EKF SLAM vs. FastSLAM – A Comparison

Michael Calonder, Computer Vision Lab
Swiss Federal Institute of Technology, Lausanne (EPFL)
michael.calonder@epfl.ch

The two algorithms are described with a planar robot application in mind. Generalization is to any spatial SLAM scenarios is straightforward. For simplicity, we assume there is no control input. The pose consists of the robot's position $(x, y)^\top$ and its heading direction δ: $\mathbf{s}_t := (x, y, \delta)^\top$. The landmarks are denoted θ_i, simply consisting of a pair of planar coordinates.

1 Probabilistical EKF Formulation

The Extended Kalman Filter (EKF) can be viewed as a variant of a Bayesian Filter; EKFs provide a recursive estimate of the state of a dynamic system, or more precise, solve an unobservable, nonlinear estimation problem. Roughly speaking, the state $\mathbf{x}_t$ of a system at time t can be considered a random variable where the uncertainty about this state is represented by a probability distribution.

One is interested in the posterior density $p(\mathbf{x}_t \,|\, \mathbf{z}^t)$, where $\mathbf{z}^t := \{\mathbf{z}_1, \dots, \mathbf{z}_t\}$ is the set of measurements up to time t and $\mathbf{x}_t := \left(\mathbf{s}_t^\top, \theta_0^\top, \theta_1^\top, \dots, \theta_K^\top\right)^\top$ is the state vector. (Note that the superscript ^{t} refers to the *set* of variables at time t.) In general, the complexity of computing such a density grows exponentially with time; to make the computation tractable, the true state is being assumed to be an unobserved Markov process implying that

- the true state is conditionally independent of all earlier states except the previous state: $p\left(\mathbf{x}_t \,|\, \mathbf{x}^{t-1}\right) = p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})$ where $\mathbf{x}^t := \{\mathbf{x}_1, \dots, \mathbf{x}_t\}$ denotes the set of states.
- the measurement at the t-th timestep depends only upon the current state and is conditionally independent of all other states: $p(\mathbf{z}_t \,|\, \mathbf{x}^t) = p(\mathbf{z}_t \,|\, \mathbf{x}_t)$.

EKF uses a prediction and an update step. The prediction step calculates the probability of the current state $\mathbf{x}_t$, while the measurement $\mathbf{z}_t$ for the current time step k is not yet available:

$$p\left(\mathbf{x}_t \,|\, \mathbf{z}^{t-1}\right) = \int p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})\; p\left(\mathbf{x}_{t-1} \,|\, \mathbf{z}^{t-1}\right)\; d\mathbf{x}_{t-1}.$$

The left PDF term of the integrand is the PDF of the motion model (see below) where the right part is the PDF of the state estimated in the last time step, rendering the estimation recursive.

Once we know $p\left(\mathbf{x}_t \,|\, \mathbf{z}^{t-1}\right)$ we can find the posterior or corrected estimate using a new measurement $\mathbf{z}_t$:

$$p\left(\mathbf{x}_t \,|\, \mathbf{z}^t\right) = \frac{p(\mathbf{z}_t \,|\, \mathbf{x}_t)\; p\left(\mathbf{x}_t \,|\, \mathbf{z}^{t-1}\right)}{p(\mathbf{z}_t \,|\, \mathbf{z}^{t-1})}$$

with $p\left(\mathbf{z}_t \,|\, \mathbf{z}^{t-1}\right) = \int p(\mathbf{z}_t \,|\, \mathbf{x}_t)\; p\left(\mathbf{x}_t \,|\, \mathbf{z}^{t-1}\right)\; d\mathbf{x}_t$ being some (constant) normalization term. The PDF $p(\mathbf{z}_t \,|\, \mathbf{x}_t)$ is where the measurement model comes in.

