Location based Applications for Mobile Augmented Reality

Gerhard Reitmayr **Dieter Schmalstieg**

Vienna University of Technology
Favoritenstr. 9-11/188/2,
A1040 Vienna, Austria
Email: {`reitmayr|schmalstieg`}`@ims.tuwien.ac.at`

Abstract

In this work we investigate building indoor location based applications for a mobile augmented reality system. We believe that augmented reality is a natural interface to visualize spacial information such as position or direction of locations and objects for location based applications that process and present information based on the user's position in the real world. To enable such applications we construct an indoor tracking system that covers a substantial part of a building. It is based on visual tracking of fiducial markers enhanced with an inertial sensor for fast rotational updates. To scale such a system to a whole building we introduce a space partitioning scheme to reuse fiducial markers throughout the environment. Finally we demonstrate two location based applications built upon this facility, an indoor navigation aid and a library search application.

Keywords: Augmented Reality, Mobile Computing, Wearable Computing, Location based Computing, Tracking.

1 Introduction

Augmented reality (AR) is a powerful user interface technology that augments the user's environment with computer generated entities. Azuma points to three important aspects that define AR (Azuma 1997). It blends the real and virtual within a real environment, is real-time interactive and registered in 3D. In contrast to virtual reality which completely replaces the real world, augmented reality displays virtual objects and information along with the real world registered to real world locations.

Location based systems take the user's location into account when processing and presenting information to the user. While augmented reality systems can be viewed as falling into this category, location based systems become interesting when the supported range of locations expands beyond a single laboratory room. There is a wealth of work regarding such types of applications within the wearable and ubiquitous computing area. Both can make good use of augmented reality to visualize abstract and spacial information as described in (Starner, Mann, Rhodes, Levine, Healey, Kirsch, Picard & Pentland 1997).

To employ AR in a large environment mobile systems were built that support the graphical and computational demands of it. Examples of such developments are the Touring machine by (Feiner, MacIntyre, Höllerer & Webster 1997) or the Tinmith system by (Piekarski & Thomas 2001). The Touring machine is also a good example of a location based application that presents the user information about buildings they are looking at.

Figure 1: A user navigates through a building guided by the heads-up display graphics providing him with direction.

Most AR based applications so far either relied on high quality tracking with a small environment such as a laboratory room with dedicated tracking equipment or on GPS technology that delivers positional information world wide but works only outdoors and with low accuracy and update rate. There are just a few solutions such as the BAT system (Addlesee, Curwen, Hodges, Hopper, Newman, Steggles & Ward 2001) that allow position tracking indoors covering a large environment with reasonable accuracy.

We are interested in visualizations of spacial data using augmented reality. To investigate such applications for a mobile user we developed a mobile AR system (Reitmayr & Schmalstieg 2001*a*). A natural step beyond the research lab is the building we are residing in. Therefore we developed and deployed a tracking system to support building wide location

---

based applications and that would deliver reasonable quality. This then became the basis of two location based applications we present in this work.

The first application is a navigation guide called Signpost that supports the user in finding a desired location within our building. It provides her with directional hints and highlights exits of rooms to be taken to proceed towards the destination. The second application is a a search and retrieval application that helps a library visitor to locate a desired book or its designated place in the library bookshelves.

The remainder of the paper is structured as follows. The next section will describe related work in the area of tracking and navigation applications for AR systems. Then we give a short overview of our mobile AR platform. Next we describe our tracking system in detail and follow with a description of our demonstration applications. We finish with a discussion of the results and some outlook to future work.

2 Related work

2.1 Tracking

Location tracking is a prerequisite for any location aware application. In order to be able to provide the user with services and information related to her location, a system needs to sense or otherwise be told the current location of the user. Augmented reality applications require a very accurate position tracking to register visual information accurately with the user's environment. Otherwise the augmented information may be positioned incorrectly resulting in a confusing or unusable user interface.

There is a wealth of work related to position tracking ranging from indoor systems covering a room size area to outdoor systems supporting the entire planet. This diversity is also present in the accuracy of tracking systems ranging from millimeters to several meters.

Outdoors, GPS provides global position with accuracy between several meters and centimeters depending on additional techniques such as broadcasting correction signals or using the phase information of the received signals to improve the accuracy. However, GPS requires a direct line of sight to several satellites and therefore is not working properly inside buildings or in areas covered by trees or tall buildings (appropriately termed 'city canyons').

Indoors, tethered tracking systems using magnetic (Raab, Blood, Steiner & Jones 1979), ultrasonic (Intersense 2002) and optical technologies (Welch, Bishop, Vicci, Brumback, Keller & Colucci 2001) achieve high accuracy in the millimeter to centimeter range. These systems are typically able to cover a room and require installations of large devices or dense arrays of beacons or sensors mounted in the covered area. Another research system (Addlesee et al. 2001) can cover a whole building but is not available to the public.

A large class of tracking solutions uses computer vision to track the movement of a camera. Some solutions place fiducial markers (Rekimoto 1998) in the environment to achieve good accuracy. There is also experimental work that uses marker free vision based tracking by selecting salient natural features in the environment (Pinz 2001) or comparing the camera's view with prerecorded imagery (Kourogi & Sakaue 2001).

Other approaches try to use local sensors and dead reckoning techniques to compute a user's location (Lee & Mase 2001). However these are prone to accumulation of subsequent errors in their computations unless they are synchronized with absolute positioning systems. Including knowledge about the environment in the form of geometric models and accessibility graphs (Höllerer, Hallaway, Tinna & Feiner 2001) allows increasing the accuracy of such approaches significantly.

