Computer Vision

Alessia Paccagnella

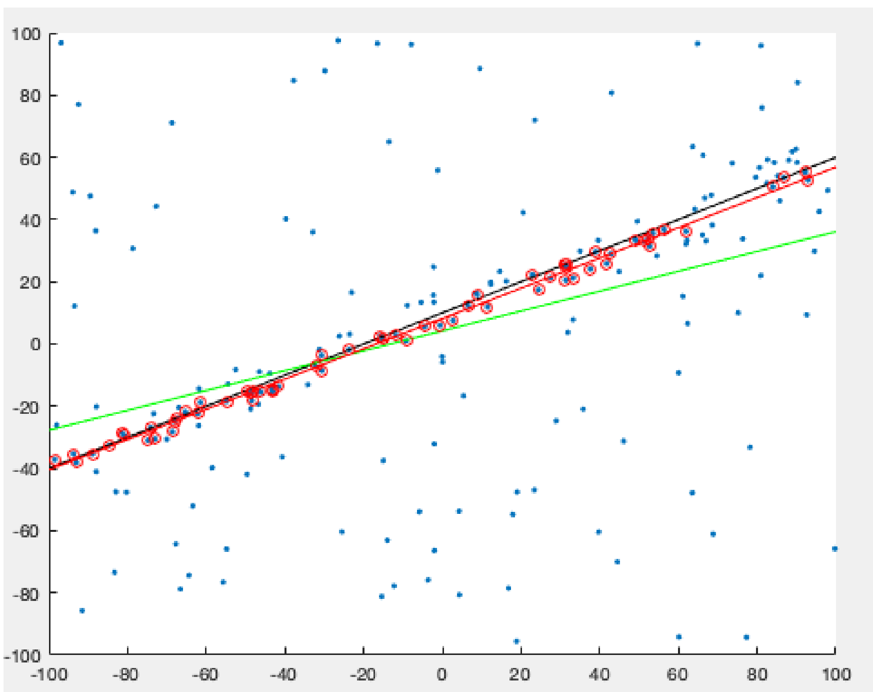
October 30, 2019

1 Line fitting

For this task I filled the *ransac.m* file.
I randomly selected 2 points on the data set, and checked that they had different x coordinates (to prevent a division for 0 later). I calculated the model line, and found the number of inliers, the points whose perpendicular distance from the line was minor then the threshold. If the number of inliers was higher then the previous number of inliers, then I updated the best model parameters.

```
err_real = 40.6666
err_ls = 124.9643
err_ransac = 43.0701
```

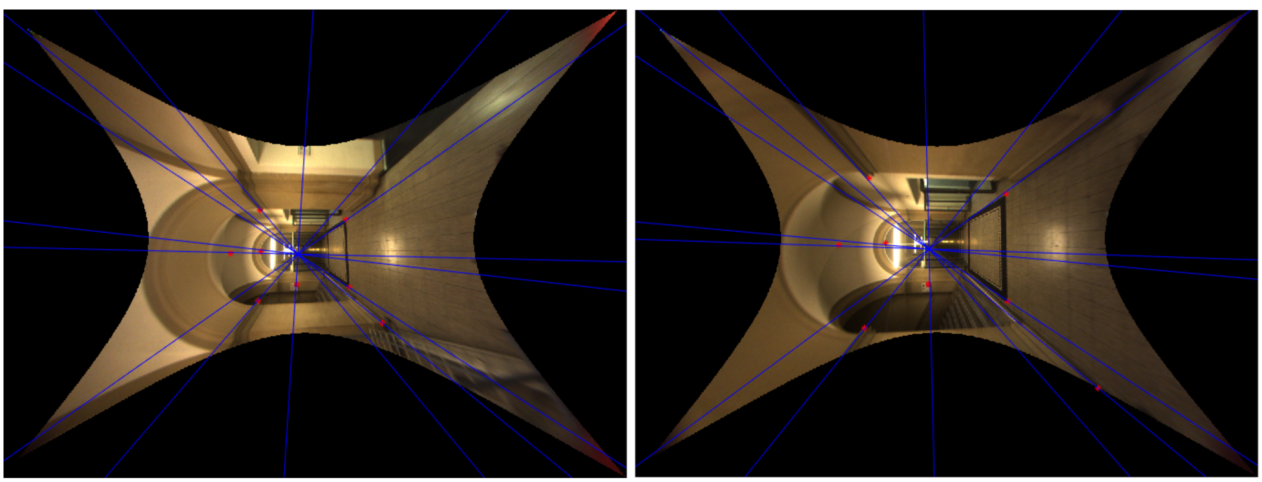
In the image, the black line represents the real model, the green line represents the least square fitting and the red line the RANSAC.