EKF filtering makes the following assumptions about the respective PDFs:

- $p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}) \sim \mathcal{N}(\mathbf{f}(\mathbf{x}_{t-1}), \mathbf{Q}_t)$
- $p(\mathbf{z}_t \,|\, \mathbf{x}_t) \sim \mathcal{N}(\mathbf{h}(\mathbf{x}_t), \mathbf{R}_t)$
- $p\left(\mathbf{x}_{t-1} \,|\, \mathbf{z}^{t-1}\right) \sim \mathcal{N}(\mathbf{x}_{t-1}, \mathbf{P}_{t-1})$

where $\mathcal{N}(\mu, \mathbf{\Sigma})$ denotes the Gaussian distribution with mean μ and covariance matrix $\mathbf{\Sigma}$, $\mathbf{f}(\mathbf{x})$ the motion model and $\mathbf{h}(\mathbf{x})$ the measurement model.

The error covariance $\mathbf{P}$ is in the prediction step at time k projected ahead by

$$\mathbf{P}_t^- = \mathbf{A}_t\, \mathbf{P}_{t-1}\, \mathbf{A}_t^\top + \mathbf{W}_t\, \mathbf{Q}_{t-1}\, \mathbf{W}_t^\top$$

and finally updated in the correction step according to

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t)\, \mathbf{P}_t^-$$

where $\mathbf{K}_t$ is the Kalman gain matrix [8]. The Jacobians involved are

$$\mathbf{A}_t = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}_{t-1}}, \quad \mathbf{W}_t = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{w}}\right|_{\mathbf{x}_{t-1}}, \quad \text{and} \quad \mathbf{H}_t = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}_t^-}$$

with $\mathbf{w}$ being the process noise vector [8].

2 FastSLAM Algorithm

The key idea of FastSLAM exploits the fact that knowledge of the robot's path $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_t$ renders the individual landmark measurements independent [3], as originally observed by Murphy [6]. FastSLAM decomposes the SLAM problem into one robot localization problem, and a collection of K landmark estimation problems.

In FastSLAM, alike in EKF SLAM, poses are assumed to behave according to a probabilistic law named motion model with an underlying density

$$p\left(\mathbf{s}_t \,|\, \mathbf{s}_{t-1}\right).$$

Likewise, the measurements are governed by the (probabilistic) measurement model $p\left(\mathbf{z}_t \,|\, \mathbf{s}_t, \theta, n_t\right)$ with $\mathbf{z}_t$ measurement, $\theta = \{\theta_1, \ldots, \theta_K\}$ the set of landmarks, and $n_t \in \{1, \ldots, K\}$ the index of the observed landmark at time t (only one at a time). The ultimate goal is to estimate the posterior $p\left(\mathbf{s}^t, \theta \,|\, \mathbf{z}^t\right)$.

The exact factored representation employed by FastSLAM reads

$$\begin{aligned} p\left(\mathbf{s}^t, \theta \,|\, \mathbf{z}^t, n^t\right) &= p\left(\mathbf{s}^t \,|\, \mathbf{z}^t, n^t\right)\, p\left(\theta \,|\, \mathbf{z}^t, n^t\right) \\ &= p\left(\mathbf{s}^t \,|\, \mathbf{z}^t, n^t\right) \prod_{1 \le k \le K} p\left(\theta_k \,|\, \mathbf{s}^t, \mathbf{z}^t, n^t\right). \end{aligned} \tag{1}$$

Note the conditional dependence on n^t. The $K+1$ factors in (1) are computed as follows.

- **Estimation of path** $p\left(\mathbf{s}^t \,|\, \mathbf{z}^t, n^t\right)$
This is achieved by a particle filter: FastSLAM maintains a set of M particles, $\{\mathbf{s}^{t,[m]}\} = \{\mathbf{s}_1^{[m]}, \ldots, \mathbf{s}_t^{[m]}\}$, where $^{[m]}$ refers to the m-th particle in the set. To create the particle set at time t, $\mathbf{s}^{t,[m]}$, one first samples from the motion model M particles into a temporary set; $\mathbf{s}_t^{[m]} \sim p\left(\mathbf{s}_t \,|\, \mathbf{s}_{t-1}^{[m]}\right)$. Then the importance factor weights $w_t^{[m]}$ are calculated; they determine the probability that a certain particle from the temporary set enters the final set.

