Mixed Reality Entertainment Systems for Social and Physical Human Interaction

Teo Hui Siang, Jason
(B.Eng.(Hons.), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2005

Abstract

Computer gaming has become a dominant force in recent years due to its higher level of attractiveness to game players. The main advantage it has over traditional games is its seemingly unlimited ability to create graphical illusions to satisfy the fantasies of imaginative minds. However, most of the time, playing computer games decreased physical activities and social interactions.

Tangible mixed reality gaming is one of the new form gaming that was introduced to assume a prominent role in fusing the exciting interactive features of computer gaming with the real physical world. In this thesis, we will explore how tangible mixed reality gaming moves away from conventional keyboard and mouse interface that bound users to their desks and chairs. We will also introduce two tangible mixed reality games that were developed during this course of study, namely, Human Pacman and Magic Land. Both are novel interactive entertainment systems that venture to embed the natural physical world seamlessly with a fantasy virtual playground by capitalizing on new aged technologies such as high speed local area network, motion tracking technology and more.

Acknowledgement

I would like to express my heartfelt thanks to the following people for their invaluable guidance and assistance during the course of my work.

- Dr Adrian David Cheok
- Ms Tracy Lim
- Ms Veron Ng
- Ms Liu Wei
- Mr Teo Sze Lee
- Mr Lee Shangping
- Ms Li Yu
- All others from Mixed Reality Laboratory (Singapore) who have helped me in one way or another.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Interactivity plays a central role in today's mainstream computer game design because people enjoy entertainment that they can control and feel fully involved [1]. However, the fact remains that many computer games today are still limited in physical and social interaction. Natural interactions, such as physical movement, behavioral engagement and cognitive states are also lost [2]. In these games, the players' attention is focused mainly on the computer screen or 2D/3D virtual environments, and players are bounded to the use of keyboards and mouse while gaming, therefore constraining interactions. However, as pointed out by Bowlby [3], Human, as social creatures, find physical interaction, touch, and human-to-human presence essential for the enjoyment of life. Hence, the need to extend digital gaming beyond standard desktop environments is well recognized by many.

In the early 1980s, Human computer interaction (HCI) arose as a field from intertwined roots in computer graphics, operating systems, human factors, ergonomics, industrial engineering, cognitive psychology, systems part of computer science and possibly more. One of the objectives of advanced research in HCI is to fill in this gap by bringing more physical movements and social interactions into games while still utilizing the benefit of computing and graphical systems. Tangible mixed reality gaming is one of the new form in gaming that was introduced to assume a prominent role in fusing the exciting interactive features of computer gaming with the real physical world.

In this thesis, we will explore how tangible mixed reality gaming moves away from conventional keyboard and mouse interface that bound users to their desks and chairs, and highly promotes social and physical interaction. In particular, we will look at two HCI inspired mixed reality games in the later chapters that were developed during this course of study. In the subsequent part of this chapter, a list of publications on all the work will be given. Then, the prelude to subsequent chapters of this thesis will follow.

1.1 List of Publications

Conference

- Adrian David Cheok, Kok Hwee Goh, Wei Liu, Farzam Farbiz, Sze Lee Teo, Hui Siang Teo, Shang Ping Lee, Yu Li, Siew Wan Fong, Xubo Yang, "Human Pacman: A Mobile Wide-Area Entertainment System based on Physical, Social and Ubiquitous Computing.", *Proceedings of International Conference on*

Advances in Computer Entertainment Technology 2004, Singapore (pg 360-361)

- Adrian David Cheok, Kok Hwee Goh, Farzam Farbiz, Wei Liu, Yu Li, Siew Wan,Fong, Xubo Yang, Sze Lee Teo, "Human Pacman: A Wide Area Socio-Physical Interactive Entertainment System in Mixed Reality.", *Extended abstracts of the 2004 Conference on Human Factors in Computing Systems, CHI 2004, Vienna, Austria (pg 779-780)*

- Tran Cong Thien Qui, Ta Huynh Duy Nguyen, Asitha Mallawaarachchi, KeXu,Wei Liu, Shang Ping Lee, ZhiYing Zhou, Sze Lee Teo, Hui Siang Teo, Le Nam Thang, Yu Li, Adrian David Cheok, Hirokazu Kato, "Magic Land: Live 3D Human Capture Mixed Reality Interactive System.", *Extended Abstract Sessions of the 2005 Conference on Human Factors in Computing Systems, CHI 2005, Portland, Oregon USA (pg 1142-1143)*

- Kok Hwee Goh, Hui Siang Teo, Sze Lee Teo, Wei Liu, Farzam Farbiz, Yu Li,Shang Ping Lee, Siew Wan Fong, Xubo Yang, Adrian David Cheok, "Does Pacman Need a Helper?" Analyzing Experience of Physical and Social Interactivity in a Mixed Reality Entertainment Environment.", *Extended Abstract Sessions of the 2005 Conference on Human Factors in Computing Systems,* *alt.CHI 2005, Portland, Oregon USA (pg 2182-2191*

Journal

- Adrian David Cheok, Kok Hwee Goh, Wei Liu, Farzam Farbiz, Siew Wan Fong, Sze Lee Teo, Yu Li, Xubo Yang, "Human Pacman: A Mobile, Wide-Area Entertainment System Based on Physical, Social, and Ubiquitous Computing.", *Personal and Ubiquitous Computing Journal, Volume 8, Issue 2 (May 2004) (pg 71-81)*

- Ta Huynh Duy Nguyen, Tran Cong Thien Qui, Ke Xu, Adrian David Cheok, Sze Lee Teo, ZhiYing Zhou, Mallawaarachchi Asitha, Shang Ping Lee, Wei Liu, Hui Siang Teo, Le Nam Thang, Yu Li, "Real Time Mixed Reality 3d Human Capture System for Interactive Art and Entertainment ", *IEEE Transaction On Visualization And Computer Graphics (TVCG), November / December Issue Of 2005.*

Invited Paper

- Adrian David Cheok, Ta Huynh Duy Nguyen, Tran Cong Thien Qui, Sze Lee Teo, Hui Siang Teo, "Future Interactive Entertainment Systems Using Tangible Mixed Reality", *International Animation Festival, Hangzhou, China,* *2005.*

Workshop

- Tran Cong Thien Qui, Ta Huynh Duy Nguyen, Adrian David Cheok, Sze Lee Teo, Ke Xu, ZhiYing Zhou, Asitha Mallawaarachchi, Shang Ping Lee, Wei Liu, Hui Siang Teo, Le Nam Thang, Yu Li, Hirokazu Kato, "Magic Land: Live 3D Human Capture Mixed Reality Interactive System", *International Workshop: Re-Thinking Technology in Museums: Towards a new understanding of visitors experiences in museums, June 2005*

Media Article

- TODAY (23rd June 2004) – English newspaper in Singapore
- Lian He Zao Bao (14th June 2004) – Chinese newspaper in Singapore
- The New Paper (9th August 2004) – English newspaper in Singapore
- NewScientist.com news service (18th November 2004) – online article
- BBC News UK edition (6th June 2005) – online article
- MTV.com (23rd June 2005) – online article
- Lian He Zao Bao (11th June 2005) – Chinese newspaper in Singapore
- Sunday Times (3rd July 2005) – English newspaper in Singapore

1.2 Prelude to Subsequent Chapters

Chapter 2 will cover some background literature on social and physical interaction and details on its importance. A general introduction to virtual, augmented and mixed reality will also be covered in this chapter. This will be followed by some discussions on the current limitations in mixed reality gaming. Finally, some examples of how mixed reality is applied in some earlier entertainment systems. This will be generally classified into outdoor and indoor augmented reality games.

Chapter 3 will look at Human Pacman, a novel interactive entertaining system that ventures to combine a virtual playground seamlessly with the real physical world. Human Pacman is one of the projects that was completed during this course of study. At the start of this chapter, we will look at the interesting varieties of Pacman games that have been made and the novel aspects of Human Pacman which makes it stands out. These include mobile gaming, ubiquitous computing, and virtual and physical interactions.

Next, we will give a detailed discussion of the game play for Human Pacman. Apart from the Pacman and Ghost characters which already existed in the original Pacman, we have added two other roles known as Pacman helper and Ghost helper. The reasons for the inclusion of the helpers will be discussed. This is followed by the overall system design, hardware and software of the system. Finally, we will present the user study that was conducted and evaluate how the Human Pacman system fits into the interaction theme.

Chapter 4 will look at Magic Land, another tangible mixed reality entertainment system, where captured 3D avatars of human and computer generated 3D virtual animations are able to play and interact with each other. In this chapter, we will first highlight the technical achievements and contributions of Magic Land to the field of research.

Next, we will cover the game play and see how this system utilizes mixed reality technology and tangible interfaces that differs radically from conventional computer entertainment systems. This will be followed by the system design, hardware and software.

Chapter 5 summarizes the thesis, highlighting the novel aspects of Human Pacman and Magic Land. The chapter ends with a conclusion to future possibilities tangible mixed reality entertainment entails.

Appendix A is a tutorial for setting up and using the wearable computer in Human Pacman. A detailed description of the connections between the individual modules within the wearable computer will be given. Settings for configuring the system follow and the chapter ends off with a step-by-step guide to starting the Human Pacman program.

Appendix B will describe the setting up of Ceiling Tracker for Magic Land. This includes the hardware and the software setting up.

Appendix C will briefly outline the Human Pacman demos done at CHI 2004 and ACE 2004.

Appendix D shows newspaper articles featuring Human Pacman.

Chapter 2

Background Literature

2.1 Social and Physical Computing

2.1.1 Introduction

Human as social creatures find physical interaction, touch, and human-to-human presence essential for the enjoyment of life [3]. In pre-computer age, interactions involves physical movement, behavioral engagement and cognitive states. However, with the pervasion of computers and the continual propagation of digital communication and entertainment, there are many changes in societal psyche and lifestyle. Many computing applications focus the user's attention on elaborated graphical presentation on screen in a 2D/3D virtual environment, thereby constraining physical user's interaction with her environment. Even with advancement in networking technology, physical and social interaction is still limited with no behavioral engagement and cognitive exchange.

The need to extend computing beyond standard desktop environment and to bring in more physical and social interaction, has been well recognized by many. Human Computer Interaction (HCI) has been a key issue of research starting in the early 1980s. One of the aims of advanced research in HCI is to fill in the gap and bring more physical movements and social interactions into computing. To date, HCI practitioners have been fairly successful in bridging this gap and compared to the early years of computing there has been a shift from technical to more human centered perspective nowadays

2.1.2 Physical and Social Interactions in Computer Entertainment

In relation to the work done in this thesis, we will look at physical and social interactions in computer entertainment. The digital entertainment industry is the third largest industry, just after Hollywood movie industry and recording industry [4], with a huge number of players around the world. However, like many typical computer applications, the development of computer games often decreased physical activities and social interactions.However, there has been a growing interest in physical gaming. The pioneers in physical gaming have been arcade games like Dance Dance Revolution [5] and Para Para Paradise [6] (see Figure 2.1), in which players are required to dance in sync with music and graphical objects. More recently, Sony has released their EyeToy [7] camera which is the first product that allows physical games to be played on a game console (see Figure 2.2). Nonetheless,

Figure 2.1: ParaParaParadise

these games still require players to be in a certain area in front of game machine and focus on the screen therefore constraining their physical movements.

Social interactions in computer entertainment has also been widely studied and in the independent researches from XEODesign [8] and the Entertainment Software Association (ESA) [4], it was shown that one of the top reasons why people play games and especially play games together is the human factor. People enjoy playing with others inside and outside the game. They see games as a mechanism for social interaction. Teamwork flourishes as players pursue shared goals. Rivalry runs hot when they compete. After all, there is no opponent like a human opponent as no current computer models can bring the experience and rival the richness of human

Figure 2.2: A little girl playing Sony EyeToy.

interaction [9].

With the advancement in networking technology, especially the continuous growth in bandwidth of the internet access of each household, multi-player online games have gained popularity and helped players from distant areas overcome the spatial barrier to play with and against each other. However, as mentioned earlier, in such a networked game setting, social interaction between players is limited since natural interactions which are native to human being like behavioral engagement and cognitive states are lost [2].

2.1.3 Future Development of Computing

In general, HCI has been a key issue with continuous growth in importance up to now and means by which humans interact with computers continues to evolve rapidly. Embodied interaction [10] in computing is generally regarded as the next generation computing paradigm. “Embodiment” is the way physical and social phenomena unfold in real time and real space as a part of the world in which we are situated, right alongside and around us. It is in the center of phenomenology which explores our experiences as embodied actors interacting in the world, participating in it and acting through it, in the absorbed and unreflective manner of normal experience.

Embodied computing involves the elements of ubiquitous computing, tangible interfaces and interaction and social computing. It moves the computer interface away from the traditional keyboard and mouse and into the environment. The three important research paradigms on which embodied computing is founded, are Weiser's ubiquitous computing [11], Ishii's tangible bits or "things that think" [12], and Suchman's sociological reasoning to problems of interaction [13]. Weiser's philosophy of ubiquitous computing is derived from two observations: "That the most successful technologies are those which recede into the background, and become an unnoticed feature of the world we live in", and secondly, "That computing power is becoming so small and so cheap that it is now really possible to embed computing devices in almost every object and every facet of our physical environment". Weiser inferred from these two observations that computation would recede into the environment, allowing new possibilities and novel uses of computing. In essence, the environment becomes a distributed computer and responds to people's needs and actions in a contextual manner.

Ishii's vision of tangible bits or "things that think" has its origins in Weiser's work in terms of embedding computing in the environment, but builds on it with the observation that we operate in two different worlds. These two worlds are the computational world and the world of physical reality. Ishii termed these two worlds the world of "bits" and "atoms". Through the tangible bits approach, Ishii has set out to bring these two worlds together, and allow the computational world to engage and employ physical and tactile skills that we are intimately familiar with.

It can be seen that ubiquitous computing deals with computing in the environment and with activities that take place in the context of the environment. Tangible interaction deals with using the physical world and physical object manipulation to interact with the digital world. Both share the viewpoint that interaction with computers should exploit our natural familiarity with the physical environment and physical objects. Both tie the computer interaction with physical activities in such a manner that the computer is embedded in the activity. In this way, the environment and physical objects become the computer interface.

The third research paradigm is social computing, or the study of the context in which interaction with computation occurs. The important work of Suchman [13] on this topic draws on ethnomethodology to analyze interaction and social conduct. In ethnomethodology, social conduct is an improvised affair, which is real-time and non-linear. This perspective argues that it is the context in which an interaction takes place that allows people to find it meaningful. Experimental investigations have found that people's interaction with technology does not follow formal theoretical abstracts but are improvised in real-time [11].

2.2 Virtual, Augmented and Mixed Reality

2.2.1 Introduction

Virtual reality (VR) was a phrase originated by Jaron Lanier, the founder of VPL Research, one of the original companies selling virtual reality systems. It is a technology that encompasses a broad spectrum of ideas. Virtual Reality is defined as "a computer generated, interactive, 3-dimensional environment in which a person is immersed" and there are three key points in this definition. First, this virtual environment is a computer generated three-dimensional scene that requires high performance computer graphics to provide an adequate level of realism. The second point is that the virtual world generated is interactive. A user requires real-time response from the system to be able to interact with it in an effective manner. The last point is that the user is immersed in this virtual environment. Figure 2.3 shows an example of VR application.

Figure 2.3: VR game: CAVE Quake

However, even with the most current technology available, it is difficult to duplicate the massive information of the real world environment around us and use it to create a virtual environment that is real enough to deceive human senses. Thus, the virtual environments created for immersive entertainment and games are very simplistic. For those systems that can create a more realistic environment, such as flight simulators, they normally have a million dollar price tag.

To create a more simplistic virtual environment, an Augmented Reality (AR) system is used. It works by creating a virtual object or environment that simultaneously appears with an object or environment in the real world. This is achieved by combining a real scene and a virtual scene using a computer as seen in application in Figure 2.4.

Figure 2.4: Some Augment Reality Applications.

Augmented Reality (AR) has become a growing area in virtual reality research. An augmented reality system generates a composite view of the real scene viewed by the user and a virtual scene generated by the computer. The computer basically augments the scene with additional information. The ultimate goal is to create a system such that the user cannot tell the difference between the real world and the virtual augmentation of it. To the user of this ultimate system it would appear that he is looking at a single real scene.

In 1994, [14] describes a taxonomy that identifies how augmented reality and virtual reality work are related. The real world and a totally virtual environment are at the two ends of this continuum with the middle region called Mixed Reality (MR). Augmented reality lies near the real world end of the line with the predominate perception being the real world augmented by computer generated data. Augmented virtuality is a term created by Milgram to identify systems, which are mostly synthetic with some real world imagery added such as texture mapping video onto virtual objects. This is a distinction that will fade as the technology improves and the virtual elements in the scene become less distinguishable from the real ones.

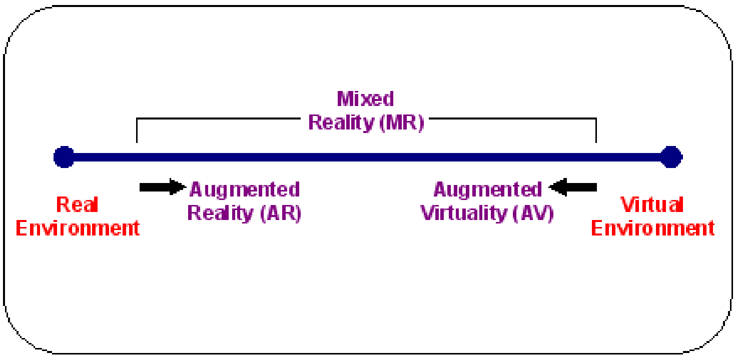


Figure 2.5: Milgram's Reality-Virtuality Continuum.

2.2.2 Mixed Reality in Entertainment

Work on tangible mixed reality computing started to appear in literatures in the early 90s with the introduction of Computer-Augmented Environments [15] that have envisioned the merging of electronic systems with the physical world instead of attempting to replace them as in virtual reality environments.

Over the years, a number of projects have explored this new paradigm of interaction termed tangible computing. Early attempts include Bishop's Marble Answering Machine [16] that has made a compelling demonstration of passive marbles as containers for voice messages; "Brick" by Fitzmaurice [17] that are essentially new input devices that can be tightly coupled to virtual objects for manipulation or for expressing action (e.g., to set parameters or for initiating processes).

With the rapid advance of computing technologies, entertainment systems these days are capable of seamlessly integrating real and virtual worlds and on top of that, provide a tangible interface with mixed reality objects. Compared to conventional computer games, tangible mixed reality gaming presents to its players more compelling experiences of physical and virtual interactions [18].

There are currently a variety of different augmented and mixed-reality entertainment systems developed, that vary in their motivations and approaches. Indoor AR gaming is the more mature domain at present even though there are also some examples of good outdoor AR gaming systems. The key reason behind this is the different position tracking technology used for indoor and outdoor AR games. Tracking is essential for all AR applications, including games, because virtual objects need to augment onto the real world environment correctly for optimal effect and performance. While it is still possible to ignore variable positional accuracy in an augmented reality user interface, this can make for a confusing system; for example, when accuracy is low, virtual objects that are nominally registered with real ones may be too far off to be useful.

To date, there are many methods that can be used for indoor tracking. The more common ways includes using vision-based camera tracking and relatively more expensive motion trackers like the InterSense IS-900. Outdoor AR games mostly rely on GPS technology to track. Unfortunately, most off-the-shelf GPS receivers are unable to provide the required tracking accuracy needed for AR applications. Hence, there are more indoor AR games are more mature in terms of development than outdoor games. In the following part of this section, a few examples of indoor and outdoor AR games will be presented.

Indoor AR Games

Figure 2.6: AR2 Hockey

One very first example of tangible augmented reality game is AR2 Hockey [19]. It is an air-hockey AR game in which users use real mallet to play with a virtual puck on a real table. This game enhances physical interactions and social communication, but does not utilize the graphical power of computer systems.

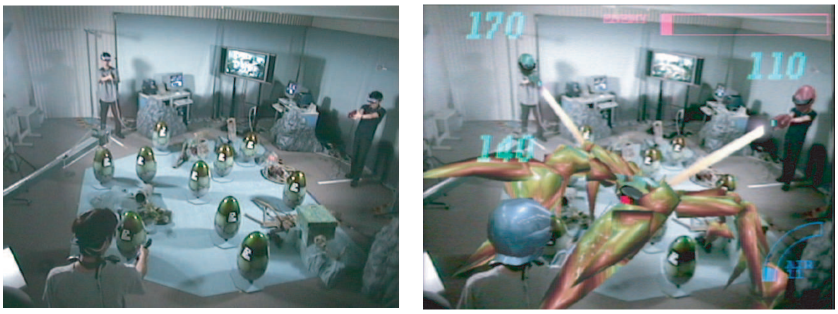


Figure 2.7: AquaGauntlet

AquaGauntlet [20] is another augmented game in which several players gather in a small place with some physical egg-shaped objects to shoot computer-generated creatures superimposed onto the real scene as if they came from these egg-shaped objects. This game enhances physical interactions and social communication, and also utilizes the graphical power of computer system. However, players of Aqua-Gauntlet, as well as AR2 Hockey, still have limited movement and little interaction with the physical space (as they must stand in a fairly constant location).

More recent computing based mixed reality game which enhances physical interactions and social communication is Touch-Space [21]. This game is carried out in the physical world with a room-size space where two players will collaboratively finish some tasks and then rescue a princess in castle controlled by a witch. This game provides different levels of interaction in different environments: physical environment, augmented reality, and virtual reality.

Tilt-pad Pacman, as shown in Figure 2.9, is another recent indoor mixed reality

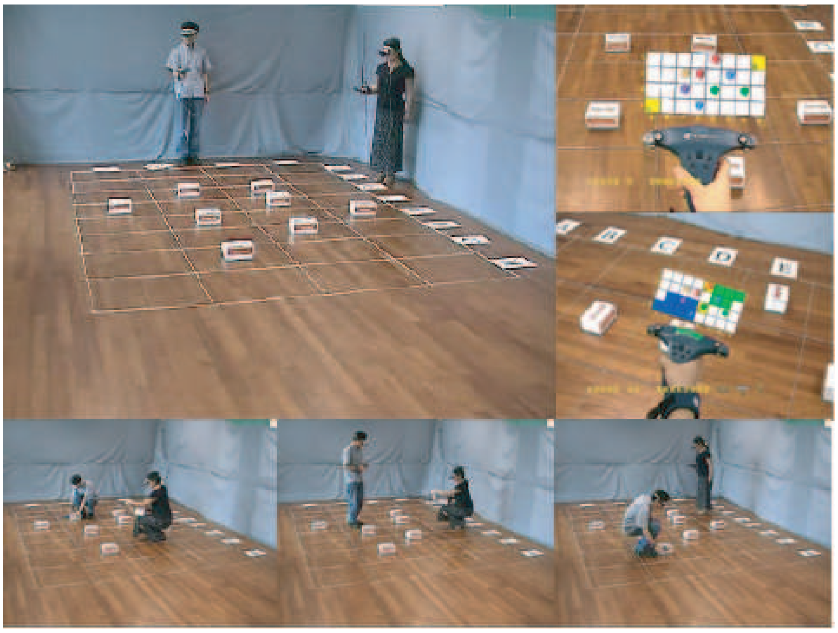


Figure 2.8: Touch Space

and tangible interaction game which exploits physical tangible interaction. It also utilizes 3D graphics rendering to create an attractive imaginative virtual world.

Outdoor AR Games

ARQuake was the first outdoor AR game, where the popular desktop video game Quake is converted to work with an AR backpack system. The game took uses positioning from a GPS unit and orientation from a digital compass, so the game environment could be explored by physically walking and looking around.

Another example of a typical outdoor mixed reality games aiming for enhancing physical activities and social interactions is “Pirates!” [22]. “Pirates!” uses hand-held computers and proximity-sensing technology to make real world properties, such as locations or objects, important elements of game mechanics. However, as

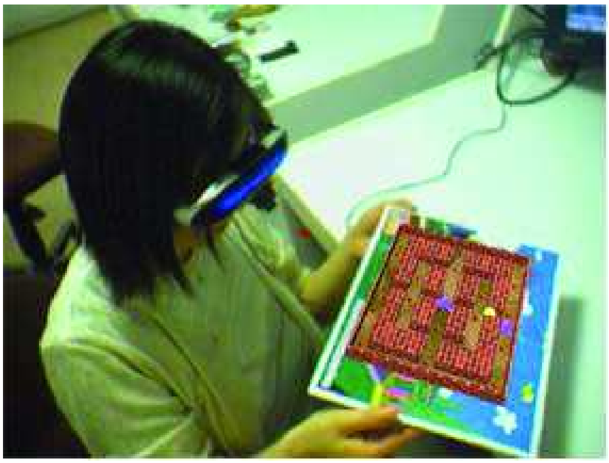


Figure 2.9: A player playing Tiltpad Pacman.

