AI Magazine Volume 21 Number 4 (2000) (© AAAI)

Probabilistic Algorithms in Robotics

Sebastian Thrun

■ This article describes a methodology for programming robots known as *probabilistic robotics*. The probabilistic paradigm pays tribute to the inherent uncertainty in robot perception, relying on explicit representations of uncertainty when determining what to do. This article surveys some of the progress in the field, using in-depth examples to illustrate some of the nuts and bolts of the basic approach. My central conjecture is that the probabilistic approach to robotics scales better to complex real-world applications than approaches that ignore a robot's uncertainty.

Building autonomous robots is a central objective of research in AI. Over the past decades, researchers in AI have developed a range of methodologies for developing robotic software, ranging from model-based to purely reactive paradigms. More than once, the discussion on what the right way to program robots has been accompanied with speculations concerning the very nature of intelligence as such in animals and people.

One of these approaches, probabilistic robotics, has led to fielded systems with unprecedented levels of autonomy and robustness. Although the roots of this approach can be traced to the early 1960s, in recent years, the probabilistic approach has become the dominant paradigm in a wide array of robotic problems. Probabilistic algorithms have been at the core of a series of fielded autonomous robots, exhibiting an unprecedented level of performance and robustness in the real world. These recent successes can be attributed to at least two developments: (1) the availability of immense computational resources even on low-end PCs and, more importantly, (2) fundamental progress on the basic algorithmic and theoretical levels.

What exactly is the probabilistic approach to robotics? At its core is the idea of representing information through probability densities. In particular, probabilistic ideas can be found in *perception,* that is, the way sensor data are processed, and *action,* that is, the way decisions are made.

Probabilistic perception: Robots are inherently uncertain about the state of their environments. Uncertainty arises from sensor limitations, noise, and the fact that most interesting environments are—to a certain degree—unpredictable. When "guessing" a quantity from sensor data, the probabilistic approach computes a probability distribution over what might be the case in the world, instead of generating a single best guess only. As a result, a probabilistic robot can gracefully recover from errors, handle ambiguities, and integrate sensor data in a consistent way. Moreover, a probabilistic robot knows about its own ignorance—a key prerequisite of truly autonomous robots.

Probabilistic control: Autonomous robots must act in the face of uncertainty, a direct consequence of their inability to know what is the case. When making decisions, probabilistic approaches take the robot's uncertainty into account. Some approaches consider only the robot's current uncertainty; others anticipate future uncertainty. Instead of considering the most likely situations only (current or projected), many probabilistic approaches strive to compute a decision-theoretic optimum, in which decisions are based on all possible contingencies.

These two items are the basic characterization of the probabilistic approach to robotics.

What is the benefit of programming robots probabilistically? My central conjecture is nothing less than the following: A robot that carries a notion of its own uncertainty and that acts accordingly will do better than one that does not. In particular, probabilistic approaches are typically more robust in the face of sensor limitations, sensor noise, and environment dynamics. They often scale much better to complex environments, where the ability to handle uncertainty is of even greater importance. In fact, certain probabilistic algorithms are currently the only known working solutions to hard robotic estimation problems, such as the *kidnapped robot problem* (Engelson and McDermott 1992) in which a mobile robot must recover from localization failure, or the problem of building accurate maps of very large environments in the absence of a global positioning device such as GPS. Additionally, probabilistic algorithms make much weaker requirements on the accuracy of models than many classical planning algorithms, thereby relieving the programmer from the (insurmountable) burden of coming up with accurate models. Viewed probabilistically, the *robot learning problem* is a long-term estimation problem. Thus, probabilistic algorithms provide a sound methodology for many flavors of robot learning. Finally, probabilistic algorithms are broadly applicable to virtually every problem involving perception and action in the real world.

Approached probabilistically, the localization problem is a density estimation problem, where a robot seeks to estimate a posterior distribution over the space of its poses conditioned on the available data.

However, these advantages come at a price. Traditionally, the two most frequently cited limitations of probabilistic algorithms are (1) computational inefficiency and (2) a need to approximate. Certainly, there is some truth to these criticisms. Probabilistic algorithms are inherently less efficient than nonprobabilistic ones because they consider entire probability densities. However, they carry the benefit of increased robustness. The need to approximate arises from the fact that most robot worlds are continuous. Computing exact posterior distributions is typically infeasible because distributions over the continuum possess infinitely many dimensions. Sometimes, one is fortunate in that the uncertainty can be approximated tightly with a compact parametric model (for example, discrete distributions or Gaussians); in other cases, such approximations are too crude, and more complex representations most be used.

None of these limitations, however, pose serious obstacles. Recent research has led to a range of algorithms that are computationally efficient and also highly accurate. To illustrate probabilistic algorithms in practice, this article describes three such algorithms in detail. I argue that the probabilistic paradigm is unique in its ability to solve certain hard robotics problems in uncertain and complex worlds.

Mobile Robot Localization

Let us first take a deeper look into a specific probabilistic algorithm, which solves an important problem in mobile robotics, namely, that of localization. *Localization* is the problem of finding out a robot's coordinates relative to its environment, assuming that one is provided with a map of the environment. Localization is a key component in various successful mobile robot systems (see, for example, Kortenkamp, Bonasso, and Murphy [1998] and Borenstein, Everett, and Feng [1996]). Occasionally, it has been referred to as "the most fundamental problem to providing a mobile robot with autonomous capabilities" (Cox 1991, p. 193). Particularly challenging is the *global localization problem,* where the robot does not know its initial position and therefore has to globally localize itself.

Approached probabilistically, the localization problem is a *density estimation problem,* where a robot seeks to estimate a posterior distribution over the space of its poses conditioned on the available data. The term *pose*, in this article, refers to a robot's *x*-*y* coordinates together with its heading direction θ. Denoting the robot's pose at time t by s_t and the data leading to time t by $d_{0\ldots t}$, the posterior is conveniently written as

$$p\left(s_t \middle| d_{0\ldots t}, m\right) \tag{1}$$

Here, m is the model of the world (for example, a map). For brevity, I denote this posterior $b_t(s_t)$ and refer to it as the robot's *belief state* at time t.

Sensor data typically come in two flavors: First are data that characterize the momentary situation (for example, camera images, laser range scans), and second are data relating to a change in the situation (for example, motor controls or odometer readings). Referring to the first form as observations and the second form as action data, let us without loss of generality assume that both types of data arrive in an alternating sequence:

$$d_{0\ldots t} = o_0, a_0, o_1, a_1, \ldots, a_{t-1}, o_t \tag{2}$$

Here o_t denotes the observation, and a_t denotes the action data item collected at time t.

To estimate the desired posterior $p(s_t | d_{0\ldots t}, m)$, probabilistic approaches frequently resort to a Markov assumption, which states that the past is independent of the future given knowledge of the current state. The Markov assumption is often referred to as the *static world assumption* because it assumes the robot's pose is the only state in the world that would impact more than just one isolated sensor reading. Practical experience suggests, however, that probabilistic algorithms are robust to mild violations of the Markov assumption, and extensions exist that go beyond this assumption (for example, Fox et al. [1998]).

The desired posterior is now computed using a recursive formula, which is obtained by applying Bayes's rule and the theorem of total probability to the original expression, exploiting the Markov assumption twice. See figure 1. Here, η_t is a constant normalizer that ensures that the result sums up to 1. Within the context of mobile robot localization, the result of this transformation

$$b_t(s_t) = \eta_t\, p(o_t|s_t,m) \int p(s_t|a_{t-1},s_{t-1},m)\, b_{t-1}(s_{t-1})\,ds_{t-1} \tag{3}$$

is often referred to as *Markov localization* (Fox, Burgard, and Thrun 1999; Burgard et al. 1996; Kaelbling, Cassandra, and Kurien 1996; Koenig and Simmons 1996; Simmons and Koenig 1995), but it equally represents the basic updated equation in Kalman (1960) filters, Hidden Markov models (Rabiner and Juang 1986), and dynamic belief networks (Russell and Norvig 1995; Dean and Kanazawa 1989). The Kalman (1960) filter, which is historically the most popular approach for position tracking, represents beliefs by Gaussians. The vanilla Kalman filter also assumes Gaussian noise and linear motion equations; however, extensions exist that relax some of these assumptions (Maybeck 1990; Jazwindsky 1970). Kalman filters have been applied with great success to a range of tracking and mapping problems in robotics (Leonard, Durrant-Whyte, and Cox 1992; Smith, Self, and Cheeseman 1990), although they tend to not work well for global localization or the kidnapped robot problem. Markov localization using discrete, topological representations for b were pioneered (among others) by Simmons and Koenig (1995), whose mobile robot XAVIER traveled more than 230 kilometers through Carnegie Mellon University's (CMU) hallways over a period of several years (Simmons et al. 1997).

To implement equation 3, one needs to specify $p(s_t | a_{t-1}, s_{t-1}, m)$ and $p(o_t | s_t, m)$. Both densities are usually *time invariant;* that is, they do not depend on t, so the time index can be omitted. The first density characterizes the effect of the robot's actions a on its pose and can therefore be viewed as a probabilistic generalization of mobile robot kinematics; see figure 2 for examples. The other density, $p(o | s, m)$, is a probabilistic model of perception. Figure 3 illustrates a sensor model for range finders, which uses ray tracing and a mixture of four parametric densities to calculate $p(o | s, m)$. In most implementations, both of these probabilistic models are quite crude, using uncertainty to account for model limitations.

Figure 4 illustrates global mobile robot localization based on sonar measurements in an office environment. The robot's path is outlined in the first diagram along with four refer-

$$\begin{aligned}
b_t(s_t) &= p(s_t|o_0,\ldots,a_{t-1},o_t,m)\\
&\overset{\text{Bayes}}{=} \eta_t\, p(o_t|o_0,\ldots,a_{t-1},s_t,m)\, p(s_t|o_0,\ldots,a_{t-1},m)\\
&\overset{\text{Markov}}{=} \eta_t\, p(o_t|s_t,m)\, p(s_t|o_0,\ldots,a_{t-1},m)\\
&\overset{\text{Tot. Prob}}{=} \eta_t\, p(o_t|s_t,m)\int p(s_t|o_0,\ldots,a_{t-1},s_{t-1},m)\, p(s_{t-1}|o_0,\ldots,a_{t-1},m)\,ds_{t-1}\\
&\overset{\text{Markov}}{=} \eta_t\, p(o_t|s_t,m)\int p(s_t|a_{t-1},s_{t-1},m)\, p(s_{t-1}|o_0,\ldots,o_{t-1},m)\,ds_{t-1}\\
&= \eta_t\, p(o_t|s_t,m)\int p(s_t|a_{t-1},s_{t-1},m)\, b_{t-1}(s_{t-1})\,ds_{t-1}
\end{aligned}$$

Figure 1. Derivation of Bayes's Filters.

