ARQuake: An Outdoor/Indoor Augmented Reality First Person Application

Bruce Thomas, Ben Close, John Donoghue, John Squires,
Phillip De Bondi, Michael Morris and Wayne Piekarski
School of Computer and Information Science
University of South Australia
Bruce.Thomas@UniSA.Edu.Au

Abstract

This paper presents an outdoor/indoor augmented reality first person application ARQuake we have developed. ARQuake is an extension of the desktop game Quake, and as such we are investigating how to convert a desktop first person application into an outdoor/indoor mobile augmented reality application. We present an architecture for a low cost, moderately accurate six degrees of freedom tracking system based on GPS, digital compass, and fiducial vision-based tracking. Usability issues such as monster selection, colour, and input devies are discussed. A second application for AR architectural design visualisation is presented.

1 Introduction

Many current applications place the user in a first-person perspective view of a virtual world [6], such as games, architectural design viewers [2], geographic information systems and medical applications [12]. In this paper we describe a project to move these forms of applications outdoors, displaying their relevant information by augmenting reality. In particular we consider the game Quake [4] and the viewing of architectural designs [13]. As with other researchers [3], we wish to place these applications in a spatial context with the physical world, which we achieve by employing our wearable computer system Tinmith-4 [9, 10]. Tinmith-4 is a context-aware wearable computer system, allowing applications to sense the position of the user's body and the orientation of the user's head. The technique we are developing will genuinely take computers out of the laboratory and into the field, with geographically-aware applications designed to interact with users *in the physical world*, not just in the confines of the computer's artificial reality. The key to this exciting practical technology is *augmented reality* (AR). Users wear see-through head-mounted displays through which they see not only the world around them, but also overlaid computer-generated information that enriches the user's perception. Unlike virtual reality, where the computer generates the entire user environment, augmented reality places the computer in a relatively unobtrusive, assistive role.

In the ARQuake application, the physical world is modelled as a Quake 3D graphical model. The augmented reality information (monsters, weapons, objects of interest) is displayed in spatial context with the physical world. The Quake model of the physical world (walls, ceiling, floors) is not shown to the user: the see-through display allows the user to see the actual wall, ceilings and floors which ARQuake need only model internally. Coincidence of the actual structures and virtual structures is key to the investigation; the AR application models the existing physical outdoor structures, and so omission of their rendered image from the display becomes in effect one of our rendering techniques.

1.1 Aims

Our *aim* is to construct first-person perspective applications with the following attributes: 1) The applications are situated in the physical world. 2) The point of view which the application shows to the user is completely determined by the position and orientation of the user's head. 3) Relevant information is displayed as augmented reality via a head-mounted see-through display. 4) The user is mobile and able to walk through the information space. 5) The application is operational in both outdoor and indoor environments. 6) The user interface additionally requires only a simple hand-held button device.

1.2 Research issues

To achieve these aims, we investigate a number of research issues in the areas of user interfaces, tracking, and conversion of existing desktop applications to AR environments.

User interfaces for augmented reality applications which simultaneously display both the physical world and computer generated images require special care. The choice of screen colours for the purely virtual images which the application must display requires attention to the lighting conditions and background colours of the outdoors. The keyboard and mouse interactions

In ISWC2000 - 4th International Symposium on Wearable Computers
October, 2000 - Atlanta, Ga, USA - Copyright (C) 2000 IEEE
Please visit http://www.tinmith.net for more information

Proceedings of the Fourth International Symposium on Wearable Computers (ISWC'00)
0-7695-0795-6/00 $10.00 @ 2000 IEEE

must be replaced with head/body movement and simple buttons. The layout of the user interface must accommodate the AR nature of the application.

The six degrees of freedom (6DOF) tracking requirements for these forms of applications must be addressed. We require a low cost, moderately accurate 6DOF tracking system. Tracking is required for indoor and outdoor environments over large areas, for example our usual testing environment is our campus [7]. GPS positional error has a less noticeable effect for the registration of augmented reality information at distance, but we need to address positional error when registering augmented information at close distances ($< 50\ m$). Such a tracking system could be used for other applications, such as tourism information, visualisation of GIS information, and as described in this paper architectural visualisation.

