

# 9 – Stereo Reconstruction

Slides from A. Zisserman & S. Lazebnik

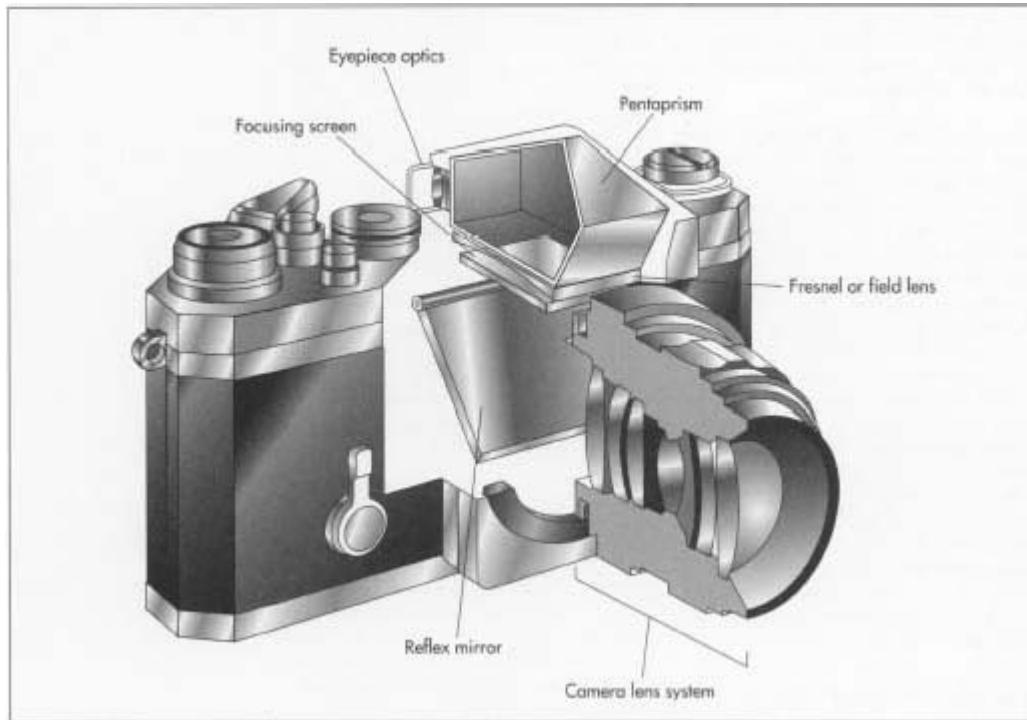
# Overview

---

- Single camera geometry
  - Recap of Homogenous coordinates
  - Perspective projection model
  - Camera calibration
- Stereo Reconstruction
  - Epipolar geometry
  - Stereo correspondence
  - Triangulation

