

□ BRIDGING THE GAPS: HYBRID TRACKING FOR ADAPTIVE MOBILE AUGMENTED REALITY

DREXEL HALLAWAY and STEVEN FEINER
Department of Computer Science, Columbia
University, New York, NY, USA

TOBIAS HÖLLERER
University of California, Santa Barbara,
CA, USA

Tracking accuracy in a location-aware mobile system can change dynamically as a function of the user's location and other variables specific to the tracking technologies 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 that 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 an experimental mobile augmented reality system that: (1) employs multiple position-tracking technologies, including ones that apply heuristics based on environmental knowledge; (2) coordinates these concurrently monitored tracking systems; and (3) automatically adapts the user interface to varying degrees of confidence in tracking accuracy. We share our experiences with managing these multiple tracking technologies, employing various techniques to facilitate smooth and reasonable “hand-offs” between the cooperating systems. We present these results in the context of a intelligent navigational guidance system that helps users to orient themselves in an unfamiliar environment, using path planning to guide them toward destinations they choose, and sometimes towards ones the system infers as equally relevant.

The research described here is funded in part by ONR Contracts N00014-99-1-0249, N00014-99-1-0394, and N00014-99-0683, NSF Grants IIS-00-82961 and IIS-01-21239, and gifts from Intel, Microsoft, and Mitsubishi. We wish to thank Navdeep Tinna for his invaluable contributions toward the work with the DRM; 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 floor-plan blueprints.

Address correspondence to Drexel Hallaway, Department of Computer Science, Columbia University, 1214 Amsterdam Avenue, MC 0501, New York, NY 10027, USA. E-mail: drexel@cs.columbia.edu

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 (Beadle et al. 1997). 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 (Hightower and Borriello 2001; Welch and Foxlin 2002). 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 several 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 + GLONASS system covering outdoor areas with adequate visibility of the sky. To bridge the gap between both these tracking systems, when outside their range, we have developed dead-reckoning and infrared approaches. Our dead-reckoning approach combines a pedometer and an orientation tracker with heuristics applied to environmental knowledge expressed in a spatial map and an accessibility graph. Our infrared tracker leverages the partitioning effects of the intersections and subtractions of overlapping beacon zones of influence to provide a position estimate whose accuracy is largely a function of the density of the chosen beacon layout. We have experimented within an adaptive user interface that is designed to serve as an intelligent navigational assistant, helping users to orient themselves in an unfamiliar environment. Inferencing and path-planning components use environmental knowledge to guide users toward destinations they choose—and sometimes toward those *not* explicitly chosen, if the system reasons that the user will find them more proximate and similar.

PREVIOUS WORK

Many approaches to position tracking require that the user's environment be equipped with sensors (Golding and Lesh 1999), beacons (Getting 1993; Starner et al. 1997; Butz et al. 2000), or visual fiducials (Kato et al. 2000). Tethered position and orientation tracking systems have attained high accuracy for up to room-sized areas using magnetic (Raab et al. 1979), ultrasonic (Foxlin et al. 1998), and optical technologies, including dense arrays of ceiling-mounted optical beacons (3rdTech Corp. 2002; Welch et al. 1999). The Bat system relies on ultrasonic sensors distributed throughout a wide area, triangulating on radio-synchronized acoustic signals received from tracked objects (Newman et al. 2001). It has been shown to be effective, not only in position-tracking, but also in coarse orientation-tracking—especially when fused with superior local sensors for the latter.

Though a somewhat coarser approach, the signal strengths of multiple IEEE 802.11b WiFi network access-point antennae can afford a reasonable determination of position in a context such as a university campus (Griswold et al. 2002). The RADAR system (Bahl and Padmanabhan 2000) uses multilateration and pre-computed signal strength maps for this purpose, while Castro et al. (2001) employ a Bayesian networks approach. The achievable resolution depends on the density of access points deployed to form the wireless network. Ekahau (2002), which offers a commercial solution based on this technology, claims that with sufficient transmitters their solution can achieve meter-level accuracy.

Sparsely placed infrared beacons can support tetherless navigation throughout an entire building at much lower accuracy (Butz et al. 2001; Butz et al. 2000). In the Swarm of Locusts (Starner et al. 1997), infrared beacons mapping to individual cells provide coarse location and/or object tagging. While our infrared tracking research shares many of the same goals and some of the same hardware as that of Butz and colleagues, we concentrate on user interfaces for augmented reality, while their initial implementation focuses on small portable devices and stationary displays. In further contrast, our infrared tracking approach exploits layout designs that create overlapping signals, allowing a signal *set* to uniquely denote an area fragment smaller than the entire coverage area of any one beacon.

For outdoor tracking, satellite-based global positioning system (GPS) receivers track 3-degrees-of-freedom (3DOF) position when at least four satellites are visible. Differential GPS systems improve accuracy by broadcasting correction information from a stationary base station to roving users, based on comparing the computed position with the known position of a carefully surveyed reference antenna. Real-time-kinematic (RTK) GPS uses information about the GPS signal's carrier phase at the base station and the rover to reach even better (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”). GPS signal loss is often addressed through dead-reckoning techniques (Lee and Mase 2001) that rely on tetherless local sensors, such as magnetometers, gyroscopes, accelerometers, odometers, and pedometers (Bowditch 1802).

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 (Borenstein et al. 1997).

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 (InterSense 2001) and research prototypes (Golding and Lesh 1999; Laerhoven and Cakmakci 2000; Clarkson et al. 2000; Lee and Mase 2001). 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 accuracy and other environment-specific factors. One approach (MacIntyre and Coelho 2000; MacIntyre et al. 2002) introduces the notion of level-of-error filtering for augmented reality—addressing the issue of object tracking error at the viewport-projection level: Registration error values are used to select one of a set of alternate representations for a specific augmentation. In addition to this viewport-projection approach, it seems useful to retain a sense of the certainty of each dimension estimate in 3D (e.g., x , y , z , yaw , $pitch$ and $roll$)—or at least of sets of them (e.g., position and orientation)—perhaps also to account for other varying tracking characteristics, such as update rates and likelihood to drift. We use the outputs of filtering techniques to provide standard deviations for each dimension of measurement.

COMPLEMENTARY TRACKING MODES

Our system addresses the problem of tracking the user across three different environments: indoors in our lab, in hallways and other rooms outside our lab, and outdoors. In all three circumstances, we currently handle orientation tracking with an InterSense IS 300 Pro hybrid inertial/magnetic tracker. We can track both the user’s head and body orientation by connecting head-worn and belt-mounted sensors to the unit. In portions of our indoor environment, we have to switch off the magnetic component of

the tracker to avoid being affected by stray magnetic fields from nearby labs, and rely on purely inertial orientation information.

Each of these three environments requires a different approach to position tracking, however. When *outdoors*, with line of sight to at least four GPS (US) or GLONASS (Russia) global navigation satellites, our system is position tracked by an Ashtech GG24 Surveyor real-time–kinematic differential GPS + GLONASS system. For indoor tracking in our *lab*, we employ an InterSense IS 600 Mark 2 ceiling-mounted tracker. Wearing its wireless ultrasonic beacon allows the user to roam untethered beyond the confines of that portion of our lab served by it. When the user is under the IS 600’s crossbar(s), we have the benefit of its high-precision position tracking. In transitional regions, serviceable neither by GPS nor by our ceiling tracker, we bridge the gaps with one of two experimental systems. The first employs a pedometer and supplements its capabilities with knowledge of the environment. The second is our experimental infrared tracker (Hallaway et al. 2003), which strategically poses an inexpensive array of unsynchronized, infrared beacons—whose zones of influence intersect to partition the covered area into a set of uniquely defined fragments—and infers position from that set of beacons currently received by a user-worn array of low-cost, off-the-shelf, infrared dongles.

