Multiple View Geometry

CS 6320, Spring 2013

Guest Lecture Marcel Prastawa

adapted from Pollefeys, Shah, and Zisserman

Single view computer vision

- Projective actions of cameras
- Camera callibration

- Photometric stereo (geometrically single view, with multiple lightings)

Multi view computer vision

Two (or more) images, from

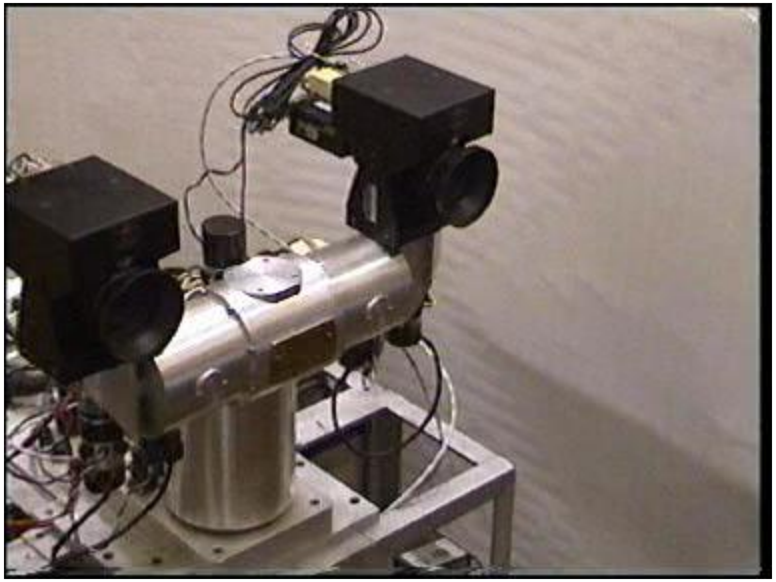
- A stereo rig consisting of two cameras
  - the two images are acquired simultaneously

or

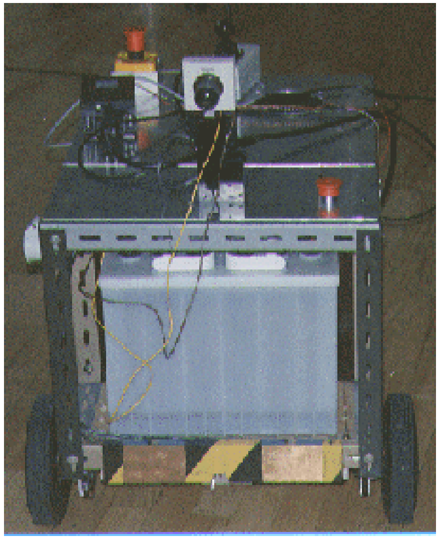
- A single moving camera (static scene)
  - the two images are acquired sequentially

The two scenarios are geometrically equivalent

Stereo head

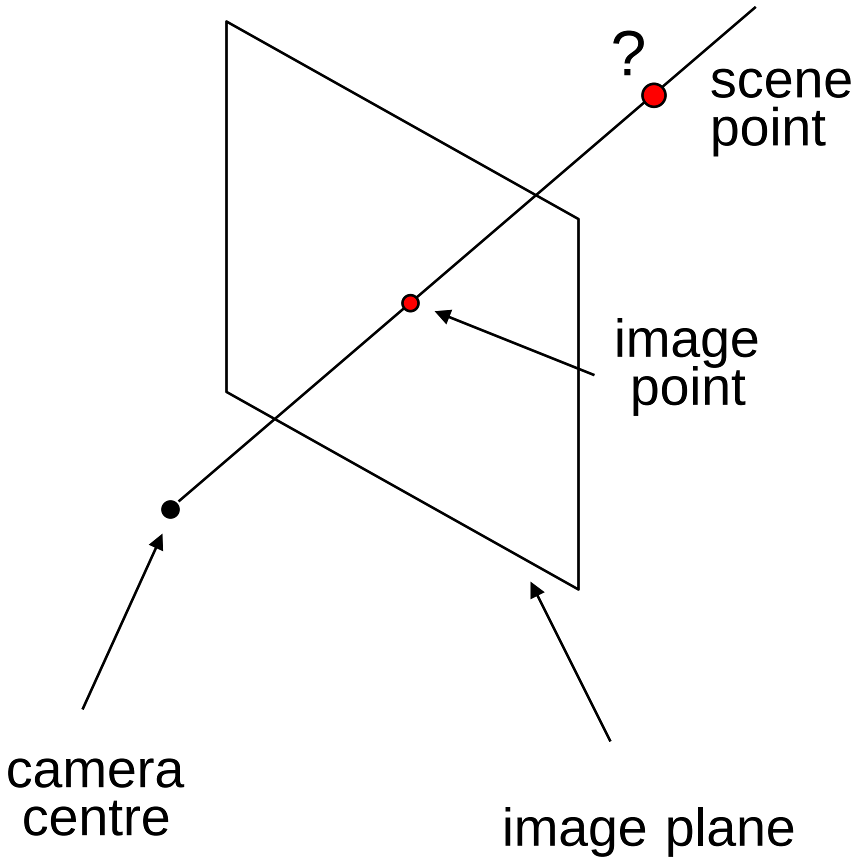


Camera on a mobile vehicle



Imaging geometry

- central 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



The objective

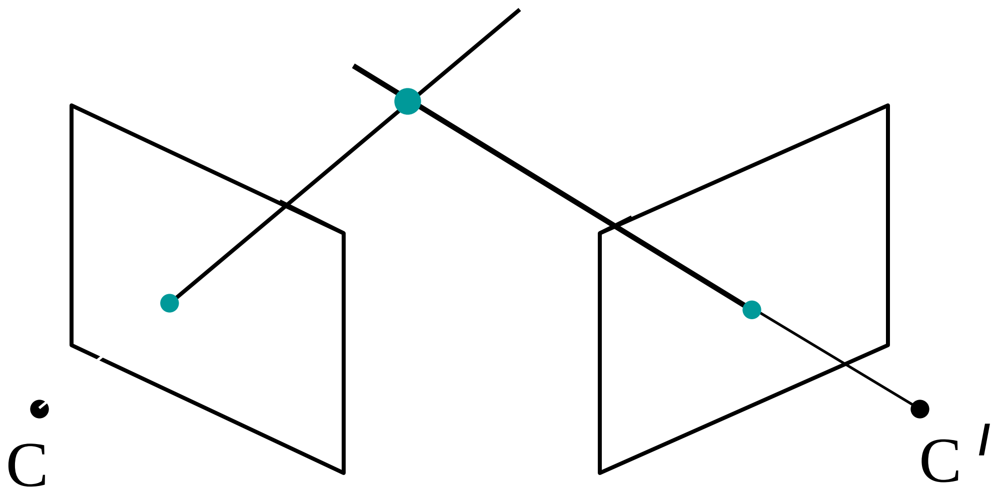
Given two images of a scene acquired by known cameras compute the 3D position of the scene (structure recovery)

Basic principle: triangulate from corresponding image points

- Determine 3D point at intersection of two back-projected rays

Corresponding points are images of the same scene point

Triangulation



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

An algorithm for stereo reconstruction

1. For each point in the first image determine the corresponding point in the second image

   (this is a search problem)

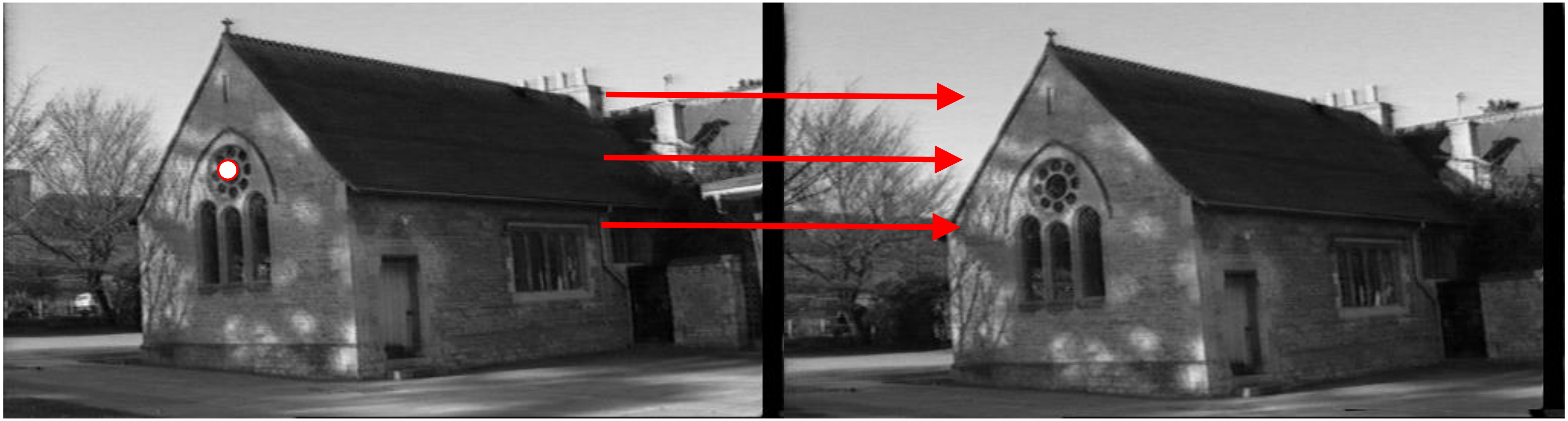
2. For each pair of matched points determine the 3D point by triangulation

   (this is an estimation problem)

The correspondence problem

Given a point x in one image find the corresponding point in the other image

Example with translation:

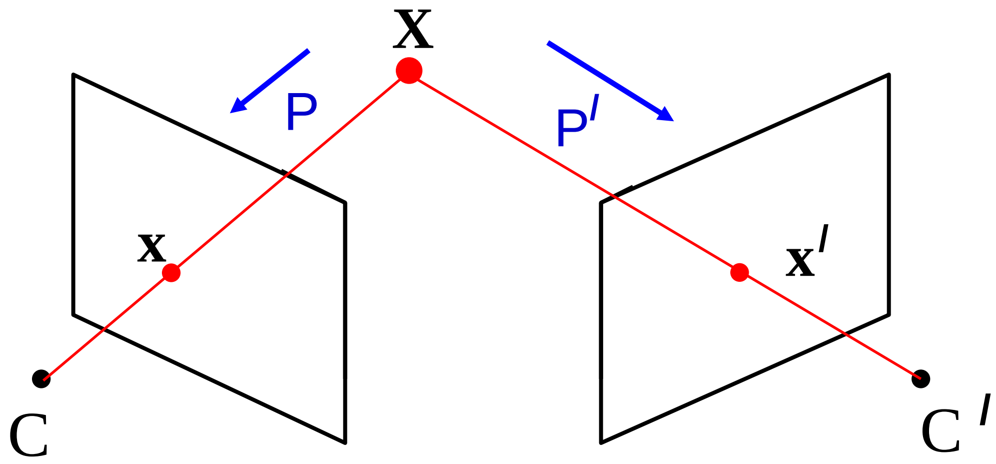


This appears to be a 2D search problem, but it is reduced to a 1D search by the epipolar constraint

Notation

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

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



$\mathtt{P}$: 3×4 matrix

$\mathbf{X}$: 4-vector

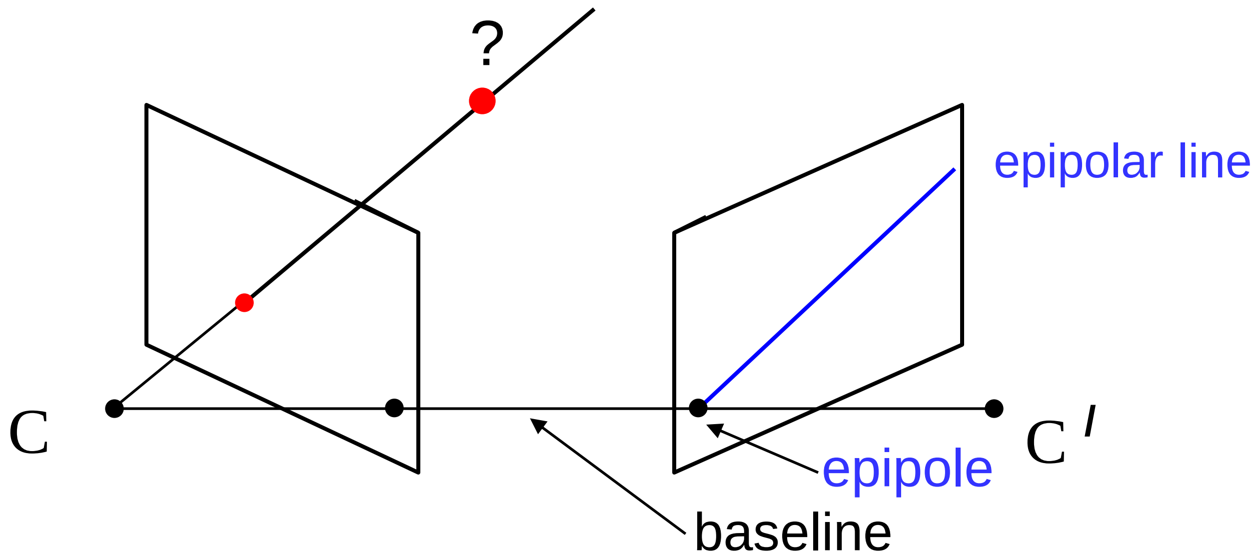
$\mathbf{x}$: 3-vector

Warning

for equations involving homogeneous quantities ‘=’ means ‘equal up to scale’

