Simultaneous Localization and Mapping with Unknown Data Association Using FastSLAM

Michael Montemerlo, Sebastian Thrun

Abstract— **The Extended Kalman Filter (EKF) has been the de facto approach to the Simultaneous Localization and Mapping (SLAM) problem for nearly fifteen years. However, the EKF has two serious deficiencies that prevent it from being applied to large, realword environments: quadratic complexity and sensitivity to failures in data association. FastSLAM, an alternative approach based on the Rao-Blackwellized Particle Filter, has been shown to scale logarithmically with the number of landmarks in the map [10]. This efficiency enables FastSLAM to be applied to environments far larger than could be handled by the EKF. In this paper, we will show that FastSLAM also substantially outperforms the EKF in environments with ambiguous data association. The performance of the two algorithms is compared on a real-world data set with various levels of odometric noise. In addition, we will show how negative information can be incorporated into FastSLAM in order to improve the accuracy of the estimated map.**

I. Introduction

The problem of simultaneous localization and mapping, also known as SLAM, has attracted immense attention in the mobile robotics literature. SLAM addresses the problem of building a map of an unknown environment from a sequence of noisy landmark measurements obtained from a moving robot. Since robot motion is also subject to error, the mapping problem necessarily induces a robot localization problem—hence the name SLAM. SLAM is considered by many to be an essential capability for autonomous robots operating in environments where precise maps and positioning are not available [3], [7], [14].

The dominant approach to the SLAM problem was introduced in a seminal paper by Smith, Self, and Cheeseman [13]. This paper proposed the use of the Extended Kalman Filter (EKF) for incrementally estimating the joint posterior distribution over robot pose and landmark positions. In the last decade, this approach has found widespread acceptance in field robotics, as a recent tutorial paper documents [2].

EKF-based approaches to SLAM suffer from two important limitations. First, sensor updates require time quadratic in the total number of landmarks K in the map. This complexity stems from the fact that the covariance matrix maintained by the Kalman filter has $O(K^2)$ elements, all of which must be updated even if just a single landmark is observed. Quadratic complexity limits the number of landmarks that can be handled by this approach to only a few hundred—whereas natural environment models frequently contain millions of features.

Michael Montemerlo and Sebastian Thrun are with the Robotics Institute at Carnegie Mellon University, Pittsburgh, Pennsylvania. Email: {mmde, thrun}@cs.cmu.edu

Second, EKF-based SLAM algorithms rely heavily on the assumption that the mapping between observations and landmarks is known. Associating a small number of observations with incorrect landmarks in the EKF can cause the filter to diverge. The problem of determining the correct mapping of observations to landmarks is commonly referred to as the data association, or correspondence problem.

An alternative approach to the SLAM problem has been introduced that factors the SLAM posterior into a localization problem and K independent landmark estimation problems conditioned on the robot pose estimate. This algorithm, called FastSLAM [10], uses a modified particle filter for estimating the posterior over robot paths. Each particle possesses K independent Kalman filters that estimate the landmark locations conditioned on the particle's path. The resulting algorithm is an instance of the Rao-Blackwellized particle filter [5], [11]. By representing particles as binary trees of Kalman Filters, observations can be incorporated into FastSLAM in $O(MlogK)$ time, where M is the number of particles, and K is the number of landmarks. FastSLAM has been demonstrated with up to 100,000 landmarks, problems far beyond the reach of the EKF.

Since each particle represents a different robot pose hypothesis, data association can be considered separately for every particle. This has two advantages. First, the noise of robot motion does not affect the accuracy of data association. Second, if observations are associated correctly in some particles and incorrectly in others, the incorrect particles will receive lower probabilities and will be removed in future resampling steps. In this way, FastSLAM can "forget" incorrect associations from the past, when other correct associations better explain the data.

We will demonstrate that FastSLAM substantially outperforms the Kalman Filter on real-world data with ambiguous data association. By adding extra odometric noise, we will show that FastSLAM continues to perform well in situations in which the Kalman Filter inevitably diverges. In fact, FastSLAM can estimate an accurate map in this environment without any odometry at all. Finally, we will show how to incorporate negative information into FastSLAM. The consideration of negative evidence results in a measurable increase in the accuracy of the resulting map.

II. SLAM Problem Definition

The SLAM problem is best described as a probabilistic Markov chain. The robot's pose at time t will be denoted s_t. If the robot is operating in a planar environment, this pose is the robot's x, y position and its heading orientation. The robot's environment is assumed to be comprised of K immobile, point landmarks. Each landmark is characterized by its location in space, denoted θ_i for $i = 1, \ldots, K$. The set of all landmarks will be denoted as θ.