Our contribution is relying on using a set of trained fiducial markers together with additional knowledge about the environment. This allowed us to improve the performance of the optical tracking component by reducing the number of markers required for a stable operation of the system. A similar solution for indoor tracking has been attempted (Thomas, Close, Donoghue, Squires, Bondi, Morris & Piekarski 2000), however without an optimization of the reuse of markers.

2.2 Navigation applications

Augmented reality as a user interface for mobile computing is particularly powerful when the computer has access to information on location and situation, so it can provide contextual information. A prominent example that served as an inspiration to our approach is Columbia's "Touring Machine", which is used as a mobile multimedia information system (Feiner et al. 1997) and journalistic tool (Höllerer & Pavlik 1999). Other mobile applications include tourist information tools (Satoh, Hara, Anabuki, Yamamoto & Tamura 2001) and model constructing applications for indoor (Baillot, Brown & Julier 2001) or outdoor use (Piekarski & Thomas 2001).

Examples of navigation applications are also found. Some of them concentrate on outdoor use and rely on GPS tracking. They display such information as direction and distance to the next way point of a path (Thomas, Demczuk, Piekarski, epworth & Gunther 1998), or directions of objects that are augmented with further information (Suomela & Lehikoinen 2000). Indoors there are applications that provide models and directional hints for navigation (Höllerer et al. 2001) and offer also additional information such as visualizing the path a user has taken (Newman, Ingram & Hopper 2001).

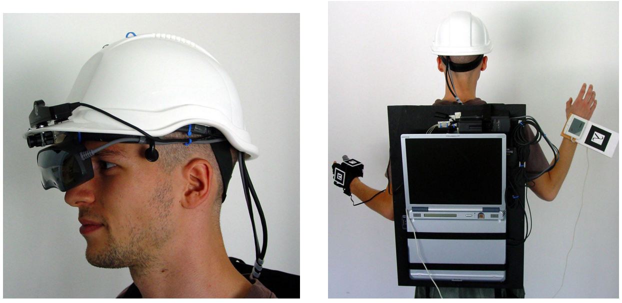
3 The mobile AR setup

While the computational power for stereoscopic rendering and computer vision is becoming available in mobile computer systems, the size and weight of such systems is still not optimal. Nevertheless, our setup is solely build from off-the-shelf hardware components to avoid the effort and time required for building our own (Reitmayr & Schmalstieg 2001*a*). On one hand this allows us to quickly upgrade old devices or add new ones and to change the configuration easily. On the other hand we do not obtain the smallest and lightest system possible.

3.1 Hardware

Our current setup uses a notebook computer with a 2Ghz processor and a powerful NVidia Quadro4 graphics accelerator. It operates under Windows XP. We also added a wireless LAN network adapter to the notebook to enable communication with our stationary setup or a second mobile unit. The equipment is mounted to a backpack worn by the user. As an output device, we use a Sony Glasstron see-through stereoscopic color HMD. The display is fixed to a helmet worn by the user. Moreover, an InterSense InterTrax2 orientation sensor and a web camera for fiducial tracking are mounted on the helmet (see Figure 2).

The main user interface consists of two self made pinch gloves and a wrist mounted touch pad. These



a) b)

Figure 2: The mobile AR setup. (a) The helmet with an HMD, a miniature camera and an inertial sensor mounted to it. (b) The notebook computer and peripheral devices mounted on a backpack.

are equipped with optical markers to be tracked by the head mounted camera (see Figure 3). We constructed a rigid frame to mount three markers around the wrist to allow a wide range of hand postures to be reliably tracked.

As gloves we use robust finger-free bicycle gloves and cover only the thumb with two layers of cotton that have a pressure sensitive foil embedded at the inner side tip of the thumb. The pressure sensor acts as a button. Due to the relatively fixed offset from the back of the hand to the thumb tip when thumb and index finger are pressed together, the 3D cursor can easily be calculated. Four fingers are left free to manipulate physical objects and to operate the touch panel that does not respond to skin covered in cloth. This design is similar to the work of Piekarski and Thomas (Piekarski & Thomas 2002). However we focus on 3D interaction and require a larger range of postures to be tracked. We also use only one button as we do not rely on a complex menu system.

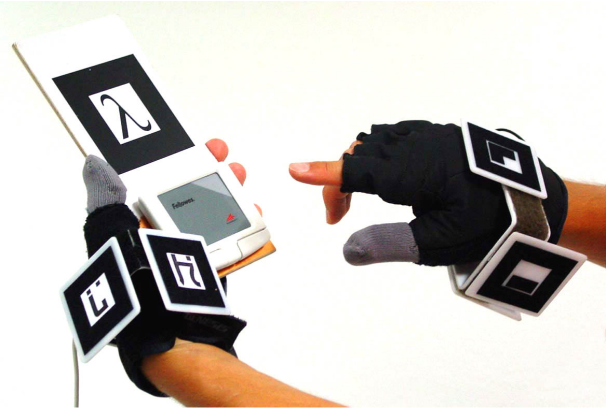


Figure 3: The interaction props of our setup. The gloves are equipped with 3 markers to be able to track the hands in a wide range of orientations.

The touch panel is mounted to the wrist with an elastic band, and also has a marker attached. While the graphical overlay of the panel is only presented when it is in the users field of view, the panel operation itself does not rely on the tracking of the panel marker, and can be done while looking at something else.

3.2 User interface management software

As our software platform we use Studierstube (Schmalstieg, Fuhrmann, Hesina, Szalavari, Encarnao, Gervautz & Purgathofer 2002), a user interface management system for AR based on, but not limited to stereoscopic 3D graphics. It provides a multi-user, multi-application environment, and supports a variety of display devices including stereoscopic HMDs. It also provides the means of 6DOF interaction, either with virtual objects or with user interface elements registered with and displayed on the pad.