# Single camera geometry

---



# Projection

---



CoolOpticalIllusions.com

JULIAN BELVER

# Projection

---

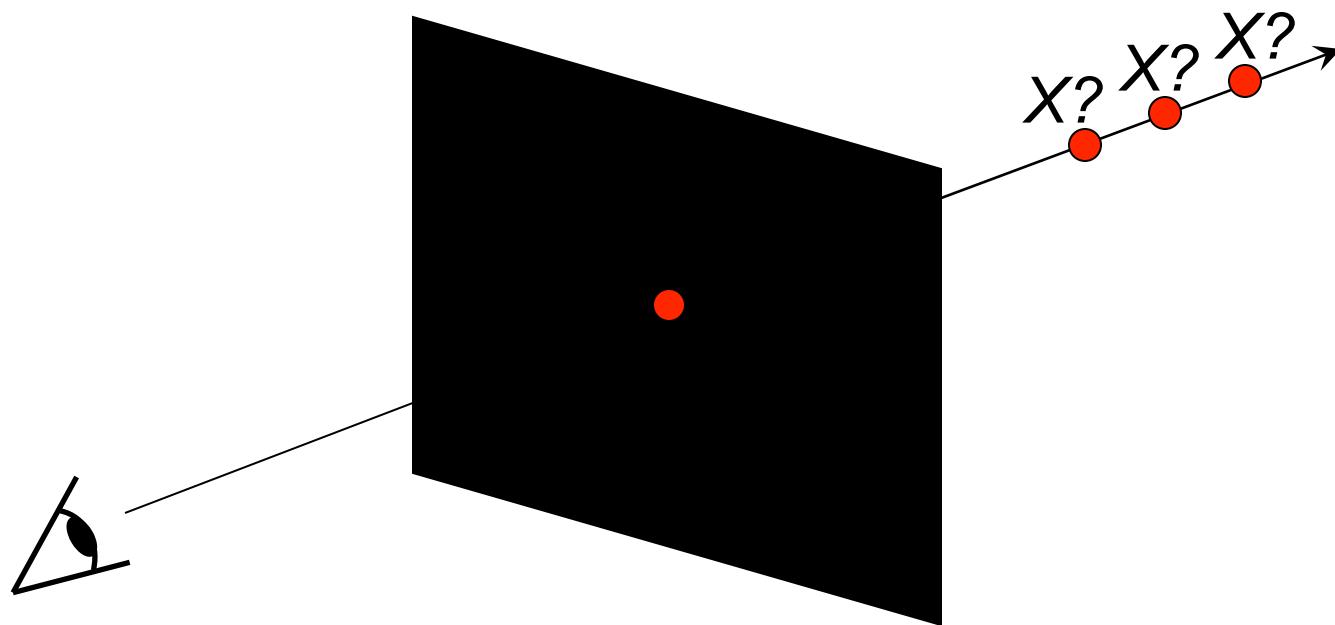


CoolOpticalIllusions.com

# Projective Geometry

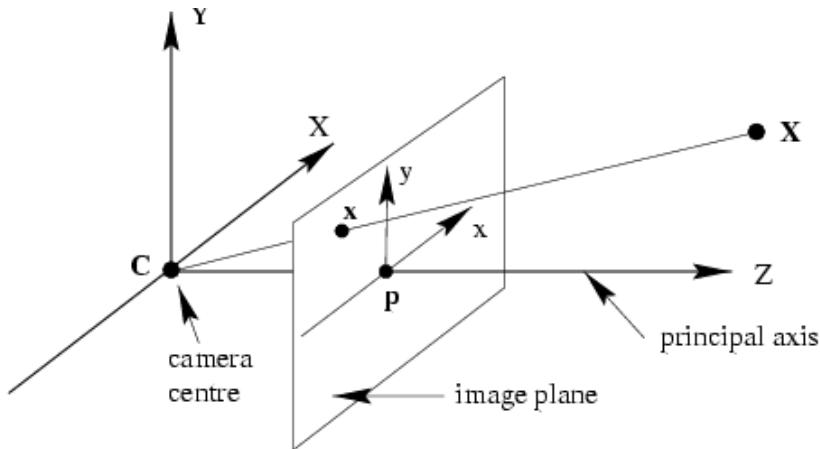
---

- Recovery of structure from one image is inherently ambiguous
- Today focus on geometry that maps world to camera image



# Recall: Pinhole camera model

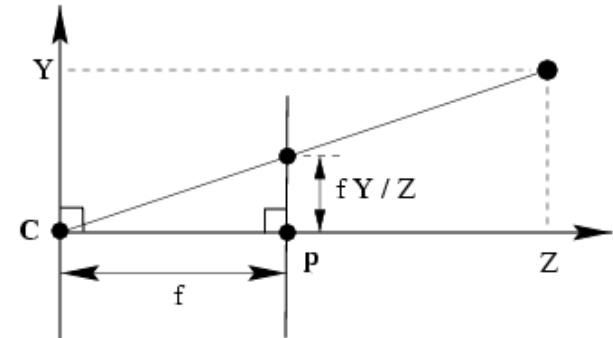
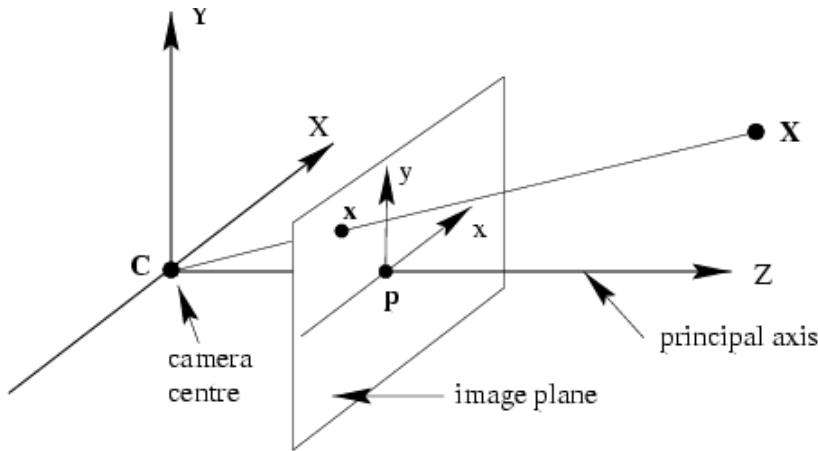
---



- **Principal axis:** line from the camera center perpendicular to the image plane
- **Normalized (camera) coordinate system:** camera center is at the origin and the principal axis is the z-axis

# Recall: Pinhole camera model

---



$$(X, Y, Z) \mapsto (fX/Z, fY/Z)$$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad \mathbf{X} = \mathbf{P}\mathbf{X}$$

# Recap: Homogeneous coordinates

---

- Is this a linear transformation?  $(x, y, z) \rightarrow (f \frac{x}{z}, f \frac{y}{z})$ 
  - no—division by  $z$  is nonlinear

Trick: add one more coordinate:

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image  
coordinates

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

homogeneous scene  
coordinates

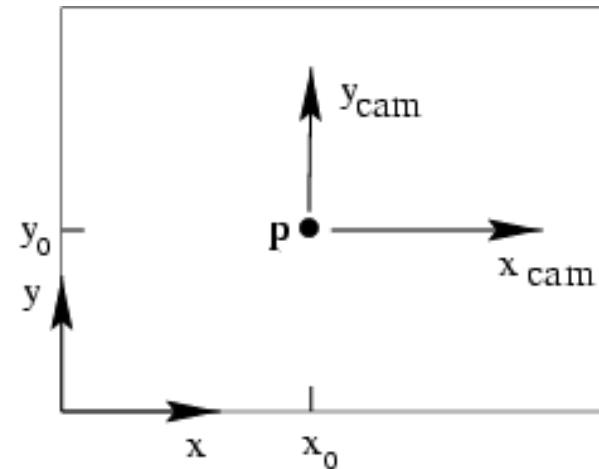
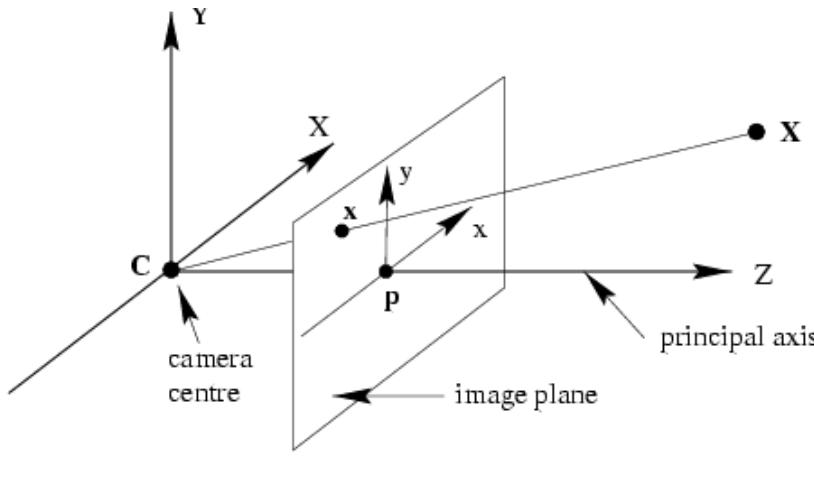
Converting *from* homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

