# MAEG 5720: Computer Vision in Practice

## Lecture 12:

## Epipolar Geometry and Stereo Vision


Dr. Terry Chang


2021-2022

Semester 1


香港中文大學
The Chinese University of Hong Kong

Department of Mechanical and Automation Engineering
機械與自動化工程學系

# Recall – Back Projection of points to rays

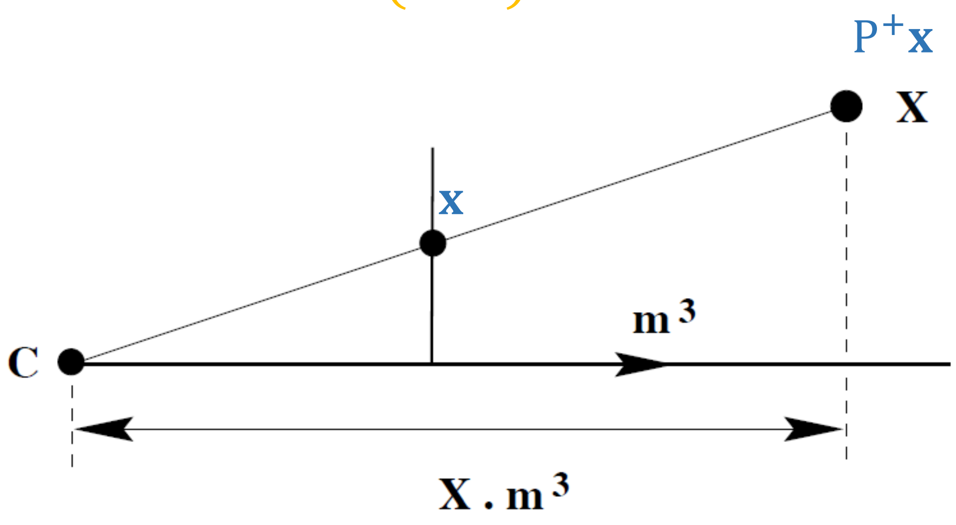
**Back-projection of points to rays:**

- Given a point $\mathbf{x}$ in an image, we wish to determine the set of points in space map to this point.
- We know two points on the ray.
  - The ***camera centre*** $\mathbf{C}$ where $\mathrm{P}\mathbf{C} = \mathbf{0}$
  - The point $\mathrm{P}^{+}\mathbf{x}$ where $\mathrm{P}^{+}$ is ***pseudo-inverse*** of $\mathrm{P}$ where $\mathrm{P}^{+} = \mathrm{P}^{\mathrm{T}}(\mathrm{P}\mathrm{P}^{\mathrm{T}})^{-1}$
- The point $\mathrm{P}^{+}\mathbf{x}$ lies on the ray because

$$\mathrm{P}(\mathrm{P}^{+}\mathbf{x}) = \mathrm{I}\mathbf{x} = \mathbf{x}$$

- Hence the ray is a line

$$\boxed{\mathbf{X}(\lambda) = \mathrm{P}^{+}\mathbf{x} + \lambda\mathbf{C}}$$



# Motivation

- Given a single image, we are not able to infer its three-dimensional structure as the depth information is lost.

- Therefore image pairs are considered!

# Today’s Agenda

- Number of Parameters
- Coplanarity Constraints for Image of Uncalibrated Cameras
- Fundamental Matrix
- Epi-polar Geometry
- Estimation of Fundamental Matrix by 8 Point Theorem
- The Essential Matrix
- Estimation of Essential Matrix by 8 Point Theorem

# Principle of two-view analysis

- A scene point $\mathcal{X}$ is projected into two images forming $\mathcal{X}'$ and $\mathcal{X}''$

- $\mathcal{X}'$ and $\mathcal{X}''$ are *corresponding image points* representing the projection of $\mathcal{X}$ on the two image planes.

- Tasks
  - Determine the *orientation* of the image pairs
  - Reconstruct the *coordinates* of the scene features observed in the image pairs

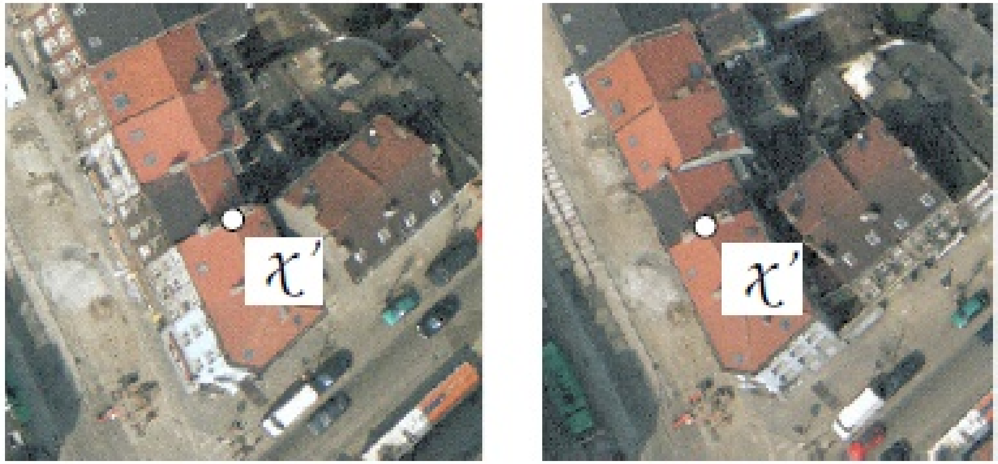
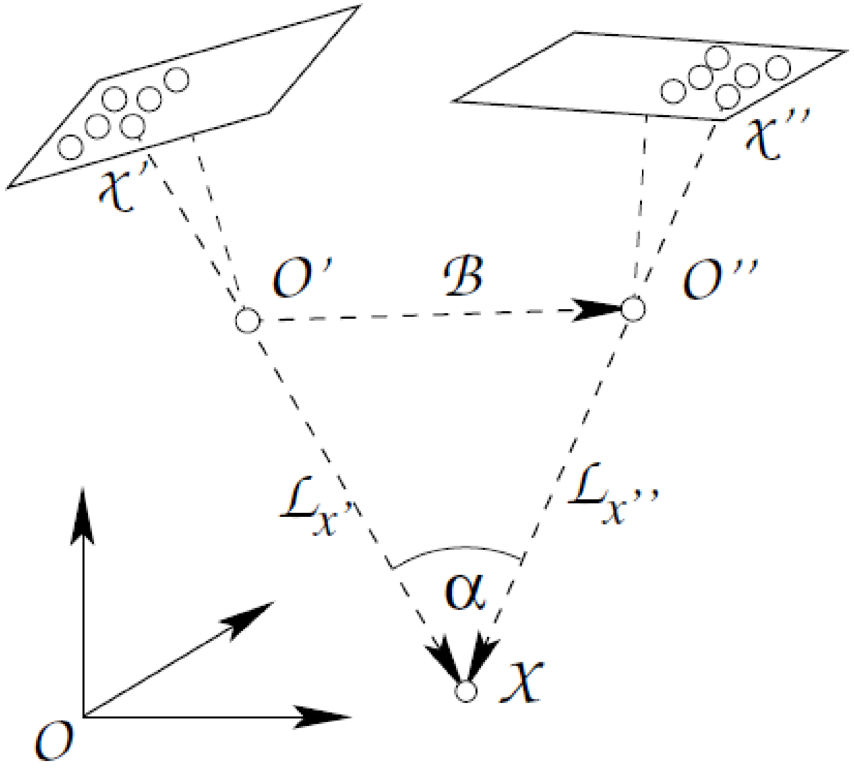



Image Credit: Fronster

# Representation of image pairs

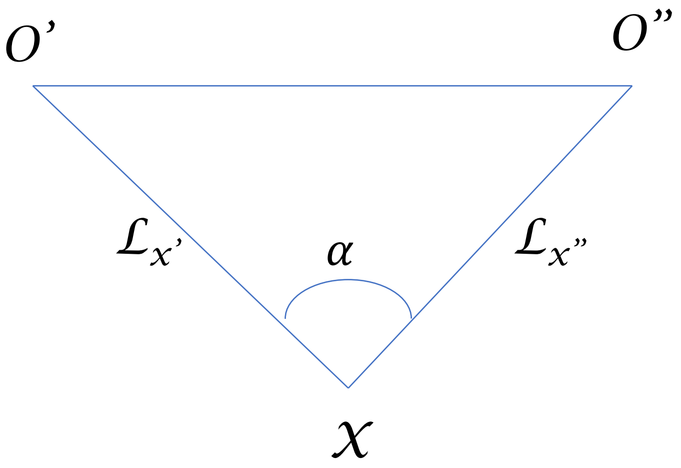



- $\mathcal{X}$ : 3D Point in Scene
- $\mathcal{X}', \mathcal{X}''$ : Projection of $\mathcal{X}$ on both images
- $\mathcal{L}_{x'}, \mathcal{L}_{x''}$ : Lines joining $\mathcal{X}\mathcal{X}'$ and $\mathcal{X}\mathcal{X}''$ respectively
- $O'$ and $O''$ : projection centers
- $\mathcal{B}$ : Line joining the projection centers and

Image Credit: Fronster

# Geometry of image pairs

- Given $X', X''$ on both images. With No prior knowledge of 3D Scenes
- $\mathcal{L}_{x'}$, $\mathcal{L}_{x''}$ are projection of rays to the 3D Scene through $O'$ and $O''$ must intersection and lies on a plane.
- $\alpha$ is the angle between $\mathcal{L}_{x'}$, $\mathcal{L}_{x''}$ ***parallactic angle*** between two projection rays.
-
- This gives the ***coplanarity constraints***.
- We can then obtain the ***Relative Orientations*** describing the geometry relations of the image pairs up to an unknown scale parameter



# The Number of Orientation Parameters

- Calibrated case,
  - For *calibrated camera*, number of parameters required is 6 (extrinsic parameters (three rotations and three translations)
  - For *calibrated camera pair*, there are *12 parameters*

- *Extra control points* or *lines* in the object space are required to determine the *exterior orientation* of the camera.

# The Number of Orientation Parameters

- Uncalibrated case
  - For *uncalibrated camera*, we assume
    - non-linear error sufficiently small,
    - Camera has different intrinsic parameters
    - Two camera follow a straight line-preserving perspective camera model
  - number of parameters required is *6*(extrinsic)+*5*(intrinsic) = *11 parameters*
  - For *uncalibrated camera pair*, there are *22 parameters*

- *Extra control points* or *lines* in the object space are required to determine the *exterior orientation* of the camera.

# Relative Orientation of image pairs

- *Relative Orientation* refers to the determination of the *relative baseline (translations) vector* of two projection centers and the *relative rotations* of one image relative to the other.

- Without prior knowledge of the 3D Scenes, how many parameters we can retrieve?

- What parameters we can retrieve?

# For Calibrated Camera case

- Calibrated camera pairs have *12 parameters*

- Assumption:
  - Two camera projection centers are distinct
  - Angle Preserving projection
  - $\mathcal{L}_{x'}$, $\mathcal{L}_{x''}$ intersect $\mathcal{X}$ in the scene

- Object can be reconstructed up to *similarity transformation* which is has *7 DoF*
  - Three translation parameters
  - Three rotation parameters
  - One scale parameter

- Number of parameters we can obtain is *12-7=5 parameters*

# Parameters related to relative orientation

- We fixed $O'$, what motions of the $O''$ to maintain the intersection of $\mathcal{L}_{x'}$, $\mathcal{L}_{x''}$?

- $O''$ can only move alone direction of baseline $\mathcal{B}(O'O'')$

- What parameters we can obtain?

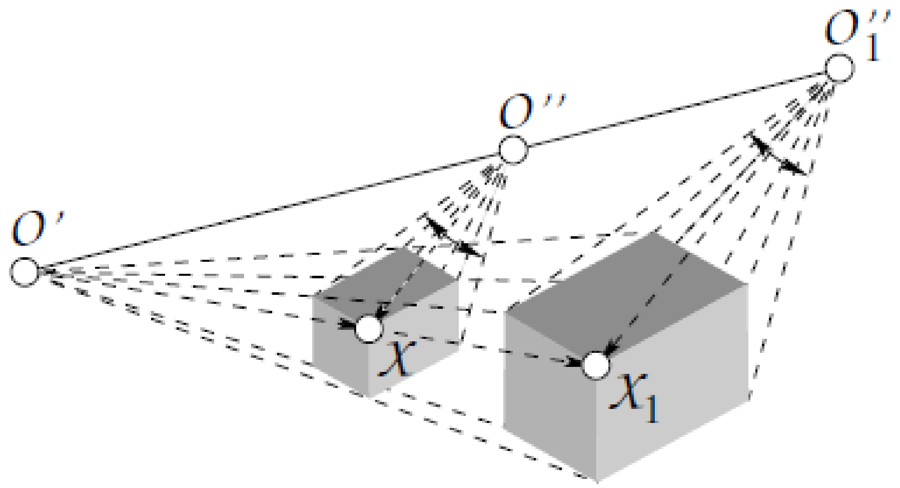



Image Credit: Fronster

# What parameters can we obtain for calibrated camera pairs?

- $O$" can only move alone direction of baseline $\mathcal{B}(O'O")$

- The following parameters we can obtain
  - Rotation $R_{12}$ of the second camera relative to the first (*3 parameters*)
  - The direction $\mathcal{B}(O'O")$ connecting two projection centers (*2 parameters*)

- Number of parameters we CANNOT obtain = *12-5=7 parameters*

- Number of *control points* required (7 parameters) = *3 points*

# For Uncalibrated Cameras

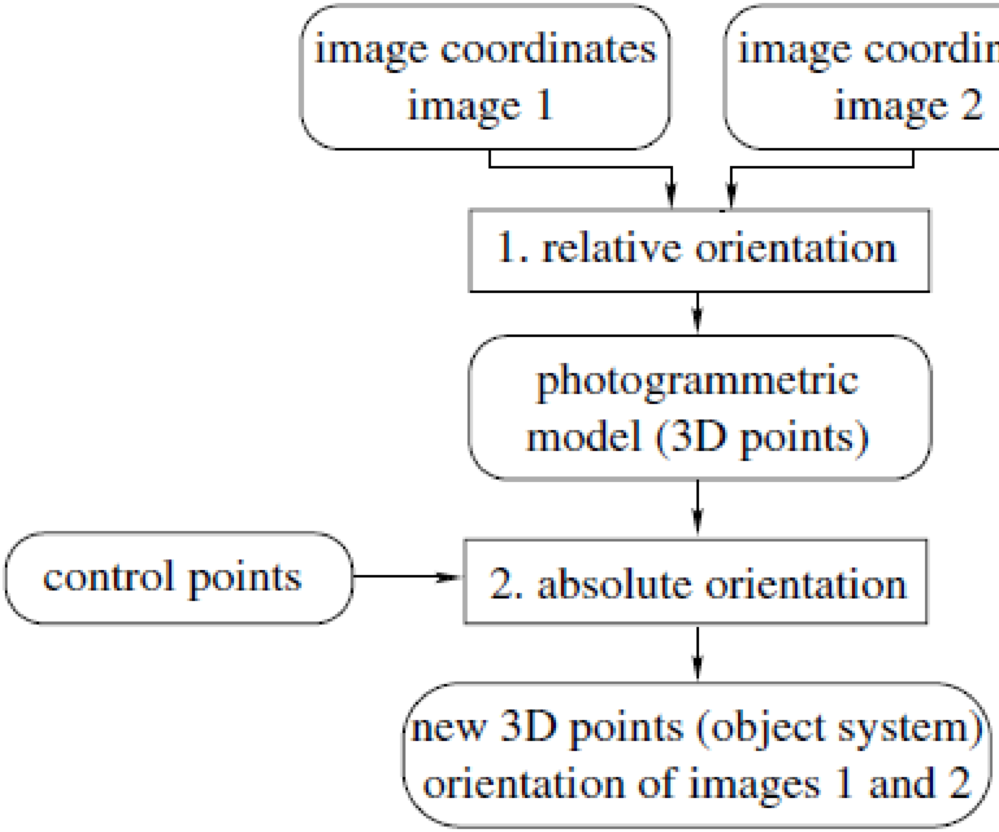
- For uncalibrated image pairs, we have totally *22 parameters*
- Assume *Straight line-preserving* perspective cameras
- It is represented by a spatial homography (4x4 matrix) with *15 parameters*.

