

3D interaction design and application development

Lim, Kian Teck

2008

Lim, K. T. (2008). 3D interaction design and application development. Master' s thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/13495>

<https://doi.org/10.32657/10356/13495>



**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**3D INTERACTION DESIGN AND APPLICATION
DEVELOPMENT**

**LIM KIAN TECK
SCHOOL OF MECHANICAL AND AEROSPACE
ENGINEERING
2008**

3D INTERACTION DESIGN AND APPLICATION DEVELOPMENT

LIM KIAN TECK

School of Mechanical and Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Master of Engineering

2008

Abstract

Human-computer interaction (HCI) is an area concerned with the design, evaluation and implementation of interactive computing systems for human use. A 3D environment makes use of the HCI to provide visual and tactile interaction to the user. However the HCI hardware devices and software development are expensive. Hence it is essential to explore an affordable solution for HCI that can be implemented for educational and research uses.

The interface should be easy to use, easy to implement, reusable for other program and less costly to develop. The scope of this research covers hardware and software interfacing of various commercially available input devices into the different applications for evaluation.

The proposed research used inexpensive commercially available devices such as steering wheel device and P5 data game glove to provide the visual and tactile interface for the 3D environment. The objective of this research is to develop a reusable and inexpensive interactive interfacing solution for the 3D environment.

This interfacing solution includes a 3D interaction module that is developed to integrate different game devices into the 3D environment. It allows different applications to interface with the commercial game devices easily and the integration is straightforward and cost effective. The research concluded with three applications that were developed using the 3D interaction module.

Acknowledgement

The author would like to express his heartfelt gratitude and appreciation to:

Associate Professor Dr CAI YIYU, Research Supervisor for his invaluable guidance, in-depth advice, help and encouragement, not only in the understanding of research, but also in the development of it.

Dr WAN HUAGEN for his Protein Molecules Visualization Software used in one of the application that was done in the research.

The author would also like to thanks the following people for their precious help throughout the research:

1. Dr FAN ZHAOWEI (Research staff)
2. LU BAIFANG (Research staff)
3. WONG CHING HO (Research staff)
4. GUAN YUNQING (Research student)
5. SU LI (Research student)
6. Technical Executives staff of the DRC Lab

Last but not least, the author would like to express his most sincere gratitude to all those who have offered their gracious assistance and support in one way or another to make this research a success, but have been left out involuntarily.

Table of Contents

Abstract.....	i
Acknowledgement.....	ii
Table of Contents	iii
List of Figures	v
List of Tables.....	vii
1. Introduction	1
1.1. Objective.....	1
1.2. Scope	1
1.3. Organization of the Thesis.....	2
2. Literature Review	4
2.1. Human-Computer Interaction Technology.....	4
2.2. Hardware Interfacing Technology	7
2.3. Software Interfacing Technology	9
3. 3D Interfacing Evaluation	13
3.1. Desktop input devices	14
3.1.1. Keyboard.....	14
3.1.2. Mouse	17
3.2. Tracking devices	22
3.3. Direct human input.....	24
3.4. Glove Technology.....	25
3.4.1. Driven Technology	27
3.4.2. Glove Comparisons	30
3.5. 3D Visual Interfacing Device	36
3.5.1. Stereo Viewing.....	36
3.6. Interfacing Software.....	39
3.6.1. DirectX	39
4. Game Device	41
4.1. Joystick Device	41
4.2. Game-pad Device.....	42
4.3. Steering Wheel Device.....	43
4.4. P5 Glove Device	44
5. 3D Interaction and System Design	47
5.1. 3D Interaction	47
6. Tactile Interface Development	57
6.1. Graphic Manipulation and Visual Interaction	57
6.2. Implementation	60
6.3. Evaluation.....	69
7. VR-enhanced Bio Edutainment Application.....	72
7.1. VR-enhanced Bio Edutainment Technology	72
7.2. Implementation	73
7.3. Evaluation.....	74
8. Collaborative Game	77
8.1. Game design	77
8.2. Implementation	80
8.3. Evaluation.....	80
9. Conclusions and Future work.....	81
9.1. Conclusions	81
9.2. Contribution.....	83

9.3. Future work.....	85
References.....	86

List of Figures

Figure 1: The Path from Human to Computer	8
Figure 2: Window System Architecture	11
Figure 3: QWERTY keyboard	14
Figure 4: Alphabetic keyboard	15
Figure 5: DVORAK keyboard	15
Figure 6: Chord keyboard	16
Figure 7: Left-handed keyboard	16
Figure 8: Mouse	18
Figure 9: Wireless mouse, Optical mouse and Left-handed mouse	19
Figure 10: Footmouse	19
Figure 11: Trackball	20
Figure 12: 3D Infra-red mouse	20
Figure 13: Digital Data Entry Glove	26
Figure 14: Fibre optics glove	27
Figure 15: Ultrasonic glove	28
Figure 16: Magnetic glove	29
Figure 17: Electrical resistance glove	29
Figure 18: DataGlove	31
Figure 19: Hand Master	32
Figure 20: PowerGlove	32
Figure 21: CyberGlove	33
Figure 22: PINCH Glove	34
Figure 23: 5DT DataGlove	35
Figure 24: LCD shutter glasses	38
Figure 25: Anaglyph glasses	38
Figure 26: Joystick	41
Figure 27: Game-pad	42
Figure 28: Steering Wheel	43
Figure 29: P5 glove	44
Figure 30: P5 Glove Unit	45
Figure 31: P5 Glove Unit – Close Up	45
Figure 32: 3D Interaction design architecture	47
Figure 33: 3D Interaction integration diagram	53
Figure 34: 3DInteraction class diagram	54
Figure 35: Wireframe mode	58
Figure 36: Stick mode	58
Figure 37: Ball-and-stick mode	58
Figure 38: Sphere mode	59
Figure 39: Ribbon mode	59
Figure 40: Tactile interface development design architecture	60
Figure 41: Steering wheel interface	62
Figure 42: A ride through the ribbon structure of protein molecules	62
Figure 43: The sequence through the ribbon structure of protein molecules	63
Figure 44: Steering wheel interface for molecule navigation	63
Figure 45: Tactile protein interaction	64
Figure 46: Virtual hand represented by the cone	65
Figure 47: Cone approaches an atom of protein molecules	66

Figure 48: Select an atom	66
Figure 49: Pick the atom.....	67
Figure 50: Pick command	67
Figure 51: Move the atom.....	68
Figure 52: Pick the bond and rotate.....	68
Figure 53: Move command and Rotation command	69
Figure 54: Protein surface interaction using P5	70
Figure 55: The core bio edutainment technologies and supporting technologies...	72
Figure 56: Game devices used in the bio games	74
Figure 57: Online virtual reality game design	77
Figure 58: Main GUI form.....	78
Figure 59: Dual Player mode GUI	79
Figure 60: Single Player mode GUI	79

List of Tables

Table 1: Application layer interface	48
Table 2: 3D interaction common data interface	49
Table 3: 3D interaction common data	49
Table 4: Hardware device manager interface	49
Table 5: Data captured by commercial game devices	50
Table 6: Common game device interface	50
Table 7: Common game device.....	50
Table 8: Data captured by P5 glove	51
Table 9: P5 glove interface	51
Table 10: P5 glove data	52
Table 11: Tactile interface development hardware cost	70
Table 12: Tactile interface development cost comparison	71
Table 13: VR-enhanced Bio edutainment hardware cost	75
Table 14: VR-enhanced Bio edutainment cost comparison.....	76
Table 15: Collaborative game hardware cost.....	80
Table 16: Collaborative game cost comparison	80

1. Introduction

Human-computer interaction (HCI) is the study of interaction between computing systems and human. It is an area involves with the design, evaluation and implementation of interactive computing systems for human use, which includes both software and hardware. A 3D environment makes use of the HCI to provide visual and tactile interaction to the user. However the HCI hardware devices and software development are expensive. Hence it is essential to explore an affordable solution for HCI that can be implemented for educational and research uses.

1.1. Objective

The purpose of this research is to develop a reusable and inexpensive tactile interfacing solution for 3D environment. It is achieved by the following processes:

1. To investigate the interactive interfacing game devices
2. To study the 3D interfacing environment
3. To design and develop an interactive solution
4. To apply the interactive solution for bio-edutainment

1.2. Scope

The scope of this research covers the hardware and software interfacing in a 3D environment, the evaluation of commercially available input devices and the development of the 3D interface adopting the Object-Oriented C++ programming language. The programming tool used in the research was the Microsoft Visual Studio 6 with service pack 2 under the Window environment.

All the hardware cost listed is based on information gather during the research being carried out and for this research only. It cannot be use as the purchase cost of the hardware devices.

1.3. Organization of the Thesis

The rest of the thesis is organized into the following chapters.

Chapter two describes the literature directly or indirectly related to the research. It is a concise review done by the author in three different areas of study. They are the human-computer interaction technology, the hardware interfacing technology, and the software interfacing technology.

Chapter three evaluates the 3D interfacing. This includes the desktop input devices, tracking devices, direct human input, glove technologies, 3D visual interfacing device and interfacing software.

Chapter four explains commercial game devices that the author studied during the research. It evaluates the commercial available game devices. Each device is given a brief explanation and an assessment of its pros and cons.

Chapter five discusses the 3D interactive interfacing and system design. The 3D interaction is a special module designed to integrate different game devices into the 3D environment. The chapter proceeds with the 3D interaction design architecture and the integration diagram.

Chapter six illustrates an application of the 3D interaction module on the development of the tactile interface. It utilizes game devices to carry out graphic manipulation and interaction.

Chapter seven presents a VR-enhanced bio edutainment application that used the 3D interaction module. It is an immersive VR protein structure learning environment.

Chapter eight covers a collaborative game application. It is an interactive network guessing game played by two persons.

Chapter nine concludes the research, highlights the contributions and proposes the future works of the research.

2. Literature Review

2.1. Human-Computer Interaction Technology

Historical development of major advances in human-computer interaction (HCI) technology in United State benefited from research at both corporate research labs and universities, much of it funded by its government [20]. HCI covered the design, evaluation and implementation of interactive computing systems for human use and with the understanding of major phenomena surrounding them [7, 8]. It is the study of the way human operates computer in order to create computer system that best serves the needs of the human [1]. The studies included both the machine side and the human side, but of a smaller group of devices. On the machine side, the scope included techniques in computer graphics, operating systems, programming languages, and development environments. On the human side, communication theory, graphic and industrial design disciplines, linguistics, social sciences, cognitive psychology, and human performance were covered [3]. However, the main focus was on interaction between one or more humans and one or more computational machines.

Human-computer interaction is the aspect of the science, engineering, and design. It is concerned with the combined performance of tasks by humans and machines; the structure of communication between human and machine; the human capabilities to use machines including the learning ability of the interfaces; the algorithms and programming of the interface itself; the engineering concerns that arise in designing and building interfaces; the process of specification, design, and implementation of interfaces; and design trade-offs [3]. It is a multidisciplinary field arose from the different interrelated fields in computer graphics, operating systems, human factors, ergonomics, industrial engineering, cognitive psychology, and the systems part of computer science.

Computer graphics has a natural interest in HCI as "interactive graphics" [3]. It is a branch of science and technology concerned with methods and techniques for converting data to or from visual presentation using computers. This was led by the development of several human-computer interaction techniques that essentially marked the beginning of computer graphics as a discipline.

Operating systems is a discipline that developed techniques for interfacing input or output devices, for tuning system response time to human interaction times, for multiprocessing, and for supporting windowing environments and animation [3]. This strand of development has currently given rise to "user interface management systems" and "user interface toolkits".

Human factor as a discipline deals with the problems of human operation of computers, the substantial cognitive, the communication, and the interaction aspects [3]. It is the study of how humans behave physically and psychologically in relation to particular environments, products, or services. Ergonomics is similar to human factors, but it arose from the studies of work. It is the study of people in work environment [1]. Human interaction with computers is also a natural topic for ergonomics, but again, a cognitive extension to the field was necessary resulting in the current "cognitive ergonomics" and "cognitive engineering". Cognitive ergonomics is the study of the mental activities of people in the work environment [1].

Industrial engineering is used to incorporate computers to fit into the larger design of work methods [3]. It arose out of attempts to raise industrial productivity starting in the early years of 20th century.

Cognitive psychology is an area that concentrates on the learning of systems, the transfer of that learning, the mental representation of systems by humans, and human performance on such systems [3].

Usability is the central to the HCI since the whole point of interface design is to produce systems that are easy to learn [5]. Usability refers to the degree to which a computer system is effectively used by its users in the performance of tasks [6]. This is to allow users to work efficiently, effectively and comfortably.

One of basic principles of HCI is the GOMS Model [7]. A GOMS model refers to Goals, Operations, Methods, and Selection (GOMS) Models. It includes Methods needed to achieve specified Goals. The Methods are a sequence of steps consisting of Operators that the user performs. If more than one Method can accomplish a Goal, then the GOMS model uses Selection Rules to select the best Method depending on the context [10].

The literature of visual and graphic perception in HCI shows that the visual perception and how the specifics of human visual perception may affect it. In visual perception, how light transmits information to the eye of the perceiver, how that information is processed, and finally how that information results in conscious experience of the external world are studied [4]. The main focus of the psychology of visual perception is the perceiver being a processor of information [12]. Visual perception is underutilized by the graphical user interfaces. Graphical user interfaces locate information faster than with natural language queries, and user gain higher comprehension and satisfaction from these interfaces [11]. There is also indication to support that humans recall pictures better than words [9].

Animation is movements of either text or graphics on the computer screen [4]. It is the use of graphic art occurring over time [13]. There are many specific uses for animation including reviewing, identification of an application, emphasizing transitions to orient the user, to provide choices in complex menus, to demonstrate actions, to provide clear explanations, to give feedback on computer status, to show history of navigation, to provide guidance when a user needs help [13]. Basically, animation is very effective in establishing mood, in increasing sense of identification in the user, for persuasion, and for explication [13, 15]. However in HCI,

animation is seldom applied due to the fact that few studies providing clear evidence of the positive affects of animation for the user [14].

The visualization and creativity are implemented using artificial intelligence in HCI. Visualization is supporting creative work by providing users the relevant information, and identifying patterns [16]. The importance of creativity in computer assistance includes constructing meaningful overviews, zooming in on desired items, filtering out undesired items, and showing relationships among items [16]. In scientific area, understanding of complicated data can be helpful using visualization through both simple and advanced computer graphics [17].

Simulation is a main part of HCI. It can assist learning to be more effectively, through learning by doing. It is also useful where actual environments are expensive and impractical to recreate constantly [18]. Simulation is effective because it can create a context for learning [18]. The two limiting factors for using computer simulations in teaching are the difficulty of creating good simulations and the difficulty of learners using simulations by themselves [18].

The above disciplines and literatures are not the exhaustive in the HCI, but each of them has made significant contribution to it.

2.2. Hardware Interfacing Technology

Computer interface design is a subset of HCI and focuses specifically on the computer input and output devices such as the screen, keyboard, and mouse [4]. Input device requires input or data acquisition, while output device is a peripheral through which information from the computer is communicated to the outside world.

The input or data acquisition is the way in which information about the user is conveyed to the computer [2]. Figure 1 shows how information from the

human is passed to the computer. This involves three processes: sensors, signal conditioning, and data acquisition. The design of these systems will determine how intuitive, appropriate, and reliable the interaction is between human and computer.

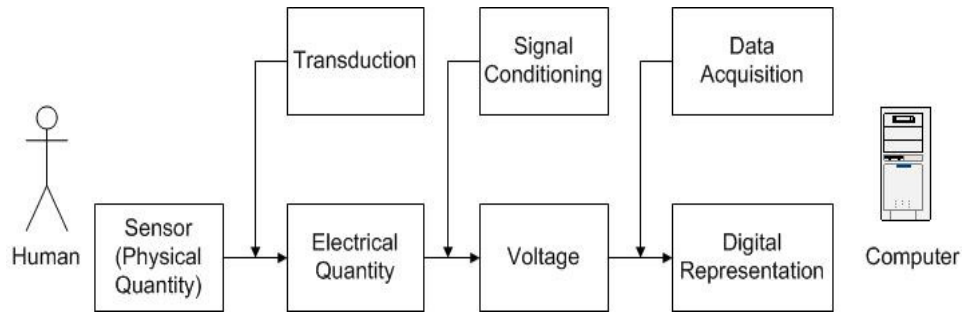


Figure 1: The Path from Human to Computer

There are different classifications of sensors. They are grouped by the underlying physics of their operation, by the particular phenomenon they measure and by a particular application [2]. However there is no method of categorization being clearly superior to any other. Different approach can be acquired to model computer sensor. The most common one is to model computer sensing after the five human senses: gustatory (taste), olfactory (smell), tactile, auditory, and visual [2]. Another method is to decide what volitional or even non-volitional actions of the user will be important for the particular computer application [2]. Before determining what computer inputs to use, it is important to decide what human outputs are appropriate for the application and determine what sensor is optimal in measuring it.

