

Multiple View Geometry

Andrew Zisserman, University of Oxford

az@robots.ox.ac.uk

Marc Pollefeys, University of North Carolina at
Chapel Hill

marc@cs.unc.edu

Outline

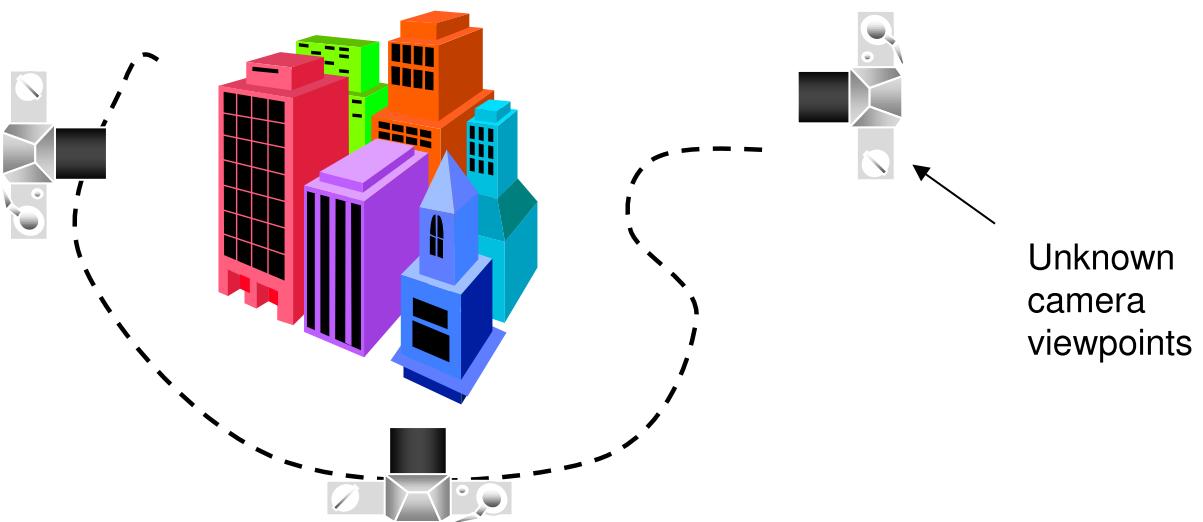
We will look at two types of problem:

1. Transfer: given a point in one image where does it appear in another ?
2. Reconstruction: compute the cameras and 3D structure

- Informal treatment – no proofs or derivations
- Three aspects: geometry, algebra, estimation
- Notes available at
 - www.robots.ox.ac.uk/~az/tutorials
 - www.cs.unc.edu/~marc/tutorial/

The reconstruction problem ...

Given a video sequence



Reconstruct

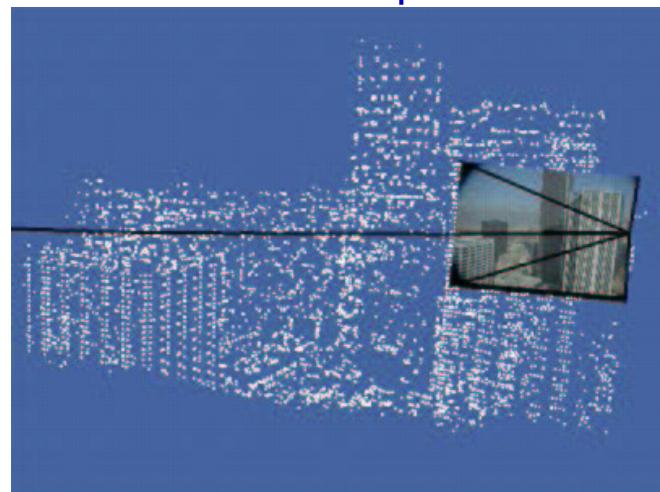
- Scene geometry
- Camera motion

Example

image sequence



cameras and points



Application – movie special effects



Organization

Part 1: Basics and two view geometry

- projective cameras, affine cameras
- 2 view geometry: homographies, epipolar geometry

Part 2: Reconstruction and estimation

- projective reconstruction from 2 views
- triangulation, estimation of the fundamental matrix

Break (9.45-10.15)

Part 3: Three and more views

- trifocal tensor, factorization
- bundle adjustment, auto-calibration

Part 4: Reconstruction from multiple views and applications

- correspondences for sequences
- augmenting video sequences
- surface reconstruction from sequences

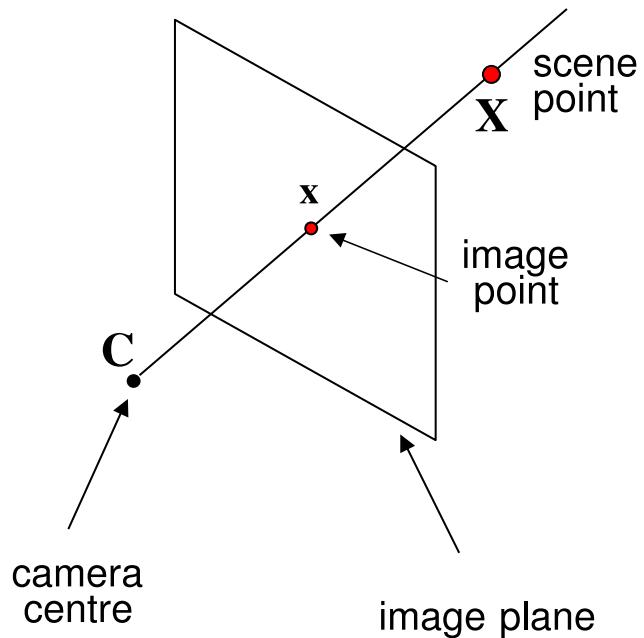
Part I

- projective cameras, affine cameras
- 2 view geometry: homographies, epipolar geometry

Camera geometry

Imaging and camera geometry

- perspective projection
- camera centre, image point and scene point are collinear
- an image point back projects to a ray in 3-space
- depth of the scene point is unknown



Notation – A Projective Camera

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

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

Image Point

$$\lambda x =$$

Scene Point

$$P \quad X$$

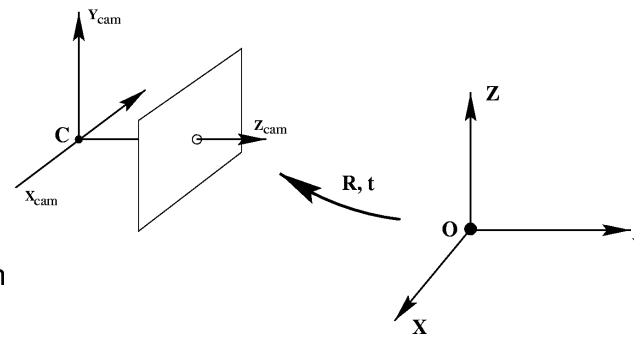
e.g. if $P = [I|0]$ then

$$x = \frac{X}{Z} \quad y = \frac{Y}{Z}$$

- P has 11 degrees of freedom (essential parameters).

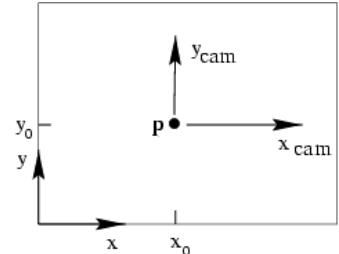
$$P = K [R \mid t]$$

internal calibration rotation translation
from world to camera coordinate frame



K is a 3×3 upper triangular matrix, called the camera calibration matrix:

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

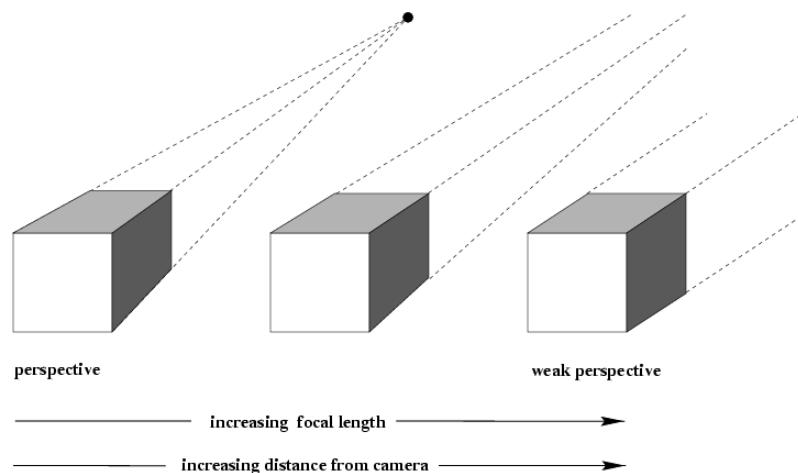


There are five parameters:

1. The **scaling** in the image x and y directions, α_x and α_y .
2. The **principal point** (x_0, y_0) , which is the point where the optic axis intersects the image plane.
3. The **skew** s , which is the angle between the image x and y axes.

