# Computer Vision - FMAN85
# Assignment 3 - Spring 2020

## Epipolar Geometry: Fundamental and Essential Matrices

Hicham Mohamad, hi8826mo-s
hsmo@kth.se

March 8, 2020

## 1 Introduction

In this assignment, we are going to work on the basic elements of **Epipolar Geometry**. We will use the **Fundamental matrix** and the **Essential matrix** for simultaneously reconstructing the structure and the camera motion from two images.

The resulting plots and values in this report are obtained by running the implemented Matlab scripts `ass3_CE1_CE2.m` and `ass3_CE3_CE4.m`.

## 2 The Fundamental Matrix

A point in one view defines an **epipolar line** in the other view on which the corresponding point lies. The epipolar geometry depends only on the cameras, i.e. their relative position and their internal parameters. It does not depend at all on the scene structure. The epipolar geometry is represented by a $3 \times 3$ matrix called the fundamental matrix F.

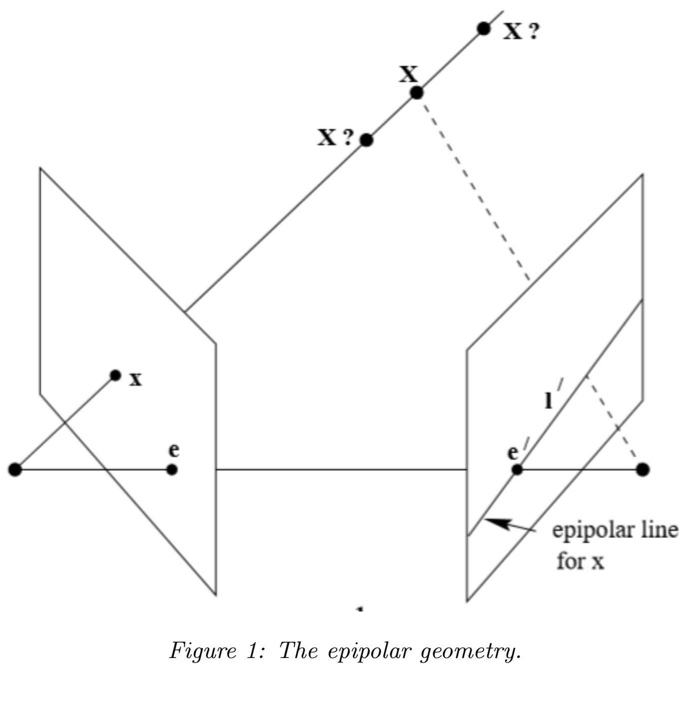



*Figure 1: The epipolar geometry.*

## 2.1 Exercise 1

### a) the fundamental matrix

In this task, we need to compute the fundamental matrix when $P_1 = [I \quad 0]$ and

$$P_2 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{1}$$

We know that the fundamental matrix can be computed by the relation

$$F = [t]_\times A = \begin{pmatrix} 0 & 0 & 2 \\ 0 & 0 & 0 \\ -2 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 2 \\ 0 & 0 & 0 \\ -2 & -2 & 0 \end{pmatrix} \tag{2}$$

where $A = P_2(:, 1:3)$ and $t = P_2(:, 4)$. The skew symmetric cross matrix can be calculated by

$$[t]_\times = \begin{pmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{pmatrix} \tag{3}$$

### b) The epipolar line

Assuming the point $x = (1, 1)$ is the projection of a 3D-point $\mathbf{X}$ into $P_1$, we need to compute the epipolar line in the second image generated from x

$$l_2 = [e_2]_\times Ax = [t]_\times Ax = Fx = \begin{pmatrix} 0 & 0 & 2 \\ 0 & 0 & 0 \\ -2 & -2 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ -4 \end{pmatrix} \tag{4}$$

### c) Projection of X into $P_2$

Here we need to investigate which of the points $x_1 = (2, 0)$, $x_2 = (2, 1)$ and $x_3 = (4, 2)$ could be a projection of the same point $\mathbf{X}$ into $P_2$. In other words, these points should be on the epipolar line $l_2$, i.e.

$$l_2^T x_1 = \begin{pmatrix} 2 & 0 & 4 \end{pmatrix} \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix} = 0 \tag{5}$$

$$l_2^T x_2 = \begin{pmatrix} 2 & 0 & 4 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} = 0 \tag{6}$$

$$l_2^T x_3 = \begin{pmatrix} 2 & 0 & 4 \end{pmatrix} \begin{pmatrix} 4 \\ 2 \\ 1 \end{pmatrix} = 4 \neq 0 \tag{7}$$

From these results we can say that the points $x_1$ and $x_2$ only can be two projections of $\mathbf{X}$ into $P_2$.

## 2.2 Exercise 2

In this task, we need to compute the epipoles and the fundamental matrix when $P_1 = [I \quad 0]$ and

$$P_2 = \begin{pmatrix} 1 & 1 & 1 & 2 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{8}$$

**a) The Epipoles**

By projecting the camera centers, we can write

$$P_1\mathbf{C}_1 = 0 \implies \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} C_1 \\ 1 \end{bmatrix} = 0 \implies C_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{9}$$

The same can be done for the center $\mathbf{C}_2$

$$P_2\mathbf{C}_2 = 0 \implies \begin{bmatrix} A & t \end{bmatrix} \begin{bmatrix} C_2 \\ 1 \end{bmatrix} = 0 \implies AC_2 + t = 0 \implies C_2 = -A^{-1}t \tag{10}$$

Now we can compute the epipoles by writing

$$e_1 \sim P_1\mathbf{C}_2 \implies e_1 \sim \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} C_2 \\ 1 \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} -A^{-1}t \\ 1 \end{bmatrix} \implies e_1 \sim -A^{-1}t = -\begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 2 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ 0 \end{pmatrix} \tag{11}$$

