Fundamental matrix & Trifocal tensor

- Computation of the Fundamental Matrix F
- Introduction into the Trifocal tensor

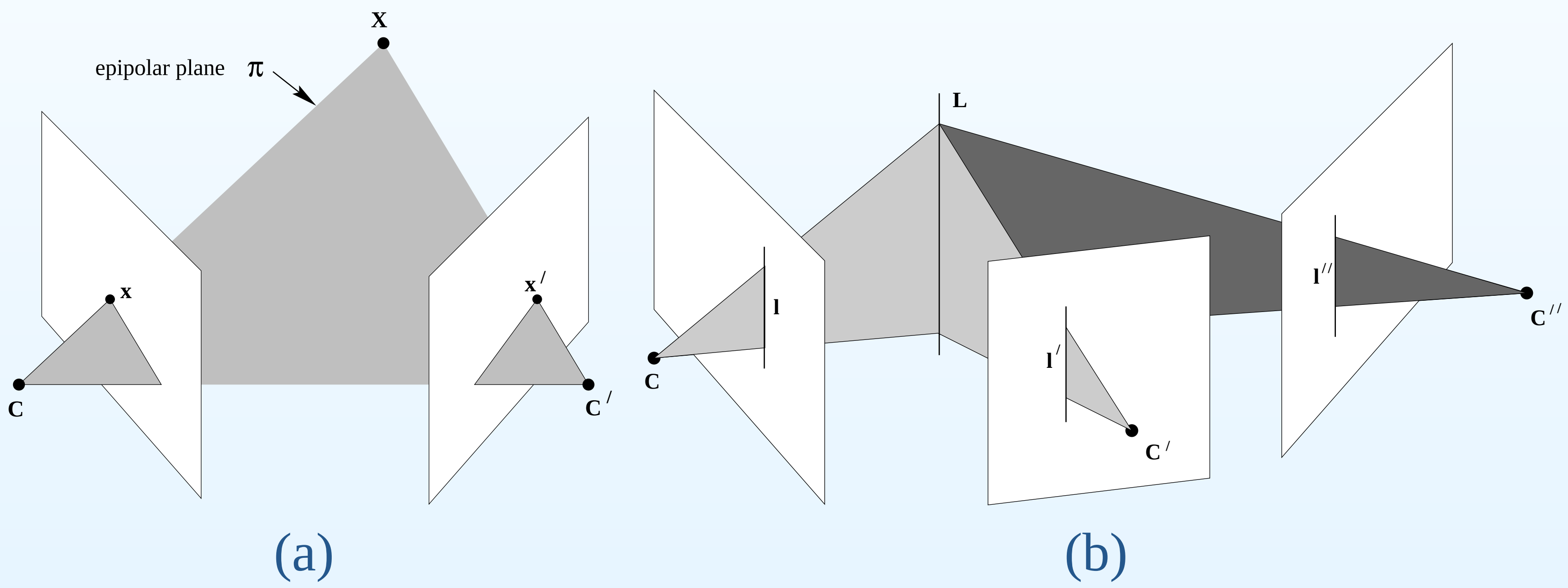


Figure 1: Two-view geometry(a), Tri-view geometry(b).

Computation of the Fundamental Matrix F

- The Fundamental Matrix F

Computation of the Fundamental Matrix F

- The Fundamental Matrix F
- Normalized 8-point algorithm

Computation of the Fundamental Matrix F

- The Fundamental Matrix F
- Normalized 8-point algorithm
- Normalized 7-point algorithm

Computation of the Fundamental Matrix F

- The Fundamental Matrix F
- Normalized 8-point algorithm
- Normalized 7-point algorithm
- The Gold standard method

Computation of the Fundamental Matrix F

- The Fundamental Matrix F
- Normalized 8-point algorithm
- Normalized 7-point algorithm
- The Gold standard method
- Automatic computation of F

Computation of the Fundamental Matrix F

- The Fundamental Matrix F
- Normalized 8-point algorithm
- Normalized 7-point algorithm
- The Gold standard method
- Automatic computation of F
- Special cases of F-computation

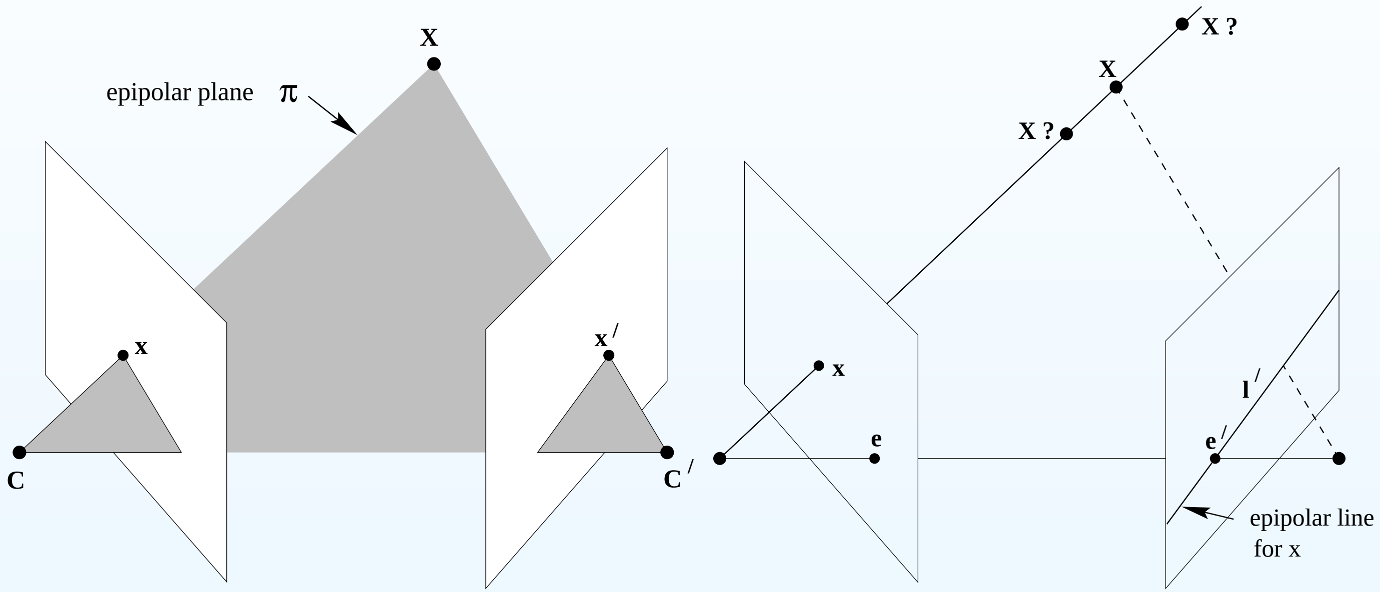
Computation of the Fundamental Matrix F

- The Fundamental Matrix F
- Normalized 8-point algorithm
- Normalized 7-point algorithm
- The Gold standard method
- Automatic computation of F
- Special cases of F-computation
- Using F for image rectification

Computation of the Fundamental Matrix F

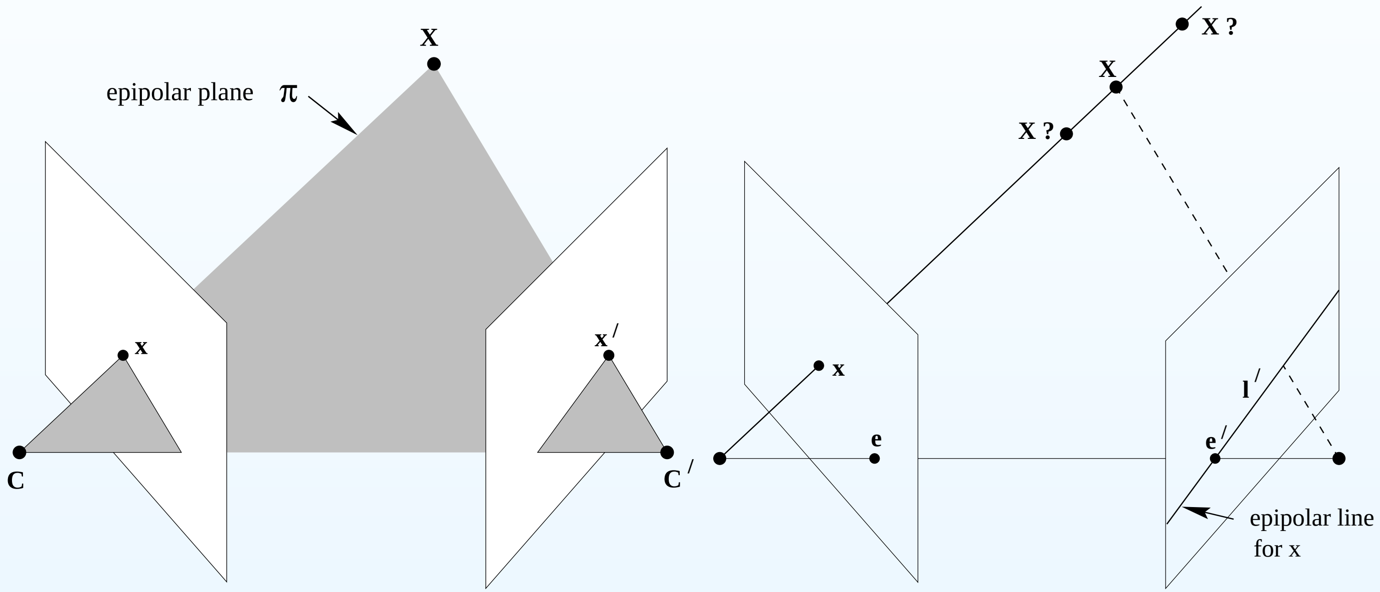
- The Fundamental Matrix F
- Normalized 8-point algorithm
- Normalized 7-point algorithm
- The Gold standard method
- Automatic computation of F
- Special cases of F-computation
- Using F for image rectification

The fundamental matrix



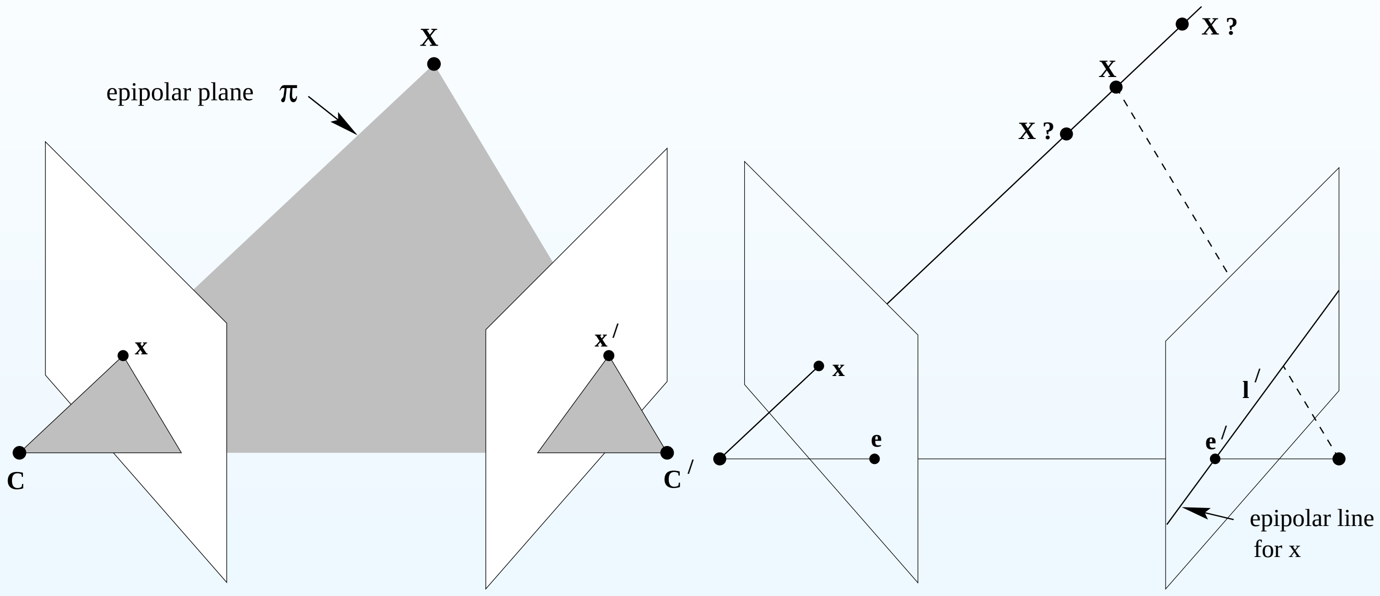
- F relates x to its epipolar line $\mathrm{l}' = \mathrm{F}\mathrm{x}$

