

# CS4277 / CS5477

## 3D Computer Vision

### Lecture 9: Three-View Geometry from Points and/or Lines

Assoc. Prof. Lee Gim Hee

AY 2022/23

Semester 2

# Course Schedule

Week	Date	Topic	Assignments
1	11 Jan	2D and 1D projective geometry	<b>Assignment 0:</b> Getting started with Python (Ungraded)
2	18 Jan	3D projective geometry, Circular points and Absolute conic	
3	25 Jan	Rigid body motion and Robust homography estimation	
4	01 Feb	Camera models and calibration	<b>Assignment 1:</b> Metric rectification and robust homography (10%) <b>Due:</b> 2359hrs, 07 Feb
5	08 Feb	Single view metrology	<b>Assignment 2:</b> Affine 3D measurement from vanishing line and point (10%) <b>Due:</b> 2359hrs, 14 Feb
6	15 Feb	The Fundamental and Essential matrices	
-	22 Feb	Semester Break	No lecture
7	01 Mar	<b>Mid-term Quiz (20%)</b> Lecture: Generalized cameras	<b>In-person Quiz (LT 15, 1900hrs – 2000hrs)</b> Lecture: 2000hrs – 2130hrs
8	08 Mar	Absolute pose estimation from points or lines	
9	15 Mar	Three-view geometry from points and/or lines	
10	22 Mar	Structure-from-Motion (SfM) and bundle adjustment	<b>Assignment 3:</b> SfM and Bundle adjustment (10%) <b>Due:</b> 2359hrs, 28 Mar
11	29 Mar	Two-view and multi-view stereo	<b>Assignment 4:</b> Dense 3D model from multi-view stereo (10%) <b>Due:</b> 2359hrs, 04 Apr
12	05 Apr	3D Point Cloud Processing	
13	12 Apr	Neural Field Representations	

**Final Exam: 03 MAY 2023**

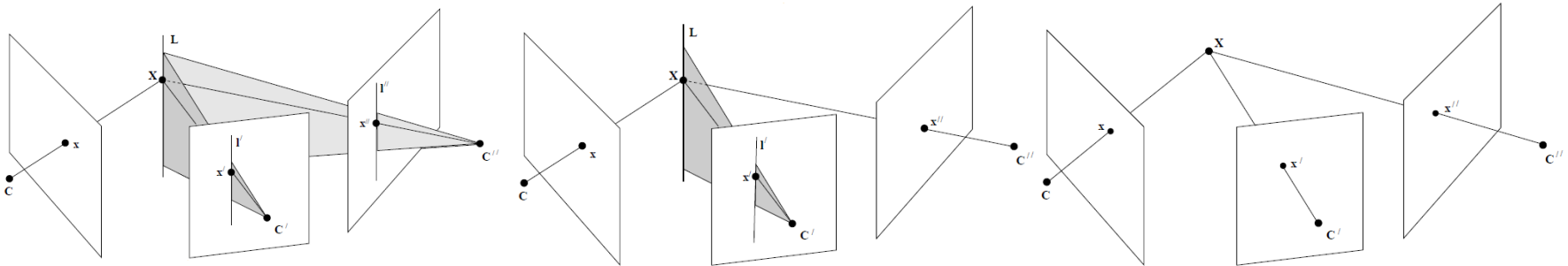
# Learning Outcomes

- Students should be able to:
  1. Derive the **trifocal tensor** constraint from point and/or line image correspondences of 3 views.
  2. Describe the **homography relations** between 3 views.
  3. Extract the 3-view **epipoles** and **epipolar lines** from the trifocal tensor.
  4. Decompose the trifocal tensor into the **camera** and **fundamental matrices** of 3 views.
  5. Compute the **trifocal tensor** from point and/line image correspondences of 3 views.

# Acknowledgements

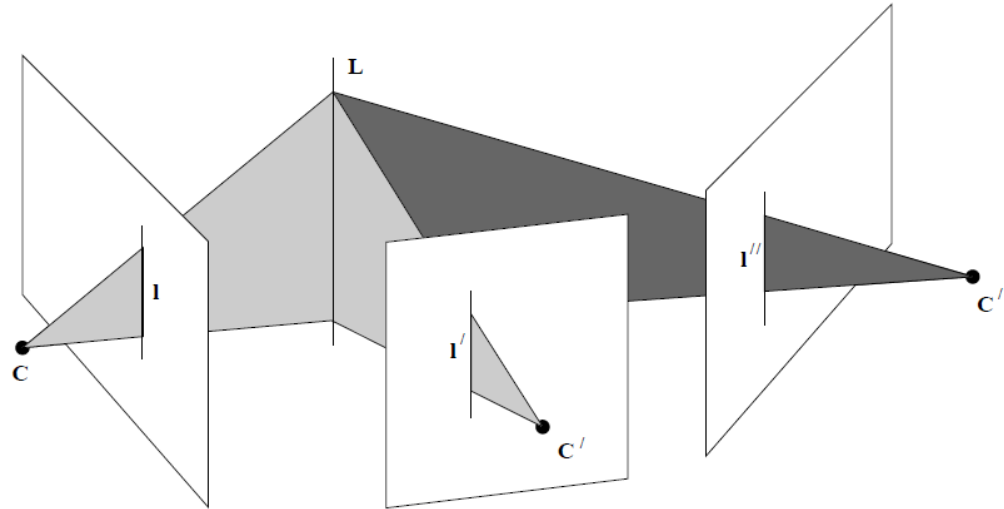
- A lot of slides and content of this lecture are adopted from:
  1. R. Hartley, and A. Zisserman: “Multiple view geometry in computer vision”, Chapter 15 and 16.
  2. Y. Ma, S. Soatto, J. Kosecka, S. S. Sastry, “ An invitation to 3-D vision”, Chapter 8.

# Three-View Geometry



- The **trifocal tensor** plays an analogous role in three views to that played by the fundamental matrix in two.
- It encapsulates all the (projective) geometric relations between three views that are **independent of scene structure**.

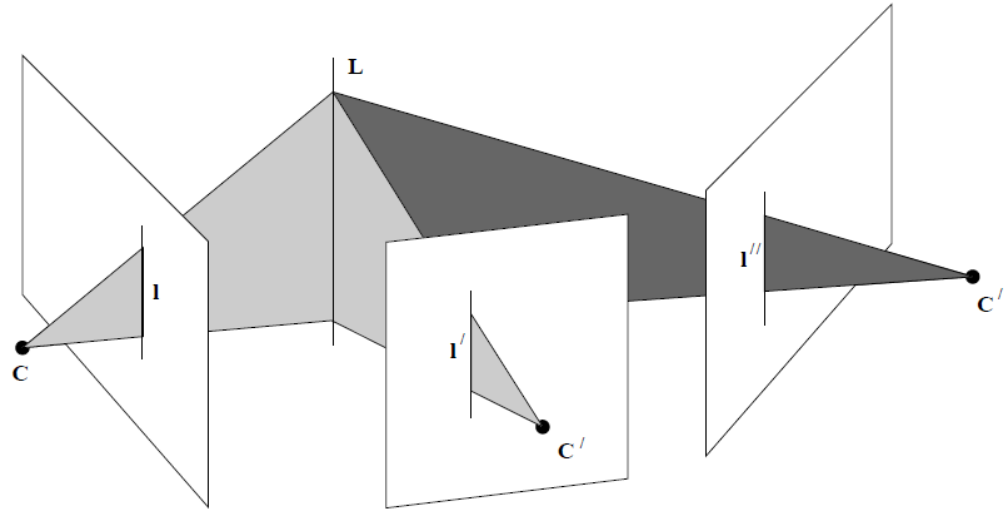
# The Geometric Basis for the Trifocal Tensor



- We use the incidence relationship of **three corresponding lines** to derive the trifocal tensor.
- Planes back-projected from the lines in each view must all **meet in a single 3D line**, i.e. the line that projects to the matched lines in the three images.

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

# The Geometric Basis for the Trifocal Tensor



- In general, three arbitrary planes in space **do not meet** in a single line.
- Hence, this **geometric incidence** condition provides a genuine constraint on sets of corresponding lines.
- We denote a set of corresponding lines as  $\mathbf{l}_i \leftrightarrow \mathbf{l}'_i \leftrightarrow \mathbf{l}''_i$ .

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

# The Geometric Basis for the Trifocal Tensor

- Let the **camera matrices** for the three views be  $P = [I \mid \mathbf{0}]$ , as usual, and  $P' = [A \mid \mathbf{a}_4]$ ,  $P'' = [B \mid \mathbf{b}_4]$ .
- $A$  and  $B$  are  $3 \times 3$  matrices, and the vectors  $\mathbf{a}_i$  and  $\mathbf{b}_i$  are the  $i$ -th columns of the respective camera matrices for  $i = 1, \dots, 4$ .
- $\mathbf{a}_4$  and  $\mathbf{b}_4$  are the **epipoles** in views two and three respectively, arising from the first camera.
- These epipoles are denoted by  $\mathbf{e}'$  and  $\mathbf{e}''$ , with  $\mathbf{e}' = P'\mathbf{C}$ ,  $\mathbf{e}'' = P''\mathbf{C}$ , where  $\mathbf{C}$  is the **first camera centre**.



