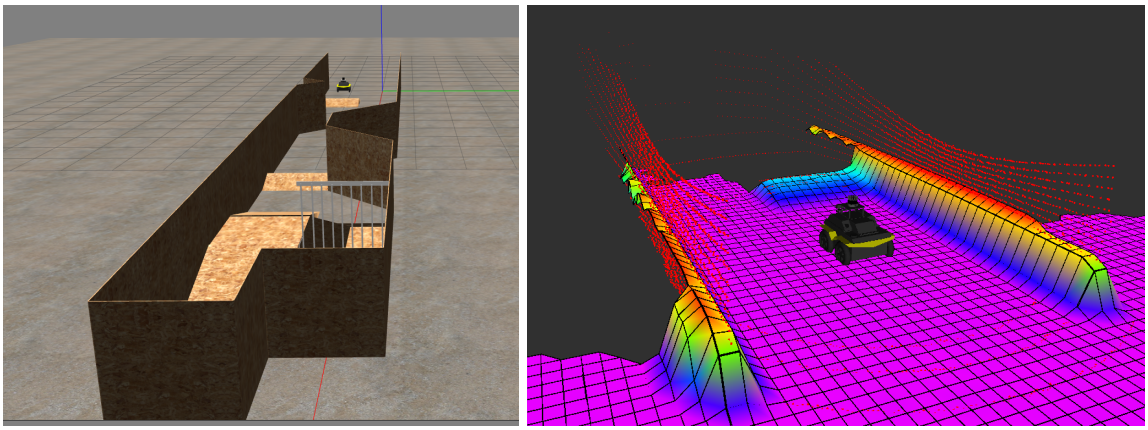
UNIVERSITÄT ZU LÜBECK
INSTITUTE FOR ROBOTICS
AND COGNITIVE SYSTEMS

Rescue Robotics
24.June.2021

Lars Schilling
schilling@rob.uni-luebeck.de

Exercise 3. - Path Planning

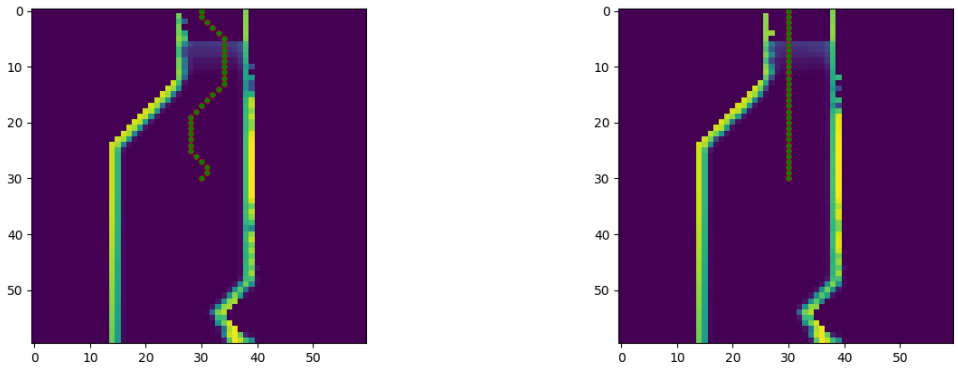
In this scenario the fire department wants to implement an algorithm for path planning for their UGV and evaluate it in their test environment.



(a) Gazebo **(b)** RVIZ

Figure 1 Exercise Illustration

Fork following repository: `https://github.com/l-schilling/rescue_jackal_exercise` and clone it to your VM. You can submit your solution as zip-file via e-mail. The deadline for this exercise is the **13.07.2021**.
The robot is equipped with a 3D LIDAR and the *Elevation Mapping* package will be used to create a surface map. This surface map can be represented as a height map. For the path planning we want to use the **A* algorithm** provided by the pathfinding package (`https://pypi.org/project/pathfinding/`, can be installed with: *pip install pathfinding*). The A* algorithm can work on any kind of cost map. This can be a height map as well as a distance map. As shown in Figure 2 (a) a path that is only optimized based on the height map and not on the distance can be inefficient.