The fundamental matrix



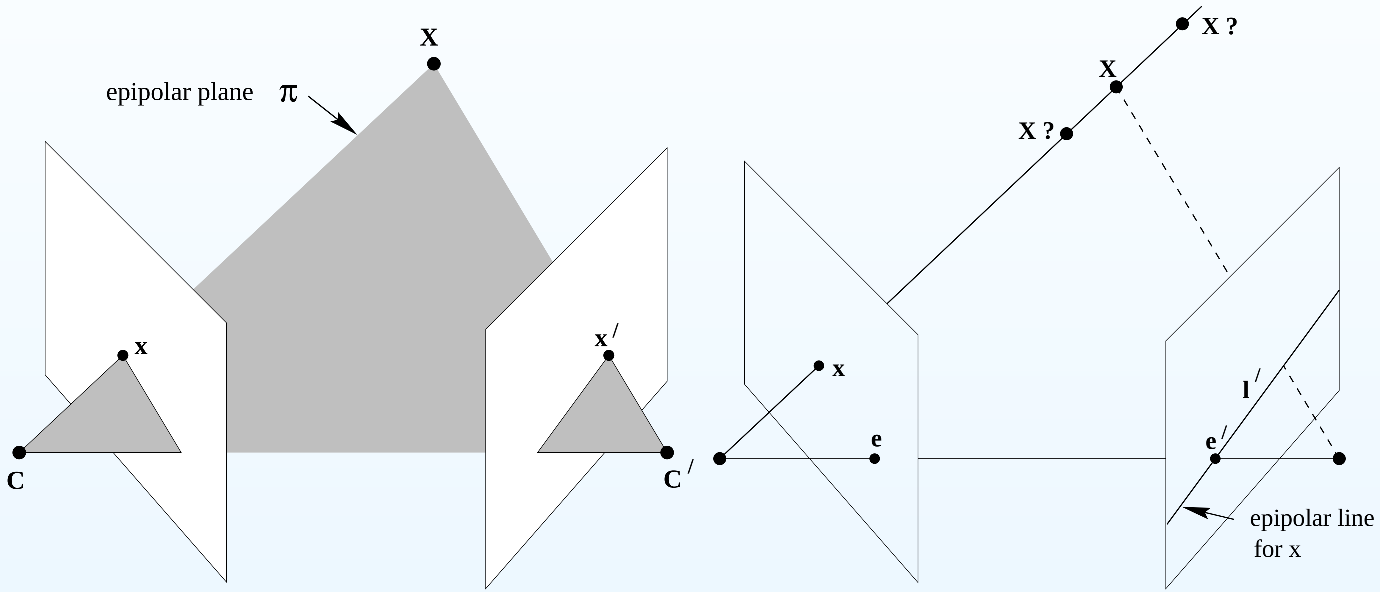
- F relates x to its epipolar line $\mathrm{l}' = \mathrm{F}\mathrm{x}$
- Since x' must be on l' we have $\mathrm{x}'^{\top}\mathrm{l}' = 0$

The fundamental matrix



- F relates x to its epipolar line $\mathrm{l}' = \mathrm{F}\mathrm{x}$
- Since x' must be on l' we have $\mathrm{x}'^{\top}\mathrm{l}' = 0$
- Thus $\mathrm{x}'^{\top}\mathrm{F}\mathrm{x} = 0$

The fundamental matrix



- F relates x to its epipolar line $\mathrm{l}' = \mathrm{F}\mathrm{x}$
- Since x' must be on l' we have $\mathrm{x}'^{\top}\mathrm{l}' = 0$
- Thus $\mathrm{x}'^{\top}\mathrm{F}\mathrm{x} = 0$
- F is singular, of rank 2, det F = 0 and F has seven degrees of freedom.

8-point algorithm

$$x = (x, y, 1) \quad x' = (x', y', 1)$$

Let f be the vector representation (row-major) of F then

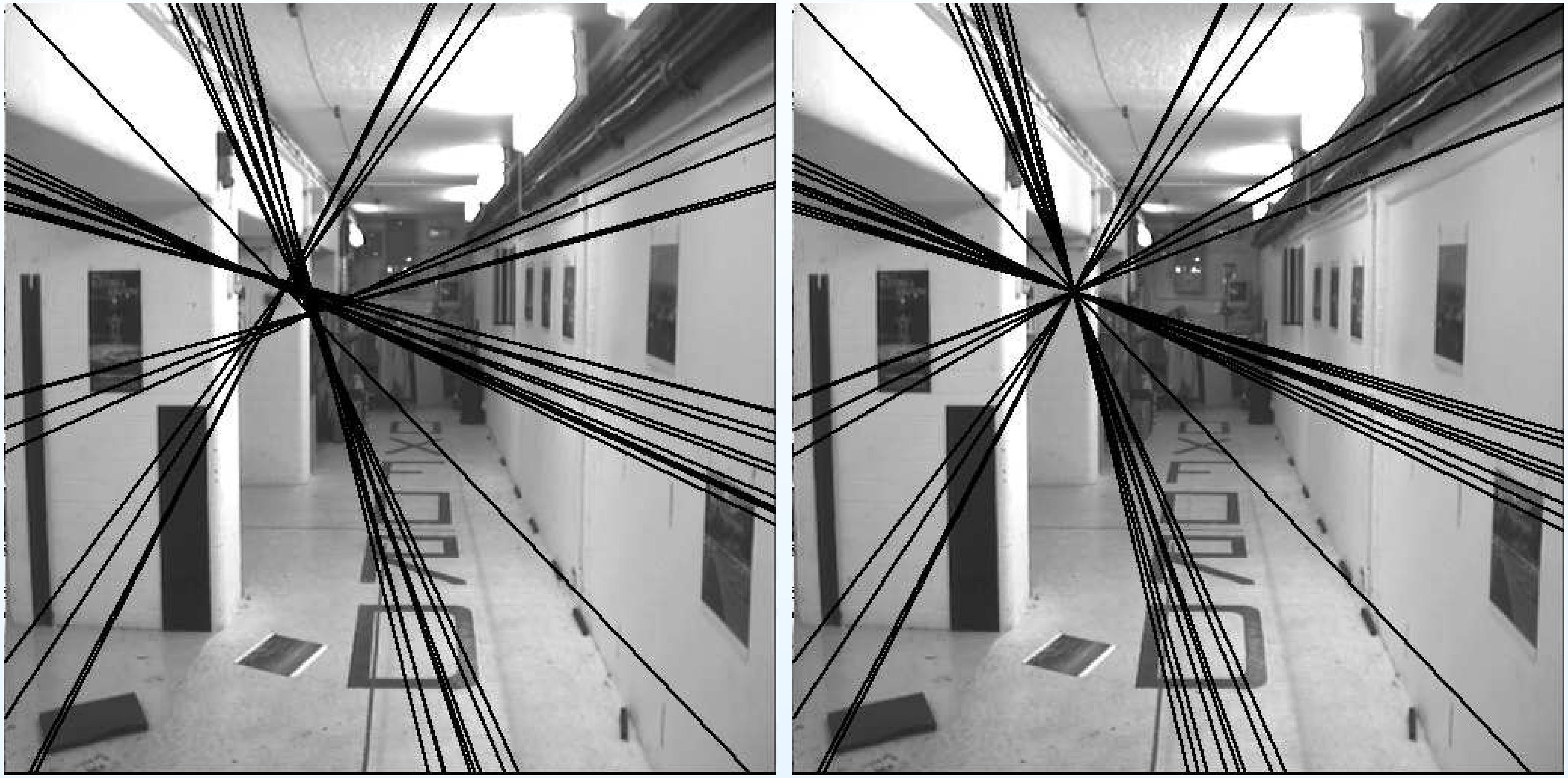
$$\mathrm{x}'^{\top}\mathrm{F}\mathrm{x} = 0 \text{ becomes } (x'x, x'y, x', y'x, y'y, y', x, y, 1)^{\top} f = 0.$$

For n corresponding points we get the set of homogeneous equations:

$$Af = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} f = \mathbf{0}.$$

The least-squares solution can be found using the SVD of A i.e. f is the singular vector of A with the smallest singular value.

Enforcing the singularity constraint



F found by solving the set of linear equations does not guarantee that F has rank 2 and thus is singular.

Enforcing the singularity constraint

To enforce rank 2 on F, replace F with F' where F' minimizes the Frobenius norm $\|\mathrm{F} - \mathrm{F}'\|_{Frobenius}$.

$$\|\mathrm{M}\|^2_{Frobenius} = \sum_{1}^{\min\{m,n\}} \sigma_i^2$$

with σ_n being the singular values of M.

This can be solved with the SVD of F.
Given $\mathrm{F} = \mathrm{UDV}^\top$and $\sigma_1 > \sigma_2$ are the two largest singular values of F then:

$$F' = U \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^\top.$$

7-point algorithm

- Using the singularity constraint we can also compute F when A has rank seven and is made of seven point correspondences.
- In this case the solution to $\mathrm{Af} = 0$ becomes two-dimensional. The solution is in the form:

$$\mathrm{F} = \alpha \mathrm{F}_1 + (1 - \alpha)\mathrm{F}_2$$

  where F_1 and F_2 are the matrices corresponding to the generators of the right null-space f_1 and f_2.
- Note that the singularity constraint enforces det $\mathrm{F} = 0$ thus det($\mathrm{F} = \alpha \mathrm{F}_1 + (1 - \alpha)\mathrm{F}_2$)=0. This gives a cubic polynomial in α from which we can solve for α.
- From this we get one or three real solutions for α. Given these solutions, we can put them in $\mathrm{F} = \alpha \mathrm{F}_1 + (1 - \alpha)\mathrm{F}_2$ to retrieve the F's.

Normalization

- When computing the fundamental matrix (using the algorithms described thus far) normalizing the points is the key to good performance.

Normalization

- When computing the fundamental matrix (using the algorithms described thus far) normalizing the points is the key to good performance.
- Translate the points so that the centroid of the reference points is at the origin.

Normalization

- When computing the fundamental matrix (using the algorithms described thus far) normalizing the points is the key to good performance.
- Translate the points so that the centroid of the reference points is at the origin.
- Scale the points so that the RMS distance of the points from the origin is $\sqrt{2}$.

Normalization

- When computing the fundamental matrix (using the algorithms described thus far) normalizing the points is the key to good performance.
- Translate the points so that the centroid of the reference points is at the origin.
- Scale the points so that the RMS distance of the points from the origin is $\sqrt{2}$.
- let T and T' be these appropriate normalization (translation and scaling) matrices. Then estimate F on the points $\mathrm{x}_i = \mathrm{T}\mathrm{x}_\mathrm{i}$ and $\mathrm{x}'_i = \mathrm{T}'\mathrm{x}'_\mathrm{i}$.

