Integrating Virtual and Augmented Realities in an Outdoor Application

Wayne Piekarski, Bernard Gunther, and Bruce Thomas

Advanced Computing Research Centre
University of South Australia
Mawson Lakes, SA 5095
Australia
{piekarski, gunther, thomas}@cs.unisa.edu.au

Abstract

This paper explores interconnecting outdoor AR systems with a VR system to achieve collaboration in both domains simultaneously. We envisage multiple mobile users of wearable AR systems interacting with a stationary VR facility via a wireless network. An application in simulated combat training is described, where the AR users are soldiers with wearable computers, and the VR system is located at a command and control centre. For soldiers, AR provides enhanced information about the battlefield environment, which may include the positions and attributes of simulated entities for the purpose of training outdoors at low cost. At the same time a complete picture of the battlefield, including real and simulated troops and vehicles, is available via the VR system. As soldiers move about, their GPS and digital compass hardware provide the remote VR user and other AR users with the means to track their position in real-time. We describe a working system based on our modular Tinmith-II wearable computer, which interacts with a combat simulator to create a synthetic battle environment for safe training and monitoring.

Figure 1. Linking a wearable AR system to a VR system enables mobile entities (the AR user, top) to be shown in relation to fixed entities (the trees and buildings) that exist in the terrain model, bottom.

1. Introduction

The advent of wearable computers [3,8,15,16] and lightweight head-mounted displays (HMDs) has made augmented reality applications feasible outdoors. Augmented reality (AR) is the process of a user viewing the physical world and virtual information simultaneously, where the virtual information is overlaid and aligned to the physical world view [1]. Many of the existing applications of AR, such as head-up displays in aviation, assistance for surgery, and maintenance work, are characterised by requiring precise tracking in small operating regions.

In IWAR1999 - 2nd International Workshop on Augmented Reality
October 20-21, 2001 - San Francisco, Ca, USA - Copyright (C) 1999 IEEE
Please visit http://www.tinmith.net for more information

However, through coupling global positioning system (GPS) receivers and digital compass with geographic databases, we can create spatially-aware computer systems for mobile users working outdoors. We anticipate outdoor users wishing hands-free operation, and thus related AR applications are especially well supported by wearable computers and non-traditional input devices

The concept of collaboration in virtual reality (VR) has been explored previously [5,7,17], whereas collaboration in AR is somewhat less developed [4,13]. We are intrigued by the possibilities of interconnecting VR and AR systems to achieve collaboration in both domains *simultaneously*. At the simplest level, we can imagine two users—one mobile and one stationary—interacting as suggested in Figure 1. The mobile user dons a wearable AR system and is able to interact with the world outside, while the stationary user (not shown in the figure) wears a HMD or views the screen(s) of a VR workstation situated indoors. Both computer systems are in radio contact with each other. Since the AR user carries no video camera, the VR system must rely on a three-dimensional database of the outdoor environment in order to place the AR user in it. The wearable computer transmits its GPS position so that the VR system can draw an avatar at the appropriate location. Now if the VR user were to insert an object into the virtual world, an iconic representation of that object would appear overlaid on the AR user's outdoor view. We are therefore able to exploit the large computing power of the VR system, without affecting the computing requirements of the wearable AR systems.

Our previous work has demonstrated AR systems functioning as powerful navigation and visualisation tools for individuals [10,12,14]. This paper describes extensions to our work that facilitate collaborative activities in joint AR/VR applications, sited outdoors. We show how the necessary infrastructure for collaborative AR applications readily integrates with a remote VR system. To illustrate the concepts we will focus on an application for military training that incorporates both real and simulated entities, with the latter being used to enrich outdoor training exercises cost-effectively and safely.

In collaboration with the Australian Defence Science and Technology Organisation (DSTO) and the Australian Army we are investigating technology that may improve the effectiveness of the dismounted combat soldier. The initial application that we investigated was the use of a wearable computer to support navigational tasks, since navigation is an essential and time-consuming task for soldiers on foot. The significance of the navigation application lies in its departure from tracking users in small operating regions. In this regard, it relates to recent work in large area augmented reality, such as the *Touring Machine* project by Feiner *et.al* [6], which allows users to walk around the Columbia University campus and access information via a tracked see-through display and a hand-held display.

