Lecture 8A

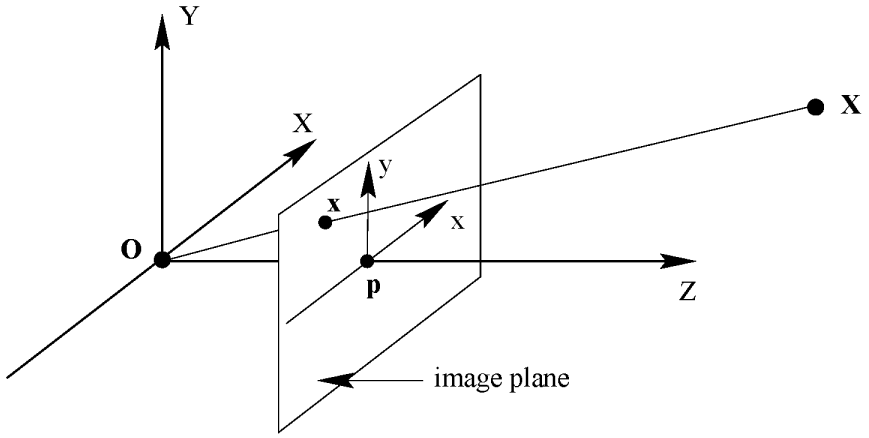
Projective Reconstruction.

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.

Concatenating the three matrices,

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

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

$$\mathbf{x} = \mathtt{P}\mathbf{X} \qquad \mathtt{P} = \mathtt{K}\,[\mathtt{R}|\ \mathbf{t}] = \mathtt{K}\mathtt{R}[\mathtt{I}|\ \mathtt{R}^\top \mathbf{t}]$$

Concatenating the three matrices,

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

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

$$\mathbf{x} = \mathtt{P}\mathbf{X} \qquad \mathtt{P} = \mathtt{K}[\mathtt{R}|\ \mathbf{t}] = \mathtt{K}\mathtt{R}[\mathtt{I}|\ \mathtt{R}^\top \mathbf{t}]$$

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.

  (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.

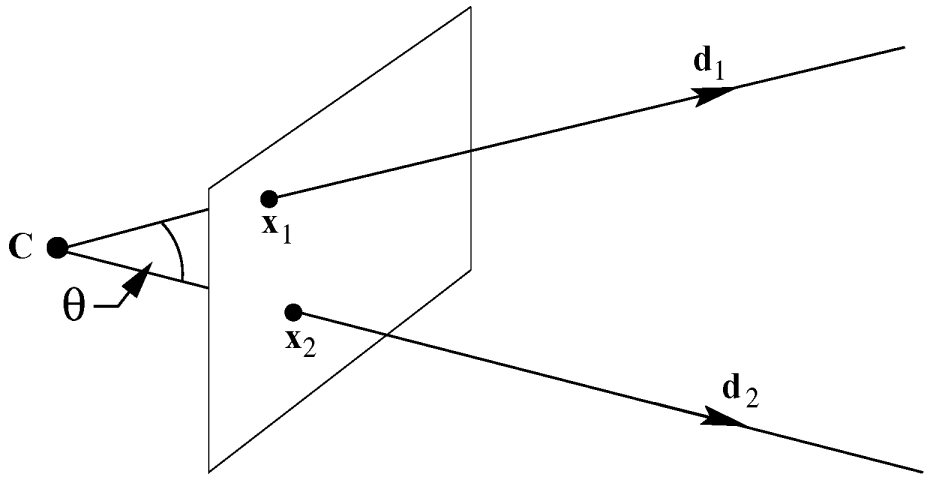
What does calibration give?

- $\mathtt{K}$ 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

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



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

$$= \frac{\mathbf{x}_1{}^\top\omega\mathbf{x}_2}{(\mathbf{x}_1{}^\top\omega\mathbf{x}_1)^{1/2}(\mathbf{x}_2{}^\top\omega\mathbf{x}_2)^{1/2}}$$

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

What does calibration give us.

(i) Direction to points in the world.

Suppose:

(i) We have an RGB-D camera. Every pixel has a known "depth".

(ii) World and camera coordinate systems are aligned.

(a) Camera is located at origin $(0, 0, 0)^\top$.

(b) Viewing along Z axis, world X and Z axis aligned with image axes.

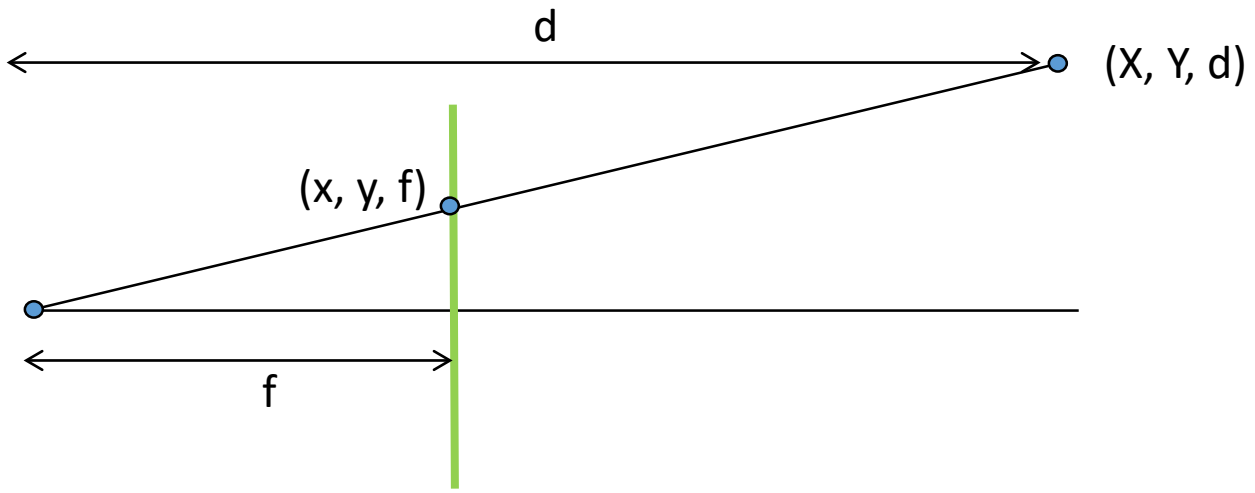
(iii) Calibration matrix is

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

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



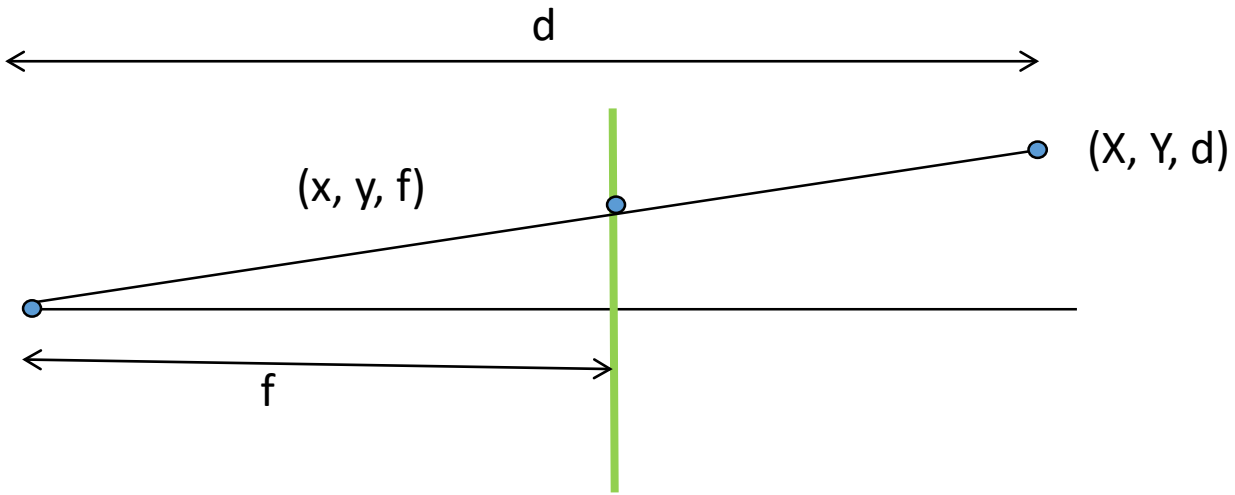
Where is a point with image (x, y, d) located?

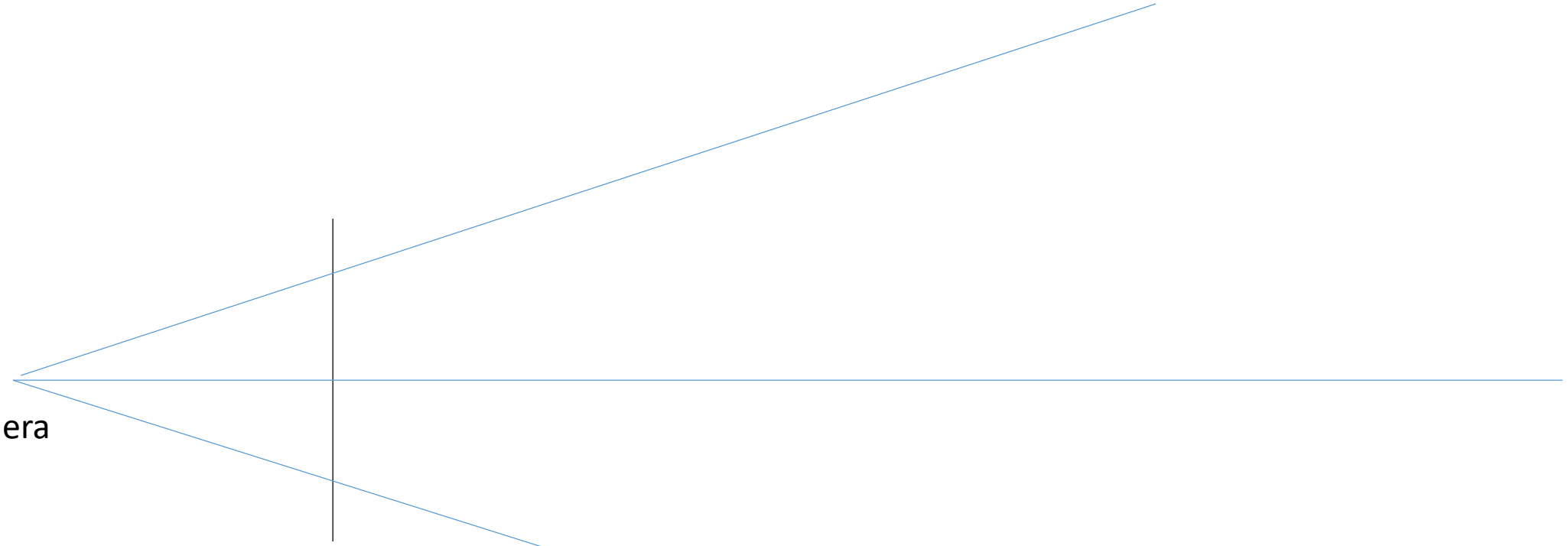
What happens if f is replaced with αf.

Then conclude point is located at point

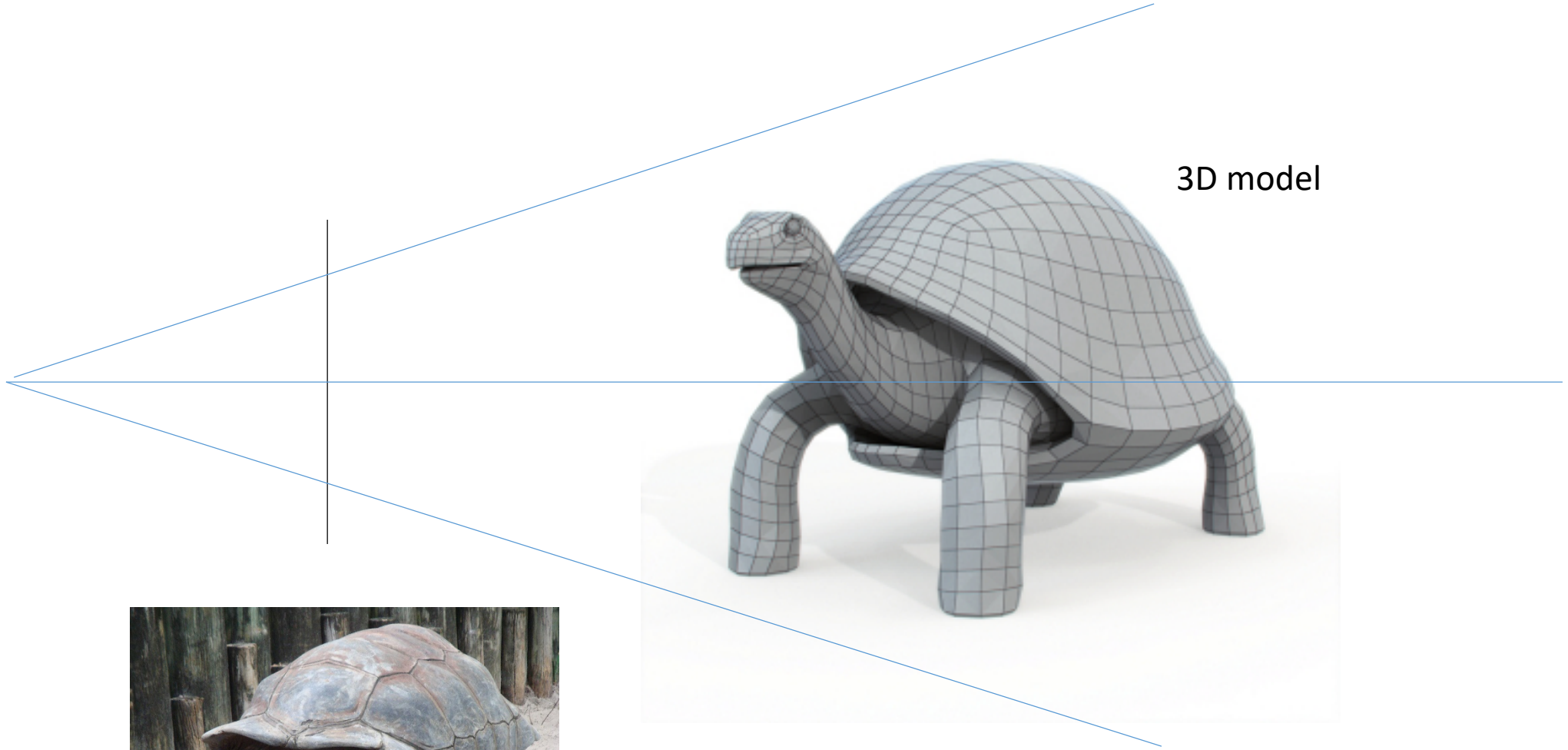
$$\left(\frac{dx}{\alpha f}, \frac{dy}{\alpha f}, d\right) = \left(\frac{\mathrm{X}}{\alpha}, \frac{\mathrm{Y}}{\alpha}, \mathrm{Z}\right)$$

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



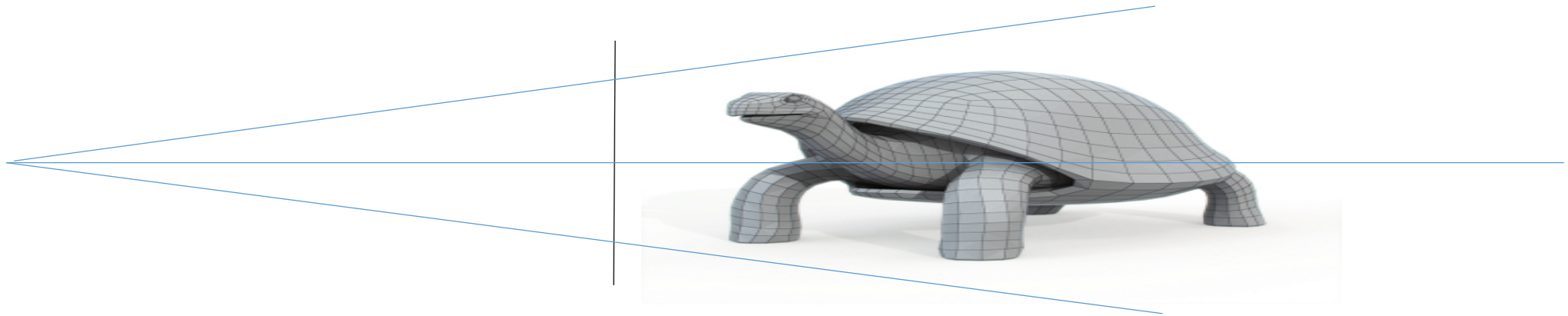


Image

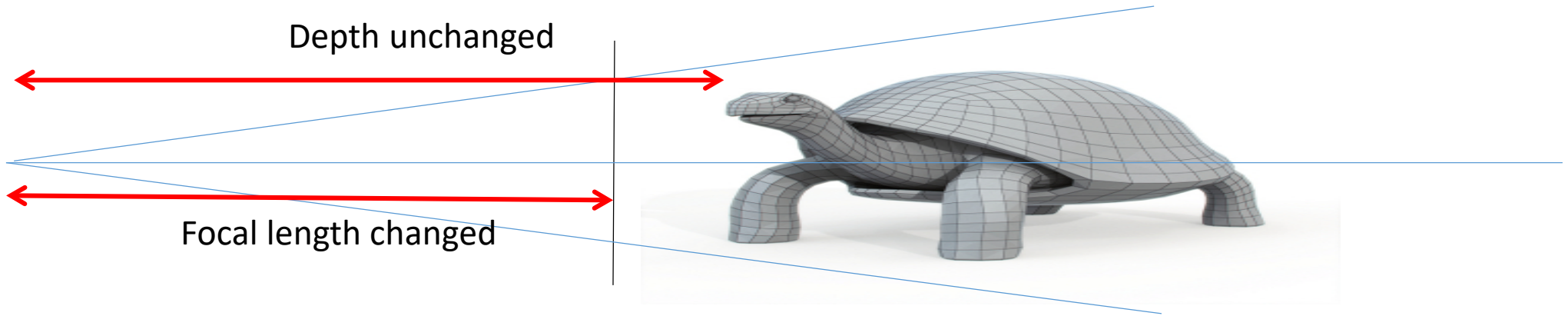


Image

Effect of wrong focal-length

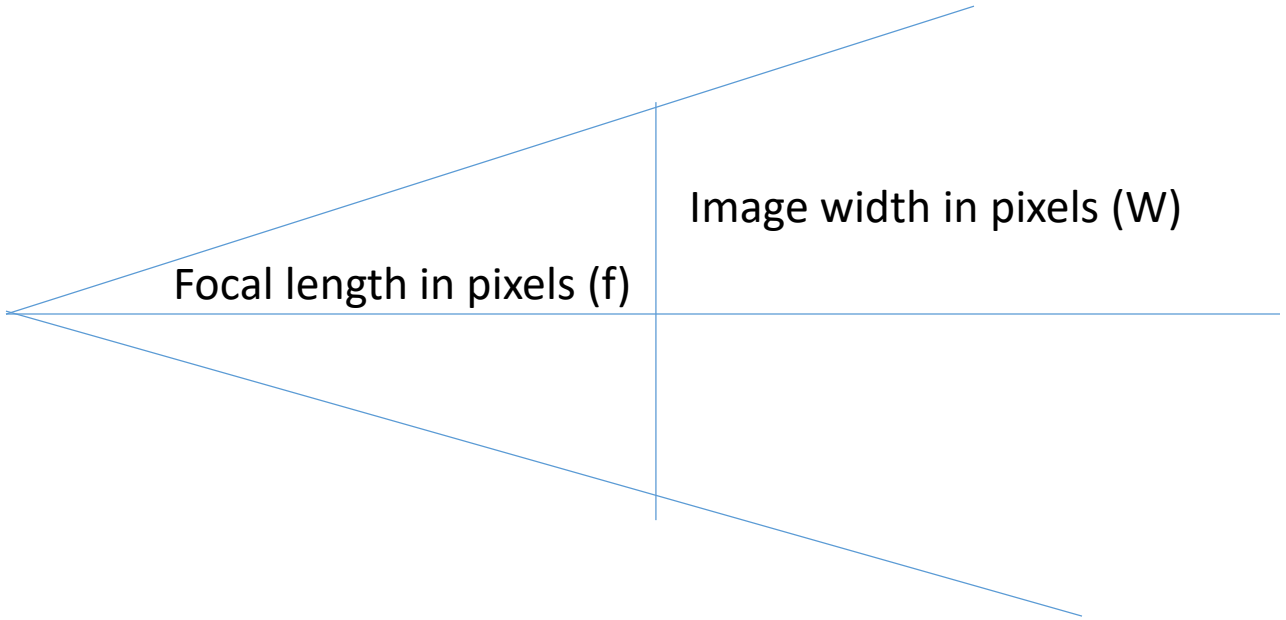


Effect of wrong focal-length



Guessing the calibration matrix

- Principal point (x_0, y_0) is usually near the centre of the image.
- Guess the field of view, theta
- Compute the focal length, f
- Note width of the image (W)



$(W/2)/ f = \tan(\text{theta}/2)$

$f = W / (2 \tan(\text{theta}/2))$

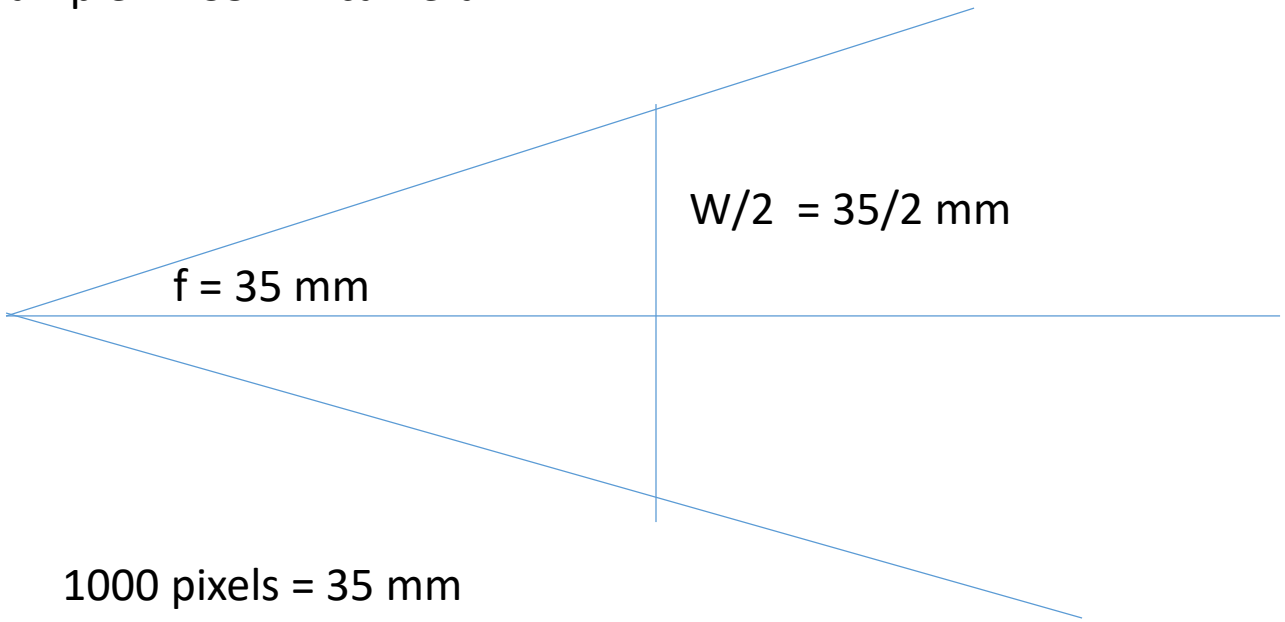
Example:

W = 1000 pixels
Theta = 60 degrees

F = 500 / tan(30)
= 866

$$K = \begin{bmatrix} 866 & 0 & 500 \\ 0 & 866 & 500 \\ 0 & 0 & 1 \end{bmatrix}$$

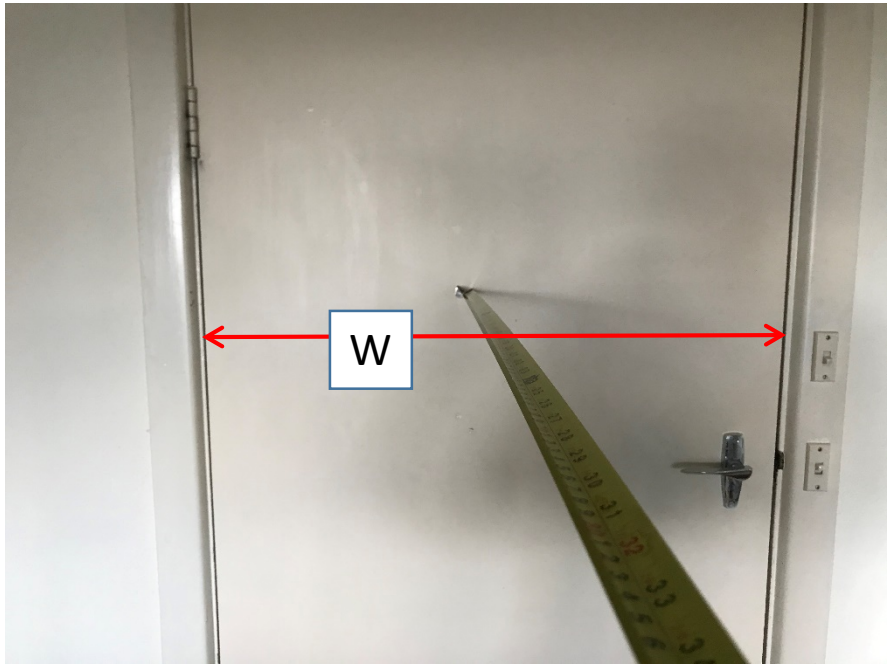
Example 2: 35mm camera

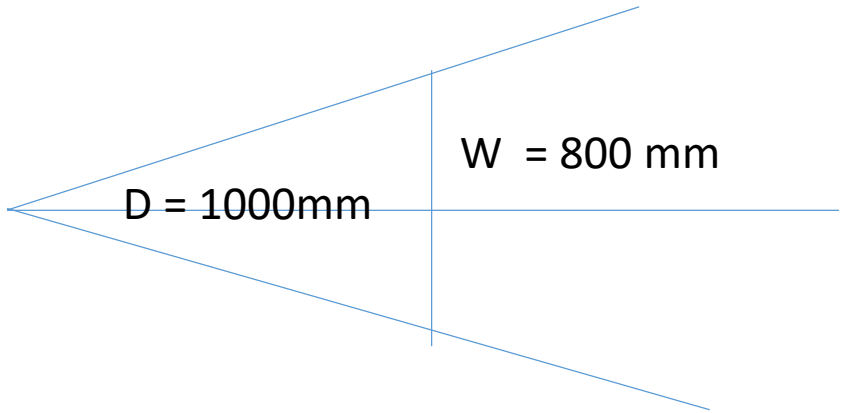


1000 pixels = 35 mm
f = 35mm = 1000 pixels.

$$K = \begin{bmatrix} 1000 & 0 & 500 \\ 0 & 1000 & 500 \\ 0 & 0 & 1 \end{bmatrix}$$

Quick and dirty calibration.





