Localization in Mobile Robotics

Part I.

Miroslav Kulich

Intelligent and Mobile Robotics Group
Czech Institute of Informatics, Robotics and Cybernetics
Czech Technical University in Prague

Wednesday 5/10/2016
Lecture outline

- Taxonomy
- Continuous localization
  - Iterative Closest Point
  - Iterative Matching Range Point
  - Iterative Dual Correspondence
  - Correlative Scan Matching
- Probabilistic methods
  - Bayes filter
  - Particle filter
  - Histogram filter
  - Kalman filter
  - Extended filter
Localization

• Localization is the problem of robot position estimation
  • relatively with respect to the start robot position
  • in the given coordinate system (in a known map)
• Pose is supposed also with orientation: i.e. \((x, y, \phi)\), resp. \((x, y, z, roll, pitch, yaw)\)
• Input: odometer information, data measurements
Why is localization difficult?

- Position can be measured with no single sensor \(\Rightarrow\) position is determined from sensory data
- But sensors are inaccurate (subject of noise)!
- Control commands are realized inaccurately.
- One single measurement is not enough.
**Taxonomy**

- **Position tracking** *(continuous, local)*
  - The initial position is known.
  - Correction of odometer inaccuracies.
  - The error is bounded.
  - Unimodal distribution (Gaussian).

- **Lost robot problem** *(global, absolute)*
  - The initial position is not known.
  - The error is unbounded.
  - Multi-modal distribution.

- **Kidnapped robot problem**
  - Detection of position errors $\Rightarrow$ correction.
  - Suitable for testing.
Taxonomy

- **Static environment**
  - The robot is the only one what moves.
  - The robot position is the only variable.
  - “Nice” mathematical features.

- **Dynamic environment**
  - Objects or/and other robots move (change their positions/state).
  - Objects: humans, light (for cameras), doors, ...
  - Two approaches
    - Object’s movement can be modeled $\implies$ it is a part of state description of the localization task.
    - Data about moving objects are filtered.
Taxonomy

• **Passive localization**
  - Localization module only passively observes, what is happening - it has no influence on the robot control.
  - Localization is a “complementary product of other task.

• **Active localization**
  - The robot is controlled in order to correct/determine its position.
  - Gives better results . . .
  - . . . but it can’t be applied every time.
  - Goals alternation: the robot is localized sometimes x the primary task is solved sometimes.
  - The goal is generated as a compromise.
Passive localization
Active localization
Taxonomy

- **Single robot**
  - The classical problem.
  - No communication is need, everything at one place.

- **Multiple robots**
  - Every robot is localized separately/independently (it pretends that other robots do not exist).
  - But if the robots are able communicate or detect each other, it is better.
  - Self-localization in the other robot map.
  - My position estimate is known, other’s robot position estimates are known as well as relative positions $\Rightarrow$ optimization („rubber bands“).
We will suppose

- A terrestrial robot operating on a horizontal plane (2D).
- Only one plane of the environment is sensed.
- Majority of objects is static and detectable by sensors.
- Distance measuring sensors are used.
- Passive localization.
- Single robot.
- Continuous, absolute localization, and kidnapped robot.
Continuous localization

- The position is supposed to be in the close neighborhood of the estimate.
- High precision needed (continuous process).
- Input: local a and global maps (scans).
- Output: transformation minimizing measure of maps discrepancy.
Example
Continuous localization

- Many methods differing in the representation of the maps (scans)
- It is always optimization problem.
- Approaches:
  - Localization on occupancy grids.
  - Histograms
  - Localization based on processing of geometric primitives (point-to-line, line-to-line, Hough transform)
Iterative Closest Point (ICP)

(Lu, Milios)

- Iterative
  - The actual scan is interpolated with a polyline.
  - Corresponding points are determined (the smallest distance).
  - Transformation is determined (using least squares).

\[
E_{dist}(T_x, T_y, \omega) = \sum_{i=1}^{n} |R_\omega P + T - P'|^2
\]

- It has the analytical solution:
  \[
  T_x = \bar{x}' - (\bar{x} \cos \bar{\omega} - \bar{y} \sin \bar{\omega})
  \]
  \[
  T_y = \bar{y}' - (\bar{x} \sin \bar{\omega} + \bar{y} \cos \bar{\omega})
  \]
  \[
  \bar{\omega} = \arctan \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i' - \bar{y}') - \sum_{i=1}^{n} (y_i - \bar{y})(x_i' - \bar{x}')} {\sum_{i=1}^{n} (x_i - \bar{x})(x_i' - \bar{x}') + \sum_{i=1}^{n} (y_i - \bar{y})(y_i' - \bar{y}')} \]