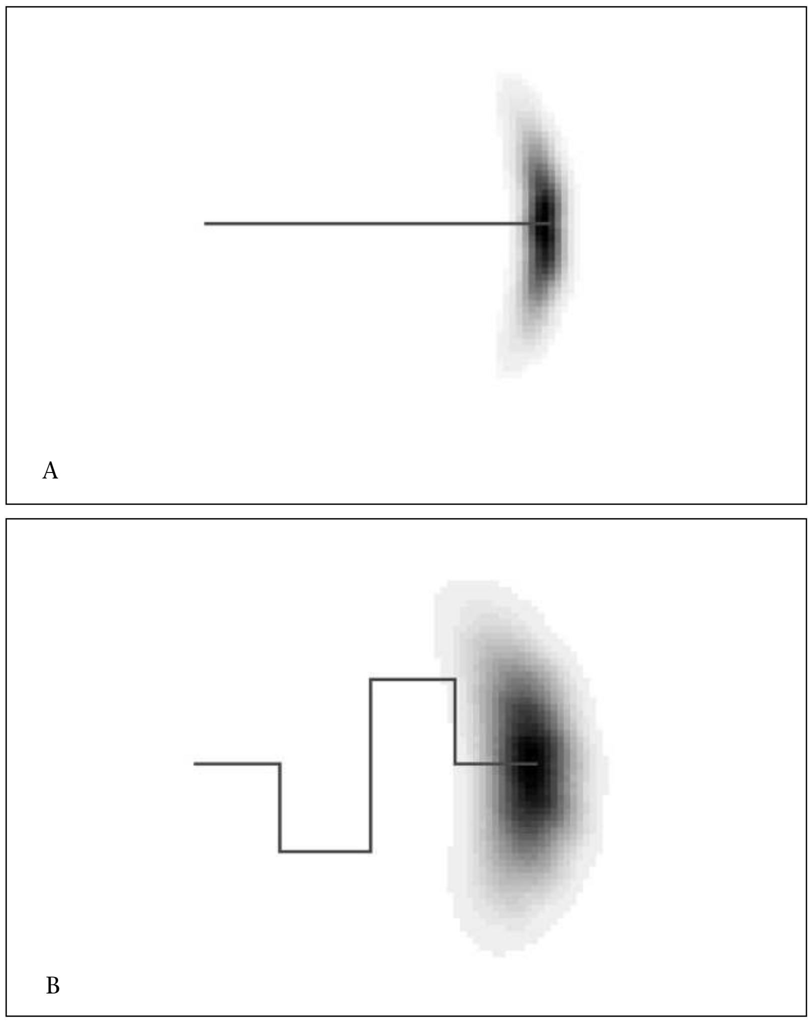
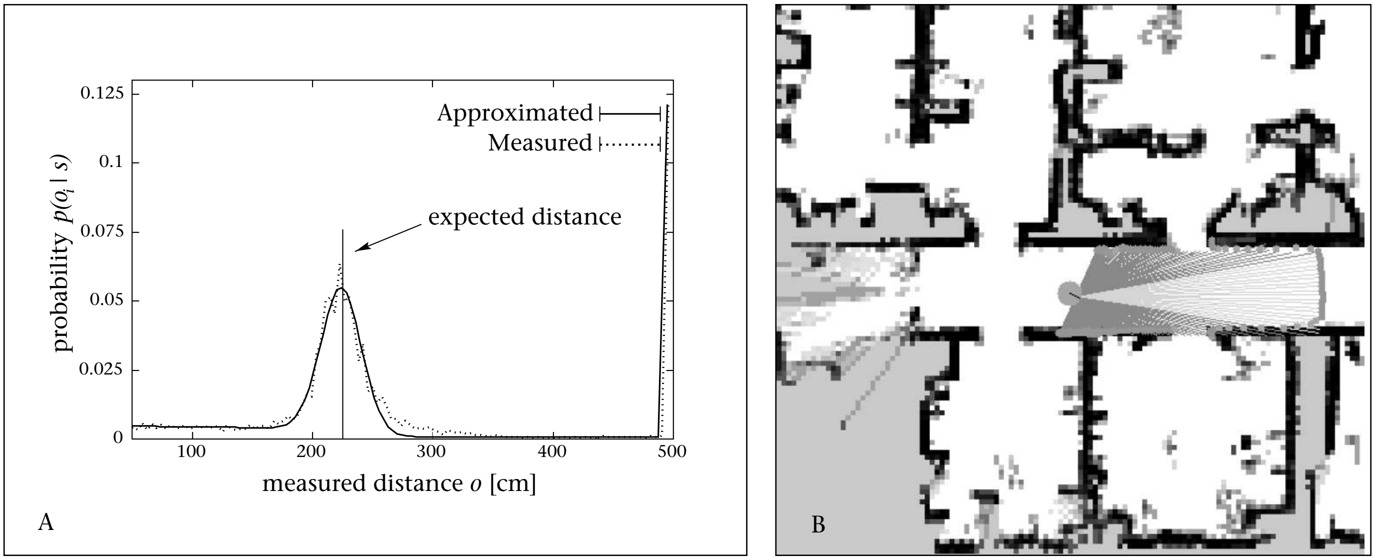


Figure 2. Probabilistic Generalization of Mobile Robot Kinematics.

Each dark line illustrates a commanded robot path, and the shaded area shows the posterior distribution of the robot's pose; the darker an area, the more likely it is the corresponding pose. A. This path is 40 meters long. B. This path is 80 meters long.



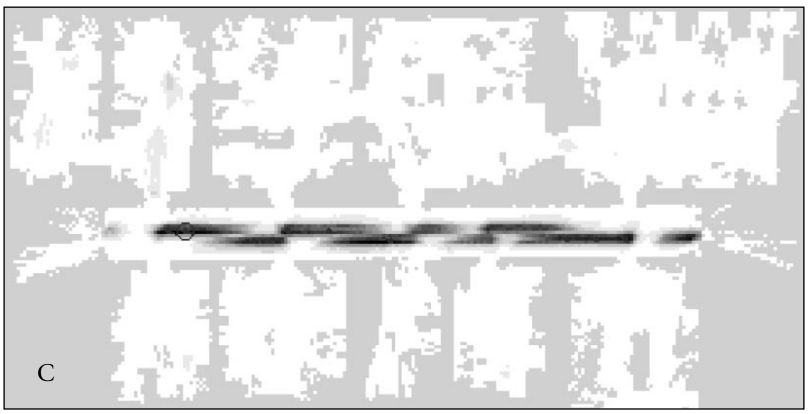


Figure 3. Probabilistic Sensor Model for Laser Range Finders.

A. The density $p(o \mid s, m)$ relates the actual, measured distance of a sensor beam to its expected distance computed by ray tracing, under the assumption that the robot's pose is s. A comparison of actual data and our (learned) mixture model shows good correspondence. B. This diagram shows a specific laser range scan o. C. This diagram plots the density $p(o \mid s, m)$ for different locations in the map.

ence locations. Also shown is the initial belief, which is uniform, because the robot does not know where it is. The posterior belief after moving from the first to the second reference location is shown in figure 4b. At this point, most of the probability mass is located in the corridor, but the robot still does not know where it is. This diagram nicely illustrates one of the features of the probabilistic approach, namely, its ability to pursue multiple hypotheses, weighted by sensor evidence. After moving to the third reference position, the belief is centered around two discrete locations, as shown in figure 4c. Finally, after moving into one of the rooms, the symmetry is broken, and the robot is highly certain about where it is (figure 4d).

Of fundamental importance for the design of probabilistic algorithms is the choice of the representation. One of the most powerful approximations is known as *particle filters* (Doucet, Gordon, and deFreitas 2000; Pitt and Shepherd 1999; Doucet 1998; Liu and Chen 1998), *condensation algorithm* (Isard and Blake 1998, 1997), and Monte Carlo localization (Dellaert et al. 1999; Fox et al. 1999); here, I refer to it as Monte Carlo localization (MCL). The basic idea of MCL is to approximate with a weighted set of samples (particles) so that the discrete distribution defined by the samples approximates the desired one. The weighting factors are called *importance factors* (Rubin 1998). The initial belief is represented by a uniform sample of size m, that is, a set of m samples drawn uniformly from the space of all poses, annotated by the constant importance factor m^{-1}. MCL implements the update equation (3) by constructing a new sample set from the current one in response to an action item a_{t-1} and an observation o_t:

First, draw a random sample from the current belief $b_{t-1}(s_{t-1})$ with probability given by the importance factors of the belief $b_{t-1}(s_{t-1})$.

Second, for this s_{t-1}, randomly draw a successor pose s_t according to the distribution $p(s_t \mid a_{t-1}, s_{t-1}, m)$.

Third, assign the (unnormalized) importance factor $p(o_t \mid s_t, m)$ to this sample and add it to the new sample set representing $b_t(s_t)$.

Repeat steps 1 through 3 m times. Finally, normalize the importance factors in the new sample set so that they add to 1.

Figure 5 shows MCL in action. Shown in the figure 5a is a belief distribution (sample set) at the beginning of the experiment when the robot does not (yet) know its position. Each dot is a three-dimensional sample of the robot's x-y location along with its heading

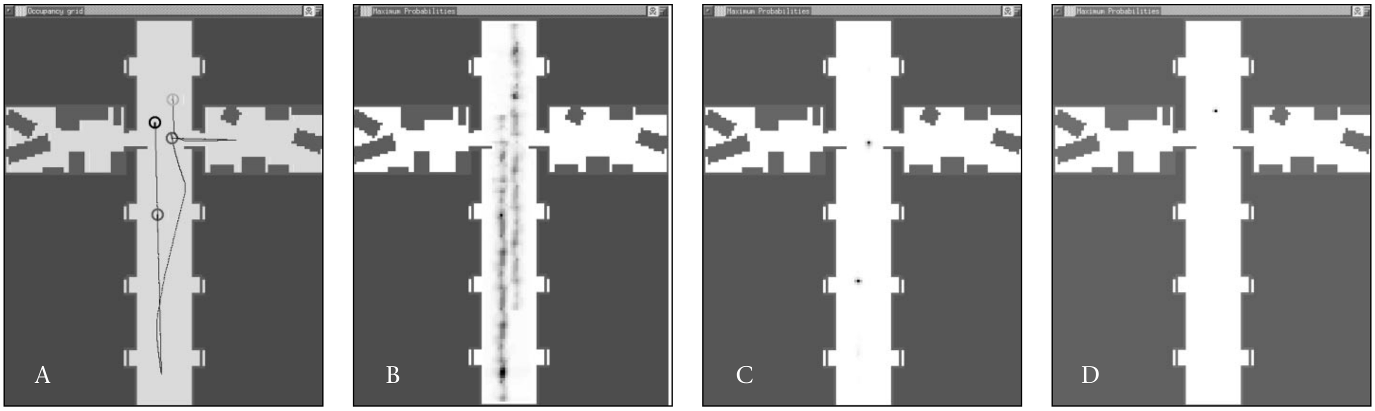


Figure 4. Grid-Based Markov Localization.

direction θ. Figure 5b shows the belief after a short motion segment, incorporating several sonar readings. At this point, most samples concentrate on two locations; however, the symmetry of the corridor makes it impossible to disambiguate them. Finally, figure 5c shows the belief after the robot moves into one of the rooms, enabling it to disambiguate its location with high confidence.

