Computer Vision

Exercise4

Name: hanxue Liang

Student ID: 16-912-156

In this exercise, I will computer fundamental matrix and essential matrix with eight-point algorithm. And then I will realize RANSAC algorithm to compute fundamental matrix estimation and line fitting.

1. Line fitting with ransac

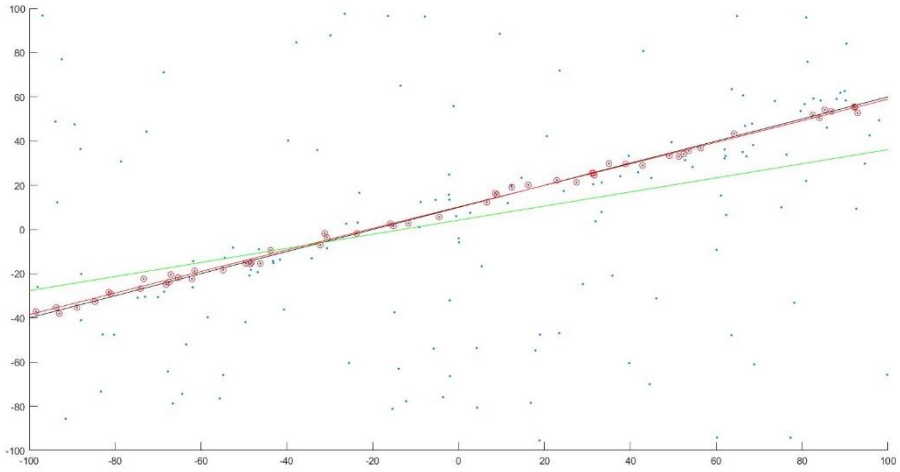


Figure1. Images for line fitting

I implement a line fitting algorithm with RANSAC, with the threshold of 'inliners' distance as 3.0 and the number of iteration as 300. Finally, my algorithm gets the k=0.5251 and b=9.8531, compared with the true value k=0.5 and b=10. Correspondingly, I get the err_ls(194.2498), err_ransac(40.8317) and err_real(40.2471). The results are shown in the figure above. It's obvious that RANSAC algorithm could get a much better results than least square fitting algorithm.

2. Fundamental matrix

I implement the eight-points algorithm to get the fundamental matrix and show the

epipolar lines for non-singular and singular matrix and the results are shown below.

For the image pair ladybug

Fh=

8.46199632703236e-06 -0.000142651229086274 0.0489220710317916
0.000139628288854016 -1.11884526677121e-05 -0.0625128599370519
-0.0530800331180422 0.0721416433416011 -1.44058188985709

F=

8.39450320308751e-06 -0.000142709059866823 0.0489929283843070
0.000139540006068001 -1.12640968537542e-05 -0.0624201766513470
-0.0529944960954804 0.0722149348488159 -1.53038254815421

