Assignment 05
TTK4255 - Robotsyn

Elias Mohammed Elfarri

Mars, 2021

Contents

1 Part 1 - The Epipolar Constraint

1.1

As there is a zero on the other side of the equation, a scaling factor will just be cancelled out. such that $c\tilde{\mathbf{l}} = [ccos\theta, csin\theta, -c\rho]$, and therefore:

$$\tilde{\mathbf{x}}^T c\tilde{\mathbf{l}} = \tilde{\mathbf{x}}^T\tilde{\mathbf{l}} = 0$$

Normalizing an arbitrary homogeneous line as such:

$$\begin{bmatrix} \frac{a}{\sqrt{a^2+b^2}} & \frac{b}{\sqrt{a^2+b^2}} & \frac{c}{\sqrt{a^2+b^2}} \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta & -\rho \end{bmatrix}$$

1.2

Proving that 2 points a and b satisfy the line equation:

$$\tilde{\mathbf{l}} = a \times b = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} -a_3b_2 + a_2b_3 \\ a_3b_1 - a_1b_3 \\ -a_2b_1 + a_1b_2 \end{bmatrix} = \begin{bmatrix} cos\theta \\ sin\theta \\ -\rho \end{bmatrix}$$

for 2D coordinates where $(a_1, a_2, a_3) = (a_1, a_2, 1)$ and $(b_1, b_2, b_3) = (b_1, b_2, 1)$, it gives the following cross product:

$$\begin{bmatrix} -b_2 + a_2 \\ b_1 - a_1 \\ -a_2b_1 + a_1b_2 \end{bmatrix}$$

There is combination that allow the line equation $\tilde{\mathbf{l}} = (k, l, -m)$ where $(a_1, a_2, a_3) = (\frac{m-kl}{k}, k, 1)$ and $(b_1, b_2, b_3) = (l + \frac{m-kl}{k}, 0, 1)$:

$$\tilde{\mathbf{l}} = a \times b = \begin{bmatrix} k \\ l + \frac{m-kl}{k} - \frac{m-kl}{k} \\ -k(l + \frac{m-kl}{k}) \end{bmatrix} = \begin{bmatrix} k \\ l \\ -m \end{bmatrix}$$

Hence the line through any two points satisfies the line equation.

It is also known that $\tilde{\mathbf{l}} \cdot a = 0$ and $\tilde{\mathbf{l}} \cdot b = 0$ where the cross product between them gives the line, hence all points that are orthogonal with the line gives a dot product of zero with the line and a cross product that results in the line.

1.3

when $\lambda = 0$ then $\tilde{\mathbf{x_2}} = \mathbf{t}$ which means that x_2 is the projection of vector $\mathbf{t}$ which is the projection of the origin of the camera 1 - the epipole. How ever when $\lambda = \text{inf}$ then you get the entire epipolar line on camera 2.

1.4

Deriving the epipolar constraint by starting from:

$$\tilde{\mathbf{x}}_2 = \mathbf{R}\tilde{\mathbf{x}}_2\lambda + \mathbf{t}$$

Taking the cross product of t on both sides of the equality:

$$\mathbf{t} \times \tilde{\mathbf{x}}_2 = \mathbf{t} \times \mathbf{R}\tilde{\mathbf{x}}_1\lambda + \mathbf{t} \times \mathbf{t}$$

The cross product $\mathbf{t} \times \mathbf{t} = 0$ giving the following:

$$\mathbf{t} \times \tilde{\mathbf{x}}_2 = \mathbf{t} \times \mathbf{R}\tilde{\mathbf{x}}_1\lambda$$

Now we take the dot product of both sides of the equality:

$$\tilde{\mathbf{x}}_2 \cdot (\mathbf{t} \times \tilde{\mathbf{x}}_2) = \tilde{\mathbf{x}}_2 \cdot (\mathbf{t} \times \mathbf{R}\tilde{\mathbf{x}}_1\lambda)$$

