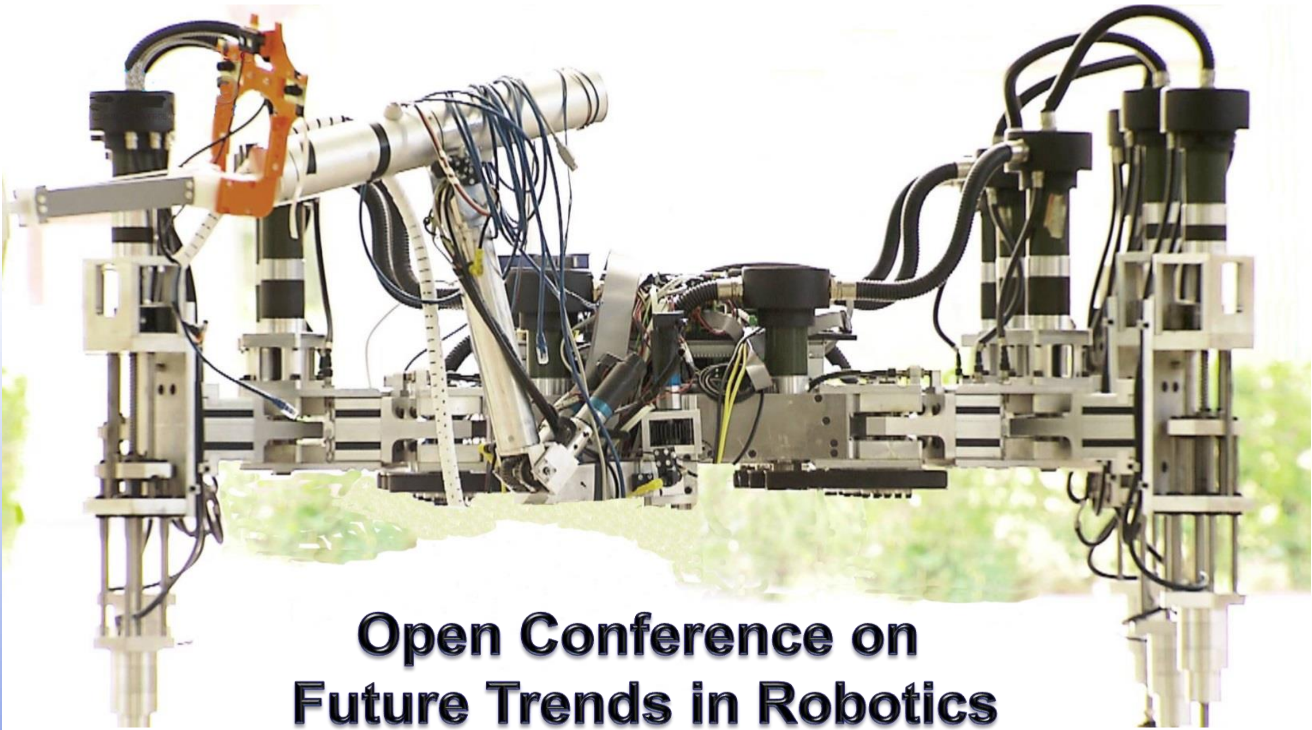
RoboCity16

Robots for citizens



Open Conference on Future Trends in Robotics

Edited by:

Roemi E. Fernández

Héctor Montes

RoboCity2030.org

Comunidad de Madrid

UNIÓN EUROPEA
Fondos Estructurales
Invertimos en su futuro

Universidad Carlos III de Madrid

CSIC
Consejo Superior de Investigaciones Científicas

Politécnica

Universidad de Alcalá

Universidad Rey Juan Carlos

UNED 40

May 2016

RoboCity16 Open Conference on Future Trends in Robotics

Editors
Roemi E. Fernández Saavedra
Héctor Montes Franceschi

Madrid, 26 May 2016

Edited by: Consejo Superior de Investigaciones Científicas
Printed by: AFERTA SG S.L.
Legal Deposit: M-17804-2016
ISBN: 978-84-608-8452-1

CHAPTER 44

GRABBING OBJECTS THROUGH A ROBOTIC ARM AND HAND IN A SAFETY WAY

A. LÁZARO, J. MORIANO, L.M. BERGASA, R. BAREA and E. LOPEZ

Robesafe group, Universidad de Alcalá, alberto2991lazaro@gmail.com.