Applications are implemented as runtime loadable objects executing in designated volumetric containers, a kind of 3D window equivalent. While the original stationary Studierstube environment allowed a user to arrange multiple application in a stationary work-space (see Figure 4 for a more elaborate example), our mobile setup allows to arrange 3D information in a wearable workspace that travels along with a user. It supports different presentation mechanisms that have been identified as useful for AR (Feiner, MacIntyre, Haupt & Solomon 1993, Billinghurst, Bowskill, Morphett & Jessop 1998): *Head-stabilized*, where information is fixed to the users viewpoint, *Body-stabilized*, where information is fixed relative to the users body position, and *World-stabilized*, where information is registered with real world locations. In this setup we do not distinguish between the user's head and body.

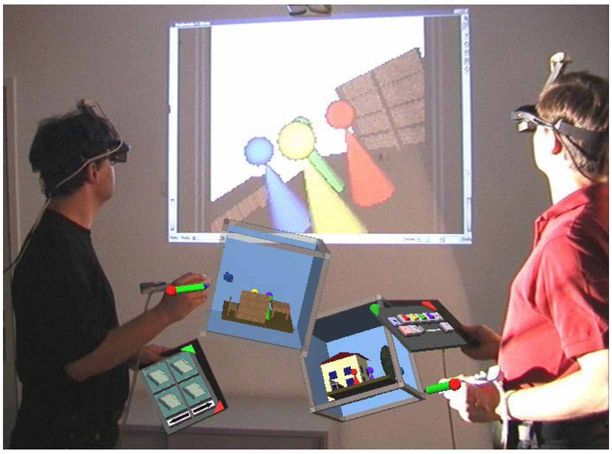


Figure 4: Two users working in a shared workspace on an application to design stage sets. They are working with virtual geometry as well as projections of special views of the environment. The Studierstube framework provides 3D windows and widgets and supports multiple applications and output devices.

3.3 Tracking support

Mobile AR requires significantly more complex tracking than a traditional VR application. In a typical VR or AR application tracking data passes through a series of steps. It is generated by tracking hardware, read by device drivers, transformed to fit the requirements of the application and send over network connections to other hosts. These tasks are handled by the OpenTracker library (Reitmayr & Schmalstieg 2001*b*), an open software architecture for the different tasks involved in tracking input devices and processing multi modal input data.

The main concept behind OpenTracker is to break up the whole data manipulation into these individual steps and build a data flow network of the transformations. The framework's design is based on XML, taking full advantage of this new technology by allowing the use of standard XML tools for development, configuration and documentation.

OpenTracker uses a vision tracking library called ARToolkit (Billinghurst & Kato 1999) to implement the tracking of the fiducial markers on the interaction props or in the environment. It analyzes the video images delivered by the web camera mounted to the helmet and establishes the position of known markers relative to the camera. This information is then used to either track the interaction props such as gloves and the tablet relative to the user or to track the user within the environment.

4 Indoor tracking

To build an environment where we could test drive our mobile AR kit, we implemented an indoor tracking solution to cover a floor of our building. As we did not have access to a proprietary building-wide positioning infrastructure (such as AT&T Cambridge's BAT system used by (Newman et al. 2001)), we choose to rely on a hybrid optical/inertial tracking solution. This approach proved very flexible in terms of development of positioning infrastructure, but also pushes the limits of what ARToolkit tracking can provide.

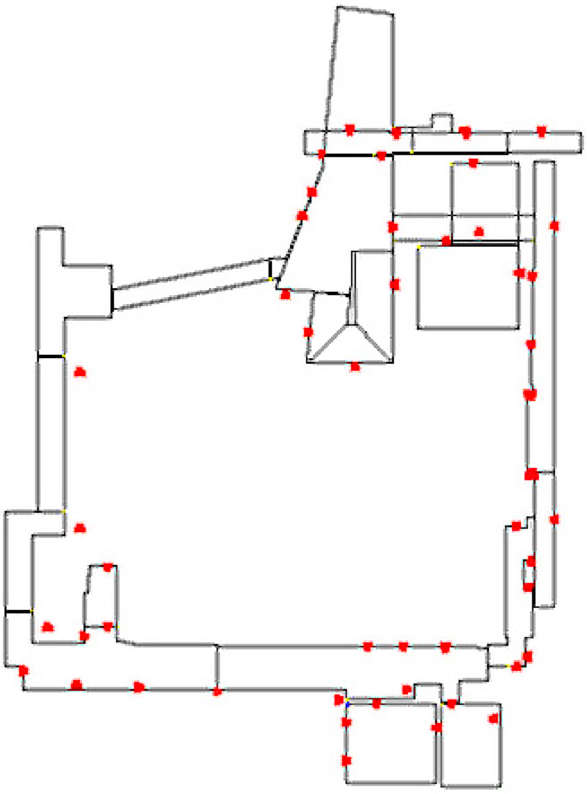


Figure 5: This diagram shows the geometric model of our floor. The (red) dots denote the locations of measured markers used to track the user within the environment.

To implement a wide area indoor tracking solution we resolved to use a set of well-known markers that were distributed in the environment. Together with a geometric model of the building that includes the location of the well-known markers (see Figure 5) we can compute the user's location as soon as a marker is tracked by the optical tracking system. The model is structured into individual rooms and the connections between these rooms called portals. The location of a room within a world coordinate system completes the model. These models and the location of the markers were obtained by tape measurements.

