IEEE CG&A, July/August 2003, pp. 14-18

Projects in VR

Editors: Lawrence Rosenblum and Michael Macedonia

Pervasive Pose-Aware Applications and Infrastructure

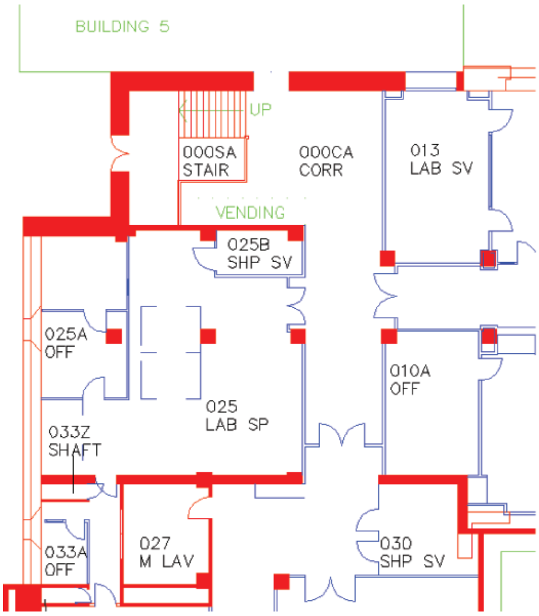
Seth Teller, Jiawen Chen, and Hari Balakrishnan

Massachusetts Institute of Technology

Location-aware or context-aware devices and applications let users view and interact with location-dependent information and resources. In outdoor settings, Global Positioning System-enabled applications with access to specialized geographic databases have had a significant impact on military operations, civilian navigation and surveying, commercial shipping and supply-chain management, aerial photography, precision agriculture, and many other areas over the past decade.

More recently, we and others have begun developing analogous applications for indoor use. Because GPS is often unavailable indoors, we developed a scalable infrastructure called Cricket,[1,2] based on active radio-frequency and ultrasound beacons and passive listeners, to support pervasive indoor location and orientation determination by handheld devices. In contrast to devices that report location alone, such as GPS receivers, a pose-aware device reports both its location and orientation.

Pose-aware devices, in concert with functional geometric models describing architectural spaces, enable a new class of indoor applications, including resource location, route finding, direct population and annotation of world models, and direct information overlay.



1 Functional geometric model with space identifiers (the numbers in each space), space types (for example, OFF stands for *office*), and adjacency information (such as, doorways and stairwells).

This article describes these applications along with the device infrastructure and algorithms required to support them. We demonstrate a few prototype devices and applications now underway and point to some future directions in which these techniques might evolve.

Pose awareness and extensions

Even in the absence of contextual information (that is, a database of the environment), pose awareness is useful. For example, with a display of your current and past pose sampled at some interval, you could retrace your steps back to some starting point, and therefore avoid getting lost. Things get more interesting, however, when we extend the notion of pose awareness in two ways: adding functional models and additional device capabilities.

First, we make the application context aware by providing it with access to a functional model (see Figure 1). This model contains a geometric representation of the environment in which the application is to be used, along with functional information delineating named, interconnected spaces and their contents. The database is readable, providing space information keyed on the user's pose, and also mutable, allowing the user to indicate and annotate objects in the world directly, as we later describe.

Second, we augment the pose-aware device with a laser range finder and digital projector. The range finder places a visible mark (a red dot) on some object that the user selects and reports the distance from the device to that object in metric units. The application uses this calculated range and the pose information to deduce the location of one point on the indicated object. The projector enables the application to overlay information, such as text or schematics, onto the object.

The applications we discuss in the following sections use various pose-aware devices with a functional geometric model.

Navigation

The most basic use of pose information is finding your way around inside an unknown environment. Even without a functional model of the environment, pose information can provide the direction and distance to your point of entrance, letting you retrace your route. We call a pose-aware device configured for navigation a *software compass* (see Figure 2).

