Photogrammetry & Robotics Lab

Direct Solutions for Computing Fundamental and Essential Matrix

Cyrill Stachniss

The slides have been created by Cyrill Stachniss.

5 Minute Preparation for Today

https://www.ipb.uni-bonn.de/5min/

Motivation

$\mathrm{F}/\mathrm{E} \quad R, \mathbf{b}$ Image courtesy: Collins

Topics of Today

Compute the

- **Fundamental matrix**
  given corresponding points
- **Essential matrix**
  given corresponding points
- **Rotation matrix and basis**
  given an essential matrix

Computing the Fundamental Matrix Given Corresponding Points

Fundamental Matrix

- The **fundamental matrix** F **is**

$$\boxed{\mathsf{F} \quad = \quad (\mathsf{K}')^{-\mathsf{T}} R' \mathsf{S}_b {R''}^{\mathsf{T}} (\mathsf{K}'')^{-1}}$$

- It encodes the relative orientation for two uncalibrated cameras
- **Coplanarity constraint** through F

$$\boxed{{\mathbf{x}'}^{\mathsf{T}} \mathsf{F} \mathbf{x}'' \quad = \quad 0}$$

Fundamental Matrix

The fundamental matrix F can directly be computed if we know the

- $\mathsf{K}', \mathsf{K}''$ calibration matrices
- R', R'' viewing direction of the cameras
- S_b baseline
- or the projection matrices $\mathsf{P}', \mathsf{P}''$

How to compute F given ONLY corresponding points in images?

Problem Formulation

- **Given:** N corresponding points

$$(x', y')_n, (x'', y'')_n \quad \text{with} \quad n = 1, \ldots, N$$

- **Wanted:** fundamental matrix F

Fundamental Matrix From Corresponding Points

- For each point, we have the coplanarity constraint

$$\mathbf{x}'^{\mathsf{T}}_n \mathsf{F} \mathbf{x}''_n = 0 \qquad n = 1, ..., N$$

Fundamental Matrix From Corresponding Points

- For each point, we have the coplanarity constraint

$$\mathbf{x}'^{\mathsf{T}}_n \mathsf{F} \mathbf{x}''_n = 0 \qquad n = 1, ..., N$$

- or

$$[x'_n, y'_n, 1] \left[\boxed{\begin{matrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{matrix}} \right] \begin{bmatrix} x''_n \\ y''_n \\ 1 \end{bmatrix} = 0$$

unknowns!

What is the Issue here?

- In standard least squares problems, we have a **vector of unknowns**
- Here, the **matrix elements** of F **are the unknowns**

Question:
How to turn the unknown matrix elements into a vector of unknowns?

Linear Dependency

- **Linear function** in the unknowns F_{ij}

$$[\boxed{x'_n}, y'_n, 1] \begin{bmatrix} \boxed{F_{11}} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} \boxed{x''_n} \\ y''_n \\ 1 \end{bmatrix} = 0$$

$$\boxed{x''_n F_{11} x'_n} + x''_n F_{21} y'_n + \ldots = 0$$

Linear Dependency

- **Linear function** in the unknowns F_{ij}

$$[x'_n \; \boxed{y'_n} \; 1] \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ \boxed{F_{21}} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} \boxed{x''_n} \\ y''_n \\ 1 \end{bmatrix} = 0$$

$$x''_n F_{11} x'_n + \boxed{x''_n F_{21} y'_n} + \ldots = 0$$

Linear Dependency

- **Linear function** in the unknowns F_{ij}

$$[x'_n, y'_n, \boxed{1}] \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ \boxed{F_{31}} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} \boxed{x''_n} \\ y''_n \\ 1 \end{bmatrix} = 0$$

$$x''_n F_{11} x'_n + x''_n F_{21} y'_n + \boxed{.}\,.\,. = 0$$

Linear Dependency

- **Linear function** in the unknowns F_{ij}

$$[x'_n, y'_n, 1] \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} x''_n \\ y''_n \\ 1 \end{bmatrix} = 0$$

$$\Downarrow$$

$$[x''_n x'_n, x''_n y'_n, x''_n, y''_n x'_n, y''_n y'_n, y''_n, x'_n, y'_n, 1] \cdot$$
$$[F_{11}, F_{21}, F_{31}, F_{12}, F_{22}, F_{32}, F_{13}, F_{23}, F_{33}]^{\mathsf{T}} = 0$$
$$n = 1, \dots, N$$

Linear Dependency

- Linear function in the unknowns F_{ij}

$$\boldsymbol{a}_n^\mathsf{T} \longrightarrow [x''_n x'_n, x''_n y'_n, x''_n, y''_n x'_n, y''_n y'_n, y''_n, x'_n, y'_n, 1] \cdot$$

$$\mathbf{f} \longrightarrow [F_{11}, F_{21}, F_{31}, F_{12}, F_{22}, F_{32}, F_{13}, F_{23}, F_{33}]^\mathsf{T} = 0$$

$$n = 1, \dots, N$$

$$\Downarrow$$

$$\boldsymbol{a}_n^\mathsf{T} \mathbf{f} = 0 \qquad n = 1, \dots, N$$

Using the Kronecker Product

- Linear function in the unknowns F_{ij}

$$\boldsymbol{a}_n^{\mathsf{T}} \longrightarrow [x''_n x'_n, x''_n y'_n, x''_n, y''_n x'_n, y''_n y'_n, y''_n, x'_n, y'_n, 1] \cdot$$

$$\mathbf{f} \longrightarrow [F_{11}, F_{21}, F_{31}, F_{12}, F_{22}, F_{32}, F_{13}, F_{23}, F_{33}]^{\mathsf{T}} = 0$$

$$n = 1, \ldots, N$$

$$\Downarrow$$

$$(\mathbf{x}''_n \otimes \mathbf{x}'_n)^{\mathsf{T}} \mathrm{vec}\mathsf{F} = \underbrace{\boldsymbol{a}_n^{\mathsf{T}}}_{(\mathbf{x}''_n \otimes \mathbf{x}'_n)^{\mathsf{T}}} \underbrace{\mathbf{f}}_{\mathrm{vec}\mathsf{F}} = 0 \qquad n = 1, \ldots, N$$

(it holds in general: $\mathbf{x}^{\mathsf{T}} \mathsf{F} \mathbf{y} = (\mathbf{y} \otimes \mathbf{x})^{\mathsf{T}} \mathrm{vec}\mathsf{F}$)

Linear System From All Points

- We directly obtain a linear system if we consider **all N points**

$$\underbrace{\boldsymbol{a}_n^{\mathsf{T}}}_{(\mathbf{x}''_n \otimes \mathbf{x}'_n)^{\mathsf{T}}} \underbrace{\mathbf{f}}_{\text{vec}\mathsf{F}} = 0 \qquad n = 1, ..., N$$

$$A = \begin{bmatrix} \boldsymbol{a}_1^{\mathsf{T}} \\ \dots \\ \boldsymbol{a}_n^{\mathsf{T}} \\ \dots \\ \boldsymbol{a}_N^{\mathsf{T}} \end{bmatrix} \Rightarrow A\mathbf{f} = \mathbf{0}$$

Solving the Linear System

- Singular value decomposition solves

$$A\mathbf{f} = \mathbf{0}$$

- and thus provides a solution for

$$\mathbf{f} = [F_{11}, F_{21}, F_{31}, F_{12}, F_{22}, F_{32}, F_{13}, F_{23}, F_{33}]^{\mathsf{T}}$$

- SVD: $\mathbf{f}$ is the right-singular vector corresponding to a singular value of A that is zero

How Many Points Are Needed?

