



3D Computer Vision II

Multiple View Geometry The Trifocal Tensor

Nassir Navab

based on a course given at UNC by Marc Pollefeys & the book "Multiple View Geometry" by Hartley & Zisserman

December 8, 2009

Multiple View geometry



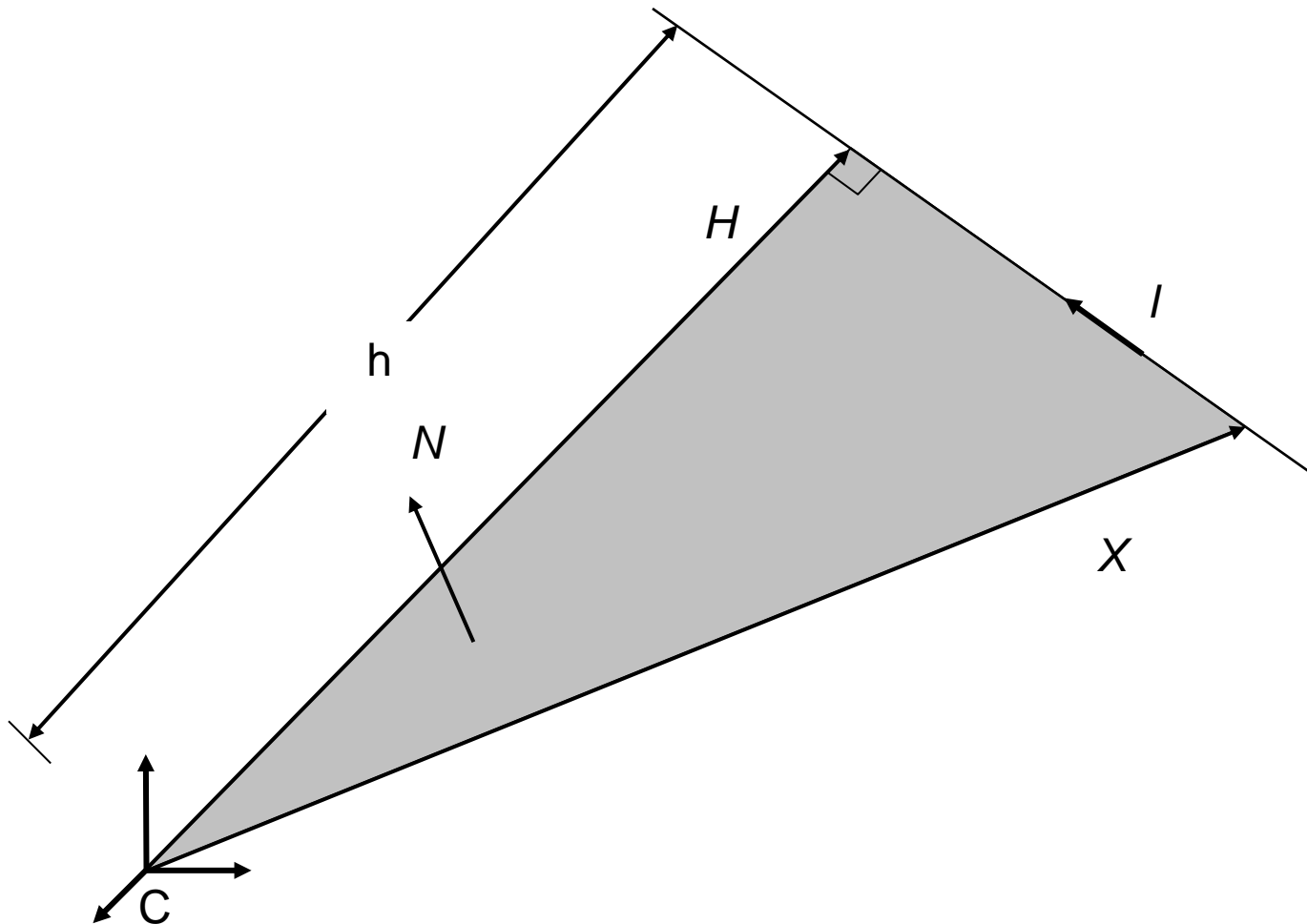
Outline – Multiple View Geometry

- Introduction
 - Representation and motion of lines
- Geometric Introduction of the Trifocal Tensor
 - Line incidence
 - Homographies induced by a plane
 - Point and line Relationships induced by the trifocal tensor
- Extraction of projection & fundamental matrices
- Computing the trifocal tensor
 - Linear method (7-point)
 - Minimal method (6-point)
 - Geometric error minimization method
 - RANSAC method

Representation and Motion of Lines:

Here we will first describe the estimation of motion of lines from a triplet of calibrated views. This will help the further understanding of the tri-focal tensors.

The two vectors (\mathbf{I} , \mathbf{N}) define the Plücker coordinates of a line L .

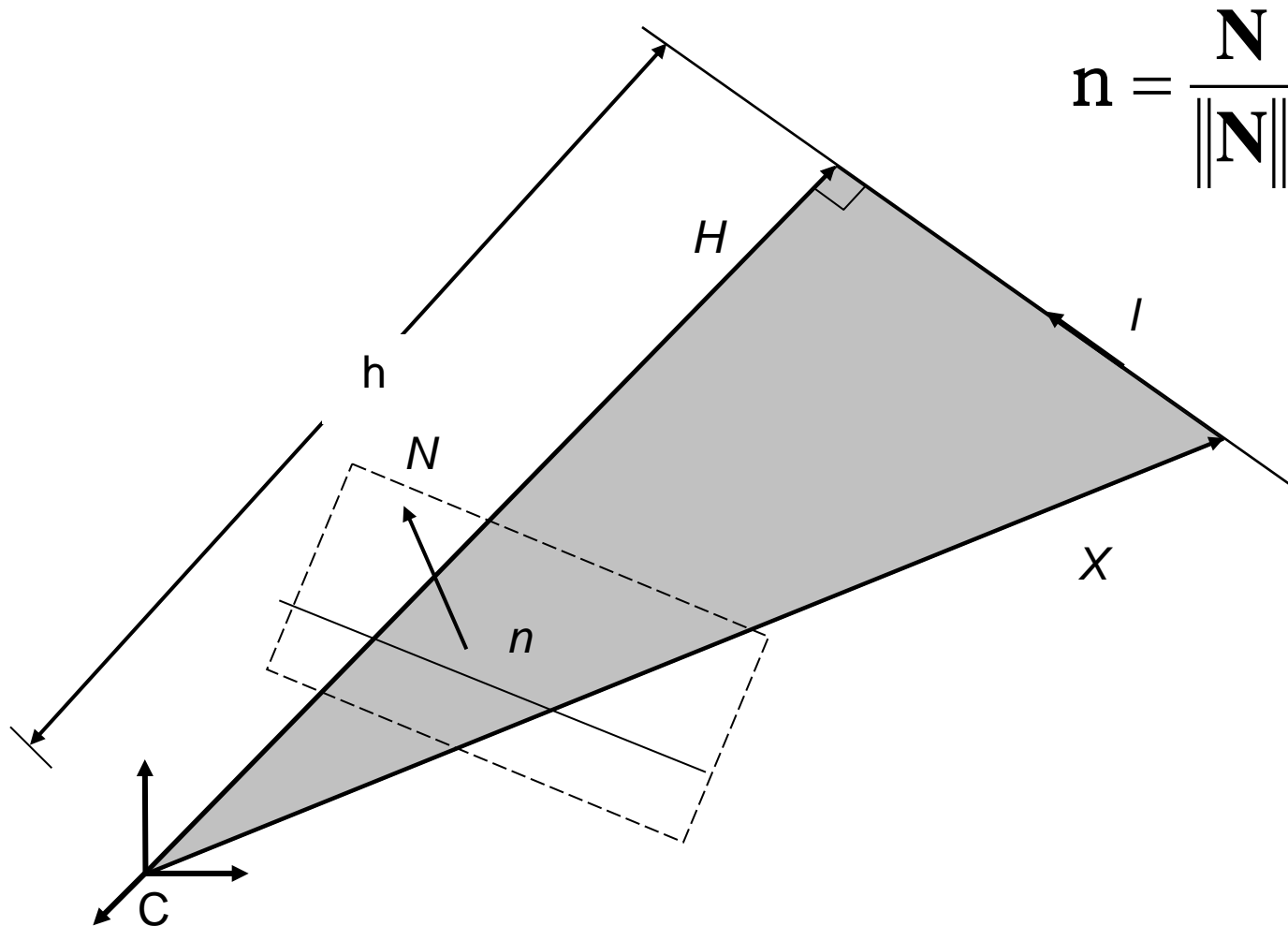


$$\mathbf{X} \times \mathbf{I} = \mathbf{N}$$

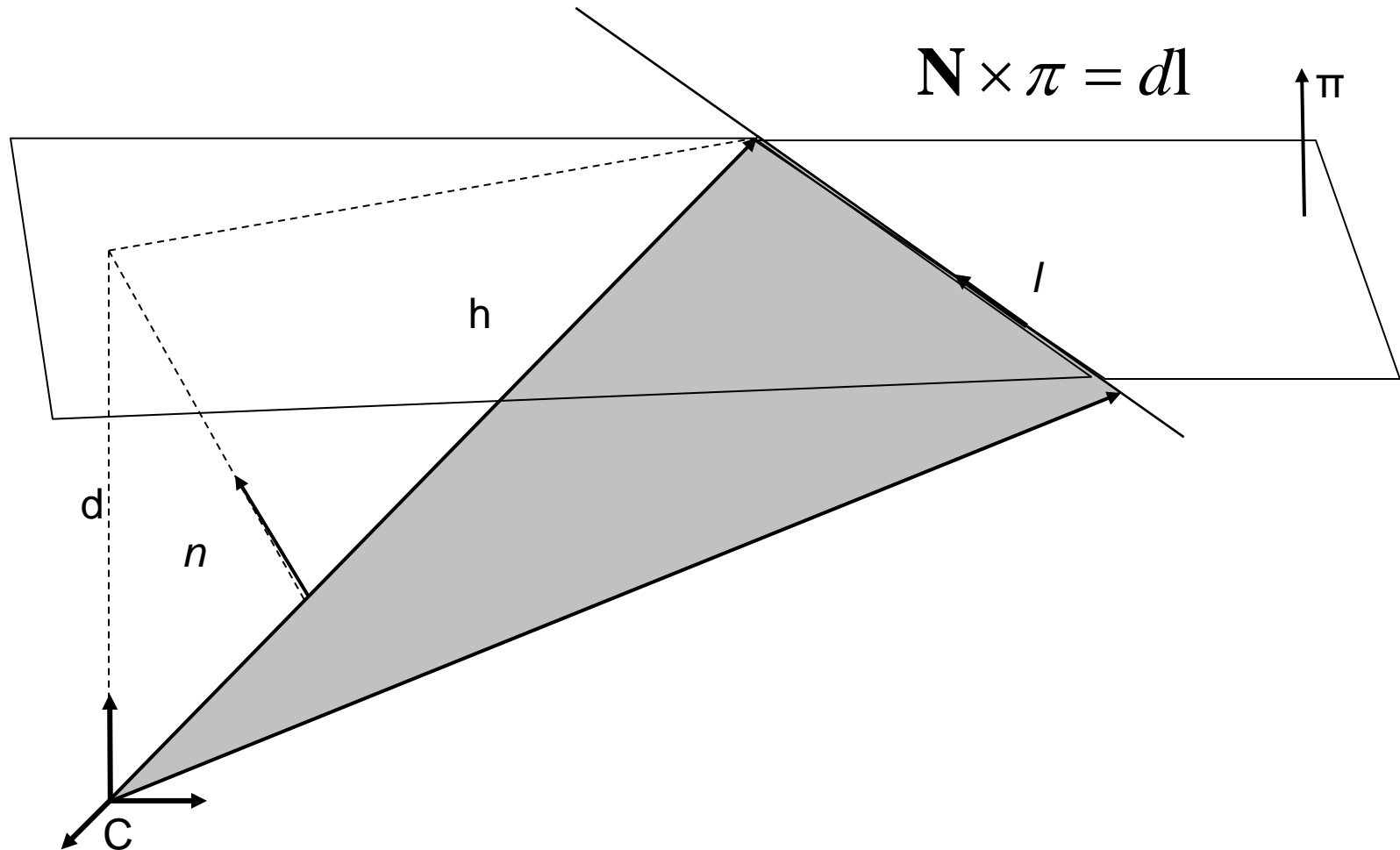
$$\|\mathbf{N}\| = h$$

$$\mathbf{N}^T \mathbf{I} = 0$$

In camera coordinates image of (I, N) is defined as:



A line (\mathbf{l}, \mathbf{N}) lies on a plane Γ defined by (π, d) if and only if:



Two intersecting lines $(\mathbf{l}_1, \mathbf{N}_1)$ and $(\mathbf{l}_2, \mathbf{N}_2)$:

$$\mathbf{X} \times \mathbf{l}_1 = \mathbf{N}_1 \quad \text{and} \quad \mathbf{X} \times \mathbf{l}_2 = \mathbf{N}_2$$

$$\mathbf{l}_2^T (\mathbf{X} \times \mathbf{l}_1) = \mathbf{l}_2^T \mathbf{N}_1 \quad \text{and} \quad \mathbf{l}_1^T (\mathbf{X} \times \mathbf{l}_2) = \mathbf{l}_1^T \mathbf{N}_2$$

$$\mathbf{l}_2^T \mathbf{N}_1 + \mathbf{l}_1^T \mathbf{N}_2 = \mathbf{l}_2^T (\mathbf{X} \times \mathbf{l}_1) + \mathbf{l}_1^T (\mathbf{X} \times \mathbf{l}_2)$$

$$\mathbf{l}_2^T \mathbf{N}_1 + \mathbf{l}_1^T \mathbf{N}_2 = 0$$

Two intersecting lines $(\mathbf{l}_1, \mathbf{N}_1)$ and $(\mathbf{l}_2, \mathbf{N}_2)$:

$$\mathbf{N}_1 \times \pi' = \mathbf{l}_1 \quad \text{and} \quad \mathbf{N}_2 \times \pi' = \mathbf{l}_2 \quad (\pi' = \pi / d)$$

$$\mathbf{N}_2^T (\mathbf{N}_1 \times \pi') = \mathbf{N}_2^T \mathbf{l}_1 \quad \text{and} \quad \mathbf{N}_1^T (\mathbf{N}_2 \times \pi') = \mathbf{N}_1^T \mathbf{l}_2$$

$$\mathbf{l}_2^T \mathbf{N}_1 + \mathbf{l}_1^T \mathbf{N}_2 = \mathbf{N}_1^T (\mathbf{N}_2 \times \pi') + \mathbf{N}_2^T (\mathbf{N}_1 \times \pi')$$

$$\mathbf{l}_2^T \mathbf{N}_1 + \mathbf{l}_1^T \mathbf{N}_2 = 0$$

Motion of Lines

Representation of a line (\mathbf{l} , \mathbf{N}) after a rigid transformation (R, T):

$$X \rightarrow X' = RX + T$$

$$l = \frac{X_1 - X_2}{\|X_1 - X_2\|}$$

$$l \rightarrow l' = Rl$$

$$N \rightarrow N' = (RX + T) \times Rl = RN + [T]_{\times} Rl$$

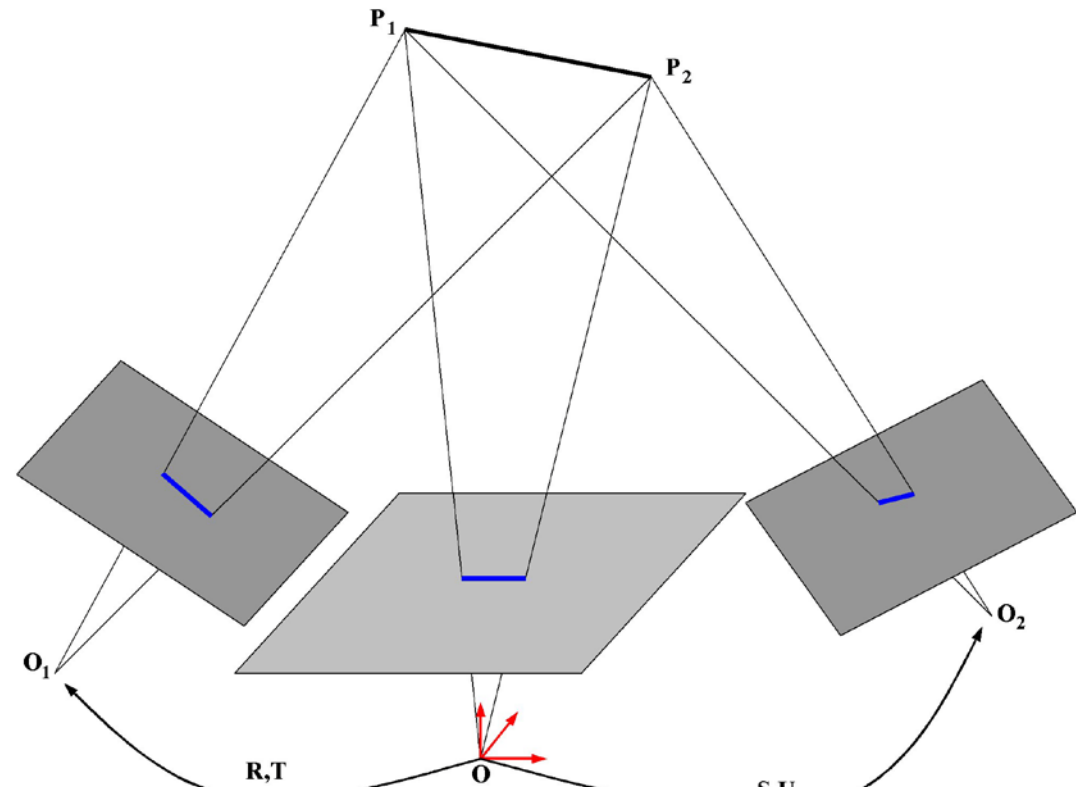
$$\begin{bmatrix} l \\ N \end{bmatrix} \rightarrow \begin{bmatrix} l' \\ N' \end{bmatrix} = \begin{bmatrix} R & 0 \\ E & R \end{bmatrix} \begin{bmatrix} l \\ N \end{bmatrix}$$

Representation of the image of a line (\mathbf{l} , \mathbf{N}) after a rigid transformation (R, T):

$$l \rightarrow l' = Rl$$

$$N' = RN + [T]_{\times} Rl$$

Three View Geometry of Lines



$$\Pi = \begin{bmatrix} n \\ 0 \end{bmatrix}$$

$$\Pi' = \begin{bmatrix} R'^T n' \\ T'^T n' \end{bmatrix}$$

$$\Pi'' = \begin{bmatrix} R''^T n'' \\ T''^T n'' \end{bmatrix}$$

Three View Geometry of Lines

The three plane intersecting in one line are dependent. Therefore the following matrix is of rank two:

$$\begin{bmatrix} n & R'^T n' & R''^T n'' \\ 0 & T'^T n' & T''^T n'' \end{bmatrix}$$

All 3x3 sub-matrices are of determinant zero.

Three View Geometry of Lines

The three plane intersecting in one line are dependent. Therefore the following matrix is of rank two:

$$\det \begin{vmatrix} n & R'^T n' & R''^T n'' \end{vmatrix} = 0$$

$$n^T (R'^T n' \times R''^T n'') = 0$$

One non-linear equation in 6 parameters of camera orientations. 6 line correspondences results in non-linear solution to orientation estimation.

Three View Geometry of Lines

$$n^T (R'^T n' \times R''^T n'') = 0$$

Geometric interpretation: the three viewing planes include the 3D line. Their normals are all orthogonal to the line direction and therefore coplanar. The cross product of two of the normals is parallel to the 3D line direction and therefore orthogonal to the third normal.

Three View Geometry of Lines

The three plane intersecting in one line are dependent. Therefore the following matrix is of rank two:

$$\begin{bmatrix} n & R'^T n' & R''^T n'' \\ 0 & T'^T n' & T''^T n'' \end{bmatrix}$$

Rank of the matrix being two, the first two columns of the following matrix need to be parallel:

$$\begin{bmatrix} n & (T''^T n'')R'^T n' - (T'^T n')R''^T n'' & R''^T n'' \\ 0 & 0 & T''^T n'' \end{bmatrix}$$

Three View Geometry of Lines

The three plane intersecting in one line are dependent. Therefore the following matrix is of rank two:

$$\begin{bmatrix} n & (T''^T n'')R'^T n' - (T'^T n')R''^T n'' & R''^T n'' \\ 0 & 0 & T''^T n'' \end{bmatrix}$$

$$n \times \left[(T''^T n'')R'^T n' - (T'^T n')R''^T n'' \right] = 0$$

If orientation of the cameras are obtained through non-linear estimation using 6 triplets of line correspondences, the translation vectors can then be estimated linearly.

Three View Geometry of Lines

In order to find a linear solution for motion estimation, one can represent the following equations in term of three 3x3 matrices as follows:

$$n \times \left[(T''^T n'') R'^T n' - (T'^T n') R''^T n'' \right] = 0$$

$$n \times \left[(T''^T n'') \begin{bmatrix} R'_1{}^T n' \\ R'_2{}^T n' \\ R'_3{}^T n' \end{bmatrix} - (T'^T n') \begin{bmatrix} R''_1{}^T n'' \\ R''_2{}^T n'' \\ R''_3{}^T n'' \end{bmatrix} \right] = 0$$

Three View Geometry of Lines

In order to find a linear solution for motion estimation, one can represent the following equations in term of three 3x3 matrices as follows:

$$n \times \left[\begin{array}{c} n'^T R'_1 \\ n'^T R'_2 \\ n'^T R'_3 \end{array} \right] (T''^T n'') - (n'^T T') \left[\begin{array}{c} R_1''^T n'' \\ R_2''^T n'' \\ R_3''^T n'' \end{array} \right] = 0$$

$$n \times \left[\begin{array}{c} n'^T (R'_1 T''^T - T' R_1''^T) n'' \\ n'^T (R'_2 T''^T - T' R_2''^T) n'' \\ n'^T (R'_3 T''^T - T' R_3''^T) n'' \end{array} \right] = 0$$

Three View Geometry of Lines

$$n \times \begin{bmatrix} n'^T (R_1' T''^T - T' R_1''^T) n'' \\ n'^T (R_2' T''^T - T' R_2''^T) n'' \\ n'^T (R_3' T''^T - T' R_3''^T) n'' \end{bmatrix} = 0$$