Beyond knowing his own position, a soldier must also gain awareness of enemy threats and friendly positions alike. The latter case reflects the fact that soldiers normally operate collaboratively. However, collaborators need not be restricted to other dismounted soldiers, but may include vehicles and aircraft, whose presence adds diversity and thus hinders rapid assimilation. We use AR to supply the soldier with easily understood information about his battlefield environment, for example: indications of threats, the location of friendly forces, and the status of aircraft as they approach. Fortunately, the acceptable tracking accuracy required for this application is attainable without special motion compensation in the display system. The open problem of accurate feature registration outdoors is not addressed any further in this paper [2].

Radio data communications between wearable computers underpins an outdoor collaborative AR system. The mobile units are also able to interact with the central command and control centre, which operates a stationary radio base station. This facilitates potentially two-way interaction: communication of situational awareness information to the soldier, and, conversely, detailed information back to the base for populating its 3D model. Thus, in a live training or combat situation the base can track field positions dynamically to provide high-level command and control information; however, in a simulated exercise the base can host a simulated battle accessible through a VR system. This latter scenario is the one we explored in more detail.

In the following sections we define the operational requirements of the larger system, and how it functions. We then describe our wearable AR system, and discuss how this was modified to support collaborative work. We conclude with observations gained from a test evaluation.

2. Augmented Reality Meets Virtual Reality

We set out to develop a system for demonstrating AR techniques for use by an individual using a wearable computer interacting with the Modular Semi-Automated Forces (ModSAF) simulated combat system [18]. The Synthetic Environment Research Facility (SERF) in DSTO's Land Operations Division employs ModSAF to generate synthetic battle environments in simulated exercises. A synthetic battle may, of course, involve fictitious entities (soldiers, vehicles, and aircraft) as well as the real entities actually participating in the exercise. The system is designed so that a helicopter pilot in a training exercise, for example, can participate in a virtual battle via a VR system. The pilot sees the output of the VR camera through three angled CRTs, driven by MetaVR running on a high-end PC. The MetaVR software [9] displays photorealistic battlefield terrain and receives dynamic positions of entities via the Distributed Interactive Simulation (DIS) protocol (IEEE Standard 1278), which is the principal protocol used between AR and VR systems in this project. DIS is a high-bandwidth, stateless protocol: entity positions must be continuously retransmitted if the entities are to be displayed, otherwise they are considered to be nonexistent after 5 seconds.

Persistent data about entities known to exist on the battlefield are maintained by the Land Situational Awareness Picture System (LSAP), which provides a Java front-end to an Oracle DBMS. The positions and attributes of entities held in LSAP can be accessed by command and control staff to provide them with situational awareness of the engagement. Units in the field can radio reports back to LSAP to update the battlefield view as entities are detected, eliminated, or known to move.

2.1 Objectives and Overview of the System

The main objective of the system was to demonstrate the possibility of a dismounted soldier interacting with information from remote simulation and situational awareness systems (ModSAF and LSAP) using AR techniques. LSAP and DIS were used to filter appropriate information to users in the field, who see overlaid on their view of the physical battlefield iconic representations of both real and simulated entities that populate the synthetic battle environment. In turn, users of the VR facility have a complete, albeit artificial view of the entire battlefield. We anticipate that the combination of these capabilities can facilitate operational and training developments in information transfer between individual soldiers and higher command systems.

2.2 Interaction with the Wearable Computer

As previously mentioned, we wish to strengthen collaboration by improving communication between mobile and stationary personnel. The interactions we support include:

- the wearable computer overlays on the user's HMD iconic information locating friendly and enemy forces; for example, in a training application the icons may represent the positions of dismounted forces generated by the ModSAF simulator at SERF
- users of wearable computers can update or populate the 3D database maintained centrally; for example, the user may locate a previously unknown enemy and inform the LSAP system as to the *relative* position and nature of the entity
- hardware at SERF presents a virtual view of the battlefield, which includes avatars for 3D viewing, and icons for 2D plan views; as wearable users move, their GPS systems continue to provide real-time positional data to command personnel
- furthermore, with additional real-time data specifying the pitch, roll, and yaw of the soldier's head, the presentation of the virtual world at SERF can be slaved to the soldier's direction of view; in other words, we can provide a virtual camera without video equipment

3. Wearable Computer System

The wearable computer system we used for this work was based on Tinmith-II [10], a complete research system developed at the Wearable Computer Laboratory at the University of South Australia. Figure 2 pictures the current hardware platform.