It is also necessary to modify the Quake game to accommodate the AR nature of the new application. The user's movement changes from a keystroke-based relative movement mode to a tracking-based absolute mode. The game's coordinate system must be calibrated to the physical world. Finally, the field of view of the display must be calibrated to the physical world.

1.3 Progress to date

This is an ongoing project, and it is not complete at the time this paper was submitted. We wish to clarify the status of the different portions of system presented in this paper. The ARQuake game has been ported to our our wearable computer platform and operates with GPS and digital compass as a means of tracking. The keyboard and mouse interaction with the game has been completely replaced with user movements and a two-button input device, and is fully functional within the accuracy of this tracking system. We have modelled in our Quake world an outdoor section of our campus and the interior of our Wearable Computer Laboratory (WCL). The graphics of the game runs at 30 frames per second, but GPS updates once per second and the compass updates at 15 times per second. The colours of the graphics in the game have been optimised for the user with a see-through display in an outdoor environment.

The major hurdle left is the vision based tracking system. We have the vision based tracking system exporting a common coordinate system as to the GPS/compass system. There are issues of accuracy and speed which are currently being investigating, which prevents us from stating this portion of system is functional. The paper presents the current state we have achieved in this regard.

2 Background

There are key technologies we are employing in our investigations. A brief review of tracking as applied to this project, the Quake game, and our wearable computer platform are supplied.

2.1 Tracking

Previous research has established that outdoor tracking with inexpensive differential GPS and commercial grade magnetic compasses are inaccurate for augmented reality applications [1]. Traditional hybrid approaches combine a number of different systems such as inertial, optical, electro-magnetic and GPS. We combine vision-based optical tracking with GPS and a magnetic compass.

A number of researchers are investigating *fiducial* vision-based tracking [8, 11]. We based our optical tracking system on the fiducial marker tracking system ARToolKit developed by Kato and Billinghurst [5]. The ARToolKit is a set of computer vision tracking libraries that can be used to calculate camera position and orientation relative to physical markers in real time. ARToolKit features include the use of a single camera for position/orientation tracking, fiducial tracking from simple black squares, pattern matching software that allows any marker patterns to be used, calibration code for video and optical see-through applications, and sufficiently fast performance for real-time augmented reality applications.

The fiducial markers are known-sized squares with high contrast patterns in their centres. Figure 4 shows an example marker. The ARToolKit determines the relative distances and orientation of the marker from the camera. In addition, the ARToolKit incorporates a calibration application to determine the placement of the camera relative to the user's line of sight; thus the ARToolKit can determine proper placement of graphical objects for AR applications.

2.2 The original Quake game

We chose Quake as the primary application for a number of reasons. Quake fits the general model of AR which we are studying, as it is a first-person 3D application with autonomous agents to interact with the user. The application itself is public domain, with open source code. Finally, the Quake graphics engine is very quick and runs on a wide range of computing platforms and operating systems.

Quake is a first-person *shoot 'em up* game. Quake has two stated goals: "First, stay alive. Second, get out of the place you're in" [4]. The user interface is based around a single, first-person perspective screen. The large top part of the screen is the view area, showing monsters and architecture. Status information is immediately below at the bottom of the screen.

One moves around Quake in one of four modes: walking, running, jumping or swimming, and performs one of three actions: shooting a weapon, using an object, or picking up an object. Weapons are aimed by changing the view direction of the user, and fired by pressing a key. To push a button or open a door, the user walks up to the door or button. A user picks up items by walking over them. Part of the challenge of the game is finding special objects like buttons, floor-plates doors, secret doors, platforms, pressure plates and motion detectors. Quake incorporates platforms that move up and down, or follow tracks around rooms or levels. Pressure plates and motion detectors may be invisible or visible, and there are sensors which open doors, unleash traps, or warn monsters.

