ECE 549 Homework 3

Fangwen Wu (fwu11)
March 21, 2017

1 Single-View Metrology

A. The main idea of this part is to calculate the vanishing points from at least three selected parallel lines in the scene (I used three lines). The point and the line are represented in homogeneous coordinates so the vanishing points are the cross product of the lines. In order to solve the case when the lines do not intersect at exactly the same point, the vanishing points are chosen based on the minimized angular difference. The angular difference is the difference between the direction from the line segment midpoint to the vanishing point and the direction of the line segment. The scoring function used for measuring the minimum angular difference is $s_j = \sum_i |l_i| e^{(-\frac{|a_i - \theta_i|}{2\sigma^2})}$. Vanishing points are shown in the images below in three directions specified in part c.

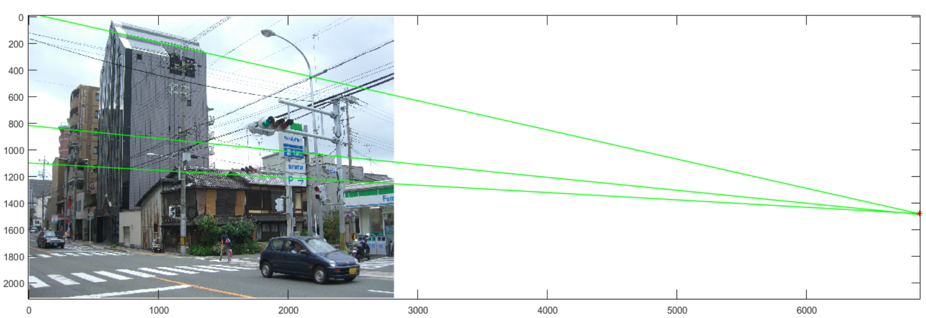

Figure 1: Vanishing point in X direction

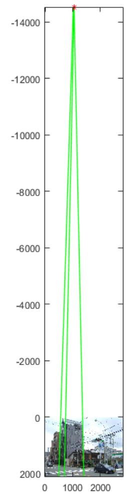

Figure 2: Vanishing point in Y direction

Figure 3: Vanishing point in Z direction

$$
\begin{aligned}
VP_x &= (6869.77, 1480.43) \\
VP_y &= (1050.96, -14514.39) \\
VP_z &= (-466.64, 1570.84)
\end{aligned}
$$

The horizon is found by the cross product of the vanishing points VP_x and VP_z. The normalized equation is

$$0.0123u + 0.9999v - 1564.9 = 0 \tag{1}$$

It is plotted as below:

Figure 4: Horizon

B. In this problem we try to work out the intrinsic camera matrix from orthogonal vanishing points. Since we get three vanishing points that are orthogonal to each other, we Notice that

$$p_i = KRX_i \tag{2}$$

where intrinsic matrix $K = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}$ We take the advantage of the orthogonality which gives

$$p_i^T p_j = 0 \quad \text{for} \quad i \neq j \tag{3}$$

which relates two vanishing points to

$$p_i^T K^{-T} R^{-T} R^{-1} K^{-1} p_j = 0 \tag{4}$$

Since rotation matrix R has the property that $R^{-T}R^{-1} = I$ the above equation can be simplified into

$$p_i^T K^{-T} K^{-1} p_j = 0 \quad i,j = 1,2,3 \quad \text{and} \quad i \neq j \tag{5}$$

We can use matlab **solve** function to solve the systems of equations and get the f, u_0,v_0 values in K.

$$K = \begin{bmatrix} 3047 & 0 & 1241.6 \\ 0 & 3047 & 949.4 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

C. The rotation matrix can be computed from equation (2). We already know the intrinsic matrix K and three vanishing points. Each vanishing point provides one column of rotation matrix R (r_{xi}) and finally we normalize the Rotation matrix to make each row and column of R has unit length.

