Multiple View Geometry

Richard Hartley and Andrew Zisserman

CVPR June 1999

Image Transformations

Translation

$$\mathbf{x} \mapsto \mathbf{x} + \mathbf{t} \quad \text{where} \quad \mathbf{x}, \mathbf{t} \in \mathbb{R}^2$$

Can be written in homogeneous coordinates as

$$\begin{aligned} \mathbf{x}' &= \mathtt{T}\,\hat{\mathbf{x}} \\ &= [\mathtt{I} \mid \mathbf{t}] \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \end{aligned}$$

where $\hat{\mathbf{x}}$ represents $\mathbf{x}$ in homogeneous coordinates.

This can be written entirely in homogeneous coordinates as

$$\hat{\mathbf{x}}' = \begin{bmatrix} \mathtt{I} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

Rotation and translation

$$\mathbf{x} \mapsto \mathtt{R}\mathbf{x} + \mathbf{t} \quad \text{where} \quad \mathbf{x}, \mathbf{t} \in \mathbb{R}^2$$

and

$$\mathtt{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} .$$

Can be written in homogeneous coordinates as

$$\begin{aligned} \mathbf{x}' &= \mathtt{T}\,\hat{\mathbf{x}} \\ &= [\mathtt{R} \mid \mathbf{t}] \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} . \end{aligned}$$

This can be written entirely in homogeneous coordinates as

$$\hat{\mathbf{x}}' = \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

Rotation, translation and scaling

$$\mathbf{x} \mapsto s(\mathtt{R}\mathbf{x} + \mathbf{t})$$

Can be written in homogeneous coordinates as

$$\begin{aligned}\mathbf{x}' &= \mathtt{T}\,\hat{\mathbf{x}} \\ &= s[\mathtt{R} \mid \mathbf{t}] \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} .\end{aligned}$$

This can be written entirely in homogeneous coordinates as

$$\hat{\mathbf{x}}' = s \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

Affine transformation

$$\mathbf{x} \mapsto \mathtt{A}\mathbf{x} + \mathbf{t} \ .$$

Can be written in homogeneous coordinates as

$$\begin{aligned} \mathbf{x}' &= \mathtt{T}\,\hat{\mathbf{x}} \\ &= [\mathtt{A} \mid \mathbf{t}] \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \ . \end{aligned}$$

This can be written entirely in homogeneous coordinates as

$$\hat{\mathbf{x}}' = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

Preserving the plane at infinity

We note

- All these transformations *preserve the plane at infinity*.
- Plane at infinity is the set of points with homogeneous coordinates $\mathbf{x} = (x, y, 0)^\top$.
- Verify:

$$\begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ 0 \end{pmatrix}$$

- They map parallel lines to parallel lines.
- They are transformations of $\mathbb{R}^2$.

Notation

- From now on, we assume that all points are given in homogenous coordinates
- Points (in homogeneous coordinates) are denoted simply by $\mathbf{x}$, and not by $\hat{\mathbf{x}}$.
- The symbol $\approx$ means *equal up to scale*, or *equivalent as homogeneous vectors*.

Homographies

- Affine (infinity-preserving) mappings are sometimes not enough to capture the transformations that apply to images.
  - Views taken from different viewpoints.
  - Correspondences between world-plane and image.
- We need a transformation:

$$\begin{aligned}\mathbf{x}' &\approx \mathtt{H}\mathbf{x} \\ &= \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}\end{aligned}$$

- In non-homogeneous coordinates:

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{13}}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{13}}$$

- They are transformations of P^2 (the projective plane).

3D transformations

- Translations

$$\mathbf{x}' = \begin{bmatrix} \mathtt{I} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{x}$$

- Rotation and translation

$$\mathbf{x}' = \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{x}$$

- Affine transformation

$$\mathbf{x}' = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{x}$$

- Projective transformation

$$\mathbf{x}' \approx \mathtt{H}\mathbf{x}$$

3D rotations - Euler angles / rotation matrices

A 3D rotation is represented by a 3×3 matrix (in non-homogeneous coordinates).

$$\mathtt{R} = \mathtt{R}_z(\psi)\mathtt{R}_y(\phi)\mathtt{R}_x(\theta) \ .$$

where

$$\mathtt{R}_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathtt{R}_y(\phi) = \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix}$$

$$\mathtt{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$$

3D rotations - quaternions

A 3D rotation may be represented by a unit quaternion. A *quaternion* is a 4-vector

$$\mathbf{q} = (q_0, q_1, q_2, q_3)$$

sometimes written as

$$\mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \ .$$

where

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \ .$$

Otherwise stated:

$$\begin{aligned}
\mathbf{ij} &= \mathbf{k} \\
\mathbf{ji} &= -\mathbf{k} \\
\mathbf{jk} &= \mathbf{i} \\
\mathbf{kj} &= -\mathbf{i} \\
\mathbf{ki} &= \mathbf{j} \\
\mathbf{ik} &= -\mathbf{j}
\end{aligned}$$

quaternions and rotations

The quaternion $\mathbf{q} = (q_0, q_1, q_2, q_3)$ with

$$\|\mathbf{q}\|^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$$

represents a rotation through angle

$$\theta = 2\arccos(q_0)$$

about the axis

$$\mathbf{v} = \frac{(q_1, q_2, q_3)}{\sqrt{q_1^2 + q_2^2 + q_3^2}} \, . \qquad (*)$$

The other way round: a rotation through angle θ about a unit axis $=(v_1, v_2, v_3)$ is represented by the quaternion

$$\mathbf{q} = \cos(\theta/2) + \sin(\theta/2)(v_1\mathbf{i} + v_2\mathbf{j} + v_3\mathbf{k}) \, .$$

Question: What happens to $(*)$ when $q_1 = q_2 = q_3 = 0$?

More notes on quaternions

- A quaternion $\mathbf{q}$ and its negative, $-\mathbf{q}$ represent the same rotation.
- Given $\mathbf{q}_1$ and $\mathbf{q}_2$ and their corresponding rotations: $\mathtt{R}(\mathbf{q}_1)$ and $\mathtt{R}(\mathbf{q}_2)$, then the product quaternion $\mathbf{q}_1\mathbf{q}_2$ represents the rotation

$$\mathtt{R}(\mathbf{q}_1\mathbf{q}_2) = \mathtt{R}(\mathbf{q}_1)\mathtt{R}(\mathbf{q}_2) \ .$$

- The *conjugate* of a quaternion $\mathbf{q} = (q_0, q_1, q_2, q_3)$ is the quaternion $\bar{\mathbf{q}} = (q_0, -q_1, -q_2, -q_3)$.
- For a unit quaternion $\mathbf{q}$, we have $\mathbf{q}\bar{\mathbf{q}} = 1$, so we can also write $\bar{\mathbf{q}} = \mathbf{q}^{-1}$. Exercise: For an arbitrary quaternion (not unit) show that $\mathbf{q}\bar{\mathbf{q}} = \|\mathbf{q}\|^2$.
- A *vector quaternion* is one for which $\mathbf{q}_0 = 0$.
- A quaternion $\mathbf{q}$ rotates the vector quaternion $\mathbf{p}$ to the vector (quaternion) $\mathbf{q}\mathbf{p}\bar{\mathbf{q}}$. Exercise: if $\mathbf{p}$ is a vector quaternion, then $\mathbf{q}\mathbf{p}\mathbf{q}^{-1}$ is also a vector quaternion.

Exercise: If $\mathbf{q}$ and $\mathbf{p}$ are two quaternions, and $\mathbf{p}\mathbf{q} = 1$, show $\mathbf{q}\mathbf{p} = 1$.

Quaternion and matrix representation

The rotation matrix $\mathtt{R}(\mathbf{q})$ corresponding to a unit quaternion $\mathbf{q} = (q_r, q_i, q_j, q_k)$ is given by

$$\mathtt{R} = \begin{bmatrix} 1 - 2s(q_j^2 + q_k^2) & 2s(q_i q_j - q_k q_r) & 2s(q_i q_k + q_j q_r) \\ 2s(q_i q_j + q_k q_r) & 1 - 2s(q_i^2 + q_k^2) & 2s(q_j q_k - q_i q_r) \\ 2s(q_i q_k - q_j q_r) & 2s(q_j q_k + q_i q_r) & 1 - 2s(q_i^2 + q_j^2) \end{bmatrix}$$

Source:
https://en.wikipedia.org/w/index.php?title=Quaternions_and_spatial_rotation

Reading assignment

- Read the Wikipedia page

`Source:`
`https://en.wikipedia.org/w/index.php?title=Quaternions_and_spatial_rotation`

- Read the paper:

Angle-axis representation of rotations

A rotation through the angle θ about a unit axis $\mathbf{v} \in \mathbb{R}^3$ is represented by the vector

$$\mathbf{r} = \theta \mathbf{v}$$

Conversely, a vector $\mathbf{r}$ represents the rotation through angle $\|\mathbf{r}\|$ about the axis $\mathbf{r}/\|\mathbf{r}\|$ (unit axis).

Question: What happens when $\mathbf{r} = \mathbf{0}$?

Advantage over quaternions:

- The angle-axis representation is a minimal representation (3 parameters) whereas the quaternion representation needs 4 parameters.
- Quaternion representation needs to be normalized to have norm 1.

Disadvantage of angle-axis rotation:

- The angle-axis representation works best for rotations close to the identity (small rotations).
- For other rotations, a rotation should be represented as a delta with respect to a reference rotation.

$$\mathtt{R} = \mathtt{R}_0 \cdot \delta\mathtt{R}(\mathbf{v})$$

Rodriguez formula

How to transform between angle-axis representation and rotation matrix.

Notation

- From now on, we assume that all points are given in homogenous coordinates
- Points (in homogeneous coordinates) are denoted simply by $\mathbf{x}$, and not by $\hat{\mathbf{x}}$.
- The symbol $\approx$ means *equal up to scale*, or *equivalent as homogeneous vectors*.

Homographies

- Affine (infinity-preserving) mappings are sometimes not enough to capture the transformations that apply to images.
  - Views taken from different viewpoints.
  - Correspondences between world-plane and image.
- We need a transformation:

$$\begin{aligned}\mathbf{x}' &\approx \mathtt{H}\mathbf{x} \\ &= \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}\end{aligned}$$

- In non-homogeneous coordinates:

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{13}}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{13}}$$

- They are transformations of P^2 (the projective plane).

The DLT algorithm – computing a homography

Problem Statement:

Given n correspondences $\mathbf{x}'_i \leftrightarrow \mathbf{x}_i$, points in two images.

Compute homography $\mathtt{H}$ such that $\mathbf{x}'_i \approx \mathtt{H}\mathbf{x}_i$.

Each correspondence generates two equations

$$x'_i = \frac{h_{11}x_i + h_{12}y_i + h_{13}}{h_{31}x_i + h_{32}y_i + h_{33}} \qquad y'_i = \frac{h_{21}x_i + h_{22}y_i + h_{23}}{h_{31}x_i + h_{32}y_i + h_{33}}$$

Multiplying out gives equations linear in the matrix elements of $\mathtt{H}$

$$\begin{aligned} x'_i(h_{31}x_i + h_{32}y_i + h_{33}) &= h_{11}x_i + h_{12}y_i + h_{13} \\ y'_i(h_{31}x_i + h_{32}y_i + h_{33}) &= h_{21}x_i + h_{22}y_i + h_{23} \end{aligned}$$

These equations can be rearranged as

$$\begin{pmatrix} x & y & 1 & 0 & 0 & 0 & -x'x & -x'y & -x' \\ 0 & 0 & 0 & x & y & 1 & -y'x & -y'y & -y' \end{pmatrix} \mathbf{h} = \mathbf{0}$$

with $\mathbf{h} = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33})^\top$ a 9-vector.

Computing homography continued ...

Solving for $\mathtt{H}$

(i) Concatenate the equations from ($n \geq 4$) correspondences to generate $2n$ simultaneous equations, which can be written: $\mathtt{A}\mathbf{h} = \mathbf{0}$, where $\mathtt{A}$ is a $2n \times 12$ matrix.

(ii) In general this will not have an exact solution, but a (linear) solution which minimizes $|\mathtt{A}\mathbf{h}|$, subject to $|\mathbf{h}| = 1$ is obtained from the eigenvector with least eigenvalue of $\mathtt{A}^\top\mathtt{A}$. Or equivalently from the vector corresponding to the smallest singular value of the SVD of $\mathtt{A}$.

(iii) This linear solution is then used as the starting point for a non-linear minimization of the difference between the measured and projected point:

$$\min_{\mathtt{H}} \sum_i \left((x_i', y_i') - \mathrm{dehom}(\mathtt{H}(x_i, y_i, 1))\right)^2$$

where

- hom is the homogenizing operation $(x, y) \mapsto (x, y, 1)$.
- dehom is the dehomogenizing operation $(x, y, w) \mapsto (x/w, y/w)$.

The DLT algorithm – camera resection

Problem Statement:

Given n correspondences $\mathbf{x}_i \leftrightarrow \mathbf{X}_i$, points in the image and the world.

Compute camera matrix $\mathtt{P}$ such that $\mathbf{x}'_i \approx \mathtt{P}\mathbf{X}_i$.

Each correspondence generates two equations

$$x_i = \frac{p_{11}\mathrm{X}_i + p_{12}\mathrm{Y}_i + p_{13}\mathrm{Z}_i + p_{14}}{p_{31}\mathrm{X}_i + p_{32}\mathrm{Y}_i + p_{33}\mathrm{Z}_i + p_{34}} \qquad y_i = \frac{p_{21}\mathrm{X}_i + p_{22}\mathrm{Y}_i + p_{23}\mathrm{Z}_i + p_{24}}{p_{31}\mathrm{X}_i + p_{32}\mathrm{Y}_i + p_{33}\mathrm{Z}_i + p_{34}}$$

Multiplying out gives equations linear in the matrix elements of $\mathtt{P}$

$$\begin{aligned} x_i(p_{31}\mathrm{X}_i + p_{32}\mathrm{Y}_i + p_{33}\mathrm{Z}_i + p_{34}) &= p_{11}\mathrm{X}_i + p_{12}\mathrm{Y}_i + p_{13}\mathrm{Z}_i + p_{14} \\ y_i(p_{31}\mathrm{X}_i + p_{32}\mathrm{Y}_i + p_{33}\mathrm{Z}_i + p_{34}) &= p_{21}\mathrm{X}_i + p_{22}\mathrm{Y}_i + p_{23}\mathrm{Z}_i + p_{24} \end{aligned}$$

These equations can be rearranged as

$$\begin{pmatrix} \mathrm{X} & \mathrm{Y} & \mathrm{Z} & 1 & 0 & 0 & 0 & 0 & -x\mathrm{X} & -x\mathrm{Y} & -x\mathrm{Z} & -x \\ 0 & 0 & 0 & 0 & \mathrm{X} & \mathrm{Y} & \mathrm{Z} & 1 & -y\mathrm{X} & -y\mathrm{Y} & -y\mathrm{Z} & -y \end{pmatrix} \mathbf{p} = \mathbf{0}$$

with $\mathbf{p} = (p_{11}, p_{12}, p_{13}, p_{14}, p_{21}, p_{22}, p_{23}, p_{24}, p_{31}, p_{32}, p_{33}, p_{34})^\top$ a 12-vector.

Camera resection continued

Solving for $\mathtt{P}$

(i) Concatenate the equations from ($n \geq 6$) correspondences to generate $2n$ simultaneous equations, which can be written: $\mathtt{A}\mathbf{p} = \mathbf{0}$, where $\mathtt{A}$ is a $2n \times 12$ matrix.

(ii) In general this will not have an exact solution, but a (linear) solution which minimizes $|\mathtt{A}\mathbf{p}|$, subject to $|\mathbf{p}| = 1$ is obtained from the eigenvector with least eigenvalue of $\mathtt{A}^\top\mathtt{A}$. Or equivalently from the vector corresponding to the smallest singular value of the SVD of $\mathtt{A}$.

(iii) This linear solution is then used as the starting point for a non-linear minimization of the difference between the measured and projected point:

$$\min_{\mathtt{P}} \sum_i ((x_i, y_i) - \mathrm{dehom}(\mathtt{P}(\mathtt{X}_i, \mathtt{Y}_i, \mathtt{Z}_i, 1)))^2$$

Camera resection continued: Decompose $\mathtt{P}$ into $\mathtt{K}, \mathtt{R}$ and $\mathtt{t}$

The first 3×3 submatrix, $\mathtt{M}$, of $\mathtt{P}$ is the product ($\mathtt{M} = \mathtt{KR}$) of an upper triangular and rotation matrix.

(i) Factor $\mathtt{M}$ into $\mathtt{KR}$ using the $\mathtt{QR}$ matrix decomposition. This determines $\mathtt{K}$ and $\mathtt{R}$.

(ii) Then

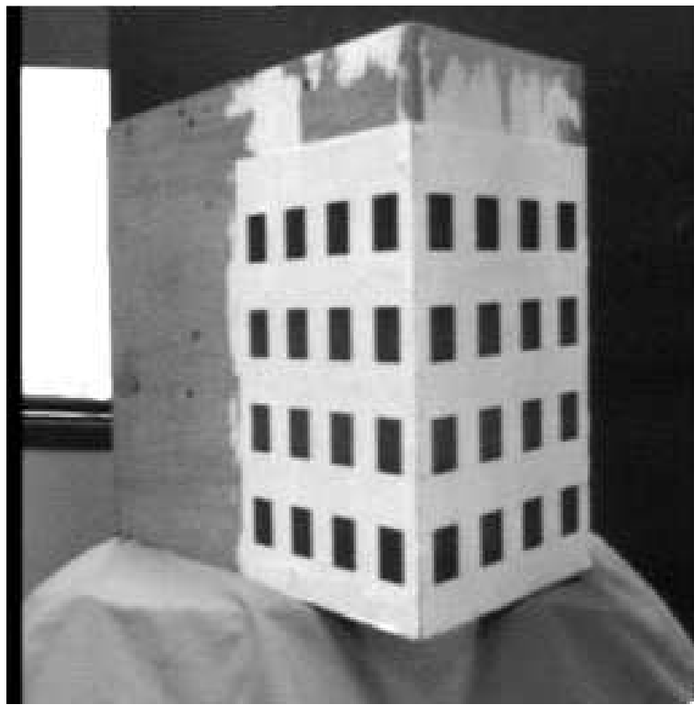
$$\mathbf{t} = \mathtt{K}^{-1}(p_{14}, p_{24}, p_{34})^\top$$

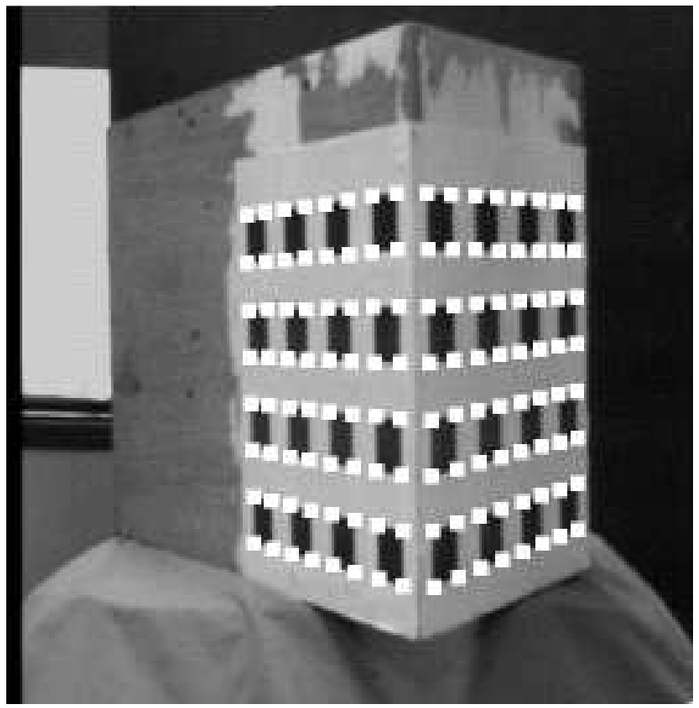
Note, this produces a matrix with an extra skew parameter s

$$\mathtt{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

with $s = \tan\theta$, and θ the angle between the image axes.

Example - Calibration Object





Determine accurate corner positions by

(i) Extract and link edges using Canny edge operator.
(ii) Fit lines to edges using orthogonal regression.
(iii) Intersect lines to obtain corners to sub-pixel accuracy.

The final error between measured and projected points is typically less than 0.02 pixels.

Algorithm step 2: Decompose $\mathtt{P}$ into $\mathtt{K}, \mathtt{R}$ and $\mathtt{t}$

The first 3×3 submatrix, $\mathtt{M}$, of $\mathtt{P}$ is the product ($\mathtt{M} = \mathtt{KR}$) of an upper triangular and rotation matrix.

(i) Factor $\mathtt{M}$ into $\mathtt{KR}$ using the $\mathtt{QR}$ matrix decomposition. This determines $\mathtt{K}$ and $\mathtt{R}$.

(ii) Then

$$\mathbf{t} = \mathtt{K}^{-1}(p_{14}, p_{24}, p_{34})^\top$$

Note, this produces a matrix with an extra skew parameter s

$$\mathtt{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

with $s = \tan\theta$, and θ the angle between the image axes.

What does calibration give us.

- Direction to points in the world.

Suppose:

- We have an RGB-D camera. Every pixel has a known “depth”.
- World and camera coordinate systems are aligned.
  - Camera is located at origin $(0, 0, 0)^\top$.
  - Viewing along Z axis, world X and Z axis aligned with image axes.
- Calibration matrix is

$$\mathtt{K} = \begin{bmatrix} f & & 0 \\ & f & 0 \\ & & 1 \end{bmatrix}$$

that is, principal point at $(0, 0)$ in image – pixel coordinates are measured with respect to the principal point.

Where is a point with image (x, y, d) located?

Answer: At location $(\mathrm{X}, \mathrm{Y}, \mathrm{Z}) = (dx/f, dy/f, d)$.

What happens if f is replaced with αf.

Then conclude point is located at point

$$\left(\frac{dx}{\alpha f}, \frac{dy}{\alpha f}, d\right) = \left(\frac{\mathrm{X}}{\alpha}, \frac{\mathrm{Y}}{\alpha}, \mathrm{Z}\right)$$

Point is shrunk by factor α in the X and Y directions, but not in the Z direction.

Factorization algorithm – Affine cameras

Consider an affine camera

$$\mathbf{X} \mapsto \mathtt{A}_{2\times 3}\mathbf{X} + \mathbf{t}$$

Can be written in terms of camera matrix $\mathtt{P}_{3\times 4}$ as

$$\begin{aligned}\mathbf{x} &\approx \mathtt{P}\mathbf{X} \\ &\approx \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{pmatrix} \mathtt{X} \\ \mathtt{Y} \\ \mathtt{Z} \\ 1 \end{pmatrix}\end{aligned}$$

In non-homogeneous coordinates

$$\mathbf{x} = \begin{bmatrix} \mathtt{A} & \mathbf{t} \end{bmatrix} \begin{pmatrix} \mathtt{X} \\ \mathtt{Y} \\ \mathtt{Z} \\ 1 \end{pmatrix}$$

Factorization algorithm – Affine cameras

Suppose that we have n affine images of m points:

$$\mathbf{x}_{ij} = \mathtt{A}_i \mathbf{X}_j + \mathbf{t}_i$$

where

$$\begin{aligned} \mathbf{x}_{ij} &\in \mathbb{R}^2 \\ \mathtt{A}_i &\in \mathbb{R}^{2\times 3} \\ \mathbf{X}_i &\in \mathbb{R}^3 \\ \mathbf{t}_i &\in \mathbb{R}^2 \end{aligned}$$

We know the points $\mathbf{x}_{ij}$, but not $\mathtt{P}_i = [\mathtt{A}_i \mid \mathbf{t}_i]$ or $\mathbf{X}_j$.

We are working in Euclidean (non-homogeneous) coordinates.

First step: centering

Let $\bar{\mathbf{X}}$ be the centroid of the (unknown) points $\mathbf{X}_j$. Thus

$$\bar{\mathbf{X}} = \frac{1}{m}\sum_{j=1}^{m}\mathbf{X}_j \ .$$

Then,

$$\begin{aligned}\mathtt{A}_i\bar{\mathbf{X}} &= \frac{1}{m}\sum_{j=1}^{m}\mathtt{A}_i\mathbf{X}_j = \frac{1}{m}\sum_{j=1}^{m}\left(\mathbf{x}_{ij} - \mathbf{t}_i\right)\\ &= \bar{\mathbf{x}}_i - \mathbf{t}_i \ .\end{aligned}$$

Thus, for each i, camera $\mathtt{A}_i$ takes the centroid of the points $\mathbf{X}_j$ to the centroid of the points $\mathbf{x}_{ij}$. Then

$$\begin{aligned}\mathbf{x}_{ij} &= \mathtt{A}_i\mathbf{X}_j + \mathbf{t}_i\\ \mathbf{x}_{ij} - \bar{\mathbf{x}}_i &= \mathtt{A}_i\mathbf{X}_j + \mathbf{t}_i - \mathtt{A}_i\bar{\mathbf{X}} - \mathbf{t}_i\\ &= \mathtt{A}_i(\mathbf{X}_j - \bar{\mathbf{X}})\\ \mathbf{x}'_{ij} &= \mathtt{A}_i\mathbf{X}'_j\end{aligned}$$

where $\mathbf{x}'_{ij}$ and $\mathbf{X}_j$ are the "centered points" with respect to centroid.

Matrix equation

Write as one matrix equation:

$$\begin{bmatrix} \mathbf{x}'_{11} & \mathbf{x}'_{12} & \dots & \mathbf{x}'_{1m} \\ \mathbf{x}'_{21} & \mathbf{x}'_{12} & \dots & \mathbf{x}'_{2m} \\ \vdots & & & \vdots \\ \mathbf{x}'_{n1} & \mathbf{x}'_{n2} & \dots & \mathbf{x}'_{nm} \end{bmatrix} = \begin{bmatrix} \mathtt{A}_1 \\ \mathtt{A}_2 \\ \vdots \\ \mathtt{A}_n \end{bmatrix} \quad \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \dots, \mathbf{X}_m \end{bmatrix}$$

Factor LHS as shown to compute all the matrices $\mathtt{A}_i$ and points $\mathbf{X}_j$ at once.

Low-rank matrix factorization.

Low rank matrix factorization

Given

$$\mathtt{W} = \begin{bmatrix} \mathbf{x}'_{11} & \mathbf{x}'_{12} & \dots & \mathbf{x}'_{1m} \\ \mathbf{x}'_{21} & \mathbf{x}'_{12} & \dots & \mathbf{x}'_{2m} \\ \vdots & & & \vdots \\ \mathbf{x}'_{n1} & \mathbf{x}'_{n2} & \dots & \mathbf{x}'_{nm} \end{bmatrix} ,$$

Compute SVD:

$$\mathtt{W} = \mathtt{U}\mathtt{D}\mathtt{V}^\top$$

Since $\mathtt{W}$ has rank 3, we can write

$$\mathtt{W} = \mathtt{U}_{1\dots3} \left(\mathbf{diag}(d_{11}, d_{22}, d_{33}) \mathtt{V}_{1\dots3}{}^\top \right)$$

where $\mathtt{U}_{1\dots3}$ means the first 3 columns of $\mathtt{U}$, etc.

This gives the desired factorization

$$\mathtt{W} = \begin{bmatrix} \mathtt{A}_1 \\ \mathtt{A}_2 \\ \vdots \\ \mathtt{A}_n \end{bmatrix} \quad \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \dots, \mathbf{X}_m \end{bmatrix} .$$

Ambiguity

If

$$\mathtt{W} = \begin{bmatrix} \mathtt{A}_1 \\ \mathtt{A}_2 \\ \vdots \\ \mathtt{A}_n \end{bmatrix} \quad \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \ldots, \mathbf{X}_m \end{bmatrix}$$

is one solution, then

$$\mathtt{W} = \begin{bmatrix} \mathtt{A}_1 \\ \mathtt{A}_2 \\ \vdots \\ \mathtt{A}_n \end{bmatrix} \quad \mathtt{T}^{-1}\mathtt{T} \quad \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \ldots, \mathbf{X}_m \end{bmatrix}$$

is also a solution for any $\mathtt{T}$ (a 3×3 matrix).

- Each projection $\mathtt{A}_i$ replaced by $\mathtt{A}_i\mathtt{T}^{-1}$
- Each point $\mathbf{X}_j$ replaced by $\mathtt{T}\mathbf{X}_j$.

Solution is only unique up to an affine (or linear) transformation $\mathtt{T}$.

Projective factorization

Suppose $\mathbf{x}_{ij} \approx \mathtt{P}_i \mathbf{X}_j$ (homogeneous coordinates). This can be written

$$\begin{bmatrix} \lambda_{11}\mathbf{x}_{11} & \lambda_{12}\mathbf{x}_{12} & \dots & \lambda_{1m}\mathbf{x}_{1m} \\ \lambda_{21}\mathbf{x}_{21} & \lambda_{22}\mathbf{x}_{12} & \dots & \lambda_{2m}\mathbf{x}_{2m} \\ \vdots & & & \vdots \\ \lambda_{n1}\mathbf{x}_{n1} & \lambda_{n2}\mathbf{x}_{n2} & \dots & \lambda_{nm}\mathbf{x}_{nm} \end{bmatrix} = \begin{bmatrix} \mathtt{P}_1 \\ \mathtt{P}_2 \\ \vdots \\ \mathtt{P}_n \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \dots, \mathbf{X}_m \end{bmatrix}$$

The scale-factors λ_{ij}, as well as the $\mathbf{X}_1$ and $\mathtt{P}_j$ are unknown.

Alternating strategy:

(i) Initialize all λ_{ij} (perhaps to 1) and compute matrix $W = [\lambda_{ij}\mathbf{x}_{ij}]$

(ii) Factorize (rank 4) to compute $\mathtt{P}_i$ and $\mathbf{X}_j$.

(iii) Reproject $\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j$ and compute new value of λ_{ij}.

(iv) Recompute matrix $W = [\lambda_{ij}\mathbf{x}_{ij}]$

(v) Factorize to compute new estimate $\mathtt{P}_i$ and $\mathbf{X}_j$.

(vi) ...

Projective ambiguity

If

$$\mathtt{W} = \begin{bmatrix} \mathtt{P}_1 \\ \mathtt{P}_2 \\ \vdots \\ \mathtt{P}_n \end{bmatrix} \quad \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \ldots, \mathbf{X}_m \end{bmatrix}$$

is one solution, then

$$\mathtt{W} = \begin{bmatrix} \mathtt{P}_1 \\ \mathtt{P}_2 \\ \vdots \\ \mathtt{P}_n \end{bmatrix} \quad \mathtt{H}^{-1}\mathtt{H} \quad \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \ldots, \mathbf{X}_m \end{bmatrix}$$

is also a solution for any $\mathtt{H}$ (a 4×4 matrix).

- Each projection $\mathtt{P}_i$ replaced by $\mathtt{P}_i\mathtt{H}^{-1}$
- Each point $\mathbf{X}_j$ replaced by $\mathtt{H}\mathbf{X}_j$.

Solution is only unique up to a projective transformation.

Part I: Single and Two View Geometry

The main points covered in this part are:

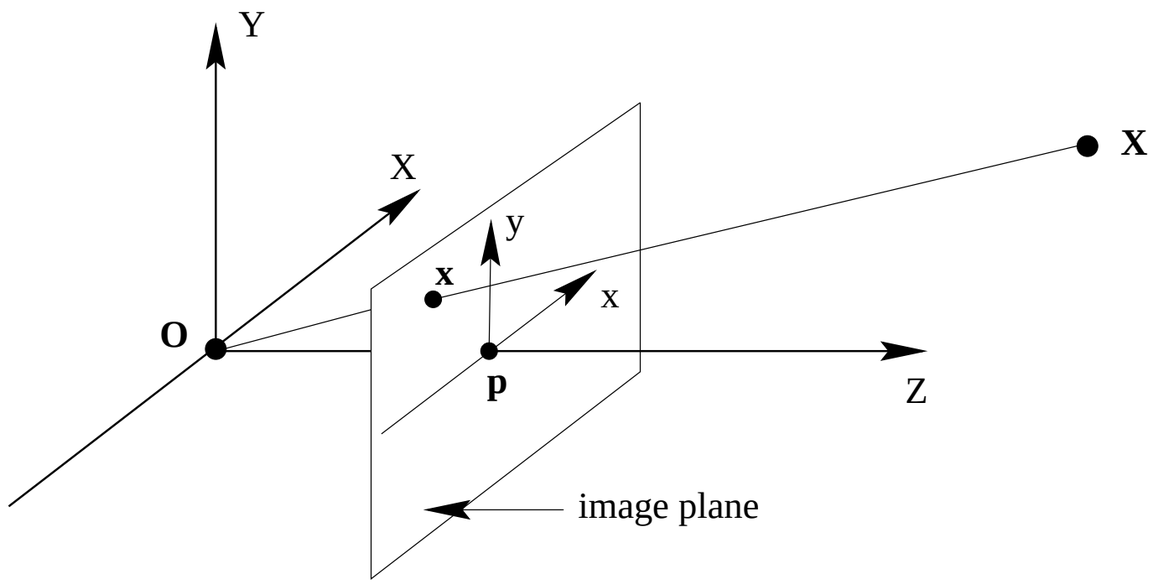
- A perspective (central) projection camera is represented by a 3×4 matrix.
- The most general perspective transformation transformation between two planes (a world plane and the image plane, or two image planes induced by a world plane) is a plane projective transformation. This can be computed from the correspondence of four (or more) points.
- The epipolar geometry between two views is represented by the fundamental matrix. This can be computed from the correspondence of seven (or more) points.

Imaging Geometry

Perspective projection

$$\lambda \begin{pmatrix} x \\ y \\ f \end{pmatrix} = \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \end{pmatrix}$$

where $\lambda = \mathrm{Z}/f$.



This can be written as a linear mapping between homogeneous coordinates (the equation is only up to a scale factor):

$$\begin{pmatrix} x \\ y \\ f \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix}$$

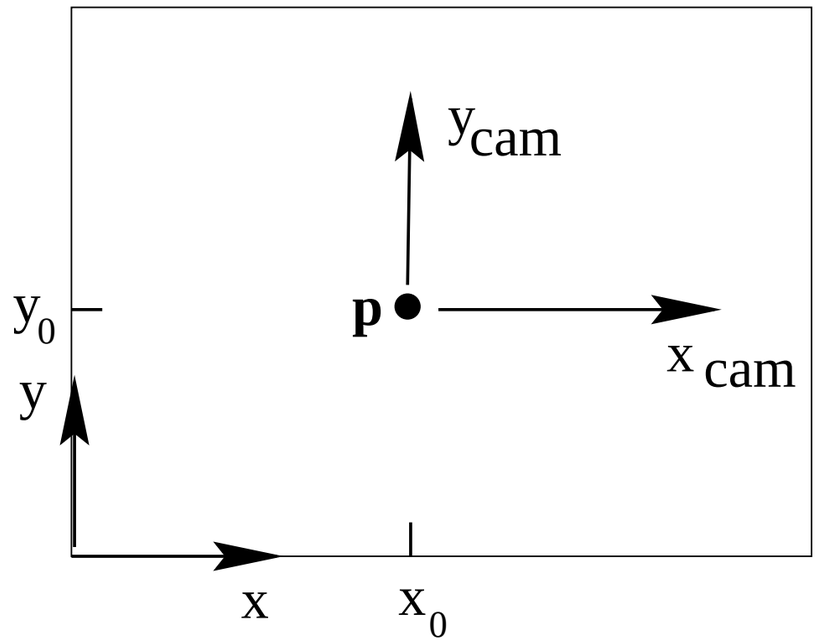
where a 3×4 projection matrix represents a map from 3D to 2D.

Image Coordinate System

Internal camera parameters

$$k_x\, x_{\text{cam}} = x - x_0$$
$$k_y\, y_{\text{cam}} = y - y_0$$

where the units of k_x, k_y are $[\text{pixels}/mm]$.



$$\mathbf{x} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \frac{1}{f} \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \begin{pmatrix} x_{\text{cam}} \\ y_{\text{cam}} \\ f \end{pmatrix} = \mathtt{K} \begin{pmatrix} x_{\text{cam}} \\ y_{\text{cam}} \\ f \end{pmatrix}$$

where $\alpha_x = f k_x,\ \alpha_y = f k_y$.

Image Coordinate System

Internal camera parameters

This gives

$$\mathbf{x} \approx \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \begin{pmatrix} x_{\text{cam}} \\ y_{\text{cam}} \\ f \end{pmatrix} = \mathtt{K} \begin{pmatrix} x_{\text{cam}} \\ y_{\text{cam}} \\ f \end{pmatrix}$$

where $\alpha_x = fk_x,\ \alpha_y = fk_y$.

Units of α_x and α_y.

$$\begin{aligned} \alpha_x &= fk_x \\ &= \text{(focal length in mm)} \times \text{(pixels / mm)} \\ &= \text{focal length in pixels} \end{aligned}$$

Camera Calibration Matrix

$\mathtt{K}$ is a 3×3 upper triangular matrix, called the camera calibration matrix:

$$\mathtt{K} = \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

- There are four parameters:

  (i) The scaling in the image x and y directions, α_x and α_y equal to focal-length in pixel units.

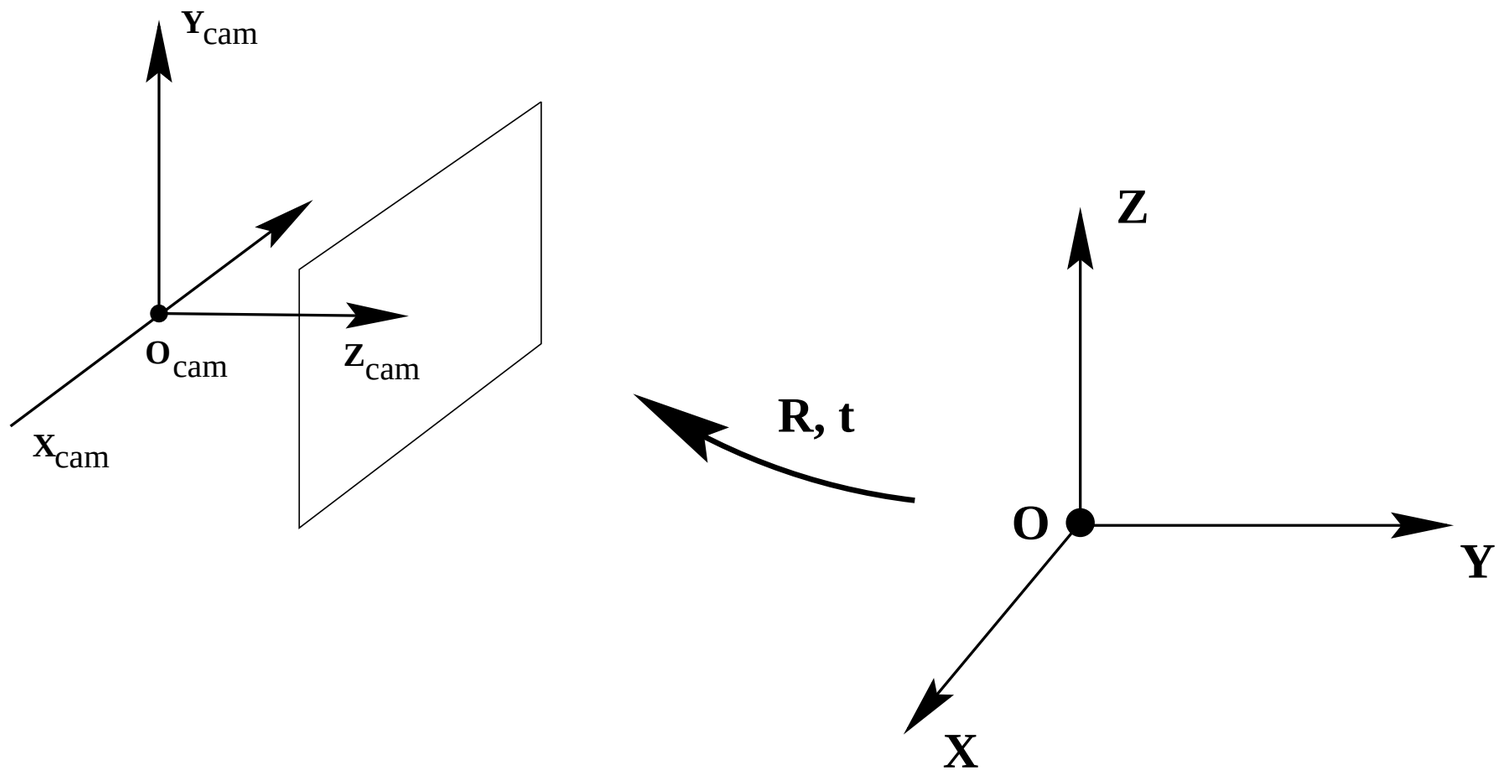
  (ii) The principal point (x_0, y_0), which is the point where the optic axis intersects the image plane.

- The aspect ratio is α_y/α_x.

World Coordinate System

External camera parameters

$$\begin{pmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{pmatrix} = \begin{bmatrix} R & t \\ 0^\top & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$



Euclidean transformation between world and camera coordinates

- R is a 3×3 rotation matrix
- t is a 3×1 translation vector

Concatenating the three matrices,

$$\mathbf{x} \approx \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathtt{K} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{pmatrix} \mathtt{X} \\ \mathtt{Y} \\ \mathtt{Z} \\ 1 \end{pmatrix} \approx \mathtt{K} \begin{bmatrix} \mathtt{R} | & \mathbf{t} \end{bmatrix} \mathbf{X}$$

which defines the 3×4 projection matrix from Euclidean 3-space to an image as

$$\begin{aligned} \mathbf{x} &\approx \mathtt{P}\mathbf{X} \\ &\approx \mathtt{P} = \mathtt{K} \begin{bmatrix} \mathtt{R} | & \mathbf{t} \end{bmatrix} \end{aligned}$$

Note, the camera centre is at $(\mathtt{X}, \mathtt{Y}, \mathtt{Z})^\top = -\mathtt{R}^\top \mathbf{t}$.

In the following it is often only the 3×4 form of $\mathtt{P}$ that is important, rather than its decomposition.

A Projective Camera

The camera model for perspective projection is a linear map between homogeneous point coordinates

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{bmatrix} & \mathtt{P} \quad (3 \times 4) & \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix}$$

Image Point $\lambda \mathbf{x}$ $=$ $\mathtt{P}$ Scene Point $\mathbf{X}$

- The camera centre is the null-vector of $\mathtt{P}$
  e.g. if $\mathtt{P} = [\mathtt{I}|0]$ then the centre is $\mathbf{X} = (0, 0, 0, 1)^\top$.
- $\mathtt{P}$ has 11 degrees of freedom (essential parameters).
- $\mathtt{P}$ has rank 3.

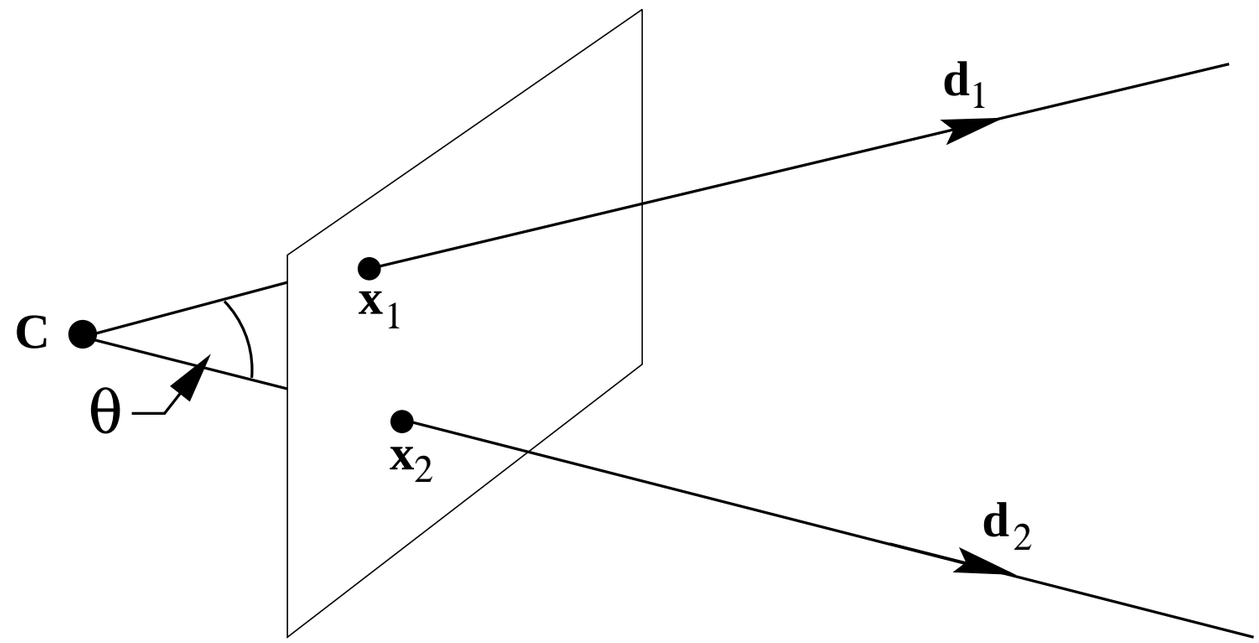
What does calibration give?

- $\mathtt{K}$ provides the transformation between an image point and a ray in Euclidean 3-space.
- Once $\mathtt{K}$ is known the camera is termed calibrated.
- A calibrated camera is a direction sensor, able to measure the direction of rays — like a 2D protractor.

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \begin{pmatrix} \mathrm{X}_{\mathrm{cam}} \\ \mathrm{Y}_{\mathrm{cam}} \\ \mathrm{Z}_{\mathrm{cam}} \end{pmatrix} = \mathtt{K}\mathbf{d}$$

Angle between rays

$$\cos\theta = \frac{\mathbf{d}_1.\mathbf{d}_2}{(\mathbf{d}_1.\mathbf{d}_1)^{\frac{1}{2}}(\mathbf{d}_2.\mathbf{d}_2)^{\frac{1}{2}}}$$



$$\begin{aligned}\cos\theta &= \frac{\mathbf{d}_1{}^\top\mathbf{d}_2}{(\mathbf{d}_1{}^\top\mathbf{d}_1)^{1/2}(\mathbf{d}_2{}^\top\mathbf{d}_2)^{1/2}} = \frac{\mathbf{x}_1{}^\top(\mathtt{K}^{-\top}\mathtt{K}^{-1})\mathbf{x}_2}{(\mathbf{x}_1{}^\top(\mathtt{K}^{-\top}\mathtt{K}^{-1})\mathbf{x}_1)^{1/2}(\mathbf{x}_2{}^\top(\mathtt{K}^{-\top}\mathtt{K}^{-1})\mathbf{x}_2)^{1/2}} \\ &= \frac{\mathbf{x}_1{}^\top\omega\mathbf{x}_2}{(\mathbf{x}_1{}^\top\omega\mathbf{x}_1)^{1/2}(\mathbf{x}_2{}^\top\omega\mathbf{x}_2)^{1/2}}\end{aligned}$$

where $\omega = (\mathtt{K}\mathtt{K}^\top)^{-1}$.

