National University of Singapore
School of Computing

CS4277 / CS5477
3D Computer Vision

Lecture 8:
Three-View Geometry from Points and/or Lines

Asst. Prof. Lee Gim Hee

AY 2019/20

Semester 2

Course Schedule

Week	Date	Topic	Assignments
1	15 Jan	2D and 1D projective geometry	
2	22 Jan	Circular points and 3D projective geometry	
3	29 Jan	No Lecture	
4	05 Feb	Absolute conic and robust homography estimation	**Assignment 1**: Panoramic stitching (15%)
5	12 Feb	Camera models and calibration	
6	19 Feb	Single view metrology	**Due:** Assignment 1 **Assignment 2**: Camera calibration (15%)
-	26 Feb	Semester Break	No lecture
7	04 Mar	The fundamental and essential matrices	**Due:** Assignment 2
8	11 Mar	Absolute pose estimation from points or lines	**Assignment 3**: Relative and absolute pose estimation (20%)
9	18 Mar	Three-view geometry from points and/or lines	
10	25 Mar	Structure-from-Motion (SfM) and Visual Simultaneous Localization and Mapping (vSLAM)	**Due:** Assignment 3
11	01 Apr	Two-view and multi-view stereo	**Assignment 4:** Dense 3D model from multi-view stereo (20%)
12	08 Apr	Generalized cameras	
13	15 Apr	Factorization and non-rigid structure-from-motion	**Due:** Assignment 4

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.

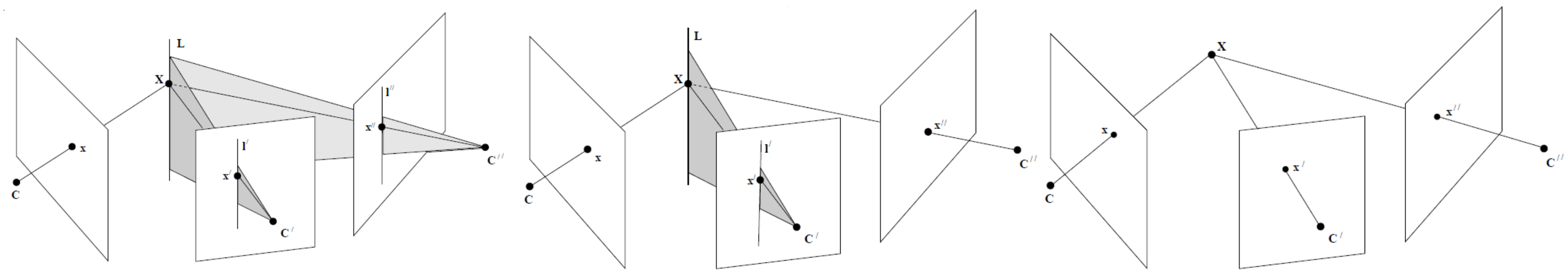
Acknowledgements

- 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.

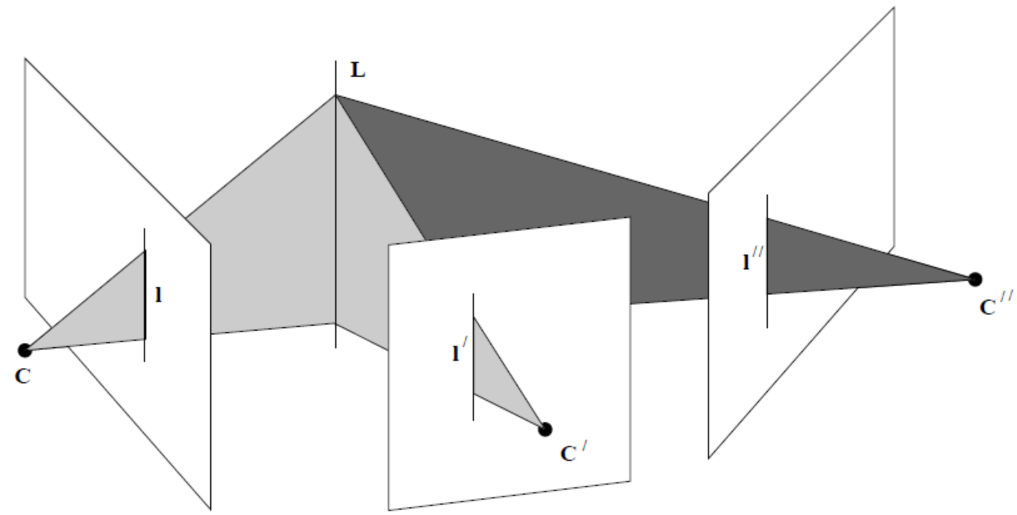
Three-View Geomtery



- 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.

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

The Geometric Basis for the Trifocal Tensor

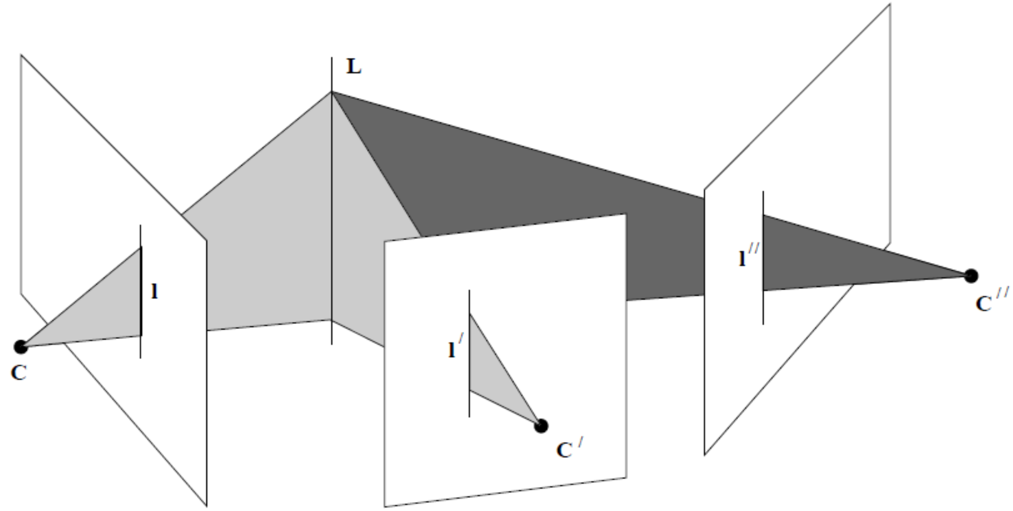


- We use the incidence relationship of three corresponding lines to derive the trifocal tensor.

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

The Geometric Basis for the Trifocal Tensor

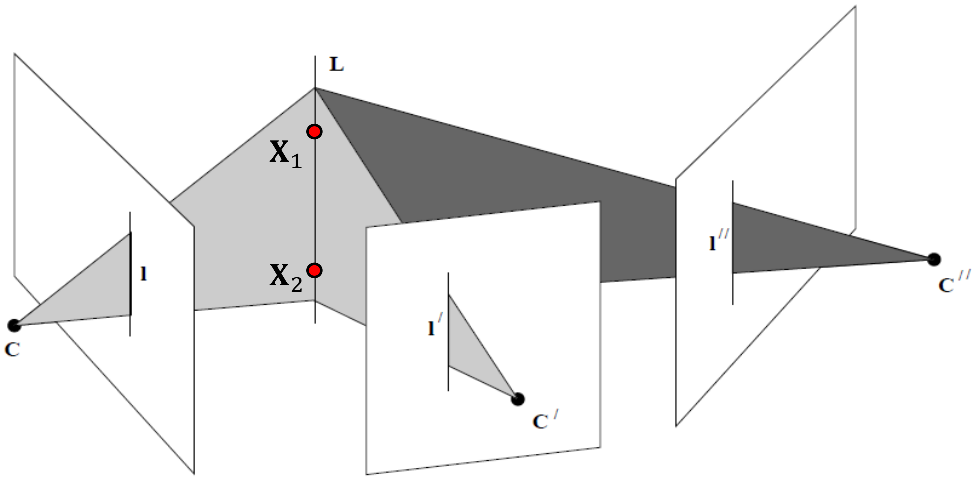
- Each image line back-projects to a plane, i.e.