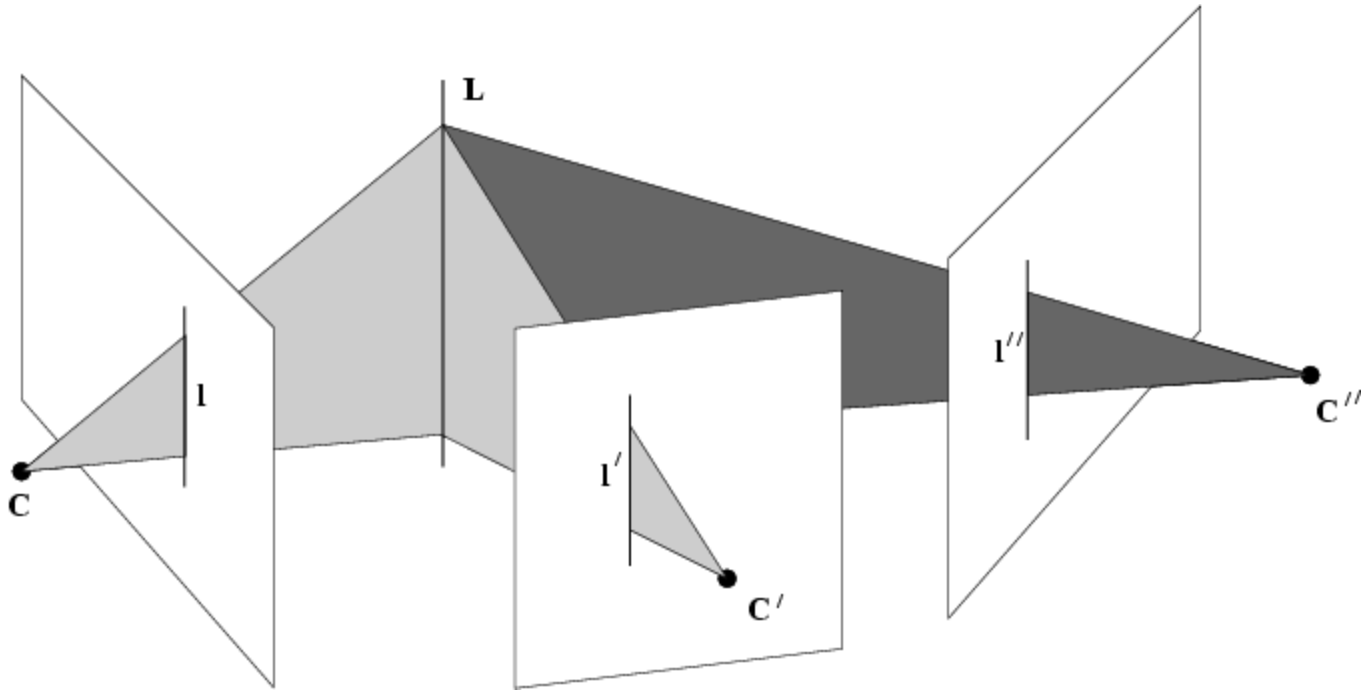
$$n \times \begin{bmatrix} n'^T E n'' \\ n'^T F n'' \\ n'^T G n'' \end{bmatrix} = 0$$

13 line correspondences over three calibrated views result in linear estimation of E, F and G. Motion parameters can then be recovered from these matrices.

Outline – Multiple View Geometry

- Introduction
 - Representation and motion of lines
- Geometric Introduction of the Trifocal Tensor
 - Line incidence
 - Homographies induced by a plane
 - Point and line Relationships induced by the trifocal tensor
- Extraction of projection & fundamental matrices
- Computing the trifocal tensor
 - Linear method (7-point)
 - Minimal method (6-point)
 - Geometric error minimization method
 - RANSAC method

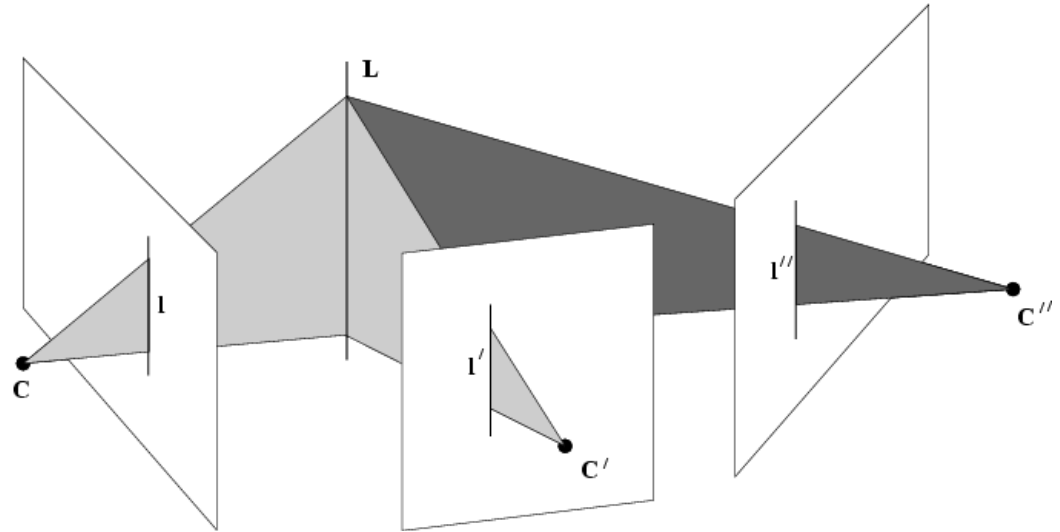
The Trifocal Tensor



Three back-projected lines have to meet in a single line.
Incidence relation provides constraint on lines.
Let us derive the corresponding algebraic constraint...

Notation

$$l_i \leftrightarrow l'_i \leftrightarrow l''_i$$



$$P = [I \mid 0]$$

$$P' = [A \mid a_4]$$

$$P'' = [B \mid b_4]$$

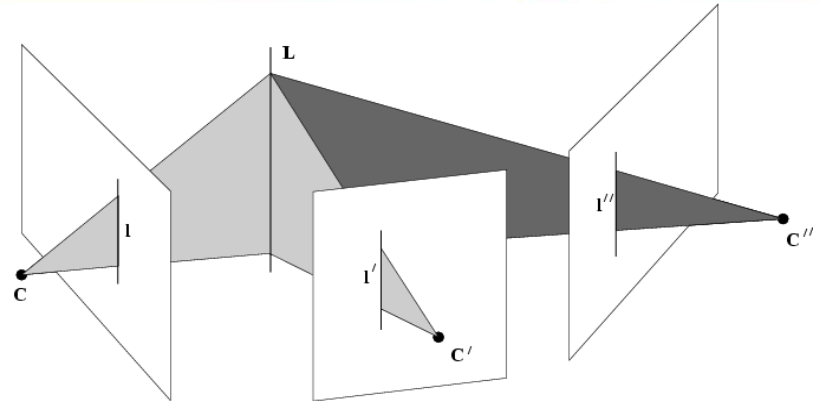
$$\pi = P^T l = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\pi' = P'^T l' = \begin{pmatrix} A^T l' \\ a_4^T l' \end{pmatrix}$$

$$\pi'' = P''^T l'' = \begin{pmatrix} B^T l'' \\ b_4^T l'' \end{pmatrix}$$

Incidence

$$l_i \leftrightarrow l'_i \leftrightarrow l''_i$$



$$\pi = P^T l = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \pi' = P'^T l' = \begin{pmatrix} A^T l' \\ a_4^T l' \end{pmatrix} \quad \pi'' = P''^T l'' = \begin{pmatrix} B^T l'' \\ b_4^T l'' \end{pmatrix}$$

$M = [\pi, \pi', \pi'']$ is rank 2

e.g. π is part of bundle formed by π' and π''

$$M = [m_1, m_2, m_3] = \begin{bmatrix} 1 & A^T l' & B^T l'' \\ 0 & a_4^T l' & b_4^T l'' \end{bmatrix}$$

$$m_1 = \alpha m_2 + \beta m_3 \quad \Rightarrow \quad 0 = \underbrace{\alpha a_4^T l'}_{kb_4^T l''} + \underbrace{\beta b_4^T l''}_{-ka_4^T l'}$$

$$\Rightarrow 1 = (b_4^T l'') A^T l' - (a_4^T l') B^T l''$$

Incidence Relation

$$l = (\mathbf{b}_4^T \mathbf{l}''') \mathbf{A}^T \mathbf{l}' - (\mathbf{a}_4^T \mathbf{l}') \mathbf{B}^T \mathbf{l}''' = (\mathbf{l}'''^T \mathbf{b}_4) \mathbf{A}^T \mathbf{l}' - (\mathbf{l}'^T \mathbf{a}_4) \mathbf{B}^T \mathbf{l}'''$$

$$l_i = \mathbf{l}'''^T \mathbf{b}_4 \mathbf{a}_i^T \mathbf{l}' - \mathbf{l}'^T \mathbf{a}_4 \mathbf{b}_i^T \mathbf{l}''' = \mathbf{l}'^T (\mathbf{a}_i \mathbf{b}_4^T) \mathbf{l}''' - \mathbf{l}'^T (\mathbf{a}_4 \mathbf{b}_i^T) \mathbf{l}'''$$

$$l_i = \mathbf{l}'^T \mathbf{T}_i \mathbf{l}'''$$

$$\mathbf{T}_i = \mathbf{a}_i \mathbf{b}_4^T - \mathbf{a}_4 \mathbf{b}_i^T$$

The Trifocal Tensor = $\{T_1, T_2, T_3\}$

$$\boxed{l^T = l'^T [T_1, T_2, T_3] l''} = (l'^T T_1 l'', l'^T T_2 l'', l'^T T_3 l'')$$

Only depends on image coordinates and is thus independent of 3D projective basis.

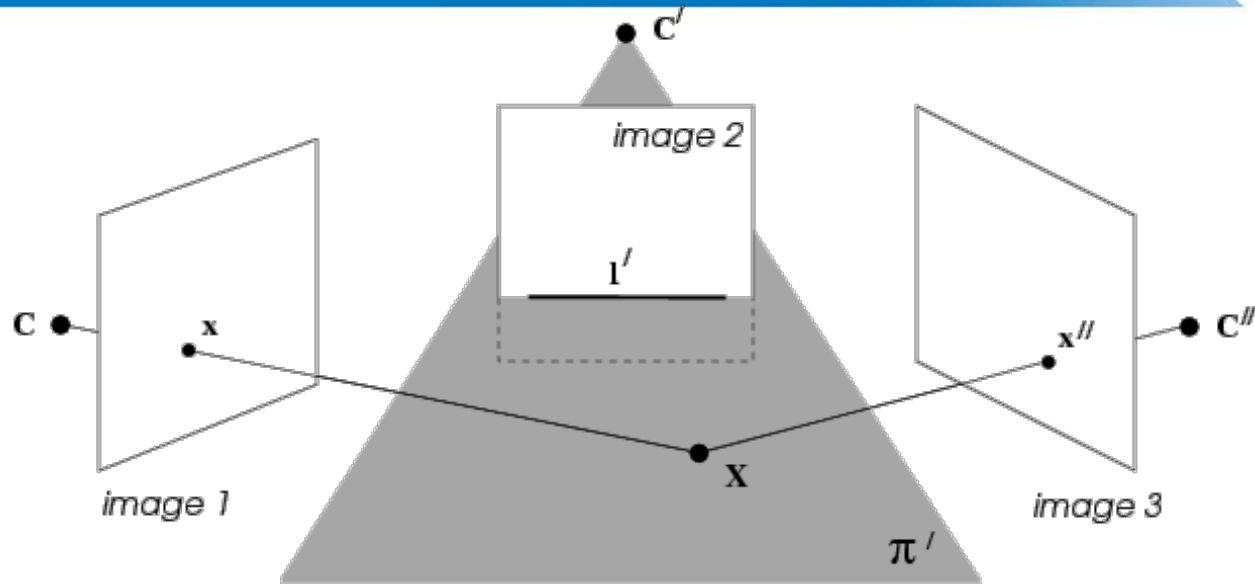
Also $l'^T = l^T [T'_i] l''$ and $l''^T = l^T [T''_i] l'$ but no simple relation.