Camera Calibration (Resectioning)

Problem Statement:

Given n correspondences $\mathbf{x}_i \leftrightarrow \mathbf{X}_i$, where $\mathbf{X}_i$ is a scene point and $\mathbf{x}_i$ its image:

Compute

$$\mathtt{P} = \mathtt{K}\left[\mathtt{R}\middle|\ \mathbf{t}\right] \text{ such that } \mathbf{x}_i = \mathtt{P}\mathbf{X}_i.$$

The algorithm for camera calibration has two parts:

(i) Compute the matrix $\mathtt{P}$ from a set of point correspondences.

(ii) Decompose $\mathtt{P}$ into $\mathtt{K}, \mathtt{R}$ and $\mathbf{t}$ via the $\mathtt{QR}$ decomposition.

Algorithm step 1: Compute the matrix P

$$\mathbf{x}_i = \mathtt{P}\mathbf{X}_i$$

Each correspondence generates two equations

$$x_i = \frac{p_{11}\mathrm{X}_i + p_{12}\mathrm{Y}_i + p_{13}\mathrm{Z}_i + p_{14}}{p_{31}\mathrm{X}_i + p_{32}\mathrm{Y}_i + p_{33}\mathrm{Z}_i + p_{34}} \qquad y_i = \frac{p_{21}\mathrm{X}_i + p_{22}\mathrm{Y}_i + p_{23}\mathrm{Z}_i + p_{24}}{p_{31}\mathrm{X}_i + p_{32}\mathrm{Y}_i + p_{33}\mathrm{Z}_i + p_{34}}$$

Multiplying out gives equations linear in the matrix elements of P

$$\begin{aligned} x_i(p_{31}\mathrm{X}_i + p_{32}\mathrm{Y}_i + p_{33}\mathrm{Z}_i + p_{34}) &= p_{11}\mathrm{X}_i + p_{12}\mathrm{Y}_i + p_{13}\mathrm{Z}_i + p_{14} \\ y_i(p_{31}\mathrm{X}_i + p_{32}\mathrm{Y}_i + p_{33}\mathrm{Z}_i + p_{34}) &= p_{21}\mathrm{X}_i + p_{22}\mathrm{Y}_i + p_{23}\mathrm{Z}_i + p_{24} \end{aligned}$$

These equations can be rearranged as

$$\begin{pmatrix} \mathrm{X} & \mathrm{Y} & \mathrm{Z} & 1 & 0 & 0 & 0 & 0 & -x\mathrm{X} & -x\mathrm{Y} & -x\mathrm{Z} & -x \\ 0 & 0 & 0 & 0 & \mathrm{X} & \mathrm{Y} & \mathrm{Z} & 1 & -y\mathrm{X} & -y\mathrm{Y} & -y\mathrm{Z} & -y \end{pmatrix} \mathbf{p} = \mathbf{0}$$

with $\mathbf{p} = (p_{11}, p_{12}, p_{13}, p_{14}, p_{21}, p_{22}, p_{23}, p_{24}, p_{31}, p_{32}, p_{33}, p_{34})^\top$ a 12-vector.

Algorithm step 1 continued

Solving for $\mathtt{P}$

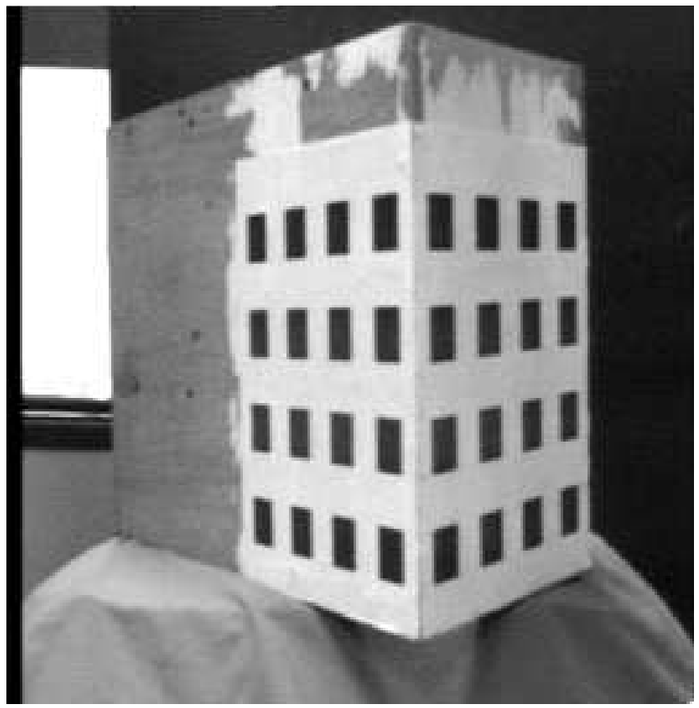
(i) Concatenate the equations from ($n \geq 6$) correspondences to generate $2n$ simultaneous equations, which can be written: $\mathtt{A}\mathbf{p} = \mathbf{0}$, where $\mathtt{A}$ is a $2n \times 12$ matrix.

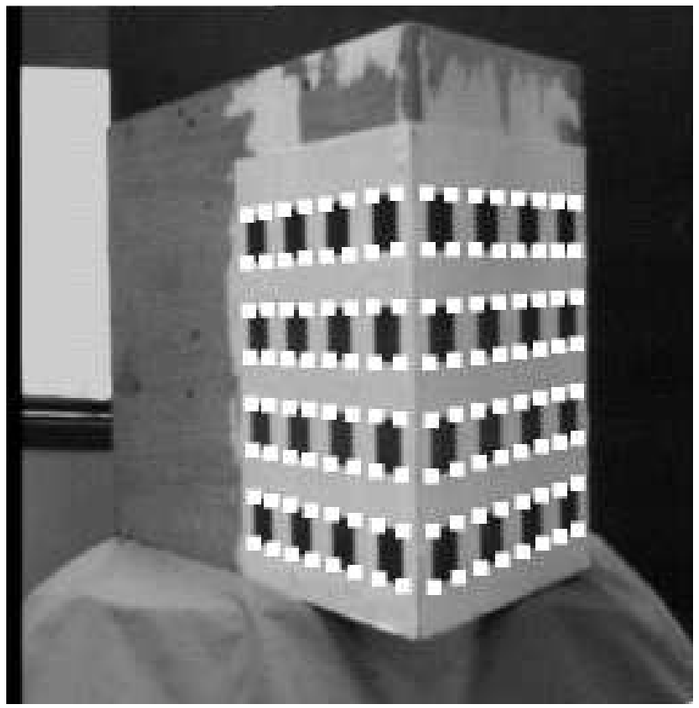
(ii) In general this will not have an exact solution, but a (linear) solution which minimises $|\mathtt{A}\mathbf{p}|$, subject to $|\mathbf{p}| = 1$ is obtained from the eigenvector with least eigenvalue of $\mathtt{A}^\top\mathtt{A}$. Or equivalently from the vector corresponding to the smallest singular value of the SVD of $\mathtt{A}$.

(iii) This linear solution is then used as the starting point for a non-linear minimisation of the difference between the measured and projected point:

$$\min_{\mathtt{P}} \sum_i \left((x_i, y_i) - P(\mathtt{X}_i, \mathtt{Y}_i, \mathtt{Z}_i, 1)\right)^2$$

Example - Calibration Object





Determine accurate corner positions by

(i) Extract and link edges using Canny edge operator.
(ii) Fit lines to edges using orthogonal regression.
(iii) Intersect lines to obtain corners to sub-pixel accuracy.

The final error between measured and projected points is typically less than 0.02 pixels.

Algorithm step 2: Decompose $\mathtt{P}$ into $\mathtt{K}, \mathtt{R}$ and $\mathtt{t}$

The first 3×3 submatrix, $\mathtt{M}$, of $\mathtt{P}$ is the product ($\mathtt{M} = \mathtt{KR}$) of an upper triangular and rotation matrix.

(i) Factor $\mathtt{M}$ into $\mathtt{KR}$ using the $\mathtt{QR}$ matrix decomposition. This determines $\mathtt{K}$ and $\mathtt{R}$.

(ii) Then

$$\mathbf{t} = \mathtt{K}^{-1}(p_{14}, p_{24}, p_{34})^\top$$

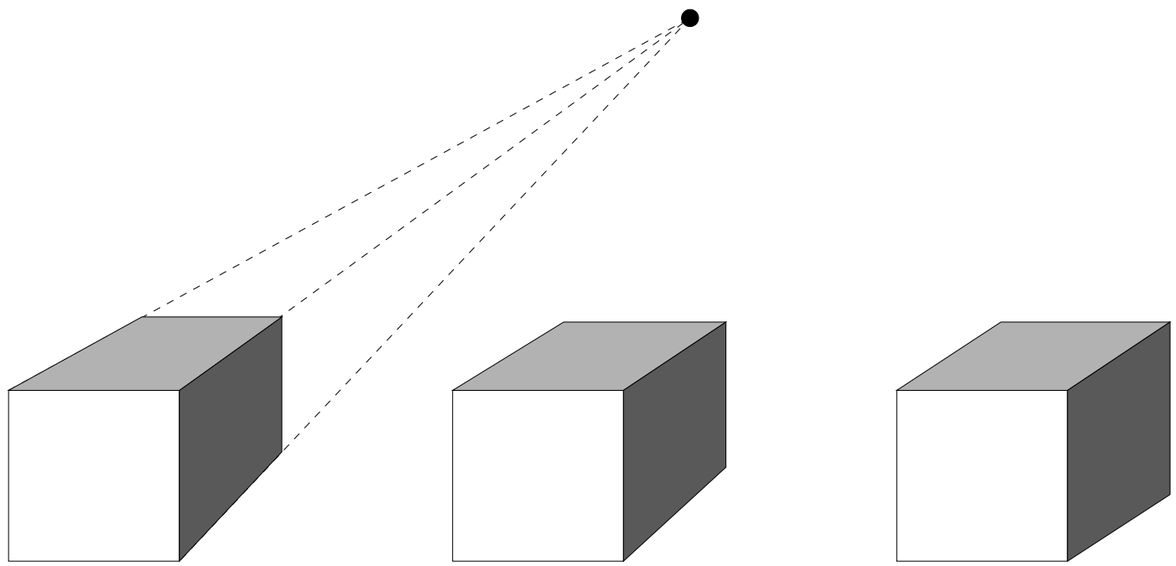
Note, this produces a matrix with an extra skew parameter s

$$\mathtt{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

with $s = \tan\theta$, and θ the angle between the image axes.

Weak Perspective

Track back, whilst zooming to keep image size fixed



perspective **weak perspective**

The imaging rays become parallel, and the result is:

$$\mathtt{P} = \mathtt{K} \begin{bmatrix} r_{11} & r_{12} & r_{13} & * \\ r_{21} & r_{22} & r_{23} & * \\ 0 & 0 & 0 & * \end{bmatrix}$$

A generalization is the *affine camera*

The Affine Camera

$$\mathtt{P} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & t_1 \\ m_{21} & m_{22} & m_{23} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

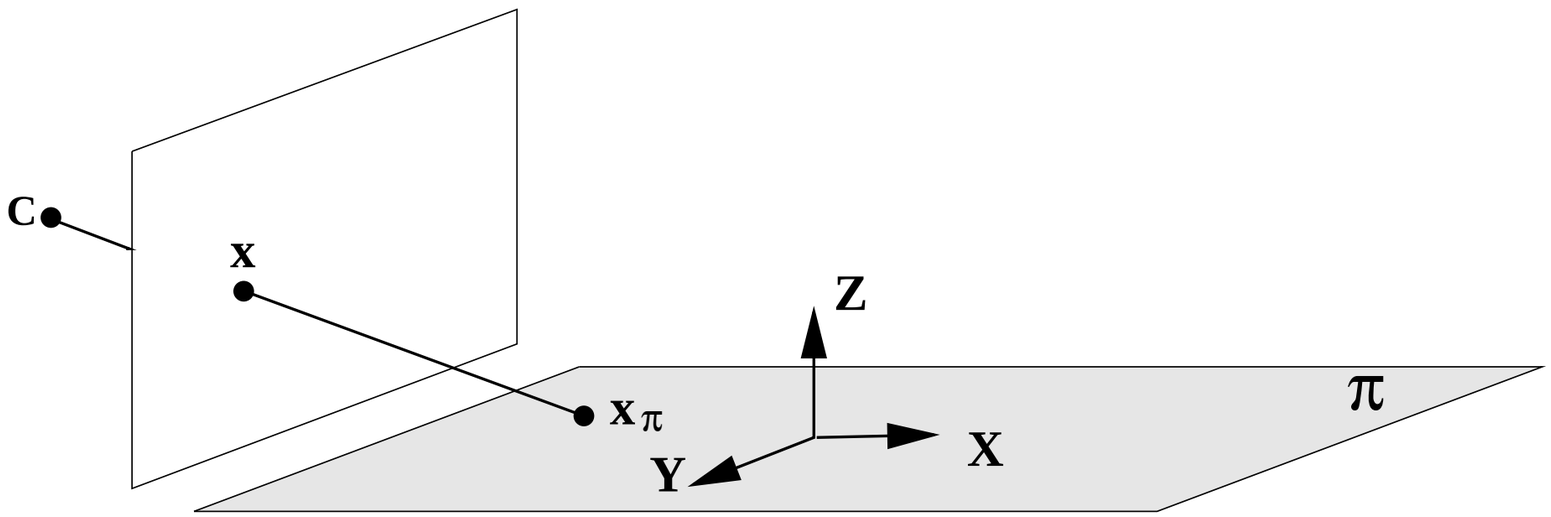
The matrix $\mathtt{M}_{2\times 3}$ has rank two.

Projection under an affine camera is a linear mapping on non-homogeneous coordinates composed with a translation:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{pmatrix} \mathtt{X} \\ \mathtt{Y} \\ \mathtt{Z} \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$$

- The point $(t_1, t_2)^\top$ is the image of the world origin.
- The centre of the affine camera is at infinity.
- An affine camera has 8 degrees of freedom.
- It models weak-perspective and para-perspective.

Plane projective transformations



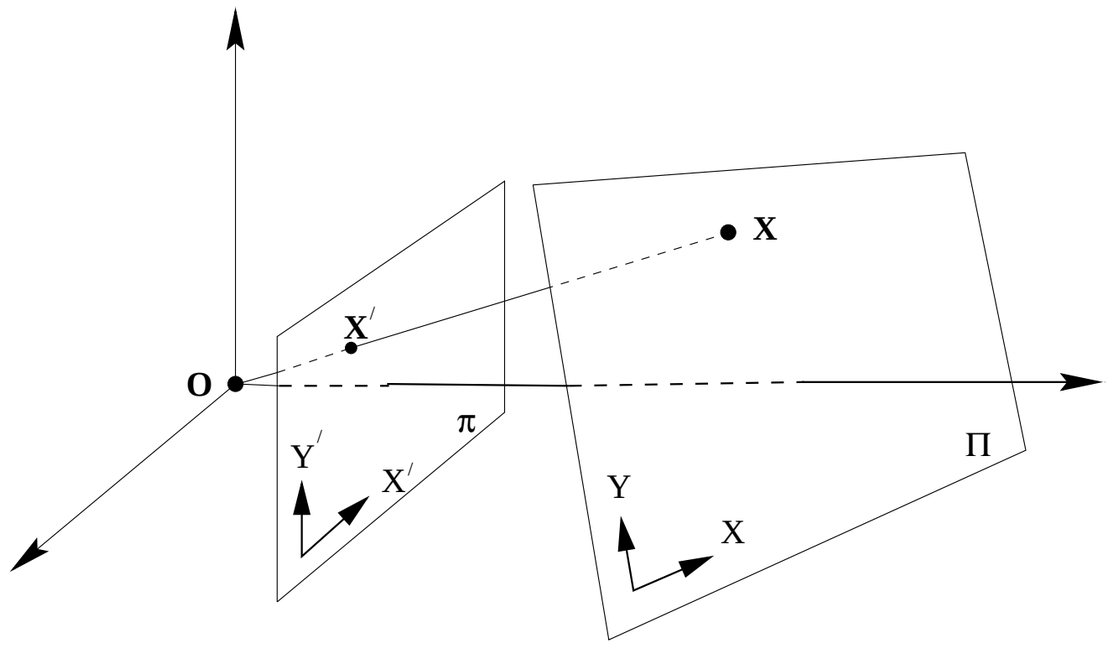
Choose the world coordinate system such that the plane of the points has zero Z coordinate. Then the 3×4 matrix $\mathtt{P}$ reduces to

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ 1 \end{pmatrix}$$

which is a 3×3 matrix representing a general plane to plane projective transformation.

Projective transformations continued

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$



or $\mathbf{x}' = \mathtt{H}\mathbf{x}$, where $\mathtt{H}$ is a 3×3 non-singular homogeneous matrix.

- This is the most general transformation between the world and image plane under imaging by a perspective camera.
- It is often only the 3×3 form of the matrix that is important in establishing properties of this transformation.
- A projective transformation is also called a “homography” and a “collineation”.
- $\mathtt{H}$ has 8 degrees of freedom.

Four points define a projective transformation

Given n point correspondences $(x, y) \leftrightarrow (x', y')$

Compute $\mathtt{H}$ such that $\mathbf{x}'_i = \mathtt{H}\mathbf{x}_i$

- Each point correspondence gives two constraints

$$x' = \frac{x'_1}{x'_3} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \qquad y' = \frac{x'_2}{x'_3} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

and multiplying out generates two linear equations for the elements of $\mathtt{H}$

$$\begin{aligned} x'\,(h_{31}x + h_{32}y + h_{33}) &= h_{11}x + h_{12}y + h_{13} \\ y'\,(h_{31}x + h_{32}y + h_{33}) &= h_{21}x + h_{22}y + h_{23} \end{aligned}$$

- If $n \geq 4$ (no three points collinear), then $\mathtt{H}$ is determined uniquely.
- The converse of this is that it is possible to transform any four points in general position to any other four points in general position by a projectivity.

Example 1: Removing Perspective Distortion

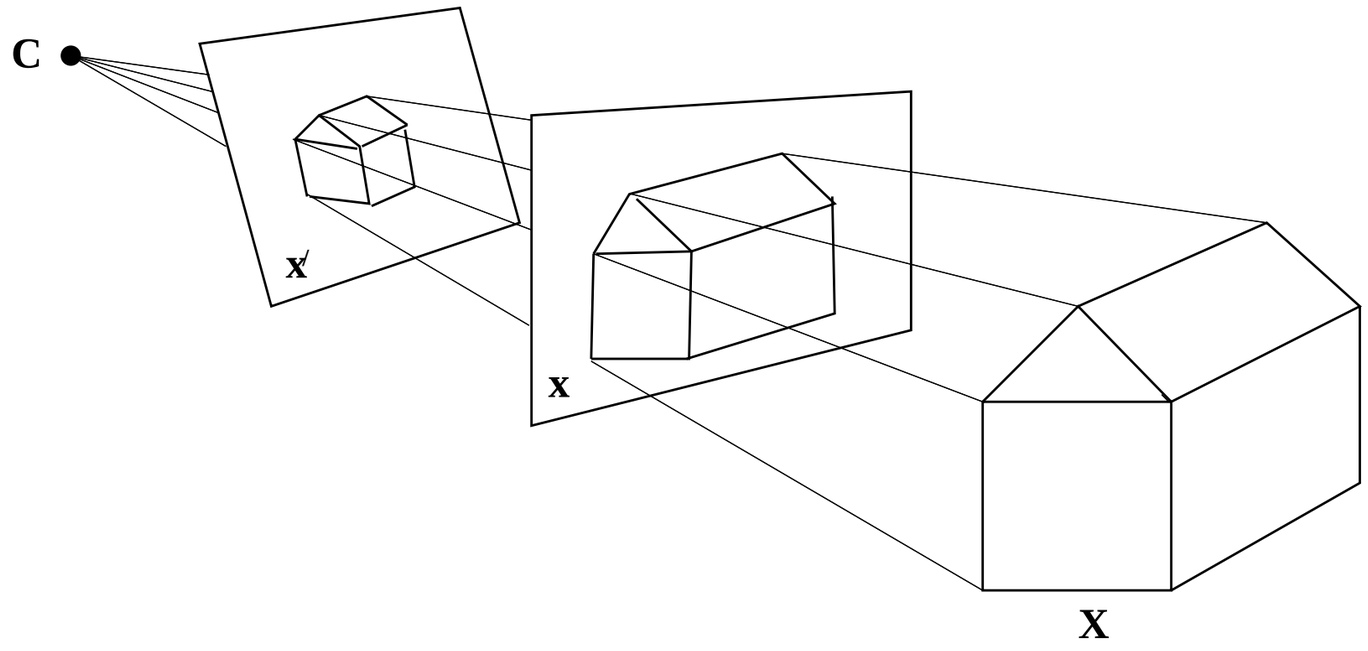
Given: the coordinates of four points on the scene plane

Find: a projective rectification of the plane

- This rectification does not require knowledge of any of the camera's parameters or the pose of the plane.
- It is not always necessary to know coordinates for four points.

The cone of rays

An image is the intersection of a plane with the cone of rays between points in 3-space and the optical centre. Any two such “images” (with the same camera centre) are related by a planar projective transformation.



$$\mathbf{x}' = \mathtt{H}\mathbf{x}$$

e.g. rotation about the camera centre

Example 2: Synthetic Rotations

The synthetic images are produced by projectively warping the original image so that four corners of an imaged rectangle map to the corners of a rectangle. Both warpings correspond to a synthetic rotation of the camera about the (fixed) camera centre.

Two View Geometry

- Cameras $\mathtt{P}$ and $\mathtt{P}'$ such that

$$\mathbf{x} = \mathtt{P}\mathbf{X} \qquad \mathbf{x}' = \mathtt{P}'\mathbf{X}$$

- Baseline between the cameras is non-zero.

Given an image point in the first view, where is the corresponding point in the second view?

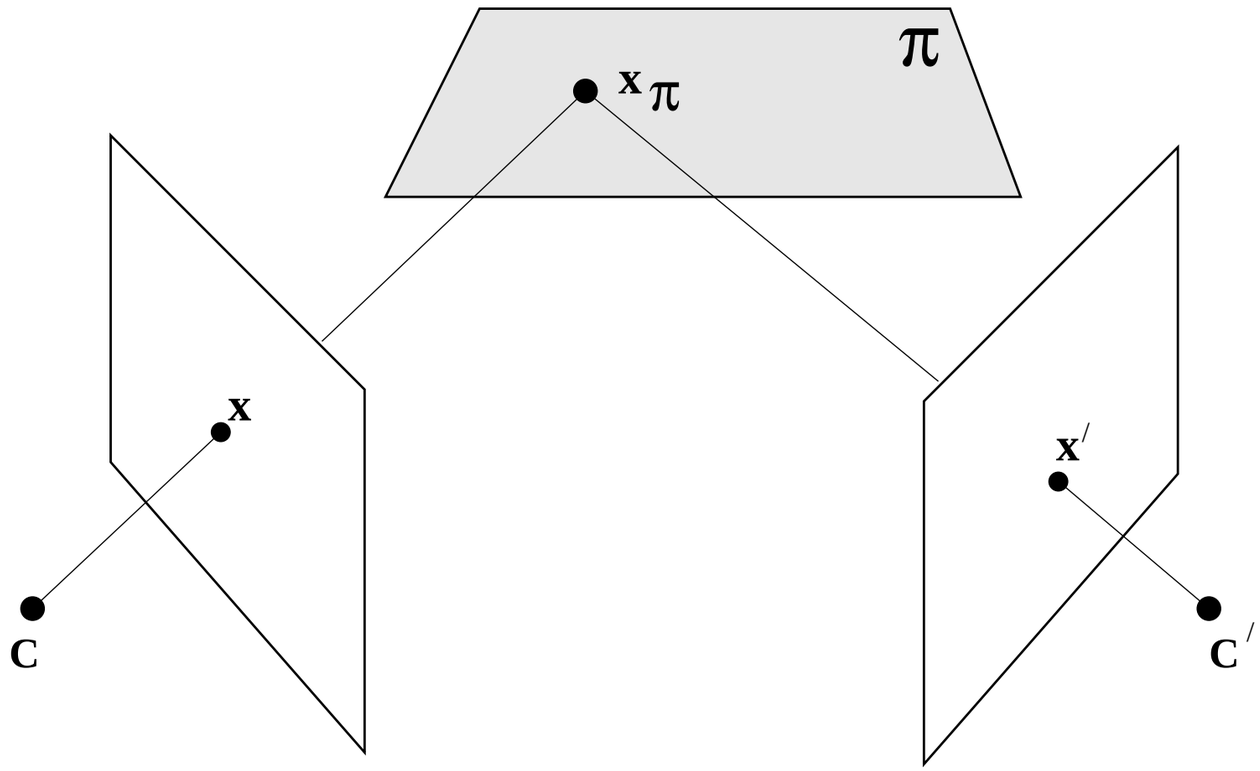
What is the relative position of the cameras?

What is the 3D geometry of the scene?

Images of Planes

Projective transformation between images induced by a plane

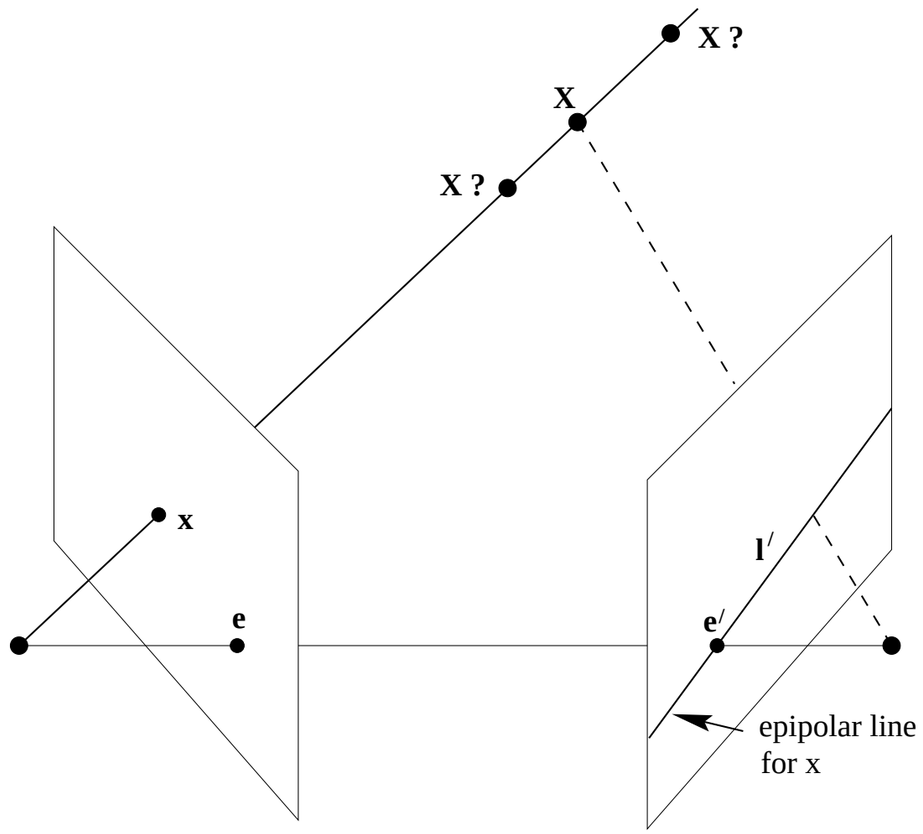
$$\begin{aligned}\mathbf{x} &= \mathtt{H}_{1\pi}\mathbf{x}_\pi \\ \mathbf{x}' &= \mathtt{H}_{2\pi}\mathbf{x}_\pi \\ &= \mathtt{H}_{2\pi}\mathtt{H}_{1\pi}^{-1}\mathbf{x} = \mathtt{H}\mathbf{x}\end{aligned}$$



- H can be computed from the correspondence of four points on the plane.

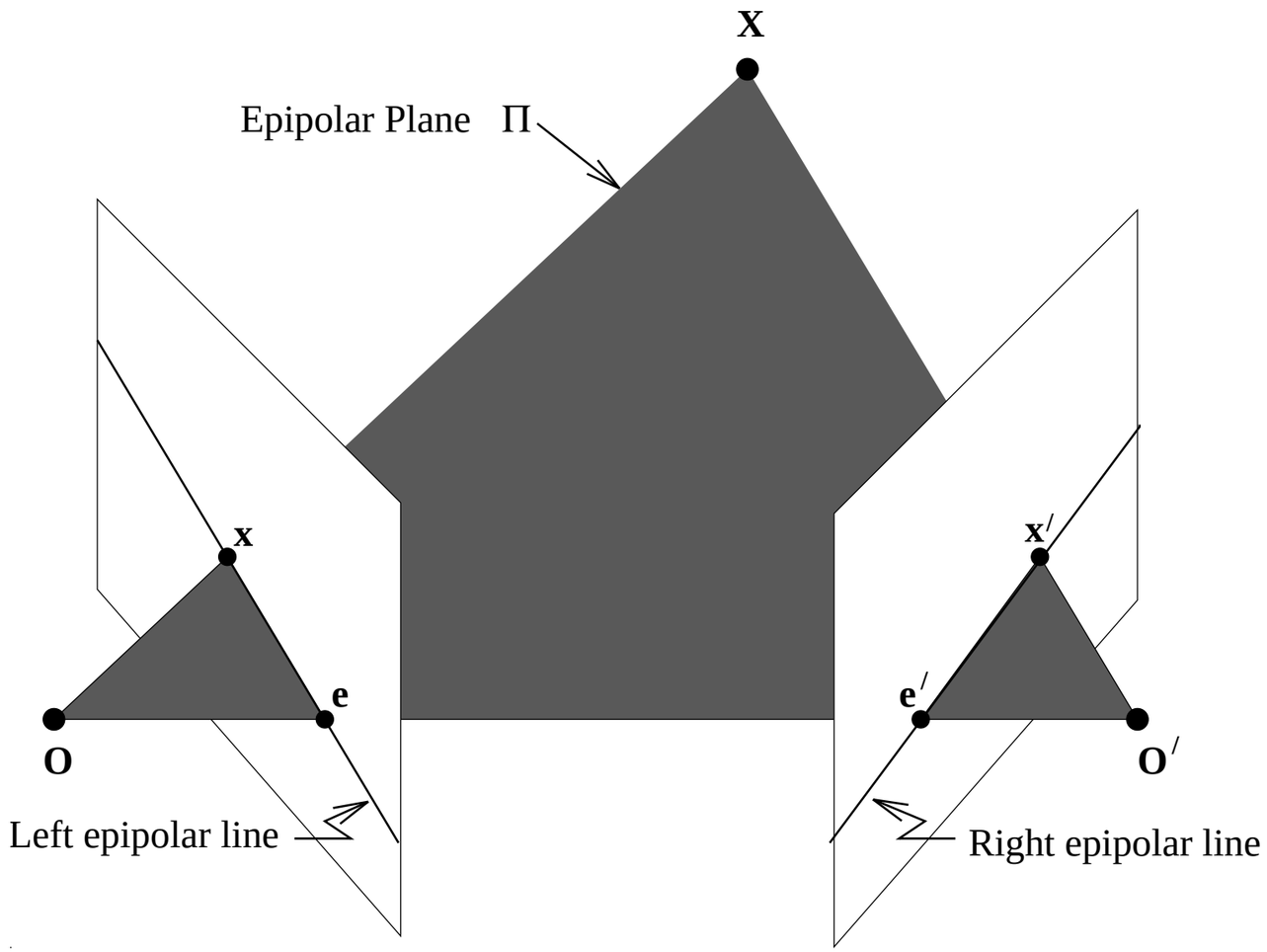
Correspondence Geometry

Given the image of a point in one view, what can we say about its position in another?



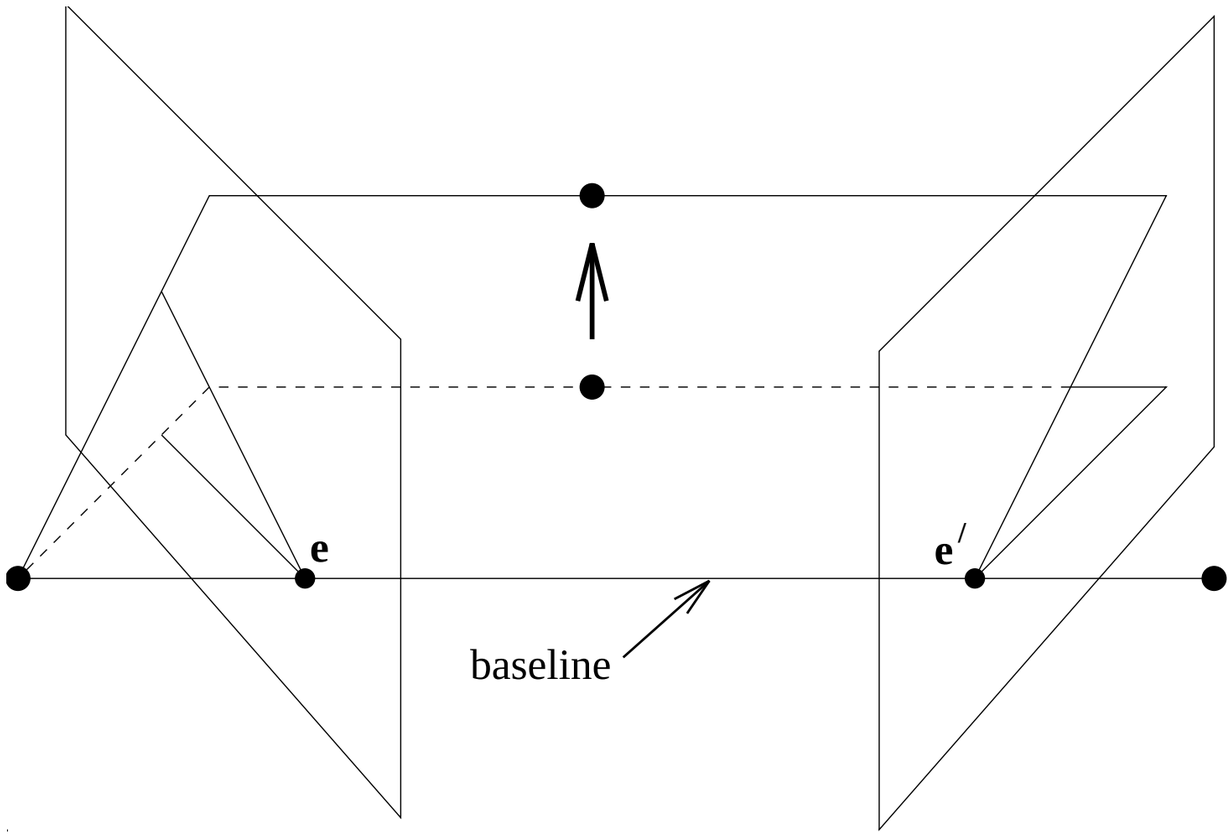
- A point in one image “generates” a line in the other image.
- This line is known as an epipolar line, and the geometry which gives rise to it is known as epipolar geometry.

Epipolar Geometry



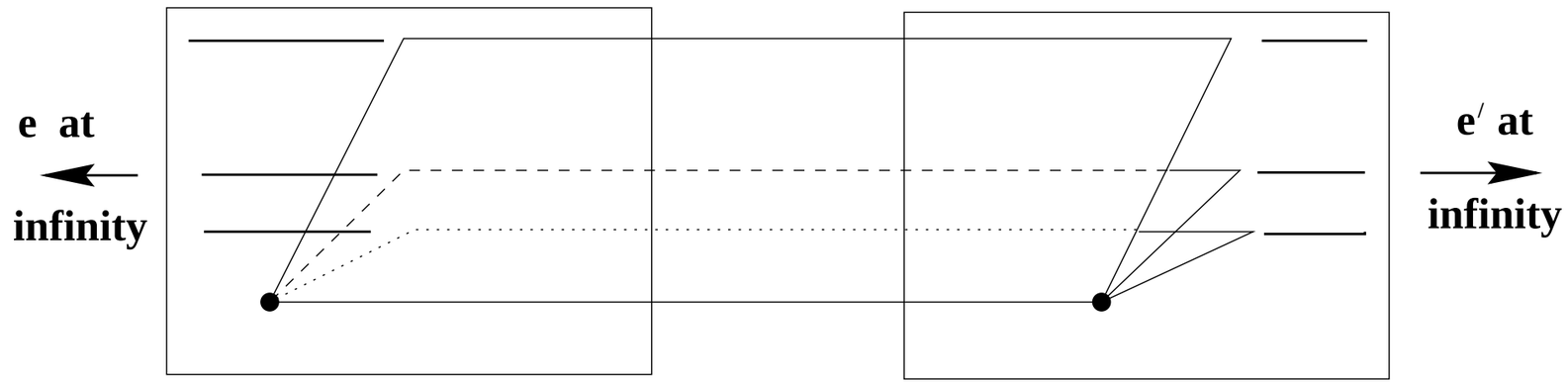
- The epipolar line $\mathbf{l}'$ is the image of the ray through $\mathbf{x}$.
- The epipole $\mathbf{e}'$ is the point of intersection of the line joining the camera centres—the baseline—with the image plane.
- The epipole is also the image in one camera of the centre of the other camera.
- All epipolar lines intersect in the epipole.

Epipolar pencil



As the position of the 3D point $\mathbf{X}$ varies, the epipolar planes “rotate” about the baseline. This family of planes is known as an epipolar pencil. All epipolar lines intersect at the epipole.

Epipolar geometry example



Epipolar geometry depends only on the relative pose (position and orientation) and internal parameters of the two cameras, i.e. the position of the camera centres and image planes. It does not depend on structure (3D points external to the camera).

Homogeneous Notation Interlude

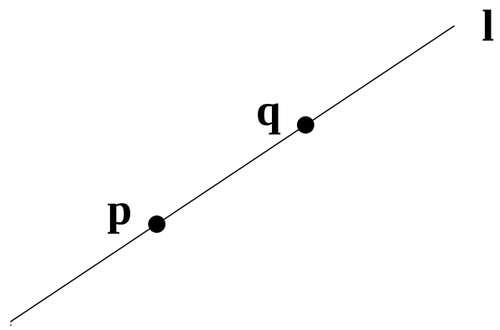
- A line $\mathbf{l}$ is represented by the homogeneous 3-vector

$$\mathbf{l} = \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix}$$

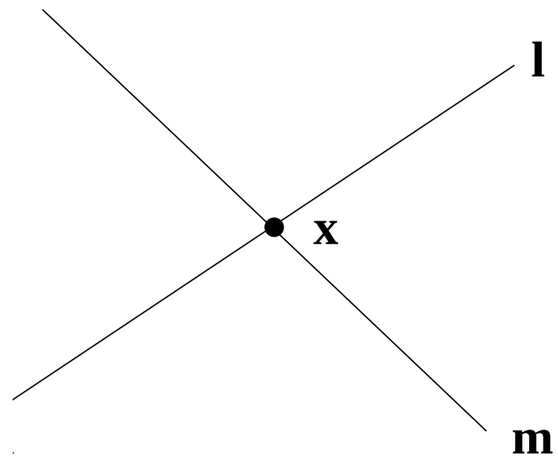
for the line $l_1 x + l_2 y + l_3 = 0$. Only the ratio of the homogeneous line coordinates is significant.

- point on line: $\mathbf{l}.\mathbf{x} = 0$ or $\mathbf{l}^\top \mathbf{x} = 0$ or $\mathbf{x}^\top \mathbf{l} = 0$

- two points define a line: $\mathbf{l} = \mathbf{p} \times \mathbf{q}$



- two lines define a point: $\mathbf{x} = \mathbf{l} \times \mathbf{m}$



Matrix notation for vector product

The vector product $\mathbf{v} \times \mathbf{x}$ can be represented as a matrix multiplication

$$\mathbf{v} \times \mathbf{x} = [\mathbf{v}]_{\times}\mathbf{x}$$

where

$$[\mathbf{v}]_{\times} = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix}$$

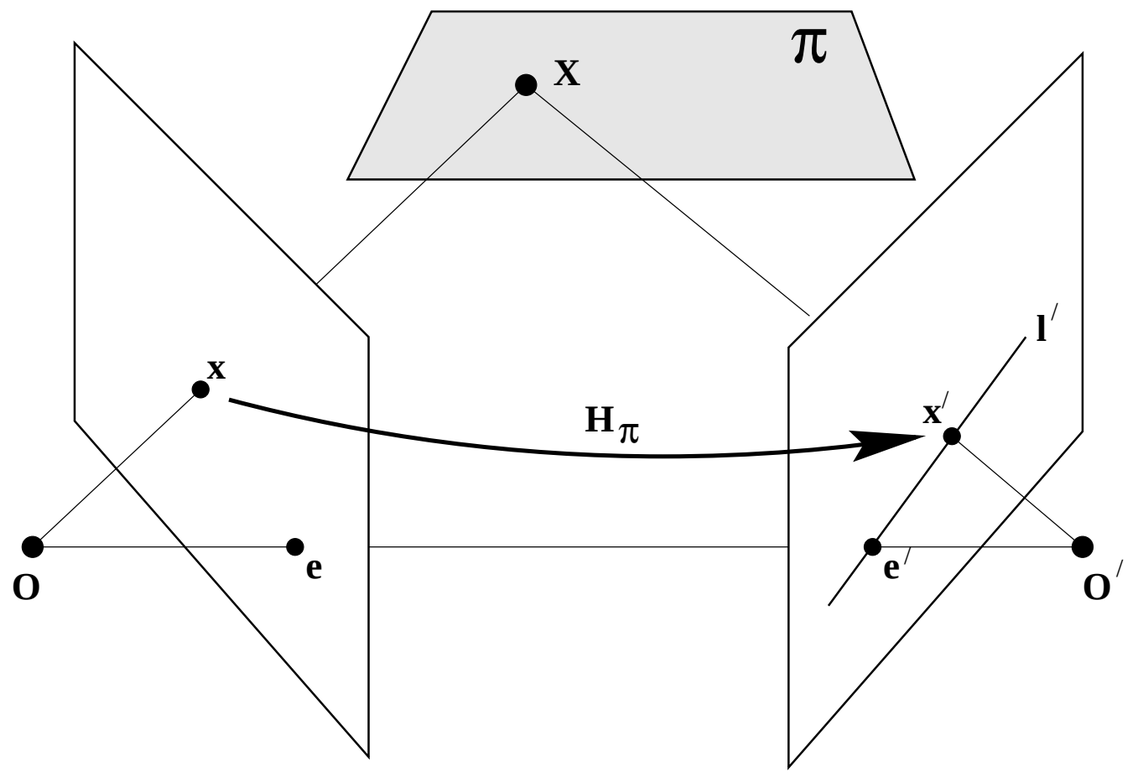
- $[\mathbf{v}]_{\times}$ is a 3×3 skew-symmetric matrix of rank 2.
- $\mathbf{v}$ is the null-vector of $[\mathbf{v}]_{\times}$, since $\mathbf{v} \times \mathbf{v} = [\mathbf{v}]_{\times}\mathbf{v} = \mathbf{0}$.

Algebraic representation - the Fundamental Matrix

$$\mathbf{x}'^{\top}\,\mathtt{F}\,\mathbf{x} = 0 \qquad \mathbf{l}' \;=\; \mathtt{F}\,\mathbf{x}$$

- $\mathtt{F}$ is a 3×3 rank 2 homogeneous matrix
- $\mathtt{F}^{\top}\mathbf{e}' = \mathbf{0}$
- It has 7 degrees of freedom
- Counting: $2 \times 11 - 15 = 7$.
- Compute from 7 image point correspondences

Fundamental matrix - sketch derivation



Step 1: Point transfer via a plane $\mathbf{x}' = \mathtt{H}_\pi \mathbf{x}$

Step 2 : Construct the epipolar line $\mathbf{l}' = \mathbf{e}' \times \mathbf{x}' = [\mathbf{e}']_\times \mathbf{x}'$

$\mathbf{l}' = [\mathbf{e}']_\times \mathtt{H}_\pi \mathbf{x} = \mathtt{F}\mathbf{x}$

$\mathtt{F} = [\mathbf{e}']_\times \mathtt{H}_\pi$

This shows that $\mathtt{F}$ is a 3×3 rank 2 matrix.

Properties of $\mathtt{F}$

- $\mathtt{F}$ is a rank 2 homogeneous matrix with 7 degrees of freedom.
- **Point correspondence**: If $\mathbf{x}$ and $\mathbf{x}'$ are corresponding image points, then $\mathbf{x}'^{\top}\mathtt{F}\mathbf{x} = 0$.
- **Epipolar lines**:
  - $\mathbf{l}' = \mathtt{F}\mathbf{x}$ is the epipolar line corresponding to $\mathbf{x}$.
  - $\mathbf{l} = \mathtt{F}^{\top}\mathbf{x}'$ is the epipolar line corresponding to $\mathbf{x}'$.
- **Epipoles**:
  - $\mathtt{F}\mathbf{e} = \mathbf{0} \qquad \mathtt{F}^{\top}\mathbf{e}' = \mathbf{0}$
- **Computation from camera matrices** $\mathtt{P}, \mathtt{P}'$:
  - $\mathtt{F} = [\mathtt{P}'\mathbf{C}]_{\times}\mathtt{P}'\mathtt{P}^{+}$, where $\mathtt{P}^{+}$ is the pseudo-inverse of $\mathtt{P}$, and $\mathbf{C}$ is the centre of the first camera. Note, $\mathbf{e}' = \mathtt{P}'\mathbf{C}$.
  - Canonical cameras, $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}], \mathtt{P}' = [\mathtt{M} \mid \mathbf{m}]$,
    $\mathtt{F} = [\mathbf{e}']_{\times}\mathtt{M} = \mathtt{M}^{-\top}[\mathbf{e}]_{\times}$, where $\mathbf{e}' = \mathbf{m}$ and $\mathbf{e} = \mathtt{M}^{-1}\mathbf{m}$.

Plane induced homographies given $\mathtt{F}$

Given the fundamental matrix $\mathtt{F}$ between two views, the homography induced by a world plane is

$$\mathtt{H} = [\mathbf{e}']_\times \mathtt{F} + \mathbf{e}'\mathbf{v}^\top$$

where $\mathbf{v}$ is the inhomogeneous 3-vector which parametrizes the 3-parameter family of planes.

e.g. compute plane from 3 point correspondences.

