

Bayesian Filtering for Location Estimation

Bayesian-filter techniques provide a powerful statistical tool to help manage measurement uncertainty and perform multisensor fusion and identity estimation. The authors survey Bayes filter implementations and show their application to real-world location-estimation tasks common in pervasive computing.

Location awareness is important to many pervasive computing applications. Unfortunately, no location sensor takes perfect measurements or works well in all situations. Thus, the motivation behind this article is twofold. First, we believe the pervasive computing community will benefit from a concise survey of Bayesian-filter techniques. Because no sensor is perfect, representing and operating on uncertainty with a statistical tool

such as Bayes filters is key in any system using many sensors.

Second, estimating an object's location is arguably the most fundamental sensing task in many pervasive computing scenarios. It is thus a natural domain in which to illustrate the application of Bayesian filter

techniques. Representing locations statistically enables a unified interface for location information. This lets us write applications independent of the sensors used—even when using very different sensor types, such as GPS and infrared badges. (A comparative survey of location systems appears elsewhere.¹)

Here, we illustrate fusing sensor data from ultrasound and infrared tags. We also discuss how to combine high-resolution location information from anonymous laser range finders with low-resolution location sensors that provide identification.

Bayes filters

Bayes filters² probabilistically estimate a dynamic system's state from noisy observations. In location estimation for pervasive computing, the *state* is a person's or object's location, and location sensors provide observations about the state. The state could be a simple 2D position or a complex vector including 3D position, pitch, roll, yaw, and linear and rotational velocities.

Bayes filters represent the state at time t by random variables x_t . At each point in time, a probability distribution over x_t , called *belief*, $Bel(x_t)$, represents the uncertainty. Bayes filters aim to sequentially estimate such beliefs over the state space conditioned on all information contained in the sensor data.

To illustrate, let's assume that the sensor data consists of a sequence of time-indexed sensor observations z_1, z_2, \dots, z_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 :

$$Bel(x_t) = p(x_t | z_1, z_2, \dots, z_t). \quad (1)$$

Roughly speaking, the belief answers the question, "What is the probability that the person is at location x if the history of sensor measurements is z_1, z_2, \dots, z_t ?" for all possible locations x . In general, the complexity of computing such posterior densities grows exponentially over time, because

Dieter Fox, Jeffrey Hightower,
Lin Liao, and Dirk Schulz
University of Washington

Gaetano Borriello
University of Washington and
Intel Research Seattle

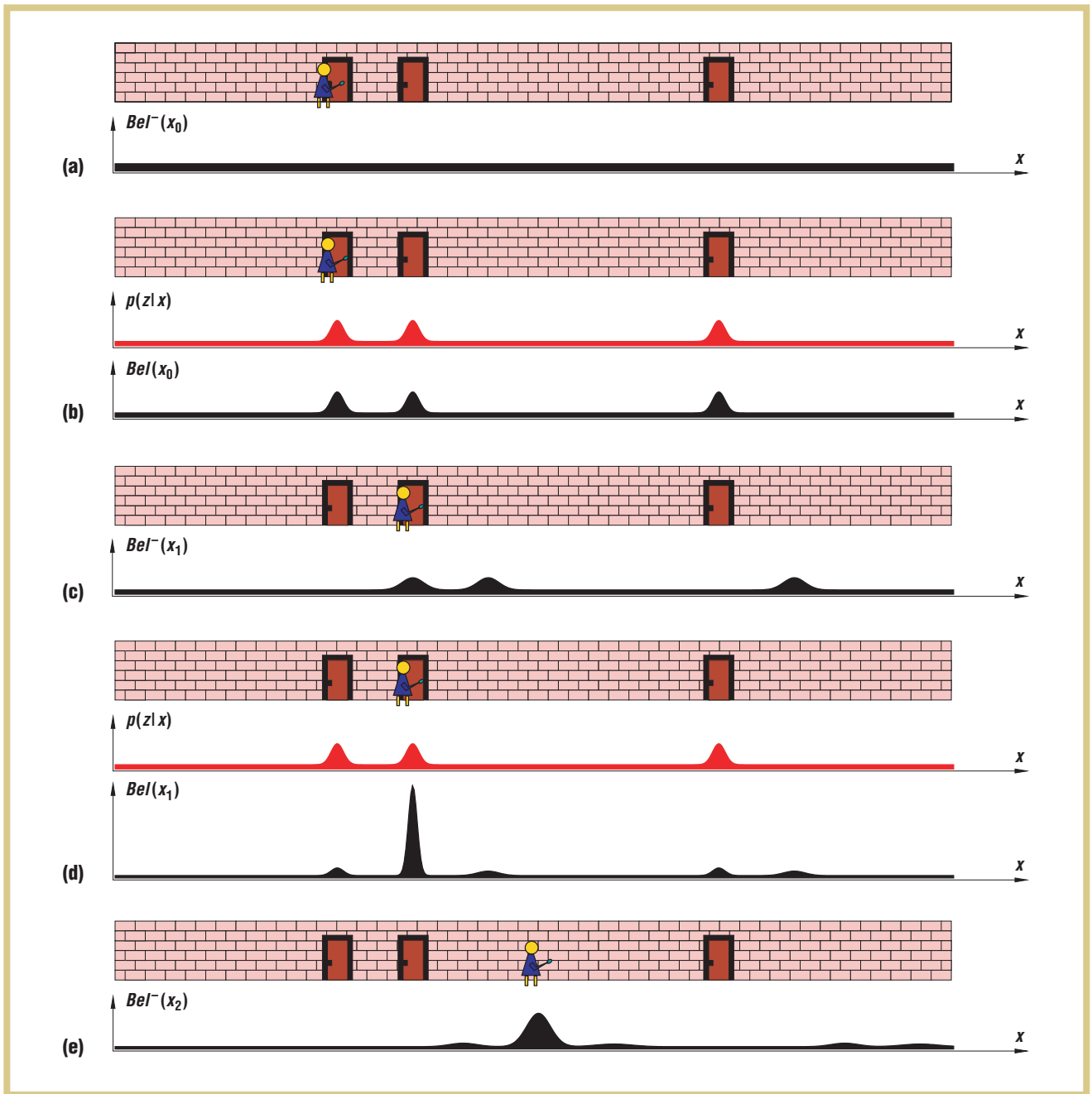


Figure 1. A one-dimensional illustration of Bayes filters. A person carries a door-sensing camera that cannot distinguish different doors. Each frame depicts the person's position in the hallway and the current belief $Bel(x)$: (a) The person's location is unknown. (b) The sensor sends a "door found" signal. (c) The person moves. (d) The sensor observes another door. (e) The person moves again. Additionally, (b) and (d) depict the observation model $p(z|x)$, the probability of observing a door at different locations in the hallway.