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

- 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 4×3 matrix $\mathrm{M} = [\boldsymbol{\pi}, \boldsymbol{\pi}', \boldsymbol{\pi}'']$ 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:

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

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

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 $\mathtt{M}$ is 2, there is a linear dependence between its columns $\mathbf{m}_i$, i.e.

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

$$\mathtt{M} = [\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3] = \begin{bmatrix} \mathbf{l} & \mathtt{A}^\top \mathbf{l}' & \mathtt{B}^\top \mathbf{l}'' \\ 0 & \mathbf{a}_4^\top \mathbf{l}' & \mathbf{b}_4^\top \mathbf{l}'' \end{bmatrix}.$$

The Geometric Basis for the Trifocal Tensor

- Noting that the bottom left element of M is zero, it follows that $\alpha = k(\mathbf{b}_4^\mathsf{T}\mathbf{l}'')$ and $\beta = -k(\mathbf{a}_4^\mathsf{T}\mathbf{l}')$ for some scalar *k*.

- Applying this to the top 3-vectors of each column shows that (up to a homogeneous scale factor):

$$\begin{aligned}\mathbf{l} &= (\mathbf{b}_4^\mathsf{T}\mathbf{l}'')\mathtt{A}^\mathsf{T}\mathbf{l}' - (\mathbf{a}_4^\mathsf{T}\mathbf{l}')\mathtt{B}^\mathsf{T}\mathbf{l}'' \\ &= (\mathbf{l}''^\mathsf{T}\mathbf{b}_4)\mathtt{A}^\mathsf{T}\mathbf{l}' - (\mathbf{l}'^\mathsf{T}\mathbf{a}_4)\mathtt{B}^\mathsf{T}\mathbf{l}''.\end{aligned}$$

The Geometric Basis for the Trifocal Tensor

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

$$l_i = \mathbf{l}''^{\top}(\mathbf{b}_4\mathbf{a}_i^{\top})\mathbf{l}' - \mathbf{l}'^{\top}(\mathbf{a}_4\mathbf{b}_i^{\top})\mathbf{l}'' = \mathbf{l}'^{\top}(\mathbf{a}_i\mathbf{b}_4^{\top})\mathbf{l}'' - \mathbf{l}'^{\top}(\mathbf{a}_4\mathbf{b}_i^{\top})\mathbf{l}''$$

- And introducing the notation:

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

- The incidence relation can be written:

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

The Geometric Basis for the Trifocal Tensor

- The set of three matrices $\{\mathrm{T}_1, \mathrm{T}_2, \mathrm{T}_3\}$ constitute the *trifocal tensor* in matrix notation.

- We can now write the incidence relation as:

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

represent the vector

$$(\mathbf{l}'^{\mathsf{T}}\mathrm{T}_1\mathbf{l}'', \mathbf{l}'^{\mathsf{T}}\mathrm{T}_2\mathbf{l}'', \mathbf{l}'^{\mathsf{T}}\mathrm{T}_3\mathbf{l}'')$$

The Geometric Basis for the Trifocal Tensor

- Note that there exists similar relations:

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

- The three tensors $[\mathrm{T}_i]$, $[\mathrm{T}_i']$ and $[\mathrm{T}_i'']$ 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.

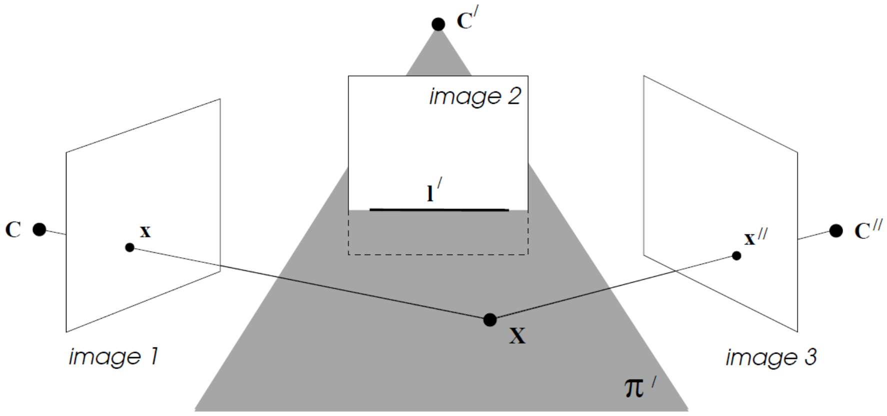
The Geometric Basis for the Trifocal Tensor

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.



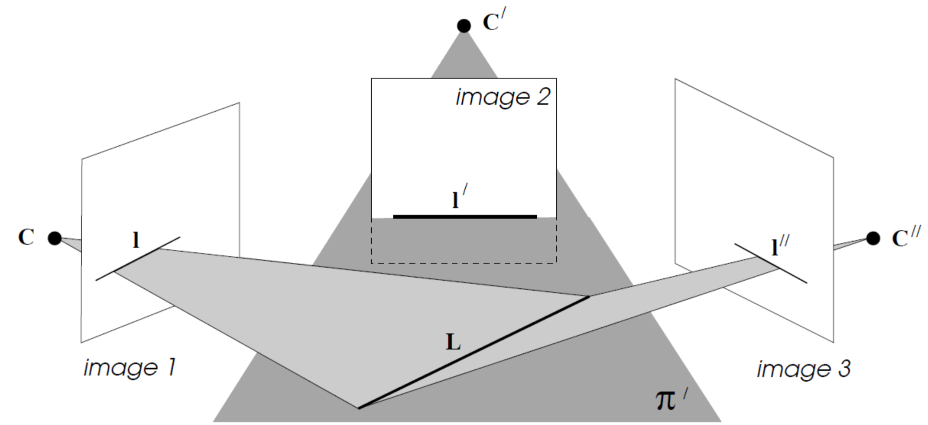
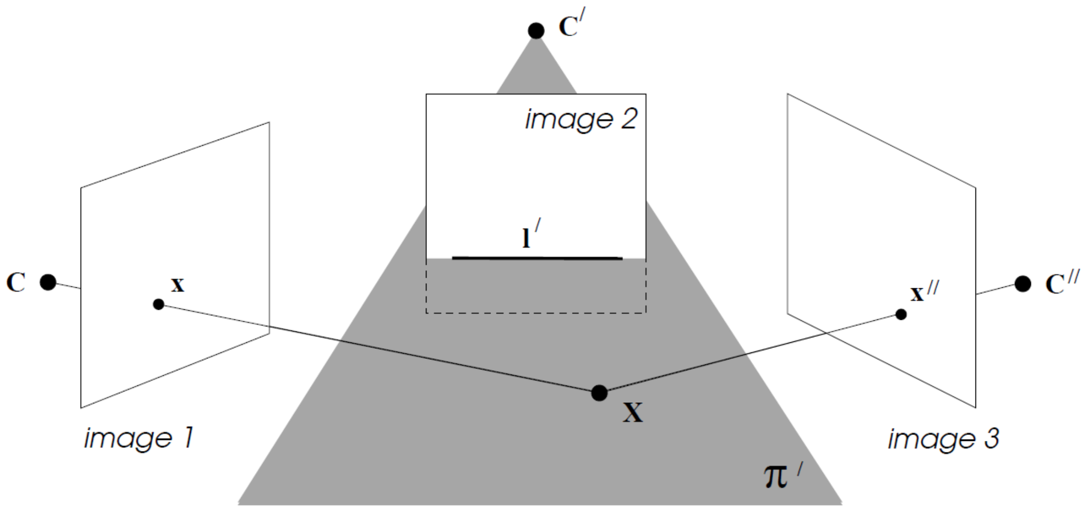