Normalization

- When computing the fundamental matrix (using the algorithms described thus far) normalizing the points is the key to good performance.
- Translate the points so that the centroid of the reference points is at the origin.
- Scale the points so that the RMS distance of the points from the origin is $\sqrt{2}$.
- let T and T' be these appropriate normalization (translation and scaling) matrices. Then estimate F on the points $\mathrm{x}_i = \mathrm{T}\mathrm{x}_\mathrm{i}$ and $\mathrm{x}'_i = \mathrm{T}'\mathrm{x}'_\mathrm{i}$.
- Transform the solution for F back to the unnormalized frame with $\mathrm{F} = \mathrm{T}'^{\top}\mathrm{F}\mathrm{T}$.

The Gold standard method and reprojection

- Make an initial estimate of F using the normalized 8-point algorithm.

The Gold standard method and reprojection

- Make an initial estimate of F using the normalized 8-point algorithm.
- From this F extract two camera matrices $\mathrm{P} = [\mathrm{I}|\mathbf{0}]$ and $\mathrm{P} = [[\mathbf{e}']_\times |\mathbf{e}']$ with $\mathbf{e}'$ obtained from F

The Gold standard method and reprojection

- Make an initial estimate of F using the normalized 8-point algorithm.
- From this F extract two camera matrices $\mathrm{P} = [\mathrm{I}|\mathbf{0}]$ and $\mathrm{P} = [[\mathbf{e}']_\times |\mathbf{e}']$ with $\mathbf{e}'$ obtained from F
- From the correspondences and F estimate the 3D positions of the real-world points relating to the imaged points.

The Gold standard method and reprojection

- Make an initial estimate of F using the normalized 8-point algorithm.
- From this F extract two camera matrices $\mathrm{P} = [\mathrm{I}|\mathbf{0}]$ and $\mathrm{P} = [[\mathbf{e}']_\times |\mathbf{e}']$ with $\mathbf{e}'$ obtained from F
- From the correspondences and F estimate the 3D positions of the real-world points relating to the imaged points.
- Given this 3D points project them back to both image planes using the estimate of the camera projection matrices (that were based on F).

The Gold standard method and reprojection

- Make an initial estimate of F using the normalized 8-point algorithm.
- From this F extract two camera matrices $\mathrm{P} = [\mathrm{I}|\mathbf{0}]$ and $\mathrm{P} = [[\mathbf{e}']_\times |\mathbf{e}']$ with $\mathbf{e}'$ obtained from F
- From the correspondences and F estimate the 3D positions of the real-world points relating to the imaged points.
- Given this 3D points project them back to both image planes using the estimate of the camera projection matrices (that were based on F).
- The difference in the real points and the backprojected points is what we want to minimize by varying the camera matrices P and P' and the coordinates of the 3D points (and thus also implicitly by varying F).

The Gold standard method

- Minimize a geometric distance (cost):

  where d is differentiable in parameters relating to F, x_i and x'_i are the correspondence points and $\hat{\mathrm{x}}_i$ and $\hat{\mathrm{x}}'_i$ are their reprojections given the current F.

The Gold standard method

- Minimize a geometric distance (cost):

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

where d is differentiable in parameters relating to F, x_i and x'_i are the correspondence points and $\hat{\mathrm{x}}_i$ and $\hat{\mathrm{x}}'_i$ are their reprojections given the current F.

The Gold standard method

- Minimize a geometric distance (cost):

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

  where d is differentiable in parameters relating to F, x_i and x'_i are the correspondence points and $\hat{\mathrm{x}}_i$ and $\hat{\mathrm{x}}'_i$ are their reprojections given the current F.
- Because the cost is differentiable we can locally approximate it with a linear function.

The Gold standard method

- Minimize a geometric distance (cost):

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

where d is differentiable in parameters relating to F, x_i and x'_i are the correspondence points and $\hat{\mathrm{x}}_i$ and $\hat{\mathrm{x}}'_i$ are their reprojections given the current F.

- Because the cost is differentiable we can locally approximate it with a linear function.
- This means we can use Gauss-Newton, gradient descent or preferably Levenberg-Marquardt to iteratively find a solution.

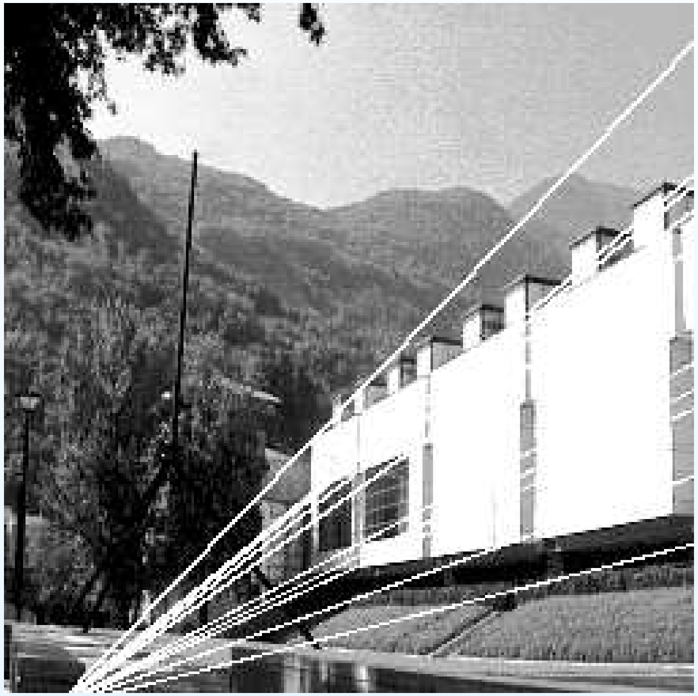
The Gold standard method

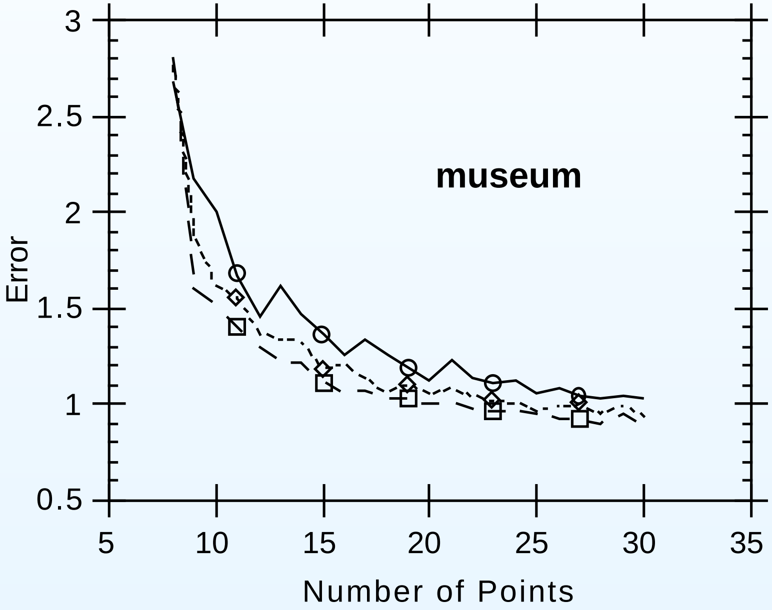
- Minimize a geometric distance (cost):

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

  where d is differentiable in parameters relating to F, x_i and x'_i are the correspondence points and $\hat{\mathrm{x}}_i$ and $\hat{\mathrm{x}}'_i$ are their reprojections given the current F.
- Because the cost is differentiable we can locally approximate it with a linear function.
- This means we can use Gauss-Newton, gradient descent or preferably Levenberg-Marquardt to iteratively find a solution.

A comparison

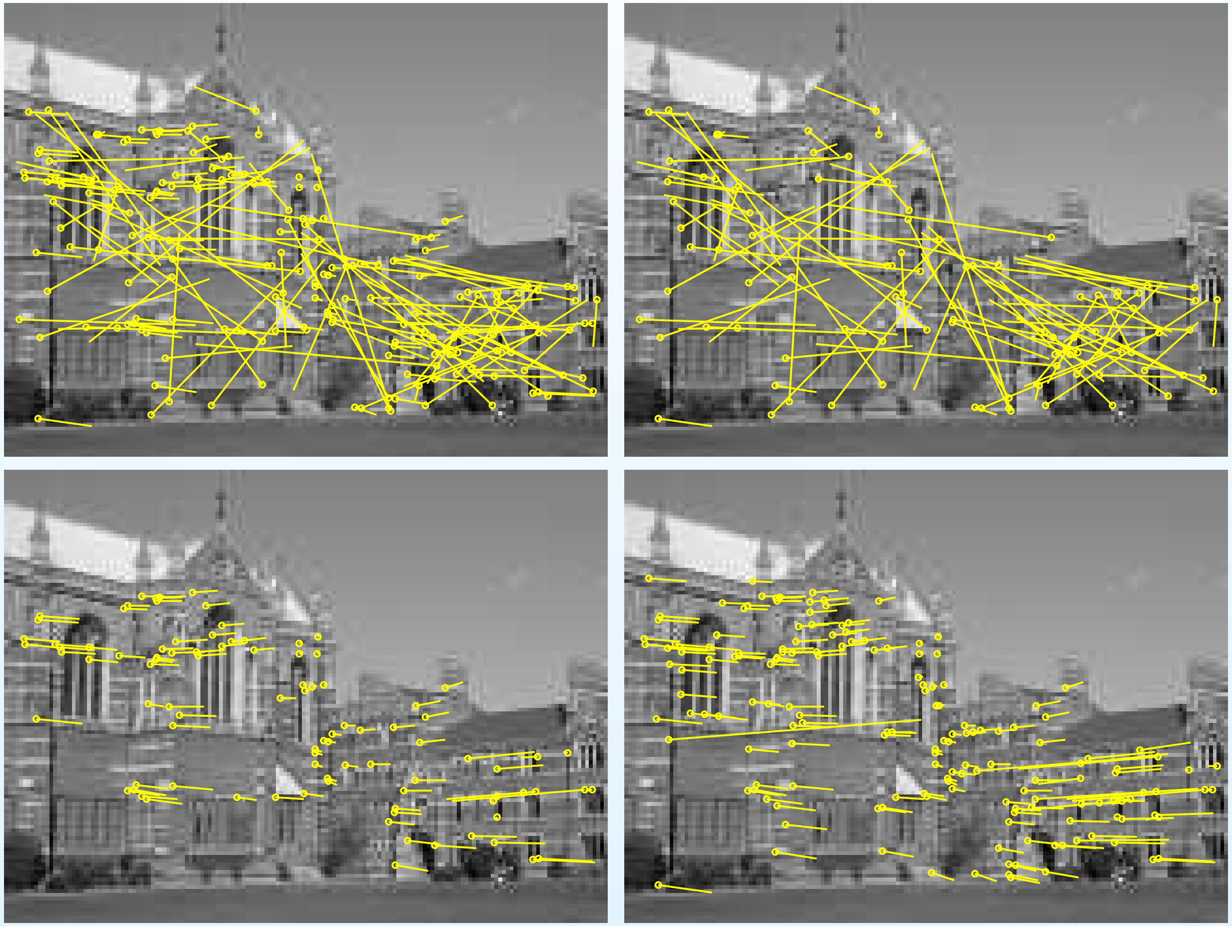




$$\frac{1}{N}\sum_i d(x_i', Fx_i)^2 + d(x_i, F^T x_i')^2$$

Automatic computation of F

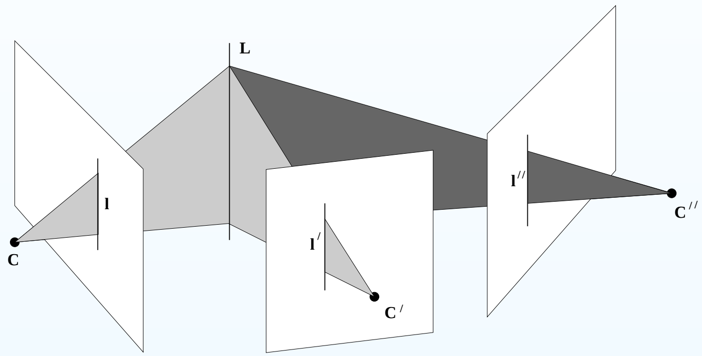
Look at Hartley and Zisserman page 291 Algorithm 11.4.



