gMapping

Pieter Abbeel
UC Berkeley EECS

For more details, see paper: “Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters” by Giorgio Grisetti, Cyrill Stachniss, Wolfram Burgard, IEEE Transactions in Robotics, 2006

gMapping Overview

- gMapping is probably most used SLAM algorithm
- Implementation available on openslam.org (which has many more resources)
- Currently the standard algorithm on the PR2

Problem Formulation

- Given
  - observations $z_{1:t} = z_1, \ldots, z_t$
  - odometry measurements $u_{2:t} = u_1, \ldots, u_t$
- Find
  - Posterior $p(x_{1:t}, m \mid z_{1:t}, u_{2:t})$
  - With m a grid map

Key Ideas

- Rao-Blackwellized Particle Filter
  - Each particle = sample of history of robot poses + posterior over maps given the sample pose history; approximate posterior over maps by distribution with all probability mass on the most likely map whenever posterior is needed
- Proposal distribution π
  - Approximate the optimal sequential proposal distribution $p^*(x_t) = p(x_t \mid x^i_{1:t-1}, z_{1:t}, u_{1:t}) \propto p(z_t \mid m^i_{t-1}, x_t) \, p(x_t \mid x^i_{t-1}, u_t)$ [note integral over all maps → most likely map only]
    - 1. find the local optimum $\text{argmax}_x \, p^*(x)$
    - 2. sample x^k around the local optimum, with weights $w^k = p^*(x^k)$
    - 3. fit a Gaussian over the weighted samples
    - 4. this Gaussian is an approximation of the optimal sequential proposal p^*
  - Sample from (approximately) optimal sequential proposal
- Weight update for optimal sequential proposal is $p(z_t \mid x^i_{1:t-1}, z_{1:t-1}, u_{1:t}) = p(z_t \mid m^i_{t-1}, x^i_{t-1}, u_{t-1})$, which is efficiently approximated from the same samples as above by
- Resampling based on the effective sample size S_{eff}

Algorithm 1 Improved RBPF for Map Learning

Require:
$\mathcal{S}_{t-1}$, the sample set of the previous time step
z_t, the most recent laser scan
u_{t-1}, the most recent odometry measurement

Ensure:
$\mathcal{S}_t$, the new sample set

$\mathcal{S}_t = \{\}$
for all $s_{t-1}^{(i)} \in \mathcal{S}_{t-1}$ **do**
$< x_{t-1}^{(i)}, w_{t-1}^{(i)}, m_{t-1}^{(i)} >= s_{t-1}^{(i)}$

// scan-matching
$x_t'^{(i)} = x_{t-1}^{(i)} \oplus u_{t-1}$
$\hat{x}_t^{(i)} = \mathrm{argmax}_x \, p(x \mid m_{t-1}^{(i)}, z_t, x_t'^{(i)})$

if $\hat{x}_t^{(i)} =$ **failure then**
$x_t^{(i)} \sim p(x_t \mid x_{t-1}^{(i)}, u_{t-1})$
$w_t^{(i)} = w_{t-1}^{(i)} \cdot p(z_t \mid m_{t-1}^{(i)}, x_t^{(i)})$
else
// sample around the mode
for $k = 1, \ldots, K$ **do**
$x_k \sim \{x_j \mid |x_j - \hat{x}^{(i)}| < \Delta\}$
end for

// compute Gaussian proposal
$\mu_t^{(i)} = (0, 0, 0)^T$
$\eta^{(i)} = 0$
for all $x_j \in \{x_1, \ldots, x_K\}$ **do**
$\mu_t^{(i)} = \mu_t^{(i)} + x_j \cdot p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_t \mid x_{t-1}^{(i)}, u_{t-1})$
$\eta^{(i)} = \eta^{(i)} + p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_t \mid x_{t-1}^{(i)}, u_{t-1})$
end for
$\mu_t^{(i)} = \mu_t^{(i)} / \eta^{(i)}$
$\Sigma_t^{(i)} = \mathbf{0}$
for all $x_j \in \{x_1, \ldots, x_K\}$ **do**
$\Sigma_t^{(i)} = \Sigma_t^{(i)} + (x_j - \mu^{(i)})(x_j - \mu^{(i)})^T \cdot$
$p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_j \mid x_{t-1}^{(i)}, u_{t-1})$
end for
$\Sigma_t^{(i)} = \Sigma_t^{(i)} / \eta^{(i)}$
// sample new pose
$x_t^{(i)} \sim \mathcal{N}(\mu_t^{(i)}, \Sigma_t^{(i)})$

