Robust Estimation and Geometric Vision

Machine Problem 3 : CS 543 Computer Vision UIUC

Naman Shukla

namans2@illinois.edu

INTRODUCTION

The goal of this assignment is to implement robust homography and fundamental matrix estimation to register pairs of images, as well as attempt triangulation and single-view 3D measurements. This assignment consists of the following parts:

1. Stitching pair of images.
2. Fundamental matrix estimation and triangulation.
3. Single view geometry

PART 1 : STITCHING PAIR OF IMAGES

Goal of this part is to stitch two given images based of key points matched. This algorithm is used for creating panoramas and photo sphere with advanced functionality.

ALGORITHM:

1. Load the images and process it (data type , grayscale conversion + remove border)
2. Find corners through harris detector.
3. Generate patches from the harris corner detector.
    a. Set patch size.
    b. Select the patch based on patch size neighborhood.
4. Create : Distance Matrix [corners from image 1 x corners from image 2] which have distance (i,j) where i is the patch from image 1 and j is from image 2.
5. Generate Bracket :
    a. Set top candidates to select in a bracket.
    b. Sort the Distance Matrix based on the distances.
    c. Select top candidates for the bracket and store their coordinates in a bracket list.
6. RANSAC :
    a. Select a minimum of 4 points from the bracket.
    b. Calculate Homography from step 4.a
        i. Create A matrix from the 4 pair coordinates.
        ii. Find eigenvector of transpose(A) * A corresponding to the last eigenvalue.
        iii. Reshape and set as Homography matrix H.

c. Find Inliers:
    i. For all pairs in bracket find the homographic transformation.
    ii. Compare with the corresponding matches.
        1. If it is less than threshold value then return inliers.
        2. Else set the 4.a as outliers.
d. Repeat the above steps till maximum iterations.
e. Return matches and residuals.

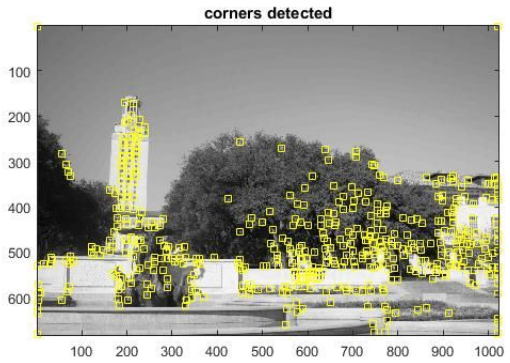
7. Transform two images according to these matches and corresponding H matrix.

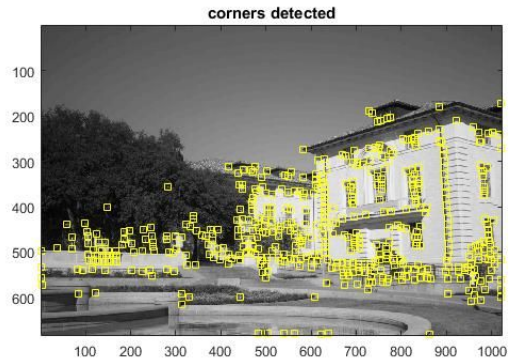
RESULTS:

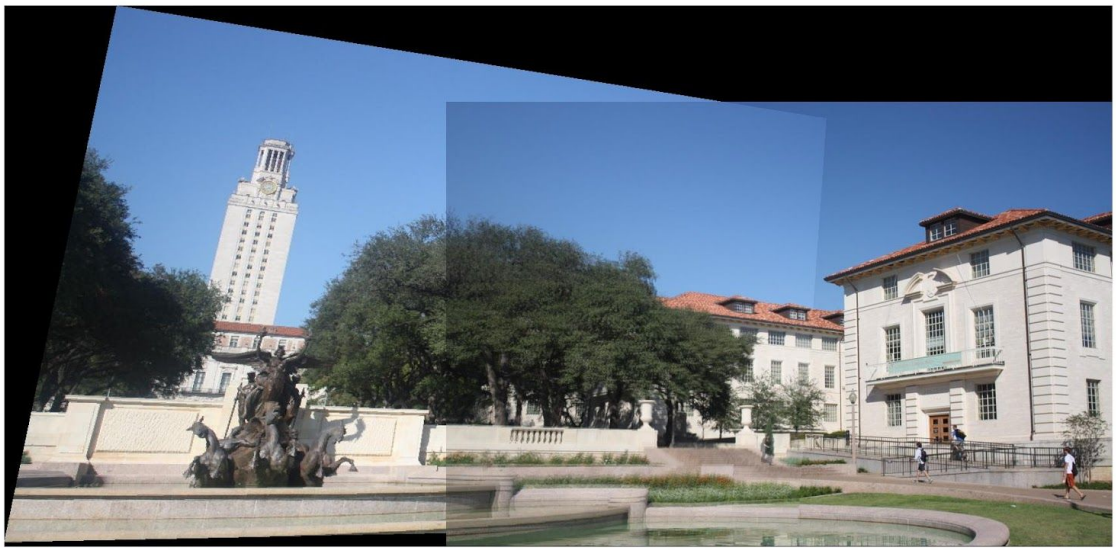
Given Images

Result :

The yellow boxes are the detected corners and the red boxes are the inliers after RANSAC operation in the image below.


corners detected
100
200
300
400
500
600
100 200 300 400 500 600 700 800 900 1000


corners detected
100
200
300
400
500
600
100 200 300 400 500 600 700 800 900 1000



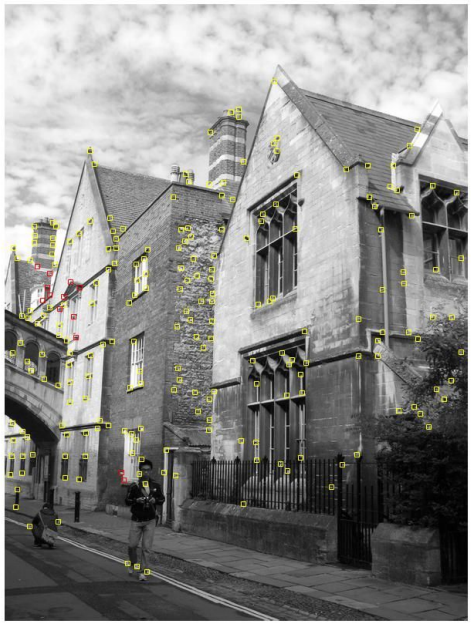
RANSAC output :