Weak Perspective

Track back, whilst zooming to keep the image size fixed



The imaging rays become parallel, and the result is:

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

A generalization is the **affine camera**

The Affine Camera

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

The matrix $M_{2 \times 3}$ has rank two.

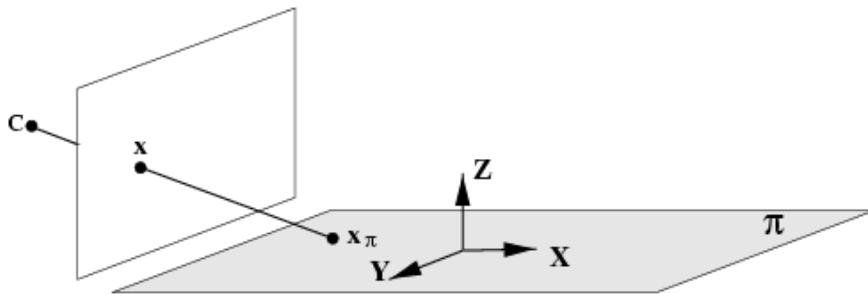
Projection under an affine camera is a linear mapping on non-homogeneous coordinates composed with a translation:

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

- 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.

Planar transformations for image transfer

Plane projective transformations



Choose the world coordinate system such that the plane of the points has zero z coordinate.

Then the 3×4 matrix P reduces to

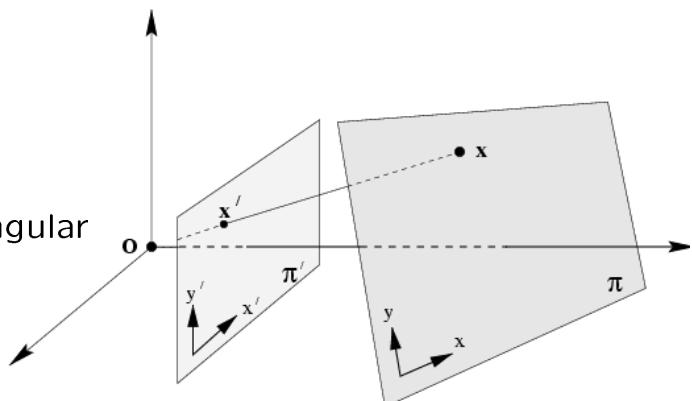
$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} &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

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 $x' = Hx$, where 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''.
- H has 8 degrees of freedom.

Four points define a projective transformation

Given n point correspondences $\{x_i, y_i\} \leftrightarrow \{x'_i, y'_i\}$

Compute H such that $x'_i = Hx_i$

- Each point correspondence gives two constraints

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

and multiplying out generates two linear equations for the elements of H

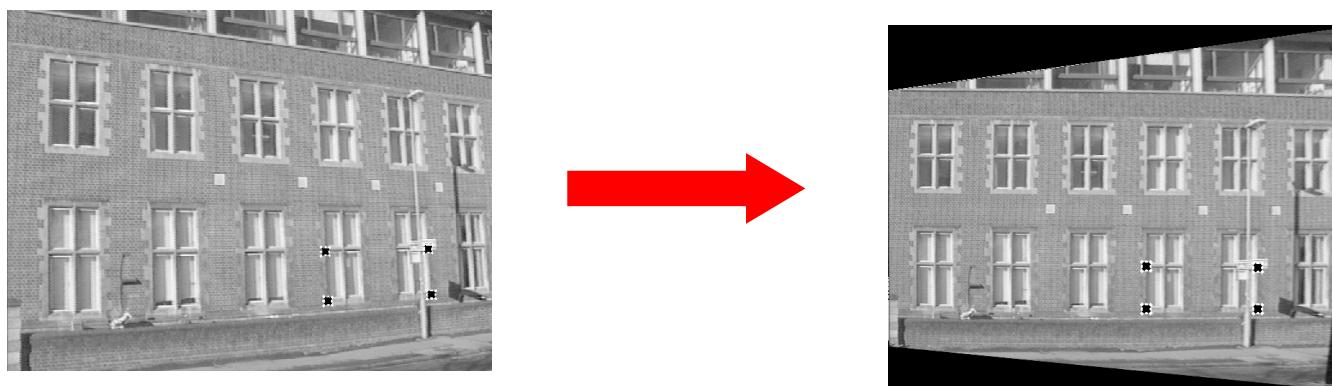
$$\begin{aligned} x'(h_{31}x + h_{32}y + h_{33}) &= h_{11}x + h_{12}y + h_{13} \\ y'(h_{31}x + h_{32}y + h_{33}) &= h_{21}x + h_{22}y + h_{23} \end{aligned}$$

- If $n \geq 4$ (no three points collinear), then 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

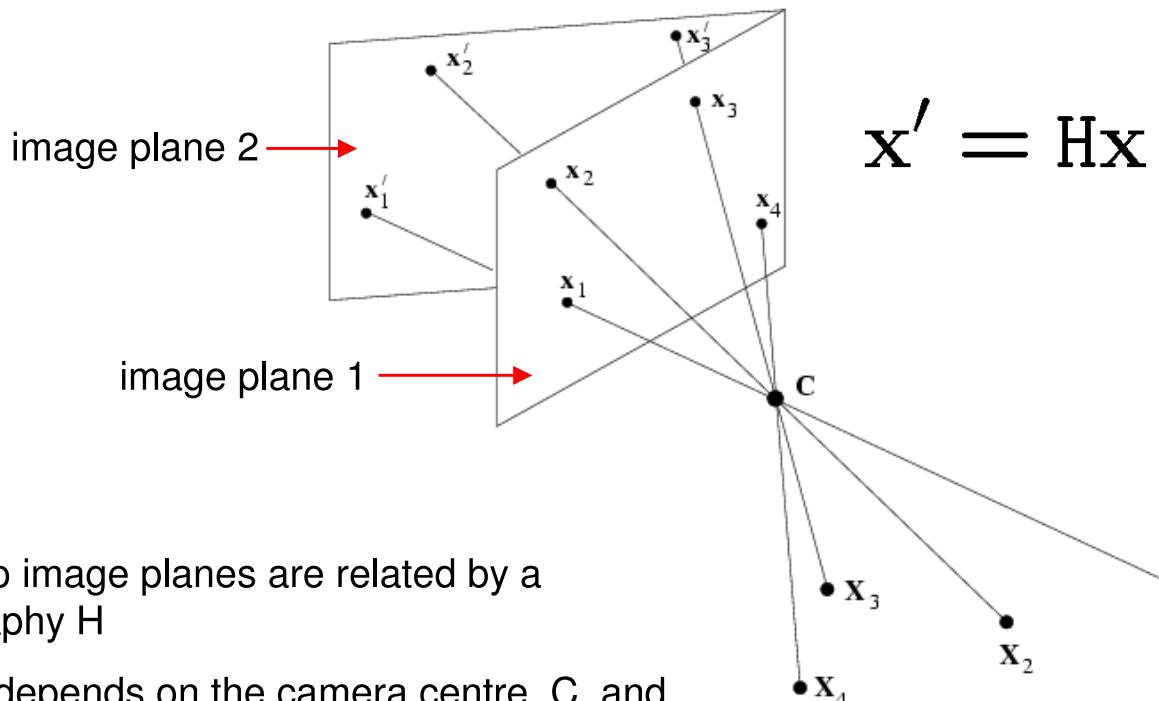
Given: the coordinates of four points on the scene plane

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 importance of the camera centre



- The two image planes are related by a homography H
- H only depends on the camera centre, C , and the planes, **not** on the 3D structure

Example 2: Synthetic Rotations



original

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.

For a rotation $H = K R K^{-1}$

Example III: Building panoramic mosaics

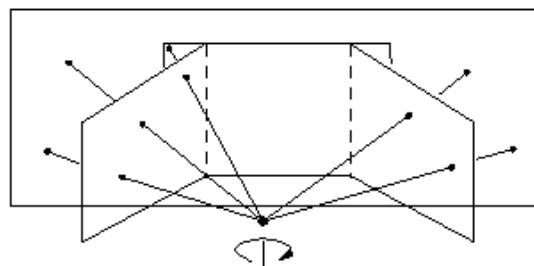
30 frames



The camera rotates (and zooms) with fixed centre

Choice of mosaic frame

Choose central image as reference



This produces the classic "bow-tie" mosaic.



