RoboCup@Work 2018 Team AutonOHM

Jon Martin(✉), Helmut Engelhardt, Marco Masannek, Tobias Scholz,
Kay Gillmann, and Benjamin Schadde

University of Applied Sciences Nuremberg Georg-Simon-Ohm,
Kesslerplatz 12, 90489 Nuremberg, Germany
{jon.martingarechana,engelhardthe57850,
masannekma61828,scholzto52032}@th-nuernberg.de
http://www.autonohm.de

Abstract. This work presents the team AutonOHM which won the RoboCup@Work competition in Montreal 2018. The tests and main changes of the 2018 world cup competition are presented and a detailed description of the team's hardware and software concepts are exposed. Furthermore, improvements for future participations are discussed.

1 Introduction

The RoboCup@Work league, established in 2012, focuses on the use of mobile manipulators and their integration with automation equipment for performing industrial-relevant tasks [4]. This work presents our teams major improvements and changes with regard to last years work [5]. This year, the team has focused on further increasing the system stability by developing a new gripper, improving the inventory slots, speeding up task executions and developing a new approach for the rotating table test.

Section 4 shows the team's hardware concept. In Sect. 5 the main software modules such as the state machine, localization and perception are presented. Finally, the conclusion provides a prospect to further work of team AutonOHM (Sect. 7).

2 AutonOHM

The AutonOHM-@Work team at the University of Applied Sciences Nuremberg Georg-Simon-Ohm was founded in September 2014. In 2018, having most of the formal members yet taking part on the competition, the main goal was to defend the German and the World Championship titles. The team is organized so that each formal member takes care of a specific main task: Team coordination and state machine, navigation, perception, manipulation and rotating table. Newer students support the different tasks or develop new packages such as a faster task planner this year.

In order to retain the titles, the robots hard- and software were improved based on the knowledge summed during the past tournaments. This includes a

D. Holz et al. (Eds.): RoboCup 2018, LNAI 11374, pp. 500–511, 2019.
https://doi.org/10.1007/978-3-030-27544-0_41

Fig. 1. Team AutonOHM at the winners ceremony in Magdeburg

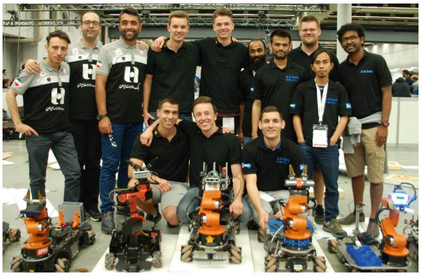


Fig. 2. Team AutonOHM with the second (b-it-bots) and third (mrl) placed teams in Montreal

new 3D camera model with a more accurate point cloud. The inventory slots are now more resistant against vibrations. The old gripper mechanism has been replaced by a single motor solution, which enables the robot to grasp heavy objects more reliably. Combined with high precision perception, the team was the only one that never lost the heaviest and most difficult object in the competition. In addition to the hardware improvements, the perception and manipulation for moving objects has been adapted to the new challenges added in the 2018 rulebook.

Even though remarkable changes have been done since the last year, most integration problems and bugs could be solved and the team was able to defend both titles against the strong competitors (Figs. 1 and 2).

3 RoboCup@Work

In this section we introduce briefly the tests and most remarkable changes of the 2018 RoboCup@Work world championship. As a common change to uniform the different tasks, the robots must now start in the starting position and end in the finishing position for every run. The 2018 rulebook release [3] contains more detailed information about the changes.

Arena: As in previous competitions, the arena has well defined start and finish positions and is entirely shut either by a wall or by yellow-black barrier tape (see Fig. 3). It contains workstations with heights of 0/5/10/15 cm as well as in shelves. The 0 cm areas are marked by the blue-white barrier tape. The Fig. 4 displays the map used by the robot for navigation.

Basic Navigation Test Removed: The BNT test had the purpose of demonstrating robots navigation and obstacle avoidance capabilities. However, these abilities can also be proven during any other test and the league has thus decided to remove the BNT test and distribute the points among the following runs. Teams get now 25 points per station reached during the whole competition.

