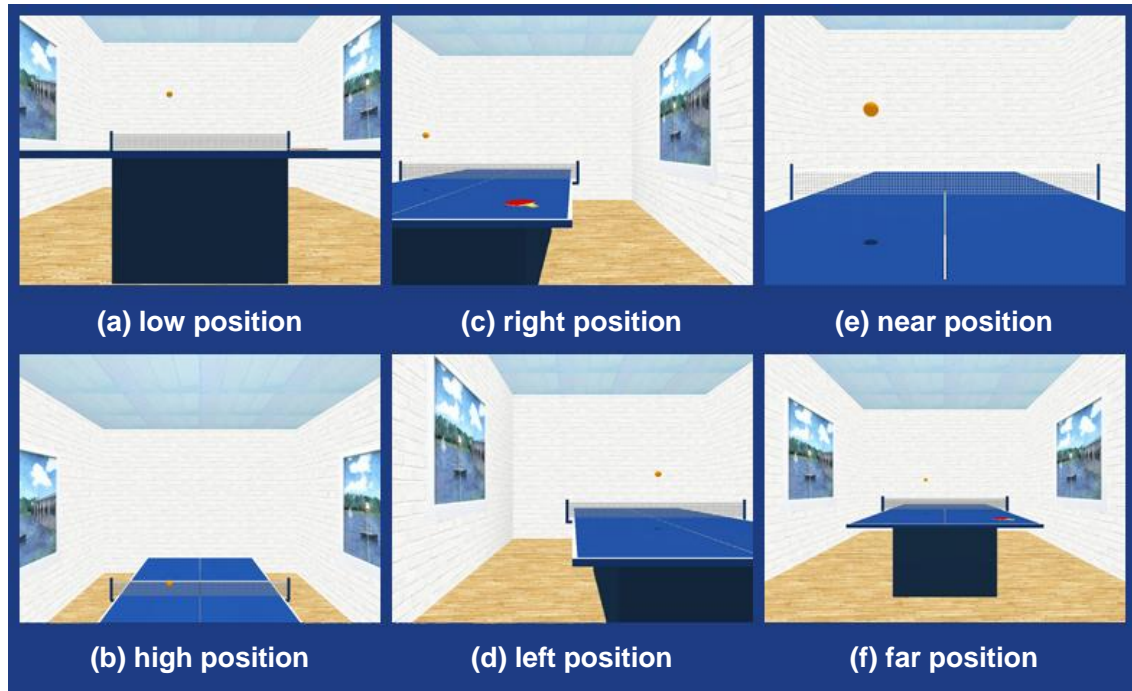


Figure 4.6 Views at Different Camera Locations

As OpenGL-default viewing directions, \vec{n}_{view} (points forward) and camera normal \vec{n}_{cam} (points upward) refer to homogeneous coordinates $\vec{n}_{view} = [0,0,-1,1]^T$ and $\vec{n}_{cam} = [0,1,0,1]^T$ in the camera's local coordinate system. Their corresponding coordinates in the world coordinate system after camera orientation, denoted by \vec{n}'_{view} and \vec{n}'_{cam} , are expressed by

$$\vec{n}'_{view} = T\vec{n}_{view} \quad (4.7)$$

$$\vec{n}'_{cam} = T\vec{n}_{cam} \quad (4.8)$$

$$T = \begin{bmatrix} S_r S_p S_y + C_r C_y & S_r C_y - C_r S_p S_y & -C_p S_y & 0 \\ -C_p S_r & C_p C_r & -S_p & 0 \\ C_r S_y - S_r S_p S_y & C_r S_p S_y + S_r S_y & C_p C_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.9)$$

where C and S indicate cosine and sine calculation. Subscripts y , p and r denote the rotation data along yaw, pitch and roll directions, respectively. This transformation matrix is same as the orientation part of the composite matrix used for hand-tracking (i.e. Equation 4.5). The computed \vec{n}'_{view} decides the pitch and yaw of the virtual camera, while \vec{n}'_{cam} refers to the camera roll.

If the virtual camera was fixed at the origin of the world coordinate system, the displayed views with different camera orientations are shown in Figure 4.7. In Figure 4.7(a) and (b), the viewer looked up and down (i.e. camera pitch) 30 degrees from the horizontal z-axis. Figure 4.7(c) and (d) show the results if the viewer swivel her/his head (i.e. camera yaw) 30 degrees to the right and left side. If the viewer skews head 30 degrees to her/his right and left shoulders (i.e. camera roll), the views are shown in Figure 4.7(e) and (f).

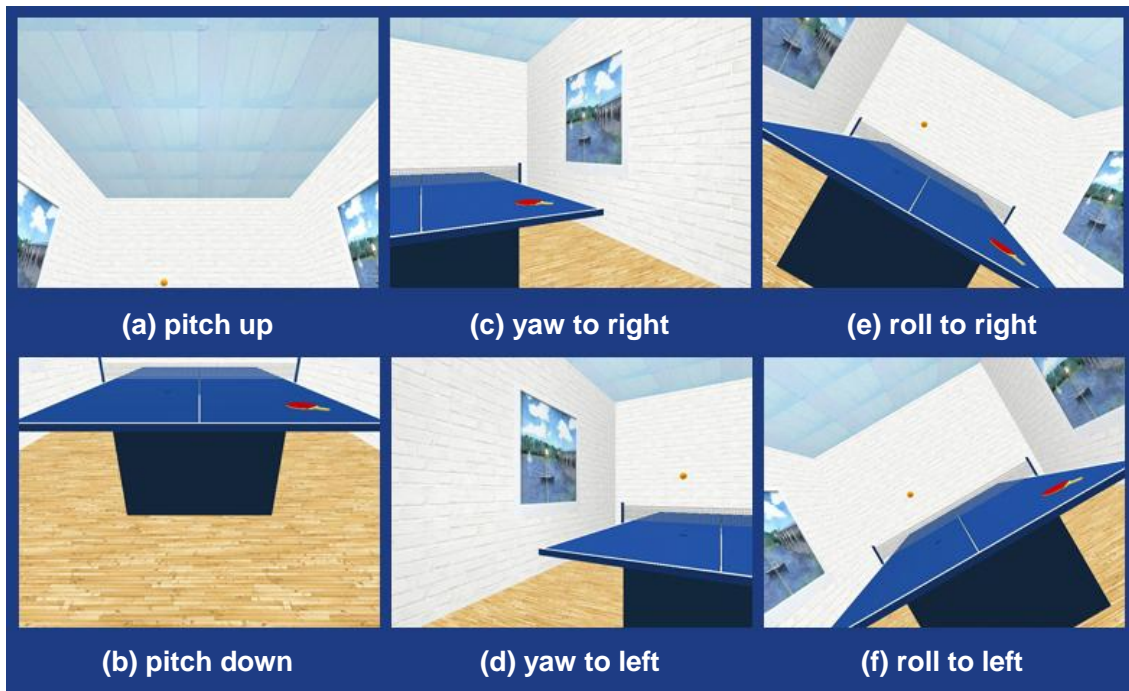


Figure 4.7 Views with Different Camera Rotations

4.4.3 Frustum Setting based on Head Tracking

With the change of viewpoint and viewing direction based on head-tracking, virtual objects can be observed from different directions. The results shown in Figures 4.6 and 4.7 in the previous section are similar as the views of a common first-person game. However, when a player was walking around in front of the screen, she/he felt that the virtual room with the table was not stationary. The whole scene moved and tilted with the translation and orientation of the head tracker. The further the player went away from the origin of the world coordinate system, the more significant the distortion was observed. This distortion is due to the relative movement between the player's eyes and the fixed screen, which can be corrected by real-time adjustment of the OpenGL frustum according to head-tracking.

The viewing frustum is a volume that assigns which part of the 3D scene is potentially visible on the screen, which has been introduced in Section 3.4.3.2. The significance of real-time setting of the frustum depends on the degree of relative movement between viewer's eyes and screen, as well as the required degree of immersion. In other words, if there is no relative movement, such as HMD moving with the head of viewer, the projected images are always correct. Another example is VR cockpit simulation with a big screen display, the projected landscape is correct since the pilot does not walk around. For the applications not requiring realistic immersion effect, such as rendering a conventional video game, the viewing frustum is fixed.

Since the table tennis sport requires accurate spatial information to be perceived by players and the VR system developed in this project is proposed to be highly immersive, the viewing frustum need to be adjusted in real-time according to head-tracking.

4.4.3.1 Distortion Analysis

The distortion caused by a fixed frustum is demonstrated by an extreme situation that the player's viewpoint is on the ground as illustrated in Figure 4.8. Since the origin of the frustum is defaulted to the origin of the world coordinate system, the projected virtual table seems hanging in the air from the player's view.

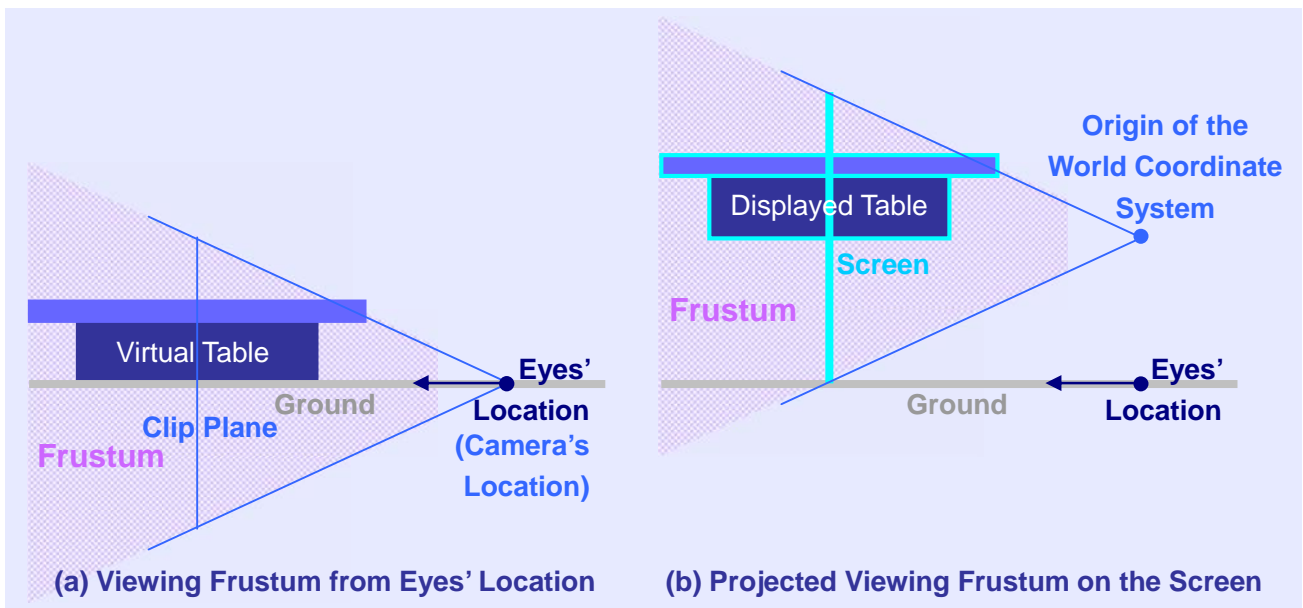


Figure 4.8 Incorrect Frustum Setting with Distortion (Side View)

In order to display the table in a correct location, the frustum's origin should be shifted to the eyes' location, and the shape of the frustum needs to be changed to match the screen plane. Figure 4.9 illustrates the correct frustum setting for this case.

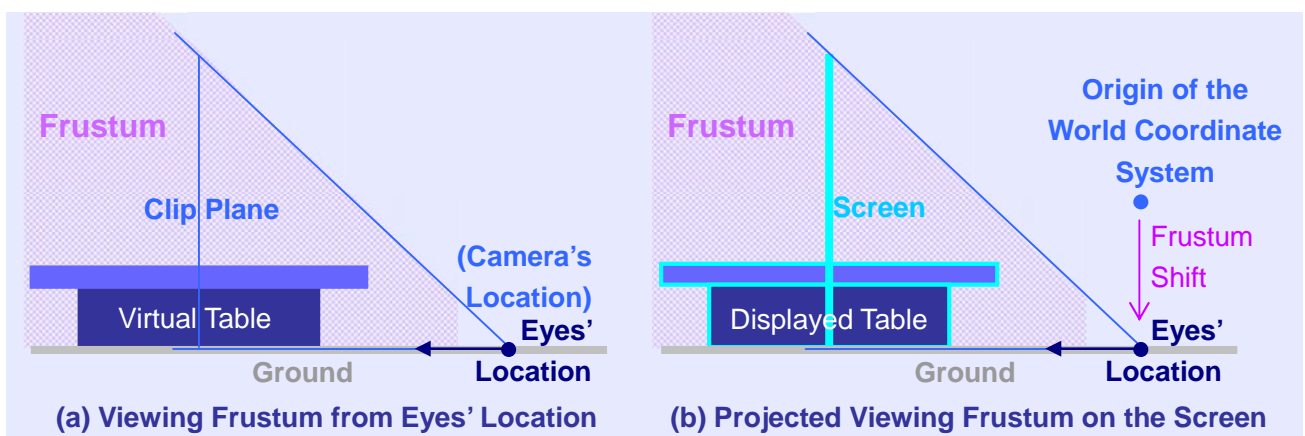


Figure 4.9 Correct Frustum Setting without Distortion (Side View)

The results with incorrect and correct projecting methods of the above example are shown in Figure 4.10(a) and (b), respectively. Obviously, the view in Figure 4.10(b) is same as what the player should see.

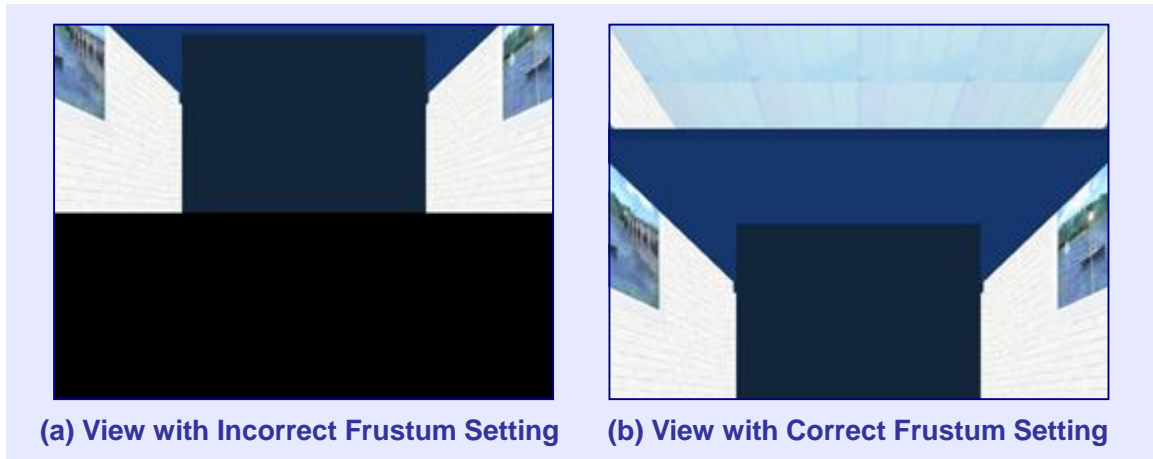


Figure 4.10 Views with incorrect (left) and correct (right) Frustum Setting

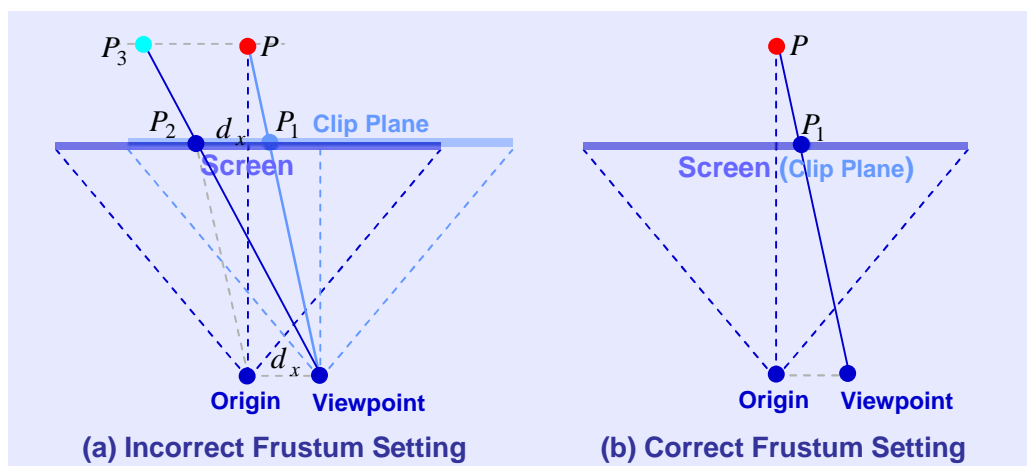


Figure 4.11 Distortion Analysis for Viewpoint Displacement along x-axis

In the above example, the viewpoint moves along the y-axis. If the virtual camera moves along the x-axis, the distortion again occurs due to the same reason. Figure 4.11 (top view of the screen) illustrates the situation of moving the viewpoint to the right side along the x-axis by a distance of d_x . The incorrect frustum setting is shown in Figure 4.11 (a), where a red point at position P is projected onto the camera's clip plane at P_1 . When the view is displayed on the screen, the projected red point is at P_2 . As a result, the red point is at P_3 from the player's view. Therefore, if the frustum is symmetrical and fixed, with the movement of the viewpoint along x-axis or y-axis,

the projected position on the screen of a spatial point moves to the opposite direction by the same distance. With the stereoscopic display, the 3D position of the point perceived by the viewer moves even further. In Figure 4.11(b), the clip plane of an asymmetric frustum is mapped onto the screen, and the red point is projected at a correct position.

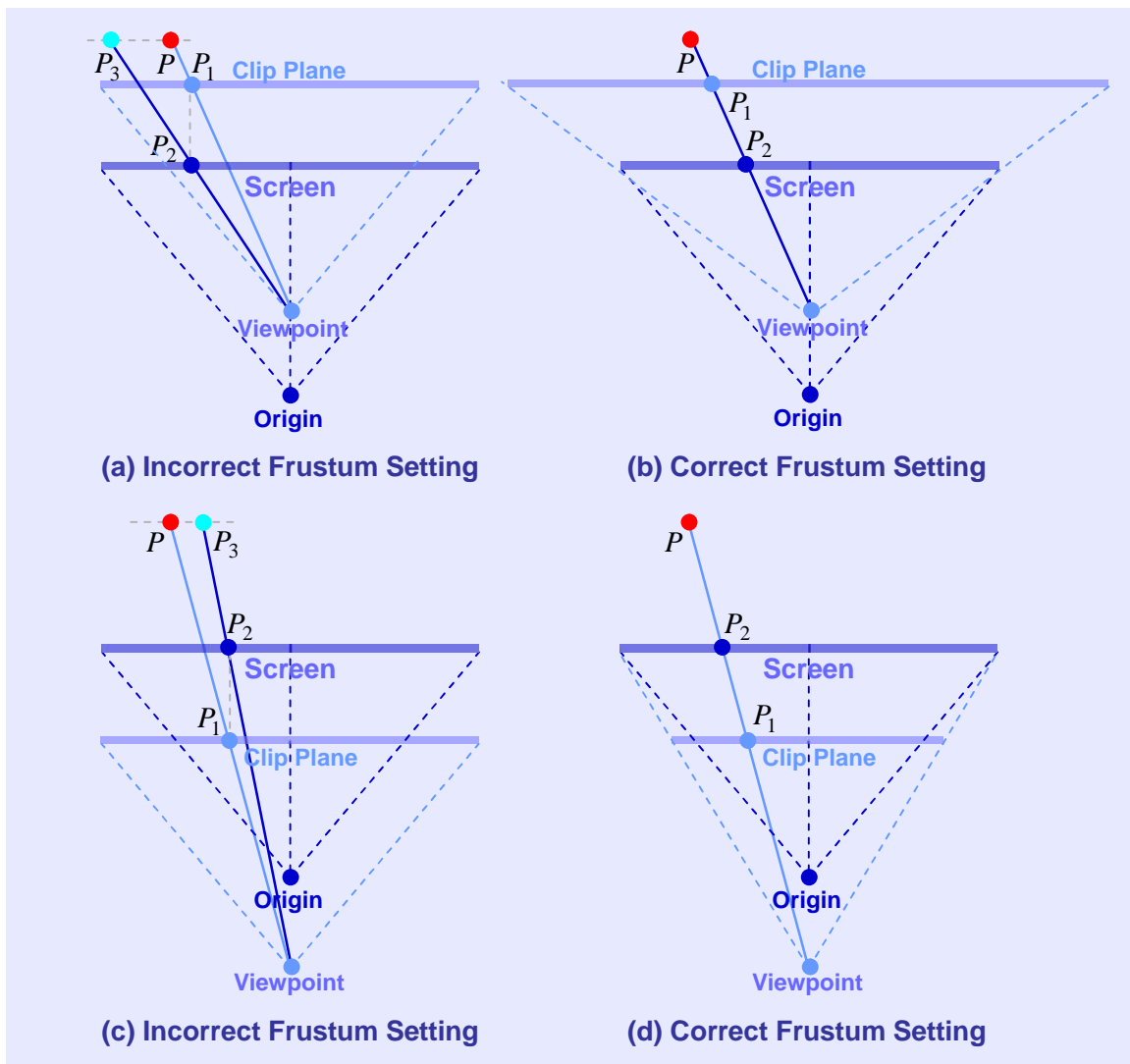


Figure 4.12 Distortion Analysis for Camera Displacement Along z-axis

The distortion caused by the viewpoint movement along the z-axis is analysed in Figure 4.12 (top view of the screen). If a viewer walks forward or backward from the origin of the world coordinate system, with a fixed viewing angle, the spatial positions of the red point, P , perceived by the viewer is at the position P_3 as shown in Figure 4.12(a) and (c). Therefore, if the viewing angle of the frustum is fixed, all the

3D objects tend to be bigger and smaller if the viewer walks forward and backward, respectively. With the change of viewing angle as illustrated in Figure 4.12(b) and (d), the red point can be observed at a correct location, and the scale of the scene is therefore correct.

The orientation distortion is illustrated in Figure 4.13, with the principle similar to that of translation distortions, and leads to an even worse result due to the tilt effect.

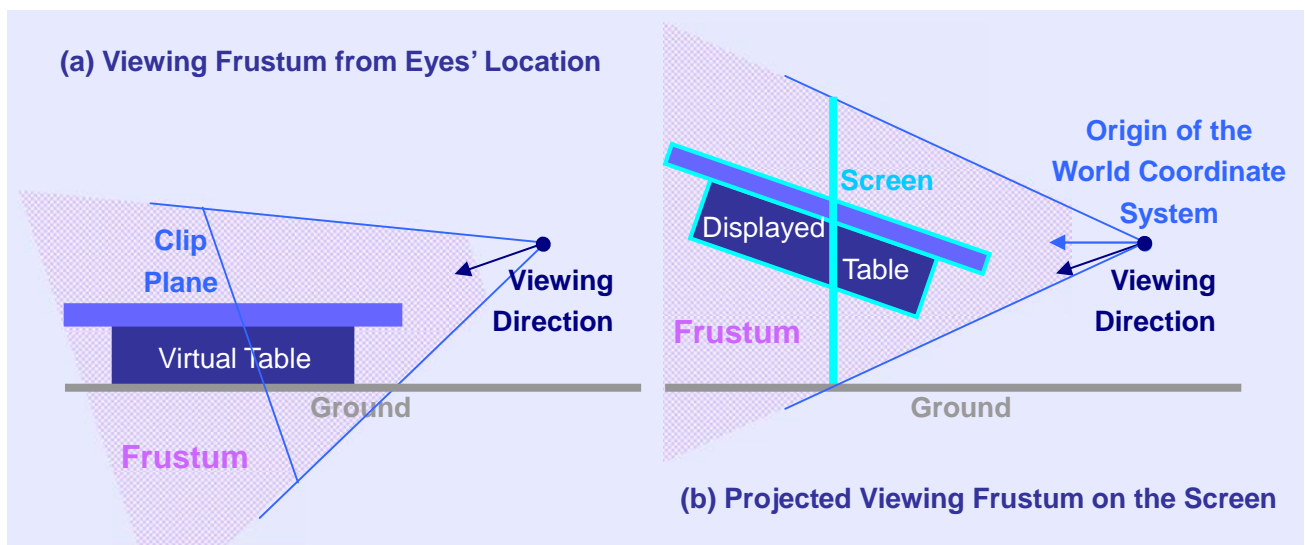


Figure 4.13 Distortion Caused by the Scene Orientation (Side View)

In fact, when the player gazes on different points on the screen, the orientation of the viewing direction has already been achieved. Since the screen is big enough, player has a wide angle of view without the need of rotating the 3D scene. Therefore, the orientation data of head-tracking was not employed in this project. The viewing direction was defined as pointing to the screen along the z-axis.

4.4.3.2 Real-time Frustum Adjustment

In order to set an asymmetrical frustum with a correct position for the origin, a method similar to the Off-axis projection setting for stereoscopic display introduced in Chapter 3 is used. As illustrated in Figure 4.14, the origin of the symmetrical

frustum is translated onto the viewpoint's position, and then the frustum is shifted to match the screen plane. Let $\vec{p}_{eye} = [x_{eye}, y_{eye}, z_{eye}, 1]$ be the coordinates of the tracked viewpoint in the world coordinate system. A parameter of d_{xshift} describes the amount of shift along the x-axis and is expressed by

$$d_{xshift} = -x_{eye} \left(\frac{d_{near}}{d_{screen}} \right) \quad (4.10)$$

where the negative sign means that the frustum shifts to the opposite direction. For instance, if x_{eye} is positive, the frustum shifts to the negative x direction, and vice versa.

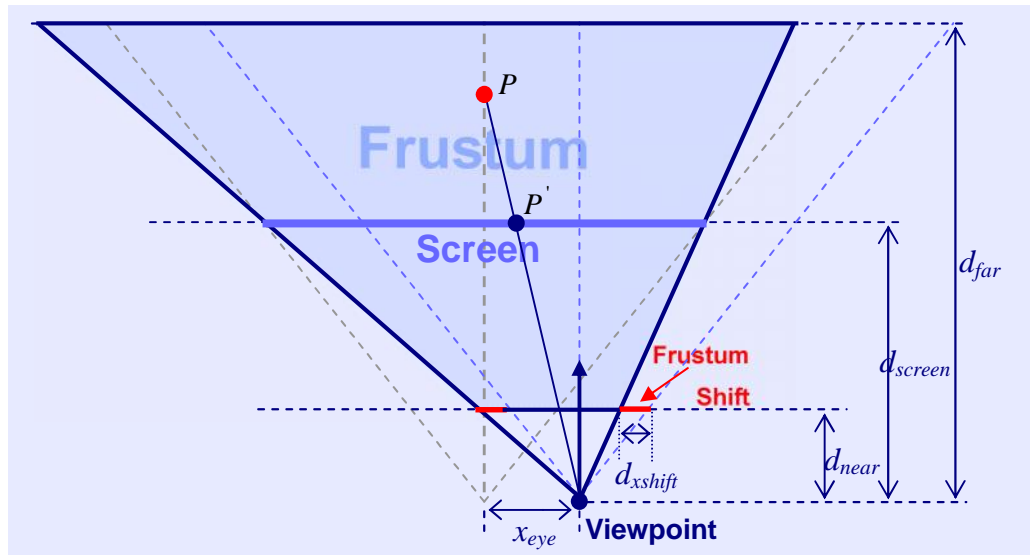


Figure 4.14 Asymmetrical Frustum Setting (Top View)

In the same way, the shift parameter along the y-axis can be calculated by Equation 4.11. The frustum shifts downward with a positive d_y , and shifts upward if d_y is negative.

$$d_{yshift} = -y_{eye} \left(\frac{d_{near}}{d_{screen}} \right) \quad (4.11)$$

If the movement of viewpoint along the z-axis is taken into account, d_{screen} is changed in real-time. Since the distance from the defined origin to the screen plane is 1.958m as described in Section 4.3.2, d_{screen} is given by

$$d_{screen} = 1.958 + z_{eye} \quad (4.12)$$

Therefore, both of d_{xshift} and d_{yshift} , as well as the viewing angle, denoted by θ , are changed in real-time, and their equations are expressed by

$$d_{xshift} = -x_{eye} \left(\frac{0.01}{1.958 + z_{eye}} \right) \quad (4.13)$$

$$d_{yshift} = -y_{eye} \left(\frac{0.01}{1.958 + z_{eye}} \right) \quad (4.14)$$

$$\theta = \arctan\left(\frac{h/2}{d_{near}}\right) = \arctan\left(\frac{1.03}{1.958 + z_{eye}}\right) \quad (4.15)$$

where h is the height of screen. In order to generate a stereo pair to support stereoscopic display, the frustum needs to be further shifted by half of intraocular distance along x-axis according to the ‘‘Off-axis’’ method introduced in Chapter 3. With an intraocular distance set by 0.06m, the shift parameters along the x-axis for views of the left and right eyes are expressed in Equation 4.16 and 4.17, respectively.

$$d_{xleft} = -(x_{eye} - 0.03) \left(\frac{0.01}{1.958 + z_{eye}} \right) \quad (4.16)$$

$$d_{xright} = -(x_{eye} + 0.03) \left(\frac{0.01}{1.958 + z_{eye}} \right) \quad (4.17)$$

4.4.3.3 Results Comparison

This section shows and compares the displaying results with and without head-tracking based frustum setting, thereby demonstrating the importance of correct frustum setting in terms of providing correct spatial information to players. Since the scene can not be rotated as explained in Section 4.4.3.1, the viewing direction is always in the forward looking direction.

If the player’s eyes move along the y-axis, the views on the screen are shown in Figures 4.15 and 4.16, which refer to low-position (table height, $y_{eye}=-0.27$) and

high-position ($3/4$ screen height $y_{eye}=0.515$) viewpoints, respectively. The red line in each figure indicates the eye level of the viewer.

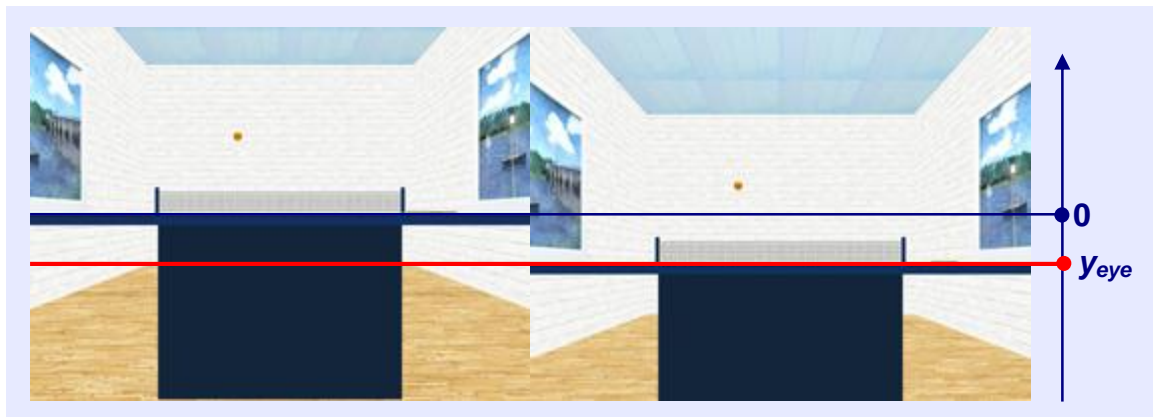


Figure 4.15 Views with Fixed (left) and Head-tracking based (right) Frustum for a Low-position Viewpoint

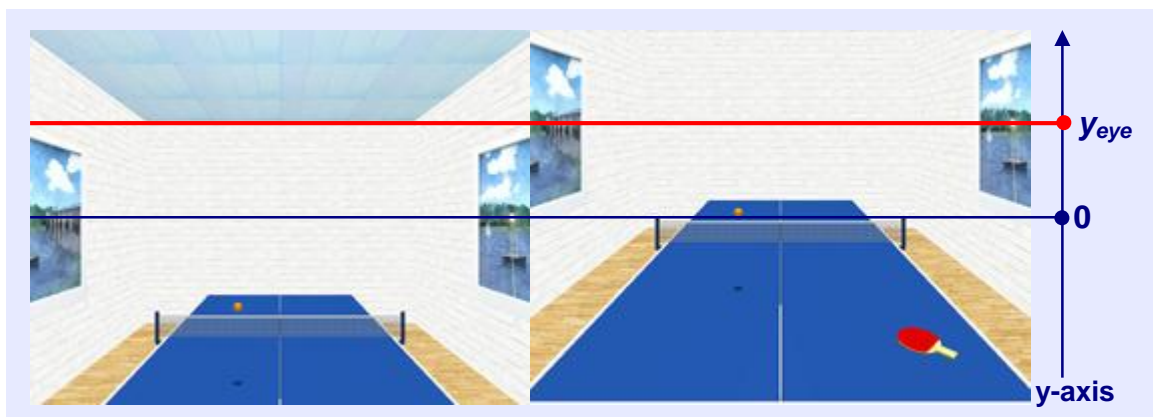


Figure 4.16 Views with Fixed (left) and Head-tracking based (right) Frustum for a High-position Viewpoint

In Figure 4.15, the table's upper surface should be on the plane of the apparent horizon. In Figure 4.16, the bricks texture near the plane of the apparent horizon should tend to be a line. Therefore, the views with head-tracking based frustum (i.e. right figures) are correct.

