AIMS '01: Second Int. Workshop on Artificial Intelligence in Mobile Systems, Seattle, WA, Aug. 4, 2001, pp.31–37

# Steps Toward Accommodating Variable Position Tracking Accuracy in a Mobile Augmented Reality System

Tobias Höllerer Drexel Hallaway Navdeep Tinna Steven Feiner

Department of Computer Science
Columbia University
500 West 120th St., 450 CS Building
New York, NY 10027
+1 212 939 7000
{htobias,drexel,nst7,feiner}@cs.columbia.edu

## Abstract

The position-tracking accuracy of a location-aware mobile system can change dynamically as a function of the user's location and other variables specific to the tracker technology used. This is especially problematic for mobile augmented reality systems, which ideally require extremely precise position tracking for the user's head, but which may not always be able to achieve the necessary level of accuracy. While it is possible to ignore variable positional accuracy in an augmented reality user interface, this can make for a confusing system; for example, when accuracy is low, virtual objects that are nominally registered with real ones may be too far off to be of use.

To address this problem, we describe the early stages of an experimental mobile augmented reality system that adapts its user interface automatically to accommodate changes in tracking accuracy. Our system employs different technologies for tracking a user's position, resulting in a wide variation in positional accuracy: an indoor ultrasonic tracker and an outdoor real-time kinematic GPS system. For areas outside the range of both, we introduce a dead-reckoning approach that combines a pedometer and orientation tracker with environmental knowledge expressed in spatial maps and accessibility graphs. We present preliminary results from this approach in the context of a navigational guidance system that helps users to orient themselves in an unfamiliar environment. Our system uses inferencing and path planning to guide users toward targets that they choose.

## 1 Introduction

One of the strongest advantages of mobile and wearable computing systems is the ability to support *location-aware* or *location-based* computing, offering services and information that are relevant to the user's current locale [3]. Location-aware computing systems need to sense or otherwise be told their current position, either absolute within some reference coordinate system or relative to landmarks known to the system.

Augmented reality systems, which overlay spatially registered information on the user's experience of the real world, offer a potentially powerful user interface for location-aware computing. To register visual or audio virtual information with the user's environment, an augmented reality system must have an accurate estimate of the user's position and head orientation. There are many competing tracking technologies, which vary greatly as to their range, physical characteristics, and how their spatial and temporal accuracy is affected by properties of the environments in which they are used. One particularly appealing approach is to combine multiple tracking technologies to create hybrid trackers, using the different technologies either simultaneously or in alternation, depending upon the current environment. In all cases, however, if information registration techniques designed for accurate tracking are employed when tracker accuracy is too low, virtual information will not be positioned properly, resulting in a misleading or even unusable user interface.

To address this problem, we are developing an experimental mobile augmented reality system that adapts its user interface automatically to accommodate changes in tracking accuracy. Our system employs different technologies for tracking a user's position, resulting in a wide variation in positional accuracy. These technologies include a ceiling-mounted ultrasonic tracker covering a portion of an indoor lab, and a real-time kinematic GPS system covering outdoor areas with adequate visibility of the sky. For areas outside the range of both of these tracking systems, we have developed a dead-reckoning approach that combines a pedometer and orientation tracker with environmental knowledge expressed in spatial maps and accessibility graphs. Our adaptive user interface is designed to serve as a navigational assistant, helping users to orient themselves in a unfamiliar environment. Inferencing and path planning components use the environmental knowledge to guide users toward targets that they choose.

In the remainder of this paper, we first describe previous related work in Section 2. Next, in Section 3, we present our hybrid tracking approach, concentrating on our method for improving the accuracy of dead reckoning through the use of spatial maps and accessibility graphs. Then, in Section 4, we introduce an adaptive augmented reality user interface for navigational guidance that accommodates differences in positional accuracy. Within this context, we describe the intelligent navigation aids that we have developed for our system in Section 5. Finally, in Section 6, we present our conclusions and plans for future work.

## 2 Previous Work

Many approaches to position tracking require that the user's environment be equipped with sensors[17], beacons [15] [28] [6], or visual fiducials [20]. Tethered position and orientation tracking systems have attained high accuracy for up to room-sized areas using magnetic [26], ultrasonic, and optical technologies, including dense arrays of ceiling-mounted optical beacons [1]. Alternatively, sparsely placed infrared beacons can support tetherless position-only tracking over an entire building at much lower accuracy [28], [6].

Mobile phone technology has also been used to provide coarse position tracking over a potentially unlimited area. Among others, British mobile phone companies Vodafone and BT Cellnet already offer cell identification and cell broadcasting services, that inform customers of the code for the area in which they are currently located, and give them access to local services [9]. One can achieve better than cell-size resolution (about 50m for areas with good coverage) by employing triangulation methods, measuring time-of-flight information for the radio signals to three or more transceiver base stations [8].

