

# CMSC 733, Computer Processing of Pictorial Information

## Final Project: Sexy Semantic Mapping!

Due on: 11:59:59PM on Friday, May 19 2017

Prof. Yiannis Aloimonos,  
Nitin J. Sanket

May 1, 2017

The aim of this project is to build semantic map of a 3D scene. This project gives you a peek through of robotics perception, i.e., how robots understand and build the model of a world. Aren't you excited?

In the next few sections, we'll detail how this has to be done along with the specifications of the functions for each part. **Note that this project has to be done in groups of two and only one submission per group.**

## 1 Extract object of interest from Table Top Images

You are given RGB-D frames of table top images (**of single objects**) and you need to extract the 'object of interest' from each frame and build a 3D model of the filtered object. When we say extract object of interest, you need to remove the table, walls of the room (if any) and any such irrelevant data from the scene. You also might have to filter stray points.

### 1.1 Build 3D model of Object from RGB-D Images

You are given RGB-D data from a Asus Xtion Pro sensor and also a helper code to extract 3D Point Cloud for each frame. Using the method in the previous stage, you get the 3D view of the object from one single camera view point, use many such views from different frames of the same object to iteratively build a 3D model of the object.

The idea here is you need to find out 3D translation and rotation parameters between two 3D point clouds of the same object from different frames, use these parameters to back project 3D point cloud from second frame on to the first and now you have little more information of the 3D model of the object as compared to the 3D point cloud from a single frame. You do it iteratively until you build a complete (or more or less complete) 3D model of the object.

You will need to implement any variant of the Iterative Closest Point (ICP) algorithm which gives the relative translation and rotation between two 3D point clouds. We have uploaded a lot of ICP related papers with this description for your reference, please cite the reference you follow.

## 1.2 Extract the ‘Collection of Objects’ from a given scene

Here you will pretty much redo the same thing from previous sections, only difference being the set of table top images now have a collection of objects. And you need to extract just the objects without any irrelevant data from the scene.

## 2 Segment the scene using known Objects

Now that you have reconstructed the 3D point cloud of the scene you need to segment each individual object of the scene by finding a suitable match in the set of 3D objects that you have already constructed. And while segmenting it you need to color code each object in the scene to the label of that object (feel free to use random color codes for each object such that each object is a different color). **Also, note that you cannot assume that you know the number of objects in the scene, this has to be computed by your algorithm.**

## 3 Build a Semantic Map

Now try to see if you can derive relationships between the segmented objects, like this object is at this angle with respect to another object or this object is so much smaller than another object or this object is on top of another object and so on. Think of this as modelling information how a human would think of this scene.

## 4 Stuff Given to you

The Starter code along with the calibration parameters and the function to convert images to point clouds is in the `Code` folder. Note that you are allowed to use third party code for basic stuff but not for the whole algorithm. You are only allowed to use `pcread`, `pcshow` from the MATLAB’s point cloud library. Functions like `KDTreeSearcher` and `knnsearch` can be used for point to point correspondence search. You are also allowed to use any other basic functions built into the computer vision or image processing toolbox in MATLAB.

Make sure to check `Reference Papers` folders for papers regarding ICP (Standard ICP paper is the easiest but the worst performing) and Object Segmentation. `Dataset` folder contains the data of various scenes containing individual objects and multiple objects.

## 5 Submission Guidelines

Make a video of all the objects and all the segmented scenes where you pan the 3D point cloud to show it is fully reconstructed and is correctly segmented for each object. You **MUST** submit a report written in IEEE double column format in  $\text{\LaTeX}$  and should not exceed 6 pages (Template given in **Draft** folder). The report should be of a conference paper quality. Feel free to add videos you feel are cool or paste YouTube links in your report. You should also include a detailed **README** file explaining how to run your code.

Submit your .fig files (of 3D point clouds for all individual objects), codes (.m files) with the naming convention **YourDirectoryID\_P5.zip** onto ELMS/Canvas (**Please compress it to .zip and no other format**). If your e-mail ID is ABCD@terpmail.umd.edu or ABCD@umd.edu your Directory ID will be ABCD.

To summarize, you need to submit these things and in the following strcture: A zip file with the name **YourDirectoryID\_P5.zip** onto ELMS/Canvas. A main folder with the name **YourDirectoryID\_P5** with the following things (for EACH of the given scene)):

- A video showing a full 3D reconstruction for each object
- A video showing a full 3D reconstruction and segmentation of each scene
- ‘.fig’ files for 3D models of all individual objects and scenes (color coded with segmentation labels)
- Code used for this project with a detailed **README** file
- A conference paper quality report written in IEEE double column format in  $\text{\LaTeX}$  format (Check **Draft** folder for necessary template and class files)

If your submission does not comply with the above guidelines, you’ll be given **ZERO** credit.

**EACH TEAM WILL HAVE TO MAKE A PRESENTATION AND PRESENT FOR ABOUT 7-10MINS ON THE DAY OF SUBMISSION. MORE DETAILS ABOUT TIME AND VENUE COMING SOON.**

## 6 Collaboration Policy

You can discuss the ideas with any number of people. But the code you turn-in should be from your own team and if you **SHOULD NOT USE** codes from other students. For other honor code refer to the CMSC733 Spring 2017 website.

**DON’T FORGET TO HAVE FUN AND PLAY AROUND WITH IMAGES!.**

## Acknowledgements

We would like to thank Aleksandrs Ecins for the dataset with a lot of hints and tips and Bhoram Lee (University of Pennsylvania) for help with the code. This fun project was

inspired from Nitin's research and a course project at University of Pennsylvania <http://www.grasparche.com/>.