General expression not as simple as $T_i = a_i b_4^T - a_4 b_i^T$

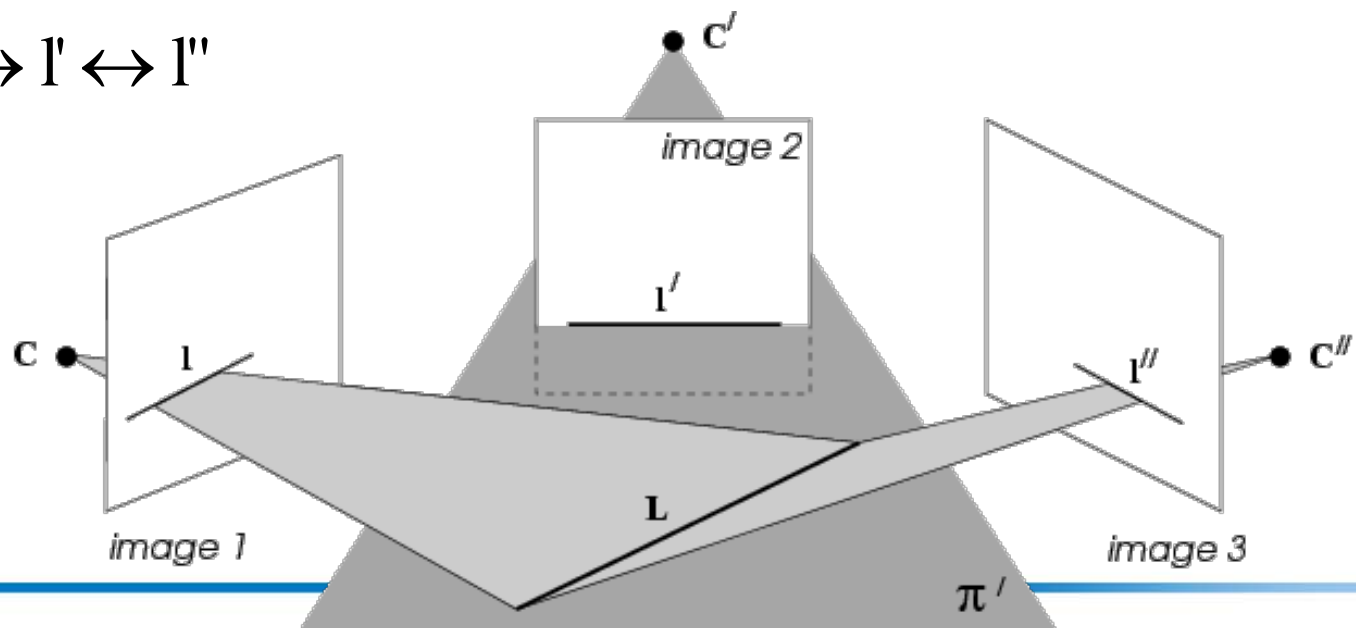
DOF T: $3 \times 3 \times 3 = 27$ elements, 26 up to scale

3-view relations: $11 \times 3 - 15 = 18$ dof

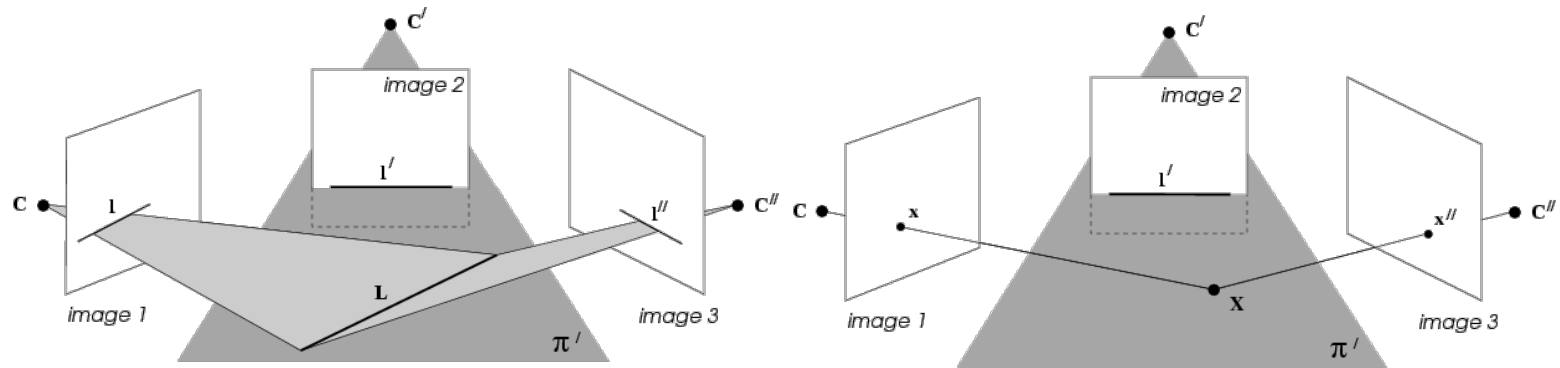
$8 (= 26 - 18)$ independent algebraic constraints on T
(compare to 1 for F, i.e. rank-2)



$$l \leftrightarrow l' \leftrightarrow l''$$



Homographies Induced by a Plane



$$l = H^T l''$$

$$x'' = Hx$$

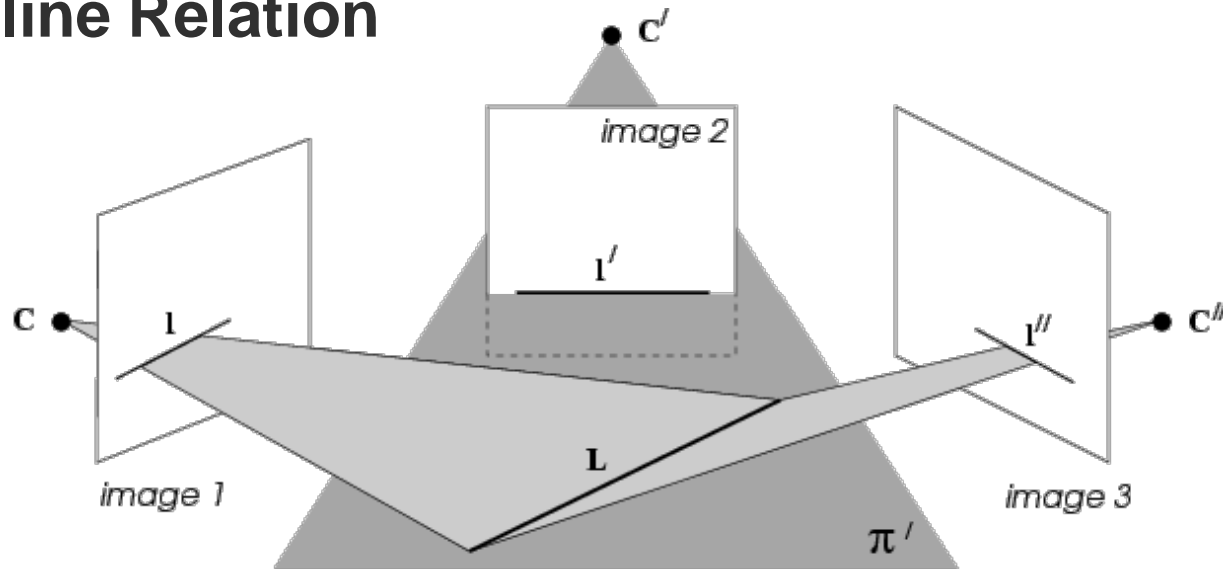
$$l_i = l'^T T_i l''$$

$$H = [h_1, h_2, h_3] \text{ with } h_i = T_i^T l'$$

$$x'' = H_{13}(l') x \quad H_{13}(l') = [T_1^T, T_2^T, T_3^T] l'$$

$$x' = H_{12}(l'') x \quad H_{12}(l'') = [T_1, T_2, T_3] l''$$

Line-line-line Relation



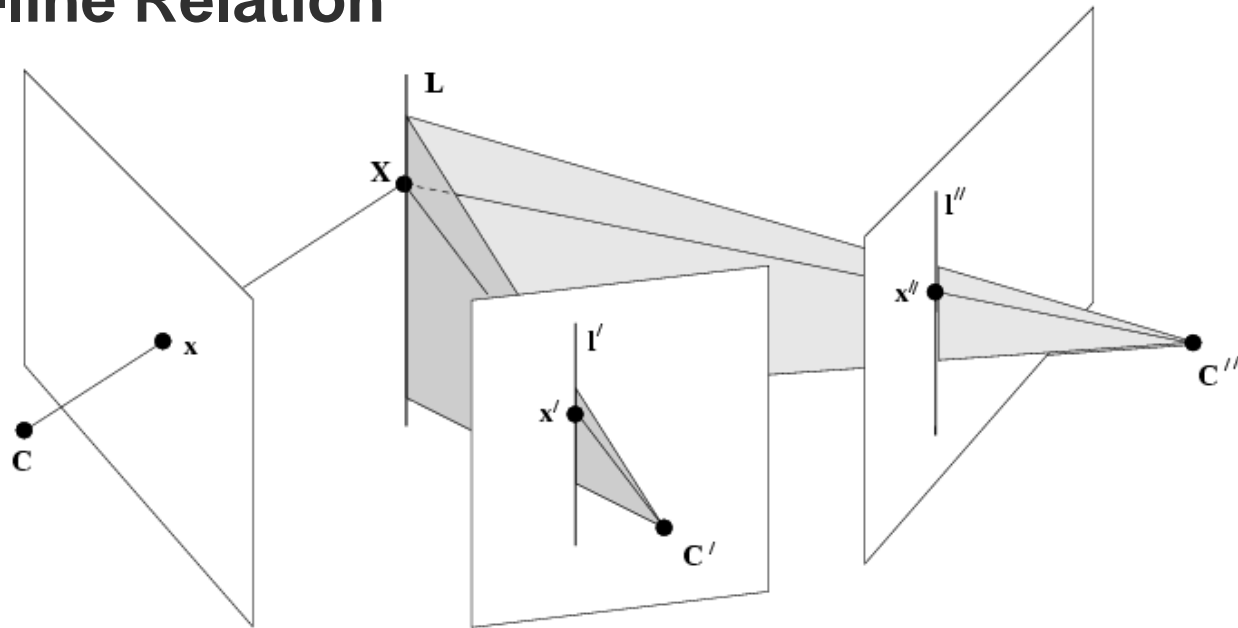
$$\mathbf{l}^\top = \mathbf{l}'^\top [\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3] \mathbf{l}'' \quad (\text{up to scale})$$

Eliminate scale factor: $(\mathbf{l}'^\top [\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3] \mathbf{l}'') [\mathbf{l}]_\times = \mathbf{0}^\top$

$$(\mathbf{l}'^\top [\mathbf{T}_i] \mathbf{l}'') [\mathbf{l}]_\times = \mathbf{0}^\top$$

$$(\mathbf{l}''^\top [\mathbf{T}_i^\top] \mathbf{l}') [\mathbf{l}]_\times = \mathbf{0}^\top$$

