IROS'05 Tutorial

SLAM - Getting it Working in Real World Applications

Rao-Blackwellized Particle Filters and Loop Closing

Cyrill Stachniss and **Wolfram Burgard**

University of Freiburg, Dept. of Computer Science,
Georges-Koehler-Allee 79, D-79106 Freiburg, Germany

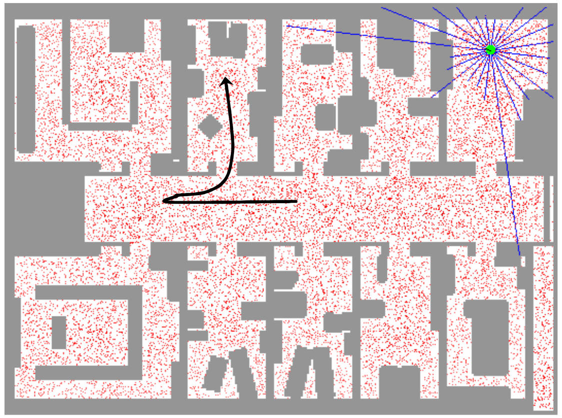
Particle Filters

- Represents a posterior by random samples
- Estimation of non-Gaussian, nonlinear processes
- Set of *N* weighted samples $\{< x^{(1)}, w^{(1)} >, \ldots, < x^{(N)}, w^{(N)} >\}$ containing the **state** x and an **importance weight** w is used to represent the posterior.
- **Sampling**: Create the next generation of particles
- **Weighting**: Assign an important weights to the particles (according to an observation)
- **Resampling**: Draw *N* samples from the set according to the individual importance weights

Monte-Carlo Localization

- For each **motion** Δ do:
  - **Sampling**: Generate from each sample in a new sample according to the motion model $$x^{(i)} \leftarrow x^{(i)} + \Delta'$$
- For each **observation** ***s*** do:
  - **Weigh** the samples with the observation likelihood $$w^{(i)} \leftarrow P(z \mid x^{(i)})$$
  - **Resampling**

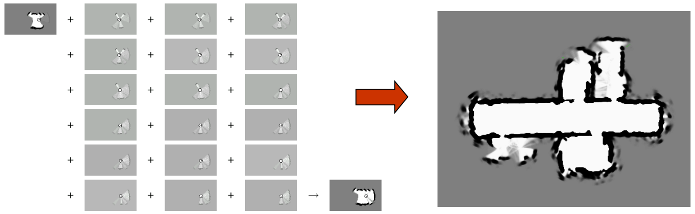
MCL: Global Localization (Sonar)



[Fox et al., 99]

Occupancy Grids

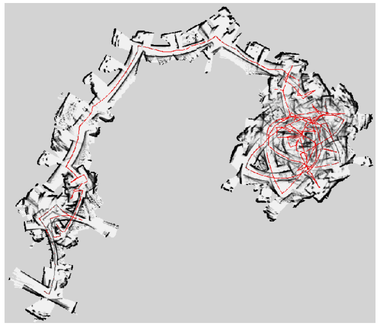
- Grid maps are a discretization of the environment into free and occupied cells
- Mapping with known robot poses is easy.



5

Mapping using Raw Odometry

- Why is SLAM hard? Chicken and egg problem:
  - a map is needed to localize the robot and
  - a pose estimate is needed to build a map



6

Grid-based SLAM

- Particle filters have successfully been applied to localization, can we use them to solve the SLAM problem?
- Posterior over poses x and maps m

$p(x \mid m, z, u)$ (localization) ➡ $p(x, m \mid z, u)$ (SLAM)

Observations:

- The map depends on the poses of the robot during data acquisition
- If the poses are known, mapping is easy

