A Probabilistic Technique for Simultaneous Localization and Door State Estimation with Mobile Robots in Dynamic Environments

Dzintars Avots[1], Edward Lim[1], Romain Thibaux[1], Sebastian Thrun[2]

[1] *Stanford University, {dzin,elim,thibaux}@cs.stanford.edu*
[2] *Carnegie Mellon University, thrun@cs.cmu.edu*

Abstract

Virtually all existing mobile robot localization techniques operate on a static map of the environment. When the environment changes (e.g., doors are opened or closed), there is an opportunity to simultaneously estimate the robot's pose and the state of the environment. The resulting estimation problem is high-dimensional, rendering current localization techniques inapplicable. This paper proposes an efficient, factored estimation algorithm for mixed discrete-continuous state estimation. Our algorithm integrates particle filters for robot localization, and conditional binary Bayes filters for estimating the dynamic state of the environment. Experimental results illustrate that our algorithm is highly effective in estimating the status of doors, and outperforms a state-of-the-art localizer in dynamic environments.

1 Introduction

In the past decade, mobile robot localization and mapping has received substantial interest in AI and robotics [1, 12]. The localization problem concerns itself with estimating the pose of a robot relative to a fixed map [9, 11], whereas the mapping addresses the problem of learning a map from sensor data [4, 13, 16].

A striking characteristic of the rich literature on this topic is that virtually all published work assumes that the environment is static. This is in contrast to most robotic environments, which usually change over time. For example, most office environments possess doors, chairs, and other items whose location or state changes over time. A worthwhile research goal, thus, is to extend existing techniques by algorithms that can perform localization and mapping functions in dynamic environments. Of course, the recognition that natural environments are dynamic is not new. For example, Fox et al. [8] have developed a localization algorithm that is robust to the presence of people in the environment, as demonstrated in a crowded museum experiment [2]. However, such algorithms still assume a static map, which is never revised in accordance to sensor evidence. Instead, the paper is an example of a more common methodology of treating dynamic effects. In mobile robotics, dynamic effects are usually regarded as noise and filtered out [7].

This paper addresses a specific dynamic environment problem: Estimating the state of a set binary state variables (e.g., doors, or grid cells in an occupancy grid map) in an environment that changes. The difficulty of this problem arises from the fact that localizing a robot in such environment is generally difficult. From an estimation point of view, there is interaction between the problem of estimating the robot's location and the state of the environment. For example, if a robot encounters an obstacles in an area that previously corresponded to an open door, there exist two quite complementary ways to explain such measurements: Either the door status has changed, or the robot is not where it believes to be (a less plausible explanation would be that the observation is the effect of sensor noise). As this example illustrates, the problem of localization and environment state estimation cannot be decoupled.

This paper proposes an efficient algorithm for localizing a robot in an environment with discrete states, while simultaneously estimating the state of the environment. We show that under the appropriate formulation, the estimation problem can be factored into a problem reminiscent (but not identical) to conventional mobile robot localization, and a number of conditional Bayes filters for estimating the environment's state. The specific algorithm extends the work by Murphy [15] on state estimation in Bayesian networks. Like Murphy, our approach employs a particle filter for estimating the robot's pose. The discrete environment state variables are estimated via discrete Bayesian filters (similar in nature to the standard occupancy grid algorithm [14]). However, those Bayes filters are conditioned on the robot's path estimate, hence are attached to individual particles. The resulting algorithm scales linear in the number of state variables in the environment, and linear in the number of particles. This is significantly more efficient than existing existing mapping algorithms that combine localization and map estimation [4, 13], which scale quadratically in the number of environment state variables (and typically involve only continuous state variables).

Experimental results summarized in this paper illustrate that our algorithm is highly effective in estimating the state of doors that change dynamically in the experimental setting. It also outperforms an existing state-of-the-art localization algorithm that assumes a static world. The environmental testbed includes a physical robot operating in an office building, and a high-fidelity robot simulator for quantitative evaluation.