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

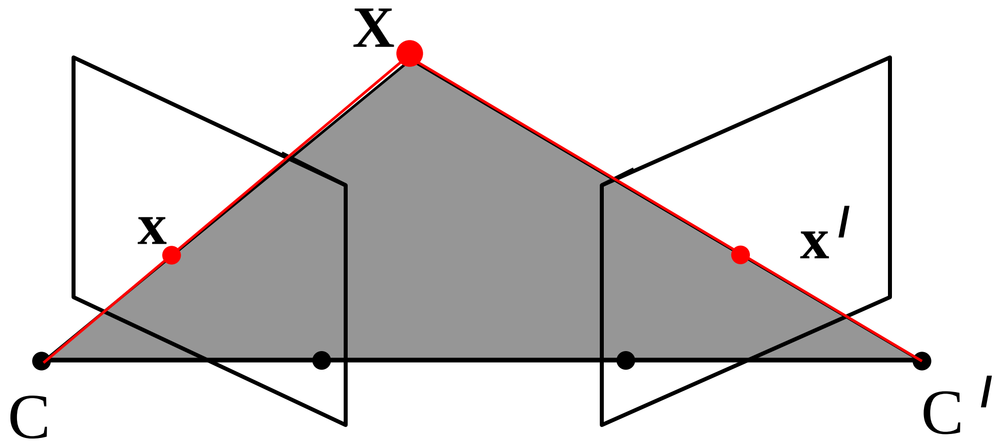
Epipolar line

Epipolar constraint

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

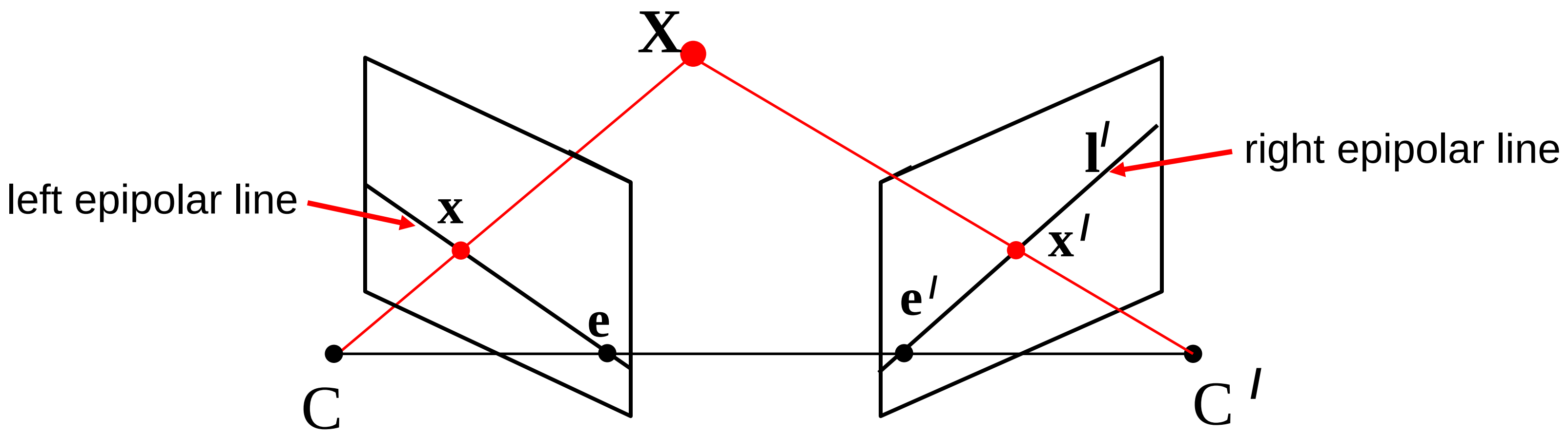
Epipolar geometry continued

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



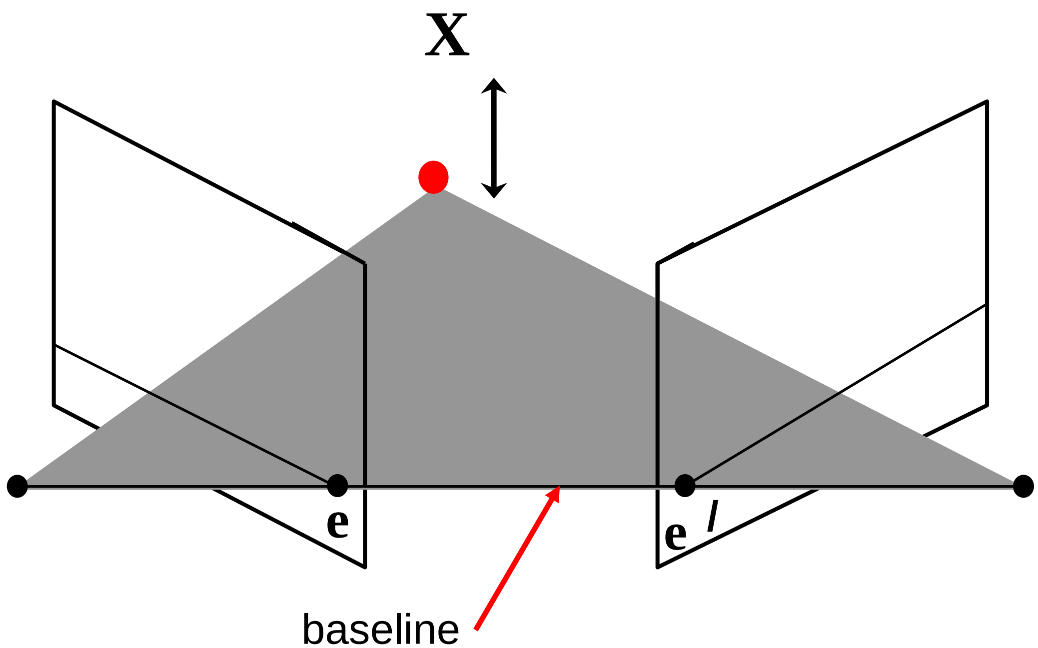
The camera centres, corresponding points and scene point lie in a single plane, known as the epipolar plane

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: $\mathbf{e} = \mathrm{P}\mathbf{C}'$, $\mathbf{e}' = \mathrm{P}'\mathbf{C}$

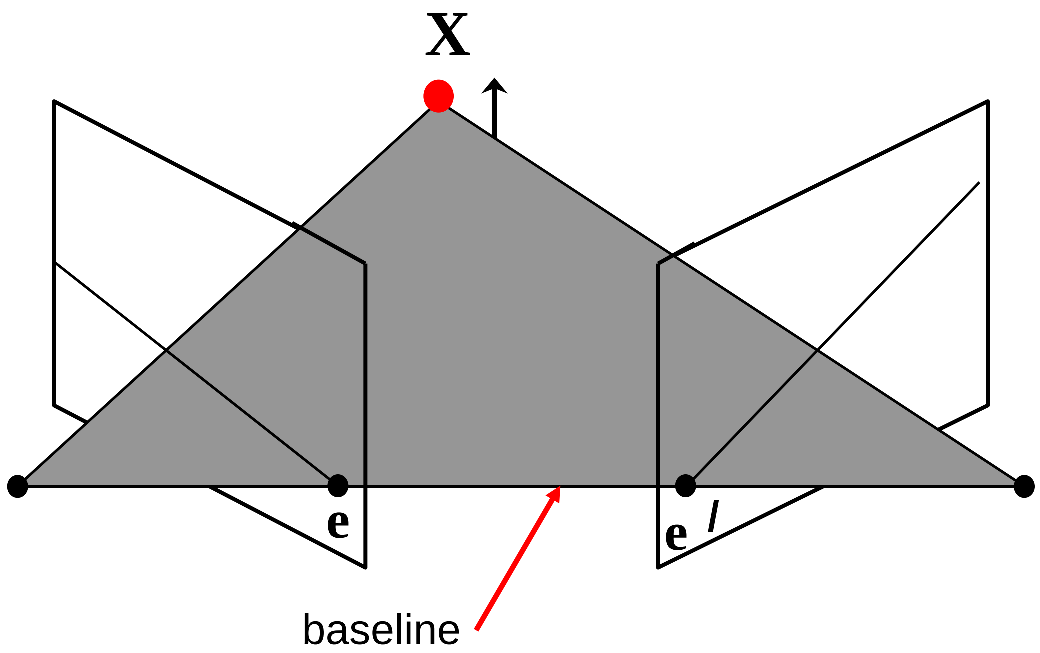
The epipolar pencil



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.

(a pencil is a one parameter family)

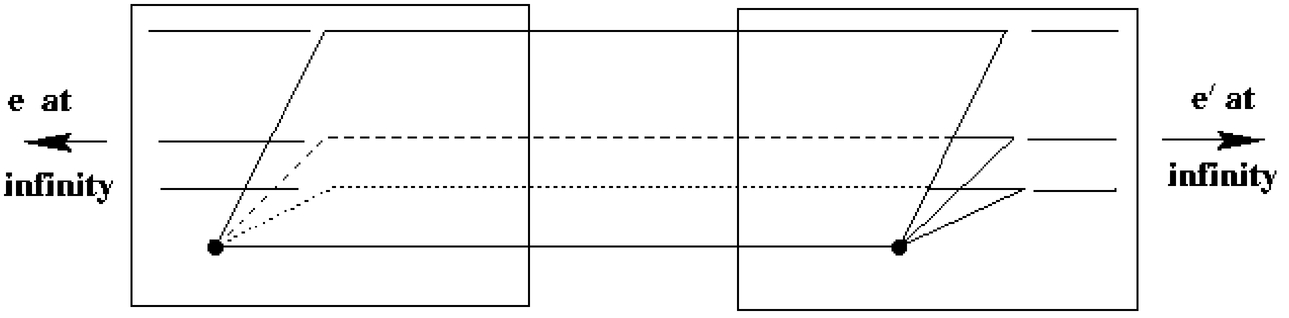
The epipolar pencil

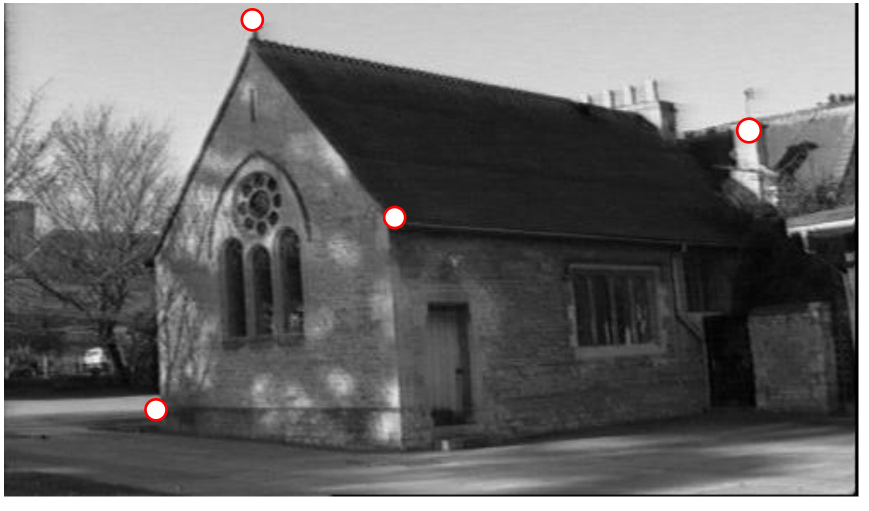


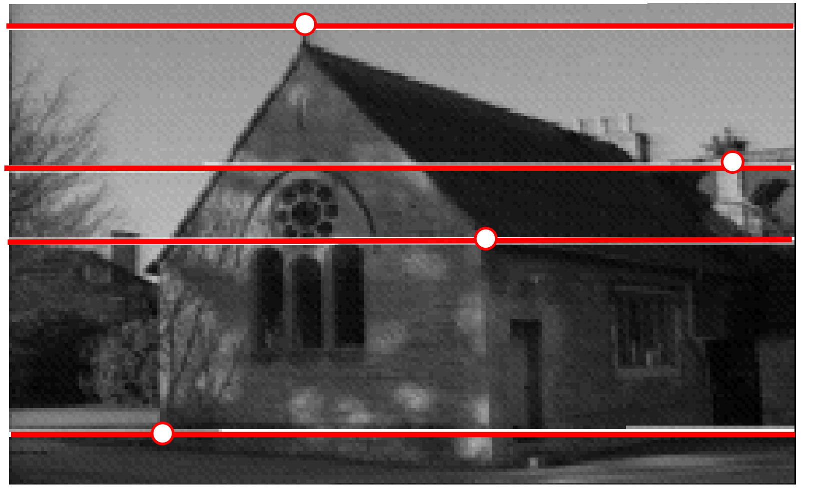
As the position of the 3D point **X** varies, the epipolar planes “rotate” about the baseline. This family of planes is known as an epipolar pencil. All epipolar lines intersect at the epipole.