$$vp_1 = KR \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = Kr_{x1}$$

$$vp_2 = KR \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = Kr_{x2}$$

$$vp_3 = KR \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = Kr_{x3}$$

$$R = \begin{bmatrix} 0.8764 & -0.0121 & 0.4815 \\ 0.0827 & -0.9811 & -0.1751 \\ 0.4745 & 0.1933 & -0.8588 \end{bmatrix}$$

D.First we calculate the horizon using the same method as part a. We first get two vanishing points in the horizontal direction and then take the cross product of them:

$$vp_1 = \begin{bmatrix} -1.415 \times 10^3 & 1.247 \times 10^3 & 1 \end{bmatrix}$$
$$vp_2 = \begin{bmatrix} -4.728 \times 10^3 & 1.292 \times 10^3 & 1 \end{bmatrix}$$
$$horizon : 0.0074u - v + 1257 = 0$$

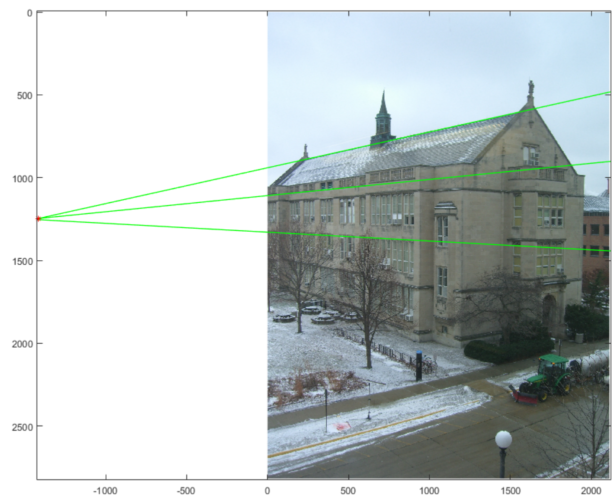

Figure 5: Vanishing point

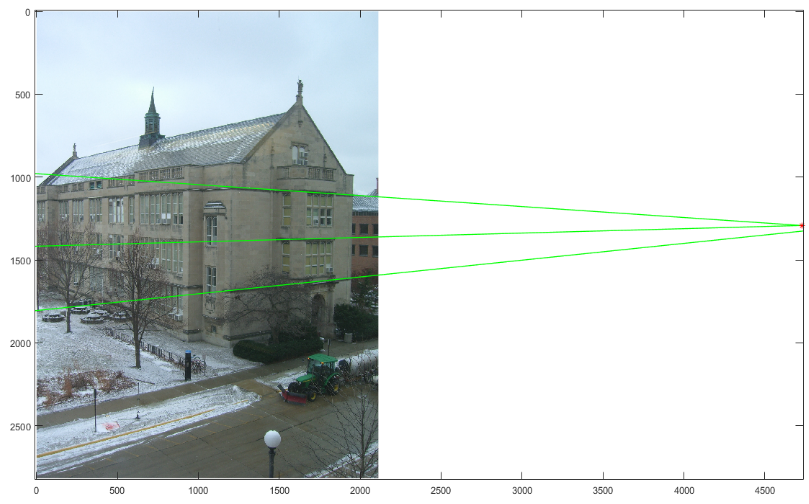

Figure 6: Vanishing point

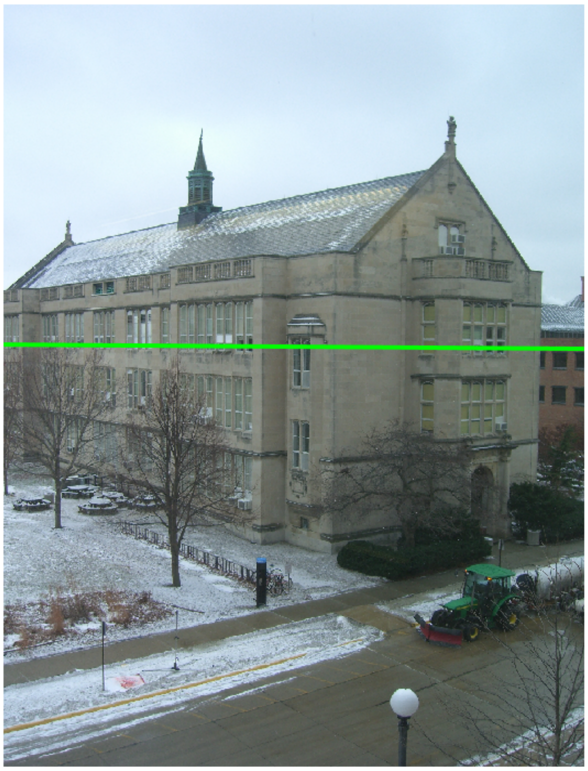

Figure 7: Horizon

Notice that the three images below have the same coordinate system, some are just zoomed to show better detail of the marked coordinates and objects.

