Efficient Humanoid Navigation through Cluttered 3D Environments

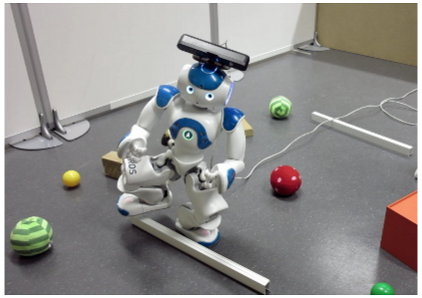
Maren Bennewitz

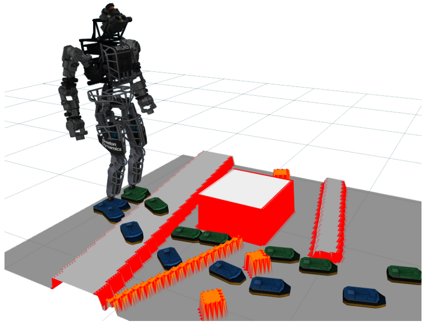
Joint work with P. Karkowski, S. Oßwald, and P. Regier

Robots in Human Environments

Need to navigate through challenging passages with

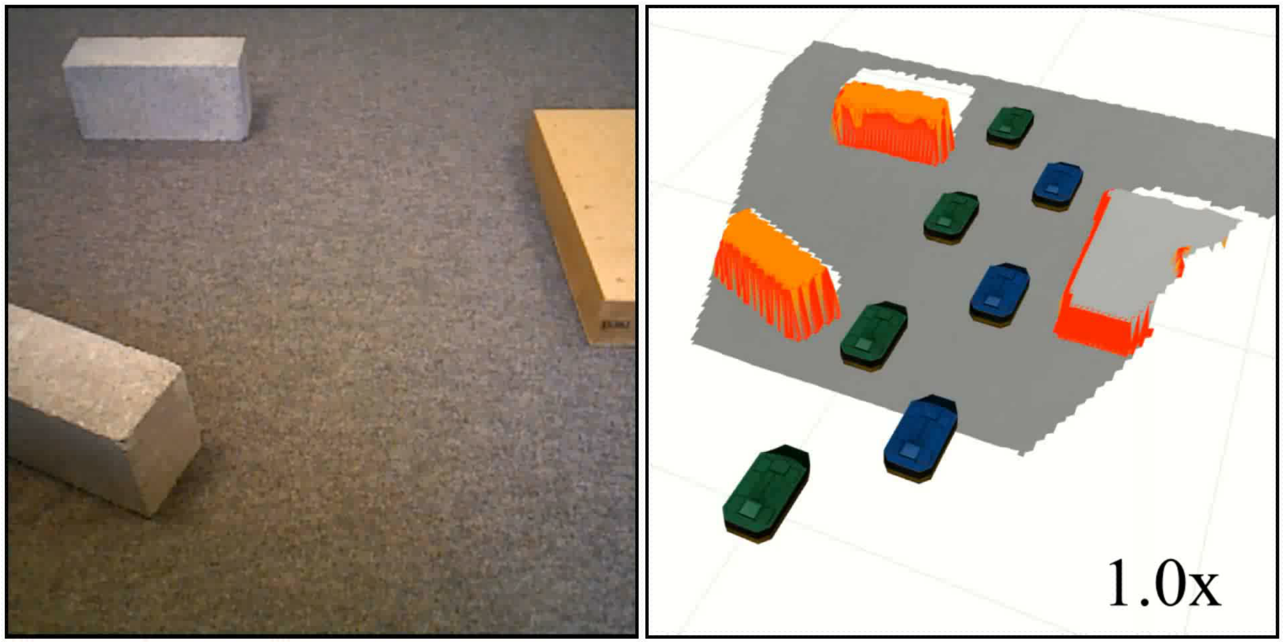
- Objects blocking the path
- Highly cluttered regions
- Different levels
- Dynamic obstacles





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

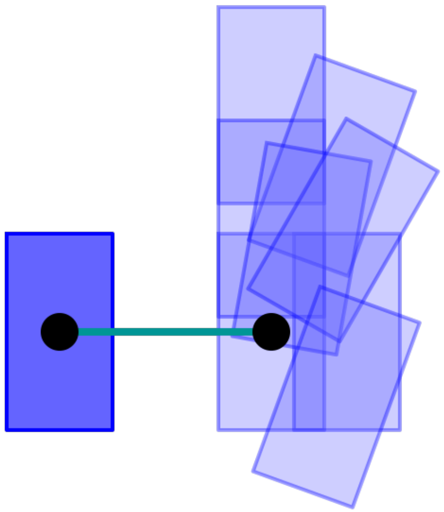
- Footstep planning using rapidly-exploring random trees (RRTs), e.g., Baudouin et al.

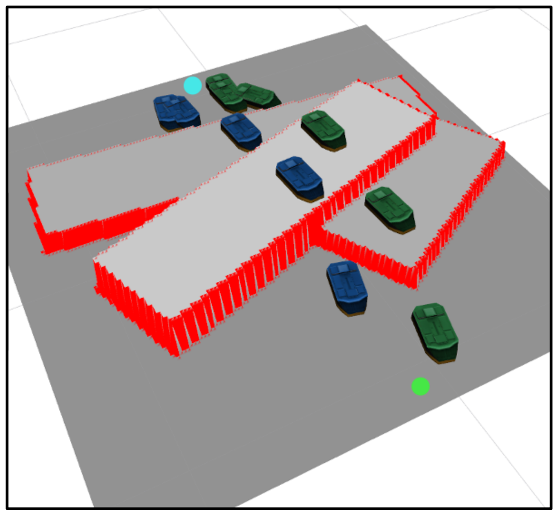
- Mixed integer optimization on convex regions, e.g., Deits et al.

- A* footstep planning using fixed footstep sets, e.g., Hornung et al., Chestnutt et al.

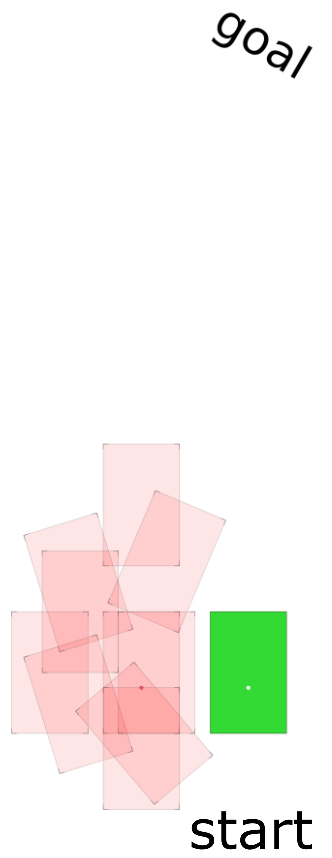
Footstep Planning with A*

- Uses a set of footstep actions to reduce the computational demand
- Standard approach: fixed set of actions

