Mike Hazas, James Scott and John Krumm (eds.)

2003 WORKSHOP ON
location-aware computing
PART OF UBICOMP 2003

Proceedings of the
2003 Workshop on Location-Aware Computing

Held in conjunction with UbiComp 2003
October 12 2003, Seattle, Washington, USA

http://www.ubicomp.org/ubicomp2003/workshops/locationaware/

Preface

Background

Determining the location of people and objects has been the focus of much research in ubiquitous computing. Many location sensing technologies have been devised, resulting in systems which perform sensing using diverse physical media, such as infrared light, ultrasound, electromagnetic signals, ground reaction force, physical/electrical contact, and visible light. Naturally, these systems have an equally diverse set of properties; each implementation has its own level of accuracy, update rate, infrastructure cost, deployment difficulty, robustness, and capacity for privacy guarantees.

Location-aware applications are numerous. Examples include portable memory aids, conference assistants, environmental resource discovery and control, support systems for the elderly, tour guides, augmented reality, mobile desktop control, 3D mice, and virtual buttons. Each demands different levels of service from the supporting systems, for example in terms of location accuracy and update rate.

There has also been a recent focus on location-aware "platforms," which link data-gathering systems and the data-consuming applications in a flexible manner. Such work includes location representation, sensor fusion to combine location data from many sources, and software frameworks supporting the distributed nature of location-aware computing. Such abstractions are essential for the interoperability, usability and development of location-aware systems and applications.

Finally, location-aware computing includes many issues related to the user experience, such as privacy preservation and its associated legal and ethical implications, the questions of usability and user acceptance, and the need for security in the determination and transfer of personal data.

Workshop Details

The 2003 Workshop on Location-Aware Computing, held as part of UbiComp 2003 in Seattle, aimed to bring together researchers from all aspects of location-aware computing, and collectively examine the state of the field and identify areas for future research. The workshop was organised to be highly interactive, with each participant providing a position statement, and with much of the day organised around inclusive activities such as panel discussions and breakout groups.

The result was a series of very interesting discussions, with all twenty-three participants contributing information and opinions from their various viewpoints. It was intended that the participants would take home a broad and current view of the field, and that the opportunity to make important contacts in the field might result in future work which is complementary or even collaborative in nature.

This proceedings is mainly comprised of expanded, three-page position statements from selected participants, representing influential opinions and work in the field. Also included are some results from "straw polls" conducted at the workshop.

Acknowledgements

The organisers would like to thank the following people:
Michael Beigl and Christian Decker, the UbiComp 2003 workshop co-chairs, for enabling the workshops, which received very positive feedback from the conference attendees.
Joe McCarthy, the UbiComp 2003 conference chair, for his tireless work in organising the conference.
Sukeshi Grandhi, our student volunteer, who also participated in one of the breakout groups.
All the participants, for taking the time to produce position statements, and for the animated and interesting discussions that ensued.

October 2003

Mike Hazas, Lancaster University
James Scott, Intel Research
John Krumm, Microsoft Research

Poll Results

As an "ice-breaker" activity at the start of the workshop, some quick polls of the twenty-three participants were conducted. Disclaimer: these results are meaningless; the questions are leading, the surveyed people unrepresentative and statistically insignificant, and the counting and recording dodgy at best. But they were fun.

I have used a location-aware computing system habitually: 17%

I have sold or deployed a location-aware system: 23%

I see location-aware computing as (pick all that apply):

- Widely used in 5 years: 83%
- Being as groundbreaking as the Internet: 4%
- As an add-on to enhance existing apps: 78%
- As a red herring: 0%
- Ending up as a utility (regarded like the water or electricity supply): 61%
- Improving my quality of life: 65%

The biggest blocking factor against the wide-spread deployment of location-awware computing is (pick one):

- Useful applications: 35%
- Privacy: 26%
- Return on investment: 17%
- Indoor location technologies: 4%
- Reliability: 4%
- Proliferation: 0%

I would be happy if a third party who I sometimes do business with (e.g. a retailer) knew my location on a Saturday afternoon's shopping trip to an accuracy of (pick all that apply):

- None at all: 96%
- Existence but no accuracy: 43%
- Country: 35%
- City: 35%
- Street: 17%
- 10m: 13%
- 1m: 4%
- 1cm: 4%

I would be happy to have custom location-based ads around me (pick all that apply):

- On screens and audio: 17%
- On screen: 22%
- Localised audio: 4%
- SMS: 0%
- Emailed to me: 0%
- Heavily regulated by law: 61%
- Pushed to me, but under my control: 52%

Social acceptance of location-awware systems is going to be (pick one):

- Very hard: 4%
- Hard: 43%
- Doable: 35%
- Not an issue: 13%

Table of Contents

Bootstrapping the Location-enhanced World Wide Web

Bill N. Schilit[1], Anthony LaMarca[1], David McDonald[4], Jason Tabert[3], Eithon Cadag[3], Gaetano Borriello[1,2], William G. Griswold[3]

[1]Intel Research Seattle, 1100 NE 45th Street, Seattle, WA 98105
[2]Dept. of Computer Science and Engineering, University of Washington, Seattle, WA 98195
[3]Dept. of Computer Science and Engineering, University of California at San Diego, La Jolla, CA 92093
[4]Information School, University of Washington, Seattle, WA 98195

ABSTRACT

Our challenge to the research community is to make location-enhanced web services valuable and readily accessible to a very large number of people in daily, real world, situations. We envisage a global scale, multi-organization and interdisciplinary initiative, *Place Lab*, that will bootstrap the broad adoption of the location-enhanced Web. Our research collective is developing an open software base (providing low-cost private positioning technology) and fostering the formation of user and developer communities. Through individual Place Labs initially seeded on the campuses of universities, colleges, and research organizations this initiative will be a vehicle for research, instruction, collaboration and application sharing. This paper describes some of our first steps towards meeting this challenge.

Keywords

Location-aware; context-aware; ubiquitous; positioning systems; WiFi; GPS; web services; wireless hotspots; wardriving.

INTRODUCTION

Location-aware computing has been "in the lab" for the last decade where applications, frameworks, technology infrastructure and much more has been extensively explored. So far location-based services have not made an impact in the mobile computing world (except perhaps with E911 services). The reason is the ubiquitous computing dilemma: how to bootstrap a concept that requires both infrastructure investment and the "killer" (or at least a valuable) application. Without the application people won't invest in infrastructure and without the infrastructure the open-market for iterating towards valuable applications and their business models doesn't exist.

Place Lab is a community-based effort to break this deadlock and make location-aware computing a reality on a mass scale. We see at least three major barriers that must be overcome to realize this vision: low-cost, highly convenient position-sensing technology; making users comfortable with respect to their location privacy; and having existing web content easily customized to geographic locations.

Our current approach exploits the proliferation of wireless networking hotspots that can provide positioning comparable to GPS in urban settings and also function indoors where GPS does not. A downloaded and continually updated distributed contributor database of all the WiFi access points in the world will allow clients to compute their own positions and divulge their location information only when they want to. Services accessed through a web browser will provide users rich information and services associated with their location.

Figure 1: WiFi density in urban centers is such that multiple access points are within range of many locations. Each AP beacons a unique identifier that, along with a mapping database, can be used to lookup a course grain position. In this image each dot is an estimate of the position of a WiFi AP in downtown Seattle mapped in a single "wardrive."

The Place Lab "research challenge" provides an endeavor where the lessons from the field of location aware computing can be applied. This knowledge includes the idea of location-enhanced web browsers, proposed in 1995 [2], and the extensive contributions around WiFi for location [1]. We believe, however, that in order to take location from the laboratory to the real world there remain significant research challenges. In this paper we first present a scenario, describe three challenges, and conclude with a preview of a Place Lab demonstration occurring at Ubicomp 2003.

USAGE SCENARIO

A Place Lab user subscribes to databases, potentially from multiple providers, that the client WiFi Positioning algorithms use to convert an access point BSSID (plus signal strengths) into a geographic position. We expect these databases would be updated once a week or so and might cover large geographic regions such as North America, Europe or Asia. Over time we see this collection of WiFi Positioning databases growing to include every access point in the world (later we describe some ideas on how to bootstrap and maintain the databases). Given such a collection of databases, whenever the client receives BSSID beacons they are able to calculate position without additional network communication. This client-based calculation of position-without-communication is a fundamental principal of the privacy mechanism proposed for Place Lab.

On visiting a location-enhanced web service, the user is able to trade privacy of their location for utility of the web service. We imagine a Place Lab component, the Place Bar, which integrates WiFi Positioning into the user's web browser and allows users to flexibly send location information at various fidelities to enhanced sites. For example, the user might choose to reveal only one of these about their location: country; state (prefecture, canton, province, etc.); city; neighborhood; postal code; street; street address; and longitude/latitude.

HARD PROBLEMS

We have identified at least three hard problems that stand in the way of realizing wide-scale deployment of location:

1. How to bootstrap and manage a worldwide hotspot database for positioning?
2. What is the trust model at the client, what is being revealed, and how can we avoid the "big brother" hot button?
3. How to associate any page on the web with a place in the real world where it might be useful? How can multiple pages appropriate for a location be organized for easy browsing?

These problems, and probably many more, must be addressed by the research community as this challenge moves forward. In the following sections we describe potential solution directions.

How to Bootstrap a Global WiFi Positioning Database?

The first technique to bootstrap a WiFi Positioning database is to generate war-driving data for a region, such as the UCSD campus and town of La Jolla. The idea is to create a rough, incomplete map of the hotspots in an area. With this database, notebook computer and PDA users without GPS can start contributing more information into the database. For example, assume a user goes to a Starbuck's and receives beacons from three APs but only two are in the database. The third AP can then be added to the database with some high confidence that it is near the location of the other two APs. Data can also be added when an unknown AP is detected temporarily between two known APs. This collection of techniques for refining the details of the WiFi Positioning database as a side effect of people using their mobile computers is the second, geographic statistical technique.

Clearly, the data being collected by the geographic statistical technique would be much more useful if it was sent back into the infrastructure and then redistributed to all users as part of the WiFi Positioning database. The third technique is to employ a distributed contributor update mechanism for the WiFi Positioning database similar to the one made famous by the CDDB service:

The WiFi Positioning database could aggregate and statistically process AP sightings, and even use the distributed contributor model to improve the precision of the data over time. In some situations, users might be presented with the current location information that is being sent off to the location-enhanced web service. If users notice an error in the location, or the database just holds the city and not the street, the user could enter the corrected or more precise location information that would eventually be added to the database. Of course, users should not be able to corrupt the database. Statistical methods coupled with authoritative sources of hotspot location can be used to ensure high-quality.

What is the Trust Model & What is Being Revealed?

Whenever a location system is developed we can expect to hear shouts of "big brother!" Some of the news headlines that came out of the Active Badge location systems include: "big brother pinned to your chest," "Orwellian dream come true, a badge that pinpoints you," "badges monitor staff."

The privacy problem is due in part to the choices we present people: either opt-in or opt-out with no levels in between. When opting-in the systems we design generally send location to a central server, that we expect users to trust. Most users do not trust centralized location tracking servers run by the government, large corporations, or even your University's IT staff. As an example you can look at the debate over E-911 in congress.

For Place Lab the questions "when I'm using this what am I revealing?" and "when I'm *not* using this what am I revealing?" are make-or-break questions for adoption. Our approach is two fold: (1) client-only position calculation; and (2) multi-fidelity location revelation.

Client-only position calculation is the antithesis of the "big brother" location server: all computation of a device's location occurs at the trusted client. GPS is a good example of this model. In the case of Place Lab, the inputs to the computation are AP beacons received at the client and a cached copy of a database that allows mapping the WiFi beacons (possibly with signal strength data) to locations. At this basic level of WiFi Positioning, if a client does not use the APs for communication, then a totally private positioning system is possible.

Generally, we think people would want to interact with location-enhanced web services interactively online. However, it is worth noting that an offline "occasionally

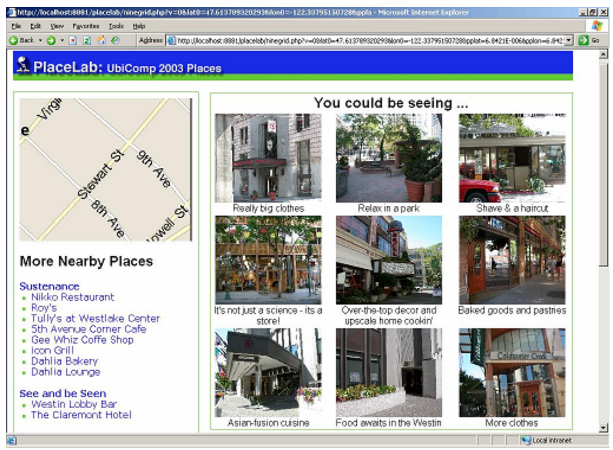

Figure 2: The main page of the Place-Enhanced Conference shows interesting "sights" from around the conference. The content (images, factoids, opinions, and links) is geo-coded and placed in an install package. The entire web site runs without network connectivity and uses beacons from the last seen WiFi hotspot to approximate location

connected computing" (OCC) model (e.g., as supported by .NET) is also reasonable. For example, if the Zagat restaurant guide was an OCC location-enhanced site, you could get information about nearby restaurants without actually revealing location data to the Zagat server.

How to Associate Web Pages to Places?

Another of the challenges in the place-enhanced Web is content discovery: "how do you find information associated with a location?" The most obvious approach is to ask content-providers to annotate their pages with location information. However, how is this "geocoding" structured? Are pages to be tied to specific coordinates? How big is the region around a coordinate for which the page is still relevant? To further complicate matters these regions are unlikely to be simple rectangles and will undoubtedly overlap with each other.

There are two main approaches to dealing with this problem: asking content providers (and third parties) to code their pages with location information; or deriving the locations associated with a page through observation of users' browsing habits. In the first case, we may not get many associations at all since we have put an extra burden on content providers. In the second case, we have to determine how privacy preserving aggregation techniques can be used to collaboratively associate pages with locations. An important issue is where to store and compute these associations.

Even when we have these associations in place, we must still tackle the problem of how to present this information to the user. What happens when a user asks for information associated with a place? What will they see in the browser?

UBICOMP 2003 DEMO

At UbiComp 2003 we are demonstrating a proof-of-concept system to launch our community development effort. We have developed a stand-alone system that conference participants can download and install onto their laptops that will give them a location-aware conference guide for the neighborhood that surrounds the Ubicomp '03

In our demo, users will interact with the conference guide via a standard web browser accessing HTML pages. The map view on each page will place the user on a map of downtown Seattle (or a detailed map of the conference Hotel). The page will also present images of nearby locales. The users can drill down from the basic view to find interesting images, facts and opinions.