Using F for image rectification

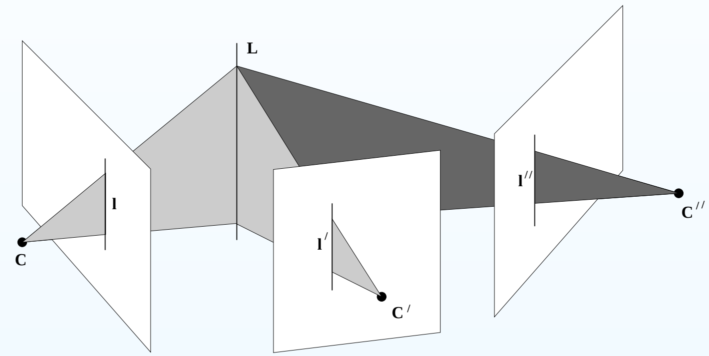
Look at Hartley and Zisserman page 307 Algorithm 11.12.3

The Trifocal tensor



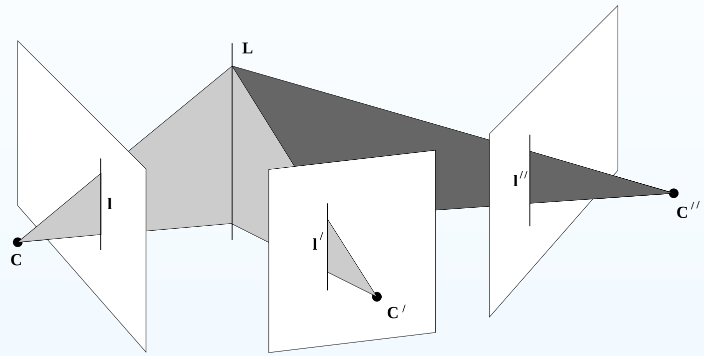
- For three camera's the trifocal tensor is what the Fundamental matrix is for two camera's.

The Trifocal tensor



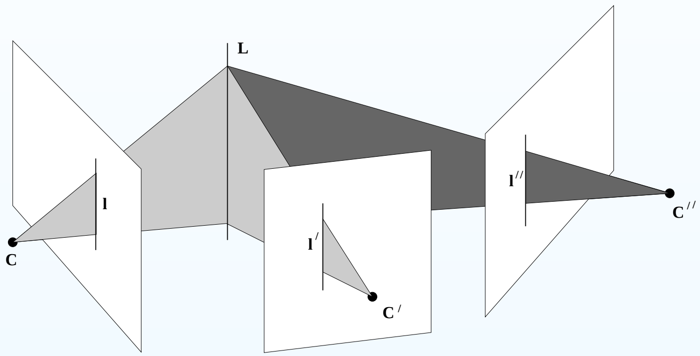
- For three camera’s the trifocal tensor is what the Fundamental matrix is for two camera’s.
- It captures the complete (projective)geometric relations between the three views.

The Trifocal tensor



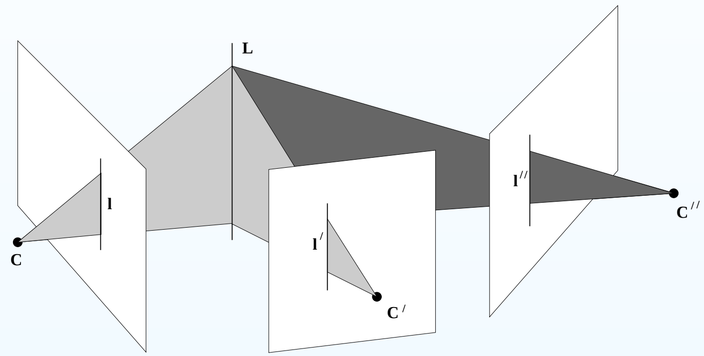
- For three camera's the trifocal tensor is what the Fundamental matrix is for two camera's.
- It captures the complete (projective)geometric relations between the three views.
- It is uniquely defined by the internal and external camera properties.

The Trifocal tensor



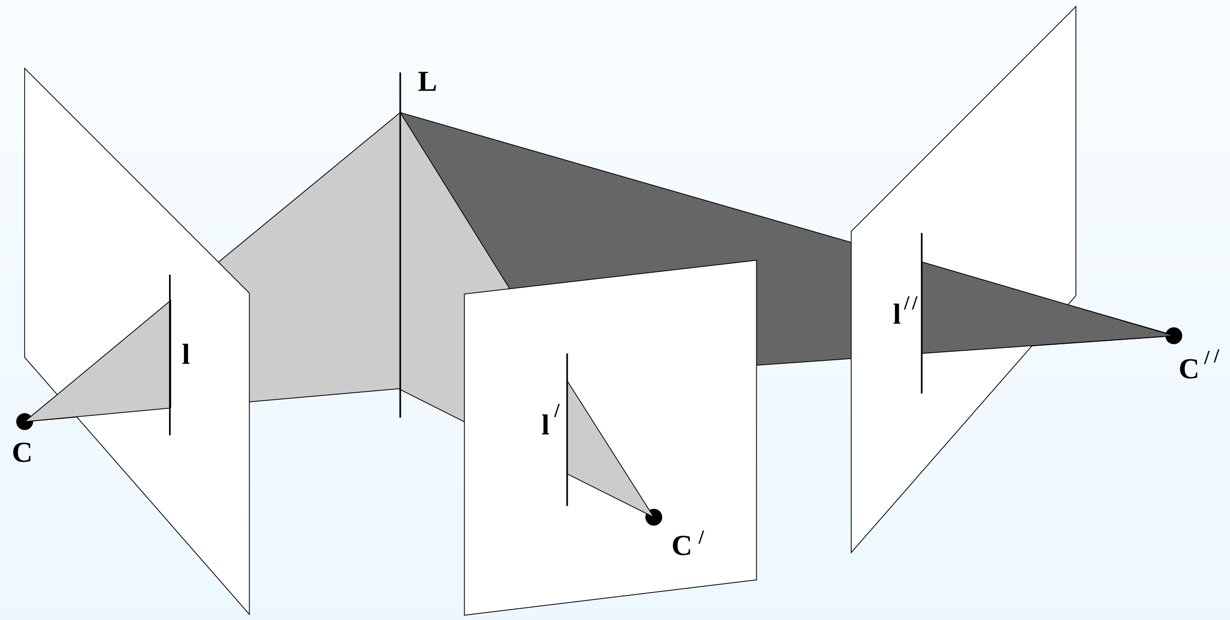
- For three camera’s the trifocal tensor is what the Fundamental matrix is for two camera’s.
- It captures the complete (projective)geometric relations between the three views.
- It is uniquely defined by the internal and external camera properties.
- However, it can be computed directly from image correspondence without knowledge of the internal and external camera matrices.

The Trifocal tensor



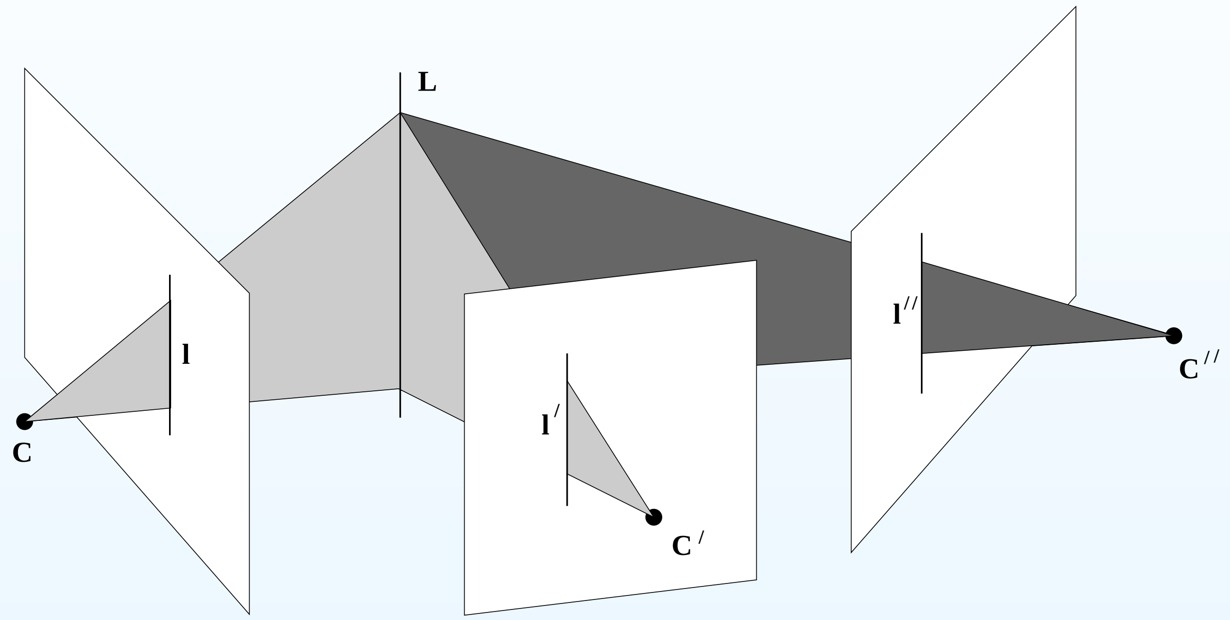
- For three camera's the trifocal tensor is what the Fundamental matrix is for two camera's.
- It captures the complete (projective)geometric relations between the three views.
- It is uniquely defined by the internal and external camera properties.
- However, it can be computed directly from image correspondence without knowledge of the internal and external camera matrices.

The Trifocal tensor



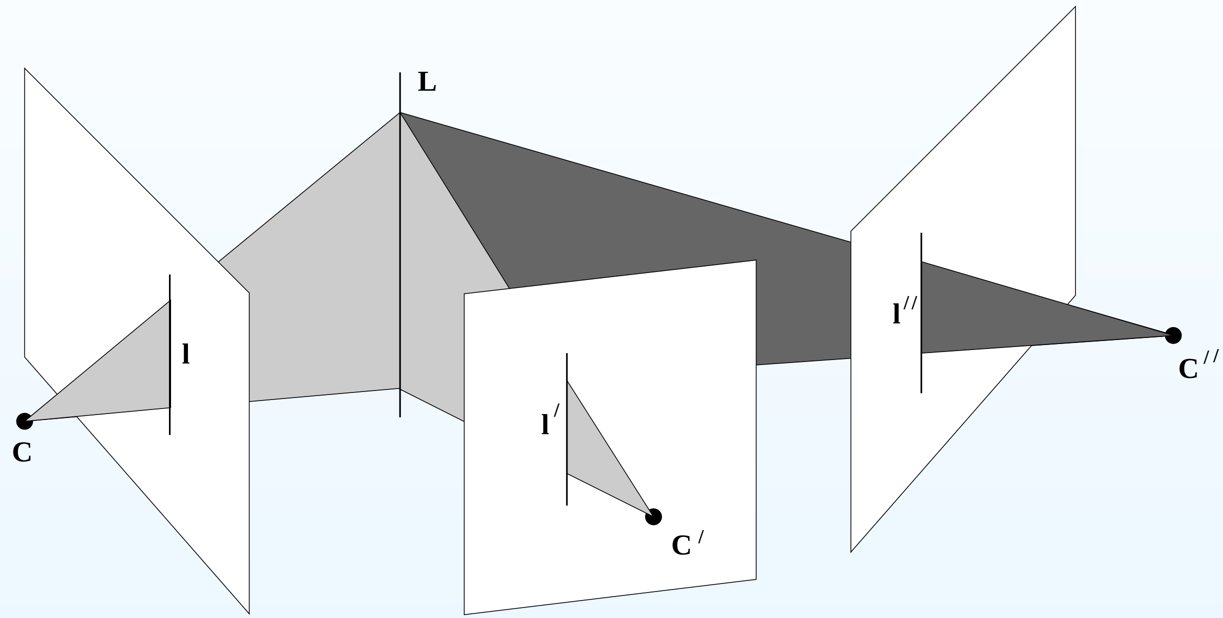
- Image lines back project to scene planes.