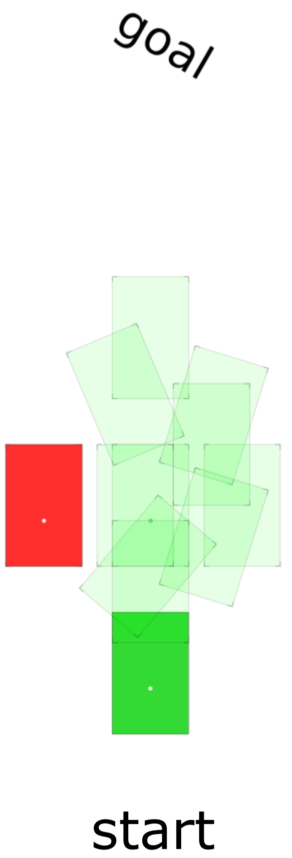




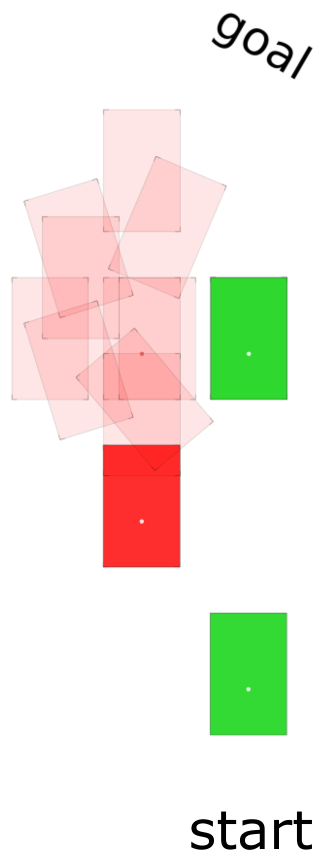
Footstep Planning with A*



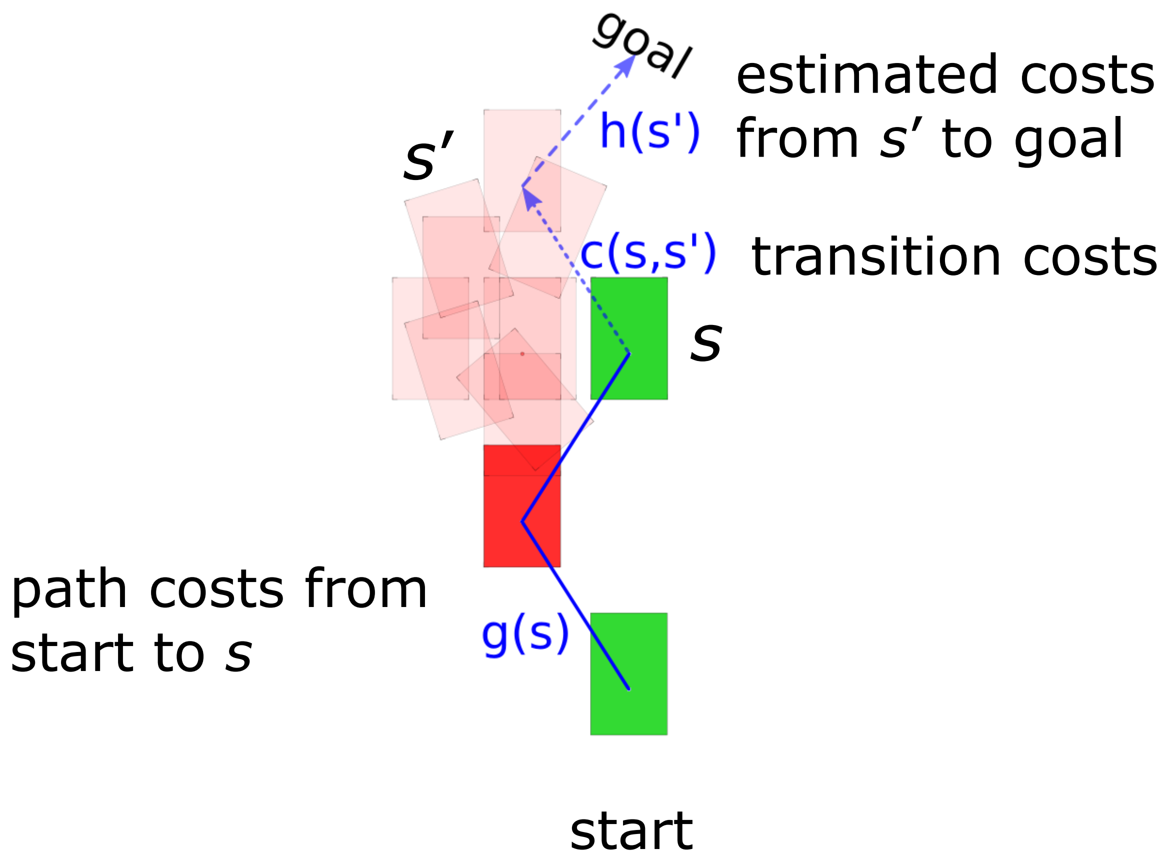
Footstep Planning with A*



Footstep Planning with A*

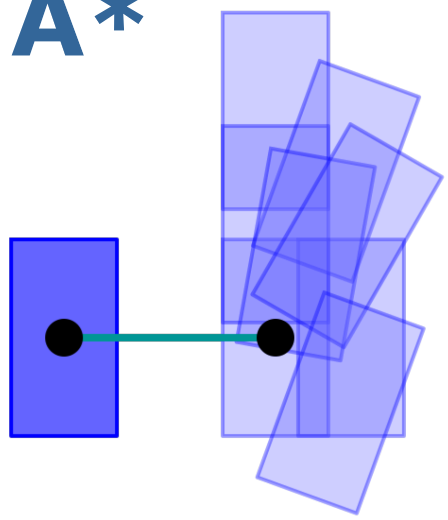


Footstep Planning with A*

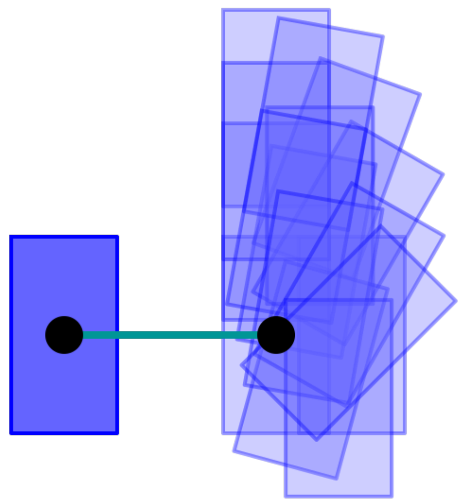


Footstep Planning with A*

Small set → fast planning
limited search space



Large set → 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

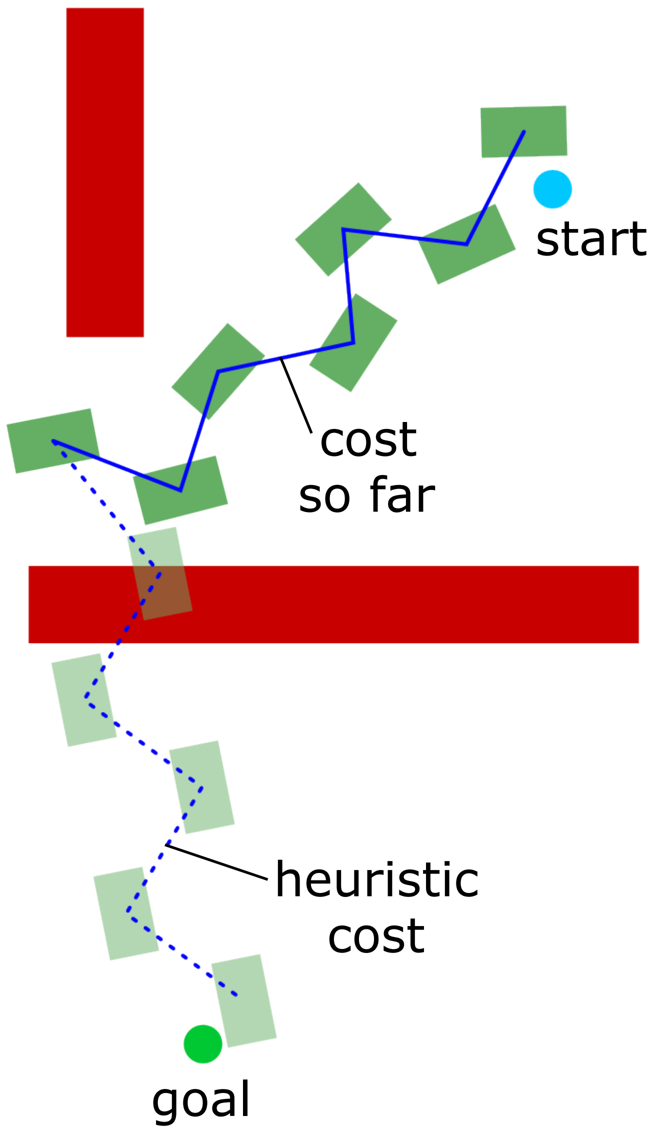
Cost function *g*:

$$c = ||\mathbf{x}' - \mathbf{x}|| + \nu \cdot |\theta| + \mu \cdot |e' - e|$$

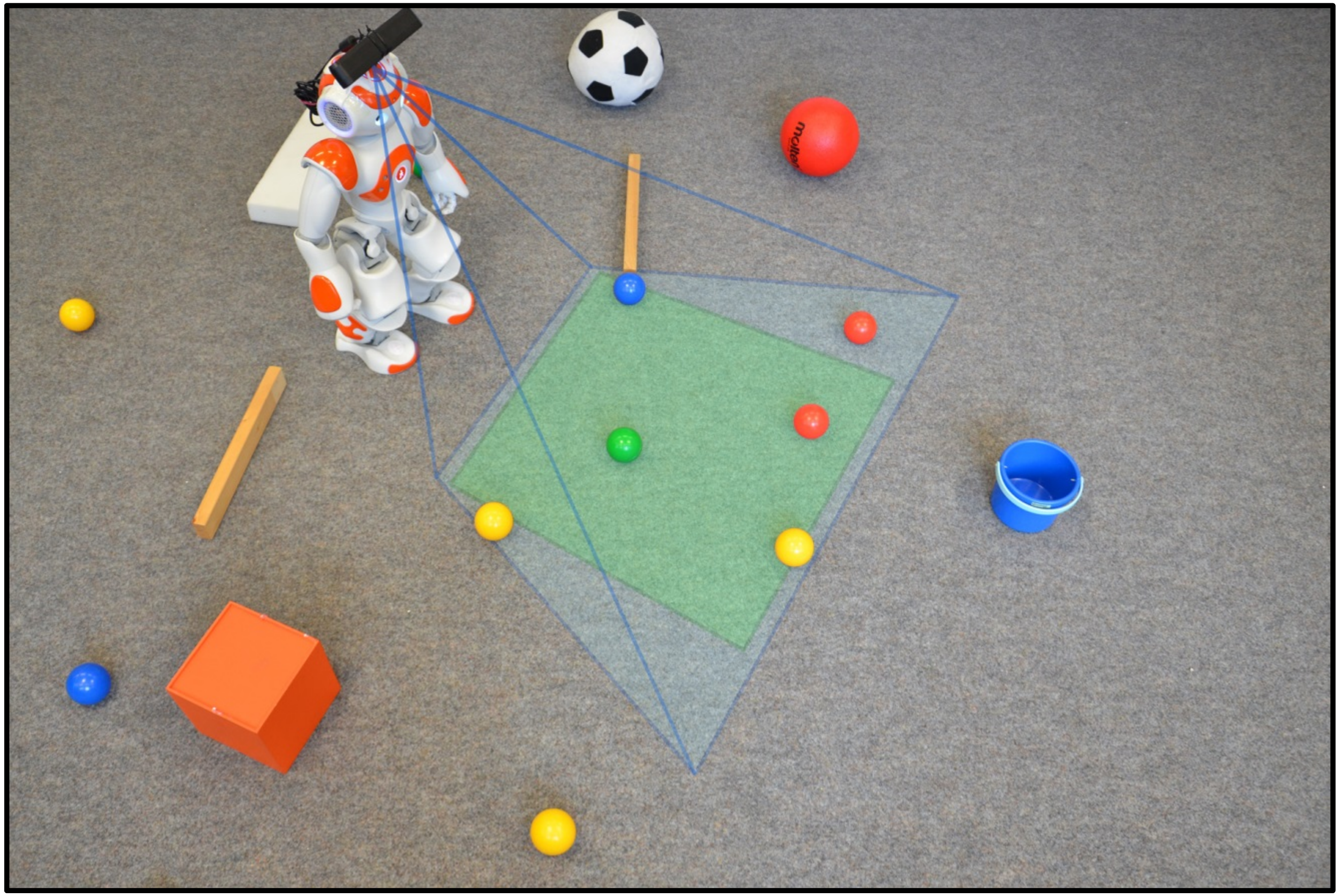
step distance — relative rotation — height difference

Heuristic cost *h*:

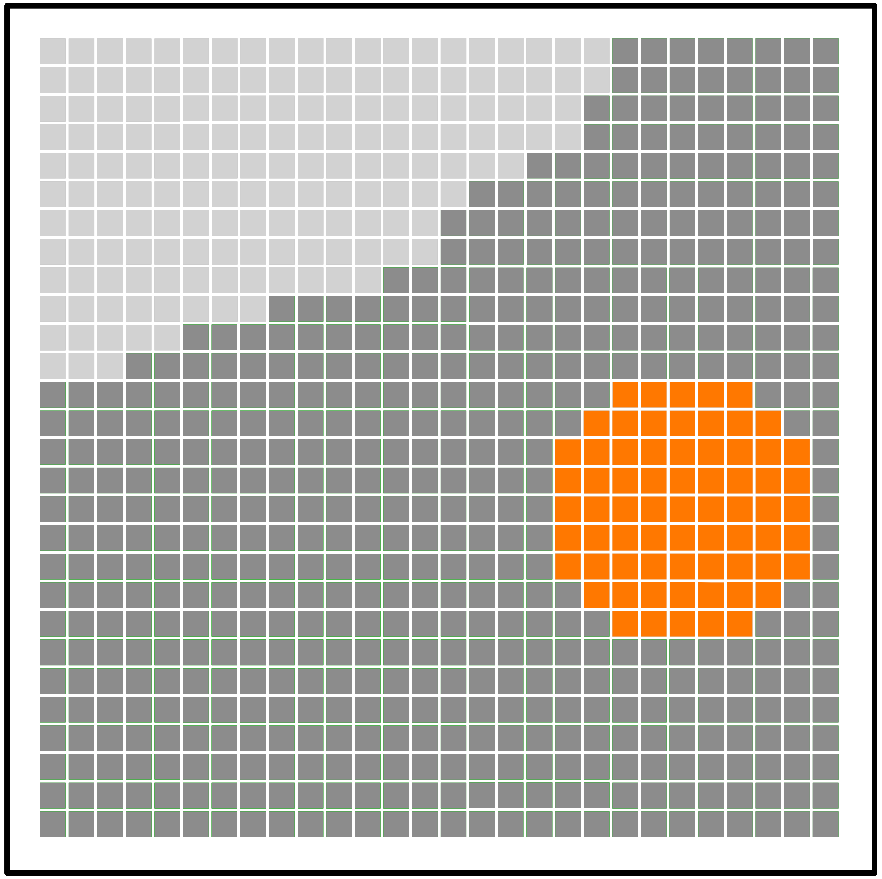
Cost of the direct path to the goal using maximum step length