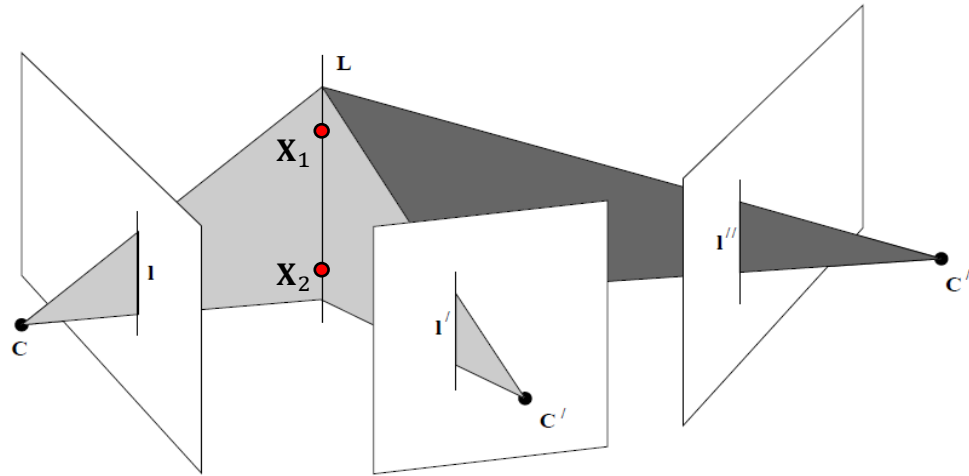
# The Geometric Basis for the Trifocal Tensor

- Each image line **back-projects** to a plane, i.e.

$$\boldsymbol{\pi} = \mathbf{P}^T \mathbf{l} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \boldsymbol{\pi}' = \mathbf{P}'^T \mathbf{l}' = \begin{pmatrix} \mathbf{A}^T \mathbf{l}' \\ \mathbf{a}_4^T \mathbf{l}' \end{pmatrix} \quad \boldsymbol{\pi}'' = \mathbf{P}''^T \mathbf{l}'' = \begin{pmatrix} \mathbf{B}^T \mathbf{l}'' \\ \mathbf{b}_4^T \mathbf{l}'' \end{pmatrix}.$$

- These three planes are **not independent** and must meet in a common line in 3-space.
- The **intersection constraint** can be expressed algebraically by the requirement that the  $4 \times 3$  matrix  $\mathbf{M} = [\boldsymbol{\pi}, \boldsymbol{\pi}', \boldsymbol{\pi}'']$  has **rank 2**.

# The Geometric Basis for the Trifocal Tensor



- Points on the line of intersection:  $\mathbf{X} = \alpha\mathbf{X}_1 + \beta\mathbf{X}_2$  .
- Such points **lie on** all three planes and so:

$$\pi^T \mathbf{X} = \pi'^T \mathbf{X} = \pi''^T \mathbf{X} = 0 \implies \mathbf{M}^T \mathbf{X} = \mathbf{0}.$$

- Consequently,  $\mathbf{M}$  has a **2-dimensional null-space** since  $\mathbf{M}^T \mathbf{X}_1 = \mathbf{0}$  and  $\mathbf{M}^T \mathbf{X}_2 = \mathbf{0}$ .

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

# The Geometric Basis for the Trifocal Tensor

- This intersection constraint **induces a relation** amongst the image lines  $\mathbf{l}, \mathbf{l}', \mathbf{l}''$ .
- Since the rank of  $M$  is 2, there is a **linear dependence** between its columns  $\mathbf{m}_i$ , i.e.

$$\mathbf{m}_1 = \alpha \mathbf{m}_2 + \beta \mathbf{m}_3, \text{ where}$$

$$M = [\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3] = \begin{bmatrix} 1 & A^T \mathbf{l}' & B^T \mathbf{l}'' \\ 0 & \mathbf{a}_4^T \mathbf{l}' & \mathbf{b}_4^T \mathbf{l}'' \end{bmatrix}.$$

# The Geometric Basis for the Trifocal Tensor

- Noting that the bottom left element of  $M$  is **zero**, it follows that  $\alpha = k(\mathbf{b}_4^\top \mathbf{l}'')$  and  $\beta = -k(\mathbf{a}_4^\top \mathbf{l}')$  for some scalar  $k$ .
- Applying this to the top 3-vectors of each column shows that (up to a homogeneous **scale factor**):

$$\begin{aligned}\mathbf{l} &= (\mathbf{b}_4^\top \mathbf{l}'')\mathbf{A}^\top \mathbf{l}' - (\mathbf{a}_4^\top \mathbf{l}')\mathbf{B}^\top \mathbf{l}'' \\ &= (\mathbf{l}''^\top \mathbf{b}_4)\mathbf{A}^\top \mathbf{l}' - (\mathbf{l}'^\top \mathbf{a}_4)\mathbf{B}^\top \mathbf{l}''.\end{aligned}$$

# The Geometric Basis for the Trifocal Tensor

- The *i*-th coordinate  $l_i$  of  $\mathbf{l}$  may therefore be written as:

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

- And introducing the notation:

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

- The incidence relation can be written:

$$l_i = \mathbf{l}'^T \mathbf{T}_i \mathbf{l}'', \quad i = 1, 2, 3.$$

# The Geometric Basis for the Trifocal Tensor

- The set of three matrices  $\{T_1, T_2, T_3\}$  constitute the *trifocal tensor* in matrix notation.
- We can now write the **incidence relation** as:

$$\mathbf{l}^T = \mathbf{l}'^T \underbrace{[T_1, T_2, T_3]}_{\text{represent the vector}} \mathbf{l}''$$

represent the vector



$$(\mathbf{l}'^T T_1 \mathbf{l}'', \mathbf{l}'^T T_2 \mathbf{l}'', \mathbf{l}'^T T_3 \mathbf{l}'')$$

# The Geometric Basis for the Trifocal Tensor

- Note that there exists **similar relations**:

$$\mathbf{l}'^T = \mathbf{l}^T [\mathbf{T}'_i] \mathbf{l}'' \quad \text{and} \quad \mathbf{l}''^T = \mathbf{l}^T [\mathbf{T}''_i] \mathbf{l}' .$$

- The three tensors  $[\mathbf{T}_i]$ ,  $[\mathbf{T}'_i]$  and  $[\mathbf{T}''_i]$  exist, but **are distinct**.
- Although all three tensors may be computed from any one of them, there is **no very simple relationship** between them.
- Hence, we will just consider only one of them.

# The Geometric Basis for the Trifocal Tensor

## Degrees of freedom:

- The trifocal tensor has only **18 independent** degrees of freedom defined up to a common scale.
- Each of 3 camera matrices has **11 degrees of freedom**, which makes 33 in total.
- However, **15 degrees of freedom** must be subtracted to account for the projective world frame.
- Thus leaving  $33 - 15 = 18$  degrees of freedom.



# Homographies Induced by a Plane

- A fundamental geometric property encoded in the trifocal tensor is the **homography between the first view and the third** induced by a line in the second image.
- A line in the second view defines (by backprojection) **a plane in 3-space**, and this plane induces a homography between the first and third views.

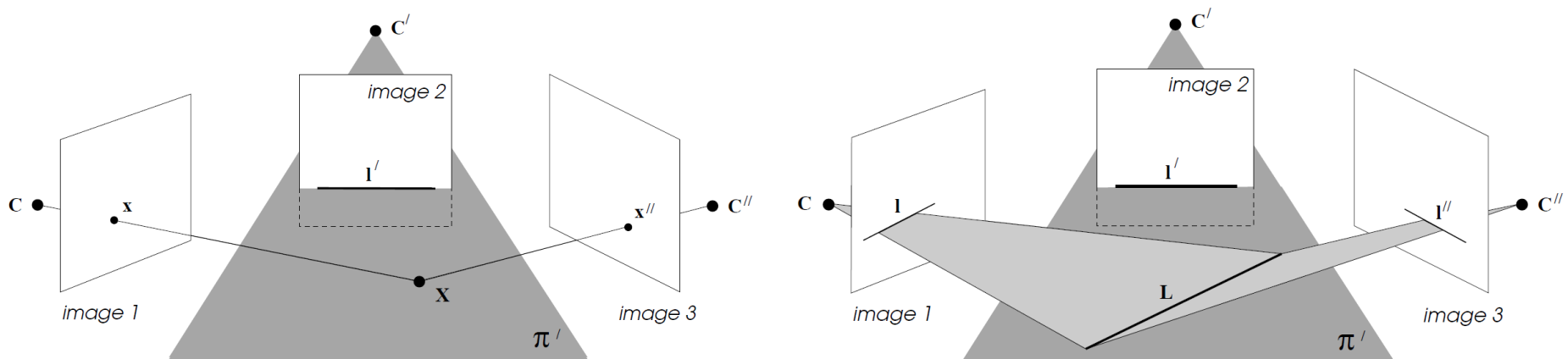
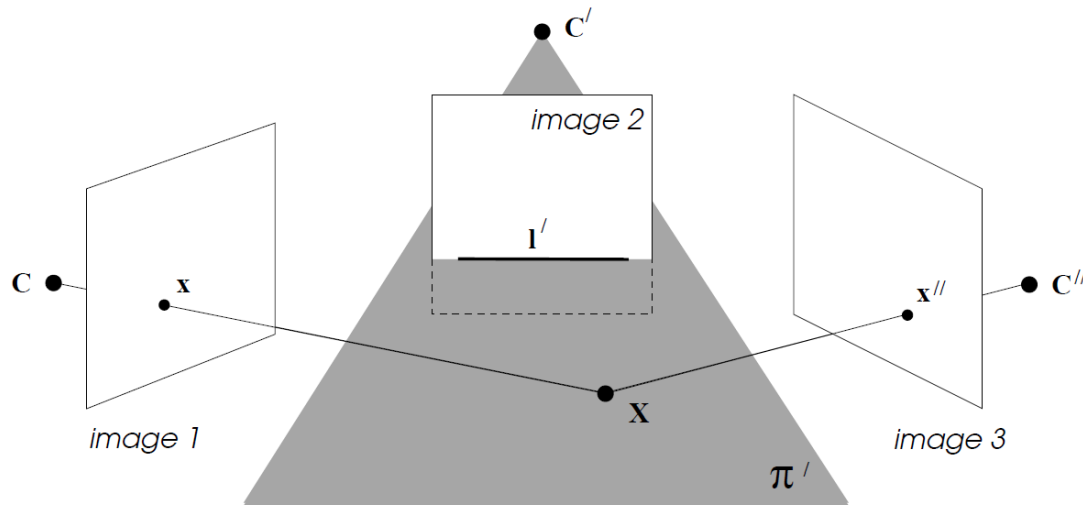


Image Source: R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision"