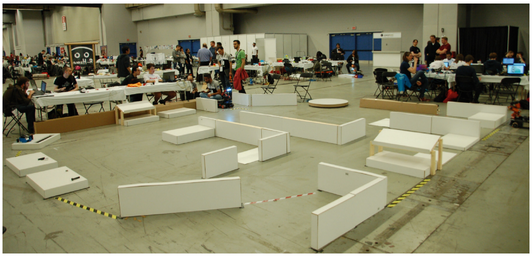


Fig. 3. The @Work arena during the RoboCup world cup in Montreal (Color figure online)

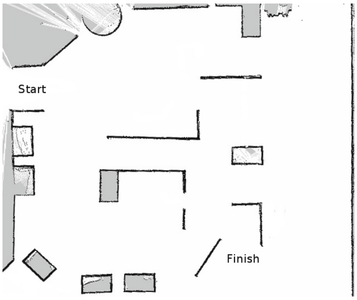


Fig. 4. Map used for navigation

Basic Manipulation Test: The purpose of the Basic Manipulation Test (BMT) is to demonstrate basic manipulation capabilities by the robots. Here, five objects must be grasped and delivered to a nearby workstation.

Basic Transportation Test: The purpose of the Basic Transportation Test (BTT) is to assess the ability of the robots for combined navigation and manipulation tasks. The robot receives the start and end positions of the objects to be transported in the arena and autonomously create a plan to perform this grasping and delivery tasks. This test is repeated three times with an increment of the difficulty and penalties during the competition. Here, unknown dynamic obstacles and yellow barrier tapes will limit the mobility of the robot.

Precision Placement Test: The purpose of the Precision Placement Test (PPT) is to assess advance perception and manipulation abilities. The robot needs to detect object-specific cavities and introduce the grasped objects into them.

Rotating Table Test: The purpose of the Rotating Table Test (RTT) is to assess the robot's ability to detect and grasp moving objects which are placed on a rotating turntable. This year, there are six objects laying on the turntable including three objects to be grasped, such as in previous years, and three extra decoy objects. Moreover, the objects position in the table is now defined by referees. As a result each object has its own circular path, tangential speed and position. The possibility to fix a grasping configuration for the robot in front of the table is eliminated. As in previous years, the direction of rotation of the table is fixed and the speed is set by the referees before just the test starts.

Final: The final round is a combination of all the above mentioned tests performed in a single round.

4 Hardware Description

Table 1 shows our main hardware specifications. We use the KUKA omni directional mobile platform youBot (Fig. 5), as it provides a hardware setup almost

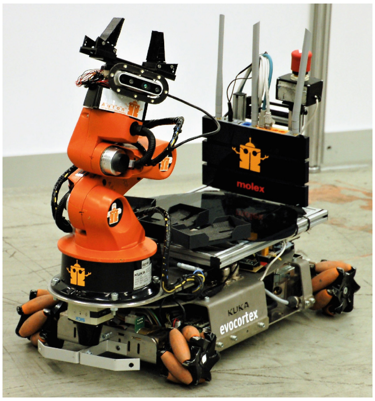


Fig. 5. KUKA youBot platform of the team AutonOHM.

Table 1. Hardware Specifications.

PC 1	
CPU	NUC7i7BNH
RAM	16 GB DDR4
OS	Ubuntu 16.04
Gripper	
Type	3D printed, parallel rail
Motor	Dynamixel AX-12A
Sensors	
Lidar front	SICK TiM571
Lidar back	SICK TiM571
3D-cam arm	Intel RealSense D435
2D-cam gripper	Endoscope Cam
3D-cam back	Intel RealSense D435

ready to take part in the competition. Nevertheless, we made some modifications for a better performance.

The platform comes with two PCs with hardware drivers installed, which we replaced by a single Intel NUC i7, because the default processors were outdated and caused performance issues. This main PC is used to control the base and arm of the mobile platform, as well as for image processing and task planning. The KUKA youbot also comes with a Hokuyu 2D-Lidar, which was replaced by two SICK SICK TiM571, one at the front and one at the back of the robot. They are used for mapping, localization, navigation and obstacle avoidance.