# Principal point

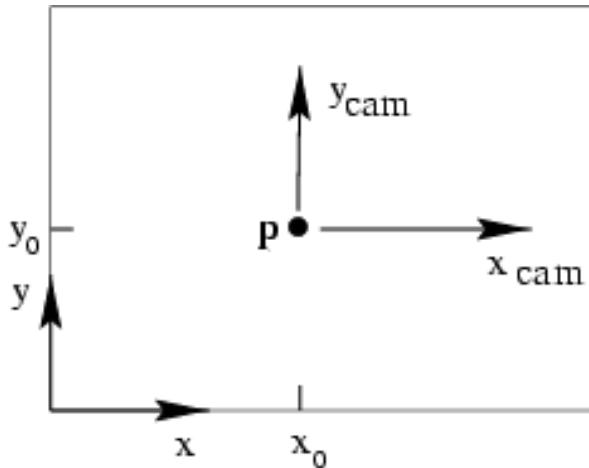
---



- **Principal point (p):** point where principal axis intersects the image plane (origin of normalized coordinate system)
- Normalized coordinate system: origin is at the principal point
- Image coordinate system: origin is in the corner
- How to go from normalized coordinate system to image coordinate system?

# Principal point offset

---



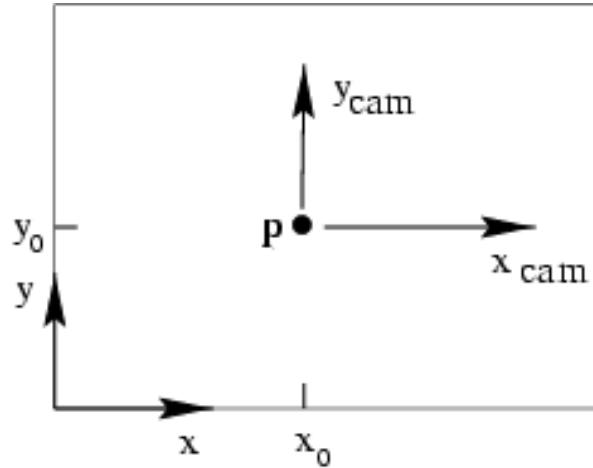
principal point:  $(p_x, p_y)$

$$(X, Y, Z) \mapsto (fX/Z + p_x, fY/Z + p_y)$$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} f & p_x & 0 \\ f & p_y & 0 \\ 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

# Principal point offset

---



principal point:  $(p_x, p_y)$

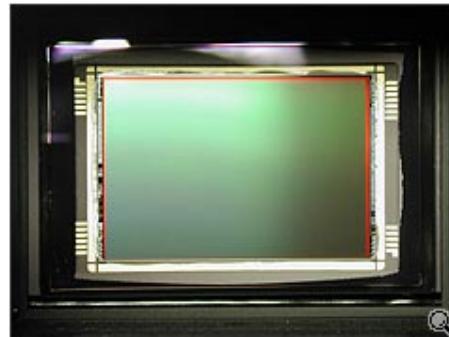
$$\begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{bmatrix} f & p_x \\ f & p_y \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & X \\ 1 & 0 & Y \\ 1 & 0 & Z \\ 1 & 0 & 1 \end{bmatrix}$$