Robot poses evolve according to a probabilistic law, referred to as the *motion model*:

$$p(s_t \mid u_t, s_{t-1}) \tag{1}$$

The current pose s_t is a probabilistic function of the robot control u_t and the previous pose s_{t-1}.

To map its environment, the robot can sense landmarks. It may be able to measure range and bearing to landmarks, relative to its local coordinate frame. The measurement at time t will be denoted z_t. Sensor measurements are also governed by a probabilistic law, referred to as the *measurement model*:

$$p(z_t \mid s_t, \theta_{n_t}, n_t) \tag{2}$$

where n_t is the index of the landmark currently being perceived. The observation z_t is a probabilistic function of the current pose of the robot s_t and the landmark being observed θ_{n_t}. While robots often can sense more than one landmark at a time, we follow the common practice of assuming that each observation corresponds to a measurement of exactly one landmark [2]. This convention is adopted solely for mathematical convenience. It poses no restriction, as multiple landmark sightings at a single time step can be processed sequentially.

In short, SLAM is the problem of determining the locations of all landmarks θ and robot poses s^t from measurements $z^t = z_1, \ldots, z_t$ and controls $u^t = u_1, \ldots, u_t$. In probabilistic terms, this is expressed by the following posterior:

$$p(s^t, \theta \mid z^t, u^t) \tag{3}$$

Here we use the superscript ^{t} to refer to a set of variables from time 1 to time t. If the associations n^t are known, the SLAM problem is simpler. The posterior becomes:

$$p(s^t, \theta \mid z^t, u^t, n^t) \tag{4}$$

III. Data Association

In real-world SLAM problems, the mapping n^t between observations and landmarks is rarely known. The total number of landmarks K is also unknown. Every time the robot makes an observation, that reading must be corresponded with an existing landmark or considered as coming from a previously unseen landmark. If this mapping is not obvious, picking the wrong association can cause a filter to diverge. A better understanding of how uncertainty in the SLAM posterior generates ambiguity in data association will demonstrate how simple data association heuristics often fail.

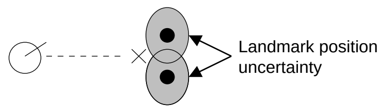


Fig. 1. Measurement Ambiguity: Two landmarks (shown as black circles) are close enough that the observation (shown as an x) plausibly could have come from either one.

Two factors contribute to uncertainty in the SLAM posterior: measurement noise and motion noise. Each leads to a different type of data association ambiguity. Noise in the measurement model will result in higher uncertainty in the landmark positions. Uncertain landmark positions will lead to *measurement ambiguity*, or confusion between nearby landmarks. (See Figure 1.) A mistake due to measurement ambiguity will have a relatively small effect on estimation error because the observation plausibly could have come from either landmark.

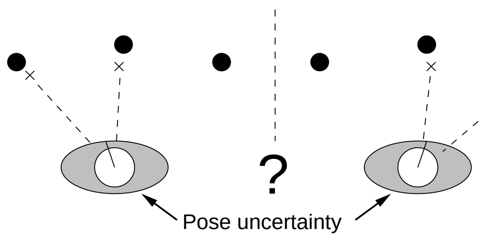


Fig. 2. Motion Ambiguity: Observations may be associated with completely different landmarks if the orientation of the robot changes a small amount.

Ambiguity due to motion noise can have much more severe consequences. Higher motion noise will lead to higher robot pose uncertainty after incorporating a control. If this uncertainty is high enough, different plausible poses of the robot will lead to drastically different data association hypotheses for the subsequent observations. *Motion ambiguity* is easily induced if there is significant angular uncertainty in the robot pose estimate. (See Figure 2.) If multiple observations are incorporated per timestep, the motion of the robot will correlate the associations of all of the landmarks. If a SLAM algorithm chooses the wrong association for a single landmark due to motion ambiguity, with high probability the rest of the associations will also be wrong. This will add a large amount of error to the robot's pose, and often cause a filter to diverge.