# Homographies Induced by a Plane

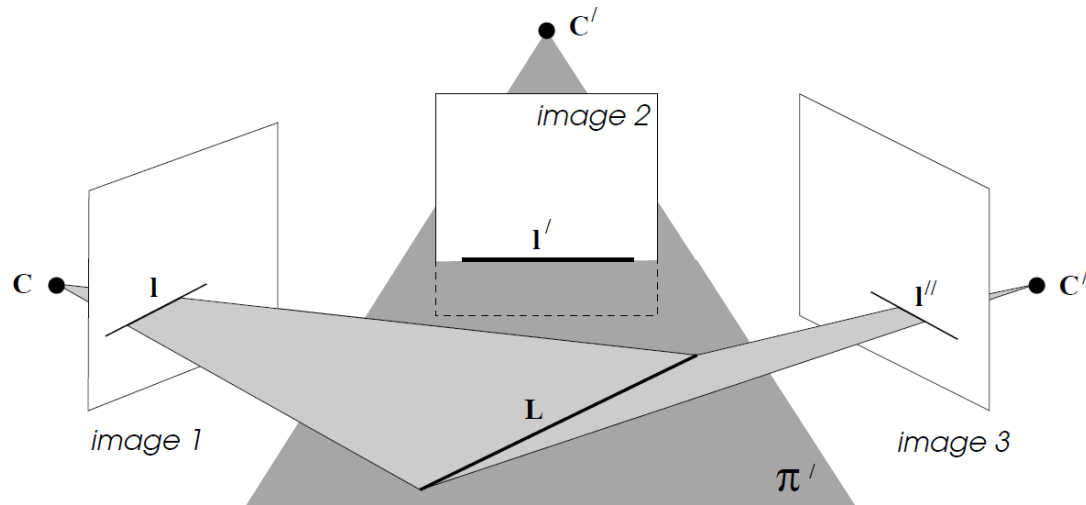


- The homography from the first to the third image induced by a line  $\mathbf{l}'$  in the second image is given by  $\mathbf{x}'' = H_{13}(\mathbf{l}') \mathbf{x}$ , where

$$H_{13}(\mathbf{l}') = [\mathbf{T}_1^T, \mathbf{T}_2^T, \mathbf{T}_3^T] \mathbf{l}'.$$

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

# Homographies Induced by a Plane

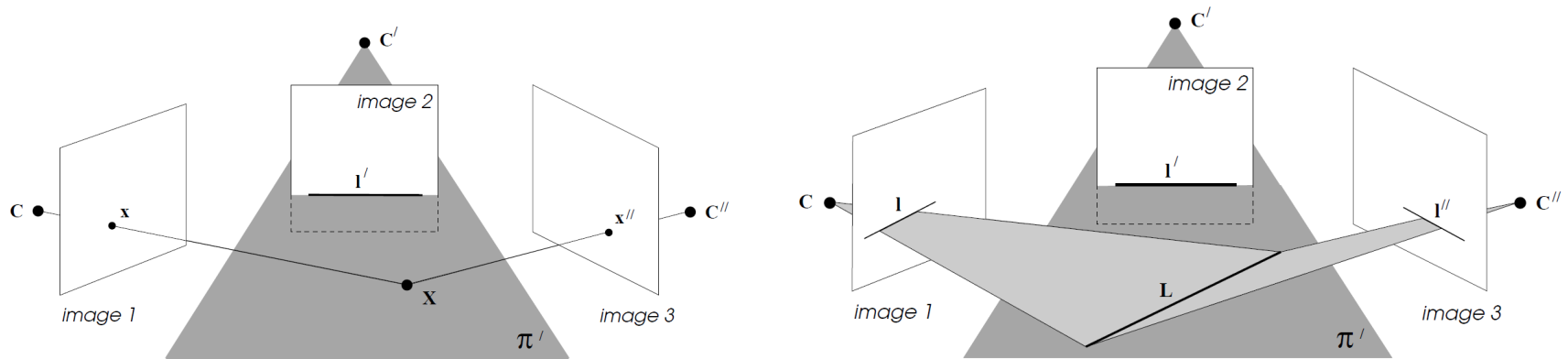


- Similarly, a line  $l''$  in the third image defines a homography  $x' = H_{12}(l'') x$  from the first to the second views, given by

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

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

# Homographies Induced by a Plane

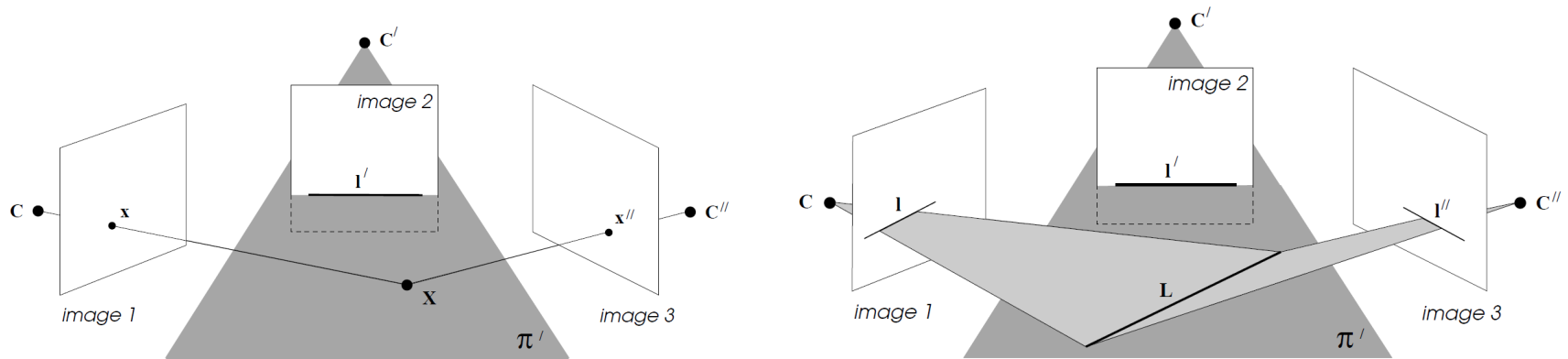


## Proof:

- The **homography map** between the first and third images may be written as  $\mathbf{x}'' = \mathbf{H}\mathbf{x}$  and  $\mathbf{l} = \mathbf{H}^T \mathbf{l}''$ , respectively.
- We saw earlier that the **line incidence** relationship is given by  $l_i = \mathbf{l}'^T \mathbf{T}_i \mathbf{l}''$ .

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

# Homographies Induced by a Plane



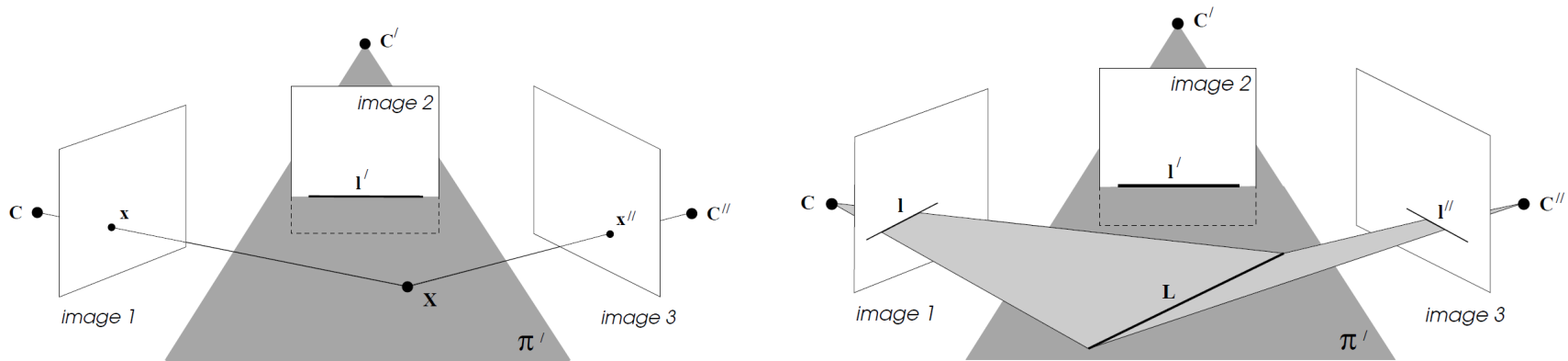
## Proof (cont.):

- Comparison of  $l_i = \mathbf{l}'^\top \mathbf{T}_i \mathbf{l}''$  and  $\mathbf{l} = \mathbf{H}^\top \mathbf{l}''$  shows that

$$\mathbf{H}_{13}(\mathbf{l}') = [\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3] \text{ with } \mathbf{h}_i = \mathbf{T}_i^\top \mathbf{l}'.$$

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

# Homographies Induced by a Plane



## Proof (cont.):

- We can also rewrite the homography as between the **first and second view**  $\mathbf{l}^\top = \mathbf{l}'^\top \mathbf{H}$ , where

$$\mathbf{H}_{12}(\mathbf{l}'') = [h_1, h_2, h_3] \text{ with } h_i = \mathbf{T}_i \mathbf{l}''$$

□

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

# Point and Line Incidence Relations

## Line-Line-Line Correspondences:

- Taking the cross product of the **incidence relation**

$\mathbf{l}^\top = \mathbf{l}'^\top [\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3] \mathbf{l}''$  to eliminate the unknown scale, we get:

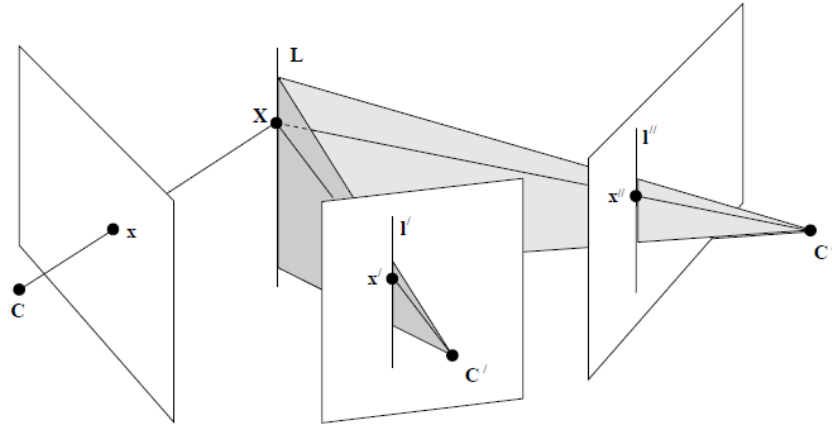
$$(\mathbf{l}'^\top [\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3] \mathbf{l}'') [\mathbf{l}]_\times = \mathbf{0}^\top,$$