Figures 4.17 and 4.18 show the results that the player moves to left and right side, respectively (i.e. along the x-axis) by half of the table width. The red line in each figure indicates the horizontal position of the player's viewpoint. Since the table's side edge should be in front of the player, the views with head-tracking based frustum (i.e. bottom figures) are correct.

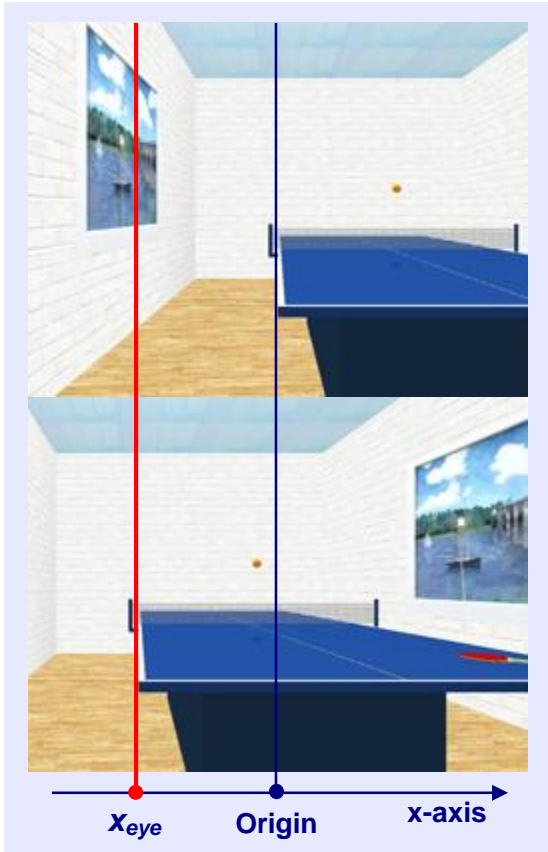


Figure 4.17 Views with Fixed (top) and Head-tracking based (bottom) Frustum for a Left-side Viewpoint

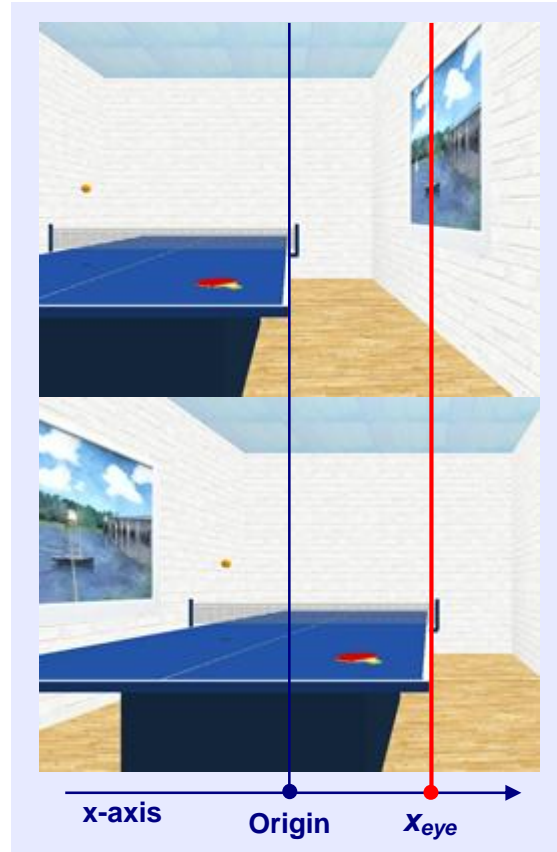


Figure 4.18 Views with Fixed (top) and Head-tracking based (bottom) Frustum for a Right-side Viewpoint

The viewpoint's movement along z-axis refers to the change of scale. If the player steps back by 1m, the views with default and head-tracking based viewing angle are shown in Figure 4.19. For the one with default viewing angle (i.e. left figure), its frustum has a default parameter of d_{screen} that equals to 1.958m. But in fact, d_{screen} changes to 2.958m due to stepping back by 1m. Therefore, the correct view should be the zoom-in version of this figure by $2.958/1.958=1.51$ times. The right figure shown in Figure 4.19 is the view with head-tracking based viewing angle, which is correct since it is same as the zoom-in version (1.51 times) of the left figure.

Figure 4.20 refers to a situation of stepping forward by 1m. The correct view should be the zoom-out version of the left figure by $0.958/1.958=0.49$ times. Since the right figure is same as the zoom-out result, it presents a correct view.

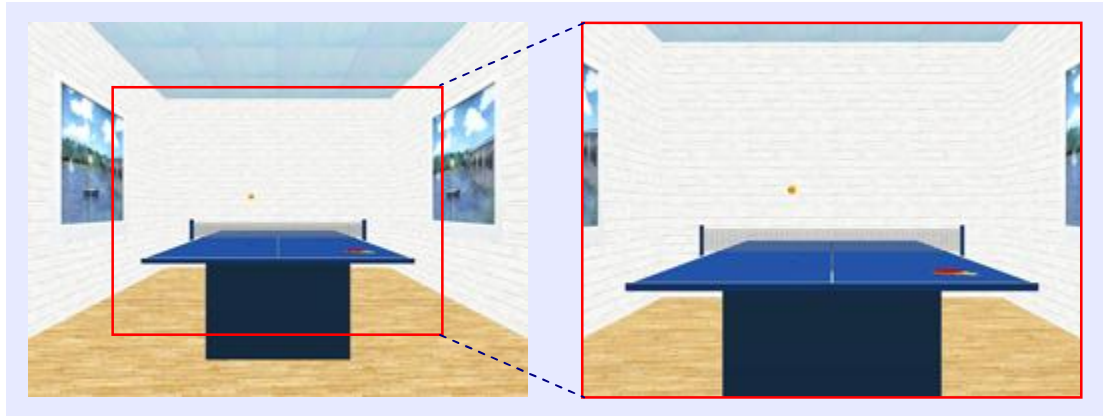


Figure 4.19 Views with Default (left) and Head-tracking based (right) Viewing Angle

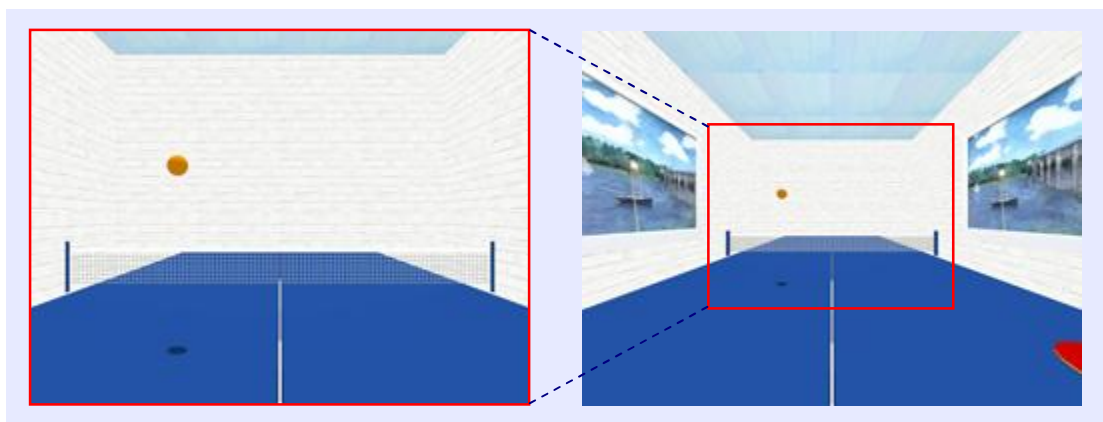


Figure 4.20 Views with Default (left) and Head-tracking based (right) Viewing Angle

Through the observation of Figures 4.15 to 4.20, the projection method introduced in the previous section is proved to be correct, and the real-time frustum adjustment based on head-tracking is shown to be very important.

4.5 Ball Animation

There are two most crucial motion states of an animated ball, flying and collision. A flying state describes a projectile motion, in which the ball keeps moving after an initial impulse (e.g. collision) based on the inertia effect. The velocity of the ball is changed by constant gravity acceleration, and affected by some aerodynamic effects,

such as air resistance (drag force) and Magnus effect. The procedure of dealing with a collision state can be described by two distinct aspects, collision detection and collision response. Collision detection refers to the determination of whether and where two objects (e.g. ball and racquet) have collided with each other, which needs to be checked every frame in a real-time simulation. Collision response is a physics problem that describes the subsequent movements of the two collided objects after collision, such as bounce, slide and stop. If the ball bounces after a collision event, new initial position and velocity of a new flying state of the ball are produced.

Different from the real world, time is discrete in computer animation. Therefore, in order to simulate a moving trajectory, the translation of the object in each sampled instances of time need to be computed (Liarokapis, 2006; Stahler, 2004). In the physics model used for this project, the linear motion of the ball is treated as particle movement (Watt and Policarpo, 2000; Bourg, 2001), in which the particle is located at its center of mass. When the ball's angular motion is taken into account, the rotation-axis of the ball passed through the particle. All the calculations described in the following sections are based on the world coordinate system.

4.5.1 Physics Model of Flying Ball

The trajectory of a flying ball is affected by three external forces, which are gravity, drag, and lift forces (Bourg, 2001; Noser, et al., 2000; Alam, et al., 2008). As illustrated in Figure 4.21, gravity force labelled by F_G acts in the downward direction constantly, while drag force, F_D , is always against the direction of motion. The direction of lift force, denoted by F_L , depends on the ball's moving and spinning direction due to the Magnus effect. For instance, if a ball rotates with backspin (as shown in the figure), it experiences an upward lift force. For a topspin ball, the

direction of lift force is downward. Therefore, the Magnus effect may causes the ball to curve upward, downward, sideways, or a combination of these.

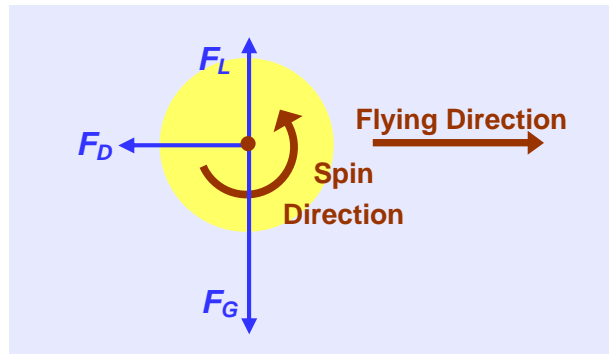


Figure 4.21 Forces Acting on a Flying Ball

Gravity Force

According to the Newton's second law and the universal formula of gravitational acceleration, gravity force is given by

$$F_G = mg \quad (4.18)$$

where m is the mass of the ball, and g is the gravitational acceleration. With $m = 2.7 \cdot 10^{-3} \text{kg}$ and $g = 9.8 \text{m/s}^2$, the gravity force acting on a table tennis ball is $26.46 \cdot 10^{-3}$ Newton.

As expressed in Equation 4.19, let \vec{n}_y be a unit vector (i.e. amplitude=1) along the positive y-axis direction, $-\vec{n}_y$ indicates that \vec{F}_G points to the downward direction.

$$\vec{F}_G = -(26.46 \cdot 10^{-3})\vec{n}_y \quad (4.19)$$

Drag Force

Drag force opposes the relative motion of an object through a fluid, which is proportional to the relative velocity or relative velocity squared depending on different cases (Timmerman, et al., 1999). For a table tennis simulation, Equation 4.20 that describes a drag force experienced by a high-speed moving object is widely used. Since air speed was set to zero in this project, the relative velocity of ball to air was

equal to the ball's velocity.

$$F_D = \frac{1}{2} C_D \rho A v^2 \quad (4.20)$$

In the equation, ρ is the density of air, and A denotes a circular cross sectional area, which are given by

$$\rho = 1.225 \text{ kg/m}^3 \text{ (Temperature=15}^\circ\text{C)} \text{ and } A = \pi r^2 \approx 1.256 \cdot 10^{-3} \text{ m}^2 \text{ (} r = 0.02 \text{ m)}$$

The drag coefficient, denoted by C_D , varies with many factors (e.g. fluid's viscosity).

By setting $C_D = 0.5$ that is an approximate drag coefficient for a smooth sphere going through air, the computed drag force is equal to $0.385 \cdot 10^{-3} \text{ kg/m}$ multiplying the velocity squared with a unit of Newton.

The formula expressed by vectors with direction information is given by

$$\vec{F}_D \approx -\left(0.385 \cdot 10^{-3}\right) \vec{v} |\vec{v}| \quad (4.21)$$

where $|\vec{v}|$ denotes the amplitude of ball's velocity \vec{v} , and the negative sign means that the direction of \vec{F}_D is opposite to the direction of \vec{v} .

Lift Force (Magnus Force)

Since different points on the surface of a spinning ball are under different air pressures, a lift force is produced, which is perpendicular to the line of motion and the rotation-axis of the ball. For a spinning smooth sphere moving through a fluid, the lift force can be approximated by

$$F_L \approx f r^3 \check{S} v \quad (4.22)$$

where \check{S} denotes the angular velocity of the ball in radians per second.

With $f r^3 \approx 0.03 \cdot 10^{-3} \text{ kg}$, the equation with direction information is expressed by

$$\vec{F}_L = \left(0.03 \cdot 10^{-3}\right) \check{S} |\vec{v}| \left(\frac{\check{S} \times \vec{v}}{|\check{S} \times \vec{v}|} \right) \quad (4.22)$$

where $\frac{\check{S} \times \vec{v}}{|\check{S} \times \vec{v}|}$ is a unit vector referred to the direction of \vec{F}_L , and \times denotes cross product.

Composed Acceleration

By using the Newton's second law of motion, the composed acceleration of a flying ball is expressed by

$$\vec{a} = \frac{\vec{F}_G + \vec{F}_D + \vec{F}_L}{m} \quad (4.23)$$

Therefore, the composed acceleration is

$$\vec{a} \approx -9.8\vec{n}_y - 0.143|\vec{v}|\vec{v} + 0.01|\check{S}||\vec{v}|\left(\frac{\check{S} \times \vec{v}}{|\check{S} \times \vec{v}|}\right) \quad (4.24)$$

Since this acceleration is not constant, it needs to be computed frame by frame based on the current linear and angular velocity of the ball. To simplify the calculation, the acceleration is regarded as constant in each time interval between update (Noser, et al., 2000). Since the program is running on a frame rate of 60FPS, the time between update is only about 0.017s. Therefore, the change of acceleration in each time interval can be ignored.

Based on the assumption of constant acceleration, if the virtual ball is at a position \vec{p} in the current frame with a velocity of \vec{v}_o , its position and velocity in the next frame (with time interval $\Delta t = 0.017s$) labeled by \vec{p}' and \vec{v}_t can be calculated by Equation 4.25 and 4.26, respectively, if there is no collision.

$$\vec{p}' = \vec{v}_o \Delta t + \frac{1}{2} \vec{a} \Delta t^2 + \vec{p} \quad (4.25)$$

$$\vec{v}_t = \vec{v}_o + \vec{a} \Delta t \quad (4.26)$$

Therefore, if the initial position, linear velocity, and angular velocity of a ball are known, the ball's moving trajectory can be computed by the above equations until a collision occurs.

4.5.2 Physics Model of Collision Detection

In a table tennis game, at least three objects collide with a ball, which are table (i.e. table' upper surface), net, and racquet (two racquets for two-player game). To simplify the collision model, all the collision events are regarded as a particle colliding with a flat rigid surface (Noser, et al., 2000). The velocity of the particle was assumed to be constant during each time interval between update.

4.5.2.1 Collision with Stationary Object

Figure 4.22 illustrates a collision event between a moving ball and a stationary object, where the object's flat surface is parallel to the plane formed by the x-axis and z-axis. As shown in the figure, a particle (i.e. ball's mass centre) is at \vec{p}_b in the current frame, and its position in the next frame at \vec{p}_b' can be calculated by Equation 4.25 if there is no collision. If the ball collides with the object's surface, the particle's trajectory changes at colliding-point \vec{p}_c . A colliding-surface is defined as a collection of all the potential colliding-points, which are parallel to the object's surface and has the same area as illustrated in the figure. The perpendicular distance from the colliding-surface to the object's surface is equal to the radius of ball, denoted by r_b . The normal of the colliding-surface is labeled by \vec{n} , which is a unit vector at the orthogonal direction to the plane (i.e. positive y-axis direction in this case).

To determine whether a collision occurs, two important conditions need to be satisfied.

1. The trajectory of the particle intersects with a colliding-plane.
2. The point of intersection is at the colliding-surface.

(Note: Colliding-plane refers to the whole 2D space, and colliding-surface belongs to the colliding-plane with the same area of object's surface)

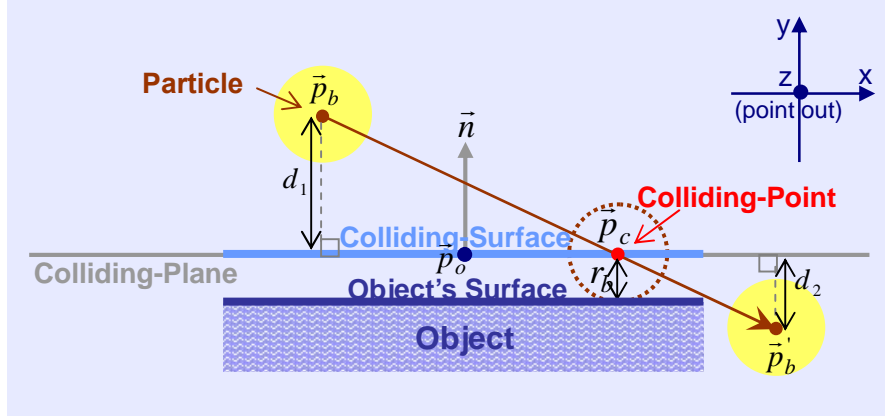


Figure 4.22 Collision between a Ball and a Stationary Flat Surface

Let d_1 and d_2 be the displacements to the collision plane from \vec{p}_b and \vec{p}_b' , respectively, the 1st condition of collision (i.e. particle passes through the colliding-plane) is expressed by

$$d_1 d_2 \leq 0 \quad (4.27)$$

$$d_1 = (\vec{p}_b - \vec{p}_o) \cdot \vec{n} \quad d_2 = (\vec{p}_b' - \vec{p}_o) \cdot \vec{n} \quad (4.28)$$

where \vec{p}_o can be any point at the colliding-plane, and \bullet denotes dot product.

Since table and net are stationary objects in the 3D space, their colliding-surfaces are known and fixed. Let $\vec{p}_t = [x_t, y_t, z_t, 1]$ be the centre position of a table's upper surface (i.e. centre of rectangle), the centre position of the table's colliding-surface \vec{p}_{tab} and the surface's normal \vec{n}_{up} are expressed by

$$\vec{p}_{tab} = [x_t, (y_t + r_b), z_t, 1]^T \quad (4.29)$$

$$\vec{n}_{up} = [0, 1, 0, 1]^T \quad (4.30)$$

Since a ball may collides with two sides of a net, there are two collision surfaces with two opposite normals, \vec{n}_{come} and \vec{n}_{away} , which are given by

$$\vec{n}_{come} = [0,0,1,1]^T \quad (4.31)$$

$$\vec{n}_{away} = [0,0,-1,1]^T \quad (4.32)$$

Let $\vec{v}_o = [x_{vo}, y_{vo}, z_{vo}, 1]^T$ be the velocity of a flying ball in the current frame, the net's colliding-surface with \vec{n}_{come} is treated as a potential colliding-surface if $z_{vo} < 0$. On the contrary, the potential colliding-surface has a normal of \vec{n}_{away} if $z_{vo} > 0$.

Let $\vec{p}_n = [x_n, y_n, z_n, 1]^T$ be the centre position of the net (i.e. centre of rectangle), the centre positions of the net's two colliding-surfaces are expressed by

$$\vec{p}_{near} = [x_n, y_n, (z_n + r_b), 1]^T \quad (4.33)$$

$$\vec{p}_{far} = [x_n, y_n, (z_n - r_b), 1]^T \quad (4.34)$$

where \vec{p}_{near} is at the colliding-surface with a normal \vec{n}_{come} , and \vec{p}_{far} with \vec{n}_{away} .

If the 1st condition of collision is satisfied, the point of intersection denoted by \vec{p}_i can be computed by

$$\vec{p}_i = \left(\frac{|d_1|}{|d_1| + |d_2|} \right) \vec{v}_b \Delta t + \vec{p}_b \quad (4.35)$$

$$\vec{v}_b = \frac{\vec{v}_o + \vec{v}_t}{2} \quad (4.36)$$

As mentioned before, this collision detection model is based on the assumption of a constant ball's velocity in each time interval between update. Therefore, \vec{v}_b is the average velocity, and \vec{v}_o and \vec{v}_t are the ball's instantaneous velocity when the ball is at \vec{p}_b and \vec{p}'_b , respectively.

In order to satisfy the 2nd condition of collision, the distance between \vec{p}_i and the centre of the corresponding colliding-surface should be in a limited range. In Figure 4.23, one colliding-surface of the table, and two opposite colliding-surfaces of the net are illustrated, respectively.

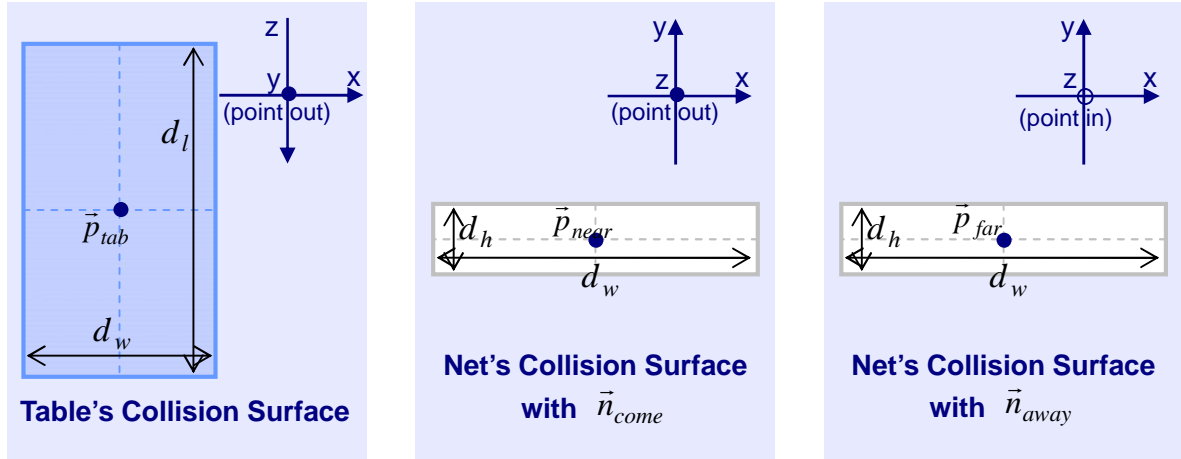


Figure 4.23 Colliding-Surfaces' Area of Table and Net

Let the position of intersection be $\vec{p}_i = [x_i, y_i, z_i, 1]^T$, and the centre positions of the colliding-surfaces be $\vec{p}_{tab} = [x_{tab}, y_{tab}, z_{tab}, 1]^T$, $\vec{p}_{near} = [x_{near}, y_{near}, z_{near}, 1]^T$ and $\vec{p}_{far} = [x_{far}, y_{far}, z_{far}, 1]^T$, the 2nd conditions of collision can be satisfied if

$$\text{Table: } |x_i - x_{tab}| \leq \frac{d_w}{2} \quad \text{and} \quad |z_i - z_{tab}| \leq \frac{d_l}{2} \quad (4.37)$$

$$\text{Net } (\vec{n}_{come}): |x_i - x_{near}| \leq \frac{d_w}{2} \quad \text{and} \quad |y_i - y_{near}| \leq \frac{d_h}{2} \quad (4.38)$$

$$\text{Net } (\vec{n}_{away}): |x_i - x_{far}| \leq \frac{d_w}{2} \quad \text{and} \quad |y_i - y_{far}| \leq \frac{d_h}{2} \quad (4.39)$$

where d_w and d_l denote the width and length of the table, and d_h is the height of the net.

Since collisions may occur more than once in a time interval between two consecutive frames, especially when the colliding-point is near both table and net as illustrated in

Figure 4.24 (the ball's trajectory in the figure is approximated). Therefore, the potential next collision needs to be calculated subsequently after the first collision is detected.

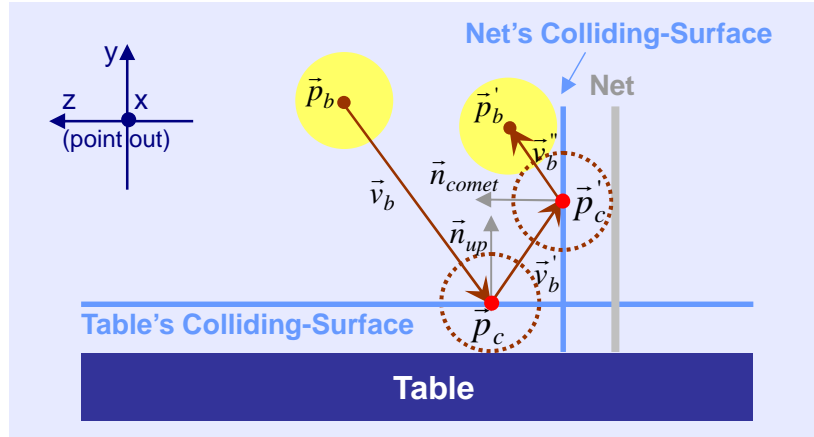


Figure 4.24 Two Collision Events in One Frame

In the figure, the ball flies onto the table from \vec{P}_b with velocity \vec{v}_b , after it collides with the table at the first colliding-point \vec{P}_c , its velocity changes to \vec{v}_b' . After the net-ball collision at the second colliding-point \vec{P}_c' , the ball arrives \vec{P}_b' with velocity \vec{v}_b'' . The method of calculating \vec{P}_c' and \vec{v}_b'' is similar as the method used for the first collision. The difference is the time interval that given by

$$\Delta t'' = \Delta t - \Delta t' = \left(\frac{|d_2|}{|d_1| + |d_2|} \right) \Delta t \quad (4.40)$$

where $\Delta t'$ is the time spend before the first collision, and $\Delta t''$ is the time that ball is flying from \vec{P}_c to \vec{P}_c' . This equation is derived from (4.35), where d_1 and d_2 are labelled in Figure 4.22.

4.5.2.2 Collision with Tracked Moving Object

Since a racquet's movement involves complex translation and orientation in high speed, the collision detection of a ball colliding with a racquet is more complex than

colliding with stationary table and net. Furthermore, since all the spatial information of a racquet used for calculation is based on the hand tracking result, there are errors between a real racquet's trajectory and a predicted one. Although this error can be minimised through complex prediction algorithm, the resultant computation burden leads to a slow speed, and affects real-time simulation. Therefore, an application-oriented prediction method was developed.

In this project, the racquet's position in the next frame is predicted by its positions in the current and previous two frames. To simplify the prediction model, this prediction is based on an assumption of constant racquet's acceleration. As illustrated in Figure 4.25, a racquet moves from \vec{P}_{pre2} , and then passes \vec{P}_{pre1} to \vec{P}_r in three consecutive frames. The average velocities in the two time intervals between update, denoted by \vec{v}_r and \vec{v}_{pre} , as well as the average acceleration \vec{a}_r can be calculated by Equations 4.41 to 4.43.

$$\vec{v}_r = \frac{\vec{P}_r - \vec{P}_{pre1}}{\Delta t} \quad (4.41)$$

$$\vec{v}_{pre} = \frac{\vec{P}_{pre1} - \vec{P}_{pre2}}{\Delta t} \quad (4.42)$$

$$\vec{a}_r = \frac{\vec{v}_r - \vec{v}_{pre}}{\Delta t} \quad (4.43)$$

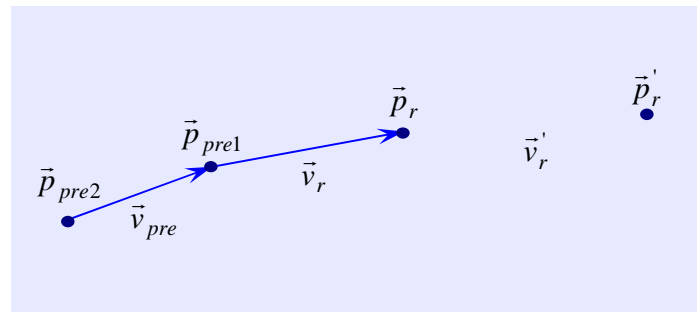


Figure 4.25 Prediction of Racquet's Position

Due to the assumption of constant racquet's acceleration, the racquet's position and velocity in the next frame can be predicted by

$$\vec{v}_r' = \vec{v}_r + \vec{a}_r \Delta t = 2\vec{v}_r - \vec{v}_{pre} \quad (4.44)$$

$$\vec{p}_r' = \vec{v}_r' \Delta t + \vec{p}_r \quad (4.45)$$

However, this prediction model may lead to a significant error if the racquet's acceleration has significant change during the three frames. For instant, if the racquet jitters, and the velocities of \vec{v}_{pre} and \vec{v}_r have opposite directions as illustrated in Figure 4.26, the predicted velocity may become very large due to large acceleration. To avoid this error, the directions of velocity along three axes need to be checked.

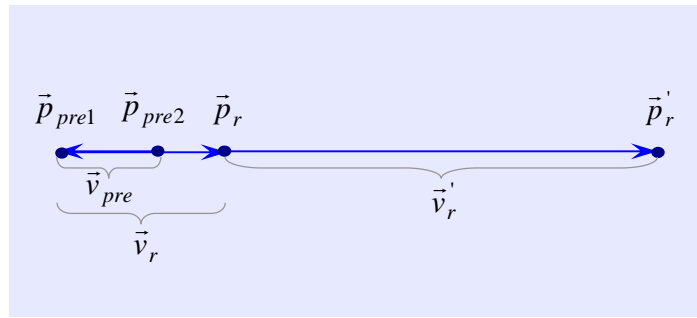


Figure 4.26 Situation of Prediction Error

Let the velocities be given by $\vec{v}_r = [x_{vr}, y_{vr}, z_{vr}, 1]^T$ and $\vec{v}_{pre} = [x_{vp}, y_{vp}, z_{vp}, 1]^T$,

the predicted velocity $\vec{v}_r' = [x_{vr}', y_{vr}', z_{vr}', 1]^T$ is expressed by

$$\text{if } x_{vr}x_{vp} > 0, \quad x_{vr}' = 2x_{vr} - x_{vp}; \text{ otherwise, } x_{vr}' = \frac{x_{vr} + x_{vp}}{2} \quad (4.46)$$

$$\text{if } y_{vr}y_{vp} > 0, \quad y_{vr}' = 2y_{vr} - y_{vp}; \text{ otherwise, } y_{vr}' = \frac{y_{vr} + y_{vp}}{2} \quad (4.47)$$

$$\text{if } z_{vr}z_{vp} > 0, \quad z_{vr}' = 2z_{vr} - z_{vp}; \text{ otherwise, } z_{vr}' = \frac{z_{vr} + z_{vp}}{2} \quad (4.48)$$

Equations 4.46 to 4.48 are based on the Equation 4.43 if the velocities in consecutive frames point to the same direction. Otherwise, the predicted velocity is approximated as an average velocity.