- The vector $\mathbf{f}$ has 9 dimensions

$$A = \begin{bmatrix} \boldsymbol{a}_1^\top \\ \cdots \\ \boldsymbol{a}_n^\top \\ \cdots \\ \boldsymbol{a}_N^\top \end{bmatrix} \Rightarrow A\mathbf{f} = \mathbf{0}$$

How Many Points Are Needed?

- The vector $\mathbf{f}$ has 9 dimensions

$$A = \begin{bmatrix} \boldsymbol{a}_1^\top \\ \cdots \\ \boldsymbol{a}_n^\top \\ \cdots \\ \boldsymbol{a}_N^\top \end{bmatrix} \quad \Rightarrow \quad A\mathbf{f} = \mathbf{0}$$

- Matrix A has at most rank 8
- **We need 8 corresponding points**

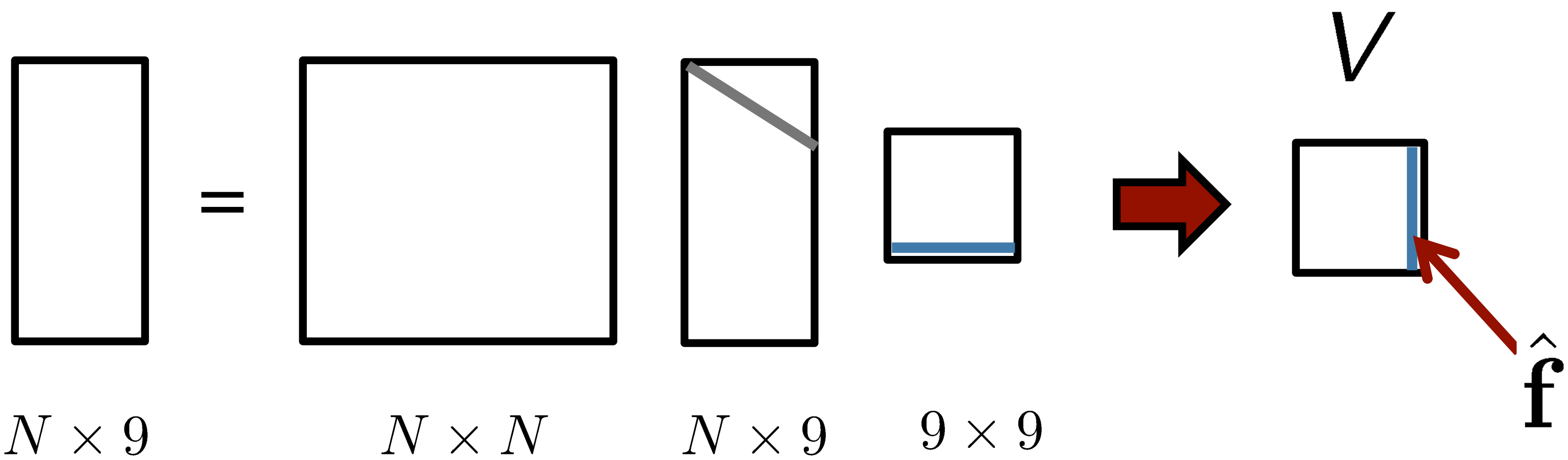
More Than 8 Points...

- In reality: noisy measurements
- With more than 8 points, the matrix A will become regular (but should not!)
- Use the singular vector $\hat{\mathbf{f}}$ of A that corresponds to the **smallest** singular value is the solution $\hat{\mathbf{f}} \rightarrow \hat{\mathsf{F}}$

Singular Vector

- Use the singular vector $\hat{\mathbf{f}}$ of A that corresponds to the **smallest** singular value is the solution $\hat{\mathbf{f}} \rightarrow \hat{\mathrm{F}}$

$$A = UDV^{\mathsf{T}}$$



8-Point Algorithm 1st Try

```
function F = F_from_point_pairs(xs, xss)
% xs, xss: Nx3 homologous point coordinates, N > 7
% F:       3x3 fundamental matrix

% coefficient matrix
for n = 1 : size(xs, 1)
    A(n, :) = kron(xss(n, :), xs(n, :));
end

```

8-Point Algorithm 1st Try

```
function F = F_from_point_pairs(xs, xss)
% xs, xss: Nx3 homologous point coordinates, N > 7
% F:       3x3 fundamental matrix

% coefficient matrix
for n = 1 : size(xs, 1)
    A(n, :) = kron(xss(n, :), xs(n, :));
end

% singular value decomposition
[U, D, V] = svd(A);

% select the singlar vector with the minimal singular value
F = reshape(V(:, 9), 3, 3);
```

singular vector of the smallest singular value

Not necessarily a matrix of rank 2
(but F should have: rank(F)=2)

Enforcing Rank 2

- We want to **enforce** a matrix F with $\operatorname{rank}(\mathsf{F}) = 2$
- F should **approximate** our computed matrix $\hat{\mathsf{F}}$ as close a possible

What to do?

Enforcing Rank 2

- We want to **enforce** a matrix F with $\mathrm{rank}(\mathsf{F}) = 2$
- F should **approximate** our computed matrix $\hat{\mathsf{F}}$ as close a possible
- Use a second SVD (this time of $\hat{\mathsf{F}}$)

  $$\mathsf{F} = UD^aV^\mathsf{T} = U\mathrm{diag}(D_{11}, D_{22}, 0)V^\mathsf{T}$$

  with $\mathrm{svd}(\hat{\mathsf{F}}) = UDV^\mathsf{T}$
  and $D_{11} \geq D_{22} \geq D_{33}$

8-Point Algorithm

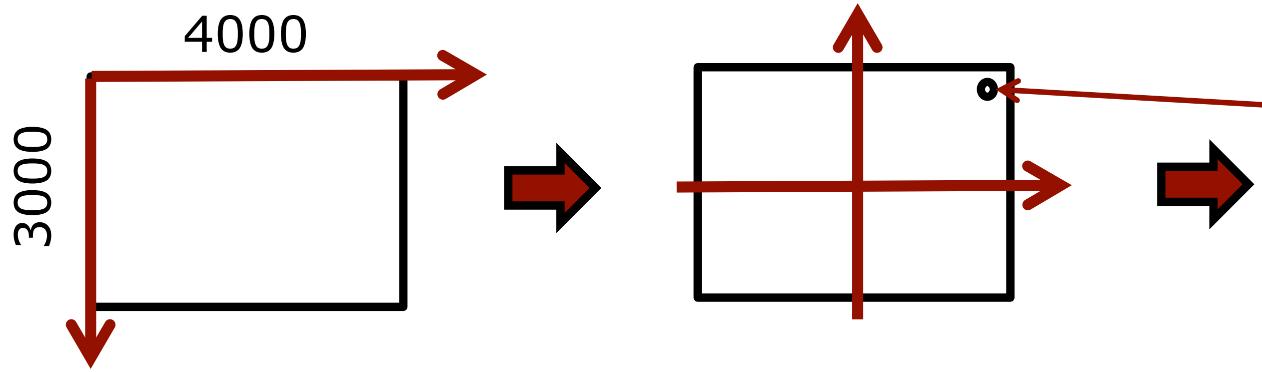
```
function F = F_from_point_pairs(xs, xss)
% xs, xss: Nx3 homologous point coordinates, N > 7
% F:       3x3 fundamental matrix

% coefficient matrix
for n = 1 : size(xs, 1)
    A(n, :) = kron(xss(n, :), xs(n, :));
end

% singular value decomposition
[U, D, V] = svd(A);

% approximate F, possibly regular
Fa = reshape(V(:, 9), 3, 3);