EKF SLAM algorithms commonly determine data association using a maximum likelihood approach. Each observation is associated with the landmark that was most likely to have generated it. At every time step, only the single most probable data association hypothesis is considered. More sophisticated EKF data association algorithms have

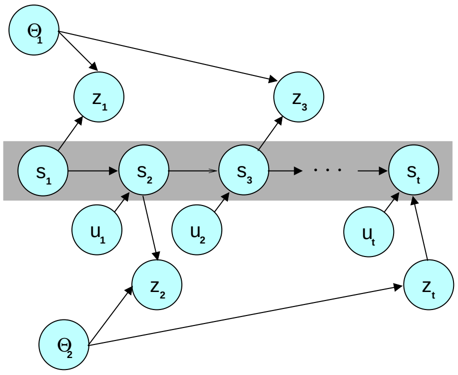


Fig. 3. The SLAM problem: The robot moves through poses $s_1 \ldots s_t$ based on a sequence of controls, $u_1 \ldots u_t$. As it moves, it observes nearby landmarks. At time $t = 1$, it observes landmark θ_1. The measurement is denoted z_1. At time $t = 2$, it observes the other landmark, θ_2, and at time $t = 3$, it observes θ_1 again. The SLAM problem is concerned with estimating the locations of the landmarks and the robot's path from the controls u and the measurements z. The gray shading illustrates the fact that knowledge of the robot's path renders the landmark positions θ_1 and θ_2 conditionally independent.

been developed to consider the best association of all observations simultaneously [12], however these approaches still rely on a single data association hypothesis at every timestep. In a scenario with ambiguous data association, an algorithm that maintains a single data association will sometimes pick the wrong association. If the ambiguity is due to the robot's motion, this will lead to divergence of the EKF.

The following sections of this paper will describe FastSLAM, an alternative approach to the SLAM problem that can sample over multiple data association hypotheses. Experimental data will show that this results in better performance in situations with significant motion ambiguity.

IV. FastSLAM with Known Data Association

Figure 3 illustrates a generative probabilistic model (dynamic Bayes network) that describes the SLAM problem. From this diagram it is clear that the SLAM problem contains important conditional independences. In particular, knowledge of the robot's path $s_1, \ldots, s_t$ renders the individual landmark measurements *independent*. So for example, if an oracle provided us with the exact path of the robot, the problem of determining the landmark locations could be decoupled into K independent estimation problems, one for each landmark. This conditional independence is the basis for the FastSLAM algorithm.

This conditional independence implies that the posterior (4) can be factored as follows into a robot path posterior and a product of individual landmark posteriors conditioned on the robot's path:

$$\begin{aligned} &p(s^t, \theta \mid z^t, u^t, n^t) \\ &= \quad p(s^t \mid z^t, u^t, n^t) \prod_{i=1}^{K} p(\theta_i \mid s^t, z^t, u^t, n^t) \end{aligned} \tag{5}$$

A derivation of this factorization is given in the Appendix. FastSLAM estimates the factored SLAM posterior using a modified particle filter, with K independent Kalman Filters for each particle to estimate the landmark positions conditioned on the hypothesized robot paths. The resulting algorithm is an instance of the Rao-Blackwellized particle filter [5], [11].

A. Particle Filter Path Estimation

FastSLAM estimates the robot path posterior in (5) using a particle filter, in a way that is similar (but not identical) to the *Monte Carlo Localization (MCL)* algorithm [1]. At each point in time, FastSLAM maintains a set of particles representing the posterior $p(s^t \mid z^t, u^t, n^t)$, denoted S_t. Each particle $s^{t,[m]} \in S_t$ represents a "guess" of the robot's path:

$$S_t \quad = \quad \{s^{t,[m]}\}_m \quad = \quad \{s_1^{[m]}, s_2^{[m]}, \ldots, s_t^{[m]}\}_m \tag{6}$$

We use the superscript notation $^{[m]}$ to refer to the m-th particle in the set.

The particle set S_t is calculated incrementally, from the set S_{t-1} at time $t-1$, a robot control u_t, and a measurement z_t. First, each particle $s^{t,[m]}$ in S_{t-1} is used to generate a probabilistic guess of the robot's pose at time t:

$$s_t^{[m]} \quad \sim \quad p(s_t \mid u_t, s_{t-1}^{[m]}) \tag{7}$$

This guess is obtained by sampling from the probabilistic motion model. This estimate is then added to a temporary set of particles, along with the path $s^{t-1,[m]}$. Under the assumption that the set of particles in S_{t-1} is distributed according to $p(s^{t-1} \mid z^{t-1}, u^{t-1}, n^{t-1})$ (which is an asymptotically correct approximation), the new particle is distributed according to:

$$p(s^t \mid z^{t-1}, u^t, n^{t-1}) \tag{8}$$

This distribution is commonly referred to as the *proposal distribution* of particle filtering.

