Efficient Humanoid Navigation through Cluttered 3D Environments

Maren Bennewitz

Joint work with P. Karkowski, S. Oßwald, and P. Regier
Robots in Human Environments

Need to navigate through challenging passages with

- Objects blocking the path
- Highly cluttered regions
- Different levels
- Dynamic obstacles
Requirements

- Fast sensor data interpretation
- Real-time footstep planning
- Reactive balance and dynamic walking control
Our Approach

- Fast traversability analysis from depth data
- Avoidance of local minima by finding complete 3D footstep plans to local goals
- Real-time planning and replanning in case of sudden changes
- Only low CPU usage
Related Work

- Footstep planning using **rapidly-exploring random trees** (RRTs), e.g., Baudouin et al.
- Mixed integer optimization on convex regions, e.g., Deits et al.
- **A* footstep planning** using fixed footstep sets, e.g., Hornung et al., Chestnutt et al.
Footstep Planning with A*

- Uses a set of footstep actions to reduce the computational demand
- Standard approach: fixed set of actions
Footstep Planning with A*
Footstep Planning with A*
Footstep Planning with A*
Footstep Planning with A*

- Transition costs
- Path costs from start to $s$
- Estimated costs from $s'$ to goal
- Transition costs $c(s,s')$
- Path costs from start to $s$

Diagram:
- Start
- $S$
- $S'$
- Goal
- $h(s')$
- $g(s)$
Footstep Planning with A*

Small set $\rightarrow$ fast planning
limited search space

Large set $\rightarrow$ large coverage
long planning time
Adaptive Node Expansion

Our approach:

- Add only a small set of nodes at each expansion step
- Systematically search for valid successors
- Apply fast validity checks using height information

- Leads to a high success rate, short paths, and fast planning times
Cost Functions

**Cost function** $g$:
\[
c = \|x' - x\| + \nu \cdot |\theta| + \mu \cdot |e' - e|
\]

- **Step distance**
- **Relative rotation**
- **Height difference**

**Heuristic cost** $h$:
Cost of the direct path to the goal using maximum step length
Field of View
Planar Region Segmentation
Finding Edges
World Representation

Karkowski et al., ICRA 2016
Reachability Map for Footsteps

- Discretization of feasible footsteps
- Reachability map can be precomputed using inverse kinematics
- Maximum displacement from the previous step, depending on the displacement direction
- Maximum displacement along the upward and downward directions
A* with an Adaptive Action Set

- Local search around maximum forward step
- Validity checks:
  - Footstep feasible according to the reachability map?
  - Footstep on a planar region?
  - Later: No collision of the robot’s swept volume with obstacles?
- Result: set of viable successor states that adapt to the local environment
Node Expansion
Node Expansion
Node Expansion
Node Expansion
Experiments

- Planning area of 2.4m x 2.4m, randomly generated obstacles
- Resolution of 1.5cm for the height map
- Local goal located at the opposite side on the map
- Comparison to A* with fixed sets of 10 and 20 footsteps
- Computations performed on single Intel Core i7 3770 CPU
Example Map

<table>
<thead>
<tr>
<th>A* (small)</th>
<th>A* (large)</th>
<th>Our approach</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Diagram" /></td>
<td><img src="image2.png" alt="Diagram" /></td>
<td><img src="image3.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>
### Example Map

#### Planning times

<table>
<thead>
<tr>
<th></th>
<th>A* (small)</th>
<th>A* (large)</th>
<th>Our approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning times</td>
<td>32.0 ms</td>
<td>32.2 ms</td>
<td>19.7 ms</td>
</tr>
</tbody>
</table>

#### Total path costs

<table>
<thead>
<tr>
<th></th>
<th>A* (small)</th>
<th>A* (large)</th>
<th>Our approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total path costs</td>
<td>9.9</td>
<td>5.0</td>
<td>4.3</td>
</tr>
</tbody>
</table>
Further Examples

<table>
<thead>
<tr>
<th>A* (small)</th>
<th>A* (large)</th>
<th>Our approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>map 1</td>
<td>map 2</td>
<td></td>
</tr>
</tbody>
</table>

