Team Description Paper – Team AutonOHM

Jon Martin, Daniel Ammon, Helmut Engelhardt, Tobias Fink, Tobias Scholz, and Marco Masannek

University of Applied Science Nueremberg Georg-Simon-Ohm,
Kesslerplatz 12, 90489 Nuremberg, Germany
{Jon.Martingarechana,EngelhardtHe57850,
Finkto51573,ScholzTo52032,MasannekMa61828}@th-nuernberg.de
daniel.ammon@evocortex.com
http://www.autonohm.de

Abstract. This team description paper describes the participation of Team AutonOHM in the RoboCup@work league. It gives a detailed description of the team's hard- and software and the previous achievements. Furthermore, improvements for the participation in future competitions are discussed.

1 Introduction

The AutonOHM@work team at the Nuremberg University of Applied Science Georg-Simon-Ohm was founded in September 2014. The team consists of Bachelor and Master students, supervised by research assistants and a professor.

The team is divided into groups, each developing specific parts of the robot control software, such as manipulation, object detection, localization and navigation. AutonOHM participated for the first time in the German Open 2015, achieving the 5th place out of seven. Since this was the first competition ever, the team was satisfied with the result. In 2016, the team participated in the RoboCup World Championship in Leipzig, achieving the 5th place out of nine. This time, a better navigation performance was shown, a bigger variety of objects were grasped and some new functions such as box and barrier tape detections were implemented. With new team members and new motivation, this team competed in the RoboCup German Open tournament in 2017 in Magdeburg. Considerable improvements of the perception and navigation lead our team to the 1st place in this competition. However, the current gripper is still a big handicap, as is not able to pick up heavy objects.

At first, this paper shows the team's hardware concept (chapter 2). In chapter 3, the main software modules such as the state machine, localization and object detection are presented. Finally, the conclusion provides a prospect to further work (chapter 4).

2 Hardware Description

We use the KUKA omni directional mobile platform youBot (Figure 1), as it provides a hardware setup almost ready to take part in the competition. At the end effector of the manipulator, an Intel RealSense 3D SR300 camera has been mounted for detecting objects. This 3D camera has been chosen due to its ability to furnish a 3D point cloud in short distances. Next to the camera, we replaced the standard gripper from the youBot with a soft two-finger gripper. It allows grasping bigger and more complex objects more precisely. As already mentioned, this grasper is still not able to pick up heavy objects, e.g. a M20 bolt.

Two laser scanners, one at the front and one at the back of the youBot platform, are used for localization, navigation and obstacle avoidance. The youBot's default Intel Atom computer has been replaced with an external Intel Core i7-4790K computer, providing more computing power for intensive tasks like 3D point cloud processing. Table 1 shows our hardware specifications.

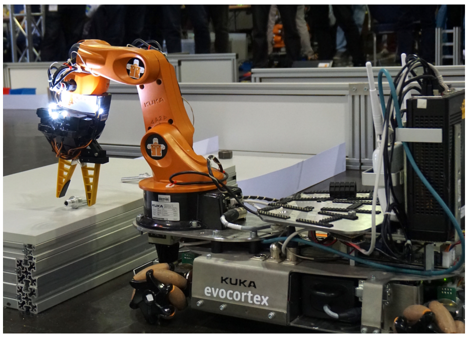

Fig. 1. KUKA youBot platform of the team AutonOHM.

Tab. 1. Hardware Specifications

PC 1	
CPU	Intel i7-4790K
RAM	16 GB DDR3
OS	Ubuntu 14.04
Gripper	
Type	soft, two-finger
Stroke	20 mm
Sensors	
Lidar Front	SICK TiM571
Lidar Back	SICK TiM571
Camera	Intel RealSense SR300

3 Software

We use different software packages to compete in the contests. Image processing is handled with OpenCV library (2D image processing and object recognition) and PCL (3D image processing). For mapping and navigation, we use gmapping and the ROS navigation stack.[1] Additionally, robot-pose-ekf package is used for fusing the data from the IMU and the wheel encoders, to provide more accurate data to the navigation and localization system.

The main packages we developed are further explained in the following sections. These include the state machine (controlling the robot, see chapter 3.1), modules for global localization and localization in front of service areas (see

[1] http://wiki.ros.org/

chapter 3.2) and packages for object detection (chapter 3.3) and manipulation (chapter 3.4). As a new feature for the RoboCup 2017 German Open contest, we developed a module for grasping moving objects (see chapter 3.5).

Furthermore, there are other small packages including:

task_planner After the task list is received from the referee box, the best route is calculated considering the maximum transport capacity and distances between the workstations.

youbot_inventory With youbot_inventory it is possible to save destination points, part positions and camera data for each workstation/shelve in the arena.

3.1 State machine

For the main control of the system, a state machine with a singleton pattern design is used. Every state is designed to be as small as possible.

In the initialization state, the robot receives the map and localizes itself on it. Then the robot waits on the “stateIdle” until a task is received from the referee box, which provides random tasks to perform for each test. The complete task is divided into smaller subtasks and managed in the ”stateNext”.