Signal conditioning converts the signal from the sensors output to an appropriate form for the input of the data acquisition system. In most applications this means changing the output to a voltage if it is not already, modifying the sensors dynamic range to maximize the accuracy of the data acquisition system, removing unwanted signals, limiting the sensor's spectrum and analogue signal processing both linear and nonlinear [2]. Therefore, signal conditioning system is critical in correct mapping of the sensor output to the data acquisition input.

Data acquisition uses sampling and quantization techniques to convert analogue and continuous time signals into the binary form that a computer can understand. In human gestures to computer interfacing, data acquisition requires an additional signal conditioning circuitry to modify the signal that measures by the sensor, before it is input into the computer.

The Visual Display Units (VDUs) are the predominant output device in use today. It is also known as the computer screen. Other output devices are also available to allow computer to communicate to the outside world. They include CD-ROM device, printer, video, virtual reality headset, sound system, microfilm, microfiche and various types of electronic outputs [5].

2.3. Software Interfacing Technology

The history of User Interfacing software begins with Batch-processing, which had no interactive capabilities. All the user inputs were specified in advance (punch cards, etc.) and all system outputs were collected at end of program run (printouts, etc.). Therefore the applications had no user interface component to distinguish from the file Input/Output (I/O).

The next user interfacing development came from Time-sharing Systems. It was a command-line based interaction with simple terminals. It provided shorter turnaround (per-line), but similar program structure with Batch-processing. The applications read arguments from the command line, return results. The Full-screen textual interfaces offered better interaction with even shorter turnaround (per-character) that gave a "real-time" feeling. The applications received User Interface input and reacted immediately in main "loop". Following this was the development of the Menu-based systems. It explored the advantageous "Read & Select" method over "Memorize & Type" but was still text-based. The applications had explicit User Interface component but the choices were limited to certain menu items, one at a

time for hierarchical selection. The application still had control of the program.

Then came the Graphical User Interface Systems. It shifted from character generator to bitmap display user interfacing. Pointing devices were also introduced in addition to keyboard. Finally, with the Event-based program structure, it was the most dramatic paradigm shift for application development. User was in control of the program, the application only reacted to user (or system) events through call-back paradigm and event handling. Initially it was application-explicit, later it became system-implicit.

In general, there are two types of interaction styles recognized, the command language and the direct manipulation systems [4]. During the command language systems, users are communicating with the computer using text command. Direct manipulation systems are the graphic user interfaces (GUI) which is now common to users in the Windows environment [9].

The Window system's basic tasks are Input handling, Output handling and Window management. In Input handling, the Window system passes user input to appropriate application while the output handling visualises the application output in windows. Window management manages and provides user controls for windows.

Window System architectures is made up of 4-Layer Model, as shown in Figure 2. They are the User Interface Toolkit, Window Manager, Base Window System and Graphics & Event Library.

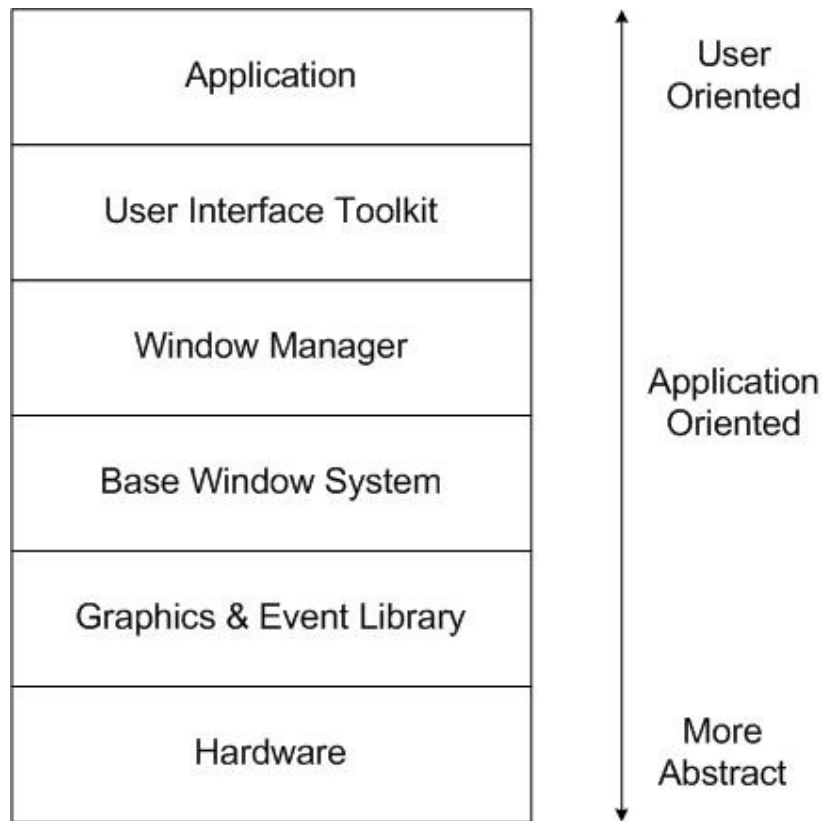


Figure 2: Window System Architecture

The User Interface Toolkit also known as the “Construction Set” offers standard user interface objects.

The Window Manager is used to implement user interface to window functions. It changes from system-centred to user-centred view of Window system. It also allows for the implementation of new variants of desktop metaphor without having to change entire system.

The Base Window System provides logical abstractions from physical resources (e.g., windows, mouse actions). It works with device-independent and Operating System-independent abstractions with only very general assumptions about Operation System. It supports system security and consistency through encapsulation and synchronization and offers basic API for higher levels.

The Graphics & Event Library implements graphics model. It is a high-performance graphics output functions for applications, register user input actions. It hides the hardware and Operating System aspects, offering virtual graphics or event machine.

3. 3D Interfacing Evaluation

Input devices allow users to communicate with the applications. Input devices are just physical tools that are used to implement various interaction techniques [25]. There are many different characteristics used to describe an input device, degree of freedom (DOF), input type and frequency of the data generated, device's physical interaction, and device's intended usage.

DOF describes the number of independent way an object moves in space. Device's DOF is used to specify the complexity of the device and the various interaction techniques that the device can accommodate.

Data generated by input device are classified either discrete, continuous, or combination of both. Discrete input devices produce single data value that is normally used to specify the mode or state of the application. Continuous input devices create multiple data values resulted from user's action. Combination of both components gives the device large variety interaction techniques to implement.

The device physical interaction describes the physical involvement required by the user in order to generate data. It can be purely active input device that requires user physical action or a purely passive input device that does not require any physical action for the device to capture data.

Devices can be grouped by their intended usage. A locator device is use to determine position and orientation, while a choice device is use to select a particular element of a set.

The following sections will evaluate different categories of input device:

1. Desktop input device
2. Tracking device
3. Direct human input

3.1. Desktop input devices

3.1.1. Keyboard

The keyboard is one of the most common input devices in use today. It is used for transferring textual data and commands into the computer. There are several different types of keyboard available in the market.



Figure 3: QWERTY keyboard

QWERTY keyboard

The QWERTY keyboard in Figure 3 was actually designed to slow typists. It was introduced to overcome the mechanical constraints of the manual typewriter. The QWERTY layout was developed to increase the spacing between common pairs of letters so that the sequentially struck type-bars would not jam. Although the electric typewriter and computer's keyboard are not subjected to this problem, the QWERTY keyboard is still the dominant layout. The main reasons are trained typists are reluctant to relearn their craft due to cultural acceptance and the management are not prepared to accept the initial inefficiency cause during relearn phase. There is also large cost involved in the replacement.



Figure 4: Alphabetic keyboard

Alphabetic keyboard

The alphabetic keyboard in Figure 4 simply arranges letters in alphabetical order. This design is to ease inexperienced users to search the letter they want. However, the design fails to consider of the fact that some letters such as e and s, are more frequently being used as compared to x and z. The alphabetic keyboard also offers no speed advantages to touch typists.

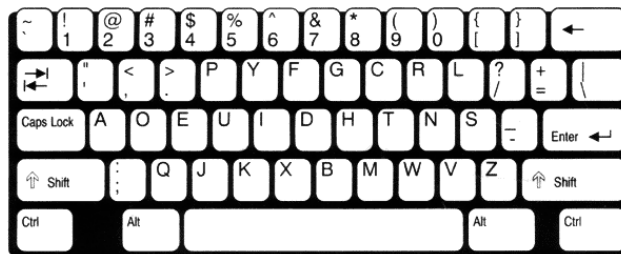


Figure 5: DVORAK keyboard

DVORAK keyboard

The DVORAK keyboard in Figure 5 uses a similar layout of keys to the QWERTY system, but assigns the letters to different keys. Based upon an analysis of typing, the keyboard is designed for people achieved faster typing speeds. Typists using the DVORAK keyboard typically make fewer errors as compared with a QWERTY keyboard, and are 10% faster.

However the fail of the DVORAK keyboard are the need to retrain million of typists and replace millions of keyboards.



Figure 6: Chord keyboard

Chord keyboard

The Chord keyboard in Figure 6 is significantly different from the rest. It has only a few keys, 4 or 5. The letters are produced by pressing on or more of the keys at once. The keyboard is very compact, and suitable for one-handed operation. It takes only a few hours to learn and is capable of quite high-speed typing but is tiring to use compare to normal keyboard.



Figure 7: Left-handed keyboard

Other types of keyboard

There is the left-handed keyboard as shown in Figure 7 similar to QWERTY keyboard except for the cursor movement. The Home and End keys are moved from right to the left hand side, on the principal that left-handed users will prefer to use their left hand more than their right. However, there seems no evidence to the principal in practice, with left handed user of such keyboards still tending to automatically look to the right for the transposed keys.

There are also small keyboards which are approximately 50cm by 20cm. they are widely

Used by the laptop and sub-notebook computers.

A virtual keyboard is a keyboard that has no physical buttons for the user to key in the input. It is a wireless and Bluetooth enabled device that makes used of light positioning and detection devices to sense the intention of the user. A keyboard layout is projected on to any flat surface to allow the user to input command and the detection sense captured the simple movement or gesture. It has the advantages of being portable and no mechanical failure but it losses the feel of touch which is not suitable for the blind and typist.

Keyboards are the most common input device and the QWERTY keyboard is the preferred option in many of the keyboard designs. The reason for this is social. Some keyboards provide end-user feedback. The keyboards which do not provide feedback have the greatest effect on touch typists' performance. Evidence suggests that even allowing the typists to practice with such keyboard, their speed typically reduced by 20% and double error rates.

Keyboards are easy to integrate into the application. It is one of the basic input devices found in the operation system (OS). OS normally makes use of interrupt mechanism to obtain user input. Programs an application responding to keyboards input is done through interfacing with the input/output (I/O) layer of the OS. A key of the keyboard is represented by 1 byte data for non-Unicode or 2 bytes data for Unicode.

3.1.2. Mouse

The mouse in Figure 8 is one of the major input devices found in the computer systems. In its most basic forms, the mouse is a small, palm-sized box housing a weighted ball. As the device moves around on a flat surface, the weighted ball transmits information about the movement of the mouse to the computer via a wire attached to it. This relative motion synchronizes to the movement of a pointer on

the computer's screen, called the cursor. The whole arrangement tends to look rodent-like; hence the term 'mouse' is used for the device. In addition to detecting motion, the mouse has typically one, two, or three buttons on top. They use to convey the information like selection or to initiate action pointed to the innards of the computer. The figure below shows a typical mouse device.



Figure 8: Mouse

The mouse operates in a planar fashion, moving around the desktop, and is an indirect input device, since a transformation is required to map from the horizontal nature of the desktop to the vertical alignment of the screen. This type of mouse is also called 2D mouse. A direct mapped for left-right motion but the up-down motion on the screen is mapped to the away-toward motion relatively to the user.

The mouse only provides relative information of the motion of the ball within the housing so it can be physically lifted up from the desktop and replaced in a different position without moving the cursor. This advantage provides less space requirement but suffers from being less intuitive for novice users. However, the user still needs a minimum room on the desktop to move the mouse around. This confined space is usually accomplished via the use of mouse pad. As the mouse operates on the desk, moving it about is easy and the user suffers little arm fatigue, even since the indirect nature of the device can lead to problem with hand-eye coordination. Another plus point of the mouse is that the cursor itself is small, and it can be easily manipulated without obscuring the display.



Figure 9: Wireless mouse, Optical mouse and Left-handed mouse

There are different types of mouse available in the market as shown in Figure 9. Some mouse omit the wire attached to the computer, instead use radio frequency to transmit information to the computer. This type of mouse is classified as wireless mouse. Another type is optical mouse. It works differently from mechanical mouse. The optical mouse replaces the weighted ball by a light-emitting diode that emits a weak red light. This is reflected off and the fluctuations in reflected intensity as the mouse moves are recorded by a sensor in the base of the mouse and translated into relative x and y motion. The optical mouse is less susceptible to dust and dirt, and its mechanism is less likely to become sticky. Some manufacturers also produce left-handed mouse with moulding being higher on one side than the other.



Figure 10: Footmouse

Not all mouse are hand operated. There are mouse-like devices that operated by the foot which is known as Foot-mouse shown in Figure 10. Tilting the Foot-mouse left or right or up or down moves the cursor in the appropriate direction. Just as using the feet in driving a car. Using Foot-mouse in a computer leaves both hands free for other tasks which provide better integration between mouse device and keyboard devices.



Figure 11: Trackball

The trackball in Figure 11 is a little like an upside-down mouse. It is composed of a fixed housing holding a ball that can be rotated freely in any direction by the fingertips. Movement of the trackball is detected by optical or shaft encoders. This in turn generates output which is used to determine the movement of the display cursor. Because of the design of the trackball, it requires no additional space in which to operate, and is therefore a very compact device. Another advantage of using the trackball is that it is more comfortable when using it for extended periods of time because the user can rest the forearm, keep the hand in one place, and spin and stop the trackball with his fingers. The trackball also provides direct tactile feedback from the ball's rotation and speed. The main disadvantage using the trackball is the limitation usage for drawing tasks. It cannot be used to trace drawings or to handprint characters since the trackball is a rotational device that operates only in relative mode.



Figure 12: 3D Infra-red mouse

3D mouse either handheld or user-worn are input devices consisting of motion tracking and physical interaction component. It is differentiated from the 2D mouse by providing position and orientation information while moving in the 3D space.

Computers have become powerful with three-dimensional (3D) graphic capabilities that allow new applications to incorporate 3D interactivity. However, there are few low cost 3D input devices available for the desktop interactive environment [24]. Interactive systems that use a 3D world should ideally be complimented with a 3D input device [24]. For the desktop systems there is a need for a matching development in input devices suited to the 3D applications beyond the simple 2D mouse [24].

The 3D mouse offers a low cost and a general-purpose function to computer users. It covers both 2D and 3D human-computer interaction spaces, and is aimed at the large market of both professional and non-professional users. There are different types of 3D mouse available in the market. Figure 12 shows a typical 3D infra-red mouse.

The 3D mouse can be used just like a standard mouse, with the same look and feel, for the general graphic user interface. However, when the 3D mouse moves across a 3D graphic object or application, it can perform intuitive, simultaneous 3D input control. The market for 3D interaction is evolving rapidly, and the 3D mouse provides the missing link for the 3D infrastructure of modern computer interfaces.

The 3D mouse is perfect for Internet navigation application. It allows the user to surf and browse web pages just like a standard 2D mouse. At the same time, when 3D graphics such as the 3D Web are present, the 3D mouse readily enables the user to perform 3D direct manipulation. It also provides more fun for computer games. With the 3D mouse, the user can play the 3D games that are currently played only by special 3D joysticks. The 3D mouse alone allows the user to play the games that usually require a player to use both a standard mouse and a keyboard. Last but not least, the 3D mouse meets the long-felt need to provide an affordable 3D input device for professional 3D applications such as

CAD/CAM, animation, virtual reality, information visualization, and the direct generation of 3D graphics.

Mouse and trackball, like the keyboards, are the basic input devices of the OS. Acquiring data from them are simple and straightforward using the API provided by the I/O layer. The I/O layer bridges the application and the hardware devices by allowing application to access the device's driver. The mouse and trackball share the same data structure (X and Y coordinates) which represents the screen coordinate system.

3.2. Tracking devices

Tracking devices are purely passive and continuous data generating device. These devices usually involve no physical interaction from user in data capturing. The following paragraph gives a brief description of the tracking devices available.