Our system detects when the wireless, ultrasonic beacon is beyond the range of the ceiling tracker, and a meta-tracking filter effects a hand-off to one of the less-accurate systems.

Accuracy and update rate both vary widely among these position-tracking technologies, as shown below in Table 1. The ceiling tracker can track the position of one ultrasonic beacon to a resolution of about 1 cm at 20–50 Hz. The outdoor RTK GPS + GLONASS system has a maximum tracking resolution of 1–2 cm at an update rate of up to 1–2 Hz. Its accuracy may degrade to meter-level when fewer than six satellites are visible. If we lose communication to our RTK error correction base station, we fall back

TABLE 1 Area, Accuracy and Update Rates for Several Tracking Technologies We Use

	Coverage	Accuracy	Update rate (Hz)
IS 600 Mark II ¹	3 m × 3 m	1 mm–1 cm	20–50
GPS + GLONASS ²	worldwide	10–20 m	1–5
RTK GPS + GLONASS ³	near base station	1–5 cm	1–5
DRM ⁴	modeled area	1–2 m	step rate
Infrared ⁵	variable	~1 m	2

¹one crossbar with wireless beacon in position-only mode.

²requires line of sight to at least four satellites.

³requires line of sight to at least five satellites, and a base station.

⁴as we implement it here, requires model of environment.

⁵because cover roughly 7 m × 3 m elliptical zone—need to be overlapped.

to an uncorrected accuracy of 10–20 m. Both the dead-reckoning and the infrared tracking schemes offer accuracies at the meter level.

In our hardware implementation, the ceiling tracker is connected to a stationary tracking server, with its position updates relayed to the user's wearable computer over an IEEE 802.11b wireless network (Höllerer et al. 1999). The mobile user wears our testbed backpack system based on a Dell Inspiron 8000 with a 1.8-GHz Pentium III and an nVIDIA GeForce2 Go graphics processor. The user interface is presented on a Sony LDI-D100B see-through head-worn display.

As will be later described, our augmented reality user interface for intelligent navigational guidance automatically adapts to the levels of accuracy associated with these different position-tracking technologies, by monitoring the filter that coordinates their inputs. We have focused here on indoor tracking—on managing the ceiling tracker, infrared tracker, and the DRM tracker.

Wide-Area Indoor Tracking using Dead Reckoning and Environmental Heuristics

Our dead-reckoning system relies on local sensors and knowledge about the environment to determine its approximate position. Unlike existing hybrid sensing approaches for indoor position tracking (Golding and Lesh 1999; Laerhoven and Cakmakci 2000; Clarkson et al. 2000), we try to minimize the amount of additional sensor information to collect and process. The only additional sensor is a pedometer, in the form of Point Research Point-Man Dead-Reckoning Module (DRM) (Judd 1997)—the orientation tracker is already part of our mobile augmented reality system. 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 IS 300 Pro hybrid, inertial/magnetic orientation tracker, which is more accurate than the DRM's built-in magnetometer.

Unlike some (Lee and Mase 2001) who use digital compass information for their heading information, we have a much more adverse environment. Figure 1(a) illustrates the problems we had using magnetometer-based tracking. The plot corresponds to a user walking a rectangular path around the outer hallways of the sixth floor of our research building, using the IS 300 in hybrid (inertial + magnetic) 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 presence of a magnetic resonance imaging device for material testing two floors above us. Since the IS 300 affords the option of using it in inertial-only mode, we chose to use that mode, and to correct both for the resulting drift, and for the positional errors associated with the pedometer-based approach, by means

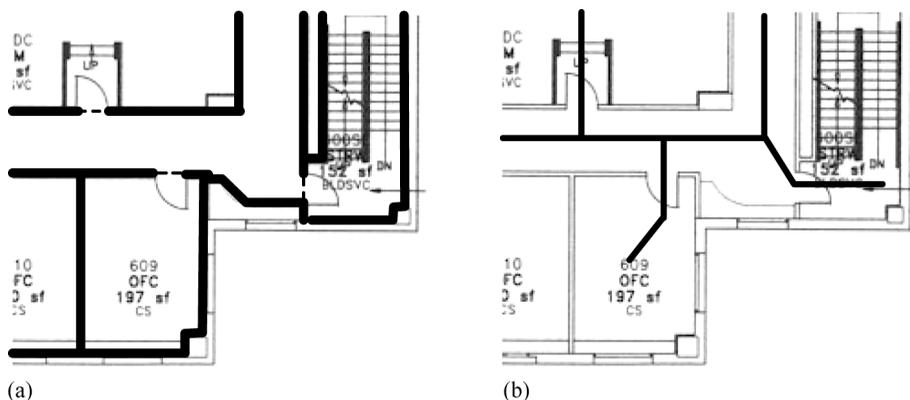


FIGURE 2. Two different representations of a small part of our building infrastructure, as used in the dead-reckoning-based tracking approach: (a) spatial map and (b) accessibility graph.

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 walls and other obstacles in a two-dimensional, top-view representation of the environment. Doors are represented as special line segments (denoted in the figure by the dashed lines connecting the door posts). Each step impulse registered by the pedometer generates a “step vector” in our software, the length of which is user-configurable, and the heading of which is given by the orientation tracker. One of our heuristics is to then check the *spatial map* to determine if this step vector, applied to the previous position estimate, would cross an impenetrable boundary (e.g., a wall). If it does, the system has to resolve a contradiction. In our current approach, the angle of collision—that between the step vector and the (most angularly proximate) vector lying along the linear obstacle (e.g., wall)—is computed. If this angle is below a configurable threshold (we used 30°), the conflict is classified as an artifact caused by orientation drift and the orientation output of the IS 300 is software-adjusted to correspond to heading parallel to the obstacle boundary—we bounce off the wall, for instance.

If the collision angle is greater than that arbitrary threshold, the system searches for a nearby segment on the *accessibility graph* that is not separated from the current estimate of user position by an impenetrable boundary, and is the closest match to the current heading estimate. That is, since the position estimate is most likely in error, the system determines where the user might really be located, so that his last step would *not* cross an impenetrable barrier. The system adjusts the position and orientation estimates so that the last step vector aligns with the solution edge of the accessibility graph and hence does not cross any barrier.

Doors are special cases—semi-impermeable barriers. First, expecting positional error, we define effective door segments as somewhat wider (currently one meter) than the physical doorframe. In case of a “door event” (the step vector crossing a door segment), the angle of collision is determined. As above, if the angle is below our arbitrary threshold, the system assumes it “shut,” and “bounces” the user away. If the angle is greater than (currently) 60° , the system assumes that the user is really passing through that door—adjusting his position only if passage was through the virtual extension of the door’s physical width. 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) 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.

This method is targeted mainly at environments with clear-cut passage constraints, like hallways and laboratories in which navigation is limited by desks and cubicles. With less constrained spaces, it would become important to model “typical walkways,” in order to form an adequate accessibility graph.