If the pose-aware application has access to a func-

Published by the IEEE Computer Society

0272-1716/03/$17.00 © 2003 IEEE

tional model of the environment and its resources, it can provide resource location services—such as identifying the nearest bathroom, color printer, or coffee machine. The application can provide route finding as well, reporting the shortest or fastest route to a specified person, resource, or exit. It can parameterize this route according to the user's capabilities. For example, the route for a user in a wheelchair would avoid stairs and narrow elevators or doorways, traversing ramps instead.

Finally, if the application can access real-time environmental information—for example, from a distributed sensor network—it could generate routes dynamically, optimized to current or predicted conditions. Examples include avoiding heavily trafficked corridors during class or work shift changes, routing to food vendors with the shortest lines, and distributing people across unblocked escape routes during emergencies.

Functional model population

Creating CAD models of buildings, analyzing them to extract space type and adjacency information, and populating them with representations of furniture and other resources can be enormously time consuming, requiring significant human effort even with a specialized interface.[3] Moreover, the traditional CAD workflow places users at a desk with a workstation, typically far from the actual space that they are describing or annotating.

A pose-aware device lets users correct or augment a world model in situ, by indicating model components directly. We could use this capability to correct *as planned* CAD models, transforming them into more useful *as built* models, reflecting engineering changes between the design and construction phases. We could also use direct object indication for much more rapid model population with canonical object types—standard-issue furniture, appliances, and so forth. In a typical built environment, the list of resources is finite and comes from a small set of resource specifiers or types (printer, water fountain, bathroom, fire alarm, and so on). Thus, a user could simply indicate an object by pointing a pose-aware device, then specify the object class through a traditional interface (such as text, menu, or speech).

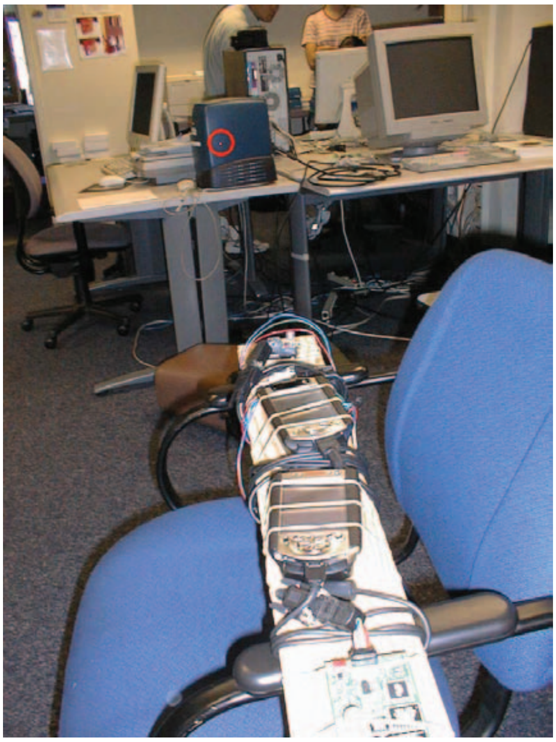
Virtual tagging

Builders and maintainers of physical infrastructure (surveyors, contractors, and so on) often make annotations directly on objects in the world. For example, the Dig Safe public works infrastructure (used by many cities in the Northeast US) employs technicians who use various sensing techniques to locate buried service lines, and then mark the surface accordingly with fluorescent paint (see Figure 3). Such markings are vulnerable to disappearance through erosion or subsequent construction activity.

In contrast, we propose a *software marker*, a pose-aware device in concert with a mutable database, to make annotations to the database's representation of the object (see Figure 4). These virtual tags can persist even if the environment changes. Also, read access to virtual tags becomes controllable in software, rather than being an unavoidable consequence of the tag's public nature. So, for example, a security consultant could mark faulty locks, fence gaps, and so on with annotations that would be invisible without a suitable device and access permissions to read the tags, avoiding any public announcement of the security issues. Also, the