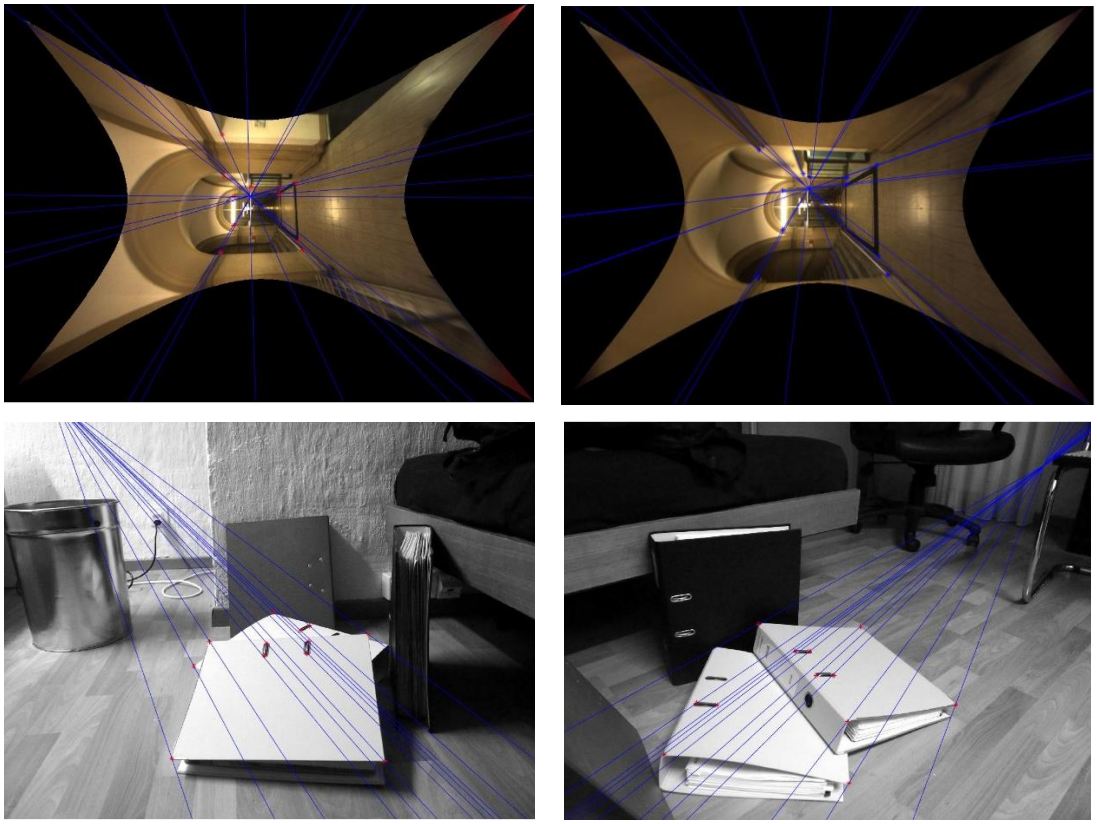
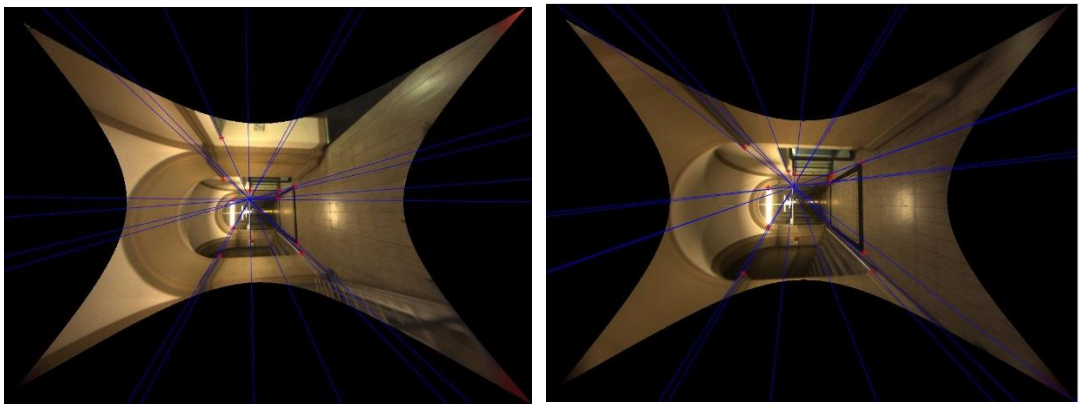


Figure2. Images with non-singular F



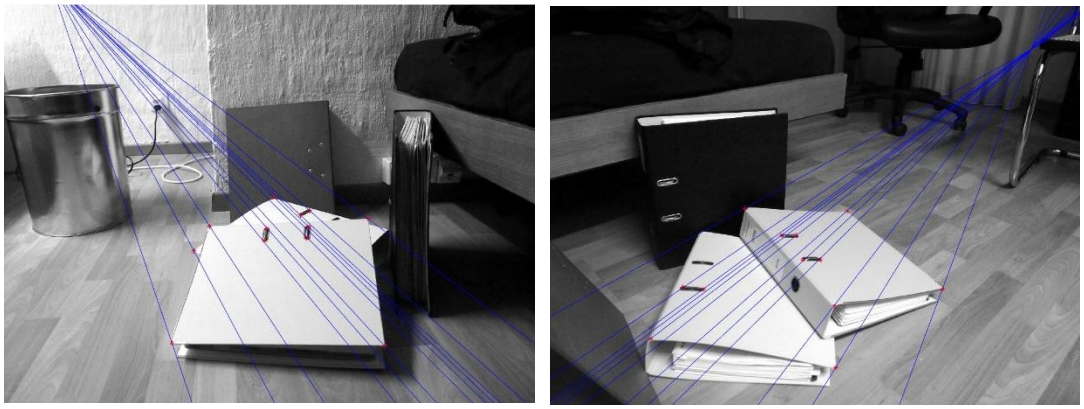


Figure3. Images with singular $\hat{F}$

I find that both for singular $\hat{F}$ and non-sigular F, all the epipolar lines could interact at the same points, which are the epipoles of the fundamental matrix. But for $\hat{F}$, the epipolar lines do not always cross the points I initially clicked. This phenomenon is true for the other two examles.

