2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Rao-Blackwellized Particle Filtering for Mapping Dynamic Environments

Isaac Miller and Mark Campbell
Sibley School of Mechanical and Aerospace Engineering
Cornell University, Ithaca NY, 14853
{itm2, mc288}@cornell.edu

Abstract— **A general method for mapping dynamic environments using a Rao-Blackwellized particle filter is presented. The algorithm rigorously addresses both data association and target tracking in a single unified estimator. The algorithm relies on a Bayesian factorization to separate the posterior into 1) a data association problem solved via particle filter and 2) a tracking problem with known data associations solved by Kalman filters developed specifically for the ground robot environment. The algorithm is demonstrated in simulation and validated in the real world with laser range data, showing its practical applicability in simultaneously resolving data association ambiguities and tracking moving objects.**

I. INTRODUCTION

Many robotic sensing and mapping algorithms exist to estimate a static environment in a probabilistically rigorous Bayesian framework, the most popular being the probabilistic approach to the Simultaneous Localization And Mapping (SLAM) problem [1]. No completely general equivalent yet exists for the dynamic environment mapping problem, however, where environment objects are not assumed static. The SLAM with Detection and Tracking of Moving Objects (DATMO) approach presented in Ref. [2] is an attempt at such a general framework, though it does so under a restricting independence assumption between static and dynamic obstacles. Its main drawback, however, is that it does not address the data association problem within the Bayesian framework, opting instead to assign measurements to objects based on experimental feature matching criteria. Other common approaches to this assignment problem include scan registration techniques, which match raw measurement data to estimated occupancy grids, and maximum likelihood association, which assigns measurements to maximize measurement likelihood [3], [4], [5]. These techniques work well in practice, but either assume a known number of dynamic objects, a known static environment, or do not concern themselves with dynamic objects at all. In many of these and related algorithms, the primary focus is to estimate robot position and the static (or slowly changing) background map [3], [5], [6].

On the other end of the spectrum, a multitude of target tracking techniques have been developed to track small numbers of moving objects [7]. Work in this field typically concerns itself with data association, the task of determining which sensor measurements belong to which targets. These techniques have progressed from feature based assignments to statistical techniques such as multiple hypothesis tracking and simultaneous estimation of data association and dynamic target states. Most recently, Monte Carlo and Bayesian factorization based data associations have also become a viable alternative [8].

This paper combines techniques from both static robotic mapping and target tracking to develop a general, rigorous Bayesian framework for mapping a dynamic environment. Here, a Rao-Blackwellized Particle Filter is developed to estimate data associations and obstacle states simultaneously. This results in a particle-based tracking algorithm with goals similar to the people tracking system presented in Ref. [9], though the algorithms are fundamentally different. In particular, the approach here purports that particles are most efficiently used to represent discrete data association variables, with traditional Kalman Filters used for tracking continuous obstacle states. This approach is exactly the opposite of that used in Ref. [9], which utilizes a Joint Probabilistic Data Association Filter to estimate data assignments and particles to represent target states. The approach here has extremely low particle requirements even for complex dynamic scenes, and it makes intuitive discrete data assignment decisions. In addition, the number of obstacles in the mapped environment is determined automatically without the need for adding extra states or particles to the estimator.

Throughout the paper, emphasis is placed on mapping an environment containing obstacles with fast dynamics as well as those with traditional static features. The system is discussed in the context of mapping relative to the local sensing platform for obstacle avoidance, though robot position states could be added if a global map were desired instead. Section II gives a technique to jointly estimate data associations and obstacles, relying on a Bayesian factorization of the posterior to make the problem tractable. Section III discusses object models and estimators for effective and efficient dynamic mapping. Section IV demonstrates the effectiveness of the generalized framework in mapping a simulated dynamic environment. Section V validates the algorithm using roadside data collected with a laser rangefinder against moving traffic. Section VI provides a summary and conclusion of the work.

I. Miller is a Graduate Research Fellow at the Sibley School of Mechanical and Aerospace Engineering, Cornell University

M. Campbell is a Senior Member of AIAA and Associate Professor at the Sibley School of Mechanical and Aerospace Engineering, Cornell University

II. RAO-BLACKWELLIZED PARTICLE FILTER FOR SIMULTANEOUS DATA ASSOCIATION AND DYNAMIC ESTIMATION

The proposed dynamic environment mapping problem begins as a standard filtering problem by identifying the desired posterior to be estimated. For this problem, two quantities are of interest: discrete data assignments or associations N, and obstacle or object states O. The data associations are indicator variables connecting each measurement to its obstacle of origin, which is generally uncertain. It is then desired to estimate the joint posterior probability distribution over these two quantities, or:

$$p\left(N_k, O_k | Z_k\right) \tag{1}$$

where Z_k are the set of observed sensor measurements, and the subscript k represents an integer time index. The use of capital letters N_k, O_k, and Z_k indicates a full time history of these quantities, from the filter's inception at time 0 to the present time k.

