

The Location Stack

Jeffrey Hightower, Dieter Fox, and Gaetano Borriello

UW-CSE-03-07-01
University of Washington
Computer Science and Engineering
Box 352350
Seattle, WA 98195

Abstract. The Location Stack is a set of design abstractions for location systems for ubiquitous computing. Our contribution presented in this paper is twofold. First, the Location Stack design abstractions themselves give the ubiquitous computing community a common vocabulary and comparative framework for multi-sensor location systems. Second, our implementation of the Location Stack employs novel probabilistic techniques for fusing readings from multiple different sensor technologies while providing a uniform programming interface to applications. Our implementation is publicly available and supports many existing location technologies.

1 Introduction

Location is essential information for many ubiquitous computing systems: We want our home to learn and respond to its inhabitants' movements. We want to capture and optimize workflow in a factory. We need directions from one place to another. We want to interact naturally with input-output devices casually encountered in the environment. Yet despite the need for location information, existing location-aware ubicomp systems could be improved in two areas:

1. Solid design abstractions can provide a common vocabulary and enable comparative evaluation of the many existing location systems.
2. Fusing readings from multiple different sensor technologies can exploit the advantages of each technology while presenting a uniform location programming interface to applications.

Our contribution is in both of these areas. Based on lessons from a previous survey of location systems [4] we present the abstractions and an implementation of the Location Stack, evocative of the Open System Interconnect (OSI) model of computer networks. The Location Stack is a common vocabulary and general framework for multi-sensor location-aware ubiquitous computing.

Section 2 presents the Location Stack abstractions. Section 3 then describes our publicly available implementation. We implement the bottom four abstraction layers and are engaged in active collaborations to develop the remaining two. In this paper, we give particular emphasis to our Fusion layer's use of Bayesian

filter techniques, more specifically, particle filters and multi-hypothesis tracking to estimate location in real multi-sensor environments. Our implementation supports many location sensor technologies including infrared proximity badges, passive RFID tags, ultrasound ranging tags, active radio proximity tags, global positioning system receivers, infrared laser range-finders, 802.11b wireless clients, and, more importantly, any combination of these. Our architecture consists of scalable distributed services communicating with asynchronous XML messages and remote procedure calls, similar to many modern ubiquitous computing systems. Finally, Section 6 concludes with a discussion of the current state of our work and its future directions.

2 The Location Stack Abstractions

Modern networking technology is easily understood largely thanks of the Open Systems Interconnection (OSI) abstraction layers. A client application need not worry about physical layer variations such as Ethernet versus 802.11 versus cellular packet radio. As long as it speaks HTTP, a web browser can post and retrieve data from remote servers. Location-aware ubicomp systems need similar abstractions to provide a common vocabulary and to avoid common inefficiencies such the tendency to create specialized sensing hardware for every new ubicomp scenario. The Location Stack codifies such a set of abstractions based on classifiers and properties identified in a previous survey of location systems [4]. Figure 1 shows the Location Stack (note that the layers are quite different then the OSI model, which we only use as an evocative analogy).

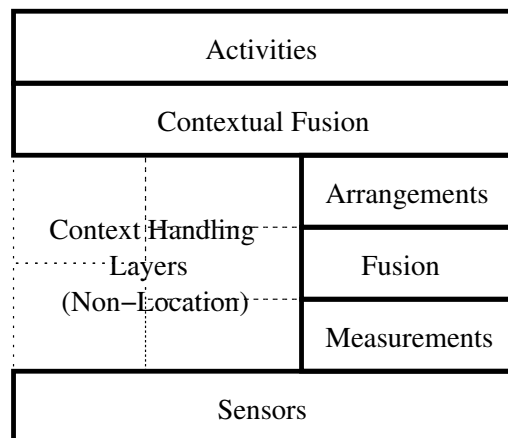


Fig. 1. The Location Stack abstractions are a general framework and common vocabulary for location-aware ubiquitous computing systems.

We now briefly discuss the abstraction layers and the interfaces they provide.