- or more briefly  $(\mathbf{l}'^\top [\mathbf{T}_i] \mathbf{l}'') [\mathbf{l}]_\times = \mathbf{0}^\top$ .
- The **symmetry** between  $\mathbf{l}'$  and  $\mathbf{l}''$  means the following is true too:

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

# Point and Line Incidence Relations

## Point-Line-Line Correspondences:



- Now, **a point  $\mathbf{x}$  on the line  $\mathbf{l}$**  must satisfy  $\mathbf{x}^\top \mathbf{l} = \sum_i x^i l_i = 0$ ; since  $l_i = \mathbf{l}'^\top \mathbf{T}_i \mathbf{l}''$ , this may be written as:

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

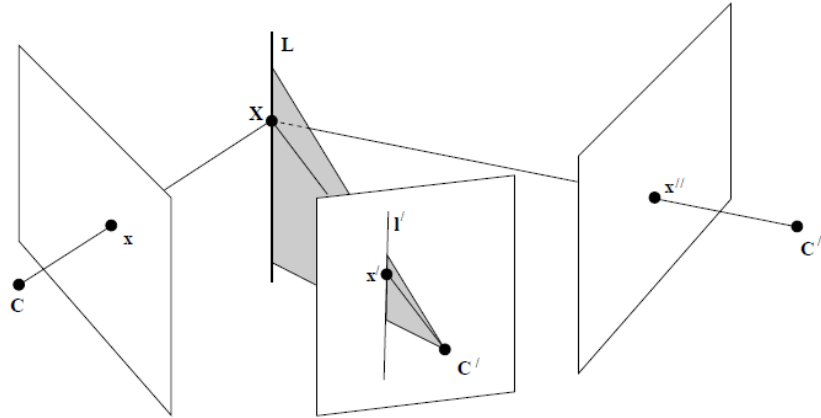
- Note that  $\sum_i x^i \mathbf{T}_i$  is simply a  $3 \times 3$  matrix.

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



# Point and Line Incidence Relations

## Point-Line-Point Correspondences:



- Consider a **point–line–point correspondence** so that

$$\mathbf{x}'' = H_{13}(\mathbf{l}') \mathbf{x} = [T_1^T \mathbf{l}', T_2^T \mathbf{l}', T_3^T \mathbf{l}'] \mathbf{x} = \left( \sum_i x^i T_i^T \right) \mathbf{l}'$$

- Which is valid for any line  $\mathbf{l}'$  passing through  $\mathbf{x}'$  in the second image.

# Point and Line Incidence Relations

## Point-Line-Point Correspondences:

- The homogeneous **scale factor** may be eliminated by (post-)multiplying the transpose of both sides by  $[\mathbf{x}]_{\times}$  to give

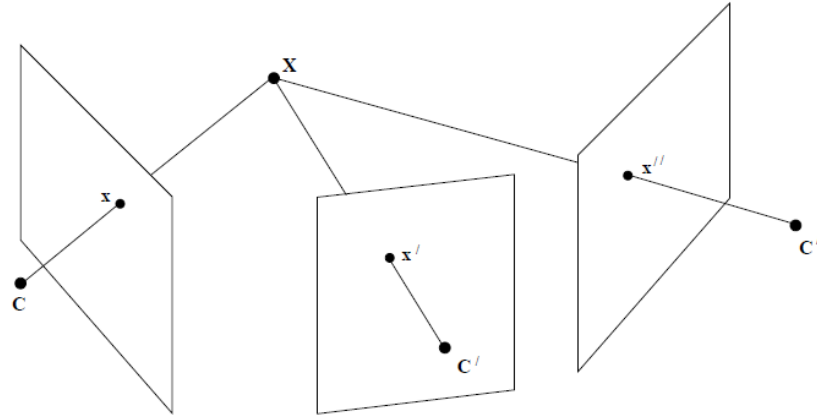
$$\mathbf{x}''^T [\mathbf{x}'']_{\times} = \mathbf{l}'^T \left( \sum_i x^i \mathbf{T}_i \right) [\mathbf{x}'']_{\times} = \mathbf{0}^T,$$

- A similar analysis may be undertaken with the roles of the second and third images, i.e. **point–point–line** correspondence:

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

# Point and Line Incidence Relations

## Point-Point-Point Correspondences:



- For a **3-point correspondence** as shown in figure, there is a relation:

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

# Point and Line Incidence Relations

## Proof:

- The line  $\mathbf{l}'$  in  $\mathbf{l}'^T \sum_i x^i \mathbf{T}_i [\mathbf{x}'']_{\times} = \mathbf{0}^T$  passes through  $\mathbf{x}'$  and so may be written as  $\mathbf{l}' = \mathbf{x}' \times \mathbf{y}' = [\mathbf{x}']_{\times} \mathbf{y}'$  for some point  $\mathbf{y}'$  on  $\mathbf{l}'$ .

- Consequently,

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

- Since  $\mathbf{l}'^T \sum_i x^i \mathbf{T}_i [\mathbf{x}'']_{\times} = \mathbf{0}^T$  is true for all lines  $\mathbf{l}'$  through  $\mathbf{x}'$ , and so **independent of  $\mathbf{y}'$** ; this implies the following is true:

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

□

# Point and Line Incidence Relations

(i) Line–line–line correspondence

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

(ii) Point–line–line correspondence

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

(iii) Point–line–point correspondence

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

(iv) Point–point–line correspondence

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

(v) Point–point–point correspondence

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

Table Source: Page 372, R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”

# Epipolar Lines

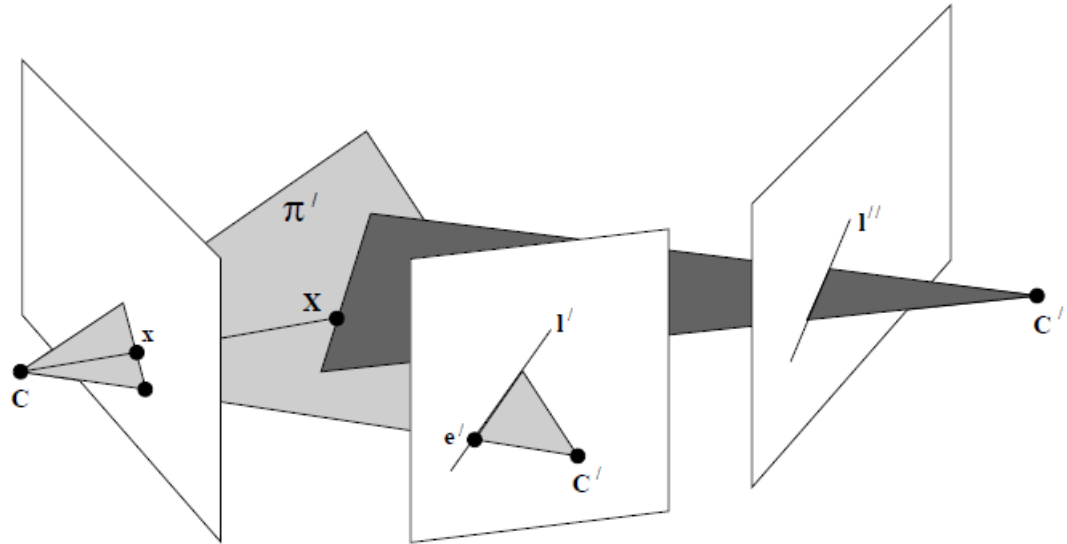
- If  $\mathbf{x}$  is a point and  $\mathbf{l}'$  or  $\mathbf{l}''$  are the corresponding **epipolar lines** in the second and third images, then

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

- Consequently, the epipolar lines  $\mathbf{l}'$  and  $\mathbf{l}''$  corresponding to  $\mathbf{x}$  may be computed as the **left** and **right null-vectors** of the matrix  $\sum_i x^i \mathbf{T}_i$ .

# Epipolar Lines

**Proof:**

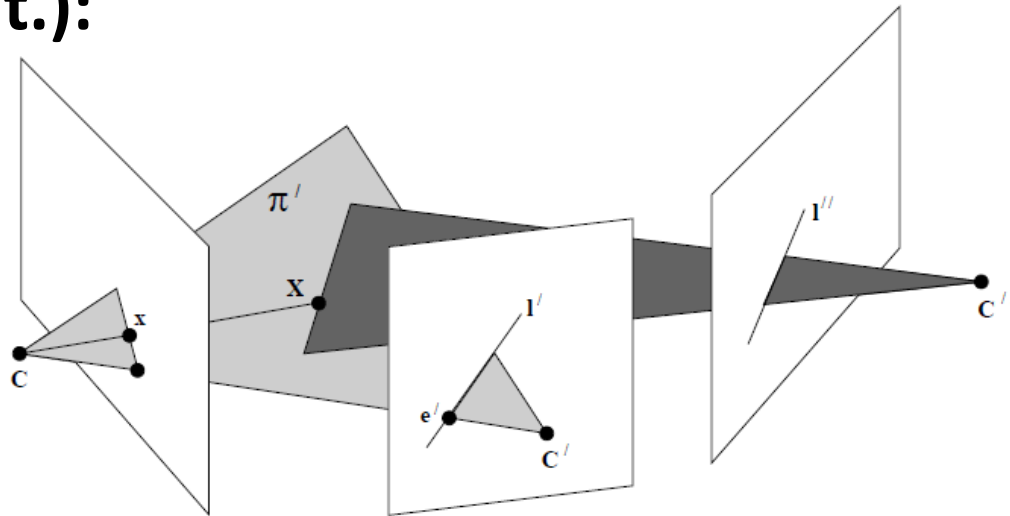


- A special case of a point–line–line correspondence occurs when the plane  $\pi'$  backprojected from  $l'$  is an **epipolar plane** with respect to the first two cameras.
- And hence passes through the camera centre  $C$  of the first camera.