Point-line-line Relation

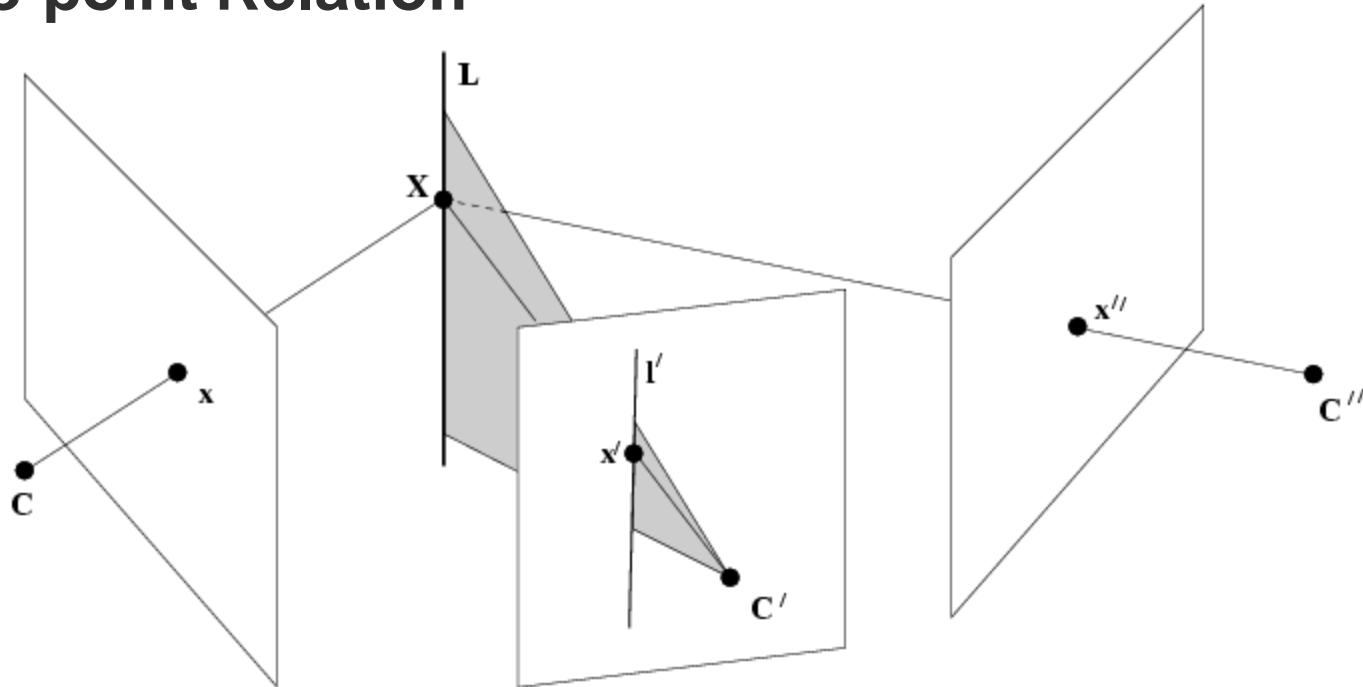


$$\mathbf{x}^\top \mathbf{l} = \sum_i x^i l_i = 0 \quad l_i = \mathbf{l}'^\top \mathbf{T}_i \mathbf{l}''^\top$$

$$\mathbf{l}'^\top \left(\sum_i x^i \mathbf{T}_i \right) \mathbf{l}''^\top = 0$$

$$\left(\sum_i x^i \mathbf{T}_i \right) \quad 3 \times 3 \text{ matrix}$$

Point-line-point Relation

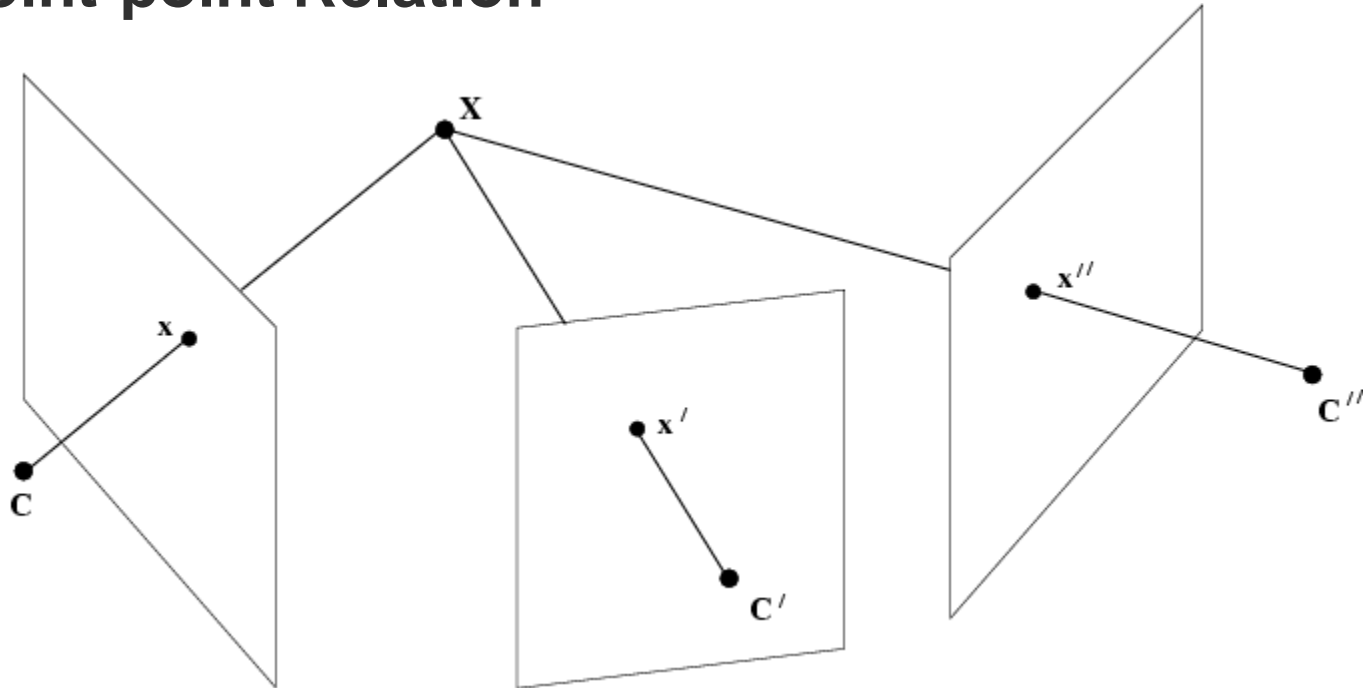


$$\mathbf{l}'^\top \left(\sum_i x^i \mathbf{T}_i \right) \mathbf{l}''^\top = 0$$

Note: valid for any line through x'' , e.g. $l'' = [x'']_x x''$ arbitrary

$$\mathbf{l}'^\top \left(\sum_i x^i \mathbf{T}_i \right) [x'']_x = \mathbf{0}^\top$$

Point-point-point Relation



$$l'^T \left(\sum_i x^i T_i \right) [x'']_{\times} = 0^T$$

Note: valid for any line through x' , e.g. $l' = [x']_{\times} X'_{\text{arbitrary}}$

$$[x']_{\times} \left(\sum_i x^i T_i \right) [x'']_{\times} = 0_{3 \times 3}$$

Overview: Incidence Relations

(i) Line–line–line correspondence

$$\mathbf{I}'^\top [\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3] \mathbf{I}'' = \mathbf{I}'^\top \quad \text{or} \quad (\mathbf{I}'^\top [\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3] \mathbf{I}'') [\mathbf{l}]_\times = \mathbf{0}^\top$$

(ii) Point–line–line correspondence

$$\mathbf{I}'^\top \left(\sum_i x^i \mathbf{T}_i \right) \mathbf{I}'' = \mathbf{0} \quad \text{for a correspondence } \mathbf{x} \leftrightarrow \mathbf{l}' \leftrightarrow \mathbf{l}''$$

(iii) Point–line–point correspondence

$$\mathbf{I}'^\top \left(\sum_i x^i \mathbf{T}_i \right) [\mathbf{x}'']_\times = \mathbf{0}^\top \quad \text{for a correspondence } \mathbf{x} \leftrightarrow \mathbf{l}' \leftrightarrow \mathbf{x}''$$

(iv) Point–point–line correspondence

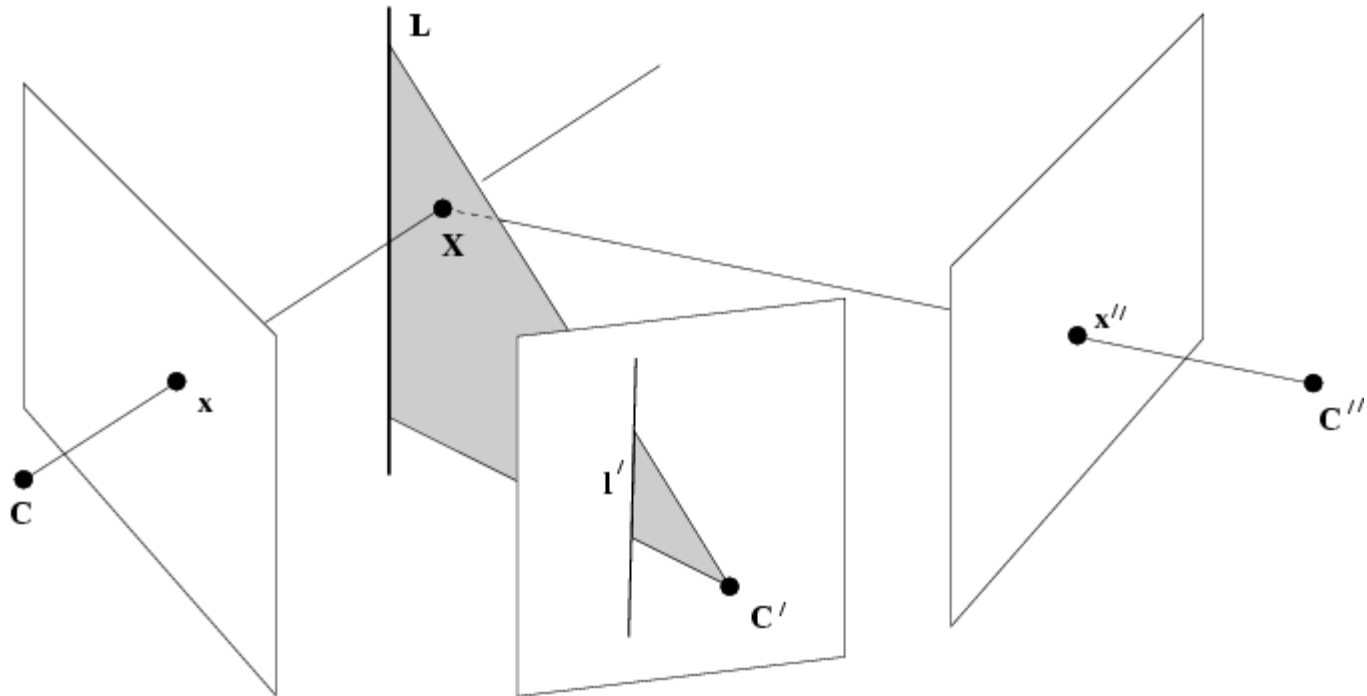
$$[\mathbf{x}']_\times \left(\sum_i x^i \mathbf{T}_i \right) \mathbf{I}'' = \mathbf{0} \quad \text{for a correspondence } \mathbf{x} \leftrightarrow \mathbf{x}' \leftrightarrow \mathbf{l}''$$