$$K = \begin{bmatrix} f & p_x \\ f & p_y \\ 1 \end{bmatrix}$$

calibration matrix

$$P = K[I | 0]$$

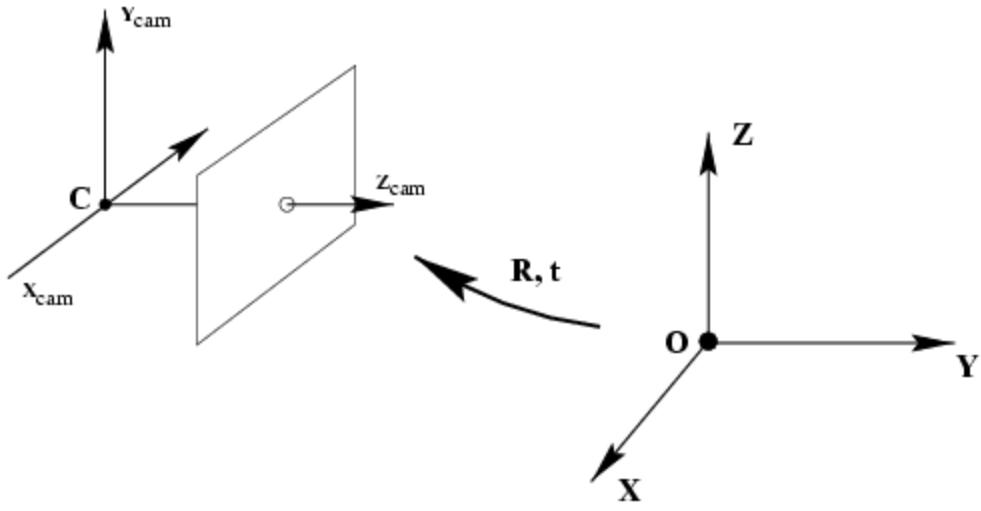
# Pixel coordinates



**Pixel size:**  $\frac{1}{m_x} \times \frac{1}{m_y}$

- $m_x$  pixels per meter in horizontal direction,  
 $m_y$  pixels per meter in vertical direction

# Camera rotation and translation

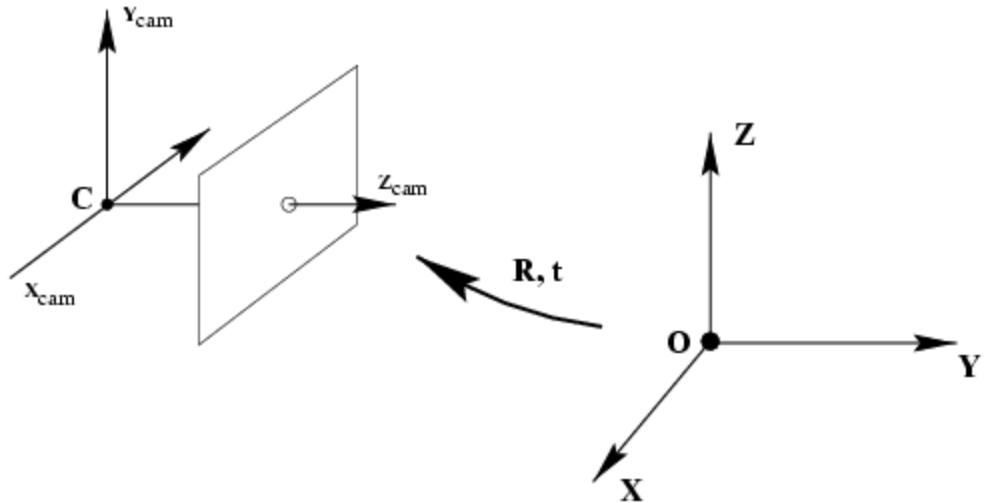


- In general, the camera coordinate frame will be related to the world coordinate frame by a rotation and a translation

$$\tilde{X}_{\text{cam}} = R(\tilde{X} - \tilde{C})$$

coords. of point in camera frame      coords. of camera center in world frame  
coords. of a point in world frame (nonhomogeneous)

# Camera rotation and translation



In non-homogeneous coordinates:

$$\tilde{X}_{\text{cam}} = R(\tilde{X} - \tilde{C})$$

$$X_{\text{cam}} = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \tilde{X} \\ 1 \end{pmatrix} = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} X$$

$$x = K[I | 0]X_{\text{cam}} = K[R | -R\tilde{C}]X \quad P = K[R | t] \quad t = -R\tilde{C}$$

Note: C is the null space of the camera projection matrix (PC=0)

# Camera parameters

---

- Intrinsic parameters

- Principal point coordinates
- Focal length
- Pixel magnification factors
- *Skew (non-rectangular pixels)*
- *Radial distortion*

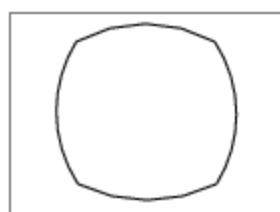
$$K = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} f & p_x \\ f & p_y \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & \beta_x \\ \alpha_y & \beta_y \\ 1 & 1 \end{bmatrix}$$