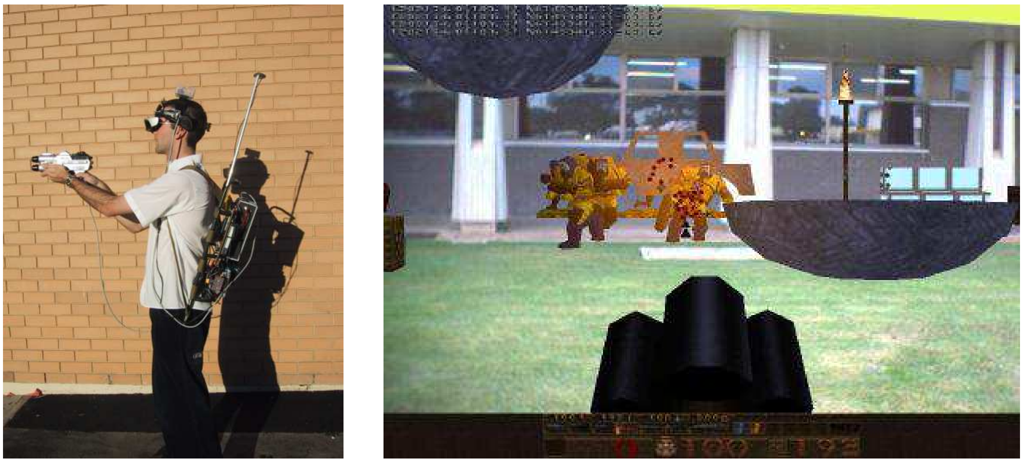


Figure 2.10: A player playing ARQuake.

"Pirates!" is played on a PDA screen, and unlike ARQuake, it does not allow any 3D mixed reality experience.

In the subsequent chapters, we analyze Human Pacman and Magic Land, two interactive systems that were developed during this course of study. Both systems are anchored on tangible mixed reality entertainment. However, Human Pacman allows multiple users to play outdoors in an AR environment, emulating the game play of the original 2D arcade game Pacman; whereas Magic Land is a kind of "free play" indoor AR game, in which players are free to use their imagination and creativity to design the game story and rules. Thus, as mentioned before, the game story and rules are not fixed but depends on players' imagination and decision.

Chapter 3

Human Pacman

Figure 3.1: The original arcade version of the Pacman game.

The origins of Pacman can be traced back to 1979 when Namco® first launch the arcade game in Japan. The game turned out to be an instant success and won the hearts of millions of arcade gamers around the world. Pacman has since gone through numerous stages of development including Ms. Pacman (1982), Pacman Plus (1982), Super Pacman (1982), Baby Pacman (1982), Jr. Pacman (1983), Professor Pacman (1983), Pac & Pal (1983), Pac-Land (1984), Pac-Mania (1986),

Figure 3.2: The 3D version of Pacman game.

Pac-Attack (1993). A VR version of the game, Pacman VR, was also introduced in 1996. A 3D version of the game, Pacman World, was introduced 3 years later in 1999. Despite the numerous variation of the game, the ultimate goal of the game remains fundamentally unchanged; Pacman needs to collect all the cookies in a maze before any ghost capture and devour him.

Human Pacman has been designed to be in close resemblance to the original Pacman in terms of game objectives so that the players' learning curves are very much leveled to the point that they can pick up the game in very little time and enjoy the associated familiarity. Human Pacman stands out from all other versions of Pacman ever made due to the injection of pervasive computing and the elements of physical and social interaction.

In this chapter, we shall first examine the novelty of the game. This will be followed by details of the actual game play and the system design.

3.1 Highlights of Human Pacman

Human Pacman is one of the few mixed reality games which epitomizes pervasive computing and several novel HCI aspects. Some novel aspects of Human Pacman are as follow:

- Mobile Gaming

  In the game, players are free to roam about, socialize and interact with each other while playing the role of Pacman or Ghost. This freedom of moving around while playing the game is rather uncommon even in modern era computer games. Most of the time, players are bounded to their computers when they play computer games. However, Human Pacman is able to utilize wearable computers and wireless communication technologies to support its "human-centered" computing design.

- Ubiquitous Computing

  There is automatic communication between wearable computers and Bluetooth devices embedded in certain physical objects used in game play. The movement and status of each individual player is tracked by a main server and disseminated real time to all other players. Hence, every players remains updated with what is happening in the game.

- Virtual and Physical Interactions

  Human Pacman takes mobile gaming to a new level of sophistication by incorporating virtual fantasy and imaginative play activity elements. These are the factors which propelled the popularity of computer game. Through the use of Head Mounted Displays (HMD) and Mixed Reality Toolkit software, virtual cookies are seamless integrated with the real world and Pacman can simply collect these virtual cookies by walking through them.

3.2 Game Play

Human Pacman is a novel interactive entertainment system that ventures to combine a virtual playground seamlessly with the real physical world. By the name itself, Human Pacman implies a human player taking the role of Pacman literally. Besides Pacman, there are other players who will assume the role of Ghost physically. The ultimate goal of the game remains in close resemblance to the original arcade Pacman.

In Human Pacman, Pacman is to collect all virtual plain cookies in Pac-World by walking through these cookies. However, he has to avoid the Ghosts while going about to achieve his goal. On the other hand, the aim of the Ghost is to capture Pacman in Pac-World. Through the use of Head Mounted Displays (HMD) Pacman and Ghost players can see virtual cookies scattered over the real physical game area, waiting to be collected by the Pacman. These cookies are displayed in the first person perspective of the player, dependent on her physical position and head motion. Hence, the player sees cookies on the footpath, as if she were a real

Figure 3.3: Pacman Collecting Virtual Cookies

Pacman in a Pacman maze.

Figure 3.3 illustrates a real scenario that occurred during a game session. In this scenario, Pacman saw and tried to collect the virtual cookie placed in front of his path. Scenes on the left show the first person views of Pacman at different instance of time. Scenes on the right show the corresponding views in the virtual world.

To add to the excitement of the game play, after "eating" a special cookie or

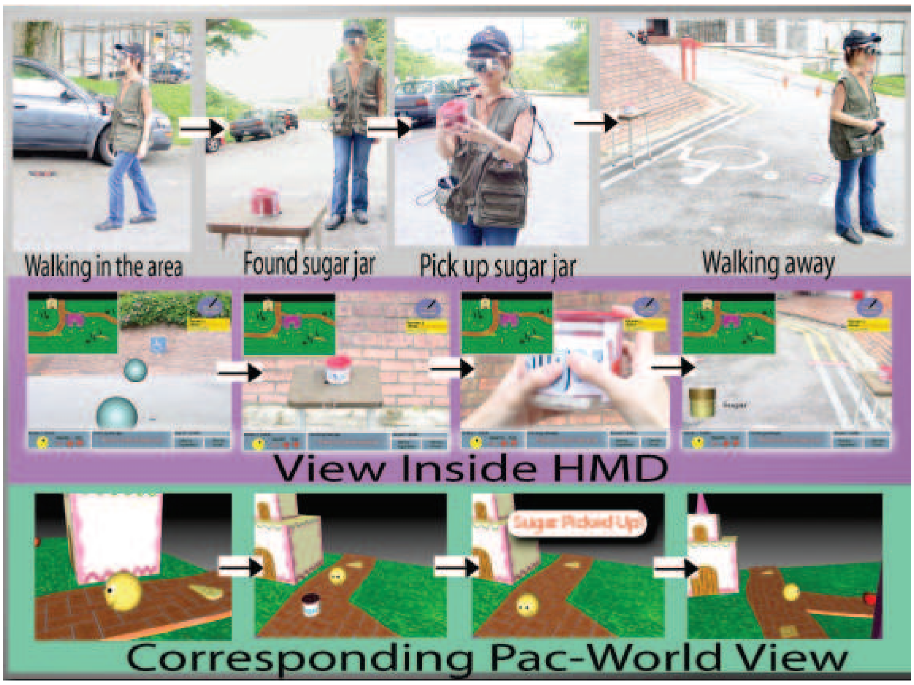


Figure 3.4: Sequence of pictures showing the collection of the special cookie.

sweet ingredient, Pacman gains "Ghost-capturing" capability for a limited period of time. In the game, these special cookies are in the form of real Bluetooth-embedded boxes hidden in different parts of the game area. The Pacman player has to hunt for this in the surrounding physical area. Having found the special cookie or sweet ingredient, collection is done simply by physically holding the object in her hands. Figure 3.4 shows a sequence of pictures illustrating how Pacman collects the Bluetooth-embedded box.

Pacman will win the game upon collecting all the virtual cookies in the game. On the other hand, Ghost player wins the game by capturing Pacman before he reaches his goal. This can be achieved by tapping on a capacitive sensor attached to the Pacman's backpack. Likewise, the Ghost can be captured by Pacman endowed with "Ghost-capturing" powers from eating a special cookie. Figure 3.5 shows how Ghost player can capture Pacman.

Figure 3.5: "Capturing" Pacman by touching the touch sensor on the backpack.

3.2.1 Extending Beyond the Physical Domain

To extend the game beyond physical domain and promote social interaction and collaboration, the Helper is an additional role that is not found in the original Pacman game. This new role enhances the game by contributing an alternate means of hybrid interaction between the real and virtual players. The Helper can connect to the game server from almost anywhere in the world through the Internet and view the game "live" in virtual reality form from any angle and distance. Every movement in the physical world will be reflected immediately in the virtual realm.

This mode of virtual viewing provides a new dimension of watching experience for participants, which is both efficient and entertaining.

The Helpers, however, are not restricted to watching the game passively. They can actually communicate with the Pacman or Ghost in real time via text messaging bi-directionally as shown in Figure 3.6. While the Helpers use computer keyboard for text-inputting, the Pacman and Ghost communicate and respond to their Helpers in the chat by using the Twiddler, a handheld inputting device. The communication that takes place could be either casual chat or discussion on the winning strategy. Such interactions could promote social cooperation and establish relationships between humans who are operating across radically different contexts.

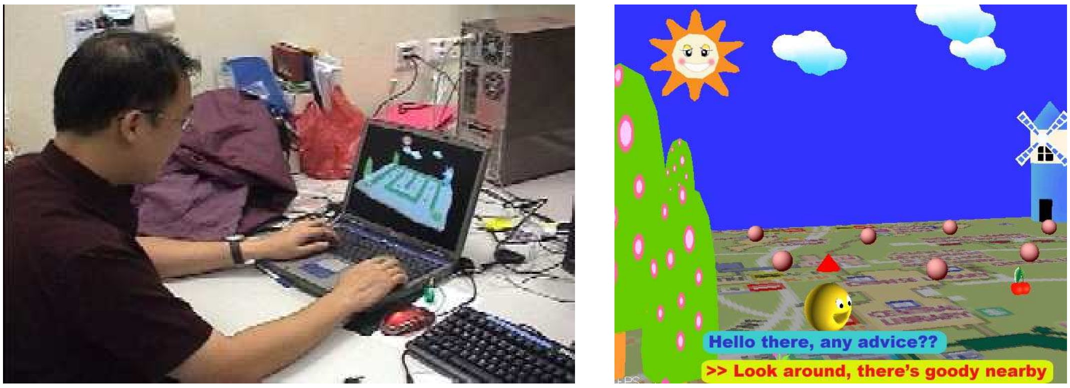


Figure 3.6: From Left: A Helper guiding a physical player in the game; Helper's View.

From another perspective, with the assistance of Helper, a Pacman/Ghost will be having a guardian watching over her, even though the identities or whereabouts of the Helpers might be unknown. It would be as if they possess an extra pair of eyes roaming in the sky (like a bird) aiding and advising them.

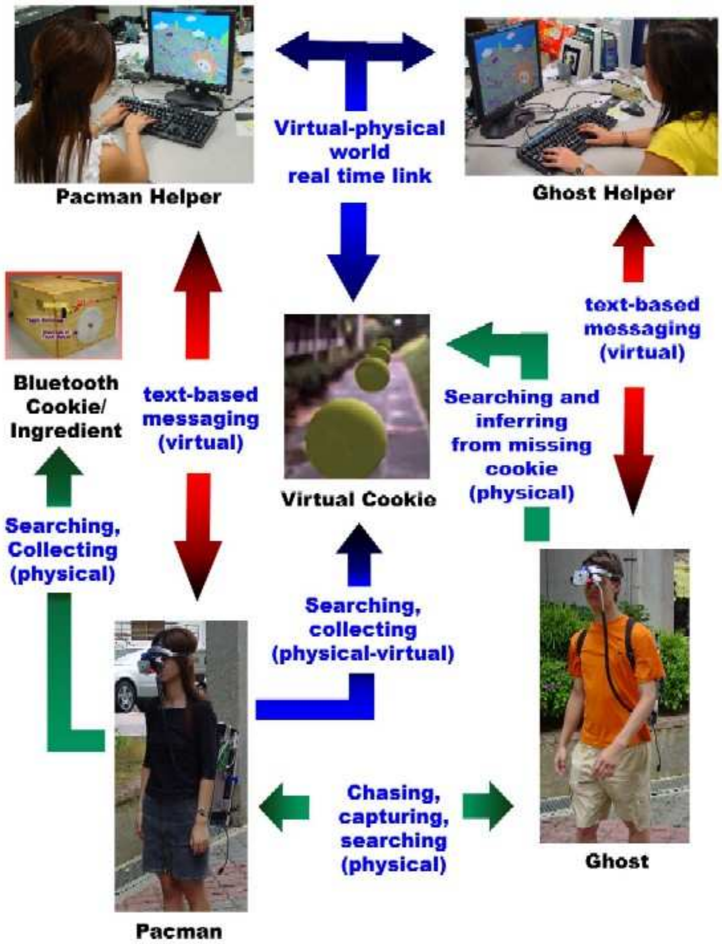


Figure 3.7: Overview of the Human Pacman Game

To a great extent, the existence of Helpers for both teams resulted in a powerful synergy of interaction between real and virtual domains in the social context of mixed reality gaming. Collaboration within each team becomes more effective despite being bridged by only text messages. Winning or losing largely becomes a function of teamwork, and this adds a greater thrill and fun factor to the game as a whole. In short, the introduction of the role of Helper enriches the interaction theme of Human Pacman. Figure 3.7 shows the overview of Human Pacman.

3.3 System Design

Figure 3.8: An Overview of the System Design of Human Pacman.

Human Pacman features a centralized client-server architecture that is made up of four main entities, namely, a central server, client wearable computers, Bluetooth-embedded objects and the Helper system. An overview of the system is shown in Figure 3.8. Communication between clients and server is maintained using either WiFi or wired LAN.

The game can be played either indoor or outdoor. The main difference, in terms of technology, between the indoor and outdoor game is the position sensing technology used. This is discussed below.

- Outdoor Gaming

  Figure 3.9 shows the real person playing Pacman, the first person view and the virtual representation. In an outdoor environment, the system uses Real-

Figure 3.9: Outdoor Human Pacman

Time Kinematic Global Positioning System (GPS) technology to track the Pacman and Ghost players' physical location. This state-of-the-art technology offers accuracy of up to centimeters in real time; which is not possible using conventional off the shelf GPS. The position of each player in the game is updated at 10 Hz and will be transmitted to the server during the game using wireless LAN. The server will then disseminate this information to helper systems so that the position of Pacman and Ghost is correctly reflected in the VR environment on the helper PC.

- Indoor Gaming

For indoor gaming, a miniature maze was custom made for the game to be

Figure 3.10: Indoor Human Pacman

played. Figure 3.10 shows the physical maze, its virtual representation, and the maze the maze layout. The system uses a mixture of infrared-red (IR) sensors and camera tracking to locate the players' position in the maze robustly and accurately. IR receivers are fitted on the floor boards of the maze at regular intervals. As each wearable computer carries an IR emitter of unique frequency, the system will be able to sense the position of Pacman and Ghost in the maze. There are also four dragonfly cameras strategically placed over the maze to visually track the players inside the maze. As both tracking methods are independent of each other, there will be two sets of location readings obtained. These readings will be averaged at the game server and disseminated to the helpers just as in the outdoor system.

The original intention for Human Pacman is to play the game in a wide area, outdoor environment so that players enjoy more mobility. However, under the following situations, the indoor version can be used instead:

- When there are indoor exhibitions.
- When the GPS receivers on the wearable computers cannot locate enough satellites.
- When under time constrain, the actual game area cannot be surveyed and the virtual world is not created.

The hardware and software used in Human Pacman will be discussed in detail in the subsequent parts of this chapter.

3.4 Hardware

The hardware of Human Pacman has gone through several revisions since the project started in 2002. Aided by the vast improvement in technology, Human Pacman has grown from a mere proof of concept to a full-fledged multi-player outdoor game. This section describes the hardware used in the current Human Pacman System in detail.

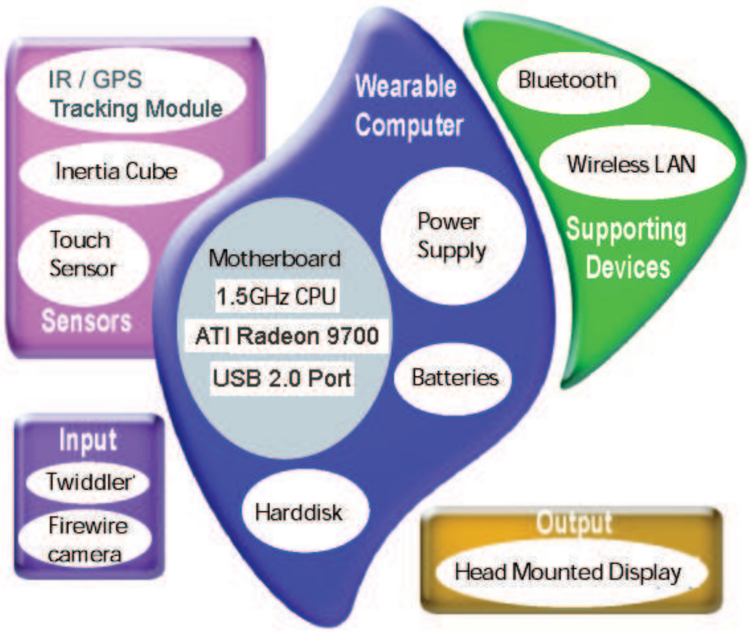


Figure 3.11: An Overview of the Wearable Computer System

3.4.1 Wearable Computer

Figure 3.11 shows the overview of the wearable computer system for players playing Pacman or Ghost. A description of each component is as follows.

- Acer TravelMate 4001LMi

  At the heart of the whole system is an Acer laptop with an Intel Pentium 4 1.5GHz processor and an ATI Radeon 9700 graphics card. The processor for the wearable computer does not need to be extremely power to run the software for Human Pacman, however, a separate graphics card is highly recommended for the frame rate and the augmented display to be acceptable (30 frames per second). Other points taken into consideration while making the choice for laptop includes battery-life, Integrated WiFi, number of USB2.0

ports, weight and price.

- LCD Virtual Display GVD300

  LCD Virtual Display GVD300 is a mini LCD Head Mounted Display (HMD). The unique optical system is adopted to dummy the mini-display into 36 inch large displays 2 meters away. Operating at 9V, it consumes about 1.0 watts of power from a small lightweight battery when fitted to the wearable computer. The battery is also very durable and is able to last more than 2 hours.

- Logitech QuickCam Pro 4000

  Logitech QuickCam Pro 4000 is a USB 2.0 that supports a 640 x 480 VGA resolution with a frame rate of up to 30Hz. The high frame rate and good resolution makes it suited for use in the wearable computer system as a video input of the surroundings for display in the HMD. The camera is mounted in front of the HMD. The camera is powered from the USB2.0 port on the PC and hence no external source is required.

- InertiaCube^3

  InertiaCube^3 is a 3-DOF (Degree of Freedom) orientation tracking system. It is used to track the user's head orientation (yaw, pitch, roll) with a dynamic accuracy of 3° RMS. With a low latency of 2ms and high update rate of 180Hz, it is possible to have a real-time tracking of the user's head motion. This is important because the virtual cookies are kept in absolute real space positions in the user's AR view as she moves her head. A high latency and low update rate will affect the realism of the experience of having virtual cookies in real space.InertiaCube^3 is also lightweight (17g) and operates at 6VDC, 40mA. It can be powered directly via the USB port provided by the manufacturer.

- Bluetooth USB Dongle

  Bluetooth communication is made with TDK's Bluetooth USB Adaptor. The adapter is a Bluetooth 1.1 compliant device with a communication range of 10m. This makes it ideal for Human Pacman as its short range makes it essential for the user to be physically near the Bluetooth embedded device before she is being alerted to the latter. Furthermore, the device does not pose too much burden on the wearable computer's battery as it has a low power consumption of only 0.175 watts under normal operation.

- Touch Sensor

  Touch-sensor is placed on the backpack of a player and is triggered when touched by an enemy. The touch sensor uses a capacitive touch sensor chip, QT110 and the E11x demo board. Figure 3.12 shows the circuit diagram of the touch sensor. The circuitry operates on two AAA size batteries and uses a LED and piezo beeper as indicators when a touch is sense. When switched on, the touch sensor calibrates itself to the current state of capacitance it is in. Any contact made to the proximity plate subsequently will cause a change

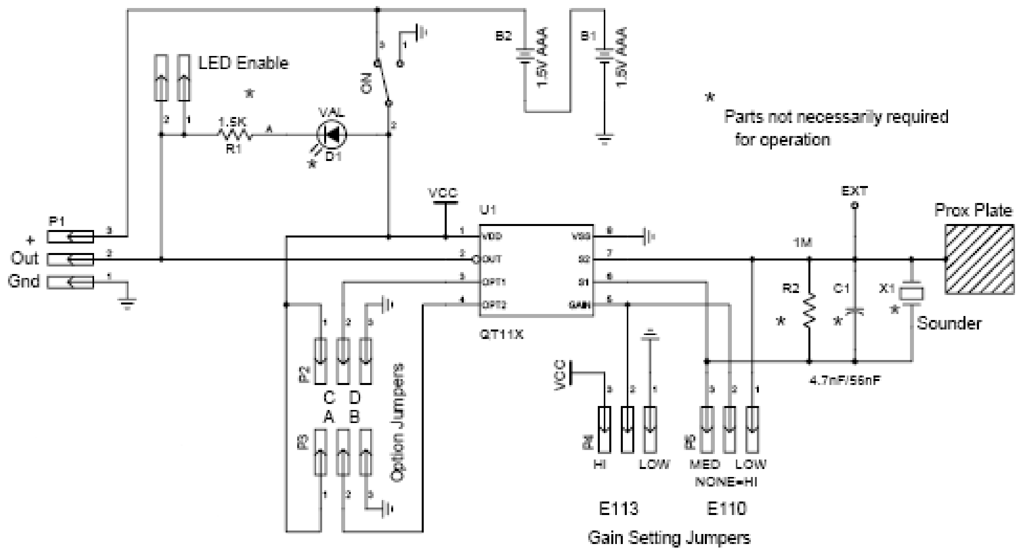


Figure 3.12: Touch Sensor Circuit Diagram

Figure 3.13: Touch Sensor and USB Mouse PCB in one PCB Housing

Indicator	**Function**
Bluetooth	Shows whether there is a bluetooth connection with another device, eg computer. Not applicable if device is connected using wired means.
Radio Indicator	Shows whether there is a radio data link with reference station (see section 3.5.3).
RTK Indicator	Shows whether the z-max is operating as a RTK GPS receiver or normal GPS receiver.
Satellite Indicator	Shows the number of satellites in view.

Table 3.1: Z-Max indicators' descriptions.

in capacitance and the output to be triggered.

The touch sensor needs to communicate with the wearable computer and informs the wearable computer whenever a touch is sensed. This can be done via serial port or USB port using a microcontroller. However, to simplify the implementation, our system uses the microcontroller from a readily available USB mouse to achieve the same result. The switch on middle button of the mouse's PCB is removed and the touch sensor's output (normally 3V) is soldered in its place. The middle button was chosen as it has not been utilized in our program earlier. When a touch is sensed, the touch sensor's output becomes 0V and the wearable computer will get a middle click response from the mouse. This will trigger the other chains of event on wearable computer (for example, sending a message to the game server to indicate that Pacman has lost the game).

- Z-Max^{tm} Surveying System *(for outdoor only)*

  The Z-Max surveying system from Thales Navigation [23] (see Figure A.6

Figure 3.14: Z-Max^{tm} Surveying System

is a precision GPS surveying solution that offers centimeter-accurate Real Time Kinematics (RTK) performance. This is achieved using Automatic Decorrelation and Parameter Tuning (ADAPT)-RTK technology which allows rapid adapting to current conditions. The positions (in longitude and latitude) obtained from Z-Max are real time updates. Utilizing the software toolkit provided by Thales Navigation, the Z-Max can be pre-programmed to stream this information to the wearable computer via serial port connection. Hence, the wearable computer only needs reading off this position data from the relevant port. The few indicators shown in Figure 3.14 allows easy trouble-shooting. Their functions are listed in Table 3.1.