2.1 Sensors

The Sensors layer consists of the sensing hardware for detecting a variety of physical phenomena. The interface to this layer is usually very specific to particular sensing technologies; software drivers deliver raw data in a wide variety of formats. Example outputs include blob pixels from a camera, ultrasonic time of flight measurements, keystrokes from a computer login event, or a proximity beacon event. Information flow is predominantly up the stack as sensors generate new information about the changing state of the physical world.

2.2 Measurements

The Measurements layer transcribes raw sensor data into canonical measurement types for use by higher level operators. For example, data reported by many ubiquitous location sensors can be classified as distances, angles, proximities, or positions. Each measurement may also have an uncertainty model derived from the physical characteristics of the sensor and the environment. For example, infrared badge and RFID sensors both produce proximity measurements with respective uncertainty models based on the power of the infrared emitters and the range and antenna characteristics of the radio. Information flow is again predominantly up the stack as sensors generate new measurements about the changing state of the physical world.

2.3 Fusion

The Fusion layer continually merges measurements into a probabilistic representation objects' locations and presents a uniform programming interface to this representation. The Fusion layer also often addresses the interlaced problem of estimating objects' identities in situations where explicit identity information is not provide by the lower layers.

No particular representation or coordinate system is implied. Representations could be symbolic room numbers, a simple two dimensional geometric coordinate, or a highly complex state such as three dimensional position, pitch, roll, yaw, and derivatives (speed, acceleration). Fusion implementations may choose to use a standard map datum (*e.g.* NAD27 or WGS84 used by GPS and topographic maps), implement a custom coordinate system, or implement converters allowing free representation in multiple coordinate frames.

The most significant contribution of our implementation presented in Section 3 is at the Fusion layer. We apply Bayesian filter techniques including particle filters and multi-hypothesis tracking to both the basic sensor fusion problem and the problem of simultaneous identity estimation.

2.4 Arrangements

The Arrangements layer provides operators to relate the current probabilistic location estimates of *multiple* objects which are individually locatable using the Fusion layer interface. Example operators include tests for multi-object proximity, containment in a region, or for specific geometric formations. The operators often produce probabilistic output, for example, a pairwise confidence matrix that a given group of objects are within 4 meters of one another or the probability that 4 people are in a given room.

2.5 Contextual Fusion

The Contextual Fusion layer combines location or arrangement information with other non-location contextual information such as personal data (schedules, email threads, contact lists, task lists), temperature, light level, and galvanic skin response. The interface allows applications to recognize interesting states or sequences that may be useful to a variety of applications.

2.6 Activities

Applications are concerned with users' activities. The reason to capture location and other context data is typically not for direct use but to enable task-level reasoning. For example, ubiquitous applications want to respond a certain way when all residents of a house are asleep for the night, a presentation on zoology is starting in Conference Room B and certain participants are missing, or an Alzheimer patient has deviated significantly from their daily routine.

The Activities layer categorizes context information into semantic states defining an application's interpretation of the world. The interface to the Activities layer is application specific. A common implementation is a set of rule-based triggers driving particular application scenarios or supporting common user tasks.

3 Implementation

Our implementation of the Location Stack is a publicly available Java package containing a useful framework for operating a multi-sensor location system in a ubiquitous computing environment. Our implementation is deployed in our laboratory and workspace and is used by other other research projects as a reliable source of object location information. The implementation is typical of a modern ubiquitous computing system: the architecture consists of a set of reliable distributed services communicating using asynchronous XML messages and remote procedure calls. Services are connected using dynamic service discovery capability provided by the middleware. Key services include the following:

1. Each sensor technology has one or more custom driver services implementing a standard interface. Device drivers interact with sensor hardware and multicast sensor measurements to other interested services such as fusion services and logging services.
2. The database services manage objects and provides administrative functionality to group objects into hierarchies. For example, our experiments rely on physical track packs consisting of two infrared badges and an ultrasonic ranging tag. Managing these packs as a hierarchical group of sensors makes it convenient to dynamically associate and disassociate the packs with people and test subjects.
3. The fusion service runs the sensor fusion algorithms. Multiple fusion services running simultaneously can provide higher performance, enhanced reliability, and be used to comparatively evaluate different fusion algorithms on real-time data.
4. A simulator service allows us to replay measurement sequences either captured by the logging services or synthesized algorithmically in order to evaluate improvements to the sensor fusion algorithms or to replay application scenarios.