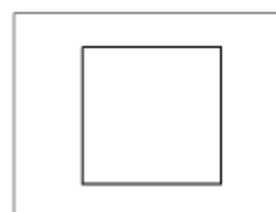
radial distortion



linear image



correction



# Camera parameters

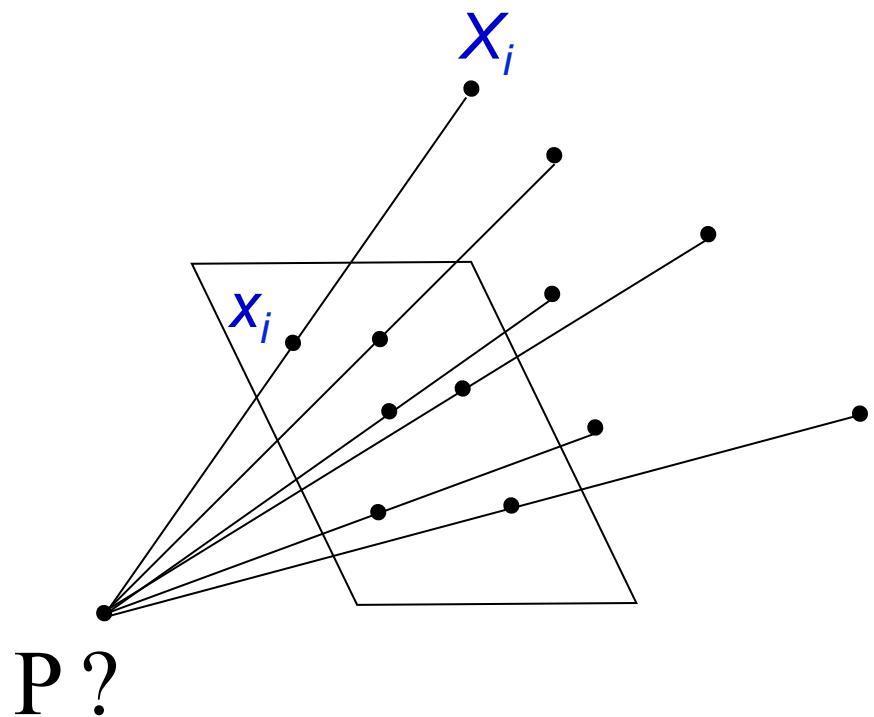
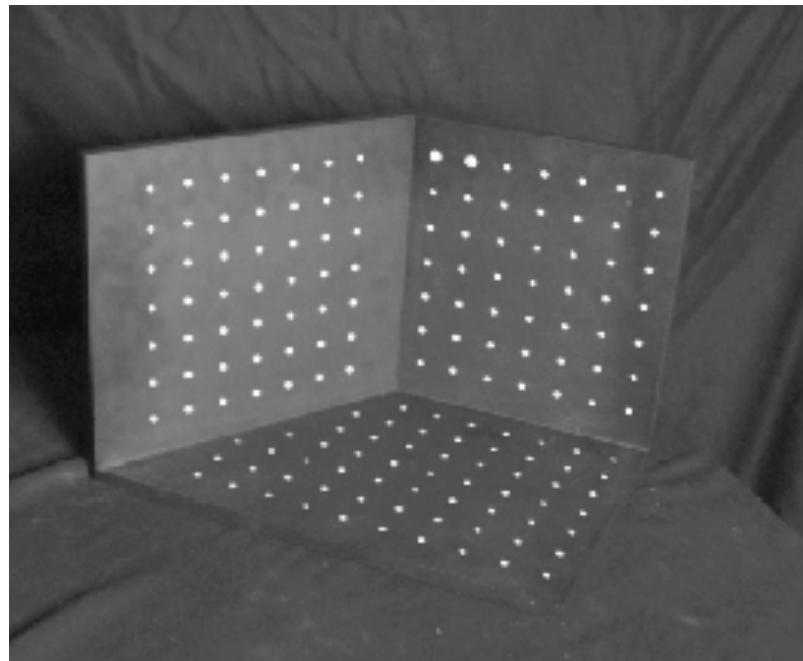
---

- Intrinsic parameters
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew (non-rectangular pixels)*
  - *Radial distortion*
- Extrinsic parameters
  - Rotation and translation relative to world coordinate system

# Camera calibration

---

- Given  $n$  points with known 3D coordinates  $X_i$  and known image projections  $x_i$ , estimate the camera parameters



# Camera calibration

---

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

$$\mathbf{x}_i \times \mathbf{P} \mathbf{X}_i = 0$$

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \times \begin{bmatrix} \mathbf{P}_1^T \mathbf{X}_i \\ \mathbf{P}_2^T \mathbf{X}_i \\ \mathbf{P}_3^T \mathbf{X}_i \end{bmatrix} = 0$$

$$\begin{bmatrix} 0 & -\mathbf{X}_i^T & y_i \mathbf{X}_i^T \\ \mathbf{X}_i^T & 0 & -x_i \mathbf{X}_i^T \\ -y_i \mathbf{X}_i^T & x_i \mathbf{X}_i^T & 0 \end{bmatrix} \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix} = 0$$

Two linearly independent equations

# Camera calibration

---

$$\begin{bmatrix} 0^T & X_1^T & -y_1 X_1^T \\ X_1^T & 0^T & -x_1 X_1^T \\ \dots & \dots & \dots \\ 0^T & X_n^T & -y_n X_n^T \\ X_n^T & 0^T & -x_n X_n^T \end{bmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} = 0 \quad Ap = 0$$

- $P$  has 11 degrees of freedom (12 parameters, but scale is arbitrary)
- One 2D/3D correspondence gives us two linearly independent equations
- Homogeneous least squares
- 6 correspondences needed for a minimal solution

# Camera calibration

---

$$\begin{bmatrix} 0^T & \mathbf{X}_1^T & -y_1\mathbf{X}_1^T \\ \mathbf{X}_1^T & 0^T & -x_1\mathbf{X}_1^T \\ \dots & \dots & \dots \\ 0^T & \mathbf{X}_n^T & -y_n\mathbf{X}_n^T \\ \mathbf{X}_n^T & 0^T & -x_n\mathbf{X}_n^T \end{bmatrix} \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix} = 0 \quad \mathbf{A}\mathbf{p} = 0$$

- Note: for coplanar points that satisfy  $\Pi^T \mathbf{X} = 0$ , we will get degenerate solutions  $(\Pi, 0, 0)$ ,  $(0, \Pi, 0)$ , or  $(0, 0, \Pi)$