Figure 2. Photograph of the Tinmith-II wearable computer.

3.1 Hardware Components

Tinmith-II is built upon a Toshiba 320CDS notebook running the freely available LinuxOS. The laptop is about the size of an A4 book and fits comfortably on a wearer's back. A Sony PLM-100 transparent display, worn on the user's head as shown in Figure 3, allows the video output of the computer to superimpose images over the real world. A Phoenix miniature keyboard attached to the forearm enables the user to interact with the system and enter commands.

Figure 3. Users of the wearable computer see through a Sony PLM-100 transparent head-mounted display, and enter commands on a Phoenix forearm keyboard.

To support the navigation functions of our application, a GPS module (with differential receiver) connects to the laptop, and provides position fixes at most places in the world to within 5 metres accuracy. A TCM2 3-axis digital compass, also attached to the display, allows the computer to determine exactly how the wearer's head is oriented relative to the surface of the Earth. This information is used to render the display so that it remains in correspondence with the user's physical world view.

All of the equipment is attached to a rigid backpack, along with batteries and antennae. The prototype hardware and software system is fully functional in outdoor environments.

3.2 Head-Up Displays and Interfaces

The system is capable of presenting alternative interfaces to the user, namely, two-dimensional and three-dimensional. Since navigation remains an important activity for the soldier, all of the user interfaces feature our original navigational aids.

The 2D interface incorporates a first person perspective, God's-eye view, and traditional non-spatially aware information on one display, as shown in Figure 6. At the top of the display is the compass heading, which is represented as notches and a value every 45 degrees. As the user rotates their head, the compass updates itself and scrolls left or right to indicate the new heading. The various pieces of text in formation placed around the display are used to show position, GPS accuracy, and steering instructions to the nominated waypoint so as to aid navigation by the wearer. Underneath the text is the map and navigation gadget display. At the centre of the display is a cross, indicating the current position of the system and user. Shown in the figure are outlines of objects in the environment, and the circular object in the centre of the screen gives steering instructions to the user indicating in which direction they must turn to reach the target. Every visual cue is rotated in real time as the user moves around. In operation, we were able to maintain approximately 2 to 5 metre accuracy given a good fix on 6 or more GPS satellites.

The alternative interface features a 3D immersive display superimposed on the wearer's field of view. When looking through this display we are able to register the display with the outline of a surveyed object within the accuracy limits of the differential GPS and digital compass. For example, we were able to make a cube lock around an outdoor bench as we walked around it.

3.3 System Architecture

To support a wide range of AR and navigation tasks applications, we developed a highly modular architecture. The software system is broken up into various modules that communicate with each other using a connection-oriented protocol—in this implementation, TCP/IP.

Modular approach

An example of an application specific module is the *navigation* module, which reads waypoints from a database, along with position and heading information, to produce steering instructions for other modules to present to the user. The *display* module presents data from other modules in a graphical format to the user via the head up display. The modular architecture supports many concepts such as data abstraction and hiding, collaboration with others, and the flexibility to plug new components in without modifications.

Communications

To interconnect modules, we used a client-server style architecture. The server is a data source for other modules, which subscribe to it. Whenever the server updates the value of system data (a GPS position, for example), it will send the new value out to all clients that have registered an interest in the message. A client receiving new data may use it to update the screen, or calculate new navigation parameters for example. Actually, many servers in the system also act as clients for other servers as well. The entire system operates asynchronously, and is data driven; if there is no new data in the system, no action will be taken by any of the software modules.

Suppose a new incoming position was received from the GPS hardware. The new data will be formatted appropriately and then distributed to all client modules. The navigation module will receive this update, and recalculate navigation information. The display module will eventually receive a position update, along with the new steering instructions from the navigation module, and use these to redraw the screen to reflect the user's new location.

3.4 Software design

To implement the modular architecture, an appropriate supporting software library was designed, with goals to be flexible, extensible, and layered. Layering was exploited to provide increasing levels of abstraction for allowing modules to interact with the system at the appropriate level they require, while at the same time minimising code replication across the system, and localising possible errors. The libraries provide functionality for distributive processing, asynchronous I/O, dynamic configuration, and automatic code generation.