3.1 Sensors and Measurements

Our implementation has driver implementations for many common ubicomp location technologies including infrared proximity badges, passive RFID tags, ultrasound ranging tags, active radio proximity tags, global positioning system receivers, infrared laser range-finders, and 802.11b wireless clients. Each sensor driver discretizes and classifies the data produced into measurements of type Distance, Angle, Proximity, or Position as well as several aggregate types such as Scan (a distance-angle combination). Each measurement has an uncertainty model derived from the physical characteristics of the sensor and the environment. This measurement likelihood model $p(z | x)$ describes the probability of observing a certain sensor measurement z given a location x of the person or object. Such a model consists of two types of information: First, the sensor noise and, second, a map of the environment. The problem of constructing maps of indoor environments has already received substantial attention in the robotics research community [11] and is not the focus of our research. Fig. 2(a)–(c) illustrate sensor models for ultrasound tags, infrared badges, and laser range-finders, respectively.

Fig. 2(a) shows the likelihood model at all locations in the environment for a 4.5 meter ultrasound distance measurement. The likelihood function is a ring around the location of the sensor where the width of the ring is the uncertainty in the measured distance. Such noise may be represented by a Gaussian distribution centered at the measured distance. Furthermore, since ultrasound sensors frequently produce measurements that are far from the true distance due to reflections, all locations in the environment have some likelihood, as indicated by the gray areas in the map. White areas indicate locations blocked by obstacles.

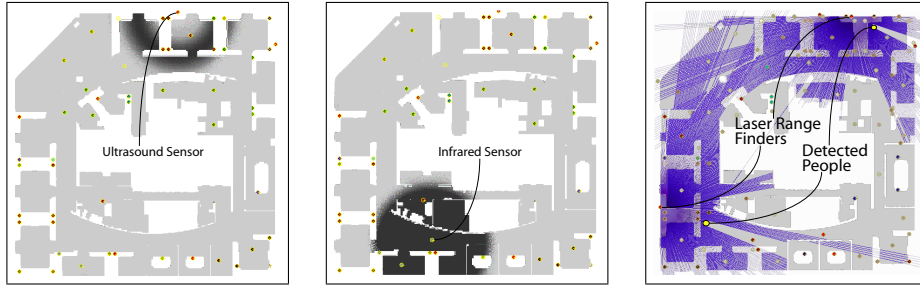


Fig. 2. Measurement likelihood models for (a) ultrasound tags, (b) infrared badges, and (c) laser range-finders. Darker areas represent higher likelihoods in (a) and (b) while the individual laser beams and the detected person are shown in (c).

Fig. 2(b) illustrates the sensor model for the infrared badge systems. Such sensors provide only proximity information, so likelihood is a circular region around the sensor location. Fig. 2(c) show scans taken from two laser range-finders. Since these sensors are placed at fixed positions, it is straightforward to detect sensor beams that return unexpectedly short measurements indicating the presence of people. The likelihood for such detected features is usually represented by a Gaussian distribution with small uncertainty.

3.2 Fusion Layer

Our Fusion layer implementation applies Bayesian filtering techniques to the problem of location estimation. We present two scenarios to illustrate the capabilities of our fusion implementation. First, we show fusion of two representative location technologies: infrared proximity badges and ultrasonic time-of-flight tags. Our hardware is the commercial VersusTech infrared badge system (<http://www.versustech.com>) and MIT's Cricket ultrasound tags [8]. Second, we present a novel approach to multi-object tracking that combines the accuracy benefits of anonymous sensors like scanning laser range finders with the identification certainty of less accurate infrared and ultrasonic ID-sensors. Our anonymous sensor hardware is the LMS200 180 degree scanning laser range-finder from SICK, Inc (<http://www.sick.com>). Figure 3 shows a photo inside our multi-sensor environment.