The same for $e_2$

$$e_2 \sim P_2\mathbf{C}_1 \implies e_2 \sim \begin{bmatrix} A & t \end{bmatrix} \begin{bmatrix} C_1 \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \implies e_2 \sim t = \begin{pmatrix} 2 \\ 2 \\ 0 \end{pmatrix} \tag{12}$$

**b) The fundamental matrix**

Here we need to compute the fundamental matrix, its determinant and verify that $e_2^T F = 0$ and $Fe_1 = 0$. The fundamental matrix can be computed by

$$F = [t]_\times A = \begin{pmatrix} 0 & 0 & 2 \\ 0 & 0 & -2 \\ -2 & 2 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 2 \\ 0 & 0 & -2 \\ -2 & 2 & -2 \end{pmatrix} \tag{13}$$

Then we can see that $det(F) = 0$ and

$$e_2^T F = \begin{pmatrix} 2 & 2 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 2 \\ 0 & 0 & -2 \\ -2 & 2 & -2 \end{pmatrix} = 0 \tag{14}$$

$$Fe_1 = \begin{pmatrix} 0 & 0 & 2 \\ 0 & 0 & -2 \\ -2 & 2 & -2 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \\ 0 \end{pmatrix} = 0 \tag{15}$$

## 2.3 Exercise 3

When computing the fundamental matrix $F$ using the 8-point algorithm it is recommended to use normalization. Suppose the image points have been normalized using

$$\tilde{\mathbf{x}}_1 \sim N_1\mathbf{x}_1 \quad \text{and } \tilde{\mathbf{x}}_2 \sim N_2\mathbf{x}_2 \tag{16}$$

If $\tilde{F}$ fulfills $\tilde{\mathbf{x}}_2^T \tilde{F} \tilde{\mathbf{x}}_1 = 0$, we can find the fundamental matrix $F$ that fulfills $\mathbf{x}_2^T F \mathbf{x}_1 = 0$ for the original (un-normalized) points

$$\tilde{\mathbf{x}}_2^T \tilde{F} \tilde{\mathbf{x}}_1 = 0 \implies \mathbf{x}_2^T N_2^T \tilde{F} N_1 \mathbf{x}_1 \implies F = N_2^T \tilde{F} N_1 \tag{17}$$

## 2.4 Computer Exercise 1

In this exercise you will compute the fundamental matrix for the two images of a part of the fort Kronan in Gothenburg. The file `compEx1data.mat` contains a cell $x$ with matched points for the two images. We summarize the different steps of the implemented Eight Point algorithm here:

- Extract at least 8 point correspondences.

- Normalize the coordinates.
- Form M and solve the minimization problem using svd

$$\min_{||v||^2=1} ||Mv||^2 \tag{18}$$

- Form the matrix F and ensure that $det(F) = 0$.
- Transform back to the original (un-normalized) coordinates.
- Compute $P_2$ from $F$.
- Compute the scene points using triangulation

### a) Normalizing

First we compute in Matlab the normalization matrices $N_1$ and $N_2$. These matrices should subtract the mean and re-scale using the standard deviation, as in assignment 2. The we normalize the image points of the two images with $N_1$ and $N_2$ respectively.

### b) Setting up Matrix M

Next we set up the matrix M in the eight point algorithm where we use all the points, and then we solve the homogeneous least squares system using SVD. After checking the values in Matlab we verify that the minimum singular value and $||Mv||$ are both small.

### c) Normalized F

We construct the normalized fundamental matrix from the solution $v$. We remember to make sure that $det(\tilde{F} = 0$ for your solution, and to check that the epipolar constraints $\tilde{\mathbf{x}}_2^T F \tilde{\mathbf{x}}_1 = 0$ are roughly fulfilled, as shown in Figure 2.

The resulting $F$ may not have $det(F) = 0$. Thus we need to pick the closest matrix A with $det(A) = 0$. This can be solved using svd, in matlab:

```
[U,S,V] = svd(F);
S(3,3) = 0;
A = U*S*V';
```

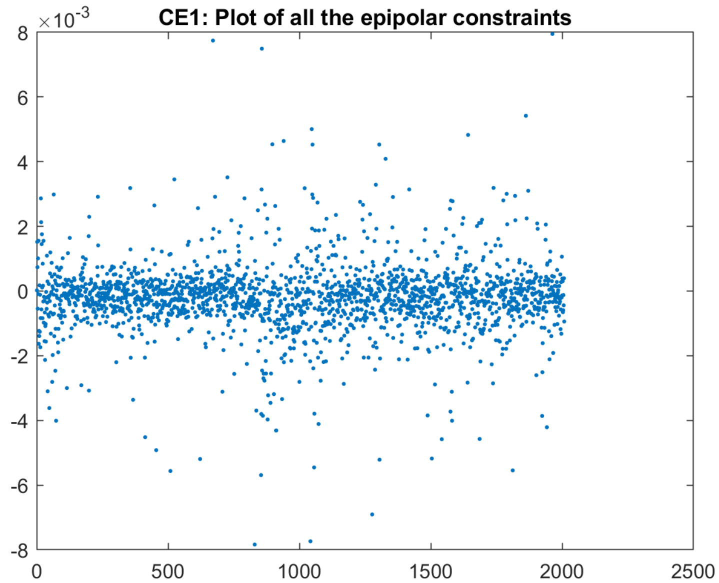



*Figure 2: plot of all the epipolar constraints. As we notice in the plot they look to be roughly 0.*