The Trifocal tensor



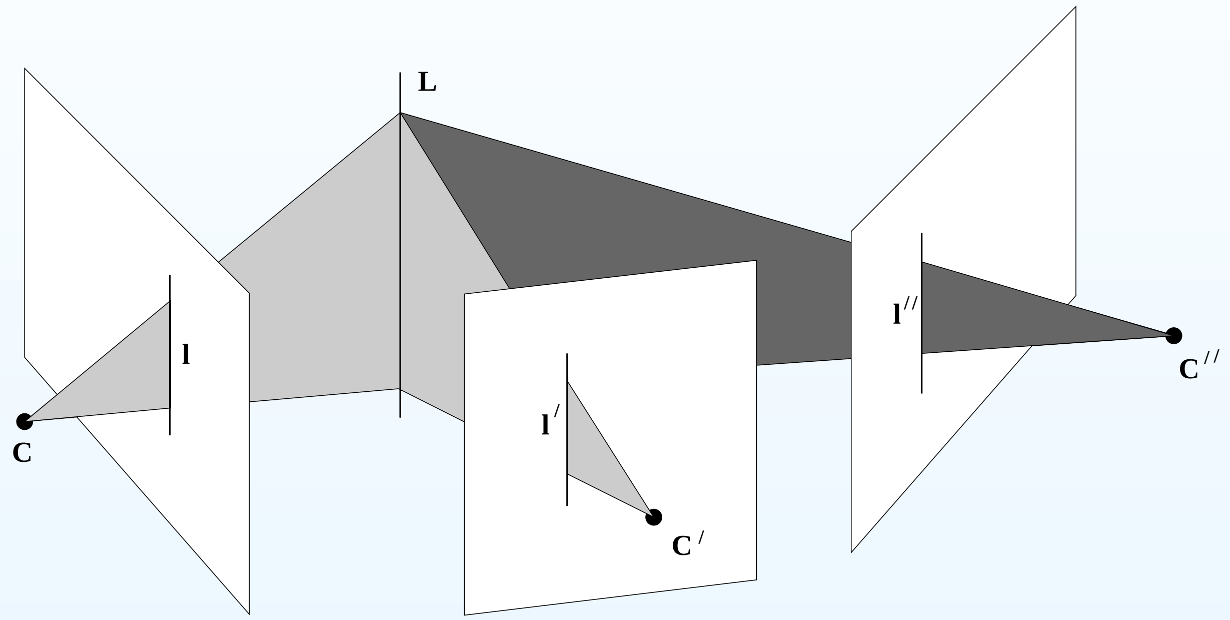
- Image lines back project to scene planes.
- In general three planes do not intersect in a single line.

The Trifocal tensor



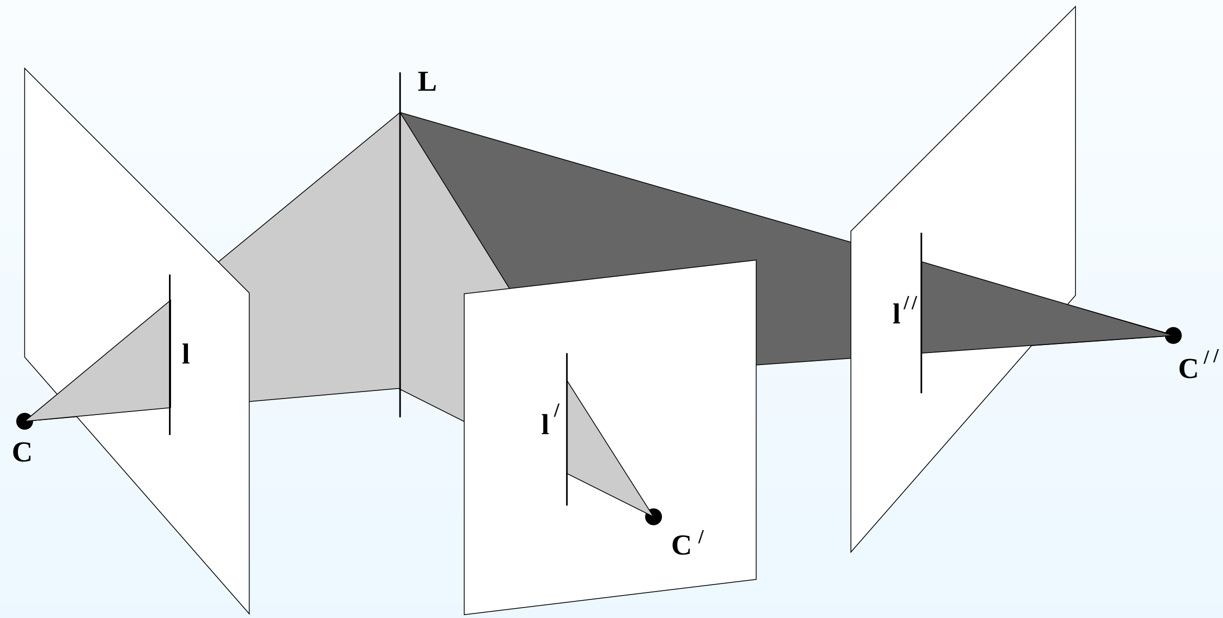
- Image lines back project to scene planes.
- In general three planes do not intersect in a single line.
- Thus, the fact that three image lines correspond to the same scene line, provides a geometric constraint.

The Trifocal tensor



- Image lines back project to scene planes.
- In general three planes do not intersect in a single line.
- Thus, the fact that three image lines correspond to the same scene line, provides a geometric constraint.

The Trifocal tensor



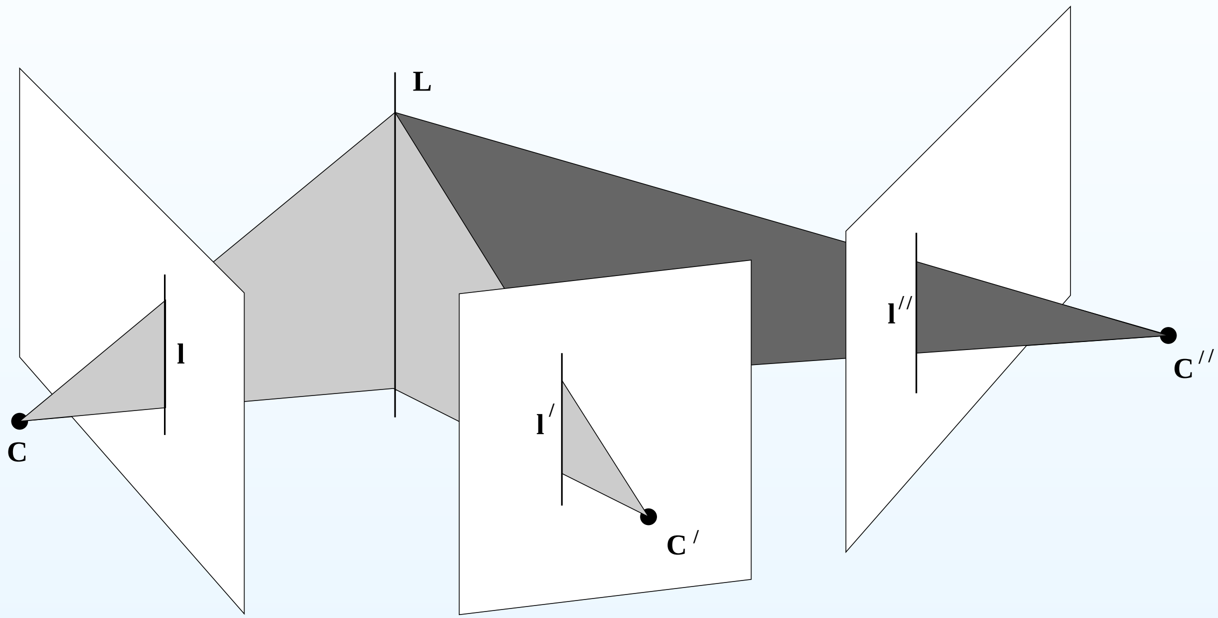
Three corresponding image lines: $\mathrm{l} \leftrightarrow \mathrm{l}' \leftrightarrow \mathrm{l}''$

Camera matrices (3x4) for the three views:

$$\mathrm{P} = [\mathrm{I} \,|\, 0]\,, \quad \mathrm{P}' = [\mathrm{A} \,|\, \mathrm{a}_4]\,, \quad \mathrm{P}'' = [\mathrm{B} \,|\, \mathrm{b}_4]$$

$\mathrm{a}_4 = \mathrm{e}'$ and $\mathrm{b}_4 = \mathrm{e}''$ are the epipoles arising from the first camera center C thus: $\mathrm{e}' = \mathrm{P}'\mathrm{C}$ and $\mathrm{e}'' = \mathrm{P}''\mathrm{C}$

The Trifocal tensor

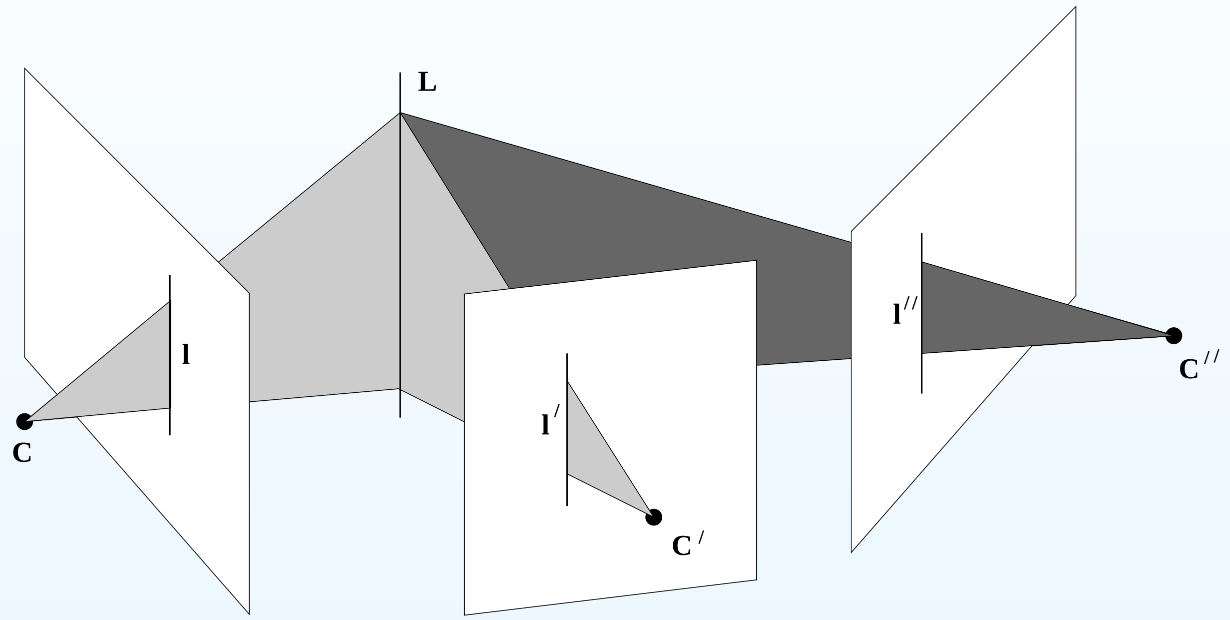


The lines: $\mathrm{l} \leftrightarrow \mathrm{l}' \leftrightarrow \mathrm{l}''$ back project to the planes:

$$\pi = \mathrm{P}^\top \mathrm{l} = \begin{pmatrix} \mathrm{l} \\ 0 \end{pmatrix}, \pi' = \mathrm{P}'^\top \mathrm{l}' = \begin{pmatrix} \mathrm{A}^\top \mathrm{l}' \\ \mathrm{a}_4^\top \mathrm{l}' \end{pmatrix},$$

$$\pi'' = \mathrm{P}''^\top \mathrm{l}'' = \begin{pmatrix} \mathrm{B}^\top \mathrm{l}'' \\ \mathrm{b}_4^\top \mathrm{l}'' \end{pmatrix}.$$

