Computer Vision Exercise 3

Calibration & Structure from Motion

Tasks

1. Calibration with a known target
   – Data normalization
   – DLT
   – Optimization
   – Decomposition

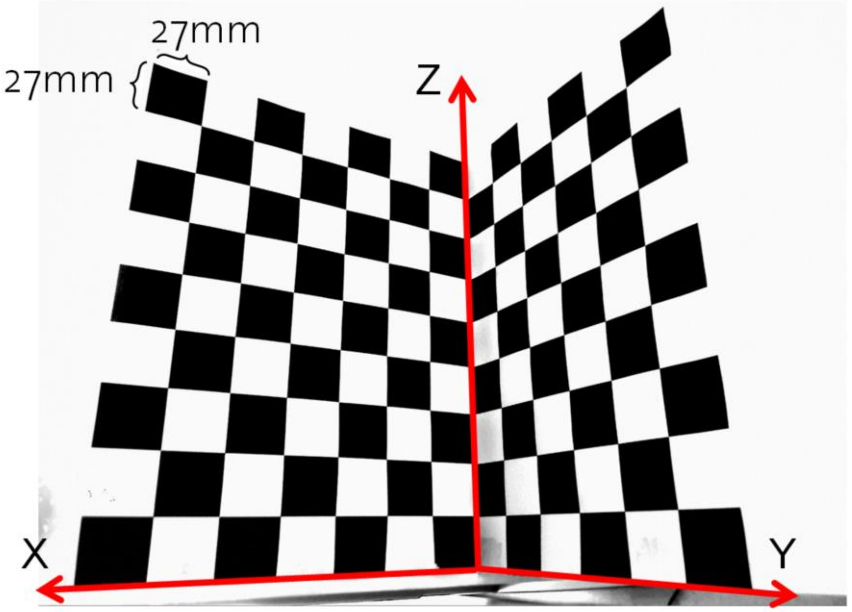
2. Scene reconstruction with SfM
   – DLT (Essential matrix)
   – Testing decompositions
   – Map extension

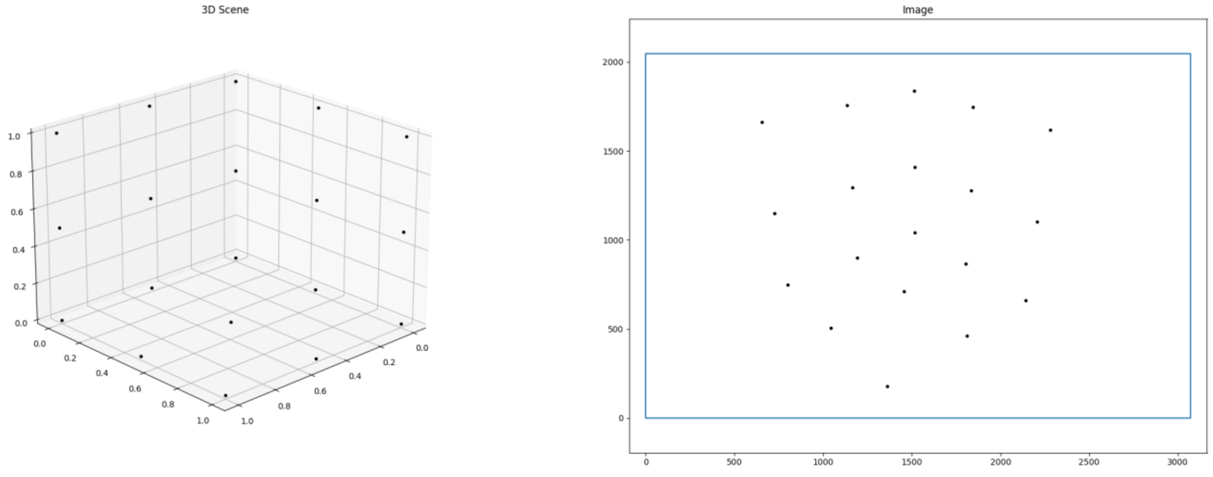
Setup

```
cd code
python3 -m venv venv
source venv/bin/activate
pip install --upgrade pip
pip install -r requirements.txt
```