After generating M particles in this way, the new set S_t is obtained by sampling from the temporary particle set. Each particle $s^{t,[m]}$ is drawn (with replacement) with a probability proportional to a so-called *importance factor* $w_t^{[m]}$, which is calculated as follows [9]:

$$w_t^{[m]} = \frac{\text{target dist.}}{\text{proposal dist.}} = \frac{p(s^{t,[m]} \mid z^t, u^t, n^t)}{p(s^{t,[m]} \mid z^{t-1}, u^t, n^{t-1})} \tag{9}$$

The exact calculation of (9) will be discussed further below. The resulting sample set S_t is distributed according to an approximation to the desired path posterior $p(s^t \mid z^t, u^t, n^t)$, an approximation which is correct as the number of particles M goes to infinity. We also notice that only the most recent robot pose estimate $s_{t-1}^{[m]}$ is used when generating the particle set S_t. This will allows us to silently "forget" all other pose estimates, rendering the size of each particle independent of the time index t.

B. Landmark Location Estimation

FastSLAM represents the conditional landmark estimates $p(\theta_i \mid s^t, z^t, u^t, n^t)$ in (5) using Kalman filters. Since this estimate is conditioned on the robot pose, the Kalman filters are attached to individual pose particles in S_t. More specifically, the full posterior over paths and landmark positions in the FastSLAM algorithm is represented by the sample set

$$\mathcal{S}_t = \{s^{t,[m]}, \mu_1^{[m]}, \Sigma_1^{[m]}, \ldots, \mu_K^{[m]}, \Sigma_K^{[m]}\}_m \qquad (10)$$

Here $\mu_i^{[m]}$ and $\Sigma_i^{[m]}$ are the mean and covariance of the Gaussian representing the i-th landmark θ_i, attached to the m-th particle. In the planar robot navigation scenario, each mean $\mu_i^{[m]}$ is a two-element vector, and $\Sigma_i^{[m]}$ is a 2 by 2 matrix.

The posterior over the i-th landmark pose θ_i is easily obtained. Its computation depends on whether or not the landmark was observed. If the landmark is observed, we obtain:

$$\begin{aligned} & p(\theta_{i=n_t} \mid s^t, z^t, u^t, n^t) & (11) \\ & \quad \propto \quad p(z_t \mid \theta_{n_t}, s_t, n_t)\, p(\theta_{n_t} \mid s^{t-1}, z^{t-1}, u^{t-1}, n^{t-1}) & \end{aligned}$$

If landmark i is not observed, we simply leave the Gaussian unchanged:

$$p(\theta_{i\neq n_t} \,|\, s^t, z^t, u^t, n^t) = p(\theta_i \,|\, s^{t-1}, z^{t-1}, u^{t-1}, n^{t-1}) \quad (12)$$

The FastSLAM algorithm implements the update equation (11) using the extended Kalman filter (EKF). As in existing EKF approaches to SLAM, this filter uses a linearized version of the perceptual model $p(z_t \mid s_t, \theta_{n_t}, n_t)$ [2]. One significant difference between FastSLAM's use of Kalman filters and that of the traditional SLAM algorithm is that the updates in the FastSLAM algorithm involve only a Gaussian of dimension two (for the two landmark location parameters). In the EKF-based SLAM approach a Gaussian of size $2K + 3$ has to be updated (with K landmarks and 3 robot pose parameters). This calculation can be done in constant time in FastSLAM, whereas it requires time quadratic in K in the EKF.

C. Importance Weights

Before the robot path particles can be resampled, the importance weights (9) must be calculated. For the sake of brevity, the derivation of these importance weights has been omitted. The weight w_t is equal to:

$$\begin{aligned} w_t^{[m]} &\propto \frac{p(s^{t,[m]} \mid z^t, u^t, n^t)}{p(s^{t,[m]} \mid z^{t-1}, u^t, n^{t-1})} \\ &= \int p(z_t \mid \theta_{n_t}^{[m]}, s_t^{[m]})\, p(\theta_{n_t}^{[m]})\, d\theta_{n_t} \qquad (13) \end{aligned}$$

This quantity can be computed in closed form because the landmark estimators are Kalman filters. For a complete derivation of the importance weights, see [10].

