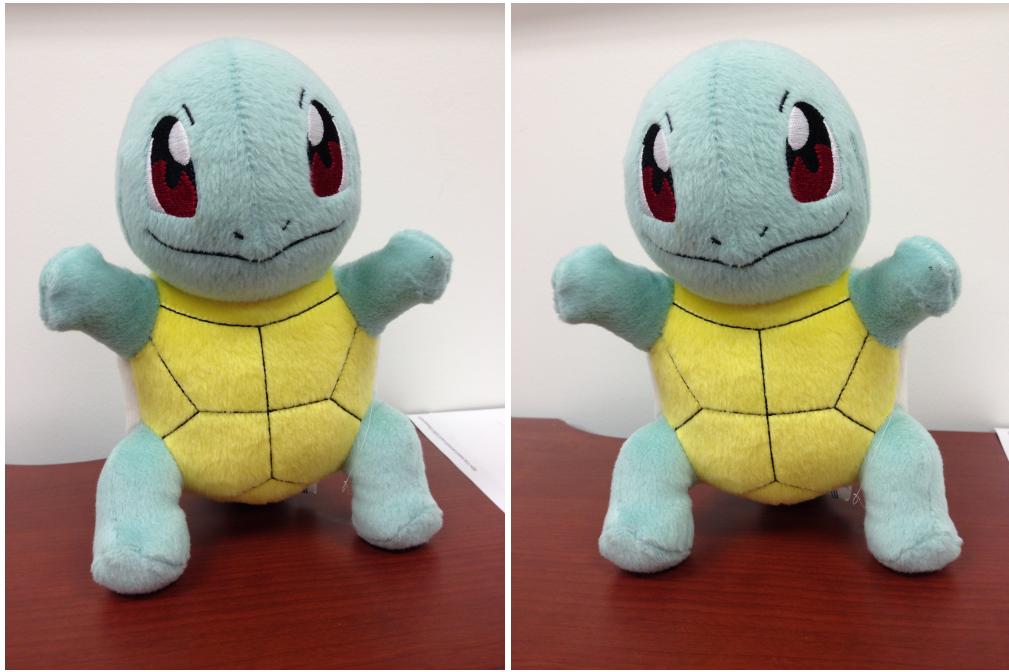


# CS 217 Homework 2

Zachary DeStefano, 15247592

Due Date: May 14, 2015



(a) Left Image

(b) Right Image

Figure 1: Original Pictures of Stuffed Squirtle

## Problem 1

The following are examples of running the SIFT matching code on my own objects

### Squirtle Example

Figure 1 shows the original pictures that I took of a stuffed Pokemon named Squirtle. Figure 2 show 50 random points found by running SIFT. Figure 3 shows some of the matches found using SIFT.

