Augmented Reality in a Wide Area Sentient Environment

Joseph Newman*, David Ingram*, Andy Hopper*†
{jfn, dmi, ah}@uk.research.att.com

* AT&T Laboratories Cambridge, 24a Trumpington Street
Cambridge CB2 1QA, United Kingdom

† Laboratory for Communications Engineering
Cambridge University Department of Engineering,
Trumpington Street, Cambridge CB2 1PZ, United Kingdom

Abstract

Augmented Reality (AR) both exposes and supplements the user's view of the real world. Previous AR work has focussed on the close registration of real and virtual objects, which requires very accurate real-time estimates of head position and orientation. Most of these systems have been tethered and restricted to small volumes. In contrast, we have chosen to concentrate on allowing the AR user to roam freely within an entire building. At AT&T Laboratories Cambridge we provide personnel with AR services using data from an ultrasonic tracking system, called the Bat system, which has been deployed building-wide.

We have approached the challenge of implementing a wide-area, in-building AR system in two different ways. The first uses a head-mounted display connected to a laptop, which combines sparse position measurements from the Bat system with more frequent rotational information from an inertial tracker to render annotations and virtual objects that relate to or coexist with the real world. The second uses a PDA to provide a convenient portal with which the user can quickly view the augmented world. These systems can be used to annotate the world in a more-or-less seamless way, allowing a richer interaction with both real and virtual objects.

1. Introduction

We perceive the real world at a much higher level of detail than we can possibly hope to define for an artificial Virtual Environment. We want to be able to retain this level of detail, whilst augmenting it, where appropriate, with extra data obtained from a variety of sensor systems. Such context-sensitive visualisation of data could be useful in many tasks, ranging from a technician fixing a complex piece of equipment to a tourist locating objects in an art gallery.

In order to fully determine the potential of mobile Augmented Reality (AR), we believe that it is important to deploy these systems throughout a large and populous area. The deployment process highlights practical issues such as cost and ease of installation. Furthermore, if large numbers of people are working in the augmented environment on a day-to-day basis, we are forced to consider the social integration aspects of the system.

We also believe that a properly-designed AR system could be thought of as an instance of a Ubiquitous Computing [21] system; that is to say, it would provide AR facilities throughout the environment, but these facilities would be relatively unobtrusive to users. Other authors have pointed out that this goal could be realised by displaying information using either *personal* or *environment* displays [10]—in this project we take the personal display approach, displaying augmentation information on appliances that are associated with a particular user. Our system is careful to avoid what we refer to as visual pollution of the environment, since it does not require targets, cameras or fixed panel displays. We also wish to avoid burdening the user with excessively heavy or cumbersome equipment which they have to carry around.

1.1. A sentient environment

Recently, a system has been designed at AT&T Laboratories Cambridge that uses sensors to update a model of the real world [13] [1]. The state of the environment is encapsulated within the model, and by using the data within it we can create applications that respond in an appropriate way to changes in the environment. To the users of the system, it therefore appears that the system shares the user's perception of the world, and so we refer to it as a *sentient* system.

The model is implemented as a collection of CORBA software objects, each of which corresponds to a single object in the real world. Every software object stores the description and current state of the corresponding real world object. To determine the current state of the environment, applications contact and query the set of software objects corresponding to the real world objects they are concerned with, via their CORBA interfaces.

Information from a range of sensors is used to update the state of the software objects—we gather information, for example, from a specially designed object location system, our CTI telephone switch and resource and keyboard monitors running on computers. Software objects perform filtering of the raw sensor data before it is used to maintain object state, permitting different filtering schemes for different types of object.

The system has been deployed throughout our building, and the model currently contains 1900 software objects corresponding to personnel, telephones, computers, walls, windows, etc. in the real world. In this project, we aim to utilise the detailed data set inherent in the sentient system to provide users with a rich AR experience.

1.2. Location system

To create an accurate model of the environment, the sentient system requires detailed knowledge of the 3D positions and orientations of objects in the environment. Similarly, to provide users with an AR experience, it is necessary to be able to ascertain the 3D position and viewing direction of the user with high accuracy and low latency [4]. As part of the sentient computing system we have developed a scalable, in-building tracking system that can provide location information for both of these purposes.

We have built small wireless devices called Bats which are worn by personnel and attached to equipment. The sensor system measures the time taken for ultrasonic pulses emitted by the Bats to reach receivers in known, fixed positions in the ceiling. It uses these times-of-flight to calculate the position of each Bat. Estimates of the position of the tagged object can then be made. If the object is multiply-tagged then an estimate of orientation can also be made.

The 3D position fixing accuracy of the system is within 3cm, 95% of the time. The system is capable of making 150 location updates each second across the whole building, locating up to three Bats in every 20ms period (an interval of time known as a *timeslot*). Location update opportunities are shared between Bats using a quality-of-service (QoS) measure. The Bats used directly by the AR system are given a high QoS, which allows the system to make the good, timely estimates of orientation and position needed to give users a sense of immersiveness.