Example IV: Video augmentation



Original- camera panning sequence

Augmented sequence



Two View Geometry

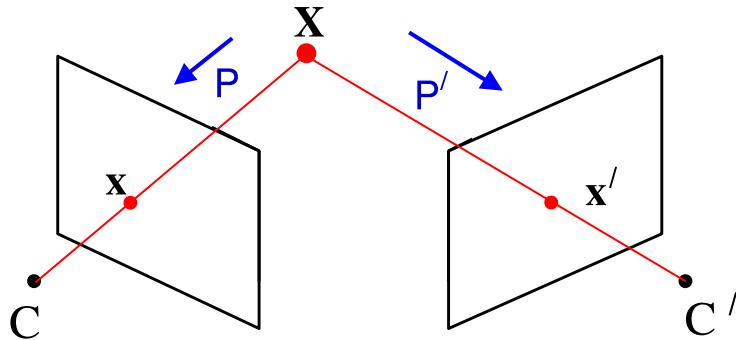
Camera centres **not** coincident

- 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?

Notation

The two cameras are P and P' , and a 3D point \mathbf{X} is imaged as

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



P : 3×4 matrix
 \mathbf{X} : 4-vector
 \mathbf{x} : 3-vector

Warning

for equations involving homogeneous quantities '=' means 'equal up to scale'

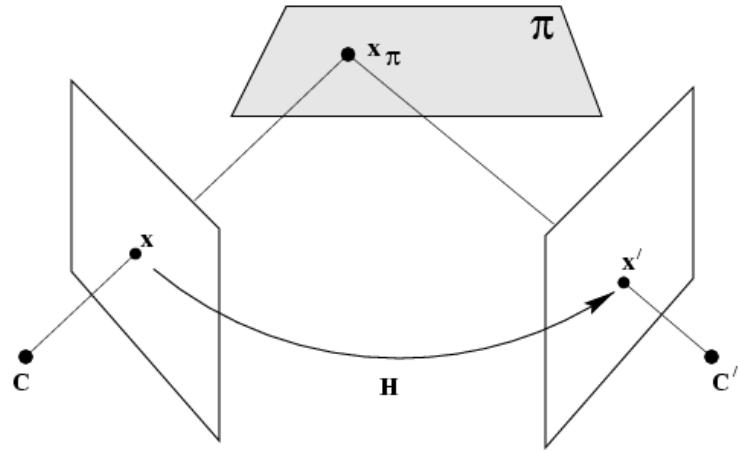
Images of Planes

Projective transformations between images induced by a plane

$$x = H_{1\pi}x_\pi \quad x' = H_{2\pi}x_\pi$$

$$x' = H_{2\pi}x_\pi$$

$$= H_{2\pi}H_{1\pi}^{-1}x = Hx$$

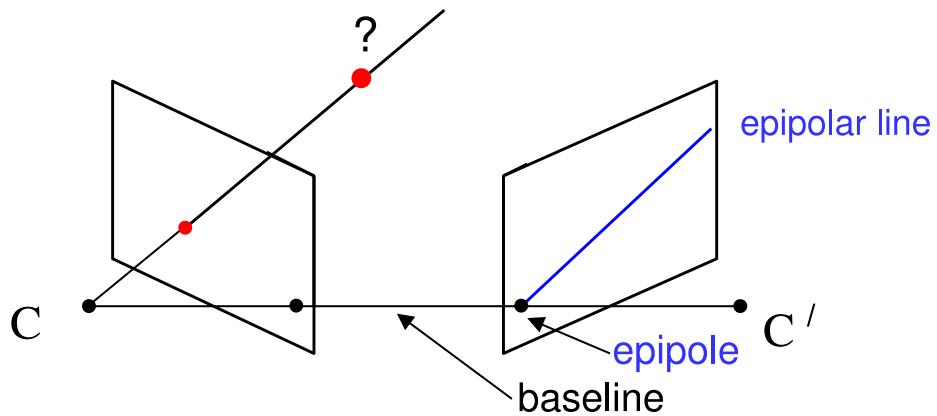


- H can be computed from the correspondence of four points on the plane

Epipolar geometry

Epipolar geometry

Given an image point in one view, where is the corresponding point in the other view?



- A point in one view “generates” an **epipolar line** in the other view
- The corresponding point lies on this line

Epipolar line

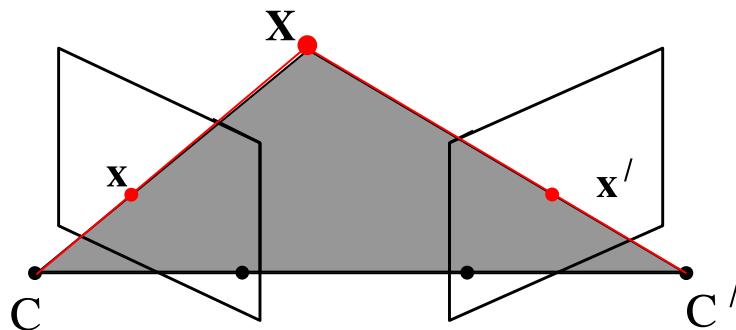


Epipolar constraint

- Reduces correspondence problem to 1D search along an **epipolar line**

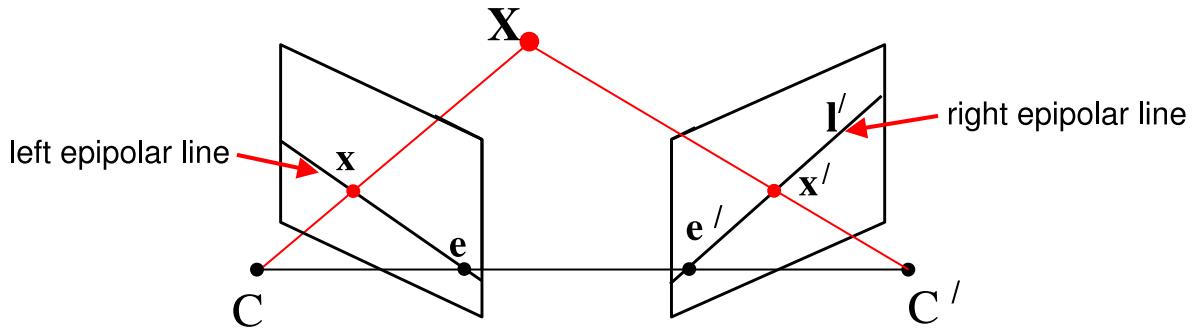
Epipolar geometry continued

Epipolar geometry is a consequence of the **coplanarity** of the camera centres and scene point



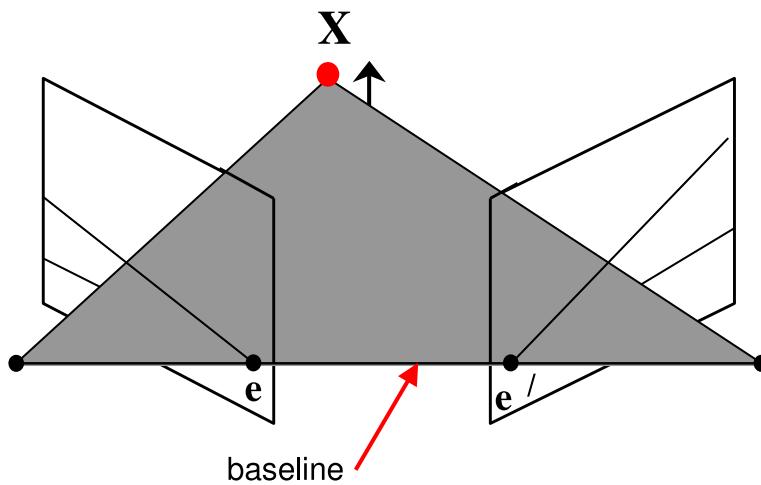
The camera centres, corresponding points and scene point lie in a single plane, known as the **epipolar plane**

Nomenclature



- 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 with the image plane
 - this line is the **baseline** for a stereo rig, and
 - the translation vector for a moving camera
- The epipole is the image of the centre of the other camera: $\mathbf{e} = \mathbf{P}\mathbf{C}'$, $\mathbf{e}' = \mathbf{P}'\mathbf{C}$