2 The Concurrent Localization and Environment State Estimation Problem

The algorithm presented in this paper assumes that the environment possesses K binary features that may change dynamically (and independently of each other) over time. Each underlying state variable will be denoted y_k, where $k = \ldots, K$. The state of the variable y_k at time t will be denoted $y_{k,t}$. Similarly, the robot pose at time t (a three-dimensional continuous variable comprising the coordinates and heading direction of the robot) will be denoted x_t. For brevity, we will write:

$$y_t \quad = \quad \{y_{1,t}, \ldots, y_{K,t}\} \tag{1}$$

for the set of all K state variables at time t, and will use the superscript ^{t} to refer to a set of quantities from time 0 to time t:

$$x^t \quad = \quad x_0, \ldots, x_t \tag{2}$$

As usual, we assume the environment is Markov, that is, x_t and y_t are the complete state of the environment.

To estimate the robot's pose x_t and the discrete state variables y_t, the robot processes two types of information: Sensor measurements (e.g., range scans) and motion controls. The sequence of sensor measurements will be denoted

$$z^t \quad = \quad z_0, \ldots, z_t \tag{3}$$

and the sequence of control

$$u^t \quad = \quad u_1, \ldots, u_t \tag{4}$$

which we assume (without loss of generality) to arrive in alternation. In our implementation, the controls are velocity commands to a mobile robot, and the measurements are obtained by laser range finders.

In addition, the dynamics of the robot and the environment are known—a common feature of localization and mapping algorithms. The dynamics are specified by the following conditional probabilities

$$p(x_t | u_t, x_{t-1}) \tag{5}$$

$$p(y_{k,t} | y_{k,t-1}) \tag{6}$$

which specify the effect of controls u_t on the robot's pose x, and the temporal behavior of the the discrete environment variables y, respectively. Notice that we assume that controls do not affect the environment state—however, this is not a principle restriction of our approach.

Finally, the robot is given inverse sensor models that enables it to relate sensor measurements to the state variables:

$$p(x_t, y_t | z_t) \tag{7}$$

One of the key assumptions underlying our work is that knowledge of the robot's pose x_t renders the discrete state variables y_k conditionally independent, for $k = 1, \ldots, K$. Put differently, if some oracle informed us of the correct pose x_t, we could decompose the sensor data in ways that we can independently infer the individual discrete variables y_k:

$$p(y_t | x_t, z_t) \quad = \quad \prod_k p(y_{k,t} | x_t, z_t) \tag{8}$$

This important assumption is the assumption in the door state tracking application that originally motivated this research. It is also a common assumption underlying the vast field of occupancy grid mapping [7, 14], where it makes it possible to update grid cells independently of each other. However, situations exist where this assumption is violated—those will be outside the scope of this paper.

We are now ready to formulate the basic problem addressed in this paper. The problem of estimating the environment state y_t and the robot pose x_t is the problem of estimating the joint posterior

$$p(y_t, x_t | z^t, u^t) \tag{9}$$

Unfortunately, the "size" of the posterior state space scales exponentially in the number of state variables K. Furthermore, since the robot's pose is not known, the state variable estimates depend on each other. This suggest that the problem of calculating (9) requires time exponential in K—which is indeed the case for the full Bayesian solution to this problem.

3 Factoring The Posterior

The key idea for devising an efficient algorithm for estimating all state variables is to represent the posterior in a factored form—an idea closely related to Rao Blackwellized filters, as presented in [6]. Unfortunately, the posterior (9) cannot easily be factored (see, for example, the discussion in [4]). Therefore, our approach estimates a different but related posterior, which is the joint posterior over the environment state and robot paths x^t (not just poses x_t):

$$p(y_t, x^t | z^t, u^t) \tag{10}$$

While such an estimation seems wasteful—after all, the space of robot paths is much larger than that of momentary robot poses—it allows us to arrive at a factored representation:

$$p(y_t, x^t | z^t, u^t) = p(x^t | z^t, u^t) \prod_k p(y_{k,t} | x^t, z^t, u^t) \tag{11}$$

As shown further below, both types factors, $p(x^t | z^t, u^t)$ and $p(y_{k,t} | x^t, z^t, u^t)$, can be approximated highly efficiently.