The system is installed throughout our three floor, 100 000 cubic foot office, which has over 50 rooms. The system is continually used by all 50 staff, and tracks over 200 Bats. The Bats have a battery lifetime of 12 months. The ultrasonic receivers are mounted recessed in the centre of the ceiling tiles, with cables in the roof, which makes the tracking infrastructure extremely unobtrusive.

2. Head-mounted Display

One approach to providing users with an AR experience via a personal display is to project augmentation information onto an optical see-through head-mounted display (HMD) unit.

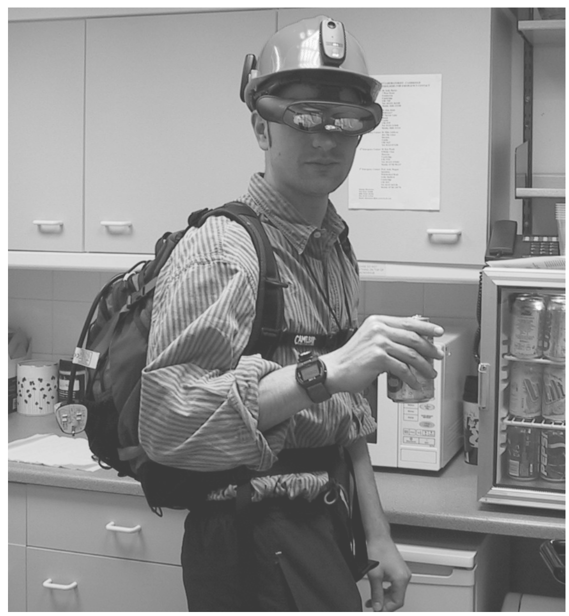


Figure 1. Prototype system

2.1. Hardware

Our HMD system consists of a 750 MHz IBM Thinkpad T21 equipped with a Lucent WaveLAN card to provide networking. Tracking is performed by an InterSense InterTrax inertial tracker, and by three Bats which are mounted onto a hard hat along with a Sony Glasstron head-mounted display, running at a resolution of 800×600 pixels. The laptop, rechargeable batteries and power supplies are mounted in a backpack with a single power cable, enabling the system to be docked anywhere in the building (see Figure 1). The

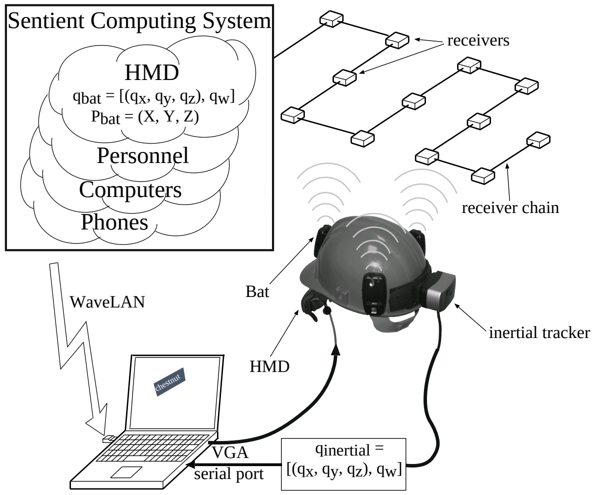


Figure 2. Head-mounted display tracking

HMD system can run for approximately 3-4 hours before it needs to be recharged.

2.2. Tracking and fusion

The HMD-mounted Bats are used to obtain a least squares estimate of the position and orientation of the user's head [3] (see Figure 2 for an overview of the system). The HMD software object is responsible for taking in the raw Bat readings, and giving the best estimate of the current head position and orientation. The software object takes the three position readings from the Bats and calculates the orientation as a quaternion.

The sentient computing system arranges that the three adjacent Bats transmit their ultrasonic emissions in consecutive 20 ms timeslots, which ensures that the signals from the Bats cannot interfere with one another. However, the calculation assumes that the measurements of absolute Bat position are simultaneous. In near-static cases, this assumption does not lead to gross errors, but, during rapid head motions this is no longer the case because the Bats have moved during the 40ms interval between the first and last measurement. The HMD software object attempts to reject erroneous measurements by testing how well the measured geometry of the Bats conforms to the known geometry. If any of the three measurements of inter-Bat distance are more than 5cm in error then the associated orientation is rejected. Each set of Bat readings yields a raw measurement of head position and orientation. Noise will cause these measurements to differ from the true head position and orientation, and it is therefore worthwhile considering how filtering may be used to determine more accurate estimates of the current head position and orientation using a series of measurements. In practice, slight errors in the head position are not noticed by the user, but random errors in head orientation are very disconcerting to the user, and so our filtering scheme concentrates on improving estimates of head orientation.

The HMD software object takes the current estimate of head orientation and uses a non-linear filter to make a new estimate based on the latest raw measurement. The filter assesses how close the estimate of orientation is to the latest reading. If the difference between the two orientations is small, a small correction is applied to the old estimate, resulting in slow HMD orientation changes being heavily damped. If the difference between the orientations is large, then a much larger correction is made to the old estimate, and hence fast HMD movements are lightly or negligibly damped.