For outdoor tracking, satellite-based global positioning system (GPS) receivers track 3DOF position when at least four satellites are visible, yielding roughly 10–20 m accuracy. Differential GPS systems broadcast correction information from a stationary base station to roving users, based on comparing computed position with the known position of a carefully surveyed reference antenna; this can substantially improve tracking accuracy within many miles of the reference antenna, achieving submeter accuracy. Real-time kinematic (RTK) GPS uses information about the GPS signal's carrier phase at the base station and the rover to reach centimeter-level accuracy.

GPS is line of sight and it loses track easily when indoors, under tree cover, or near tall buildings (especially in so-called "urban canyons"). Terrestrial GPS transmitters or transceivers, known as *pseudolites* can supplement the satellite signals or replace them altogether [23]. More typically, GPS signal loss is addressed through *dead-reckoning* techniques that rely on tetherless local sensors (e.g., magnetometers, gyroscopes, accelerometers, odometers, and pedometers) [32].

Local-sensor–based methods form a class of tracking techniques in their own right [16, 22]. However, their position updates are reported relative to the previous location, resulting in errors that accumulate over time unless the position is periodically synchronized with absolute values from a reliable source.

Knowledge about the environment and the constraints that it imposes on navigation can serve as an important source of information to correct for inaccuracies in the tracking systems of choice. Example studies can be found in the field of mobile robotics, where this concept is called *model matching* or *map-based positioning* [5].

One sensor technology that is commonly used for model matching is computer vision. Vision-based tracking has been used to make *orientation* tracking much more accurate and stable; for example, for augmented reality [33, 4]. Using computer vision for *position* updates in unconstrained environments is disproportionately more challenging. However, by strategically placing visual fiducials in the environment, very accurate short-range position tracking has been demonstrated [20]. Larger fiducials are also being tried for wider area outdoor and indoor tracking [31].

Experimental fiducial-free vision-based tracking approaches compare features of recorded imagery to a catalogue of previously gathered features for the given environment, making it possible to identify discrete events, such as entering a room [29, 2]. Other recent work attempts to identify continuous paths through an environment based on omnidirectional imagery captured from the user's head [27].

Given the wide range of strengths and weaknesses that different tracking technologies have in different circumstances, one promising approach is to combine a set of complementary technologies to create hybrid trackers that are more robust or accurate than any of the individual technologies on which they rely. Hybrid tracking systems have been developed both as commercial products [19] and research prototypes [16, 21, 10, 22]. Hybrid tracking systems, in which different technologies are used in alternation, may experience large variations in accuracy from one point in time to another, as the specific technologies in use are phased in and out.

Several researchers have begun to explore the question of how user interfaces can take into account tracking errors and other environment-specific factors. MacIntyre and Coelho [24] introduce the notion of *level-of-error* filtering for augmented reality: computing a registration error value that is used to select one of a set of alternate representations for a specific augmentation. We believe that their notion of only one single pose measurement error value needs to be extended to distinguish position errors (as we explore here) from orientation errors, and to account for other varying tracking characteristics (e.g., update rates or likelihood to drift). Butz and colleagues [7] describe an adaptive graphics generation system for navigational guidance. While our projects share many of the same goals, we concentrate on user interfaces for augmented reality, while their initial implementation focuses on small portable devices and stationary displays.

# 3 Complementary Tracking Modes

The experimental adaptive mobile augmented reality user interface that we describe in this paper is intended to assist a user in navigating through an unfamiliar environment. It is designed for use with our custom-built backpack computer, based on an Intel Pentium III 700MHz processor, and nVidia GeForce2 MX 3D graphics accelerator, and connected to our campus backbone through IEEE 802.11b wireless networking [18]. The user interface is presented on a Sony LDI-D100B see-through head-worn display, and is implemented in Java 3D. Our system relies on different technologies for tracking a user's position in three different circumstances: within part of a research laboratory served by a high-precision ceiling tracker, in indoor hallways and rooms outside of the ceiling tracker range, and outdoors.

Orientation tracking is done with an InterSense IS300 Pro hybrid inertial/magnetic tracker. We can track both the user's head and body orientation by connecting head-mounted and belt-mounted sensors to the unit. When walking around indoors, we have to switch off the magnetic component of the tracker to avoid being affected by stray magnetic fields from nearby labs (see Section 3.1) and rely on purely inertial orientation information.

When outdoors with line of sight to at least four GPS or Glonass satellites, our system is position tracked by an Ashtech GG24 Surveyor RTK differential GPS system. For indoor tracking, we use a Point Research PointMan Dead-Reckoning Module (DRM) and an InterSense Mark II SoniDisk wireless ultrasonic beacon. The system can detect whether the beacon is in range of an InterSense Mark II ceiling tracker. The Mark II tracker is connected to a stationary tracking server and the position updates of the roaming user's SoniDisk beacon are relayed to the user's wearable computer using our Java-based distributed augmented reality infrastructure [18].