One of our concerns designing the conference guide was that the location algorithms we are using provide rough grain information. Although we expect that in time other researchers will apply better algorithms to improve this aspect of Place Lab, we knew that it was possible that position reports could be off by a city block or more! Our first interface had a text-based style and included specific descriptions of computed position. We decided to generalize the interface with imagery, including a map containing of few blocks, in order to avoid confusion if the positioning broke down.

CONCLUSION

A growing multi-organization group of researchers is developing the concepts, open code base, and collaborations that comprise Place Lab. We plan on seeding several partner universities with the necessary elements to develop Place Lab enabled applications and expect a variety of classes from different departments to start the development of relevant and valuable location-aware applications. Our near term objective is to create a way to share applications across all campuses. Our long term goal is to break the cycle that is preventing location-aware usage models from developing on a large scale.

REFERENCES

1. Paramvir Bahl, Venkata N. Padmanabhan: RADAR: An In-Building RF-Based User Location and Tracking System. INFOCOM 2000: 775-784
2. Geoffrey M. Voelker and Brian N. Bershad. Mobisaic: An Information System for a Mobile Wireless Computing Environment. In Proceedings of the Workshop on Mobile Computing Systems and Applications, pp. 185-190, December, 1994

Deploying Commercial Location-Aware Systems

Jay Cadman
Ubisense Inc.
1792 Wynkoop St., #508
Denver, CO 80202
jay.cadman@ubisense.net
http://www.ubisense.net

INTRODUCTION

Ubisense was founded in January 2003 by the team who developed the Bat ultrasonic location sensor and the SPIRIT distributed middleware platform [1][2] at AT&T Laboratories Cambridge. The company is headquartered in Cambridge, England, with offices in Denver, CO.

Ubisense has developed an in-building ultrawideband (UWB) radio based tracking system which can determine the positions of people and objects to an accuracy of a few tens of centimetres, using small tags which are attached to objects and carried by personnel, and a network of receivers which are placed around buildings.

Ubisense has also developed a scalable middleware platform which can manage and distribute large volumes of real-time location information to very many clients, and which simplifies creation of location-aware applications.

UBISENSE'S VIEWPOINT

Ubisense is targeting its sensing and middleware technologies at a number of markets, including healthcare, security, workplace productivity and military training. The end applications for location-aware technology in these markets are varied, and range from ensuring that a baby isn't removed from a care facility by anyone other than its mother, to making sure that a 'simunition' (an explosive charge used in military training) doesn't detonate if a soldier happens to be standing too close. However, we believe that these apparently disparate applications have much in common, and that proponents of location-aware computing must focus on these common elements if the field of location-awareness is to move beyond its research roots.

Some of the most important factors to be addressed when considering commercial deployment of location-aware systems include:

Value: This is clearly the most important factor! For a technology to be commercially successful, it must address a real (not imagined!) need. To us, it seems that location-aware research often focuses too strongly on the mechanics of the underlying technology, with the result that the justification for the work involves spurious applications with questionable value to end-users. We think that more effort needs to be put early on into identifying real-world situations where location-awareness can make a difference – only then can appropriate technology be applied to meet the need.

Examples of potential location-aware applications with which end-users have approached Ubisense include:

- Maximizing productivity of a hospital environment. Certain areas of hospitals, such as operating theatres, represent very high value space which must be utilized as efficiently as possible. This has led to a desire to treat healthcare in hospitals as a highly-pipelined process, somewhat like a production line, to maximise throughput. However, efficient workflow planning requires up-to-date information about the state of the real world, including the locations of staff, patients and equipment. Analysis of historical location data also provides hospital administrators with a method of determining bottlenecks within the healthcare process, and measuring the effectiveness of steps taken to improve efficiency.
- Visitor management in security-conscious sites. Companies in the pharmaceutical and defense industries often require that visitors are escorted by a member of staff at all times, both to ensure visitor safety and protect confidential or classified information. By tracking visitors and their hosts, the visitor management system can alert hosts and visitors when security regulations are being breached (either accidentally or deliberately), and can record a security audit for future analysis.
- Fire fighter training. Location information can be used to enhance the overall training experience for fire fighters. The goal of the training is to provide an experience as close as possible to a real fire, whilst being as safe as possible for the participants. Training exercises can include the use of fixed gas jets – by utilizing accurate real-time location information, we can ensure that jets are not triggered when a trainee is close by and could be injured by them.

- Military training in urban combat scenarios. The changing nature of warfare in the 21st century means that close-quarter combat training in built-up environments is increasingly important. At present, there is no satisfactory way to monitor participants in large-scale, in-building exercises to provide an effective 'after-action review', so there is a requirement to monitor the precise positions of soldiers at all times. Beyond this basic tracking requirement, it is desirable to model the effects of 'area' weapons such as grenades (which have some simulated effect around a particular point in space), and control aspects of the environment in a similar way to the fire training example.

Each of these applications represents a problem that is both pressing and unsolved, and where location-awareness is a valuable system attribute. Clearly, technology choices will have to be made based on the demands of each application and the scenario in which it is to be deployed, but the technology plays a secondary, supporting role to the application.

Robustness: A commercially-successful technology must do more than solve a problem in principle; it must also give customers and end-users confidence that it can solve that problem robustly, in harsh, 24x7 environments. We believe that robustness isn't something that can be 'bolted on' to a proposed location-aware solution – instead, it must be considered at all stages of the design and implementation of location sensing hardware and processing software; this is another reason for understanding the application requirements and environment early on!

In particular, we believe that high location accuracy will be important for many in-building applications – as the accuracy of their tracking system increases, location-aware applications become more robust, increasing end-users' confidence in them. Accuracy specifications for location sensors are often misleading and hard to interpret, and make comparisons between systems difficult. However, we believe that examining the accuracy of a sensor at the 95% confidence level provides a good indication of how a system will work in practice. At this level, the vast majority of sensor readings will fall within the quoted specification, allowing system designers to match an application to a sensor technology with confidence that end-users will perceive the system as being robust and trustworthy.

The role of infrastructure: Our view is that for most commercial deployments of location-aware technology, infrastructure is not a problem, and in many cases it is an advantage.

There is obviously a cost involved with installing infrastructure in an environment, but this cost can be reduced by appropriate design and quantified before the system is installed, and a properly-designed infrastructure will have minimal maintenance requirements once installed. Furthermore, fixed infrastructure lets the vendor give predictions and guarantees about system performance, giving purchasers confidence that applications will be robust.

We also note that in many environments of interest, infrastructure is already accepted. Hospitals, for example, have substantial infrastructure in the form of electrical wiring, air conditioning, pipework, telephone cabling, computer networking and so on. In these target environments, there already exist well-developed processes for managing infrastructure, and we anticipate that similar processes will be used to support installed location sensor technology.

Of course, some environments, such as the home, may be less amenable to the deployment of sensor infrastructure. However, we believe that significant applications for location-aware technology that meet a need in the home have yet to be identified.

Technology for scalable business: Although each location-aware application will have different requirements of its underlying technology, we don't believe that it will be feasible to develop ad-hoc solutions for each and every application – the commercial cost of doing so would be prohibitive, because it would be difficult for companies to reuse existing work and leverage economies of scale. However, having studied applications of location-awareness in a number of markets, we believe that the same UWB-based location technology can satisfy the requirements of a number of different markets involving in-building tracking.

UWB seems well-suited to in-building location-aware applications, because of its non-line-of-sight nature, modest infrastructure requirements and high tracking accuracy. A properly-architected UWB tracking system is low-power (thus low-maintenance), and the fundamental technology is simple and low-cost – this latter factor will also improve as UWB technology becomes more widely deployed (in communications products, for example).

Of course, UWB is also a novel technology, and radio regulatory authorities around the world have been cautious in approving its use. A barrier to sales clearly exists if use of the core technology is prohibited! However, the FCC approved the use of UWB technology in the US in 2002, and spectrum management authorities in other territories are expected to follow suite in the near future.

Standardisation: The process of standardisation has been important in creating and growing global markets for computing and communications systems. Standards like GSM, Ethernet and IEEE802.11 enable interoperability between equipment from different manufacturers, lower costs, and reassure users that their investment in technology will be viable beyond the short term.

Our view is that in the long term, an integrated standard for location and low-rate communications could create huge value. The proposed ZigBee/IEEE802.15.4 standard [3] represents one current effort towards the goal of ubiquitous, low-power communications, and location-awareness is already being discussed in the context of that proposed standard. However, we also note that the market for location-awareness is in its infancy – location-aware application requirements are still not well understood, and these requirements will directly impact the device characteristics that are normally addressed by the standards. We think that standardisation activities for location-aware technology must be rooted in the market, and care must be taken to avoid making early decisions that will impede market adoption.

Business model: The standards and APIs developed in the computing and communications industries over the years have allowed different companies to focus on different aspects of those industries – for example, hardware, middleware, applications, services, etc.

We might imagine that in the long-term, a similar picture will be built up for location-aware technologies and applications. The relative immaturity of the location-awareness market suggests to us, though, that full system sales (including sensor hardware, data management middleware, and, perhaps, application software) will predominate for some time, until the understanding and language required to define appropriate interfaces at each level of the system has been developed.

In fact, it may be that even when suitable interfaces have been identified, the limitations of sensor technology will make it difficult for middleware to completely abstract away the properties of supporting sensor systems, leading to a more complex relationship between players at different levels of the system in the location-awareness space.

CONCLUSIONS

We believe that location-awareness can solve real problems for which there is no existing, effective solution, and that there is increasing understanding in the commercial arena that location-awareness is a viable approach. We also believe that successful commercial deployment of location-aware systems will require proponents to take into account a wide range of considerations which have been little addressed by research work to date.

BIOGRAPHY

Jay Cadman has over ten years experience selling and marketing high-tech products around the world. He was one of the founders of the North American operations of Smallworld (a leading Geographical Information Systems company), helping it grow to over 200 employees, eventually running global marketing. He then spent two years running commercial and risk management for GE Network Solutions and was appointed marketing lead for Automation and Network Services, an approximately $300M GE business. Jay has experience in both direct and channel sales, creating and executing product marketing strategy, global marketing campaigns, mergers and acquisitions and developing technology pricing strategies, and now manages North American sales operations for Ubisense.

REFERENCES

1. Addlesee, M., Curwen, R., Hodges, S., Newman, J., Steggles, P., Ward, A. and Hopper, A. : Implementing a Sentient Computing System. *IEEE Computer Magazine*, Vol. 34, No. 8, August 2001, 50-56.
2. Bat and SPIRIT project pages, AT&T Laboratories Cambridge archive website. Available at http://www.uk.research.att.com/bat and http://www.uk.research.att.com/spirit
3. ZigBee Alliance website. Available at http://www.zigbee.org/

The Cricket Indoor Location System: Experience and Status*

Hari Balakrishnan and Nissanka Bodhi Priyantha
MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), Cambridge, MA
http://nms.lcs.mit.edu/cricket/

October 2003

This note summarizes some lessons we have learned in the Cricket indoor location project at MIT. We discuss the rationale for our original design decisions, and what we have learned from the original prototype (Cricket v1). We discuss how this experience has helped improve the design of the next generation of Cricket, Cricket v2. Like Cricket v1, the Cricket v2 hardware design and software will be released as open-source; v2 units will also be commercially available from Crossbow by early 2004.

1 Background

We started working on the Cricket indoor location system in Fall 1999, motivated by the importance of mobile and context-aware applications in pervasive computing environments, and the poor indoor performance of the Global Positioning System (GPS). Our goal was to build a location-sensing technology that would: (1) scale well to large numbers and high densities of devices that needed location information (2) make it hard to track users, thereby helping with the user privacy problem that plagued previous location systems (such as Xerox PARC's pioneering Active Badge system,[1] and (3) be easy to deploy and manage in large buildings. In addition, we wanted to build a system that costs small tens rather than hundreds of dollars.

Our goals led us to an architecture that was radically different from other indoor location systems like the Active Badge or Active Bat.[2] In Cricket, ceiling or wall-mounted *active beacons* send periodic chirps of RF and ultrasonic signals.*Passive listeners*, connected to *host devices* (handheld, laptop, sensor, etc.), estimate distances to individual beacons using the standard time-difference-of-arrival technique.

This architecture "inverts" the architecture of the Badge and Bat systems, which use passive ceiling-mounted receivers that obtain information from active transmitters attached to devices carried by users.

Qualitatively, the Cricket architecture offers the following advantages: over the Badge/Bat architecture:

+ **Good scalability**: The RF and ultrasonic channel use is independent of the number of devices in any region; when host devices actively transmit, high-density deployments are harder to achieve.

+ **Ease of deployment**: Cricket beacons are easy to deploy; they do not require any infrastructure connecting back to a base station, and can be placed with few constraints inside rooms or corridors.

+ **User privacy**: Cricket's architecture allows a host device to infer its location without the infrastructure or any other entity learning that information. While Cricket by itself cannot guarantee user privacy, it makes centralized tracking of users hard.

These advantages come at some cost:

− **Continuous tracking is hard**: In Cricket, a listener hears only one beacon at a time. Updating the position of a moving device is more complex than in a system that simultaneously obtains multiple distance estimates from the device to known positions.

− **Beacon scheduling**: Cricket requires a distributed beacon scheduling scheme that avoids collisions at the listeners.

− **Energy consumption is potentially higher**: Active beacons tend to consume more energy than passive ceiling-mounted receivers.

We believed that the advantages of our architecture were significant, and that good technical solutions could be found in time to overcome these limitations. The next section discusses our experience in more detail.

1.1 Current status

The start of our effort on Cricket coincided with MIT's Project Oxygen, and Cricket soon became a core technology in that large CSAIL-wide effort. Versions of Cricket have been used by several groups at MIT for applications

*The Cricket project has benefited greatly from contributions from a dedicated team of students and staff. We thank Roshan Baliga, Anit Chakraborty, Dorothy Curtis, Erik Demaine, Michel Goraczko, Albert Ma, Allen Miu, Adam Smith, Ken Steele, Seth Teller, and Kevin Wang for their contributions to various parts of the project. We also thank everyone who built Cricket applications over the past few years and gave us valuable feedback. Cricket v1 was funded in part by NTT Inc. under the NTT-MIT research collaboration, by Acer Inc., Delta Electronics Inc., HP Corp., NTT Inc., Nokia Research Center, and Philips Research under the MIT Project Oxygen partnership, and by IBM Corp. under a university faculty award. Cricket v2 is being funded by NSF under grant number ITR ANI-0205445 and in part by the Oxygen partnership.

[1] R. Want, *et al.*, The Active Badge Location System, ACM Trans. on Info. Systems, 10(1), 91–102, Jan. 1992.

