Semantic Mapping – Some Details

Joachim Hertzberg

Osnabrück University
and DFKI Robotics Innovation Center
(Own) Material

• H. Surmann, A. Nüchter, J. Hertzberg. 
  An Autonomous Mobile Robot with a 3D Laser Range Finder for 3D 
  Exploration and Digitalization of Indoor Environments. 

• A. Nüchter, J. Hertzberg. 
  Towards Semantic Maps for Mobile Robots. 

• M. Günther, T. Wiemann, S. Albrecht, J. Hertzberg. 
  Building Semantic Object Maps from Sparse and Noisy 3D Data. 
  Proc. IROS-2013, pp. 2228-2233

• M. Günther, T. Wiemann, S. Albrecht, J. Hertzberg. 
  Model-Based Furniture Recognition for Building Semantic Object Maps 
  J. Artificial Intelligence, forthcoming
A semantic map for a mobile robot is a map that contains, in addition to spatial information about the environment, assignments of mapped features to entities of known classes. Further knowledge about these entities, independent of the map contents, is available for reasoning in some knowledge base with an associated reasoning engine.

A semantic (spatial) map exists only in relation to a KB!
There is More to Semantic Mapping than Labeling Objects in a Given Map!

- Reasoning in the domain theory allows hypotheses to be generated
- Hypotheses may need to be checked
- The area (space and the objects in it) get actively explored
- Exploration means “going there”, but possibly exploring (manipulating, inspecting, …) objects, too
Overview

1. Pose Planning in Autonomous (Semantic) Mapping
2. Some More on CAD Model Matching
3. Open Issues
Overview

1. Pose Planning in Autonomous (Semantic) Mapping

2. Some More on CAD Model Matching

3. Open Issues
3D SLAM with 6D Poses
What’s Missing for Autonomous SLAM?

• Online registration of 3D Scans
  – (some) literature, including (quite some) own

• Online pose correction according registration transformation
  – (some) literature, including (quite some) own

• Online loop detection
  – literature, including (some) own

• Online planning of next pose/path to optimize mapping
  – only very little literature
  – including (some) own and today’s paper
  – criteria:
    • fill up geometry map
    • verify object hypotheses
    • … and many more
All in Integration in a(nother) Castle

3D-Scannen in Schloss Birlinghoven
3D-SPLAM (1/5)

Planning

Raw Material
3D-Scan/s
Expl.: corridor scene,

Extract 2D Slice
Reduction to 2D
Expl.: all points with $y=150\pm2\text{cm}$
3D-SPLAM (2/5)

Sort and complete lines

Polar angles $\alpha_i$ of the line ends induce unique order. Connect neighboring scan lines by added artificial ones → Slice Polygon

Points to lines

Slice Polygon
3D-SPLAM (3/5)

Draw scan position candidates

Uniformly distributes Random Sampling in slice polygon, fixed number of test positions

The slice polygon ...

... borders the area mapped until now

... touches un-mapped area with its artificial lines

... is not necessarily free of “gaps” and “holes”
3D-SPLAM (4/5)

Rate scan pose candidates $x$

- $IG(x)$ (*information gain*): # virtual laser beams that cut across any artificial lines (the more, the better!)

- $||x_{\text{Start}} - x||$: Distance to $x$ from current robot position $x_{\text{Start}}$ (the smaller, the better!)

- $||\theta_{\text{Start}} - \theta(x)||$: Angular difference between the current robot orientation $\theta_{\text{Start}}$ and the orientation $\theta(x)$ of pose $x$ (the smaller, the better!)

Optimal scan pose

\[ x_{\text{Ziel}} = \arg\max_x \left[ w_1 IG(x) + w_2 ||x_{\text{Start}} - x|| + w_3 ||\theta_{\text{Start}} - \theta(x)|| \right] \]
3D-SPLAM (5/5)

Plan trajectory
- Either in closed-form continuous solution (elegant, somewhat brittle)
- Or with „discrete“ approach:
  - turn to goal point;
  - drive there straight;
  - turn into goal orientation
- Both maybe with intermediate targets
- Then check in 3D model, whether trajectory free!
  - If not, take next pose

Physical ride with obstacle avoidance, of course!