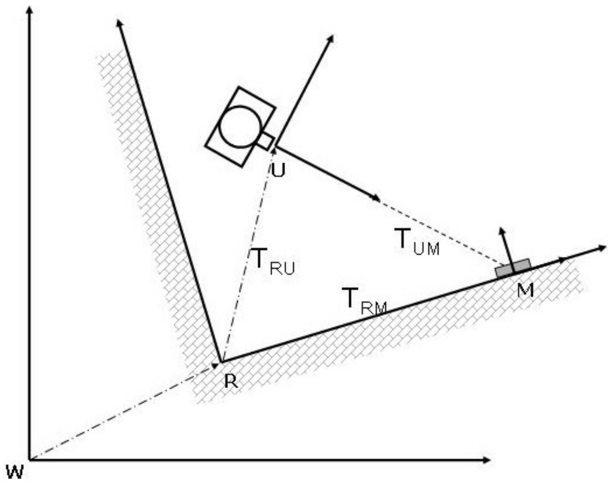


Figure 6: This diagram shows the different coordinate systems involved to compute the user's position. U is the user's position and orientation, R the room coordinate system, M a markers position and orientation within that room and W the world or model coordinate system.

A measurement of a marker by the optical tracking returns the markers position and orientation T_{UM} within the user's coordinate system U. This is essentially the transformation to convert coordinates from the system U into the system M. Inverting T_{UM} gives the user's position and orientation within the marker's coordinate system M as $T_{MU} = T_{UM}^{-1}$. By combining the fixed and measured transformation T_{RM}between the room coordinate system R and the marker's M with the value T_{MU}, we calculate the user's position and orientation $T_{RU} = T_{RM} \cdot T_{MU}$ within the room coordinate system R. This information was then used to compute the direction of the indication arrow and to render the augmentation of portal and the room's geometry.

In addition to a geometric model of the environment we also constructed an adjacency graph for the individual rooms (see Figure 7). This graph was used to provide the path finding functionality described in section 5.1 and to optimize the use of markers as described in the next section.

4.1 Marker reuse

The implemented tracking approach requires a large set of markers. It is necessary to place a marker about every two meters and to cover each wall of a single room with at least one marker. Deploying this in our floor covering about 20 rooms and long hallways would require several hundred different markers. However, marking up a large indoor space with unique ARToolkit markers is not feasible for two reasons:

- The more markers, the higher the degree of similarity of markers will be. Near rotational symmetry can become a major problem when a large amount of markers is used. Additionally lighting conditions vary often from one room to another. All this leads to inferior recognition accuracy. For a larger set of markers this implies a higher number of false recognitions.

- A large set of markers makes the search space that ARToolkit has to traverse larger, leading to significant decrease in performance.

As a consequence, it is not possible to scale the use of ARToolkit to arbitrary large marker assemblies.

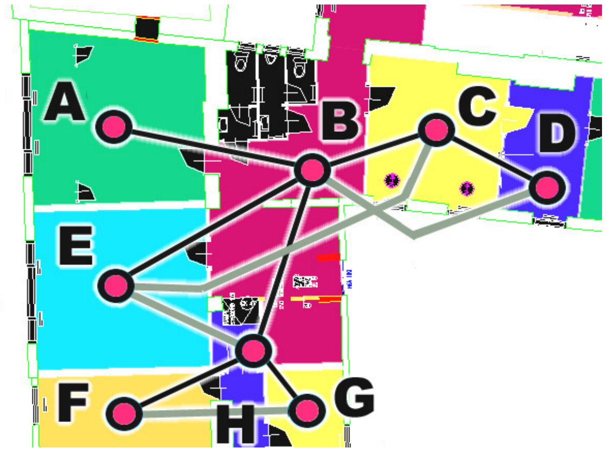


Figure 7: A part of the floor plan of our building with rooms colored differently. The resulting adjacency graph is overlayed with black edges giving portals between rooms and grey edges giving line-of-sight connections.

To overcome this restriction, we developed a spatial marker partitioning-scheme that allows reusing sets of markers within the given environment. The idea behind this approach is that, if the tracking system knows the user's location, it can rule out large parts of the building because they will not be visible to the camera. Therefore, for two areas, which are not visible to each other, it becomes possible to use the same set of markers. This problem is equivalent to approaches for indoor visibility computation based on potentially visible sets (Airey, Rohlf & Brooks 1990).

We use the room definition in the geometric model as the basic element of this approach. Then we can build an adjacency graph for all rooms using rooms as nodes and portals between rooms as edges. In addition to the portals representing logical connections we also add further edges that represent a line-of-sight between to rooms (see Figure 7). We call this second graph the extended adjacency graph.

The next step is to generate a minimal number of disjoint sets of markers, such that each room is assigned to a set of markers. This is similar to graph coloring problems where a minimal set of colors is used to color nodes in a graph with certain restrictions. The following constraints must be fulfilled to yield a useful distribution of marker sets.

- Two nodes connected in the extended adjacent graph must have disjoint sets of markers. Otherwise the camera tracking system cannot decide which room the marker belongs to, if both instances are visible from one room.

- Looking at one node all adjacent nodes in the extended adjacent graph must have disjoint marker sets. This constraint has to be fulfilled by all nodes in the extended adjacency graph. Otherwise the common node provides a point of view that allows sighting of a marker in two different places.

