ESE650 Final Project:
[Option2] Object Pose Estimation using ICP

Submission Due: **5/2/2016 at 11:59pm** on Canvas

In-class Presentation Date: **4/21/2016** or **4/26/2016**

In this project, you will implement the Iterative Closest Point (ICP) algorithm and use it for fine object pose estimation from RGBD images. Obtaining object pose is crucial for robots to perform interactive tasks such as grasping and manipulation. First, you will need to extract the planar background using some plane estimation technique. Then you will use ICP to search (roughly) and estimate (finely) the pose of the given object.

Download: Now available at https://upenn.box.com/2016ese650-final-opt2

Upload: on Canvas

(1) Code and Result (due 5/2/2016 11:59pm, <pennkeyID>_final.zip)
: Please do not include the data if you choose this topic, but include the object pose estimate in a mat file.

(2) Write-up (due 5/2/2016 11:59pm, <pennkeyID>_final.pdf)
: Make sure your report includes a good outline, proper visualization of results, and relevant equations (if any). You should follow the final project guidelines distributed in class as well.

Grading: Since topics vary, there is no single rubric for the final project. However, graders will consider relevance and difficulty of the topic, performance of your algorithms, completed status. Clearly presenting your approach in the form of report and presentation is also important.

Instructions and Tips

1. [CHECK DATA] You will find two sets of RGBD image sequence, 3d object model, ground truth camera poses in two different subfolders. There are also some demo scripts provided to help you interpret the data. There are no training/testing data distinctions provided, but you are free to use parts of them for validation or analysis.

2. [IMPLEMENT ICP] You should implement your ICP function. First try the standard algorithm taught in class **[1]**. At this stage, you can use the model point clouds as the two sets of points to be registered. (Take one model. That is one set. Make a copy of it and add small rotation and translation errors to the copy. That is the other set.) You are allowed to use KDTreeSearcher and knnsearch of MATLAB. Make sure your ICP function properly works before using RGBD data directly.

3. [PLANAR BACKGROUND REMOVAL] You can use any algorithm including RANSAC in order to find the floor and remove it from the scene. This step can be done independently from ICP implementation. Clean the points left in the foreground by applying some filtering if needed. The purpose of this step is to obtain the 3D location of the object candidate point cloud.

4. [ROUGH SEARCH] You will need to generate a good initial pose from the candidate point cloud. (Remember ICP is a local algorithm.) You may assume that the object is not moving and only rotated along the z-axis of the object frame in order to reduce the search space of pose. You can discretize the yaw angles and use statistics from results of ICP runs (e.g., mean-squared-errors of correspondences). You can use other algorithms to obtain an initial pose if you want.

5. [FINE ESTIMATION] Once you have a reasonable initial pose, then use ICP again to obtain refined pose. You may use an estimate from previous frames, but have to return a single best estimate at the end of the sequence.

6. [MORE ICP VARIANTS] In case you want to try other variants of ICP or more objects, please contact Bhoram! She will be happy to help.

7. [REQUIRED RESULT] Your written result should include the numeric values of the object pose estimate and proper visualization of it.

8. [CLASS PRESENTATION] For the presentation in class, you are expected to bring your own laptop or use the classroom computer. The projector has a VGA port and you may need a VGA adaptor for your laptop. Remember you MUST present for the final project.

Reference

[1] Arun, K.S., Huang, T.S. and Blostein, S.D., Least-squares fitting of two 3-D point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (5), pp.698-700, 1987.