- IR emitter *(for indoor only)*

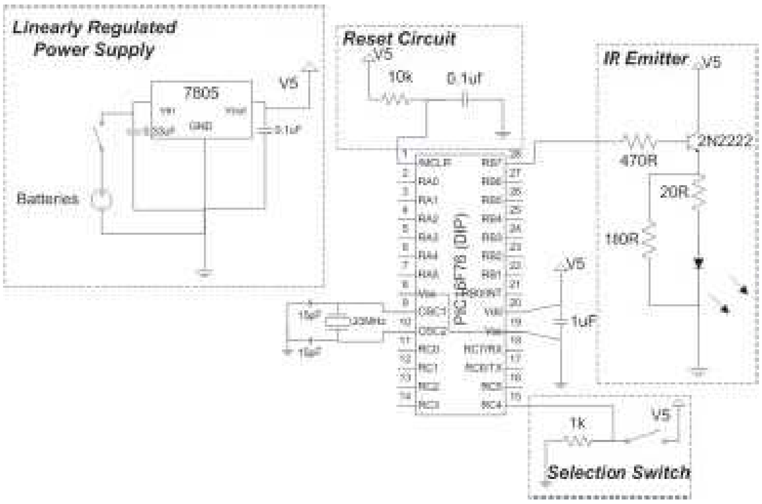


Figure 3.15: The circuit diagram of the infrared-red emitter.

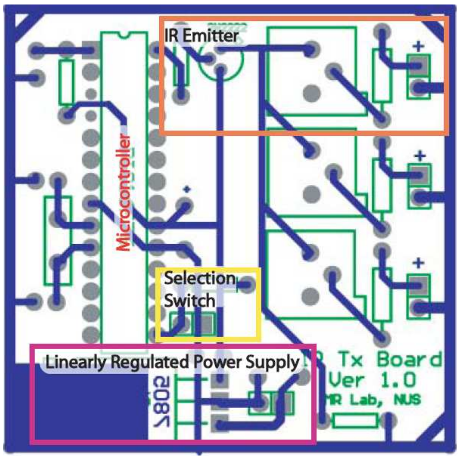


Figure 3.16: PCB layout of IR Emitter with main modules labelled.

The emitter circuit is shown in Figure 3.15. The PCB design and actual circuit board made are shown in Figure 3.16 and 3.17 respectively. The main modules in this circuit are highlighted on these boards, namely the linearly regulated power supply, microcontroller, IR emitter module, and the selector switch module. A 7805 linear voltage regulator is used for regulating the power supply at 5V. In the IR emitter module, the IR emitter used is the TSUS5400 [24]. TSUS5400 is a low cost emitter with peak wavelength

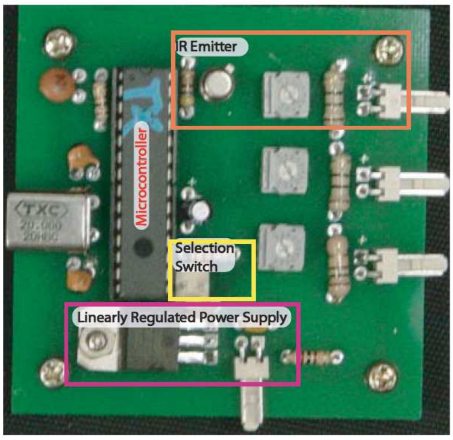


Figure 3.17: Actual circuit board of IR Emitter with main modules labelled.

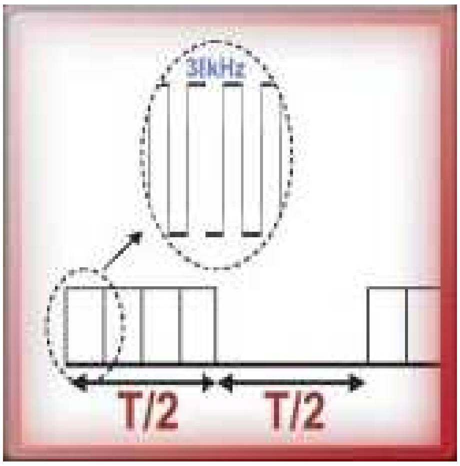


Figure 3.18: Signal generated at the IR emitter.

of 950nm and angle of half intensity at ±22°. It is connected to the microcontroller PIC16F76 [25] through pin RB7. A square wave of frequency $\frac{1}{T}$, modulated with a 38kHz square wave, is generated by the controller at the IR emitter (see Figure 3.18). This is later demodulated at the receiver end. The frequency $\frac{1}{T} = 700Hz$ is used to denote Pacman and $\frac{1}{T} = 1000Hz$ for Ghost. The 2N2222 NPN transistor acts as a current buffer for the emitter. A 100Ω resistor is connected in parallel to the emitter to tie the voltage across it to zero during the ‘off’ phase of the transistor. A selector switch is added for selection of either the Pacman or Ghost frequency.

The New and Old Wearable Computers

Figure 3.19: *(top)*Version 1: 2002. *(bottom)*Version 2: 2003

To highlight the difference between the earlier versions of Human Pacman and the current version, Figure 3.19 shows the earlier versions of the wearable computer in Human Pacman. The current version equipped with the RTK GPS, input and output devices can be seen on Figure 3.20.

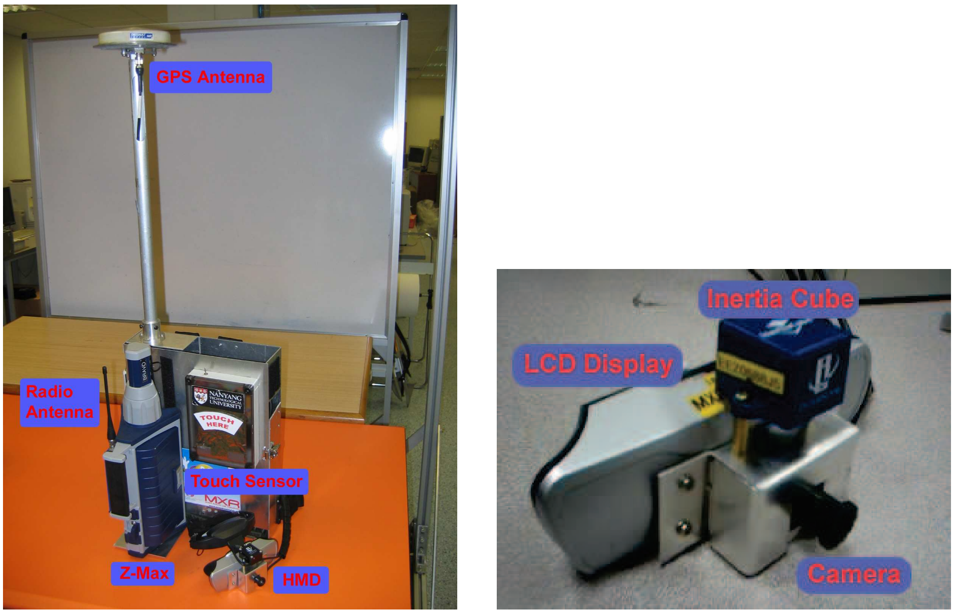


Figure 3.20: Version 3: 2005. Fully Integrated Wearable Computer

3.4.2 Bluetooth Embedded Object

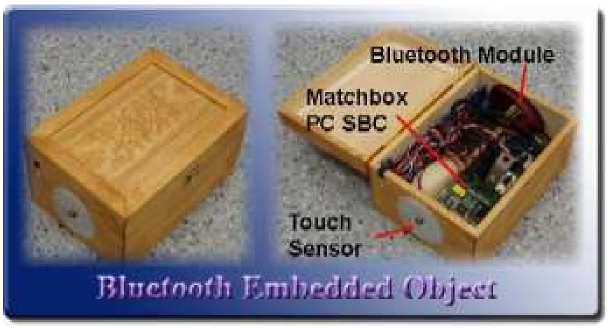


Figure 3.21: Bluetooth Embedded Box

Figure 3.21 shows the bluetooth embedded box in the game which acts as the special cookie that can give Pacman a limit period of Ghost-capturing abilities. The bluetooth embedded box is custom-built in the laboratory for the game. The system's makeup is relative simple, namely, a single board computer (SBC), a serial Bluetooth dongle, touch sensor circuitry, controller module, and power supply.

- Tiqit System

  The single board computer (SBC) used in Tiqit System is a Matchbox PC (MPC). The MPC has a 66MHz 486-SX based CPU with 16MB SDRAM. It has two RS232 serial ports and has an IDE controller controlling a fixed 16MB Flash and a 1GB Microdrive. It requires 3 to 7.5 watts of power. The Bluetooth module used in tandem with the MPC is the BL510 [26] from Brain Boxes. It is a RS232 Class 1 Bluetooth module compliant with Bluetooth V1.1. The module requires a 5V DC supply.

- Touch Sensor

  The touch sensor implemented in the Bluetooth embedded box is similar to that in the wearable computer discussed earlier. It sense touch by detecting the difference in capacitance at its electrodes. Figure 3.21 shows the electrodes, which are actually aluminium disc, located at the side of the wooden box.

- Power Supply and Controller Unit

  The Bluetooth embedded object runs on a 12V rechargeable lead acid battery. Non-regulated power is supplied to the SBC. A 5V regulated supply from LM7805 is connected to the Bluetooth module, touch sensor, and controller circuit. This circuit comprises of a few main modules, which includes the microcontroller (the PIC16F76 chip), touch sensor module, LED indicator module, toggle switch module and RS232 interface module.

Module	**Function**
Controller	Serves to link the touch sensor output to the MPC and to control the other external interfaces such as LED indicators.
Touch sensor module	Operates in a similar manner to the one described for the wearable computer, informs the microcontroller of any "touch" event it senses on its electrode.
LED display module	Provides a visual information of the touch sensor status and SBC boot-up status.
Toggle switch module	Allows the user to send commands to the SBC, such as to shutdown the operating system, by toggling it.
RS232 interface module	Acts as an interfacing bridge to facilitate communication between the microcontroller and the serial port of the SBC. Information is relayed between the PIC16F76 and the SBC through UART communication configured as a 9600bps 8-bit communication, with 1 stop bit and no parity bit.

Table 3.2: Functions of each module on Controller Circuit.

As the Bluetooth embedded object has to be a solitary device with no external display or keyboard inputs, visual indications and external inputting methods has to be implemented for the user to easily operate the device. The function of each module is described in Table 3.2 below.

3.4.3 Central Server and Helper Systems

The system requirements of both the server and Helper systems are less stringent as compared to the wearable computer. Both systems should have a 100Mbps LAN adapter and be able to run Microsoft Windows XP. In addition, the Helper systems should have a graphics card which supports OpenGL 1.1. The server and helper program can run concurrently on one

3.5 Software

The program in the SBC of the Bluetooth embedded object and the ceiling tracking system are coded in GNU C. All other programs are coded in Microsoft Visual C++ 6.0. OpenGL 1.1 is used for rendering of graphics in the Pacman, Ghost and Helper systems. MXRToolkit, an open source development kit developed in our laboratory, is used in the Pacman and Ghost systems for AR calculations.

A summary of the overall system data flow between the wearable computer, server, Helper system and the Bluetooth embedded object as shown in Figure 3.22.

- Server System

  At the heart of the Human Pacman system is the server which acts to relay and broadcast information between the systems of Pacmen, Ghosts and Helpers. The server sends the following data to Pacman/Ghost wearable computer system.

  - The game status such as whether Pacman or Ghost has won.
  - The client ID of user's partner if she has been assigned one.
  - Initial position of all virtual cookies in the game area.
  - Positions of all the virtual cookies that has already been collected.

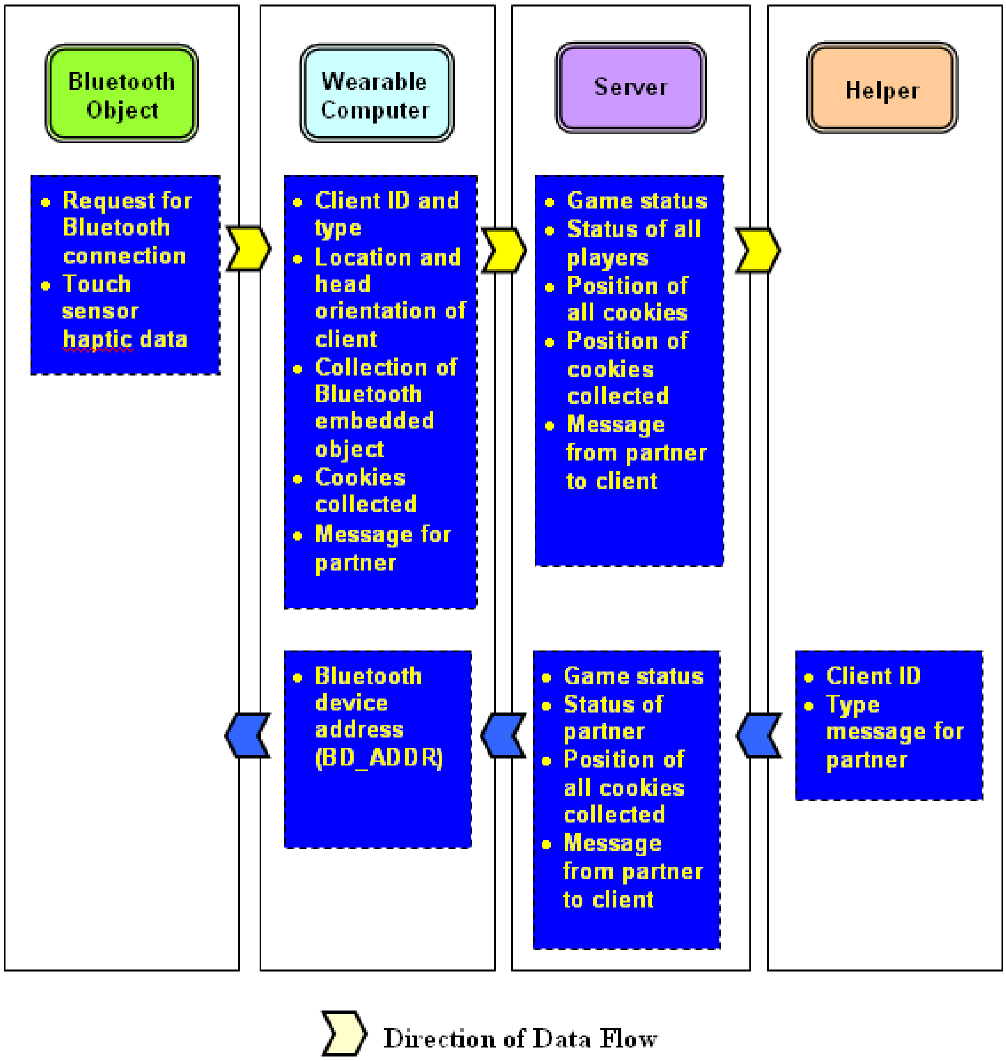


Figure 3.22: Overall System Flow

  - Any message that has been sent by the user's partner.

- Helper System

  The Helper system receives the same kind of data from the server as the wearable computer. In addition she also receives status of all Pacman and Ghost in the game. This includes the position and orientation of these players, the event of a Pacman turning into a Super-Pacman, and the devouring of any Pacman/Ghost. She also receives the position of the Bluetooth embedded object in the game area. In turn the Helper system sends to the server her client ID and type (i.e. Helper), and any message for her partner player.

- Wearable Computer System

  Wearable computers system has data exchange with the server and Bluetooth embedded object. The wearable computer system sends the following data to the server.

  - The client ID and type (i.e. Pacman or Ghost) of the player.
  - The player's location and her head orientation data.
  - If it is a Pacman system, whether the player has become a Super-Pacman.
  - If it is a Pacman system, the virtual cookies collected by the Pacman.
  - Any message for player's partner Helper.

- Bluetooth Embedded Object

> The Bluetooth embedded object sends the request for Bluetooth connection to the Pacman wearable computer system. In turn it receives the unique Bluetooth device address (BD ADDR) for it to ascertain that the Bluetooth server is a Pacman wearable computer system. The Bluetooth embedded object also sends any touch sensor haptic data, indicating that the touch sensor has been activated, to the Pacman system, if one is connected.
>
> The subsequent part of this chapter will cover program of each module that makes up the entire Human Pacman System.

The subsequent parts will discuss in more detail about the techniques use in each module of Human Pacman.

3.5.1 Server System

The central server program is a Win32 console-based program that could be executed in standard desktop running the Windows operating system. The physical location and players' status updates are sent from the client wearable computers to the server every 10-21 ms, depending on the processing load on the client. The server maintains up-to-date player information (location, status, etc.), and presides over any communication between the Bluetooth objects and the wearable computers. Figure 3.23 shows the server flow chart.

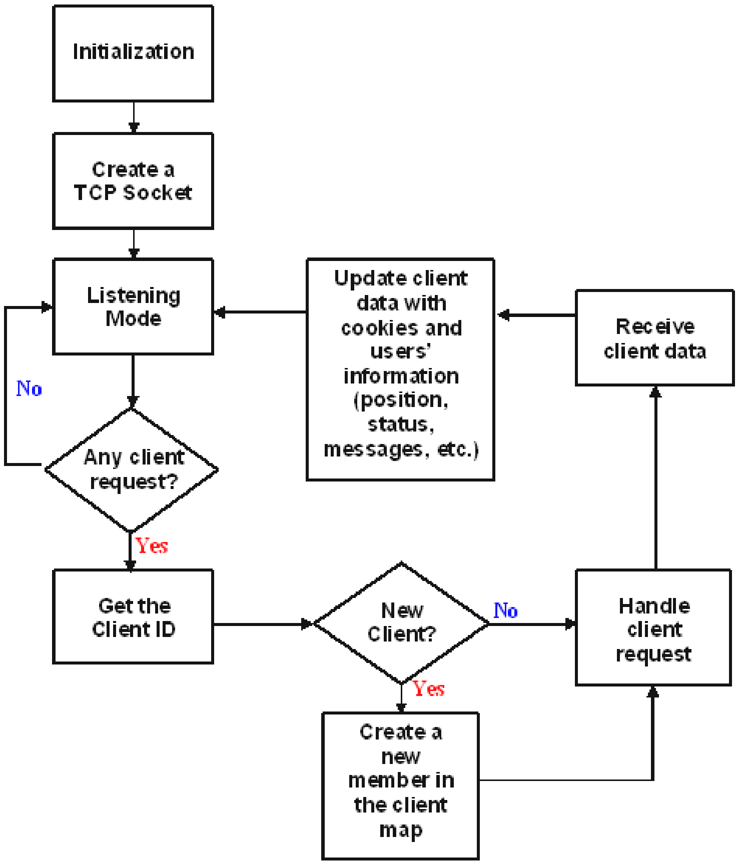


Figure 3.23: Server System Flow Chart

3.5.2 Indoor Tracking System

For indoor gaming, a maze (2.4 m x 2.4 m) was custom made to allow one Pacman and Ghost to play the game. To make the maze portable for demonstration at various places, it was designed like a jigsaw puzzle with 4 x 4 matrix block, each measuring 0.6 m x 0.6 m. Indoor tracking is achieved by two independent means, namely, infrared red tracking and visual tracking. As both tracking methods are independent of each other, there will be two sets of location readings obtained. These readings will be averaged by the game server and disseminated to the helpers.

If one of the tracking means fails, the system is still able to make use of the other available mean. Hence, the players' positions in the maze can be tracked robustly and accurately.

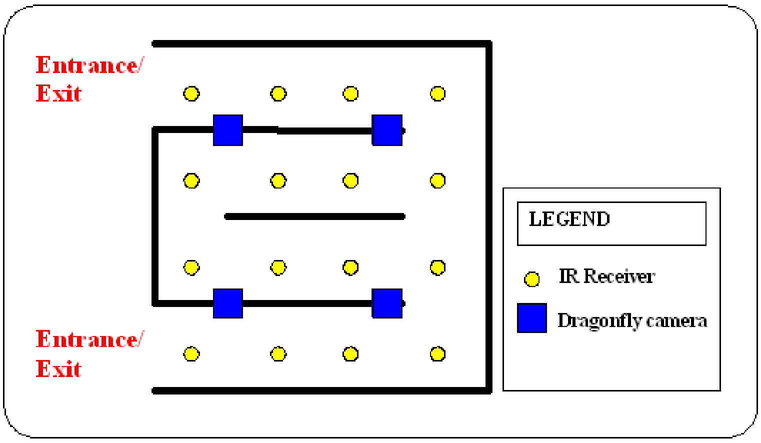


Figure 3.24: Layout of the maze for indoor Human Pacman.

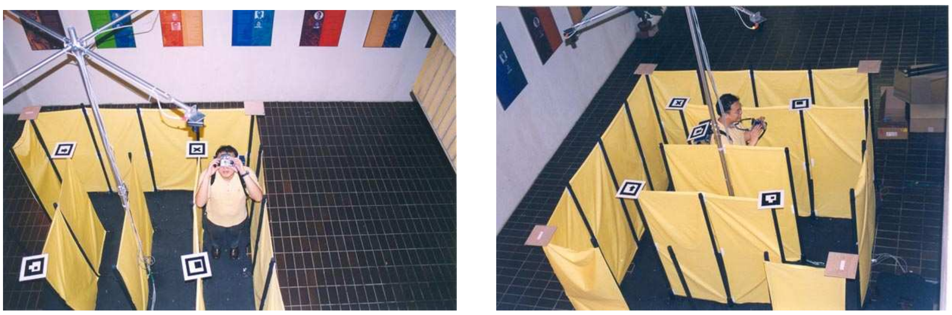


Figure 3.25: Ghost Player in Maze

Figure 3.24 shows the 2D layout of the maze and the cameras for the indoor tracking hardware. The actual view of a Ghost player inside the maze can be seen in Figure 3.25. The cameras are mounted over the maze to capture the top-down view of the activities inside the maze. The two means of indoor tracking will be discussed next.

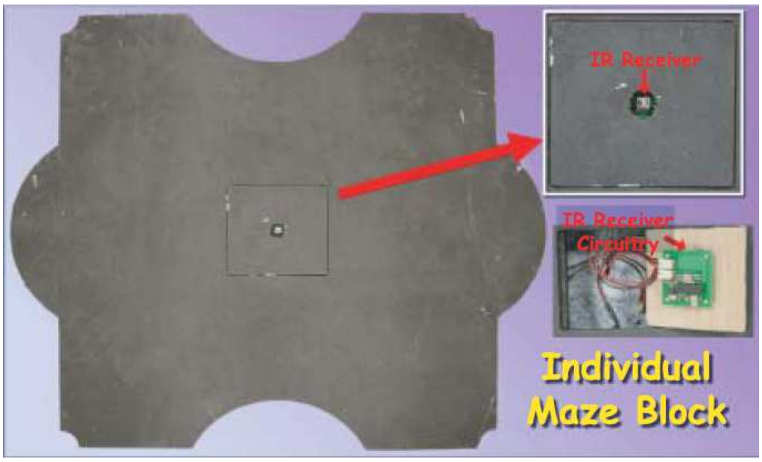


Figure 3.26: A single maze block with infrared-red receiver.

Infrared Red Sensing

On each maze block, there is an infrared receiver embedded inside for position sensing purpose as show on Figure 3.26. These receivers will receive different infrared red signals emitted by the wearable computers whenever Pacman or Ghost stands over the block. If IR signal from Pacman's wearable computer is detected over a block, the system will be able to determine the position of Pacman in the maze. The similar case applies for the Ghost player.

To simplify the issue of data transferring and power provision, the IR receiver in each block is designed to be connected to one another sequentially when the maze is being assembled. Figure 3.27 shows how two neighboring maze blocks are joined together. Hence, after the maze is assembled, there will only be one connection running through the maze to the main game server. This connection serves to provide power to all the receivers in the maze and transfer the detected positions of Pacman and Ghost.

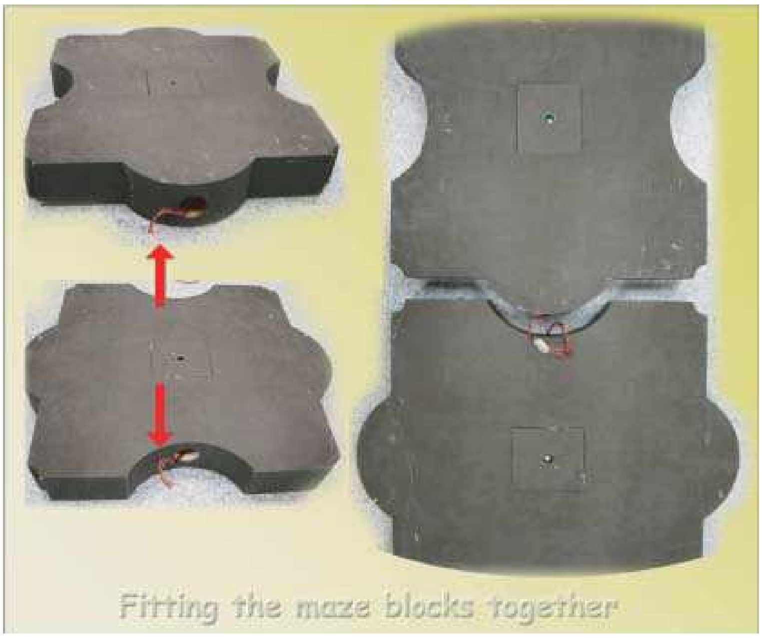


Figure 3.27: Connecting neighboring maze blocks.