(a pencil is a one parameter family)

Epipolar geometry example I: parallel cameras

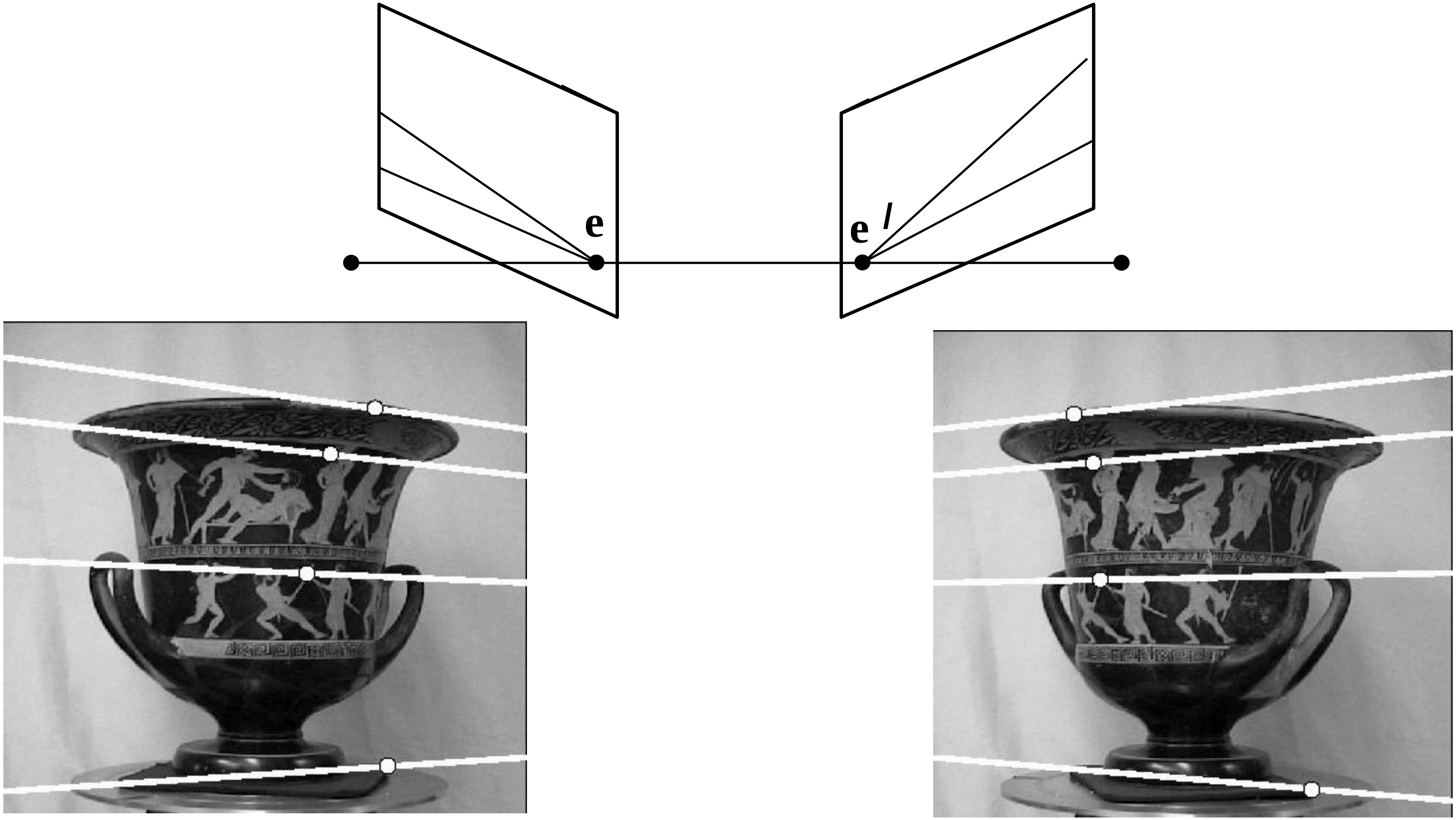






Epipolar geometry depends only on the relative pose (position and orientation) and internal parameters of the two cameras, i.e. the position of the camera centres and image planes. It does not depend on the scene structure (3D points external to the camera).

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 $l_1 x + l_2 y + l_3 = 0$.

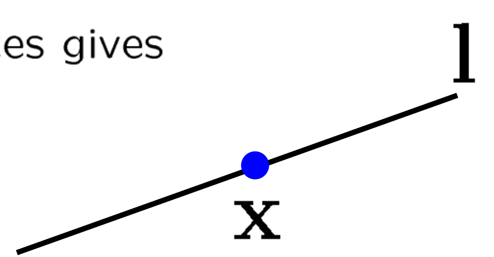
Example represent the line $y = 1$ as a homogeneous vector.

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 $\mu(l_1 x + l_2 y + l_3) = 0$ represents the same line (only the ratio of the homogeneous line coordinates is significant).

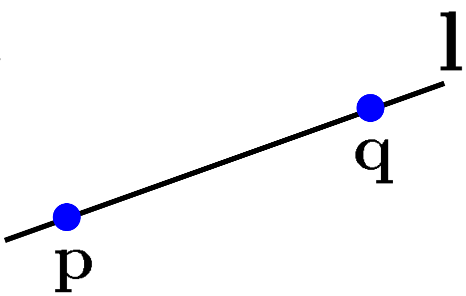
Writing both the point and line in homogeneous coordinates gives

$$l_1 x_1 + l_2 x_2 + l_3 x_3 = 0$$



- point on line $\mathbf{l}.\mathbf{x} = 0$ 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

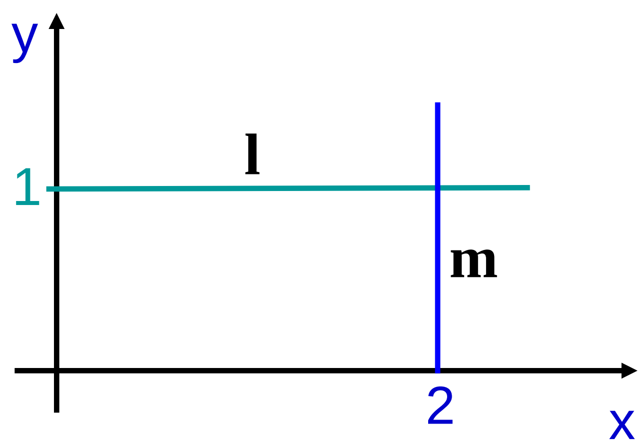
$$\mathbf{l}.\mathbf{p} = (\mathbf{p} \times \mathbf{q}).\mathbf{p} = 0 \qquad \mathbf{l}.\mathbf{q} = (\mathbf{p} \times \mathbf{q}).\mathbf{q} = 0$$

- 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

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

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



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

$$\mathbf{v} \times \mathbf{x} = \begin{pmatrix} v_2 x_3 - v_3 x_2 \\ v_3 x_1 - v_1 x_3 \\ v_1 x_2 - v_2 x_1 \end{pmatrix} = [\mathbf{v}]_\times \mathbf{x}$$

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 $\mathbf{v} \times \mathbf{v} = [\mathbf{v}]_\times \mathbf{v} = \mathbf{0}$.

Example: compute the cross product of **l** and **m**

$$\mathbf{l} = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} \qquad \mathbf{m} = \begin{pmatrix} -1 \\ 0 \\ 2 \end{pmatrix} \qquad [\mathbf{v}]_\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}$$

$$\mathbf{x} = \mathbf{l} \times \mathbf{m} = [\mathbf{l}]_\times \mathbf{m} = \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{pmatrix} -1 \\ 0 \\ 2 \end{pmatrix} = \begin{pmatrix} -2 \\ -1 \\ -1 \end{pmatrix}$$

Note

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

Algebraic representation of epipolar geometry

We know that the epipolar geometry defines a mapping

$$\mathbf{x} \quad \rightarrow \quad \mathbf{l}'$$

point in first image

epipolar line in second image

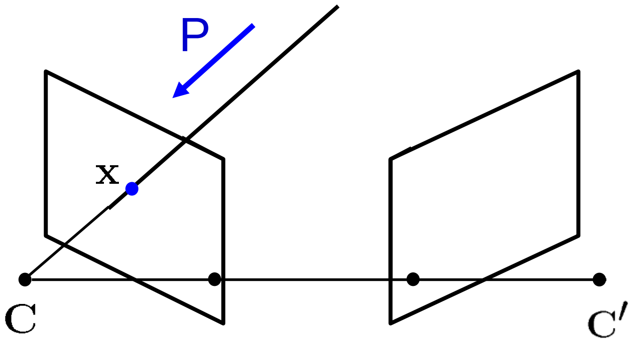
- the map ony depends on the cameras P, P' (not on structure)

- it will be shown that the map is linear and can be written as $\mathbf{l}' = \mathrm{F}\mathbf{x}$, where F is a 3×3 matrix called the fundamental matrix

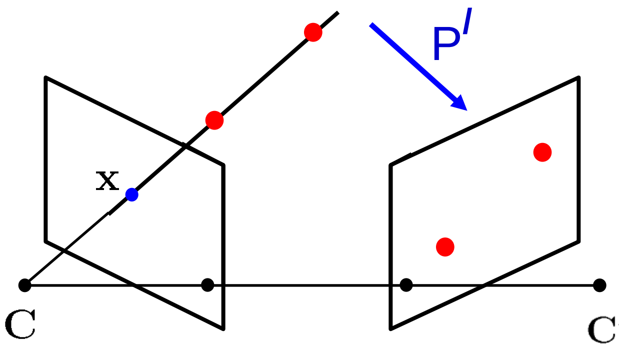
Derivation of the algebraic expression $\mathbf{l}' = \mathtt{F}\mathbf{x}$

Outline

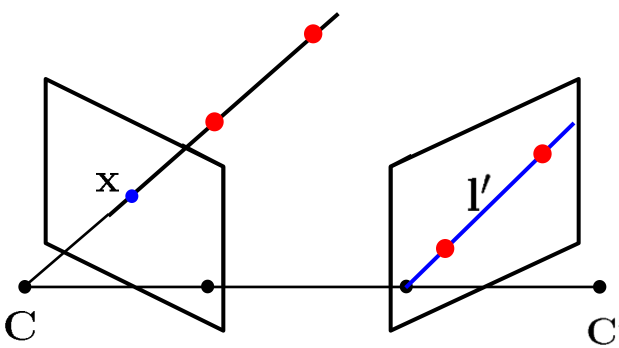
Step 1: for a point x in the first image back project a ray with camera P



Step 2: choose two points on the ray and project into the second image with camera P′



Step 3: compute the line through the two image points using the relation $\mathbf{l}' = \mathbf{p} \times \mathbf{q}$



- choose camera matrices

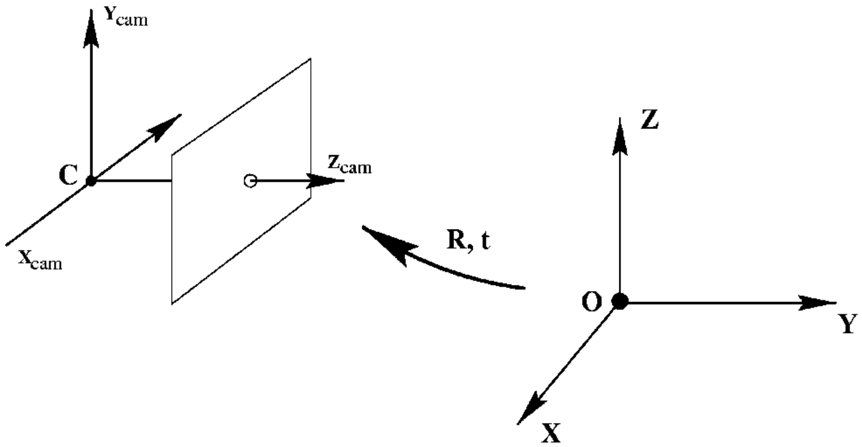
$$P = K [R \mid t]$$

internal calibration

rotation

translation

from world to camera coordinate frame



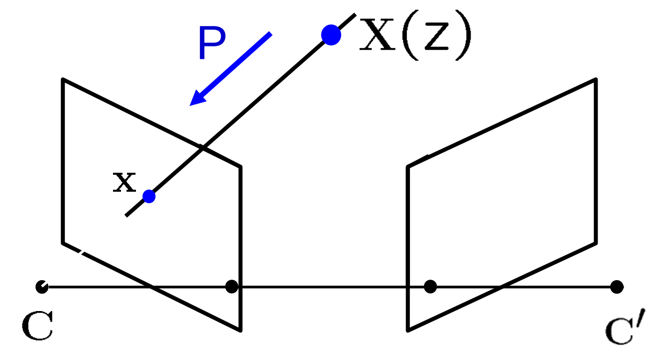
- first camera $P = K [I \mid 0]$

world coordinate frame aligned with first camera

(i.e., first camera defines a reference space)

- second camera $P' = K' [R \mid t]$

Step 1: for a point x in the first image
back project a ray with camera $\mathrm{P} = \mathrm{K}\,[\,\mathrm{I} \mid \mathbf{0}\,]$