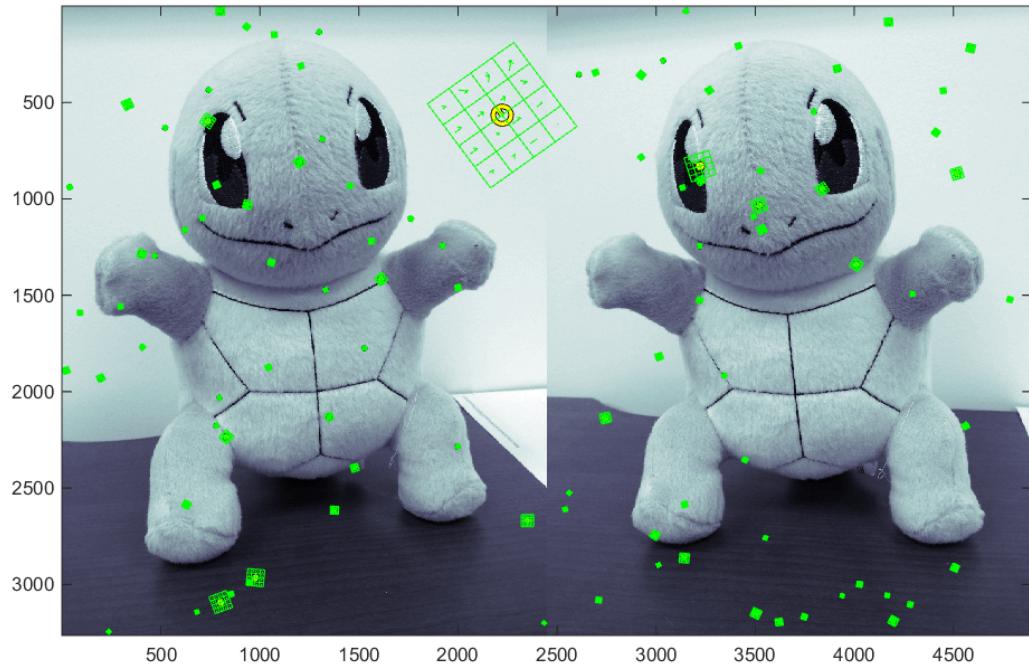


Figure 2: Squirtle Images With Points Marked

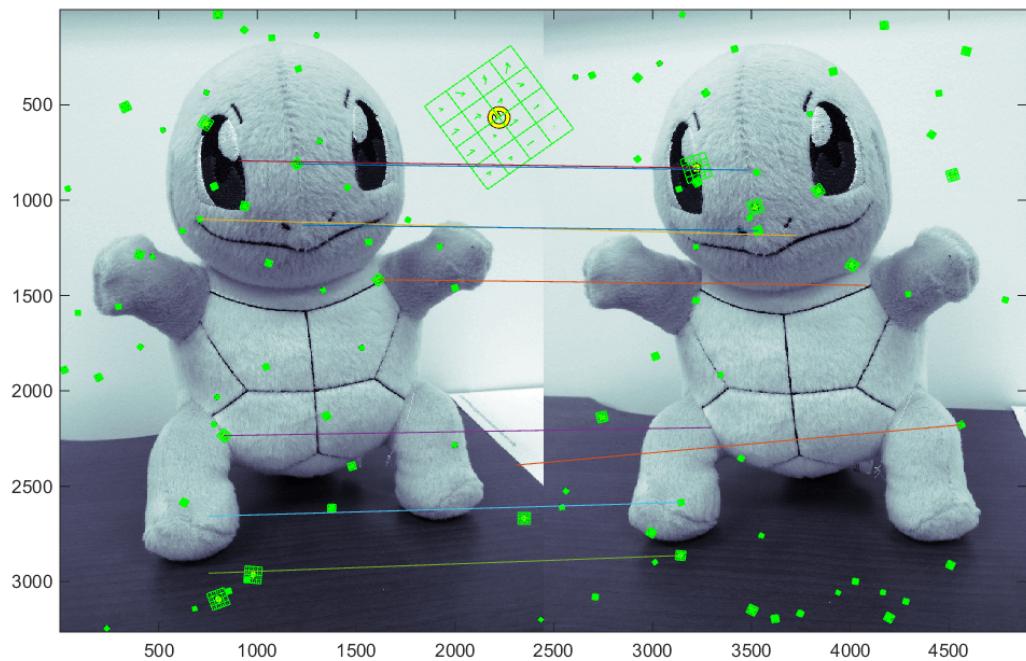
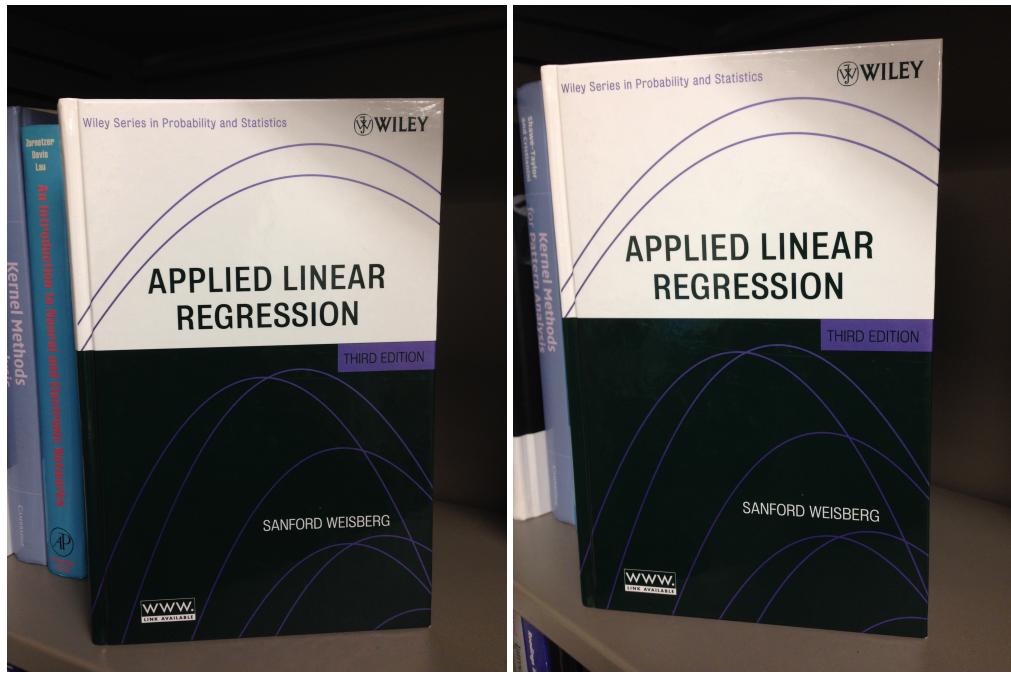


Figure 3: Squirtle Images With Points Marked and Some Matches Shown



(a) Left Image

(b) Right Image

Figure 4: Original Pictures of Textbook

## Book Example

Figure 4 shows the original pictures that I took of a textbook. Figure 5 show 50 random points found by running SIFT. Figure 6 shows some of the matches found using SIFT.

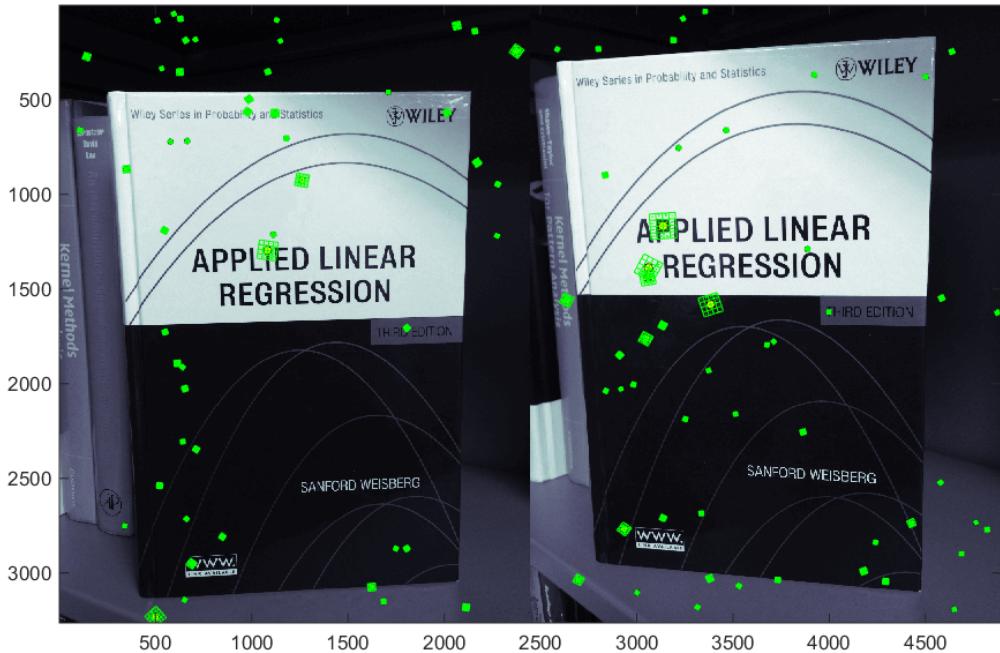


Figure 5: Textbook Images With Points Marked

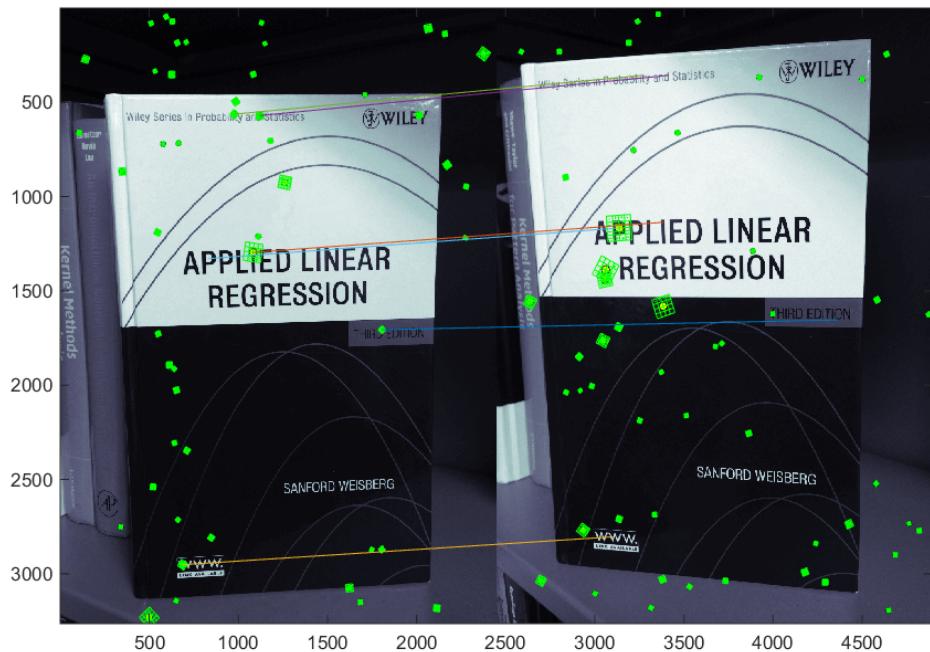


Figure 6: Textbook Images With Points Marked and Some Matches Shown



(a) Left Image

(b) Right Image

Figure 7: Original Pictures of Coffee Can

### Coffee Can Example

Figure 7 shows the original pictures that I took of a coffee can. Figure 8 show 50 random points found by running SIFT. Figure 9 shows some of the matches found using SIFT.



