Intelligent flat-and-textureless object manipulation in Service Robots

Abel Pacheco-Ortega[1], Hugo Estrada[1], Edgar Vázquez[1], Reynaldo Martell[1], Jesús Hernández[1], Julio Cruz[1], Edgar Silva[1], Jesus Savage[1,*] and Luis Contreras[2]

***Abstract*— This work introduces our approach to the flat and textureless object grasping problem. In particular, we address the tableware and cutlery manipulation problem where a service robot has to clean up a table. Our solution integrates colour and 2D and 3D geometry information to describe objects, and this information is given to the robot action planner to find the best grasping trajectory depending on the object class. Furthermore, we use visual feedback as a verification step to determine if the grasping process has successfully occurred. We evaluate our approach in both an open and a standard service robot platform following the RoboCup@Home international tournament regulations.**

I. INTRODUCTION

One way to evaluate the state-of-the-art on object manipulation is through standard setups in international competitions such as Robocup@Home [1], whose purpose is the development of Domestic Service Robots, with a high relevance in the future of domestic and assistance applications. Since these agents move and interact in non-structured spaces and objects, the range of strategies, algorithms, and approaches that each challenge implies is considerable.

Grasping objects from a table is a common task for Service Robots. Robots mounted with active sensors such as RGB-D cameras are able to detect the dominant plane, cluster point clouds on the plane and analyze them in order to detect the objects on it and determine the best strategy to manipulate the objects as in [2],[3] and [4]. However, when the objects lack texture and or volume (i.e. flat objects), the manipulation task becomes more challenging.

In the 2018 edition of Robocup@Home, one of the tests was the Procter & Gamble (TM) Dishwasher Challenge (PGDC) in which they have to remove all objects from a table and place them into a dishwasher [5]. The set of objects that were used during the test are made of plastic, without any visual pattern on them but colours with high contrast, as shown in Figure 1a. Besides, the objects' surface is very reflective provoking a significant variation in colour. In consequence, the information sensed by the RGB-D camera results noisy and the objects seem to belong to the table, as shown in Figure 1b.

In this work, we present our strategy to successfully manipulate the objects in the dishwasher challenge. In the next sections, we will detail the techniques used to detect, recognize and manipulate tableware and cutlery objects.

The authors are with the [1]Faculty of Engineering, National Autonomous University of Mexico and the [2]Advance Intelligence and Robotics Research Center at Tamagawa University, Japan.

This work was supported by PAPIIT-DGAPA UNAM under Grant IG-100818

[*]savage at servidor.unam.mx

(a)

(b)

Fig. 1: a) Tableware and cutlery objects used in the RoboCup@Home's P&G Challenge and b) table setup 3D view from an RGBD camera. It can be observed that the reconstructed scene presents some errors due to sensor noise, varying illumination conditions and high object reflectance.

II. OBJECT DETECTION AND MANIPULATION

Typical object recognition strategies consist on object segmentation by plane extraction and point cloud features clustering ([6], [7], [8], [9]) or object detection in the whole image by feature matching ([10], [11], [12],[13]). However, in situations where the objects are flat, plane extraction is no possible; similarly, when textureless objects are present, feature detection and matching become unavailable and other approaches need to be explored.

Another popular approach to solve this problem is the use of Deep Neural Networks, in particular the YOLO [14] architecture provides state of the art performance on the object detection task. Our approach does not rely on neural networks and thus we do not rely on specialized hardware such as GPU's or online servers. Also, we need much less training examples compared to YOLO, which needs a couple hundreds of images per object; in the same sense, YOLO requires labelled examples from different poses and in different backgrounds in order to generalize correctly, our approach uses just a couple images from the same pose in the same background.

In this section, we present our solution to the flat and textureless object detection and recognition problem. We first use colour features in the detection process and then extract the geometry properties using the point cloud to describe them. This information also results useful in the manipulation process.

