The research leading to these results has received funding from the RoboCity2030-III-CM project (Robótica aplicada a la mejora de la calidad de vida de los ciudadanos, fase III; S2013/MIT-2748), funded by Programas de Actividades I+D en la Comunidad de Madrid and cofunded by Structural Funds of the EU.
RoboCity16 Open Conference on
Future Trends in Robotics

Editors
Roemi E. Fernández Saavedra
Héctor Montes Franceschi

Madrid, 26 May 2016
This paper presents the perception system involved in an application for
grabbing objects in an industrial environment through a robotic arm by us-
ing 3D pointcloud images taken from a RGBD camera. This system recog-
nizes and estimates the pose of the objects placed on the working area. It
uses VFH descriptors and Kd-tree classifiers for the recognition and geo-
metrical models and a RANSAC estimator for the pose estimation using
this information. Some experimental results in a real environment validate
our proposal.

1 Introduction

This work is focused on object recognition and pose estimation in industri-
al environments using a RGBD camera. This vision system is able to de-
tect objects, recognize them and estimate their position offering the possi-
bility of detecting important features which can be used to automate
processes. This target can be achieved using RGB cameras but they have
some drawbacks faced with depth cameras which will be explained hereaft-
er. The stages that have to be completed by the vision system to provide
the necessary data are segmentation, recognition, pose estimation and sce-
ne reconstruction. Segmentation identifies the interested objects to be
grabbed. Once these objects are segmented they pass to the recognition al-
gorithm. Shape estimator is used to recognize and obtain the position and
orientation of the objects. Taking advantage of ROS communication facili-
ties, the obtained data are published to be reconstructed thanks to RVIZ as
it can be seen in Fig. 1.
2 State of the art

Regarding to the topic of automatic objects classification, several projects have been carried out. Hereafter we show some of the most important. In (Bdiwi, 2012) a more complex classification process is developed, in this case the system is able to difference among books depending on their shapes and using alphanumeric codes. They use a CCD camera, it uses a rectangle to compare the different shapes. Another example for the object classification is included in (Li, 2014) using a depth sensor. In this work, a multilevel part-based object model was proposed using latent support vector machine as a core learning machine for training.

Our proposal, as difference with the above works, uses a depth camera for obtaining 3D Point Cloud images (Cruz, 2012), (Rusu, 2011) and then to recognize the shape of the different objects. It provides several advantages, the main ones are the use of an RGB-D camera and a better 3D reconstruction of the scene.

3 Object recognition method

The vision system must detect the objects placed in the working scene. So 3D pointcloud is clustered in foreground and background classes using depth. After that, foreground class is downsampled using Uniform Sampling algorithm named VoxelGrid filter before being clustered in objects.

The aim of clustering the foreground data is to divide an unorganized PCL into smaller parts, which contain the points that belongs to a same object (Muja, 2009). This clustering has been done relying on spatial decomposition techniques that find boundaries and subdivisions. The algorithm used is based on Euclidean distances.
Fig. 2. Background extraction example: in the left image the original capture.

Fig. 3. Downsampling the captured scene example: the PCL depicted on the right.

Fig. 4. Cluster extraction diagram: the algorithm works out the distance between two points, depending on this value the evaluated points are considered as part of the same cluster or as part of two different clusters according to the diagram.

Once the scene is clustered, is time to tackle the object recognition of the different objects placed in the scene (Rusu, 2010; Rusu, 2009). To carry out this recognition VFH descriptors, which is based on the FPFH descriptor, are used (Rusu, 2009; Rusu, 2008). This recognition consists in two processes:
3.1 Training process

The first step is to build a dataset that contains some captures (36 in our case) taken from different points of view for each object that is wanted to be recognized by the system and to calculate their VFH descriptor. This dataset is used to train the object recognition system. The training process builds a space-partitioning data structure that stores a set of K-dimensional points in a tree structure that enables efficient range searches and nearest neighbor searches called Kd-tree (Rusu, 2008), (Salti, 2011).

3.2 Testing process

This process compares VFH descriptors for each of the segmented objects extracted from the captured scene and the ones that are saved in the training dataset through K-Nearest neighbors algorithm (k-NN) (Muja, 2011). The result of this comparison is a set of index which values are inversely proportional to the likeness between the analyzed cluster and each element of the dataset (Bariya, 2010). These index are provided to a voting method to determine the class among the trained classes which belongs to the unknown object if the analyzed object belongs to one of the trained classes. If so, the voting system classifies it.
4 Pose estimation method

Once each of the elements placed in the working scene are recognized, the coefficients that determine their position and orientation are recovered by using RANdom SAmple Consensus (RANSAC). RANSAC algorithm is usually employed to carry out in objects recognition approaches when the set of objects to be distinguished have different shapes (Schnabel, 2007), (Papazov, 2010), but in this work it has been used just for estimating the pose of the objects and their dimensions.