Figure 8: Coffee Can Images With Points Marked



Figure 9: Coffee Can Images With Points Marked and Some Matches Shown

**Code for Problem 1**

The code for this problem is in the folder as prob1script.m

## Problem 2

For problem 2, I used the same images as the previous problem and I used points chosen by myself as well as points chosen through SIFT. The first part of this write-up shows the results when I used my own parts. The second part of the write up shows the results when I used the points found through SIFT. With my own points, I made sure they were selected in the same order in both images so the matches had the same index. With the SIFT points, I used the matching results from running `ubcmatch`. In both cases, I decided to test each matching pair using RANSAC in order to get a final set of pairs.

### Part A, My Points

Figures 10, 11, and 12 show the points that I chose in each image. After running my own implementation of RANSAC, I got the correspondences that are shown in figures 13, 14, and 15.

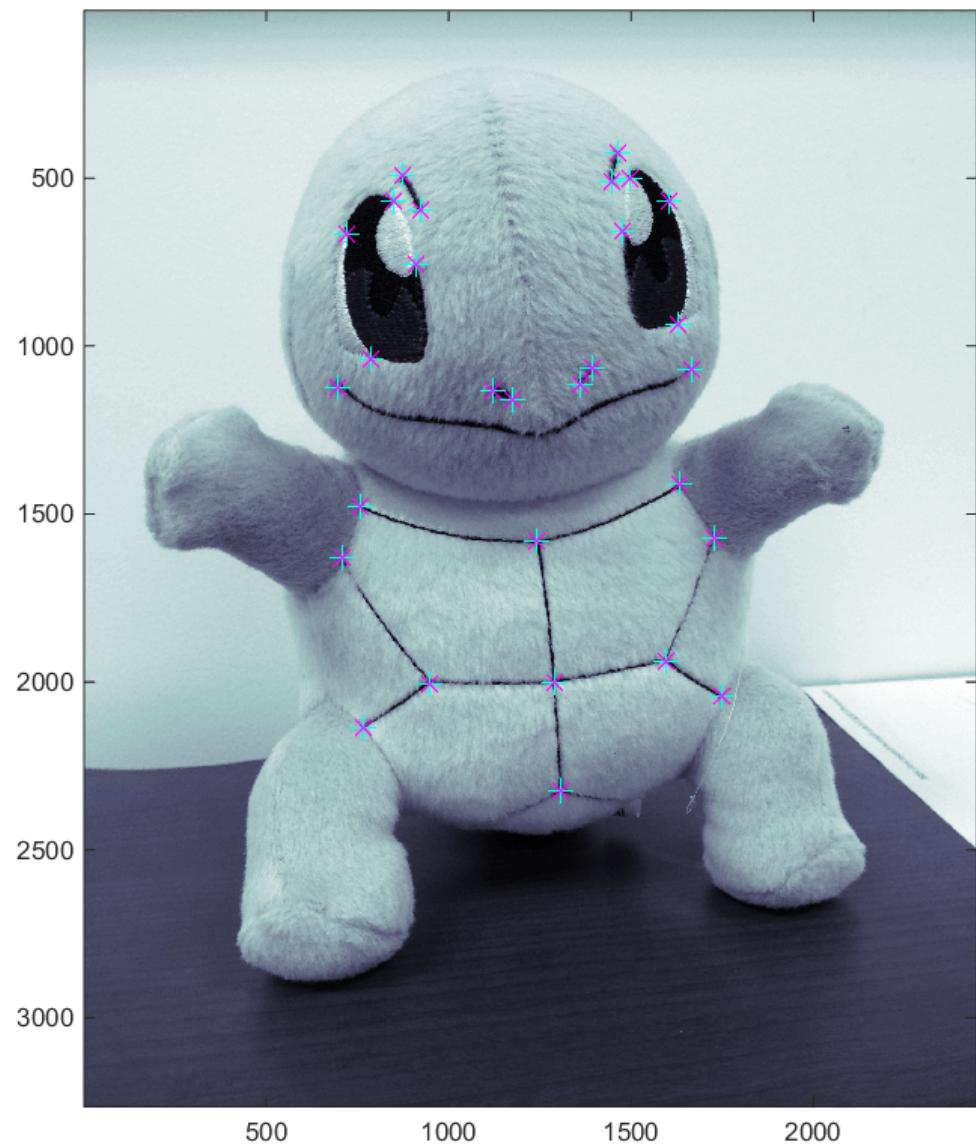


Figure 10: Squirtle Image with Points Marked

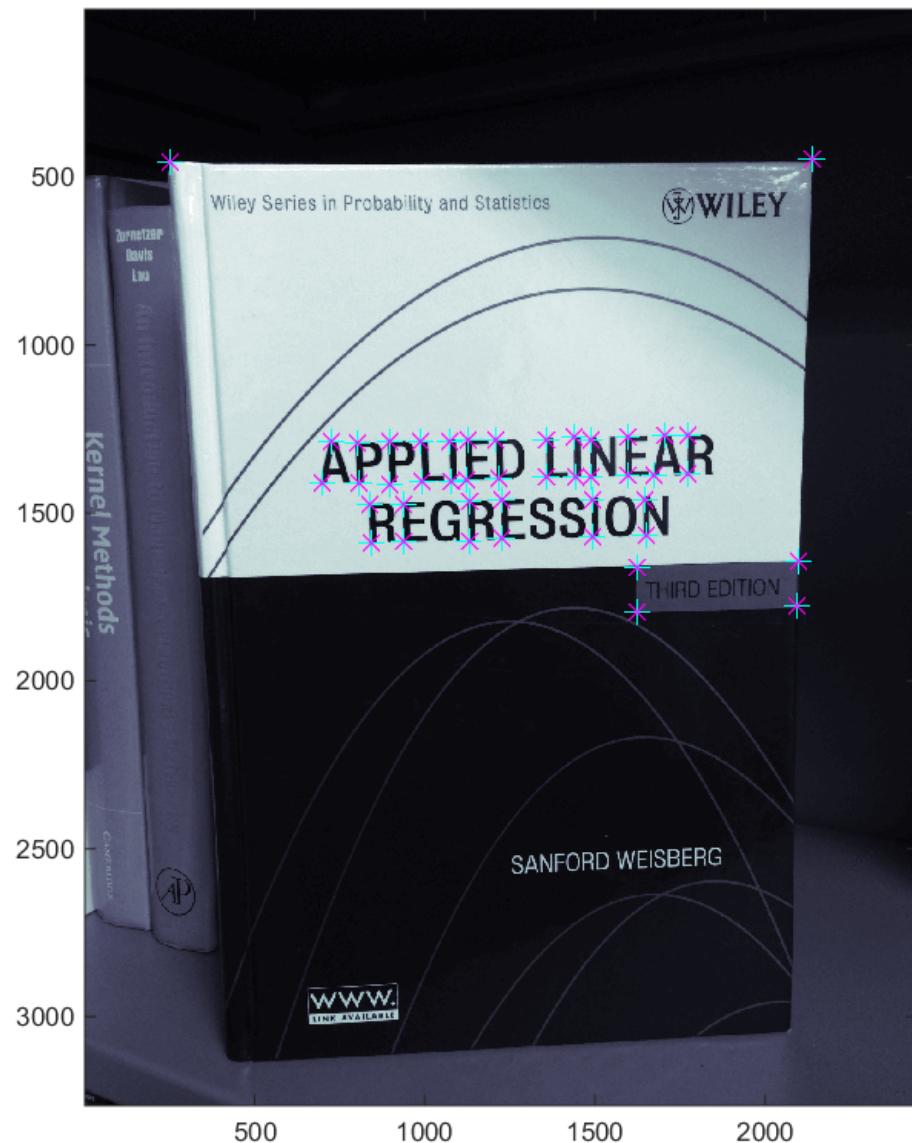


Figure 11: Textbook Image with Points Marked



Figure 12: Coffee Can Image with Points Marked

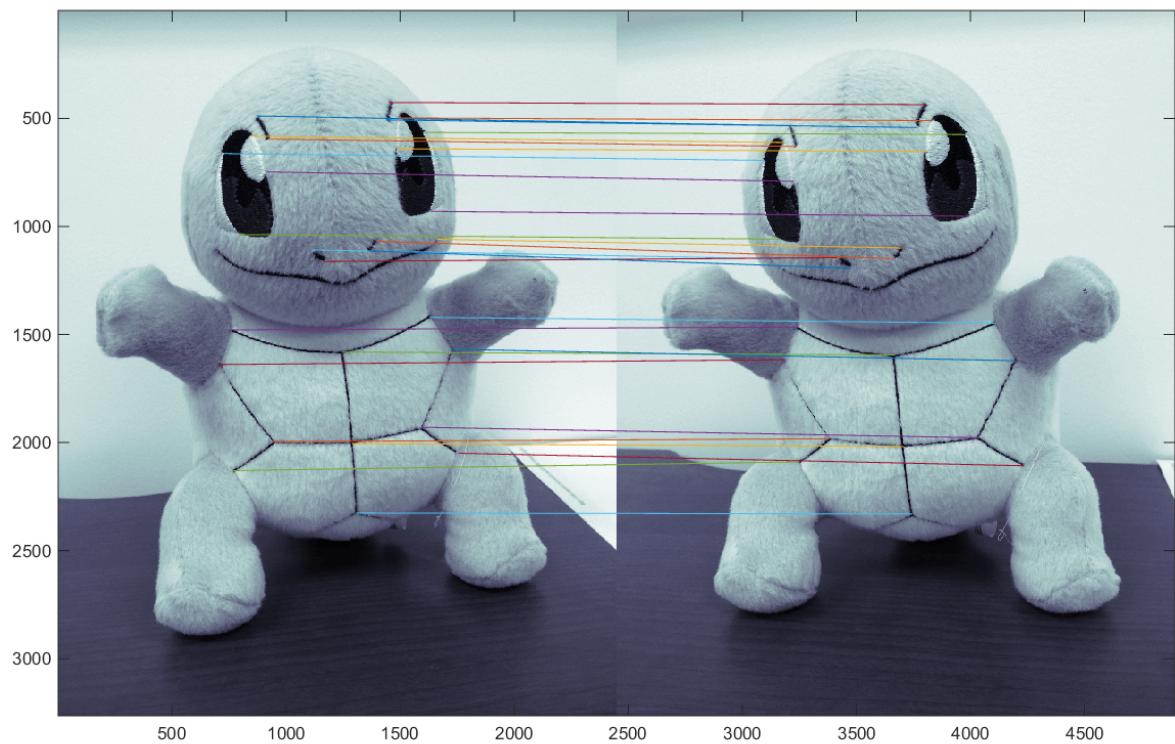


Figure 13: Squirtle Images with matching points

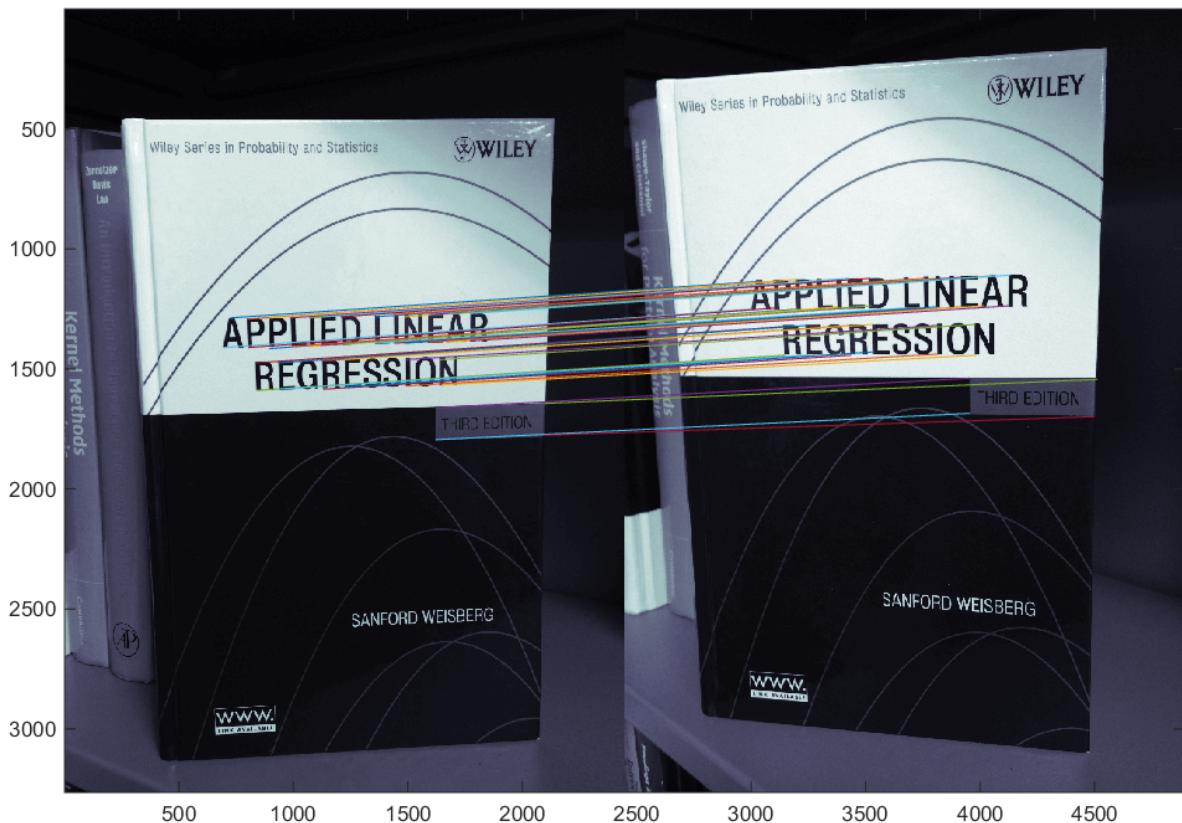


Figure 14: Textbook Images with matching points



Figure 15: Coffee Can Images with matching points

**Part B, My Points**

With the squirtle image, the epipole for the left image lies slightly above the epipole for the right image, suggesting that the left image was taken from a slightly raised viewpoint as compared to the right image. With the coffee can image, the epipole of the left image is below the epipole for right image, suggesting that it was taken from a lowered viewpoint.