The standard endeffector of the Youbot was also replaced by a self developed parallel gripper. The gripper is based on a single Dynamixel servo motor which is attached to a 3D printed rail. Simple mechanics allow an efficient power transmission which enables the motor to grasp with its full torque rather than it being reduced by the lever in the old gripper version. The fin-ray fingers are custom printed out of rubber filament, making them soft and enabling them to close around grasped objects. They are also wider than standard FESTO fin-ray fingers, so they have an enlarged attack surface and therefore have more tolerance for very small and/or moving objects.

Both sides of the gripper mount are also used to mount the cameras used for perception. The main camera is an Intel RealSense D435 which has been chosen due to its ability to provide a 3D point cloud in short distances. The point of view can be changed with different arm positions, enabling different fields of view. The secondary perception camera is an endoscope webcam used to increase the precision while grasping moving objects. Its field of view points directly towards the gripper and therefore enables better timing of gripper controls. For the World Championship, an additional Intel RealSense D435 was mounted at the back of the robot for improving the barrier tape detection.

The robots inventory consists of three identical 3D printed slots mounted on an adaptable rail system. They are equipped with anti-slip pads, which prevent any movement of the objects, even with heavy robot vibrations.

5 Software Description

We use different open source 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 navigation-stack ROS-packages[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 software packages are based on ROS and explained in the following sections. These include the state machine (Sect. 5.1), global and local localization (Sect. 5.2) and packages for perception (Sect. 5.3) and manipulation (Sect. 5.4). We also improved the rotating table approach (Sect. 5.5). To perform the transportation logistics, a *task_planner* node processes the orders received from the referee box and calculates the best route considering the maximum transport capacity and distances between the workstations. This module finds the optimal solution up to five objects. From six objects on, we need to split the orders in groups of five due to the long computing time of the current solution, resulting on an suboptimal result.

5.1 State Machine

The main control of the robot is coordinated over the state machine in Fig. 6. It starts with an initialization state where the robot receives the map and tries to localize itself on it. From there, it moves to the “stateIdle” and waits for new tasks to perform. The Referee Box provides the orders which are processed by the *task_planner* node and sent to the state machine divided into a vector of smaller subtasks. The subtasks *Move*, *Grasp*, *Delivery*, *PreciseDelivery* and *RotatingTable* are now managed in the “stateRunning”. Once every subtask is finished it returns to the “stateIdle” to wait again for new tasks to perform.

The first subtask is usually a *Move* action performed over the *navigation* node. Depending on the required accuracy on the localization, the robot may execute a *fine navigation* approach. Both modules are explained in Sect. 5.2. After a specific workstation location is reached, the robot may look for a specific object, container or cavity on the workstation. In case of a *Grasp* subtask, the exact pose of the desired object is identified. For *Delivering* an object, the robot must recognize the exact pose of containers or cavities for *PreciseDelivery.* Once the desired pose is located, the arm manipulation is activated, whether for picking up and storing the object on the robot or for delivering it. The perception and manipulation nodes are explained in Sects. 5.3 and 5.4 respectively. In case

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

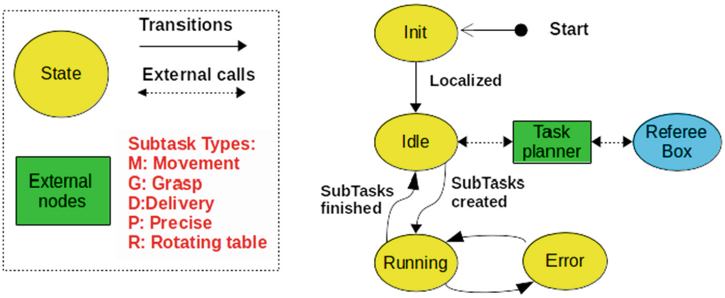


Fig. 6. Global overview of the AutonOHM State Machine

of a *RotatingTable* subtask, before grasping an object, a preprocessing step to determine objects velocity and pose in the table is required (Sect. 5.5). Once the manipulation subtask is finished, the robot moves away from the service area and returns to the "stateNextSubtask" to manage the following subtask to do (Fig. 7).