Given a homography $\hat{\mathtt{H}}$ induced by a particular world plane, then a homography induced by any plane may be computed as

$$\mathtt{H} = \hat{\mathtt{H}} + \mathbf{e}'\mathbf{v}^{*\top}$$

Projective Reconstruction from 2 views

Statement of the problem

Given

Corresponding points $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ in two images.

Find

Cameras $\mathtt{P}$ and $\mathtt{P}'$ and 3D points $\mathbf{X}_i$ such that

$$\mathbf{x}_i = \mathtt{P}\mathbf{X}_i \quad ; \quad \mathbf{x}'_i = \mathtt{P}\mathbf{X}_i$$

Projective ambiguity of reconstruction

- Solution is not unique without camera calibration
- Solution is unique up to a projective mapping :

$$\begin{aligned} \mathtt{P} &\mapsto \mathtt{P}\mathtt{H}^{-1} \\ \mathtt{P}' &\mapsto \mathtt{P}'\mathtt{H}^{-1} \\ \mathbf{X}_i &\mapsto \mathtt{H}\mathbf{X}_i \end{aligned}$$

Then verify

$$\begin{aligned} \mathbf{x}_i &= (\mathtt{P}\mathtt{H}^{-1})(\mathtt{H}\mathbf{X}_i) = \mathtt{P}\mathbf{X}_i \\ \mathbf{x}'_i &= (\mathtt{P}'\mathtt{H}^{-1})(\mathtt{H}\mathbf{X}_i) = \mathtt{P}'\mathbf{X}_i \end{aligned}$$

- Same problem holds however many views we have

Projective Distortion demo

< Projective distortion demo >

Basic Theorem

Given sufficiently many points to compute unique fundamental matrix :

- 8 points in general position
- 7 points not on a ruled quadric with camera centres

Then 3D points may be constructed from two views **Up to a 3D projective transformation**

- Except for points on the line between the camera centres.

Steps of projective reconstruction

Reconstruction takes place in the following steps :

- Compute the fundamental matrix $\mathtt{F}$ from point correspondences
- Factor the fundamental matrix as

$$\mathtt{F} = [\mathbf{t}]_{\times}\mathtt{M}$$

- The two camera matrices are

$$\mathtt{P} = [\mathtt{I} \mid \mathbf{0}] \text{ and } \mathtt{P}' = [\mathtt{M} \mid \mathbf{t}] \ .$$

- Compute the points $\mathbf{X}_i$ by triangulation

Details of Projective Reconstruction - Computation of F.

Methods of computation of F left until later

Several methods are available :

(i) Normalized 8-point algorithm
(ii) Algebraic minimization
(iii) Minimization of epipolar distance
(iv) Minimization of symmetric epipolar distance
(v) Maximum Likelihood (Gold-standard) method.
(vi) Others , . . .

Factorization of the fundamental matrix

SVD method

(i) Define

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

(ii) Compute the SVD

$$\mathtt{F} = \mathtt{U}\mathtt{D}\mathtt{V}^\top \text{ where } \mathtt{D} = \mathrm{diag}(r, s, 0)$$

(iii) Factorization is

$$\mathtt{F} = (\mathtt{U}\mathtt{Z}\mathtt{U}^\top)(\mathtt{U}\mathtt{Z}\mathtt{D}\mathtt{V}^\top)$$

- Simultaneously corrects $\mathtt{F}$ to a singular matrix.

Factorization of the fundamental matrix

Direct formula

Let $\mathbf{e}'$ be the epipole.

Solve $\mathbf{e}'^{\top}\mathtt{F} = 0$

Specific formula

$$\mathtt{P} = [\mathtt{I} \mid \mathbf{0}] \;\; ; \;\; \mathtt{P}' = [[\mathbf{e}']_{\times}\mathtt{F} \mid \mathbf{e}'] = [\mathtt{M} \mid \mathbf{e}']$$

This solution is identical to the SVD solution.

Non-uniqueness of factorization

- Factorization of the fundamental matrix is not unique.
- General formula : for varying $\mathbf{v}$ and λ

$$\mathtt{P} = [\mathtt{I} \mid \mathbf{0}] \;\; ; \;\; \mathtt{P}' = [\mathtt{M} + \mathbf{e}'\mathbf{v}^\top \mid \lambda \mathbf{e}']$$

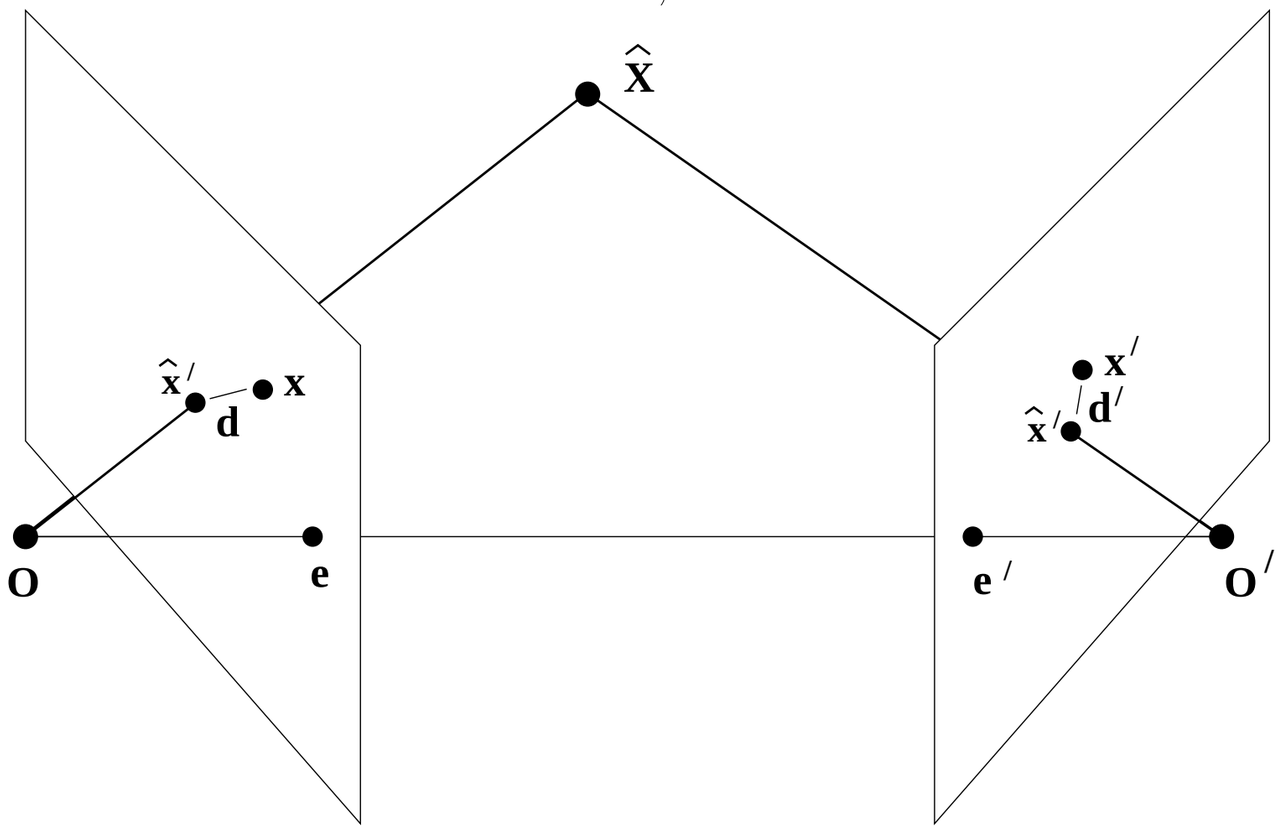
- Difference factorizations give configurations varying by a projective transformation.
- 4-parameter family of solutions with $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$.

Triangulation

Triangulation :

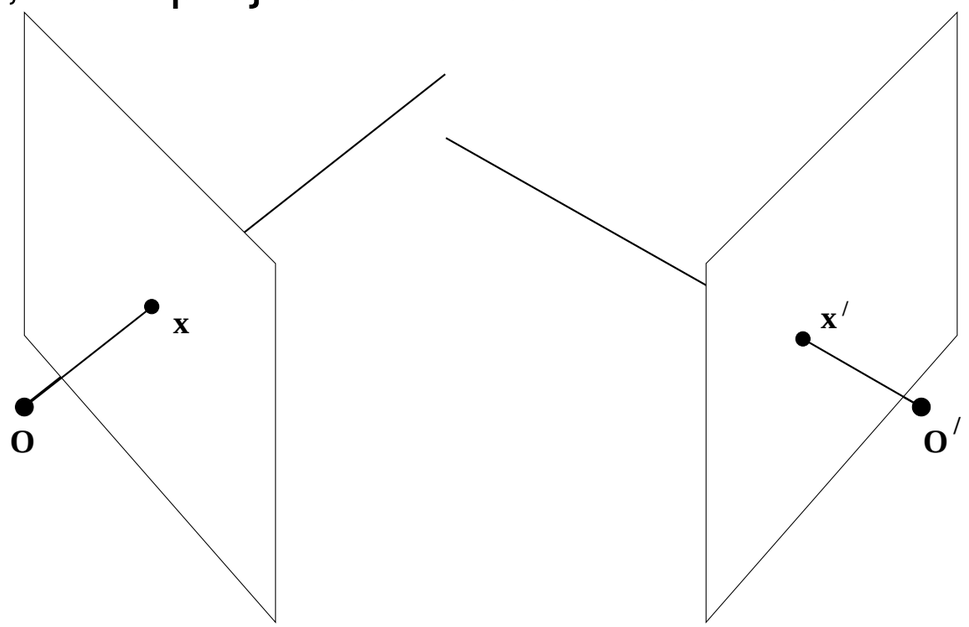
- Knowing $\mathtt{P}$ and $\mathtt{P}'$
- Knowing $\mathbf{x}$ and $\mathbf{x}'$
- Compute $\mathbf{X}'$ such that

$$\mathbf{x} = \mathtt{P}\mathbf{X} \quad ; \quad \mathbf{x}' = \mathtt{P}'\mathbf{X}$$

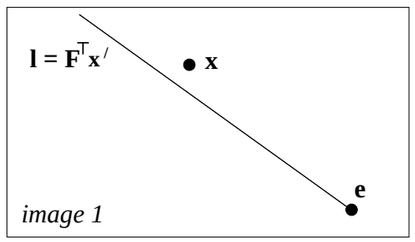


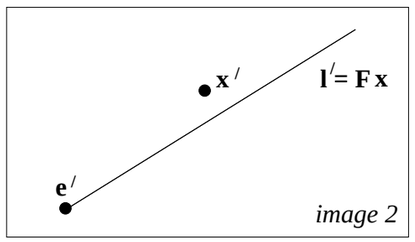
Triangulation in presence of noise

- In the presence of noise, back-projected lines do not intersect.



Rays do not intersect in space





Measured points do not lie on corresponding epipolar lines

Which 3D point to select ?

- Mid-point of common perpendicular to the rays ?
  - **Not** a good choice in projective environment.
  - Concepts of mid-point and perpendicular are meaningless under projective distortion.
- Weighted point on common perpendicular, weighted by distance from camera centres ?
  - Distance is also undefined concept.
- Some algebraic distance ?
  - Write down projection equations and solve ?
  - Linear least squares solution.
  - Minimizes nothing meaningful.

Problem statement

- Assume camera matrices are given without error, up to projective distortion.
- Hence `F` is known.
- A pair of matched points in an image are given.
- Possible errors in the position of matched points.
- Find 3D point that minimizes suitable error metric.
- Method must be invariant under 3D projective transformation.

Linear triangulation methods

- Direct analogue of the linear method of camera resectioning.
- Given equations

$$\mathbf{x} = \mathtt{P}\mathbf{X}$$
$$\mathbf{x}' = \mathtt{P}'\mathbf{X}$$

- $\mathbf{p}^{i\top}$ are the rows of $\mathtt{P}$.
- Write as linear equations in $\mathbf{X}$

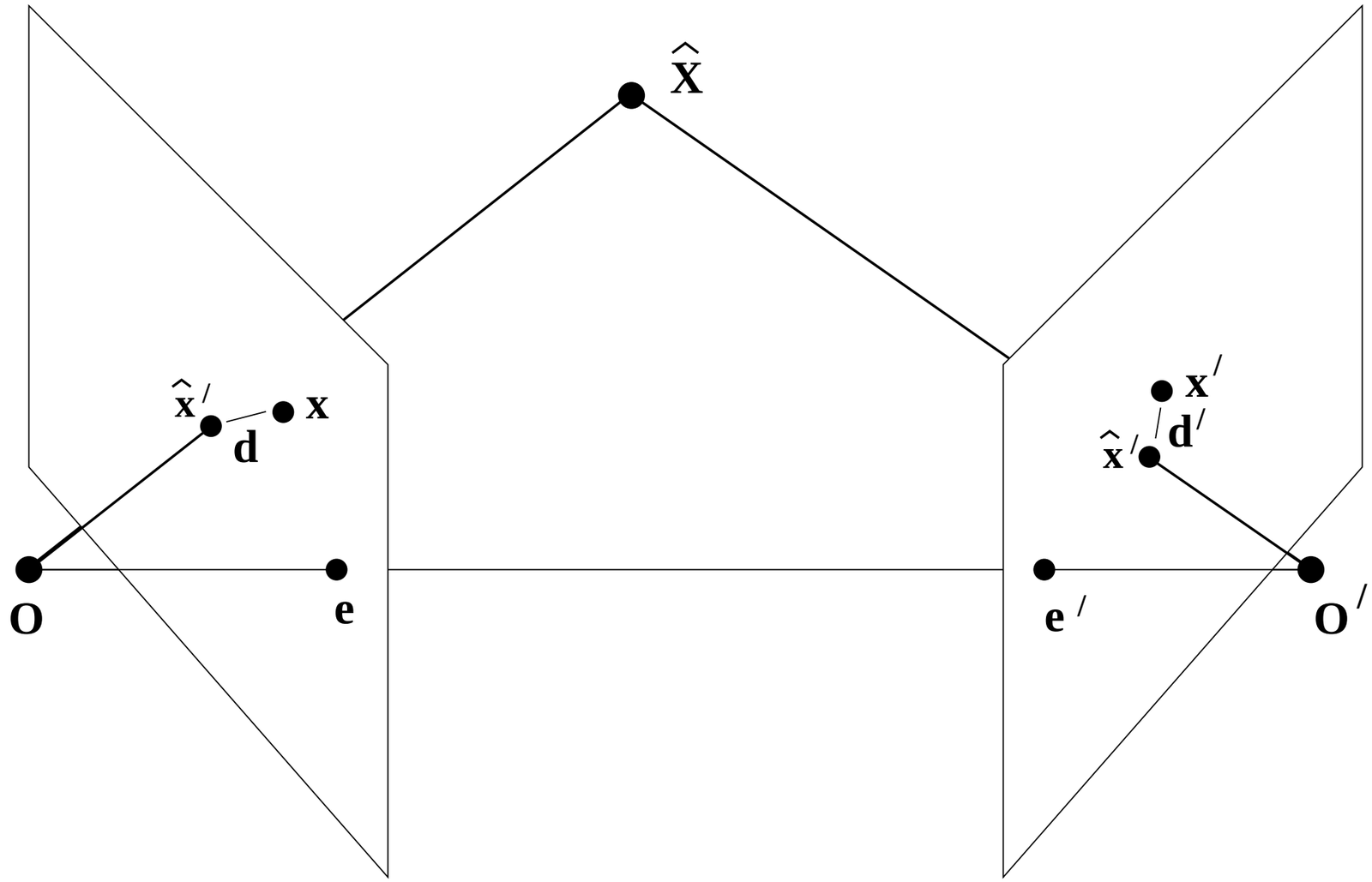
$$\begin{bmatrix} x\mathbf{p}^{3\top} - \mathbf{p}^{1\top} \\ y\mathbf{p}^{3\top} - \mathbf{p}^{2\top} \\ x'\mathbf{p}'^{3\top} - \mathbf{p}'^{1\top} \\ y\mathbf{p}'^{3\top} - \mathbf{p}'^{2\top} \end{bmatrix} \mathbf{X} = 0$$

- Solve for $\mathbf{X}$.
- Generalizes to point match in several images.
- Minimizes no meaningful quantity – not optimal.

Minimizing geometric error

- Point $\mathbf{X}$ in space maps to projected points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ in the two images.
- Measured points are $\mathbf{x}$ and $\mathbf{x}'$.
- Find $\mathbf{X}$ that minimizes difference between projected and measured points.

Geometric error . . .



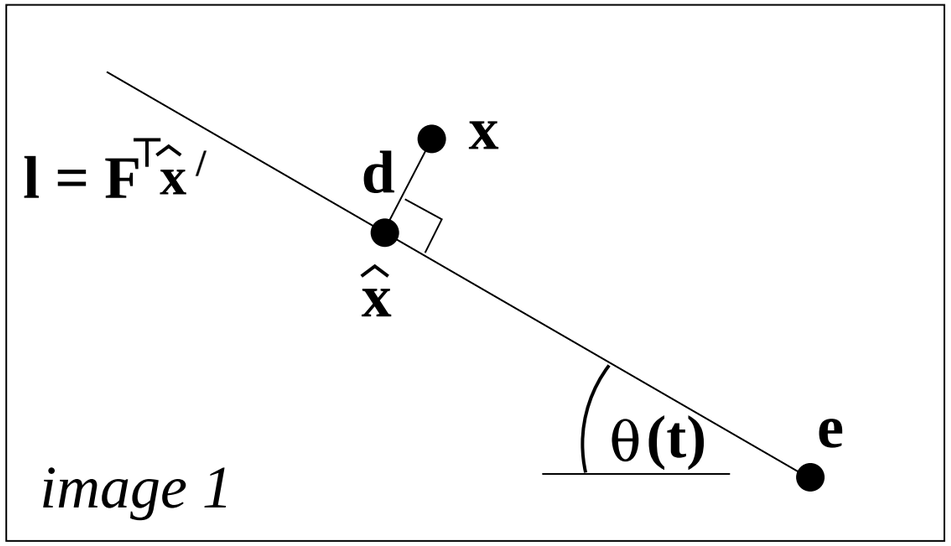
Cost function

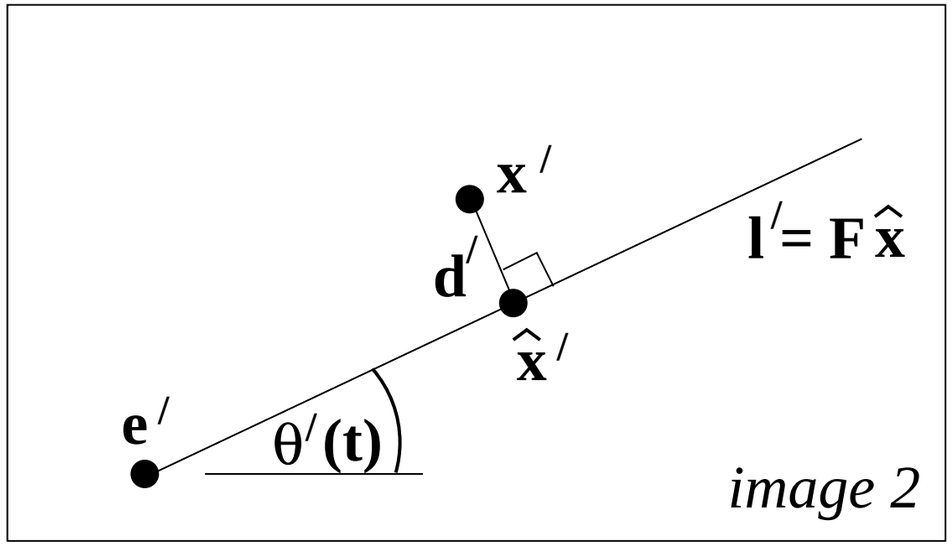
$$\mathcal{C}(\mathbf{X}) = d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \hat{\mathbf{x}}')^2$$

Different formulation of the problem

Minimization problem may be formulated differently:

- Minimize
$$d(\mathbf{x}, \mathbf{l})^2 + d(\mathbf{x}', \mathbf{l}')^2$$
- $\mathbf{l}$ and $\mathbf{l}'$ range over all choices of corresponding epipolar lines.
- $\hat{\mathbf{x}}$ is the closest point on the line $\mathbf{l}$ to $\mathbf{x}$.
- Same for $\hat{\mathbf{x}}'$.





Minimization method

Our strategy for minimizing cost function is as follows

(i) Parametrize the pencil of epipolar lines in the first image by a parameter t. Epipolar line is $\mathbf{l}(t)$.

(ii) Using the fundamental matrix $\mathtt{F}$, compute the corresponding epipolar line $\mathbf{l}'(t)$ in the second image.

(iii) Express the distance function $d(\mathbf{x}, \mathbf{l}(t))^2 + d(\mathbf{x}', \mathbf{l}'(t))^2$ explicitly as a function of t.

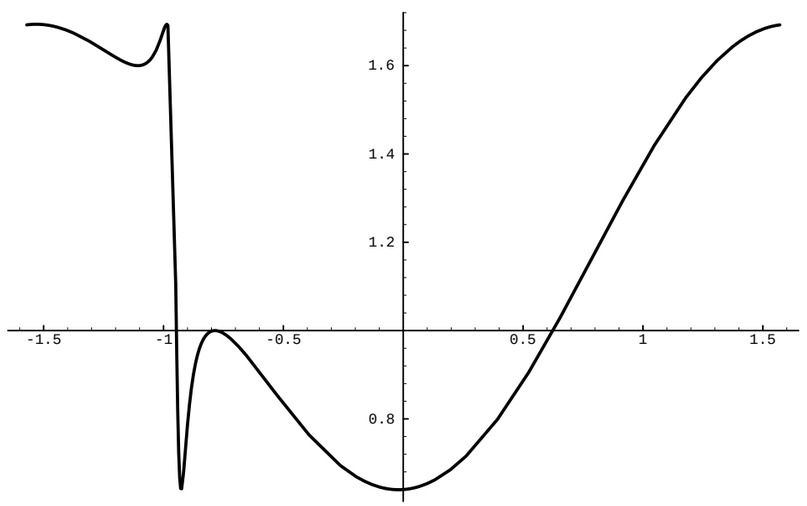
(iv) Find the value of t that minimizes this function.

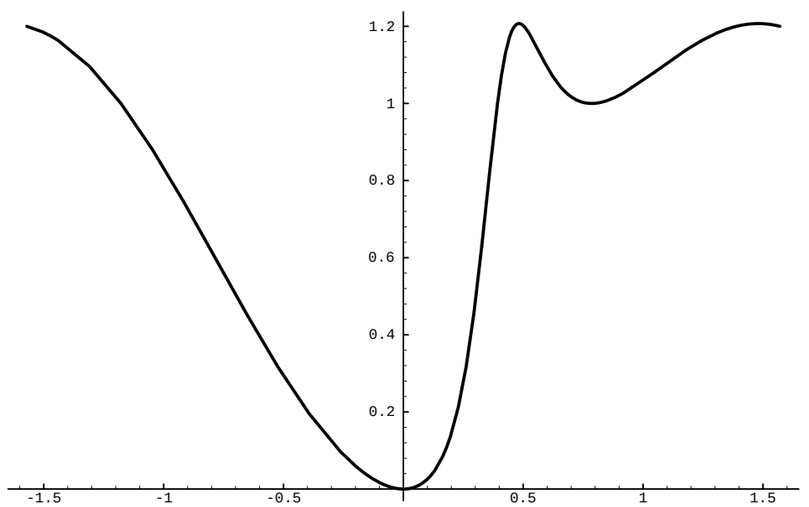
Minimization method ...

- Find the minimum of a function of a single variable, t.
- Problem in elementary calculus.
- Derivative of cost reduces to a 6-th degree polynomial in t.
- Find roots of derivative explicitly and compute cost function.
- Provides global minimum cost (guaranteed best solution).
- Details : See Hartley-Sturm “Triangulation”.

Multiple local minima

- Cost function may have local minima.
- Shows that gradient-descent minimization may fail.

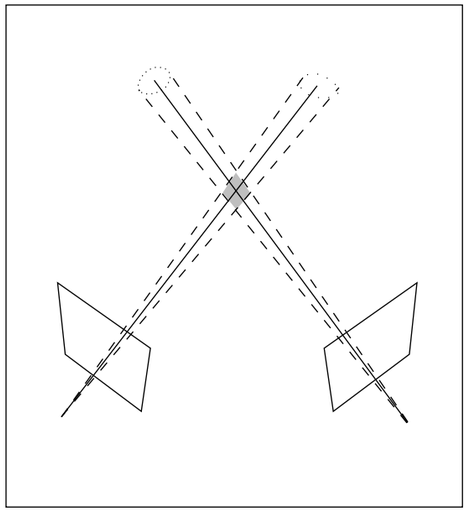


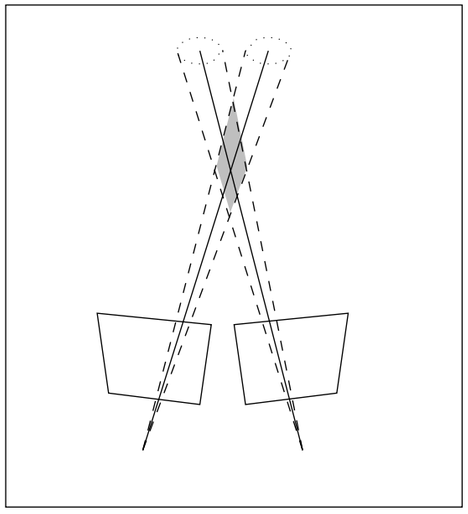


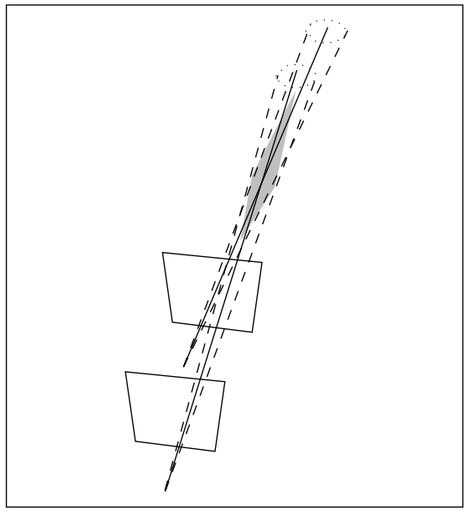
Left : Example of a cost function with three minima.

Right : Cost function for a perfect point match with two minima.

Uncertainty of reconstruction







Uncertainty of reconstruction. The shape of the uncertainty region depends on the angle between the rays.

Computation of the Fundamental Matrix

Basic equations

Given a correspondence

$$\mathbf{x} \leftrightarrow \mathbf{x}'$$

The basic incidence relation is

$$\mathbf{x}'^{\top}\mathbf{F}\mathbf{x} = 0$$

May be written

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0 \quad .$$

Single point equation - Fundamental matrix

Gives an equation :

$$(x'x, x'y, x', y'x, y'y, y', x, y, 1) \begin{pmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{pmatrix} = 0$$

where

$$\mathbf{f} = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})^\top$$

holds the entries of the Fundamental matrix

Total set of equations

$$\mathbf{Af} = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \begin{pmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{pmatrix} = \mathbf{0}$$

Solving the Equations

- Solution is determined up to scale only.
- Need 8 equations $\Rightarrow$ 8 points
- 8 points $\Rightarrow$ unique solution
- > 8 points $\Rightarrow$ least-squares solution.

Least-squares solution

(i) Form equations $\mathtt{A}\mathbf{f} = \mathbf{0}$.

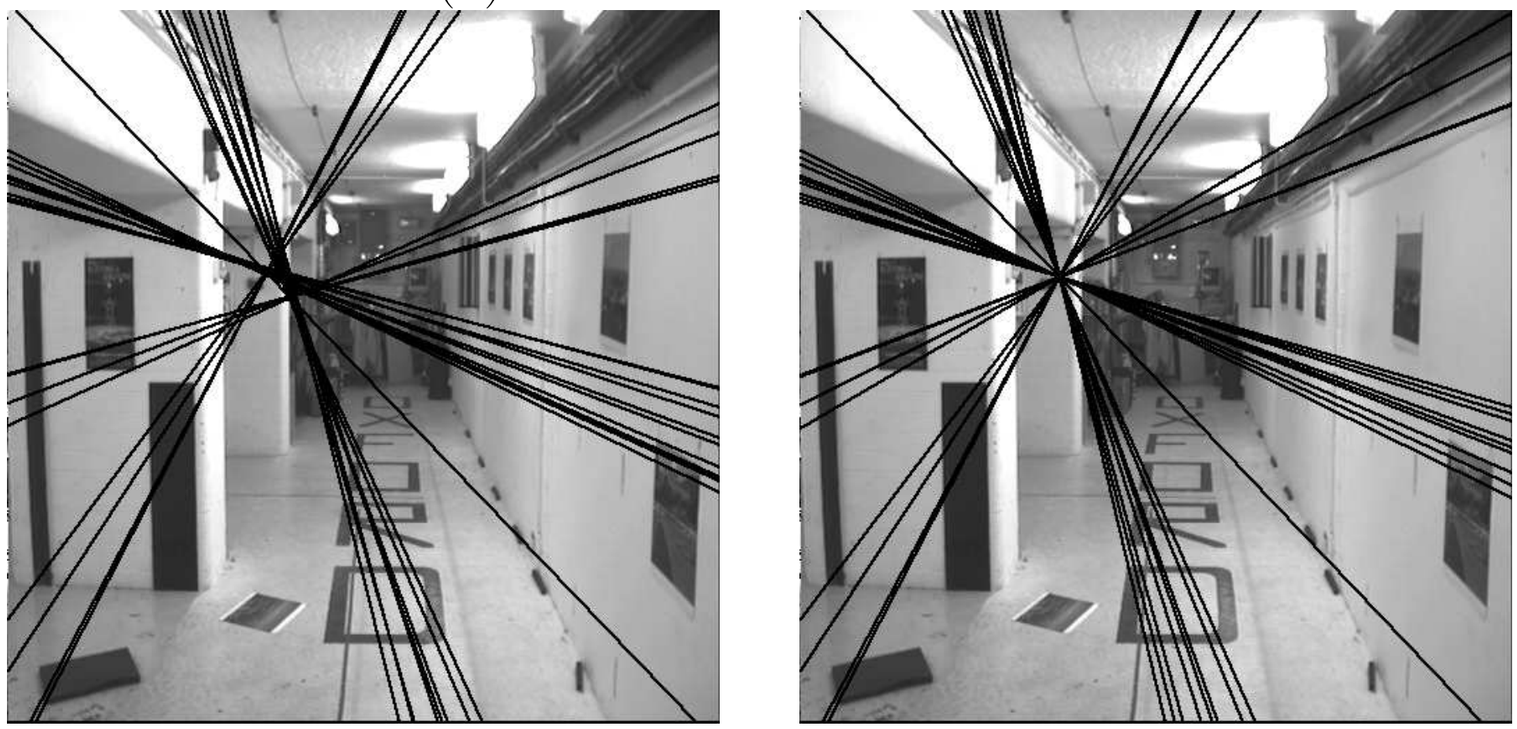
(ii) Take SVD : $\mathtt{A} = \mathtt{U}\mathtt{D}\mathtt{V}^\top$.

(iii) Solution is last column of $\mathtt{V}$ (corresp : smallest singular value)

(iv) Minimizes $||\mathtt{A}\mathbf{f}||$ subject to $||\mathbf{f}|| = 1$.

The singularity constraint

Fundamental matrix has rank 2 : $\det(\mathtt{F}) = 0$.



Left : Uncorrected $\mathtt{F}$ – epipolar lines are not coincident.

Right : Epipolar lines from corrected $\mathtt{F}$.

Computing $\mathtt{F}$ from 7 points

- $\mathtt{F}$ has 9 entries but is defined only up to scale.
- Singularity condition $\det \mathtt{F} = 0$ gives a further constraint.
- $\mathtt{F}$ has 3 rows $\Longrightarrow \det \mathtt{F} = 0$ is a cubic constraint.
- $\mathtt{F}$ has only 7 degrees of freedom.
- It is possible to solve for $\mathtt{F}$ from just 7 point correspondences.

7-point algorithm

Computation of $\mathtt{F}$ from 7 point correspondences

(i) Form the 7×9 set of equations $\mathtt{A}\mathbf{f} = 0$.

(ii) System has a 2-dimensional solution set.

(iii) General solution (use SVD) has form

$$\mathbf{f} = \lambda \mathbf{f}_0 + \mu \mathbf{f}_1$$

(iv) In matrix terms

$$\mathtt{F} = \lambda \mathtt{F}_0 + \mu \mathtt{F}_1$$

(v) Condition $\det \mathtt{F} = 0$ gives cubic equation in λ and μ.

(vi) Either one or three real solutions for ratio $\lambda : \mu$.

Correcting $\mathtt{F}$ using the Singular Value Decomposition

If $\mathtt{F}$ is computed linearly from 8 or more correspondences, singularity condition does not hold.

SVD Method

(i) SVD : $\mathtt{F} = \mathtt{U}\mathtt{D}\mathtt{V}^\top$

(ii) $\mathtt{U}$ and $\mathtt{V}$ are orthogonal, $\mathtt{D} = \mathrm{diag}(r, s, t)$.

(iii) $r \geq s \geq t$.

(iv) Set $\mathtt{F}' = \mathtt{U}\,\mathrm{diag}(r, s, 0)\,\mathtt{V}^\top$.

(v) Resulting $\mathtt{F}'$ is singular.

(vi) Minimizes the Frobenius norm of $\mathtt{F} - \mathtt{F}'$

(vii) $\mathtt{F}'$ is the ”closest” singular matrix to $\mathtt{F}$.

Complete 8-point algorithm

8 point algorithm has two steps :

(i) **Linear solution.** Solve $\mathtt{A}\mathbf{f} = 0$ to find $\mathtt{F}$.

(ii) **Constraint enforcement.** Replace $\mathtt{F}$ by $\mathtt{F}'$.

Warning This algorithm is unstable and should never be used with unnormalized data (see next slide).

The normalized 8-point algorithm

Raw 8-point algorithm performs badly in presence of noise.

Normalization of data

- 8-point algorithm is sensitive to origin of coordinates and scale.
- Data must be translated and scaled to “canonical” coordinate frame.
- Normalizing transformation is applied to both images.
- Translate so centroid is at origin
- Scale so that RMS distance of points from origin is $\sqrt{2}$.
- “Average point” is $(1, 1, 1)^\top$.

Normalized 8-point algorithm

(i) **Normalization:** Transform the image coordinates :

$$\hat{\mathbf{x}}_i = \mathtt{T}\mathbf{x}_i$$
$$\hat{\mathbf{x}}'_i = \mathtt{T}'\mathbf{x}'_i$$

(ii) **Solution:** Compute $\mathtt{F}$ from the matches $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$

$$\hat{\mathbf{x}}_i'^{\top}\widehat{\mathtt{F}}\hat{\mathbf{x}}_i = 0$$

(iii) **Singularity constraint :** Find closest singular $\widehat{\mathtt{F}}'$ to $\widehat{\mathtt{F}}$.

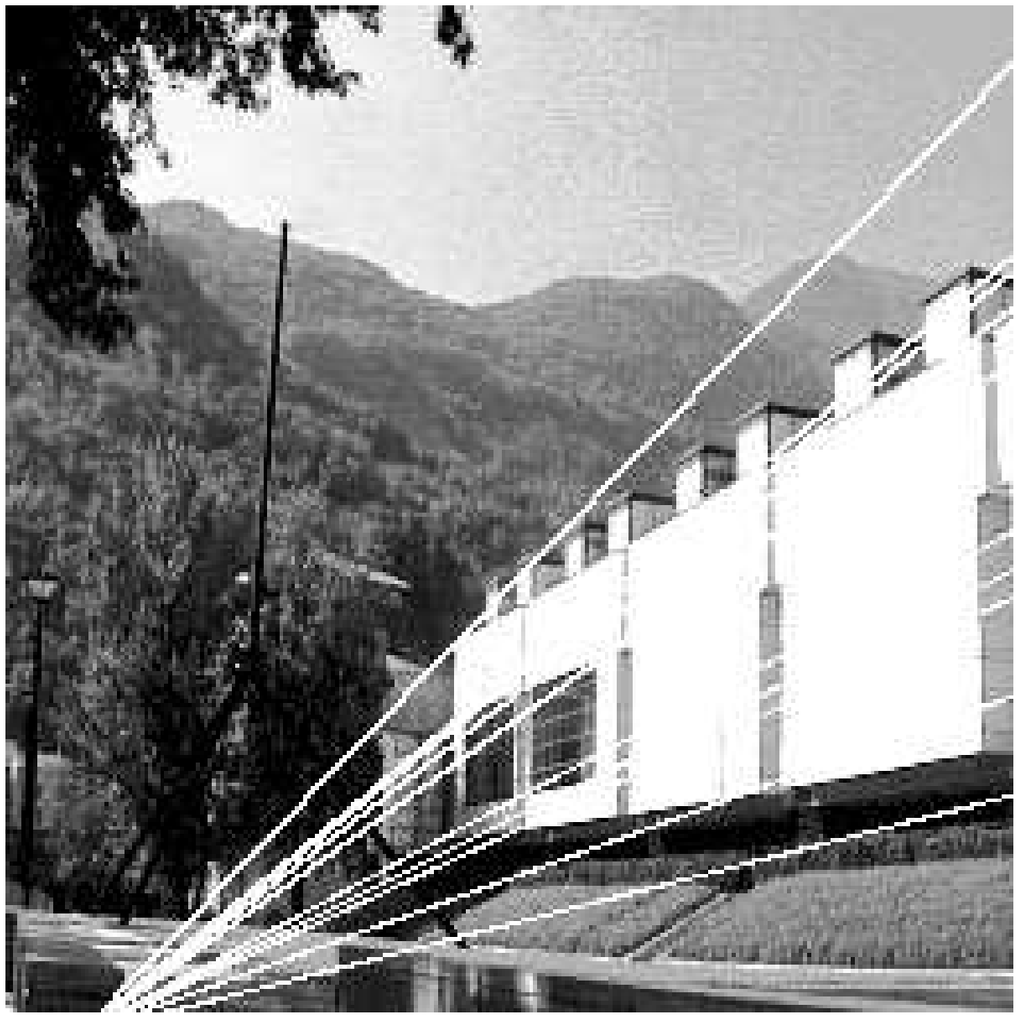
(iv) **Denormalization:** $\mathtt{F} = \mathtt{T}'^{\top}\widehat{\mathtt{F}}'\mathtt{T}$.

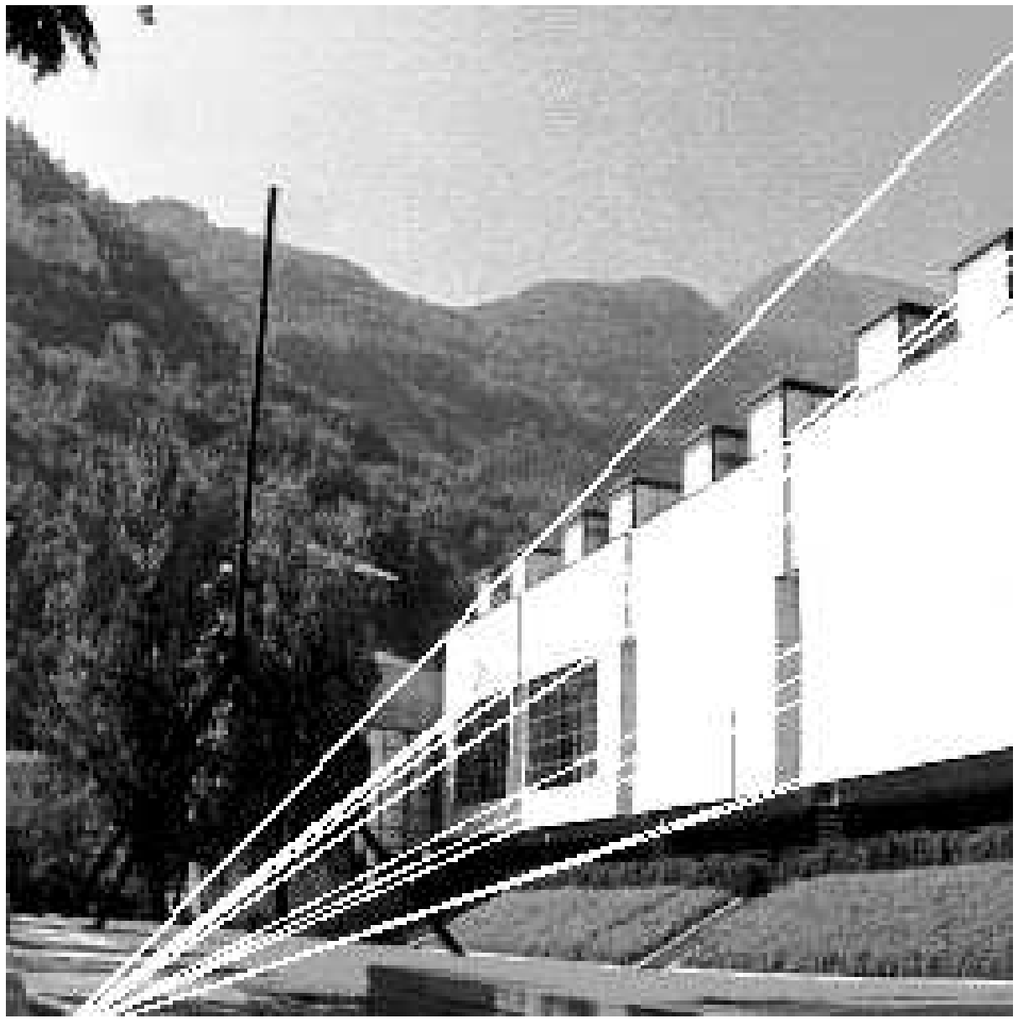
Comparison of Normalized and Unnormalized Algorithms

Lifia House images

Statue images

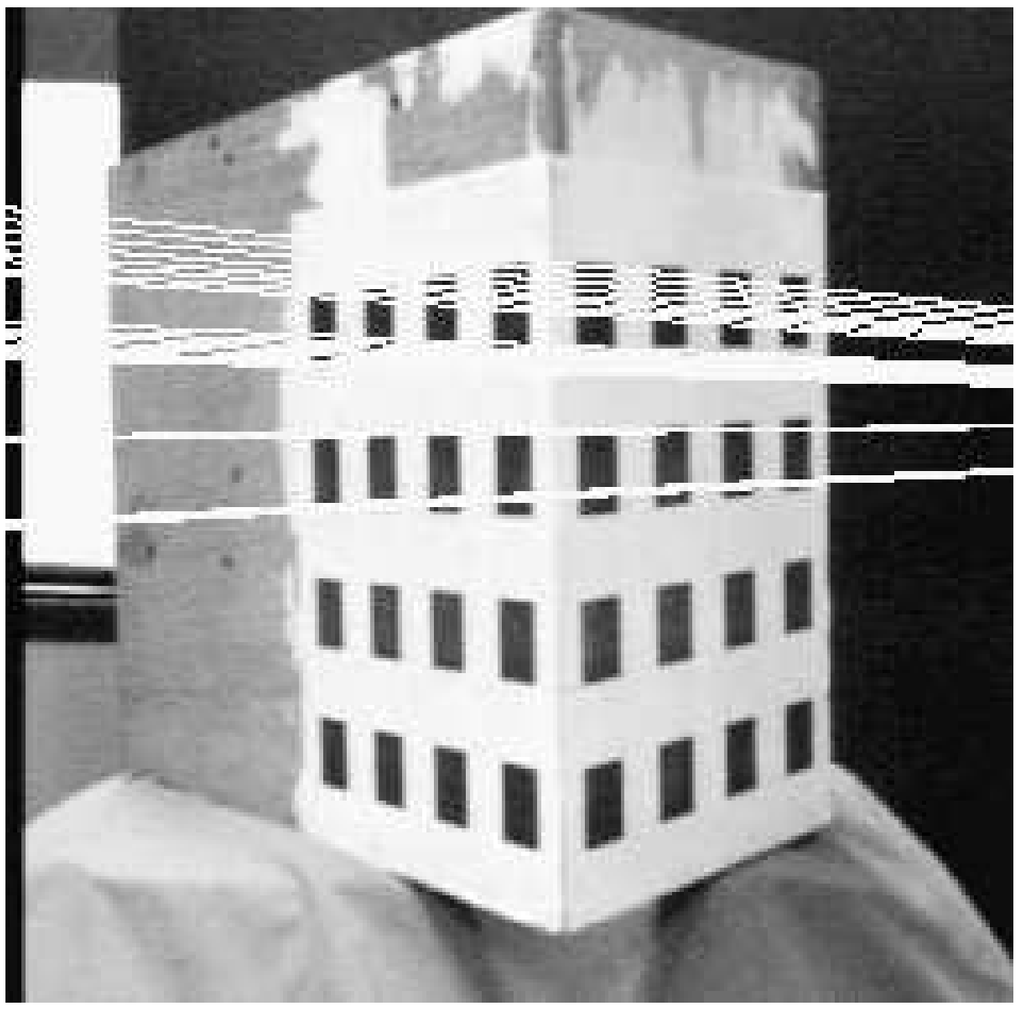
Grenoble Museum

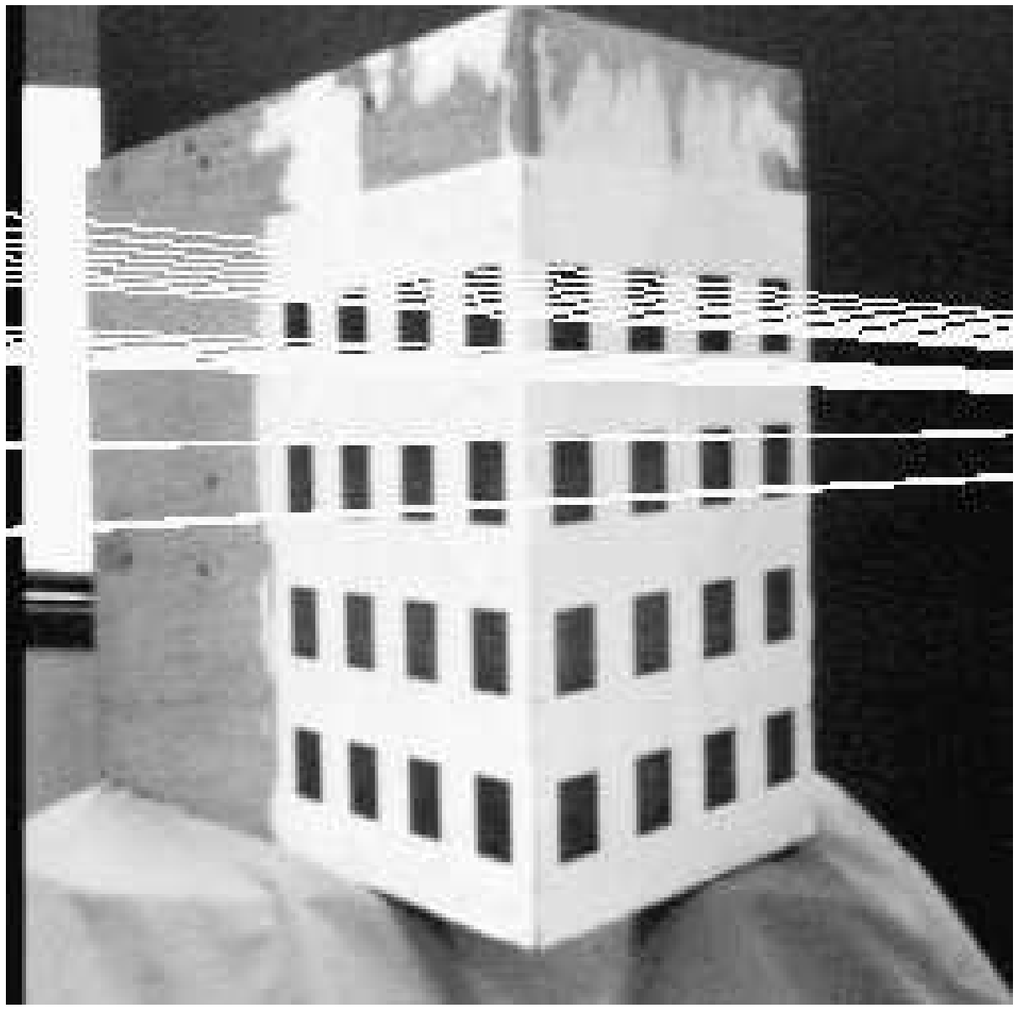




Oxford Basement

Calibration object





Testing methodology

(i) Point matches found in image pairs and outliers discarded.