2.3 Wearable computer platform

The Tinmith-4 wearable computer system hardware is all mounted on a rigid backpack so that the items can be attached firmly. Processing is performed by a Toshiba 320CDS notebook (Pentium-233, 64 Mb RAM) running the freely available LinuxOS and associated programs and development tools. The laptop is very generic, and not even the latest in available CPUs, so another computing unit could be substituted. The limited I/O capabilities of the single serial port are augmented with the use of a four serial port Quatech QSP-100 communications card. Connected to the laptop are a Precision Navigation TCM2-80 digital compass for orientation information, a Garmin 12XL GPS receiver for positioning, and a DGPS receiver for improved accuracy. For the head mounted display (HMD), we use alternately the i-Glasses unit from I-O Display Systems, and the Sony Glasstron PLM-S700E. Various other devices are present as well, such as a small forearm keyboard for data entry, power converters for the different components, and necessary connection cabling and adaptors. The construction of the backpack was directed with ease of modifications in mind, at the sacrifice of wearability and miniaturisation.

The Tinmith system [10] supports outdoor augmented reality research. The system is comprised of a number of interacting libraries and modules. Firstly, a number of software libraries form a support base for writing code in the system: libGfx - a graphics interface on top of X windows; libConvert - coordinate and datum transformations, numeric conversions; libProtocol - encode/decode libraries for transmitting structures over a network; libSystem - network communications and high level I/O; libCodeBase - low level interfaces to Unix system calls, asynchronous I/O code, string handling, event generation, error checking. These libraries are used in turn to implement software modules (implemented as individual Unix processes) that perform the actual tasks in the system. These software modules process input from hardware devices and other modules, and then pass this output on to other modules which are interested in these values. The communication between modules is performed using TCP/IP, and allows the system to be distributed over a network of wearable processors, and also for other machines to collaborate and share information.

The original Tinmith-1 system was an outdoor augmented reality navigation system, supporting 2D top-down maps, and a 3D immersive wire frame renderer [10]. This system was then extended as Tinmith-2 to share information with standard VR entity protocols [9]. Simulated and real entities appear on the HMD, and outdoor wearable machines appear back on the indoor displays.

3 Using ARQuake

The goal of ARQuake was to bring the intuitive nature of VR/AR interfaces into an indoor/outdoor game. A user first dons the wearable computer on their back, places the HMD on their head, and holds a simple two-button input device. The user then performs a simple calibration exercise to align the HMD with their eyes, and then they start playing the game. All of the keyboard and mouse controls have been replaced with position/orientation information and a two-button input device. As movement aspects of the game have been engineered to fit the physical world, there is no concept of commands to walk, run, jump, swim, or of moving platforms. The user's own movement determines the rate and direction of movement. The remainder of this section describes the Quake level we developed and its user interaction.

3.1 Monsters

There are sixteen different types of monster in the Quake world. Some have attributes that make them unsuitable for inclusion in this type of level. Because of the limitations on movement imposed by the tracking hardware, the best monsters were those that walked or leaped and those that were relatively easy to destroy and did not inflict extreme damage on the user with their first attack.

We chose seven types of monsters to be included in this level. These monsters types are all land-based creatures which use weapons from a distance, and all seem well suited to the system. The monsters' *skin* colour and texture were changed to make them easier to see and distinguish from the physical world. The choice of colours used in the texture maps or skins of the monsters are based on the user testing described later in Section 5.

3.2 Campus level

We created a Quake level representing a portion of the Levels campus of the University of South Australia. The walls in Quake are the walls of the external and interior of the WCL. The walls are rendered in two fashions, black for game mode and a grid patterned for testing mode. In both these modes, the walls occlude the graphic objects in Quake that may be located behind the walls. As described earlier, in the game mode black walls are invisible to the users during the game. The Quake graphics engine renders only monsters, items on the ground, and regions of interest. This Quake level was derived from architectural drawings of the campus provided by the university; where the architect's drawings had become incorrect, we surveyed those portions ourselves.