Image Source: R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision"

Homographies Induced by a Plane

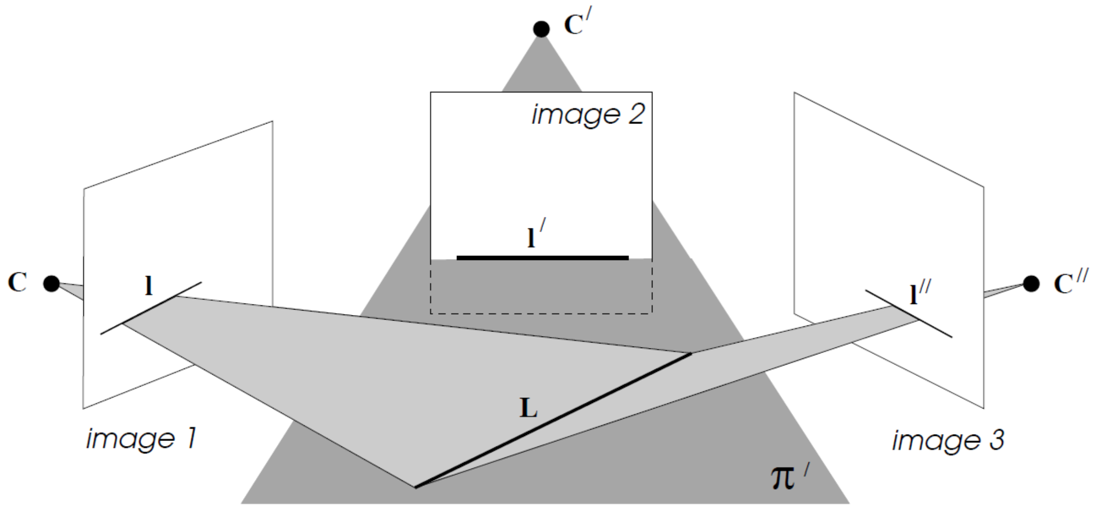


- The homography from the first to the third image induced by a line $\mathbf{l}'$ in the second image is given by $\mathbf{x}'' = \mathrm{H}_{13}(\mathbf{l}')\,\mathbf{x}$, where

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

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

Homographies Induced by a Plane

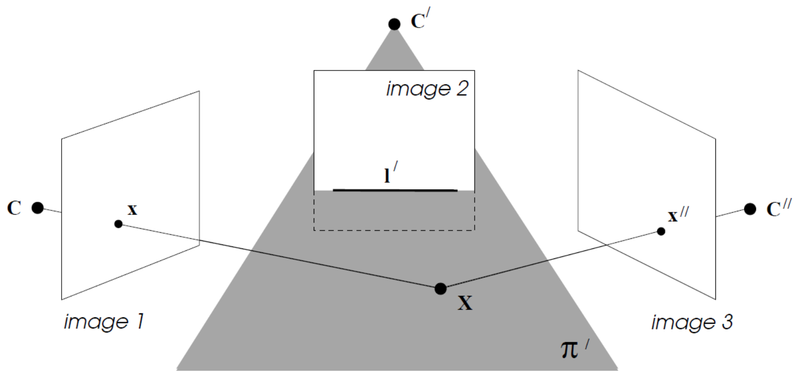


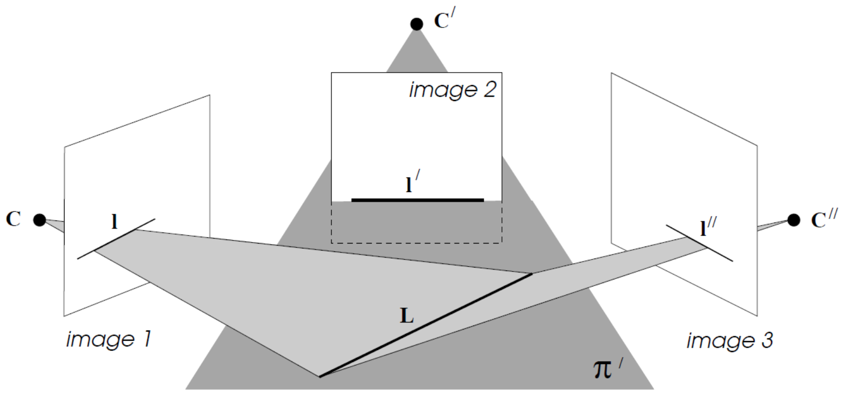
- Similarly, a line $\mathbf{l}''$ in the third image defines a homography $\mathbf{x}' = \mathrm{H}_{12}(\mathbf{l}'')\,\mathbf{x}$ from the first to the second views, given by

$$\mathrm{H}_{12}(\mathbf{l}'') = [\mathrm{T}_1, \mathrm{T}_2, \mathrm{T}_3]\mathbf{l}''.$$

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

Homographies Induced by a Plane



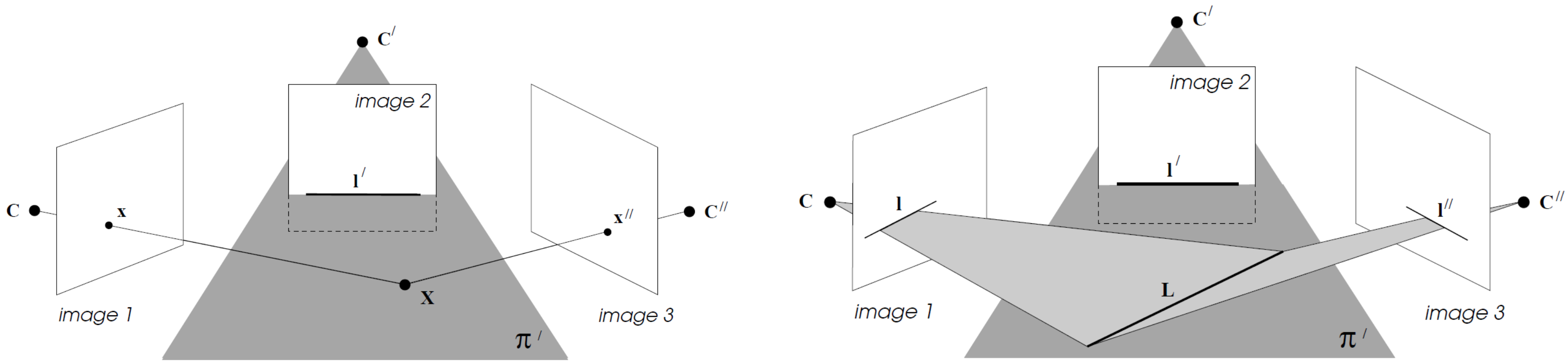


Proof:

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

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

Homographies Induced by a Plane



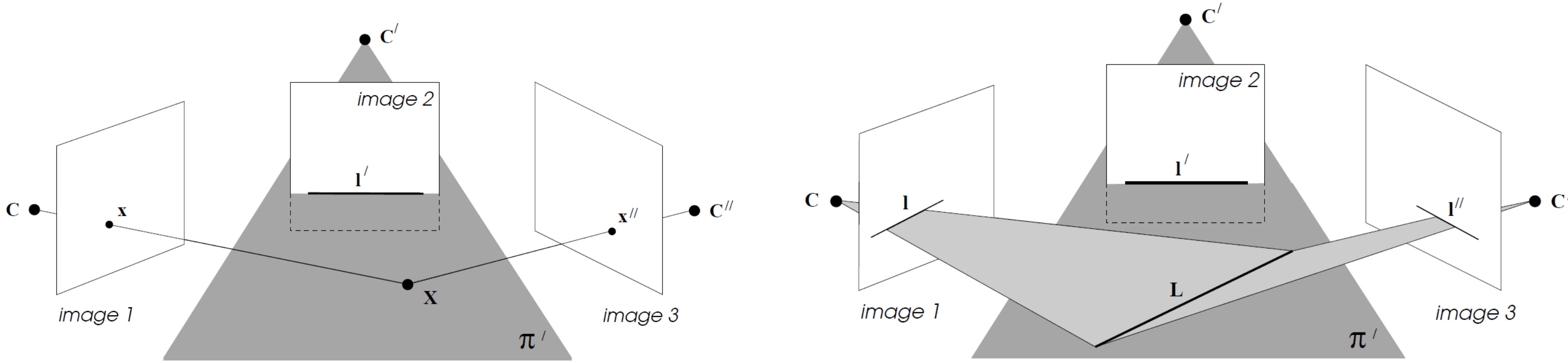
Proof (cont.):

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

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

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 $\mathbf{l}^{\mathrm{T}} = \mathbf{l}'^{\mathrm{T}}\mathrm{H}$, where

$$\mathrm{H}_{12}(\mathbf{l}'') = [\mathrm{h}_1, \mathrm{h}_2, \mathrm{h}_3] \text{ with } \mathrm{h}_i = \mathrm{T}_i \mathbf{l}''$$

□

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

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

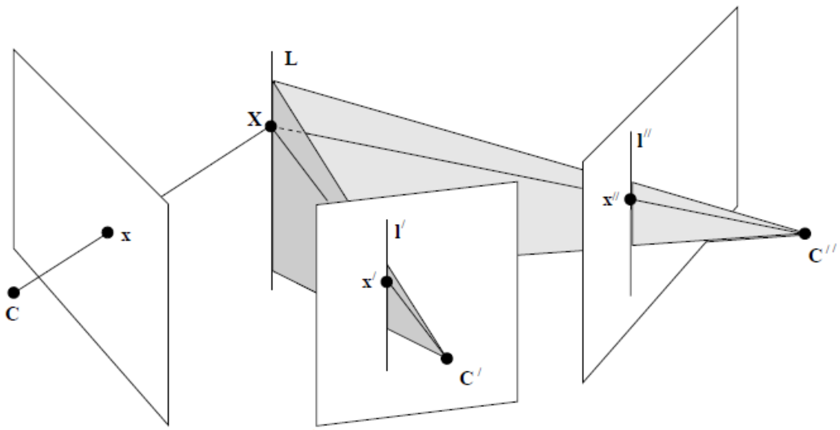
- or more briefly $(\mathbf{l}'^{T}[\mathrm{T}_i]\mathbf{l}'')[\mathbf{l}]_\times = \mathbf{0}^T$.

- The symmetry between $\mathbf{l}'$ and $\mathbf{l}''$ means the following is true too:

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

Point and Line Incidence Relations

Point-Line-Line Correspondences:



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

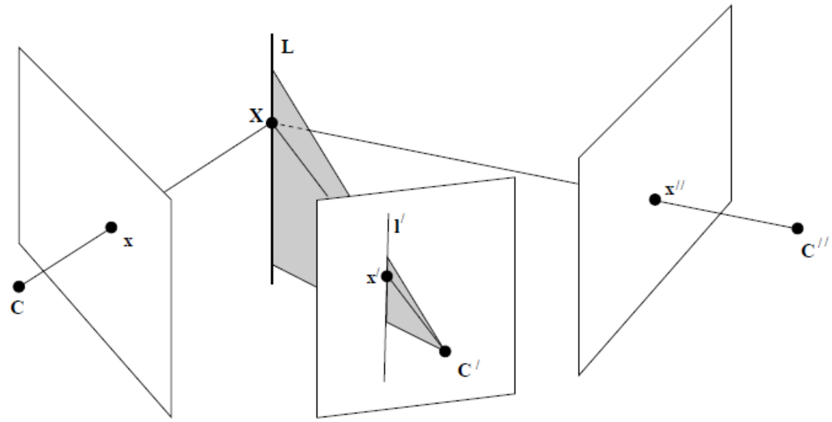
$$\mathbf{l}'^{\mathsf{T}}\left(\sum_i x^i \mathrm{T}_i\right)\mathbf{l}'' = 0$$

- Note that $\sum_i x^i \mathrm{T}_i$ is simply a 3×3 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

$$\mathrm{x}'' = \mathrm{H}_{13}(\mathbf{l}')\,\mathrm{x} = [\mathrm{T}_1^\top \mathbf{l}', \mathrm{T}_2^\top \mathbf{l}', \mathrm{T}_3^\top \mathbf{l}']\,\mathrm{x} = (\sum_i x^i \mathrm{T}_i^\top)\mathbf{l}'$$

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

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

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

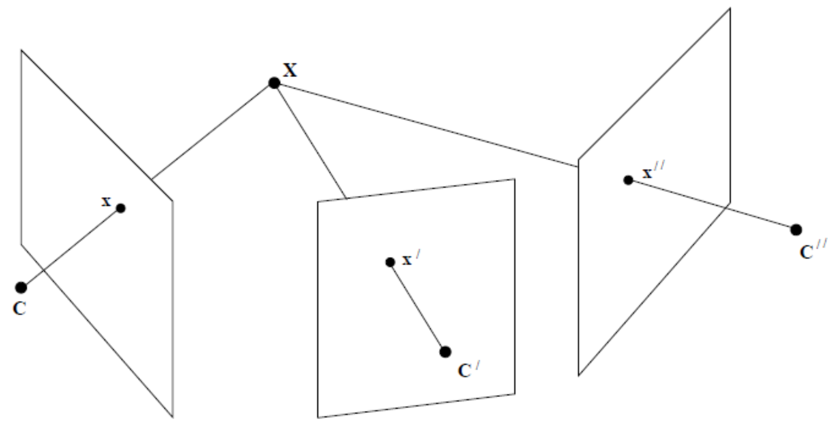
$$\mathbf{x}''^\mathsf{T}[\mathbf{x}'']_\times = \mathbf{l}'^\mathsf{T}(\sum_i x^i \mathrm{T}_i)[\mathbf{x}'']_\times = \mathbf{0}^\mathsf{T},$$

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

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

Point and Line Incidence Relations

Point-Point-Point Correspondences:



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

$$[\mathbf{x}']_\times \left(\sum_i x^i \mathrm{T}_i\right)[\mathbf{x}'']_\times = 0_{3\times3}.$$

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

Point and Line Incidence Relations

Proof:

- The line $\mathbf{l}'$ in $\mathbf{l}'^T \sum_i x^i \mathrm{T}_i [\mathbf{x}'']_\times = \mathbf{0}^T$ 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,