[2] A. Harter, *et al.*, The Anatomy of a Context-Aware Application, Proc. 5th ACM MOBICOM, Seattle, WA, Aug. 1999.

including people location, multi-player physical/virtual games, human and robot navigation, stream migration, and also for several student projects in an undergraduate pervasive computing course (6.964) at MIT offered by Larry Rudolph and Anant Agarwal. We have also offered Cricket courses at MIT and in Asia to our sponsors.

We have given Cricket units to researchers at other institutions, including the University of Washington, Intel Research, NTT Labs, Nokia Research, Delta Electronics, Acer group, Rutgers University, Philips Research, and HP Labs (the last two have also made their own versions).[3]

2 Lessons learned from Cricket v1

2.1 Applications

At the start of the project, we were interested in resource discovery, for which information about the *space* that a user or device is in is important.[4] Providing spatial information requires good boundary detection. But over time, we found that users wanted to build navigation applications, which require knowledge of *position coordinates* and *orientation* in some coordinate system, in addition to spatial information. This led us to implement algorithms to provide these two additional forms of location information.[5]Cricket v1 provides information about space, position, and orientation (using the Cricket compass).[6]

We also found that users wanted to build *tracking* applications, to help a moving device to continually track its position. In MIT's 6.964 pervasive computing course, students modified the software of the popular computer game Doom, so that a player moving in the real world would have his or her moves reflected in the virtual game world. Because we had not implemented any algorithms for fast tracking with a passive listener, those users ended up attaching a beacon to the user device and listeners at known locations. The result was a usable game, and evidence that the Cricket platform was flexible enough to be used in new ways.

Cricket v2 fixes several shortcomings of v1 based on our experience with several applications. First, because Cricket v1 was primarily optimized for good spatial boundary detection, its position accuracy in real deployments had high variance, being accurate to only about 30-40cm. Cricket v2 improves this significantly, being able to obtain distance estimates to within 1cm on average and 3cm most of the time. We achieve this using better ultrasonic signal processing, a better outlier rejection method to filter out bad distance samples, and by implementing finer-grained sample timing at the listener.

The Cricket v2 software will also incorporate a "single constraint at a time" Kalman-filter tracking algorithm that will allow moving devices to be tracked without requiring an active transmitter on the device. Preliminary tests of this algorithm show that its performance at pedestrian speeds compares well with a system that simultaneously obtains multiple distance estimates of the device to known positions and performs a least-square minimization on the resulting linear simultaneous equations.

Our biggest lesson from user applications was that there appears to be no single killer location-aware application in indoor environments. Several potential and active users have told us about their applications across a variety of domains. They fall into the following categories: indoor navigation for robots and people, games and virtual realty, asset tracking, content redirection (music, video, desktop), and embedded sensor network applications. To support this wide range, the underlying platform must provide as much useful information as possible to applications.

2.2 Platform

Cricket v1 provides a simple and general API: the listener passes all distance samples from each beacon to the attached host device, which implements all the processing to obtain the host's location. We found this to be a good design decision, because different applications processed raw distance samples in different ways.

Cricket v2 continues to provide "raw" access to the information collected at the listener to host applications. Additionally, Cricket v2 listeners also perform a significant amount of embedded processing, which allows them to be used with a variety of host devices including sensors that don't perform any Cricket processing.

In Cricket v1, we had erroneously assumed that users would not be interested in changing the firmware running in the beacon and the listener. However, we found that some users wanted to make changes to beacon scheduling, listener filtering, etc. However, the use of a commercial compiler, and software that was tightly coupled to the underlying hardware, made such changes both expensive and time consuming. To overcome this shortcoming, we have rearchitected Cricket v2's embedded software and implemented it in the TinyOS environment.[7]

Cricket v1 listeners interface to a host using a RS232-serial interface. This turned out to be inconvenient for mobile users because it required an unwieldy and obtrusive cable, and was a significant barrier to wider adoption. Cricket v2 provides a more convenient compact flash interface. To enable easy integration with sensor platforms, Cricket v2 also provides a connector to the Berkeley mote / Crossbow Mica platform.

[3] Some other groups have taken units, but we don't know how they are using them.

[4] N. Priyantha, *et al.*, The Cricket Location-Support System, Proc. 6th ACM MOBICOM, Boston, MA, Aug. 2000.

[5] N. Priyantha, *et al.*, The Cricket Compass for Context-Aware Mobile Applications, Proc. 7th ACM MOBICOM, Rome, Italy, July 2001.

[6] We expect to disseminate compass units as an attachable board to Cricket v2 sometime in 2004.

[7] `http://webs.cs.berkeley.edu/tos/`

2.3 Deployment

One can easily put together a Cricket location system by attaching a small number of beacons on the ceiling and by connecting a listener to a host running Cricket application software. The ability to rapidly deploy a working location infrastructure has been a significant user-perceived advantage of Cricket. Demonstrations of the system have been relatively easy to perform both on-site and off-site, students have found it easy to make fast progress in class projects, and users have told us that they have found it painless to get going with the system.

Although Cricket v1 works well at moderate beacon densities, high deployment densities (twelve or more beacons all within range of each other) causes problems. This problem became apparent in situations where users deployed a large number of redundant beacons to protect against batteries running out! This problem is mainly due to the poor noise immunity of the Cricket v1 radio (amplitude modulation and surface-acoustic-wave based receivers). Cricket v2 overcomes the noise problems using a better radio based on frequency modulation and a superheterodyne receiver (CC1000 from Chipcon), and appears to perform well at high densities.

Cricket v1 used separate RF transmit and receive circuits. We found that when batteries on beacons ran down and voltage dropped, the receiver unit failed before the transmit unit, causing the carrier sense mechanism to fail and leading to poor performance. We have corrected this problem in v2.

Cricket v1 beacons are not optimized for good power consumption. We found that the two AA batteries powering a beacon needed to be changed every two to three weeks. To handle deployments where batteries could not be changed this frequently, we successfully powered beacons with solar cells placed near indoor lighting. The Cricket v2 design reduces power consumption by using sub-modules that can be powered down, by using low-power chips, and by implementing better beacon scheduling algorithms. The reduced power consumption of v2 beacons enables them to be effectively powered using small solar cells.

We found that precise distance measurements require sensitive ultrasonic sensors, but such sensors react to ambient ultrasonic noise and high-energy sound pulses. In particular, we found that malfunctioning fluorescent lights, people jangling keys, and loud noises caused by slamming doors cause the listener to record bad distance samples. Fortunately, we have been able to implement good outlier detection algorithms to filter out many such sources of error.

Upon estimating its distance to multiple beacons, a listener associates itself with the space advertised by the closest one, and infers its position coordinates using the known position coordinates of the beacons. Originally, we had envisioned that each beacon would send its space and coordinate information on the RF channel. Over time, we found that it was simpler in many (but not all) cases for a beacon to only send a unique ID, and for the mapping between the ID and the space/coordinate information to be maintained in a central database. In this approach, the listener or host device downloads the database for each building of interest.

Maintaining spatial information for each beacon is straightforward, but coordinate information is considerably harder. The ideal resolution to the problem of configuring beacon coordinates is an auto-configuration method. We, and others in the community, have worked on the following *auto-localization* problem: Given a network of beacons and distances between beacons that are in range of each other, obtain a coordinate assignment for the beacons that satisfies the measured distances.

While there are practical solutions to this problem, we have found that those solutions do not apply well to a Cricket beacon network because obtaining inter-beacon distances in a building-wide or floor-wide deployment of beacons is constrained by two factors. First, ultrasonic signals do not travel across walls, limiting the connectivity across rooms. Second, the directional nature of ultrasonic transmissions limit the inter-beacon connectivity even when they are in the same room.

These two constraints imply that a new solution is required. Our approach, which we are currently refining, is to use *mobile-assisted* localization, where a mobile roving Cricket transceiver is used to "patch together" disconnected portions of the beacon network to obtain a number of distance estimates.

2.4 Manufacturing and dissemination

We have had many more requests for Cricket units than we have been able to handle. We underestimated the difficulty of providing hardware units to interested users, and found that few users were interested in taking the hardware design available on the Web and making their own units. It was hard for us to justify spending money on hardware support, especially because the design changed on a regular basis. As a result, we were unable to satisfy several dozen requests over the past two years. For Cricket v2, we have partnered with Crossbow, who will make units available for purchase and will provide customer support. We will continue to maintain and evolve both the software and hardware designs.

3 Summary

We believe that the active-beacon / passive-listener architecture of Cricket was indeed the right one for the reasons described in this note. We have learned a number of lessons to Cricket v1 and have applied them to Cricket v2.

From Position to Place

Jeffrey Hightower
University of Washington
Dept. of Computer Science & Engineering
Box 352350
Seattle, WA 98195 USA
+1 206 543 1695
jeffro@cs.washington.edu

ABSTRACT
Emerging proactive applications want to reason about "place", not coordinates. Existing systems rely on manually defining places which, while useful, does not scale to ubiquitous deployment. In this paper I define place and challenge the research community with learning and labeling places automatically.

Keywords
sensor fusion, location sensing, activity inference

INTRODUCTION

Several researchers including myself have created systems to fuse live measurements from multiple location technologies. Such systems provide a technology-independent location interface and allow probabilistic queries for objects' geometric positions and relationships. Applications like moving-map navigation and distance-aware buddy lists can easily be built on such infrastructure. However, emerging applications require a more symbolic notion: place. Generically, place is a human-readable labeling of positions. A more rigorous definition is an evolving set of both communal and personal labels for potentially overlapping geometric volumes. An object contained in a volume is reported to be in that place. This programming model maps well to event-driven application programming and is used by most existing location-aware computing frameworks including MSR Easy Living [1], AT&T Sentient Computing [2], and my own Location Stack work [3]. The latter also provides probabilistic confidence values for each labeling. One might argue to skip the geometric intermediary and determine place directly from sensor hardware, the common indoor example being infrared basestations corresponding to rooms, however this is unsatisfying due to the extensive engineering and rigidity of such an approach and has largely been rejected by the community as a general solution.

CHALLENGES

Current approaches require manual definition of places. I must, by hand, delineate and label my neighborhood, property, rooms, furniture, and service areas of my devices, for example, the area in which each of my display screens is visible. Then I can add specific semantics to build applications. Manual definition does not scale. Instead, ubiquitous deployment requires automatically learning significant regions and semantically labeling them as places. I pose these two challenges to the location research community and discuss work in progress.

Predicting Places from Maps and Behavior

The world has static structure such as roads, parks, rooms, and buildings. Maps capture this information well. The world also has dynamic physical constraints observable indirectly through the behavior of people and other entities. For example, people congregate in certain places at certain times, there are travel congestion points, and certain paths and are commonly taken from A to B. The challenge is to augment maps of physical features with the dynamic data to, over time, suggest geometric regions which are good candidates to label as places.

Promising work in this area uses automatic integration of maps, geographic information systems (GIS) data, and usage logs. By looking over time at where I go, how long I spend there, who else is around, and other things I do while there, my system can learn my hubs of activity and methods of transportation. The work in [4] applies this approach to learning typical modes and routes of transportation around a metropolitan area. In indoor environments, [5] shows how to compute the graph-like structure of rooms and hallways from maps to improve the performance of location estimation and enable path prediction. Also, the robotics community has explored the problem of automatically dissecting grid-based maps through robot exploration to learn detailed features and topological layout [6,7]. These efforts must continue with increasing emphasis on wide-area deployment and larger numbers of users.

Labeling Places

Labeling a geometric region assigns it semantics and can help prune and improve place predictions. More importantly, a label directly represents the place's demographic, environmental, historic, personal, or commercial significance and is the desired abstraction for emerging proactive applications. Manually labeling places does not scale so the research challenge is to automate the process.

Simple automated place labeling is already commercialized. Merging web data such as postal addresses with maps enables Nearest-X services where I can map and route myself to “coffee places” offering nearby purchase of espresso beverages – perhaps of a specific brand. Nearest-X services are useful, but they assume the learning input is mostly static and there is only a single type of place. New research seeks to relax both assumptions.

Labeling Places by Inferring Activities

This research seeks to automatically label places by inferring people’s activities in those places. For example, here is the library where people select and read books, there is the kitchen where they cook food and wash dishes, and over there is an office where they use the computer and telephone. The Guide project at Intel Research Seattle is research in this space. Guide infers users' activities from knowledge of recently touched physical objects which are tagged with cheap RFID stickers and sensed by a wrist watch short-range reader. Guide mines the web offline via Google and eHow to build activity models and object-activity correlation probabilities, then can predict activities in real-time using a dynamic Bayes net sampling technique [8].

Guide is interesting because it also consumes place information and may be considered an application of place. The basic `isTouching(object,time)` primitive is easily augmented with `isAt(place,time)` and the web mining extended to populate place-activity correlation probabilities. Considering Guide an application is valuable because it highlights an important requirement for the place programming interface: The interface should be capable of answering both “What place labels are associated with my current coordinates?” and “What is the probability I am currently is in a place P?”

Labeling Places using Grassroots Contributors

Another research path starts by observing that the aggregate of many people periodically labeling their positions is a global place database. For example, if I occasionally provide a name quick or description of my coordinates, over time we can learn significant places by aggregating my labels with those contributed by other people. There are 3 challenges to this approach.

1. **Collaborative Filtering**. Like collaborative filtering for eCommerce web sites, a user-contributor place database must be robust to both incongruous and manipulative contributions. Meeting this challenge requires new research into combining reputation management with machine learning and location sensing. For example, much like Google’s PageRank algorithm foils attempts to artificially inflate a given web page’s search results, a place database must inherently resist similar attempts at manipulation by commercial or other interests.
2. **Data Management**. Making a scalable grassroots data management service for place information can call on the substantial expertise in the systems and databases communities around distributed peer-to-peer data management. It also requires developing the schemas and ontology of place data. For example, knowing that “diner”, “restaurant”, and “International House of Pancakes” are comparable entities allows refinement of the region in question.
3. **Human Interface**. Grassroots place contributor research must pay careful attention to interaction issues such as: What are the incentives for contributing to the communal database? How simple and transparent is the process of adding a place name? How are privacy concerns allayed?

CONCLUSION

Automatically predicting and labeling places is important because manual methods do not scale. Success in this research will enable more ubiquitous deployment of location technology and pave the way for revolutionary new applications which can reason about place instead of coordinates. To inform the work, we must create more applications which use place as a primary input for higher level inference, and, to evaluate scalability, it is critical these applications be widely deployed and have value to real users outside the research lab. The interdisciplinary PlaceLab program may be an ideal venue for this effort to move from position to place. PlaceLab is a grassroots effort to create a privacy-observant, planetary, indoor & outdoor positioning system with low barriers to participation. See `www.placelab.org and` [9] for more information.