For background subtraction, we find the range of values in each channel on three different colour spaces (RGB, HSV and HLS) for all objects and a raw segmentation mask is obtained where morphological operations of closing and dilation are applied to close the gaps where segmentation was wrongTo deal with cases where objects present significant illumination changes, a convex hull operation is applied as the final step in the segmentation process. Finally, we obtain an RGB mask and a 2D bounding box per object, as illustrated in Figure 2.

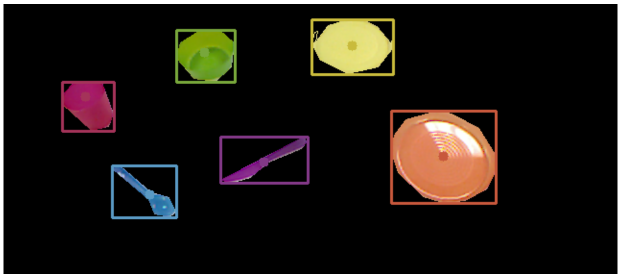

Fig. 2: Typical output after object detection using colour features.

We then categorise objects into four different classes, namely glass, plate, bowl, and cutlery, by normalising the image with respect to the average depth and then extracting the area in the RGB image and the number of points and the eigenvectors from the point cloud, as will be described next.

We consider the differences in the area among the objects – for example, the glass-type object, despite having greater height, its area is not very large due to its transparency –, that can provide an initial hint on the object class. In a training step, from a set of segmented images, a valid area range per class is obtained to determine if it is a given category in case the object's area is within a valid range.

However, due to changes in illumination, the visible area may change and therefore we also consider the 3D point count per object. From the same point of view, the dimensions of the cutlery and glass differ greatly from that of the plate or bowl, hence, we set a valid threshold among those classes, so if an object is within this range it is said that it can be in the cutlery and glass sub-category, and in the plate and bowl sub-category otherwise.

To correctly classify objects in the sub-categories, we consider their geometry properties as follows. From an object's point cloud of 3D features, we extract their eigenvector and eigenvalues using principal component analysis (PCA) and align them to the dominant plane coordinate system (plane XY and normal Z), as can be seen in Figure 3. For the cutlery and glass sub-category, we assign the cutlery class to any object with a small height and a small ratio of the shortest XY axis to the largest one, due to the flat and elongated shape, and glass class otherwise. In the case of the dish and bowl sub-category, we only consider the height component due to the similarity on the shape in the XY plane between a plate (flat, small Z) and bowl (deep, large Z). An example of the classification result is shown in Figure 4.

After testing 1500 times the algorithm under different conditions and using different object configurations we obtained the results presented in the following confusion matrix, where a high recognition rate is observed.

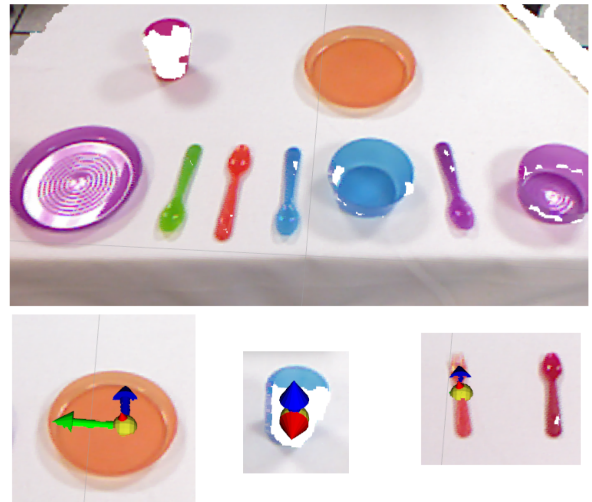

Fig. 3: Eigenvectors for a group of common cutlery used in P&G's test. Blue colour shows the eigenvector with bigger magnitude. Eigenvectors were obtained with PCA.