The correctness of the factorization (11) is easily established. We notice that Bayes rule implies:

$$p(y_t, x^t | z^t, u^t) = \eta \, p(z_t | y_t, x^t, z^{t-1}, u^t) p(y_t, x^t | z^{t-1}, u^t) \quad (12)$$

Here η is a normalization constant. Exploiting the Markov property, we can transform this as follows:

$$= \eta \, p(z_t | x_t, y_t) \; p(x^t, y_t | z^{t-1}, u^t) \quad (13)$$

We can further decompose the last factor:

$$p(y_t, x^t | z^{t-1}, u^t) = p(y_t | x^t, z^{t-1}, u^t) \; p(x^t | z^{t-1}, u^t) \quad (14)$$

where

$$\begin{aligned} & p(y_t | x^t, z^{t-1}, u^t) \quad (15) \\ &= \sum_{y_{t-1}} p(y_t | x^t, y_{t-1}, z^{t-1}, u^t) \; p(y_{t-1} | x^t, z^{t-1}, u^t) \\ &= \sum_{y_{t-1}} p(y_t | y_{t-1}) \; p(y_{t-1} | x^{t-1}, z^{t-1}, u^{t-1}) \\ &= \sum_{y_{t-1}} \prod_k p(y_{k,t} | y_{k,t-1}) \; p(y_{k,t-1} | x^{t-1}, z^{t-1}, u^{t-1}) \\ &= \prod_k \sum_{y_{k,t-1}=0}^{1} p(y_{k,t} | y_{k,t-1}) \; p(y_{k,t-1} | x^{t-1}, z^{t-1}, u^{t-1}) \end{aligned}$$

The factorization (product) in (15) is simply a consequence of the assumption that the different environment state variables change independently of each other, with probability $p(y_{k,t} | y_{k,t-1})$. The derivation also exploits the Markovness of the environment.

The remaining factor in the right-hand side of (14) is calculated as follows:

$$\begin{aligned} p(x^t | z^{t-1}, u^t) &= p(x_t | x^{t-1}, z^{t-1}, u^t) \; p(x^{t-1} | z^{t-1}, u^t) \\ &= p(x_t | x_{t-1}, u_t) \; p(x^{t-1} | z^{t-1}, u^{t-1}) \quad (16) \end{aligned}$$

Here the second transformation exploits the Markov property.

Finally, the term $p(z_t | x_t, y_t)$ in (13) is obtained as follows:

$$p(z_t | x_t, y_t) = \frac{p(x_t, y_t | z_t) \; p(z_t)}{p(y_t, x_t)} \quad (17)$$

We can safely assume that in the absence of any data, all state variables are equally likely. Furthermore, $p(z)$ is constant relative to our posterior estimation problem. Hence we can subsume several terms in (17) into a constant factor η' and obtain:

$$\begin{aligned} p(z_t | x_t, y_t) &= \eta' \; p(x_t, y_t | z_t) \\ &= \eta' \; p(x_t | z_t) \; p(y_t | x_t, z_t) \\ &= \eta' \; p(x_t | z_t) \prod_k p(y_{k,t} | x_t, z_t) \quad (18) \end{aligned}$$

The last transformation exploited (8).