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

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

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

□

Point and Line Incidence Relations

(i) Line–line–line correspondence

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

(ii) Point–line–line correspondence

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

(iii) Point–line–point correspondence

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

(iv) Point–point–line correspondence

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

(v) Point–point–point correspondence

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

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

Epipolar Lines

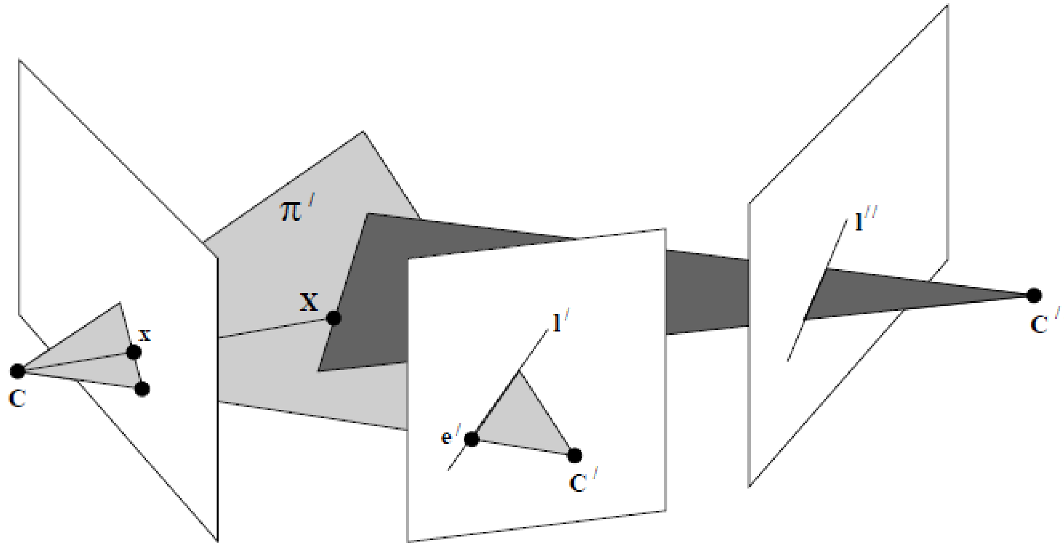
- If $\mathbf{x}$ is a point and $\mathbf{l}'$ or $\mathbf{l}''$ are the corresponding epipolar lines in the second and third images, then

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

- 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 $\sum_i x^i \mathrm{T}_i$.

Epipolar Lines

Proof:

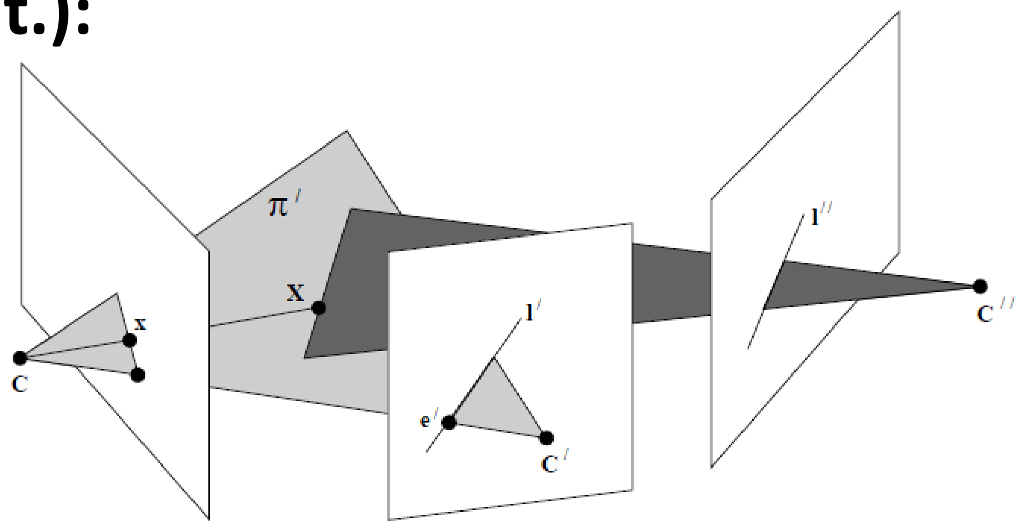


- A special case of a point–line–line correspondence occurs when the plane $\boldsymbol{\pi}'$ backprojected from $\mathbf{l}'$ is an epipolar plane with respect to the first two cameras.
- And hence passes through the camera centre $\mathbf{C}$ of the first camera.

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

Epipolar Lines

Proof (cont.):

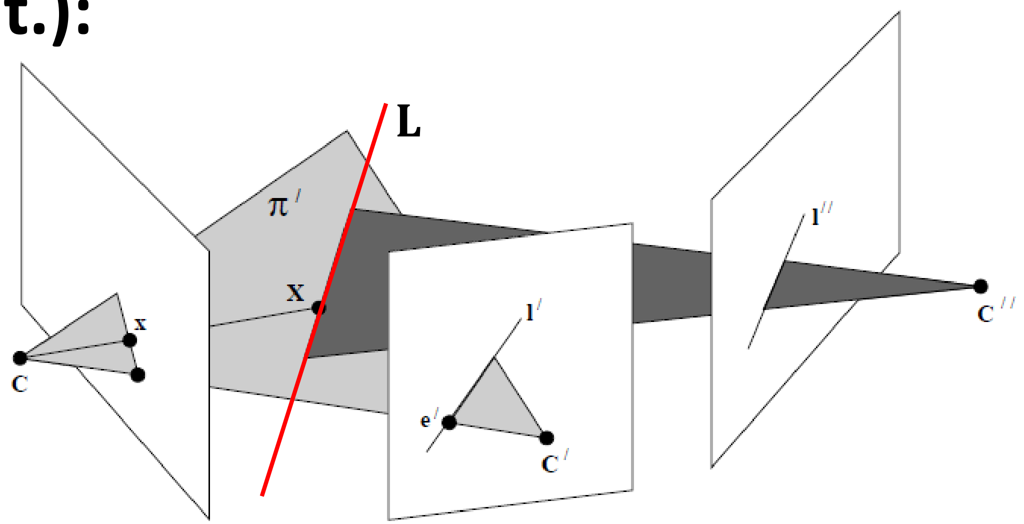


- Suppose $\mathbf{X}$ is a point on the plane $\boldsymbol{\pi}'$; then the ray defined by $\mathbf{X}$ and $\mathbf{C}$ lies in this plane.
- And $\mathbf{l}'$ is the epipolar line corresponding to the point $\mathbf{x}$, the image of $\mathbf{X}$.

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

Epipolar Lines

Proof (cont.):



- The plane $\boldsymbol{\pi}''$ back-projected from a line $\mathbf{l}''$ in the third image will intersect the plane $\boldsymbol{\pi}'$ in a line $\mathbf{L}$.

- Further, since the ray corresponding to $\mathbf{x}$ lies entirely in the plane $\boldsymbol{\pi}'$ it must intersect the line $\mathbf{L}$.

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

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 $\mathbf{l}'^T \sum_i x^i \mathrm{T}_i \, \mathbf{l}'' = 0$.
- The important point now is that this is true for *any* line $\mathbf{l}''$, and it follows that $\mathbf{l}'^T \sum_i x^i \mathrm{T}_i = 0$.
- The same argument holds with the roles of $\mathbf{l}'$ and $\mathbf{l}''$ reversed $\Rightarrow \sum_i x^i \mathrm{T}_i \, \mathbf{l}'' = 0$, where $\mathbf{l}''$ is the epipolar line and $\mathbf{l}'$ is any line in the second view.