Running modules in parallel over TCP/IP

Each of the modules are implemented as separate Unix processes, which communicate via kernel network services. This allows modules to be distributed over multiple processors on one machine, or multiple machines due to the network support. The ability for the system to support this at a fundamental level improves the scalability for larger, resource-intensive applications. For those modules confined to a single machine the networking between them is virtualised by the kernel, and hence the corresponding latency and bandwidth of the communication channels are orders of magnitude better than through a real external network.

Dynamic configuration from a DBMS

Most software tends to use statically compiled controls, or possibly a configuration file. Our system takes configuration to the next level by loading all system parameters such as the 2D maps, 3D object models, location of modules, port numbers, device names, and screen configuration into a collection of relational database tables. When the software initialises, it queries the database and loads the values required. By sending messages throughout the system when changes are made, it is possible for clients to reconfigure themselves by querying the database. The software does not have to be restarted as would be required if the controls were static. The database proved to be very powerful because it can be changed remotely over the wireless network. This feature turned out to be useful when testing outdoors, for example, tuning the various display options such as colours and font sizes. An added feature is the strong type checking by the database engine (in our case PostgreSQL v6.4 [11]) rather than relying on parsing a text file.

4. Integrating a VR Simulation System

Figure 4 pictures the dataflows and interfaces in our concept demonstrator system. A PC running our *modlsapdis* software mediates between the base station and one or more wearables in the field. Internally *modlsapdis* comprises two modules: a protocol converter that translates packet formats between DIS and our internal format; and a communication module that relays packets via a wireless network to any wearables. Lucent WaveLan 2.4GHz wireless network cards at the base station and in each wearable form the basis for a high-speed (faster than 2Mbps) network.

In effect the communication module of *modlsapdis* is really a part of the wearable systems, except that it runs on a stationary computer. As each wearable unit goes on-line it registers with the communication module in order to establish a wireless point-to-point network link with the base station. In the case where several wearables have registered such links the communication module must multicast entity state packets to all registered computers so that every user in the field receives updates to their environment. Since communication between Tinmith modules is based on TCP/IP, the base station appears logically to a wearable system as just another software module.

At the base the *modlsapdis* protocol converter listens on the ethernet network for incoming entity state DIS packets from the simulator, while also communicating textual data to LSAP via a Java remote method invocation (RMI) interface. *Modlsapdis* continuously receives information about entities in the battlefield, reformats the information, and passes it to the communication module. To avoid flooding the wireless links, duplicate updates and objects that have not changed are removed from the data stream. Moreover, the AR software is not required to respond to all simulation events.

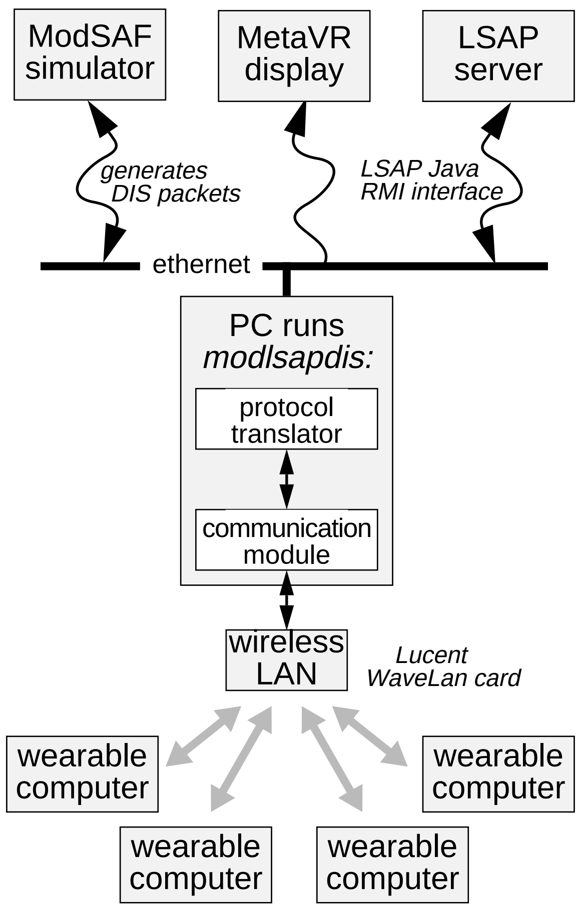

Figure 4. Block diagram of the combined AR/VR system. ModSAF generates a simulated battle environment, which is tracked in the LSAP database. One or more wearable computers in the field interact with the VR host PC via wireless network links. The entire engagement is visible on the MetaVR display.