Where we know that $\tilde{\mathbf{x}}_2 \cdot (\mathbf{t} \times \tilde{\mathbf{x}}_2) = 0$ as these two vectors are orthogonal, hence:

$$0 = \tilde{\mathbf{x}}_2 \cdot (\mathbf{t} \times \mathbf{R}\tilde{\mathbf{x}}_1\lambda)$$

Dot product can be written as leftmost vector transposed:

$$0 = \tilde{\mathbf{x}}_2^T [\mathbf{t}]_\mathbf{x} \mathbf{R}\tilde{\mathbf{x}}_1\lambda$$

Divide by λ:

$$0 = \tilde{\mathbf{x}}_2^T [\mathbf{t}]_\mathbf{x} \mathbf{R}\tilde{\mathbf{x}}_1$$

Since $E = [\mathbf{t}]_\mathbf{x}\mathbf{R}$ we get that:

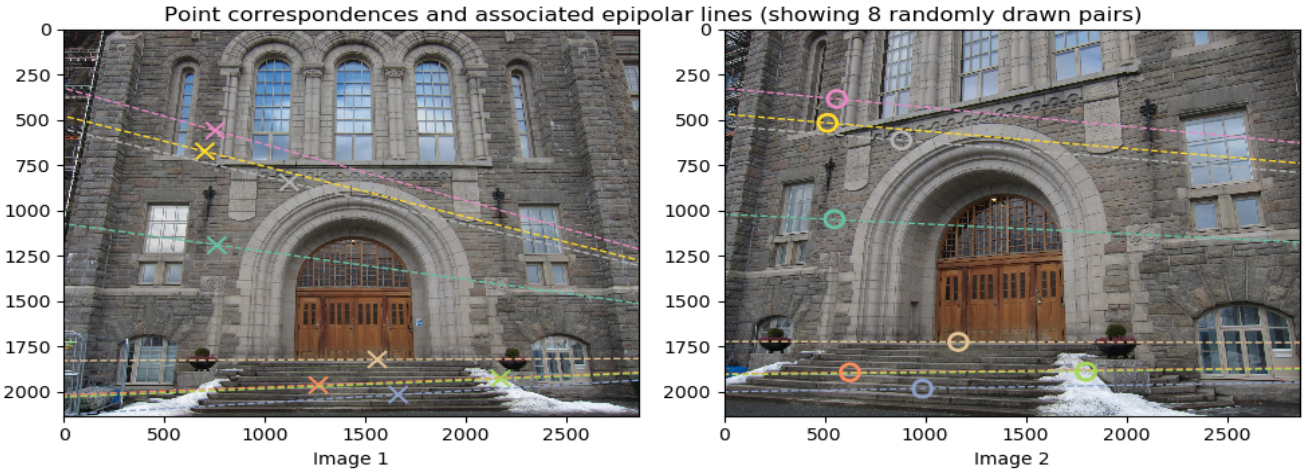
$$\tilde{\mathbf{x}}_2^T \mathbf{E}\tilde{\mathbf{x}}_1 = 0$$

1.5

Given that the correspondences are correct during movement and they are within the camera frame of both camera 1 and 2, then there is no reason that the essential matrix does not exist between the two images after small movement.