Putting everything together—Equations(15), (16), (18) back into (14) and then (13)—we obtain the desired factored form (11):

$$p(y_t, x^t | z^t, u^t) = p(x^t | z^t, u^t) \prod_k p(y_{k,t} | x^t, z^t, u^t) \quad (19)$$

with

$$p(x^t | z^t, u^t) = \eta'' p(x_t | z_t) \; p(x_t | x_{t-1}, u_t) \; p(x^{t-1} | z^{t-1}, u^{t-1}) \quad (20)$$

and

$$p(y_{k,t} | x^t, z^t, u^t) = \eta''' p(y_{k,t} | x_t, z_t) \sum_{y_{k,t-1}} p(y_{k,t} | y_{k,t-1}) \; p(y_{k,t-1} | x^{t-1}, z^{t-1}, u^{t-1}) \quad (21)$$

The two rightmost terms in (20) and (21), respectively, constitute the estimate at time $t-1$, which we can assume to be factored by induction. All other terms are part of the dynamics and measurement models of the robot.

4 Exact Bayes Filtering for the Binary Environment Variables

One of the nice features of our factored representation is that the conditional posterior over the discrete variables $p(y_{k,t} | x^t, z^t, u^t)$ can be calculated in closed form. Since each $y_{k,t}$ is a binary variable in $\{0, 1\}$, the posterior is a single numerical probability defined as $\pi_{k,t}$:

$$\pi_{k,t} = p(y_{k,t} = 1 | x^t, z^t, u^t) \quad (22)$$

This definition enables us to write (21) as follows—here with the normalizer spelled out:

$$\pi_{k,t} = \frac{p(y_{k,t} = 1 | x_t, z_t) \; p(z_t | x^t)}{p(y_{k,t} = 1) \; p(z_t | x^t, z^{t-1}, u^t)} \; \pi^+_{k,t} \quad (23)$$

where $p(y_{k,t})$ is a prior on the state variable $y_{k,t}$, and $\pi^+_{k,t}$ is defined as follows:

$$\begin{aligned} \pi^+_{k,t} &= p(y_{k,t} = 1 | y_{k,t-1} = 1) \pi_{k,t-1} \\ &\quad + p(y_{k,t} = 1 | y_{k,t-1} = 0)[1 - \pi_{k,t-1}] \quad (24) \end{aligned}$$

The denominator $p(z_t | x^t, z^{t-1}, u^t)$ can be eliminated analytically since (23) is easily posed for the opposite event, $y_{k,t} = 0$. The probability of this event is $1 - \pi_{k,t}$:

$$1 - \pi_{k,t} = \frac{p(y_{k,t} = 0 | x_t, z_t) \; p(z_t | x^t)}{p(y_{k,t} = 0) \; p(z_t | x^t, z^{t-1}, u^t)} \; \pi^-_{k,t} \quad (25)$$

with

$$\begin{aligned} \pi^-_{k,t} &= p(y_{k,t} = 0 | y_{k,t-1} = 1) \pi_{k,t-1} \\ &\quad + p(y_{k,t} = 0 | y_{k,t-1} = 0)(1 - \pi_{k,t-1}) \quad (26) \end{aligned}$$

Dividing (23) by (25) gives us the following quotient, commonly referred to as *odds ratio*:

$$\frac{\pi_{k,t}}{1 - \pi_{k,t}} = \frac{p(y_{k,t} = 1 | x_t, z_t)}{1 - p(y_{k,t} = 1 | x_t, z_t)} \frac{1 - p(y_{k,t} = 1)}{p(y_{k,t} = 1)} \frac{\pi^+_{k,t}}{\pi^-_{k,t}} \quad (27)$$

which is easily calculated from the discrete state transition probability $p(y_{k,t} | y_{k,t-1})$ and the previous estimate $\pi_{k,t-1}$. The desired probability $\pi_{k,t}$ is easily recovered from the quotient in (27) by virtue of the following equality:

$$\pi_{k,t} = 1 - \left(1 + \frac{\pi_{k,t}}{1 - \pi_{k,t}}\right)^{-1} \quad (28)$$

Thus, albeit somewhat complex mathematically, the individual posteriors over the binary state variables can be calculated exactly in closed form.