### d) Un-normalized F and epipolar lines

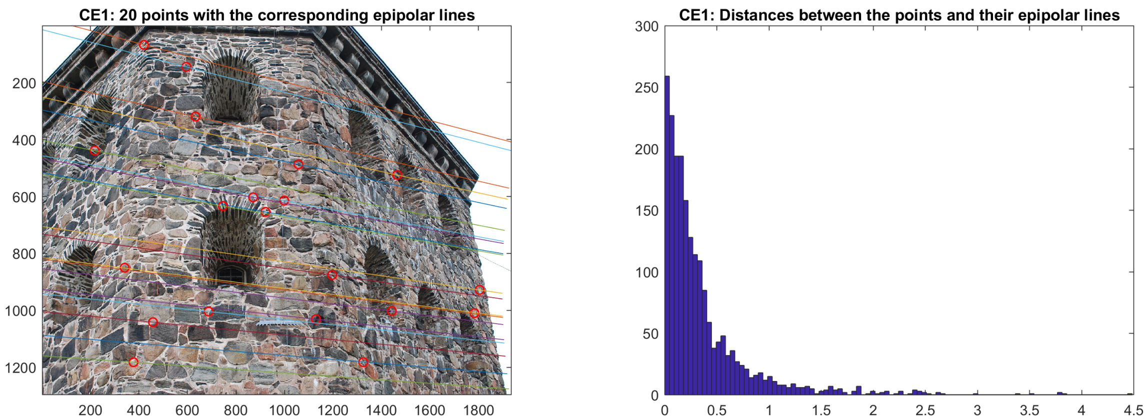
In this step, we compute the un-normalized fundamental matrix $F$ (using the formula from exercise 3) and the epipolar lines $l = F\mathbf{x}_1$.

$$F = \begin{pmatrix} -0.0000 & -0.0000 & 0.0058 \\ 0.0000 & 0.0000 & -0.0267 \\ -0.0072 & 0.0263 & 1 \end{pmatrix} \quad \text{where } \det(F) = -6.1873e-27 \tag{19}$$

Then we pick 20 points in the second image at random and plot these in the same figure as the image. Also we plot the corresponding epipolar lines in the same image using the function `rital.m`. As illustrated in Figure 3a, it is clear that they are close to each other.

### e) Distance

At the end, we compute the distance between all the points and their corresponding epipolar lines and plot these in a histogram with 100 bins, as shown in Figure 3b. The mean distance is 0.3612.




(a) (b)

*Figure 3: a) Plot of 20 points picked at random with the corresponding epipolar lines in the second image. b) All the distances between the 20 points and their corresponding epipolar lines.*

## 2.5 Exercise 4

In this task, we consider the fundamental matrix

$$F = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \tag{20}$$

We need to verify that the projection of the scene points (1,2,3) and (3,2,1)in the cameras $P_1 = [I \quad 0]$ and $P_2 = [[e_2]_\times F e_2]$, fulfill the epipolar constraint ($x_2^T F x_1 = 0$).

$$P_2 = \big[[e_2]_\times F \quad e_2\big] = \begin{pmatrix} -1 & 0 & 0 & -1 \\ 0 & 2 & 2 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix} = \big[A \quad t\big] \tag{21}$$

where

$$e_2 = t = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}, \quad [e_2]_\times = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}, \quad [e_2]_\times F = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 2 & 2 \\ -1 & 0 & 0 \end{pmatrix} = A \tag{22}$$

Now we can write

$$\mathbf{x}_1 \sim P_1 \mathbf{X}_1 = \big[I \quad 0\big]\, \mathbf{X}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \\ 1 \end{pmatrix} \tag{23}$$

we can also write

$$\mathbf{x}_2 \sim P_2\mathbf{X}_2 = \begin{pmatrix} -1 & 0 & 0 & -1 \\ 0 & 2 & 2 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 2 \\ 1 \\ 1 \end{pmatrix} \tag{24}$$

At the end we get that the epipolar constraint fulfills, i.e. $\mathbf{x}_2^T F\mathbf{x}_1 = 0$.

**Camera center of $P_2$**

The camera center of $P_2$ is given by

$$P_2\mathbf{C}_2 = 0 = \begin{bmatrix} [e_2]_\times F & e_2 \end{bmatrix} \begin{bmatrix} C_2 \\ \rho \end{bmatrix} \implies [e_2]_\times FC_2 + e_2\rho = 0 \tag{25}$$

$$\implies \begin{cases} [e_2]_\times FC_2 = 0 \\ e_2\rho = 0 \end{cases} \implies \begin{cases} C_2 = e_1 \\ \rho = 0 \end{cases} \tag{26}$$

This gives that

$$\mathbf{C}_2 = \begin{pmatrix} e_1 \\ 0 \end{pmatrix} \tag{27}$$

which it is a center at infinity and the epipole $e_1$ is given by

$$e_1 \sim P_1\mathbf{C}_2 = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} -A^{-1}t \\ 0 \end{bmatrix} = -A^{-1}t = -\begin{pmatrix} -1 & 0 & 0 \\ 0 & 2 & 2 \\ -1 & 0 & 0 \end{pmatrix}^{-1} \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} \tag{28}$$

Here we see that $A$ is invertible, so we can say that $An = 0$ and compute in Matlab

$$null(A) = n = \begin{pmatrix} 0 \\ -0.7071 \\ 0.7071 \end{pmatrix} \implies \mathbf{C}_2 = \begin{pmatrix} n \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ -0.7071 \\ 0.7071 \\ 0 \end{pmatrix} \tag{29}$$

this result can also be obtained by computing $null(P_2)$ in Matlab getting the same.