Width of door = W = 800mm
Width of door in image = w = 2666 pixels
Image size = 4032 x 3024
Distance to door = D = 1000mm
Field of view of door = theta = 2 arctan(400/1000)

W / D = w / f

800 / 1000 = 2666 / f
f = 3333

$$K = \begin{bmatrix} 3333 & 0 & 2016 \\ 0 & 3333 & 1512 \\ 0 & 0 & 1 \end{bmatrix}$$

Conversely, knowing the size of an object, and the camera calibration, I can work out how far I am from it.

More about camera models, and reconstruction

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:

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

- 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.

Thomasi-Kanade factorization algorithm.

Factorization algorithm – Affine cameras

Consider an affine camera

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

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

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

In non-homogeneous coordinates

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

Factorization algorithm – Affine cameras

Suppose that we have n affine images of m points:

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

where

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

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

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

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

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

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

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,

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

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

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

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

Matrix equation

Write as one matrix equation:

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

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

Low-rank matrix factorization.

Matrix equation

Write as one matrix equation:

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

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

Low-rank matrix factorization.

Matrix equation

Write as one matrix equation:

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

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

Low-rank matrix factorization.

Low rank matrix factorization

Given

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

Compute SVD:

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

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

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

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

This gives the desired factorization

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

Low rank matrix factorization

Given

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

Compute SVD:

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

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

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

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

This gives the desired factorization

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

Low rank matrix factorization

Given

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

Compute SVD:

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

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

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

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

This gives the desired factorization

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

Ambiguity

If

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

is one solution, then

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

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

Replace

(i) each projection $\mathbf{A}_i$ by $\mathbf{A}_i\mathbf{T}^{-1}$

(ii) each point $\mathbf{X}_j$ by $\mathbf{T}\mathbf{X}_j$.

Solution is only unique up to an affine transformation.

Ambiguity

If

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

is one solution, then

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

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

Replace

(i) each projection $\mathbf{A}_i$ by $\mathbf{A}_i\mathbf{T}^{-1}$

(ii) each point $\mathbf{X}_j$ by $\mathbf{T}\mathbf{X}_j$.

Solution is only unique up to an affine transformation.

Ambiguity

If

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

is one solution, then

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

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

Replace

(i) each projection $\mathbf{A}_i$ by $\mathbf{A}_i \mathbf{T}^{-1}$

(ii) each point $\mathbf{X}_j$ by $\mathbf{T}\mathbf{X}_j$.

Solution is only unique up to an affine transformation.

Objective

Given $n \geq 4$ image point correspondences over m views $\mathbf{x}_j^i$, $j = 1, \ldots, n$; $i = 1, \ldots, m$, determine affine camera matrices $\{\mathtt{M}^i, \mathbf{t}^i\}$ and 3D points $\{\mathbf{X}_j\}$ such that the reprojection error

$$\sum_{ij} ||\mathbf{x}_j^i - (\mathtt{M}^i \mathbf{X}_j + \mathbf{t}^i)||^2$$

is minimized over $\{\mathtt{M}^i, \mathbf{t}^i, \mathbf{X}_j\}$, with $\mathtt{M}^i$ a 2×3 matrix, $\mathbf{X}_j$ a 3-vector, and $\mathbf{x}_j^i = (x_j^i, y_j^i)^\mathsf{T}$ and $\mathbf{t}^i$ are 2-vectors.

Algorithm

(i) **Computation of translations.** Each translation $\mathbf{t}^i$ is computed as the centroid of points in image i, namely

$$\mathbf{t}^i = \langle \mathbf{x}^i \rangle = \frac{1}{n} \sum_j \mathbf{x}_j^i.$$

(ii) **Centre the data**. Centre the points in each image by expressing their coordinates with respect to the centroid:

$$\mathbf{x}_j^i \leftarrow \mathbf{x}_j^i - \langle \mathbf{x}^i \rangle.$$

Henceforth work with these centred coordinates.

(iii) **Construct the** $2m \times n$ **measurement matrix** $\mathtt{W}$ from the centred data, as defined in (18.5), and compute its SVD $\mathtt{W} = \mathtt{U}\mathtt{D}\mathtt{V}^\mathsf{T}$.

(iv) Then the matrices $\mathtt{M}^i$ are obtained from the first three columns of $\mathtt{U}$ multiplied by the singular values:

$$\begin{bmatrix} \mathtt{M}^1 \\ \mathtt{M}^2 \\ \vdots \\ \mathtt{M}^m \end{bmatrix} = \begin{bmatrix} \sigma_1 \mathbf{u}_1 & \sigma_2 \mathbf{u}_2 & \sigma_3 \mathbf{u}_3 \end{bmatrix}.$$

The vectors $\mathbf{t}^i$ are as computed in step (i) and the 3D structure is read from the first three columns of $\mathtt{V}$

$$\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \ldots & \mathbf{X}_n \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix}^\mathsf{T}.$$

Algorithm 18.1. *The factorization algorithm to determine the MLE for an affine reconstruction from n image correspondences over m views (under Gaussian image noise).*

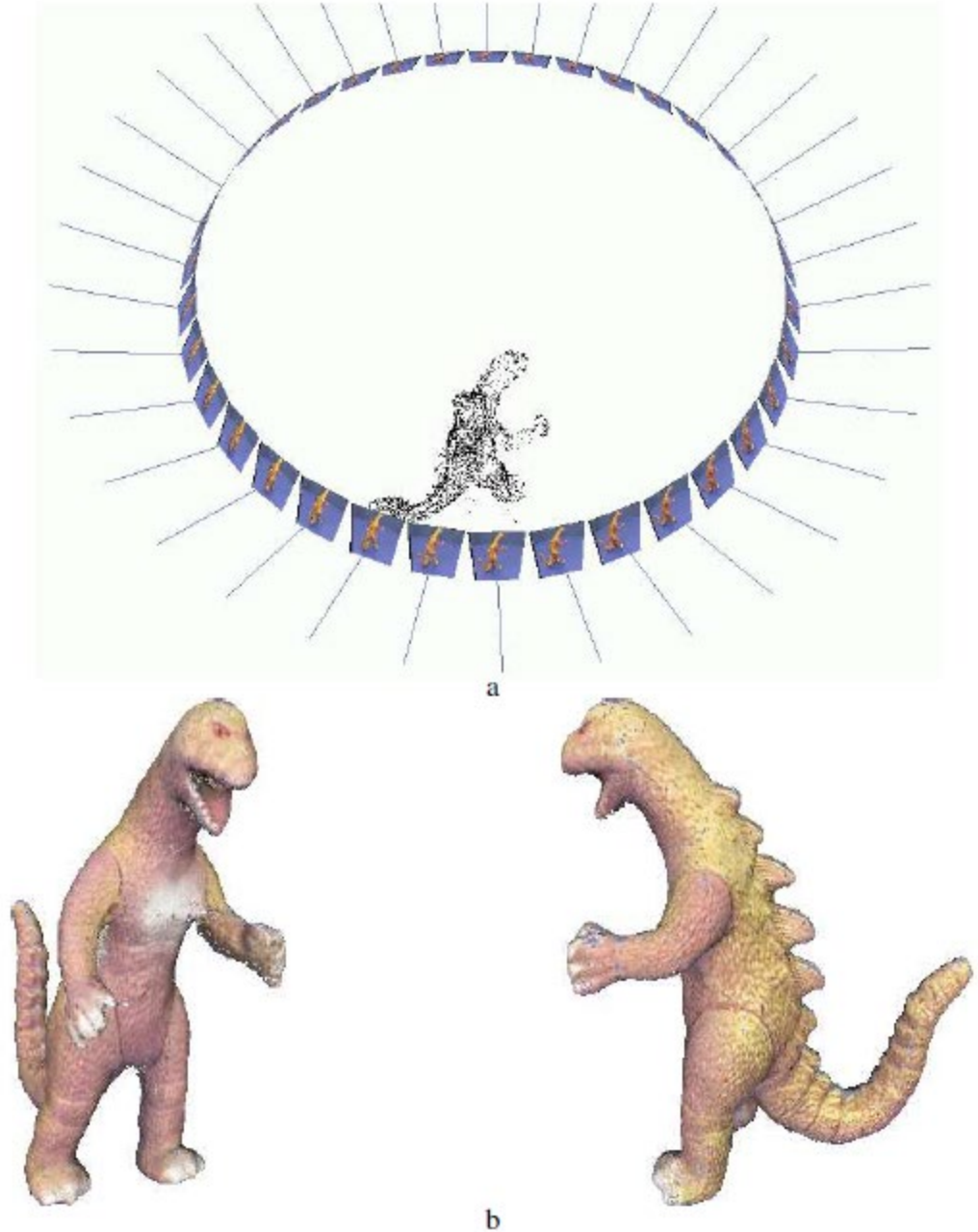


Fig. 19.8. **Dinosaur:** *(a) 3D scene points (about 5000) and camera positions for the Dinosaur sequence. (b) The automatic computation of a 3D graphical model for this sequence is described in [FCZ98].*

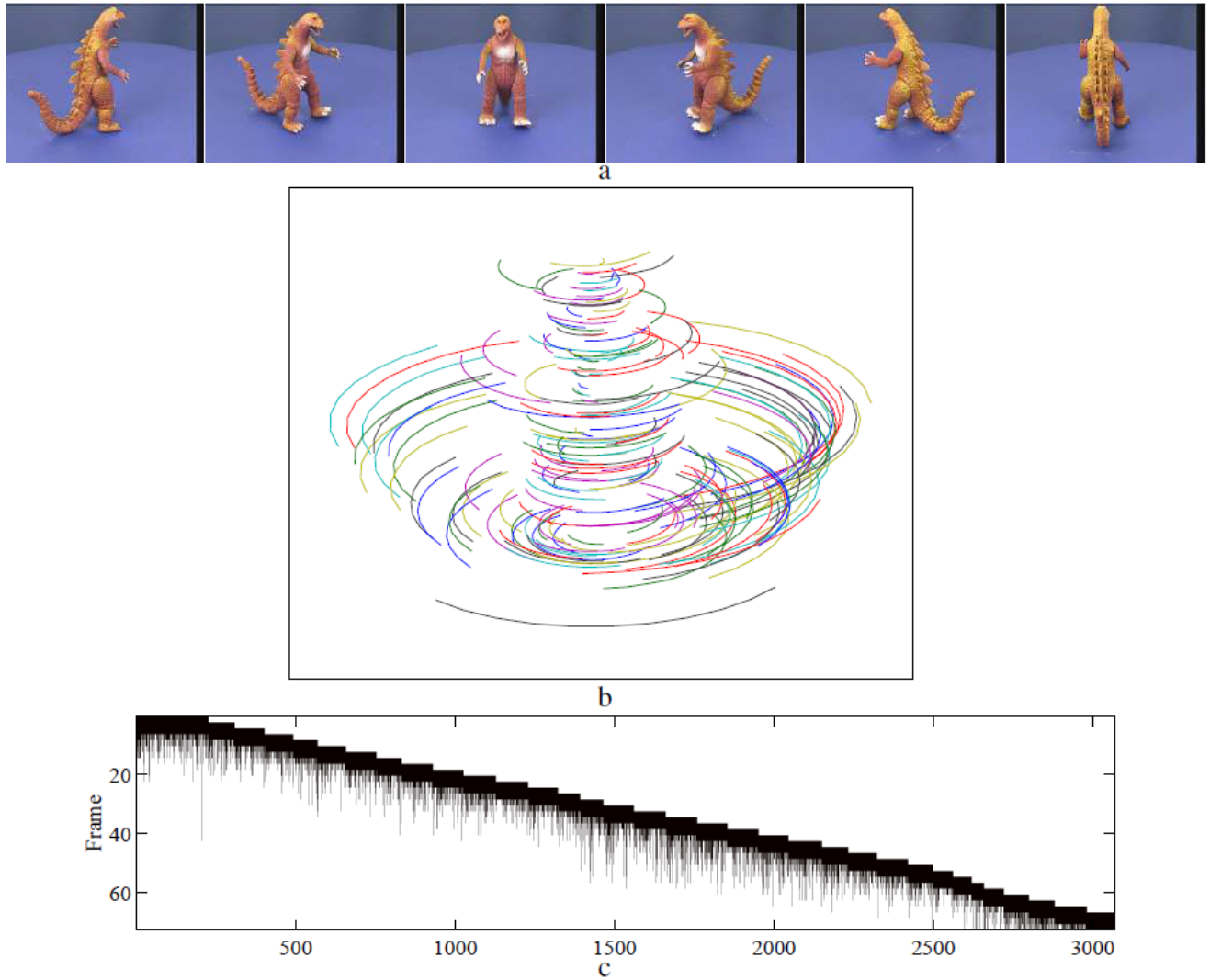


Fig. 19.7. **Dinosaur sequence and tracking:** *(a) six frames from a sequence of 36 of a dinosaur rotating on a turntable (Image sequence courtesy of the University of Hannover [NB94]). (b) A subset of the point tracks – only the 200 tracks which survived for longer than 7 successive views are shown. (c)* **Track lifetimes***: Each vertical bar corresponds to a single point track, extending from the first to last frame in which that point was seen. The horizontal ordering is according to where the point first appeared in the sequence. The measurement matrix is relatively sparse, and few points survive longer than 15 frames.*

Damped Newton Algorithms for Matrix Factorization with Missing Data

A. M. Buchanan, A. W. Fitzgibbon

IEEE Conference on Computer Vision and Pattern Recognition, Volume 2, page 316--322, 2005

Download the publication :

Damped Newton Algorithms for Matrix Factorization with Missing Data A.M.Buchanan and A.W.Fitzgibbon

Abstract

The problem of low-rank matrix factorization in the presence of missing data has seen significant attention in recent computer vision research. The approach that dominates the literature is EM-like alternation of closed-form solutions for the two factors of the matrix. An obvious alternative is non-linear optimization of both factors simultaneously, a strategy which has seen little published research. This paper provides a comprehensive comparison of the two strategies by evaluating previously published factorization algorithms as well as some second order methods not previously presented for this problem.

We conclude that, although alternation approaches can be very quick, their propensity to glacial convergence in narrow valleys of the cost function means that average-case performance is worse than second-order strategies. Further, we demonstrate the importance of two main observations: one, that schemes based on closed-form solutions alone are not suitable and that non-linear optimization strategies are faster, more accurate and provide more flexible frameworks for continued progress; and two, that basic objective functions are not adequate and that regularization priors must be incorporated, a process that is easier with non-linear methods.

BibTex reference:

```
@InProceedings{Buchanan05,
  author       = "Aeron M. Buchanan and Andrew W. Fitzgibbon",
  title        = "Damped {Newton} Algorithms for Matrix Factorization with Missing Data",
  booktitle    = "IEEE Conference on Computer Vision and Pattern Recognition",
  volume       = "2",
  pages        = "316--322",
  year         = "2005",
}
```

PowerFactorization : 3D reconstruction with missing or uncertain data

January 2003

Authors:

Richard Hartley

Frederik Schaffalitzky
17.39 · University of Oxford

Abstract and Figures

Many problems in computer vision may be considered as low-rank approximation problems, in which a matrix of mea-sured data must be approximated by a matrix of given low rank. If the matrix has no missing entries, then this is easily accomplished by a Singular Value Decomposition (SVD). If some measurements are missing however, and the matrix has holes, then the SVD method can not be applied. We present here a practical iterative method for approximating a data matrix, possibly with missing entries, with another matrix of small rank r. For a complete data matrix the method re-duces to the well-known "Power Method" which is provably convergent to a unique global optimum. If the data is well approximated by a matrix of rank r the Power Method has rapid convergence. Our method for incomplete data is ap-plied to several problems of 3D reconstruction, generalizing the Tomasi-Kanade method for orthographic cameras and the Sturm-Triggs method for projective cameras to missing and uncertain data.

Projective factorization

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

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

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

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

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

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

(vi) ...

Projective ambiguity

If

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

is one solution, then

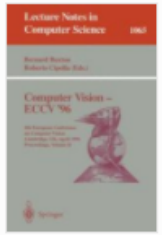
$$\mathbf{W} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_n \end{bmatrix} \boxed{\mathbf{H}^{-1}\mathbf{H}} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \dots, \mathbf{X}_m \end{bmatrix}$$

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

(i) Each projection $\mathbf{P}_i$ replaced by $\mathbf{P}_i\mathbf{H}^{-1}$

(ii) Each point $\mathbf{X}_j$ replaced by $\mathbf{H}\mathbf{X}_j$.

Solution is only unique up to a projective transformation.



European Conference on Computer Vision
ECCV 1996: Computer Vision — ECCV '96 pp 709-720 | Cite as

A factorization based algorithm for multi-image projective structure and motion

Authors | Authors and affiliations

Peter Sturm, Bill Triggs

Conference paper
First Online: 02 June 2005

Part of the Lecture Notes in Computer Science book series (LNCS, volume 1065)

Abstract

We propose a method for the recovery of projective shape and motion from multiple images of a scene by the factorization of a matrix containing the images of all points in all views. This factorization is only possible when the image points are correctly scaled. The major technical contribution of this paper is a practical method for the recovery of these scalings, using only fundamental matrices and epipoles estimated from the image data. The resulting projective reconstruction algorithm runs quickly and provides accurate reconstructions. Results are presented for simulated and real images.

Keywords

Singular Value Decomposition | Image Point | Fundamental Matrix | Fundamental Matrice | Epipolar Line

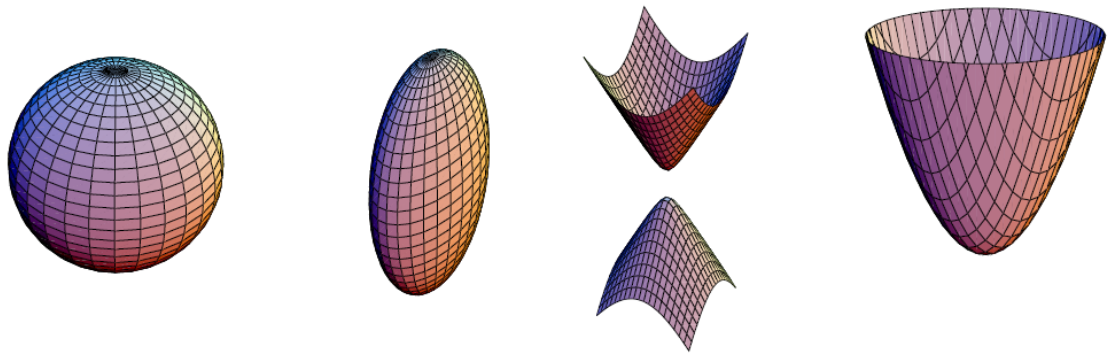


Fig. 3.2. **Non-ruled quadrics.** *This shows plots of a sphere, ellipsoid, hyperboloid of two sheets and paraboloid. They are all projectively equivalent.*

Projectively equivalent quadrics.

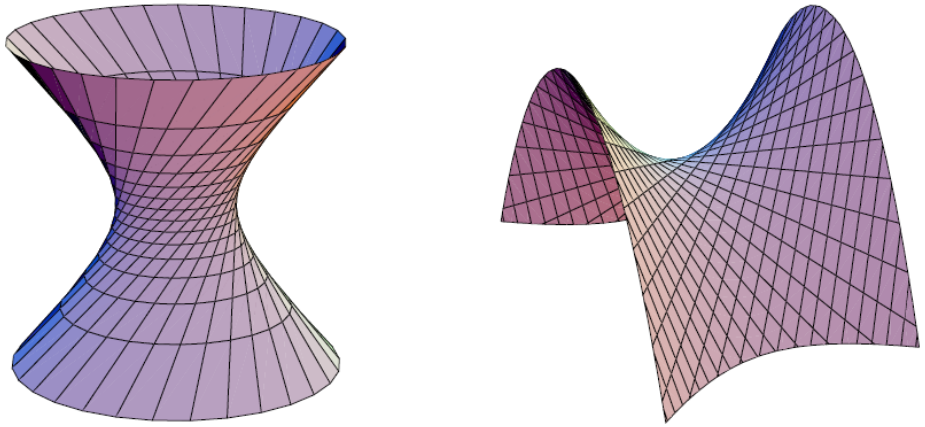


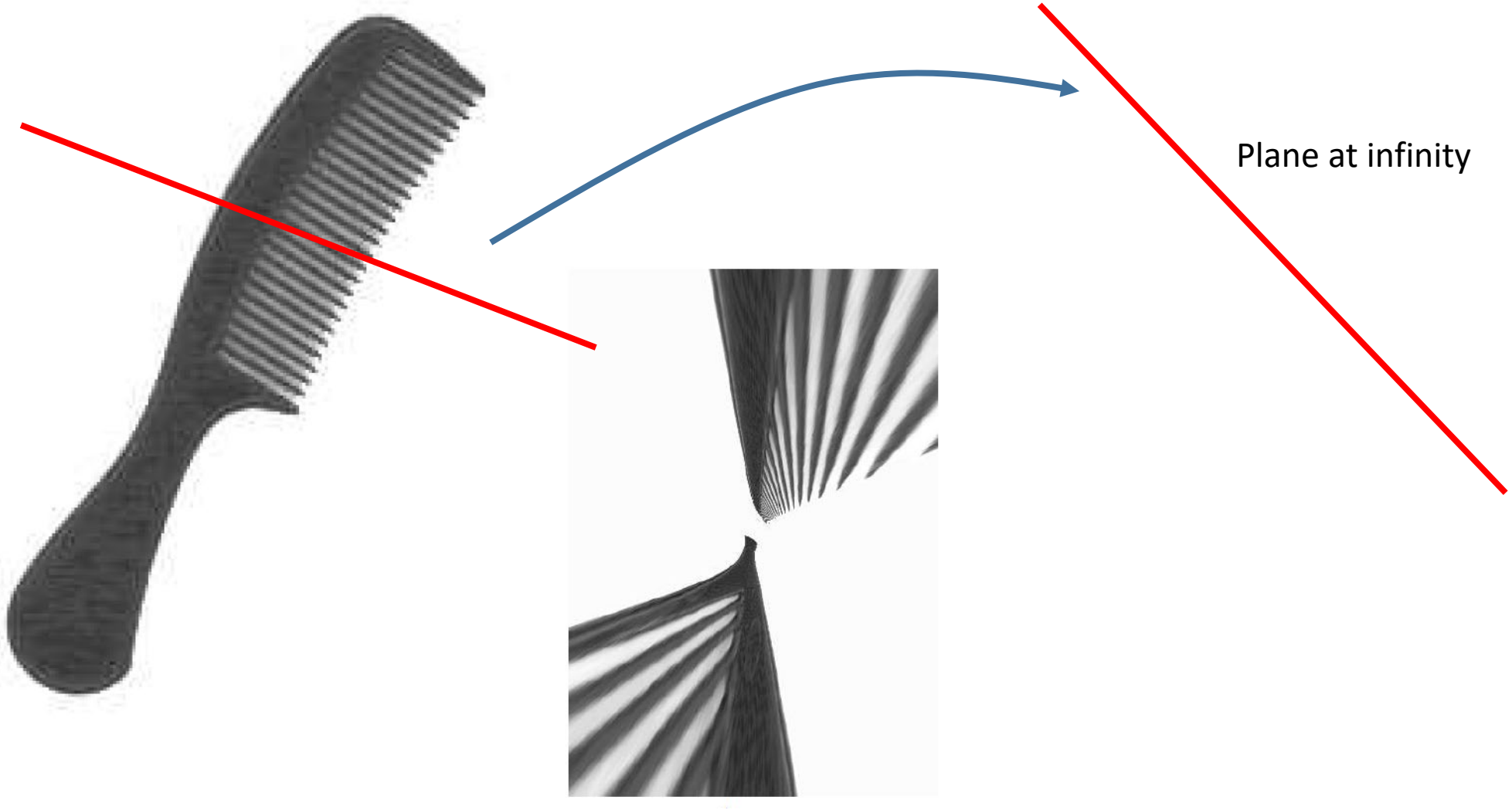
Fig. 3.3. **Ruled quadrics.** *Two examples of a hyperboloid of one sheet are given. These surfaces are given by equations* $X^2 + Y^2 = Z^2 + 1$ *and* $XY = Z$ *respectively, and are projectively equivalent. Note that these two surfaces are made up of two sets of disjoint straight lines, and that each line from one set meets each line from the other set. The two quadrics shown here are projectively (though not affinely) equivalent.*