Number of homography inliers	24
Average Residual	0.3036259

Additional Images:

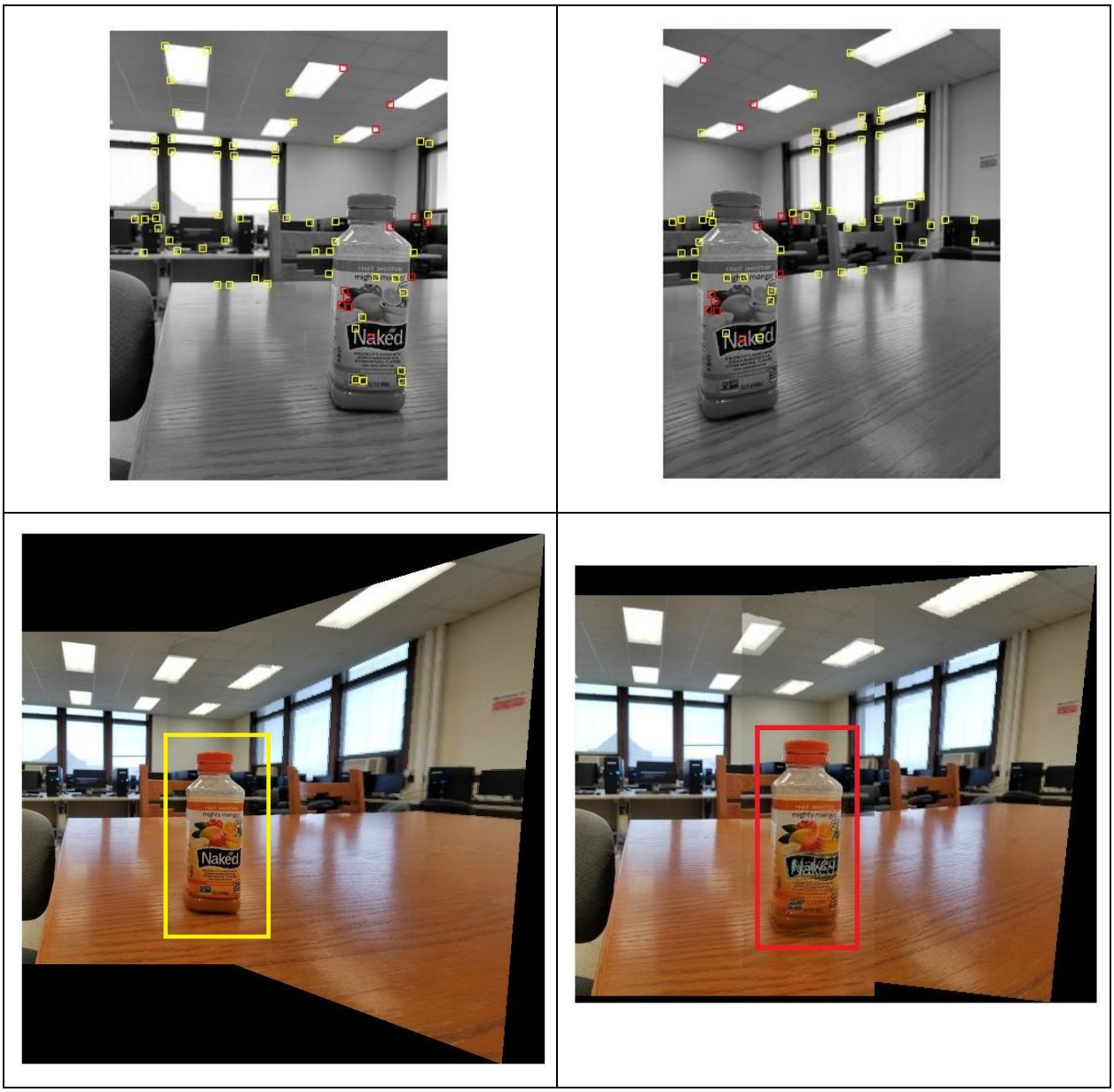
	MOVING OBJECT The reason for choosing this image to identify how stitching produces result when there is an object that gets displaced. This image is from oxford street having some of the pedestrians stationary and some moving.
	PARALLAX The reason for choosing this image is to test parallax on stitching images. This photo is shot by myself in the operations lab. The detailed results are given below.

Results:



Comment :

The image above demonstrates perfectly how a moving object influence the final image quality. It is clear that there exists a blur effect for person in the red circle. This is because he moved within two frames.



Comment :

Some artifacts are clearly observable on the bottom two images. The left image with yellow box is much clearer than right image with red box. The only difference in the two images is that I forced the algorithm to match points that are on the bottle in the left image. For the right image, the matches lies at the back of the room. This results in parallax in the stitching process. As we can clearly see that parallax in the match could result in quality of image.

PARAMETER ANALYSIS

- **Patch Size :** This is the parameter to decide the neighborhood of the corners from harris. With different values of patch size, the function performance varies a lot. A window of 2x2 is very fast but sometimes give blurry image as the RANSAC is unable to identify the correct inlier due to high number of inliers. Huge window of 9x9 gives better results but it is extremely slow in runtime. The optimal window, according to me is 4x4 or 5x5.
- **Threshold :** I changed the values from 0.1 to 5. Higher values are usually unstable and gives weird matches often. I have set 0.8 as threshold.
- **Bracket :** This parameter also affected the performance of the function quite a lot. Higher values like 1000 or more gives usually better result but it takes more time to compute. I have chosen 500 for larger images and 200 for relatively smaller images. These parameters are optimal for my algorithm.
- **RANSAC Iterations :** maximum iterations for RANSAC I have chosen in 500. Any number more than 200 for larger images with more number of corners works relatively well. Hence, I have fixed 500 iterations. This is the key runtime contributing parameter.
- **Harris - sigma, threshold, radius :** I have used 3, 0.05 and 3 for sigma, threshold and radius respectively for harris corner detector. These values I have selected from MP2 and seems to perform better than other combinations.