Volumes occupied space
Art Gallery Problem

Where put \( N \) guards, so that they can see all points of the inside area of a polygon (without holes)?

**Theorem**
For Polygon of \( P \) vertices ex.
solution for \( N=[P/3] \)

Is it simple to find good solutions?

**Theorem**
The Art Gallery Problem is NP-hard

But we want only 1 robot!
Cousins from Computational Geometry II

**Watchman Problem** Find a (minimal) path for one watchman, that allows him to oversee the inside area of the polygon completely!

**Theorem**
The Watchman Problem is NP-hard
(because the Art Gallery Problem is)

But we watch in static poses only!
...and we do not have the map!
The Problem of Optimal Exploration

What is, dependent on start information and real environment geometry, a drivable (kinematic, collision), expectedly shortest path between scan poses, at the end of which the polygon’s inside area is completely mapped?

Solution currently unknown!
More Reasons for a Robot for Particular Target Points

• Self organization
  • e.g., “Go to charging station!”

• Pose disambiguation
  • e.g., if entropy in probabilistic localization too high (“Go to landmark!”)
  • or planned right away as intermediate targets to avoid losing the pose estimation in the first place ("coastal navigation")

• “Transit poses”
  • e.g. door passing: Pass through pose on the door normal to avoid crossing through the door in an angle

• Poses to manipulate individual objects
  • e.g. clear table: Drive to pose allowing to reach as many clearable items as possible
Overview

1. Pose Planning in Autonomous (Semantic) Mapping
2. Some More on CAD Model Matching
3. Open Issues
Reminder: Architecture Context

Other object recognition methods may/should co-exist!

Model-based object recognition

Semantic Map

image-feature-based object recognition
shape-feature-based object recognition
hypothesis generation
hypothesis verification
geometric primitive detection

Physical Sensors:
3D laser scanners / cameras / RGB-D sensors / TOF cameras
Practical Issues in 3D Semantic Mapping

• Practically, Semantic Mapping is based on single scans/frames, rather than fully registered scene models
• In particular RGB-D cameras have small opening angle: only partial object views per frame, blurred by sensor noise
• Surfaces in real-world scenes are frequently cluttered
• Shiny and transparent objects exist
• A great many object models are available for matching

Care about robustness
- against occlusion
- of CAD matching
Robustness Against Occlusion

• When does object detection break in face of clutter?
• When does plane detection break?
• Table experiment
Point Cloud, Mesh and Segmentation

Figure 1: Segmentation results for a table top setup. First column: the captured point clouds; second column: initially created triangle mesh; third column: segmentation results. Triangles that were not classified as belonging to a planar patch are rendered in green. In each step more objects were added. In the last line a shadow disrupted the outer contour and the segmentation broke the table top plane into two clusters.

Figure reproduced from [5] of furniture. Also, the physical object might differ from the CAD model for other reasons like damage or other modifications.

To investigate how robust our ICP matching step is with respect to such differences between CAD model and actual object, we conducted an experiment where we matched several CAD models of chairs against recorded point data of a chair. A photograph of the chair and a view of the resulting point cloud can be seen in Figure 2. The point cloud displayed in Figure 2b was created from five registered Kinect frames.

We matched this data against six different CAD models of chairs that were retrieved from Google’s Warehouse (Figure 5). We considered the Chair 1 model to be most similar to the actual chair, while Chair 5 were expected to be similar enough to produce meaningful matching results. For comparison, we included a model of a stool (Chair 5) and a wing chair (Chair 6), which we expected to be too different from the chair in the sensor data.
4. Results

We performed three experiments to evaluate the robustness and accuracy of our recognition system. First, we investigated the effectiveness of our hole filling procedure separately to see whether it is capable of estimating the true surface area of a table under the influence of increasing amounts of clutter. Second, we tested the robustness of our system with respect to the required similarity between CAD model and actual object. Lastly, we evaluated the detection accuracy of our complete system on two series of point clouds captured by a mobile robot.

4.1. Robustness Against Occlusion

In real life applications, furniture is usually used to store objects, so an obvious problem for our detection procedure is that the surfaces relevant for recognition may be partially occluded. To evaluate the robustness of the surface extraction procedure against occlusions, we gradually added typical objects like books, cups, and bottles to a table surface and tried to segment the table top. The results of this experiment are shown in Fig. 3 and Table 1. The estimated area remains close to the ground truth area even for significant amounts of clutter; only when the outer contour is disrupted, the area estimation breaks down.