V. FASTSLAM WITH UNKNOWN DATA ASSOCIATION

If the mapping between observations and landmarks is known, the FastSLAM algorithm samples over robot paths, and computes the conditional landmark estimates analytically for every path sample. When this mapping is unknown, FastSLAM can be extended to sample over possible data associations as well as robot paths. There are several ways that this sampling can be done.

A. Per-Particle Maximum Likelihood Data Association

The simplest approach to sampling over data associations is to adopt the maximum likelihood assignment procedure used by EKFs, but on a per-particle basis. Particles that pick the correct data association will receive high probabilities because they explain the measurements well. Particles that assign observations incorrectly will receive lower probabilities and be removed in future resampling steps. This procedure can be written as:

$$n_t^{[m]} = \underset{n_t}{\operatorname{argmax}}\, p(z_t \mid s_t^{[m]}, \theta_{n_t}, n_t) \qquad (14)$$

Per-particle data association has two clear consequences. First, robot motion noise does not affect the accuracy of data association, given an appropriate number of particles. This fact alone will result in significantly improved performance in situations with substantial motion ambiguity. If applied to the scenario shown in Figure 2, some of the particles will represent the pose on the left and pick the first data assoociation hypothesis, and other particles will pick the second hypothesis.

The second consequence of per-particle data association is built-in, delayed decision-making. At any given time, some fraction of the particles will receive plausible, but wrong data associations. In the future, new observations may be received that clearly refute these prior assignments. These particles will receive low probability and be removed from the filter. As a result, the effect of wrong associations made in the past can be removed from the filter at a later time. This is in stark contrast to the EKF, in which the effect of an incorrect data association can never be removed once it is incorporated. Moreover, no heuristics are needed to manage the removal of old associations. This is done in a statistically valid manner, simply as a consequence of the resampling step of the particle filter.

Naturally, sampling over robot paths and data associations will require more particles than sampling over robot paths alone. Results in the next section will show that even a modest number of particles ($M = 100$) will result in substantially improved data association over the EKF.

B. Monte-Carlo Data Association

Per-particle data association can be taken a step further. Instead of assigning each particle the most likely data association, each particle can draw a random association weighted by the probabilities of each landmark having generated the observation. Using this approach FastSLAM will also generate correct data association hypotheses given measurement ambiguity. If a small number of landmarks exhibit measurement ambiguity, this procedure can have a small positive effect on estimation accuracy. However, uniformly high measurement error induces a combinatorial number of plausible data associations for every set of observations. This, in turn, would require exponentially more particles than the same scenario with known data association.

C. Mutual Exclusion

If more than one observation is received per timestep, mutual exclusion can be used to eliminate data association hypotheses that assign multiple measurements to the same landmark. Mutual exclusion can be applied in EKFs, but only if the data associations of all observations are considered simultaneously. This consideration is, in general, computationally difficult with a large number of observations. Since FastSLAM maintains a set of data association hypotheses, the mutual exclusion constraint can be applied in a greedy fashion. Each observation is associated with the most likely landmark in each particle that has not received an observation yet. Since the greedy heuristic will sometimes apply the mutual exclusion constraint incorrectly, more particles will be needed when applying this technique. However, mutual exclusion makes the process of deciding when to add new landmarks a much simpler problem. Instead of incorrectly assigning an observation from an unseen landmark to a nearby, previously seen landmark, mutual exclusion will force the creation of a new landmark.

D. Negative Information

The point landmark representation is acceptable for making inferences about where object are in the world. In general, it does not allow us to say where objects *are not*. Simply deciding that "areas that do not contain landmarks are empty" is not correct. However, one type of inference can be made using negative information. If a robot expects to see a particular landmark and does not, the robot should become less confident that this landmark actually exists.

In order to exploit this kind of negative information in SLAM, we will borrow a technique normally used for making evidence grids. For each landmark in every particle, we will estimate a single binary variable r indicating whether this landmark represents a real landmark in the world. Instead of keeping track of the probability $p(r|z^t)$, we will instead estimate the log odds ratio:

$$log \frac{p(r \mid z^t)}{1 - p(r \mid z^t)} \tag{15}$$

The log-odds formulation of the binary Bayes filter is extremely easy to update. A complete description of this procedure is given in [15]. In short, every time the landmark is observed, a constant value is added to the log odds ratio. Every time the landmark is not observed when it should have been, a constant value is subtracted from the log odds ratio. If this ratio falls below a given threshold, the landmark is considered to be spurious and removed.