The MCL algorithm is, in fact, quite efficient; slight modifications of the basic algorithms (Lenser and Veloso 2000; Thrun, Fox, and Burgard 2000) require as few as 100 samples for reliable localization, consuming only a small fraction of time available on a low-end PC. It can also be implemented as an any-time algorithm (Zilberstein and Russell 1995; Dean and Boddy 1988), meaning that it can adapt to the available computational resources by dynamically adjusting the number of samples m. With slight modifications, such as sampling from the observation (Thrun, Fox, and Burgard 2000), MCL has been shown to recover gracefully from global localization failures, such as manifested in the *kidnapped robot problem* (Engelson 1994), where a well-localized robot is teleported to some random location without being told. For these reasons, probabilistic algorithms such as MCL are currently the best-known methods for such hard localization problems.

Another feature of MCL is that its models, in particular $p(s' \mid a, s, m)$, $p(o \mid s, m)$ and the map, can be extremely crude and simplistic because probabilistic models carry their own notion of uncertainty, thus making them relatively easy to code. In comparison, traditional robotics algorithms that rely on deterministic models make much stronger demands on the accuracy of the underlying models.

$$b_t(s_t, m_t) = \eta'_t \, p(o_t \mid s_t, m_t) \iint p(s_t, m_t \mid a_{t-1}, s_{t-1}, m_{t-1}) \, b_{t-1}(s_{t-1}, m_{t-1}) \, ds_{t-1} \, dm_{t-1} \quad (4)$$

$$b_t(s_t, m) = \eta'_t \, p(o_t \mid s_t, m) \int p(s_t \mid a_{t-1}, s_{t-1}, m) \, b_{t-1}(s_{t-1}, m) \, ds_{t-1} \quad (5)$$

Equations 4 and 5.

Mapping

A second area of robotics where probabilistic algorithms have proven remarkably successful is mapping. *Mapping* is the problem of generating maps from sensor measurements. This estimation problem is much higher dimensionally than the robot localization problem; in fact, in its pure form, one could argue the problem possesses infinitely many dimensions. What makes this problem particularly difficult is its chicken-and-egg nature, which arises from the fact that position errors accrued during mapping are difficult to compensate (Rencken 1993). Put differently, localization with a map is relatively easy, as is mapping with known locations. In combination, however, this problem is hard.

In this section, I review three major paradigms in mobile robot mapping, all of which are probabilistic and follow from the same mathematical framework. Let us begin with the most obvious idea, which is using the same approach for mapping as for localization. If we augment the state s that is being estimated by the map—the subscript t indicates that we allow the map to change over time—equation 3 becomes equation 4 (see above).If the map is assumed to be static, which is common in the literature, the maps at times t and t–1 will be equivalent, implying that $p(s_t, m_t \mid a_{t-1}, s_{t-1}, m_{t-1})$ is zero if $m_t \neq m_{t-1}$ and $p(s_t \mid a_{t-1}, s_{t-1}, m_{t-1})$ if $m_t = m_{t-1}$. The integration over maps in equation 4 is therefore eliminated, yielding equation 5 (see above). The major problem with

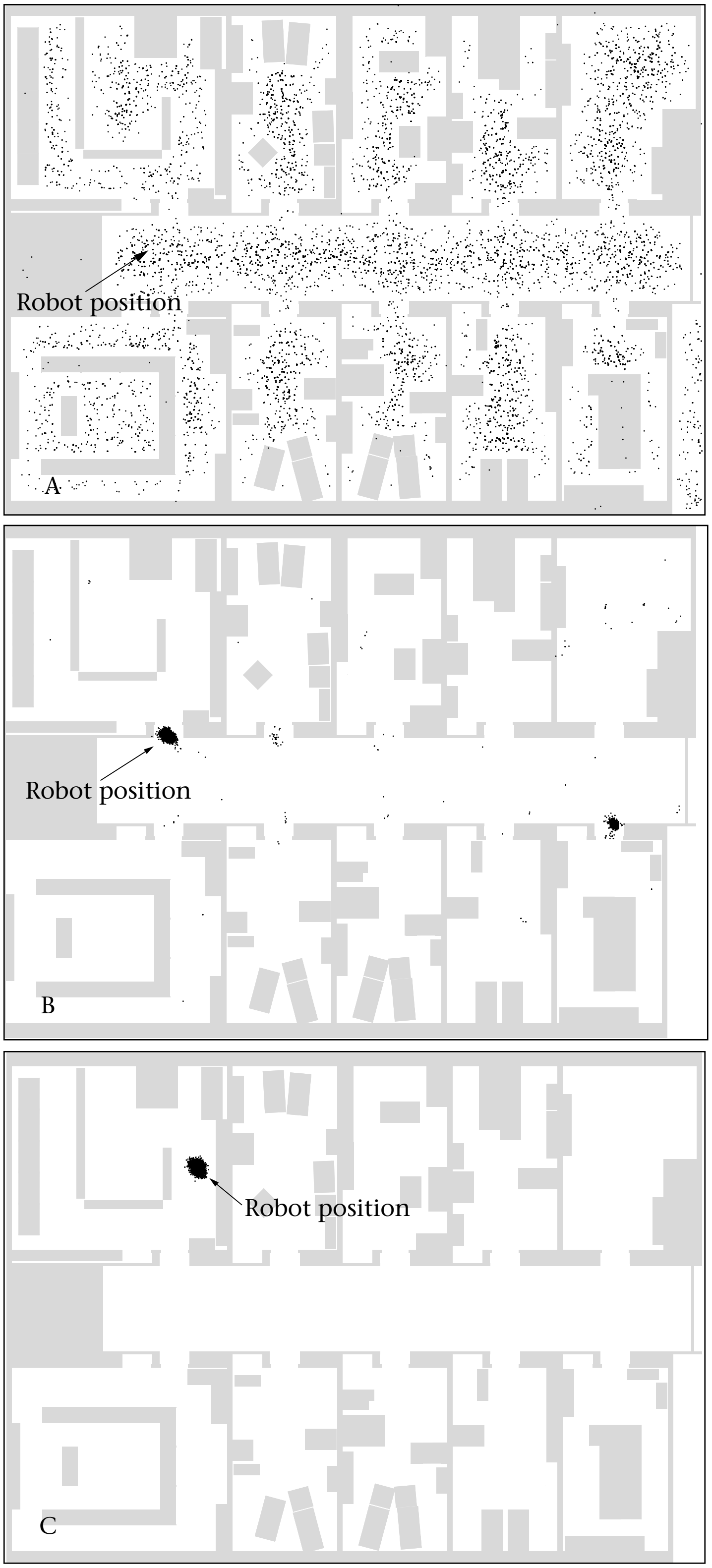


Figure 5. Global Localization of a Mobile Robot Using MCL.
A. Initial belief. B. Intermediate belief. C. Final belief.

equation 5 is complexity. The belief $b_t(s_t, m)$ is a density function in an $(N + 3)$-dimensional space, where N is the number of free parameters that constitute a map (for example, a constant times the number of landmarks), and the additional three parameters specify the robot's pose. N can be very large (for example, 1000), which makes the posterior-estimation problem hard. To make matters worse, the belief $b_t(s_t, m)$ cannot easily be factorized because the uncertainty of map items and robot poses are often highly correlated (Smith, Self, and Cheeseman 1990).

The most successful attempt to implement equation 5 uses Kalman filters (Castellanos and Tardós 2000; Castellanos et al. 1999; Moutarlier and Chatila 1989a, 1989b; Leonard and Durrant-Whyte 1992; Leonard, Durrant-Whyte, and Cox 1992), which goes back to a seminal paper by Smith, Self, and Cheeseman (1990). Recall that Kalman filters represent beliefs by Gaussians; thus, they require $O(N^2)$ parameters to represent the posterior over an N-dimensional space. Calculating equation 5 involves matrix multiplication, which can be done in $O(N^2)$ time (Maybeck 1990). Thus, the number of features that can be mapped are critically limited (see Leonard and Feder [1999] for a recent attempt to escape this limitation using hierarchies of maps). In practice, this approach has been applied to mapping several hundreds of free parameters (Leonard and Durrant-Whyte 1992).

The basic Kalman filtering approach to mapping is also limited in a second, more important way. In particular, it requires that features in the environment can uniquely be identified, which is a consequence of the Gaussian noise assumption. For example, it does not suffice to know that the robot faces a door; instead, it must know which door it faces to establish correspondence to previous sightings of the same door. This limitation is of great practical importance. It is common practice to extract a small number of identifiable features from the sensor data at the risk of discarding all other information. Some recent approaches overcome this assumption by "guessing" the correspondence between measurements at different points in time, but they tend to be brittle if these guesses are wrong (Gutmann and Nebel 1997; Lu and Milios 1997). However, if the assumptions are met, Kalman filters generate optimal estimates, and in particular, they outperform any nonprobabilistic approach.

An alternative approach, proposed in Thrun, Fox, and Burgard (1998), seeks to estimate the mode of the posterior, $\text{argmax}_m\, b(m)$, instead of the full posterior $b(m)$. This goal might appear quite modest compared to the full posterior estimation. However, if the correspondence is unknown (and noise is non-Gaussian), this problem in itself is challenging. To see, note that the posterior over maps can be obtained in closed form (see equation 6) where the initial pose is, somewhat arbitrarily, set to $S_0 = \langle 0, 0, 0 \rangle$. This expression is obtained from equation 5 by integrating over s_t, followed by recursively substituting the belief from time t–1 to time 0, and resorting of the resulting terms and integrals. For convenience, we assume a uniform prior $p(m)$, transforming the problem into a maximum-likelihood estimation problem. Notice that equation 6 integrates over all possible paths, a rather complex integration. Unfortunately, I know of no way to calculate $\text{argmax}_m\, b_t(m)$ analytically for data sets of reasonable size.

To find a solution, we notice that the robot's path can be considered "missing variables" in the optimization problem; knowing them indeed greatly simplifies the problem. The statistical literature shows a range of algorithms for such problems, one of which is the EM algorithm (McLachlan and Krishnan 1997; Dempster, Laird, and Rubin 1977). In the context of mapping, this algorithm computes a sequence of maps, denoted $m^{[0]}$, $m^{[1]}$, …, with successively increasing likelihood. The superscript *[k]* is not to be confused with the time index t or the index of a particle i; all it refers to is the iteration of the optimization algorithm.

EM calculates a new map by iterating two steps: (1) an *expectation step*, or E-step, and (2) a *maximization step*, or M-step:

In the E-step, EM calculates an expectation of a joint log-likelihood function of the data and the poses, conditioned on the k-th map $m^{[k]}$ (and conditioned on the data) (see equation 7 above). This might appear a bit cryptic, but the key thing here is that computing Q involves calculating the posterior distribution over poses $s_o, \ldots, s_t$ conditioned on the k-th model $m^{[k]}$. We have already seen how to estimate the posterior over poses given a map. Technically, calculating equation 7 involves two localization runs through the data, a forward run and a backward run, because all the data have to be taken into account when computing the posterior $p(s_t \mid d_{0\ldots t})$ (the earlier algorithm only considers data to time t). Also note that in the first iteration, we do not have a map. Thus, $Q[m \mid m^{[k]}]$ calculates the posterior for a *blind robot,* that is, a robot that ignores its measurements $o_1, \ldots, o_t$.