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

# Epipolar Lines

Proof (cont.):

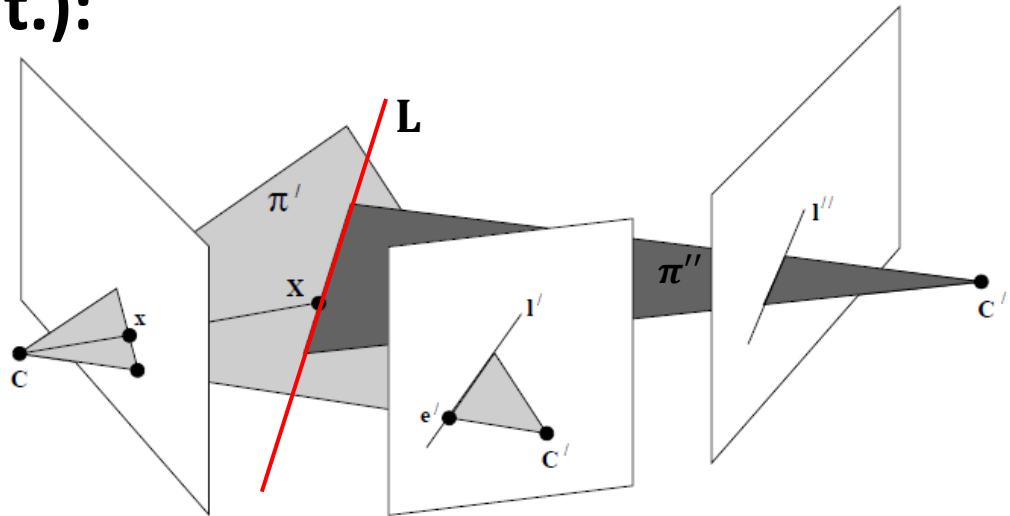


- Suppose  $\mathbf{X}$  is a point on the plane  $\pi'$ ; then the ray defined by  $\mathbf{X}$  and  $\mathbf{C}$  lies in this plane.
- And  $\mathbf{l}'$  is the **epipolar line** corresponding to the point  $\mathbf{x}$ , the image of  $\mathbf{X}$ .



# Epipolar Lines

**Proof (cont.):**



- The plane  $\pi''$  back-projected from a line  $l''$  in the third image will intersect the plane  $\pi'$  in a line  $L$ .
- Further, since the ray corresponding to  $x$  lies entirely in the plane  $\pi'$  **it must intersect** the line  $L$ .

# Epipolar Lines

## Proof (cont.):

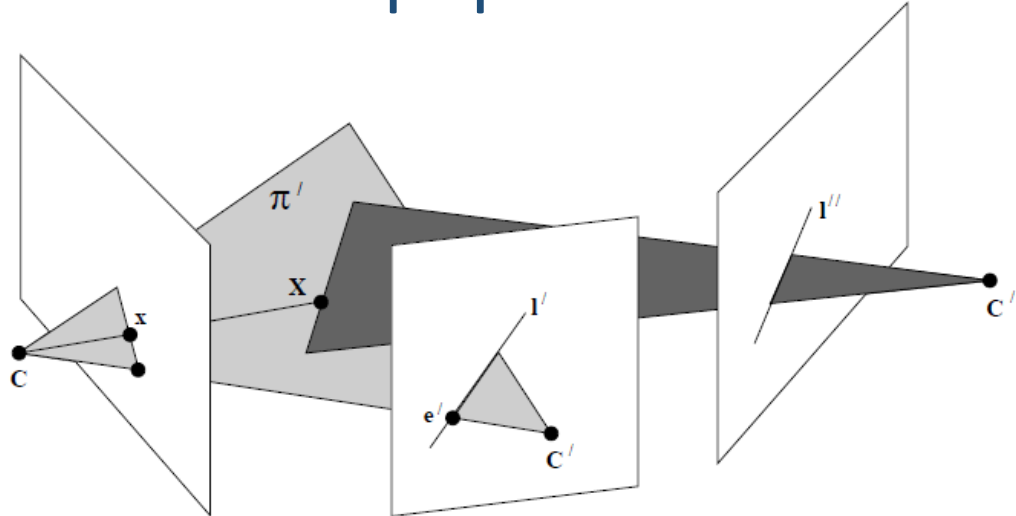
- This gives a **3-way intersection** between the ray and planes back-projected from point  $\mathbf{x}$  and lines  $\mathbf{l}'$  and  $\mathbf{l}''$ .
- So, they constitute a **point–line–line** correspondence, satisfying  $\mathbf{l}'^\top \sum_i x^i \mathbf{T}_i \mathbf{l}'' = 0$ .
- The important point now is that this is **true for any line**  $\mathbf{l}''$ , and it follows that  $\mathbf{l}'^\top \sum_i x^i \mathbf{T}_i = 0$ .
- The **same argument holds** with the roles of  $\mathbf{l}'$  and  $\mathbf{l}''$  reversed  $\Rightarrow \sum_i x^i \mathbf{T}_i \mathbf{l}'' = 0$ , where  $\mathbf{l}''$  is the epipolar line and  $\mathbf{l}'$  is any line in the second view.



# Epipole

- As the point  $\mathbf{x}$  varies, the corresponding epipolar lines vary, but **all epipolar lines** in one image pass through the epipole.
- Thus, this epipole can be computed as the **intersection of the epipolar lines** for varying values of  $\mathbf{x}$ .
- Three **convenient choices** of  $\mathbf{x}$  are the points represented by homogeneous coordinates  $(1, 0, 0)^T$ ,  $(0, 1, 0)^T$  and  $(0, 0, 1)^T$ .
- Hence,  $\sum_i x^i T_i$  equal to  $T_1$ ,  $T_2$  and  $T_3$  respectively for these three choices of  $\mathbf{x}$ .

# Epipole



- The epipole  $e'$  in the second image is the **common intersection** of the epipolar lines represented by the **left null-vectors** of the matrices  $T_i, i = 1, \dots, 3$ .
- Similarly, the epipole  $e''$  is the **common intersection** of lines represented by the **right null-vectors** of the  $T_i$ .

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

# Algebraic Properties of the $T_i$ Matrices

- Each matrix  $T_i$  has **rank 2**; this is evident from since  $T_i = \mathbf{a}_i \mathbf{e}''^T - \mathbf{e}' \mathbf{b}_i^T$  is the sum of two outer products.
- The **right null-vector** of  $T_i$  is  $\mathbf{l}_i'' = \mathbf{e}'' \times \mathbf{b}_i$ , and is the **epipolar line** in the third view for the point  $\mathbf{x} = (1, 0, 0)^T$ ,  $(0, 1, 0)^T$  or  $(0, 0, 1)^T$ , as  $i = 1, 2$  or  $3$ , respectively.

## Proof:

The epipolar line is given by  $\mathbf{l}'' = (P''C) \times (P''P^+ \mathbf{x})$

$$= \mathbf{e}'' \times [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4] \begin{bmatrix} \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{1 \times 3} \end{bmatrix} \mathbf{x} = \mathbf{e}'' \times [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3] \mathbf{x}$$

$$\Rightarrow \mathbf{l}_i'' = \mathbf{e}'' \times \mathbf{b}_i.$$

□

# Algebraic Properties of the $T_i$ Matrices

- The epipole  $e''$  is the **common intersection** of the epipolar lines  $l_i''$  for  $i = 1, 2, 3$ .
- The **left null-vector** of  $T_i$  is  $l_i' = e' \times a_i$ , and is the epipolar line in the second view for the point  $x = (1, 0, 0)^T$ ,  $(0, 1, 0)^T$  or  $(0, 0, 1)^T$ , as  $i = 1, 2$  or  $3$ , respectively.
- The epipole  $e'$  is the **common intersection** of the epipolar lines  $l_i'$  for  $i = 1, 2, 3$ , i.e. the null-vectors of:

$$e'^T [l_1', l_2', l_3'] = 0 \text{ and } e''^T [l_1'', l_2'', l_3''] = 0.$$

# Algebraic Properties of the $T_i$ Matrices

- The *sum* of the matrices  $M(\mathbf{x}) = \sum_i x^i T_i$  also has **rank 2**.
- The **right null-vector** of  $M(\mathbf{x})$  is the epipolar line  $\mathbf{l}''$  of  $\mathbf{x}$  in the third view, and its **left null-vector** is the epipolar line  $\mathbf{l}'$  of  $\mathbf{x}$  in the second view.

# Extracting the Fundamental Matrices

- We know earlier that a line  $\mathbf{l}''$  in the third view induces a **homography** from the first to the second view given by  $\mathbf{x}' = ([T_1, T_2, T_3]\mathbf{l}'') \mathbf{x}$ .
- The **epipolar line** corresponding to  $\mathbf{x}$  is then found by joining  $\mathbf{x}'$  to the epipole  $\mathbf{e}'$ .
- This gives  $\mathbf{l}' = [\mathbf{e}']_{\times} ([T_1, T_2, T_3]\mathbf{l}'') \mathbf{x}$ , which can be written as  $\mathbf{l}' = \mathbf{F}_{21}\mathbf{x}$ , i.e. the **fundamental matrix** is:

$$\mathbf{F}_{21} = [\mathbf{e}']_{\times} [T_1, T_2, T_3]\mathbf{l}''.$$



# Extracting the Fundamental Matrices

- This formula holds for any vector  $\mathbf{l}''$ , but it is important to choose  $\mathbf{l}''$  to **avoid the degenerate condition** where  $\mathbf{l}''$  lies in the null-space of any of the  $T_i$ .
- A **good choice is  $\mathbf{e}''$**  since it is perpendicular to the right null-space of each  $T_i$ .