Tracking accuracies and update rates vary widely among these three position tracking approaches. The IS600 Mark II ceiling tracker can track the position of one SoniDisk to a resolution of about 1 cm at 20–50 Hz. The outdoor RTK differential GPS system has a maximum tracking resolution of 1–2 cm at an update rate of up to 5 Hz. The GPS accuracy may degrade to 10 cm, or even meter-level when fewer than six satellites are visible. If we lose communication to our GPS base station, we fall back to regular GPS accuracy of 10–20 m.

Our augmented reality user interface for navigational guidance adapts to the levels of positional tracking accuracy associated with different tracking modes. changes. In this paper, we focus on ceiling tracker and DRM tracking modes.

## 3.1 Wide Area Indoor Tracking using Dead Reckoning

Whenever the user is not in range of an appropriate ceiling tracker, our system has to rely on local sensors and knowledge about the environment to determine its approximate position. Unlike existing hybrid sensing approaches for indoor position tracking [16, 21, 10], we try to minimize the amount of additional sensor information to collect and process. The only additional sensor is a pedometer (the orientation tracker is already part of our mobile augmented reality system). Compared with [22] who use digital compass information for their heading information, we have a much more adverse environment to deal with (see discussion below). Therefore, we decided to rely on inertial orientation tracking and to correct for both the resulting drift and positional errors associated with the pedometer-based approach by means of environmental knowledge in the form of spatial maps and accessibility graphs of our environment.

Our dead reckoning approach uses the pedometer information from the DRM to determine when the user takes a step, but uses the orientation information from the more accurate IS300 Pro orientation tracker instead of the DRM's built-in magnetometer. We do this because the IS300 Pro's hybrid approach is more accurate and less prone to magnetic distortion. Furthermore, we have the option to use the IS300 Pro in inertial-only tracking mode. Figure 1(a) illustrates the problems that our indoor environment poses for magnetometer-based tracking. The plot corresponds to a user walking around the outer hallways of the 6th floor of our research building, using the IS300 Pro tracker in hybrid mode. The plot reflects a lot of magnetic distortion present in our building. In particular, the loop in the path on the left edge of the plot dramatically reflects the location of a magnetic resonance imaging device for material testing two floors above us.

For indoor environments with magnetic distortions of such proportions we decided to forgo magnetic tracker information completely and rely on inertial orientation data alone. Figure 1(b) shows the results for a user traveling the same path, with orientation tracking done by the IS300 Pro tracker in purely inertial mode. The plot clearly shows much straighter lines for the linear path segments but there is a linear degradation of the orientation information due to drift, resulting in the "spiral" effect in the plot, which should have formed a rectangle.

Figure 1(c) and (d) show the results after correcting the method of (b) with information about the indoor environment. Plot (c) shows a similar path through the outer hallway as those of plots (a) and (b). In contrast, plot (d) shows an "S"-shaped path from our lab door at the southeast, around the outside hallway at the east and north, down through the center corridor to the south hallway, then heading to and up the west hallway, and across the north hallway back to the north end of the center corridor. To perform these corrections, we use two different representations of the building infrastructure in conjunction: spatial maps and accessibility graphs.

*Spatial maps* accurately model the building geometry (walls, doors, passageways), while *accessibility graphs* give a coarser account of the main paths a user usually follows. Figure 2 compares the two representations for a small portion of our environment. Both the spatial map and the accessibility graph were modeled by tracing over a scanned floorplan of our building using a modeling program that we developed.

The spatial map models all walls and other obstacles. Doors are represented as special line segments (as denoted by the dashed lines connecting the door posts). In addition to its role in tracking correction, the accessibility graph is also the main data structure used by the path planning component described in Section 5.1.

For each step registered by the pedometer, and taking into account the heading computed by the orientation tracker, our dead reckoning algorithm checks the spatial map to determine if the user will cross an impenetrable boundary (e.g., a wall). If that is the case, then the angle of collision is computed. If this angle is below a threshold (currently 30 degrees), the conflict is classified as an artifact caused by orientation drift and the directional information is corrected to correspond to heading parallel to the obstacle boundary.

If the collision angle is greater than the threshold, the system searches for a segment on the accessibility graph that is close to