5 Particle Filtering for the Continuous Variables

Unfortunately, the continuous posterior $p(x^t|z^t, u^t)$ over robot paths in (20) cannot be computed exactly, since robot motion equations are highly non-linear. Our approach, thus, approximates $p(x^t|z^t, u^t)$ using particle filters. Particle filters are Monte Carlo approximation that have been extremely popular in recent years [5, 10]. They are usually applied for estimating posteriors over states—but they can equally be thought of sampling over entire state trajectories. This, and their ability to handle non-linear dynamics, makes them ideally suited for the estimation of the posterior $p(x^t|z^t, u^t)$.

Particle filters incrementally generate particles in state space. Since in our formulation, the posterior of the discrete variables is conditioned on the continuous variables, each particle is of the form:

$$\left\langle x^{t[n]}, \pi_{1,t}^{[n]}, \pi_{2,t}^{[n]}, \ldots, \pi_{K,t}^{[n]} \right\rangle \qquad (29)$$

The total number of particles will be denoted by N, and the set of particles at time t will be denoted Y_t. Particles in Y_t are asymptotically distributed according to $p(x^t|z^t, u^t)$ for $N \longrightarrow \infty$.

This is achieved by the following sampling procedure. By assumption, the $(t-1)$th particle set Y_{t-1} is distributed according to $p(x_{t-1}|z^{t-1}, u^{t-1})$. Thus, if we draw a particle at random we obtain

$$\langle x_{t-1}^{[n]}, \pi_{1,t-1}^{[n]}, \ldots, \pi_{K,t-1}^{[n]} \rangle \sim p(x_{t-1}|z^{t-1}, u^{t-1}) \qquad (30)$$

where the superscript $^{[n]}$ indicates that the lefthand side is a particle (the n-th of N particles). Furthermore, if we use this particle to draw

$$x_t^{[n]} \quad \sim \quad p(x_t|x_{t-1}^{[n]}, u_t) \qquad (31)$$

this particle is distributed according to

$$p(x_t|x_{t-1}, u_t)\ p(x_{t-1}|z^{t-1}, u^{t-1}) \qquad (32)$$

To account for the mismatch between this so-called *proposal distribution* and the target distribution $p(x_t|z^t, u^t)$, particles are now resampled. In particular, the (non-normalized) *importance weight* is given by the quotient of the target posterior distribution and the proposal distribution:

$$\begin{aligned} w_t^{[n]} &= \frac{p(x_t^{[n]}|z^t, u^t)}{p(x_t^{[n]}|x_{t-1}, u_t)\ p(x_{t-1}|z^{t-1}, u^{t-1})} \qquad (33) \\ &= [\ldots] \\ &= p(z_t|x_t^{[n]}) \prod_{k=1}^{K} \left\{ \sum_{y_{k,t}} \frac{p(y_{k,t}|x_t^{[n]}, z_t)}{p(y_{k,t})}\ \pi_{k,t}^{[n]+} \right\} \end{aligned}$$

The derivation of this equation is mostly analogous to our derivation of the factored posterior. The importance factor $w_t^{[n]}$ is used to resample particles, resulting in a sample that converges to the posterior $p(x^t|z^t, u^t)$ as the sample size N approaches infinity.