ACKNOWLEDGEMENTS

Gratitude goes to Matthai Philipose, Gaetano Borriello, and Anthony LaMarca for vetting ideas and reviewing text.

BIOGRAPHY

Jeffrey Hightower is a doctoral candidate at the University of Washington in Seattle. His research interests are in employing devices, services, sensors, and interfaces so computing can calmly fade into the background of daily life. Specifically, he investigates abstractions and statistical sensor fusion techniques for location sensing. He received an MS in Computer Science & Engineering from the University of Washington in 2000 and is a member of the ACM and the IEEE.

REFERENCES

1. Brumitt, B., Krumm J., Meyers, B., and Shafer, S. Ubiquitous Computing and the Role of Geometry. *IEEE Personal Communications*, October 2000, pp. 41-43.
2. Addlesee, M., Curwen, R., Hodges, S., Newman, J., Steggles, P., Ward A., and Hopper A. Implementing a Sentient Computing System. *Computer* 34(8), pp.50-56, August, 2001.

3. Hightower, J., Fox, D., and Borriello, G. The Location Stack. *IRS-TR 03-008*, Intel Research Seattle, 1100 NE 45th Street, Seattle, WA 98105, June 2003.

4. Patterson, D. J., Liao, L., Fox, D., Kautz, H. Inferring High-Level Behavior from Low-Level Sensors. *Proceedings of the Fifth International Conference on Ubiquitous Computing,* October, 2003

5. Liao, L., Fox, D., Hightower, J., Kautz, H., and Schulz, D. Voronoi Tracking: Location Estimation using Sparse and Noisy Sensor Data. *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, IEEE/RSJ, 2003.

6. Schröter, D., Beetz, M., and Gutmann, J-S. RG Mapping: Learning Compact and Structured 2D Line Maps of Indoor Environments. *IEEE Intternational Workshop on Robot and Human Interactive Communication,* September, 2002.

7. Thrun, S. and Bücken, A. Integrating Grid-Based and Topological Maps for Mobile Robot Navigation. *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI)*, Portland, Oregon. 1996.

8. Philipose, M., Fishkin, K.P., Fox, D., Kautz, H., Patterson, D., and Perkowitz, M. Guide: Towards Understanding Daily Life via Auto-Identification and Statistical Analysis. *The 2nd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications*. Seattle, Washington, USA, October 2003

9. Schilit, W.N., LaMarca, A., Borriello, G., Griswold, W., McDonald, D., Lazowska, E., Balachandran, A., Hong, J., and Iverson, V. Ubiquitous Location-Aware Computing and the "Place Lab" Initiative. *The First ACM International Workshop on Wireless Mobile Applications and Services on WLAN*, September 19, 2003, San Diego, CA, USA.

Location Authorities for Ubiquitous Computing

Steven A. N. Shafer
Microsoft Corp.
One Microsoft Way
Redmond, WA 98052 USA
+1 425 703 1298
stevensh@microsoft.com

ABSTRACT
In this position paper, I define the Location Authority as a repository of spatial element data. Location Authorities are a generic infrastructure component for ubiquitous computing, and every location-based system uses some form of Location Authority. However, if location-based computing is to become "ubiquitous", we need to standardize a number of aspects of Location Authorities.

Keywords
Location-based computing, map, floorplan

WHAT IS A LOCATION AUTHORITY?

A Location Authority is any set of referents for location references, for example:

- A coordinate system such as the WGS84 datum for latitude / longitude
- A map or floorplan in computational form
- A list of street addresses
- The list: "Home", "Office", "Elsewhere"

All location-based computing systems make references to locations. However, it is almost always the case that a given system assumes a specific Location Authority. For example, a "ubiquitous" system to show your location on a map may take location data from GPS, assumed to be lat/lon in WGS84, and show a dot on a map at that spot. But, such a system could not interpret being at "my house" or in "Wean Hall at Carnegie-Mellon University". Because we lack universality in Location Authorities and the means of referring to them, most so-called "ubiquitous computing" systems in fact work only with outdoor GPS data (if WGS84 is used) or in a specific venue with a custom-prepared database (if anything else is used).

To create truly ubiquitous location-based computing, we need to define universality for Location Authorities. This requires that we think about all the variations in how we refer to and use locations, and develop a single framework in which all of these variations can co-exist systematically.

GEOMETRIC AND TOPOLOGICAL LOCATIONS

If a "Location" is taken as an atom, then a Location Authority is a set of these atoms, possibly with annotations. There are two fundamentally different ways for a Location Authority to define locations:

Geometric (Coordinates): The Location Authority defines an n-dimensional space, and the locations are points in this space. Computations are typically Euclidean, and many rich relations can be computed without pre-storing them. However, there are sometimes mismatches in the meaningful precision of the coordinates in various locations. Also, while coordinates are easy for computers to manipulate and for humans to manipulate graphically, they may not convey intrinsic meaning to humans, and listing them in text may be rather tedious.

Topological (Entities): Each Location is simply an atom, possibly with a name, and the Location Authority stores the set of atoms. In many cases, the Location Authority may also store some externally-defined relationships, such as containment or intersection. Containment is the most frequently seen, and probably most powerful, relationship. The Entities may have types according to some Ontology.

Hybrid: The most powerful Location Authorities span these domains, typically by attaching polygons or other point-sets to each topological entity.

FUNCTIONAL VIEW OF LOCATION AUTHORITIES

Location authorities in use today are so varied that it seems unlikely we can define a universal data representation. However, we might be more successful at a functional definition, in which we codify the operations a Location Authority might perform. A given Authority might implement them all, or just some, and might support various operations to various degrees. Here are the generic operations stated in the form of questions to be answered:

1. **What is the set of defined locations?** In a geometric system, these may be implicit, such as "all latitude/longitude coordinates"; in an entity system, these are typically the entities themselves.
2. **Which location(s) are referred to by string "X"?** Lookup by some kind of location ID is the simplest form of this; lookup by description, such as "the office behind the main stairwell" are more complex. An important characteristic of a Location Authority is: "What set of strings (or other data) can it interpret as locations?"
3. **Which location(s) corresponds to a given location defined in a different Location Authority?** Outdoor Location Authorities

typically can convert their internal locations to some universal standard such as WGS84 lat/lon. However, most others are closed systems that do not provide any conversion outside of their own domain.

4. **What is the relationship between X and Y?** (Equivalently: What location(s) have a given relationship to X?) In a geometric system, there may be a large repertoire of relationship operators that can be applied to any set of locations. However, in an entity system, the relationships are explicitly stored, and may be semantic as well as geometric.
5. **What does the vicinity of X look like?** Can the Location Authority draw a map or diagram of the area? This is fundamental for human interaction, and may in fact be the reason the authority exists. Maps are typically generated according to various parameters, and may be pre-calculated or calculated when needed from the raw data.
6. **How do I get from X to Y?** (and at what cost?) The ability to calculate paths, and particularly the shortest path, is fundamental for many applications. Approaches may include Euclidean geometry, following pointers in the entity-graph, or a combination; and the route elements may be calculated on the fly or pre-stored as data.
7. **Where is thing T**? (or things of type T?) (also: What things of type T are at or near location X?; the "Find-Nearest" problem) Some Location Authorities only describe a set of spaces. Others also have databases of "contents", that is, lists of things that are present in the domain, possibly with type information according to Ontology. There may be privacy concerns in revealing this information.
8. **Where am I?** This is actually a special case of #7, but it is important to single it out because it is frequently supported by a dedicated sensor suite or information network. Indeed, many people assume that "location-based computing" means specifically that there is a Location Authority to answer this question, "Where am I?", even if no other question can be answered.

Most Location Authorities cannot answer all of these questions; and the degree of detail or variation for answering each question also varies. We cannot assume that all Location Authorities can answer all questions to the same great depth.

Examples: Location Authorities at Microsoft

There are many Location Authorities at Microsoft, which are interesting because they exhibit very different, yet typical, "profiles" of capabilities:

- Facilities management database with floorplans of all buildings, some site maps, and a database of all rooms. Locations are rooms and other spaces; conversions are only from buildings to street addresses; the only relationships available are containment and room adjacency. Floorplans can be drawn. There is no path planning or storage of "contents".
- Network administration resources include the location of all printers and other network resources, as well as the occupants of offices. This (composite) authority can answer questions about things, but not about spaces themselves.
- MapPoint and other global mapping systems are not generally thought of as location authorities for an enterprise. Yet, they can map building addresses to latitude/longitude, plan paths between buildings along public streets, etc.

ISSUES IN LOCATION AUTHORITIES

There are some additional key issues that define the nature of Location Authorities.

Federation

Any non-global Location Authority typically covers a domain that corresponds to some jurisdictional boundary. For example, the Microsoft facilities database covers the buildings and campuses operated by Microsoft Corp.

However, a person might want to make an inquiry that spans across jurisdictional boundaries, for example "What is the nearest Taco Chef restaurant to Microsoft Building Q?" This might require accessing location authorities for Microsoft, Taco Chef, and also the city (or a global location authority such as MapPoint, MapQuest, etc.).

To do this effectively, some kind of "cross-references" are needed to relate entities or coordinates in one authority to those in another authority. These create a "federation" of cooperating Location Authorities. There are different kinds of cross-references needed for different situations.

One kind of cross-reference is a "detail", for example if a global location authority defines a city, it might create a cross-reference from that city entity in its own database, to the root entity of the location authority for that city. The city has a clear operator – the city government – which makes this possible.

However, suppose the city defines a shopping mall. The city would probably cross-reference the mall authority as a "detail". The mall, in turn, might cross-reference from the root entity in its own database, to the corresponding entity in the city database. This would be a "context" reference, since the city provides context for the mall.

A third relationship is the "peer". If the mall defines entities for shop spaces, Taco Chef might link its own root node to the node for its space in the mall. Clearly the mall is a context for the restaurant; but it may also be a detail, for example if it defines some utilities infrastructure, HVAC data, etc., which is inside the restaurant space.

Outdoors v. Indoors

Outdoor location authorities and indoor location authorities tend to differ in some subtle yet important ways.

Coordinate Systems: Outdoors, there is one primary space to represent. Various coordinate systems may be used, but they typically have well-defined transformations between them. Indoors, however, each floor of a building typically acts as a separate universe – two points on different floors may have the same coordinates on their respective floors, but have an unknown relationship in the real 3D world. One way to solve this problem is to attach coordinate systems to entities such as building floors. This approach also works when buildings are more complex, for example if they have wings or towers, or if two buildings whare a parking lot or structure.

Sensors: Outdoors, GPS is universally used. It is now even being used in telephones to provide emergency location data. Typical precision is better than 5m on the surface of the earth, i.e. 1 part in 8,000,000 or more. On the other hand, indoors, there is no universal sensor. The best wireless sensors might have a precision of 1 cm over a field of coverage about 100m across, for a resolution of 1 part in 10,000 or less – about 3 orders of magnitude less than GPS! More typical resolutions of available sensors are about 10m in 100m, for a resolution of 1 part in 10. Navigation by people is likewise quite different – outdoors, it is common to give directions in terms of distance or lat/lon; whereas indoors it is more common to use visual landmarks such as signs, doorways, or objects.

Jurisdictions: Outdoors, vast amounts of data are publicly available, including geographical features, governmental boundaries, public roads and other transportation, etc. Indoors, even if data is available online, it is most often created, maintained, and controlled by a private enterprise of some kind. Outdoor data is frequently available from a "global location authority"; indoor data requires connection to some kind of private location authority. Since large buildings frequently sublet space to tenants, it may be necessary to span several location authorities. Thus, federation issues are central to indoor location authorities. In addition, outdoors, names of entities are typically matters of public record; whereas indoors, names are assigned by the operator of the space in question, and may not be publicly available.

Information Hiding

Just because a Location Authority is able to compute the answer to a question, doesn't mean that it is willing to do so. For example, an enterprise may keep its building floorplans secret from non-employees. Even if an employee can see the floorplans, there may be some sensitive areas omitted from the published floorplans. Also, some contents might be omitted from some published data. In the general case, a Location Authority may have some mechanism for hiding some of its information based on who is making an inquiry.

One situation in particular has engendered great controversy, and that is the publication of the whereabouts of individuals. If this information is generated centrally, it can be published according to the gatherer's policies; if it is generated by each individual, then each person can choose whether to forward the information to a central authority for publication.

LOCATION- AND APPLICATION-INDEPENDENCE

All of the discussion in this paper is based on the premise that we desire to have location-based applications that can run anywhere. To do this requires that the location data they will access must be available to them anywhere. Since these programs will not be rewritten and recompiled at every place, it follows that they must have a common way of accessing this location data, i.e. a location-independent way of getting answers to the key questions posed earlier. It may be that the answer to a specific question is simply not available in a particular place; but the program must be able to ask the question in a uniform way if it is to achieve location-independence.

In order to accomplish this, location authorities must also be constructed in an application-independent way. To date, this has not generally been the case (except for lat/lon data outdoors). For example, at a conference, there may be a custom-built "guide" application to show the events in various rooms. But, such custom binding of application to location resources is precisely what prevents "ubiquitous computing" from being ubiquitous!

Creating application-independent location authorities is a challenge, indeed it is not clear that it is even possible. On the one hand, we have GIS (Geographic Information Systems), which more-or-less accomplish this for outdoor data. On the other hand, even GIS systems are frequently tailored at numerous levels for a given application, even if their underlying model is standardized.

To achieve application-independent location data, and thus location-independent location-based applications, several conventions will be required:

- An application-independent interface for answering the key questions presented above;
- A location-authority-independent representation of "a location" as a data object;
- A "universal federation" in which all location authorities combine to create a "virtual location authority" to cover the world in one framework.

Industry Resources

OpenGIS, www.opengis.org/

Open Mobile Alliance, www.openmobilealliance.org

Bayesian Techniques for Location Estimation

Dieter Fox, Jeffrey Hightower, Henry Kauz, Lin Liao, and Donald J. Patterson
University of Washington, Department of Computer Science & Engineering, Seattle, WA

Keywords
Bayesian filtering, location estimation, uncertainty

1. INTRODUCTION

Location awareness is important to many pervasive computing applications. A fundamental problem in this context is *location estimation*, which is the estimation of a person's location from a stream of sensor data. Since no location sensor takes perfect measurements, it is crucial to represent uncertainty in sensed location information and combine information from different types of sensors. Bayesian filter techniques provide a powerful tool to help manage measurement uncertainty and perform multi-sensor fusion. Their statistical nature makes Bayes filters applicable to arbitrary sensor types and representations of environments. For example, Bayes filters provide a sound approach to location estimation using GPS data along with street maps or signal strength information along with topological representations of indoor environments. Furthermore, they have been applied with great success to a variety of state estimation problems including speech recognition, target tracking, vision, and robotics. In this article, we briefly survey the basics of Bayes filters and their different implementations. Furthermore, we discuss directions for future research in Bayesian techniques for location estimation.