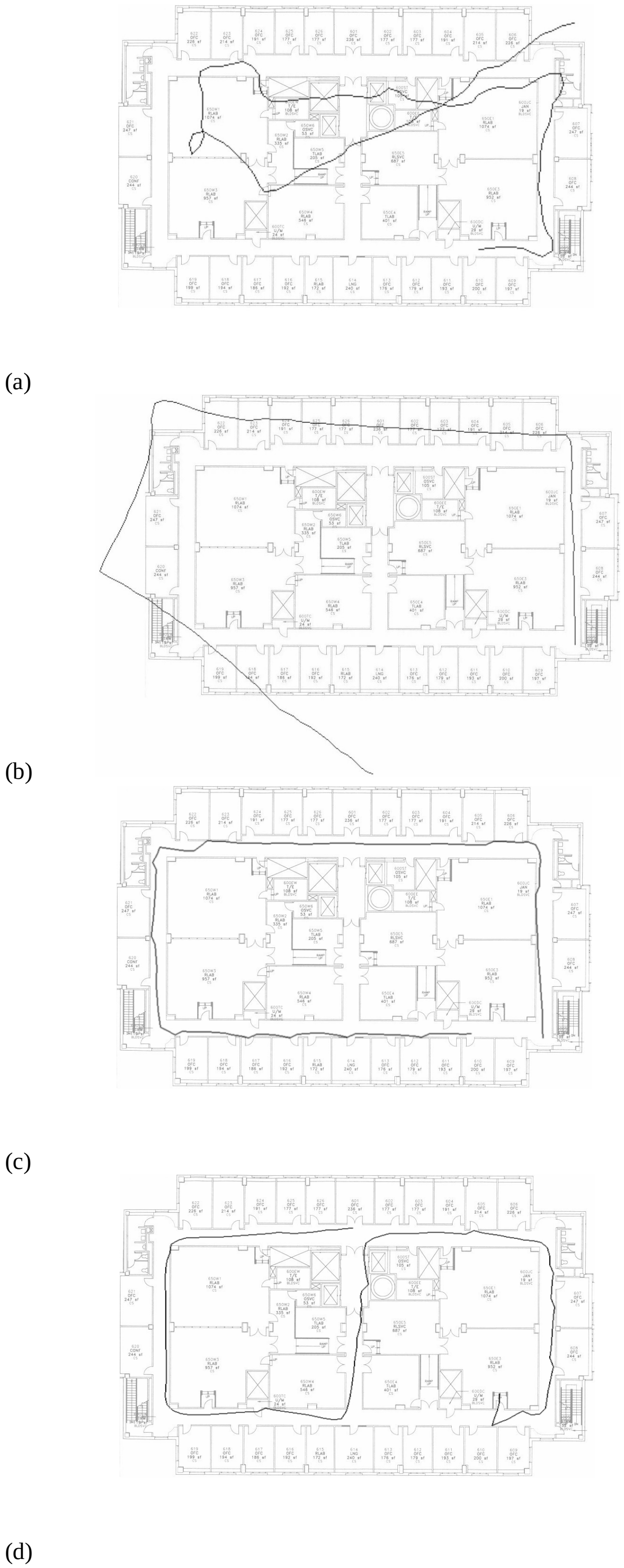

Figure 1: Tracking plots using the DRM in our indoor environment. (a) pedometer and magnetic orientation tracker, (b) pedometer and inertial orientation tracker, (c) & (d) pedometer, inertial orientation tracker, and environmental knowledge.

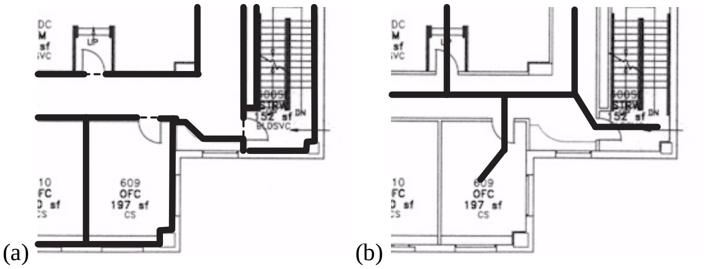

Figure 2: Two different representations of a small part of our building infrastructure, as used in the dead-reckoning–based tracking approach: (a) spatial map, (b) accessibility graph.

the currently assumed position, is accessible from the currently assumed position (i.e., is not separated from it by an impenetrable boundary, which is checked with the spatial map data structure), and is the closest match in directional orientation to the currently assumed heading information. The system assumes that the user is really currently located at the beginning of that segment and changes the last step accordingly to transport the user there.

Doors are handled as special cases. First, the sensitive door area is assumed to be larger than the doorframe itself (currently, all walls in the immediate continuation of the door 1 m to either side will trigger door events if the user attempts to cross them). In case of a door event, the angle of collision is determined. If the angle is below our 30 degree threshold, the system behaves as if the door were a simple wall segment and no passage occurs. If the angle is greater than 60 degrees, the system assumes that the user really wanted to enter through that door and proceeds correspondingly. If the angle is in between the two thresholds, the system continues with the accessibility graph search described above.

Our initial results with this approach are very promising. The plot in Figure 1(d) for example corresponds to a path along which the user successfully passed through three doors (the lab door at the east end of the south corridor, and two doors at the north end and middle of the center corridor), and never deviated far from the correct position. We are in the process of collecting more quantitative results on the adequacy of our approach.

## 4 Adaptive Augmented Reality User Interface

Figure 3 shows a view through the see-through head-mounted display when the user is accurately position tracked by the ceiling tracker. The system overlays features of the surrounding room, in this case a wireframe model consisting of our lab's walls and ceiling, doors, static objects of interest (e.g., a rear projection display), and rooms in the immediate neighborhood. Labels are realized as Java 3D [12] Text2D objects: billboarded polygons with transparent textures representing the label text. Labels are anchored at their corresponding 3D world positions, so that closer objects appear to have bigger labels. The color scheme highlights important objects (e.g., results of a navigational query, described in Section 5, and passageways from the current room to the main corridors).

