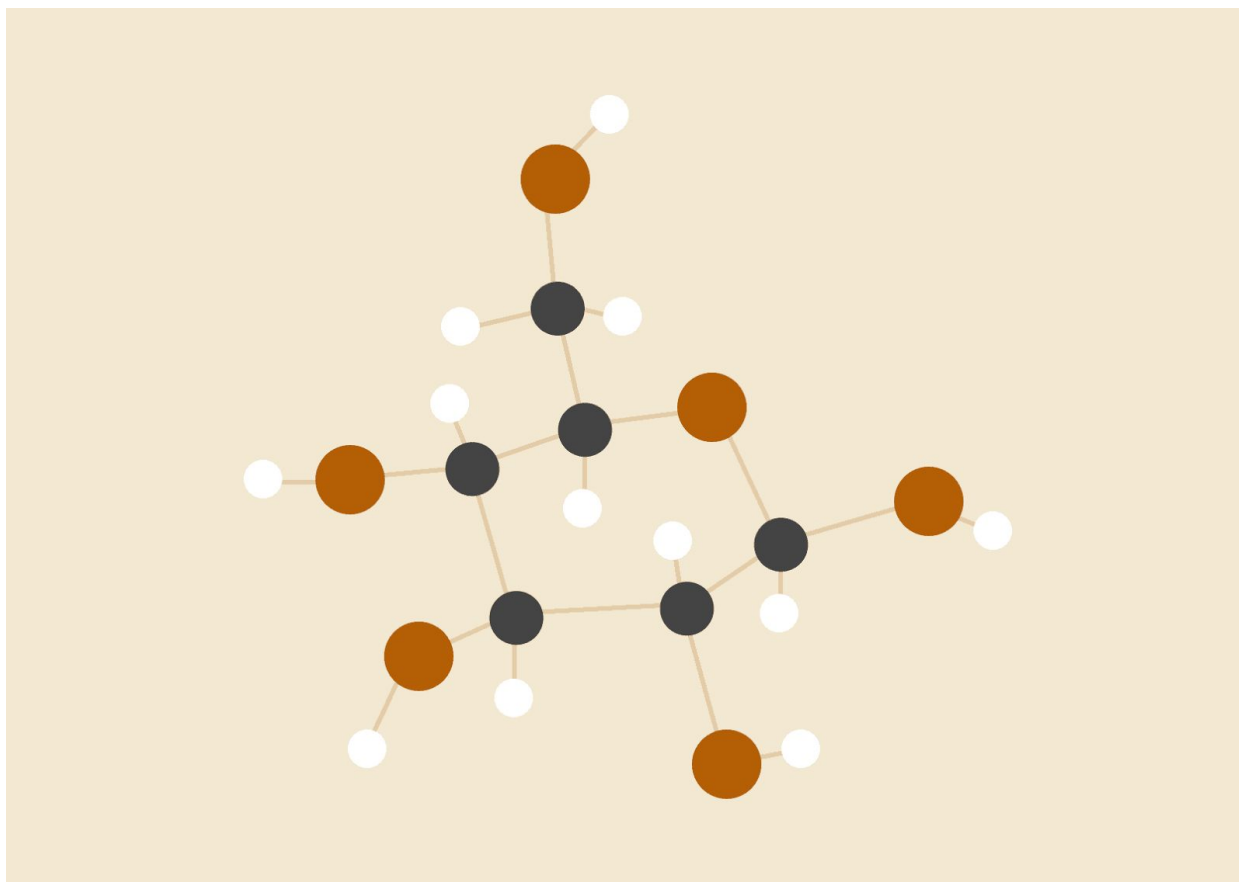


HOMework 3 REPORT

Homography and fundamental matrix estimation

Computer Vision and Image Processing CSE 573



Jayant Solanki

UBIT Name: jayantso

Person Number: 50246821

Fall 2017 CSE

INTRODUCTION

This assignment report talks about the implementation and results of the application of the Homography matrix in stitching pairs of images taken from same camera at same position but at different rotation along the camera axis. Report also talks about the estimation of 3-D coordinates from the 2-D coordinates of an object where stereo images have been taken of it by two camera using Epipolar Geometry and Fundamental Matrix.

In the first part, using pairs of two consecutively taken images of same object, we estimate the homography matrix using the keypoints features detected on both the images and applying RANSAC algorithm to get the inliers. The Homography matrix is then used for stitching both image to generate Panorama.

In the second part of the assignment, we generate the fundamental matrices of a pair of images using the groundtruth points and inliers generated during the first part. We use normalised and unnormalised approach to generate the matrix. We also compare the result of the both approach. Later on we also approximate the 3-D coordinates of the object in the pairs of image using Triangulation method. All the coding has been done in the MATLAB environment and proper references have been highlighted in the report.