In the M-step, the most likely map is computed given the pose estimates obtained in the E-step, which is formally written as

$$b_t(m) = p(m \mid d_{0\ldots t}) = \int b_t(s_t, m)\, ds_t$$
$$= \eta_t'' p(m) \iint \ldots \int \prod_{\tau=0}^{t} p(o_\tau \mid s_\tau, m) \prod_{\tau=1}^{t} p(s_\tau \mid a_{\tau-1}, s_{\tau-1}, m)\, ds_1\, ds_2 \ldots ds_t \quad (6)$$

$$Q\left[m \mid m^{[k]}\right] = E_{m^{[k]}}\left[\log p\left(s_0, \ldots, s_t, d_{0\ldots t} \mid m^{[k]}\right) \mid d_{0\ldots t}\right] \quad (7)$$

Equations 6 and 7.

$$m^{[k+1]} = \underset{m}{\text{argmax}}\, Q\left[m \mid m^{[k]}\right] \quad (8)$$

Technically, this problem is still very difficult because it involves finding the optimum in a high-dimensional space. However, it is common practice to decompose the problem into a collection of one-dimensional maximization problems by stipulating a grid over the map and solving equation 8 independently for each grid cell. The maximum-likelihood estimation for the resulting single-cell random variables is mathematically straightforward.

Iterations of both steps tend to increase the log-likelihood (currently, a proof of convergence is lacking because of the decomposition in the M-step). However, this approach works very well in practice (Thrun, Fox, and Burgard 2000), solving hard mapping problems that were previously unsolved (see also Shatkay [1998] and Shatkay and Kaelbling [1997]).

The decomposition in the M-step is quite common for mapping algorithms that assume knowledge of the robot's pose. It goes back to the seminal work by Elfes and Moravec on *occupancy grid mapping* (Elfes 1989; Moravec 1988), a probabilistic algorithm that is similar, though not identical, to the M-step, which brings us to the third mapping algorithm. Occupancy grid mapping is currently the most widely used mapping algorithm for mobile robots (Thrun 1998; Guzzoni et al. 1997; Yamauchi and Langley 1997; Borenstein 1987; Elfes 1987), often augmented by ad hoc methods for localization during mapping. It is another prime example of the success of probabilistic algorithms in robotics.

Occupancy grid mapping addresses a much simpler problem than the previous one, namely, estimating a map from a set of sensor measurements given that one already knows the corresponding poses. Let $\langle x, y \rangle$ denote a specific grid cell and

$$m_t^{\langle xy \rangle}$$

be the random variable that models its occu-

$$b_t\left(m_t^{\langle xy\rangle}\right) = \eta_t\, p\left(o_t \middle| m_t^{\langle xy\rangle}\right) \sum_{m_t^{\langle xy\rangle}=0}^{1} p\left(m^{\langle xy\rangle} \middle| a_{t-1}, m_{t-1}^{\langle xy\rangle}\right) b_{t-1}\left(m_{t-1}^{\langle xy\rangle}\right) \tag{9}$$

$$b_t\left(m^{\langle xy\rangle}\right) = \eta_t\, p\left(o_t \middle| m^{\langle xy\rangle}\right) b_{t-1}\left(m^{\langle xy\rangle}\right) = \eta_t \frac{p\left(m^{\langle xy\rangle} \middle| o_t\right) p\left(o_t\right)}{p\left(m^{\langle xy\rangle}\right)} b_{t-1}\left(m^{\langle xy\rangle}\right) \tag{10}$$

$$\frac{b_t\left(m^{\langle xy\rangle}=1\right)}{b_t\left(m^{\langle xy\rangle}=0\right)} = \frac{p\left(m^{\langle xy\rangle}=1 \middle| o_t\right)}{p\left(m^{\langle xy\rangle}=0 \middle| o_t\right)} \frac{p\left(m^{\langle xy\rangle}=0\right)}{p\left(m^{\langle xy\rangle}=1\right)} \frac{b_{t-1}\left(m^{\langle xy\rangle}=1\right)}{b_{t-1}\left(m^{\langle xy\rangle}=0\right)} \tag{11}$$

$$b_t\left(m^{\langle xy\rangle}=1\right) = 1 - \left\{1 + \frac{p\left(m^{\langle xy\rangle}=1\right)}{1-p\left(m^{\langle xy\rangle}=1\right)} \left[\prod_{\tau=0}^{t} \frac{p\left(m^{\langle xy\rangle}=1 \middle| o_\tau\right)}{1-p\left(m^{\langle xy\rangle}=1 \middle| o_\tau\right)} \frac{1-p\left(m^{\langle xy\rangle}=1\right)}{p\left(m^{\langle xy\rangle}=1\right)}\right]\right\}^{-1} \tag{12}$$

Equations 9, 10, 11, and 12.

pancy at time *t*. Occupancy is a binary concept; thus, we write

$$m_t^{\langle xy\rangle} = 1$$

if a cell is occupied, and

$$m_t^{\langle xy\rangle} = 0$$

if it is not. Substituting

$$m_t^{\langle xy\rangle}$$

into equation 3 under the consideration that occupancy is a binary random variable yields equation 9 (see above) which in static worlds can be simplified to equation 10 (see above) The second transformation pays tribute to the fact that in occupancy grid mapping, one often is given $p(m^{<xy>} \mid o_t)$ instead of $p(o_t \mid m^{<xy>})$. One could certainly leave it at this and calculate the normalization factor η_t at run time. However, for a binary random variable, the normalizer can be eliminated by noticing the so-called *odds*, which are the quotient in equation 11 (see above): As is easily shown, this expression has the closed-form solution shown in equation 12 above.

All three of these algorithms have shown to be highly robust and accurate in practice, and they are among the best algorithms in existence. For example, figure 6a shows a raw data set of a large hall (approximately 50 meters wide) as well as the result of first applying the EM algorithm and then occupancy grid mapping using the poses estimated with EM (figure 6b). The map in figure 6c has been generated using a similar probabilistic algorithm that runs online (unlike EM) (see also Gutman and Konolige [2000]); figure 6d shows an architectural blueprint for comparison. Cyclic environments are among the most difficult ones to map because the odometry error can be very large when closing the cycle. These results illustrate that EM and occupancy grid mapping yield excellent results in practice. Although the maps shown here are two dimensional, probabilistic algorithms have also successfully been applied to build three-dimensional maps (Thrun, Burgard, and Fox 2000).

These results illustrate that probabilistic algorithms are well suited for high-dimensional estimation and learning problems; in fact, I know of no comparable algorithm that can solve problems of equal hardness that does not explicitly address the inherent uncertainty in perception. To date, the best mapping algorithms are probabilistic, and most of them are versions of the three algorithms described here. My analysis also suggests that probabilistic algorithms are somewhat of a natural fit for problems such as those studied here. Past research has shown that many estimation and learning problems in robotics have straightforward solutions when expressed using the language of probability theory, with mapping being just one example.

Robot Control

Finally, let us turn our attention to the issue of robot control. The ultimate goal of robotics is to build robots that do the right thing. As stated in the introduction, I conjecture that a robot that takes its own uncertainty into account when selecting actions will be superior to one that does not.

Unfortunately, the field of probabilistic

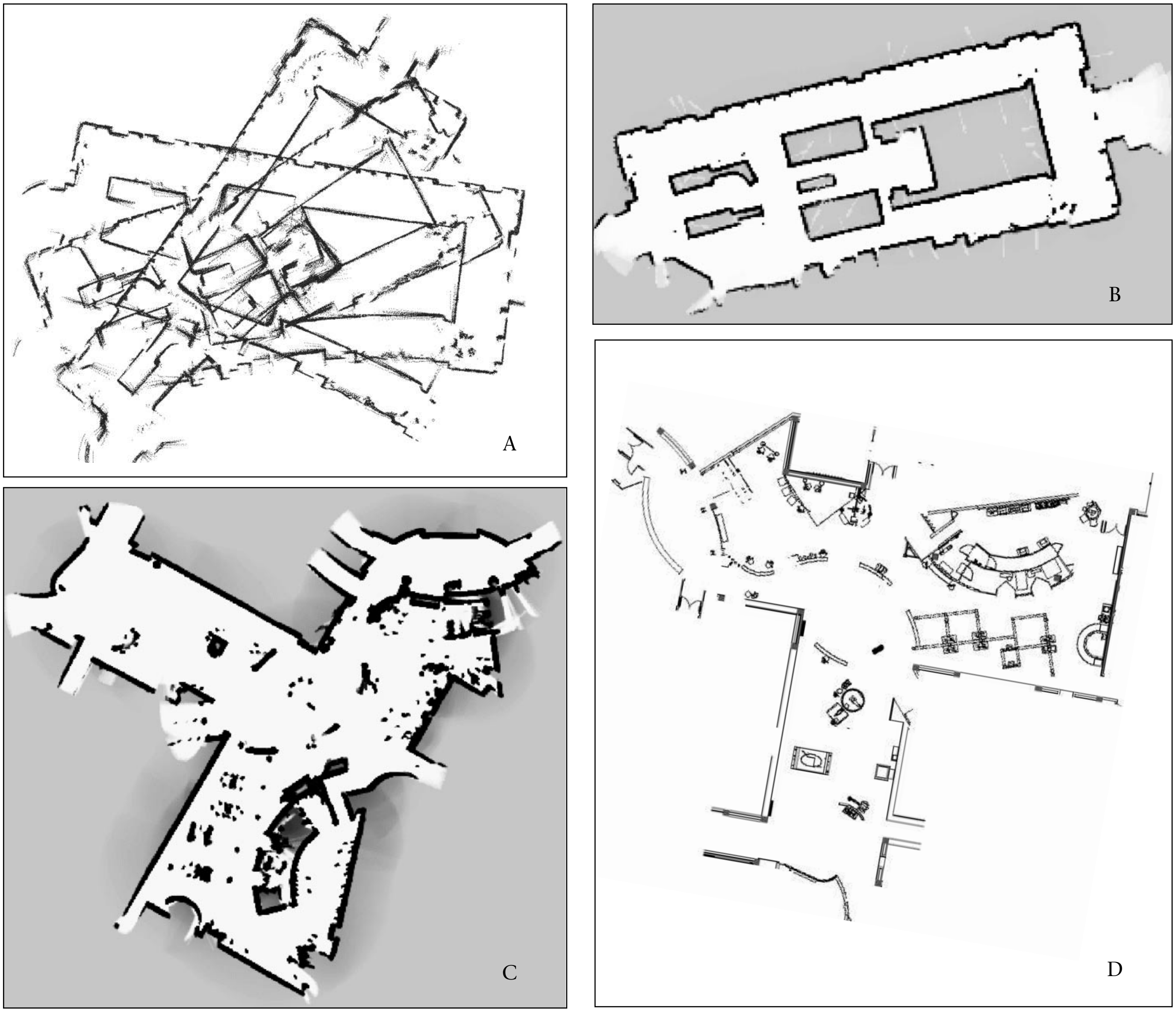


Figure 6. Raw Data, Maps, and a Computer-Aided Design Model.