- **Estimation of landmark locations** $p\left(\theta_k \,|\, \mathbf{s}^t, \mathbf{z}^t, n^t\right)$
The posterior update depends on whether the k-th landmark has been observed or not,

$$p\left(\theta_k \,|\, \mathbf{s}^t, \mathbf{z}^t, n^t\right) \quad \propto \quad \begin{cases} p\left(\mathbf{z}_t \,|\, \theta_k, \mathbf{s}_t, n_t\right) \cdot p\left(\theta_k \,|\, \mathbf{s}^{t-1}, \mathbf{z}^{t-1}, n^{t-1}\right) & \text{if } k = n_t \\ p\left(\theta_k \,|\, \mathbf{s}^{t-1}, \mathbf{z}^{t-1}, n^{t-1}\right) & \text{otherwise} \end{cases}$$

  FastSLAM implements the above update equation by means of an EKF.

- **Data association problem**
FastSLAM (as most existing SLAM solutions based on EKF) solves the data association problem via maximum likelihood estimation, see section 6.1.

3 Similarities

Basically, EKF SLAM and FastSLAM solve the same problem while making use of the identical probabilistic motion and measurement models. Furthermore, both use Kalman filtering: EKF SLAM applies the filter once to a high dimensional filtering problem where FastSLAM employes $M \cdot K$ tiny EKFs (K of them in each particle).

4 Differences

4.1 State Vector

A fundamental advantage of FastSLAM over EKF based approaches to the SLAM problem is that the EKF suffers from a $\mathcal{O}(K^2)$ complexity where K being the number of landmarks. In contrast, FastSLAM has an $\mathcal{O}(M \log K)$ complexity with $M = \text{const}$ denoting the number of particles.

EKF SLAM	**FastSLAM**
State vector is comprised of current pose estimate as well as landmark estimates: $$\mathbf{x}_t = \left(\mathbf{s}_t^\top, \theta_0^\top, \theta_1^\top, \ldots, \theta_K^\top\right)^\top .$$ The map management will have to change the size of the state vector at runtime and update a $\dim(\mathbf{x}_t) \times \dim(\mathbf{x}_t)$ matrix in each step, $\dim(\mathbf{x}_t) = 2K + 3$.	No state vector as in EKF SLAM, but loose set of small parts instead; the K covariance matrices in each particle storing the landmark uncertainties have a small constant size (here: 2×2). New observations just slightly alter some tree structure storing the parameters of the Gaussians of each particle.

4.2 Posterior Density

EKF SLAM	**FastSLAM**
All densities involved in the calculation of the posterior $$p\left(\mathbf{x}_t \,\vert\, \mathbf{z}^t\right) = \frac{p\left(\mathbf{z}_t \,\vert\, \mathbf{x}_t\right)\; p\left(\mathbf{x}_t \,\vert\, \mathbf{z}^{t-1}\right)}{p\left(\mathbf{z}_t \,\vert\, \mathbf{z}^{t-1}\right)}$$ are modeled as Gaussians.	The problem is decomposed into $K+1$ subproblems: one problem of estimating a posterior over robot paths $\mathbf{s}^t$, and K problems of estimating the locations of the K landmarks, $$p\left(\mathbf{s}^t, \theta \,\vert\, \mathbf{z}^t, n^t\right) = p\left(\mathbf{s}^t \,\vert\, \mathbf{z}^t, n^t\right) \prod_k p\left(\theta_k \,\vert\, \mathbf{s}^t, \mathbf{z}^t, n^t\right) .$$ This factorization is exact.

4.3 Data Association

EKF SLAM	**FastSLAM**
Different approaches, e.g. initialization of a 3D line into the map and successive estimation of depth before introducing it definitely. But: all attempts use just *one* data association, if it is wrong for some reason, e.g. because of an ambigous environment, the filter will probably diverge.	Each particle has its own hypothesis of data association, $n_t^{[m]}$. The concept of resampling according to importance weights lets wrong associations disappear (in expectation) and makes the filter more robust.