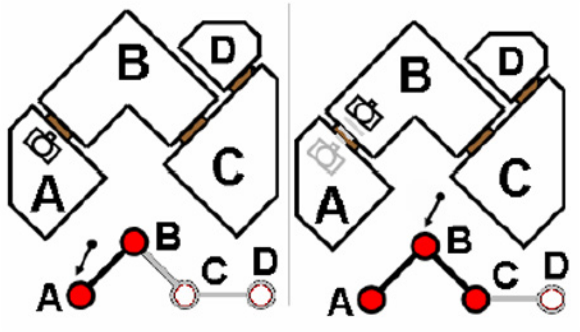


Figure 8: This figure shows the change of the tracking system when a user walks from one room into another. On the left, the user is in room A and therefore the tracking system only tracks markers of room A and B. When the user moves to room B, it also starts to track markers of room C.

Starting with a given correct room position, the system tracks the user within the current room and into neighboring rooms. It can detect a room change by comparing tracked markers with the sets of markers assigned to the current room and its neighbors. If it recognizes a marker from a set of a neighbor room it assumes that the user entered this room, which then becomes the current room. Figure 8 gives a small example of the algorithm.

The left image shows that the current node A and its adjacent node B are marked active, when the user is in room A, while node C and D are inactive (gray). As soon as the user moves to room B a room change is triggered and the state of the tracking system is updated. Now the current room is room B and the status of both adjacent rooms A and C are switched to active. Note that room D is still inactive, because there is no line of sight between room B and room D. We could also reuse markers of room A in room D as these rooms are not common neighbors of either room B or C.

We extended the scheme by placing a single globally unique marker in every room, which can be used to reinitialize the system in case of an error. It is also suitable for the determination of the current room at start up. The user just has to turn around until the system detects the unique marker and switches into interactive mode. This unique marker set is disjoint from any reusable marker set.

4.2 Inertial Tracker Fusion

Our mobile AR kit is also equipped with an inertial orientation tracker providing low latency updates on the user's head orientation. This information is used to update the user's view direction in between measurements from the optical tracking system. However the tracking information is subject to drift that can lead to large errors after a short period of time. Therefore we need to correct the measurements of the inertial tracker. This is implemented similar to (Newman et al. 2001).

Every time we receive a measurement by the optical tracking system, we compute the user's true head orientation $q_{optical}$ as described in section 4. Then we compute a correction orientation for the measured

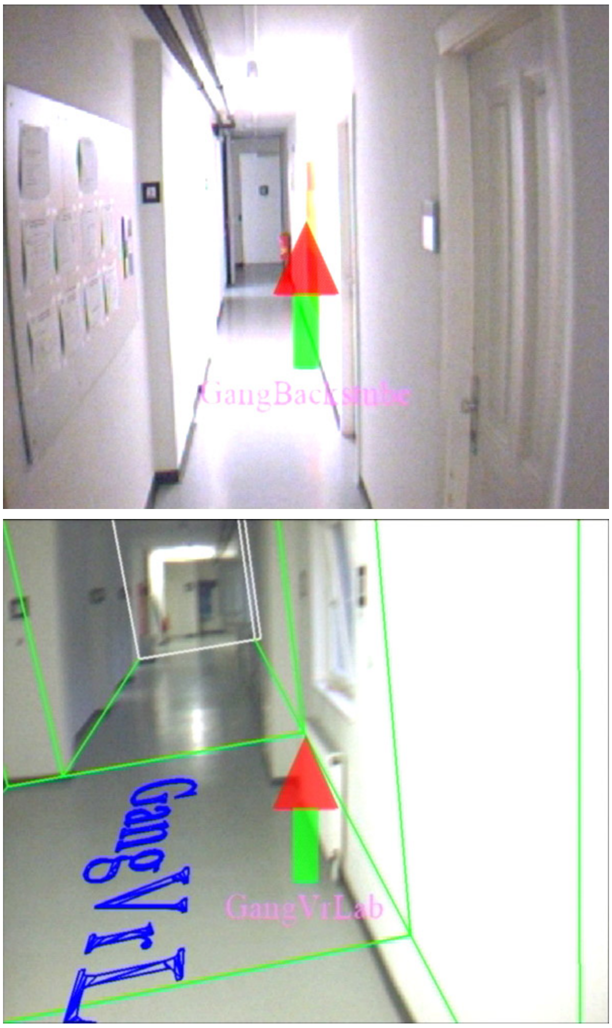


Figure 9: (Top) The direction is indicated by an overlayed arrow. (Bottom) The wire frame augmentation of the room and the portals can be switched on.

inertial orientation $q_{inertial}$ as $q_{correction} = q_{optical} \cdot q_{inertial}^{-1}$. This correction is than applied to any subsequent measurement of the inertial tracker to provide a correct user orientation $q_{user} = q_{correction} \cdot q_{inertial}$.

5 Applications

As examples of location-based applications we chose two typical tasks that appear in an academic building. Firstly, we implemented a navigation application called Signpost that guides the user to a selected destination room. Secondly, we augmented a library by highlighting locations of books within the bookshelves.

5.1 Signpost - navigation application

Many visitors to our institute have a hard time to find their way through the complicated and poorly labelled corridors of our office building. We considered this situation a useful test case for a location-based service provided through our mobile augmented reality system. The system should guide a user on the way through the building with direction and map based contextual information.

User interface

Interaction with the application is attained through a user interface displayed on the wrist pad. The user has to select the desired destination room, either by clicking into a world in miniature model or by selecting its description from a list. Then the application computes the shortest path from the current room to the destination room (see section 5.1) and gives appropriate feedback. The user may change the destination room again at any moment.

The system continuously provides the user with two modes of visual feedback: a heads-up display with directional arrows and a world in miniature model of the environment. These two information displays are described now in more detail.