- Number of parameters we CANNOT obtain is *15 parameters*

- Number of parameters we can obtain is 22-15 = *7 parameters*

- Number of control points required (for 15 parameters) = *5 points*

# Two-Step procedures for the orientation of image pair



# Summary of Parameters and Control Points

| Camera | Parameters for single camera | Parameters for image pair | Relative Orientation | Absolute Orientation | Number of Control Points |
|---|---|---|---|---|---|
| Calibrated | 6 | 12 | 5 | 7 | 3 |
| UnCalibrated | 11 | 22 | 7 | 15 | 5 |

# Today's Agenda

- Number of Parameters
- Coplanarity Constraints for Image of Uncalibrated Cameras
- Fundamental Matrix
- Epi-polar Geometry
- Estimation of Fundamental Matrix by 8 Point Theorem
- The Essential Matrix
- Estimation of Essential Matrix by 8 Point Theorem

# The Coplanarity Constraint for images of uncalibrated camera

- Given: Two images taken with uncalibrated line-preserving cameras

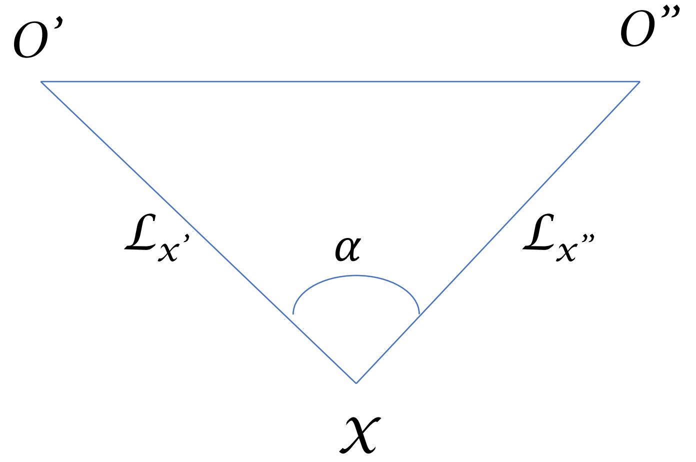
- Let two camera characterized by two projection matrices P' and P" such that

$$x' = P'X \qquad\qquad x'' = P''X$$

*Camera Projection*
$\lambda \boldsymbol{x} = P\boldsymbol{X}$

with

$$P' = K'R'[I_3 | -Z'] \qquad\qquad P'' = K''R''[I_3 | -Z'']$$



# The Coplanarity Constraint

- Since $O'X'$ ,$O'O''$, $O''X'$ are coplanar, we have

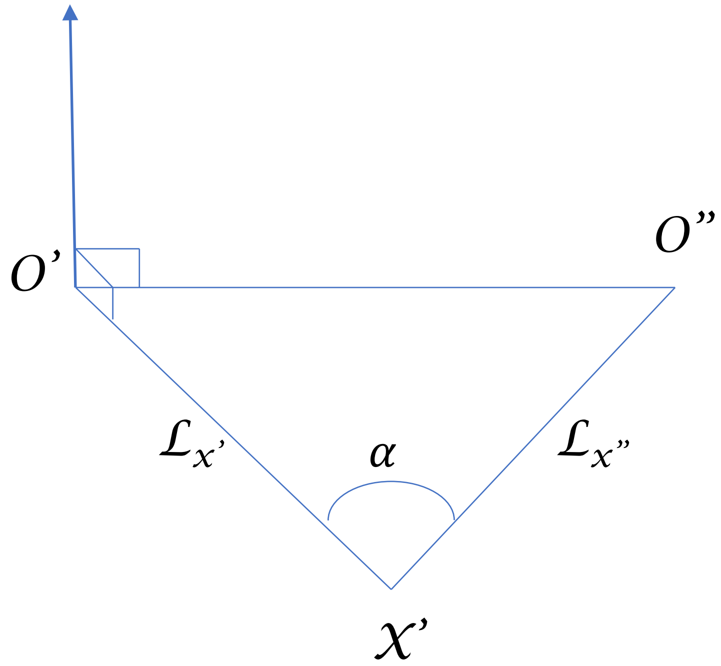
$$O'X' \times O'O'' \perp O''X'$$

Therefore

$$(O'X' \times O'O'') \cdot O''X' = 0$$

This is represented by scalar triple product below:

$$\det[O'X' , O'O'', O''X']=0$$

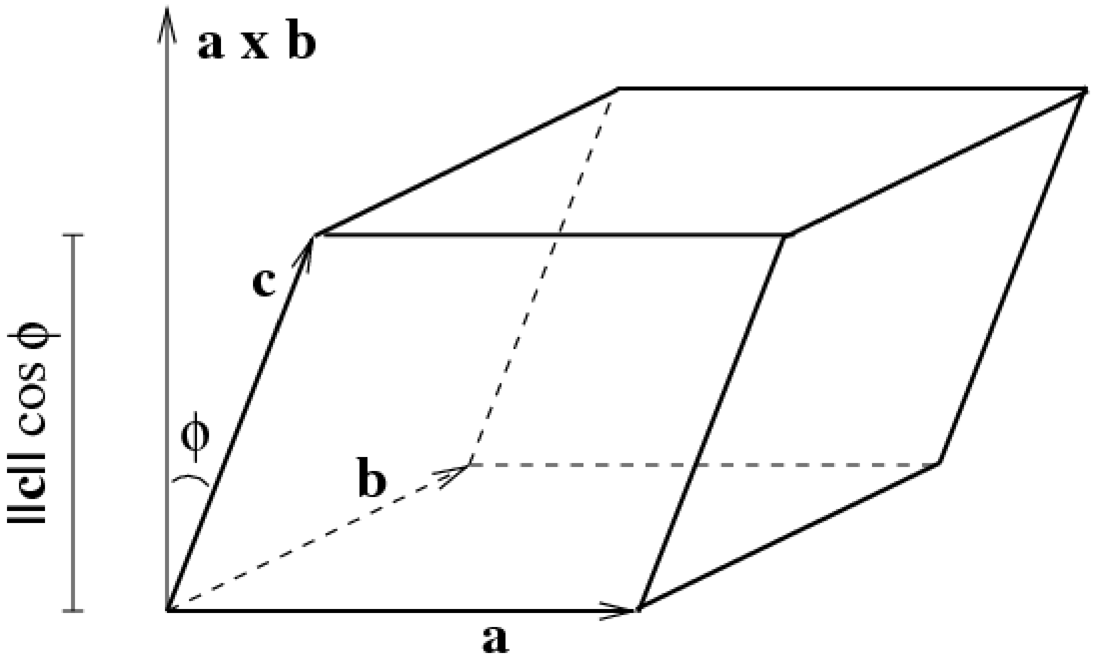


# Scalar Triple Product

- The Scalar Triple Product represents the volume of parallelepiped

- Volume = $a \times b \cdot c$

- What if a, b and c are coplanar?




https://mathinsight.org/scalar_triple_product

# Coplanarity Constraint

- Assume Normalized Camera System, the *normalized direction* $O'X'$ and $O'X''$ is given by

$$^{n}\mathbf{x}' = (R')^{T}(\mathrm{K}')^{-1}\mathbf{x}'$$
$$^{n}\mathbf{x}'' = (R'')^{T}(\mathrm{K}'')^{-1}\mathbf{x}''$$

**Recall-**

$$\mathbf{X} = (\mathrm{K}R)^{-1}\mathrm{K}R\mathbf{X}_{\mathbf{0}} + \lambda(\mathrm{K}R)^{-1}\mathbf{x}$$
$$\mathbf{X} = \mathbf{X}_{\mathbf{0}} + \lambda R^{T}\mathrm{K}^{-1}\mathbf{x}$$

The *base vector* $O'O''$ can be represented directly from coordinates of projection centres

$$\mathbf{b} = \boldsymbol{B} = \boldsymbol{Z}'' - \boldsymbol{Z}'$$

# Coplanarity Constraint

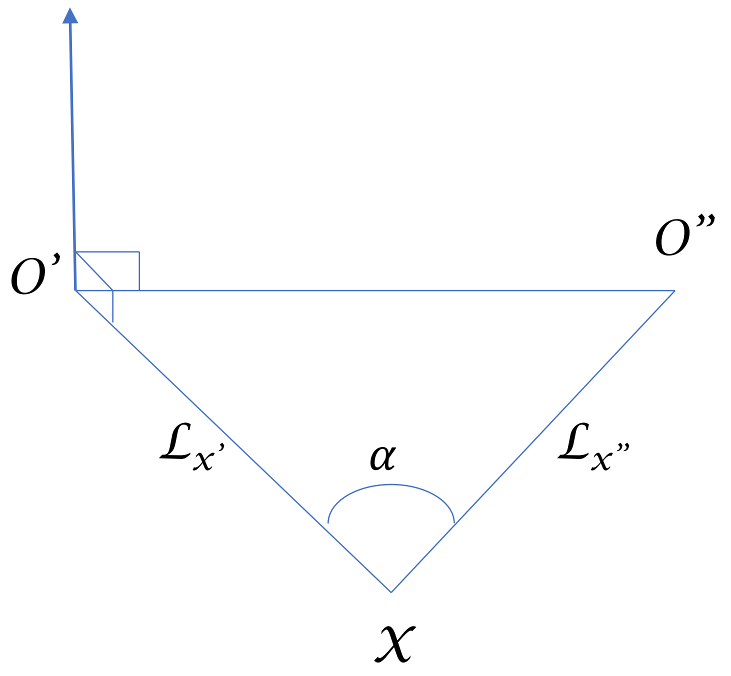
- The coplanarity constraint

$$\det[\mathcal{O'X'}, \mathcal{O'O''}, \mathcal{O''X'}]=0$$

can be rewritten as

$$\det[{}^{n}\mathbf{x}', \mathbf{b}\ {}^{n}\mathbf{x}'']=0$$

$$\left| {}^{n}\mathbf{x}', b, {}^{n}\mathbf{x}'' \right| = \begin{vmatrix} {}^{n}x' & B_x & {}^{n}x'' \\ {}^{n}y' & B_y & {}^{n}y'' \\ 1 & B_z & 1 \end{vmatrix} = {}^{n}\mathbf{x}' \cdot (\mathbf{b} \times {}^{n}\mathbf{x}'') = \mathbf{0}$$



# Cross Product in Matrix Form

$$\mathbf{b} \times {}^n\mathbf{x}'' = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \times \begin{bmatrix} {}^n x'' \\ {}^n y'' \\ 1 \end{bmatrix}$$

$${}^n\mathbf{x}'' = \begin{bmatrix} b_2 - b_3\,{}^n y'' \\ b_3\,{}^n x'' - b_1 \\ b_1\,{}^n y'' - b_2\,{}^n x'' \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -b_3 & b_2 \\ b_3 & 0 & -b_1 \\ -b_2 & b_1 & 0 \end{bmatrix}}_{S_b} \begin{bmatrix} {}^n x'' \\ {}^n y'' \\ 1 \end{bmatrix}$$

- $S_b$ is skew-symmetric matrix

- Then we have

$${}^n\mathbf{x}' \cdot (\mathbf{b} \times {}^n\mathbf{x}'') = {}^n\mathbf{x}'^{\mathrm{T}} S_b\, {}^n\mathbf{x}'' = \mathbf{0}$$

# What is Skew Symmetric Matrix?

- Let $\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}, \mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$

$$S_b = \begin{bmatrix} 0 & -b_3 & b_2 \\ b_3 & 0 & -b_1 \\ -b_2 & b_1 & 0 \end{bmatrix}$$

- $\mathbf{b} \times \boldsymbol{c} = \begin{vmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{vmatrix} = \begin{bmatrix} b_2 c_3 - b_3 c_2 \\ b_3 c_1 - b_1 c_3 \\ b_1 c_2 - b_2 c_1 \end{bmatrix}$

$$S_b \times \boldsymbol{c} = \begin{bmatrix} 0 & -b_3 & b_2 \\ b_3 & 0 & -b_1 \\ -b_2 & b_1 & 0 \end{bmatrix} \times \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} -b_3 c_2 + b_2 c_3 \\ b_3 c_1 - b_1 c_3 \\ -b_2 c_1 + b_1 c_2 \end{bmatrix}$$

$S_b$ is also written as $[b]_\times$

# Coplanarity Constraint and Fundamental Matrix

- Putting

$$^{n}\mathbf{x}' = (R')^{T}(\mathrm{K}')^{-1}\mathbf{x}'$$
$$^{n}\mathbf{x}'' = (R'')^{T}(\mathrm{K}'')^{-1}\mathbf{x}''$$

into

$$^{n}\mathbf{x}'^{\mathrm{T}} S_b \, {}^{n}\mathbf{x}' = \mathbf{0}$$

$$\mathbf{x}'^{\,T} \underbrace{(\mathrm{K}')^{-T} R' \, S_b \, (R'')^{T} (\mathrm{K}'')^{-1}}_{F} \mathbf{x}'' = 0$$

$$\boxed{\mathbf{x}'^{\,T} \mathrm{F}\, \mathbf{x}'' = 0}$$

F is fundamental matrix

# Today’s Agenda

- Number of Parameters
- Coplanarity Constraints for Image of Uncalibrated Cameras
- Fundamental Matrix
- Epi-polar Geometry
- Estimation of Fundamental Matrix by 8 Point Theorem
- The Essential Matrix
- Estimation of Essential Matrix by 8 Point Theorem

# Fundamental Matrix

F is the *fundamental matrix* of the *relative orientation* of a pair of images of *uncalibrated cameras*

$$F=(\mathrm{K}')^{-T} R' \; S_b \; (R'')^T (\mathrm{K}'')^{-1}$$

Which fulfills the equation

$$\mathbf{x}'^{\,T} \mathrm{F} \, \mathbf{x}'' = 0$$

- Remarks:
  - Fundamental matrix F 3x3 matrix with 7 DoF.
  - F is homogenous (AF=0)
  - F is singular det(F)=0

# Fundamental Matrix from Projection Matrices

- If the ***projection matrices*** are given for the two cameras, we can derive the ***fundamental matrix.***

- Let the projection matrix be partitioned into 3x3 matrix $(\mathrm{A}')$ and 3x1 vector $(\mathbf{a}')$, then

$$\mathrm{P}' = \mathrm{K}'R'[I_3|-Z'] = [\mathrm{K}'R'| - \mathrm{K}'R'Z'] = [\mathrm{A}'|\mathbf{a}']$$

where

$$\mathrm{A}' = \mathrm{K}'R' \; \& \qquad \mathbf{a}' = -\mathrm{K}'R'Z'$$

- The fundamental matrix is given by

$$\mathrm{F} = \mathrm{A}'^{-\mathrm{T}} \mathrm{S}_{b_{12}} \mathrm{A}''^{-1} \quad \text{with} \qquad \mathrm{b}_{12} = \mathrm{A}''^{-1}\,\mathbf{a}'' - \mathrm{A}'^{-1}\,\mathbf{a}'$$

# Proof

- $\mathrm{F} = \mathrm{A}'^{-\mathrm{T}} \mathrm{S}_{b_{12}} \mathrm{A}''^{-1}$ with $\mathrm{A}' = \mathrm{K}' R'$ & $\mathbf{a}' = -\mathrm{K}' R' Z'$

$$\mathrm{A}'^{-\mathrm{T}} = (\mathrm{K}' R')^{-\mathrm{T}} = (R'^{-1} \mathrm{K}'^{-1})^{\mathrm{T}} = \mathrm{K}'^{-\mathrm{T}} R'^{-\mathrm{T}} = \mathrm{K}'^{-\mathrm{T}} R'$$

$$\mathrm{A}''^{-1} = (\mathrm{K}'' R'')^{-1} = R''^{-1} \mathrm{K}''^{-1}$$

Therefore Fundamental matrix can be computed directly from projection matrices

$$\mathrm{F} = (\mathrm{K}')^{-T} R' \, S_b \, (R'')^{T} (\mathrm{K}'')^{-1} = \mathrm{A}'^{-\mathrm{T}} \mathrm{S}_{b_{12}} \mathrm{A}''^{-1}$$

# Fundamental Matrix from Projection Matrices

*From the projection matrix, we have*

$$[\mathrm{K}'R'| - \mathrm{K}'R'Z']=[\mathrm{A}'|\mathbf{a}']$$

Then

$$\mathrm{A}' = \mathrm{K}'R'$$

$$\mathrm{K}'R'Z' = -\,\mathbf{a}'$$

$$Z' = -(\mathrm{A}')^{-1}\,\mathbf{a}'$$

- $\mathrm{b}_{12} = \mathrm{Z}'' - \mathrm{Z}' = (\mathrm{A}'')^{-1}\,\mathbf{a}'' - (\mathrm{A}')^{-1}\,\mathbf{a}'$

- $\mathrm{b}_{12} = \mathrm{A}''^{-1}\,\mathbf{a}'' - \mathrm{A}'^{-1}\,\mathbf{a}'$

# Degree of Freedom

- Fundamental Matrix F=$(\mathrm{K}')^{-T} R'\ S_b\ (R'')^T (\mathrm{K}'')^{-1}$ has seven degrees of freedom. This is because F is homogeneous and singular. As the skew symmetric matrix $S_b$ is singular with rank two.
- Therefore Rank(F) = min rank$(\mathrm{K}', R'\ , S_b\ , R'', \mathrm{K}'') = 2$

- Fundamental matrix is of the form
  - $\mathrm{F} = U\mathrm{Diag}(S_1, S_2, 0)V^T$ with $S_1$ & $S_2 > 0$

- Where $U$ and $V$ are orthogonal matrices

# Degree of Freedom

- Fundamental Matrix $F = (K')^{-T} R' \, S_b \, (R'')^T (K'')^{-1}$ has seven degrees of freedom. This is because F is homogeneous and singular. As the skew symmetric matrix $S_b$ is singular with rank two.
- Therefore Rank(F) = min rank$(K', R' \, , S_b \, , R'', K'') = 2$

- Fundamental matrix is of the form
  - $\mathrm{F} = U\mathrm{Diag}(S_1, S_2, 0)V^T$ with $S_1$ & $S_2 > 0$

- Where $U$ and $V$ are orthogonal matrices

# Summary of Fundamental Matrix

- Fundamental Matrix F=$(\mathrm{K}')^{-T} R' \, S_b \, (R'')^T (\mathrm{K}'')^{-1}$
- F is Singular (det(F)=0)
- F consist of the *Relative Orientation* of image pair from uncalibrated cameras
- F has 7 DoF
- Given the projection matrices P’ and P’’ of the two cameras, F can be computed directly

$$\mathrm{F} = \mathrm{A}'^{-\mathrm{T}} \mathrm{S}_{b_{12}} \mathrm{A}''^{-1}$$

- Coplanarity Constraint

$$\mathbf{x}'^{\,T} \mathrm{F} \, \mathbf{x}'' = 0$$

# Equation to Remember

## Fundamental Matrix F

$$F=(K')^{-T} R' \, S_b \, (R'')^T (K'')^{-1}$$

Movie

# Today's Agenda

- Number of Parameters
- Coplanarity Constraints for Image of Uncalibrated Cameras
- Fundamental Matrix
- Epi-polar Geometry
- Estimation of Fundamental Matrix by 8 Point Theorem
- The Essential Matrix
- Estimation of Essential Matrix by 8 Point Theorem

# Epipolar Geometry

# Epipolar Geometry Motivation

- Given: A point $x'$ in the first image
- Aim: predicting the position of a point $x''$ in the second image
- Epipolar geometry defines the geometry relations between the image pair which reduces the search space for finding corresponding points in image images.
- Given a straight-line preserving properties, the search space is reduced from the whole image (*2D*) to a straight line (*1D*)