We first estimate the height of the building. The height of the building can be estimated from drawing a line from the bottom of the sign passing through the bottom of the building and reach the horizon at a vanishing point. The other line passes through the top of the sign to the same vanishing. These two lines indicates parallel relationship in the scene thus, line segment indicated by two coordinates in the middle share the same height of the sign. The coordinates of the top and bottom of the line segment and the top and bottom of the building are shown in the image. We can then estimate the height of the building which is proportional to the height of the line segment.

$$h_{building} = \frac{529.2 - 264.5}{529.2 - 508} \times 1.65 = 20.6m \tag{7}$$

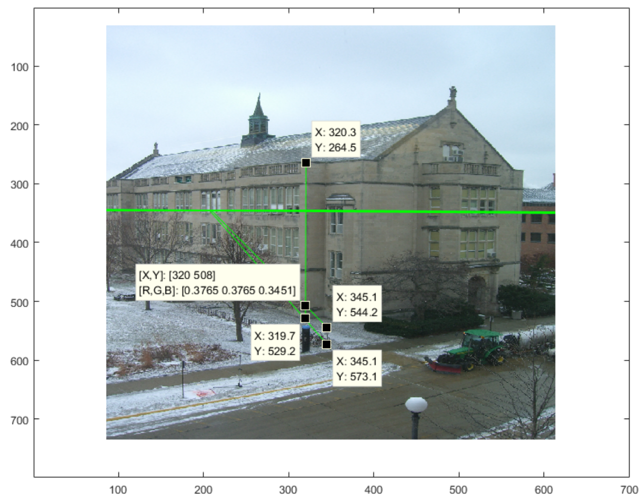

Figure 8: Building height estimation

Next, we estimate the height of the tractor. We can use the similar method above and we get

$$h_{tractor} = \frac{621.3 - 563}{616.6 - 579.4} \times 1.65 = 2.33m \tag{8}$$

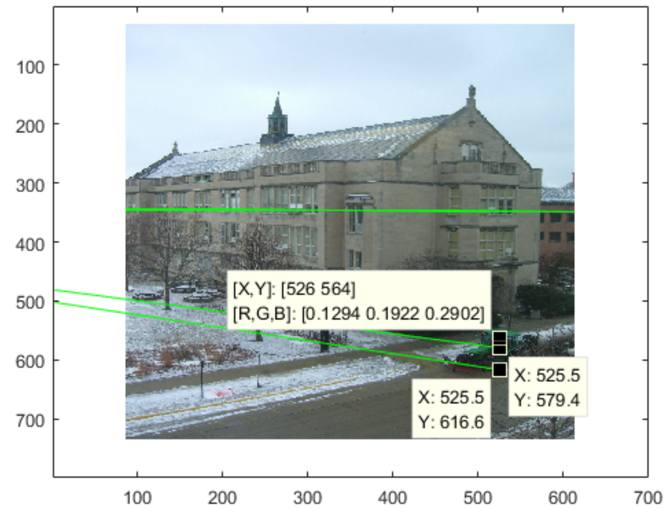

Figure 9: Tractor height estimation

Lastly, the height of the camera is represented by the horizon. Since we already know the coordinates of the top and the bottom of the sign in the previous questions and the coordinates of the horizon is shown below in the image, thus we can estimate the height of the camera by