1. Motion tracker
2. Eye tracker
3. Hand tracker

Motion Tracker

Motion tracker determines the position and orientation of the body movement. There are many different motion tracking technologies available. They are classified according to the method the data are being captured.

Magnetic tracking uses magnetic field to determine the position and orientation of the object. A magnetic transmitter and receiver works in pair to detect the relative 3D displacement of the object. The accuracy of the magnetic tracking is greatly influenced by conductive object present in the surrounding.

Mechanical tracking makes use of mechanical linkages and electromechanical transducers to track the object movement. The object is attached to the fixed base linkage. When the object moved, the transducers which mounted on the linkage

will measure the linkage displacement and orientation. Directly the object position and orientation is obtained. This method of tracking is more accurate and faster as compared to the other but it has its limitation in term of mobility.

Acoustic tracking utilizes high-frequency sound (ultrasonic) to track an object. It comprises of a sound source transmitter and microphone receiver. Either the transmitter or the receiver can be placed on the object while the other on the environment. The position is determined by computing the distance from 3 points using multiple receivers. The distance between source and a point is calculated by taking the time for the sound to travel from source to the receiver and multiply with the speed of sound. Acoustic tracking has its benefit of lightweight and low cost. However it is more subjected to external interference from the ambient noises. This greatly reduces its accuracy and popularity in usage.

Inertial tracking employs inertial measuring device to determine the derivative measurement of the object. Normally an integrator is used to resolve the capture data into position and orientation information. Accuracy and error accumulation are the few drawbacks to implement inertial tracking.

Optical tracking exploits the light domain to find out the position and orientation of the object. It can be an emitted or a reflected light being applied as a tracking media. The principle of this method is based on computer vision technique and optical sensor. The optical sensor captures the image and passes to the computer to process and determine the position and orientation of the object using computer vision algorithms. The shortcoming of this method is it tends to be subjected to occlusion which will affect the accuracy of the data capturing. Increasing the number of optical sensor will reduce the occlusion but it makes the tracking algorithm more complex to implement.

Hybrid tracking combines several tracking methods together to achieve higher accuracy and lowest latency during data capturing. Tracking devices implementing hybrid tracking are comparably much more complex to deal with.

Eye Tracker

Eye tracking makes use of computer vision technique to determine the direction the user is looking. The reflected light from the cornea is captured and calculated using complex algorithm to determine the eye position.

Hand Tracker

Hand tracker provides detailed tracking information of the human's hand. Both the hand movement and the fingers bending motion are captured by the data glove using different sensing techniques. The bend-sensing glove uses bend sensor, a device that detects the angle of bending, to detect the movement of the hand and its associated gesture. These sensors normally measure the joint angle and the data is processed to determine the hand movements and gestures. The bending sensor can be a light-based, resistive ink, strain-gauge and fiber-optic located on the glove or embedded in the glove. The pinch glove uses electrical contacts to determine the hand movements and gestures. Conductive materials are placed on the back of the glove along the fingers and thumb. These materials are typically light weight which is much preferred by most users as they do not feel tired when wearing the glove. Section 3.4 provides detailed description of the hand tracking devices.

Most of the tracking devices have their own device's driver written. Unlike the keyboards and mouse, the I/O layer of the OS did not provide a direct access to their driver. Therefore programming an application to use these devices is harder and complex. This involves interfacing with the hardware device's driver and processing the raw data.

3.3. Direct human input

Direct human input is the best interacting method to communicate with the application. Data are directly obtained from the human body and fetched into the application. In this case, the human body becomes the input device for the application. The speech, the bioelectric and brain input from the human body are

typical examples of direct human inputs. The speech input makes use of voice command to interact with the application. The bioelectric input reads the muscle nerve signals to communicate with the application. The brain input utilizes brain signal to command the computer.

Interfacing with direct human input can be expensive. Special input and output hardware device's driver and program are required to read the information. Developing application to use these devices is time consuming and application specific.

3.4. Glove Technology

The glove technology started in 1983 by Dr. G. Grimes, AT&T Bell Labs. He invented the Digital Data Entry Glove as shown in Figure 13. This was the first glove-like device (cloth) built with a numerous of touch, bend, and inertial sensors. The glove measured finger flexure, hand-orientation and wrist-position, and had tactile sensors at fingertips. The orientation of hand is tracked by video camera that required clear line-of-sight observation in order for the glove to function. The glove was designed to provide alternative to keyboard to match recognized gestures or hand orientations to specific characters, specifically to recognize the Single Hand Manual Alphabet for the American Deaf. The glove also contains special circuitry to recognize 80 unique combinations of sensor readings to output a subset of the 96 printable ASCII characters. This provides a tool to "finger-spell" words. The positions of sensors were changeable.

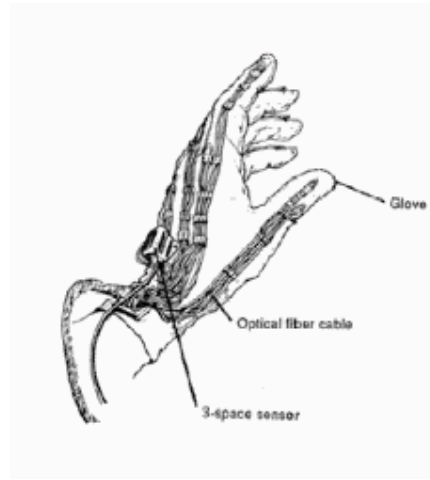


Figure 13: Digital Data Entry Glove

Being one of the mainstays of the new virtual reality systems, the glove is a multi-dimensional input device. Traditional data input devices have a limited range of the amount of data they can input at a given time because they are limited to one, two or three degrees of freedom. Degree of freedom (DOF) is a measure of the number of positions at which the device can be read as inputting a different data value. The glove offers a far superior data input potential since it provides multiple DOFs for each finger and the hand as a whole. By tracking orientation of fingers and the relative position of the hand, the glove device can track an enormous variety of gestures, each of which corresponds to a different type of data entry. This gives the glove remarkably rich expressive power, which can be used in the inputting of extremely complicated data.

The glove also allows direct manipulation of objects in the virtual world such as simulated surgery on a patient. All the gloves contain pressure and movement sensors connected to a display terminal which allows manipulation of images and objects.

Modern glove devices measure finger flexure and hand orientation to a greater or lesser extent. Each type of glove measures either four or five of the fingers (the little finger is sometimes excluded) for the degree of flexure. Gloves can track hand orientation by measuring roll, pitch and yaw or position of the hand as a whole.

Since the first glove device available in 1983, the technology has evolved to several different techniques to track the orientation of the hand and fingers. Mainly they are fibre optics, ultrasonic, magnetic, electrical resistance, or some combination of these methods. The glove device will pass data about hand and finger positions to a tracker, which is a piece of equipment that processes the data so that it can be understood by the computer. The computer then interprets data according to its algorithm and responds to the event.

Some of the technologies commonly used in the market to measure finger flexion and hand orientation or position will be briefly described below.

3.4.1. Driven Technology

Fibre Optics

Optical fibres are inserted inside the glove run along the fingers to measure finger flexion. Figure 14 shows a fibre optics glove. As the fibres bend, the transmitted light is attenuated and the signal strength for each of the fibres is sent to a processor that determines joint angles based on pre-calibrations for each user. Fibre optic requires a light source and a photo diode receptor for every single strand. To ensure accurate measurements, precise calibration is needed in order to measure the change in the attenuated light during flexion. Fibre optics is fragile and costly for mass production.



Figure 14: Fibre optics glove

Ultrasonic

Ultrasonic makes use of frequency above the audible range of the human ear which is 20 KHz to sense the hand displacement. The signal is emitted out and sensors such as the microphone are required to receive it. The combined distance from individual sensors and the emitters are used to triangulate the hand's position and/or orientation. These systems require line-of-sight and the interference from external devices may affect the results. The distance and other factors can hinder the accuracy of the measurements in an ultrasonic system. Figure 15 shows an ultrasonic glove.

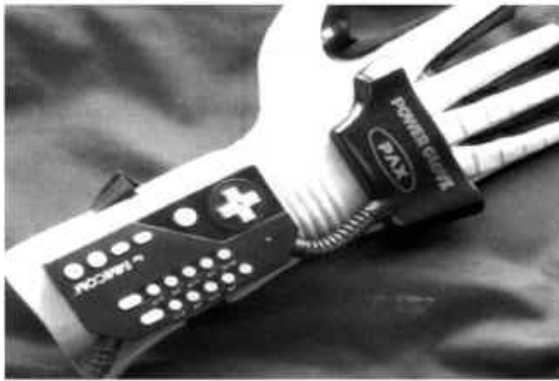


Figure 15: Ultrasonic glove

Magnetic

Small magnets are housed within each of the joints providing the ability to measure flexure of all three joints per finger. The strength of the magnetic signal within each joint varies according to flexure of the joint and is translated into a measurement of the bend of each finger joint which is shown in Figure 16. The magnetic signal may also be affected by external interference from other devices.



Figure 16: Magnetic glove

Electrical Resistance

Electrical resistance device uses fibres with conductive material running up through each finger; the change in electrical resistance is being measured as the fingers are bent. Electrical resistance technologies can be sensitive to environmental magnetic fields and the presence of ferromagnetic materials can provide erroneous readings. A commercially available electrical resistance glove is shown in Figure 17.



Figure 17: Electrical resistance glove

Most glove devices utilize the Polhemus or Ascension tracking devices for accurate hand position and orientation measurements. For hand position and orientation, gloves often interface with the Polhemus Tracker to track orientation of hand as a whole. Moreover, this requires a wire bundle to be attached to the glove device in some fashion which limits the movement of the operator. Current

technologies available for measuring hand position and orientation in an unencumbered fashion are not as accurate as the Polhemus and Ascension systems and therefore the overall functionality of the glove devices may be limited in scope. Calibration may be time-consuming but is necessary in order to obtain the before and after flexion measurements of the joints accurately as reflected by the motion of the user. Calibration is required for each individual user.

3.4.2. Glove Comparisons

Glove devices are the most common hardware devices used in the 3D environment. They are wired glove-like input devices that equipped with various sensor technologies and motion trackers to capture the physical data of the human's hands. Physical data such as bending of fingers, global glove's position or rotation are then interpreted by the software, so any one movement is mapped into the 3D environment. The following section presents the commonly use commercial glove devices and their cost.

VPL Research Inc.: - DataGlove



Figure 18: DataGlove

1. Consists of a Lycra glove with optical fibres running up each finger with photodiode at one end and light source at other.
2. Combines with Polhemus tracking device.
3. Monitors 10 finger joints (lower two of each finger, two for thumb) and six DOF of the hand's position and orientation (magnetic sensor on back of glove).
4. Some had abduction sensors to measure angle between adjacent fingers.
5. Flex accuracy to be closer to 5 or 10 degrees (not accurate enough for fine manipulations or complex gesture recognition; originally rated at 1 degree).
6. Speed of approximately 30Hz (Insufficient to capture rapid hand motions for time-critical applications).
7. Average cost US\$11700.

Exos Dextrous: - Hand Master

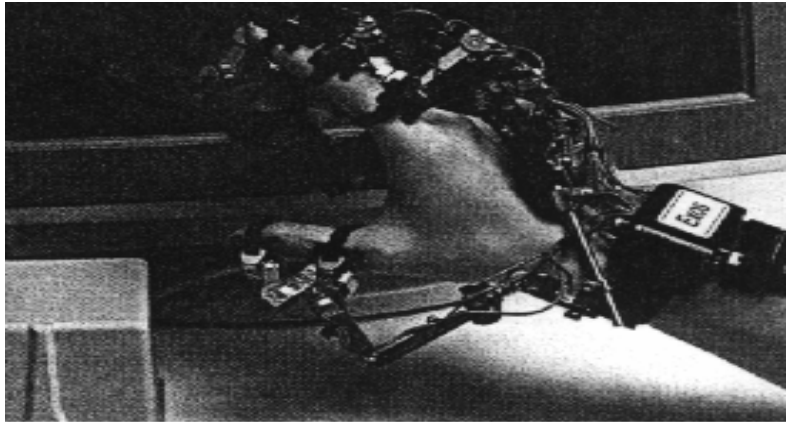


Figure 19: Hand Master

1. Lightweight aluminium exoskeleton for the hand.
2. Measures flexure of all three finger joints via magnet housed within each joint.
3. Much more accurate than DataGlove, provides 20 degrees of freedom with 8 bits of accuracy at up to 200 Hz.
4. Typically uses Polhemus tracker for 6 DOF hand tracking.
5. Average cost US\$17400

Mattel/Nintendo/AGE Inc.: - PowerGlove



Figure 20: PowerGlove

1. Cheap substitute for other glove devices.

2. Consists of a sturdy Lycra glove with flat plastic strain gauge fibres coated with conductive ink running up each finger; measures change in resistance during bending to measure the degree of flex for the finger as a whole.
3. Measures bend for only one segment per finger and guesses at degree of flexure of other segments; sensors on first four fingers; each bend reported as a two-bit integer.
4. Employs ultrasonic system (back of glove) to track roll of hand (reported in one of twelve possible roll positions), ultrasonic transmitters must be oriented toward the microphones to get accurate reading; pitching or yawing hand changes orientation of transmitters and signal would be lost by the microphones; poor tracking mechanism. (4D - x, y, z, roll)
5. Series of buttons along back of glove complete data entry possibilities; originally designed to be firing and moving buttons for video games.
6. Glove is only usable at up to roughly 45 degrees, and within five to six feet of the receivers, its (x, y, z) coordinate information is accurate to within 0.25 inches.
7. Average cost US\$120

Virtual Technologies: - CyberGlove



Figure 21: CyberGlove

1. Measures flexure of first two knuckles of each finger by means of strain gauges.
2. Two models: 18 DOF and 22 DOF (22 measures fingertip joint).
3. Measures abduction between fingers and a number of additional measures around thumb (since it has 5 DOF); measures wrist.
4. Combines with Polhemus or Ascension Tracker.

5. Available for both hands.
6. Requires calibration.
7. Sensor resolution of 0.5 degrees.
8. Written in C; supports OpenGL and Performer.
9. Average cost US\$9800

FakeSpace: - PINCH Gloves



Figure 22: PINCH Glove

1. Gesture recognition system to allow users to work within virtual environment.
2. Uses cloth gloves with electrical sensors in each fingertip; contact between any two or more digits completes conductive path, and a complex variety of actions based on these simple "pinch" gestures can be programmed into applications.
3. Compatible with Ascension and Polhemus trackers.
4. No calibration since not measuring anything.
5. Average cost US\$2000

Fifth Dimension Technologies: - 5DT DataGlove



Figure 23: 5DT DataGlove

1. Correspondingly less accurate, with only finger bend measuring (one sensor per finger) and no thumb flex or abduction measures; 8-bit fibre-optic resolution for each finger.
2. Measures roll and pitch of a user's hand with 2-axis built-in.
3. Tilt sensor for 60 degree range.
4. Right hand and left hand gloves.
5. Requires calibration.
6. Average cost US\$3495 (5 Sensor) and US\$6995 (14 Sensor)

Although the glove provides rich multidimensional input, most of the applications today do not require such a comprehensive form of data input, whilst those that do cannot afford it. However with the availability of cheaper versions encourage the development of more complex systems that are able to utilize the full power of the glove as an input device.

The glove has the advantage that it is very easy to use, and is potentially very powerful and expressive. It can provide ten joint angles, plus the 3D spatial information and degree of wrist rotation. The disadvantages are that it is extreme expensive and it is difficult to use in conjunction with keyboard.

The potential for the glove technology is vast. Areas like gesture recognition allow user to start a program by pointing finger at an icon on the screen and then close it by waving goodbye. Sign language interpretation is also an obvious area of focus in the research. Sign language can be translated through the device into sound from a vocally-impaired person. Another useful application for glove technology is for performing remote surgery on patient where the doctor is located miles away.

Developing application using these gloves can be difficult and expensive. Each type of glove has its own software development kit (SDK) to read the data from the device's driver, therefore most of the code written is not reusable. This will increase the cost of development and also the developing cycle. Data reading can be poll or interrupt depending on their driver. This makes the generalization of software design more difficult to implement.

3.5. 3D Visual Interfacing Device

3.5.1. Stereo Viewing

Stereo viewing is a common technique to increase visual realism or enhance user interaction with 3D scenes. The technique creates two viewing scenes, one for the left eye and the other for the right eye. Special viewing hardware is used together with the display scene, so each eye only sees the view created for it. The apparent depth of objects is a function of the difference in their positions from the left and right eye views. When done properly, the object appears to have an actual depth, especially with respect to each other. The stereo image creates the illusion of 3D by presenting a different view to each eye. The brain deciphers the small parallax differences between each view and reconstructs the depth information. When animating, the left and right back buffers are used, and must be updated each frame. A variety of methods can be used to turn a 2D display into a 3D display. The commonly methods are the active stereo, the Red-Green or Red-Blue stereo and the passive stereo.