3. Essential matrix

The essential matrix E is just fundamental matrix for calibrated cameras. So I still follow the eight-points algorithm, except that the coordinates of the points have been changed by left multiplying the intrinsic calibration matrix of the camera. After computing the essential matrix, I get the fundamental matrix by multiplying the camera matrix and the results are show below.

For the image pair ladybug

Fh=

-5.08386608228308e-06	-0.000143957681793745	0.0558747164433533
0.000139050520405319	2.23342639476019e-06	-0.0672201596140131
-0.0463998143534050	0.0680529562425078	-3.11210108761815

F=

8.39450320308745e-06	-0.000142709059866822	0.0489929283843069
0.000139540006068001	-1.12640968537544e-05	-0.0624201766513469
-0.0529944960954802	0.0722149348488159	-1.53038254815423

Compare results with the fundamental matrix obtained from last exercise, I find that the fundamental matrix F obtained from these two methods are very similar. It tells us that we can either use corresponding images points to directly get F, or we can also firtly calculate essential matrix and then calculate F with intrinsic camera matrix. However, after forcing singular constraint, the fundamental matrix $\hat{F}$ is different. That's because for exercise 4.2 we forcing constraint directly on F, while this time we enforce it on E.

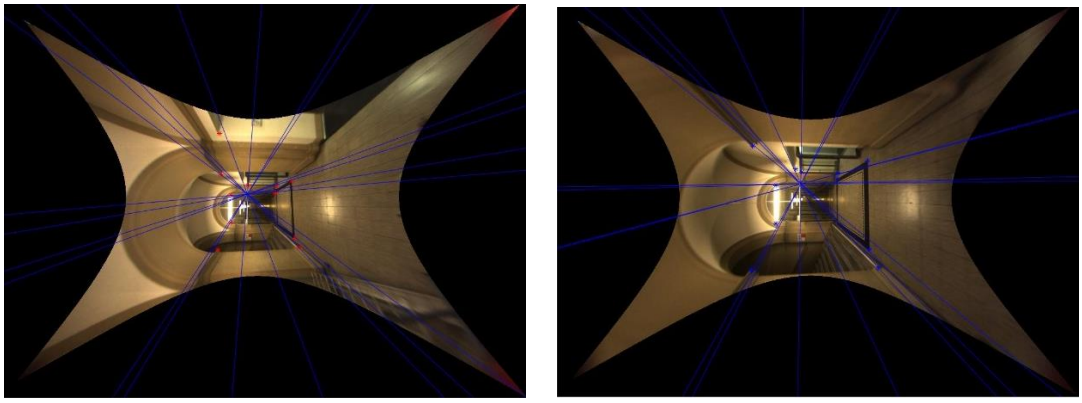


Figure4. Images with F computed from singular E

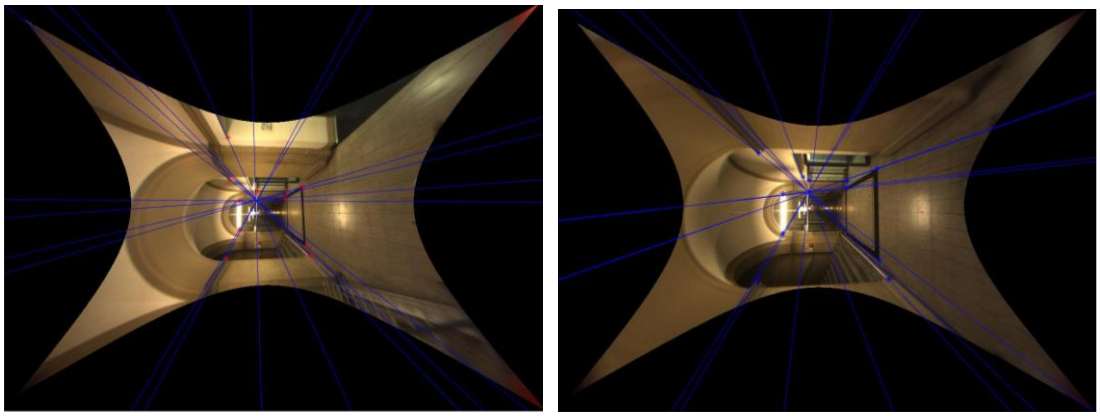


Figure5. Images with F computed from non-singular E