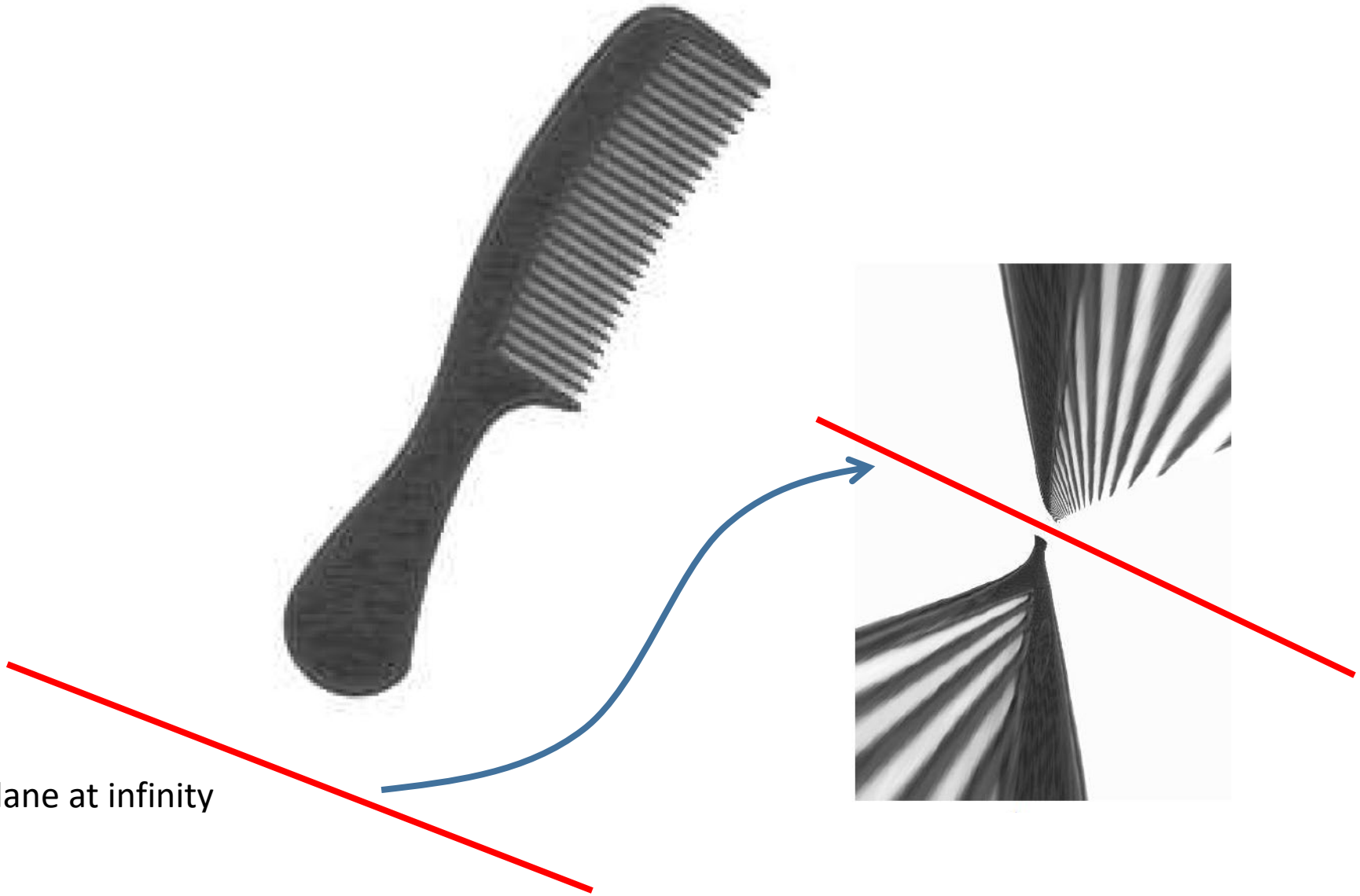
Projectively equivalent quadrics (but not equivalent to the sphere).

a b

Fig. 21.1. *(a) an image of a comb, and (b) the result of applying a projective transformation to the image. The projective transformation does not however preserve the convex hull of the set of points constituting the comb. In the original image, the convex hull of the comb is a finite set contained within the extent of the visible image. However, some of the points in this convex hull are mapped to infinity by the transformation.*

Projectively equivalent objects.

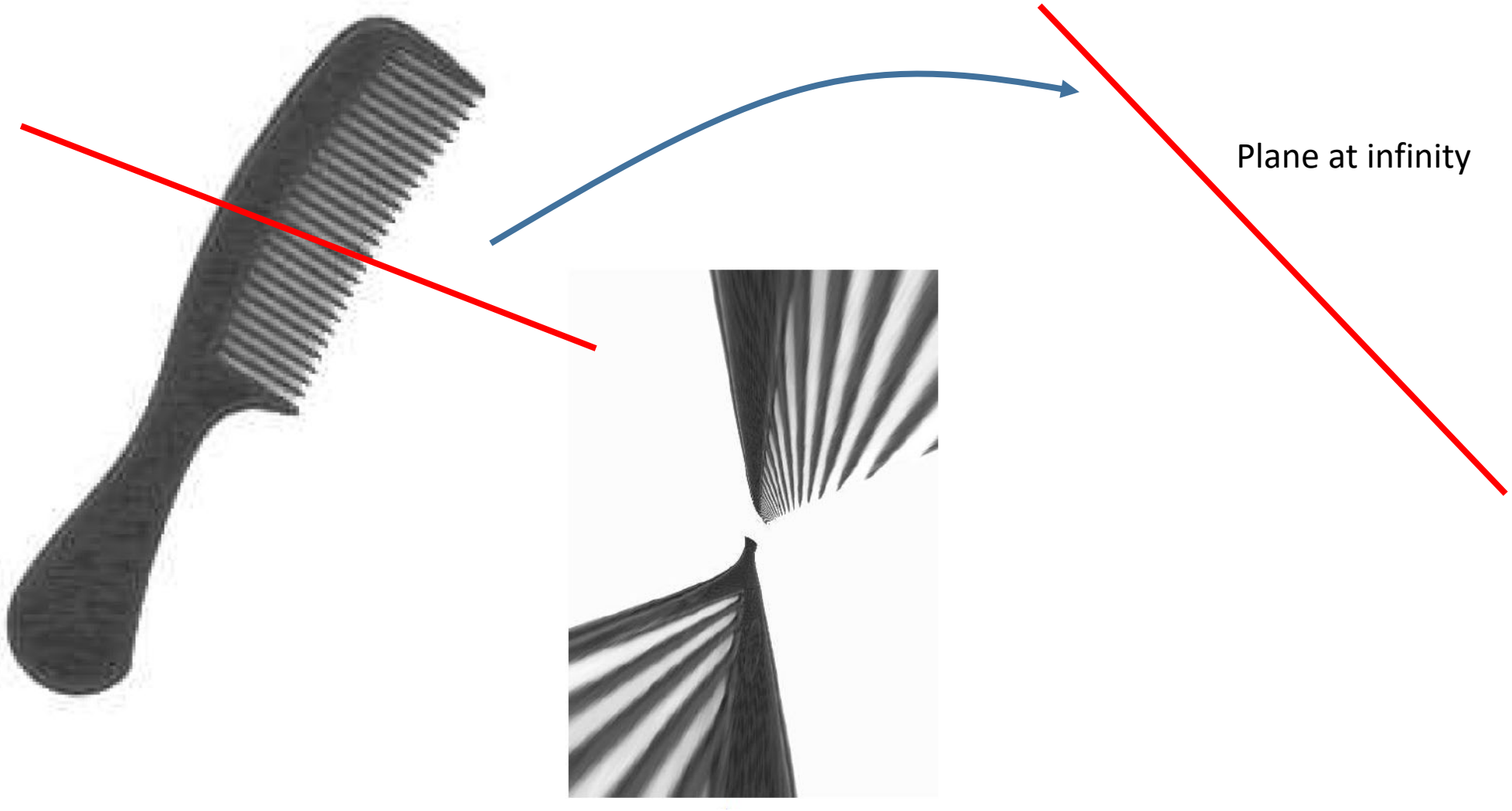

Plane at infinity


Plane at infinity

Concept: Quasi-affine transformation

A projective transformation H is “quasi-affine” with respect to a set S in R^2 (or R^3) if no point in the convex hull of the object is sent to infinity by H.

In other words, the transform does not “split” the set S across the plane at infinity.



Not a quasi-affine transformation

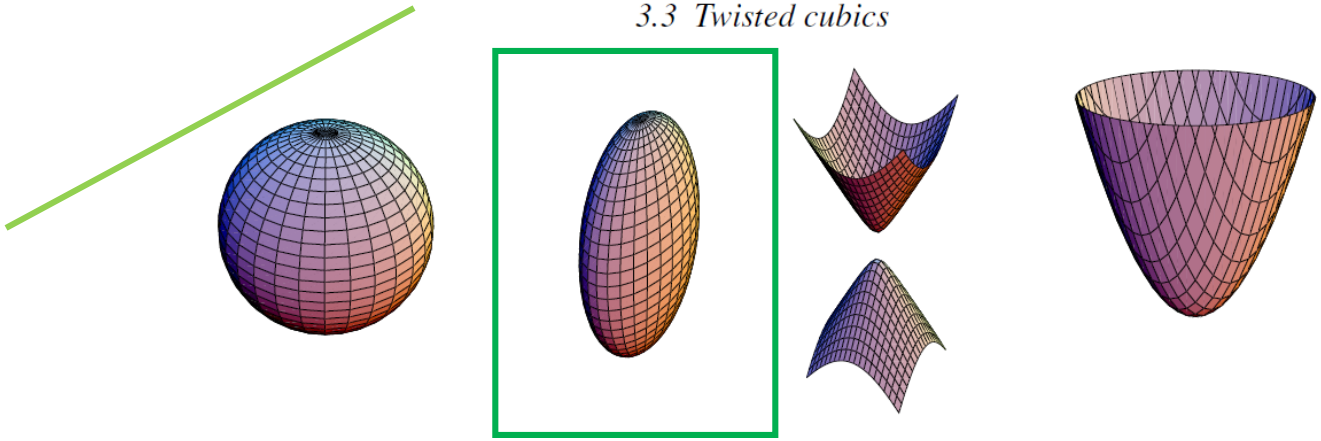


Fig. 3.2. **Non-ruled quadrics.** *This shows plots of a sphere, ellipsoid, hyperboloid of two sheets and paraboloid. They are all projectively equivalent.*

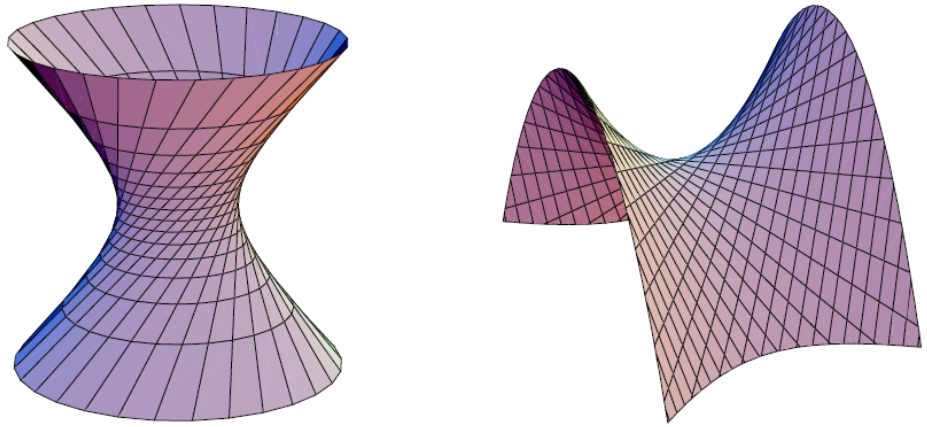


Fig. 3.3. **Ruled quadrics.** *Two examples of a hyperboloid of one sheet are given. These surfaces are given by equations* $X^2 + Y^2 = Z^2 + 1$ *and* $XY = Z$ *respectively, and are projectively equivalent. Note that these two surfaces are made up of two sets of disjoint straight lines, and that each line from one set meets each line from the other set. The two quadrics shown here are projectively (though not affinely) equivalent.*

Quasi-affine transformation is of the sphere.

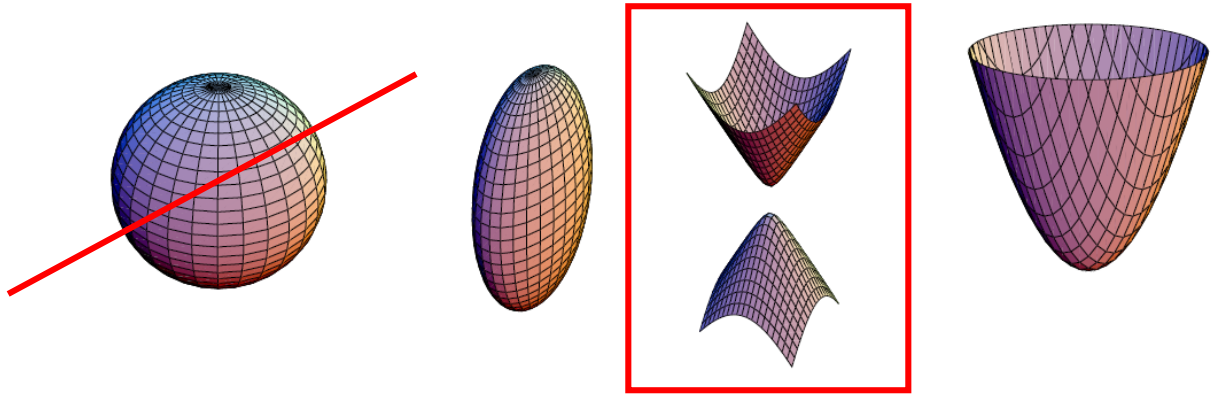


Fig. 3.2. **Non-ruled quadrics.** *This shows plots of a sphere, ellipsoid, hyperboloid of two sheets and paraboloid. They are all projectively equivalent.*

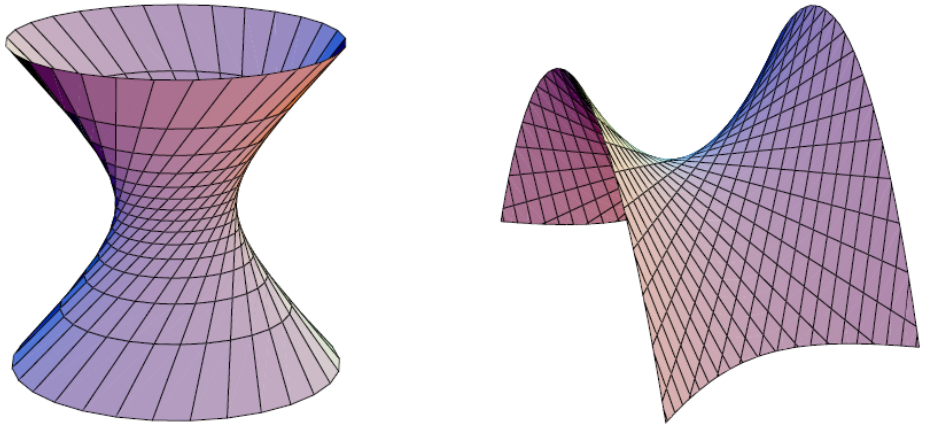


Fig. 3.3. **Ruled quadrics.** *Two examples of a hyperboloid of one sheet are given. These surfaces are given by equations* $X^2 + Y^2 = Z^2 + 1$ *and* $XY = Z$ *respectively, and are projectively equivalent. Note that these two surfaces are made up of two sets of disjoint straight lines, and that each line from one set meets each line from the other set. The two quadrics shown here are projectively (though not affinely) equivalent.*

Non-quasi-affine transformation is of the sphere.

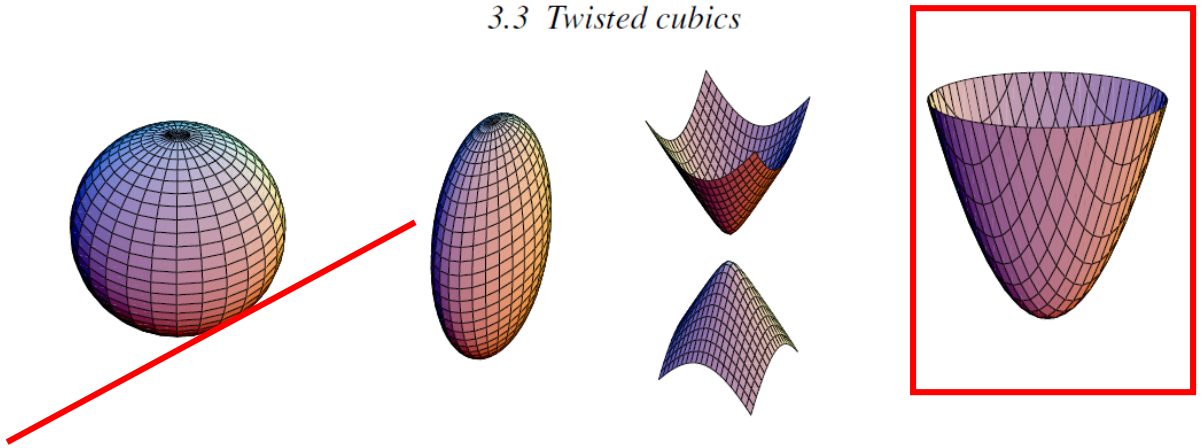


Fig. 3.2. **Non-ruled quadrics.** *This shows plots of a sphere, ellipsoid, hyperboloid of two sheets and paraboloid. They are all projectively equivalent.*

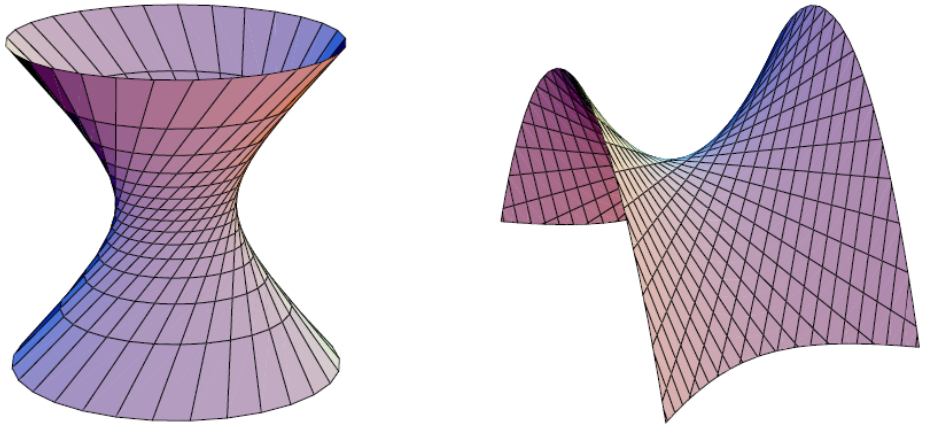


Fig. 3.3. **Ruled quadrics.** *Two examples of a hyperboloid of one sheet are given. These surfaces are given by equations* $X^2 + Y^2 = Z^2 + 1$ *and* $XY = Z$ *respectively, and are projectively equivalent. Note that these two surfaces are made up of two sets of disjoint straight lines, and that each line from one set meets each line from the other set. The two quadrics shown here are projectively (though not affinely) equivalent.*

Non-quasi-affine transformation is of the sphere.

Non-rigid factorization.

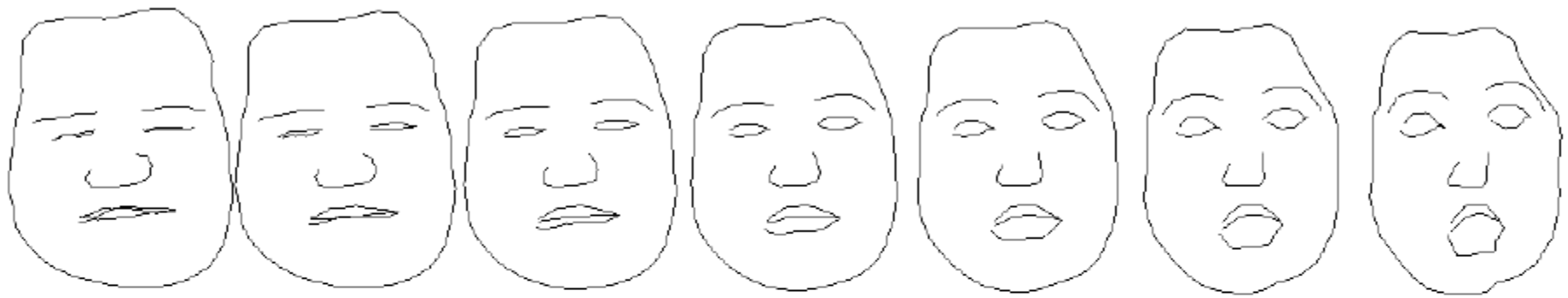


Fig. 18.1. **Shape basis**. *A face template is represented by N equally spaced 2D points (here $N = 140$). The central face of the seven is the mean shape and the faces to the left or right are generated by adding or subtracting, respectively, the basis shape that accounts for the maximum variation in the training set. In this case the basis spans expressions from surprised to disgruntled. Facial expressions are learnt by tracking the face of an actor with the template whilst he changes expression but does not vary his head pose. Each frame of the training sequence generates a set of N 2D points, and the coordinates of these are rewritten as a $2N$-vector. A $2N \times f$ matrix is then composed from these vectors, where f is the number of training frames, and the basis shapes are computed from the singular vectors of this matrix. Figure courtesy of Richard Bowden.*

$$\left[\begin{array}{cccc} \mathbf{X}_1^i & \mathbf{X}_2^i & \dots & \mathbf{X}_n^i \end{array} \right] = \sum_{k=1}^{l} \alpha_k^i \left[\begin{array}{cccc} \mathbf{B}_{1k} & \mathbf{B}_{2k} & \dots & \mathbf{B}_{nk} \end{array} \right] = \sum_k \alpha_k^i \mathbf{B}_k$$

$$\begin{bmatrix} \mathbf{X}_1^i & \mathbf{X}_2^i & \dots & \mathbf{X}_n^i \end{bmatrix} = \sum_{k=1}^{l} \alpha_k^i \begin{bmatrix} \mathbf{B}_{1k} & \mathbf{B}_{2k} & \dots & \mathbf{B}_{nk} \end{bmatrix} = \sum_k \alpha_k^i \mathrm{B}_k$$

- Basis shapes, indexed by k
- Each basis shape consists of n points, so B_nk is nth point in kth basis shape
- Several frames, indexed by i
- X^i_n is the n-th point as seen in frame I
- Shape in frame i is a linear combination with coefficients \alpha^i_k of the basis shapes.

In the forward model of image generation each view i is acquired by an affine camera and gives the image points

$$\mathbf{x}_j^i = \mathtt{M}^i \sum_k \alpha_k^i \mathbf{B}_{jk} + \mathbf{t}^i.$$

It will again be assumed that image point matches are available for all views. Our goal is to estimate cameras $\{\mathtt{M}^i, \mathbf{t}^i\}$ and 3D structure $\{\alpha_k^i, \mathbf{B}_{jk}\}$ from the measured image points $\{\mathbf{x}_j^i\}$, such that the distance between the estimated image points $\hat{\mathbf{x}}_j^i = \mathtt{M}^i \sum_k \alpha_k^i \mathbf{B}_{jk} + \mathbf{t}^i$ and measured image points is minimized

$$\min_{\mathtt{M}^i, \mathbf{t}^i, \alpha_k^i, \mathbf{B}_{jk}} \sum_{ij} \left\| \mathbf{x}_j^i - \hat{\mathbf{x}}_j^i \right\|^2 = \min_{\mathtt{M}^i, \mathbf{t}^i, \alpha_k^i, \mathbf{B}_{jk}} \sum_{ij} \left\| \mathbf{x}_j^i - (\mathtt{M}^i \sum_k \alpha_k^i \mathbf{B}_{jk} + \mathbf{t}^i) \right\|^2 .$$

As in affine factorization the translation may be eliminated by centring the measured image points, and it will be assumed from here on that this has been done. Then the problem reduces to

$$\min_{\mathtt{M}^i,\alpha_k^i,\mathbf{B}_{jk}} \sum_{ij} \left|\left| \mathbf{x}_j^i - \hat{\mathbf{x}}_j^i \right|\right|^2 = \min_{\mathtt{M}^i,\alpha_k^i,\mathbf{B}_{jk}} \sum_{ij} \left|\left| \mathbf{x}_j^i - \mathtt{M}^i \sum_k \alpha_k^i \mathbf{B}_{jk} \right|\right|^2 . \tag{18.7}$$

The complete set of equations $\hat{\mathbf{x}}_j^i = \mathtt{M}^i \sum_k \alpha_k^i \mathbf{B}_{jk}$ may be written

$$\hat{\mathtt{W}} = \begin{bmatrix} \mathtt{M}^1 \left(\alpha_1^1 \mathtt{B}_1 + \alpha_2^1 \mathtt{B}_2 + \ldots \alpha_l^1 \mathtt{B}_l\right) \\ \mathtt{M}^2 \left(\alpha_1^2 \mathtt{B}_1 + \alpha_2^2 \mathtt{B}_2 + \ldots \alpha_l^2 \mathtt{B}_l\right) \\ \vdots \\ \mathtt{M}^m \left(\alpha_1^m \mathtt{B}_1 + \alpha_2^m \mathtt{B}_2 + \ldots \alpha_l^m \mathtt{B}_l\right) \end{bmatrix} = \begin{bmatrix} \alpha_1^1 \mathtt{M}^1 & \alpha_2^1 \mathtt{M}^1 & \ldots & \alpha_l^1 \mathtt{M}^1 \\ \alpha_1^2 \mathtt{M}^2 & \alpha_2^2 \mathtt{M}^2 & \ldots & \alpha_l^2 \mathtt{M}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^m \mathtt{M}^m & \alpha_2^m \mathtt{M}^m & \ldots & \alpha_l^m \mathtt{M}^m \end{bmatrix} \begin{bmatrix} \mathtt{B}_1 \\ \mathtt{B}_2 \\ \vdots \\ \mathtt{B}_l \end{bmatrix} \tag{18.8}$$

This rearrangement shows that the $2m \times n$ matrix $\hat{\mathtt{W}}$ may be decomposed as a product of a $2m \times 3l$ motion matrix $\hat{\mathtt{M}}$ and $3l \times n$ structure matrix $\hat{\mathtt{B}}$, and consequently $\hat{\mathtt{W}}$ has maximum rank $3l$.