```

8-Point Algorithm

```
function F = F_from_point_pairs(xs, xss)
% xs, xss: Nx3 homologous point coordinates, N > 7
% F:       3x3 fundamental matrix

% coefficient matrix
for n = 1 : size(xs, 1)
    A(n, :) = kron(xss(n, :), xs(n, :));
end

% singular value decomposition
[U, D, V] = svd(A);

% approximate F, possibly regular
Fa = reshape(V(:, 9), 3, 3);

% svd decomposition of F
[Ua, Da, Va] = svd(Fa);

% algebraically best F, singular
F = Ua * diag([Da(1, 1), Da(2, 2), 0]) * Va';
```

Well-Conditioned Problem

- Example image 12MPixel camera



$$\begin{bmatrix} 1800 \\ 1400 \\ 1 \end{bmatrix}$$

- Ill-conditioned, numerically instable

$$\begin{bmatrix} 1800 \\ 1400 \\ 1 \end{bmatrix}$$ ☹

new coordinate system

$$\begin{bmatrix} 0.9 \\ 0.7 \\ 1 \end{bmatrix}$$ ☺

Conditioning/Normalization to Obtain a Well-Conditioned Problem

- Normalization of the point coordinates substantially **improves** the **stability**
- **Transform** the points so that the center of mass of all points is at (0,0)
- **Scale** the image so that the x and y coordinated are within [-1,1]

☺

Conditioning/Normalization

- Define $\mathsf{T} : \mathsf{T}\mathbf{x} = \hat{\mathbf{x}}$ so that coordinates are zero-centered and in [-1,1]
- Determine fundamental matrix $\hat{\mathsf{F}}$ from the transformed coordinates

$$
\begin{aligned}
\mathbf{x}'^{\mathsf{T}} \mathsf{F} \mathbf{x}'' &= (\mathsf{T}^{-1}\hat{\mathbf{x}}')^{\mathsf{T}} \mathsf{F} (\mathsf{T}^{-1}\hat{\mathbf{x}}'') \\
&= \hat{\mathbf{x}}'^{\mathsf{T}} \mathsf{T}^{-\top} \mathsf{F} \mathsf{T}^{-1} \hat{\mathbf{x}}'' \\
&= \hat{\mathbf{x}}'^{\mathsf{T}} \hat{\mathsf{F}} \hat{\mathbf{x}}''
\end{aligned}
$$

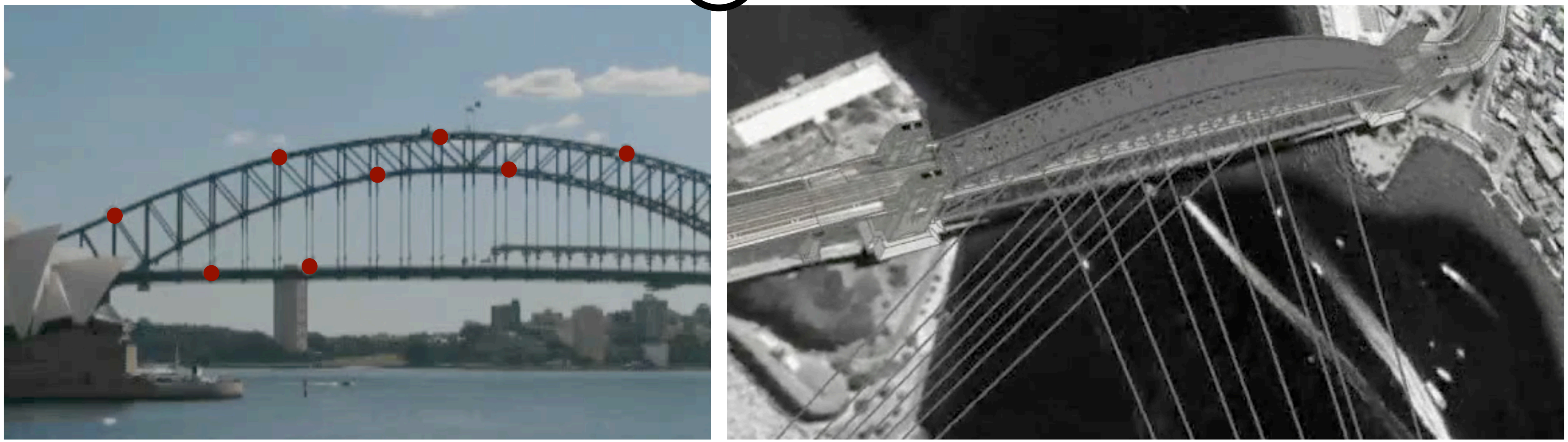
- Obtain essential matrix F through

$$
\begin{aligned}
\hat{\mathsf{F}} &= \mathsf{T}^{-\top} \mathsf{F} \mathsf{T}^{-1} \\
\mathsf{F} &= \mathsf{T}^{\mathsf{T}} \hat{\mathsf{F}} \mathsf{T}
\end{aligned}
$$

Singularity – Points on a Plane

- If all corresponding points lie on a plane, then $\mathrm{rank}(A) < 8$
- Numerically instable if points are close to a plane

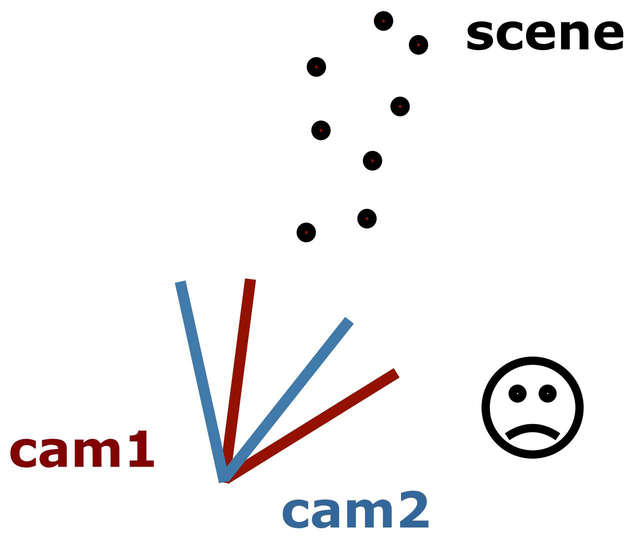
☹



Images from the “Fundamental Matrix Song” Video by D. Wedge

Singularity – No Translation

- The projection centers of both cameras are identical: $\boldsymbol{X}_{O'} = \boldsymbol{X}_{O''}$
- This happens if the translation of the camera is zero between both images



Summary so far

- Estimating the fundamental matrix from N pairs of corresponding points
- Direct solution of N>7 points based on solving a homogenous linear system ("8-Point Algorithm")

Computing the Fundamental Matrix Given 7 Corresponding Points

Direct Solution with 7 Points

- We know that the fundamental matrix has **seven degrees of freedom**
- There exists a direct solution for 7 pts