- It converges slowly, especially in the rotational part.
Iterative Matching Range Point (IMRP)

(Lu, Milios)

- The procedure is the same as in ICP, except the correspondence rule.
- Consider that translation is irrelevant \( \Rightarrow |P| \approx |P'| \) and \( \tilde{\phi} \approx \phi + \omega \).
- Neighborhood size for correspondence search is decreasing.
- Initially slower convergence than ICP, but it is faster later on.
- IMRP tries to correct translation error with rotation \( \Rightarrow \) instability.
Iterative Dual Correspondence (IDC)

- Combination of the two rules: translation from ICP, rotation from IMRP

**Algorithm**
- For each $P_i$ from the actual scan:
  - Apply the *closest-point* rule to determine the corr. point $P'$
  - Apply the *matching-range-point* rule $\Rightarrow P''$
  - Compute the least-squares solution $(\omega_1, T_1)$ from the set of correspondence pairs $(P, P')$
  - Compute the least-squares solution $(\omega_2, T_2)$ from the set of correspondence pairs $(P, P'')$
  - Form $(\omega_2, T_1)$ as the solution
- Repeat the previous steps until the process converges.

**Improvement:** “Bad” correspondence pairs are discarded.
Correlative Scan Matching
(Edwin B. Olson)

- Shape of a cost function is really complex with number of local extrema.
- Initial position estimation (odometry) can be inaccurate.
- ICP finds a local optimum only.
- Input: two scans
- Goal: find a global optimum
Correlative Scan Matching

Look-table building

- For each point in a map compute the probability that the point corresponds to a reference measurement.
Correlative Scan Matching

Algorithm

- **Brute force**
  - Compute correlation for each pose (including orientation) of the scan.
  - The pose with the highest correlation is chosen.

- **Computing 2D slices**
  - A huge amount of time is spent with scan transformation.
  - Idea: rotation is done in the outer-most loop. The inner loops simply translate the query points.
Correlative Scan Matching

Algorithm

- Multi-level resolution
  - Two lookup tables: low resolution (LRT) and high resolution table (HRT)
  - Cell in LRT is the maximum value of corresponding cells in HRT
  - This guarantees that we will not miss the maximum.
- Algorithm
  - Compute correspondences in LRT.
  - Find the best voxel in $LRT_B$ in LRB not yet considered.
  - If the value of $LRT_B \mid H_{best}$ terminate ($H_{best}$ is the maximum).
  - Evaluate the voxel in HRT and set $H_{best}$ if new maximum is found.
### Correlative Scan Matching

#### Results

- **PC:** Intel Core2 6600 @ 2.4 Ghz

<table>
<thead>
<tr>
<th></th>
<th>ICP</th>
<th>ICL</th>
<th>Hill-Climb</th>
<th>Correlative (Naive)</th>
<th>Correlative (2D Slices)</th>
<th>Correlative (Multi-Res)</th>
<th>Correlative (GPU 7600GS)</th>
<th>Correlative (GPU GTX260)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5 m, 20 deg</td>
<td>56 ms</td>
<td>64 ms</td>
<td>1.1 ms</td>
<td>246 ms</td>
<td>27 ms</td>
<td>8.4 ms</td>
<td>98 ms</td>
<td>26 ms</td>
</tr>
<tr>
<td>2.0 m, 40 deg</td>
<td>99 ms</td>
<td>105 ms</td>
<td>1.3 ms</td>
<td>7512 ms</td>
<td>692 ms</td>
<td>20.8 ms</td>
<td>1563 ms</td>
<td>289 ms</td>
</tr>
<tr>
<td>4.0 m, 90 deg</td>
<td>145 ms</td>
<td>174 ms</td>
<td>1.0 ms</td>
<td>65282 ms</td>
<td>5029 ms</td>
<td>86.1 ms</td>
<td>13166 ms</td>
<td>2012 ms</td>
</tr>
</tbody>
</table>
Gentle introduction to probability theory

- Idea: explicit representation of uncertainty using calculus of the probability theory
- $p(X=x)$ probability that the random variable $X$ is $x$
- $0 \leq p(x) \leq 1$
- $p(true) = 1$, $p(false) = 0$
- $p(A \lor B) = p(A) + p(B) - p(A \land B)$
**Discrete and continuous random variable**

- **Discrete**: $X$ is countable, i.e. $X = x_1, x_2, \ldots, x_n$
- **Spojitá**: $X$ can have an uncountable number of values (from some interval)
- $p$ is probability density
- Various distributions
- Most known: Normal (Gaussian)
- $p(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
Multi-dimensional normal distribution

\[ p(x = x_1, \ldots, x_k) = \frac{1}{\sqrt{(2\pi)^k|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}, \]