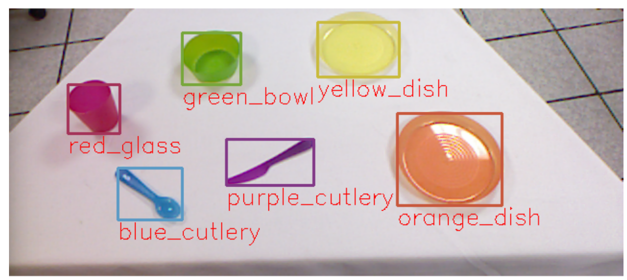

Fig. 4: Classification result using 3D information to the setup given in Figure 2.

Predicted Class

Actual Class	Glass	Dish	Bowl	Cutlery
Glass	93%	2%	3%	2%
Dish	2%	90%	6%	2%
Bowl	3%	5%	91%	1%
Cutlery	2%	1%	2%	95%

III. Interaction with Real World

We evaluated our approach in two different platforms (both open and standard), namely robot Justina – a domestic service robot developed at the Bio-robotics Laboratory in the

National Autonomous University of Mexico [15], [16] and the Toyota Human Support Robot (HSR) – a service robot developed by Toyota Company [17] –, and we use the Robot Operating System (ROS) [18]. This implementation allowed us to be one of the two teams with highest score in the standard category.

In our implementation, after the objects on the table are detected and recognised, this information is sent through ROS messages to the action planner module. The message contains information about each object, such as: a) related to the point cloud: eigenvalues, eigenvector, size, nearest point and centre point in robot coordinates, b) related to the image: position of top left corner, width and height of the bounding box, c) related to the object: centroid, roll, pitch and yaw manipulation angles, and a priority constant (this variable is set given the grasping ability of a given object with the robot hand – the priority, in increasing order, is as follows: dish, cutlery, bowl and glass).

To decide which object to take first, two measures are taken into consideration: the robot-to-object distance and the manipulation priority constant, both measures are combined to decide which object to take first, through manipulation process described below.

A. Object Manipulation

The robot end-effector roll, pitch and yaw angles are determined according to the object class to be manipulated, as shown in Figure 5. Considering a table plane XY with normal Z, in the case of glasses, the manipulator moves parallel to the plane XY towards the object's centroid, i.e. its yaw angle is 90 degrees and the other two angles are zero. For plates and bowls, the best way to handle them is by taking them from above in any point on the edge, so the end-effector moves in the Z axis in the direction of a selected point on the object's edge, i.e. its three angles are zero. Finally, to manipulate cutlery, the robot manipulator grasps them from above, where the hand moves in the Z axis centred at the cutlery centroid, i.e. pitch and yaw angles are zero, and the manipulator's roll angle is parallel to the direction of the cutlery's principal component, according to the Equation (1).

$$roll = atan2(e_y, e_x) \tag{1}$$

where e_x is the X coordinate and e_y is the Y coordinate of the principal eigenvector in robot hand coordinates.

The object manipulation process is as follows. First, the robot moves its base such that the arm is located just in front of the object to be manipulated. Then, from the inverse kinematics models, we calculate the arm joint angles to reach the goal position as well as the end-effector orientation, according to the object class to be manipulated, as described above.

Different cases are handled in the action planer regarding the end-effector trajectories depending on the object to be manipulated. In the case of glasses and cups, as the robot hand moves on the table plane and trajectories are drawn in a straight line from the robot manipulator to the object centroid, we place the hand at a safe height such as it does not collide with the table but is still able to reach the object. For the other objects, the robot hand moves in the Z axis (normal to the table plane) so the manipulator is placed at a safe distance over a target object and slowly decreased until it reaches the goal position.