A. Raw data of a large open hall (the Dinosaur Hall in the Carnegie Museum of Natural History, Pittsburgh, Pennsylvania). B. Map constructed using EM and occupancy grid mapping. C. Occupancy grid map of another museum (the Tech Museum in San Jose, California). D. Architectural blueprint for comparison.

robot control is much poorer developed than probabilistic perception because of the enormous computational complexity of decision making. However, within AI, this issue has recently received considerable attention. Even in robotics, some noticeable successes have been achieved, where probabilistic algorithms outperformed conventional, nonprobabilistic algorithms (Kaelbling, Cassandra, and Kurien 1996; Simmons and Koenig 1995).

One such algorithm is the *coastal navigation algorithm* (Roy et al. [1999]), a motion planning algorithm for mobile robots that takes uncertainty into account. The algorithm was originally motivated by the observation that ships that navigate through open water without a global positioning system (GPS) often stay in close proximity to the coast to reduce the danger of getting lost. The same applies to mobile robots: The choice of control can have a profound impact on the likelihood of localization errors. The coastal navigation algorithm selects paths accordingly, explicitly considering uncertainty.

To study this algorithm, let us step back a little and consider the mathematical framework that underlies this and many other probabilistic control algorithms: *partially observable Markov decision processes* (POMDPs). A POMDP is a framework for acting optimally under uncertainty in sequential decision tasks. Although POMDPs can be traced back to the 1970s (Monahan 1982; Sondik 1978; Smallwood and Sondik 1973), the AI community has only recently begun to pay attention to this framework, motivated by the important work of Littman, Cassandra, and Kaelbling (Kaelbling, Littman, and Cassandra 1998; Littman, Cassandra, and Kaelbling 1995). POMDPs address the problem of choosing actions to minimize a scalar (immediate) cost function, denoted $C(s)$. For example, in robot motion planning, one might set $C(s) = 0$ for goal locations and –1 elsewhere. Because reaching a goal location typically requires a whole sequence of actions, the control objective is to minimize the expected cumulative cost:

$$J = \sum_{\tau=t+1}^{t+T} E\left[C(s_\tau)\right] \tag{13}$$

Here the expectation is taken over all future states. T may be ∞, in which case, cost is often discounted over time by an exponential factor. Many important POMDP algorithms (Kaelbling, Littman, and Cassandra 1998; Littman, Cassandra, and Kaelbling 1995) are *offline algorithms*, in the sense that they calculate a policy for action selection for arbitrary situations (that is, belief states) in an explicit, offline phase. The policy is denoted π and maps belief states into actions. The most prominent approach to calculating π is *value iteration* (Howard 1960; Bellman 1957), a version of dynamic programming for computing the expected cumulative cost of belief states that has become highly popular in the field of reinforcement learning (Sutton and Barto 1998; Kaelbling, Littman, and Moore 1996). Value iteration in belief space computes a value function, denoted by V, that in the ideal case measures the expected cumulative cost if one starts in a state s drawn according to the belief distribution b and acts optimally thereafter. Thus, the value $V(b)$ of the belief state is the best possible cumulative cost one can expect for being in b. This is expressed as

$$V(b) = \int \sum_{\tau=t+1}^{t+T} E\left[C(s_\tau) \middle| s_t = s\right] b(s)\, ds \tag{14}$$

Following Bellman (1957) and Sutton and Barto (1998), the value function can be computed by recursively adjusting the value of individual belief states b according to

$$V(b) \leftarrow \min_a \int \left[V(b') + C(b')\right] p\left(b' \middle| a, b, m\right) db' \tag{15}$$

which assigns $V(b)$ to the expected value at the next belief, b'. Here, the immediate cost of a belief state b' is obtained by integrating over all states $C(b') = \int C(s')b'(s')ds'$. The conditional distribution $p(b' \mid a, b, m)$ is the belief space counterpart to the next state distribution, which is obtained as follows:

$$p\left(b' \middle| a, b, m\right) = \int p\left(b' \middle| o', a, b, m\right) p\left(o' \middle| a, b, m\right) do' \tag{16}$$

where $p(b' \mid o', a, b, m)$ is a Dirac distribution defined through equation 3, and

$$p\left(o' \middle| a, b, m\right) = \iint p\left(o' \middle| s', m\right) p\left(s' \middle| a, s, m\right) b(s)\, ds'\, ds \tag{17}$$

Once V has been computed, the optimal policy is obtained by selecting actions that minimize the expected V value over all available actions:

$$\pi(b) = \operatorname*{argmin}_a \int V(b')\, p\left(b' \middle| a, b, m\right) db' \tag{18}$$

Although this approach defines a mathematically elegant and consistent way to compute the optimal policy from the known densities $p(s' \mid a, s, m)$ and $p(o' \mid s', m)$, which are in fact the exact same densities used in MCL, there are two fundamental problems. First, in continuous domains, the belief space is the space of all distributions over the continuum, which is an infinitely dimensional space. Consequently, no exact method exists for calculating V in the general case. Second, even if the state space is discrete, which is commonly assumed in the POMDP framework, the computational burden can be enormous because for state spaces of size n, the corresponding belief space is an $(n–1)$-dimensional continuous space. The best known solutions, such as the witness algorithm (Kaelbling, Littman, and Cassandra 1998), can only handle problems of the approximate size of 100 states and a planning horizon of no more than $T = 5$ steps. In contrast, state spaces in robotics routinely possess orders of magnitude more states even under crude discretizations, which makes approximating imperative.

Coastal navigation is an extension of POMDPs that relies on an approximate representation for belief states b. The underlying assumption is that the exact nature of the uncertainty is irrelevant for action selection; instead, it suffices to know the degree of uncertainty as expressed by the entropy of a belief state $H[b]$. Thus, coastal navigation represents belief states by the following tuple:

$$\bar{b} = \left\langle \operatorname*{argmax}_s b(s);\, H[b] \right\rangle \tag{19}$$

Although this approximation is coarse, it causes value iteration to scale exponentially better to large state spaces than the full POMDP solution. Moreover, it still exhibits good performance in practice.

Figure 7 shows an example trajectory calculated by the coastal navigation algorithm for a large, featureless environment: a Smithsonian museum in Washington, D.C. The goal of motion is to reach a target location with high probability. By considering uncertainty, the coastal planner generates paths that actively seek the proximity of known obstacles to minimize the localization error—at the expense of an increased path length when compared to the shortest path. Experimental results (Roy et al. 1999) have shown that the success rate of the coastal planner is superior to conventional shortest path planners that ignore the inherent uncertainty in robot motion.

Coastal navigation is only one out of many examples. It highlights an important principle, namely, that crude approximations can go a long way when implementing probabilistic control algorithms. Recent research led to a range of other control algorithms that rely on approximate probabilistic representations. Of particular importance are algorithms for maximizing knowledge gain, which typically rely on a single-step search horizon to generate robot control. Examples include the rich work on robot exploration in which robots (or teams) select actions to maximally reduce their uncertainty about their environments (Simmons et al. 2000; Thrun 1998; Yamauchi et al. 1998; Koenig and Simmons 1993; Dudek et al. 1991; Kuipers and Byun 1991). They also include work on active localization (Fox, Burgard, and Thrun 1998a; Burgard, Fox, and Thrun 1997), where a robot moves to places that maximally disambiguate its pose. Another class of approaches relies on tree search for policy determination, such as the work on active perception and sensor planning by Kristensen (1997, 1996). His approach uses models of uncertainty to select the appropriate sensors in an indoor navigation task. All these approaches have demonstrated that probabilistic algorithms lead to more robust solutions to important control problems in robotics.

A Case Study: Museum Tour-Guide Robots

Probabilistic algorithms have been at the core of a number of state-of-the-art robot systems (see, for example, Bennett and Leonard [2000] and Dickmanns et al. [1994]), such as the XAVIER robot mentioned earlier (Simmons et al. 1997). In fact, recently, the number of publica-

