Assignment 5 - 3D Geometry

IMAGES. All images may be found in the DATASETS folder https://drive.google.com/drive/folders/1fUo6TQmvkCPWg9MVaEQTyx3tOZOyTEkG?usp=share_link.

1. **Independent Reading.** Read Chapters 11 and 12 from Szeliski's book.

2. **(4 pts) Homography Estimation.** You will be using the Checkerboard Images for this problem. Take a pair of images and estimate the Homography matrix H between them. You can use the `findChessboardCorners` function in OpenCV to extract the corner points. Make sure there is an one-to-one correspondence between the corner points in the two images. Report the homography matrix for image pair 1 and 9. Specifically, you need to estimate H, where $P_9 = HP_1$, where P_1 and P_9 are the extracted points in image 1 and 9, respectively.

3. **(4 pts) Fundamental Matrix Estimation.** In this problem, you will be using the stereo image pairs `viprectification_deskLeft.png` and `viprectification_deskRight.png`, provided in the DATASETS folder. Load the images and detect the key-points i.e. corner points using any predefined corner detector in Python (refer to Assignment 1). Next you will obtain the feature descriptors of the corners using the ResNet50 model in Assignment 3. Use 8×8 patches around each of the corner points to extract the features. If required, make sure to pad or resize the patches before inputting them to the model. Using the matched points, employ the 8-point algorithm to obtain the fundamental matrix. Use random subsets of 8 points to estimate a set of fundamental matrices. Then, for each fundamental matrix, compute the mean error on the rest of matched points using the Sampson distance. Report the fundamental matrix, which produced the least error and also report the mean error. **You should not use built-in functions to estimate the fundamental matrix**. Sampson distance is defined as: $\frac{(x^T F x')^2}{||F^T x||_2^2 + ||F x'||_2^2}$, where x and x' are the 2D points on the image plane in the image pair.

4. **(4 pts) Structure from Motion.** For this problem, you will be implementing the factorization based Structure from Motion (SFM). The starter code is `StructureFromMotionExample.m`, which is a modified version of SFM example in `Matlab`. You need to fill in Line 135 and write the function in Line 136. `get3DPoints` is a function which takes in a matrix of dimension $2F \times P$, F is the number of frames ($F = 2$ here), and P is the number of matched points. The output of this function should be a matrix of 3D points of dimension $P \times 3$. Go through the reading material in `Matlab` for the algorithm and data format. You can use functions related to linear equation solving and symbolic representations in `Matlab` if needed.

Submission Protocol. For problems 2 and 3, you must submit a Jupyter notebook i.e., `.ipynb` file. You are recommended to use Colab. For problem 4 you have to finish the starter code in `Matlab` and submit that. **Failure to follow instructions will lead to a deduction in marks.** You should add comments to your codes to make them reader-friendly. All codes must be uploaded in separate folders named after the problem number. Keep all the images necessary to run a code in the same folder as the code while you are submitting. You must provide explanations in your notebooks or .m files if required. The zip file to be uploaded must have your name.