Autonomous Robots

4: FastSlam algorithm

Miguel Cazorla

Robot Vision Group

http://www.rvg.ua.es

Doctorado Depto. CCIA - Robótica y Visión

Contents

FastSlam

1. Particle filter solution to the SLAM problem.
2. SLAM problem with known correspondences possesses a conditional independence between landmarks, given the robot pose.
3. This make possible to apply Rao-Blacwellized particle filter: uses particle filters to represent posterior over some variables and Gaussians to represent other variables.

FastSlam

1. Particle filters estimate the robot path.
2. For these particles, the individual map errors are conditionally independient.
3. The mapping problem can be factored into many separated problems, one for each feature.
4. To do that, an EKF for each landmark is used.
5. Data association is made in each particle.

SLAM problem

- A solution for the SLAM problem maximizes

$$p(s^t, \Theta | z^t, u^t) \tag{1}$$

- We can assume (without losing generality) that landmarks are observed once each time:

$$p(s^t, \Theta | z^t, u^t, n^t) \tag{2}$$

Formulation: Rao-Blackwellized

$$
\begin{aligned}
p(s^t, \Theta | z^t, u^t, n^t) &= p(s^t | z^t, u^t, n^t) p(\Theta | s^t, z^t, u^t, n^t) && (3) \\
&= p(s^t | z^t, u^t, n^t) \prod_{k=1}^{K} p(\Theta_k | s^t, z^t, u^t, n^t) && (4)
\end{aligned}
$$

Decomposition

- The problem can be decomposed into $K + 1$ estimation problems:
  - Posteriori over the robot path: it is implemented with a particle filter.

$$p(s^t|z^t, u^t, n^t) \tag{5}$$

  - K problems estimating the location of the landmarks: they are estimated with Kalman filters.

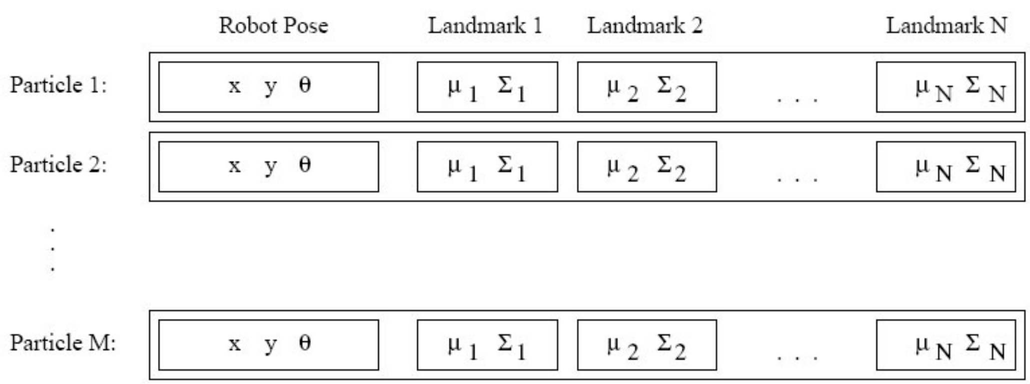
$$p(\Theta_{n_t}|s^t, z^t, u^t, n^t) \tag{6}$$

FastSLAM with known data association

- FastSLAM uses a *particle filter* to estimate the posterior.
- Each particle m in time t is denoted by $s^{t,[m]}$ and it represents a guess of the particle path s^t.
- At time t we have a population of particles $S_t = \left\{s^{t,[m]}\right\}_{m=1}^{M}$

FastSLAM with known data association

- A particle consists of the estimated position of the robot and K EKF estimating the landmarks positions:



How the whole thing works: robot position

- Remember we are in the *known data association* case. We have K landmarks.
- We suppose that the initial set of particles can be generated arbitrarily.
- In each initial particle, the robot position s_0 is $(0, 0, 0)$.
- The robot moves u_t and we use the motion model to generate (for each particle):

$$s_t \sim p(s_t | u_t, s_{t-1}) = h(u_t, s_{t-1}) + \delta_t \tag{7}$$

How the whole thing works: landmarks position

- One EKF (non-linear) for each landmark.
- Landmark positions are represented by the mean $\mu_{n,t-1}^{[m]}$ and the covariance $\Sigma_{n,t-1}^{[m]}$.
- We assume that only one landmark is seen in each observation. At time t, every landmark not seen is not updated, so:

$$\langle \mu_{n,t}^{[m]}, \Sigma_{n,t}^{[m]} \rangle = \langle \mu_{n,t-1}^{[m]}, \Sigma_{n,t-1}^{[m]} \rangle \tag{8}$$