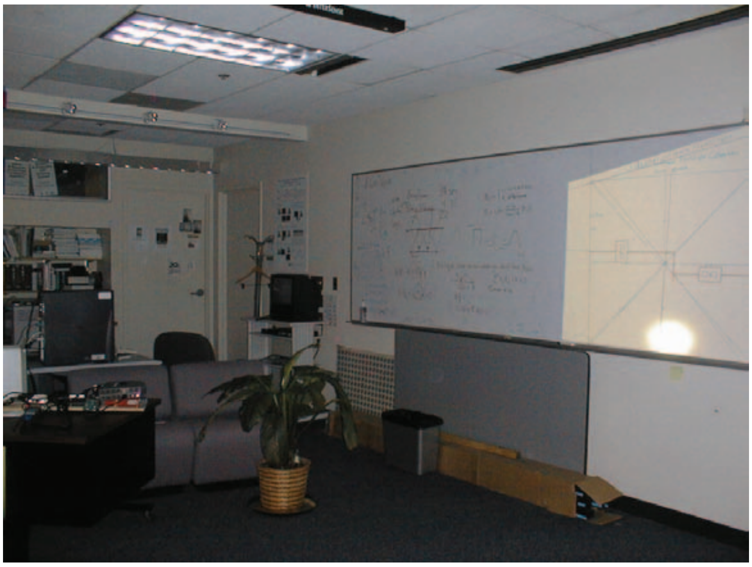
2 Prototype software compass. Two Cricket position listeners (at either end of board) separated by a fixed baseline and a handheld computer (center) fuse the position estimates into a single orientation estimate for the compass. A laser range finder is at the left.

3 A Dig Safe technician applying an annotation to a road surface.

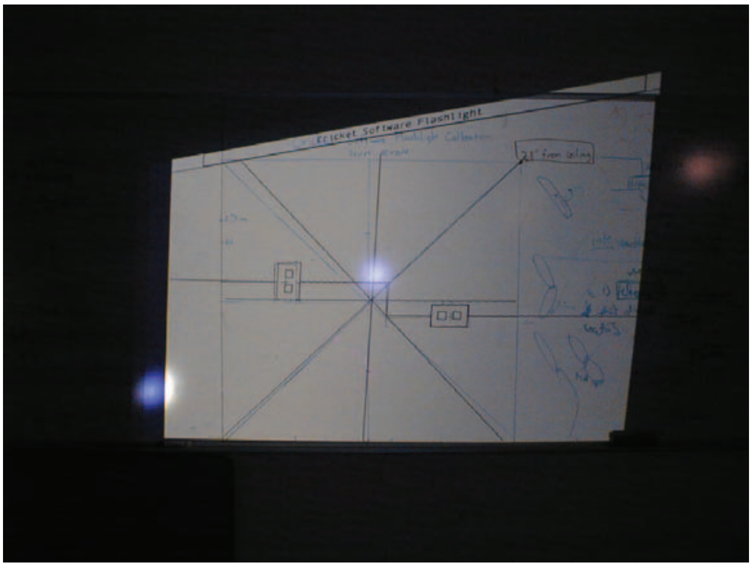


4 Software marker indicating a computer on a desk. The red dot cast by the laser range finder is circled.

5 Prototype software flashlight. A software compass augmented with a PDA capable of image generation, and a compact digital projector.



6 Prototype information overlay: a software flashlight (left, on the desk) overlays geometric information (geometric registration pattern and planned electrical outlets) onto an existing wall (right).



7 Detail view of the overlay; note the oblique projection.

application can filter the tag display to respect visibility constraints, in contrast to tagging approaches that trigger display based only on proximity.[4]