The solution itself is more complex, so just the idea should matter here

Direct Solution with 7 Points

- We know that the fundamental matrix has **seven degrees of freedom**
- There exists a direct solution for 7 pts
- Idea: 2-dimensional null space of A
- Matrix F must fulfill $\mathbf{f} = \lambda \mathbf{f}_1 + (1 - \lambda)\mathbf{f}_2$

vectors spanning the null space

Direct Solution with 7 Points

- We know that the fundamental matrix has **seven degrees of freedom**
- There exists a direct solution for 7 pts
- Idea: 2-dimensional null space of A
- Matrix F must fulfill $\mathbf{f} = \lambda \mathbf{f}_1 + (1-\lambda)\mathbf{f}_2$
- We also know that the determinant of the 3x3 matrix must be zero: $|\mathsf{F}| = 0$
- Can be combined to an equation of degree 3 $\Rightarrow$ up to three solutions

Direct Solution with 7 Points

- We know that the fundamental matrix has **seven degrees of freedom**
- There exists a direct solution for 7 pts
- Exploit
  - 2-dimensional null space of *A*
  - Determinant of the 3x3 matrix must be zero: $|\mathrm{F}| = 0$
- Can be combined to an equation of degree 3 $\Rightarrow$ up to three solutions

Summary so far

- Estimating the fundamental matrix from N pairs of corresponding points
- Direct solution of N>7 points based on solving a homogenous linear system ("8 point algorithm")
- Idea for a direct solution with 7 points (up to 3 solutions)

Let’s Do the Same for the Essential Matrix

Reminder: Essential Matrix

- **Essential matrix** = "fundamental matrix for calibrated cameras"

$$\boxed{\mathsf{E} = R'\mathsf{S}_b R''^{\mathsf{T}}}$$

- Often parameterized through (general parameterization of dependent images)

$$\boxed{\mathsf{E} = \mathsf{S}_b R^{\mathsf{T}}}$$

- Coplanarity constraint for calibrated cameras

$$\boxed{{}^{k}\mathbf{x}'^{\mathsf{T}}\,\mathsf{E}\,{}^{k}\mathbf{x}'' = 0}$$

Essential Matrix from 8+ Corresponding Points

- For each point, we have the coplanarity constraint

$$ {}^{k}\mathbf{x}'^{\mathsf{T}}_{n} \, \mathsf{E} \; {}^{k}\mathbf{x}''_{n} = 0 \qquad n = 1, ..., N $$

- **Note:** Same equation as for the fundamental matrix but for the points in the camera c.s. Remember: ${}^{k}\mathbf{x}' = (\mathsf{K}')^{-1}\mathbf{x}'$

Essential Matrix from 8+ Corresponding Points

- For each point, we have the coplanarity constraint

$$
{}^{k}\mathbf{x}'^{\mathsf{T}}_{n}\;\mathsf{E}\;{}^{k}\mathbf{x}''_{n} = 0 \qquad n = 1, \ldots, N
$$

- or

$$
[\,{}^{k}x'_{n},\; {}^{k}y'_{n}, c'] \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{bmatrix} \begin{bmatrix} {}^{k}x''_{n} \\ {}^{k}y''_{n} \\ c'' \end{bmatrix} = 0
$$

As for the Fundamental Matrix...

$$[\,{}^k x'_n,\ {}^k y'_n, c'] \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{bmatrix} \begin{bmatrix} {}^k x''_n \\ {}^k y''_n \\ c'' \end{bmatrix} = 0$$

```
function E = E_from_point_pairs(xs, xss)
% xs, xss: Nx3 homologous point coordinates, N > 7
% E:       3x3 essential matrix

% coefficient matrix
for n = 1 : size(xs, 1)
    A(n, :) = kron(xss(n, :), xs(n, :));
end

% singular value decomposition
[U, D, V] = svd(A);

% select the singlar vector with the minimal singular value
E = reshape(V(:, 9), 3, 3)';
```

build matrix A

solve Ae=0

build matrix E

Which constraints to consider?

Constraints

- For the fundamental matrix, we enforced the $\mathrm{rank}(\mathsf{F}) = 2$ constraint

$$\mathsf{F} = UDV^{\mathsf{T}} = U \begin{bmatrix} D_{11} & 0 & 0 \\ 0 & D_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix} V^{\mathsf{T}}$$

- For the essential matrix, both non-zero singular values are identical

$$\boxed{\mathsf{E} = U \begin{bmatrix} d & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & 0 \end{bmatrix} V^{\mathsf{T}} \underset{\substack{\uparrow \\ \text{homogenous}}}{=} U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^{\mathsf{T}}}$$

More details: Förstner, Skript Photogrammetrie II, Sect 1.2.3

8-Point Algorithm for the Essential Matrix

```
function E = E_from_point_pairs(xs, xss)
% xs, xss: Nx3 homologous point coordinates, N > 7
% E:        3x3 essential matrix

% coefficient matrix
for n = 1 : size(xs, 1)
    A(n, :) = kron(xss(n, :), xs(n, :));
end

% singular value decomposition
[U, D, V] = svd(A);

% approximate E, possibly regular
Ea = reshape(V(:, 9), 3, 3);

% svd decomposition of E
[Ua, Da, Va] = svd(Ea);

% algebraically best E, singular, same singular values
E = Ua * diag([1, 1, 0]) * Va';
```

build matrix A

solve Ae=0

build matrix Ea

compute SVD of Ea

build matrix E from Ea by imposing constraints

Conditioning/Normalization to Obtain a Well-Conditioned Problem (As Done Before)

- As for the 8-Point algorithm for the fundamental matrix, normalization of the point coordinates is **essential**
- **Transform** the points so that the center of mass of all points is at (0,0)
- **Scale** the image so that the x and y coordinated are within [-1,1]

Conditioning/Normalization

- Define $\mathsf{T} : \mathsf{T}\mathbf{x} = \hat{\mathbf{x}}$ so that coordinates are zero-centered and in [-1,1]
- Determine essential matrix $\hat{\mathsf{E}}$ from the transformed coordinates

$$
\begin{aligned}
{}^k\mathbf{x}'^{\mathsf{T}} \mathsf{E}\, {}^k\mathbf{x}'' &= (\mathsf{T}^{-1}\, {}^k\hat{\mathbf{x}}')^{\mathsf{T}} \mathsf{E} (\mathsf{T}^{-1}\, {}^k\hat{\mathbf{x}}'') \\
&= {}^k\hat{\mathbf{x}}'^{\mathsf{T}} \mathsf{T}^{-\top} \mathsf{E} \mathsf{T}^{-1}\, {}^k\hat{\mathbf{x}}'' \\
&= {}^k\hat{\mathbf{x}}'^{\mathsf{T}} \hat{\mathsf{E}}\, {}^k\hat{\mathbf{x}}''
\end{aligned}
$$

- Obtain essential matrix E through

$$
\Rightarrow \quad
\begin{aligned}
\hat{\mathsf{E}} &= \mathsf{T}^{-\top} \mathsf{E} \mathsf{T}^{-1} \\
\mathsf{E} &= \mathsf{T}^{\mathsf{T}} \hat{\mathsf{E}} \mathsf{T}
\end{aligned}
$$

Properties of the Essential Mat.