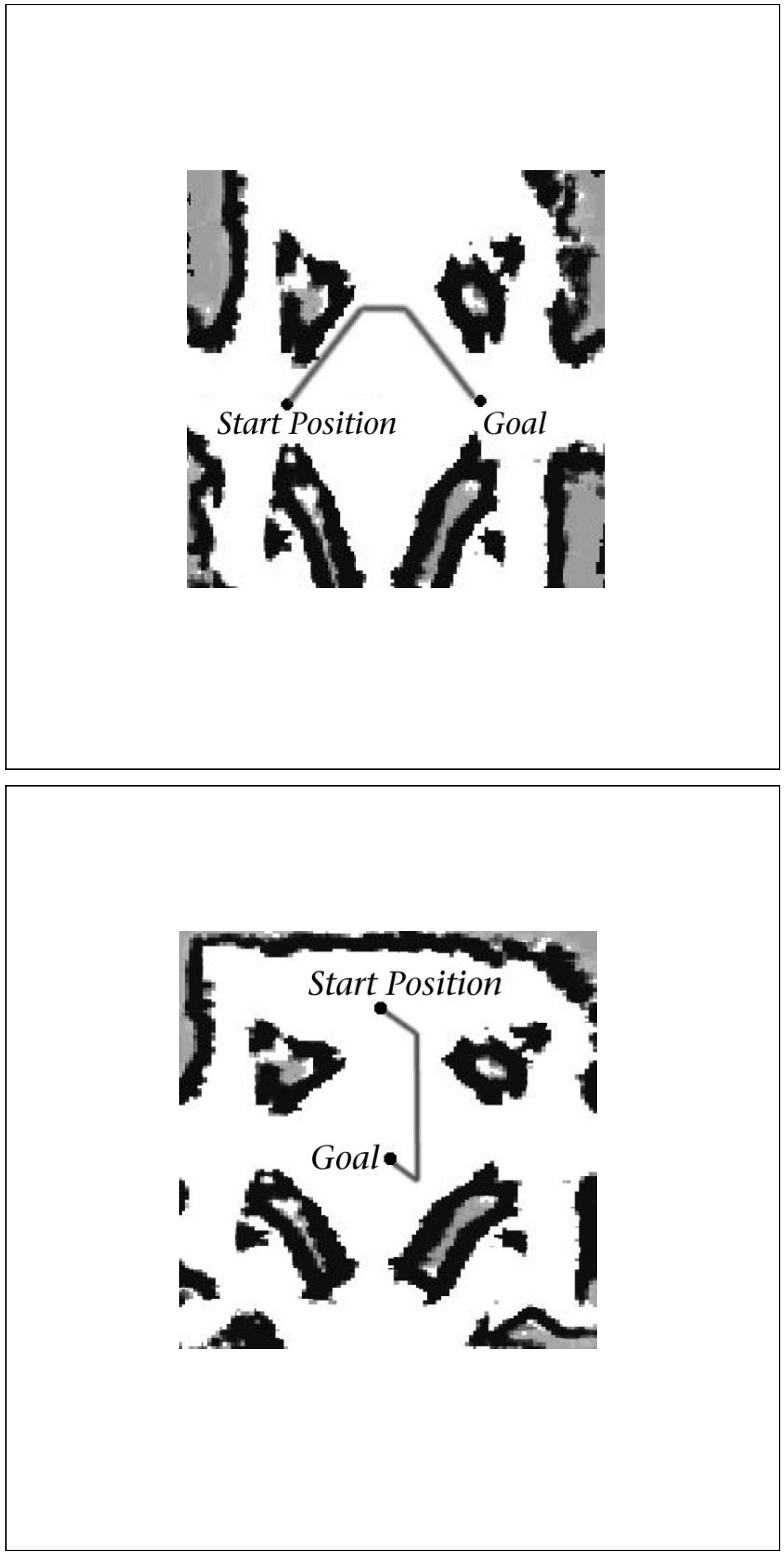
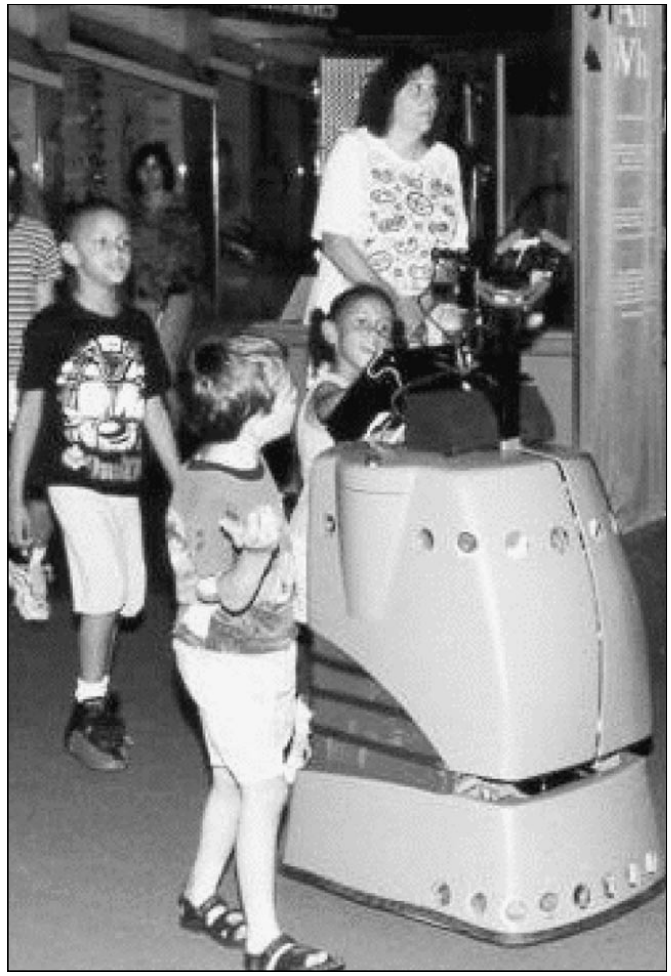


Figure 7. Coastal Plans: The Robot Actively Seeks the Proximity of Obstacles to Improve Its Localization.

The large open area in the center of this Smithsonian museum is approximately 20 meters wide and is usually crowded with people.



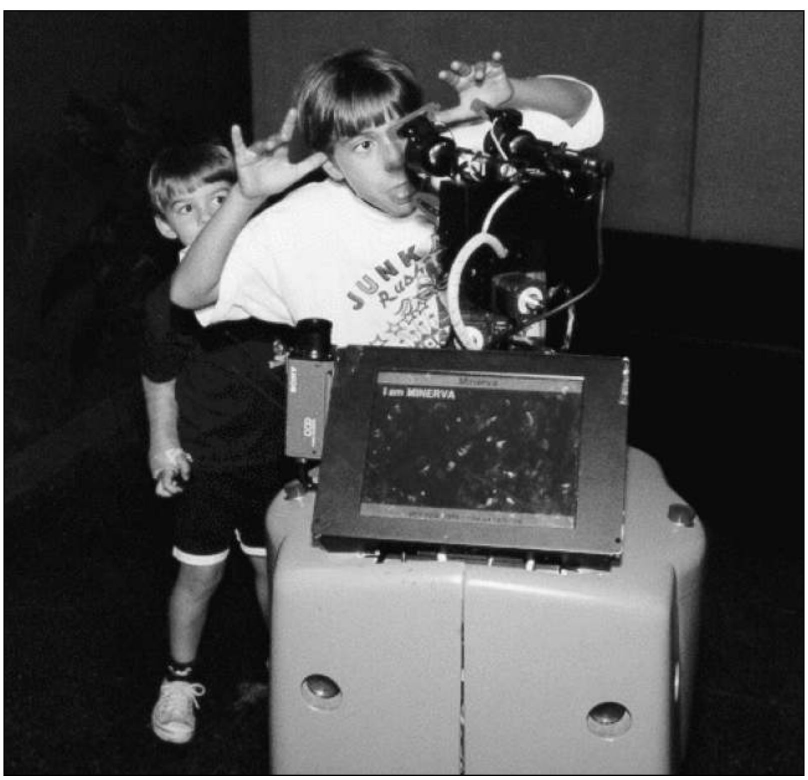


Figure 8. Probabilistic algorithms were used pervasively for the Musuem Tour Guide Robots RHINO *(top left) and* MINERVA *(top right and bottom left images).*

tions on statistically sound algorithms for perception and control has increased dramatically at leading robotics conferences.

In work at CMU and the University of Bonn, we recently developed two autonomous museum tour-guide robots (see also Horswill [1993] and Nourbakhsh et al. [1999]), which pervasively used probabilistic algorithms for navigation and people interaction. Pictures of both robots are shown in figure 8. The robot shown on the left, RHINO, was the world's first museum tour-guide robot, which was deployed at the Deutsches Museum in Bonn, Germany, in 1997. The other robot, MINERVA, led thousands of people through a crowded Smithsonian museum in Washington, D.C. Both robots were developed to showcase probabilistic algorithms in complex and highly dynamic environments. They were unique in their ability to navigate safely and reliably in the proximity of people.

The task of these robots was simple: to attract people, interact with them, and guide them from exhibit to exhibit. Several factors made this problem challenging: To find their way around, the robots had to know where they were. However, large crowds of people almost permanently blocked the robots' sensors, making localization a difficult problem. In fact, people frequently sought to confuse the robot or force it close to hazards such as downward staircases. To make matters worse, the robots' ability to sense such hazards was extremely limited. For example, the sensors consistently failed to sense glass cases put up for the protection of certain exhibits, and neither robot possessed a sensor to detect staircases. Thus, accurate localization played a prime role in avoiding collisions with such "invisible" obstacles and hazards as staircases, whose location was modeled in the map.

To challenge the autonomy of our robots, we did not modify the environment in any way. Even though the museums were crowded, the robots had to navigate at approximate walking speeds to sustain people's interest while they avoided collisions with people at all costs.

Detailed descriptions of the robots' software architecture and experimental findings are beyond the scope of this article (see Burgard et al. [1999] and Thrun et al. [1999]); I simply note here that probabilistic algorithms were used at virtually all levels of the software architecture. In total, both robots traversed a distance of more than 60 kilometers, with average speeds between 30 centimeters a second and 80 centimeters a second and top speeds well above 160 centimeters a second. In RHINO's case, every failure was carefully evaluated; only one major localization failure was observed over a period of six days (Burgard 1999); however, this localization failure coincided with a malfunctioning of the sonar sensors. RHINO used a probabilistic collision-avoidance routine that guaranteed, with high probability, that the robot would not collide with "invisible" obstacles even when the robot was highly uncertain where it was (Fox, Burgard, and Thrun 1998b). In addition, MINERVA utilized probabilistic algorithms to learn occupancy grid maps of the museums. In other experiments, a practical probabilistic algorithm was devised for active exploration, both in pursuit of finding out where a robot was (Burgard, Fox, and Thrun 1997) and learning maps of unknown terrain (Thrun 1998) with teams of collaborating robots (Burgard et al. 2000).

In all these cases, the probabilistic nature of the algorithms has been essential for achieving robustness in the face of uncertain and dynamic environments. In addition, all these algorithms rely on sometimes remarkably simple approximations and shortcuts that make hard problems computationally tractable.

Discussion

The last few decades have seen a flurry of different software design paradigms for autonomous robots. Early work on model-based robotics often assumed the availability of a complete and accurate model of the robot and its environment, relying on planners (or theorem provers) to generate actions (Latombe 1991; Canny 1987; Schwartz, Scharir, and Hopcroft 1987). Such approaches are often inapplicable to robotics because of the difficulty of generating models that are sufficiently accurate and complete. Recognizing this limitation, some researchers have advocated model-free reactive approaches. Instead of relying on planning, these approaches require programmers to program controllers directly. A popular example of this approach is the *subsumption architecture* (Brooks 1989), where controllers are composed of small finite-state automata that map sensor readings into control yet still retain a minimum of internal state. Some advocates of this approach went so far as to refuse the need for internal models and internal state altogether (Connell 1990; Brooks 1989). Observing that "the world is its own best model," behavior-based approaches usually rely on immediate sensor feedback for determining a robot's action. Obvious limits in perception (for example, robots can't see through walls) pose clear boundaries on the type of task that can be tackled with this approach, leading to the conclusion that although the world might well be its most accurate model, it is not necessarily its most accessible one. And accessibility matters!

The probabilistic approach is somewhere between these two extremes. Probabilistic algorithms rely on models, just like the classical plan-based approach. For example, Markov localization requires a perception model $p(o \mid s, m)$, a motion model $p(s' \mid a, s)$, and a map of the environment. However, because these models are probabilistic, they only need to be approximate, making them much easier to implement (and to learn!) than if we had to meet the accuracy requirements of traditional approaches. Additionally, the ability to acknowledge existing uncertainty and even anticipate upcoming uncertainty in planning leads to qualitatively new solutions in a range of robotics problems, as demonstrated in this article.

Probabilistic algorithms are similar to behavior-based approaches in that they place a strong emphasis on sensor feedback. Because probabilistic models are insufficient to predict the actual state, sensor measurements play a vital role in state estimation and, thus, in the determination of a robot's actual behavior. However, they differ from behavior-based approaches in that they rely on planning and that a robot's behavior is not just a function of a small number of recent sensor readings. To illustrate the importance of the latter difference, imagine placing a mobile robot in a crowded place full of invisible hazards! Surely, the problem can be solved by adding more sensors; however, such an approach is expensive at best, but more often, it will be plainly infeasible because of the lack of appropriate sensors. The robot RHINO, for example, was equipped with five different sensor systems—(1) vision, (2) laser, (3) sonar, (4) infrared, and (5) tactile—yet it still could not perceive all the hazards and obstacles in this fragile environment with the necessary reliability (see Burgard et al. [1999] for a discussion). Thus, it seems unlikely that a reactive approach could have performed anywhere near as reliably and robustly in this task.