How the whole thing works: landmarks position

- The update is:

$$p(\Theta_{n_t}|s^t, z^t, n^t) = \eta p(z_t|s_t, \Theta_{n_t}, n_t) p(\Theta_{n_t}|s^{t-1}, z^{t-1}, n^{t-1}) \tag{9}$$

where the second term is represented by a Gaussian (the previous mean and variance) and the first term is linearized as EKF does.

How the whole thing works: landmarks position (cont.)

We have to linearized the measurement function g by a first degree Taylor expansion:

$$g(\Theta_{n_t}, s_t^{[m]}) \approx \hat{z}_t^{[m]} + G_t^{[m]}(\Theta_{n_t} - \mu_{n_t,t-1}^{[m]}) \qquad (10)$$

where

$$\begin{aligned} \hat{z}_t^{[m]} &= g(\mu_{n_t,t-1}^{[m]}, s_t^{[m]}) && (11) \\ G_t^{[m]} &= g'(\mu_{n_t,t-1}^{[m]}, s_t^{[m]}) && (12) \end{aligned}$$

How the whole thing works: landmarks position (cont. II)

The new mean and covariance are obtained using the standard EKF measurement update:

$$K_t^{[m]} = \Sigma_{n_t,t-1}^{[m]} G_t^{[m]} (G_t^{[m]T} \Sigma_{n_t,t-1}^{[m]} G_t^{[m]} + R_t)^{-1} \quad (13)$$

$$\mu_{n_t,t}^{[m]} = \mu_{n_t,t-1}^{[m]} + K_t^{[m]} (z_t - \hat{z}_t^{[m]})^T \quad (14)$$

$$\Sigma_{n_t,t}^{[m]} = (I - K_t^{[m]} G_t^{[m]T}) \Sigma_{n_t,t-1}^{[m]} \quad (15)$$

How the whole thing works: weight for each particle

- The weight indicates the goodness of this particle. It is necessary due to the new robot position is calculated based only on u_t, paying no attention to z_t.

- This weight, $w_t^{[m]}$:

$$w_t^{[m]} \approx \eta \det(2\pi Q_t^{[m]})^{-1/2} \exp^{\left\{-\frac{1}{2}(z_t - \hat{z}_t^{[m]})^T Q_t^{[m]-1} (z_t - \hat{z}_t^{[m]})\right\}}$$

with the covariance

$$Q_t^{[m]} = G_t^{[m]T} \Sigma_{n,t-1}^{[m]} G_t^{[m]} + R_t$$

FastSLAM 2.0

- In the previous algorithm, the path posterior (the robot position) is calculated based only on the movement u_t. This is problematic when the accuracy of control is low relative to the accuracy of the robot's sensors.

- This version of FastSLAM takes into account the observations z_t.

Reformulation: robot pose

- Now, the pose is drawn from (including observations):

$$s_t^{[m]} \sim p(s_t | s^{t-1,[m]}, u^t, z^t, n^t) \tag{16}$$

- Rewritten 16 (with a lot of calculus)

$$\begin{gathered} p(s_t | s^{t-1,[m]}, u^t, z^t, n^t) = \\ \eta^{[m]} \int p(z_t | \Theta_{n_t}, s_t, n_t) p(\Theta_{n_t} | s^{t-1,[m]}, z^{t-1}, n^{t-1}) d\Theta_{n_t} \\ p(s_t | s_{t-1}^{[m]}, u_t) \end{gathered} \tag{17}$$

- Convolution of two Gaussians multiplied by a third

Reformulation: robot pose

- How g is not linear so we need to replace g by a linear approximation.