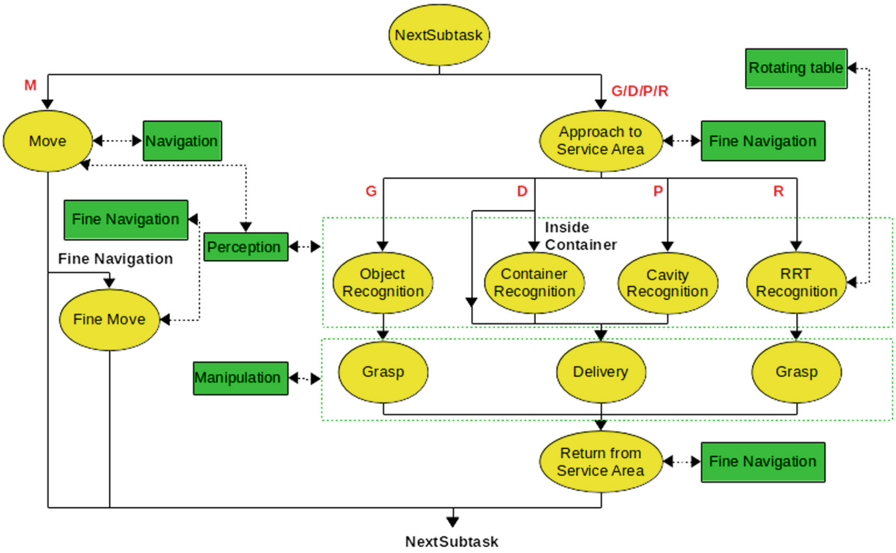


Fig. 7. The Running state is divided into substates where the SubTasks are managed

In addition, most of the states have error handling behaviors that manage recovery actions such as in case a navigation goal is not reachable, an object cannot be found or a grasping was unsuccessful. It is important to notice these failures and react to them by repeating the action or triggering planning modifications. The state machine framework can be found on GitHub under our laboratory's repository.[2]

5.2 Navigation and Localization

For the navigation, the ROS navigation stack has been used. The localization is based on a particle filter algorithm, close to amcl localization, as described in [1]. The algorithm is capable of using two laser scanners and an omnidirectional movement model. Due to the Monte Carlo filtering approach, our localization is robust and accurate enough to provide useful positioning with an approximate error of about 6 cm, depending on the complexity and speed of the actual movement.

For the fine navigation, 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 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. We achieved a mean positioning error of under 3 cm during a navigation benchmark tests performed in the European Robotics League local tournament in Milan.

5.3 Perception

This section introduces the implemented nodes for the different perception tasks. The object detection is presented first. Subsequent the detection of the barrier tape is described. Finally, the box detection is depicted.

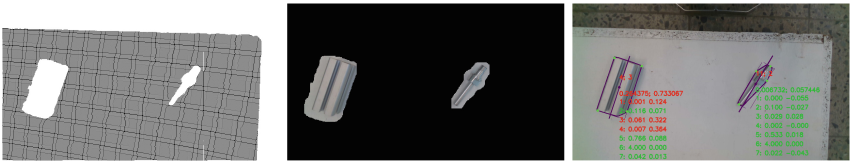
Object Detection: To grasp objects reliably, a stable object recognition is required. For this purpose, an Intel® RealSense™D435 RGB-D camera is used.

Firstly, the robot navigates to a pregrasp position. Once the base reaches this position, the arm is positioned above the service area. Due to the limited field of view, the robot base moves first left, then right so all the objects in the workstation can be discovered.

On each position, the plane of the service area is searched in the point cloud using the RANSAC [2] algorithm. Afterwards the detected points are projected to the 2D-RGB image and used as a mask to segment the objects in the 2D-image (Fig. 8a and b).