(a) A* algorithm only based on a height map, not optimized based on distance

(b) A* algorithm based on combination of height and distance map

Figure 2 Exercise Illustration

Therefore we want to consider both the height and the distance cost maps and combined them into a combined cost map as shown in Figure 3.

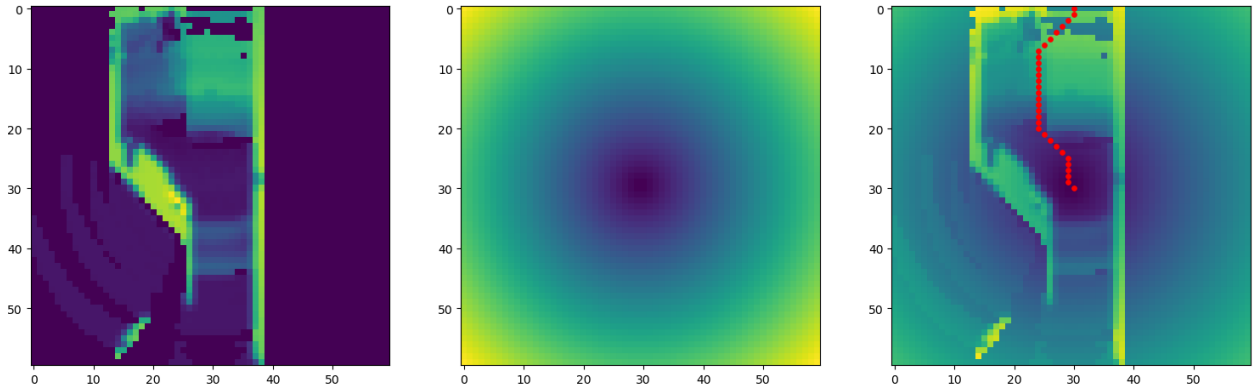
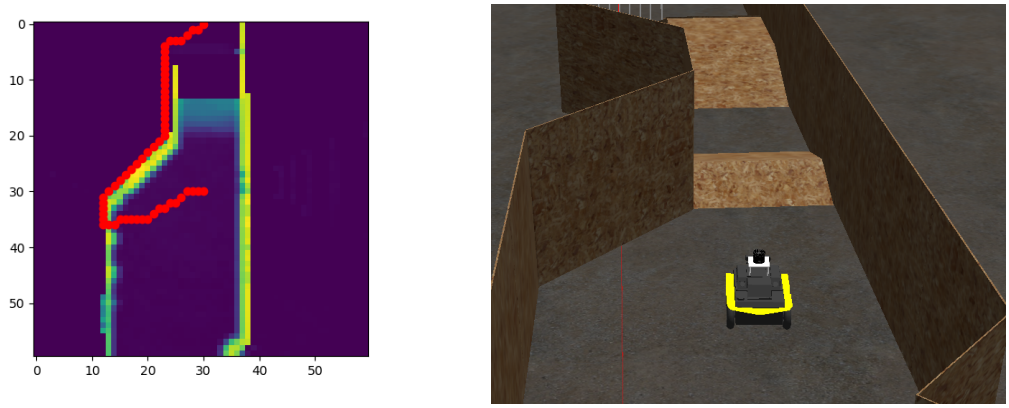


Figure 3 Height cost map, distance cost map, combined cost map

However choosing the right representation of a cost map is not trivial. As shown in the following picture an algorithm may favor going through a wall instead of a small steady slope.



(a) Unfavorable cost map

(b) In Gazebo

Figure 4 Exercise Illustration

(a) Integrate the pathfinding package

(b) Add a nonlinearity to the height map to create a height cost map

(c) Create a distance cost map, where pixels around the center have a low cost and get higher at the edges

(d) Define a combined cost map based on the height and distance cost maps

(e) Implement the A* algorithm from the pathfinding package on the combined cost map. With the starting point at the center of the image (30,30) and the goal at the upper edge of the image (30,0)

(f) Plot your height, distance and combined cost maps, as well as the A* path