$$h_{camera} = \frac{573.1 - 347}{573.1 - 544.2} \times 1.65 = 12.9m \tag{9}$$

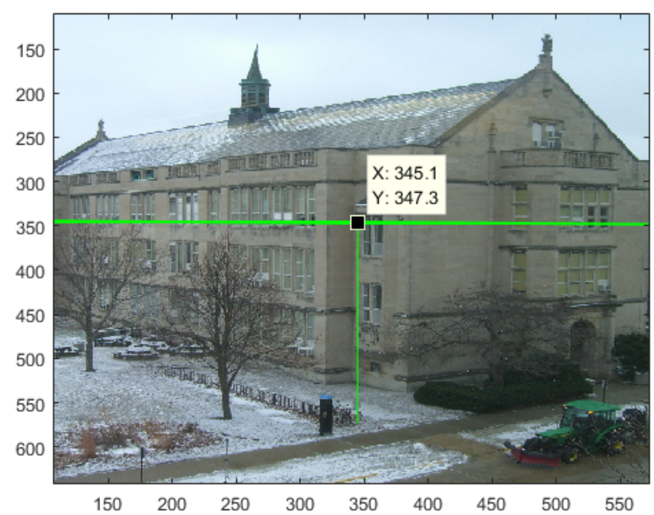

Figure 10: Camera height estimation

E. Extra credit
In this problem we implement the RANSAC to detect the orthogonal vanishing points. We used the method suggested in [1] which first assumes there exists 500 vanishing point hypothesis and randomly choose a pair of edges to determine the vanishing point, we use the consistency measure described in [1] to determine the edges that belongs to the hypothesis vanishing point. The threshold for detecting inlier points from

consistency measure is 1 pixel. The consistency measure is the distance between the line defined by the line segment midpoint to the vanishing point and the endpoint of the segment. Then we can get 3 clusters from the code provided and we perform similar method as part A to find the vanishing point by finding the minimum angular difference. Then in part 2 we repeat what we done in 1A to find the camera matrices. The results are shown below.