# Epipolar Geometry

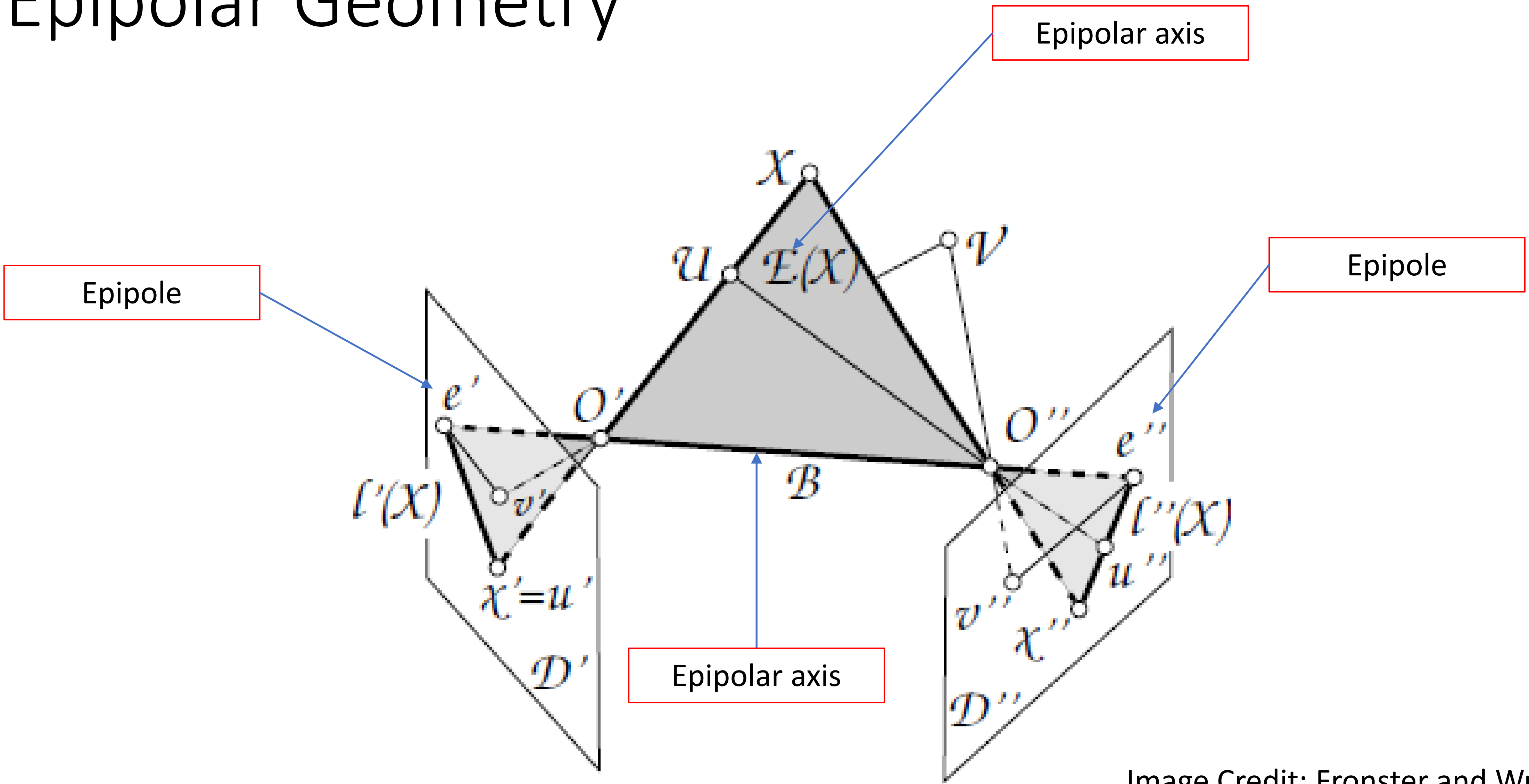



Image Credit: Fronster and Wrobel, "Photogrammetric Computer Vision"

# Definition

- The *epipolar axis* - The line through two projection centres $O', O''$

$$\mathcal{B} = O' \wedge O''$$

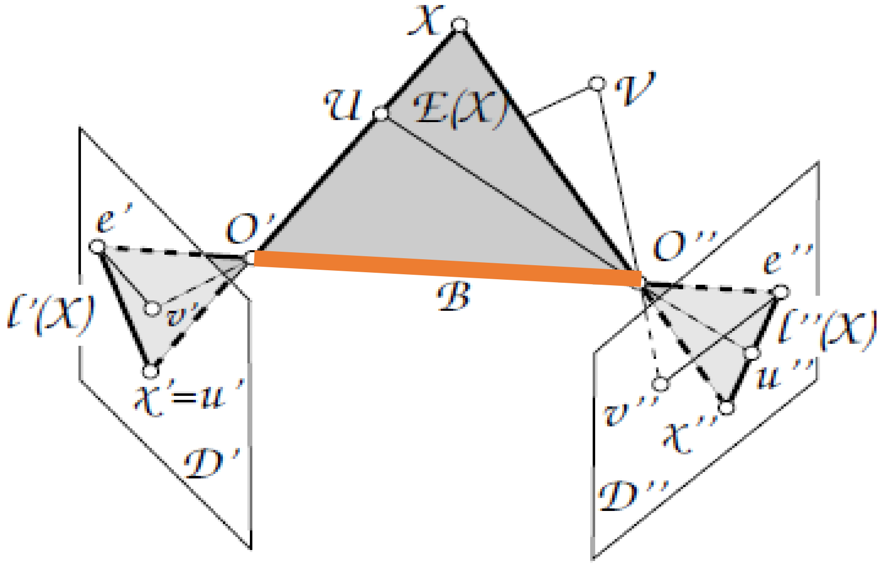



Image Credit: Fronster and Wrobel, "Photogrammetric Computer Vision"

# Definition

- The *epipolar plane* – The plane formed by two projection centres $O'$ , $O''$ and the object point $\mathcal{X}$

$$\mathcal{E}(\mathcal{X}) = O' \wedge O'' \wedge \mathcal{X}$$

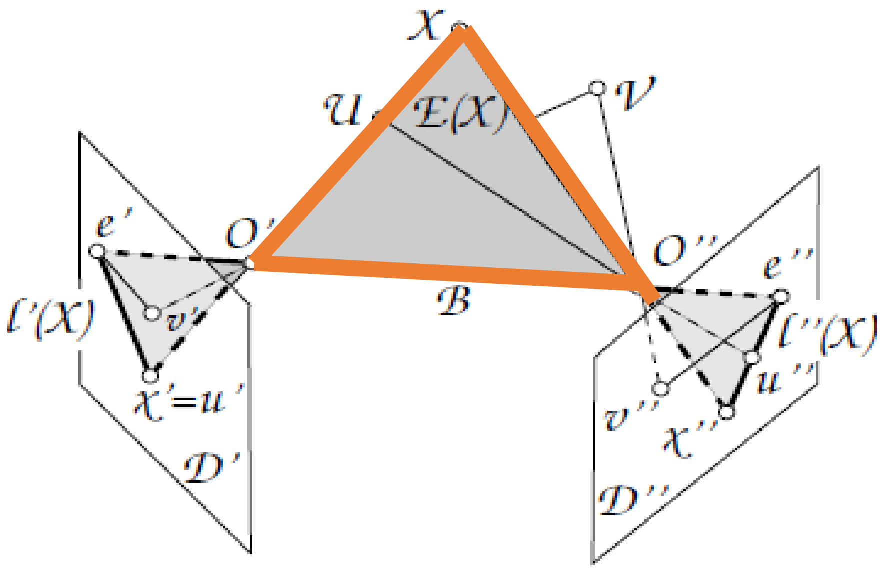



Image Credit: Fronster and Wrobel, "Photogrammetric Computer Vision"

# Definition

- The *epipoles* – The image of other projection centres using projection $\mathcal{P}'$ and $\mathcal{P}''$

$$e' = \mathcal{P}'(O'') \;\&\; e'' = \mathcal{P}''(O')$$

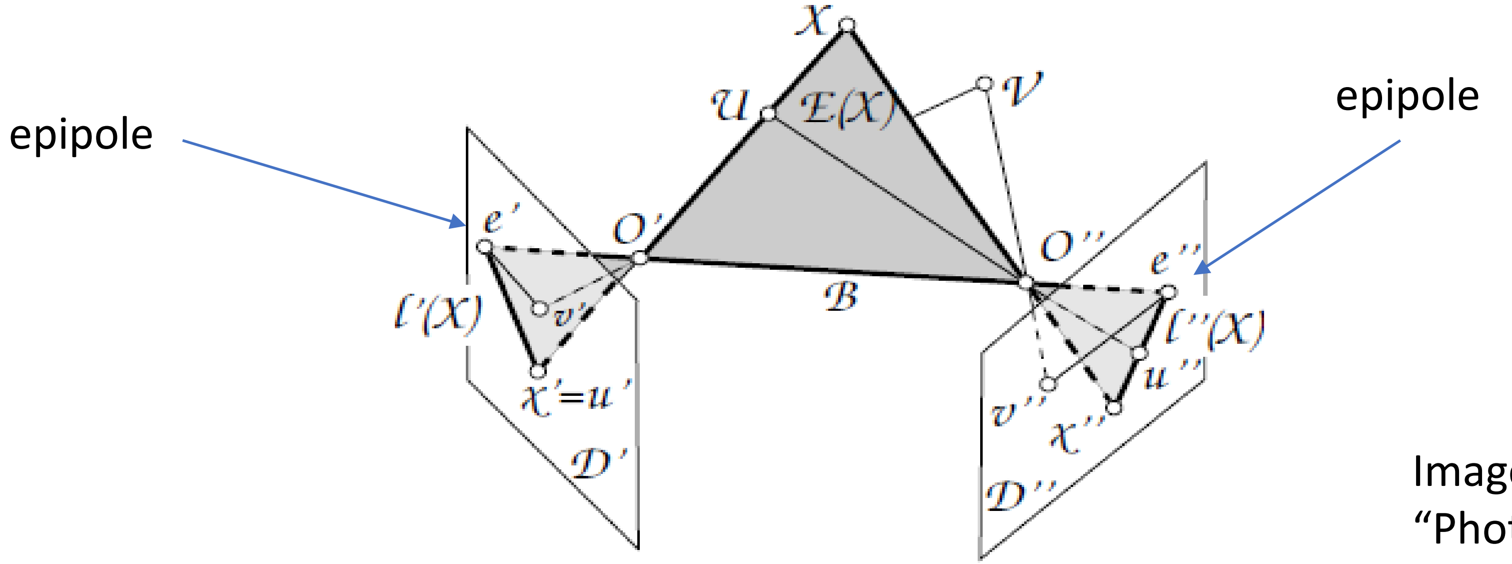



Image Credit: Fronster and Wrobel, "Photogrammetric Computer Vision"

# Definition

- The *epipolar lines* – Projection of the line between point $\mathcal{X}$ and projection centre $O$ onto *THE OTHER* image

$$\ell'(\mathcal{X})=\mathcal{P}'(O'' \wedge \mathcal{X}) \ \& \ \ell'(\mathcal{X})=\mathcal{P}'(O'' \wedge \mathcal{X})$$

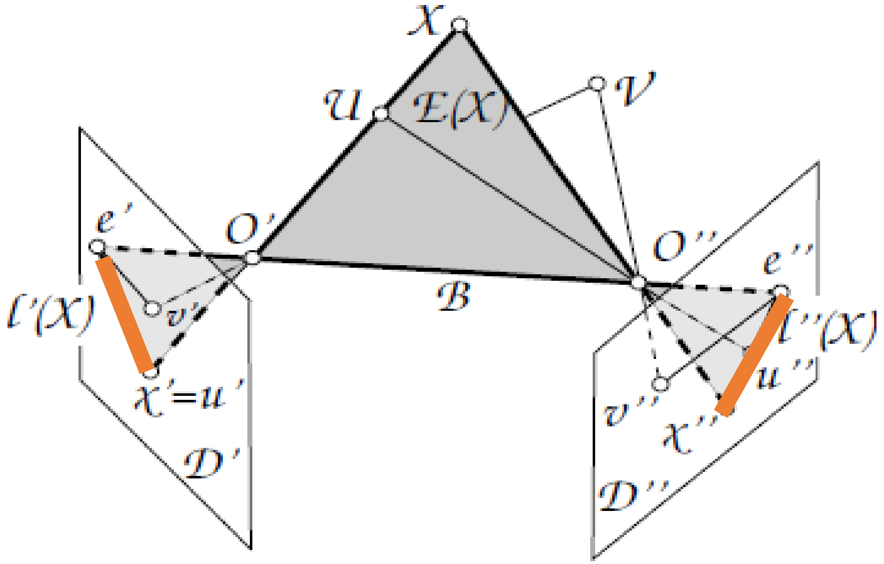
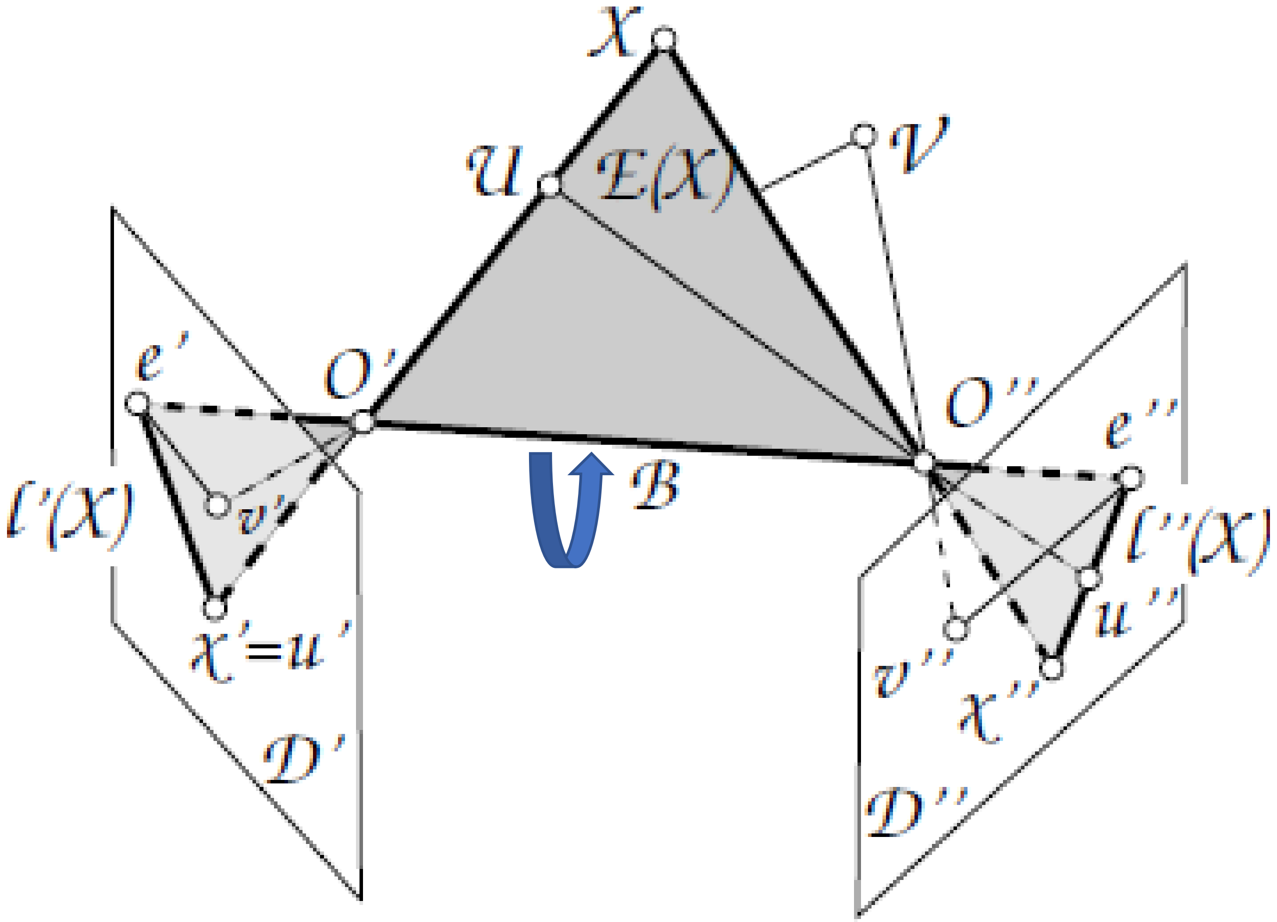



Image Credit: Fronster and Wrobel, “Photogrammetric Computer Vision”

# Multiple Epipolar planes on epipolar line



# More on Epipole and Epipolar line

- The *epipoles* $e' = \mathcal{P}'(O'')$ , $e'' = \mathcal{P}''(O')$ are the image of other projection centres. For perspective cameras, they can also be represented by

$$e' = (O' \wedge O'') \cap \mathcal{D}' \qquad \& \qquad e'' = (O' \wedge O'') \cap \mathcal{D}''$$

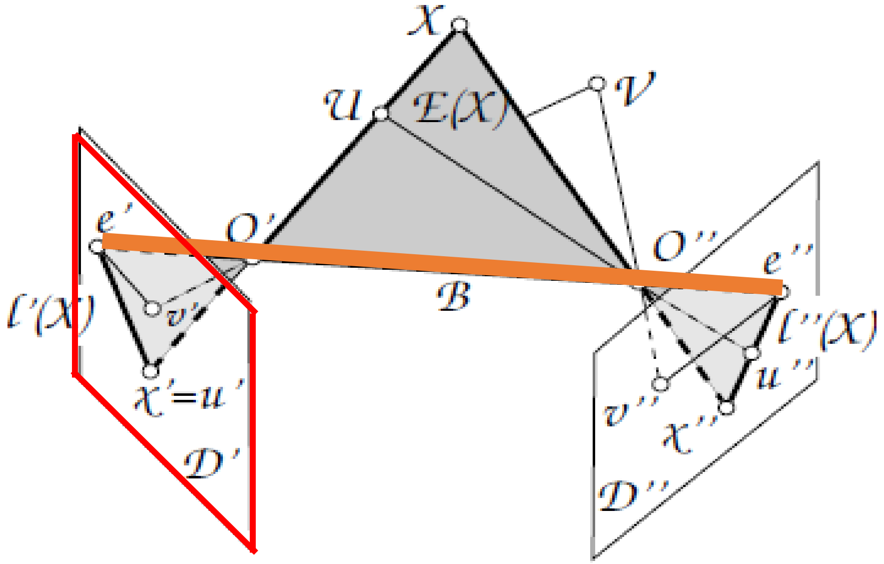



Image Credit: Fronster and Wrobel, "Photogrammetric Computer Vision"

# More on Epipole and Epipolar line

- The *epipolar line* $\mathcal{l}'(\mathcal{X}) = \mathcal{P}'(\mathcal{O}'' \wedge \mathcal{X})$, $\mathcal{l}''(\mathcal{X}) = \mathcal{P}''(\mathcal{O}' \wedge \mathcal{X})$ can be written as the intersections of epipolar plane with the image plane below:

$$\mathcal{l}'(\mathcal{X}) = \mathcal{E}(\mathcal{X}) \cap \mathcal{D}' \qquad \& \qquad \mathcal{l}''(\mathcal{X}) = \mathcal{E}(\mathcal{X}) \cap \mathcal{D}''$$