the number of sensor measurements increases over time. To make the computation tractable, Bayes filters assume the dynamic system is Markov—that is, the current state variable x_t contains all relevant information. For locating objects, the Markov assumption implies that sensor measurements depend only on an object's current physical location and that

an object's location at time t depends only on the previous state x_{t-1} . States before x_{t-1} provide no additional information.

Under the Markov assumption, we can efficiently compute the belief in Equation 1 without losing information. Before we provide the update equations, let's briefly examine the recursive Bayes filter update steps using Figure 1. In this

hypothetical one-dimensional scenario, a person walks down a hallway carrying a sensor (such as a camera) that signals if the person walks by a door. The sensor cannot distinguish different doors. We are not suggesting this mobile camera scenario is a viable location system implementation; it simply illustrates important Bayes filters properties.

As Figure 1a illustrates by showing a uniform distribution over possible locations, the person's location is at first unknown. The sensor then sends a "door found" signal. The resulting belief places high probability at places next to doors and low probability elsewhere (see Figure 1b). This distribution possesses three peaks, each corresponding to one of the environment's (indistinguishable) doors. Furthermore, the resulting distribution assigns high probability to three distinct locations, illustrating that the probabilistic framework can handle multiple, conflicting hypotheses that naturally arise in ambiguous situations. Finally, even nondoor locations possess nonzero probability because of the uncertainty inherent in sensing; with a small but nonzero probability, the sensor might err and actually not be next to a door.

Figure 1c shows the person moving and this motion's effect on the belief, assuming the person moves to the right with typical walking speed. The Bayes filter shifts the belief in the direction of motion and smoothes it out to account for inherent uncertainty in motion estimates. Finally, Figures 1d and 1e depict the belief after the sensor observes another door and after the next motion, respectively. Most of the probability mass is placed on a location near one of the doors, and the filter is now quite confident of the person's location.

Here's how to update the Bayes filter. Whenever a sensor provides a new observation z_t , the filter predicts the state according to

$$Bel^-(x_t) \leftarrow \int p(x_t | x_{t-1}) Bel(x_{t-1}) dx_{t-1}. \quad (2)$$

The filter then corrects the predicted estimate using this sensor observation:

$$Bel(x_t) \leftarrow \alpha_t p(z_t | x_t) Bel^-(x_t). \quad (3)$$

Here, $p(x_t | x_{t-1})$ describes the *system dynamics*—that is, how the system's state changes over time.

In location estimation, this conditional probability is the motion model—where the object might be at time t , given that it was previously at location x_{t-1} . This model strongly depends on the information available to the estimation process. It can range from predicting the next position by estimating a person's motion velocity to predicting when a person will exit the elevator by estimating the person's goal.

In the hallway example, the motion update corresponds to the change in belief leading to Figure 1c and 1e. The belief immediately after the prediction and before the observation is the *predictive belief* $Bel^-(x_t)$. The *perceptual model*, $p(z_t | x_t)$, describes the likelihood of making observation z_t given that the person is at location x_t . As Figure 1b and 1d show, the observation update rule increases the probability of locations with high observation likelihoods. For location estimation, the perceptual model is usually considered a given sensor technology's property. It depends on the sensors' types and positions and captures their error characteristics. In Equation 3, α_t is simply a normalizing constant that ensures that the posterior over the entire state space sums up to one. $Bel(x_0)$ is initialized with prior knowledge about the object's location, typically uniformly distributed if no prior knowledge exists.

Bayes filters are an abstract concept in that they provide only a probabilistic framework for recursive state estimation. Implementing Bayes filters requires specifying 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 how they represent probability densities over the state x_t .

We now discuss different belief representations and their properties in the

context of location estimation for pervasive computing.

Kalman filters

Kalman filters are the most widely used variant of Bayes filters.³ They approximate beliefs by their first and second moment, which is virtually identical to a unimodal Gaussian representation:

$$\begin{aligned} Bel(x_t) &\approx \mathcal{N}(x_t; \mu_t, \Sigma_t) \\ &= \frac{1}{(2\pi)^{d/2} |\Sigma_t|^{1/2}} \\ &\quad \exp\left[-\frac{1}{2}(x_t - \mu_t)^T \Sigma_t^{-1} (x_t - \mu_t)\right]. \end{aligned} \quad (4)$$

Here, μ_t is the distribution's mean (first moment) and Σ_t is the $d \times d$ covariance matrix (second moment), where d is the state's dimension. $\mathcal{N}(x_t; \mu_t, \Sigma_t)$ denotes the probability of x_t given a Gaussian with mean μ_t and covariance Σ_t , as Equation 4 specifies. Σ_t represents the uncertainty in the estimate—the larger the covariance, the wider the distribution's spread. Kalman filters are optimal estimators, assuming the initial uncertainty is Gaussian and the observation model and system dynamics are linear functions of the state. Because most systems are not strictly linear, people typically apply extended Kalman filters, which linearize the system using first-order Taylor series expansions.³

Kalman filters' main advantage is their computational efficiency. We can implement both the prediction (Equation 2) and correction (Equation 3) using efficient matrix operations on the mean and covariances. This efficiency, however, comes at the cost of restricted representational power because Kalman filters can represent only unimodal distributions. So, Kalman filters are best if the uncertainty in the state is not too high, which limits them to location tracking using either accurate sensors or sensors

with high update rates. For example, we can apply Kalman filters to the estimation problem shown in Figure 1 only when we know the person's initial location and can limit sensor uncertainty.