Camera Tracking

The Camera Tracker runs on Redhat Linux 7.3 and uses four high-resolution dragonfly cameras to track the players' position in the custom made maze. The system requirements for the camera tracking system is not stringent but two firewire PCI cards are needed to support all four cameras as one card can only support at maximum three cameras. The visual tracking system provided by the cameras complements with the IR tracking provided by the maze blocks discussed earlier, to provide more robust and accurate tracking.

The Camera Tracker uses libraries in ARToolKit 3.27 for tracking the patterns on the top of the player's backpack. ARToolKit is a software toolkit that is popularly used for building AR applications and its video tracking libraries allows calculation of real camera position and orientation relative to physical markers in

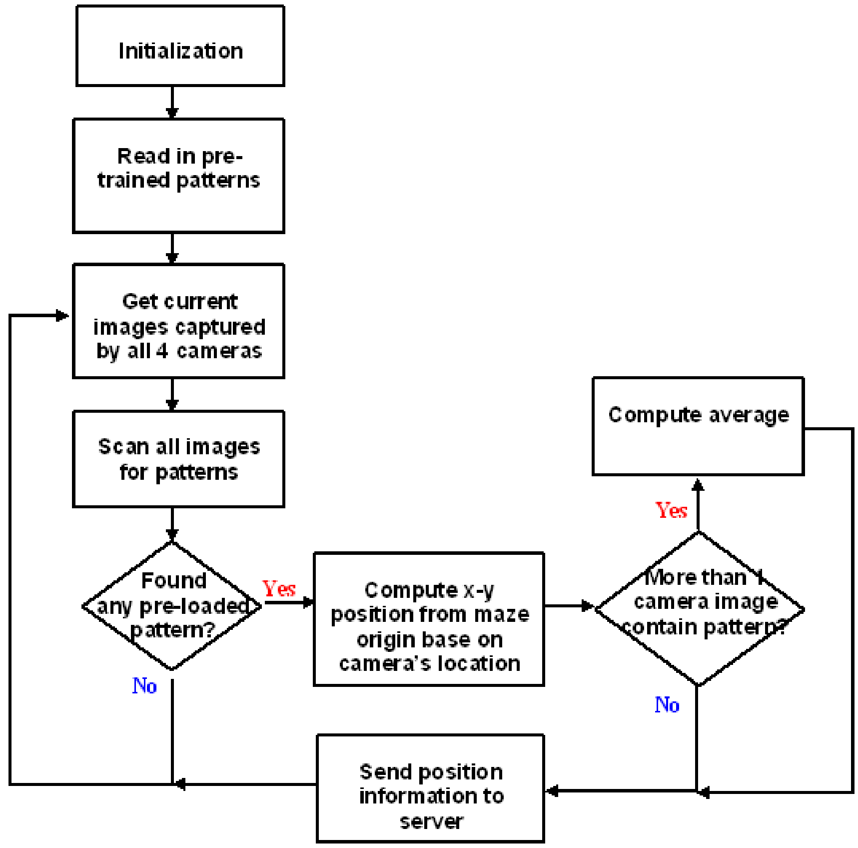


Figure 3.28: The Flow Chart of Algorithm in the Camera Tracking System

real time. Some of the features of ARToolKit include:

- single camera position/orientation tracking.
- tracking code that uses simple black squares.
- the ability to use any square marker patterns.
- easy camera calibration code.
- fast enough for real time AR applications.

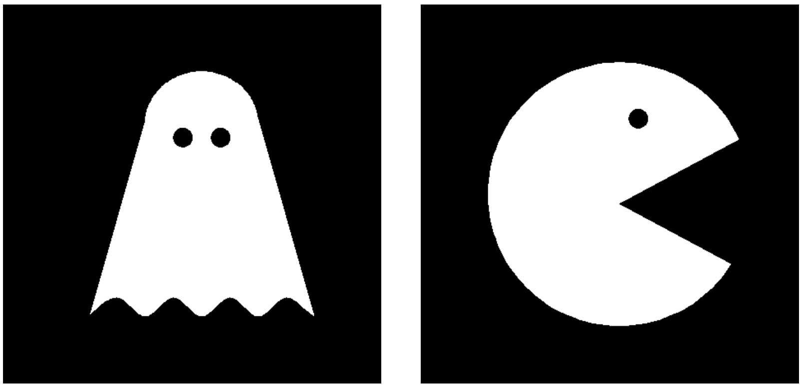


Figure 3.29: Ghost and Pacman Pattern used in Camera Tracking System.

The flow chart showing the algorithm in the camera tracking system is show in Figure 3.28. The program starts by first initializing the four dragonfly cameras and the networking with the game server. Then, it will load the pre-trained patterns for Pacman and Ghost (Figure 3.29 using AR Toolkit library. The patterns for Pacman and Ghost need to be uniquely designed so that the system is able to differentiate the players' identities. Figure 3.29 show the patterns that were used to identify Ghost and Pacman players respectively. After these sequences are completed, the program will run in a continuous cycle.

Within each cycle, the images captured at almost the same instance of time by the cameras will be obtained and the program will scan these images for the Pacman and Ghost patterns. If any is found, the program can compute the translation and orientation of Pacman or Ghost relative to the cameras utilizing the libraries in ARToolkit. Four other different patterns are placed at known positions inside the maze acting as markers. Using matrix multiplication , we can then compute the position of Pacman or Ghost relative to the marker as seen in Figure 3.30. Since

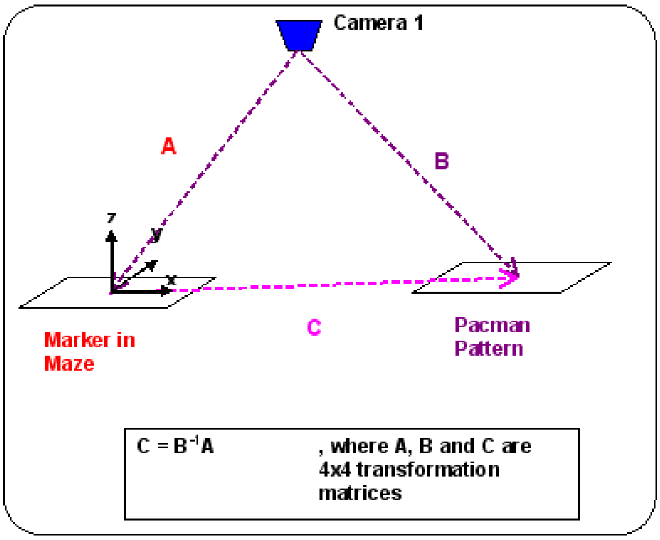


Figure 3.30: The computation of Pacman relative position to a particular Marker.

we have prior knowledge of the positions of these markers, we can finally obtain the x-y position of Pacman or Ghost with respect to the origin of the maze (see Figure 3.31).

3.5.3 Outdoor Tracking System

In addition to the Z-Max surveying system (known as the rover unit) mounted on the wearable computer described earlier, RTK positioning requires another static GPS receiver to serve as a reference or base station at a known location. The Scorpio 6502SK, pre-configured by Thales Navigation, functions as such a station. Figure 3.32 shows the Scorpio 6502SK, used in Human Pacman. Both GPS receivers (Z-Max and Scorpio) make observations of the GPS signals at the same time and a radio data link between the two receivers permits data to be sent from reference to rover, where the calculation of coordinates is carried out by the pre

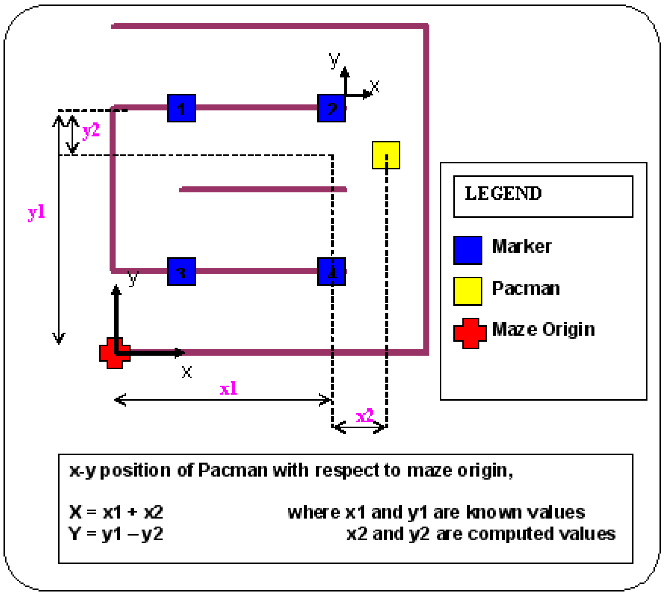


Figure 3.31: The computation of Pacman relative position to the maze origin.

programmed Z-Max surveying system. To achieve centimeter tracking accuracy, the base station and Z-Max needs more than 5 satellites are in view. Hence, ideally, the base station should be place on open elevated land. Further details can be found in [27].

3.5.4 Helper System

Multiple threads run in the Helper system. The main thread initializes the system, renders the VR world using VRML, and updates the position and orientation of the VRML models of the Pacman, Ghost and cookies. A network thread exchanges data with the server. Data obtained from the server includes positions of the mobile players and cookies, and data sent includes messages sent to the Helper's partner. The software flow of the Helper system is given in Figure 3.33.

Figure 3.32: Base Unit for RTK GPS.

3.5.5 Wearable Computer System

Similar to the Helper system, the Pacman and Ghost system runs multiple threads. The most important being the main thread and the network thread. In the Pacman system, the purpose of the main thread includes initializing of the hardware modules, updating of the visual display, collection and processing of sensors data, deciding if a cookie has been collected, and updating of shared data between the threads. In addition, events from the keyboard (Twiddler2) inputs and from the Bluetooth are also processed and the corresponding shared data updated. The network thread exchanges this shared data with data from the remote server. Figure 3.34 gives the flow of the main and network threads of the Pacman system. The program starts by initializing parameters and starting a new network thread and Bluetooth thread. Data flow between the threads is done through setting a shared

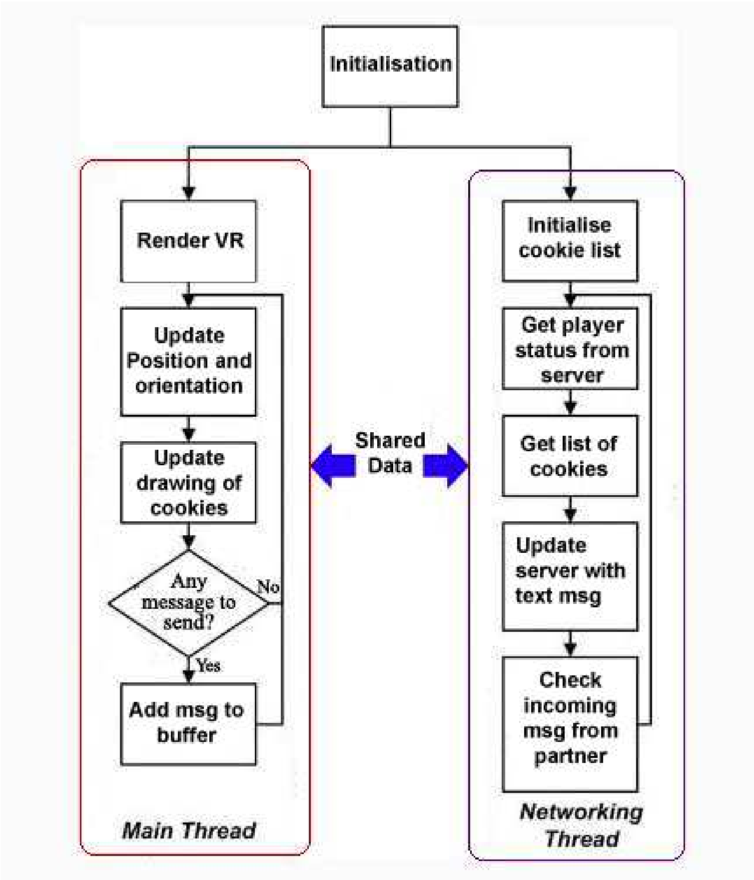


Figure 3.33: The Flow Chart of Algorithm in the Helper System

semaphore. The Ghost system is similar to the Pacman system, except that it does not have cookies collection and Bluetooth events.

3.6 User Study

It is important to analyze and study the actual system interactions with the Helper, because in social computing it has been shown that the study of the context in which interaction with computation occurs is most important. To gain useful feedback from the end user, we conducted an experimental user study survey on the Human

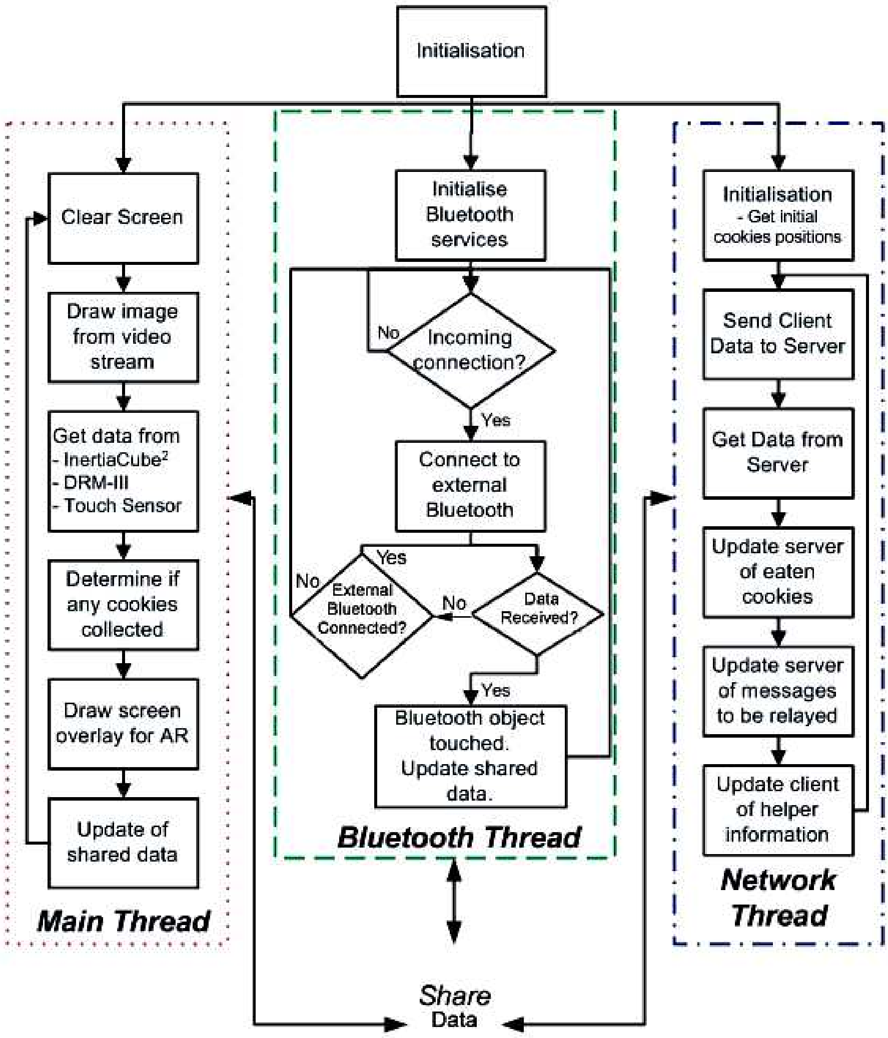


Figure 3.34: The Flow Chart of Algorithm in the Wearable Computer System

Pacman system. Our aim is to find out from actual users their experience of the positive and negative aspects and interaction in using the Human Pacman system.

In these tests, the focus had been placed on the physical interactions between physical players and environment as well as virtual interactions of Pacman and Ghost players with the Helpers presented by the game. Our study involved 51 subjects between the age of 21 and 40, of which twelve were females and 39 were males. The experiment setup consisted of three parts. Each session would require 3-4 participants. First, a 3 minutes Human Pacman video was shown to give the participants a better understanding of the game. This was followed by a trial where the subjects tried the roles of Pacman, Ghost, Pacman's Helper and Ghost's Helper, along with other subjects taking different roles. The text messages within each team were logged during the game for later analysis. Finally the subjects had to fill up a questionnaire and to provide comments on the system. The questions listed in the Table 3.3.

Questions and aims: Table 3.3 shows the list of questions that were asked in the survey. Following each question are the reasons for asking this question in the user study.

3.6.1 Discussions

All of the figures derived from the user study are plotted as bar charts as shown in Figure 3.35 for better analysis. Insights provided by users' comments are also

Questions	**R**eason for asking
i. How intuitive do you think it is to collect cookies by physically walking through them?	In everyday life, collection of an item is seldom, if ever, made by walking through it. We seek to understand if the user finds it intuitive to collect virtual cookies by walking through them, just as it is done in the original Pacman game.
ii. Does the physical collection of a real object (special cookies) enhance the gaming experiences?	The collection of special cookies is a tangible interaction with a physical object that translates into a digital meaning (i.e., update of Pacman's inventory list). We want to find out if such graspable interaction enhances the game for the user.
iii. What do you think of the "capturing" event implemented in our system (touching the Pacman by the Ghost)?	The "capturing" event is a reflection of the naturalistic and physical approach Human Pacman took towards tangible interaction. We seek to find out if user enjoyed this feature.
iv. Do you think the Helper increases the chances of winning?	The Helper is an additional feature to the original game. It is introduced to breach the geographical barrier in mixed reality gaming. However, we do not want to add this feature just for this sole purpose. We would like to find out if this enhancement to the game is useful.
v. Does Helper make the game more enjoyable for Pacman/Ghost?	The Helper is supposed to be an enhancement to the original game. Thus, if this new role does not make the game more enjoyable to the original players, there may be a need to reconsider the inclusion of the Helper.
vi. How boring/ exciting is the game for Helpers? Please rank from the 1 (Very boring) to 7 (Very Exciting.	As the role of the Helper is played by another person, we should find out from the point of view of the Helper whether he or she is enjoying the game as well.
vii. Currently, the game is played by one Pacman/Ghost to a Helper. Do you think the game will be more interesting for the Helper if he/she can assist more numbers of Pacman/Ghost? If yes, what is the optimal number? If not, why?	Human Pacman basically revolves around social and physical interaction among all players. Generally, these interactions will increase with more players and thus making the game more enjoyable. However, as number of players increased, confusion will inevitably start to creep in. We would like to find out what is generally the optimal number of players
viii. Do you think the frequency of conversation will increase when the complexity of the game (in terms of area size and number of Pacman/Ghost) is increased?	As Human Pacman is based on virtual and real interactions, we would like to know if complexity of the maze will affect the frequency of communication.
ix. Do you think Pacman needs a Helper?	The main role of Pacman helper is to guide Pacman to all the cookies and avoid the path of the Ghost. However, some players may find it more exciting to be able to explore the real world freely and without any reins.
x. Do you think Ghost needs a Helper?	The main role of Ghost is to track down Pacman. Similar to the above situation, some players may feel that the additional help is redundant and would like to explore the real world freely.

Table 3.3: Questions in the user study

used in the analysis.

Question (i), (ii) and (iii) examine the perceptions of the physical players (Ghosts and Pacmen) toward the physical interactions involved in the gameplay. A number of users commented that they liked the idea of "physical involvement" and "physical movement" in Human Pacman. Some said that such movement is a good form of exercise. From question (i), collection of virtual cookies by walking through them is deemed to be intuitive by 87% of the users (as seen in Figure 3.35 (i). The rest found the experience to be acceptable, though not intuitive. Evidently, as no one rejected it as a bad idea, it implies that physical-based interactions are well-received by the end users and are worth to be fully expanded and developed.

Moving on to tangible interaction element in Human Pacman, from question (ii), 78.3% (as seen in Figure 3.35 (ii)) found that the graspable interaction offered by the collection of real objects enhances the game. The other 21.7% gave a neutral response towards having this collection as part of the game. Almost all of the users indicated in question (iii) that they liked the "capturing" event. Despite both events being naturalistic interactions with the physical world, users seemed to like the "capturing" event more. This suggests the physical human-to-human interaction in the process of "capturing" makes the event more enjoyable than the picking up of the Bluetooth Special Cookie or Ingredient.

The subsequent three questions (question (iv), (v) and (vi)) focus on the experience of Helper in the game. Nearly 4 in every 5 participants' response to question (iv) agreed on the fact that the Helper increases the chances of winning, while the rest thought maybe, and none disagreed on the role of Helper in winning the game. From question (v), a lesser 64.7% reflected that the role of Helper made the game more enjoyable for Pacman and Ghost and 5.9% felt otherwise. The remaining 29.4% hesitated and thought it probably does so. Question (vi) sought to examine from the perspective of users who played the Helpers on how enjoyable they felt toward the role. Using a scale of 1 (very boring) to 7 (very exciting), a mean rating of 4.96 with variance 1.93 was obtained. Evidently, these illustrate the fact that while Helpers are definitely beneficial to the game, the issue of fun and thrill has not been addressed well, since the user (Helper role) did not enjoy it as much as they understood its strategic importance. This may be due to the fact that Helpers interactions are limited only to text-based messaging. To improve on this, perhaps Helper could be assigned to more interesting tasks. For example, Helpers could be granted abilities to temporary make the virtual cookies invisible to the enemies or to make their assisted player immune from being captured, as was suggested by some users.

Question (vii) explores the other possibilities and options to be expanded on the interactions between the real physical and virtual online players. More than two-third of the users revealed that the game will be more interesting if a Helper can assist more than one Pacman or Ghost at the same time, so that they can collaborate and form strategies to win the game. According to the feedback, the average optimal number of Pacman or Ghost per Helper is three, as most users commented that it would be too confusing if there are too many players involved.

Regarding the issue of interaction in terms of textual conversation versus complexity of the game, we found out from question (viii) that 70.6% felt that the communication will definitely be more active as the complexity of the game is increased in terms of the area size and the number of players. It can be hypothesized that when more physical players are introduced, the Helpers have to put in more effort to coordinate the players who are roaming in real world. The physical players would also need more information too if the game area becomes wider and harder to explore physically.

A correlation test was done on those who answered question (v) and question (vii) to find out the relationship between users who enjoy (and those who not) the game and their attitudes toward the issue of higher communication with increasing the complexity of the game. A high degree of correlation coefficient of 0.876 was derived. This suggests that those who enjoyed the game initially would most likely choose to participate and contribute actively to their teammates when complexity of the game is increased. On the other hand, the remaining who do not find the game interesting in the first place (question (v)) would not feel communication would increase with increasing complexity (question (vii)). In other words, it could be hypothesized from these results that a higher enjoyment can promote a positive increase in communicative interactions.

The two remaining questions - question (ix) and (x) seek conclusion from the users' point of view on the necessity of the Helper role for both Pacman and Ghost. Surprisingly, the results vary slightly instead of being close to each other - 92.3%

believed that Pacman needs a Helper, while a slightly lesser 82.4% thought Ghosts required such support. While the majority agreed on the essential existence of Helpers, a small group commented that it might be better to let the physical players to explore on their own as there could be more fun in self exploration, especially for Ghost whose sole job is simply to eliminate the Pacman. Nonetheless, the fact that a large majority agreed on the necessity of the Helper indicated that it is not something added for the sake of adding; it serves a purpose in completing the whole framework of the game.

3.6.2 Analysis of Message Logs

In the user study conducted, we also analyzed some excerpts from the text messages that contain the conversation which was logged during the user study. Some interesting and unexpected results have been obtained and should be useful to in order to gain more knowledge in the context of real-to-virtual interaction which we are studying here.

One significant result from the user study is illustrated by the fact that the players often assumed beforehand what their team member would like to know or ask. A typical message log often looks like this:

Pacman's Helper : Carry on.

Pacman's Helper : Go right.

Pacman's Helper : Go straight.