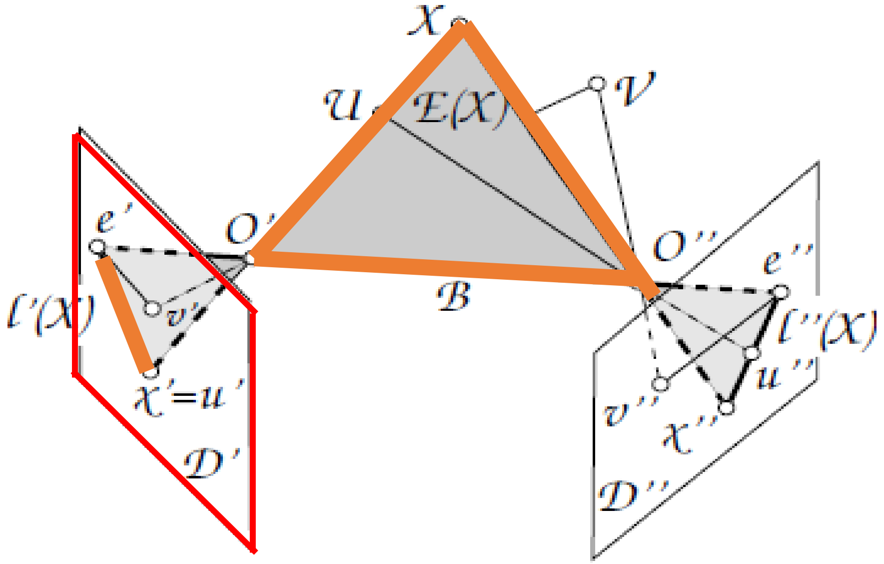
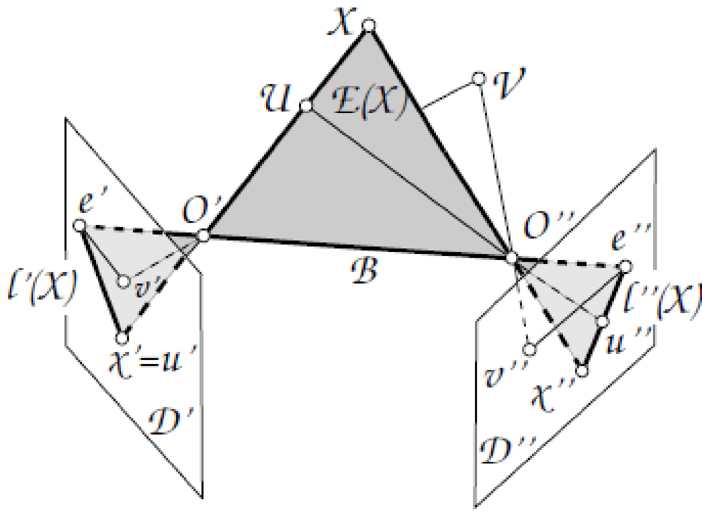



Image Credit: Fronster and Wrobel, "Photogrammetric Computer Vision"

# Epipolar Plane

- The following lies on the same plane epipolar plane $\mathcal{E}$
  - Projection Centres $O', O''$
  - Object point $\mathcal{X}$
  - Epipolar lines $l'(\mathcal{X}), l''(\mathcal{X})$
  - The image points $\mathcal{X}', \mathcal{X}''$




- This properties ***reduces the dimension*** for corresponding points searching between images

Image Credit: Fronster and Wrobel, "Photogrammetric Computer Vision"

# Searching for the correspondence points

- Given point $\mathcal{X}'$ one image one, search for the correspondence $\mathcal{X}''$ in the other image

- This can be easily solved
- ***Epipolar plane*** : $\mathcal{E}(\mathcal{X}')=(O',O'',\mathcal{X}')$
- ***Epipolar line*** *:* Intersection of $\mathcal{E}(\mathcal{X}')$ and image plane $\mathcal{D}''$ gives $l''(\mathcal{X}')$
- Corresponding point $\mathcal{X}''$ must lies on $l''(\mathcal{X}')$

- Searching can be confined to the ***epipolar line***

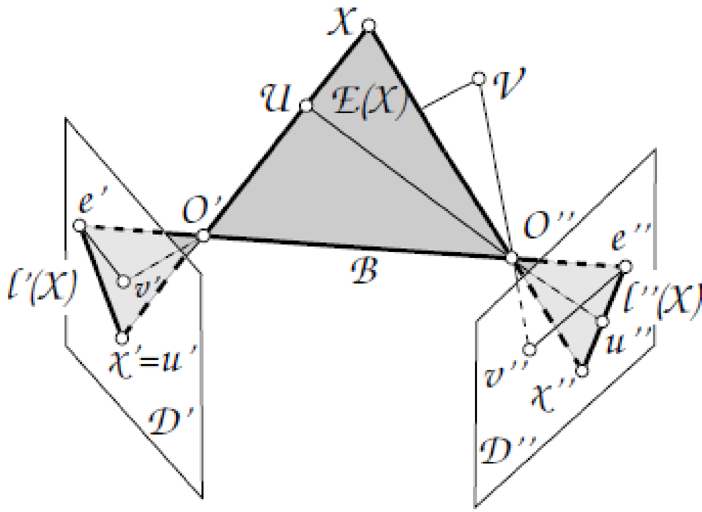


Image Credit: Fronster and Wrobel, "Photogrammetric Computer Vision"

# Computing epipolar axis

- Given: The projection matrices or Fundamental Matrix, we can compute the entities

- The ***epipolar axis*** has the direction of

$$\mathbf{b} = B = O'' - O'$$

**b** is the direction of the vector B between the two projection centres

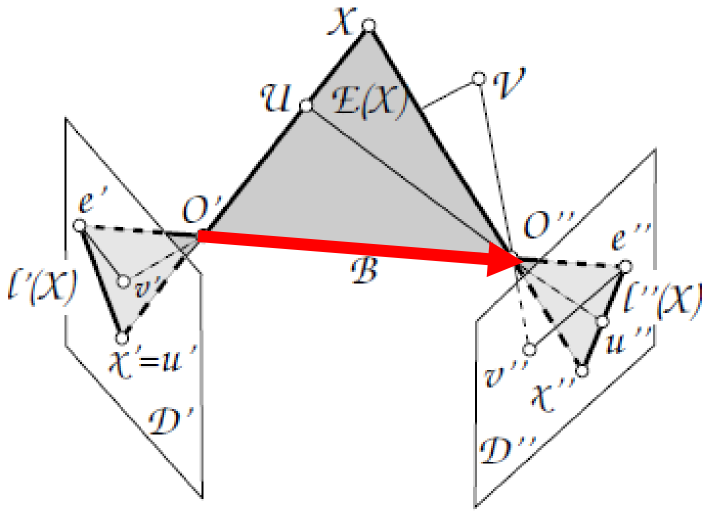



Image Credit: Fronster and Wrobel, “Photogrammetric Computer Vision”

# Computing epipolar lines

- ***Epipolar lines*** are the projections of the projected lines $\mathcal{L}_{x'}$ and $\mathcal{L}_{x''}$ into other image
- For point $\mathcal{X}$ , the project $\mathcal{X}'$ must lies on $l'(\mathcal{X})$, we have
  - $\mathbf{x}'^T \cdot \mathrm{l}' = 0$
- By coplanarity Constraint, we have

  $\mathbf{x}'^T \mathsf{F}\, \mathbf{x}'' = 0$
- Hence, we have

  $\mathrm{l}' = \mathsf{F}\, \mathbf{x}''$
- A *point* on one image corresponds to an *epipolar line* on other image

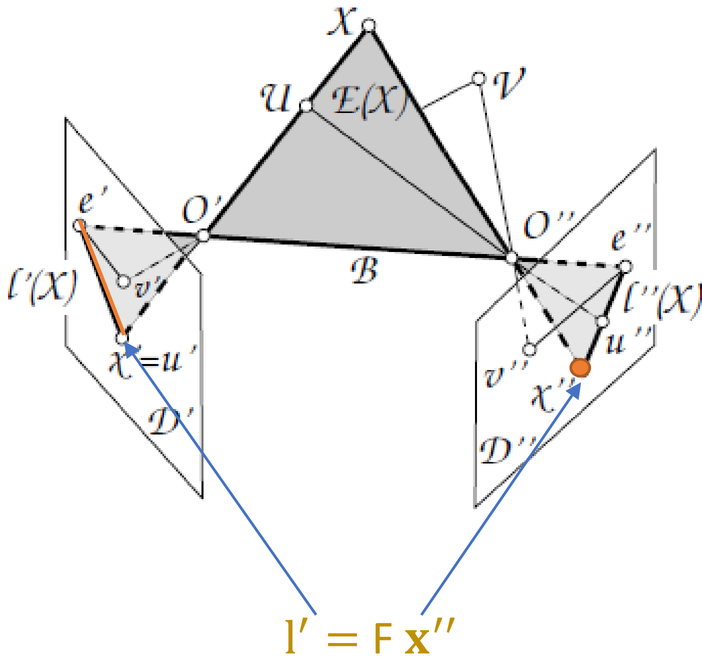



Image Credit: Fronster and Wrobel, "Photogrammetric Computer Vision"

# Similarly

- For point $\mathcal{X}$ , the project $\mathcal{X}''$ must lies on $\mathcal{l}''(\mathcal{X})$, we have
  - $\mathbf{x}''^{T} \cdot \mathrm{l}'' = 0$
- By Coplanarity Constraint

$$\mathbf{x}'^{\,T} \mathsf{F}\, \mathbf{x}'' = 0 \quad \text{or}$$

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

We again have

$$\mathrm{l}'' = \mathsf{F}^{T} \mathbf{x}'$$

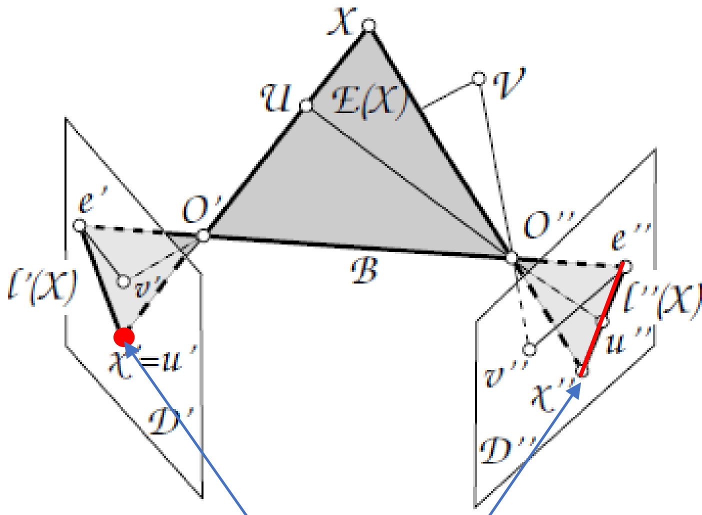



Image Credit: Fronster and Wrobel, “Photogrammetric Computer Vision”

# Example – Epipolar Line

- Given: Two stereo image with known correspondence.
- Aim: to find the ***epipolar lines*** and ***epipoles***.

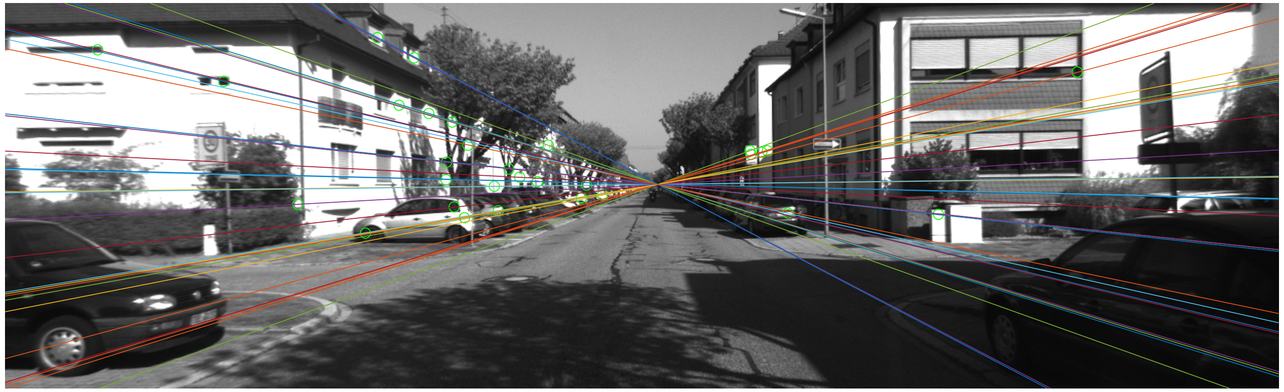
I1 (left); I2 (right)

# Correspondence Points

- Given: Correspondence Points in Image 1 and Image 2
- $\mathrm{l}'' = \mathrm{F}^{\boldsymbol{T}}\mathbf{x}'$

Correspondence Points in Image 1

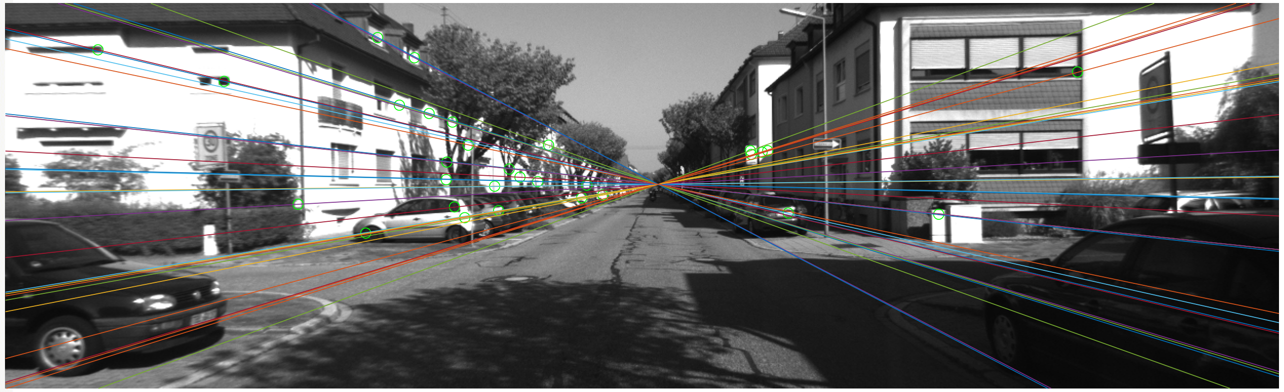
Image 2 Epipolar Lines correspond to $\mathbf{x}'$



# MATLAB Codes

```
figure; ax=axes;
imshow(I2);

%showMatchedFeatures(I1, I2, inlierPoints1a, inlierPoints2a);
%legend('Inlier points in I1', 'Inlier points in I2');
hold on;
%plot(inlierPoints2a(:,1),inlierPoints2a(:,2),'go')

%drawing the epipole
epipoleHC1=null(fMatrix2);
plot(epipoleHC1(1)/epipoleHC1(3),epipoleHC1(2)/epipoleHC1(3),'go');


%Calculate by formula l''=F'X'
inlierPoints1aHC=cart2hom(inlierPoints1a);
lss = zeros(30,3);
for i=1:30
    point = inlierPoints1aHC(i,:);
    lss(i,:)=fMatrix2'*point';  %the epipolar lines in image2
end

points = lineToBorderPoints(lss, size(I2));
line(points(:, [1,3])', points(:, [2,4])');
truesize;
hold off;
```

Epipolar Lines correspond to $\mathbf{x}'$



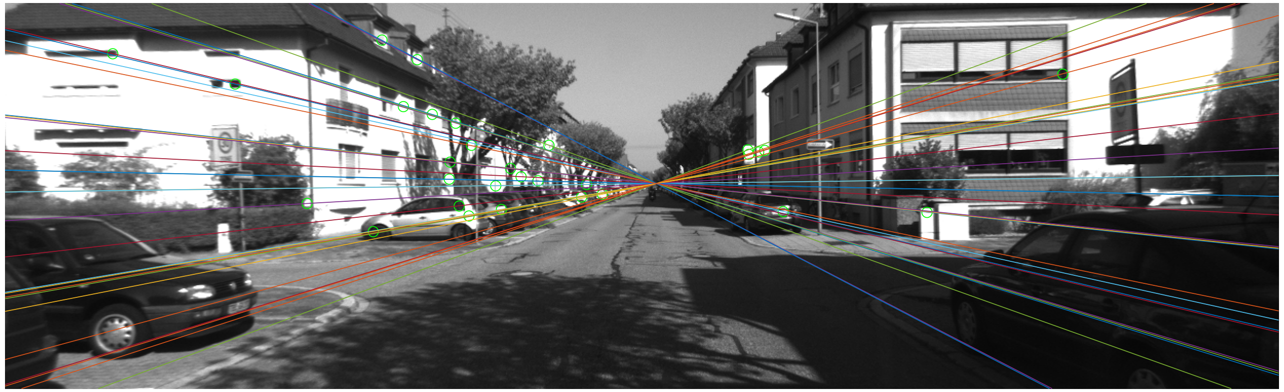
# Similarly for Image 2

- Similarly for Image 1
- Epipolar lines in Image 1 are

$$\mathbf{l}' = \mathbf{F}\mathbf{x}''$$

Correspondence Points in Image 2

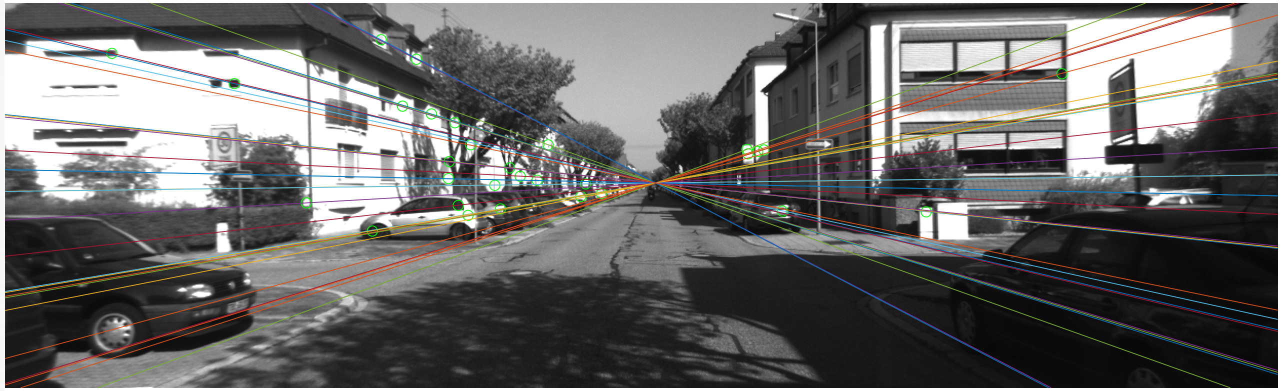
Image 1: Epipolar Lines correspond to $\mathbf{x}''$