Despite Kalman filters' restrictive assumptions, practitioners have applied them with great success to various tracking problems, where the filters yield efficient, accurate estimates, even for some highly nonlinear systems.

Multihypothesis tracking

Multihypothesis tracking can overcome Kalman filters' limitation to unimodal distributions. MHT represents the belief as mixtures of Gaussians:⁴

$$Bel(\mathbf{x}_t) \approx \sum_i w_t^{(i)} \mathcal{N}(\mathbf{x}_t; \mu_t^{(i)}, \Sigma_t^{(i)}). \quad (5)$$

The MHT tracks each Gaussian hypothesis using a Kalman filter. The technique determines the hypotheses' weights $w_t^{(i)}$ on the basis of how well the hypotheses predict the sensor measurements. In other words, at each update, the weights are set proportional to the likelihoods of the sensor measurements, given the individual hypotheses.

Owing to MHT approaches' ability to represent multimodal beliefs, they are more widely applicable than the Kalman filter. For example, in contrast to Kalman filters, we can apply MHT approaches to the problem illustrated in Figure 1. However, MHT techniques are computationally more expensive and require sophisticated techniques or heuristics to determine when to add or delete hypotheses. Because each hypothesis is tracked using a Kalman filter, these methods still rely on the linearity assumptions of Kalman filters. In practice, however, multihypothesis approaches have been robust to violations of these assumptions.

Grid-based approaches

These approaches overcome the restric-

tions imposed on Kalman filters by relying on discrete, piecewise constant representations of the belief. The update equations of discrete approaches are identical to the Bayes filter updates (Equations 2 and 3), with summation replacing integration. For indoor location estimation, grid-based filters tessellate the environment into small patches, typically between 10 cm and 1 m in size. Each grid cell contains the belief that the person or object is in each cell.

A key advantage of these approaches is that they can represent arbitrary dis-

tributions over the discrete state space and can solve estimation problems such as the one in Figure 1. The mobile-robot-localization community has shown that metric approximations provide accurate position estimates in combination with high robustness to sensor noise.⁵ Grid-based approaches' disadvantage is the computational and space complexity required to keep the position grid in memory and to update it for every new observation. Because the complexity grows exponentially with the number of dimensions, we can apply grid-based approaches only to low-dimensional estimation problems, such as estimating a person's position and orientation.

Topological approaches

We can avoid the computational complexity of grid-based methods using non-metric representations of an environment. For instance, many indoor environments provide a natural way to represent a person's location at a symbolic level such as the room or hallway the person is cur-

rently in. Such representations result in topological implementations of Bayes filters, where a graph represents the environment.⁶ Each node in the graph corresponds to a location, and the edges describe the environment's connectivity, typically given by hallways. The motion model $p(\mathbf{x}_t | \mathbf{x}_{t-1})$, which describes where a person walks, can be trained to represent typical motion patterns of individuals moving through the environment.

Topological approaches' advantage is their efficiency, because they represent distributions over small, discrete state

Grid-based approaches' disadvantage is the computational and space complexity required to keep the position grid in memory and to update it for every new observation.

spaces. Their disadvantage is the representation's coarseness. Estimates provide 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

Particle filters represent beliefs by sets of samples, or particles:⁷

$$Bel(\mathbf{x}_t) \approx S_t = \left\{ \left\langle \mathbf{x}_t^{(i)}, w_t^{(i)} \right\rangle \mid i = 1, \dots, n \right\}. \quad (6)$$

Here, each $\mathbf{x}_t^{(i)}$ is a state, and the $w_t^{(i)}$ are nonnegative weights called *importance factors*, which sum up to one. Particle filters realize Bayes filter updates according to a sampling procedure, often called *sequential importance sampling with resampling*. (An overview of particle filters and various applications appears elsewhere.⁷)