(ii) Fundamental matrix was found from varying number (n) of points.

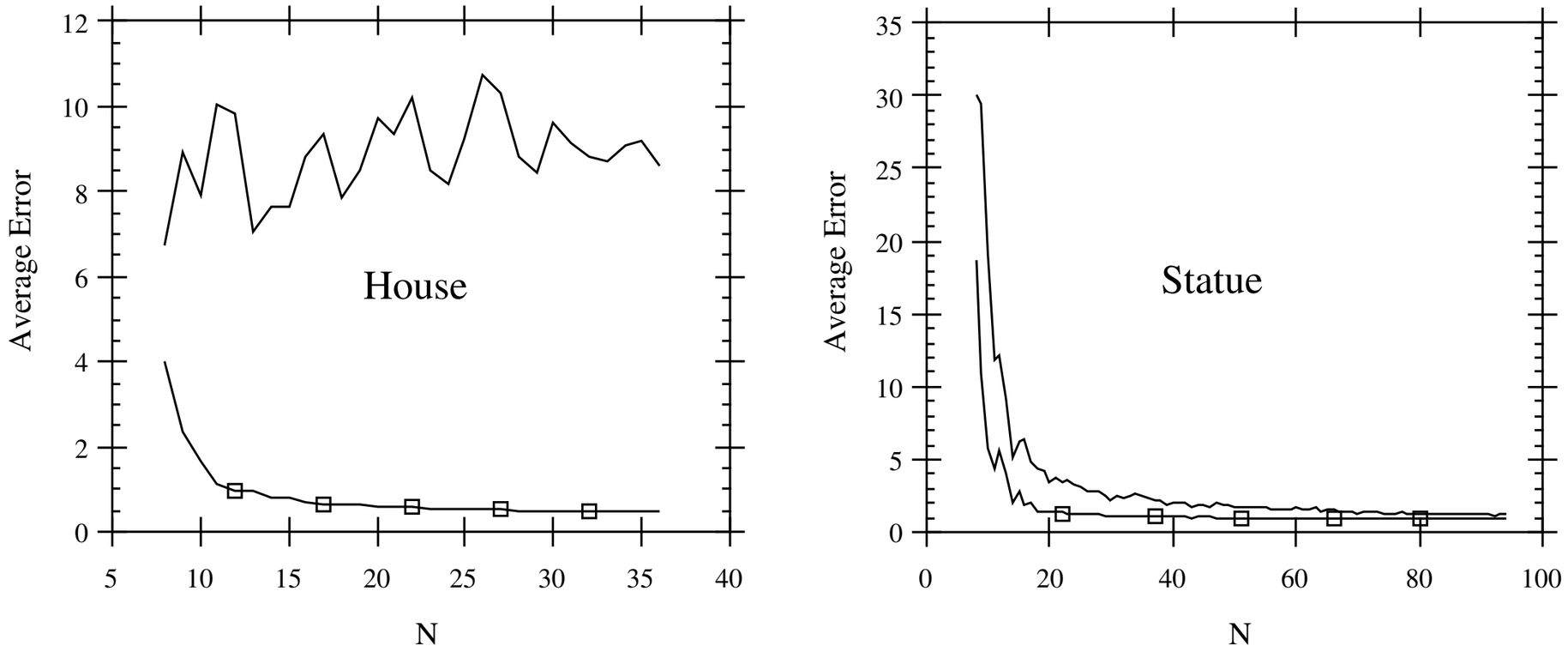
(iii) `F` was tested against other matched points not used to compute it.

(iv) Distance of a point from predicted epipolar line was the metric.

(v) 100 trials for each value of n.

(vi) Average error is plotted against n.

Comparison of normalized and unnormalized 8-point algorithms.

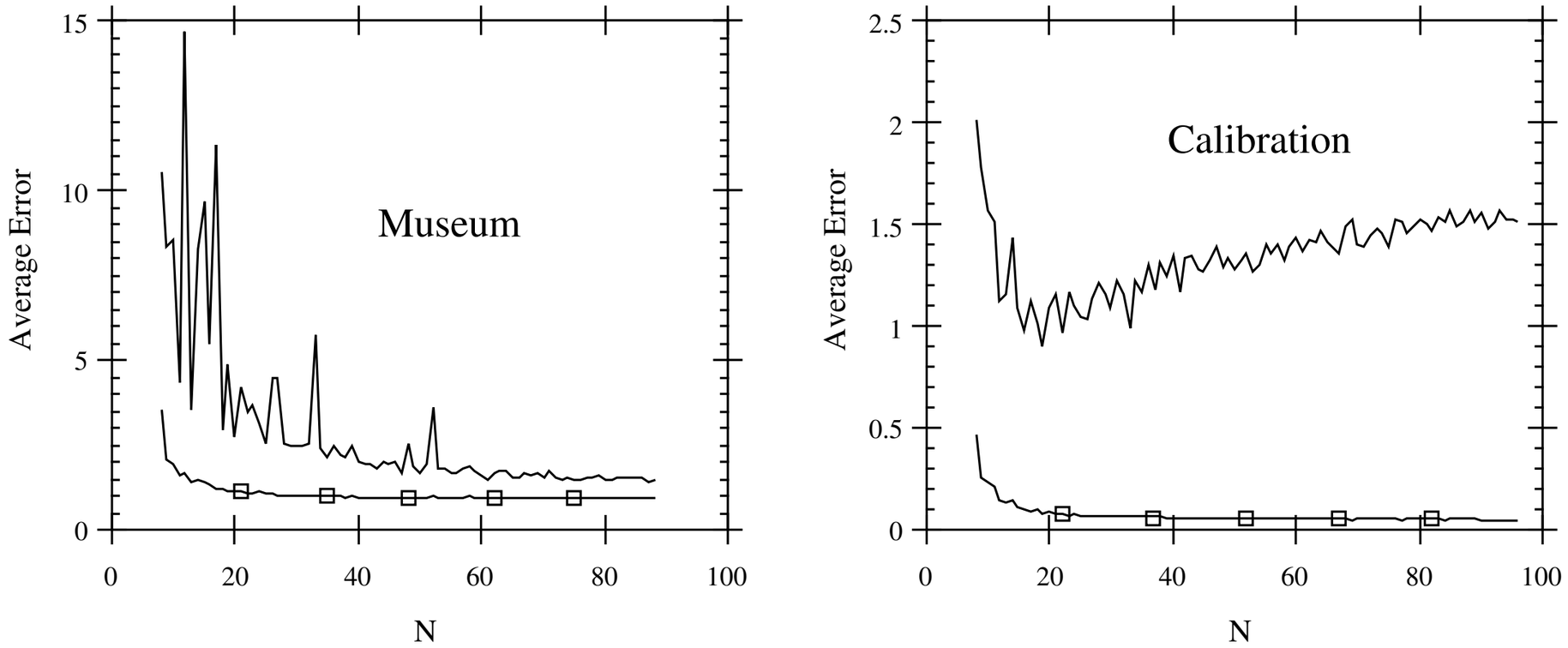


Distance of points from epipolar lines :

Top : Unnormalized 8-point algorithm.

Bottom : Normalized 8-point algorithm.

Comparison of normalized and unnormalized 8-point algorithms.

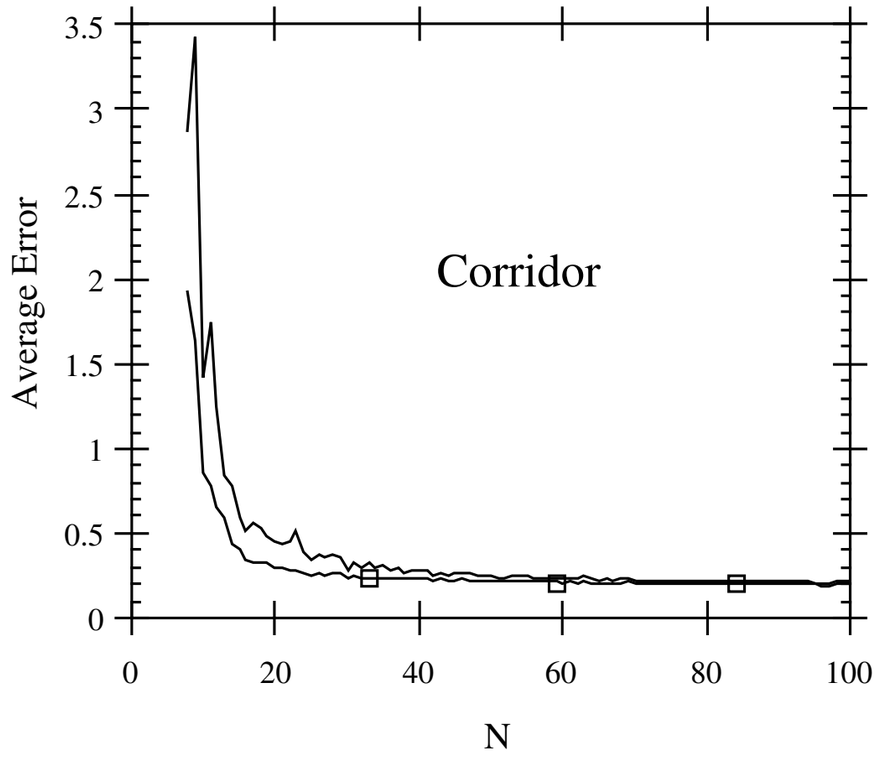


Distance of points from epipolar lines :

Top : Unnormalized 8-point algorithm.

Bottom : Normalized 8-point algorithm.

Comparison of normalized and unnormalized 8-point algorithms.



Distance of points from epipolar lines :

Top : Unnormalized 8-point algorithm.

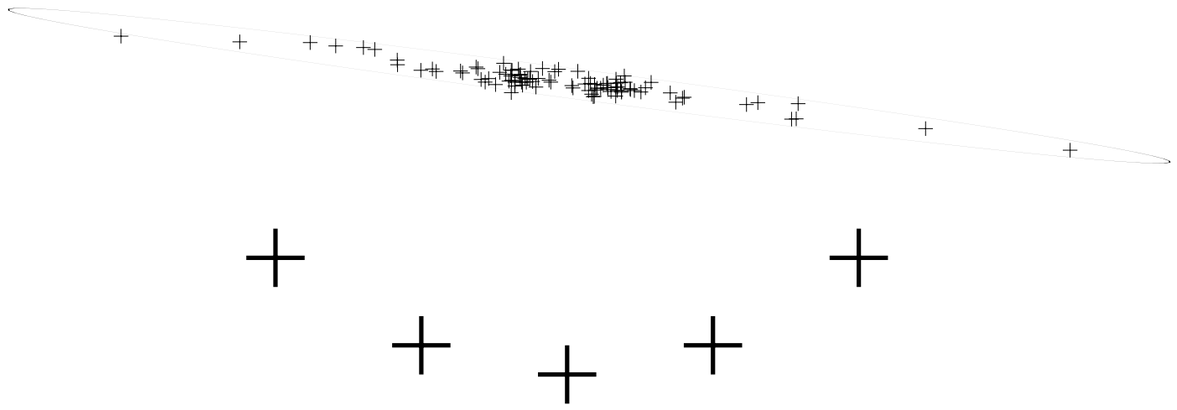
Bottom : Normalized 8-point algorithm.

Illustration of Effect of Normalization

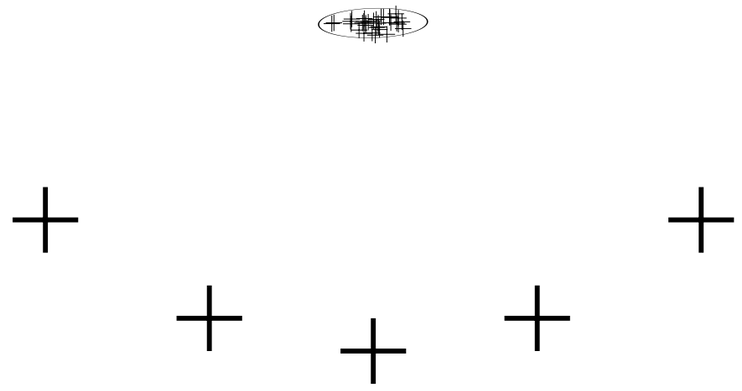
Similar problem : Computation of a 2D projective transformation given point matches in two images.

(i) Homography is computed from 5 noisy point matches.
(ii) Homography is applied to a further (6th) point
(iii) Noise level approximately equal to with of lines in the crosses (next page)
(iv) Repeated 100 times.
(v) Spread of the transformed 6th point is shown in relation to the 5 data points.
(vi) 95% ellipses are plotted.

Normalization and 2D homography computation

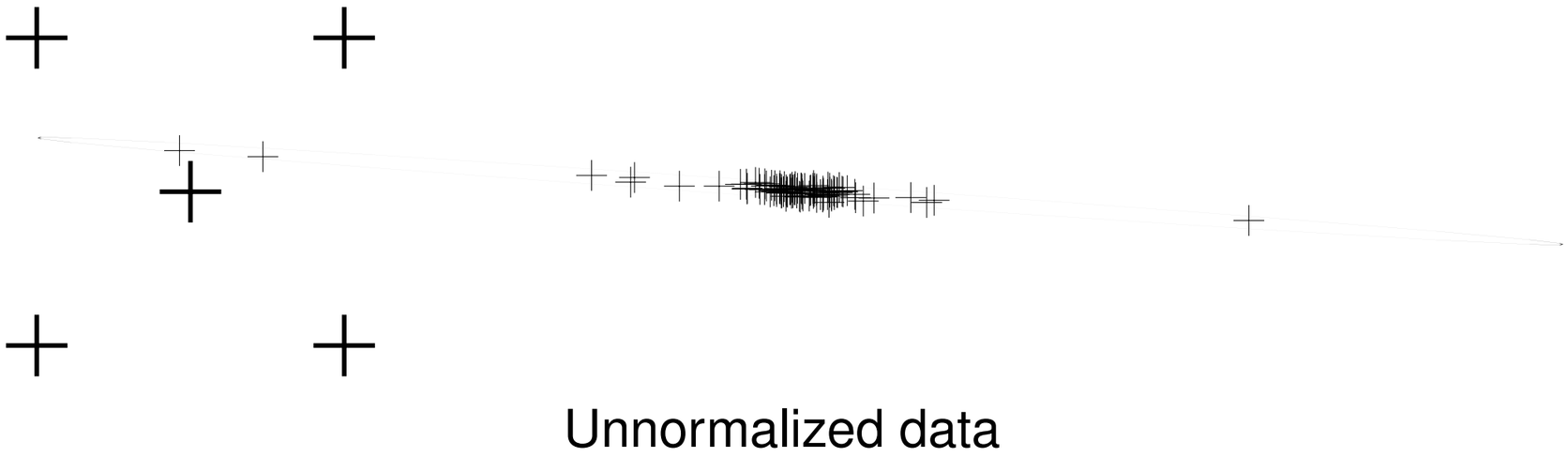


Unnormalized data

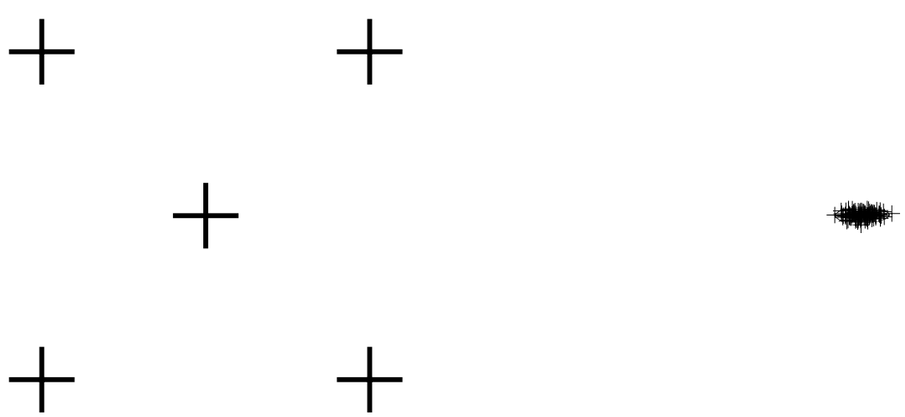


Normalized data

Normalization and 2D homography computation



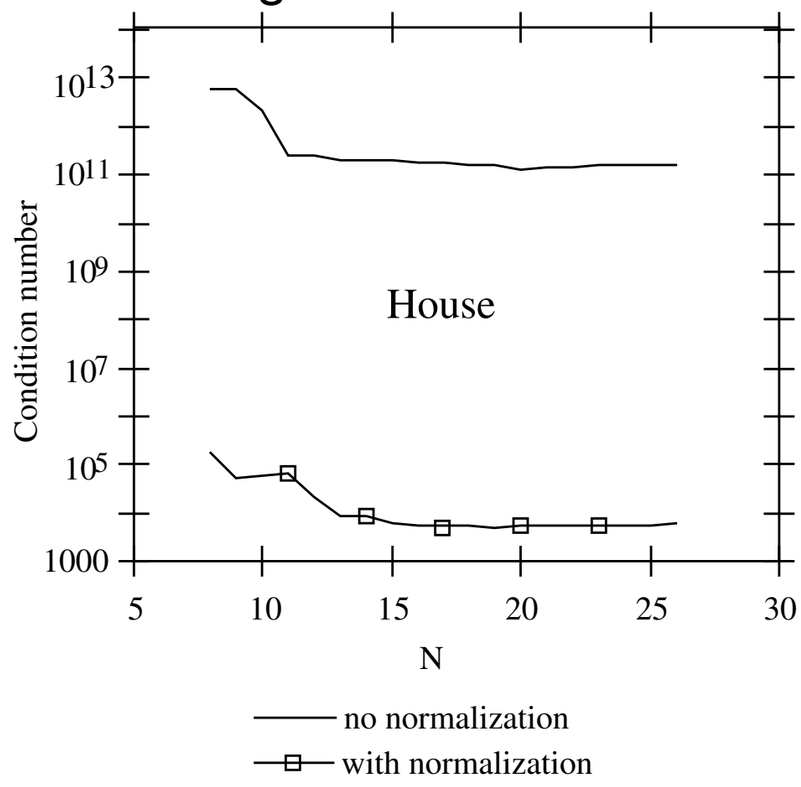
Unnormalized data



Normalized data

Condition number

- Bad condition number the reason for poor performance.
- Condition number $= \kappa_1/\kappa_8$, ratio of singular values of $\mathtt{A}$.



- Bad conditioning acts as a noise amplifier.
- Normalization improves the condition number by a factor of 10^8.
- **Reference** : Hartley ”In defence of the 8-point algorithm”.

Algebraic Minimization Algorithm

The algebraic minimization algorithm

Enforcing the singularity constraint

- SVD method minimizes $||\mathtt{F}' - \mathtt{F}||$.
- simple and rapid.
- Not optimal
- Treats all entries of $\mathtt{F}$ equally.
- However, some entries of $\mathtt{F}$ are more tightly constrained by the data.

The Algebraic Method

- Minimize $||\mathtt{A}\mathbf{f}'||$ subject to $||\mathbf{f}'|| = 1$ **AND** $\det \mathtt{F}' = 0$.
- $\det \mathtt{F}' = 0$ is a cubic constraint.
- Requires an iterative solution.
- However, simple iterative method works.

Solution assuming known epipole

- We may write $\mathtt{F} = \mathtt{M}[\mathbf{e}]_\times$, where $\mathbf{e}$ is epipole.
- $\mathtt{F}$ of this form is singular.
- Assume $\mathbf{e}$ is known, find $\mathtt{M}$.
- Write $\mathtt{F} = \mathtt{M}[\mathbf{e}]_\times$ as $\mathbf{f} = \mathbf{E}\mathbf{m}$

$$\begin{pmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{pmatrix} = \begin{bmatrix} 0 & -e_3 & e_2 & & & & & & \\ e_3 & 0 & -e_1 & & & & & & \\ -e_2 & e_1 & 0 & & & & & & \\ & & & 0 & -e_3 & e_2 & & & \\ & & & e_3 & 0 & -e_1 & & & \\ & & & -e_2 & e_1 & 0 & & & \\ & & & & & & 0 & -e_3 & e_2 \\ & & & & & & e_3 & 0 & -e_1 \\ & & & & & & -e_2 & e_1 & 0 \end{bmatrix} \begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{31} \\ m_{32} \\ m_{33} \end{pmatrix}$$

Solution assuming known epipole - continued

- Write $\mathtt{Af} = \mathtt{AEm}$.
- Minimize $||\mathtt{AEm}||$ subject to $||\mathtt{Em}|| = 1$.
- This is a linear least-squares estimation problem.
- Non-iterative algorithm involving SVD is possible
- Reference : Hartley, “Minimizing Algebraic Error”, Royal Society Proceedings, 1998.

Iterative Algebraic Estimation

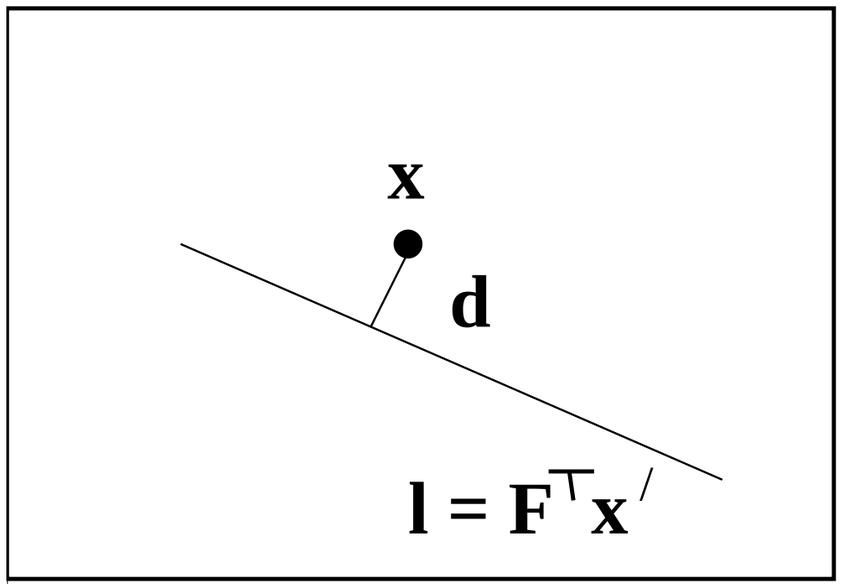
Find the fundamental matrix $\mathtt{F}$ that minimizes the algebraic error $||\mathbf{Af}||$ subject to $||\mathbf{f}|| = 1$ and $\det \mathtt{F} = 0$.

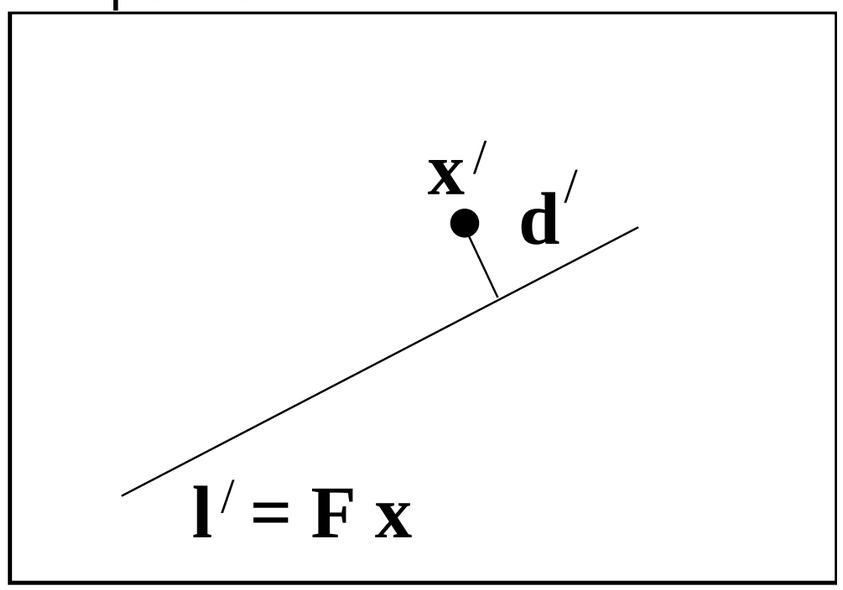
- **Concept :** Vary epipole $\mathbf{e}$ to minimize the algebraic error $||\mathbf{Af}'|| = ||\mathbf{AEm}||$.
- **Remark :** Each choice of epipole $\mathbf{e}$ defines a minimimum error vector $\mathbf{AEm}$ as above.
- Use Levenberg-Marquardt method to minimize this error.
- Simple 3×9 minimization problem.
  - 3 inputs – the coordinates of the epipole
  - 9 outputs – the algebraic error vector $\mathbf{Af}' = \mathbf{AEm}$.
- Each step requires estimation of $\mathbf{m}$ using SVD method.
- Tricks can be used to avoid SVD (see Hartley-Royal-Society).

Minimization of Geometric Error

Minimization of Geometric Error

- Algebraic error vector $\mathtt{A}\mathbf{f}$ has no clear geometric meaning.
- Should be minimizing geometric quantities.
- Errors derive from incorrect measurements of match points.





- We should be measuring distances from epipolar lines.

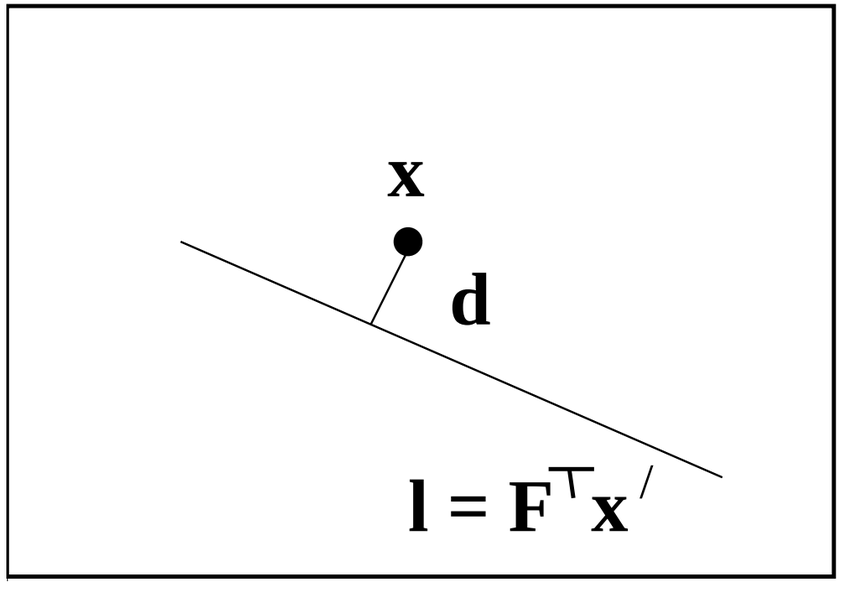
The Gold Standard (ML) Method

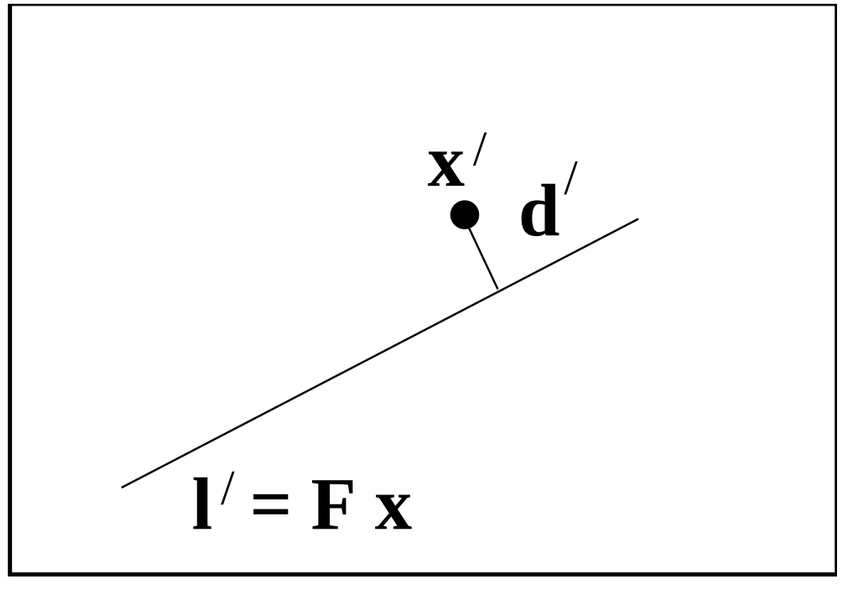
- Assumes a Gaussian distributed noise.
- Measured correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$.
- Estimated correspondences $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}'_i$

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

subject to $\hat{\mathbf{x}}_i'^{\top} \mathtt{F} \hat{\mathbf{x}}_i = 0$ exactly for some $\mathtt{F}$.

- Simultaneous estimation of $\mathtt{F}$ and $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}'_i$.





The Gold Standard (ML) Method

Minimizing the Gold-Standard error function.

- Initial 3D reconstruction :

$$\begin{aligned} \mathtt{P} &= [\mathtt{I} \mid \mathbf{0}] \\ \mathtt{P}' &= [\mathtt{M} \mid \mathbf{t}] \\ \mathbf{X}_i &= (\mathtt{X}_i, \mathtt{Y}_i, 1, \mathtt{T}_i)^\top \end{aligned}$$

- Compute $\hat{\mathbf{x}}_i = \mathtt{P}\mathbf{X}_i = (\mathtt{X}_i, \mathtt{Y}_i, 1)^\top$ and $\hat{\mathbf{x}}'_i = \mathtt{P}'\mathbf{X}_i$.
- Iterate over $\mathtt{P}'$ and $\mathbf{X}_i = (\mathtt{X}_i, \mathtt{Y}_i, 1, \mathtt{T}_i)^\top$ to minimize cost function :

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

- Total of $3n + 12$ parameters.
  - 12 parameters for the camera matrix $\mathtt{P}'$
  - 3 parameters for each point $\mathbf{X}_i$.
  - Once $\mathtt{P}' = [\mathtt{M} \mid \mathbf{t}]$ is found, compute $\mathtt{F} = [\mathbf{t}]_\times \mathtt{M}$.

Sparse Levenberg-Marquardt

Reference : Hartley- Azores

(i) Coordinates of $\mathbf{X}_i$ do not affect $\hat{\mathbf{x}}_j$ or $\hat{\mathbf{x}}'_j$ (for $i \neq j$).

(ii) Sparse LM takes advantage of sparseness

(iii) Linear time in n (number of points).

(iv) Reference : Hartley- Azores

Parametrization of rank-2 matrices

- Estimation of $\mathtt{F}$ may be done by parameter minimization.
- Parametrize $\mathtt{F}$ such that $\det \mathtt{F} = 0$.
- Various parametrizations have been used.

Overparametrization.

- Write $\mathtt{F} = [\mathbf{t}]_\times \mathtt{M}$.
- 3 parameters for $\mathbf{t}$ and 9 for $\mathtt{M}$.

Epipolar parametrization.

$$\mathtt{F} = \begin{bmatrix} a & b & \alpha a + \beta b \\ c & d & \alpha c + \beta d \\ e & f & \alpha e + \beta f \end{bmatrix}$$

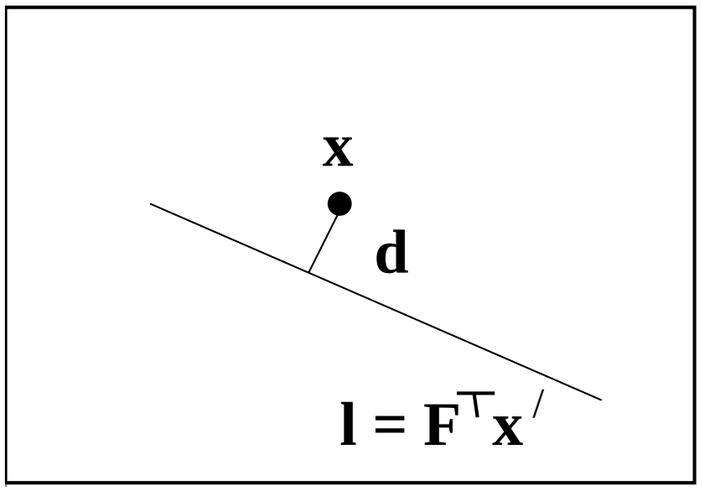
To achieve a minimum set of parameters, set one of the elements, for instance f to 1.

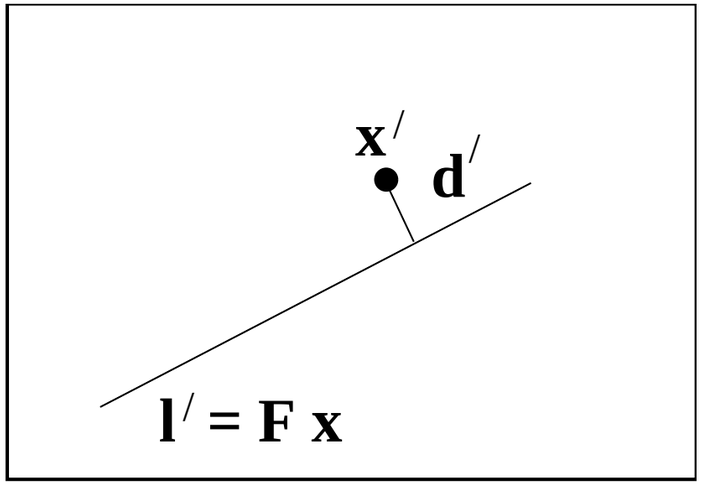
Parametrization of rank-2 matrices

Both epipoles as parameters. The resulting form of $\mathtt{F}$ is

$$\mathtt{F} = \begin{bmatrix} a & b & \alpha a + \beta b \\ c & d & \alpha c + \beta d \\ \alpha' a + \beta' c & \alpha' b + \beta' d & \alpha' \alpha a + \alpha' \beta b + \beta' \alpha c + \beta' \beta d \end{bmatrix}$$

Epipolar distance





- Point correspondence $\mathbf{x}' \leftrightarrow \mathbf{x}$:
- Point $\mathbf{x} \mapsto$ epipolar line $\mathtt{F}\mathbf{x}$.
- Epipolar distance is distance of point $\mathbf{x}'$ to epipolar line $\mathtt{F}\mathbf{x}$.
- Write $\mathtt{F}\mathbf{x} = (\lambda, \mu, \nu)^\top$ and $\mathbf{x}' = (x', y', 1)^\top$.
- Distance is

$$d(\mathbf{x}', \mathtt{F}\mathbf{x}) = \mathbf{x}'^\top \mathtt{F}\mathbf{x}(\lambda^2 + \mu^2)^{-1/2}$$

Epipolar distance - continued

- Epipolar distance may be written as

$$d(\mathbf{x}', \mathtt{F}\mathbf{x}) = \frac{\mathbf{x}'^{\top}\mathtt{F}\mathbf{x}}{((\mathtt{F}\mathbf{x})_1^2 + (\mathtt{F}\mathbf{x})_2^2)^{1/2}}$$

- Total cost function :

$$\sum_i d(\mathbf{x}'_i, \mathtt{F}\mathbf{x}_i)^2 = \sum_i \frac{(\mathbf{x}_i'^{\top}\mathtt{F}\mathbf{x}_i)^2}{(\mathtt{F}\mathbf{x}_i)_1^2 + (\mathtt{F}\mathbf{x}_i)_2^2}$$

- Total cost : sum over all $\mathbf{x}'_i \leftrightarrow \mathbf{x}_i$.
- Minimize this cost function over parametrization of $\mathtt{F}$.

Symmetric epipolar distance

- Epipolar distance function is not symmetric.
- Prefer sum of distances in both images.
- Symmetric cost function is

$$d(\mathbf{x}', \mathtt{F}\mathbf{x})^2 + d(\mathbf{x}, \mathtt{F}^\top \mathbf{x}')^2$$

- Sum over all points :

$$\text{Cost} = \sum_i (\mathbf{x}_i'^\top \mathtt{F}\mathbf{x}_i)^2 \left(\frac{1}{(\mathtt{F}\mathbf{x}_i)_1^2 + (\mathtt{F}\mathbf{x}_i)_2^2} + \frac{1}{(\mathtt{F}^\top \mathbf{x}_i')_1^2 + (\mathtt{F}^\top \mathbf{x}_i')_2^2} \right)$$

Problem

- Points near the epipole have a disproportionate influence.
- Small deviation in point makes big difference to epipolar line.

Luong / Zhang's other error function

$$\sum_i \frac{(\mathbf{x}_i'^{\top}\mathtt{F}\mathbf{x}_i)^2}{(\mathtt{F}\mathbf{x}_i)_1^2 + (\mathtt{F}\mathbf{x}_i)_2^2 + (\mathtt{F}^{\top}\mathbf{x}_i')_1^2 + (\mathtt{F}^{\top}F\mathbf{x}_i')_2^2}$$

Represents a first-order approximation to geometric error.

More Algorithm Comparison

Experimental Evaluation of the Algorithms

Three of the algorithms compared

(i) The normalized 8-point algorithm

(ii) Minimization of algebraic error whilst imposing the singularity constraint

(iii) The Gold Standard geometric algorithm

Experimental procedure

(i) Find matched points in image pair.

(ii) Select n matched points at random

(iii) Compute the fundamental matrix

(iv) Compute epipolar distance for all other points.

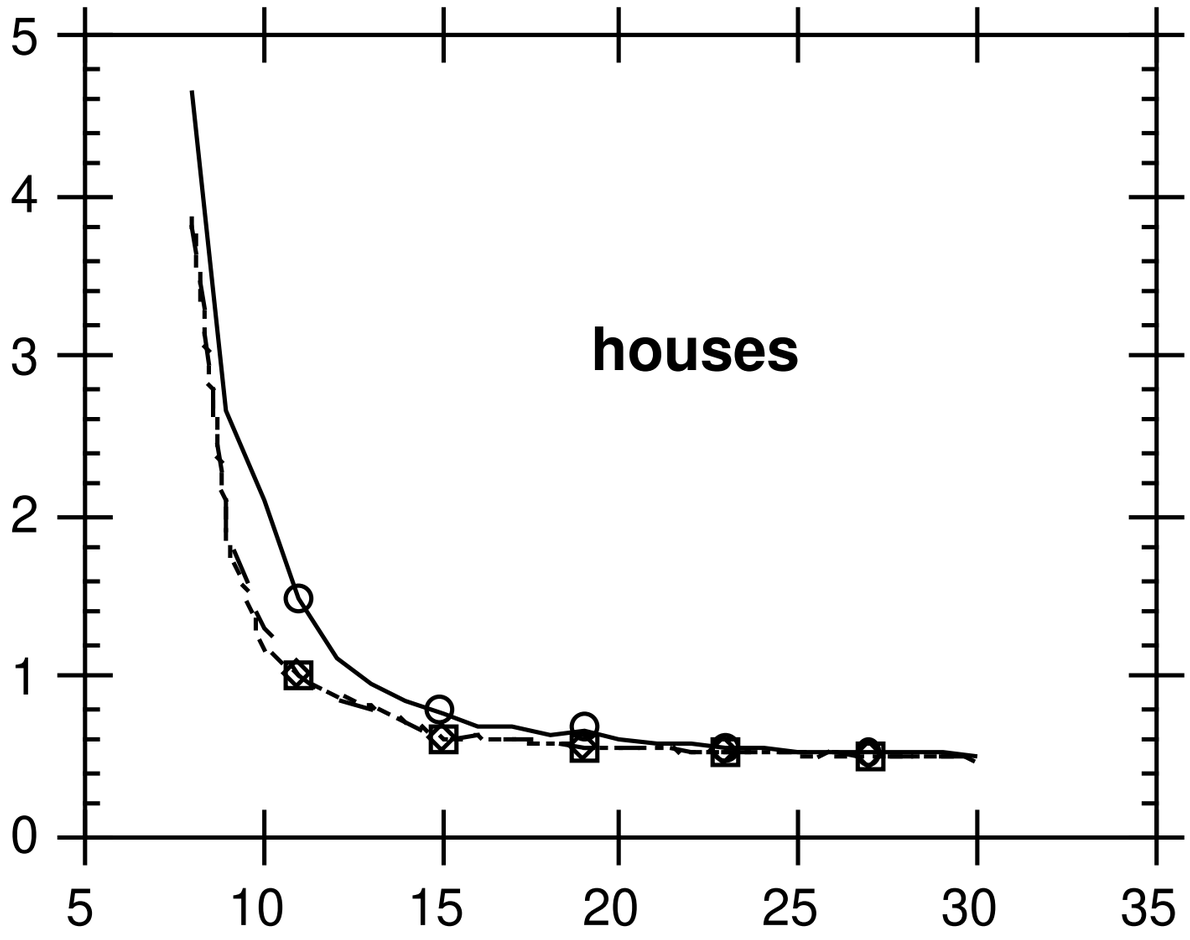
(v) Repeat 100 times for each n and collect statistics.

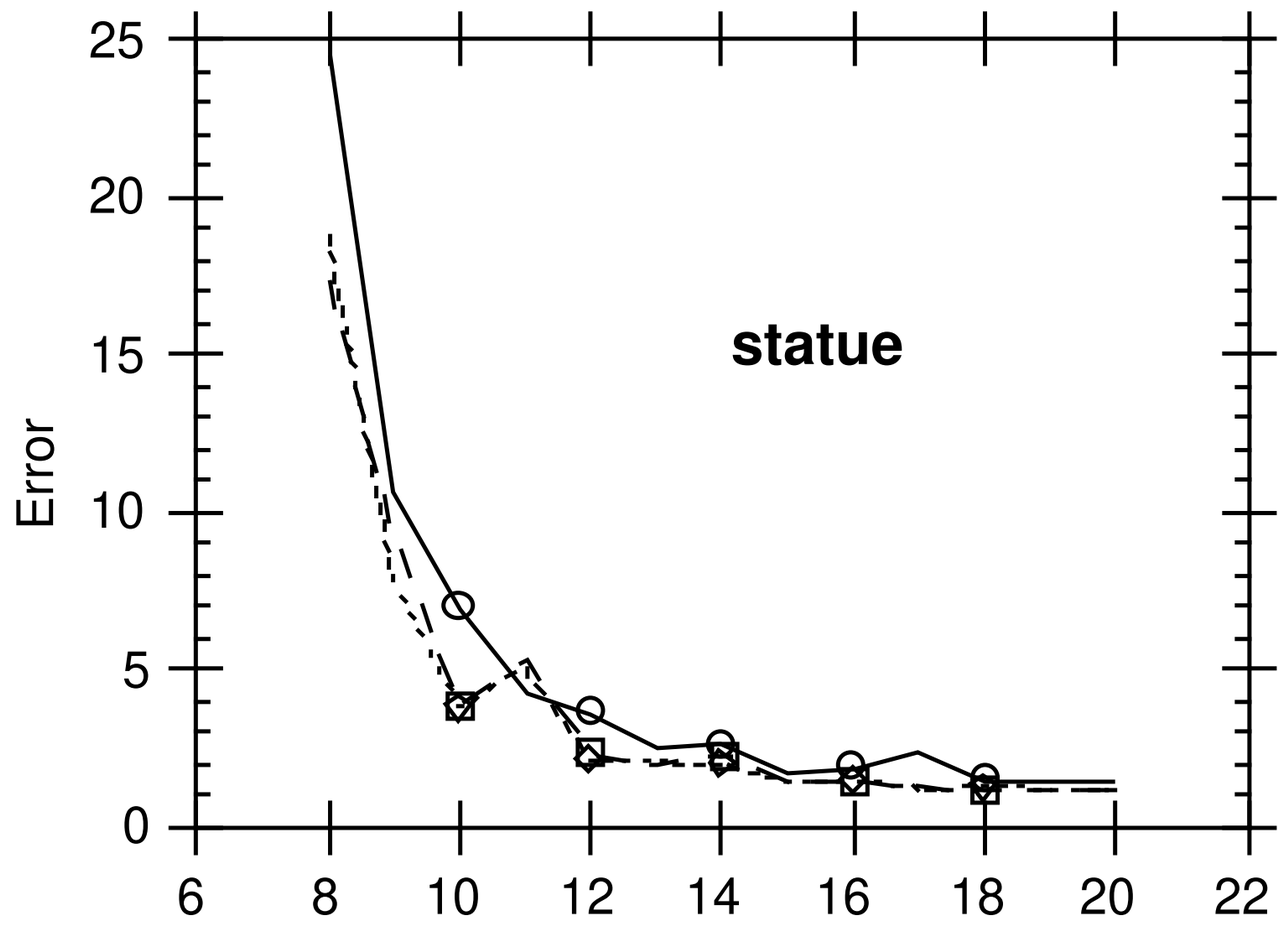
The error is defined as

$$\frac{1}{N}\sum^{N}_{i}(d(\mathbf{x}'_i, \mathtt{F}\mathbf{x}_i) + d(\mathbf{x}_i, \mathtt{F}^\top\mathbf{x}'_i))$$

i.e Average symmetric epipolar distance.