# MATLAB Codes

```
figure; ax=axes;
imshow(I1);
hold on;
%drawing the corresponding points
%plot(inlierPoints1a(:,1),inlierPoints1a(:,2),'go');

%drawing the epipole
epipoleHC1=null(fMatrix2');
plot(epipoleHC1(1)/epipoleHC1(3),epipoleHC1(2)/epipoleHC1(3),'go');


% Compute the Epipolar Lines from corresondence point on second image
inlierPoints2aHC=cart2hom(inlierPoints2a);
ls = zeros(30,3);
for i=1:30
    point = inlierPoints2aHC(i,:);
    ls(i,:)=fMatrix2*point'; % The epipolar line in image 1
end

%drawing the lines
points = lineToBorderPoints(ls, size(I2));
line(points(:, [1,3])', points(:, [2,4])');
truesize;
hold off;

figure; ax=axes;
imshow(I2);
```

Epipolar Lines correspond to $\mathbf{x}''$

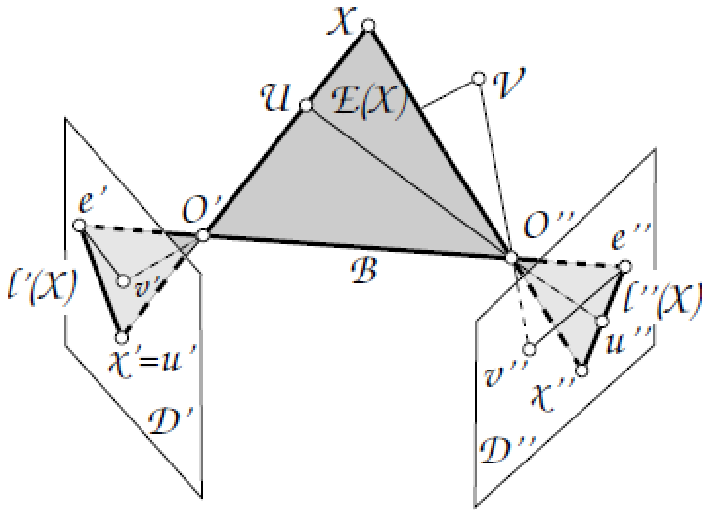


# Computing epipoles

- The *epipoles* are the image of *projection centres* to another image.

- This can be direction computed by *projection matrices* and centres

$$e' = P'O'' \qquad \text{and} \qquad e'' = P''O'$$



# Computing epipoles

- Also *epipole* must lies on the *epipolar line*
  - $\mathrm{e}'^{\mathrm{T}} \cdot \mathrm{l}' = 0$ *and* $\mathrm{e}''^{\mathrm{T}} \cdot \mathrm{l}'' = 0$
- As the *epipolar line* is defined through the *fundamental matrix* F, i.e. $\mathrm{l}' = \mathsf{F}\,\mathbf{x}''$ and therefore for all point $\mathbf{x}''$
  - $\mathrm{e}'^{\mathrm{T}} \mathsf{F}\,\mathbf{x}'' = 0$
- Since this must be true for every $\mathbf{x}''$, $\mathrm{e}'^{\mathrm{T}}\mathsf{F}=0$
- The epipole $\mathrm{e}'$ is the *null space* of $\mathrm{F}^{\mathrm{T}}$

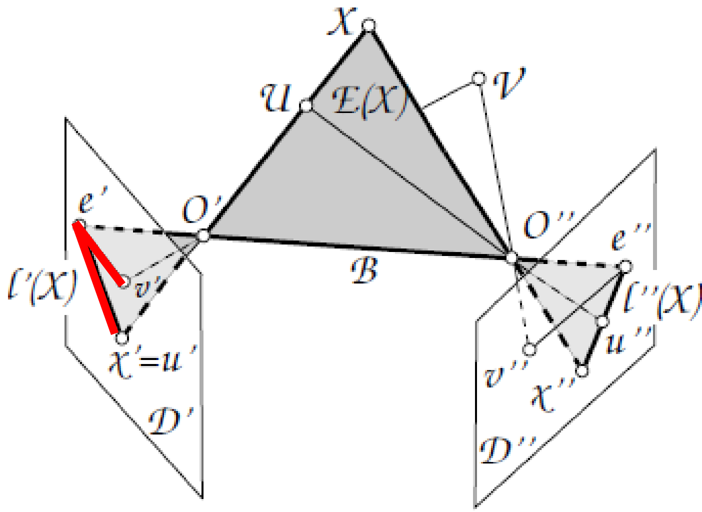



Image Credit: Fronster and Wrobel, "Photogrammetric Computer Vision"

# Computing Epipoles

- Similarly we have the epipole of the second image

$$\mathrm{x}'^{\mathrm{T}}\mathsf{F}\,\mathbf{e}'' = 0$$

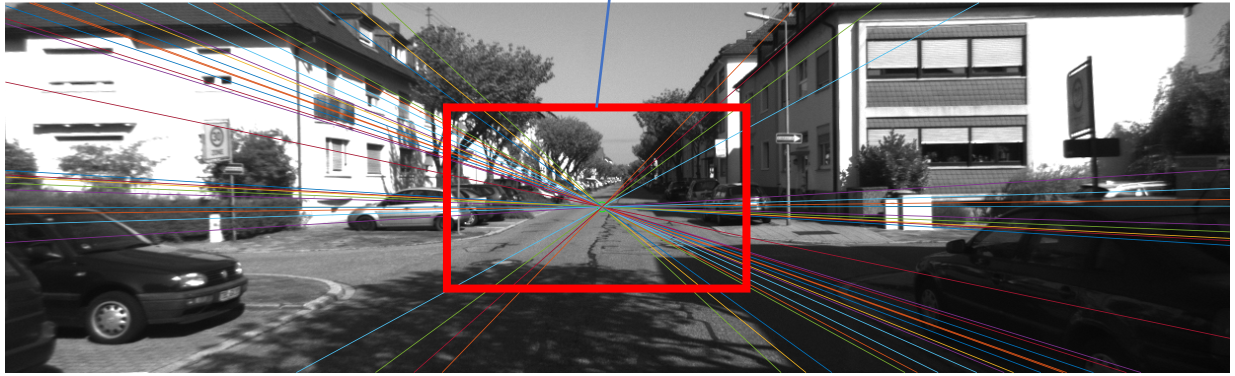
- Hence

$$\mathrm{e}' = \mathrm{null}(\mathrm{F}^{\mathrm{T}}) \qquad\qquad \mathrm{e}'' = \mathrm{null}(\mathrm{F})$$

# Example Epipole (Image 1)

```
figure; ax=axes;
imshow(I1);
hold on;
%drawing the corresponding points
%plot(inlierPoints1a(:,1),inlierPoints1a(:,2),'go');

%drawing the epipole
epipoleHC1=null(fMatrix2');
plot(epipoleHC1(1)/epipoleHC1(3),epipoleHC1(2)/epipoleHC1(3),'go');


% Compute the Epipolar Lines from corresondence point on second image
inlierPoints2aHC=cart2hom(inlierPoints2a);
ls = zeros(30,3);
for i=1:30
    point = inlierPoints2aHC(i,:);
    ls(i,:)=fMatrix2*point'; % The epipolar line in image 1
end

%drawing the lines
points = lineToBorderPoints(ls, size(I2));
line(points(:, [1,3])', points(:, [2,4])');
truesize;
hold off;

figure; ax=axes;
imshow(I2);
```

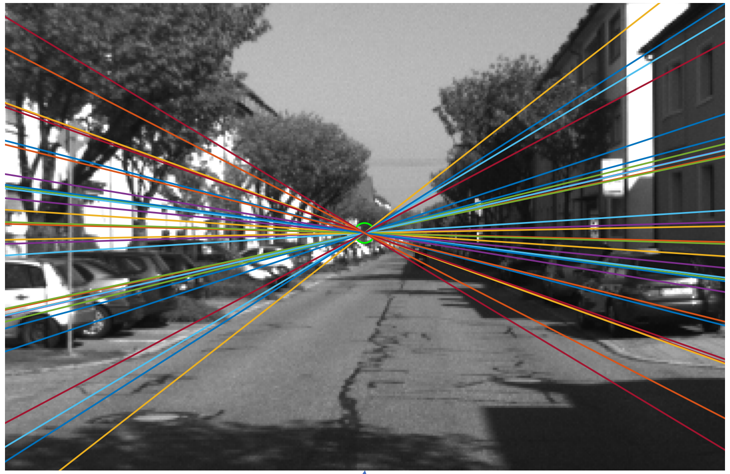
$$e' = \mathrm{null}(F^T)$$

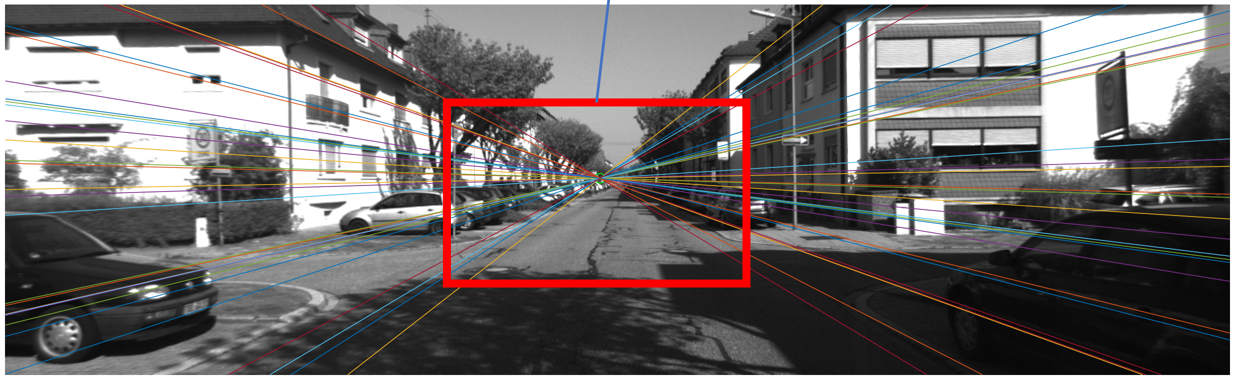


# Similarly for Image 2

```
figure; ax=axes;
imshow(I2);

%showMatchedFeatures(I1, I2, inlierPoints1a, inlierPoints2a);
%legend('Inlier points in I1', 'Inlier points in I2');
hold on;
%plot(inlierPoints2a(:,1),inlierPoints2a(:,2),'go')

%drawing the epipole
epipoleHC1=null(fMatrix2);
plot(epipoleHC1(1)/epipoleHC1(3),epipoleHC1(2)/epipoleHC1(3),'go');


%Calculate by formula l''=F'X'
inlierPoints1aHC=cart2hom(inlierPoints1a);
lss = zeros(30,3);
for i=1:30
    point = inlierPoints1aHC(i,:);
    lss(i,:)=fMatrix2'*point';  %the epipolar lines in image2
end

points = lineToBorderPoints(lss, size(I2));
line(points(:, [1,3])', points(:, [2,4])');
truesize;
hold off;
```

$$e'' = \mathrm{null}(F)$$

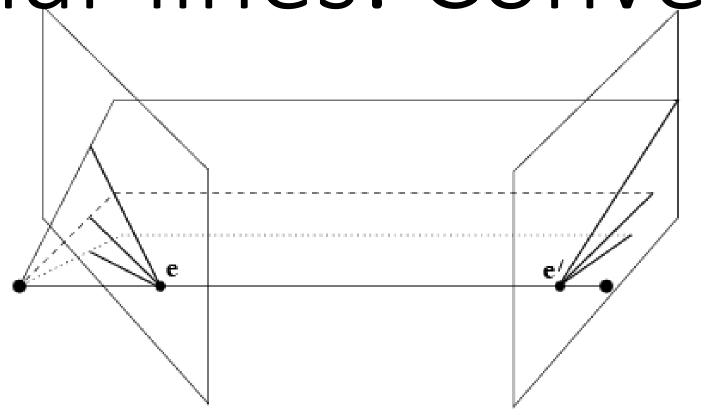




# Summary of Epipolar Geometry

- We assume *straight-line preserving* (*uncalibrated*) camera
- We defined the key elements of *epipolar geometry*
- Search for the corresponding points from one image on another is confined to *1D* case.

# More on Epipolar lines: Converging cameras



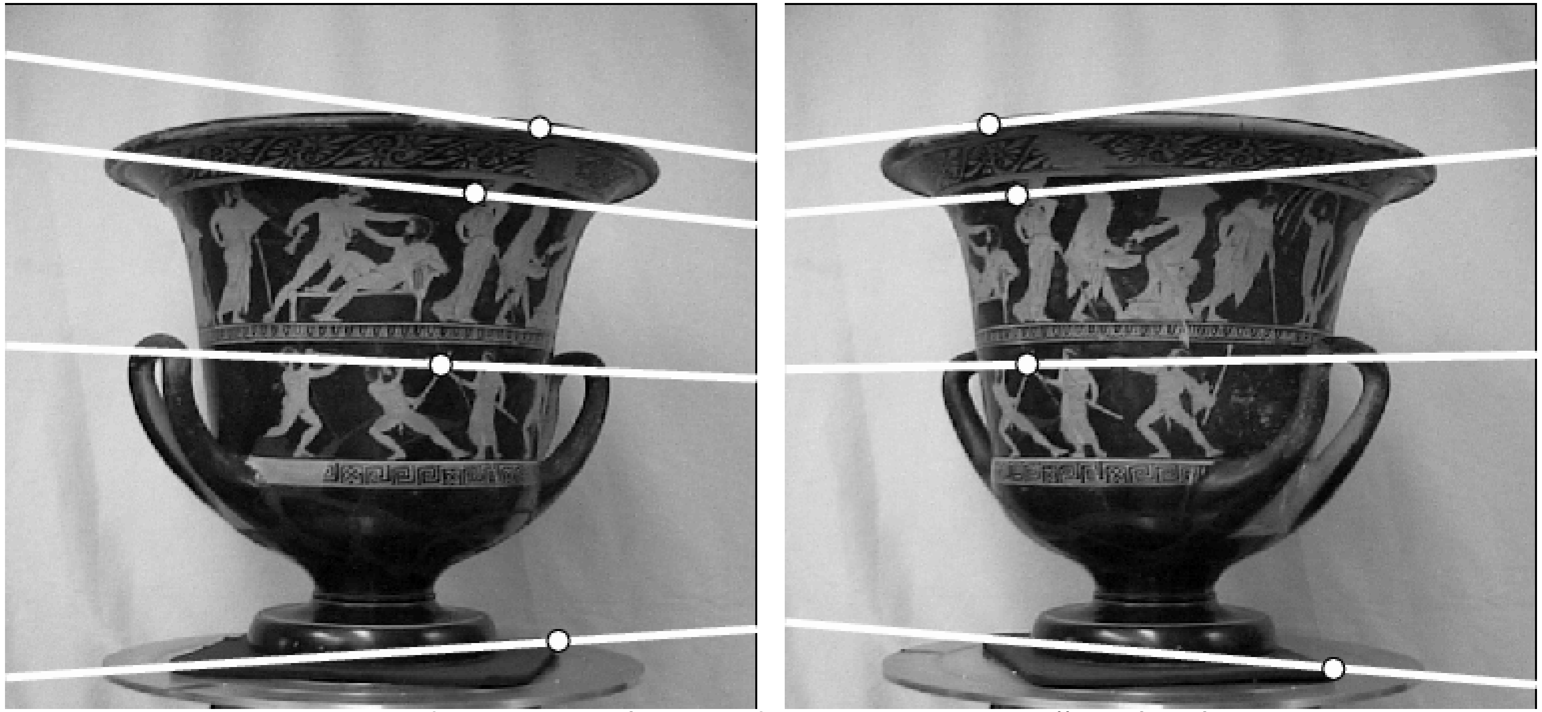
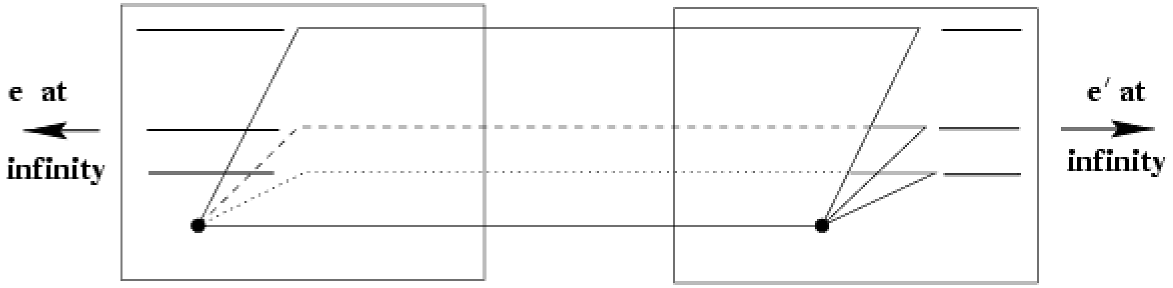



Image Credit: R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”