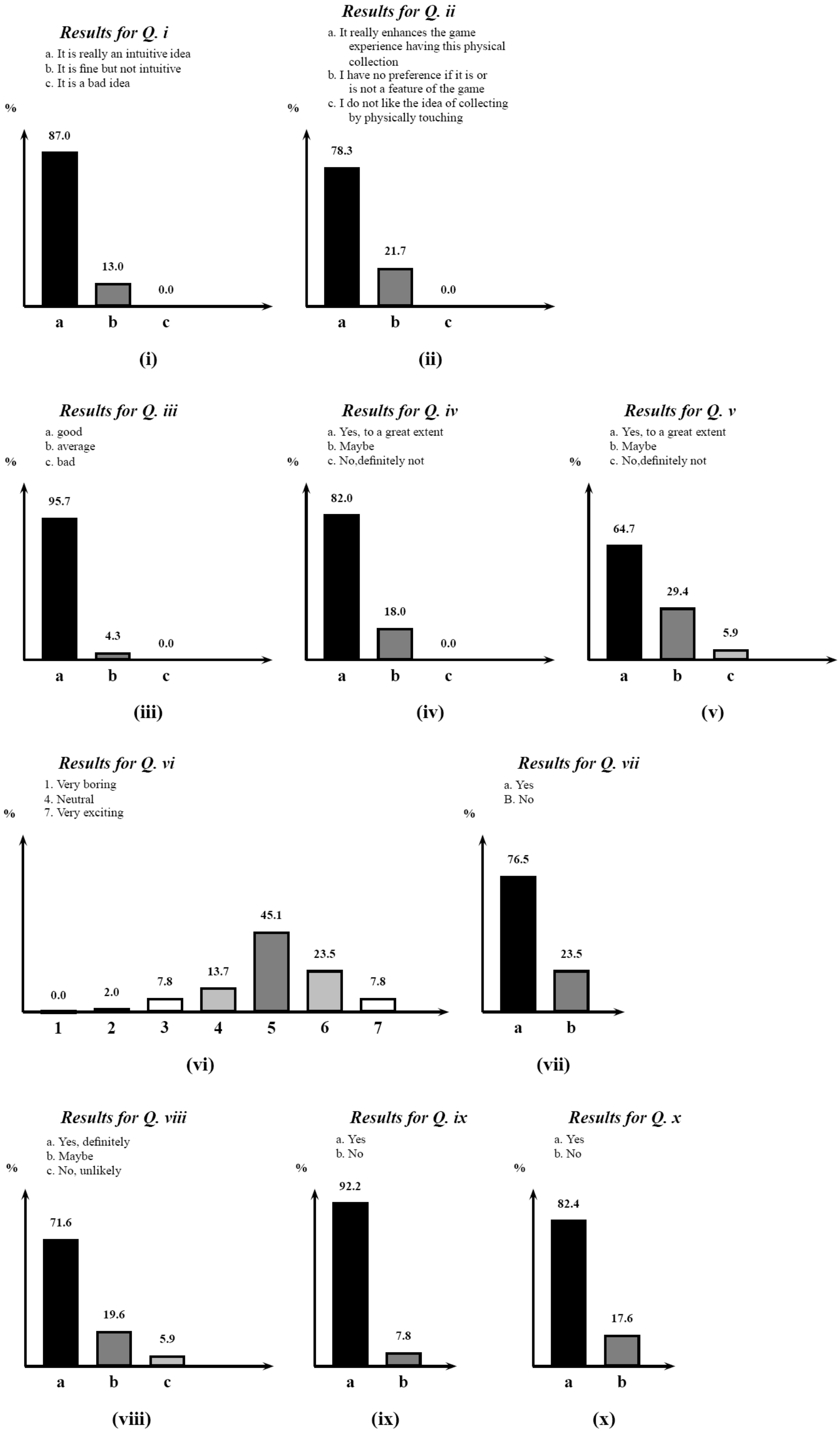


Figure 3.35: Graph of the results for all multiple-choice questions

Pacman's Helper : Left.

Pacman's Helper : Go left!!!

Pacman's Helper : Run!!!

Here the Pacman remained silent and her helper dominated the conversation. As a matter of fact, the Pacman was preoccupied with her physical activities of moving and searching that she could find no time to reply. Moreover, probably all that she needed (and did not need) to know was already answered by her Helper. The conversation had in fact turned into a monologue and interaction has lost its meaning here.

However, for users who are more proactive, we can still observe some two-way traffic of interactions taking place even in trivial cases of guiding the way.

Pacman : What do you mean?

Pacman's Helper : Try going backwards.

Pacman's Helper : Stop!

Pacman : What's next?

Pacman's Helper : Turn right

Therefore, despite the same settings, we recognize that the degree of interactions that take place might vary for different individuals. Encouraging the players to collaborate in reaching the common goal is essential to promote more interactions.

The importance of collaborations often becomes obvious in the face of difficult problems that arise during critical moments, for instance toward the near end of the game. The role of Helpers in the following scenarios contributes significantly to the eventual outcome.

Pacman : Where's the last cookie?

Pacman's Helper : I suggest you try to kill Ghost first.

Pacman : Why?

Pacman's Helper : Coz he is still guarding the last cookie!

Pacman : What shall I do!?

While the element of surprise seemed to be spoilt by having a Pacman's Helper informing the Pacman what actually happened, it in fact did not spoil the game or caused a stalemate. The next scenario followed from the previous scenario, and the Ghost decided a strategy after she obtained the information from the Helper.

Ghost's Helper : 1 cookie left for the Pacman.

Ghost : This one??

Ghost : Tell me when he's coming.

Ghost: I'm gonna camp here.

Ghost's Helper : Incoming! But... she's a Super Pacman now!

Through Helpers on the both sides, each had devised their own strategies. The Ghost, after informed by her Helper that only one last cookie is remaining for the Pacman decided that it would be a wise idea to lay ambush near the cookie. On learning the Ghost idling around the last cookie, the Pacman's Helper passed the information to the Pacman. To outwit the Ghost, the Pacman found the Super Cookie with the aid of the Helper, and became a Super Pacman. Subsequently she eliminated the Ghost by surprise, finished the last cookie and won the game.

Pacman's Helper : Cool we won!

Pacman : Yeah, cool stuff, great job!

Pacman's Helper: Same to you.

Through simple text messaging, ideas and emotions are channelled mutually between real physical and virtual online players, resulting in unique interplay of social behaviors across different platforms.

3.6.3 Summary of Findings

Through the user study survey, we evaluated how the Human Pacman system model fits into the interaction theme we started out with through the users' responses.

We confirmed that people do enjoy the tangibility effect of physical interaction with humans and physical objects despite the fact that this is through the electronic hardware that acts as a medium. The users also welcomed the ideas of using body movements. The importance and benefits of virtual players to overcome physical world limitations through the demonstration Helper were also well recognized by the results.

Nonetheless, the mean rating of 4.96 out of 7 received for the Helper role in terms of enjoyment level indicated that perhaps more features should be added to enhance the role. Interaction in the form of text messaging alone may not be sufficient to encourage Helpers to stay engaged or maintain their level of interest. Consequently, some users reflected that even if the overall game complexity is increased, the level of interaction might not increase proportionally. This would mainly be due to the fact that they will still be primarily guiding the physical players through text messaging. Besides, some might have rejected the game as boring in the first place, no matter what the complexity.

We also found that interactions might not always take place readily as what the designers had intended in the early stage, whereby the game was actually customized to promote two-way communication between the physical players and the virtual online players. However, the a priori knowledge of users on their partner's requirement often led to pre-assumptions of which in turn led to one-way communication, i.e. Helper giving commands on where to go while Pacman/Helper just follow the instructions. Nonetheless, as the users approach critical stages (collecting the final cookie or capturing the Pacman) collaborations become inevitable as the users interact with each other for discussion to achieve their ultimate goal of winning the game.

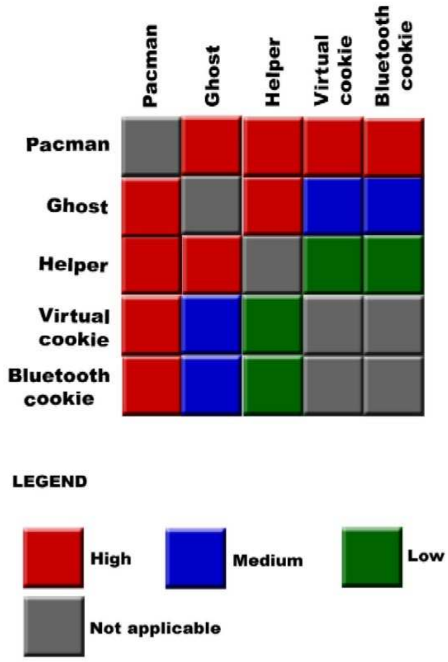


Figure 3.36: Interactive matrix

Based on the user study, we did some qualitative analysis on users' perceptions toward the different form of interactions and produced a color interactive matrix as shown in Figure 3.36. We assigned different levels of importance, in terms of high, medium and low, to each interaction. The levels of importance are obtained based on the user study results of the general effectiveness (useful, intuitive, enjoyable) of each form of interaction. Evidently, Pacman experiences most of the important interactions compared to other users, as the game design is initially revolved around the Pacman character to a larger extent. Interactions between users of any roles are also higher than to that of between users and objects, implying the importance of social interactions in mixed reality gaming environment.

3.7 Concluding Human Pacman

The continual propagation of digital communication and entertainment in recent years forces many changes in societal psyche and lifestyle, i.e. how we think, work and play. With physical and mobile gaming gaining popularity, traditional paradigms of entertainment will irrevocably shake from the stale television-set inertia. We believe that Human Pacman heralds the conjuration and growth of a new genre of computer game that is built on mobility, physical actions and the real world as a playground. Reality, in this case, is becoming more exotic than fantasy because of the mixed reality element in the game play.

Chapter 4

Magic Land

Magic Land is a mixed reality game, in which players are free to use their imagination and creativity to design the game story and rules. It demonstrates novel ways for users in real space to interact with virtual objects and virtual collaborators. In the game, users are able to capture 3D humans models and "play" with these models in a novel manner, such as picking them up and placing them on a desktop or getting them to interact with other virtual 3D models.

Magic Land symbolizes the dawn of an era as real and virtual interactive experience grow to become part and parcel of daily lives. In this chapter, we shall examine the contributions to the research field in realizing Magic Land. This will be followed by details of the actual game play and the system design.

4.1 Highlights of Magic Land

There are several points of highlights in Magic Land, which includes technical achievements and contributions to the research field, in realizing Magic Land Project are as follows:

- Real Time and Live Human 3d Recording System

  Magic Land encompass a complete, robust and real time human 3d recording system, from capturing images, processing background subtraction, to rendering for novel view points. Originating from the previous system [28], this novel system was developed by integrating new techniques to improve speed and quality. For background subtraction, the quality was improved by filtering misclassified pixels and the network speed was increased by optimizing the data size. For 3D rendering, new methods were utilized to compute visibility and blend color. These contributions have significantly improved quality and performance of the system, and are very useful for mixed reality researchers.

- MXRToolkit Software Package

  MXRToolkit software package was developed and used in this research project. The MXR Software Development Kit consists of a library of routines to help with all aspects of building mixed reality applications, together with a number of applications to aid with common tasks such as camera calibration, camera tracking, etc. The software is open-source and released under the

GNU General Public License.

- Wide Area Stereo Vision Tracking

  Magic Land is an indoor mixed reality entertainment system and as discussed in the earlier chapter, such systems require very stringent tracking for the virtual and real environment integration to be seamless. Hence, in this system, 4 dragonfly cameras were used to implement stereo vision tracking over the entire Interactive Table. Stereo Vision Tracking, using libraries in ARToolKit, ensures tracking accuracy of millimeters and hence, the virtual world can be fused seamlessly with the real world.

- Virtual and Physical Interactive System

  Magic Land signifies a cross-section where art and technology meet. It not only combines latest advances in human-computer interaction and human-human communication: mixed reality, tangible interaction, and 3D Live technology; but also introduces to artists of any discipline intuitive approaches of dealing with mixed reality content. It brings together the processes of art creation, acting and reception in one environment, and creates new forms of human interaction and self reflection. Moreover, future development of this system will create a new trend of mixed reality games, where players actively play a role in the game story.

4.2 Game Play

Magic Land is a mixed reality environment where 3D Live captured avatars of human and 3D computer generated virtual animations play and interact with each other. The system includes two main areas: recording room and interactive room.

- Recording Room

  The recording room is where users can have themselves captured into live 3D models which will interact in the mixed reality scene. This room, which has nine Dragonfly cameras mounted inside, is a part of the 3D capture system described above. After the user gets captured inside the system, she can go to the interactive room to play with her own figure. The recording area plays the role of the interface between human being and computer. It is also a special experience for the users to watch themselves acting in 3D on the interactive table from the external point of view like the "Bird in the sky". Figure 4.1 are two bird's eye views of this system.

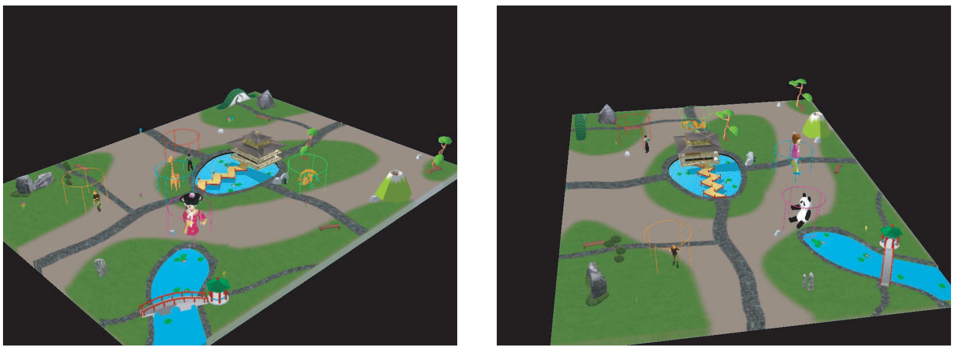


Figure 4.1: Main Table: The bird's eye views of the Magic Land.

- Interactive Room

The interactive room consists of three main components: a Menu Table, a Main Interactive Table, and five playing cups. On top of these tables and cups are different marker patterns. A four cameras system (ceiling tracking system) is put high above the Main Interactive Table to track the relative position of its markers with the markers of the cups currently put on it. The users view the virtual scenes and/or virtual characters which will be overlaid on these tables and cups via the video-see-through HMDs with the Unibrain cameras mounted in front and looking at the markers. The Main Interactive Table is first overlaid with a digitally created setting, an Asian garden in our case, whereas the cups serve as the containers for the virtual characters and also as tools for users to manipulate them tangibly. There is also a large screen on the wall reflecting the mixed reality view of the first user when he/she uses the HMD. If nobody uses this HMD for 15 seconds, the large screen will change to the virtual reality mode, showing the whole magic land viewed from a very far distant viewpoint.

4.2.1 Tangible Interactions in Magic Land

The Menu Table is where users can select the virtual characters they want to play with. There are two mechanical push buttons on the table corresponding with two types of characters: the human captured 3D Live models on the right and VRML models on the left. Users can press the button to change the objects showed on the Menu Table, and move the empty cup close to this object to pick it up. To empty

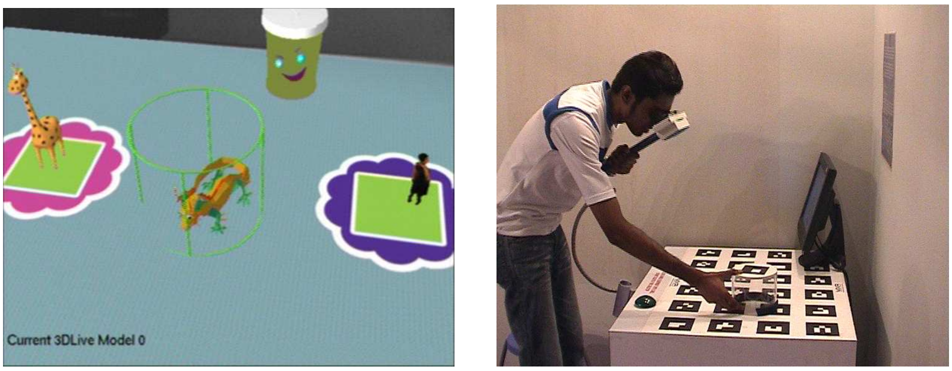


Figure 4.2: Tangible Interaction on Menu Table

a cup (trash), users can move this cup close to the virtual bin placed at the middle of the Menu Table. In the Figure 4.2, in the left image, we can see a user using a cup to pick up a virtual object. At the edge of the table closest to the user are two mechanical buttons. In the right image we can see the augmented view seen by this user. The user had selected a dragon previously which is inside the cup.

Figure 4.3: Tangible Interaction on Interactive Table

After picking up a character, users can bring the cup to the Main Interactive Table to play with it. Consequently, there will be many 3D models moving and interacting in a virtual scene on the table, which forms a beautiful virtual world of those small characters. If two characters are close together, they would interact with each other in the pre-defined way. For example, if the dragon comes near to the 3D Live captured real human, it will blow fire on the human. This gives an exciting feeling of the tangible merging of real humans with the virtual world. Another example of the interaction is seen in the Figure 4.3. In this case, the witch, which can be moved tangibly using the cup, turns the 3D Live human character into a stone when he comes physically close to her.

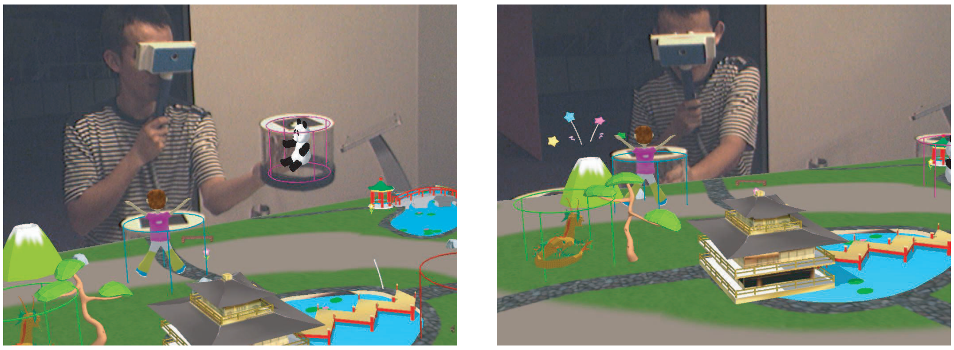


Figure 4.4: Tangible interaction on the Main Table.

Another example of the tangible interaction on the Main Interactive Table is shown in the Figure 4.4. Here, a user used a cup to tangibly move a virtual panda object (left image) and using another cup to trigger the volcano by putting the character physically near the volcano (right image).

4.3 System Design

As shown in Figure 4.5, the software system of Magic Land consists of five main parts: 3D Live Recording, 3D Live Rendering, Main Rendering, Ceiling Camera Tracking, and Game Server. Beside these parts, there is a Sound module that produces audio effects including background music and interactive sounds for the

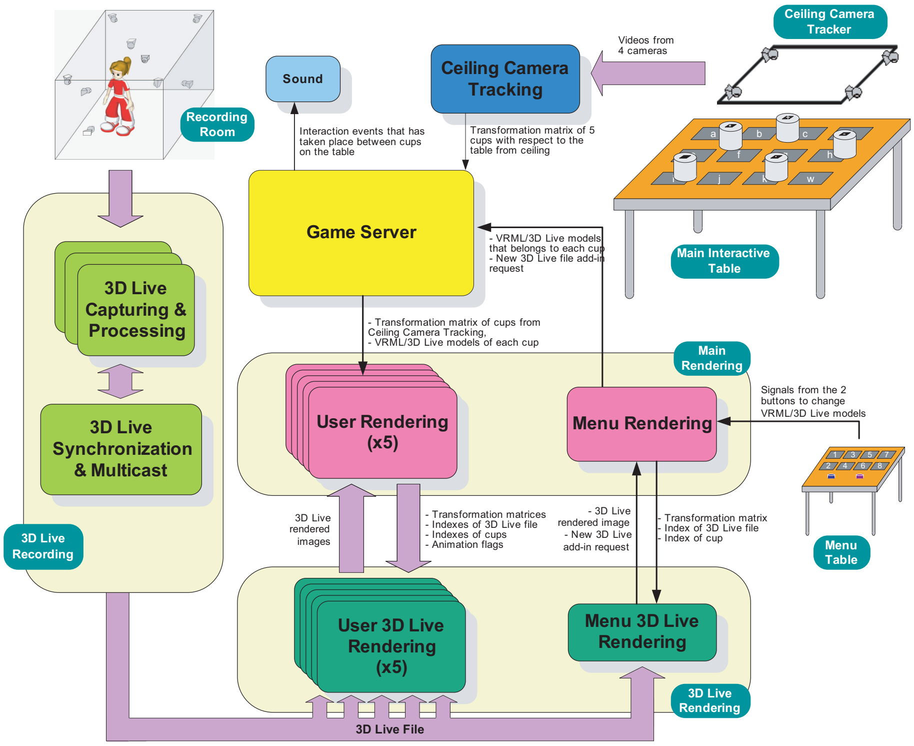


Figure 4.5: System Overview of Magic Land

whole system.

In this system users can record their live model for playback. The 3D Live Recording and 3D Live Rendering parts are a recording capturing system described in the previous section. After going inside the recording room and pressing a button, the user will be captured for 20 seconds. The captured images are then processed and sent to all 3D Live Rendering modules. However, unlike the live version which sends the processed images of nine cameras immediately for each frame, the recorded version sends all the processed images of all the frames captured in 20 seconds at a time. Another difference is that, instead of using TCP/IP to send the 3D Live data to each User 3D Live Rendering and Menu 3D Live Rendering module of the 3D Live Rendering part, we use multicast to send the data to all of them. This helps to utilize bandwidth of the network as well as to ensure that all the receivers finish receiving data at the same time.

The Main Rendering part includes a Menu Rendering module and five User Rendering modules. These modules track the users' viewpoints, and render the corresponding images to the users. First, they obtain images from the Unibrain cameras mounted on the users' HMDs, track the marker patterns and calculate the transformation matrix relating the coordinates of these markers with the coordinate of the camera. After that, basing on the transformation matrix, each module will render the image and output the result to the corresponding HMD. Especially, the Menu Rendering module also handles the users' inputs when they press the buttons on the Menu Table, or when they use the cups to select and remove virtual characters.

The Ceiling Camera Tracking module receives images from four cameras put above the Main Interactive Table. It tracks the markers of the table and cups, and calculates the transformation matrices of the cups relative to the table from top view. After that, it sends these matrices to the Game Server.

Finally, Game Server is the heart of the system. It links all the modules together. It receives and forwards information from the Ceiling Camera Tracking, Menu Rendering and User Rendering modules. This Game Server coordinates and synchronizes what every user has in their cup in terms of type of the character and its animation, position and orientation. First of all, it receives the camera tracking data from the Ceiling Camera Tracking module and determines the interaction between the characters inside the cups, basing on the distances between cups. After that, it forwards this interaction information to the User Rendering and Sound modules so that these modules can render the respective animations and produce the corresponding interactive sound. The ceiling camera tracking data is also forwarded to the User Rendering modules for usage in the case that the user's camera lost the tracking of their cups' marker. When the users select a new character, the Game Server also receives the new pair of cup-character indexes from the Menu Rendering and forwards to all the User Rendering modules to update the change.

4.4 Hardware

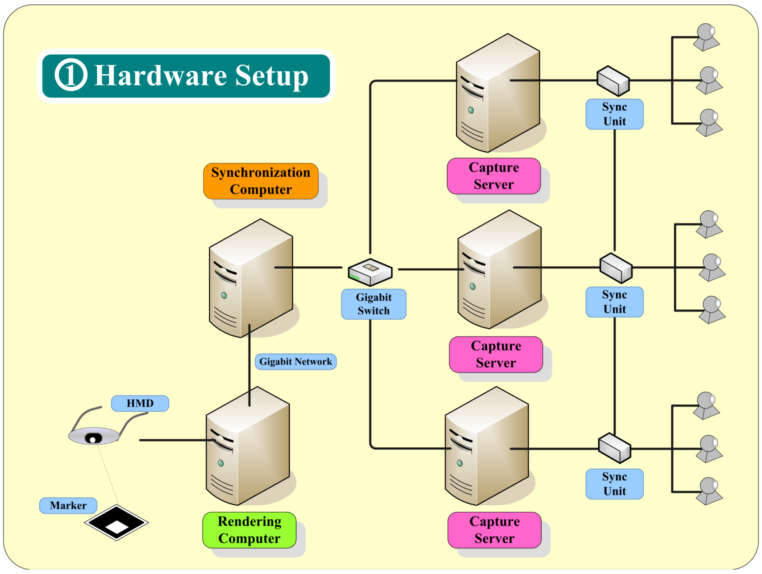


Figure 4.6: Hardware Architecture

Figure 4.6 represents the overall system structure. Eight Dragonfly FireWire cameras from Point Grey Research [5], operating at 30fps, 640 x 480 resolution, are equally spaced around the subject, and one camera views him/her from above. Three Sync Units from Point Grey Research are used to synchronize image acquisition of these cameras across multiple FireWire buses. Three Capture Server machines, each one being DELL Precision Workstation 650 with Dual 2.8GHz Xeon CPUs and 2GB of memory, receive the three 640x480 video-streams in Bayer format at 30Hz from three cameras each, and pre-process the video streams.

The Synchronization machine is connected with three Capture Sever machines through a Gigabit network. This machine receives nine processed images from three Capture Server machines, synchronizes them, and sends them also via gigabit Ethernet links to the Rendering machine, which is another DELL Precision Workstation 650. The user views the scene through a video-see-through head mounted display (HMD) connected directly to the Rendering machine. A Unibrain firewire camera, capturing 30 images per second at a resolution of 640x480, is attached to the front of this HMD. The Rendering machine obtains images from this Unibrain camera, tracks the marker pattern on these images, calculates the position of the virtual viewpoint, generates a novel view of the captured subject from this viewpoint and then superimposes this generated view to the images obtained from the Unibrain camera and display it on the HMD.

