Computer Vision

Structure from Motion

Name: hanxue Liang

Student ID: 16-912-156

In this exercise, I have created a simple structure from motion pipeline using the methods learned in previous lecture. The initial estimate of camera matrices is obtained by using the Essential matrix.

1. Feature extraction and initialization with epipolar geometry

In this part, I extract features from every images and initialize the projective matrix for second view presuming that the first is [I | 0].

a) Extract SIFT features and match them for two images

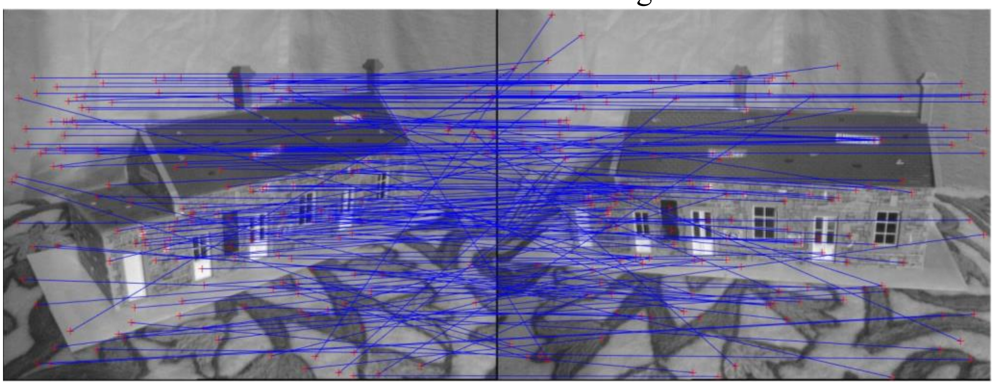

Figure1: sift features between house0 and house4

b) Run either 8-point or 5-point RANSAC to compute the inlier-set and then compute the essential matrix. Decompose the essential matrix in to R and t and create the projection matrix for the second view assuming that the first one is [I j 0].

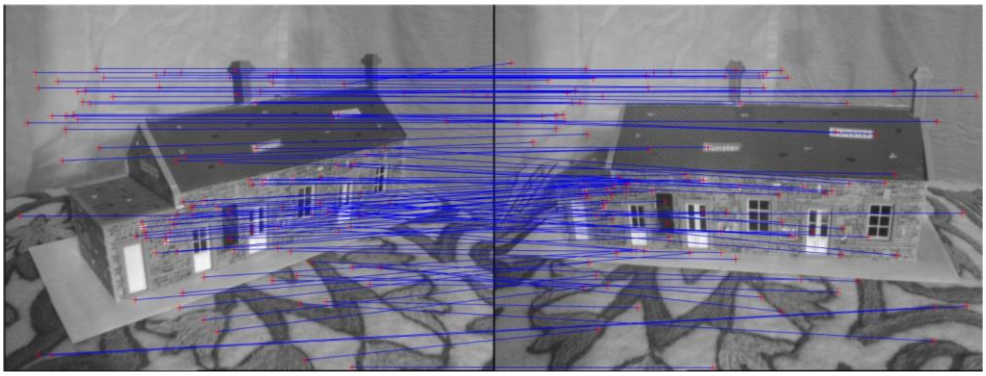

Figure2: Inlier matches after 8-point RANSAC (images 0 and 4)