7

Factorization of the Posterior

poses map observations & odometry

$$p(x, m \mid z, u)$$

$$= p(m \mid x, z, \cancel{u})\, p(x \mid z, u)$$

Mapping with known poses

Particle filter representing trajectory hypotheses

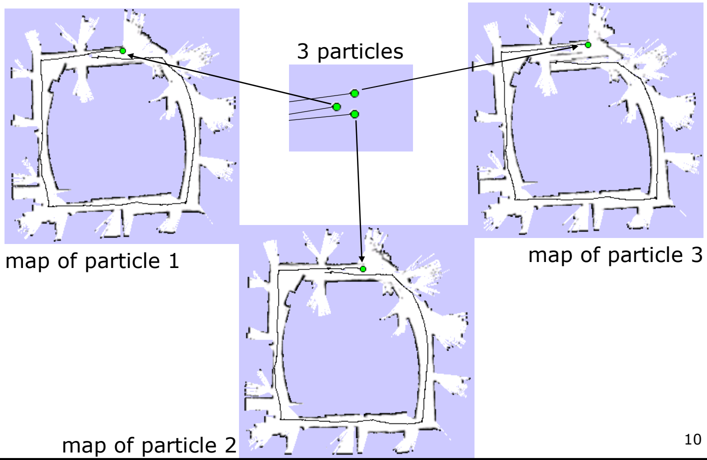
Factorization first introduced by Murphy in 1999

8

Rao-Blackwellized Mapping

- Each particle **represents a possible trajectory** of the robot
- Each particle
  - **maintains its own map** and
  - updates it upon "**mapping with known poses**"
- Each particle **survives with a probability proportional to the likelihood of the observations** relative to its own map

Particle Filter Example



Problem

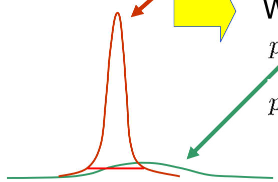
- Huge space complexity: each map is big and each particle maintains its own map
- Therefore, one needs to keep the number of particles small
- **Our Solution**:
  Improved proposal distributions reduce the number of particles needed to build an accurate map!

Improving the Proposal Distribution

$$p(x_t|x_{t-1}^{(i)}, m^{(i)}, z_t, u_t) = \frac{p(z_t|x_t, m^{(i)})p(x_t|x_{t-1}^{(i)}, u_t)}{\int p(z_t|x_t, m^{(i)})p(x_t|x_{t-1}^{(i)}, u_t)dx_t}$$

[Arulampalam et al., 01]

For lasers $p(z_t|x_t, m^{(i)})$ is extremely peaked and dominates the product.



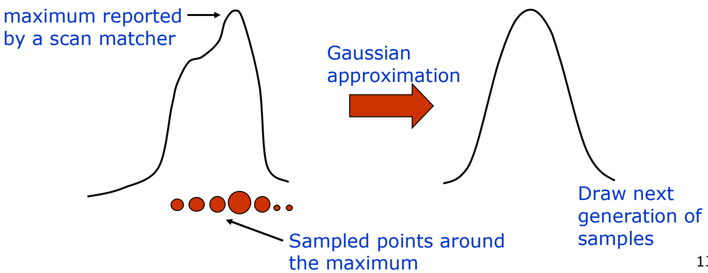
We can safely approximate $p(x_t|x_{t-1}^{(i)}, u_t)$ by a constant:

$$p(x_t|x_{t-1}^{(i)}, u_t)\,|_{x_t: p(z_t|x_t, m^{(i)}) > \epsilon} = c$$

Resulting Proposal Distribution

$$p(x_t|x_{t-1}^{(i)}, m^{(i)}, z_t, u_t) \simeq \frac{p(z_t|x_t, m^{(i)})}{\int_{x_t \in \{x | p(z_t|x, m^{(i)}) > \epsilon\}} p(z_t|x_t, m^{(i)}) dx_t}$$

Approximate this equation by a Gaussian:



13

Resulting Proposal Distribution

$$p(x_t|x_{t-1}^{(i)}, m^{(i)}, z_t, u_t) \simeq \frac{p(z_t|x_t, m^{(i)})}{\int_{x_t \in \{x | p(z_t|x, m^{(i)}) > \epsilon\}} p(z_t|x_t, m^{(i)}) dx_t}$$

Approximate this equation by a Gaussian:

$$p(x_t|x_{t-1}^{(i)}, m^{(i)}, z_t, u_t) \simeq \mathcal{N}(\mu^{(i)}, \Sigma^{(i)})$$

$$\mu^{(i)} = \frac{1}{\eta} \sum_{j=1}^{K} x_j \, p(z_t|x_j, m^{(i)})$$

$$\Sigma^{(i)} = \frac{1}{\eta} \sum_{j=1}^{K} (x_j - \mu^{(i)})(x_j - \mu^{(i)})^T p(z_t|x_j, m^{(i)})$$

