FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
FACULTY OF ENGINEERING

Camera Calibration & Stereo Vision

Computer Vision Project

Tobias Geimer
Pattern Recognition Lab, Friedrich-Alexander University of Erlangen-Nuremberg, Germany
November 20th, 2017

Outline

Recap: Camera Models

Homogeneous Coordinates
Extrinsic Parameters
Intrinsic Parameters

Camera Calibration

Zhang's Method
Distortion Correction

Stereo Vision

Epipolar Geometry
Triangulation

Exercises

Section 1

Recap: Camera Models

Homogeneous Coordinates (1)

2-D Case

$\boldsymbol{x}_k \in \mathbb{R}^2, \boldsymbol{x}_h \in \mathbb{P}^2$

$$\boldsymbol{x}_k = \begin{pmatrix} u \\ v \end{pmatrix} \rightarrow \boldsymbol{x}_h = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tilde{=} \begin{pmatrix} \lambda u \\ \lambda v \\ \lambda \end{pmatrix} \quad (1)$$

$$\boldsymbol{x}_h = \begin{pmatrix} u \\ v \\ w \end{pmatrix} \rightarrow \boldsymbol{x}_k = \begin{pmatrix} \frac{u}{w} \\ \frac{v}{w} \end{pmatrix}, w \in \mathbb{R} \setminus \{0\} \quad (2)$$

Remember

$\boldsymbol{x}_h = \begin{pmatrix} u \\ v \\ 0 \end{pmatrix}$ denote points at infinity that are not part of the euclidean space $\mathbb{R}^2$.

Homogeneous Coordinates (1)

2-D Case

$\boldsymbol{x}_k \in \mathbb{R}^2, \boldsymbol{x}_h \in \mathbb{P}^2$

$$\boldsymbol{x}_k = \begin{pmatrix} u \\ v \end{pmatrix} \rightarrow \boldsymbol{x}_h = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \mathrel{\hat{\tilde{=}}} \begin{pmatrix} \lambda u \\ \lambda v \\ \lambda \end{pmatrix} \tag{1}$$

$$\boldsymbol{x}_h = \begin{pmatrix} u \\ v \\ w \end{pmatrix} \rightarrow \boldsymbol{x}_k = \begin{pmatrix} \frac{u}{w} \\ \frac{v}{w} \end{pmatrix}, w \in \mathbb{R} \setminus \{0\} \tag{2}$$

Remember

$\boldsymbol{x}_h = \begin{pmatrix} u \\ v \\ 0 \end{pmatrix}$ denote points at infinity that are not part of the euclidean space $\mathbb{R}^2$.

Homogeneous Coordinates (2)

3-D Case

$\boldsymbol{x}_k \in \mathbb{R}^3, \boldsymbol{x}_h \in \mathbb{P}^3$

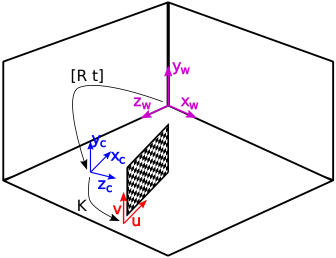
$$\boldsymbol{x}_k = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad \rightarrow \quad \boldsymbol{x}_h = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \hat{\cong} \begin{pmatrix} \lambda x \\ \lambda y \\ \lambda z \\ \lambda \end{pmatrix} \tag{3}$$

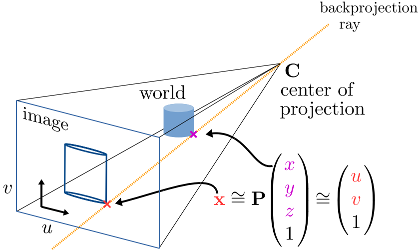
$$\boldsymbol{x}_h = \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} \quad \rightarrow \quad \boldsymbol{x}_k = \begin{pmatrix} \frac{x}{w} \\ \frac{y}{w} \\ \frac{z}{w} \end{pmatrix}, w \in \mathbb{R} \setminus \{0\} \tag{4}$$

Projection Matrix

$$\boldsymbol{P} = \boldsymbol{K}\left[\boldsymbol{R} \mid \boldsymbol{t}\right]$$

- Intrinsic parameters: $\boldsymbol{K}$
- Extrinsic parameters: $\left[\boldsymbol{R} \mid \boldsymbol{t}\right]$
    - $\boldsymbol{R}$: rotation
    - $\boldsymbol{t}$: translation

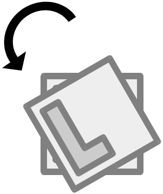




Extrinsic Camera Parameters

Rotation

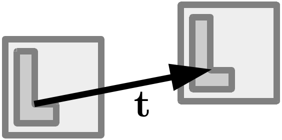
$$\boldsymbol{R} = [\boldsymbol{r}_1 \ \boldsymbol{r}_2 \ \boldsymbol{r}_3]$$



Translation

- Euclidean: $\mathbb{R}^2$

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} t_u \\ t_v \end{pmatrix}$$