Via the HMD a wire frame model of the current room is superimposed on top of the real scene. The application uses a shortest path search on an adjacency graph of the building to determine the next door/portal on the way to the destination. The doors/portals are then always highlighted in white. The wire frame overlay can optionally be turned on and off by the user. In addition, a direction arrow shows the direction to the next door or portal (see Figure 9), indicating either to move forward, turn left/right or to make a U-turn.

In addition to that the application presents a world in miniature model (WIM) of the building to the user on the augmented wrist pad (see Figure 10). It always shows a full miniature view of all rooms on the floor in order to allow the user to determine his current position in the building, which is highlighted in yellow. Additionally the path to the selected destination room is highlighted in cyan, while the destination room itself is shown in red. Using such a WIM model to help users understand virtual environments better was described by (Stoakley, Conway & Pausch 1995).

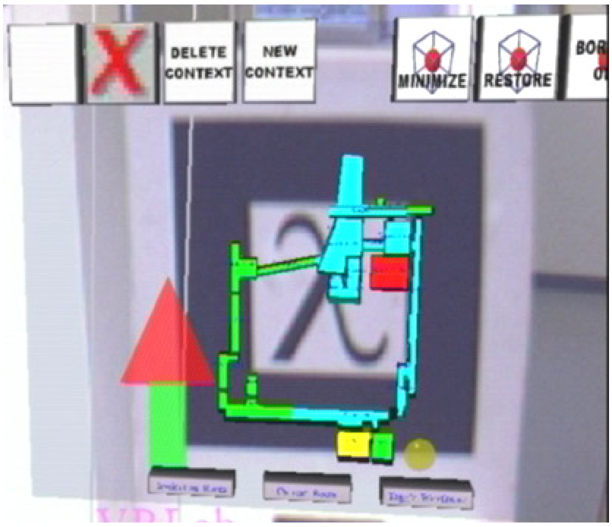


Figure 10: The world in miniature model shown on the pip. The found path to the destination room is highlighted.

As the wrist pad is tracked, the user can bring the WIM model into her view by moving her arm with the wrist pad in front of her eyes. As a consequence, the WIM model does not permanently cover the screen, and the user can rather decide herself when she wants the floor plan to be displayed as well as she can adjust the size by moving her arm nearer to her eyes or farther away.

Path finding

The Signpost application uses a path finding component for computing the shortest path from the user's current room to the desired destination. It uses the adjacency graph from the tracking system and also the information about the room the user is currently in.

Using this adjacency graph, the application is able to compute the shortest path between two rooms. This is done by utilizing an algorithm based on the shortest path algorithm introduced by Dijkstra (Dijkstra 1959). The information about the path is then used to highlight the rooms along the way on the WIM display and to augment the portal that leads in to the next room from the current one. The portal information is also used to compute the direction of the overlayed arrows which need to point towards that portal.

Although the application could recalculate and update the shortest path several times per second, a new path is only calculated when a room-change occurs (which is triggered whenever the tracking system recognizes a marker of a neighboring room to the current room). As the application always adapts the shortest path starting from the current room, even if the user walks the wrong way, a new shortest path based on the user's current position is proposed.

5.2 ARLib - Library information application

ARLib aims to aid the user in stereotypical tasks that are of use in a library by augmenting a book's position on a shelf. It features two main modes of use.

Assistance in searching for a book

The user is interacting with the application via a user interface presented on the touch pad. First, one of the search options like searching by title, author, etc., must be selected. Next, text is entered by using either a virtual keyboard or a graffiti pad (see Figure 11). A list of books matching the search is returned and displayed. The user then selects the book she is looking for. Now the corresponding book's position on the shelf and the shelf itself are highlighted in the heads-up display to aid the user in finding the book (see Figure 12). Also, all available information about the publication is displayed on the wrist panel.

a) b)

Figure 11: User interface for the ARLib application. (a) The user may select one of different search modes. (b) Input of search terms using graffiti text input.

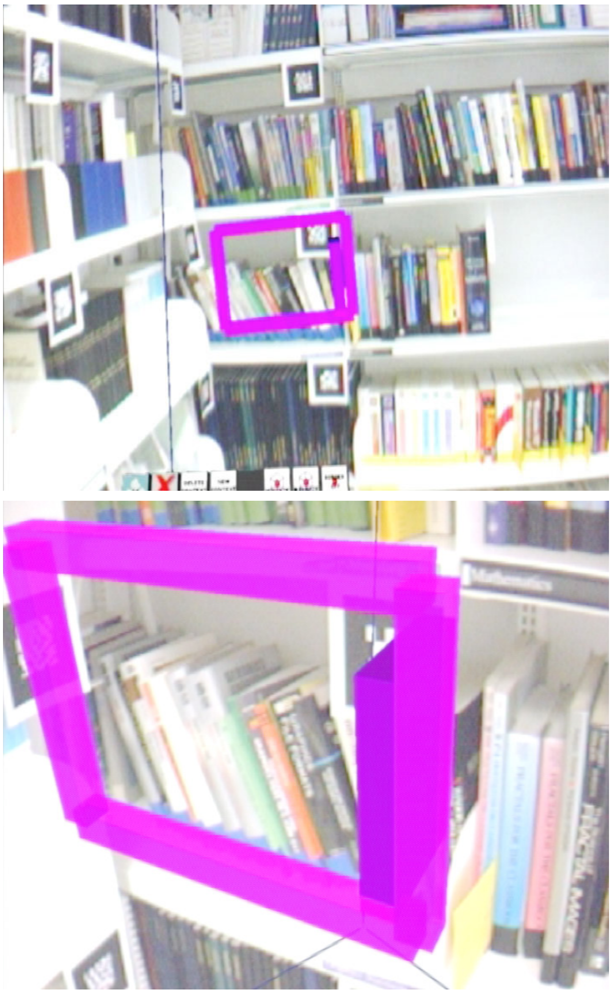


Figure 12: Augmented position of a book as seen by the user. These images were produced by overlaying the tracking camera's video image by the rendered graphics.