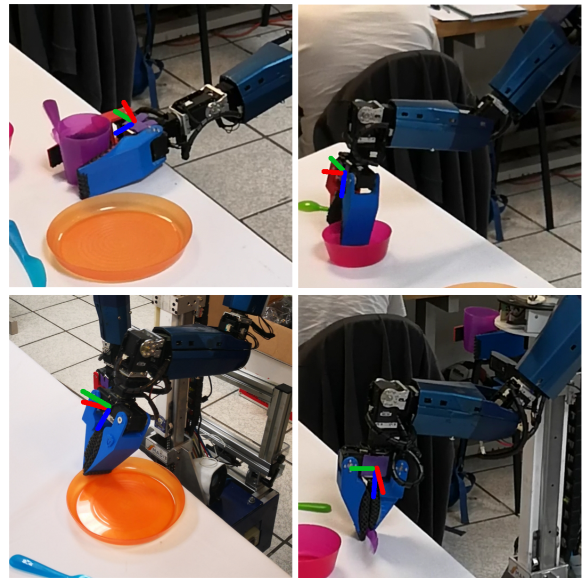

Fig. 5: Gripper orientation and reference system for different objects, top left to bottom right, cup, bowl, plate, and cutlery. The red, green and blue arrows are the manipulator X, Y, and Z axes, respectively.

B. Visual Feedback

For conventional objects such as cans, bottles, or small boxes, our robot uses as feedback the torque exerted by the motors on the object to know if the object was taken successfully. However, due to the size and shape of the objects in this test, we use a different feedback strategy based on the visual information to verify whether the object was taken.

As a result of the recognition process, we know the object's class, colour, centroid, and bounding box, and we use this information to validate that the object has been taken by finding whether the object is still present on the surface where it should have been removed (for example a plate on the table). We look for the closest objects to the target centroid on the table after the manipulation process and, if an object with the same class and colour is present within a valid distance to the target centroid, we infer that the grasping process failed and start it again.

IV. RESULTS IN COMPETENCE

In the PGDC test, robot "Justina" obtained a good performance in the manipulation tasks, as we can see in TABLE I where the robot was able to grasp 3 out of 4 objects that the robot attempted to manipulate, two of them in the first attempt, and the other in the second attempt after visual feedback confirmation was received. Similarly, robot ”Takeshi” managed to grasp 2 out of 3 objects that it tried to manipulate, as shown in TABLE II.

TABLE I: Robot Justina's manipulation results.

Type	Successful Graspping	Attempts to grasp
glass	yes	1
dish	yes	1
cutlery	yes	2
cutlery	no	3

TABLE II: Robot Takeshi's manipulation results.

Type	Successful Graspping	Attempts to grasp
bowl	yes	1
dish	no	1
dish	yes	1

It is important to notice that the manipulator in robot ”Takeshi” (Toyota HSR, as in Figure 6) is more precise at the hardware level than the one in robot ”Justina”, but visual feedback helps to improve the performance in both cases to the point to have similar outcomes, a result that shows the relevance of using intelligent manipulation strategies regardless of the hardware architecture.

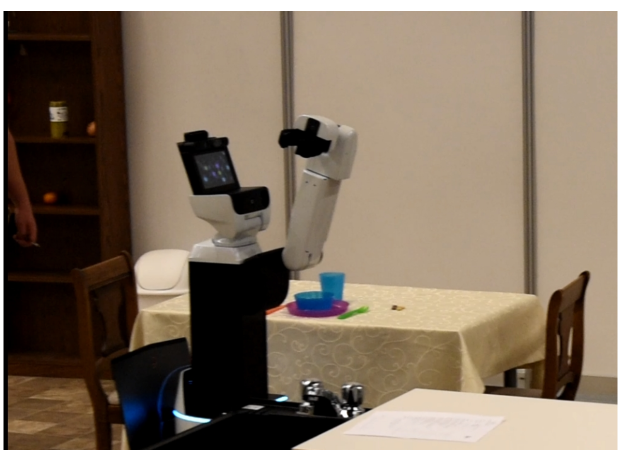

Fig. 6: Robot Takeshi (Toyota HSR) at the RoboCup@Home P&G Dishwasher Challenge. The robot moves from a known position and aims at cleaning the table.