2. BAYESIAN FILTERING

Bayes filters probabilistically estimate the state of a dynamic system from a sequence of noisy sensor observations. In the most basic form of location estimation, the state of interest is the location of a person or object, and observations are provided by sensors either placed in the environment or carried by the person.

2.1 Belief Update

Bayes filters represent the state at time t by random variables x_t. At each point in time, the uncertainty is represented by a probability distribution over x_t called *belief* $Bel(x_t)$. The key idea of Bayes filters is to sequentially estimate such beliefs over the state space conditioned on the information contained in the sensor data. Let us assume that the sensor data consists of a sequence of time indexed sensor observations $z_{1:t}$. The belief $Bel(x_t)$ is then defined by the posterior density over the random variable x_t conditioned on all sensor data available at time t:

$$Bel(x_t) \;=\; p(x_t \mid z_{1:t}) \qquad (1)$$

Roughly speaking, the belief provides an answer to the question "What is the probability that the person is at location x if the history of sensor measurements is $z_{1:t}$?", for all possible locations x. In general, the complexity of computing such posterior densities grows exponentially over time since the number of sensor measurements increases over time. To make the computation tractable, Bayes filters assume the dynamic system is Markov, *i.e.* all relevant information is contained in the current state variable x_t. The update of the Bayes filter is performed in two steps:

Prediction: At each time update, the state is *predicted* according to the following update rule.

$$Bel^-(x_t) \;\longleftarrow\; \int p(x_t \mid x_{t-1})\; Bel(x_{t-1})\; dx_{t-1} \qquad (2)$$

Here, the term $p(x_t \mid x_{t-1})$ describes the *system dynamics*, *i.e.* how the state of the system changes over time. In location estimation, this conditional probability is the motion model – where the person might be at time t, given that she previously was at location x_{t-1}. The motion model strongly depends on the information available to the estimation process. It can range from predicting the next position using estimates of a person's motion velocity to the prediction of when a person will exit the elevator using an estimate of the person's goal.

Correction: Whenever new sensor information z_t is received, the measurement is used to correct the predicted belief using the observation.

$$Bel(x_t) \;\longleftarrow\; \alpha_t\, p(z_t \mid x_t)\; Bel^-(x_t) \qquad (3)$$

$p(z_t \mid x_t)$, the *perceptual model*, describes the likelihood of making observation z_t given that the person is at location x_t. For location estimation, the perceptual model is usually considered a property of a given sensor technology. It depends on the types and positions of the sensors and captures a sensor's error characteristics. The term α_t in (3) is simply a normalizing constant which ensures that the posterior over the entire state space sums up to one.

$Bel(x_0)$ is initialized with prior knowledge about the location of the person, typically uniformly distributed if no prior knowledge exists. Bayes filters are an abstract concept in that they only provide a probabilistic framework for recursive state estimation. To implement Bayes filters, one has to specify the perceptual model $p(z_t|x_t)$, the dynamics $p(x_t|x_{t-1})$, and the representation of the belief $Bel(x_t)$. The properties of the different implementations of Bayes filters strongly differ in the way they represent probability densities over the state x_t.

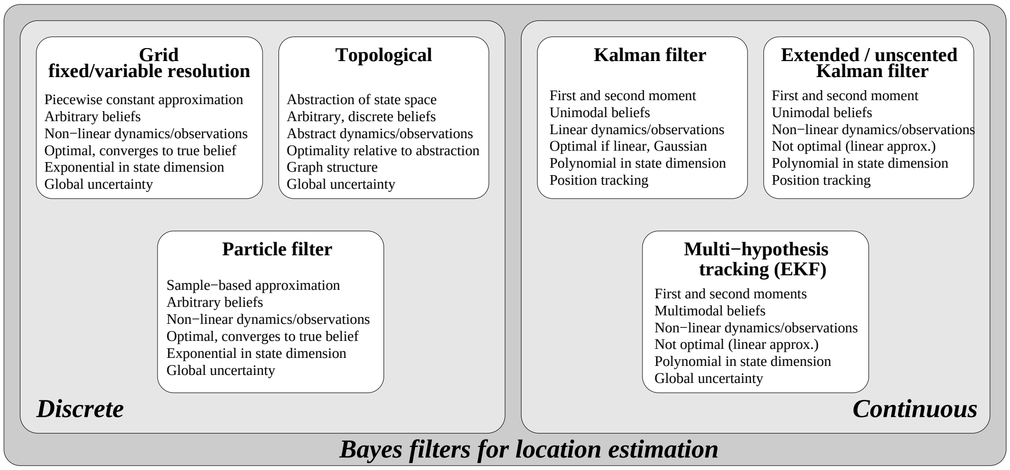

Figure 1: Properties of the most common implementations of Bayes filters for location estimation.

2.2 Belief Representations

This section gives a brief overview of different representations for the beliefs of Bayes filters (see also Figure 1).

Kalman filters are the most widely used variant of Bayes filters [1]. Roughly speaking, these filters approximate beliefs by unimodal Gaussian distributions, represented by their mean and variance. While the mean gives the expected location of the person, the variance represents the uncertainty in the estimate. Even though Kalman filters make strong assumptions about the nature of the sensors and a person's motion, they have been applied with great success to various estimation problems. The main advantage of Kalman filters is their computational efficiency, which comes at the cost of restricted representational power since Kalman filters can only represent unimodal distributions. Hence, Kalman filters are best if the uncertainty in a person's location is not too high. Typical sensors used for Kalman filter based estimation are cameras, laser range-finders, and GPS systems.

Multi-hypothesis tracking (MHT) extends Kalman filters to multi-modal beliefs [1]. MHT represent the belief by *mixtures* of Gaussians where each hypothesis is tracked using a Kalman filter. The weights of the hypotheses are determined by how well they predict the sensor measurements. Due to their ability to represent multi-modal beliefs, MHT approaches are more widely applicable than the Kalman filter.

Grid-based approaches overcome the restrictions imposed on Kalman filters by relying on discrete, piecewise constant representations of the belief. For indoor location estimation, grid-based filters tessellate the environment into small patches, typically of size between 10cm and 1m. Each grid cell contains the belief the person is currently in the cell. A key advantage of these approaches is that they can represent arbitrary distributions over the discrete state space. The disadvantage of grid-based approaches is the computational complexity, which makes them applicable to low-dimensional estimation problems only, such as estimating the position and orientation of a person.

The computational complexity of grid-based methods can be avoided by non-metric representations of an environment. For instance, *graph structures* are well suited to represent the motion of people in buildings [5] or even in cities [8]. Each node in the graph corresponds to a location and the edges describe the connectivity of the environment. The advantage of topological approaches is their efficiency since they represent distributions over small, discrete state spaces. Their disadvantage is the coarseness of the representation which enables only rough information about a person's location. Topological approaches are typically adequate if the sensors in the environment provide only very imprecise location information.

Particle filters represent beliefs by sets of weighted samples distributed according to the belief [3]. Particle filters realize Bayes filter updates according to a sampling procedure, often referred to as sequential importance sampling with resampling. The key advantage of particle filters is their ability to represent arbitrary probability densities, which makes them applicable to problems for which Kalman filters are not well-suited. Compared to grid-based approaches, particle filters are very efficient since they automatically focus their resources (particles) on regions in state space with high probability. However, since the worst-case complexity of these methods grows exponentially in the dimensions of the state space, one has to be careful when applying particle filters to high-dimensional estimation problems. Recently, Rao-Blackwellised particle filters [2], the combination of particle filters with Kalman filters, have been applied successfully to tracking the locations and identities of multiple people [10].

2.3 Parameter Learning

The parameters of the perceptual and motion models can be learned from data using expectation maximization (EM), a popular approach to parameter estimation from incomplete data [9]. The perceptual model $p(z_t \mid x_t)$ is typically independent of the person and can be learned beforehand. The motion model, on the other hand, might be different for each person. Learning the parameters of the motion model allows the system to adapt to a specific person, thereby increasing the accuracy and efficiency of the estimation process. For example, [6] show how to use EM to learn typical motion patterns of a person in indoor environments using a graph-based Bayes filter. [8] use the same technique to learn the navigation patterns of a person through an urban environment.

3. RESEARCH DIRECTIONS

In this section we briefly discuss directions for future research in Bayesian location estimation.

Adaptive Estimation

Most applications of Bayes filters use the same, fixed representation of the state space during the entire estimation process. However, especially in the context of location estimation, this is not appropriate. For example, the location of a person moving through an urban environment can be tracked well using multi-hypothesis tracking along with a GPS sensor and a street-map. However, as soon as the person enters a building, other sensors and representations are needed. Furthermore, even within the same building, different areas might be covered by completely different types of sensors requiring different representations of the belief state. A key question is thus when and how to switch between different representations in a statistically sound way.

High-level Representations

The location of a person provides only very limited information about the person's current activity. Richer representations might include information such as the time of day, the mode of transportation, the destination of the current trip, and the purpose of a specific location. *Dynamic Bayesian networks*, a variant of Bayes filters, provide a sound way of describing and reasoning with such structured, hierarchical information [7]. Some questions remain: What are important locations in a person's life? How can they be described in a general way and learned from sensor data? How can we transfer experience gained from one person to another person? *Relational probabilistic models* [4], which can represent relations between classes of objects, provide a promising framework for addressing these problems.

User Errors

In the context of assisting cognitively impaired people, the detection of when a person seems to be lost is an important aspect of location estimation. *Online model selection* is a technique that can potentially solve this problem. Model selection aims at identifying the model that is best suited to explain the observed data [11]. To apply model selection in the location context, one could generate generic *and* user-specific Bayes models of activities. Both models are able to track a user's activities, but the specific model is tuned towards the typical actions of one particular user. The specific model additionally contains all errors that are typical for the user. The idea is that as long as the user performs her usual activities, the tuned model will be much better in predicting these activities. Surprising actions, *i.e.* potential errors, however, are not well predicted by the specific model, in which case the generic model receives higher probability. For example, if a person exits the bus every morning at the same bus stop, then the specific model predicts this action with very high probability. If the person fails to exit the bus at the usual stop, then the general model predicts it with higher probability, thereby triggering the detection of a potential user error. Obviously, such an approach can provide valuable information to user intervention modules.

4. CONCLUSIONS

We presented Bayes filters as a general framework for location estimation, allowing the integration of sensor information over time. The application of Bayes filters goes well beyond location estimation. The generation of hierarchical models allows the seamless integration of location estimation into user activity estimation. We consider Bayesian techniques to be an extremely promising tool for location aware computing.

REFERENCES

[1] Y. Bar-Shalom and X.-R. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. Yaakov Bar-Shalom, 1995.

[2] A. Doucet, J.F.G. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2000.

[3] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo in Practice*. Springer-Verlag, New York, 2001.

[4] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *IJCAI*, 1999.

[5] J. Krumm, L. Williams, and G. Smith. SmartMoveX on a graph – an inexpensive active badge tracker. In *International Conference on Ubiquitous Computing (UbiComp)*, 2002.

[6] L. Liao, D. Fox, J. Hightower, H. Kautz, and D. Schulz. Voronoi tracking: Location estimation using sparse and noisy sensor data. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.

[7] K. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, UC Berkeley, Computer Science Division, 2002.

[8] D. Patterson, L. Liao, D. Fox, and H. Kautz. Inferring high-level behavior from low-level sensors. In *International Conference on Ubiquitous Computing (UbiComp)*, 2003.

[9] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*. IEEE, 1989. IEEE Log Number 8825949.

[10] D. Schulz, D. Fox, and J. Hightower. People tracking with anonymous and id-sensors using Rao-Blackwellised particle filters. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.

[11] M. West and P.J. Harrison. *Bayesian Forecasting and Dynamic Models*. Springer-Verlag, New York, 2nd edition, 1997.

SpaceSemantics: An Architecture for Modeling Environments

Richard James Glassey and Robert Ian Ferguson
The Pervasive and Global Computing Group
Department of Computer and Information Science
University of Strathclyde
Glasgow, Scotland
+44 141 548 4300
{rjg, if}@cis.strath.ac.uk

ABSTRACT
The notion of modeling location is fundamental to location awareness in ubiquitous computing environments. The investigation of models and the integration with the myriad of location sensing technologies makes for a challenging discipline. Despite notable development of location models, we believe that many challenges remain unresolved. Complexity and scalability, diverse environments coupled with various sensors and managing the privacy and security of sensitive information are open issues. In this paper we discuss our previous experience combining location sensing with mobile agents and how the lessons learnt have lead to the conception of *SpaceSemantics*, an open architecture for modeling environments.

Keywords
Location modeling, semantic model, typed object/relationship graph, distributed persistent object model.

INTRODUCTION
The discipline of ubiquitous computing is well established. A sub-domain of this study is location-aware computing in which a device will behave differently depending on its location and perhaps pro-actively supplying information to a user appropriate for a particular location or situation. Whilst location-sensing technologies can answer the standard question, "*Where am I*?" to varying degrees of accuracy, there exists a need to frame '*where*' in the context of a modeled environment in order to move beyond simple inferences of position to a better understanding of what '*where*' relates to semantically.

Previous research has produced various types of location models that provide an abstraction between users/devices and the raw data provided by various location sensing technologies. The earliest investigations into ubiquitous computing environments [6] introduced abstract models of location for the environment. Since then the models of location have developed in terms of representation and complexity. Previous work [4,5] has identified three types of location model:

- *Geometric* - allows points, areas and volumes to be modeled; however a point in geometric space has no relationship to what it points to. The resolution of this model is as fine as the units of measurement used.
- *Symbolic* - describes location and space in terms of names and abstractions. Unlike the previous model type, humans and computational devices can understand this model, however they lack the precision of geometric models.
- *Hybrid* - represents a logical step forward in combining the advantages of the previous model types in order to overcome their respective disadvantages. As a consequence the hybrid model is more complex, requiring greater amounts of data.

Despite these advances challenges still exist within the field of location modeling. This paper aims to highlight some of the current areas of research interest in location models for location aware computing and ubiquitous computing, discuss our early experience with location sensing technologies and present our vision of a novel approach for modeling environments.

CHALLENGES
There are several complex aspects inherent in ubiquitous computing environments that make location modeling difficult. The earlier challenges of location modeling, representing position accurately and ensuring this representation was understandable to both humans and computers, have been superseded by greater challenges. The following list summarizes a selection of pertinent challenges facing future location models:

- *Managing complexity and scalability*: As models increase in complexity the management and integrity of the information becomes a critical design issue. In addition the design of a model should not only take into account the potentially large number of entities in a single environment, but also factor for multiple environments linked together.
- *Transient environments and aggregation of sensor data*: Designing a model that successfully bridges the

difference between administrative, social and home environments is challenging. Focusing the design on a single environment may obscure difficulties when applying it to another environment type. Many environments will support one or more differing location sensing technologies [3]. Aggregation of this multiple sensor data would rely on an abstract location model not directly connected to or dependent upon a particular location sensing technology.