This paper presents a system for grabbing objects in an industrial environment through a robotic arm and a hand grip in a safety way including three dimensional obstacle avoidance. The environment information is provided by a vision system, based on a RGBD camera, through the Robotic Operating System (ROS) (Quigley, 2009). A comparison between the different solvers is also carried out for a typical industrial scenario and experimental results with an IRB120 robotic arm are also shown.

1 Introduction

Industrial robotics systems tend to autonomously be able to complete more complex tasks with the minimal human intervention. In last years, taking advantage of the newest object recognition techniques in computer vision, it is possible to automatically classify the detected objects in different groups.

In a working space with several different objects, once the vision system has classified every object in the group it belongs, it is possible to physically classify and grab all of them by means of a robotic arm and a robotic hand. In the proposed work, the trajectory calculation for grabbing and placing the target object, comparing different solvers, is faced. These trajectories are calculated avoiding collisions with the rest of detected objects in a three-dimensional way.

The employed vision system using a RGBD camera is explained in (Lázaro, 2016) and it includes object classification, pose and dimension estimation. The communication between the vision and robotic modules has been implemented taking advantage of ROS communication facilities.

The main goal of this paper is, by reconstructing a scene with the provided information, in which the objects, the robot arm and its environment are included, be able to interact with the different objects placed in the working scene in a safety way, avoiding damages in the robot and in the manipulated objects. Different planner strategies have been analyzed and compared to select the one that better fits with the required needs.

2 State of the art

Regarding to the topic of automatic objects classification employing robotic arm, several projects have been carried out. Hereafter we show some of the most important.

In (Szabo, 2012), (Bdiwi, 2012) and (Li, 2014) object classification and sorting is faced depending on the object color or shape. They also employ industrial robots for grabbing the different objects and placing them. These works are mainly focus in computer vision algorithm for detection and classification of the different objects.

Our proposal, as difference with the above works, employs different techniques for robotic arm planning and they are compared to choose the one that best fits our requirements. The selected planning techniques in ROS are employed to develop the robot manage system and to carried out comparison results between the different OMPL (Open Motion Planning Library).

Furthermore, the trajectories are calculated at the computer employing a robotic CAD model. This way the system is easily portable for a different robotic arm. Using OMPL planners and incorporating the detected objects to the scene, the path is calculated as a non-collision path.

3 Set up and robot Control

All the communication processes have been implemented by means of ROS, employing different nodes and topics. The vision system is set up as a unique ROS node while the robotic system is separated in several ROS nodes. The communication is established through a common topic where the vision system is publishing continuously.

To carry out the trajectories planning, an URDF model which includes the IRB 120 robotic arm shape and its configuration parameters, was developed using the robot CAD models as it is depicted in Fig. 1. The robot parameters, that are set up in the URDF file, define the axes and the joints movement limits.

Once the model is completed it is introduced into the "MoveIt!" software, where the robot environment is added with the provided information (object class, position, orientation and dimensions), so trajectories plan can be faced.

Due to several objects can be touched by the robot when some of them is grabbed, in the grabbing maneuvers the rest of the objects will be defined as "collision objects", which cannot collide with any part of the robot. A safety area is added around the collision object in order to increase the security of the maneuver.

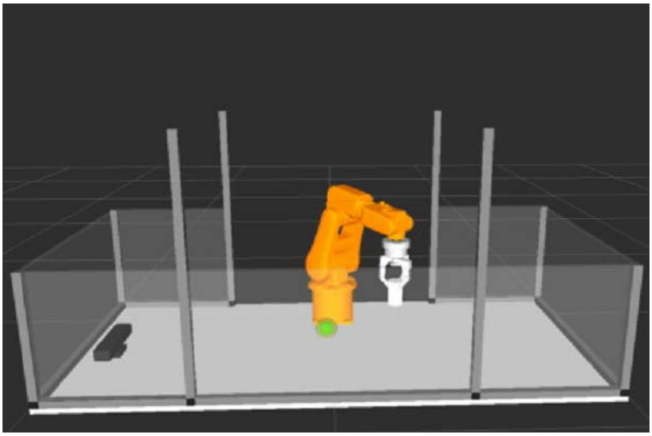


Fig. 1. Robot Model: the robot model and its working environment are shown. There are two different objects placed in the working scene: a can (cylinder) and a ball (sphere), which is depicted with a safety area around it

4 Robot Control

Once all the objects placed in the working scene are detected and the virtual environment is created, the robot recognizes the different objects according to the explained methods. The robot classifies the objects from the biggest to the smallest. Different grab processes have been developed to optimize the hand-grip taking into account the shape the objects have.
The robot uses the information of the vision system to estimate the height where the robotic hand should close.