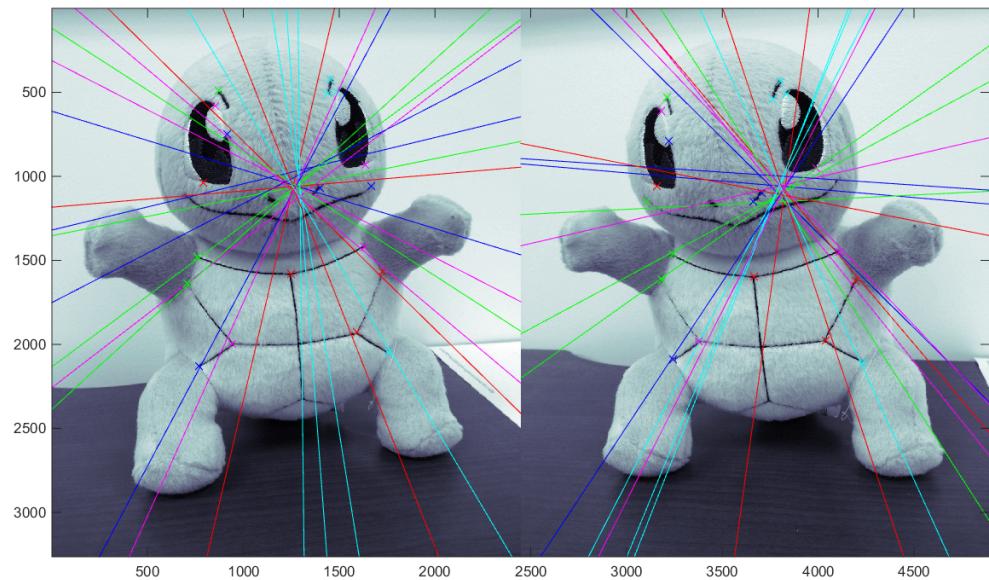


Figure 16: Squirtle Image with Epipolar Lines and their points

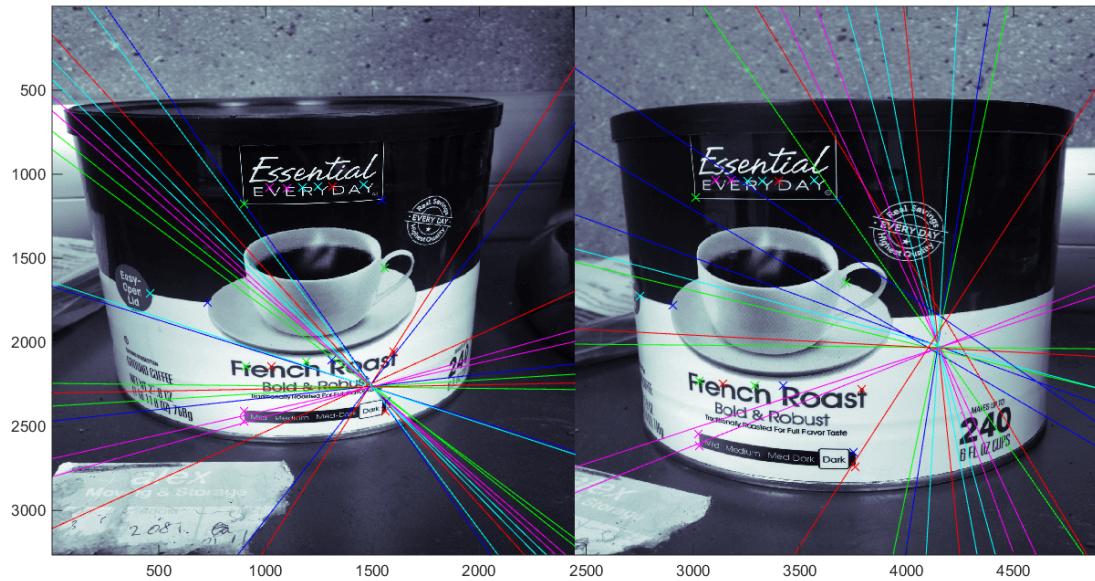


Figure 17: Coffee Can Image with Epipolar Lines and their points

### Part C, My Points

The estimate for F is quite stable. I decided to compute it multiple times and find the standard deviation of the entries in the F matrix. I then summed the standard deviation.

For squirtle, the sum was near zero and 4 computations of the F matrix were done.

For the textbook, the sum was 0.9567 and 6 computations of the F matrix were done.

For the coffee can, the sum was 1.0140 and 6 computations of the F matrix were done.

The best way to make F more robust would have been to add more points and then see if they correspond. With more data, the estimates get more robust.

### Part A, SIFT points

Figures 18, 19, and 20 show the points chosen by SIFT in each image. I then ran my own implementation of RANSAC and only used the SIFT pairs that matched well. Afterwards, I got the correspondences that are shown in figures 21, 22, and 23.

