Simultaneous Localization and Dynamic State Estimation in Reconfigurable Environments

Gian Diego Tipaldi  Daniel Meyer-Delius
Maximilian Beinhofer  Wolfram Burgard

Autonomous Intelligent Systems Lab
University of Freiburg, Germany

KUKA Laboratories GmbH
Autonomous Navigation

- Robot must:
  - Know its position in the world and where to go
  - Have a common reference frame with the user
  - Avoid obstacles on the way

- In “robotic” words:
  - Map building
  - Global localization
  - Obstacle avoidance
Autonomous Navigation

- Robot must:
  - Know its position in the world and where to go
  - Have a common reference frame with the user
  - Avoid obstacles on the way

- In “robotic” words:
  - Map building
  - Global localization
  - Obstacle avoidance
Typical Assumption

- The environment does not change over time

- Typical approach
  - Drive the robot around to collect data
  - Use the data to build a map of the environment
  - Localize the robot using that map

- Problem: environments DO change
  - Robot get confused: localization fails
  - Map is out of date: path planning fails
Difficulties of Real Environments

- Environments change over time

**Different kind of change:**
- Fast changes: people, forklifts, doors
- Mid-Term changes: new production lines, doors
- Long-Term changes: new buildings

**Properties of changes**
- Fast: happens very often, similar to outliers
- Mid-Term: happens often, stay for a while
- Long-Term: happens rarely, structural change
Difficulties of Real Environments

- Environments change over time
- Different kind of change:
  - Fast changes: people, forklifts, doors
  - Mid-Term changes: new production lines, doors
  - Long-Term changes: new buildings

- Properties of changes
  - Fast: happens very often, similar to outliers
  - Mid-Term: happens often, stay for a while
  - Long-Term: happens rarely, structural change
How do changes may look like
Naïve 1 - Filter Dynamic Objects
Naïve 1 – Analysis

Advantages
- Efficient
- Global reference
- Fast changes

Disadvantages
- Mid-Term changes
- No map update
Naïve 2 – Infinite SLAM
Naïve 2 – Analysis

Advantages
- Map update (slow)
- Mid-Term changes

Disadvantages
- Fast changes
- Complex
- Local reference
Naïve Solutions - Comparison

**Dynamic Filtering**

**Advantages**
- Efficient
- Global reference
- Fast changes

**Disadvantages**
- Mid-Term changes
- No map update

**Infinite SLAM**

**Advantages**
- Map update (slow)
- Mid-Term changes

**Disadvantages**
- Fast changes
- Complexity
- Local reference
## Naïve Solutions - Comparison

### Dynamic Filtering

**Advantages**
- Efficient
- Global reference
- Fast changes

**Disadvantages**
- Mid-Term changes
- No map update

### Infinite SLAM

**Advantages**
- Map update (slow)
- Mid-Term changes

**Disadvantages**
- Fast changes
- Complexity
- Local reference
Localization and Map Estimation

- **Idea:**
  - Simultaneous Localization and Mapping
  - Reuse the general map structure
  - Model the rate of change of the world

Global localization
Localization and Map Estimation

- **Idea:**
  - Simultaneous Localization and Mapping
  - Reuse the general map structure
  - Model the rate of change of the world

Global localization  
Our approach
Dynamic Occupancy Grid

- HMM represents (for each cell in the grid)
  - Belief about occupancy state, and
  - state transition probabilities

- Specified by
  - Initial state distribution $p(c_0)$
  - Observation model $p(z_t|c_t)$
  - Transition model $p(c_t|c_{t-1})$
Localization and Map Estimation

- Exploit the same factorization of SLAM

\[ p(x_{1:t}, m_t \mid z_{1:t}, u_{1:t}, m_{t-1}) = \]
\[ p(m_t \mid x_{1:t}, z_{1:t}, m_{t-1}) p(x_{1:t} \mid z_{1:t}, u_{1:t}, m_{t-1}) \]