IMPLEMENTATION

PART 1

Section 1.0: Pair of images of the **UTTOWER** have been loaded and converted into double data-type and grayscale image format for later pre-processing.

Section 2.0: Using the sample code provided for the Harris detector, feature points in both images have been detected are shown below in the **Figure 1.1.1 and Figure 1.1.2**

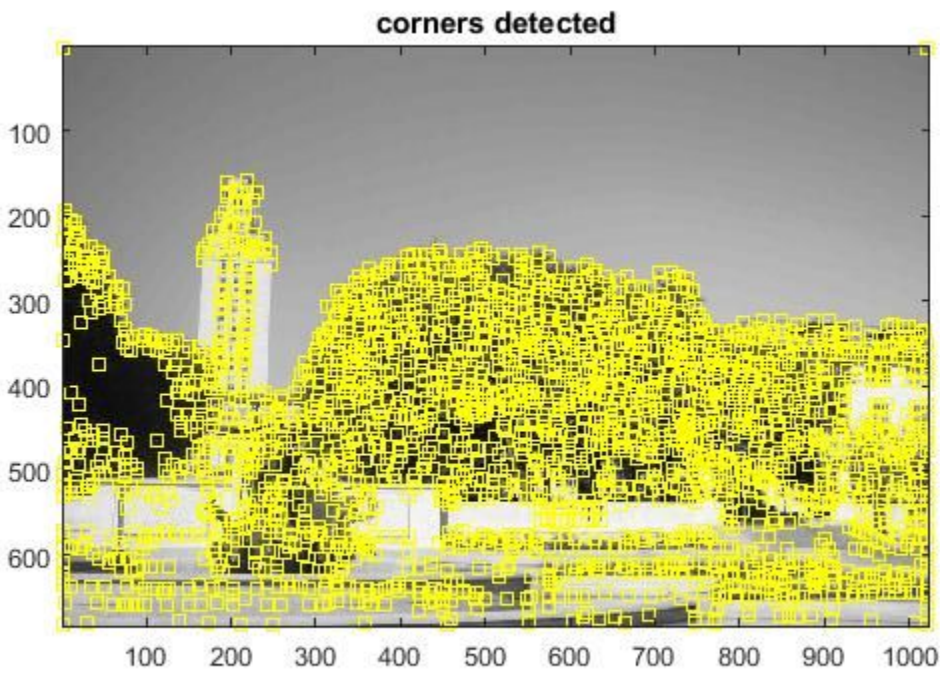


Figure 1.1.1 UT Tower left.jpg

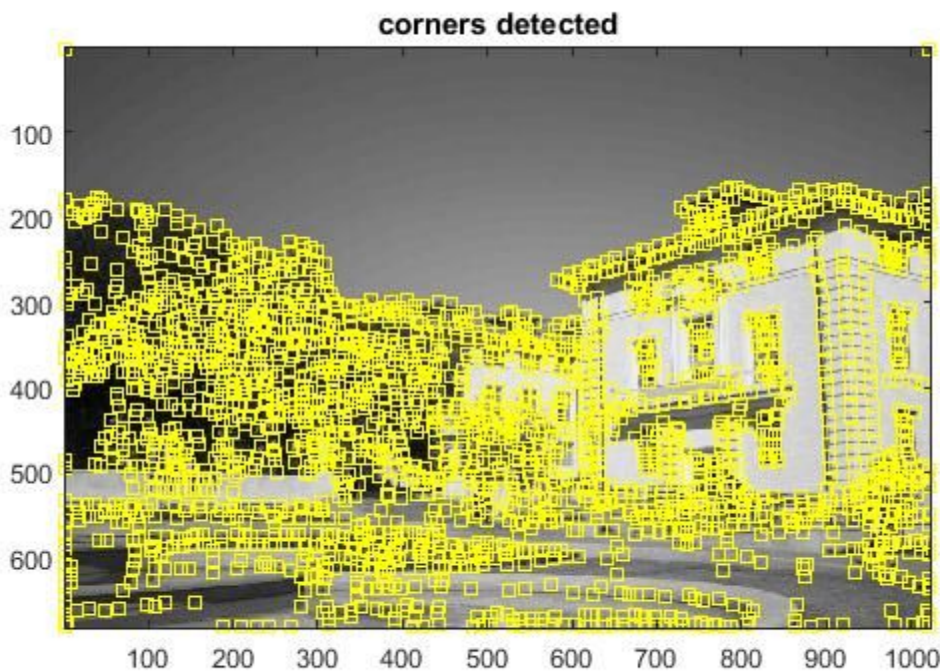


Figure 1.1.2 UT Tower right.jpg

Section 3.0: After storing the feature points of both images in 2-D arrays I found the local neighbourhood pixels for the each feature points and created **keypoint feature**