An active stereo also called quad-buffered stereo or page-flipped stereo, alternatively displayed the stereo image pairs and the display is synced with shutter glasses through infrared transmitter attached to the computer to create the stereo effect. Since the stereo image pairs are being flipped, the refresh rate becomes half for each image. At the minimum required refresh rate of 100 Hz, each image is only going at 50 Hz. Generally, a refresh rate of 120 Hz is suitable for normal viewing. Active stereo can only be done on CRT type displays due to the refresh requirement. The advantages are high quality stereo viewing, only one monitor is needed and no special polarization screen is needed. However Active stereo only works with CRT Monitors and Projectors which are very expensive. CRT monitors do not make seamless display walls and stereo capable video cards are expensive.

The Red-Green or Red-Blue stereo completely relies on the program or images being displayed. In red-blue stereo, a composite image is created by replacing the red channel from the right eye image with the red channel from the left eye image. The advantages in this kind of stereo are using cheap tinted glasses to view the stereo, easy to implement and no special hardware or anything else is needed. However, the stereo is of poor quality.

A passive stereo uses vertically polarized light for one eye image and horizontally polarized light for the other eye image. This is usually done using 2 projectors. Then again monitors cannot do passive stereo so a special polarised filter screen and a special inexpensive polarized glasses are needed. The advantages are quality stereo, no refresh rate requirements, inexpensive polarized glasses and low video card requirements but projector cost is expensive and has alignment problem.

There are numerous commercial available techniques that have been invented to correctly deliver the left and right views from a flat print or display; many require additional equipment such as 3D glasses. Here three most common ways of creating stereo using a monitor display are presented.

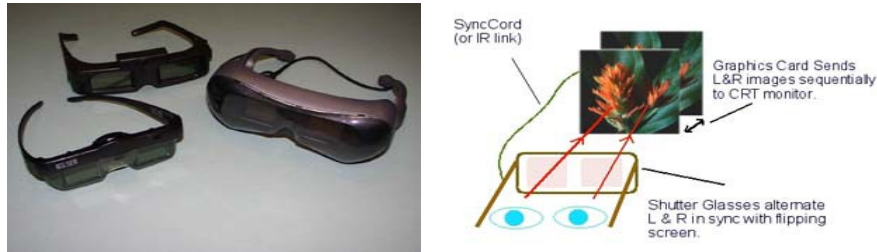


Figure 24: LCD shutter glasses

Frame Sequential

In the frame sequential method, stereo shutters are used to ensure that the two eyes receive alternate frames of the video image. One common technique incorporates the shutters in glasses which synchronize with the monitor. The LCD shutter glasses shown in Figure 24 alternately block out light coming to the left and right eyes which is done at high frequency in order to eliminate flicker. The shutters are synchronized with the sequential presentation of the left and right images to the monitor by the graphics board and the sync is done using a wire or a remote infra-red link. In another technique the polarization of a screen covering the monitor can be changed for alternate frames; in which case the user wears Polaroid glasses that are polarized differently for each eye. Circular polarization is used since this is not affected by head orientation. Either technique works well if the graphics system can provide at least 50 updates per second to each eye. Systems that generate fewer updates to each eye are not recommended because this may cause an irritating flicker. Another problem with this method of stereo presentation is the ghosting that occurs primarily because of too slow decay of the green phosphor on the monitor. This means that each eye sees a faint image from the other eye's signal.



Figure 25: Anaglyph glasses

Red-Green Anaglyphs

In red-green anaglyphs, the two images are generated by using only the red (for one eye) and green (for the other eye) monitor primaries. The user wears red and green filters over the opposite eyes, which effectively block the undesired images. Figure 25 shows an anaglyph glasses and the image. This is a low-cost solution for experimenting with stereo displays. However, its disadvantages are that only a monochrome image can be displayed and the combination of red and green to the two eyes often produces strange colour effects in the combined image. In practice, there is some colour leakage so that the red (Left) eye may see a little of the blue signal, and vice versa. This leads to ghosting, which can become a major problem for images with extremely contrast.

Mirror Systems

In the mirror systems, the screen is divided into two parts, one part for the right eyes image and the other part for the left eyes image. These images are displaced by a system of mirrors (or sometimes prisms) so that the two parts appear superimposed. This is an excellent low cost solution that creates a high quality stereo image, with no ghosting but with a sacrifice of the effective display size, since half of the screen must be devoted to each eye. It also provided the most constrained viewing configuration.

3.6. Interfacing Software

3.6.1. DirectX

The Microsoft DirectX is an application program interface (API) for creating and managing graphic images and multimedia effects in applications such as games or active Web pages that will run in Microsoft's Windows operating systems. It is an integral part of Microsoft Windows 98, Microsoft Windows Millennium Edition (Me), and Microsoft Windows 2000, as well as Microsoft Internet Explorer. It is designed to free the microprocessor for other works by allowing the graphics

accelerator card to perform some functions. The accelerator manufacturer will provide the driver especially for DirectX through the Driver Development Kit (DDK) that lets hardware developers create drivers for display, audio, and other I/O devices.

The DirectX Software Development Kit (SDK) includes tools that allow a software developer create or integrate graphic images, overlays, sprites, and other game elements, including sound. DirectX provides software developers with tools that help them get the best possible performance from the machines they use. It provides explicit mechanisms for applications to determine the current capabilities of the system's hardware so they can enable optimal performance.

The APIs in DirectX are low-level functions, including graphics memory management and rendering; support for input devices such as joysticks, keyboards, and mouse; and control of sound mixing and sound output. These functions are grouped into components that make up DirectX and consist of five components:

1. DirectDraw, an interface that lets you define two-dimensional images, specify textures, and manage double buffers (a technique for changing images)
2. Direct3D, an interface for creating three-dimensional images
3. DirectSound, an interface for integrating and coordinating sound with the images
4. DirectPlay, a plug-in for end users, is also used by developers to test their application
5. DirectInput, an interface for input from I/O devices

DirectInput supports wide range of input device in the market. It is capable of providing action mapping in the system. This allows user to allocate the exact actions to the buttons and axis of input devices. Therefore data extraction is possible without the system knowing what type of device generate it. Moreover it is also capable of handling force-feedback devices.

4. Game Device

When it comes to playing the favourite games, the type of gaming genre to play will be important. Whether it is role play game (RPG) (Baldur's Gate, Fallout, Asheron's Call, Final Fantasy VIII), first-person shooter (Serious Sam, Counter-strike, Tribes 2, Giants: Citizen Kabuto), third-person action adventure (Tomb Raider, Frogger 2D, Severance: Blade of Darkness), strategy (Black & White, Civilisation III, Fallout Tactics) or sports (Need for Speed, Motor City Online, FIFA series), the gaming peripherals or interfacing devices to use are important. The keyboard and mouse combo have been touted as the king of peripheral gaming for almost all forms of personal computer (PC) games since the days of ID Software's Quake. Only certain stats-intensive games will require the use of a decent joystick, steering wheel or game-pad.

4.1. Joystick Device

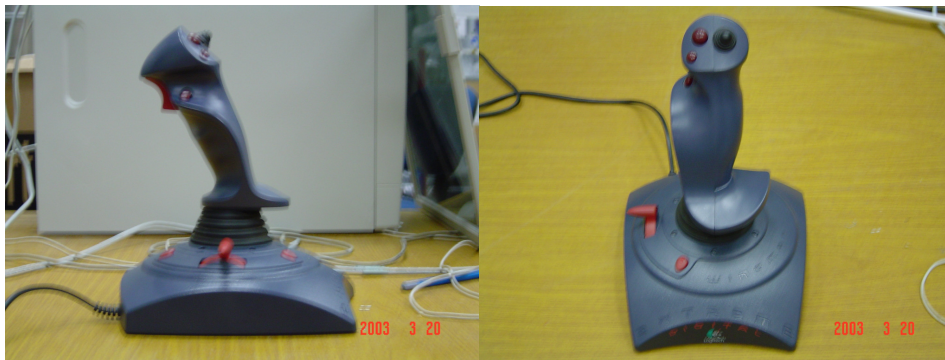


Figure 26: Joystick

It is an indirect input device, occupied very small space. Consisting of a small palm-sized box with a stick or shaped grip sticking up from it, the joystick is a simple device with which movements of the stick cause a corresponding movement of the screen cursor. There are two types of joystick, the absolute and the isometric. In the absolute joystick, the position of the joystick directly indicates the direction of the screen cursor relative to the world. In the isometric joystick,

strain gauges measure the force applied to the joystick in any direction. The cursor moves in proportion to the amount of force applied. This type of joystick is also called the velocity-controlled joystick. The buttons are usually placed on the top of the stick, or on the front like a trigger, or the base. Joysticks are inexpensive and fairly robust, and for this reason they are often found in computer games. The disadvantage of joystick is the limitation in the usage for drawing tasks. Joystick cannot be used to trace or digitize drawings. The average cost of a joystick ranges from US\$20 to US\$80.

4.2. Game-pad Device



Figure 27: Game-pad

Though still the de facto peripheral in console gaming, game-pads for PC games have so far lost their lustre. Most PC's game-pads are used only in soccer game or platform game. A typical game-pad connection is basically USB-based, so setting up the game-pad is easy. It hardly needs to reconfigure all the buttons (except the preliminary calibration). Some the game-pad also supports force feedback which equipped with a motor housed within the left grip, would vibrate and shake according to the severity of the user's manoeuvre. Even though the rattle is not as strong as expected, at least the vibration is obvious and sometimes constant. Game-pad normally cost from US\$10 to US\$40.

4.3. Steering Wheel Device



Figure 28: Steering Wheel

Racing wheels began to earn some limelight after the record-breaking success of Electronic Arts' Need for Speed. The game peripheral companies produced wheels to cater to a home-arcade crowd for gamers to lap up this true-to-life PC game. The Steering Wheel normally comes with USB connection which delivers true professional quality with features such as an ergonomic designed steering wheel and paddle shifters. It offers precision handling for the performance that racing fanatics demand. There are programmable buttons mounted on the centre plate of the wheel, and a floorboard which mounted the gas and break pedals, complete the pro racing experience. Professional styling for the wheel and pedals means quality, comfort, and control. Some steering wheels also come with force feedback, allow user to have a grip on the most realistic, high-end PC racing experience available. Likewise for the joysticks, they also have the disadvantage in drawing tasks. The average cost of steering wheel normally price from US\$40 to US\$100.

Game devices are easy to interface and less expensive than the tracking and glove devices. Although they are not part of the basic input devices of the OS, their device's drivers follow a common set of interfaces. Their drivers also use common polling mechanism for data reading. Owing to the use of DirectX's DirectInput, programming application to use these devices is relatively simple.

4.4. P5 Glove Device



Figure 29: P5 glove

The Essential Reality P5 Data Glove is a 3D input device capturing finger-bend and relative hand-position that enables intuitive interaction with 3D environments. The P5 glove fits over the hand and senses all its movements in three dimensions, becoming the interface to a PC or game console. It is a featured rich and cost-effective data glove which is suitable for 3D interactive gaming and virtual environments. It is also a lightweight device and can be used as a mouse in both Windows and Mac OS 9 systems which cost about US\$89.

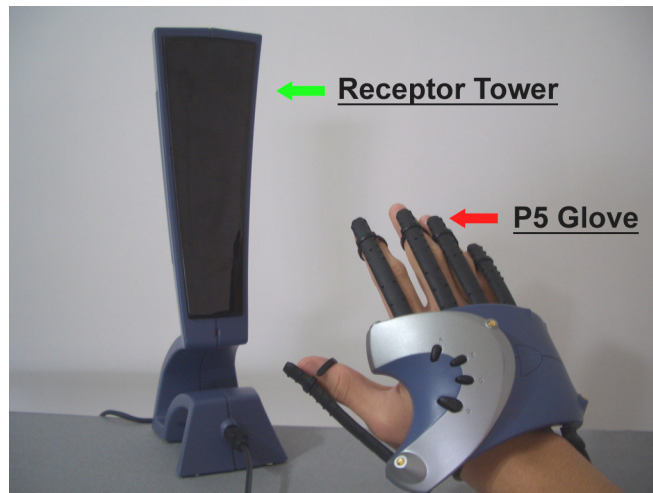


Figure 30: P5 Glove Unit

The P5 glove is made up of the the receptor tower and the glove. The receptor tower comprises of an array of infra red emitter while the glove consists of four buttons on the top, eight light emitter diode (LED) sensors on the data glove and bends sensors in its five fingers. The P5 glove is attached to the receptor tower and the receptor tower is in turn connected to the PC via USB port. The sensed data which are the position (X, Y and Z), the orientation (Pitch, Yaw and Row) and the bend information via the connected cable is transmitted to the connected PC.

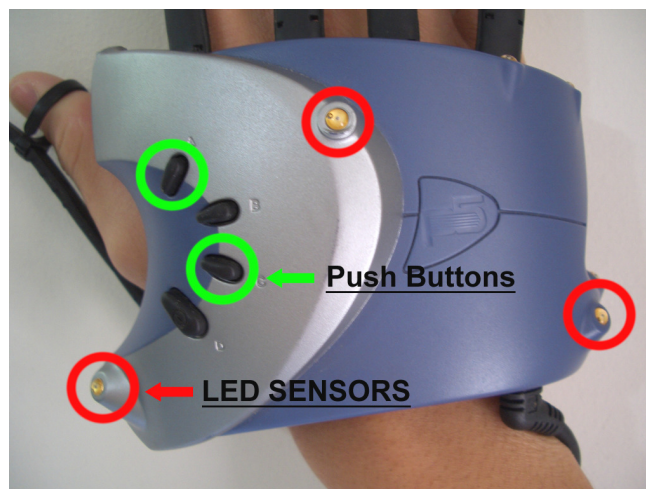


Figure 31: P5 Glove Unit – Close Up

The P5 glove uses Essential Reality's proprietary bend sensor and tracking technologies to enable full interaction with a 3D environment. The P5 glove is designed to enhance the PC game-playing experience and provides user with these extraordinary features:

1. Lightweight, ergonomic design for easy, intuitive play. Weighs just 4.5 oz.
2. The first widely available virtual 3D controller
3. Mouse-mode compatible with any application
4. 6 degrees of tracking (X, Y, Z, Yaw, Pitch and Roll) to ensure realistic movement - most trackball, joystick and mouse controllers offer only 2 degrees of freedom
5. Bend-sensor and optical-tracking technology to provide true-to-life mobility
6. Easy, plug-and-play setup - plugs right into the USB port of your PC
7. Infrared control receptor with scratch-resistant, anti-reflective lens
8. Easy-to-use anywhere – desktop and living room.

The P5 glove is an innovative, glove-like peripheral device that provides users total intuitive interaction with 3D virtual environments, such as games, websites and educational software.

The P5 glove is designed like a game device. Its driver used polling to access its data and it can be programmed to use as a mouse device. It is not difficult to integrate P5 glove into an application using its SDK. Moreover, it has almost similar data structure as joystick device. This makes the generalization of 3D interfacing module much easier.

5. 3D Interaction and System Design

5.1. 3D Interaction

3D interaction is a special module developed to integrate different game devices into virtual reality environment. Using those devices help to reduce the overall cost of the virtual reality system.

The 3D interaction module is made up of four sub components, the application layer, the 3D interaction common data, the common game device and the glove device components. Figure 32 shows the design architecture of the 3D interaction module. It gives an overview of the 3D interaction module's structural framework.

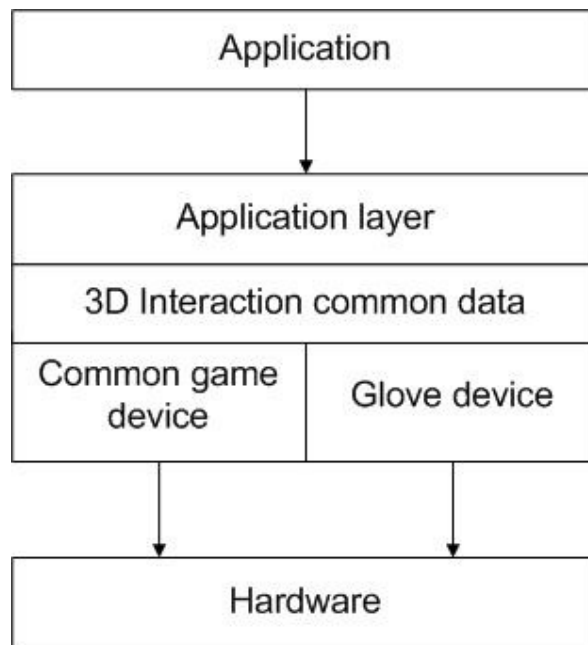


Figure 32: 3D Interaction design architecture

The framework organizes the components into different layers to deal with the complexity of the system. There are 3 layers in the system, the top layer interfaces with the application, the middle layer holds the data, and the bottom layer manages and accesses to the hardware of the devices. The top layer comprises of the

application layer component, the middle layer is the 3D interaction common data component, and the last layer includes the common game and glove device components.