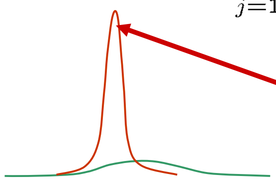
h is a normalizer

Sampled around the scan-match maxima

14

Computing the Importance Weight

$$\begin{aligned} w_t^{(i)} &= w_{t-1}^{(i)} p(z_t|x_{t-1}^{(i)}, m^{(i)}) \\ &\simeq w_{t-1}^{(i)} \int p(z_t|x_t, m^{(i)}) p(x_t|x_{t-1}^{(i)}, u_t) dx_t \\ &\simeq w_{t-1}^{(i)} c \int_{x_t \in \{x | p(z_t|x, m^{(i)}) > \epsilon\}} p(z_t|x_t, m^{(i)}) dx_t \\ &\simeq w_{t-1}^{(i)} c \sum_{j=1}^{K} p(z_t|x_j, m^{(i)}) \end{aligned}$$

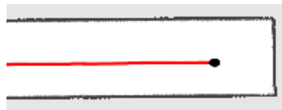


Sampled points around the maximum of the observation likelihood

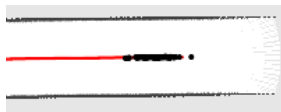
15

Incorporating the Measurements

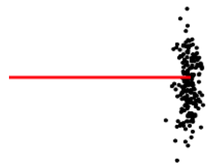
End of a corridor:



Corridor:



Free space:



16

Selective Re-sampling

- Re-sampling is dangerous, since important samples might get lost (particle depletion problem)
- In case of suboptimal proposal distributions re-sampling is necessary to achieve convergence.
- Key question:
  When should we re-sample?

17

Number of Effective Particles

$$n_{eff} = \frac{1}{\sum_i \left(w^{(i)}\right)^2}$$

particle weights

- Empirical measure of how well the goal distribution is approximated by samples drawn from the proposal
- n_{eff} is maximal for equal weights. In this case, the distribution is close to the proposal
- n_{eff} is closely related to the variance of the particle weights

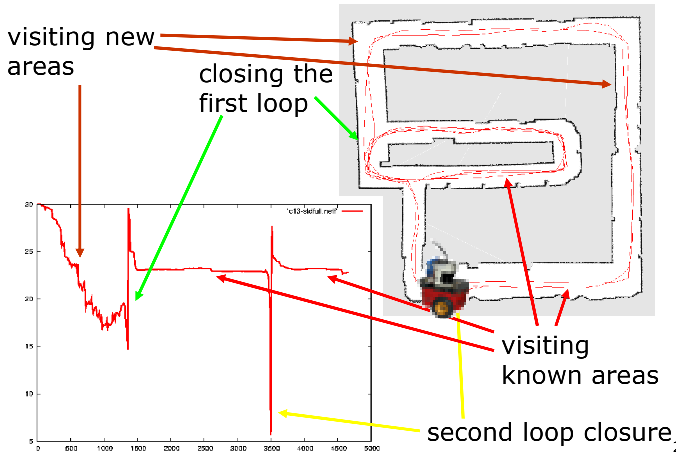
18

Resampling with Neff

- If our approximation is close to the proposal, no resampling is needed
- We only re-sample when n_{eff} drops below a given threshold (n/2)
- See [Doucet, '98; Arulampalam, '01]

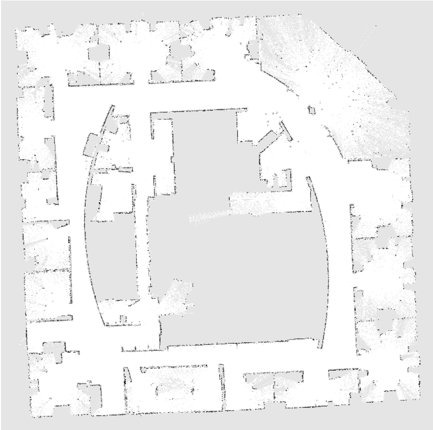
19

Typical Evolution of n_{eff}



20

Intel Lab



- **15 particles**
- four times faster than real-time P4, 2.8GHz
- 5cm resolution during scan matching
- 1cm resolution in final map