- *Inference beyond position*: Whilst determination of position remains important there is potential for greater contextual inferences to be made from a model in terms representing conceptual, logical and physical connectivity.
- *Privacy and security*: Although previously acknowledged there are still many issues surrounding the access control and management of potentially sensitive location information.
- *Ontology for location*: The decision of how to describe space is not a trivial matter, however a common means to represent location across various different models may be useful.
- *Open and extensible model*: The task of providing location information for the model should not rely solely on a single source. The ability for other providers to supply additional information is desirable. In order for a model to evolve along with changes in the environment it and the sensing technologies employed it must be easily extensible and adaptive.

This list is not exhaustive, but it does reflect the effort still required in designing models. The following section discuses our previous experience in investigating location aware computing through combining location sensing with mobile agents.

PREVIOUS EXPERIENCE

The idea behind Boarders is very simple: Imagine a handheld device (such as the IPAQ) with a radio networking capability. When the device is carried into range of a radio base station, a mobile agent jumps onto (boards) the device and carries out whatever computation is deemed necessary. When the device goes out of range (and by implication the user leaves the location), the boarder agent is deleted from the device (after a suitable delay). In this manner software and data can be associated with a location. This technique is eminently suitable for the limited capability devices that are currently available for mobile computing: Software is installed on demand only where and when needed.

The real surprise with Boarders was how easy it was to implement on top of the mobile agent framework, AgentSpace2 [2]. It required the creation of an agent (which became known as a range-server agent) that listened for devices entering the range of the base-station and then created instances of appropriate application agents and launched them onto the newly discovered devices. This experience convinced the author that mobile agents are an interesting approach to the problems of supporting location aware computing.

Several applications have been built using the boarders framework. One, Whiteboarders, is a communal whiteboarding application for use with students in lecture theatres. As students with suitably equipped and set-up devices enter a lecture theatre, they come within range of a base station that causes a whiteboard agent to be installed. The agents communicate with a coordinating agent allowing any drawings or marks made on the mobile devices to be displayed upon the projected display in front of the class. The ability to freely share impromptu diagrams and emphasizing marks is (usually) constructive.

A second example system was based on the idea of being able to associate information with both a location and a person (or group). In it's simplest form it allows users to leave virtual post-it notes in given locations for particular people or groups to pick up ("Since you're passing the store dear, can you pick up some potatoes please?") but the same design framework is now being evaluated as the basis for a larger scale (city-wide) tourist guide.

There are problems with the approach and reasons why it might not scale well: As yet there has been no serious consideration of security aspects, thus the system has prior knowledge (via configuration files) of devices that are allowed to participate. Introduction of a new device requires manual intervention. In the restricted context of a single department it has, however, proved successful.

The limitations of the Boarders approach soon became apparent - equating a radio base station with a room is a course-grained approach and ignores all manner of issues of signal propagation. The obvious solution, allowing a singe base station/range server combination to control whatever physical rooms/regions are within its range, leads to the need for a second, finer grained stage of location service to be employed once a device enters a range. Several technologies could provide this level of service but for experimental purposes an audio-based solution was used. Basic operation remains the same - once a device's location is determined via the two-stage process, boarder agents appropriate to the location are launched onto the device.

Providing such a service requires that a range server maintain a description of the physical environment it is responsible for and have the ability to cross-reference that with any location services. The desirability of a general technique for describing ranges, locations, services and information rapidly became apparent and has lead to a project to provide such a technique for the development of a distributed (potentially global-scale) world-model. The project utilizes several other features of the mobile-agent paradigm - globally unique identifiers, persistence, interagent communication - to facilitate the construction of a globally available distributed persistent object database and various languages including X3D and XML. Admittedly in this scenario, mobility is relegated to a minor role of supporting load balancing.

In the following section we propose an open architecture for modeling location in ubiquitous environments that will progress beyond previous models, address the challenges highlighted earlier and build upon the experience gained.

VISION

We envisage a novel architecture, *SpaceSemantics*, which provides descriptions of arbitrary aspects (conceptual, logical and physical) of the real world and the relationships between entities existing therein, implemented across an open, distributed computing environment. *SpaceSemantics* will provide a universal model on top of which applications and devices can query over location knowledge and data from various location sensors can be aggregated under a common model.

Our proposed solution takes the form of a graph consisting of typed nodes connected by typed relationships. The nodes are implemented as distributed persistent objects and the relationships as references between the objects.

Unlike previous models that largely relied upon hierarchical arrangements we believe that a graph permits intuitive traversal whilst still allowing hierarchies to be modeled. The relationships between nodes will form various typed networks ranging from simple containment and connectedness to ownership, all of which can be linked and traversed.

SpaceSemantics will possess properties of openness and extensibility such that the model is not maintained nor provided by one authoritative source. If for example a department provides a basic floor map with little detail other than room nodes with names, then an occupant can overlay their own model of their room to create a more detailed model. Inspiration for an open model is drawn from the same distributed and participatory approach, to constructing a world-model, as demonstrated by the World Wide Web and Peer-to-Peer networks.

The overview of features that this architecture will possess place great demands on the technology employed to implement it. The network of nodes is not static; instead it is expected to grow and evolve (as information becomes available or is refined) and requires a flexible solution. A distributed network database of persistent objects is adaptive and flexible to change making it a suitable approach. The objects, descriptions and relationships are collapsible to XML, X3D and Xlink in order to maintain parts of the model as they become unavailable through disconnection from the network.

SUMMARY AND FUTURE WORK

The modeling of location information is of great importance to location-aware computing. The results of previous research have increased our understanding of the issues inherent in the field. Challenges however are still present and require effort to overcome. This paper has highlighted a few of the interesting challenges facing location modeling and we have introduced our approach with *SpaceSemantics*.

Work on *SpaceSemantics* is currently at the design/first prototype stage. The desire is to integrate *SpaceSemantics* with the Strathclyde Context Infrastructure [1], a project investigating the composition of contextual components to support context-aware applications.

REFERENCES

1. R. Glassey, G. Stevenson, M. Richmond, P. Nixon, S. Terzis, F. Weng and R.I. Ferguson. Towards a Middleware for Generalized Context Management. *In Proc. 1st International Workshop for Middleware for Pervasive and Ad-hoc Computing*, pp. 45-52, Middleware 2003 companion.
2. D. Grey, P. Dunne and R.I. Ferguson. A Mobile Agent Architecture for Searching the WWW. *In Proc. Workshop on Agents in Industry, 4th International Conference of Autonomous Agents*, Barcelona, 2000.
3. J. Hightower and G. Borriello. Location Systems for Ubiquitous Computing. *IEEE Computer*, 34(8), pp. 57-66, 2001.
4. C. Jiang and P. Steenkiste. A Hybrid Location Model with a Computable Location Identifier for Ubiquitous Computing. *Lecture Notes in Computer Science*, v. 2498, pp. 246-263, 2002.
5. U. Leonhardt. Supporting Location-Awareness in Open Distributed Systems. PhD Thesis, University of London, UK, 1998.
6. B. Schilit and M. Theimer. Disseminating Active Map Information to Mobile Hosts. *IEEE Network*, 8(5), pp. 22-32, 1994.

Relative Positioning

Albert Krohn
Telecooperation Office
Institute for Telematics
Department of Informatics
University of Karlsruhe
Vincenz-Prießnitz-Str. 1
76131 Karlsruhe, GERMANY
+49 721 690 229
krohn@teco.edu

ABSTRACT

Positioning of mobile devices is a core requirement in many mobile computing applications. Typically, such applications use external infrastructure installed in the environment to obtain position information. Several different technologies have been investigated such as ultra sound, magnetic fields or inertial systems. In contrast to most of the established location systems, a promising approach is to avoid the need of any infrastructure and to gather location information in a distributed and peer-to-peer manner.

LOCATION SYSTEMS

In the past few years, a reasonable number of location systems for small objects have been published. Different technologies have been investigated and brought to a concrete implementation, some even to a commercial product. Initial research projects were e.g. the Active Badge System [5], a system using ultrasound for location estimation and SpotOn [3] using RF field strength measurements. These systems use mainly the theory of triangulation like the GPS system. Measurements are made between mobile objects on one side and a well positioned and known infrastructure on the other side. The measurements give the distances between the concerning objects and data is collected afterwards in order to combine the measurements to a complete location estimation of the objects. This estimation can be done in the object itself, like in GPS, or in the infrastructure, like in the Badge System. In some cases, the measurements include more than just the distance but also give orientation in terms of axis-angles between objects and the infrastructure.

These system -and other similar approaches- have one important thing in common: They are all in the need of an infrastructure in order to fulfill the task of localization of objects. The infrastructure in most cases consists of beacons sending out position information or complex receiver circuitry. The used physical measurands are in all cases either the time of flight or the received intensity of a physical signal. To use those physical values it is either necessary to establish a synchronization (for a system using time of flight) or having calibration concepts for electronic components (in a system using received signal strength for localization).

The installation of the infrastructure is in many cases a long and elaborate task that included precise measurements and various interconnection between infrastructure components. The infrastructure normally works as a whole and not with independent components. But the most important thing to say about those location systems in this context is that the functionality is *dependent* on the infrastructure. That means, the localization of objects is only possible when and where the enabling infrastructure is available. Applications that rely on positioning of their objects are as well limited to spots where the supporting infrastructure is present.

Further on, infrastructure based location systems often centralize the location algorithms and all necessary computing into one central processing unit. All applications and objects are then dependent on the performance and reliability of this unit. That limits the design and implementation of application in a reasonable extend. Applications that work on distributed objects including the need of a position statement of those objects would never work outside the coverage of the localization enabling infrastructure.

Most of the applications built on those location systems are as well infrastructure related. Such as surveillance of the presence of documents in offices or people in a plant to track their movements and find their current position. Such surveillance systems normally collect all the data from the objects that are being tracked and combine them with knowledge from databases and e.g. floor plans to generate maps or log the history. Such a scenario strongly depends on a centralized analysis and is therefore anyway in the need of an infrastructure that brings together location information and databases.

Based on the badge system, e.g. interactive user interfaces have been built that use the location of general trigger devices like a mobile button to trigger events in a backend system to fulfill e.g. home automation tasks. Other applications automatically control the environment depending on the present people like switching displays nearby a person to their personal desktop or adjust interactive panels to a users' profiles when they come in close range. Those application are back-end centric and need an established infrastructure not only for the location algorithms but as well for the applications themselves. In those cases, arguing for an infrastructure-based location system seem rational.

In contrast to this, application that are not backend centric should not depend on any infrastructure. Applications that work on a peer to peer basis must be freed from the limitations of the infrastructure. That is where relative positioning plays an important role.

RELATIVE POSITIONING

Relative positioning in this context means that objects determine their spatial relation not using an underlying infrastructure. Necessary sensing and measurements such as time of flight of signals or intensity of emitted fields are taken only between enabled objects not including a central access point or any external support. The objects containing the enabling technology can determine their relative position only depending on the partner object that as well is enhanced with the positioning technology. This means, an active collaboration between the objects is intended.

This approach brings in some new challenges special to the distributed and intercommunicative design. First, scalability is an important issue. If objects e.g. measure their spatial parameters in a pair-wise way, the scaling of the whole system is pretty poor as the number of necessary measurements increase strongly with an increasing number of active objects. Concepts for efficient measurements must be developed that several objects can determine their spatial relationship at the same time – ideally using the same emitted signal from one partner object. Secondly, the choice of technology plays an important role. Technologies that rely of a line-of-sight connection between the corresponding objects have a special weakness for the possible target scenario. It's much easier to guarantee a line-of-sight connection between a mobile object and an installed infrastructure than between two mobile objects. The infrastructure could cover e.g. complete walls or floors and ceilings in buildings which clearly increases the probability of having line-of-sight to the infrastructure components. Mobile objects in contrast will not have line-of-sight connection between each other in many cases. Thirdly, as the relative positioning should work in an independent way, all computation must be done within the object. This requires all sensing and computation hardware to be included in the mobile objects and can significantly influence the size and weight of objects that are subject to localization. On the other hand, especially the distributed and self-sustaining computation brings excellent scalability. The more objects participate, the more computation power is available. A positive side-effect of robustness appears when sensing and localization of partner objects is done in a distributed and decentralized way: The drop out of one object will generally not affect the functionality of the remaining ones.

DECENTRALIZED COLLABORATIVE SENSING

The decentralized sensing appears to be one of the most challenging issue in this context, as technologies must be found that can be used in mobile and small objects. Research has started to evaluate different possibilities like [1] and [4]. The sensing and measuring of the spatial relation of object must be designed in a way that it works collaborative. This means, objects would not sense their environment and discover other objects which are physically there, but the objects would find each other e.g. through a broadcast communication on a radio frequency channel and then determine their location actively together. That is as well an important condition for the scaling of the whole system. Exchanging existing data to minimize the actual sensing is as well necessary.

Additionally, energy issues and the size will be important considerations for the system implementation. The advantages and new problems of a distributed approach will include independent and ubiquitous operability, cost and set-up time, broad- and multicast communication.

THE RELATE PROJECT

In the RELATE project [1] we investigate relative positioning in the specific context of tangible interfaces that involve spatial arrangement of physical interaction objects on 2D surfaces, such as white board or tables. Relate is an approach that uses dedicated positioning technology to obtain finer-grained relative position, targeted at close-range operation. The research is driven by positioning requirements that we observe in tangible interface systems composed of physical interaction objects. Tangible interfaces have recently attracted considerable research interest, as part of the paradigm shift toward ubiquitous computing, aiming to provide interaction in ways that are intuitive and seamlessly integrated with people's activity in a physical world [6].

We seek to make such interfaces independent of particular environments and fit for deployment beyond lab spaces. We specifically target objects that collectively constitute a tangible interface but the approach applies to mobile devices in general. The embedded technology allows objects to measure distance from neighboring objects as well as the angle at which these are observed. It further foresees wireless communication for the objects to share observations and to collectively establish the overall spatial configuration of the set of objects.

In particular, we discuss the following relevant research issues during the RELATE project: First the enabling technology that includes the actual position sensing. Different technologies are such as infrared light and ultrasound are under evaluation and prototypes are being built. Second, the design of the distributed and decentralized sensing of object in a collaborative manner seams an interesting research issue. Objects, that e.g. create signals for measurements of the location of partner objects expect them to work simultaneously. A pair-wise approach would scale bad and cause slow update rates that might not be useful for the target application: human interfaces.