$$g(\Theta_{n_t}, s_t) \approx \hat{z}_t^{[m]} + G_\theta(\Theta_{n_t} - \mu_{n_t,t-1}^{[m]}) + G_s(s_t - \hat{s}_t^{[m]}) \quad (18)$$

where

$$\begin{aligned}
\hat{z}_t^{[m]} &= g(\mu_{n_t,t-1}^{[m]}, \hat{s}_t^{[m]}) && (19) \\
\hat{s}_t^{[m]} &= h(s_{t-1}^{[m]}, u_t) && (20) \\
G_\Theta &= \Delta_{\Theta_{n_t}} g(\Theta_{n_t}, s_t) \mid_{s_t=\hat{s}_t^{[m]};\Theta_{n_t}=\mu_{n_t,t-1}^{[m]}} && (21) \\
G_s &= \Delta_s g(\Theta_{n_t}, s_t) \mid_{s_t=\hat{s}_t^{[m]};\Theta_{n_t}=\mu_{n_t,t-1}^{[m]}} && (22)
\end{aligned}$$

Reformulation: robot pose

- Now the robot pose is sampled by a particle filter with mean and variance:

$$\Sigma_{s_t}^{[m]} = \left[G_s^T Q_t^{[m]-1} G_s + P_t^{-1} \right]^{-1} \tag{23}$$

$$\begin{aligned}
\mu_{s_t}^{[m]} &= \Sigma_{s_t}^{[m]} \left[G_s^T Q_t^{[m]-1} (z_t - \hat{z}_t^{[m]} + G_s \hat{s}_t^{[m]}) \right. \\
&\quad \left. + P_t^{-1} \hat{s}_t^{[m]} \right] \\
&= \Sigma_{s_t}^{[m]} G_s^T Q_t^{[m]-1} (z_t - \hat{z}_t^{[m]}) \\
&\quad + \Sigma_{s_t}^{[m]} \left[G_s^T Q_t^{[m]-1} G_s + P_t^{-1} \right] \hat{s}_t^{[m]} \\
&= \Sigma_{s_t}^{[m]} G_s^T Q_t^{[m]-1} (z_t - \hat{z}_t^{[m]}) + \hat{s}_t^{[m]}
\end{aligned} \tag{24}$$

Reformulation: landmark EKF

$$K_t^{[m]} = \Sigma_{n_t,t-1}^{[m]} G_\Theta^T Q_t^{[m]-1} \tag{25}$$

$$\mu_{n_t,t}^{[m]} = \mu_{n_t,t-1}^{[m]} + K_t^{[m]}(z_t - \hat{z}_t^{[m]}) \tag{26}$$

$$\Sigma_{n_t,t}^{[m]} = (I - K_t^{[m]} G_\Theta)\Sigma_{n_t,t-1}^{[m]} \tag{27}$$

Reformulation: weight for each particle

$$w_t^{[m]} \approx \det(2\pi L_t^{[t]})^{-1/2} \exp^{\left\{-\frac{1}{2}(z_t - \hat{z}_t^{[m]})^T L_t^{[t]-1} (z_t - \hat{z}_t^{[m]})\right\}} \quad (28)$$

$$L_t^{[t]} = G_s P_t G_s^T + G_\Theta \Sigma_{n_t,t-1}^{[m]} G_\Theta^T + R_t \quad (29)$$

FastSLAM with Unknown data association

- Let's move to the next challenge: we do not have the n_t
- We have a new landmark observation z_t but we do not know to which θ_k owns this observation.

FastSLAM with Unknown data association

- We can first choose n_t so that it maximizes the likelihood of the sensor measurement z_t:

$$\hat{n}_t = \arg\max_{n_t} p(z_t|n_t, \hat{n}^{t-1}, s^t, z^{t-1}, u^t) \tag{30}$$

$$p(z_t|n_t, \hat{n}^{t-1}, s^t, z^{t-1}, u^t) = \det(2\pi Q_t^{[m]})^{-1/2} \exp\left\{-\frac{1}{2}(z_t - g(\mu_{n_t,t-1}^{[m]}, s_t^{[m]})^T Q_t^{[m]-1}(z_t - g(\mu_{n_t,t-1}^{[m]}, s_t^{[m]})\right\} \tag{31}$$

Unknown Data association

- A single data association can be easily obtained (nearest neighbor method, using Mahalanobis distance).
- This is the approach followed by EKF-style methods and can be followed here.
- Simply calculate which is the most likely landmark and if the probability in 30 is below a fixed threshold, say p_0, consider the observation as a new landmark