## 2.6 Computer Exercise 2

In this exercise we should use the fundamental matrix $F$ obtained in Computer Exercise 1 to compute the camera matrices in Exercise 4. After running the implemented code in Matlab we get the resulting camera matrices

$$P1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad P2 = \begin{pmatrix} -0.0016 & 0.0057 & 0.2163 & 0.9763 \\ 0.0070 & -0.0257 & -0.9763 & 0.2163 \\ 0.0000 & 0.0000 & -0.0273 & 0.0001 \end{pmatrix} \tag{30}$$

Then we need to use triangulation with DLT to compute the 3D-points. A plot of both the image, the image points, and the projected 3D points is illustrated in Figure 4.

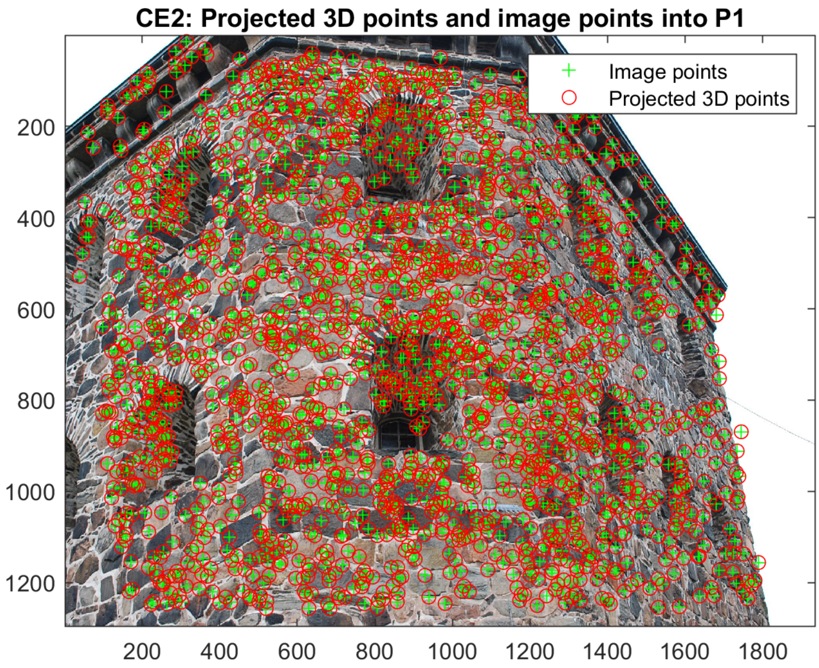



*Figure 4: Plot of both the image, the image points, and the projected 3D points. After using the fundamental matrix F obtained in Computer Exercise 1, we compute the camera matrices in Exercise 4, then we use triangulation with DLT to compute the 3D-points.*

When triangulating, we should remember to normalize . Since the point sets are the same as in Computer Exercise 1 the normalization matrices are the same. Alternatively one could compute cameras and 3D points from the fundamental matrix $\tilde{F}$ obtained with the normalized points and transform the cameras afterwards. This also gives a valid solution, but it is a different one.

When plotting the 3D-points in a 3D plot, it doesn't look like what we expect, it seems that we have a distortion problem !

## 3 The Essential Matrix

The fundamental matrix corresponding to the pair of **normalized** cameras is customarily called the essential matrix. Thus, the fundamental matrix for a pair of cameras of the form $[I \quad 0]$ and $[R \quad t]$ is given by

$$E = [t]_{\times} R \tag{31}$$

and is called the Essential matrix. In addition to having $det(E) = 0$ the two non-zero **singular values** have to be equal. Furthermore, since the scale is arbitrary we can assume that these singular values are both 1. Therefore E has the SVD

$$E = Udiag([1 \quad 1 \quad 0])V^T \tag{32}$$

We can summarize the steps of the modified 8-point algorithm here:

- Extract at least 8 point correspondences.
- Normalize the coordinates using $K_1^{-1}$ and $K_2^{-1}$ where $K_1$ and $K_2$ are the inner parameters of the cameras.
- Form M and solve the minimization problem using SVD

$$\min_{||v||^2=1} ||Mv||^2 \tag{33}$$

- Form the matrix E and ensure that E has the singular values 1,1,0.
- Compute $P_2$ from E.
- Compute the scene points using triangulation.

## 3.1 Computer Exercise 3

In this task, the file `compEx3data.mat` contains the calibration matrix $K$ for the two images in Computer Exercise 1. First we normalize the image points using the inverse of $K$.

then we set up the matrix $M$ in the eight point algorithm, and solve the homogeneous least squares system using SVD. After that we check that the minimum singular value and $Mv$ are both small.

### 3.1.1 The Essential Matrix

Next, we construct the Essential matrix from the solution $v$. We should remember to make sure that $E$ has two equal singular values and the third one zero. Also we check that the epipolar constraints $\tilde{\mathbf{x}}_2^T E \tilde{\mathbf{x}}_1 = 0$ are roughly fulfilled, as shown in Figure 5. The resulting essential matrix is

$$E = E./E(3,3) = \begin{pmatrix} -8.9 & -1005.8 & 377.1 \\ 1252.5 & 78.4 & -2448.2 \\ -472.8 & 2550.2 & 1.0 \end{pmatrix} \tag{34}$$