Fig. 18.2. **Non-rigid motion sequence**. *Top row: alternate frames from a sequence in which a giraffe gracefully walks and flexes its neck, whilst the camera pans to match its speed. Bottom row: point tracks showing the motion over (a) the 10 previous, and (b) the 10 forthcoming frames. These tracks are computed using non-rigid factorization and lie in a six dimensional subspace. Note the very different trajectories of the (rigid) background from the (deforming) foreground. Yet these are all spanned by the six basis vectors of the motion matrix. The rank can be accounted for as follows: the sequence motion is effectively that of two planes of points moving independently relative to the camera. The background is a rigid object represented by a plane and contributes 2 to the rank. The giraffe in the foreground is represented as a non-rigid object by a set of $l = 2$ planar basis shapes and contributes 4 to the rank. Figures courtesy of Andrew Fitzgibbon and Aeron Morgan.*

Rene Vidal

FOLLOW

MINDS Director & Seder Professor at Johns Hopkins University, Amazon Scholar, NORCE Chief Scientist
Verified email at jhu.edu - Homepage

Machine Learning Computer Vision Biomedical Data Science Robotics Control Theory

TITLE	CITED BY	YEAR
Sparse subspace clustering: Algorithm, theory, and applications E Elhamifar, R Vidal IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (11), 2765 ...	1797	2013
Sparse subspace clustering E Elhamifar, R Vidal IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2790-2797	1345	2009
Generalized principal component analysis (GPCA) R Vidal, Y Ma, S Sastry IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (12), 1945 ...	1293	2005
Subspace clustering R Vidal Signal Processing Magazine, IEEE 28 (2), 52-68	960	2011
A benchmark for the comparison of 3-d motion segmentation algorithms R Tron, R Vidal IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1-8	699	2007

Two-view geometry

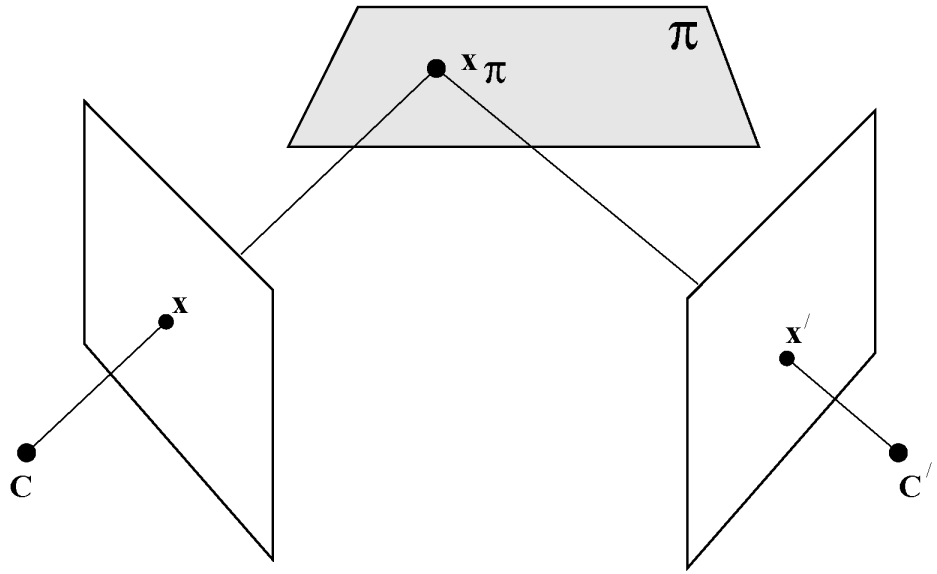
Images of Planes

Projective transformation between images induced by a plane

$$\mathbf{x} = \mathtt{H}_{1\pi}\mathbf{x}_\pi \qquad \mathbf{x}' = \mathtt{H}_{2\pi}\mathbf{x}_\pi$$

$$\mathbf{x}' = \mathtt{H}_{2\pi}\mathbf{x}_\pi$$

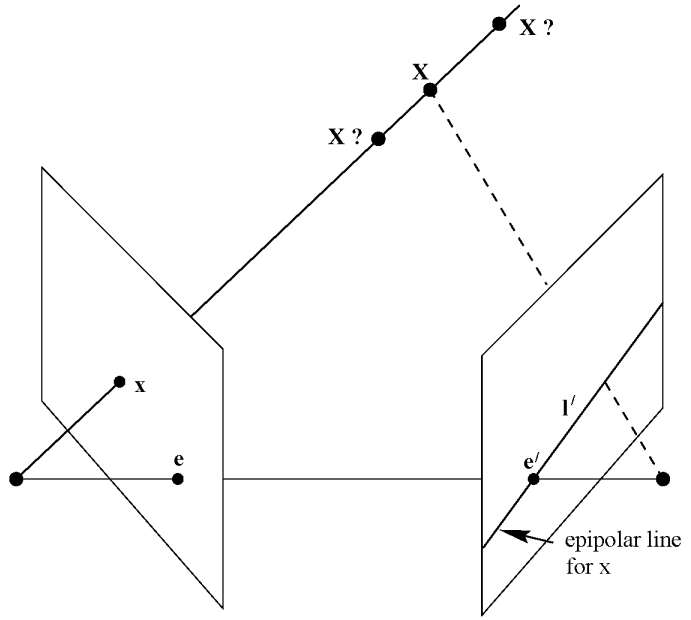
$$= \mathtt{H}_{2\pi}\mathtt{H}_{1\pi}^{-1}\mathbf{x} = \mathtt{H}\mathbf{x}$$



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

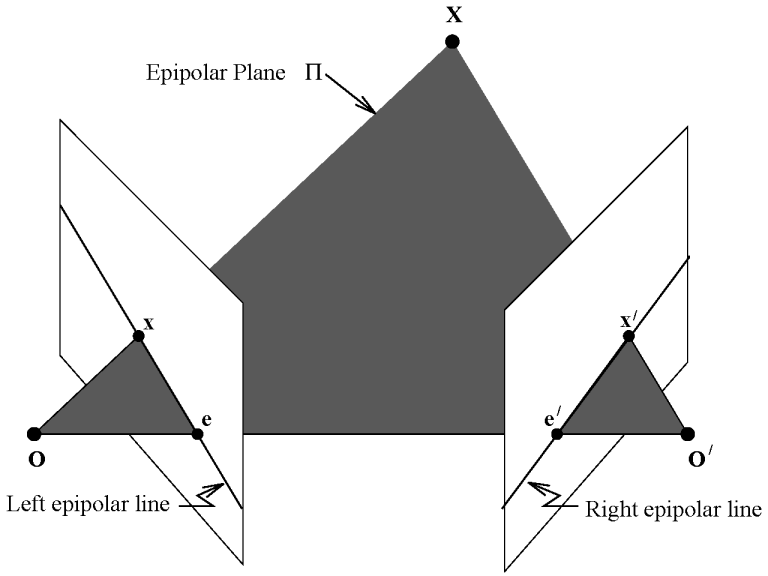
Correspondence Geometry

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



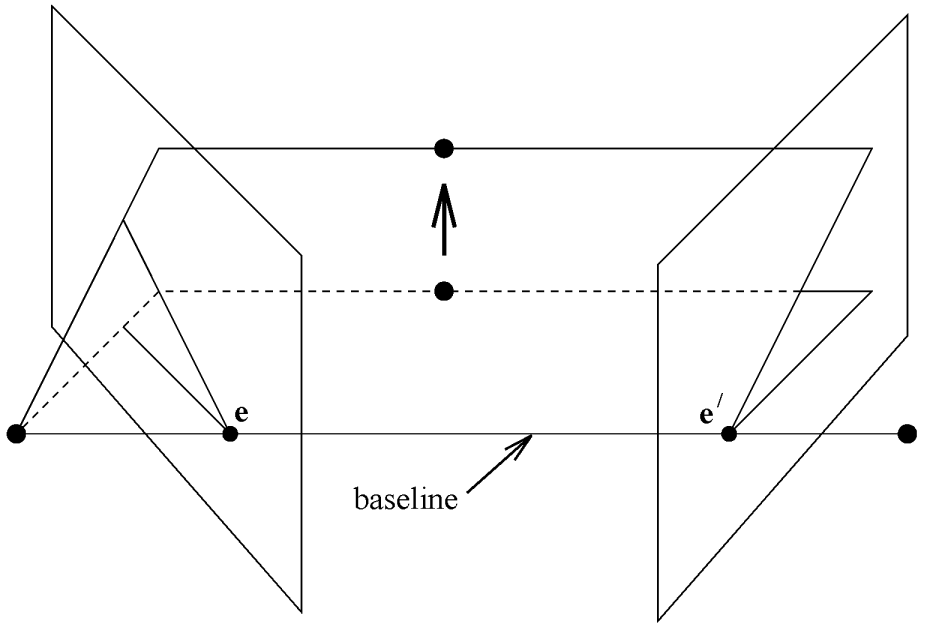
- A point in one image “generates” a line in the other image.
- This line is known as an epipolar line, and the geometry which gives rise to it is known as epipolar geometry.

Epipolar Geometry



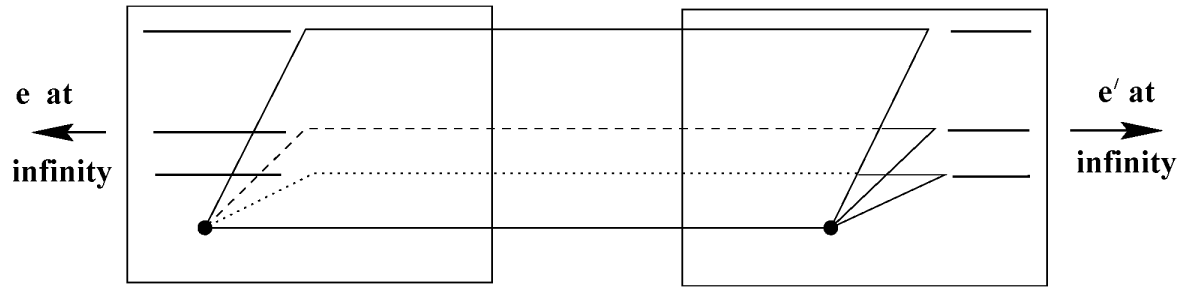
- The epipolar line $\mathbf{l}'$ is the image of the ray through $\mathbf{x}$.
- The epipole $\mathbf{e}'$ is the point of intersection of the line joining the camera centres—the baseline—with the image plane.
- The epipole is also the image in one camera of the centre of the other camera.
- All epipolar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

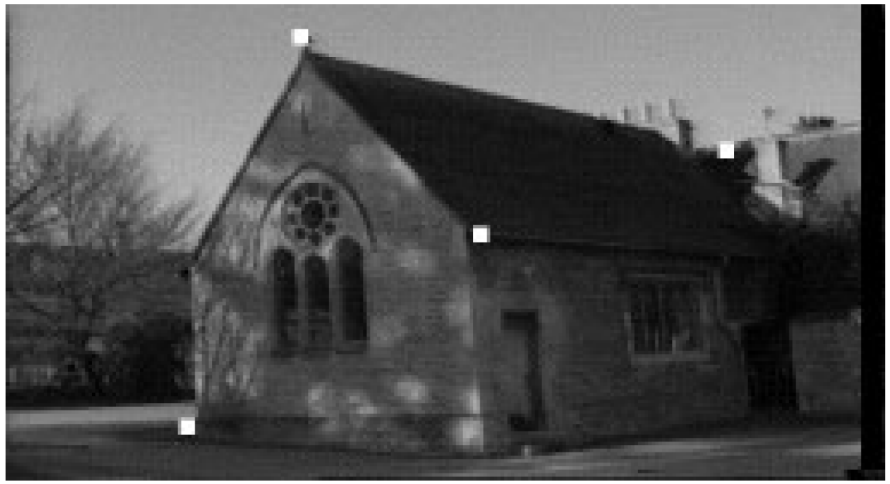
Epipolar pencil

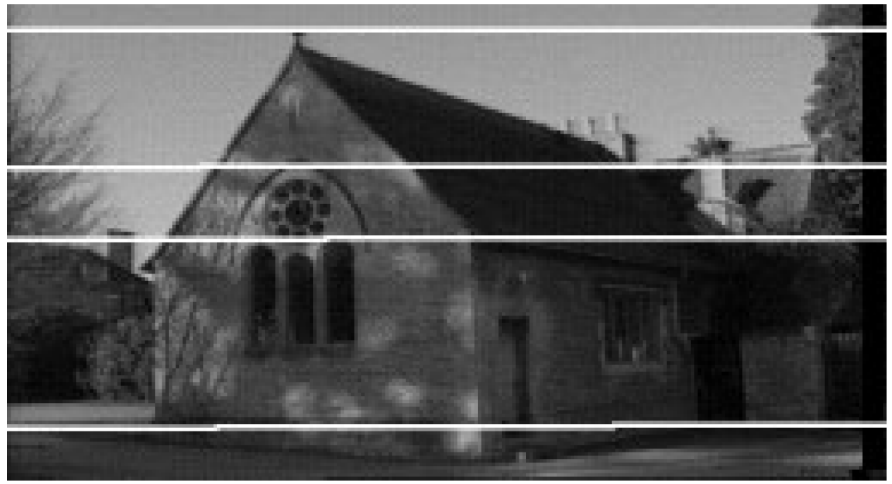


As the position of the 3D point $\mathbf{X}$ varies, the epipolar planes “rotate” about the baseline. This family of planes is known as an epipolar pencil. All epipolar lines intersect at the epipole.

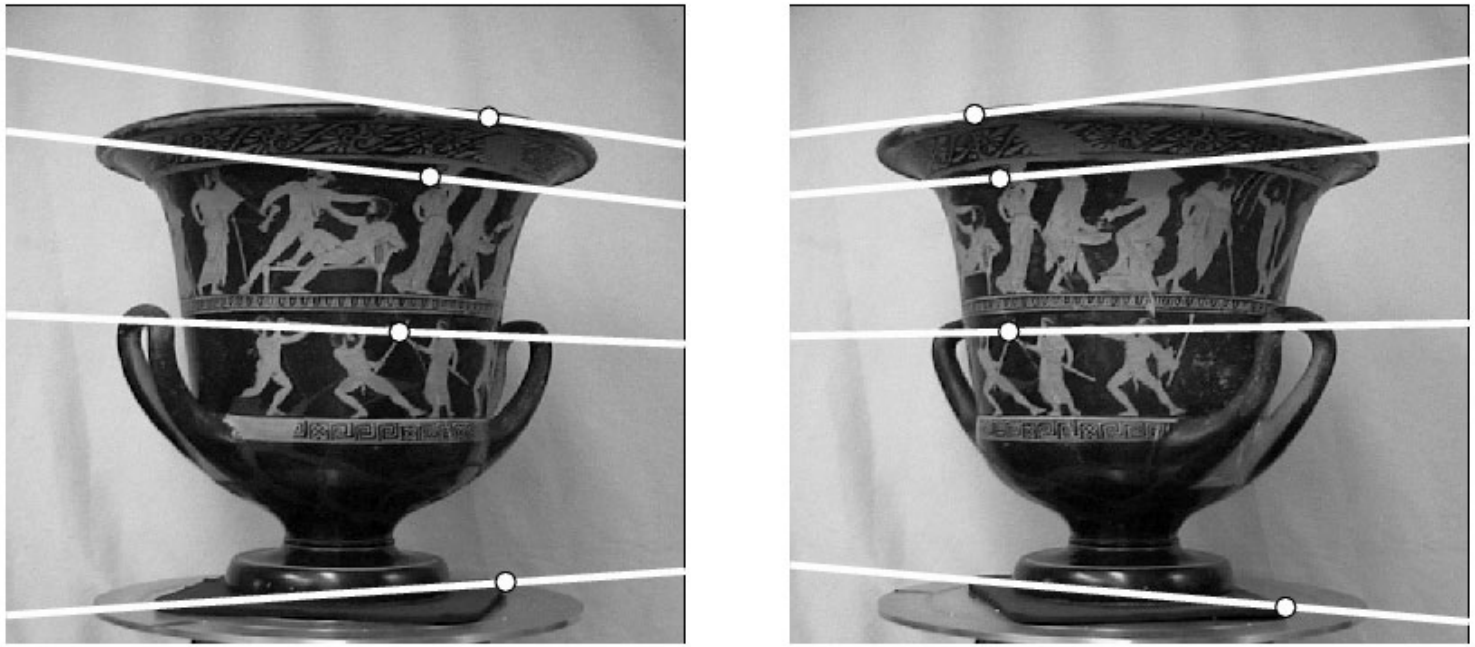
Epipolar geometry example







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).

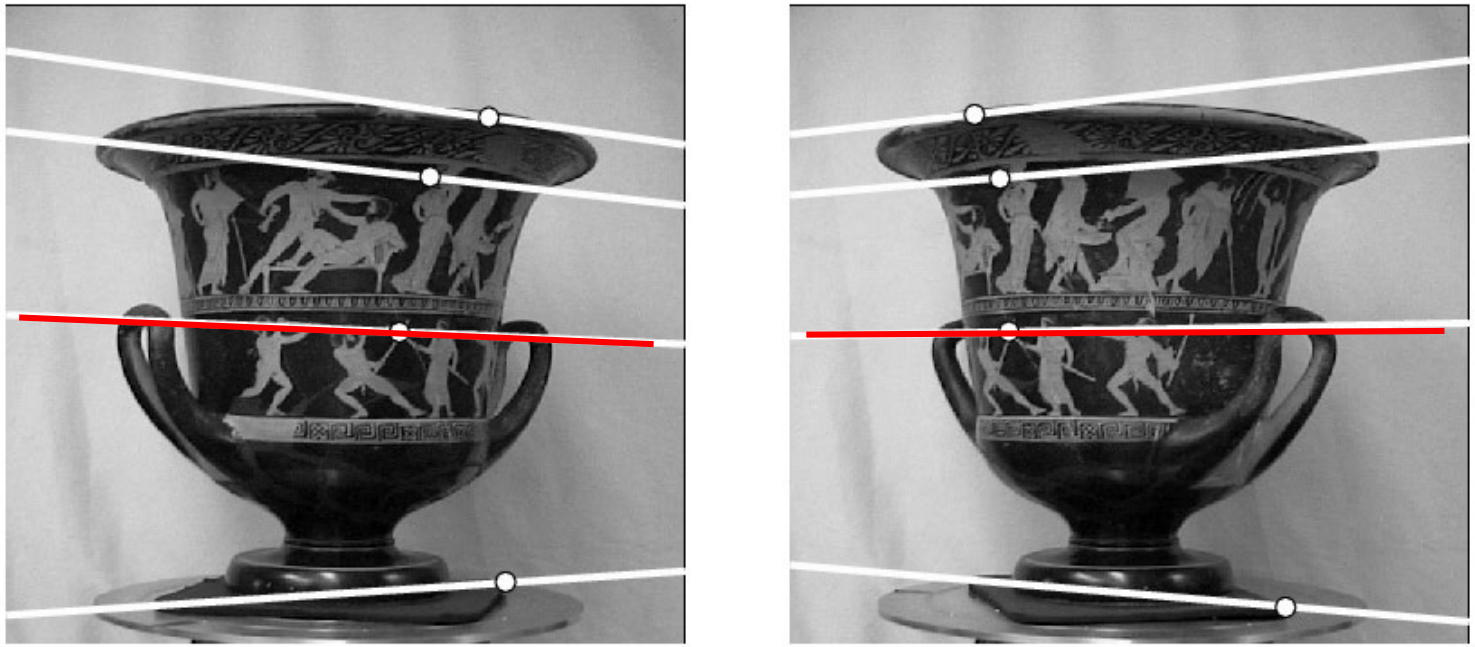


Examples of epipolar lines:

These are the intersections of the epipolar planes with the images.

Corresponding points lie on corresponding lines.

Point matching becomes a 1-dimensional search.

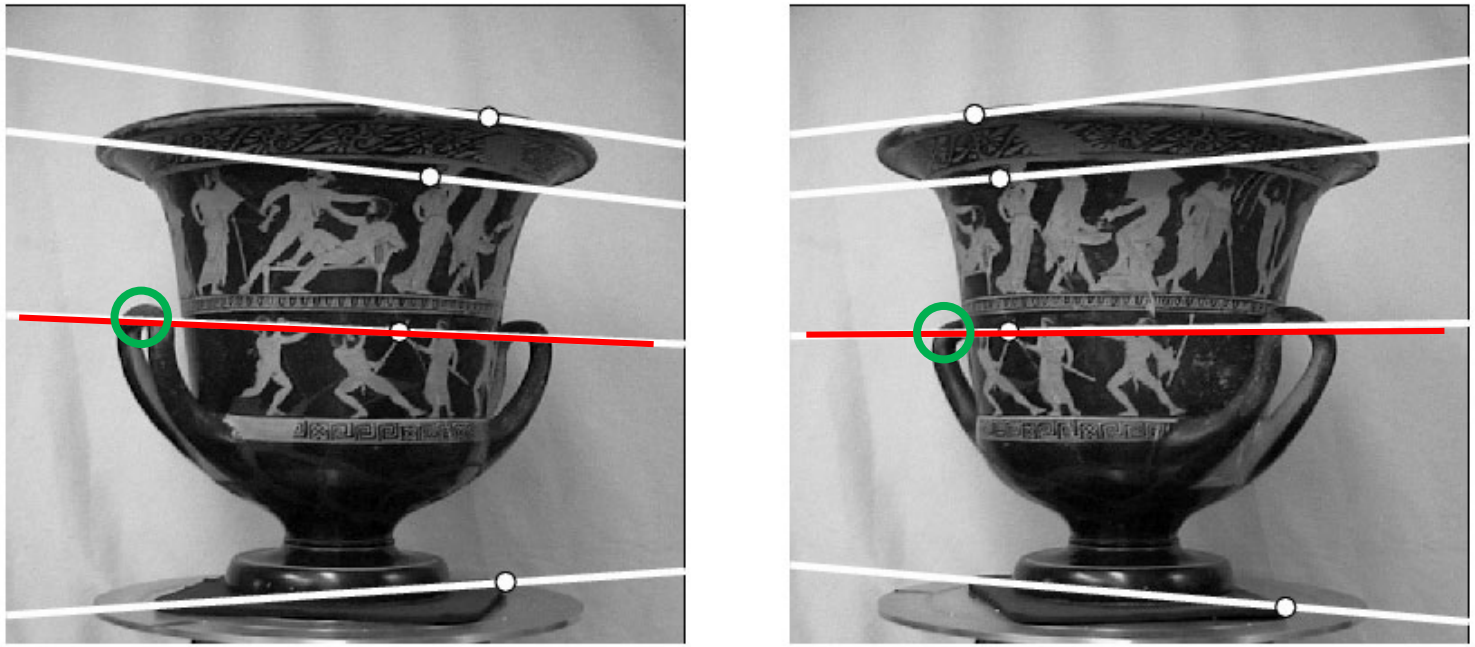


Examples of epipolar lines:

These are the intersections of the epipolar planes with the images.

Corresponding points lie on corresponding lines.

Point matching becomes a 1-dimensional search.