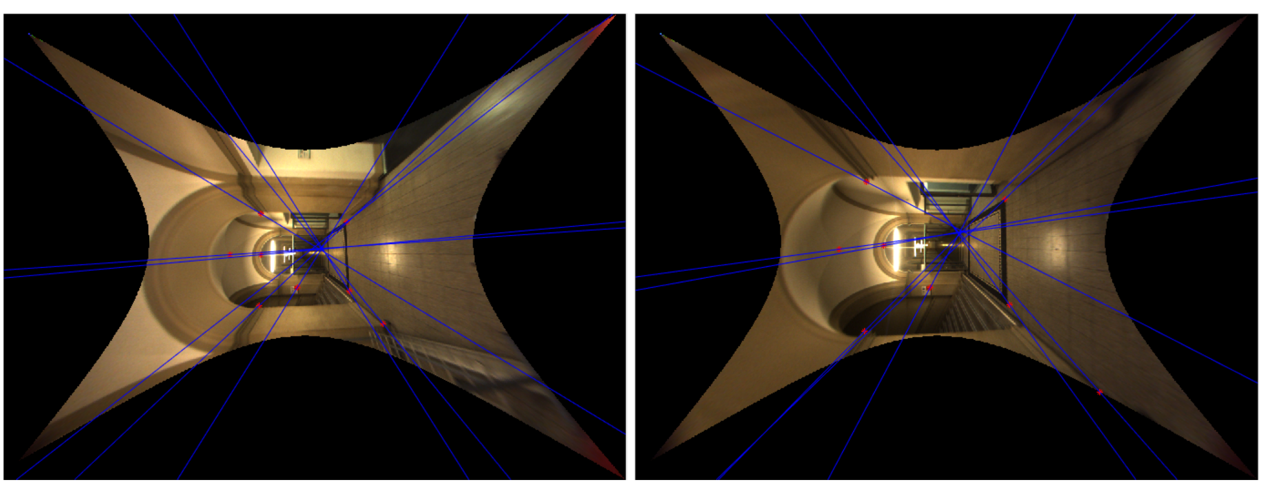
2 Fundamental matrix estimation

To compute the fundamental matrix F, I implemented the eight-point algorithm.
• The first thing I did was to normalize the 2D points. However, I noticed that the points in the *fundamentalMatrix.m* file were in homogeneous coordinates (had size 3). This led me to add a check in the normalization file: if the points are in homogeneous coordinates, i have to change the coordinates (so they have size 2). After that, I normalize them with the matrix T from the first lab assignment.
• I created the A matrix just like in the slides.
• I did the SVD of A, and constructed F as a 3x3 matrix with the solution of the last column of V.
• I applied the constraint of the determinant equals to 0 to the diagonal matrix, and then re-calculated F matrix.
• I return the fundamental matrix Fh with the determinant constraint, and the F Initial fundamental matrix obtained from the eight point algorithm.

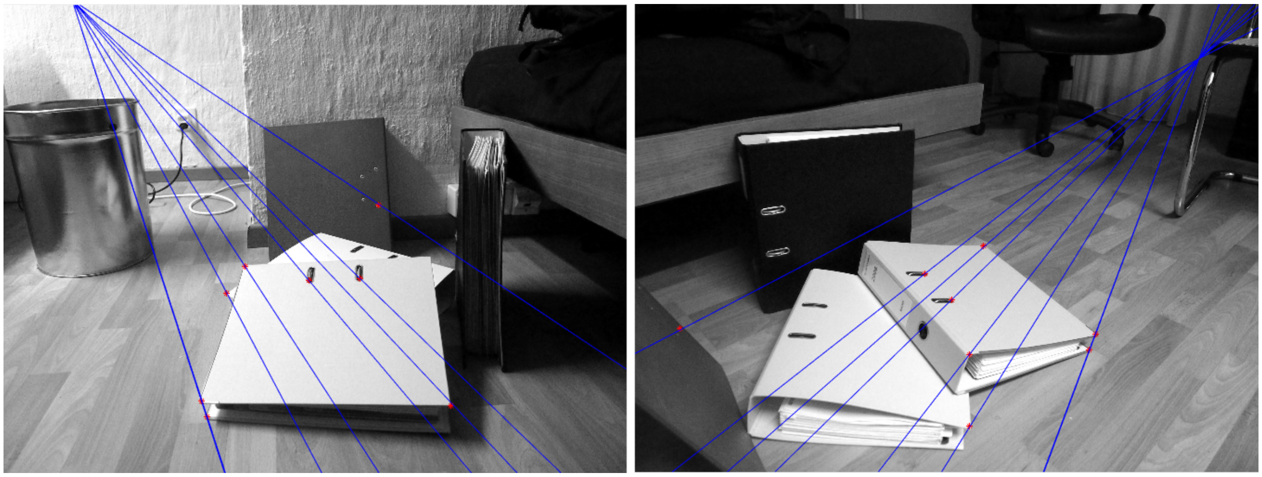
Showing epipolar lines. Matrix F with singularity constraint.