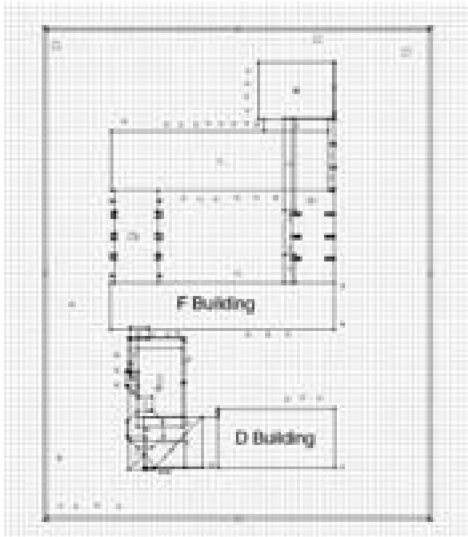

Figure 1: Quake campus level

Figure 2: User's Heads Up Display

The size of the outside modelled area is 94 metres (East/West) by 156 metres (North/South). Figure 1 depicts a top-down view of the level we created. We have placed 16 monsters in the levels as follows: two enforcers on top of D Building, two ogres on the 2nd floor of F building, and the rest spread around the ground level. There are 51 items placed on the ground for the user to pick up: six pieces of armour, 22 rockets, four rocket launchers, nine shotgun shells, and ten health boxes.

The system of tracking used in this system tends to make the user less agile than the "super-human" agility found in the normal game. Therefore we have included more support equipment than would be found in the normal game, armour, weapons and ammunition.

3.3 Walking around

Once the system is up and running, the user moves through the level by walking, and changes view by looking around. The user views the game and the physical world through the HMD, an example is shown in Figure 2. The bottom portion of the screen is a status bar containing information about armor, health, ammo and weapon type. The majority of the screen is reserved for the AR images of monsters and game objects.

In the original Quake, certain actions are performed by the user being in a certain proximity to a location in a Quake level. We have retained most of those actions. Doors open when the user attempts to walk through them. Users pick up objects as in the original Quake by walking over them. Traps are triggered by standing in or moving through predetermined locations. Actions which are not easily reflected in the physical world are removed from the game, such as secret and locked doors.

The tracking of the user's position and orientation of the user's head handles the majority of the interaction for the user. The only other interactions for the user to perform are to shoot or change the current weapon. We employ a two-button (thumb button and index finger button) hand-held device as a physical input device for these actions. The thumb button is used to change weapons, and the index finger button fires the current weapon. The direction the weapon fires is the center of the current view of the HMD.

3.4 Field of view

Even if alignment of the Quake world with the physical world is exact, an incorrect perspective or field of view will be highlighted as inconsistencies in the virtual world. The default field of view for the game is 90 degrees (45 degrees each side), allowing a reasonable coverage of the world to fit onto a computer screen. This field of view unfortunately suffers from the *fish eye* distortion effect when comparing the objects in the Quake world with real objects. The HMD we are using, I-Glasses, has approximately a 25 degree horizontal field of view. The only calibration adjustment for the HMD with Quake is changing the game's field of view setting and scaling of the graphical objects. We are currently using a field of view value of 25 degrees, but there are artifacts introduced as in the user is positioned farther forward. We are investigating the graphics model of Quake to determine how it differs from traditional graphics models.

4 Tracking

As previously stated, one of the goals of the system is to provide continuous indoor and outdoor tracking. The system tracks through the combination of a GPS/compass system with a vision-based system. Our tracking needs are categorized into three areas as follows: outdoors far from buildings, outdoors near buildings, and indoors. Each of these require a different approach, while maintaining position and orientation information in a common format of WGS 84/UTM positioning information and heading/pitch/roll angles for orientation information. The use of visual landmarks can improve registration in one of two ways, firstly to allow the system to correct the final image by aligning the landmark with a known position in the graphical image, and secondly to use the landmarks to extract a relative position and orientation of the camera from the landmarks. We have chosen the second option to investigate as it provides the most general tracking solution.

