

ERASMUS MUNDUS MASTER IN VISION AND ROBOTICS

B31XP:ROBOTICS PROJECT

Office Object Recognition

Los Lobos Solitarios v2.0

Abraham AYALA

Quim SANCHEZ

Waldemar FRANCZAK

Supervisor:

Dr. Zeyn SAIGOL

December 11, 2013

CONTENTS

1 Introduction	2
2 Problem description and Previous work	2
3 Project Planning	3
4 Tools selection and evaluation	3
4.1 Project Reconsideration	3
5 3D mapping	3
5.1 CCNY visaul odometry	3
5.2 Point cloud registration using PCL	3
5.3 Octomap 3D mapping	4
6 Plane Segmentation	5
6.1 Algorithm outline	5
6.1.1 Normals Computations	6
6.1.2 Normals Clustering	6
6.1.3 Parallel planes distinction	6
7 Point Cloud Clustering	7
8 Object Recognition	7
9 Future Work	7
10 Conclusions	7

1 INTRODUCTION

During this report it will be described the advances done by Lobos Solitarios v 2.0 team in the Office Object Recognition project. The general goal of the project is to implement a pipeline which provides the ability to a mobile robot of performing object recognition of commonly present objects in an office environment. This information will give some insight to the robot about its environment, and will help it to perform more advanced and useful tasks. This work is organized as follows, in section 2 the problem description and previous work will be explained. In sections 3 and 4 is documented the project planning and learning process to propose a solution. In sections 5,6,7 and 8 is explained the proposed solution of the team in this starting point of the project. Finally in section 9 and 10 is settled the future work to be done and the conclusions about the work done.

2 PROBLEM DESCRIPTION AND PREVIOUS WORK

Preferably the system should be able to recognize the common office objects even though the object is partially occluded or only part of its components are seen, to achieve this a possible solution is to define object classes which are composed by different elements. For example a chair is composed by different elements as shown in figure 1, in some situations is possible that the robot only observes one or several of this subcomponents of the class chair. The system should be able to recognize or give probabilities of how likely is a set of structural elements to form an object class.

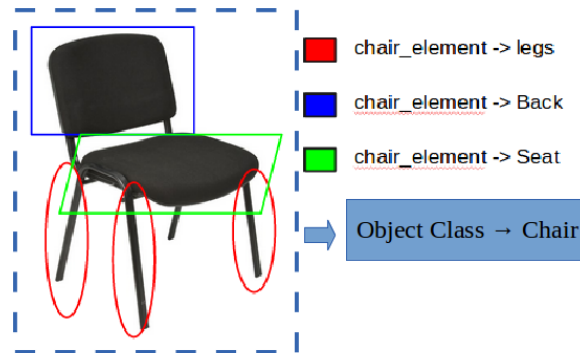


Figure 1: Decomposition of the class 'chair' into its structural elements

In a more advanced phase of the project, is intended to provide the robot the ability to identify when an object is incomplete, this with the purpose of moving the robot to the unknown part of the object and fuse the information of different views. The whole project of object recognition is intended to be realized by several VIBOT generations, and for the case of this semester the goal is to propose a starting point for the problem.

The available hardware platform for this starting point is:

- Turtlebot 2 mobile robot from willow garage and clearpath
- Microsoft Kinect (640x480) RGB-D camera integrated in turtlebot

In terms of software framework, the desired or preferable platforms to develop the project are:

- ROS (Robot Operating System) from willow garage
- ORK (Object Recognition Kitchen)[2]

One important requirement to follow is the integration of the proposed pipeline with ROS, since turtlebot robot is optimized to be controlled by ROS. About previous work realized, in [1] is described a general approach using ORK pipeline, manual labelling and 3D data registration to train different types of office objects to fuse and segment them in different views. A initial requirement of the supervisor of this project is to try to solve the application using of-the-shelf algorithms of OpenCV [3] provided by the object recognition framework, following the work done in [1].