When we roam with our mobile system—away from the ceiling tracker, but not yet outdoors where GPS can take over—we currently depend upon our hybrid, dead-reckoning system for positional data. As a result, we have relatively more accurate orientation tracking than position tracking. To leverage the relatively superior orientation accuracy in this situation, we have chosen to situate much of the overlaid material when roaming within the context of a World in Miniature (WIM) [30]: a scaled-down 3D model of our environment.

Figure 3: Augmented reality user interface in accurate tracking mode (imaged through see-through head-worn display). Labels and features (a wireframe lab model) are registered with the physical environment.

Our WIM has a stable position relative to the user's body, but is oriented relative to the surrounding physical world. That is, it hovers in front of the user, moving with her as she walks and turns about, while at the same time maintaining the same 3D orientation as the surrounding environment of which it is a model. In related work on navigational interfaces, Darken and colleagues [11] explore different ways of presenting 2D and 3D map information to a user navigating in a virtual environment. They conclude that while there is no overall best scheme for map orientation, a self-orienting "forward-up" map is preferable to a static "north-up" map for targeted searches. The WIM is a 3D extension of the "forward up" 2D option in Darken's work. Because our WIM's position is body-stabilized, the user can choose whether or not to look at it—it is not a constant consumer of head-stabilized head-worn display space, and doesn't require the attention of a tracked hand or arm to position it. If desired, the WIM can exceed the bounds of the HMD's restricted field of view, allowing the user to review it by looking around, since the head and body orientation are independently tracked. The WIM incorporates a model of the environment and an avatar representation of the user's position and orientation in that environment. It also provides the context in which paths are displayed in response to user queries about routes to locations of interest.

When the user moves out of range of the ceiling tracker, position tracking is shifted to the dead-reckoning tracker. To notify the user that this is happening, we first replace the registered world overlay with the WIM model, but at full-scale and properly registered. Then the WIM is interpolated in scale and position to its destination configuration [25].

Figure 4 shows the user interface just after this transition. Because the head–body alignment is relatively constant between these two pictures, the position of the projected WIM relative to the display is similar in both pictures, but the differing position and orientation of the body relative to the world reveal that the WIM is world aligned in orientation. These images also include route arrows that point the way along a world-scale path to a location that the user has requested (in this case, the nearest stairway). As the user traverses this suggested path, the arrows advance, always showing the two next segments. The WIM also displays the entire path, which is difficult to see in these figures because of problems imaging through

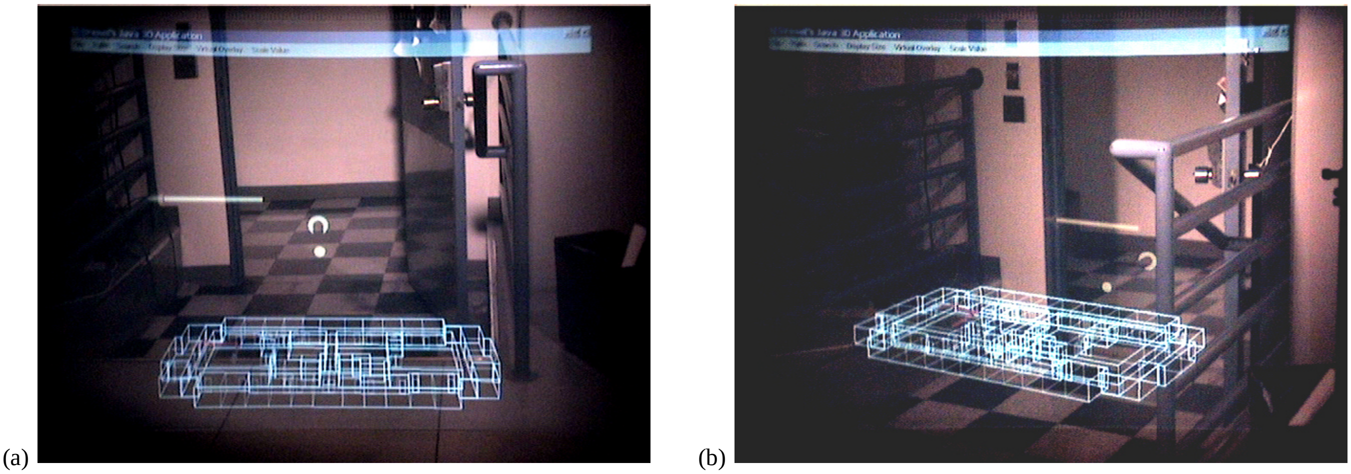

Figure 4: Augmented reality user interface in DRM-tracked mode (imaged through see-through head-worn display). (a) A body-stabilized world-aligned WIM with world-space arrows. (b) The same WIM with the user at a different position and orientation.

the see-through head-worn display. (A more legible view of a path is in shown in Figure 5(b), which is a direct frame-buffer capture, and therefore doesn't show the real world on which the graphics are overlaid.)