Tracking with Infrared Beacons

In contrast to the dead-reckoning approach described in the previous section, our infrared-based tracking method (Hallaway et al. 2003) uses a collection of strategically placed infrared beacons. These beacons, manufactured by Eyeled GmbH, broadcast a configurable, numerical ID, twice per second, at a 2400-baud data rate.

Butz and his colleagues at Eyeled have investigated architectures that map each beacon to a single logical entity near which it is positioned (Butz et al. 2000), such as a booth on a conference floor or an exhibit in a museum. When a *single* beacon signal is received, their systems infer that the user is near the logical entity to which that beacon maps. Ambiguity arises if *multiple* beacons with conflicting IDs are received. To avoid this, any overlapping beacon volumes must share the same ID or logical mapping—for instance, to expand a particular logical volume beyond that serviced by a single beacon.

In contrast, our tracking system—though coarse, in its attempt to minimize cost—aspires to a finer level of granularity than that afforded by systems intended to answer the question: “Which single beacon am I receiving, so what am I near?” (Butz et al. 2000; Starner et al. 1997). Each beacon has a unique ID, but we do not map that ID to a logical entity, nor do we stop at simply associating it with the volume over which it broadcasts. Rather, we

design beacon layouts that *strategically create* overlaps. Applying the operations of intersection and subtraction to these zones of influence (ZOIs), we partition the tracked area as uniformly and as finely as we are able, given the area to be covered and the number of beacons available for that coverage.

Our tests, and those of Eyeled, show these beacons as having a ZOI that conforms reasonably well to an ellipsoid, at one end of whose major axis is the beacon. With our coarse-tracking goals, we found it sufficient to model the ZOIs as ellipsoids. Given the nature of navigation indoors, our current experimental model operates in 2D—on the elliptical intersections of these ellipsoids with a plane parallel to the floor on which users are tracked. Once layout-strategy decisions are made, we store the modeled elliptical-zone poses in a configuration file. Figure 3 shows several layouts we have considered, (b) being the one we currently use in our laboratory, which involves ten inexpensive beacons.

An array of infrared “dongles” (Extended Systems XTNDAccess sensors) watches the beacons. In our experiments, we mounted the dongles to a helmet, although we anticipate attaching them to the upper posts of our backpack frame. The dongles are multiplexed into the mobile computer via

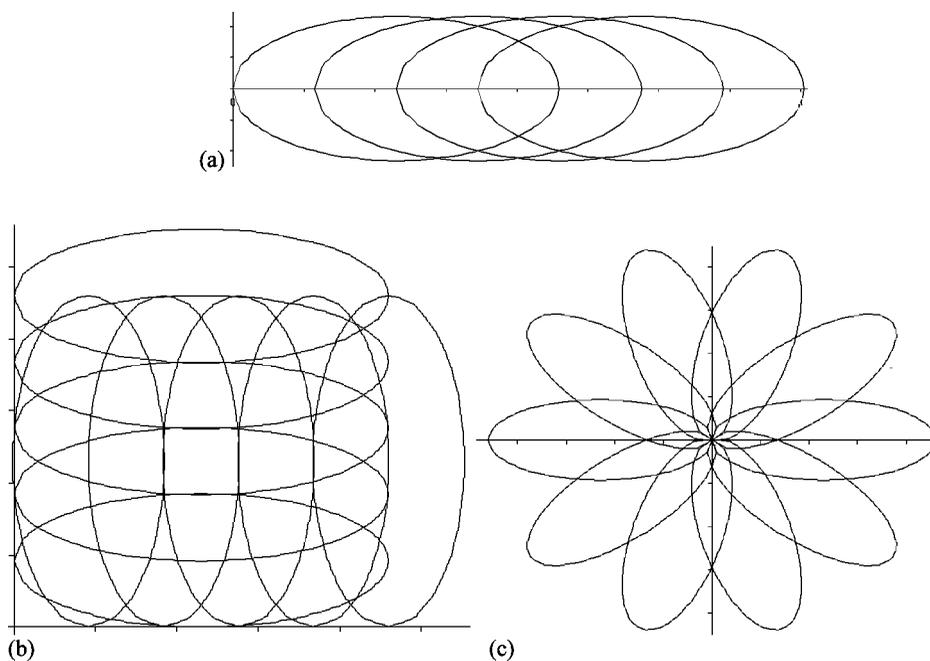


FIGURE 3. Efficient layouts for: (a) hallway or long, narrow room, (b) square room or section, and (c) round room with finer detail toward center.

a Socket Communications ruggedized PCMCIA card/adaptor cable that terminates in four DB-9 jacks. The results we present here were obtained using four dongles, mounted in a more or less planar fashion, oriented 90° apart.

Our low-level infrared dongle driver sets each dongle to receive the 2400-baud data rate at which the beacons broadcast their unique IDs. We should note that, to minimize the cost and complexity of our system, the beacons are not networked in any way: They operate without any synchronization, with clocks that likely drift with respect to one another. Hence, despite the fact that their brief, broadcast “bursts” are separated by nearly a half second of “silence,” there is a non-zero probability that during certain brief periods, a pair of beacons in the system may be in temporal collision. The dongle drivers currently address this concern by maintaining a lookup table of legitimate beacon IDs, ignoring broadcasts not found in it. Given our situation—using ten beacons with IDs from one to ten—the probability of two colliding signals appearing to a dongle as the broadcast of a legitimate ID seems vastly improbable. Moreover, it should be noted that not all potentially colliding pairs of beacons have spatially overlapping ZOIs. For those that do not, there will never be a conflict. Additionally, some pairs of beacons may have ZOIs that overlap, but are oriented in significantly different directions. Our receiver arrangement, which consists of several receiver dongles oriented in different directions, might be reached by signals from such beacons simultaneously, but no single dongle in our receiver arrangement will see both of the signals—the user might be in the intersection of temporarily colliding beacons, but no dongle (driver) will be so confused.

A higher-level driver maintains a working set of IDs “currently” received across all installed dongles during a brief, sliding time window, since there is nearly one-half second between each ID reiteration. Given this beacon-ID set, the higher-level driver invokes a method on an “area collection” object, and retrieves from it an area fragment to which that ID set maps.

We have developed an initialization algorithm for this area collection that pre-computes *two* sets of area fragments, given a coverage universe and a set of elliptical ZOI poses. The first is a true partition of that universe into “cells.” Each cell is generated by taking the intersection of the set of ZOIs mapped to by the beacon-IDs *received*, and then also subtracting the remaining ZOIs, whose beacon IDs are *not* received. Often these cells are empty, non-singular, or too small to inspire measurement confidence, so our algorithm also pre-computes a second set of simple intersections—the intersection of those ZOIs whose beacon IDs are *received*, without regard for those not received. Each such intersection fragment is always singular. It is also always a superset (often proper) of, and is less frequently empty than, its corresponding cell.