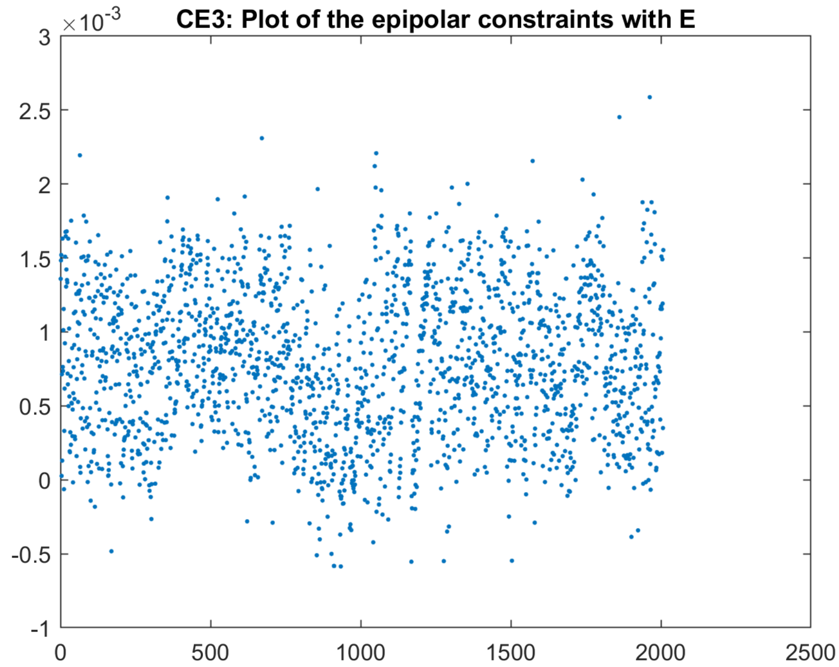



*Figure 5: Plot of all the epipolar constraints with the essential matrix.*

### 3.1.2 Un-normalized Fundamental Matrix

The defining equation for the essential matrix is

$$\tilde{x_2} E \tilde{x_1} = 0 \tag{35}$$

in terms of the normalized image coordinates for corresponding points $x_1 \leftrightarrow x_2$. Substituting for $\tilde{x_1}$ and $\tilde{x_2}$ gives

$$x_2^T K_2^{-T} E k^{-1} x_1 = 0 \tag{36}$$

Comparing this with the relation

$$x_2 F x_1 = 0 \tag{37}$$

for the fundamental matrix, it follows that the relationship between the fundamental and essential matrices is

$$E = K_2^T F K_1 \implies F = K_2^{-T} E K_1^{-1} \tag{38}$$

In this way, we get the un-normalize fundamental matrix

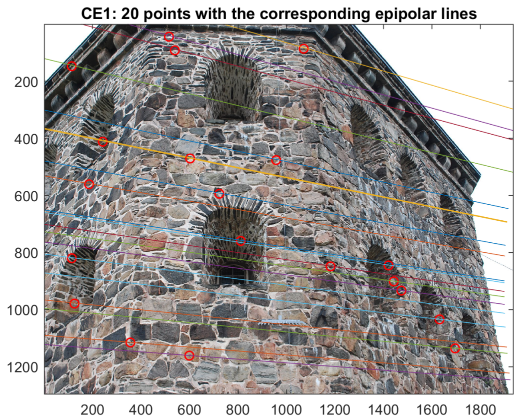
$$F = \begin{pmatrix} 0.0000 & 0.0000 & -0.0001 \\ -0.0000 & -0.0000 & 0.0004 \\ 0.0001 & -0.0004 & -0.0147 \end{pmatrix} \tag{39}$$

After computing the fundamental matrix for the un-normalized coordinate system from the essential matrix, we compute the epipolar lines $l = F\mathbf{x}_1$. Then we pick 20 of the detected points in the second image at random and plot these in the same figure as the image, as illustrated in Figure 6a. Also we plot the corresponding epipolar lines in the same figure using the function `rital.m`.

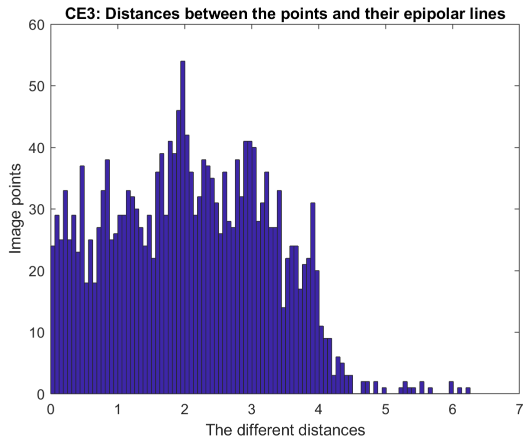
#### 3.1.3 Distance and Histogram

Then we compute the distance between the points and their corresponding epipolar lines and plot these in a histogram with 100 bins, as shown in Figure 6b.

How does this result compare to the corresponding result Computer Exercise 1???



(a)



(b)

*Figure 6: a) Plot of 20 points picked at random with the corresponding epipolar lines in the second image. b) All the distances between the points and their corresponding epipolar lines.*

### 3.2 Exercise 6

An essential matrix has the singular value decomposition

$$E = Udiag(\begin{bmatrix}1 & 1 & 0\end{bmatrix})V^T \tag{40}$$

$$U = \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \text{and } V = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \tag{41}$$

By computing

$$UV^T = \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 0 & -1 & 0 \end{pmatrix} \tag{42}$$

we can verify that $det(UV^T) = 1$.