4.1 Outdoors away from buildings

GPS positional inaccuracies are less of a problem for our Quake application when a user is at a large distance ($> 50\ m$) from an object which requires registration, while orientation errors remain constant as to angular deviations in the user's field of view. An extreme example of how positional errors have a reduced registration error effect at distance is the using of the ARQuake game on a flat open field, where the system does not require graphics to be registered to any physical object except the ground. In this scenario there are no walls to occlude the monsters and items of interest. Since the game is slaved to the screen, what the user sees on the display is what the game believes is the user's current view. Therefore the user's actions will perform correctly in the context of the game.

In the case where a building is visible but the user is a large distance from the building, the inaccuracies are low and therefore not distracting. The problems come when monsters are not occluded properly by the physical buildings. The visual effect of poor occlusion is that monsters appear to walk through walls or pop out of thin air, but at distance these errors do not detract from the game. Such occlusion problems exist but they are visually very minor, because the user is generally moving their head during the operation of the game.. At 50 meters a difference of 2-5 metres (GPS tracking error) of the user's position is approximately a 2-5 degree error in user's horizontal field of view, and the compass itself has an error of +/- 1 degrees.

4.2 Outdoors near buildings

When using ARQuake with the GPS/compass tracking less then 50 metres from a building, the poor occlusion of monsters and objects near the physical buildings, due to GPS error, becomes more apparent. As the user moves closer to buildings, inaccuracies in GPS positional information become prevalent. The system is now required to slave the Quake world to the real world, and furthermore in real time. As an example when a user is ten metres from a building their position is out by 2-5 metres, this equates to an error of 11-27 degrees; this is approximately a half to the full size of the horizontal field of view of the HMD. When the error is greater than the horizontal field of view, the virtual object is not visible on the HMD.

Figure 3: Fiducial marker on a building

To enhance the accuracy when the user is near buildings we use an extended version of ARToolKit. By using fiducial markers specifically engineered for outdoor clarity (approximately 1 metre in size), and with each marker setup on a real world object with known coordinates, accurate location information can be obtained. Figure 3 shows a fiducial marker on the corner of a building in our Quake world. These markers provide a correction in the alignment of the two worlds. We are investigating the use of multiple fiducial markers to reduce uncertainty due to marker mis-detection caused by lighting issues. Since the extended ARToolKit we are developing supplies positioning and orientation information in the same format as the GPS/compass system, ARQuake can transparently use either the GPS/compass or vision-based tracking systems. Our initial approach for determining when to use the information from the GPS/compass or the ARToolKit methods is use the ARToolKit's information first, when the ARToolKit is confident of registering a fiducial marker. As ARToolKit recognises a fiducial marker, the toolkit returns a confidence value, and the system will have a threshold of when to switch over to use the the toolkit. When the confidence value goes below the the threshold, the GPS/compass information is used.

4.3 Indoors

As a user walks in to a building with fiducial markers on the inside walls and/or ceilings, the tracking system starts using the vision-based component of the tracking system. This form of tracking is similar to the work of Ward, et al. [14]. Our system is lower-cost and is not as accurate, but does keep tracking errors within the accuracy which our application needs, 2-5 degree of error in user's horizontal field of view. We are experimenting with placing markers on the walls and/or the ceilings. Figure 4 shows one configuration of how we are using wall mounted markers.

When the markers are placed on the wall, we point the vision-based tracking camera forwards. It was necessary to size and position the patterns on the walls so

Proceedings of the Fourth International Symposium on Wearable Computers (ISWC'00)
0-7695-0795-6/00 $10.00 @ 2000 IEEE

Figure 4: Fiducial marker on the wall

that they would be usable by the system regardless of whether the user was very close or very far from the wall. In this case, we chose to use patterns that were a size of 19 cm^2. From testing, we found that the system could register a pattern at a range of 22.5 *cm* to 385 *cm* from the wall. In a 8 *x* 7 *m* room this range would be sufficient (for the initial stages of the project) and an accuracy of within 10 cm at the longer distances. It is important that no matter where the user looks in the room that, at least one pattern must be visible in order to provide tracking. For this reason, we realised that to implement the patterns on the walls as the sole means of tracking would require different size targets. We are investigating the use of targets themselves as patterns inside larger targets; therefore one large target may contain four smaller targets when a user is close to a target.