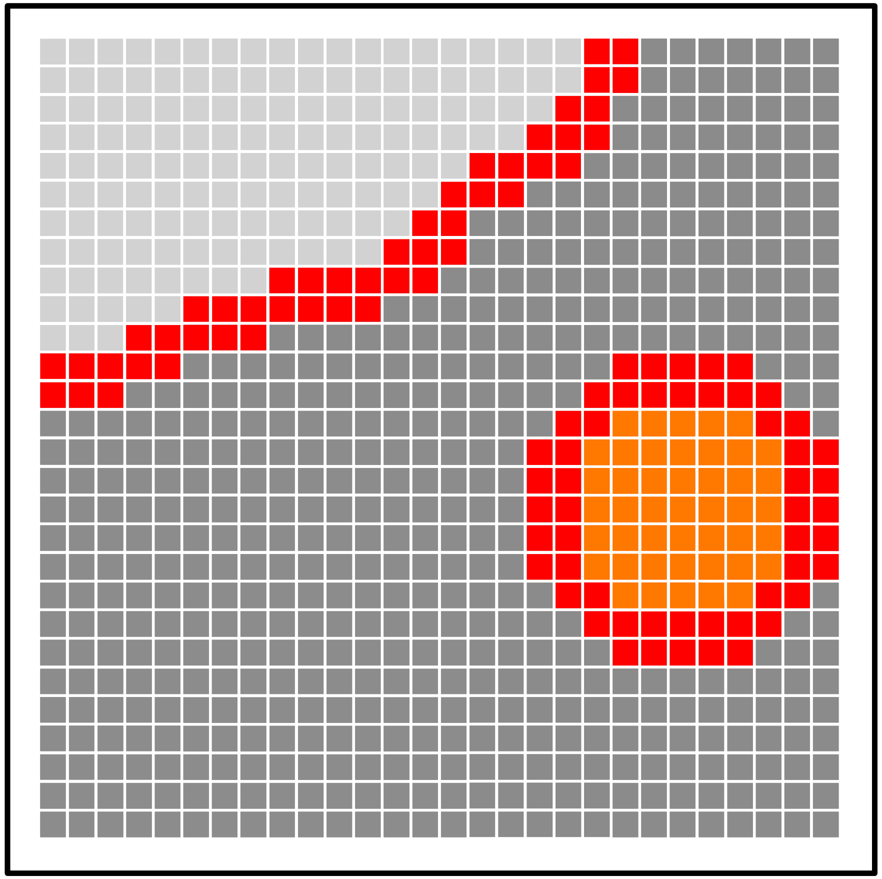
Field of View



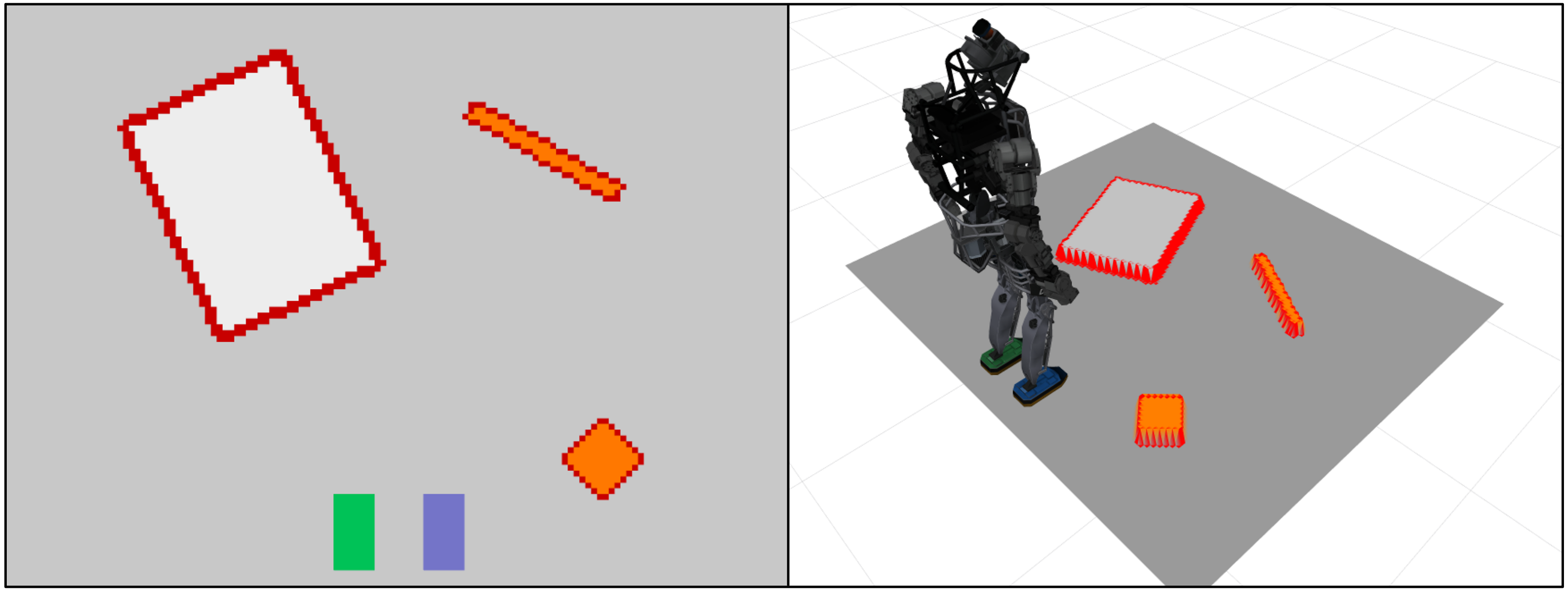
Planar Region Segmentation



Finding Edges



World Representation

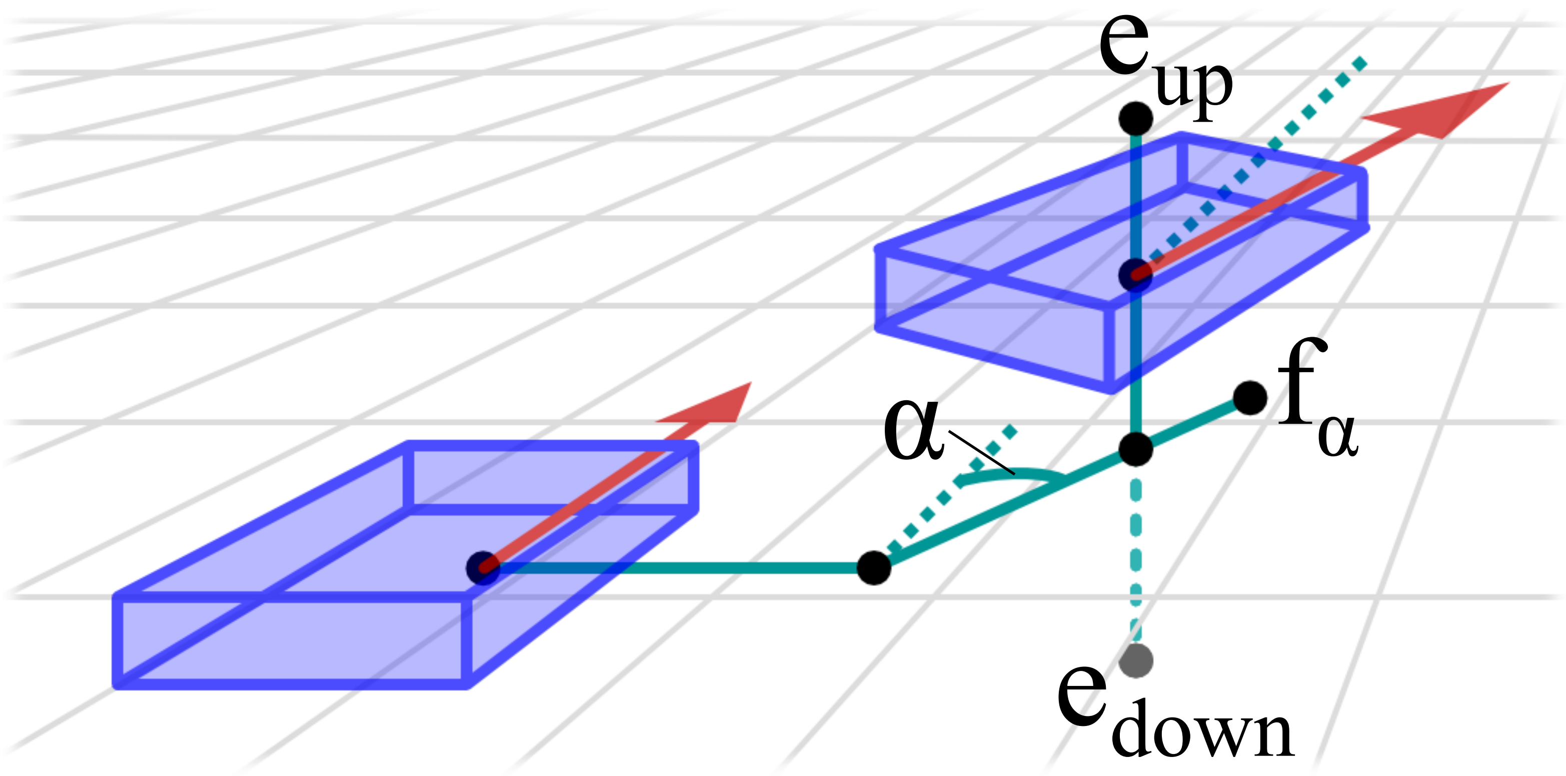


Karkowski et al., ICRA 2016

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

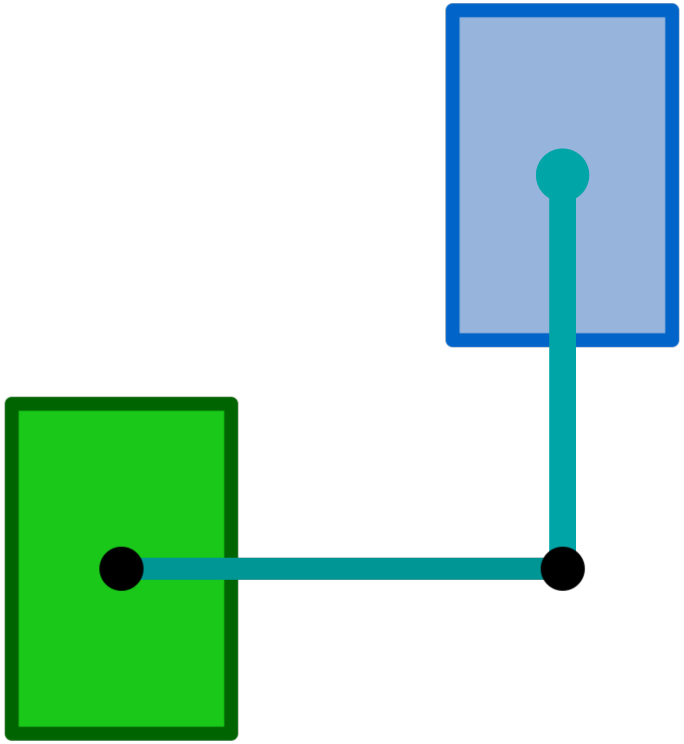
Reachability Map



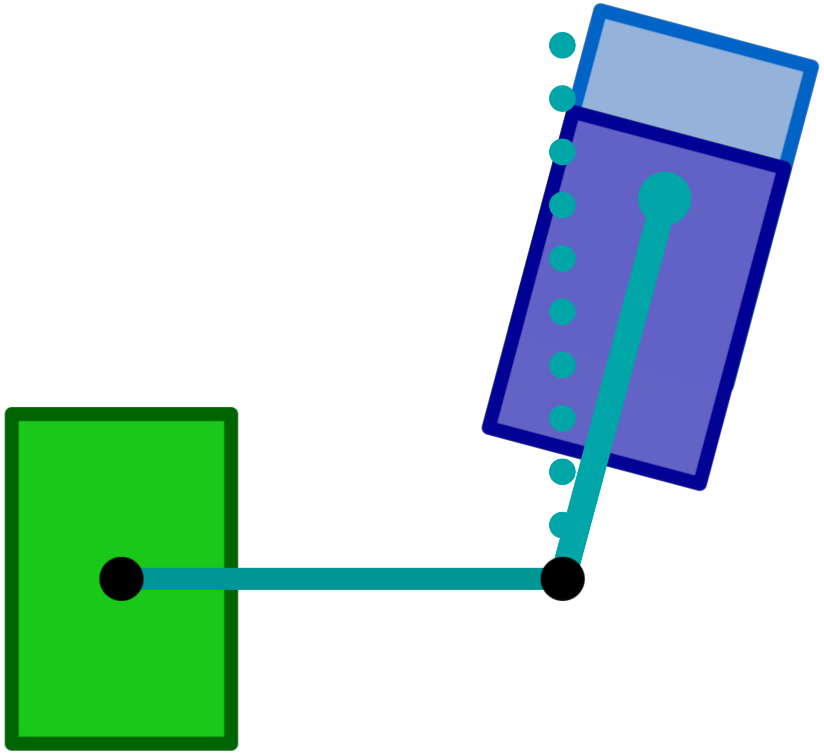