Assistance in returning a book to the right shelf

In this mode, ARLib attempts to detect markers that are attached to books. If a marked book is spotted, all available information about the publication is presented on the wrist panel, and the book's designated position on the shelf is highlighted to aid the user in returning the book to its correct position. This enables the user to simply look at the book in her hand and trigger the applications behavior.

Implementation

ARLib maintains a model of the library's shelves and the positions of the individual books within the shelves. Furthermore meta information about the books is stored such as the title, authors, category and keywords associated with it. Using the same tracking technology as described in section 4 the application continuously tracks the user's position within the library. In addition to that a designated set of markers is attached to the books themselves to allow tracking of their positions as well.

Depending on the state of the application different parts of the model are rendered. If the location of a book is augmented, its containing shelf is rendered as well as an augmented view of the book itself within

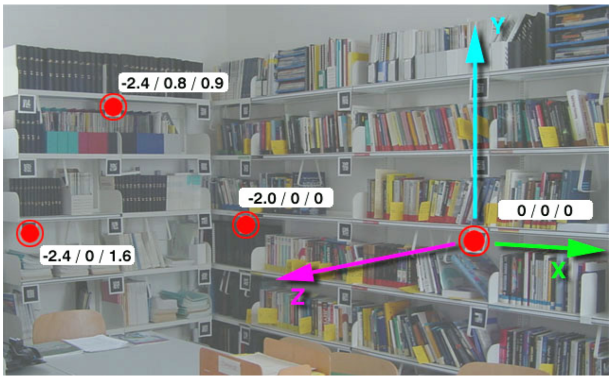


Figure 13: Markers are mounted to the book shelves and registered.

the shelf. If the book itself is to be returned to the library the application highlights the book to signal the user that it identified it correctly.

To use ARLib with a real-life library, some preparation work had to be done. ARLib's books database is an OpenInventor scene graph file automatically created from a proprietary library database. The shelf geometry and marker positions must be surveyed after attaching the markers to the shelves to register shelf information from the database with the geometric model (see Figure 13).

To test ARLib, we used our in-house library for which a database of roughly 650 publications in 13 categories on 20 shelves was available. 62 markers were created, of which 52 were placed at shelf joints, 8 on book covers and 2 large ones on the walls to improve tracking from a distance.

6 Results

In this work we described a simple tracking solution for wide area indoor tracking. It tries to fill the gap between high accuracy tracking solutions that only cover a relatively small area and outdoor positioning techniques that are not applicable indoors. Furthermore we implemented two example applications that build upon this development.

The tracking error varies largely. As long as markers were continuously visible the augmentation of the room geometry is fairly stable and correct. However as soon as all markers are obscured the tracking can only rely on the inertial tracker and starts to become incorrect. In this case the augmentation is not useable anymore, because if the overlay is off by a few centimeters the interpretation of the information is more difficult for the user.

This led to the conclusion that more markers are needed to improve the quality of the tracking. The directional arrows also performed better than the highlighting of the portals under such circumstances as the information they provide is less prone to tracking errors. In the future we want to automatically switch between these different presentations based on the estimated tracking error as a form of Level of Error filtering (MacIntyre & Coelho 2000).

Both applications were developed as part of a lab exercise by groups of two or three students. This encouraged us that the general platform and the framework provided by the Studierstube simplifies the development of such applications.

7 Future work

For future work we plan to extend the area covered by the optical tracking. Both applications were deployed in our building on a scale that exceeded the typical lab environment. However they still not cover a desirable area such as a whole building or a typical university library. We plan to extend the navigational application to cover at least two floors of our building which amounts to about 4 times the size of the current installation. Also the number of markers should be increased to improve the quality of the tracking. The surveying will be redone with a more accurate professional surveying tools to provide a more detailed and correct model of the building.

The general indoor location information will be provided as a service within the framework so that future applications can directly leverage on this information and must only concentrate on their functionality. We also plan to cover the same area with a wireless LAN network to enable research into collaborative scenarios using multiple AR setups.

Acknowledgements

This work was sponsored by the Austrian Science Foundation under contracts no. P14470-INF, P14470-N04 and START Y193, and Vienna University of Technology by an infrastructure lab grants ("MARDIS"). Many thanks go to Michael Kalkusch, Michael Knapp, Thomas Lidy, Eike Umlauf and Harald Piringer for their inspired work on the implementation of the example applications.

References

Addlesee, M., Curwen, R., Hodges, S., Hopper, A., Newman, J., Steggles, P. & Ward, A. (2001), 'A sentient computing system', *IEEE Computer: Location-Aware Computing* .

Airey, J., Rohlf, J. & Brooks, F. P. (1990), Towards image realism with interactive update rates in complex virtual building environments, *in* 'Proceedings of the 1990 Symposium on Interactive 3D Graphics', pp. 41–50. not read.

Azuma, R. T. (1997), 'A survey of augmented reality', *Presence* Teleoperators and Virtual Environments 6(4), 355–385.

Baillot, Y., Brown, D. & Julier, S. (2001), Authoring of physical models using mobile computers, *in* 'Proc. ISWC 2001', IEEE, Zurich, Switzerland, pp. 39–46.

Billinghurst, M., Bowskill, J., Morphett, J. & Jessop, M. (1998), A wearable spatial conferencing space, *in* 'Proc. ISWC'98', Pittsburgh, Penn., USA.