2 Part 2 - The 8-point algorithm

2.1



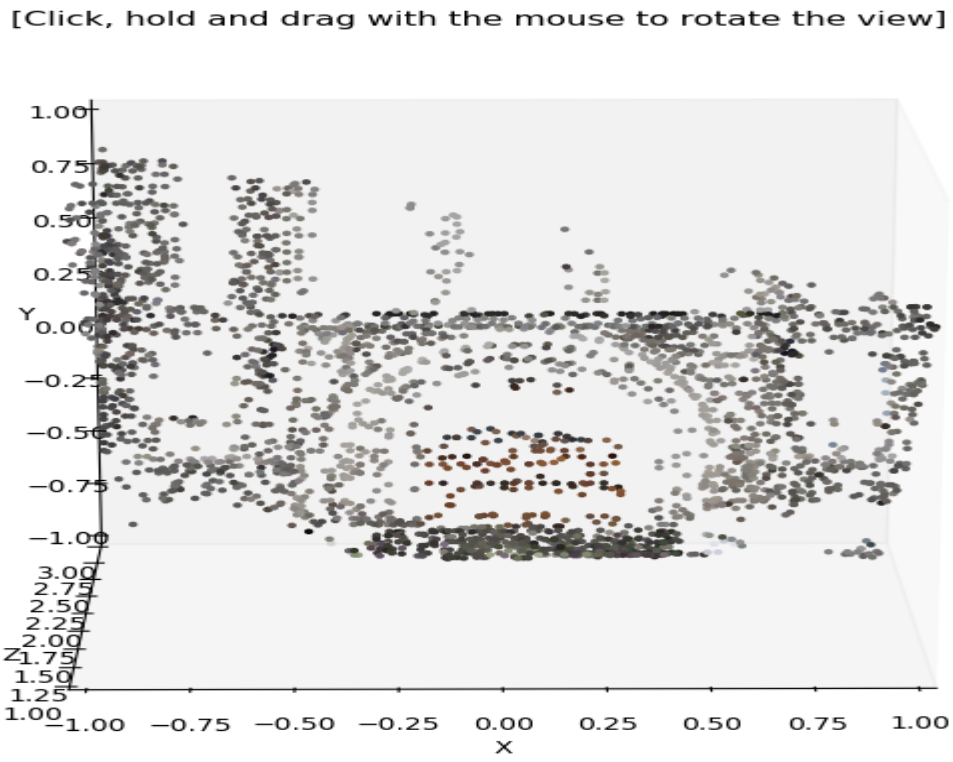
It seems that the 8-point algorithm performs generally very well, which is impressive for such an easy algorithm. How ever after running a few randomzied points i could clearly see an anomaly in the beige point as the point is displayed differently on image 1 and image 2. This made me question if the rest of them might have small errors but not as visible errors as well.

3 Part 3 - Triangulation and 3D reconstruction

3.1

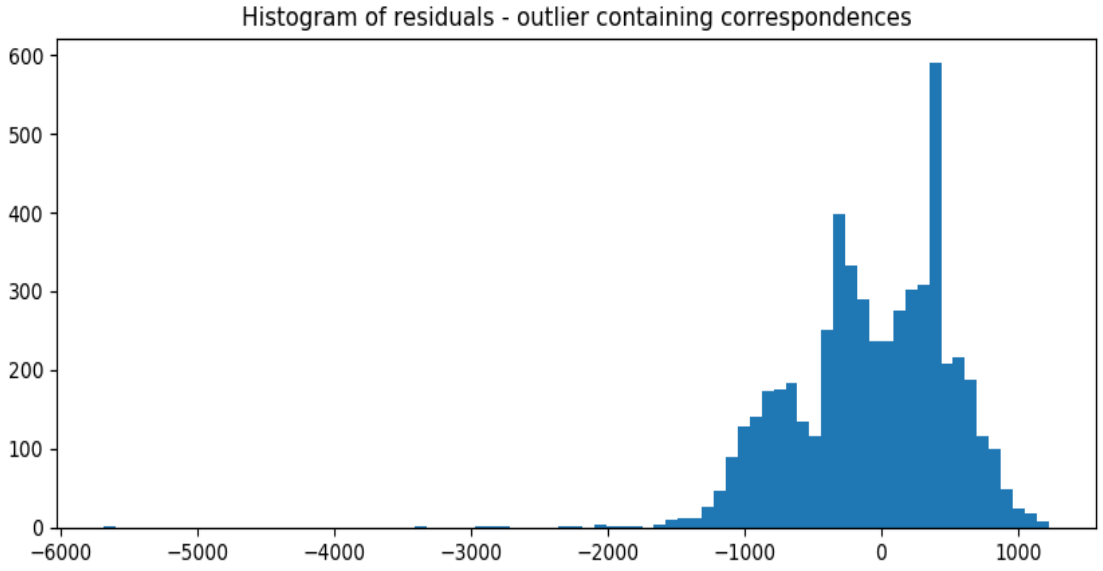
the $\tilde{W}$ is not necessarily close to zero for the SVD component. As one point might be far away from camera 1 in the z-axis but not as far away from the z-axis of camera 2. Using theoretical values of calibrated image coordinates in image 1 and 2 I discovered that if the z-axis is very big for both then the $\tilde{W}$ component of the SVD would be equal to $\tilde{Z}$ or one, which justifies my answer above of that the $\tilde{W}$ component of the SVD is not necessarily close to zero for points that are far away from the cameras.