# More on Epipolar lines : Motion parallel to image plane



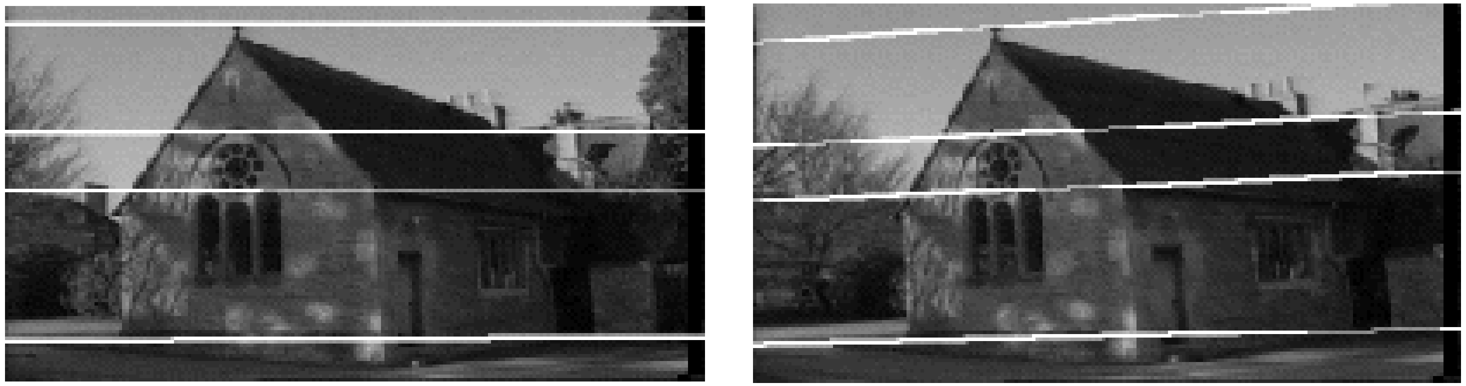
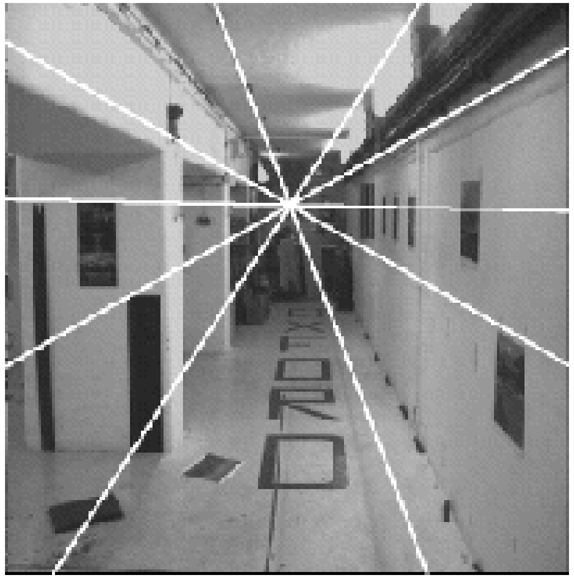



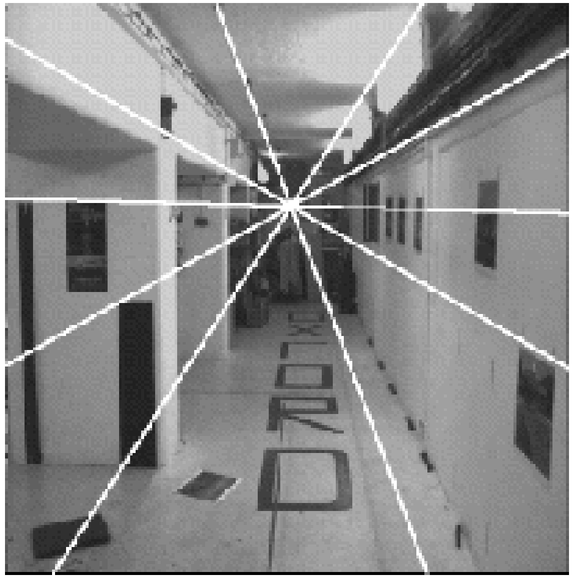
Image Credit: R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”

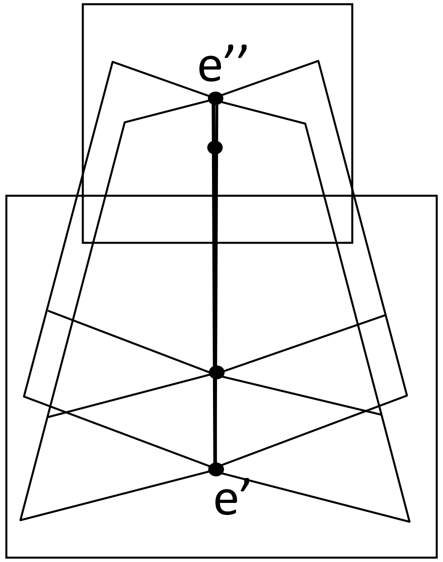
# Example: Forward motion

What would the epipolar lines look like if the camera moves directly forward?

# Example: Forward motion








Epipole has same coordinates in both images.
Points move along lines radiating from e:
“Focus of expansion”

Image Credit: R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”

# Today's Agenda

- Number of Parameters
- Coplanarity Constraints for Image of Uncalibrated Cameras
- Fundamental Matrix
- Epi-polar Geometry
- Estimation of Fundamental Matrix by 8 Point Theorem
- The Essential Matrix
- Estimation of Essential Matrix by 8 Point Theorem

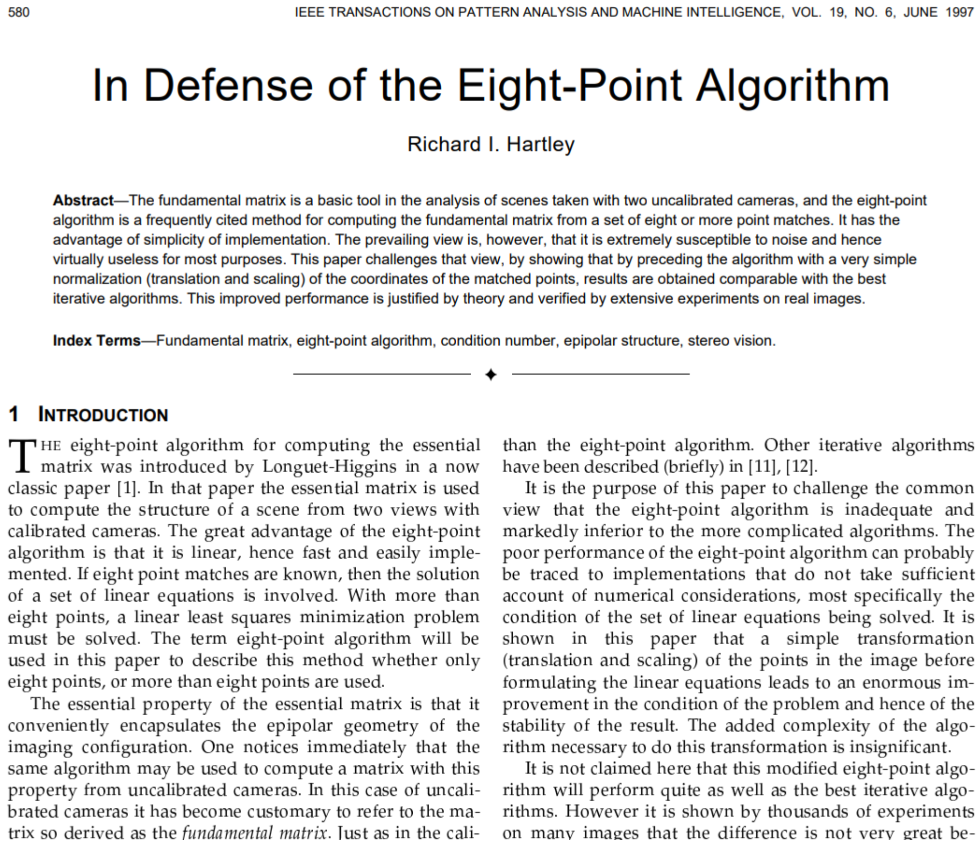
# Solving F from correspondent points

Given: *N* corresponding points from the image pair

$$(x_i', y_i') \text{ and } (x_i'', y_i'') \text{ for } i = 1 \ldots N$$

- Aim: Compute the *fundamental matrix* F based on the coplanarity Constraints for *uncalibrated*, *straight line-preserving camera* is

$$\mathbf{x}'^{T} \mathrm{F}\, \mathbf{x}'' = 0$$





## In Defense of the Eight-Point Algorithm

Richard I. Hartley

**Abstract**—The fundamental matrix is a basic tool in the analysis of scenes taken with two uncalibrated cameras, and the eight-point algorithm is a frequently cited method for computing the fundamental matrix from a set of eight or more point matches. It has the advantage of simplicity of implementation. The prevailing view is, however, that it is extremely susceptible to noise and hence virtually useless for most purposes. This paper challenges that view, by showing that by preceding the algorithm with a very simple normalization (translation and scaling) of the coordinates of the matched points, results are obtained comparable with the best iterative algorithms. This improved performance is justified by theory and verified by extensive experiments on real images.

**Index Terms**—Fundamental matrix, eight-point algorithm, condition number, epipolar structure, stereo vision.

### 1 Introduction

THE eight-point algorithm for computing the essential matrix was introduced by Longuet-Higgins in a now classic paper [1]. In that paper the essential matrix is used to compute the structure of a scene from two views with calibrated cameras. The great advantage of the eight-point algorithm is that it is linear, hence fast and easily implemented. If eight point matches are known, then the solution of a set of linear equations is involved. With more than eight points, a linear least squares minimization problem must be solved. The term eight-point algorithm will be used in this paper to describe this method whether only eight points, or more than eight points are used.

The essential property of the essential matrix is that it conveniently encapsulates the epipolar geometry of the imaging configuration. One notices immediately that the same algorithm may be used to compute a matrix with this property from uncalibrated cameras. In this case of uncalibrated cameras it has become customary to refer to the matrix so derived as the *fundamental matrix*. Just as in the cali- than the eight-point algorithm. Other iterative algorithms have been described (briefly) in [11], [12].

It is the purpose of this paper to challenge the common view that the eight-point algorithm is inadequate and markedly inferior to the more complicated algorithms. The poor performance of the eight-point algorithm can probably be traced to implementations that do not take sufficient account of numerical considerations, most specifically the condition of the set of linear equations being solved. It is shown in this paper that a simple transformation (translation and scaling) of the points in the image before formulating the linear equations leads to an enormous improvement in the condition of the problem and hence of the stability of the result. The added complexity of the algorithm necessary to do this transformation is insignificant.

It is not claimed here that this modified eight-point algorithm will perform quite as well as the best iterative algorithms. However it is shown by thousands of experiments on many images that the difference is not very great be-

# The Coplanarity Constraint

- For each corresponding point pair, we have

$$\mathbf{x}_i'^T \mathsf{F} \mathbf{x}_i'' = 0 \qquad \text{for } i = 1 \ldots N$$

We have for each point

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

# The Coplanarity Constraint

- For each corresponding pair, we have

$$[\boxed{x_i'}, y_i', 1] \begin{bmatrix} \boxed{F_{11}} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} \boxed{x_i''} \\ y_i'' \\ 1 \end{bmatrix} = 0 \qquad \text{for } i = 1 \dots N$$

- Expanding

$$\boxed{x_i'' F_{11} x_i'} + x_i'' F_{21} y_i' + x_i'' F_{31}$$
$$+ y_i'' F_{12} x_i' + y_i'' F_{22} y_i' + y_i'' F_{32}$$
$$+ F_{13} x_i' + F_{23} y_i' + F_{33} = 0 \qquad \text{for } i = 1 \dots N$$

# Represented using Kronecker Product

- Define

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

$$\mathrm{a}_i^{\mathrm{T}} = \mathrm{x}_i''^T \otimes \mathrm{x}_i'^T = [x_i'' x_i', x_i'' y_i', x_i'', y_i'' x_i', y_i'' y_i', y_i'', x_i', y_i', 1]$$

Therefore the constraints are written as

$\mathrm{a}_i^{\mathrm{T}}\mathrm{f} = 0$ for for $i = 1 \ldots N$

Hence we stack up all points

$$\mathrm{A} = \begin{bmatrix} \mathrm{a}_i^{\mathrm{T}} \\ \vdots \\ \mathrm{a}_n^{\mathrm{T}} \end{bmatrix} \Rightarrow \mathrm{A}\mathbf{f}=0$$

Kronecker product - Wikipedia

# How many points do we need?

- We want to solve the homogenous linear system.

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

- $\mathbf{f}$ is 3x3 matrix and has *9 unknown*.
- For homogenous system, we less 1 DoF.
- Therefore, for the solution to be unique(up to a scale factor) the rank of A must be 8 and *we need* $N = 8$ *points*
- If $N > 8$ points, we will have least square solution

# For $N > 8$ *points*

- In theory, we have unique solution when $N = 8$
- However, the correspondences have *errors*.
- The reconstructed matrix will not have *rank deficiency*
- The solution to this system of homogeneous equations can be found in *Least Squares method* by *Singular Value Decomposition* and with the eignvector correspond to the smallest eigenvalue $F = f_N$

# The rank of reconstructed F

- $A$ can be reconstructed decomposed using SVD

  $\mathrm{A} = U\Sigma V^T$

- The solution is $\hat{\mathrm{f}} = v_9$ with $V = [v_1 \dots v_9]$

- The estimated $\hat{\mathrm{F}}$ may have full rank $(\mathrm{del}(\hat{\mathrm{F}}) \neq 0)$

- However *fundamental matrix* is *Rank 2*.

- What can we do?

# Enforcing Rank(F)=2

- We want to enforce the matrix F to *rank 2*
- Find F (a rank 2 matrix) that best approximate $\hat{\mathrm{F}}$
- i.e. Find F that minimize

$$\|\mathrm{F} - \hat{\mathrm{F}}\| = 0 \qquad \text{(Frobenius norm)}$$

- Then

$$\hat{\mathrm{F}} = U\Sigma V^T$$

$$\hat{\mathrm{F}} = UDiag(\sigma_{1,}\sigma_{2,}\sigma_3)V^T$$

- Setting the small diagonal value of $\Sigma$, and reconstruct F

$$\boxed{\mathrm{F} = UDiag(\sigma_{1,}\sigma_{2,}0)V^T}$$

# Enforcing F to Rank 2

Uncorrected F – epipolar lines are not coincident

Corrected F – epipolar lines coincident

Image Credit: R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision"

# MATLAB Example

1. Solve a system of homogeneous linear equations
   a. Write down the system of equations

Matlab:
```
For n=1 : size(xs,1)
      A(n,:) = kron(xss(n,:),xs(n,:));
End;
```

   b. Solve **f** from A**f**=**0** using SVD

Matlab:
```
[U, S, V] = svd(A);
f = V(:, end);
F = reshape(f, 3, 3)';
```

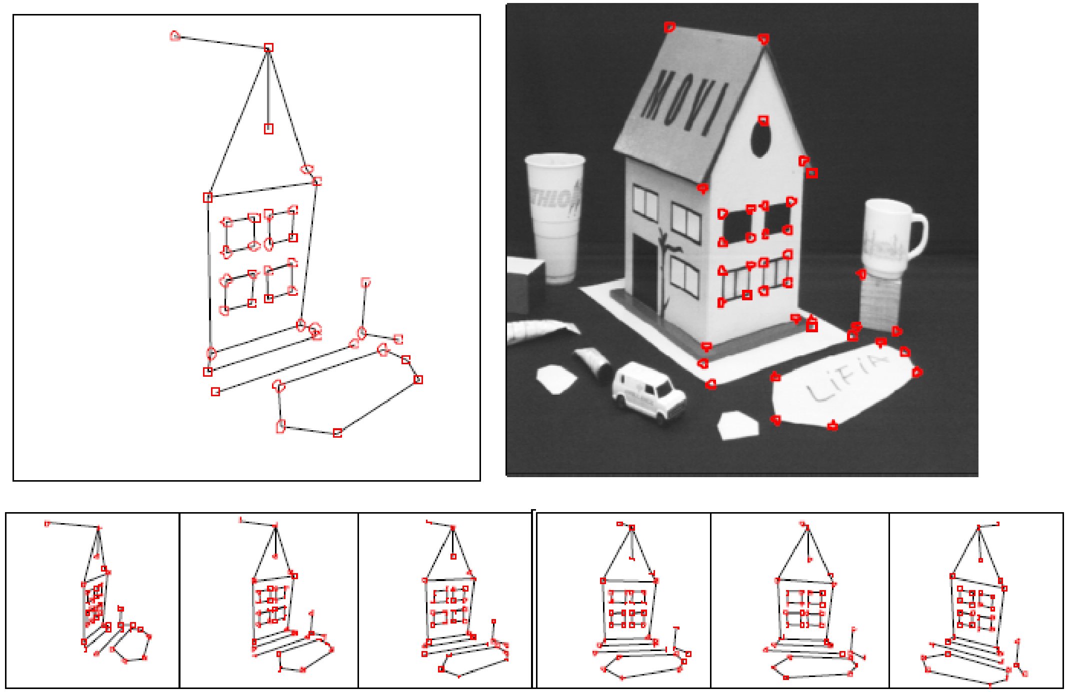
- Resolve det(F) = 0 constraint

  using SVD

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

# Problem with 8 points algorithm

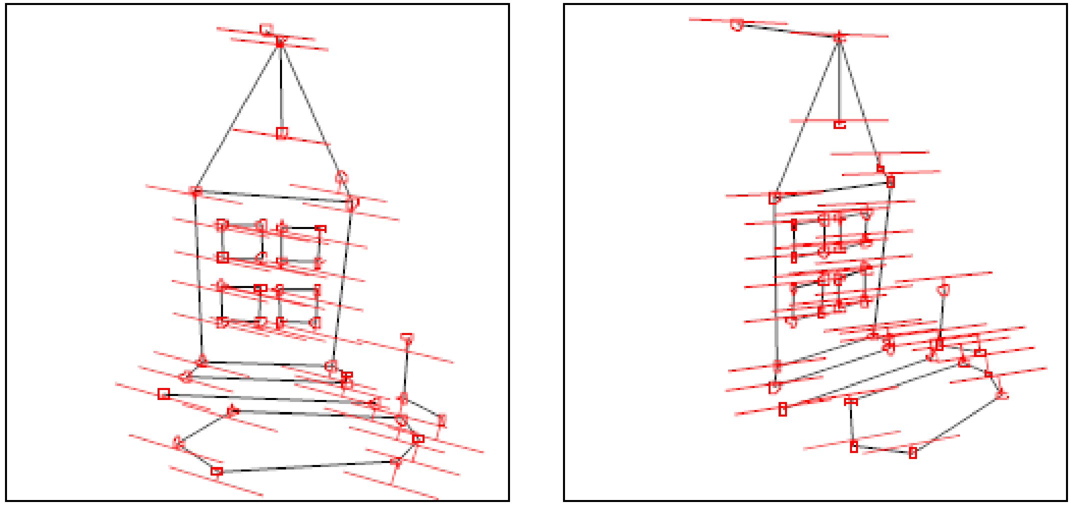
- F is often *ill-conditioned* (i.e.) small variation in the data points (x,y) coordinates selected will mess up the calculation
- For example




Data courtesy of R. Mohr and B. Boufama.