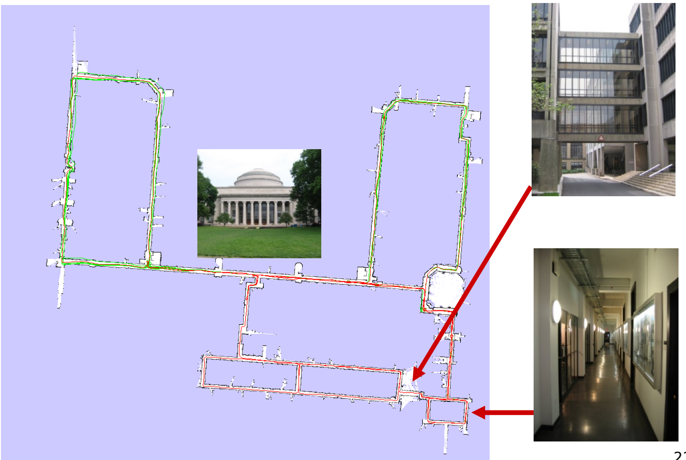
21

Outdoor Campus Map

- **30 particles**
- 250x250m^2
- 1.75 km (odometry)
- 20cm resolution during scan matching
- 30cm resolution in final map

22

MIT Killian Court



23

Conclusion (RBPF-SLAM)

- A Rao-Blackwellized particle filter is a great tool to solve the SLAM problem using grid maps
- Utilizing accurate sensor observation leads to good proposals and highly efficient filters
- It is similar to scan-matching on a per-particle base
- The number of necessary particles and re-sampling steps can seriously be reduced
- Improved versions of grid-based RBPF-SLAM can handle larger environments than naïve implementations in "real time" since they need one order of magnitude fewer samples

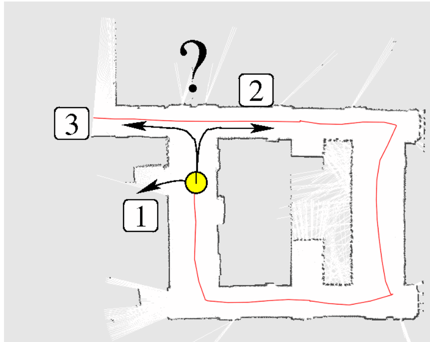
24

Exploration

- The technique seen so far is purely passive
- By reasoning about control, the mapping process can be made more effective
- Question: **Where to move next?**

25

Where to Move Next?



26

Decision-Theoretic Approach

- Apply an exploration approach that minimizes the overall uncertainty in the Rao-Blackwellized particle filter
- The uncertainty of a RBPF has two components:
  - map uncertainty and
  - pose uncertainty
- Utility = Uncertainty Reduction - Cost

27

Computing the Map and Pose Uncertainty

data (laser and odometry)

$$H(p(x, m \mid d))$$

$$= H(p(x \mid d)) + \int_x p(x \mid d) H(p(m \mid x, d))\, dx$$

$$\approx H(p(x \mid d)) + \sum_{i=1}^{\#particles} \omega^{(i)} H(p(m^{(i)} \mid x^{(i)}, d))$$

trajectory uncertainty

particle weight

map uncertainty

28

Computing the Entropy of the Map Posterior

Occupancy Grid map *m*:

$$H(p(m)) = -\sum_{c \in m} p(c) \log p(c) + (1-p(c)) \log(1-p(c))$$

map uncertainty

grid cells

probability that the cell is occupied

29

Computing the Entropy of the Trajectory Posterior

1. High-dimensional Gaussian

$$H(\mathcal{G}(\mu, \Sigma)) = \log((2\pi e)^{(n/2)} |\Sigma|)$$

reduced rank for sparse particle sets

2. Grid-based approximation

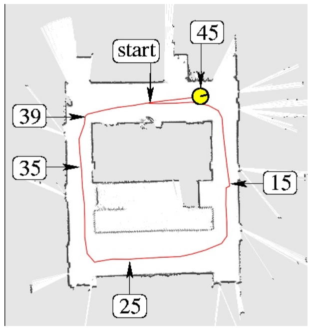
$$H(p(x \mid d)) \rightsquigarrow const.$$