In Figure 4, we present a screen-shot of our test program at the end of a typical example of the many walk-arounds we tracked using this infrared

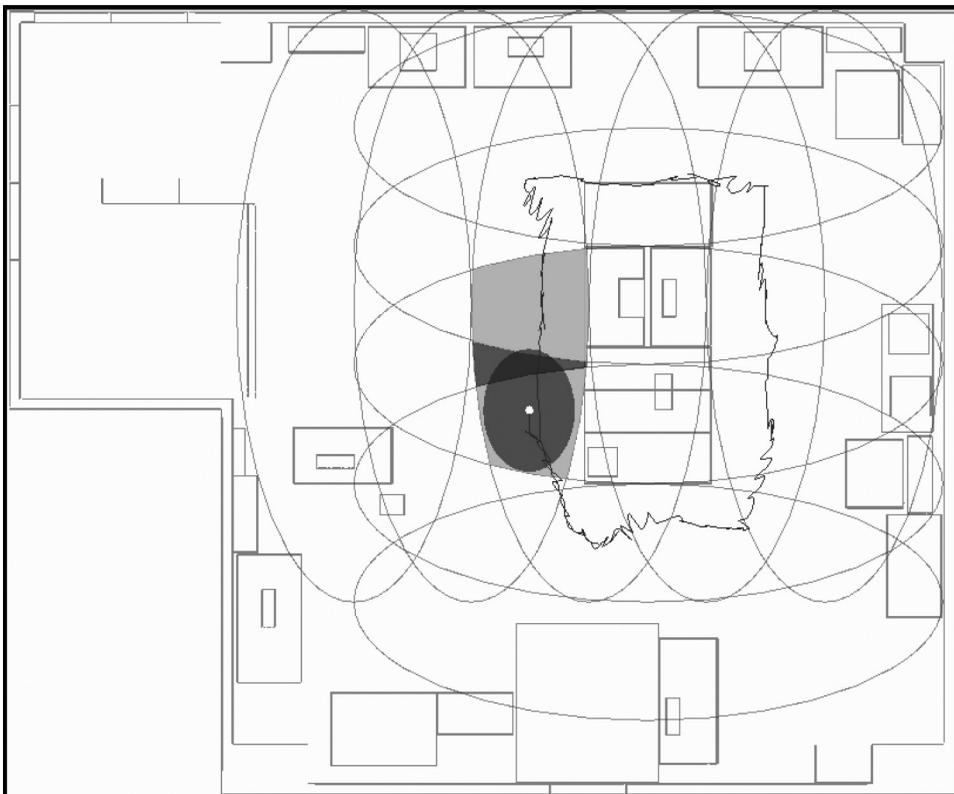


FIGURE 4. One of many tracked traversals of a rectangular path around the tables in the center of our lab: The “cell” fragment is dark gray, its lighter gray superset fragment is the intersection, and the transparent gray ellipse with the white estimate dot at its centroid is the ellipse of confidence.

system in the context of our lab. The *intersection* area fragment is rendered in those images in medium gray, and is the larger of two fragments, bounded by always convex elliptical segments. The *cell* area fragment is the intersection’s (usually) smaller subset, in darker gray, the bounds of which may also include concave segments. The later-discussed “ellipse of confidence” appears as a transparent gray ellipse, with a white estimate dot at its centroid.

We are experimenting with various policies of fragment usage for measurements. Current experience suggests that using the cell fragment generated by the full knowledge of beacons *not received* often produces measurements that are too specific and occasionally too far from the current consensus position to be believed—in short, we get noisy results because we cannot rely on the assumption that one of our receiver dongles will invariably pick up a signal from every beacon whose ZOI the receiver is currently in. While we will continue our investigations, the images presented in this paper are the result of defaulting to the intersection area fragment.

Observing many fragments, we noticed that always using their centroids as x - y measurements could result in position estimates that jumped more erratically than desirable, especially with larger intersections. We currently handle this potential “noise” in three ways. First, we have implemented a Kalman filter (Kalman 1960). Using an *adjusted* fragment’s axially aligned bounding box (see next subsection), its centroid provides the measurements for x and y , and some configurable ratio of its height and width are the basis for the x and y variances—all necessary filter inputs. Second, we maintain a configurable cap on the dynamic velocity values used by the filter’s state-transition computations. Third, we proceed to further leverage the Kalman filter corrections by maintaining an axially aligned “ellipse of confidence,” the dimensions of whose bounding rectangle are in some configurable, constant ratio to the standard deviations we calculate from the filter’s output. This ellipse of confidence is shown in Figures 4 and 5 as a transparent gray

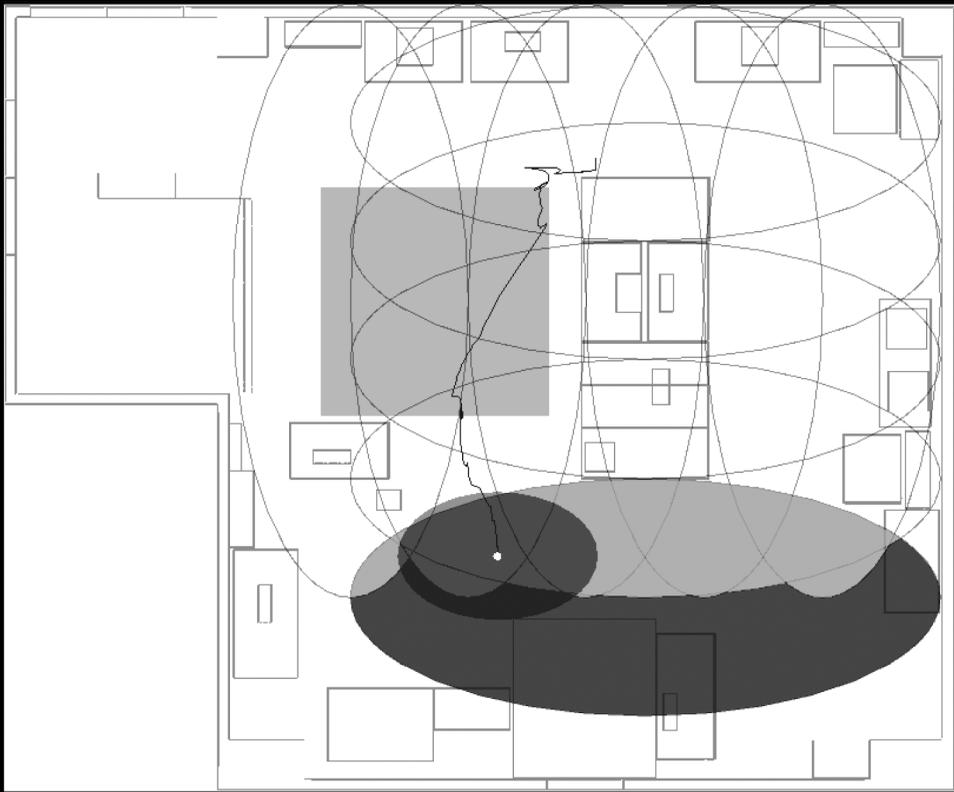


FIGURE 5. An example of the meta-tracker “handoff,” first from our infrared tracker to the ceiling tracker, and then back again. The light gray shaded rectangle shows where the ceiling tracker is in range. The handoffs are easy to see within those bounds. Other shadings are as in Figure 4.

ellipse with a white estimate dot at its centroid. We *adjust* (above) the area fragment supplied for the next measurement by intersecting it with the current ellipse of confidence. Since the receiver is most likely inside the ellipse of confidence, *and* is very likely inside the next supplied area fragment, its position would seem to be most likely within the *intersection* of the two. Certainly, if not the case, some near-future update adjusting the effects of that assumption would be doubtless forthcoming.

Managing Multiple Tracking Systems

Our experiences with filtering the infrared tracker output suggested two ideas: (1) using the variance outputs from such a filter to address the problem of how to structure the communication between a tracker’s driver level and the application’s user-interface and (2) employing some form of a Kalman filter to act as a “meta-tracker,” a device contrived to manage multiple, simultaneously running tracking systems.