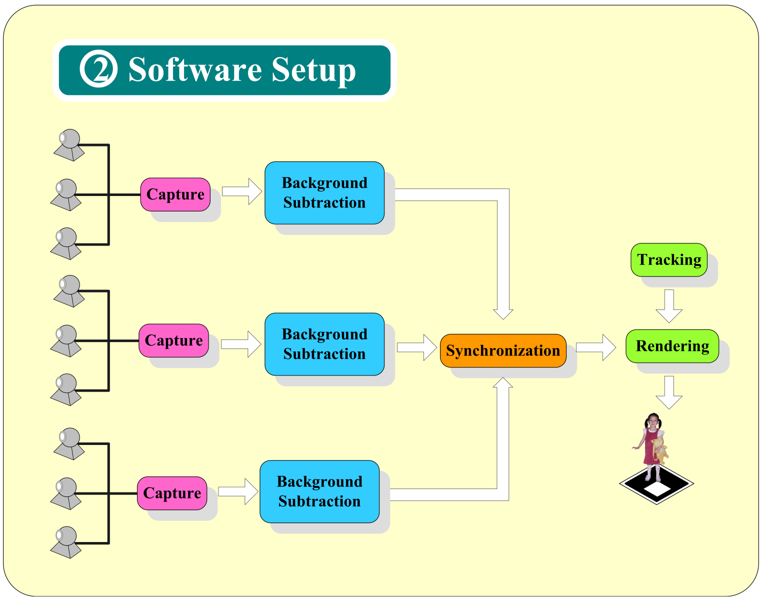


Figure 4.7: Software Architecture

4.5 Software

All basic modules and the processing stages of system are represented in Figure 4.7. The Capturing and Image Processing modules are placed at each Capture Server machine. After Capturing module obtains raw images from the cameras, the Image Processing module will extract parts of the foreground objects from the background scene to obtain the silhouettes, compensate for the radial distortion component of the camera mode, and apply a simple compression technique.

The Synchronization module, on the Synchronization machine, is responsible for getting the processed images from all the cameras, and checking their timestamps to synchronize them. If those images are not synchronized, basing on the timestamps, Synchronization module will request the slowest camera to continuously capture and send back images until all these images from all nine cameras appear to be captured at nearly the same time. The Tracking module will obtain the images from the Unibrain camera mounted on the HMD, track the marker pattern and calculate the Euclidian transformation matrix relating the marker co-ordinates to the camera co-ordinates.

After receiving the images from the Synchronization module, and the transformation matrix from the Tracking module, the Rendering module will generate a novel view of the subject based on these inputs. The novel image is generated such that the virtual camera views the subject from exactly the same angle and position as the headmounted camera views the marker. This simulated view of the remote collaborator is then superimposed on the original image and displayed to the user. The subsequent parts will discuss in more detail about the techniques we use in each module.

4.5.1 Image Processing Module

The Image Processing module processes the raw captured image in three steps:

1. background subtraction (which extracts parts of the foreground objects from the image to obtain the silhouettes), radial distortion compensation, and image size reduction.

2. The second step is done by applying the intrinsic parameters of the camera to estimate the correct position of each pixel.

3. The remaining of this part will concentrate on the background subtraction

and image size reduction steps.

- Background subtraction

  The result of visual hull construction in the Rendering module depends largely on the output of background subtraction step. This pre-processing step is one of the most crucial steps to determine quality of the final 3d model. Not only having to produce the correct foreground object, the chosen background subtraction algorithm must be very fast to fulfill the realtime requirement of this system. Another important requirement to guarantee the good shape of the visual hull is that the background subtraction algorithm must be able to eliminate the shadow caused by the objects.

  There are many works on background subtraction, which produce rather good results, such as [29], [30], [31]. However, there normally exist the significant trade-off between processing time and quality of the result. The simple statistical method we used in our previous work on 3D Live [32] is very fast, but does not produce good enough quality. To fulfill our needs, we use a modified method based on the scheme of Horpraset [29], which has the good capabilities of distinguishing the highlighted and shadow pixels. However, this algorithm has been modified in our research to reduce the computational intensiveness and optimize for the real time constraints of this system.

  The main idea of this method is to learn the statistics of properties of each background pixel over N pre-captured background frames, and obtain the statistical values modelling for the background. The pixel properties to be calculated here are the chromaticity and the brightness which is obtained from a new model of the pixel color. Basing on this, the algorithm can then classify each pixel into "foreground", "background", "highlighted background" or "shadow/shading backgroun" after getting its new brightness and chromaticity color values. In our application, we only need to distinguish the "foregroun" type from the rest.

The new color model which separates the brightness from the chromaticity component is summarized in Figure 4.8. In this RGB color space, the point $I(i)$ represents the color value of pixel i_th, and $E(i)$ represents the expected color value of this pixel, which coordinates $(\mu_R(i), \mu_G(i), \mu_B(i))$ are the mean values of the R, G, B components of this pixel obtained from the learning stage. $J(i)$ is the projection of $I(i)$ on the line $OE(i)$.

The brightness distortion (α_i) and color distortion (CD_i) of this pixel are defined and calculated as:

$$\alpha_i = \frac{J(i)}{E(i)} = argmin_{\alpha_i} \left[\left(\frac{I_R(i) - \alpha_i \mu_R(i)}{\sigma_R(i)} \right)^2 + \left(\frac{I_G(i) - \alpha_i \mu_G(i)}{\sigma_G(i)} \right)^2 + \left(\frac{I_B(i) - \alpha_i \mu_B(i)}{\sigma_B(i)} \right)^2 \right] \quad (4.1)$$

$$CD_i = \sqrt{ \left(\frac{I_R(i) - \alpha_i \mu_R(i)}{\sigma_R(i)} \right)^2 \left(\frac{I_G(i) - \alpha_i \mu_G(i)}{\sigma_G(i)} \right)^2 \left(\frac{I_B(i) - \alpha_i \mu_B(i)}{\sigma_B(i)} \right)^2 } \quad (4.2)$$

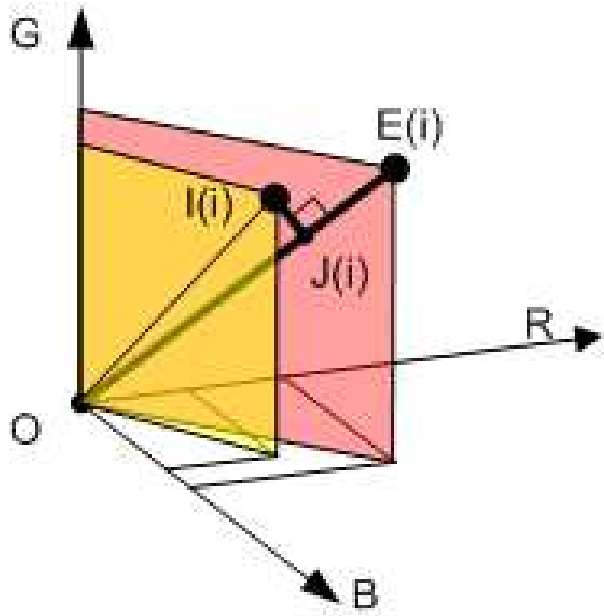


Figure 4.8: Color model

In the above formula, $\sigma_R(i), \sigma_G(i), \sigma_B(i)$ are standard deviations of the i_th pixel's red, green, blue values computed in the learning stage. In our version, we assume that the standard deviations are the same for all pixels to make CD_i formula simpler:

$$CD_i = (I_R(i) - \alpha_i \mu_R(i)) (I_G(i) - \alpha_i \mu_G(i)) (I_B(i) - \alpha_i \mu_B(i)) \qquad (4.3)$$

Another assumption is that the distributions of α_i and CD_i are the same for all pixel i. With this assumption, we do not need to normalize α_i and CD_i as was being done in the previous work of [29].

These modifications reduce the complexity of the formula and quite drastically increase the calculation speed from 33ms/frame to 13ms/frame, but produce more small misclassified pixels than the original algorithm. However, these small errors can be easily filtered in the next step.

- Filtering

  The filtering step is necessary to remove the small misclassified regions. There are many filtering methods to process the images after background subtraction. However, regarding the real-time constraint, we use the simple morphological operators open and close to filter out small misclassified regions.

- Data size for real time network constraints

  One very important factor is the amount of data to transfer over the network. In order to reach the fastest network speed, the size of data has to be as small as possible. In our system, we try to optimize the data size by using two main following methods:

  1. Reducing the image size by only storing the smallest rectangular region containing the foreground objects. An algorithm is implemented to find out the contour of the foreground and base on this result to calculate the smallest bounding box. This finding the contour algorithm also acts as another filtering method, which filters all small misclassified foreground regions which contour lengths are less than a predefined threshold. The size of this smallest rectangular region bounding the foreground objects depends on how close the camera looks at the object, and how large

the object is. All cameras must be adjusted so that they view the object from a far enough distance to guarantee quality of the visual hull. Consequently, for each camera, the average size of this bounding box of the foreground is normally less than 1/8 the size of the whole image, which is a significant reduction in the data size.

2. Using Bayer format [33] with background in-formation encoded to store the images. Instead of using 3 bytes to encode 3 color components Red, Green, Blue for each pixel, we encode the whole image in Bayer format, which costs only 1 byte for each pixel. Moreover, for each pixel, the background information is encoded in the least significant bit of the byte at the position of this pixel in the Bayer image, value 1 for background pixel and 0 for foreground pixel. Obviously, this method of storing images leads to some color information lost. However, because the lost information is not much, the color quality of the output images is still good. Consequently, the lost information is trivial, compared with the benefit of reducing much data size, which is at least 3 times smaller than the RGB format with background information encoded.

- Results

  The sample results of the image processing step are shown in Figure 4.9. We can see that there are small errors after we subtract the background by our optimized algorithm. In the figure, the small green pixels inside the body is the foreground pixels misclassified as background ones, and the small black pixels outside the body is the background pixels misclassified as foreground ones. However, these errors are completely removed after the filtering step. The speed of this step is only around 15ms/frame. Compared with the non-simplified algorithm, which is 37ms/frame including the filtering step, the optimized algorithm is fast enough for this real time application.

Figure 4.9: Results of Background subtraction: before and after filtering

4.5.2 Synchronization Module

The main function of this module is receiving and synchronizing images having been processed by Image Processing module. The purpose of synchronization is to ensure that all images are captured at the same time. Figure 4.10 describes the data transferred from Image Processing to Synchronization. It includes three parts. The first part is the image which is processed by Image Processing Module. Instead of sending the whole image, we only transmit the smallest rectangle area of the

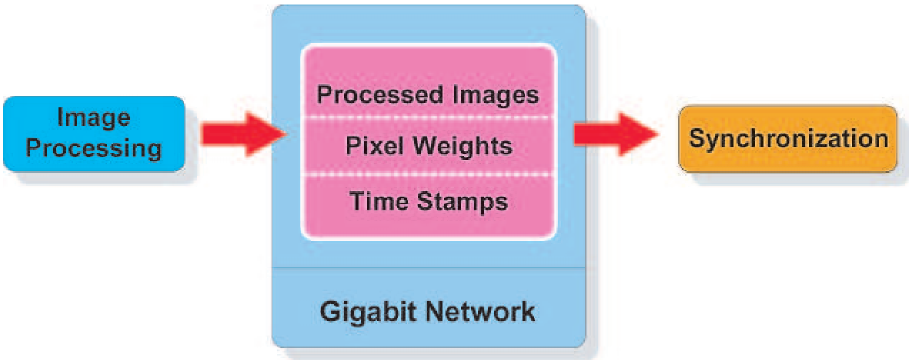


Figure 4.10: Data Transferred From Image Processing To Synchronization

original image that contains the silhouette. This significantly reduces the amount of data to be transmitted. The second part is the pixel-weights for this image. These weights will be used for blending color in the rendering steps. We will present more about this weight in the Rendering section of this paper. The last part to be transmitted is the Time Stamp, which is the time when this image is captured. Using this timing information, the Synchronization module will synchronize images captured from all nine cameras.

Once receiving one set of images from nine cameras, the time stamp of each image will be compared. If the difference in time between the fastest camera and the slowest camera is larger than 30 ms, the Synchronization module will require Image Processing module to provide a new image from the slowest camera. This synchronizing process will keep looping until the difference is smaller than 30ms. The reason to choose 30ms as the threshold is because our system operates at 30 fps.

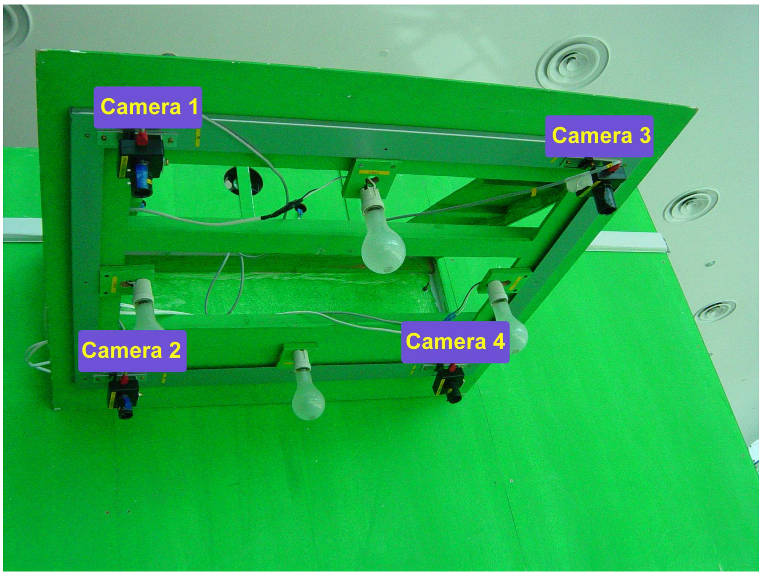


Figure 4.11: Ceiling Tracker Module

4.5.3 Ceiling Tracker Module

The Ceiling Camera Tracker module (see Figure 4.11) is developed using the stereo tracking library in ARToolKit4.068 and uses high resolution dragonfly cameras. Its main function is to track the cups' positions relative to the main table and send this information to the game server. When the server receives the tracking data from the Ceiling Camera Tracking module, it will use this information to determine whether there should be interaction between the characters inside the cups. This is normally based on the distances between cups. Apart from this, the bird's eye view of the whole Magic Land on the large screen of the wall is also drawn using the tracking information from the ceiling tracker. As the sole purpose of the ceiling camera is tracking and not rendering, the frame rate of the ceiling tracker can be increased by turning off the display.

Figure 4.12 shows the connections of the Camera Tracking module. There are

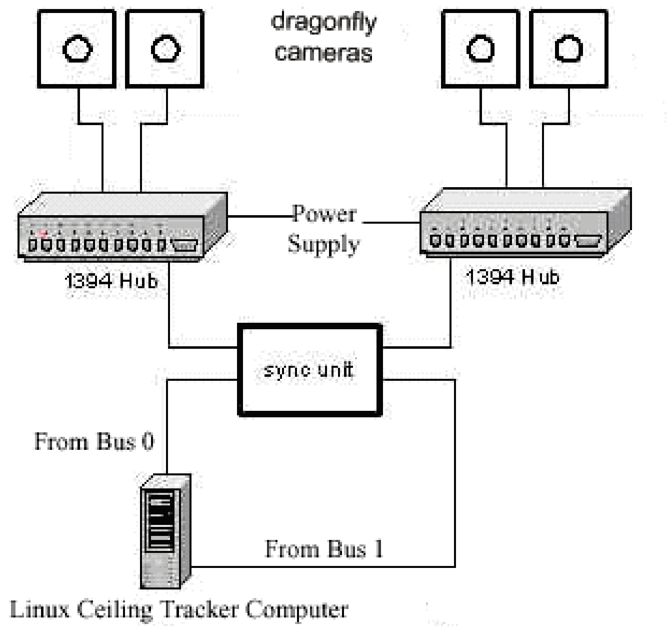


Figure 4.12: Ceiling Tracker module Connections

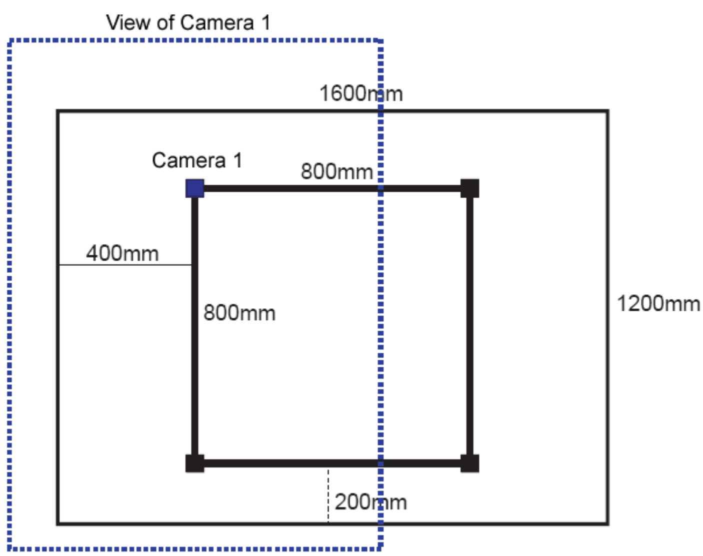


Figure 4.13: Camera Positioning

four cameras used for the Ceiling Tracker module. However, the bandwidth of a single firewire card bus can, at maximum, support up to three cameras. Hence, two firewire card buses are required. The purpose of the sync unit in Figure 4.12 is to synchronize the captures of the dragonfly cameras between the two firewire card buses. This is to ensure that the images captured by the cameras are within an instance of time, which is vital for stereo tracking.

Before mounting the dragonfly cameras onto ceiling, it is important to note that each camera needs to be carefully calibrated. This is done using ARToolKit (see Appendix B). It is also important for each camera's view to cover a sizable portion of the Main Interactive Table and that every part of the table is in the view of at least two cameras. Figure 4.13 shows the overall camera layout and the blue rectangle indicates the visible region of camera 1. The visible region for camera 1 should be use as a guideline for mounting the rest of the cameras. The flowchart showing the operation of the Ceiling Tracker is shown in Figure 4.14.

Initialization

The program starts by first reading in several default configuration files. They are as follows:

- Cup Configuration File

  This file contains the port number, unique ID of each cameras, shuttle speed, gain, threshold and white balance. The port number identifies the firewire card bus a particular camera, based on it's ID, should be connected to.

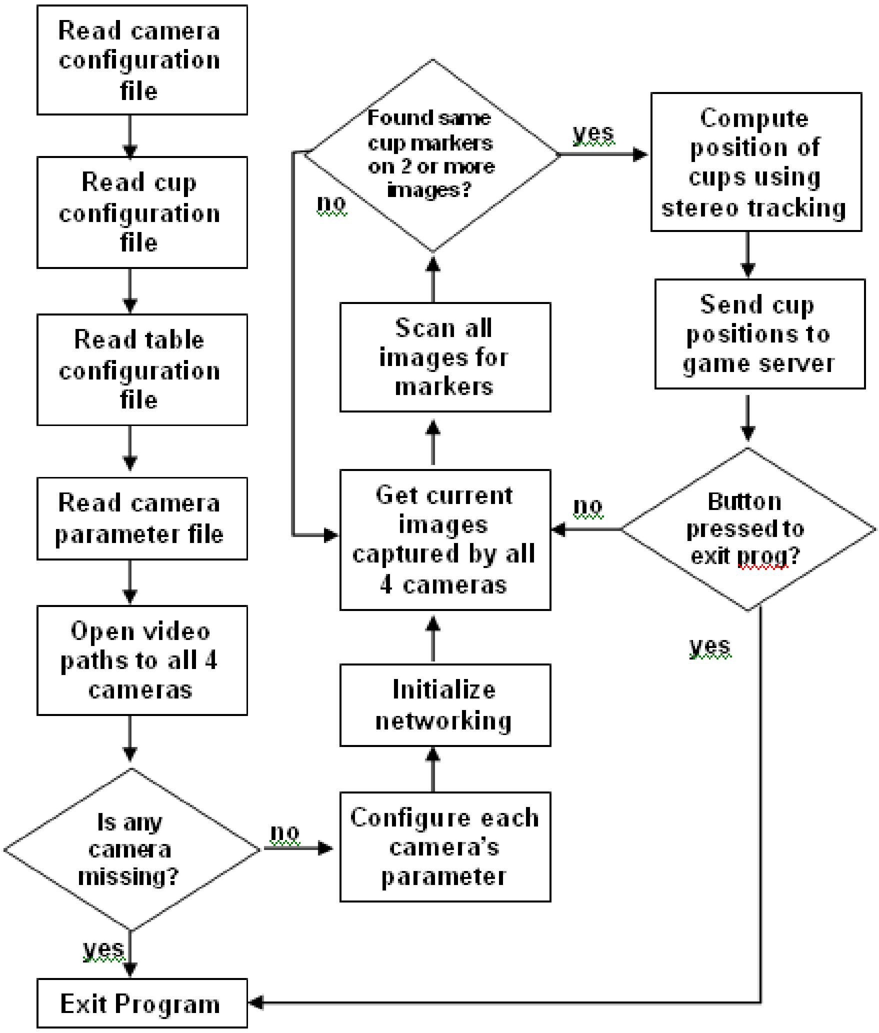


Figure 4.14: Ceiling Tracker Algorithm

The shuttle speed, gain and threshold directly affect the tracking accuracy of each camera. As each camera is, by itself, unique and placed at different position and orientation, these values need to be adjusted individually for each camera. This can be done inside the ceiling tracker program and once completed, can be saved over the old default.

The white balance is a parameter that affects the color of the image captured by the cameras. They have some impact on tracking accuracy but is not overly important.

- Cup Configuration File

  This file contains the information of each cup used in Magic Land. This includes the ID of the cup, it's width, and the x,y and z offset which are normally set to 0.

  The ID of each marker is important as the Ceiling Tracker module uses 2D matrix coding method in ARToolKit4.068 for tracking. This is achieve by calling the function `arSetPatternDetectionMode` in the initialization process. In this case, there is no need to call the function `arLoadPatt` to load in pre-trained patterns.

  Each marker's ID is represented by the 3x3 pattern shown in Figure 4.15. In our program, '1' denotes a black square and '0' denotes a white square. P1 and P7 in Figure 4.15 is always black ('1') and P9 is always white ('0'). The reason for this is to fix the orientation of the pattern and hence avoid

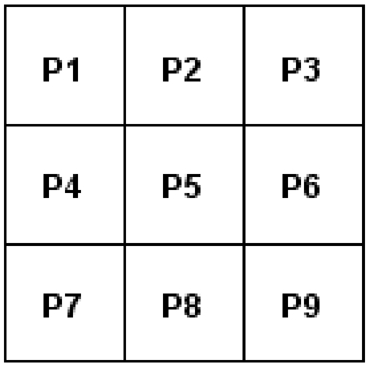


Figure 4.15: 2D Matrix Coding

confusion. The ID is derived from a 6-bit binary code form by P2 P3 P4 P5 P6 P8. Some examples of how a pattern's ID is derived are given in Table 4.1. The 6 bits representation in column 2 is formed by ignoring P1, P7 and P9 in column 1. The marker ID in column 3 is the decimal equivalent of the 6 bit binary code. Figure 4.16 shows the actual markers in the example.

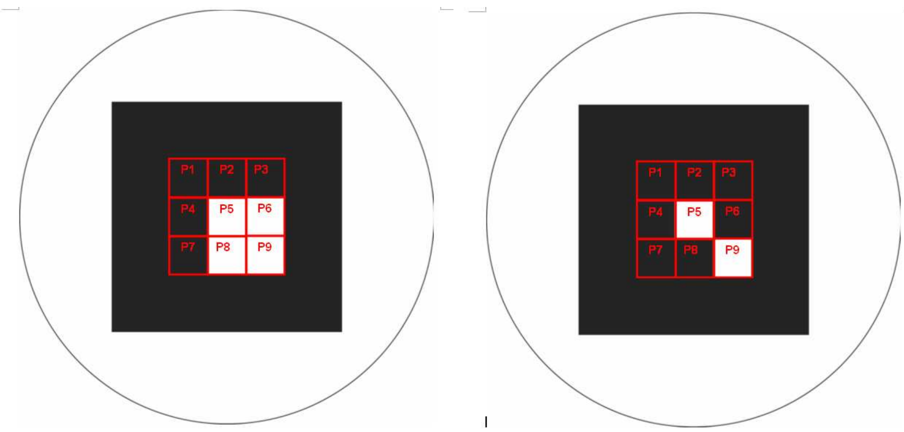


Figure 4.16: Actual markers given in Table 4.1

- Table Configuration File

  This file contains the IDs of all the patterns found on the Main Interactive

P1 P2 P3 P4 P5 P6 P7 P8 P9	**P2 P3 P4 P5 P6 P8**	**Pattern ID**
111 100 100	111000	56
111 101 110	111011	59

Table 4.1: Some examples of 2D matrix coding in ARToolKit4.068.

Table. This is also using the 2D matrix coding described earlier. The reason for having multiple patterns on the Main Interactive Table is to ensure robustness in the tracking. When some of the patterns are covered, the ceiling tracker can still track using the rest.

Apart from the patterns'ID, this file also contains the x,y location (in mm) of each pattern found on the table with respect to the center of the table. The unit of measurement (mm) is then use as a reference by the Ceiling Tracker module and subsequently, the cups' positions will also be tracked in the same unit of measurement.

- Camera Parameter Files

  Each of these files contains the intrinsic parameters of a camera. The intrinsic parameters of a camera include information like it's focal length and the optical center. This information is essential for tracking accurately. Hence, before mounting the cameras onto the ceiling, each camera needs to be carefully calibrated. This is done using ARToolKit (see Appendix B).