matrix for each feature points. However I ignored the feature points at the image boundaries as they were hampering the generation of the keypoint feature matrix. We tried different dimensions of neighbourhood matrix. Comparative study is shown in the **table 6.1**. From the table we can see that on increasing the neighbourhood pixels dimension, distance threshold has to be increased in order to get the good amount of putative matches. Nevertheless one of the downside of increased dimension is that a keypoint feature starts to interfere with other neighbourhood keypoint features. This leads to overlapping and one-to-many mapping. This also hampers with RANSAC process. In the RANSAC, the error threshold has to be increased in order to compensate for this mismatch. In order to balance both side, we found **5X5** dimension with distance threshold **3** and RANSAC threshold **25** is the best parameter covering all the images. Harris threshold is set to 0.005 and sigma 2. The 5X5 neighbourhood was later flattened to 25X1 column vector. Also the keypoint features had their pixels values normalised with zero mean and unit standard deviation.

Section 4.0 and 5.0: In order to get the putative matches we used the dist2.m script to find the euclidean pixels value distance for every keypoint features of image to the every keypoint features of other image. The euclidean matrix has been set to 3 for filtering the unnecessary matches. To check the verity of this I calculated the standard deviation and mean for the output returned from the dist2.m script. For the 5X5 distribution, the mean was found to be ~150 and std ~40. This meant that the keypoints features were correctly calculated and the output distribution was very broad and after the application of the distance threshold good amount of putative matches can be found with good number of inliers. Putative matches have been shown in the **Figure 1.4.1** for the UTTower.

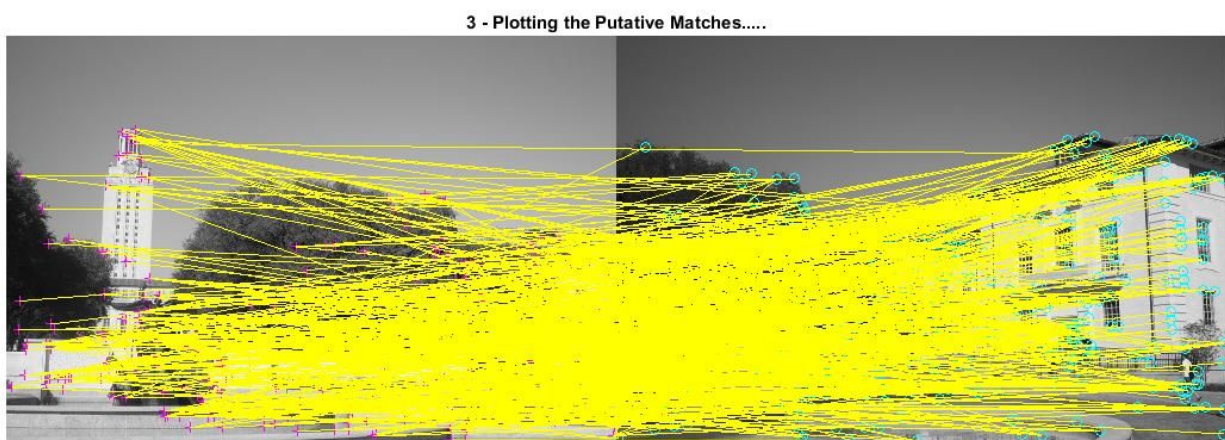


Figure 1.4.1 Putative matches, **matches count is 356**

Do note that I could have fine tuned the distance thresholding by calculating the Top 200-300 putative matches with lower distance thresholds and further filtering the distance thresholds of one keypoints pairing by taking the ratio of their distances and selecting those pairs only whose ratio were less than 0.5. But the current implementation was all right enough to use these features too.

Section 6.0: The Putative matches had both inliers and outliers in it as you can see in the above **Figure 1.4.1**. Outliers are those lines which start from the same keypoint feature on left image and map to non-corresponding keypoint features of the other image reverse of what inliers do. Outliers can be seen at the top point of the UTTower of the left image. In order to get the inliers from the putative matches we wrote a simple RANSAC algorithm, stated below:

1. Select 4 random putative pairs from the putative matches
2. Generate the Homography matrix H by using the formula mentioned in the HW3 document.
3. Transform the left image coordinates of the putative matches using the H matrix.
4. Find the error between the transformed left image coordinates and the actual coordinates of the putative matches in the right image.
5. Below a error threshold get the indices of the coordinates in the putative matches. This will be inliers.
6. Store the indices of the inliers when inliers count is maximum.
7. Repeat above steps for 20000 times.
8. Return the Homography matrix, best count of the inliers and their indices.