4.1 Simulation and VR System

ModSAF is a complex, complete military simulation environment with a graphical interface. It can simulate a complete battle, including the movements of tanks, helicopters, and soldiers. Thus, soldiers participate in simulated combat, drive and refuel vehicles, reload ammunition, and fly aircraft, in addition to numerous other activities. During simulation, ModSAF maintains a terrain database, emits DIS packets defining the dynamic positions of entities, and produces a complete map showing a top-down view of all entities moving over the terrain.

The MetaVR visualisation software provides us with battlefield views so that a battle can be observed as it unfolds. MetaVR renders a world with photorealistic terrain, which includes all DIS entities that need to be drawn and animated, such as rotating helicopter rotors, and movable tank turrets.

MetaVR is also used to render displays for an attached helicopter simulator. Three computers running MetaVR provide three views from the simulated cockpit; one machine simulates the motion and dynamics of the helicopter, while another draws views of the helicopter's infrared turret. This is a good example of how DIS caters for distributed processing, even though only one helicopter is being simulated. With wearable computers in the field it is possible for the helicopter pilot to see the location of each wearable unit, while a soldier with a wearable can see simulated helicopters moving around in his field of view.

4.2 Wearable Computer Software

The internal software architecture of the wearable computer is depicted in Figure 5. The *modtracker* module is designed to distribute the data read from the wireless network link to all of the other installed modules. In our system, modules subscribe to other modules. For example, we have the *navigation* and *display* modules both listening for entity information, and it would be wasteful of bandwidth to send two copies over the network each time an update occurs. Moreover, some code might wish to determine the latest location of all battlefield entities, and thus another purpose of *modtracker* is to build and maintain a local copy of the entire environmental state for the remote wearable computer.

In some sense, *modtracker* acts as a proxy; it receives all the external information available to it, stores it, and then redistributes it to other modules on demand. Since *modtracker* is the only module allowed to access the wireless link, we can easily hide the implementation of the communication protocols from the other modules, such as

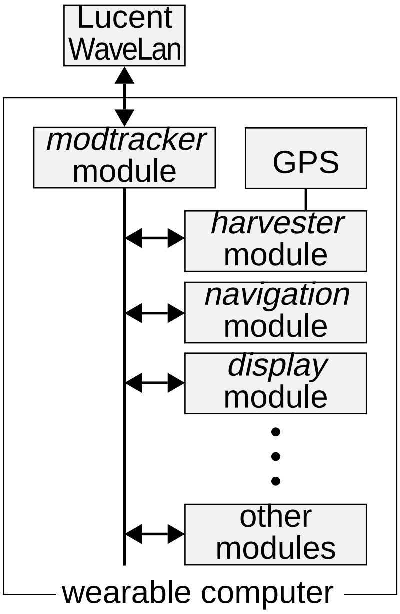

Figure 5. Software architecture of the wearable computer equipped to interact with ModSAF.

the *display* module, for instance. This allows future implementations to include data compression or replacement of TCP/IP without affecting the remainder of the system.

The reverse function of the wearable computer system is to return the position and orientation of each soldier. The *harvester* module sends the GPS location out through *modtracker*, which transmits data packets to the fixed computer at SERF. Ordinarily, these packets identify individuals in the field so that a VR user at SERF can visualise both the simulated and real entities from a bird's eye view. However, it would be equally feasible to slave the VR camera to any individual's head movements so as to gain their *personal view*—albeit with artificial renditions of real objects.

5. Initial Evaluation

The wearable computer system was used in demonstrations of the capability of an individual walking around and interacting simultaneously with the ModSAF and LSAP systems at DSTO.

5.1 Practical Trials

The first exercise involved some 50 entities of all kinds (including tanks, helicopters, and soldiers) generated by ModSAF and populating a synthetic battlefield. We wished to determine whether our system could cope with a large number of dynamic entities, and keep them in the correct position on the AR display. We observed that the display frame rate did not degrade due to the data load, and that the display was being updated with entity positions as they were generated in real-time.