The application layer provides a common interface for all the applications to access the 3D interaction module's data. The data contain the DOF information which describes an object moves in 3D space. Most of the application will need this information such as the X, Y and Z translations and X, Y and Z rotations to manipulate its 3D objects. Additional to the DOF information, the application layer also supplies state information to the application. State information is equivalent to the device's button; it reflects a true or false state that corresponds to the device's button on or off. The application layer is designed based on subscription methodology. Each application is required to register with the application layer for the information it is interested. Table 1 shows the interface of this component.

Application layer interface	Description
CreateDevice	Create a device for application use
DeleteDevice	delete device from the device list
GetDevices	Get the devices from list
GetNumDevices	Get the number of device
ConfigureDevice	Configure devices to the application
Register	Register callback to the device
UnRegister	Unregistered to the device
GetHardwareList	Get a list of hardware device
AssignDevices	Assign devices to hardware device
UnassignDevices	Unassign devices to hardware device

Table 1: Application layer interface

The 3D interaction common data stores the normalized data that acquired from each hardware device. It provides a share space for different devices and the application to link together. The application will view this share space as a single device through the application layer, while different devices will update the share space as though it is the only device. Table 2 and Table 3 show the interface and the data.

3D interaction common data Interface	Description
CreateDeviceData	Create a common data

InitData	Initialize common data
AddHardwareToData	Associate hardware to common data
RemoveHardwareToData	Disassociate hardware to common data
GetDeviceData	Get common data
SetDeviceData	Set common data

Table 2: 3D interaction common data interface

Common data	Description
m_fXpos	Normalized X position
m_fYpos	Normalized Y position
m_fZpos	Normalized Z position
m_fXrot	Normalized X rotation
m_fYrot	Normalized Y rotation
m_fZrot	Normalized Z rotation
m_pButton	Pointer to states
m_pFeature	Pointer to feature
m_nNumOfButton	Number of button
m_nNumOfFeature	Number of feature
m_nDeviceType	Device type
m_bValidflag	Data valid flag

Table 3: 3D interaction common data

Both the common game device and glove device components are designed to be a hardware device manager. They are inherited from an abstract interface called the hardware device manager interface which allows the 3D interaction common data component to linkup with different hardware devices. The abstract interface is showed in Table 4 .

Hardware device manager interface	Description
BuildHardwareDevice	Build hardware device

Table 4: Hardware device manager interface

The common game device component handles all the commercial game devices interfacing. It extracts and normalizes the data from the joystick, game-pad, steering wheel and mouse devices and updates the results to the 3D interaction common data. Table 5 shows the data that are acquired from each commercial game device. Table 6 shows its interfaces require to communication to the hardware devices and Table 7 shows the extracted data stored.

Device\DOF	X	Y	Z	Roll	Pitch	Yaw	Button
Joystick	•	•		•	•		•
Game-pad	•	•	•				•
Steering wheel	•	•					•
Mouse	•	•					•

Table 5: Data captured by commercial game devices

Common game device interface	Description
InitDevice	Initialize the device pointer
InitJoystick	Initialize the common game device
HowManyButtons	Find out how many buttons the attached device has
CreateDevice	Create a device pointer for a GUID
GetFirstJoystickID	Get First common game device data for enumerated devices
GetNextJoystickID	Get next common game device data for enumerated devices
GetFirstButtonName	Get first common game device button friendly name for enumerated device
GetNextButtonName	Get first common game device button friendly name for enumerated device
GetJoystickStateInfo	Get common game device state information

Table 6: Common game device interface

Common game device data	Description
m_DevicePOS	Contains a pointer list to button names for selected common game device
m_ButtonPOS	Used in pointer list to keep track of next item
m_DIJoystickList	Contains a pointer list to attached common game devices
m_DIButtonNames	Contains a pointer list to button names for selected common game device
m_JoystickGUID	Current common game device GUID
m_dijs	Holds common game device state information

Table 7: Common game device

The glove device component takes care of the P5 glove interfacing. Information from the P5 glove device is obtained, normalized and reflected up to the 3D interaction common data, which then notified the respective applications. The glove device has the full set of data that requires in most 3D object, however manipulating of the P5 glove in 3D environment imposes some challenge to the user. Table 8 presents the data of a P5 glove while Table 9 and Table 10 show its interfaces and data between the hardware devices.

Device\DOF	X	Y	Z	Roll	Pitch	Yaw	Button
P5 glove	•	•	•	•	•	•	•

Table 8: Data captured by P5 glove

P5 glove interface	Description
P5_Close	Close P5
P5_Init	Initialize P5
P5_SetClickSensitivity	Set click sensitivity
P5_GetClickSensitivity	Get click sensitivity
P5_SaveBendSensors	Save bend sensors
P5_CalibrateBendSensors	Calibrate bend sensors
P5_CalibratePositionData	Calibrate position data
P5_GetMouseState	Get mouse state
P5_SetMouseState	Set mouse state
P5_SetMouseStickTime	Set mouse stick time
P5_GetMouseStickTime	Get mouse stick time
P5_GetMouseButtonAllocation	Get mouse button allocation
P5_SetMouseButtonAllocation	Set mouse button allocation

Table 9: P5 glove interface

P5 glove data	Description
m_nDeviceID	Device ID
m_nGloveType	Glove type
m_fx	X translation
m_fy	Y translation
m_fz	Z translation
m_fyaw	Yaw
m_fpitch	Pitch
m_froll	Roll
m_byBendSensor_Data	An array of bend sensor data
m_byButtons	An array of button data

m_fRotMat	A 4X4 matrix for inverse kinematics
-----------	-------------------------------------

Table 10: P5 glove data

The 3D interaction module can be easily integrated into different applications. The applications need to load the dynamic linked library, create an instance of the interfacing object and perform the respective initialization. The hardware device information will be constantly captured into the common accessible memory which allows the applications to acquire the required data. Callback features are also built to facilitate good response time for the application. Figure 33 shows the 3D interaction module integration diagram. This diagram describes the inner flowchart of the 3D interaction module with the application. The 3D interaction module starts off in an instantiation stage when the application created the 3D interaction module object. Next, it proceeds with all the hardware devices acquisition and common data initialization. Finally it enters the data acquisition phase where hardware device's information is continuously captured, processed and stored into the common access memory.

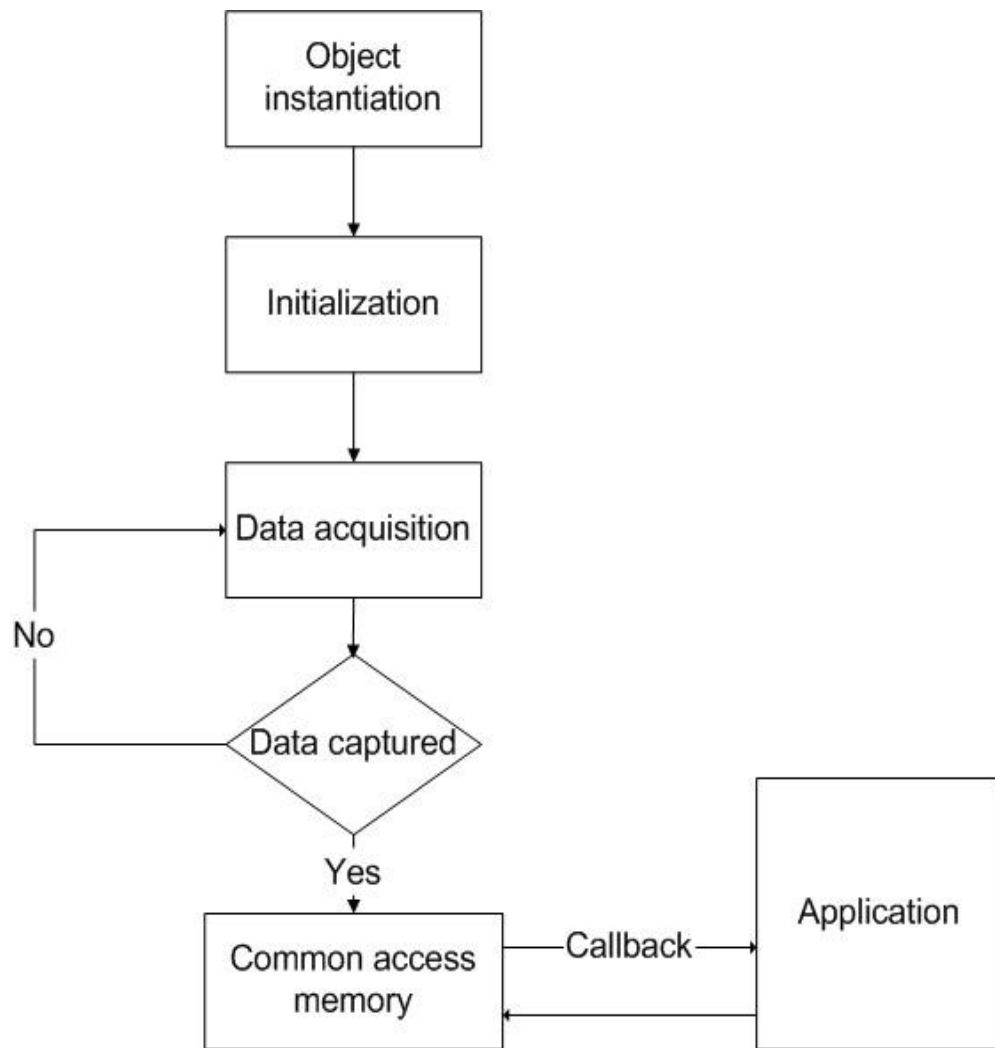


Figure 33: 3D Interaction integration diagram

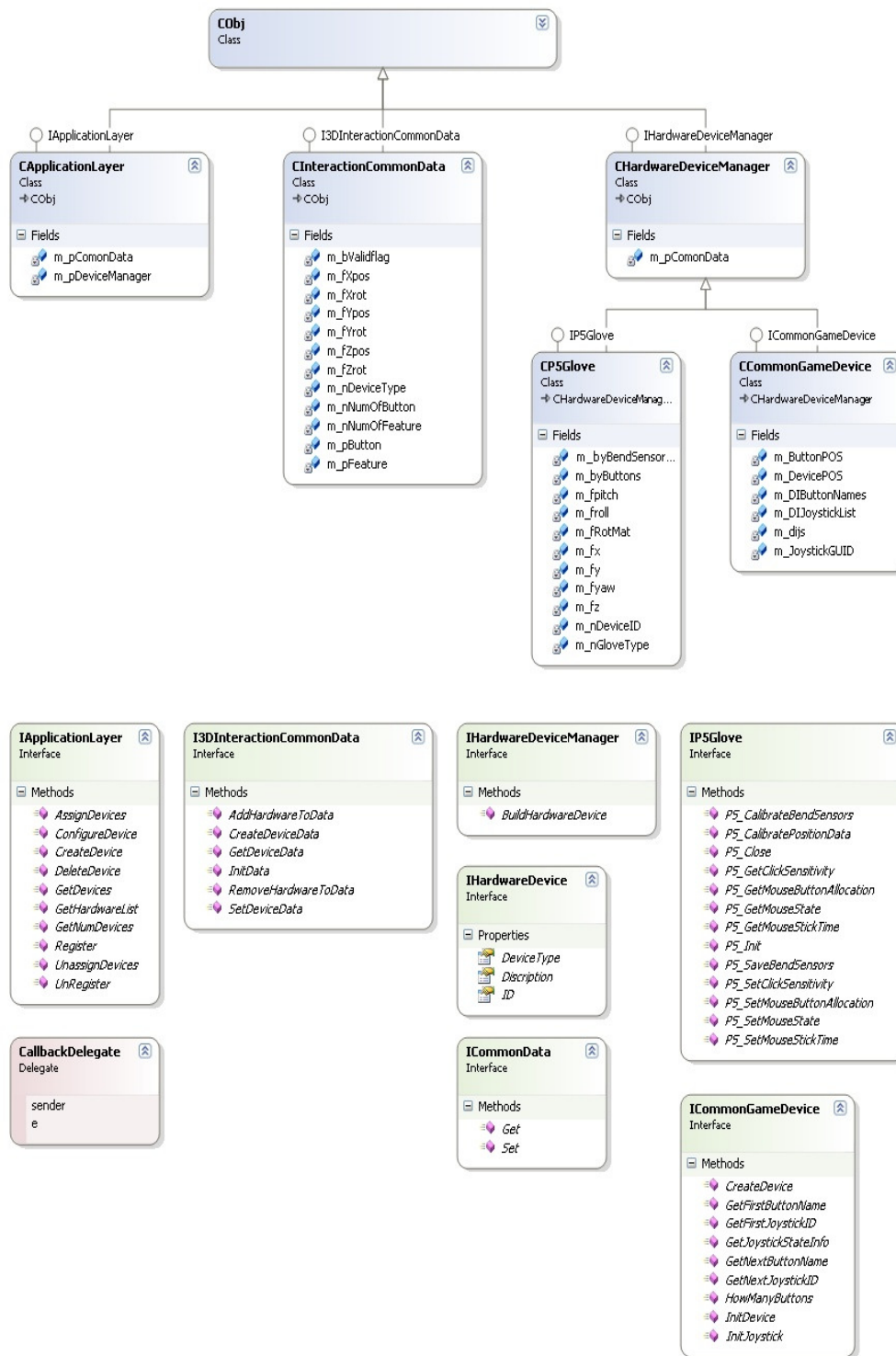


Figure 34: 3DInteraction class diagram

Figure 34 shows the class diagram of the 3D interaction. It describes the implementation of the 3D interaction module from the design architecture.

CObj class: This is the base class for all the classes that are implemented in the 3D interaction module.

CApplicationLayer class: This class realizes the IApplicationLayer interface. It holds the pointer to the hardware device manager and the 3D interaction common data objects. It is responsible for creating the hardware device manager and the 3D interaction common data objects for the application.

CInteractionCommonData class: This class realizes the I3DInteractionCommonData interface. It contains all the normalized 3D information.

CHardwareDeviceManager class: This class realizes the IHardwareDeviceManager interface. It is an abstract class for the CP5Glove and CCommonGameDevice classes. It serves as an abstraction layer for the application layer component. Therefore the application layer component needs not to know about the CP5Glove and CCommonGameDevice objects. It also contains a pointer to the 3D interaction common data object.

CCommonGameDevice class: This class realizes the ICommonGameDevice interface. It holds the game device's information. It is responsible for linking up with the game devices available in the system and obtaining their information.

CP5Glove class: This class realizes the IP5Glove interface. It stores the P5 glove's information. It is responsible for connecting to the P5 glove and extracting the glove's information.

IApplicationLayer interface: This defines the functions of the application layer. Table 1 shows its functions and their descriptions.

I3DInteractionCommonData interface: This defines the functions of the 3D interaction common data. Table 2 shows its functions and their descriptions.

IHardwareDeviceManager interface: This defines the functions of the hardware device manager. Table 4 shows its function and the description.

ICommonGameDevice interface: This defines the functions of the common game device component. Table 6 shows its functions and their descriptions.

IP5Glove interface: This defines the functions of the P5 glove device component. Table 9 shows its functions and their descriptions.

IHardwareDevice interface: This defines the hardware device's information stored by the hardware device manager. It is used to identify individual hardware device found in the system.

ICommonData interface: This defines the functions for the common data used by the application layer and the hardware device manager to store and retrieve data.

CallbackDelegate delegate: This defines the call back function the application needs to provide for the 3D interaction module. This allows the 3D interaction module to signal the application that the data is ready.

6. Tactile Interface Development

The application illustrates the use of inexpensive game devices and 3D interaction module to perform the graphic manipulation and visual interaction.

6.1. Graphic Manipulation and Visual Interaction

The use of graphic in computer system can demonstrate and communicate information and concepts in an easy and clear manner to the user. With advancements in computer technology, attractive and insightful images can be created. However, the computer screen is still two-dimensional and special techniques must be used to create the illusion of the third dimension. Therefore, several methods are employed to enhance the perception of three-dimensionality of object. They are the depth-cueing technique, the stereovision or stereo viewing, the colours combination and the ray-tracing technique. All these give very powerful perception of shape and sculpture of the object surface. Even with three dimensions, it is not enough to display continuously changing properties, like electrostatic potential or electron density, so others approach like animation or using the contour geometry have to be implemented.