Inliers have been displayed below in the **Figure 1.6.1**, **Residual error is 2.377**

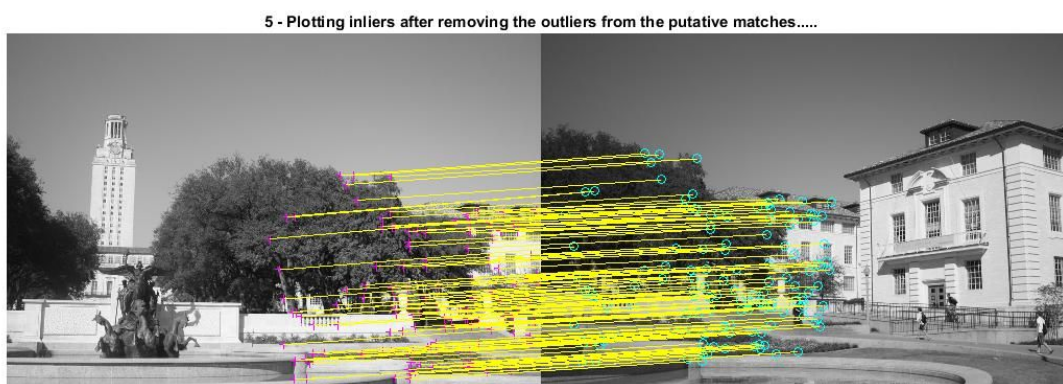


Figure 1.6.1 Inliers in the UTTowers images **Inliers count is 61**

Section 7.0: Image on the left side is transformed using the Homography matrix by applying the *maketform* and *imtransform* functions. Its shown in the **Figure 1.7.1**



Figure 1.7.1, left.jpg

Section 8.0 and 9.0: Composite image of both the image of UT Towers is displayed below in the **Figure 1.8.1**

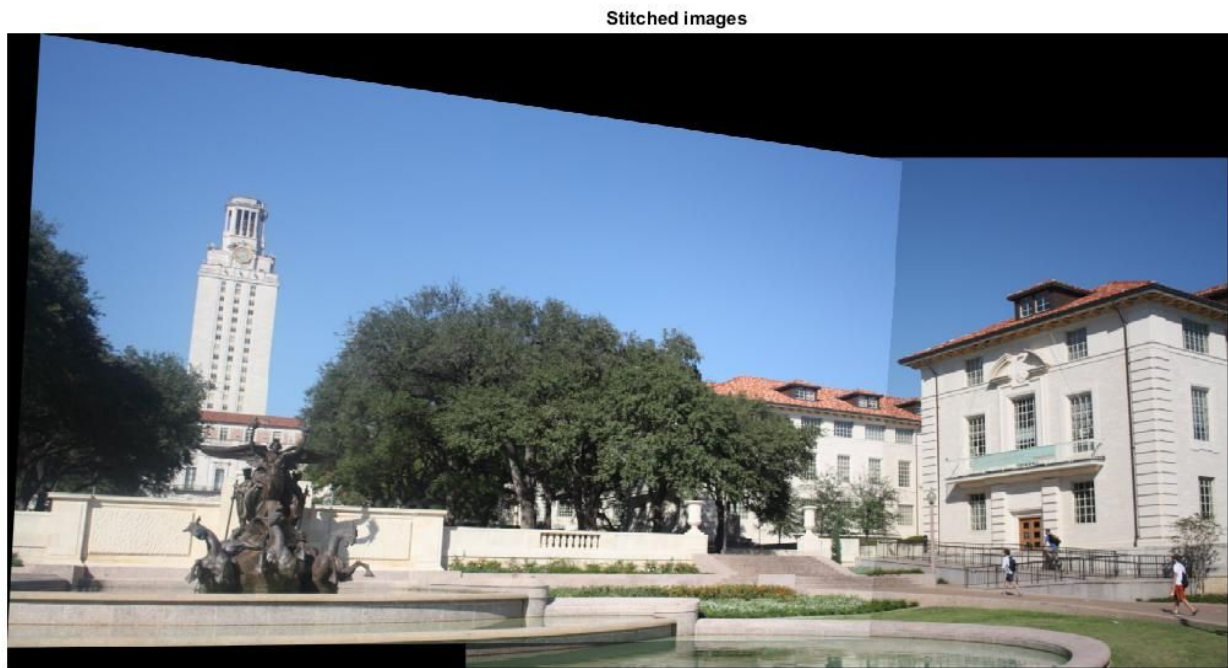


Figure 1.8.1

Table 6 .1

RANSAC at error-threshold = 25 and iterations = 20000								
Sr. No	Neighbourhood-Size	harris-Threshold	distance-Threshold	putative-Matches	inliers-Count	residual-Error	run-time (sec)	remark
1	3X3	0.05	3	35648	185	7.799	66.025	failed
2	3X3	0.05	0.3	313	17	0.923	28.843	pass
3	3X3	0.005	3	45000	memory error	NaN	NaN	failed
4	3X3	0.005	0.3	1002	13	7.037	11.282	failed
5	3X3	0.005	0.05	29	6	3.356	7.499	failed
6	5X5	0.05	3	356	61	2.377	9.507	pass, Optimum
7	5X5	0.05	0.3	1	1	NaN	1	failed, few inliers

