Lab Assignment - Structure from Motion

Kaiyue Shen

November 27, 2020

1 Introduction of the Pipeline of Structure from Motion

Based on my understanding, the Structure from Motion is consisted of a sequence of operations: first, given two sets of 2D image points, the fundamental matrix and essential matrix is computed using eight-point RANSAC algorithm with extracted sift matching. Second, we decompose the essential matrix into rotation R and translation t, where we get the projection matrix P. Third, we do triangulation to get the first set of 3D points. Finally, every time a set of new 2D image points come, we use six-point DLT RANSAC algorithm to compute the new camera pose. I implement the whole pipeline as what shown in figure 1 and the results of main steps are shown in the following section.

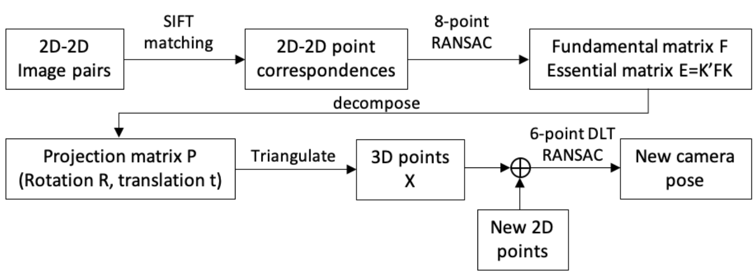

Figure 1: Pipeline of SFM

2 Experiment result

2.1 Epipolar geometry of the image pair used for initialization

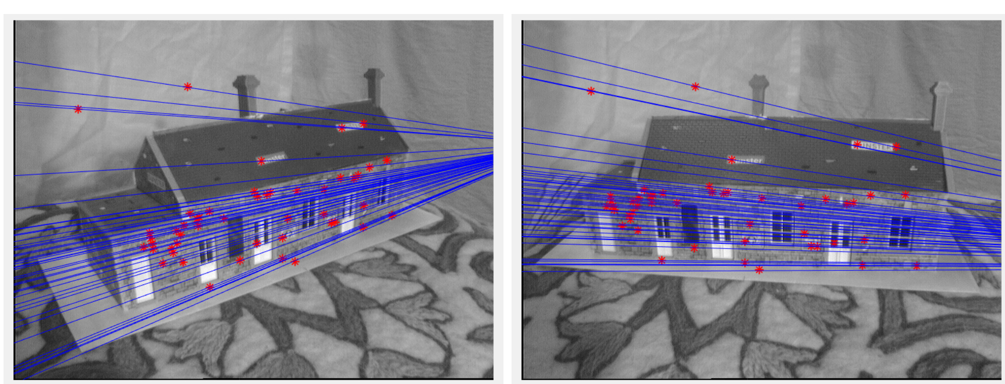

Figure 2: Epipolar geometry of the image pair used for initialization (images 0 and 4)

We use the eight-point algorithm to compute the fundamental matrix F and inlier matching. Based on these, we plot the epipolar lines for both two images that are used for initialization. We show the result in figure 2 and we can see that epipolar lines all pass the inliers. However, after a series of experiments, we find that the result is very sensitive to the randomness in RANSAC algorithm. The output of each run is different and some may result in poor results.

2.2 Inlier and outlier matches for every image used in each step

The matched inliers are connected with blue lines, and the matched outliers are connected with green lines. As we can see, at the first step, there are many outliers and mismatches, but the situation becomes better in the next few steps.

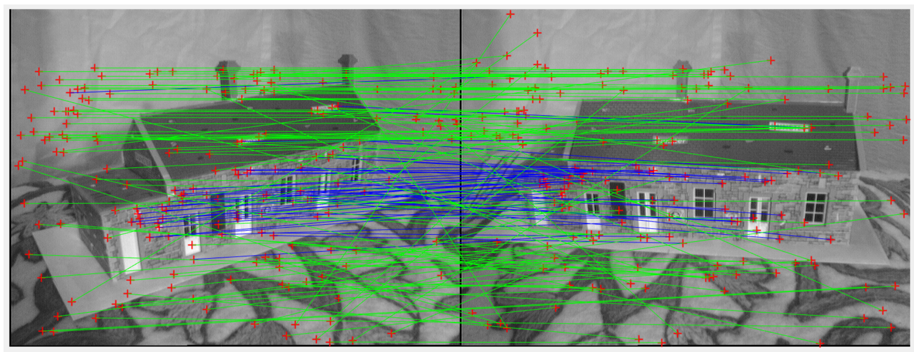

Figure 3: Inlier and outlier matches after 8-point RANSAC (images 0 and 4).