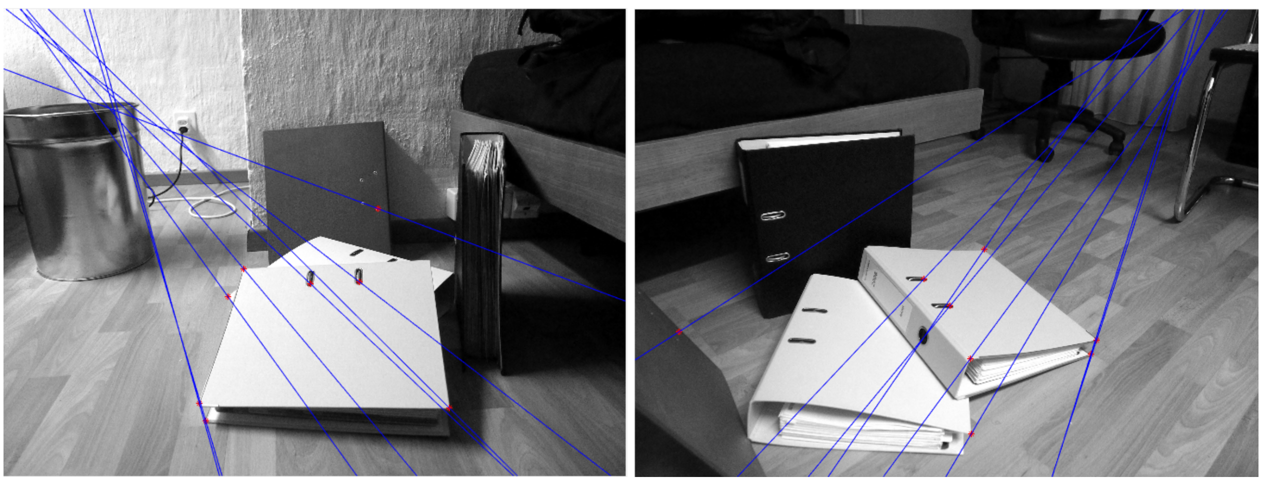
Showing epipolar lines. Matrix F without singularity constraint.



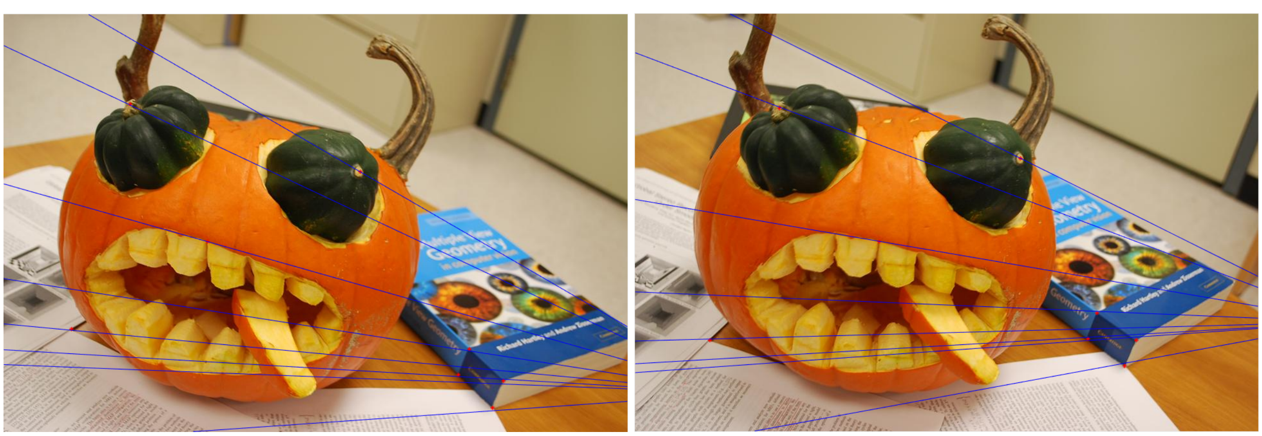
Showing epipolar lines. Matrix F with singularity constraint.



