Direct Embodied Data for Localisation and Mapping

Jakob Schwendner

DFKI Bremen & Universität Bremen
Robotics Innovation Center
Director: Prof. Dr. Frank Kirchner
www.dfki.de/robotics
robotics@dfki.de
Outline

Problem/Approach

Method
- Overview
- System/Contact Point Model
- Odometry Model
- Environment Model
- Measurement Model
- Particle Filter

Localisation Results

SLAM current state

Conclusion/Outlook
Localisation and Mapping in 3D Environments

- Localisation
- Mapping
- Combined Localisation and Mapping
- Visual SLAM works
Motivation for using embodied data

- Could be used in blind scenarios
- Augment visual means of Localisation and Mapping
- Reduce requirements for vision
- Acknowledge the fact that robots have bodies, too
- Things are not always what they look like
Embodied Data in Context

*Embodied Data* is defined as sensory information originated within or on the border of the system in question.

Two categories of Embodied Data: *Direct* and *Indirect*. 
Method

Method Overview

- System/Contact Point Model
- Odometry Model
- Environment Model
- Measurement Model
- Particle Filter
Method

Asguard System/Contact Point Model

- Asguard has five degrees of freedom
- Four Wheels, free body joint
- \( c \in C = (\gamma_1, \ldots, \gamma_4, \beta) \)
- Contact with environment mainly through feet
- Modeling of Contact Points based on \( c \) and orientation \( q \)
- Frames \( W \), \( B \) and \( Y \)
Method

Asguard Odometry Model used for Approach

- Extended 2D skid steering to 3D
- Difference in orientation from IMU
- Travelled distance from wheel turns
- Compensation for center of rotation
Method

Odometry Error Model

- Mixture model
- Gaussian is with covariance $A(d, \text{tilt}, \Delta \theta, 1)^T$
- constant part for modeling slip
- Projection to $Y$ frame
Method

3D Environment Model

- Requirements: Cartesian, fast, handles Test-Track
- Modes: A-priori & Live
- Options
  - Pointcloud: simple, accurate, slow
  - DEM: simple, high information-loss, very fast
  - MLSM: more complex, medium information-loss, fast
Method

Multi Level Surface Maps

- Regular grid cells partitioning xy-plane
- Multiple patches per cell
- Two cell types
  - horizontal patch with $\mu, \sigma$
  - vertical patch $\mu, \sigma, h$

$$m(p, l) = \begin{cases} (z, \sigma) & \text{surface with } z \in [p_z - l/2, p_z + l/2] \\ \emptyset & \text{no surface in interval} \end{cases}$$

Direct Embodied Data for Localisation and Mapping
April 6, 2011
Method

Single wheel contact estimation

IMU ($q$), encoder readings ($c$)
Particle pose ($T$)

For each wheel

- Contact points in $W$ from $T$ and $c$
- Remove unlikely contacts
- Pick contact with lowest $z$ diff to map
- Wheel is valid if all feet have map value

Not needed if contact information available
Method

Robot body measurement

\[
\hat{p}(z_k|m, c, T) = \prod_{(d, \sigma) \in z_k} \phi\left(\frac{d + \xi}{\sigma}\right)
\]

- maximise for \( \xi \) to get \( z \) offset
- probability is not normalised yet
Method

Particle Encoding and Measurement Normalisation

- Particle distribution over pose space \((x, y, \theta)\)
- Carries extra information \((z, \sigma_z)\)
- measurement \(z_k\) and state \(x_k\)
- \(\bar{p}\) discounted probability of found contacts
- \(p(z_k|x_k^{[m]}) = \hat{p}(z_k|x_k^{[m]})\bar{p}^{4-|z_k|}\)
- normalisation factors contacts per pose sample
Method

Description of Particle Filter used

- Sampling Importance Resampling (SIR) filter
- Initial particles created with given distribution
- Project particles using odometry
- Update particle weight based on $p(z_k|x_k)$
- Update particle $z$ and $\sigma$
- Mark floating particles
- Resample if Effective Particle measure fall below threshold
Localisation Results

Sand Field Experiments

- 50 m x 30 m sand field
- height variation up to 1 m
- a-priori map
- grid spacing 0.05 m
Localisation Results

Track lap (125 m)

Position Plot

- start/end position
- gps
- odometry
- filter centroid
Localisation Results

Track cross (88 m)

Position Plot

- North (m)
- East (m)

- gps
- odometry
- filter centroid
Localisation Results

Side Loop (143 m)

Position Plot

- North (m)
- East (m)

- gps
- odometry
- filter centroid
Localisation Results

Side Loop (143 m) vs time

![Graphs showing localisation results for side loop.](image)
## Localisation Results

### Total Distance Travelled

<table>
<thead>
<tr>
<th>Run</th>
<th>Distance Travelled [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Centroid</td>
</tr>
<tr>
<td>Lap1</td>
<td>125.83</td>
</tr>
<tr>
<td>Lap2</td>
<td>128.28</td>
</tr>
<tr>
<td>Lap3</td>
<td>124.81</td>
</tr>
<tr>
<td>Side Loop</td>
<td>136.84</td>
</tr>
<tr>
<td>Cross</td>
<td>89.67</td>
</tr>
</tbody>
</table>
## Localisation Results

### Position Error

<table>
<thead>
<tr>
<th>Run</th>
<th>Mean Position Error [m]</th>
<th>Max Error [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Centroid</td>
<td>Odometry</td>
</tr>
<tr>
<td>Lap1</td>
<td>0.35</td>
<td>8.74</td>
</tr>
<tr>
<td>Lap2</td>
<td>0.37</td>
<td>9.34</td>
</tr>
<tr>
<td>Lap3</td>
<td>0.36</td>
<td>10.33</td>
</tr>
<tr>
<td>Side Loop</td>
<td>0.49</td>
<td>4.29</td>
</tr>
<tr>
<td>Cross</td>
<td>0.40</td>
<td>3.23</td>
</tr>
</tbody>
</table>
Localisation Results

Error vs Particle Count

![Graph showing error vs particle count for filter and odometry methods.]
SLAM current state

Mapping

- Use Laserscanner for Mapping
- Uncertainty transformation into map
- One map per particle
- Work in progress . . .
Conclusion/Outlook

Concluding the work and further steps

- Approach improves localisation over odometry alone
- Localisation filter has bounded error
- Should benefit from improved odometry
- Look into indirect embodied data
- Combine vision and embodied data to improve SLAM
Thank you for your attention!