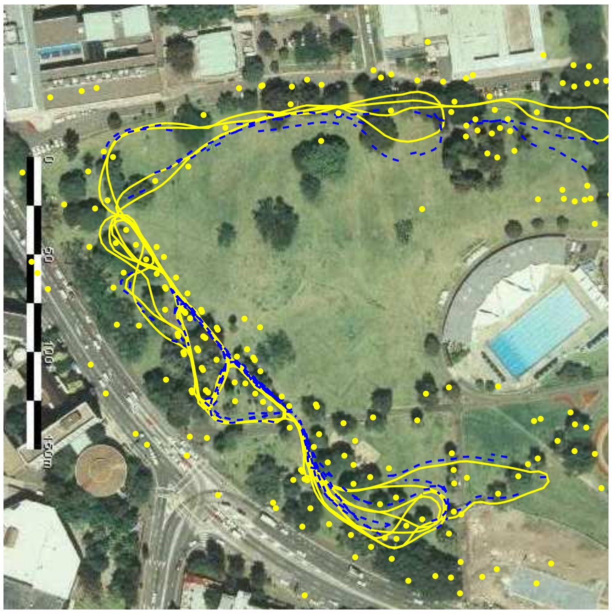


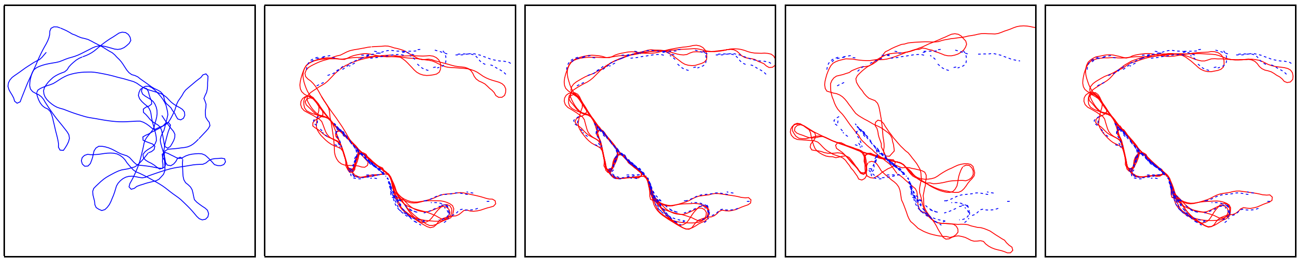
Fig. 4. Typical FastSLAM run. The yellow path is the estimated path of the vehicle. The blue dashed line is the GPS ground truth data. The yellow circles are the estimated positions of the landmarks.

Negative information is particularly useful for removing spurious features from the map. These features may be the result of false positives generated by the feature detection algorithm, or they may correspond to moving objects. Results in the next section will show that using negative information dramatically reduces the number of spurious landmarks in the estimated map.

VI. Experimental Results

The FastSLAM and EKF algorithms were compared using the University of Sydney's Victoria Park data set. An instrumented vehicle with a laser rangefinder was driven through Victoria Park. Encoders on the vehicle recorded velocity and steering angle. Ranges and bearings to nearby trees were extracted from the laser data using a local minima detector. The vehicle was driven around the park for approximately 30 minutes, covering a distance of over 4 km. Filter accuracy was calculated by comparing the estimated vehicle path with GPS. An example of a complete run of FastSLAM is shown in Figure 4.

Figure 5(a) shows the estimated path of the vehicle based solely on odometry. Even though this demonstrates that the vehicle's odometry is quite inaccurate, data association is this data set is generally straightforward. The accuracy of the laser sensor, and the widely spaced fea-



(a) (b) (c) (d) (e)

Fig. 5. (a) raw odometry (b) EKF with low odometric error (c) FastSLAM with low odometric error (d) EKF with high odometric error (e) FastSLAM with high odometric error - In Figures (b) through (e) the red solid path is the estimated path. The blue dashed path is the GPS data.

tures rarely generate any kind of data association ambiguity. It comes as no surprise that the performance of FastSLAM and the EKF are comparable. Example path estimates for the EKF and FastSLAM with low odometric noise are shown in Figures 5(b) and 5(c).

The performance of the two algorithms was also compared after adding various amounts of noise to the observed controls. The results of this comparison are shown in Figure 6. The increased motion noise had no measurable effect on the accuracy of FastSLAM. Additional motion noise caused the EKF to diverge, resulting in very high position error on average. In all experiments, FastSLAM was run with 100 particles. Example path estimates for the EKF and FastSLAM with high odometric noise are shown in Figures 5(d) and 5(e). In Figure 5(d), the EKF has clearly diverged.