By using a similar approach, a racquet's orientation in the next frame can be predicted by its orientation in the current and previous two frames. This prediction model is based on an assumption of constant angular acceleration. The radian angles between a racquet's normal (defined as the red surface's normal) and the x, y, and z axes can be expressed by vector, denoted by $\vec{n}_r = [x_{n_r}, y_{n_r}, z_{n_r}, 1]^T$.

Figure 4.27 illustrates the orientation of a racquet in four consecutive frames, where the radian angles of the racquet's normal are \vec{n}_{pre2} , \vec{n}_{pre1} , \vec{n}_r and \vec{n}'_r , respectively. The average angular velocities and acceleration are given by

$$\check{S}_r = \frac{\vec{n}_r - \vec{n}_{pre1}}{\Delta t} \quad (4.49)$$

$$\check{S}_{pre} = \frac{\vec{n}_{pre1} - \vec{n}_{pre2}}{\Delta t} \quad (4.50)$$

$$\vec{r}_r = \frac{\check{S}_r - \check{S}_{pre}}{\Delta t} \quad (4.51)$$

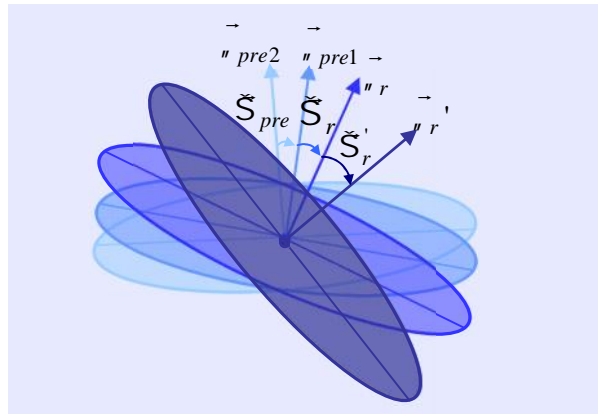


Figure 4.27 Prediction of Racquet's Rotation

Let the angular velocities be given by $\check{S}_{pre} = [x_{\check{S}_p}, y_{\check{S}_p}, z_{\check{S}_p}, 1]^T$ and $\check{S}_r = [x_{\check{S}_r}, y_{\check{S}_r}, z_{\check{S}_r}, 1]^T$, the predicted angular velocity $\check{S}'_r = [x_{\check{S}'_r}, y_{\check{S}'_r}, z_{\check{S}'_r}, 1]^T$ is expressed by Equations 4.52 to 4.54.

$$\text{if } x_{\check{S}_r}x_{\check{S}_p} > 0, \quad x'_{\check{S}_r} = 2x_{\check{S}_r} - x_{\check{S}_p}; \text{ otherwise, } x'_{\check{S}_r} = \frac{x_{\check{S}_r} + x_{\check{S}_p}}{2} \quad (4.52)$$

$$\text{if } y_{\check{S}_r}y_{\check{S}_p} > 0, \quad y'_{\check{S}_r} = 2y_{\check{S}_r} - y_{\check{S}_p}; \text{ otherwise, } y'_{\check{S}_r} = \frac{y_{\check{S}_r} + y_{\check{S}_p}}{2} \quad (4.53)$$

$$\text{if } z_{\check{S}_r}z_{\check{S}_p} > 0, \quad z'_{\check{S}_r} = 2z_{\check{S}_r} - z_{\check{S}_p}; \text{ otherwise, } z'_{\check{S}_r} = \frac{z_{\check{S}_r} + z_{\check{S}_p}}{2} \quad (4.54)$$

Then the predicted orientation is calculated by

$$\vec{n}'_r = \check{S}'_r \Delta t + \vec{n}_r \quad (4.55)$$

Same as the situation of a ball colliding with stationary objects, a collision event between a ball and a moving racquet is detected if the two conditions are satisfied. For the 1st condition, the particle should go through a colliding-plane of the racquet. For the 2nd one, the distance between the intersection and the centre of the colliding-surface should be smaller than the radius of the racquet.

Since a racquet has two potential colliding-surfaces with two opposite normals of \vec{n}_{red} and \vec{n}_{black} as illustrated in Figure 4.28, a collision surface is considered as a correct one if it satisfies

$$\vec{v}_b \bullet \vec{n} < 0 \quad (4.56)$$

where \vec{v}_b is the ball's velocity and \vec{n} (i.e. \vec{n}_{red} or \vec{n}_{black}) is the normal of the detected collision plane.

Since the normal of red surface is defined as the racquet's normal, it can be obtained from the orientation data of hand tracking. The normal of black surface that points to the opposite direction is given by Equation 4.57.

$$\vec{n}_{black} = -\vec{n}_{red} \quad (4.57)$$

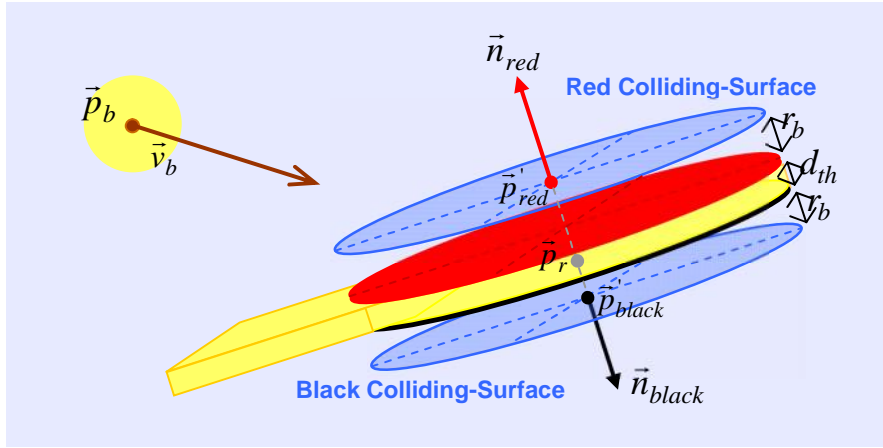


Figure 4.28 Two Colliding-Surfaces of a Racquet

Therefore, the positions of the two colliding-surfaces' centres are expressed by

$$\vec{p}_{red} = \vec{p}_r + \left(r_b + \frac{d_{th}}{2} \right) \vec{n}_{red} \quad (4.58)$$

$$\vec{p}_{black} = \vec{p}_r + \left(r_b + \frac{d_{th}}{2} \right) (-\vec{n}_{red}) \quad (4.59)$$

where r_b and d_{th} are the radius and thickness of the racquet, and \vec{p}_r denotes the racquet's centre.

Since both ball and racquet are moving objects, to detect the intersection between a ball's trajectory and a racquet's colliding-plane, the position of the ball needs to be translated into a moving coordinate system with respect to the racquet (Noser, et al., 2000). This translation is illustrated in Figure 4.29, where a ball flies from \vec{p}_b to \vec{p}_b' in two consecutive frames with velocity \vec{v}_b if there is no collision, and a racquet moves from \vec{p}_r to \vec{p}_r' with velocity \vec{v}_r at the same time. To detect the collision, the predicted position \vec{p}_b' is shifted to \vec{p}_b'' by the negative movement of the racquet. This transforms the moving racquet into a static one with respect to the ball flying from \vec{p}_b to \vec{p}_b'' with a velocity of \vec{v}_b' expressed by

$$\vec{v}_b' = \vec{v}_b - \vec{v}_r \quad (4.60)$$

$$\vec{p}_b'' = \vec{v}_b' \Delta t + \vec{p}_b \quad (4.61)$$

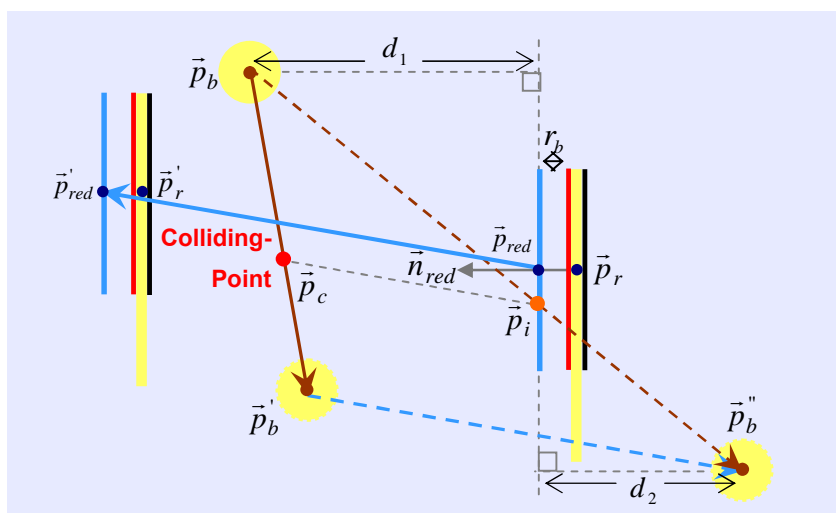


Figure 4.29 Relative Movement of Ball to Racquet

By using (4.35) and (4.36) introduced in the previous section, a point of intersection labelled by \vec{p}_i can be detected, which is given by

$$\vec{p}_i = \left(\frac{|d_1|}{|d_1| + |d_2|} \right) \vec{v}_b' \Delta t + \vec{p}_b \quad (4.62)$$

To determine whether \vec{p}_i is at the colliding-surface of the racquet's red side, Equation 4.63 needs to be satisfied.

$$|\vec{p}_i - \vec{p}_{red}| \leq r_b \quad (4.63)$$

Then the colliding-point can be calculated by Equation 4.64.

$$\vec{p}_c = \left(\frac{|d_1|}{|d_1| + |d_2|} \right) \vec{v}_b' \Delta t + \vec{p}_b \quad (4.64)$$

4.5.3 Physics Model of Collision Response

The simplest rebounding model used for ball-plane collisions is normally stated as “angle of incidence equals to angle of reflection” (Garwin, 1969), which describes a

collision in an ideal frictionless elastic collision without energy loss. However, this simple model is not sufficient for a realistic table tennis simulation. In practice, the ball's trajectory is largely affected by friction, as well as the velocity reduced by energy loss. The force acting on the contact point on the ball can be investigated by a component along surface's normal direction and a component along ball's tangential direction. In the physics model described in this section, the contacting time during collision and the racquet's rotation at the brief moment of collision are not taken into account.

4.5.3.1 Rebounding from Stationary Surface

When a ball collides with a flat plane, the friction force that acts on the contact point on the ball is along the ball's tangential direction and tends to resist movement. This friction does not only create a torque that changes the ball's angular velocity in terms of spinning speed and rotational direction, but also changes the ball's translation velocity (i.e. linear velocity of mass centre) at the same time.

In order to investigate the effect of friction, Figure 4.30 illustrates two examples of the ball-table collision. In Figure 4.30(a), a ball without spin flies onto a table with approaching velocity \vec{v}_b . Since there is a linear velocity of the contact point \vec{v}_f that equals to the horizontal velocity component of \vec{v}_b (i.e. ball's tangential direction), denoted by \vec{v}_{bt} , the contact point suffers a friction force F_f along the opposite direction of \vec{v}_{bt} . This friction force not only reduces the ball's translation velocity along the horizontal direction, but also makes the ball spin. For the rebound dynamics along the normal direction, there is a damping factor that reduces the rebound speed.

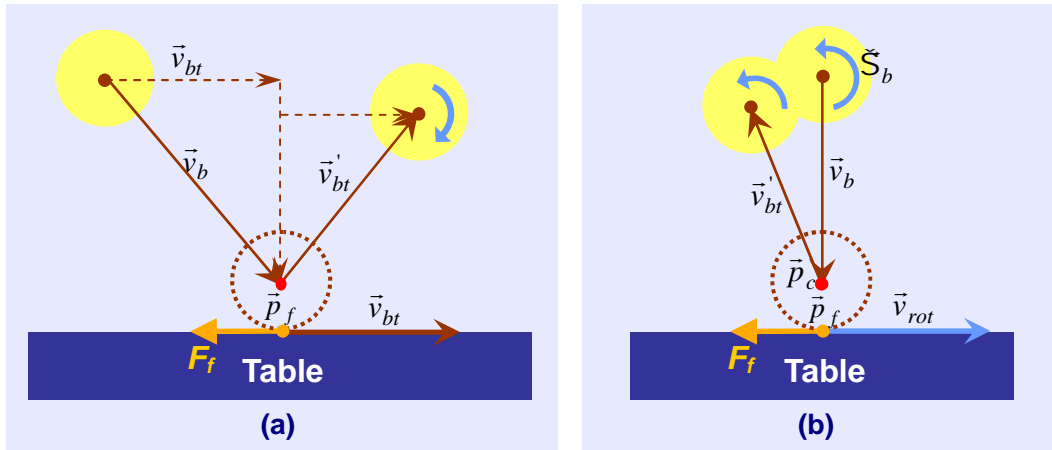


Figure 4.30 Two Examples of Ball-Table Collision

In the situation illustrated in Figure 4.30(b), a spinning ball vertically drops on a table. Although there is no velocity component along horizontal direction of the mass centre \vec{p}_c , the contact point \vec{p}_f has a linear velocity due to the ball's spinning. After collision, the friction force F_f leads to a horizontal translation velocity, and reduces the ball's angular velocity.

The friction (dry friction) is subdivided into kinetic friction and static friction, and its direction always resists movement (kinetic friction) or potential movement (for static friction). In the case of ball-table collisions, if there is a relative movement between the table and ball's contact point at the moment of contact, the ball suffers a kinetic friction force. According to the Coulomb's Law of Friction, kinetic friction denoted by

\vec{F}_k is expressed by

$$\vec{F}_k = \sim_k |\vec{F}_n| \vec{i}_f \quad (4.65)$$

where \sim_k is the kinetic CoF (coefficient of friction) (Nakashima, et al, 2009; Bourg, 2001) and \vec{i}_f is a unit vector that indicates the direction of friction. The amplitude of a dynamic friction force is proportional to the amplitude of normal reaction force \vec{F}_n (i.e. vertical impact force acting on the ball). If the relative velocity between the table and ball's contact point equals to zero, the friction is static. The amplitude of this static friction depends on the force acting on the ball along its tangential direction.

The maximum possible static friction force $\vec{F}_{s\max}$ is given by Equation 4.66, by using a static CoF, denoted by \sim_s .

$$\vec{F}_{s\max} = \sim_s |\vec{F}_n| \vec{i}_f \quad (4.66)$$

Since the values of kinetic and static friction depend on different components of the impact force, applying a correct type of friction is important to calculate the ball's new translation and rotational velocities after collision.

In practice, the linear velocity of the contact point on the ball is a composition of the ball's linear velocity component along the horizontal direction and a linear rotational velocity caused by ball's spin. As illustrated in Figure 4.31, a ball flies onto the table with an approaching linear velocity \vec{v}_b and an angular velocity \check{S}_b . At the moment of collision, the contact velocity (i.e. composition linear velocity of the contact point \vec{p}_f) is expressed by Equation 4.67.

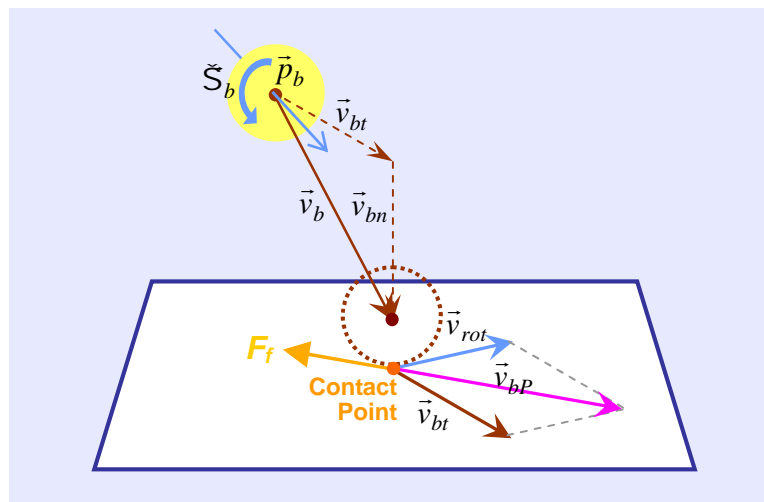


Figure 4.31 Ball-Table Collision

$$\vec{v}_{bP} = \vec{v}_{bt} + \vec{v}_{rot} \quad (4.67)$$

where \vec{v}_{bt} is the horizontal component of \vec{v}_b , and \vec{v}_{rot} a linear velocity caused by \check{S}_b , which can be computed by Equations 4.68 and 4.69.

$$\vec{v}_{bt} = \vec{v}_b - (\vec{v}_b \bullet \vec{n}_{up}) \vec{n}_{up} \quad (4.68)$$

$$\vec{v}_{rot} = r_b (\vec{n}_{up} \times \vec{\xi}_b) \quad (4.69)$$

This contact velocity results in a friction F_f , which gives impulse on the ball's tangential direction. Let \vec{P}_f be this impulse, which can be expressed as

$$\vec{P}_f = m_b (\vec{v}'_{bt} - \vec{v}_{bt}) \quad (4.71)$$

$$r_b \vec{P}_f \times \vec{n}_{up} = I_b (\vec{\xi}'_b - \vec{\xi}_b) \quad (4.72)$$

where \vec{v}'_{bt} is the separation linear velocity component along tangential direction, and $\vec{\xi}'_b$ denotes angular velocity after collision. Parameter I_b indicates the moment of inertia of the ball, which is a measure of an object's resistance to the change of rotation. By taking the ball as a thin spherical shell (Cross, 2002), I_b is given by

$$I_b = \frac{2}{3} m_b r_b^2 \quad (4.73)$$

Let \vec{v}'_{bP} be the separation contact velocity, according to (4.67) and (4.69), (4.72) can be rewritten as

$$\vec{P}_f = \frac{I_b}{r_b^2} [\vec{v}'_{rot} - \vec{v}_{rot}] = \frac{I_b}{r_b^2} [(\vec{v}'_{bP} - \vec{v}'_{bt}) - (\vec{v}_{bP} - \vec{v}_{bt})] \quad (4.74)$$

By combining (4.71), (4.73) and (4.74), the separation contact velocity is yielded as

$$\vec{v}'_{bP} = \left(\frac{r_b^2}{I_b} + \frac{1}{m_b} \right) \vec{P}_f + \vec{v}_{bP} = \frac{5}{2m_b} \vec{P}_f + \vec{v}_{bP} \quad (4.75)$$

Let \vec{i}_f be the direction of impulse as given in Equation 4.76, \vec{P}_f can be expressed by the product of its amplitude and a direction vector. (4.75) is then substituted by Equation 4.77.

$$\vec{i}_f = -\frac{\vec{v}_{bP}}{|\vec{v}_{bP}|} \quad (4.76)$$

$$\vec{v}'_{bP} = \left(1 - \frac{5}{2m_b} \frac{|\vec{P}_f|}{|\vec{v}_{bP}|} \right) \vec{v}_{bP} \quad (4.77)$$

In order to judge the type of friction, the calculation is conducted based on an assumption of “friction is kinetic”. According to (4.65), the amplitude of \vec{P}_f is given by

$$|\vec{P}_f| = \sim_k |\vec{P}_n| \quad (4.78)$$

where \vec{P}_n is the impulse along surface’s normal direction that can be expressed as

$$\vec{P}_n = m_b (\vec{v}'_{bn} - \vec{v}_{bn}) \quad (4.79)$$

With respect to the dynamics along surface’s normal direction, Equation 4.66 expresses a coefficient of restitution (CoR) (Bourg, 2001; Cross, 2000) that is employed to simulate the energy loss along normal direction. This coefficient depends on the characteristics of colliding objects, such as material, construction and geometry.

$$e = - \frac{|\vec{v}'_{1n} - \vec{v}'_{2n}|}{|\vec{v}_{1n} - \vec{v}_{2n}|} \quad (4.80)$$

where $\vec{v}_{1n} - \vec{v}_{2n}$ expresses the approaching relative velocity of object1 to object2, and $\vec{v}'_{1n} - \vec{v}'_{2n}$ refers to their relative velocity after collision.

In the case of a ball colliding with a stationary table, the CoR that is denoted by e_{tab} is given by

$$\vec{v}'_{bn} = -e_{tab} \vec{v}_{bn} \quad (4.81)$$

By combining equations of (4.78), (4.79) and (4.81), the separation contact velocity is expressed in Equation 4.82.

$$\vec{v}'_{bP} = \left[1 - \frac{5}{2}(1 + e_{tab})\tilde{\sim}_k \frac{|\vec{v}_{bn}|}{|\vec{v}_{bP}|} \right] \vec{v}_{bP} \quad (4.82)$$

According to the ITTF's standard, a ball should bounce about 23cm from the table if it is dropped on to it from a height of 30cm (ITTF, 2004). By assuming that the ball's trajectory in air is only affected by gravity, the speed of the ball at the moment of collision is about 2.42m/s. To achieve a height of 23cm, the initial speed after bounce should be 2.12m/s. Therefore, the CoR of a ball-table collision can be calculated as

$$e_{tab} = \frac{|\vec{v}'_{bn}|}{|\vec{v}_{bn}|} = \frac{2.12}{2.42} = 0.88. \quad (4.83)$$

Since \vec{v}'_{bP} and \vec{v}_{bP} are in the same direction, the type of friction can be determined based on (4.82).

$$\text{Kinetic Friction if } \frac{|\vec{v}_{bn}|}{|\vec{v}_{bt} + \vec{v}_{rot}|} \leq \frac{0.213}{\tilde{\sim}_k} \quad (4.84)$$

$$\text{Static Friction if } \frac{|\vec{v}_{bn}|}{|\vec{v}_{bt} + \vec{v}_{rot}|} > \frac{0.213}{\tilde{\sim}_k} \quad (4.85)$$

If the friction force is kinetic, a ball's spin becomes slower and slower. If the friction is static, the ball's spin is stopped. (A ball's velocity at the contact point is zero in the case of static friction. Nakashima, et al, 2009) According to (4.84) and (4.85), a

bigger $\tilde{\sim}_k$ gives a smaller boundary value of $\frac{|\vec{v}_{bn}|}{|\vec{v}_{bt} + \vec{v}_{rot}|}$ to achieve static friction.

In other words, a bigger kinetic CoF requires smaller vertical velocity (compare with horizontal velocity) at contact point to stop the spin.

According to the ITTF standard, the kinetic CoF between the ball and table's upper surface should not be greater than 0.6. For the table tennis tables in market, $\tilde{\sim}_k$ are

usually from 0.3 to 0.6 depend on different material. In this project, μ_k was set by 0.5. Therefore, (4.84) and (4.85) can be rewritten by

$$\text{Kinetic Friction if } \frac{|\vec{v}_{bn}|}{|\vec{v}_{bt} + \vec{v}_{rot}|} \leq 0.426 \quad (4.86)$$

$$\text{Static Friction if } \frac{|\vec{v}_{bn}|}{|\vec{v}_{bt} + \vec{v}_{rot}|} > 0.426 \quad (4.87)$$

With respect to the static friction, \vec{v}'_{bP} is zero, therefore the impulse due to friction can be calculated according to (4.77).

$$\vec{P}_f = -\frac{2}{5}m_b\vec{v}_{bP} \quad (4.88)$$

Combining with (4.71) and (4.74), the translation linear velocity and the rotational linear velocity after collision can be computed by

$$\vec{v}'_{bt} = 0.6\vec{v}_{bt} - 0.4\vec{v}_{rot} \quad (4.89)$$

$$\vec{v}'_{rot} = -0.6\vec{v}_{bt} + 0.4\vec{v}_{rot} \quad (4.90)$$

With respect to the kinetic friction, by combining the three equations of impulse (4.71), (4.74) and (4.82), the computed results for the separation linear and angular velocities along ball's tangential direction are expressed in Equations 4.91 and 4.92.

$$\vec{v}'_{bt} = \left(1 - 0.94 \frac{|\vec{v}_{bn}|}{|\vec{v}_{bt} + \vec{v}_{rot}|}\right) \vec{v}_{bt} - 0.94 \frac{|\vec{v}_{bn}|}{|\vec{v}_{bt} + \vec{v}_{rot}|} \vec{v}_{rot} \quad (4.91)$$

$$\vec{v}'_{rot} = -1.41 \frac{|\vec{v}_{nb}|}{|\vec{v}_{bt} + \vec{v}_{rot}|} \vec{v}_{bt} + \left(1 - 1.41 \frac{|\vec{v}_{nb}|}{|\vec{v}_{bt} + \vec{v}_{rot}|}\right) \vec{v}_{rot} \quad (4.92)$$

In the surface's normal direction, according to (4.81), the separation velocity is given by

$$\vec{v}'_{bn} = -0.88\vec{v}_{bn} \quad (4.93)$$

In a ball-net collision, the net is deformed due to the impact force, and there is no longer only one contact point on the ball. Therefore, the realistic physics model for ball-net collisions is different from ball-table collisions and much more complex. However, since the ball's subsequent trajectory is normally not important after the ball hitting the net, this project uses a very simple model when dealing with the ball-net collision, which is expressed as

$$\vec{v}'_{bz} = -0.6\vec{v}_{bz} \quad \text{and} \quad \check{S}'_b = 0 \quad (4.94)$$

There is a special case of “net ball”, which means that the ball touches the top of net and keeps going on. To simulate this case, if the ball's trajectory intersects with net's colliding plane, and the vertical distance between the intersection and the top of net is smaller than the radius of ball, the ball's linear and angular velocities are approximated by multiplying a damping coefficient (set by 0.6) without direction change that is given by

$$\vec{v}'_b = 0.6\vec{v}_b \quad \text{and} \quad \check{S}'_b = 0.6\check{S}_b \quad (4.95)$$

4.5.3.2 Rebounding from Moving Surface

To simplify the situation, the ball is translated into a moving coordinate system with respect to the racquet (Cross, 2000). The method of this translation is illustrated in Figure 4.29, and the ball's approaching velocity relative to the racquet is expressed by

$$\vec{v}_{br} = \vec{v}_b - \vec{v}_r \quad (4.96)$$

where \vec{v}_r is the racquet's linear velocity.

Most of the equations introduced in the last section for the ball-table collision can be used to compute the ball-racquet collision as well. Since the normal of the contact surface is changed, in (4.68), (4.69), and (4.72), the table's normal \vec{n}_{up} should be replaced by the racquet's normal (i.e. \vec{n}_{red} or $-\vec{n}_{red}$).

In this way, Equation 4.97 is derived from (4.82), where \vec{v}_{brP} and \vec{v}'_{brP} are the contact velocities relative to the racquet along ball's tangential direction before and after collision, and \vec{v}_{brn} denotes the normal velocity component of \vec{v}_{br} . Besides, e_{rac} indicates CoR in ball-racquet collision.

$$\vec{v}'_{brP} = \left[1 - \frac{5}{2}(1 + e_{rec}) \sim_k \frac{|\vec{v}_{brn}|}{|\vec{v}_{brP}|} \right] \vec{v}_{brP} \quad (4.97)$$

CoR and CoF are different for different racquets due to their different rubber materials. Xiao (Xiao, 2001) experimentally measured CoRs and kinetic CoFs of 11 racquets from different brand in market. The results showed that their CoR were from 0.69 to 0.72, and kinetic CoF were in a range of 0.70 to 0.99. A bigger CoR leads to a bigger rebound speed due to less energy loss, and as explained previously, a bigger kinetic CoF requires smaller vertical velocity (compare with horizontal velocity) at contact point to stop the ball's spin. The values employed in this project were $e_{rac} = 0.7$ and $\sim_{kr} = 0.8$, which are proper for a common racquet and can be adjusted based on player's requirement.

By calculating (4.97) and according to (4.86) and (4.87), the type of friction can be determined by

$$\text{Kinetic Friction if } \frac{|\vec{v}_{brn}|}{|\vec{v}_{brt} + \vec{v}_{rot}|} \leq 0.29 \quad (4.98)$$

$$\text{Static Friction if } \frac{|\vec{v}_{brn}|}{|\vec{v}_{brt} + \vec{v}_{rot}|} > 0.29 \quad (4.99)$$

By employing the same method used for the ball-table collision, the translation linear velocity and the rotational linear velocity after collision can be computed by

$$\text{Static Friction: } \vec{v}'_{brt} = 0.6\vec{v}_{brt} - 0.4\vec{v}_{rot} \quad (4.100)$$

$$\text{Static Friction: } \vec{v}'_{rot} = -0.6\vec{v}_{brt} + 0.4\vec{v}_{rot} \quad (4.101)$$

$$\text{Kinetic Friction: } \vec{v}'_{bri} = \left(1 - 1.36 \frac{|\vec{v}_{brn}|}{|\vec{v}_{bri} + \vec{v}_{rot}|}\right) \vec{v}_{bri} - 1.36 \frac{|\vec{v}_{brn}|}{|\vec{v}_{bri} + \vec{v}_{rot}|} \vec{v}_{rot} \quad (4.102)$$

$$\text{Kinetic Friction: } \vec{v}'_{rot} = -2.04 \frac{|\vec{v}_{brn}|}{|\vec{v}_{bri} + \vec{v}_{rot}|} \vec{v}_{bri} + \left(1 - 2.04 \frac{|\vec{v}_{brn}|}{|\vec{v}_{bri} + \vec{v}_{rot}|}\right) \vec{v}_{rot} \quad (4.103)$$

$$\text{Normal Direction: } \vec{v}'_{brn} = -0.7 \vec{v}_{brn} \quad (4.104)$$

However, when dealing with the dynamics of impact along the racquet's normal direction, the racquet can not be treated as a fixed "ground" such as the table, therefore its mass needs to be taken into account. Based on the CoR's definition (4.80), the CoR in a ball-racquet collision is defined as

$$e_{rac} (\vec{v}_{bn} - \vec{v}_{rn}) = \vec{v}'_{rn} - \vec{v}'_{bn} \quad (4.105)$$

By applying the principle of conservation of kinetic momentum expressed in Equation 4.106, and combining (4.105), the ball's separation velocity component along the normal direction after collision is expressed by Equation 4.107

$$\vec{v}_{bn} m_b + \vec{v}_{rn} m_r = \vec{v}'_{bn} m_b + \vec{v}'_{rn} m_r \quad (4.106)$$

$$\vec{v}'_{bn} = \left(\frac{m_b - e_{rac} m_r}{m_b + m_r} \right) \vec{v}_{bn} + \left(\frac{m_r + e_{rac} m_r}{m_b + m_r} \right) \vec{v}_{rn} \quad (4.107)$$

In the above equations, \vec{v}_{rn} and \vec{v}'_{rn} denote the racquet's velocity components along the normal direction before and after collision. In practice, if the racquet is tightly grasped in a player's hand, m_r approaches the sum of ball's mass and player's mass, which is much bigger than the mass of racquet. Therefore, by assuming that $m_r \gg m_b$, the mass of ball can be ignored. In this way, the ball's velocity along the normal direction can be calculated.

$$\vec{v}'_{bn} = -e_{rac} \vec{v}_{bn} + (1 + e_{rac}) \vec{v}_{rn} = -0.7 \vec{v}_{bn} + 1.7 \vec{v}_{rn} \quad (4.108)$$

4.6 Conclusion

The implementation of a real-time single player table tennis game is introduced in this chapter in details. The InterSense motion tracking system and the stereoscopic display system introduced in Chapter 3 have been successfully integrated. Virtual objects were generated according to physical measurement of real objects. Through the system calibration and coordinate transformation, the virtual racquet was animated correctly as player's action. The virtual camera in the 3D scene and the frustum volume were set in real-time based on tracked viewpoint. A complete design of the physics-based ball animation model was developed, and the detailed analysis of the ball's trajectory is described in this chapter as well.

Chapter 5

Virtual Reality Environment for Two-player Table Tennis

5.1 Introduction

In order to implement a highly competitive table tennis game, the single-player system introduced in Chapter 4 was extended to a two-player game environment, which is presented in this chapter. There are six major differences between a single-player game and a two-player game: (1) A more complicated communication workflow was implemented for data transfers between server and client computers. (2) Two more trackers were added to acquire another player's viewing position and racquet movement information. (3) One more screen was provided for immersive display of the game for the second player. (4) The world coordinate system had to be transformed to each player's local viewing coordinate frame, so that all the actions can be visualised correctly by the opponent. (5) The SR camera was integrated to the system to enable a real-time 3D representation of the player's real appearance and action. (6) The game menu and strategy were introduced to control the whole workflow and determined the winner of the game. Detailed descriptions of each part can be found in the following sections.

5.2 System Implementation and Communication

The hardware configuration for the two-player game is illustrated in Figure 5.1. One more screen, two more trackers and PCs, as well as a 3D camera were successfully integrated into the system. The operation principle of the two-player system is similar to the system developed for single user, and the main difference is the need of data communication.