Although well-known methods exist to solve such a filtering problem, the particular case of dynamic obstacle mapping on a potentially mobile robot presents several difficulties. First, the distribution of equation 1 is hybrid, because data associations N are discrete while obstacle states O are continuous. Second, the large number of obstacles and sensor measurements curse the problem with an exponentially large number of data association permutations for a modestly small number of obstacles and sensor measurements. A final challenge is the fact that in a dynamic environment, the obstacles often exhibit nonlinear dynamics in time. These three challenges in concert make most traditional estimation approaches infeasible for the dynamic environment mapping problem.

Rather than apply filtering techniques directly to the posterior, the posterior will first be factorized into more manageable terms. To that end, the definition of conditional probability is applied to equation 1 to rewrite the posterior:

$$p\left(N_k, O_k | Z_k\right) = p\left(N_k | Z_k\right) \cdot p\left(O_k | Z_k, N_k\right) \tag{2}$$

This factorization is exact, and is similar to one made in Ref. [10] for use on robot positions and static obstacles in a variant of the SLAM problem. Closer inspection reveals that the $p\left(O_k | Z_k, N_k\right)$ term in equation 2, which describes the probability distribution of obstacle states given measurements and known data associations, is the output of a traditional tracking algorithm such as a Kalman Filter applied with known measurement correspondences. The $p\left(N_k | Z_k\right)$ term in equation 2, which is a distribution over the history of data associations given measurements, remains unknown.

Despite the factorization, the unknown posterior $p\left(N_k | Z_k\right)$ in general still represents too many data association permutations to be estimated in real-time with closed form filtering techniques. Instead, this portion of the posterior is approximated with a small number of Monte Carlo sampled data associations, similar to the implementation of a particle filter; see, for example, Refs. [11] and [12]. This type of 'reduced state' particle filter, when coupled with a closed form filter to estimate a joint distribution, is known as a Rao-Blackwellized Particle Filter (RBPF); see, for example, Ref. [13]. In the present application, the RBPF is used to separate target tracking from data association in a computationally feasible manner. Such a filter has been derived previously in Ref. [8] in the context of general target tracking, where it is shown to be robust against target confusion and other classically challenging data association problems. In the present study, the RBPF data association filter will be adapted for estimating a map of dynamic obstacles relative to a ground-based sensing platform, i.e. a ground robot.

Following the particle filter derived in Ref. [12] for data association, each sampled particle acts as a potential data association history drawn according to a 'proposal distribution' $q\left(N_k | Z_k\right)$. These particles are used to reconstruct and approximate the desired posterior $p\left(N_k | Z_k\right)$ over the data association history:

$$p\left(N_k | Z_k\right) \approx \sum_i w_k^i \cdot \delta\left(N - N_k^i\right) \tag{3}$$

where w_k^i is the likelihood weight of the i^{th} particle N_k^i at time index k:

$$w_k^i = \frac{p\left(N_k^i | Z_k\right)}{q\left(N_k^i | Z_k\right)}, \sum_i w_k^i = 1 \tag{4}$$

and $\delta\left(\cdot\right)$ is the Kronecker delta function.

Applying the particle filter approximation of equation 3 to the data association portion of the dynamic environment mapping problem in equation 2, the estimate of the full posterior $p\left(N_k, O_k | Z_k\right)$ is written as

$$p\left(N_k, O_k | Z_k\right) \approx \sum_i w_k^i \cdot \delta\left(N - N_k^i\right) \cdot p\left(O_k | Z_k, N_k^i\right) \tag{5}$$

where the posterior obstacle distributions $p\left(O_k | Z_k, N_k^i\right)$ are now conditioned on the specific data association history N_k^i of a particular particle. When conditioned on a specific data association history, these posterior obstacle distributions are closed form outputs of the tracking filters. In essence, the closed form posterior of the obstacles has been exploited so that particles need only be sampled over a portion of the joint distribution of N_k and O_k. Each particle then represents a particular data association history, plus the obstacle states pertaining to that history.

Most generally, the proposal distribution $q\left(N_k^i | Z_k\right)$ used to sample the particles is a design choice, though ideally it is 'similar to' the true distribution $p\left(N_k | Z_k\right)$. In particular, a judiciously-chosen factorized form of q will yield a recursive weight update [12]:

$$w_k^i = w_{k-1}^i \cdot \frac{p\left(z_k | Z_{k-1}, N_k^i\right) \cdot p\left(n_k^i | Z_{k-1}, N_{k-1}^i\right)}{q\left(n_k^i | Z_k, N_{k-1}^i\right)} \tag{6}$$

where lowercase variables z_k and n_k indicate measurements and data associations at a particular time index k. Note equation 6 has been obtained, up to a normalization constant in the weights, via straightforward application of Bayes' rule

to the posterior $p\left(N_k|Z_k\right)$. Both terms in the numerator of equation 6 are straightforward to evaluate: $p\left(z_k|Z_{k-1}, N_k^i\right)$ is the filter likelihood of the obstacle to which the measurement is assigned, and $p\left(n_k^i|Z_{k-1}, N_{k-1}^i\right)$ represents the data associations' *a priori* transition model [14].