Probabilistic robotics is closely related to a rich body of literature on uncertainty in AI (UAI) (Heckerman [1995] and Pearl [1988] are good starting points, as is any recent UAI proceedings). In fact, many of the basic algorithms in robotics have counterparts in the UAI community, the major difference being that their focus tends to be on discrete spaces, whereas robots typically live in continuous spaces. Also, building real robotic systems constrains the assumptions under which one can operate. Consequently, approximations and real-time algorithms play a greater role in robotics than they currently play in mainstream AI.

One of the most exciting aspects of the probabilistic paradigm is that it allows for great new opportunities in basic robotics and AI research, with high potential for high impact in robotics and beyond. Probabilistic algorithms are still far from mainstream in robotics, and a range of problems appear to be highly amenable to probabilistic solutions. I conclude this article by laying out five broad areas of research that I deem to be highly important: (1) representations, (2) learning, (3) high-level reasoning and programming, (4) theory, and (5) innovative applications.

Representations: The choice of representation is crucial in the design of any probabilistic algorithm because it determines its robustness, efficiency, and accuracy. Recent research has already led to a range of representations for probabilistic information in continuous spaces, such as the particle representation in the example described earlier. However, the development of new representations is absolutely essential for scaling up to more complex problems, such as the control of highly articulated robots or multirobot coordination.

Learning: The probabilistic paradigm lends itself naturally to learning, but little work has been carried out on automatically learning models (or behaviors) in real-world robotic applications using probabilistic representations. Many of today's best learning algorithms are grounded in statistical theory similar to the one underlying the current approach. I conjecture that a better understanding of how to automatically acquire probabilistic models and behaviors over the lifetime of a robot has the potential to lead to new, innovative solutions to a range of hard open problems in robotics.

High-level reasoning and programming: Current research on probabilistic robotics predominately focuses on low-level perception and control. However, the issues raised in this article apply to all levels of reasoning and decision making (Boutilier et al. 2000). The issue of probabilistic high-level reasoning and programming for robots remains poorly explored. Research is needed on algorithms that integrate probabilistic representations into high-level robot control (see, for example, Glesner and Koller [1995], Poole [1993], and Halpern [1990]).

Theory: The groundedness in statistical theory makes probabilistic approaches to robotics well suited for theoretical investigation. However, existing models are often too restrictive to characterize robot interaction in complex environments. For example, little is known about the consequences of violating the Markov assumption that underlies much of today's work on localization and mapping, even though virtually all interesting environments violate this assumption. Little is also known about the effect of approximate representation on the performance of robotic controllers. A better theoretical understanding of probabilistic robotic algorithms is clearly desirable and would further our understanding of the benefits and limitations of this approach.

Innovative applications: Finally, there is tremendous opportunity in applying probabilistic algorithms to a range of important robotic problems, including multirobot coordination, sensor-based manipulation, and human-robot interaction.

I hope that this article motivated the probabilistic approach to robotics and stimulates new thinking in this exciting area. Ultimately, I believe that probabilistic algorithms are essential for a much broader class of embedded systems equipped with sensors and actuators.

Acknowledgments

I want to thank Wolfram Burgard, Frank Dellaert, Dieter Fox, Nicholas Roy, and the other members of CMU's Robot Learning Lab and the Mobile Robot Lab at the University of Bonn, whose contributions to this research were essential.

This research is sponsored by the National Science Foundation (CAREER grant IIS-9876136 and regular grant IIS-9877033) and by DARPA-ATO by TACOM (contract DAAE07-98-C-L032) and DARPA-ISO by Rome Labs (contract F30602-98-2-0137) and is gratefully acknowledged. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of the United States Government or any of the sponsoring institutions.

References

Bellman, R. E. 1957. *Dynamic Programming*. Princeton, N.J.: Princeton University Press.

Bennett, A., and Leonard, J. J. 2000. A Behavior-Based Approach to Adaptive Feature Mapping with Autonomous Underwater Vehicles. *IEEE Journal of Oceanic Engineering* 25(2): 213–226.

Borenstein, J. 1987. The Nursing Robot System. Ph.D. dissertation, Technion, Haifa, Israel.

Borenstein, J.; Everett, B.; and Feng, L. 1996. *Navigating Mobile Robots: Systems and Techniques*. Wellesley, Mass.: A. K. Peters.

Boutilier, C.; Reiter, R.; Soutchanski, M.; and Thrun, S. 2000. Decision-Theoretic, High-Level Robot Programming in the Situation Calculus. In Proceedings of the Seventeenth National Conference on Artificial Intelligence, 355–362. Menlo Park, Calif.: American Association for Artificial Intelligence.

Brooks, R. 1990. Elephants Don't Play Chess. *Autonomous Robots* 6:3–15.

Brooks, R. A. 1989. A Robot That Walks: Emergent Behaviors from a Carefully Evolved Network. *Neural Computation* 1(2): 253.

Burgard, W.; Fox, D.; and Thrun, S. 1997. Active Mobile Robot Localization. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 1346–1352. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Burgard, W.; Fox, D.; Hennig, D.; and Schmidt, T. 1996. Estimating the Absolute Position of a Mobile Robot Using Position Probability Grids. In Proceedings of the Thirteenth National Conference on Artificial Intelligence, 896–901. Menlo Park, Calif.: American Association for Artificial Intelligence.

Burgard, W.; Fox, D.; Moors, M.; Simmons, R.; and Thrun, S. 2000. Collaborative Multirobot Exploration. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 476–481. Washington, D.C.: IEEE Computer Society.

Burgard, W.; Cremers, A. B.; Fox, D.; Hahnel, D.; Lakemeyer, G.; Schulz, D.; Steiner, W.; and Thrun, S. 1999. Experiences with an Interactive Museum Tour-Guide Robot. *Artificial Intelligence* 114(1–2): 3–55.

Canny, J. 1987. *The Complexity of Robot Motion Planning*. Cambridge, Mass.: The MIT Press.

Castellanos, J. A., and Tardos, J. D. 2000. *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Boston, Mass.: Kluwer Academic.

Castellanos, J. A.; Montiel, J. M. M.; Neira, J.; and Tardos, J. D. 1999. The SPMAP: A Probabilistic Framework for Simultaneous Localization and Map Building. *IEEE Transactions on Robotics and Automation* 15(5): 948–953.

Connell, J. 1990. *Minimalist Mobile Robotics*. San Diego, Calif.: Academic.

Cox, I. J. 1991. BLANCHE—An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle. *IEEE Transactions on Robotics and Automation* 7(2): 193–204.

Dean, T. L., and Boddy, M. 1988. An Analysis of Time-Dependent Planning. In Proceedings of the Seventh National Conference on Artificial Intelligence, 49–54. Menlo Park, Calif.: American Association for Artificial Intelligence.

Dean, T. L., and Kanazawa, K. 1989. A Model for Reasoning about Persistence and Causation. *Computational Intelligence* 5(3): 142–150.

Dellaert, F.; Fox, D.; Burgard, W.; and Thrun, S. 1999. Monte Carlo Localization for Mobile Robots. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 1322–1328. Washington, D.C.: IEEE Computer Society.

Dempster, A. P.; Laird, A. N.; and Rubin, D. B. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society Series B* 39(1): 1–38.

Dickmanns, E. D.; Behringer, R.; Dickmanns, D.; Hildebrandt, T.; Maurer, M.; Schiehlen, J.; and Thomanek, F. 1994. The Seeing Passenger Car VAMORS-P. In Proceedings of the International Symposium on Intelligent Vehicles. Washington, D.C.: IEEE Computer Society.

Doucet, A. 1998. On Sequential Simulation-Based Methods for Bayesian Filtering. Technical Report, CUED/F-INFENG/TR, 310, Department of Engineering, Cambridge University.

Doucet, A.; Gordon, N. J.; and de Freitas, J. F. G., editors. 2001. Sequential Monte Carlo Methods in Practice. New York: Springer-Verlag. Forthcoming.

Dudek, G.; Jenkin, M.; Milios, E.; and Wilkes, D. 1991. Robotic Exploration as Graph Construction. *IEEE Transactions on Robotics and Automation* 7(6): 859–865.

Elfes, A. 1989. Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation. Ph.D. dissertation, Department of Electrical and Computer Engineering, Carnegie Mellon University.

Elfes, A. 1987. Sonar-Based Real-World Mapping and Navigation. *IEEE Journal of Robotics and Automation*

RA-3(3): 249–265.

Engelson, S. 1994. Passive Map Learning and Visual Place Recognition. Ph.D. dissertation, Department of Computer Science, Yale University.

Engelson, S., and McDermott, D. 1992. Error Correction in Mobile Robot Map Learning. In Proceedings of the 1992 IEEE International Conference on Robotics and Automation, 2555–2560. Washington, D.C.: IEEE Computer Society.

Fox, D.; Burgard, W.; and Thrun, S. 1999. Markov Localization for Mobile Robots in Dynamic Environments. *Journal of Artificial Intelligence Research* 11:391–427.

Fox, D.; Burgard, W.; and Thrun, S. 1998a. Active Markov Localization for Mobile Robots. *Robotics and Autonomous Systems* 25(3–4): 195–207.

Fox, D.; Burgard, W.; and Thrun, S. 1998b. A Hybrid Collision Avoidance Method for Mobile Robots. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Washington, D.C.: IEEE Computer Society.

Fox, D.; Burgard, W.; Dellaert, F.; and Thrun, S. 1999. Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. In Proceedings of the Sixteenth National Conference on Artificial Intelligence. Menlo Park, Calif.: American Association for Artificial Intelligence.

Fox, D.; Burgard, W.; Thrun, S.; and Cremers, A. B. 1998. Position Estimation for Mobile Robots in Dynamic Environments. In Proceedings of the Fifteenth National Conference on Artificial Intelligence. Menlo Park, Calif.: American Association for Artificial Intelligence.

Glesner, S., and Koller, D. 1995. Constructing Flexible Dynamic Belief Networks from First-Order Probabilistic Knowledge Bases. In *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty,* eds. C. Froidevaux and J. Kohlas, 217–226, Berlin: Springer Verlag.

Gutmann, J.-S., and Konolige, K. 2000. Incremental Mapping of Large Cyclic Environments. In Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA). Washington, D.C.: IEEE Computer Society.

Gutmann, J.-S., and Nebel, B. 1997. Navigation Mobiler Roboter mit Laserscans (Navigation of Mobile Robots with Laser Scans). In *Autonome Mobile Systeme* (Autonomous Mobile Systems). Berlin: Springer Verlag.

Guzzoni, D.; Cheyer, A.; Julia, L.; and Konolige, K. 1997. Many Robots Make Short Work. *AI Magazine* 18(1): 55–64.

Halpern, J. 1990. An Analysis of First-Order Logics of Probability. *Artificial Intelligence* 46:311–350.