- Projective: $\mathbb{P}^2$

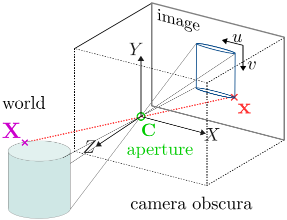
$$\boldsymbol{T} = \begin{pmatrix} 1 & 0 & t_u \\ 0 & 1 & t_v \\ 0 & 0 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \begin{pmatrix} u + t_u \\ v + t_v \\ 1 \end{pmatrix} = \boldsymbol{T} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

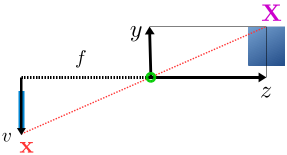
Intrinsic Camera Parameters
Pinhole Model

$$\boldsymbol{K} = \begin{pmatrix} f_u & s & u_0 \\ & f_v & v_0 \\ & & 1 \end{pmatrix}$$

- $\begin{pmatrix} u_0 \\ v_0 \end{pmatrix}$ principal point
- f_u, f_v focal length in pixels
- s skew

$\rightarrow$ Invariant under external motion of the camera.





As seen from the world X-axis

Section 2

Camera Calibration

Camera Calibration
How many parameters to estimate?

Extrinsic Parameters

Position & orientation of the camera in reference to the world.

- Translation: 3 DoF
- Rotation: 3 DoF

$\rightarrow$ 6 parameters

Intrinsic Parameters

Parameters that determine how a scene is represented on the image plane.

- Pinhole model
- f_u, f_v, s, u_0, v_0

$\rightarrow$ 5 parameters

$\rightarrow$ 3×4 projection matrix is determined up to scale.

Zhang's Method[1]

Calibration Phantom

Known Checkerboard Pattern ($\boldsymbol{X}$)

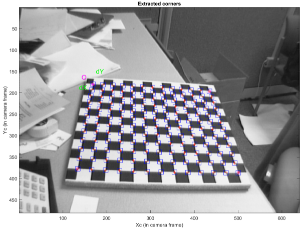
- number of squares N_x, N_y
- size of squares d_x, d_y

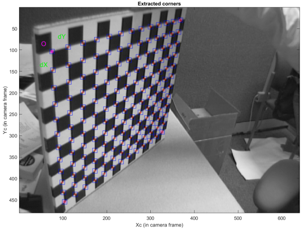
$$\boldsymbol{X}_{ij} = (i \cdot d_x, j \cdot d_y, z)^T, \qquad (5)$$

with $i = 0, \cdots, N_x$
$j = 0, \cdots, N_y$.

Feature Extraction ($\boldsymbol{x}$)

- Images at $n \geq 3$ diff. orientations
- Define *Region of Interest* (ROI)
- Harris Corner Detector etc.





[1] Z. Zhang, "A flexible new technique for camera calibration", IEEE Trans. Pattern Anal. Machine Intell. **22**, 1330–1334 (2000).

Zhang's Method

Homography: Model Plane to Image

W.l.o.g. model plane $z = 0$:

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \boldsymbol{K} [\underbrace{\boldsymbol{r}_1 \; \boldsymbol{r}_2 \; \boldsymbol{r}_3}_{\boldsymbol{R}} \; \boldsymbol{t}] \begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix} = \boldsymbol{K} [\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boldsymbol{t}] \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

represents a Homography

$$\lambda \boldsymbol{x} = \boldsymbol{H} \boldsymbol{X}$$

between image point $\boldsymbol{x}$ and his 3-D space point $\boldsymbol{X}$.
$\rightarrow$ *DLT* in $\mathbb{R}^3$ ($2n \times 9$ matrix) from 2nd lecture.

Zhang's Method
Homography Decomposition

$$\boldsymbol{H} = [\boldsymbol{h}_1 \quad \boldsymbol{h}_2 \quad \boldsymbol{h}_3] = \lambda \boldsymbol{K} [\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boldsymbol{t}] \tag{6}$$

Idea

Entries of the homography are actually a combination of intrinsic and extrinsic parameters.

$\rightarrow$ Extrinsic parameters are calculated directly from (6).

$\rightarrow$ Intrinsic parameters can be recovered by introducing additional constraints.

Zhang's Method
Constraints on Intrinsic Parameters

With $\boldsymbol{r}_1, \boldsymbol{r}_2$ being orthonormal, there are two constraints:

$$\boldsymbol{r}_1^T \boldsymbol{r}_2 = 0 \qquad \& \qquad \boldsymbol{r}_1^T \boldsymbol{r}_1 = \boldsymbol{r}_2^T \boldsymbol{r}_2$$

$$\Downarrow$$

$$\boldsymbol{h}_1^T \boldsymbol{K}^{-T} \boldsymbol{K}^{-1} \boldsymbol{h}_2 = 0 \tag{7}$$

$$\boldsymbol{h}_1^T \boldsymbol{K}^{-T} \boldsymbol{K}^{-1} \boldsymbol{h}_1 = \boldsymbol{h}_2^T \boldsymbol{K}^{-T} \boldsymbol{K}^{-1} \boldsymbol{h}_2 \tag{8}$$

$\boldsymbol{K}^{-T} \boldsymbol{K}^{-1}$ is called the absolute conic.