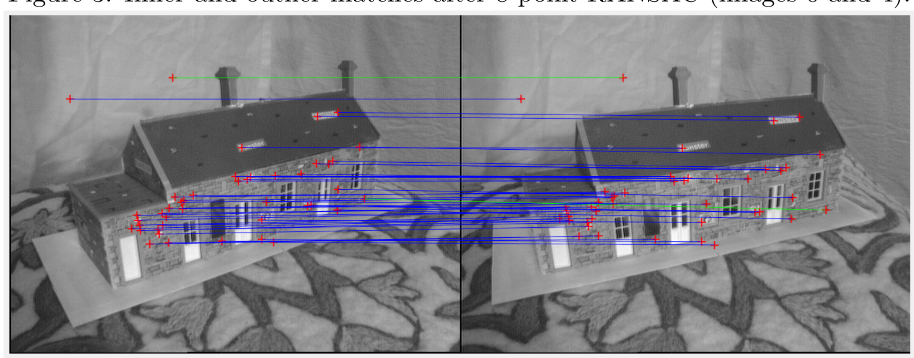

Figure 4: Inlier and outlier matches after 6-point RANSAC (images 0 and 1).

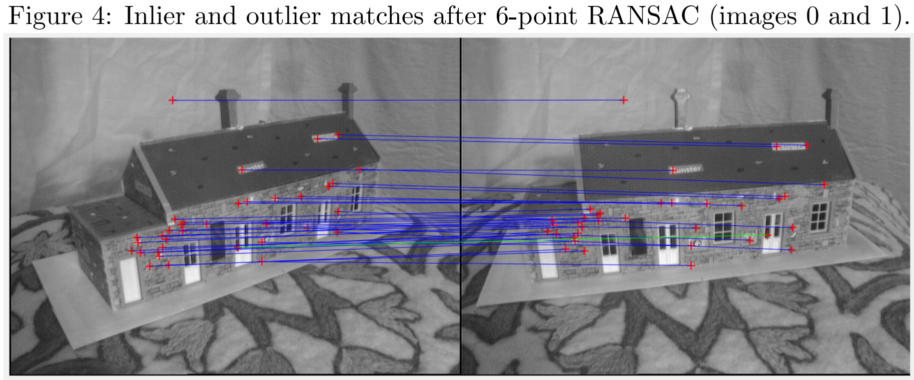

Figure 5: Inlier and outlier matches after 6-point RANSAC (images 0 and 2).

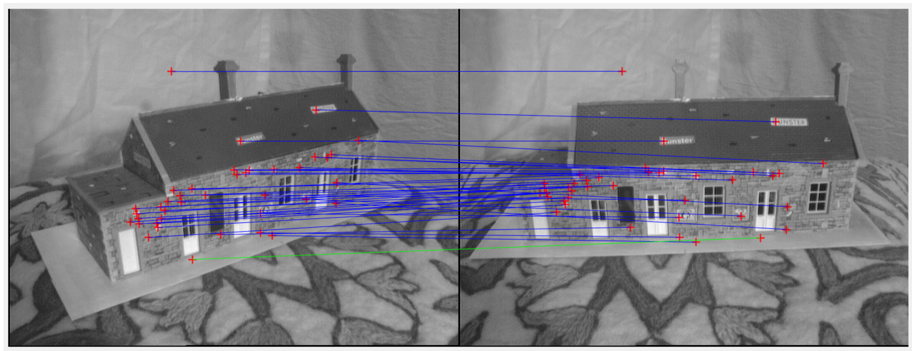

Figure 6: Inlier and outlier matches after 6-point RANSAC (images 0 and 3).

2.3 3D image of the triangulated feature point and cameras

After plotting all triangulated 3D points we get so far, we can easily find that there exit some obvious outliers: some red dots are far away from others, since we do not run the bundle adjustment to globally optimize the feature and camera poses. From figure 7, we will have a rough idea of the camera movement trajectory and 3D object point cloud. The pose that centered at the origin is from the first view, which is as we assumed. Also, the 3D points got from different views almost align with each other, showing that the results are self-consistent.