Three models (sphere, cylinder, plane) jointly with their RANSAC estimators are used to estimate the parameters that defines the pose and the dimensions of the different objects (balls, cans, milk cartons) (Aldoma, 2012), (Mishra, 2011). To estimate the dimensions of the carton box once the different planes have been estimated geometric constraints are applied.

5 Perception system experiments

The following tests have been carried out to measure the power of the designed vision system evaluating different aspects such as time costs or percentage success. Hereafter we present the different tests carried out to evaluate our vision system performance regarding the processing time an object recognition results. The first experiment tests the performance of our objects recognition system for 5 different objects of each class not
used in the training process. As it can be seen in Table 1 the correct recognition ratio is close to 100%.

Table 1. Object recognition results: index shown are the returned by the system.

<table>
<thead>
<tr>
<th>Object/Candidate</th>
<th>Ball</th>
<th>Can</th>
<th>Milk carton</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ball</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Can</td>
<td>1%</td>
<td>95%</td>
<td>4%</td>
</tr>
<tr>
<td>Milk carton</td>
<td>1%</td>
<td>5%</td>
<td>94%</td>
</tr>
</tbody>
</table>

The object recognition system runs in a computer with Intel Core i7 CPU 860 @ 2.8GHz x8 processor. Table 2 evaluates the different processing time depending on the recognized elements.

Table 2. Object recognition processing times.

<table>
<thead>
<tr>
<th>Object</th>
<th>No. balls</th>
<th>No. cans</th>
<th>No. Cartons</th>
<th>Time costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>190ms</td>
</tr>
<tr>
<td>Test 2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>275ms</td>
</tr>
<tr>
<td>Test 3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>410ms</td>
</tr>
<tr>
<td>Test 4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>345ms</td>
</tr>
<tr>
<td>Test 5</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>650ms</td>
</tr>
<tr>
<td>Test 6</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>890ms</td>
</tr>
<tr>
<td>Test 7</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>515ms</td>
</tr>
<tr>
<td>Test 8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1345ms</td>
</tr>
</tbody>
</table>

Fig. 7 shows the results extracted from all tests to determine the dimension, position and orientation errors through the mean and the deviation for the different primitive shapes measured in meters.

6 Conclusions and future work

This paper has addressed the object recognition, pose estimation and dimensions appraisal. Due to the real time constraints geometric restrictions, VFH descriptors and a Kd-tree classifier have been applied over the 3D point cloud images. Object pose estimations are precise enough for a robot to interact with them. The vision system algorithm could provide a robotic system the information needed to move or avoid objects. Although we have presented a real time solution for the addressed problem we will go
on working on reducing the processing time in order to make our system able to recognize 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., & Suchý, J. 2012. Robot control system with integrated vision/force feedback for automated sorting system. In Tech. for Practical Robot Applications (TePRA), 2012 IEEE Int. Conf. on (pp. 31-36). IEEE.


fast, robust, scalable recognition infrastructure. In Robotics and Au-
atomation (ICRA), 2011 IEEE Intern. Conference on (pp. 2939-2946). IEEE.

recognition in noisy and occluded scenes. In Computer Vision–ACCV 
2010 (pp. 135-148). Springer Berlin Heidelberg.

Rusu, R. B., Blodow, N., & Beetz, M. 2009. Fast point feature histograms 
IEEE International Conference on (pp. 3212-3217). IEEE.

and pose using the viewpoint feature histogram. In Intelligent Robots and 
Systems (IROS), 2010 IEEE/RSJ Intern. Conf. on (pp. 2155-2162). IEEE.

Rusu, R. B., Holzbach, A., Beetz, M., & Bradski, G. 2009. Detecting and 
segmenting objects for mobile manipulation. In Computer Vision Work-
shops (ICCV WS), 2009 IEEE 12th Int. Conf. on (pp. 47-54). IEEE.

Rusu, R. B., Marton, Z. C., Blodow, N., & Beetz, M. 2008. Learning in-
formative point classes for the acquisition of object model maps. In Con-
trol, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th Interna-
tional Conference on (pp. 643-650). IEEE.

feature histograms for 3D point clouds. In Proc 10th Int Conf Intel Auton-
omous Syst (IAS-10), Baden-Baden, Germany (pp. 119-128).

Rusu, Radu Bogdan and Cousins, Steve. 2011. 3D is here: Point Cloud Li-
brary (PCL). ICRA.

Salti, S., Tombari, F., & Stefano, L. D. 2011. A performance evaluation of 
3d keypoint detectors. In 3D Imaging, Modeling, Processing, Visualization 
and Transmission (3DIMPVT), 2011 Intern. Conf. on (pp. 236-243). IEEE.

point-cloud shape detection. In Computer graphics forum (Vol. 26, No. 2, 