Our second approach has been to place the markers on the ceiling, with the vision-based tracking camera pointed upwards. The camera does not have the problem of variable area of visible wall space, as the distance to the ceiling is relatively constant. The main differences are the varying heights of users. In the first instance we are implementing a small range of head tilt and head roll (+/- 45 degrees). Perspectives such as those from laying down or crawling will be investigated in the future.

The patterns on the ceiling were placed so that at one time at least one pattern could be reliably identified by the tracking software. With the camera mounted on the backpack at at height of 170 *cm* and with the room height of 270 *cm*, our current lens for the camera views a boundary of at least 130 cm^2; we chose a pattern of 10 cm^2 in size.

5 Choosing colours

The choice of colours is important for outdoor augmented reality applications, as some colours are difficult to distinguish from natural surroundings or in bright sunlight. The original Quake game incorporates a "dark and gloomy" colour scheme to give the game a foreboding feelings. Dark colours appear translucent with the see-through HMDs. Monsters and items need different colours to be more visible in an outdoor environment. We ran a small informal experiment to determine a starting point in picking colours for use in an outdoor setting. This informal study was to gauge the visibility and opaqueness of solid filled polygons display of the see-through HMD. We are interested in which colours to texture large areas of the monsters and items in the game. These colours are not necessarily appropriate for textual or wire-frame information. Further studies are required for these and other forms of AR information.

The testing method was to view different colours in one of four conditions: 1) standing in shade and looking into the shady area, 2) standing in shade and looking into a sunny area, 3) standing in a sunny area and looking at a shady area, and 4) standing in a sunny area and looking to a sunny area. We tested 36 different colour and intensity combinations, nine different colours (green, yellow, red, blue, purple, pink, magenta, orange, and cyan) and four different intensities. The combination is indicated as the name of colour followed by intensity in parenthesis, for example green(2), with (1) the highest and (4) the lowest intensity. The testing was performed outside with the Tinmith-3 wearable computer using the I-Glasses see-through HMD. The colour/intensity combinations were scored for visibility and opaqueness in each of the four viewing conditions on a scale of one (very poor) to ten (very good)).

Our strongest set of criteria for colour/intensities were both a mean score of at least seven over the four viewing conditions, and as well a minimum score of six on each of the conditions. Nine colours satisfy this quality level: purple(2), purple(3), blue(2), blue(3), yellow(1), purple(1), yellow(3), green(1) and green(3). Should a particular application require a larger palette, weaker criteria of a mean score six and above with no score below five gives seven additional colours: blue(1), pink(2), yellow(2), green(2), pink(1), blue(4) and red(1).

6 Informal User Study of ARQuake

To gauge how well the ARQuake game appeals to user's, we performed a simple informal user study of people using the system outside. The user's were prompted to how they felt about aspects of the system. Many subjects thought that the visibility of the ARQuake system was good; however many of the subjects found that bright lighting made seeing through the display difficult. Despite only using the system once, the users found the hand held input device intuitive, easy to use, and very quick to learn. A few of the users found themselves pointing the device in a gun like fashion when firing at the targets. No one reported feeling nauseated while using the system. Subjects believed that it was easy to pick up items although it was difficult to tell when a item had been picked up without some form of confirmation.

People disliked the colours on the status bar and thought the range of colours were limited. The monster colours were good and easy to see, and the users were able to easily identify monsters. When asked "Is the movement in the augmented reality relative to the real world?" – Most people thought that the movement relative to the real world was okay but commented on the lag when rotating their heads. When asked "Is it easy to shoot at the monsters?" – Most subjects found that the lag made it difficult to align the cross hairs at the targets. The actual process of firing the weapon was easy.

7 Implementing ARQuake

The original Tinmith modules which we had previously constructed have not required any modifications to support the extensions mentioned in this paper. Two additional modules `modpatt` and `modquake` have been added to provide new features.