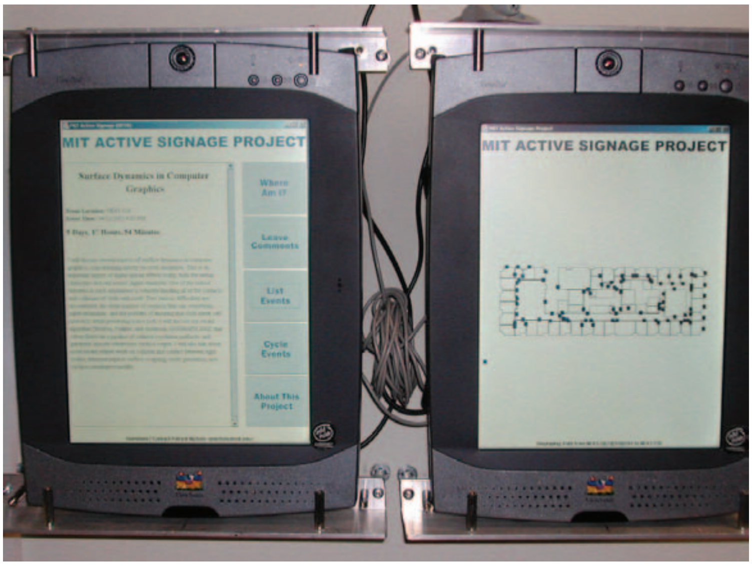
Another type of annotation is a site-specific maintenance request made by a physical plant worker or member of the general public. Traditionally, requesters would explain their problem in writing or by telephone, including a description and location. As in the previous model population scenario, a pose-aware device would simplify maintenance requests: the user need only indicate the problem area and supply a text or voice description of the maintenance issue for the newly-created virtual tag. Finally, annotations can themselves be geometric. For example, a user could sketch a desired routing for an electrical or network conduit on a wall, creating an as-planned CAD document, or correcting an as-built document, in situ.

Information overlay

With projection capability, the pose-aware device becomes a *software flashlight,* capable of direct information overlay by projecting metadata from the functional model onto surfaces or objects of interest (see Figure 5).

The overlaid metadata can take the form of text or geometric information. Overlaid text could describe the object's construction or emplacement date, maintenance history, or ownership information. Overlaid geometric information could depict hidden infrastructure (for example, electric mains or plumbing inside walls), or access or disassembly instructions. Overlays could also provide as-planned information, for example, CAD diagrams of new construction, including wiring, cabinetry, and so on (see Figure 6). One surprising and appealing aspect of direct overlay is that, for planar geometry, the projection need not be fronto-parallel, that is, the optical axis of the projector need not be perpendicular to the projection surface, but can be at any angle. Thus, the display requires no special geometric computation. A correctly situated and oriented projector whose intrinsic calibration parameters are known will produce perspectively correct rendering if the planar projection surface corresponds to the model geometry plane (see Figure 7).

Information overlay with a software flashlight addresses two limitations of traditional augmented reality that overlays textual or graphical information on a user's view through a transparent, head-mounted display.[5] First, transferring the information overlay function to a hand-held device enables intermittent rather than constant use (for example, flipping the eyepiece up). In contrast, the software flashlight can be put down when not in use. We think of it as a tool like others on your tool belt—for example a screwdriver or drill—rather than as an article of clothing that must be worn all the time. It's much more likely that lay users, such as contractors, will adopt it in the near term. Second, transferring the registration problem from the user's head to the device eases the burden on the tracking infrastructure. That is, the software flashlight doesn't require refresh rate updating, even when handheld, and needs refreshing far less frequently if it's rested on a static surface (for example, a table or countertop).

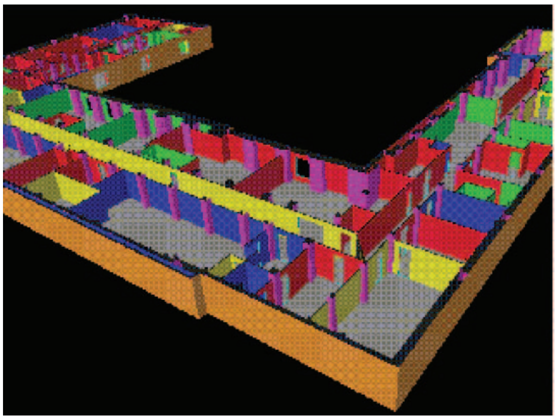


8 Prototype active sign displaying an event listing (left) and a route map (right) from the viewer's location to the event.