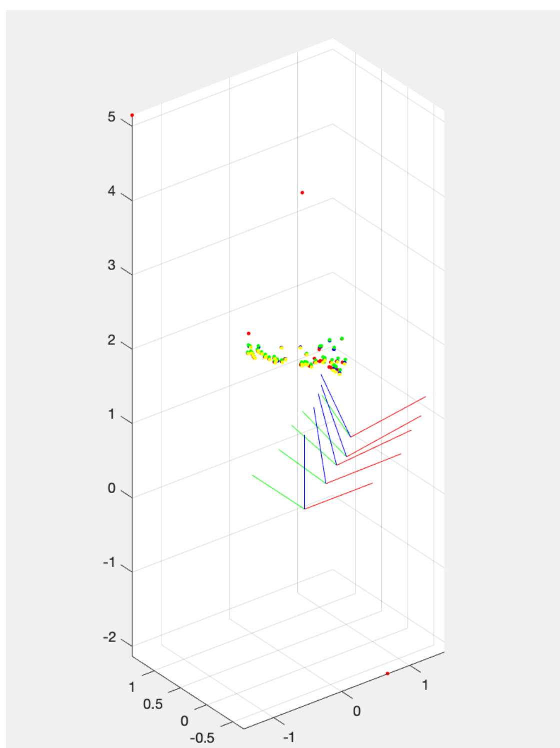

Figure 7: Triangulated points and camera poses plotted in 3D. Red dots are the triangulated points from the initialization, green dots are the triangulated points from the first additional image, blue dots are from the second addtional image and yellow dots are from the third addtional image.

3 Dense reconstruction

In this task, I choose the image 0 and image 1 to compute the depth map that needed in dense reconstruction. I reuse the code in the last exercise: image rectification (shown in figure 8), Automate disparity range selection and graph cut, to get the disparity map as shown in figure 9. I further use the formula $Z = -\frac{Bf}{d}$ to get the depth map, where f refers to the focal length, which can be got from the intrinsic matrix K, B refers to baseline length, which can be computed as the distance between two camera poses. Finally, I use the depth map to get a dense reconstruction as shown in figure 10. There are two types of 3D models, one is 3D point cloud, another is 3D mesh.

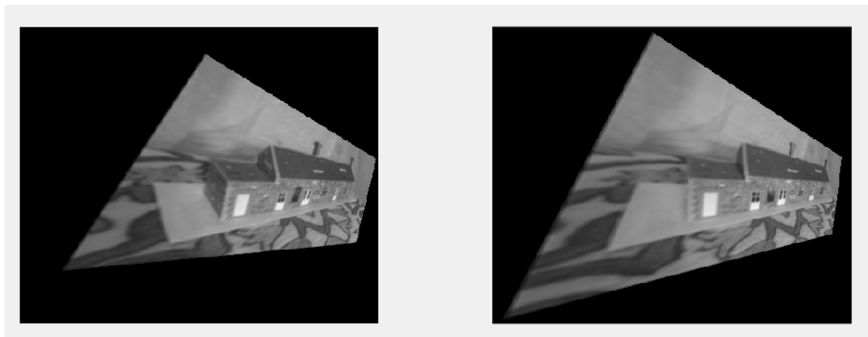

Figure 8: Rectified image 0 and 1

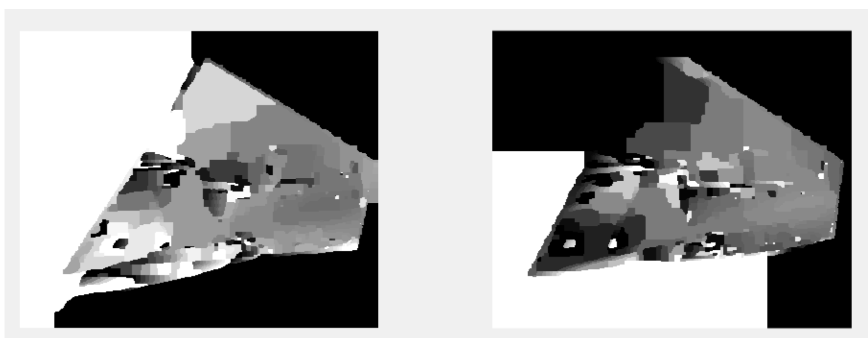

Figure 9: Disparity map of image 0 and 1

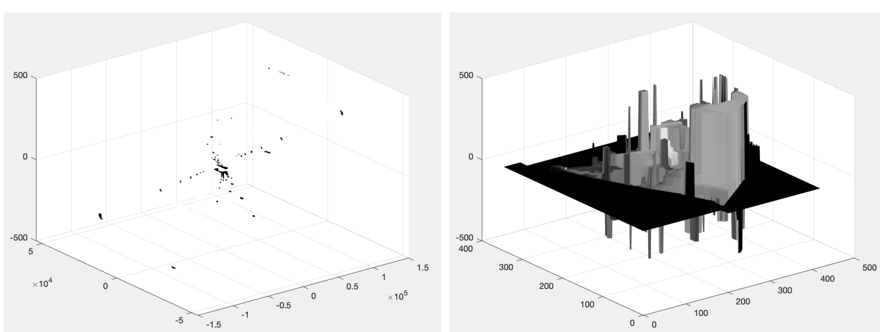

Figure 10: Dense reconstruction: (a) 3D point cloud (b) 3D mesh