3.2


[Click, hold and drag with the mouse to rotate the view]
1.00
0.75
0.50
0.25
Y 0.00
−0.25
−0.50
−0.75
−1.00
3.00
2.75
2.50
2.25
2.00
Z 1.75
1.50
1.25
1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00
X

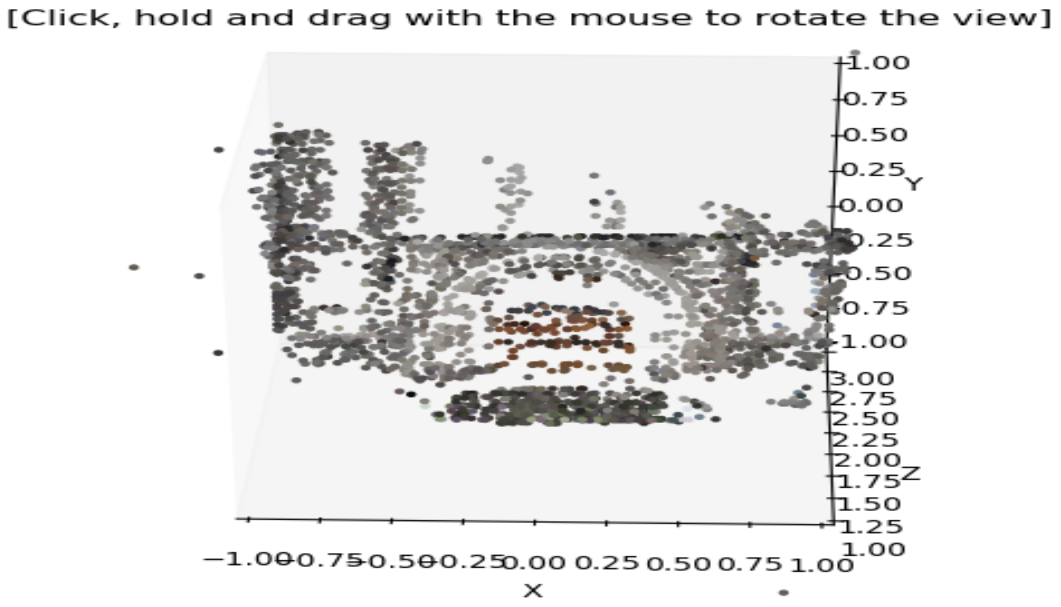
4 Part 4 - Robust Estimation with RANSAC

4.1



4.2

Size of the inlier set after 20000 iterations of RANSAC is 2853 with the resulting figure with only inliers is the following:



4.3

These anomalies are obviously outliers and they are by-passing the filteration of the RANSAC algorithm because the distance threshold is probably still too large (at 4 pixels). The simplest way to therefore be able to get rid of them is to keep testing out smaller distance thresholds that gets rid of as many outliers as possible. The problem then is that one can risk getting rid of points that are not outliers and that could give a more detailed cloud of the gate image.

4.4

This equation does not take into consideration the distance threshold which is important in determining what is an inlier set. How ever it is able to produce almost as large of inlier set with much fewer iterations with an inlier size of 2734 in 1077 iterations vs inlier size of 2853 in 20000 iterations.

This is when the probability per iteration is choosen to be (p = 0.5), k = 8 and probability of success of P =0.99. From testing different values of p it is easy to see that for big values of it we wont get 99% of the inlier set but for p values from 0.5 and lower, this can almost be guarenteed. The litterature further suggests that one should decrease the random sample size to decrease the time complexity.