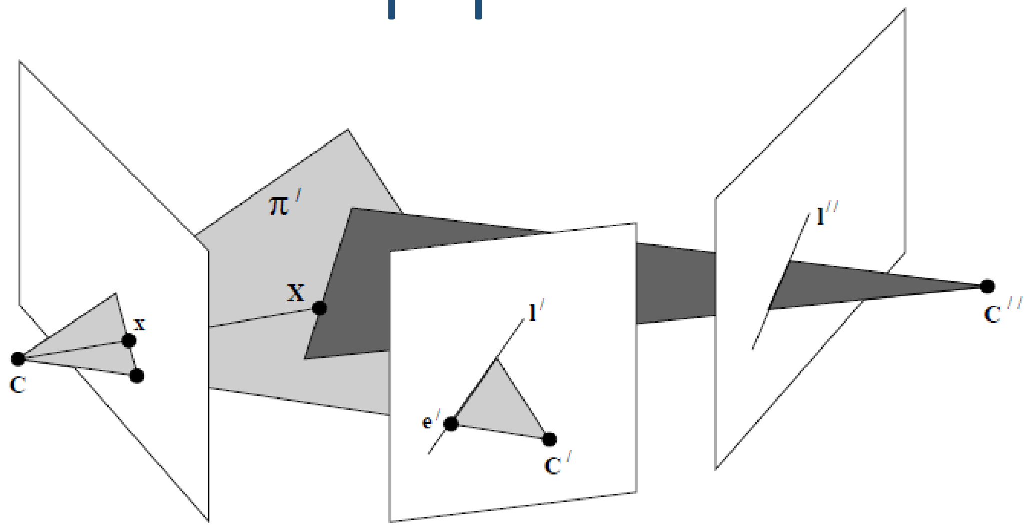
$\square$

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, $\sum_i x^i \mathrm{T}_i$ equal to T_1, T_2 and T_3 respectively for these three choices of $\mathbf{x}$.

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

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 $\mathrm{T}_i, i = 1, \ldots, 3$.
- 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 $\mathrm{T}_i = \mathbf{a}_i \boldsymbol{e}''^T - \boldsymbol{e}'\mathbf{b}_i^T$ is the sum of two outer products.

- The right null-vector of T_i is $\mathbf{l}_i'' = \boldsymbol{e}'' \times \mathbf{b}_i$, 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 $\mathbf{l}'' = (\mathrm{P}''\mathrm{C}) \times (\mathrm{P}''\mathrm{P}^+\mathbf{x})$

$$= \mathbf{e}'' \times [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4] \begin{bmatrix} \mathbf{I}_{3\times3} \\ \mathbf{0}_{1\times3} \end{bmatrix} \mathbf{x} = \mathbf{e}'' \times [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3]\mathbf{x}$$

$$\Longrightarrow \mathbf{l}_i'' = \mathbf{e}'' \times \mathbf{b}_i. \qquad \square$$

Algebraic Properties of the T_i Matrices

- The epipole $\boldsymbol{e}''$ is the common intersection of the epipolar lines $\mathbf{l}_i''$ for $i = 1, 2, 3$.

- The left null-vector of T_i is $\mathbf{l}_i' = \boldsymbol{e}' \times \mathbf{a}_i$, and is the epipolar line in the second 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.

- The epipole $\boldsymbol{e}'$ is the common intersection of the epipolar lines $\mathbf{l}_i'$ for $i = 1, 2, 3$, i.e. the null-vectors of:

$$\boldsymbol{e}'^T [\mathbf{l}_1', \mathbf{l}_2', \mathbf{l}_3'] = \mathbf{0} \text{ and } \boldsymbol{e}''^T [\mathbf{l}_1'', \mathbf{l}_2'', \mathbf{l}_3''] = \mathbf{0}.$$

Algebraic Properties of the T_i Matrices

- The *sum* of the matrices $\mathrm{M}(\mathbf{x}) = \sum_i x^i \mathrm{T}_i$ also has rank 2.
- The right null-vector of $\mathrm{M}(\mathbf{x})$ 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 $\mathbf{x}' = ([\mathrm{T}_1, \mathrm{T}_2, \mathrm{T}_3]\mathbf{l}'')\,\mathbf{x}$.

- The epipolar line corresponding to $\mathbf{x}$ is then found by joining $\mathbf{x}'$ to the epipole $\boldsymbol{e}'$.

- This gives $\mathrm{l}' = [\mathrm{e}']_\times ([\mathrm{T}_1, \mathrm{T}_2, \mathrm{T}_3]\mathrm{l}'')\,\mathbf{x}$, which can be written as $\mathbf{l}' = \mathrm{F}_{21}\mathbf{x}$, i.e. the fundamental matrix is:

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

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 $\boldsymbol{e}''$ 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 $\sum_i x^i \mathrm{T}_i \, \mathbf{l}'' = 0$.

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

Extracting the Fundamental Matrices

- This gives the formula

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

- A similar formula holds for

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

Retrieving the Camera Matrices

- Since the trifocal tensor expresses a relationship between image entities only, 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 $\mathrm{P} = [\mathrm{I} \mid \mathbf{0}]$.

- Since F_{21} is known, we can make use of:

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

to derive the form of the second camera as

$$\mathrm{P}' = [[\mathrm{T}_1, \mathrm{T}_2, \mathrm{T}_3]\mathbf{e}'' \mid \mathbf{e}']$$

and the camera pair $\{\mathrm{P}, \mathrm{P}'\}$ 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 $\mathrm{P}'' = [[\mathrm{T}_1^{\mathsf{T}}, \mathrm{T}_2^{\mathsf{T}}, \mathrm{T}_3^{\mathsf{T}}]\mathbf{e}' \mid \mathbf{e}'']$, but this is incorrect.
- To see this, suppose the camera pair $\{\mathrm{P}, \mathrm{P}'\}$ 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 by $\{\mathrm{P}, \mathrm{P}'\}$, and a consistent camera P'' may be computed from $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$.
- Clearly, P'' depends on the frame defined by $\{\mathrm{P}, \mathrm{P}'\}$.
- 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 $(\mathrm{P}, \mathrm{P}')$ and $(\tilde{\mathrm{P}}, \tilde{\mathrm{P}}')$, where

$$\mathrm{P} = [\mathrm{I} \mid \mathbf{0}] \text{ and } \mathrm{P}' = [\mathrm{A} \mid \mathbf{a}], \text{ and}$$

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

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

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

$$\mathrm{P}' = [[\mathrm{T}_1, \mathrm{T}_2, \mathrm{T}_3]\mathbf{e}'' + \mathbf{e}'\mathbf{v}^{\top}|\lambda\mathbf{e}']$$

Retrieving the Camera Matrices

- Because of the projective ambiguity, we are free to choose $\mathrm{P}' = [\,[\mathrm{T}_1, \mathrm{T}_2, \mathrm{T}_3]\boldsymbol{e}'' \mid \boldsymbol{e}'\,]$, thus $\mathbf{a}_i = \mathrm{T}_i \boldsymbol{e}''$.

- This choice fixes the projective world frame so that P'' is now defined uniquely (up to scale).

- Substituting into $\mathrm{T}_i = \mathbf{a}_i \boldsymbol{e}''^T - \boldsymbol{e}' \mathbf{b}_i^T$, we get

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

from which it follows that

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

Retrieving the Camera Matrices

- Since the scale may be chosen such that

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

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

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

so

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

Summary of F and P Retrieval from the Trifocal Tensor

Given the trifocal tensor written in matrix notation as $[\mathrm{T}_1, \mathrm{T}_2, \mathrm{T}_3]$.