We had already been investigating ways to make diverse tracking systems work together more or less seamlessly. Applying something like a Kalman filter to sensor outputs from multiple hardware tracking solutions, we reasoned, would give the systems designer the ability to avoid making explicit, error-prone, binary decisions about when to totally ignore input from one system and start depending entirely on that from another. Rather, the software system might feed the “meta-tracker” filter with estimates from all systems contemporaneously, and the standard deviations of error accorded the estimates from each system would cause them to be appropriately weighted in the correction cycles within the managing filter.

For our initial explorations using this approach, we employed an Inter-Sense IS 600 Mark 2 ceiling tracker, with a single, wireless ultrasonic beacon, for our relatively small-area, precision tracker. We paired it with the experimental infrared tracker we describe above, as a coarse-tracking, wider-area alternative. We updated the filter at 40 Hz. not only with the infrared estimates, but also with input from the ceiling tracker, whenever its mobile beacon was in range of the receiving crossbars. The ceiling tracker’s base unit was connected to a desktop computer, from which we forwarded its updates to our mobile notebook computer with a simple, custom server that sent UDP updates through the wireless network.

As can be seen in Figure 5, these “handoffs” worked rather well—the filter ensuring that transitions to and from the coarser tracking mode did not happen with an instantaneous leap from one mode’s current measurement to that of another’s. On the side of our lab where the ceiling tracker and the infrared coverage areas overlapped, the beacons were at the far extremes of their ranges, so somewhat less reliable, but this actually served to make the handoff more visible. Note from Figure 5 that continuing the

infrared updates with even the noisiest of data during the ceiling tracker's domination, was not visibly detrimental to the aggregate estimates.

ADAPTIVE AUGMENTED REALITY USER INTERFACE

Our experimental augmented reality user interface, implemented in Java3D (Deering and Sowizral 1997) is an adaptive one, focusing on the user's navigational needs. When the user is under the ceiling tracker, we exploit its higher accuracy by overlaying well-registered labels and sometimes a wire-frame model on such objects as rooms and doors (Figure 6).

In our experiments with the meta-tracking filter implementation described above, when the user moves out of range of the ceiling tracker, position-tracking dominance is shifted to the infrared tracker. The filter exposes variance data for each dimension of measurement it manages. As it retrieves the estimates, it needs to update its camera transformation, for instance, our user interface can also poll the filter for its current levels of confidence in those estimates. When position-estimate standard deviations rise

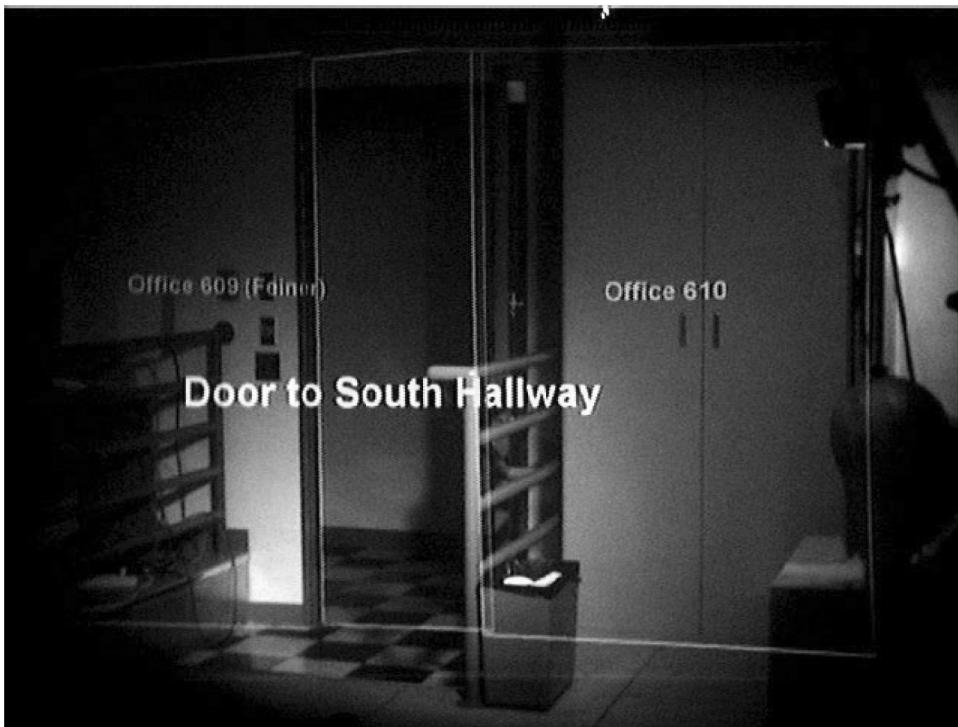


FIGURE 6. Augmented reality user interface in accurate tracking mode (imaged through optical see-through head-worn display). Labels and features (a wireframe lab model) are registered with the physical environment.

above a configurable threshold for a reasonable time interval, the user interface can use this event to change to a mode better reflecting its diminished certainty of position.

In one such rudimentary interface, we notify the user that this is happening by first replacing the registered world overlay with a World in Miniature (WIM) (Stoakley et al. 1995) model, but at full world-scale. That model is then animated in translation and scale, down to its normal position and miniature size (Pausch et al. 1995). During the brief animation, the user doesn't have any helpful augmentation, but he does have time to recognize a coherent shift between well-registered, world-scale augmentation, and largely unregistered, miniature-scale augmentation in the WIM.

Pairing either of our two alternative position-tracking solutions (the DRM-based method or our IR-beacon architecture) with the IS 300 Pro orientation tracker seemed a very useful way to bridge the gaps. This pairing afforded significantly more accurate orientation tracking than position tracking, however. We wanted to reflect tracking granularity in the interface itself, and to avoid confusing the user with misplaced augmentation. Considering this, we found the idea of a WIM a nice way to express the relatively superior orientation accuracy under such circumstances. This WIM, an alternative approach to another we presented (Bell et al. 2002), 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. The superior orientation tracking supports this world alignment—which is clearly evident to the user—but the miniature nature of this interface obviates the need to register augmentation with the world. The only way positional tracking error might be revealed would be in any (miniaturized) deviations of the user's avatar from her true WIM-frame position.

Related work on navigational interfaces (Darken and Cevik 1999) explored different ways of presenting 2D and 3D map information to a user navigating in a virtual environment. It was concluded that while there is, in general, no best scheme for map orientation, a self-orienting “forward-up” map is preferred over a static “north-up” map for targeted searches.

The WIM is a 3D extension of the “forward up” 2D option in Darken's and Cevik'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, nor does it require the attention of a tracked hand or arm to position it. Moreover, if desired, the WIM can exceed the bounds of the head-worn display'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.

Figures 7 and 8 show the user interface after one such transition to coarse position tracking and the WIM interface. Because the head–body alignment is relatively constant between these two pictures, the position of the projected WIM relative to the head-mounted display is similar in both pictures, but the differing position and orientation of the body relative to the world show the WIM’s world-aligned characteristics.

These images also include world-situated route arrows that point the way along the path to a location that the user has requested (in this case, a nearby 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 the see-through head-worn display. (A more legible view of a path is in shown in Figure 10, which is a direct frame-buffer capture, and therefore doesn’t show the real world on which the graphics are overlaid.)