or just install the dependencies manually

Calibration





We provide 2D-3D correspondence with the code

Calibration

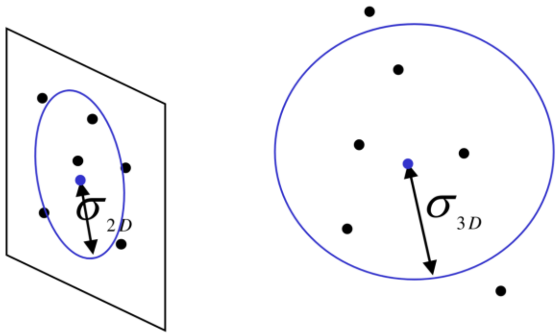
Data Normalization (& Denormalization)

$$\begin{bmatrix} 0.01 & & 46872.1 & \cdots & 2.74 \\ \vdots & \ddots & & & \vdots \\ & & & \cdots & 0.21 \\ & & \vdots & \ddots & \vdots \\ 0.62 & & 14793.2 & \cdots & 1.84 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Calibration

Data Normalization (& Denormalization)

1. **move center of mass to origin**
2. **scale to yield order 1 values**



$$\hat{x} = T\,x \qquad \hat{X} = U\,X$$

$$\hat{x} = \hat{P}\hat{X} \;\leftrightarrow\; x = PX$$

$$\mathrm{T}x = \hat{P}UX$$

$$x = T^{-1}\hat{P}UX$$

$$x = (T^{-1}\hat{P}U)X$$

$$P = T^{-1}\hat{P}U$$

Calibration

Direct Linear Transform (DLT)

$$x \cancel{=} PX \qquad x \propto PX \;\rightarrow x \;\times PX = \;[x]_\times PX = 0$$

$$\begin{bmatrix} w\,x_1 \\ w\,x_2 \\ w \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} \qquad \begin{bmatrix} W\,X_1 \\ W\,X_2 \\ W\,X_3 \\ W \end{bmatrix} \rightarrow \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{bmatrix}$$

$$P_{11}X_1 + P_{12}X_2 + P_{13}X_3 + P_{14} + \; P_{31}(\,-x_1X_1\,) \; + P_{32}(\,-x_1X_2\,) + P_{33}(\,-x_1X_3\,) + P_{34}(\,-x_1) = 0$$

Calibration

Direct Linear Transform (DLT)

$$P_{11}X_1 + P_{12}X_2 + P_{13}X_3 + P_{14} + \ P_{31}(\,-x_1X_1\,) \ + P_{32}(\,-x_1X_2\,) + P_{33}(\,-x_1X_3\,) + P_{34}(\,-x_1) = 0$$

$$\begin{bmatrix} \mathbf{0}^T & -\boldsymbol{X}^T & x_2\boldsymbol{X}^T \\ \boldsymbol{X}^T & \mathbf{0}^T & -x_1\boldsymbol{X}^T \end{bmatrix} \begin{bmatrix} P_{11} \\ \vdots \\ P_{34} \end{bmatrix} = \mathbf{0}$$

Calibration

Optimization

$$P^* = \min_P \sum_i \|\boldsymbol{x}_i - P\,\boldsymbol{X}_i\|^2$$

Normalize homogeneous coordinates!

Calibration

Decomposition

$$P = \mathrm{K}\,[R \mid t] = K\,[R \mid -RC] = [\,KR \mid -KRC\,]$$

$$M = KR$$

$$K^{-1}, R^{-1} = qr(M^{-1})$$

$$PC = \mathbf{0}$$

Calibration

Decomposition

K should have a positive diagonal!

$$KR = \widehat{K}TT^{-1}R \text{ , T} = \mathrm{diag}\left(\mathrm{sign}\left(\mathrm{diag}\left(\mathrm{K}\right)\right)\right)$$

R should have determinant 1!

$$R = -\hat{R} \text{ if } \det\left(\hat{R}\right) < 0$$

SfM

- Initialization (Relative pose)
- Point Triangulation
- Absolute Pose estimation

Not covered:

- Feature matching
- Robust estimation (Model fitting)
- Bundle adjustment

SfM

SfM

Initialization

$$\hat{x} = K^{-1}x$$

$$\hat{\boldsymbol{x}}_1^T E \hat{\boldsymbol{x}}_2 = 0$$

Same approach as for P (DLT)!