8	5X5	0.005	3	892	129	2.65	13.69 3	pass
9	5X5	0.005	0.3	1	1	NaN	5	failed, few inliers
10	7X7	0.05	3	39	22	3.772	8.24	pass
11	7X7	0.05	0.3	NaN	NaN	NaN	1	failed, no matches
12	7X7	0.005	3	61	38	2.097	8.619	pass
13	7X7	0.005	0.3	NaN	NaN	NaN	1	failed, no matches
14	9X9	0.05	3	12	9	3.155	7.761	pass
15	9X9	0.05	0.3	NaN	NaN	NaN	1	failed, no matches
16	9X9	0.005	3	22	18	2.653	8.297	pass
17	9X9	0.005	0.3	NaN	NaN	NaN	1	failed, no matches

Bonus Section:

1 - Multiple image stichings:

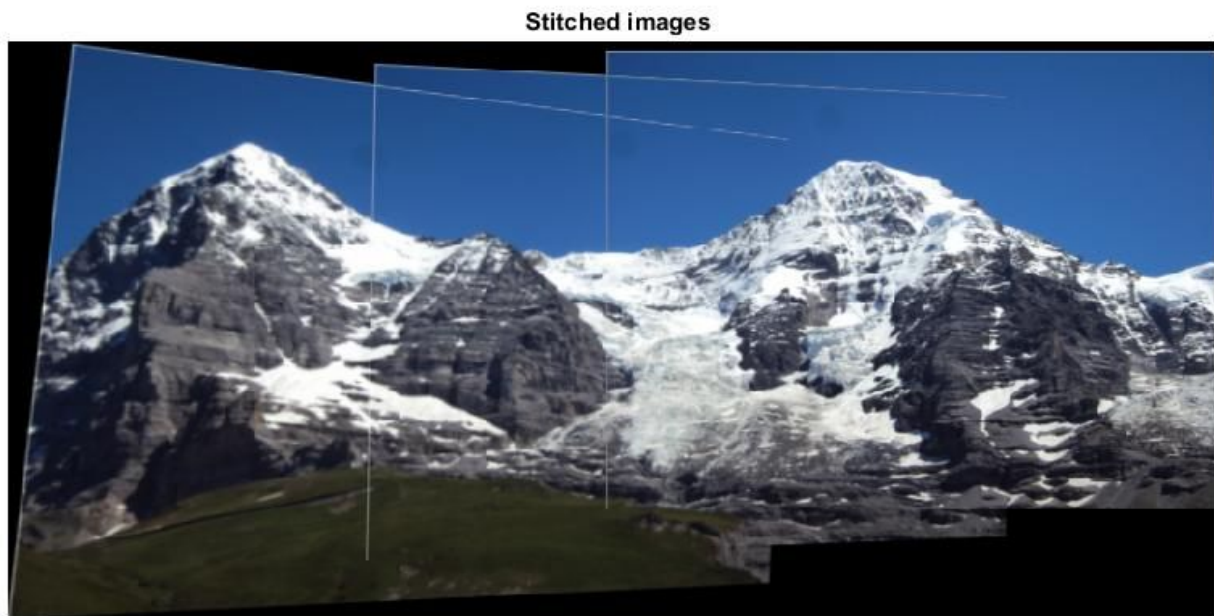


Figure: hill

Stitched images



Figure: Pier



Figure: Ledge

PART 2

Section 1: Using the `sample_code.m` script the house images are loaded along with their ground truths.

Section 2, 3: Epipolar lines on the images and Residual errors for both Ground truths and Ransac data is provided below.