- Start
- Goal
## Experimental Results

<table>
<thead>
<tr>
<th>Map 1</th>
<th>A* (small)</th>
<th>A* (large)</th>
<th>Our approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning Time</td>
<td>85.7 ms</td>
<td>60.3 ms</td>
<td><strong>14.3 ms</strong></td>
</tr>
<tr>
<td>Path Cost</td>
<td>7.6</td>
<td>4.0</td>
<td><strong>3.7</strong></td>
</tr>
<tr>
<td>Expanded Nodes</td>
<td>28260</td>
<td>13835</td>
<td><strong>1516</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Map 2</th>
<th>A* (small)</th>
<th>A* (large)</th>
<th>Our approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning Time</td>
<td>67.2 ms</td>
<td>75.0 ms</td>
<td><strong>48.7 ms</strong></td>
</tr>
<tr>
<td>Path Cost</td>
<td>6.9</td>
<td>5.0</td>
<td><strong>4.3</strong></td>
</tr>
<tr>
<td>Expanded Nodes</td>
<td>22374</td>
<td>16805</td>
<td><strong>5852</strong></td>
</tr>
</tbody>
</table>
Real-Time Footstep Planning

Karkowski et al., Humanoids 2016
Real-Time Footstep Planning 2D
Navigation through Cluttered Regions

Typically, objects appear in clusters:
- Children’s rooms with toys on the floor
- Workshops with tools lying around
- Storage rooms with piles of boxes

Challenges
- Accurate sensing of obstacles
- Precise motion execution
- Obstructed view of the sensor
Navigation through Cluttered Regions

Leads to
- Decreased velocity
- Frequent turns
- Potentially getting stuck
Our Approach

- Choose efficient paths by avoiding regions predicted to be too cluttered
- **Predict the occurrence of objects beyond observed areas**
- Estimate navigation costs corresponding to potential obstacles to navigate foresightedly
- Achieve shorter completion time of navigation tasks
Occupation Density and Costs

- Estimate **occupation densities for 2D grid cells** based on observed objects
- **Increase the traversal costs** for cells in close-by regions that are not yet visible
- Plan the robot’s **global 2D path on a cost grid map** that contains:
  - Standard inflation costs around obstacles
  - Predicted costs from estimated occupation densities
Estimated Occupation Density
Estimated Occupation Density
Estimated Occupation Density
Estimated Occupation Density

Estimate the density $\rho$ inside circle

$$\rho = \frac{\# \text{ occupied cells}}{\# \text{ cells}}$$
Estimated Occupation Density

\[
\rho = \sum_{j=1}^{3} w_j \rho_j, \text{ where } w_j = e^{-R_j}
\]

predicted navigation costs: \( C_{predict} = \alpha \cdot \rho \)
Cost Grid Map
So far: Tested on Wheeled Base

ASUS Xtion Pro Live for object detection

Robotino robot, Festo Didactics

laser scanner for localization
Foresighted Navigation

Regier et al., IROS 2016
Experimental Evaluation

- Extensive simulation experiments
- Randomly sampled objects within a rectangular area of size $23 \times 8 \text{ m}^2$
- Obstacle density as a parameter: average number of objects per $1 \text{ m}^2$
Simulation Result

w/o prediction

standard approach

with prediction

our approach
The behavior is different when the clutter
  - is sparse enough for the robot to fit through
  - but dense enough to impede the robot

Our approach achieves significantly shorter completion times for object densities values between 0.3 and 0.8
Ongoing Work

- Learn clutter distributions for individual environments
- Learn the cost function for the specific navigation capabilities of the robot
- Humanoid autonomously decides whether to move through only partially observable, cluttered region, or take a path around it
Conclusions

- Real-time map segmentation and footstep planning in 3D at low CPU cost
- Reduced planning time compared to A* with fixed footstep sets
- Lower path costs due to adaptive node expansion
- Prediction of obstacle occurrences and corresponding navigation costs
- Avoidance of regions predicted to be too cluttered leads to shorter completion time
Thank you!