- Homogenous
- Singular: $|\mathsf{E}| = 0$ (determinant is zero)
- Two identical non-zero singular values

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

- As a result of the skew-sym. matrix:

$$2\mathsf{E}\mathsf{E}^{\mathsf{T}}\mathsf{E} - \operatorname{tr}(\mathsf{E}\mathsf{E}^{\mathsf{T}})\mathsf{E} = \underset{3\times3}{\mathbf{0}}$$

5-Point Algorithm

5-Point Algorithm

- Proposed by Nistér in 2003/2004
- Standard solution today to obtaining the direct solution
- Solving a polynomial of degree 10
- 10 possible solutions
- Often used together RANSAC
  - RANSAC proposes correspondences
  - Evaluate all 5-point solutions based on the other corresponding points

5-Point Algorithm

- More details in the script by Förstner "Photogrammetrie II", Ch 1.2
- Stewenius, Engels, Nistér: "Recent Developments on Direct Relative Orientation", ISPRS 2006
- Li and Hartley: "Five-Point Motion Estimation Made Easy"

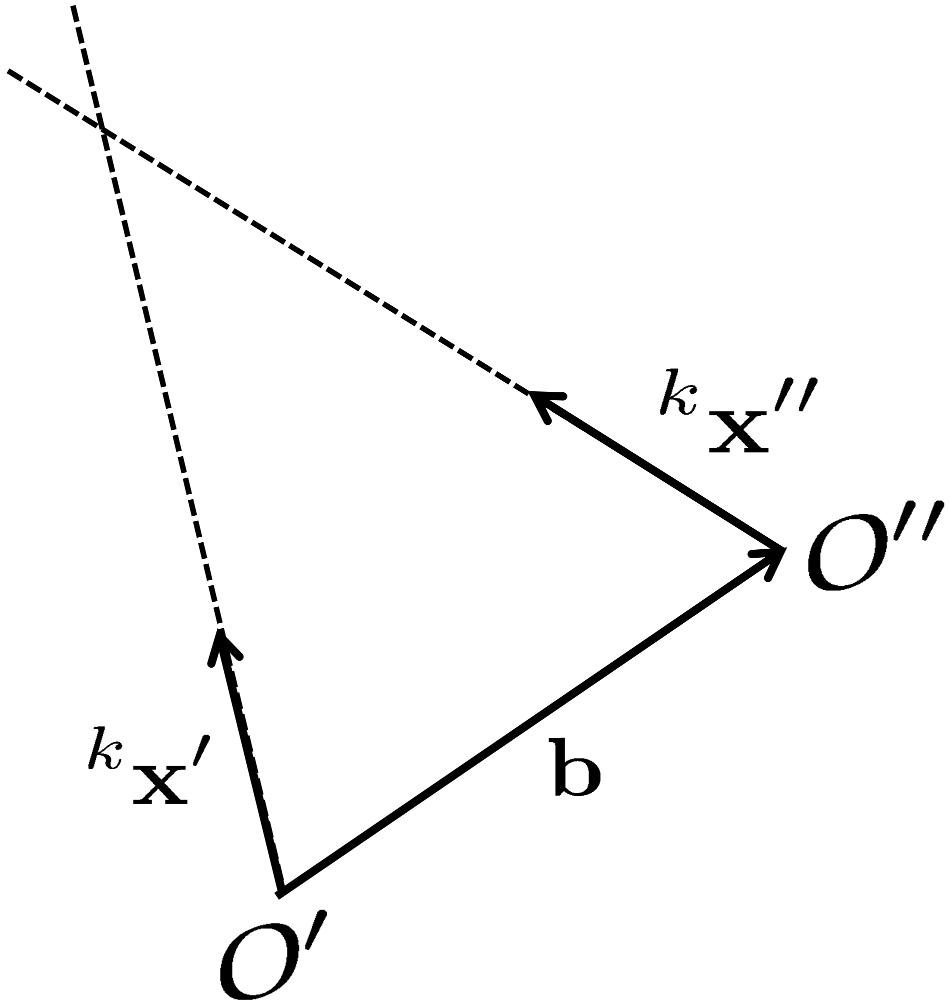
Computing the Orientation Parameters Given the Essential Matrix

Compute Basis and Rotation Given E

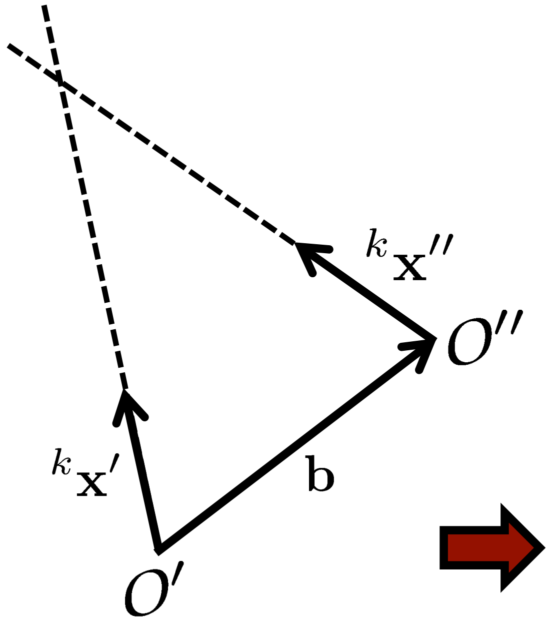
- In short: $\mathrm{E} \rightarrow S_B, R$

Question: Is there a unique solution?