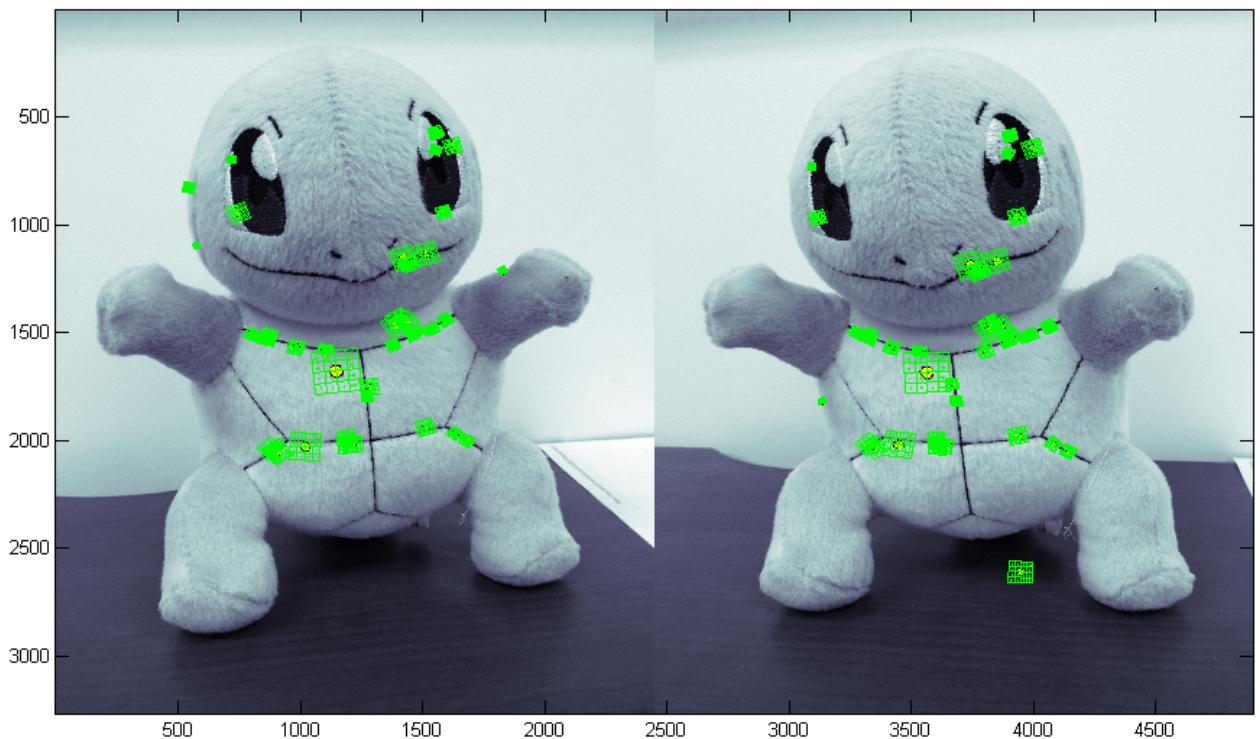


Figure 18: Squirtle Image with Points Marked

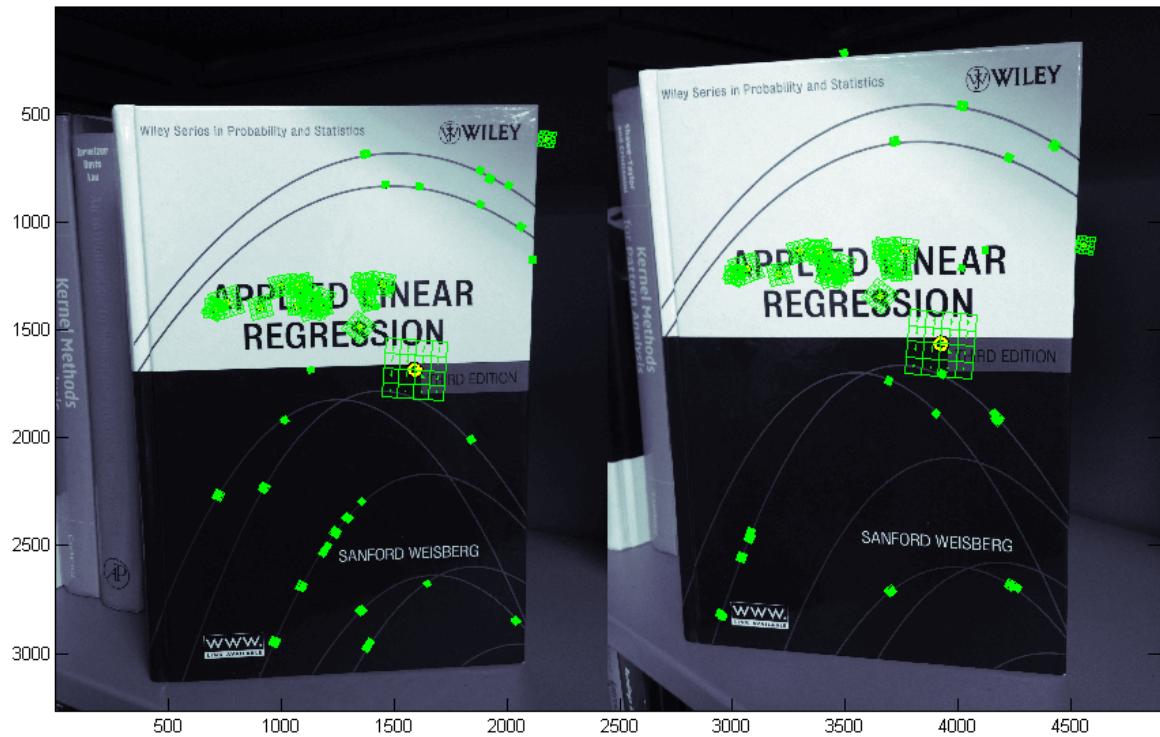


Figure 19: Textbook Image with Points Marked



Figure 20: Coffee Can Image with Points Marked

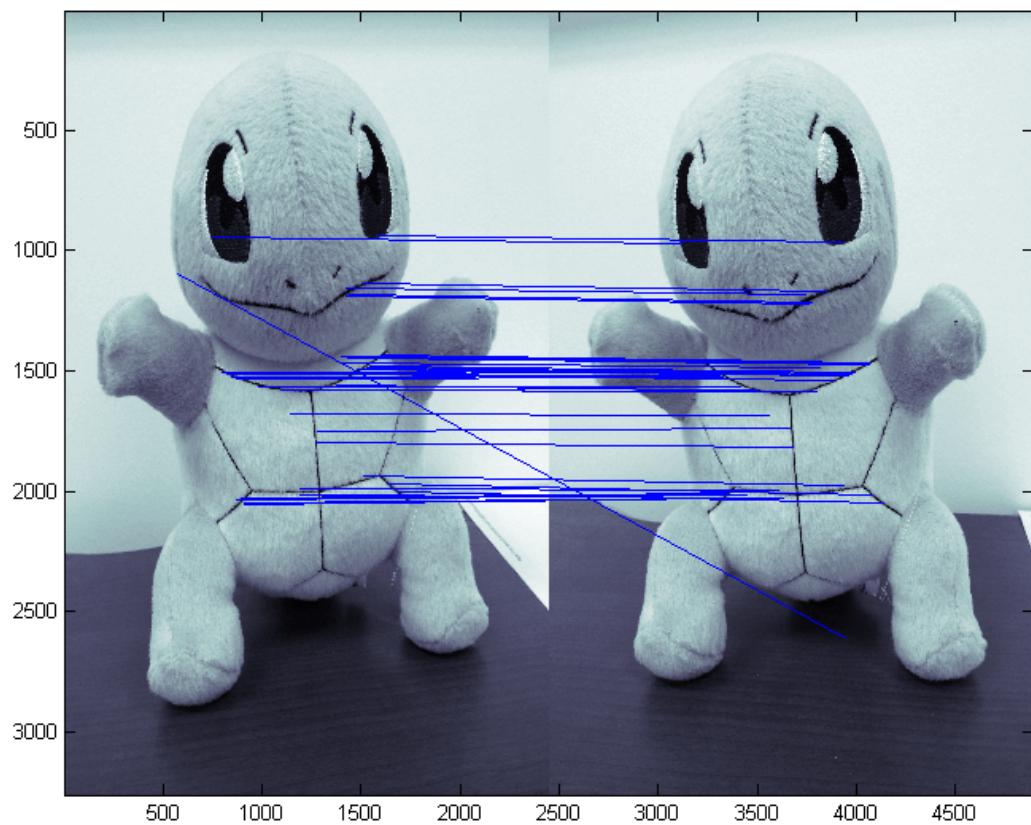


Figure 21: Squirtle Images with matching points

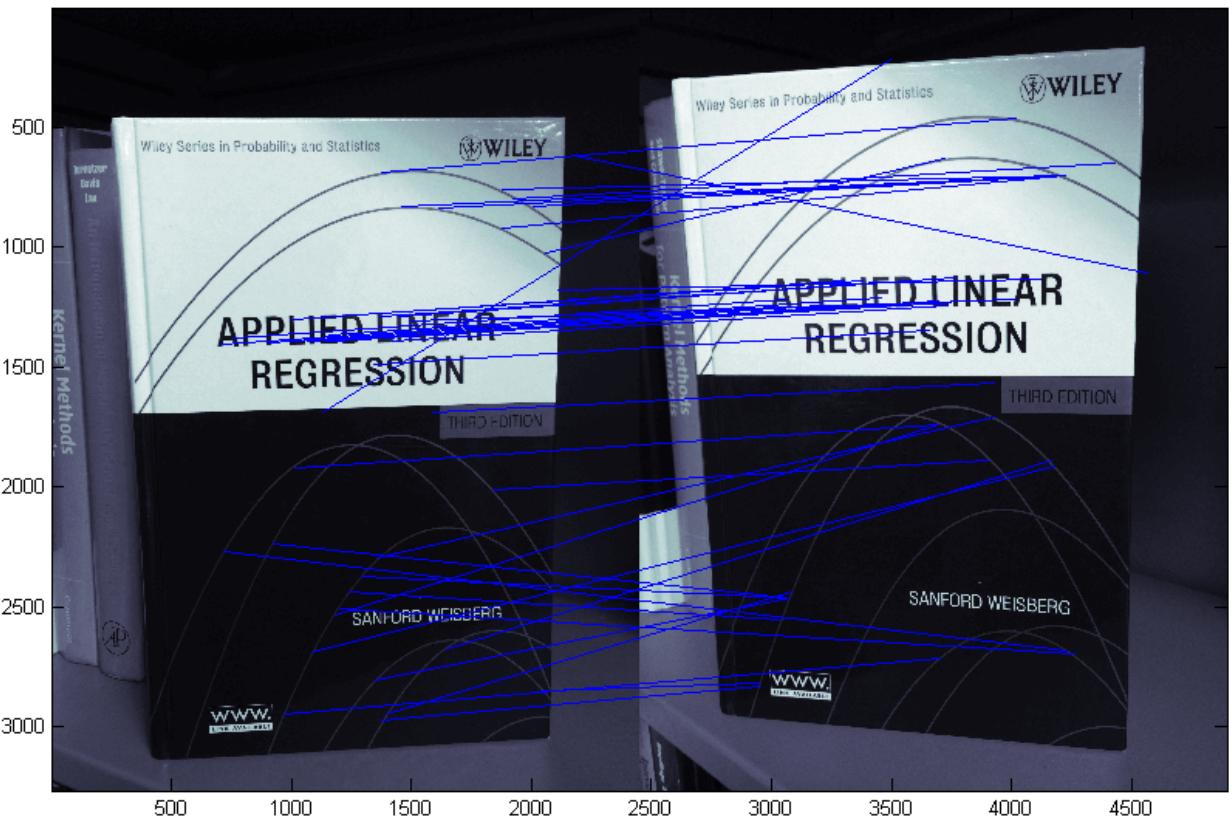


Figure 22: Textbook Images with matching points

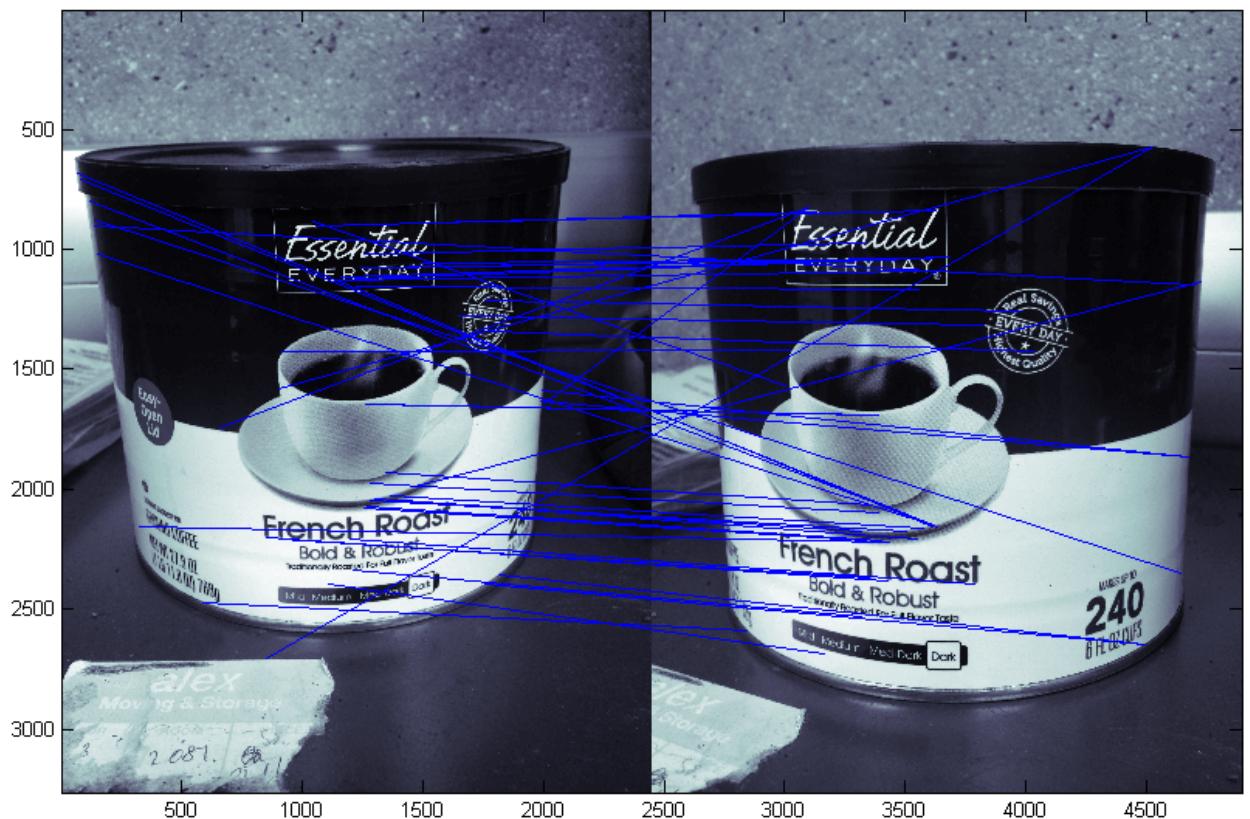


Figure 23: Coffee Can Images with matching points

**Part B, SIFT points**

With the squirtle image epipoles for these points, they are nearly in the same spot. This would suggest that I did not move the camera much between the two images, which is consistent with what happened. With the coffee can image, there is a clear epipole in the right image but it is not clear in the left image where the epipole is located. This suggests that there were some inaccurate picks of points in this case.