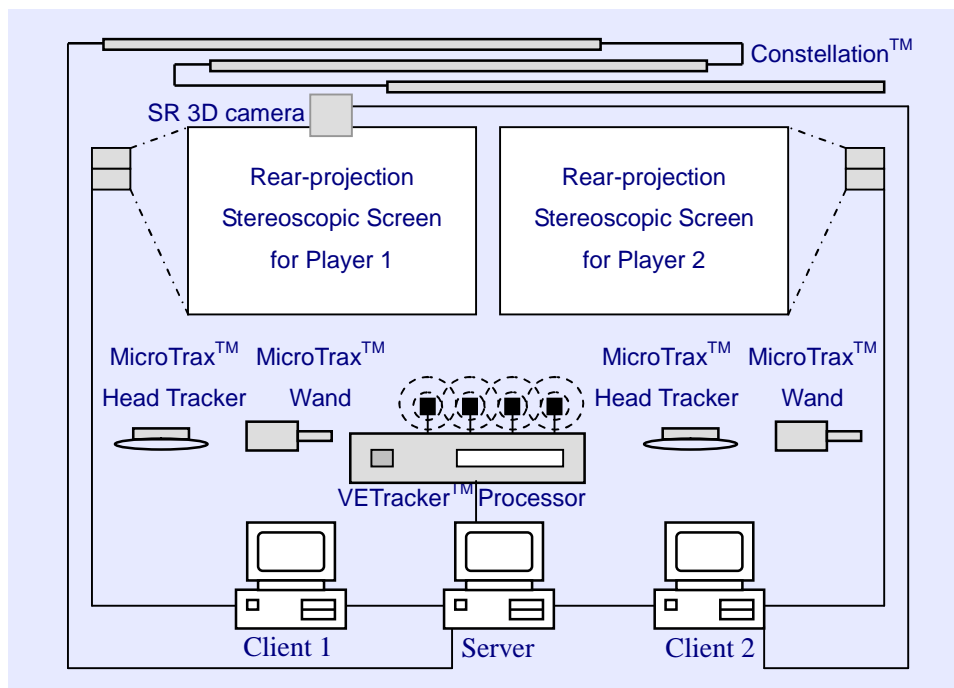


Figure 5.1 Hardware Configuration of Two-player System

Some current single-user real-time VR systems employed two computers to share the computation burden through client-server communication (Wan, et al., 2011). This idea can be extended to be used for multiuser system. In this project, a special client-server architecture was designed for data communication among three PCs. The overall system workflow is illustrated in Figure 5.2, where the server PC is responsible for control of the InterSense tracking system, motion data processing, and application related computation (e.g. calculation of the ball's position). Two client PCs render the 3D scene according to the computation results of the server. Since two players are in the same room, the body tracking data captured by the 3D SwissRanger

camera is transmitted to the opposite computer directly to reduce transmission burden (The current system uses only one 3D camera, another one can be easily integrated to the system in the same way).

The server PC is in charge of motion data acquisition, physical model calculation, game status control and data transfer to client PCs. At each updating frame, the head-tracking and hand-tracking data of the two players are streamed from the tracking devices. By applying the spatial information of the racquet, and combining with the collision detection algorithm (in Section 4.5.2), the ball's position can be calculated. The computed ball's position (3-DOF), head's position (3 DOF) and racquet's movement (6-DOF) of each player are then transferred to the corresponding client PCs through the TCP/IP protocol. Another major task performed by the server is to control the status of the game (e.g. training or competition, play or pause). This part of work will be presented in Section 5.5.

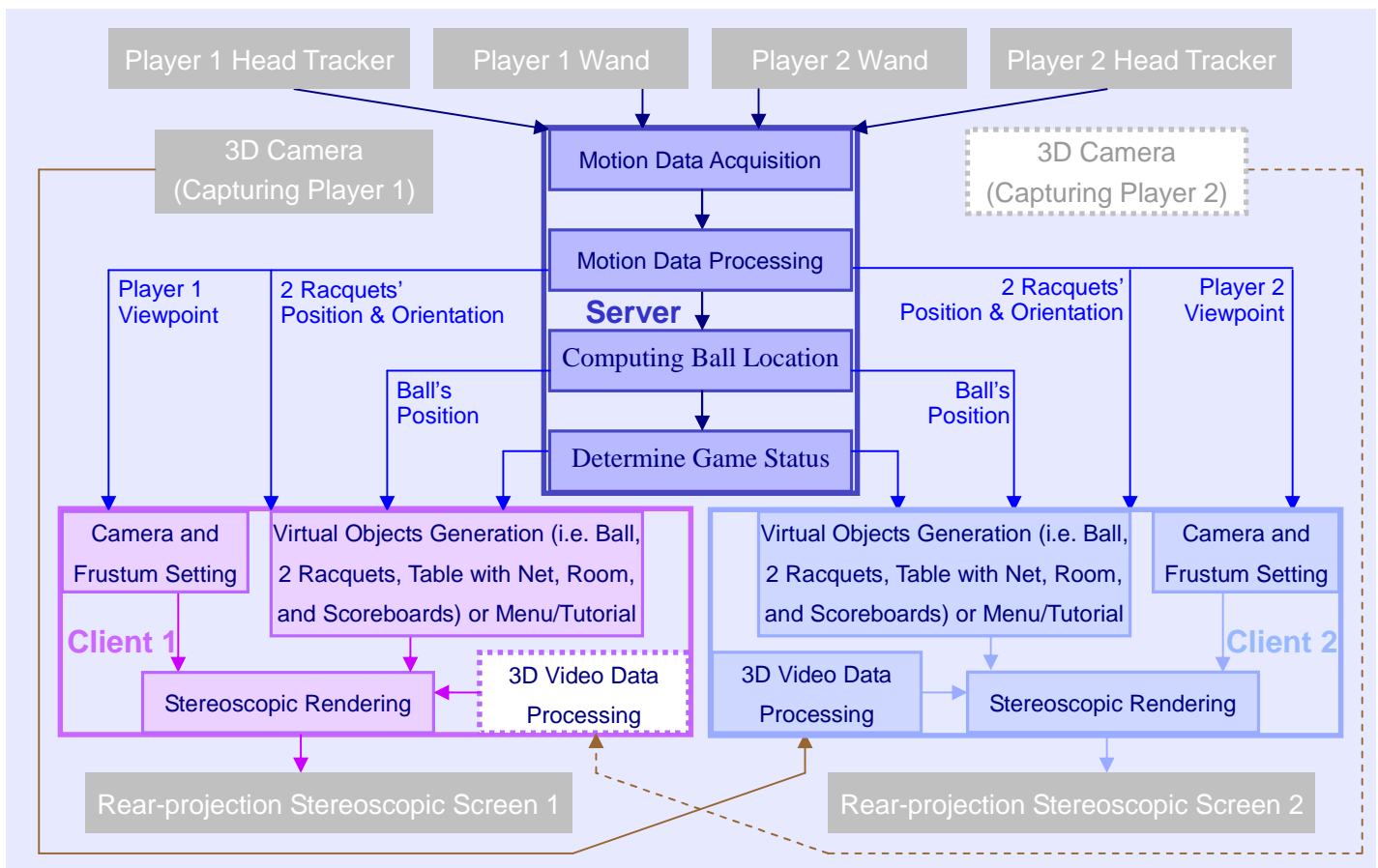


Figure 5.2 Workflow of Two-player System

With respect to the client PCs, after receiving the data from the server, the head's position is used to set the virtual camera position and viewing frustum, whereas the racquet's position and orientation, as well as ball's position are used for object generation. Additionally, the 3D mesh of the opponent is generated based on the data acquired from the SR 3D camera. Finally, the entire game environment at the current frame is rendered by the client PC for each player.

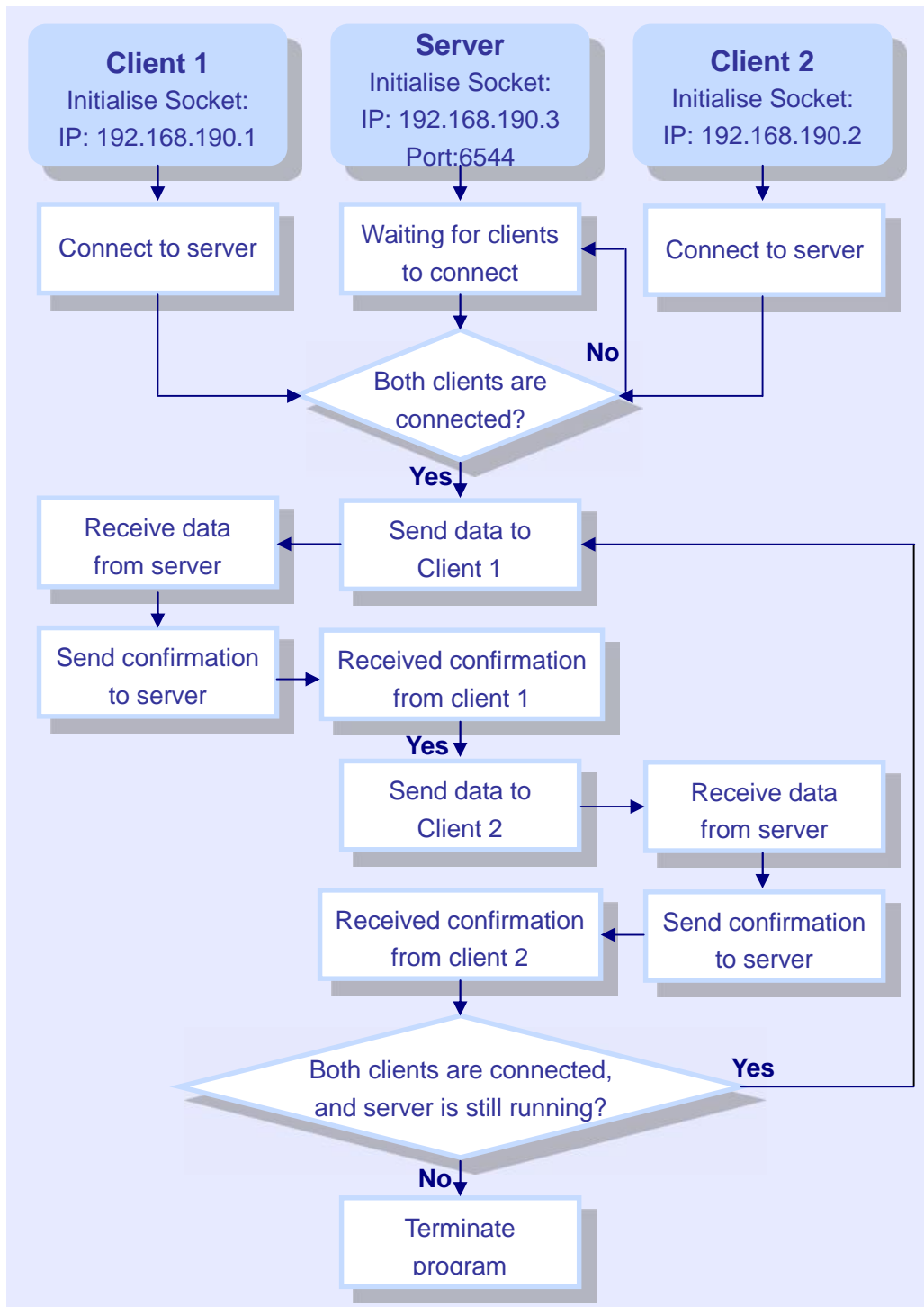


Figure 5.3 Flowchart of Communication

The communication of the system is based on the TCP/IP protocol, and the data is transmitted through a 1G Ethernet cable linking the server and two client PCs. The general data transmission procedure, IP addresses and port settings are shown in Figure 5.3. After the program is executed, the server PC waits for the two client PCs to be connected. Once connected, the server PC transmits the tracking data, ball's position and game status to Client 1 and Client 2 in turn, and waits for a confirmation message from clients at each transmission procedure. If the server and both clients are active, the new tracking data and new ball position are updated frame by frame, and the updated data are transmitted from the server PC to each client PC. The program will terminate if either the server program or the client program is closed by the user. Ideally, the data set transmitted to the two client PCs should be synchronised and transferred at the same time. In the implemented system with only two clients, the data is transferred in turn, and there are no noticeable differences in the rendering results and response time between the two clients.

5.3 Coordinate Systems of Two-Player Game Environment

For a VR environment that enables multiuser interacting with each other, it is important to transform the tracking data to each player's local viewing coordinate frame. In this case, the positions and orientations of the two tracked racquets and the positions of the two players' viewpoints need to be transformed accordingly. Figure 5.4 (top view) illustrates the transformation relationship between the world coordinate system (defined in Section 4.3.2) and each player's viewing coordinate system.

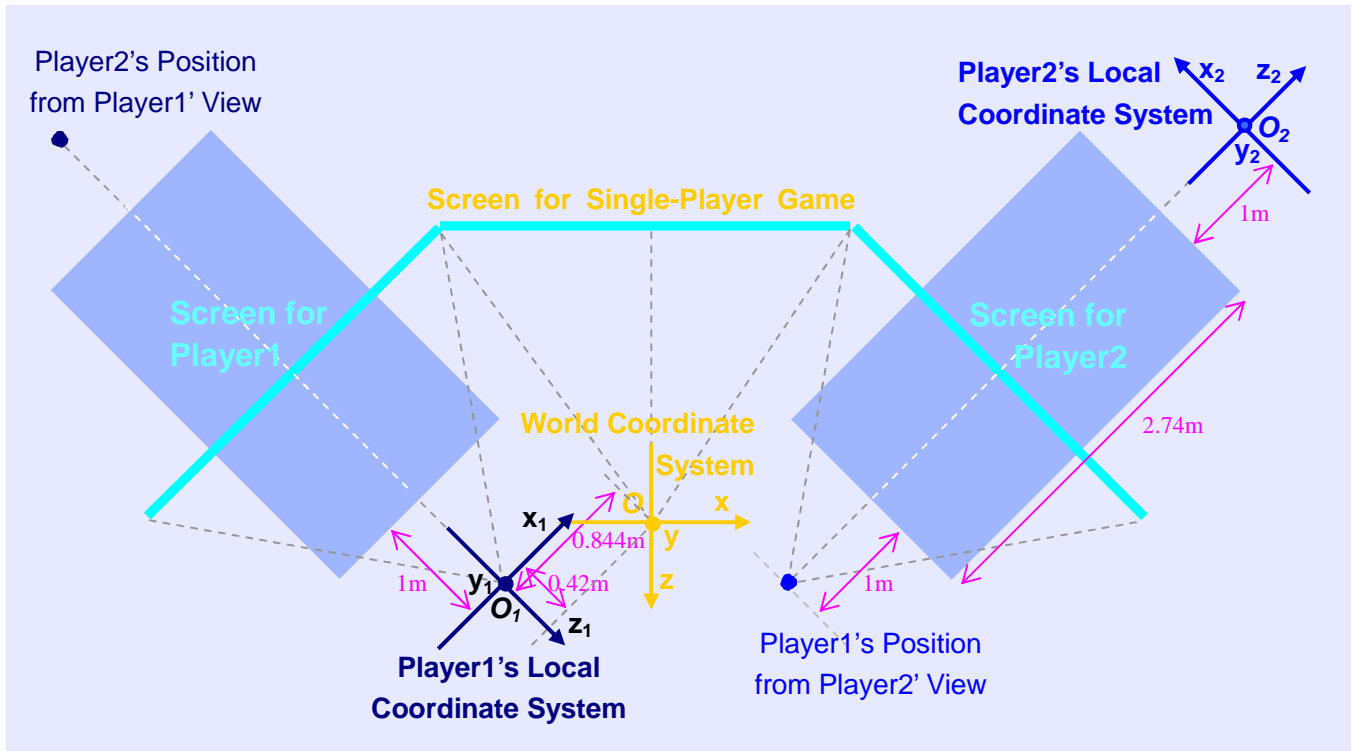


Figure 5.4 Coordinate Systems of Two-Player Game

In order to enable two players to see the same scene with opposite viewing directions, as illustrated in the figure, the defined origin of player1's local coordinate system is at O_1 , which is also the defined coordinate system used for OpenGL (at the world coordinate system for the single-player game). Since the OpenGL origin for rendering the player2's screen is at also at player1's origin position, the defined origin of player2's local coordinate system should be moved to O_2 .

The positions of the racquets of player1 $\vec{p}_1 = [tx_1, ty_1, tz_1, 1]^T$ and player2 $\vec{p}_2 = [tx_2, ty_2, tz_2, 1]^T$ in their corresponding local coordinate systems can be calculated as

$$\vec{p}_1 = T_1 \vec{p}_{w1} \quad (5.1)$$

$$\vec{p}_2 = T_2 \vec{p}_{w2} \quad (5.2)$$

where \vec{p}_{w1} and \vec{p}_{w2} are the two tracked racquets' positions in the world coordinate system. The transformation matrices T_1 and T_2 are expressed as

$$T_1 = \begin{bmatrix} \cos 45^\circ & 0 & -\sin 45^\circ & 0.844 \\ 0 & 1 & 0 & 0 \\ \sin 45^\circ & 0 & \cos 45^\circ & 0.420 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3)$$

$$T_2 = \begin{bmatrix} \cos 135^\circ & 0 & -\sin 135^\circ & 0.844 \\ 0 & 1 & 0 & 0 \\ \sin 135^\circ & 0 & \cos 135^\circ & -0.420 - D \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.4)$$

In T_2 , D is the distance between the origins of the two players' local coordinate system, which equals twice the player-to-table distance (i.e. 1m) plus the table length (i.e. 2.74m), giving $D=4.74$.

Similar to \vec{p}_1 and \vec{p}_2 , the new translations of the players' local viewpoints \vec{p}_{eye1} and \vec{p}_{eye2} can be calculated by

$$\vec{p}_{eye1} = T_1 \vec{p}_{eyew1} \quad (5.5)$$

$$\vec{p}_{eye2} = T_2 \vec{p}_{eyew2} \quad (5.6)$$

where \vec{p}_{eyew1} and \vec{p}_{eyew2} are the tracked viewpoints in world coordinates.

By using Equation 4.1 in Chapter 4, the vertex on the 3D virtual racquet in the world coordinate system can be obtained, and denoted by $\vec{v}' = [x_w, y_w, z_w, 1]^T$. Then \vec{v}' can be further transformed to each player's local viewing coordinate by

$$\vec{v}_1 = T^{wv1} \vec{v}' \quad (5.7)$$

$$\vec{v}_2 = T^{wv2} \vec{v}' \quad (5.8)$$

$$T^{wv1} = \begin{bmatrix} \cos 45^\circ & 0 & -\sin 45^\circ & tx_1 \\ 0 & 1 & 0 & ty_1 \\ \sin 45^\circ & 0 & \cos 45^\circ & tz_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.9)$$

$$T^{wv2} = \begin{bmatrix} \cos 135^\circ & 0 & -\sin 135^\circ & tx_2 \\ 0 & 1 & 0 & ty_2 \\ \sin 135^\circ & 0 & \cos 135^\circ & tz_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.10)$$

where $tx_1, ty_1, tz_1, tx_2, ty_2, tz_2$ are the new translation parameters, which are calculated by Equations 5.1 and 5.2.

In summary, the new viewing positions for player 1 and 2 at their local coordinate systems can be obtained by using Equations 5.5 and 5.6, respectively. Additionally, the racquets of the two players can be drawn at their local coordinate systems using Equations 5.7 and 5.8. Other objects (i.e. table, ball, net and 3D avatar) are drawn in exactly the same way as described in Chapter 4 without requiring coordinate transformations.

5.4 3D Opponent Display

Chapter 3.3.2 described the working principle and the output format of the SR camera. This section, the method of integrating the SR camera to the designed VR environment is presented. As shown in Figure 5.6, the SR camera is mounted on top of the screen, and it is looking down at a 20-degree angle with respect to the horizontal axis. The capture volume is measured to be about 0.8m to 2m in height, and $\pm 0.8m$ to the left and right. Hence the generated 3D surface is distorted by the camera viewing angle, which can be corrected by rotating the output 3D points in the reverse direction by the same amount.

If the 3D coordinate of each point recorded by the SR camera is defined as $P_{sr} = [x_{sr}, y_{sr}, z_{sr}]^T$, the new 3D points $P'_{sr} = [x'_{sr}, y'_{sr}, z'_{sr}]^T$ after view correction can be calculated by

$$P'_{sr} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 20^\circ & -\sin 20^\circ \\ 0 & \sin 20^\circ & \cos 20^\circ \end{bmatrix} P_{sr} \quad (5.11)$$

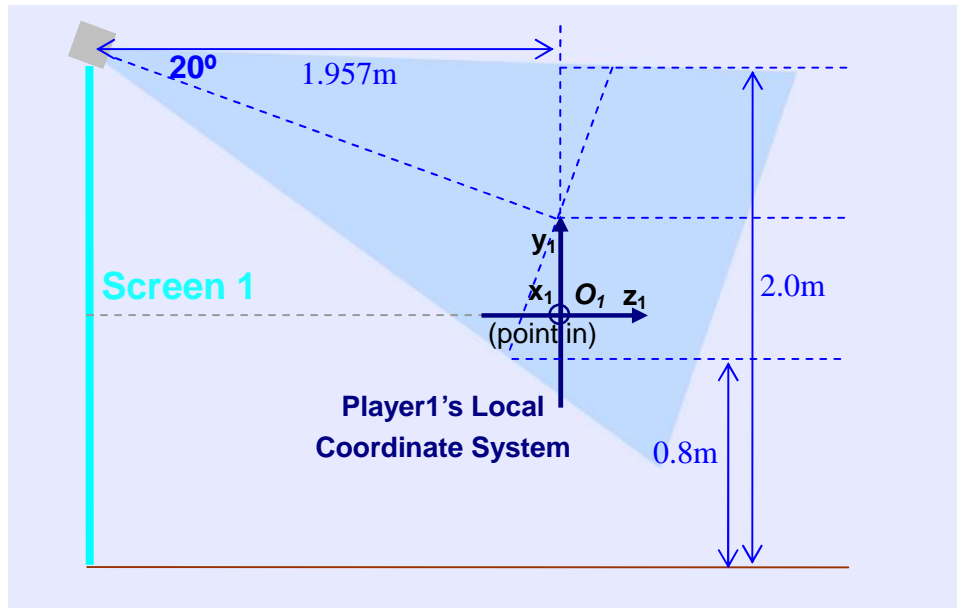


Figure 5.6 3D Camera Positioning

By treating the coordinates of calculated spatial points P'_{sr} as vertices, a 3D surface of the measured scene with objects can be created by triangles in each frame. Figure 5.7 illustrates the way of building a triangular mesh based on the output data. As shown in Figure 5.7 (a), the pixel index is in a range from 0 to 25343, and the pixel indexes of the four neighbouring pixels can be denoted by n , $n+1$, $n+176$, and $n+176+1$, respectively. As illustrated in Figure 5.7 (b), each pixel position is used as a vertex. Therefore, the total number of triangles generated in each frame is given by $2(176-1)(144-1) = 50050$.

There was a visible delay when these 50,050 triangles were rendered frame by frame. In order to reduce the rendering time, the vertices were down-sampled by a factor of 2,

and only $(176/2)(144/2) = 6336$ 3D points were used to generate the surface. Therefore, the pixel indexes of the four neighbouring pixels were changed to n , $n+2$, $n+352$, and $n+352+1$. The total number of triangles was $2(88-1)(72-1) = 12354$. With the displayed opponent standing about 5m away from the player in the virtual space, there was no obvious change of the observed image quality.

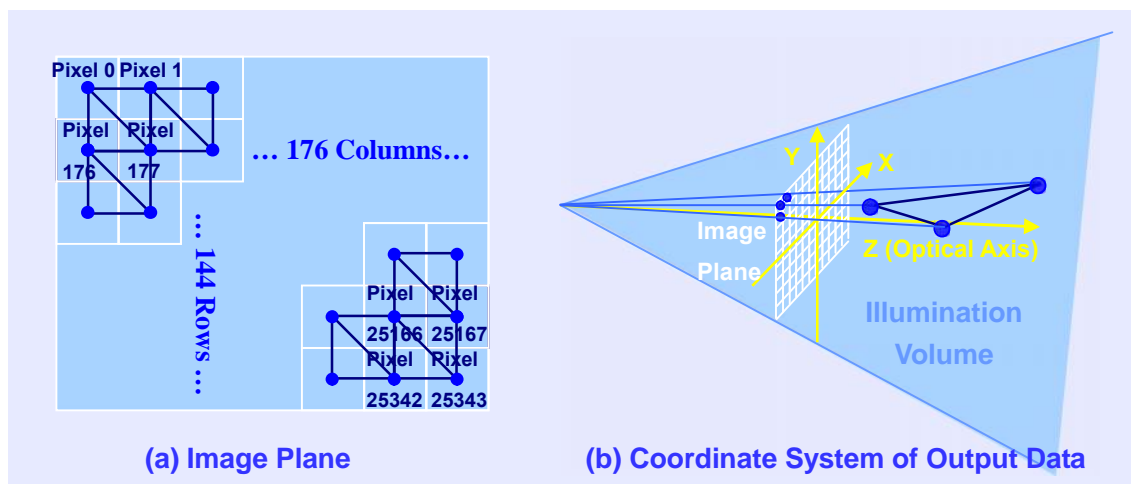


Figure 5.7 Generating 3D Surface by Triangles

As discussed in Chapter 3, the tracking stability is sensitive to the infrared reflection characteristics of the objects and their distances in the illumination volume, incorrect spatial points appear as noise. Especially for the points outside the camera's illumination volume, their acquired 3D coordinates may be replaced by some extreme values due to a back-folding phenomenon. Although a 3×3 median filter had already been employed to present a smoother 3D surface, the noise with extreme values may badly destroy the displayed opponent's appearance.

Since all 12354 triangles were rendered, all the captured objects were represented as one surface. With a 20 degrees rotation, the background that is far away from the player is also connected with the foreground player surfaces. The background is eliminated by a threshold set to 1m experimentally, thereby removing points further than 1m. Additionally, a simple noise removal algorithm was used in this project.

During the noise removal procedure, the depth difference between 4 neighbouring 3D points was checked. If the difference was greater than a pre-set threshold (set to 15cm), then the corresponding triangle was not displayed. The 3D surfaces before and after noise removal procedures are shown in Figure 5.8.

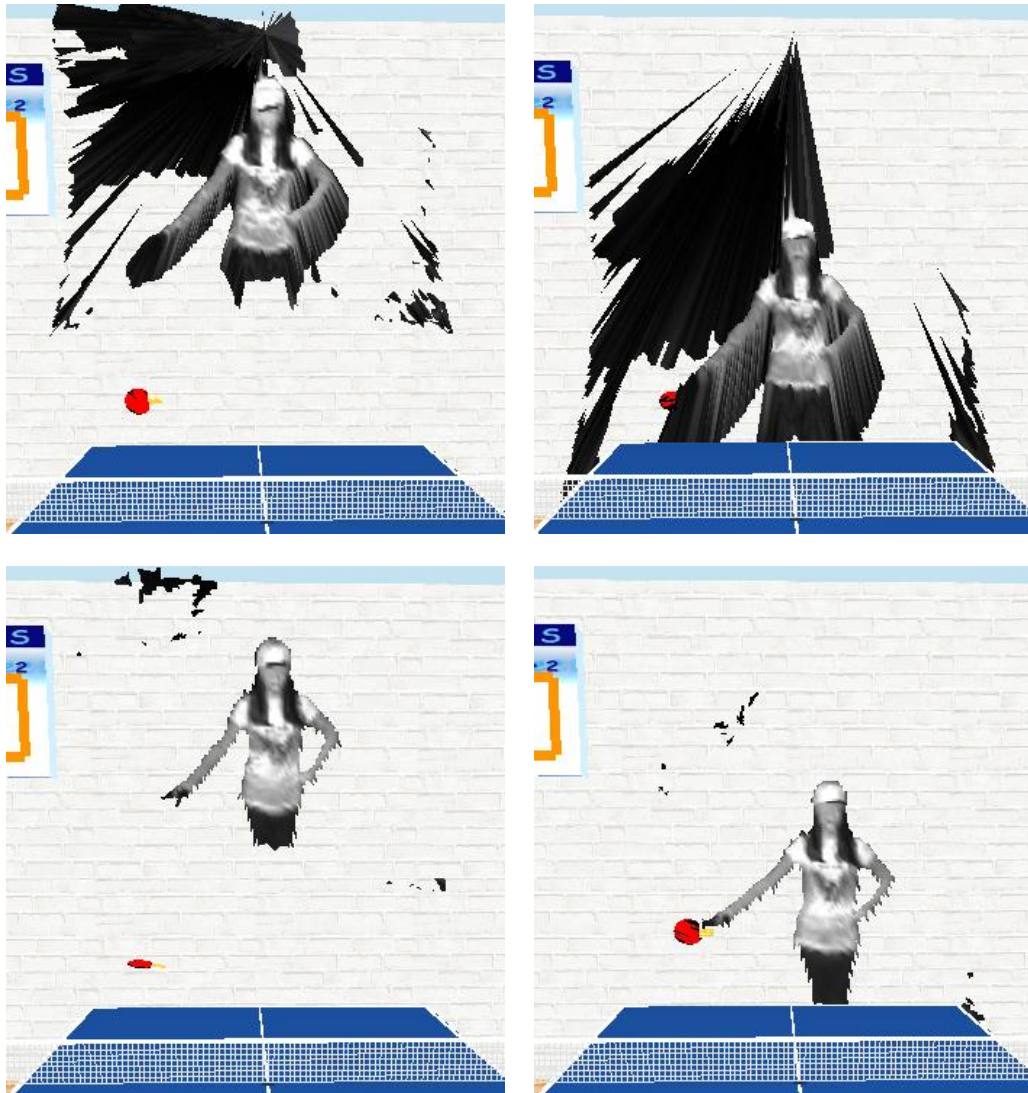


Figure 5.8 Opponent's 3D Surfaces with (right) and without (left) rotation, with (bottom) and without (top) noise reduction

5.5 Game Strategy and Results

When program is executed, the game scene and four virtual buttons (menu) appear, and a virtual stick is provided for menu selection by player1. The position of the stick is based on hand-tracking. If a button is touched by the virtual stick, and the main key on the hand tracker is pressed at the same time, game enters the corresponding mode.

Figure 5.9 shows a simple 4-page tutorial designed for the fresh system users. They are displayed in the scene by texture mapping. To go to the next page, player needs to click virtual buttons in the scene.

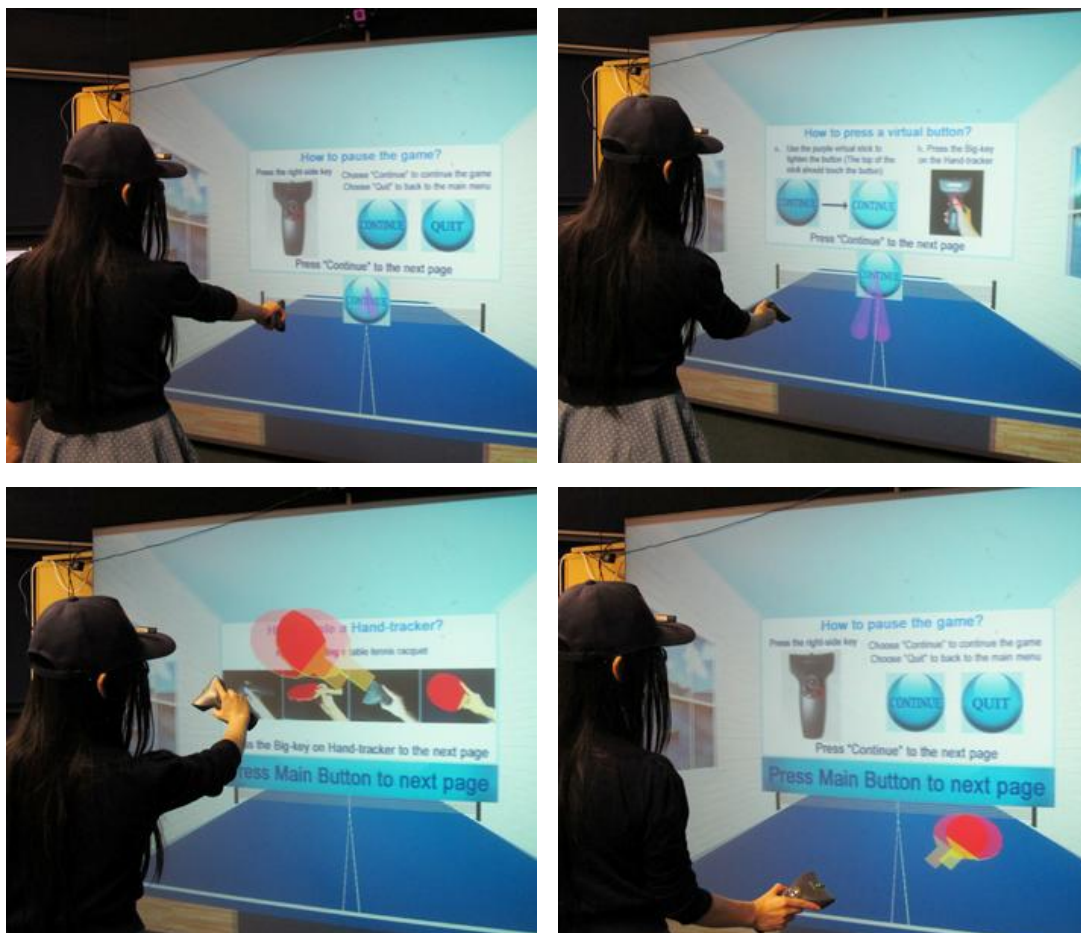


Figure 5.9 Player is Reading a 4-pages Tutorials

The game for single player is enabled if the player chooses the “Play” button, and the computer serve balls to the player from different positions when different keys on the tracker are pressed. Two players can play at the same time, but not the same game. Therefore, the calculation of the ball’s trajectory in this game mode is performed by the clients. The game can pause or exit during playing, if both players exit, the game is back to the main menu.