Table 2.1

Sr. No	Matrix Algorithm	putative matches	inliers_count	residual_error(image1)	residual_error(image2)
1	Normalised	Ground truths	-	20.219	17.158
2	Normalised	HARRIS+RANSAC	18	0.93	0.581
3	Unnormalised	Ground truths	-	16.498	26.401
4	Unnormalised	HARRIS+RANSAC	18	0.657	0.44

Plotting the epipolar lines on the Ground truths in First Image

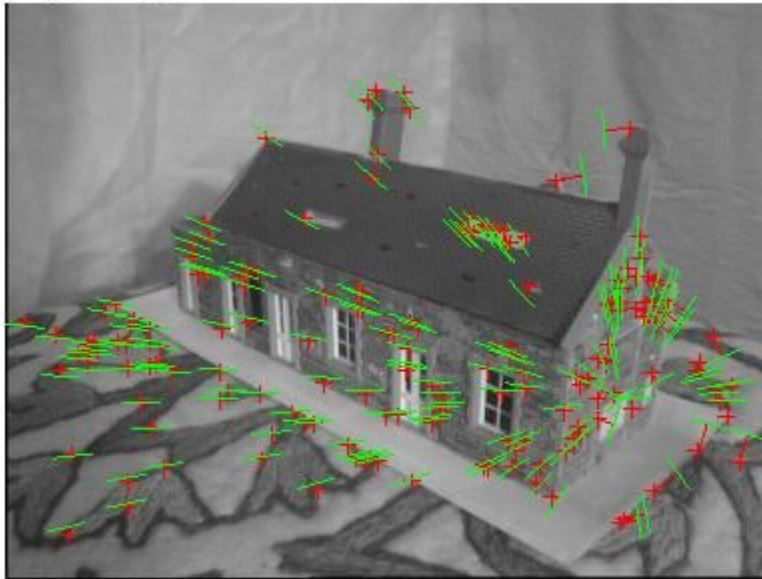


Figure 2.2 Image 1 Ground Truth, unnormalised

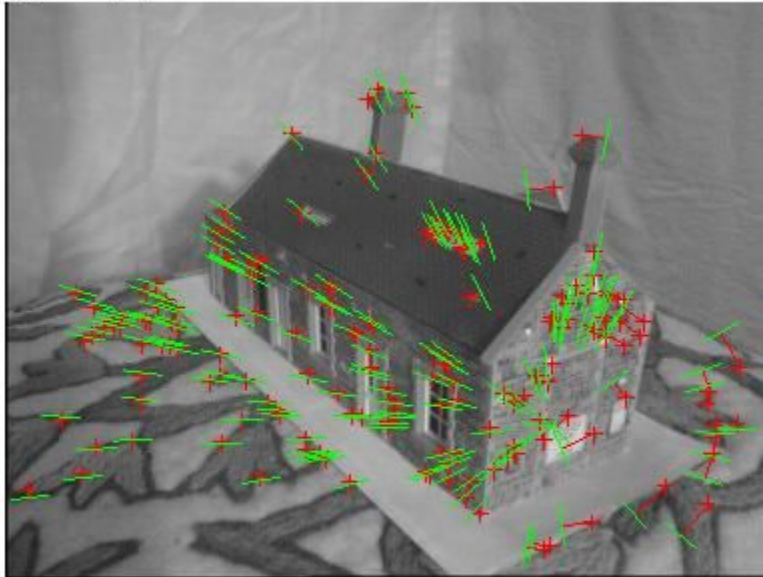
Plotting the epipolar lines on the Ground truths in Second Image

Figure 2.2 Image 2 Ground Truth, unnormalised

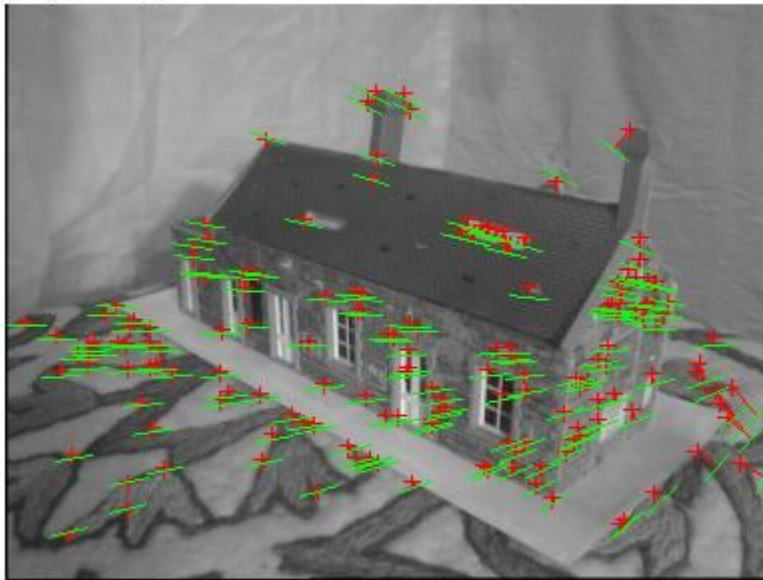
Plotting the epipolar lines on the Ground truths in First Image