PART 2 : Fundamental Matrix Estimation and Triangulation

The goal of this section is to estimate the fundamental matrix and perform triangulation on two images with given matches.

FUNDAMENTAL MATRIX :

This subsection is consists of 4 parts analysis.

1. Fitting fundamental matrix with raw ground truth matches:

For this section we will assume that we have full confidence in given matches. The algorithm will find the fundamental matrix without normalizing the data with 8 point algorithm.

Algorithm :

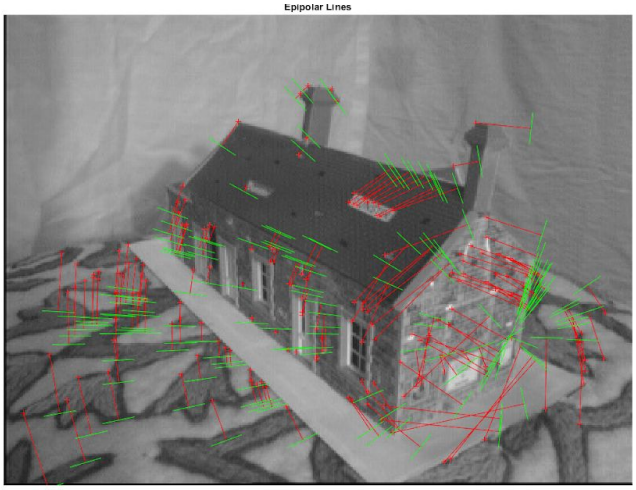
1. Load the images and ground truth matches.
2. Convert cartesian coordinates to homogeneous coordinates.
3. Perform eight point algorithm.
4. Enforce rank 2 constraint.
5. Calculate residuals and epipolar lines.

Results :

Given Images with ground truth matches :

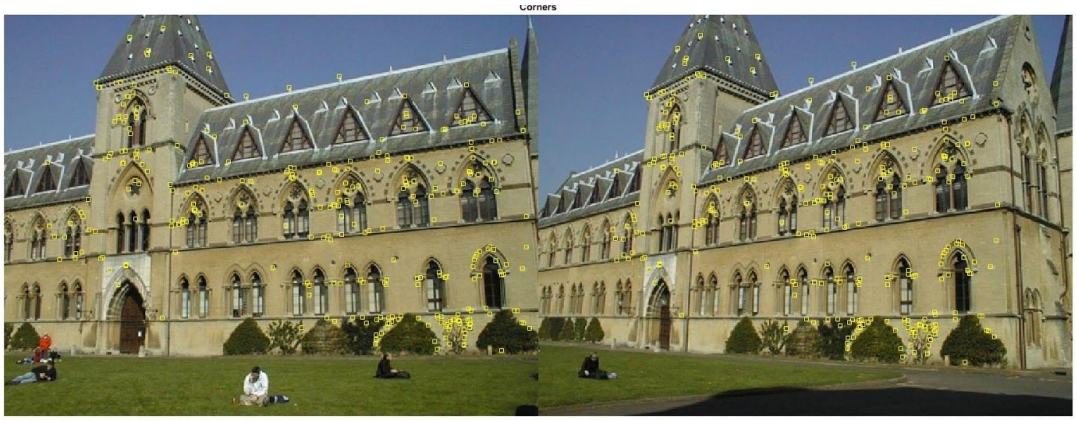
Example 1: House

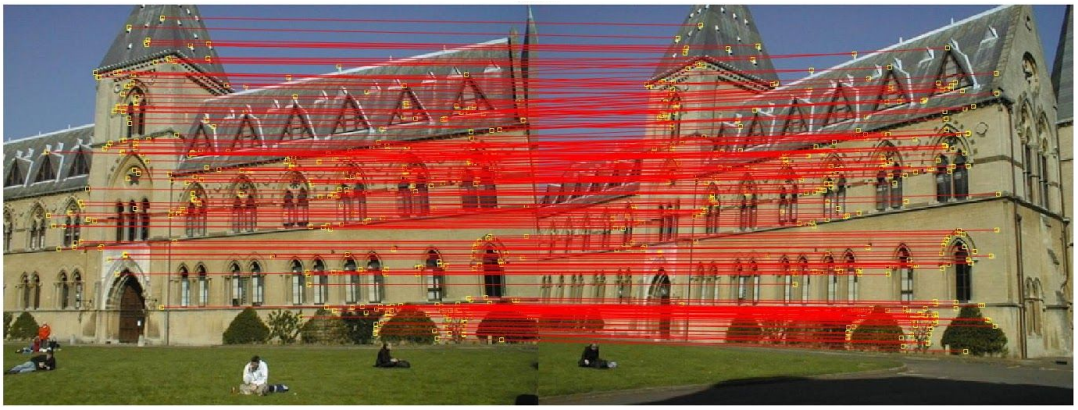
The image below describes the epipolar lines made by unnormalized ground truth.



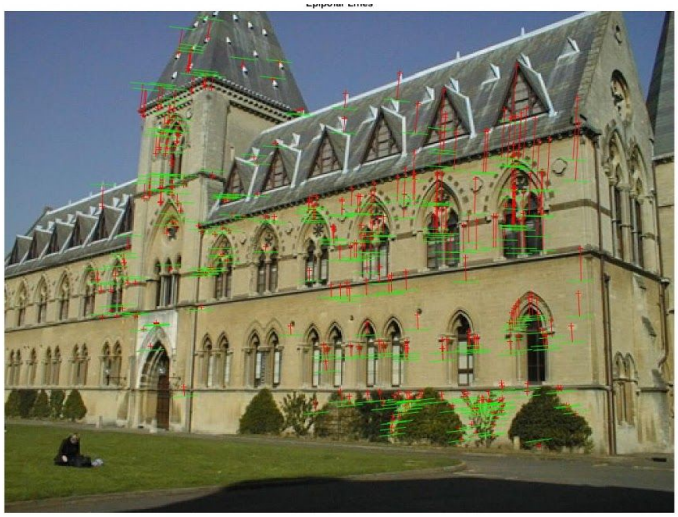
The Mean residual for TrueFit mode: 26.7532

Example 2: Library





The image below describes the epipolar lines made by unnormalized ground truth.