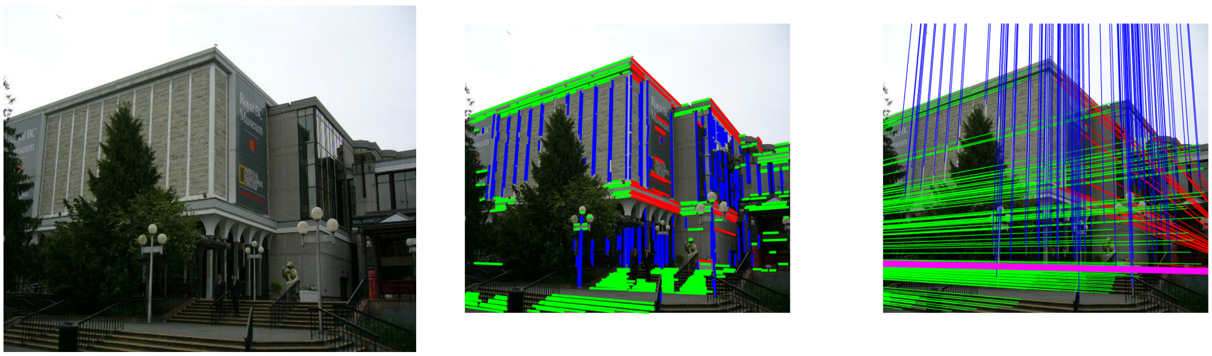

Figure 11: Image 1

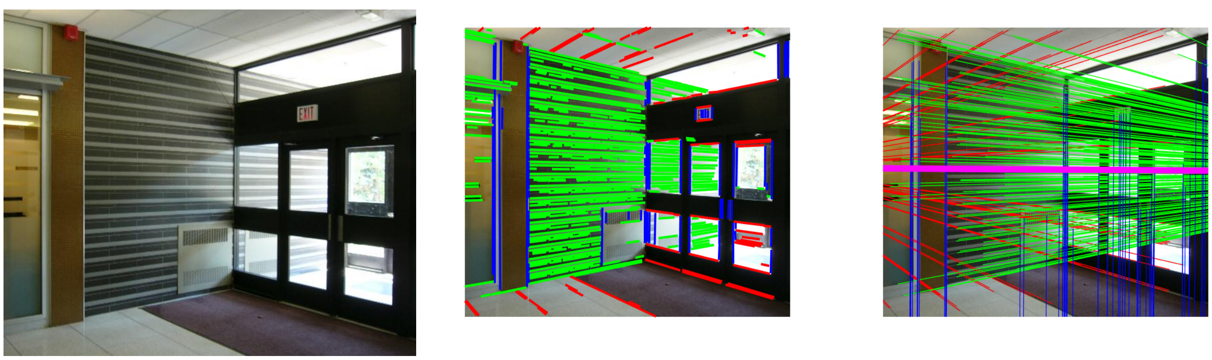

Figure 12: Imgae 8

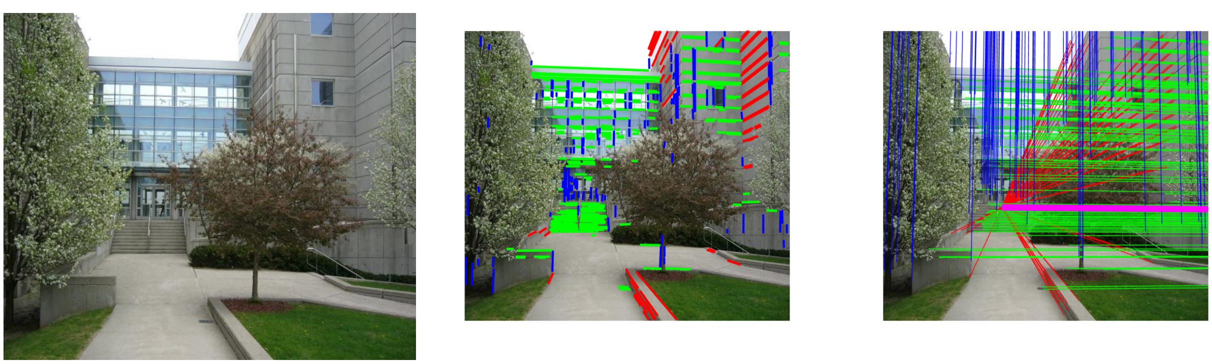

Figure 13: Image 20

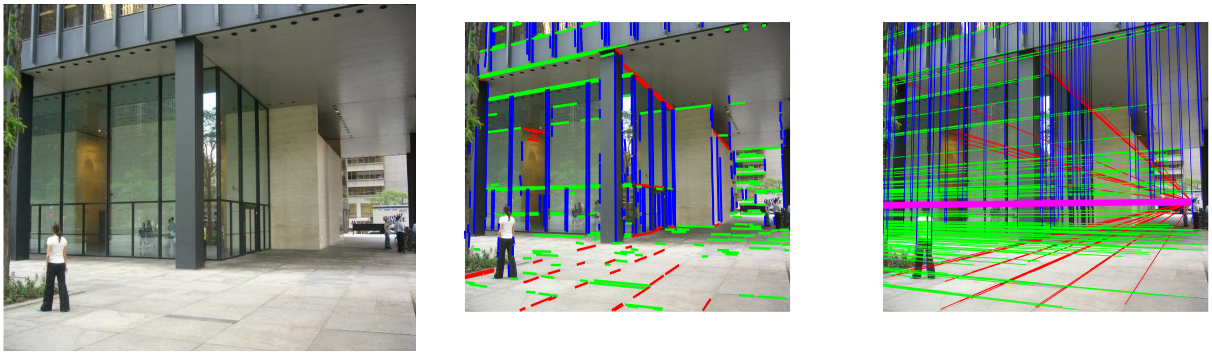

Figure 14: Image 36

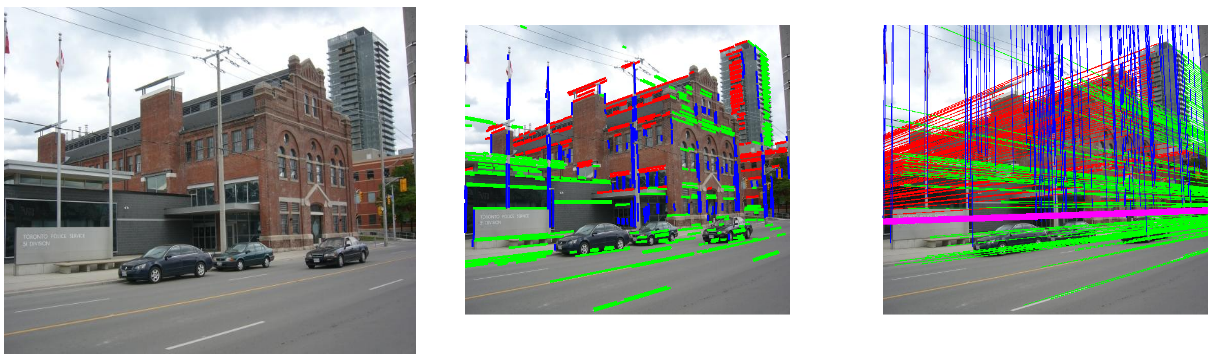

Figure 15: Image 42

Image number	focal length(mm)	u_0 (pixel)	v_0 (pixel)
1	670.08	304.73	248.72
8	616.73	102.57	235.95
20	702.31	299.67	228.05
36	635.10	348.46	248.48
42	677.64	322.60	258.95

$$R_1 = \begin{bmatrix} 0.549 & -0.836 & 0.020 \\ 0.201 & 0.109 & -0.974 \\ 0.812 & 0.538 & 0.227 \end{bmatrix}$$

$$R_8 = \begin{bmatrix} -0.495 & 0.869 & -0.006 \\ -0.005 & 0.004 & 1 \\ 0.869 & 0.495 & 0.002 \end{bmatrix}$$

$$R_{20} = \begin{bmatrix} -0.078 & 0.997 & 0.008 \\ 0.082 & 0.014 & -0.997 \\ 0.994 & 0.077 & 0.083 \end{bmatrix}$$

$$R_{36} = \begin{bmatrix} 0.378 & -0.926 & -0.010 \\ 0.072 & 0.040 & -0.997 \\ 0.923 & 0.376 & 0.082 \end{bmatrix}$$

$$R_{42} = \begin{bmatrix} -0.738 & 0.674 & -0.029 \\ 0.0789 & 0.0439 & -0.996 \\ 0.670 & 0.738 & 0.0856 \end{bmatrix}$$

Image number	focal length error (mm)	image center error (pixel)	zenith angle error (degree)	horizon error
1	0.0478	2.236	0.0525	0.0167
8	0.7848	131.67	0.1585	0.0227
20	1.2246	36.83	0.3413	0.0112
36	0.4181	43.36	0.0368	0.0245
42	0.0244	16.81	0.2593	0.0359

2 Mission Possible

a)Yes. The pixel in the image plane of the camera and a 3D point in the scene has the following relationship $x = K[Rt]X$. In order to fool the security camera, the scene displayed on the projection screen should be the same as the real scene seen by the camera. Since the camera is free to rotate and is fixed in location (t is fixed, thus can be estimated beforehand), the displayed content should be adjust according to the rotation of the camera. We need to know the intrinsic matrix K and the rotation angle of the camera in order to compute the rotation matrix R. Once we know these parameters, we can project the empty hallway in a way such that it looks like the real hallway from the camera.

b)Yes. In this case we need to estimate the intrinsic matrix of the eye of the guard. Since the guard is free to move around, we need to estimate both rotation and translation $[Rt]$. The projected content should update according to the movement of the security guard.

c)No, it is impossible. The camera and the security guard have different intrinsic matrices, and the projector system is not able to project two different scenes at the same time to fool both of them.