- Trans Files

  Each of these files contains the orientation and position of the camera relative to the center of the table with the z-axis perpendicular to the table top. x-axis and y-axis run along the width and length of the table respectively. The Trans files are vital for stereo tracking as the ARToolKit's software library needs these files to compute the relative position between two cameras.

After reading in all the default configuration files, the program proceeds to open the video paths to all the four cameras and set the default configuration of all the cameras. If the program is not able to detect any particular camera, it will exit with an error message. If all the cameras are in place, the program will initialize the network connection to the game server using TCP/IP.

Main Algorithm

After doing all the preliminaries of loading and initializations, the program will run in the following sequence in a loop:

1. Detecting Marker

Figure 4.17: The screen shots from the Ceiling Tracker module.

All the images captured at the same instance of time by the four cameras will be processed in the main loop. Using the `arDetectMarker` function in ARToolKit, the program scans these images for recognizable cup markers. If more than one camera pick up the same marker, the program will select the image with the least estimated error as the main reference. Subsequently, the program uses the stereo tracking library in ARToolkit and pairs this image with the other images that contain the same marker. The tracked position from the pair of images that yields the least error will be saved. In this manner, the most accurate tracking result for a cup is obtained. The left image in Figure 4.17 shows a screen shot from the Ceiling Tracker module. All the cups on the Main Interactive Table are detected by more than one cameras. This is indicated by the virtual cube drawn on each cup in the images. The right image shows the tracking results being put in actual use by Magic Land.

2. Update to server with cups information.

   After obtaining the tracked position of all the cup markers on the Main Interactive Table, the information will be appended onto a structure and sent to the server. The format of the structure is as shown below.

```
typedef struct {
int cupId;
```

```
int cupFound;
double cupTrans[3][4];
} CupTrackData;
```

The purpose of each element inside the structure is as follows:

- cupId

  The cupId helps to identify the cup. The server needs to differentiate between all the cups so that the correct virtual objects is rendered on the cups.

- cupFound

  This value will be either '0' or '1' to indicate whether the cup with the particular ID is found on the Main Interactive Table.

- cupTrans

  This is a 3x4 transformation matrix which contains the translation and orientation of the cup with respect to a common reference point on the Main Interactive Table.

3. Detect Button Press for Adjustments.

   There are several functions in the program that can be used to fine tune the performance of the ceiling tracking. These can be seen in the Table 4.2. A point to note is that after adjusting the threshold, gain or shuttle speed of each camera, we need to save these values (by pressing 's') over the current default so that when the program restarts, the optimal value selected previously will be loaded.

Buttons	Operations
+	Increase Threshold value of image
-	Decrease Threshold value of image
1	Decrease Shutter speed
2	Increase Shutter speed
3	Decrease Gain
4	Increase Gain
d	Toggle between B/W and Color/Grayscale image
c	Toggle between different camera viewpoint
e	Toggle between compensation mode(warping correction vs no correction)
s	Save cameras setting
o	Turn on/off graphics information
S	Toggle synchronization mode+
C	Re-calibration of camera relationship*
w	Save extrinsic parameters of cameras
p	Toggle between image processing mode
h	Display information

Table 4.2: Buttons and Operations available in Ceiling Tracker Program

The Ceiling Tracker module should not be confused with the camera tracker on the HMD of each user as both serve a different purpose. The camera tracker on the HMD uses a Unibrain camera and tracks the marker pattern (also using ARToolkit but non-stereo) and calculate the Euclidian transformation matrix relating the marker co-ordinates to the camera co-ordinates. Then, this information is use to by the rendering module to display the virtual images correctly. When a particular marker is partially covered by other real objects or outside the Unibrain camera's view, the information from the Ceiling Camera Tracker will be utilized instead to draw the virtual images.

4.5.4 Rendering Module

Our rendering algorithm used in this system is a new development over our previous algorithm which is described in [28]. It is based entirely on a per-pixel basis. We denote the desired image, the "virtual camera image" and its constituent pixels "virtual pixels". The virtual camera can be determined by taking the product of the (head mounted) camera calibration matrix and the estimated transformation matrix. Given this 4 x 4 camera matrix, the center of each pixel of the virtual image is associated with a ray in space that starts at the center and extends outward. Any given distance along this ray corresponds to a point in 3D space. We calculate an image based depth representation by seeking the closest point along this ray that is inside the visual hull. This 3D point is then projected back into each of the real cameras to obtain samples of the color at that location. These samples are then combined to produce the final virtual pixel color. In summary, the algorithm must perform three operations for each virtual pixel: determining the depth of the virtual pixel as seen by the virtual camera, finding corresponding pixels in nearby real images, and determining pixel color based on all these measurements. We briefly describe each of these operations in turn.

- Determining Pixel Depth

  The depth of each virtual pixel is determined by an explicit search starting at the virtual camera projection center and proceeding outward along the ray corresponding to the pixel center (see Figure 4.18). Each candidate

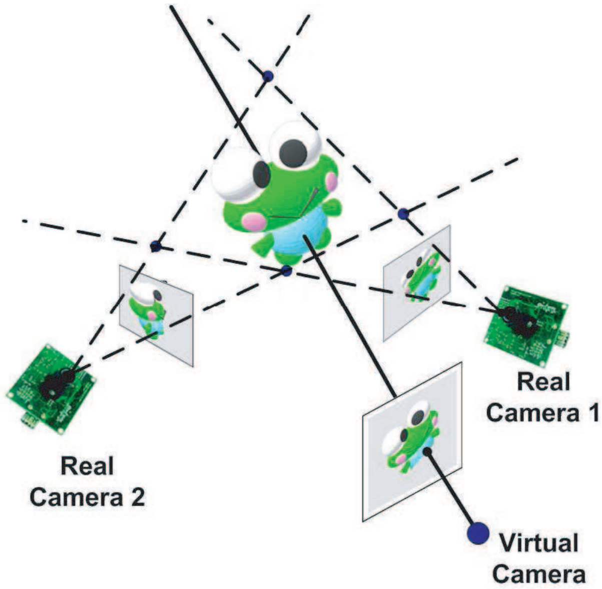


Figure 4.18: Novel View Point is generated by Visual Hull

3D point along this ray is evaluated for potential occupancy. A candidate point is unoccupied if its projection into any of the silhouettes is marked as background. When a point is found for which all of the silhouettes are marked as fore-ground, the point is considered occupied, and the search stops.

Using this method, we can generate the visual hull very efficiently. One problem with visual hull is that the geometry it reconstructs is not very accurate. When photographed by only a few cameras, the scene's visual hull is much larger than the true scene [34]. One well-known improvement for visual hull which have been discussed in [35], [36], [37], [34], [38], and [39] is to utilize color constraint. Although, using this constraint, we can generate "photo-hull" which is a better approximation than visual-hull, the rendering speed will be decreased significantly and thus this is not suitable for real-time applications. Alternatively, we reduce the errors of visual hull by using more cameras and a larger recording room.

- Finding Corresponding Pixels in Real Images

  The resulting depth is an estimate of the closest point along the ray that is on the surface of the visual hull. However, since the visual hull may not accurately represent the shape of the object, this 3D point may actually lie outside of the object surface. Hence, care needs to be taken in choosing the cameras from which the pixel colors will be combined. Depth errors will cause incorrect pixels to be chosen from each of the real camera views.

  To minimize the visual effect of these errors, it is better to choose incorrect pixels that are physically closest to the simulated pixel. So the optimal camera should be the one minimizing the angle between the rays corresponding to the real and virtual pixels. For a fixed depth error, this minimizes the distance between the chosen pixel and the correct pixel. We rank the cameras proximity once per image, based on the angle between the real and virtual camera axes.

  We can now compute where the virtual pixel lies in each candidate cameras image. Unfortunately, the real camera does not necessarily see this point in space - another object may lie between the real camera and the point. If the real pixel is occluded in this way, it cannot contribute its color to the virtual pixel.

- Determining Virtual Pixel Color

After determining the depth of a virtual pixel and which cameras have an unoccluded view, all that remains is to combine the colors of real pixels to produce a color for the virtual pixel. In the previous research, we took a weighted average of the pixels from the closest N cameras, such that the closest camera is given the most weight. This method can avoid producing sharp images that often contain visible borders where adjacent pixels were taken from different cameras. However, there are still some errors along the edge of the silhouette. In next section, we propose a new method to blend color which can overcome this problem.

4.6 Concluding Magic Land

Magic Land demonstrates novel ways for users in real space to interact with virtual objects and virtual collaborators. Using the tangible interaction and the 3D Live human capture system, the system allows users to manipulate the captured 3D humans in a novel manner, such as picking them up and placing them on a desktop, and being able to "drop" a person into a virtual world using users' own hands. This offers a new form of human interaction where one's hands can be used to interact with other players captured in 3D Live models. The artistic aspect of this installation introduces to artists easy, tangible and intuitive approaches in dealing with mixed reality content. The main challenge of the project is to create a new medium located somewhere between theater, movie and installation. The outcome of the project is an infrastructure that gives artists new opportunities to transport audiovisual information and encourage artists of any discipline to deal with those new approaches.

Chapter 5

Conclusion

5.1 Summary

Computer gaming has become a dominant force in recent years due to its higher level of attractiveness to game players. The most obvious advantage it has over traditional games is its seemingly unlimited ability to create graphical illusions to satisfy the fantasies of imaginative minds. However, most of the time, playing computer games decreased physical activities and social interactions.

Hence, there is a need to fill in this gap by bringing more physical movements and social interactions into games while still utilizing the benefit of computing and graphical systems. To address this issue, a lot of research has been carried out to introduce new forms of gaming that incorporates physical movements. Tangible mixed reality gaming is one solution to this problem as it fuses the exciting interactive features of computer gaming with the real physical world.

Much earlier on, in this thesis, we have covered some background literature on social and physical computing and mixed reality. Some recent works that capitalized on the modern technology to create entertaining mixed reality games are also highlighted. Then, we discussed in detail about the work we have done, namely, Human Pacman and Magic Land. The subsequent parts summarize the novel aspects of each project.

In Human Pacman, our research work encompasses several novel aspects of human computer interactive entertainment. Firstly, Human Pacman takes mobile gaming to a new level of sophistication by incorporating virtual fantasy and imaginative play activity elements, factors which propelled the popularity of computer game [40], with the implementation of Mixed Reality on the Head Mounted Displays (HMD).

Secondly, Human Pacman also explores novel tangible aspects of human physical movement, senses and perception, both on the player's environment and on the interaction with the digital world.

Thirdly, users enjoy unrestricted movement outdoor and indoor while maintaining social contact with each other. Players interact both face-to-face with other players when in proximity or indirectly via the wireless local area network (WLAN).

Human Pacman ventures to elevate the sense of thrill and suspended disbelief of the players in this atypical computer game. Each of the novel interactions mentioned is summarized in Table 1.

In Magic Land, our aim is to develop a complete system for novel real-time

Table 5.1: Detail descriptions of each novel features of Human Pacman.

Feature	**Details**
Physical Gaming	Players are physically role-playing the characters of Pacmen and Ghost; with wearable computers donned, they use free bodily movements as part of interaction between each person, and among objects in the real wide area landscapes and virtual environments.
Social Gaming	Players interact both directly with other players when they are in physical proximity, or indirectly via the Wireless LAN network by instant messaging. All Internet users can participate in the game by viewing and collaborating with real Human Pacmen and Ghosts.
Mobile Gaming	Players are free to move about in the indoor\outdoor space without being constrained to the 2D\3D screen of desktop computers.
Ubiquitous Computing	Everyday objects throughout the environment seamlessly have a real-time fantasy digital world link and meaning. There is automatic communication between wearable computers and Bluetooth devices embedded in certain physical objects used in game play.
Tangible Interaction	Throughout the game people interact in a touch and tangible manner. For example, Players need to physically pick up objects and tap on the shoulder of other players to devour them.
Outdoor Wide-Area Gaming Arena	Large outdoor areas can be set up for the game whereby players carry out their respective missions for the role they play.

capturing and rendering 3D images of live subjects in a mixed reality environment. We believe that this is a significant step towards the goal of perfect "tele-presence" for remote collaborations in the near future.

Beside being a remote live 3D conferencing and collaborating system, Magic land can also be perceived as an experimental laboratory and a wide range of artistic content, which is only limited by the imagination of the creators, can be added. To watch the scene from above with the possibility of tangible manipulation of elements creates a new form of art creation and art reception that generates an intimate situation between the artist and audience. The project itself brings together the processes of creation, acting and reception in one environment. These processes are optimized to the visitors experience in order to better understand the media and lead to a special kind of self reflection. The recording area plays the role of the interface between human being and computer. It is also a special experience for the users to watch themselves acting in 3D on the interactive table from the external point of view like the "Bird in the sky". This system demonstrates many technologies in human computer interaction: mixed reality, tangible interaction, and 3D communication.

5.2 Conclusion and Future Work

During this course of study, we have utilized our engineering skills and HCI knowledge, to fully capitalize on the technologies that are already available and created entertainment systems that is built on social and physical interactions. Being game prototypes, there are definitely more grounds for improvement in Human Pacman as well as Magic Land.

Further enhancement on the Human Pacman system can be made in terms of miniaturization and improvement in robustness of the wearable computers. Other accurate, sensitive and precise positioning methodologies and systems could be explored to enhance the realism of the game. Given the small network data exchange of the Human Pacman system, existing mobile GPRS network infrastructure could be tapped into to create a truly on-the-move gaming system happening anytime, anywhere. The use of haptic data gloves could also add a new intuitive dimension to the collection of virtual cookies, bringing a tangible feel to an intangible object. In time to come, emphasis on physical actions might even bring forth the evolvement of professional physical gaming as competitive sport of the future, for example "Pacman International League".

For Magic Land, enhancement can be make by exploiting "real time" capability of 3D Live technology, in which, outside players can see on the Main Interactive Table the 3D images of the person inside the recording room in real time. In the current version, the processed images of all the frames captured can only be sent 20 seconds after the recording. We can use RTP [41] and IP multicast to stream the processed images to all User 3D Live Rendering modules for each frame immediately.

Hopefully, the future version of Magic Land will open a new trend for mixed reality games, in which players can actively play a role of a main character in the game story, be submerged totally in the virtual environment, and explore the virtual world themselves, while at the same time in mixed reality environment, other players can view and construct the virtual scene and new virtual characters to challenge the main character. Consequently, the game story is not fixed but will depend on the players creativity and imagination, and follow their reactions when they travel around the virtual word.

To conclude, HCI has been has been a key issue with a continuous growth in importance and the means by which humans interact with computers will continue to evolve rapidly. Embodied interaction [10] in computing is generally regarded as the next generation computing paradigm. Both Human Pacman and Magic Land have their roots in embodied interaction in the computer gaming arena. We believe that both projects herald the conjuration and growth of a new genre of computer game that is built on mobility, physical actions and the real world as a playground. In a bigger picture, Human Pacman and Magic Land can well symbolize the dawn of such era where real and virtual interactive experience will form part of daily lives routine, allowing users to indulge in the seamless links across different domains be it for entertainment or socializing purpose.

Appendix A

Human Pacman: Wearable Computer Tutorial

A.1 Connections Within the Wearable Computer

The wearable computer is made up of several components. It is essential that these individual modules are connected correctly for the proper functioning of the system.

The components connected to the wearable computer are as seen in Table A.1 below:

As there are only 3 USB ports on the Acer TravelMate 4001LMi, a USB hub is required for the 6 USB connections. Figure A.1 shows the input/output connections from the Acer TravelMate 4001LMi. A close up of the integrated HMD can also be seen on Figure A.2. As most of the components are connected via

Component	**Connection Type**	**Power Source**
LCD Virtual Display GVD300	S-Video Connector	Manufactorer's Battery
Logitech QuickCam Pro 4000	USB2.0	USB2.0
InertiaCube^3	USB2.0.	USB2.0
Bluetooth USB Dongle	USB1.0	USB2.0
Touch Sensor	USB1.0 (via USB mouse)	2 AAA batteries
Twiddle2	USB1.0	USB
Z-Max^{tm} Surveying System *(for outdoor only)*	USB1.0 (via Keyspan:USB Serial Adapter)	Manufacturer's Battery

Table A.1: Components and Connections on Wearable Computer.

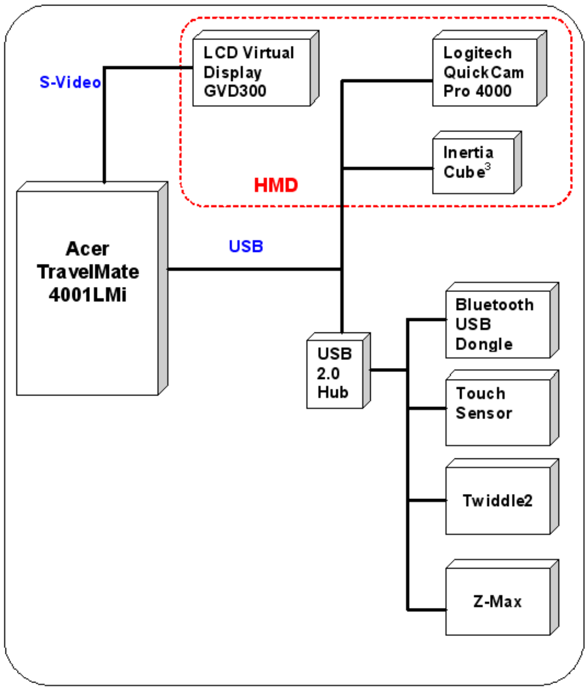


Figure A.1: I/O connections from Acer TravelMate 4001LMi

USB, the entire configuration is simply plug and play. An exception is the Z-Max^{tm} Surveying System. As it is connected to the wearable computer via a Keyspan, we need to run the Keyspan Serial Assistant to manually configure the relevant port.

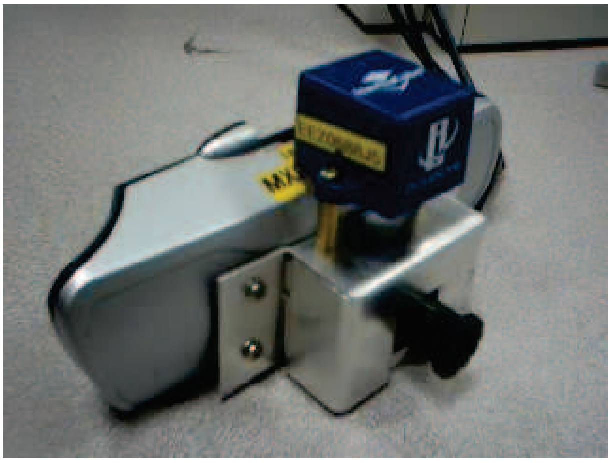


Figure A.2: Integrated Head Mounted Display

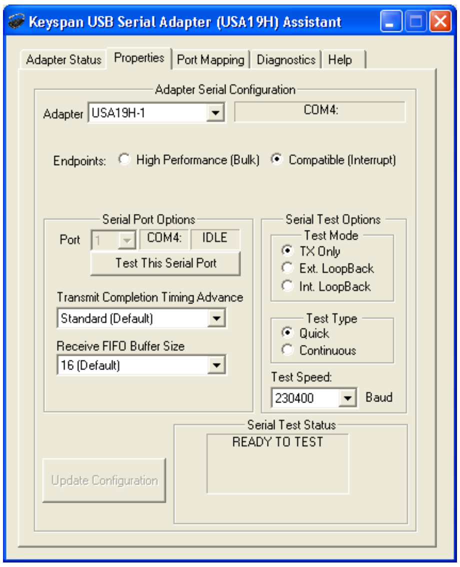


Figure A.3: Keyspan Serial Assistant

The baud rate should be set as 9600. Figure A.3 shows the dialog box of Keyspan Serial Assistant.

Figure A.4 shows the views inside the backpack of the wearable computer. The top image shows Acer laptop that is strapped securely inside the backpack. In the bottom image, we can see all the cables tugged neatly underneath laptop. This design allows the user easy access to the laptop when required.

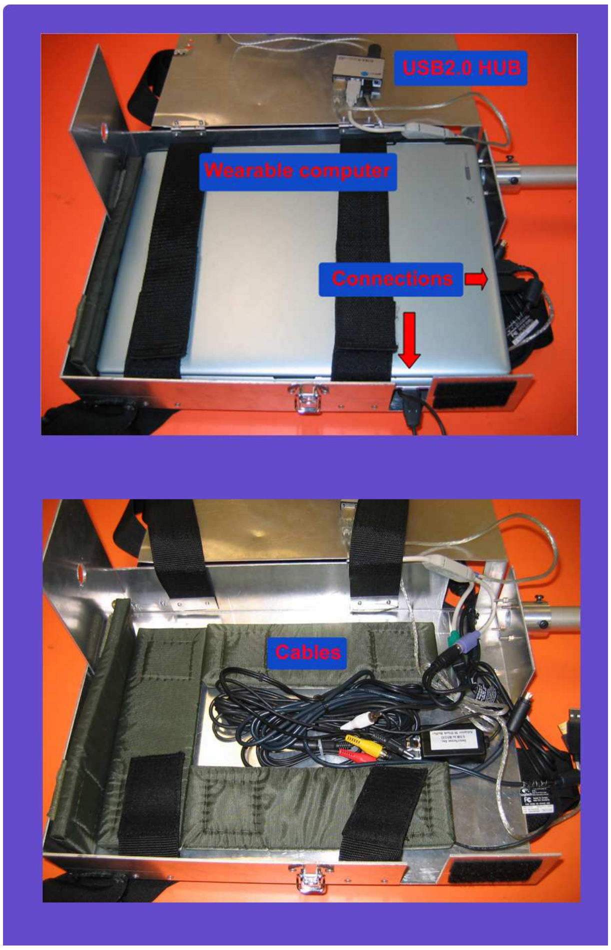
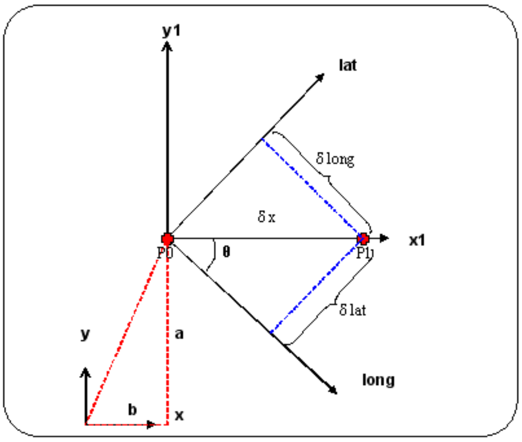


Figure A.4: Views inside the backpack

A.2 GPS Reading Conversion

Apart from performing the above mentioned algorithm, if the game is played outdoor, the wearable computer system also needs to convert the RTK GPS readings from longitude and latitude to the x-y coordinates system that is conventionally used in the game. To perform this, the system needs to be calibrated once before the game starts.

To calibrate, 2 points (P0 and P1) along the x1-axis within the game location is selected and the longitude and latitude reading from the GPS receiver is recorded.



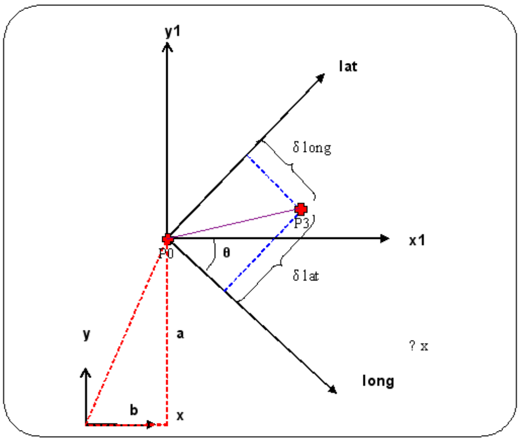


Figure A.5: GPS Reading Conversion

This is shown on left image in Figure A.5. The angle θ can then be calculated using the trigonometry rule:

$$\tan\theta = \frac{\delta lat}{\delta long}$$

Subsequently, any point, in longitude and latitude, (eg. P3 in the right image of Figure A.5) can be converted into x-y coordinates using the formula:

δx = σ(δlong cosθ + δlat sinθ)

δy = σ(-δlong sinθ + δlat cosθ) ,where σ is a scaling factor

σ is a scaling factor to convert degree to a metric measurement. If P0 is not at the origin, the x and y translation from the origin to P0 can be taken into consideration in the final result.

A.3 Starting the system

The sequence of starting the wearable computer system should be as follows.

Figure A.6: Z-Max Surveying System

1. Boot up the Acer TravelMate 4001LMi.
2. Turn on the Z-Max system by pressing the start button in Figure A.6. Wait until more than 5 satellites are in view before proceeding. The number of satellites in view can be seen from the number of green blinks following a red blink at the satellites indicator shown in Figure A.6.
3. Run the WinCom software from Thales Navigation to enable the Z-Max to stream GGA information through serial to wearable computer.
4. Ensure that all connections are properly connected.
5. Calibrate the tracking system (see Section A.2.

6. Run the program with the correct server IP address parameters.

Appendix B

Magic Land: Ceiling Tracker Module Tutorial

There are is a number of things to do in order to properly set up the Ceiling Tracker Module for Magic Land. This includes the proper calibration and positioning of the 4 dragonfly cameras and the installation of the software. These will be discussed in the subsequent sections.