The first step is always moving the robot to a specific position. Once the position is reached, the state machine returns to “stateNext” to process the next task. In case of grasping, the robot approaches the service area, looks for an object, grasps it and stores it on the robot’s platform. For delivering an object, the robot picks the object from its platform and delivers it to the service area. Depending on the task, the robot will first look for specific surfaces, such as the red and blue containers or the cavities for the precise placement task, before delivering the object.[2]

3.2 Localization

For localization in the arena, we use our own particle filter algorithm. Its functionality is close to amcl[3] localization, as described in [1] and [3]. The algorithm is capable of using two laser scanners and an omnidirectional movement model. Due to the Monte Carlo filtering approach, our localization is very robust and accurate enough to provide useful positioning data to the navigation system. Positioning error with our particle filter is about 6 cm, depending on the complexity and speed of the actual movement.

For more accurate positioning, such as approximation to service areas and moving left and right to find the objects on them, we use an approach based on the front laser scanner data. Initially, the robot is positioned by means of the

[2] The state machine framework can be found on GitHub under our laboratory’s repository: https://github.com/autonohm/obviously

[3] wiki.ros.org/amcl

particle filter localization and ROS navigation. If the service area is not visible in the laser scan due to its small height, the robot is moved to the destination pose using particle filter localization and two separate controllers for x and y movement. If the service area is high enough, RANSAC algorithm [2] is used to detect the workstation in the laser scan. Out of this, the distance and angle relative to the area are computed. Using this information, the robot moves in a constant distance along the workstation.

3.3 Object Detection

To grasp objects reliably, a stable object recognition is required. For this purpose, am Intel RealSense SR300 RGB-D camera is used. In the first step, the plane of the service area is searched in the point cloud using the RANSAC algorithm. The detected points are then projected to the 2D image and used as a mask to segment the objects in the 2D image. For better results, the canny edge detector is used in order to find the borders in the segmented images. To detect an object, the following features are extracted and stored for each object: Length, width, area, circle factor, corner count and black area. With the help of a kNN classifier and the extracted features, the most similar item of the known items with their features is searched. To estimate the location of the object, its mass center is calculated. For the rotation of the object, the main axis of inertia is computed and used.

3.4 Object Manipulation

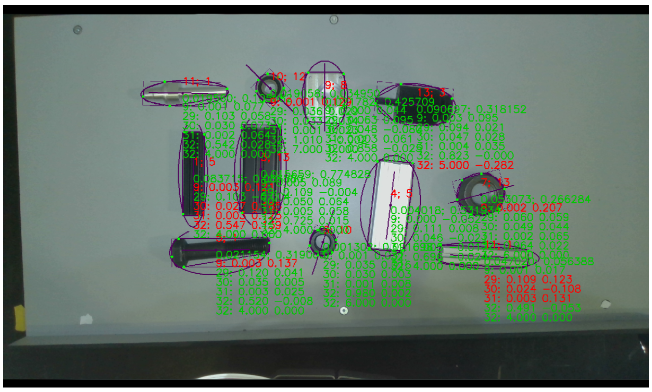

Fig. 2. Detected objects on a service area

For object manipulation, the results from the object recognition are used. Firstly, the robot navigates to a pregrasp position. Once the base has reached this position, the arm is positioned above the object. The object recognition is activated again to measure the final gripping pose.

For the calculation of the joint angles of the manipulator, a self developed inverse kinematic is deployed. To get feedback whether an object has been grasped, an Arduino based sensor was integrated into the end effector. In certain tasks, linear movement are helpful, for example for precision placement or picking up objects out of a box. To achieve linear movements, all joints need to be synchronized and points get interpolated between the start and goal destination. Fully synchronized movement will be integrated in the future. To be able to pick all kind of objects, a new grasper development has already been started. This new grasper will be controlled by two DC motors to reach a strong grasp and a reliable picking response.

3.5 Rotating Turntable

A new task for us in the "RoboCup@Work German Open 2017" tournament was the implementation of the round table. In this task, the robot needs to grasp objects in motion autonomously, which are placed on a rotating turntable. To achieve this functionality, several steps need to be implemented to be prepared for each possible option.

First of all, the robot will start in its normal object detection position and starts recording the 2D positions of the objects. With this 2D data, a RANSAC based algorithm calculates the center and the radius of the circular path. Having the necessary circle properties, the robot is then able to calculate the speed of the object and the amount of time which the objects need to arrive at the grasping position. Another feature is the detection of the table rotation direction so the youBot is able to grasp an object with various radius and speed, independent of the table rotation direction. Another feature of our implementation is the multiple object recognition. The first object which is detected by the camera will be tracked. New data from different objects will not effect the calculation. In the future, it is planned to improve the implementation by considering the orientation of each object to increase the reliability.

4 Conclusion and Future Work

This paper described the participation of team AutonOHM in the RoboCup@work league. It provided detailed information of the hardware setup and software modules like localization, autonomy, image processing and object manipulation.

Despite our good results in Robocup German Open 2017, we will not rest or stop developing our robot platform. Quite the contrary, we will take advantage of this tournament and improve all our tasks to be faster and more solid. The main priority is to upgrade the existing and well working foundation instead of implementing new features to the robot.

To improve the robots capabilities, we want to revise the energy concept for a longer run-time of the youBot platform. To increase the reliability of grasping and to get feedback about force and position of the gripper, we plan to develop a new grasping concept. Furthermore, a 3D simulation environment is planned. This will allow us a faster and more simultaneous testing of the developed software packages without the need of the youBot. To increase the reliability of the object detection, a 3D map creation of the workstation is desired. Also, the use of a more complex classifier or a neuronal network approach for better object recognition would be imaginable.

References

1. F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, 2(May):1322–1328, 1999.
2. Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.
3. S. Thrun, W. Burgard, and D. Fox. *Probablistic Robotics*. Massachusetts Institute of Technology, 2006.