# Problem with 8 points algorithm

- Compute F
- Plot the *epipolar lines* of each correspondence
- Calculate the mean error (distance between epi-polar line and the correspond point)
- Result significantly large mean error
- Solution: Use *Normalized Eight-Point Algorithm*



Mean errors approx. 10/9.1 pixels

# Problem with 8 points algorithm

- What have

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

$$\begin{bmatrix} x_1''x_1' & x_1''y_1' & x_1'' & y_1''x_1' & y_1''y_1' & y_1'' & x_1' & y_1' & 1 \\ x_2''x_2' & x_2''y_2' & x_2'' & y_2''x_2' & y_2''y_2' & y_2'' & x_2' & y_2' & 1 \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ x_8''x_8' & x_8''y_8' & x_8'' & y_8''x_8' & y_8''y_8' & y_8'' & x_8' & y_8' & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{21} \\ F_{31} \\ F_{12} \\ F_{22} \\ F_{32} \\ F_{13} \\ F_{23} \\ F_{33} \end{bmatrix} = 0$$

- Highly un-balanced
- Value of A must have similar magnitude
- Otherwise will encounter problems during SVD

# Normalization

- Idea: Transform the image coordinates of the image pair such that the matrix A is well-conditioned

- For each image, apply a following transformation of $T$ (translation and scaling) such that

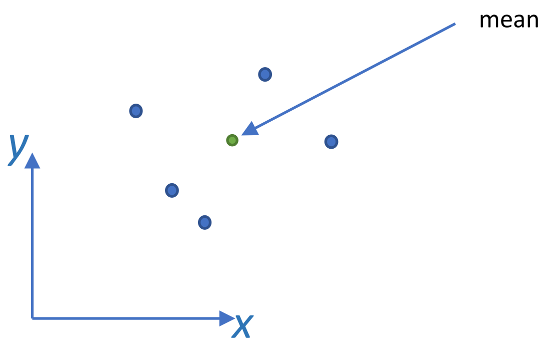
  - Origin at Centre
  - Mean square distance of image from origin approx. sqrt(2) pixels.

$$\mathbf{q}_i' = T\mathbf{x}_i' \qquad \& \quad \mathbf{q}_i'' = T\mathbf{x}_i''$$

# Compute the normalization matrix

- Compute the mean
  - `ctrd = mean(pts,2);`
- Compute distance from every point to mean
  - `dist = sqrt(sum((pts-ctrd).^2));`
- Scale it to sqrt(2)
  - `s = sqrt(2)/Mdist;`
- Define the Transformation matrix

$$T = \begin{bmatrix} s & 0 & -sctrd_x \\ 0 & s & -sctrd_y \\ 0 & 0 & 1 \end{bmatrix}$$



# Normalization

- Define

$$\mathbf{q}_i' = \mathrm{T}\mathbf{x}_i' \qquad \& \;\; \mathbf{q}_i'' = \mathrm{T}\mathbf{x}_i''$$

The fundamental matrix F

$$\mathbf{x}_i'^{T}\mathrm{F}\mathbf{x}_i'' = 0$$

$$(\mathrm{T}^{-1}\mathbf{q}_i')^{\mathrm{T}}\mathrm{F}(\mathrm{T}^{-1}\mathbf{q}_i'') = 0$$

$$\mathbf{q}_{\mathrm{i}}'^{\mathrm{T}}(\mathrm{T}^{-\mathrm{T}}\mathrm{F}\,\mathrm{T}^{-1})\mathbf{q}_i'' = 0$$

$$\mathbf{q}_{\mathrm{i}}'^{\mathrm{T}}F_q\mathbf{q}_i'' = 0$$

$$\mathrm{F}_{\mathrm{q}} = \mathrm{T}^{-\mathrm{T}}\mathrm{F}\mathrm{T}^{-1}$$

$$\mathrm{F} = \mathrm{T}^{\mathrm{T}}\mathrm{F}_{\mathrm{q}}\mathrm{T}$$

# Steps for normalized Eight-Point Algorithm

1. Compute $T$
2. Normalize the coordinates in images 1 and 2

   $q_i' = T\mathrm{x}_i'$ & $q_i''=T\mathrm{x}_i''$
3. Use SVD to compute $\hat{F}_q$ from $q_i'$ and $q_i''$
4. Enforce $\hat{F}_q$ to rank sqrt(2)
5. De-normalize $\mathrm{F} = \mathrm{T}^{\mathrm{T}}\mathrm{F}_{\mathrm{q}}\mathrm{T}$

# Today’s Agenda

- Number of Parameters
- Coplanarity Constraints for Image of Uncalibrated Cameras
- Fundamental Matrix
- Epi-polar Geometry
- Estimation of Fundamental Matrix by 8 Point Theorem
- The Essential Matrix

# Calibrated Camera Case

- For images of calibrated cameras, the coplanarity constraint can be simplified using the direction ${}^{c}\mathrm{x}'$ and ${}^{c}\mathrm{x}''$ of the camera.

- ${}^{c}\mathrm{x}'$ and ${}^{c}\mathrm{x}''$ can be computed by the calibration matrices

$${}^{c}\mathrm{x}' = \mathrm{K}'^{-1}\mathrm{x}' \qquad\qquad {}^{c}\mathrm{x}'' = \mathrm{K}''^{-1}\mathrm{x}''$$

- From the projection

$$\mathrm{x}' = \mathrm{P}'\mathrm{X}' = \mathrm{K}'R'[I_3| - \mathbf{X_0}]\boldsymbol{X} = \mathrm{K}'\ {}^{c}\mathrm{x}'$$

- Hence $\qquad {}^{c}\mathrm{x}' = \mathrm{K}'^{-1}\mathrm{x}'$

# Coplanarity constraints

- For uncalibrated camera, the coplanarity constraints is as follow:

$$\mathbf{x}'^{T}\mathrm{F}\,\mathbf{x}'' = 0 \qquad \text{where} \qquad \mathrm{F}=(\mathrm{K}')^{-T}R'\,S_b\,(R'')^{T}(\mathrm{K}'')^{-1}$$

$$\underbrace{\mathbf{x}'^{T}(\mathrm{K}')^{-T}}_{{}^{c}\mathrm{x}'^{T}}R'\,S_b\,(R'')^{T}\underbrace{(\mathrm{K}'')^{-1}\mathbf{x}''}_{{}^{c}\mathrm{x}''} = 0$$

$${}^{c}\mathrm{x}'^{T}\underbrace{R'\,S_b\,(R'')^{T}}_{E}\;{}^{c}\mathrm{x}'' = 0$$

# Essential Matrix

$$ {}^{c}\mathrm{x}'^{T} R' S_b (R'')^{T} \; {}^{c}\mathrm{x}'' = 0 $$

$$ {}^{c}\mathrm{x}'^{\mathrm{T}} \mathrm{E} \; {}^{c}\mathrm{x}'' = 0 $$

This gives the coplanarity constraint for calibrated camera

- E is the essential matrix of the calibrated camera pair
  - $\mathrm{E} = R' S_b (R'')^{T}$

# The properties of Essential Matrix

The essential matrix

$$\mathrm{E} = R' \, S_b \, (R'')^T \quad \text{or} \qquad \mathrm{E} = S_b \, (R'')^T \ \text{ for } \quad R' = I$$

- The essential matrix are five degrees of freedom (
- The five parameters determine the relative orientation of the image pair for calibrated cameras
- Essential matrix E is 3x3 matrix and requires 9 - 5 = 4 constraints.
- The essential matrix E is singular, det(E)=0
- The essential matrix is homogenous

# Estimating the Essential Matrix with 8 points

- For each point, we have the coplanarity constraint

$$ {}^{c}\mathrm{x}_i'^{T}\,\mathrm{E}\,{}^{c}\mathrm{x}_i'' = 0 \qquad \text{for} \qquad \text{for } i = 1 \dots N $$

- In Matrix form

$$ [\,{}^{c}\mathrm{x}_i',\ {}^{c}\mathrm{y}_i', 1] \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{bmatrix} \begin{bmatrix} {}^{c}\mathrm{x}_i'' \\ {}^{c}\mathrm{y}_i'' \\ 1 \end{bmatrix} = 0 $$

for $i = 1 \dots N$

- Represented by Knrocker product

# Represented using Kronecker Product

- Define

$$\mathrm{e} = \mathrm{vecE} = [E_{11}, E_{21}, E_{31}, E_{12}, E_{22}, E_{32}, E_{13}, E_{23}, E_{33}]^T$$

$$\mathrm{a}_i^{\mathrm{T}} = {}^c\mathrm{x}_i''^T \otimes {}^c\mathrm{x}_i'^T = [\, {}^c x_i'' \, {}^c x_i', \; {}^c x_i'' \, {}^c y_i' \; \, {}^c x_i'', \; {}^c y_i'' \, {}^c x_i', \; {}^c y_i'' \, {}^c y_i', \; {}^c y_i'', \; {}^c x_i' \, {}^c y_i', 1]$$

Therefore the constraints are written as

$$\mathrm{a}_i^{\mathrm{T}}\mathrm{e} = 0 \text{ for} \qquad \text{for } i = 1 \ldots N$$

Hence we stack up all points

$$\mathrm{A} = \begin{bmatrix} \mathrm{a}_i^{\mathrm{T}} \\ \vdots \\ \mathrm{a}_n^{\mathrm{T}} \end{bmatrix} \Rightarrow \mathrm{A}\mathbf{E}=0$$

# Properties of the Essential Matrix

- E is *homogenous*
- E is *singular* det(E)=0
- Two *identical* non-zero singular values (Fronster Photogrametry P.557)

$$\mathrm{E} = U\Sigma V^T = U\begin{bmatrix} d & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T$$

# Constraints for E

- For fundamental matrix, we enforced to rank 2

$$\mathrm{E} = U\Sigma V^T = UDiag(D_{11}, D_{22}, 0)V^T$$

- For essential matrix what is the rank?
- Due to homogenous property, E is represented as

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

# Normalization

- Idea: Transform the image coordinates of the image pair such that the matrix A is well-conditioned

- For each image, apply a following transformation of $T$ (translation and scaling) such that
  - Origin at Centre
  - Mean square distance of image from origin approx. sqrt(2) pixels.
  - $\mathbf{q}_i' = T\mathbf{x}_i{}'$ & $\mathbf{q}_i'' = T\mathbf{x}_i{}''$

# MATLAB Example

1. Solve a system of homogeneous linear equations
   a. Write down the system of equations

Matlab:

```
For n=1 : size(xs,1)
      A(n,:) = kron(xss(n,:),xs(n,:));
End;
```

   b. Solve **e** from A**f**=**0** using SVD

Matlab:

```
[U, S, V] = svd(A);
e = V(:, end);
E = reshape(f, 3, 3)';
```

- Resolve det(E) = 0 constraint
  using SVD

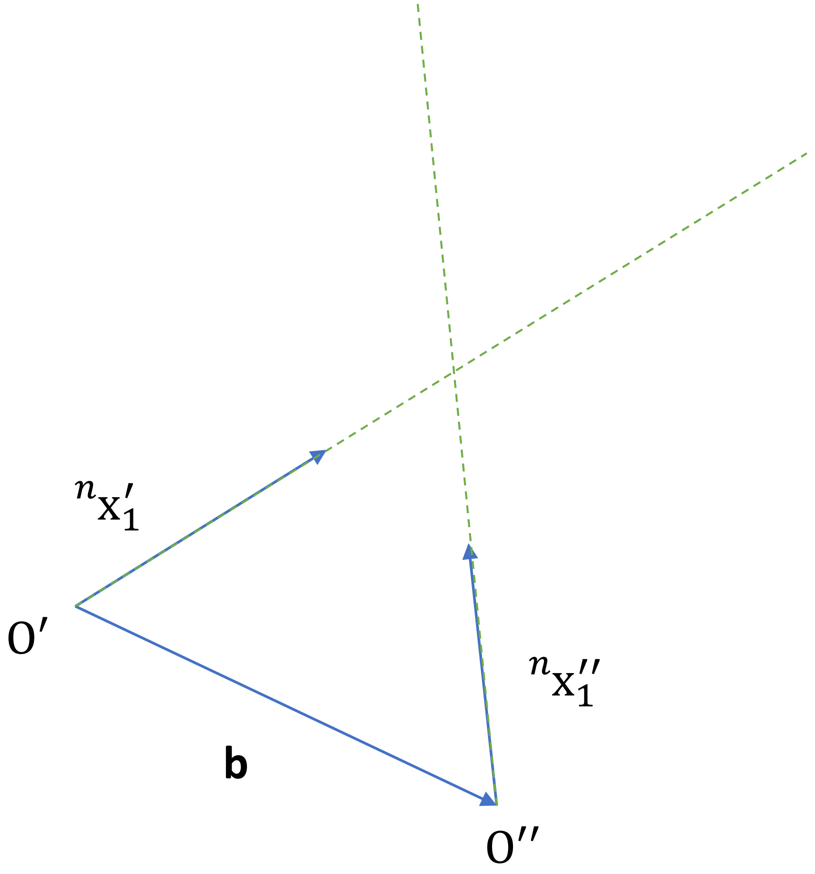
Matlab:

```
[U, S, V] = svd(E);
E = U*diag([1,1,0],*V';
```

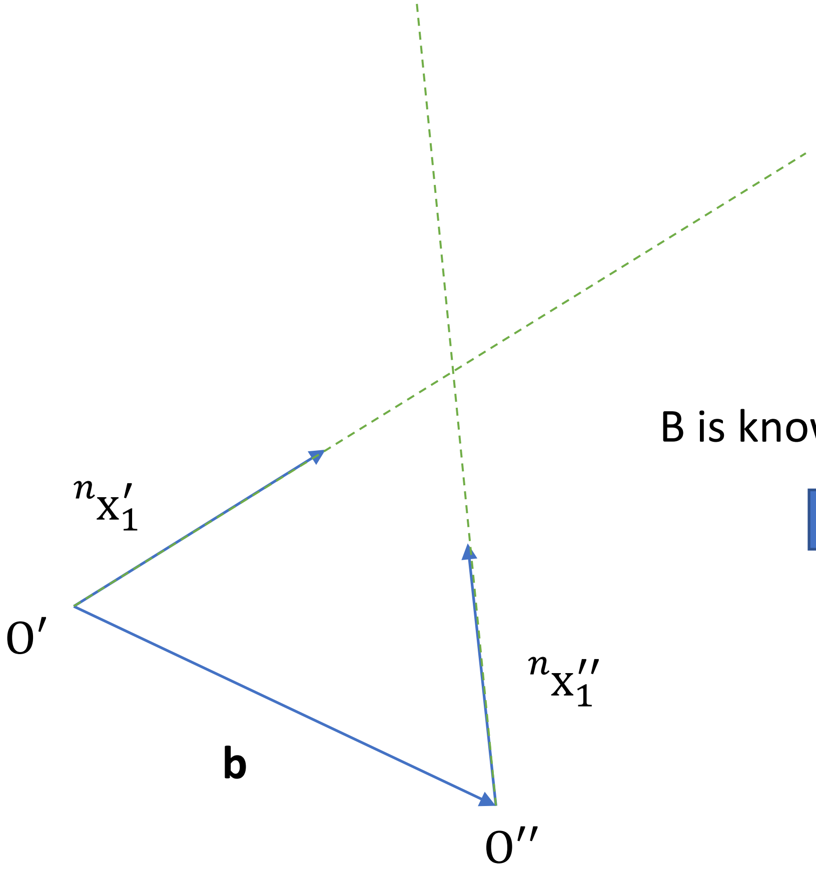
# Determining the Orientation Parameters

- The Essential Matrix E can be written as
  - $\mathrm{E} = S_b R^T$
- where $S_b$ is the *skew symmetric matrix* of the base vector, $R = R''$ is the rotation of second image w.r.t. the first image
- Is it a unique solution?

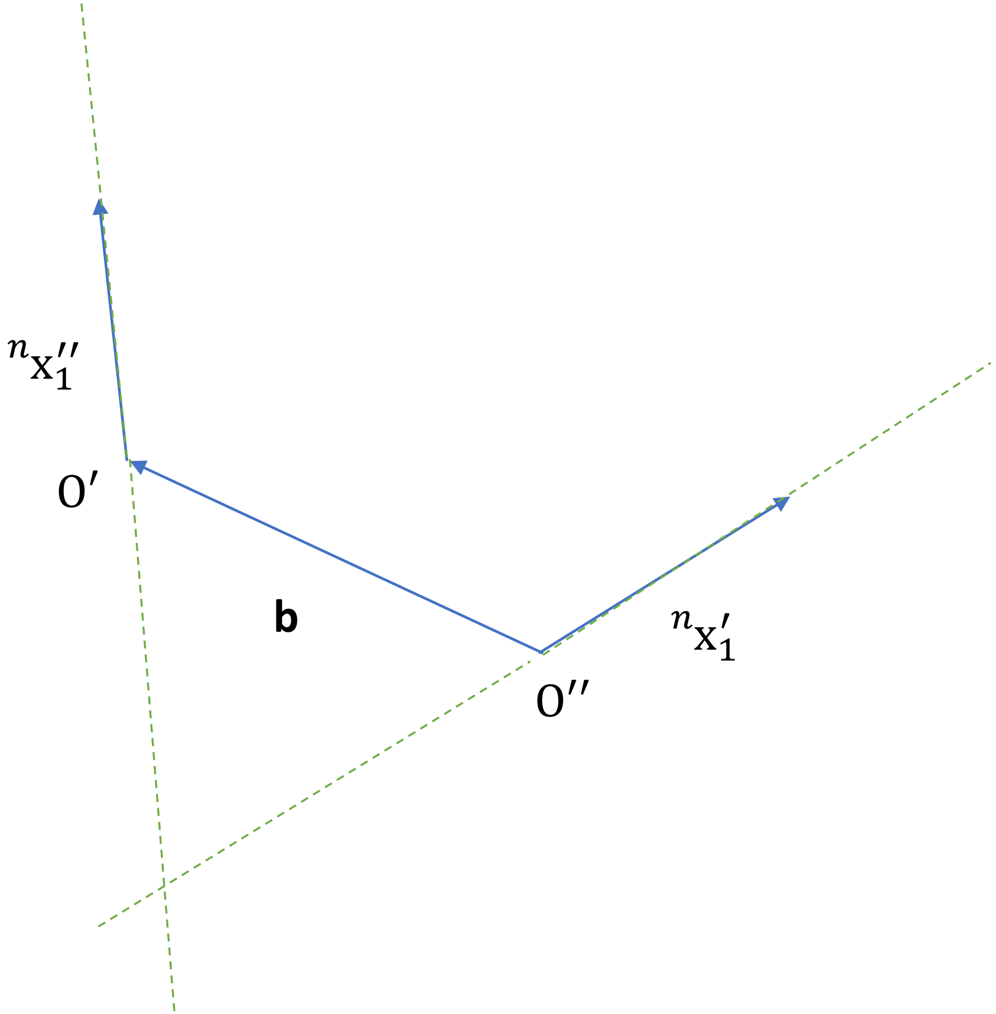
# Is camera reconstructed from E unique?