3 PROJECT PLANNING

The initial gant diagram of the project was planned to follow the milestones shown in figure 2, in which the first two weeks were intended to determine if the ORK framework could provide the necessary software tools to develop the application.

Milestone / week #	3	4	5	6	7	8	9	10	11	12	13
Explore ORK functions											
Research and develop algorithms											
Implement algorithms in turtlebot											
Write Report											

Figure 2: Lobos Solitarios gant diagram

4 TOOLS SELECTION AND EVALUATION

Explanation of ORK and the work done in ORK, justification on why was not used the ORK and why PCL is a useful solution

4.1 PROJECT RECONSIDERATION

Description of the new planning and the splitting of the tasks between the team members,as well the general pipeline to follow to achieve a first approach to the problem with the pcl tools

5 3D MAPPING

As explained in section 2, and the new requirement of the supervisor to perform the 3D mapping in this first part of the project, The robot should be able to build a 3D map of the places it has visited, for later use to detect the objects and identify their position in the map. The next possible solutions to register point clouds or build the 3D representation of the readings of the robot were analysed:

- CCNY visual odometry and 3D map building
- PCL point cloud registration
- octomap 3D mapping

5.1 CCNY VISAUL ODOMETRY

Explain about this method

5.2 POINT CLOUD REGISTRATION USING PCL

Explain

The *pcl* library has some methods to perform point cloud registration, which are iterative process that aligned two given point clouds.

5.3 OCTOMAP 3D MAPPING

Based in octrees, several depths or resolutions can be queried Each node saves the probability to be occupied or free (helps to handle noise) Map build incrementally , known and unknown areas available.

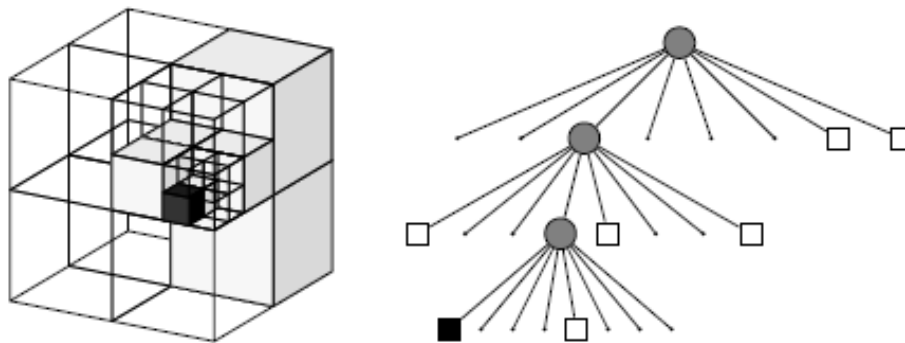


Figure 3: Octree representation

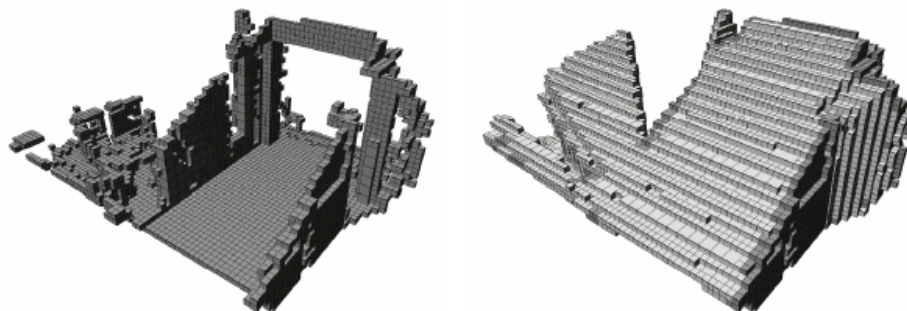


Figure 4: In the left the representation of occupied voxels, in right the representation of free and occupied voxels