One final step remains in deriving the particle filter for data association, and that is choosing the proposal distribution $q\left(n_k^i|Z_k, N_{k-1}^i\right)$. This distribution defines how new particles are sampled, so it must necessarily be a distribution that can be sampled efficiently and accurately. In traditional particle filtering, the distribution is often chosen based only on the system's transition model and its process noise, ignoring the measurement history. This choice is suboptimal in the sense that it does not minimize the sample variance on the particles' weights w_k^i, and the resulting filter may require more frequent resampling and larger numbers of particles. Nonetheless, the suboptimal proposal distribution is often made due to the infeasibility of sampling from the optimal one: $q\left(n_k^i|Z_k, N_{k-1}^i\right) = \alpha_k^i \cdot p\left(n_k^i|Z_k, N_{k-1}^i\right)$ [12]. Note the explicit inclusion of the normalization constant α_k^i to make clear that the proposal distribution must sum to unity across the set of available data associations.

In the special case of particle filtering for data association, the optimal proposal distribution may be sampled directly [8]. To see this, consider first rewriting the optimal distribution using Bayes' rule:

$$\begin{aligned} p\left(n_k^i|Z_k, N_{k-1}^i\right) &= \alpha_k^i \cdot p\left(z_k|n_k^i, Z_{k-1}, N_{k-1}^i\right) \\ &\cdot \quad p\left(n_k^i|Z_{k-1}, N_{k-1}^i\right) \end{aligned} \tag{7}$$

where α_k^i is the particle-specific normalizing constant ensuring the data association probabilities for the obstacles of the i^{th} particle sum to unity. Of the remaining two terms, the first is the measurement likelihood of a particular obstacle, which is often and typically accurately assumed to be Gaussian [14]. The second is a data association transition model, which is here assumed uniform to represent a lack of knowledge about which obstacle will next generate a measurement. Equation 7 allows data associations for a particular measurement to be sampled with a probability weighted by the measurement likelihood, as defined by the closed form obstacle filter within the i^{th} particle.

Substituting the optimal proposal distribution from equation 7 into equation 6 yields a cancelation, resulting in the final form of the weight update:

$$w_k^i = w_{k-1}^i \cdot \frac{1}{\alpha_k^i} \tag{8}$$

The remaining constant α_k^i is left over from the likelihood ratio computation. In the specific case of a uniform transition probability for data associations, it evaluates to:

$$\alpha_k^i = \left[\frac{1}{M} \cdot \sum_{m=1}^{M} p\left(z_k|n_{m,k}^i\right)\right]^{-1} \tag{9}$$

where the sum is performed across all M obstacles stored within the i^{th} particle, and $n_{m,k}^i$ is the event that the measurement z_k taken at time k is assigned to the m^{th} of M obstacles in the i^{th} particle. This final form of the weight update has the satisfying interpretation that particles whose obstacles match the sensor measurements more closely have higher weights.

The sampling and reweighting calculations complete the data association particle filter defined by equation 3 and the RBPF defined by equation 5. General steps for running the RBPF for data association are given below:

1) Draw an initial set of particles N_0^i, $\forall i$.
2) Predict all obstacles in each particle forward in time to the next measurement to yield either $p\left(O_k|Z_{k-1}, N_{k-1}^i\right)$ or a sufficient set of parameters to represent that distribution.
3) For each particle, pick a data association for the measurement using the proposal distribution in equation 7.
4) Update each particle's associated obstacle with the measurement to yield $p\left(O_k|Z_k, N_k^i\right)$.
5) Update particle weights according to equation 8.
6) Resample particles to keep the filter well-conditioned, if necessary. Effective resampling strategies are discussed in Refs. [12] and [15], but are not addressed here.
7) Go to step 2.

III. OBJECT MODELS FOR GROUND-BASED DYNAMIC OBSTACLE ESTIMATION

The RBPF discussed in section II is a powerful tool for the multitarget tracking problem because it runs a complex particle filter over only the data associations, which for even modest environments cannot feasibly be estimated using a traditional closed-form filter. In this context, the RBPF's efficiency over an exhaustive hypothesis tracking scheme such as those presented in Ref. [7] depends on data associations being 'obvious,' so that a relatively small number of particles are sufficient to cover the most likely data association histories. While the optimal proposal distribution certainly helps to keep particle requirements more manageable, a good obstacle model is required to make objects more distinguishable for data association. In this sense, the obstacle model defines and specializes the RBPF data association algorithm for each application. The obstacle model must be defined with target separability and distinguishability in mind if the filter is to be tractable in real-time.

In the present ground-based implementation of the RBPF for dynamic environment mapping, the environment is divided into two types of objects: mature obstacles and the birth obstacle. Mature obstacles are persistent features of the environment. In other words, the set of mature obstacles within a particular particle comprises that particle's estimate of the dynamic environment. The birth obstacle, in contrast, represents any newly-observed feature of the environment. A measurement assigned to the birth obstacle within a particle represents that particle's belief that the measurement belongs to a new feature of the environment rather than to any existing mature objects. Other obstacle models may also be added to enrich the description of the environment estimated

by the RBPF. Such models may include (but certainly are not limited to) a parameterized estimate of the ground itself, a distinction between static and moving mature obstacles, a distinction between types of static and moving mature obstacles, and even a model for false sensor detections. The inclusion or exclusion of such models is entirely problem specific, and all can be incorporated into the general RBPF framework proposed here. At a minimum, however, the filter must include a mature obstacle model to describe the occupancy of the environment, and a birth model to describe how newly-observed obstacles enter into the RBPF's dynamic map estimate.