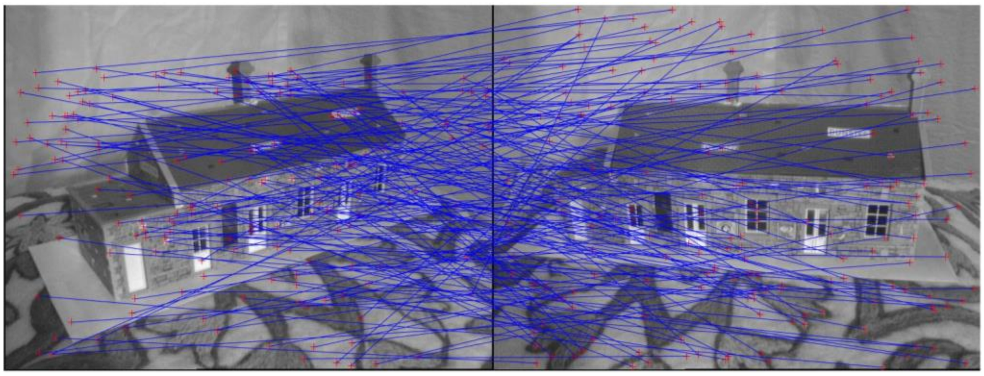

Figure3: Outlier matches after 8-point RANSAC (images 0 and 4)

I computed E as:

-0.351220404922194	0.0533954728866649	0.313614228684681
3.82962177197030	-1.15857277564801	-3.48031845760818
0.364068643442493	3.29006100363884	0.0275006590022724

I found R as:

0.650079386361685	-0.169561516386118	0.740706205987540
0.234511986826005	0.971970061631328	0.0166831450066399
-0.722773076049832	0.162859115351991	0.671621909324161

And t as:

[-0.995834285089350 -0.0898577371078936 -0.0154810762685279]'

c) Epipolar geometry of the image pair used for initialization

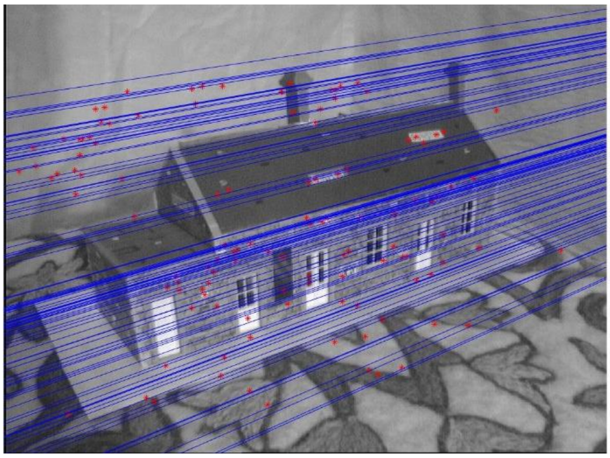

Figure4: Epipolar lines on image house0

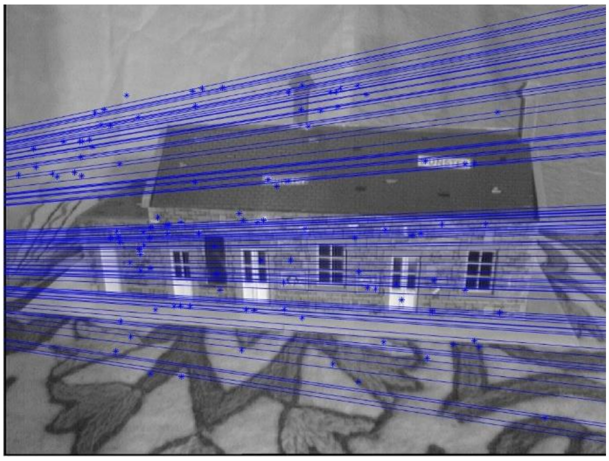

Figure5: Epipolar lines on image house4

2. Triangulation and adding new views

In this part, I will add additional views of the scene by matching 2D features of the new view with those of the initial house0 view. The inlier and outlier matches for every image used in this step are shown below:

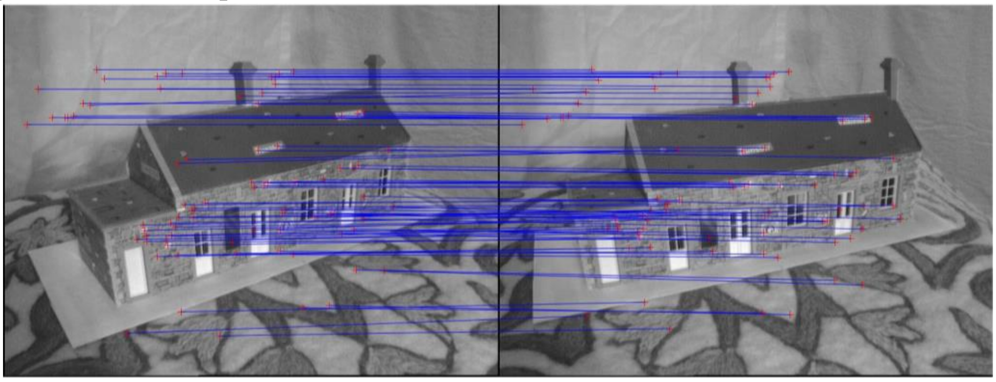

Figure6: Inlier matches after 8-point RANSAC (images 0 and 1)

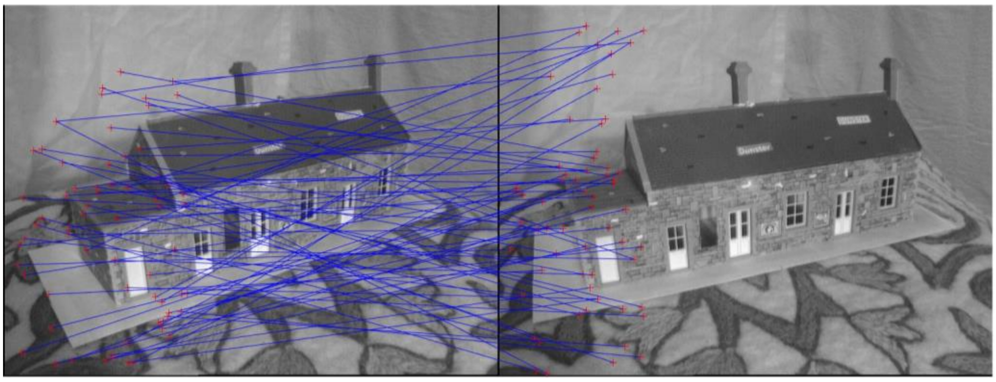

Figure7: Outlier matches after 8-point RANSAC (images 0 and 1)

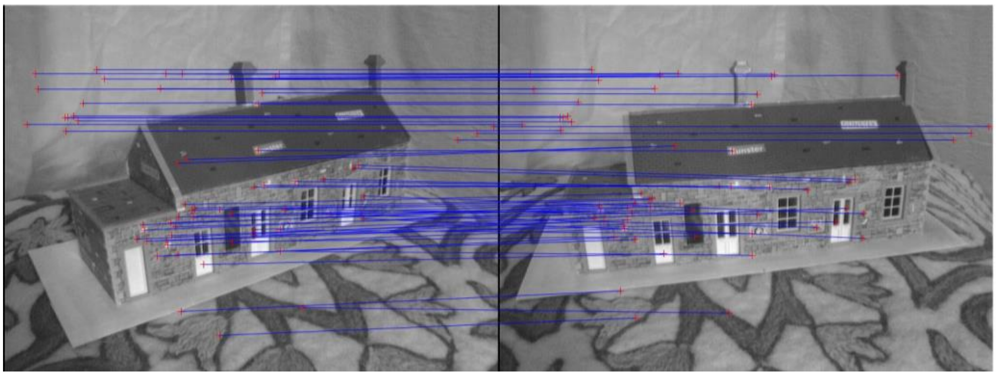

Figure8: Inlier matches after 8-point RANSAC (images 0 and 2)