## Remarks:

Assuming  $\mathbf{l}''$  is the **epipolar line**, it has to lie on the right-null space of each  $T_i$  since  $\sum_i x^i T_i \mathbf{l}'' = 0$ .

$\mathbf{e}''$  is **perpendicular** to the right-null space  $\mathbf{l}''$  since it lies on  $\mathbf{l}''$ , i.e.  $\mathbf{e}''^T \mathbf{l}'' = 0$ .

# Extracting the Fundamental Matrices

- This gives the formula

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

- A similar formula holds for

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

# Retrieving the Camera Matrices

- Since the trifocal tensor expresses a relationship between image entities only, it is **independent of** 3D projective transformations.
- Conversely, this implies that the camera matrices may be computed from the trifocal tensor **only up to a projective ambiguity**.

# Retrieving the Camera Matrices

- Due to the projective ambiguity, the **first camera** may be chosen as  $P = [I \mid \mathbf{0}]$ .
- Since  $F_{21}$  is known, we can make use of:

The **fundamental matrix** corresponding to a pair of camera matrices  $P = [I \mid \mathbf{0}]$  and  $P' = [A \mid \mathbf{a}]$  is equal to  $[\mathbf{a}]_{\times}A$ .

to derive the form of the **second camera** as

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

and the camera pair  $\{P, P'\}$  then has the fundamental matrix  $F_{21}$ .

# Retrieving the Camera Matrices

## A Fallacy:

- It might be thought that the third camera could be chosen in a similar manner as  $P'' = [[T_1^T, T_2^T, T_3^T]e' \mid e'']$ , but **this is incorrect**.
- To see this, suppose the camera pair  $\{P, P'\}$  is chosen and points  $\mathbf{X}_i$  reconstructed from their image correspondences  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ .

# Retrieving the Camera Matrices

## A Fallacy:

- Then  $\mathbf{X}_i$  are specified in the projective world frame defined by  $\{P, P'\}$ , and a consistent camera  $P''$  may be computed from  $\mathbf{X}_i \leftrightarrow \mathbf{x}_i''$ .
- Clearly,  $P''$  **depends on** the frame defined by  $\{P, P'\}$ .
- However, it is **not necessary** to explicitly reconstruct 3D structure, a consistent camera triplet can be recovered from the trifocal tensor directly.

# Retrieving the Camera Matrices

- We learned in Lecture 6 that a fundamental matrix can be **decomposed into**  $(P, P')$  and  $(\tilde{P}, \tilde{P}')$ , where

$$P = [I \mid \mathbf{0}] \text{ and } P' = [A \mid \mathbf{a}], \text{ and}$$
$$\tilde{P} = [I \mid \mathbf{0}] \text{ and } \tilde{P}' = [A + \mathbf{a}\mathbf{v}^T \mid \lambda\mathbf{a}]$$

for some vector  $\mathbf{v}$  and scalar  $\lambda$ .

- This implies the **more general form** of  $P'$  is:

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

# Retrieving the Camera Matrices

- Because of the projective ambiguity, we are **free to choose**  $P' = [ [T_1, T_2, T_3] \mathbf{e}'' \mid \mathbf{e}' ]$ , thus  $\mathbf{a}_i = T_i \mathbf{e}''$ .
- This choice fixes the projective world frame so that  $P''$  is now **defined uniquely** (up to scale).
- Substituting  $\mathbf{a}_i = T_i \mathbf{e}''$  into  $T_i = \mathbf{a}_i \mathbf{e}''^T - \mathbf{e}' \mathbf{b}_i^T$ , we get

$$T_i = T_i \mathbf{e}'' \mathbf{e}''^T - \mathbf{e}' \mathbf{b}_i^T$$

from which it follows that

$$\mathbf{e}' \mathbf{b}_i^T = T_i (\mathbf{e}'' \mathbf{e}''^T - \mathbf{I}).$$



# Retrieving the Camera Matrices

- Since the scale may be chosen such that

$$\|\mathbf{e}'\| = \mathbf{e}'^T \mathbf{e}' = 1,$$

we may multiply on the left by  $\mathbf{e}'^T$  and transpose to get

$$\mathbf{b}_i = (\mathbf{e}'' \mathbf{e}''^T - \mathbf{I}) \mathbf{T}_i^T \mathbf{e}'$$

so

$$\mathbf{P}'' = [(\mathbf{e}'' \mathbf{e}''^T - \mathbf{I})[\mathbf{T}_1^T, \mathbf{T}_2^T, \mathbf{T}_3^T] \mathbf{e}' | \mathbf{e}''] .$$

# Summary of F and P Retrieval from the Trifocal Tensor

**Given** the trifocal tensor written in matrix notation as  $[T_1, T_2, T_3]$ .

(i) **Retrieve the epipoles  $e', e''$**

Let  $u_i$  and  $v_i$  be the left and right null-vectors respectively of  $T_i$ , i.e.  $u_i^T T_i = 0^T$ ,  $T_i v_i = 0$ . Then the epipoles are obtained as the null-vectors to the following  $3 \times 3$  matrices:

$$e'^T [u_1, u_2, u_3] = 0 \text{ and } e''^T [v_1, v_2, v_3] = 0.$$

(ii) **Retrieve the fundamental matrices  $F_{21}, F_{31}$**

$$F_{21} = [e']_{\times} [T_1, T_2, T_3] e'' \text{ and } F_{31} = [e'']_{\times} [T_1^T, T_2^T, T_3^T] e'.$$

(iii) **Retrieve the camera matrices  $P', P''$  (with  $P = [I \mid 0]$ )**

Normalize the epipoles to unit norm. Then

$$P' = [[T_1, T_2, T_3] e'' \mid e'] \text{ and } P'' = [(e'' e''^T - I) [T_1^T, T_2^T, T_3^T] e' \mid e''].$$

Table Source: Page 375, R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision"

# The Trifocal Tensor and Tensor Notation

- We use the tensor notation to denote the **cumbersome three indices** in a tensor.
- **Image point:** homogeneous **row**  $\mathbf{x} = (x^1, x^2, x^3)^\top$ .
- **Image line:** homogeneous **column**  $\mathbf{l} = (l_1, l_2, l_3)$ .
- The  $ij$ -th entry of a matrix  $A$  is denoted by  $a_j^i$ , index  $i$  being the **contravariant (row)** index and  $j$  being the **covariant (column)** index.

# The Trifocal Tensor and Tensor Notation

- The indices repeated in the contravariant and covariant positions imply **summation over** the range  $(1, \dots, 3)$  of the index.

## Example:

The equation  $\mathbf{x}' = \mathbf{A}\mathbf{x}$  is equivalent to

$$x'^i = \sum_j a_j^i x^j,$$

which may be written as

$$x'^i = a_j^i x^j.$$

# The Trifocal Tensor and Tensor Notation

- The **trifocal tensor** can now be written as:

$$\mathbf{T}_i = \mathbf{a}_i \mathbf{b}_4^\top - \mathbf{a}_4 \mathbf{b}_i^\top \quad \Leftrightarrow \quad \mathcal{T}_i^{jk} = a_i^j b_4^k - a_4^j b_i^k.$$

- In tensor notation, the basic **incidence relation** becomes:

$$\mathbf{l}^\top = \mathbf{l}'^\top [\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3] \mathbf{l}''$$

$$\Leftrightarrow$$

$$l_i = l'_j l''_k \mathcal{T}_i^{jk} = \sum_{j,k} l'_j l''_k \mathcal{T}_i^{jk} = \sum_{j,k} l'_j \mathcal{T}_i^{jk} l''_k = l'_j \mathcal{T}_i^{jk} l''_k$$

# The Trifocal Tensor and Tensor Notation

- The **homography maps** becomes:

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

$$\Leftrightarrow$$

$$l_i = l'_j l''_k \mathcal{T}_i^{jk} = l''_k (l'_j \mathcal{T}_i^{jk}) = l''_k h_i^k \text{ where } h_i^k = l'_j \mathcal{T}_i^{jk} ,$$

$h_i^k$  are the elements of the homography matrix  $\mathbf{H}$ .

- This homography **maps points** between the first and third view as:

$$x''^k = h_i^k x^i .$$

# The Trifocal Tensor and Tensor Notation

**Definition.** The trifocal tensor  $\mathcal{T}$  is a valency 3 tensor  $\mathcal{T}_i^{jk}$  with two contravariant and one covariant indices. It is represented by a homogeneous  $3 \times 3 \times 3$  array (i.e. 27 elements). It has 18 degrees of freedom.

**Computation from camera matrices.** If the canonical  $3 \times 4$  camera matrices are

$$P = [I \mid 0], \quad P' = [a_j^i], \quad P'' = [b_j^i]$$

then

$$\mathcal{T}_i^{jk} = a_i^j b_4^k - a_4^j b_i^k.$$

See (17.12–p415) for computation from three general camera matrices.

**Line transfer from corresponding lines in the second and third views to the first.**

$$l_i = l'_j l''_k \mathcal{T}_i^{jk}$$

**Transfer by a homography.**

- (i) **Point transfer from first to third view via a plane in the second**

The contraction  $l'_j \mathcal{T}_i^{jk}$  is a homography mapping between the first and third views induced by a plane defined by the back-projection of the line  $l'$  in the second view.

$$x''^k = h_i^k x^i \quad \text{where} \quad h_i^k = l'_j \mathcal{T}_i^{jk}$$

- (ii) **Point transfer from first to second view via a plane in the third**

The contraction  $l''_k \mathcal{T}_i^{jk}$  is a homography mapping between the first and second views induced by a plane defined by the back-projection of the line  $l''$  in the third view.

$$x'^j = h_i^j x^i \quad \text{where} \quad h_i^j = l''_k \mathcal{T}_i^{jk}$$

Table Source: Page 377, R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”

# The Tensor $\epsilon_{rst}$ and $\epsilon^{rst}$

- The tensor  $\epsilon_{rst}$  is defined for  $r, s, t = 1, \dots, 3$  as follows:

$$\epsilon_{rst} = \begin{cases} 0 & \text{unless } r, s \text{ and } t \text{ are distinct} \\ +1 & \text{if } rst \text{ is an even permutation of } 123 \\ -1 & \text{if } rst \text{ is an odd permutation of } 123 \end{cases}$$

- The tensor  $\epsilon_{rst}$  and  $\epsilon^{rst}$  relate to the **cross product** of two vectors:

$$c_i = (\mathbf{a} \times \mathbf{b})_i = \epsilon_{ijk} a^j b^k \quad \text{and} \quad ([\mathbf{a}]_{\times})_{ik} = \epsilon_{ijk} a^j.$$

$$\text{Similarly, } ([\mathbf{v}]_{\times})^{ik} = \epsilon^{ijk} v_j.$$



# The Tensor $\epsilon_{rst}$ and $\epsilon^{rst}$

- The trifocal tensor **incidence relations** in  $\epsilon_{rst}$  and  $\epsilon^{rst}$ .