A. Mature Obstacle Model

Perhaps the most challenging aspect of dynamic environment mapping lies in reducing the complicated dynamics of the environment as a whole into a few very simple maneuvers performed by objects relevant to the sensing platform. Several very general dynamics models exist for such objects, including constant velocity models, constant acceleration models, and even more complicated maneuver-based mode switching models [2], [14]. Because the goal in modeling such dynamic objects is to make them distinguishable from each other for improved data association, the dynamic obstacle models should be as specific as possible for each particular application. The decision merits caution, however, because a tradeoff exists between model complexity and estimation accuracy. While the dynamics model may become arbitrarily complex with a large number of internal states and possibly large numbers of discrete model options, typically the number and type of sensor measurements does not change. Estimation accuracy may therefore decrease due to the difficulty of estimating higher order dynamical states from a fixed number of sensor measurements.

The model used here is that of a typical wheeled vehicle moving under no-slip conditions: a constant speed, constant curvature model. The model is more complex than a constant velocity model, though it expresses an important motion constraint for four-wheeled full-sized ground vehicles. The dynamics of such an object expressed in platform-centric coordinates are:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{s} \\ \dot{\psi} \\ \dot{\gamma} \\ \dot{l} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} s \cdot \cos(\psi) - v_x + \omega_z \cdot y \\ s \cdot \sin(\psi) - v_y - \omega_z \cdot x \\ 0 \\ s \cdot \gamma - \omega_z \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{10}$$

where x and y are the obstacle's position relative to the sensing platform, s is the obstacle's ground speed, ψ is the obstacle's heading relative to the platform, γ is the curvature, v_x and v_y are components of the sensing platform's ground velocity, and ω_z is the sensing platform's rotation rate. Note that curvature is used instead of radius of curvature to avoid the computational difficulties of the radius of curvature becoming infinite. Also, the object's ground speed is used instead of speed relative to the sensing platform to simplify the dynamics equations. It is assumed that this the object moves in the plane of the sensing platform, though the same model results under small angle approximations to platform pitch and roll. The model is also augmented with a length parameter l and a width parameter w to account for the fact that obstacles have finite size. These and other geometric parameters are not required, but they can be included to help distinguish objects based on size and shape. In this study, three variants of these parameters are used for different situations: a variant with no length and width (point obstacles), a variant with only a width (circular obstacles), and a variant with both a length and width (rectangular obstacles). In practice, the decision to include these or other geometric parameters will depend on the types of obstacles expected and the types of measurements that can be generated. Length and width are mentioned here because many robotic sensors can generate some type of size measurement in addition to pointwise detections.

Although the dynamics model of equation 10 has been presented for describing a moving obstacle, it is important to realize that it describes static obstacles equally well. Under these circumstances, the filter will simply report a negligible ground speed to indicate the obstacle is not moving. Obstacle heading is meaningless in that circumstance and will therefore not converge in the estimate, but the position and size of the static obstacle will converge with the model in equation 10 and will still be estimated correctly. The proposed dynamic mapping algorithm thus handles both static and dynamic elements of the environment in a true Bayesian sense.

B. Birth Obstacle Model

Unlike the persistent mature obstacle model described in section III-A, the birth obstacle model is transient. When a measurement is assigned to the birth model of a particle, it represents the particle's belief that the measurement does not apply to any existing mature obstacle within the particle's map. Instead, the measurement must be used to create a new mature obstacle to be added to the particle's map. This type of decision is most likely made when new obstacles appear, such as when they first come into sensing range or become unobstructed by other obstacles.

The birth model handles the appearance of new obstacles in a rigorous Bayesian manner. It is a track initialization algorithm, a method for initializing a new mature obstacle from measurements that do not correspond to existing mature obstacles. The key point is that the choice to use a measurement to initialize a new obstacle is made in a non-deterministic manner by each particle, along with every other data association decision made within the RBPF. In this way new obstacles are created naturally within the Bayesian framework of the dynamic mapping algorithm.

Track initialization algorithms vary depending on the sensors and types of obstacles expected. In this study, where position and size measurements are used, obstacles are initialized with measured position and size. The unmeasured

states, speed, heading, and curvature, are initialized with arbitrary guesses and large uncertainty.

IV. DYNAMIC MAPPING IN SIMULATION

In order to gain an intuition of the behaviors of the RBPF dynamic mapping algorithm discussed in sections II and III, its capabilities are validated here via simple planar simulation. To begin, a truth model is created by randomly generating initial states for a pre-specified number of obstacles, both static and dynamic. These obstacles are assumed to follow the dynamics of equation 10, where static obstacles have zero speed, and dynamic obstacles move at a constant speed and curvature. Static obstacles are placed at random locations anywhere within the sensing platform's field of view. Dynamic obstacles all start at the periphery of the sensor's field of view, and are chosen to travel up to $15m/s$ in roughly the same direction to resemble loosely-organized motion. Obstacle sizes are chosen randomly with a uniform distribution up to $3m$, and obstacles are represented as circles to simplify geometrical calculations within the simulation. Figure 1 shows typical obstacle conditions within a simulation, with circles representing the obstacles and arrows indicating their instantaneous headings.