As all workstations have a white surface, the canny edge detector is used in order to find the concave border of the object in the segmented images for a more

[2] https://github.com/autonohm/obviously.

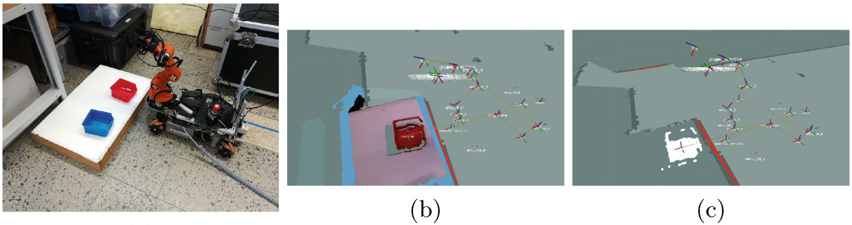


(a) (b) (c)

Fig. 8. Segmentation mask: The projected point cloud to camera's RGB image (a). Filled border and morphological operations (b). Classified objects (c).

accurate result. To classify an object, the following features are extracted: length, width, area, circle factor, corners count, height and black area. The distance to the workstation surface and the camera calibration matrix is used to calculate distance invariant values. With the help of a kNN classifier and the extracted features, the similarity to each previously trained item is calculated. With this information and the inventory information from the referee box, the best possible fitting combination for the detected object on the workstation 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. The robot will now move in front of the elected object and activate the object recognition again to obtain a more accurate gripping pose. For the newly introduced challenge of unknown orientation of the objects, the objects are trained from all possible orientations. The corresponding height of the detected object will be passed to the manipulation node for correct grasping. The use of the same features of the corresponding objects is an advantage of this approach. The features of black area and height are not considered, as they are not needed for a successful classification.

Box Detection: Some tasks require an object placement into a blue or a red box (see Fig. 9a).



(a) (b) (c)

Fig. 9. Box Detection: Blue and red box on workstation (a). Point cloud of workstation (b). Red filtered point cloud and mass center (c). (Color figure online)

The boxes are easily distinguishable from the background because of their color. Therefore a different strategy is used instead of the described object detection in Sect. 5.3. The advantage is a faster detection of the drop point. In front of the workstation the robot arm is moved in order to position the camera in a 45° angle to the workstation. Subsequently, the point cloud is filtered by the color of the searched box (Fig. 9c). If the filtered point cloud is too small, the robot drives closer to the workstation. If no colored points could be detected, the robot will move to the left side first, then to the right side, until a significant amount of points is found. After that the mass center of the filtered point cloud is calculated and passed to the manipulation node as the drop point for the object.

Barrier Tape Detection: Yellow/black barrier tapes are used to mark restricted areas in the RoboCup@Work competition. If the robot crosses this tape the team is penalized with point deduction (Fig. 10).

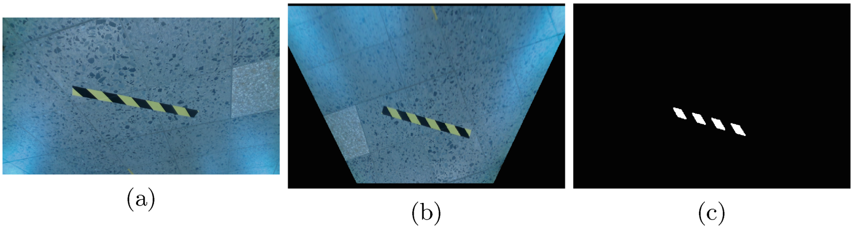


Fig. 10. Barrier Tape Detection: Camera image of the barrier tape (a) Birdview (b) Filter for yellow RGB and HSV values and HU-Moments (c) (Color figure online)

In order to detect this barrier tape the camera image is transformed in bird's-eye perspective. Next the image is filtered by RGB and HSV values, which correspond with the yellow part of the barrier tape. For the next step the HU-Moments are calculated and compared to filter out false shapes. Afterwards the detected shapes are transformed and saved in a global map. This gives the robot the ability to avoid the barrier tape even it is not visible in the camera image anymore.