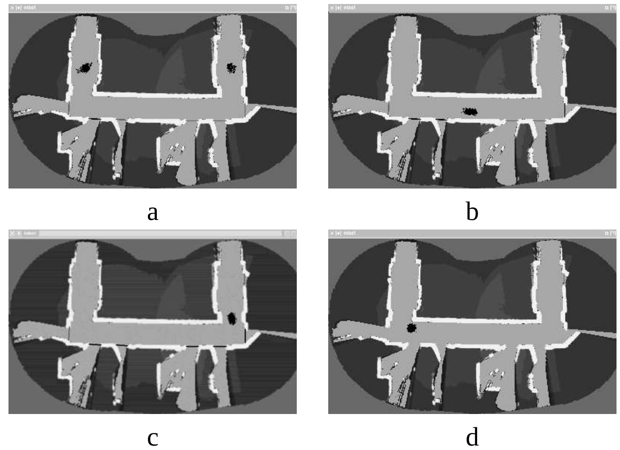

Figure 1: *Experimental Results*

6 Experimental Results

Systematic experimental results were obtained in the domain of robot localization in dynamic environments. The goal of our experiments was to establish evidence that the algorithm can effectively localize mobile robots in environments where doors change their status in unpredictable ways. Furthermore, we were interested in evaluating the algorithm's ability to estimate the state of the various doors. Comparing the results of localization using a static world particle filter localizer (Monte Carlo Localization [3]), a particle filter with door position knowledge but no belief state maintenance for the doors, and our hybrid filter algorithm, we find that our filter gives the best results in dynamic environments with doors.

Experiments were carried out using an RWI Pioneer 1 robot, equipped with a laser range finder. The robot navigated within the region mapped in Fig. 1. Starting in the lefthand corridor, the robot moved to the righthand corridor, then returned to its point of origin. All doors were closed in the robot's first pass and were open in the return pass.

In Fig. 1, the color of line segments in the door locations indicate the dominant belief about the state of doors. Black indicates $> 65\%$ closed likelihood, grey indicates within ± 15% of 50% open likelihood, and blank indicates $> 65\%$ open likelihood.

Ambiguity in the initial data resulted in convergence on two locations(Fig. 1a). The dominant belief was that door 2 (from the left) was closed. The other doors were unobserved and had the default 50% open likelihood.

In Fig. 1b, the robot has correctly localized and has observed doors 2, 3, and 6 to be closed. In Fig. 1c, the robot has correctly observed that doors 2 through 6 are closed. The doors were subsequently opened, and Fig. 1d shows that the robot has correctly updated its belief of the door state as it returned to the lefthand corridor.