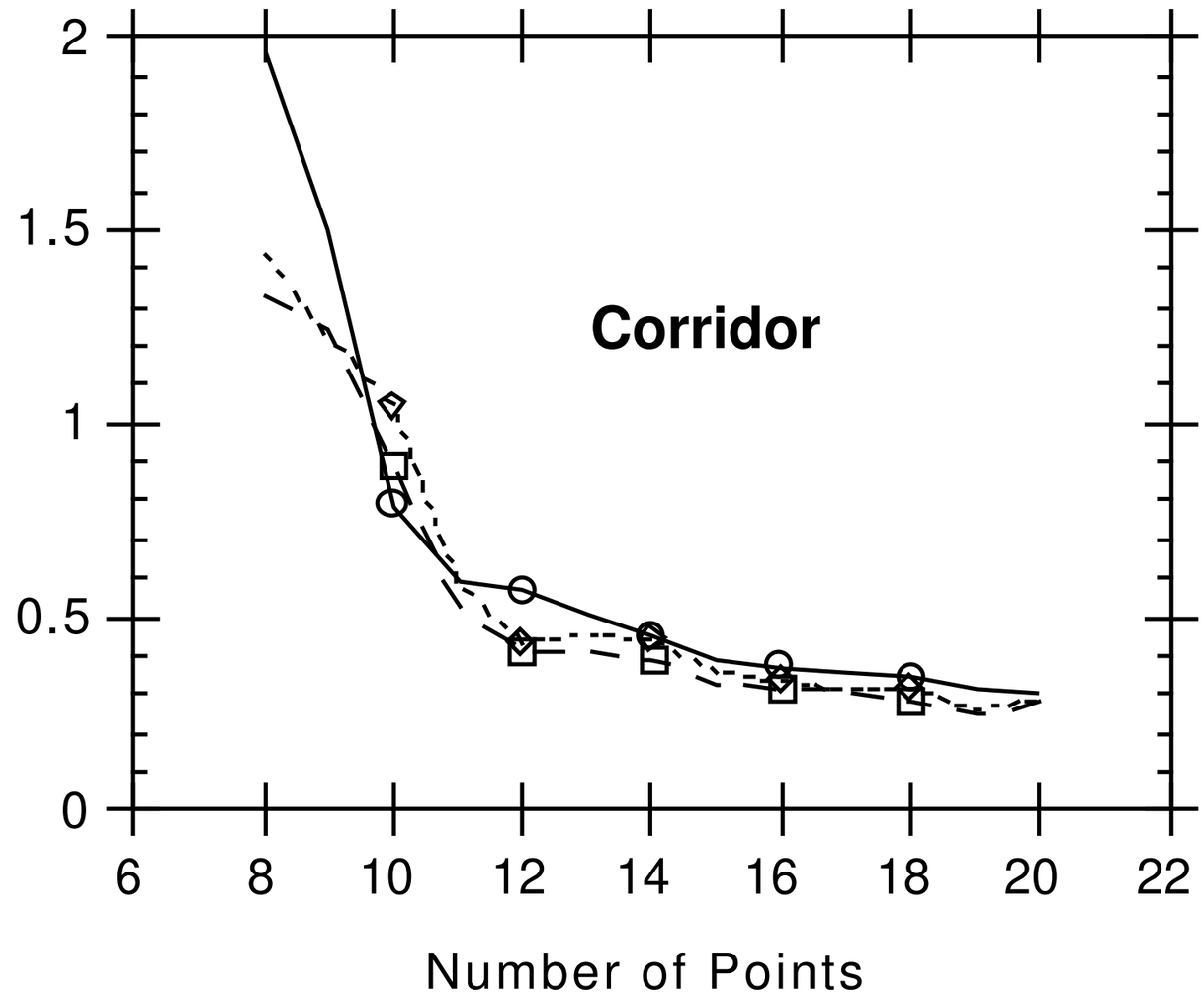
Results

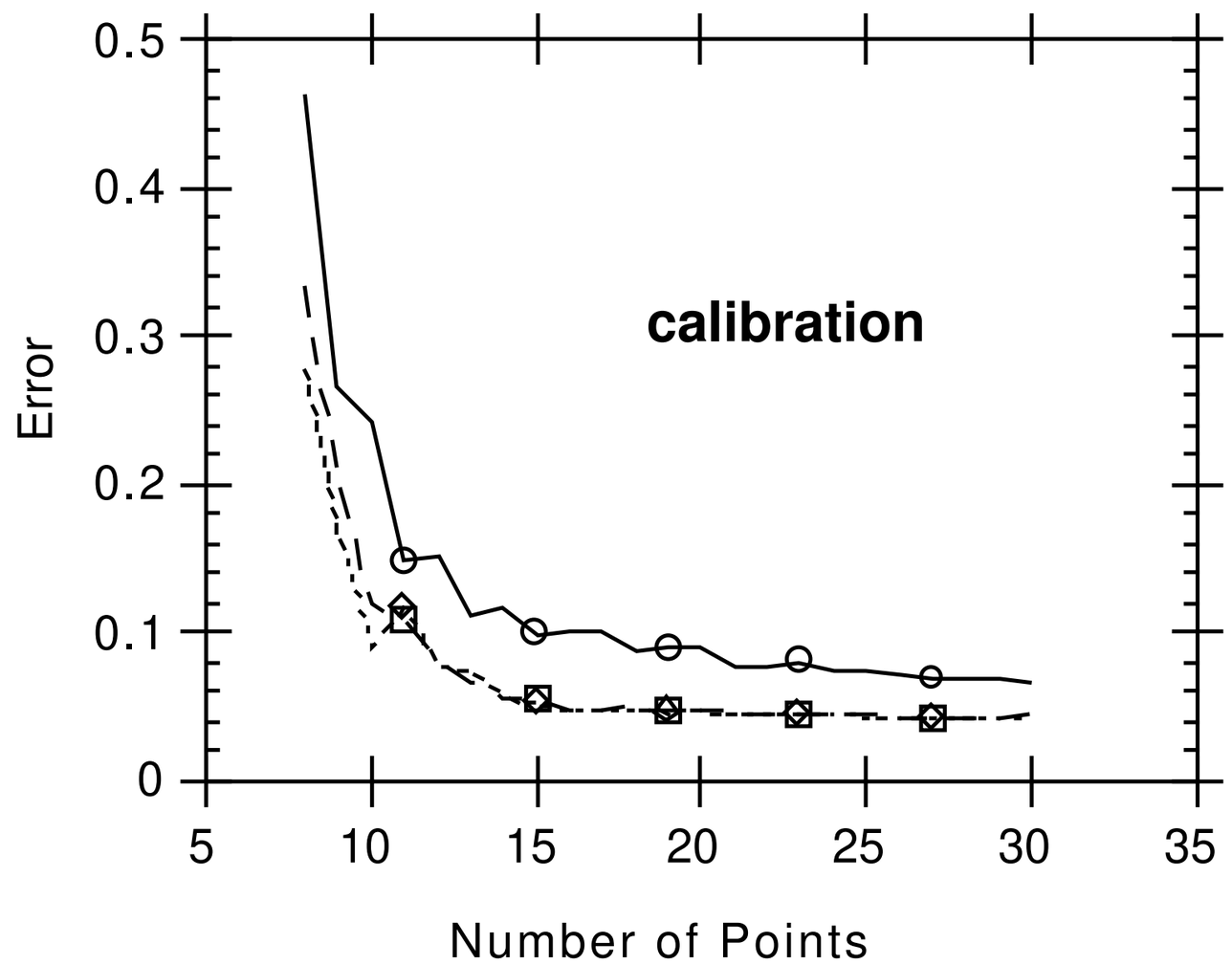




Normalized 8-point, geometric and algebraic error.

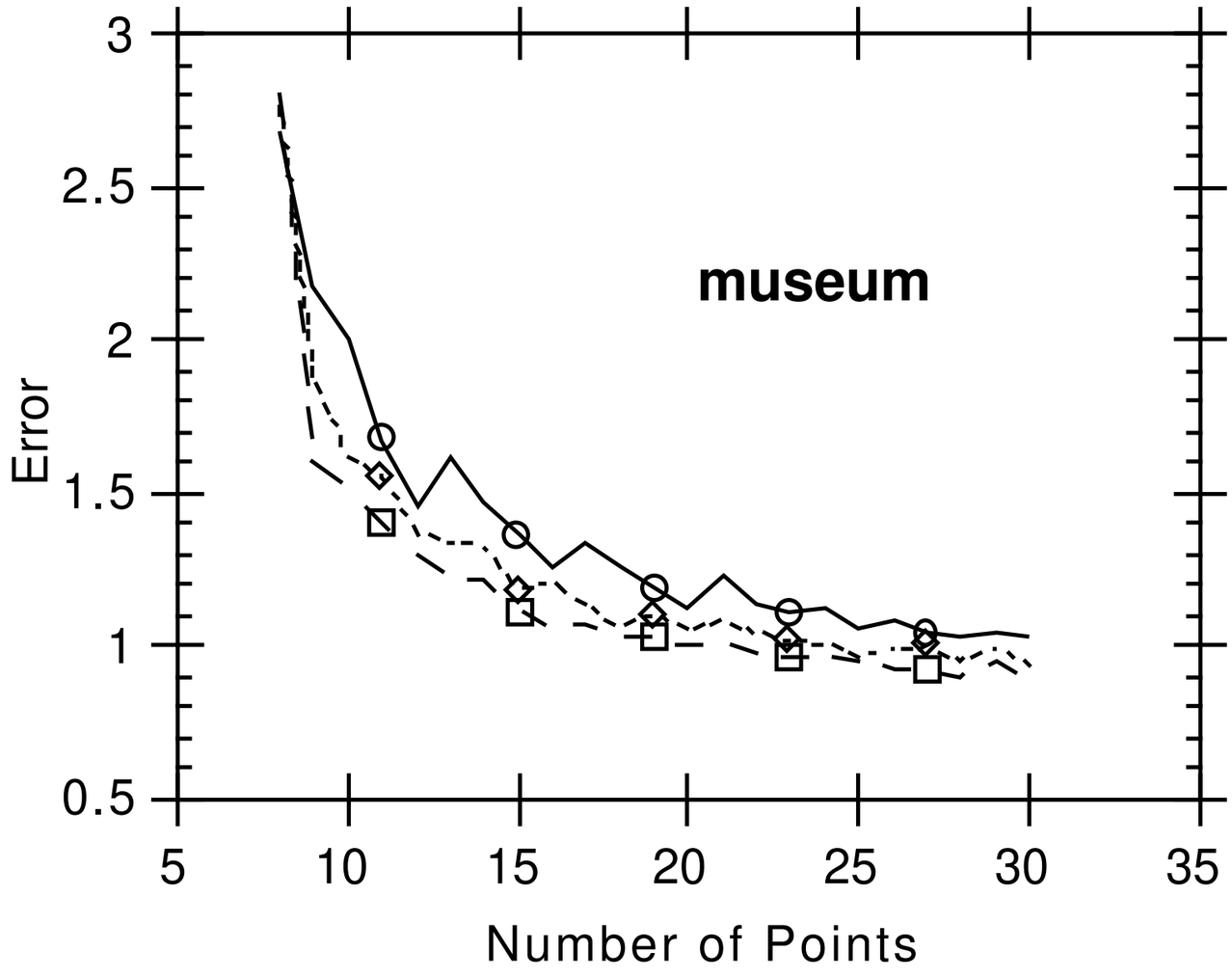
Results - continued





Normalized 8-point, geometric and algebraic error.

Results - continued



Normalized 8-point, geometric and algebraic error.

Covariance computation

Covariance of $\mathtt{P}'$

To compute the covariance matrix of the entries of $\mathbf{P}'$:

(i) Define $\mathbf{Q} = (\mathbf{x}_i, \mathbf{x}'_i)^\top$ – vector of meaurments in both images.

(ii) Compute the derivative matrices

$$\mathtt{A}_i = [\partial \hat{\mathbf{Q}}_i / \partial \mathbf{P}'] \text{ and } \mathtt{B}_i = [\partial \hat{\mathbf{Q}}_i / \partial \mathbf{X}_i]$$

(iii) Compute in steps

$$\begin{aligned}
\mathtt{U} &= \sum_i \mathtt{A}_i{}^\top \Sigma_{\mathbf{Q}_i}^{-1} \mathtt{A}_i \\
\mathtt{V}_i &= \mathtt{B}_i{}^\top \Sigma_{\mathbf{Q}_i}^{-1} \mathtt{B}_i \\
\mathtt{W}_i &= \mathtt{A}_i{}^\top \Sigma_{\mathbf{Q}_i}^{-1} \mathtt{B}_i \\
\Sigma_{\mathbf{P}'} &= (\mathtt{U} - \sum_i \mathtt{W}_i \mathtt{V}_i^{-1} \mathtt{W}_i{}^\top)^+ \text{ (pseudo-inverse)}
\end{aligned}$$

Covariance of $\mathtt{F}$

To compute the covariance of $\mathtt{F}$:

(i) Given $\mathtt{P}' = [\mathtt{M} \mid \mathbf{m}]$, then $\mathtt{F} = [\mathbf{m}]_\times \mathtt{M}$.

(ii) Compute $\mathtt{J} = \partial \mathtt{F} / \partial \mathtt{P}'$.

(iii)

$$\Sigma_{\mathtt{F}} = \mathtt{J} \Sigma_{\mathtt{P}'} \mathtt{J}^\top$$

Covariance of epipolar line corresponding to $\mathbf{x}$

(i) Epipolar line is $\mathbf{l} = \mathtt{F}\mathbf{x}$.

(ii) Given $\mathbf{x}$ and $\Sigma_{\mathtt{F}}$ compute $\mathtt{J} = \partial \mathbf{l} / \partial \mathtt{F}$

(iii)

$$\Sigma_{\mathbf{l}} = \mathtt{J} \Sigma_{\mathtt{F}} \mathtt{J}^\top$$

The envelope of epipolar lines

- May compute envelope of epipolar lines.

$$\mathtt{C} = \bar{\mathbf{l}}\,\bar{\mathbf{l}}^{\top} - k^2 \Sigma_{\mathbf{l}}$$

- $\mathtt{C}$ is a hyperbola that contains the epipolar line with a given probability α.
- k^2 chosen such that $F_2^{-1}(k^2) = \alpha$,
- $F_2(k^2)$ represents the cumulative χ^2_2 distribution,
- with probability α the lines lie within this region.

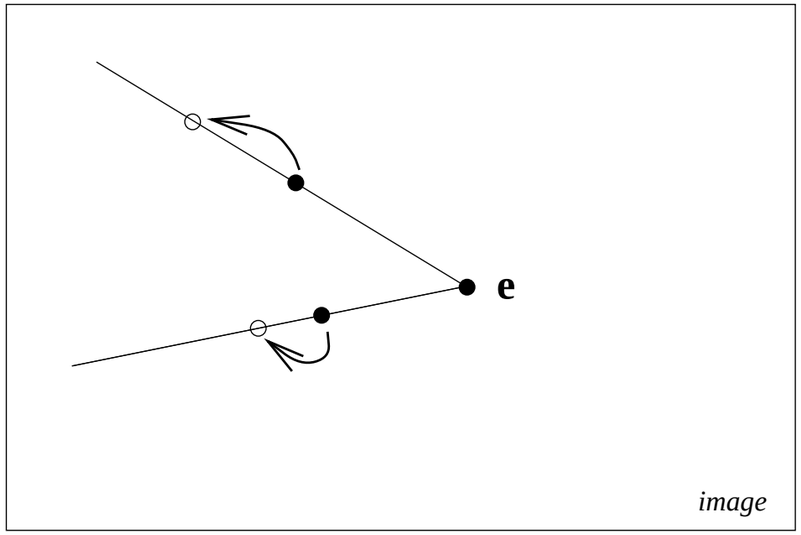
< epipolar line demonstration here >

Special cases of F-computation

Using special motions can simplify the computation of the fundamental matrix.

Pure translation

- Can assume $\mathtt{P} = [\mathtt{I}|\mathbf{0}]$ and $\mathtt{P}' = [\mathtt{I}|\mathbf{t}]$.
- $\mathtt{F} = [\mathbf{t}]_\times$.
- $\mathtt{F}$ is skew-symmetric – has 2 dof.
- Being skew-symmetric, automatically has rank 2.



For a pure translation the epipole can be estimated from the image motion of two points.

Cameras with the same principal plane

- Principal plane of the camera is the third row of $\mathtt{P}$.
- Cameras have the same third row.
- Affine cameras - last row is $(0, 0, 0, 1)^\top$.

Simple correspondences exist :

$$(x', y', 0)\mathtt{F}(x, y, 0)^\top = 0$$

for any $(x', y', 0)^\top$ and $(x, y, 0)^\top$.

$\mathtt{F}$ has the following form :

$$\mathtt{F} = \begin{bmatrix} & & a \\ & & b \\ c & d & e \end{bmatrix}$$

Degeneracies

Correspondences are degenerate if they satisfy more than one $\mathtt{F}$.

$$\mathbf{x}_i{}^{\top}\mathtt{F}_1\mathbf{x}_i{}^{\top} = 0 \quad \text{and} \quad \mathbf{x}_i'\mathtt{F}_2\mathbf{x}_i = 0 \qquad\qquad (1 \leq i \leq n) \ .$$

Points on a ruled quadric

(i) If all the points and the two camera centres lie on a ruled quadric, then there are three possible fundamental matrices.

(ii) points lie in a plane. The correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ lead to a 3-parameter family of possible fundamental matrices $\mathtt{F}$ (note, one of the parameters accounts for scaling the matrix so there is only a *two*-parameter family of homogeneous matrices).

(iii) Two cameras at the same point :

- The fundamental matrix does not exist.
- There is no such thing as an epipolar plane, and epipolar lines are not defined.
- Correspondences $\mathbf{x}'_i \leftrightarrow \mathbf{x}_i$ give at least a 2-parameter family of $\mathtt{F}$.

Automatic Estimation of Epipolar Geometry

Problem Statement

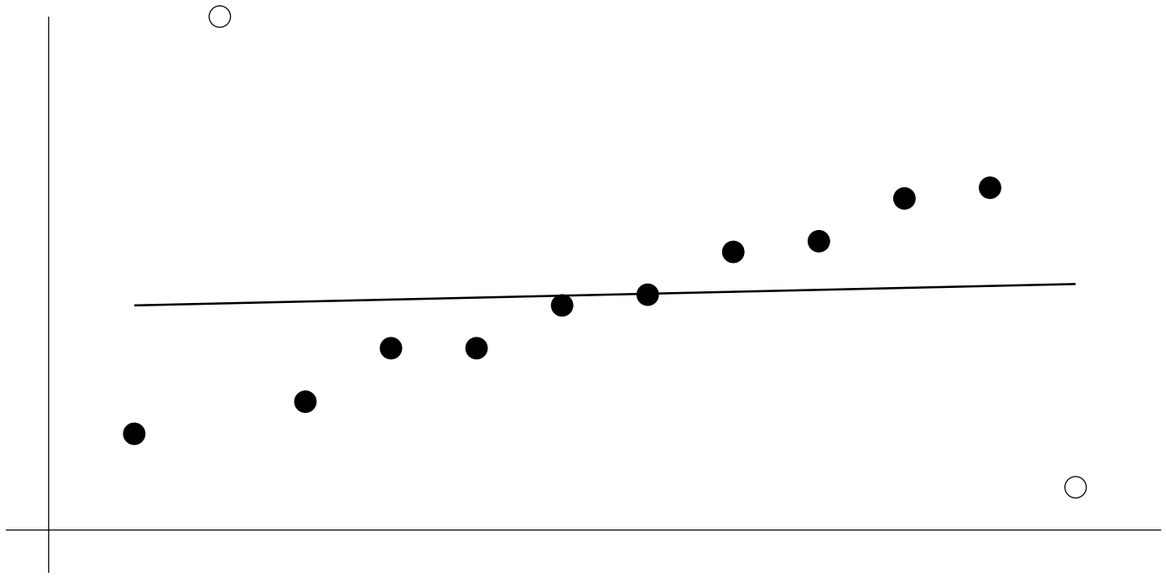
Given Image pair

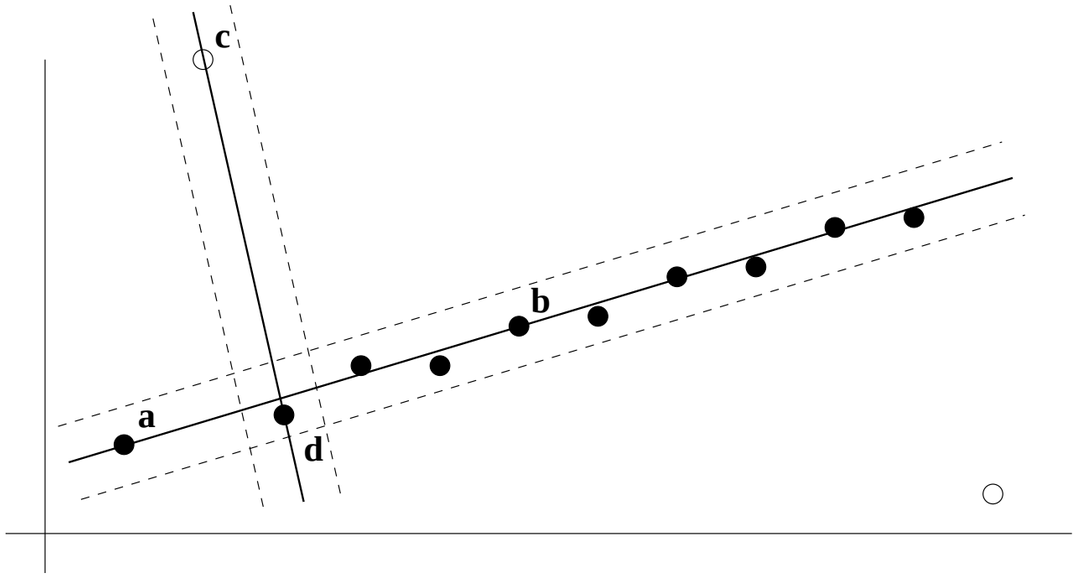
Find The fundamental matrix $\mathtt{F}$ and correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$.

- Compute image points
- Compute correspondences
- Compute epipolar geometry

Robust line estimation

Fit a line to 2D data containing outliers





There are two problems:

(i) a line fit to the data $\min_{\mathbf{l}} \sum_i d_{\perp i}^2$; and,

(ii) a classification of the data into inliers (valid points) and outliers.

RANdom SAmple Consensus (RANSAC)

[Fischler and Bolles, 1981]

Objective Robust fit of a model to a data set S which contains outliers.

Algorithm

(i) Randomly select a sample of s data points from S and instantiate the model from this subset.

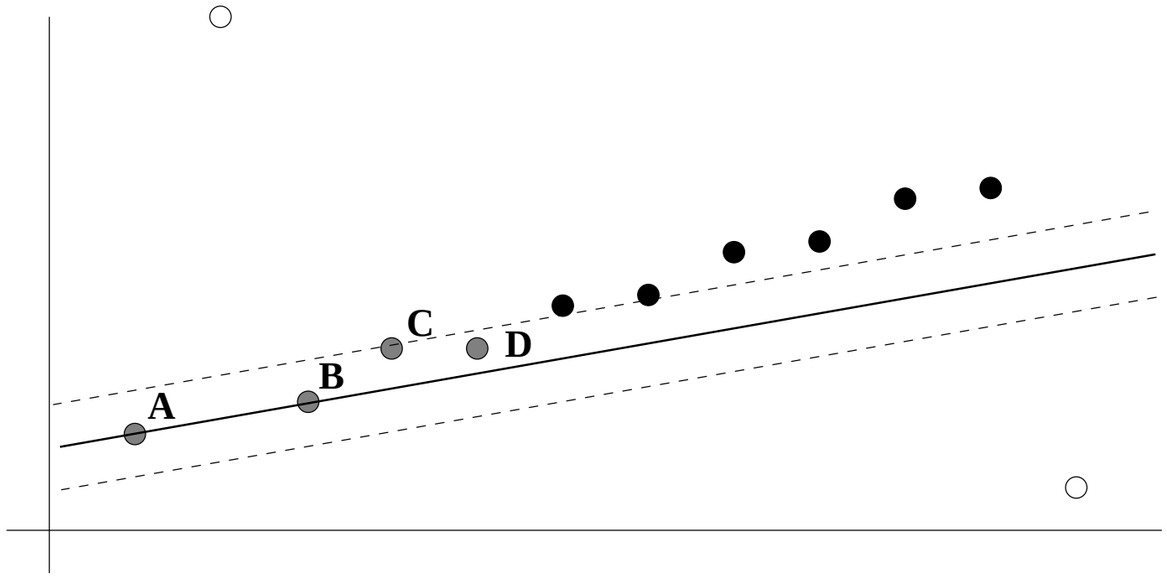
(ii) Determine the set of data points S_i which are within a distance threshold t of the model. The set S_i is the consensus set of the sample and defines the inliers of S.

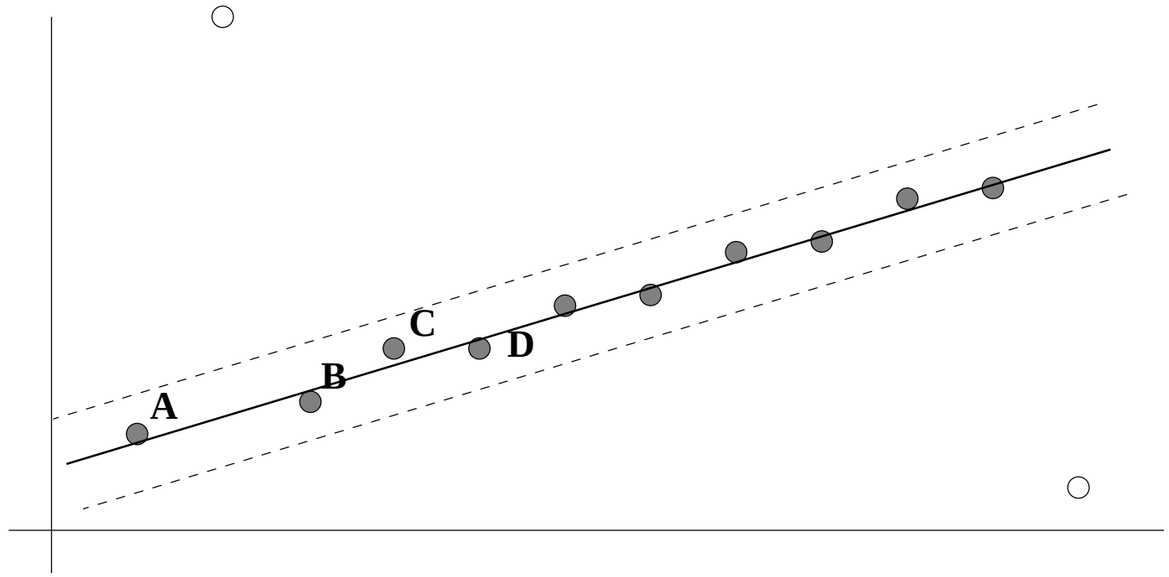
(iii) If the size of S_i (the number of inliers) is greater than some threshold T, re-estimate the model using all the points in S_i and terminate.

(iv) If the size of S_i is less than T, select a new subset and repeat the above.

(v) After N trials the largest consensus set S_i is selected, and the model is re-estimated using all the points in the subset S_i.

Robust ML estimation





An improved fit by

- A better minimal set
- Robust MLE: instead of $\min_{\mathbf{l}} \sum_i d_{\perp i}^2$

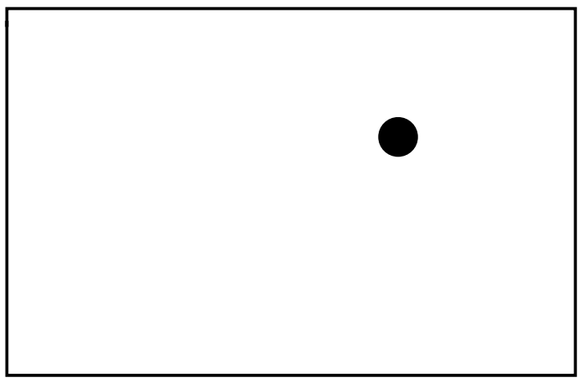
$$\min_{\mathbf{l}} \sum_i \gamma\left(d_{\perp i}\right) \qquad \text{with} \;\; \gamma(e) = \begin{cases} e^2 & e^2 < t^2 \;\; \text{inlier} \\ t^2 & e^2 \geq t^2 \;\; \text{outlier} \end{cases}$$

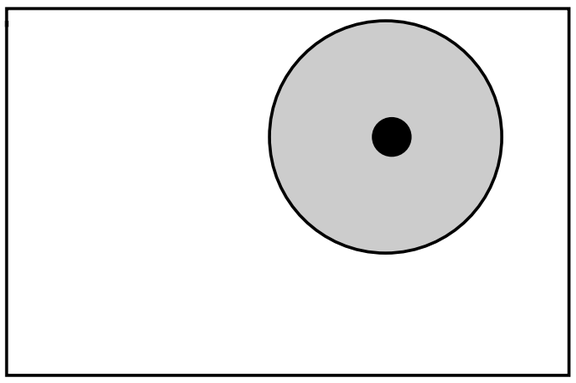
Feature extraction: "Corner detection"

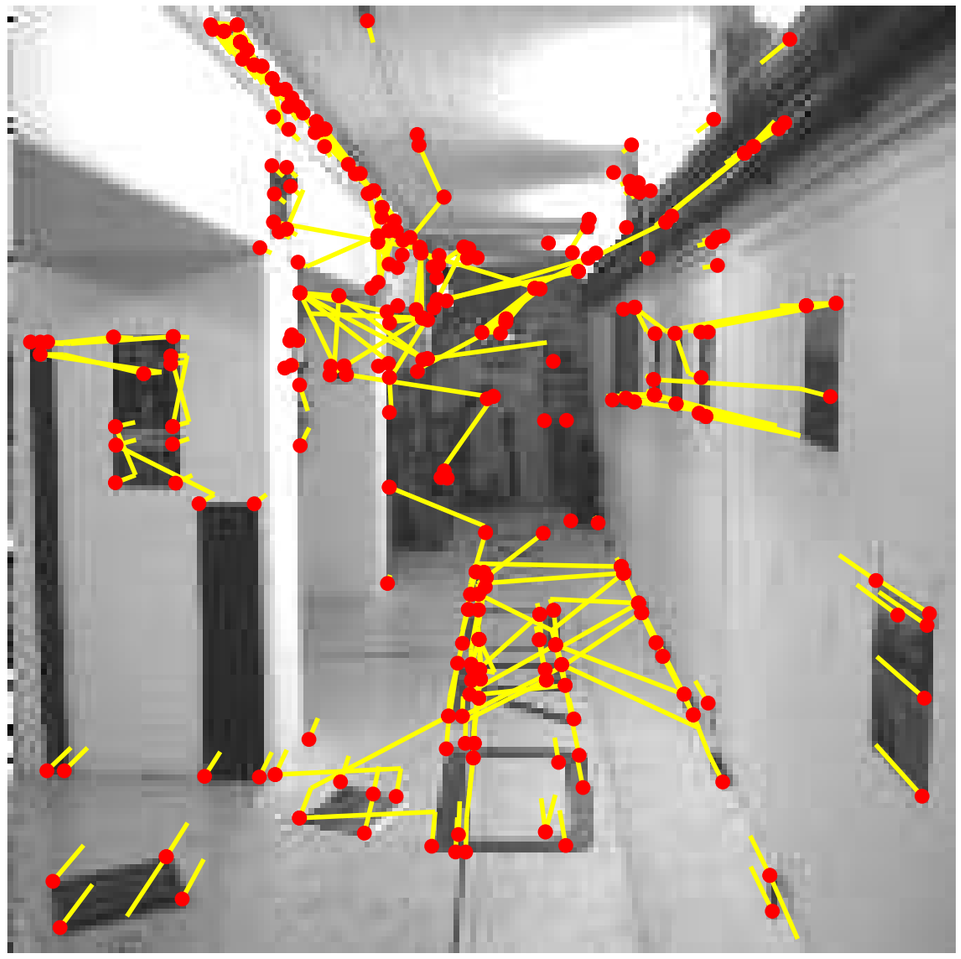
Interest points [Harris]

- 100s of points per image

Correlation matching

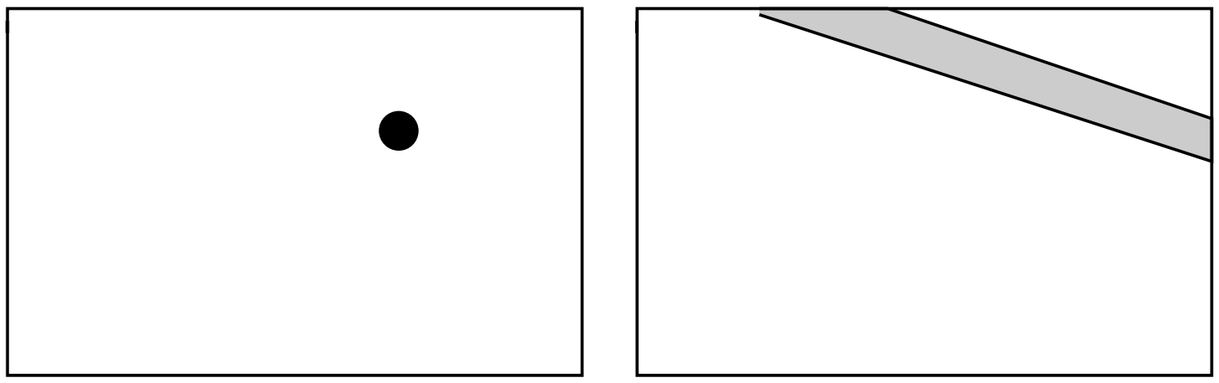


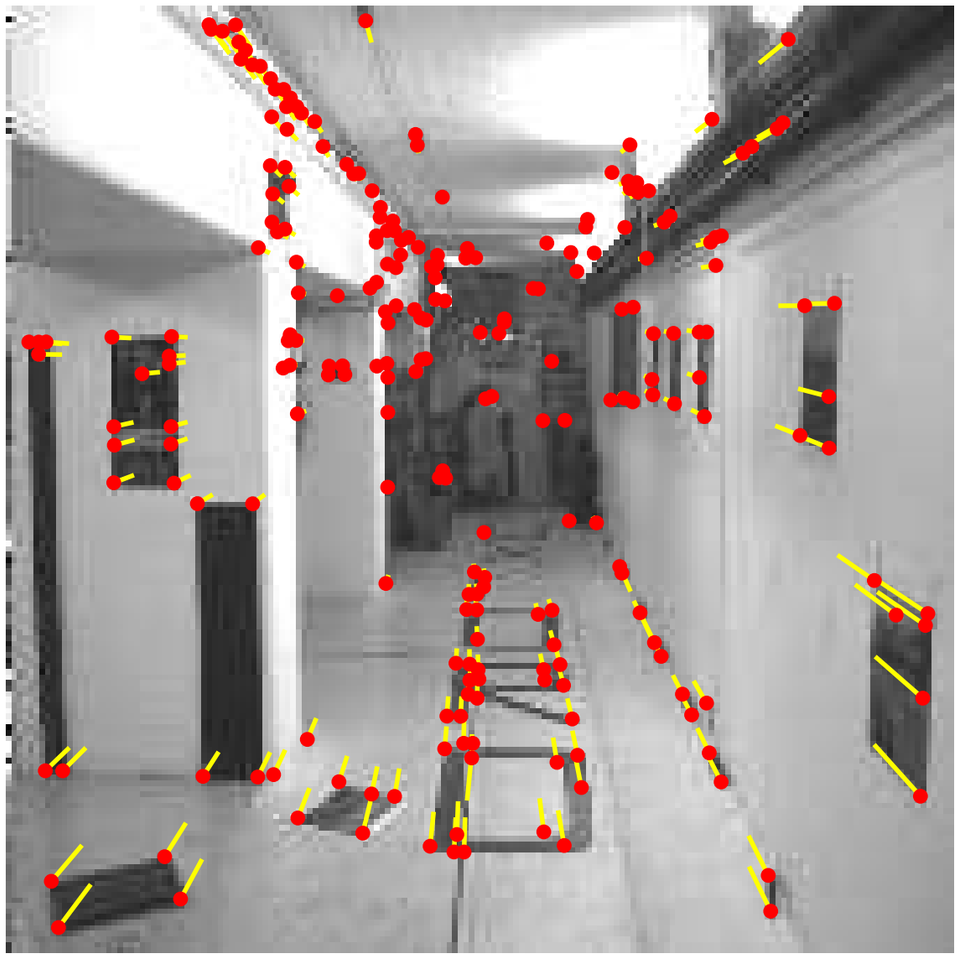




- Match each corner to most similar looking corner in the other image
- Many wrong matches (10-50%), but enough to compute the fundamental matrix.

Correspondences consistent with epipolar geometry





- Use RANSAC robust estimation algorithm
- Obtain correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ and $\mathtt{F}$
- Guided matching by epipolar line
- Typically: final number of matches is about 200-250, with distance error of $\sim$0.2 pixels.

Automatic Estimation of $\mathtt{F}$ and correspondences

Algorithm based on RANSAC [Torr]

(i) Interest points: Compute interest points in each image.

(ii) Putative correspondences: use cross-correlation and proximity.

(iii) RANSAC robust estimation:

Repeat

(a) Select random sample of 7 correspondences

(b) Compute $\mathtt{F}$

(c) Measure support (number of inliers)

Choose the $\mathtt{F}$ with the largest number of inliers.

(iv) MLE: re-estimate $\mathtt{F}$ from inlier correspondences.

(v) Guided matching: generate additional matches.

How many samples?

For probability p of no outliers:

$$N = \log(1-p)/\log(1-(1-\epsilon)^s)$$

- N, number of samples
- s, size of sample set
- ϵ, proportion of outliers

e.g. for $p = 0.95$

Sample size	Proportion of outliers ϵ						
s	5%	10%	20%	25%	30%	40%	50%
2	2	2	3	4	5	7	11
3	2	3	5	6	8	13	23
4	2	3	6	8	11	22	47
5	3	4	8	12	17	38	95
6	3	4	10	16	24	63	191
7	3	5	13	21	35	106	382
8	3	6	17	29	51	177	766

Adaptive RANSAC

- $N = \infty$, sample_count$= 0$.
- While $N >$ sample_count Repeat
  - Choose a sample and count the number of inliers.
  - Set $\epsilon = 1 - (\text{number of inliers})/(\text{total number of points})$
  - Set N from ϵ with $p = 0.99$.
  - Increment the sample_count by one.
- Terminate.

e.g. for a sample size of 4

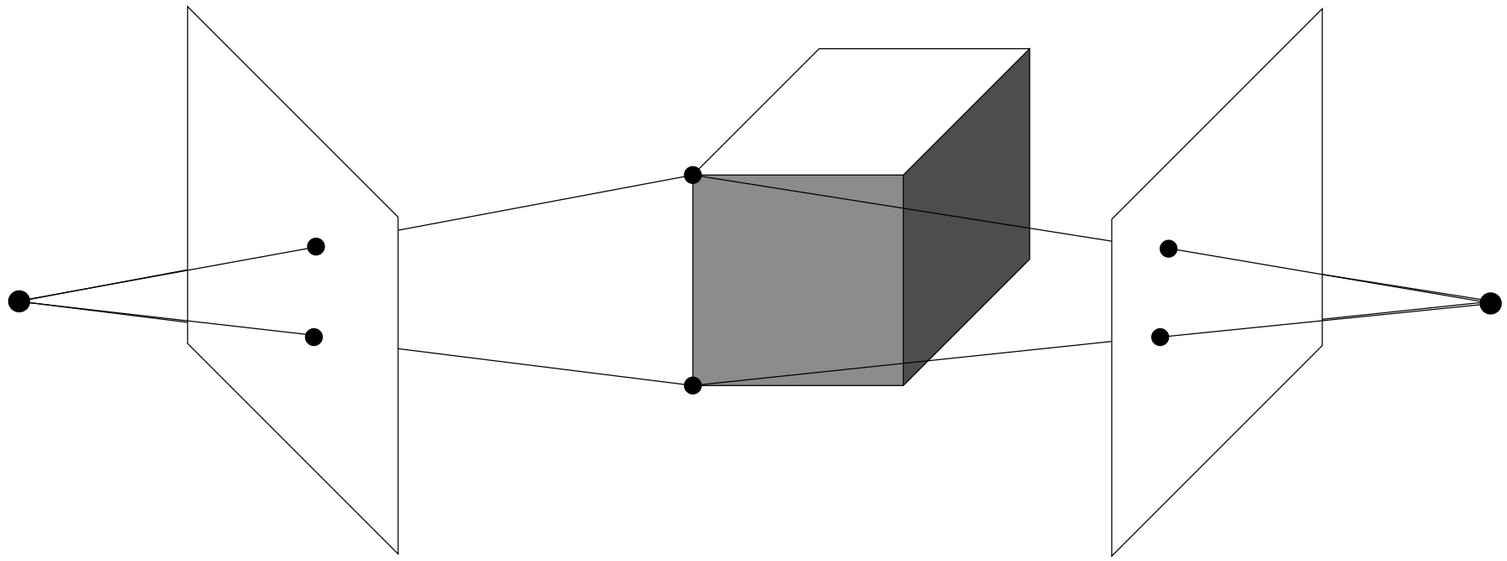
Number of inliers	1 - ϵ	Adaptive N
6	2%	20028244
10	3%	2595658
44	16%	6922
58	21%	2291
73	26%	911
151	56%	43

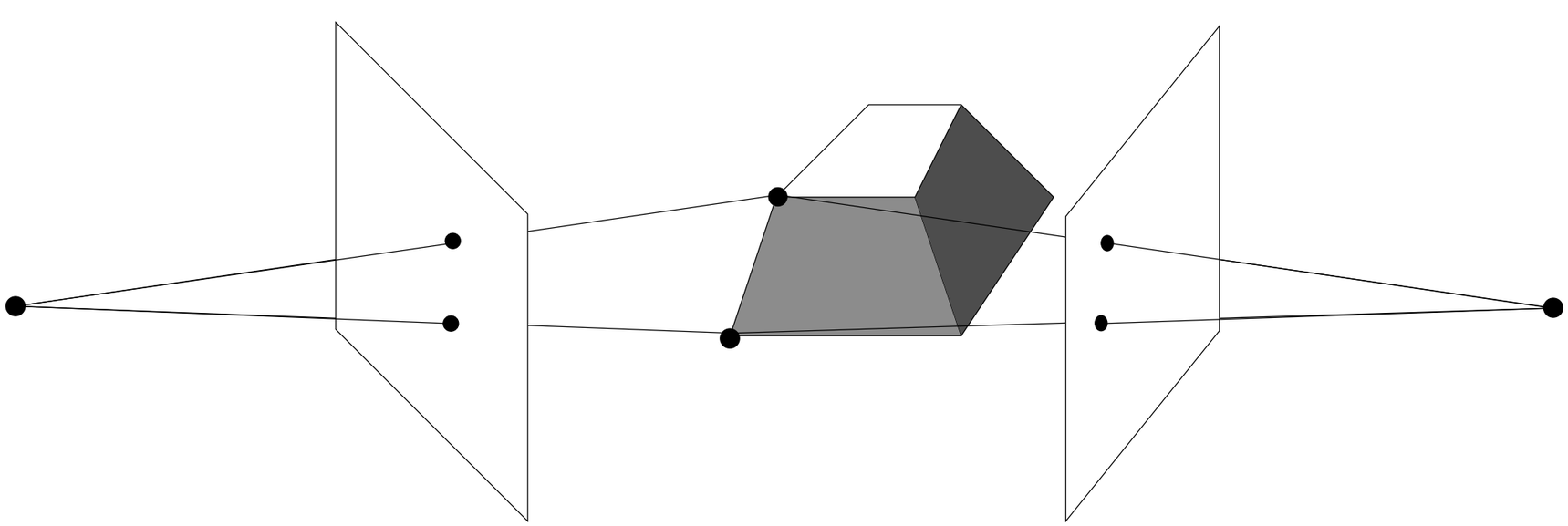
Part 2 : Three-view and Multiple-view Geometry

Computing a Metric Reconstruction

Reconstruction from two views

Given only image points and their correspondence,
what can be determined?





