First Person Indoor/Outdoor Augmented Reality Application: ARQuake

Bruce Thomas, Ben Close, John Donoghue, John Squires, Phillip De Bondi and Wayne Piekarski

Wearable Computer Laboratory, School of Computer and Information Science, University of South Australia, Mawson Lakes, South Australia

Abstract: This paper presents a first person outdoor/indoor augmented reality application ARQuake that 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, input devices, and multi-person collaboration are discussed.

Keywords: Augmented reality; Computer games; Wearable computers

1. Introduction

Many current applications place the user in a first-person perspective view of a virtual world [1], such as games, architectural design viewers [2], geographic information systems and medical applications [3,4]. 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 (idSoftware). As with other researchers [5], we wish to place these applications in a spatial context with the physical world, which we achieve by employing our wearable computer system Tinmith [6–8]. Tinmith 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 view overlaid computer-generated information by means of see-through head-mounted displays. 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, a simplified representation of the physical world gaming location 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 walls, 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 that 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 spaces.
5. The applications are 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 investigated 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 that simultaneously display both the physical world and computer-generated images require special care. The choice of screen colours for the purely virtual images that the application must display requires attention to the lighting conditions and background colours of the outdoors. The keyboard and mouse interactions must be replaced with head/body movement and simple buttons. The screen 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 [9]. 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 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.

2. Background

There are two basic styles of tracking: absolute and relative. Furthermore, machine learning can train a system to recognise locations in a building or outdoors. Golding and Lesh use an array of sensors: accelerometers, magnetometers, temperature and light to track a user in a set of known *locations* [10]. Aoki, Schiele and Pentland [11] use a camera to train the system to recognize the user's location and approaching trajectory. These systems can determine its present room, and whether it is entering or leaving that room.

Previous research has established that outdoor tracking with inexpensive differential GPS and commercial grade magnetic compasses are inaccurate for augmented reality applications [12]. Traditional hybrid approaches combine a number of different systems such as inertial, optical, electro-magnetic and GPS. In this paper, we present our hybrid approach of combining vision-based optical tracking with GPS and a magnetic compass.

A number of researchers are investigating *fiducial* vision-based tracking [3,13]. We based our optical tracking system on the fiducial marker tracking system ARToolKit developed by Kato and Billinghurst [14]. 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

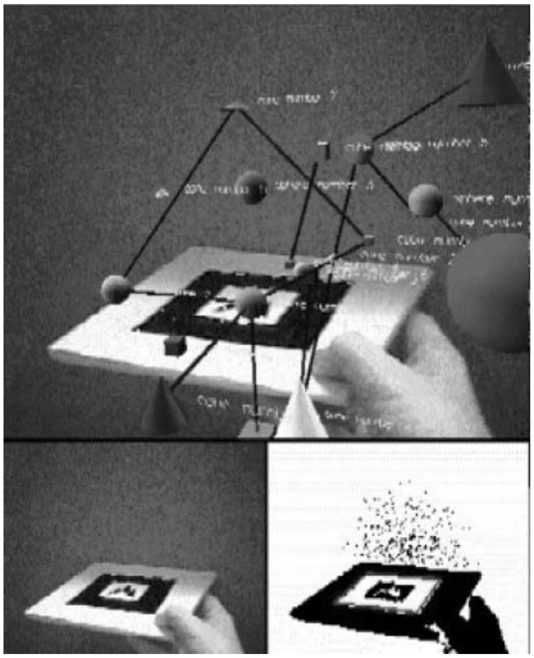

Fig. 1 Example of a fiducial marker.

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 1 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.

2.1. The original Quake game

We chose Quake as the primary application for a number of reasons. Quake fits the general model of AR that we are studying, as it is a first-person 3D application with autonomous agents to interact with the user. We were able to obtain the application 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" (idSoftware). 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. 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-plate 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.

Fig. 2. Wearable computer platform.

2.2 Wearable computer platform

The Tinmith wearable computer system hardware is all mounted on a rigid backpack so that the items can be attached firmly, (see Fig. 2). 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 (we now use an Intersense 300 tracker for head orientation), 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 power converters for the different components, 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 [8] supports outdoor augmented reality research. The system is comprised of a number of interacting libraries and modules. A number of software libraries form a support base for writing code in the system: a graphics interface on top of X windows; an interface to coordinate/datum transformations and numeric conversions; encode/decode libraries for transmitting structures over a network; tools for network communications and high level I/O; low level interfaces to Unix system calls, asynchronous I/O code, string handling, event generation, and error checking.