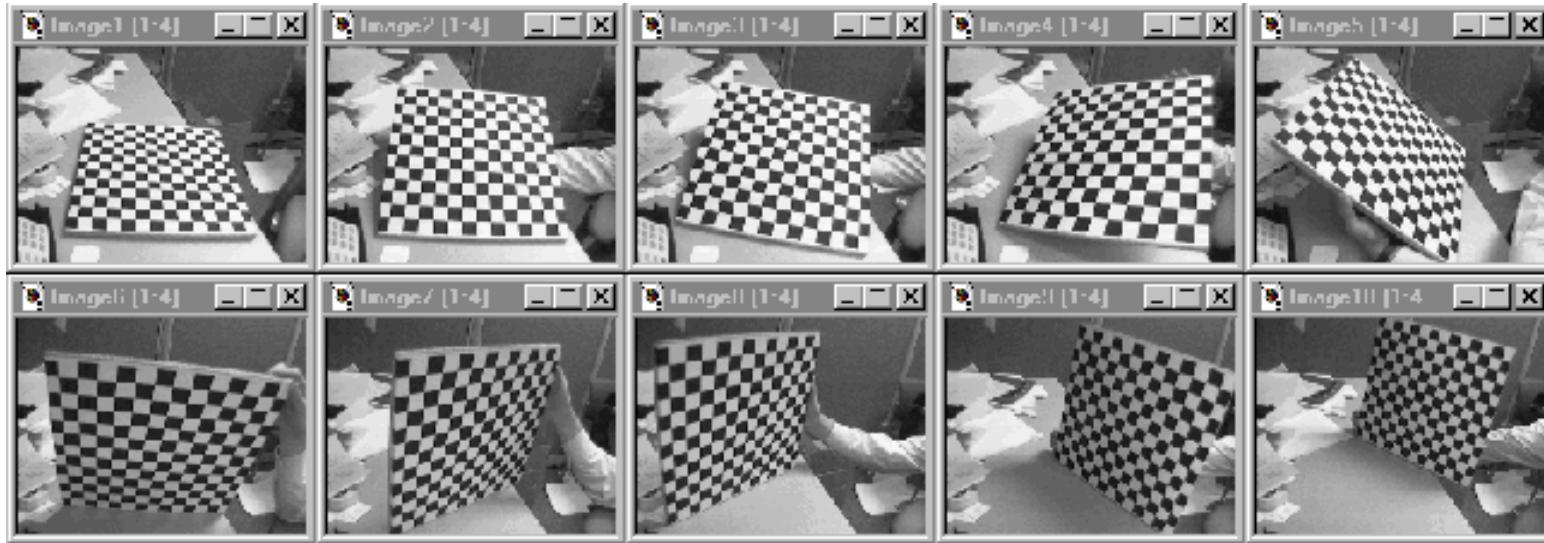
# Camera calibration

---

- Once we've recovered the numerical form of the camera matrix, we still have to figure out the intrinsic and extrinsic parameters
- This is a matrix decomposition problem, not an estimation problem (see F&P sec. 3.2, 3.3)

# Alternative: multi-plane calibration

---



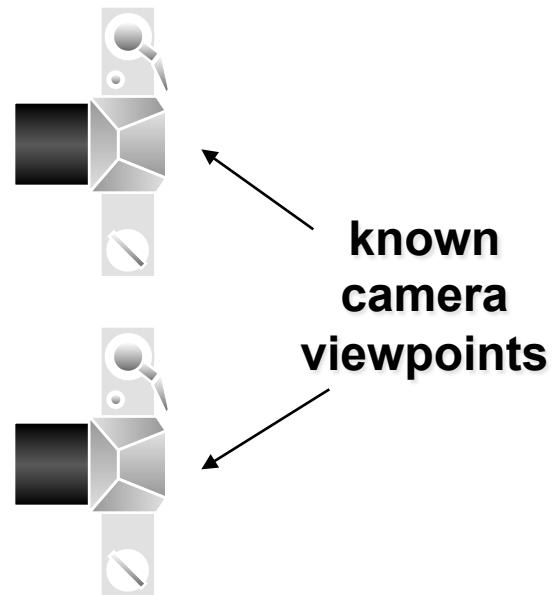
Images courtesy Jean-Yves Bouguet, Intel Corp.

## Advantage

- Only requires a plane
- Don't have to know positions/orientations
- Good code available online!
  - Intel's OpenCV library: <http://www.intel.com/research/mrl/research/opencv/>
  - Matlab version by Jean-Yves Bouguet:  
[http://www.vision.caltech.edu/bouguetj/calib\\_doc/index.html](http://www.vision.caltech.edu/bouguetj/calib_doc/index.html)
  - Zhengyou Zhang's web site: <http://research.microsoft.com/~zhang/Calib/>