The Trifocal tensor

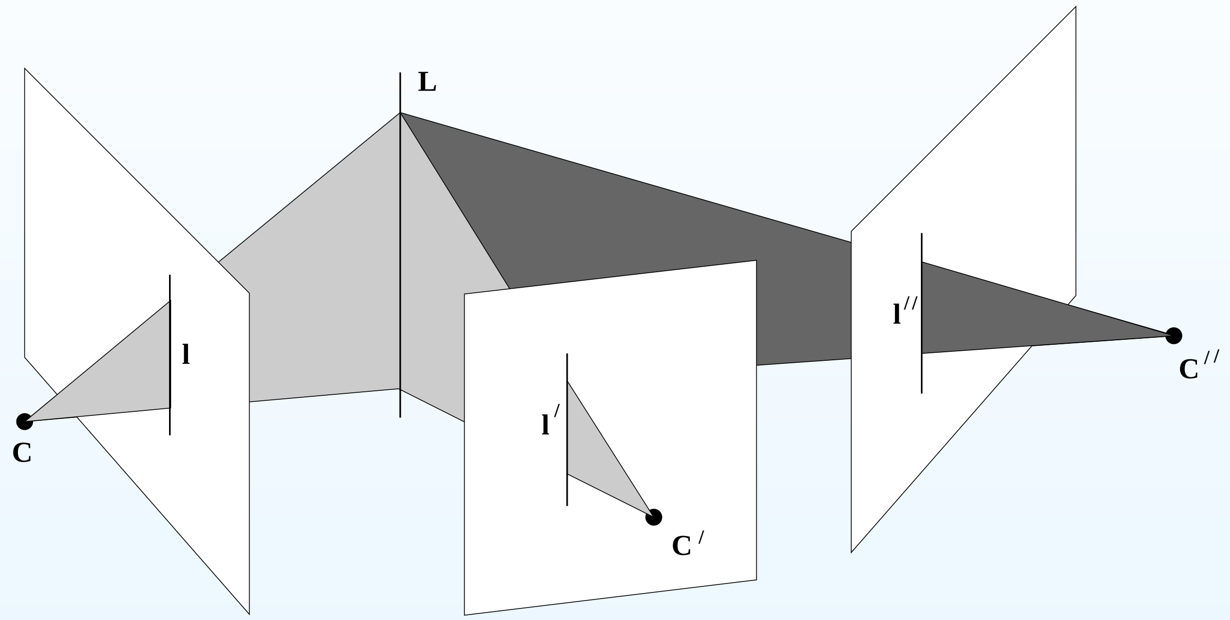


The planes π, π' and π'' coincide in the line L

This can be expressed algebraically with:

$$\mathrm{M} = [\pi, \pi', \pi''], \quad \det(\mathrm{M}) = 0$$

The Trifocal tensor

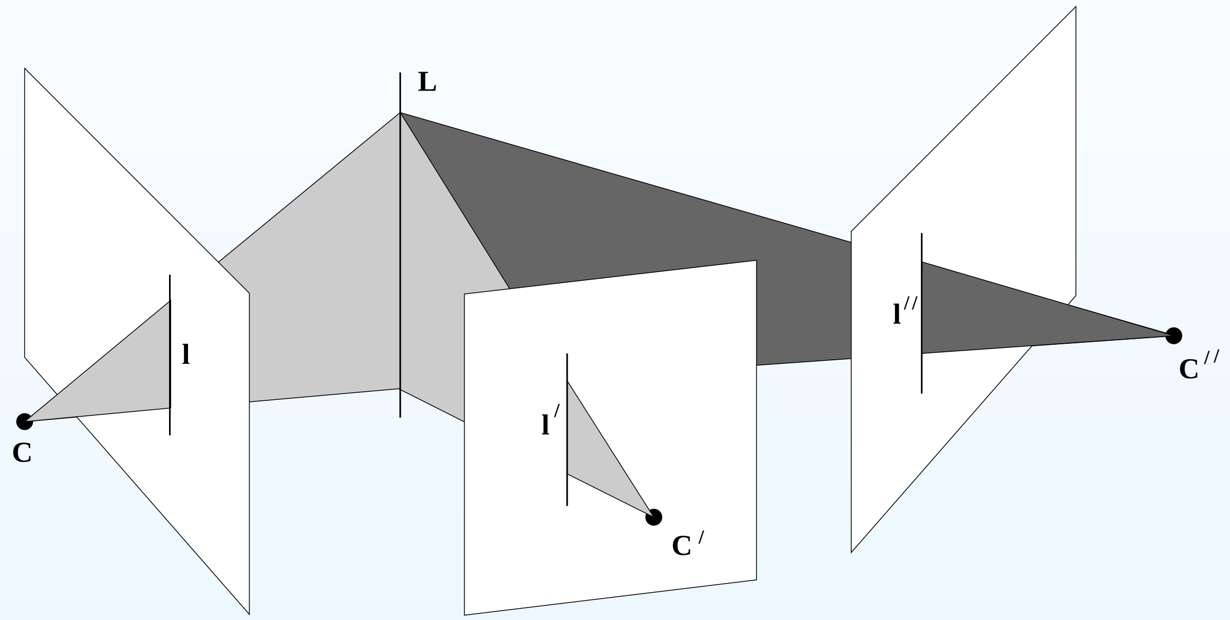


$$M = [m_1, m_2, m_3] = \begin{bmatrix} l & A^\top l' & B^\top l'' \\ 0 & a_4^\top l' & b_4^\top l'' \end{bmatrix}$$

Since $\det(M) = 0$ The columns must be linearly dependent.

Thus, $m_1 = \alpha m_2 + \beta m_3$

The Trifocal tensor

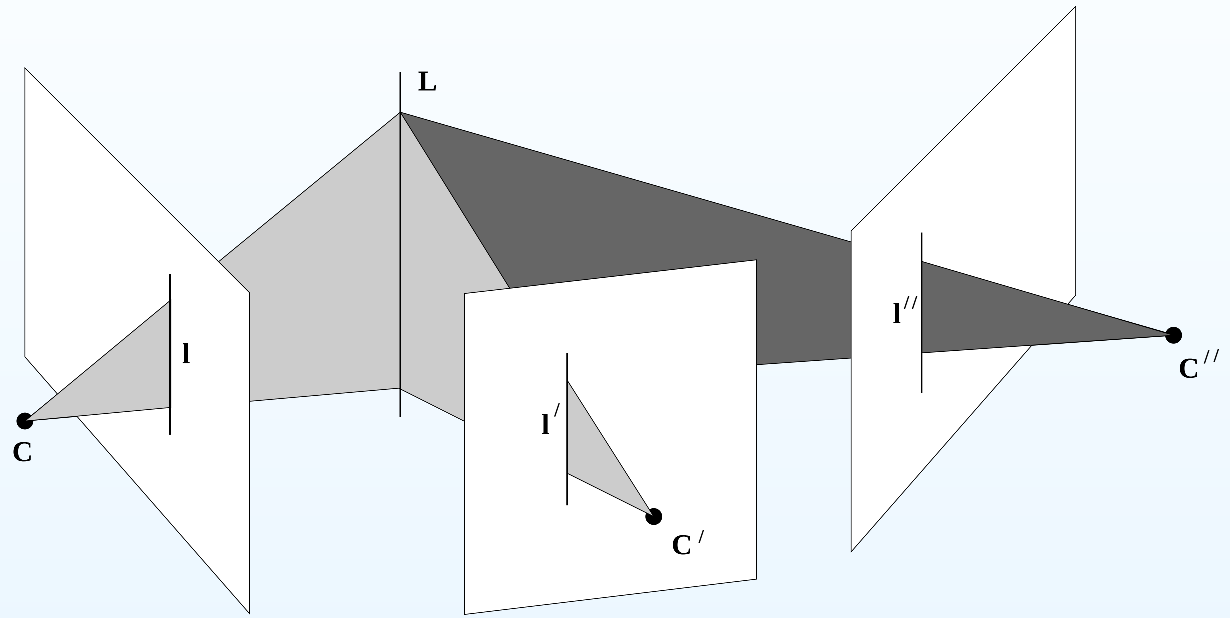


$$\mathrm{M} = [\mathrm{m}_1, \mathrm{m}_2, \mathrm{m}_3] = \begin{bmatrix} \mathrm{l} & \mathrm{A}^\top \mathrm{l}' & \mathrm{B}^\top \mathrm{l}'' \\ 0 & \mathrm{a}_4^\top \mathrm{l}' & \mathrm{b}_4^\top \mathrm{l}'' \end{bmatrix}$$

Since the bottom left element of $\mathrm{M} = 0$ it follows that:

$\alpha = k(\mathrm{b}_4^\top \mathrm{l}'')$ and $\beta = -k(\mathrm{a}_4^\top \mathrm{l}')$

The Trifocal tensor

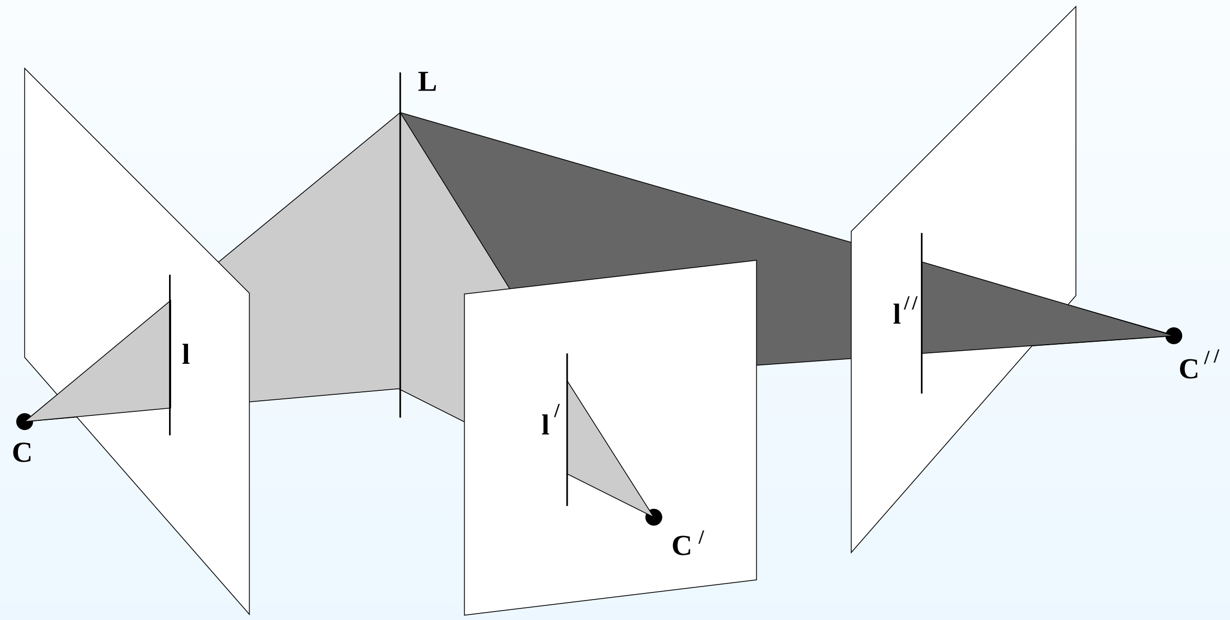


$$M = [m_1, m_2, m_3] = \begin{bmatrix} l & A^\top l' & B^\top l'' \\ 0 & a_4^\top l' & b_4^\top l'' \end{bmatrix}$$