Data association II

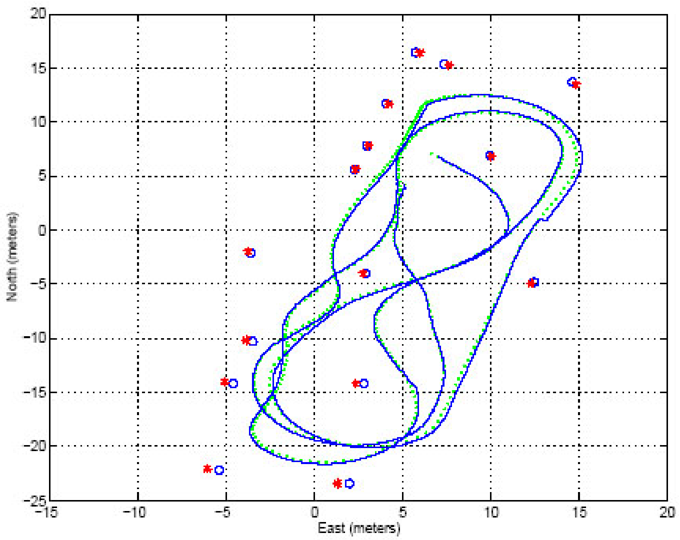
- This can be done using an additional variable ($i_{n,t}^{[m]}$). It counts the number of times a landmark has been seen.
- First, it calculates the w_n for each landmark, given the current observation.
- If all the w_n are below p_0 then the current observation is a new landmark: It generates a new EKF and initializes the counter to 1. The weight for this particle is p_0.
- If not, we associate the current observation to the most likely landmark, updating its EKF and incrementing the counter.

Data association II

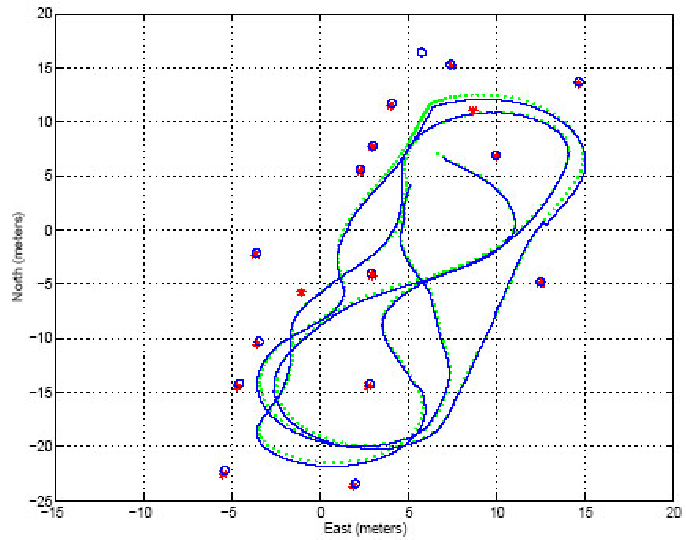
- For the rest of landmarks, we check for the landmarks we would see in our current position. If a landmark is in our observation space and it has not been observed, decrement the counter. If it has, increment the counter.

- When a landmark has its counter below 0, discard this landmark.