A* with an Adaptive Action Set

- 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

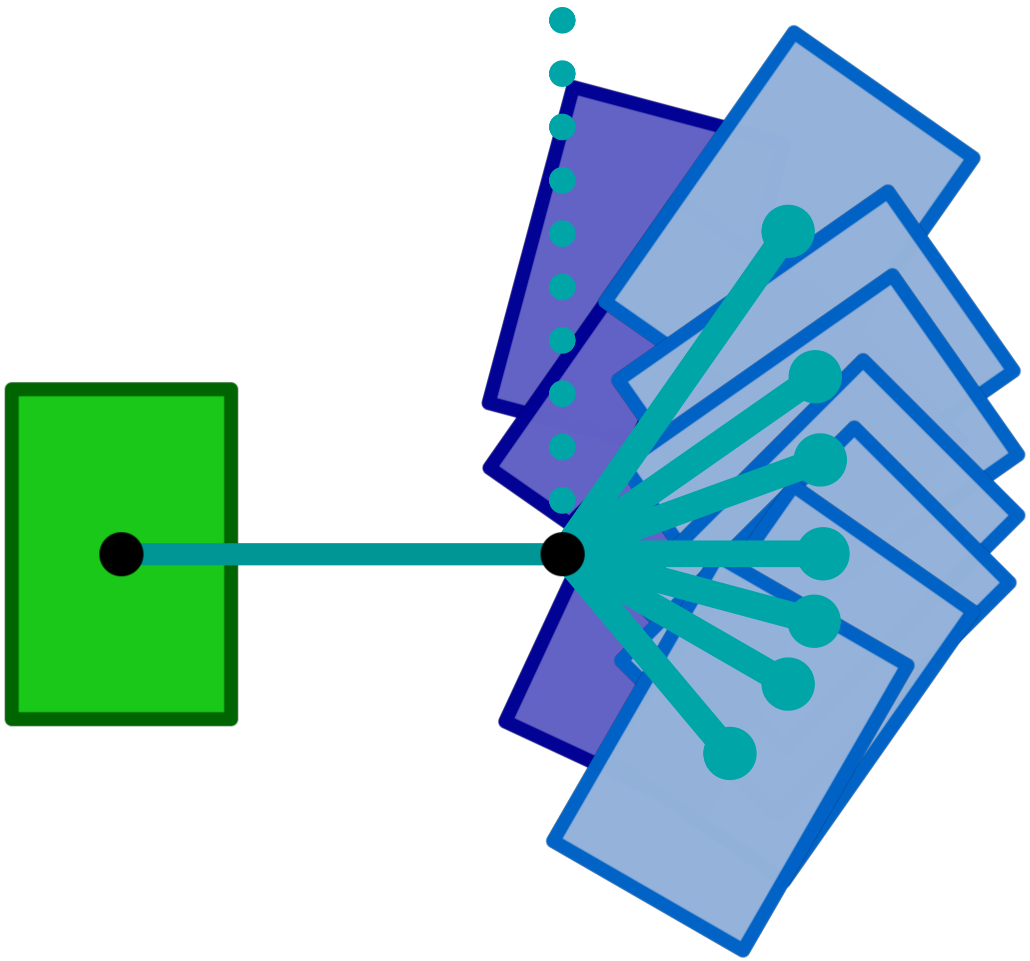
Node Expansion



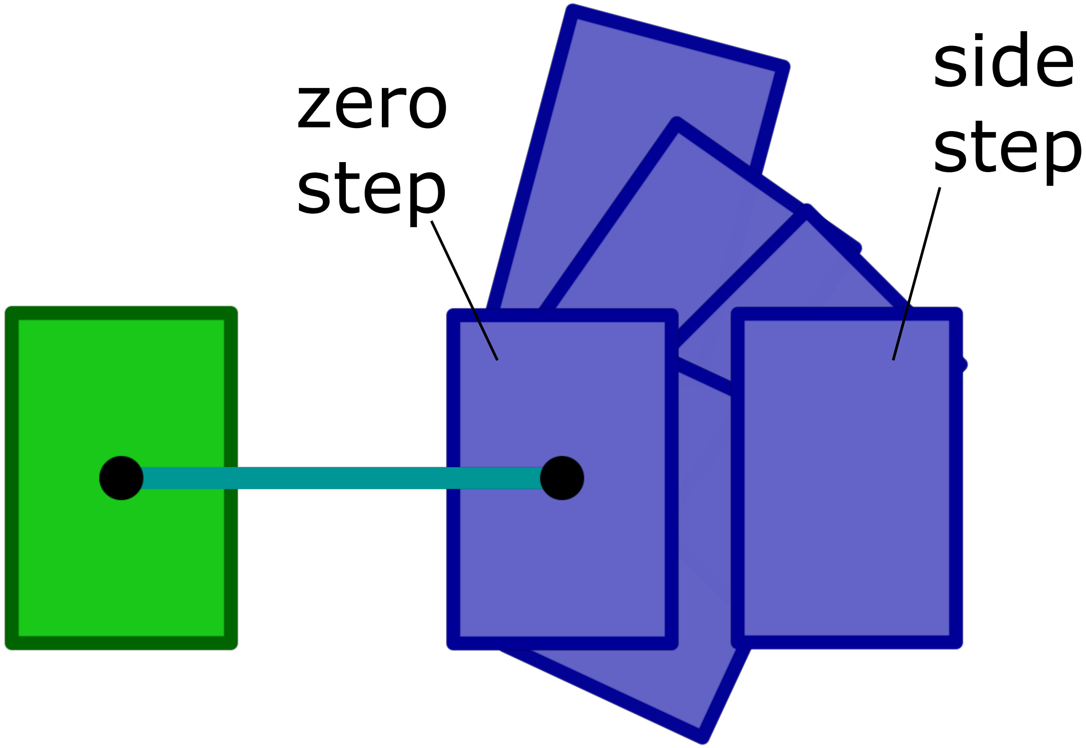
Node Expansion



Node Expansion

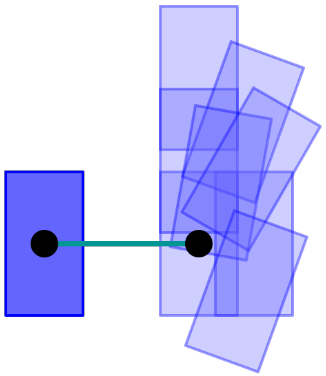


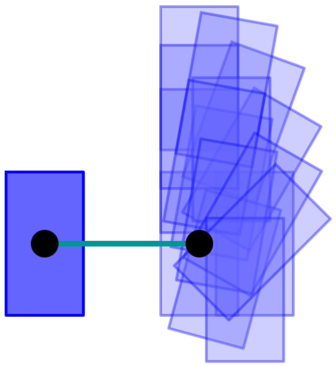
Node Expansion



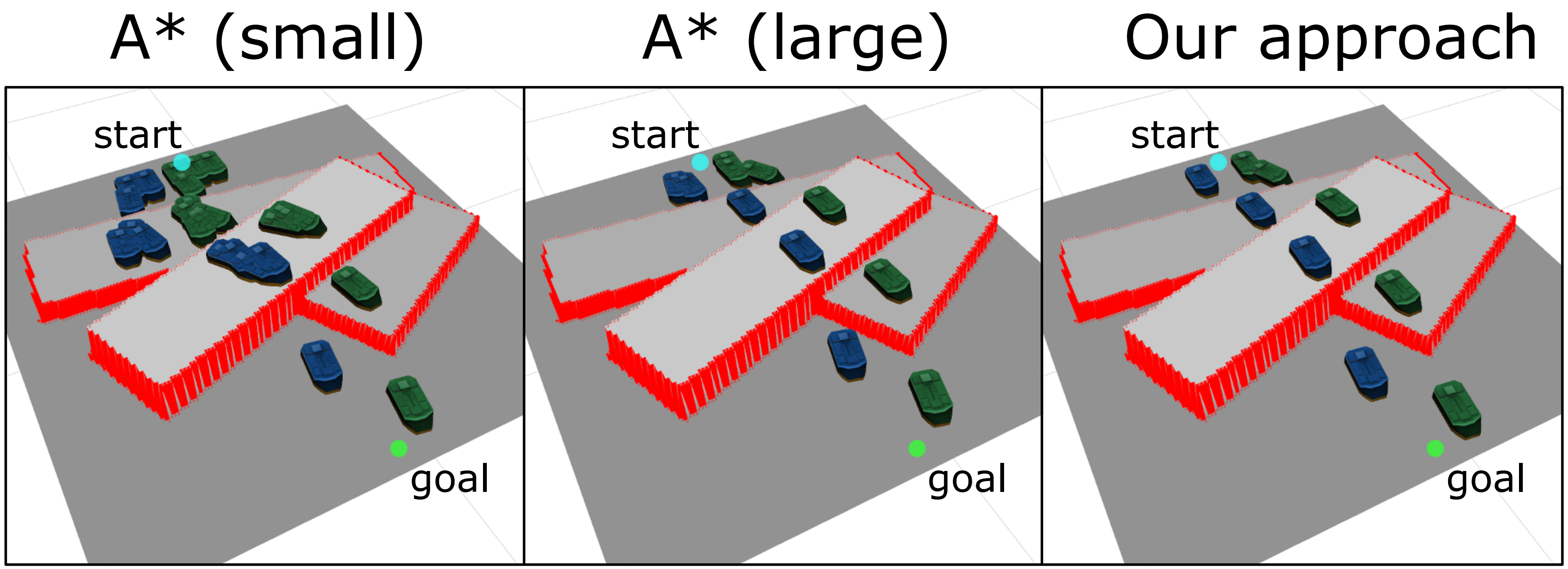
Experiments