4.4 Observation Model

EKF SLAM	**FastSLAM**
The observation model is simply assumed to have an underlying Normal distribution, $$p\left(\mathbf{z}_t \,\vert\, \mathbf{x}_t\right) \sim \mathcal{N}(\mathbf{h}(\mathbf{x}_t), \mathbf{R}_t) .$$ The covariance matrix $\mathbf{R}_t$ is updated according to a Jacobian calculation in every time step.	In addition to the EKF SLAM's observation model, FastSLAM makes explicit use of data association information in n_t: $$p\left(\mathbf{z}_t \,\vert\, \theta_k, \mathbf{s}_t, n_t\right) .$$

5 FastSLAM 1.0 vs. FastSLAM 2.0

Basically, the only modification proposed by Version 2 is that the proposal distribution (2) should not only rely on the previous estimate of the pose $\mathbf{s}_{t-1}$, but also on the actual measurement $\mathbf{z}_t$,

$$\mathbf{s}_t^{[m]} \;\sim\; p\left(\mathbf{s}_t \,|\, \mathbf{s}^{t-1,[m]}, \mathbf{z}^t, n^t\right).$$

5.1 Proposal Distribution

This section roughly introduces the consequences of the new choice for the proposal distribution. It can be expressed as

$$p\left(\mathbf{s}_t \,|\, \mathbf{s}^{t-1,[m]}, \mathbf{z}^t, n^t\right) = \eta^{[m]} \int p\left(\mathbf{z}_t \,|\, \theta_{n_t}, \mathbf{s}_t, n_t\right) \, p\left(\theta_{n_t} \,|\, \mathbf{s}^{t-1,[m]}, \mathbf{z}^{t-1}, n^{t-1}\right) \, d\theta_{n_t} \cdot p\left(\mathbf{s}_t \,|\, \mathbf{s}_{t-1}^{[m]}\right)$$

with

- $p\left(\mathbf{z}_t \,|\, \theta_{n_t}, \mathbf{s}_t, n_t\right) \sim \mathcal{N}(\mathbf{h}\left(\theta_{n_t}, \mathbf{s}_t)\right), \mathbf{R}_t))$
- $p\left(\theta_{n_t} \,|\, \mathbf{s}^{t-1,[m]}, \mathbf{z}^{t-1}, n^{t-1}\right) \sim \mathcal{N}\left(\mu_{n_t,t-1}^{[m]}, \mathbf{\Sigma}_{n_t,t-1}^{[m]}\right)$
- $p\left(\mathbf{s}_t \,|\, \mathbf{s}_{t-1}^{[m]}\right) \sim \mathcal{N}\left(\mathbf{f}\left(\mathbf{s}_{t-1}^{[m]}\right), \mathbf{P}_t\right)$

where $\mathbf{h}(\cdot)$ and $\mathbf{f}(\cdot)$ arise in the general SLAM problem as the measurement model $p\left(\mathbf{z}_t \,|\, \theta, \mathbf{s}_t, n_t\right) = \mathbf{h}\left(\theta_{n_t}, \mathbf{s}_t\right) + \epsilon_t$ and the motion model $p\left(\mathbf{s}_t \,|\, \mathbf{s}_{t-1}\right) = \mathbf{f}\left(\mathbf{s}_{t-1}\right) + \delta_t$, respectively, are introduced. Furthermore, $\epsilon_t \sim \mathcal{N}(0, \mathbf{R}_t)$ and $\delta_t \sim \mathcal{N}(0, \mathbf{Q}_t)$.

However, the new proposal distribution does not have a closed form unless $\mathbf{h}(\cdot)$ is linearized around the landmark position $\mu_{n_t,t-1}^{[m]}$ and the predicted pose estimate $\hat{\mathbf{s}}_t^{[m]}$,

$$\mathbf{h}\left(\theta_{n_t}, \mathbf{s}_t\right) \;\approx\; \hat{\mathbf{z}}_t^{[m]} + \mathbf{H}_\theta \cdot \left(\theta_{n_t} - \mu_{n_t,t-1}^{[m]}\right) + \mathbf{H}_s \cdot \left(\mathbf{s}_t - \hat{\mathbf{s}}_t^{[m]}\right).$$

where we used the predicted measurment $\hat{\mathbf{z}}^{[m]} = \mathbf{h}\left(\hat{\theta}_{n_t}^{[m]}\right)$. $\mathbf{H}_\theta$ and $\mathbf{H}_s$ stand for the respective Jacobians evaluated at $\left(\hat{\mathbf{s}}_t^{[m]}, \hat{\theta}_{n_t}^{[m]}\right)$.