4. Camera matrix

After computing the essential matrix, I'm able to computer the camera matrix. The method provides four solutions and I select the one which complies with the reality: all the points must be in front of the two cameras, means having positive Z values.

The position of two cameras taking pictures of ladybug are shown in the figure below with the function: *showCameras.m*.

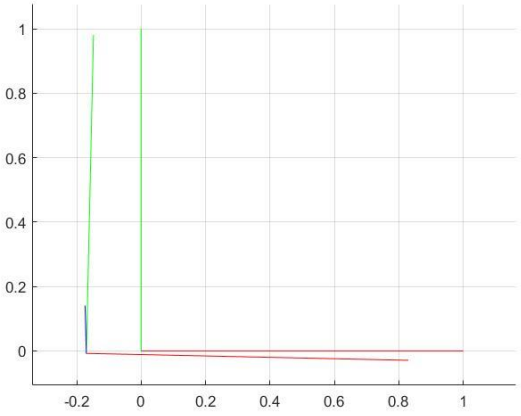


Figure6. Position of the camera for ladybug

5. Feature extraction and matching

During this part of the exercise, I use SIFT implementation: vlfeat and *showFeatureMatches.m* function. The figure is shown below.

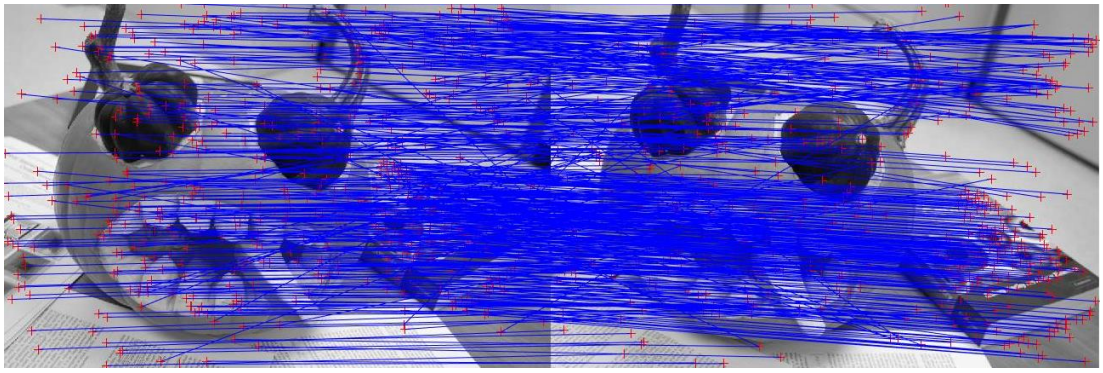


Figure7. Images overlaid with extracted SIFT features

It can be found from the image that there are some wrong matched points. Our following task will be selecting from these matched points candidates to find the correct matches and computing the fundamental matrix.

6. Eight-Point RANSAC

In this part, I implement the RANSAC algorithm which can automatically remove the incorrect matches from the previous part. Two versions of Ransac are used here.

The first one is that we fix the number of iterations to terminate the selection. Choosing eight points to define a model and compute the distance of the other matches form this model. Distances smaller than a threshold will be considered to be inliers and others will be considered to be outliers. I choose the number of iteration to be 1000.