This filter is very straightforward to implement using the technique of spherical linear interpolation (SLERP) [8]. The quaternion representation of an orientation is a point on a hypersphere (see Figure 3). In order to move smoothly, and directly from one orientation to another we use SLERP to move along the shorter great arc connecting the two points. SLERP is expressed thus:

$$\begin{aligned} \cos(\Omega) &= q_0 \cdot q_1 \\ Slerp(q_0, q_1, h) &= \frac{q_0 \sin((1-h)\Omega) + q_1 \sin(h\Omega)}{\sin(\Omega)} \end{aligned}$$

The fraction of the arc, h, which is traversed depends on the magnitude of the angle θ, through which a body would be rotated to move from the orientation represented by q_0 to that represented by q_1. The relationship between θ and Ω is given by

$$\cos\left(\frac{\theta}{2}\right) = \cos(\Omega)$$

where $-\pi \leq \theta \leq \pi$ and h is given by

$$h = \begin{cases} k|\theta| & : \ k|\theta| < 1 \\ 1 & : \ k|\theta| \geq 1 \end{cases}$$

The effect of this filtering is to apply very small corrections to the estimate of orientation when head motion is slow. When head motion is faster the corrections are much larger, and when $|\theta| \geq \frac{1}{k}$ the next estimate of orientation is taken to be true, i.e.

$$Slerp(q_0, q_1, 1) = q_1$$

The laptop connected to the HMD unit gathers information from the HMD software object and other software objects in the sentient system over the WaveLAN wireless link. In principle, the laptop could then use the HMD software object's current estimate of the head position and orientation to render augmentation information onto the HMD

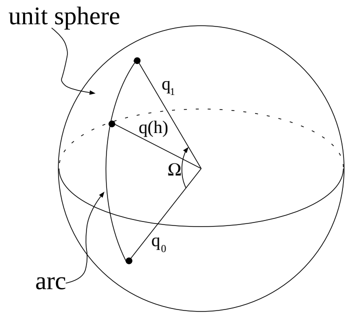


Figure 3. Spherical Linear Interpolation

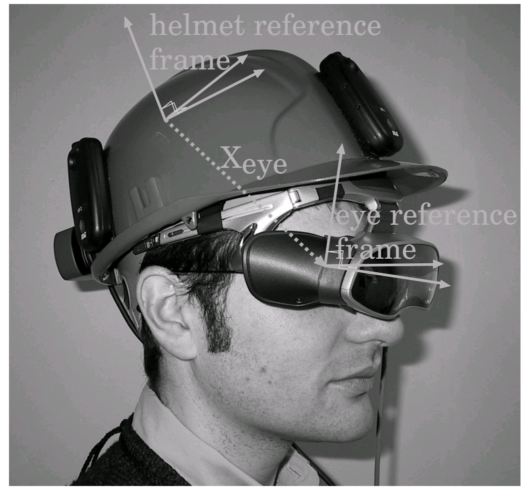


Figure 4. Calibration

display unit. However, in practice, the Bat location system provides only 2-3 measurements of the head position and orientation each second—by itself this update rate is insufficient to give any sense of immersivity.

To work around this limitation we fuse the sensor data from the HMD software object with that from the inertial tracking unit, which can provide orientation information with a very high update rate (up to 100 updates per second). The inertial tracker only provides orientation updates, but it is less crucial to provide frequent estimates of head position than orientation as angular velocities result in much larger image velocities than those caused by translational velocities. The estimate of orientation provided by the inertial tracker is prone to drift, so we use the Bats to correct for this medium-to-long-term drift, and trust the inertial tracker in the short periods of time between Bat readings.

Each time an estimate of the head orientation is made by the HMD software object the estimate is communicated using CORBA to a process running on the laptop. This estimate is compared with the most recent reading from the inertial tracker, and a raw correction to the inertial tracker is then calculated:

$$q_{correction} \quad = \quad q_{bat} q_{inertial}^{-1}$$

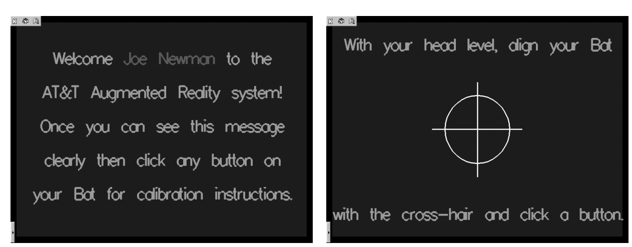
The laptop then uses the same filtering scheme as the HMD software object to make new estimates of the drift correction which must be applied to the output of the inertial tracker, based on the previous estimate and the latest raw correction. For each new frame, the most recent inertial reading is multiplied by the correction quaternion.

It would be possible to develop a similar sensor fusion scheme using an extended Kalman filter. This approach would, however, be more complex, requiring the calculation of Jacobian matrices and therefore greater computational load on the laptop.