For the two-player competitive game, the basic rules of the game are similar to a common table tennis game. Two players take turns to serve by pressing keys on their hand tracker, and the first player to reach 11 points is the winner. If any player presses the key to pauses, the game pauses. If any player presses the key to exit, the game is back to the main menu. The pictures of two players playing against each other are shown in Figures 5.10 to 5.12.

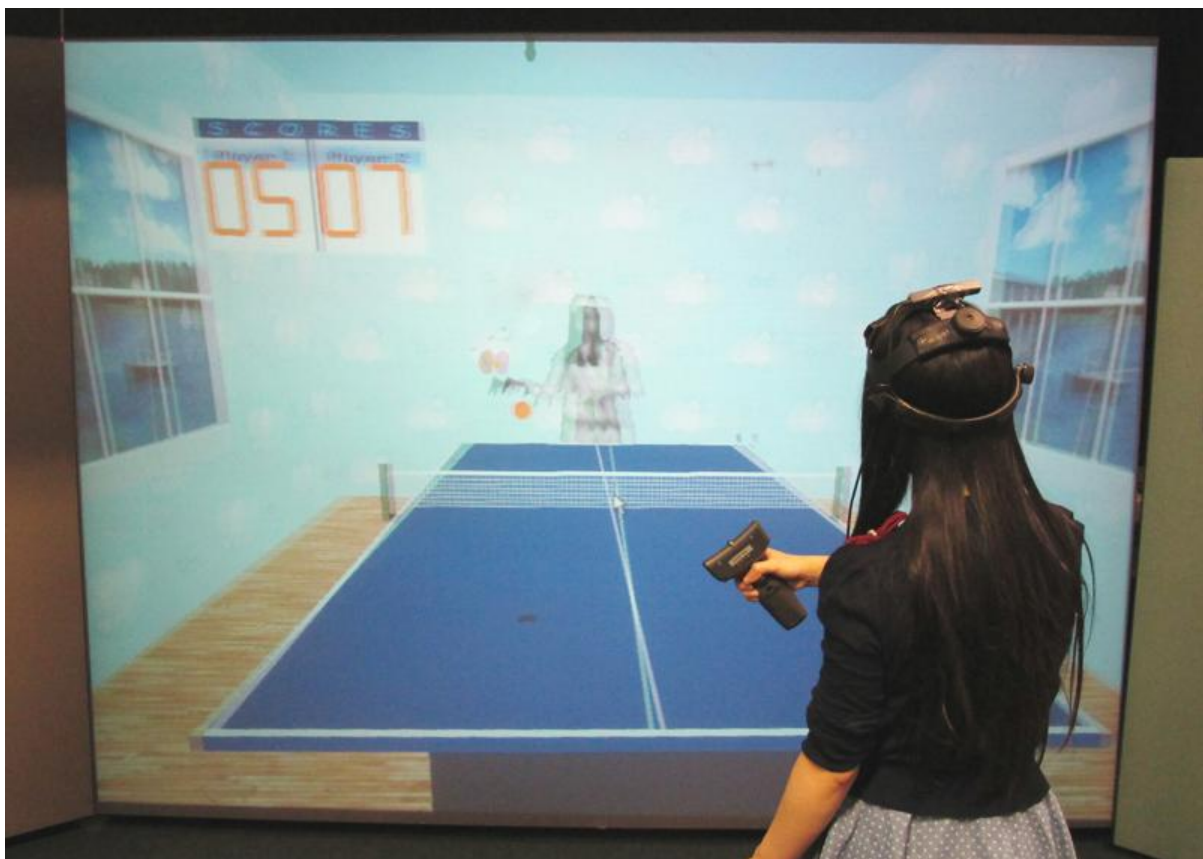


Figure 5.10 Player Plays with her Opponent

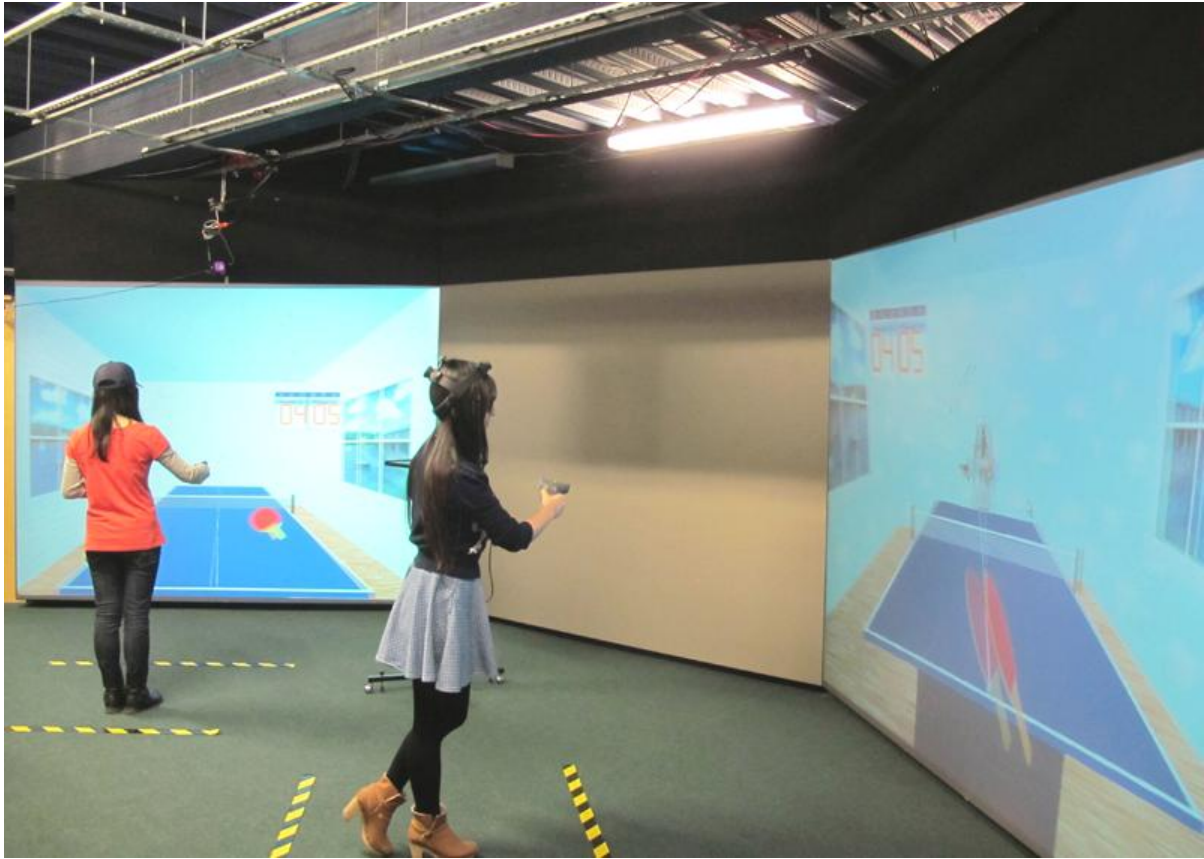


Figure 5.11 Two Players Play against Each Other

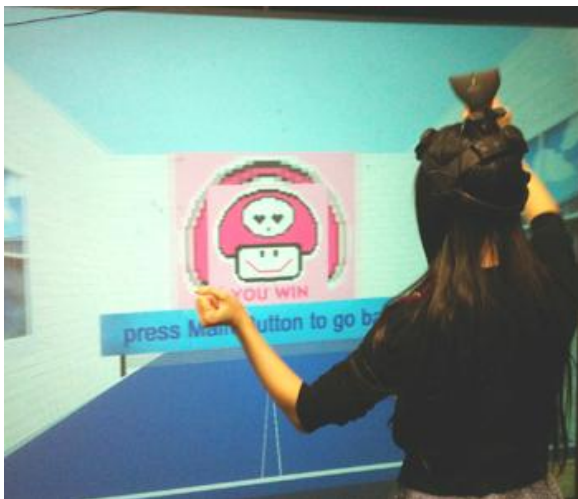


Figure 5.12 Win and Lose in the Game

5.6 Conclusion

This chapter described the elements required and implemented to extend the system from a single-player game to a competitive game environment for two players. The overall system structure, communication strategy, and the coordinate transformation from world coordinates to players' local viewing coordinates are presented in details. Additionally, the use of SR 3D camera enables a 3D opponent to be displayed in real-time. Based on some results presented in this chapter, the overall system is seen to provide an immersive and enjoyable game environment, and it could be used as a complete solution for table tennis game training and competition. The source code of the program and a video of two players playing this game are included in a CD handed in with this thesis.

Chapter 6

User-based Evaluation

6.1 Introduction

In order to assess the performance of the VR system implemented to provide an immersive and interactive table tennis game environment, a user-based evaluation was conducted. Since there is no standardised measurement available for VR applications, this chapter begins with a literature review of current VR evaluation methods. In order to organise the issues addressed by previous literatures, a framework and heuristics of VR evaluation were developed. Based on the heuristics, questionnaires were developed as well and employed to investigate participants' perspective of the system usage. Both qualitative and quantitative data were obtained from the user-based test. By using statistical analysis, both the technology achievement and the degree of presence provided by the system were evaluated, and the heuristics-based questionnaires were proved to be valid and reliable. In addition, the advantages of applying stereoscopic display, head-tracking, and opponent visualisation technologies had been proved by statistical significance.

6.2 Evaluation Methodology Development

6.2.1 Background Investigation

To evaluate a new computer simulation system in terms of its performance and functionality, it is crucial to analyse its usability systematically (Sutcliffe and Kaur, 2000). According to the ISO standard 9241-11, usability describes “the extent to which a product can be used by specific users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” (Cramer, 2004; Stewart and Travis, 2003). In the context of computing systems with traditional HCI, usability usually refers to “how easily and how effectively the computer can be used by a specific set of users” (Kalawsky, 1999).

Usability testing began in the early 1980s, and it has been an established evaluation method in the HCI area from 1990s (Dumas, 2003). Therefore, there is a rich source of information on usability standards, checklist and guidelines/heuristics to provide structured ways to guide technology design (Stewart and Travis, 2003). In a typical traditional usability assessment, experts compare system elements to a set of given guidelines based on specific design requirements.

With a VR system being a specialised HCI, there are many challenges when using the conventional evaluation methods in the VR area (Cramer, 2004; Stanney and Cohn, 2003; Chapman and Stone, 2010). First of all, due to VR involving a large design space with wide adoption of varied innovative interface devices and different types of physiological interaction methods, it is difficult to develop a standard or generalised guideline to cover all the variations. Furthermore, there is a key difference between VR and the familiar computer paradigm: participants have a perception of “presence” when they are experiencing VR. As described in Chapter 2, “presence” mainly refers

to a sensation of “being there” and an illusion of “non-mediation”. Obviously, the well established evaluation methods for traditional HCI assess neither the extent of user immersion nor the degree of intuitive interaction.

There are quite a number of literatures discussing the concept, category and effect of presence. Numerous researches have demonstrated that analysing the degree of presence is “an extremely important stage in evaluating a VR system from a user’s perspective” (Kalawsky, 1999), and has become one of the main usability criteria of VR (Sylaiou, et al., 2010; Stanney, 2002).

Although there is no standardised measurement currently available for varied VR applications (Alencar, et al., 2011), it is observed from research literature that empirical user-based studies were widely adopted for assessment of varied VR applications (Santos, et al., 2009). Both quantitative and qualitative data can be acquired through a user-based study. Quantitative data are usually collected from task-based tests, such as recording time spending in navigation tests (Van Kapri et al., 2011), recording scores in game environments (Finkelstein, et al., 2011), and testing culture understanding in archaeology exploring (Champion, et al., 2011). With respect to qualitative data, the main way of acquiring and analysing self-reported information is through questionnaire-based tests (Ijsselsteijn, et al., 2000).

On the other hand, the traditional guideline-based heuristic evaluation has been inherited and extended to VR area (Alencar, et al., 2011; Chertoff, et al., 2010). Heuristics refer to experience-based techniques for problem solving, learning, and discovery, which is “extremely useful as it has the potential to identify many major and minor usability problems” (Bowman, et al., 2002). A set of heuristics not only supports expert-based evaluations, but also provides a solid structure of the questionnaire for user-based experiments.

6.2.2 Development of Evaluation Heuristics

6.2.2.1 Existing Approaches

Many literatures focused on assessing the overall usability of a VR system by employing a set of heuristics for both expert-based and user-based assessments. Some early researches attempted to develop a universal method to cover all kinds of applications, which usually adopted a large number of questions/suggestions to form a series of heuristics. For instance, Gabbard's Taxonomy and VRUSE are typical universal heuristics, which are described in Table 6.1. In the case of practical evaluation, VR developers tend to use a selective or simplified version of existing heuristics based on their particular applications (e.g. a modified version of Gabbard's Taxonomy developed by Alencar, et al., 2011).

Name	Number of Items	Classification
Gabbard's Taxonomy (Gabbard, 1997)	195	1.VE users and user tasks; 2.VE user interface input mechanisms; 3.Virtual model; 4.VE user interface presentation components (there were sub-categories under these 4 aspects)
VRUSE (Kalawsky, 1999)	Around 100	1.Functionality; 2.User input; 3.System output; 4.User guidance and help; 5.Consistency; 6.Flexibility; 7.Simulation fidelity; 8.Error correction and handling and robustness; 9.Sense of immersion and Presence; 10.Overall system usability

Table 6.1 Gabbard's Taxonomy and VRUSE

However, few literatures on the overall usability reported an effective way to measure perceived presence by users, though most of them put presence factor into their heuristics. Despite there were some questions as "Overall I would rate my sense of presence as: ..." (Kalawsky, 1999), they may confuse unprofessional participants due to the use of unfamiliar terminologies or constructs (e.g. presence) (Lessiter, et al.,

2001).

On the other hand, there was a growing interest in measuring presence in user-centered studies to inquire participants' perspective and collect feedbacks. Since "a measure that takes account of the potential multidimensional structure of presence may prove to be more robust." (Lessiter, et al., 2001), the dimensions of presence were often discussed and categorised in literatures. Table 6.2 summarises three widely utilised questionnaires with their inner classifications.

Name	Number of Items	Classification
PQ (Presence Questionnaire) (Witmer and Singer, 1998)	19	1.Control factor; 2.Sensory factor; 3.Distracton factor; 4.Realism factor Cluster analysis: 1.Involvement/control; 2.Natural; 3.Interface quality
SUS (Slater-Usoh-Steed) (Slater, et al.1994)	6	1.Laws of physics; 2.Visual cliff; 3.Virtual actors responding to the subject; 4.Subjective factors
ITC-SOPI (ITC-Sense of Presence Inventory) (Lessiter, et al., 2001)	44	1.Sense of physical space; 2.Engagement; 3.Ecological validity; 4.Negative effects

Table 6.2 Introduction of PQ, SUS and ITC-SOPI

Although the major aim of these studies was assessing presence, quite a number of questions also contributed to usability aspects, such as Interface Quality in PQ and Laws of Physics in SUS. Therefore, some overall usability evaluation questionnaires were derived from presence questionnaires directly, such as VET (Virtual Experience Test) questionnaire (Chertoff, et al., 2010) based on ITC-SOPI. Nevertheless, since these studies focused on exploring the experience itself, rather than the link between perceived presence and technology, none of them provided a clear index for engineers to discover, identify and summarise the weakness of a particular system element from user-reported problems.

Some issues, such as simulator sickness and individual characteristic of users, are

often discussed as important factors that affect system usage. The questionnaires, such as SSQ (Simulator Sickness Questionnaire) (Kennedy, et al., 1993) and ITQ (Immersion Tendency Questionnaire) (Witmer and Singer, 1998) have been developed for assessing potential simulation symptoms and individual differences, respectively. There is a broad agreement that these issues, presence and overall usability are highly related, all factors are acting on and dependent with each other. However few literatures demonstrate a clear relationship among them.

Due to the current lack of a “top view” evaluation instrument with questionnaires covering all above issues, VR developers tend to employ several questionnaires including aspects of usability, presence, simulation symptom and individual characteristics in one user-based test. This results in usability questions being repeated in the presence questionnaire. On the other hand, it is not always necessary to measure individual differences in the case of system assessment, unless there is a need of distinguishing target users or comparing user-performances with their previous experiences.

6.2.2.2 Development of Framework and Heuristics

To address the above problems, a framework demonstrating the relationship among factors associated with usability and presence was developed and is illustrated in Figure 6.1. The arrows indicate influence and all elements inside the orange square are the content of an overall usability evaluation.

According to the framework, an overall usability evaluation consists of four major considerations: Technological Achievement, Sense of Presence, Simulator Sickness, and Effectiveness of Application.

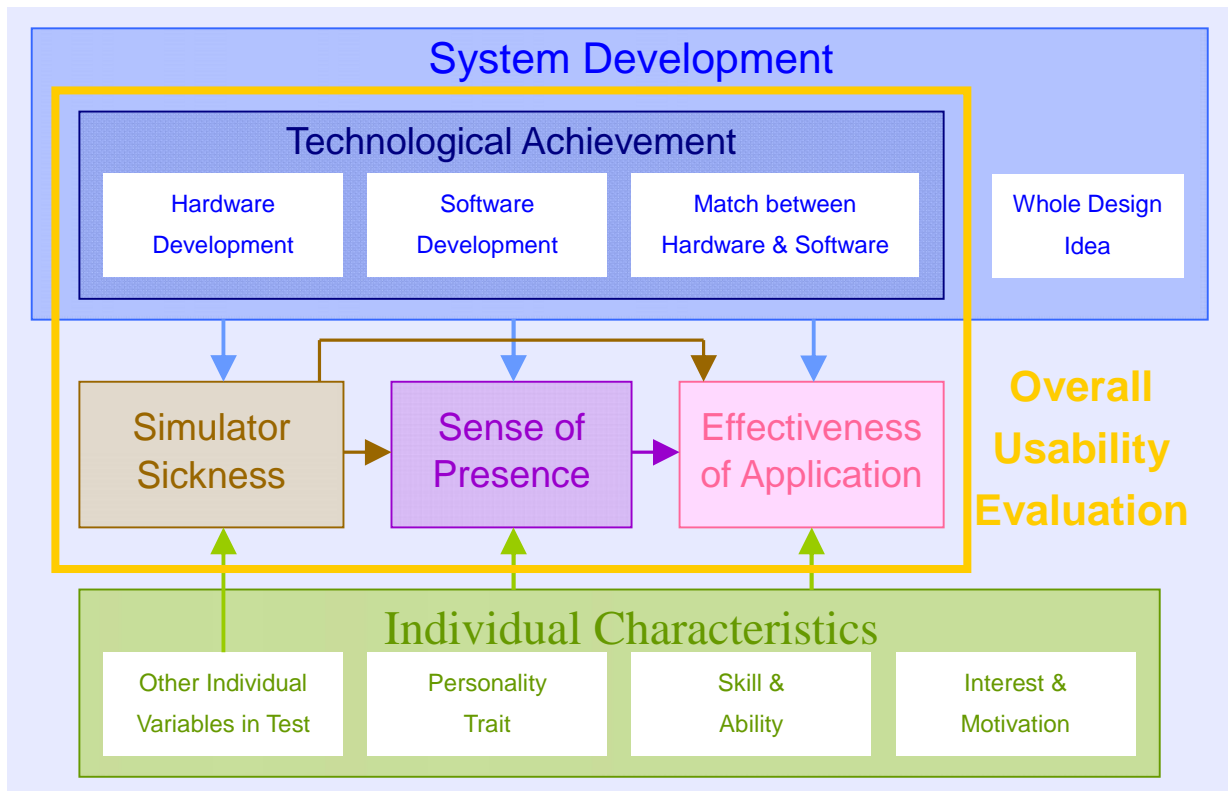


Figure 6.1 Framework of an Overall Usability Evaluation for VR Applications

Technological Achievement

Examining the technological achievement of an invented system is considered as one of the most basic requirements in usability tests, and is divided into three factors, namely, “Hardware Development”, “Software Development” and “Match between Hardware and Software”, to cover all technical aspects supplied by the system itself and to show an overall capability of the technology. Table 6.3 indicates the sub-categories of the three factors, and introduces some detailed considerations, which form a heuristic list for analysing each system element.

The sub-elements of “Spatial Consistency” and “Temporal Consistency” in the factor of “Match between Hardware and Software” are related to participants experiencing “spatial” and “temporal” problems from a phenomenon.

Technological Achievement	Sub-elements	Detailed Considerations / Descriptions
Hardware Development	Input devices	Portability / No interference / Degree of freedom / Stability / Sensitivity / Easy to use / Easy to learn ...
	Output devices	Visual: Display quality / Update rate / Field of view ... Quality of other feedbacks: Audio, Haptic ... Portability / No interference (especially for HMD)
Software Development	Scene with objects	Fidelity / Aesthetics / Richness ...
	Avatars / Agents	Fidelity / Aesthetics / Vividness ...
	User Menu / Indication	Easy to follow / Aesthetics ...
	Laws of Physics	Inertia / Gravity / Collision reaction ...
	Other theme-related factors	Storyline (for walkthrough) / Rules (for training)...
	Stereoscopic effect	Stereoscopic rendering
	Quality of Other Techniques	Stereo sound / Image processing ...
Match between Hardware and Software	Spatial Consistency between user's action and output effect	Correct location / Correct positioning / Correct orientation ...
	Temporal Consistency between user's action and output effect	Caused by: Tracking latency / Rendering speed / Communication speed / Synchronization / ...
	Spatial Consistency & Temporal Consistency	Synthesized effects for: Navigation / Operation / Other tracked objects or characters ...
	Consistency of Multimodal Information	i.e. Consistency among visual, audio and haptic feedbacks.

Table 6.3 Heuristics for Technological Achievement

As shown in the framework, the “Whole Design Idea” in the “System Development” is relevant to whether the hardware and software provide good supports to the specified application, or whether the application is suitable for target users. This element can not be evaluated from a usability test directly, but it can be explored through test results.

Sense of Presence

As mentioned in the previous section, usability is a multidimensional description and depends on system requirements. Therefore, measuring presence is a part of usability evaluation if creating presence experience is one of the major goals that the system aims to achieve. Obviously, users are expected to experiencing presence in VR.

Many sub-factors belonging to “Technological Achievement” are highly associated with the “Sense of Presence”, such as portability of hardware devices and fidelity of software content, which have been identified as presence factors in many previous literatures. However, from the view of this thesis, they are the essential components that contribute to presence, but do not reflect the participants’ perception that should be derived from a synthesized effect represented by the whole system.

The heuristics in Table 6.4 summarise some issues that reflect participants’ perceived presence from various insights. These issues are derived and modified from a summary of current literatures, which can be categorised to five aspects described as follows.

1. Existent: The sense of being located in a virtual world, and co-located with virtual objects.
2. Isolation: The sense of being isolated from real world for both physical and mental aspects.
3. Manipulation: The sense of natural interaction with virtual world.
4. Engagement: The sense of being engaged.
5. Sociality: The sense of being together with other characters.

Although “Sense of Presence” does not refer to any particular part of system, it helps to analyse the weakness and limitation of the whole system design in terms of delivering presence perception from the view of participants.

Sense of Presence	Issues	Explanation
Existent	Locating in somewhere	Users perceive that they are located in a place rather than viewing images. (Lessiter, et al., 2001; Slater, et al., 1994; IJsselsteijn, et al., 2000)
	Co-locating with objects	Users perceive that they are co-located with a set of objects. (Sylaiou, et al., 2010)
	Movement perception	The perception of users' self-movement, and the attention of objects that move relative to them. (Witmer and Singer, 1998)
Manipulation	Natural Operation	An overall degree of control ability. (Witmer and Singer, 1998)
	Anticipation of events	Users may anticipate or predict what will happen next as reaction of their manipulation. (Witmer and Singer, 1998)
Isolation	Interface unawareness	Users may be unaware of the existing interfaces in the state of deep presence. (Witmer and Singer, 1998)
	Real-world unawareness	Virtual world becomes dominant in user's mind. As a result, users may be unaware of the real world. (Sylaiou, et al., 2010; IJsselsteijn, et al., 2000; Slater, et al., 1994)
Engagement	Motivation	The willingness of performing tasks. Users will be engaged if the experience is meaningful to them. (IJsselsteijn, et al., 2000; Witmer and Singer, 1998; Chertoff, et al., 2010)
	Involvement	Psychological state as consequence of focusing one's energy and attention. (Lessiter, et al., 2001; Witmer and Singer, 1998)
Sociality	Co-locating with other characters	Users perceive that they are sharing the same physical space with other users/agents. (Lessiter, et al., 2001; IJsselsteijn, et al., 2000; Chertoff, et al., 2010)
	Co-experience	Users perceive that they are sharing the same experience (e.g. collaboration or competition) with other users/agents. (Chertoff, et al., 2010)
	Emotion Reaction	Users may have emotion reactions in virtual world, which should be same as their expressions in real world. (Chertoff, et al., 2010; Lessiter, et al., 2001)

Table 6.4 Heuristics for Sense of Presence

As mentioned in the previous paragraph, although individual characteristics affect the degree of perceived presence, the meaning and necessity of examining individual differences are various and depend on the aim of experiments. The four sub-factors in “individual characteristics” shown in Figure 6.1 are described in Table 6.5.

Individual Characteristics	Detailed Considerations / Descriptions
Personality Trait	Involvement tendency / Willingness of suspending disbelief ...
Skill & Ability	Task-related skills / Learning ability / Concentration ability ...
Interest & Motivation	Willingness of performing task
Other Variable Factors in Test	Healthy condition / Fatigue / Mood ...

Table 6.5 Sub-factors in Individual Characteristics

Simulator Sickness

“Simulator Sickness” is a kind of negative symptoms suffered by individual users while they are experiencing VR. It draws participants’ attention away from virtual experience, decreases involvement, and finally reduces the sense of presence (Witmer and Singer, 1998). Serious sickness symptoms badly disturb or even stop users’ performance. These symptoms are not only caused by individual healthy condition, but also affected by the system design, especially associated with naturalness of navigation and 3D display quality. Therefore, simulator sickness is important to be examined in evaluating a system as well as ensuring the experiment safety. To test the degree of simulator sickness, there is a standard questionnaire including sixteen potential symptoms, which has been widely accepted and put into practice.

Effectiveness of Application

“Effectiveness of Application” is especially important for training based applications, which is a synthesized effect influenced by all other factors. Since this part of evaluation is application-based, it has not been considered in detail in this thesis.

6.3 Experiment Design

6.3.1 Aim of Experiment

The aim of this user-based test includes two primary aspects. One is to evaluate the VR system developed for table tennis game, and the other is to explore the availability of the proposed framework and heuristics. The detailed processes are shown as follows.

Exploring the availability of the proposed framework and heuristics

1. Applying the proposed heuristics to practice (i.e. questionnaire development)
2. Proving the validity and reliability of the proposed questionnaires

Evaluating the VR system developed for table tennis game

1. A top-view system assessment from users' perspective. (Qualitative Measurement)
2. In order to assess importance of head-tracking and stereoscopic display (Litwiller, 2011), users' performances are compared in three different display situations (Quantitative Measurement):
 - (a) 2D display without head-tracking (similar as watching conventional TV)
 - (b) 3D display without head-tracking (similar as watching 3D film)
 - (c) 3D display with head-tracking (typical VR display technology)
3. Comparing users' perceptions in two different game situations to assess importance of displaying opponent (Qualitative Measurement):
 - (a) Player is not able to see her/his opponent.
 - (b) Player is able to see her/his opponent.

Quantitative Measurement

A quantitative measurement usually refers to recording task performance as

mentioned in the previous section. In simulation of a table tennis game, the score of each player can be recorded as his/her task performance. Since there are many factors which may affect the performance of a player in a competitive game, such as opponent's skill and strategy, "returning computer-served balls" is a better choice for acquiring quantitative data.

Furthermore the trajectories of both ball and racquet can be recorded by computer automatically in every hit. Although the analysis of these trajectories is not the focus of this thesis, they are very useful data for further kinematic investigation.

Qualitative Measurement

The qualitative measurement aims to acquire subjective feedback from participants, which was carried out with four sets of questionnaire.

1. Pre-Q: measuring individual characteristics (e.g. table tennis level, video game playing frequency, etc.)
2. TQ: measuring technological achievement (developed based on the proposed heuristics)
3. PQ: measuring sense of presence (developed based on the proposed heuristics)
4. SSQ: measuring simulator sickness suffered by individual (developed by Kennedy, et al., 1993)

In a test, SSQ needs to be completed twice, one before and one after the virtual experience. By comparing the difference between the two sets of SSQ scores, the potential simulator sickness caused by this system can be reported.

6.3.2 Test Procedure

A total of 27 unpaid participants (11 females and 16 males) who are mainly

undergraduate and postgraduate students from the University of Central Lancashire were recruited through recruitment posters. The mean age of them was 23.4 years with a standard deviation of 3.0 years. All of them had a good healthy condition and were able to play table tennis unassisted. They fully understood the aim and risk of the study, and wished to participate. The ethical approval was obtained by 4th October, 2007, and the reference number is BECT0607/09.

All facilities were in the Visualisation Laboratory in the Computing and Technology Building in the University of Central Lancashire.

Participants arrived in the Visualisation Lab in groups of two. An investigator (the author) was there to supervise and help participants to complete the experiment. The whole experiment lasted for approximately 45 minutes with four main sections: Test Introduction, Training, First Game (returning computer-served balls) and Second Game (competitive game with another player).

Test Introduction

Upon the arrival of two participants in the laboratory, a basic introduction in terms of the VR environment, survey procedure and aim of the study was given to them firstly. If they agreed to participate in the test, they were asked to sign an Informed Consent Document. In order to avoid injury, participants also completed a Physical Activity Readiness Questionnaire (Par-Q). They would not be allowed to continue the experiment if they put "NO" to any question in the Par-Q, as it means they have risks to do more physical activities. Fortunately, all 27 participants had a good healthy condition, and no one declined to participate.

Training

Before the game session, participants completed their first SSQ. Then two participants took turns to receive a short training session that helped them to become acclimated with the 3D environment and familiar with tracking equipment. First of all, the

investigator helped the participant to wear a head tracker, a hand tracker and a pair of 3D glasses. With the program running, the participant can see a virtual room with a virtual table tennis table and a 3D menu, as well as a virtual pointing stick moving with the hand-tracker. By pointing at and clicking the "TUTORIAL" button in the menu, a 4-page simple tutorial (introduced in Chapter 5) demonstrated the control method of the game. While one participant was in the training, the other participant was filling the Pre-Q.

First Game

Once the training and Pre-Q for both participants had been completed, two participants took turns playing the first game. The task of this game was returning balls served by computer. To reduce the difficulty of play, 15 balls were served according to a fixed order from 3 directions (right, middle and left) with fixed speeds (5.6m/s, 5.2m/s and 5.6m/s) and without any spin. The task performance was recorded using "Stroke" and "Point". "Stroke" was recorded if a player struck the ball with her/his virtual racket, while "Point" was recorded if a player made a good return. In this game, a good return was defined by a returned ball that passed over the net and finally touched the opponent's side of the table.