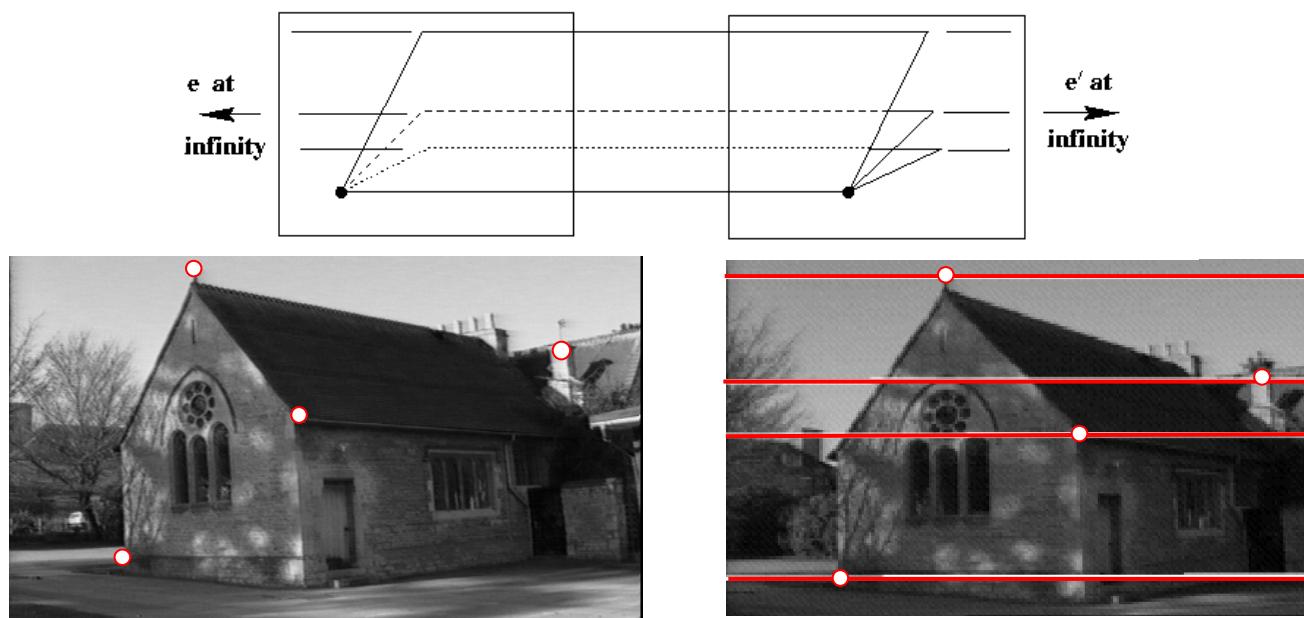
The epipolar pencil



As the position of the 3D point 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.

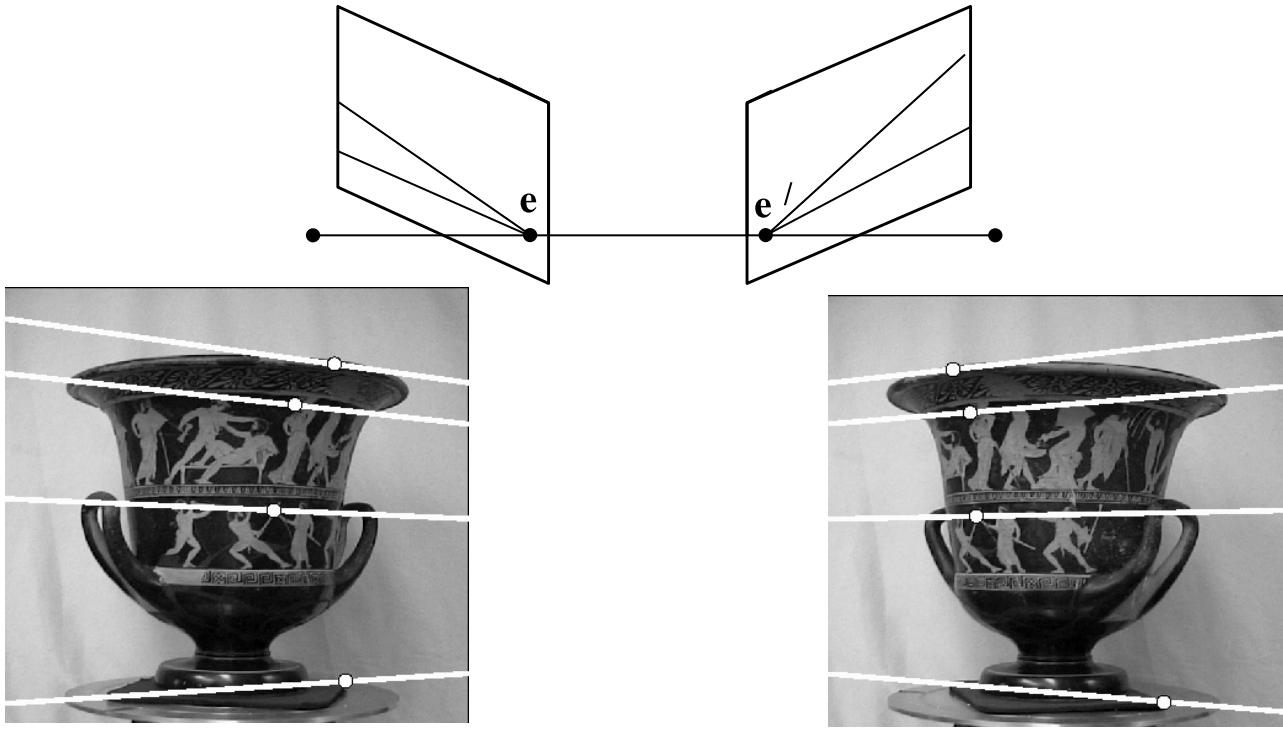
(a pencil is a one parameter family)

Epipolar geometry example I: parallel cameras



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 the scene structure (3D points external to the camera).

Epipolar geometry example II: converging cameras



Note, epipolar lines are in general **not** parallel

Homogeneous notation for lines

Recall that a point (x, y) in 2D is represented by the homogeneous 3-vector $\mathbf{x} = (x_1, x_2, x_3)^\top$, where $x = x_1/x_3, y = x_2/x_3$

A **line** in 2D is represented by the homogeneous 3-vector

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

which is the line $l_1x + l_2y + l_3 = 0$.

Example represent the line $y = 1$ as a homogeneous vector.

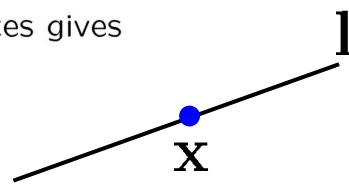
Write the line as $-y + 1 = 0$ then $l_1 = 0, l_2 = -1, l_3 = 1$, and $\mathbf{l} = (0, -1, 1)^\top$.

Note that $\mu(l_1x + l_2y + l_3) = 0$ represents the same line (only the ratio of the homogeneous line coordinates is significant).

Writing both the point and line in homogeneous coordinates gives

$$l_1x_1 + l_2x_2 + l_3x_3 = 0$$

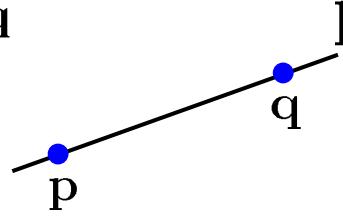
- point on line $\mathbf{l} \cdot \mathbf{x} = 0$ or $\mathbf{l}^\top \mathbf{x} = 0$ or $\mathbf{x}^\top \mathbf{l} = 0$



- The line \mathbf{l} through the two points \mathbf{p} and \mathbf{q} is $\mathbf{l} = \mathbf{p} \times \mathbf{q}$

Proof

$$\mathbf{l} \cdot \mathbf{p} = (\mathbf{p} \times \mathbf{q}) \cdot \mathbf{p} = 0 \quad \mathbf{l} \cdot \mathbf{q} = (\mathbf{p} \times \mathbf{q}) \cdot \mathbf{q} = 0$$



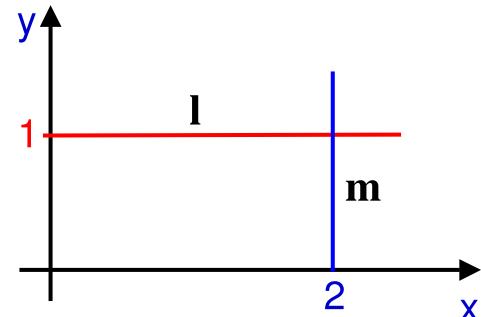
- The intersection of two lines \mathbf{l} and \mathbf{m} is the point $\mathbf{x} = \mathbf{l} \times \mathbf{m}$

Example: compute the point of intersection of the two lines \mathbf{l} and \mathbf{m} in the figure below

$$\mathbf{l} = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} \quad \mathbf{m} = \begin{pmatrix} -1 \\ 0 \\ 2 \end{pmatrix}$$

$$\mathbf{x} = \mathbf{l} \times \mathbf{m} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 0 & -1 & 1 \\ -1 & 0 & 2 \end{vmatrix} = \begin{pmatrix} -2 \\ -1 \\ -1 \end{pmatrix}$$

which is the point $(2, 1)$



Matrix representation of the vector cross product

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

$$\mathbf{v} \times \mathbf{x} = \begin{pmatrix} v_2x_3 - v_3x_2 \\ v_3x_1 - v_1x_3 \\ v_1x_2 - v_2x_1 \end{pmatrix} = [\mathbf{v}]_{\times} \mathbf{x}$$

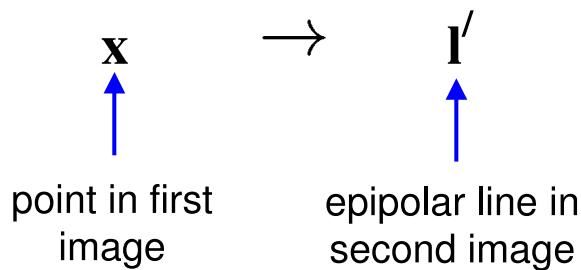
where

$$[\mathbf{v}]_{\times} = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}$$