When too many three-dimensional objects are shown simultaneously, the computer display has difficulty following. Therefore, the z-clipping technique is used to help to navigate among the different objects and see their relation. The technique filters out object outside the clipping planes and only displays the portion in between them.

There are many ways to display objects. We use proteins as an example to illustrate the different visual modes.

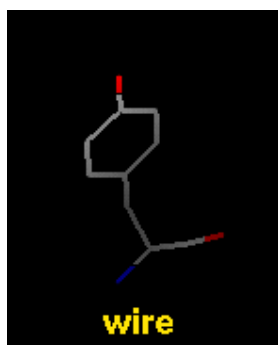


Figure 35: Wireframe mode

Wireframe mode: Atoms are shown as dots and bonds as wires.

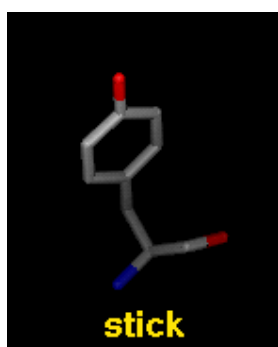


Figure 36: Stick mode

Stick mode: Nuclei of bonded atoms are connected by lines.

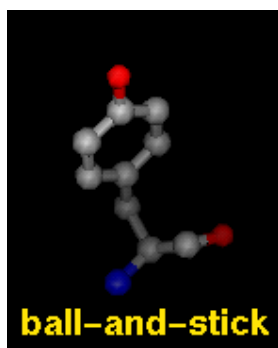


Figure 37: Ball-and-stick mode

Ball-and-stick mode: Atoms as shown as ball (small sphere) and stick bonds.

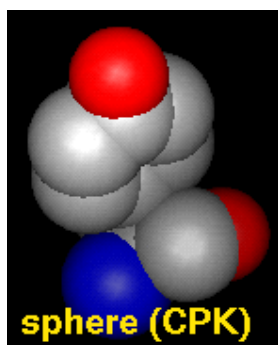


Figure 38: Sphere mode

Corey-Pauling-Koltun (CPK) or space filling mode: Atoms as shown as solid surface and stick bonds.

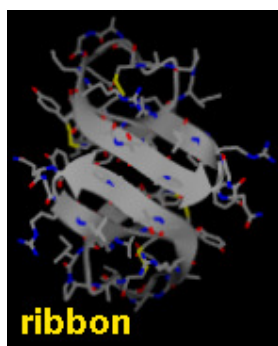


Figure 39: Ribbon mode

Ribbon mode: Proteins and nucleic acids can be drawn as a ribbon.

Graphic manipulation often refers to movement or animation. Animation is used to differentiate objects against each another by displaying them in turns, by rotating them continuously with constant speed, or by moving them. Graphic manipulation is the subset of visual interaction.

Protein visual interaction is a three-dimension (3D) interaction using the touch sensor of the body that is the hands to communicate to the virtual world. The components require for the 3D interaction are virtual reality system and a virtual environment. Others important issue are designing the user actions and the overall experience. In the design of the user actions, the movement, navigation, selection, manipulation and communication are important. While designing the overall experience, it requires presence, health and safely.

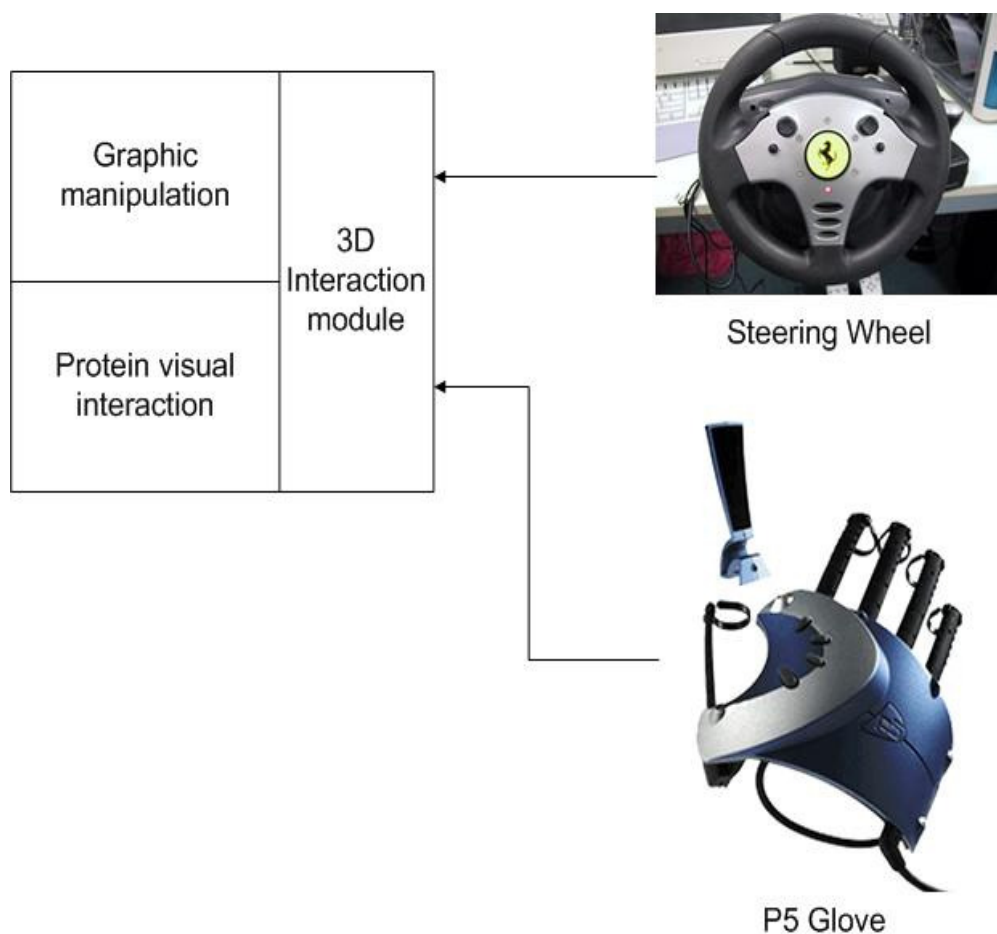


Figure 40: Tactile interface development design architecture

Figure 40 shows the tactile interface development for protein interaction design architecture. The 3D interaction module acts as the interfacing layer to the hardware device, information from those devices are captured and processed in order to realize both the graphic manipulation and protein visual interaction in the application.

6.2. Implementation

To illustrate the graphic manipulation and visual interaction in a system, a steering wheel, Formula Pro GTR, and a P5 glove are integrated into a visualization application. The visualization application will displayed a 3D protein molecular

structure from the protein data bank format. The molecular structure can be represented by different visual modes as shown from Figure 35 to Figure 39 in the application. The visualization application comprises of 3 main components, the graphic manipulation, the protein visual interaction, and 3D interaction module. The graphic manipulation component is responsible for movement and animation of the protein structure, the protein visual interaction component implements all visual modes drawing and user interacting effects and the 3D interaction module interfaces with the hardware devices.

The visualization application utilizes the 3D interaction module to capture the input data from the steering wheel and the P5 glove. Raw input information is processed and passed as 3D data to the graphic manipulation and the protein visual interaction components to interpret. The protein visual interaction component makes use of the 3D data to calculate the position and orientation of the protein structure and apply the transformation matrix into the virtual environment while the graphic manipulation component translates the 3D data to the current animated frame to simulate movement or animation of the protein structure.

To achieve graphic manipulation in the application, the steering wheel device is used to travel inside the protein molecule. Speed control is also possible through the use of the acceleration and brake pedals of the steering wheel set. Figure 41 shows the user navigating protein object using the steering wheel. User can steer his way into different part of the protein structure.



Figure 41: Steering wheel interface

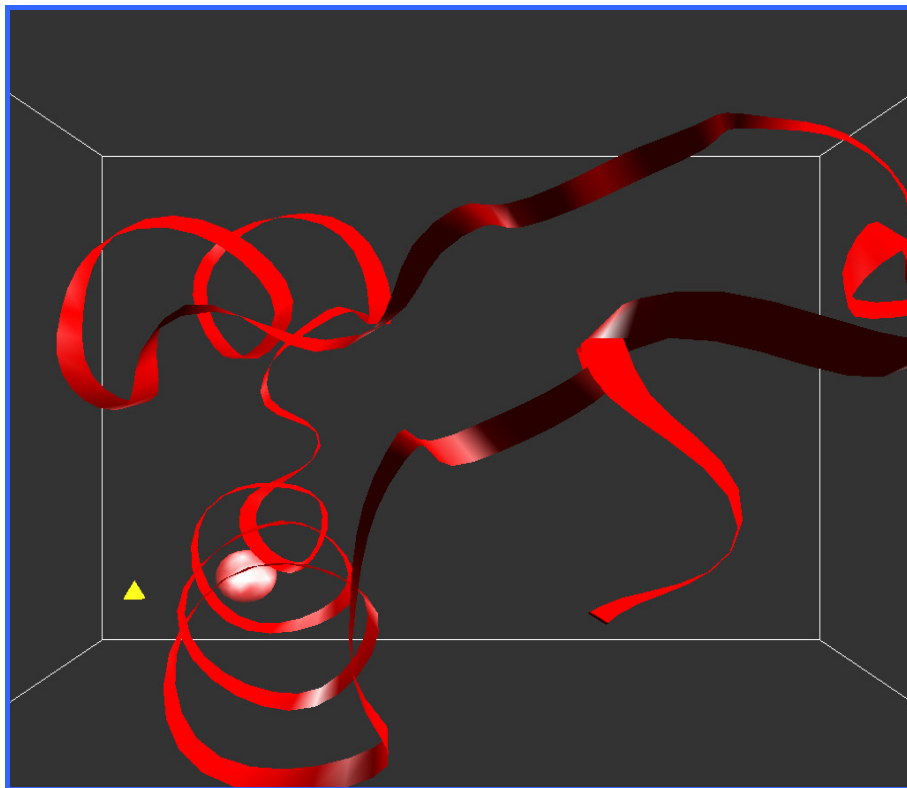


Figure 42: A ride through the ribbon structure of protein molecules

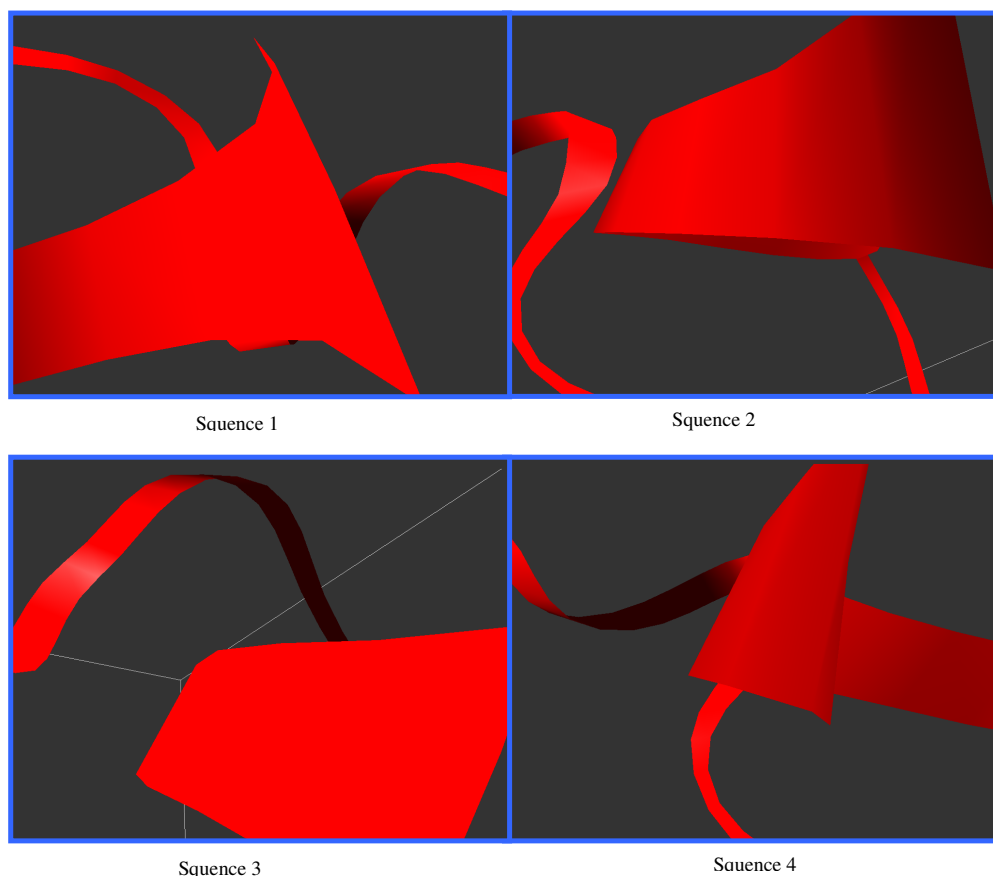


Figure 43: The sequence through the ribbon structure of protein molecules



Figure 44: Steering wheel interface for molecule navigation

To enhance the graphic manipulation, an animated journey in the ribbon structure of the protein that is interactive with the user is also created. The user uses the steering wheel to control the ride in the protein's ribbon structure. The steering wheel is used as the interactive device because it provides the end user a more

realistic feeling throughout the protein's journey since the ribbon structure of the protein resemble a rollercoaster ride. The main interaction is through the pedals of the steering wheel. The acceleration pedal is used to provide forward motion along the ribbon structure of the protein, while the brake pedal is used for slowing down and eventually created a backward motion of the ride. Speed control is applied through the degree of pressure pressed in the pedals. Figure 42 shows the ribbon structure of the protein, Figure 43 shows the sequence of the ride and Figure 44 depicts the steering wheel interface for the molecule navigation.

The author continues using the protein visualization application as the virtual environment and integrates it with the P5 data glove to demonstrate simple protein visual interaction. The protein visual interaction basically allows user to select and manipulate individual molecule of the protein structure. The P5 glove was selected because it fulfilled the basic requirement of a 3D input device and more importantly it costs much less than other glove devices available in the market. Figure 45 Figure 53 shows the usage of P5 glove in a tactile protein interaction.



Figure 45: Tactile protein interaction

The tactile protein interaction allows user to interact with the protein structure using his hands. The P5 glove senses the user's hand position and orientation and converts them to the protein's position and orientation during user selection. A user selection can allow user to choice either the whole protein structure or the protein's molecules.

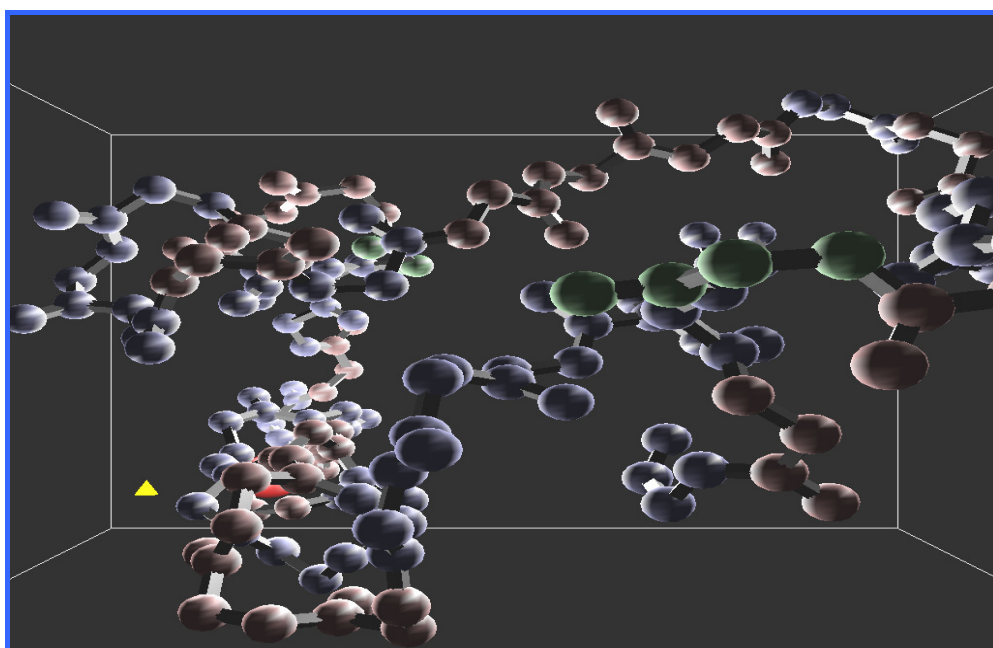


Figure 46: Virtual hand represented by the cone

When designing the user selection actions, a cone is drawn to represent the virtual hand in the virtual environment. It is used as a reference point for the end user to navigate and select protein molecules. The cone is allowed to move in the 3D space and rotate along its three axes that mapped directly to the movement of the P5 glove wore by the end user. When the cone touches the protein molecule, the molecule will change colour. Upon moving away from the molecule, it will return to its original colour. Selection or picking of the molecule is also possible when the user signals a hand gesture. The release of the selected molecule is performed through the same hand gesture. The selected molecule can also move in the 3D space as well as rotate in the cone's axes to allow user to examine it. Figure 46 captured a screenshot of the visualization application displaying the virtual hand which is represented by the yellow cone. Figure 47 to Figure 52 demonstrate the

steps to select an atom of the protein molecules and perform movement and rotation to it.