Two View Reconstruction Ambiguity

Given: image point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$,
compute a reconstruction:

$$\{\mathtt{P},\ \mathtt{P}',\ \mathbf{X}_i\} \quad \text{with} \quad \mathbf{x}_i = \mathtt{P}\mathbf{X}_i \quad \mathbf{x}'_i = \mathtt{P}'\mathbf{X}_i$$

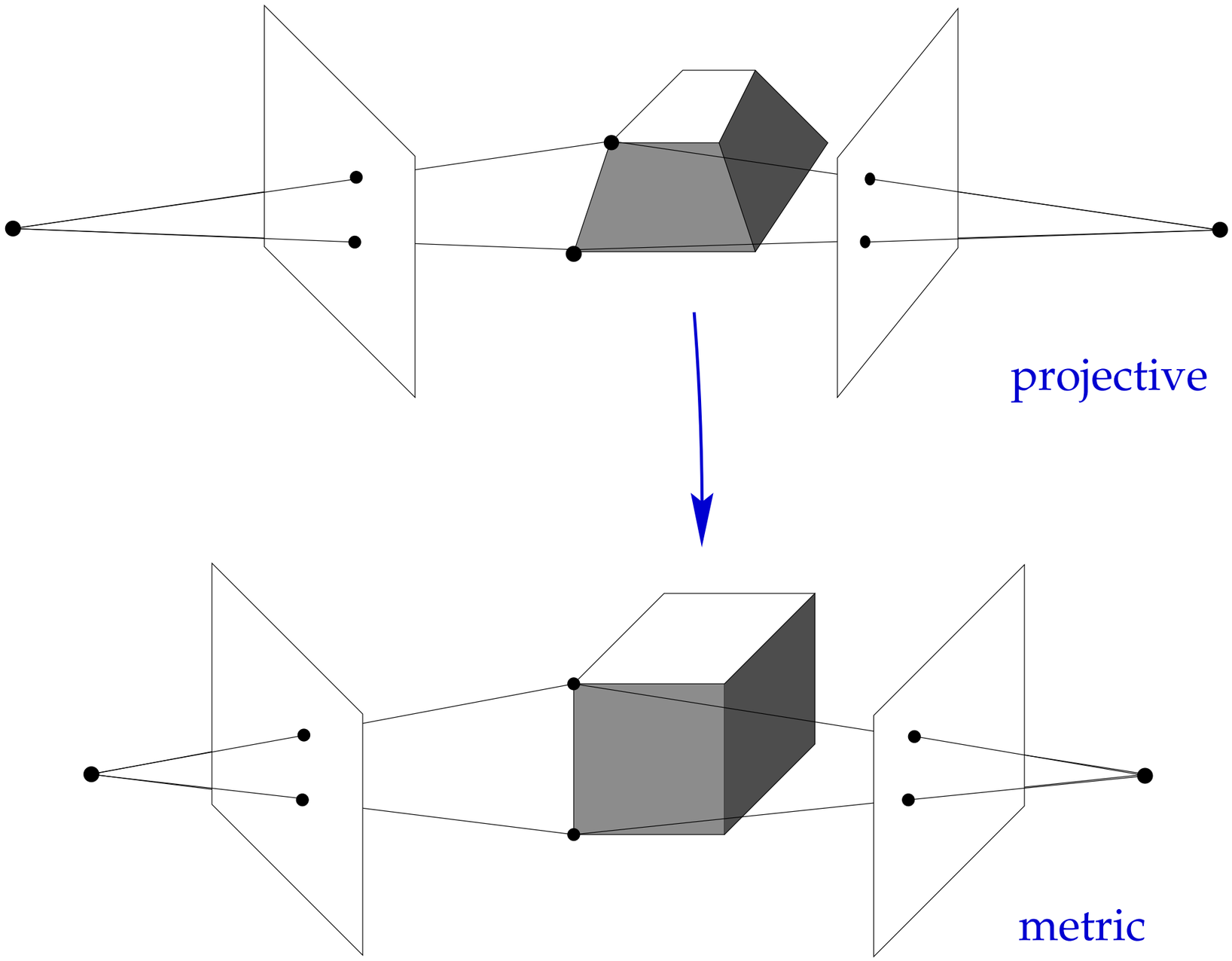
Ambiguity

$$\mathbf{x}_i \ = \ \mathtt{P}\mathbf{X}_i = \mathtt{P}\ \mathtt{H}(\mathtt{H})^{-1}\ \mathbf{X}_i = \tilde{\mathtt{P}}\tilde{\mathbf{X}}_i$$

$$\mathbf{x}'_i \ = \ \mathtt{P}'\mathbf{X}_i = \mathtt{P}'\ \mathtt{H}(\mathtt{H})^{-1}\ \mathbf{X}_i = \tilde{\mathtt{P}}'\tilde{\mathbf{X}}_i$$

$\{\tilde{\mathtt{P}},\ \tilde{\mathtt{P}}',\ \tilde{\mathbf{X}}_i\}$ is an equivalent Projective Reconstruction.

Metric Reconstruction



Correct: angles, length ratios.

Algebraic Representation of Metric Reconstruction

Compute $\mathtt{H}$

$$\{\mathtt{P}^1,\ \mathtt{P}^2, ..., \mathtt{P}^m,\ \mathbf{X}_i\} \quad \xrightarrow[\mathtt{H}]{} \quad \{\mathtt{P}^1_{\mathrm{M}},\ \mathtt{P}^2_{\mathrm{M}}, ..., \mathtt{P}^m_{\mathrm{M}},\ \mathbf{X}^{\mathrm{M}}_i\}$$

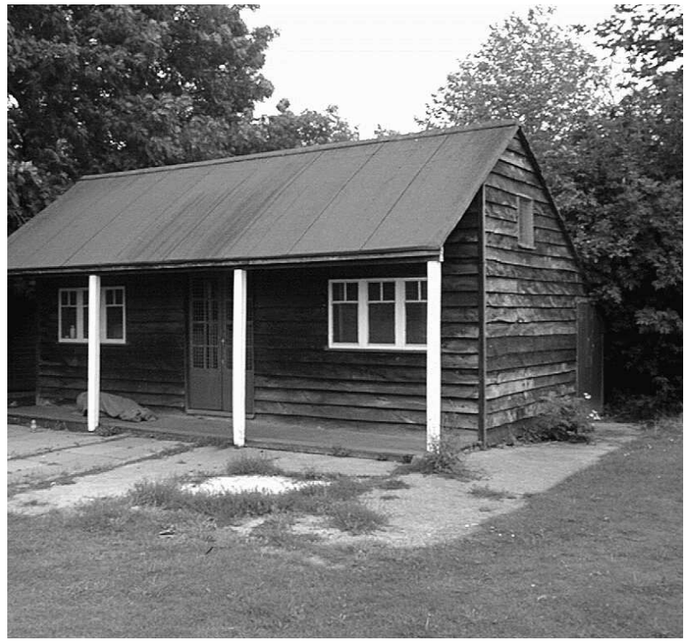
Projective Reconstruction Metric Reconstruction

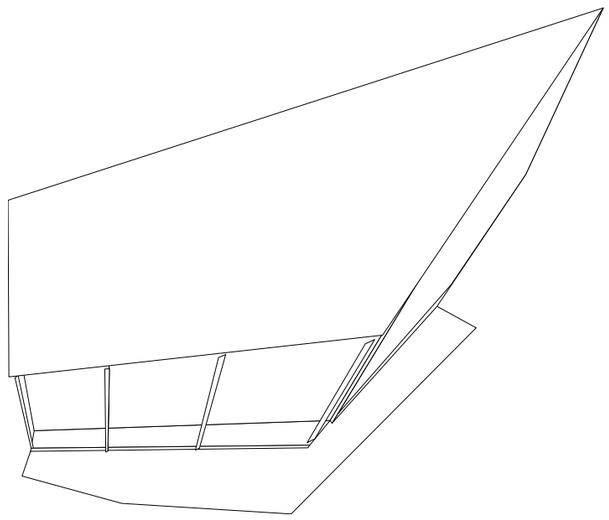
- Remaining ambiguity is rotation (3), translation (3) and scale (1).
- Only 8 parameters required to rectify entire sequence ($15 - 7 = 8$).

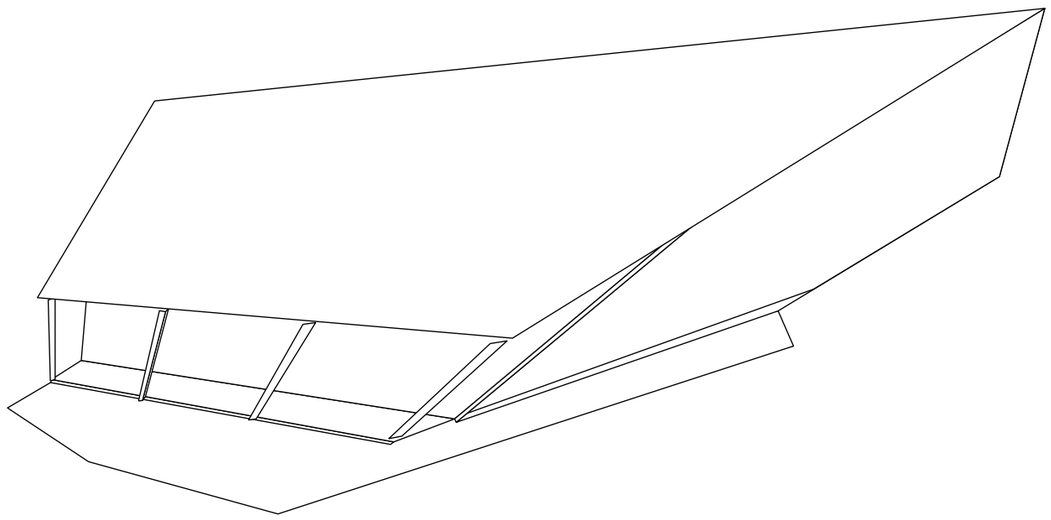
How?

- Calibration points: position of 5 scene points.
- Scene geometry: e.g. parallel lines/planes, orthogonal lines/planes, length ratios.
- Auto-calibration: e.g. camera aspect ratio constant for sequence.

Projective Reconstruction

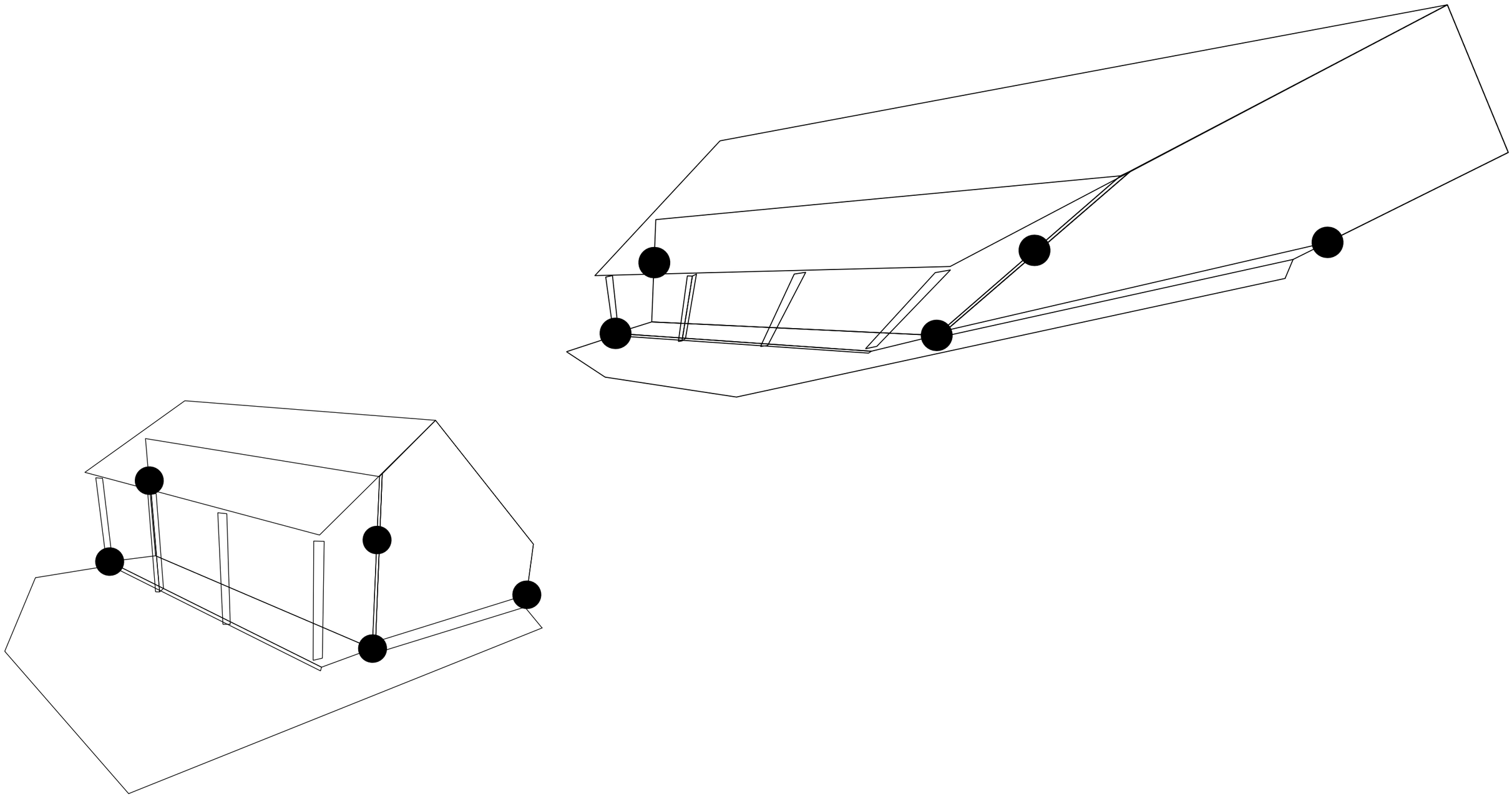






Direct Metric Reconstruction

Use 5 or more 3D points with known Euclidean coordinates to determine $\mathtt{H}$



Stratified Reconstruction

Given a projective reconstruction $\{\mathtt{P}^j, \mathbf{X}_i\}$, compute a metric reconstruction via an intermediate affine reconstruction.

(i) affine reconstruction: Determine the vector $\mathbf{p}$ which defines π_∞. An affine reconstruction is obtained as $\{\mathtt{P}^j\mathtt{H}_\mathrm{P}, \mathtt{H}_\mathrm{P}^{-1}\mathbf{X}_i\}$ with

$$\mathtt{H}_\mathrm{P} = \begin{bmatrix} \mathtt{I} & \mathbf{0} \\ -\mathbf{p}^\top & 1 \end{bmatrix}$$

(ii) Metric reconstruction: is obtained as $\{\mathtt{P}_\mathrm{A}^j\mathtt{H}_\mathrm{A}, \mathtt{H}_\mathrm{A}^{-1}\mathbf{X}_{\mathrm{A}_i}\}$ with

$$\mathtt{H}_\mathrm{A} = \begin{bmatrix} \mathtt{K} & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

Stratified Reconstruction

- Start with a projective reconstruction.
- Find transformation to upgrade to affine reconstruction.
  - Equivalent to finding the plane at infinity.
- Find transformation to upgrade to metric (Euclidean) reconstruction.
  - Equivalent to finding the “absolute conic”
- Equivalent to camera calibration
  - If camera calibration is known then metric reconstruction is possible.
  - Metric reconstruction implies knowledge of angles – camera calibration.

Anatomy of a 3D projective transformation

- General 3D projective transformation represented by a 4×4 matrix.

$$\mathtt{H} = \begin{bmatrix} s\mathtt{R}\mathtt{K} & \mathbf{t} \\ \mathbf{v}^\top & 1 \end{bmatrix} = \begin{bmatrix} s\mathtt{R} & \mathbf{t} \\ & 1 \end{bmatrix} \begin{bmatrix} \mathtt{K} & \\ & 1 \end{bmatrix} \begin{bmatrix} \mathtt{I} & \\ \mathbf{v}^\top & 1 \end{bmatrix}$$
$$= \text{metric} \times \text{affine} \times \text{projective}$$

Stratified reconstruction ...

(i) Apply the transformations one after the other :

- Projective transformation – reduce to affine ambiguity

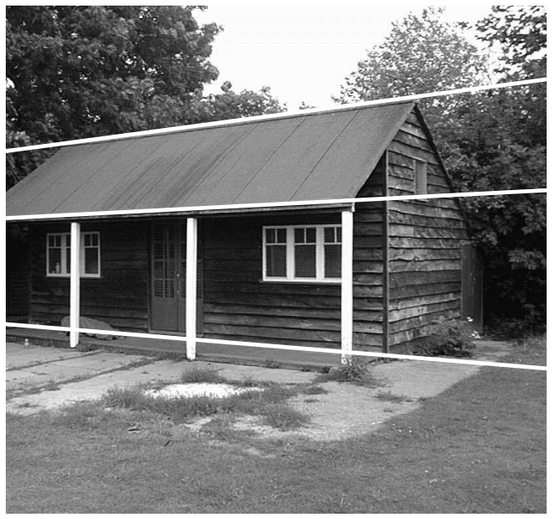
$$\begin{bmatrix} \mathtt{I} & \\ \mathbf{v}^\top & 1 \end{bmatrix}$$

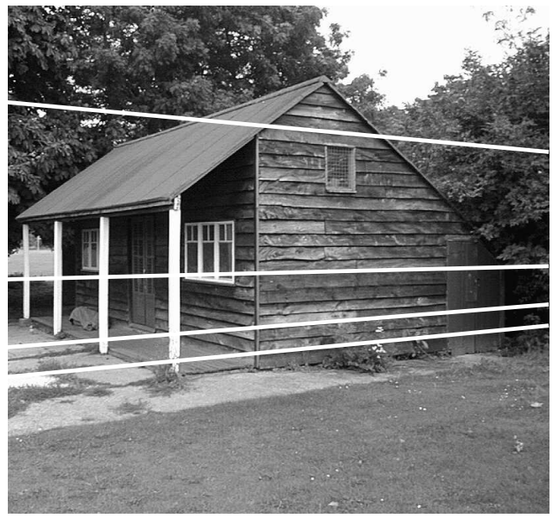
- Affine transformation – reduce to metric ambiguity

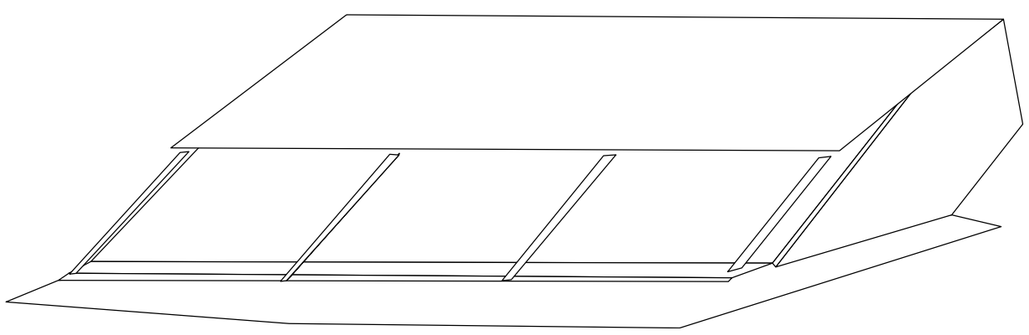
$$\begin{bmatrix} \mathtt{K} & \\ & 1 \end{bmatrix}$$

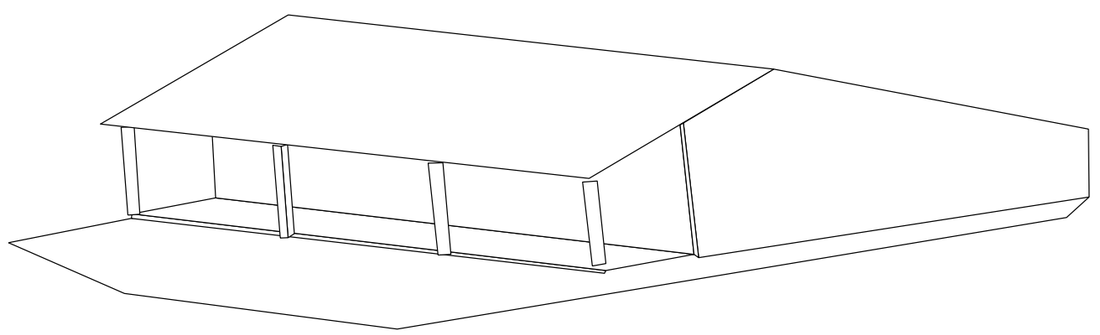
- Metric ambiguity of scene remains

Reduction to affine









Affine reduction using scene constraints - parallel lines

Reduction to affine

Other scene constraints are possible :

- Ratios of distances of points on line (e.g. equally spaced points).
- Ratios of distances on parallel lines.

Points lie in front of the viewing camera.

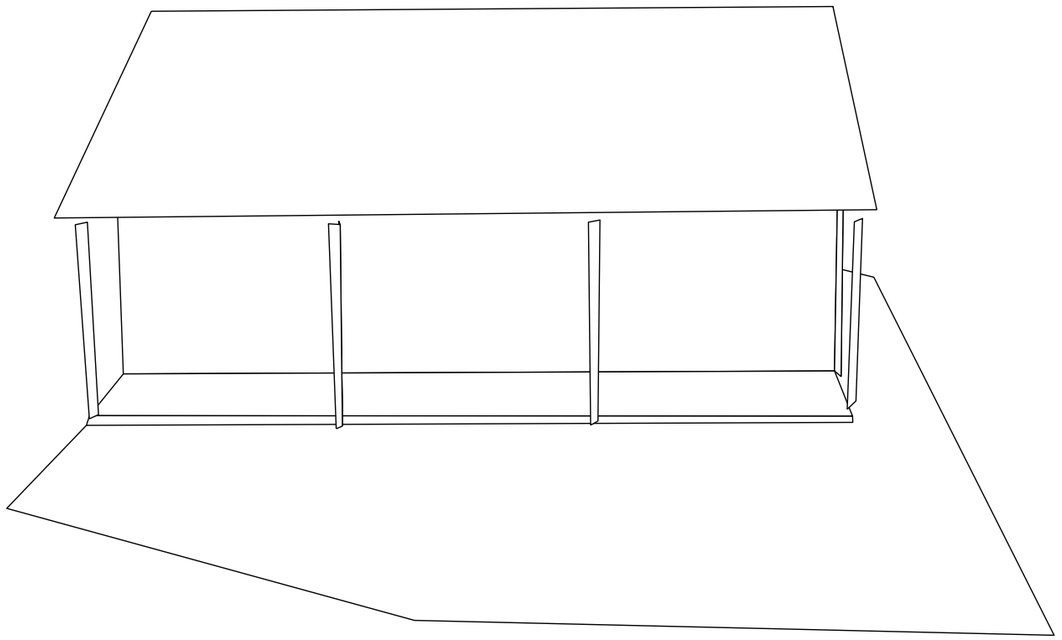
- Constrains the position of the plane at infinity.
- Linear-programming problem can be used to set bounds on the plane at infinity.
- Gives so-called “quasi-affine” reconstruction.
- Reference : Hartley-Azores.

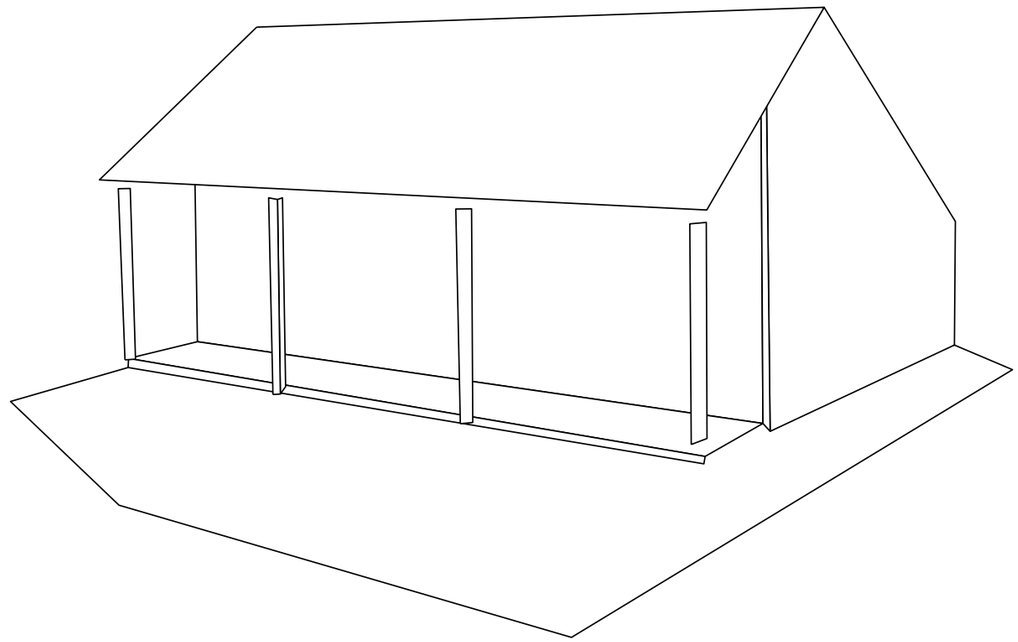
Reduction to affine . . .

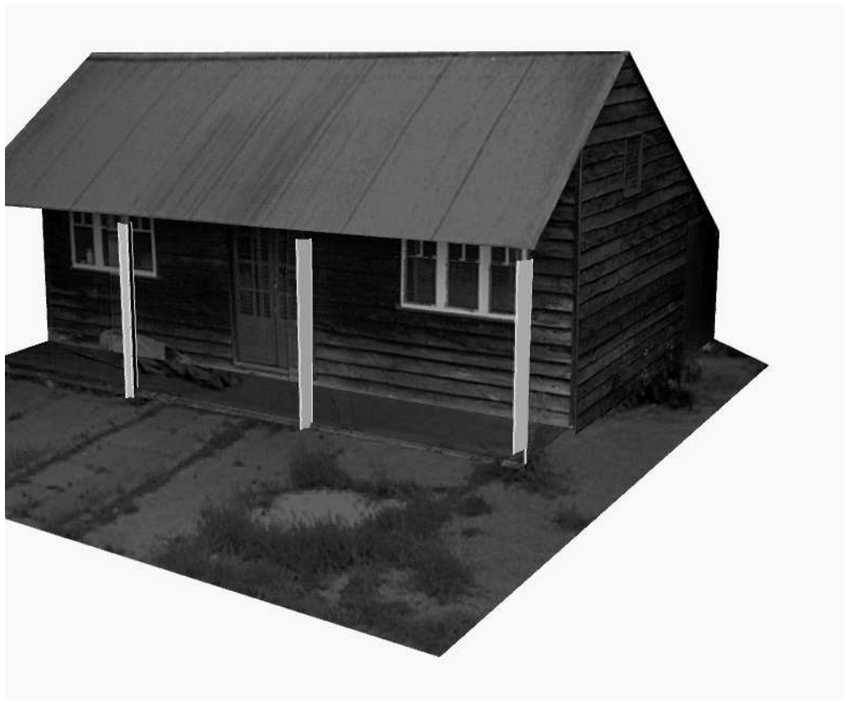
Common calibration of cameras.

- With 3 or more views, one can find (in principle) the position of the plane at infinity.
- Iteration over the entries of projective transform : $\begin{bmatrix} \mathtt{I} & \\ \mathbf{v}^\top & 1 \end{bmatrix}$.
- Not always reliable.
- Generally reduction to affine is difficult.

Metric Reconstruction







Metric Reconstruction . . .

Assume plane at infinity is known.

- Wish to make the step to metric reconstruction.
- Apply a transformation of the form $\begin{bmatrix} K & \\ & 1 \end{bmatrix}$
- Linear solution exists in many cases.

The Absolute Conic

- Absolute conic is an imaginary conic lying on the plane at infinity.
- Defined by

$$\Omega \quad : \quad \mathrm{X}^2 + \mathrm{Y}^2 + \mathrm{Z}^2 = 0 \;\; ; \;\; \mathrm{T} = 0$$

- Contains only imaginary points.
- Determines the Euclidean geometry of the space.
- Represented by matrix $\Omega = \mathrm{diag}(1, 1, 1, 0)$.
- Image of the absolute conic (IAC) under camera $\mathtt{P} = \mathtt{K}[\mathtt{R} \mid \mathbf{t}]$ is given by $\omega = (\mathtt{K}\mathtt{K}^\top)^{-1}$.
- Basic fact :

ω is unchanged under camera motion.

Using the infinite homography

(i) When a camera moves, the image of a plane undergoes a projective transformation.

(ii) If we have affine reconstruction, we can compute the transformation $\mathtt{H}$ of the plane at infinity between two images.

(iii) Absolute conic lies on the plane at infinity, but is unchanged by this image transformation :

(iv) Transformation rule for dual conic $\omega^* = \omega^{-1}$.

$$\omega^* = \mathtt{H}_j \omega^* {\mathtt{H}_j}^\top$$

(v) Linear equations on the entries of ω^*.

(vi) Given three images, solve for the entries of ω^*.

(vii) Compute $\mathtt{K}$ by Choleski factorization of $\omega^* = \mathtt{K}\mathtt{K}^\top$.

Example of calibration

Images taken with a non-translating camera:

Mosaiced image showing projective transformations

Computation of $\mathtt{K}$

Calibration matrix of camera is found as follows :

- Compute the homographies (2D projective transformations) between images.
- Form equations

$$\omega^* = \mathtt{H}_{ij}\omega^*{\mathtt{H}_{ij}}^\top$$

- Solve for the entries of ω^*
- Choleski factorization of $\omega^* = \mathtt{K}\mathtt{K}^\top$ gives $\mathtt{K}$.

Affine to metric upgrade

Principal is the same for non-stationary cameras once principal plane is known.

- $\mathtt{H}_{ij}$ is the "infinite homography" (i.e. via the plane at infinity) between images i and j.
- May be computed directly from affinely-correct camera matrices.
- Given camera matrices

$$\mathtt{P}_i = [\mathtt{M}_i|\mathbf{t}_i] \quad ; \quad \mathtt{P}_i = [\mathtt{M}'_i|\mathbf{t}_i]$$

- Infinite homography is given by

$$\mathtt{H} = \mathtt{M}'_i\mathtt{M}^{-1}$$

- Algorithm proceeds as for fixed cameras.

Changing internal camera parameters

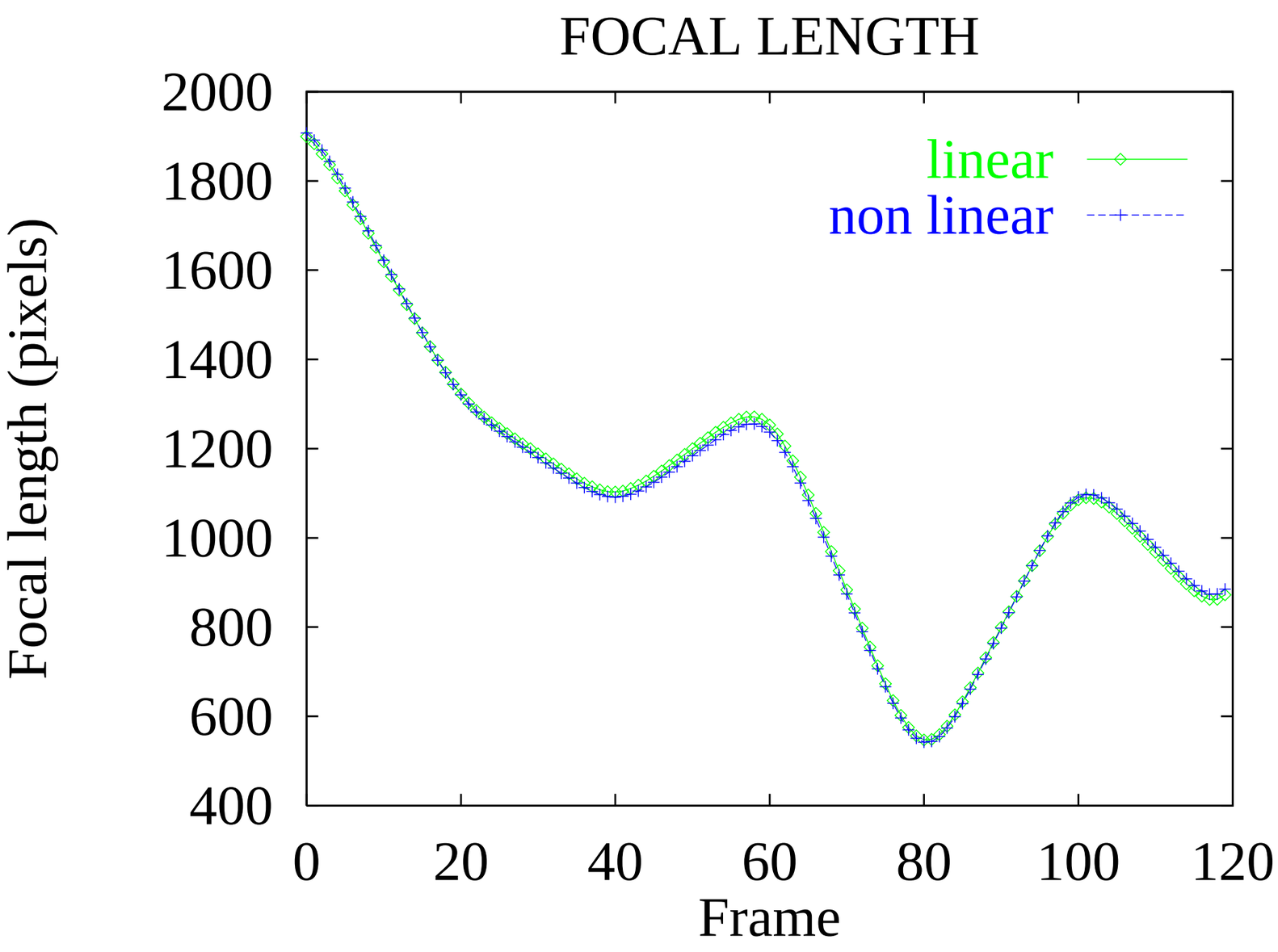
The previous calibration procedure (affine-to-metric) may be generalized to case of changing internal parameters.

See paper tomorrow given by Agapito.