(i) **Retrieve the epipoles** $\mathbf{e}', \mathbf{e}''$
Let $\mathbf{u}_i$ and $\mathbf{v}_i$ be the left and right null-vectors respectively of T_i, i.e. $\mathbf{u}_i^\mathsf{T}\mathrm{T}_i = \mathbf{0}^\mathsf{T}$, $\mathrm{T}_i\mathbf{v}_i = \mathbf{0}$. Then the epipoles are obtained as the null-vectors to the following 3×3 matrices:

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

(ii) **Retrieve the fundamental matrices** $\mathrm{F}_{21}, \mathrm{F}_{31}$

$$\mathrm{F}_{21} = [\mathbf{e}']_\times[\mathrm{T}_1, \mathrm{T}_2, \mathrm{T}_3]\mathbf{e}'' \text{ and } \mathrm{F}_{31} = [\mathbf{e}'']_\times[\mathrm{T}_1^\mathsf{T}, \mathrm{T}_2^\mathsf{T}, \mathrm{T}_3^\mathsf{T}]\mathbf{e}'.$$

(iii) **Retrieve the camera matrices** $\mathrm{P}', \mathrm{P}''$ **(with** $\mathrm{P} = [\mathrm{I} \mid \mathbf{0}]$**)**
Normalize the epipoles to unit norm. Then

$$\mathrm{P}' = [[\mathrm{T}_1, \mathrm{T}_2, \mathrm{T}_3]\mathbf{e}'' \mid \mathbf{e}'] \text{ and } \mathrm{P}'' = [(\mathbf{e}''\mathbf{e}''^\mathsf{T} - \mathrm{I})[\mathrm{T}_1^\mathsf{T}, \mathrm{T}_2^\mathsf{T}, \mathrm{T}_3^\mathsf{T}]\mathbf{e}' \mid \mathbf{e}''].$$

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 column $\mathbf{x} = (x^1, x^2, x^3)^\mathsf{T}$.
- **Image line:** homogeneous row $\mathbf{l} = (l_1, l_2, l_3)$.
- The ij-th entry of a matrix A is denoted by a^i_j, 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 $(1, \ldots, 3)$ of the index.

Example:

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

$$x'^i = \sum_j a^i_j x^j,$$

which may be written as

$$x'^i = a^i_j x^j.$$

The Trifocal Tensor and Tensor Notation

- The trifocal tensor can now be written as:

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

- In tensor notation, the basic incidence relation becomes:

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

$$\Updownarrow$$

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

The Trifocal Tensor and Tensor Notation

- The homography maps becomes:

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

$$\Updownarrow$$

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

h_i^k are the elements of the homography matrix 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 3×4 camera matrices are

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

then

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

See (17.12–*p*415) 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 $\mathbf{l}'$ in the second view.

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

(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 $\mathbf{l}''$ in the third view.

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

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

The Tensor $\boldsymbol{\epsilon}_{rst}$ and $\boldsymbol{\epsilon}^{rst}$

- The tensor $\boldsymbol{\epsilon}_{rst}$ is defined for $r, s, t = 1, \ldots, 3$ 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 $\boldsymbol{\epsilon}_{rst}$ and $\boldsymbol{\epsilon}^{rst}$ are connected with the cross product of two vectors:

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

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

The Tensor $\boldsymbol{\epsilon}_{rst}$ and $\boldsymbol{\epsilon}^{rst}$

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

(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"

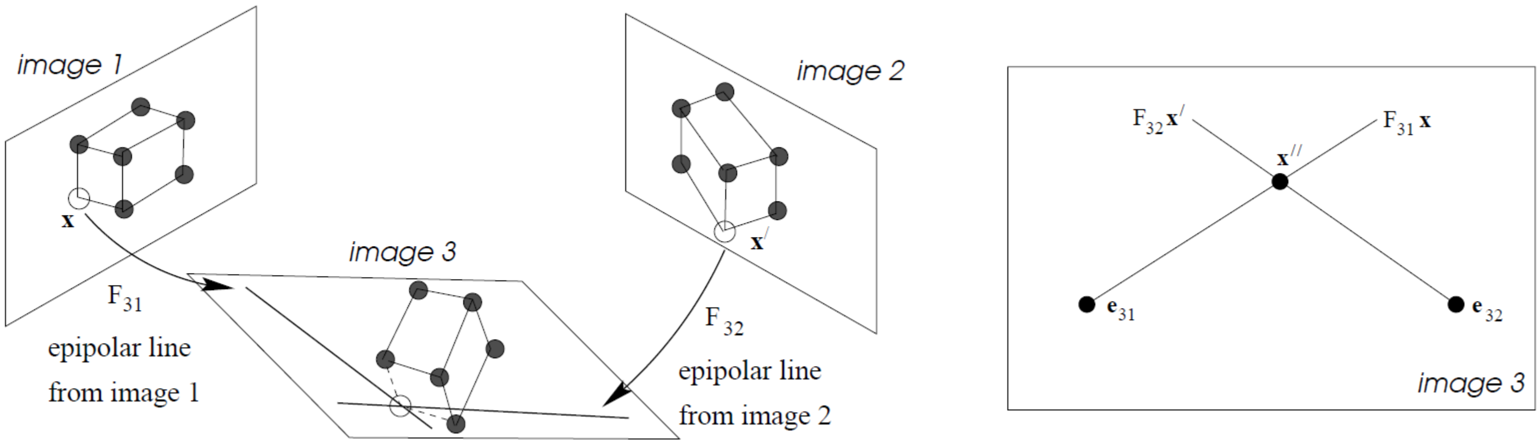
Point Transfer using Fundamental Matrices

- **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 $\mathrm{F}_{31}\mathbf{x}$ and $\mathrm{F}_{32}\mathbf{x}$.

Point Transfer using Fundamental Matrices



- The intersection of the epipolar lines gives

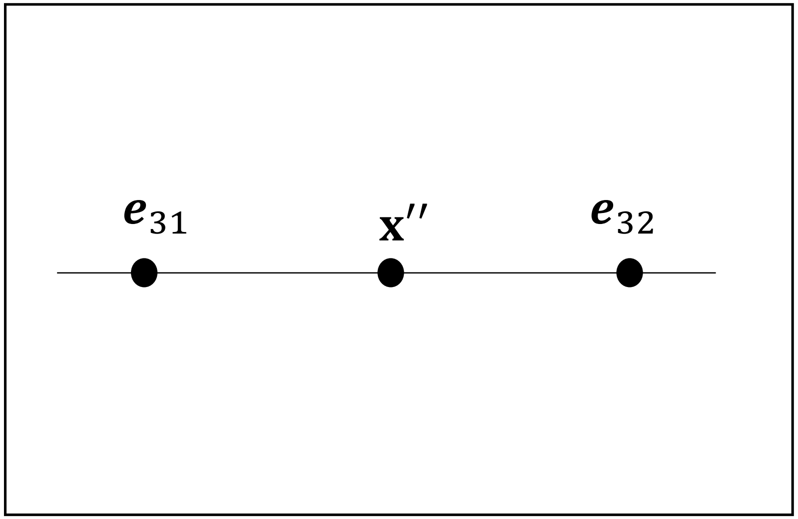
$$\mathbf{x}'' = (\mathrm{F}_{31}\mathbf{x}) \times (\mathrm{F}_{32}\mathbf{x}') \quad .$$

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

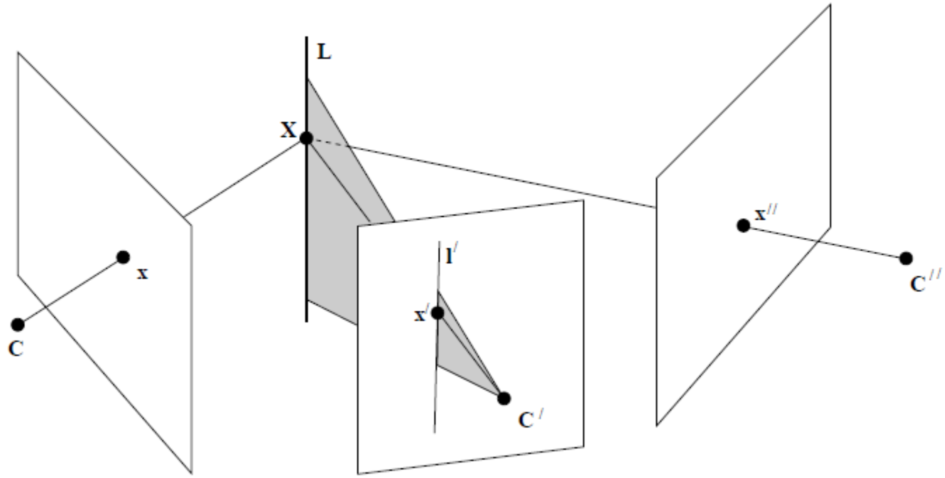
Image Source: R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”