Active signage

Signage is an important aspect of most indoor environments, directing people to conference rooms, bathrooms, exits, and so on. Conventional signage is limited by nature to displaying static directional information and can't display time-dependent information (for example, talk announcements at a conference). We are developing pose-aware, wireless-networked active signs that display timely event and route information configured to the sign's pose and spatio-temporal context (see Figure 8). Thus, a talk announcement display would include a route from the sign's (and thus the viewer's) position to the talk's location. In emergency situations, the active signs can override routine event announcements to display location-dependent escape routes.

Our active sign prototype consists of two wireless tablet PCs physically secured in a fixed pose. At present, the sign is manually programmed with its own location and orientation, but future signs will become pose-aware through use of the Cricket infrastructure. We plan to deploy less expensive pose-aware mobile signs, so users can remove signs from their mounts and carry them within the environment. The pose-aware sign can then reconfigure its display of event and route information appropriately, as its position and orientation change.



9 3D floorplan, procedurally generated by extruding appropriate elements of a 2D floorplan.

Infrastructure: Models, devices, and algorithms

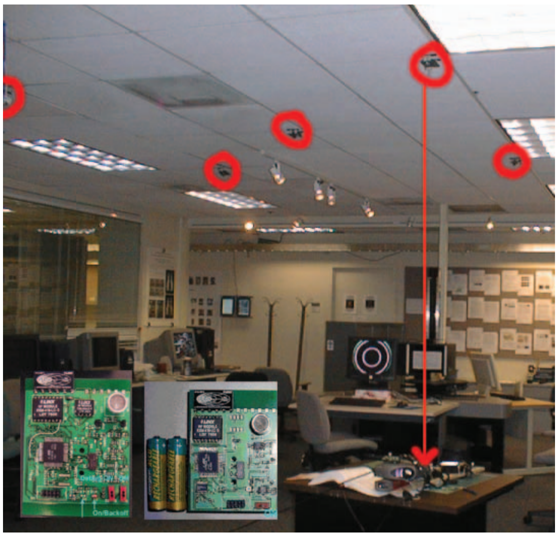
Instrumenting an extended indoor environment to support pose awareness and its associated applications requires a significant amount of infrastructure design and deployment. Aspects of this infrastructure include acquisition of functional geometric models, design and deployment of position- and orientation-determination capabilities, and scaling issues that arise in spatially extended deployments.

Functional geometric models

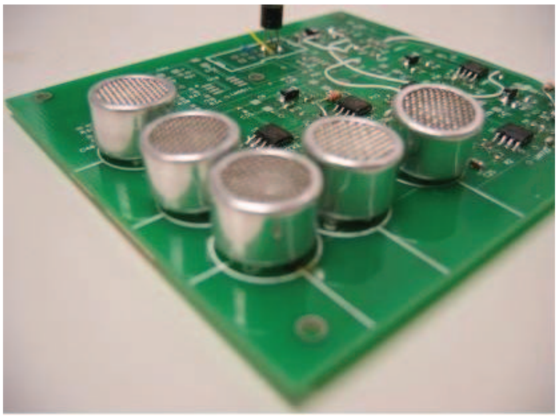
Creating geometric models from scratch is enormously time consuming even with specialized tools. The maintainers of many built environments have detailed CAD documents with overlaid room labels and space types. However, due to the variety of architects and contractors employed, the lack of standards for structuring such documents, and some other historical factors, the document set for a large collection of buildings can be strikingly heterogeneous. To use this data, we have adapted and extended a suite of processing and interpretation tools for 2D CAD floorplans.[6] These tools, at a minimum, extract delineated spaces and space types (offices, corridors, conference rooms, stairwells, elevator shafts) and adjacency information—linking two spaces only if they are physically adjoining and a person (on foot or in a wheelchair) can traverse directly from one space to the other. The tools also extrude incomplete, sometimes ill-formed 2D floor plans into well-formed 3D floor plans (see Figure 9).