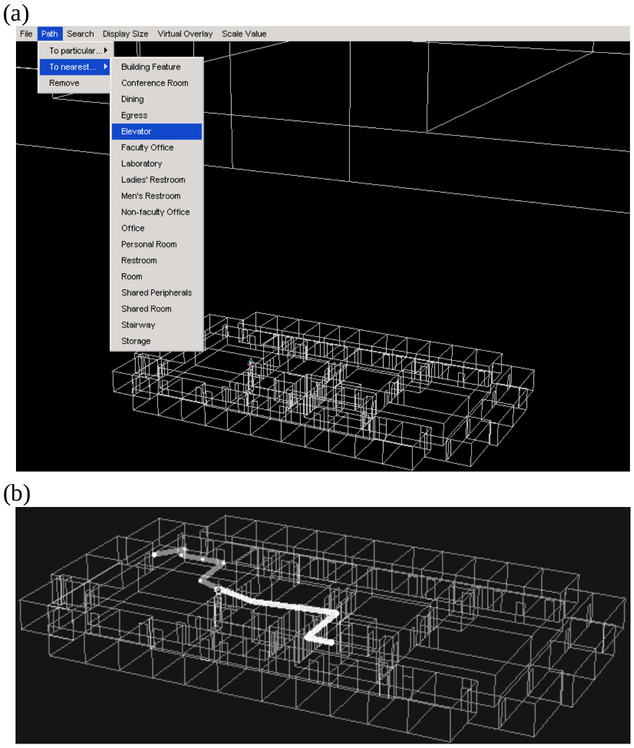

Figure 5: Navigational guidance. (a) User query. (b) Different solution paths in the WIM.

## 5 Intelligent Navigation Aids

In Figure 5(a) the user uses a menu to request the path to the nearest elevator. The system responds to this query with two solutions. The first of the two is represented in Figure 5(b) as a larger, brighter 3D path to the most literal solution—the nearest elevator. The second is plotted as a medium-sized, less bright path to the nearest stairway. A reasoning component determines that although the user has explicitly specified an interest in elevators, she may actually be interested in some means of egress, and, since the stairway is closer, it is presented as well (see below). Solution paths begin at the tracked position of the user, represented in the WIM by the avatar, and find their second vertex as the nearest accessible node in the underlying accessibility graph, which represents the set of all possible paths.

### 5.1 Knowledge Representation

The system's knowledge of the physical domain and its resources resides in a persistent database [18]. At load time, tables in that database are parsed into structures that implement a rudimentary Description Logic [14]. In the domain described here, the *concepts* [14] are the classes of resources found on the 6th floor of the building where our lab is located. At the lowest level, concepts include things like "Men's Restroom," "Dining," "Stairway," "Laboratory," and "Office." The subsumption of each concept by its more general parent creates a conceptual tree, culminating in a root—the entire set of resources that we model in our building. The *TBox* (which handles terminological knowledge about concepts) includes a list of these concepts, each associated with its subsuming parent. The *ABox* (which handles assertional knowledge about individuals) includes a list of individual resources, each associated with a concept (the most specific membership) and the path node that is its location of availability in the world. It is up to the system—at load time or run time—to infer more general concept memberships.

A more logistical, metrical concept, outside the hierarchy of resources, is that of the Path Node. To employ the graph searching techniques of A* or Dijkstra's Algorithm [13], we represent the graph (of possible paths to resources) in our database and data structures as a set of these nodes. In an ABox table independent of the individual resources above, we list a set of path nodes and associate them with 3D world positions. In a separate table we represent the edges in this graph as pairs of nodes that encode, in keeping with Description Logic theory, constructors of the *role* “connectedTo” (or “accessibleFrom”). At load time, these individual nodes and edge roles are parsed into our accessibility graph, which is typically, but not necessarily, undirected and planar.

When the user of our system asks for the path to an individual resource, the shortest path is calculated on our graph structure using Dijkstra’s Algorithm. When a user asks for the way to the nearest of a certain kind of resource, however, comparisons must be made. The length of the shortest path—from the user’s position, along the traversable edges of our graph, to a candidate resource—is the metric we want to minimize. The user indicates how many plies she wishes the search to traverse, or accepts the default number of plies. When she asks for the nearest Elevator, as shown in Figure 5(a), the first solution shows just that. The lengths of the shortest paths from her position to the path nodes associated with all the individuals in the concept Elevator are compared, and the shortest one wins: in this case, the path to the South Elevator. If the ply choice is greater than zero, though, the system notes that the concept Elevator is subsumed by that of Egress, and hence proceeds to evaluate members of that parent concept, which subsumes the concept Stairway. Since the East Stairway is nearer than the South Elevator, a path to it is plotted as a second solution, with somewhat less prominent graphical presence. In this example, since the ply count was set to 2, the system traversed one level higher, but found no solution with a shorter path in that yet more general set. Had it found one, a third path would have been plotted, with even less prominent graphical characteristics.