FastSLAM was also run on the Victoria Park data set without any odometry estimates at all. This was accomplished by adding velocity to the vehicle's state, and assuming a overly conservative motion model. The vehicle's translational and rotational velocity were assumed to vary as a continuous random walk. Figure 7 shows the estimated path of the vehicle.

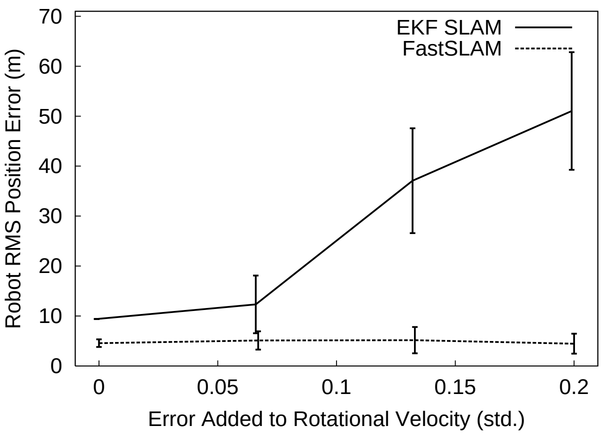


Fig. 6. Accuracy of the vehicle path with varying levels of odometry noise

Not all of the features detected by the feature extractor belonged to static objects. Some features were generated by cars, and other moving objects. Features from moving objects frequently resulted in spurious landmarks being added to the map. In general, it is difficult to measure the accuracy of the estimated map with this data set because there is no ground truth data available for the landmarks. However, incorporating negative information did result in 44 percent fewer landmarks on average, and many fewer landmarks in dynamic areas (e.g. the street).

VII. Conclusions

We have presented an extension of the FastSLAM algorithm to the case of unknown data association. In addition to sampling over robot paths, this formulation of FastSLAM also samples over potential data associations. The resulting algorithm consistently outperformed the Extended Kalman Filter on a real world data set with various levels of odometric noise. In addition, we have shown how to incorporate negative information into FastSLAM. This technique is not specific to FastSLAM and can also be applied to other SLAM algorithms, including the EKF. Use of negative evidence results in a measurable decrease in the number of false landmarks, especially if the feature detector being used generates a large number of spurious features.

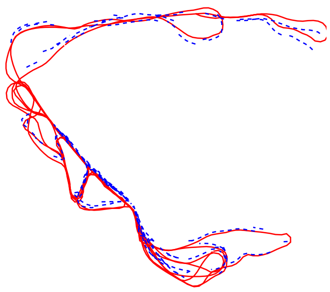


Fig. 7. Robot path estimated without odometry

VIII. Appendix: Derivation of Factorization

The SLAM posterior (4) can be rewritten as:

$$p(s^t \mid z^t, u^t, n^t)\ p(\theta \mid s^t, z^t, u^t, n^t) \tag{16}$$

Thus, to derive the factored version (5) it suffices to show that:

$$p(\theta \mid s^t, z^t, u^t, n^t) = \prod_{i=1}^{K} p(\theta_i \mid s^t, z^t, u^t, n^t) \tag{17}$$

This will be shown using induction. To do this we will need two intermediate results. The first is the probability of the landmark being observed θ_{n_t} given the robot path, the observations, and the controls.

$$\begin{aligned} & p(\theta_{n_t} \mid s^t, z^t, u^t, n^t) \qquad (18) \\ & \overset{\text{Bayes}}{=} \frac{p(z_t \mid \theta_{n_t}, s^t, z^{t-1}, u^t, n^t)}{p(z_t \mid s^t, z^{t-1}, u^t, n^t)}\ p(\theta_{n_t} \mid s^t, z^{t-1}, u^t, n^t) \\ & \overset{\text{Markov}}{=} \frac{p(z_t \mid \theta_{n_t}, s_t, n_t)}{p(z_t \mid s^t, z^{t-1}, u^t, n^t)}\ p(\theta_{n_t} \mid s^{t-1}, z^{t-1}, u^{t-1}, n^{t-1}) \end{aligned}$$

Next we solve for the rightmost term.