#### 3.2.1 The essential matrix

We compute the essential matrix

$$E = Udiag(\begin{bmatrix}1 & 1 & 0\end{bmatrix})V^T = \begin{pmatrix} 1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 0 & 0 & 0 \end{pmatrix} \tag{43}$$

Then we can verify that $x_1 = (0,0)$(in camera 1) and $x_2 = (1,1)$(in camera 2) is a plausible correspondence by computing

$$\mathbf{x}_2 E \mathbf{x}_1 = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = 0 \tag{44}$$

If $x_1$ is the projection of $\mathbf{X}$ in $P_1 = [I \quad 0]$ we need to show that $\mathbf{X}$ must be one of the points

$$\mathbf{X}(s) = \begin{pmatrix} 0 \\ 0 \\ 1 \\ s \end{pmatrix} \tag{45}$$

In fact we can write

$$\mathbf{x}_1 \sim P_1 \mathbf{X}(s) = \begin{bmatrix} I & 0 \end{bmatrix} \mathbf{X}(s) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{46}$$

#### 3.2.2 Scale Amabiguity

For each of the solutions

$$P_2 = [UWV^T \quad u_3], \quad [UWV^T \quad -u_3], \quad [UW^TV^T \quad u_3], \quad [UW^TV^T \quad -u_3] \tag{47}$$

where

$$W = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{48}$$

and $u_3$ is the third column of $U$, we need to compute $s$ such that $\mathbf{X}(s)$ projects to $x_2$.

**Solution 1:** $P_2 = [UWV^T \quad u_3]$

$$\mathbf{x}_2 \sim P_2 \mathbf{X}(s) \implies \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ s \end{pmatrix} = \begin{pmatrix} -1/\sqrt{2} \\ -1/\sqrt{2} \\ s \end{pmatrix} = -1/\sqrt{2} \begin{pmatrix} 1 \\ 1 \\ -\sqrt{2}s \end{pmatrix} \tag{49}$$

which gives that $s = -1/\sqrt{2}$

**Solution 2:** $P_2 = [UWV^T \quad -u_3]$

$$\mathbf{x}_2 \sim P_2 \mathbf{X}(s) \implies \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ 0 & -1 & 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ s \end{pmatrix} = \begin{pmatrix} -1/\sqrt{2} \\ -1/\sqrt{2} \\ -s \end{pmatrix} = -1/\sqrt{2} \begin{pmatrix} 1 \\ 1 \\ \sqrt{2}s \end{pmatrix} \tag{50}$$

which gives that $s = 1/\sqrt{2}$

**Solution 3:** $P_2 = [UW^TV^T \quad u_3]$

$$\mathbf{x}_2 \sim P_2 \mathbf{X}(s) \implies \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ s \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ s \end{pmatrix} = 1/\sqrt{2} \begin{pmatrix} 1 \\ 1 \\ \sqrt{2}s \end{pmatrix} \tag{51}$$

which gives that $s = 1/\sqrt{2}$

**Solution 4:** $P_2 = [UW^TV^T \quad -u_3]$

$$\mathbf{x}_2 \sim P_2 \mathbf{X}(s) \implies \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 0 & -1 & 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ s \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ -s \end{pmatrix} = 1/\sqrt{2} \begin{pmatrix} 1 \\ 1 \\ -\sqrt{2}s \end{pmatrix} \tag{52}$$

which gives that $s = -1/\sqrt{2}$

Now we need to investigate for which of the camera pairs is the 3D point $\mathbf{X}(s)$ in front of both cameras. To do this it is enough to look at the third coordinate of $P\mathbf{X}(s)$, if it is positive for both $P_1$ and $P_2$ we can conclude that $\mathbf{X}(s)$ is in front of both cameras.

Computing is done for camera $P_2$ and we saw that only solutions 3 and 4 fulfil this condition, i.e. $P_2\mathbf{X}$ has a positive third coordinate.

For camera $P_1$ we have seen before that

$$\mathbf{x}_1 \sim P_1\mathbf{X}(s) = \begin{bmatrix} I & 0 \end{bmatrix} \mathbf{X}(s) \implies \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ s \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1/s \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1/s \end{pmatrix} \tag{53}$$

As we can see, the projection of $X(s)$ in $P_1$ has $1/s$ at the third coordinate. Consequently we can conclude that for camera $P_1$ we need a positive $s$ which correspond to the computed values of $s$ in solutions 2 and 3 where we have $s > 0$.

**Conclusion:** At the end, we can conclude that only for solution 3 the 3D point $X(s)$ is in front of both cameras, because it fulfills the right condition for both cameras $P_1$ and $P_2$.

## 3.3 Computer Exercise 4

In this task, we need to compute the following four camera solutions

$$P_2 = [UWV^T \quad u_3], \quad [UWV^T \quad -u_3], \quad [UW^TV^T \quad u_3], \quad [UW^TV^T \quad -u_3] \tag{54}$$

for the **essential matrix** obtained in Computer Exercise 3, where

$$W = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{55}$$

### 3.3.1 Computing the four camera solutions

After implementing this in Matlab as in Listing 1, we get the four camera solutions $P_2$

$$P_{21} = \begin{pmatrix} 0.9944 & 0.0293 & 0.1017 & 0.9223 \\ -0.0311 & 0.9994 & 0.0161 & 0.1422 \\ -0.1011 & -0.0191 & 0.9947 & 0.3594 \end{pmatrix}, \quad P_{22} \begin{pmatrix} 0.9944 & 0.0293 & 0.1017 & -0.9223 \\ -0.0311 & 0.9994 & 0.0161 & -0.1422 \\ -0.1011 & -0.0191 & 0.9947 & -0.3594 \end{pmatrix} \tag{56}$$

$$P_{23} = \begin{pmatrix} 0.6221 & 0.2700 & 0.7349 & 0.9223 \\ 0.2804 & -0.9532 & 0.1129 & 0.1422 \\ 0.7310 & 0.1358 & -0.6687 & 0.3594 \end{pmatrix}, \quad P_{24} \begin{pmatrix} 0.6221 & 0.2700 & 0.7349 & -0.9223 \\ 0.2804 & -0.9532 & 0.1129 & -0.1422 \\ 0.7310 & 0.1358 & -0.6687 & -0.3594 \end{pmatrix} \tag{57}$$