Figure 2.3 Image 1 Ground Truth, normalised

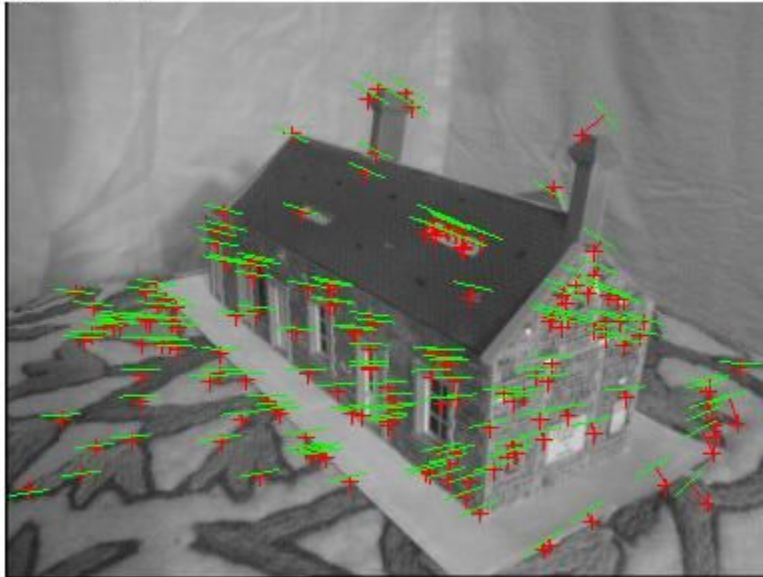
Plotting the epipolar lines on the Ground truths in Second Image

Figure 2.4 Image 2 Ground Truth, normalised

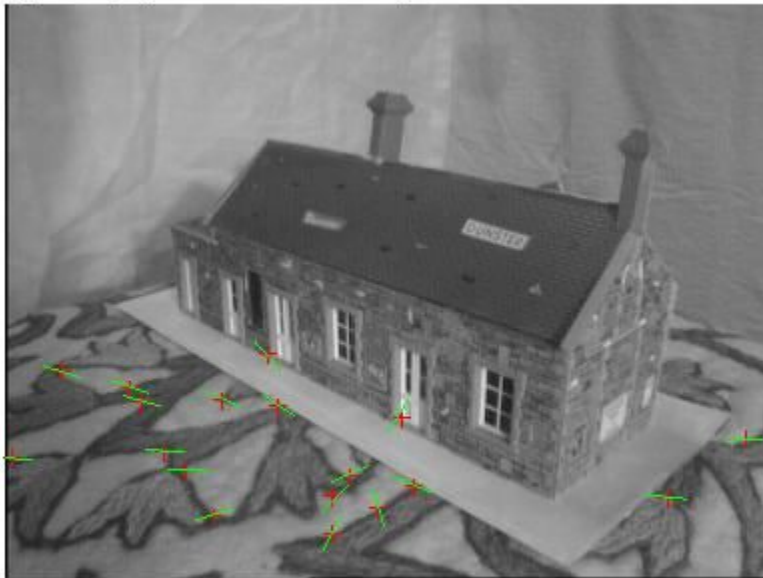
Plotting the epipolar lines on the putative matches in First Image

Figure 2.5 Image 1 Putative, unnormalised

Plotting the epipolar lines on the putative matches in Second Image

Figure 2.6 Image 2 Putative, unnormalised

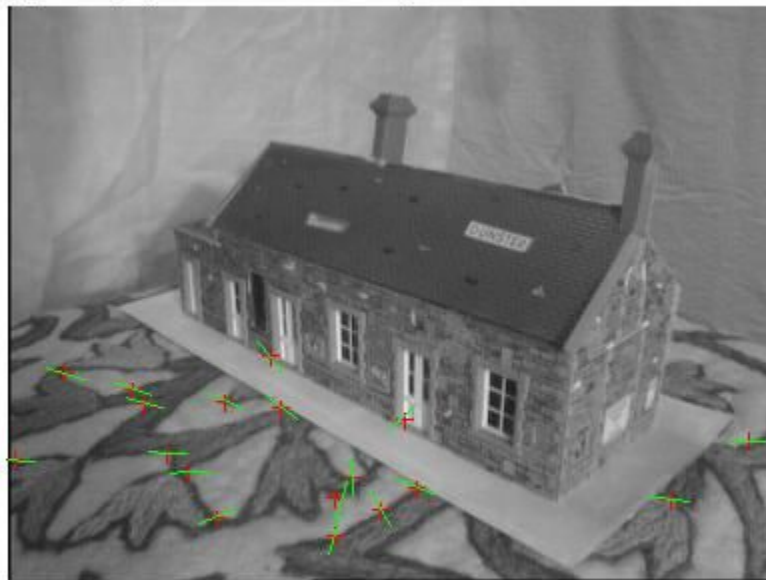
Plotting the epipolar lines on the putative matches in First Image

