NTNU
Kunnskap for en bedre verden

Department of Engineering Cybernetics

TTK4255 - Robotic Vision

Homework 8: Two-view geometry

Zahra Parvinashtiani

22.03.2024

Table of Contents

List of Figures

1 Part 1: The epipolar constraint

1.1 Part 1.1

a) $K\tilde{\ell} = (Ka, Kb, Kc)$ $\quad ax + by + cz = 0 \rightarrow Ka \cdot x + Kb \cdot y + Kc \cdot z = 0$

K non zero scalar $\quad \rightarrow a \cdot x + b \cdot y + c \cdot z = 0$ $\quad K\tilde{\ell}$ represents the same line as $\tilde{\ell}$

b) $\theta = \operatorname{atan2}(b, a)$

$\rho = -c / \sqrt{a^2 + b^2}$ $\quad \tilde{\ell}_{norm} = (\cos\theta, \sin\theta, -\rho)$

Figure 1: Answers for the part 1.1.

1.2 Part 1.2

$\tilde{x}_a = (x_a, y_a, 1) \quad \tilde{x}_b = (x_b, y_b, 1)$

$$\tilde{\ell} = \tilde{x}_a \times \tilde{x}_b = \begin{bmatrix} y_a \cdot 1 - 1 \cdot y_b \\ 1 \cdot x_b - x_a \cdot 1 \\ x_a \cdot y_b - y_a \cdot x_b \end{bmatrix} = \begin{bmatrix} y_a - y_b \\ x_b - x_a \\ x_a \cdot y_b - y_a \cdot x_b \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

$\tilde{x} = (x, y, 1)$ if lies on this line $\rightarrow$ satisfies $\tilde{\ell}^T \tilde{x} = 0$

Figure 2: Answers for the part 1.2.

1.3 Part 1.3

$$\begin{cases} X^1(\lambda) = \lambda B_1^1 : \text{the back-projection ray from camera 1} \\ X^2(\lambda) = R X^1(\lambda) + t : \text{the point's coordinates in camera 2} \\ \tilde{u}(\lambda) = K_2 X^2(\lambda) : \text{the image coordinates in image 2} \end{cases}$$

for $\lambda \to 0$, $X^1(\lambda) \to$ origin of camera 1 $\to X^2(0) = R \cdot 0 + t = t \to \tilde{u}(0) = K_2 \cdot t$

$u'(\lambda)$ as $\lambda \to 0$ projection of the translation vector t onto the image plane of camera 2, which corresponds to the epiple in image 2.

for $\lambda \to \infty$, $X^1(\lambda) \to$ back-projection ray $\to \begin{cases} X^2(\lambda) \approx \lambda R K_1^{-1} \tilde{u}_1 \\ \tilde{u}(\lambda) \approx \lambda K_2 R K_1^{-1} \tilde{u}_1 \end{cases}$

Figure 3: Answers for the part 1.3.

Interpretation of Limit Values:

$\lambda \to 0$: This represents the image of the point where the ray originating from camera 1's optical center intersects the epipole.

$\lambda \to \infty$: The coordinates going to infinity represent the direction of the back-projection ray extending to infinity. In practical terms, this could be interpreted as the vanishing point for the back-projection ray if the ray were visible in camera 2's image. This direction points towards the line of sight from camera 1's point of view as seen by camera 2.

1.4 Part 1.4

we have $\begin{cases} \tilde{u}'(\lambda) \\ x'(\lambda) := K_2^{-1} \tilde{u}'(\lambda) \\ \lambda \to 0 \Rightarrow \tilde{x}'(0) \text{ epipole } e_2 \text{ in camera 2's frame} \\ \lambda \to \infty \Rightarrow \tilde{x}'(\infty) \text{ the direction vector in camera 2's frame corresponding to the line of sight from camera 1} \end{cases}$

$$\tilde{l} = \tilde{x}'(0) \times \tilde{x}'(\infty)$$

$$\left.\begin{aligned} \tilde{x}'(0) &= K_2^{-1} \tilde{u}'(0) \\ \tilde{x}'(\infty) &= K_2^{-1} \tilde{u}'(\infty) \end{aligned}\right\} \to \tilde{l} = (K_2^{-1} \tilde{u}'(0)) \times (K_2^{-1} \tilde{u}'(\infty))$$

$$\tilde{u}_2^T F \tilde{u}_1 = 0$$

any point $\tilde{u}_2$ on the epipolar line in the second image $\to \tilde{u}_2^T \tilde{l} = 0$

line with $\tilde{u}'(0)$ & $\tilde{u}'(\infty) = K_2^{-T} \tilde{l} \to \tilde{l} = F \tilde{u}_1$

2 Part 2: The 8-point algorithm

The points correspondence between two images and their associated epipolar lines.

In the image, each pair of corresponding points is marked with the same color across both images.

Each point in one image lie on the line corresponding to its matched point in the other image. So fundamental matrix is correctly estimated.