// update importance weights
$w_t^{(i)} = w_{t-1}^{(i)} \cdot \eta^{(i)}$
end if
// update map
$m_t^{(i)} = \mathrm{integrateScan}(m_{t-1}^{(i)}, x_t^{(i)}, z_t)$
// update sample set
$\mathcal{S}_t = \mathcal{S}_t \cup \{< x_t^{(i)}, w_t^{(i)}, m_t^{(i)} >\}$
end for

$N_{\mathrm{eff}} = \frac{1}{\sum_{i=1}^{N} (\tilde{w}^{(i)})^2}$

if $N_{\mathrm{eff}} < T$ **then**
$\mathcal{S}_t = \mathrm{resample}(\mathcal{S}_t)$
end if

Motion Model

- Gaussian (EKF) approximation of odometry model from Probabilistic Robotics, pp. 121-123 [fix slide: for my edition of the book those pages describe the velocity motion model, not the odometry motion model]
- Discrete time steps (=when updates happen) correspond to whenever the robot has traveled about 0.5m
- From paper: "In general, there are more sophisticated techniques estimating the motion of the robot. However, we use that model to estimate a movement between two filter updates which is performed after the robot traveled around 0.5 m. In this case, this approximation works well and we did not observed a significant difference between the EKF-like model and the in general more accurate sample- based velocity motion model [41]"

Scan-Matching

- Find $\operatorname{argmax}_{x\_t}\, p(z_t \mid m^i_{t-1}, x_t)\, p(x_t \mid x^i_{t-1}, u_t)$
- $p(x_t \mid x^i_{t-1}, u_t)$: Gaussian approximation of motion model, see previous slide
- $p(z_t \mid m^i_{t-1}, x_t)$: "any scan-matching technique [...] can be used"
  - Used by gMapping: "beam endpoint model" = likelihood field

More on scan-matching in separate set of slides

Experiments

- "Most maps generated can be magnified up to a resolution of 1cm without observing considerable inconsistencies"
- "Even in big real world datasets covering […] 250m by 250m, [..] never required more than 80 particles to build accurate maps."

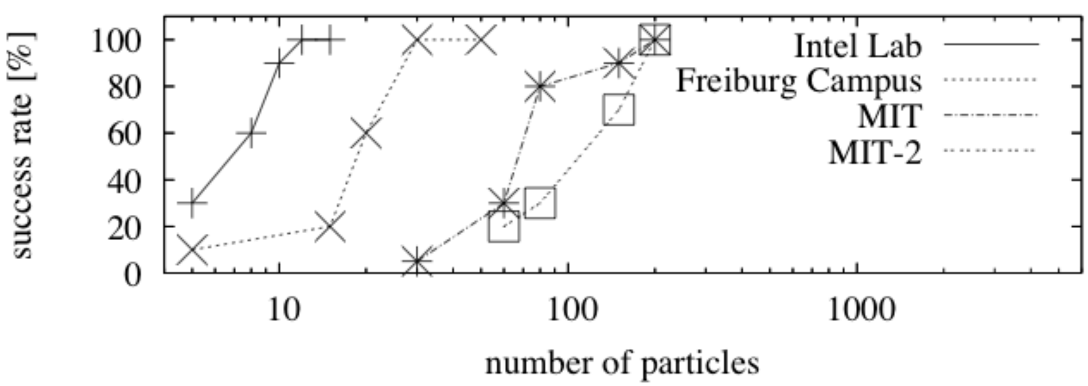

Fig. 7. Success rate of our algorithm in different environments depending on the number of particles. Each success rate was determined using 20 runs. For the experiment MIT-2 we disabled the adaptive resampling.

