Introduction to Mobile Robotics

Bayes Filter – Discrete Filters

Wolfram Burgard, Cyrill Stachniss, Maren Bennewitz, Diego Tipaldi, Luciano Spinello
\[ \text{Bel}(x \mid z, u) = \alpha p(z \mid x) \int_{x'} p(x \mid u, x') \text{Bel}(x') dx' \]
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

\[ Bel(x_t = <x, y, \theta>) \]
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></th>
<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>
<td></td>
</tr>
<tr>
<td>1/16</td>
<td>1/8</td>
<td>1/16</td>
<td></td>
</tr>
</tbody>
</table>

\[ \begin{array}{c}
\frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\
\frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\
\frac{1}{16} & \frac{1}{8} & \frac{1}{16}
\end{array} \quad \Rightarrow \quad \begin{array}{c}
\frac{1}{4} \\
\frac{1}{2} \\
\frac{1}{4}
\end{array} \quad + \quad \begin{array}{c c c}
\frac{1}{4} & \frac{1}{2} & \frac{1}{4}
\end{array} \]

- Fewer arithmetic operations
- Easier to implement
Grid-based Localization
Application Example: Rhino
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
Summary

- Discrete filters are an alternative way for implementing Bayes Filters
- They are based on histograms for representing the density.
- They have huge memory and processing requirements
- Can easily recover from localization errors
- Their accuracy depends on the resolution of the grid.
- Special approximations need to be made to make this approach having dynamic memory and computational requirements.