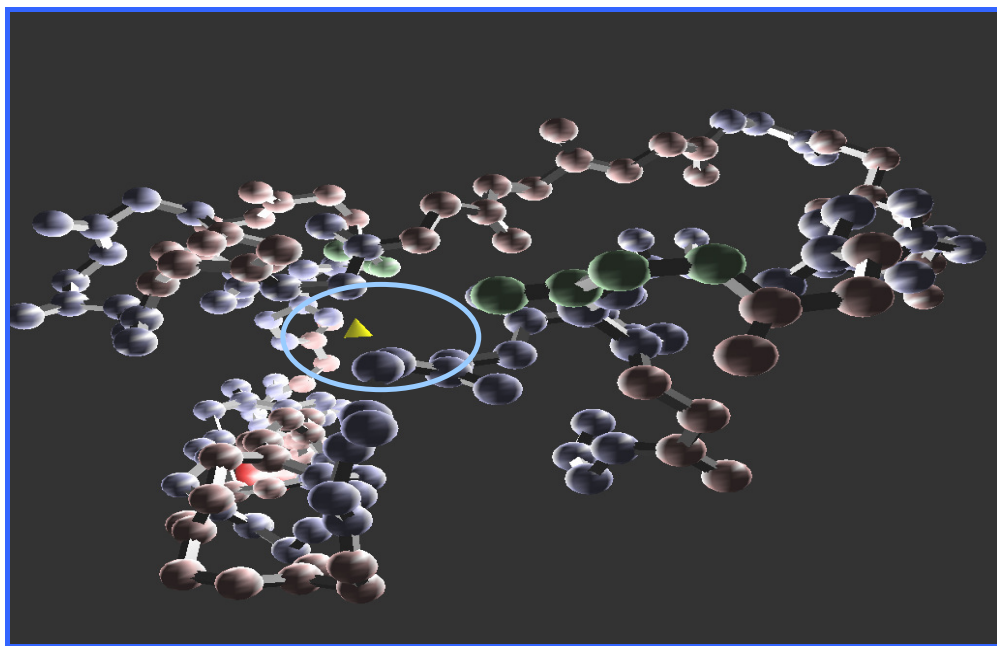


Figure 47: Cone approaches an atom of protein molecules

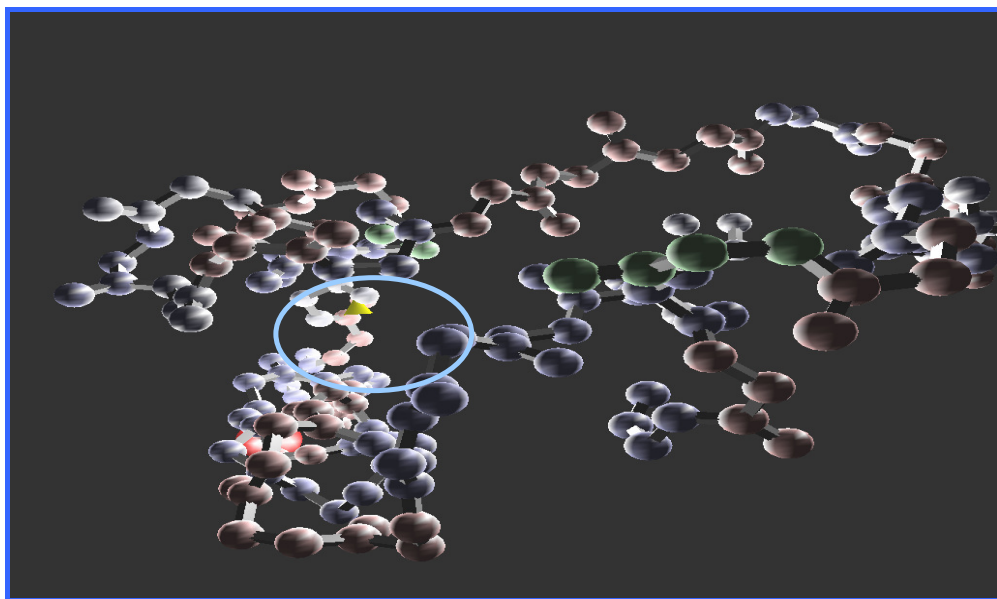


Figure 48: Select an atom

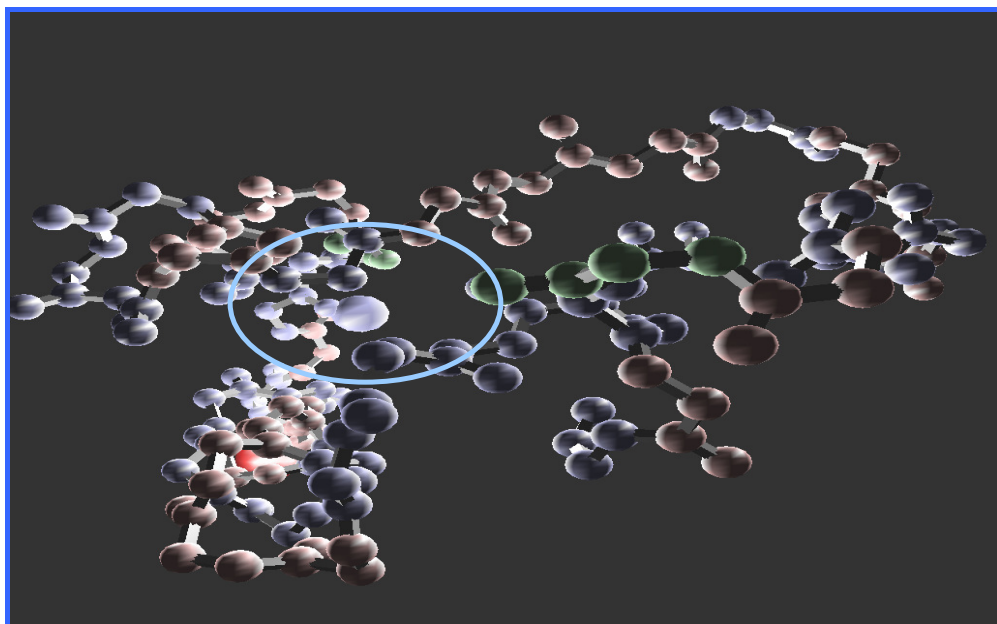


Figure 49: Pick the atom



Figure 50: Pick command

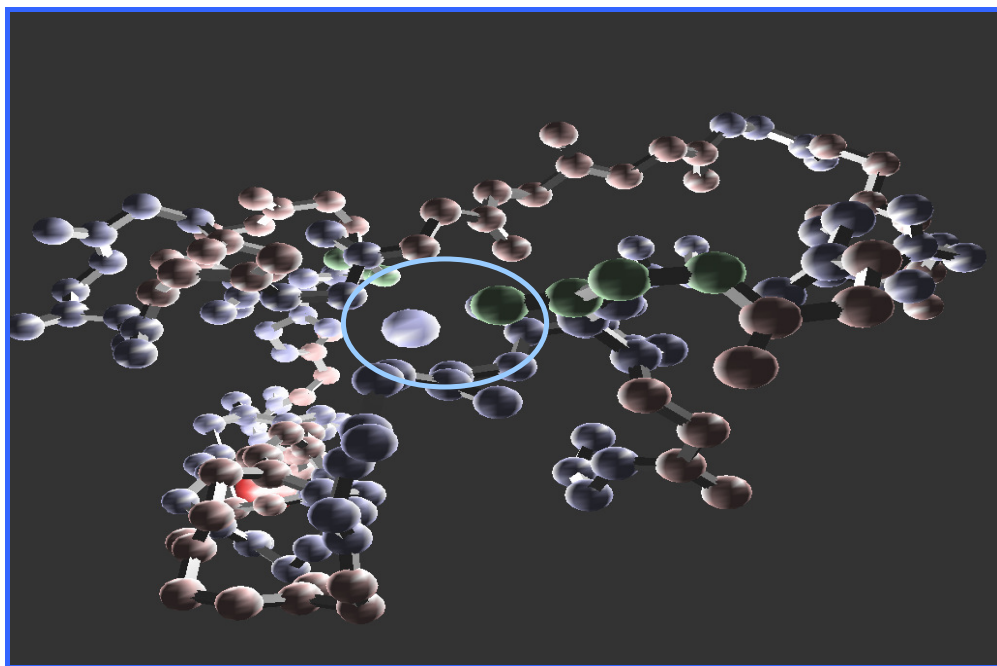


Figure 51: Move the atom

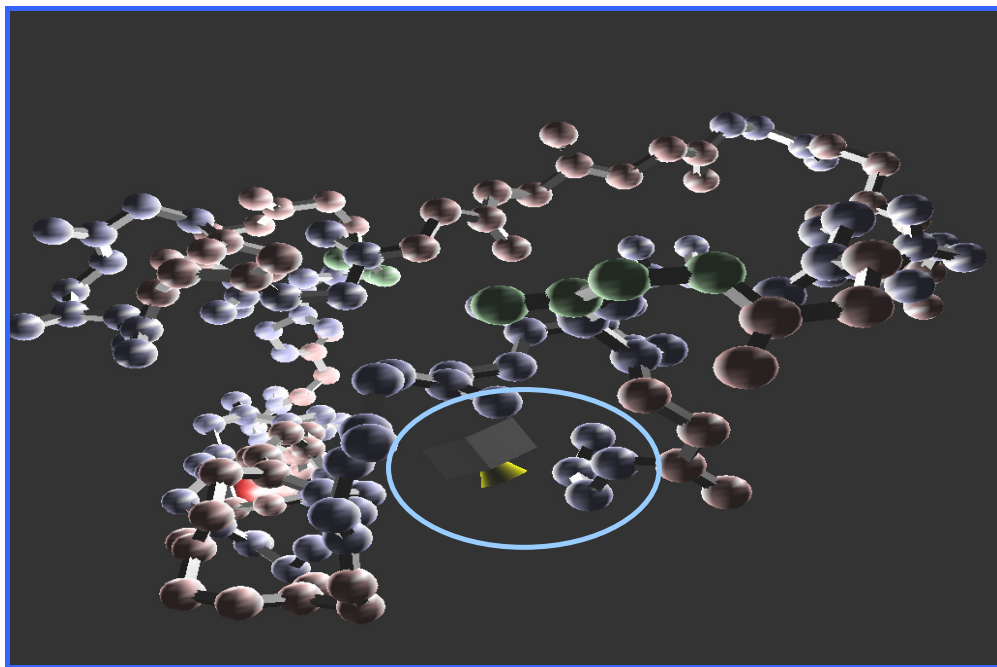


Figure 52: Pick the bond and rotate



Figure 53: Move command and Rotation command

There are total 3 commands for the protein visual interaction. They are the move, the rotate and the pick commands. Figure 50 and Figure 53 illustrate the execution of 3 commands. The move command will translate the 3D selected object. The rotate command can turn the 3D object according to the user's hand motion. The pick command locks selected 3D object and allows further manipulation to be applied on the object. The commands are produced by different hand gesture from the P5 glove and can be apply simultaneously to create a friendly interactive feel to the user.

6.3. Evaluation

The hardware integration of the Formula Pro GTR steering wheel is easy. The steering wheel comes with an USB interface that enables direct connection to the computer. The setup is also straightforward but the steering wheel is quite bulky and larger space is needed to accommodate it. Some other disadvantages of the steering wheel include (1) the stiff control buttons and (2) the need to calibrate the wheel.

The software integration of the steering device is relatively simple through the use of the 3D interaction module. Speed variation from the amount of steering wheel turn also enhances the effect of realism when moving along the protein molecules. The rollercoaster's effect is excellent to view the ribbon structure of the protein molecules. The viewpoint travelled to turns along the protein's ribbon structure

and this journey is regarded as it is travel along a 3D fixed path which gives the user a better orientation of the protein molecules.

The hardware implementation and setup of P5 glove is straightforward. The usage of the P5 glove requires some time to pickup the skill but it is rather easy to learn. The P5 glove operates with less accuracy and range than convention glove devices. It is also relatively insensitive to the user's swift response. In addition, it does not provide force and tactile feedback to the user.

The software integration of the P5 glove is simple and direct. The overall movement of the virtual hand which is the cone is present and pleasant.



Figure 54: Protein surface interaction using P5

Hardware game device used	Price
P5 glove	US\$89
Steering Wheel	US\$45
Total cost	US\$134

Table 11: Tactile interface development hardware cost

Hardware glove device	Cost Price	Cost Reduce	Reduce%
VPL Research Inc. (DataGlove)	US\$11700	US\$11566	98.85
Exos Dextrous (Hand Master)	US\$17400	US\$17266	99.23

Virtual Technologies (CyberGlove)	US\$9800	US\$9666	98.63
Fakespace (PINCH Gloves)	US\$2000	US\$1866	93.30
Fifth Dimension Technologies (5DT DataGlove)	US\$3495	US\$3361	96.17

Table 12: Tactile interface development cost comparison

Table 11 shows the hardware cost used in the tactile interface development and Table 12 illustrates a cost comparison when using commercial glove device. On average, there is a 97.24% reduction the in hardware cost.

7. VR-enhanced Bio Edutainment Application

The application exemplifies the use of inexpensive game device and the 3D interaction module to integrate into the VR-enhanced bio edutainment.

7.1. VR-enhanced Bio Edutainment Technology

VR-enhanced edutainment technology is designed for life science learning. In particular, VR motion tracking is used to enhance the modeling work; VR stereographic viewing is applied to produce 3D immersive visualization; and VR interaction is integrated with the game devices.

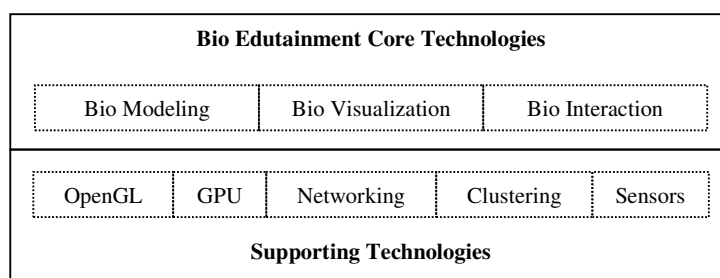


Figure 55: The core bio edutainment technologies and supporting technologies

For this purpose, three major components are designed for a VR system. The first part is a computational engine developed to serve various functions for modeling of the bio-molecular 3D world including protein structure and human interaction (Bio Modeling). The second part is a stereo native visualization in order to support the real-time and photo-realistic display and interaction of the bio-molecular 3D VR world (Bio Visualization). The third part is a VR interface which consists a series of interactive game devices to provide a natural communication between the virtual and real worlds (Bio Interaction). The Bio interaction is the interfacing layer that built on top of the 3D interaction module. The three components are seamlessly integrated thus enabling an edutainment solution for bio-molecular learning.

7.2. Implementation

The VR-enhanced bio edutainment is an application that allows user to learn molecular biology from a reach-in perspective. It is designed to give user 3 different reach-in experiences with the bio-molecular world. They are the navigation, the manipulation and the interaction. Navigation permits user to move around in the bio-molecular world, manipulation enables the user to examine the 3D molecular structure, and interaction to the molecular structure or an avatar makes learning more interesting and fun.

A game pad, a steering wheel and a P5 glove are devices that used to provide user's control to the bio-molecular world. They are popular gaming device commercially available in the market. Using those devices help to reduce the overall costs of the bio game system.

The game pad and steering wheel are integrated into the VR-enhanced bio edutainment to allow user to navigate inside the bio-molecular world. They translated the left and right, forward and backward, and up and down directional information into navigation data to be interpreted by the application. User uses those devices to move around inside the molecular structure.

The P5 glove is added to let user manipulates the bio-molecular structure. The glove information is computed in real time and mapped into the position and orientation of the bio-molecular structure. Whenever the user rotates or moves the hands, the 3D molecular structure will move according. Therefore user can inspect the molecular structure from different views.

The game pad and P5 glove are also use together to offer an interaction experience with the bio-molecular world. Interactive commands are generated through those device's buttons and hand's gestures. The molecular structure will respond to each interactive command given according. User is allowed to change the molecular structure's representations, hide the secondary structures, and control the avatar actions.