Zhang's Method

Absolute Conic

$$\boldsymbol{B} = \boldsymbol{K}^{-T}\boldsymbol{K}^{-1} = \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{{f_u}^2} & -\frac{s}{{f_u}^2 f_v} & \frac{v_0 s - u_0 f_v}{{f_u}^2 f_v} \\ -\frac{s}{{f_u}^2 f_v} & \frac{s^2}{{f_u}^2 {f_v}^2} + \frac{1}{{f_v}^2} & -\frac{s(v_0 s - u_0 f_v)}{{f_u}^2 {f_v}^2} - \frac{v_0}{{f_v}^2} \\ \frac{v_0 s - u_0 f_v}{{f_u}^2 f_v} & -\frac{s(v_0 s - u_0 f_v)}{{f_u}^2 {f_v}^2} - \frac{v_0}{{f_v}^2} & \frac{(v_0 s - u_0 f_v)^2}{{f_u}^2 {f_v}^2} + \frac{{v_0}^2}{{f_v}^2} + 1 \end{pmatrix} \quad (9)$$

Note, that $\boldsymbol{B}$ is symmetric. Denoted by 6-D vector:

$$\boldsymbol{b} = (B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33})^T \quad (10)$$

Zhang's Method
Closed-Form Solution (1)

Let $\boldsymbol{h}_i = \left(h_{i1}, h_{i2}, h_{i3}\right)^T$ be the i-th <u>column</u> vector of $\boldsymbol{H}$.

$$\boldsymbol{h}_i^T \boldsymbol{B} \boldsymbol{h}_j = \boldsymbol{g}_{ij}^T \boldsymbol{b} \qquad \text{where } \boldsymbol{g}_{ij} = \begin{pmatrix} h_{i1} h_{j1} \\ h_{i1} h_{j2} + h_{i2} h_{j1} \\ h_{i2} h_{j2} \\ h_{i3} h_{j1} + h_{i1} h_{j3} \\ h_{i3} h_{j2} + h_{i2} h_{j3} \\ h_{i3} h_{j3} \end{pmatrix}$$

Eqn. (7) & (8) can then be rewritten to homogeneous equations in $\boldsymbol{b}$:

$$\begin{bmatrix} \boldsymbol{g}_{12}^T \\ \left(\boldsymbol{g}_{11} - \boldsymbol{g}_{22}\right)^T \end{bmatrix} \boldsymbol{b} = \boldsymbol{0} \tag{11}$$

Zhang's Method
Closed-Form Solution (2)

With n observed images of the model plane, n equations as (11) can be stacked:

$$\boldsymbol{G}\boldsymbol{b} = \mathbf{0} \tag{12}$$

where $\boldsymbol{G}$ is a $2n \times 6$ matrix.

With $n \geq 3$ a unique solution $\boldsymbol{b}$ can be found, defined up to scale.

Solve (12) via *Singular Value Decomposition* (SVD) for the right singular vector of $\boldsymbol{G}$ associated with smallest singular value.

Zhang's Method

Recovering Intrinsic Parameters

Using (9) we can extract the intrinsic parameters from $\boldsymbol{B}$:

$$v_0 = \frac{B_{12}B_{13} - B_{11}B_{23}}{B_{11}B_{22} - B_{12}^2} \qquad \lambda = B_{33} - \frac{B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})}{B_{11}}$$

$$f_u = \sqrt{\frac{\lambda}{B_{11}}} \qquad f_v = \sqrt{\frac{\lambda B_{11}}{B_{11}B_{22} - B_{12}^2}}$$

$$s = -\frac{B_{12}{f_u}^2 f_v}{\lambda} \qquad u_0 = \frac{s v_0}{f_v} - B_{13}\frac{{f_u}^2}{\lambda}$$

Zhang's Method

Recovering Extrinsic Parameters

From (6) we get:

$$\boldsymbol{r}_1 = \mu\, \boldsymbol{K}^{-1} \boldsymbol{h}_1 \tag{13}$$

$$\boldsymbol{r}_2 = \mu\, \boldsymbol{K}^{-1} \boldsymbol{h}_2 \tag{14}$$

$$\boldsymbol{r}_3 = \boldsymbol{r}_1 \times \boldsymbol{r}_2 \tag{15}$$

$$\boldsymbol{t} = \mu\, \boldsymbol{K}^{-1} \boldsymbol{h}_3 \tag{16}$$

with

$$\mu = \frac{1}{||\boldsymbol{K}^{-1} \boldsymbol{h}_1||} = \frac{1}{||\boldsymbol{K}^{-1} \boldsymbol{h}_2||}$$

Due to noise, $\boldsymbol{R} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3]$ may not necessarily satisfy the properties of a rotation matrix (orthonormality). Enforce it using SVD:

$$\boldsymbol{R}' = \boldsymbol{U}\boldsymbol{V}^T, \tag{17}$$

with $\boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$.