Map estimation \hspace{5cm} \textbf{Localization}

- Use Rao Blackwellized Particle filters
  - Multimodal distribution
  - Uniform prior over pose belief space
Rao Blackwellized Particle Filter

- Each particle stores
  - Robot trajectory
  - Map of the environment
- In global localization
  - High number of particles
  - Multimodal distribution

Algorithm 1: RBPF for Changing Environments

1. $S_t = \{\}$
2. foreach $s_{t-1}^{(i)}$ in $S_{t-1}$ do
3.   $< x_{t-1}^{(i)}, m_{t-1}^{(i)}, w_{t-1}^{(i)} > = s_{t-1}^{(i)}$
4.   $x_t^{(i)} \sim p(x_t \mid x_{t-1}^{(i)}, u_{t-1})$
5.   foreach $c_{t-1}$ in $m_{t-1}^{(i)}$ do
6.     $p(c_t \mid z_{1:t-1}) = \sum_{c_{t-1}\in\{f,o\}} p(c_t \mid c_{t-1})p(c_{t-1} \mid z_{1:t-1})$
7.   end
8.   $w_t^{(i)} = w_{t-1}^{(i)} \prod_j \mathcal{N}(z_t^j; \tilde{z}_t^j, \sigma^2)$
9.   foreach $c_t$ in $m_t^{(i)}$ do
10.   $p(c_t \mid z_{1:t}) = \eta p(z_t \mid c_t)p(c_t \mid z_{1:t-1})$
11. end
12. $S_t = S_t \cup \{< x_t^{(i)}, m_t^{(i)}, w_t^{(i)} >\}$
13. end
14. $N_{\text{eff}} = \frac{1}{\sum_i (w_t^{(i)})^2}$
15. if $N_{\text{eff}} < T$ then
16.   $S_t = \text{resample}(S_t)$
17. end
Rao Blackwellized Particle Filter

- Each particle stores
  - Robot trajectory
  - Map of the environment

- In global localization
  - High number of particles
  - Multimodal distribution

- Problem
  - Memory consumption

Algorithm 1: RBPF for Changing Environments

\[
\begin{align*}
S_t &= \emptyset \\
\text{foreach } s_{t-1}^{(i)} \text{ in } S_{t-1} \text{ do} \\
  &\quad < x_{t-1}^{(i)}, m_{t-1}^{(i)}, w_{t-1}^{(i)} > = s_{t-1}^{(i)} \\
  &\quad x_t^{(i)} \sim p(x_t | x_{t-1}^{(i)}, u_{t-1}) \\
  &\quad \text{foreach } c_{t-1} \text{ in } m_{t-1}^{(i)} \text{ do} \\
  &\quad \quad p(c_t | z_{1:t-1}) = \sum_{c_{t-1} \in \{f,o\}} p(c_t | c_{t-1})p(c_{t-1} | z_{1:t-1}) \\
  &\quad \quad w_t^{(i)} = w_{t-1}^{(i)} \prod_j \mathcal{N}(z^j_t; \bar{z}^j_t, \sigma^2) \\
  &\quad \text{foreach } c_t \text{ in } m_{t}^{(i)} \text{ do} \\
  &\quad \quad p(c_t | z_{1:t}) = \eta p(z_t | c_t)p(c_t | z_{1:t-1}) \\
  &\quad \text{end} \\
  &\quad S_t = S_t \cup \{ < x_{t}^{(i)}, m_{t}^{(i)}, w_{t}^{(i)} > \} \\
  &\text{end} \\
N_{eff} &= \frac{1}{\sum_i (w_t^{(i)})^2} \\
\text{if } N_{eff} < T \text{ then} \\
  &\quad S_t = \text{resample}(S_t) \\
\text{end}
\end{align*}
\]
Rao Blackwellized Particle Filter