- $[\mathbf{v}]_{\times}$ 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 of epipolar geometry

We know that the epipolar geometry defines a mapping

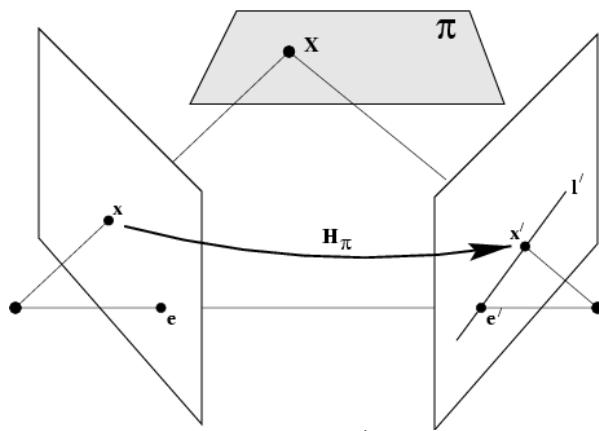


- the map only depends on the cameras P, P' (not on structure)
- it will be shown that the map is linear and can be written as $l' = Fx$ where F is a 3×3 matrix called the fundamental matrix

For corresponding points x and x' ,

$$x'^\top Fx = 0$$

Fundamental matrix – sketch derivation



Step 1: Point transfer via a plane $x' = H_\pi x$

Step 2 : Construct the epipolar line $l' = e' \times x' = [e']_x x'$

$$l' = [e']_x H_\pi x = Fx$$

$$F = [e']_x H_\pi$$

This shows that F is a 3×3 rank 2 matrix.

The Fundamental matrix properties

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

- \mathbf{F} is a 3×3 rank 2 homogeneous matrix
- $\mathbf{F}^\top \mathbf{e}' = \mathbf{0}$
- It has 7 degrees of freedom
- Compute from 7 (or more) image point correspondences
- Compute from camera matrices:
If $\mathbf{P} = [\mathbf{I} \mid \mathbf{0}]$ and $\mathbf{P}' = [\mathbf{M} \mid \mathbf{m}]$, then $\mathbf{F} = [\mathbf{m}]_{\times} \mathbf{M}$

Part 2

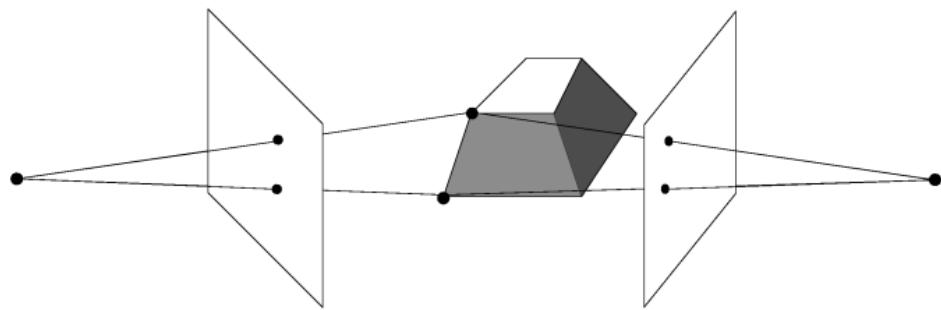
Computing multiview relations (45 min +questions)

- projective reconstruction from two views
- triangulation
- computing the fundamental matrix

Reconstruction from Two Views

Camera centres **not** coincident

Reconstruction from two views



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

Two View Reconstruction Ambiguity – algebra

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

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

Ambiguity

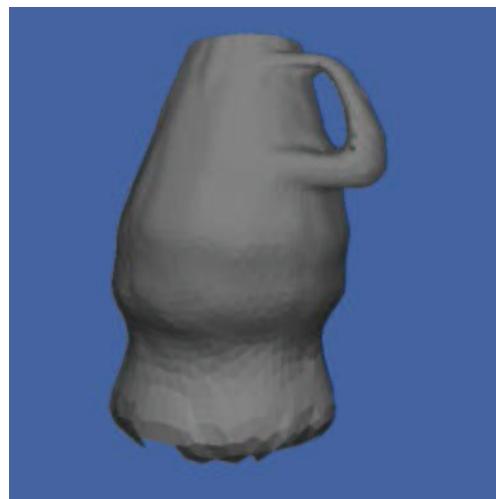
$$\mathbf{x}_i = \mathbf{P}\mathbf{X}_i = \mathbf{P} \mathbf{H}(\mathbf{H})^{-1} \mathbf{X}_i = \tilde{\mathbf{P}}\tilde{\mathbf{X}}_i$$

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

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

Same problem holds however many views we have

Projective ambiguity



How is this ambiguity removed ? See part 3.

Basic Theorem

Given: sufficiently many points to compute unique fundamental matrix

- 8 points in general position

Then: cameras and 3D points may be reconstructed from two views –
up to a 3D projective transformation

- except for points on the line between the camera centres

Projective reconstruction in three steps

1. Compute the epipolar geometry (represented by the fundamental matrix F) from point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$
2. Compute the cameras ([motion](#)) from the matrix F . Obtain

$$P = [I \mid \mathbf{0}] \quad P' = [[\mathbf{e}']_\times F \mid \mathbf{e}'], \quad \text{where } F^\top \mathbf{e}' = \mathbf{0}$$

where P and P' are up to a projective ambiguity

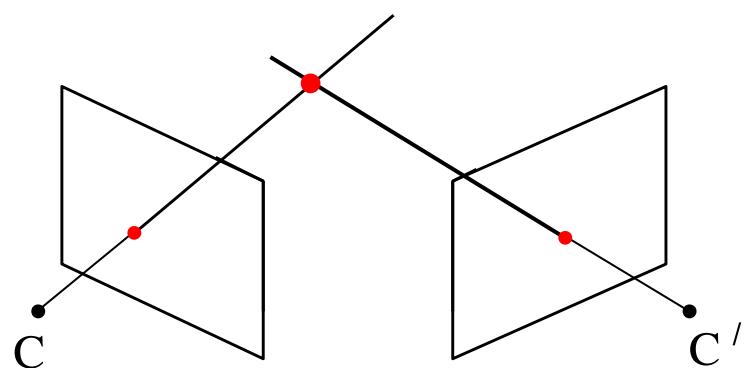
3. Compute the 3D structure \mathbf{X}_i from the cameras P , P' and point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ ([triangulation](#))