The Mean residual for TrueFit mode: 11.8459

2. Fitting fundamental matrix with normalized ground truth matches:

For this section we will continue our assumption of assume full confidence in given matches. The algorithm will find the fundamental matrix with normalizing the data with 8 point algorithm.

Algorithm :

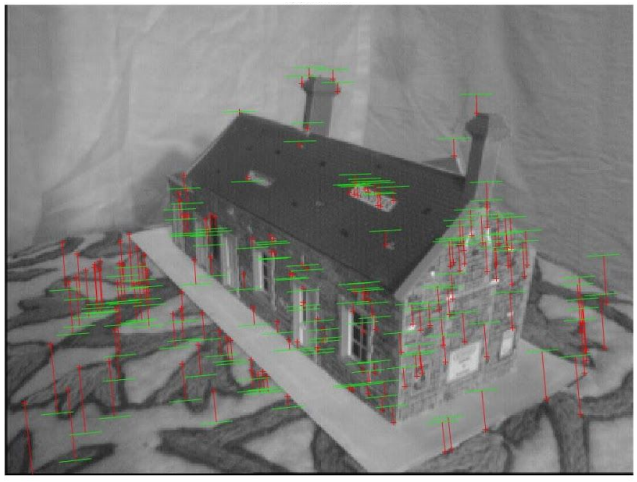
1. Load the images and ground truth matches.
2. Convert cartesian coordinates to homogeneous coordinates of ground truth matches.
3. Normalize the ground truth matches.
4. Perform eight point algorithm.
5. Enforce rank 2 constraint.
6. Calculate residuals and epipolar lines.

Results :

Given Images with ground truth matches :

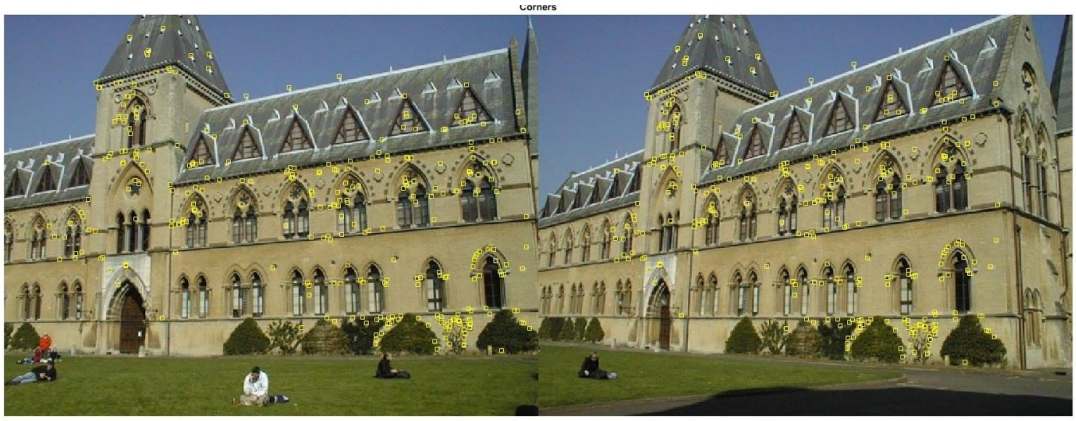
Example 1: House

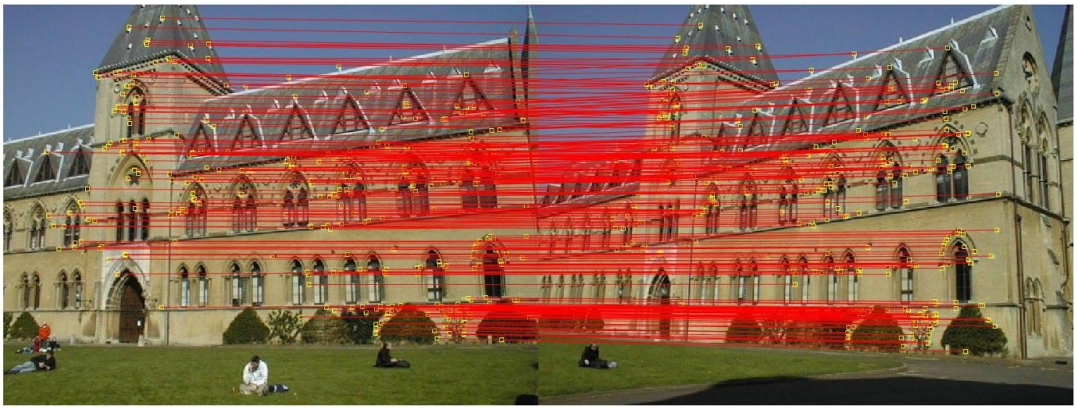
The image below describes the epipolar lines made by normalized ground truth.



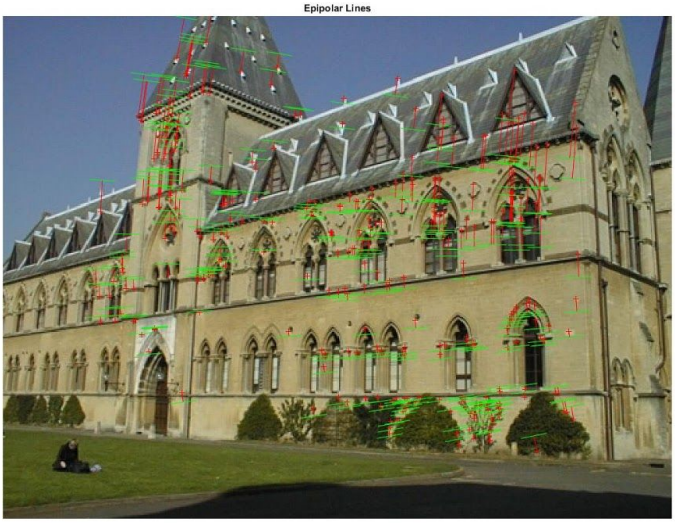
The Mean residual for TrueFit mode: 14.5839

Example 2: Library





The image below describes the epipolar lines made by normalized ground truth.