Figure 2.7 Image 1 Putative, normalised

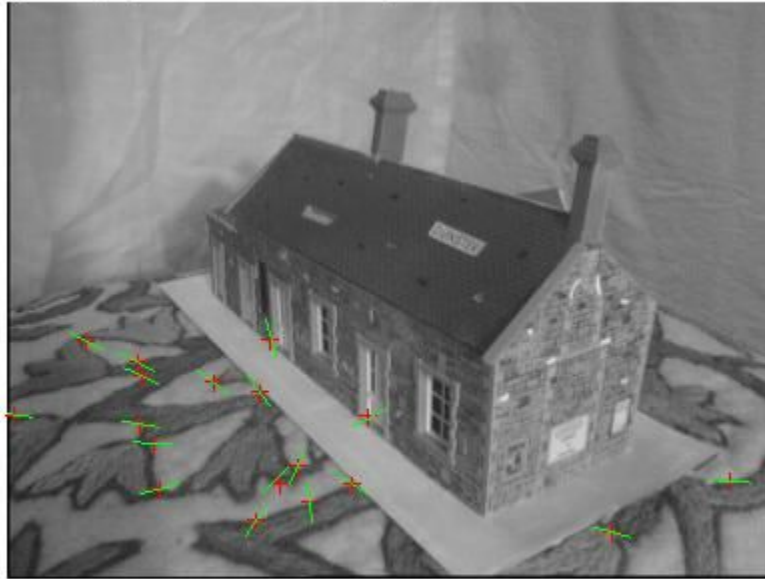
Plotting the epipolar lines on the putative matches in Second Image

Figure 2.8 Image 2 Putative, normalised

Section 4: 3-D reconstructed image is shown in the **Figure 2.4.1**

Residual Error in Image 1 is : 9386534.731

Residual Error in Image 2 is : 2559613639.199

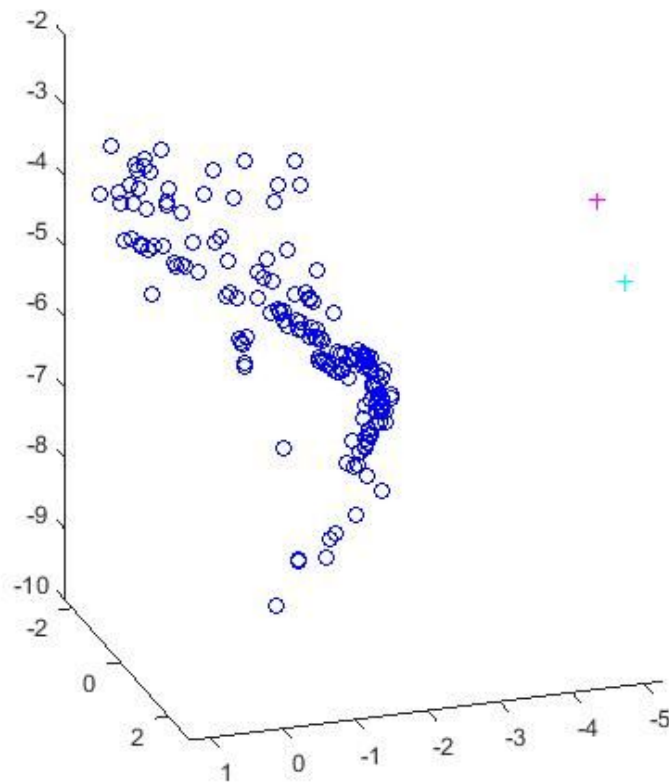


Figure 2.4.1, Cameras in magenta and cyan

Code Files

Concerned files to look for inside the folder are:

Part 1 code files

- dist2.m
- harris.m
- keypoints.m
- panO.m
- ransac.m
- stitch.m
- stitchmulti.m (bonus)

Part 2 code files (some files are reused, ransac and stitch)

- fit_fundamental.m
- ransac.m
- sample_code.m
- stitch.m

Redundant files

- Find_sift.m

How to run the code:

Part 1: Code can be run directly by calling the stitchmulti.m or stitch.m. Please go through the comment in both files to learn about the parameters to be passed.

stitch. can be directly called without parameters too

Part 2: Directly run the sample_code.m.

DATASHEET

1. Datasheet: <https://goo.gl/knJzpc>

SOFTWARE/HARDWARE USED

1. MATLAB R2017a on personal system.
2. Intel i3, 3GB Ram, Sublime Text 3 IDE.

REFERENCES

1. MATLAB Documentation
2. Hw3.pdf
3. Lecture slides
4. Brigham University Robotic Lab
5. Sample Harris detector code.

=====END=====