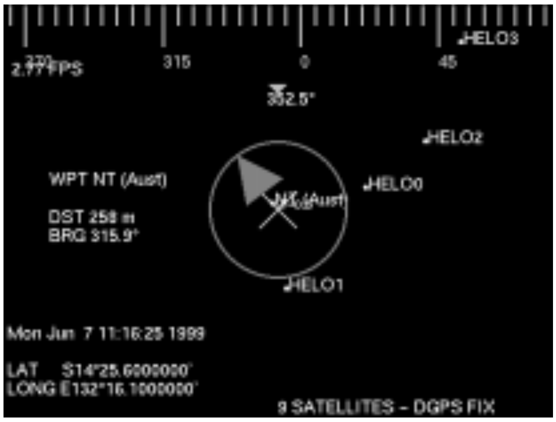

Figure 6. Two-dimensional, plan view of the battlefield, as generated to augment the user's outlook. Central cross marks the user location. Four helicopters, HELO0–HELO3, are shown as labelled dots in relation to the user's position.

In the second experiment, a simulated helicopter was made to "fly" around the wearable computer user standing in MetaVR's synthetic environment. As the helicopter circled overhead, it was easy for the wearable user to track the motion of the helicopter on their display by rotating their head. Of course, the user on the ground could see only an iconic representation of the helicopter, as the simulated craft did not really exist. This experiment proved that moving, simulated entities generated by ModSAF could be observed in real-time through the AR interface. We verified the result by comparing the AR display with a map generated by ModSAF. For example, when the helicopter was located north of the mobile user, they had to face north in order to see the helicopter.

An illustration of the generated 2D AR interface appears in Figure 6, where the dark background would normally be replaced by the user's view of the real environment. In this view, the user occupies the central cross and is navigating with the aid of the compass displayed at the top. The four (in this case, simulated) helicopters flying overhead are indicated on the plan as labelled dots in scale with the user's distance from the waypoint ahead (marked "WPT NT Aust" on the display).

With the press of a button the alternative 3D interface can be invoked, as pictured in Figure 7. Here, the central cross is directed by the user's head orientation, which appears to be just below the horizon in Figure 7. The hashed diamond to the left represents the current waypoint being navigated. Three (simulated) helicopters

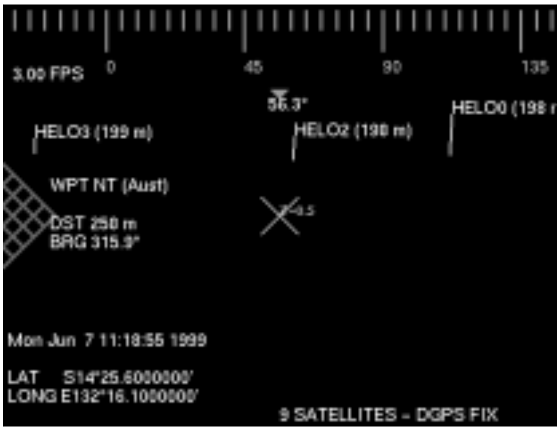

Figure 7. Three-dimensional view of a similar battlefield, as generated to augment user's outlook. Four helicopters, HELO0–HELO3, are shown as labelled altitude bars above the horizon, with HELO0's nearness indicated by the largest bar.

appear as labelled altitude bars emanating from the horizon. All the helicopters are flying at approximately 200 metres, but HELO0 is significantly closer than HELO3 and HELO2, and hence the altitude bar of HELO0 is correspondingly larger to indicate this fact. If an airborne entity is almost directly overhead its altitude bar simply extends to the top of the display.

Thirdly, we ensured that a user of the MetaVR system could visualise wearable AR units in the field. Figure 8 depicts four views of such a trial. Soldiers with AR equipment were rendered as 3D texture-mapped avatars, which correctly appeared and moved in relation to the real and simulated entities. The 3D models responded to movements in the corresponding wearable units, showing walking motion or rotations, for example. And by slaving the VR camera to the head position of a wearable unit, we were able to demonstrate an individual's view of the battlefield to a person sitting at the fixed SERF site.

(a) Mobile user, standing outdoors, observes a virtual helicopter.

(b) God's-eye view, AR screen display superimposed on wearer's view. Note a helicopter (labelled CHOPPER) directly in front of the wearer, whose position is marked by an X.

(c) Scene generated by MetaVR at the base station. It shows representations of both the mobile AR user and the virtual helicopter they are viewing.

(d) Corresponding screen shot from ModSAF, which generated the virtual entities appearing above.

Figure 8. Example application showing the integration between the AR and VR systems, simulator, and the real world in which the AR user stands.