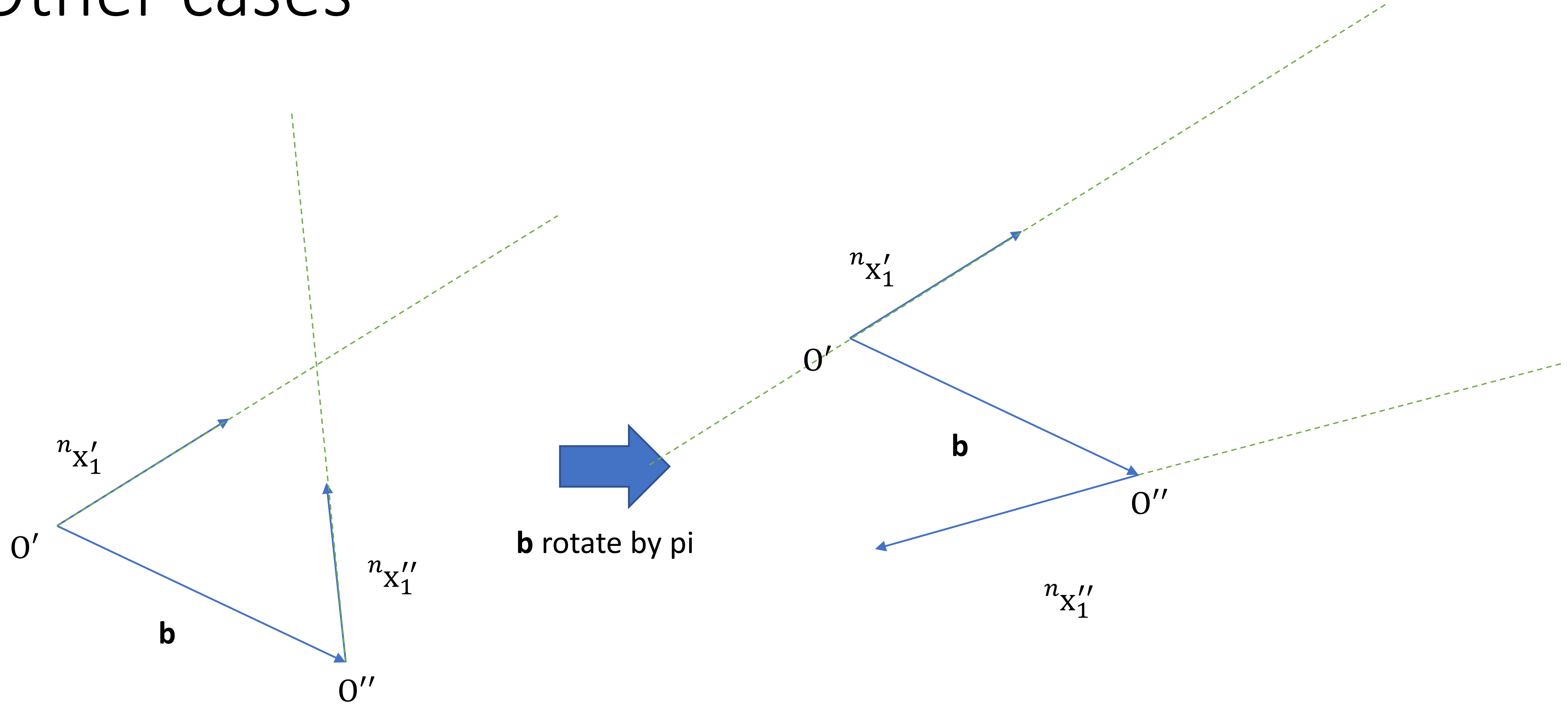
# What if?




B is known up to scale

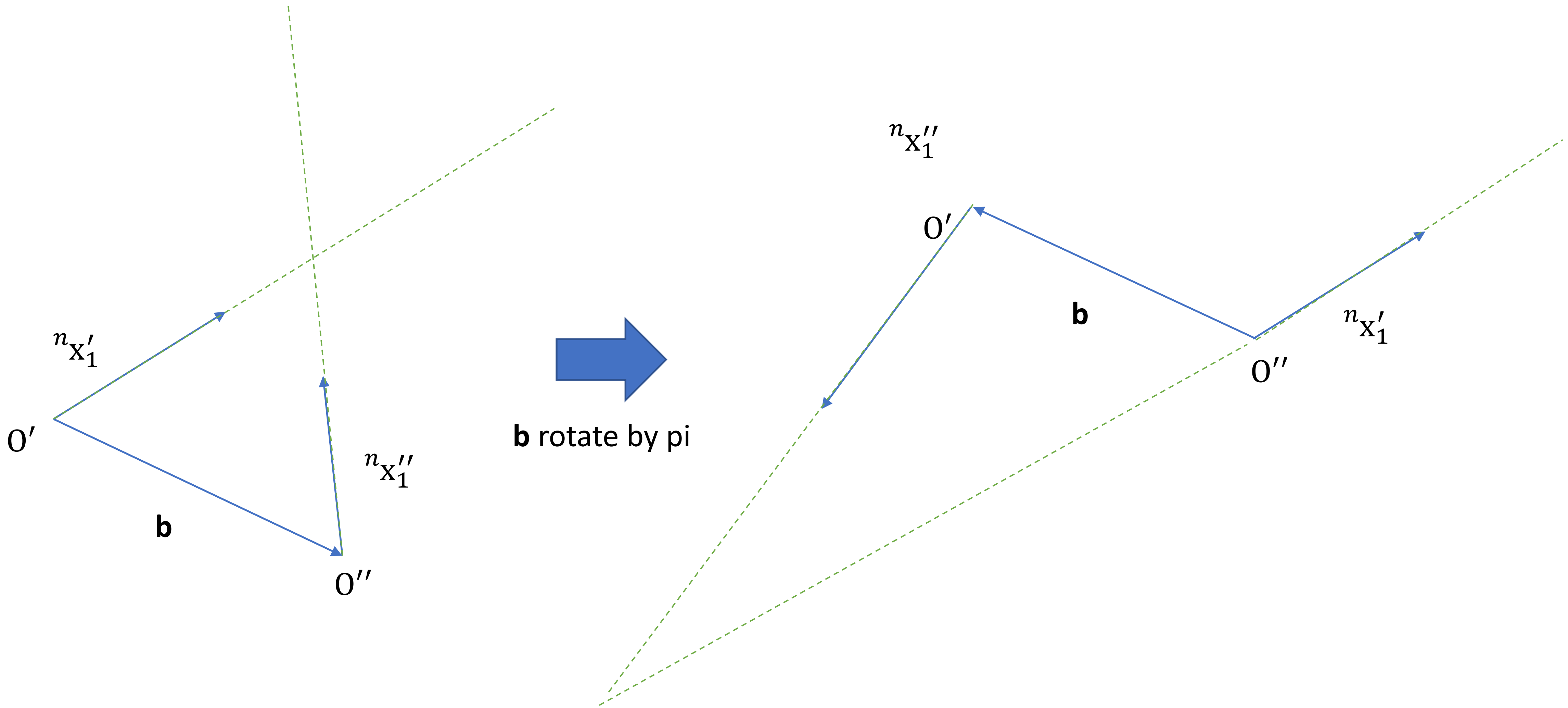


# Other cases



${}^{n}\mathrm{x}_1'$
$O'$
**b**
${}^{n}\mathrm{x}_1''$
$O''$
**b** rotate by pi
${}^{n}\mathrm{x}_1'$
$O'$
**b**
$O''$
${}^{n}\mathrm{x}_1''$

# Other cases



# Four possible solution for calibrated reconstruction from E

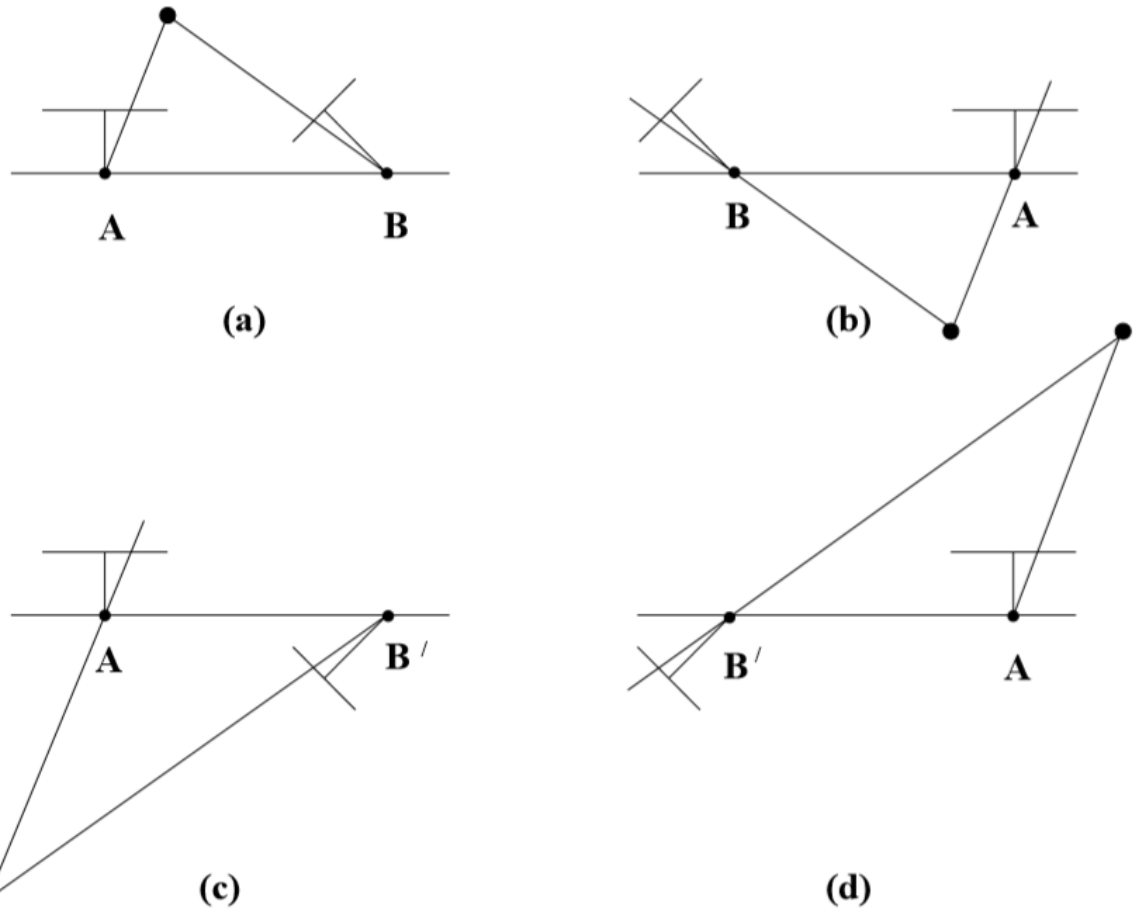



Image Credit: R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”

# Solution proposed by Hartley

- The essential Matrix

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

- Let

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

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

- Where

$$ZW = \begin{bmatrix}0 & 1 & 0\\-1 & 0 & 0\\0 & 0 & 0\end{bmatrix}\begin{bmatrix}0 & 1 & 0\\-1 & 0 & 0\\0 & 0 & 1\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 0\end{bmatrix} = Diag([1,1,0])$$

# Solution proposed by Hartley

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

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

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

# Four solutions for Z and W

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

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

# 4 solutions

- $E = \underbrace{U \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} U^T}_{S_B} \underbrace{U \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} V^{\mathrm{T}}}_{R^T}$

- There are 2 solutions for $S_B$ and $R^T$

$S_B = UZU^T$ or $S_B = UZ^TU^T$

$R^T = UWV^T$ or $R^T = UW^TV^T$

# 4 Solutions for Pose

- $E = S_B R^T$

- $S_B = UZU^T$ or $S_B = UZ^TU^T$
- $R^T = UWV^T$ or $R^T = UW^TV^T$

$$E = UZU^TUWV^T$$
$$E = UZ^TU^TUWV^T$$
$$E = UZU^TUW^TV^T$$
$$E = UZ^TU^TUW^TV^T$$

# Summary of Solution by Hartley

- Compute the SVD of E

$$U\Sigma V^T = svd(E)$$

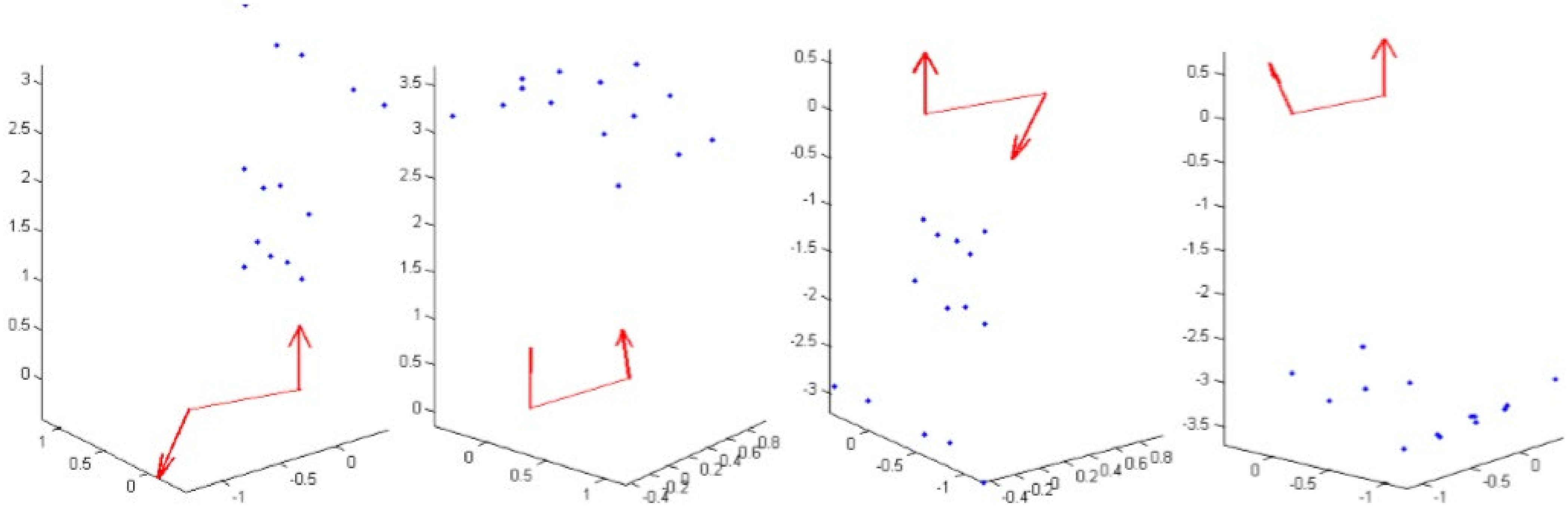
- Compute the four solutions

$$S_B = UZU^T \quad \text{or} \quad S_B = UZ^TU^T$$
$$R^T = VWV^T \text{ or } \quad R^T = VW^TV^T$$

- Test if the points are in front of the camera for each case.

- Return only physical plausible solution.

# Example – Reconstruction from calibrated cameras




Physical plausible solution

Image Courtesy: C. Olsson

# Summary of Essential Matrix

- The essential matrix $\mathrm{E} = R' \; S_b \; (R'')^T$ express the coplanarity constraint of two calibrated camera
- ${}^{c}\mathrm{x}'^{\,T} \mathrm{E} \; {}^{c}\mathrm{x}'' = 0$
- $\mathrm{E}$ express the R.O.
- $\mathrm{E}$ has 5-DoF

# Summary of Essential Matrices

| camera/parametrization<br>a priori constraints | matrix $\mathsf{F}$ or $\mathsf{E}$ |
|---|---|
| uncalibrated perspective camera | $\mathsf{F} = \mathsf{K}'^{-\mathsf{T}} \mathsf{R}' \mathsf{S}_b \mathsf{R}'^{\mathsf{T}} \mathsf{K}''^{-1}$ |
| spherical camera,<br>calibrated perspective camera | |
| indep. images general<br>$R' = I_3$ | $\mathsf{E} = \mathsf{S}_b R^{\mathsf{T}}$ |
| indep. images special<br>$R' = I_3$, $B_X$=const. | $\mathsf{E} = \mathsf{S}_b R^{\mathsf{T}}$ |
| dependent images<br>$\mathbf{b} = (B_X, 0, 0)^{\mathsf{T}}$, $\omega' = -\omega'' = -\frac{1}{2}\Delta\omega$ | $\mathsf{E} = R' \mathsf{S}_b R''^{\mathsf{T}}$ |
| normalized cameras<br>$R' = R'' = I_3$<br>$\mathsf{K}' = \mathsf{K}'' = I_3$ | $\mathsf{E} = \mathsf{S}_b$ |
| normal case<br>$R' = R'' = I_3$<br>$\mathsf{K}' = \mathsf{K}'' = \mathrm{Diag}([c, c, 1])$<br>$\mathbf{b} = (B_X, 0, 0)^{\mathsf{T}}$ | $\mathsf{E} = \mathsf{S}_b$ |

Source: Frosnter

# Fundamental Matrix Song

back

# Reference

- Wolfgang Forstner and Bernhard P. Wrobel, “Photogrammetric Computer Vision”, Chapter 13