The Solution We Want…

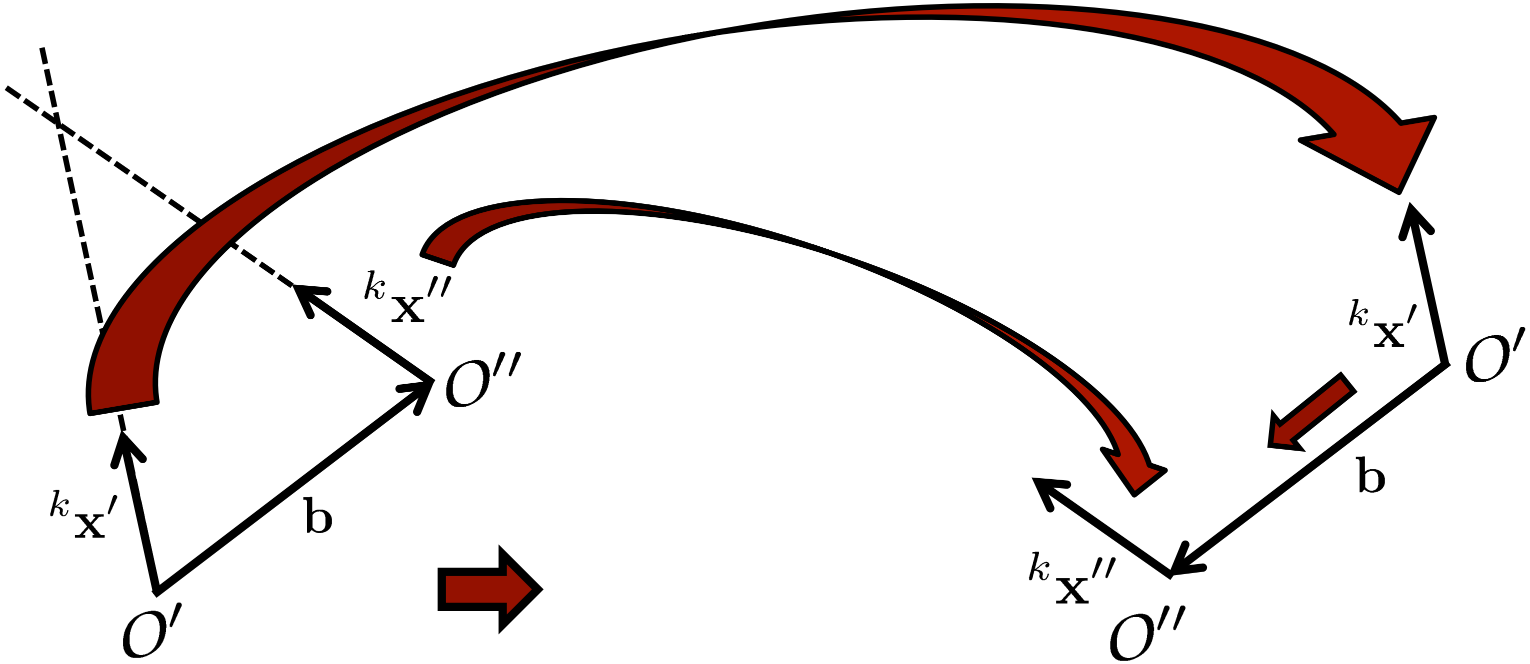


Multiple Solutions from Math...



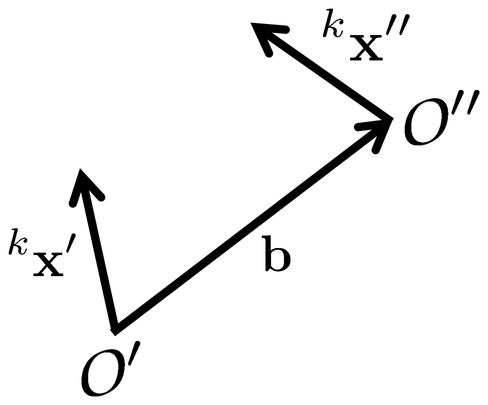
We only know b up to a scalar.
So we can multiply it by -1...

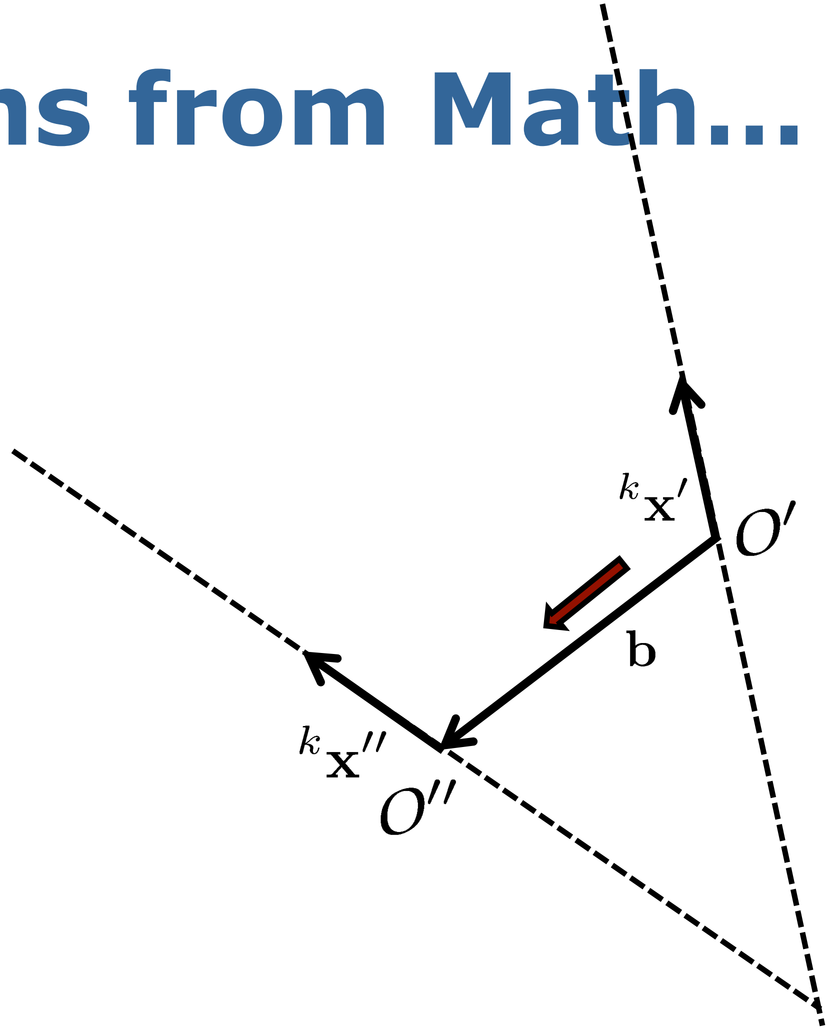
Multiple Solutions from Math...



We only know b up to a scalar.
So we can multiply it by -1…

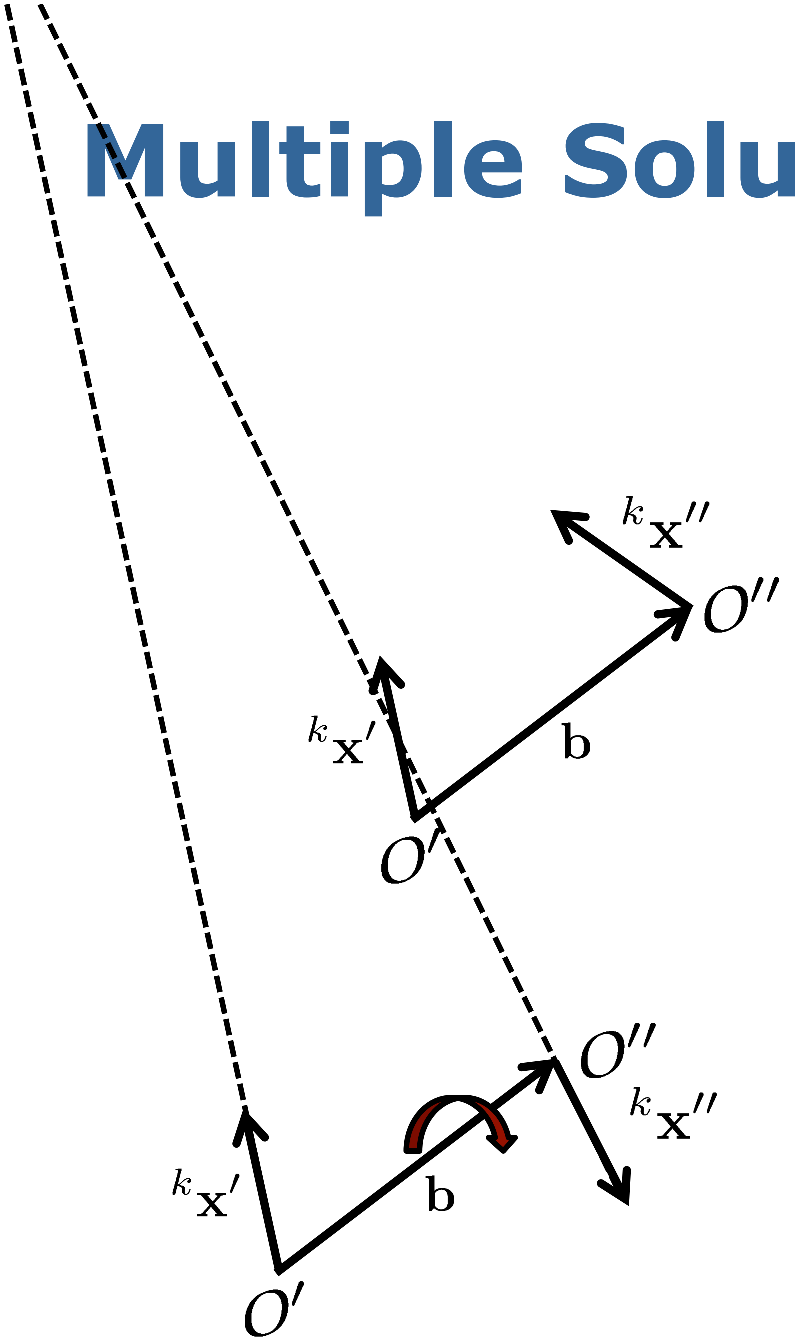
Multiple Solutions from Math...

