Humanoid Robotics

Exploiting Obstacle Type Information During Navigation

Maren Bennewitz
Undesired Robot Behavior

Oh, I’m stuck!
Navigation Through Cluttered Regions

- Requires complex actions, potentially including also the manipulation of objects

- Challenge: Finding a complete, detailed plan of all necessary actions and whole-body motions is computationally expensive
Key Idea to Speed-Up Planning

- Simplify planning by splitting the whole plan into several parts
- Exploit knowledge about different obstacle classes to choose appropriate robot actions
Our Approach

- Combine 2D path planning, 3D foot step planning, and object manipulation
- Segments objects using a convolutional neural network (CNN)
- For each object class, estimate costs of predefined actions to overcome the obstacle
- Encode the action costs of detected objects in a 2D grid map
- Use A* on this map for fast path planning
Obtaining Class Information

Train a CNN from Bonnet that transforms images in semantic information online

[Milioto and Stachniss, ICRA18]
Training Data: Images of Clutter With Added Background
# Actions for Obstacle Classes

<table>
<thead>
<tr>
<th>object class</th>
<th>action type</th>
</tr>
</thead>
<tbody>
<tr>
<td>balls</td>
<td>push, step over, pick up</td>
</tr>
<tr>
<td>cars, toy blocks</td>
<td>step over, pick up</td>
</tr>
<tr>
<td>stuffed toys, dolls</td>
<td>pick up</td>
</tr>
<tr>
<td>boxes, books</td>
<td>step onto</td>
</tr>
</tbody>
</table>
Actions for Obstacle Classes

- Use depth data to exclude actions depending on the size of objects
- Obstacles with unknown class or no executable action are considered as static
- Assign the “cheapest” possible action to each detected object
Action Costs

- Define the costs of actions according to the completion time
- Average execution time for each action from experimental runs
Path Planning

- Use the segmented objects and depth information to project the objects with their action costs onto a 2D grid
- Apply A* search on the 2D cost grid to find a path to the robot’s goal location
- The obtained 2D path contains the necessary actions
Segmentation of Camera Image

robot camera view
Mapping
From Mapping to Planning
Plan Execution

- Path segments **without objects**: walking control along the 2D path
- **Push/pickup**: walk to the last free grid cell on 2D path and execute the corresponding action using the segmented point cloud
- **Step over/onto**: 3D footstep planning in the corresponding region on a height map computed from the point cloud
Path Planning and Replanning

The Nao robot needs to reach the bottom part of the map.

For that it will need to navigate through the toy blocks or the stuffed animal.
Initially, our planner computes a path that suggests to perform the *step over* actions over the two blocks to be computed with a footstep planning algorithm.
Summary

- Exploit knowledge about obstacle classes during path planning
- Train a CNN to distinguish different obstacle classes
- Construct a 2D grid that encodes the associated action costs derived from completion time
- Compute the robot’s path, which implicitly contains all necessary actions to handle objects
Literature

- Classifying Obstacles and Exploiting Class Information for Humanoid Navigation through Cluttered Environments
  P. Regier, A. Milioto, C. Stachniss, and M. Bennewitz
  International Journal of Humanoid Robotics (IJHR), 2020