Billinghurst, M. & Kato, H. (1999), Collaborative mixed reality, *in* 'Proc. ISMR'99', Springer Verlag, Yokohama, Japan, pp. 261–284.

Dijkstra, E. (1959), 'A note on two problems in connection with graphs', *Numerische Mathematik* **1**, 269–271.

Feiner, S., MacIntyre, B., Haupt, M. & Solomon, E. (1993), Windows on the world: 2D windows for 3D augmented reality, *in* 'ACM Symposium on User Interface Software and Technology', pp. 145–155. not read.

Feiner, S., MacIntyre, B., Höllerer, T. & Webster, A. (1997), A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban enviroment, *in* 'Proc. ISWC'97', Cambridge, MA, USA, pp. 74–81.

Höllerer, T., Hallaway, D., Tinna, N. & Feiner, S. (2001), Steps toward accommodating variable position tracking accuracy in a mobile augmented reality system, *in* 'Proc. AIMS'01', Seattle, WA, USA, pp. 31–37.

Höllerer, T. & Pavlik, J. (1999), Situated documentaries: Embedding multimedia presentations in the real world, *in* 'Proc. ISWC'99', San Francisco, CA, USA, pp. 79–86.

Intersense (2002), 'Intersense IS-900', http://www.isense.com/ [2002, September 5].

Kourogi, M. & Sakaue, T. K. K. (2001), A panorama-based method of personal positioning and orientation and its real-time applications for wearable computers, *in* 'Proc. ISWC'01', Zurich, Switzerland, pp. 107–114.

Lee, S.-W. & Mase, K. (2001), Incremental motion-based location recognition, *in* 'Proc. ISWC'01', Zurich, Switzerland, pp. 123–130.

MacIntyre, B. & Coelho, E. M. (2000), Adapting to dynamic registration errors using level of error (loe) filtering, *in* 'Proc. ISAR2000', IEEE and ACM, Munich, Germany, pp. 85–88.

Newman, J., Ingram, D. & Hopper, A. (2001), Augmented reality in a wide area sentient environment, *in* 'Proc. ISAR 2001', IEEE and ACM, New York, New York, USA, pp. 77–86.

Piekarski, W. & Thomas, B. H. (2001), Tinmith-metro: New outdoor techniques for creating city models with an augmented reality wearable computer, *in* 'Proc. ISWC 2001', IEEE, Zurich, Switzerland, pp. 31–38.

Piekarski, W. & Thomas, B. H. (2002), The tinmith system - demonstrating new techniques for mobile augmented reality modelling, *in* 'Proc. AUIC2002', ACS, Melbournce, Vic, Australia.

Pinz, A. (2001), Consistent visual information processing applied to object recognition, landmark definition, and real-time tracking, *in* 'Proc. VMV'01', Stuttgart, Germany.

Raab, F., Blood, E., Steiner, T. & Jones, R. (1979), 'Magnetic position and orientation tracking system', *IEEE Transactions on Aerospace and Electronic Systems* AES-15(5), 709–718.

Reitmayr, G. & Schmalstieg, D. (2001*a*), Mobile collaborative augmented reality, *in* 'Proc. ISAR 2001', IEEE, New York, New York, USA, pp. 114–123.

Reitmayr, G. & Schmalstieg, D. (2001*b*), An open software architecture for virtual reality interaction, *in* 'Proc. VRST 2001', ACM, Banff, Alberta, Canada, pp. 47–54.

Rekimoto, J. (1998), Matrix: A realtime object identification and registration method for augmented reality, *in* 'Proc. APCHI'98'.

Satoh, K., Hara, K., Anabuki, M., Yamamoto, H. & Tamura, H. (2001), TOWNWEAR: An outdoor wearable MR system with high-precision registration, *in* 'Proc. IEEE Virtual Reality 2001', Yokohama, Japan, pp. 210–211.

Schmalstieg, D., Fuhrmann, A., Hesina, G., Szalavari, Z., Encarnao, L. M., Gervautz, M. & Purgathofer, W. (2002), 'The studierstube augmented reality project', *PRESENCE - Teleoperators and Virtual Environments* **11**(1).

Starner, T., Mann, S., Rhodes, B., Levine, J., Healey, J., Kirsch, D., Picard, R. & Pentland, A. (1997), 'Augmented reality through wearable computing', *Presence* **Special Issue on Augmented Reality**(4), 386–398.

Stoakley, R., Conway, M. J. & Pausch, R. (1995), Virtual reality on a wim: interactive worlds in miniature, *in* 'Conference proceedings on Human factors in computing systems', ACM Press, Addison-Wesley, Denver, Colorado, USA, pp. 265–272.

Suomela, R. & Lehikoinen, J. (2000), Context compass, *in* 'Proc. ISWC 2000', IEEE, Atlanta, Georgia, USA, pp. 147–154.

Thomas, B., Close, B., Donoghue, J., Squires, J., Bondi, P. D., Morris, M. & Piekarski, W. (2000), Arquake: An outdoor/indoor augmented reality first person application, *in* 'Proc. ISWC2000', IEEE, Atlanta, Georgia, USA, pp. 139–146.

Thomas, B. H., Demczuk, V., Piekarski, W., epworth, D. H. & Gunther, B. (1998), A wearable computer system with augmented reality to support terrestrial navigation, *in* 'Proc. ISWC'98', IEEE and ACM, Pittsburgh, USA, pp. 168–171.

Welch, G., Bishop, G., Vicci, L., Brumback, S., Keller, K. & Colucci, D. (2001), 'High-performance wide-area optical tracking - the hiball tracking system', *Presence: Teleoperators and Virtual Environments* 10:1.