Zhang's Method

Refinement: Maximum-Likelihood Estimation

Current solution obtained by minimizing an algebraic distance without any physical meaning.

Refinement by optimizing the functional:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} ||\boldsymbol{x}_{ij} - \hat{\boldsymbol{x}}(\boldsymbol{K}, \boldsymbol{R}_i, \boldsymbol{t}_i, \boldsymbol{X}_j)||^2, \tag{18}$$

where $\hat{\boldsymbol{x}}(\boldsymbol{K}, \boldsymbol{R}_i, \boldsymbol{t}_i, \boldsymbol{X}_j)$ is the projection of 3-D point $\boldsymbol{X}_j$ in image i.

Requires an initialization of $\boldsymbol{K}, \{\boldsymbol{R}_i, \boldsymbol{t}_i | i = 1, \ldots, n\}$ obtained by the previous closed-form technique.

Lens Distortion
Analytical Modeling[2]

Radial Distortion

$$\begin{bmatrix} \delta u_i^{(r)} \\ \delta v_i^{(r)} \end{bmatrix} = \begin{bmatrix} \tilde{u}_i(k_1 r_i^2 + k_2 r_i^4 + \dots) \\ \tilde{v}_i(k_1 r_i^2 + k_2 r_i^4 + \dots) \end{bmatrix},$$

where k_1, k_2 are radial distortion coefficients and $r_i = \sqrt{\tilde{u}_i^2 + \tilde{v}_i^2}$.

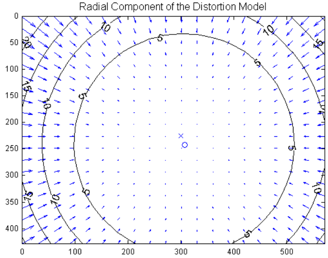
Tangential Distortion

$$\begin{bmatrix} \delta u_i^{(t)} \\ \delta v_i^{(t)} \end{bmatrix} = \begin{bmatrix} 2p_1 \tilde{u}_i \tilde{v}_i + p_2(r_i^2 + 2\tilde{u}_i^2) \\ p_1(r_i^2 + 2\tilde{v}_i^2) + 2p_2 \tilde{u}_i \tilde{v}_i \end{bmatrix},$$

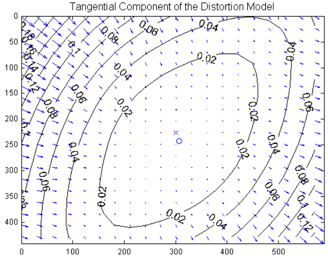
where p_1, p_2, p_3 are tangential distortion coefficients.

[2] J. Heikkila and O. Silven, “A four-step camera calibration procedure with implicit image correction”, in Computer vision and pattern recognition, 1997. proceedings., 1997 ieee computer society conference on (1997), pp. 1106–1112.

Lens Distortion
Example



Pixel error	= [0.4533, 0.3892]	
Focal Length	= (661.67, 662.829)	+/- [1.179, 1.266]
Principal Point	= (306.096, 240.79)	+/- [2.384, 2.175]
Skew	= 0	+/- 0
Radial coefficients	= (-0.2642, 0.2265, 0)	+/- [0.00934, 0.03826, 0]
Tangential coefficients	= (0.0002, 0.00023)	+/- [0.00052, 0.00053]



Pixel error	= [0.4533, 0.3892]	
Focal Length	= (661.67, 662.829)	+/- [1.179, 1.266]
Principal Point	= (306.096, 240.79)	+/- [2.384, 2.175]
Skew	= 0	+/- 0
Radial coefficients	= (-0.2642, 0.2265, 0)	+/- [0.00934, 0.03826, 0]
Tangential coefficients	= (0.0002, 0.00023)	+/- [0.00052, 0.00053]

Zhang's Method
Summary

1. Planar checkerboard pattern.
2. Images taken from different views by either moving the plane or the camera.
3. Detect feature points.
   $\rightarrow$ Remember to normalize your data points!
4. Initialization:
   Estimate 5 intrinsic and all extrinsic parameters via closed-form solution.
5. Optimization:
   Refine all parameters (affected by noise) including lens distortion.

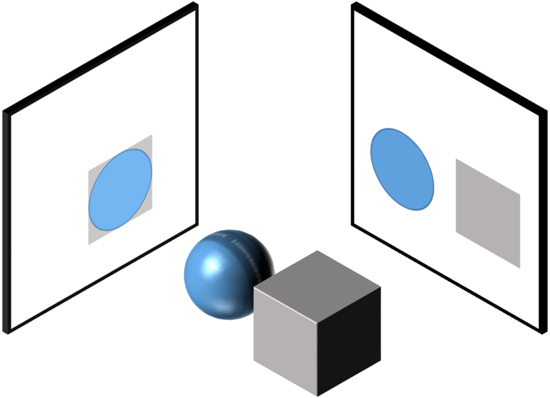
Section 3

Stereo Vision

Stereo Vision
Geometry of a Scene from Two Different Views

- Simultaneous (stereo rig)
- Continuous (moving camera)

$\Rightarrow$ Geometrically equivalent!