Before recording, each player had a chance to receive 15 balls as practice. During practice, the participant wore a head-tracker and a pair of 3D glasses, as well as holding a hand-tracker as the racquet. Immediately after practice, the player was required to repeat the game in three different display situations: playing with head-tracker and 3D glasses, playing with 3D glasses but without head-tracker, as well as playing without both head-tracker and 3D glasses. The number of "Stroke" and "Point" in each situation performed by each player was recorded as quantitative data.

Second Game

The second game was a two-player competitive game. The basic rules of the game

were similar as the common table tennis game. However, since there was no hand-tracking for player's free hand in order to perform a throw action, balls were served by computer instead of players, and were controlled by the first receiver in a rally by pressing a button on her/his hand tracker. The “first receiver” was switched after every point, and the first player to reach 11 points was the winner. During the game, two participants faced to two different screens and wore their head tracker, hand tracker and 3D glasses. One of the players clicked the "GAME" button in the menu to start the game. Both players were able to see their opponent's virtual racquet, but only one of them was able to see her/his opponent captured by the 3D camera. Once a game ended, two players swapped their trackers and changed their positions to play again. In this way, both players had experiences of seeing and not seeing opponent.

Following completion of the competitive games, two participants filled out their TQ and PQ. Some questions in PQ required two scores for seeing and not seeing opponent separately, and there were some open-ended questions asking suggestions and general comments at the end. Finally, the second SSQ were completed by both participants.

6.3.3 Questionnaire Development

Seven-point Likert-style scales were employed for rating the content of each questionnaire item, which indicate the intensity of participants' feeling (Witmer and Singer, 1998). In other words, all questions were rated on a scale from 1 to 7, which in “1” represents the lowest level and “7” the highest level. Take one of the TQ items for example, the question and its scale are shown below.

How easily did you learn to use the system?

1. Very hard 2. Hard 3. Slightly hard 4. Neither easy nor hard 5. Slightly easy 6. Easy 7. Very easy
--

Pre-test Questionnaire (Pre-Q)

Three major aspects of individual characteristic including “Personality Trait”, “Interest & Motivation” and “Skill & Ability” have been included in Pre-Q, which were employed to investigate the influence of individual differences. The Pre-Q items and their categories are shown in Table 6.6.

Category	Questionnaire Items	
Personality Trait	Involvement Tendency	Do you ever become so involved in a movie or book that you are not aware of things happening around you?
	Focus Ability	How well do you concentrate on enjoyable activities?
Interest & Motivation		How much does a usual table tennis video game involve you?
		How much does a Wii/PS3/Xbox360 table tennis game involve you?
		How much does table tennis sports involve you?
Skill & Ability		How often do you play computer games? 1. Hardly 2. Occasionally 3. Monthly 4. Weekly 5. Frequently (Many times per week)
		Please choose the most appropriate description about your table tennis skills. 1. I don't know how to play it. (Totally Beginner) 2. I know how to play it, but make mistakes most of time. (Beginner) 3. I can successfully return a low-speed ball most of time. (Intermediate) 4. I can successfully return a high-speed or spinning ball most of time. (Advanced) 5. Professional level (Expert)

Table 6.6 Questionnaire Items in TQ

Two questions related to “Personality Trait” measure the tendency of individuals to be involved in VR. They were chosen from ITQ (Immersion Tendency Questionnaire) developed by Witmer and Singer in 1998, which typically addressed two main aspects of immersion tendency: “Involvement” and “focus”. Since participants’ “Interest & Motivation” of doing this test might be associated with their interest of playing other video games and table tennis sport, three questions asked the extent of individuals can be involved in these activities. The last two questions referred to individuals’ relevant “Skill & Ability” for playing this game. One question asked their frequency of playing computer games and another inquired their self-reported table tennis level (This level was classified approximately for unprofessional players, not as a standard).

Technological Achievement Questionnaire (TQ)

A usability questionnaire is usually established by referencing each question to a specific usability category or sub-category. In this way, the weakness of a particular system element can be discovered by looking into the corresponding questionnaire items. Therefore, the proposed heuristics of “technological achievement” which were subsequently converted to TQ is shown in Table 6.7. The questionnaire items did not cover all aspects of the heuristics, only contents correlated with system features were employed.

Sub-elements / Detailed Considerations		Questionnaire Items
Hardware Development		
Input Devices	Portability	How portable were the tracking equipment?
	No interference	How much did the tracking equipment interfere or distract you from playing?
	Stability	How stable were the tracking?
	Easy to use	How easily did you learn to use the system?
Output Devices	Field of view	How satisfying was the field of view?
	Display quality	How good was the image quality on the screen?
Software Development		
Scene with objects		How well did the environment seem as a table tennis room?
User Menu / Indication		How good were the user menu and the simple tutorial?
Laws of Physics (Fly)		How natural did the ball's flying seem?
Laws of Physics (Collision)		How natural did the ball's bouncing seem?
Stereoscopic effect		How natural did the 3D effect seem?
Quality of other techniques		How well did your opponent's appearance seem?
Match between Hardware and Software (Combined Items)		
Spatial Consistency		How much were the rotation and moving direction of your virtual racquet consistent with your hand-tracker?
Temporal Consistency		How much delay did you experience between your action and expected outcome?
Spatial Consistency & Temporal Consistency	Navigation	How natural did the view seem when you were walking around?
	Operation	How much were the movements of your virtual racquet consistent with your real actions?
	Other tracked objects/characters	How much were your opponent's movements consistent with his/her real actions? (in your feeling)
Consistency of Multimodal Information		How consistent was the information coming from your various senses?

Table 6.7 Questionnaire Items in TQ

Sense of Presence Questionnaire (PQ)

Likewise, following the proposed heuristics of “sense of presence”, PQ was developed and is shown in Table 6.8.

Category	Issues	Questionnaire Items
Existent	Locating in somewhere	To what extent did you feel that you were in a room rather than facing a screen?
	Co-locating with objects	To what extent did you feel that the virtual table and ball were in a same space with you.
	Movement perception	How compelling was your sense of the ball’s flying?
Manipulation	Natural Operation	To what extent did you feel that you can interact with virtual environment naturally?
	Anticipation of events	Were you able to anticipate what would happen next in response to the actions that you performed?
Isolation	Interface unawareness	How well did you concentrate on the playing rather than on the devices used to perform activities?
	Real-world unawareness	How aware were you of events occurring in real-world around you?
Engagement	Motivation	To what degree did you intend to hit the ball during playing?
	Involvement	How much did the game involve you?
Sociality	Co-locating with other characters	To what extent did you feel that another player was being with you in a same space?
	Co-experience	To what extent did you feel that another player was playing with you?
	Emotion Reaction	To what extent did you have emotion reaction that was same as your emotion in a real table tennis game?

Table 6.8 Questionnaire Items in PQ

All questions correlated with “Isolation”, “Engagement” and “Sociality” categories required two scores for seeing and not seeing opponent situations separately. Besides, two additional questions shown below discussed the advantages of “seeing opponent”.

1. To what extent do you agree that “seeing opponent” can help you to judge the ball’s direction?
2. To what extent do you agree that “seeing opponent” can make the game more attractive?

The scale of the two questions was 7-point Likert Scale, in which “1” means “strongly disagree with it” and “7” indicates “strongly agree with it”. Some open-ended questions asked participants’ suggestions and general comments at the end of PQ.

Simulator Sickness Questionnaire (SSQ)

SSQ is a standard questionnaire developed by Kennedy and others in 1993, which measures symptoms of simulator sickness suffered by individuals. There were 16 items included in SSQ totally, which were General discomfort, Fatigue, Headache, Eyestrain, Difficulty focusing, Increased salivation, Sweating, Nausea, Difficulty concentrating, Fullness of head, Blurred vision, Dizzy (eyes open), Dizzy (eyes closed), Vertigo, Stomach awareness and Burping. The scales ranged from 0 to 4, which refer to none, slight, moderate, and severe, respectively.

6.4 Experiment Results

6.4.1 Experiment Data and Statistical Analysis

Both quantitative and qualitative data were obtained from the experiment, which are described in Table 6.9. For the task performances, the three display situations refer to “3D display with head-tracking” (3D&HT), “3D display without head-tracking” (3D) and “2D display without head-tracking” (2D), respectively. The SS(change) scores used to observe the impact of simulation sickness on individuals were calculated by using SSQ(before) scores minus SSQ(after) scores. Therefore, the range of SS(change) scores is from -3 to 3. Before starting all statistical analysis, the scores of all negatively worded questionnaire items had been reversed (i.e. ensuring higher scores to reflect more positive or favorable opinions).

Data Type	Data	Score Range	Aim of Measurement
Quantitative Data	Stroke(3D&HT) scores	0 to 15	Task Performance: Number of Stroke in three different display situations
	Stroke (3D) scores		
	Stroke (2D) scores		
	Point(3D&HT) scores	0 to 15	Task Performance: Number of Point in three different display situations
	Point(3D) scores		
	Point(2D) scores		
Qualitative Data	TQ scores	1 to 7	Technological Achievement
	PQ scores	1 to 7	Sense of Presence
	SSQ(before) scores	0 to 3	Before Test Simulation Sickness
	SSQ(after) scores	0 to 3	After Test Simulation Sickness
	SS(change) scores	-3 to 3	Difference between two sets of SSQs
	Pre-Q scores	Q1-Q5: 1 to 7 Q6: 1 to 4 Q7: 1 to 5	Involvement Tendency Q1; Focus Ability Q2; Interest of Video Game Q3, Wii Game Q4, Table Tennis Sport Q5; Playing Computer Game Frequency Q6; Table Tennis Skill Q7

Table 6.9 Experiment Data

From the view of descriptive statistics, both Stroke and Point scores are interval data in which the numbers go from low to high in equal intervals. The scores of both Q6 and Q7 in Pre-Q are ordinal data, which provides rank order without equal intervals. However, there is a debate on whether the Likert-style rating scale belongs to ordinal data. Strictly speaking, rating scale of questionnaire items are not interval data because that it does not show how much one score is more or less than the other. (e.g. a score of 6 does not present twice the level of a score at 3). However, rating scale clearly provides more numerical information than usual ordinal data (Argyrous, 2005). Therefore, in many existing literatures dealing with questionnaire scores, some measures, such as mean and standard deviation used for interval data were also employed to analyse rating scale. Furthermore, market research companies also do this as a standard procedure when describing survey data. On the other hand, many statistically parametric tests based on the assumptions of interval data type, normal distribution and homogeneity of variance, such as Pearson's *r*, Cronbach's alpha and ANOVA were also employed for questionnaire analyses in many literatures, though most of scores based on the Likert-type rating scale violate normality assumption (e.g.

left skew if most participants give positive marks).

Since the above topics are not the focus of this thesis and some widely used methods were still employed for evaluation, both parametric and non-parametric tests (i.e. with and without the assumptions of interval/ratio data type and normal distribution) were performed. Despite the debates that whether all these analyses had strictly statistical validity, they are powerful for exploring data.

All the statistical analysis was performed on SPSS (Statistical Program for Social Sciences) (Pallant, 2007; Allen and Bennett, 2010). The statistical significance in the following measurement is labeled by p . A p value less than 0.05 is considered a borderline that is a traditional level of an acceptable statistical significance, less than 0.01 indicates a good level, and less than 0.005 or 0.001 refers to very high significance (Clarke and Cooke, 2004). In addition, N , M and SD denote “number of samples”, “mean value” and “standard deviation”, respectively.

6.4.2 Reliable and Valid Measurement

A good measure must be shown to be reliable and valid. Therefore, before exploring these data, the reliability and validity of the questionnaires should be discussed.

6.4.2.1 Reliability

Reliability refers to the consistency or dependability of questionnaire items. A questionnaire is typically comprised of multiple items, which have related contents, but different with each other. To examine the reliability of measure, it is important to

validate that these items contribute to the same general construct.

Cronbach's alpha is one of the most commonly used measures of internal consistency that assesses the extent to which a set of questionnaire items tapping a consistent underlying construct (Allen and Bennett, 2010). The range of Cronbach's alpha is from 0 to 1, and an acceptable consistency usually requires an alpha value higher than 0.7. Any result higher than 0.8 indicates good reliability and a value around 0.9 is ideal. However, since excess consistency may be caused by redundant items, an alpha value greater than 0.95 is usually not desirable.

The computed Cronbach's alpha of 18-item TQ and 12-item PQ are 0.873 and 0.813, respectively, which indicates good internal consistencies of both questionnaires. However, there are three items providing less contribution to the overall PQ construct, which are Anticipation, Devices Unawareness and Emotion Reaction. The Cronbach's alpha of PQ would increase if any of the three are deleted. To judge whether they need to be dropped from PQ, further analysis, such as correlations between them and other factors were computed. The discoveries are shown in Table 6.10.

PQ Items	Statistical Analysis	Possible Reason of Inconsistency
Anticipation	It was correlated with task performance Point(3D&HT) score. ($r=0.463, p<0.05$) ($\rho=0.452, p<0.05$) (both 2-tailed)	It was affected by the difficulty of the task, rather than whether the system provided natural reaction.
Devices Unawareness	It was correlated with Pre-Q item Focus Ability ($r=0.494, p<0.001$) ($\rho=0.419, p<0.05$) (both 2-tailed)	It was affected by the individual's focus ability rather than whether the devices interfered their action.
Emotion Reaction	It has the highest mean score ($M=6.556, SD=0.5$) in all TQ and PQ items indicating very strong emotion	Intense emotion was probably caused by the competition itself rather than the presence provided by the system.

Table 6.10 Analysis of the 3 Low alpha Items

Due to the debate of data type and normality, both Pearson's Product-moment Correlation Coefficient (Pearson's r) for parametric tests and Spearman's Rank Order

Correlations (Spearman's rho) for non-parametric tests were employed to show the strength of data correlations. Both of them have a range from -1 to +1, in which 0 indicates the complete absence of linear relationship, a value greater than 0.3 can be considered as a positive relationship with medium strength, and a value of 0.5 or bigger refers to strong positive relationship. As shown in the above table, the calculation results produced by using both methods are similar.

Due to the above considerations, these three items are not adequate to detect the degree of the system-generated presence in this application. Therefore, they were dropped from PQ and all subsequent analyses of PQ in this thesis are based on the remaining 9 items. Consequently, the resultant Cronbach's alpha of PQ increases to 0.858.

The high Cronbach's alphas demonstrate high internal consistency, and the questionnaires are therefore considered as reliable measures. On the other hand, it also reveals that some questions proposed in TQ to address a single element of the system are actually associated with many other elements. For instance, participants tended to give higher marks for "scene" if they satisfied with "stereo effect", "field of view" and "navigation" (correlations between "scene" and the above items are found with the significant level at $p < 0.05$).

6.4.2.2 Validity

A valid measure should cover both valid content and valid construct, which is able to measure what it is supposed to measure precisely. The content validity involves systematic examination of theoretical and empirical evidences that should cover the universal area of the measured targets, while construct validity refers to the extent to which the practical measures are consistent with the theoretical construct.

Content Validity

The questionnaire design followed the heuristics introduced in the previous sections including both technological and presence issues, and each questionnaire item was directly derived from a sub-factor of the heuristics. Since the heuristics was developed based on existing theoretical and empirical literatures, it provides valid contents.

Construct Validity

The construct validity can be explored by investigating its relationship with other constructs (Pallant, 2007). According to the framework in Figure 6.1, the “Technological Achievement”, “Sense of Presence”, “Simulator Sickness”, and “Individual Characteristics” have relationships with each other. To prove that the experiment results were consistent with the theory, the correlations were computed among individuals’ mean scores of each questionnaire. The meaningful discoveries are summarised in Table 6.11. Based on this table, the calculation results are consistent with the theoretical construct.

Individuals’ Mean Scores	Correlations	Summaries
TQ & PQ	$r=0.767, p<0.001$ $\rho=0.669, p<0.001$	“Technological Achievement” and “Sense of Presence” have strong positive correlation statistically
TQ & SS(change)	$r=-0.598, p<0.001$ $\rho=-0.466, p<0.05$	“Simulator Sickness” was negatively correlated with both “Technological Achievement” and “Sense of Presence” statistically
PQ & SS(change)	$r=-0.661, p<0.001$ $\rho=-0.399, p<0.05$	
PQ & Pre-Q item “Involvement Tendency”	$r=0.413, p<0.05$ $\rho=0.441, p<0.05$	“Individual Characteristics” was correlated with “Sense of Presence”

Table 6.11 Correlations among Questionnaires

6.4.3 Data Exploration for System Evaluation

6.4.3.1 Simulator Sickness

The mean scores of SSQ(before) and SSQ(after) are only 0.030 and 0.047, respectively (scale: 0-none, 1-slight, 2-moderate, and 3-severe), and the mean of SS(change) that indicates increased sickness is 0.017. This result reveals a very minor overall simulator sickness.

In order to compare the mean score differences rated before and after VR experience by same participants, both Paired Samples T-tests (for parametric) and Wilcoxon Signed Rank Test (for non-parametric) were performed for all individual symptom. However, neither of them proved statistical differences at $p < 0.05$ significant level. It is concluded that the assumption of appearing sickness symptoms caused by using this VR system can not be proved statistically. This result implies that the system provides a safe environment with natural representation.

There were 6 in the total of 27 participants suffered increased sickness slightly after the experiment. All of them reported sickness on their first SSQ (before the experiment). These symptoms include Fatigue, Headache, Eyestrain, Difficulty Focusing, Sweating, Difficulty Concentrating and Dizzy (eyes open). It can be observed that the simulator sickness is easy to appear if a participant's brain or eyes feel tired before the test.

6.4.3.2 Technological Achievement

The 18-item TQ had a mean score of 5.932 ($SD=0.495$) (scale: 1-very bad, 2-bad,

3-slightly bad, 4-moderate, 5-slightly good, 6-good, 7-very good). This score refers to a good level of users' satisfaction on system usage. Since each questionnaire item refers to one aspect of the system, Figure 6.2 compares the mean score of each system element from users' perspective by histograms. Since all items got positive feedback ($M > 4$), the scale in the figure is in a range of 4 to 7.

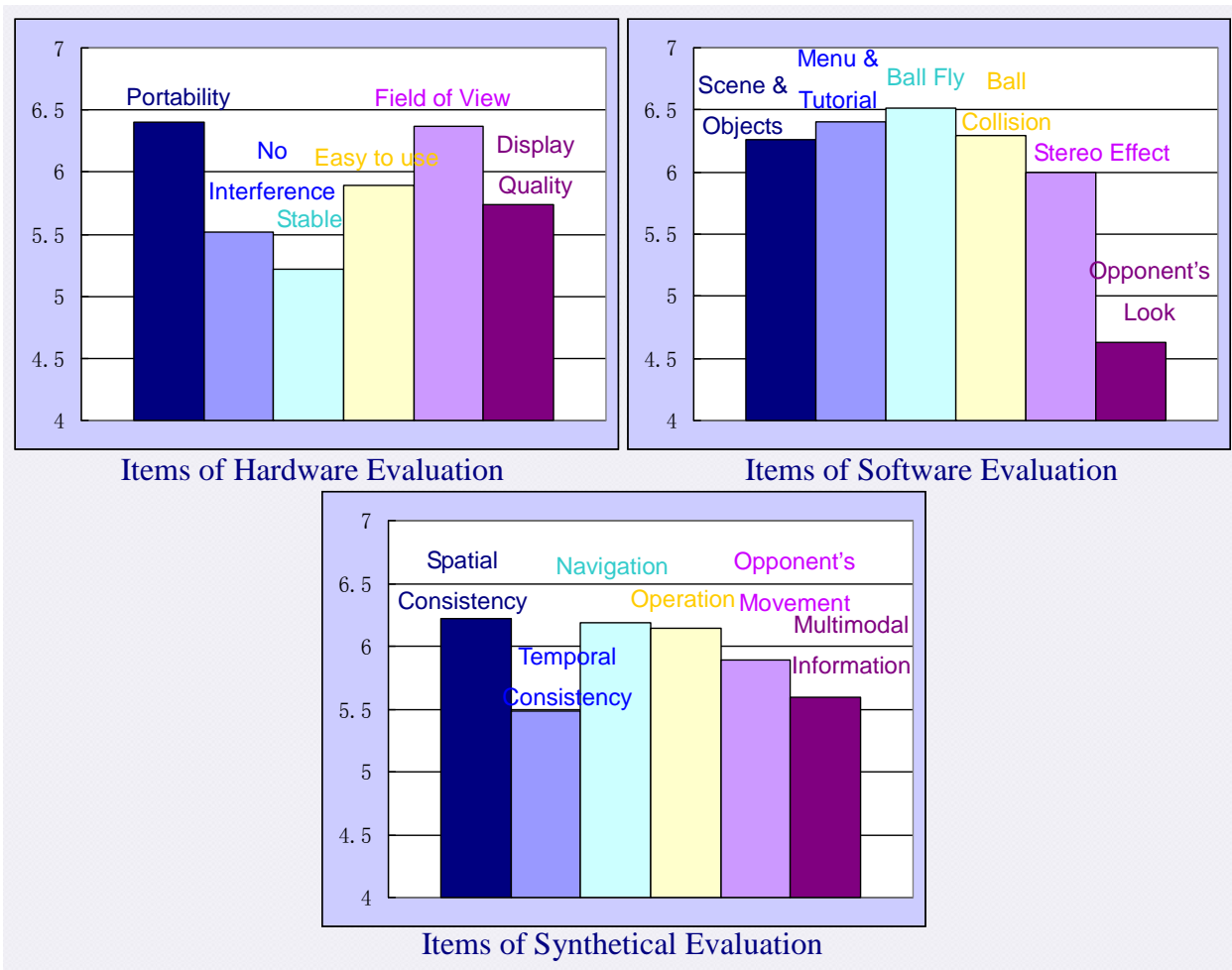


Figure 6.2 Mean Scores of 18-TQ Items

In total 10 items were rated as good or better ($M > 6$). For hardware evaluation, high “Portability” ($M=6.407$) could be explained by light weight and wireless trackers, and great “Field of View” ($M=6.370$) could be explained by the use of a large screen. According to participants' comments, the “Scene and Objects” ($M=6.259$) and “Menu and Tutorial” ($M=6.407$) were “good-looking”, and the physics of “Ball Fly” ($M=6.519$) and “Ball Collision” ($M=6.296$) were “very natural”. This resulted in high marks given to the software development category. In addition, two players

attempted to catch the virtual ball by using their real hand, and one player wanted to touch the virtual table. These phenomena confirm a good “Stereo Effect” ($M=6.000$). For the combined items, good “Spatial Consistency” ($M=6.222$) refers to correct tracking data processing, and the assessment for natural “Navigation” (navigation refers to walking around in this test) ($M=6.185$) and natural “Operation” ($M=6.148$) got good marks as well.

There are 7 items with scores between 5 and 6 (i.e. slightly good to good), in which “Easy to Use” of devices ($M=5.889$), “Display Quality” of screen ($M=5.741$) and whether the “Opponent’s Movement” looks natural ($M=5.889$) are in a relatively high score group. The mean scores of “No Interference” of devices ($M=5.519$), “Temporal Consistence” of the whole system ($M=5.481$) and the consistency of sense from “Multimodal Information” ($M=5.59$) are around 5.5, and “Stability” of the tracking system ($M=5.222$) got a relatively low score. Through an analysis combined with participants’ comments, players who reported experiencing “racquet flies away” during playing (i.e. occasional loss-tracking of the InterSence system) tended to give a low mark on “Stability”. Strongly and statistically significant correlations were found for “Stability” with “No Interference” ($r=0.767$, $p<0.001$) and “Temporal Consistence” ($r=0.604$, $p<0.001$). Therefore, it can be explained that the loss-tracking problem dropped the mean scores of the three items. On the other hand, two players complained that they felt visible delay if they wanted to smash the ball. This delay was caused by the running speed limitation, which affected the mean score of “Temporal Consistence” as well. It is also possible that the lack of haptic feedback decreased the mean score of “Multimodal Information” since players expected to “feel impact on the racquet”.

Only one item has a mean score lower than 5 (i.e. moderate to slightly good), which is “Opponent’s Look” ($M=4.630$) indicating relatively low satisfaction from players’ perspective. Four players commented that they preferred a “colourized opponent”, and one player complained not being able to observe the opponent’s face clearly.

6.4.3.3 Sense of Presence

The total mean score of the 9-item PQ is 6.189 ($SD=0.317$) (scale: 1-very weak, 2-weak, 3-slightly weak, 4-moderate, 5-slightly strong, 6-strong, and 7-very strong), which means participants perceived a strong presence during their VR experience. According to the histograms in Figure 6.3, the mean scores of all five presence issues get high level scores in the range of 6 to 6.5, which includes the sense of “Existent” ($M=6.247$), natural “Manipulation” ($M=6$), “Isolation” ($M=6.185$) from real world, “Engagement” ($M=6.370$) of playing and “Sociality” ($M=6.019$) effects. Most participants expressed their opinions in comments using words such as “exciting”, “amazing” and “impressive”, which explained why “Engagement” got the highest mean score.

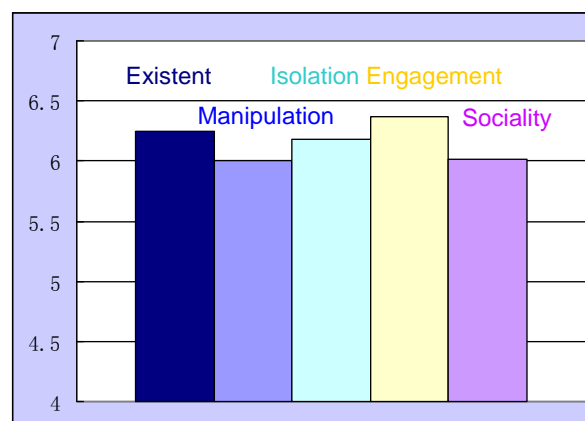


Figure 6.3 Mean Scores of the Five Pretense Issues

To compare the degree of involvement of this game with others, a question belonging to “Involvement” is used to compare with 3 Pre-Q items. These four questions with mean score and standard deviation are shown in Table 6.12. It is seen that this game got a highest mean score with lowest standard deviation.

Questions	Mean Score	Standard Deviation
How much did this game involve you?	$M=6.44$	$SD=0.89$
How much does table tennis sports involve you?	$M=5.67$	$SD=1.41$
How much does a Wii/PS3/Xbox360 table tennis game involve you?	$M=5.11$	$SD=1.60$
How much does a usual table tennis video game involve you?	$M=4.22$	$SD=1.58$

Table 6.12 Four Questions Addressing Involvement

6.4.3.4 Task Performances

There are two kinds of quantitative data collected from the “returning computer-served balls” test to assess task performance: scores of “Stroke” and “Point”. Since the test repeated three times in three different Display Situations (i.e. 2D display, 3D with and without head-tracking), three sets of “Stroke” and “Point” data were obtained.

Since players with different table tennis levels and video game playing frequencies may have different performances of playing this game, the task performance was analysed by combining with the scores of TTL (Table Tennis Level) and GF (video Game playing Frequency) in Pre-Q. Since there was neither professional table tennis players nor totally beginners in participants, the 5-level TTL was changed to 3-level (i.e. beginner, intermediate and advanced). 11 participants assessed themselves as advanced players, 8 as intermediate players and 8 as beginners. For the 5-level GF, 6 participants played video games frequently (i.e. many times per week), 4 weekly, 7 monthly, 6 occasionally, and 4 hardly.

The main idea of statistical group-compare can be described by two steps: proving differences and comparing mean scores. Proving differences demonstrates that data was recorded from different groups or situations, and comparing mean scores is used to judge which group/situation is better.

A Mixed Between-within Subjects Analysis of Variance (Mixed ANOVA) (Clarke and Cooke, 2004) was conducted to investigate the impacts of TTL and GF on participants’ task performances in three different Display Situations. ANOVA is a test (parametric) of statistically significant differences between the mean scores of more than two groups. Mixed ANOVA consists of both between-subjects and within-subjects (also called repeated measure) designs. Between-subjects analysis

refers to the tests between independent groups, while within-subjects analysis indicates that one group is tested in more than two conditions/times.

In this case, the independent variables of TTL (3 levels) and GF (5 levels) are between-subjects, and the independent variable of Display Situation (3 levels) is within-subjects. Therefore, 3×5×3 Mixed ANOVAs were conducted. Since there are two dependent variables (Stroke and Point), the calculation of ANOVA was performed twice. In order to systematically compare each pair of factors in independent variables, Post-hoc tests (for between-subjects design) and Pairwise Comparisons (for within-subjects design) were conducted following ANOVA. Both of them are used to detect differences between all possible combinations of groups. Since the group sizes of TTL and GF are unequal (i.e. 3-level and 5-level), the Gabriel's procedure was selected for Post-hoc test (Allen and Bennett, 2010). The analysis results for Stroke and Point are shown in Tables 6.13 and 6.14, respectively.

The results of the two ANOVA tests indicate that the task performance of both "Stroke" and "Point" are strongly influenced by Display Situation. First of all, the scores of task performance are proved to be different with different Display Situations ($p < 0.001$). The F-ratio of "Point" ($F = 14.711$) is higher than "Stroke" ($F = 10.415$), which indicates that the influence of Display Situation is more significant for "making a good return". In addition, through Pairwise Comparisons, the "Point" scores are significantly higher with employing 3D display and Head-tracking (mean score: "3D&HT" > "3D" > "2D"). Although there is no statistical significant difference between "3D&HT" and "3D" situations for "Stroke", the mean score of "3D&HT" is higher. Therefore, the advantages of both stereoscopic display and viewpoint tracking technologies are obvious.

Mixed ANOVA for Stroke				
Effects	Variable	F-ratio	Significance	Partial Eta-squared
Within-Subjects	Display Situation	$F(2,32)=10.415$	$p<0.001$	$\eta^2=0.394$
Between-subjects	GF	$F(4,16)=3.539$	$p<0.05$	$\eta^2=0.469$
	TTL	No Statistical Significant Difference		
	TTL&GL	No Statistical Significant Difference		
Interaction	No Statistical Significant Difference			
Post-hoc Tests (Gabriel)				
Factors	Compared Pair	Mean Difference	Significance	
TTL	No Statistical Significant Difference			
GF	Hardly & Monthly	-2.82	$p<0.05$	
	Hardly & Weekly	-3.25	$p<0.05$	
	Hardly & frequently	-2.81	$p<0.05$	
	Others	No Statistical Significance		
Pairwise Comparisons				
Factors	Compared Pair	Mean Difference	Significance	
Display Situation	(3D&HT) & (3D)	No Statistical Significance		
	(3D) & (2D)	1.806	$p<0.05$	
	(3D&HT) & (2D)	2.633	$p<0.001$	

Table 6.13 ANOVA Results for Stroke Analyses

Mixed ANOVA for Point				
Effects	Variable	F-ratio	Significance	Partial Eta-squared
Within-Subjects	Display Situation	$F(2,32)=14.711$	$p <0.001$	$\eta^2=0.479$
Between-subjects	TTL	$F(2,16)=4.586$	$p <0.05$	$\eta^2=0.364$
	GL	No Statistical Significant Difference		
	TTL&GL	No Statistical Significant Difference		
Interaction	No Statistical Significant Difference			
Post-hoc Tests (Gabriel)				
Factors	Compared Pair	Mean Difference	Significance	
TTL	Beginner & Intermediate	-1.71	$p <0.05$	
	Beginner & Advanced	-2.17	$p <0.01$	
	Intermediate & Advanced	No Statistical Significant Difference		
GF	No Statistically Significant Difference			
Pairwise Comparisons				
Factors	Compared Pair	Mean Difference	Significance	
Display Situation	(3D&HT) & (3D)	1.167	$p <0.05$	
	(3D) & (2D)	1.091	$p <0.05$	
	(3D&HT) & (2D)	2.258	$p <0.001$	

Table 6.14 ANOVA Results for Point Analyses

The Partial Eta-squared, denoted by η^2 , is an index of the effect size, which indicates the proportion of the total variation attributable to one factor, excluding other factors. As shown in both tables, the values of 0.394 and 0.469 indicate that about 39% of the variability in “Stroke” scores attributed to the factor of Display Situation, and 47% was caused by the factor of GF. For the variation of “Point” scores, 48% of the variability attributed to the factor of Display Situation, and the factor of TTL led to 36% of the variability.