(v) Point–point–point correspondence

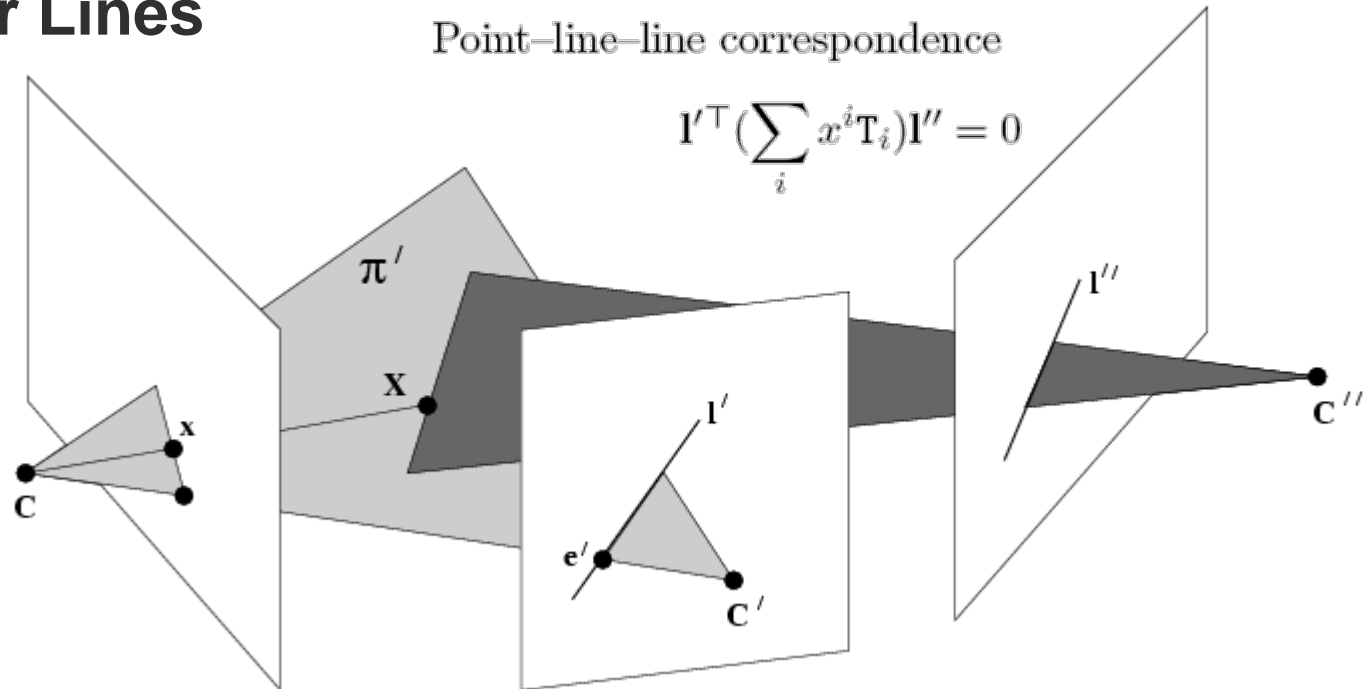
$$[\mathbf{x}']_\times \left(\sum_i x^i \mathbf{T}_i \right) [\mathbf{x}'']_\times = \mathbf{0}_{3 \times 3}$$

Non-incident Configuration

Incidence in image does not guarantee incidence in space



Epipolar Lines



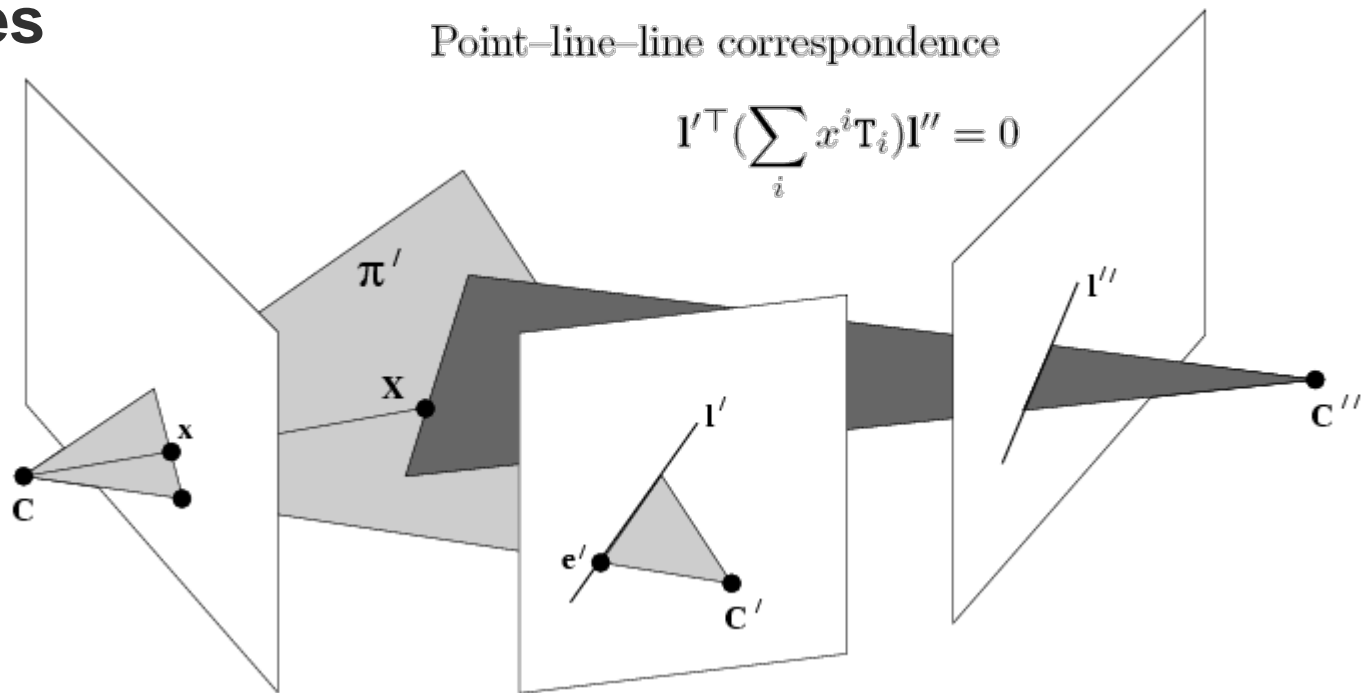
if l' is epipolar line, then satisfied for arbitrary l''

$$\mathbf{l}'^\top \left(\sum_i x^i \mathbf{T}_i \right) = \mathbf{0}^\top$$

inversely, $\left(\sum_i x^i \mathbf{T}_i \right) \mathbf{l}'' = \mathbf{0}$

epipolar lines are right and left null-space of $\left(\sum_i x^i \mathbf{T}_i \right)$.

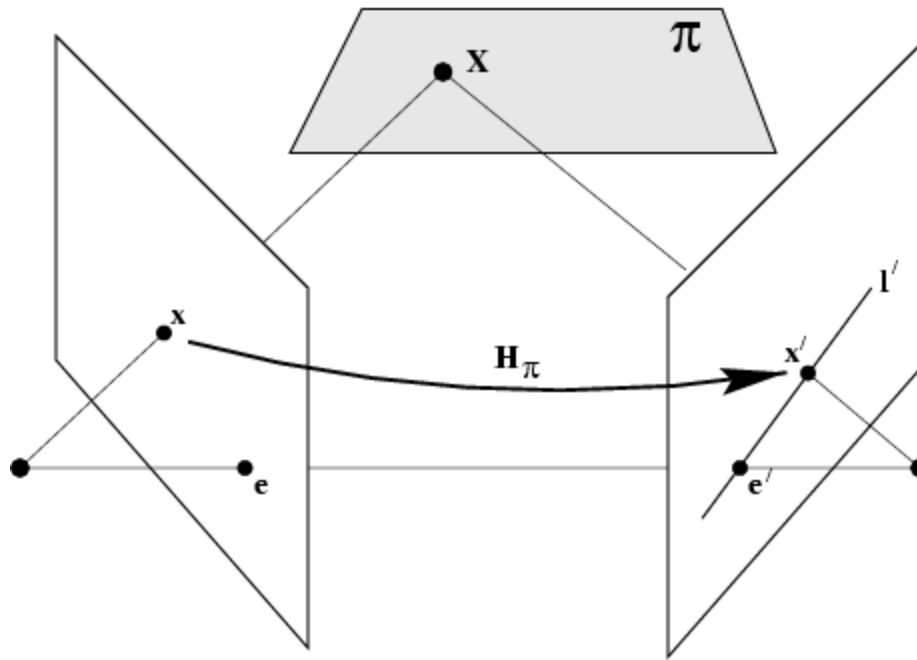
Epipoles



With points $(1, 0, 0)^T$, $(0, 1, 0)^T$ and $(0, 0, 1)^T$
 $(\sum_i x^i T_i)$ becomes T_1 , T_2 and T_3 respectively

Epipoles are intersection of right resp. left null-space of
 $(e = P'C$ and $e'' = P''C)$ T_1 , T_2 and T_3

The Fundamental Matrix F – Geometric Derivation

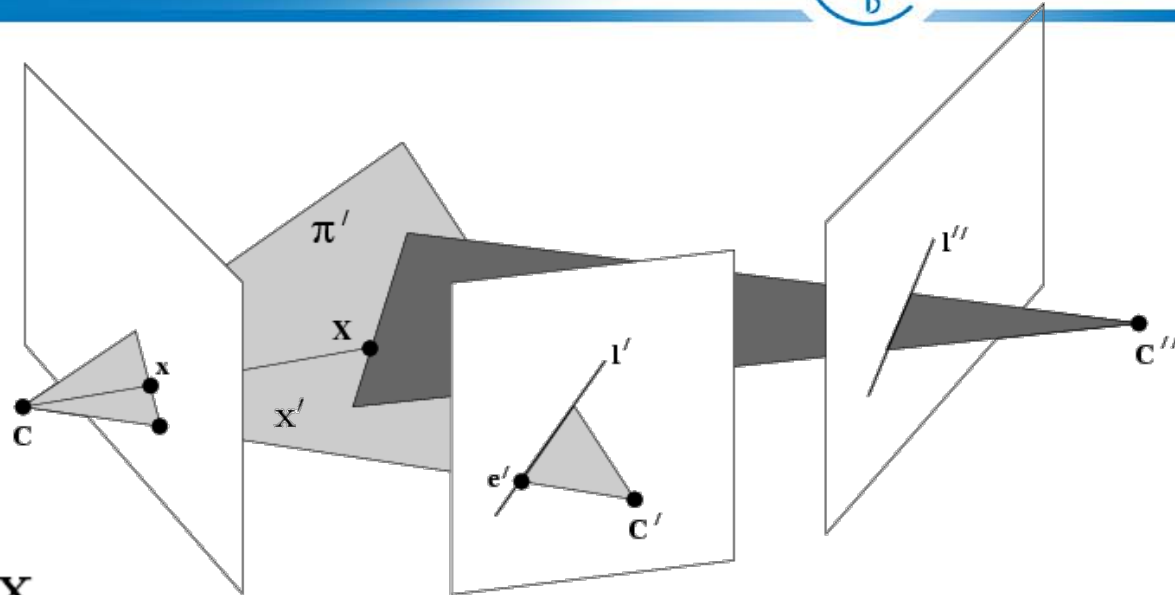


$$x' = H_{\pi} x$$

$$l' = e' \times x' = [e']_{\times} H_{\pi} x = Fx$$

mapping from 2-D to 1-D family (rank 2)

Extracting F



$$\mathbf{x}' = ([\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3] \mathbf{l}'') \mathbf{x}$$

$$\mathbf{l}' = \underbrace{[\mathbf{e}']_{\times}}_{\mathbf{F}_{21}} ([\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3] \mathbf{l}'') \mathbf{x}$$

good choice for \mathbf{l}'' is \mathbf{e}'' (orthogonal to the right null space of \mathbf{T}_i , i.e. $\mathbf{e}'' \mathbf{x} \mathbf{b}_i^T$)

$$\mathbf{F}_{21} = [\mathbf{e}']_{\times} [\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3] \mathbf{e}''$$

$$\mathbf{F}_{31} = [\mathbf{e}'']_{\times} [\mathbf{T}_1^T, \mathbf{T}_2^T, \mathbf{T}_3^T] \mathbf{e}'$$