- 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

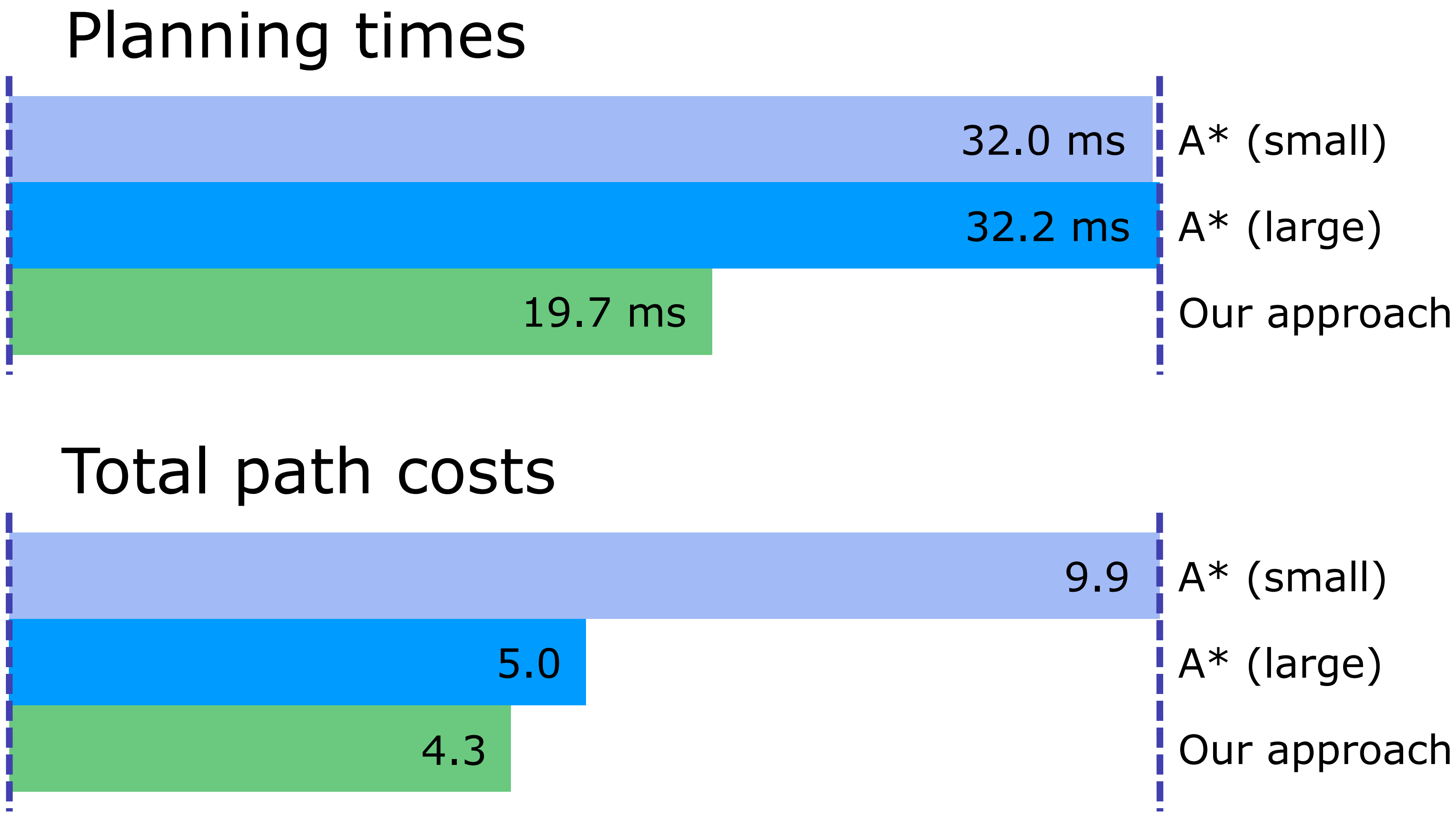




Example Map

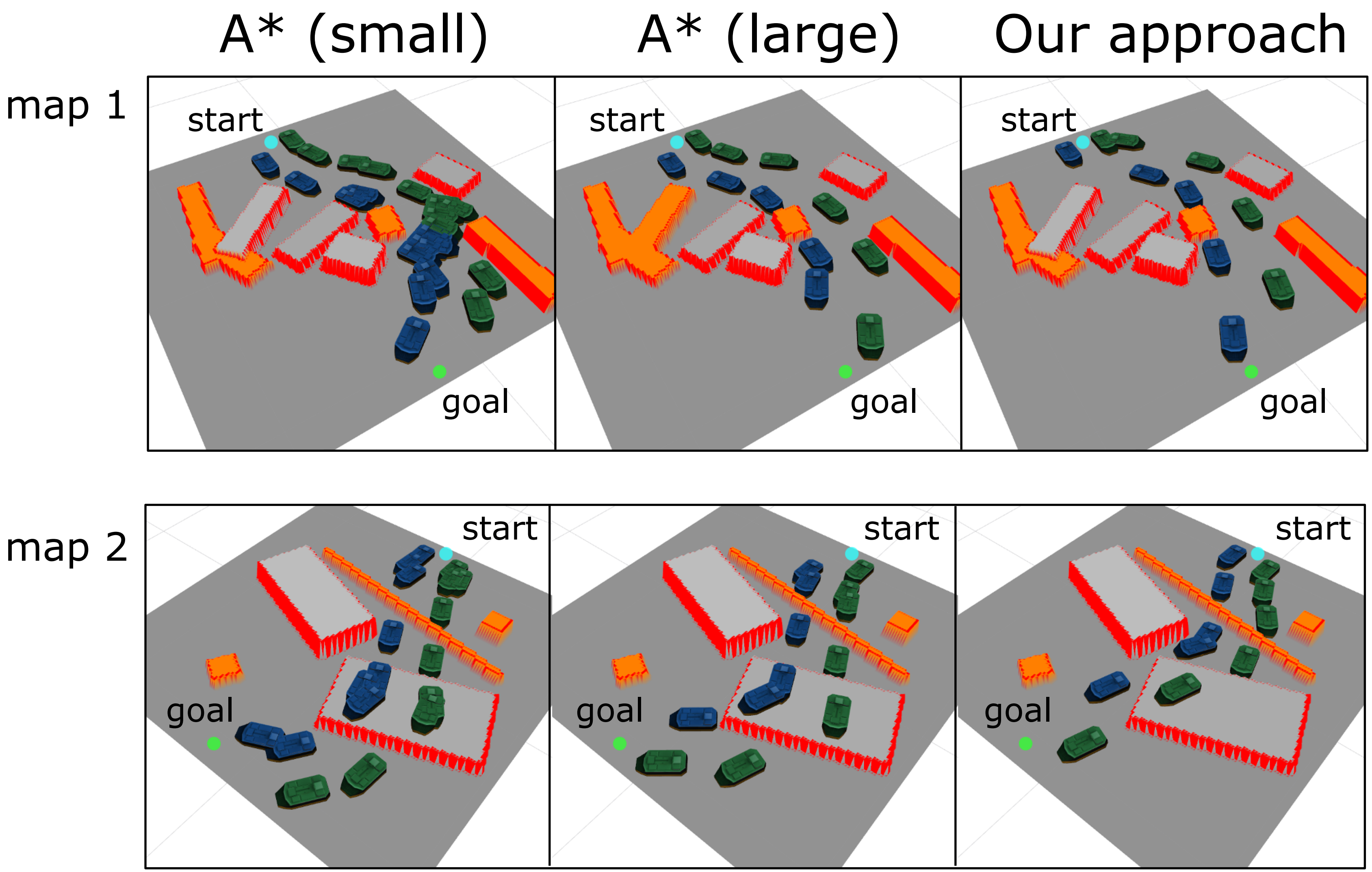

A* (small)
A* (large)
Our approach
start
goal
start
goal
start
goal

Example Map



Planning times
32.0 ms
A* (small)
32.2 ms
A* (large)
19.7 ms
Our approach
Total path costs
9.9
A* (small)
5.0
A* (large)
4.3
Our approach