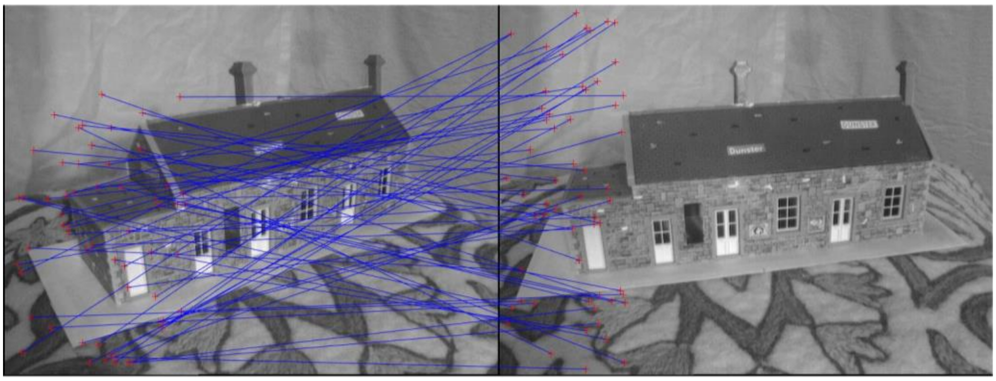

Figure9: Outlier matches after 8-point RANSAC (images 0 and 2)

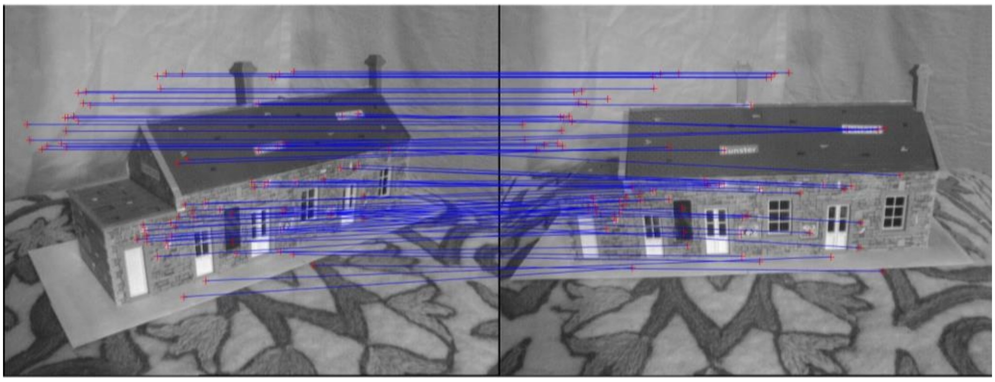

Figure10: Inlier matches after 8-point RANSAC (images 0 and 3)

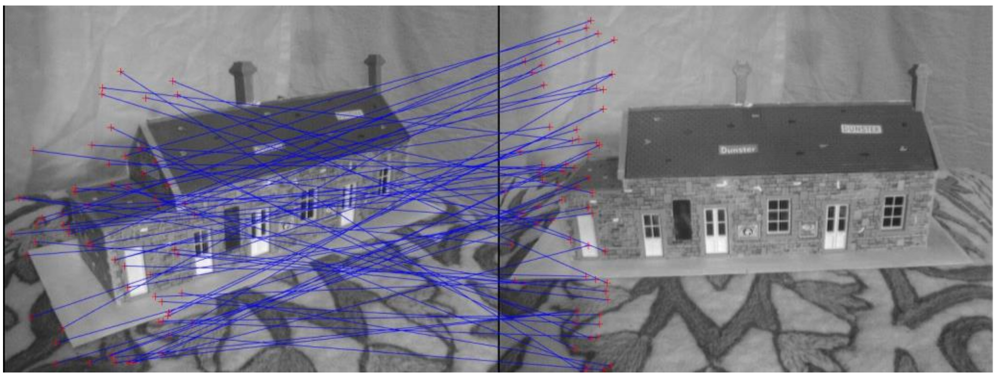

Figure11: Outlier matches after 8-point RANSAC (images 0 and 3)

3. Plotting

In this part, I triangulate all inlier matches from the previous task and plot them in 3D together with the inliers from the first task. And I also visualize the camera poses of every view using function ‘drawCameras’.