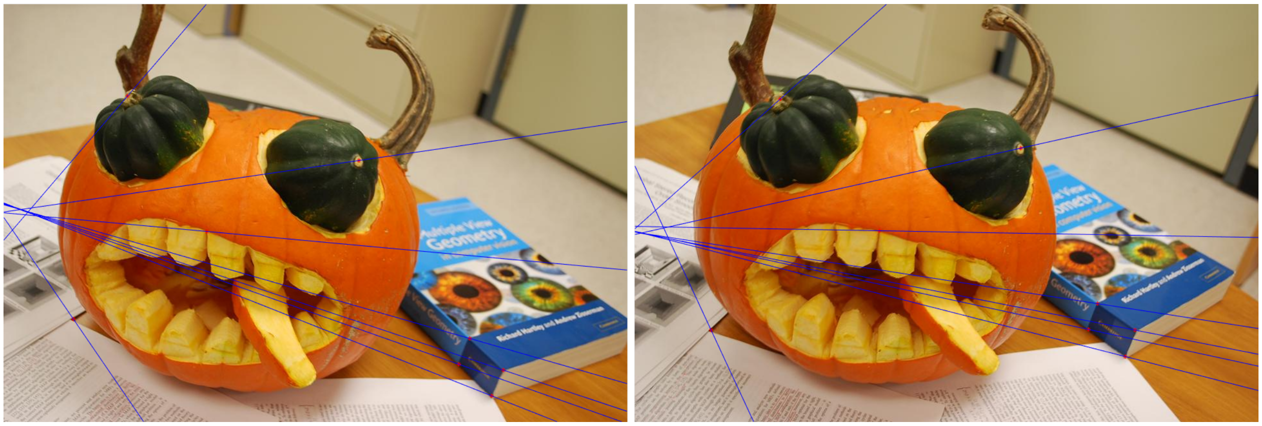
Showing epipolar lines. Matrix F without singularity constraint.