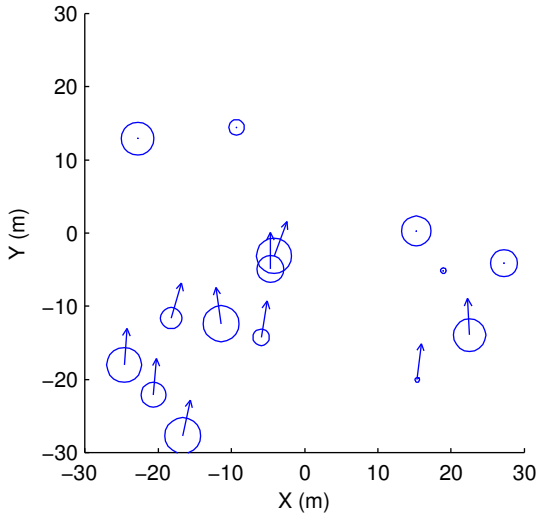

Fig. 1. Static and dynamic obstacles and their headings (arrows) within a typical simulation. The quantity and states of these obstacles are to be estimated by the dynamic mapping algorithm.

Three types of sensor measurements are assumed available for this simulation: x and y measurements of the position of each visible obstacle's center relative to the sensor, and a measurement of each visible obstacle's width. Figure 2 defines these measurements within the simulation. Techniques for producing these position and width measurements from raw sensor data are not addressed here; it is assumed the raw sensor data has already been processed to generate these measurements. Obstacle visibility is determined in simulation by checking whether the ray between the sensor and each obstacle is obstructed by any other object, in order to test the dynamic mapping algorithm under simple occlusion scenarios. The sensor's measurements are also corrupted by additive zero-mean Gaussian white noise, with noise statistics listed where appropriate. Finally, without loss of generality, the sensing platform is assumed stationary.

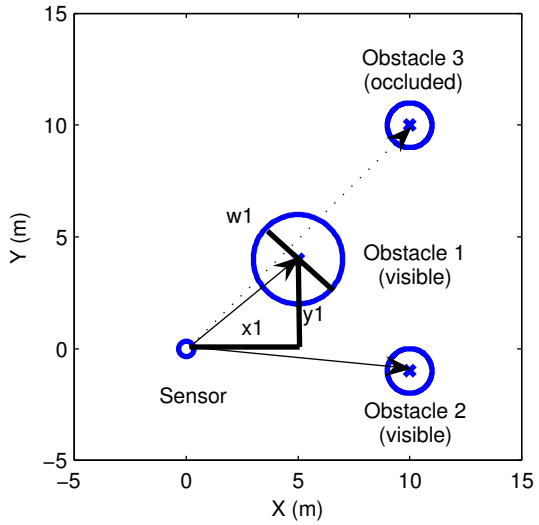

Fig. 2. Definition of measurements within a simulation. Measurements provided within the simulation are center positions x and y, and widths w. Occluded objects do not generate measurements. In this frame, the measurements generated are x_1, y_1, w_1, x_2, y_2, and w_2.

Subsections IV-A and IV-B use this simulation environment to validate the two primary capabilities of the RBPF in the dynamic mapping problem: the ability to determine the number of obstacles, and the ability to track these obstacles correctly.

A. Validating Obstacle Identification And Data Association

The first step in validating the dynamic mapping algorithm is to verify its ability to perform obstacle identification and data association. To perform this verification, 15 obstacles are drawn and simulated for 5 seconds with measurements generated every 0.1 seconds. Measurements are only generated for obstacles that are visible according to the occlusion rules depicted in Figure 2. For those obstacles that are visible, measurements are generated with the following noise parameters:

$$\sigma_x = 0.1m \quad \sigma_y = 0.1m \quad \sigma_w = 0.5m \tag{11}$$

The following process noise parameters are used to drive each mature obstacle Kalman Filter:

$$\begin{array}{ll} \sigma_x = \sigma_y = 3m & \sigma_s = 0.5m/s \\ \sigma_\gamma = 0.01m^{-1} & \sigma_w = 0.01m \end{array} \tag{12}$$

Three particles are used to track these obstacles. Each particle begins with an empty set of mature obstacles, creating a new mature obstacle each time a measurement is assigned to the birth model. When this occurs, the mature obstacle is initialized with the following state:

$$\begin{pmatrix} x_0 & y_0 & s_0 & \psi_0 & \gamma_0 & w_0 \end{pmatrix}^T = \ldots$$

$$\begin{pmatrix} z_x & z_y & 0 & \pi/2 & 0 & z_w \end{pmatrix}^T \quad (13)$$

where z_x, z_y, and z_w are the measured x, y position and width w. This birth model, having no prior knowledge of the location of a new obstacle, is given a uniform measurement likelihood over the sensor's simulated 30m field of view.