<table>
<thead>
<tr>
<th></th>
<th>0 obj.</th>
<th>2 obj.</th>
<th>3 obj.</th>
<th>4 obj.</th>
<th>5 obj.</th>
<th>6 obj.</th>
<th>7 obj.</th>
<th>12 obj.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region Growing</td>
<td>1.50</td>
<td>1.47</td>
<td>1.47</td>
<td>1.40</td>
<td>1.35</td>
<td>1.26</td>
<td>1.17</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>93%</td>
<td>92%</td>
<td>92%</td>
<td>87%</td>
<td>84%</td>
<td>79%</td>
<td>73%</td>
<td>59%</td>
</tr>
<tr>
<td>Contour Triangulation</td>
<td>1.50</td>
<td>1.50</td>
<td>1.49</td>
<td>1.52</td>
<td>1.52</td>
<td>1.52</td>
<td>1.50</td>
<td>1.20</td>
</tr>
<tr>
<td></td>
<td>93%</td>
<td>93%</td>
<td>93%</td>
<td>95%</td>
<td>95%</td>
<td>95%</td>
<td>93%</td>
<td>75%</td>
</tr>
</tbody>
</table>

- Table top area in m² and as percentage of ground truth
- Region growing starts breaking for moderate clutter
- Contour triangulation stabilizes matters, unless the contour is occluded, too
- Need help, e.g., of texture
Robustness of CAD Object Matching

Chairs are different

CAD models of different chair types applied for matching against sensed chairs

Sensed chairs even more

Chair model registered from 3 Kinect frames
Best Matches depend on …

- … good pose guess and good type guess
- A best match does always exist!

Figure 7: Transparent overlay of final poses obtained by CAD matching with point clouds. Top row: results from an aligned pose estimation. Second from top: results from pose estimation below actual pose. Second from bottom: results for slight displacement and rotation error. Bottom row: results for larger displacement and rotation error. See Figs – for visualization of initial poses.

The resulting final errors for the six CAD models in this experiment are shown in Table ws. As expected, Chair v–y converged about equally well to the reference pose except for one outlier from Chair y. Whereas Chair z and –d i d not converge well. However, it has to be noticed that the data used in this experiment was not very challenging, since the chair is completely captured in the point cloud data and there are no other objects near the chairs. To evaluate the effect of choosing a different CAD model on the performance of the complete system, we ran the complete pipeline as explained in the next subsection once for each chair model. Table x compares the final translation and rotation errors after ICP alignment on the seminar room dataset. The results clearly indicate that as long as the CAD model is “similar enough” to the objects found in the data, our system works. The results for chairs w–y were even slightly better than for chair v, which we considered.
Some Details
# Quantitative Results

<table>
<thead>
<tr>
<th>pose 1</th>
<th>final pose error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$e_{\text{translation}}$</td>
</tr>
<tr>
<td>chair 1</td>
<td>0.5 cm</td>
</tr>
<tr>
<td>chair 2</td>
<td>0.0 cm</td>
</tr>
<tr>
<td>chair 3</td>
<td>0.0 cm</td>
</tr>
<tr>
<td>chair 4</td>
<td>0.1 cm</td>
</tr>
<tr>
<td>chair 5 $\dagger$</td>
<td>3.1 cm</td>
</tr>
<tr>
<td>chair 6</td>
<td>29.5 cm</td>
</tr>
</tbody>
</table>

- Insignificant differences among “plausible” chair models
- Stool and wingchair stick out

$i$ Stool is largely rotation symmetric, don’t regard rotation error!
Overview

1. Pose Planning in Autonomous (Semantic) Mapping
2. Some More on CAD Model Matching
3. Open Issues
Open Issues/Work in Progress

• Care about transparent and shiny objects
• Find criteria for “good enough” match
• Really do multi-modal semantic mapping
• Really do active semantic mapping (to resolve ambiguity, move sensors & manipulate environment)
• Use GIS technology for storing semantic maps (space-related part) compactly and help optimize (some) queries
  – “Give me the list of green tables with at least 1 muffin on”
Thank you for your time!