- Eigenvalues and eigenvectors of the covariance matrix define an ellipse.
Joint and conditional probability distribution

- \( p(X = x \text{ a } Y = y) = p(x, y) \)
- If \( X \) and \( Y \) are independent then
  \[ p(x, y) = p(x)p(y) \]
- \( p(x|y) \) is probability \( x \) given \( y \)
  \[ p(x|y) = \frac{p(x, y)}{p(y)} \]
  \[ p(x, y) = p(x|y)p(y) \]
- If \( X \) a \( Y \) are independent then
  \[ p(x|y) = p(x) \]
Total probability theorem

Discrete case

$$\sum_x p(x) = 1$$

$$p(x) = \sum_y p(x, y)$$

$$p(x) = \sum_y p(x|y)p(y)$$

Continuous space

$$\int_x p(x) dx = 1$$

$$p(x) = \int_y p(x, y) dy$$

$$p(x) = \int_y p(x|y)p(y) dy$$
Bayes’ theorem

\[ p(x, y) = p(x|y)p(y) = p(y|x)p(x) \]

\[ \Rightarrow \]

\[ p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}} \]

\[ p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \eta p(y|x)p(x) \]

\[ \eta = p(y)^{-1} = \frac{1}{\sum_x p(y|x)p(x)} \]
Simple example of state estimation

- Assume a robot obtains measurement $z$
- What is $p(open|z)$?
- $p(open|z)$ is diagnostic
- $p(z|open)$ is causal
- Often causal knowledge is easier to obtain (counting frequencies)
- Bayes rule allows us to use causal:

$$p(open|z) = \frac{p(z|open)p(open)}{p(z)}$$
Example - open doors

- $p(z|\text{open}) = 0.6$ $p(z|\neg\text{open}) = 0.3$
- $p(\text{open}) = p(\neg) = 0.5$

$$p(\text{open}|z) = \frac{p(z|\text{open})p(\text{open})}{p(z|\text{open})p(\text{open}) + p(z|\neg\text{open})p(\neg\text{open})}$$

$$p(\text{open}|z) = \frac{0.6 \cdot 0.5}{0.6 \cdot 0.5 + 0.3 \cdot 0.5} = \frac{2}{3} = 0.67$$

- $z$ raises probability that the door is open.
Example - second measurement

- $p(z_2|\text{open}) = 0.5$  $p(z_2|\neg\text{open}) = 0.6$
- $p(\text{open}|z_1) = \frac{2}{3}$

\[
p(\text{open}|z_2z_1) = \frac{p(z_2|\text{open})p(\text{open}|z_1)}{p(z_2|\text{open})p(\text{open}|z_1) + p(z_1|\neg\text{open})p(\neg\text{open}|z_1)}
\]
\[
= \frac{\frac{1}{2} \cdot \frac{2}{3}}{\frac{1}{2} \cdot \frac{2}{3} + \frac{3}{5} \cdot \frac{1}{3}} = \frac{5}{8} = 0.625
\]

- $z_2$ lowers the probability that the door is open.
Actions

- Often the world is dynamic since
  - actions carried out by the robot,
  - actions carried out by other agents,
  - or just the time passing by change the world (plants grow).
- Actions are never carried out with absolute certainty.
- In contrast to measurements, actions generally decrease the uncertainty.
- To incorporate the outcome of an action $u$ into the current “belief”, we use the conditional pdf
  \[ p(x|u, x') \]
- This term specifies the pdf that executing $u$ changes the state from $x'$ to $x$. 
Continuing the example - closing the door

\[ p(x|u, x') \] for \( u = "\text{close door}" \)

\[ p(x, u) = \sum_{x'} p(x|u, x') p(x') \]

If the door is open, the action ”close door” succeeds in 90% of all cases.
Continuing the example - closing the door

\[
p(closed|u) = \sum_{x'} p(closed|u, x')p(x') \\
= p(closed|u, open)p(open) + p(closed|u, closed)p(closed) \\
= \frac{9}{10} \cdot \frac{5}{8} + \frac{1}{10} \cdot \frac{3}{8} = \frac{15}{16}
\]

\[
p(open|u) = \sum_{x'} p(open|u, x')p(x') \\
= p(open|u, open)p(open) + p(open|u, closed)p(closed) \\
= \frac{1}{10} \cdot \frac{5}{8} + \frac{0}{10} \cdot \frac{3}{8} = \frac{1}{16} \\
= 1 - p(closed|u)
\]