Computing P, P', P''

$$P = [I \mid \mathbf{0}]$$

$$P' = [[T_1, T_2, T_3]e'' \mid e']$$

~~$$P'' = [[T_1^\top, T_2^\top, T_3^\top]e' \mid e'']$$~~

$\{P, P'\}$ and $\{P, P''\}$ ok, but not $\{P, P', P''\}$

$$P'' = [[T_1^\top, T_2^\top, T_3^\top]e' + e''\mathbf{v}^\top \mid \lambda e'']$$

specifically, (no derivation)

$$P'' = [(e''e''^\top - I)[T_1^\top, T_2^\top, T_3^\top]e' \mid e'']$$

Projective Ambiguity of Cameras Given F

$$P = [I \mid 0], P' = [A \mid a]$$

$$\tilde{P} = [I \mid 0], \tilde{P}' = [\tilde{A} \mid \tilde{a}]$$

$$F = [a]_{\times} A = [\tilde{a}]_{\times} \tilde{A}$$

lemma:

$$\tilde{a} = ka$$

$$\tilde{A} = k^{-1}(A + av^T)$$

Projective Ambiguity of Cameras Given F

$$P = [I | 0], P' = [A | a]$$

$$\tilde{P} = [I | 0], \tilde{P}' = [\tilde{A} | \tilde{a}]$$

$$F = [a]_{\times} A = [\tilde{a}]_{\times} \tilde{A}$$

Lemma:

$$\tilde{a} = ka$$

$$\tilde{A} = k^{-1}(A + av^T)$$

$$a^T F = a^T [a]_{\times} A = 0 = \tilde{a}^T F \stackrel{\text{rank 2}}{\implies} \tilde{a} = ka$$

Projective Ambiguity of Cameras Given F

$$P = [I \mid 0], P' = [A \mid a]$$

Lemma:

$$\tilde{P} = [I \mid 0], \tilde{P}' = [\tilde{A} \mid \tilde{a}]$$

$$\tilde{a} = ka$$

$$F = [a]_{\times} A = [\tilde{a}]_{\times} \tilde{A}$$

$$\tilde{A} = k^{-1}(A + av^T)$$

$$aF = a[a]_{\times} A = 0 = \tilde{a}F \stackrel{\text{rank } 2}{\implies} \tilde{a} = ka$$

$$[a]_{\times} A = [\tilde{a}]_{\times} \tilde{A} \implies [a]_{\times} (k\tilde{A} - A) = 0$$

$$\implies (k\tilde{A} - A) = av^T$$

Canonical Cameras Given F

Possible choice:

$$P = [I | 0] \quad P' = [[e']_{\times} F | e']$$

$$P = [I | 0], P' = [A | a]$$

$$\tilde{P} = [I | 0], \tilde{P}' = [\tilde{A} | \tilde{a}]$$

$$F = [a]_{\times} A = [\tilde{a}]_{\times} \tilde{A}$$

Lemma:

$$\tilde{a} = ka$$

$$\tilde{A} = k^{-1}(A + av^T)$$

Canonical representation:

$$P = [I | 0]$$

$$P' = [[e']_{\times} F + e'v^T | \lambda e']$$

Computing P, P', P''

$$P = [I \mid \mathbf{0}]$$

$$P' = [[T_1, T_2, T_3]e'' \mid e']$$

~~$$P'' = [[T_1^\top, T_2^\top, T_3^\top]e' \mid e'']$$~~

$\{P, P'\}$ and $\{P, P''\}$ ok, but not $\{P, P', P''\}$

$$P'' = [[T_1^\top, T_2^\top, T_3^\top]e' + e''\mathbf{v}^\top \mid \lambda e'']$$

specifically,

$$P'' = [(e''e''^\top - I)[T_1^\top, T_2^\top, T_3^\top]e' \mid e'']$$

Computing P, P', P''

$$P' = [I \mid 0]$$

$$P' = [[T_1, T_2, T_3]e'' \mid e']$$

$$P'' = [[T_1, T_2, T_3]e' + e''v^T \mid \lambda e'']$$

$$T_i = a_i b_4^T - a_4 b_i^T$$

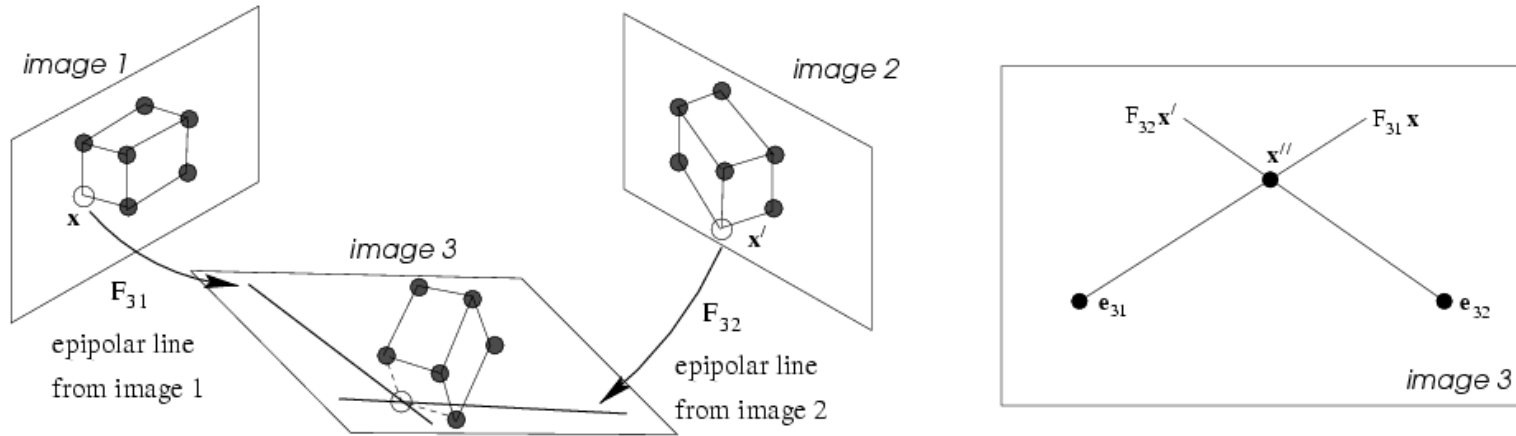
$$T_i = a_i e''^T - e' b_i^T$$

$$\rightarrow T_i = T_i e'' e''^T - e' b_i^T \Rightarrow e' b_i^T = T_i (e'' e''^T - I)$$

$$\Rightarrow b_i^T = (e'' e''^T - I) T_i e'$$

$$P'' = [(e'' e''^T - I)[T_1, T_2, T_3]e' \mid e'']$$

Transfer: Epipolar Transfer



$$x'' = (F_{31}\hat{X}) \times (F_{32}\hat{X}')$$

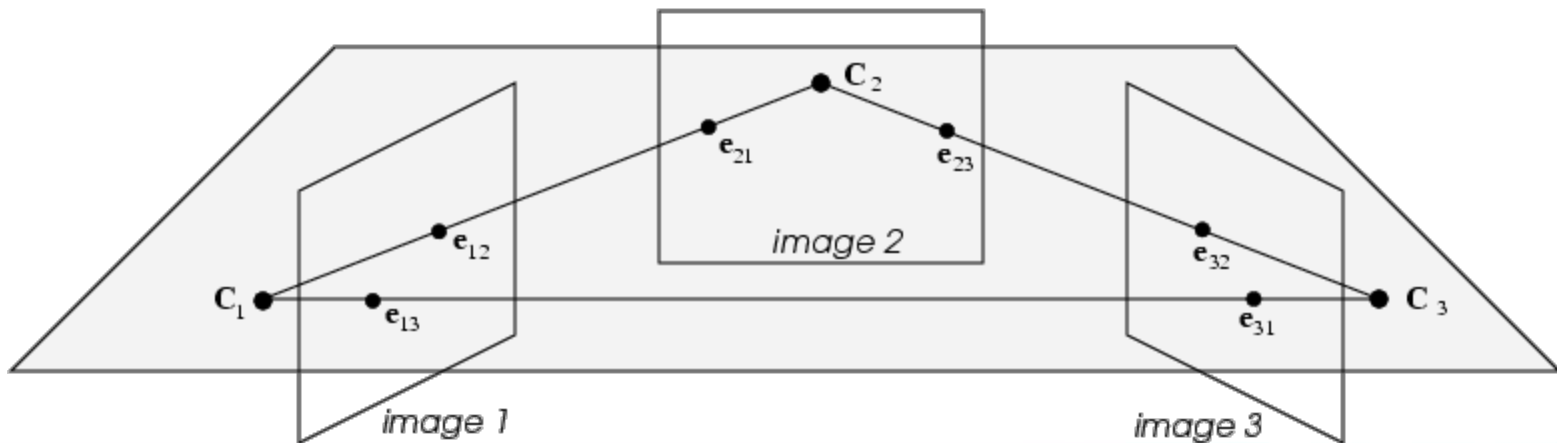


Image warping using $T(1,2,N)$

(Avidan and Shashua '97)



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)

Outline – Multiple View Geometry

- Introduction
 - Representation and motion of lines
- Geometric Introduction of the Trifocal Tensor
 - Line incidence
 - Homographies induced by a plane
 - Point and line Relationships induced by the trifocal tensor
- Extraction of projection & fundamental matrices
- Computing the trifocal tensor
 - Linear method (7-point)
 - Minimal method (6-point)
 - Geometric error minimization method
 - RANSAC method

Basic Equations

<u>Correspondence</u>	<u>Relation</u>	<u># lin. indep. Eq.</u>
Three points	$[\mathbf{x}']_{\times} \left(\sum_i x^i \mathbf{T}_i \right) [\mathbf{x}'']_{\times} = \mathbf{0}_{3 \times 3}$	4
Two points, one line	$\mathbf{I}'^{\top} \left(\sum_i x^i \mathbf{T}_i \right) [\mathbf{x}'']_{\times} = \mathbf{0}^{\top}$	2
One points, two line	$\mathbf{I}'^{\top} \left(\sum_i x^i \mathbf{T}_i \right) \mathbf{I}''^{\top} = 0$	1
Three lines	$(\mathbf{I}'^{\top} [\mathbf{T}_i] \mathbf{I}'') [\mathbf{1}]_{\times} = \mathbf{0}^{\top}$ $(\mathbf{I}''^{\top} [\mathbf{T}_i^{\top}] \mathbf{I}') [\mathbf{1}]_{\times} = \mathbf{0}^{\top}$	2

$\mathbf{A}t=0$ (26 equations)

$\min \|\mathbf{A}t\|$ with $\|t\|=1$ (more equations)

Normalized Linear Algorithm

At=0

Points

$$[\mathbf{x}']_{\times} \left(\sum_i x^i \mathbf{T}_i \right) [\mathbf{x}'']_{\times} = \mathbf{0}_{3 \times 3}$$

$$\sum_{i=1}^3 (x'_k x''_h T_i(j, l) + x'_j x''_l T_i(k, h)) x_i - \sum_{i=1}^3 (x'_j x''_h T_i(k, l) + x'_k x''_l T_i(j, h)) x_i = 0$$

$$\sum_{i=1}^3 (x'_k T_i(3, l) + x''_l T_i(k, 3)) x_i - \sum_{i=1}^3 (T_i(k, l) + x'_k x''_l T_i(3, 3)) x_i = 0 \quad k, l = 1, 2$$

Lines

$$l_i = \mathbf{I}'^{\top} \mathbf{T}_i \mathbf{I}''$$

or

$$\mathbf{I}'^{\top} \left(\sum_i x^i \mathbf{T}_i \right) \mathbf{I}''^{\top} = 0$$

Normalization: normalize image coordinates to ~ 1

Normalized Linear Algorithm

Objective

Given $n \geq 7$ image point correspondences across 3 images, or at least 13 lines, or a mixture of point and line corresp., compute the trifocal tensor.

Algorithm

- (i) Find transformation matrices H_1, H_2, H_3 to normalize 3 images
- (ii) Transform points with H and lines with H^{-1}
- (iii) Compute trifocal tensor T from $A_t=0$ (using SVD)
- (iv) Denormalize trifocal tensor with matrices H_1, H_2, H_3

$$T_i = H_2^{-1} \cdot \left(\sum_{j=1} H_1^T(i, j) \cdot \hat{T}_j \right) \cdot (H_3^{-1})^T$$

Internal Constraints

27	coefficients
1	free scale
18	parameters
8	internal consistency constraints

(not every $3 \times 3 \times 3$ tensor is a valid trifocal tensor!)