A point $\mathbf{x}$ back projects to a ray

$$\begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \end{pmatrix} = \mathrm{z}\mathrm{K}^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathrm{z}\mathrm{K}^{-1}\mathbf{x}$$

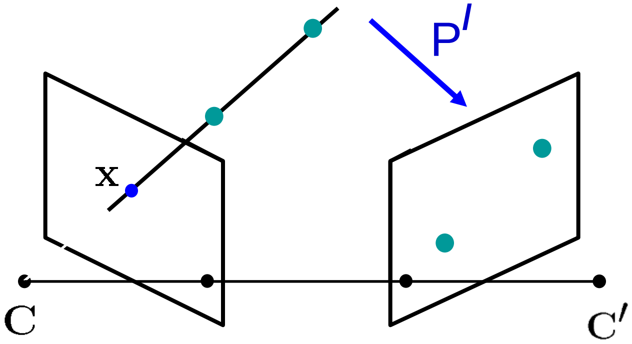
where $\mathbf{Z}$ is the point's depth, since

$$\mathrm{X}(\mathrm{z}) = \begin{pmatrix} \mathrm{z}\mathrm{K}^{-1}\mathbf{x} \\ 1 \end{pmatrix}$$

satisfies

$$\mathrm{P}\mathrm{X}(\mathrm{z}) = \mathrm{K}[\mathrm{I} \mid \mathbf{0}]\mathrm{X}(\mathrm{z}) = \mathbf{x}$$

Step 2: choose two points on the ray and project into the second image with camera $P^{/}$



Consider two points on the ray $\mathrm{X}(z) = \begin{pmatrix} z\mathrm{K}^{-1}\mathbf{x} \\ 1 \end{pmatrix}$

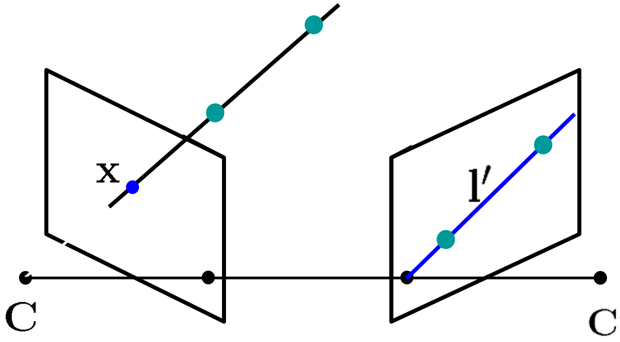
- $\mathbf{Z}$ = 0 is the camera centre $\begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}$

- $\mathbf{Z} = \infty$ is the point at infinity $\begin{pmatrix} \mathrm{K}^{-1}\mathbf{x} \\ 0 \end{pmatrix}$

Project these two points into the second view

$$\mathrm{P}' \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} = \mathrm{K}'[\mathrm{R} \mid \mathbf{t}] \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} = \mathrm{K}'\mathbf{t} \qquad \mathrm{P}' \begin{pmatrix} \mathrm{K}^{-1}\mathbf{x} \\ 0 \end{pmatrix} = \mathrm{K}'[\mathrm{R} \mid \mathbf{t}] \begin{pmatrix} \mathrm{K}^{-1}\mathbf{x} \\ 0 \end{pmatrix} = \mathrm{K}'\mathrm{R}\mathrm{K}^{-1}\mathbf{x}$$

Step 3: compute the line through the two image points using the relation $\mathbf{l}' = \mathbf{p} \times \mathbf{q}$



Compute the line through the points $\mathbf{l}' = (\mathtt{K}'\mathbf{t}) \times (\mathtt{K}'\mathtt{R}\mathtt{K}^{-1}\mathbf{x})$

Using the identity $(\mathtt{M}\mathbf{a}) \times (\mathtt{M}\mathbf{b}) = \mathtt{M}^{-\top}(\mathbf{a} \times \mathbf{b})$ where $\mathtt{M}^{-\top} = (\mathtt{M}^{-1})^{\top} = (\mathtt{M}^{\top})^{-1}$

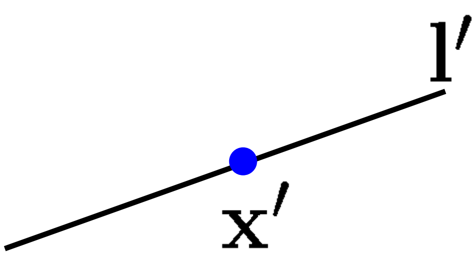
$$\mathbf{l}' = \mathtt{K}'^{-\top}\left(\mathbf{t} \times (\mathtt{R}\mathtt{K}^{-1}\mathbf{x})\right) = \underbrace{\mathtt{K}'^{-\top}[\mathbf{t}]_{\times}\mathtt{R}\mathtt{K}^{-1}}_{\mathrm{F}}\mathbf{x}$$

F is the fundamental matrix

$$\boxed{\mathbf{l}' = \mathtt{F}\mathbf{x} \qquad \mathtt{F} = \mathtt{K}'^{-\top}[\mathbf{t}]_{\times}\mathtt{R}\mathtt{K}^{-1}}$$

Points $\mathbf{x}$ and $\mathbf{x}'$ correspond $(\mathbf{x} \leftrightarrow \mathbf{x}')$ then $\mathbf{x}'^{\top}\mathbf{l}' = 0$

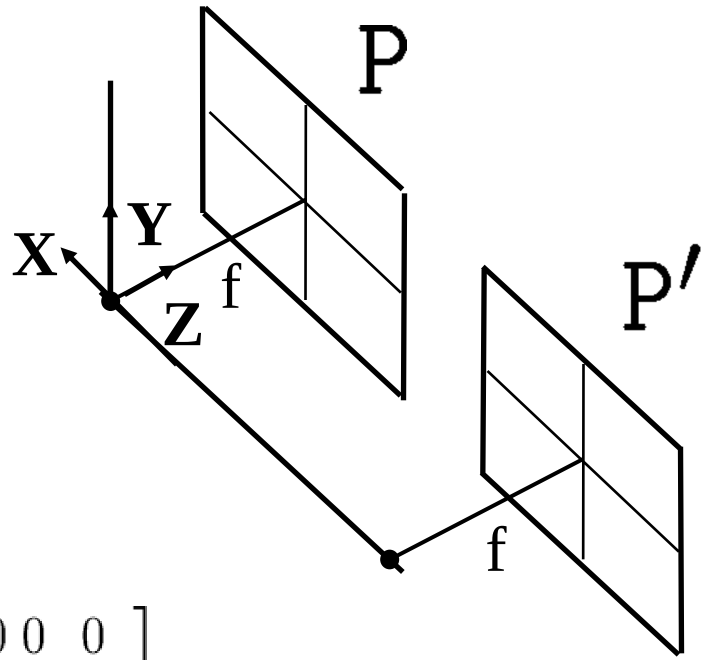
$$\boxed{\mathbf{x}'^{\top}\mathtt{F}\mathbf{x} = 0}$$



Example I: compute the fundamental matrix for a parallel camera stereo rig

$$\mathtt{P} = \mathtt{K}[\mathtt{I} \mid \mathbf{0}] \qquad \mathtt{P}' = \mathtt{K}'[\mathtt{R} \mid \mathbf{t}]$$

$$\mathtt{K} = \mathtt{K}' = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathtt{R} = \mathtt{I} \quad \mathbf{t} = \begin{pmatrix} t_x \\ 0 \\ 0 \end{pmatrix}$$



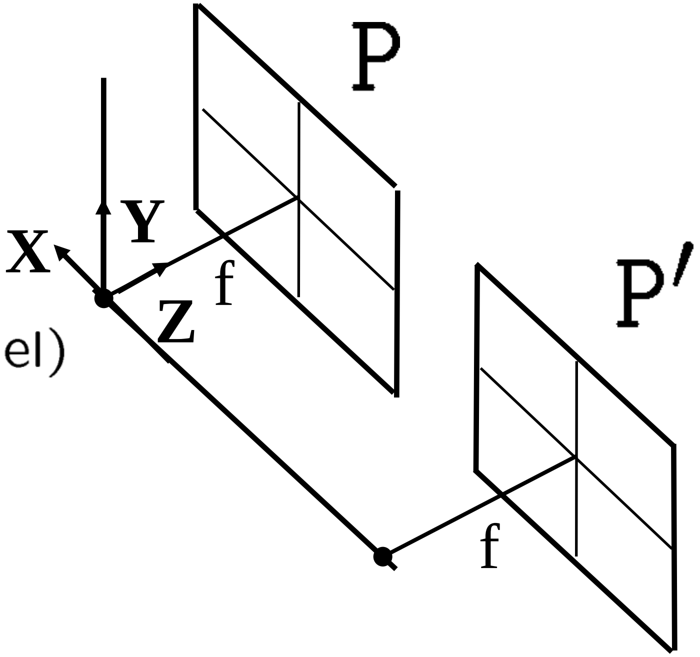
$$\mathtt{F} = \mathtt{K}'^{-\top}[\mathbf{t}]_\times \mathtt{R}\mathtt{K}^{-1}$$

$$= \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t_x \\ 0 & t_x & 0 \end{bmatrix} \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{x}'^\top \mathbf{F} \mathbf{x} = \begin{pmatrix} x' & y' & 1 \end{pmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$

- reduces to $y = y'$, i.e. raster correspondence (horizontal scan-lines)

$\mathtt{F}$ is a rank 2 matrix



The epipole $\mathbf{e}$ is the null-space vector (kernel) of $\mathtt{F}$ (exercise), i.e. $\mathtt{F}\mathbf{e} = \mathbf{0}$

In this case

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \mathbf{0}$$

so that

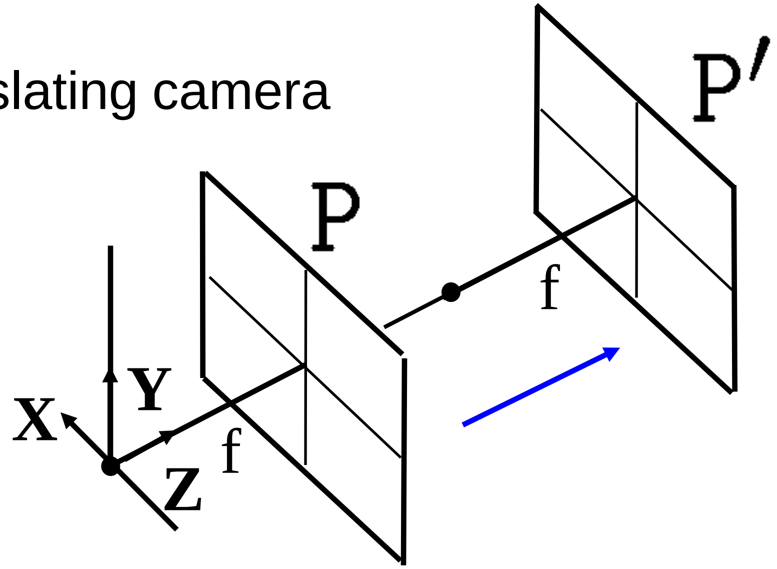
$$\mathbf{e} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Geometric interpretation ?

Example II: compute F for a forward translating camera

$$\mathtt{P} = \mathtt{K}[\mathtt{I} \mid \mathbf{0}] \qquad \mathtt{P}' = \mathtt{K}'[\mathtt{R} \mid \mathbf{t}]$$

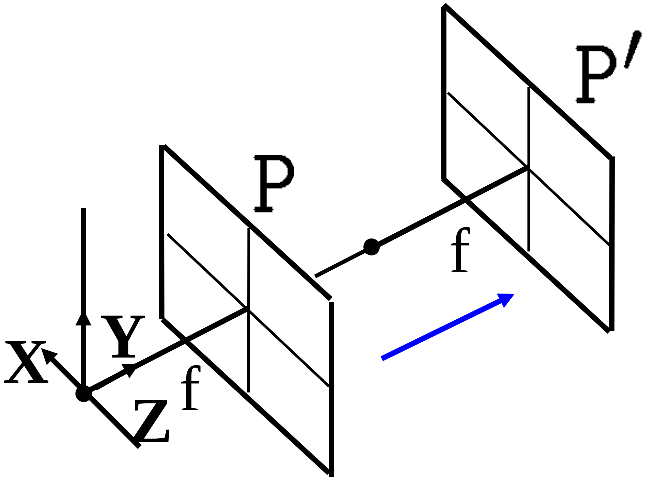
$$\mathtt{K} = \mathtt{K}' = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathtt{R} = \mathtt{I} \quad \mathbf{t} = \begin{pmatrix} 0 \\ 0 \\ t_z \end{pmatrix}$$