Degeneracy of Epipolar Transfer

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



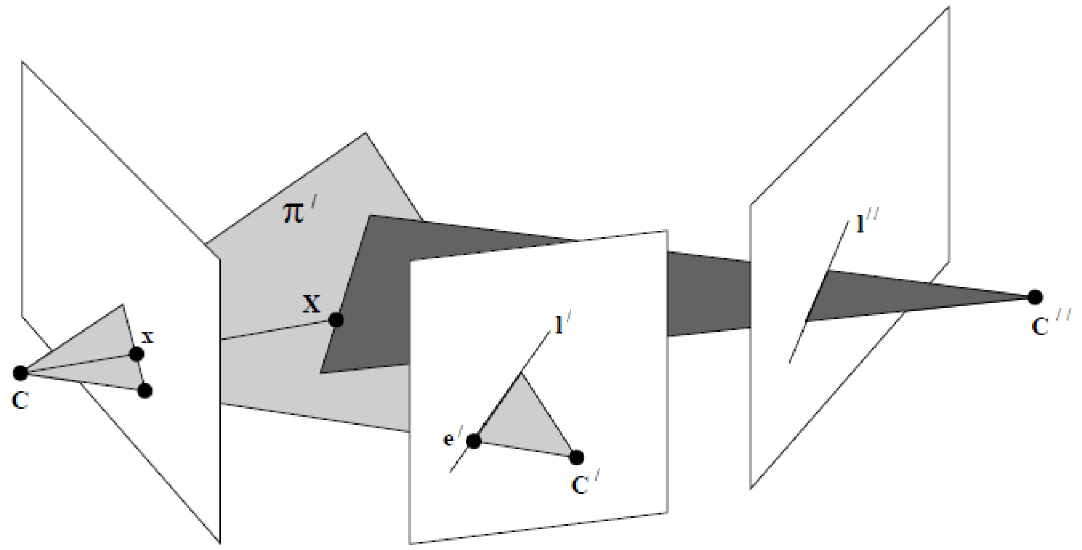
Point Transfer using the Trifocal Tensor



- 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



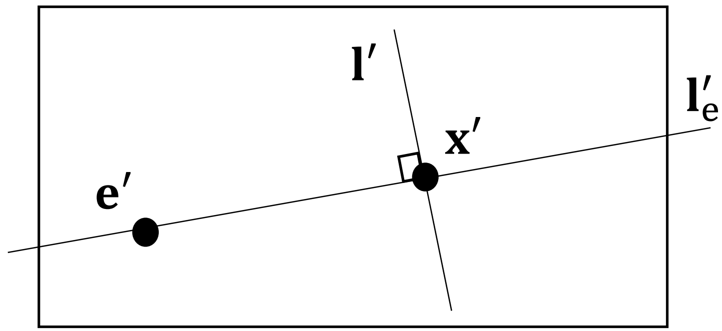
- However, if $\mathbf{l}'$ is the epipolar line corresponding to $\mathbf{x}$, then $x^i l'_j \mathcal{T}_i^{jk} = 0^k$, so the point $\mathbf{x}$ is undefined.

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

Point Transfer using the Trifocal Tensor

Solution:

- Avoid the epipolar line by computing the line $\mathbf{l}'$ through $\mathbf{x}'$ and perpendicular to $\mathbf{l}'_\mathrm{e} = \mathrm{F}_{21}\mathbf{x}$.
- If $\mathbf{l}'_\mathrm{e} = (l_1, l_2, l_3)^\mathrm{T}$ and $\mathbf{x}' = (x_1, x_2, 1)^\mathrm{T}$, then $\mathbf{l}' = (l_2, -l_1, -x_1 l_2 + x_2 l_1)^T$.
- 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 $\mathrm{A}\mathbf{t} = \mathbf{0}$.
- $\mathbf{t}$ 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:

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

- 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 $\mathbf{At} = \mathbf{0}$.

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: $s, t, w = 1, 2.$

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	$x^i x'^j x''^k \epsilon_{jqs} \epsilon_{krt} \mathcal{T}_i^{qr} = 0_{st}$	4

- The first line in the table corresponds to a set of 9 equations, one for each choice of $s, t = 1, 2, 3$.
- However, among this set of 9 equations, only 4 are linearly independent, hence, only $s, t = 1, 2$ 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 $\mathtt{H}$, $\mathtt{H}'$ and $\mathtt{H}''$ to apply to the three images.

(ii) Transform points according to $x^i \mapsto \hat{x}^i = \mathtt{H}^i_j x^j$, and lines according to $l_i \mapsto \hat{l}_i = (\mathtt{H}^{-1})^j_i l_j$. 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 $\mathtt{A}\mathbf{t} = \mathbf{0}$, using algorithm A5.4($p$593).

(iv) Compute the trifocal tensor corresponding to the original data according to $\mathcal{T}_i^{jk} = \mathtt{H}^r_i (\mathtt{H}'^{-1})^j_s (\mathtt{H}''^{-1})^k_t \hat{\mathcal{T}}_r^{st}$.

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 $\mathrm{T}_1, \mathrm{T}_2, \mathrm{T}_3$:

$$\mathrm{T}_i \mathbf{v}_i = \mathbf{0}.$$

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

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

Retrieving the Epipoles

- However, in the presence of noise, it is better to compute the epipoles as follows:

1. For each $i = 1, \ldots, 3$ find the unit vector $\mathbf{v}_i$ that minimizes $\|\mathrm{T}_i \mathbf{v}_i\|$, where $\mathrm{T}_i = \mathcal{T}_i^{\cdot\cdot}$. Form the matrix V, the i-th row of which is $\mathbf{v}_i^T$.

2. Compute the epipole $\boldsymbol{e}''$ as the unit vector that minimizes $\|\mathrm{V}\boldsymbol{e}''\|$.

- The epipole $\boldsymbol{e}'$ is computed similarly, using T_i^T instead of T_i.

Algebraic Minimization

- Now the epipoles $\boldsymbol{e}'^{j} = a_4^j$ and $\boldsymbol{e}''^{k} = b_4^k$ of the camera matrices P' and P'' are known.

- The trifocal tensor may be written linearly as $\mathbf{t} = \mathrm{E}\mathbf{a}$.

- **a** is the vector of the remaining entries a_i^j and b_i^k, and 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:

$$\|\mathtt{At}\| = \|\mathtt{AEa}\|$$

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

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

- The solution $\mathbf{t} = \mathrm{E}\mathbf{a}$ 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 $\mathtt{P}'$ and $\mathtt{P}''$ from $\mathcal{T}$.
- (b) From the correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i$ and $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}], \mathtt{P}', \mathtt{P}''$ determine an estimate of $\widehat{\mathbf{X}}_i$ using the triangulation method of chapter 12.
- (c) The correspondence consistent with $\mathcal{T}$ is obtained as $\hat{\mathbf{x}}_i = \mathtt{P}\widehat{\mathbf{X}}_i,\ \hat{\mathbf{x}}'_i = \mathtt{P}'\widehat{\mathbf{X}}_i,\ \hat{\mathbf{x}}''_i = \mathtt{P}''\widehat{\mathbf{X}}_i$.

(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 $\widehat{\mathbf{X}}_i, i = 1, \ldots, n$. The cost is minimized using the Levenberg–Marquardt algorithm over $3n + 24$ variables: $3n$ for the n 3D points $\widehat{\mathbf{X}}_i$, and 24 for the elements of the camera matrices $\mathtt{P}', \mathtt{P}''$.

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.