Results

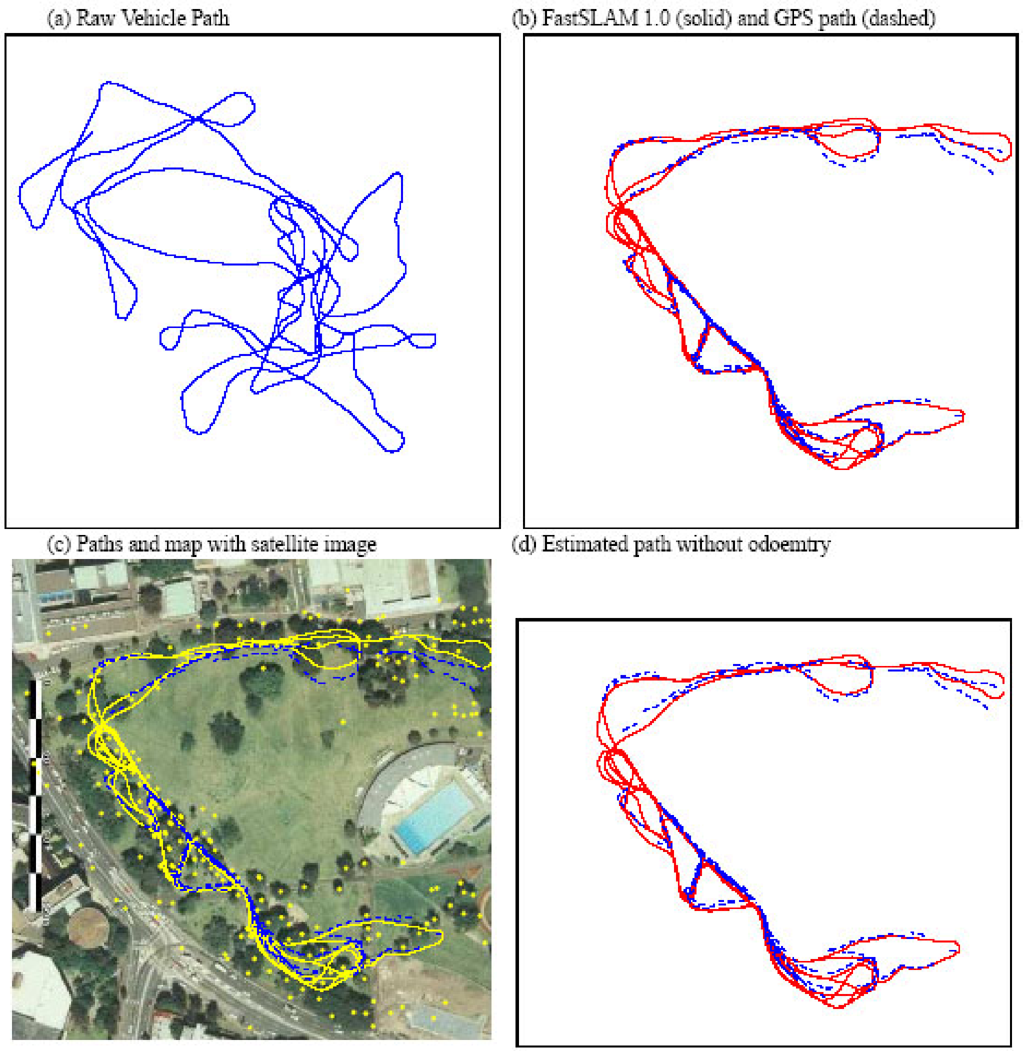


EKF SLAM



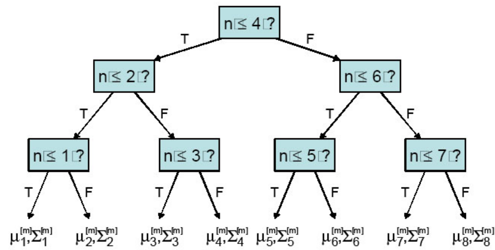
FastSLAM

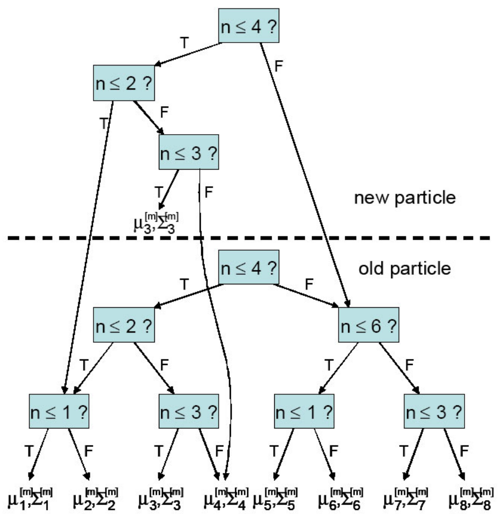
Results II


(a) Raw Vehicle Path
(b) FastSLAM 1.0 (solid) and GPS path (dashed)
(c) Paths and map with satellite image
(d) Estimated path without odoemtry

Implementation details

- KDtree





Conclusions

- Tested in small real-world and huge simulated environments.
- It needs a small number of particles ($M = 100$) even for a large number of landmarks ($K >> M$) (in simulation $K = 50000$ and $M = 250$).
- It's easy to implement (but it requires a lot of assumptions).
- It's able to manage a lot of landmarks (where other approaches don't)

References

- *Probabilistic Robotics*. S. Thrun, W. Burgard, D. Fox, MIT Press. 2005.
- S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto and E. Nebot. *FastSLAM: An efficient solution to the simultaneous localization and mapping problem with unknown data association.* 2004. Journal of Machine Learning Research.