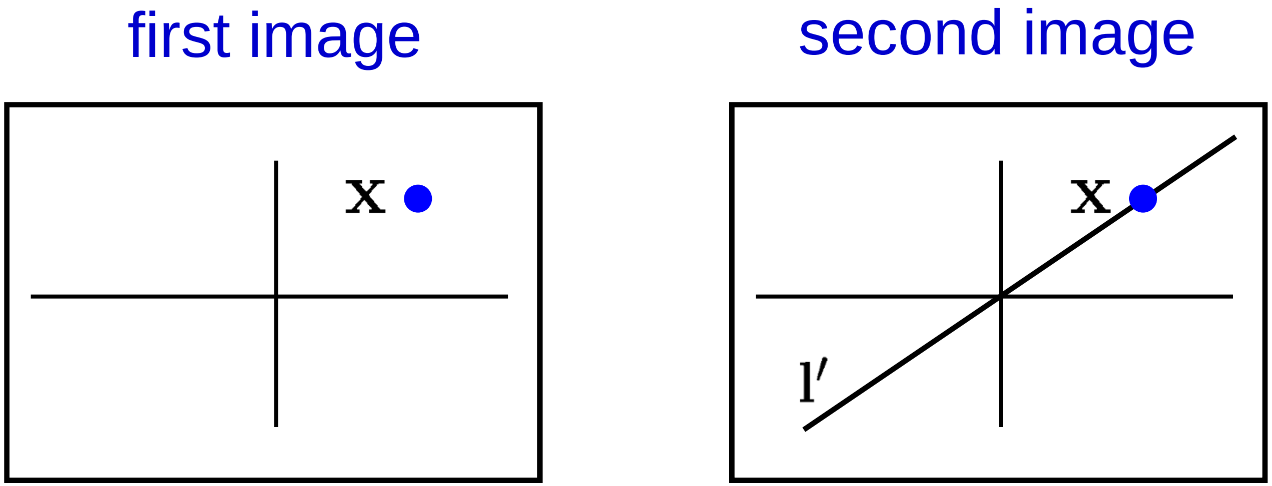
$$\begin{aligned} \mathtt{F} &= \mathtt{K}'^{-\top}[\mathbf{t}]_\times \mathtt{R}\mathtt{K}^{-1} \\ &= \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -t_z & 0 \\ t_z & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

From $\mathbf{l}' = \mathtt{F}\mathbf{x}$ the epipolar line for the point $\mathbf{x} = (x, y, 1)^\top$ is

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

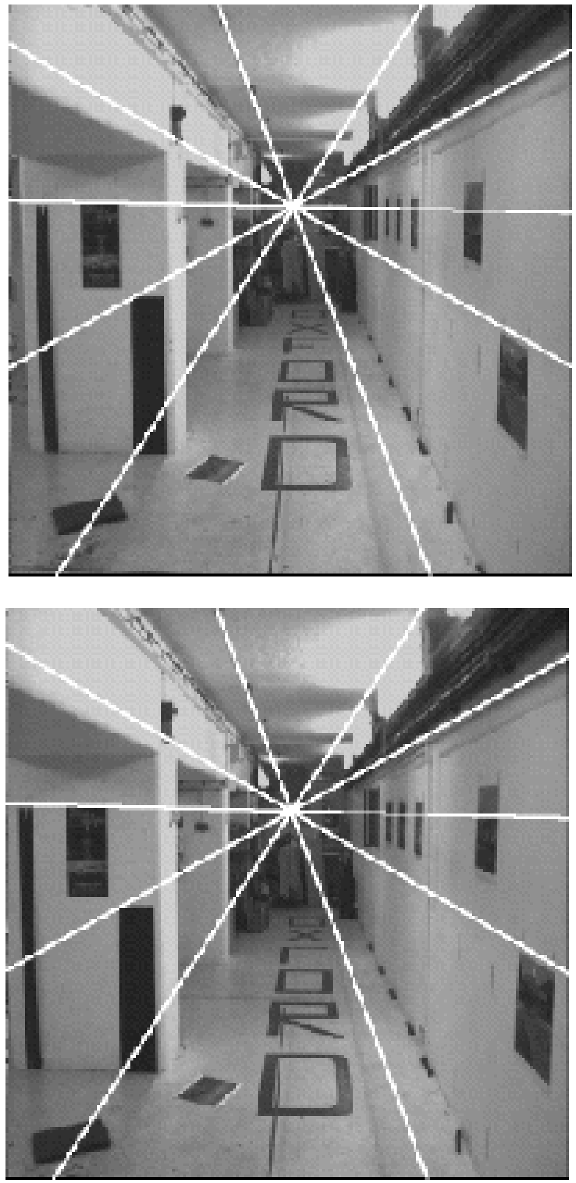


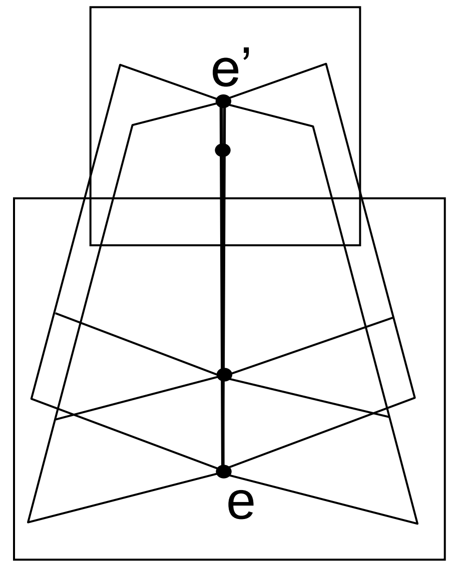
The points $(x, y, 1)^\top$ and $(0, 0, 1)^\top$ lie on this line




OXFORD


OXFORD




e’
e

Summary: Properties of the Fundamental matrix

- $\mathtt{F}$ 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:

  ◇ $\mathtt{F}\mathbf{e} = \mathbf{0}$.

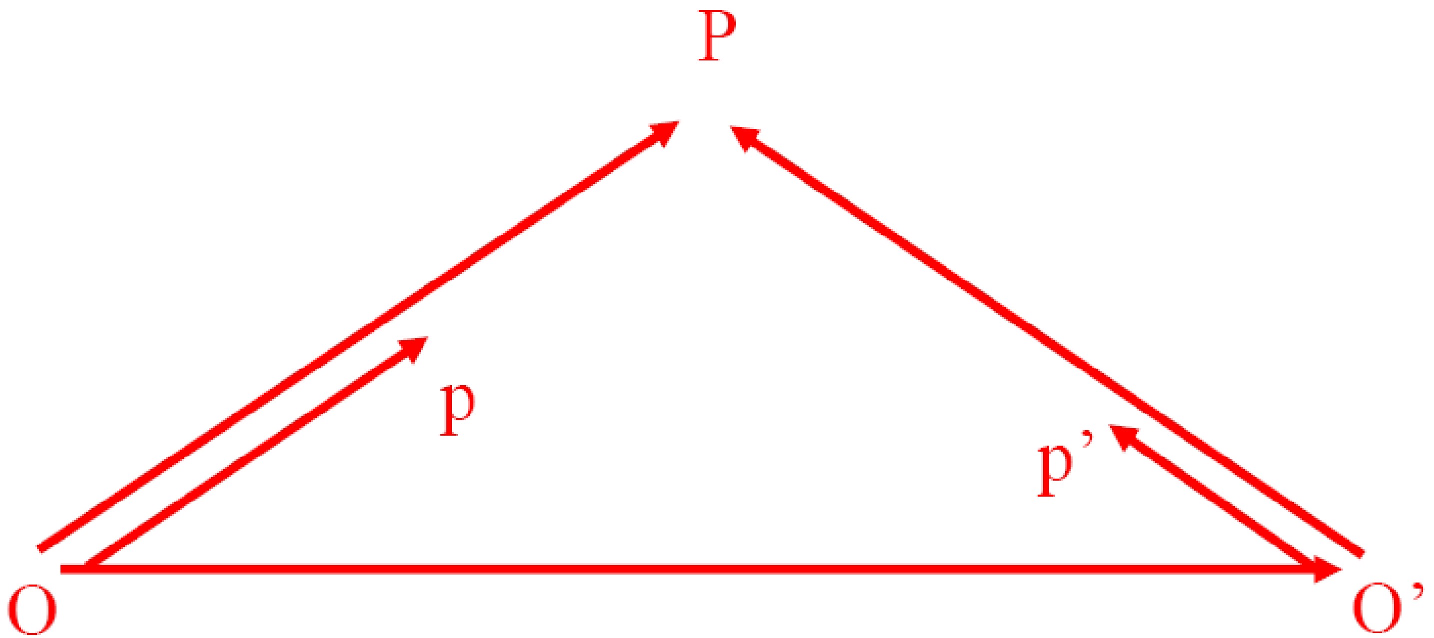
  ◇ $\mathtt{F}^{\top}\mathbf{e}' = \mathbf{0}$.

- Computation from camera matrices $\mathtt{P}, \mathtt{P}'$:

  $\mathtt{P} = \mathtt{K}[\mathtt{I} \mid \mathbf{0}]$, $\mathtt{P}' = \mathtt{K}'[\mathtt{R} \mid \mathbf{t}]$, $\mathtt{F} = \mathtt{K}'^{-\top}[\mathbf{t}]_{\times}\mathtt{R}\mathtt{K}^{-1}$

The Essential Matrix (F&P chapter 6)

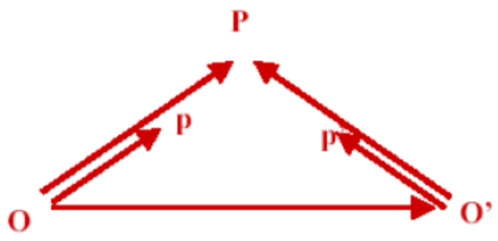
- Algebraic setup:



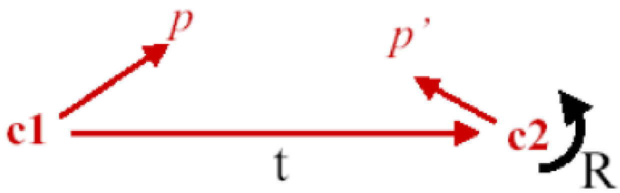
The epipolar constraint: these vectors are coplanar:

$$\overrightarrow{Op} \cdot [\overrightarrow{OO'} \times \overrightarrow{O'p'}] = 0$$

The Essential Matrix: Equation



$$\overrightarrow{Op} \cdot [\overrightarrow{OO'} \times \overrightarrow{O'p'}] = 0$$



p,p' are image coordinates of P in c1 and c2…

c2 is related to c1 by rotation R and translation t

$$\boldsymbol{p} \cdot [\boldsymbol{t} \times (\mathcal{R}\boldsymbol{p}')] = 0$$

Linear Constraint:
Should be able to express as matrix multiplication.

The Essential Matrix: Final Form

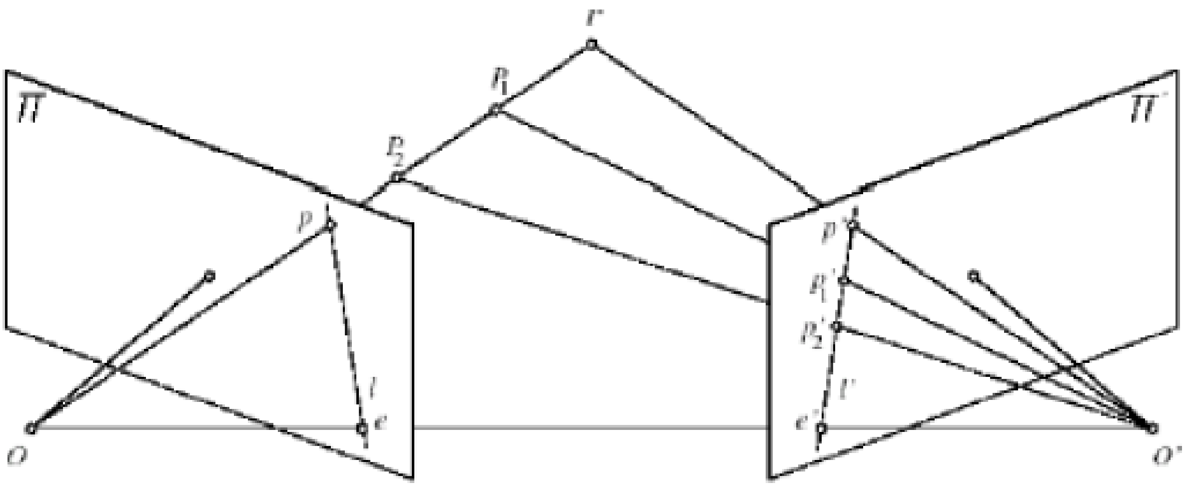
- Relates image of one point in one camera to the other, given rotation and translation

$$\boldsymbol{p} \cdot [\boldsymbol{t} \times (\mathcal{R}\boldsymbol{p}')] = 0$$

$$\mathcal{E} = [t_x]\Re$$

$$p^T [t_x] \Re p' = 0$$

$$\boldsymbol{p}^T \mathcal{E} \boldsymbol{p}' = 0$$



Essential (E) vs Fundamental (F) Matrix

- F has intrinsic and extrinsic parameters, E only has extrinsic
- Must know both camera properties for computing E
  - Need calibrations
- No calibration for F
- E maps point in one image to the other
- F maps point to corresponding epipolar lines

An algorithm for stereo reconstruction

1. For each point in the first image determine the corresponding point in the second image

   (this is a search problem)

2. For each pair of matched points determine the 3D point by triangulation

   (this is an estimation problem)

Epipolar line

Epipolar constraint

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

Algebraic representation of epipolar geometry

We know that the epipolar geometry defines a mapping

$$\mathbf{x} \quad \mapsto \quad \mathbf{l}'$$

point in first image

epipolar line in second image

- 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

$\mathbf{l}' = \mathtt{F}\mathbf{x}$ where $\mathtt{F}$ is a 3×3 matrix called the fundamental matrix