Note that when computing the data association probabilities in equation 7, only the measurement likelihood term $p\left(z_k|n_k^i, Z_{k-1}, N_{k-1}^i\right)$ varies from obstacle to obstacle. Furthermore, because each mature obstacle is estimated with a Kalman Filter, this term is just the likelihood of the filter's innovation for that measurement. Also, each obstacle is maintained within a particle so long as its position uncertainty, represented by the square root of the trace of its position estimation covariance submatrix, remains below $1.5m$. In this way, each particle simultaneously determines the number of obstacles to track as well as an estimate of each obstacle.

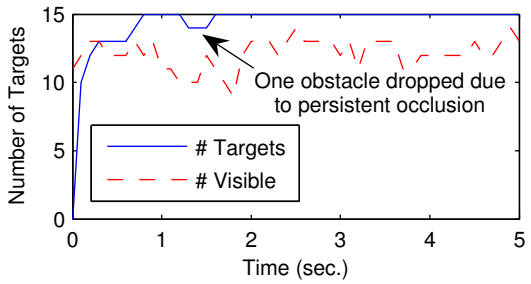

Fig. 3. Number of tracked obstacles vs. number of visible obstacles in one simulation. The dynamic mapping algorithm correctly tracks more of the 15 simulated objects than are ever visible instantaneously, resolving ambiguity despite obstacle occlusions. Note that a persistent obstacle occlusion has caused the algorithm to stop tracking one obstacle from $t = 1.3 - 1.4$ seconds.

Figure 3 shows the number of obstacles tracked within the most likely of the three particles over the course of one simulation using the above parameters. Notice that although there are 15 simulated objects, all 15 are never simultaneously visible to the sensor due to occlusions. Despite the occlusions, the dynamic mapping algorithm consistently maintains a more accurate number of targets than are ever visible. The algorithm is able to maintain obstacle distinctiveness due to the accuracy of the dynamics model, so that sparse measurements can easily be assigned to appropriate obstacles to resolve ambiguities.

Figure 4 shows the average position estimation error for each obstacle tracked within each particle at each iteration of the simulation. These average estimation errors have a transient increase in the beginning of the simulation, from $t \approx 0 - 1$ seconds as the filter determines the number of obstacles from the available measurements. Once the number of obstacles is determined, estimation errors begin to converge. Notice that at several epochs a particle makes an incorrect data association decision, causing its error to grow. Other particles in the distribution make correct decisions, however, emphasizing the benefit of storing multiple data association histories with particles. Standard filtering approaches do not have this capability.

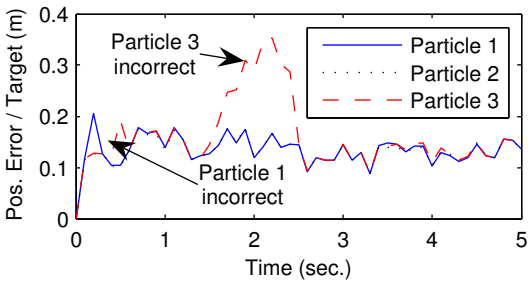

Fig. 4. Average position estimation error per obstacle within each of the three particles. Notice the incorrect data association decisions made by a particle near $t = 0.2$ and $t = 0.5$ seconds. Although one particle acquires substantial error at these epochs, the others maintain accurate estimates.

B. Validating The Effects Of The Number Of Particles On The Data Association Particle Filter

The second step in validating the dynamic mapping algorithm is to investigate the effect of the number of particles on its average obstacle estimation error. To that end, 100 different simulations are performed against the dynamic mapping algorithm. Each simulation is repeated using $N_p = 1$, 5, 10, and 20 particles and exactly the same 20 obstacles and the same sensor measurements, differing only in the random data associations made by the algorithm's particle filter. Particles are not resampled over the short simulation time, so that each particle represents a full data association history. The test has therefore been constructed so errors accumulate only from data association decisions made by the dynamic mapping algorithm.

TABLE I

ALGORITHM SUCCESS VS. NUMBER OF PARTICLES, DEFINED BY PAIRED T HYPOTHESIS TESTS EVALUATING ESTIMATION ERRORS

$N_p^R \backslash N_p^C$	1	5	10	20
1	–	0	0	0
5	88.2%	–	0	0
10	74.5%	5.9%	–	0
20	80.4%	3.9%	5.9%	–

Paired T hypothesis tests are used to evaluate the algorithm when it is run with different numbers of particles. Table I shows the effects of varying the number of particles on estimation error. Each row represents repeated trials of the algorithm run with N_p^R particles. Each column represents repeated trials of the algorithm run with N_p^C particles. Each entry in the table indicates the fraction of simulation timesteps for which the algorithm with N_p^R particles yields statistically better estimation errors than the algorithm run with N_p^C particles. For each test, comparisons are performed between estimation errors in the algorithm's most likely particle, with significance established at the 95% confidence level. From Table I, it is observed for these simulations that larger numbers of particles always result in better estimation errors. In addition, much of the benefit of multiple particles appears even at low numbers of particles, indicating the

effectiveness of the algorithm at resolving small data association ambiguities with relatively low numbers of particles.