5.4 Manipulation

The manipulation controller is responsible for arm and gripper controls, as well as for inventory management. It provides interfaces for arm positions, grasping or placing tasks and for linear arm movements (Figs. 11 and 12).

At the beginning of a grasp or placement process, it receives the target pose from the perception node. A self developed algorithm for the inverse kinematics and interpolation plans a linear and orthogonal trajectory to the workstation, object or container. This prevents the gripper from accidentally touching

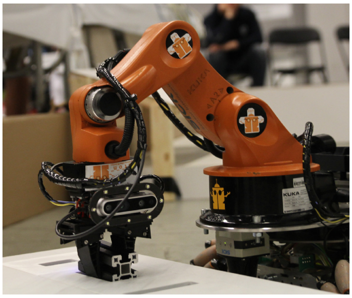


Fig. 11. Precise placement of objects.

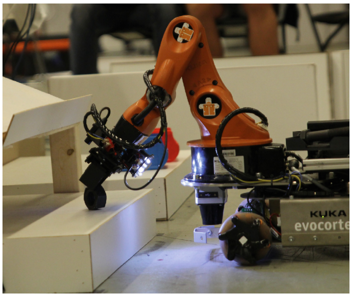


Fig. 12. Placing an object below the shelf without causing a collision.

or moving other objects lying on the workstation. Safety behaviors have been implemented during the grasping and placing process to ensure a reliable object handling and inventory management. In specific cases where objects are lost, the affected inventory slot is blocked to further use. The inventory state is broadcasted, so it can be used e.g. by the task planner.

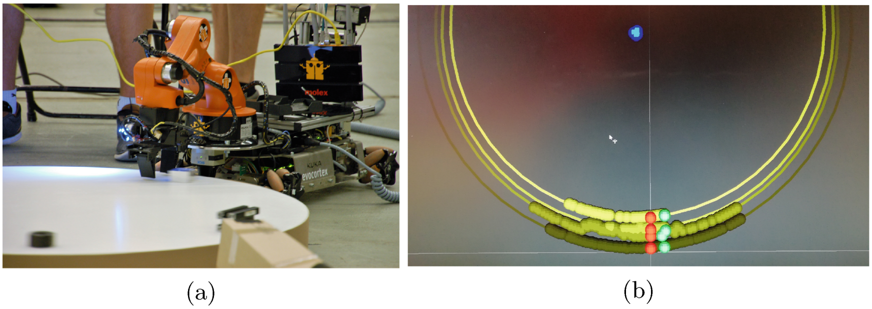
For 2018, the placement process for the shelf workstations was adapted to the changes in the rulebook. Placing an object below the shelf is higher rewarded than placing it on top, because its more likely to cause a collision with parts of the sensor head attached to the gripper. Therefore, a custom placement trajectory has been added to ensure safe operation in the enclosed space below the shelf. Additionally, the grasping process for moved objects was modified to enable more accurate timing and placement of the TCP.

The gripper controller consists of two separated nodes. The driver node runs a microcontroller program which is connected to the Dynamixel servo motors. It initializes and controls the motors position, torque and speed. The microcontroller is connected to the main PC via USB and offers an interface for motor controls and parameter settings. The gripper controller node runs on the main PC and offers dynamic reconfigure options and the grasping services used by the manipulation controller and other nodes. It uses the current torque applied to the motor to determine if an object has been grasped. The torque feedback is also used to prevent the motor from overcurrents by reducing the torque in case of high loads.

5.5 Rotating Turntable

As explained in Sect. 3 the rules for this challenge have been significantly changed in 2018. To adapt our system to the changes, the following algorithm considers various parameters such as the rotation speed, rotating direction and the pose of each object on the table.