(i) Line–line–line correspondence

$$(l_r \epsilon^{ris}) l'_j l''_k \mathcal{T}_i^{jk} = 0^s$$

(ii) Point–line–line correspondence

$$x^i l'_j l''_k \mathcal{T}_i^{jk} = 0$$

(iii) Point–line–point correspondence

$$x^i l'_j (x''^k \epsilon_{kqs}) \mathcal{T}_i^{jq} = 0_s$$

(iv) Point–point–line correspondence

$$x^i (x'^j \epsilon_{jpr}) l''_k \mathcal{T}_i^{pk} = 0_r$$

(v) Point–point–point correspondence

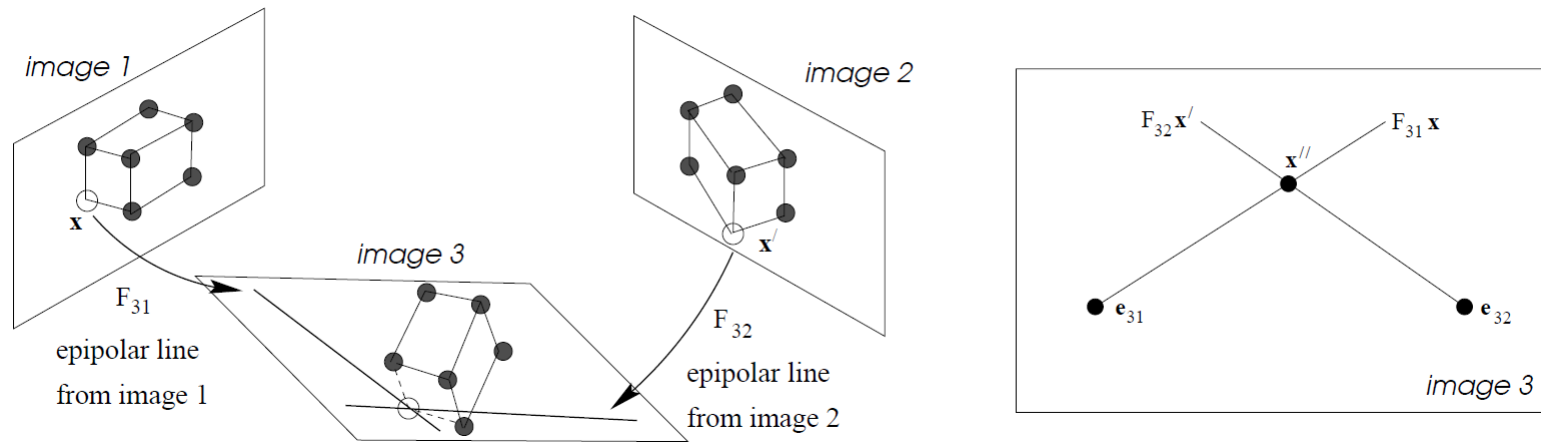
$$x^i (x'^j \epsilon_{jpr}) (x''^k \epsilon_{kqs}) \mathcal{T}_i^{pq} = 0_{rs}$$

Table Source: Page 378, R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”

# Point Transfer using Fundamental Matrices

- **Given:** The **three fundamental matrices**  $F_{21}$ ,  $F_{31}$  and  $F_{32}$  relating the three views, and the **point correspondence**  $\mathbf{x}$  and  $\mathbf{x}'$  in the first two views.
- **Find:** The corresponding point  $\mathbf{x}''$  in the **third image**.
- **Solution:** The image of  $\mathbf{X}$  in the third view may be computed by **intersecting the epipolar lines**  $F_{31}\mathbf{x}$  and  $F_{32}\mathbf{x}'$ .

# Point Transfer using Fundamental Matrices



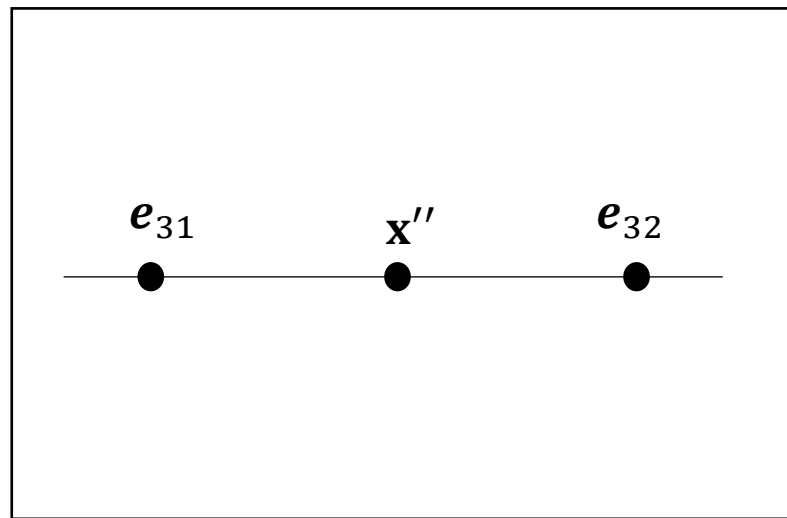
- The **intersection of the epipolar lines** gives

$$\mathbf{x}'' = (F_{31}\mathbf{x}) \times (F_{32}\mathbf{x}') .$$

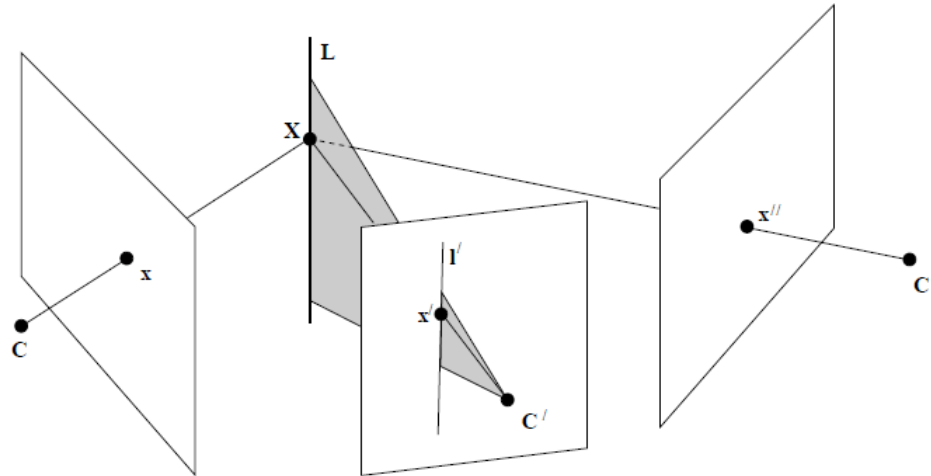
- This method of point transfer using the fundamental matrices will be called **epipolar transfer**.

# Degeneracy of Epipolar Transfer

- $\mathbf{x}''$  **cannot be** uniquely determined if it lies on the line that contains both epipoles  $e_{31}$  and  $e_{32}$ .



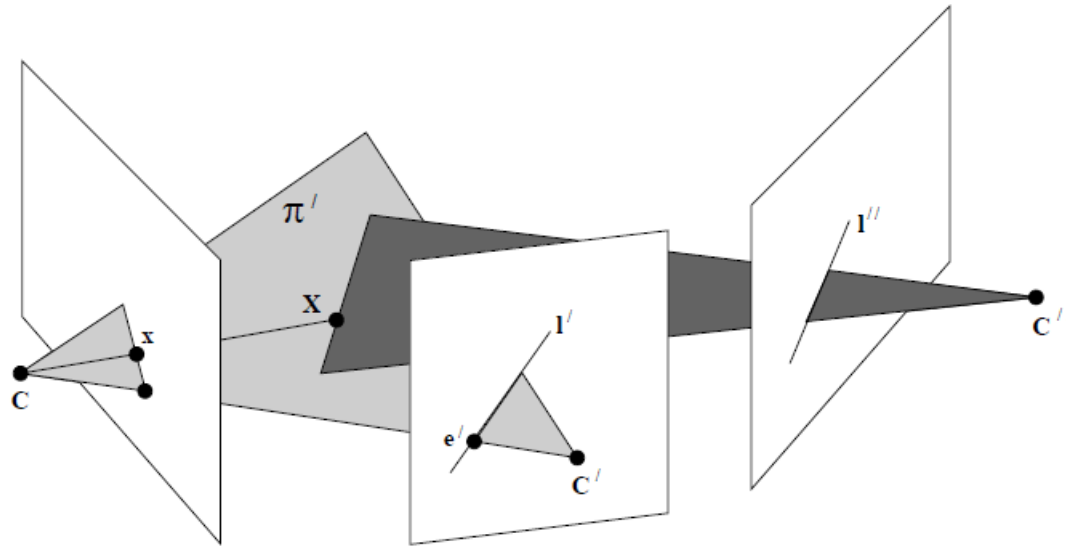
# Point Transfer using the Trifocal Tensor



- The degeneracy of epipolar transfer **is avoided** by use of the trifocal tensor.
- It is clear from the **point-line-point** incidence relation that  $x''^k = x^i l'_j \mathcal{T}_i^{jk}$  is not degenerate.