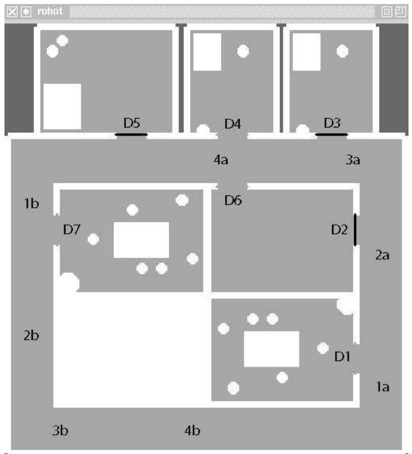

Figure 2: *Simulation*

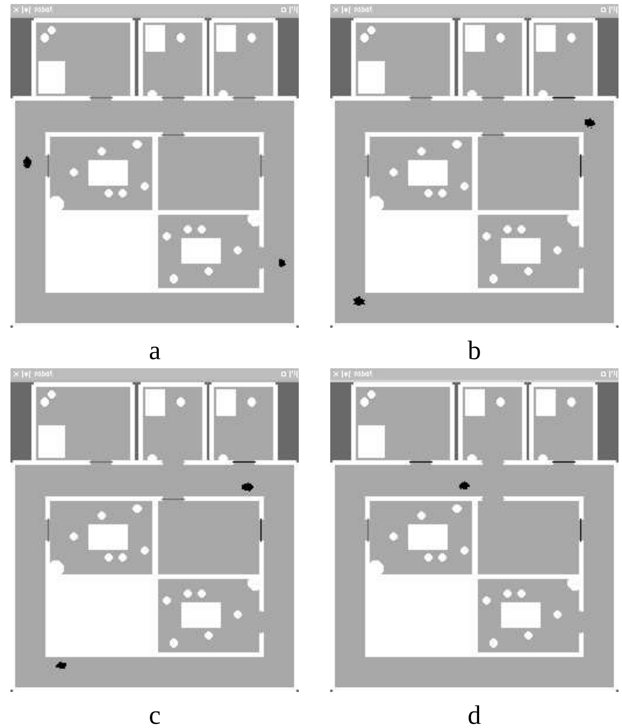

Figure 3: *Hybrid Filter Performance*

In order to more rigorously evaluate the effectiveness of door belief maintenance, we used a simulation to generate a scenario with more challenging ambiguities. In the simulation, we presented the robot with two nearly identical positions. Fig. 2 depicts the simulated environment, where the robot began in the lower right corner and followed a path through positions 1a to 4a. Closed door locations are marked with black lines. Since doors D2 and D3 were closed, the sensor data showed an unbroken wall, creating an ambiguity between the true path and the path through 1b to 4b. The robot's environmental map indicated that all doors were open.

In this environment, the hybrid algorithm was effective in localizing and estimating the door states. Fig. 3 shows the localization sequence using the hybrid algorithm. The particle filter initially maintained two conjectures (Fig. 3a). The door colors indicate the dominant belief as the robot moves through the corridor. In Fig. 3d, the robot has correctly localized upon observation of open doors D4 and D6.

These results were compared with results from a static world particle filter localizer (Monte Carlo Localization [3]) and a particle filter with door position knowledge but no belief maintenance for the doors. The latter is straightforward extension of MCL that does take door uncertainty in account is to assume 50% chance that the door is open, *independently* for each sensor measurement. This approach is very similar to MCL, and equivalent to our filter under the assumptions that door states are entirely random (no persistence).

When faced with the simulation environment, the static world particle filter localizer also initially converged upon two conjectures at 1a and 1b (Fig. 2). The particle filter used the open state of door 2 in its environment map as evidence that the righthand conjecture was less probable. Consequently, the correct conjecture was eliminated by the time the robot reached position 2a (Fig. 2), causing localization to fail.

The algorithm with door position knowledge but no belief maintenance failed similarly to the static world particle filter localizer. The 50% consideration that door 2 was closed improved the score of the particles at 2a (Fig. 2). However, they still scored lower than the particles at 2b (Fig. 2), which saw a solid wall instead of a door that had a 50% chance of being open.

The simulation clearly demonstrated a scenario in which the hybrid algorithm succeeded while a particle filter without door state considerations or one without belief updating failed.

The histogram in Fig. 4 shows the progression of belief about the state of door 1 that corresponds to the simulation sequence using the hybrid algorithm. The horizontal axis of each graph partitions the possible probability of door 1 being open into 0.01 unit intervals. The vertical measurement for each interval shows the combined weight of the particles that have a belief for door 1 within that interval. This provides an aggregate view of the set of beliefs for a particular door at a particular timestep.

Fig. 4a (t = 0 sec) shows the initial belief, where all particles have a 0.5 probability that door 1 is open. Fig. 4b (t = 25 sec) shows the belief distribution after about 30% of the particles have had some observations of door 1 as part of the righthand conjecture. The other 70% of the particles are in the lefthand conjecture and have not made an observation of door 1. As a result, the figure shows a smaller spike around 0.5 probability and a small distribution of beliefs gradually increasing towards 1.