The HMD system is similar to the InterSense IS600 [11] [12] in that it fuses data from both inertial and ultrasonic sensors. The IS600 is superior in terms of update rate, latency and accuracy, but is specifically designed to track HMDs and cameras, and in these applications the tracking hardware attached to mobile objects can afford to be relatively bulky and complex. In contrast, Bats are very simple and small devices that allow positions of a multitude of different objects to be determined on a real-time basis, whilst providing location and orientation accuracy sufficient to experiment with AR systems.

2.3. Calibration

Having calculated the position and orientation of the head, it is necessary to transform these into the reference frame of the user's eyes. To register augmentation information correctly with respect to the real world we must accurately determine the characteristics of the optical system (i.e. the coupling of the user's eye, HMD and tracking sensors) [2] [15].



(a) Instruction screen (b) Target screen

Figure 5. Calibration screens

Most calibration procedures are lengthy and do not yield convincing results given the accuracy of the Bat system. As we are not attempting to achieve extremely accurate registration of objects, we are satisfied with a less rigorous but quicker approach which involves a translation by the vector X_{eye} from the origin of the helmet reference frame to the user's eye, followed by a rotation, q_{trans}, from the helmet reference frame to that of the eye. The rendering of the scene is relatively insensitive to small changes in X_{eye}, and this value can be reused for different users even if they have very different cranial geometries. The rotation q_{trans} is more sensitive to how the helmet is being worn and must be calculated on a user-by-user basis. A series of screens (see Figure 5) ensures the user is wearing the HMD properly, and guides them through the calibration procedure. The user clicks their Bat over a cross-hair (see Figure 5(b)) in the centre of their field of view. The user is requested to keep their head level (no roll) so that both a view direction vector and up-vector can be determined. From these vectors we can make an estimate for q_{trans} which is sufficiently good for our purposes.

Figure 6 shows a typical view through the HMD. In this case the system has labelled a person, a computer (hostname tamarillo) and a telephone (number 498). Of course, if the person (or other object whose position is monitored by the sentient system) moves, the label follows them in the user's view.

Figure 6. View through HMD

3. Batportal

The Batportal is another form of AR system, and is a lightweight alternative to the HMD. We use a hand-held PDA, with obvious benefits in portability and ease of use. The display is, of course, non-immersive and the tracking capabilities are less effective than the HMD system.

3.1. Principles of operation

The Batportal consists of a Compaq iPAQ running Linux, with a Lucent WaveLAN card and a Bat attached to the top of the device. The iPAQ has a 240×320 pixel colour touch-screen.

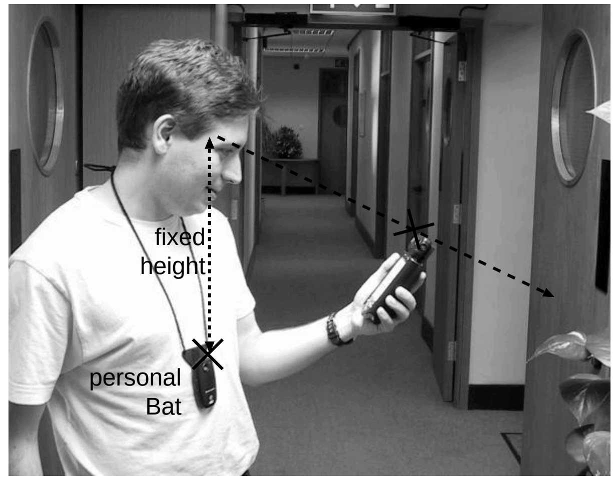


Figure 7. Direction vector

In use, the device is held at arm's length, rather like a magnifying glass (see Figure 7). The positions of the user's personal Bat and the Bat fixed to the handheld device are combined to form a direction vector in which the user is looking (this would, of course, be unnecessary for applications based purely on proximity). Augmentation information can then be rendered based on the user's location and direction of view, using the information in the sentient system's model. The magnifying glass analogy is very similar to that used by Rekimoto [16] in his NaviCam work. However, the Batportal is neither tethered nor dependent on detecting visual tags to determine its proximity to other objects since the sentient system already has a very detailed model of the environment. Figure 8 shows a typical screen display of a stylized monitor and loudspeaker.

The iPAQ is treated as a thin client, with applications running on a back-end workstation and the iPAQ being used simply as an I/O device. The iPAQ's display is accessed remotely using X11 across the WaveLAN; we have also tested a VNC [18] version which is less vulnerable to disruption caused by gaps in Wavelan coverage. The motivation behind accessing the display remotely is not due to any lack of CPU power, but is to make the endpoint stateless, reducing the effort required in maintenance.

Neither the user's Bat nor the Batportal is equipped with an inertial tracker, and so the user's position and viewing direction are only updated a few times per second, and are subject to Bat system errors. The Batportal's visual output

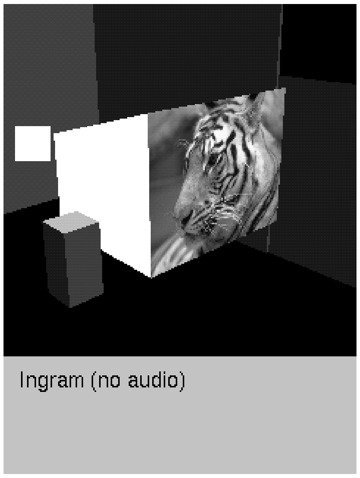


Figure 8. Batportal display

is therefore less smooth than that of the HMD. However, when the Batportal is held at arm's length the baseline between the two Bats is large enough to ensure that the angular error is reasonably small.