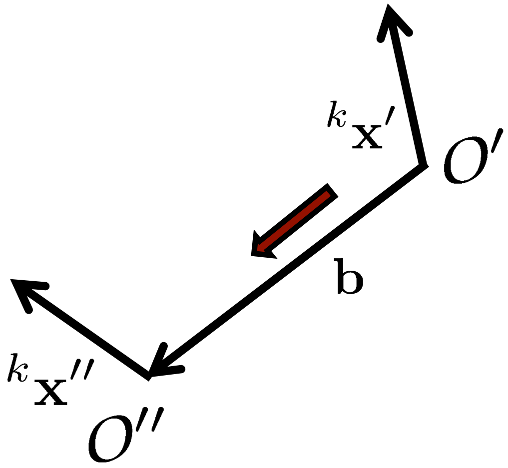




We can also rotate the (second) camera by PI

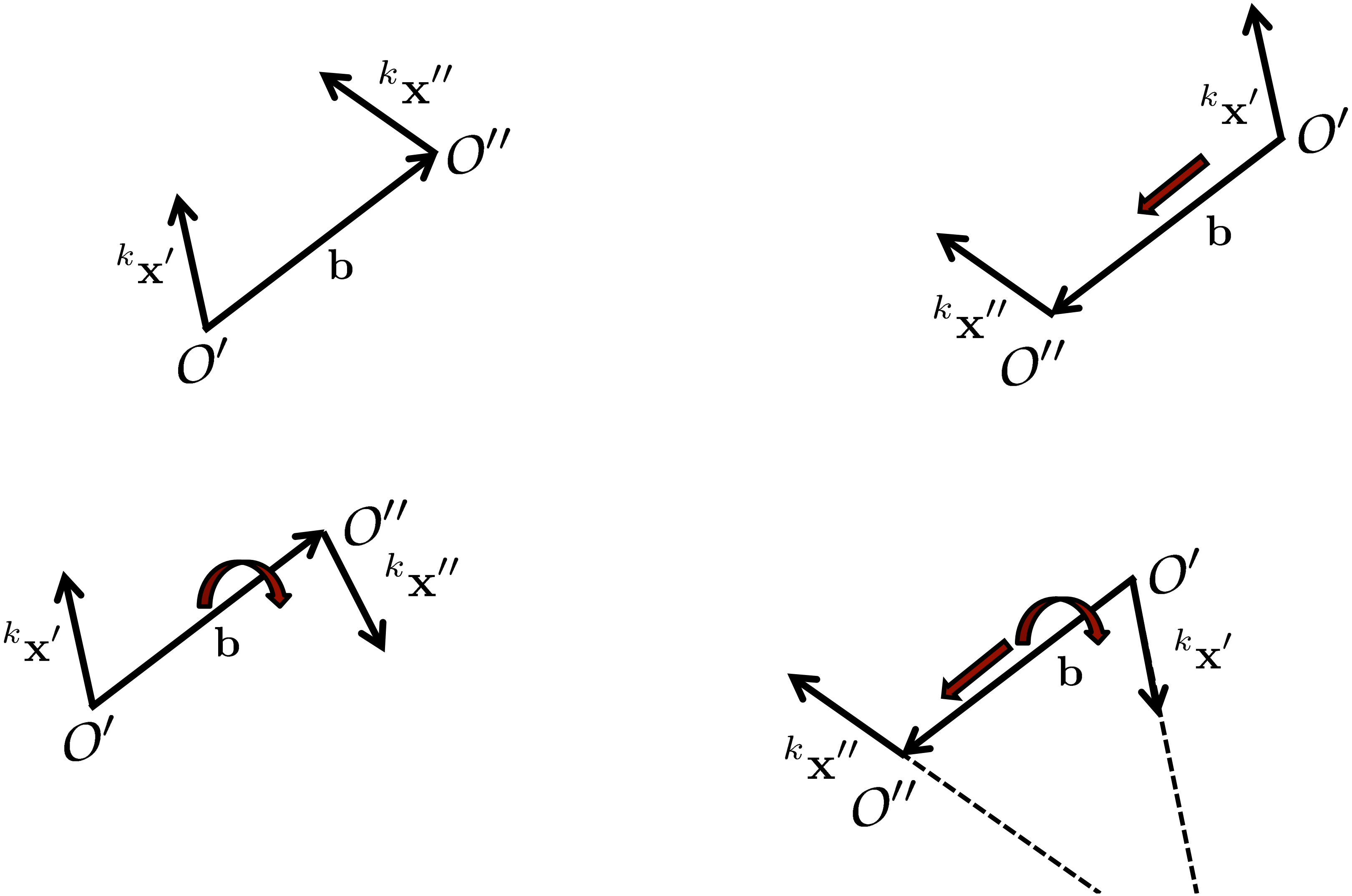
Multiple Solutions from Math…



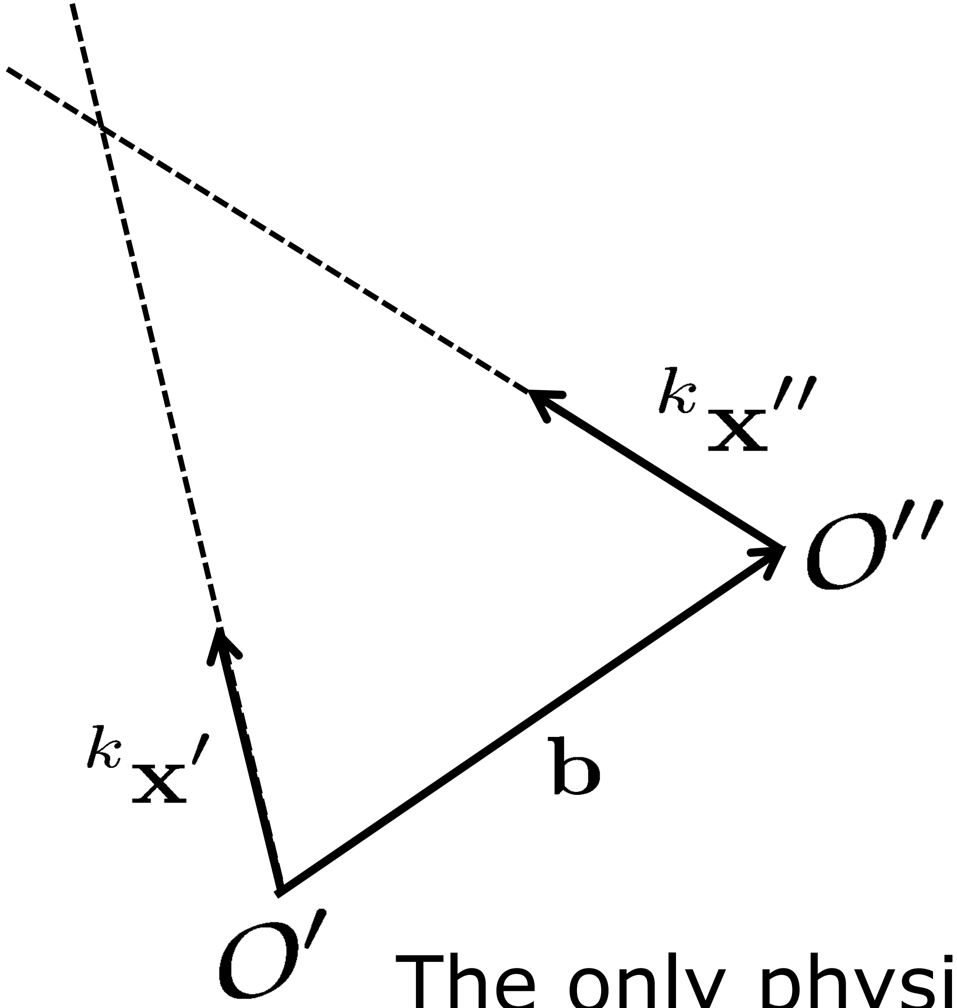


Or do both…

Four Possible Solutions from Math…



One Solution from Physics...



The only physically plausible solution that that the point are **in front of both cameras.**

Algebraic Solution

for Obtaining the Basis and Rotation Matrix Given the Essential Matrix

Solution by Hartley & Zisserman

- We know that

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

rotation matrices

Solution by Hartley & Zisserman

- We know that

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

rotation matrices

- Define $Z = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ $W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

skew-sym. mat rotation mat

- So that $ZW = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Solution by Hartley & Zisserman

$$
\begin{aligned}
\mathsf{E} &= U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^{\mathsf{T}} \\
&= U \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{Z} \underbrace{\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{W} V^{\mathsf{T}}
\end{aligned}
$$

Solution by Hartley & Zisserman

$$
\begin{aligned}
\mathsf{E} &= U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^{\mathsf{T}} \\
&= U \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{Z} \underbrace{\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{W} V^{\mathsf{T}} \\
&= U Z \underbrace{U^{\mathsf{T}} U}_{I} W V^{\mathsf{T}}
\end{aligned}
$$

Solution by Hartley & Zisserman

$$
\begin{aligned}
\mathsf{E} &= U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^\mathsf{T} \\
&= U \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{Z} \underbrace{\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{W} V^\mathsf{T} \\
&= U Z \underbrace{U^\mathsf{T} U}_{I} W V^\mathsf{T} \\
&= \underbrace{U Z U^\mathsf{T}}_{S_B} \underbrace{U W V^\mathsf{T}}_{R^\mathsf{T}}
\end{aligned}
$$

Four Possibilities to Define Z, W

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = ZW = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= -Z^{\mathsf{T}}W = -\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= -ZW^{\mathsf{T}} = -\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{\mathsf{T}}$$

$$= Z^{\mathsf{T}}W^{\mathsf{T}} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{\mathsf{T}}$$

Yields Four Solutions

$$\mathrm{E} = \underbrace{UZU^{\mathsf{T}}}_{S_B} \underbrace{UWV^{\mathsf{T}}}_{R^{\mathsf{T}}}$$

2 solutions for $\mathrm{S_B}$

2 solutions for R