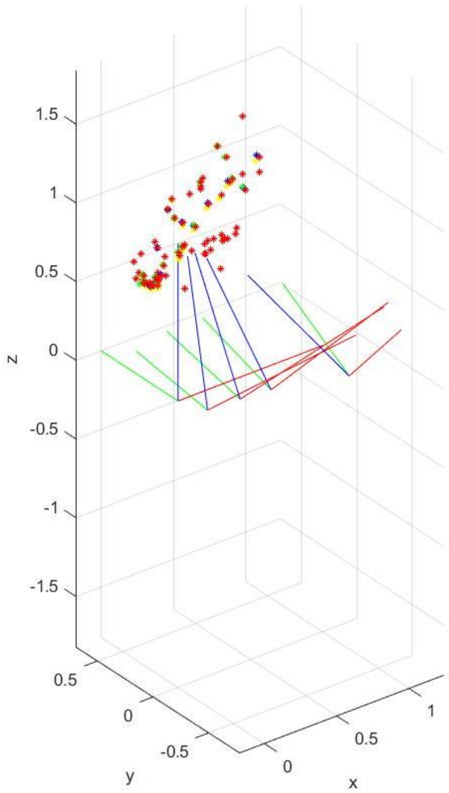

Figure 12: Triangulated points and camera poses plotted in 3D. Red dots are the triangulated points from the initialization, green dots are the triangulated points from the first additional image, blue dots are from the second additional image, yellow dots are from the third additional image.

4. Dense reconstruction

In this part, I use Graphcut algorithm to get the disparity of the pixel. And then I compute the depth map with stereo matching and triangulate every pixel that has a depth and project it into the scene. I use image0 and image1 to compute the dense reconstruction and the result is shown below:

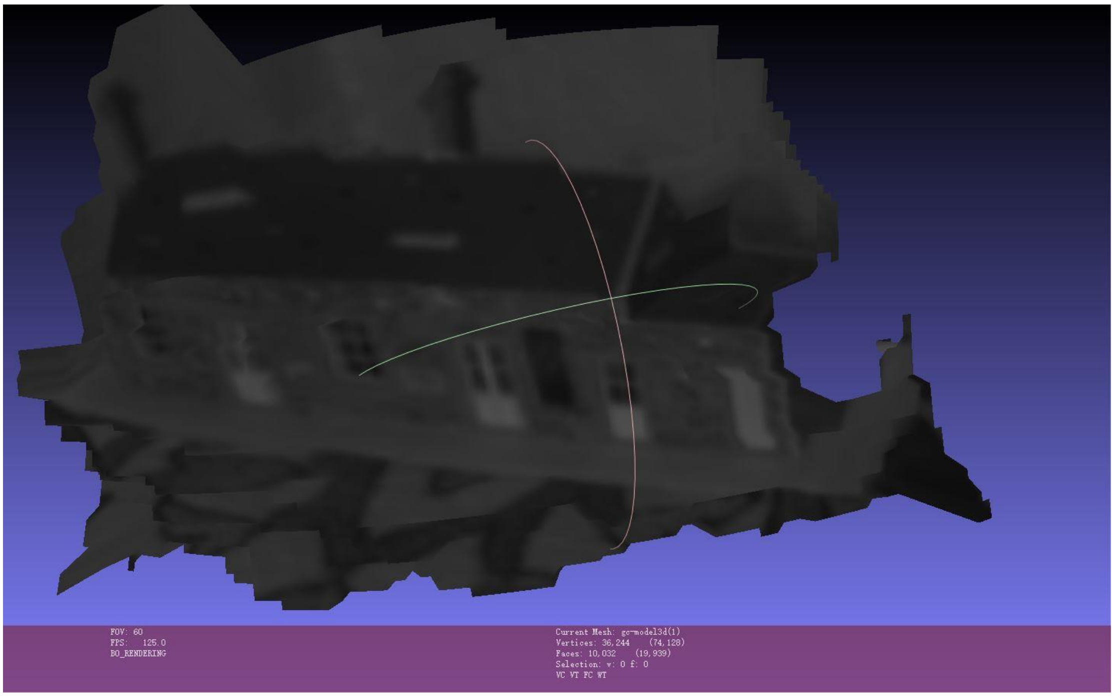

Figure 13: Triangulated points and dense reconstruction for image0