V. EXPERIMENTAL VALIDATION

Section IV validates the primary features of the dynamic environment mapping algorithm in a simple simulated environment, where all obstacles are planar circles and all measurements are corrupted only with zero mean Gaussian noise with known statistics. While the simulated results validate the algorithm's theoretical foundations, they do not dispel the practical worries of implementing such an algorithm with physical sensing hardware. This section addresses these practical issues with an example implementation, tracking object boundaries in a complex roadside scene using a laser rangefinder.

In this experiment, a SICK LMS-291 laser rangefinder (LIDAR) provides 181 range and bearing measurements of a typical roadside scene at 0.5° increments over a 90° field of view at 75 Hz. The LIDAR scan plane is approximately parallel to the ground, so measurements are only taken of obstacles protruding from the ground plane. The origin of the dynamic map is set at the LIDAR, with the x-axis pointing along the center of the LIDAR's field of view. The LIDAR is kept stationary for the duration of the experiment. Figure 5 shows the scene observed by the LIDAR.

Fig. 5. An optical camera view of the roadside scene observed by the laser rangefinder (LIDAR). In this experiment the LIDAR is used to track occluded edges of obstacles. The LIDAR's x-axis lies perpendicular to the image plane, its y-axis is in the image plane to the left, and its z-axis completes the right-handed coordinate system.

In implementing the algorithm on physical hardware, the assumption made in section IV that the sensor directly measures obstacles must be abandoned. The LIDAR does not directly measure obstacles; it measures ranges and bearings to surfaces that reflect the energy it emits. There is no guarantee that a moving object reflecting the LIDAR beam in one scan will reflect the beam in the next scan, so algorithms attempting to match groups of LIDAR points from one scan to the next will inevitably make mistakes. These mistakes stem from the fact that in deciding what data to include and what to exclude, the preprocessing algorithms make their own data association decisions in a non-Bayesian way. To mitigate this problem, a compromise must be struck. The raw data must at least be processed minimally to extract measurements of features that are stable from scan to scan, but this search for structure within the raw data must be done with as few *ad hoc* decisions as possible. In essence, the goal is to leave as much grouping and assignment to the RBPF algorithm as computational resources can tolerate.

The preprocessing method given below is a simple measurement extraction technique found to work well in this particular scenario. The method extracts measurements of edges of obstacles: that is, measurements of the maximum and minimum bearings occupied by obstacles. It relies on the mild assumption that two adjacent measurements with substantially different range contain an occluded edge between them. These occluded edges are found experimentally to be stable from scan to scan at typical road speeds, making them appropriate objects for tracking. The method used for extracting range and bearing measurements at time k to obtain these occluded edges is given below:

1) Sort the pairs of LIDAR range and bearing measurements $(\ R_k^l \quad \theta_k^l\ )$, $l = 1 \ldots N_L$ in order of ascending bearing.
2) For the l^{th} measurement, if either $R_k^l - R_k^{l+1} > \Delta_r$ or $R_k^l - R_k^{l-1} > \Delta_r$, the l^{th} measurement is a valid occluded edge measurement.
3) Discard all invalid measurements.

Here Δ_r is the minimum range difference necessary to be considered an occluded edge, and it is set to 6m for this experiment. Valid occluded edge measurements extracted from this preprocessing step are then passed to the RBPF dynamic environment mapping algorithm, which then tracks each occluded edge using a Kalman Filter. The occluded edges, which do not necessarily represent fixed points on any rigid body, are modeled as point obstacles with random walk velocities. That is, they are modeled using equation 10 without estimating curvature, length, or width. The range and bearing measurement noise parameters for the obstacle Kalman Filters are set as follows:

$$\sigma_r = 1m \quad \sigma_b = 0.5^\circ \tag{14}$$

and the following process noise parameters are used to drive each Kalman Filter:

$$\sigma_x = \sigma_y = 10m \quad \sigma_s = 0.05m/s \quad \sigma_\theta = 0.57^\circ \tag{15}$$

Three particles are used to map the occluded edges of the dynamic environment shown in Figure 5. Figure 6 shows the occluded edges tracked by the most likely particle at a particular LIDAR scan, where blue dots are the raw LIDAR data, green squares are the extracted measurements, and red circles are the tracked obstacles. Performance in this scan is typical: over the 5 minutes of test data processed, the algorithm creates no persistent false positives, and recovers from all apparent data association mistakes. Tracking accuracy is typically on the order of 0.1 m or less when measurements are available.

The dynamic mapping algorithm can also be used to track more complicated obstacles, as shown in Figure 7.

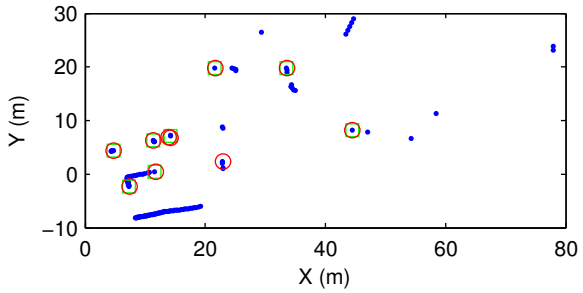

Fig. 6. Output of the dynamic mapping algorithm tracking occluded edges of objects extracted from LIDAR data. Blue dots are raw LIDAR measurements, green squares are the extracted measurements of the occluded edges, and red circles are the estimates of the occluded edges provided by the dynamic mapping algorithm.