Stereo correspondence algorithms

Problem statement

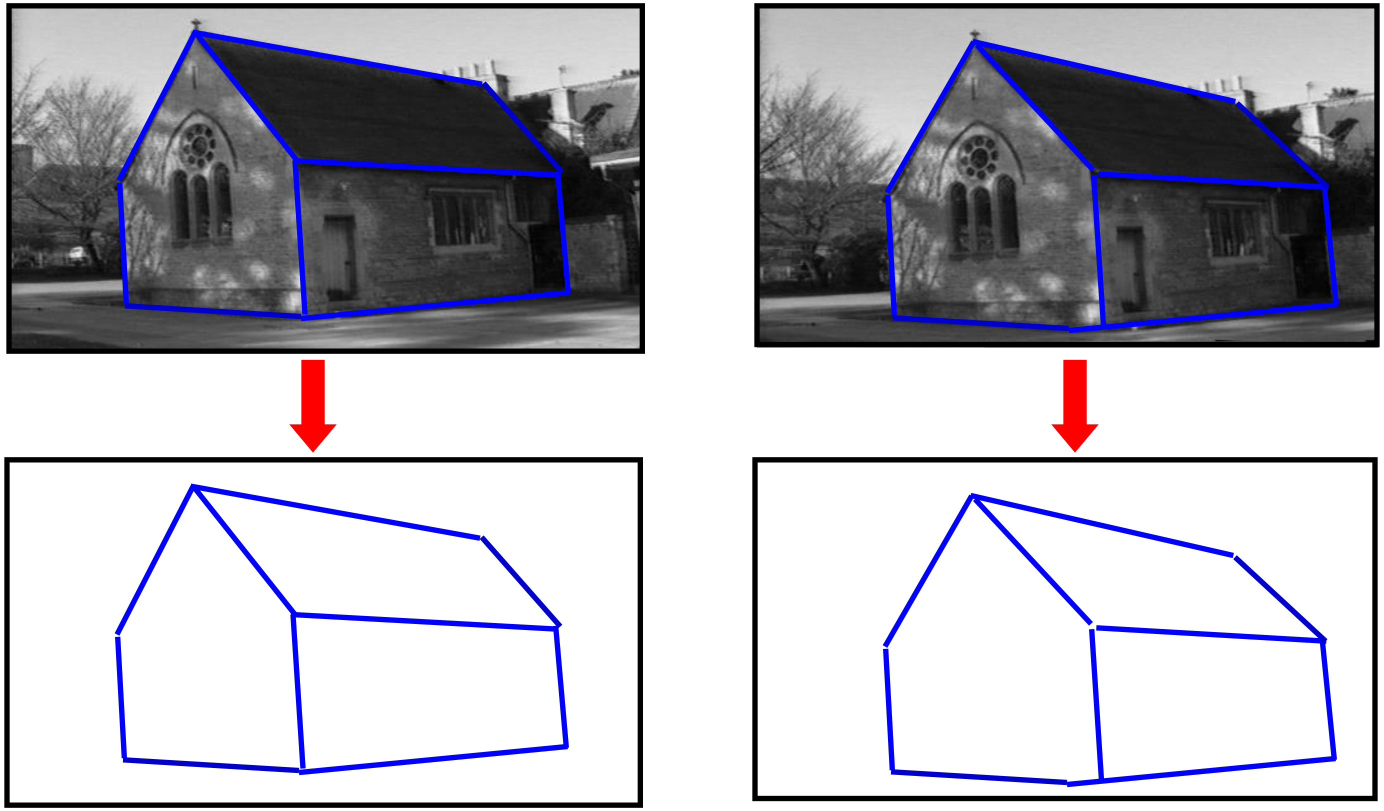
Given: two images and their associated cameras compute corresponding image points.

Algorithms may be classified into two types:

1. Dense: compute a correspondence at every pixel
2. Sparse: compute correspondences only for features

The methods may be top down or bottom up

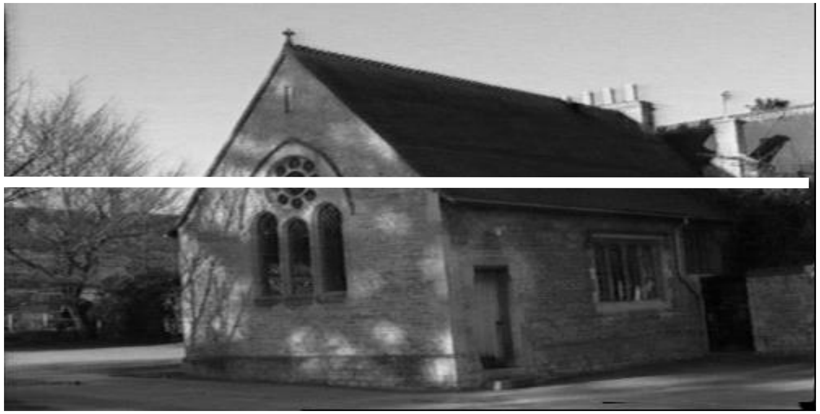
Top down matching



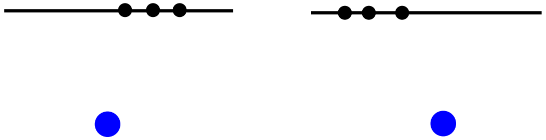
1. Group model (house, windows, etc) independently in each image
2. Match points (vertices) between images

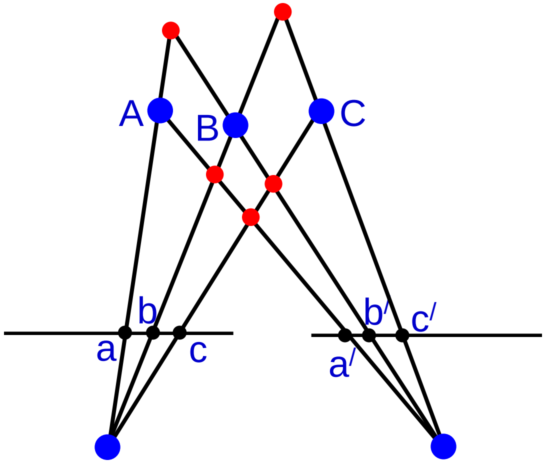
Bottom up matching

- epipolar geometry reduces the correspondence search from 2D to a 1D search on corresponding epipolar lines



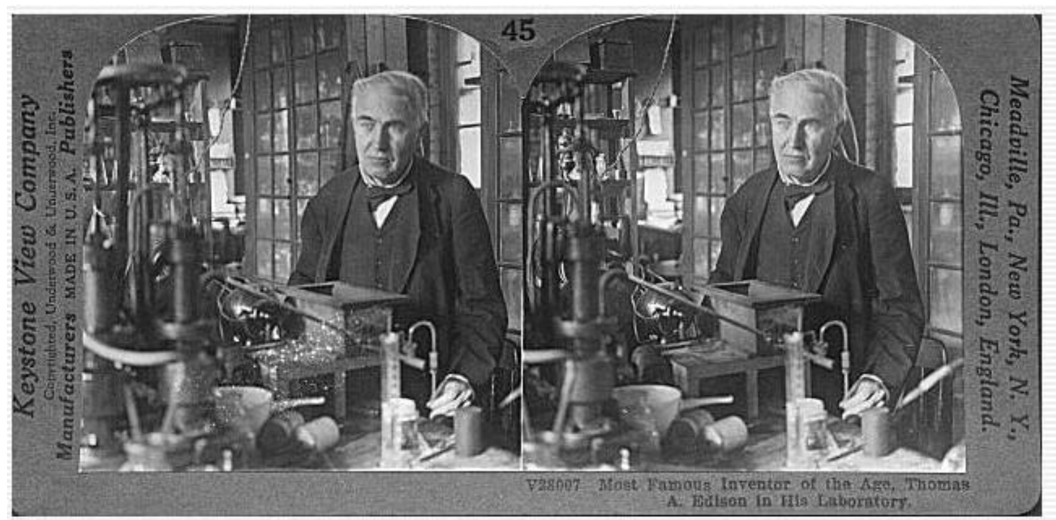
- 1D correspondence problem



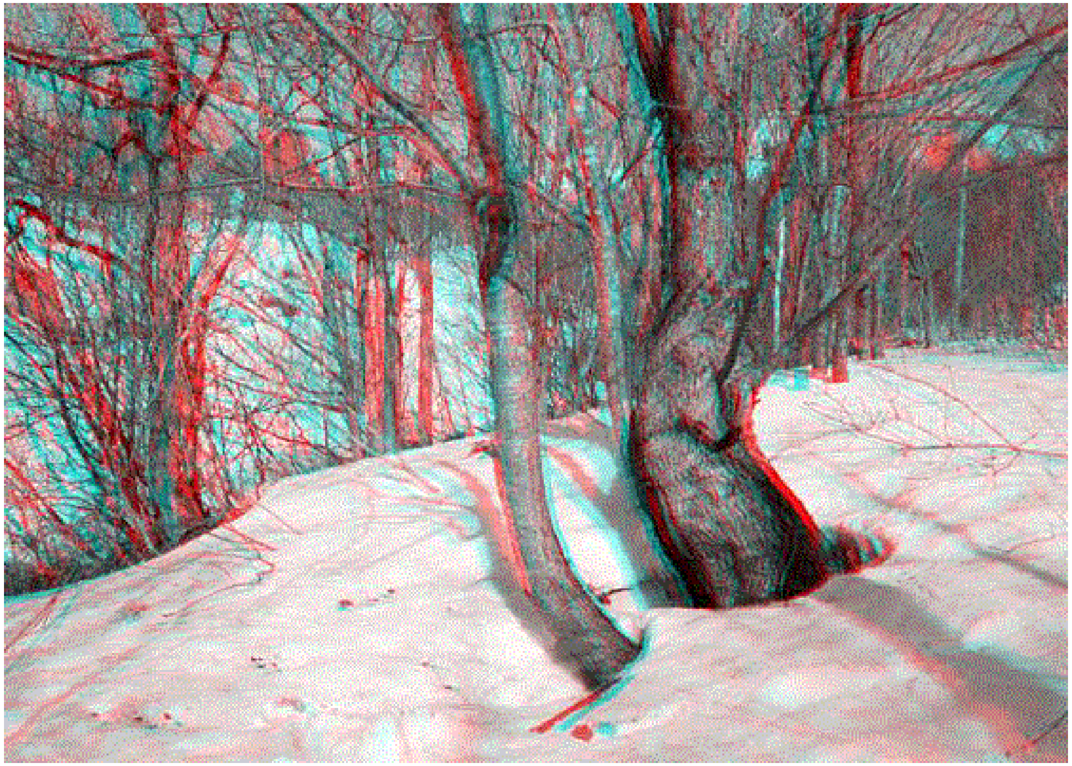


Stereograms

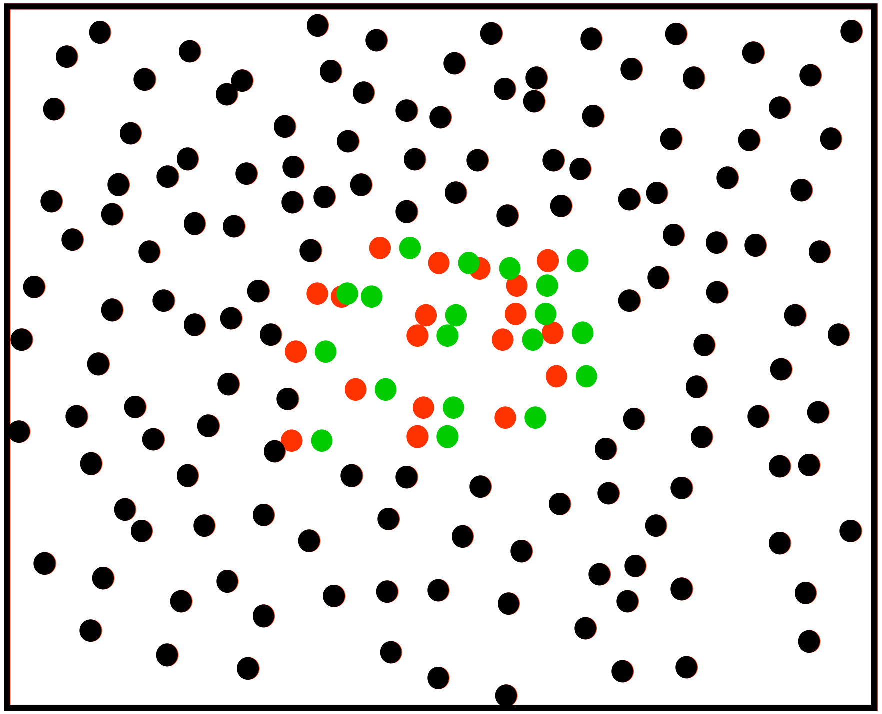
- Invented by Sir Charles Wheatstone, 1838