V. CONCLUSIONS

We presented our approach to the flat and textureless object manipulation problem. We use colour and geometry cues from RGBD sensors, and define different intelligent grasping strategies depending on the object class. Furthermore, we use a visual feedback to determine whether the grasping process has been successful. The described strategies were used in the RoboCup@home league competition under an open and standard platform, having one of the best performances. For future work, we aim at using active feedback models to update in real time the manipulator position with respect to the target object and the grasping success/failure belief.

REFERENCES

[1] T. Wisspeintner, T. Van Der Zant, L. Iocchi, and S. Schiffer, “Robocup@ home: Scientific competition and benchmarking for domestic service robots,” *Interaction Studies*, vol. 10, no. 3, pp. 392–426, 2009.

[2] D. Rao, Q. V. Le, T. Phoka, M. Quigley, A. Sudsang, and A. Y. Ng, “Grasping novel objects with depth segmentation,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2578–2585, Oct 2010.

[3] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, “Real-time plane segmentation using rgb-d cameras,” in *RoboCup 2011: Robot Soccer World Cup XV* (T. Röfer, N. M. Mayer, J. Savage, and U. Saranlı, eds.), (Berlin, Heidelberg), pp. 306–317, Springer Berlin Heidelberg, 2012.

[4] A. J. R. Neves, R. Garcia, P. Dias, and A. Trifan, “Object detection based on plane segmentation and features matching for a service robot,” *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 10, no. 4, pp. 775 – 782, 2016.

[5] M. Matamoros, C. Rascon, J. Hart, D. Holz, and L. van Beek, “Robocup@home 2018: Rules and regulations.” `http://www.robocupathome.org/rules/2018_rulebook.pdf`, 2018.

[6] A. Richtsfeld, T. Mörwald, J. Prankl, M. Zillich, and M. Vincze, “Segmentation of unknown objects in indoor environments,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4791–4796, Oct 2012.

[7] A. Ückermann, C. Elbrechter, R. Haschke, and H. Ritter, “3d scene segmentation for autonomous robot grasping,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1734–1740, Oct 2012.

[8] Y. Chen, Y. Wang, J. Hu, and R. Xiong, “Picking from clutter: An object segmentation method for robot grasping,” in *Cognitive Systems and Signal Processing* (F. Sun, H. Liu, and D. Hu, eds.), (Singapore), pp. 341–354, Springer Singapore, 2017.

[9] S. Christoph Stein, M. Schoeler, J. Papon, and F. Worgotter, “Object partitioning using local convexity,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[10] N. Ikizler-Cinbis and S. Sclaroff, “Object recognition and localization via spatial instance embedding,” in *2010 20th International Conference on Pattern Recognition*, pp. 452–455, Aug 2010.

[11] F. Alhwarin, C. Wang, D. Ristic-Durrant, and A. Gräser, “Improved sift-features matching for object recognition,” in *Visions of Computer Science - BCS International Academic Conference, Imperial College, London, UK, 22-24 September 2008*, pp. 178–190, 2008.

[12] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, pp. 91–110, Nov 2004.

[13] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features,” in *Computer Vision – ECCV 2010* (K. Daniilidis, P. Maragos, and N. Paragios, eds.), (Berlin, Heidelberg), pp. 778–792, Springer Berlin Heidelberg, 2010.

[14] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv*, 2018.

[15] M. Matamoros and J. Savage, “A blackboard based implementation of a mobile robot architecture,” in *International Conference on Advanced Robotics 2013 (Robocup @Home Worksop*, 2013.

[16] J. Savage, M. Negrete, M. Matamoros, J. Cruz, R. Lagunas, J. Marquez, J. Marentes, and F. Villaseñor, “The role of robotics competitions for the development of service robots,” in *IJCAI'16, Workshop on Autonomous Mobile Service Robots*, 2016.

[17] K. Hashimoto, F. Saito, T. Yamamoto, and K. Ikeda, “A field study of the human support robot in the home environment,” in *2013 IEEE Workshop on Advanced Robotics and its Social Impacts*, pp. 143–150, 2013.

[18] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *ICRA workshop on open source software*, vol. 3, p. 5, 2009.