One of the major issues, however, is that those devices are usually designed for 2D interaction. To enable 3D interaction in the immersive VR environment, the software is developed to handle 3D problems by integrating physical modeling. For instance, collision detection is activated when gamers are walking through the 3D protein structure. Mathematically, this involves heavily with matrix transformation and quaternion operation.

The application hardware's interface is designed to support plug-&-play. Most of the game devices now have their USB version and some of them support wireless using infrared. The VR-enhanced bio edutainment can accept any game pad and steering wheel devices by simply connected them into the computer during application's runtime.

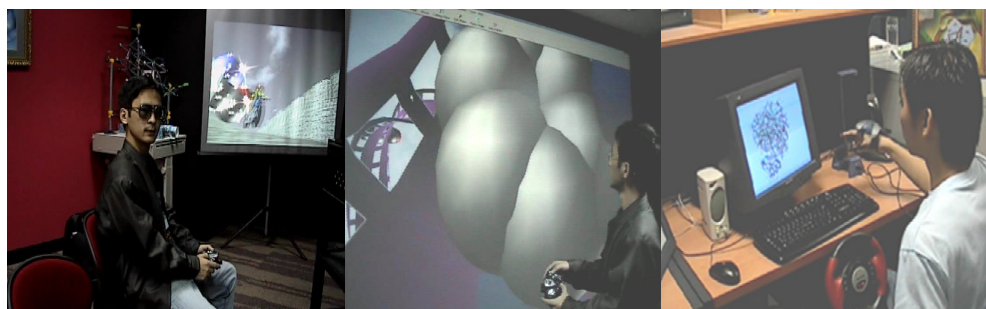


Figure 56: Game devices used in the bio games

Figure 56 shows the game devices used in the bio games. In the left, a game pad is used to control avatar's animation movement interacting with the molecular structure. In the middle figure, a wireless game pad is used to navigate the rollercoaster ride along the bio-molecular structure. In the right figure, a steering wheel and game glove are combined to navigate and manipulate the bio-molecular structure respectively.

7.3. Evaluation

Using VR, nano-scale is not a physical obstacle for users to enter the bio-molecular world. Reach-in of the nano-scale protein structure offers newer way to learn molecular biology. To assist the learning of pathogens, we further incorporate the

concept of pathogen host or carrier in the reach-in based protein learning. For instance, chimpanzees are identified as the host of HIV virus. Ideally, the reach-in should be combined in order to better understand the bio-molecular structure [55].

In order to motivate the interests of studying, avatars is introduced, which can help students to understand the protein functions and structures. Those avatars can be the hosts of the pathogens, or some media important to the functions of the proteins. In order to make the reach-in technologies more meaningful, two navigation modes are created for navigating in the special site of the protein structures. They are the “Go-to” and the “Roaming” modes that carry users directly to the specific site of the amino acid sequence.

The application has proven the used of inexpensive game devices in the VR-enhanced bio edutainment. The Bio interaction handles all the game devices interfacing. It allows different game devices to be incorporated through plug-&-play into the application without any code modification. It also permits more than one game device to be used simultaneously to control a single object. The software integration of the Bio interaction is relatively simple.

Moreover, the hardware cost is greatly reduced when using game devices. When compared to commercial glove devices, the average cost different is about 96.72%. Table 13 lists the game devices being used while Table 14 makes a comparison of the cost between the game devices and the glove devices.

Hardware game device used	Price
P5 glove	US\$89
Steering Wheel	US\$45
Game-pad	US\$25
Total cost	US\$159

Table 13: VR-enhanced Bio edutainment hardware cost

Hardware glove device	Cost Price	Cost Reduce	Reduce%
VPL Research Inc. (DataGlove)	US\$11700	US\$11541	98.64
Exos Dextrous (Hand Master)	US\$17400	US\$17241	99.09

Virtual Technologies (CyberGlove)	US\$9800	US\$9641	98.38
Fakespace (PINCH Gloves)	US\$2000	US\$1841	92.05
Fifth Dimension Technologies (5DT DataGlove)	US\$3495	US\$3336	95.45

Table 14: VR-enhanced Bio edutainment cost comparison

8. Collaborative Game

The application demonstrates the usage of the P5 glove and 3D Interaction module in the online collaborative game context.

8.1. Game design

This game is an interactive network game that allows players from all over the world to challenge each other. It is a simple guessing game which work by deducing what the other play will display (Scissor, Paper and Stone) and counter act on it. The rule of the game is as follows:

1. Stone beats Scissors
2. Scissors beats Paper
3. Paper beats Stone.

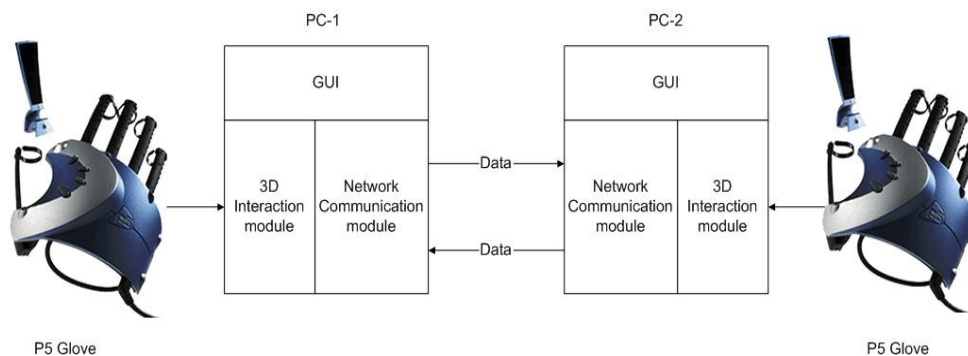


Figure 57: Online virtual reality game design

Figure 57 shows the online game design architecture. The game comprises of three modules, the Graphics User Interface (GUI), Network Communication module and 3D Interaction module.

The GUI is created to interface with the two sub-modules. Its basic function is to present virtual reality to the user during game play. Figure 58 shows the main GUI form.



Figure 58: Main GUI form

The Network Communication module is designed to facilitate the transmission of data between two or more computers. Information is sent using user datagram protocol (UDP) packet to the desire computer.

The 3D interaction module allows the P5 glove to integrate into the virtual reality game. The module read in the finger bending information of the P5 glove to the PC and the information is decrypted to one of the symbol either a scissor, paper or stone.

The virtual reality game is designed to play in two different modes, “Single player” or “Dual Player”. The single player mode allows user play the game with the computer and dual player mode lets user play the game with a remote host. The game is designed for Windows platform and it is based on graphics user interface concept. Figure 59 and Figure 60 showed the user interface of each mode.



Figure 59: Dual Player mode GUI



Figure 60: Single Player mode GUI

8.2. Implementation

The P5 glove is integrated into the game to capture the player hand gesture. The player's finger bending information is collected in real time and analyzed to determine which symbol the player have showed (Scissor, Paper and Stone). This information is sent over the network to the remote PC running the same application for comparison. The data are compared to determine who the winner is.

8.3. Evaluation

The application has fully demonstrated the used of inexpensive game device on virtual reality game. The incorporation of the game device is straightforward. The P5 glove fulfils the virtual reality game's requirement. Other game devices can also be used to replace the P5 glove without much modification in the application.

The average cost saved is 96.33% when compared with the expensive glove devices. Table 15 and Table 16 show the total hardware cost spent by the application and a comparison in the hardware cost if the application used those expensive glove devices.

Hardware game device used	Price
P5 glove x 2	US\$89
Total cost	US\$178

Table 15: Collaborative game hardware cost

Hardware device	Cost Price	Cost Reduce	Reduce%
VPL Research Inc. (DataGlove)	US\$11700	US\$11522	98.48
Exos Dextrous (Hand Master)	US\$17400	US\$17222	98.98
Virtual Technologies (CyberGlove)	US\$9800	US\$9622	98.18
Fakespace (PINCH Gloves)	US\$2000	US\$1822	91.10
Fifth Dimension Technologies (5DT DataGlove)	US\$3495	US\$3317	94.91

Table 16: Collaborative game cost comparison

9. Conclusions and Future work

9.1. Conclusions

3D Interfacing Environment

There are many different types of 3D interfacing devices available in the market for users to communicate with the application. Each device has its own advantages and disadvantages. They are classified as desktop input device, tracking device and direct human input device according to their specific use. Desktop input device is the most common device for interfacing with the computer. The tracking device is a passive device which requires continuous data generation and is mainly used to track body motion, eye and hand movement. Direct human input device is application specific device used to obtain the human body information.

Programming the application to use these devices is difficult and expensive. This involves interfacing with the hardware device's driver and processing the raw data. This will increase the cost of development and also the developing cycle. In addition, generalization of software design is difficult to realize.

Interactive Interfacing Game Devices

Keyboard and mouse are the most common devices in computer game world. Joystick, steering wheel and game-pad are used for specific games that require continuous data input to be captured. The P5 data glove is a cost-effective 3D input capturing device that is suitable for 3D interactive gaming and virtual environments.

Integrating the application to use these devices is relatively simple and less costly owing to the common set of interfaces they have. Designing a generic system to acquire input data from these devices is also easy. This greatly reduces the development cost and developing cycle.

Develop a Reusable and Inexpensive Tactile Interfacing Solution for 3D Environment

The 3D interaction module and the commercial game devices provide a total solution for the present problem the 3D environment devices faced, which is the high hardware and development cost. The solution can be used to replace the expensive 3D devices found in the market with the less costly gaming devices. In this case, the cost of the hardware devices can be substantially reduced as well as its development cycle shortened. Moreover, the implementation is simple and the data acquisition is straightforward without the need of the application to handle it explicitly.

Apply the Interactive Solution for Bio-edutainment

VR-enhanced edutainment technology is designed for life science learning. There are three components found in the VR system, the Bio Modeling, Bio Visualization and Bio Interaction. With the help of bio-edutainment, bio-molecular structure can be studied and understood in an immersive, meaningful and fun manner.

9.2. Contribution

The main objective of the research is to reduce the cost of hardware input device that used in the 3D environment. Overall hardware cost is greatly reduced when employed the solution for 3D environment. An average of 96% of cost reduction is achieved using game devices as compared with the commercial glove devices. The following are the contributions made during the course of the research:

The author intensively studied various VR interactive devices, evaluated their functions, and discussed their advantages and disadvantages.

The author narrowed down the game devices for the possible use in 3D interaction.

The author did a system design for integrating game devices in the 3D environment.

The author implemented the 3D interactive module using C++ programming as well as low level device programming. The 3D interactive module is a device independence and data integration system. Irregardless of which device and how many device is being used to acquire the data, the data will be merged into a common data that are accessible to the application. Therefore from the application perspective, it only sees a common set of data that are required for processing. With the approach of common data, device with few degrees of freedom can be enhanced by adding additional device without the application noticed.

The author developed three different applications that used the solution. By integrating commercial game devices into these applications, the cost of hardware is greatly cut down. A summary of each application and the device being employed are listed.

Tactile Interface Development

A steering wheel and a P5 glove were used to facilitate the graphic manipulation and protein visual interaction. Total cost US\$134. An average of 97.24% cost reduction.

VR-enhanced Bio Edutainment

A game pad, steering wheel and P5 glove were employed in the VR-enhanced Bio Edutainment. Total cost US\$159. An average of 96.72% cost reduction.

Collaborative Game

Two P5 gloves were utilized to demonstrate an online virtual reality game. Total cost US\$178. An average of 96.33% cost reduction.

9.3. Future work

Platform independence

The 3D interaction module should be developed to support different platform of operating system. At present, the module only works in window operating system. With the increased popularity of Linux system and other operating system, the 3D interaction module should develop under platform independence framework. One of the recommended frameworks is the Microsoft dot net framework.

Multiple platforms and network support

Beside platform independence, the 3D interaction module should be modified to support networking and integrate multiple platforms data. Irregardless of which platform and how many platform is being used to acquire the input data, it should be able to collect and combine all the input data into the common data through the network. Therefore deployment of applications can be distributed in different areas.

Force feedback and sound

Force feedback should also be included into the module. This provides the user with a more realistic experience when he or she feels the resistance generated by the device. Last but not least, sound effect should also be considered. Both the force feedback and sound effect implementation are not difficult through DirectX.

P5 glove

As for P5 glove future development, the Essential Reality has already indicated the areas it intends to pursue in the future for its second generation version of the P5, and are currently working on a wireless glove, a left-hand version of the glove, plus additional feature enhancements such as tactile feedback and a pulse sensor.

References

1. Jurek Kirakowski. Human Computer Interaction from voltage to knowledge. Chartwell-Bratt Studentlitteratur, 1988, 5, 40.
2. Ben Knapp & William Putnam. Input/Data Acquisition System Design for Human/Computer Interfacing.
3. Hewett, Baecker, Card, Carey, Gasen, Mantei, Perlman, Strong and Verplank. ACM SIGCHI Curricula for Human-Computer Interaction.
4. GARY A. BERG. Human-Computer Interaction (HCI) in Educational Environments: Implications of Understanding Computers as Media. JI. of Educational Multimedia and Hypermedia (2000) 9(4), 347-368
5. Stephen Hill. A Practical Introduction to the Human-Computer Interface. DP Publication Ltd. 1-47.
6. Carey, J. M. (Ed). (1991). Human factors in information systems: An organizational perspective. Norwood, NJ: Ablex.
7. Card, S.K., Moran, T.P., & Newell, A. (1983). The psychology of human-computer interaction. Hillsdale, NJ: Lawrence Erlbaum.
8. Head, A. (1999). Design wise: A guide for evaluating the interface design of information resources. Medford, NJ: Cyberage Books.
9. Faulkner, C. (1998). The essence of human-computer interaction. New York: Prentice Hall.
10. David E. Kieras. (1988) Handbook of Human-Computer Interaction: Chp 7- Towards a Practical GOMS Model Methodology for User Interface Design. 135.
11. Shneiderman, B. (1997b). Direct manipulation for comprehensible, predictable and controllable user interfaces. Proceedings ACM International Workshop on Intelligent User Interfaces. 97, ACM, New York, NY, 33-39. (<http://ftp.cs.umd.edu/pub/hcil/Reports-Abstracts-Bibliography/97-01.html>)
12. Haber, N., & Hersenson, M. (1973). The psychology of visual perception. New York: Holt, Rinehart and Winston.
13. Baecker, R. & Small, I. (1995). Animation at the interface. In B. Laurel, (Ed.). The art of human-computer interface design. Reading, MA: Addison-Wesley.
14. Bederson, B.B. (1998). Does animation help users build mental maps of spatial information? Computer Science Department Human-Computer Interaction Lab. University of Maryland. Unpublished.

15. Morris, J.M., Owen, G.S., & Fraser, M.D. (1994). Practical issues in multimedia user interface design for computer-based instruction. In S. Reisman (Ed). Multimedia computing: Preparing for the 21st century. Harrisburg, PA: Idea Group.
16. Shneiderman, B. (1999b). Supporting creativity with advanced information-abundant user interfaces. Position paper for National Science Foundation & European Commission meeting on human-computer interaction research agenda, Toulouse, France. To be published in book form. (<ftp://ftp.cs.umd.edu/pub/hcil/Reports-Abstracts-Bibliography/99-16html/99-16.html>)
17. Pickover, C.A. (1991). Computers and the imagination. New York: St. Martin's Press.
18. Feifer, R.G. (1994). Cognitive issues in the development of multimedia learning systems. In S. Reisman (Ed). Multimedia computing: Preparing for the 21st century. Harrisburg: Idea Group Publishing.
19. ALAN DIX, JANET FINLAY, GREGORY ABOWD, RUSSELL BEALE. HUMAN-COMPUTER INTERACTION. Prentice Hall, 1993. 49-67.
20. Brad A. Myers. "A Brief History of Human Computer Interaction Technology." ACM interactions. Vol. 5, no. 2, March, 1998. pp. 44-54.
21. Kathleen M. Potosnak. (1988) Handbook of Human-Computer Interaction: Chp 21- Keys and Keyboards. 476-479.
22. Joel S. Greenstein. (1988) Handbook of Human-Computer Interaction: Chp 22- Input Devices. 504-510.
23. Cox B.J.; Object-Oriented Programming An Evolutionary Approach; Addison-Wesley Publishing Company, Reading, MA 1986.
24. Peter E. Jones. Three-dimensional input device with six degrees of freedom. Electrical and Electronic Engineering, The University of Western Australia, Nedlands, WA 6907, Australia.
25. Foley, van Dam, Feiner, Hughes. Computer Graphics Principle and Practice. Addison Wesley
26. Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, JR, Ivan Poupyrev. User Interfaces Theory and Practice. Addison Wesley