Further Examples

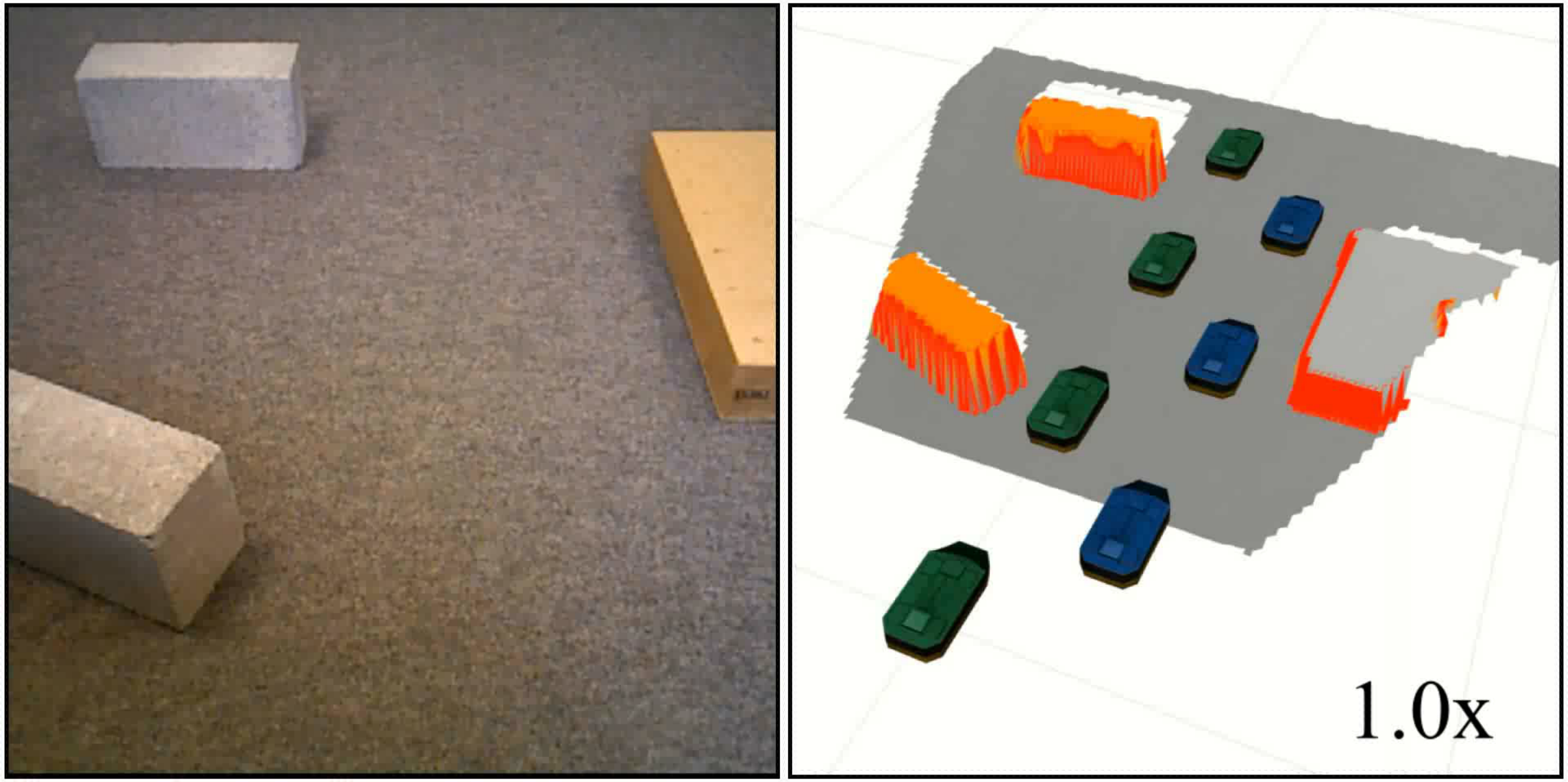


Experimental Results

Map 1	A* (small)	A* (large)	**Our approach**
Planning Time	85.7 ms	60.3 ms	**14.3 ms**
Path Cost	7.6	4.0	**3.7**
Expanded Nodes	28260	13835	1516

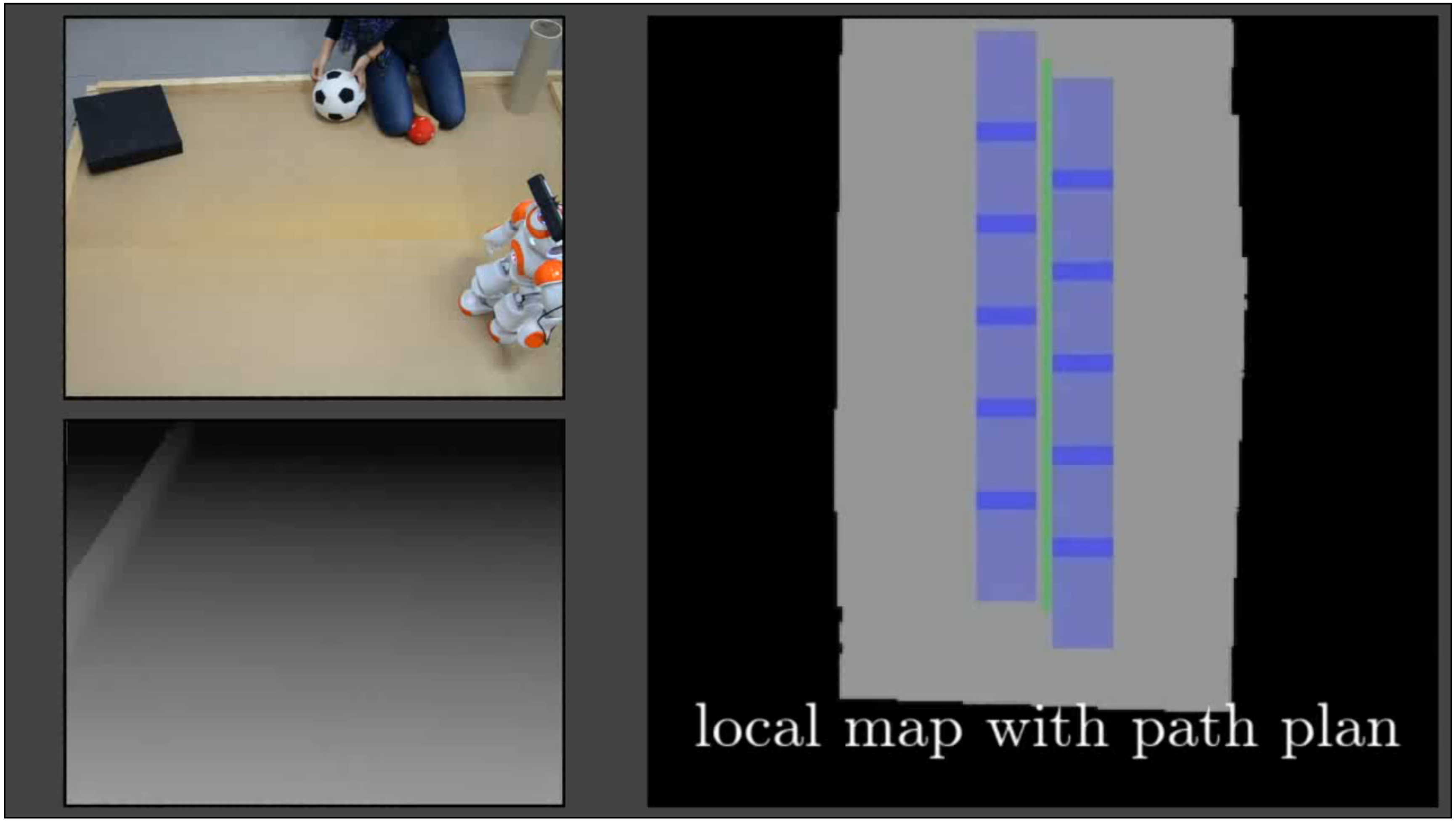
Map 2	A* (small)	A* (large)	Our approach
Planning Time	67.2 ms	75.0 ms	**48.7 ms**
Path Cost	6.9	5.0	**4.3**
Expanded Nodes	22374	16805	5852

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

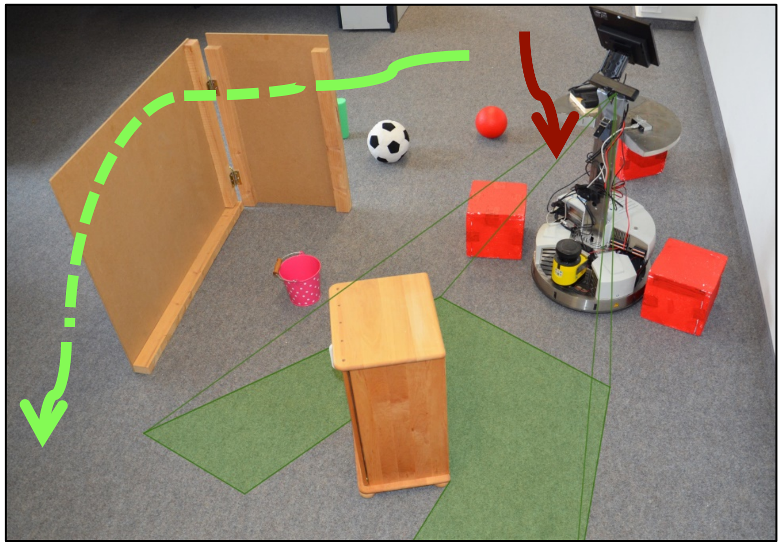
Challenges

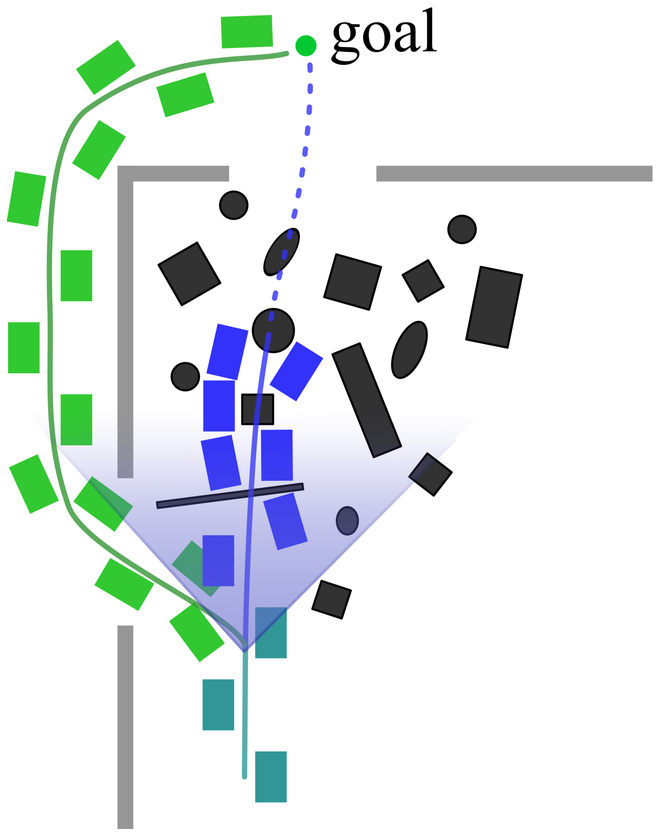
- Accurate sensing of obstacles
- Precise motion execution
- Obstructed view of the sensor