Display updates could either occur periodically, or be "interrupt driven". The latter case has been implemented as a "camera" mode. The user effectively takes a photograph of the virtual world from the current position by triggering via a button click. In practice we found that the rate of Bat position updates *was* sufficient for real-time, continuous updates. The registration tolerances for a non-immersive display are fairly relaxed, hence an automatic but time-lagged display was more compelling than manual snapshots. Therefore normal operation of the Batportal is real-time.

We assume that the user's eyes are a fixed distance (35cm) directly above their personal Bat. This is a good approximation, and varies little in practice. Any consistent offset is less important than random errors, because it produces a minor difference in viewing angle rather than causing display jitter. It has not proved necessary to calibrate the system for each user individually.

3.2. Registration

The Batportal's screen differs from the HMD in two significant respects: firstly it is not transparent, and secondly the viewing frustum is very narrow, particularly at arm's length. The lack of transparency is not a problem because the device is small, so the user can easily see the real world context around the screen. To overcome the narrow viewing angle we use a false perspective, giving a "fish-eye lens" effect (see Figure 9). Consequently, registration does not have to be precise, since objects are not seen as directly overlaid.

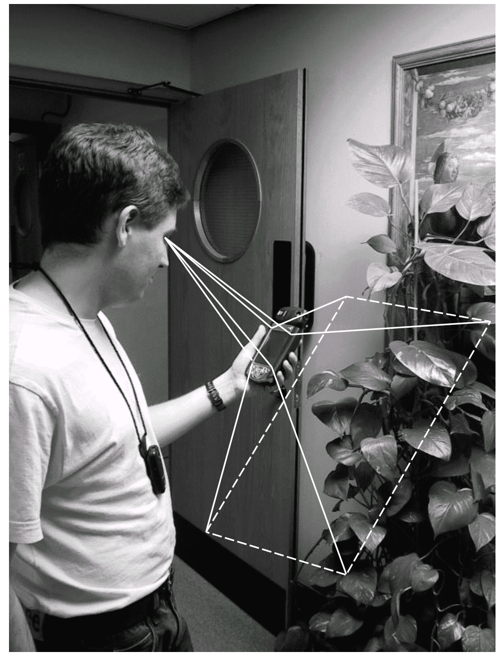


Figure 9. Viewing frustum

The viewing angle can be adjusted using the iPAQ's cursor keys. We have also implemented a mode in which the "magnification" can be continuously adjusted by holding the Batportal closer or further away from the eye.

3.3. Visualisation

The Batportal could be used to visualise the world in 3D (first person perspective), or in 2D (a plan view). We have concentrated on the more challenging 3D case, but it is expected that many practical applications will, in fact, be more comprehensible in 2D form. Other viewing modes which we anticipate include a straightforward list of "nearby" objects, and a plain-content view to describe single objects in more detail.

Further work is necessary to determine appropriate proximity conditions for a list view in various circumstances. For example we may wish to use a combination of the physical positions of objects provided by the sentient computing system, together with the relevance of each object to the user's current activity, in order to determine an ordering for objects that are "nearby".

In some applications it is useful to display several different data representations simultaneously. The small screen size of the Batportal precludes the use of split screens, and so we prefer insets and overlays when more than one type of information must be displayed. Figure 10(a) shows a semi-transparent overlay displaying a welcome message, for example.

3.4. Ownership

The Batportal is designed to be a tool which can be picked up and used immediately, with no configuration or inconvenient sign-on process. To signify temporary ownership of a Batportal, a user simply presses a button on the device. The device checks which person is closest to it (using information from the sentient system), and starts displaying the world from that person's point of view. The device could also adopt the new user's personal preferences at that point in time.

3.5. Audio

A general-purpose audio server runs on the Batportal. Mono 16-bit, 16 khz samples are generated and sent to it via the WaveLAN. A mixer runs on the iPAQ hardware to provide one-key access to mute and low volume settings (suitable for headphones).

Speech output has been added, using the Festival text-to-speech engine [5]. Festival runs continuously on the back-end machine, synthesizing utterances in its client-server mode to reduce per-invocation overheads. Speech is currently used to identify which room the user is in, and to provide feedback when the current user or mode changes. Status information that can be communicated aurally does not clutter the limited screen area, and makes the device more user-friendly.

An audio interface could be a distraction in a busy office environment, so for audio-intensive applications we use a lightweight single-sided earphone. For example it should be possible for the Batportal to narrate the subject lines of e-mails or answer queries about the environment even when the user is walking down the corridor or sitting in a meeting.

4. Applications

Wide area AR systems, like our HMD and Batportal systems, have many possible applications, including spatial annotation, navigation and remote control.