2.1 Part 2.1

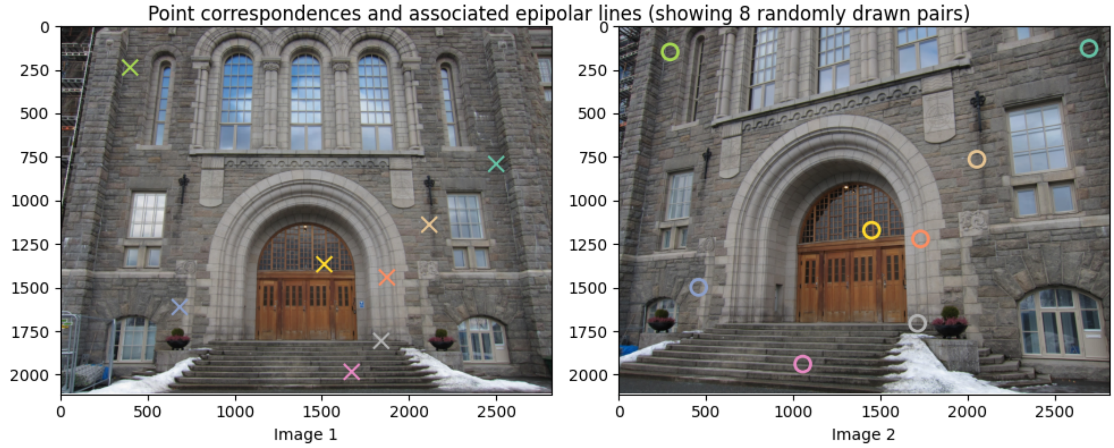

Figure 4: Figure for the part 2.1.

3 Part 3: Triangulation

3.1 Part 3.1

$\tilde{W}$ obtained from SVD for points in far away isn't always close to Zero.
because SVD doesnt considers the physical meaning of distance.
→ $\tilde{W}$ can be found out by algebraic solution that minimizes
the equations residual, not the point's actual distance from the camera.

Figure 5: Answers for the part 3.1.

3.2 Part 3.2

```
True vs. estimated 3D coordinates
---------------------------------
True: [-1. -2. -3.  1.]
Est.: [-1. -2. -3.  1.]
True: [ 2. -4. -2.  1.]
Est.: [ 2. -4. -2.  1.]
True: [-5. -4. -1.  1.]
Est.: [-5. -4. -1.  1.]
Triangulation is GOOD.
```

Figure 6: Test of the implementation.

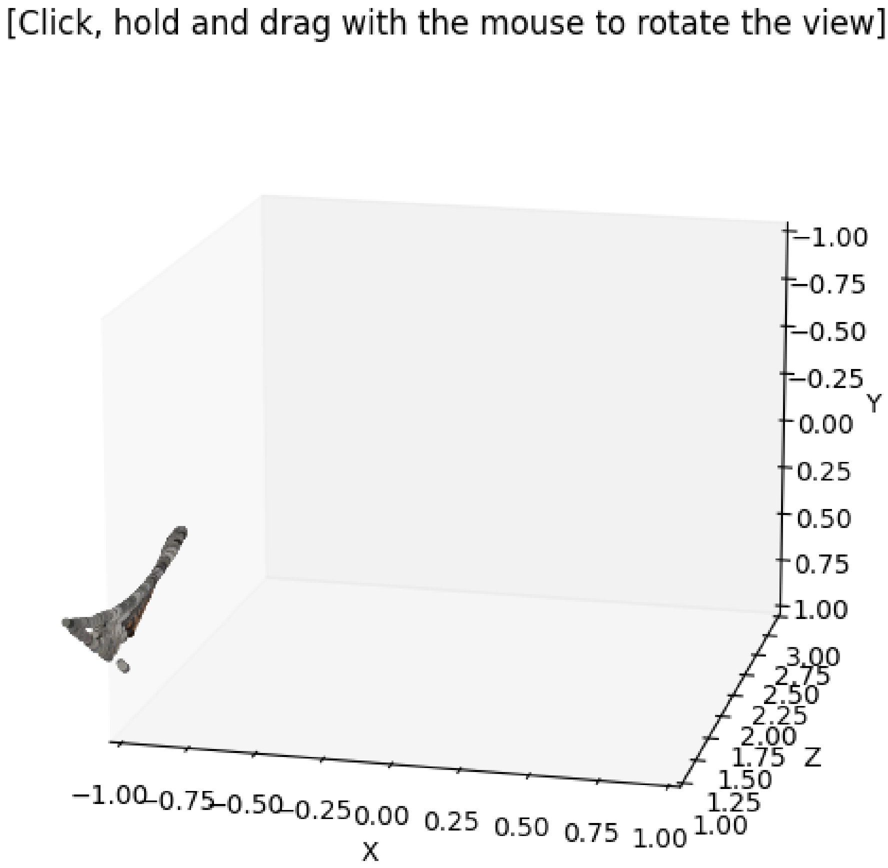

Figure 7: The plot of the 3D point cloud.