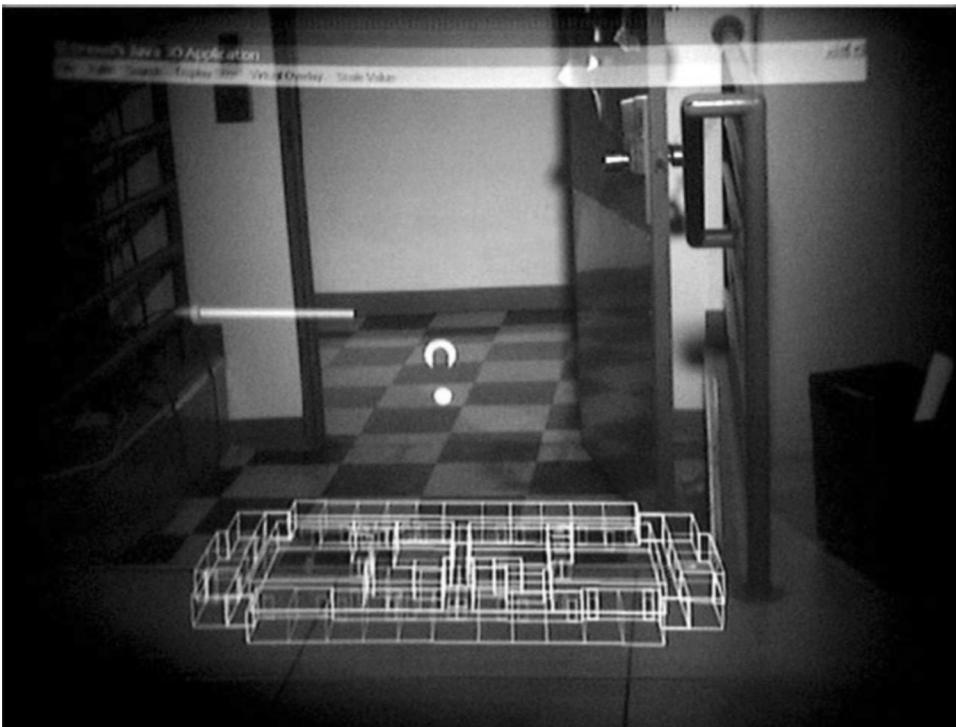


FIGURE 7. Augmented reality interface in coarsely tracked mode (imaged through optical see-through head-worn display), presenting a body-stabilized, world-aligned WIM and world-space arrows.

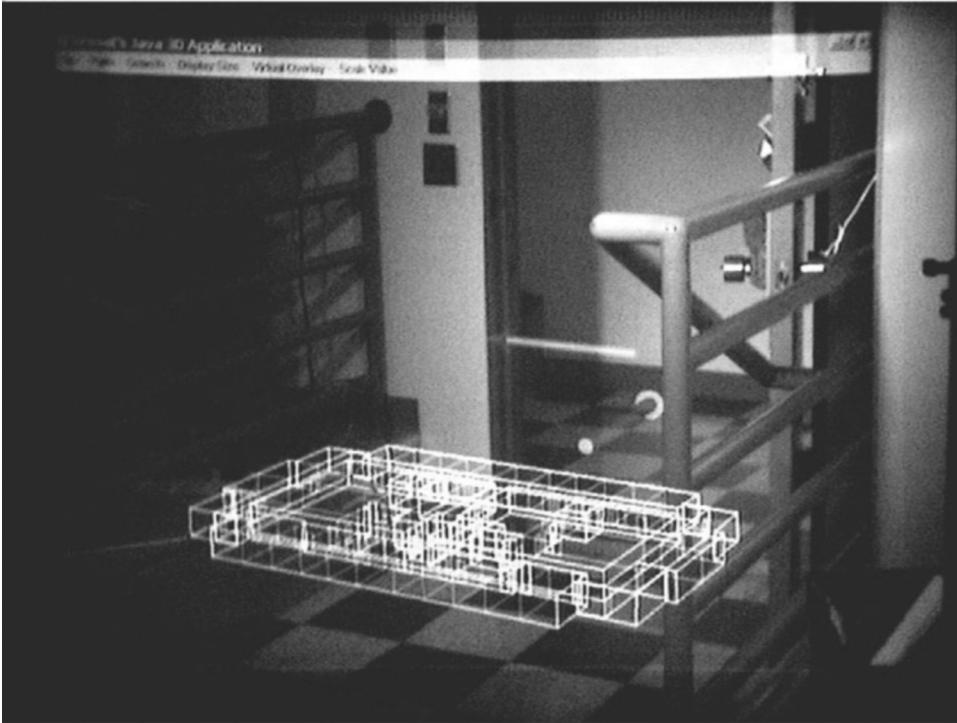


FIGURE 8. Augmented reality interface in coarsely tracked mode (imaged through optical see-through head-worn display), with the user at a different position and orientation, demonstrating the world-alignment of the WIM.

INTELLIGENT NAVIGATION AIDS

Users of augmented reality, navigational interfaces may often wish to pose questions about the locations of things which—in less than familiar territory—may be uncertain of existence and cannot be particularly named. The user may know a *kind* of thing he seeks, but sometimes he may not know whether such a thing is reasonably accessible, nor how he should ask for it. Moreover, a user on foot, who, for instance, asks for the nearest candy machine, would likely prefer being directed to a snack machine steps away—which happens to lack candy bars—to getting information about a candy machine miles away. Systems that answer particular queries too literally can be less useful and more frustrating.

Knowledge Representation

To address such considerations, we decided to experiment with a description logic (Donini et al. 1996) implementation. For a simple example of its function, notice that in Figure 9 the user uses a menu to request the path

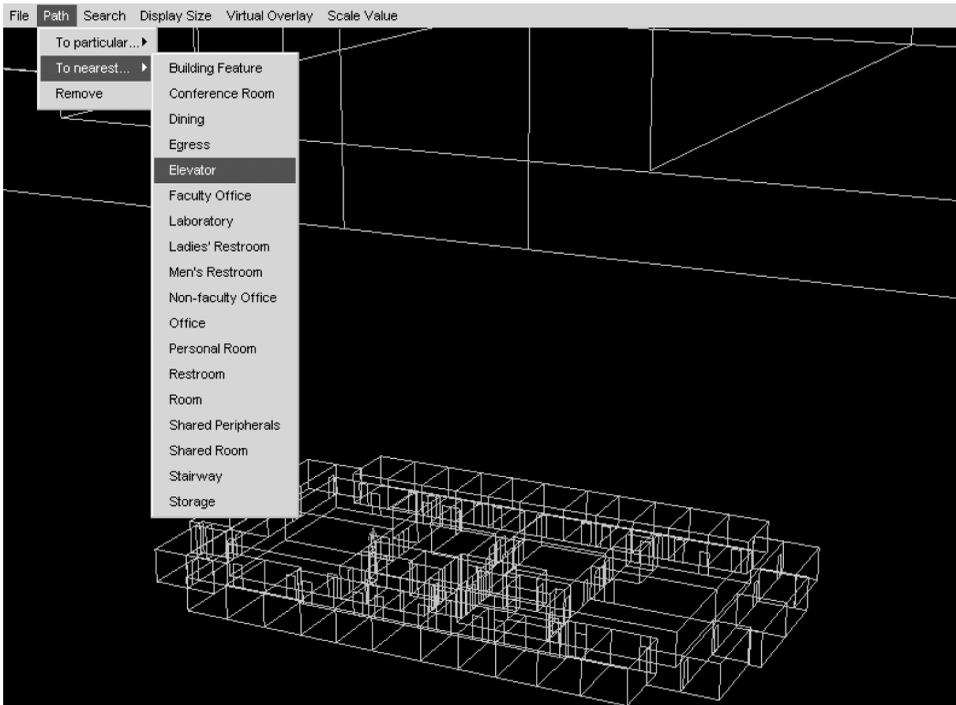


FIGURE 9. Intelligent navigational guidance with the user beginning a query.

