Simultaneous Localization and Mapping

SLAM with a particle filter

SLAM as a localization problem

We can think of SLAM as a (passive, global, probabilistic) localization problem with a huge state space.

Example:

- $M =$ space of all maps
- $X = \mathbb{R}^2 \times M$

If we know the robot's state in this space, then we know both the robot's position and the map.

Have we already solved this?

We might be tempted to use standard localization methods such as particle filters with this state space to compute:

$$P(x_1, \ldots, x_k, m \mid u_1, \ldots, u_k, y_1, \ldots, y_k)$$

However, in practice, the dimension is too high!

Bad news: For a robot moving in an $n \times n$ grid representation for k steps, we get a $n^2 + 2k$ dimensional state space. (!)

> The number of particles needed to maintain an accurate distribution over this space would be truly ridiculous.

Rao-Blackwellization

Big idea that makes particle-filter-based SLAM actually work:

> Given the path of the robot, we can compute the most likely map.

Rao-Blackwellization

Big idea that makes particle-filter-based SLAM actually work:

> Given the path of the robot, we can compute the most likely map.

We just saw how to do this in the previous section!

Rao-Blackwellization

Big idea that makes particle-filter-based SLAM actually work:

> Given the path of the robot, we can compute the most likely map.

Maintain a particles from the space of robot paths:

$$P(x_1, \ldots, x_k \mid u_1, \ldots, u_k, y_1, \ldots, y_k)$$

Each particle is a single path for the robot:

$$\boldsymbol{s}_j = (x_1^j, \ldots, x_k^j)$$

Factoring of the state space into one part that we sample (the path) and one part that we derive (the map) is called **Rao-Blackwellization**.

Particle Filtering for SLAM

Rao-Blackwellized particle filter:

1. For each particle, compute the most likely map:

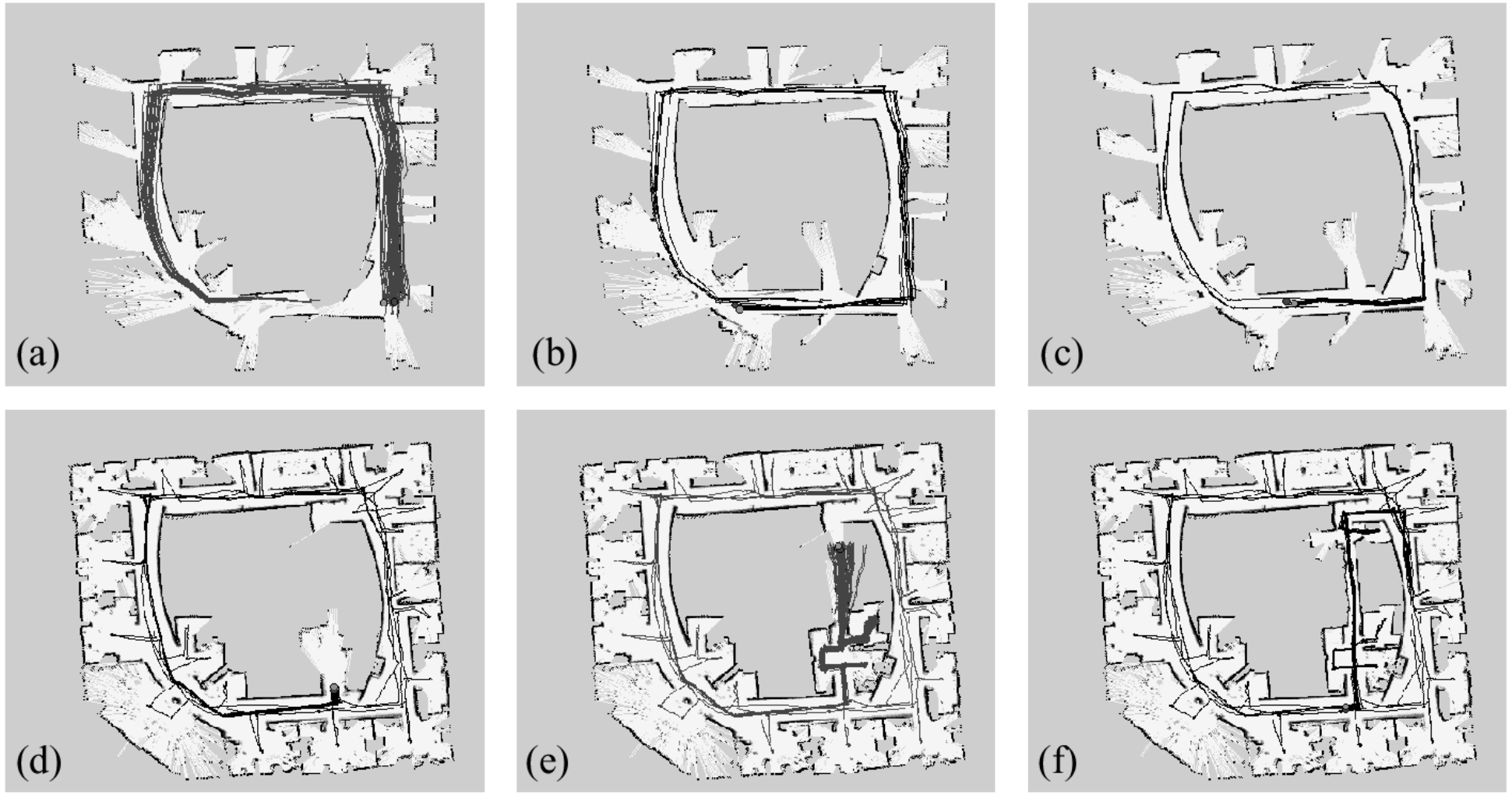
$$m(j,k) = \underset{m \in M}{\operatorname{argmax}} P(m \mid x_1^j, \ldots, x_k^j, y_1, \ldots, y_k)$$

2. Use this map along with the observation model to compute the weight for each particle:

$$w_j = P(y_k \mid x_k^j, m(j,k))$$

3. Resample using these weights as in the standard particle filter.

SLAM Example



SLAM Example