Heckerman, D. 1995 (rev. 1996). A Tutorial on Learning with Bayesian Networks. Technical Report MSR-TR-95-06, Microsoft Research, Redmond, Washington.

Horswill, I. 1993. POLLY: A Vision-Based Artificial Agent. In Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93). Menlo Park, Calif.: American Association for Artificial Intelligence.

Howard, R. A. 1960. *Dynamic Programming and Markov Processes*. Cambridge, Mass.: MIT Press.

Isard, M., and Blake, A. 1998. Condensation: Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision* 29(1): 5–28.

Isard, M., and Blake, A. 1996. Contour Tracking by Stochastic Propagation of Conditional Density. In *Proceedings of the European Conference on Computer Vision,* 343–356. New York: Springer.

Jazwinsk, A. M. 1970. *Stochastic Processes and Filtering Theory.* San Diego, Calif.: Academic.

Kaelbling, L. P.; Cassandra, A. R.; and Kurien, J. A. 1996. Acting under Uncertainty: Discrete Bayesian Models for Mobile-Robot Navigation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Washington, D.C.: IEEE Computer Society.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and Acting in Partially Observable Stochastic Domains. *Artificial Intelligence* 101(1–2): 99–134.

Kaelbling, L. P.; Littman, M. L.; and Moore, A. W. 1996. Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research* 4:237–285.

Kalman, R. E. 1960. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering* (Transactions of the American Society of Mechanical Engineers) 82:35–45.

Koenig, S., and Simmons, R. 1996. Passive Distance Learning for Robot Navigation. In Proceedings of the Thirteenth International Conference on Machine Learning, ed. L. Saitta, 266–274. San Francisco, Calif.: Morgan Kaufmann.

Koenig, S., and Simmons, R. G. 1993. Exploration with and without a Map. In Proceedings of the AAAI Workshop on Learning Action Models at the Eleventh National Conference on Artificial Intelligence, 28–32. Technical Report WS-93-06. Menlo Park, Calif.: American Association for Artificial Intelligence.

Kortenkamp, D.; Bonasso, R. P.; and Murphy, R., editors. *AI-Based Mobile Robots: Case Studies of Successful Robot Systems.* Cambridge, Mass.: MIT Press.

Kristensen, S. 1997. Sensor Planning with Bayesian Decision Theory. *Robotics and Autonomous Systems* 19(3–4): 273–286.

Kristensen, S. 1996. Sensor Planning with Bayesian Decision Analysis. Ph.D. dissertation, Faculty of Technology and Science, Aalborg University.

Kuipers, B., and Byun, Y.-T. 1991. A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations. *Journal of Robotics and Autonomous Systems* 8:47–63.

Latombe, J.-C. 1991. *Robot Motion Planning.* Boston, Mass.: Kluwer Academic.

Lenser, S., and Veloso, M. 2000. Sensor Resetting Localization for Poorly Modeled Mobile Robots. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 1225–1232. Washington, D.C.: IEEE Computer Society.

Leonard, J. J., and Durrant-Whyte, H. F. 1992. *Directed Sonar Sensing for Mobile Robot Navigation.* Boston, Mass.: Kluwer Academic.

Leonard, J. J., and Feder, H. J. 1999. A Computationally Efficient Method for Large-Scale Concurrent Mapping and Localization. Paper presented at the Ninth International Symposium on Robotics Research, 9–12 October, Snowbird, Utah.

Leonard, J. J.; Durrant-Whyte, H. F.; and Cox, I. J. 1992. Dynamic Map Building for an Autonomous Mobile Robot. *International Journal of Robotics Research* 11(4): 89–96.

Littman, M. L.; Cassandra, A. R.; and Kaelbling, L. P. 1995. Learning Policies for Partially Observable Environments: Scaling Up. Paper presented at the Twelfth International Conference on Machine Learning, 9–12 July, Tahoe City, California.

Liu, J., and Chen, R. 1998. Sequential Monte Carlo Methods for Dynamic Systems. *Journal of the American Statistical Association* 93:1032–1044.

Lu, F., and Milios, E. 1997. Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots* 4:333–349.

McLachlan, G. J., and Krishnan, T. 1997. *The EM Algorithm and Extensions.* Wiley Series in Probability and Statistics. New York: Wiley.

Maybeck, P. S. 1990. The Kalman Filter: An Introduction to Concepts. In *Autonomous Robot Vehicles*, eds. I. Cox and G. Wilfong. New York: Springer Verlag.

Monahan, G. E. 1982. A Survey of Partially Observable Markov Decision Processes: Theory, Models, and Algorithms. *Management Science* 28(1): 1–16.

Moravec, H. P. 1988. Sensor Fusion in Certainty Grids for Mobile Robots. *AI Magazine* 9(2): 61–74.

Moutarlier, P., and Chatila, R. 1989a. An Experimental System for Incremental Environment Modeling by an Autonomous Mobile Robot. Paper presented at the International Symposium on Experimental Robotics, June, Montreal, Canada.

Moutarlier, P., and Chatila, R. 1989b. Stochastic Multisensory Data Fusion for Mobile Robot Location and Environment Modeling. Paper presented at the Fifth Symposium on Robotics Research, 28–31 August, Tokyo, Japan.

Nourbakhsh, I. R.; Bobenage, J.; Grange, S.; Lutz, R.; Meyer, R.; and Soto, A. 1999. An Effective Mobile Robot Educator with a Full-Time Job. *Artificial Intelligence* 114(1–2): 95–124.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, Calif.: Morgan Kaufmann.

Pitt, M., and Shephard, N. 1999. Filtering via Simulation: Auxiliary Particle Filter. *Journal of the American Statistical Association* 94:590–599.

Poole, D. 1993. Probabilistic Horn Abduction and Bayesian Networks. *Artificial Intelligence* 64:81–129.

Rabiner, L. R., and Juang, B. H. 1986. An Introduction to Hidden Markov Models. *IEEE Acoustics, Speed, and Signal Processing Magazine* 3(1): 4–16.

Rencken, W. D. 1993. Concurrent Localization and Map Building for Mobile Robots Using Ultrasonic Sensors. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2129–2197. Washington, D.C.: IEEE Computer Society.

Roy, N.; Burgard, W.; Fox, D.; and Thrun, S. 1999. Coastal Navigation: Robot Navigation under Uncertainty in Dynamic Environments. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 35–40. Washington, D.C.: IEEE Computer Society.

Rubin, D. B. 1988. Using the SIR Algorithm to Simulate Posterior Distributions. In *Bayesian Statistics 3,* eds. M. H. Bernardo, K. M. DeGroot, D. V. Lindley, and A. F. Smith. Oxford, U.K.: Oxford University Press.

Russell, S., and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach.* Englewood Cliffs, N.J.: Prentice Hall.

Schwartz, J. T.; Scharir, M.; and Hopcroft, J. 1987. *Planning, Geometry, and Complexity of Robot Motion.* Norwood, N.J.: Ablex.

Shatkay, H. 1998. Learning Models for Robot Navigation. Ph.D. dissertation, Computer Science Department, Brown University.

Shatkay, H., and Kaelbling, L. 1997. Learning Topological Maps with Weak Local Odometric Information. In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, 920–927. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Simmons, R., and Koenig, S. 1995. Probabilistic Robot Navigation in Partially Observable Environments. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 1080–1087. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Simmons, R.; Goodwin, R.; Haigh, K.; Koenig, S.; and O'Sullivan, J. 1997. A Layered Architecture for Office Delivery Robots. Paper presented at the First International Conference on Autonomous Agents, 5–8 February, Marina del Rey, California.

Simmons, R.; Apfelbaum, D.; Burgard, W.; Fox, M.; Moors, D.; Thrun, S.; and Younes, H. 2000. Coordination for Multi-Robot Exploration and Mapping. In Proceedings of the Seventeenth National Conference on Artificial Intelligence, 852–858. Menlo Park, Calif.: American Association for Artificial Intelligence.

Smallwood, R. W., and Sondik, E. J. 1973. The Optimal Control of Partially Observable Markov Processes over a Finite Horizon. *Operations Research* 21:1071–1088.

Smith, R.; Self, M.; and Cheeseman, P. 1990. Estimating Uncertain Spatial Relationships in Robotics. In *Autonomous Robot Vehicles,* eds. I. J. Cox and G. T. Wilfong, 167–193. New York: Springer-Verlag.

Sondik, E. J. 1978. The Optimal Control of Partially Observable Markov Processes over the Infinite Horizon: Discounted Costs. *Operations Research* 26(2): 282–304.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: An Introduction.* Cambridge, Mass.: MIT Press.

Thrun, S. 1998a. Bayesian Landmark Learning for Mobile Robot Localization. *Machine Learning* 33(1): 41–76.

Thrun, S. 1998b. Learning Metric-Topological Maps for Indoor Mobile Robot Navigation. *Artificial Intelligence* 99(1): 21–71.

Thrun, S.; Burgard, W.; and Fox, D. 2000. A Real-Time Algorithm for Mobile Robot Mapping with Applications to Multi-Robot and 3D Mapping. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 321–328. Washington, D.C.: IEEE Computer Society.

Thrun, S.; Fox, D.; and Burgard, W. 2000. Monte Carlo Localization with Mixture Proposal Distribution. In Proceedings of the Seventeenth National Conference on Artificial Intelligence, 859–865. Menlo Park, Calif.: American Association for Artificial Intelligence.

Thrun, S.; Fox, D.; and Burgard, W. 1998. A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots. *Machine Learning* 31:29–53.

Thrun, S.; Bennewitz, M.; Burgard, W.; Cremers, A. B.; Dellaert, F.; Fox, D.; Hahnel, D.; Rosenberg, C.; Roy, N.; Schulte, J.; and Schulz, D. 1999. MINERVA: A Second-Generation Mobile Tour-Guide Robot. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 1999–2005. Washington, D.C.: IEEE Computer Society.

Yamauchi, B., and Langley, P. 1997. Place Recognition in Dynamic Environments. *Journal of Robotic Systems* 14(2): 107–120.

Yamauchi, B.; Langley, P.; Schultz, A. C.; Grefenstette, J.; and Adams, W. 1998. MAGELLAN: An Integrated Adaptive Architecture for Mobile Robots. Technical Report 98-2, Institute for the Study of Learning and Expertise (ISLE), Palo Alto, California.

Zilberstein, S., and Russell, S. 1995. Approximate Reasoning Using Anytime Algorithms. In *Imprecise and Approximate Computation,* ed. S. Natarajan. Dordrecht, The Netherlands: Kluwer Academic.

Sebastian Thrun is assistant professor of computer science and robotics at Carnegie Mellon University, where he pursues research on AI, machine learning, and robotics. Thrun and his group have pioneered the development of probabilistic algorithms for robotics. Thrun has authored over 150 scientific articles and won best conference paper awards at AAAI98, DAGM99, and ICRA2000. Currently, he is writing a book on probabilistic robotics together with his colleagues Dieter Fox and Wolfram Burgard. His e-mail address is thrun@cs.cmu.edu.