The `modpatt` module performs the pattern recognition using the ARToolKit, and also reads in position and orientation values from the devices in `modharvester`. (More details concerning architecture and implementation of the existing Tinmith modules can be found in [9, 10].) The `modpatt` module uses the pattern recognition extended from ARToolKit to refine the position and orientation information, which is then passed back to the Tinmith system for the other modules to process. This integration is performed by reprogramming the tables which describe how the modules connect with each other. In the case of when the system is indoors, `modpatt` is responsible for using just the camera recognition to generate position information, as the GPS does not function indoors.

The `modquake` module extracts information from the Tinmith system, such as position and orientation, and converts this into UDP packets which are then sent onto the local network. The modified Quake program then receives these UDP packets, and converts the data into Quake's local coordinate system, by scaling and translating. Quake then uses this information to render the display at the appropriate location with the correct head position, and the player can control the game using physical movement.

We found that having individual software processes (rather than threads) is an advantage because tasks such as processing the USB camera data tend to cause the process to block, in turn causing the entire system to stop processing if it was based around user threads.

8 Architecture application

As a second application domain for the use of our outdoor AR system, we are investigating how to improve the visualisation of architecture designs. The first question to ask is – Why an architect may use this tool? An AR visualization tool will aid architects in conveying their ideas to their prospective clients. A common

Figure 5: Proposal One Bridge

complaint with the architect/client relationship is that the architect not knowingly, assumes or believes that the client understands the design as well as he or she. This is not always the case, and it is proposed that this visualization tool will help in providing the client a more accurate method in which they can understand the architect's ideas.

Showing the design in an outdoor AR environment allows the client to move to different vantage points, gaining an understanding of the design. Issues such as space and form are easier for the client to understand, because they are viewing the design at the location of the final bridge. The architect may show the client other design options, such as: partially enclosing the bridge with a roof structure consisting of the same materials plus transparent roof sheeting, or fully enclosed walkway with permanently fixed windows.

An example of how an architect may wish to demonstrate to a prospective client their design ideas is as follows: the client wishes to build a walkway between two buildings. The initial requirement is to build the walkway on the second level of the buildings, directly over the existing walkway that is currently at ground level. The client has no specific requirement as to the materials to be used or the overall size of the walkway. The first proposal is a 3m wide walkway with a 1000mm high balustrade supported by three columns. Figure 5 shows our system in operation depicting this new bridge. There will be provisions for lighting located centrally down the walkway at approximately 3 meters high. This design was constructed with the professional design package AutoCAD. The design was translated and rendered with Tinmith-4 as an AR display.

9 Conclusion

This paper reports on an ongoing research project – ARQuake, an outdoor/indoor augmented reality first-person application. Although the implementation has not been completed, many interesting results have been found, in particular user interface issues for AR outdoor/indoor augmented reality games; an architecture for low cost, moderately accurate indoor/outdoor 6DOF tracking; and implementation issues for converting a desktop application into an AR application.

At the time of submitting this paper, the ARQuake game was running on our wearable computer platform with the 6DOF GPS/compass tracking system. The interaction of the game was operating within the accuracy of this tracking system. We had modelled an outdoor section of our campus and interior of the WCL. The graphics of the game were running at 30 frames per second, with GPS updates once per second and compass updates at 15 times per second.

The results of our research to this point in the implementation, which we report here, include the design and implementation of a user interface for an augmented reality application, and proposed some guidelines. Specifically, identified by informal experimental evidence, a set of colours that take into account different lighting and background colours for outdoor use. We successfully mapped the keyboard and mouse interactions of the ARQuake application to head/body movements and a simple two-button input device. A simple user interface layout was incorporated into the ARQuake application. The game is quite playable in this configuration, and we are continuing to investigate how to overcome difficulties with registration errors when indoors and close to buildings.

We proposed and described a low cost moderately accurate indoor/outdoor 6DOF tracking system, by a novel combination of optical tracking and GPS/compass tracking to provide an absolute tracking system. The solution is not restricted to overcoming registration problems, but it does provide a general tracking solution. Fiducial markers are placed at the corners of buildings to provide more accurate positioning information for ARQuake. When a fiducial marker comes into view of a head mounted camera, this more accurate tracking information is used instead of the GPS/compass data. The vision based tracking is not fully functional, and we are continuing to investigate and develop this solution.