4.1. Annotations

Our sentient computing system provides a model of the world which includes objects such as computers, furniture, phones and personnel. Not only do we know the physical locations of these objects, but we have access to other properties and state, such as which people are visitors and whether a phone is on or off the hook. We can use the AR system to augment the user's view of these objects with annotated labels. Colour coding is used give cues to the state of the object they are labelling.

Annotation can be applied to fixed points in space (such as a room) or to moving tagged targets such as other people. This is a useful way of checking someone's name, office and perhaps common interests and so on. The sentient computing platform allows this information to be shared by all users of the AR system, whether they are using the HMD, Batportal or a traditional interface on a PC.

The Bat system does not provide sufficient accuracy for very fine registration between real and virtual objects, but it is sufficiently good to make it obvious which object is being referred to by each label. On the HMD system, the labels always the face the users, but are constrained to remain horizontal relative to the real world, which helps anchor the label to the object.

Figure 10(b) shows the annotation displayed by the Batportal for a gathering of people in a corridor. The positions of people in the augmented world are indicated by square icons, which are scaled based on their distance from the user. These are annotated with the person's name and their exact distance away (in metres).

4.2. Navigation

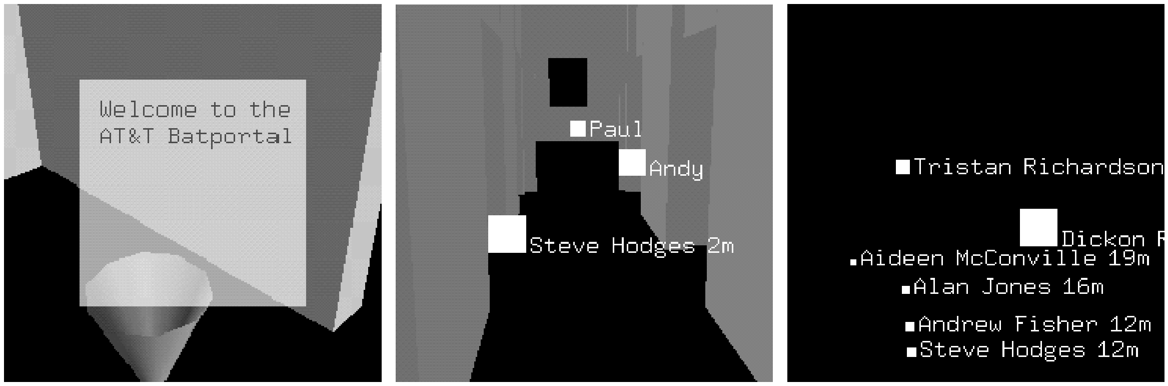
An interesting class of AR application involves navigation within buildings—these include finding one's way around, locating another person or following a personal augmented tour of the building.

We can use our AR systems, together with our sentient computing environment, to display the locations of people, walls, computers, telephones and other objects relative to the user. The level of augmentation can be varied to support the particular task that the user wishes to achieve. For example, suppose the Batportal system renders a 3D view of the current state of the building. Walls can be switched between opaque and transparent, giving the device an "X-ray vision" capability. Figure 10(c) shows how this allows the user to observe distant objects through interior walls. The user can also choose to display the structure of the entire building (see Figure 10(d)) or just the current floor.

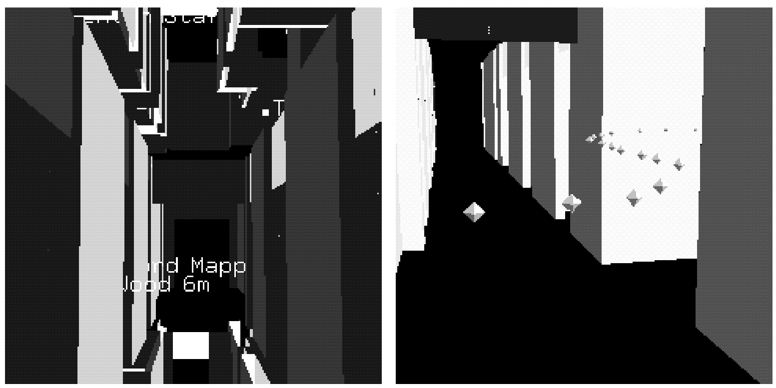
Navigation is possible using various means such as virtual signposts, a 2D map, compass arrows or turning signals. Virtual marker objects can be created by pressing a trigger button on the user's Bat (in the HMD system) or iPAQ (in the Batportal system), or by utilising a mode in which a virtual marker is automatically placed every half-second to create a trail, showing the route taken through the building by the user (see, for example, the Batportal view in Figure 10(e)).

4.3. World maintenance

A further application of the augmented visualisation is to verify that the model used by our sentient computing



(a) Information overlay (b) Gathering of people (c) Transparent walls



(d) Building structure (e) Trail markers

Figure 10. Screenshots

system is still correct. Over long periods of time, the positions of those (relatively static) objects which are not being tracked directly by the Bat system, such as computers and telephones, become stale, and housekeeping becomes a very tedious task. By entering an office with the AR system, and comparing the AR system's annotations with the real world, it is possible to tell at a glance whether old objects have moved or disappeared and if new ones have appeared. Ideally, it would then be possible to correct the database using an appropriate gesture-based or touchscreen-based interface.