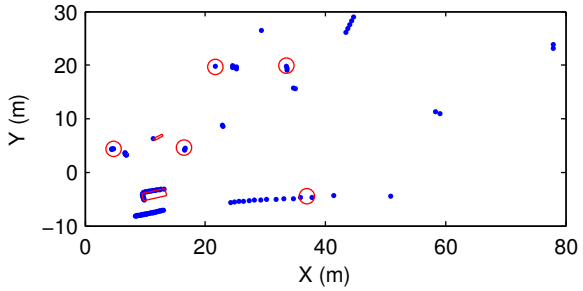

Fig. 7. Output of the dynamic mapping algorithm when occluded edges are tracked and combined into boxes. Note the algorithm correctly tracks the car near $[x, y] = [-5, 10]$ while determining its length and width.

Here, nearby pairs of occluded edges are probabilistically combined to form bounding boxes around obstacles, which are then tracked using the full ground vehicle dynamics of equation 10. Preliminary results look promising; the algorithm is able to determine the length and width of the car in Figure 7 to within 0.15 m using only occluded edges.

VI. CONCLUSION

A method for mapping dynamic environments from a ground robot has been developed. This method uses a Bayesian factorization to separate the mapping problem into two more tractable problems: data association, and target tracking under known measurement correspondences. A particle filter is used to solve the data association problem, and the particle filter has been developed utilizing the optimal distribution for sampling particles. The particle filter is coupled with a typical Kalman filtering approach for tracking specific targets under known measurement correspondences, and a typical dynamics model has been presented specifically for the ground robot environment. This dynamics model has been chosen to make different object classes more distinguishable, and to alleviate the need for a large particle filter over data association.

The algorithm has also been validated in simulation, where it was shown that a stationary sensing platform could simultaneously determine the number of obstacles and accurately estimate their states. This system was shown to work despite object occlusions by maintaining obstacles' state estimates and resolving measurement association ambiguities.

The algorithm has also been validated using measurements of occluded edges extracted from LIDAR data. These edges were tracked as obstacle boundaries, generating an accurate map of obstacle edges despite sensor limitations.

VII. ACKNOWLEDGMENTS

The authors would like to thank Dan Huttenlocher of Cornell University's Computer Science department and the members of the Cornell DARPA Urban Challenge Team for helpful discussion concerning practical implementation of the algorithm.

This material is based upon work supported under a National Science Foundation Graduate Research Fellowship.

REFERENCES

[1] S. Thrun, "Robotic mapping: A survey," School of Computer Science, Carnegie Mellon University, Pittsburgh, Tech. Rep. CMU-CS-02-111, February 2002.

[2] C. Wang, "Simultaneous localization, mapping and moving object tracking," PhD Thesis, Robotics Institute, Carnegie Mellon University, April 2004, CMU-RI-TR-04-23.

[3] D. Hähnel, R. Triebel, W. Burgard, and S. Thrun, "Map building with mobile robots in dynamic environments," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, 2003, pp. 1557 – 1563.

[4] M. Montemerlo and S. Thrun, "Conditional particle filters for simultaneous mobile robot localization and people tracking," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, 2002, pp. 695 – 701.

[5] ——, "Simultaneous localization and mapping with unknown data association using FastSLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, Taipei, Taiwan, September 2003, pp. 1985 – 1991.

[6] C. Stachniss and W. Burgard, "Mobile robot mapping and localization in non-static environments," in *Proceedings of the National Conference on Artificial Intelligence*, Pittsburgh, PA, 2005, pp. 1324 – 1329.

[7] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*. Orlando: Academic Press, Inc., 1988.

[8] S. Särkkä, A. Vehtari, and J. Lampinen, "Rao-blackwellized monte carlo data association for multiple target tracking," in *Proceedings of the Seventh International Conference on Information Fusion*, P. Svensson and J. Schubert, Eds., vol. 1, Mountain View, CA, 2004, pp. 583 – 590.

[9] D. Schulz, W. Burgard, D. Fox, and A. Cremers, "People tracking with mobile robots using sample-based joint probabilistic data association filters," *The International Journal of Robotics Research*, vol. 22, no. 2, pp. 99 – 116, 2003.

[10] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proceedings of the AAAI National Conference on Artificial Intelligence*. Edmonton, Canada: AAAI, 2002.

[11] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Upper Saddle River: Pearson Education, Inc., 2003.

[12] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gau ssian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174 – 188, February 2002.

[13] 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*, Stanford, 2000, pp. 176 – 183.

[14] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. New York: John Wiley & Sons, Inc., 2001.

[15] G. Grisetti, C. Stachniss, and W. Burgard, "Improving grid-based slam with Rao-Blackwellized particle filters by adaptive proposals and selective resampling," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005, pp. 2432 – 2437.

[16] F. Dellaert and C. Thorpe, "Robust car tracking using kalman filtering and bayesian templates," in *Proceedings of SPIE: Intelligent Transportation Systems*. Bellingham: The International Society for Optical Engineering, October 1997, vol. 3207.