UNIVERSITÄT ZU LÜBECK
INSTITUTE FOR ROBOTICS
AND COGNITIVE SYSTEMS

Rescue Robotics
03. June 2021

Lars Schilling
schilling@rob.uni-luebeck.de

Exercise 2. - Froniter Exploration

The fire department wants to use a boat to explore a flooded area. Due to the flooding and drifting material the area is not known and needs to be explored.

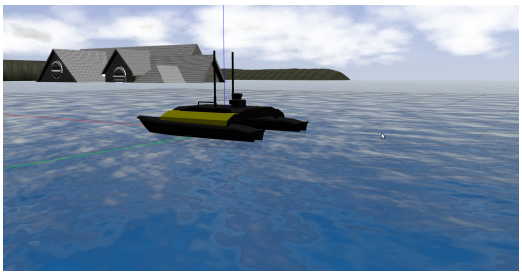

Figure 1 Boat Scenario

Fork following repository: `https://github.com/l-schilling/rescue_heron_exercise.git` and clone it to your VM. You can submit your solution as zip-file via e-mail or on Moodle. The deadline for this exercise is the **23.06.2021**.
The boat is equipped with a laser scanner and *gmapping* will be used to create a map. Through another package *hector_compressed_map_transport* it is possible to receive an image of the map around the boat. Furthermore the *move_base* package will be introduced to navigate the boat.
Your task is to write a frontier exploration algorithm. You will receive an image of the map of 64x64 pixels around the boat. On example of this is given in Figure 2 on the left. The colors represent the status of each pixel: White is explored, gray unexplored and black is an obstacle.

Figure 2 Heron Axes

(1) Detect frontier points by using edge detection (e.g. Canny Edge Detection from OpenCV). Make sure that only valid frontier points are chosen. The blue dots in Figure 2 in the middle represent valid frontier points. Notice that there are no valid points in the area with obstacles. Tune the parameters of the edge detection that you receive similar results.

(2) Find a representation for the information gain. One example would be to look at the area of unexplored terrain (Figure 2 left in gray) and use a 2D Convolution filter from OpenCV. An averaging filter kernel like K can be used to take into account how much unexplored area is around each pixel and thus gives

knowledge about the possible information gain at each pixel. The larger the kernel size the more area around each point is taken into account.

$$K = \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

The result of such an operation is shown in Figure 2 on the right.

(3) Of all frontier points find the point with the maximum information gain. In Figure 2 this point is shown in the middle as a red dot. Note that such an algorithm always prefers triangle shapes.

(4) Finally send the goal as PoseStamped message on the topic */move_base_simple/goal*.