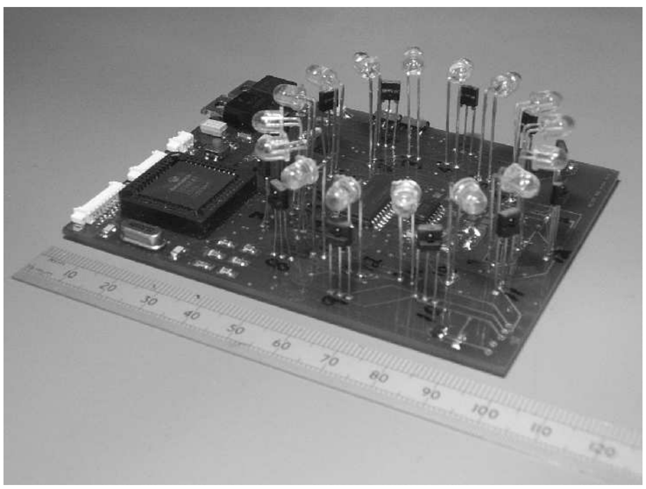

Figure 1

Thirdly, robustness, physical size and weight are properties that are evaluated during the RELATE project. If the enabling technology for acquiring location information is attached to mobile objects or human or even embedded in mobile devices like cell phones or PDAs it must fulfill additional conditions. Those even include energy efficiency and interface definitions. Further, we discuss the theory of combining data from distributed measurements to even draw a complete picture of the united arrangement of objects. In a first step, we focused our investigation on the use of infrared light intensity as basis for relative position measurement. We have implemented a number of device prototypes (see Figure 1) to facilitate initial small-scale experiments to inspect accuracy and robustness of this particular method for relative positioning. Those prototypes can cover a range of distances up to 2.5 meters and work distributed and independently together. The communication for organization and data exchange is processed on the Smart-Its platform [7] which provides an ad hoc networking with a maximum bandwidth of 125kBit/s.

AUTHOR

Albert Krohn has written his master thesis on promoting ideas towards "spectrum pooling", which involved new concepts for simultaneously running concurrent protocols. During his time as a student at TecO he established his interest in ubicomp research and location issues mainly through the MediaCup [2] project. As a research assistant at TecO he now focuses on a positioning system for ubicomp as well as adaptive ad hoc communication and human computer interaction.

REFERENCES

1. http://ubicomp.lancs.ac.uk/relate
2. http://mediacup.teco.edu
3. Roy Want, Jeffrey Hightower and Gaetano Borriello. Spoton: An indoor 3d location sensing technology based on rf signal strength. Seattle, WA, February 2000.
4. Bulusu, J. Heidemann, and D. Estrin: GPS-less low-cost outdoor localization for very small devices. IEEE Personal Communication, vol. 7, pp. 28–34, October 2000.
5. R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system," ACM Transactions on Information Systems, vol. 10, pp. 91--102, Jan. 1992
6. P.Dourish :Where the action is: The foundations of embodied interaction MIT Press, 2001.
7. Zimmer T. Krohn A. Decker C. Beigl, M. and P. Robinson: Smart-its - communication and sensing technology for ubicomp environments. Technical Report ISSN 1432-7864 2003/2., 2003.

Probabilistic Inferencing for Location

John Krumm
Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA USA
jckrumm@microsoft.com

Keywords
Ubiquitous computing, pervasive computing, location, tracking, probability

INTRODUCTION

Automatically measuring the location of a person or device for ubiquitous computing always involves the conversion of a raw measurement into a location measurement. While this process is a prepackaged part of some sensors (*e.g.* GPS and cell phones), researchers are often faced with making this conversion on their own as part of their efforts to deploy novel sensing technologies. This paper describes and discusses various general techniques that researchers have adopted for processing sensor readings into location measurements, emphasizing probabilistic approaches. Reasoning probabilistically is attractive because it naturally accounts for the uncertainty and ambiguity of sensor data, and a probabilistic representation is a good way to communicate uncertainty to higher level modules that exploit location. The paper concludes by advocating recursive filtering as the best general technique to use, with particle filtering having a slight advantage over the next best recursive technique, the hidden Markov model.

DETERMINISTIC FUNCTION INVERSION

Sometimes there is no probability involved. Suppose that at time t a sensor (or sensors) produces a vector of l measurements $\mathbf{z}_t$. This $l \times 1$ measurement vector could be l signal strength measurements from l wireless access points, or l ultrasound delays from l ultrasound detectors, or any l measures from one or more sensors. The location at time t is represented by the vector $\mathbf{x}_t$, which is what we want to estimate. $\mathbf{x}_t$ is an $m \times 1$ state vector whose elements might represent a position in space, orientation, velocity, or any combination of state variables that need to be inferred.

It is sometimes possible to model the output of the sensor as a deterministic function of the input, i.e.

$$\mathbf{z}_t = h(\mathbf{x}_t) \quad (1)$$

If $h(\mathbf{x})$ can be inverted, then $\hat{\mathbf{x}}_t = h^{-1}(\mathbf{z}_t)$, where $\hat{\mathbf{x}}_t$ is the estimate of the state vector. If $h(\mathbf{x})$ cannot be inverted, then a common technique is to find the state vector that minimizes $\left\|\mathbf{z}_t - h(\mathbf{x}_t)\right\|^2$ using an iterative least squares algorithm like Levenberg-Marquardt[1].

As a special case, if the measurement and state vectors are related linearly as $\mathbf{z}_t = H\mathbf{x}_t$ by the $l \times m$ matrix H, then the least squares solution has the closed form

$$\hat{\mathbf{x}}_t = \left(H^T H\right)^{-1} H^T \mathbf{z}_t \quad (2)$$

For most problems of interest, however, the relationship between the state vector and measurement vector is not deterministic, so probabilistic methods must be used.

MAXIMUM LIKELIHOOD ESTIMATE

Often the relationship between sensor readings and state variables is characterized by a state-conditional probability $p(\mathbf{z}_t | \mathbf{x}_t)$. This is merely a probabilistic sensor model giving the distribution of measurement vectors for a given state variable. It can be determined by simulating the sensor or by taking enough actual measurements to make a histogram of the frequency of measurement values as a function of known inputs.

Given a measurement $\mathbf{z}_t$, the maximum likelihood estimate of the state is the state that maximizes the state-conditional probability:

$$\hat{\mathbf{x}}_t = \arg\max_{\mathbf{x}_t} p(\mathbf{z}_t | \mathbf{x}_t) \quad (3)$$

One special case is a generalization of the deterministic linear relationship in the previous section. Here $\mathbf{z}_t = H\mathbf{x}_t + \mathbf{v}$. $\mathbf{v}$ is a $l \times 1$ normally distributed noise vector with zero mean and $l \times l$ covariance matrix R, *i.e.* $\mathbf{v} \sim N(\mathbf{0}, R)$. The maximum likelihood estimate is similar to Equation (2), but accounts for the covariance of the noise[2]:

$$\hat{\mathbf{x}}_t = \left(H^T R^{-1} H\right)^{-1} H^T R^{-1} \mathbf{z}_t \quad (4)$$

Another special case commonly occurs when the state space is discretized into a finite number of classes, $C = \{c_1, c_2, \ldots, c_n\}$. For instance, the classes might represent different rooms of a building or different discrete points on the floor. The state-conditional probability is then conditioned on discrete states -- $p(\mathbf{z}_t | c_i)$ -- and the maximum likelihood estimate is simply the state with the largest state-conditional probability evaluated at $\mathbf{z}_t$.The discrete states mean that this is actually a pattern classification problem[3].

Maximum likelihood is a widely accepted method for making probabilistic inferences in the absence of prior assumptions, dynamic models on the tracked subject, and past data. Dealing with prior assumptions is discussed in the next section on maximum *a posteriori* estimates, and dynamic models and past measurements are discussed in the section on recursive filtering.

MAXIMUM *A POSTERIORI* (MAP) ESTIMATE

MAP estimates depend on prior probabilities about the actor's state, denoted as $p(\mathbf{x}_t)$ for the continuous state case and $p(c_i)$ for the discrete state case. As an example, the priors might encode the fact that people generally don't spend much time in the hallway and are much more likely to be found in their offices.

For the continuous state case, the *a posteriori* probability distribution of state given a measurement is given by Bayes' rule:

$$p(\mathbf{x}_t|\mathbf{z}_t) = p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t)/p(\mathbf{z}_t) \qquad (5)$$

The maximum of $p(\mathbf{x}_t|\mathbf{z}_t)$ over $\mathbf{x}_t$ is the MAP estimate $\hat{\mathbf{x}}_t$. $p(\mathbf{z}_t)$ is unaffected by $\mathbf{x}_t$, so it is sufficient to maximize the numerator, which is just the product of the state-conditional probability and the *a priori* probability. Thus the only difference between the maximum likelihood estimate and MAP estimate is the inclusion of *a priori* assumptions for MAP. It is usually easy to make reasonable prior assumptions about peoples' location for tracking, so MAP is an easy way to improve accuracy.

RECURSIVE ESTIMATES

The techniques discussed so far lack the ability to exploit dynamic models of the tracked subject, such as expectations of possible speeds and feasible paths. They also ignore past measurements. Recursive filtering techniques maintain a probabilistic distribution of state that implicitly includes the effect of all past measurements and dynamic assumptions, and they give a technique for updating this distribution with new measurements. By looking back in time, a recursive filter looks at the path of a tracked user instead of just instantaneous position like the memoryless techniques above. Examining a sequence of measurements in time, along with a dynamic model, is an effective way to deal with ambiguious measurements that could have come from more than one location. With these abilities, recursive filtering is generally considered the best way to process sensor data for location. The following sections discuss three recursive filtering techniques: Kalman filter, hidden Markov model, and particle filter.

Kalman Filter

The discrete time Kalman filter is based on simplifying assumptions about both the measurement process and system dynamics. The Kalman filter assumes that the relationship between the measurement vector $\mathbf{z}_t$ and state vector $\mathbf{x}_t$ is linear with zero-mean, additive, Gaussian noise. It also assumes that the relationship between the previous state $\mathbf{x}_{t-1}$ and current state $\mathbf{x}_t$ is linear with zero-mean, additive, Gaussian noise. Mathematically, these assumptions are

$$\mathbf{x}_t = \Phi_{t-1}\mathbf{x}_{t-1} + \mathbf{w}_{k-1} \qquad (6)$$

$$\mathbf{z}_t = H\mathbf{x}_{t-1} + \mathbf{v}_k \qquad (7)$$

Where Φ_{t-1} is an $m \times m$ matrix, H is an $l \times m$ matrix, and $\mathbf{w}_{k-1}$ and $\mathbf{v}_k$ are zero-mean, Gaussian noise vectors.

The Kalman update equations[2] give a simple means of updating the previous state vector with new measurements using closed-form matrix math, resulting in a Gaussian distribution describing the mean and variance of the state estimate.

Some limitations of the Kalman filter for tracking are:

- System dynamics must be linear. This means that sharp turns can be hard to model, and it gives no means of constraining a path from passing through a wall or other barriers.
- Measurements must be linear in state. Most sensor models must be greatly simplified to conform to this assumption.
- Gaussian representation of state. There can be no multimodal estimates of a person's location, and the estimate must always be Gaussian-shaped. This is often much too simplistic for many tracking scenarios.
- Measurement association is fixed. The Kalman filter does not allow any ambiguity in which sensor measurement is associated with which tracked individual. Representing this ambiguity is important for certain "anonymous" sensors like motion detectors and pressure sensors.

Nonlinearities have been addressed with the Extended Kalman Filter (EKF). There are modifications to deal with multimodal distributions. The radar tracking community has developed techniques for reasoning about data association in the context of Kalman filtering[4]. In its natural state, however, the Kalman filter has been surpassed by hidden Markov models and particle filters.

Hidden Markov Model

A hidden Markov model (HMM) represents the state space as a set of n possible discrete states $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\}$. For instance, location on the floor of a building could be represented by a grid of n (x, y) points. These states can only be observed through the measurement vector $\mathbf{z}$, which is related to the states through the state-conditional probabilistic sensor model $p(\mathbf{z}_t|\mathbf{x}^{(i)})$. The states are said to be hidden by $p(\mathbf{z}_t|\mathbf{x}^{(i)})$, thus the "hidden" in HMM. $p(\mathbf{z}_t|\mathbf{x}^{(i)})$ can be any distribution, not just a Gaussian as in the Kalman filter.

The state dynamics of an HMM are governed by a first order Markov assumption that says the current state depends on only the immediately previous state in time. Since the states are discrete, the dynamics is represented by a matrix of transition probabilities $\alpha_{ij} = p(\mathbf{x}_{t+1} = \mathbf{x}^{(j)}|\mathbf{x}_t = \mathbf{x}^{(i)})$. These transition probabilities can be used to suppress impossible jumps between distant points and between points separated by barriers, and they can be adjusted to reflect assumptions on possible speeds.

For each new measurement $\mathbf{z}_t$, the Viterbi algorithm (see [5]) efficiently computes the maximum likelihood path through the states $\hat{\mathbf{x}}_{0:t} = \{\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_{t-1}, \hat{\mathbf{x}}_t\}$ that best accounts for the sequence of measurements $\mathbf{z}_{0:t} = \{\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_{t-1}, \mathbf{z}_t\}$.

The main difficulty with using an HMM for tracking location is that all possible states must be explicitly represented. Adding new dimensions to the state representation causes an exponential increase in the number of states, each of which needs its own $p\left(\mathbf{z}_t \middle| \mathbf{x}^{(i)}\right)$ and transition probabilities to all the other states. This can be mitigated by exploiting independence among subsets of the state variables and using separate HMMs for each subset. For instance, in tracking people, one HMM could be assigned to (x, y) and another could be assigned to speed with, perhaps, the results of the speed inference used to update the transition probabilities for (x, y).

Particle Filter

The particle filter represents a probability distribution of the current state as a set of N state samples and associated scalar weights: $\left\{\left(\mathbf{x}_t^{(1)}, w^{(1)}\right), \left(\mathbf{x}_t^{(2)}, w^{(2)}\right), \ldots, \left(\mathbf{x}_t^{(N)}, w^{(N)}\right)\right\}$. The weights sum to one, and larger weights indicate more likely states. Each of these "particles" is continuous and evolves as new measurements are processed. This is in contrast to the HMM where the states are discrete and predefined. Upon receipt of a new measurement $\mathbf{z}_t$, a new set of particles is computed in three steps:

1. Create $\left\{\mathbf{x}'^{(1)}_{t-1}, \mathbf{x}'^{(2)}_{t-1}, \ldots, \mathbf{x}'^{(N)}_{t-1}\right\}$ by sampling with replacement from $\left\{\left(\mathbf{x}_{t-1}^{(1)}, w^{(1)}\right), \left(\mathbf{x}_{t-1}^{(2)}, w^{(2)}\right), \ldots, \left(\mathbf{x}_{t-1}^{(N)}, w^{(N)}\right)\right\}$, where samples are drawn randomly in proportion to the scalar weights.

