Introduction to Mobile Robotics

Bayes Filter – Discrete Filters

Wolfram Burgard
Cyrill Stachniss
Giorgio Grisetti
Maren Bennewitz
Christian Plagemann
Piecewise Constant
Discrete Bayes Filter Algorithm

1. Algorithm **Discrete_Bayes_filter** (Bel(x), d):
2. \( \eta = 0 \)
3. If \( d \) is a perceptual data item \( z \) then
   4. For all \( x \) do
   5. \( Bel'(x) = P(z \mid x)Bel(x) \)
   6. \( \eta = \eta + Bel'(x) \)
   7. For all \( x \) do
   8. \( Bel'(x) = \eta^{-1}Bel'(x) \)
9. Else if \( d \) is an action data item \( u \) then
10. For all \( x \) do
11. \( Bel'(x) = \sum_{x'} P(x \mid u, x') Bel(x') \)
12. Return \( Bel'(x) \)
Piecewise Constant Representation

\[ Bel(x_t = \langle x, y, \theta \rangle) \]
Implementation (1)

- To update the belief upon sensory input and to carry out the normalization one has to iterate over all cells of the grid.
- Especially when the belief is peaked (which is generally the case during position tracking), one wants to avoid updating irrelevant aspects of the state space.
- One approach is not to update entire sub-spaces of the state space.
- This, however, requires to monitor whether the robot is de-localized or not.
- To achieve this, one can consider the likelihood of the observations given the active components of the state space.
Implementation (2)

- To efficiently update the belief upon robot motions, one typically assumes a bounded Gaussian model for the motion uncertainty.
- This reduces the update cost from $O(n^2)$ to $O(n)$, where $n$ is the number of states.
- The update can also be realized by shifting the data in the grid according to the measured motion.
- In a second step, the grid is then convolved using a separable Gaussian Kernel.
- Two-dimensional example:

```
<table>
<thead>
<tr>
<th>1/16</th>
<th>1/8</th>
<th>1/16</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/8</td>
<td>1/4</td>
<td>1/8</td>
</tr>
<tr>
<td>1/16</td>
<td>1/8</td>
<td>1/16</td>
</tr>
</tbody>
</table>
```

$\approx$

```
<table>
<thead>
<tr>
<th>1/4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2</td>
</tr>
<tr>
<td>1/16</td>
</tr>
</tbody>
</table>
```

+ 

```
| 1/4 | 1/2 | 1/4 |
```

- Fewer arithmetic operations
- Easier to implement
Grid-based Localization
Sonars and Occupancy Grid Map
Tree-based Representation

Idea: Represent density using a variant of octrees
Tree-based Representations

- Efficient in space and time
- Multi-resolution
Xavier:
Localization in a Topological Map