$$\mathrm{S}^1_{\widehat{B}} = UZU^{\mathsf{T}} \quad \mathrm{S}^2_{\widehat{B}} = UZ^{\mathsf{T}}U^{\mathsf{T}} \quad R_1^{\mathsf{T}} = UWV^{\mathsf{T}} \quad R_2^{\mathsf{T}} = UW^{\mathsf{T}}V^{\mathsf{T}}$$

4 solutions

$$\mathrm{E}^1 = UZU^{\mathsf{T}}\ UWV^{\mathsf{T}}$$

$$\mathrm{E}^2 = UZ^{\mathsf{T}}U^{\mathsf{T}}\ UWV^{\mathsf{T}}$$

$$\mathrm{E}^3 = UZU^{\mathsf{T}}\ UW^{\mathsf{T}}V^{\mathsf{T}}$$

$$\mathrm{E}^4 = UZ^{\mathsf{T}}U^{\mathsf{T}}\ UW^{\mathsf{T}}V^{\mathsf{T}}$$

Solution by Hartley & Zisserman

- Compute the SVD of E: $UDV^{\mathsf{T}} = \mathrm{svd}(\mathsf{E})$
- Normalize U, V by $U = U|U|\,, V = V|V|$
- Compute the four solutions

$$\mathsf{S}^1_{\widehat{B}} = UZU^{\mathsf{T}} \quad \mathsf{S}^2_{\widehat{B}} = UZ^{\mathsf{T}}U^{\mathsf{T}} \quad R_1^{\mathsf{T}} = UWV^{\mathsf{T}} \quad R_2^{\mathsf{T}} = UW^{\mathsf{T}}V^{\mathsf{T}}$$

- Test for which solutions all points are in front of both cameras
- Return the physically plausible configuration

Summary (1)

- Algorithms to compute the relative orientation from image data
- Allow us to estimate the camera motion (except of the scale)
- Direct solutions
  - F from N>7 points ("8-Point Algorithm")
  - E from N>7 points ("8-Point Algorithm")
  - E from N=5 points ("Nister's 5-Point Algorithm")

Summary (2)

- Direct solutions
- Extracting S_B, R from E
- Not statistically optimal
- Often used in combination with RANSAC for identifying in/outliers
- Direct solutions & RANSAC serves as initial guess for iterative solutions
- Subsequent refinement using least squares only based on inlier points

Literature

- Förstner, Wrobel: Photogrammetric Computer Vision, Ch. 12.3.1-12.3.3
- Hartley: In Defence of the 8-point Algorithm
- Stewenius, Engels, Nistér: Recent Developments on Direct Relative Orientation, ISPRS 2006

Slide Information

- The slides have been created by Cyrill Stachniss as part of the photogrammetry and robotics courses.
- I tried to acknowledge all people from whom I used images or videos. In case I made a mistake or missed someone, please let me know.
- The photogrammetry material heavily relies on the very well written lecture notes by Wolfgang Förstner and the Photogrammetric Computer Vision book by Förstner & Wrobel.
- Parts of the robotics material stems from the great Probabilistic Robotics book by Thrun, Burgard and Fox.
- If you are a university lecturer, feel free to use the course material. If you adapt the course material, please make sure that you keep the acknowledgements to others and please acknowledge me as well. To satisfy my own curiosity, please send me email notice if you use my slides.

Cyrill Stachniss, cyrill.stachniss@igg.uni-bonn.de, 2014