Stereo Vision
Questions to be answered

1. Correspondence geometry

   Given $\boldsymbol{x}$ in view A. How does this constrain the position of the corresponding $\boldsymbol{x}'$ in view B?

2. Camera geometry (motion)

   Given corresponding image points $\{\boldsymbol{x}_i \leftrightarrow \boldsymbol{x}'_i\}$, $i = 1, \ldots, n$, what are the cameras $\boldsymbol{P}$ and $\boldsymbol{P}'$ for their respective views?

3. Scene geometry (structure)

   Given corresponding image points $\{\boldsymbol{x} \leftrightarrow \boldsymbol{x}'\}$ and cameras $\boldsymbol{P}$, $\boldsymbol{P}'$, what is the position of $\boldsymbol{X}$ in 3-D?

Epipolar Geometry
Visualization

- **Epipole**
  Intersecting point of the line joining the camera centres (baseline) with the image plane.
- **Epipolar Plane**
  Plane containing the baseline.
- **Epipolar Line**
  Intersection of an epipolar plane with the image plane. All epipolar lines intersect at the epipole.

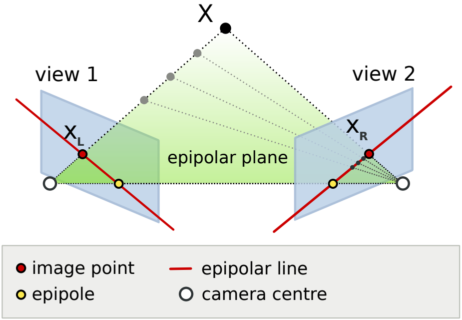


Image adapted from: https://commons.wikimedia.org

Fundamental Matrix
Algebraic Representation of the Epipolar Geometry

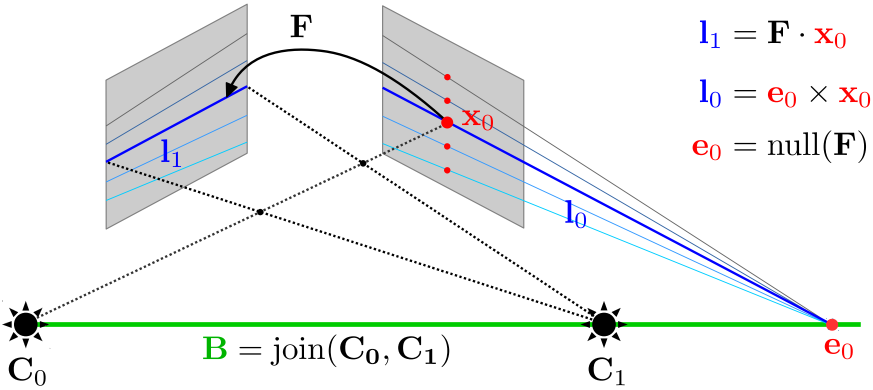


Image courtesy of *A. Aichert*.

Fundamental Matrix
Properties

- $\boldsymbol{F}$ is a rank 2 homogeneous matrix with 7 degrees of freedom.
- **Point correspondence**
  - $\boldsymbol{x}'^T \boldsymbol{F} \boldsymbol{x} = 0, \ \forall \boldsymbol{x} \leftrightarrow \boldsymbol{x}'$
- **Epipolar lines**
  - $\boldsymbol{l}' = \boldsymbol{F}\boldsymbol{x}$ is the epipolar line corresponding to $\boldsymbol{x}$.
  - $\boldsymbol{l} = \boldsymbol{F}^T \boldsymbol{x}'$ is the epipolar line corresponding to $\boldsymbol{x}'$.
- **Epipoles**
  - $\boldsymbol{e}'^T \boldsymbol{F} = \boldsymbol{0}$ (left null-vector)
  - $\boldsymbol{F}\boldsymbol{e} = \boldsymbol{0}$ (right null-vector)

Fundamental Matrix
Computation from Camera Matrices

Back-projection from $\boldsymbol{x}$ by solving $\boldsymbol{PX} = \boldsymbol{x}$:

$$\boldsymbol{X}(a) = \boldsymbol{P}^{\dagger}\boldsymbol{x} + a\boldsymbol{C},$$

$\boldsymbol{P}^{\dagger}$: pseudo-inverse of $\boldsymbol{P}$
$\boldsymbol{C}$: camera center with $\boldsymbol{PC} = 0$

Two particular points on the ray:

- at $a = 0$: $\quad \boldsymbol{X}_0 = \boldsymbol{P}^{\dagger}\boldsymbol{x}$
- at $a = \infty$: $\quad \boldsymbol{X}_\infty = \boldsymbol{C}$

Imaged by $\boldsymbol{P}'$ at $\boldsymbol{P}'\boldsymbol{P}^{\dagger}\boldsymbol{x}$ and $\boldsymbol{P}'\boldsymbol{C}$ in the second view.