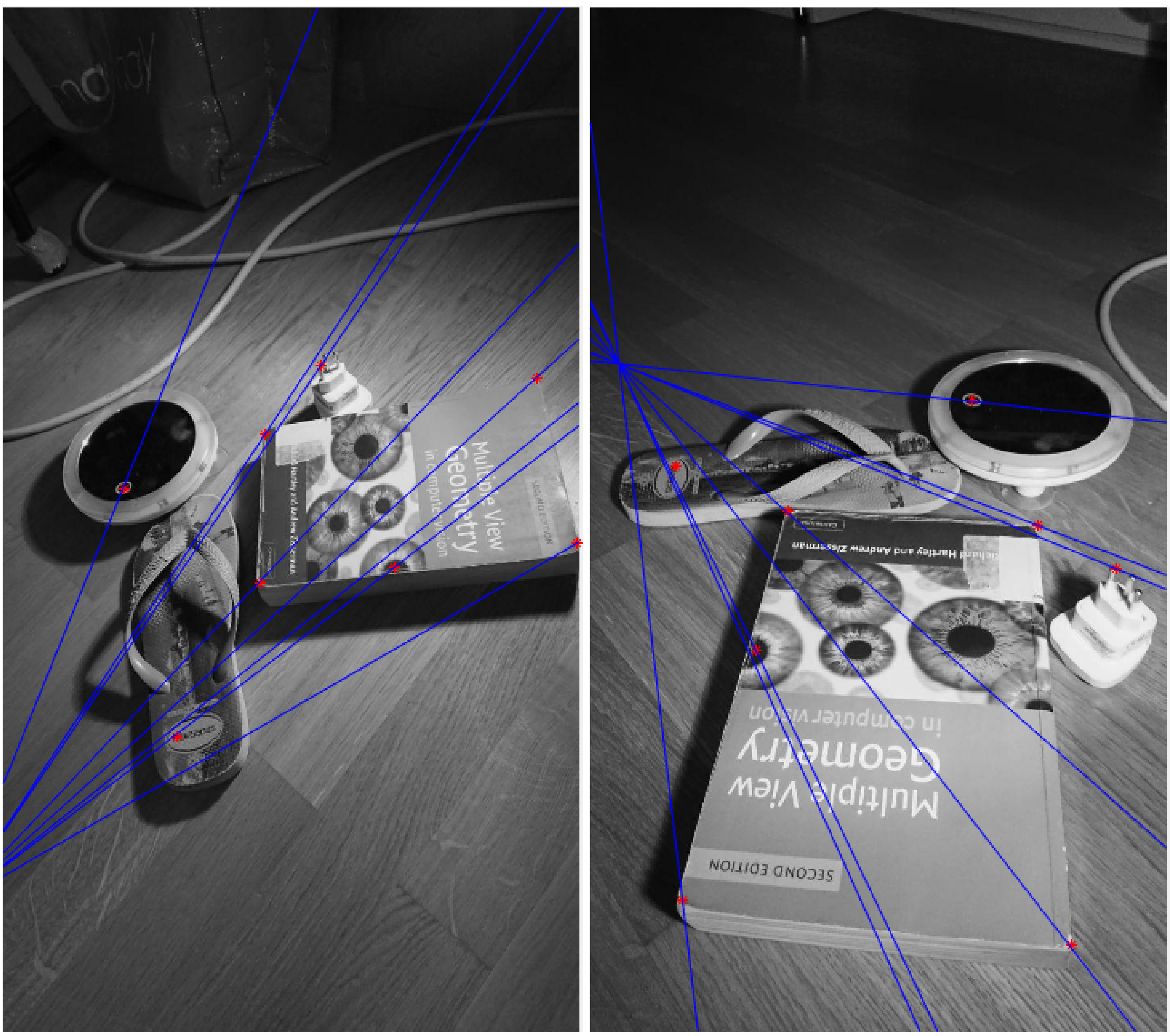
Showing epipolar lines. Matrix F with singularity constraint.



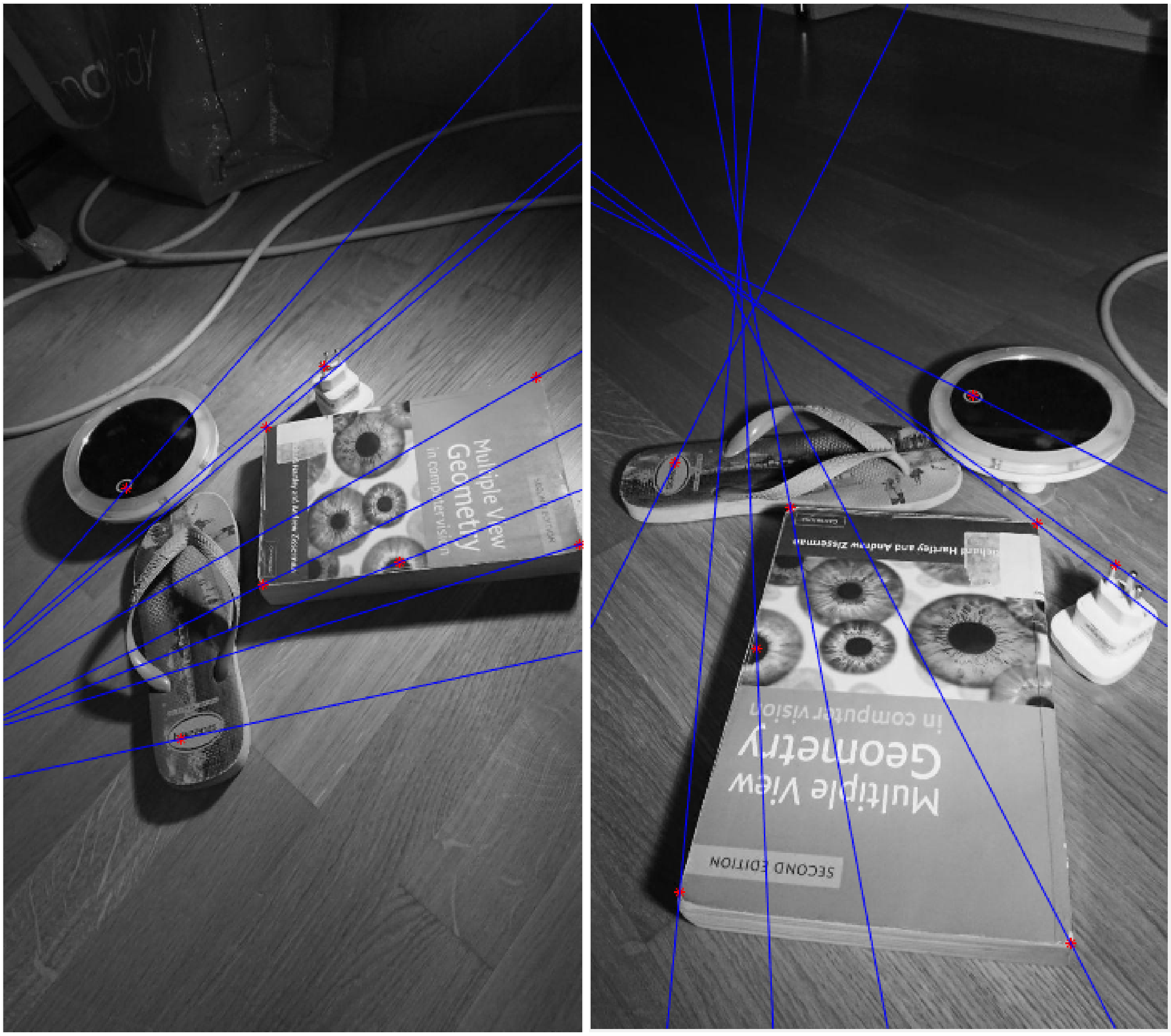
Showing epipolar lines. Matrix F without singularity constraint.