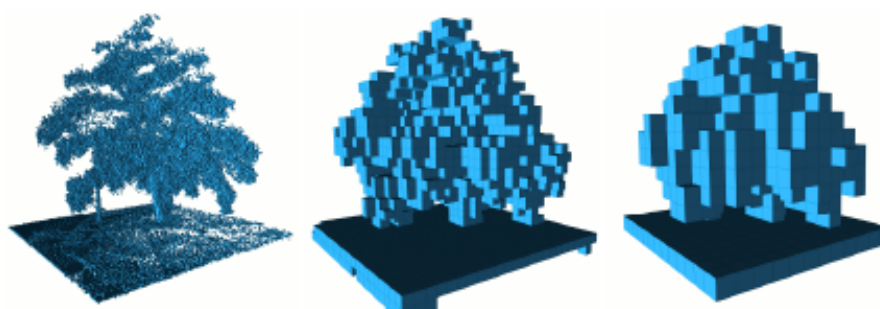


Figure 5: Different resolutions of the octomap

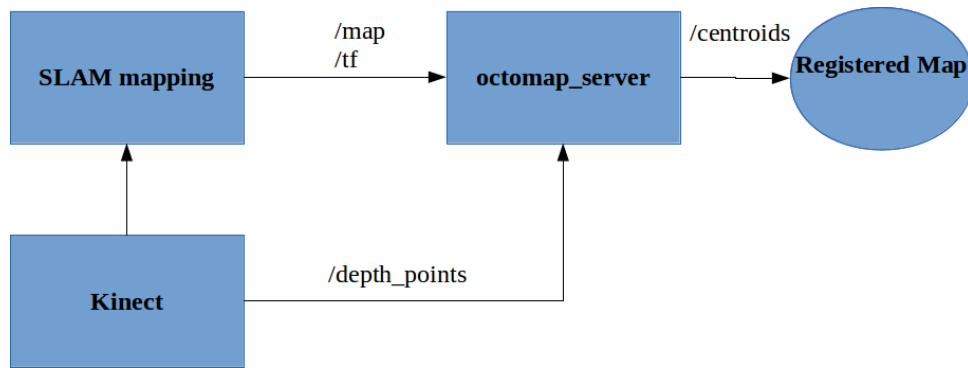


Figure 6: Octomap integration with ROS

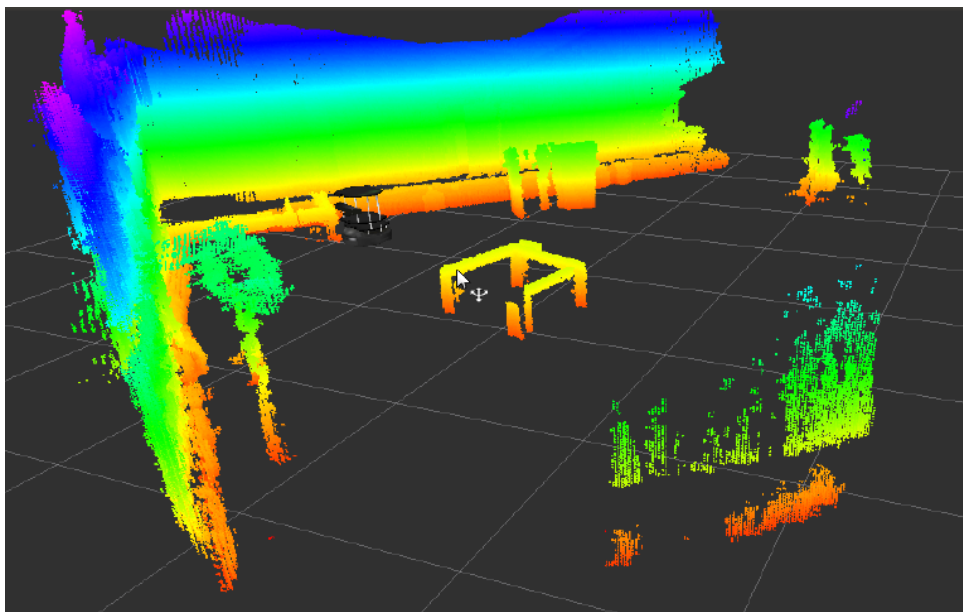


Figure 7: Octomap 3D mapping of a scene

6 PLANE SEGMENTATION

To make the search of primitives easier we first decided to segment the data following some rule that would make smaller set but at the same time would not split the objects. After doing some literature review we found this paper where they propose a method to detect and segment planes in a fast way. The main contribution from this paper is the idea of using the range image neighbourhood information to approximate the real point neighbour. Then they propose some simple ways to cluster the data as well.

6.1 ALGORITHM OUTLINE

The algorithm consists of 3 main parts:

- Normals computation
- Normals clustering
- Parallel planes distinction

6.1.1 NORMALS COMPUTATIONS

One way to represent planes is by its normal and one point belonging to the plane. To conclude it two points belong to the same plane we need to compute the Normal vector in each of the points. To compute the Normal of one point the information of its neighbours is needed and so the first problems is, how can we find the neighbour of a point? One approach is to use the closest points using KNN and some fast implementation. Once this points are found a plane can be fit and from it calculate its normal.

The problem of this approach is that the search of neighbours can be very computationally expensive even when using fast algorithm that only approximate the result. Also, if many points are fit to the plane, the computation of the normal vector using eigen decomposition can also be expensive.

In the proposed method both steps are substituted by much simpler and faster operations.

First of all the algorithm assumes that two pixels in the range image will still be neighbours once they are projected in the 3D space. This will not hold on the edges of planes but in general it can be accepted. At this step the algorithm computes two tangential vectors of the point using the difference between the points of right and left pixel for the horizontal and above and below pixel for the vertical. Then it computes the cross product of this two tangential vectors to compute the normal.