Epipolar line joining these two points:

$$\boldsymbol{l}' = \underbrace{(\boldsymbol{P}'\boldsymbol{C})\times}_{[\boldsymbol{e}']_x}(\boldsymbol{P}'\boldsymbol{P}^{\dagger}\boldsymbol{x})$$

Thus

$$\boldsymbol{l}' = [\boldsymbol{e}']_x(\boldsymbol{P}'\boldsymbol{P}^{\dagger})\boldsymbol{x} = \boldsymbol{F}\boldsymbol{x}, \;\rightarrow\; \boldsymbol{F} = [\boldsymbol{e}']_x\boldsymbol{P}'\boldsymbol{P}^{\dagger} \tag{19}$$

Fundamental Matrix
Retrieving Camera Matrices

- **Projective invariance**

$$\left.\begin{array}{rcl} \boldsymbol{l}' & = & \boldsymbol{F}\boldsymbol{x} \\ \boldsymbol{x}'^T\boldsymbol{F}\boldsymbol{x} & = & 0 \end{array}\right\} \text{projective relationships (i.e. 1 dof)}$$

- **Non-unique solution**[3]

  If $\boldsymbol{T}$ is a 4×4 matrix representing a projective transformation of 3-space, then the fundamental matrices corresponding to the pairs of camera matrices $(\boldsymbol{P}, \boldsymbol{P}')$ and $(\boldsymbol{P}\boldsymbol{T}, \boldsymbol{P}'\boldsymbol{T})$ are the same.

  - Camera matrices $(\boldsymbol{P}, \boldsymbol{P}') \rightarrow$ unique $\boldsymbol{F}$.
  - $\boldsymbol{F} \rightarrow (\boldsymbol{P}, \boldsymbol{P}')$ up to 3-D projective transform:

$$\boldsymbol{P} = [\boldsymbol{I}_3 \mid \boldsymbol{0}] \;\text{ and }\; \boldsymbol{P}' = [[\boldsymbol{e}']_x \boldsymbol{F} \mid \boldsymbol{e}'] \tag{20}$$

[3] R. Hartley and A. Zisserman, Multiple view geometry in computer vision, (Cambridge university press, 2003)

The Eight Point Algorithm

Computation of the Fundamental Matrix

Objective

Given $n \geq 8$ image point correspondences $\{\boldsymbol{x}_i \leftrightarrow \boldsymbol{x}'_i\}$, determine the fundamental matrix $\boldsymbol{F}$ such that $\boldsymbol{x}'^T_i \boldsymbol{F} \boldsymbol{x}_i = 0$.

Algorithm Breakdown

- System is linear in the components of $\boldsymbol{F}$ with $\boldsymbol{M}\boldsymbol{f} = \boldsymbol{0}$

$$\boldsymbol{M} = \begin{bmatrix} x_{11}x'_{11} & x_{11}x'_{12} & x_{11}x'_{13} & x_{12}x'_{11} & \dots & x_{13}x'_{13} \\ x_{21}x'_{21} & x_{21}x'_{22} & x_{21}x'_{23} & x_{22}x'_{21} & \dots & x_{23}x'_{23} \\ \vdots & & \ddots & & & \vdots \\ x_{n1}x'_{n1} & x_{n1}x'_{n2} & x_{n1}x'_{n3} & x_{n2}x'_{n1} & \dots & x_{n3}x'_{n3} \end{bmatrix} \in \mathbb{R}^{n \times 9} \quad (21)$$

$$\boldsymbol{f} = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})^T \quad (22)$$

- Solve using SVD.
- Assure $\text{rank}(\boldsymbol{F}) = 2$.

The Eight Point Algorithm

Details

Over-determined system of equations $\boldsymbol{Mf} = \boldsymbol{0}$

- $\boldsymbol{f}$ is in the null-space of $\boldsymbol{M}$.
- $\boldsymbol{M} \in \mathbb{R}^{n \times 9}$ & $\text{rank}(\boldsymbol{M}) = 8$: non-trivial solution next to the trivial $\boldsymbol{f} = \boldsymbol{0}$

1. SVD: $\boldsymbol{M} = \boldsymbol{U\Sigma V}^T$
   - $\sigma_9 \approx 0 \rightarrow \boldsymbol{f} = \lambda \boldsymbol{v}_9$
   - since $||\boldsymbol{F}||_F = ||\boldsymbol{f}||_2 = 1 \rightarrow \boldsymbol{f} = \boldsymbol{v}_9$
   - if $\sigma_9 > \varepsilon$: Error
2. Force $\text{Rank}(\boldsymbol{F}) = 2$ using SVD: $F = \boldsymbol{U}_F \boldsymbol{\Sigma}_F \boldsymbol{V}_F$
   - For the fundamental matrix it is: $\sigma_1 > \sigma_2 > 0, \sigma_3 = 0$
   - Set $\sigma_3 = 0$
   - Compute $\boldsymbol{F}^\star$ anew using $\boldsymbol{\Sigma}_F^\star$:

$$\boldsymbol{F}^\star = \boldsymbol{U}_F \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \boldsymbol{V}_F \tag{23}$$

Essential Matrix
A *calibrated* case of the Fundamental Matrix

Given $\boldsymbol{P} = \boldsymbol{K}\left[\boldsymbol{R} \mid \boldsymbol{t}\right]$ and $\boldsymbol{x} = \boldsymbol{P}\boldsymbol{X}$.