I also tried with my own image. Showing epipolar lines. Matrix F with singularity constraint.

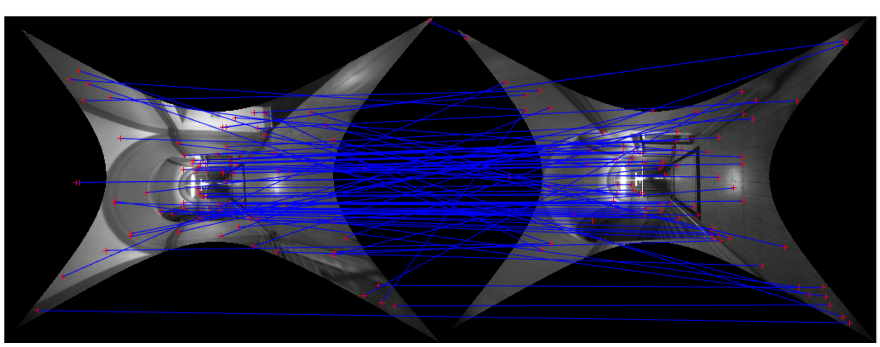


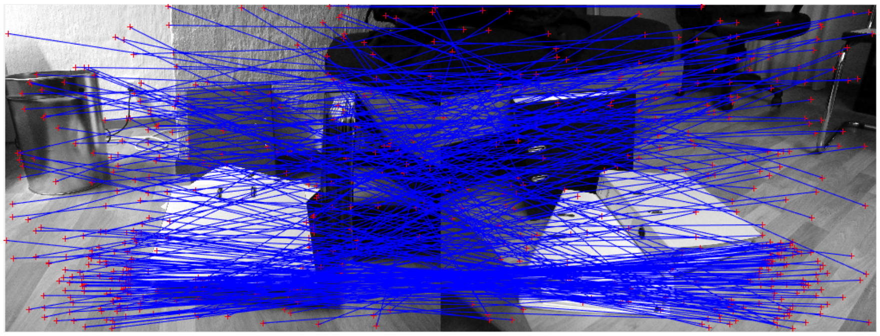
Showing epipolar lines. Matrix F without singularity constraint.