(constraints not easily expressed explicitly)

Trifocal Tensor satisfies all intrinsic constraints if it corresponds to three cameras $\{P, P', P''\}$

Minimal Algorithm

(Quan ECCV'94)

$(1, 0, 0, 0)^\top, (0, 1, 0, 0)^\top, (0, 0, 1, 0)^\top, (0, 0, 0, 1)^\top, (1, 1, 1, 1)^\top$ and $(X, Y, Z, W)^\top$

$(\bar{1}, 0, 0)^\top, (0, \bar{1}, 0)^\top, (0, 0, \bar{1})^\top, (1, 1, 1)^\top, (x_5^{(i)}, y_5^{(i)}, w_5^{(i)})^\top$, and $(x_6^{(i)}, y_6^{(i)}, w_6^{(i)})^\top$

$$\begin{bmatrix} 1 & 0 & 0 & 1 & x_5^{(i)} & x_6^{(i)} \\ 0 & 1 & 0 & 1 & y_5^{(i)} & y_6^{(i)} \\ 0 & 0 & 1 & 1 & w_5^{(i)} & w_6^{(i)} \end{bmatrix} = \begin{bmatrix} \alpha^{(i)} & 0 & 0 & \delta^{(i)} \\ 0 & \beta^{(i)} & 0 & \delta^{(i)} \\ 0 & 0 & \gamma^{(i)} & \delta^{(i)} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & X \\ 0 & 1 & 0 & 0 & 1 & Y \\ 0 & 0 & 1 & 0 & 1 & Z \\ 0 & 0 & 0 & 1 & 1 & W \end{bmatrix}$$

$$\begin{bmatrix} (-x_5^{(1)} y_6^{(1)} + x_5^{(1)} w_6^{(1)}) & (x_6^{(1)} y_5^{(1)} - y_5^{(1)} w_6^{(1)}) & (-x_6^{(1)} w_5^{(1)} + y_6^{(1)} w_5^{(1)}) \\ (-x_5^{(2)} y_6^{(2)} + x_5^{(2)} w_6^{(2)}) & (x_6^{(2)} y_5^{(2)} - y_5^{(2)} w_6^{(2)}) & (-x_6^{(2)} w_5^{(2)} + y_6^{(2)} w_5^{(2)}) \\ (-x_5^{(3)} y_6^{(3)} + x_5^{(3)} w_6^{(3)}) & (x_6^{(3)} y_5^{(3)} - y_5^{(3)} w_6^{(3)}) & (-x_6^{(3)} w_5^{(3)} + y_6^{(3)} w_5^{(3)}) \end{bmatrix} \begin{bmatrix} (-x_5^{(1)} w_6^{(1)} + y_5^{(1)} w_6^{(1)}) & (x_6^{(1)} y_5^{(1)} - y_5^{(1)} w_6^{(1)}) \\ (-x_5^{(2)} w_6^{(2)} + y_5^{(2)} w_6^{(2)}) & (x_6^{(2)} y_5^{(2)} - y_5^{(2)} w_6^{(2)}) \\ (-x_5^{(3)} w_6^{(3)} + y_5^{(3)} w_6^{(3)}) & (x_6^{(3)} y_5^{(3)} - y_5^{(3)} w_6^{(3)}) \end{bmatrix} \mathbf{t} = \mathbf{0}$$

$$\mathbf{t} = (WX - YZ, WY - YZ, WZ - YZ, XY - YZ, XZ - YZ)$$

$$\mathbf{t} = (t_1, t_2, t_3, t_4, t_5) \text{ then } t_1 t_2 t_5 - t_2 t_3 t_5 - t_2 t_4 t_5 = t_1 t_3 t_4 - t_2 t_3 t_4 - t_3 t_4 t_5$$

$$\bar{\mathbf{t}} = \mathbf{t}_1 + \alpha \mathbf{t}_2 \quad (\text{cubic equation in } \alpha)$$

$$\frac{X}{W} = \frac{t_4 - t_5}{t_2 - t_3} \quad \frac{Y}{W} = \frac{t_4}{t_1 - t_3} \quad \frac{Z}{W} = \frac{t_5}{t_1 - t_2} \quad (\alpha^{(i)}, \beta^{(i)}, \gamma^{(i)}, \delta^{(i)})$$

Maximum Likelihood Estimation

Data

$$\{x_i \leftrightarrow x'_i \leftrightarrow x''_i\}$$

Cost Function

$$\sum_i d(x_i, \hat{x}_i)^2 + d(x'_i, \hat{x}'_i)^2 + d(x''_i, \hat{x}''_i)^2$$

Parameterization

$$\hat{x}_i = [I | 0]X_i, \hat{x}'_i = P'X_i, \hat{x}''_i = P''X_i \quad (24 \text{ parameters} + 3N)$$

Also possibility to use Sampson error (24 parameters)

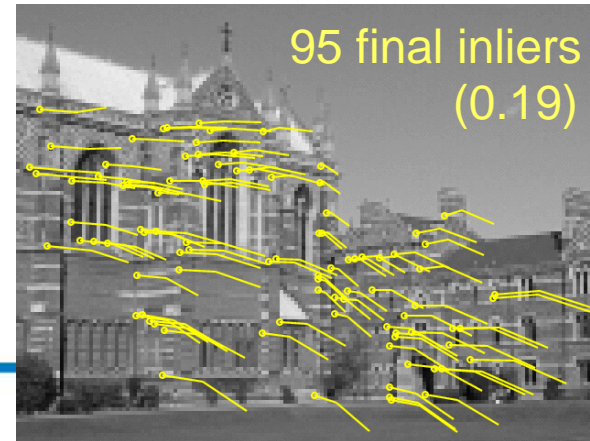
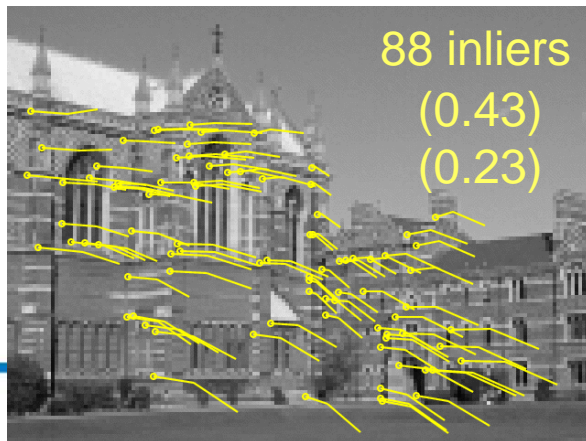
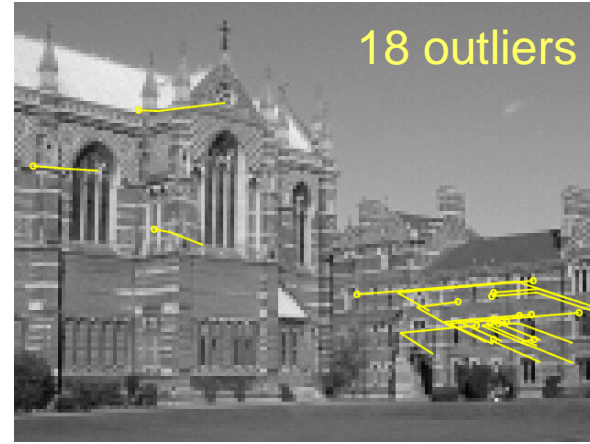
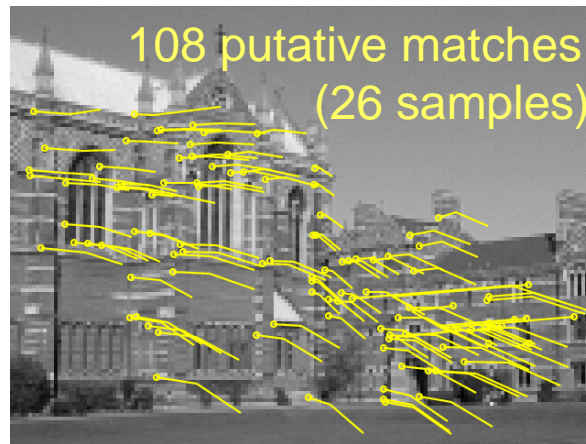
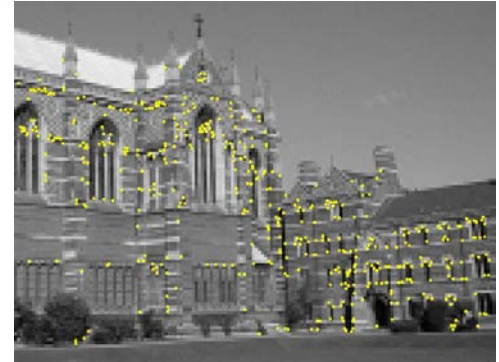
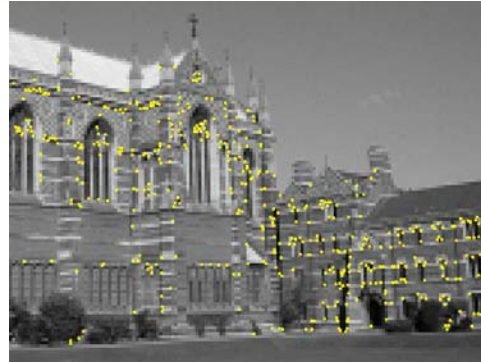
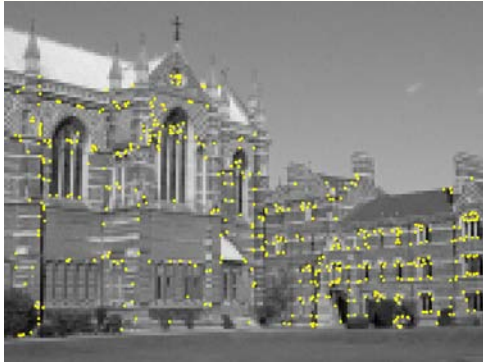
Automatic computation of T

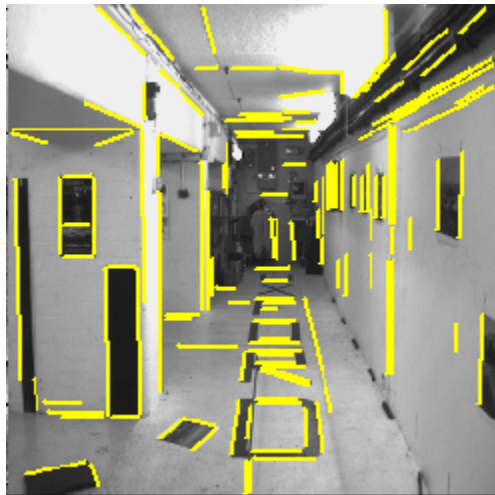
Objective

Compute the trifocal tensor between three images

Algorithm

- (i) Interest points: Compute interest points in each image
 - (ii) Putative correspondences: Compute interest correspondences (and F) between 1&2 and 2&3
 - (iii) RANSAC robust estimation: Repeat for N samples
 - (a) Select at random 6 correspondences and compute T
 - (b) Calculate the distance d_{\perp} for each putative match
 - (c) Compute the number of inliers consistent with T ($d_{\perp} < t$)
Choose T with most inliers
 - (iv) Optimal estimation: re-estimate T from all inliers by minimizing ML cost function with Levenberg-Marquardt
 - (v) Guided matching: Determine more matches using prediction by computed T
- Optionally iterate last two steps until convergence





additional line matches