*Listing 1: Matlab code for computing the four cameras P2 for the essential matrix E*

```
% compute the four camera solutions P2
P2{1,1} = [EU*W*EV' EU(:,end)];
P2{2,1} = [EU*W*EV' -EU(:,end)];
P2{3,1} = [EU*W'*EV' EU(:,end)];
P2{4,1} = [EU*W'*EV' -EU(:,end)];
```

### 3.3.2 Triangulating the points using DLT

We triangulate the points using DLT for each of the four camera solutions, as shown in Listing 2

*Listing 2: Triangulation the points using DLT for each of the 4 camera solutions.*

```
% Set up the DLT equations for triangulation
X = {};
```

```
for i=1:4
    Xpoints = [];
    for j=1:size(x{1},2)
        Mtriang = [P1 -[xn{1}(:,j)] [0 0 0]';
                   P2{i,1} [0 0 0]' -[xn{2}(:,j)]];

        % and solve the homogeneous least squares system
        % do this in a loop, once for each point
        [EU,ES,EV] = svd(Mtriang);
        vstar = EV(:,end);
        Xpoints = [Xpoints vstar(1:4,:)];
    end
    X{i,1} = Xpoints;
end
```

#### 3.3.3 Solutions with points in front of the cameras

Here we need to determine for which of the solutions the points are in front of the cameras. Since there is noise involved it might not be possible to find a solution with all points in front of the cameras. Thus we select the one with the highest number of points in front of the cameras.

After running the implemented code in Listing 3, we conclude that The **solution 2** is with the highest number of points in front of the cameras $P_1$ and $P_2$.

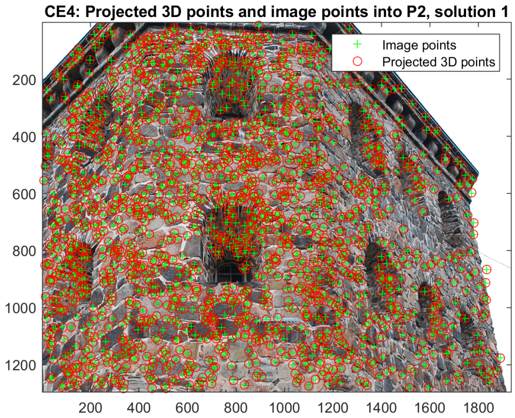
*Listing 3: Finding the solution with points in front of the cameras.*

```
[m,n] = size(X{1});
homX = {};
isInFrontOf = {};

for j=1:4
    homXpoints = zeros(m,n);
    inFrontInd = [];

    for i=1:n
        homXpoints(:,i) = pflat(X{j}(:,i));

        % determine for which of the solutions the
        % points are in front fo the cameras

        % If both P1(3,:)*homX AND P2{i}(3,:)*homX are positive
        % then the point is in front of both cameras.
        % Check if P1(3,:)*homX{j} AND P2{i}(3,:)*homX{j}.
        if (P1(3,:)*homXpoints(:,i)>0) && (P2{j}(3,:)*homXpoints(:,i)>0)
            inFrontInd = [inFrontInd i];
        end
    end
    homX{j,1} = homXpoints;
    isInFrontOf{j,1} = inFrontInd;

end
for i=1:4
    sz(i) = size(isInFrontOf{i},2);
end
    [~,highest] = max(sz);
disp(['The solution ', num2str(highest),+...
      ' is with the highest number of points ' +...
      'in front of the cameras.'])
```

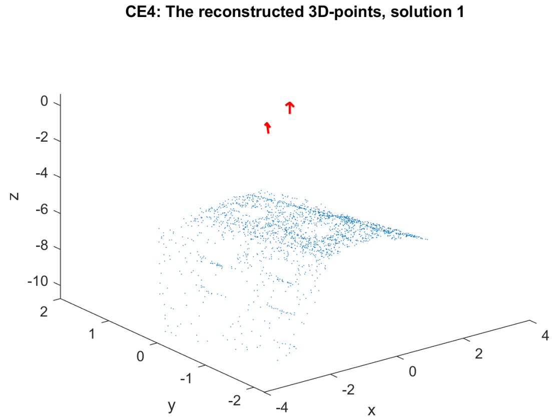
### 3.3.4 Plotting image, 3D points and cameras

We compute the corresponding camera matrices for the original (un-normalized) coordinate system and plot the image, the points and the projected 3D-points in the same figure, as shown in the resulting figures below. *We see that the errors look small.*

Also we plot the 3D points and camera centers and principal axes in a 3D plot. As expected we notice that for solution 2 the 3D points are in front of the cameras, as shown in Figure 8b. In addition, we can notice that by using calibrated cameras we don't get any distortions because such a solution is *up to a similarity transformation.*

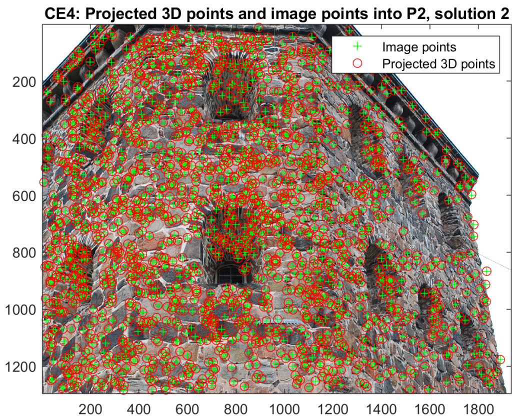



(a)