5.2 Performance Measurement

In the second phase of our evaluation we wished to test the performance of our system to establish responsiveness and capacity to operate with a large number of entities. As part of its event driven nature, Tinmith updates its display in response to any change in user position or entity locations. Since the digital compass sends updates at a rate of 15Hz, the GPS updates at 1Hz, and entities arrive at indeterminate times, the frame rate generally lies between 15 and 20 frames per second, providing a fast updating display for the user.

In order to get some quantitative results we added instrumentation code to the Tinmith network libraries, which involved taking a timestamp in microseconds, using POSIX call `gettimeofday()`, embedding this in the message, and then decoding and comparing the result at the receiving end to measure the time taken in creating and transmitting the message.

We created an artificial test situation with the following parameters:

- ModSAF generating DIS updates at 1 Hz for each of 7 entities sent over the wireless LAN to the wearable
- *modlsapdis* sending DIS packets for all 30 internal markers (waypoints) being tracked at 0.25 Hz. Since these markers are transmitted on the SERF side of the network, the wearable does not concern itself with sending these updates except when changes are made.
- the wearable position and orientation updated at 15 Hz, with this data transmitted over the wireless LAN each time.

During this test we observed that the CPU was reasonably utilised but not working at 100% capacity, and that the measured module to module delay ranged from 2.8 ms to 50 ms, although this increased to 100 ms in the event of the machine being temporarily heavily loaded. The accuracy of these measurements was marginally affected by the limited resolution of the Linux kernel timer and the overheads in calling it.

Module to module delay was measured only for messages being sent to the same machine over the local network. In attempting to measure messaging delays between *modtracker* and *modlsapdis* it was not practical to synchronise the clocks between the machines to microsecond accuracy and hence no proper measurements could be reported.

From the point of view of the MetaVR user, the delay between changes in orientation and the update of the display was definitely noticeable, with a lag of around 500 ms being observed consistently. This lag can be attributed to a small delay in the wearable, data transmission over the wireless LAN, retransmission as a DIS packet to MetaVR, and eventual rendering on the display. Since three separate computers (and their operating systems) and two networks are involved, some delay is to be expected. In practice, update lags to the VR display are not an issue because staff at headquarters are unable to see a real battlefield. Hence, the VR display is still an accurate rendition.

A powerful feature of the *modlsapdis* module is that even if the wireless LAN is disabled temporarily, due to an obstruction, for example, the module still retains state information from the unavailable wearable. *Modlsapdis* will continue to transmit waypoints and the wearable computer's last location, so the VR display and other wearables will still be able to visualise the entity and its markers. Without such a feature, a briefly unavailable wearable would disappear on account of the 5-second DIS protocol timeout.

6. Conclusions

With the emergence of outdoor AR, we expect integration with VR and collaboration to become more widely applicable for two reasons. Firstly, these capabilities give individuals remote from each other the option of an immersive user interface to substitute for physical presence with collaborators. For example, movements of users in the real environment are reflected in the VR. Secondly, multiple AR users can benefit from viewing a superimposed collective information space, which may include simulated entities where appropriate. This concept was exploited in the simulated combat training system to provide the illusion of a rich environment at relatively low cost, yet users of wearable computers are not restricted to training indoors as they might have been with VR equipment.

In regards to the architecture, use of it has shown a few places where changes could be made to improve both the throughput and latency of the system. It was not originally designed to be a system that would run optimally fast for AR registration purposes, but rather as a testbed with which we could test ideas for navigation and interaction across systems, such as those at DSTO's SERF. Future changes being considered are the use of shared memory and/or merging of modules (*display*, *navigation*, and *harvester*) to maximise the performance of local modules. The delays caused by task switching, kernel call overheads, and the like are small, but not negligible when trying to produce real time systems. However, we wish to maintain the network architecture, as it allows us to integrate modules and other software across platforms.

The Lucent WaveLAN cards are particularly suitable for this application due to their size and Mbps+ performance, which exceeds the performance offered by competing technologies such as cellular phones or standard radio modems. This allowed us to perform initial tests indoors using readily available Ethernet to substitute for the WaveLAN cards. Some problems we experienced with the cards include their requirement for line of sight reception, relatively short range (a few hundred metres), and their vulnerability to radio frequency interference.