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 haptic gun 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. Haptic gun

To improve the play-ability of ARQuake, we replaced mouse and keyboard button presses with a haptic gun device. The aiming of the weapons is still the direction of the user's head, but the firing of the weapon and changing of weapons is performed with button presses on the new gun input device. To give the gun a "recoil" feel, we installed simpe haptic feedback into the gun.

A haptic gun was developed from an appropriate toy plastic gun (see Fig. 3). Two standard

Fig. 3. Haptic gun.

commercial push buttons were installed, along with a micro-switch to replace the primitive trigger switch. A solenoid was placed towards the rear of the gun, behind the centre of gravity, to enhance the "pitch-up" sensation of the recoil. A vibrating motor was placed as far forward in the gun as possible to increase the moment arm from the centre of gravity, thereby enhancing the effect. The sound effects are generated by the solenoid and vibrating motor.

The gun provides a number of haptic sensations and sounds. There is the single shot, which provides a strong recoil with a loud bang. The number of single shot weapons allow for additional shots to be fired after a suitable reload time. The amount of time for reloading varies between single shot weapons. The shotgun provides a double shot haptic and sound effect; this effect simulates the rapid firing of both barrels serially. The multiple firing weapons, such as the machinegun, provide a less strong recoil with a short time interval between firing. The sound effect is a higher pitch bang with a lower volume. Finally, there is an energy weapon that fires a continuous stream of energy; the haptic effect is a continuous vibration of the gun with a high pitch whining sound effect.

3.2. Monsters

There are 16 different types of monster in the Quake world. Some have attributes that make them unsuitable for inclusion in our AR version of the game. 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 our game world. These monsters types are all land-based creatures that use weapons from a distance. 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.

We excluded monsters which were too large for the environment or which had unexpected effects; those which swam, as our campus does not include water features; those which flew, they move too quickly; those which surround the user or have some other interaction which would require haptic feedback; and those whose attack tended to be immediately fatal, as they are not enjoyable.

3.3. Campus level

We created a Quake level (game world) representing the Mawson Lakes campus of the University of South Australia. The walls in Quake are the external walls of the campus buildings and the interior walls of the Wearable Computer Laboratory (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 transparent to the users during the game. The Quake graphics engine renders

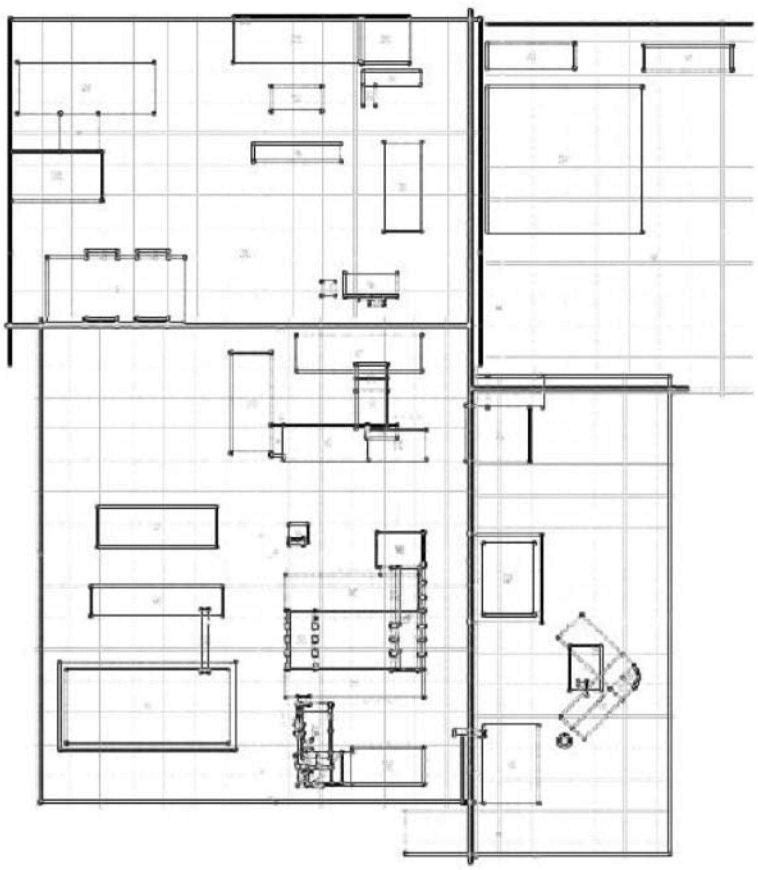

Fig. 4. Quake campus level.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase® Smarter legal research.