Finally, we have shown how our system may be applied to a different application domain, architectural visualisation. We have given an example of the working application, and described how the application would be incorporated into the architecture design cycle.

Aside from the specific technical achievements of our work to date, it is perhaps most important to point out that our work can serve as proof that augmented reality is readily achievable with inexpensive, off-the-shelf software. The translation of the application set from the desktop to incorporate the physical world brings closer the possibility of truly ubiquitous computing.

Acknowledgments: For their support of this project, we would like to thank the Advanced Computing Research Centre of the University of South Australia; John Maraist; Arron Piekarski; and the Defence Science and Technology Organisation.

References

[1] R. T. Azuma. The challenge of making augmented reality work outdoors. In *First International Symposium on Mixed Reality (ISMR '99)*, pages 379–390, Yokohama, Japan, March 1999. Springer-Verlag.

[2] F.P. Brooks. Walkthrough-a dynamic graphics system for simulating virtual buildings. In *Workshop on Interactive 3D Graphics*, October 1986.

[3] T. Hollerer, S. Feiner, and J. Pavlik. Situated documentaries: Embbeding multimedia presentations in the real world. In *3nd International Symposium on Wearable Computers*, pages 79–86, San Francisco, CA, October 1999.

[4] id Software. Quake. http://www.idsoftware.com/ (last viewed May 16th, 2000).

[5] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality '99*, pages 85–94, San Francisco, CA, October 1999.

[6] G. J. Klinker, K. H. Ahlers, D. E. Breen, P. Chevalier, C. Crampton, D. S. Greer, D. Koller, A. Kramer, E. Rose, M. Tuceryan, and R. T. Whitaker. Confluence of computer vision and interactive graphics for augmented reality. *PRESENCE: Teleoperations and Virtual Environments*, 6(4):433–451, August 1997.

[7] U. Neumann, S. You, Y. Cho, J. Lee, and J. Park. Augmented reality tracking in natural environments. In *International Symposium on Mixed Realities*, Tokyo, Japan, 1999.

[8] J. Park, B. Jiang, and U. Neumann. Vision-based pose computation: Robust and accurate augmented reality tracking. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality '99*, pages 3–12, San Francisco, CA, October 1999.

[9] W. Piekarski, B. Gunther, and B. H. Thomas. Integrating virtual and augmented realities in an outdoor application. In *Proceedings of the Second IEEE and ACM International Workshop on Augmented Reality (IWAR) '99*, San Francisco, CA, USA, October 1999.

[10] W. Piekarski, B. H. Thomas, D. Hepworth, B. Gunther, and Victor Demczuk. An architecture for outdoor wearable computers to support augmented reality and multimedia applications. In *Proceedings of the Third International Conference on Knowledge-Based Intelligent Information Engineering Systems*, Adelaide, August 1999. IEEE.

[11] A. State, G. Hirota, D. Chen, W. Garrett, and M. Livingston. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. In *Proceedings of SIGGRAPH 1996*, pages 439–446, New Orleans, Louisiana, August 1996. ACM.

[12] A. State, M. A. Livingston, W. F. Garrett, G. Hirota, M. C. Whitton, E. D. Pisano, and H. Fuchs. Technologies for augmented-reality systems: Realizing ultrasound-guided needle biopsies. In *Proceedings of SIGGRAPH 1996*, pages 429–438, New Orleans, Louisiana, August 1996. ACM.

[13] B. H. Thomas, W. Piekarski, and B. Gunther. Using augmented reality to visualise architecture designs in an outdoor environment. *International Journal of DesignComputing: Special Issue on Design Computing on the Net (DCNet'99)*, 2, November 1999. University of Sydney.

[14] M. Ward, R. Azuma, R. Bennett, S. Gottschalk, and H. Fuchs. A demonstrated optical tracker with scalable work area for head-mounted display systems. In *Proceedings of 1992 Symposium on Interactive 3D Graphics*, pages 43–52, Cambridge, March 1992.