After grabbing an object, it is placed in the assigned position according to the class it belongs as it can be seen in Fig. 2. The locations where the objects are deposited are filtered by the system, so once they are placed they disappear in the computer reconstruction.

Fig. 2. Object sorting: Different objects are shown once the robot has sorted them

4.1 Collision avoidance

Due to the fact that the number of objects placed in the working scene is usually bigger than one, a planner which is able to find free-collision-paths is required. Different motion based planners (PRM planners and Tree based planners) (Hsu, 1999), (Sanchez, 2003), (Suçan, 2009) and (Muja, 2009), which are available in OMPL are compared in our scenario.

The maximum distance among nodes is the most important parameter. The dimensions of the objects and the whole distance trajectory should be taken into account to choose it properly. This distance should reach a compromise between processing time and safety. After several tests, the maximum distance that returns better results in terms of performance is 5 cm, which is the reason why this distance is the one that has been established to estimate the trajectories.

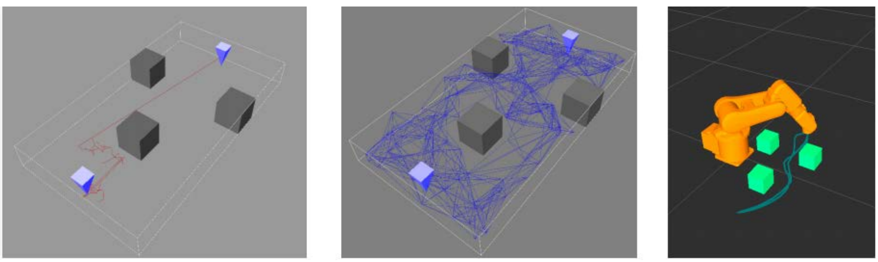


Fig. 3. In this figure the differences between probabilistic road-map and tree-based planning strategies are shown using KPIECE (left image) and PRM planner (center image). They are employed on an environment close to the real cases they will be faced. These

Simulations have been carried out thanks to OMPL application. In the right image the KPIECE planner is tested in the simulated environment and different paths were planned meeting the constraints and avoiding the three collision objects.

All the planners were tested solving the same case several times in which the robot needs to avoid three box obstacles (10x10x10 cm). The robot is forced to find a free three dimensional collision trajectory with a constrain of 10 cm in height. Average values of the obtained results are shown in Table 1 to choose the planner that has more interesting features. This analysis reveals that KPIECE and BKPIECE planners are the ones that stay closer to the desired requirements. Time costs are shown for the different tested planners in the next table:

Table 1. Planners times with 3 obstacles.

Planner	Average time (s)
BKPIECE	22.5
KPIECE	25.71
LBKPIECE	90
EST	90
PRM	45
PRM Star	180
RRT	60
RRT Connect	90
RRT Star	-
TRRT	-
SBL	36

In 3D environments, tree-based planners are really interesting because they can find a path without collisions in a small time, reducing the calculations. In the Fig. 3, the two planner categories (three-based and probabilistic) are compared applying them to the same problem.

KPIECE tree-based planner was selected as the better option because it is one of the fastest planners for our scenario and the paths are smooth enough to get our goals. Due to this planner carries out a discretization, it takes longer, compared with random ones, to define the better path.

4.2 Application Test

To test our whole application different objects such as balls, cans and boxes are used. The objects are detected by the vision system which provides to the robot the position where all of them are located so it can reach their position. According to the objects class they are grabbed and deposited by the robot in different places. The robotic system receives the path through an ROS–ABB socket.

Firstly, the robot moves to the initial position, which is high enough to allow the vision system captures properly the working scene. Afterwards the working scene simulation is updated according to the information it receives.