Figure 2 illustrates the particle filter algorithm using our one-dimensional

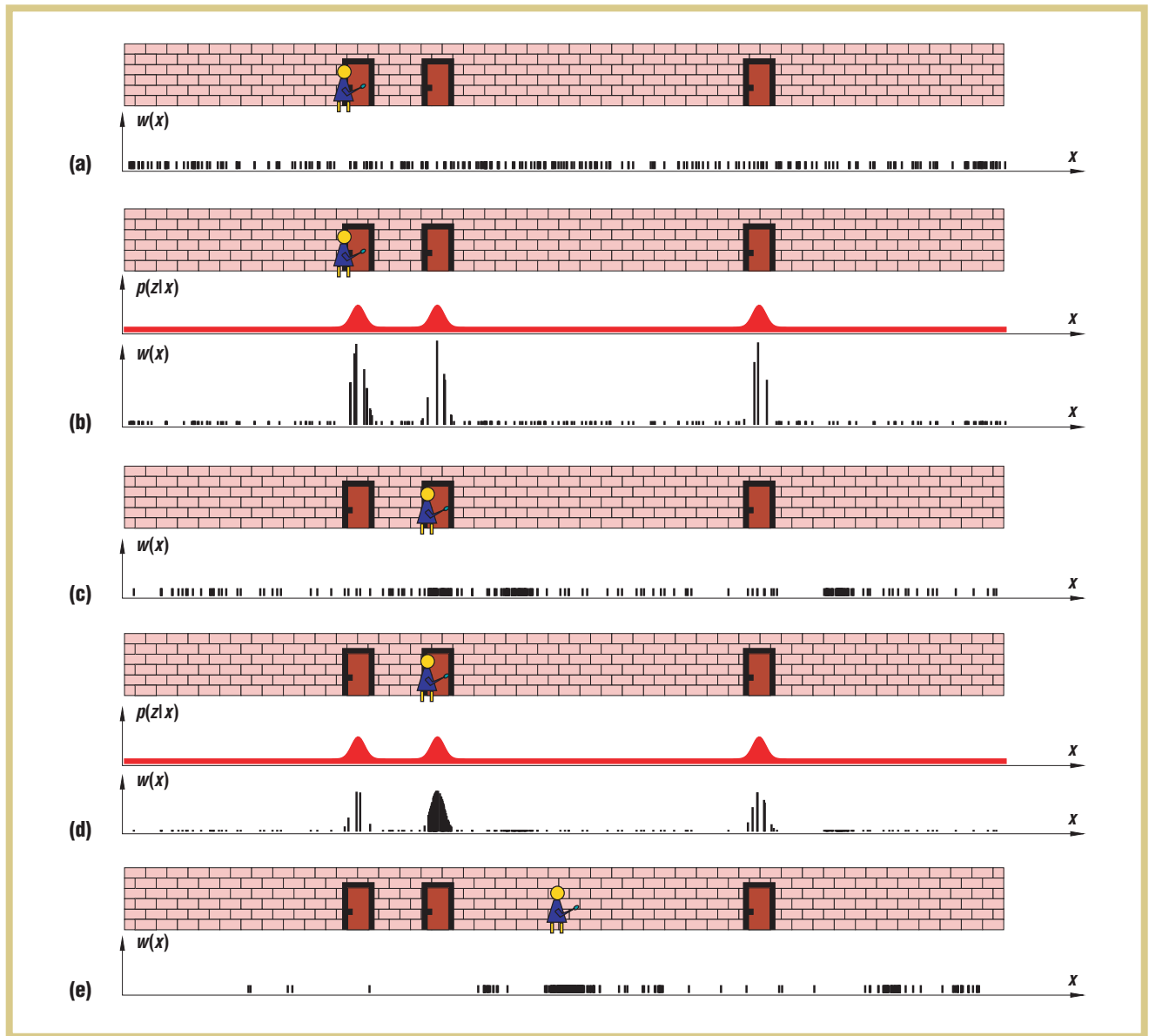


Figure 2. Applying particle filters to location estimation. The black bars depict the particles representing the belief $Bel(x)$. (a) A uniformly distributed sample set presents the person's initially unknown position. (b) A sensor detecting the left door. The sample set is obtained from weighing the importance factors in proportion to the likelihood of the measurement. (c) An implementation of the prediction step. The samples were drawn from the previous set with probability proportional to the importance factors. (d) A sensor detecting a second door. (e) The sample set obtained after another prediction step.

hallway example. A uniformly distributed sample set represents the person's initially unknown position (see Figure 2a). Each sample has the same importance factor $w(x)$, as the equal heights of all bars in Figure 2a indicate. In Figure 2b, the sensor detects the left door. The particle filter incorporates the mea-

surement by adjusting and normalizing each sample's importance factor, leading to the sample set in the lower part of Figure 2b. These samples are the same as before, but now their importance factors are proportional to the observation likelihood $p(z | x)$, shown in Figure 2b.

When the person moves to the right, particle filters randomly draw samples from the current sample set (with probability given by the importance factors). Then the filters use the model to guess the possible succeeding location for each new particle. This update implements the general Bayes filter's prediction step

TABLE 1

Comparing Bayes filters implementations (+, 0, and – represent good, neutral, and weak, respectively).

	Kalman	Multihypothesis tracking	Grid	Topology	Particle
Belief	Unimodal	Multimodal	Discrete	Discrete	Discrete
Accuracy	+	+	0	–	+
Robustness	0	+	+	+	+
Sensor variety	–	–	+	0	+
Efficiency	+	0	–	0	0
Implementation	0	–	0	0	+

(Equation 2). Figure 2c shows the resulting sample set, which differs from the original one in that most samples center around three locations. This concentration of the samples is achieved through sampling proportional to the weights.

In Figure 2d, the sensor detects a second door, leading to the likelihood $p(z | \mathbf{x})$. By weighing the importance factors in proportion to this probability, we obtain the sample set shown in the lower part of Figure 2d. After the next prediction update, most of the probability mass is consistent with the person's true location. So, the result is consistent with the general Bayes filter example in Figure 1.

Particle filters' key advantage is their ability to represent arbitrary probability densities. Furthermore, unlike Kalman filters, particle filters can converge to the true posterior even in non-Gaussian, nonlinear dynamic systems. Compared to grid-based approaches, particle filters are very efficient because they automatically focus their resources (particles) on regions in state space with high probability. Because particle filters' efficiency strongly depends on the number of samples used for filtering, several improvements have been made to more efficiently use the available samples.⁸ However, because these methods' worst-case complexity grows exponentially in the dimensions of the state space, we must be careful when applying particle filters to high-dimensional estimation problems.

Comparison

Table 1 summarizes the advantages and disadvantages of different Bayes filter implementations. Accuracy measures

how well the different approaches can estimate location given adequate sensors. Obviously, grid-based approaches can reach arbitrary accuracy but at prohibitively high computational costs. Kalman filters' limited robustness is due to the unimodal belief representation.

Both Kalman filters and MHT require accurate sensors with rather high update rates. Topological approaches require sensors that relate to an environment's layout. Grid-based approaches and particle filters can incorporate virtually any sensor type. Kalman filters are the most efficient in terms of memory and computation.

Efficiency is a key disadvantage of grid-based methods, although tree-based implementations can reduce this problem.⁵ In general, if accurate sensors are available, Kalman filters might be the best choice. For specific sensors, and if accurate location estimates are not required, topological approaches provide a good way to estimate a person's location. Otherwise, particle filters are an extremely flexible tool with low implementation overhead.