Multiple Inaccurate ID Sensors In this example we illustrate how we estimate the location of a person using multiple inaccurate ID sensors such as the ultrasound and infrared badges. Due to these sensors' low accuracy, the belief over the person's location is typically very uncertain and multi-modal, hence we apply a Bayesian filter techniques called particle filters which is commonly used in robot localization and is optimized for this type of scenario. Particle filters can naturally integrate information from different sensors. Refer to [3] for an in depth treatment of particle filters and Monte Carlo statistical techniques.

Fig. 4 shows snapshots from a typical sequence projected onto a map of the environment. The person is wearing an infrared badge and ultrasound tag and starts in



Fig. 3. A photo of the environment where our applications are deployed. Our sensing hardware for these examples is VersusTech infrared badge system, MIT's Cricket ultrasound tags scanning laser range finders from SICK, Inc.

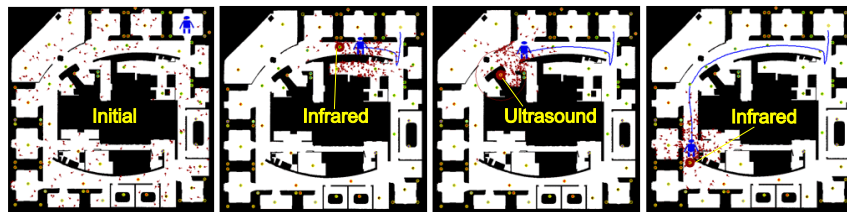


Fig. 4. Sensor fusion of infrared and ultrasound sensors. Density of the particles reflects the probability posterior of the person's location.

the upper right corner of the lab, as indicated by the icon. Since the start location is unknown to the system, the particles are spread uniformly throughout the free-space of the environment. The second picture shows the location probability after the person moved out of the cubicles and into the upper hallway. At this point, the samples are spread over different locations, which is due to the inaccurate sensor information. After the person is detected by an ultrasound sensor, the location can be estimated more accurately, as shown in the third picture in Fig. 4. Later, after moving down the hallway on the left, the samples are spread over a larger area, since this area is only covered by infrared sensors, which only provide very coarse location information.

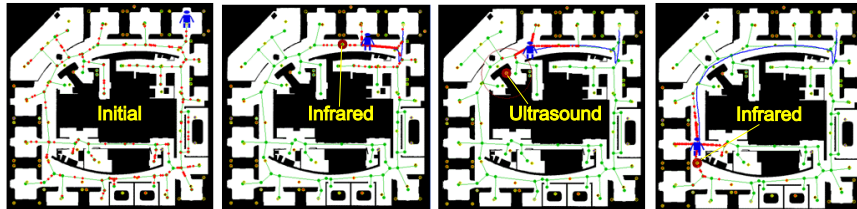


Fig. 5. Sensor fusion of infrared and ultrasound sensors using particle filters constrained to a Voronoi graph.

Particle filters can be used more efficiently by constraining the possible locations of a person to locations on a Voronoi graph of free space that naturally represents typical human motion along the main axes of the environment. The sequence shown in Fig. 5 is based on the identical data as the one using unconstrained particle filters. In experiments we found that such Voronoi graph tracking results in better estimates with less computation. Furthermore, the Voronoi graph structure can be used to learn high-level motion patterns of a person. For example, the graph can capture information such as “Joe typically goes into room 22 when walking down hallway 9”. More details on using Voronoi graphs with particle filters and on applying high-level learning can be found in [6,9].

Combination of Anonymous and ID Sensors One of the disadvantages of common ID sensors such as ultrasound badges is their limited accuracy. On the other hand, sensors such as laser range-finders provide highly accurate location estimates, but do not provide information about a person’s identity. In this example we show how to combine information from both types of sensors to get accurate location and ID information.