If $\boldsymbol{K}$ is known:

- *Normalized coordinates*: $\tilde{\boldsymbol{x}} = \boldsymbol{K}^{-1}\boldsymbol{x} = \left[\boldsymbol{R} \mid \boldsymbol{t}\right]\boldsymbol{X}$
- *Normalized camera matrix*: $\tilde{\boldsymbol{P}} = \boldsymbol{K}^{-1}\boldsymbol{P} = \left[\boldsymbol{R} \mid \boldsymbol{t}\right]$

where the intrinsic effects are removed.

The fundamental matrix to a pair of normalized cameras $\tilde{\boldsymbol{P}}' = \left[\boldsymbol{I}_3 \mid \boldsymbol{0}\right]$ and $\tilde{\boldsymbol{P}} = \left[\boldsymbol{R} \mid \boldsymbol{t}\right]$ is called *essential matrix* $\boldsymbol{E}$.

Essential Matrix
Properties

- **Definition**
  - $\boldsymbol{E} = \boldsymbol{R}\left[\boldsymbol{t}\right]_x$
  - $\tilde{\boldsymbol{x}}'^T \boldsymbol{E} \tilde{\boldsymbol{x}} = 0$
  - $\boldsymbol{E} = \boldsymbol{K}'^T \boldsymbol{F} \boldsymbol{K}$
- **5 degrees of freedom**
  - rotation (+3) & translation (+3), up to scale (-1)
- **Constraints**
  - Singular values: $\sigma_1 = \sigma_2$, $\sigma_3 = 0$.
  - SVD: $\boldsymbol{E} = \boldsymbol{U}\lambda \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \boldsymbol{V}^T, \;\; \lambda \in \mathbb{R}$

Essential Matrix
Retrieving Extrinsic Camera Parameters (1)

E holds the relative orientation between 2 normalized cameras

$$\tilde{\boldsymbol{P}}' = [\boldsymbol{I}_3 \mid \boldsymbol{0}] \text{ and } \tilde{\boldsymbol{P}} = [\boldsymbol{R} \mid \boldsymbol{t}]$$

up to scale and a 4-fold ambiguity (baseline reversal & 180° rotation).

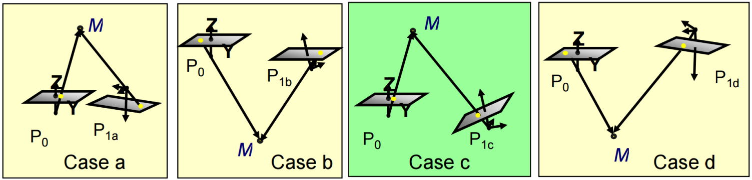


Image by: https: http://www.cs.unc.edu/ jmf/Tutorials/DAGM/DAGM-Tutorial-Frahm-Koch-Evers.pdf

Essential Matrix
Retrieving Extrinsic Camera Parameters (2)

The essential matrix

$$\boldsymbol{E} = \boldsymbol{U}\, \mathrm{diag}\big(1, 1, 0\big)\, \boldsymbol{V}^T,$$

provides four possible choices[3] for the second camera matrix $\boldsymbol{P}'$:

$$\tilde{\boldsymbol{P}}' = \begin{cases} \big[\boldsymbol{U}\boldsymbol{W}\boldsymbol{V}^T \mid \pm \boldsymbol{u}_3\big] \\ \big[\boldsymbol{U}\boldsymbol{W}^T\boldsymbol{V}^T \mid \pm \boldsymbol{u}_3\big] \end{cases}, \text{ where } \boldsymbol{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (24)$$

Correct candidate determined by calculation of depth values using Triangulation.

[3] R. Hartley and A. Zisserman, Multiple view geometry in computer vision, (Cambridge university press, 2003)

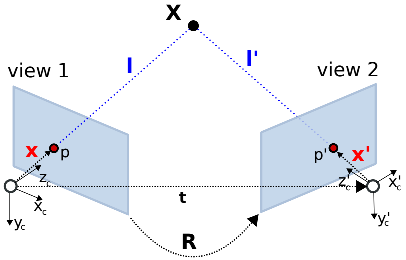
Triangulation
From images back to 3-space

Find point X in 3-space given by two corresponding image points.