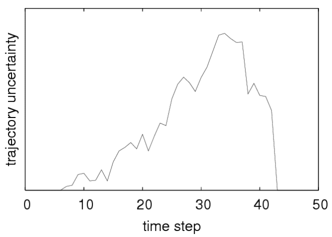
for sparse particle sets

30

Approximation of the Trajectory Posterior Entropy

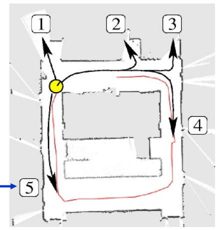
Average pose entropy over time [Roy et al., 98]:





31

Information Gain



Observations obtained when executing a

action

e.g.

$$I(\hat{z}, a) = H(p(m, x \mid d)) - H(p(m, x, \hat{x} \mid d, a, \hat{z}))$$

H before action is carried out

H after action is carried out

new poses introduced by the action a

32

Computing the Expected Information Gain

- To compute the information gain one needs to know the observations obtained when carrying out an action

- This quantity is not known! Reason about potential measurements

$$E[I(a)] \;=\; \int_{\hat{z}} p(\hat{z} \mid a, d) \cdot I(\hat{z}, a)\, d\hat{z}$$

33

Reasoning about Measurements

- The filter represents a posterior about possible maps
- Use these maps to reason about possible observation
- Simulate laser measurements in the maps of the particles

$$E[I(a)] \;=\; \int_{\hat{z}} p(\hat{z} \mid a, d) \cdot I(\hat{z}, a)\, d\hat{z}$$

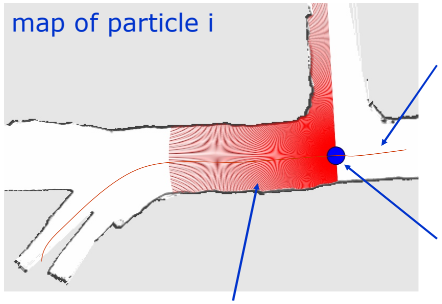
measurement sequences simulated in the maps

likelihood (particle weight)

34

Reasoning about Measurements

- Ray-casting in the map of each particle to generate observation sequences



planned trajectory (action)

pose of particle i while carrying out the action

simulated scan

35

The Utility

- To take into account the cost of an action, we compute a utility

$$U(a) \;=\; I(a) - \alpha \cdot cost(a)$$

- Select the action with the highest expected utility

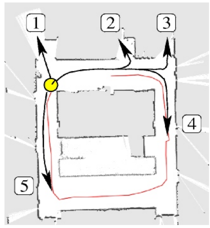
$$a^* \;=\; \underset{a}{\text{argmax}}\{E[U(a)]\}$$

36

Focusing on Specific Actions

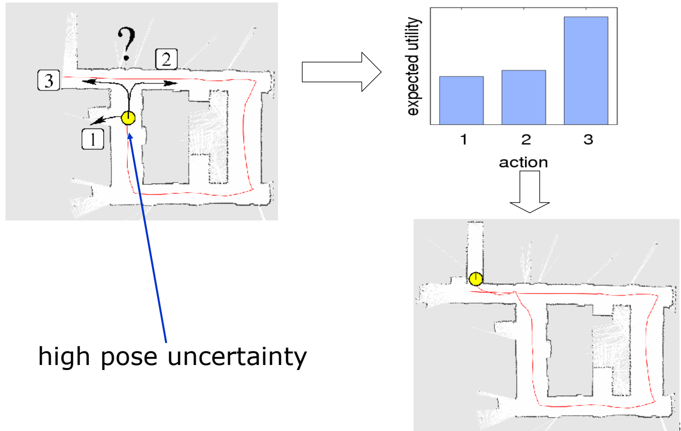
To efficiently sample actions, we consider

- **exploratory actions (1-3)**
- **loop closing actions (4)** and
- **place revisiting actions (5)**