Example applications

Here, we show sensor fusion of two representative pervasive location technologies: infrared proximity badges and ultrasonic time-of-flight tags. Then we present a novel approach to tracking multiple objects that combines the accuracy benefits of anonymous sensors such as scanning infrared laser range finders with the identification certainty of the less accurate infrared and ultrasonic ID sensors. Our hardware is the commercial VersusTech infrared badge system,

MIT's Cricket ultrasound tags, and the LMS200 180-degree scanning laser range finder from SICK, Inc. (Numerous references to the seminal work in sensors for location appears elsewhere.¹) The sensors are deployed throughout the Intel Research laboratory in Seattle.

Sensor models

To apply Bayes filters to location estimation using a specific sensor type, we must first generate a sensor model $p(z | \mathbf{x})$ describing the likelihood of observing a sensor measurement z given the person or object's state \mathbf{x} . Such a model consists of two types of information: a map of the environment and the sensor noise. The problem of constructing maps of indoor environments has received substantial attention in the robotics research community.⁹ Figure 3 illustrates sensor models for ultrasound tags, infrared badges, and laser range finders, respectively.

Figure 3a shows the likelihood of observing a 4.5-meter ultrasound measurement for the different locations in the environment. Because such time-of-flight sensors provide information about the distance between the sensor and the person, the likelihood function is a ring around the sensor's location. The ring's width represents the uncertainty in the measured distance and is often represented by a Gaussian distribution centered at the measured distance. Furthermore, because ultrasound sensors frequently produce measurements that are far from the true distance owing to reflections, all locations in the free space have a nonzero likelihood, as the map's gray areas indicate.

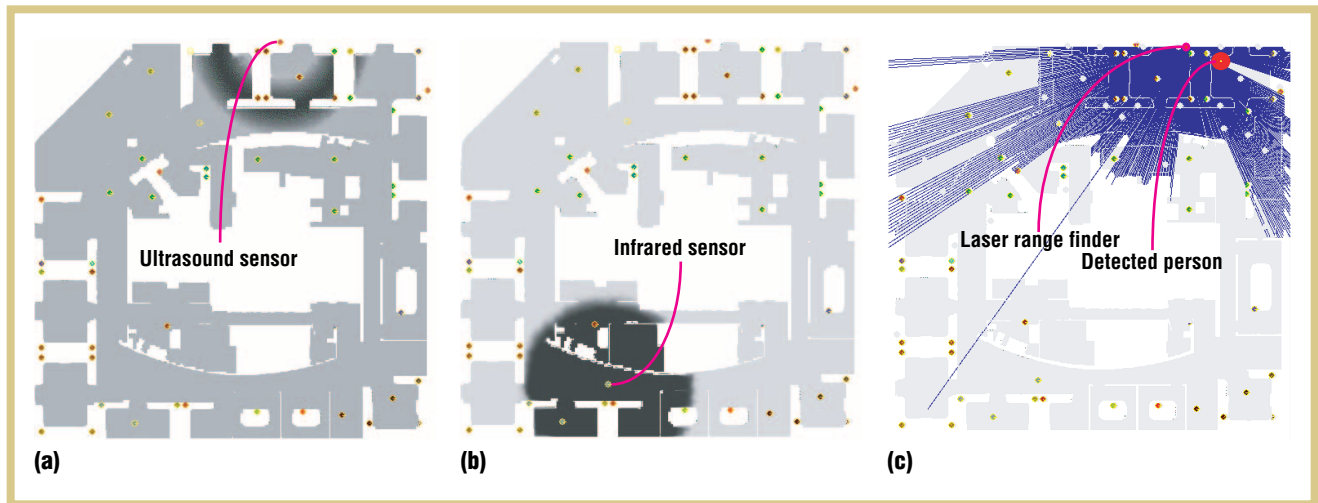


Figure 3. Probabilistic models for (a) ultrasound tags, (b) infrared badges, and (c) laser range finders. The environment is 30 m \times 30 m. In (a) and (b), darker areas indicate higher likelihoods of observing the corresponding measurement. Walls and other obstacles are white because they have zero likelihood. In (c), laser-scan beams, along with a detected person, are blue. The person obstructs the laser beams, thereby causing a “shadow” in the scan.

Figure 3b illustrates the sensor model for the infrared badge system. Such sensors provide no distance information—only information about a person’s presence within a certain range of the receiver. Accordingly, an infrared measurement has a circular likelihood around the sensor location.

Figure 3c shows a scan taken from a laser range finder. Because these sensors are at fixed positions, detecting sensor beams that return atypically short measurements is straightforward. Several adjacent short beams form a shadow region indicating a person’s presence, as Figure 3c shows. A Gaussian distribution usually represents the likelihood for such detected features with small uncertainty.

Multiple inaccurate ID sensors

Here we illustrate how to estimate a person’s location using multiple inaccurate ID sensors, such as the ultrasound and infrared badge systems in our test environment. Owing to these sensors’ high noise level, the belief over the person’s location is typically uncertain and multimodal; so, we chose particle filters for this application. The state vector x of the particles represents the person’s position, orientation, and motion veloc-

ity. Bayes filters can naturally integrate information from different sensors. In this case, whenever an ultrasound or infrared sensor detects the person, the particles are weighted with observation likelihoods such as in Figures 3a and 3b, respectively.

Figure 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 the upper-right corner as the icon indicates (see Figure 4a). Because the system doesn’t know the start location, the samples are spread uniformly throughout the environment’s free space. Figure 4b shows the belief after the person moves out of the cubicles, at which point the samples are spread over different locations owing to the inaccurate sensor information received so far. After an ultrasound sensor detects the person, the filter can estimate the location more accurately (see Figure 4c). The estimates’ certainty continues to change depending on the sensor measurements’ quality.