6.1.2 NORMALS CLUSTERING

After on normal is computed for each of the points a second step comes which will cluster similar normals with the idea that they will belong to the same plane. To do the clustering different iterative algorithm like k-means could be used but to avoid computation a different method is used. In this case an octree is build with the normals as points, this way similar normals will fall in the same octree voxel. Initially each voxel with data will be a different plane.

The performance of this method is very sensitive to the size chosen for the voxels. Too large voxels will not be able to distinguish between similar plans and too small voxels will oversegment the planes. This problems are partially attenuated with two methods, this first one is to smooth the normals using again the range image neighbourhood idea. Every component of the tangential vectors (XYZ) is set up as an image with the same shape as the range image and a Gaussian filter is applied to it. This makes the normals more homogeneous removing small changes.

A second method is to compare the clustered planes by their mean and to merge them if we decide that they are very similar. This also solves the problem of one plane cluster falling between two voxels.

6.1.3 PARALLEL PLANES DISTINCTION

So far the algorithm is able to distinguish between planes with different normals but parallel plans share a normal while not being the same plane. During this third step this differentiation is carried out.

The feature used to distinguish between points of different parallel planes is to calculate the distance from the point to the plane deffined by the mean normal and a given point, usually $(0,0,0)$. Points from the same plane will have the same distance to this reference plane.

Finally to cluster this distances the same technique from the previous step is used.

7 POINT CLOUD CLUSTERING

8 OBJECT RECOGNITION

9 FUTURE WORK

10 CONCLUSIONS

REFERENCES

- [1] Jorg Stuckler, Nenad Biresev, and Sven Behnke *Semantic Mapping Using Object-Class Segmentation of RGB-D Images*. In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Vilamoura, Portugal, October 2012 (corrected).
- [2] Willow garage. *Object recognition kitchen*. http://wg-perception.github.io/object_recognition_core/
- [3] OpenCVorg. *Open Source Computer Vision Library* <http://opencv.org/>