Triangulation

Corresponding points are images of the same scene point



Triangulation

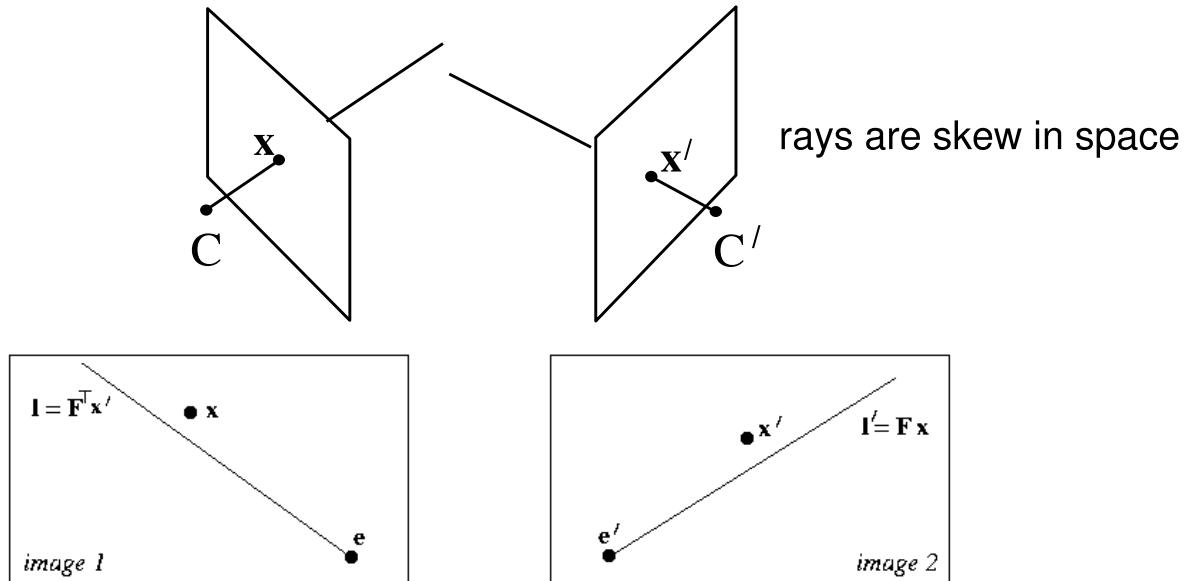


The back-projected points generate rays which intersect at the 3D scene point

Problem statement

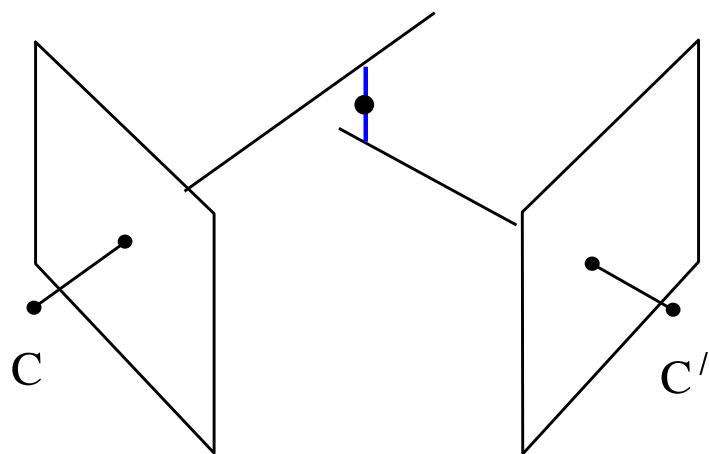
Given: corresponding measured (i.e. noisy) points \mathbf{x} and \mathbf{x}' , and cameras (exact) P and P' , compute the 3D point \mathbf{X}

Problem: in the presence of noise, back projected rays do not intersect



Measured points do **not** lie on corresponding epipolar lines

1. Vector solution



Compute the mid-point of the shortest line between the two rays

2. Linear triangulation (algebraic solution)

Use the equations $\mathbf{x} = \mathbf{P}\mathbf{X}$ and $\mathbf{x}' = \mathbf{P}'\mathbf{X}$ to solve for \mathbf{X}

For the first camera:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} = \begin{bmatrix} \mathbf{p}^{1\top} \\ \mathbf{p}^{2\top} \\ \mathbf{p}^{3\top} \end{bmatrix}$$

where $\mathbf{p}^{i\top}$ are the rows of \mathbf{P}

- eliminate unknown scale in $\lambda\mathbf{x} = \mathbf{P}\mathbf{X}$ by forming a cross product $\mathbf{x} \times (\mathbf{P}\mathbf{X}) = \mathbf{0}$

$$\begin{aligned} x(\mathbf{p}^{3\top}\mathbf{X}) - (\mathbf{p}^{1\top}\mathbf{X}) &= 0 \\ y(\mathbf{p}^{3\top}\mathbf{X}) - (\mathbf{p}^{2\top}\mathbf{X}) &= 0 \\ x(\mathbf{p}^{2\top}\mathbf{X}) - y(\mathbf{p}^{1\top}\mathbf{X}) &= 0 \end{aligned}$$

- rearrange as (first two equations only)

$$\begin{bmatrix} xp^{3\top} - p^{1\top} \\ yp^{3\top} - p^{2\top} \end{bmatrix} \mathbf{X} = \mathbf{0}$$

Similarly for the second camera:

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

Collecting together gives

$$\mathbf{A}\mathbf{X} = \mathbf{0}$$

where \mathbf{A} is the 4×4 matrix

$$\mathbf{A} = \begin{bmatrix} xp^{3\top} - p^{1\top} \\ yp^{3\top} - p^{2\top} \\ x'p'^{3\top} - p'^{1\top} \\ y'p'^{3\top} - p'^{2\top} \end{bmatrix}$$

from which \mathbf{X} can be solved up to scale.

Problem: does not minimize anything meaningful

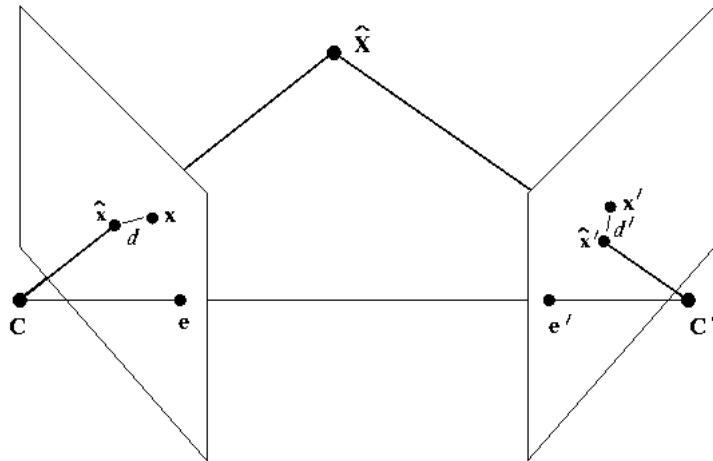
Advantage: extends to more than two views

3. Minimizing a geometric/statistical error

The idea is to estimate a 3D point $\hat{\mathbf{x}}$ which exactly satisfies the supplied camera geometry, so it projects as

$$\hat{\mathbf{x}} = \mathbf{P}\hat{\mathbf{X}} \quad \hat{\mathbf{x}}' = \mathbf{P}'\hat{\mathbf{X}}$$

and the aim is to estimate $\hat{\mathbf{x}}$ from the image measurements \mathbf{x} and \mathbf{x}' .



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

where $d(*, *)$ is the Euclidean distance between the points.

- It can be shown that if the measurement noise is Gaussian mean zero, $\sim N(0, \sigma^2)$, then minimizing geometric error is the **Maximum Likelihood Estimate** of \mathbf{X}
- The minimization appears to be over three parameters (the position \mathbf{X}), but the problem can be reduced to a minimization over one parameter

Projective reconstruction in three steps

1. Compute the epipolar geometry (represented by the fundamental matrix F) from point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$
2. Compute the cameras ([motion](#)) from the matrix F . Obtain
$$P = [I \mid \mathbf{0}] \quad P' = [[\mathbf{e}']_\times F \mid \mathbf{e}'], \quad \text{where } F^\top \mathbf{e}' = \mathbf{0}$$
where P and P' are up to a projective ambiguity
3. Compute the 3D structure \mathbf{X}_i from the cameras P , P' and point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ ([triangulation](#))

Computing the fundamental matrix

F computation – various options

- Linear method (given 8 or more point correspondences)
- Non-linear method (7 point correspondences)
- Automatic estimation – obtain point correspondences and estimate F
- Maximum likelihood estimate

Problem statement

Given: n corresponding points $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i, i = 1, \dots, n\}$
compute the fundamental matrix \mathbf{F} such that

$$\mathbf{x}'_i^\top \mathbf{F} \mathbf{x}_i = 0 \quad 1 \leq i \leq n$$

Solution

Each point correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ generates one constraint on \mathbf{F}

$$(x'_i \ y'_i \ 1) \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = 0$$

which may be written

$$x'x f_1 + x'y f_2 + x'f_3 + y'x f_4 + y'y f_5 + y'f_6 + x f_7 + y f_8 + f_9 = 0$$

$$(x'x, \ x'y, \ x', \ y'x, \ y'y, \ y', \ x, \ y, \ 1) \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{pmatrix} = 0$$

For n points

$$Af = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{pmatrix} = 0$$

A is an $n \times 9$ measurement matrix, and \mathbf{f} is the fundamental matrix written as a 9-vector

- For 8 points, A is an 8×9 matrix and \mathbf{f} can be computed as the null-vector of A, i.e. \mathbf{f} is determined up to scale

This method is known as the “8 point algorithm”

More on the 8 point algorithm

For $n > 8$ point correspondences, A is a $n \times 9$ matrix,

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

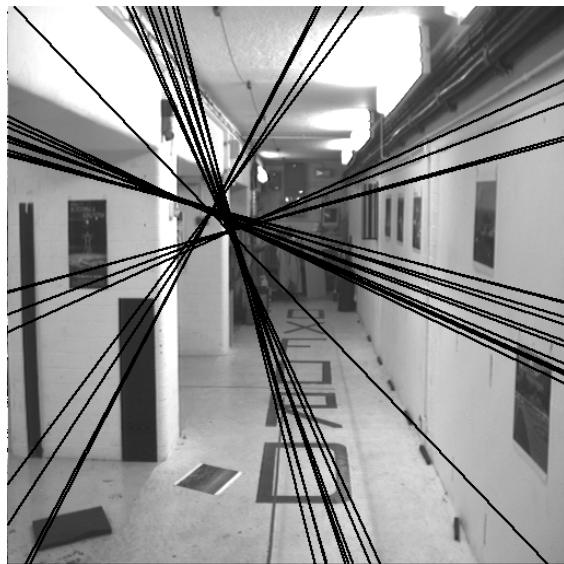
There is not an exact solution, \Rightarrow a least-squares solution

Least-squares solution

1. Minimize $\|Af\|$ subject to $\|\mathbf{f}\| = 1$.
2. Take Singular Value Decomposition (SVD) : $A = UDV^\top$.
3. Solution is last column of V (corresp : smallest singular value)

The singularity constraint

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



F not rank 2

epipolar lines are not coincident



F has rank 2

- F estimated by the 8-point algorithm will **not** satisfy this constraint in general

Enforcing the singularity constraint via the SVD

1. SVD : $F = UDV^\top$
2. U and V are orthogonal, $D = \text{diag}(r, s, t)$.
3. $r \geq s \geq t$.
4. Set $F' = U \text{diag}(r, s, 0) V^\top$.
5. Resulting F' is singular.
6. Minimizes the Frobenius norm of $F - F'$
7. F' is the "closest" singular matrix to F .

