University of Liège
Faculty of Applied Sciences

Introduction to Intelligent Robotics

Implementation of a youBot in a simulation environment

Pascal Leroy
Aurélien Wérenne

Academic Year 2016-2017

Contents

1 Abstract

The goal of the project is to program a youBot to put away groceries around a virtual house. Groceries are waiting in the hall. The youBot is provided with a program that allows it to move the groceries to their appropriate storing locations. The whole project is divided in several milestones among several categories: navigation, vision, manipulation, etc. We have chosen to implement three of them.

2 Introduction

Three features have been explored in order to program the youBot:

- Building a map using accurate localization.
- Building a map without accurate localization.
- Finding baskets and tables in the map.

Beforehand, a recall about some important hypothesis given for this simulation:

- The two tables are in the range of the Hokuyo sensors from the start position.
- Baskets are always close to room/hallway corners.
- The tables and baskets are the only cylindrical objects resting on the floor
- The size of the house is fixed.

We also made the hypothesis that the robot is able to turn on itself from its starting position. Concerning the report, it is divided into several sections. First, two sections about the general implementation: a state machine to control the robot and a GUI to see details about the current state of the youBot. Following, three sections about the different milestones explored, their current state of implementation, the different algorithm chosen and a result statement. Finally, a conclusion, the possible improvements and limitations for this project.

3 State Machine

There robot is executing following a state machine composed of seven states:

- Control: depending of the flags, switch to the right state.
- Rotate: execute a rotation on itself.
- Move: move towards a target position using the Pure Pursuit Controller of the matlab Robotic toolbox.
- Recovery: recover from a stuck position.
- Goal: compute the next target position. If the youBot is exploring unknown environment, it is computed using a customized frontier based approach. If the youBot travels from baskets to baskets, it simply select the next one.
- Transition: once the map is fully explored, compute the location of basket and table in order to visit them.

- Finish: finishing the exploration of the map or shut down the youBot after having visited all the baskets.

There are also four flags:

- flagMapping: set to one if the youBot is exploring the map and zero otherwise.
- flagComputeGoal: set to one if a new target position is required and zero otherwise.
- flagStartRotate: set to one if a rotation is needed to observe more of the environment and zero otherwise.
- flagRotateBeforeMove: set to one if the youBot need an initial rotation in order to be in the right direction for its path and zero otherwise.

4 Graphical User Interface

In order to facilitate debugging and visualise the decision of the robot, the current state of the state machine and the map.