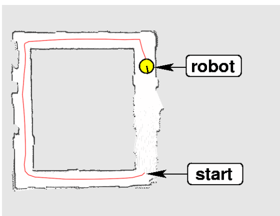
37

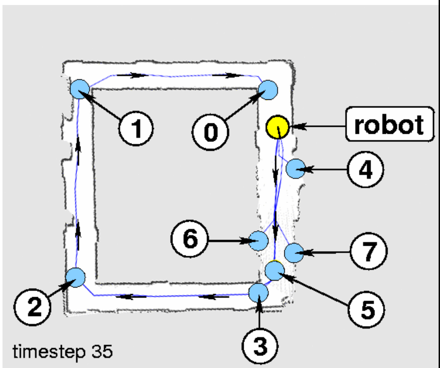
Application Example



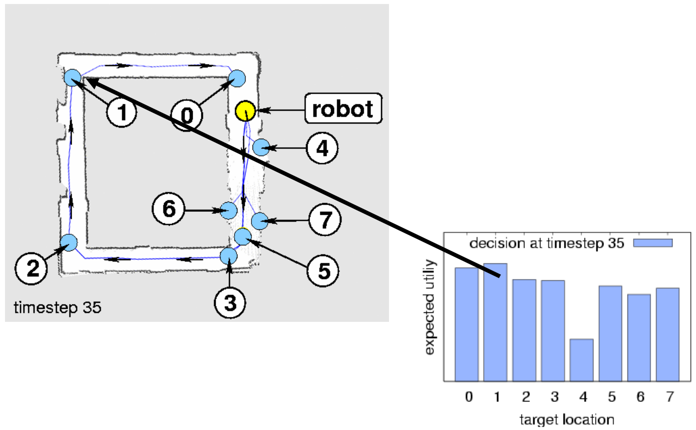
8

Example: Possible Targets



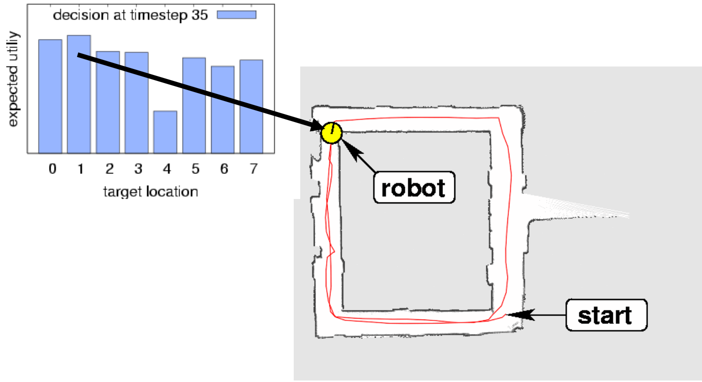


Example: Evaluate Targets

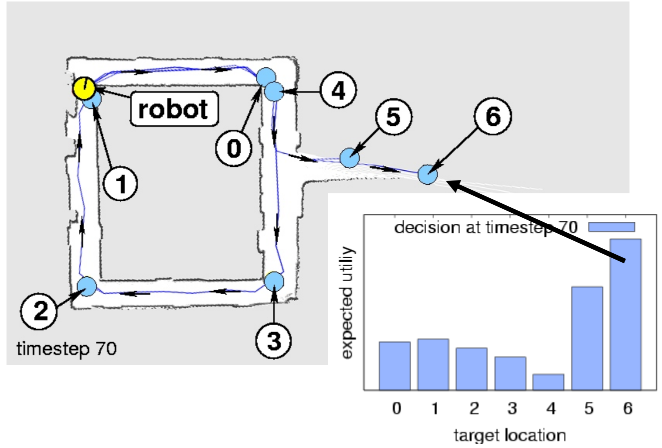


40

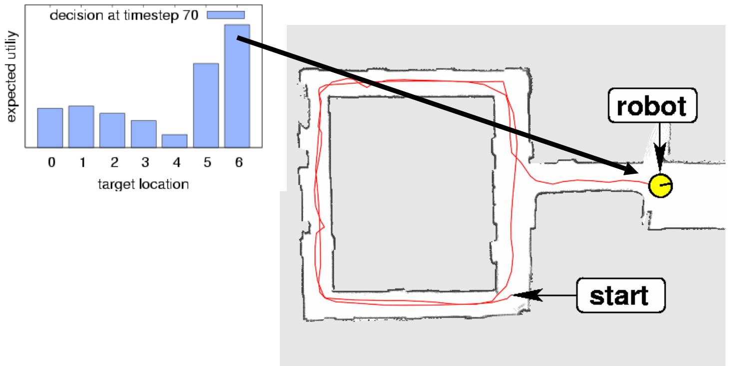
Example: Move Robot to Target



Example: Evaluate Targets

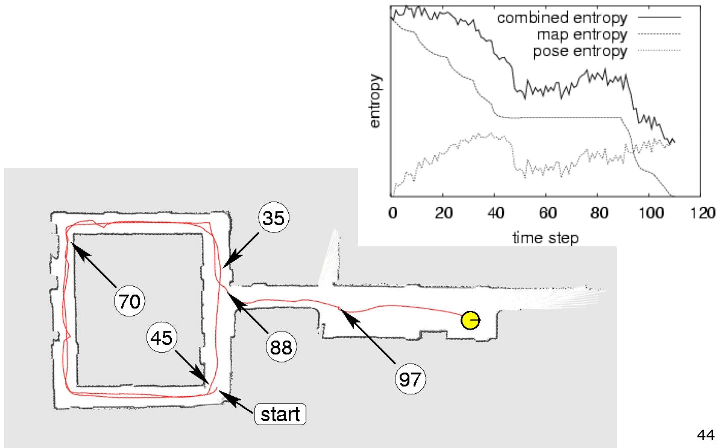


Example: Move Robot

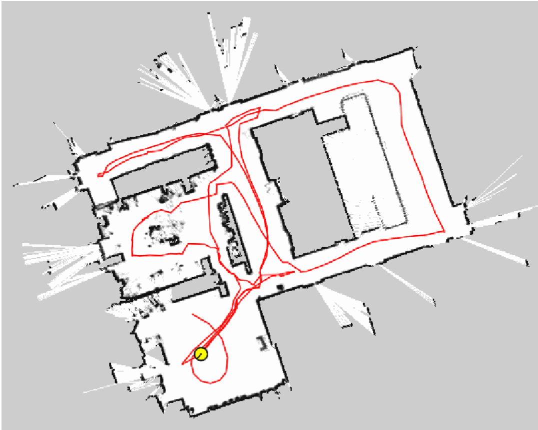


... continue ...

Example: Entropy Evolution



Real Exploration - Video

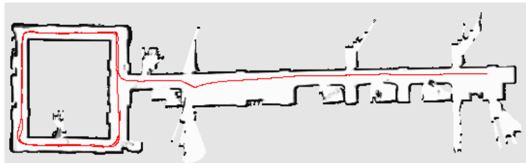


45

Exploration Comparison

Map uncertainty only:

After loop closing action:



46

Conclusion (Exploration)

- We presented a decision-theoretic approach to exploration in the context of RBPF-SLAM
- We reason about observation sequences obtained along the path of the robot
- We presented a way to compute the uncertainty for a RBPF (map and trajectory uncertainty)
- We consider a reduced action set consisting of exploration, loop-closing, and place-revisiting actions
- Experimental results demonstrate the usefulness of the overall approach

47

More Details on RBPF-SLAM

- K. Murphy. Bayesian map learning in dynamic environments, *NIPS99*.
  (First work on using Rao-Blackwellized particle filters for map learning)
- M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to simultaneous localization and mapping, *AAAI02*
  (The classic FastSLAM paper with landmarks)
- D. Haehnel, W. Burgard, D. Fox, and S. Thrun. An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements, IROS03
  (FastSLAM on grid-maps using scan-matched input)
- A. Eliazar and R. Parr. DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks, IJCAI03
  (Improved representation to handle big particle sets)
- G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling, ICRA05
  (Proposal using laser observations and adaptive resampling)
  open-source-implementation at:
  http://www.informatik.uni-freiburg.de/~stachnis/research/rbpfmapper/
- C. Stachniss, G. Grisetti, and W. Burgard. Information Gain-based Exploration Using Rao-Blackwellized Particle Filters, RSS05

48