Solution for 7 points

1. Form the 7×9 set of equations $\mathbf{Af} = \mathbf{0}$
2. The system has a 2-dimensional solution set
3. General solution (use SVD) has the form

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

4. In matrix terms

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

5. Condition $\det \mathbf{F} = 0$ gives cubic equation in λ and μ
6. Either one or three real solutions for ratio $\lambda : \mu$

Automatic Computation of the fundamental matrix

First image



Second image



Given Image pair



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

- Compute image points
- Compute correspondences
- Compute epipolar geometry

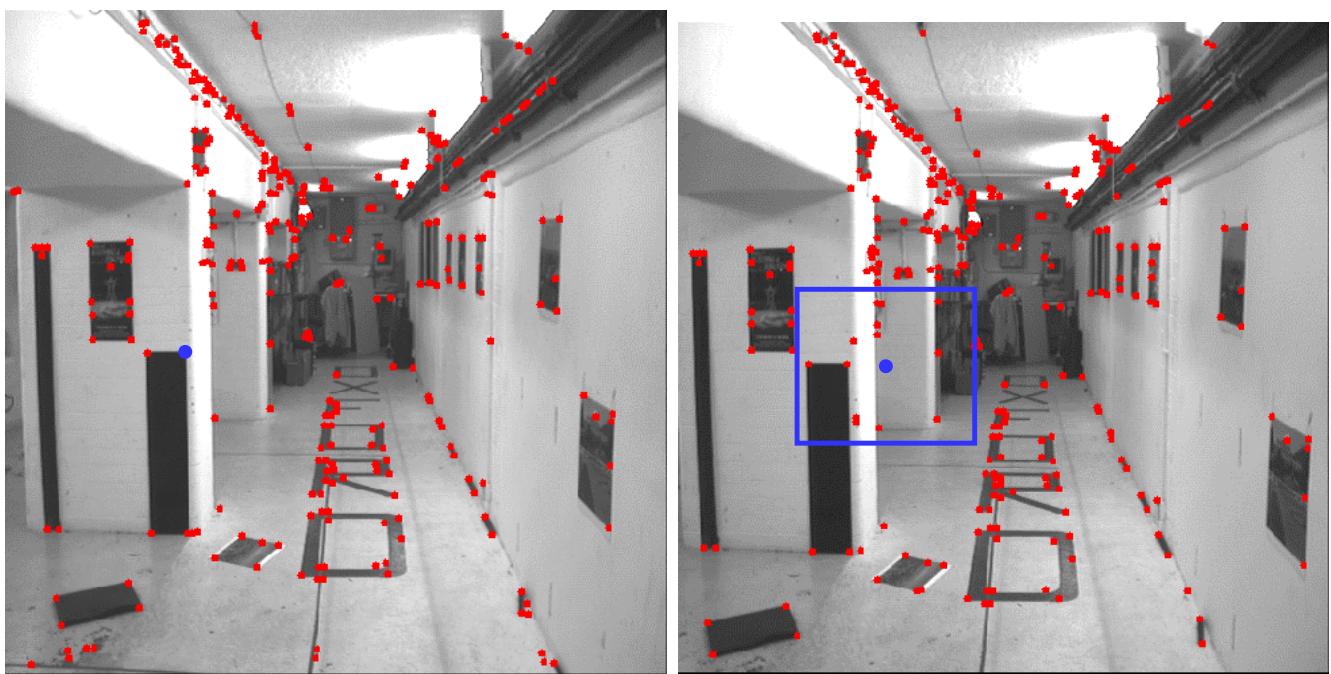
Step 1: interest points



Harris corner detector

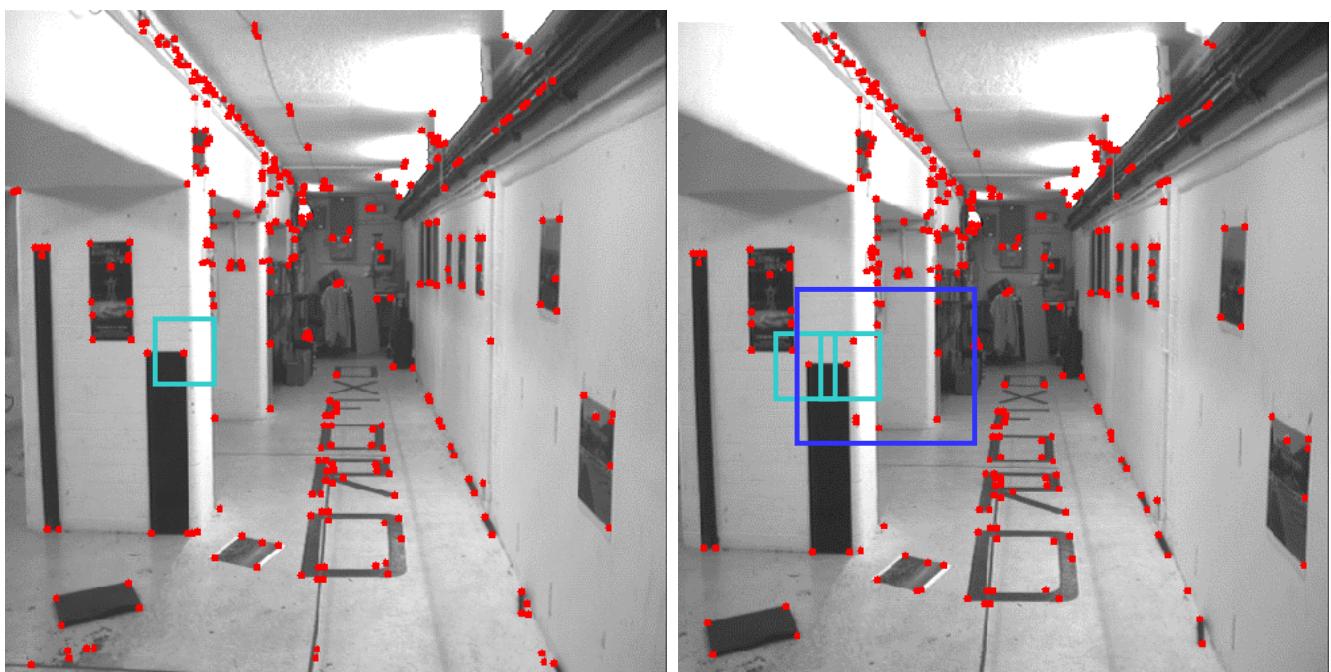
100's of points per image

Step 2a: match points – proximity



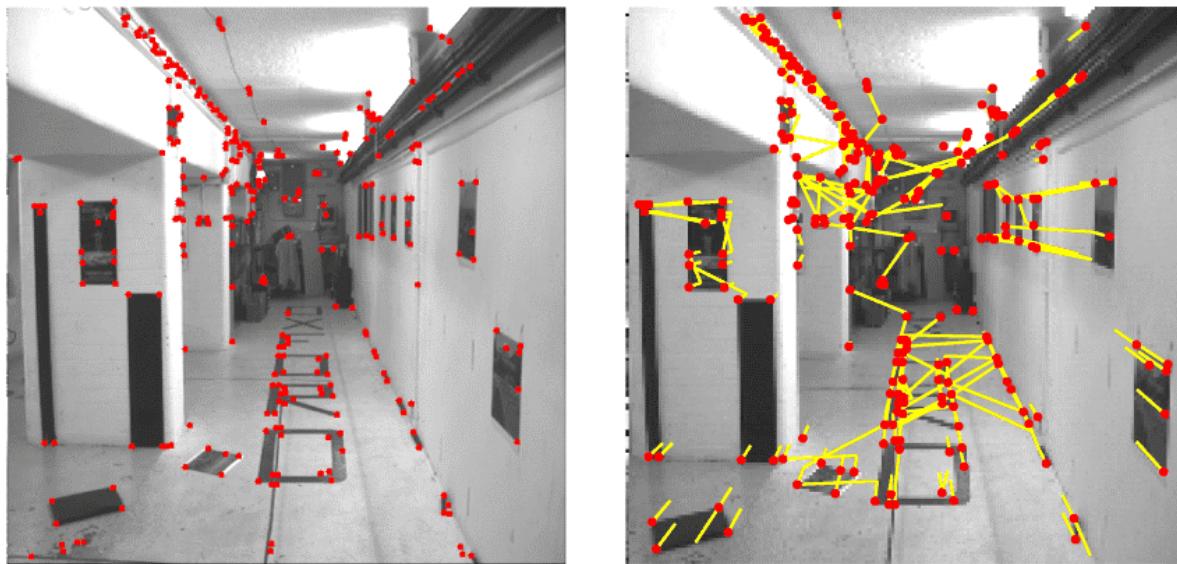
- proximity - search within disparity window