4.4. Virtual buttons

The personal Bats worn by members of staff at AT&T Laboratories Cambridge have an easily detachable mount, which means they can be held and used as 3D pointing devices. We can then construct a 3D user interface that extends throughout the building, and which is analogous to a conventional 2D GUI driven by a mouse pointer. If a Bat is held up to a point in space that has some particular application-level significance, the command associated with that point in space can be invoked by the sentient computing system. An example might be a point near a scanner that, when "clicked" using a Bat, starts a scan and automatically forwards the resulting image to the user's mailbox.

Normally, these active points in space (known as *virtual buttons*) are physically labelled by a post-it note or poster. However, we can extend this interface within the personal space of the user of the HMD system by dispensing with the physical labels, and relying on the AR annotation of the physical point to indicate that a virtual button is present, and what that virtual button controls. This approach has the advantage of reducing the amount of visual clutter in the environment, and has proved to be practicable.

The optics of the HMD make the image appear at a fixed distance (approximately 4 feet) from the user. Rendered annotations appear to be anchored to the objects to which they are attached, whereas the position of purely virtual objects can appear to be ambiguous. To anchor a virtual button in space more effectively, we constrain it to lie on a flat surface such as a wall.

4.5. Interacting with devices

The AR system is considered primarily to be an output device, since in our experience any serious data entry is best done at a desk using a keyboard. However, the ability to display and interact with "virtual" remote control panels for other networked devices in the vicinity of the user, such as printers, loudspeakers, lights, VCRs and servers is also an interesting application [20]. An HMD or Batportal can display far more status information about queued jobs than the one-line LCD panel of a shared printer, for example.

Interaction with remote control panels could be achieved using a Bat in an HMD-based system (along the lines of the virtual buttons described above), or via the touchscreen of the Batportal's iPAQ. When used as a conventional PDA, the iPAQ supports a number of data entry interfaces using a stylus on the touchscreen. Use of a stylus was considered to be too inconvenient for a casual Batportal user, so we decided to make on-screen buttons large enough to press with a finger instead of the stylus.

4.6. Example scenarios

Environments in which we envisage AR systems being particularly useful include museums, trade shows, libraries, department stores, supermarkets and hospitals.

For example, in a supermarket a simple 2D map on a Batportal could assist with locating items and indicating routes, as well as highlighting special offers and items which have been purchased before. The screen could also be used to display prices, ingredients and recipes.

We can provide further motivation by considering a hypothetical museum example. Museums are attractive environments for AR systems, because the infrastructure only has to be installed once, after which it is unnecessary to physically label objects when exhibitions change. Meta data can be added directly to the virtual world in a way which is complementary to, but easier than, creating physical signs or guidebooks.

The AR systems described in this paper are personal devices, and so do not interfere with other visitors' experiences, and can be customised to take account of each user's age, language, interests and preferences. For example, one could request that the history of each painting be displayed on approach, the titles of modern art be withheld and any African sculpture nearby be highlighted. Furthermore a personal guided tour could be created with a different emphasis from the standard order of presentation.

The system could behave quite differently for children on a school visit than for ordinary visitors. Functions would include drawing attention to objects or aspects which the teacher considers important, or monitoring a "treasure hunt" for particular items (say three pictures which contain a mermaid, on discovery of which the students are rewarded with pop-up information to complete a worksheet). The teacher can readily check if the objectives have been completed, and co-operation is also possible, since routes to interesting places can be transmitted to peer AR systems.

5. Related Work

Rekimoto *et al.* have created an *Augment-able* Reality [17] which allows annotations to be attached to objects using the system itself. Their interface also has a personal clipboard area so that objects can be retained and moved around.

Feiner *et al.* used AR to visualise architectural components of a building [9] such as joists, beams and columns as well as load analyses of these components. Further work has demonstrated the scope of a hyperlinked annotation approach with the Touring Machine [19] and MARS [14] project.

Many of the scenarios we mention are similar to those proposed by Fitzmaurice in his Chameleon [10] project. We have developed wide-area systems with which we can begin to realise these scenarios.

Butz *et al.* have tackled the combination of personal *and* shared displays in their work on collaborative AR [6]. They include consideration of the difficult privacy issues associated with public screens. The system described leads to a more cluttered environment, however. We support the creation of multiple collaborative instances of both the HMD and BatPortal.

Curtis *et al.* in their work at Boeing [7] built a fully fledged AR application which aids workers in the assembly of aircraft wire bundles. They addressed the constraints of a noisy factory environment, ensuring that the equipment was both robust, and could be used intuitively by the workers.

6. Conclusions

At AT&T Laboratories we have developed a sensor-driven, or sentient computing system that can be deployed in buildings of any size. It incorporates a wide-area, ultrasonic tracking system that can be used to unobtrusively and accurately determine the positions and orientations of many different kinds of object. The system can respond to the location and state of objects in the real world, and environment data is immediately available for *sharing* by all users of the system, regardless of the form of their computing hardware and interfaces.

