HOMEWORK ASSIGNMENT #4

DUE: Tuesday, October 23, 2018, 5 P.M.

CSCI 677: Advanced Computer Vision, Prof. Nevatia

Fall Semester, 2018

This is a programming assignment to implement a simplified version of structure from motion pipeline. We will focus on reconstructing from just a pair of calibrated camera images. The goal, given such a pair, is to reconstruct the 3-D positions of a set of matching points in the images and also to infer the camera extrinsic parameters. All results can be in a coordinate frame aligned with the first camera.

A common pipeline for SFM is to compute keypoint features, compute local matches, select globally consistent matches and compute the essential matrix, decompose the essential matrix into a rotation matrix and translation vector, compute camera matrices from these and triangulate to compute 3D point positions. A final step could be to do bundle adjustment for refining the results but we will ignore this step. The output will be a sparse set of points.

OpenCV provides functions for most of the needed steps. Besides the functions that you may have used in keypoint feature extraction and matching for HW3, the useful functions are *findEssentailMat* , *recoverpose* , *triangulatepoints* and *undistortpoints*. To reduce the effort for this assignment, we also provide code using these functions (in file HW4/hw4.py). You are asked to fill in commands to display some intermediate results as described in the rest of the assignment. You will also need to input the intrinsic matrix and image pair names.

Image Data:

Assignment folder contains HW4_data folder with 3 pairs of images ((a1,a2), (b1,b2), (c1,c2)) from NYUv2 dataset (https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html). Intrinsic matrix is

$$\begin{bmatrix} 518.86 & 0 & 285.58 \\ 0 & 519.47 & 213.74 \\ 0 & 0 & 1 \end{bmatrix}.$$

What to Output?

Your program should perform SFM with these three image pairs separately and output results at the various intermediate steps:

1. It should output the computed camera matrices, rotation and translation parameters and point positions.
2. As in HW3, display some of the matches graphically, before and after applying RANSAC.
3. Display the reconstructed 3-D points from a viewpoint other than that of the original images. You may use matplotlib 3D scatter (https://matplotlib.org/gallery/mplot3d/scatter3d.html) or other visulization tools for displaying 3D points.

What to Submit?

You should submit the followings in a SINGLE PDF.

1. A brief description of the programs you write step by step. Your program should be commented clearly.

2. Show the results of intermediate steps. Use your judgment in how to display or record these results.

3. An analysis of your test results: how well does the method work?