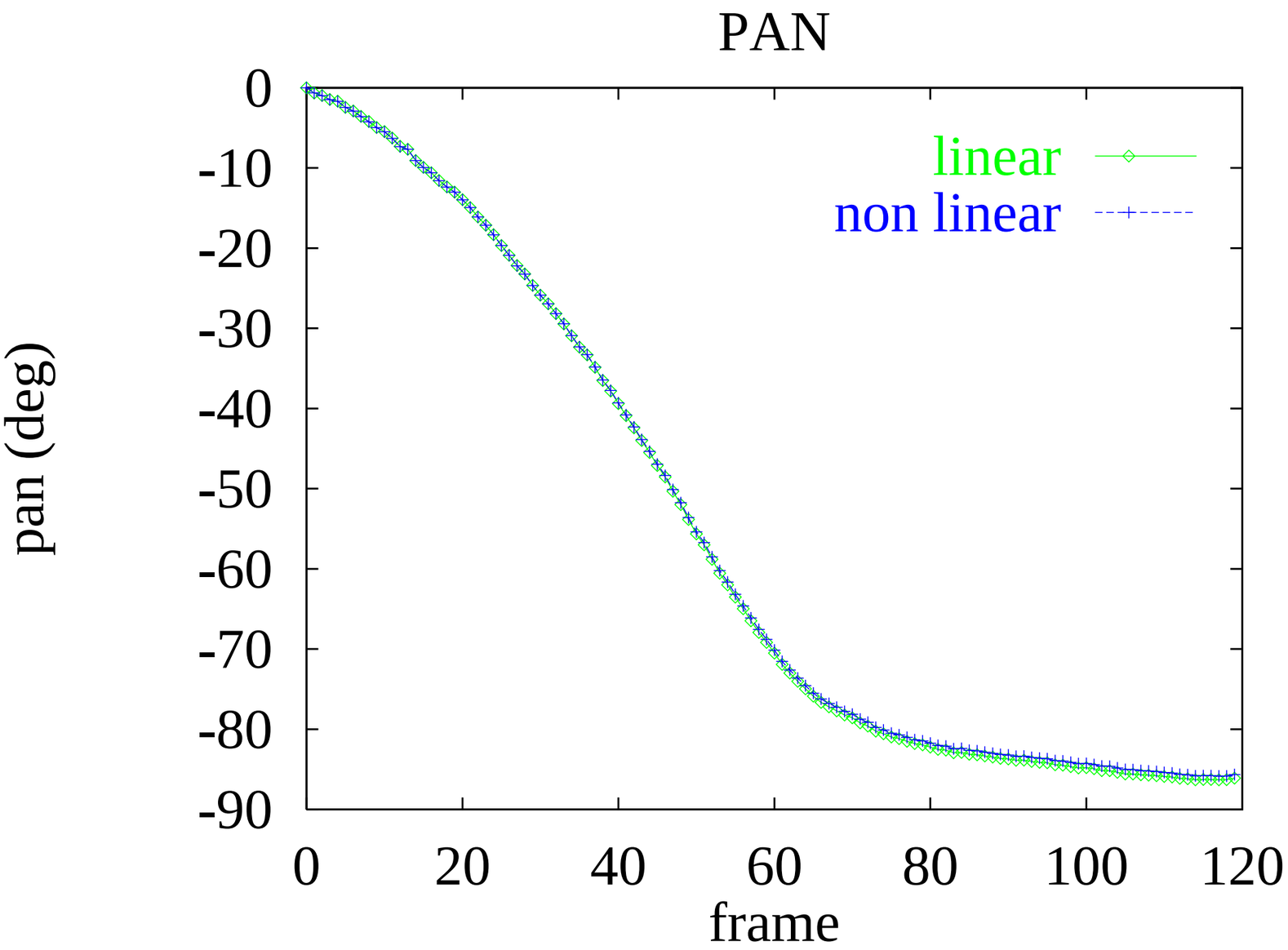
Nice Video

< Video Here >

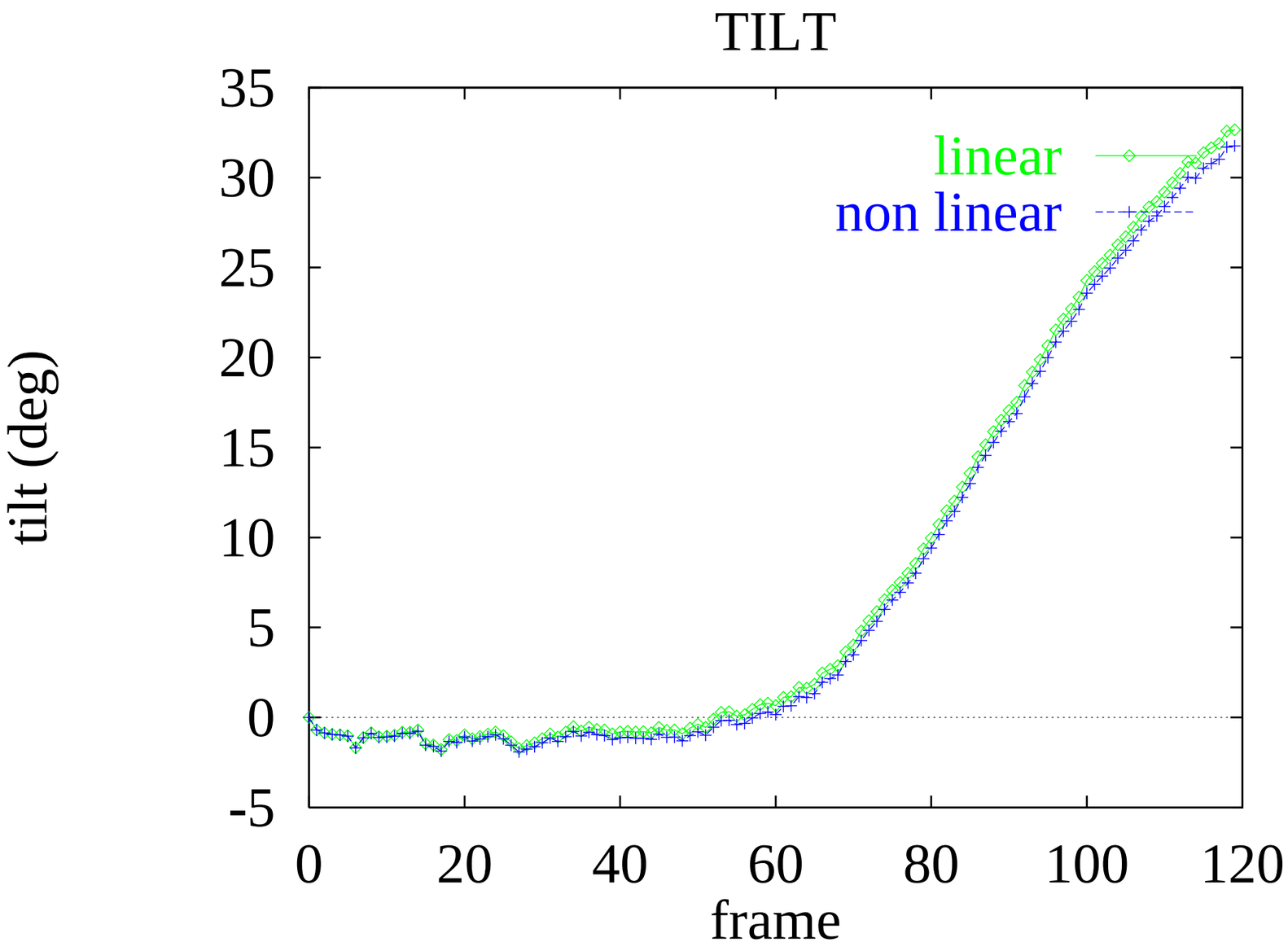
Nice Calibration Results – Focal Length



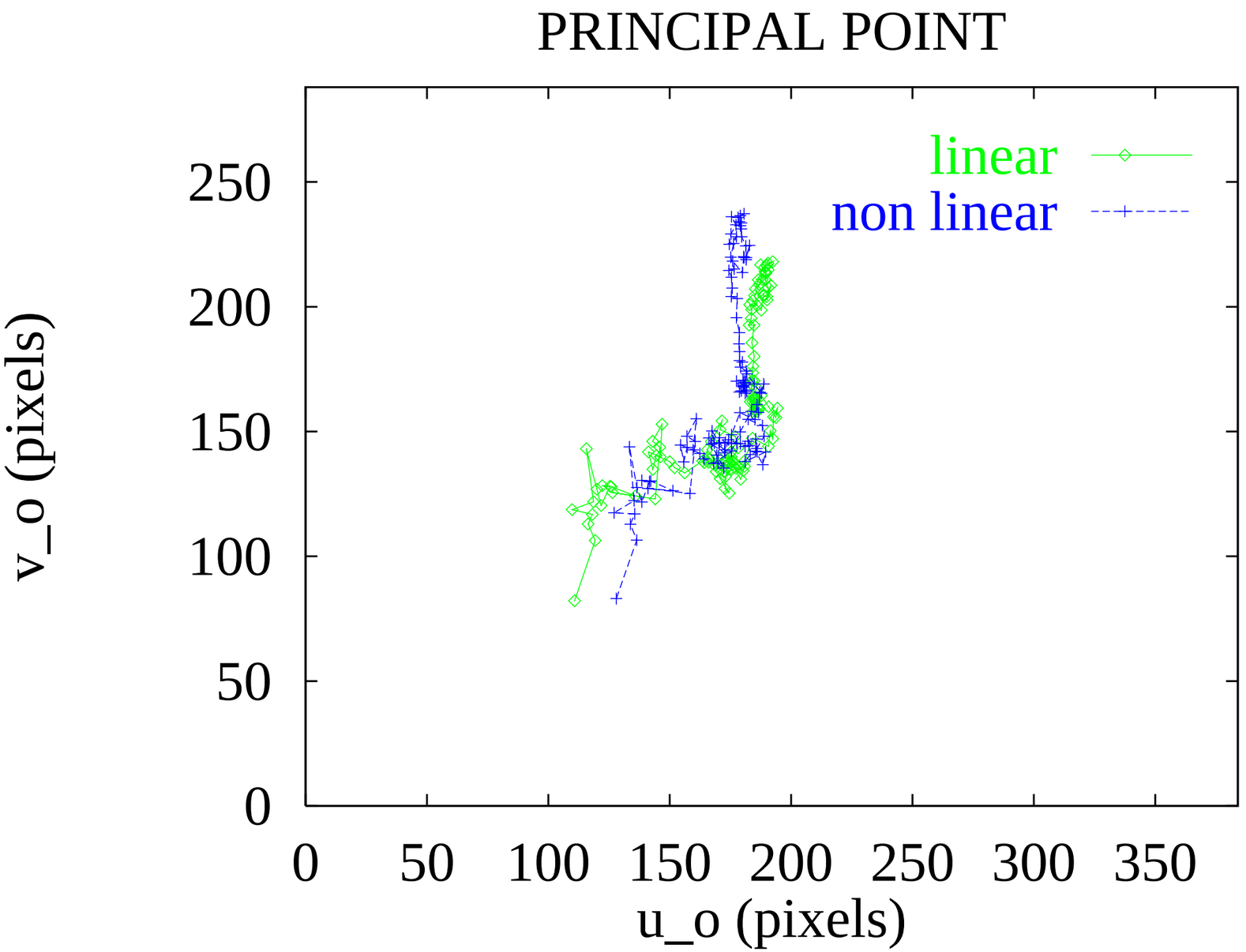
Nice Calibration Results – Pan angle



Nice Calibration Results – Tilt angle



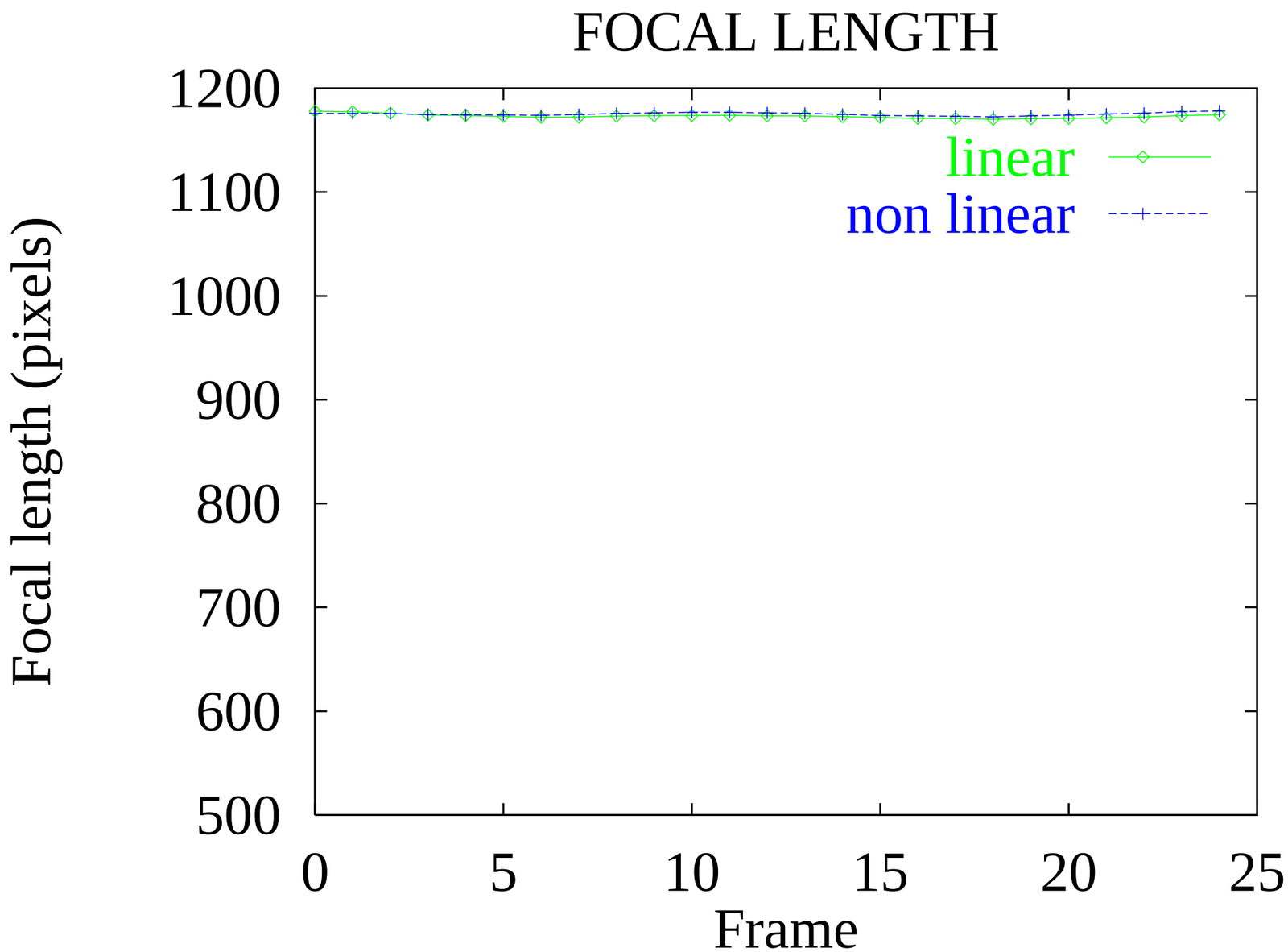
Nice Calibration Results – Focal Length



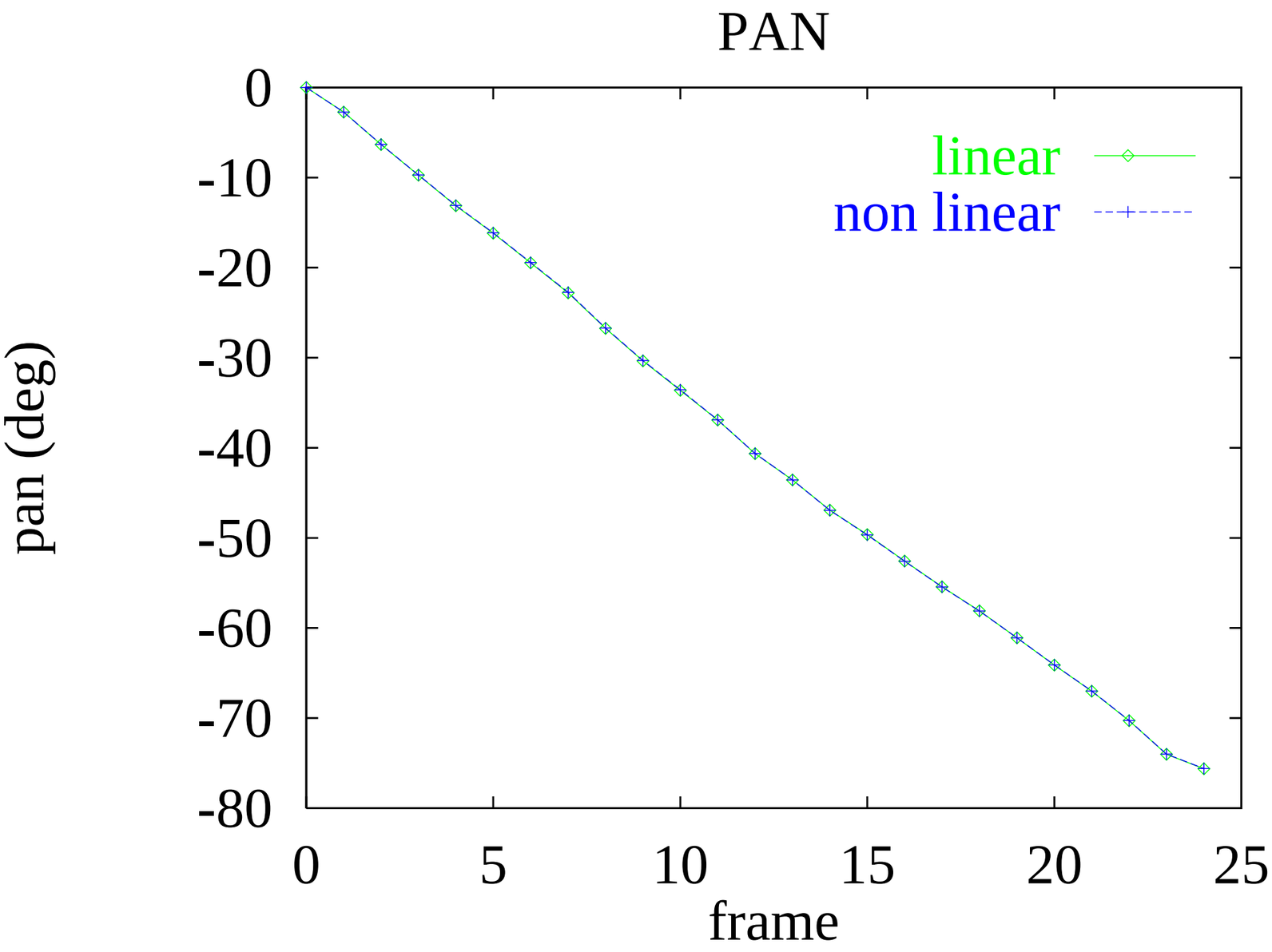
Keble College Video

< Video Here >

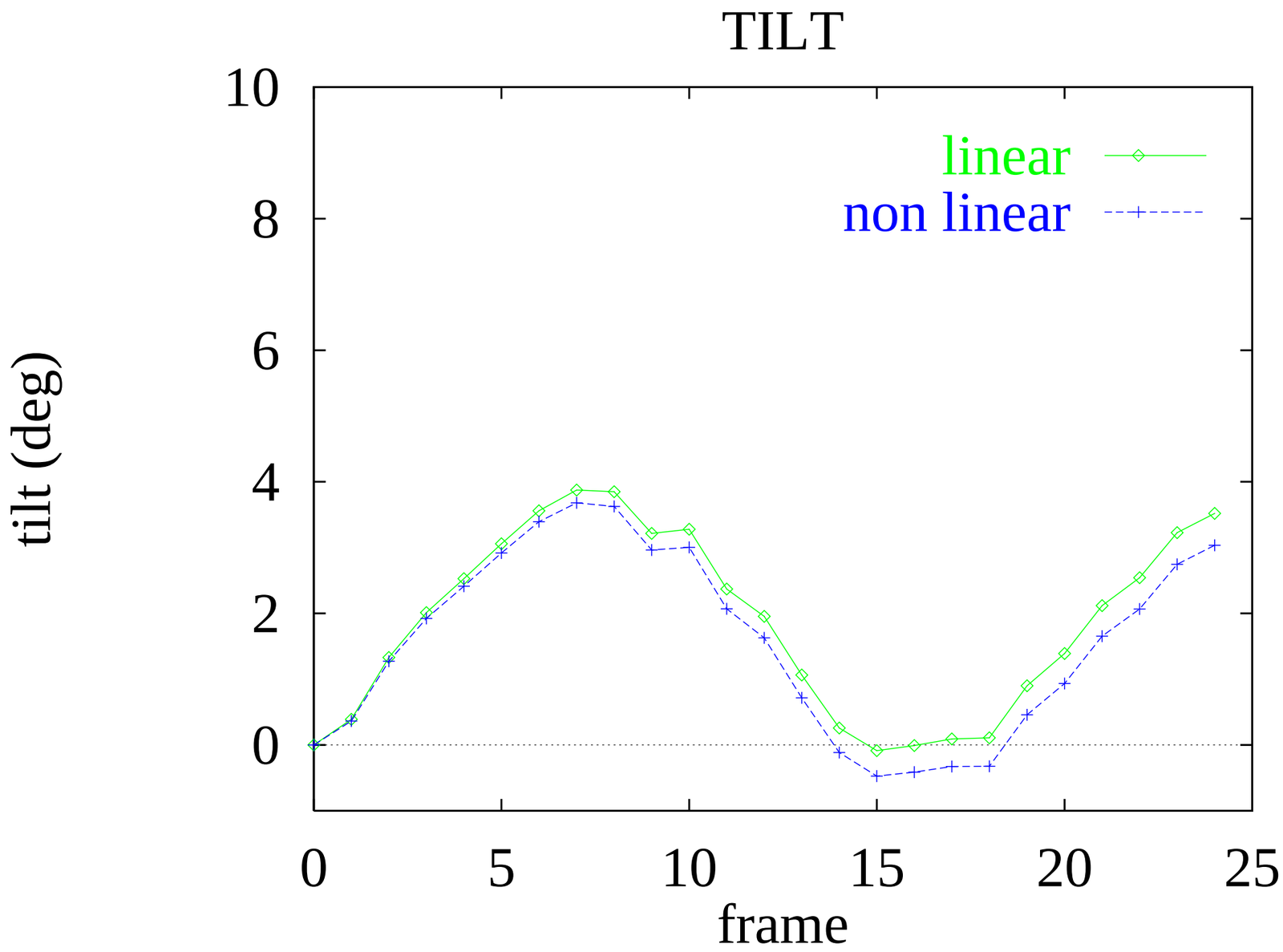
Keble Calibration Results – Focal Length



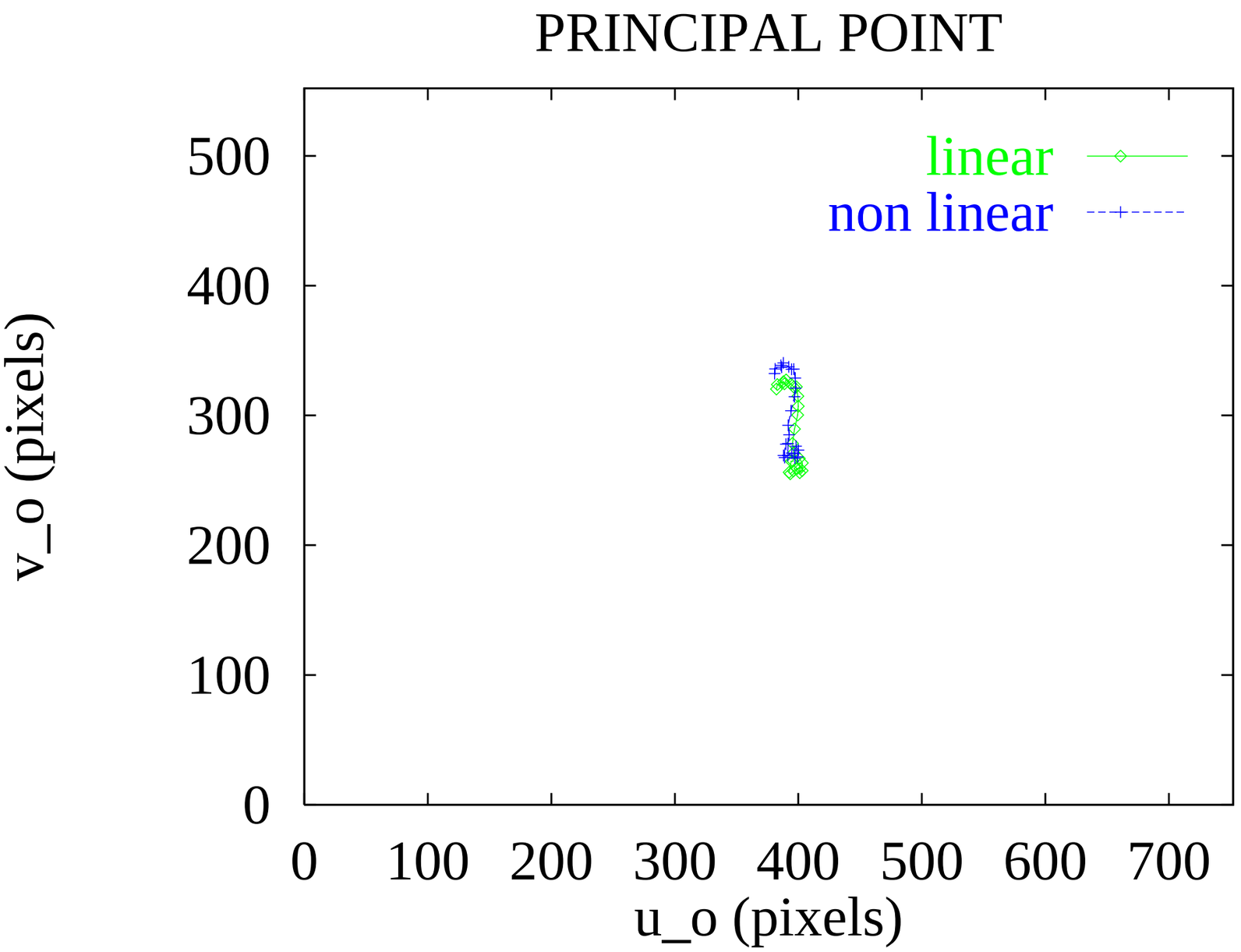
Keble Calibration Results – Pan angle



Keble Calibration Results – Tilt angle



Keble Calibration Results – Focal Length



The Trifocal Tensor

The Trifocal Tensor

(i) Defined for three views.

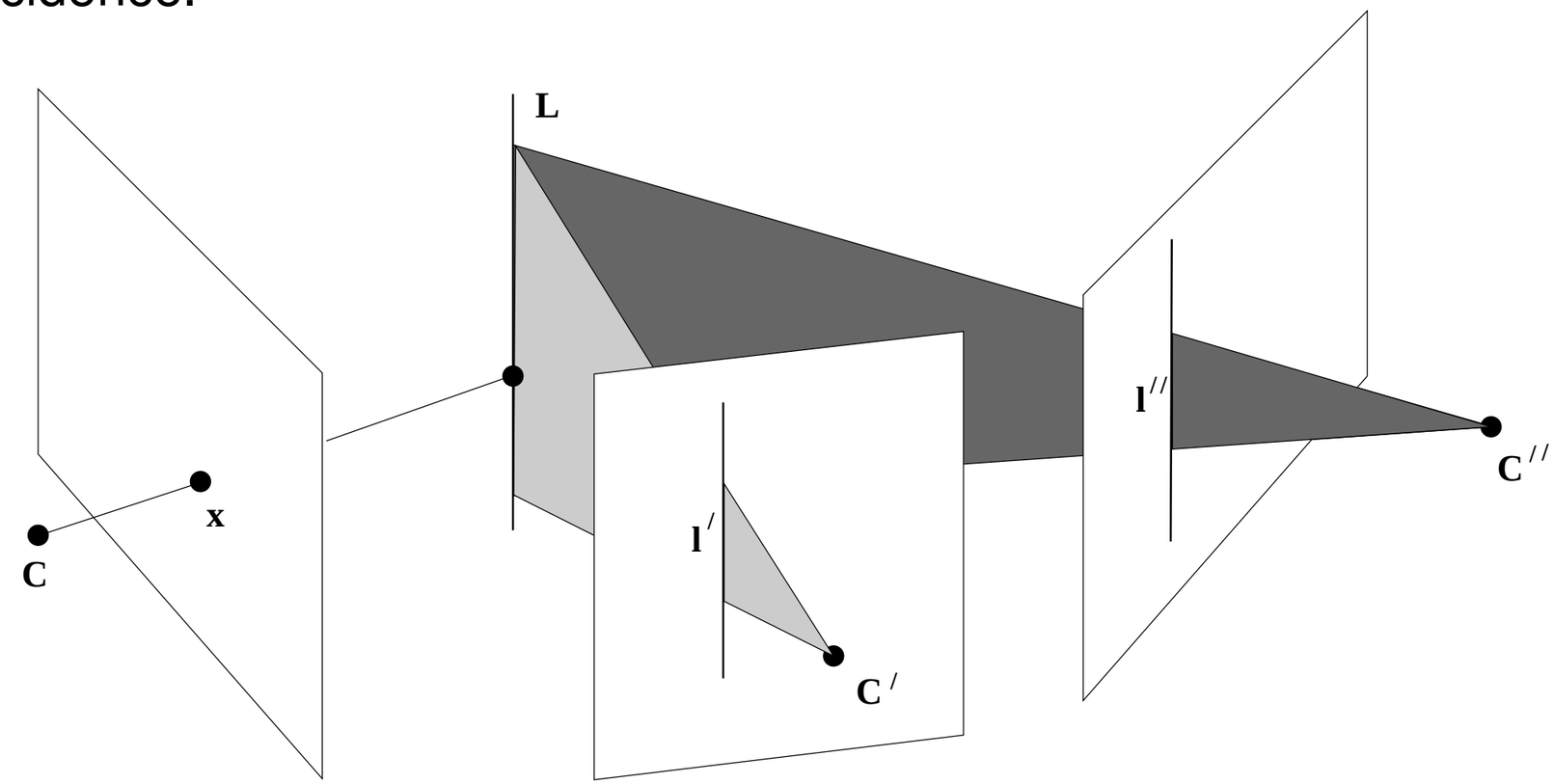
(ii) Plays a similar rôle to Fundmental matrix for two views.

(iii) Unlike fundamental matrix, trifocal tensor also relates lines in three views.

(iv) Mixed combinations of lines and points are also related.

Geometry of three views

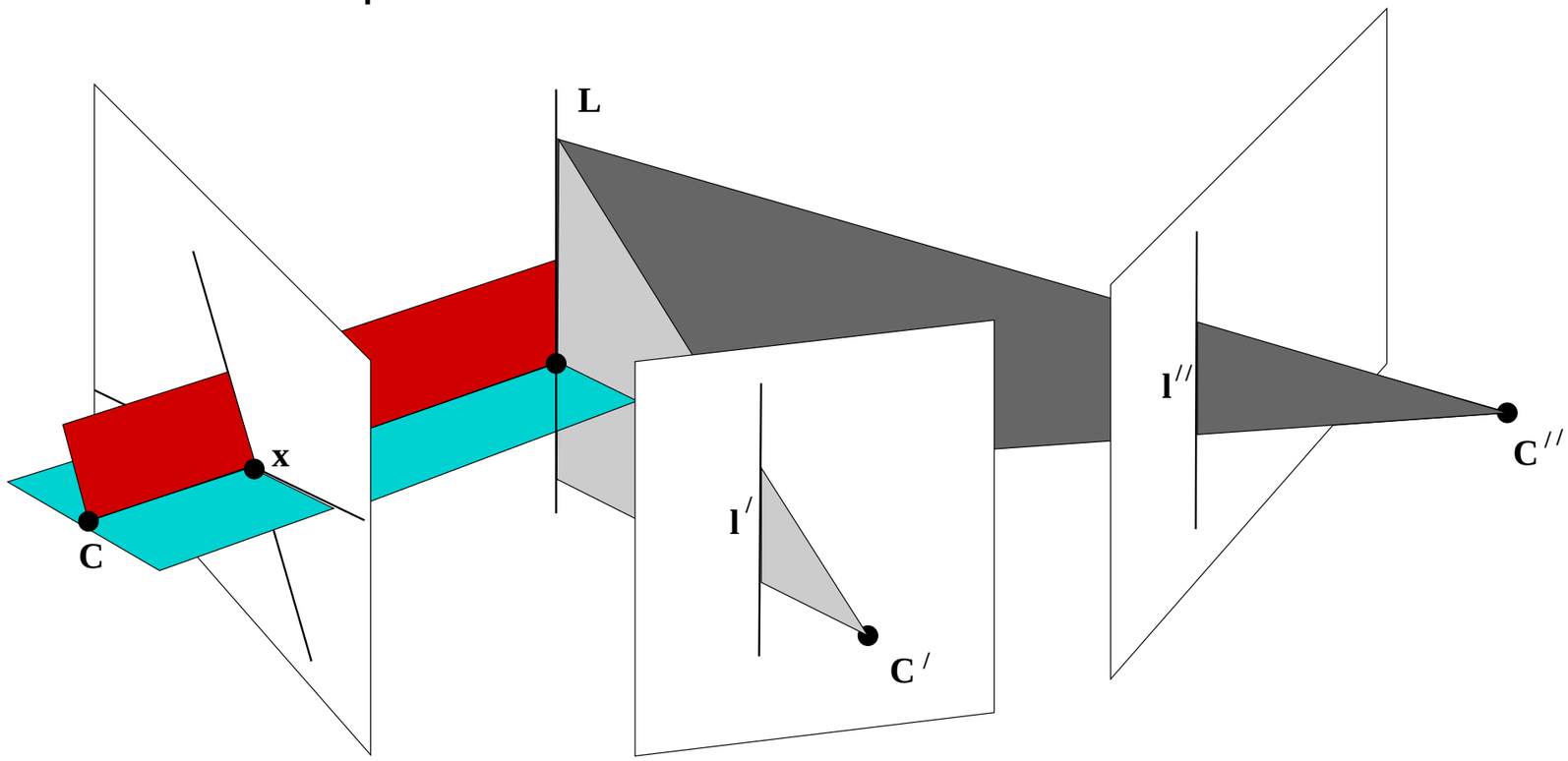
Point-line-line incidence.



- Correspondence $\mathbf{x} \leftrightarrow \mathbf{l}' \leftrightarrow \mathbf{l}''$

Geometry of three views . . .

- Let $\mathbf{l}^{(1)}$ and $\mathbf{l}^{(2)}$ be two lines that meet in $\mathbf{x}$.
- General line $\mathbf{l}$ back-projects to a plane $\mathbf{l}^\top \mathtt{P}$.
- Four plane are $\mathbf{l}^{(1)\top}\mathtt{P}$, $\mathbf{l}^{(2)\top}\mathtt{P}'$, $\mathbf{l}'^\top\mathtt{P}'$ and $\mathbf{l}''^\top\mathtt{P}''$
- The four planes meet in a point.



The trifocal relationship

Four planes meet in a point means determinant is zero.

$$\det \begin{bmatrix} \mathbf{l}^{(1)\top}\mathtt{P} \\ \mathbf{l}^{(2)\top}\mathtt{P} \\ \mathbf{l}'^{\top}\mathtt{P}' \\ \mathbf{l}''^{\top}\mathtt{P}'' \end{bmatrix} = 0$$

- This is a linear relationship in the line coordinates.
- Also (less obviously) linear in the entries of the point $\mathbf{x} = \mathbf{l}^{(1)} \times \mathbf{l}^{(2)}$.

This is the trifocal tensor relationship.

Tensor Notation

Point coordinates.

- Consider basis set $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$.
- Point is represented by a vector $\mathbf{x} = (x^1, x^2, x^3)^\top$.
- New basis : $\hat{\mathbf{e}}_j = \sum_i \mathtt{H}^i_j \, \mathbf{e}_i$.
- With respect to new basis $\mathbf{x}$ represented by

$$\hat{\mathbf{x}} = (\hat{x}^1, \hat{x}^2, \hat{x}^3) \text{ where } \hat{\mathbf{x}} = \mathtt{H}^{-1}\mathbf{x}$$

- If basis is transformed according to $\mathtt{H}$, then point coordinates transform according to $\mathtt{H}^{-1}$.
- **Terminology :** x^i transforms contravariantly.
- Use **upper indices** for *contravariant* quantities.

Tensor Notation . . .

Line coordinates

- Line is represented by a vector $\mathbf{l} = (l_1, l_2, l_3)$
- In new coordinate system $\hat{\mathbf{e}}_j$, line has coordinate vector $\hat{\mathbf{l}}$,

$$\hat{\mathbf{l}}^\top = \mathbf{l}^\top \mathtt{H}$$

- Line coordinates transform according to $\mathtt{H}$.
- Preserves incidence relationship. Point lies on line if :

$$\hat{\mathbf{l}}^\top \hat{\mathbf{x}} = (\mathbf{l}^\top \mathtt{H})(\mathtt{H}^{-1}\mathbf{x}) = \mathbf{l}^\top \mathbf{x} = 0$$

- **Terminology :** l_j transforms covariantly.
- Use **lower indices** for *covariant* quantities.

Summation notation

- Repeated index in upper and lower positions implies summation.

Example

Incidence relation is written $l_i x^i = 0$.

Transformation of covariant and contravariant indices

Contravariant transformation

$$\hat{x}^j = (\mathtt{H}^{-1})^j_i x^i$$

Covariant transformation

$$\hat{l}_j = \mathtt{H}^i_j l_i$$

More transformation examples

Camera mapping has one covariant and one contravariant index : P^i_j. Transformation rule $\hat{\mathtt{P}} = \mathtt{G}^{-1}\mathtt{PF}$ is

$$\hat{P}^j_i = (G^{-1})^j_s \, P^s_r \, F^r_i$$

Trifocal tensor T^{jk}_i has one covariant and two contravariant indices.

Transformation rule :

$$\hat{T}^{jk}_i = F^r_i \, (G^{-1})^j_s \, (H^{-1})^k_t T^{st}_r$$

The ϵ tensor

Tensor ϵ_{rst} :

- Defined for $r, s, t = 1, \ldots, 3$

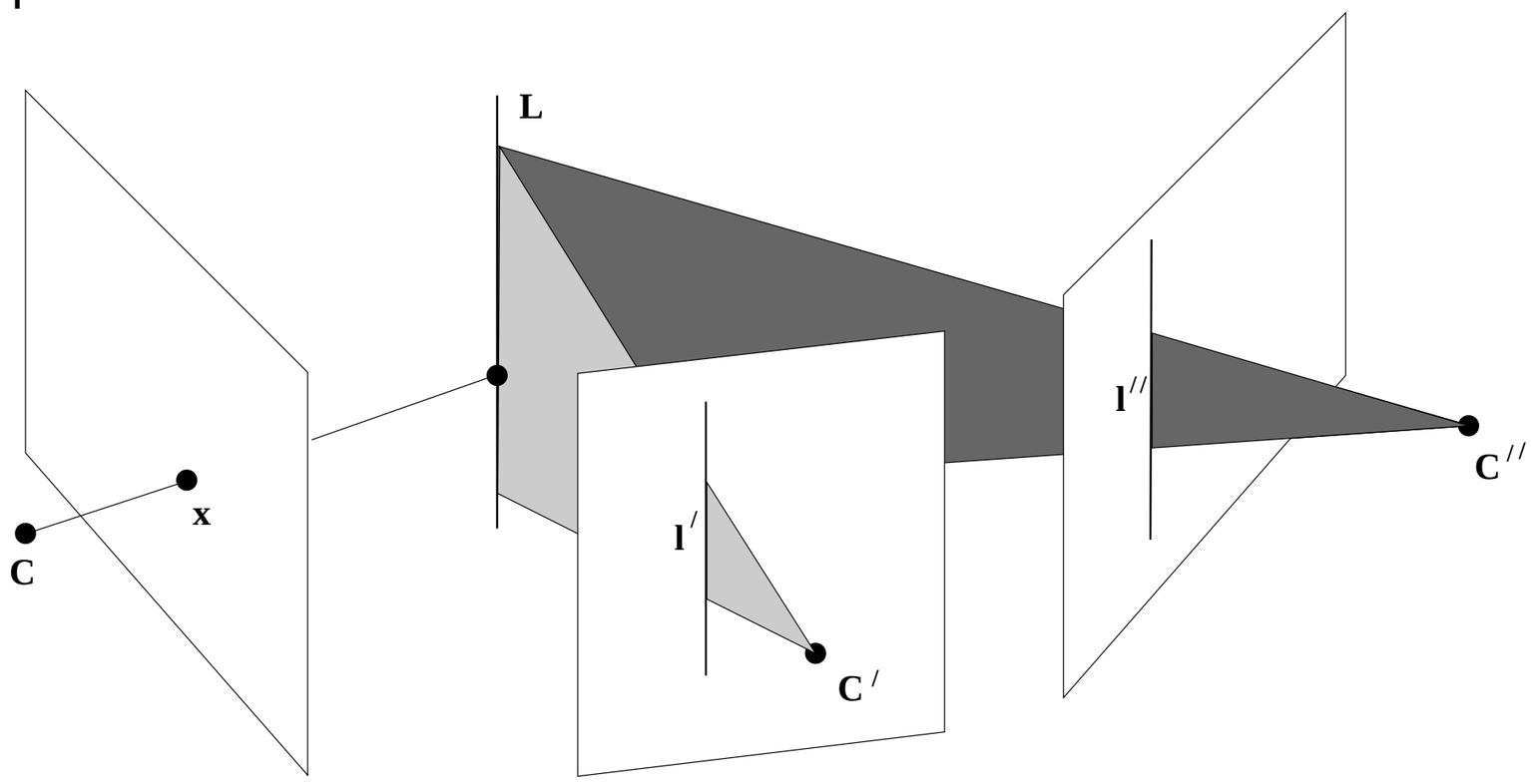
$$\begin{aligned} \epsilon_{rst} &= 0 \quad \text{unless } r, s \text{ and } t \text{ are distinct} \\ &= +1 \quad \text{if } rst \text{ is an even permutation of } 123 \\ &= -1 \quad \text{if } rst \text{ is an odd permutation of } 123 \end{aligned}$$

- Related to the cross-product :

$$\mathbf{c} = \mathbf{a} \times \mathbf{b} \quad \Longleftrightarrow c_i = \epsilon_{ijk} a^j b^k \ .$$

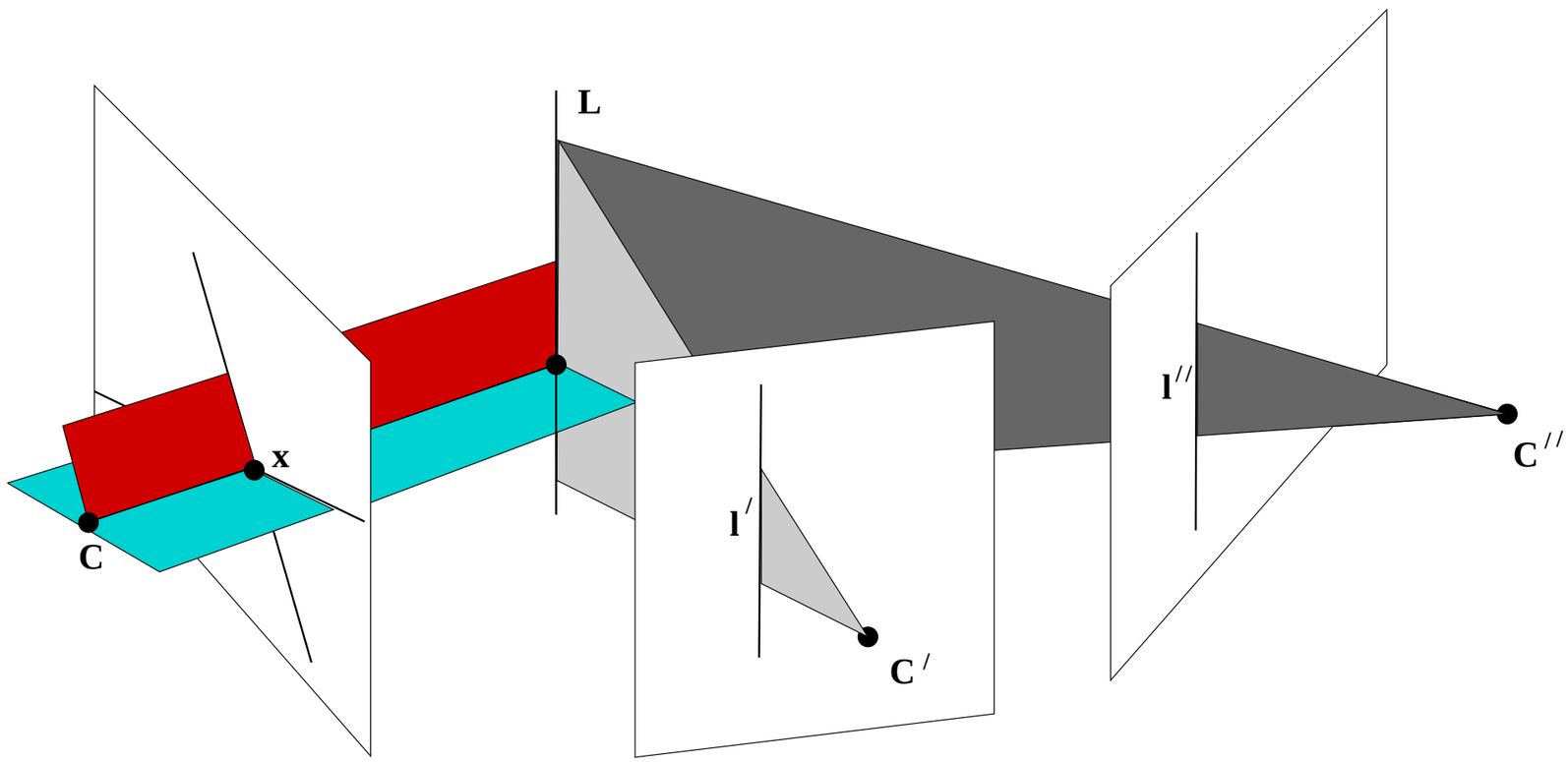
Basic Trifocal constraint

Basic relation is a point-line-line incidence.



- Point x^i in image 1
- lines l'_j and l''_k in images 2 and 3.

Basic Trifocal constraint . . .



- Let $l_r^{(1)}$ and $l_s^{(2)}$ be two lines that meet in point x^i.
- The four lines $l_r^{(1)}$, $l_s^{(2)}$, l'_j and l''_k back-project to planes in space.
- The four planes meet in a point.

Derivation of the basic three-view relationship

- Line l_i back-projects to plane $l_i \mathbf{P}^i$ where $\mathbf{P}^i$ is i-th row.
- Four planes are coincident if

$$(l_r^{(1)} \mathbf{P}^r) \wedge (l_s^{(2)} \mathbf{P}^s) \wedge (l'_j \mathbf{P}'^j) \wedge (l''_k \mathbf{P}''^k) = 0$$

where 4-way wedge $\wedge$ means determinant.

- Thus

$$l_r^{(1)} l_s^{(2)} l'_j l''_k \quad \mathbf{P}^r \wedge \mathbf{P}^s \wedge \mathbf{P}'^j \wedge \mathbf{P}''^k = 0$$

- Multiply by constant $\epsilon^{rsi}\epsilon_{rsi}$ gives

$$\epsilon^{rsi} l_r^{(1)} l_s^{(2)} l'_j l''_k \quad \epsilon_{rsi} \mathbf{P}^r \wedge \mathbf{P}^s \wedge \mathbf{P}'^j \wedge \mathbf{P}''^k = 0$$

- Intersection (cross-product) of $l_r^{(1)}$ and $l_s^{(2)}$ is the point x^i :

$$l_r^{(1)} l_s^{(2)} \epsilon^{rsi} = x^i$$

Definition of the trifocal tensor

Basic relationship is

$$x^i \, l'_j \, l''_k \quad \epsilon_{rsi} \mathbf{P}^r \wedge \mathbf{P}^s \wedge \mathbf{P}'^j \wedge \mathbf{P}''^k = 0$$

Define

$$\epsilon_{rsi} \mathbf{P}^r \wedge \mathbf{P}^s \wedge \mathbf{P}'^j \wedge \mathbf{P}''^k = T_i^{jk}$$

Point-line-line relation is

$$x^i \, l'_j \, l''_k \, T_i^{jk} = 0$$

T_i^{jk} is covariant in one index (i), contravariant in the other two.

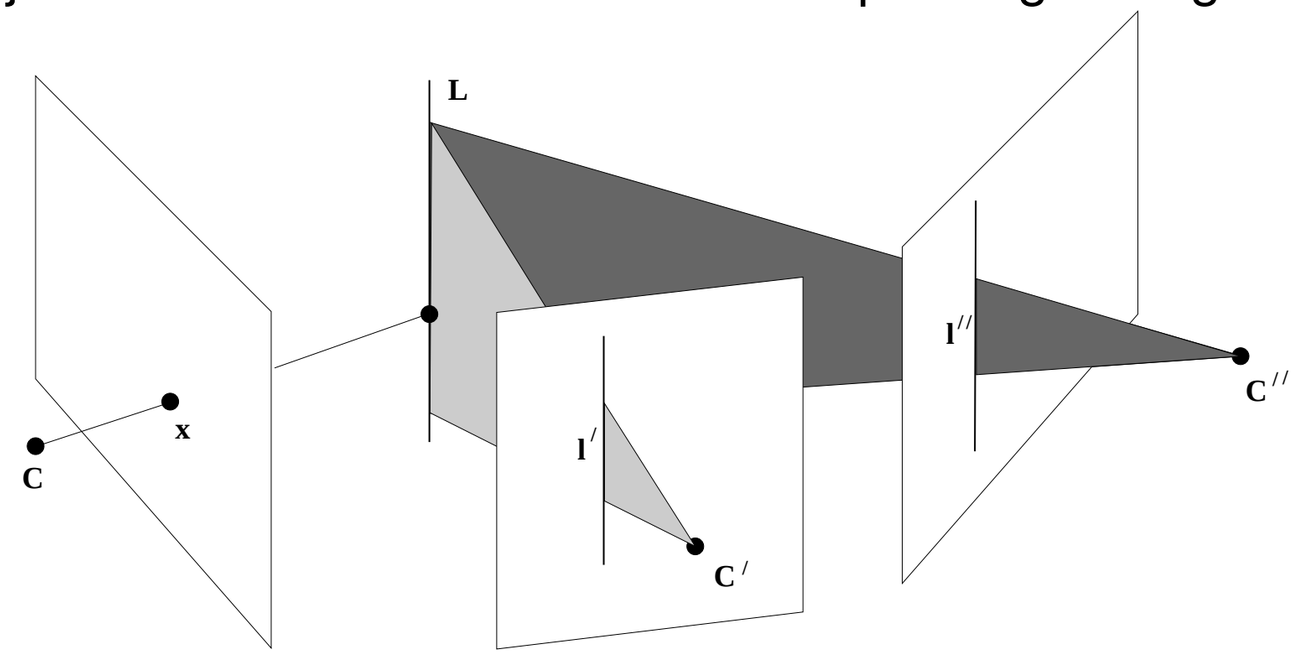
Line Transfer

Basic relation is

$$x^i \, l'_j \, l''_k \, T_i^{jk} = 0$$

Interpretation :

- Back projected ray from $\mathbf{x}$ meets intersection of back-projected planes from $\mathbf{l}'$ and $\mathbf{l}''$.
- Line in space projects to lines $\mathbf{l}'$ and $\mathbf{l}''$ and to a line passing through $\mathbf{x}$.



Line transfer

- Denote $l_i = l'_j \, l''_k \, T_i^{jk}$
- See that $x^i \, l_i = 0$ when x^i lies on the projection of the intersection of the planes.
- Thus l_i represents the transferred line corresponding to l'_j and l''_k.
- We write

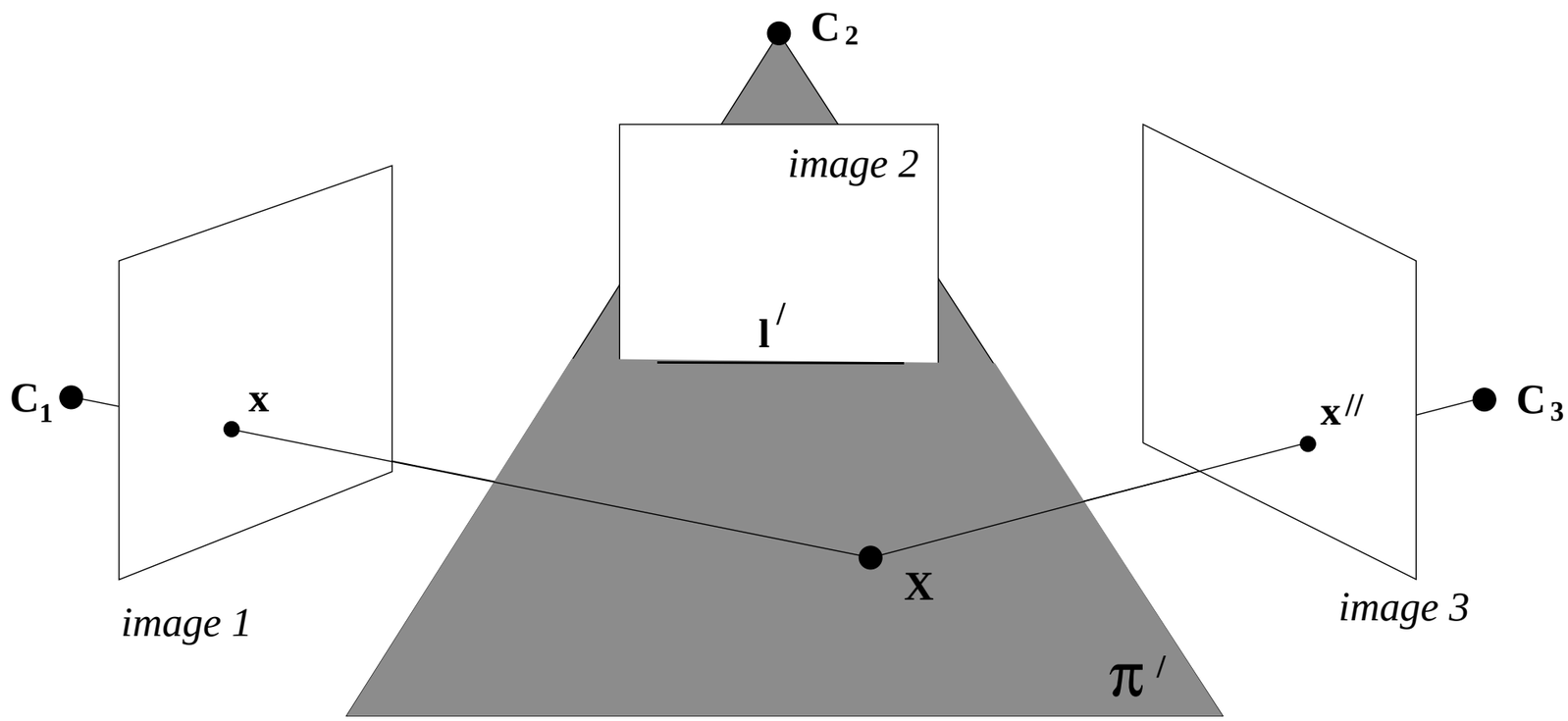
$$l_i \approx l'_j \, l''_k \, T_i^{jk}$$

- Cross-product of the two sides are equal :

$$l_r \epsilon^{irs} l'_j \, l''_k \, T_i^{jk} = 0^s$$

- Derived from basic relation $x^i l'_j l''_k T_i^{jk}$ by replacing x^i by $l_r \epsilon^{irs}$.

Point transfer via a plane



- Line $\mathbf{l}'$ back-projects to a plane π'.
- Ray from $\mathbf{x}$ meets π' in a point $\mathbf{X}$.
- This point projects to point $\mathbf{x}''$ in the third image.
- For fixed $\mathbf{l}'$, mapping $\mathbf{x} \mapsto \mathbf{x}''$ is a homography.

Point-transfer and the trifocal tensor

- If $\mathbf{l}''$ is **any** line through $\mathbf{x}''$, then trifocal condition holds.

$$l''_k \, (x^i \, l'_j \, T_i^{jk}) = 0$$

- $x^i \, l'_j \, T_i^{jk}$ must represent the point x''^k.

$$x''^k \approx x^i \, l'_j \, T_i^{jk}$$

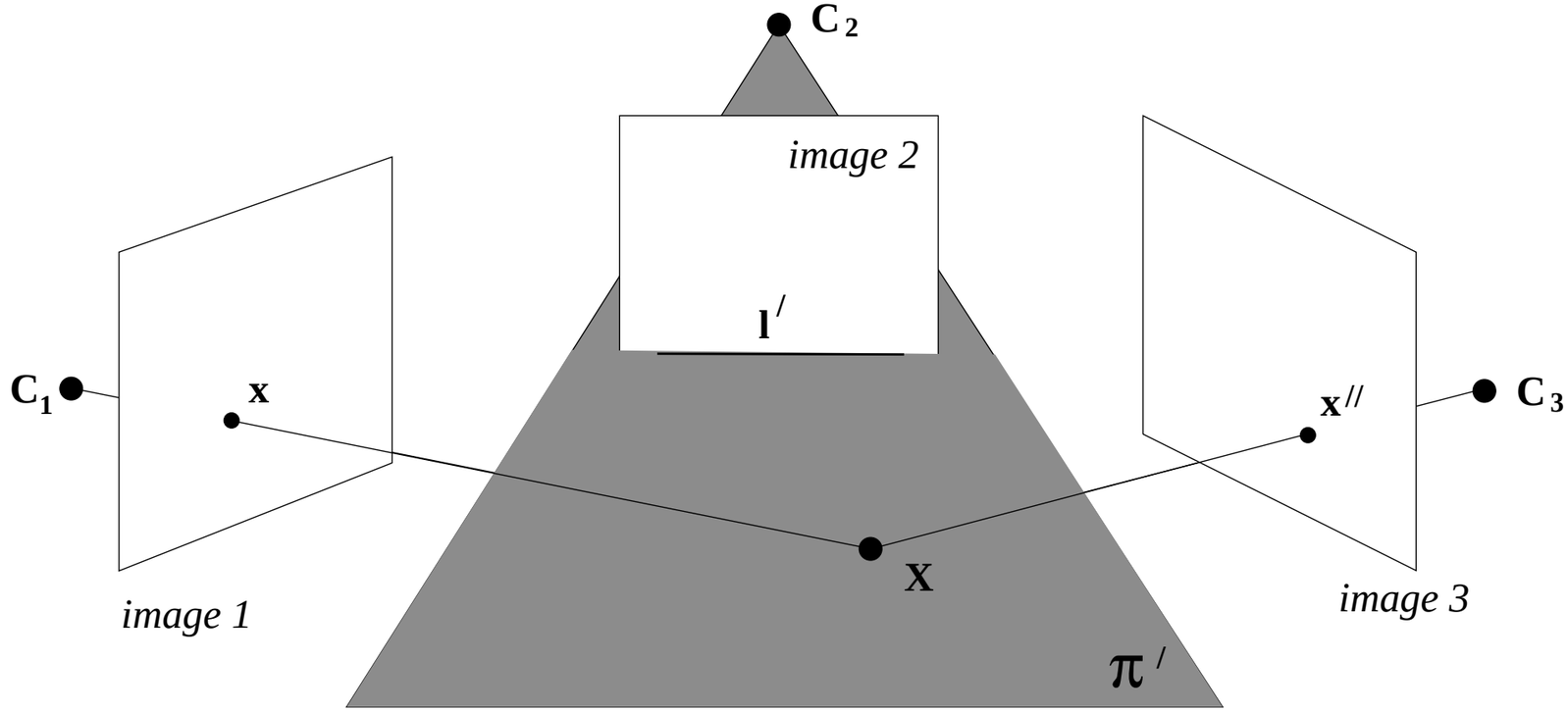
- Alternatively (cross-product of the two sides)

$$x^i \, l'_j \, \epsilon_{krs} x''^r \, T_i^{jk} = 0_s$$

- Derived from basic relation $x^i l'_j l''_k T_i^{jk}$ by replacing l''_k by $x''^r \epsilon_{krs}$.

Contraction of trifocal tensor with a line

- Write $H_i^k = l'_j \, T_i^{jk}$
- Then $x''^k = H_i^k x^i$.
- H_i^k represents the homography from image 1 to image 3 via the plane of the line l'_j.