Navigation through Cluttered Regions

Leads to

- Decreased velocity
- Frequent turns
- Potentially getting stuck





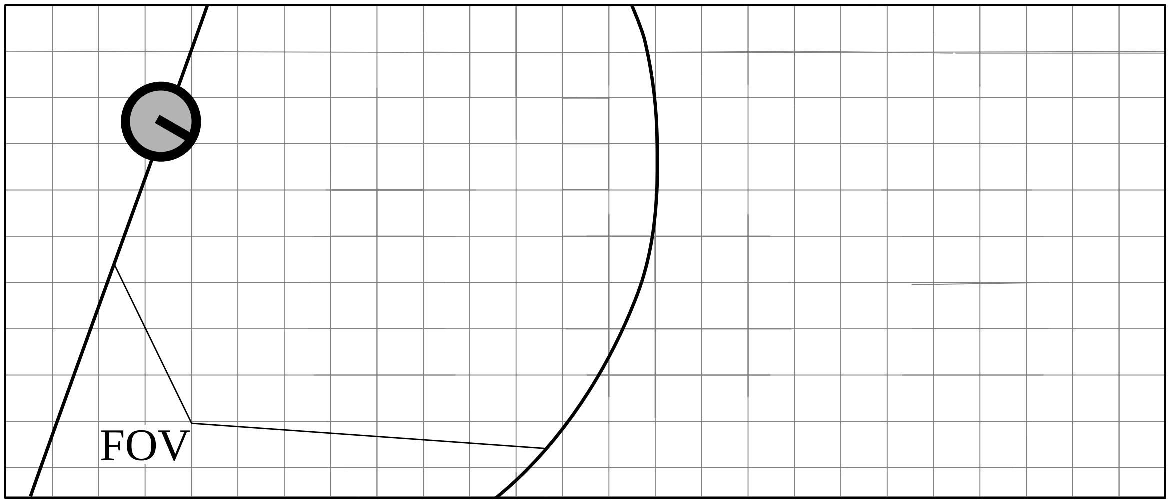
Our Approach

- 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

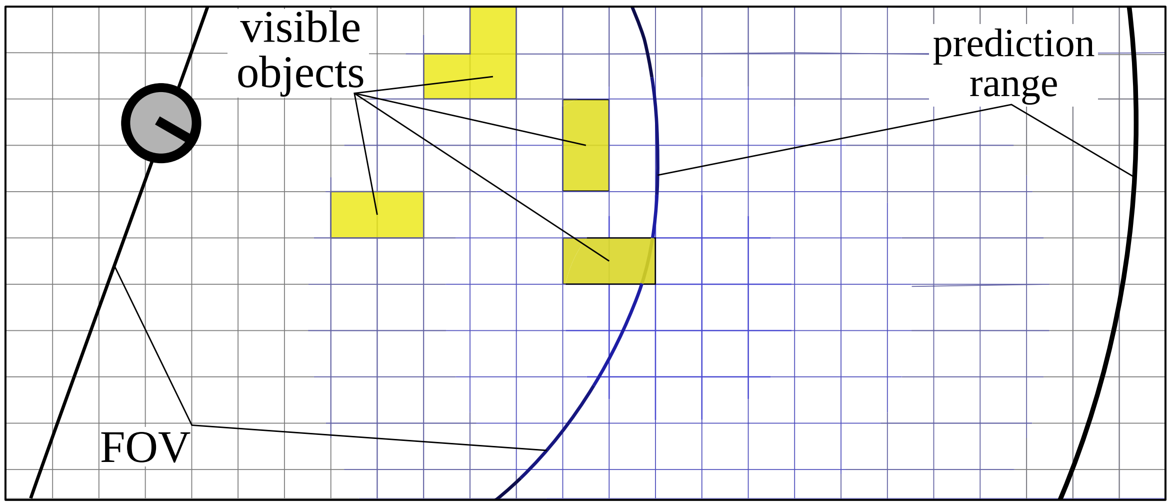
Occupation Density and Costs

- 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

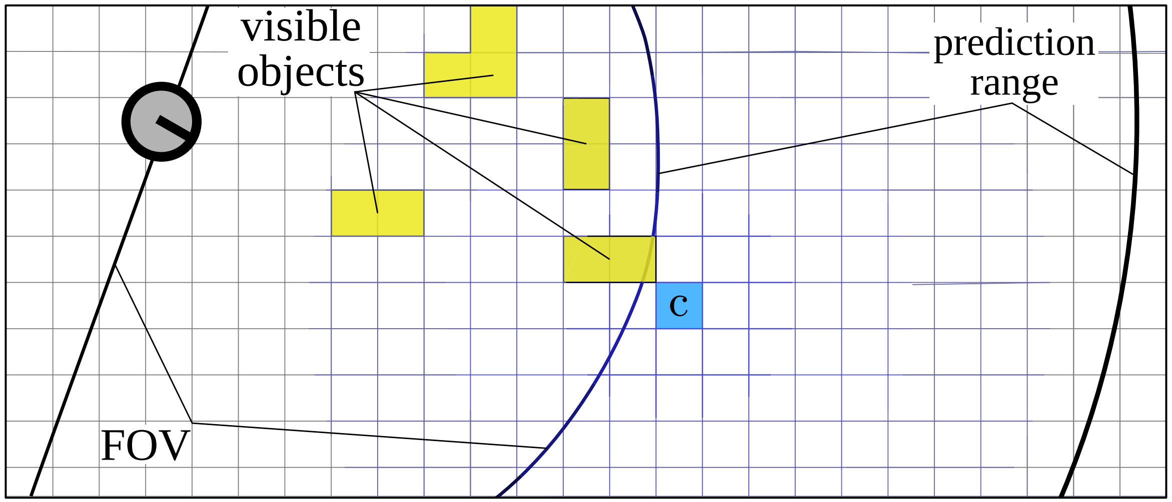
Estimated Occupation Density



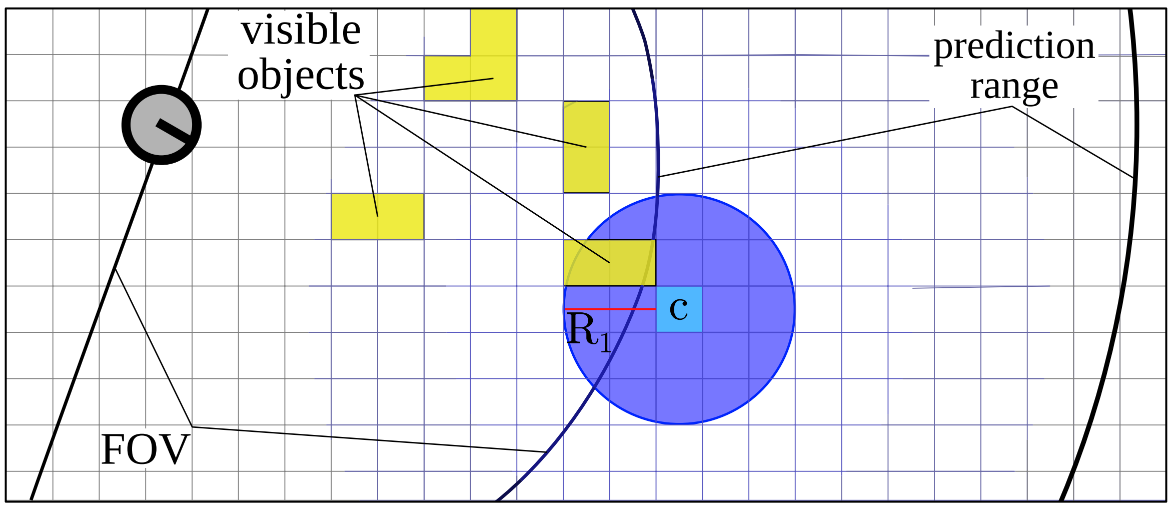
Estimated Occupation Density



Estimated Occupation Density



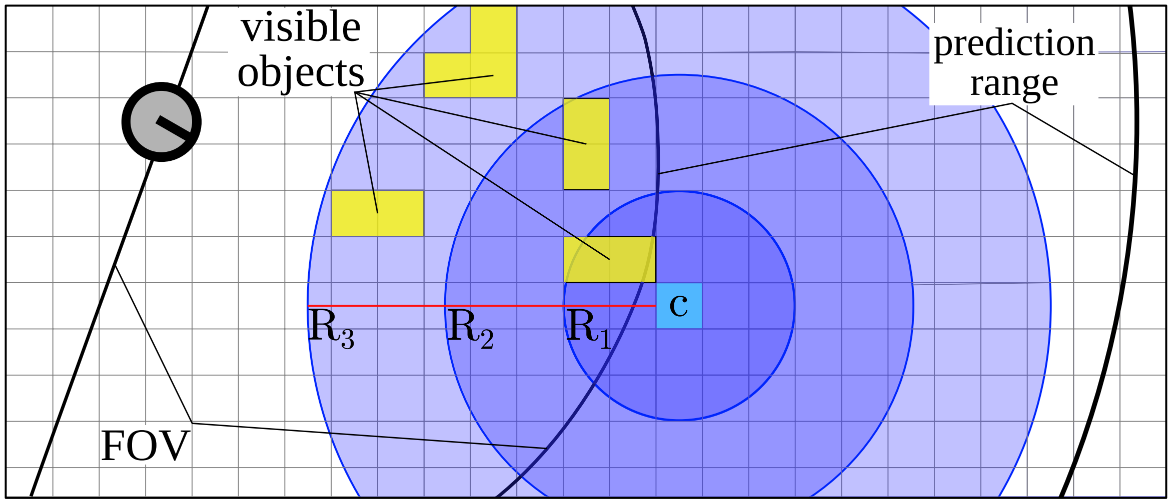
Estimated Occupation Density



Estimate the density ρ inside circle

$$\rho = \frac{\#\ \text{occupied cells}}{\#\ \text{cells}}$$

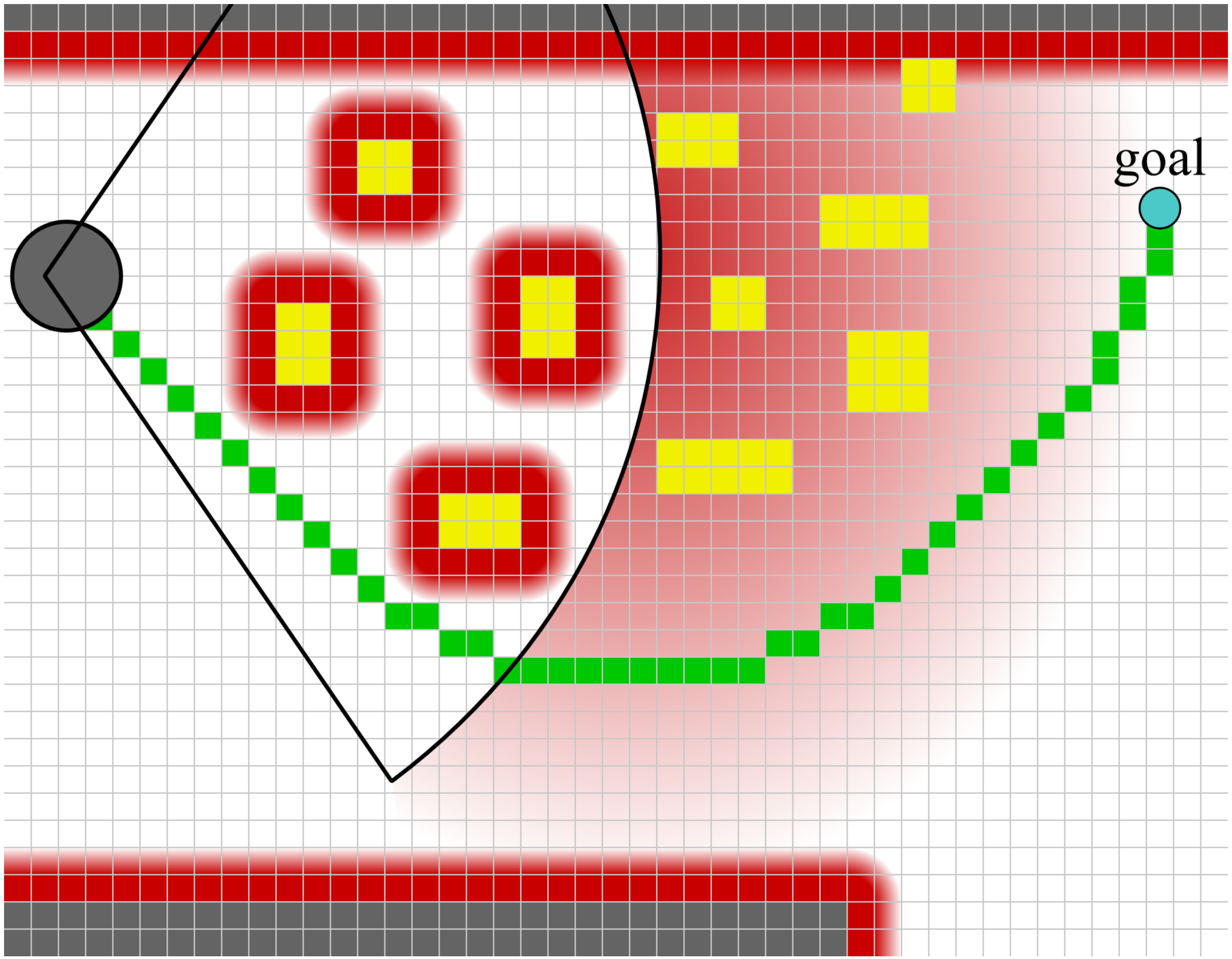
Estimated Occupation Density