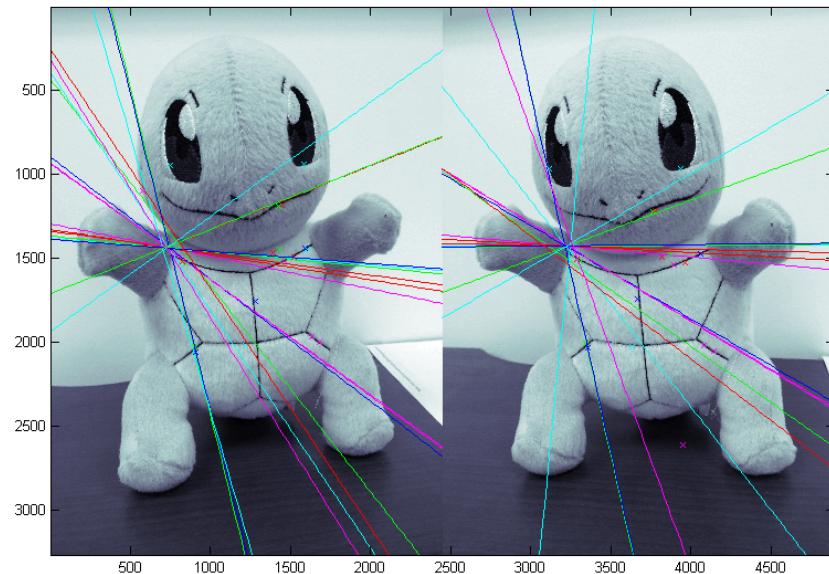


Figure 24: Squirtle Image with Epipolar Lines and their points

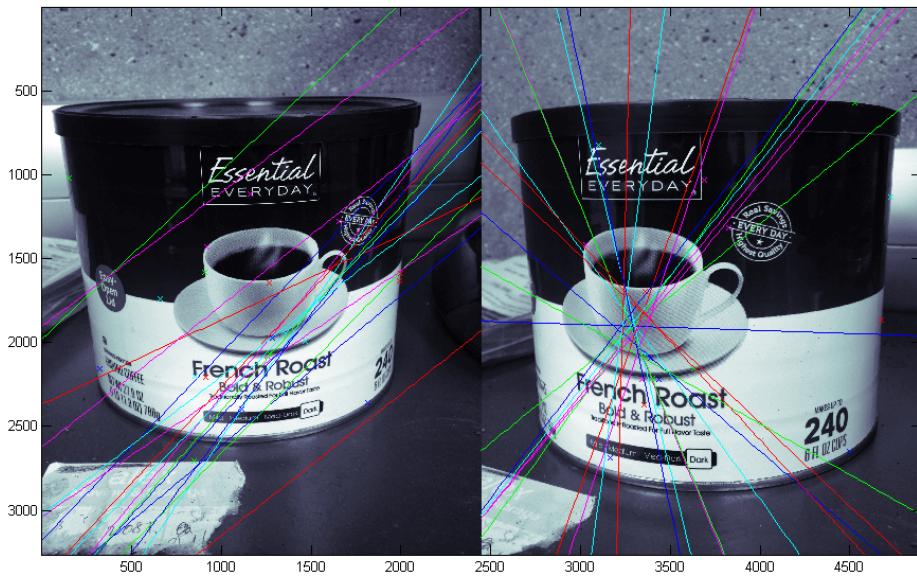


Figure 25: Coffee Can Image with Epipolar Lines and their points

### **Part C, SIFT points**

I repeated the procedure I did for my points to see the stability of F.

For squirtle, the sum was 1.0191 and 4 computations of the F matrix were done. I decided to increase the number of points to see if that would make it more stable. After using 100 points in the RANSAC computation, the sum of the standard deviations of the entries was 0.0048 thus it had nearly the stability of my other method.

For the textbook, the sum was near zero and 6 computations of the F matrix were done, thus it was more stable than my points.

For the coffee can, the sum was near zero and 6 computations of the F matrix were done, thus it was more stable than my points.

With these points, the estimates of F proved to be quite robust. As it turns out with the squirtle image, my hunch that when you add more points it becomes more robust proved correct.

### **Code for Problem 2**

The code for this problem is attached as prob2script.m

### Problem 3

After running everything, these are the two rotation matrices:

$$\begin{pmatrix} 0.9920 & 0.1264 & -3.7932\text{e-}5 \\ -0.1264 & 0.9920 & -9.8549\text{e-}5 \\ -2.5167\text{e-}5 & -0.0001 & -0.9999 \end{pmatrix}$$

$$\begin{pmatrix} -0.9920 & -0.1264 & -1.1563\text{e-}4 \\ 0.1264 & -0.9920 & 1.8543\text{e-}5 \\ 1.1704\text{e-}4 & -7.748\text{e-}6 & -0.9999 \end{pmatrix}$$

The two position vectors are as follows

(7.6779e-5, 4.0003e-5, 0.9999)

(-7.6779e-5, -4.0003e-5, -0.9999)

According to these results, it appears that the camera was moved back (or forward) and then rotated to make the second image. This appears consistent with what the images look like. I would have expected slightly more movement in the  $x, y$  direction though based on my perception of the image, but perhaps most of the translational movement was in the  $z$  direction. That is quite possible since there are only two camera positions we are considering. Figure 18 shows the image correspondences found through SIFT that were used to compute the above matrices and vectors.

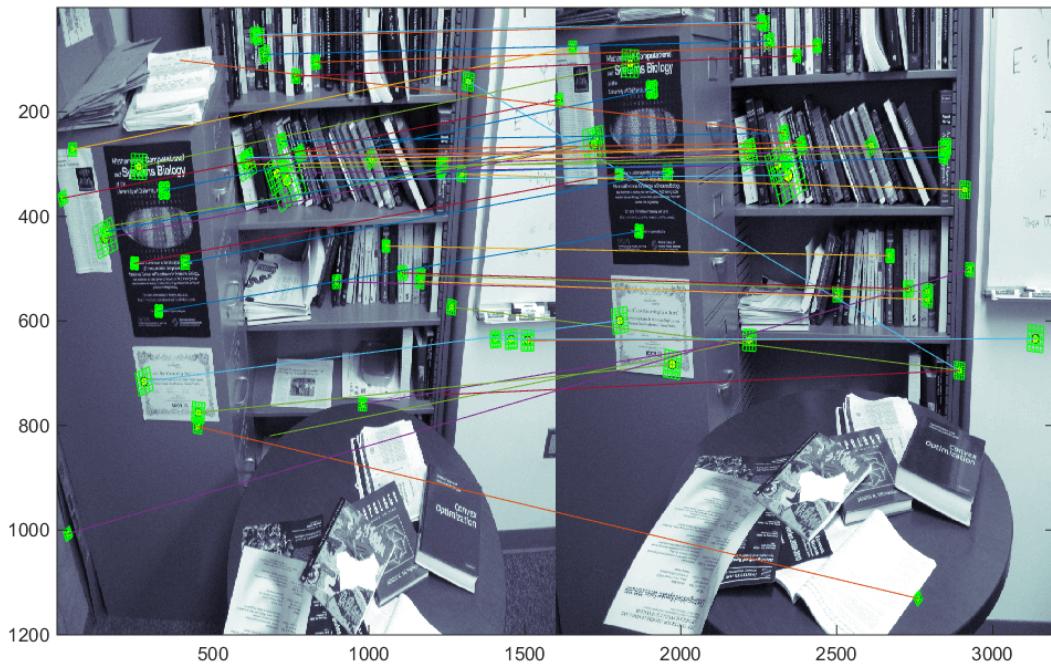


Figure 26: Image Correspondences

The code to complete the problem is attached as prob3script.m