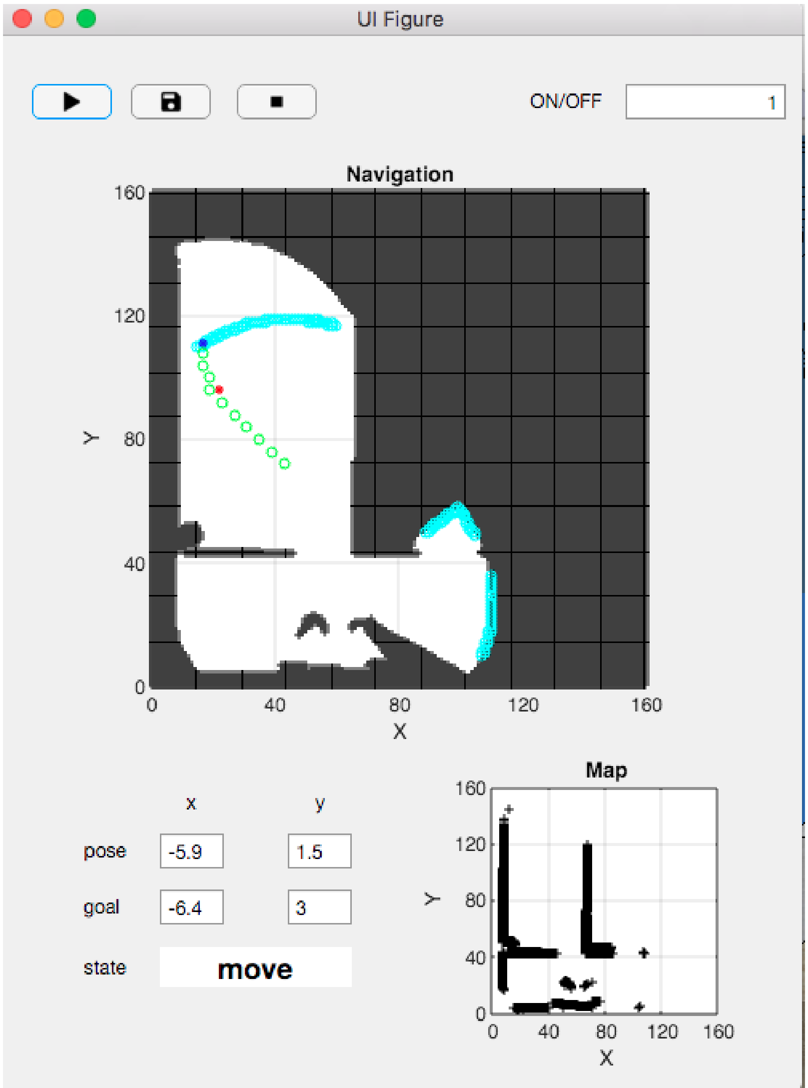

Figure 1: GUI View during exploration

5 Mapping using accurate localization

Knowing the accurate pose of the robot, the task is to built a map of an unknown and static environment with obstacles such as walls, tables and basket. The youBot is using a lazer range sensor (hokuyo sensor), an occupancy grid and the function *rayInsert* in order to perform this mapping.

5.1 Path planning

Finding the next target position during the mapping by making a trade-off between information gain and total distance travelled is the goal of this section. This is done by several steps:

1. Frontier detection: Using two different maps of the current environment, free space that are at the frontiers of the unknown environment is detected.
2. Frontier selection: Using a convolutional filter on the frontier points, only of a subset of them is selected. This subset represents the points that are located in the most unknown zones.
3. Distance evaluation: The nearest frontier point from the youBot is computed using a Probabilistic Road Map algorithm.
4. Path computing: Using the D* algorithm, the path toward this nearest point is computed.

5.2 Results

Using a grid of one cell per 20cm (here a 80*80 cells grid for the map), the youBot is able to fully explore the given map in between 2'30" and 4' which is quite good. But using this resolution for the grid doesn't give good results for the vision part. That is why the used resolution is of one cell per 10cm (so 160*160 cells).

6 Map analysis in order to find baskets and tables

In order to grasp and store the groceries, the youBot has to be able to find where are the tables and the baskets. In order to do this, the youBot will analyse the built map in order to find circles, differentiate if it is a basket or a table based on the hypothesis and will also compute an "access point" in order to be able to visualize those places from the best position possible.

6.1 Steps

There are several main steps in order to find the baskets and tables. The first ones are the same for both baskets and tables:

1. Take this binary map as a binary image such that the obstacles are in white and the free space is in black.
2. Perform a close operation on it in order to have a better view of walls, fill useless blank space.
3. Detect circles using circular Hough transform.

From there, due to imprecision of the sensor, the fact that it is a simulation and that the Hough transform may not be one hundred percent accurate, more then the seven circles are often detected. In order to find the right ones, the hypothesis are taken into account such that the method to find basket or table differ.
To find the baskets, the fact that there are in corners is used.

1. Using two convolutions in order to detect vertical and horizontal walls into the image of the map, and summing those two in order to detect the corners, the circles which are not in significant corners are thus excluded.

2. In order to find the right access point, the visible curve of the circle on the image is searched around the centre of the circle using Harris–Stephens algorithm that will find the corners (not the ones made by walls but corners made by pixels in the image) and thus evaluate the orientation of the corner made by walls such that it is possible to compute the frontier point that is the closest one from the basket and that can visualize it.

To find the table, the initial pose of the youBot is used with the *rayscale* function (that simulate the hokuyo sensor) in order to detect the circles that are in range of the sensor from the starting pose.