$$\begin{aligned} & p(\theta_{n_t} \mid s^{t-1}, z^{t-1}, u^{t-1}, n^{t-1}) \\ & = \frac{p(z_t \mid s^t, z^{t-1}, u^t, n^t)}{p(z_t \mid \theta_{n_t}, s_t, n_t)}\ p(\theta_{n_t} \mid s^t, z^t, u^t, n^t) \qquad (19) \end{aligned}$$

For our second intermediate result, we will compute the probability of any landmark that is not being observed given the robot path, the observations, and the controls.

$$\begin{aligned} & p(\theta_{i \neq n_t} \mid s^t, z^t, u^t, n^t) \\ & \overset{\text{Markov}}{=} p(\theta_{i \neq n_t} \mid s^{t-1}, z^{t-1}, u^{t-1}, n^{t-1}) \qquad (20) \end{aligned}$$

We will assume the following induction hypothesis:

$$\begin{aligned} & p(\theta \mid s^{t-1}, z^{t-1}, u^{t-1}, n^{t-1}) \\ & = \prod_{i=1}^{K} p(\theta_i \mid s^{t-1}, z^{t-1}, u^{t-1}, n^{t-1}) \qquad (21) \end{aligned}$$

For the base case of $K = 1$ equation (17) is trivially true. In general, when $K > 1$:

$$\begin{aligned} & p(\theta \mid s^t, z^t, u^t, n^t) \qquad (22) \\ & \overset{\text{Bayes}}{=} \frac{p(z_t \mid \theta, s^t, z^{t-1}, u^t, n^t)}{p(z_t \mid s^t, z^{t-1}, u^t, n^t)}\ p(\theta \mid s^t, z^{t-1}, u^t, n^t) \\ & \overset{\text{Markov}}{=} \frac{p(z_t \mid \theta_{n_t}, s_t, n_t)}{p(z_t \mid s^t, z^{t-1}, u^t, n^t)}\ p(\theta \mid s^{t-1}, z^{t-1}, u^{t-1}, n^{t-1}) \\ & \overset{\text{Induction}}{=} \frac{p(z_t \mid \theta_{n_t}, s_t, n_t)}{p(z_t \mid s^t, z^{t-1}, u^t, n^t)} \prod_{i=1}^{K} p(\theta_i \mid s^{t-1}, z^{t-1}, u^{t-1}, n^{t-1}) \end{aligned}$$

By replacing the terms of the product with (19) and (20):

$$\begin{aligned} & p(\theta \mid s^t, z^t, u^t, n^t) \\ & = p(\theta_{n_t} \mid s^t, z^t, u^t, n^t) \prod_{i \neq n_t}^{K} p(\theta_i \mid s^t, z^t, u^t, n^t) \\ & = \prod_{i=1}^{K} p(\theta_i \mid s^t, z^t, u^t, n^t) \qquad (23) \end{aligned}$$

Acknowledgments

This research has been sponsored by DARPA's MARS Program (contract N66001-01-C-6018), DARPA's CoABS Program (contract F30602-98-2-0137), and DARPA's MICA Program (contract F30602-01-C-0219), all of which is gratefully acknowledged.

Special thanks to Eduardo Nebot and Juan Niento from the University of Sydney for giving us access to the Victoria Park dataset.

We also gratefully acknowledge the Fannie and John Hertz Foundation for their support of Michael Montemerlo's graduate research.

References

[1] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. *ICRA-99*.

[2] G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. An experimental and theoretical investigation into simultaneous localisation and map building (SLAM). *Lecture Notes in Control and Information Sciences: Experimental Robotics VI*, Springer, 2000.

[3] 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.

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

[5] A Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. *UAI-2000*.

[6] J. Guivant and E. Nebot. Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transaction of Robotic and Automation*, May 2001.

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

[8] J.J. Leonard and H.J.S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. *ISRR-99*.

[9] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equations of state calculations by fast computing machine. *Journal of Chemical Physics*, 21, 1953.

[10] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. *AAAI-02*

[11] K. Murphy and S. Russell. Rao-blackwellized particle filtering for dynamic bayesian networks. In *Sequential Monte Carlo Methods in Practice*, Springer, 2001.

[12] Neira, J., Tardos, J.D., Data Association in Stochastic Mapping Using the Joint Compatibility Test. In *IEEE Trans. on Robotics and Automation* vol. 17, no. 6, pp. 890-897, Dec 2001.

[13] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous Robot Vehicles*, Springer, 1990.

[14] C. Thorpe and H. Durrant-Whyte. Field robots. *ISRR-2001*.

[15] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21-71, 1998.