to the nearest elevator. The system responds to this query with two solutions. The first of the two is represented in Figure 10 as a larger-diameter, brighter 3D path to the most literal solution—the nearest elevator. The second is plotted as a medium-diameter, somewhat dimmer path to the nearest stairway. A reasoning component infers that, although the user has explicitly

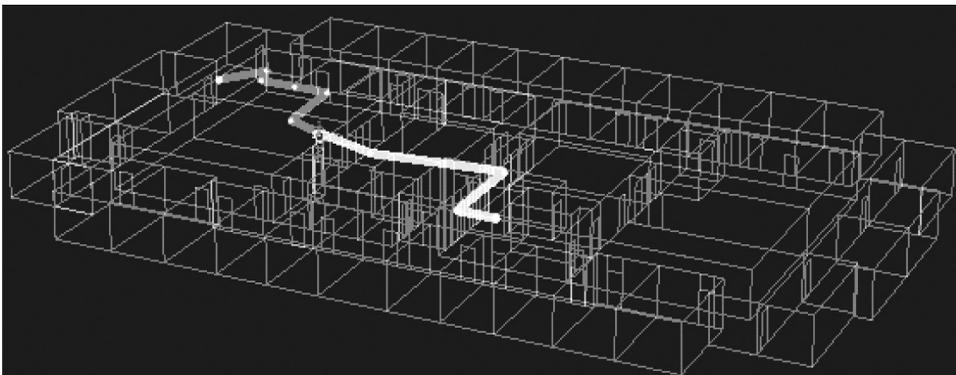


FIGURE 10. Intelligent navigational guidance—query resulting in different solution paths in the WIM.

specified an interest in elevators, she might actually be interested in any means of egress. Since the stairway is closer, it is presented as well.

Our system's knowledge of the physical domain and its resources resides in a persistent database (Höllner et al. 1999). At load time, tables in that database are parsed into structures necessary to our simple inferencing system. In the domain described here, the "concepts" (Donini et al. 1996) are the classes of resources found on the floor of the building enclosing our lab. At the lowest level, concepts include things such as "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 (Donini et al. 1996), which handles terminological knowledge about concepts, includes a list of these concepts, each associated with its subsuming parent.

In our current implementation, the database encodes simple assertions—"constructors" of these "isA" subsumption "roles" (Donini et al. 1996). Reasoning might be automated that would infer subsumptions, and more general relationships among concepts, by operating on the properties of each concept, but we have not yet implemented such. Our system does, however, automatically generate the hierarchy tree from these individual subsumption assertions.

The ABox (Donini et al. 1996), 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. As in the concepts discussed earlier, our database currently simply asserts the membership of each individual in its most specific concept. Given the asserted memberships, though, our system proceeds to automatically infer—at load time or during runtime—the more general concept memberships for each individual entity.

A metrical concept we employ, outside this hierarchy of resources, is the *PathNode*. To support the graph searching techniques of A* or Dijkstra's Algorithm (Dijkstra 1959), we represent the graph (of possible paths to resources) in our database and data structures as a set of these nodes. This is the same data structure used for the accessibility graph we described in the third section. 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 Figures 9 and 10, 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 an individual resource named “South Elevator.” Since, in this case, the ply choice was greater than zero, though, the system went on to note that the concept elevator is subsumed by that of egress, and hence proceeded to evaluate members of that parent concept. In addition to elevator, egress subsumes the concept stairway, so since the “East Stairway” is nearer the user than the “South Elevator,” a path is also plotted to it, as a second solution, with somewhat less prominent graphical presence. Since the ply count was actually two here, 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.

CONCLUSIONS AND FUTURE WORK

We have described a mobile augmented reality system that uses several different modes of tracking user position—modes that differ significantly in accuracy. One of these modes employs a dead-reckoning module, which makes use of pedometer and orientation information, applying corrections derived from knowledge about the user's immediate environment, in the form of a spatial map and an accessibility graph.

Another mode is afforded by our experimental infrared tracker, which infers position from the set of infrared signals it receives, making spatial inferences over the modeled volumes to which each signal in that set maps. The installation we have described frankly outperformed our expectations, once reasonably filtered. The accuracy of this device seems to be in direct proportion to the density of the beacon distribution. We would like to do performance testing with several layouts, and find a sound means of expressing the accuracy level that can be expected from this device, given a particular layout scheme.

One concern we hope to address more rigorously regards the Kalman filter we have implemented to smooth the infrared tracker's output. As is not uncommon, that filter is being applied to a domain in which some of its assumptions arguably do not hold. Kalman filtering assumes that the probability distribution of each measurement is Gaussian. One can reasonably

assert that having received signal set S , the probability of being in, say, the square decimeter of the fragment furthest from the operative beacons, is not equal to—indeed is surely quite a lot less than—the probability of being in the nearest one. If so, the probability distribution of the reception-location across these elliptical ZOIs, or indeed their fragments, is certainly non-Gaussian. That the filter performs as well as it does, in our view, merely serves to highlight the essentially forgiving nature of Kalman’s algorithm—another example of the benefits of applying it where some of its theoretical assumptions may not hold.

A number of user interface questions might be effectively 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 view frustum regardless of where the head looks (Bell et al. 2002)? 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 conjectured in the design of our system here, might it be better to situate the WIM with its centroid (indeed its entire volume) somewhat in front of the user’s body? Immersing the user directly in a WIM might 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, miniature “share” of the WIM’s volume—and the difficulty of determining exactly where in the WIM the user’s world-sized body really is?

We hope to soon complete the integration of our outdoor tracking system into the mix fed to the Kalman filter. We are also interested in augmenting or replacing the DRM with some other accelerometer-based source and software processing. Including altimetry (coarsely supported by the DRM) would help us track position in elevators or stairwells. Our laboratory’s demos, we hope, will soon become full walk-around mobile augmented reality applications that—without changes of gear or pressing of buttons—are capable of going from the well-tracked zones of our lab, across its remainder, out the door, through the halls, down the elevator, through the lobby, and out the front door, all stages of which are serviced by some usable level of tracking and with the user interface intelligently responsive to what it knows about the level of confidence it should accord current tracking estimates.

REFERENCES

- 3rdTech Corp. 2002. Available from <http://www.3rdtech.com/HiBall.htm>
- Bahl, P. and V.N. Padmanabhan. 2000. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of the Joint Conference of the IEEE Computer and Communications Societies, (Infocom 2000)*, pages 775–784.