## 6 Conclusions and Future Work

We have described a mobile augmented reality system that employs different modes of tracking a user’s position, resulting in a wide variation in positional accuracy between the different modes. One of these tracking modes is established by a new dead-reckoning tracking module that makes use of pedometer and orientation information, and applies corrections derived from knowledge about the user’s immediate environment in the form of area maps and accessibility graphs. We presented the early stages of an augmented reality user interface that automatically adapts to the changes in tracking accuracy associated with these different tracking modes, and modifies its visual representation accordingly. Finally we introduced the knowledge-based components used in our augmented reality user interface for navigational guidance.

Our research to date raises several interesting questions. Does a 3D WIM, stabilized in some manner with respect to the user, inviting a sense of “forward,” offer measurable navigational advantages over a 2D map with an implicit sense of “up” that might be screen-stabilized? Is a body-stabilized, world-oriented WIM significantly more powerful than ones that are head-stabilized and world-aligned, head-stabilized and north-forward, or body-stabilized and north-forward? These questions suggest the need for a taxonomy of navigational “maps.” Possible principal dimensions for such a taxonomy are spatial dimensionality (2D or 3D), positional stabilization, and orientational alignment.

A number of issues could be addressed through user studies. Considering head-stabilization of WIM position, might it be better to fix the height, allowing the head to look up (away from) and down (to) the WIM, or should the WIM remain within the frustum regardless of where the head looks? Given body stabilization and world-orientation, might it be better to have the user immersed in the WIM with the centroid of her world-sized, physical body coincident with her position in the WIM? Or, as we conjecture in the design of our system, might it be better to situate the WIM with its centroid (and its entire volume) somewhat in front of the user’s body? Immersing the user directly in a WIM would avoid the indirection and potential distraction implicit in representing her in the WIM by an avatar, but does this offset the presumed disadvantage of having the user’s physical body displace considerably more than its realistic “share” of the WIM’s volume? Does one really want the user to have to look “inside” herself to see the miniature version of the floor several meters in front of where she currently stands? Can she tell exactly where she is in the miniature, without some virtual representation of herself? Should the user’s locus in the WIM be body-stabilized (rather than stabilizing the WIM’s centroid), and the user’s position be represented by a virtual belt-buckle that would overlay the real thing (instead of being back in the centroid of her physical body)?

## 7 Acknowledgements

The research described here is funded in part by ONR Contracts N00014-99-1-0249, N00014-99-1-0394, and N00014-99-0683, NSF Grant IIS-00-82961, and gifts from Intel, Microsoft, and Mitsubishi. We wish to thank Elias Gagas for his contributions to the earlier stages of the path-finding graphical user interface, and for valuable discussions about applying Description Logic theory to navigational queries, Simon Shamoun for writing the first version of a 2D map navigation interface that helped us conduct our experiments with the dead-reckoning module, and Gus Rashid for developing the software that allows us to easily create 3D floor models, 2D spatial maps, and accessibility graphs from floorplan blueprints.

## References

[1] 3rdTech. http://www.3rdtech.com/HiBall.htm, 2001.

[2] H. Aoki, B. Schiele, and A. Pentland. Realtime personal positioning system for wearable computers. In *Proc. ISWC ’99 (Third Int. Symp. on Wearable Computers)*, pages 37–43, San Francisco, CA, October 18–19 1999.

[3] H. Beadle, B. Harper, G. Maguire Jr., and J. Judge. Location aware mobile computing. In *Proc. ICT ’97 (IEEE/IEE Int. Conf. on Telecomm.)*, Melbourne, Australia, 1997.

[4] R. Behringer. Registration for outdoor augmented reality applications using computer vision techniques and hybrid sensors. In *Proc. IEEE Virtual Reality ’99*, pages 244–251, March 13–17 1999.

[5] J. Borenstein, H. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. A K Peters, Natick, MA, 1996.

[6] A. Butz, J. Baus, and A. Krüger. Augmenting buildings with infrared information. In *Proceedings of the International Symposium on Augmented Reality ISAR 2000*, pages 93–96. IEEE Computer Society Press, Oct 5–6 2000.

[7] A. Butz, J. Baus, A. Krüger, and M. Lohse. A hybrid indoor navigation system. In *IUI2001: International Conference on Intelligent User Interfaces*, New York, 2001. ACM.

[8] Cambridge Position Systems. Cursor E-OTD mobile location technology. http://www.cursor-system.com/ sitefiles/ cursor/ tech_technology.htm, 2001.

[9] Cell broadcasting technology. http://www.btcellnet.net/, http://www.vodaphone.uk.co, 2001.

[10] B. Clarkson, K. Mase, and A. Pentland. Recognizing user context via wearable sensors. In *Proc. ISWC '00 (Fourth Int. Symp. on Wearable Computers)*, pages 69–75, Atlanta, GA, October 16–17 2000.