The Post Hoc analyses revealed that the participants who “hardly” play video games have significantly lower “Stroke” scores than the participants who play video games “frequently”, “weekly” or “monthly”. For the “Point” scores, table tennis players at the beginner level have lower “Point” scores than those at intermediate and advanced levels with statistical significance.

Although no statistically significant differences were revealed among other groups, by observing the histograms shown in Figure 6.4, the overall impacts of TTL and GF can be concluded. Generally speaking, a player with a higher table tennis level has better performances. If a player does not play video games usually, she/he tends to have lower scores of both “Stroke” and “Point”. However, if a player plays video games very often, there is no obvious tendency related to task performance.

To investigate the relationship between task performance and usability, both Pearson’s r and Spearman’s ρ were computed between task performances in “3D&HT” situation and each questionnaire item in TQ and PQ. As a result, both “Stroke” and “Point” scores have significant correlation with “Operation” (TQ item) ($r=0.436$, $\rho=0.457$ for Stroke; $r=0.433$ for Point; $p<0.05$ two-tailed), and “Co-experience” (PQ item) ($r=0.476$, $\rho=0.408$ for Stroke; $r=0.453$, $\rho=0.421$ for Point; $p<0.05$ two-tailed). Besides, players who have better performance on “Stroke” and “Point” tend to give higher marks to natural “Ball Collision” (TQ item) ($r=0.407$; $p<0.05$ two-tailed) and “Easy to use” (TQ item) ($r=0.517$; $p<0.05$ two-tailed) respectively.

Furthermore, “Stroke” and “Point” scores correlates with each other with a high statistical significance ($r=0.649$, $\rho=0.639$; $p<0.001$ two-tailed). On the other hand, there is no statistical correlation between SS-scores and task performances.

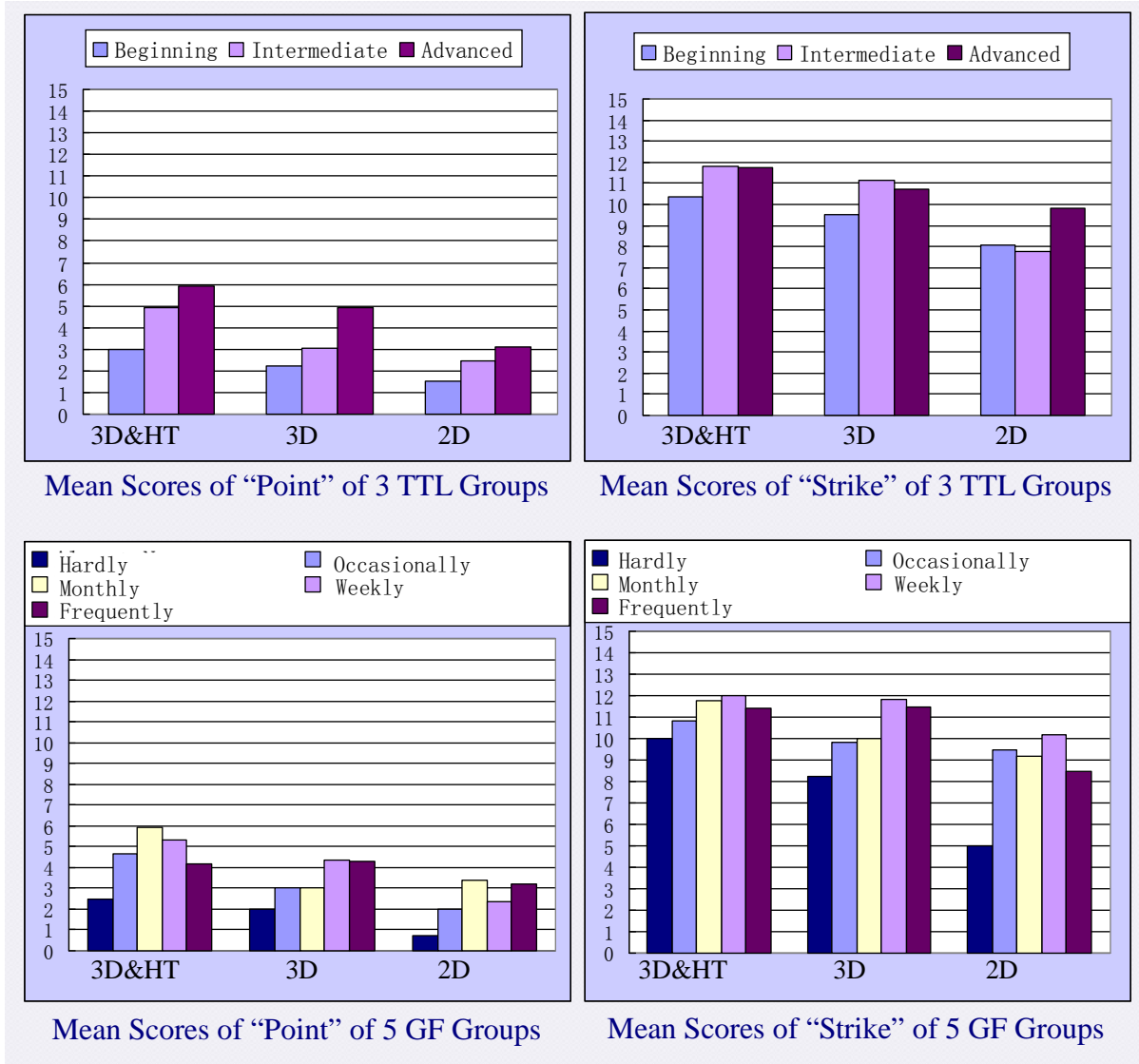


Figure 6.4 Comparing Mean Scores among Different Groups

6.4.3.5 Advantages of Displaying Opponent

Although the “Opponent’s Look” item in TQ got a relatively low score ($M=4.630$) due to the monochrome effect, players still preferred to see their opponents. To investigate

participants' perceptions for "Seeing opponent" and "Not seeing opponent" situations, five PQ items that are relevant to "Isolation", "Engagement" and "Sociality" issues were required to be rated for the two situations separately.

Both Paired Samples T-tests for parametric data and Wilcoxon Signed Rank Tests for non-parametric data were performed to compare the mean score differences of the five PQ items between "Seeing Opponent" and "Not Seeing Opponent" situations. The computed results are shown in Table 6.15, and their mean scores are compared by the histogram shown in Figure 6.5.

PQ items	Paired Samples T-tests		Wilcoxon Signed Rank Test	
	T value (df=26)	Significant (2-tailed)	Z value	Significant (2-tailed)
Unaware Real	t=3.358	P<0.05	Z=-2.739	P<0.01
Motivation	t=2.060	P<0.05	Z=-1.976	P<0.05
Involvement	t=2.884	P<0.01	Z=-2.588	P<0.01
Co-location	t=4.490	P<0.001	Z=-3.800	P<0.001
Co-experience	t=4.741	P<0.001	Z=-3.601	P<0.001

Table 6.15 Comparisons of five PQ items

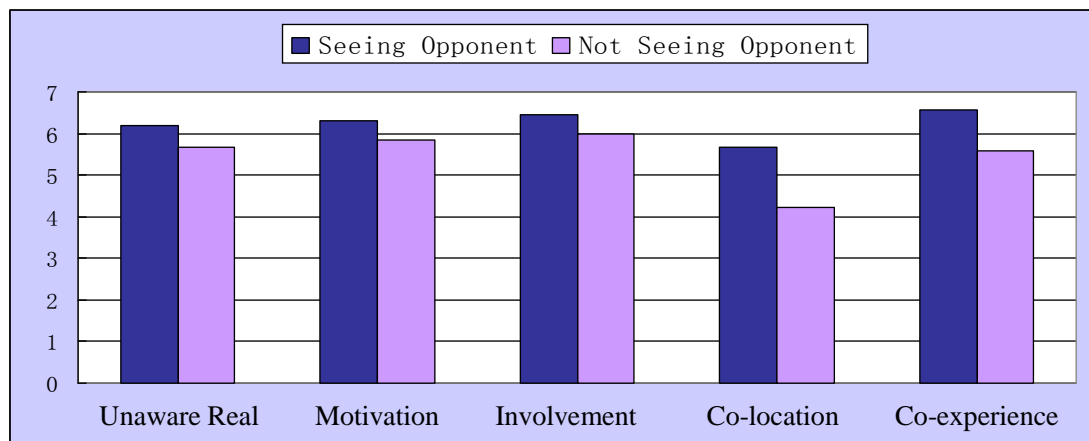


Figure 6.5 Mean scores of five PQ items

According to the above table and histogram illustration, all the five PQ items have statistically higher rating scores in "Seeing Opponent" situation, especially for the "Co-location" item.

Furthermore, two additional questions addressing participants' viewpoint of the advantages of "Seeing Opponent" are shown below.

- | |
|---|
| 1. To what extent do you agree that "seeing opponent" can help you to judge the ball's direction? |
| 2. To what extent do you agree that "seeing opponent" can make the game more attractive? |

These two questions have mean scores of 5.111 ($SD=1.476$) and 6.593 ($SD=1.010$) respectively, which indicate a high level of agreement from players' perspective.

6.5 Conclusion

Through the statistical analysis of both the qualitative and quantitative data obtained from the user-based experiment, the developed framework, heuristics and questionnaires were proved to be valid and reliable, and each component of the system was assessed by a rating score. Generally speaking, the system performance in terms of both technology and presence aspects is quite good. In addition, the statistical results prove the importance of stereoscopic display, head-tracking and displaying "opponent" technologies.

The main conclusions derived from the statistical analysis and user's feedback are listed below.

1. Questionnaires are reliable since both TQ and PQ have high internal consistency. The proposed framework, heuristics and questionnaires are valid since the correlations among TQ, PQ, Pre-Q and SSQ are consistent with the relationship expressed in the framework. Therefore, the evaluation method used in this project is valid and reliable.
2. Simulator sickness was not noticeable since the difference between the two SSQs completed before and after the experiment can not be proved statistically. However, if a participant's brain or eyes feel tired before the experiment, she/he is easier to suffer sickness symptom

3. The overall system performance in terms of technology achievement was rated as good ($M > 6$). According to the 10 high-score items (i.e. $M > 6$), the advantages of the system can be concluded. Players felt very natural when they were walking around. The 3D effect and the big size of the screen increased their immersive perception. They felt their hand tracker was portable, and movements of the virtual racquet were consistent with their real motion. In addition, the ball's flying and collision was rated as realistic, and the designs of the scene, objects, user menu and tutorial were given high marks as well.
4. Combining with player's comments, the items with relatively lower score can reveal weakness or limitation of the system, as well as the area for further development. The major weakness of the system is the loss-of-tracking problem. According to the performance evaluation of InterSense system described in Chapter 2, the probability of loss-of-tracking increases if 4 trackers work at the same time or if trackers move out from the good-tracking-range. With a two-player game requiring 4 trackers at the same time, the tracker will be out of the good-tracking-range if players put their hand down. The major limitation of the system is the running speed. Although it is quick enough to support a common real-time VR application, the delay is visible for a high-speed smash. For the area of further development, players prefer a coloured appearance of their opponent.
5. Players perceived deep immersion since the scores of all PQ items were in the range of 6 to 6.5.
6. The degree of involvement of this game was rated higher than table tennis sport, conventional table tennis video game, and Wii/PS3/Xbox360 table tennis game.
7. With 3D display and head-tracking, players had better performance in terms of both "Stroke" and "Point". It can be explained that stereoscopic display and head-tracking provide more accurate spatial information to players.
8. Good video game players tended to stroke more balls, whereas players with better table tennis skill tended to make more good returns. Since a common table tennis video game does not require players' table tennis skill, playing this game is more similar as playing a real table tennis game.

9. Seeing opponent increased the presence perception of players. In addition, players agreed that seeing opponent helped them to judge the ball's movement, and made the game more attractive.
10. The effect of colliding sounds made the game more realistic. The immersive effect can be further increased by using spatialised sound (further work).
11. Further work should also include the implementation of haptic feedback, which provides players more realistic perception during ball-racquet collision.

Chapter 7

Conclusion and Future Work

7.1 Conclusions

A realistic, immersive and high precision simulation system for the table tennis game was developed and described in this thesis. The system provides both single player mode and two players competitive mode. By integrating the state of the art 6-DOF motion tracking system, 3D body-movement tracking system, and stereoscopic display system, the proposed and developed table tennis game fulfils the comparison criteria listed in Chapter 2, and outperforms other existing systems. The advanced features of the developed system are shown in Table 7.1.

	Immersive Display	Hand Tracking	Head Tracking	Multiply Players	Other Features
Developed system	Yes	Yes	Yes	Yes	Displaying 3D opponent and Comprehensive physics model

Table 7.1 Features of the Developed System

The InterSense Motion Tracking System enables multiple (4 trackers were used in this project) high precision 6-DOF tracking stations to be tracked in high speed (120Hz). It is also compared favourably against other tracking systems for the application stated in this thesis in terms of portability, tracking range and line-of-sight occlusion.

The Swissrange SR4000TM 3D Camera is capable to acquire depth and shape information of objects in the measurement volume in up to 54FPS. A 35FPS frame rate was selected in order to achieve smooth animation with a reasonably good quality.

Three-wall rear-projection stereoscopic screens were employed as the display media, with each at the size of 2.74 m (length) by 2.06 m (height). The big size of the screen

enables the table tennis gaming environment to be provided in a real physical scale for full immersion of the participants.

On the software side, all the virtual objects were designed according to the physical measurement of the real objects. The physics based ball animation model developed in this research takes into account various physical phenomena such as gravity, air resistance, Magnus effect, ball spin, friction and coefficient of restitution in collision, which is more realistic than other available systems. For data communication, the TCP/IP protocol was used for data transfer between the server and client PCs. Multi-threaded programming was also applied to enable the whole system running smoothly.

The performance of the system was evaluated through a user-based study. Through the statistical analysis of both the qualitative and quantitative data obtained from the user-based experiment, the developed framework, heuristics and questionnaires were proved to be valid and reliable, and each component of the system was assessed by rating scores. Generally speaking, the system performance in terms of both technology and presence aspects is good. In addition, the statistical results proved the importance of stereoscopic display, head-tracking and displaying “opponent” technologies.

The overall system performance was rated as good (a mean score of 6 in 7-level Likert-Scale). According to the high-score items (i.e. $M > 6$), the advantages of the system can be concluded. Players felt very natural when they were walking around and interacting with the VR environment. The stereoscopic effect and the big size of the screen provided players a deep perception of immersion. Players satisfied with the usage of light-weight trackers, and movements of the virtual racquet were seen to be consistent with their real motion. In addition, the ball’s flying and collision were rated as realistic, and the designs of the scene, objects, and user interface were given high marks as well.

7.2 Contributions

The original contributions of the research work presented in this thesis can be summarised as follows:

- (1) A novel VR environment was designed and implemented, which provides a new and unique way for multiplayer to engage in immersive and competitive real-time play of the table tennis game. In the two-player playing mode, the world coordinate system was transferred to each player's local viewing coordinate, and a special "server-client" network was designed for communication. The system integrated the state of the art 6-DOF motion tracking system, 3D body-movement tracking system, and stereoscopic display system. Although these equipments had been employed in VR researches previously, the system design and application area are different. Current simulation systems for the table tennis application have also been reviewed, their system components and design ideas are different with the system presented in this thesis. The system design and reviews of current works are introduced in Chapter 2, the hardware selection and operation principles are presented in Chapter 3. Chapter 4 and 5 describe the implementation of single and two-player game in details.
- (2) The universal approach for viewpoint setting in 3D space is to set the virtual camera as an animated object. In this way, virtual objects can be observed from different viewing directions. But for an immersive VR environment that requires accurate 3D spatial information to be perceived by participants, head-tracking based real-time frustum adjustment is shown to be necessary. A comparison is given in this thesis to demonstrate the differences of the visualisation results between a fix frustum setting and a dynamic frustum setting, which is one of the differences between a conventional video game environment and a realistic VR environment in terms of viewpoint setting. (Section 4.4)

- (3) A complete design of the physics based ball animation model was derived. For most of the exergames with lack of tracking DOF or stereoscopic display, a simple game strategy, such as “waving arm to get point”, with a simple physical model is normally used. Therefore, most literatures that introduce game-physics do not give comprehensive analysis, especially for the complicated ball-racquet collision process. The physics concepts normally found in pure physics articles, or literatures for sports research, contain too much details (i.e. parameters of racquet’s deformation and viscosity) and not practical in such a real-time application. The physical model derived in this thesis considered most of the major effects during ball flying (i.e. gravity, air resistance, Magnus effect) and ball collision (e.g. friction acting on the contact point, change of linear and angular velocities and energy loss during elastic collision). This model also includes the method of trajectory prediction, which is simple but practical, and is more suitable for a real-time tracking-based application. (Section 4.5)
- (4) Instead of using 2D video or computer-generated avatar, a 3D surface that represented player’s real appearance was generated to display opponent’s body motion. This design is different from other existing system, thereby providing a distinct feature to give players a sense of “presence in a virtual space with a friend”. This feature enhanced the immersive perception and made the game more attractive. (Section 5.4)
- (5) Although usability tests form a common evaluation method in the HCI area, there is still no standardized measurement available for varied applications in the VR field due to its novelty and varied interaction approaches. Recent researches that addressed the VR evaluation methodology mainly focused on the heuristics-based system element check and questionnaire-based tests to inquire users’ “presence” perception. However, only a few literatures addressed both of them. A framework and a set of heuristic were proposed in this thesis, which discussed the main

factors associated with a VR system, and demonstrated the relationship among these factors. The issues addressed by previous literatures were organised, which gave a clear idea for VR system evaluation. In addition, a set of questionnaire was developed as well based on the framework and heuristic, which not only addressed the technological performance about a VR system, but also enquired the “presence” perceived by users. (Chapter 6)

- (6) A user-based study was carried out to evaluate the developed VR environment. Through the statistical analysis of both the qualitative and quantitative data obtained from the user-based experiment, the developed framework, heuristics and questionnaires were proved to be valid and reliable. Both the technology achievement and the degree of presence provided by the system were evaluated. In addition, statistical evaluation proved the advantages of applying stereoscopic display, head-tracking and displaying opponent technologies in a VR system. The design of this experiment with questionnaires was based on the theoretical investigation presented in this thesis. Therefore, it not only evaluate the system, but also proved the correlation between theories and results. (Chapter 6)

7.3 Future Work

The potential future work is summarised as follows:

- (1) System used for training: A very important application of the current system that have developed is for sports analysis and training. Since all the movements and trajectories of the racquets and ball are known and recorded. The trajectories of movement can be compared with experts’ actions. Alternatively, expert performances can be recorded and repeated to train beginners. Furthermore, special ball trajectories can be simulated and repeated for training particular

techniques.

- (2) 3D colour opponent display: The current 3D opponent was displayed in grey-level images. It would enhance the visual effects, if true colour information can be added to the depth map. This can be achieved by mounting another calibrated video camera with known internal camera parameters and external geometrical relationship to the SR camera. The colour information can then be mapped onto the depth image.
- (3) Racquet force feedback: Some of the gaming system enables a force feedback to the players' controller, or a haptic device was used. Some researchers installed a small motor to the racquet to simulate the force, which could be considered as an area for future work.
- (4) Evaluation and improvement of the physical model: Further work can also be done by comparing the simulated physics model with real ball collision and responses. This requires a high speed camera and detailed analysis of the ball's movement at fraction of second.
- (5) Extending the system to other application: Since the physics model and user interactions are similar for the ball-based games, the current system can be extended to other ball games, such as table cricket, golf and pool game.

References

Alam, F., Subic, A. J., Naser, J., Rasul, M. and Khan, M. M. K. (2008). "A study of spin effects on tennis ball aerodynamics." WSEAS transactions on fluid mechanics. Special issue on sustainable energy and environmental fluid mechanics 3(3): 271-278.

Alencar, M. F. C., Raposo, A. B. and Barbosa, S. D. J. (2011). "Composition of HCI evaluation methods for hybrid virtual environments." Proceedings of the 2011 ACM symposium on applied computing (SAC '11): 1237-1244.

Allen, P. and Bennett, K. (2010). "PASW statistics by SPSS: a practical guide." version 18. Publisher: Thomas Nelson Australia. ISBN-13: 9780170188555. ISBN-10: 0170188558.

Argyrous, G. (2005). "Statistics for research: with a guide to SPSS." 2nd edition. Publisher: Sage publications Ltd. ISBN-13: 9781412919487. ISBN-10: 1412919487.

Bianchi, G., Knörlein, B., Székely, G. and Harders, M. (2006). "High precision augmented reality haptics." Eurohaptics 2006: 169-177.

Bideau, B., Kulpa, R. Vignais, N., Brault, S., Multon, F., Craig, C. (2010). "Using virtual reality to analyze sports performance." Computer graphics and applications, IEEE 30(2): 14-21.

Bourg, D. M. (2001). "Physics for game developers." Publisher: O'Reilly media. ISBN-13: 9780596000066. ISBN-10: 0596000065.

Bowman, D. A., Gabbard, J. L. and Hix, D. (2002). "A survey of usability evaluation in virtual environments: classification and comparison of methods." Presence 11(4): 404-424.

Brunnett, G., Rusdorf, S. and Lorenz, M. (2006). "V-Pong: an immersive table tennis

simulation." *Computer graphics and applications*, IEEE 26(4): 10-13.

Büttgen, B. and Seitz, P. (2008). "Robust optical time-of-flight range imaging based on smart pixel structures." *IEEE Transactions on circuits and systems* 55(6): 1512-1525.

Champion, E., Bishop, I., and Dave, B. (2011). "The Palenque project: evaluating interaction in an online virtual archaeology site." *Virtual reality*: 1-19.

Chapman, D. D. and Stone, S. J. (2010). "Measurement of outcomes in virtual environments." *Advances in developing human resources* 12(6): 665-680.

Chertoff, D. B., Goldiez, B. and LaViola, J.J. (2010). "Virtual experience test: a virtual environment evaluation questionnaire." *IEEE Virtual reality conference (VR '2010)*: 103-110.

Clarke, G. M. and Cooke, D. (2004). "A basic course in statistics." 5th edition. Publisher: John Wiley & Sons. ISBN-13: 9780470973875. ISBN-10: 0470973870.

Cramer, H. S. M. (2004). "Usability evaluation and context analysis to support development of virtual reality systems." Master Thesis. Faculty of Science, University of Amsterdam, Netherlands,

Cross, R. (2000). "Effects of friction between the ball and strings in tennis." *Sports engineering* 3(2): 85-97.

Cross, R. (2002). "Measurements of the horizontal coefficient of restitution for a superball and a tennis ball." *American journal of physics* 70(5): 482-489.

Cruz-Neira, C., Sandin, D. J. and DeFanti T. A. (1993). "Surround-screen projection-based virtual reality: the design and implementation of the CAVE." In *proceedings of the 20th annual conference on computer graphics and interactive techniques (SIGGRAPH '93)*: 135-142.

Demiralp, C., Jackson, C. D., Karelitz, D. B., Zhang, S. and Laidlaw, D. H. (2006). "CAVE and Fishtank virtual-reality displays: a qualitative and quantitative comparison." *IEEE Transactions on visualization and computer graphics* 12(3): 323-330.

Dodgson, N. A. (2005). "Autostereoscopic 3D displays." *Computer* 38(8): 31-36.

Doshi, A., Hilton, A. and Starek, J. (2008). "An empirical study of non-rigid surface feature matching." *5th European conference on visual media production (CVMP '2008)*: 1-10.

Dumas, J. S. (2002). "User-based evaluation." *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*: 1093-117.

Finkelstein, S., Nickel, A., Lipps, Z., Barnes, T. and Wartell, Z. (2011). "Astrojumper: motivating exercise with an immersive virtual reality exergame." *Presence - teleoperation and virtual environments* 20 (Special Issue: Virtual Reality and Sports Guest Editors' Introduction): 78-92.

Foxlin, E., Harrington, M. and Pfeiffer, G. (1998). "Constellation: a wide-range wireless motion tracking system for augmented reality and virtual Set applications." *Proceedings of the 25th annual conference on Computer graphics and interactive techniques (SIGGRAPH '98)*: 371-378.

Foxlin, E., Harrington, M. and Altshuler, Y. (1998). "Miniature 6-dof inertial system for tracking HMDs." *Proceedings of SPIE helmet and head-mounted displays III*. 3362: 214-228.

Froehlich, B., Blach, R., Stefani, O., Hochstrate, J., Hoffmann, J., Kluger, K. and Bues, M. (2005). "Implementing multi-viewer stereo displays." *Proceedings of computer graphics, visualization and computer vision (WSCG '05)*: 139-146.

Funkhouser, T. Jot, J.-M. and Tsingos, N. (2004) "Survey of methods for modeling sound propagation in interactive virtual environment systems." *Presence - teleoperators and virtual*

environments.

Gabbard, J. L. (1997). "A taxonomy of usability characteristics in virtual environments." Master Thesis. Computer science and applications, Virginia Polytechnic Institute.

Garwin, R. L. (1969). "Kinematics of an ultra elastic rough ball." American journal of physics 37(1): 88-92.

Grey, J. (2002). "Human-computer interaction in life drawing, a fine artist's perspective." Proceedings 6th international conference on information visualisation: 761-770.

Guomundsson, S. A., Larsen, R., Aanaes, H., Pardas, M. and Casas, J. R. (2008). "TOF imaging in smart room environments towards improved people tracking." IEEE Computer society conference on computer vision and pattern recognition workshops (CVPRW'08): 1-6.

Hakkarainen, M. and Woodward, C. (2005). "SymBall: camera driven table tennis for mobile phones." Proceedings of the 2005 ACM-SIGCHI international conference on advances in computer entertainment technology: 391-392.

Hearn, D. D. and Baker, M. P. (2004). "Computer graphics with OpenGL." 3rd edition. Publisher: Pearson Prentice Hall. ISBN-13: 9780131202382. ISBN-10: 0131202383.

Hegde, G. M. and Ye, C. (2008). "SwissRanger SR-3000 range images enhancement by a singular value decomposition filter." International conference on information and automation: 1626-1631.

Henry, J. A. G. and Polysb, N. F. (2010). "The effects of immersion and navigation on the acquisition of spatial knowledge of abstract data networks." International conference on computational science (ICCS '2010): 1737-1746.

Herda, L., Fua, P., Plankers, R., Boulic, R., Thalmann, D. (2001). "Using skeleton-based tracking to increase the reliability of optical motion capture." *Human movement science* 20(3): 313-341.

Ijsselstein, W. A., Ridder, H. D., Freeman, J. and Avons, S. E. (2000). "Presence: concept, determinants, and measurement." *Proceedings of SPIE human vision and electronic imaging*: 520-529.

ITTF (2004). ITTF Technical Leaflet T1: The Table.

ITTF (2009). ITTF Technical Leaflet T3: The Ball.

Kalawsky, R. S. (1999). "VRUSE-a computerised diagnostic tool: for usability evaluation of virtual/synthetic environment systems." *Applied ergonomics* 30(1): 11-25.

Katzourin, M., Ignatoff, D., Quirk, L., Jr., J. J. L. and Jenkins, O. C. (2006) "Swordplay: innovating game development through VR." *IEEE Computer graphics and applications*: 15-19.

Kennedy, R. S., Lane, N. E., Berbaum, K. S. and Lilienthal, M. G. (1993). "Simulator sickness questionnaire: an enhanced method for quantifying simulator sickness." *The international journal of aviation psychology* 3(3): 203-220

Kim, Y.-B., S.-H. Han, Kim, S.-J., Kim, E.-J. and Song, C.-G. (2007). "Multi-player virtual ping-pong game." *17th international conference on artificial reality and telexistence*: 269-273.

Lessiter, J., Freeman, J., Keogh, E. and Davidoff, J. (2001). "A cross-media presence questionnaire: the ITC-sense of presence inventory." *Presence - teleoperators and virtual environments* 10(3): 282-297.

Li, S. and Sun, J. (2009). "Application of virtual reality technology in the field of sport." *First international workshop on education technology and computer science (ETCS '09)*: 455-458.

Liarokapis, F. (2006). "An exploration from virtual to augmented reality gaming." *Simulation & gaming* 37(4): 507-533.

Lipton, L., Phillips, S., Huggins, C., Shafer, A. and John, M.D. (2010). "Stereoscopic 3D." *The VES handbook of visual effects: industry standard VFX practices and procedures*: 387-464. 1st edition. Publisher: Focal Press. ISBN-13: 9780240812427. ISBN-10: 0240812425.

Litwiller, T., Joseph, J. and LaViola, J. (2011). "Evaluating the benefits of 3D stereo in modern video games." *Proceedings of the 2011 annual conference on Human factors in computing systems (CHI'11)*: 2345-2354.

Lu, G., Shark, L-K., Hall, G. and Zeshan, U. (2011). "Immersive manipulation of virtual objects through glove-based hand gesture interaction." *Virtual Reality*.: 1-10.

Marco, P., Nadia, B.-B., Betsy, D. V. and Anton, N. (2009). "Movement-based sports video games: investigating motivation and gaming experience." *Entertainment computing* 1(2): 49-61.

Magenat-Thalmann, N. and Bonanni, U. (2006). "Haptics in virtual reality and multimedia." *IEEE Multimedia*: 6-11

Morris, D., Joshi, N. and Salisbury, K. (2004). "Haptic Battle Pong: high-degree-of-freedom haptics in a multiplayer gaming environment." *Experimental gameplay workshop, GDC*.

Mueller, F. F., Edge, D., Vetere, F., Gibbs, M., Agamanolis, S., Bongers, B. and Sheridan, J. G. (2011). "Designing sports: a framework for exertion games." *Proceedings of the 2011 annual conference on Human factors in computing systems (CHI '11)*: 2651-2660.

Mueller, F. F., Vetere, F. and Gibbs, M. (2008). "The design of networked exertion games." *Virtual reality and broadcasting* 5(13).

Nakashima, A., Kobayashi, Y., Ogawa, Y. and Hayakawa, Y. (2009). "Modeling of rebound phenomenon between ball and racket rubber with spinning effect." ICCAS-SICE International Joint Conference: 2295-2300

Noser, H., Stern, C., Stucki, P. and Thalmann, D. (2000). "Generic 3D ball animation model for networked interactive VR environments." Proceedings of the 2nd international conference on virtual worlds (VW'00): 77-90.

Ogi, T., Yamada, T., Kurita, Y., Hattori, Y. and Hirose, M. (2003). "Usage of video avatar technology for immersive communication." Proceedings of first international workshop on language understanding and agents for real world interaction: 24-31.

Pallant, J. (2007). "SPSS survival manual: a step by step guide to data analysis using SPSS for Windows." 3rd edition. Publisher: Open University Press. ISBN-13: 9780335223664. ISBN-10: 0335223664.

Park, J.-S., Kim, T.-Y. and Yoon, J.-H. (2006). "AR table tennis: A video-based augmented reality sports game." In proceedings of ICAT '2006: 197-206.

Pintaric, T. and Kaufmann, H. (2007). "Affordable infrared-optical pose tracking for virtual and augmented reality." IEEE VR workshop on trends and issues in tracking for virtual environments: 44-51.

Rolland, J. P., Baillot, Y. and Goon, A. A. (2001). "A survey of tracking technology for virtual environments." Fundamentals of wearable computers and augmented reality.

Root, J., D. Gordon, et al. (2010). "Performance and motion capture." The VES handbook of visual effects: industry standard VFX practices and procedures: 335-386. 1st edition. Publisher: Focal Press. ISBN-13: 9780240812427. ISBN-10: 0240812425.

Santos, B. S., Dias, P., Santos, P., Silva, S. and Ferreira, C. (2009). "Usability evaluation in Virtual Environments through empirical studies involving users." ACM-CHI workshop challenges in evaluating usability and user experience in reality based interaction.

Sherman, W. R. and Craig, A. B. (2002). "Understanding virtual reality: interface, application, and design." 1st edition. Publisher: Morgan Kaufmann. ISBN-13: 9781558603530. ISBN-10: 1558603530.

Shreiner, D. and T. K. O. A. W. Group (2009). "OpenGL programming guide: the official guide to learning OpenGL." 7th edition. Publisher: Addison Wesley Professional. ISBN-13: 9780321552624. ISBN-10: 0321552628.

Sinclair, J., Hingston, P. and Masek, M. (2007). "Considerations for the design of exergames." Proceedings of the 5th international conference on computer graphics and interactive techniques in Australia and Southeast Asia (GRAPHITE '07): 289-295.

Slater, M., Usoh, M. and Steed, A. (1994). "Depth of presence in virtual environments." Presence - teleoperators and virtual environments 3(2): 130-144

Smolic, A. (2011). "3D video and free viewpoint video - from capture to display." Pattern recognition 44(9): 1958-1968.