5.2 Importance Weights

The ways the importance weights are calculated for Version 1 and 2 differ insofar that in FastSLAM 2.0 one has to account for the normalizer $\eta^{[m]}$ that was not there before. However, the importance weights in Version 2 are still normally distributed with mean $\hat{\mathbf{z}}_t^{[m]}$ and covariance $\mathbf{H}_s \mathbf{P}_t \mathbf{H}_s^\top + \mathbf{H}_\theta \mathbf{\Sigma}_{n_t,t-1}^{[m]} \mathbf{H}_\theta^\top + \mathbf{R}_t$.

5.3 Unknown Data Association

Similar to Version 1, FastSLAM 2.0 selects that association n_t that maximizes the probability of measurement $\mathbf{z}_t$ for the m-th particle (cf. Section 6.1). However, this probability has to be modified in order to consider the sampled pose. Linearizing $\mathbf{h}(\cdot)$ and calculating the new probability leads to a Gaussian over $\mathbf{z}_t$ with mean $\mathbf{h}\left(\mu_{n_t,t-1}^{[m]}, \mathbf{s}_t^{[m]}\right)$ and covariance $\mathbf{R}_t$.

6 FastSLAM with Unknown Data Association

Solving the unknown data association problem is a crucial step in FastSLAM since otherwise it would not be possible to decompose the posterior (1). Montemerlo et al. suggested four ways to handle this problem [4].

6.1 Per-Particle Maximum Likelihood Data Association

With Bayes and Markov one can show that

$$p\left(n_t \,|\, \mathbf{z}^t\right) \;\propto\; \sum_m p\left(\mathbf{z}_t \,|\, \mathbf{s}_t^{[m]}, n_t\right), \qquad n_t^* = \underset{n_t}{\arg\max}\; p\left(n_t \,|\, \mathbf{z}^t\right),$$

where n_t^* is the ML estimate. This procedure is also adopted by EKFs, even though here it is used on a per-particle basis. Consequences are

a) Noise in the pose estimation will be filtered out, given a reasonable number of particles.

b) Delayed decision-making: pose ambiguities (that appear reasonable at the time the decision is taken) are resolved since they will later receive low importance weights.

6.2 Monte-Carlo Data Association

The method above can be taken a step further: each particle can draw a random association weighted by the probabilities of each landmark having generated the observation. However, uniformly high measurement errors lead to an exponential increase in the number of particles required in comparison to the same scenario with known data association.

6.3 Mutual Exclusion

Mutual Exclusion requires to consider more than one observation per time step and allows to eliminate data association hypotheses that assign multiple measurements to the same landmark. They propose to apply the mutual exclusion constraint in a greedy fashion: each observation is associated with the most likely landmark in each particle that has not received an observation yet (computationally inexpensive since FastSLAM already maintains a set of data association hypotheses). Thus, mutual exclusion forces the creation of a new landmark whenever the observation cannot be explained by the actual associations.

6.4 Negative Information

Areas without (known) landmarks cannot be assumed to be empty, but still one can draw an inference from such information: if the system is expecting to see a particular landmark but actually does not, it should become less confident about the existence of the landmark. This can be achieved by borrowing a technique for making evidence grids (a detailed description is given in [7]).

References

[1] A. Davison. Real-time simultaneous localisation and mapping with a single camera, 2003.

[2] Stuart Kramer. A Bayesian perspective on why the extended Kalman filter fails in passive tracking. Technical report, Air Force Institute of Technology, 1998.

[3] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. 2002.

[4] Michael Montemerlo and Sebastian Thrun. Simultaneous localization and mapping with unknown data association using fastslam. In *ICRA*, pages 1985–1991, 2003.

[5] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *IJCAI*, pages 1151–1156, 2003.

[6] K. Murphy. Bayesian map learning in dynamic environments. 1999.

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

[8] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical Report TR 95-041, 2004.