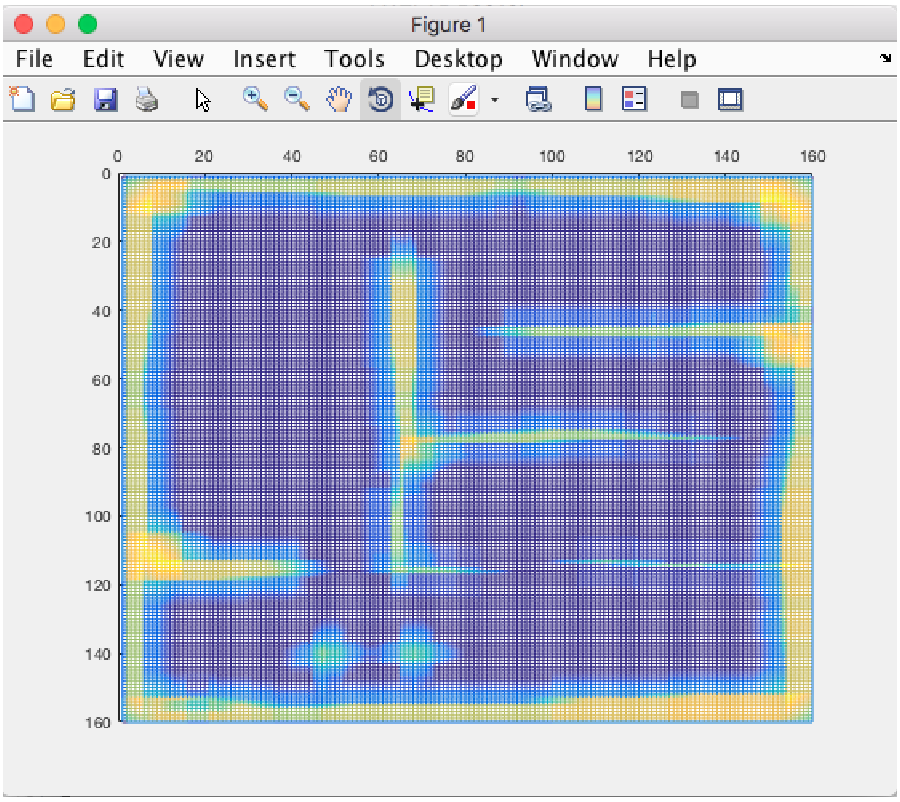

Figure 2: Filter detection of corners

6.2 Results

As introduced in the section 5.2, the resolution of the grid affects the results of this section. Moreover, sometimes, circles are detected in corner where there is no basket since this corner is not as smooth as it should be.

Figure 3: Orientation selection of frontier points

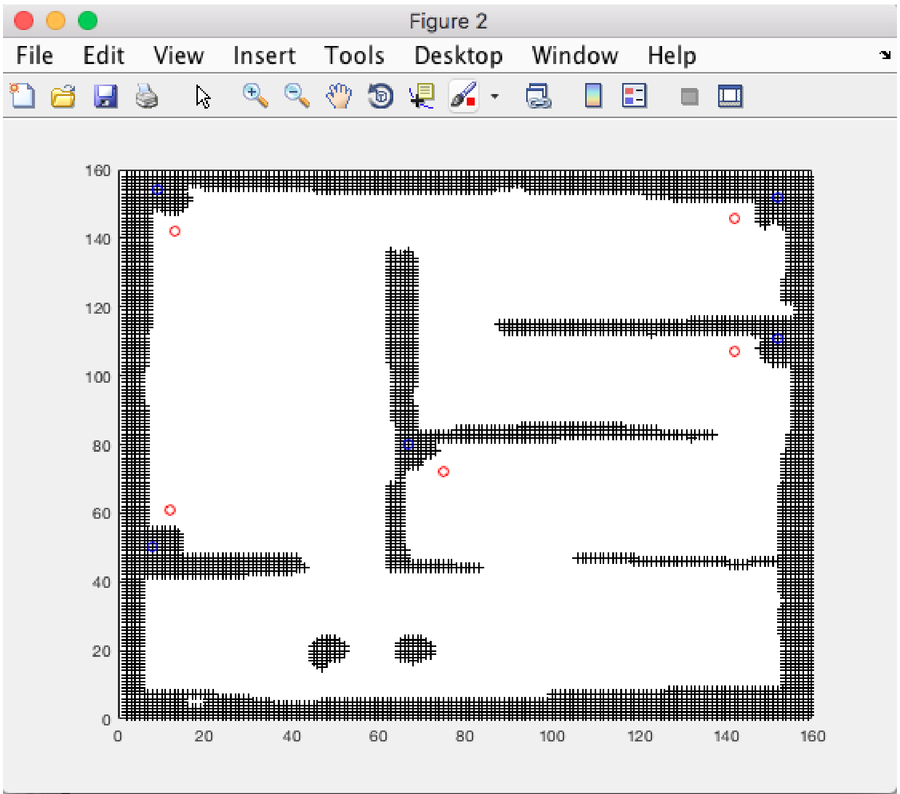

Figure 4: Display of access points near to baskets

7 Simultaneous localization and mapping

The goal of this milestone was to do the mapping of the unknown environment without using the localization of the simulator, such that the youBot is not aware of its pose.

The algorithm investigated is the scan matching algorithm. The purpose of this algorithm is to compute and compare two different scan occurred at different time step and estimate the transform.

7.1 Results

Despite a lot of investigation, the results are not good enough to integrate this algorithm into the robot.

8 Further improvements

- During the exploration, while the youBot is moving toward its destination, it is possible that it gains all the information before reaching this point during its travel. An improvement would be to stop the travelling earlier.
- During the exploration, compute next target position during the movement. (pipelining)
- Decision-making of stopping the exploration has lack.

9 Conclusion

Milestones A1 and C1 tends to have good results. Globally, this project was very interesting. Lot of trials and errors have been made in order to find the best parameters. Teamwork is always a good experience, different points of view lead to better development.