For the top three vectors of M this gives:

$$l = (b_4^\top l'')A^\top l' - (a_4^\top l')B^\top l'' = (l''^\top b_4)A^\top l' - (l'^\top a_4)B^\top l''$$

The Trifocal tensor

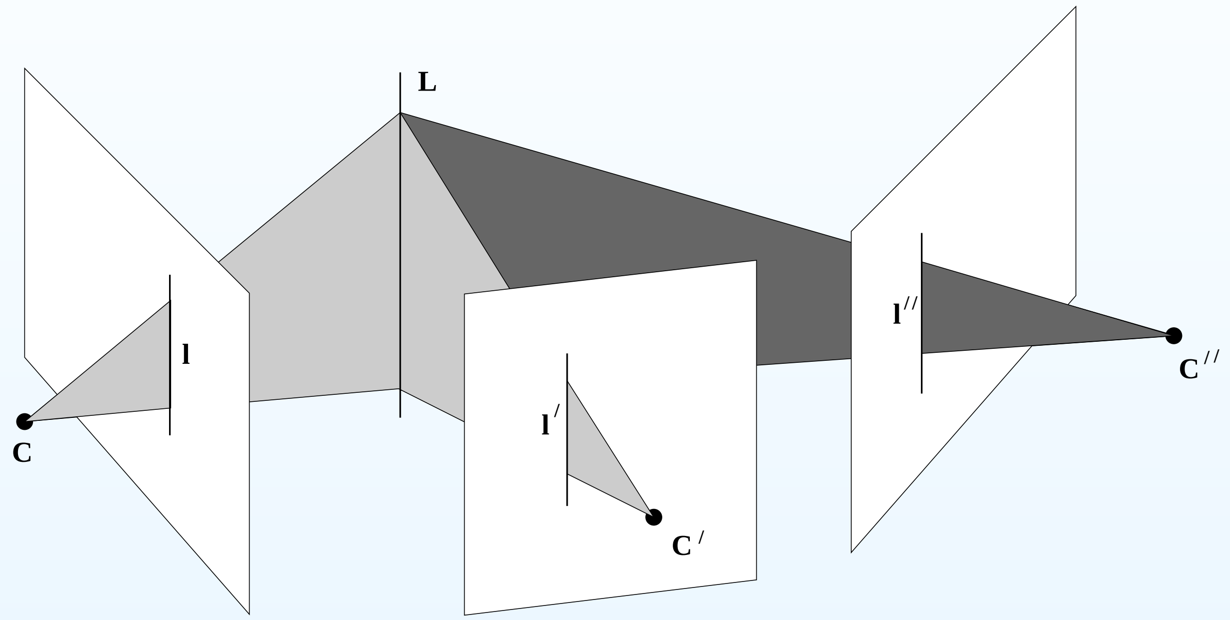


For the i-th element of we have:

$$\mathrm{l_i} = \mathrm{l}''^{\top}(\mathrm{b_4 a_i^{\top}})\mathrm{l}' - \mathrm{l}'^{\top}(\mathrm{a_4 b_i^{\top}})\mathrm{l}''$$

$$\mathrm{l_i} = \mathrm{l}'^{\top}(\mathrm{a_i b_4^{\top}})\mathrm{l}'' - \mathrm{l}'^{\top}(\mathrm{a_4 b_i^{\top}})\mathrm{l}''$$

The Trifocal tensor

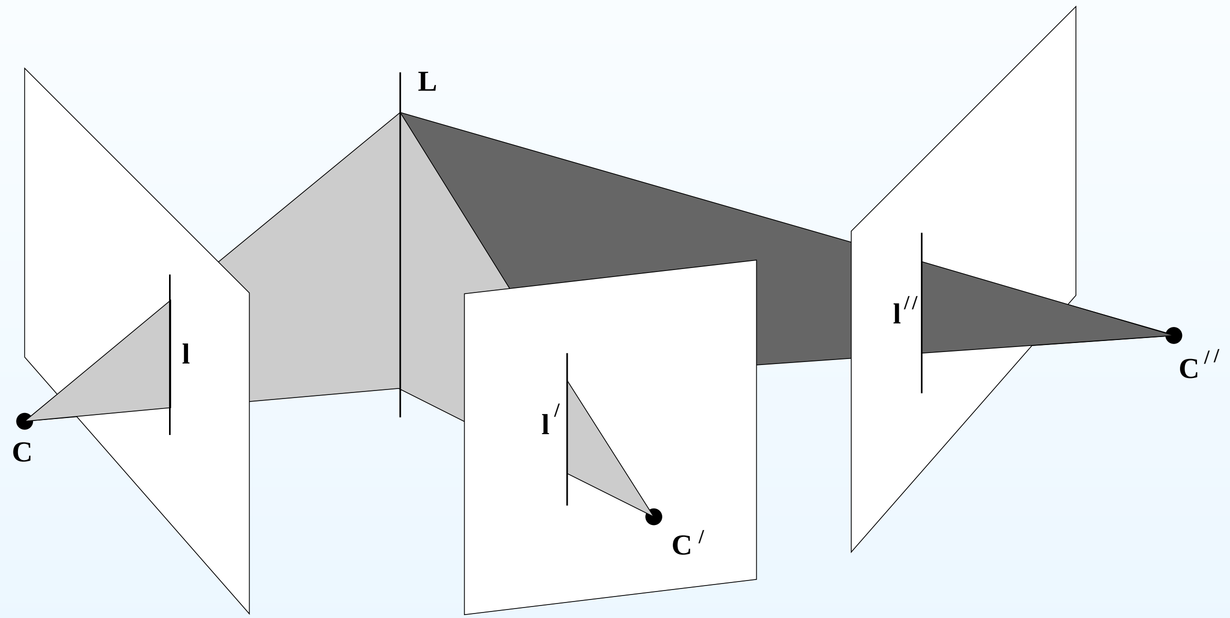


$$l_i = l'^\top (a_i b_4^\top) l'' - l'^\top (a_4 b_i^\top) l''$$

$$T_i = a_i b_4^\top - a_4 b_i^\top$$

$$l_i = l'^\top T_i l''$$

The Trifocal tensor

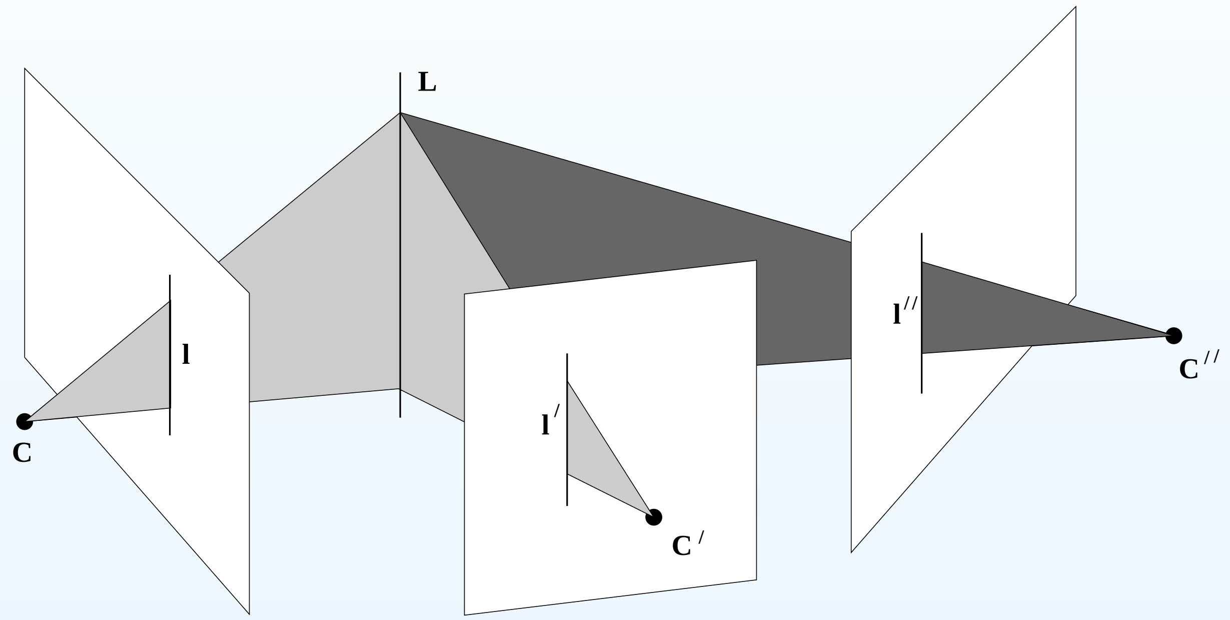


The set of the three matrices $\mathrm{T}_1, \mathrm{T}_2, \mathrm{T}_3$ constitute the trifocal tensor in matrix notation.

$$\mathrm{l}^\top = (\mathrm{l}_\mathrm{i} = \mathrm{l}'^\top \mathrm{T}_1 \mathrm{l}'', \mathrm{l}_\mathrm{i} = \mathrm{l}'^\top \mathrm{T}_2 \mathrm{l}'', \mathrm{l}_\mathrm{i} = \mathrm{l}'^\top \mathrm{T}_3 \mathrm{l}'') =$$

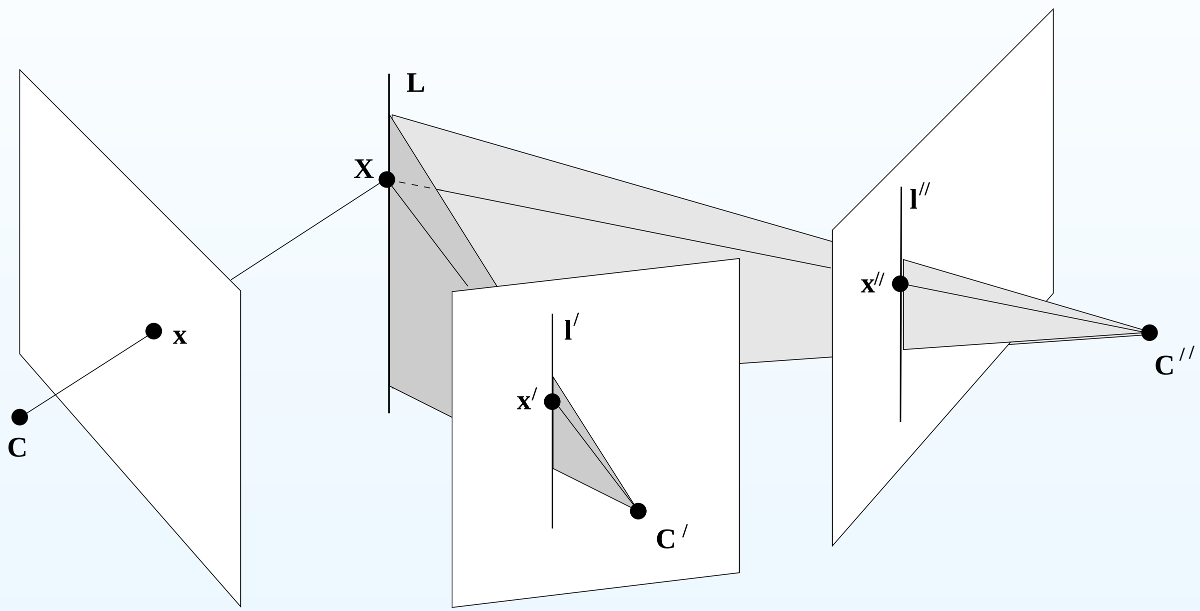
$$\mathrm{l}'^\top \left[\begin{array}{ccc} T_1 & T_2 & T_3 \end{array} \right] \mathrm{l}''$$