Each particle stores
- Robot trajectory
- Map of the environment

In global localization
- High number of particles
- Multimodal distribution

Problem
- Memory consumption

Solution
- HMMs help us!
Stable Distribution & Mixing Time

Stable distribution

- Fixed point solution

\[
\begin{bmatrix}
\pi_f \\
\pi_o
\end{bmatrix} = \frac{1}{p + q} \begin{bmatrix}
q \\
p
\end{bmatrix}
\]

- with

\[
p = p(c_t = o | c_{t-1} = f) \\
p = p(c_t = f | c_{t-1} = o)
\]

Mixing time

- Total variation distance

\[
\Delta_t = |1 - q - p|^t \Delta_0 \\
\Delta_0 = |p(f) - \pi_f|
\]

- We have

\[
t_m = \left[ \frac{\ln(\epsilon/\Delta_0)}{\ln(|1 - p - q|)} \right]
\]
Efficient Map Management

- If not observed, a cell converges to its stable distribution
- The speed of convergences is given by the mixing time

**IDEA**: Only store cells if they differ from the stable distribution
  - Share the stable map
  - Store new cell if observed
  - Forget cell if mixing time is less than 1
Experiments – How to evaluate

- What is the evaluation metric?

- What is the baseline?

- How to set up the evaluation?
Experiments – How to evaluate

- What is the evaluation metric?  
  Localization accuracy and reliability

- What is the baseline?  
  Localization in static environments

- How to set up the evaluation?  
  Static time-slices of dynamic worlds
Experiments – Set up

- Parking lot of university of Freiburg
- Data collected every hour (12 logfiles)
- Procedure
  - Compute the “static” map using SLAM
  - Groundtruth using MCL on the “static” map
- Compare with MCL on the overall occupancy grid
Experiment – Global Localization

- Monte Carlo localization
- Our approach
Experiment – Global Localization

The diagram shows the success rate over different datasets for various algorithms:
- MCL-S
- MCL-D
- RBPF

The success rate is plotted on the y-axis, ranging from 0 to 1.2, while the dataset is shown on the x-axis, ranging from 2 to 12.
Experiment – Global Localization
Experiment – Position Tracking

- Monte Carlo localization
- Our approach
Experiments – Position Tracking

![Graph showing the failure rate across different datasets for various tracking methods. The x-axis represents the dataset number, and the y-axis represents the failure rate. The methods compared include MCL-S, MCL-D, MCL-TM, and RBPF.]
Experiments – Position Tracking

![Graph showing error vs. dataset for different methods: MCL-S, MCL-D, MCL-TM, RBPF.](image)
Conclusions

- Novel localization solution in changing environments
- Directly model the environment dynamics
- Efficient solution using HMMs
- Experiments with real robot shows improved stability and accuracy
Backup Slides
HMM Maps VS. BOF Maps

HMM maps
- Map-centric
- Occupancy “change”
- Need no tracking
- Different transitions

BOF maps
- Object-centric
- Occupancy “move”
- Need tracking
- Null acceleration

\[
T = \begin{bmatrix}
p & 1 - q \\
1 - p & q
\end{bmatrix} \quad T = \begin{bmatrix}
1 - \epsilon & \epsilon \\
\epsilon & 1 - \epsilon
\end{bmatrix}
\]

### HMM Maps VS. Long Term Maps

<table>
<thead>
<tr>
<th>HMM maps</th>
<th>LT maps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>Non-parametric</td>
</tr>
<tr>
<td>Estimated rate</td>
<td>Multiple rates</td>
</tr>
<tr>
<td>Low memory</td>
<td>High memory</td>
</tr>
<tr>
<td>Low overhead</td>
<td>High overhead</td>
</tr>
<tr>
<td>Global localization</td>
<td>Pose tracking</td>
</tr>
</tbody>
</table>

Biber and Duckett. “Experimental Analysis of Sample-Based Maps for Long-Term SLAM”. IJRR 2008
Modeling the Different Dynamics

Free to free

Occupied to occupied
Modeling the Different Dynamics

Occupied to free  Free to occupied