Cricket infrastructure

The Cricket system for position determination[1] includes a collection of fixed beacons and mobile listeners (see Figure 10, next page). Each beacon broadcasts its own location in a specified (typically building-wide) coordinate system, using a radio-frequency band. Simultaneously, the beacon emits an ultrasound pulse. Any Cricket listener receiving both signals can deduce its distance (range) from the beacon from the difference in arrival times of the



10 Cricket beacons (on ceiling, circled) and listener (on table with arrow). Inset images show details of the beacon (left) and listener (right).



11 Prototype orientation-aware Cricket listener with multiple ultrasound receivers.

fast radio-frequency signal and slow ultrasound signal. With range from three or more beacons, a listener can determine its own position through trilateration.

Orientation-aware listeners

Cricket position listeners can infer their own location with an accuracy of a fraction of a meter. We extended the Cricket listener to estimate its orientation with respect to any fixed beacon by adding multiple ultrasound receivers to the listener (see Figure 11). The relative phases of the received ultrasound pulse at the multiple receivers yield the listener's orientation to an accuracy of a few degrees.[2] However, the extended listeners report orientation only with respect to a single beacon and we haven't yet fabricated a large number of them. For proof of concept, we developed a prototype orientation listener from two position listeners separated by a fixed baseline (see Figure 2). Its reported position and orientation are sufficiently accurate for text and coarse geometric overlays when the software flashlight is near the surface of interest. However, for finer geometric detail or larger standoff distances, we need position and orientation accuracy on the order of 1 centimeter and 1 degree, respectively. We have improved position listeners in fabrication. Also, as of March 2003, we began the design process for an improved orientation listener.

Extended deployment and self-calibration

We estimate that thousands of beacons (several per room) will be required to instrument a moderately sized building. Each beacon must be programmed with, or discover, its own location so that it can broadcast its coordinates. Currently, we deploy and configure beacons manually, but we are developing a self-configuration algorithm so that naively deployed beacons can range to and communicate with each other to negotiate globally consistent position assignments.[7]

Conclusions

This effort involves a number of interesting research and engineering challenges at a variety of levels: devices and device integration, geometric and database algorithms, and applications and user-interface techniques. We are actively pursuing efforts in each of these areas and deploying experimental infrastructure and applications in our existing work environment. Finally, we plan a large-scale deployment in our new building, which we will occupy in January 2004. ■

Acknowledgments

Jason Bell and Patrick Nichols developed the active signage protoptype. We gratefully acknowledge research support from NTT Corporation, from the National Science Foundation (grant ANI-0205445), and from the Office of Naval Research (grant N00014-02-1-0930).

References

1. N.B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," *Proc. 6th ACM Mobicom Conf.*, ACM Press, 2000, pp. 32-43.
2. N. Priyantha et al., "The Cricket Compass for Context-Aware Mobile Applications," *Proc. 7th ACM Mobicom Conf.*, ACM Press, 2001, pp. 1-14.
3. T. Funkhouser et al., "The UCB System for Interactive Visualization of Large Architectural Models," *Presence*, vol. 5, no. 1, Winter 1996, pp. 13-44.
4. Salil Pradhan et al., *Websign: Hyperlinks from a Physical Location to the Web*, tech. report HPL-2001-140, Internet and Mobile Systems Laboratory, HP Laboratories, June 2001.
5. S. Feiner et al., "Windows on the World: 2D Windows for 3D Augmented Reality, *ACM Symp. User Interface Software and Technology*, ACM Press, 1993, pp. 145-155.
6. R. Lewis, *Generating Three-Dimensional Building Models from Two-Dimensional Architectural Plans*, master's thesis, tech. report CSD-96-902, Univ. of Calif., Berkeley, 1996.
7. N.B. Priyantha et al., "Anchor-Free Distributed Localization in Sensor Networks," *MIT LCS* (Laboratory for Computer Science), no. 892, Apr. 2003.

Readers may contact Seth Teller at the MIT Lab for Computer Science, 200 Technology Square, Cambridge, MA 02139; teller@lcs.mit.edu.

Contact the department editors by email at rosenblu@ait.nrl.navy.mil or michael_macedonia@stricom.army.mil.