The Mean residual for TrueFit mode: 10.8974

3. Estimating fundamental matrix with normalized ground truth matches:

For this section we will find the inliers by RANSAC and not assume the ground truth to be true. The algorithm will find the fundamental matrix with normalizing the data with 8 point algorithm.

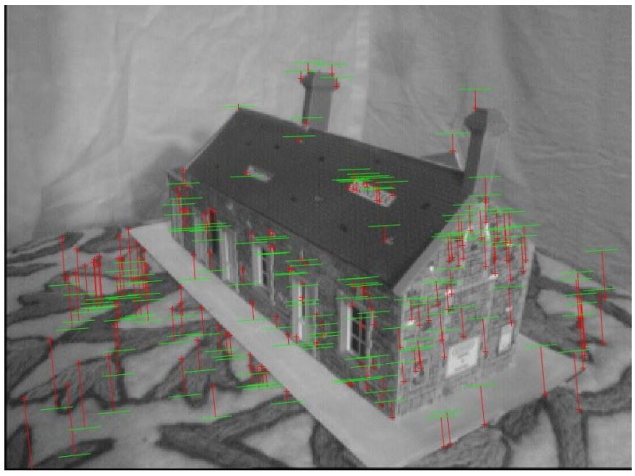
Algorithm :

1. Load the images and ground truth matches.
2. Convert cartesian coordinates to homogeneous coordinates.
3. Normalize the ground truth.
4. Perform RANSAC.
    a. Run till max iteration
    b. Objective function : sum of residuals.
    c. Fitting function : Fundamental matrix function
    d. Return matched inliers.
5. Calculate residuals and epipolar lines.

Results :

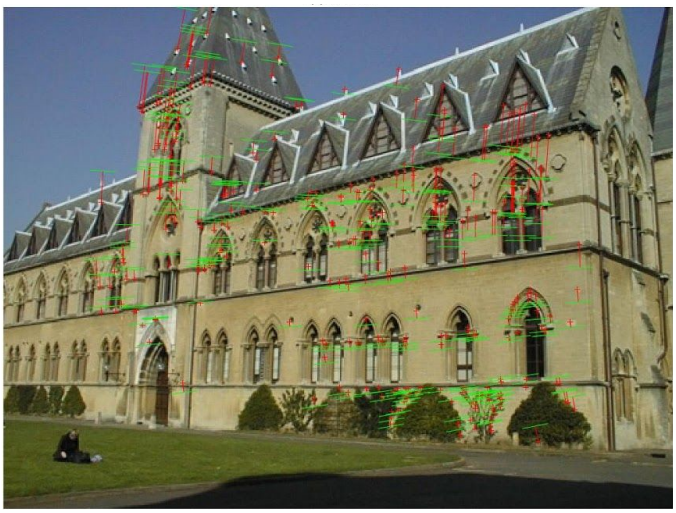
Given Images with ground truth matches :

Example 1 : House



Number of inliers in RANSAC for F is : 163
Mean Residual of Inliers in RANSAC for F is: 13.6995

Example 2 : Library



Number of inliers in RANSAC for F is : 306
Mean Residual of Inliers in RANSAC for F is: 10.6106

4. Estimating fundamental matrix with putative matching using harris:

In this section, we will not use any ground truth at all, instead we will use harris detector for putative matching. The rest of the algorithm is same as RANSAC as explained before.

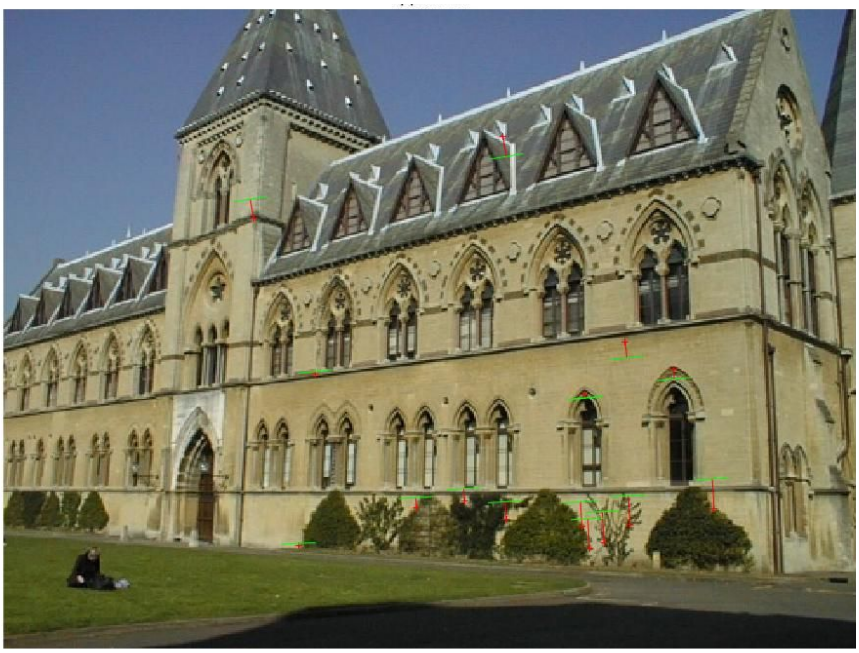
Results :

No ground truth matches are used for the examples shown below.

Example 1 : House

Number of inliers in RANSAC for F is : 17
Mean Residual of Inliers in RANSAC for F is: 14.8445

Example 2 : Library



Number of inliers in RANSAC for F is : 16
Mean Residual of Inliers in RANSAC for F is: 9.8128

TRIANGULATION :

In this section we will perform triangulation to find the following:

1. 3D centers of the two cameras.
2. 3D coordinates of the 2D observed points as ground truth matches.