Step 2b: match points – cross-correlate



- cross-correlate on intensity neighbourhoods

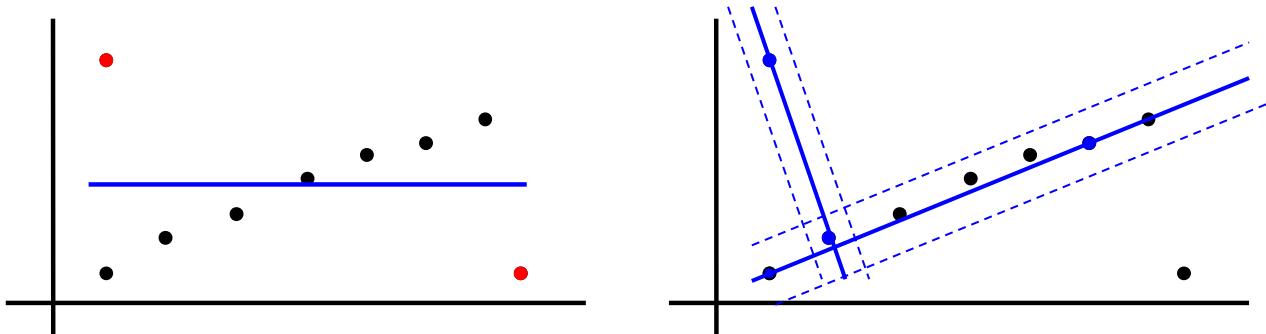
Correlation matching results



- Many wrong matches (10-50%), but enough to compute F

Robust line estimation - RANSAC

Fit a line to 2D data containing outliers



There are two problems

1. a line **fit** which minimizes perpendicular distance
2. a **classification** into inliers (valid points) and outliers

Solution: use robust statistical estimation algorithm RANSAC

(RANdom Sample Consensus) [Fischler & Bolles, 1981]

RANSAC robust line estimation

Repeat

1. Select random sample of 2 points
2. Compute the line through these points
3. Measure support (number of points within threshold distance of the line)

Choose the line with the largest number of inliers

- Compute least squares fit of line to inliers (regression)

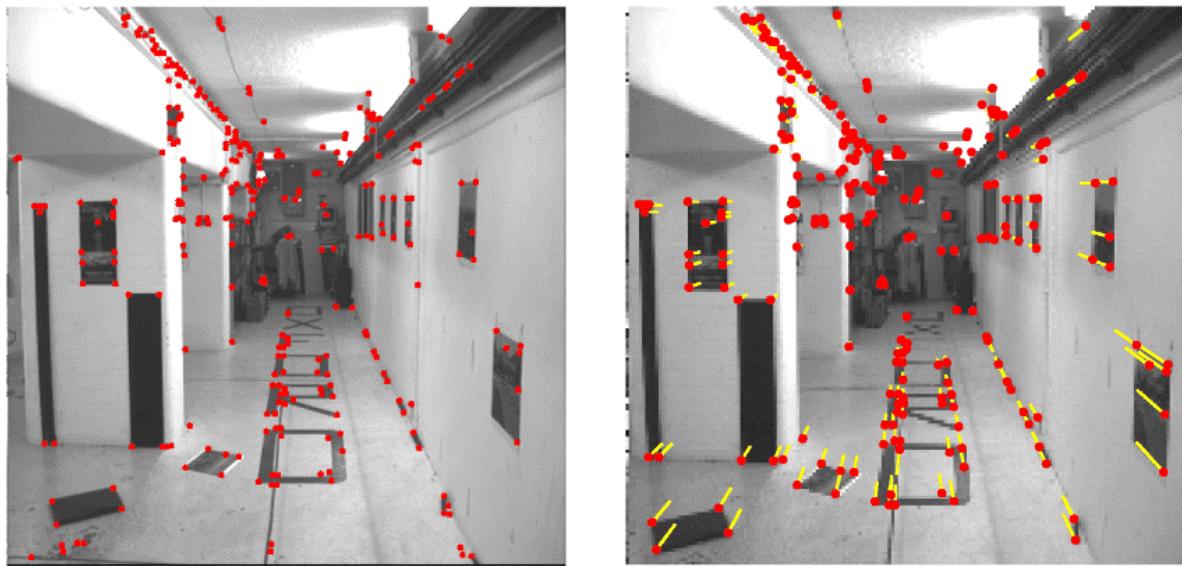
Algorithm summary – RANSAC robust F estimation

Repeat

1. Select random sample of 7 correspondences
2. Compute F (1 or 3 solutions)
3. Measure support (number of inliers within threshold distance of epipolar line)

Choose the F with the largest number of inliers

Correspondences consistent with epipolar geometry



- Use **RANSAC** robust estimation algorithm
- Obtain correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ **and** \mathbf{F}

Computed epipolar geometry

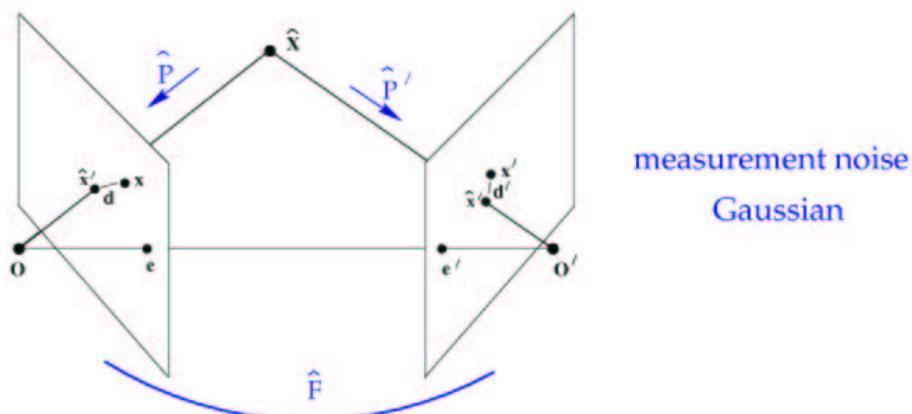


Projective reconstruction in three steps

1. Compute the epipolar geometry (represented by the fundamental matrix F) from point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$
2. Compute the cameras (**motion**) from the matrix F . Obtain
$$P = [I \mid \mathbf{0}] \quad P' = [[\mathbf{e}'] \times F \mid \mathbf{e}'], \quad \text{where } F^\top \mathbf{e}' = \mathbf{0}$$
where P and P' are up to a projective ambiguity
3. Compute the 3D structure \mathbf{X}_i from the cameras P , P' and point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ (**triangulation**)

Optimal Estimation of Reconstruction (2-views)

Given: $n > 7$ imperfectly measured image point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, compute a **Maximum Likelihood Estimate**



$$\min_{\hat{\mathbf{F}}, \hat{\mathbf{X}}_i} \sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2 + d(\mathbf{x}'_i, \hat{\mathbf{x}}'_i)^2 \quad \text{with } \hat{\mathbf{x}}_i^\top \mathbf{F} \hat{\mathbf{x}}_i = 0$$

- Parametrize F as rank 2 matrix, e.g. $[\mathbf{e}'] \times M$, 12 parameters
- Minimize over $4n + 12$ parameters using Levenberg–Marquardt

Summary: computing multiple view relations

We have seen that F can be computed from n point correspondences starting from $\mathbf{x}_i'^\top F \mathbf{x}_i = 0$

- $n \geq 8$: linear solution of form $Ax = 0$, but have to impose constraint
- $n = 7$: minimal solution, non-linear but obeys constraint
- Maximum Likelihood Estimation (MLE), robust estimation

Similar issues apply in estimating other multiple view relations, e.g.

- **Homography H** : estimate from n image-image point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ starting from $\mathbf{x}'_i = H \mathbf{x}_i$.
 - $n \geq 4$: linear solution, no internal constraints
 - MLE, robust estimation
- **Camera resectioning**: Estimate P from world-image point correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ starting from $\mathbf{x}_i = P \mathbf{X}_i$
 - Require $n \geq 5.5$.
- **Trifocal tensor** for three views: Estimate from n image-image-image point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i$
 - $n \geq 7$: linear solution of form $Ax = 0$, but have to impose eight constraints
 - $n = 6$ minimal solution, non-linear but obeys constraints
 - MLE, robust estimation