The robot goes to the location of the biggest object in the scene and grabs it setting the finger angles to the established position, then robot deposits the grabbed object in a predefined location according to the class it belongs. Finally, the robot goes back to the initial position out of the Kinect vision field and the process starts again in the same way. The working scene stops updating while the robot is in the Kinect vision field.

It should be delighted that through the feedback of robotic hand and arm states it is possible to get in real time a simulated view of the whole process thanks to RViz.

In Fig. 4 an example of the application is shown in simulation and with the real arm (IRB120 of ABB) and hand (BH8-262 of BarrettHand). In this figure it can be seen how the box is not included in the simulated scene due to the fact that it has been previously placed by the robot in its proper location employing a position filter the box is no longer included in the scene. Therefore, there are just two objects in the working scene: a can and a ball. The can is the target object for the robot in the shown case because it is the biggest one. Therefore, the ball is considered as collision object.

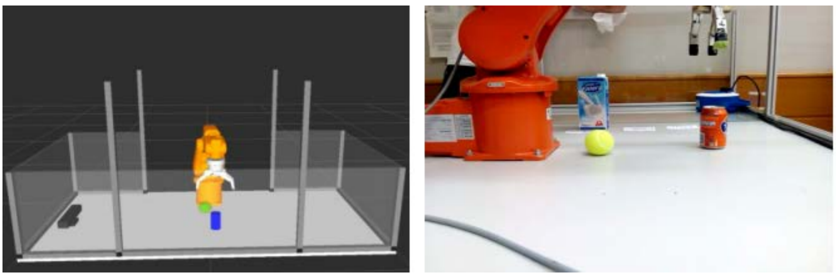


Fig. 4. Application example: The left figure shows the simulated working scene while the right one depicts the real working scene from the Kinect's point of view. Both images were taken at the same time.

5 Conclusions and future work

This paper has addressed the trajectory planning for object grabbing in a 3D free collision way, both in simulation and in a real industrial environment. To achieve the robot movement ensuring there is no collision an optimum movement planner, which is able to avoid hurdles, is used. Although it has been presented a real time solution for the addressed problem an enhancement will be done on working on reducing the processing time in order to make our system able to recognize and move objects faster than a human.

Acknowledgements

This work was supported in part by the MINECO SmartElderlyCar (TRA2015-70501-C2-1-R) project and by the RoboCity2030 III-CM project (S2013/MIT-2748) funded by Programas de I+D en la Comunidad de Madrid and cofunded by Structured Funds of the UE.

References

Bdiwi, M. 2012. Robot Control System with Integrated Vision / Force Feedback for Automated Sorting System. , pp.31–36.

Cruz, L. 2012. Kinect and RGBD images: Challenges and applications. Proceedings: 25th SIBGRAPI - Conference on Graphics, Patterns and Images Tutorials, SIBGRAPI-T 2012, pp.36–49.

Hsu, D. 1999. Path Planning in Expansive Configuration Spaces. International Journal of Computational Geometry & Applications, 09(04n05), pp.495–512.

Lázaro, A. 2016. 3D Object recognition and pose estimation using VFH descriptors. Open Conf. on Future Trends in Robotics (Robocity,'16').

Li, K. 2014. Robotic object manipulation with multilevel part-based model in RGB-D data. Proceedings - IEEE International Conference on Robotics and Automation, pp.3151–3156.

Muja, M. 2009. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. International Conference on Computer Vision Theory and Applications (VISAPP '09), pp.1–10.

Quigley, M. 2009. ROS: an open-source Robot Operating System. In ICRA workshop on open source software (Vol. 3, No. 3.2, p. 5).

Rusu, R.B. 2011. 3D is here: point cloud library. IEEE International Conference on Robotics and Automation, pp.1 – 4.

Sanchez. 2003. A single-query bi-directional probabilistic roadmap planner with lazy collision checking. Robotics Research, pp.403–417.

Şucan, I. A. 2009. Kinodynamic motion planning by interior-exterior cell exploration. In Algorithmic Foundation of Robotics VIII (pp. 449-464). Springer Berlin Heidelberg.

Szabo, R. 2012. Automated colored object sorting application for robotic arms. 2012 10th International Symposium on Electronics and Telecommunications, ISETC 2012 - Conference Proceedings, pp.95–98.