Algorithm:

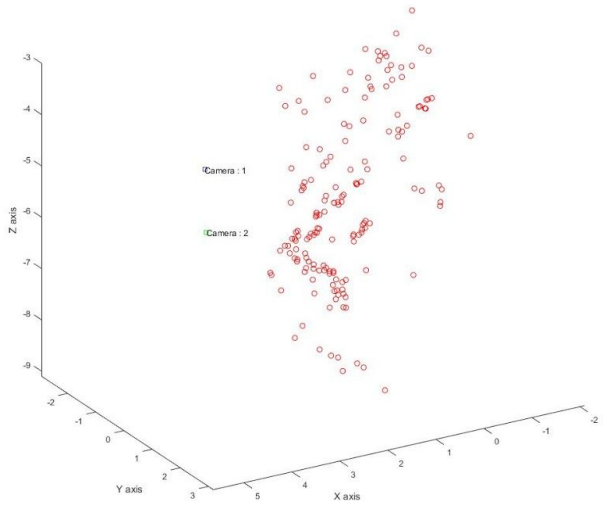
1. Load images and camera matrices.
2. Find camera centers :
    a. The last eigenvector corresponding to the SVD (camera matrix)
    b. Convert it to cartesian coordinates.
3. Create pointwise cross matrix from both matches.
4. Assign triangulation matrix to be the last eigenvector of SVD of step 3.
5. Convert homogeneous system to cartesian coordinates.

Results :

Note : *Videos and GIFs are also available for better visualization in the output folder (‘data/triangulation’)*

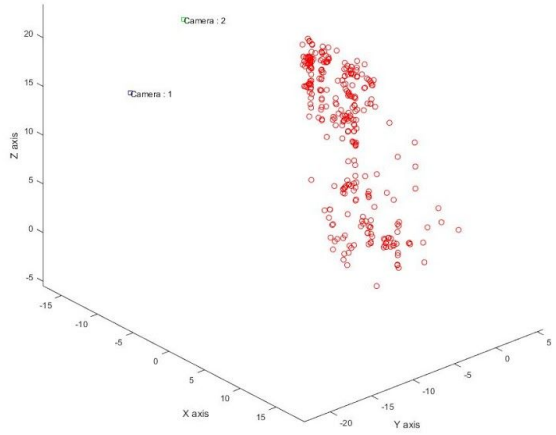
Comments: We can observe slight structure of the roof of house. It is slightly challenging to imagine the true structure from the 3D point matches as these points looks to be independently distributed over the space. Nonetheless, slight structure of both house and library can be identified with closer observation. The library 3D construction is flatter than house model because most of the matching points lie on a single plane (i.e. one wall) in case of library as compared to house model.

Example 1 : House



The Mean residual for triangulation 1:	0.0025221
The Mean residual for triangulation 2:	0.15655

Example 2 : Library



The Mean residual for triangulation 1:	0.073128
The Mean residual for triangulation 2:	0.26768

PARAMETER ANALYSIS

- **RANSAC Iterations :** I have played with bunch of values for RANSAC. The values for this parameter only affects the runtime as we have observed in part 2 of the machine problem. I have opted for value of 1000 maximum number of RANSAC iterations, although 500 also gives relatively similar results.
- **Threshold :** As RANSAC function needs threshold value to compare the ratio of the distance, instead an average distance is used in this implementation. Therefore, the values are slightly larger. I have chosen 35 as the threshold with gives better results. Values for threshold close to 50 tends to give all the ground truth matches as inliers. Whereas single digit values usually give very less number of inliers.
- **Bracket :** The bracket for RANSAC is 8 because of the minimum number of points require to find fundamental matrix.

PART 3 : Fundamental Matrix Estimation and Triangulation

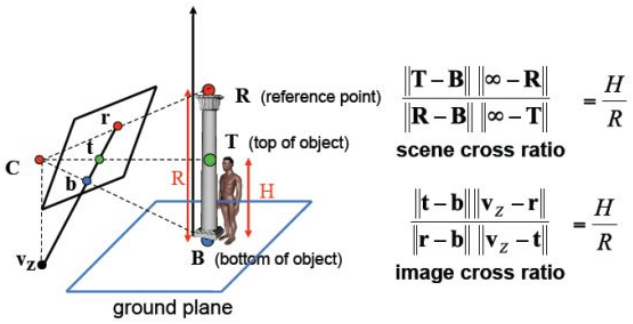
The goal of this part is to identify height of various objects with respect to the reference object. The reference object used here is the height of the one person walking down. The height is calculated by finding the horizon in the image by obtaining the vanishing points through interactive interface.

Algorithm :

1. Process the image:
    a. Get the vanishing points in the image i.e vx, vy, vz
    b. Finding horizon in the image:
        i. Find vanishing points that is not infinity.
        ii. Find line joining them : take cross product of those vanishing points
2. Finding focal length and optical center :

$$\begin{bmatrix} w\cdot u \\ w\cdot v \\ w \end{bmatrix} = \mathbf{K}\,[\mathbf{R}\ \mathbf{t}] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad or \quad \begin{bmatrix} w\cdot u \\ w\cdot v \\ w \end{bmatrix} = \begin{bmatrix} f & s & u_0 \\ 0 & \alpha f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

    a. Optical center (u and v): Solve the set of equation with vanishing points
    b. Focal length (f) : Solve the equation with u and v.
    c. Make K matrix.
3. Finding the rotation matrix :
    a. R : Inverse(K) times the vanishing point
    b. Normalize R
4. Finding height of objects :
    a. Record the objects top point and bottom point
    b. Find the cross ratio.
    c. Multiply cross ratio with reference height.



$$\frac{\|\mathbf{T}-\mathbf{B}\|\,\|\infty-\mathbf{R}\|}{\|\mathbf{R}-\mathbf{B}\|\,\|\infty-\mathbf{T}\|} = \frac{H}{R}$$

scene cross ratio

$$\frac{\|\mathbf{t}-\mathbf{b}\|\,\|\mathbf{v}_z-\mathbf{r}\|}{\|\mathbf{r}-\mathbf{b}\|\,\|\mathbf{v}_z-\mathbf{t}\|} = \frac{H}{R}$$

image cross ratio

Results :