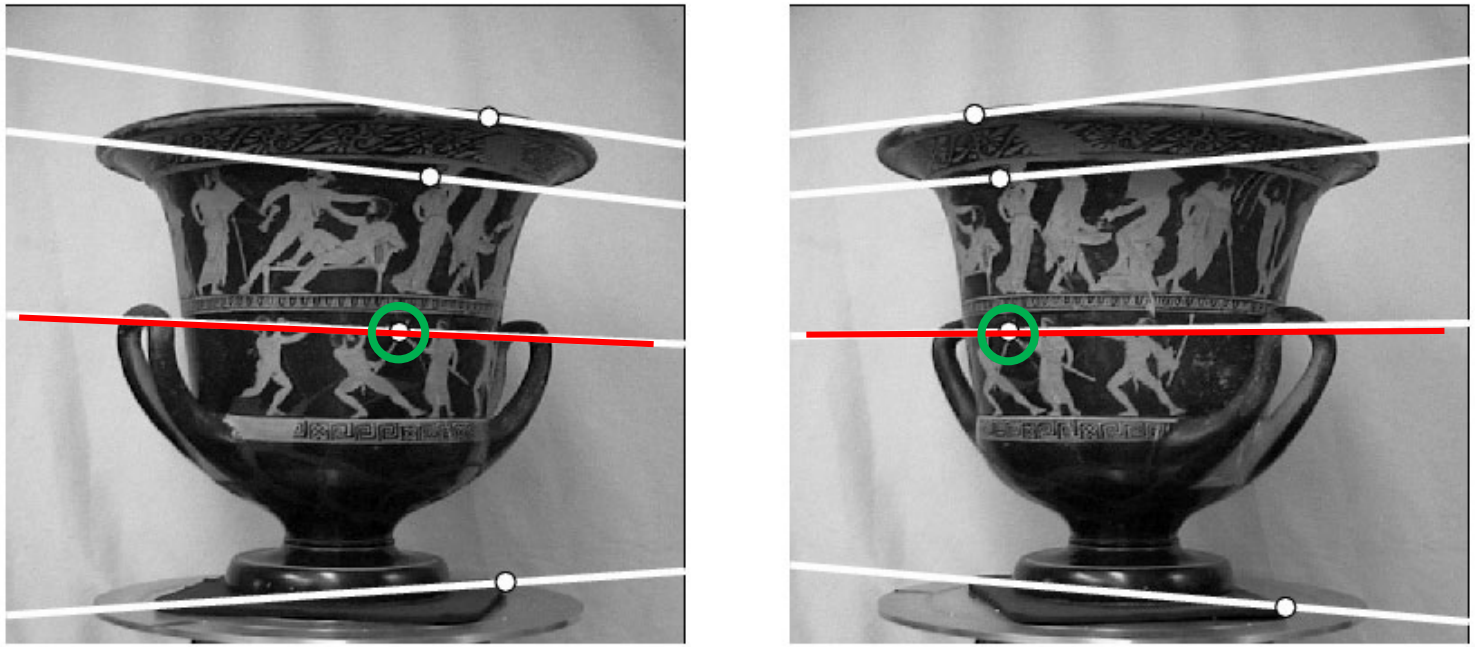


Examples of epipolar lines:

These are the intersections of the epipolar planes with the images.

Corresponding points lie on corresponding lines.

Point matching becomes a 1-dimensional search.

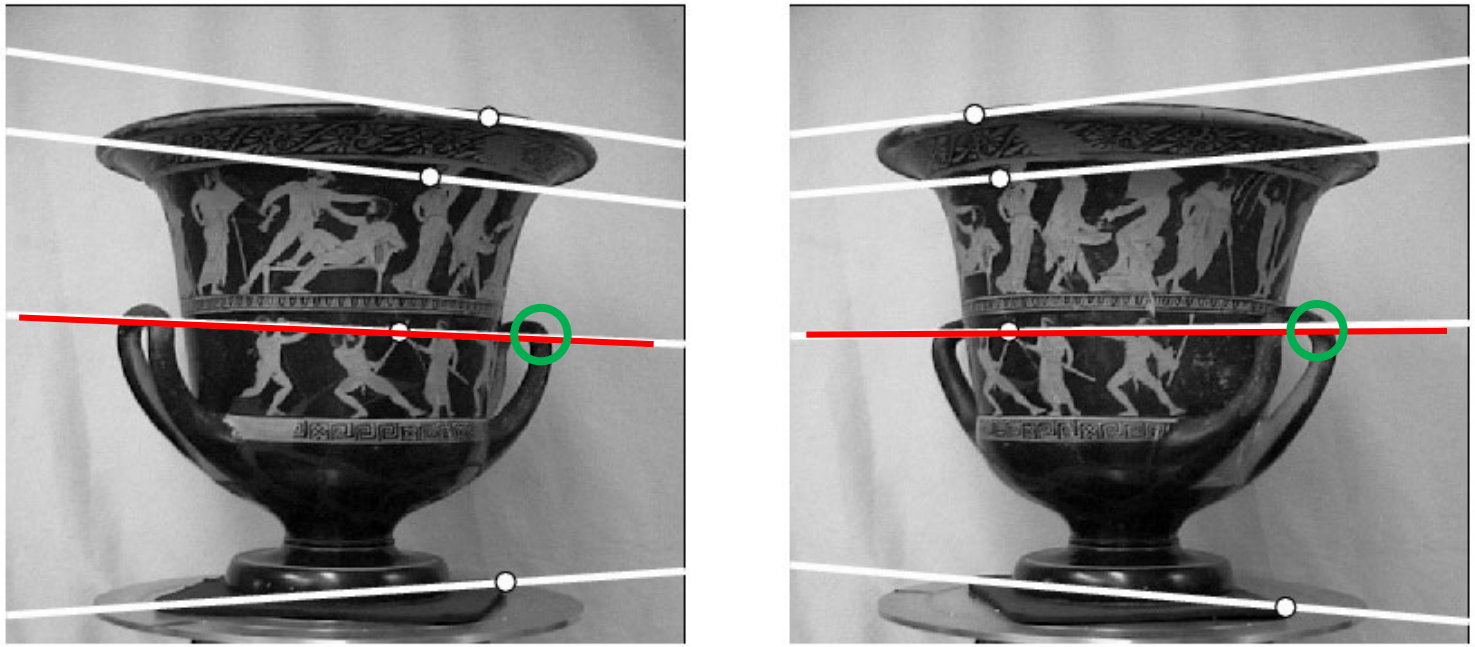


Examples of epipolar lines:

These are the intersections of the epipolar planes with the images.

Corresponding points lie on corresponding lines.

Point matching becomes a 1-dimensional search.

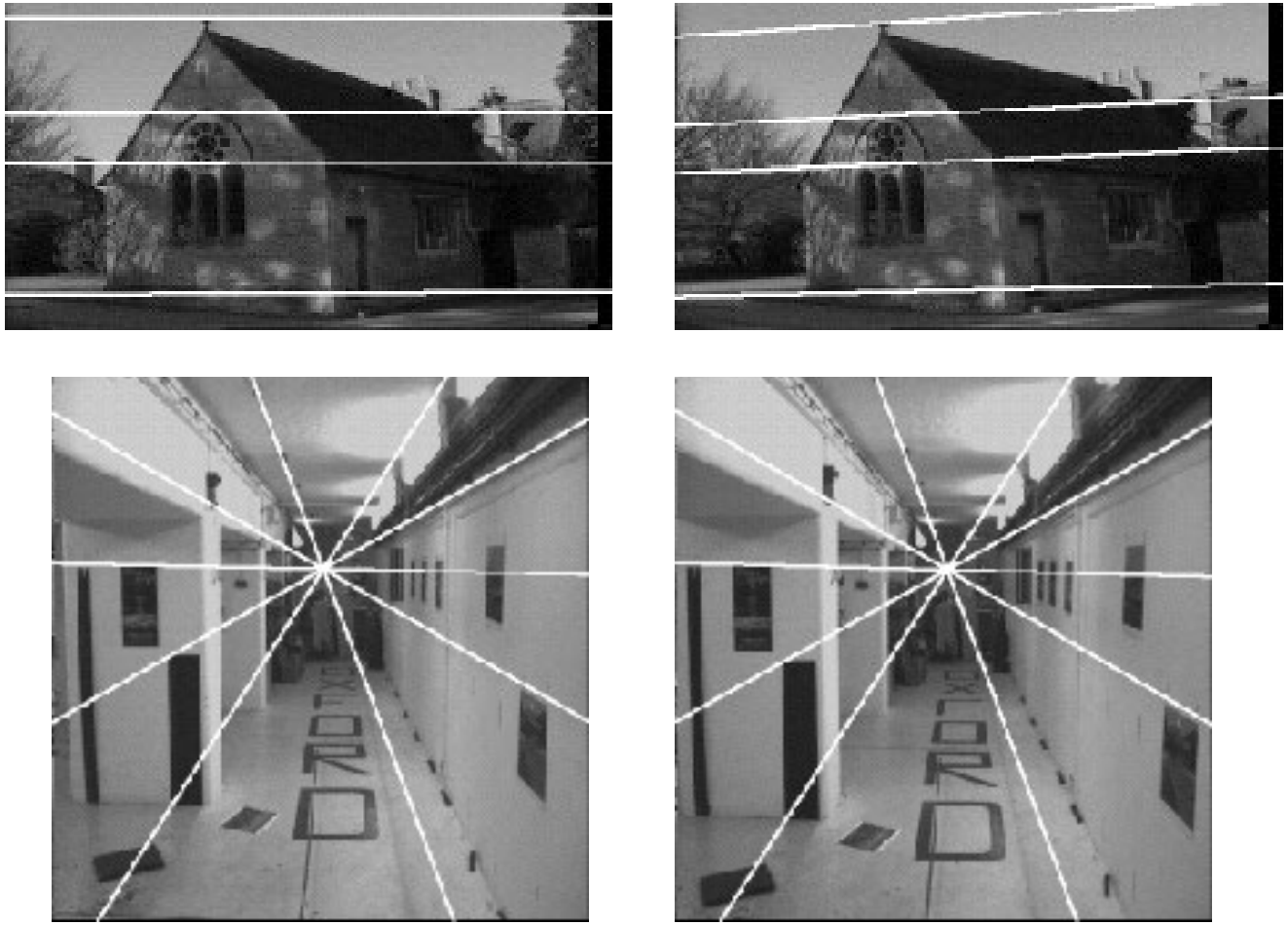


Examples of epipolar lines:

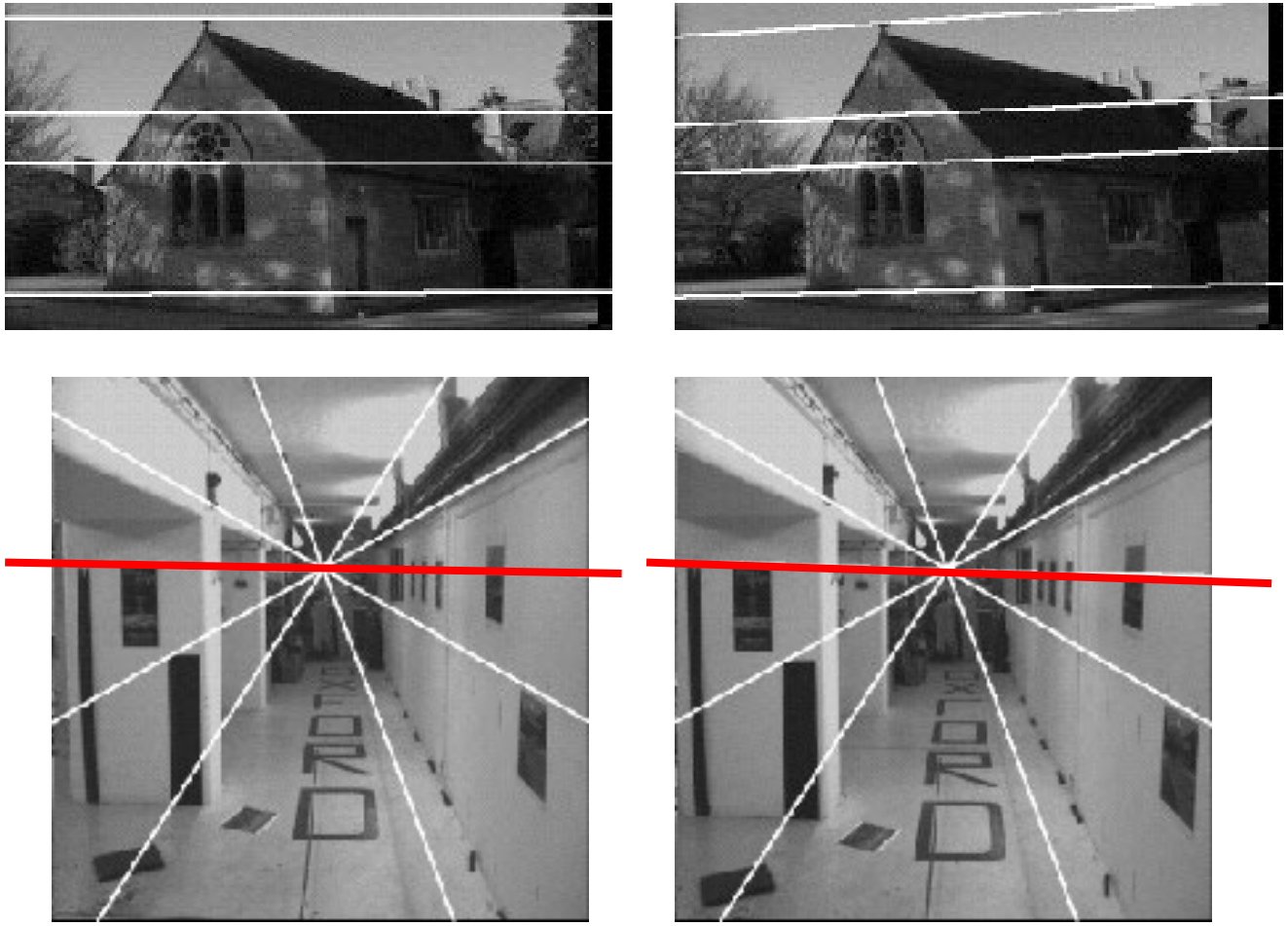
These are the intersections of the epipolar planes with the images.

Corresponding points lie on corresponding lines.

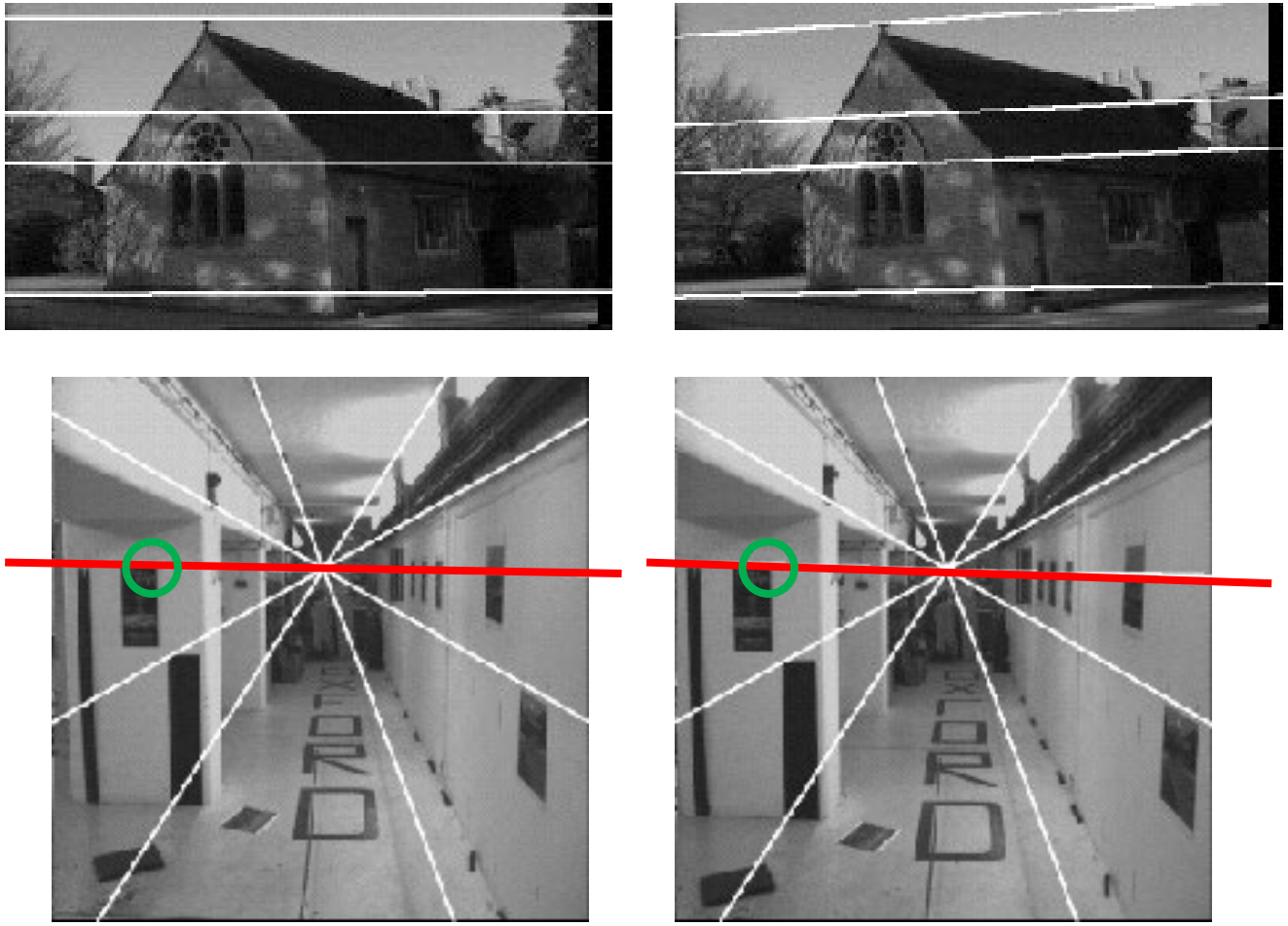
Point matching becomes a 1-dimensional search.



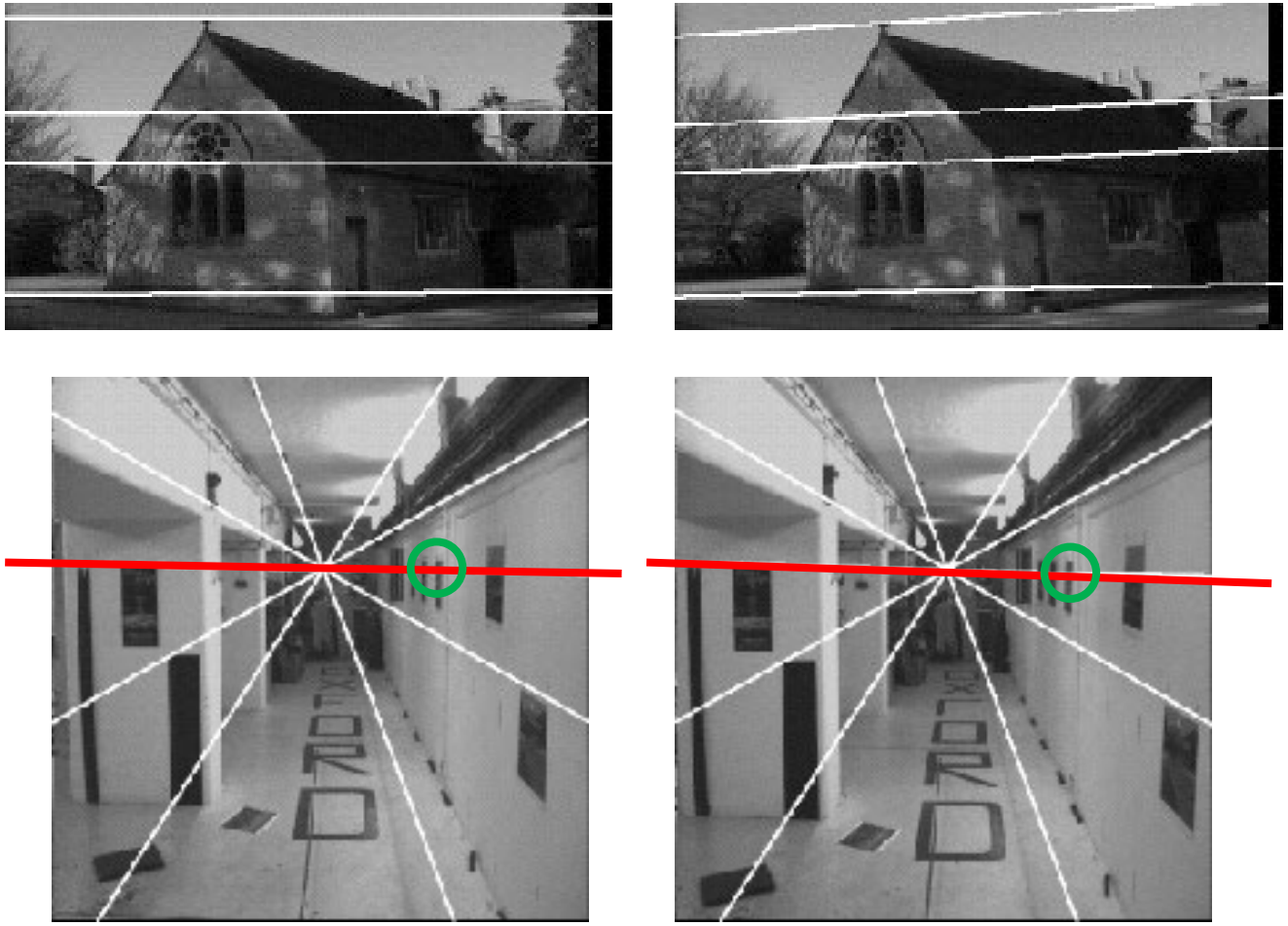
More examples of epipolar lines.



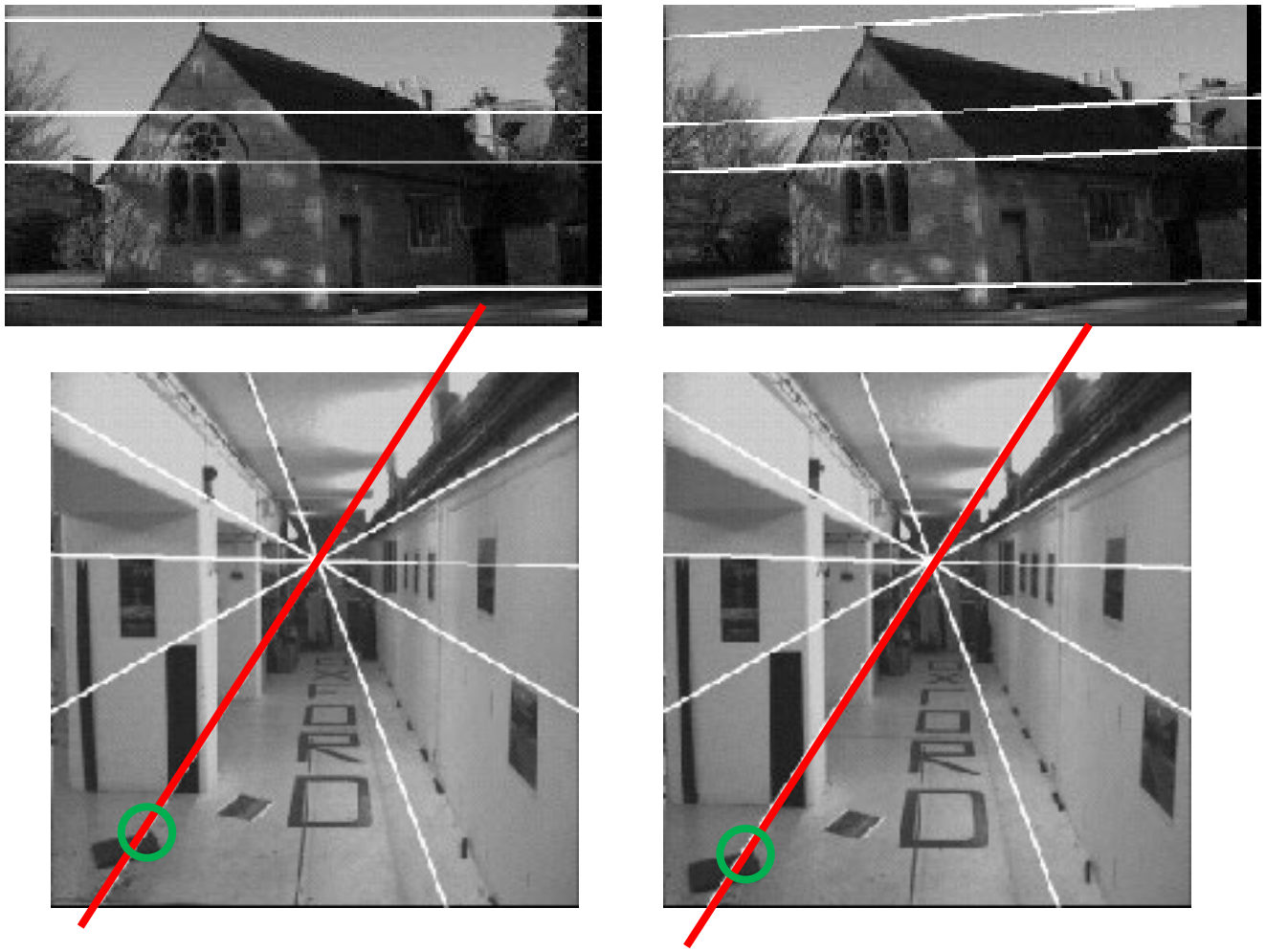
More examples of epipolar lines.