# Stereo Reconstruction

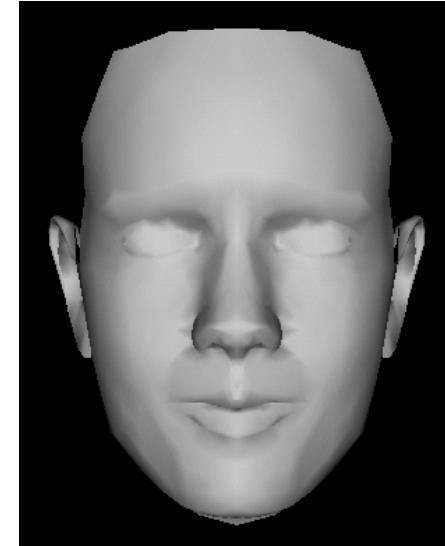
Shape (3D) from two (or more) images



# Example

---

images



shape



surface  
reflectance

# Scenarios

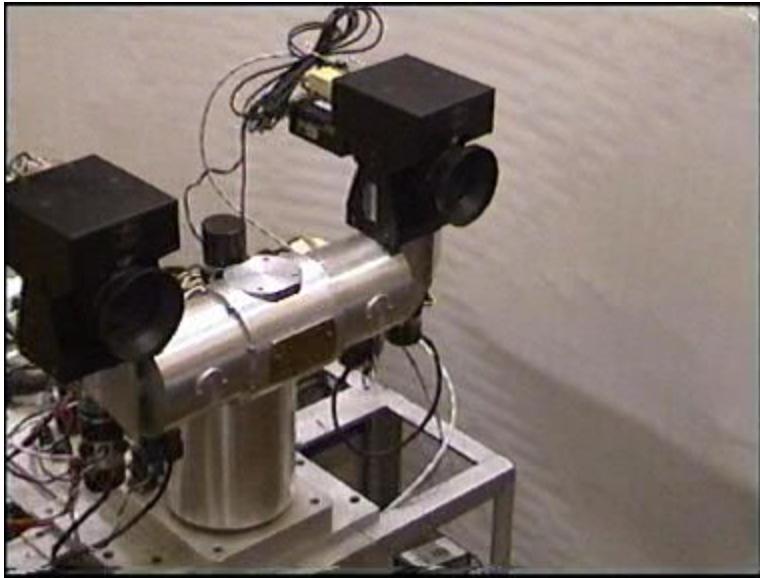
---

The two images can arise from

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



(COURTESY SONY)

# The objective

---

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



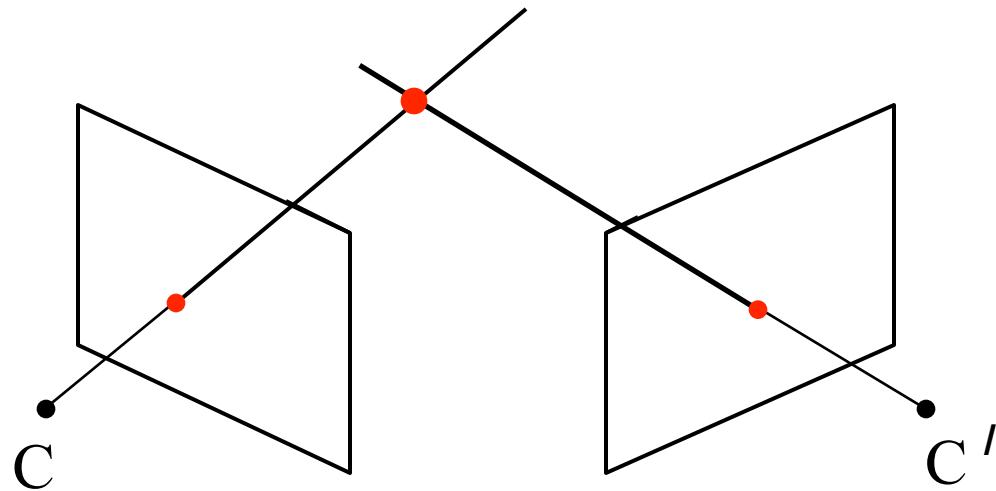
Basic principle: triangulate from corresponding image points

- 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



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

# Outline

---

## 1. Epipolar geometry

- the geometry of two cameras
- reduces the correspondence problem to a line search