We have developed an Augmented Reality system around this sentient environment using two different types of endpoint: a head-mounted display and a handheld PDA. Both endpoints can be used whilst performing other activities, and the PDA at least is sufficiently lightweight and discreet to come close to meeting social acceptance criteria. We have developed prototype user interfaces, using a mixture of 2D and 3D graphics, speech and minimal or automatic (implicit) input methods.

The sentient environment is used every day by all 50 staff, and provides huge amounts of data describing the thousands of interactions which take typically take place. The Augmented Reality systems we have developed to exploit this data have enabled us to experiment with new ways in which we can visualise and interact with the world around us, and we intend to explore the most compelling applications of these technologies.

References

[1] M. Addlesee, R. Curwen, S. Hodges, A. Hopper, J. Newman, P. Steggles, and A. Ward. A Sentient Computing System. *IEEE Computer: Location-Aware Computing*, August 2001.

[2] W. P. Anton Fuhrmann, Dieter Schmalstieg. Fast Calibration for Augmented Reality. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 166–167, 1999.

[3] K. S. Arun, T. Huang, and S. D. Blostein. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, September 1987.

[4] R. Azuma. Tracking Requirements for Augmented Reality. *Communications of the ACM*, 36(7):50–51, July 1993.

[5] A. W. Black and P. Taylor. The Festival Speech Synthesis System: System Documentation. Technical Report HCRC/TR-83, University of Edinburgh, Scotland, University of Edinburgh, University of Edinburgh, Scotland, UK, 1997.

[6] A. Butz, T. Hollerer, S. Feiner, B. MacIntyre, and C. Beshers. Enveloping Users and Computers in a Collaborative 3D Augmented Reality, 1999.

[7] D. Curtis, D. Mizell, P. Gruenbaum, and A. Janin. Several Devils in the Details: Making an AR App Work in the Airplane Factory. In *First IEEE Workshop on Augmented Reality*, November 1998.

[8] M. K. Erik B. Dam and M. Lillhom. Quaternions, Interpolation and Animation. Technical Report DIKU-TR-98/5, University of Copenhagen, July 1998.

[9] S. K. Feiner, A. C. Webster, T. E. K. III, B. MacIntyre, and E. J. Keller. Architectural Anatomy. *Presence: Teleoperator and Virtual Environments*, 4(3):318–325, Summer 1995.

[10] G. W. Fitzmaurice. Situated Information Spaces and Spatially Aware Palmtop Computers. *Communications of the ACM*, 36(7):38–49, July 1993.

[11] E. Foxlin. IS-600 Mark 2 Precision Motion Tracker, http://www.isense.com/products/prec/is600/index.htm.

[12] E. Foxlin, M. Harrington, and G. Pfeifer. Constellation: A Wide-Range Wireless Motion-Tracking System for Augmented Reality and Virtual Set Applications. In *ACM SIGGRAPH Conference Proceedings*, pages 371–378, 1998.

[13] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The Anatomy of a Context-Aware Application. In *Mobile Computing and Networking*, pages 59–68, 1999.

[14] T. Hollerer, S. Feiner, T. Terauchi, G. Rashid, and D. Hallaway. Exploring MARS: Developing Indoor and Outdoor User Interfaces to a Mobile Augmented Reality System. *Computers & Graphics*, 23(6):779–785, December 1999.

[15] E. McGarrity and M. Tuceryan. A Method for Calibrating See-Through Head-Mounted Displays for Augmented Reality. Technical Report TR–CIS–0499-13, Indiana University Purdue University Indianapolis, April 1999.

[16] J. Rekimoto. NaviCam: A Magnifying Glass Approach to Augmented Reality. *Presence: Teleoperator and Virtual Environments, Vol. 6, No. 4, pp. 399-412, August 1997*, 6(4):399–412, August 1997.

[17] J. Rekimoto, Y. Ayatsuka, and K. Hayashi. Augmentable reality: Situated Communications through Physical and Digital Spaces. In *Proceedings of the 2nd International Symposium on Wearable Computers*, pages 68–75, October 1998.

[18] T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper. Virtual Network Computing. *IEEE Internet Computing*, 2(1):33–38, January/February 1998.

[19] S.Feiner, B. MacIntyre, and T. Hllerer. A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment. In *Proceedings of the International Symposium on Wearable Computers*, pages 74–81, 1997.

[20] R. Want, B. N. Schilit, N. I. Adams, R. Gold, K. Petersen, D. Goldberg, J. R. Ellis, and M. Weiser. An Overview of the PARCTAB Ubiquitous Computing Experiment. *IEEE Personal Communications*, 2(6):28–33, Dec 1995.

[21] M. Weiser. Ubiquitous Computing. *IEEE Computer 'Hot Topics'*, 26(10):71–72, October 1993.

7. Acknowledgements

The authors would like to extend their thanks to Andy Ward, Steve Hodges, Mike Addlesee, who were responsible for the design, construction and deployment of the Bat system. We would also like to express our gratitude to Pete Steggles and Rupert Curwen who designed and developed the sentient framework.