Correctness evaluated through visual inspection by non-authors

Effect of Proposal Distribution

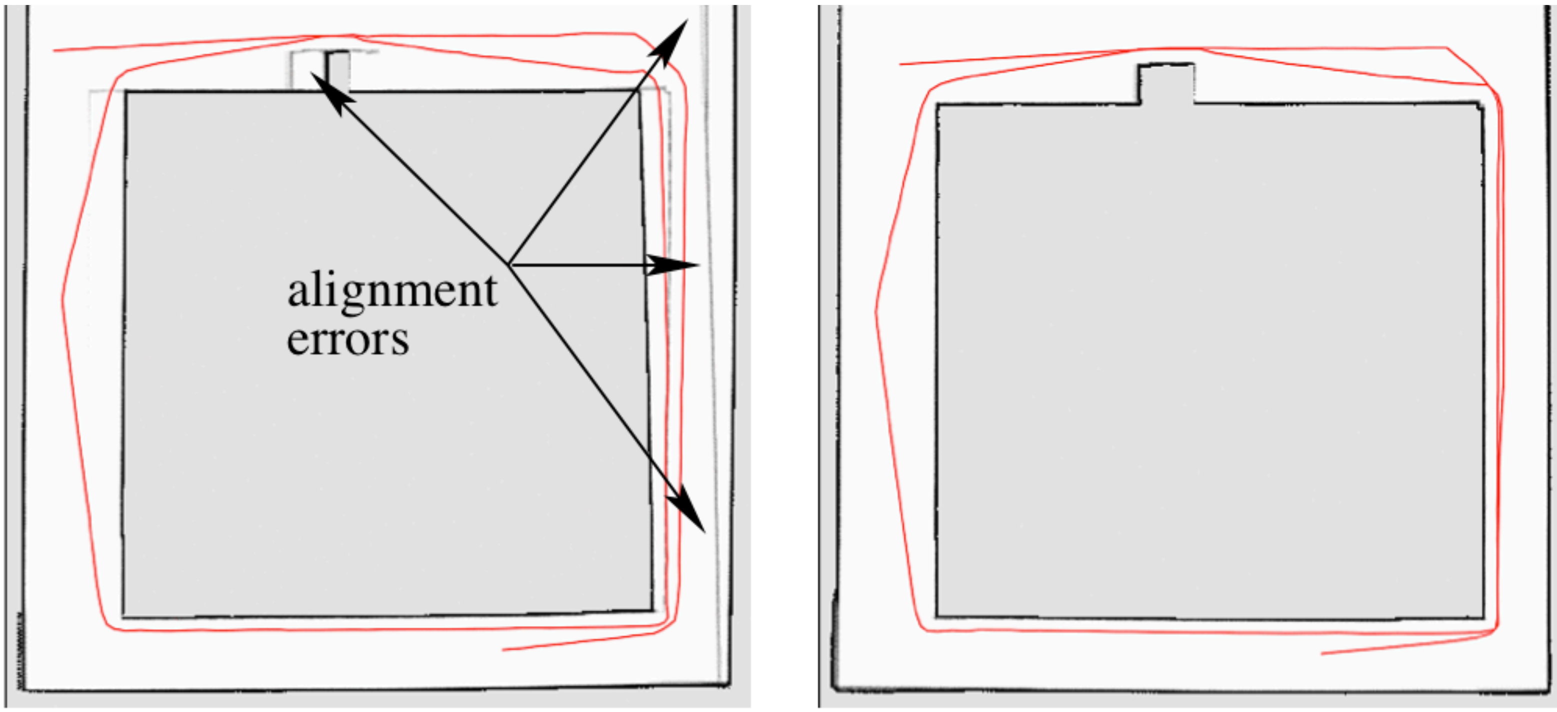

Fig. 11. Different mapping results for the same data set obtained using the proposal distribution which ignores the odometry (left image) and which considers the odometry when drawing the next generation of particles (right