2. Propagate these samples to the current time by randomly generating a new sample $\mathbf{x}_t^{(i)}$ from each $\mathbf{x}'^{(i)}_{t-1}$ via the transition probability $p\left(\mathbf{x}_t \middle| \mathbf{x}_{t-1}\right)$. This models the state dynamics. This is the same Markov assumption as the HMM, only for continuous states.

3. Assign a new weight to each $\mathbf{x}_t^{(i)}$ according to the state-conditional probability $p\left(\mathbf{z}_t \middle| \mathbf{x}_t\right)$. Normalize these weights so they sum to one. This gives $\left\{\left(\mathbf{x}_t^{(1)}, w^{(1)}\right), \left(\mathbf{x}_t^{(2)}, w^{(2)}\right), \ldots, \left(\mathbf{x}_t^{(N)}, w^{(N)}\right)\right\}$.

These weighted samples give a versatile way of approximating any *a posteriori* state probability distribution. Each iteration through the three steps requires knowledge of only the probabilistic transition probabilities $p\left(\mathbf{x}_t \middle| \mathbf{x}_{t-1}\right)$ and the state-conditional probabilities $p\left(\mathbf{z}_t \middle| \mathbf{x}_t\right)$. These two probability functions can be arbitrarily complex, allowing for the modeling of realistic dynamics and sensors.

The main problem with particle filters is that the required number of particles N is hard to predict in advance without experimentation, and the number may be unreasonably large depending on the dimensionality of the state space. As with the HMM, there are techniques to exploit the independence of the state variables, such as Partitioned Sampling[6]. In fact the general topic of sequential importance sampling is still an active area of research, and the algorithm presented above is only one of many possible depending on what prior assumptions can be made about the probabilistic processes involved. One example of the use of a particle filter for tracking inside buildings is explained in [7].

CONCLUSIONS

Deterministic methods of sensor interpretation are burdened by the difficultly of modeling all aspects of a location sensing system. Lumping the unmodeled effects under the label "random" leads to probabilistic models which have the benefit of explicitly representing the inherent uncertainty and, depending on the model used, the multimodal ambiguity. Recursive filtering can efficiently take into account dynamic models and past measurements to compute location, which makes them preferred over the memoryless methods of simple maximum likelihood and MAP estimation. Of the recursive techniques, the Kalman filter is limited to Gaussian distributions and linear dynamics, while the HMM and particle filter offer much more flexibility at the expense of more memory-intensive representations. The continuous state representation of the particle filter has a slight advantage over the inherently discrete-state HMM provided the required number of particles is not too large.

REFERENCES

1. Press, W.H., et al., *Numerical Recipes in C*. 1992, Cambridge, UK: Cambridge University Press.
2. Gelb, A., *Applied Optimal Estimation*. 1974, Cambridge, MA USA: The M.I.T. Press.
3. Duda, R.O. and P.E. Hart, *Pattern Classification and Scene Analysis*. 1973, New York: John Wiley & Sons.
4. Blackman, S.S., *Multiple-Target Tracking with Radar Applications*. 1986: Artech House, Inc.
5. Rabiner, L.R., *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.* Proceedings of the IEEE, 1989. 77(2): p. 257-286.
6. MacCormick, J. and A. Blake, *Probabilistic exclusion and partitioned sampling for multiple object tracking.* International Journal of Computer Vision, 2000. 39(1): p. 57-71.
7. Liao, L., et al. *Voronoi Tracking: Location Estimation Using Sparse and Noisy Sensor Data*. in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* 2003.

Fully Distributed Location-Aware Computing

John Light[1], Eric Pattison[2], Trevor Pering[1], Murali Sundar[1], Roy Want[1]

[1]Intel Research
2111 NE 25th Ave. JF3-375., Hillsboro OR 97124
503 264 9080
first.last@intel.com

[2]University of Calgary
Dept. of Computer Science
2500 University Drive NW, Calgary, AB T2N 1N4
ericp@cpsc.ucalgary.ca

ABSTRACT

There have been many proposals for location-aware computing that involve centralized infrastructure. Using cell-phone systems or GPS databases requires considerable fixed resources to maintain location information and provide services. We propose the Ubiquitous Walkabout, which, by using independent Information Beacons and an "always on" mobile device, provides a richer and more flexible user experience. The approach provides more capability, privacy and scalability than a cellular approach (and may also be less expensive), and more capability and flexibility than a GPS approach.

Author Keywords

Location-aware computing, ubiquitous computing, Personal Server, Information Beacons.

THE UBIQUITOUS WALKABOUT

The Ubiquitous Walkabout, part of the Ubiquity project in Intel Research, is an investigation into distributed location-aware computing. It involves Information Beacons, which wirelessly broadcast specific information about a location to a small (30 ft) vicinity, and a Personal Server, which can receive the beacon messages and process them as they are received. Walking down a street equipped with this technology, a user would receive the information from many Information Beacons in turn, each of which would transmit information about a specific service or other offering in its vicinity. Based on previously expressed policies and preference provided by the user, the Personal Server might report the service or offering to the user, respond to the offering automatically, log the information for later use, or ignore it.

THE PERSONAL SERVER

The Personal Server [1] is a capability that can be part of any small mobile device such as a cell phone or PDA. It can run continuously for a long time (one or more days) and provide considerable computational and storage resources to its user during that time. It uses advanced power management capabilities to provide the appearance of being "always on", though it may in fact enter sleep modes from which it can quickly return. In its current form, the Personal Server communicates with the world through one or more wireless (radio) interfaces. It can talk to Personal Computers, public displays, PDAs, cell phones, or personal I/O devices like a wireless watch.

INFORMATION BEACONS

An Information Beacon can be as simple as a radio chip, microcontroller, and power supply. It need only be able to store a small amount of information that it broadcasts repeatedly, and can be manufactured for under $20 in large quantities. To establish the content of a beacon, it can simply be plugged into a standard Personal Computer.

We are currently using Berkeley Motes as Information Beacons. They use a simple radio technology to broadcast a short (300 byte) message repeatedly. We have recently used mote technology based on a Bluetooth radio (iMote) to provide larger messages.

FULLY DISTRIBUTED

The Ubiquitous Walkabout provides a location-aware computing experience that differs from some previous descriptions.

- The Personal Server can store and utilize considerable user context, both explicit and inferred, as well as volumes of acquired content.
- The Personal Server can run agents of considerable complexity on behalf of the user, and the user can freely experiment with choice and configuration of agents.
- The beacon owner has complete control over content, and the Personal Server owner has complete control over response to it.
- All preferences, policies, actions, and recorded data are limited to the Personal Server (or other user specified devices), so privacy is maximized.
- All of the above capabilities can be delivered in a fully scalable manner that doesn't involve a centralized resource.

COMPARISON TO OTHER APPROACHES

We will compare the proposed approach to Location-Aware Computing (LAC) to existing directions. The primary directions being pursued now are:

1. Cell phone-based infrastructure approach. This is based on knowing what cell you are in, which the cellular provider can use to base services on.

2. Cell phone-based database approach. Since the cell you are in is available at the handset, an application running in the handset can provide services based on a local or online database mapping cells to services.
3. 911-based infrastructure approach. Soon most cellular systems will be able to track users to within a few meters (many already do). This information is typically only available to the cellular provider (and its designees).
4. GPS database approach. If the mobile device contains a GPS receiver, it can access a local or online database to find out about services in the vicinity.
5. Hotspot database approach. Since hotspots are common in some areas, a directory mapping hotspot IDs to their locations can be used to access the same databases that the GPS approach uses.

We will discuss these approaches and the proposal in the following dimensions:

1. Scalability.
2. Services.
3. Performance.
4. Reliability.
5. Privacy.
6. Security.
7. Economics.

All of the current approaches have one thing in common: they depend on some form of infrastructure. In the case of the three cellular approaches, the infrastructure includes the cellular system itself. The three database approaches require creation and maintenance of a publicly available database. While the distribution of these databases can be decentralized, their maintenance and control may need to be centralized. The cell-based and 911-based infrastructure approaches involve databases privately held by the cellular providers (and its designees).

Consider the potential magnitude of these central databases. If half the stores in the United States and a third of its citizens wanted to have a LAC presence, the size of these databases would be enormous. You can think about it as the union of all the yellow and white phone books in the U.S. If the LAC is to meet its potential, much of this information would need to be dynamic (unlike a phone book), and the update process would be daunting (and likely slow). Think of it as if someone would try to implement the web as a centralized database. (The databases could be decentralized, a la Yellow Pages, but even online YPs can take over a week to update.)

The proposed approach eliminates central infrastructure. Just as with the web, it allows the problem to be naturally distributed. Just as with the web, each user communicates directly with the information provider, without an intermediary. As individuals acquire mobile devices that work with the proposal and as shops and individuals acquire information beacons, the system scales naturally.

What services can be offered with the various approaches? We submit that any service can ultimately be offered by any of these approaches, existing or proposed. If the hypothesis is that there is no difference among them in this respect, then we are faced with proving a negative. We invite you to provide counterexamples.

The quality of the services offered may vary, however. The database approaches may suffer update latencies, making it difficult to provide highly targeted changes. The update latency consists of two factors: updating the database itself (including distribution delays) and updating the mobile device with changes. Minimizing both latencies seems difficult. Updating the private databases in the cell-based approaches is not much easier, though the delay updating the mobile device is not present since the LAC functionality is implemented in the cellular provider’s back end, which is near the database.

Another aspect of service quality is the level of innovation that can be brought to bear on the problem. Can vendors try different approaches and change approaches to meet perceived needs. The database approaches allow plenty of innovation using the distributed database data, but they may restrict innovation that requires new data formats or types of information. Cellular approaches require innovation either by the cellular provider or by third parties with access to the providers back end. History has shown cellular providers to be slow innovators, and supportive of third party innovation only at a steep price. The proposed approach, since it is fully distributed, allows innovation on multiple levels since new ideas can be tried out locally and migrated elsewhere at little cost.

Apparent performance will likely be seen as the timeliness of getting up-to-date information from the source (e.g., store) to the user. This time includes both the time to determine location and the timeliness of the database updates. The cellular approaches and the proposal can determine location quickly. The GPS approach can suffer acquisition delays or failures. The database update problem was discussed above.

All of the approaches depend to some extent on real-time radio traffic, which is inherently unpredictable. Cellular providers have spent vast sums to ensure reliable operation under most circumstances. GPS systems are known to fail in urban areas and seldom work indoors. The proposed approach depends on untested (in this use) radio technology, so its reliability is yet to be determined. We believe it can be made locally reliable, but many problems will have to be worked out.

Privacy is critical to user acceptance of Location-Aware Computing. There are many aspects to privacy, and the various approaches affect different aspects differently.

Here are some relevant privacy issues:

1. Stealth. This is the ability to be somewhere without anyone knowing you are there. GPS preserves stealth since the device doesn't radiate. Cellular approaches are not stealthy, but we assume that no one is going to use the information. Information beacons can preserve stealth if they don't require a response from the mobile device.
2. Anonymity. This is freedom from being identified. As long as you have stealth privacy, you have anonymity, but if the stealth is not preserved, you might still be able to acquire beacon information without disclosing your identity or some information that could be used to establish your identity (such as a MAC address).
3. Tracking. This is the ability for someone to track your comings and goings through your use of the technology. The cellular network can track you, and with the advent of 911 technology, it can track you in great detail. The GPS user is largely safe from being tracked, except by observing downloads of the services database. Information beacons allowing stealth operation can't be used to track you, but if they require non-anonymous ID, or even a consistent anonymous ID, they can.
4. Interests. Independent of knowing where you are, there is the question of learning about your interests. Can an outsider learn about your preferences and interests by observing some aspect of your behavior? Observation of selective incremental downloads of a database could be used for that purpose. The proposed approach provides this class of privacy if using stealth operation.
5. Data. This is specific information about you, typically collected by you for your own use. Access by others is considered a violation of privacy. Protection of this data is usually implemented as a form of computer security, which is discussed in a later paragraph.

All of these privacy issues depend to a large extent on how well you trust the support infrastructure that is required to provide it. Most people trust their cellular provider with all these forms of privacy. We trust that the provider isn't tracking us or listening in on our conversations with the bank. Public databases are another thing. Who is going to manage and monitor those databases? Only the current proposal in stealth mode preserves all of these, with some question about data privacy.

Security is a difficult question because all security is relative to expectations. We trust the cellular provider to keep information about us secure, and mostly it is. The public database proposals, along with the current proposal, share the issue that the security of the mobile device determines our overall security since they contain considerable information about us. Can mobile devices be made secure? We suspect they can be secured against access through their normal I/O mechanisms, but what happens when the device falls into a miscreant's hands? As more and more important information is kept on mobile devices, the importance of this question grows.

What are the economics of the various approaches? The cost of GPS capability in a mobile device will continue to drop, so we assume that it will eventually be nearly free. The cost of cellular service is well understood, but the cost of backend services is not. Some reports on European experience imply that cellular providers are loath to part with location information, making it intolerably expensive. This begs the question of what their actual costs of providing these services might be. The cost of maintaining and delivering public service databases is unknown, but it will probably involve a monthly fee, since the information must be kept fresh. The proposed approach involves a small incremental cost to a standard mobile device (for the Personal Server capability) and a cost of $20-50 to purchase each information beacon, and some cost to run it. We suspect that the one-time cost of an information beacon is much smaller than the fees to support the ongoing costs of maintaining public or private service databases.

An additional question for all these approaches is how you start them up, since they suffer to varying extents from the "chicken and egg" problem.

CONCLUSION

We have presented an alternative approach for Location-Aware Computing that offers several advantages over current approaches. We discussed what those approaches are and their advantages and disadvantages relative to the proposed alternative.

We hope we have successfully conveyed that the proposal is worthy of more extensive consideration by the larger research community.

REFERENCES

1. Want R., Trevor Pering, Gunner Danneels, Muthu Kumar, Murali Sundar and John Light. *The Personal Server: changing the way we think about ubiquitous computing*,. Proceedings of UbiComp 2002, Springer LNCS #2498, Goteborg, Sweden, pp194-209.

BIOGRAPHY

John is a member of the Ubiquity Strategic Research Project in Intel Research, working with Roy Want and others to investigate the applicability of the Personal Server concept to various aspects of Ubiquitous and Proactive Computing. Prior to that, John worked on Information Visualization and Knowledge Management research at Intel, including the Miramar 3D workspace, and at other companies on graphics, operating system, and CAD products.

Author Index