Computer Vision 2023
(CSE344/ CSE544/ ECE344/ ECE544)
Assignment-3

Max Marks (UG/PG): 80/ 80 **Due Date**: 26/04/2023, 11:59 PM

Instructions

- Keep collaborations at high-level discussions. Copying/plagiarism will be dealt with strictly.
- Your submission should be a single zip file **Roll_Number_HW[n].zip**. Include only the **relevant files** arranged with proper names. A single **.pdf report** explaining your codes with relevant graphs, visualization and solution to theory questions.
- Remember to **turn in** after uploading on Google Classroom. No justifications would be taken regarding this after the deadline.
- Start the assignment early. Resolve all your doubts from TAs during their office hours **two days before the deadline.**
- Kindly **document** your code. Don't forget to include all the necessary plots in your report.
- All [**PG**] questions, if any, are **optional for UG** students but are **mandatory for PG** students. UG students will get BONUS marks for solving that question.
- All [**BONUS**] questions, if any, are optional for all the students. As the name suggests, BONUS marks will be awarded to all the students who solve these questions.
- Your submission **must include a single python (*.py*) file for each question**. You can submit *.ipynb* along with the *.py* files. Failing to follow the naming convention or not submitting the python files will incur a **penalty**.

---

1. (40 points) **Point Cloud Registration**
   Download the dataset from link. There are three objects and for each of them there are two point clouds and a ground truth transformation matrix. The ground truth matrix transforms points from *directory/source.ply* to the coordinate frame of *directory/target.ply*. Complete the following steps for all the 3 objects.

   1. (2 points) Visualize the source and target point clouds and make observations about their relative orientations and densities. You can make use of tools like Matplotlib, Plotly, Open3D or any other tool as per your convenience to visualize the same (Ref.).

2. (3 points) Using the ground truth transformation matrix, transform the *directory/source.ply* point cloud and visualize the transformed source point cloud. Compare the transformed source point cloud with that of *directory/target.ply*. Comment on why the transformed source point cloud is less dense than the actual target point cloud.

3. (10 points) Run point-to-point ICP (Iterative Closest Point) registration algorithm on the target and point clouds with hyperparameters of your choice. You can use open3d(link) for this task. The output of ICP will be a "learnt" transformation matrix. Report the error as the Frobenius norm of the difference in the learnt and ground truth transformation matrices. Furthermore, also report the translation error and the rotation error (Frobenius norm of the difference between translation and rotations components of the estimated and ground truth T-matrices). You can assume that no scaling is taking place during this whole time. **Hint: While making an initial guess of the T-matrix for the ICP algorithm make sure that it has valid Rotation and Translation components, i.e., the matrix should be orthonormal. You may refer to the** *ortho_group.rvs()* **function. Also, make sure that the original T-guess that you make isn't the same as the ground truth transformation matrix. If found to be the same, 0 marks would be awarded for this and the following parts.**

4. (8 points) Run multiple experiments with different hyperparameter settings to improve the performance of the model in terms of error in the transformation matrix. Make sure to add these experiments in a tabular form in your final report and highlight the best hyper-parameters for which you get the minimum error.

5. (2 points) Save the best estimated T-matrix using ICP and its corresponding T-guess matrix as pickled files. Also, mention both matrices very clearly in your report.

6. (5 points) With the best hyperparameter settings, transform your source point cloud using the estimated transformation matrix. Visualize the same and compare with the actual target point cloud and estimated target point cloud that you got using the ground truth transformation matrix. For comparison you can use mean absolute error and/or mean squared error. Give reasons for the results that you get.

7. (10 points) Repeat steps 3 through 6 for point-to-plane ICP algorithm, also supported by open3d. **Hint: For point-to-plane ICP algorithm you have to ensure that you have pre-computed the normals for the target point cloud if they are not there. To do so, you can refer to the open3d function -** *estimate_normals()*.

2. (40 points) **Panorama Generation**
Download the dataset from link. The dataset contains set of images that will be used for panorama generation. For the steps 1 to 5 you are required to use only the first two images from the set.

   1. (5 points) Keypoint detection: Extract the keypoints and descriptors from the first two images using the SIFT algorithm. SIFT (Scale-Invariant Feature Transform)

is a computer vision algorithm used for feature detection and description. After extracting the keypoints and descriptors, draw them overlaid on the original images to visualize and verify their correctness.

2. (5 + 5 points) Feature matching: Match the extracted features using two different algorithms: BruteForce and FlannBased . BruteForce is a simple algorithm that matches features by comparing all the descriptors of one image with all the descriptors of the other image. FlannBased (Fast Library for Approximate Nearest Neighbors) is a more efficient algorithm that uses a hierarchical structure to speed up the matching process. After performing the matching, display the matched features by drawing lines between them.

3. (5 points) Homography estimation: Compute Homography matrix using RANSAC. RANSAC (Random Sample Consensus) is an iterative algorithm used for robust estimation of parameters in a mathematical model. The homography matrix is used to align the two images so that they can be stitched together to form a panorama.

4. (5 points)Perspective warping: Perspective warping is a process that transforms the perspective of an image so that it appears as if it was taken from a different viewpoint. Wrap the two images together using the homography matrix and display the left and right sides of the resulting panorama before cropping.

5. (5 points) Stitching: Two images should be stitched together to form a panorama. Display the final panorama without any cropping or blending, along with the panorama after cropping and blending.

6. (10 points) Multi-Stitching: Perform multi stitching for all the images in the folder and display the final result. (Hint: Use the function implemented for Stitching).