3 Epipolar Geometry

In this question the test used for deciding inliers and outliers is RANSAC and normalized 8-point algorithm. We first use 8-point algorithm to determine the fundamental matrix through randomly picking 8 pairs matched points. Then use the fundamental matrix to check if the distance between the matched point in the first image $\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$ and the epipolar line L calculated by the matched point in the second image multiply by the fundamental matrix, $L = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = F \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$ is within the distance threshold.

$$d = \frac{|au + bv + c|}{\sqrt{a^2 + b^2}} < \delta \tag{10}$$

The matched points inside the distance threshold are counted as the inliers. In this problem, the threshold is set to be 1 pixel.
The fundamental matrix

$$F = \begin{bmatrix} -2.705 \times 10^{-7} & -5.406 \times 10^{-5} & 0.0156 \\ 5.146 \times 10^{-5} & 2.525 \times 10^{-6} & 0.0982 \\ -0.0146 & -0.0974 & 0.9902 \end{bmatrix} \tag{11}$$

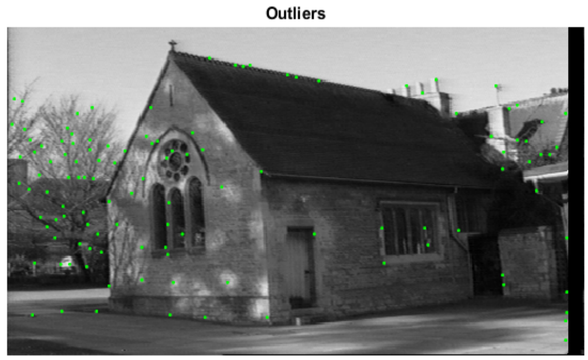

Figure 16: Outliers on top of the first image with threshold of 1 pixel

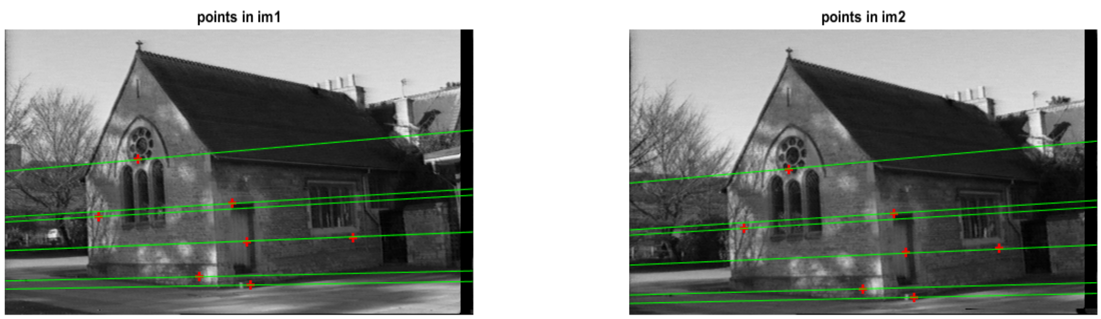

Figure 17: 7 sets of matching points with epipolar lines

4 Affine Structure from Motion

Summary of the algorithm with pseudocode

Algorithm 1 Affine structure from motion

Input: m images and n tracked features **x**
Output: Motion matrix A, shape matrix X and Camera position matrix K.
Procedure:

1.Initialization: For each image, calculate and subtract the mean off all the features
2.Construct a 2m by n measurement matrix D which is a collection of all the 2D points(n) with x and y coordinates in all m images
3.Factorize D using Singular Value Decomposition $D = UWV^T$
4. $U_3 = U(:, 1:3)$
5. $W_3 = W(1:3, 1:3)$
6. $V_3^T = V^T(1:3, :)$
7. Calculate the Motion and shape matrices: $A = U_3\sqrt{W_3}$, $X = \sqrt{W_3}V_3^T$