SfM

Initialization – Constraints on E

$$U, S, V^T = svd(\hat{E})$$

$$E = U \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0 \end{bmatrix} V^T$$

SfM

Initialization – Finding the right decomposition

Decomposing E gives 4 possible relative poses

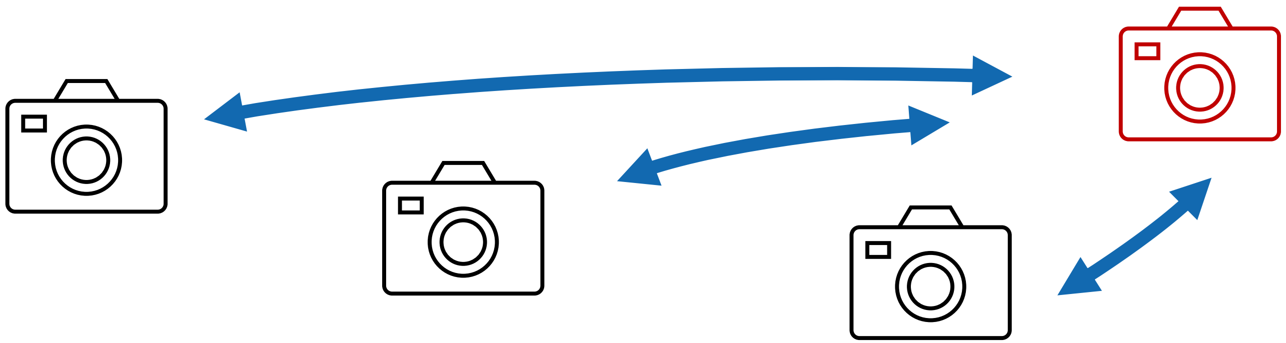
$$(R_1, t), (R_1, -t), (R_2, t), (R_2, -t)$$

Try each one to see where points end up in front of the cameras

SfM

Map extension

For each new image, call the point triangulation with every previous image

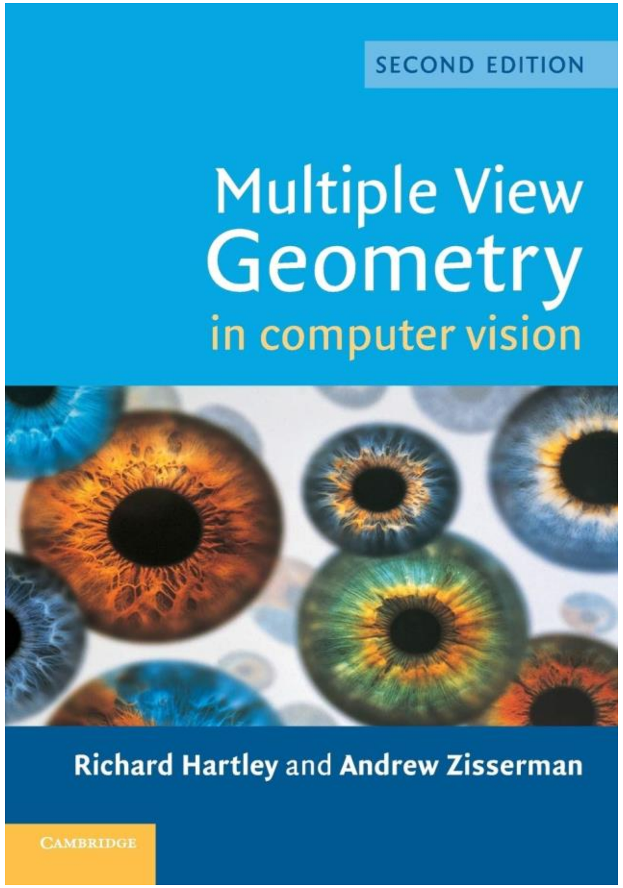


Hand-in

- Report (PDF)
- Code

Upload to moodle until Nov. 19, 23:59

Literature



Digital version available through ETH library