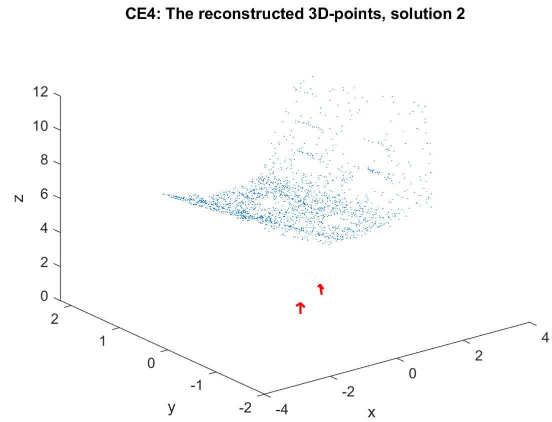



(b)

*Figure 7: Solution 1. a) Plot of both the image, the image points, and the projected 3D points. After using the essential matrix E obtained in Computer Exercise 3, we compute the camera matrices in Exercise 6, then we use triangulation with DLT to compute the 3D-points. b) Plot of the 3D-points and the camera centers in the same 3D plot.*

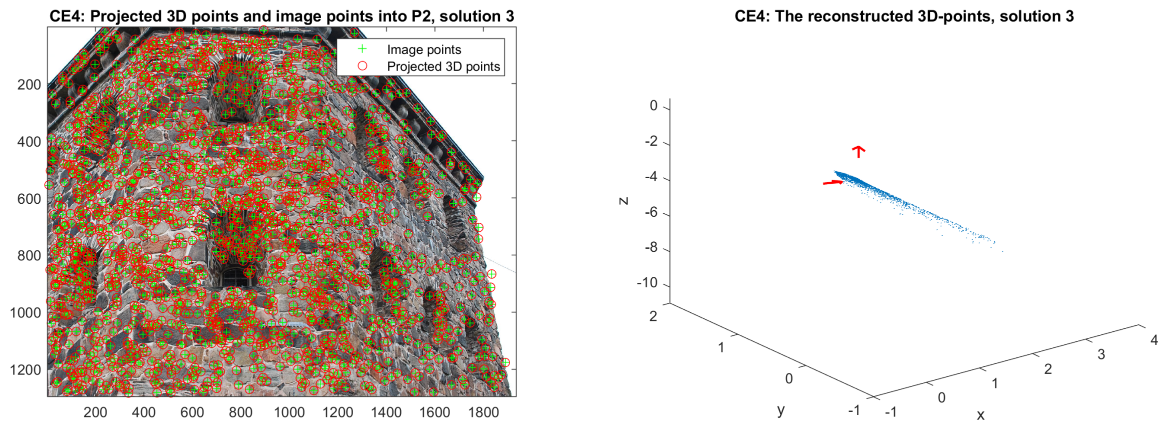



(a)



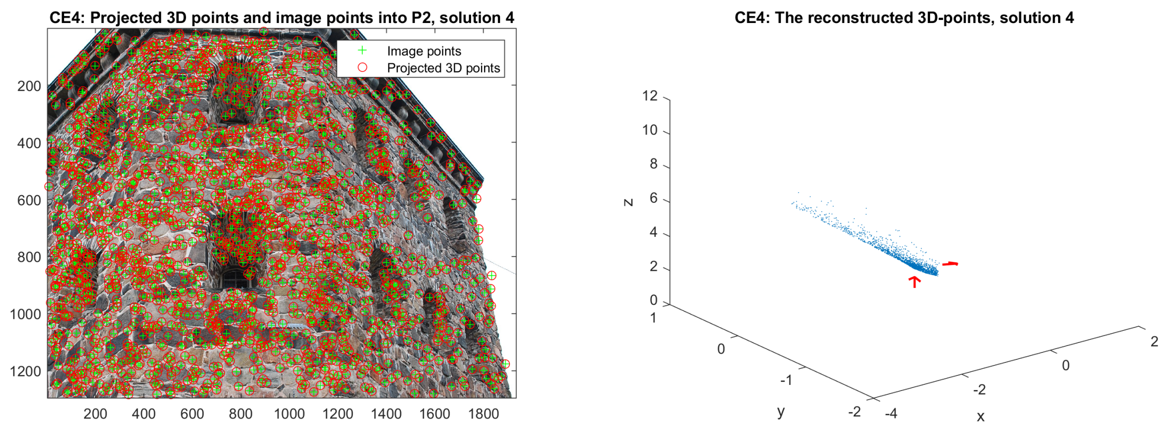

(b)

*Figure 8: Solution 2. a) Plot of both the image, the image points, and the projected 3D points. After using the essential matrix F obtained in Computer Exercise 3, we compute the camera matrices in Exercise 6, then we use triangulation with DLT to compute the 3D-points. b) Plot of the 3D-points and the camera centers in the same 3D plot.*




(a) (b)

*Figure 9: Solution3. a) Plot of both the image, the image points, and the projected 3D points. After using the essential matrix E obtained in Computer Exercise 3, we compute the camera matrices in Exercise 6, then we use triangulation with DLT to compute the 3D-points. b) Plot of the 3D-points and the camera centers in the same 3D plot.*




(a) (b)

*Figure 10: Solution 4. a) Plot of both the image, the image points, and the projected 3D points. After using the essential matrix E obtained in Computer Exercise 3, we compute the camera matrices in Exercise 6, then we use triangulation with DLT to compute the 3D-points. b) Plot of the 3D-points and the camera centers in the same 3D plot.*

# References


[1] Carl Olsson, Computer Vision - FMAN85, Lectures notes: `https://canvas.education.lu.se/courses/3379`

[2] Hartley, Zisserman, Multiple View Geometry, 2004.

[3] Szeliski, Computer Vision - Algorithms and Applications, Springer.