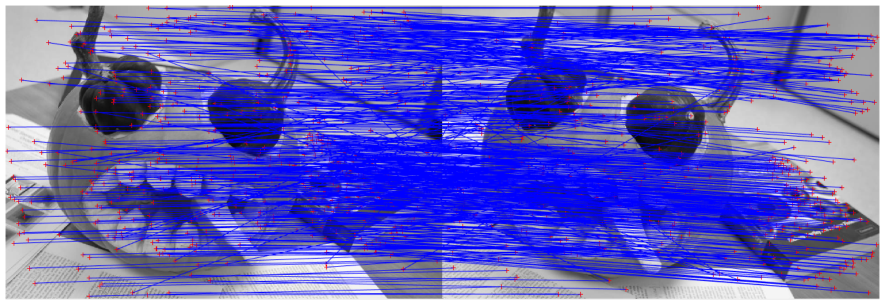


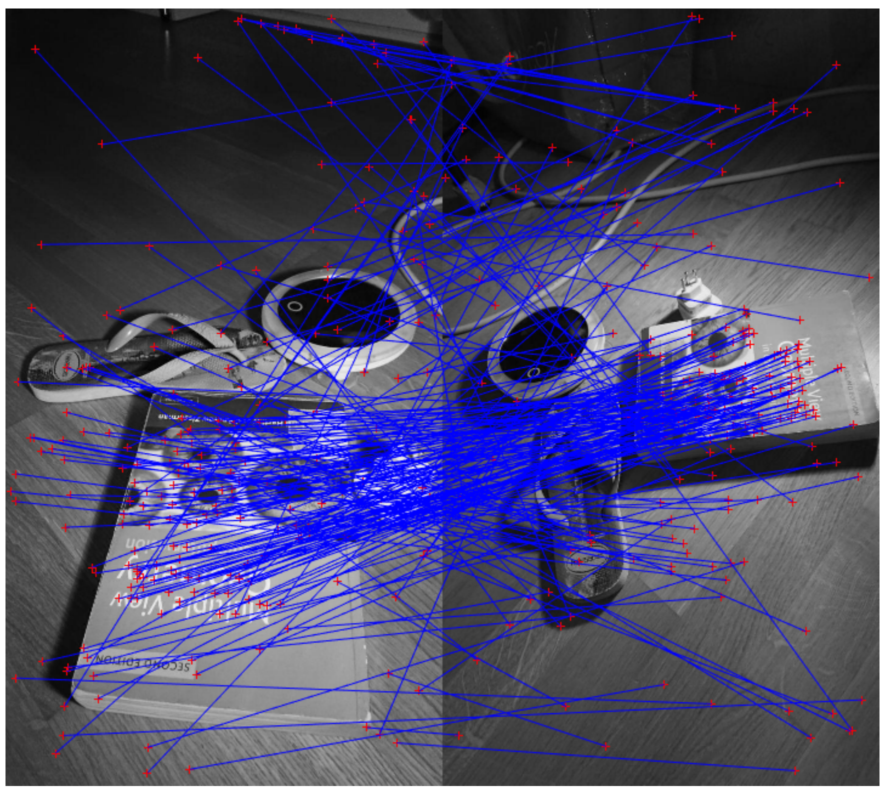
3 Feature extraction and matching

I installed VLFeat to extract and match SIFT features for the same image pairs used before. I used vl_sift and after I used showFeatureMatches function to plot the obtained matches:









4 Eight-point RANSAC

4.1 Simple Ransac

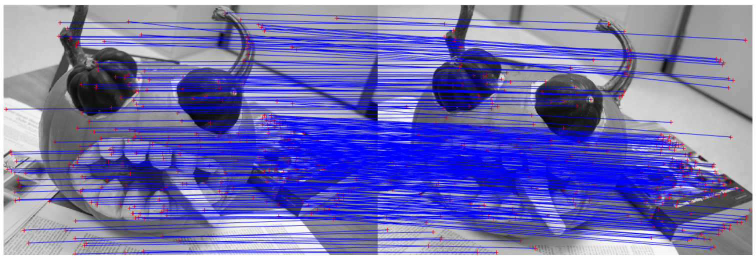
To eliminate the wrong matches from before, we use RANSAC. I iterated 1000 times, each time selecting 8 random points from the data set and estimated the fundamental matrix. I computed the Sampson error with this formula:

$$\mathcal{S}(x, x') = d(x', Fx) + d(x, F^T x')$$

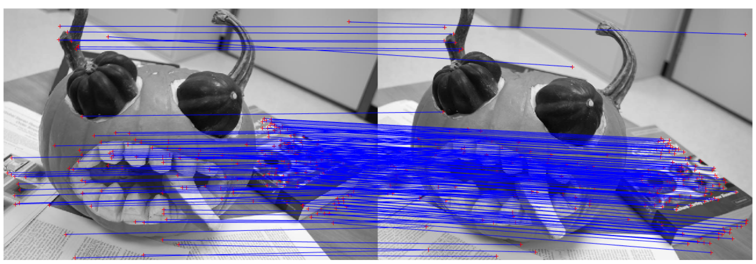
by implementing the file *distPointsLines.m* and took only the points that had a distance which was minor then the threshold. If i got a major number of inliers, I updated the best parameters.