Stahler, W. (2004). "Beginning math and physics for game programmers." Publisher: New Riders Games. ISBN-13: 9780735713901. ISBN-10: 0735713901.

Stanney, K. M. (2002). "Handbook of virtual environments: design, implementation, and applications." Publisher: Lawrence Erlbaum Associates. ISBN-13: 9780805832709. ISBN-10: 080583270X.

Stanney, K. M. and J. V. Cohn (2003). "Virtual environment." The human-computer interaction

handbook: fundamentals, evolving technologies and emerging applications. J. A. Jacko and A. Sears, Routledge: 621-634.

Stewart, T. and D. Travis (2003). "Guidelines, standards, and style guides." The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications. J. A. Jacko and A. Sears, Routledge.

Sutcliffe, A. G. and Kaur, K. D. (2000). "Evaluating the usability of virtual reality user interfaces." Behaviour and information technology 19(6): 415-426.

Sylaiou, S., K. Mania, K., Karoulis, A. and White, M. (2010). "Exploring the relationship between presence and enjoyment in a virtual museum." International journal of human-computer studies 68(5): 243-253.

Timmerman, P. and van der Weele, J. P. (1999). "On the rise and fall of a ball with linear or quadratic drag." America journal of physics 67(6): 538-546.

Von Kapri, A., Rick, T. and Feiner, S. (2011) "Comparing steering-based travel techniques for search tasks in a CAVE." IEEE Virtual reality conference: 91-94

Wan, H., Zou, S., Dong, Z., Lin, H. and Bao, H. (2011). "MRStudio: A mixed reality display system for aircraft cockpit." IEEE International symposium on VR innovation (ISVRI): 129-135.

Watt, A. and Policarpo, F. (2000). "3D Games: Real-Time Rendering and Software Technology." Publisher: Addison Wesley. ISBN-13: 9780201619218. ISBN-10: 0201619210.

Weinland, D., Ronfard, R. and Boyer, E. (2011). "A survey of vision-based methods for action representation, segmentation and recognition." Computer vision and image understanding 115(2): 224-241.

Witmer, B. G. and M. J. Singer (1998). "Measuring presence in virtual environments: a presence questionnaire." *Presence: teleoperators and virtual environments* 7(3): 225-240.

Woodward, C., Honkamaa, P., Jäppinen, J. and Pyökkimies, E. (2004). "CamBall: augmented networked table tennis played with real rackets." *Proceedings of the 2004 ACM-SIGCHI international conference on advances in computer entertainment technology*: 275 - 276.

Wormell, D. and Foxlin, E. (2003). *Advancements in 3D interactive devices for virtual environments. Proceedings of the eurographics workshop on virtual environments (EGVE '03)*: 47-56.

Wright, R. S. and Lipchak, B. (2004). "OpenGL superBible." 3rd edition. Publisher: Sams. ISBN-13: 9780672326011. ISBN-10: 0672326019.

Xiao, M. (2001). "Research on coefficient of friction and coefficient of restitution of different table tennis racquets (In Chinese)." *TPEC Press* 9: 45 -62.

Zargarpour, H., H. LaBounta, et al. (2010). "Interactive games." *The VES handbook of visual effects: industry standard VFX practices and procedures*: 707-736. 1st edition. Publisher: Focal Press. ISBN-13: 9780240812427. ISBN-10: 0240812425.

Appendix A

Published Work

Li, Y., Shark, L-K, Hobbs, S. J., Ingham, J. (2010). "Real-time immersive table tennis game for two players with motion tracking." IEEE 14th International conference of information visualisation (IV' 2010): 500-505.

Real-Time Immersive Table Tennis Game for Two Players with Motion Tracking

Yingzhu Li / Lik-Kwan Shark

Applied Digital and Signal Image Processing Centre
School of Computing, Engineering and Physical
Sciences

University of Central Lancashire
Preston, U.K.

YZLi@uclan.ac.uk / LShark@uclan.ac.uk

Sarah Jane Hobbs

Centre for Applied Sport and Exercise Science
University of Central Lancashire
Preston, U.K.

SJHobbs1@uclan.ac.uk

James Ingham

School of Journalism, Media and Communications
University of Central Lancashire
Preston, U.K.

JIngham@uclan.ac.uk

Abstract—Presented in this paper is a novel real-time virtual reality game developed to enable two participants to play table tennis immersively with each other’s avatar in a shared virtual environment. It uses a wireless hybrid inertial and ultrasonic tracking system to provide the positions and orientations of both the head (view point) and hand (racket) of each player, as well as two large rear-projection stereoscopic screens to provide a view-dependent 3D display of the game environment. Additionally, a physics-based ball animation model is designed for the game, which includes fast detection of the ball colliding with table, net and quick moving rackets. The system is shown to offer some unique features and form a good platform for development of other immersive games for multiple players.

Keywords- *Immersive Game, Motion Tracking, Stereoscopic Display, Interaction Techniques*

I. INTRODUCTION

One of the most challenging research areas in virtual reality (VR) [1] is to enable fast interaction among multiple participants in real time, which requires not only high speed movement tracking of multiple objects but also high update rate of multiple displays. In an effort to address the challenge, the paper reports the design and development of a virtual table tennis game to allow two participants to immerse in a physical play against each other in real-time.

To achieve high speed movement tracking of rackets and player view points, the hybrid inertia and ultrasonic sensing technology from InterSense [2] is used, with each player holding a hand tracker as grasping a racket, and wearing a head tracker. To achieve visual immersion, two large rear-projection stereoscopic screens are used to provide an individual view for each player based on the player’s perspective. By wearing a pair of polarised glasses, each player is able to see his/her own virtual racket, a standard table tennis table and a flying ball, as well as the opponent’s avatar holding a racket, in 3D with an impression of depth.

This paper is organised into five sections. Section 2 provides a brief review of related work, and Section 3 presents system hardware in terms of tracking, display and computing systems. This is followed by software implementation in Section 4, which describes scene generation, physics based calculation of the ball’s trajectory, stereoscopic rendering, and a simple user interface. System performance is presented in Section 5 and finally, conclusions are given.

II. PREVIOUS WORK

There are several ball-racket based games that support real-time physical interaction between players. One based on network is CamBall [3] that allows two remote users to see and play with each other using PCs with web cameras. The others based on mobile phones are SymBall [4] and ARTennis [5] which allow face-to-face playing over the Bluetooth connection. All these three games employ optical tracking technology and use simple game strategy. In the video game industry, the best-selling console Nintendo Wii [6] with its motion sensing controllers allows the player to interact with virtual items or opponent by the hybrid inertia and optical tracking technology. Some ball-racket games were released for Wii, which deliver complex game strategy and support multiple players. However, none of the above games deliver sufficient visual immersion.

By using stereoscopic display, more immersive games have been developed. One employs marker-based infrared tracking, rear-projection stereoscopic screen and complex physics-based animation model [7]; and the other is AR Table Tennis [8] that employs video tracking for a Head-Mounted Display (HMD). However, these two systems are only available for a single player. Although CyberTennis [9] employs two HMDs to enable interaction of players, its use of magnetic tracking restricts the movements of players.

The investigation of the related work has led to the design of the proposed system based on the hybrid inertia

and ultrasonic sensing technology for fast motion tracking and two large stereoscopic screens to provide view dependent real scale immersion. Since table tennis game is characterised by fast movements in a relatively small space, it is a good prototype for demonstrating the capabilities of the proposed system.

III. SYSTEM HARDWARE

The hardware of the system contains three major components: the tracking system, the display system and computers with advanced graphic cards.

A. Tracking System

Two most crucial parts for tracking in an immersive table tennis game are the hand and head of each player. While hand tracking is used to control the virtual racket, head tracking enables the virtual scene to be changed in real-time according to the viewpoint of each player. An IS-900 wireless tracking system from InterSense is used for this purpose, since it is able to track in 6-DOF (6 degrees of freedom: X, Y, Z, Yaw, Pitch, and Roll) without line of sight requirements.

The InterSense system configuration is shown in Fig. 1, SoniStrips containing ultrasonic SoniDisc transponders is mounted on the ceiling, which transmits ultrasonic pulses upon receiving addressed signals from the Processor Unit connected to the serial port of the application host computer. Each player wears a MicroTrax wireless head tracker and holds a MicroTrax wireless wand (shown in Fig. 2), with each one of them containing inertial sensors and ultrasonic receivers. Whilst the outputs from the inertial sensors, consisting of accelerometers and gyros, are used to determine the position and orientation of each sensor in 3D space, the range measurements based on time-of-flight between ultrasonic emitters and receivers are used to correct the drifting effect inherent within the inertial sensors.

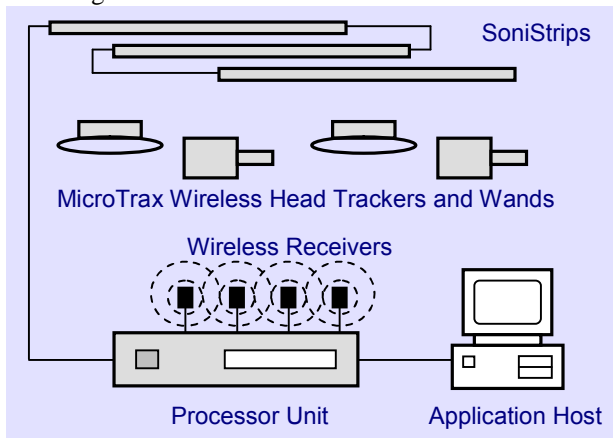


Figure 1. InterSense System Configuration



Figure 2. MicroTrax Tracking Devices

B. Display System

To create a visual illusion of depth, two large rear-projection stereoscopic screens are used with each providing a correct 3D view for each player. The size of each screen is 2.74m in length and 2.06m in height to enable display of the table tennis gaming environment in a real physical scale, and the resolution is 1024×768 pixels. The configuration for each screen is shown in Fig. 3. Two Epson PowerLite8800 projectors, situated at the back of the screen with a pair of circular polarizing filters placed in front of the lens, are used to superimpose two differently polarized images on the same screen via a reflecting mirror. Each screen with two projectors is driven by one computer through its graphics card output ports. By wearing a pair of light-weight polarised glasses, each player is able to see the 3D table tennis gaming environment with depth effect.

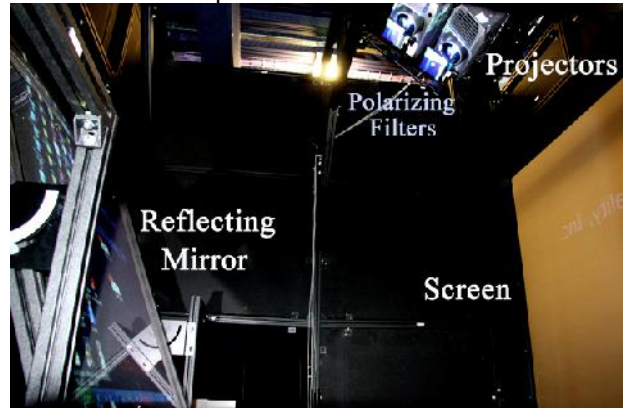


Figure 3. Rear-projection Stereoscopic Screen

C. Computing System

The computing system consists of three PCs and each one runs on an Intel Xeon 3.06GHz CPU with 2G RAMs and a 256MB NVIDIA Quadro FX3000 Graphic Card. One PC is used as the server to run the application program, which is responsible for the InterSense tracking control, data processing and animation computation. The other two PCs are used as clients, which each one renders the scene according to the computation results sent from the server, and drives a pair of projectors to provide an individual stereoscopic display according to the viewpoint of each player. The communication between the server and two clients is based on the TCP/IP protocol, and data is transmitted through a 1G Ethernet connection.

IV. SOFTWARE IMPLEMENTATION

With the application program run in the server and the displays generated by the two clients, the software implemented using the C++ programming language is based on the software modules and data flow diagram illustrated in Fig. 4. For the server computer, it runs the Motion Data Acquisition Module to acquire position and orientation data of the head and hand of each player from the tracking system based on the InterSense Application Programming Interface (API), the Motion Data Processing Module to provide player viewpoints as well as positions and orientations of the virtual rackets and avatars to be drawn by the two client computers, and the Ball Animation and Audio Feedback Module to provide the motion of the ball according to simplified physical laws and a sound if a collision is detected. For the two client computers, all fixed static virtual objects (e.g. table, wall and floor) are pre-computed, and each one runs its own Scene Generation Module to produce a stereo pair for each screen upon receiving the dynamic object data from the server.

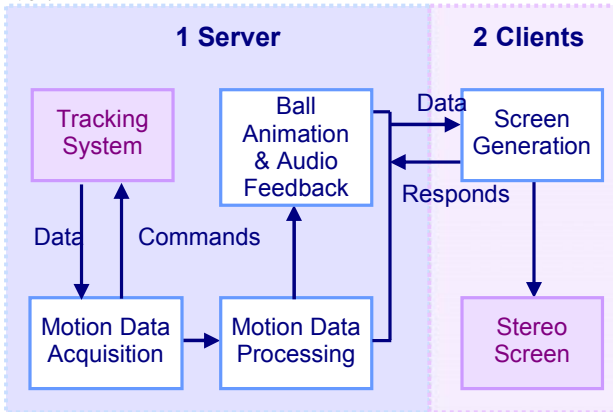


Figure 4. Software Modules and Data Flow

A. Scene Generation

All objects in the scene are generated by using the C++ programming language with the OpenGL API [10]. OpenGL uses a frustum of a pyramid to assign which parts of the scene need to be rendered, and the virtual camera (player's viewpoint) is placed at the apex of the pyramid. With Fig. 5 illustrating the spatial relationship between the global coordinate system, the local coordinate system used by InterSense, and the OpenGL frustum. By using 60° viewing angle, the origin of the global coordinate system is located at the middle of the screen at a distance of 2.37m in front of the screen along the z-axis. Since the InterSense coordinate system has different coordinate orientations and origin position as shown in Fig. 5, geometrical transformations are needed to bring InterSense data into the global coordinate system, and this is performed in the Motion Data Processing Module by the server before sending the acquired motion to the two clients.

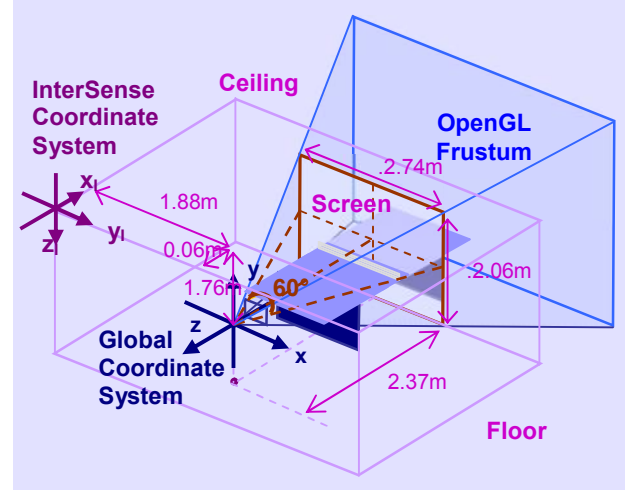


Figure 5. Projection and Coordinate Systems

The objects in the scene are formed by using basic geometric models. The sizes of the ball (0.04m diameter) and table (2.74m in length; 1.525m in width; 0.76m in height) with net (0.1525m in height) are created according to the specifications of a standard table tennis game. Since there are no particular specifications for size or shape of the rackets, two round shape rackets with the size of 0.16m in diameter are used.

Two simple avatars with rigid bodies are used to show the opponent position based on the player head positions acquired. The right shoulder of each avatar is connected to the corresponding racket with its position based on the hand positions acquired. Although the movements of the avatars are not realistic, they give players a good impression of "playing with a moving opponent". A screenshot of the view created for one player is shown in Fig. 6

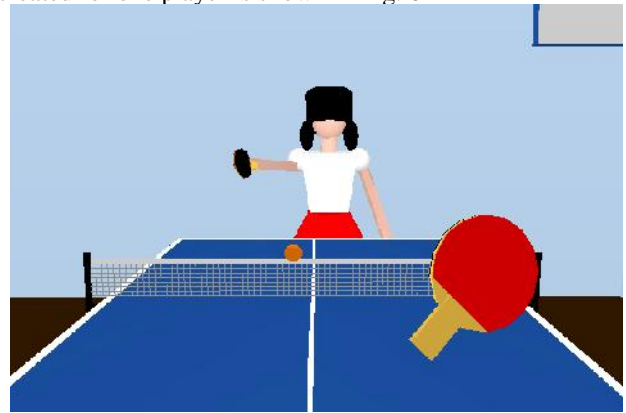


Figure 6. Scene Generated

B. Ball Animation

The ball movement model is derived from the method proposed in [11], by applying simplified physical laws to two basic states. One is flying that describes ball movement due to inertia and is influenced by gravity (air resistance is ignored for simplicity). The other is collision that describes

interaction between the ball and other virtual objects to gives a new initial position and velocity of flying.

1) Flying

For a ball at coordinate \vec{p}_b in the current frame with a velocity \vec{v}_o , if there is no collision, the position of the ball in next frame based on simplified physics laws is given by

$$\vec{p}_b' = \vec{v}_o t + \frac{1}{2} \vec{a} t^2 + \vec{p}_b \quad (1)$$

where t is the time between update, and $\vec{a} = [0 \ -g \ 0]^T$ with $g = 9.8 \text{ m/s}^2$ to provide acceleration in the vertical direction due to gravity. To simplify the calculation, the velocity of ball in each time interval between two frames is regarded as constant, and is given by

$$\vec{v}_b = \vec{v}_o + \frac{\vec{a} t}{2} \quad (2)$$

2) Collision Detection

In a table tennis game, the ball collides with three kinds of objects at least: table, net and racket. The collisions with table and net are treated as the situation that a ball collides with a fixed stationary racket.

All the collision events are simulated as a particle colliding with a flat plane. The collision between a ball and the stationary racket is illustrated in Fig. 7, where the particle is at the centre of the ball, which flies from \vec{p}_b to a collision point \vec{p}_c . The racket is simplified as a disc with a thickness d_{thick} , and its centre is at \vec{p}_r . Since the vertical distance from any potential collision point to the red surface of the racket is equal to the radius of the ball r_b , the centre of the collision plane is given by Eqn.3.

$$\vec{p}_p = \vec{p}_r + (r_b + d_{thick}) \vec{n}_{red} \quad (3)$$

where \vec{n}_{red} is the normal of the red surface. Since a racket has two potential collision planes (with two normals of \vec{n}_{red} and \vec{n}_{black}), a collision plane is considered as correct if it satisfies

$$\vec{v}_b \cdot \vec{n} < 0 \quad (4)$$

where \vec{v}_b is the velocity of the ball, and \vec{n} is the normal of the corresponding collision plane.

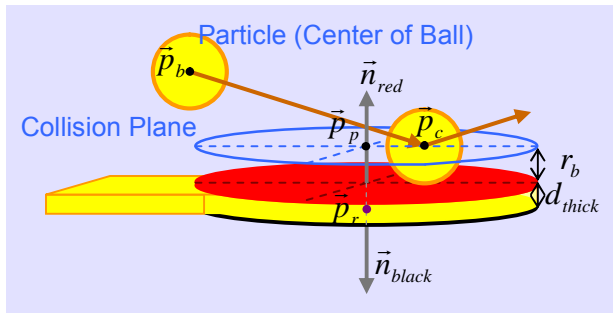


Figure 7. Particle and Collision Plane

If both ball and racket are moving, the position of the ball is transformed into a moving coordinate system with respect to the racket for collision detection. This is illustrated in Fig.

8, where a ball flies from \vec{p}_b to \vec{p}_b' in two consecutive frames with velocity \vec{v}_b if no collision occurs, and a racket moves from \vec{p}_p to \vec{p}_p' with velocity \vec{v}_r at the same time. To detect the collision, the predicted position \vec{p}_b' is shifted to \vec{p}_b'' by the negative movement of the racket. This transforms the moving racket into a static one with respect to the ball flying from \vec{p}_b to \vec{p}_b'' with velocity \vec{v}_b'' , with \vec{v}_b'' and \vec{p}_b'' given by

$$\vec{v}_b'' = \vec{v}_b - \vec{v}_r \quad \vec{p}_b'' = \vec{v}_b t + \vec{p}_b \quad (5)$$

If there is an intersection between the flying path and racket, such as \vec{p}_c shown in Fig. 8, a collision occurs. Let r_r be the radius of the racket, d_1 and d_2 be the displacements to the collision plane from \vec{p}_p and \vec{p}_b'' respectively, the conditions for a collision to occur can be expressed as

$$d_1 d_2 \leq 0 \quad \text{and} \quad |\vec{p}_c' - \vec{p}_p| \leq r_r \quad (6)$$

where

$$d_1 = [\vec{p}_b - \vec{p}_p] \cdot \vec{n} \quad d_2 = [\vec{p}_b - \vec{p}_p] \cdot \vec{n}$$

$$\vec{p}_c' = \left(\frac{|d_1|}{|d_1| + |d_2|} \right) \vec{v}_b' t + \vec{p}_b$$

the real collision point \vec{p}_c is given by

$$\vec{p}_c = \left(\frac{|d_1|}{|d_1| + |d_2|} \right) \vec{v}_b t + \vec{p}_b \quad (7)$$

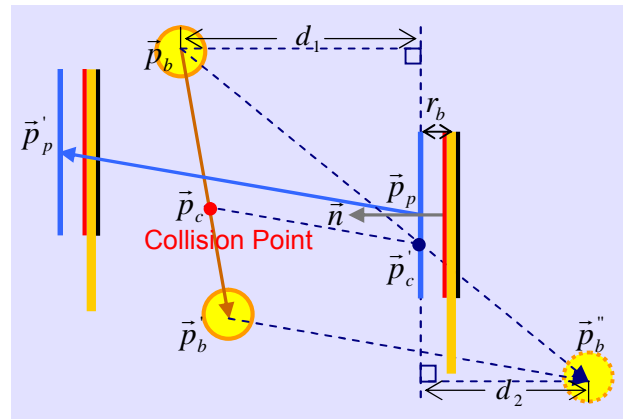


Figure 8. Ball-Racket Collision Detection

3) Collision Response

Collision response follows collision detection. As illustrated in Fig. 9 for a collision between a ball and a racket, if the racket is stationary and there is no bounce damping, the

ball flying into the racket from \vec{p}_b with velocity \vec{v}_b will move to \vec{p}_b' with velocity \vec{v}_b' after collision. Since the angle of incidence is equal to the angle of reflection, \vec{v}_b' is given by

$$\vec{v}_b' = -2(\vec{v}_b \cdot \vec{n})\vec{n} + \vec{v}_b \quad (8)$$

If the racket is moving and the friction effect (spin) is ignored for simplicity, the velocity of the racket \vec{v}_r only affects the ball along \vec{n} as \vec{v}_r' if the racket goes forward to the ball ($\vec{v}_r \cdot \vec{n} \geq 0$). In this case, the ball will move towards \vec{p}_b'' with velocity \vec{v}_b'' after collision, with \vec{v}_b' and \vec{p}_b' given by.

$$\vec{v}_b'' = f_r [\vec{v}_b' + (\vec{v}_r \cdot \vec{n})\vec{n}] \quad (0 \leq f_r \leq 1) \quad (9)$$

$$\vec{p}_b'' = \vec{v}_b'' t \quad (10)$$

where f_r is the bounce damping of the racket. If the racket moves backward, its velocity does not act on the ball movement.

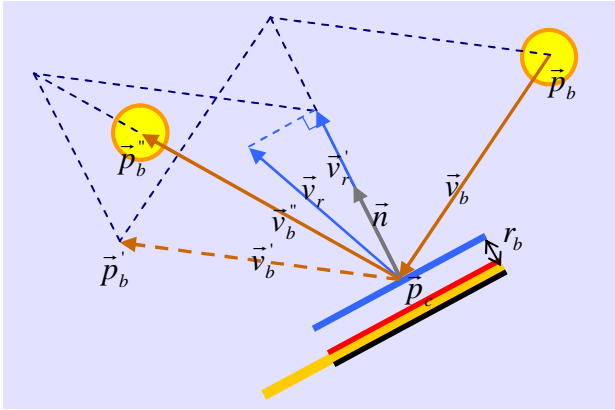


Figure 9. Ball-Racket Collision Response

Since collisions may occur more than once in a time interval between two consecutive frames, the next potential collision point need to be calculated subsequently after the first collision is detected. Therefore, if there is another collision occurs before the ball arrives \vec{p}_b'' , the trajectory of the ball will be changed again.

C. Stereoscopic Rendering

OpenGL supports stereo pair display, which is rendered by left and right buffers. The asymmetric frustum parallel projection method is adopted. Fig. 10 illustrates the frustum settings for the right eye, where the frustum is first translated from the tracked eye position (middle of two eyes) to right side by half of intraocular distance d_{eye} , then shifted to match the viewing screen. With $d_{eye} = 0.06m$, $d_{near} = 0.1m$, and $d_{screen} = 2.37m$, the Frustum Shift parameter d_{shift} that describes the degree of asymmetry is given by

$$d_{shift} = \left(\frac{d_{eye}}{2} \right) \left(\frac{d_{near}}{d_{screen}} \right) = 0.00127m \quad (11)$$

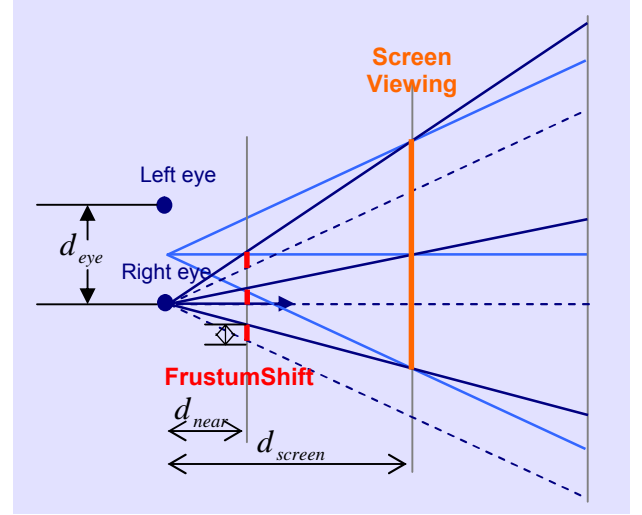


Figure 10. Frustum Setting for Right Eye

D. User Interface

As shown in Fig. 11, a simple menu is provided with three buttons of 'Training', 'Start Game' and 'Quit', which can be selected by a virtual stick tracked by wireless wand. The training mode allows a player to adjust the distance between the virtual racket floating in space and his/her actual hand holding the wireless wand, and to play with a ball served by computer that is triggered by pressing the button on the wireless wand. Selection of 'Start Game' activates the game for two players.



Figure 11. User Menu with Virtual Stick

V. SYSTEM PERFORMANCE

As an example to show the immersive table tennis game in operation, Fig. 12 shows two players standing in front of their own stereoscopic screen and playing interactively with each other through each other's avatar in a shared virtual environment.

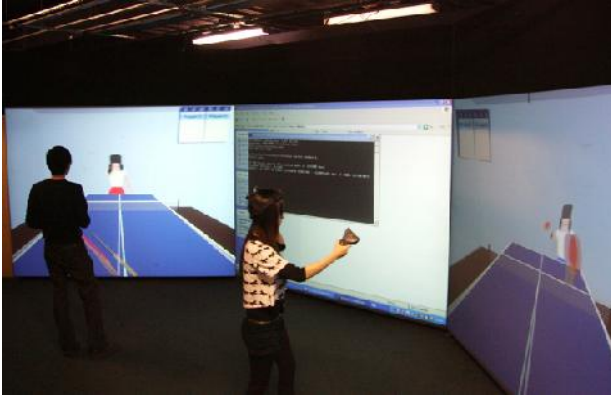


Figure 12. Two Players Play against Each Other

From the perspective of static object visualisation, each player can physically move around in front of the display screen and see a correct view of the 3D virtual table tennis table with a depth impression.

From the perspective of dynamic object visualisation, each player is able to see the position and orientation of his/her virtual racket floating in space at the near side of the virtual table tennis table, but also the position and orientation of the opponent virtual racket at the other side of the virtual table tennis table. The move made by one player will result in a corresponding move of the position of his/her avatar being displayed in the other screen. Although the movement of avatar is not realistic, it does provide a good indication of the opponent position. Furthermore, each player is able to see the virtual ball flying into/out of their own screen as well as hear a colliding sound when the virtual ball hits the virtual table and rackets.

From the perspective of interaction, the simplicity and intuitiveness of the game were seen to enable a new player to control the virtual racket to hit the virtual ball quickly by just holding and waving the wireless wand. Adjustment of the virtual racket position with respect to the actual hand position is found to be a good feature, as each player has each own preferred distance to hit the virtual ball.

The performance of the system in terms of speed was investigated. By recording the time taken for executing every 100 cycles, the execution time for each cycle was found to vary between 16.56ms and 17.04ms with an average of 16.77ms. Hence, the system is capable to provide an updating rate of 58.69 frames/s in the worst case.

CONCLUSIONS

The paper presents a new and unique way for two players to engage in immersive play of the table tennis game in real-time. The hardware development is based on integration of a high speed wireless tracking system, two large rear-projection stereoscopic screens and three computers running

in a client-server mode. The software development involves the use of simplified physical laws to model ball movement and collision. The system is seen to offer both players good visual and audio effects with physical interaction. These effects and physical interaction will be further investigated by running a competition and questionnaire based user evaluation. The system is viewed to provide not only a good platform for many possible improvements, such as more realistic avatars with full-body tracking and more realistic ball movement by including other physical effects like spinning, but also a useful basis for developing various cooperative or competitive virtual reality applications.

REFERENCES

- [1] Kay M. Stanney. Handbook of Virtual Environments: Design, Implementation, and Applications. Lawrence Erlbaum Associates. ISBN-10: 080583270X. ISBN-13: 9780805832709. 2002.
- [2] D. Wormell, E. Foxlin. Advancements in 3D Interactive Devices for Virtual Environments. *Proceedings of the workshop on Virtual environments 2003*. ISBN: 1-58113-686-2. 47-56. 2003.
- [3] Charles Woodward, Petri Honkamaa, Jani Jäppinen and Esa-Pekka Pyökkimies. Camball: augmented virtual table tennis with real rackets. *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*. ISBN: 1-58113-882-2. Vol.74. 275-276. 2004.
- [4] Mika Hakkarainen, Charles Woodward. Symball: camera driven table tennis for mobile phones. *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*. ISBN: 1-59593-110-4. Vol.265. 391-392. 2005.
- [5] Anders Henrysson, Mark Billingham, Mark Ollila. Face to Face Collaborative AR on Mobile Phones. *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*. ISBN: 0-7695-2459-1. 80-89. 2005.
- [6] http://www.nintendo.co.uk/NOE/en_GB/wii_54.html, Nintendo of UK.
- [7] Stephan Rusdorf, Guido Brunnett, Mario Lorenz and Tobias Winkler. Real-Time Interaction with a Humanoid Avatar in an Immersive Table Tennis Simulation. *Transactions on Visualization and Computer Graphics*. IEEE. Vol.13. Issue1. 15-25. 2007.
- [8] Jong-Seung Park, TaeYong Kim and Jong-Hyun Yoon. AR Table Tennis: A Video-Based Augmented Reality Sports Game. *Advances in Artificial Reality and Tele-Existence*. Springer Berlin/Heidelberg. ISBN: 978-3-540-49776-9. Vol.4282/2006. 197-206. 2006.
- [9] Tom Molet, Amaury Aubel, Tolga Çapın, Stéphane Carion, Elwin Lee, Nadia Magnenat-Thalmann, Hansrudi Noser, Igor Pandzic, Gaël Sannier, Daniel Thalmann. Anyone for Tennis? *Teleoperators and Virtual Environments*. Vol.8. Issue.2. 140-156. 1999.
- [10] Mason Woo, Jackie Neider, Tom Davis, Dave Shreiner. OpenGL Programming Guide: The Official Guide to Learning OpenGL(R). Version 2. Addison-Wesley Professional. Edition 5th. ISBN: 0321335732. 2005.
- [11] Hansrudi Noser, Christian Stern, Peter Stucki, Daniel Thalmann. Generic 3D Ball Animation Model for Networked Interactive VR Environments. *Proceedings of the Second International Conference on Virtual Worlds*. ISBN: 3-540-67707-0. Vol.1834. 77-90. 2000.