Line-Line-Line correspondence



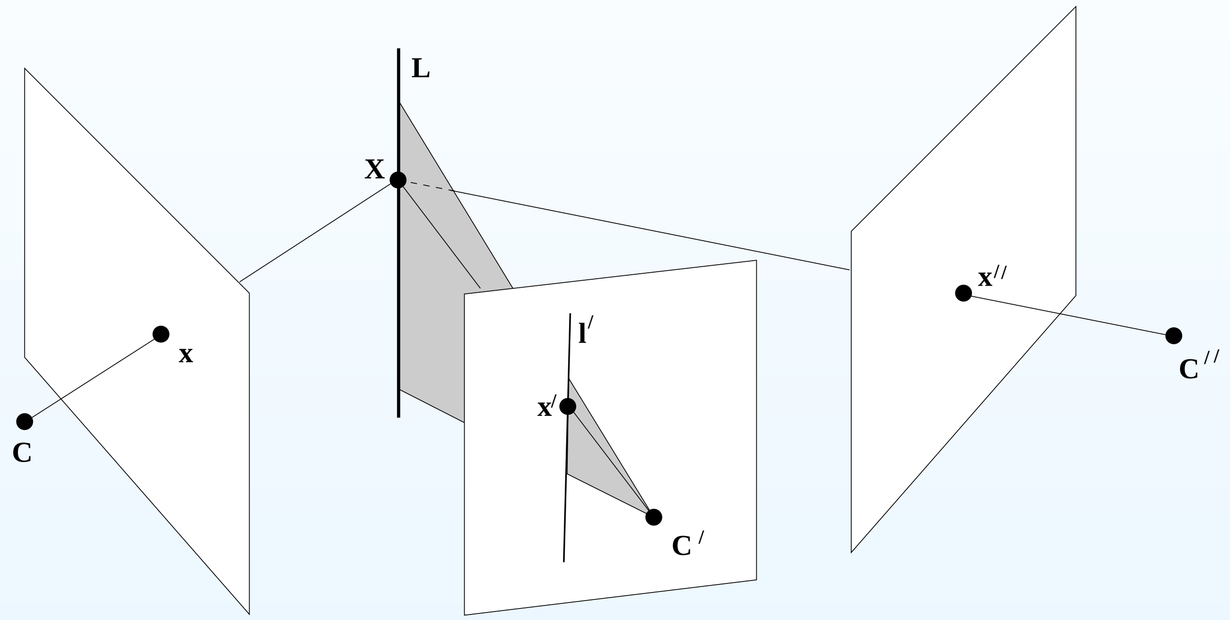
$$\mathbf{l}^{\top} = \mathbf{l}'^{\top} \left[\begin{array}{ccc} T_1 & T_2 & T_3 \end{array} \right] \mathbf{l}''$$

Point-Line-Line correspondence



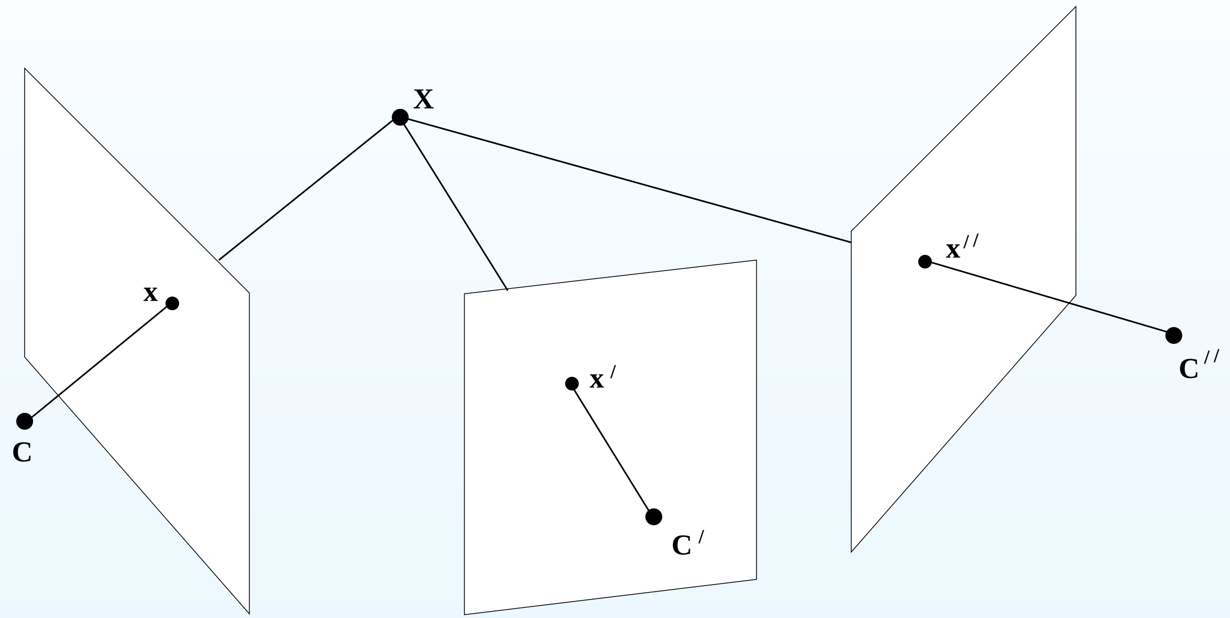
$$\mathrm{l}'^{\top}\left(\sum_i x^i \mathrm{T}_i\right)\mathrm{l}'' = 0$$

Point-Line-Point correspondence



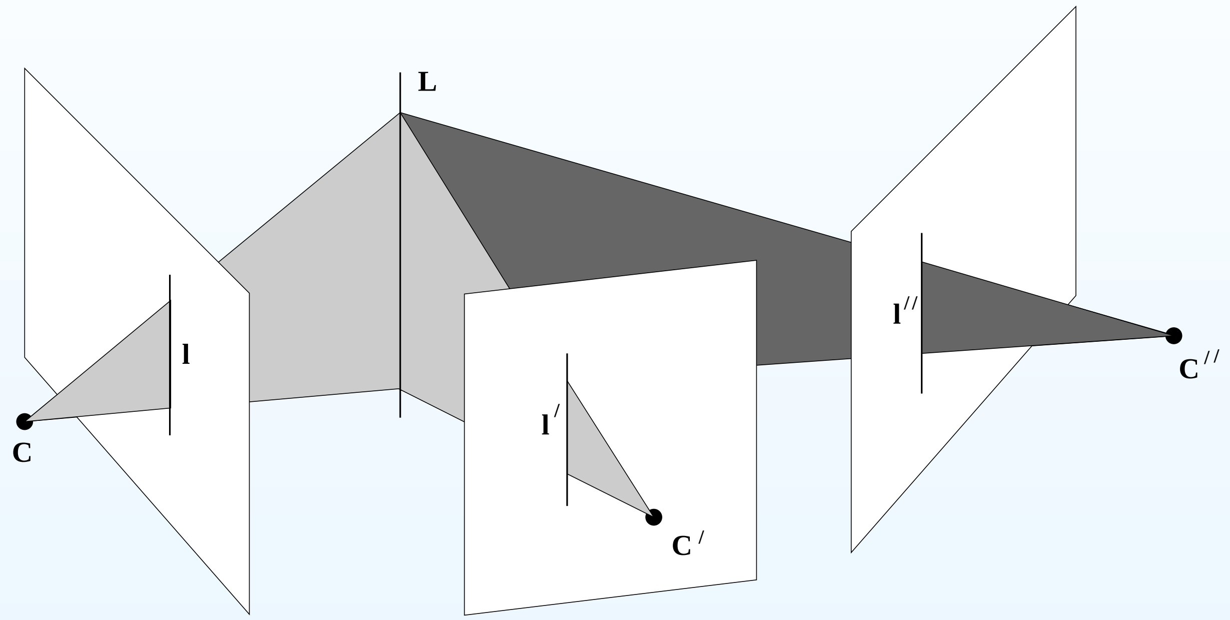
$$\mathrm{l}'^{\top}\left(\sum_i x^i \mathrm{T}_i\right)[x'']_{\times} = \mathbf{0}^{\top}$$

Point-Point-Point correspondence



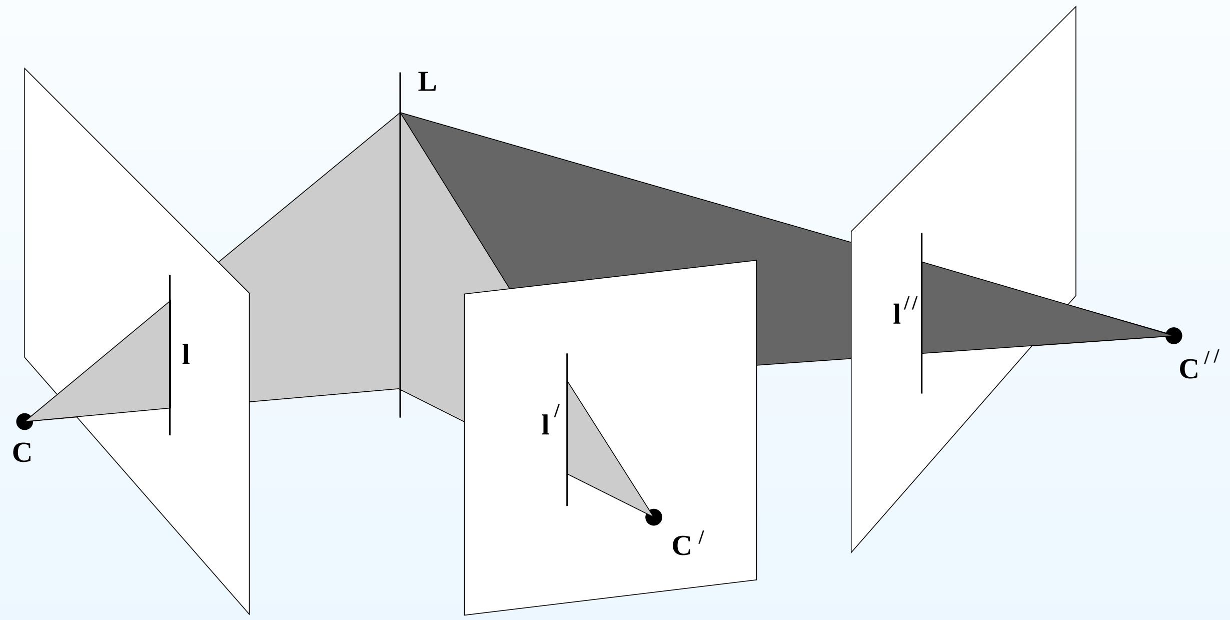
$$[x']_\times \left(\sum_i x^i \mathrm{T}_i \right) [x'']_\times = 0_{3\times 3}$$

Extracting the fundamental matrix



$$F_{21} = [e']_\times [T_1, T_2, T_3] e''$$

Retrieving the camera matrices



$$P' = [[T_1, T_2, T_3]e''|e']$$

Retrieving food