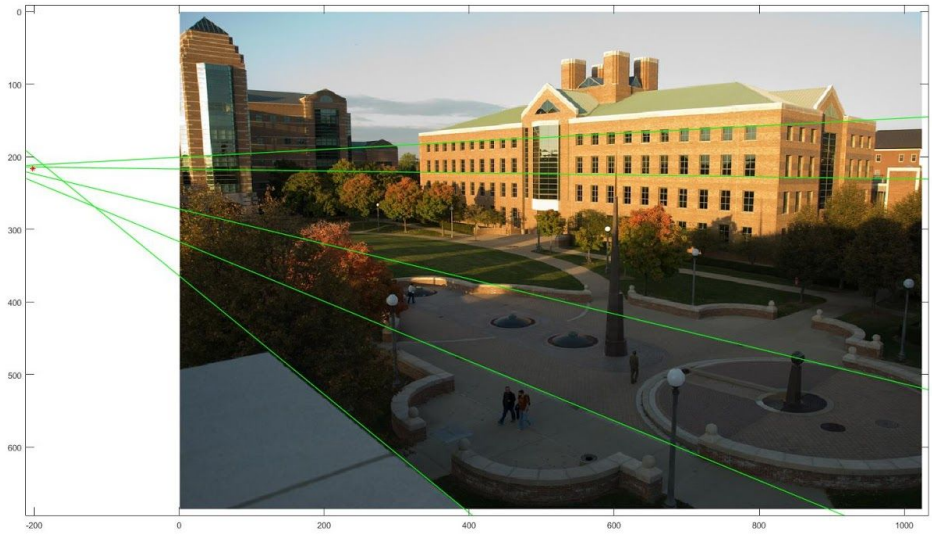
Give image of CSL building

Vanishing points:

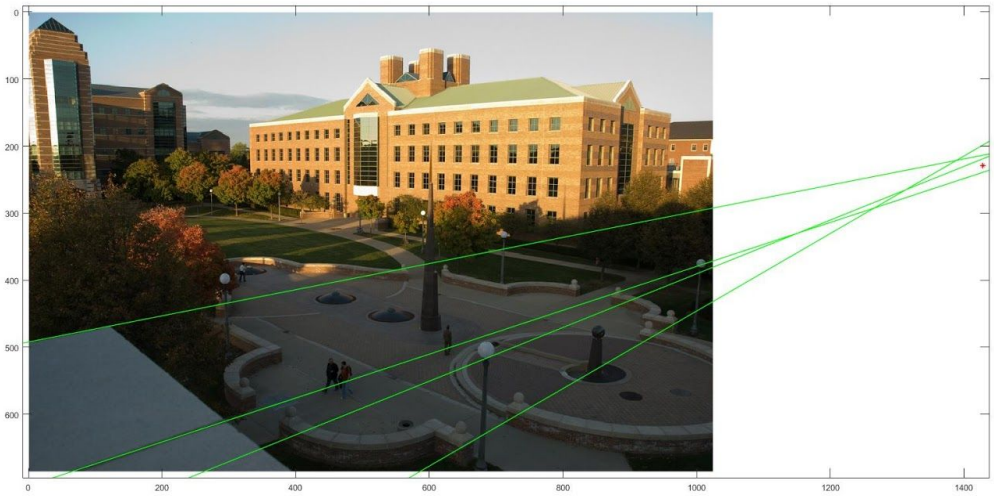
Note : *In my implementation, I have an option for getting interactive interface as well as stored vanishing points calculated from previous attempts. This is used to save time and doing repetitive calculations through interactive interface.*

To change the option, toggle the parameter 'estimate' in the main function.

Vanishing Point at Z : (-215, 195)

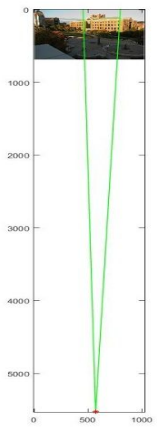


Vanishing Point at X : (1430, 185)



Vanishing Point at Y: (430, 4680)

(The image is smaller due to the scale of the vanishing point)



Horizon:

Height of CSL :

Height of the CSL building : 30.82 meters

Height of Lamp :

Height of the Lamp Post : 4.92 meters

Height of Spike :

Height of the Spike : 10.01 meters

The K matrix :

-734.12	0	458.91
0	-734.12	343.67
0	0	1

Comparing the above matrix with the standard K matrix we will get:

- Camera centers are [458.91, 343,67]
- Focal length 734.12

Why my focal length is negative?
The reason for getting a negative focal length is that we have modeled our observations as the image formation in front of the camera (analogous to inverted mirror) Hence, we have negative focal length. To interpret this as a focal length of the lense, we have to take the absolute value.

The Rotation matrix:

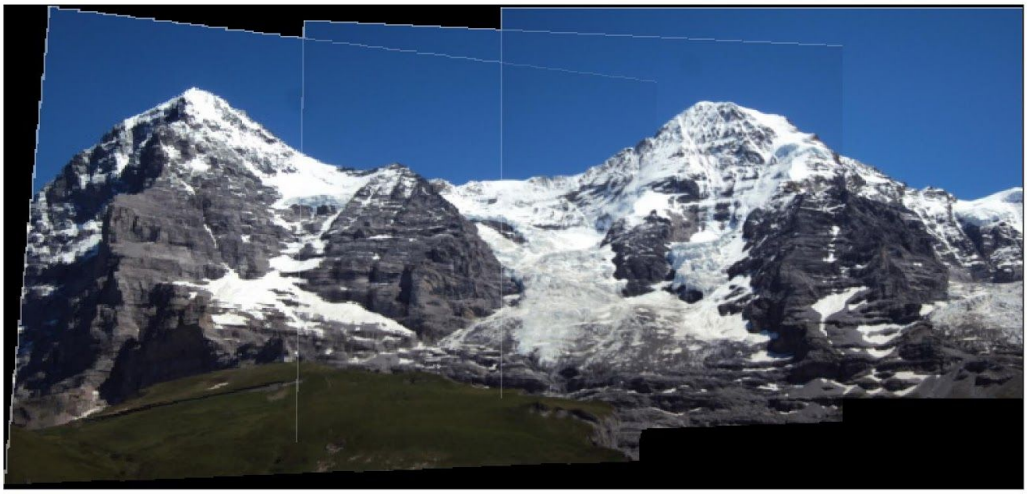
-0.749169802243528	0.00657299401931706	0.662345380565166
0.105705929543880	-0.985949102897558	0.129346909335958
0.653889030238050	0.166916632619767	0.737948489997250