In Fig. 4c (t = 60 sec), nearly all the particles in the right-

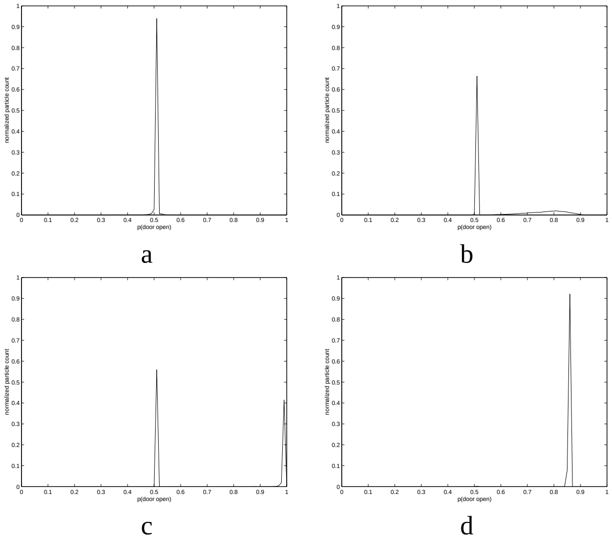

Figure 4: *Histogram of the Belief State of Door 1*

hand conjecture have a belief near 1. About 50% of the particles remain in the lefthand conjecture and still maintain a belief around 0.5 probability.

Finally, Fig. 4d (t = 220 sec) shows a timestep where the particles have converged to the correct location, and the robot has not observed door 1 for some time. The belief is near 0.8. As time progresses without more observations of door 1, the belief degrades towards 0.5, reflecting the growing uncertainty of that door's state.

7 Conclusion

This paper proposed an algorithm for mobile robot localization in environments with dynamic binary states. In its most simple formulation, the localization problem in such dynamic environment requires computation exponential in the number of state variables. Our approach exploited a natural conditional independence in the domain, to arrive at a factored posterior. This posterior was computed via a combination of particle filters and binary Bayes filters, where binary Bayes filters were attached to the particle filters. The resulting algorithm is linear in the number of environment variables. We recently developed an efficient data structure that makes it feasible to compute the posterior in logarithmic time; however, at the time of submission, this algorithm was not fully implemented, and hence is not included in this paper.

Experiments demonstrated that the hybrid filter algorithm successfully localizes robots in situations in which the original particle filter localizer failed due to inconsistencies between the robot map and the state of the world. Our experiments also illustrate that the explicit estimation of the environment state improves the overall localization accuracy

Future work includes the extension to multi-robot localization problems and the implementation of more efficient data structures for logarithmic update time.

References

[1] J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. A. K. Peters, Ltd., Wellesley, MA, 1996.

[2] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2):3–55, 1999.

[3] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999.

[4] G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localisation and map building (SLAM) problem. *IEEE Transactions of Robotics and Automation*, 2001. In Press.

[5] A. Doucet, J.F.G. de Freitas, and N.J. Gordon, editors. *Sequential Monte Carlo Methods In Practice*. Springer Verlag, New York, 2001.

[6] A Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 176–183, Stanford, 2000.

[7] A. Elfes. *Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation*. PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1989.

[8] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427, 1999.

[9] D. Fox, S. Thrun, W. Burgard, and F. Dellaert. Particle filters for mobile robot localization. In A. Doucet, N. de Freitas, and Gordon. N., editors, *Sequential Monte Carlo Methods in Practice*, pages 499–516. Springer Verlag, 2001.

[10] M. Isard and A. Blake. CONDENSATION: conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[11] K. Konolige and K. Chou. Markov localization using correlation. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Stockholm, Sweden, 1999. IJCAI.

[12] D. Kortenkamp, R.P. Bonasso, and R. Murphy, editors. *AI-based Mobile Robots: Case studies of successful robot systems*, Cambridge, MA, 1998. MIT Press.

[13] J.J. Leonard and H.J.S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In J. Hollerbach and D. Koditschek, editors, *Proceedings of the Ninth International Symposium on Robotics Research*, Salt Lake City, Utah, 1999.

[14] H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74, 1988.

[15] K. Murphy and S. Russell. Rao-blackwellized particle filtering for dynamic bayesian networks. In A. Doucet, N. de Freitas, and Gordon. N., editors, *Sequential Monte Carlo Methods in Practice*, pages 499–516. Springer Verlag, 2001.

[16] S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363, 2001.