Computer Vision 2025
(CSE344/ CSE544/ ECE344/ ECE544)
Assignment-2

Max Marks (UG/PG): 105 + 20/ 105 + 20 **Due Date**: 04/04/2025, 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. (15 points) Consider a vector $(3, -1, 4)^T$, which undergoes the following transformations in sequence:

   1. A rotation of $-\pi/6$ about the Y-axis.
   2. A rotation of $\pi/4$ about the X-axis.
   3. A reflection across the XZ-plane.
   4. Finally, a translation by $(1, 0, -2)^T$.

   (1) (4 points) Determine the overall coordinate transformation matrix (including both rotation and reflection).

(2) (3 points) Compute the new coordinates of the given vector under this transformation. Additionally, determine where the origin of the initial frame of reference is mapped.

(3) (4 points) Calculate the direction of the axis of the combined rotation (excluding reflection) in the original frame of reference and find the angle of rotation about this axis.

(4) (4 points) Using Rodrigues' formula, show that the rotation matrix obtained from the two rotations matches the matrix derived by direct computation.

2. (10 points) The image formation process can be summarized in the equation $\mathbf{x} = \mathbf{K}[\mathbf{R}|\mathbf{t}]\mathbf{X}$, where $\mathbf{K}$ is the intrinsic parameter matrix, $[\mathbf{R}|\mathbf{t}]$ are the extrinsic parameters, $\mathbf{X}$ is the 3D point and $\mathbf{x}$ is the image point in the homogeneous coordinate system. Consider a scenario where there are two cameras ($\mathbf{C}_1 \& \mathbf{C}_2$) with intrinsic parameters $\mathbf{K}_1 \& \mathbf{K}_2$ and corresponding image points $\mathbf{x}_1 \& \mathbf{x}_2$ respectively. Assume that the first camera frame of reference is known and is used as the world coordinate frame. The second camera orientation is obtained by a pure 3D rotation $\mathbf{R}$ applied to the first camera's orientation. Show that the homogeneous coordinate representation of image points $\mathbf{x}_1$ and $\mathbf{x}_2$ of $\mathbf{C}_1$ and $\mathbf{C}_2$ respectively, are related by an equation $\mathbf{x}_1 = \mathbf{H}\mathbf{x}_2$, where $\mathbf{H}$ is an invertible 3×3 matrix. Find $\mathbf{H}$ in terms of $\mathbf{K}_1, \mathbf{K}_2$ & $\mathbf{R}$.

3. (40 points) **Camera Calibration:**

   Refer to the following tutorials on camera calibration: Link1 and Link2. You are required to perform the camera calibration separately for both the 25 images provided to you in the dataset link and for 25 images that you will click separately. Place your camera (laptop or mobile phone) stationary on a table. Take the printout of a chessboard calibration pattern as shown in the links above and stick it on a hard, planar surface. Click $\sim$ 25 pictures of this chessboard pattern in many different orientations. Be sure to cover *all degrees of freedom* across the different orientations and positions of the calibration pattern. Make sure that each image *fully* contains the chessboard pattern. Additionally, the corners in the chessboard pattern should be detected automatically and correctly using appropriate functions in the OpenCV library. Include the final set of images that you use for the calibration in your report. For the dataset provided, submit a JSON file containing the estimated intrinsic parameters, extrinsic parameters and radial distortion coefficients. The JSON file should follow the specified format given here.

   1. (5 points) Report the estimated intrinsic camera parameters, i.e., focal length(s), skew parameter and principal point along with error estimates if available.
   2. (5 points) Report the estimated extrinsic camera parameters, i.e., rotation matrix and translation vector for the first 2 images
   3. (5 points) Report the estimated radial distortion coefficients. Use the radial distortion coefficients to undistort 5 of the raw images and include them in your report. Observe how straight lines at the corner of the images change upon application of the distortion coefficients. Comment briefly on this observation.

4. (5 points) Compute and report the re-projection error using the intrinsic and extrinsic camera parameters for each of the 25 selected images. Plot the error using a bar chart. Also report the mean and standard deviation of the re-projection error.

5. (10 points) Plot figures showing corners detected in the image along with the corners after the re-projection onto the image for all the 25 images. Comment on how is the reprojection error computed.

6. (10 points) Compute the checkerboard plane normals $\mathbf{n}_i^C,\ \ i \in \{1, ..25\}$ for each of the 25 selected images in the camera coordinate frame of reference (O^c).

4. (40 points) **Panorama Generation**
Download the dataset from link. The dataset consists of three sets of images, which will be used to create three distinct panoramas. Since the images are mixed, you will need to separate them using K-means clustering based on their color histograms or Visual Bag of Words. Visually inspect the results to determine which method provides more accurate separation and go ahead with that. The code for clustering should be present in the notebook. For steps 1 to 5, use only the first two images from the entire set (named image1 and image2). In step 6, perform stitching on all three sets to generate the three complete panoramas.

   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 the Homography matrix using RANSAC. Save and submit the matrix as a csv file. 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. Warp the first two images (with overlapping field of view) using their respective homography matrices and display image1 and image2 side-by-side. These *warped* images will be part of your first panorama. Display the images without cropping the images or stitching them (as asked in the next part).

   5. (5 points) Stitching: The two images need to be stitched together to form a panorama. Display the final panorama without any cropping or blending, along

with the panorama obtained after cropping and blending.

6. (10 points) Multi-Stitching: Perform multi stitching for all the images in the folder and display the final result. Multi-Stiching has to be performed on each of the three sets of images obtained after clustering. The output should be three panoramas. (Hint: Use the function implemented for Stitching).

5. (20 points) **[BONUS] Point Cloud Registration**
Download the dataset from link. The dataset contains multiple sequentially recorded point cloud (.pcd) files. These files were recorded by mounting a 3D LiDAR on a TurtleBot and capturing the point clouds during its motion. Complete the following steps to estimate the TurtleBot trajectory and visualize the registered point clouds.

   1. (5 points) Run point-to-point ICP (Iterative Closest Point) registration algorithm on any 2 consecutive 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 fitness and inlier RMSE for initial and estimated transformation matrices between the 2 point clouds. **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.**

   2. (8 points) Run multiple experiments with different hyperparameter settings to improve the performance of the model in terms of error and fitness in the transformation matrix. Compare with different threshold values and different initial guess of the identity matrix (random orthogonal matrix and RANSAC based initial guess). Also compare the error between initial and "learned" 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. **For estimating normals (if required) you can refer to the open3d function -** $estimate\_normals()$ Also mention the estimated T-matrix in your report.

   3. (2 points) With the best hyperparameter settings, transform your source point cloud using the estimated transformation matrix. Visualize the same and give reasons for the results that you get.

   4. (5 points) Repeat steps 1 through 4 for point-to-point ICP algorithm for all the point clouds and report the global registered point cloud. Also report and plot the estimated 3D trajectory of the TurtleBot. Save the trajectory in a csv file and submit that along with your report.