Objects Manipulation

Alex Lavriv and Vladimir Shargorodsky
Advisor: Prof. Ronen Brafman
20.06.2019
The goal

Using a robotic arm, rearrange objects positions and remove obstacles in order to reach and grab the desired object:

The red object is unreachable for the robotic arm.

The red object is reachable for the robotic arm.
Basic configuration video
Limitations

● The arm can push objects back or push them aside.
● The arm cannot grab the green (obstacle) objects, so moving an object towards the arm is impossible.
● The arm cannot access an object behind another one.
● Limited reaching area: the arm movement is limited by its length and joints state.
Technology

ROS - Open source, publisher/subscriber based robotic operating system.

GAZEBO - Open source, simulation environment.

Manipulator H - Open source, arm controlling library.

Ubuntu 16.04 - Open source, unix based computer operating system.

Node.JS - Open source, runtime environment.
Development process

Programming languages: C++ , Python and JavaScript.

The development process involved choosing between two popular arm manipulation frameworks:

1. Move-It motion planning framework
2. ROBOTIS Manipulator-H
Web Client

Objects current positions can be inserted using the developed web application. The border dimensions and object count are configurable.
Virtual Model

Using the gazebo simulator:

The obstacle objects are represented as green boxes.

The desired object is represented as the red cylinder.
Architecture

User Control Logic

- Web Client
- NodeJS Server

Business Logic - ROS Nodes

- A* algorithm
- Object Manipulation
- Manipulator H
- Robotic arm
The goal

Using a robotic arm, rearrange objects positions and remove obstacles in order to reach and grab the desired object:

The red object is unreachable for the robotic arm.

The red object is reachable for the robotic arm.
Algorithm

A* with the Manhattan distance heuristic:

Algorithm:

- Each object is represented as a point in an NxN grid.
- Every possible board configuration is represented as a Node.
- Using the A* algorithm, we find the shortest path from the initial configuration to the goal configuration.
- The Manhattan distance heuristic is useful for 4 directional movement systems.
The A* Algorithm

- \( f(n) = g(n) + h(n) \)
- The algorithm picks the next node by the minimal \( f(n) \) value of all the nodes from the current one.
- Our version is modified to keep the arm limitations in mind.
References

Manipulator-H:

ROS:
https://www.ros.org/

A* search algorithm
https://en.wikipedia.org/wiki/A*_search_algorithm