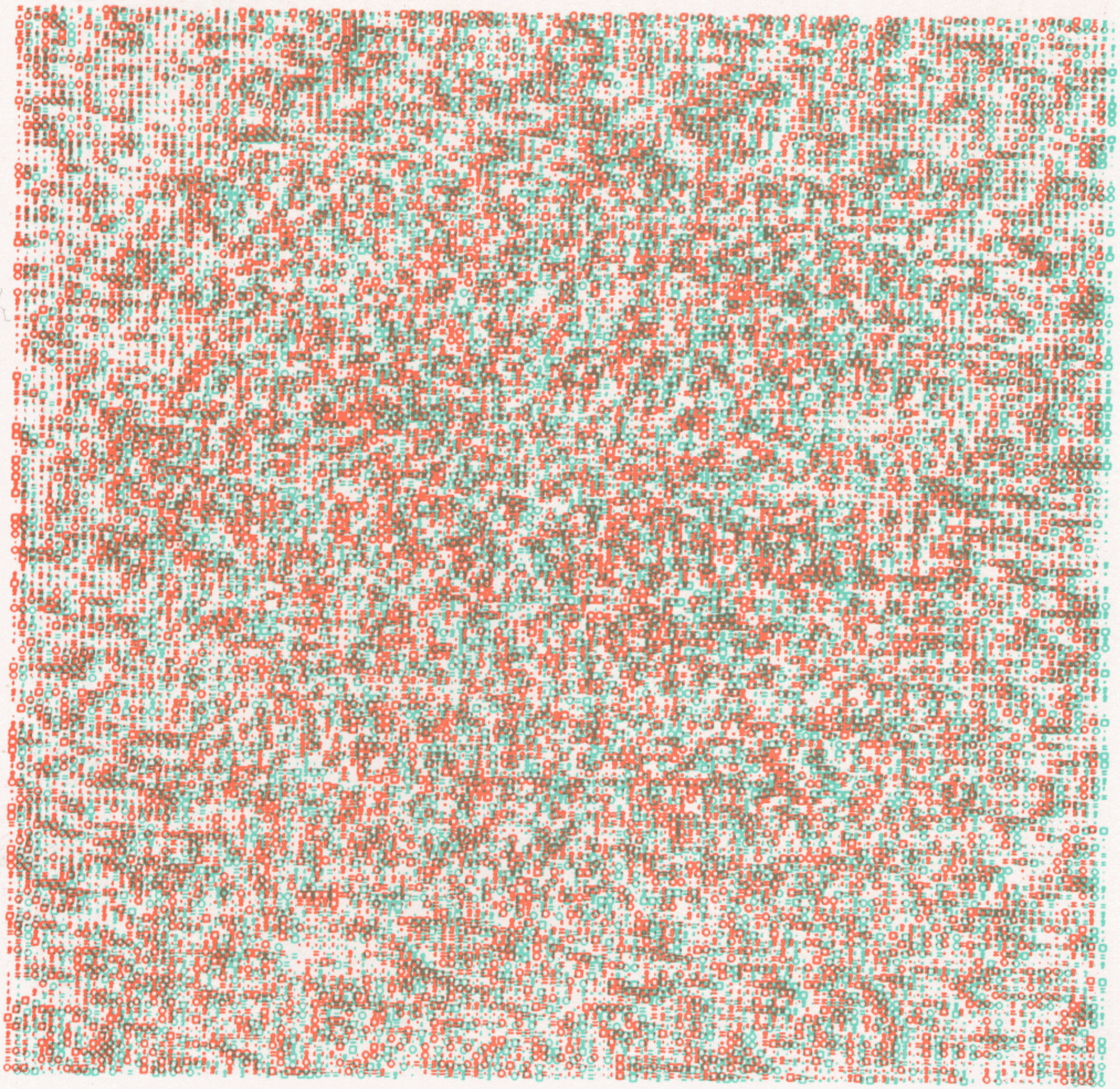


Red/green stereograms



Random dot stereograms





Autostereograms

Autostereograms: www.magiceye.com

Correspondence algorithms

Algorithms may be top down or bottom up – random dot stereograms are an existence proof that bottom up algorithms are possible

From here on only consider bottom up algorithms

Algorithms may be classified into two types:

1. Dense: compute a correspondence at every pixel
2. Sparse: compute correspondences only for features

Example image pair – parallel cameras

First image

Second image

Dense correspondence algorithm

Parallel camera example – epipolar lines are corresponding rasters

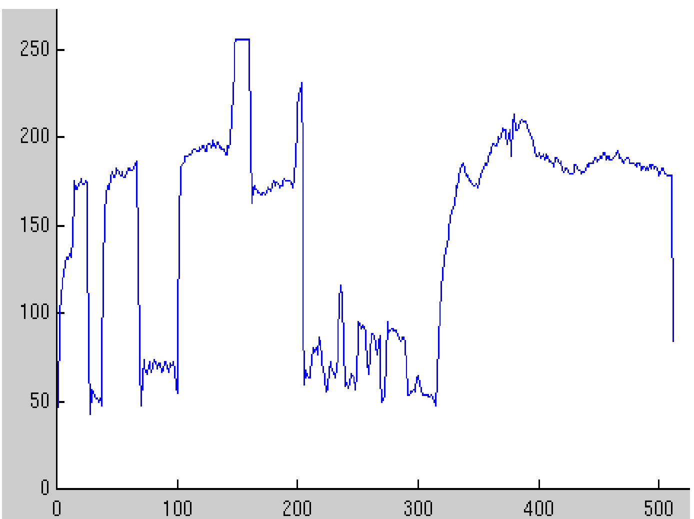
epipolar line

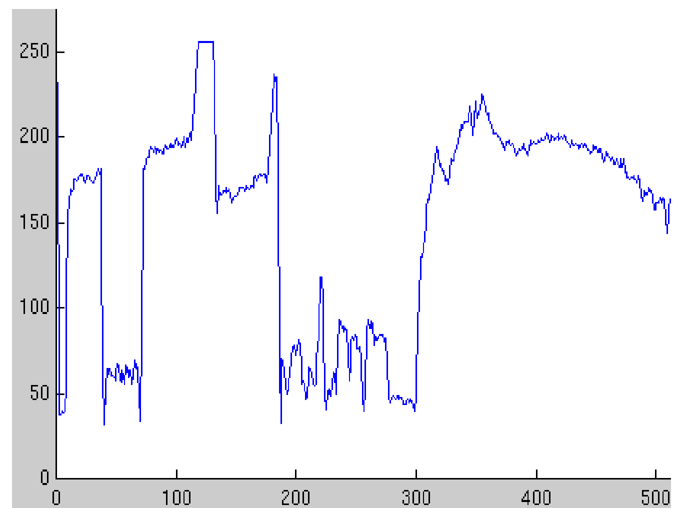
Search problem (geometric constraint): for each point in the left image, the corresponding point in the right image lies on the epipolar line (1D ambiguity)

Disambiguating assumption (photometric constraint): the intensity neighbourhood of corresponding points are similar across images

Measure similarity of neighbourhood intensity by cross-correlation

Intensity profiles





- Clear correspondence between intensities, but also noise and ambiguity

Normalized Cross Correlation

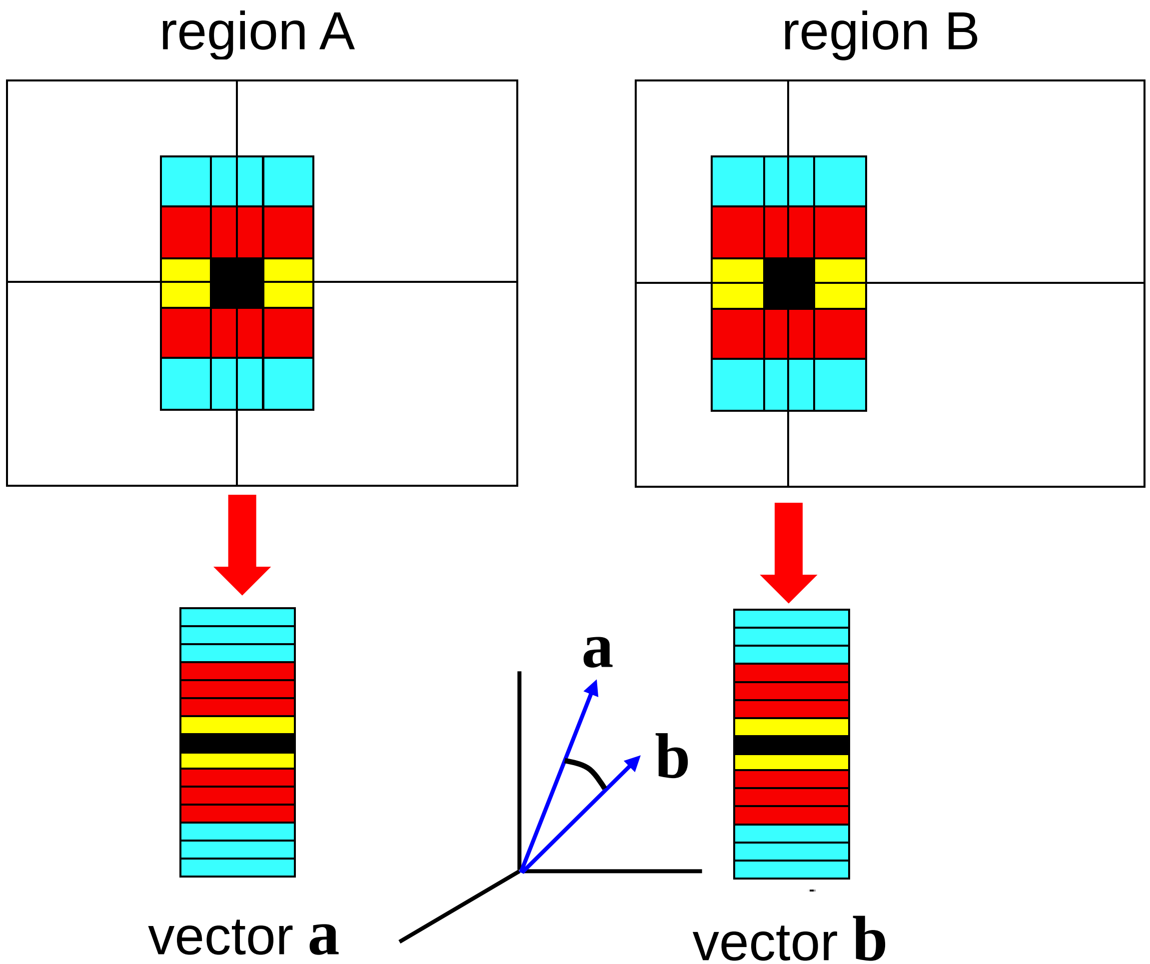
$$\mathrm{NCC} = \frac{\sum_i \sum_j A(i,j)B(i,j)}{\sqrt{\sum_i \sum_j A(i,j)^2}\sqrt{\sum_i \sum_j B(i,j)^2}}$$

write regions as vectors

$$\mathtt{A} \rightarrow \mathbf{a}, \quad \mathtt{B} \rightarrow \mathbf{b}$$

$$\mathrm{NCC} = \frac{\mathbf{a}.\mathbf{b}}{|\mathbf{a}||\mathbf{b}|}$$

$$-1 \leq \mathrm{NCC} \leq 1$$



Cross-correlation of neighbourhood regions

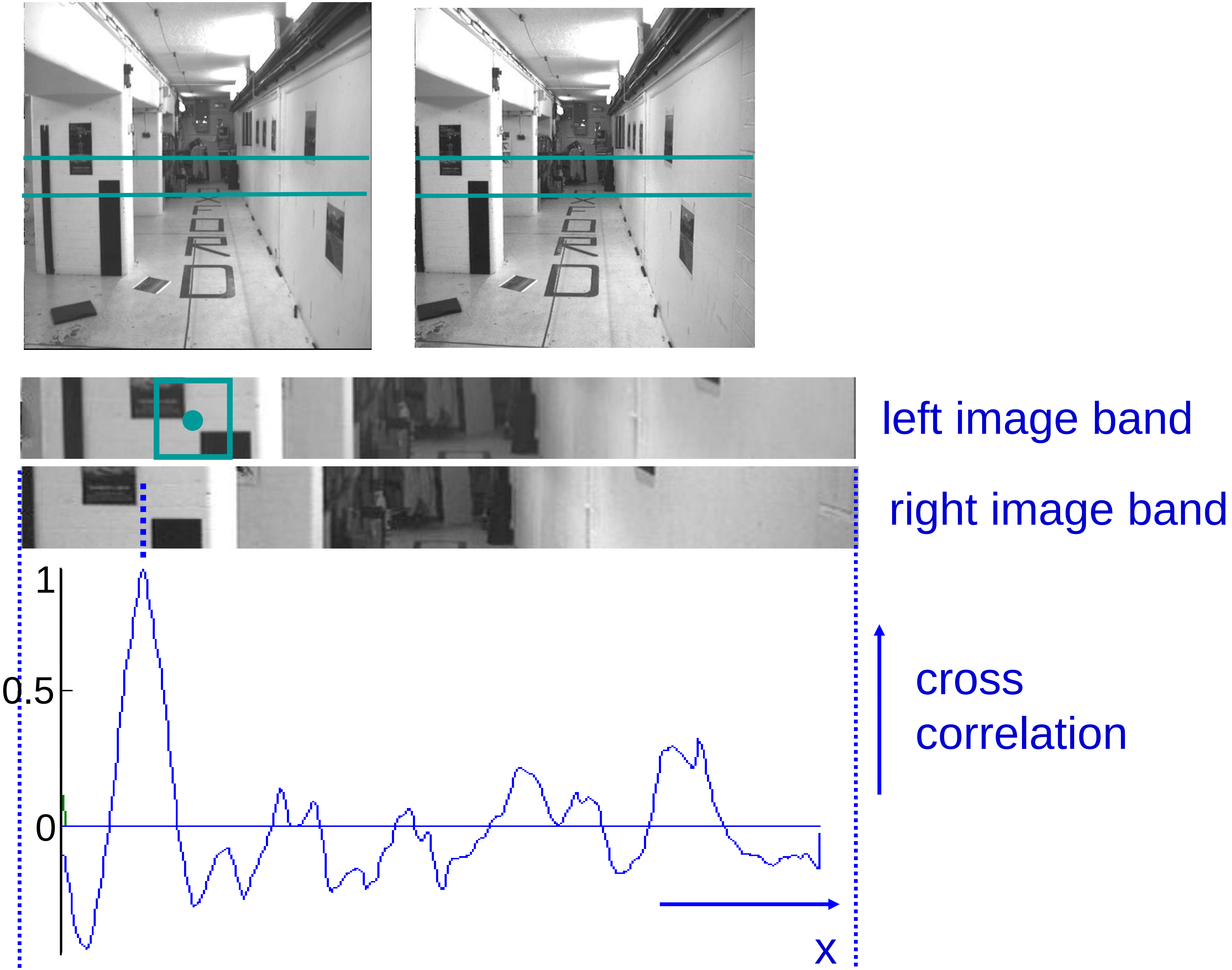
regions $\mathtt{A}, \mathtt{B}$, write as vectors $\mathbf{a}, \mathbf{b}$

