

ROB05 – Mandatory project (06/10) - The Cup Collector

Problem:

The cantina at SDU has in a period of time collected empirical data that suggests that researchers and students at SDU are lazy and forgetful when it comes to bringing back used cups to the cantina. To avoid situations where there are no coffee cups and thereby losing coffee sales (or fresh students), the cantina has decided to assign the task of collecting cups from offices and hallways of SDU, to one person for 2 hours per day.

A master student with some knowledge on vision and point clouds has using a Kinect and a robot arm implemented an algorithm to detect and collect cups within a distance of 2 meters. Also he has implemented the stable grasping of cups that are within 1 meter of the robot. Unfortunately, this master student never attended ROB05 and therefore has no skill in navigating robots.

The problem is therefore presented to this year's class in ROB05.

The task contains 3 parts. All 3 must be targeted in the project.

Part 1. Planning

All rooms in the map must be checked in order to find cups. The robot must be within 2 meters of a cup in order to actually detect the cup and within 1 meter in order to collect it. Cups are marked in the map using one pixel with grayscale value 150. Cups can be offloaded at the two offloading stations in the cantina. The offloading stations are represented with pixel values 100. The robot must start and end at an offloading station.

You are free in regard to choice of algorithms. However, please document what algorithm you choose, how many kilometres the robot moves and how long it takes to calculate the path the robot takes.

Part 2. Coverage

The Dean feels that it is not economically justified to buy a robot system for 80.000 euro just to collect cups. Hence, it is interesting to have the robot do a second task, namely washing of the floors. Therefore calculate a coverage path that covers most of the floor in the map. Again the robot must start and end at the offloading stations.

Again, you are free to choose algorithm and you should document the choice, the distance in kilometres that the robot moves and how long it takes to compute the coverage path.

Part 3. Localisation

Finally, a method to compute the state (configuration) of the robot is required. The method should be applied to a real system such as the Nexus platform from the course and due to the size of the university it is important that the robot is able to use features to precisely measure its whereabouts. These features could be based on the Hokuyo 2D laser scanner mounted on the robot.

Again document what algorithm, and test how well it performs. You should at least write what model you choose for the robot and show that the localisation works better than odometry alone.

Note:

The map "complete_map_project.pgm" use a scale 10pixel:1m.

What to hand in:

The hand-in consist of two parts:

1. A program that solves 1, 2 and 3. This should be handed in electronically via e-learn at the 15'th of December at 13.00 hours.
2. A report that documents the solution of 1, 2 and 3. Should be delivered at the same time as the program.

Limitations and info:

- The map use the scale 10px to 1 meter
- The robot can only carry 20 cups at the time
- The robot is NOT a point robot. It has a circular shape with a radius of 0.4m
- The robot can drive up to 5km per hour
- The elevators are denoted with pixelvalues 128,129,130,131 and 132.

Stuff that should go into the report:

- Method
 - Shortly how you solved the problem
 - Only describe algorithms in detail if you changed them in relation to the literature.
- Results
 - How long does it take the robot to inspect all rooms?
 - How long does it take to find cups and deliver them to the offloading station.
 - How long time does it take to wash the floors?
 - How good is the localisation?
- Conclusion:
 - What works, what does not? Why?
 - Should the dean invest in a mobile robot as described above? Could it become feasible when a man hour costs 28 euro?
- Future work:
 - What changes could make the solution more optimal/feasible?
 - Algorithm wise?
 - Hardware wise?