This example shows that particle filters can extract reasonable location information even from inaccurate sensors by integrating measurements over time. However, we could use particle filters even more

efficiently by constraining the possible locations. More specifically, we could constrain the state space to locations on a Voronoi graph, which is a structure similar to a skeleton of an environment’s free space.¹⁰ The key advantage of such graphs is that they naturally represent typical motion along the environment’s main axes. This technique differs from a topological approach, however, because it maintains high precision in the location estimate—the particles can flow freely along the graph’s edges in contrast to the discrete probabilities stored only at the graph nodes in a topological approach.⁶

Figure 5 shows a sample run using particle filters on Voronoi graphs. The lines in the first frame indicate the Voronoi graph. The sequence is based on data identical to that in Figure 4 using unconstrained particle filters. In experiments we found that such Voronoi graph tracking produces better estimates using far fewer particles. Furthermore, we can use the Voronoi graph structure to learn a person’s high-level motion patterns, such as “person A typically goes into the printer room when she walks down hallway 3.” We can dynamically predict someone’s destination in real time as he or she moves about the environment, which is inval-

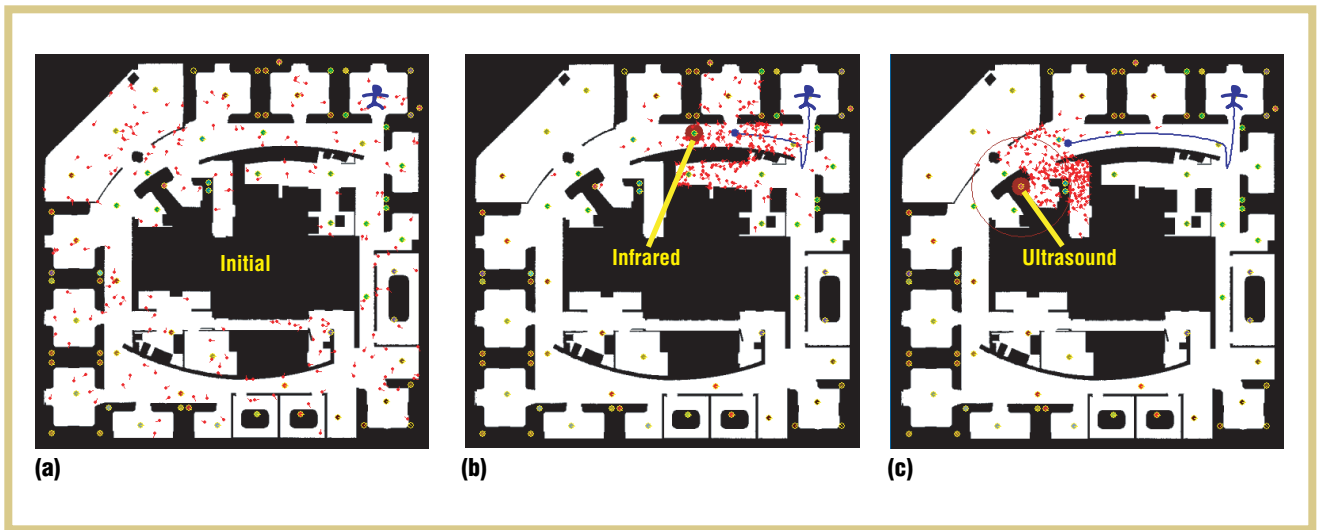


Figure 4. A particle filter for location estimation using infrared and ultrasound sensors: (a) A person wearing an infrared badge and ultrasound tags starts in the upper right corner. (b) After the person moves out of the cubicles, the samples are spread over different locations. (c) An ultrasound sensor detects the person.

able to applications seeking to take predictive action. (More details on such learning approaches appear elsewhere.^{2,10})

Combining anonymous and ID sensors

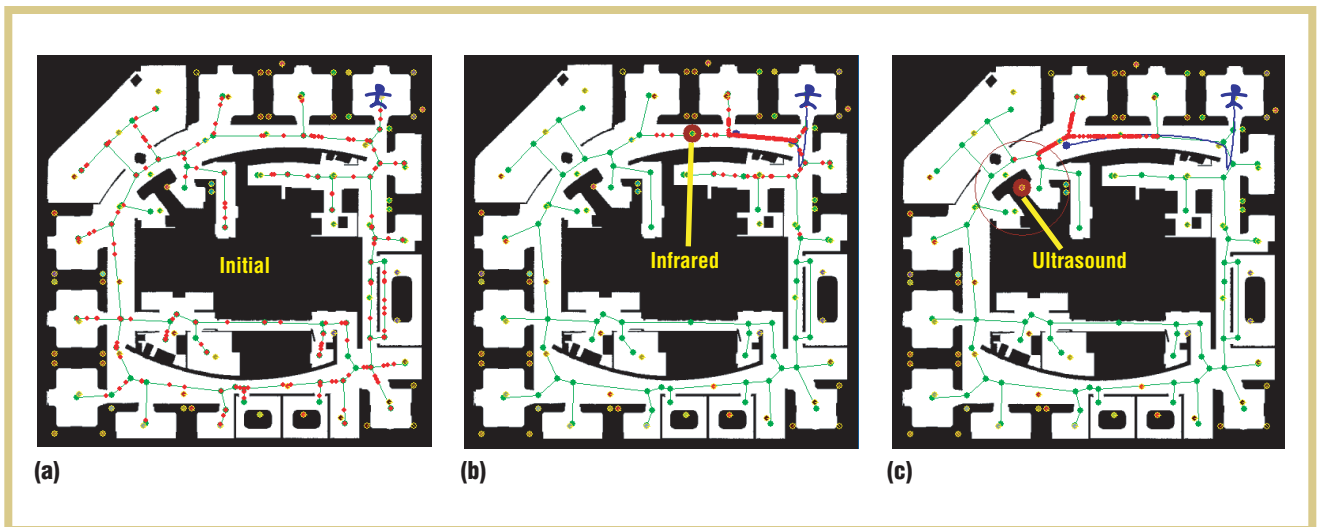
One of the disadvantages of common ID sensors is their limited accuracy. On the other hand, sensors such as laser

range finders provide highly accurate location estimates but do not identify objects. In this example, we describe how to use Bayesian filter techniques to combine information from both types of sensors to get accurate location and ID information.