I am attaching only the image of the pumkin but I tried with all of them.

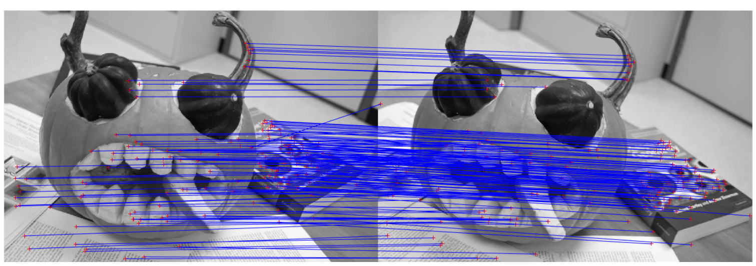
With a threshold of 5, the total best inliers count was 371 and I got an average inliers distance of 1.8660.



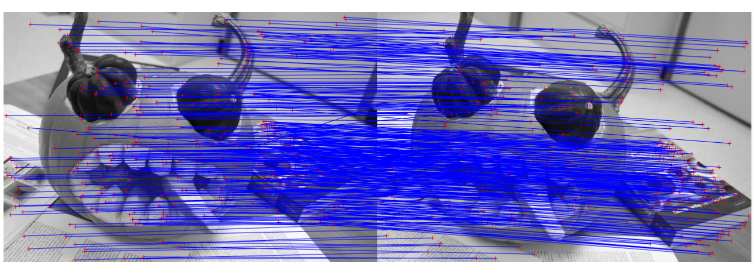
With a threshold of 2, the total best inliers count was 255 and I got an average inliers distance of 0.7437.



With a threshold of 1, the total best inliers count was 149 and I got an average inliers distance of 0.1382.



With a threshold of 20, the total best inliers count was 452 and I got an average inliers distance of 8.7112.



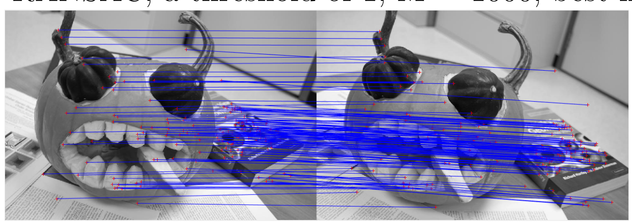
4.2 Adaptive Ransac

In this case we terminate RANSAC after M trials if we know that at least one of the samples of 8 points is free from outliers with a probability of p. By knowing that r is the best number of inliers until now divided for the number of points, N is 8 and M is the iteration, I used this formula to calculate p.

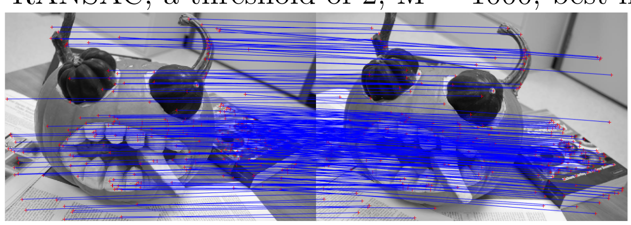
$$p = 1 - (1 - r^N)^M$$

I terminated the iteration if the probability p calculated was major or equal to 0.99.

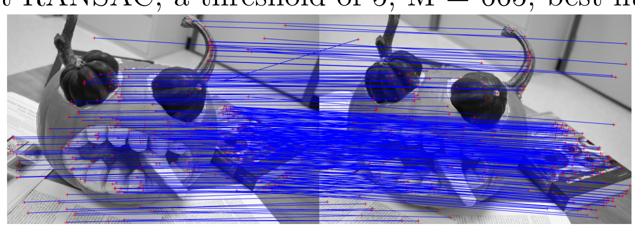
Inliers using 8-Point RANSAC, a threshold of 1; M = 1000; best num of inliers = 157.



Inliers using 8-Point RANSAC, a threshold of 2; M = 1000; best num of inliers = 291.



Inliers using 8-Point RANSAC, a threshold of 5; M = 665; best num of inliers = 357.



Inliers using 8-Point RANSAC, a threshold of 20; M = 90; best num of inliers = 440.