Eliminate the affine ambiguity

8. Set up system of equations from orthographic constraints
9. Compute $L = QQ^T$ by least squares method.
10. Recover Q from L by Cholesky decomposition
11. Update A and X: $A = A'Q, X = Q^{-1}X'$
12. Calculate camera position for each frame and normalize: $K = A(1:2:end) \times A(2:2:end)$

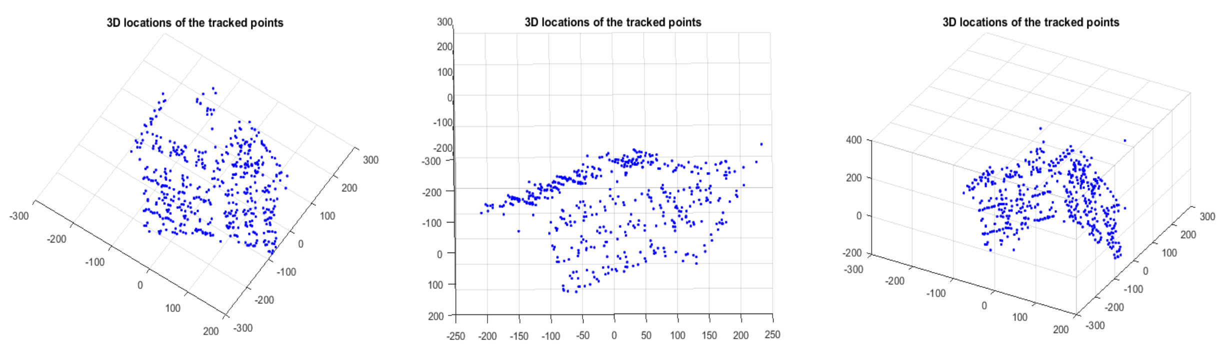

Figure 18: Three different viewpoints of the predicted 3D locations of the tracked points

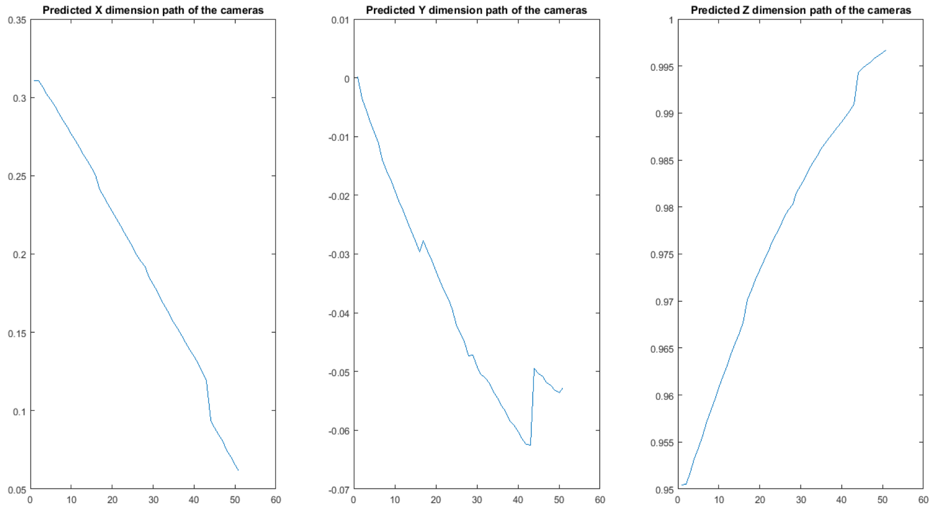

Figure 19: Predicted 3D path of the cameras in three dimensions

References [1] Jean-Philippe Tardif, Non-Iterative Approach for Fast and Accurate Vanishing Point Detection