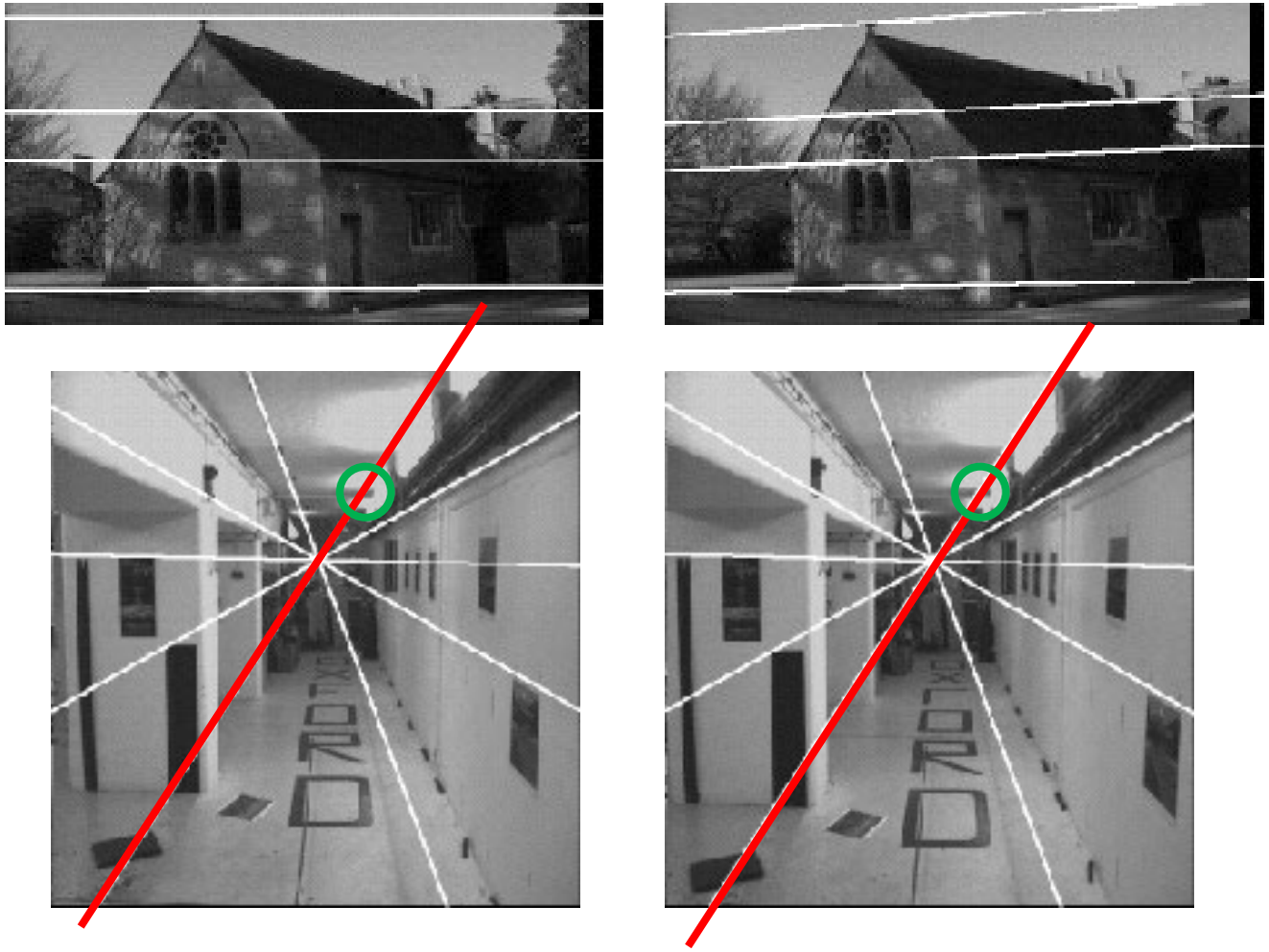
More examples of epipolar lines.



More examples of epipolar lines.

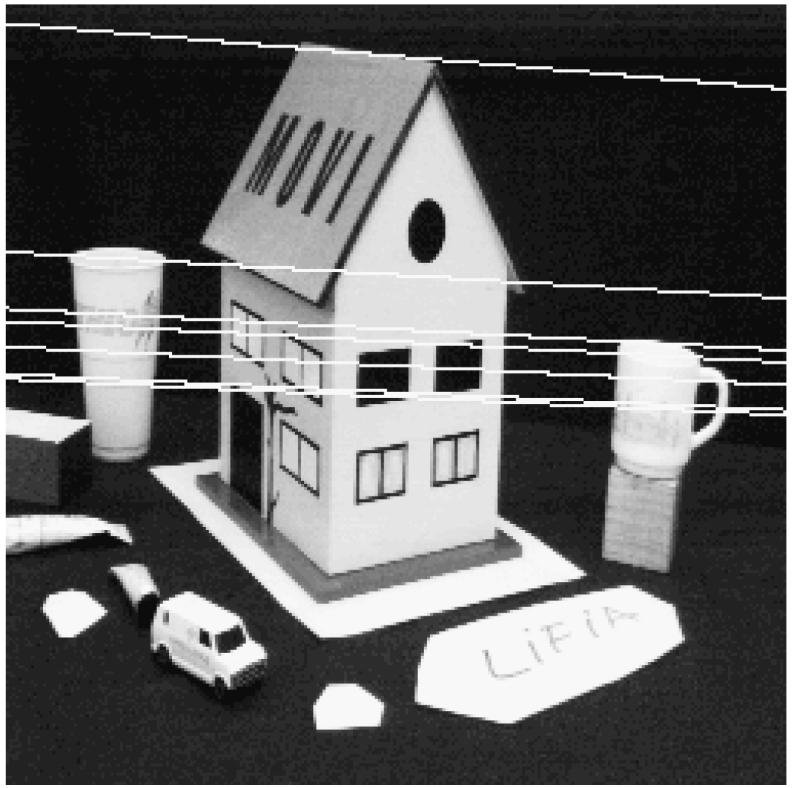


More examples of epipolar lines.



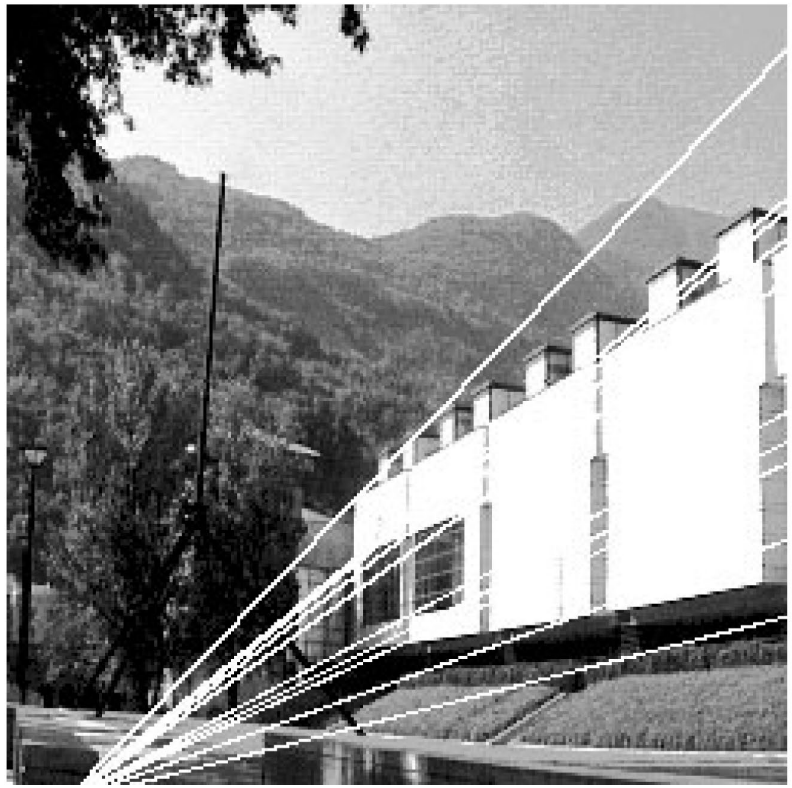
More examples of epipolar lines.

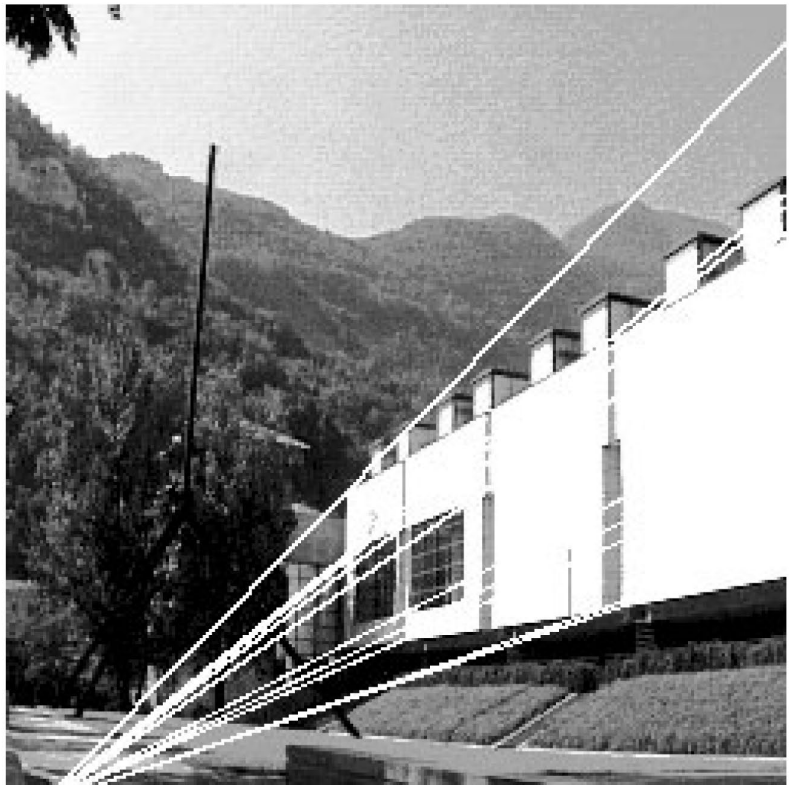
Lifia House images



Statue images

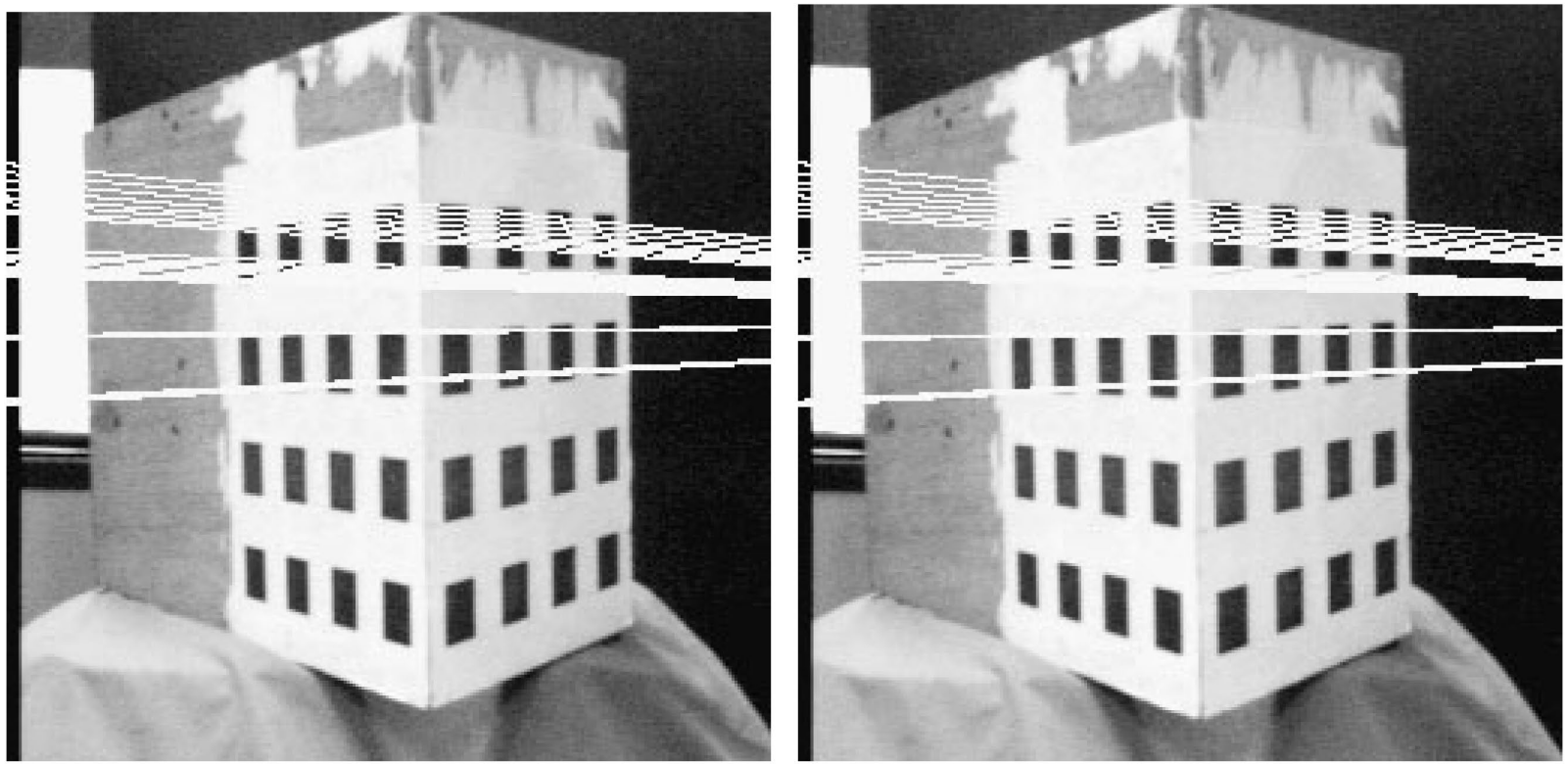
Grenoble Museum





Oxford Basement

Calibration object



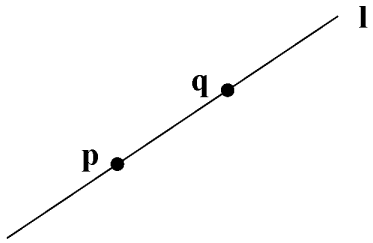
Homogeneous Notation Interlude

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

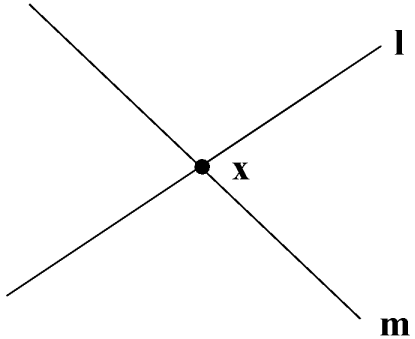
$$\mathbf{l} = \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix}$$

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

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

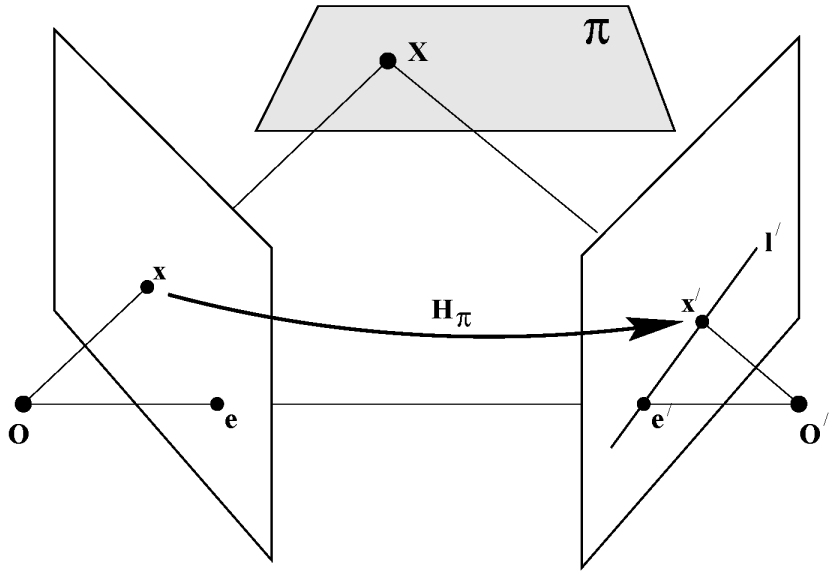
- $[\mathbf{v}]_\times$ is a 3×3 skew-symmetric matrix of rank 2.
- $\mathbf{v}$ is the null-vector of $[\mathbf{v}]_\times$, since $\mathbf{v} \times \mathbf{v} = [\mathbf{v}]_\times \mathbf{v} = \mathbf{0}$.

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

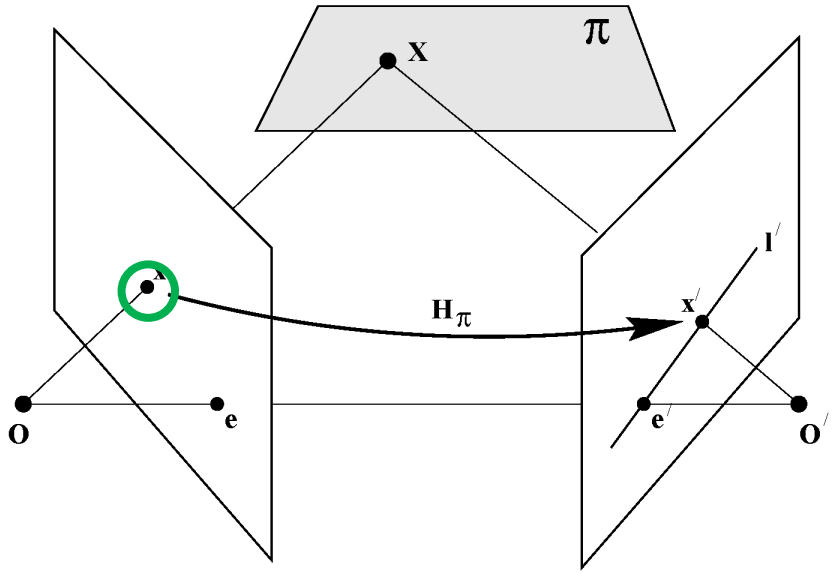
Step 2 : Construct the epipolar line $\mathbf{l}' = \mathbf{e}' \times \mathbf{x}' = [\mathbf{e}']_\times \mathbf{x}'$

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

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

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

Fundamental matrix - sketch derivation



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

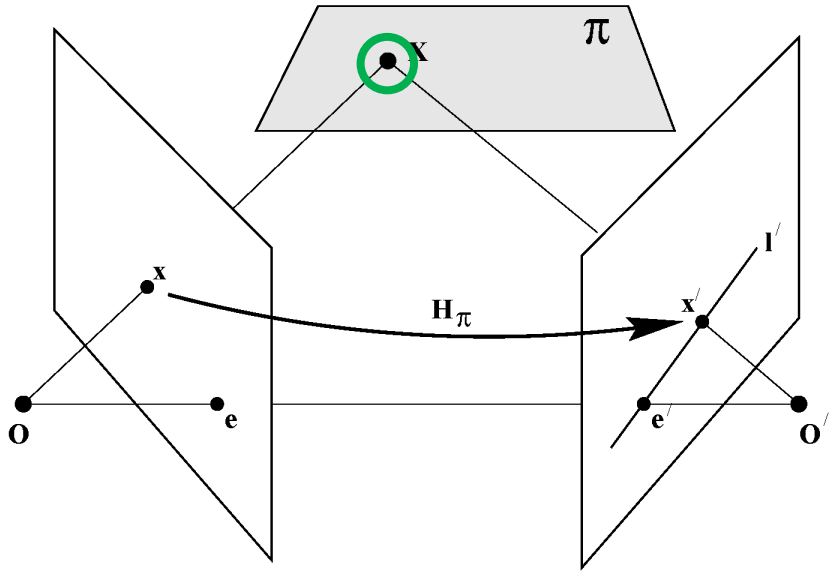
Step 2 : Construct the epipolar line $\mathbf{l}' = \mathbf{e}' \times \mathbf{x}' = [\mathbf{e}']_\times \mathbf{x}'$

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

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

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

Fundamental matrix - sketch derivation



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

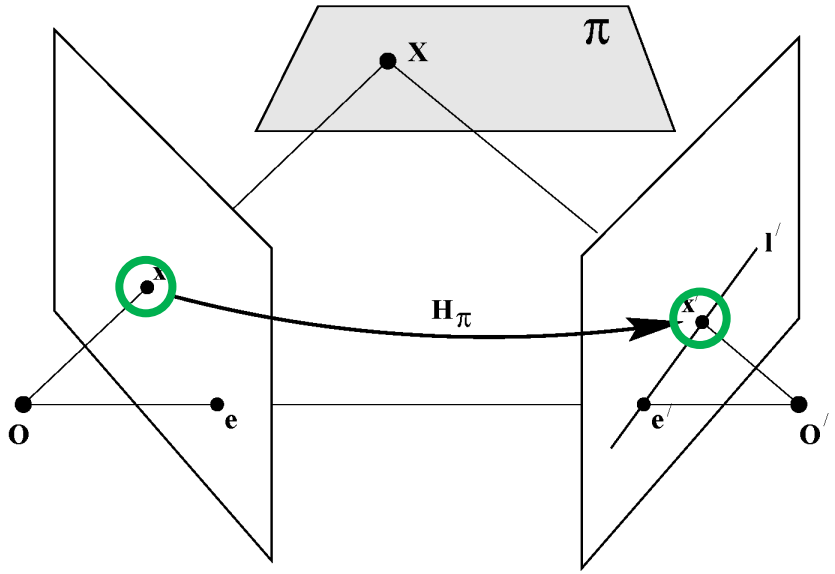
Step 2 : Construct the epipolar line $\mathbf{l}' = \mathbf{e}' \times \mathbf{x}' = [\mathbf{e}']_\times \mathbf{x}'$

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

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

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

Fundamental matrix - sketch derivation



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

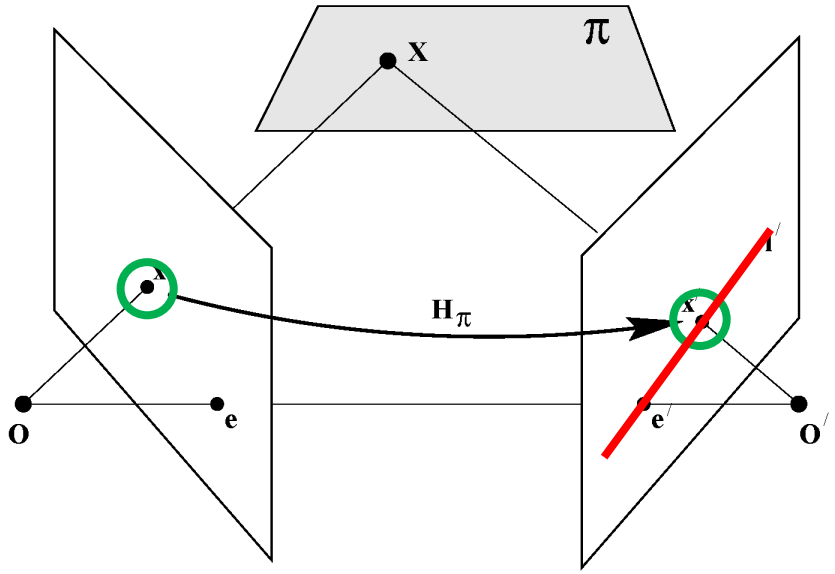
Step 2 : Construct the epipolar line $\mathbf{l}' = \mathbf{e}' \times \mathbf{x}' = [\mathbf{e}']_\times \mathbf{x}'$

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

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

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

Fundamental matrix - sketch derivation



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

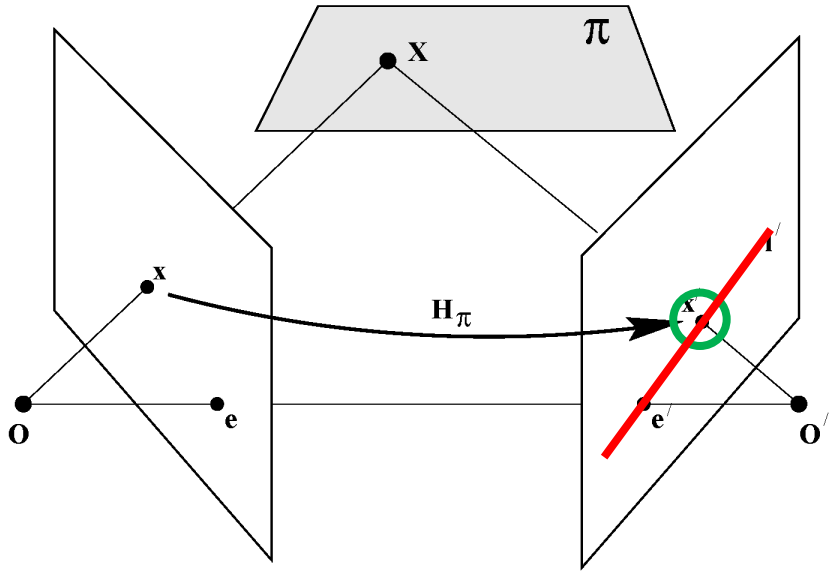
Step 2 : Construct the epipolar line $\mathbf{l}' = \mathbf{e}' \times \mathbf{x}' = [\mathbf{e}']_\times \mathbf{x}'$

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

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

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

Fundamental matrix - sketch derivation



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

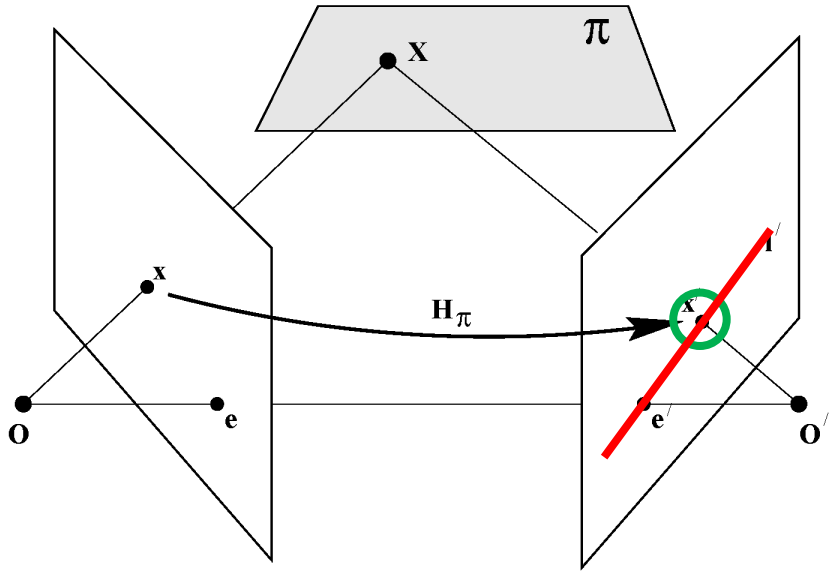
Step 2 : Construct the epipolar line $\mathbf{l}' = \mathbf{e}' \times \mathbf{x}' = [\mathbf{e}']_\times \mathbf{x}'$