Figure 6 illustrates the problem of tracking multiple people with anonymous and

ID sensors. The solid and dotted lines are trajectories of persons A and B, respectively. As they walk, the anonymous sensor observes their locations frequently. Initially, their identity is unknown, but they are separated enough to be tracked reliably using the anonymous sensor. Until they reach ID sensor areas 3 and 4, both trajectories have the same probability of belonging

Figure 5. A particle filter constrained to a Voronoi graph structure. The particles move along the graph's edges: (a) The initial location of the person is not known and the particles are spread over the entire graph. (b) After the person moves out of the cubicles, the samples start to concentrate in the upper part of the graph. (c) An ultrasound sensor detects the person.



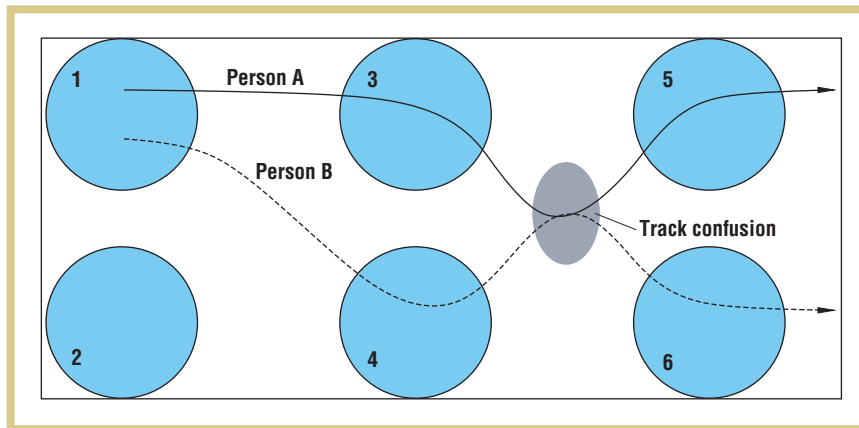


Figure 6. Tracking with anonymous and ID sensors. The blue-shaded circles indicate areas covered by ID sensors such as infrared receivers. Because these sensors provide no information about the person's location in the area, two people in the same area can't be distinguished. Not shown is an additional anonymous sensor, such as a laser range finder, providing accurate location information at a high rate but no information about the persons' IDs.

to either person A or B. Passing through the coverage of ID sensors 3 and 4 resolves the ambiguity and determines the IDs of both trajectories. Then, after the paths cross, confusion exists about the continuation of the two tracks. When the people leave the light-gray area, the anonymous sensor can't determine which trajectory to associate with which person.

The multitarget-tracking community refers to this problem as the *data-association* problem.⁴ Without ID sensors, resolving this ambiguity would be impossible. In our scenario, we can resolve the ambiguity as soon as the people reach the areas covered by ID sensors 5 and 6. To do so, however, requires maintaining the hypotheses for both possible track con-

tinuations—A going down and B going up, or B going down and A going up.

To solve the identity-estimation problem, we use a combination of particle filters and Kalman filters, closely related to MHT. The key idea is to track individual people using Kalman filters, which is possible owing to the accuracy of laser range data (see Figure 3c). A particle filter maintains multiple hypotheses regarding the identities of people, where each particle is one hypothesis for the identity of each tracked person. Put differently, each particle is a collection of Kalman filters annotated with identities.¹¹ The resulting approach can track multiple people and their identities.

Figure 7 shows a small part of the data set used for evaluating the approach. We collected the complete data set with six people, each of whom moved between 230 and 410 meters. In this challenging sensor log, people's paths frequently crossed, and situations existed in which up to four people were occluded to the laser range finders by other people. Nevertheless, our approach successfully estimated each person's location and identity.

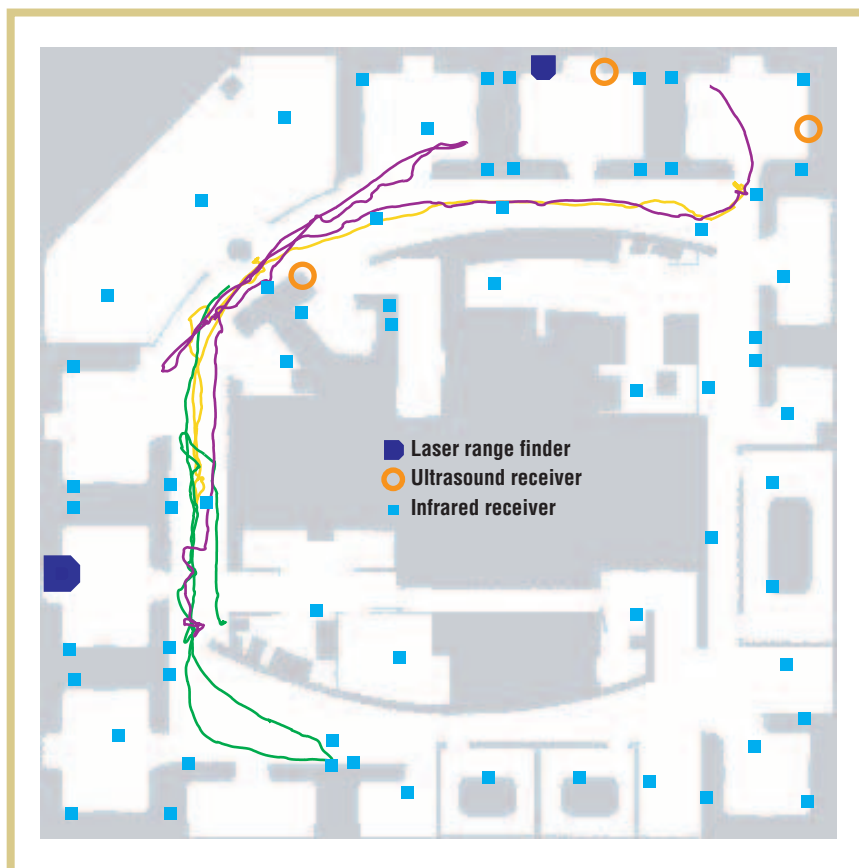


Figure 7. The trajectories of three people moving through the test environment. We successfully tested the combination of particle filters and Kalman filters on more challenging data sets involving six people.