BONUS IMPLEMENTATIONS

PART 1 : STITCHING MULTIPLE IMAGES

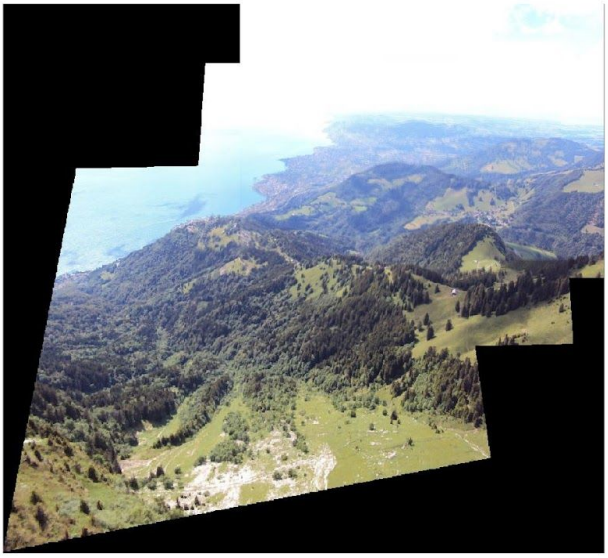
In this section, implementation is done on pictures with multiple frames. Some of the results are mentioned below:

Image 1 : Hill [3 frames]



Number of homography inliers: 48
Residual : 2.542733e-01
Number of homography inliers: 94
Residual : 2.438230e-01

Image 2 : Ledge [3 frames]



Number of homography inliers: 31
Residual : 2.403084e-01
Number of homography inliers: 10
Residual : 1.054626e-01

Image 3 : Pier [3 frames]

Number of homography inliers: 39
Residual : 2.400108e-01
Number of homography inliers: 43
Residual : 2.295640e-01

Additional Images :

Image 4 : Random Building [3 frames]

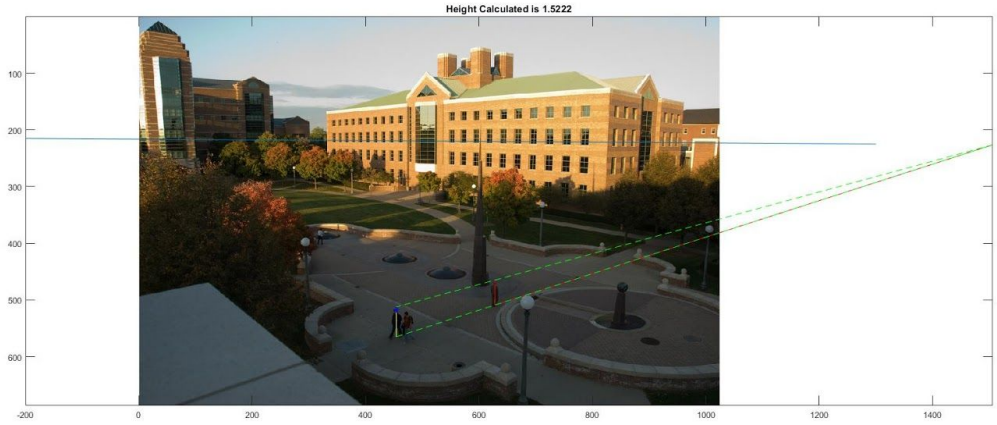
Number of homography inliers: 14
Residual : 1.152514e-01
Number of homography inliers: 27
Residual : 2.735850e-01

Image 5 : UIUC Transportation Building [4 Frames]

Number of homography inliers: 15
Residual : 2.039291e-01
Number of homography inliers: 10
Residual : 1.965400e-01
Number of homography inliers: 10
Residual : 6.716199e-02

PART 3 : ADDITIONAL MEASUREMENTS

Height Comparison of two people:



Height of man with red jacket : 171 cm
Height of the other man : 152 cm

Window Measurements :

Height of Bigger Window : 5.86m

Height of Bigger Window : 4.92m

PART 3 : AUTOMATED CORNER HEIGHT DETECTION (BLOB DETECTOR)

I have integrated the blog detector from machine problem 2 with the above height detection. Now, the blogs which used to detect the corners, can detect height of the object inside the blob.

Detected 197 blobs , threshold: 0.2 , sigma : 2.0 , scale-space size : 12

Comment: Although, the implementation shown above prints some useless blobs that are not useful for finding heights. But, this method gives consistent result with same object at different location along with automated detection.

Analysis : Now we can perform multiple observations of same object which is present at multiple locations. For example, lamp post is present at 4 visible places. Averaging the values will help in increasing the measurement accuracy (due to law of averages in error). Given below an illustration about the above method having objects at different locations.

	Location 1	Location 2	Location 3	Location 4	Average Height
Lamp top	0.939	1.156	0.845	1.2967	1.059
Window 1	5.057	4.974	4.995	4.871	4.974

PARAMETER ANALYSIS

- **Sigma :** Larger sigma values detects larger features in the image whereas smaller sigma value detect smaller features in the image. I used sigma of 2 and found to give better results over others as I am interested in finding circles capturing windows and lamp post top.
- **Threshold :** I've chosen 0.2 for this parameter as this parameter majorly decides the number of circles. I changed this parameters depending on size of image.
- **Space Step :** This is a major contributor to running time as well. I realized any value from 10 to 15 is good for implementation.

REFERENCES :

- ❏ SIFT matlab code.
- ❏ Distance calculation code.
- ❏ CS 543 Lecture Slides. [1] [2] [3] [4]
- ❏ Derek Hoiem - Chapter 2 : Single View Geometry
- ❏ Additional references.