B.1 Camera Positioning

The position of the cameras with respect to the table and one another is important for accurate tracking. Figure B.1 shows the relative position of the cameras from one another. In order to ensure that the cameras position are correctly fixed, a frame was built to mount all the cameras. This frame will be mounted on the

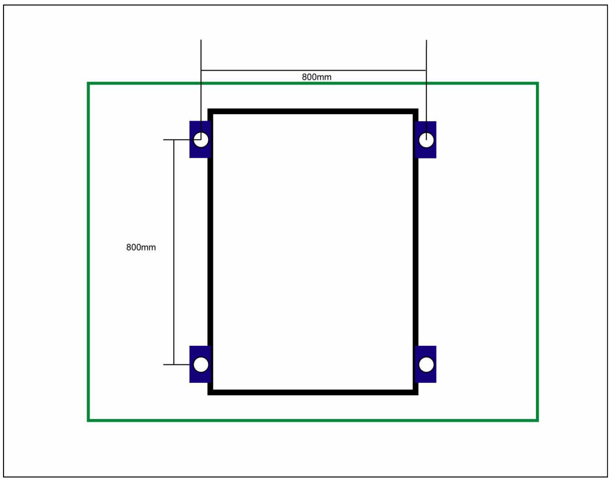


Figure B.1: Camera Layout

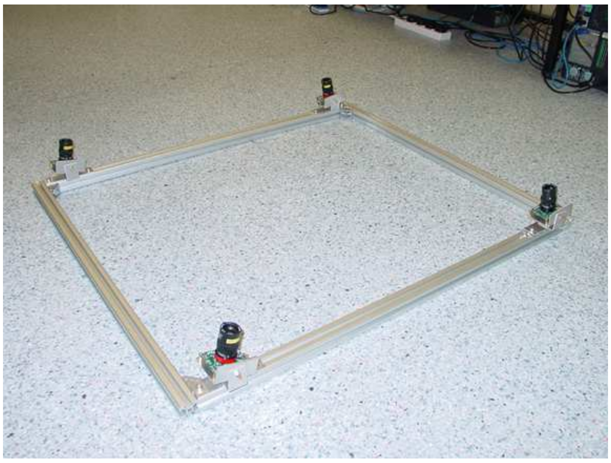


Figure B.2: Cameras on frame

ceiling such that it is directly above the main table. The prototype frame with the dragonfly cameras is shown in Figure B.2. It should also be noted that the cameras should slightly angled inwards (approx 13 degrees) to maximize their area of capture within the table region.

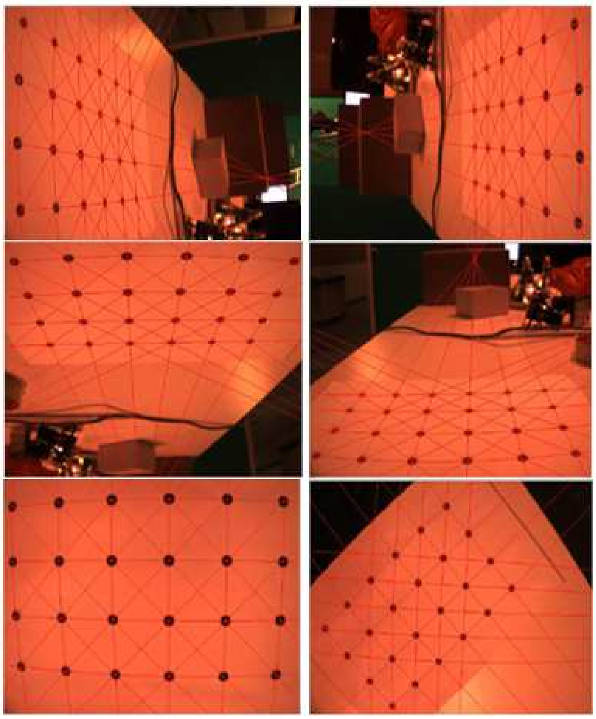


Figure B.3: Images taken for Calibration

B.2 Camera Calibration

For good accurate tracking, each individual cameras should be calibrated for their intrinsic parameters. This can by done using the calib_camera program found in ARToolKit bin folder. For calibration, a special marker needs to be printed. During calibration, the camera is used to capture 5-8 images of the marker from different perspective. The error after calibration will be printed out by the program and its value should ideally be less than 1 mm. Figure B.3 shows an example of the images that was taken for calibration.

B.3 Installation of Software and Starting System

1. Install Redhat Linux 9.0

2. Upgrade the kernel to version 2.4.25 and the ieee1394 driver needs to be upgraded at the same time

3. Install the Nvidia driver for graphic card

4. Install the 1394 libraries

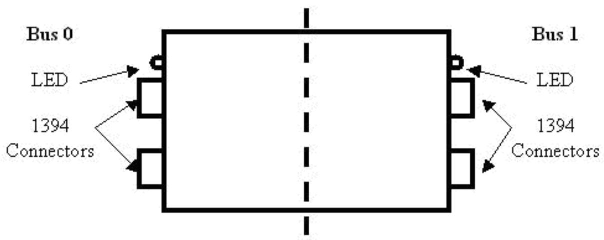


Figure B.4: Syn Unit

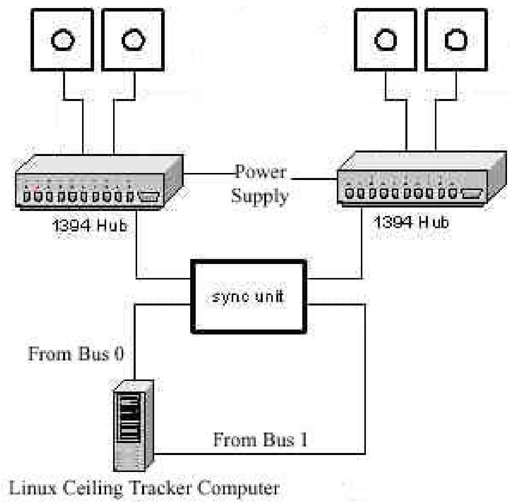


Figure B.5: Cameras Connection

5. Ensure that the 4 dragonfly cameras are properly connected to the LINUX CEILING TRACKER COMPUTER as shown in the Figure B.5 below.

6. Copy the file named "SSC.CeilingTracker.tar.gz" from the source codes folder into the /home directory of the LINUX CEILING TRACKER COMPUTER.

7. Untar the file in the /home directory using the command "tar -zxvf SSC.CeilingTracker.tar.g

8. Change directory to /home/SSC/Tracker/bin using command "cd /home/SSC/Tracker/bin

9. **Run the program using the command "./tracking" or "./MAIN".

10. "Esc" button on the keyboard is used to exit running the program.

** If the executable file does not exist in the /home/SSC/Tracker/bin directory, recompile the codes by following the below instructions:

- Change directory to /home/SSC/Tracker/Tracking.

- Remake all the object files by typing the command "make clean" clean all object files then "make" to recreate all object files and executable files.

- Run the program again at the /home/SSC/Tracker/bin directory.

Appendix C

Demo Write-up

C.1 CHI 2004 - Connect

Date : **27th April(Tuesday) to 29th April 2004(Thursday)**

Venue : **Austria Centre Vienna (ACV) (Vienna, Austria)**

CHI 2004 was held in Vienna, Austria over a period of three days, from 27th to 29th April 2004. The theme for this year's conference is CONNECT. Our lab sent a team of 6 people for the demonstration of our Human Pacman project. We were also invited to host a station in the Games SIG: The Untapped World of Video Gaming during the conference. It took us 8 months to prepare for this big event held for an international audience from the HCI community.

The Game SIG took place on the 27th from 1430hrs to 1600hrs. The SIG is meant for researchers who use games as their research platform. Essentially this was meant to be an informal opportunity to allow other games researchers hands-on access to the game that we worked with. Session attendees would be able to move from station to station during the 90 minute session to play the games and discuss games research with the station hosts. We showed our lab's work on Tilt Pad Pacman, which brought interest from several people. Fig. C.1 shows the demo being done.

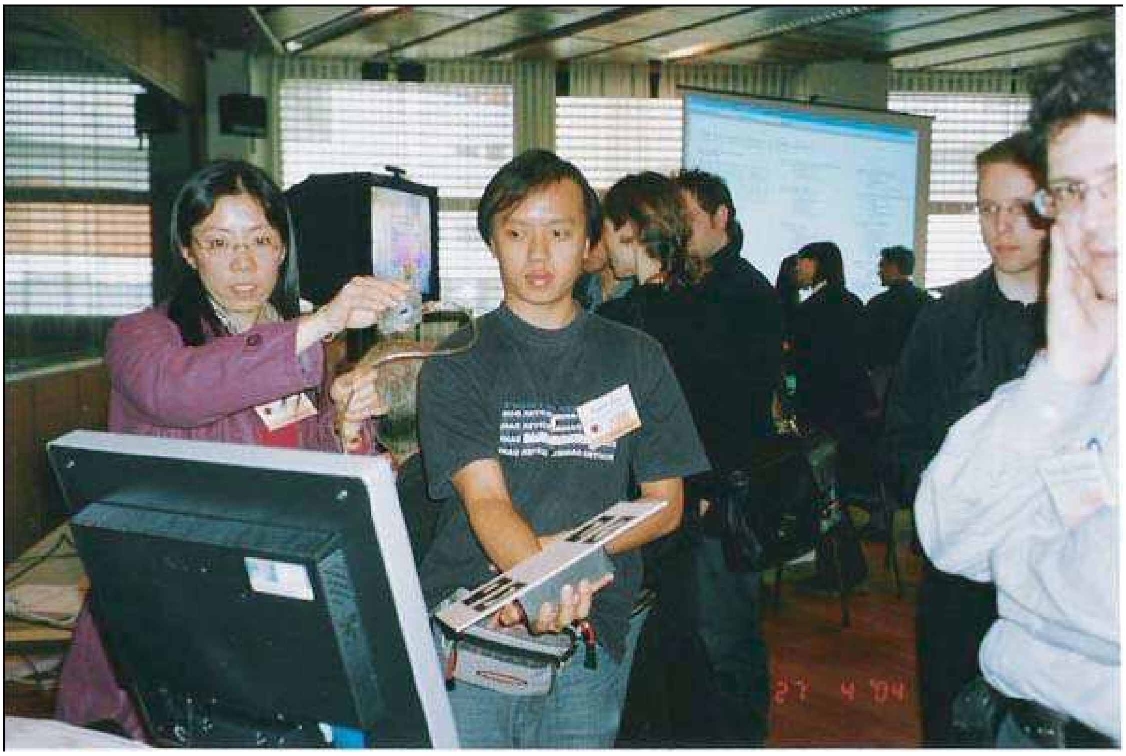


Figure C.1: Tilt-pad Pacman demo during the Game SIG

Our demo, under the session of Games and Virtual Environments, was shown on the 29th between 1130hrs - 1200hrs. Despite having some network glitches which caused one of our wearable computer to crash, a successful demo was still shown when the whole system was up and working finally. We successfully presented our work to the HCI community. During the Q & A, the demo chair stood up, didn't ask a question but just said “Thank you; you have raised the bar of demos in CHI”, bringing on applause from the spectators in the Hall. After the session, several enthusiastic members of the public tried the Human Pacman system, enacting as either the Pacman or the Ghost.

In all, CHI 2004 was a success for Mixed Reality Lab. We look forward in sharing more work with the HCI community in the near future.

Figure C.2: The Human Pacman team after the demo in CHI2004

C.2 ACE 2004

Date : **3rd June (Thursday) to 5th June 2004(Saturday)**

Venue : **National University of Singapore (Singapore)**

ACE 2004 was held in Singapore over a span of three days, from 3rd June to 5th June 2004. The purpose of this conference was to bring together academic and industry researchers, as well as computer entertainment developers and practitioners, to address and advance the research and development issues related to computer entertainment.

Over the course of the conference, researchers from all over the world came together to share their work on computer entertainment. Works range from system that detects anxiety to vary difficult of game, to physical cubes that are put together to form virtual objects. Our lab featured several research works during the poster/demo on the 4th from 1420hrs to 1620hrs. These include “Human Pacman”, “AR Post-It System”, “Jumanji Singapore”, “Magic Story Cube”, and “‘A Step Towards Anywhere Gaming”. Fig. C.3 shows the Human Pacman demo during the demo session. The demo went smoothly for the whole session, with several interested parties enquiring about the game. Local reporters did an interview with us, and also took some materials and photos of the Human Pacman system.

In all, ACE 2004 was a meaningful exchange for the researchers, and a successful avenue for the Mixed Reality Lab to share our work with the international audience.

Figure C.3: Human Pacman demo in ACE 2004

Appendix D

News Articles

39

PLAYING AS PACMAN, Prof Adrian Cheok is chased by researcher Liu Wei, who is acting as one of the Ghosts. Their team took two years to develop the game.

PHOTO: EDWIN KOO

Shawn Woo

HUMAN PACMAN

A Singaporean version of the classic 1980s game has been showcased to 'overwhelming' response at a prestigious IT fair

A SINGAPOREAN upgrade of a classic 1980s arcade game has created such a buzz in the IT world that it was showcased at the recent Wired NextFest 2005.

The prestigious international IT show — held last week in Chicago — showcased the top 100 high-impact and visionary technologies and attracted 30,000 visitors from all over the world.

Human Pacman is the brainchild of Dr Adrian David Cheok, an associate professor in both the School of Engineering and the School of Art Design and Media *at Nanyang Technological University (NTU), and his team from the Mixed Reality Lab.*

The Washington Times called it a "human but virtual version of a classic game".

In the classic game, Pacman has to take all the cookies on a map. It also has to avoid being eaten by Ghosts that are chasing him. If Pacman eats a "super cookie", it can turn the tables by eating the Ghosts.

In Human Pacman, participants play the roles of Pacman and Ghosts. They carry a backpack and wear goggles which allow them to envision themselves in the maze. As they walk along the maze, participants "devour" one another by tapping the other's backpack.

Dr Cheok, 33, came up with the invention as he felt that existing computer games were too static. He also believed that there was room for increased social contact in computer game technology.

Human Pacman, he said, was inspired by a combination of computer game software and childhood games like "catching".

A combination of Bluetooth, GPS, virtual reality and wi-fi technology was used to create Human Pacman. For instance, when the Pacman is within range of the Bluetooth-enabled cookie, communication takes place between the laptop in the backpack and the cookie. The goggles enable the participants to visualise the virtual objects in the game.

The $200,000 game took Dr Cheok's team more than two years to realise.

Besides being the only entry from Singapore at NextFest, the Mixed Reality Lab team also showcased another game called Magicland. The game allows participants to enter a virtual reality world with dragons and other creatures.

Meanwhile, a jubilant Dr Cheok who returned from Chicago earlier this week, described response to Human Pacman as "overwhelming".

He told The Sunday Times: "Some queued for more than half an hour to try the game, and we had to skip lunch to cope with the demand."

He feels that the game's non-violent nature appeals to people of all ages.

Making the game, however, was not plain sailing. The version on display in Chicago was the fourth one by the team.

Dr Cheok, who has a Chinese father and a Greek mother, said: "The biggest challenge we faced was building up a team of gifted and creative individuals."

His team include undergraduate, masters and PhD students, as well as researchers employed from funding research grants.

There were times when he felt like giving up: "Preparation for exhibitions like this are the most taxing. Last year in Vienna, we went three days without sleep."

Despite these problems, he keeps going. Budding creators, he said, "need creativity and passion, but most importantly, you need perseverance".

Unfortunately, it is not likely that the public can lay their hands on this fascinating game.

At $7,000 per unit, he said it is not cost effective to market it. Instead, the software used in creating the game will be utilised for research purposes, such as possible inclusion in 3G-enabled cellular phones.

However, in your spare time, take a trip down to NTU's Interaction and Entertainment Centre, where you can have a shot at grabbing all the cookies before the Ghosts catch you.

Figure D.1: Human Pacman article in Sunday Times(3rd July 2005). © MediaCorp Press Ltd.

后港小学将设混合实境教室

戴上特制眼镜上课 学生能"触摸动植物"

科技天地

传真: 63198121

张嘉玲●报道

由本地科研人员研发的混合实境（mixed reality）科技，不久后将应用在多个领域。除创造全球首个混合实境教室，这项科技今后也可用来设计适合年长者使用的互动游戏。

由南洋理工大学互动与娱乐研究中心（Interaction and Entertainment Centre）主任卓龙星副教授领导的小组所研发的混合实境科技，结合虚拟和现实画面，开拓一种全新的视觉体验。

科研人员应用这项新科技设计新的互动游戏，让人们走入电脑游戏中，化身为游戏角色。

游戏者只要戴上特制的眼镜和装有环球定位系统（GPS）的接收器，周围的环境就会转换成游戏环境。

眼镜镜片就类似手机荧幕，眼镜上端配有网络摄像机，可将外在环境拍摄下来，结合游戏指示呈现荧幕上。

玩的人就变成真的游戏人物，穿梭在游戏环境里，根据外在指挥者的指示行走，并和其他游戏人物互动。

卓龙星（33岁）受访时说，他们正与教育部合作，计划在后港小学成立全球首个"混合实境教室"，学生上科学课时戴上特制眼镜，伸手就能触摸各种植物或动物。由教育部赞助的计划预计可在今年11月落实。

他说："使用这种教学法，教材不再抽象，学生可真正'触摸'到眼前的事物，课堂上的互动性也大大提高。"

把游戏带进社区 让年长者尝试

除此之外，为让年长者更积极参与社区活动，卓龙星也正研发适合年长者的互动游戏，玩游戏者在特定范围内穿上特制鞋子，就能和小孩一起玩射击游戏。

卓龙星说，他们打算在年底把游戏带进社区，让年长者尝试。除让他们有机会活动筋骨外，也能让他们保持头脑灵活。

他说："随着科技发展，人与人的交流减少了，混合实境就是要扭转这样的趋势，虽然我们运用先进科技，但与人的交流却是不可缺少的一部分。"

混合实境科技也获得国际肯定。上月，卓龙星的小组所研发的"吃豆人"（Human PacMan）和"魔术乐园"（Magic Land）游戏，受邀在美国芝加哥"Wired Nextfest"展览会上展出。

这个国际权威展览会，每年选出全球100个顶尖发明品展出，卓龙星两个游戏都获得评审青睐，在展览会上引起各界兴趣。

卓龙星说，混合实境科技引起广泛注意，说明新加坡也有能力研发前卫的科技，领先其他发展国家。

他认为，创……的崛起，说明了……工程科技已不能……代社会的需求，……朝结合工程与艺……值领域发展。

他说："未来……讯科技人才不能只……才，他必须懂得一……计、懂得一点工……识、懂得一点艺术……重要的是要懂得把……结合起来，应用在……领域上。"

南洋理工大学互动与娱乐研究中心主任卓龙星（左）带领的小组所研发的混合实境科技，结合虚拟和现实画面，开拓一……的视觉体验。

联邦通道组屋

妇女参政的阻力：能力仍受到质疑

Figure D.2: Human Pacman article in Lian He Zao Bao(11th July 2005). © MediaCorp Press Ltd.


TODAY

Human Pacman on Orchard Road

The game uses Head Mounted Display to allow physical interaction and free body movement

Figure D.3: Human Pacman article in TODAY (23rd June 2004). © MediaCorp Press Ltd.

Here!

联合早报

结合虚拟与现实画面
让你走入电脑游戏中

Figure D.4: Human Pacman article in Lian He Zao Bao (14th June 2004). © Singapore Press Holdings.

PACMAN ALIVE!

Figure D.5: Human Pacman article in The New Paper (9th August 2004). © Singapore Press Holdings.

Bibliography

[1] Z. Zhang and Y. Shan. State of the industry report. Report 2000-2001, Entertainment Software Association, 2001.

[2] Biocca, F., Burgoon, J., Harms, C. and Stoner, M. Criteria and scope conditions for a theory and measure of social presence. In *4th Annual Inter Workshop Presence*, 2001.

[3] J., Bowlby. *Attachment and Loss, Volume I: Attachment.* Basic Books, New York, USA, 1983.

[4] Entertainment Software Association. E3expo 2005 state of the industry address. [Online], May 2005. Available at http://www.theesa.com/archives/2005/05/.

[5] Konami. Dance dance revolution. [Online]. Available at http://www.konami.co.jp/am/ddr/.

[6] Konami. Para para paradise. [Online]. Available at http://www.konami.co.jp/am/para/.

[7] Sony. Eyetoy. [Online]. Available at http://www.eyetoy.com/english/index.html.

[8] Inc XEODesign. Why we play game. [Online]. Available at http://www.xeodesign.com/whyweplaygames.html.

[9] C. Crawford. Live: What a concept! ... networked games. *Digital Illusion*, pages 241 – 248, 1998.

[10] Dourish. P. *Where The Action Is: The Foundations of Embodied Interaction.* MIT Press, Cambridge, Massachusetts, 2001.

[11] M. Weiser. The computer for the twenty-first century. 265.

[12] H. Ishii, B. Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI '97), ACM*, pages 234–241, 1997.

[13] Suchman L.

[14] Paul Milgram, Fumio Kishino. A taxonomy of mixed reality visual displays. E77-D.

[15] Jacques Cohen. Computer augmented environments: Back to the real world. *Communications of the ACM*, 36(7), July 1993.

[16] Smith, C.G. The hand that rocks the cradle. *I.D.*, pages 60–65, May/June 1995.

[17] Fitzmaurice, G., et al. Bricks: Laying the foundations for graspable user interfaces. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI '95), ACM*, pages 442–449, Denver, May 2001.

[18] Starner, T., Leibe, B., Singletary, B. and Pair, J. Mind-warping:towards creating a compelling collaborative augmented reality game. In *Proc. Intelligent User Interfaces*, pages 256–259, 2000.

[19] T. Oshima, K. Satoh, H. Yamamoto, and H. Tamura. Ar2 hockey system: A collaboration mixed reality system. 3(2):55–60, 1998.

[20] H. Tamura, H. Yamamoto, and A. Katayama. Mixed reality: Future dreams seen at the border between real and virtual worlds. 21(6):64–70, 2001.

[21] A. D. Cheok, X. Yang, Z. Zhou, M. Billinghurst, and H. Kato. Touch-space: Mixed reality game space based on ubiquitous, tangible, and social computing. *Journal of Personal and Ubiquitous Computing*, pages 430–442, 2002.

[22] Björk, S., Holopainen, J., Ljungstrand, P. and Mandryk, R. Pirates! using the physical world as a game board. In *Proc.Interact 2001*, 2001.

[23] Z-max surveying system. [Online]. Available at: http://www.cadden.fr/fr/pdf/zmax.pdf.

[24] Vishay Telefunken. *TSUS540.* 1999.

[25] MicroChip. *PIC16F7X Datasheet.* 2002.

[26] TIQIT Computers. *Matchbox PC Manual.* Oct 2000.

[27] Pascal Lutz and Ren Gounon. Thales navigation's lrk, an improved rtk solution for long baselines: technique, applications and field results.

[28] S. J. D. Prince, A. D. Cheok, F. Farbiz, T. Williamson, N. Johnson, M. Billinghurst, and H. Kato. 3d live: real time captured content for mixed reality. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, 2002.

[29] T. Horprasert et al. A statistical approach for robust background subtraction and shadow detection. In *Proc. IEEE ICCV'99 Frame Rate Workshop, Greece*, 1999.

[30] M. Seki, H. Fujiwara, and K. Sumi. A robust background subtraction method for changing background. In *Proceedings of the Fifth IEEE International Workshop on Applications of Computer Vision*, 2000.

[31] R. Mester, T. Aach, and L. Dmbgen. Illumination-invariant change detection using statistical colinearity criterion. In *DAGM2001, number 2191 in LNCS. Springer*, pages 170–177.

[32] S. J. D. Prince, A. D. Cheok, F. Farbiz, T. Williamson, N. Johnson, M. Billinghurst, and H. Kato. Live 3-dimensional content for augmented reality. In *IEEE Transactions on Multimedia (submitted).*

[33] RGB “Bayer” Color and MicroLenses. [Online].
http://www.siliconimaging.com/RGB Bayer.htm.

[34] G. Slabaugh, R. Schafer, and M. Hans. Image-based photo hulls. In *Proceedings of the 1st International Symposium on 3D Data Processing, Visualization, and Transmission*, 2002.

[35] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. In *International Journal of Computer Vision, Vol.38, No. 3*, 2000.

[36] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 1997.

[37] G. Slabaugh, W. B. Culbertson, T. Malzbender, and R. Schafer. A survey of volumetric scene reconstruction methods from photographs. In *Proc. International Workshop on Volume Graphics*, 2001.

[38] G. G. Slabaugh, W. B. Culbertson, T. Malzbender, M. R. Stevens, and R. W. Schafer. Methods for volumetric reconstruction of visual scenes. In *The International Journal of Computer Vision*, 2004.

[39] R. Szeliski. Video mosaics for virtual environments. In *IEEE Computer Graphics and Applications*, March 1996.

[40] D. Myers. *Computer game semiotics, Play and Culture*, 4:334–345, 1991.

[41] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. Rtp: A transport protocol for real-time applications. *Internet Engineering Task Force, Audio-Video Transport Working Group*, 1996.