Although not originally designed to support interaction with ModSAF, our Tinmith-II wearable computer was readily adapted to provide AR support for simulated combat training. The modular, data driven architecture and layered implementation of the software made it possible to incorporate new display sources with relative ease. The navigation module, which previously updated the display as the user moved, now also returned position data to the VR site to enable visualisation remotely. We found the accuracy afforded by differential GPS to be acceptable in a training application, particularly when relying on the two-dimensional AR interface. DSTO are interested in continuing development of the concepts described here, and we hope to conduct more detailed evaluations of the system under actual training conditions in future.

Acknowledgements

The authors would like to acknowledge the assistance of Victor Demczuk and Franco Principe of Land Operations Division at DSTO, Salisbury, South Australia. This work was supported in part by a DSTO research contract, No. 486216, "Concept Demonstrator System for Demonstration of Augmented Reality Techniques."

References

[1] R. Azuma, "Survey of augmented reality," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, 1997.

[2] R. Azuma, "The challenge of making augmented reality work outdoors," in *Mixed Reality: Merging Real and Virtual Worlds*, Springer-Verlag, chap. 21, pp. 379–390, 1999.

[3] L. Bass, C. Kasabach, R. Martin, D. Siewiorek, A. Smailagic, J. Stivoric, "The design of a wearable computer," in *CHI'97 Looking to the Future*, ACM SIGCHI, pp 139–146, 1997.

[4] M. Bauer, T. Heiber, G. Kortuem, and Z. Segall, "A collaborative wearable system with remote sensing," in *Proceedings 2nd. Int'l Symp. on Wearable Computers*, pp. 10–17, Oct. 1998.

[5] C. Carlsson and O. Hagsand, "Dive—a platform for multi-user virtual environments," *Computers and Graphics*, pp. 663–669, 1993.

[6] S. Feiner, B. MacIntyre, T. Hollerer, A. Webster, "A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment," in *Proceedings 1st. Int'l Symp. on Wearable Computers*, pp. 74–81, Oct. 1997.

[7] T.A. Funkhouser, "Ring: A client-server system for multi-user virtual environments," in *Computer Graphics*, pp. 85–92, April 1995.

[8] S. Mann, "WearCam (The Wearable Camera): Personal imaging systems for long-term use in wearable tetherless computer-mediated reality and personal photo/videographic memory prosthesis," in *Proceedings 2nd. Int'l Symp. on Wearable Computers*, pp. 124–131, Oct. 1998.

[9] MetaVR, Inc., *MetaVR Virtual Reality Scene Generator*—http://www.metavr.com, 1999.

[10] W. Piekarski, D. Hepworth, V. Demczuk, B. Thomas, B. Gunther, "A mobile augmented reality user interface for terrestrial navigation," in *Proc. 22nd Australasian Computer Science Conference*, Auckland, NZ, pp 122–133, Jan. 1999.

[11] *PostgreSQL v6.4.2 Database Engine*—http://www.postgresql.org, 1998.

[12] B. Thomas, V. Demczuk, W. Piekarski, D. Hepworth, B. Gunther, "A wearable computer system with augmented reality to support terrestrial navigation," in *Proceedings 2nd. Int'l Symp. on Wearable Computers*, pp. 168–171, Oct. 1998.

[13] B.H. Thomas and S.P. Tyerman, "Collaboration issues for augmented realities in an outdoor environment," in *Proceedings INTERACT97 combined Workshop on CSCW in HCI*, Sydney, pp. 29–33, Jan. 1997.

[14] B. Thomas, S. Tyerman, K. Grimmer, "Evaluation of three input mechanisms for wearable computers," in *Proceedings 1st. Int'l Symp. on Wearable Computers*, pp. 2–9, Oct. 1997.

[15] E.O. Thorp, "The invention of the first wearable computer," in *Proceedings 2nd. Int'l Symp. on Wearable Computers*, pp. 4–8, Oct. 1998.

[16] T. Starner, B. Schiele, and A. Pentland, "Visual contextual awareness in wearable computing," in *Proceedings 2nd. Int'l Symp. on Wearable Computers*, pp. 50–57, Oct. 1998.

[17] B. Thomas, D. Stotts, and L. Kumar, "Warping distributed system configurations," in *Proceedings 4th. Int'l Conf. on Configurable Distributed Systems*, pp. 136–143,May 1998.

[18] US Army Simulation, Training, and Instrumentation Command (STRICOM), *ModSAF*—http://www-leav.army.mil/nsc/stow/saf/modsaf/index.htm