- Beadle, H.W.P., B. Harper, G.Q. Maguire, and J. Judge. 1997. Location aware mobile computing. In *Proceedings of the IEEE/IEE Int'l Conference on Telecommunications, (ICT '97)*, pages 1319–1324, Melbourne, Australia.
- Bell, B., T. Höllerer, and S. Feiner. 2002. An annotated situation-awareness aid for augmented reality. In *Proceedings of the ACM Symp. on User Interface Software and Technology, (UIST-2002)*, pages 213–216, Paris, France.
- Borenstein, J., H. Everett, L. Feng, and D. Wehe. 1997. Mobile robot positioning: Sensors and techniques. *Journal of Robotic Systems* 14(4):231–249.
- Bowditch, N. 1802. Dead Reckoning. In *The American Practical Navigator, an Epitome of Navigation*.
- Butz, A., J. Baus, and A. Krüger. 2000. Augmenting buildings with infrared information. In *Proceedings of the IEEE and ACM Int'l Symp. on Augmented Reality, (ISAR 2000)*, pages 93–96, Munich, Germany.
- Butz, A., J. Baus, A. Krüger, and M. Lohse. 2001. A hybrid indoor navigation system. In *Proceedings of the Int'l Conference on Intelligent User Interfaces, (IUI 2001)*, pages 25–32, Santa Fe, NM, USA.
- Castro, P., P. Chiu, T. Kremenek, and R. R. Muntz. 2001. A probabilistic room location service for wireless networked environments. In *Proceedings of the Int'l Conference on Ubiquitous Computing (UbiComp 2001)*, pages 18–34, Atlanta, GA, USA.
- Clarkson, B., K. Mase, and A. Pentland. 2000. Recognizing user context via wearable sensors. In *Proceedings of the Int'l Symp. on Wearable Computers, (ISWC 2000)*, pages 69–75, Atlanta, GA, USA.
- Darken, R. and H. Cevik. 1999. Map usage in virtual environments: Orientation issues. In *Proceedings of the IEEE Virtual Reality, (VR'99)*, pages 133–140.
- Deering, M. and H. Sowizral. 1997. Java3D Specification, Version 1.0.: Sun Microsystems, 2550 Garcia Avenue, Mountain View, CA 94043, USA.
- Dijkstra, E.W. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik* 1:269–271.
- Donini, F.M., M. Lenzerini, D. Nardi, and A. Schaerf. 1996. Reasoning in description logics. In *Principles of Knowledge Representation, Studies in Logic, Language and Information*, ed. G. Brewka, CSLI Publications.
- Ekahau, Inc. 2002. Accurate Positioning in Wireless Networks, Ekahau Positioning Engine 2.0 2002 [cited July 2002]. Available from <http://www.ekahau.com>
- Foxlin, E., M. Harrington, and G. Pfeifer. 1998. Constellation: A wide-range wireless motion-tracking system for augmented reality and virtual set applications. In *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques, (SIGGRAPH '98)*, pages 371–378.
- Getting, I.A. 1993. Perspective/Navigation -The global positioning system. *IEEE Spectrum* 30(12):36–38, 43–47.
- Golding, A.R. and N. Lesh. 1999. Indoor navigation using a diverse set of cheap, wearable sensors. In *Proceedings of the Int'l Symp. on Wearable Computers, (ISWC '99)*, pages 29–36, San Francisco, CA, USA.
- Griswold, W.G., R. Boyer, S.W. Brown, T.M. Truong, E. Bhasker, G.R. Jay, and R.B. Shapiro. 2002. *Active Campus-Sustaining Educational Communities through Mobile Technology*. San Diego, CA: Univ. of California Press.
- Hallaway, D., T.Höllerer, and S. Feiner. 2003. Coarse, inexpensive, infrared tracking for wearable computing. In *Proceedings of the Int'l Symp. on Wearable Computers, (ISWC 2003)*, pages 69–78, White Plains, NY, USA.
- Hightower, J. and G. Borriello. 2001. Location systems for ubiquitous computing. *IEEE Computer* 34(8):57–66.
- Höllerer, T., S. Feiner, T. Terauchi, G. Rashid, and D. Hallaway. 1999. Exploring MARS: Developing indoor and outdoor user interfaces to a mobile augmented reality system. *Computers & Graphics* 23(6):779–785.
- InterSense, Inc. 2002. IS-900 Wide Area Precision Motion Tracker 2001. Available from <http://www.isense.com>
- Judd, C.T. 1997. A personal dead reckoning module. In *Institute of Navigation's ION GPS*, September, Kansas City, MO.
- Kalman, R.E. 1960. A new approach to linear filtering and predictive problems. *Trans. ASME—Journal of Basic Engineering* 82 (Series D):35–45.

- Kato, H., M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana. 2000. Virtual object manipulation on a table-top AR environment. In *Proceedings of the Int'l Symp. on Augmented Reality (ISAR 2000)*, pages 111–119.
- Laerhoven, K.V. and O. Cakmakci. 2000. What shall we teach our pants? In *Proceedings of the Int'l Symp. on Wearable Computers, (ISWC 2000)*, pages 77–83, Atlanta, GA, USA.
- Lee, S.W. and K. Mase. 2001. A personal indoor navigation system using wearable sensors. In *Proceedings of the Int'l Symp. on Mixed Reality, (ISMR 2001)*, pages 147–148, Yokohama, Japan.
- MacIntyre, B. and E.M. Coelho. 2000. Adapting to dynamic registration errors using Level of Error (LOE) filtering. In *Proceedings of the Int'l Symposium on Augmented Reality, (ISAR 2000)*, pages 85–88, Munich, Germany.
- MacIntyre, B., E.M. Coelho, and S.J. Julier. 2002. Estimating and adapting to registration errors in augmented reality systems. In *Proceedings of the IEEE Virtual Reality, (VR 2002)*, pages 73–80, Orlando, FL, USA.
- Newman, J., D. Ingram, and A. Hopper. 2001. Augmented reality in a wide area sentient environment. In *Proceedings of the IEEE and ACM Int'l Symp. on Augmented Reality, (ISAR 2001)*, pages 77–86, New York, NY, USA.
- Pausch, R., T. Burnette, D. Brockway, and M. Weiblen. 1995. Navigation and Locomotion in Virtual Worlds via Flight into Handheld Miniatures. In *Proceedings of the SIGGRAPH ACM Conference on Computer Graphics and Interactive Techniques, (SIGGRAPH '95)*, pages 399–401.
- Raab, F.H., E.B. Blood, T.O. Steiner, and H.R. Jones. 1979. Magnetic position and orientation tracking system. *IEEE Trans. on Aerospace and Electronic Systems* 15(5):709–718.
- Starner, T., D. Kirsch, and S. Assefa. 1997. The locust swarm: An environmentally-powered, networkless location and messaging system. In *Proceedings of the IEEE Int'l Symp. on Wearable Computers, (ISWC '97)*, pages 169–170, Cambridge, MA, USA.
- Stoakley, R., M. Conway, and R. Pausch. 1995. Virtual reality on a WIM: Interactive worlds in miniature. In *Proceedings of the Human Factors in Computing Systems, (CHI '95)*, pages 265–272.
- Welch, G., G. Bishop, L. Vicci, S. Brumback, K. Keller, and D. Colucci. 1999. The hi-ball tracker: High-performance wide-area tracking for virtual and augmented environments. In *Proceedings of the ACM Symp. on Virtual Reality Software and Technology, (VRST '99)*, pages 1–11, London, U.K.
- Welch, G., and E. Foxlin. 2002. Motion tracking: No silver bullet, but a respectable arsenal. *IEEE Computer Graphics and Applications* 22(6):24–38.