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

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

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

Fundamental matrix - sketch derivation



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

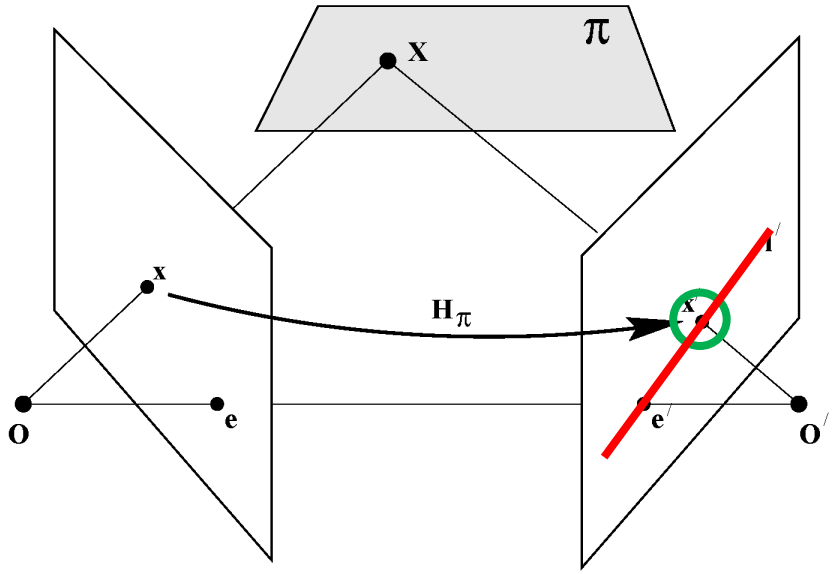
Step 2 : Construct the epipolar line $\mathbf{l}' = \mathbf{e}' \times \mathbf{x}' = [\mathbf{e}']_\times \mathbf{x}'$

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

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

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

Fundamental matrix - sketch derivation



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

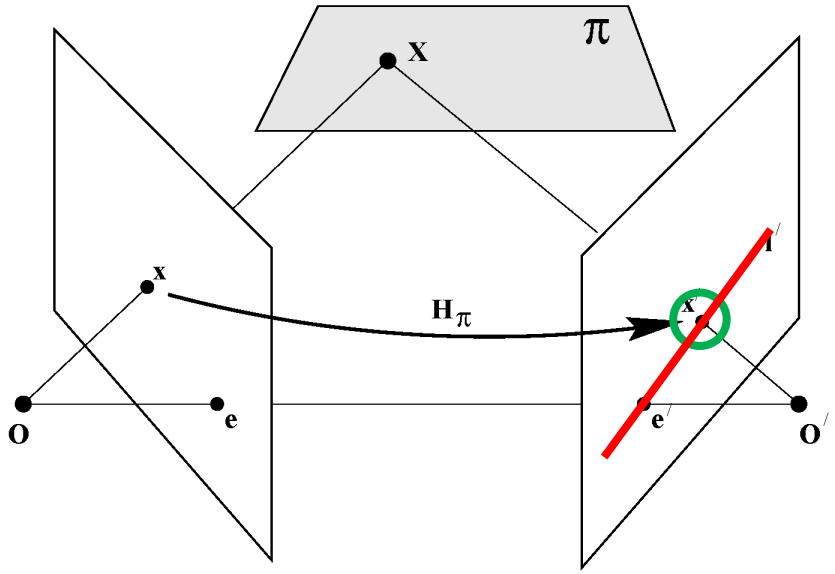
Step 2 : Construct the epipolar line $\mathbf{l}' = \mathbf{e}' \times \mathbf{x}' = [\mathbf{e}']_\times \mathbf{x}'$

$$\mathbf{l}' = \mathtt{F}\mathbf{x}$$

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

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

Fundamental matrix - sketch derivation



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

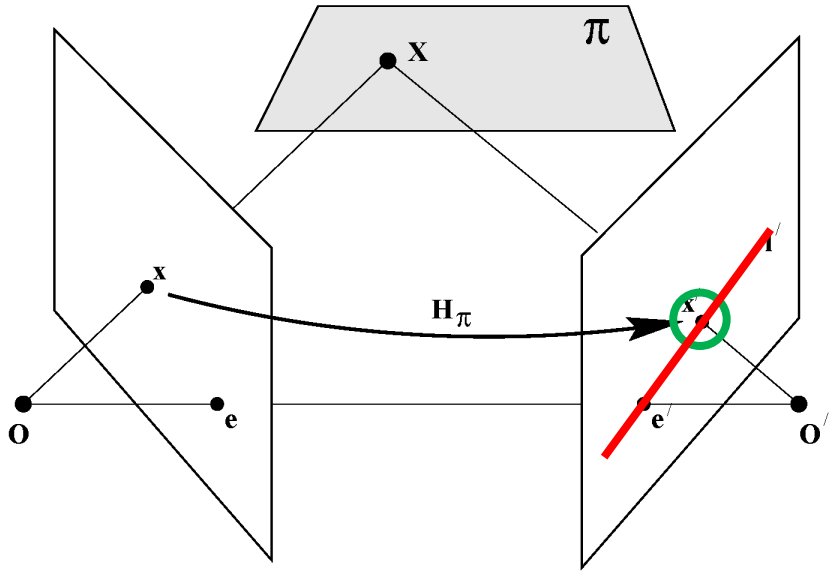
Step 2 : Construct the epipolar line $\mathbf{l}' = \mathbf{e}' \times \mathbf{x}' = [\mathbf{e}']_\times \mathbf{x}'$

$$\mathbf{l}' = \mathtt{F}\mathbf{x}$$

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

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

Fundamental matrix - sketch derivation



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

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

Point $\mathbf{x}'$ lies on line $\mathbf{l}'$. Hence:

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

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** $\mathtt{P}, \mathtt{P}'$:
  - $\mathtt{F} = [\mathtt{P}'\mathtt{C}]_{\times}\mathtt{P}'\mathtt{P}^{+}$, where $\mathtt{P}^{+}$ is the pseudo-inverse of $\mathtt{P}$, and $\mathtt{C}$ is the centre of the first camera. Note, $\mathbf{e}' = \mathtt{P}'\mathtt{C}$.
  - 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 $\mathbf{e} = \mathtt{M}^{-1}\mathbf{m}$.

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** $\mathtt{P}, \mathtt{P}'$:
  - $\mathtt{F} = [\mathtt{P}'\mathtt{C}]_{\times}\mathtt{P}'\mathtt{P}^{+}$, where $\mathtt{P}^{+}$ is the pseudo-inverse of $\mathtt{P}$, and $\mathtt{C}$ is the centre of the first camera. Note, $\mathbf{e}' = \mathtt{P}'\mathtt{C}$.
  - 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 $\mathbf{e} = \mathtt{M}^{-1}\mathbf{m}$.

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** $\mathtt{P}, \mathtt{P}'$:
  - $\mathtt{F} = [\mathtt{P}'\mathtt{C}]_{\times}\mathtt{P}'\mathtt{P}^{+}$, where $\mathtt{P}^{+}$ is the pseudo-inverse of $\mathtt{P}$, and $\mathtt{C}$ is the centre of the first camera. Note, $\mathbf{e}' = \mathtt{P}'\mathtt{C}$.
  - 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 $\mathbf{e} = \mathtt{M}^{-1}\mathbf{m}$.

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** $\mathtt{P}, \mathtt{P}'$:
  - $\mathtt{F} = [\mathtt{P}'\mathtt{C}]_{\times}\mathtt{P}'\mathtt{P}^{+}$, where $\mathtt{P}^{+}$ is the pseudo-inverse of $\mathtt{P}$, and $\mathtt{C}$ is the centre of the first camera. Note, $\mathbf{e}' = \mathtt{P}'\mathtt{C}$.
  - Canonical cameras, $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}], \mathtt{P}' = [\mathtt{M} \mid \mathbf{m}]$, $\mathtt{F} = [\mathbf{e}']_{\times}\mathtt{M}$

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** $\mathtt{P}, \mathtt{P}'$:
  - $\mathtt{F} = [\mathtt{P}'\mathtt{C}]_{\times}\mathtt{P}'\mathtt{P}^{+}$, where $\mathtt{P}^{+}$ is the pseudo-inverse of $\mathtt{P}$, and $\mathtt{C}$ is the centre of the first camera. Note, $\mathbf{e}' = \mathtt{P}'\mathtt{C}$.
  - Canonical cameras, $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}], \mathtt{P}' = [\mathtt{M} \mid \mathbf{m}]$, $\mathtt{F} = [\mathbf{e}']_{\times}\mathtt{M}$

Derivation

If $\mathtt{F}$ factorizes as

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

Then a pair of cameras that correspond to this fundamental matrix are

$$\begin{aligned} \mathtt{P} &= [\mathtt{I} \mid \mathbf{0}] \\ \mathtt{P}' &= [\mathtt{H} \mid \mathbf{e}'] \end{aligned}$$

Check:

1. Centre of camera $\mathtt{P}$ is the origin, with coordinates $(0, 0, 0, 1)^\top$.
2. Point projects via camera $\mathtt{P}' = [\mathtt{H} \mid \mathbf{e}']$ to point $\mathbf{e}'$.
3. This is what it should be, since epipole $\mathbf{e}'$ is defined as the image of centre of $\mathtt{P}$ as viewed by camera $\mathbf{P}'$.
4. Verify that $\mathtt{H}$ is the image-to-image homography for the plane at infinity.
   (a) Let point $\mathbf{x} = (x, y, 1)$ be in image of $\mathtt{P}'$.
   (b) Point $\mathbf{X} = (x, y, 1, 0)$ on plane at infinity satisfies $\mathtt{P}\mathbf{X} = \mathbf{x}$.
   (c) $\mathbf{X}$ maps to point in camera $\mathtt{P}'$ given by

$$\mathbf{x}' = [\mathtt{H} \mid \mathbf{e}']\mathbf{X} = \mathtt{H}\mathbf{x} \ .$$

Plane induced homographies given $\mathtt{F}$

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

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

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

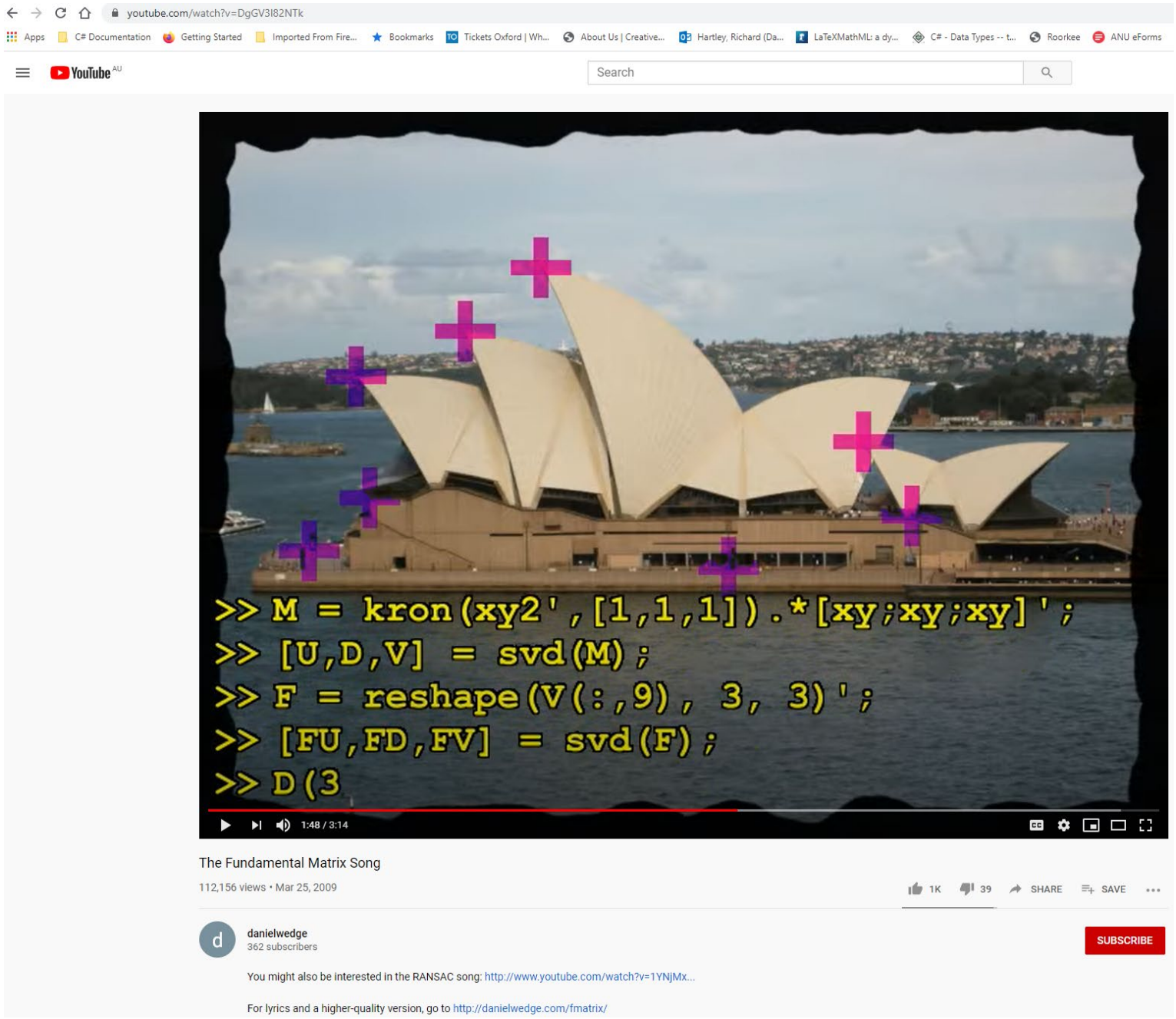
e.g. compute plane from 3 point correspondences.

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

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

The fundamental matrix song

Daniel Wedge, youtube or http://danielwedge.com/fmatrix



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 X_i such that

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

Projective ambiguity of reconstruction

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

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

Then verify

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

- Same problem holds however many views we have

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

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

- Compute the points 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 , . . .

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

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

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 $\mathtt{A}\mathbf{f} = \mathbf{0}$.

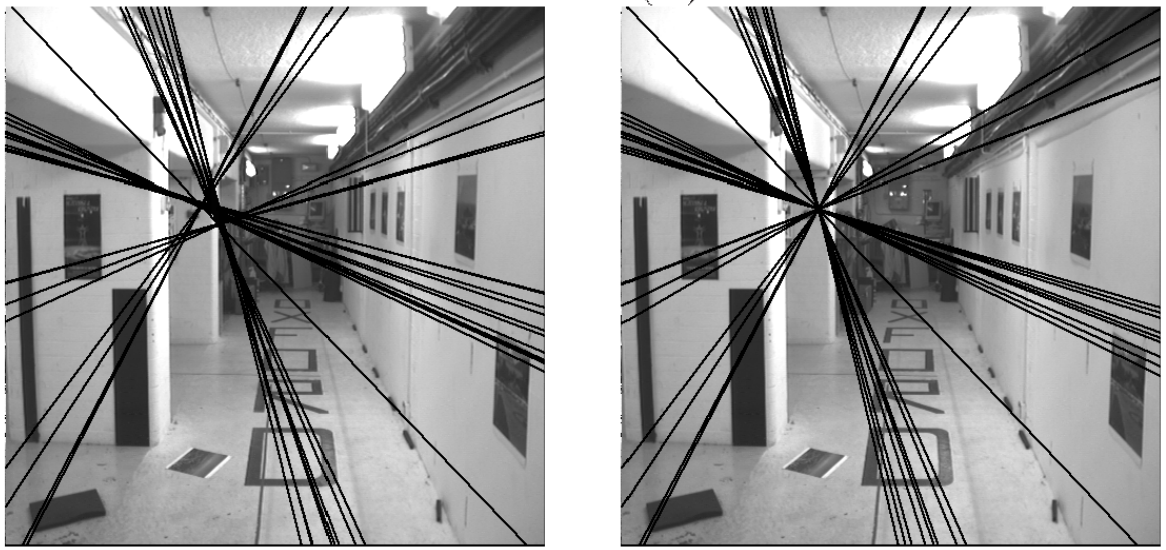
(ii) Take SVD : $\mathtt{A} = \mathtt{U}\mathtt{D}\mathtt{V}^\top$.

(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 : $\det(\mathbf{F}) = 0$.



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

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

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.

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 $\mathtt{F}$.

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.

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

Specific formula

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

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

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

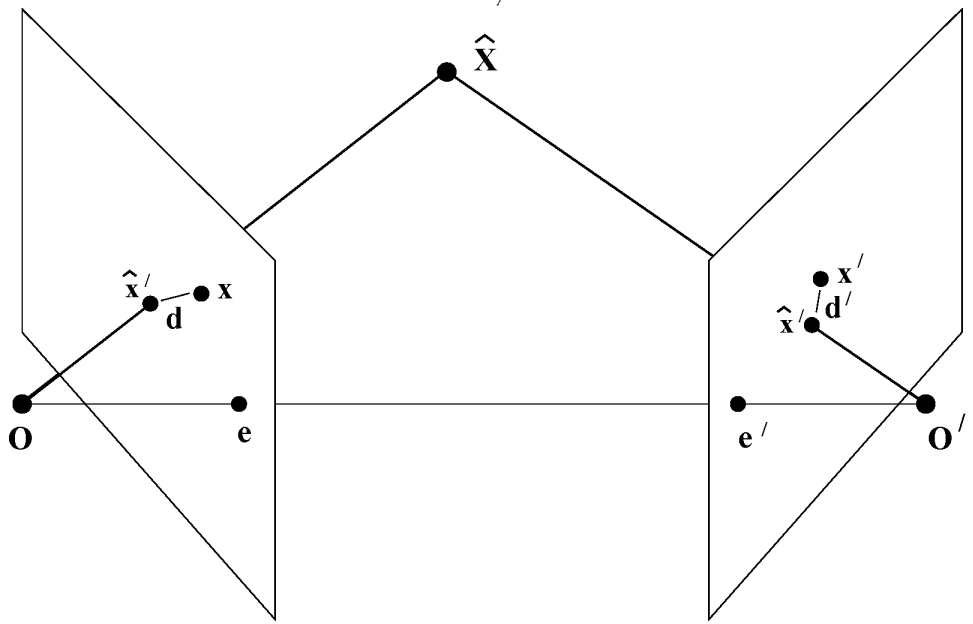
- Difference factorizations give configurations varying by a projective transformation.
- 4-parameter family of solutions with $\mathtt{P} = [\mathtt{I} \mid \mathtt{0}]$.

Triangulation

Triangulation :

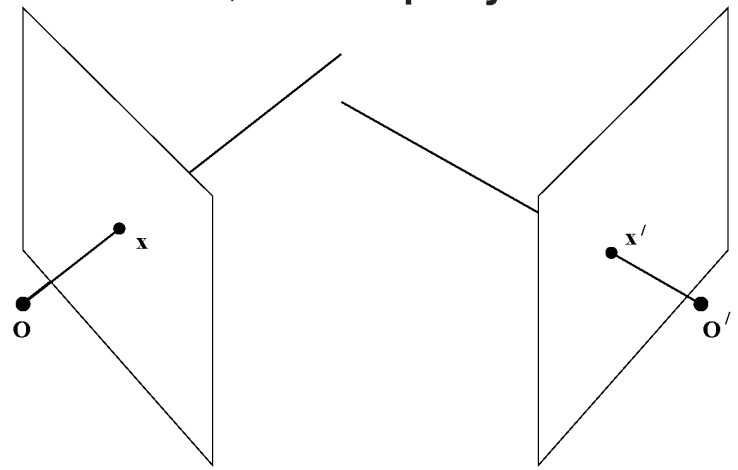
- Knowing P and P'
- Knowing x and x'
- Compute x' such that

$$\mathbf{x} = \mathrm{P}\mathrm{x} \quad ; \quad \mathbf{x}' = \mathrm{P}'\mathrm{x}$$

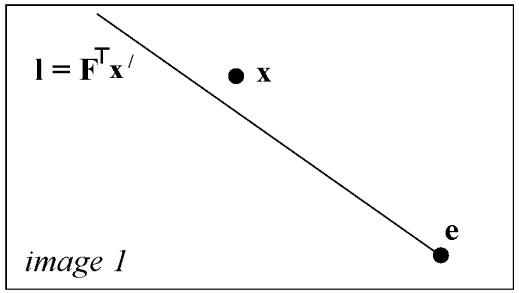


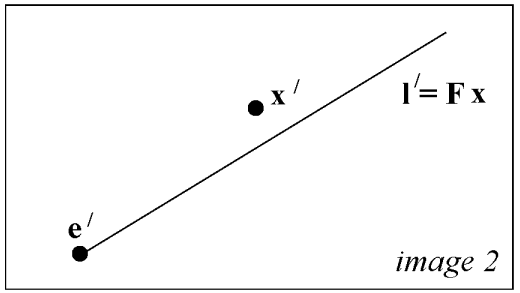
Triangulation in presence of noise

- 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

$$\mathbf{x} = \mathrm{P}\mathrm{x}$$
$$\mathbf{x}' = \mathrm{P}'\mathrm{x}$$

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

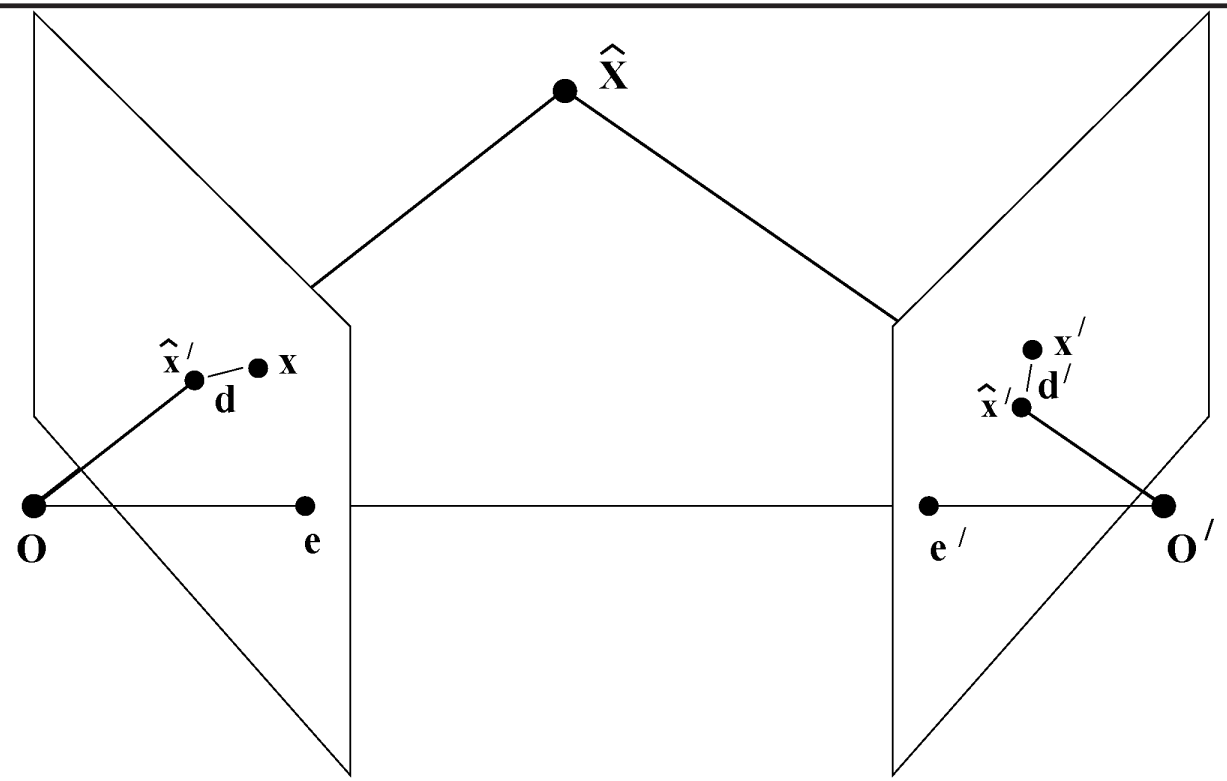
$$\begin{bmatrix} x\mathbf{p}^{3\top} - \mathbf{p}^{1\top} \\ y\mathbf{p}^{3\top} - \mathbf{p}^{2\top} \\ x'\mathbf{p}'^{3\top} - \mathbf{p}'^{1\top} \\ y\mathbf{p}'^{3\top} - \mathbf{p}'^{2\top} \end{bmatrix} \mathrm{x} = 0$$

- Solve for x.
- Generalizes to point match in several images.
- Minimizes no meaningful quantity – not optimal.

Minimizing geometric error

- Point 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 x that minimizes difference between projected and measured points.

Geometric error . . .