[11] R. Darken and H. Cevik. Map usage in virtual environments: Orientation issues. In *Proceedings of IEEE VR '99*, pages 133–140, 1999.

[12] M. Deering and H. Sowizral. *Java3D Specification, Version 1.0*. Sun Microsystems, 2550 Garcia Avenue, Mountain View, CA 94043, USA, Aug. 1997.

[13] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[14] F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Reasoning in description logics. In G. Brewka, editor, *Principles of Knowledge Representation*, Studies in Logic, Language and Information, pages 193–238. CSLI Publications, 1996.

[15] I. Getting. The global positioning system. *IEEE Spectrum*, 30(12):36–47, Dec. 1993.

[16] A. R. Golding and N. Lesh. Indoor navigation using a diverse set of cheap, wearable sensors. In *Proc. ISWC '99 (Third Int. Symp. on Wearable Computers)*, pages 29–36, San Francisco, CA, October 18–19 1999.

[17] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. In *Proc. of the Fifth ACM/IEEE Int. Conf. on Mobile Computing and Networking (MobiCom)*, pages 59–68, Seattle, WA, Aug. 1999.

[18] T. Höllerer, S. Feiner, T. Terauchi, G. Rashid, and D. Hallaway. Exploring MARS: Developing indoor and outdoor user interfaces to a mobile augmented reality system. *Computers and Graphics*, 23(6):779–785, 1999.

[19] InterSense IS-900 Wide Area Precision Motion Tracker. http://www.isense.com, 2001.

[20] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana. Virtual object manipulation on a table-top ar environment. In *Proceedings of the International Symposium on Augmented Reality ISAR 2000*, pages 111–119. IEEE Computer Society Press, Oct 5–6 2000.

[21] K. V. Laerhoven and O. Cakmakci. What shall we teach our pants? In *Proc. ISWC '00 (Fourth Int. Symp. on Wearable Computers)*, pages 77–83, Atlanta, GA, October 16–17 2000.

[22] S.-W. Lee and K. Mase. A personal indoor navigation system using wearable sensors. In *Proc. ISMR '01 (Second Int. Symp. on Mixed Reality)*, pages 147–148, Yokohama, Japan, March 14–15 2001.

[23] E. LeMaster and S. Rock. Field test results for a self-calibrating pseudolite array. In *Proc. of Institute of Navigation GPS-2000*, Salt Lake City, Utah, Sept. 2000.

[24] B. MacIntyre and E. M. Coelho. Adapting to dynamic registration errors using level of error (LOE) filtering. In *Proc. ISAR '00 (Int. Symposium on Augmented Reality)*, pages 85–88, Munich, Germany, October 5–6 2000.

[25] R. Pausch, T. Burnette, D. Brockway, and M. Weiblen. Navigation and locomotion in virtual worlds via flight into hand-held miniatures. *Proc. SIGGRAPH '95*, pages 399–401, 1995.

[26] F. Raab, E. Blood, T. Steiner, and R. Jones. Magnetic position and orientation tracking system. *IEEE Trans. on Aerospace and Electronic Systems*, AES-15(5):709–718, September 1979.

[27] W. Rungsarityotin and T. Starner. Finding location using omnidirectional video on a wearable computing platform. In *Proc. ISWC '00 (Fourth Int. Symp. on Wearable Computers)*, pages 61–68, Atlanta, GA, October 16–17 2000.

[28] T. Starner, D. Kirsch, and S. Assefa. The locust swarm: An environmentally-powered, networkless location and messaging system. In *Proc. ISWC '97 (First Int. Symp. on Wearable Computers)*, pages 169–170, Cambridge, MA, October 13–14 1997.

[29] T. Starner, B. Schiele, and A. Pentland. Visual contextual awareness in wearable computing. In *Proc. ISWC '98 (Second Int. Symp. on Wearable Computers)*, pages 50–57, Pittsburgh, PA, October 19–20 1998.

[30] R. Stoakley, M. Conway, and R. Pausch. Virtual reality on a WIM: Interactive worlds in miniature. In *Proceedings of Human Factors in Computing Systems (CHI '95)*, pages 265–272, May 7–11 1995.

[31] B. Thomas, B. Close, J. Donoghue, J. Squires, P. De Bondi, M. Morris, and W. Piekarski. Arquake: An outdoor/indoor augmented reality first person application. In *Proc. ISWC '00 (Fourth Int. Symp. on Wearable Computers)*, pages 139–146, Atlanta, GA, October 16–17 2000.

[32] Trimble Navigation Ltd. Automatic vehicle location tutorial: Dead reckoning. http://www.trimble.com/mpc/avl/dr.htm, 2001.

[33] S. You, U. Neumann, and R. Azuma. Hybrid inertial and vision tracking for augmented reality registration. In *Proc. IEEE Virtual Reality '99*, pages 260–267, Houston, TX, March 13–17 1999.