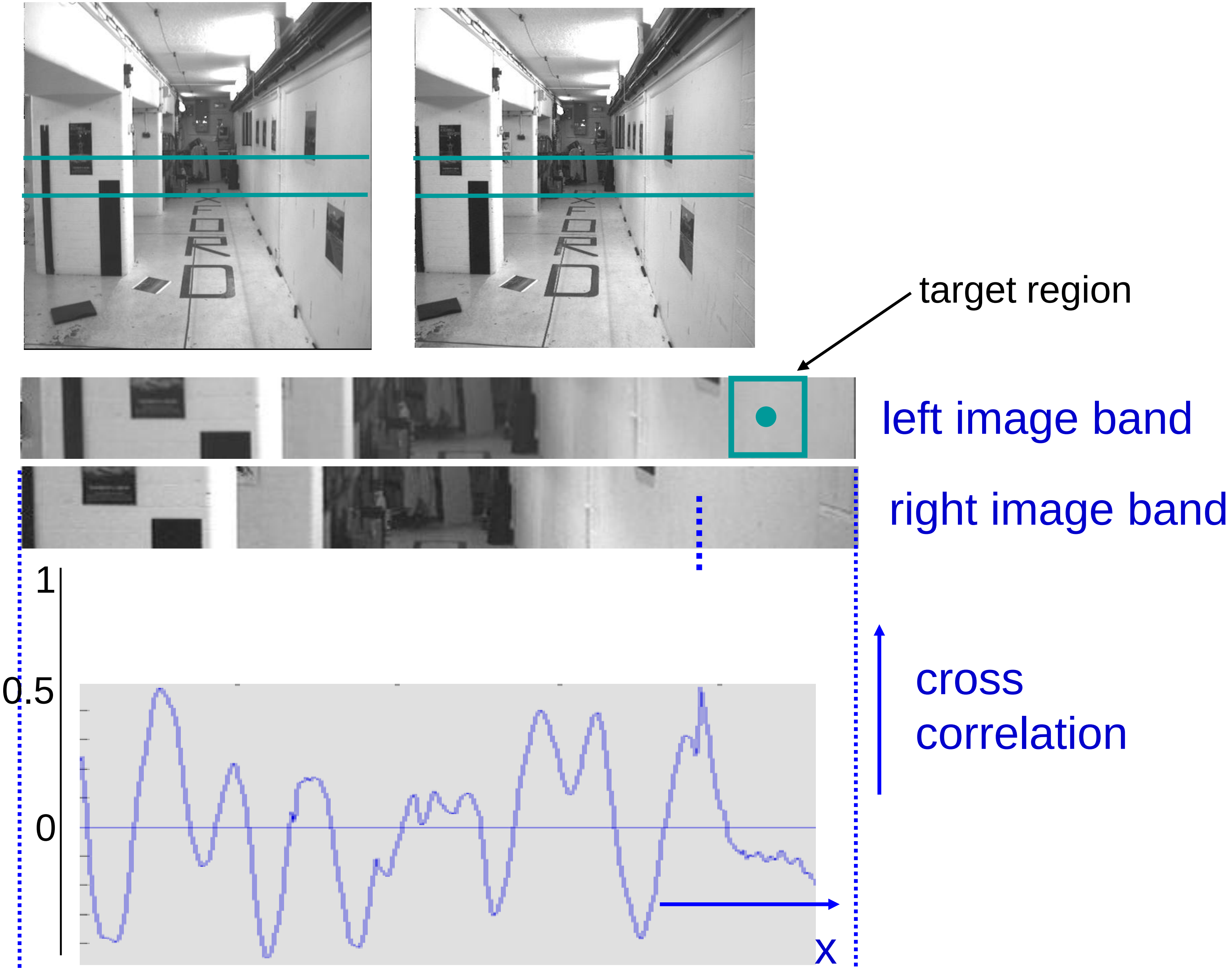
translate so that mean is zero

$$\mathtt{a} \rightarrow \mathtt{a} - \langle \mathbf{a} \rangle, \quad \mathtt{b} \rightarrow \mathtt{b} - \langle \mathbf{b} \rangle$$

$$\text{cross correlation} = \frac{\mathbf{a}.\mathbf{b}}{|\mathbf{a}||\mathbf{b}|}$$

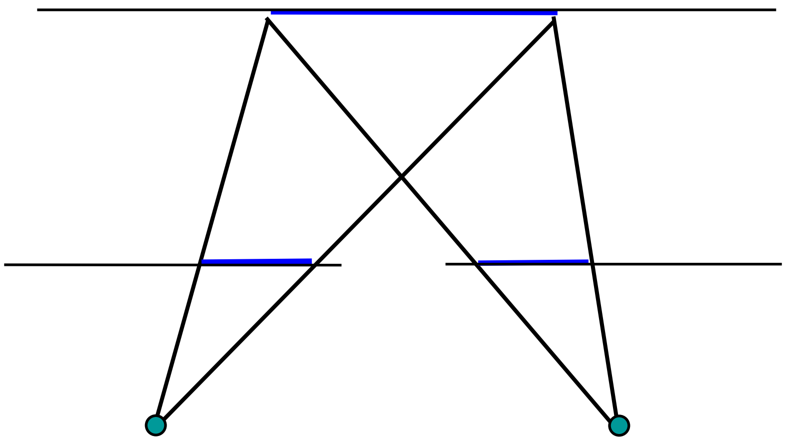
Invariant to $I \rightarrow \alpha I + \beta$


left image band
right image band
1
0.5
0
cross
correlation
x


target region
left image band
right image band
1
0.5
0
cross
correlation
x

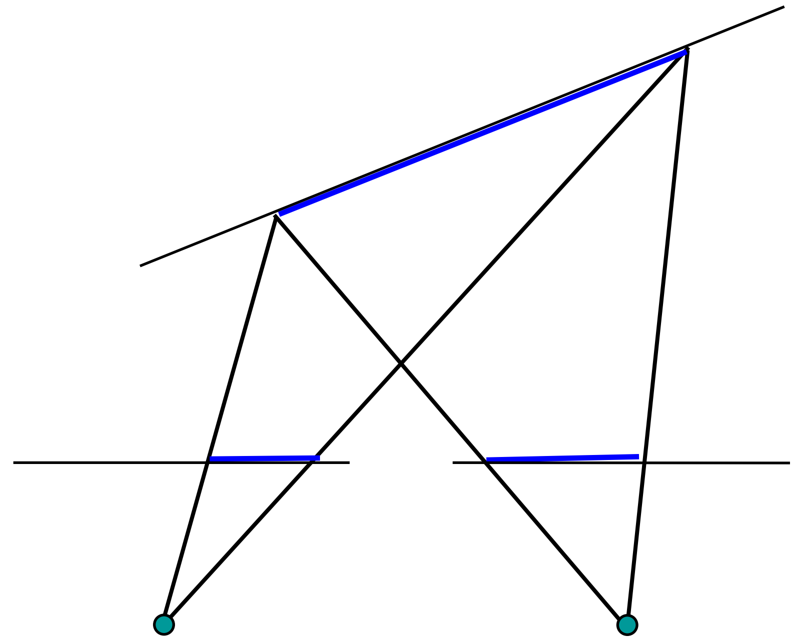
Why is cross-correlation such a poor measure in the second case?

1. The neighbourhood region does not have a “distinctive” spatial intensity distribution
2. Foreshortening effects



fronto-parallel surface

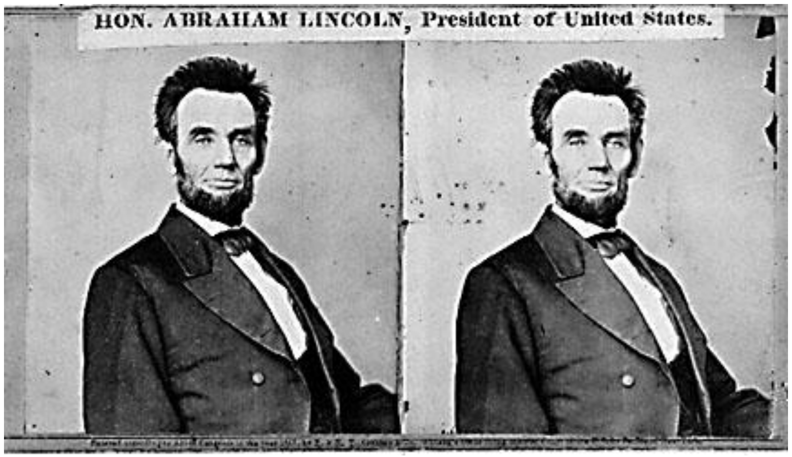
imaged length the same



slanting surface

imaged lengths differ

Limitations of similarity constraint



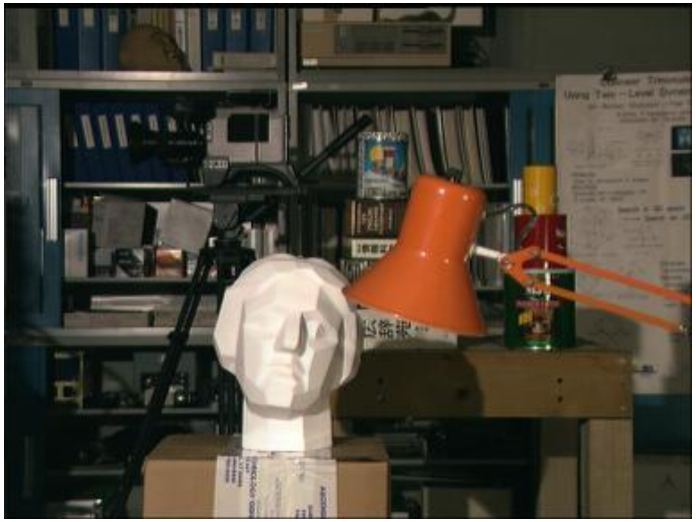
Textureless surfaces

Occlusions, repetition

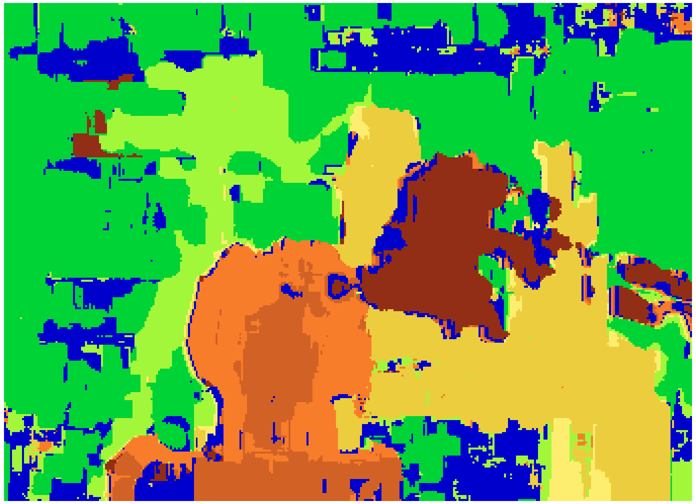
Non-Lambertian surfaces, specularities

Results with window search

Data



Window-based matching



Ground truth

Sketch of a dense correspondence algorithm

For each pixel in the left image

- compute the neighbourhood cross correlation along the corresponding epipolar line in the right image
- the corresponding pixel is the one with the highest cross correlation

Parameters

- size (scale) of neighbourhood
- search disparity

Other constraints

- uniqueness
- ordering
- smoothness of disparity field

Applicability

- textured scene, largely fronto-parallel