When choosing the threshold to be 5, initial matches, inliers and outliers are shown below:

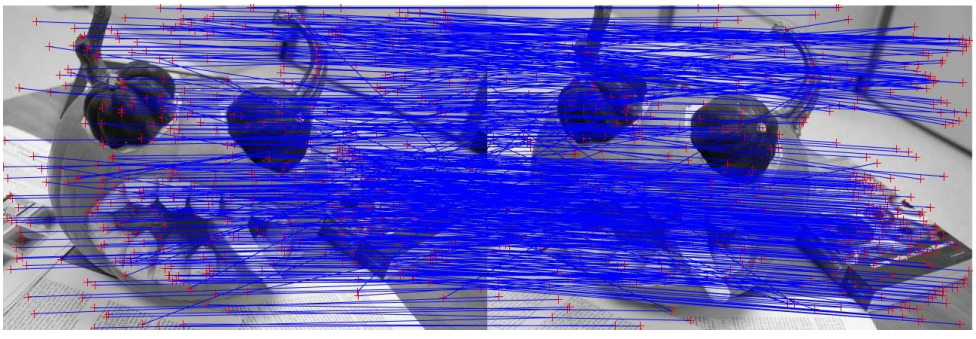


Figure8a.initial matches with vlfeat (630 matches)

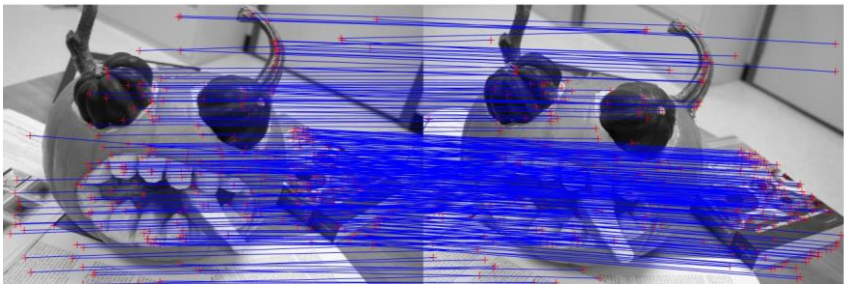


Figure8b.inliers with threshold 5 (349 matches)

Figure9c. outliers with threshold 5 (281 matches)

When I choose the threshold to be 3, the number of inliers decrease to 302 and number of outliers increase to 328. When I further decrease the threshold, the number of inliers continues to decrease.

The second version of ransac will be Adaptive RANSAC. Instead of fixing the number of iteration, we terminate the RANSAC process after M trials if we know with a probability 0.99 that at least one of the random samples of N = 8 points from these M trials is free from outliers.

When I choose the threshold to be 3， m =987, the number of inliers is 322 and number of outliers is 308. The figure is shown below:

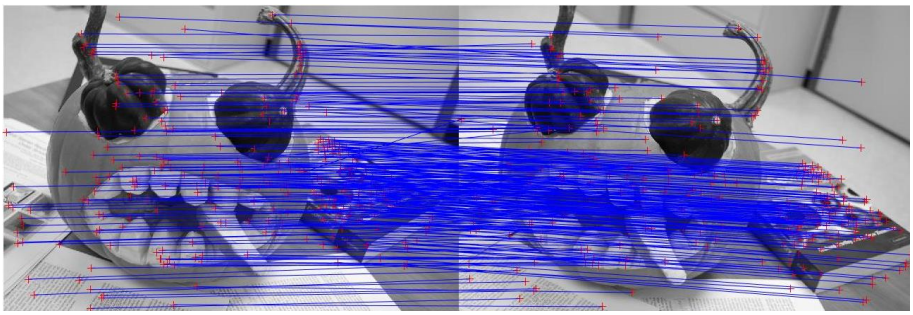


Figure10a.Inliers with threshold 3 (adaptive Ransac)

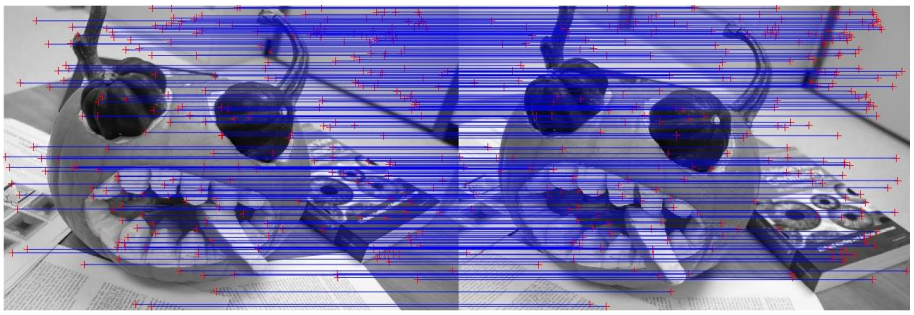


Figure10b.Outliers with threshold 3 (adaptive Ransac)