- Lines through each camera center.
- Triangulated point can be expressed by both camera matrices.

$$\boldsymbol{x} = \boldsymbol{P}\boldsymbol{X} \tag{25}$$

$$\boldsymbol{x}' = \boldsymbol{P}'\boldsymbol{X} \tag{26}$$



$\rightarrow$ Combined to $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{0}$, equations linear in $\boldsymbol{X}$.

Linear Triangulation
Algorithm

Rewriting (25) to $\boldsymbol{x} \times (\boldsymbol{PX}) = \boldsymbol{0}$ gives:

$$x \cdot (\boldsymbol{p}_3^T \boldsymbol{X}) - 1 \cdot (\boldsymbol{p}_1^T \boldsymbol{X}) = 0$$
$$y \cdot (\boldsymbol{p}_3^T \boldsymbol{X}) - 1 \cdot (\boldsymbol{p}_2^T \boldsymbol{X}) = 0$$
$$x \cdot (\boldsymbol{p}_2^T \boldsymbol{X}) - y \cdot (\boldsymbol{p}_1^T \boldsymbol{X}) = 0$$

where $\boldsymbol{p}_i^T$ are the rows of $\boldsymbol{P}$ and $\boldsymbol{x} = (x, y, 1)^T$.
Include two equations from each image to build $\boldsymbol{AX} = \boldsymbol{0}$ with

$$\boldsymbol{A} = \begin{bmatrix} x\boldsymbol{p}_3^T - \boldsymbol{p}_1^T \\ y\boldsymbol{p}_3^T - \boldsymbol{p}_2^T \\ x'\boldsymbol{p}_3'^T - \boldsymbol{p}_1'^T \\ y'\boldsymbol{p}_3'^T - \boldsymbol{p}_2'^T \end{bmatrix}. \tag{27}$$

$\rightarrow$ Solve using SVD (again).

Section 4

Exercises

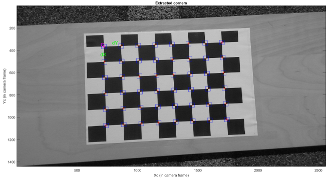
Zhang's Method

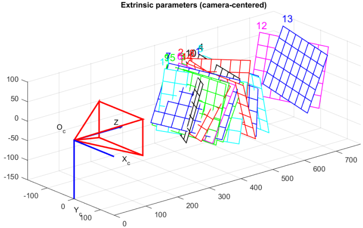
Compute an initialization for

- $\boldsymbol{K}$ (once)
- $[\boldsymbol{R}\ \boldsymbol{t}]_i$ (for each image i)

from a set of given feature points.

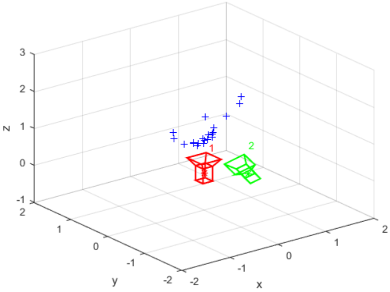
No non-linear optimization.
However, you may try.





Stereo Vision

- Fundamental Matrix: Computation & Visualization.
- Determine the relative orientation between both cameras.
- ***K*** is given.



A Note on Normalization

... in the context of DLT/SVD

! Data normalization is not optional.

- Compute centroid or both measured and reference points.
- Move center of mass to origin.
- Scale to range $\sqrt{2}$ and $\sqrt{3}$ for 2-D and 3-D respectively.

$$\boldsymbol{N}_2 = \begin{bmatrix} s^{-1} & 0 & -\bar{x}_0 s^{-1} \\ 0 & s^{-1} & -\bar{x}_1 s^{-1} \\ 0 & 0 & 1 \end{bmatrix}$$

with $\bar{\boldsymbol{x}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i$

and $s = \frac{1}{\sqrt{2}n} \sum_{i=1}^{n} ||\boldsymbol{x}_i - \bar{\boldsymbol{x}}||_2$.

A Note on Normalization
... in the context of DLT/SVD

- Normalize data points

$$\boldsymbol{x}_n = \boldsymbol{N}\boldsymbol{x}$$
$$\boldsymbol{x}'_n = \boldsymbol{N}'\boldsymbol{x}'$$

- Calculate normalized Homography $\boldsymbol{H}_n$ using DLT/SVD.

$$\boldsymbol{x}'_n = \boldsymbol{H}_n\boldsymbol{x}_n$$
$$\boldsymbol{N}'\boldsymbol{x}' = \boldsymbol{H}_n\boldsymbol{N}\boldsymbol{x}$$
$$\boldsymbol{x}' = \underbrace{\boldsymbol{N}'^{-1}\boldsymbol{H}_n\boldsymbol{N}}_{\boldsymbol{H}}\boldsymbol{x}$$

- Denormalized homography: $\boldsymbol{H} = \boldsymbol{N}'^{-1}\boldsymbol{H}_n\boldsymbol{N}$.

Thank you.

Any Questions?

www5.cs.fau.de/~geimer