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

predicted navigation costs: $C_{predict} = \alpha \cdot \rho$

Cost Grid Map



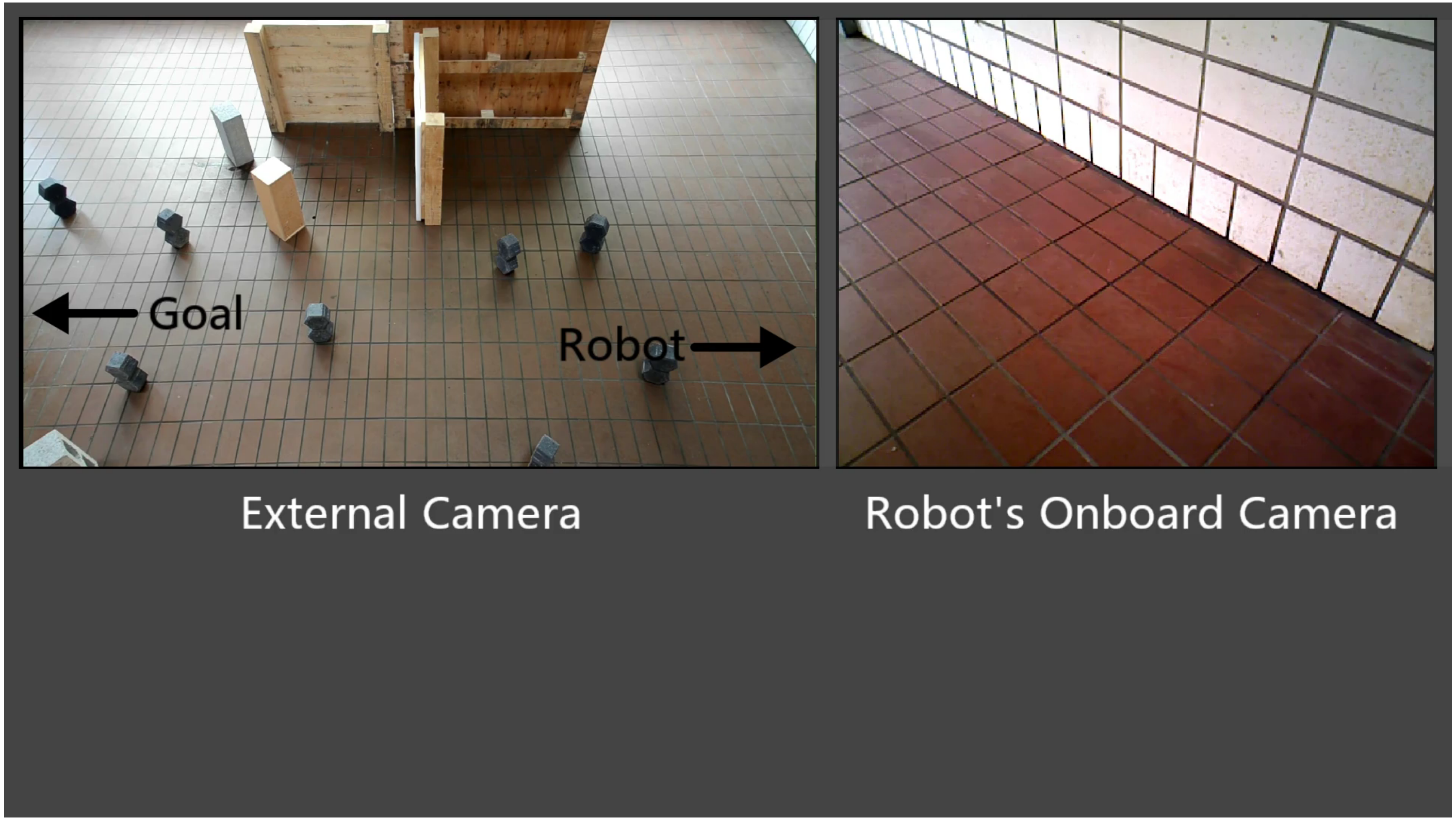
So far: Tested on Wheeled Base

ASUS Xtion Pro Live
for object detection

Robotino robot,
Festo Didactics

laser scanner for
localization

Foresighted Navigation



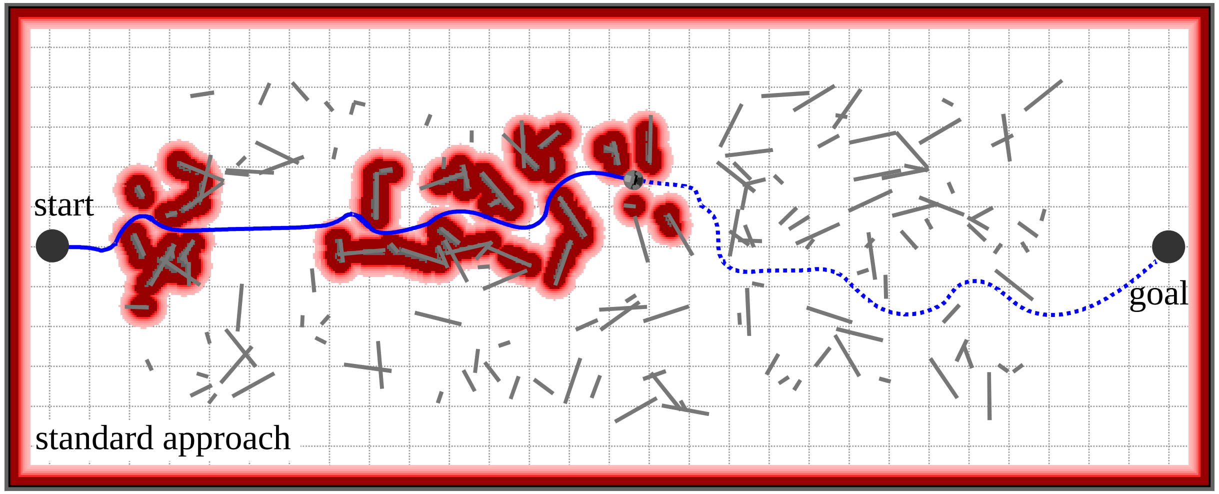
Regier et al., IROS 2016

Experimental Evaluation

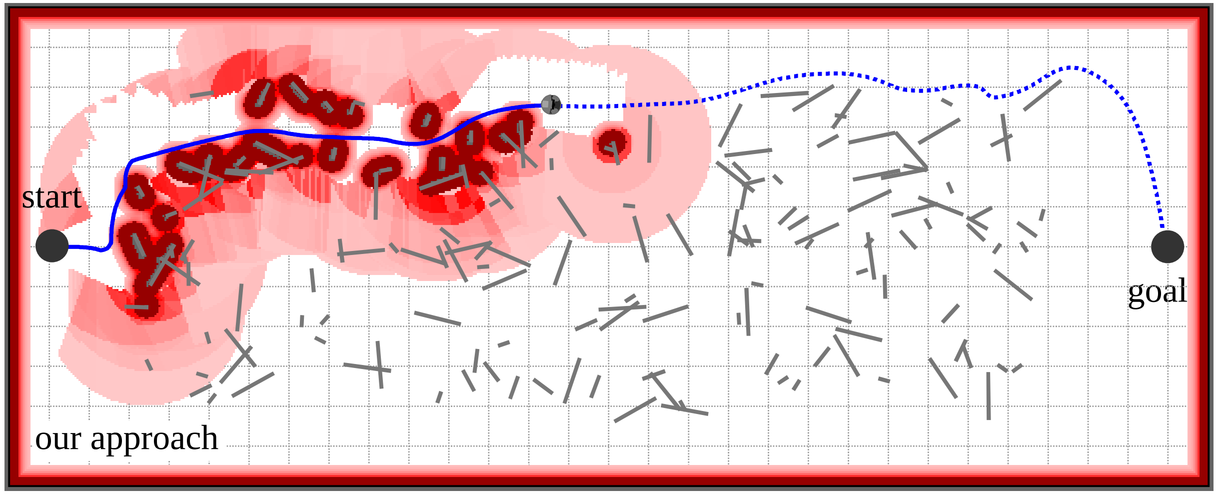
- Extensive simulation experiments
- Randomly sampled objects within a rectangular area of size 23×8 m^2
- Obstacle density as a parameter: average number of objects per 1 m^2

Simulation Result

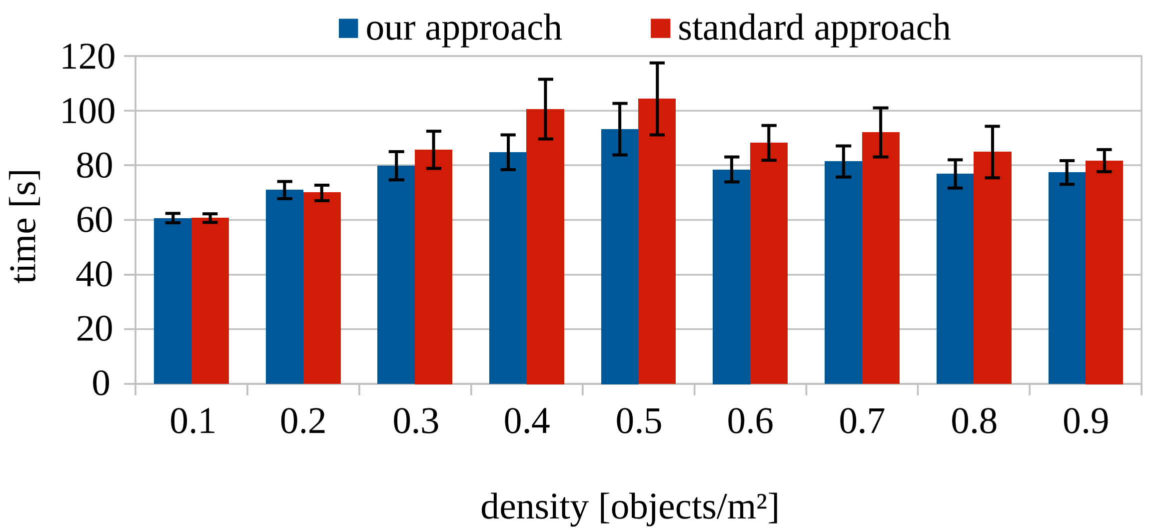
w/o
prediction



with
prediction



Results



- 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!