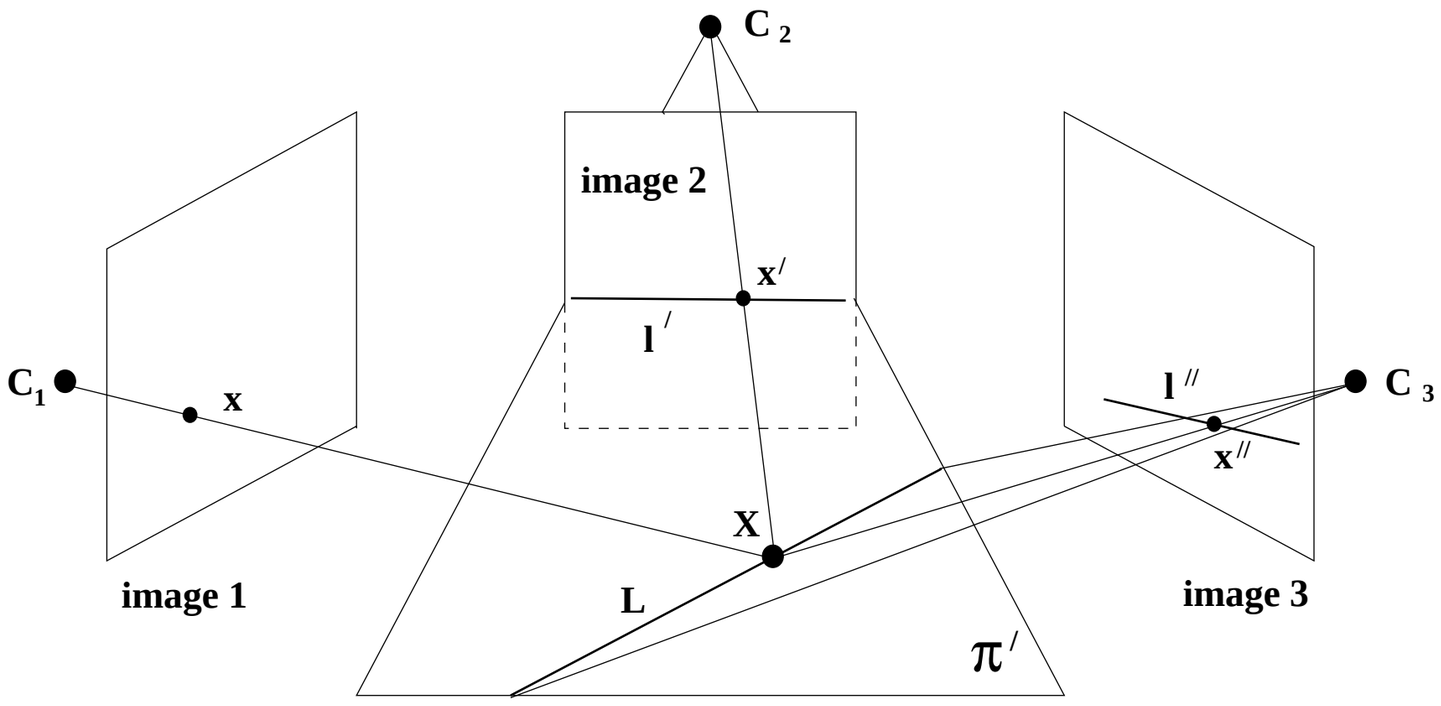


Three-point correspondence

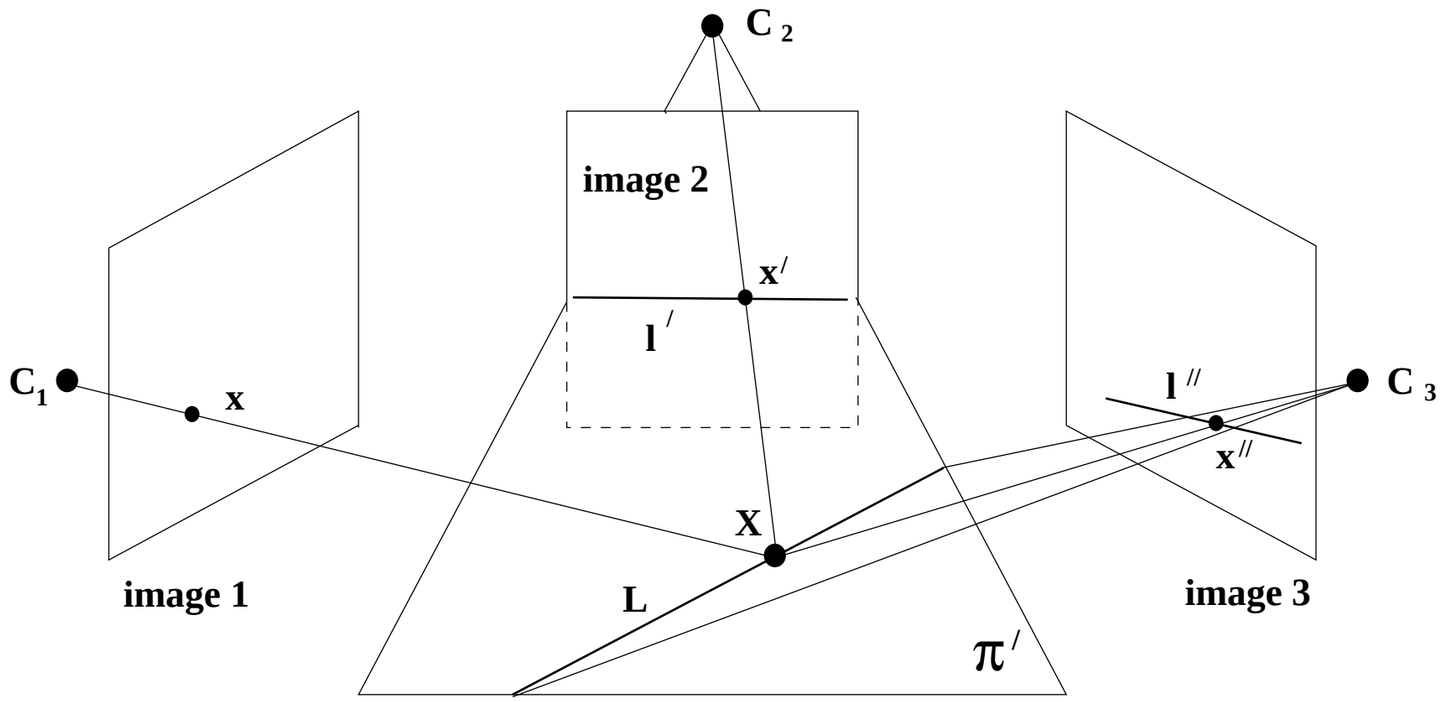
Given a triple correspondence $\mathbf{x} \leftrightarrow \mathbf{x}' \leftrightarrow \mathbf{x}''$

- Choose any lines $\mathbf{l}'$ and $\mathbf{l}''$ passing through $\mathbf{x}'$ and $\mathbf{x}''$
- Trifocal condition holds

$$x^i \, l'_j \, l'_k \, T_i^{jk} = 0$$



Geometry of the three-point correspondence



- 4 choices of lines $\Rightarrow$ 4 equations.
- May also be written as

$$x^i x'^r \epsilon_{rsi} x''^t \epsilon_{tuj} T_i^{jk} = 0$$

- Gives 9 equations, only 4 linearly independent.

Summary of transfer formulas

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

$$x''^k = x^i \, l'_j \, T_i^{jk}$$

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

$$x'^j = x^i \, l''_k \, T_i^{jk}$$

(iii) Line transfer from third to first view via a plane in the second; or, from second to first view via a plane in the third.

$$l_i = l'_j \, l''_k \, T_i^{jk}$$

Summary of incidence relations

(i) Point in first view, lines in second and third

$$l'_j \, l''_k \, T_i^{jk} x^i = 0$$

(ii) Point in first view, point in second and line in third

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

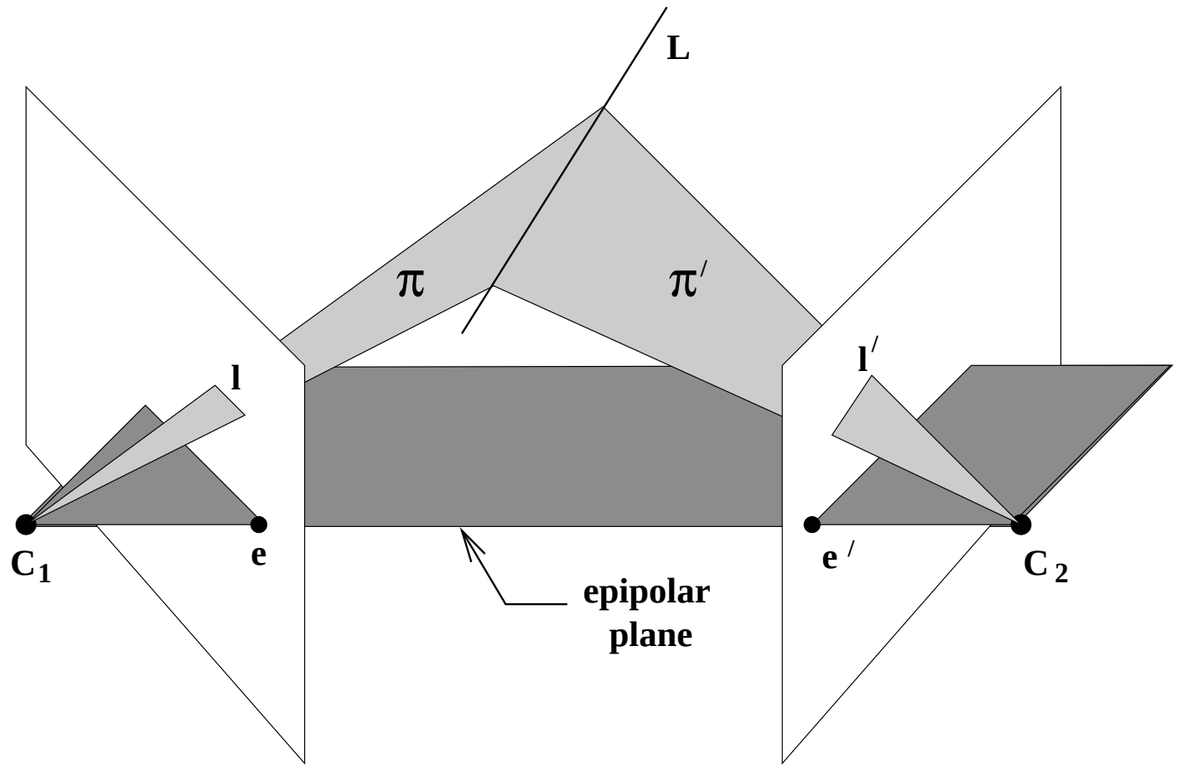
(iii) Point in first view, line in second and point in third

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

(iv) Point in three views

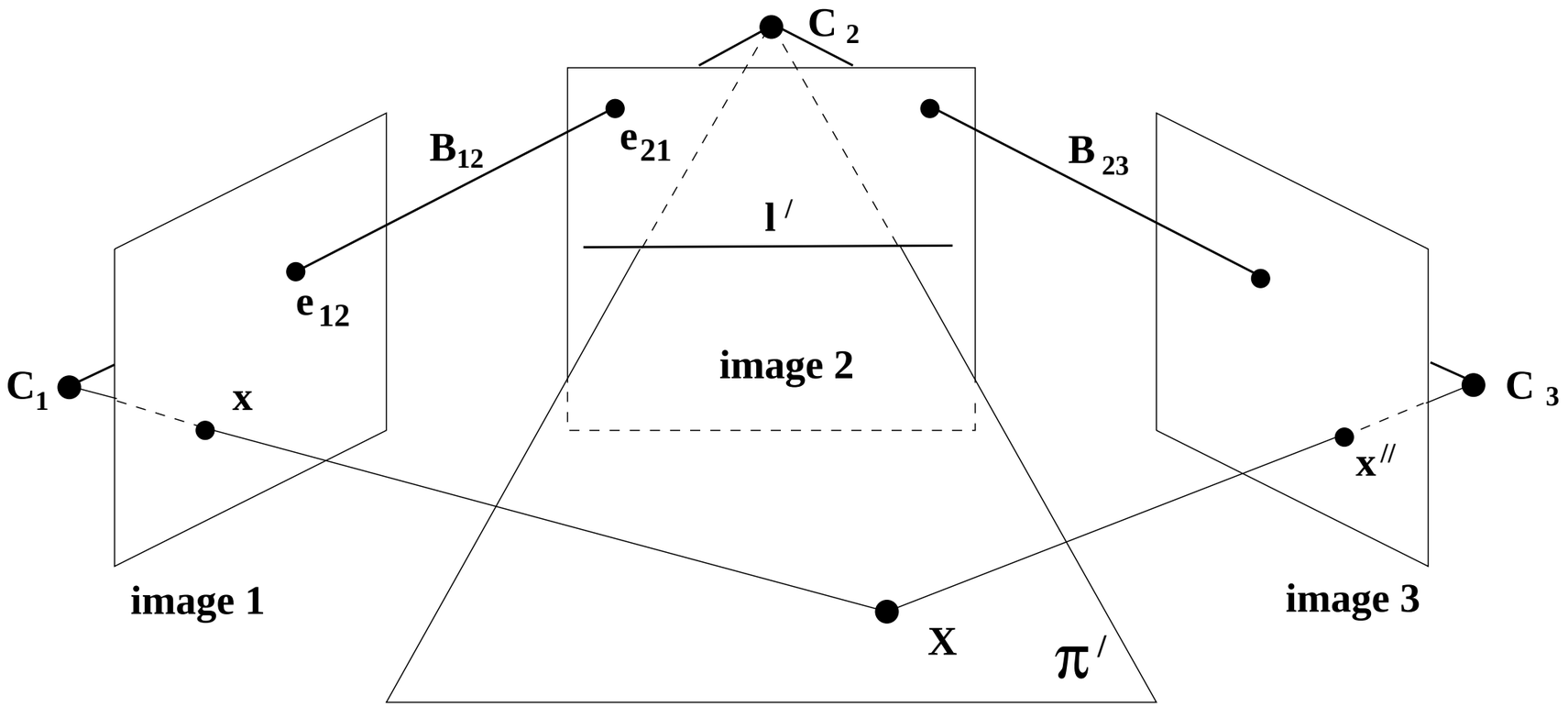
$$x^i x'^j x''^k \, \epsilon_{jpr} \epsilon_{kqs} \mathcal{T}_i^{pq} = 0_{rs}$$

Degeneracies of line transfer



- Degeneracy of line transfer for corresponding epipolar lines.
- When the line lies in an epipolar plane, its position can not be inferred from two views.
- Hence it can not be transferred to a third view.

Degeneracy of point transfer



Transferring points from images 1 and 2 to image 3 :

Only points that can not be transferred are those points on the base-line between centres of cameras 1 and 2.

For transfer with fundamental matrix, points in the trifocal plane can not be transferred.

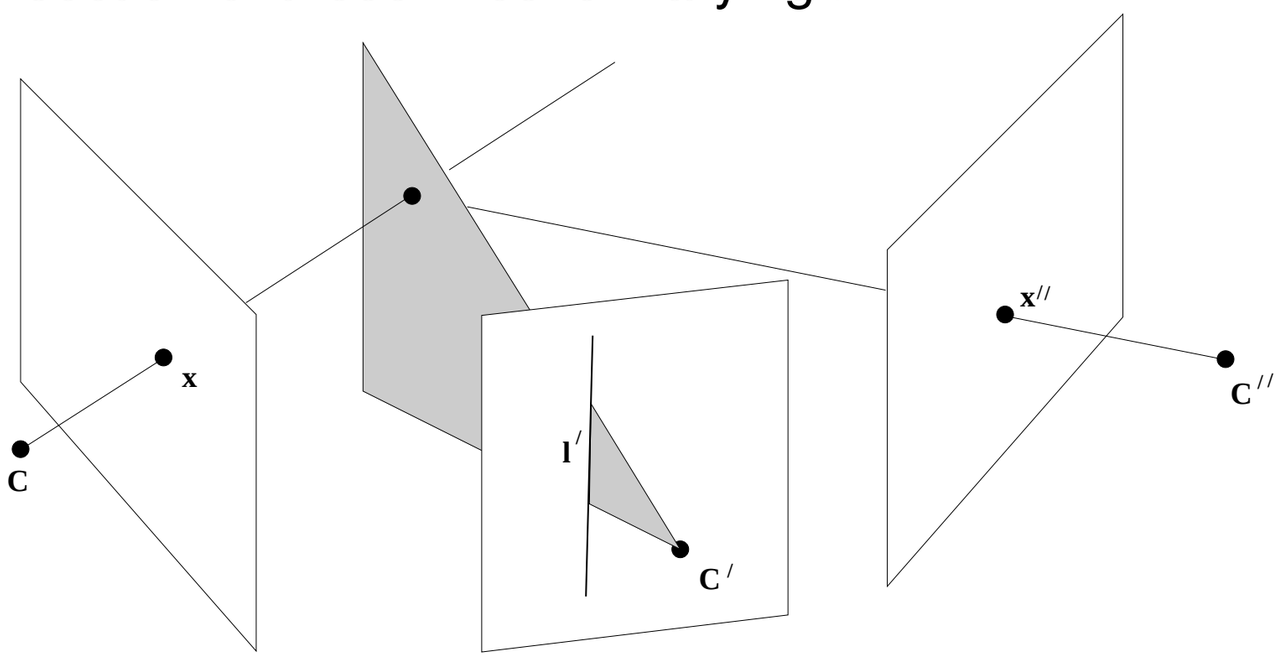
Contraction on a point

$$\text{In } x''_k \approx x^i l'_j T_i^{jk} \text{ write } x^i T_i^{jk} = G^{jk}$$

- Represents a mapping from line l''_k to the point x'^j :

$$x''^k \approx G^{jk} l'_j = (x^i T_i^{jk}) l'_j$$

- As l'_j varies x''^k traces out the projection of the ray through x^i.
- Epipolar line in third image.
- Epipole is the intersection of these lines for varying x^i.



Finding epipolar lines

To find the epipolar line corresponding to a point x^i :

- Transfer to third image via plane back-projected from l'_j

$$x''_k = x^i T_i^{jk} l'_j$$

- Epipolar line satisfies $l''_k x''^k = 0$ for each such x''_k.
- For *all* l'_j

$$x^i \, l'_j \, l''_k \, T_i^{jk} = 0$$

- Epipolar line corresponding to x^i found by solving

$$l''_k \, (x^i T_i^{jk}) = 0^j$$

Result : Epipole is the common perpendicular to the null-space of all $x^i T_i^{jk}$.

Extraction of camera matrices from trifocal tensor

Formula for Trifocal tensor

- Trifocal tensor is independent of projective transformation.
- May assume that first camera is $[\mathtt{I} \mid \mathbf{0}]$
- Other cameras are $[\mathtt{A}|\mathbf{a}_4]$ and $[\mathtt{B}|\mathbf{b}_4]$
- Formula

$$T_i^{jk} = a_i^j b_4^k - a_4^j b_i^k$$

- Note : $\mathbf{a}_4$ and $\mathbf{b}_4$ represent the epipoles :
  - Centre of the first camera is $(0, 0, 0, 1)$.
  - Epipole is image of camera centre.

$$\mathbf{a}_4 = [\mathtt{A}|\mathbf{a}_4] \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Where does this formula come from ?

Formula for trifocal tensor

$$\begin{aligned} T_i^{jk} &= \epsilon_{rsi}\mathbf{P}^r \wedge \mathbf{P}^s \wedge \mathbf{P}'^j \wedge \mathbf{P}''^k \\ &= 2\det\begin{bmatrix} \sim \mathbf{P}^i \\ \mathbf{P}'^j \\ \mathbf{P}''^k \end{bmatrix} \end{aligned}$$

Notation : $\sim \mathbf{P}^i$ means omit row i.

Example, when $i = 1$

$$\begin{aligned} T_1^{jk} &= \det\begin{bmatrix} & 1 & & \\ & & 1 & \\ a_1^j & a_2^j & a_3^j & a_4^j \\ b_1^k & b_2^k & b_3^k & b_4^k \end{bmatrix} \\ &= a_1^j b_4^k - a_4^j b_1^k \end{aligned}$$

Extraction of the camera matrices.

Basic formula

$$T_i^{jk} = a_i^j b_4^k - a_4^j b_i^k$$

- Entries of T_i^{jk} are quadratic in the entries of camera matrices.
- **But** if epipoles a_4^j and b_4^k are known, entries are linear.

Strategy :

- Estimate the epipoles.
- Solve linearly for the remaining entries of A and B.
- 27 equations in 18 unknowns.

Exact formulae are possible, but not required for practical computation.

Matrix formulas involving trifocal tensor

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

(i) **Retrieve the epipoles** $\mathbf{e}_{21}, \mathbf{e}_{31}$

Let $\mathbf{u}_i$ and $\mathbf{v}_i$ be the left and right null vectors respectively of $\mathtt{T}_i$, i.e. ${\mathtt{T}_i}^\top \mathbf{u}_i = \mathbf{0}$, $\mathtt{T}_i \mathbf{v}_i = \mathbf{0}$. Then the epipoles are obtained as the null-vectors to the following 3×3 matrices

$$[\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3]\mathbf{e}_{21} = \mathbf{0} \quad [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]\mathbf{e}_{31} = \mathbf{0}$$

(ii) **Retrieve the fundamental matrices** $\mathtt{F}_{12}, \mathtt{F}_{13}$

$$\mathtt{F}_{12} = [\mathbf{e}_{21}]_\times [\mathtt{T}_1, \mathtt{T}_2, \mathtt{T}_3]\mathbf{e}_{31} \quad \mathtt{F}_{13} = [\mathbf{e}_{31}]_\times [{\mathtt{T}_1}^\top, {\mathtt{T}_2}^\top, {\mathtt{T}_3}^\top]\mathbf{e}_{21}$$

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

Normalize the epipoles to unit norm. Then

$$\mathtt{P}' = [(\mathtt{I} - \mathbf{e}_{21}{\mathbf{e}_{21}}^\top)[\mathtt{T}_1, \mathtt{T}_2, \mathtt{T}_3]\mathbf{e}_{31} \mid \mathbf{e}_{21}] \quad \mathtt{P}'' = [-[{\mathtt{T}_1}^\top, {\mathtt{T}_2}^\top, {\mathtt{T}_3}^\top]\mathbf{e}_{21} \mid \mathbf{e}_{31}]$$

Computation of the trifocal tensor

Linear equations for the trifocal tensor

Given a 3-point correspondence

$$\mathbf{x} \leftrightarrow \mathbf{x}' \leftrightarrow \mathbf{x}''$$

The trifocal tensor relationship is

$$x^i \, x'^j \, x''^k \epsilon_{jqu} \epsilon_{krv} T_i^{qr} = 0_{uv}$$

- Relationship is linear in the entries of T.
- each correspondence gives 9 equations, 4 linearly independent.
- T has 27 entries – defined up to scale.
- 7 point correspondences give 28 equations.
- Linear or least-squares solution for the entries of T.

Summary of relations

Other point or line correspondences also yield constraints on T.

Correspondence	Relation	number of equations
three points	$x^i\, x'^j\, x''^k \epsilon_{jqu}\epsilon_{krv} T_i^{qr} = 0_{uv}$	4
two points, one line	$x^i\, x'^j\, l''_r \epsilon_{jqu} T_i^{qr} = 0_u$	2
one point, two lines	$x^i\, l'_q\, l''_r T_i^{qr} = 0$	1
three lines	$l_p\, l'_q\, l''_r \epsilon^{piw} T_i^{qr} = 0^w$	2

Trilinear Relations

Solving the equations

Given 26 equations we can solve for the 27 entries of T.

- Need 7 point correspondences
- or 13 line correspondences
- or some mixture.

Total set of equations has the form

$$\mathbf{Et} = \mathbf{0}$$

- With 26 equations find an exact solution.
- With more equations, least-squares solution.

Solving the equations ...

- Solution :
  - Take the SVD : $\mathtt{E} = \mathtt{UDV}^\top$.
  - Solution is the last column of $\mathtt{V}$ corresponding to smallest singular value).
  - Minimizes $||\mathtt{Et}||$ subject to $||\mathbf{t}|| = 1$.
- Normalization of data is *essential*.

Constraints

- T has 27 entries, defined only up to scale.
- Geometry only has 18 degrees of freedom.
  - 3 camera matrices account for $3 \times 11 = 33$ dof.
  - Invariant under 3D projective transformation (15 dof).
  - Total of 18 dof.
- T must satisfy several constraints to be a geometrically valid trifocal tensor.
- To get good results, one must take account of these constraints (cf Fundamental matrix case).

What are the constraints

Some of the constraints are easy to find.

(i) Each $T^{jk}_{\cdot}$ must have rank 2.

(ii) Their null spaces must lie in a plane.

(iii) This gives 4 constraints in all.

(iv) 4 other constraints are not so easily formulated.

Constraints through parametrization.

- Define T in terms of a set of parameters.
- Only valid Ts may be generated from parameters.

Recall formula for T:

$$T_i^{jk} = a_i^j b_4^k - a_4^j b_i^k$$

- Only valid trifocal tensors are generated by this formula.
- Parameters are the entries a_i^j and b_i^k.
- Over-parametrized : 24 parameters in all.

Algebraic Estimation of T

Similar to the algebraic method of estimation of $\mathtt{F}$.

Minimize the algebraic error $||\mathtt{Et}||$ subject to

(i) $||\mathbf{t}|| = 1$
(ii) $\mathbf{t}$ is the vector of entries of T.
(iii) T is of the form $T_i^{jk} = a_i^j b_4^k - a_4^j b_i^k$.

Difficulty is this constraint is a quadratic constraint in terms of the parameters.

Minimization knowing the epipoles

Camera matrices $[\mathtt{I} \mid \mathbf{0}]$, $[\mathtt{A}|\mathbf{a}_4]$ and $[\mathtt{B}|\mathbf{b}_4]$.

$$T_i^{jk} = a_i^j b_4^k - a_4^j b_i^k$$

As with fundamental matrix, $\mathbf{a}_4$ and $\mathbf{b}_4$ are the epipoles of the first image.

If $\mathbf{a}_4$ and $\mathbf{b}_4$ are known, then T is linear in terms of the other parameters. We may write

$$\mathbf{t} = \mathtt{G}\mathbf{p}$$

- $\mathbf{p}$ is the matrix of 18 remaining entries of camera matrices $\mathtt{A}$ and $\mathtt{B}$.
- $\mathbf{t}$ is the 27-vector of entries of T.
- $\mathtt{G}$ is a 27×18 matrix.

Minimization knowing the epipoles ...

Minimization problem

$$\text{Minimize } ||\mathtt{E}\mathbf{t}|| \text{ subject to } ||\mathbf{t}|| = 1 \ .$$

becomes

$$\text{Minimize } ||\mathtt{E}\mathtt{G}\mathbf{p}|| \text{ subject to } ||\mathtt{G}\mathbf{p}|| = 1 \ .$$

- Exactly the same problem as with the fundamental matrix.
- Linear solution using the SVD.
- Reference : Hartley – Royal Society paper.

Algebraic estimation of T

Complete algebraic estimation algorithm is

(i) Find a solution for T using the normalized linear (7-point) method

(ii) Estimated T will not satisfy constraints.

(iii) Compute the two epipoles $\mathbf{a}_4$ and $\mathbf{b}_4$.

(a) Find the left (*respectively* right) null spaces of each $T_{\cdot}^{jk}$.

(b) Epipole is the common perpendicular to the null spaces.

(iv) Reestimate T by algebraic method assuming values for the epipoles.

Iterative Algebraic Method

Find the trifocal tensor T that minimizes $||\mathtt{E}\mathbf{t}||$ subject to $||\mathbf{t}|| = 1$ and $T_i^{jk} = a_i^j b_4^k - a_4^j b_i^k$.

- **Concept :** Vary epipoles $\mathbf{a}_4$ and $\mathbf{b}_4$ to minimize the algebraic error $||\mathtt{E}\mathbf{t}'|| = ||\mathtt{EG}\mathbf{p}||$.
- **Remark :** Each choice of epipoles $\mathbf{a}_4$ and $\mathbf{b}_4$ defines a minimimum error vector $\mathtt{EG}\mathbf{p}$ as above.
- Use Levenberg-Marquardt method to minimize this error.
- Simple 6×27 minimization problem.
  - 6 inputs – the entries of the two epipoles
  - 27 outputs – the algebraic error vector $\mathtt{E}\mathbf{t}' = \mathtt{EG}\mathbf{p}$.
- Each step requires estimation of $\mathbf{p}$ using algebraic method.

Automatic Estimation of a Projective Reconstruction for a Sequence

Outline

first frame of video

(i) Projective reconstruction: 2-views, 3-views, N-views

(ii) Obtaining correspondences over N-views

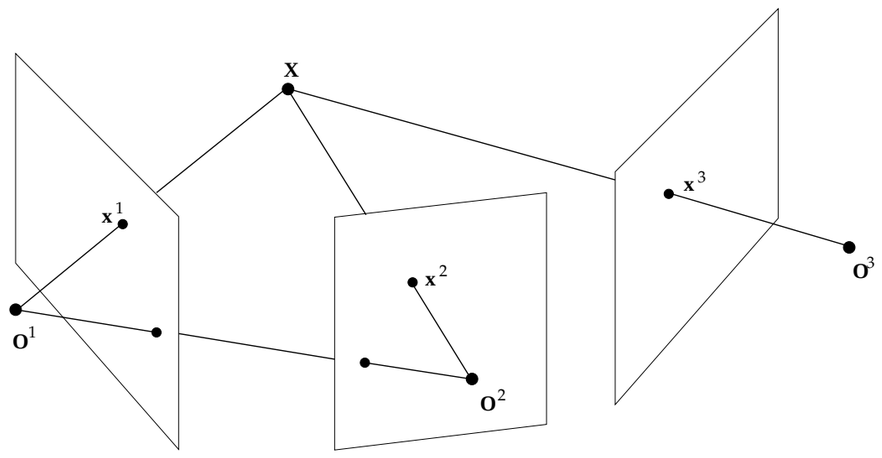
Reconstruction from three views

Given: image point correspondences $\mathbf{x}_i^1 \leftrightarrow \mathbf{x}_i^2 \leftrightarrow \mathbf{x}_i^3$,
compute a projective reconstruction:

$$\{\mathtt{P}^1,\ \mathtt{P}^2,\ \mathtt{P}^3,\ \mathbf{X}_i\} \quad \text{with} \quad \mathbf{x}_i^j \ = \ \mathtt{P}^j\ \mathbf{X}_i$$

What is new?

verify correspondences



- 3-view tensor: the trifocal tensor
- Compute from 6 image point correspondences.
- Automatic algorithm similar to F. [Torr & Zisserman]

Automatic Estimation of the trifocal tensor and correspondences

(i) Pairwise matches: Compute point matches between view pairs using robust `F` estimation.

(ii) Putative correspondences: over three views from two view matches.

(iii) RANSAC robust estimation:

Repeat

(a) Select random sample of 6 correspondences

(b) Compute T (1 or 3 solutions)

(c) Measure support (number of inliers)

Choose the T with the largest number of inliers.

(iv) MLE: re-estimate T from inlier correspondences.

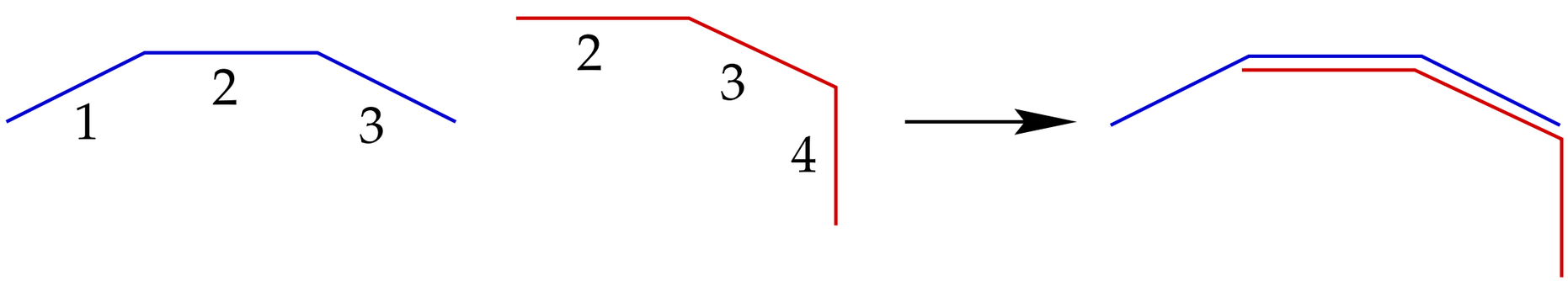
(v) Guided matching: generate additional matches.

Projective Reconstruction for a Sequence

(i) Compute all 2-view reconstructions for consecutive frames.

(ii) Compute all 3-view reconstructions for consecutive frames.

(iii) Extend to sequence by hierarchical merging:



(iv) Bundle-adjustment: minimize reprojection error

$$\min_{\mathtt{P}^j \ \mathbf{X}_i} \sum_{i \in \text{points}} \sum_{j \in \text{frames}} d\left(\mathbf{x}_i^j, \ \ \mathtt{P}^j \mathbf{X}_i\right)^2$$

(v) Automatic algorithm [Fitzgibbon & Zisserman]

Cameras and Scene Geometry for an Image Sequence

Given video

first frame of video

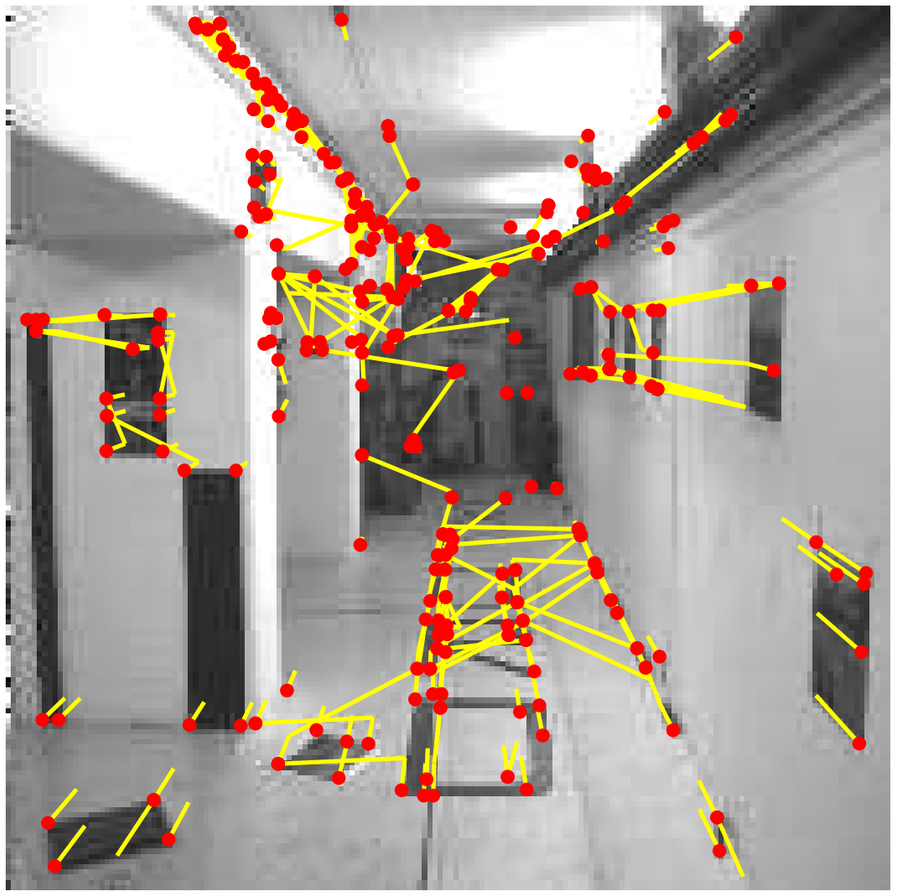
- Point correspondence (tracking).
- Projective Reconstruction.

Interest points computed for each frame

first frame of video

- About 500 points per frame

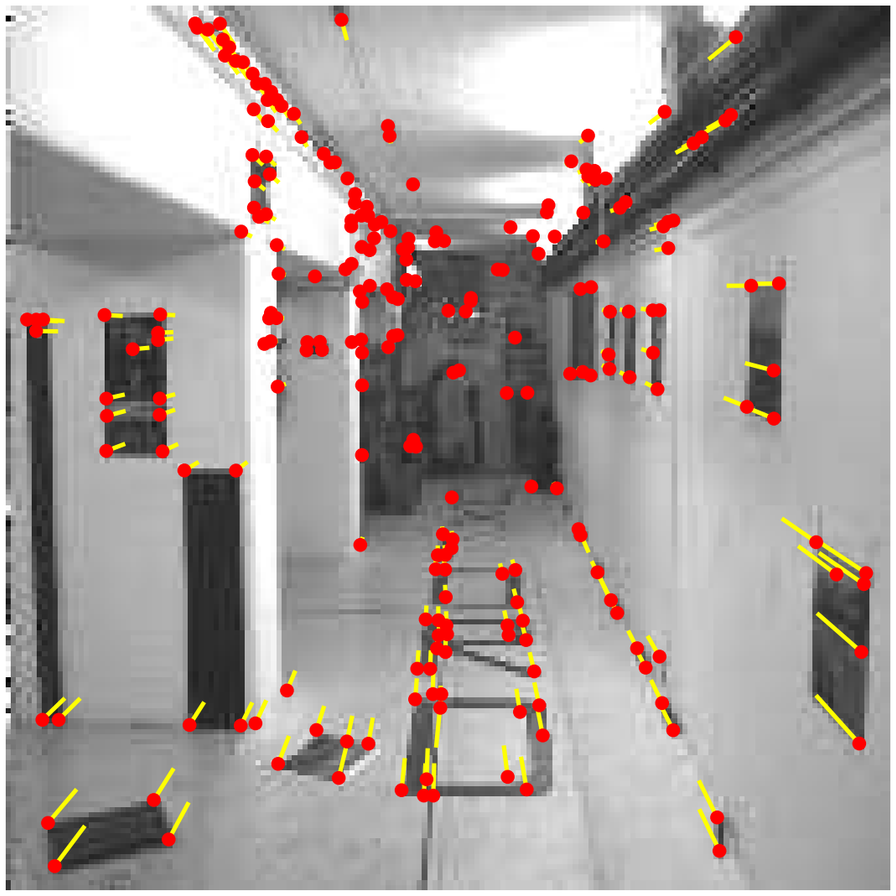
Point tracking: Correlation matching



first frame of video

- 10-50% wrong matches

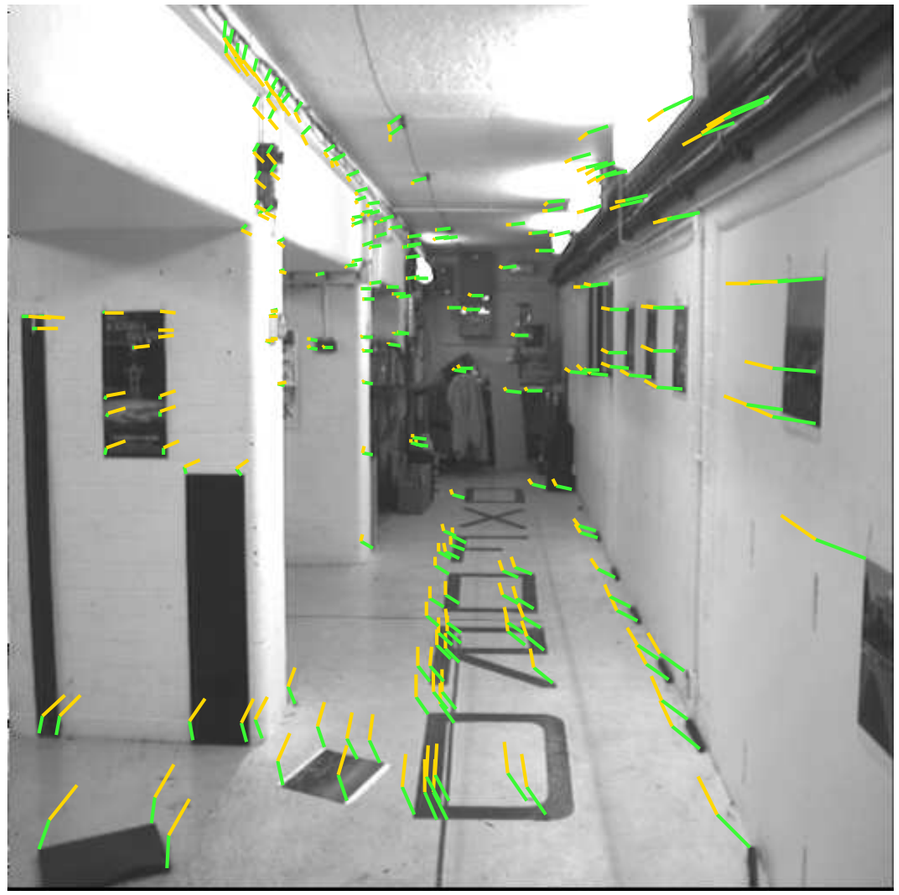
Point tracking: Epipolar-geometry guided matching



first frame of video

- Compute `F` so that matches consistent with epipolar geometry.
- Many fewer false matches, but still a loose constraint.

Point tracking: Trifocal tensor guided matching



first frame of video

- Compute trifocal tensor so that matches consistent with 3-views.
- Tighter constraint, so even fewer false matches.
- Three views is the last significant improvement.

Reconstruction from Point Tracks

Compute 3D points and cameras from point tracks

a frame of the video

- Hierarchical merging of sub-sequences.
- Bundle adjustment.

Application I: Graphical Models

Compute VRML piecewise planar model

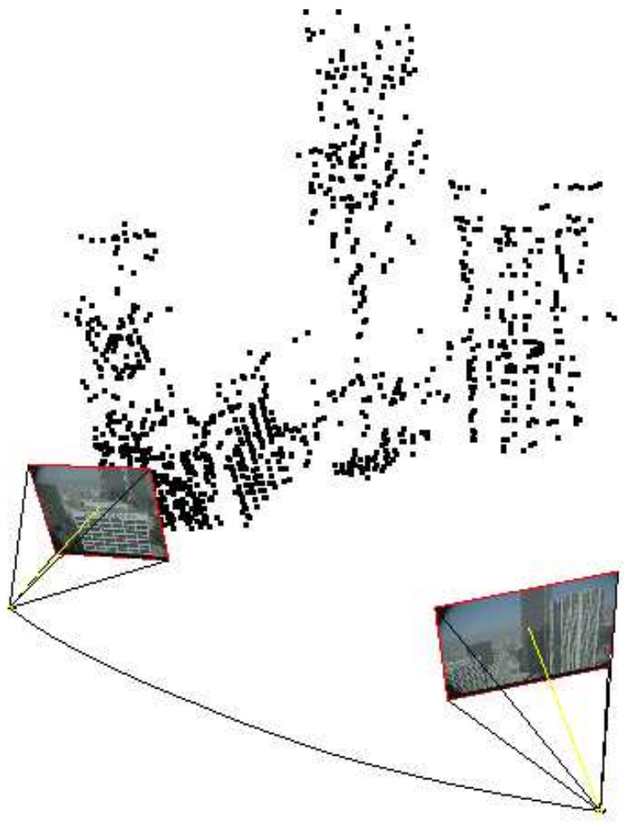
Example II: Extended Sequence

140 frames of a 340 frame sequence

a frame of the video

Metric Reconstruction

140 frames of a 340 frame sequence



a frame of the video

Application II: Augmented Reality

Using computed cameras and scene geometry,
insert virtual objects into the sequence.

a frame of the video

330 frames

3D Insertion

a frame of the video

Further Reading

Some of these papers are available from http://www.robots.ox.ac.uk/∼vgg

- Beardsley, P., Torr, P. H. S., and Zisserman, A. 3D model acquisition from extended image sequences. In B. Buxton and Cipolla R., editors, *Proc. 4th European Conference on Computer Vision, LNCS 1065, Cambridge*, pages 683–695. Springer–Verlag, 1996.
- Deriche R., Zhang Z., Luong Q. T., and Faugeras O., Robust recovery of the epipolar geometry for an uncalibrated stereo rig. In J. O. Eckland, editor, *Proc. 3rd European Conference on Computer Vision, LNCS 800/801, Stockholm*, pages 567–576. Springer-Verlag, 1994.
- Faugeras, O. “What can be seen in three dimensions with an uncalibrated stereo rig?”, *Proc. ECCV*, LNCS 588. Springer-Verlag, 1992.
- Faugeras, O., “Three-Dimensional Computer Vision”, MIT Press, 1993.
- Faugeras, O. “Stratification of three-dimensional vision: projective, affine, and metric representation”, *J. Opt. Soc. Am.*, A12:465–484, 1995.

- Fischler, M. A. and Bolles, R. C., “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography”, *CommACM*, 24, 6, 381–395, 1981.
- Fitzgibbon, A and Zisserman, A, Automatic camera recovery for closed or open image sequences. In *Proc. ECCV*, pages 311–326. Springer-Verlag, June 1998.
- Hartley, R., Gupta, R., and Chang, T. “Stereo from uncalibrated cameras”, *Proc. CVPR*, 1992.
- Hartley, R. I., In defence of the 8-point algorithm. In *ICCV5*, 1995.
- Hartley, R. I., A linear method for reconstruction from lines and points. In *Proc. ICCV*, pages 882–887, 1995.
- Hartley, R. I. and Zisserman, A., *Multiple View Geometry in Computer Vision*, Cambridge University Press, to appear.
- Kanatani, K., *Geometric Computation for Machine Vision*. Oxford University Press, Oxford, 1992.
- Longuet-Higgins H. C., A computer algorithm for reconstructing a scene from two projections. *Nature*, vol.293:133–135, 1981.

- Luong, Q. T. and Vieville, T., Canonical representations for the geometries of multiple projective views. *CVIU*, 64(2):193–229, September 1996.
- Mundy, J. and Zisserman, A., "Geometric invariance in computer vision", MIT Press, 1992. Introduction and Chapter 23 (projective geometry).
- Semple, J. and Kneebone, G. *Algebraic Projective Geometry*. Oxford University Press, 1979.
- Shashua, A. Trilinearity in visual recognition by alignment. In *Proc. ECCV*, volume 1, pages 479–484, May 1994.
- Spetsakis, M. E. and Aloimonos, J, Structure from motion using line correspondences. *IJCV*, 4(3):171–183, 1990.
- Torr P. H. S., *Outlier Detection and Motion Segmentation*. PhD thesis, University of Oxford, Engineering Dept., 1995.
- Torr, P. H. S. and Zisserman, A, Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15:591–605, 1997.

Acknowledgements

The images in this document were created by Paul Beardsley, Antonio Criminisi, Andrew Fitzgibbon, David Liebowitz, Luc Robert, and Phil Torr.