The robot first navigates to the rotating turntable and extends the manipulator arm to an object detection position. Only performed once, an object recognition preprocessing approach is started to obtain the rotating table speed



(a) (b)

Fig. 13. Rotating Turn Table: Robot in front of the rotating turntable grasping an object (a) All data points, given by the object recognition, and the result of the determined circular paths of all objects on the turntable with different grasp points (red marked) (b). (Color figure online)

and the direction of rotation. First, the 2D position, the time stamp and the type of incoming objects into the camera visual field are recorded over a defined time. Second, the gathered data is used to determine objects circular paths, defining specific grasping position for each circular path. Figure 13b shows a result of this process determining four circular paths with four different grasping positions (red marked).

With the collected data points of each circular path, a RANSAC-based algorithm [2] calculates the rotation speed of the table, its center (blue marked in Fig. 13b) and the radius of each determined path. Having all necessary information and making use of the previously recorded time stamps, it is possible to estimate an approximate moment, when each object passes the object grasping position. To achieve an accurate grasping, an additional stereoscope RGB camera has been attached on top of the manipulator. A background change algorithm is now applied to the image in order to detect the object entrance in the camera view. The previously calculated circular path velocity is used to close the gripper at the right moment.

With the implemented feedback of the gripper the robot recognizes, whether grasping was successful or has failed. In case of success, the object is placed on the robot and the manipulator then moves over the next circular path to grasp the remaining objects. If the grasping fails, the manipulator stays in the position and waits one more time until the same object arrives at the RGB camera. If this retry fails again, the robot tries to grasp the next object on the rotating turntable. The smallest object of the @Work competition, a small nut, was successfully grasped demonstrated a remarkable accuracy.

6 Results

Table 2 presents the scorings of the world cup in Montreal 2018.

Table 2. Results of the RoboCup@Work world cup competition.

Place	Team	BMT	BTT1	BTT2	PPT	BTT3	RTT	Final	Total
1	AutonOHM	1043,75	900	975	550	1050	425	1375	6318,75
2	b-it-bots	675	800	900	450	450	225	1175	4675
3	MRL@work	525	550	850	0	875	0	1050	3850
4	RobOTTO	575	725	1050	0	0	25	1075	3450
5	LUHbots	675	475	0	37,5	1000	25	650	2862,5
6	RED	325	75	525	0	0	0	0	925
7	RoboErectus	225	0	200	50	350	25	0	850

7 Conclusion and Future Work

This paper described the participation of team AutonOHM in the RoboCup@ work league. It contains detailed information of the hardware setup and software packages like navigation, perception and manipulation. We believe that our system stability and repeatability are the key factors to achieve such a regular and high performance shown in Table 1.

To further increase the system stability and defend the RoboCup@Work champions title we introduced several improvements in different fields. First, the new gripper and its feedback function has improved the reliability for correctly grasping objects, specially during the RTT. Second, the object detection rate was increased to improve the decoy objects detection during the RTT.

Our main goal for 2019 is to develop a new robot platform to participate in the German open RoboCup@work, because our Youbot is getting unstable. Besides, there are several software modules which must be adapted to the new platform. Additionally, it is planned a software reorganization to make the different modules more modular, independent of each other and reusable.

References

1. Dellaert, F., Fox, D., Burgard, W., Thrun, S.: Monte Carlo localization for mobile robots. In: Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C), vol. 2, pp. 1322–1328, May 1999
2. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
3. Hochgeschwender, N., et al.: Work Rulebook (2017)
4. Kraetzschmar, G.K., et al.: RoboCup@Work: competing for the factory of the future. In: Bianchi, R.A.C., Akin, H.L., Ramamoorthy, S., Sugiura, K. (eds.) RoboCup 2014. LNCS (LNAI), vol. 8992, pp. 171–182. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18615-3_14
5. Martin, J., Engelhardt, H., Fink, T., Masannek, M., Scholz, T.: RoboCup@Work winners 2017 team AutonOHM. In: Akiyama, H., Obst, O., Sammut, C., Tonidandel, F. (eds.) RoboCup 2017. LNCS (LNAI), vol. 11175, pp. 498–508. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00308-1_41