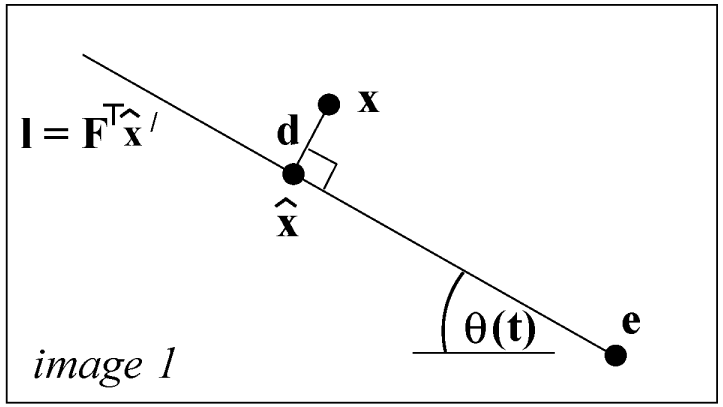
Cost function

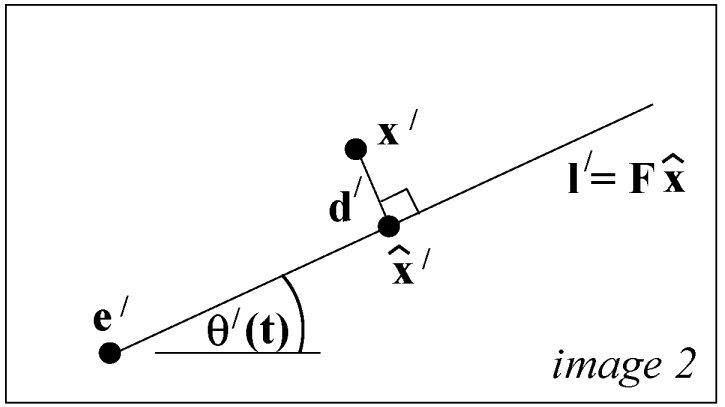
$$\mathcal{C}(\mathrm{x}) = d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \hat{\mathbf{x}}')^2$$

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.

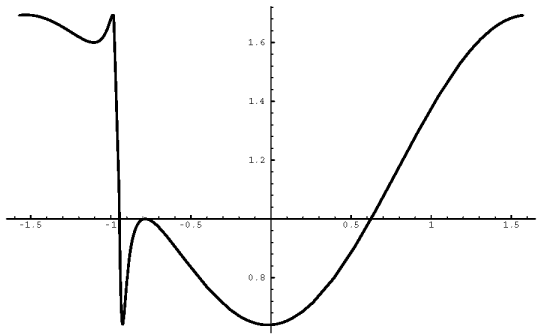
(iv) Find the value of t that minimizes this function.

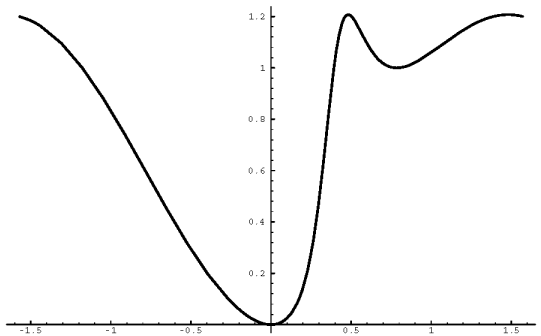
Minimization method ...

- 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.

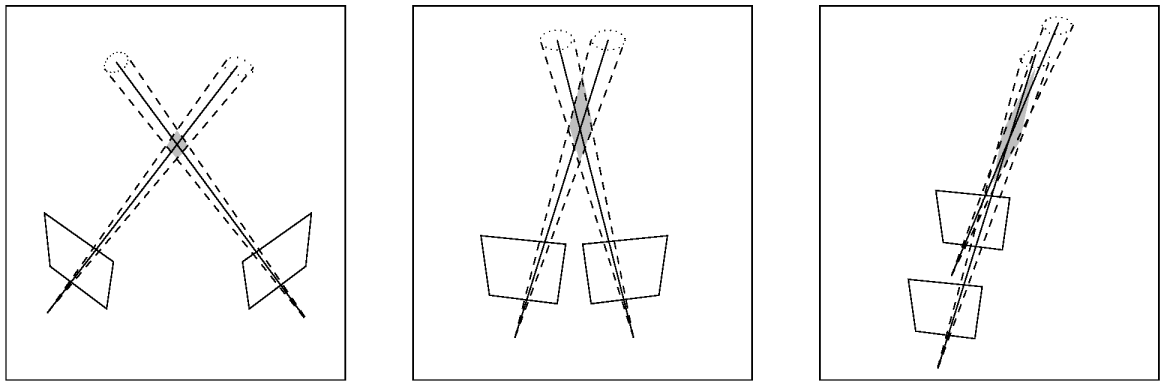




Left : Example of a cost function with three minima.

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.

Complete 8-point algorithm

8 point algorithm has two steps :

(i) **Linear solution.** Solve $\mathtt{A}\mathtt{f} = 0$ 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 $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$

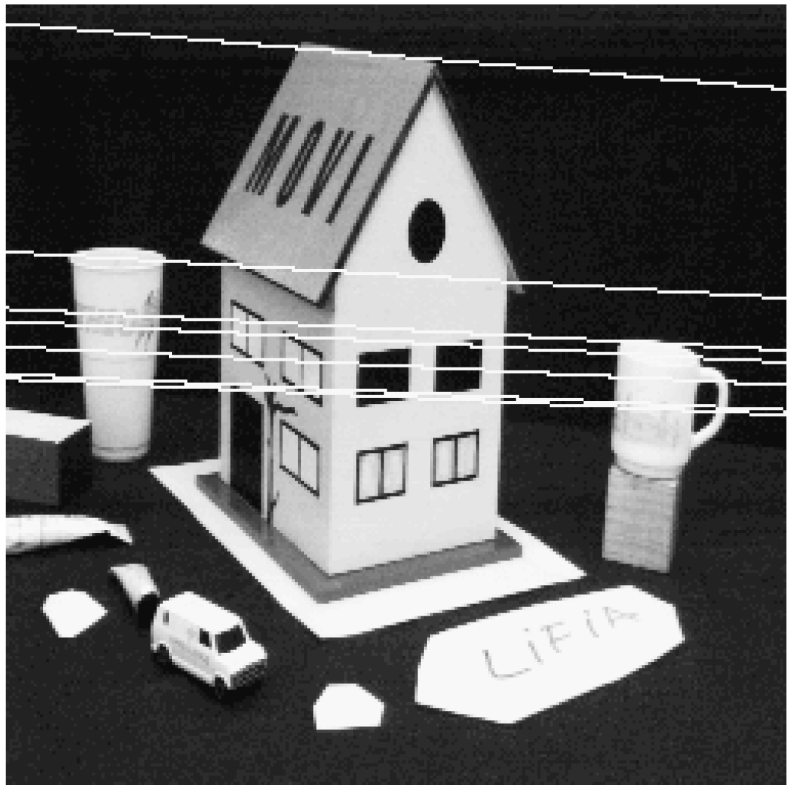
$$\hat{\mathbf{x}}_i'^{\top}\widehat{\mathtt{F}}\hat{\mathbf{x}}_i = 0$$

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

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

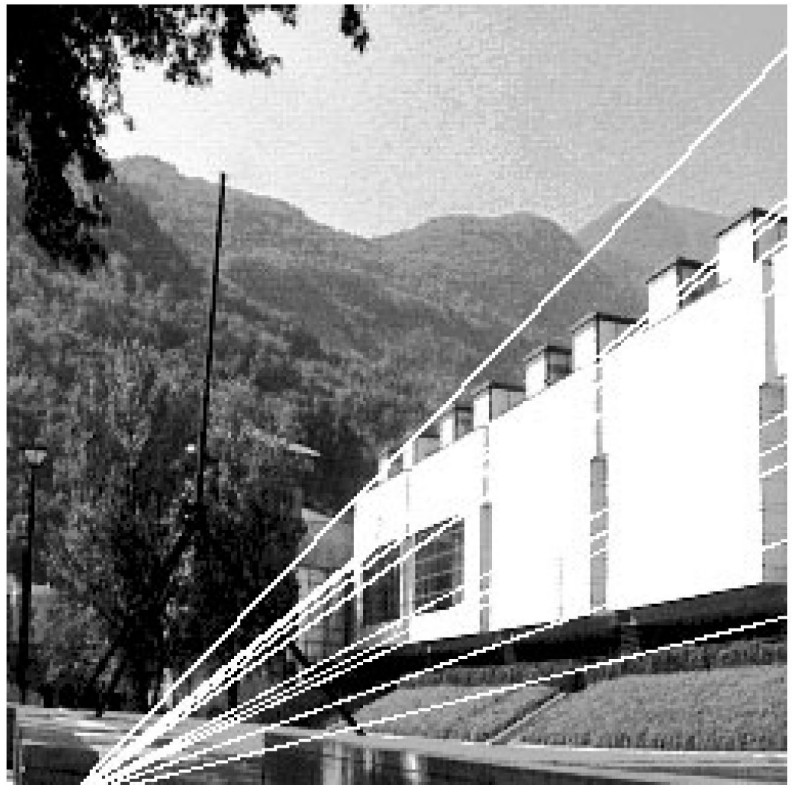
Comparison of Normalized and Unnormalized Algorithms

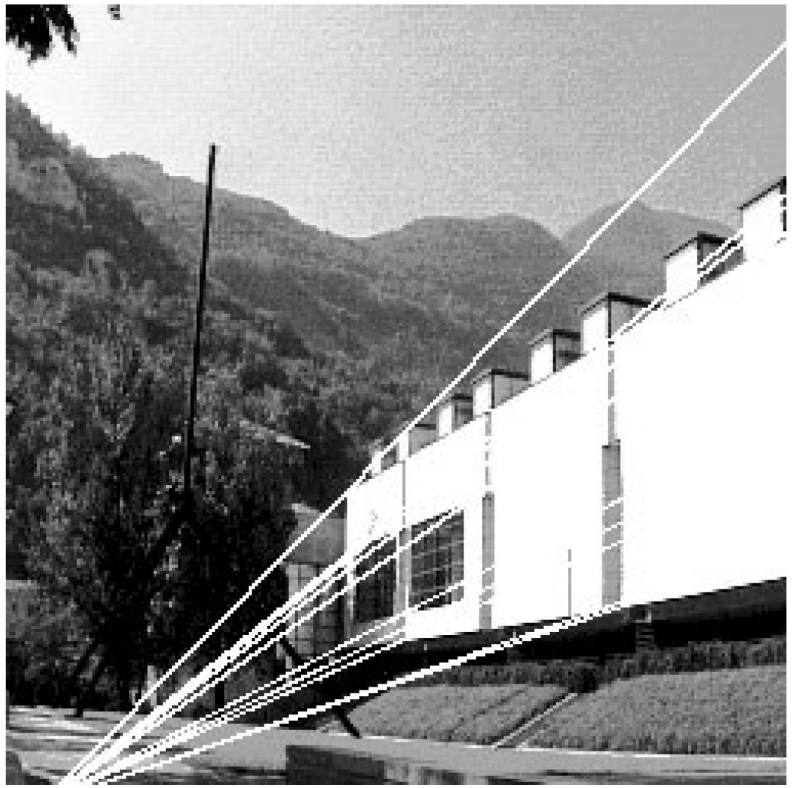
Lifia House images



Statue images

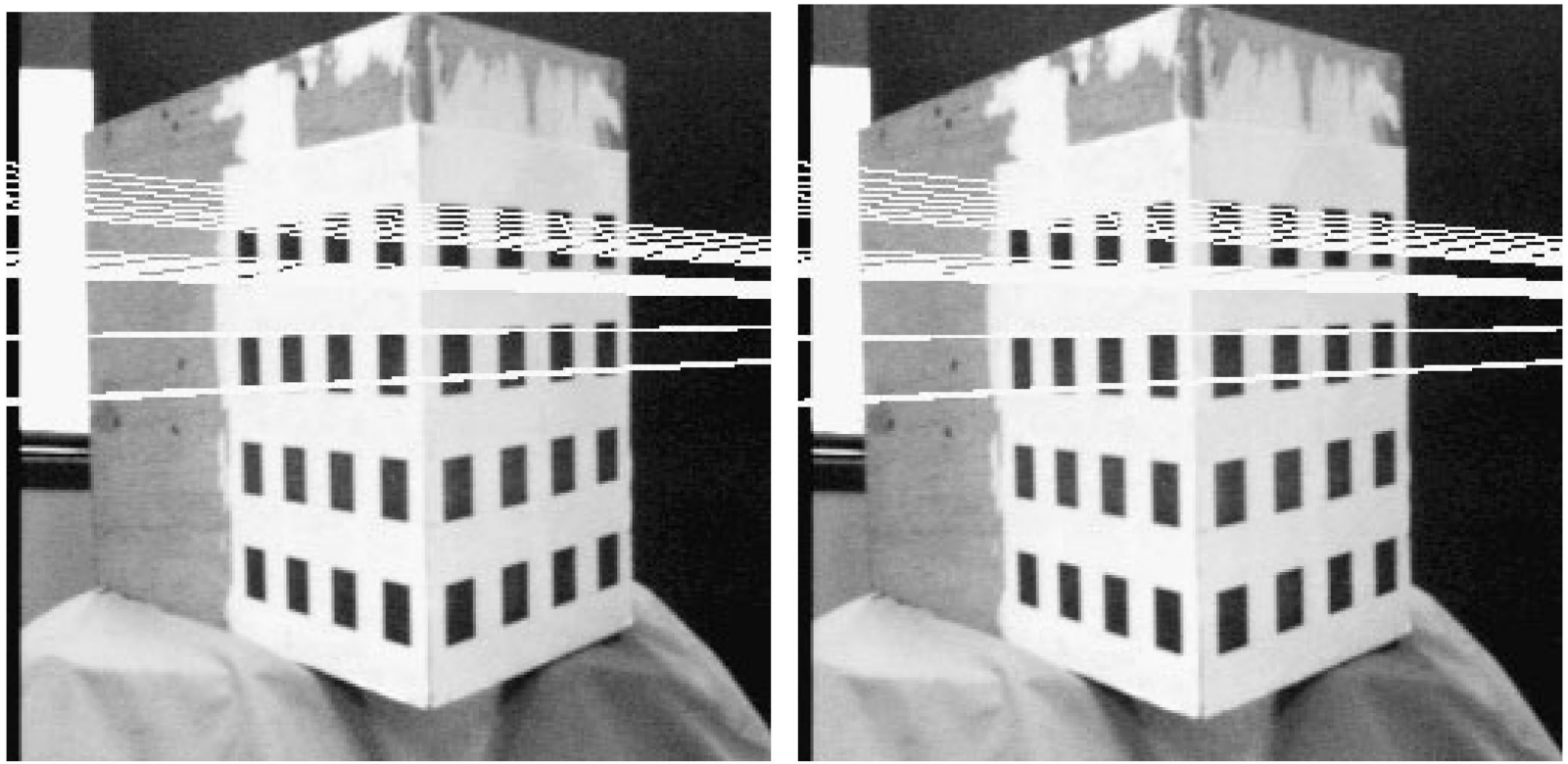
Grenoble Museum





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.

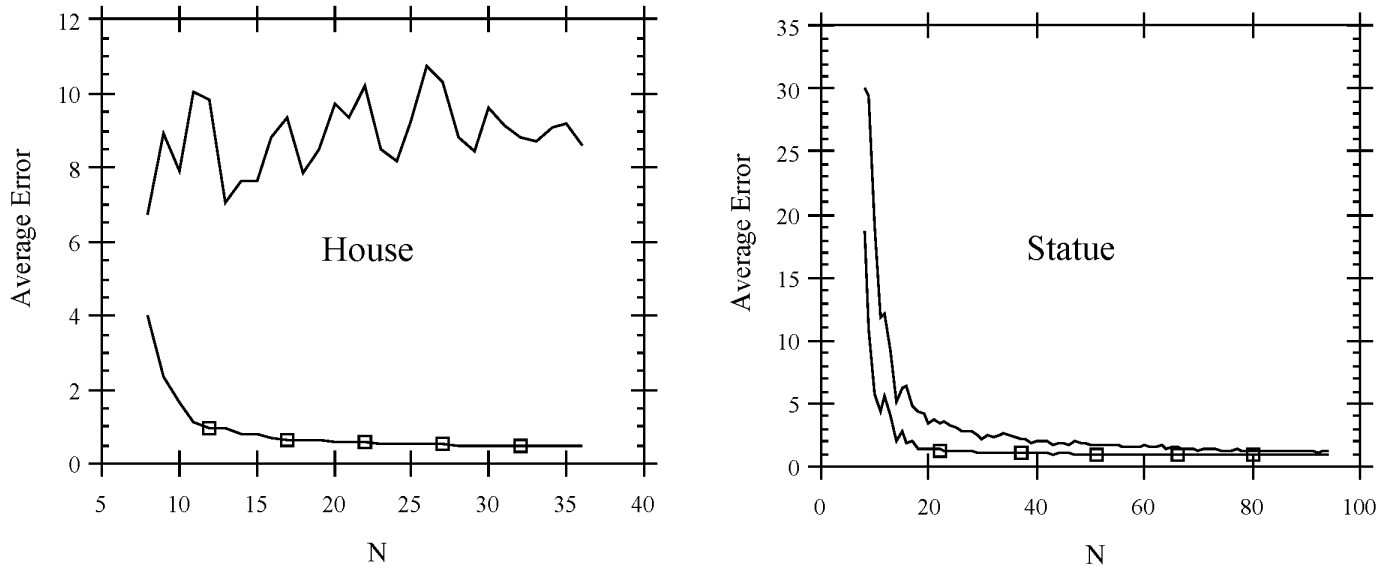
(iii) `F` was tested against other matched points not used to compute it.

(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.

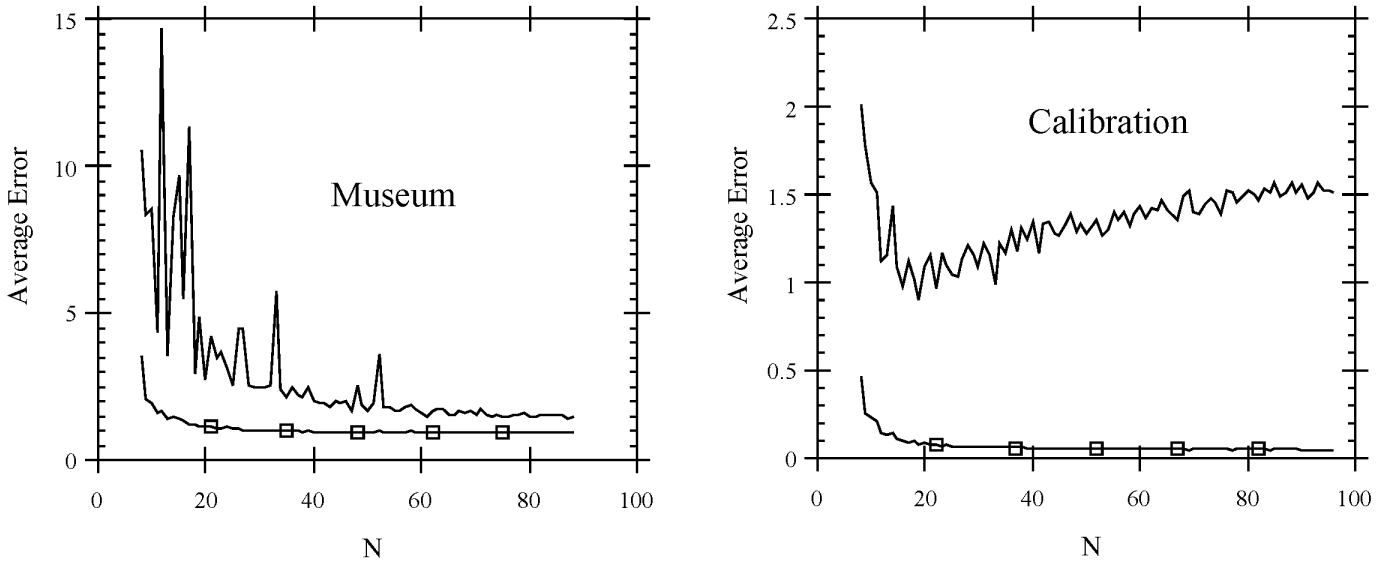


Distance of points from epipolar lines :

Top : Unnormalized 8-point algorithm.

Bottom : Normalized 8-point algorithm.

Comparison of normalized and unnormalized 8-point algorithms.

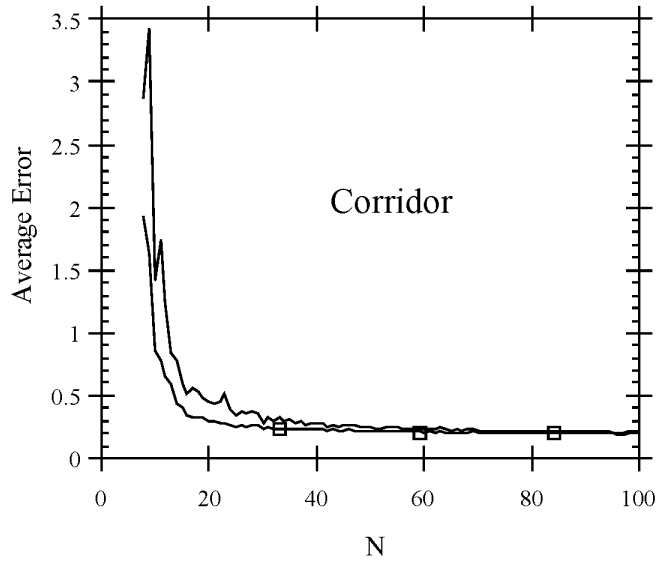


Distance of points from epipolar lines :

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

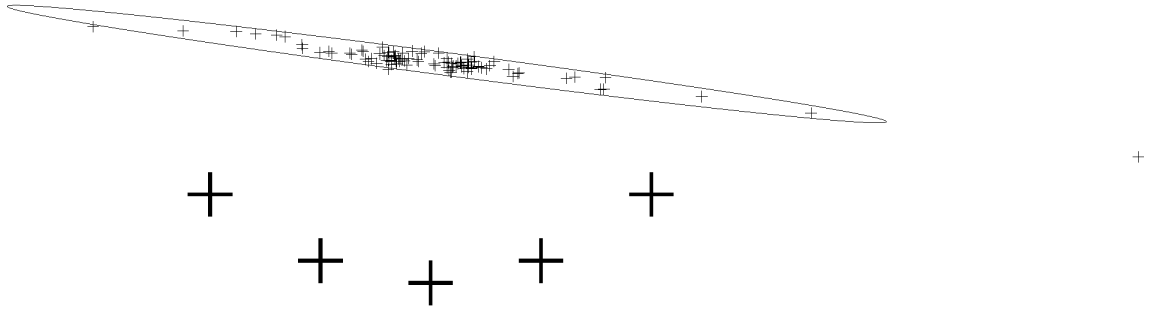
(iii) Noise level approximately equal to with of lines in the crosses (next page)

(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.

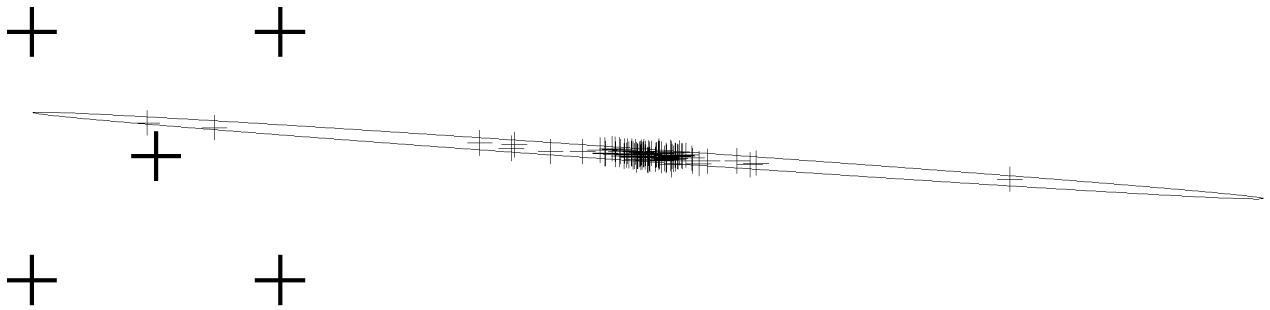
Normalization and 2D homography computation



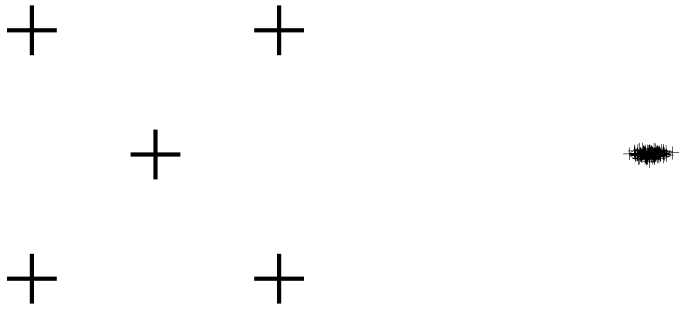
Unnormalized data

Normalized data

Normalization and 2D homography computation



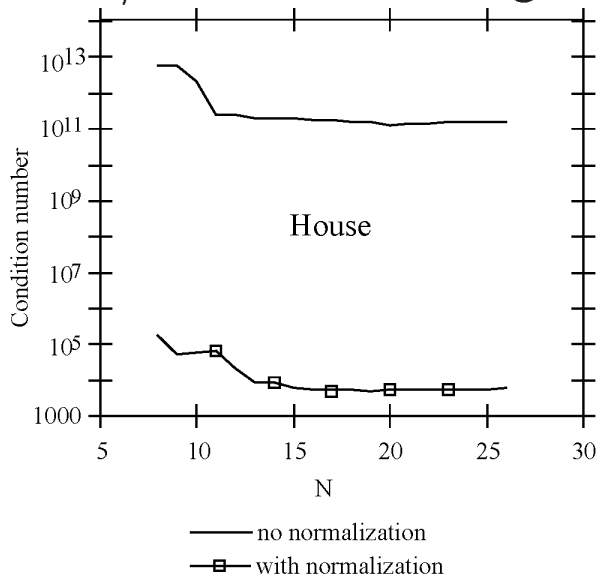
Unnormalized data



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

- 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.