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

# Point Transfer using the Trifocal Tensor

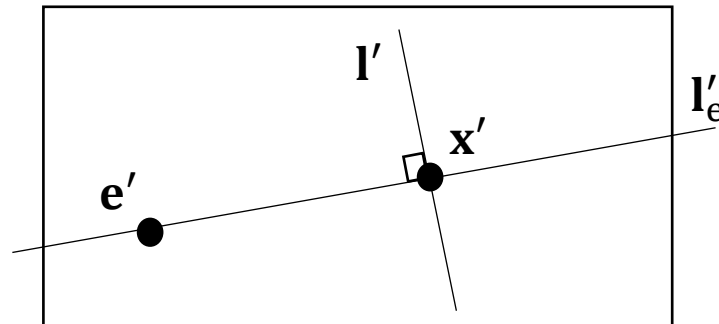


- However, if  $l'$  is the **epipolar line** corresponding to  $x$ , then  $x^i l'_j \mathcal{T}_i^{jk} = 0^k$ , so the point  $x''$  is undefined.

# Point Transfer using the Trifocal Tensor

## Solution:

- Avoid the epipolar line by computing the line  $\mathbf{l}'$  **through**  $\mathbf{x}'$  and **perpendicular** to  $\mathbf{l}'_e = \mathbf{F}_{21}\mathbf{x}$ .
- If  $\mathbf{l}'_e = (l_1, l_2, l_3)^\top$  and  $\mathbf{x}' = (x_1, x_2, 1)^\top$ , then  $\mathbf{l}' = (l_2, -l_1, -x_1l_2 + x_2l_1)^\top$ .
- The transferred point is  $x''^k = x^i l'_j \mathcal{T}_i^{jk}$ .



# Computation of the Trifocal Tensor: Linear Method

- Given several point or line correspondences between three images, the complete set of equations generated is of the form  $A\mathbf{t} = \mathbf{0}$ .
- $\mathbf{t}$  is the **27-vector** made up of the entries of the trifocal tensor.
- With more than 26 equations, a **least-squares solution** is computed by:

$$\min_{\mathbf{t}} \|\mathbf{A}\mathbf{t}\| \quad \text{s.t. } \|\mathbf{t}\| = 1.$$

- We can solve this minimization using SVD.



# Trilinear Relations between Point and Lines

- We use these equations that are **linear** in the entries of the trifocal tensor to form  $\mathbf{A}\mathbf{t} = \mathbf{0}$ .

Correspondence	Relation	Number of equations
three points	$x^i x'^j x''^k \epsilon_{jqs} \epsilon_{krt} \mathcal{T}_i^{qr} = 0_{st}$	4
two points, one line	$x^i x'^j l''_r \epsilon_{jqs} \mathcal{T}_i^{qr} = 0_s$	2
one point, two lines	$x^i l'_q l''_r \mathcal{T}_i^{qr} = 0$	1
three lines	$l_p l'_q l''_r \epsilon^{piw} \mathcal{T}_i^{qr} = 0^w$	2

**Note:**  $s, t, w \in \{1, 2\}$ .

Table Source: Page 391, R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”

# Trilinear Relations between Point and Lines

Correspondence	Relation	Number of equations
three points	$x^i x'^j x''^k \epsilon_{jq s} \epsilon_{k r t} \mathcal{T}_i^{q r} = 0_{s t}$	4

- The first line in the table corresponds to a set of **9 equations**, one for each choice of  $s, t \in \{1, 2, 3\}$ .
- However, among this set of 9 equations, only **4 are linearly independent**, hence, only  $s, t \in \{1, 2\}$  are considered.
- This is due to **rank 2 constraint** of the trifocal tensors, we will skip the complete proof.

# Normalization

- The recommended normalization is **much the same** as that given for the computation of the fundamental matrix:
  1. **A translation** is applied to each image such that the centroid of the points is at the origin.
  2. And then **a scaling** is applied so that the average (RMS) distance of the points from the origin is  $\sqrt{2}$ .
- In the case of **lines**, the transformation should be defined by considering each line's **two endpoints**.

# The Normalized Linear Algorithm for Computation of $\mathcal{T}$

## Objective

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

## Algorithm

- (i) Find transformation matrices  $H$ ,  $H'$  and  $H''$  to apply to the three images.
- (ii) Transform points according to  $x^i \mapsto \hat{x}^i = H_j^i x^j$ , and lines according to  $l_i \mapsto \hat{l}_i = (H^{-1})_i^j l_j$ . Points and lines in the second and third image transform in the same way.
- (iii) Compute the trifocal tensor  $\hat{\mathcal{T}}$  linearly in terms of the transformed points and lines using the equations in table 16.1 by solving a set of equation of the form  $A\mathbf{t} = \mathbf{0}$ , using algorithm A5.4(p593).
- (iv) Compute the trifocal tensor corresponding to the original data according to  $\mathcal{T}_i^{jk} = H_i^r (H'^{-1})_s^j (H''^{-1})_t^k \hat{\mathcal{T}}_r^{st}$ .

Table Source: Page 394, R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision"

# The Algebraic Minimization Algorithm

- The linear algorithm will give a tensor **not necessarily** corresponding to any geometric configuration.
- The next task is to **correct the tensor** to satisfy all required constraints.
- Just as with the fundamental matrix, we need to **enforce the epipole constraints**.

# Retrieving the Epipoles

- Recall that the **epipolar lines**  $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$  in the third view that correspond to 3 points  $(1, 0, 0)^T$ ,  $(0, 1, 0)^T$  and  $(0, 0, 1)^T$  in the first view is the **right null-space** of  $T_1, T_2, T_3$ :

$$T_i \mathbf{v}_i = \mathbf{0}.$$

- And the epipole is obtained at the intersection of these three lines, i.e. **the null-vectors** of the  $3 \times 3$  matrix formed by  $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ :

$$\mathbf{e}''^T [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3] = \mathbf{0}.$$

# Retrieving the Epipoles

- However, in the **presence of noise**, it is better to compute the epipoles as follows:
  1. For each  $i = 1, \dots, 3$  find the unit vector  $\mathbf{v}_i$  that **minimizes  $\|\mathbf{T}_i \mathbf{v}_i\|$** , where  $\mathbf{T}_i = \mathcal{J}_i''$ . Form the matrix  $V$ , the  $i$ -th row of which is  $\mathbf{v}_i^T$ .
  2. Compute the epipole  $\mathbf{e}''$  as the unit vector that **minimizes  $\|V \mathbf{e}''\|$** .
- The epipole  $\mathbf{e}'$  is computed similarly, using  $\mathbf{T}_i^T$  instead of  $\mathbf{T}_i$ .

# Algebraic Minimization

- Now the epipoles  $\mathbf{e}'^j = a_4^j$  and  $\mathbf{e}''^k = b_4^k$  of the **camera matrices**  $P'$  and  $P''$  are known.
- The trifocal tensor may be **written linearly** as  $\mathbf{t} = \mathbf{E}\mathbf{a}$ .
- $\mathbf{a}$  is the vector of the remaining entries  $a_i^j$  and  $b_i^k$ , and  $\mathbf{E}$  is the **linear relationship** expressed by:

$$\mathcal{T}_i^{jk} = a_i^j b_4^k - a_4^j b_i^k.$$



# Algebraic Minimization

- We wish to minimize the **algebraic error**:

$$\|A\mathbf{t}\| = \|A\mathbf{E}\mathbf{a}\|$$

over all choices of  $\mathbf{a}$  constrained such that

$$\|\mathbf{t}\| = 1, \text{ that is } \|\mathbf{E}\mathbf{a}\| = 1.$$

- The solution  $\mathbf{t} = \mathbf{E}\mathbf{a}$  represents a trifocal tensor **satisfying all constraints**, and minimizing the algebraic error, s.t. the given choice of epipoles.

# Geometric Distance

## Objective

Given  $n \geq 7$  image point correspondences  $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i\}$ , determine the Maximum Likelihood Estimate of the trifocal tensor.

The MLE involves also solving for a set of subsidiary point correspondences  $\{\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i \leftrightarrow \hat{\mathbf{x}}''_i\}$ , which exactly satisfy the trilinear relations of the estimated tensor and which minimize

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

## Algorithm

- (i) Compute an initial geometrically valid estimate of  $\mathcal{T}$  using a linear algorithm such as algorithm 16.2.
- (ii) Compute an initial estimate of the subsidiary variables  $\{\hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i, \hat{\mathbf{x}}''_i\}$  as follows:
  - (a) Retrieve the camera matrices  $P'$  and  $P''$  from  $\mathcal{T}$ .
  - (b) From the correspondence  $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i$  and  $P = [I \mid 0]$ ,  $P'$ ,  $P''$  determine an estimate of  $\hat{\mathbf{X}}_i$  using the triangulation method of chapter 12.
  - (c) The correspondence consistent with  $\mathcal{T}$  is obtained as  $\hat{\mathbf{x}}_i = P\hat{\mathbf{X}}_i$ ,  $\hat{\mathbf{x}}'_i = P'\hat{\mathbf{X}}_i$ ,  $\hat{\mathbf{x}}''_i = P''\hat{\mathbf{X}}_i$ .
- (iii) Minimize the cost

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

over  $\mathcal{T}$  and  $\hat{\mathbf{X}}_i, i = 1, \dots, n$ . The cost is minimized using the Levenberg–Marquardt algorithm over  $3n + 24$  variables:  $3n$  for the  $n$  3D points  $\hat{\mathbf{X}}_i$ , and 24 for the elements of the camera matrices  $P', P''$ .

Table Source: Page 397, R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”

# Summary

- We have looked at how to:
  1. Derive the **trifocal tensor** constraint from point and/or line image correspondences of 3 views.
  2. Describe the **homography relations** between 3 views.
  3. Extract the 3-view **epipoles** and **epipolar lines** from the trifocal tensor.
  4. Decompose the trifocal tensor into the **camera** and **fundamental matrices** of 3 views.
  5. Compute the **trifocal tensor** from point and/line image correspondences of 3 views.