Fig. 6 illustrates the main problem of tracking multiple people with anonymous and ID-sensors. Initially, the identity of the two people is not known. As they walk, the anonymous sensor observes their locations frequently. In the beginning, the people are far enough apart so that their positions can be tracked reliably using the anonymous sensor. However, until they reach ID-sensor areas 3 and 4, both trajectories have the same probability of belonging to either person *A* or *B*. Hence there are two different hypotheses for the ID’s of the two trajectories. After passing through the coverage of ID-sensors 3 and 4, the ambiguity is resolved and the ID’s of both trajectories are

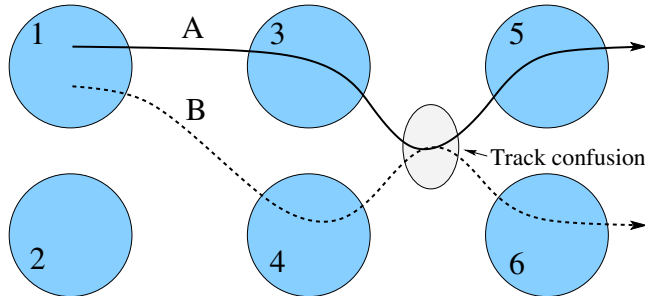


Fig. 6. The blue shaded circles indicate areas covered by infrared receivers. Since these sensors provide no information about the person's location within the area, two people in the same area can not be distinguished. A laser range-finder scans the entire scene and provides accurate location information at a high-rate, but no information about their identities.

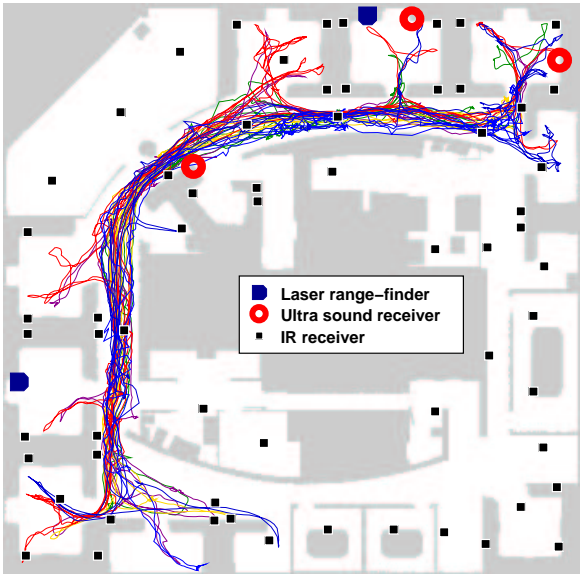


Fig. 7. Our mixed ID-anonymous approach can track the trajectories and identities of six people moving through the environment.

determined. Then, after the paths cross there is confusion about the continuation of the two tracks. When the people leave the light gray area, the anonymous sensor can not determine which observations to associate with which trajectory. This is called the *data association* problem in the multitarget tracking community [2]. If there are no ID-sensors, it would be impossible to resolve this ambiguity. In our approach the ambiguity can be resolved as soon as the people reach the areas covered by ID-sensors 5 and 6. To do so, however, it is necessary to maintain the hypotheses for both possible track continuations, *A* going down and *B* going up, or *B* going down and *A* going up.

To solve this problem we use a combination of particle and Kalman filters, closely related to multi-hypothesis tracking. We track individual persons using Kalman filters, which is possible due to the accuracy of laser range-data (see Fig. 2(c)), and maintain multiple hypotheses regarding the identities of people using a particle filter. Each particle is one hypothesis for the identity of a tracked person: a collection of Kalman filters annotated with identities. Refer to [10] for more details. The resulting approach is able to track several people and their identities in real-time as shown in Fig. 7. The duration of the log in this example is 10 minutes. During this time the six individual people moved between 230 and 410 meters each. In this challenging log, people's paths frequently cross and there are situations in which up to 4 people are occluded to the laser range-finders by other people, yet our approach can successfully estimate both the location and identity of each person.

4 Arrangements

We currently provide two operators to relate the probability posteriors of *multiple* objects which are individually locatable using the Fusion layer interface. We provide the ability to test for *multi-object proximity* given a specified distance and to test for object *containment* with a predefined region of a map. Because we operate on the location probability posteriors, the results of these tests can also be probabilistic and specify the confidence that the test is true. For example the proximity test produces a pairwise confidence matrix that a given group of objects are within 4 meters of one another. Taken together, these operators provide a probabilistic implementation of the "programming with space" metaphor as used with great success in AT&T Sentient Computing project [1]. Future work in our implementation of the Arrangements layer is to provide an additional operator to test for more general geometric formations of multiple objects.