We have codified our sensor fusion techniques in a project called the Location Stack, a general framework for location-aware pervasive computing with a publicly available implementation.¹² We are applying Bayes filters to more complex scenarios, such as learning a person's activities from long-term sensor logs. More complex estimation problems apply structured versions of Bayes filters, such as dynamic Bayesian networks.¹³

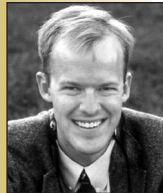
We strongly believe that probabilistic-filter techniques have tremendous potential for inference problems in pervasive computing. Location estimation is just the beginning of connecting Bayesian reasoning systems to raw sensor data. The full power of such techniques lies in their ability to represent uncertainty at different levels of abstractions, thereby enabling truly context-aware pervasive computing systems. ■

REFERENCES

1. J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *Computer*, vol. 34, no. 8, Aug. 2001, pp. 57–66.
2. S.J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed., Prentice Hall, 2002.
3. Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, John Wiley, 2001.
4. Y. Bar-Shalom and X.-R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, Yaakov Bar-Shalom, 1995.
5. D. Fox, "Adapting the Sample Size in Particle Filters through KLD-Sampling," *Int'l J. Robotics Research*, vol. 22, to be published, 2003.
6. J. Krumm, L. Williams, and G. Smith, "SmartMoveX on a Graph: An Inexpensive Active Badge Tracker," *Proc. Int'l Conf. Ubiquitous Computing (UbiComp 02)*, LNCS, Springer-Verlag, 2002, pp. 299–307.
7. A. Doucet, N. de Freitas, and N. Gordon, eds., *Sequential Monte Carlo in Practice*, Springer-Verlag, 2001.
8. D. Fox, W. Burgard, and S. Thrun, "Markov Localization for Mobile Robots in Dynamic Environments," *J. Artificial Intelligence Research*, vol. 11, 1999, pp. 391–427.
9. S. Thrun, "Robotic Mapping: A Survey," *Exploring Artificial Intelligence in the New Millennium*, G. Lakemeyer and B. Nevel, eds., Morgan Kaufmann, 2002.
10. L. Liao et al., "Voronoi Tracking: Location Estimation Using Sparse and Noisy Sensor Data," *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems (IROS)*, IEEE Press, to be published.
11. D. Schulz, D. Fox, and J. Hightower, "Rao-Blackwellised Particle Filters for People Tracking with Anonymous and ID Sensors," *Proc. Int'l Joint Conf. Artificial Intelligence (IJCAI)*, Morgan Kauffman, to be published, 2003.
12. J. Hightower, B. Brumitt, and G. Borriello, "The Location Stack: A Layered Model for Location in Ubiquitous Computing," *Proc. 4th IEEE Workshop Mobile Computing Systems & Applications (WMCSA 2002)*, IEEE CS Press, 2002, pp. 22–28.
13. K. Murphy, *Dynamic Bayesian Networks: Representation, Inference and Learning*, PhD thesis, Computer Science Division, UC Berkeley, 2002.



Dieter Fox is an assistant professor of computer science and engineering at the University of Washington. His research has focused on probabilistic sensor interpretation and state estimation and their application to mobile robotics. He introduced particle filters as a powerful tool for state estimation in robotics. His current research projects include multirobot coordination and human activity recognition. He obtained his PhD from the University of Bonn, Germany, in the area of state estimation in mobile robotics. He received an NSF CAREER award and several best paper awards at major robotics and AI conferences. He is a member of the IEEE and AAAI. Contact him at the University of Washington, Dept. of Computer Science & Engineering Campus Box 352350, Seattle, WA 98195; fox@cs.washington.edu; www.cs.washington.edu/homes/fox.



Jeffrey Hightower is a doctoral candidate at the University of Washington. 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 design abstractions and statistical sensor fusion techniques for location sensing. He received an MS in computer science and engineering from the University of Washington. He is a member of the ACM and the IEEE. Contact him at jeffro@cs.washington.edu; www.cs.washington.edu/homes/jeffro.



Lin Liao is a graduate student in the Department of Computer Science and Engineering at the University of Washington, Seattle. He is interested in probabilistic approaches to artificial intelligence. He received a MS in computer science from the University of Washington, Seattle. Contact him at the University of Washington, Dept. of Computer Science and Engineering, Box 352350, Seattle, WA 98195; liaolin@cs.washington.edu.



Dirk Schulz is a research associate in the Department of Computer Science and Engineering at the University of Washington. His main research interests are in the field of mobile robotics, probabilistic state estimation, object tracking, and Web-controlled mobile robots. He received his PhD in computer science from the University of Bonn in 2002. Contact him at the University of Washington, Dept. of Computer Science and Engineering, Campus Box 352350, Seattle, WA 98195; schulz@cs.washington.edu; www.cs.washington.edu/homes/schulz.



Gaetano Borriello is a professor of computer science and engineering at the University of Washington. He is on partial leave to direct the Intel Research Seattle laboratory, which is engaged in ubiquitous computing research with an aim toward addressing the scalability and usability issues that will be faced when the ratio of computing devices to people increases from 10:1 to over 1000:1. His research interests include location-based systems, sensor-based inferring, and tagging objects with passive and active tags. He serves on the *IEEE Pervasive Computing* editorial board. Contact him at the Dept. of Computer Science & Engineering, University of Washington, Campus Box 352350, Seattle, WA 98195; gaetano@cs.washington.edu; www.cs.washington.edu/homes/gaetano.