5 Context and Activities

Our implementation of the Context and Activities layers is in its infancy because these abstraction layers are largely unexplored. Few ubiquitous computing systems have been deployed which take sensor information all the way up to the level of human activity inference. To make inroads, we are collaborating with the Assisted Cognition research group, a group seeking to create novel computer systems that will enhance the quality of life of people suffering from Alzheimer's Disease and similar cognitive disorders [5]. Our goal for this collaboration is to design general interfaces for the Context and Activities layer based on usage patterns of the existing Fusion and Arrangements layers in support of these higher level learning tasks. An example project of the Assisted Cognition group is the Activity Compass, a location-aware hand-held computer which

assists Alzheimer sufferers in carrying out their daily routine [7]. Its user interface is a simple arrow directing the user to a location based on the inferred goal. The Activity Compass approach is to learn a model of daily activities from long term data logs of location and other information.

6 Conclusion

Our publicly available implementation of the Location Stack abstractions can perform sensor fusion of multiple common location technologies, provides useful operators to reason about location, and has the potential, with further investigation, to estimate and learn the *activities* of people and what their next goals might be. Our implementation operates 24x7 in our laboratory and workspace and is used by other other research projects to provide reliable object location capabilities.

A single sensor fusion service running on a modern PC (1.8 GHz with 512 MB memory) can perform real-time multi-sensor probabilistic tracking of more than 40 objects at a sustainable rate of 2 measurements per second per object. Objects are tracked using generalized particle filters in 7 dimensions (x, y, z, pitch, roll, yaw, and linear velocity). Higher performance (more objects or a faster measurement rate) can be realized by reducing the state space to two dimensions or through more advanced techniques such as our technique of constraining the particle filters to Voronoi graphs of the environment. We expect to be able to achieve an order of magnitude improvement using these optimizations. Another way to increase performance is to distribute objects across multiple fusion services, although applying certain Arrangements layer operators to distributed objects presents additional challenges which we are investigating.

In summary, the Location Stack abstractions structure location systems into a layered architecture with robust separation of concerns. The Location Stack enables evolution of our systems as we deploy new technologies, supports comparative evaluations of location systems, and allows us to partition the work and research problems appropriately.

References

1. Mike Addlesee, Rupert Curwen, Steve Hodges, Joe Newman, Pete Steggle, Andy Ward, and Andy Hopper. Implementing a sentient computing system. *Computer*, 34(8):50–56, August 2001.
2. Y. Bar-Shalom and X.-R. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. Yaakov Bar-Shalom, 1995.
3. A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo in Practice*. Springer-Verlag, New York, 2001.
4. Jeffrey Hightower and Gaetano Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, August 2001.
5. Henry Kautz, Oren Etzioni, Dieter Fox, Dan Weld, and Lokendra Shastri. Foundations of assisted cognition systems. UW-CSE 03-AC-01, University of Washington, Department of Computer Science and Engineering, Seattle, WA, March 2003.
6. L. Liao, D. Fox, J. Hightower, H. Kautz, and D. Schulz. Voronoi tracking: Location estimation using sparse and noisy sensor data. 2003. Submitted, <http://www.cs.washington.edu/homes/fox/voronoi.pdf>.

7. Donald J. Patterson, Oren Etzioni, Dieter Fox, and Henry Kautz. Intelligent ubiquitous computing to support alzheimer's patients: Enabling the cognitively disabled. In *UbiCog '02: First International Workshop on Ubiquitous Computing for Cognitive Aids*, October 2002.
8. Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proceedings of MOBICOM 2000*, pages 32–43, Boston, MA, August 2000. ACM, ACM Press.
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. Rao-blackwellised particle filters for people tracking with anonymous and id sensors. 2003. To appear.
11. Sebastian Thrun. A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5), 2001.