## 2. Stereo correspondence algorithms

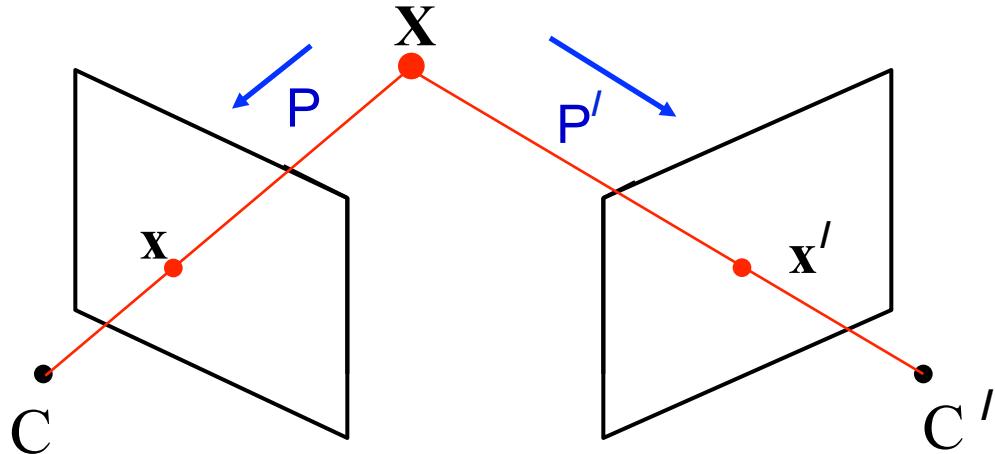
## 3. Triangulation

# Notation

---

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

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



$P$  :  $3 \times 4$  matrix

$\mathbf{X}$  : 4-vector

$\mathbf{x}$  : 3-vector

## Warning

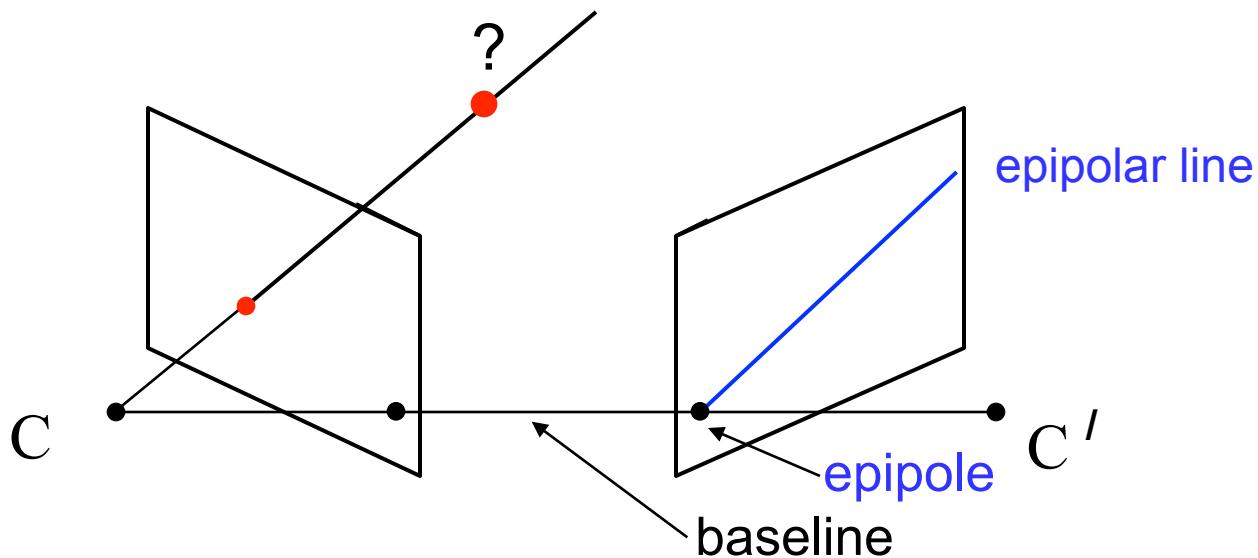
for equations involving homogeneous quantities '=' means 'equal up to scale'

# 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

---



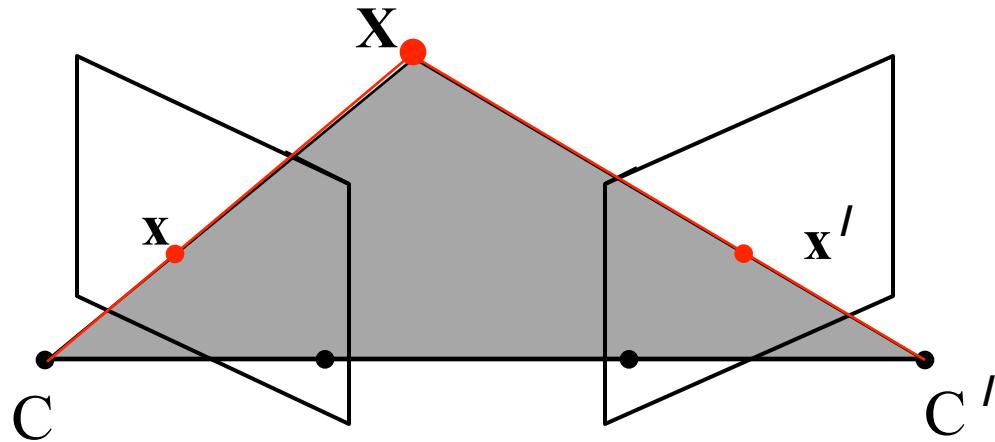
## Epipolar constraint

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

# Epipolar geometry continued

---

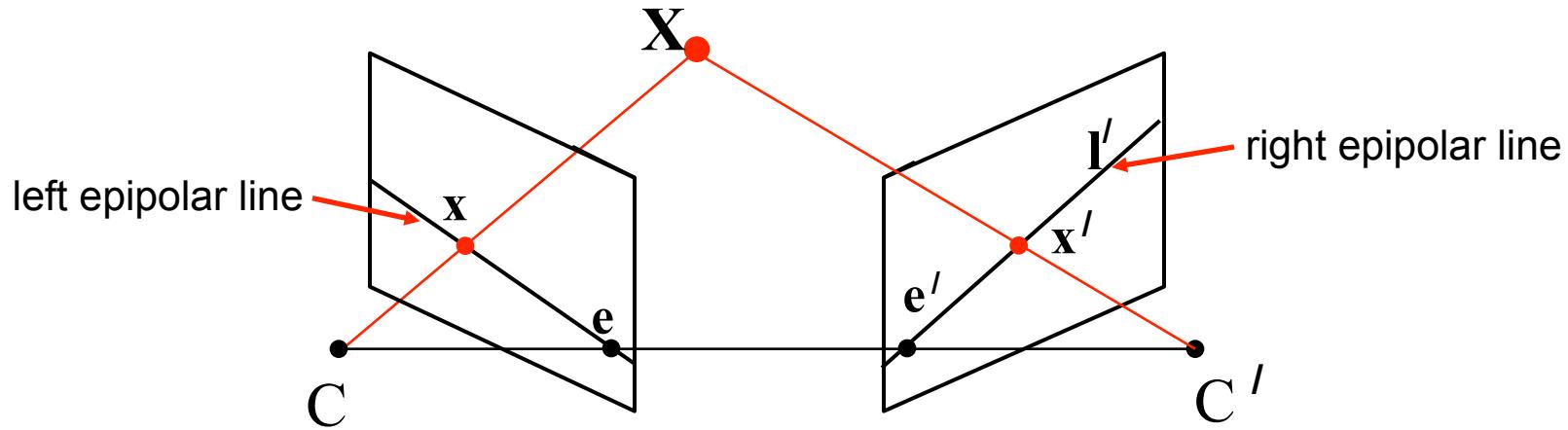
Epipolar geometry is a consequence of the **coplanarity** of the camera centres and scene point



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

# Nomenclature

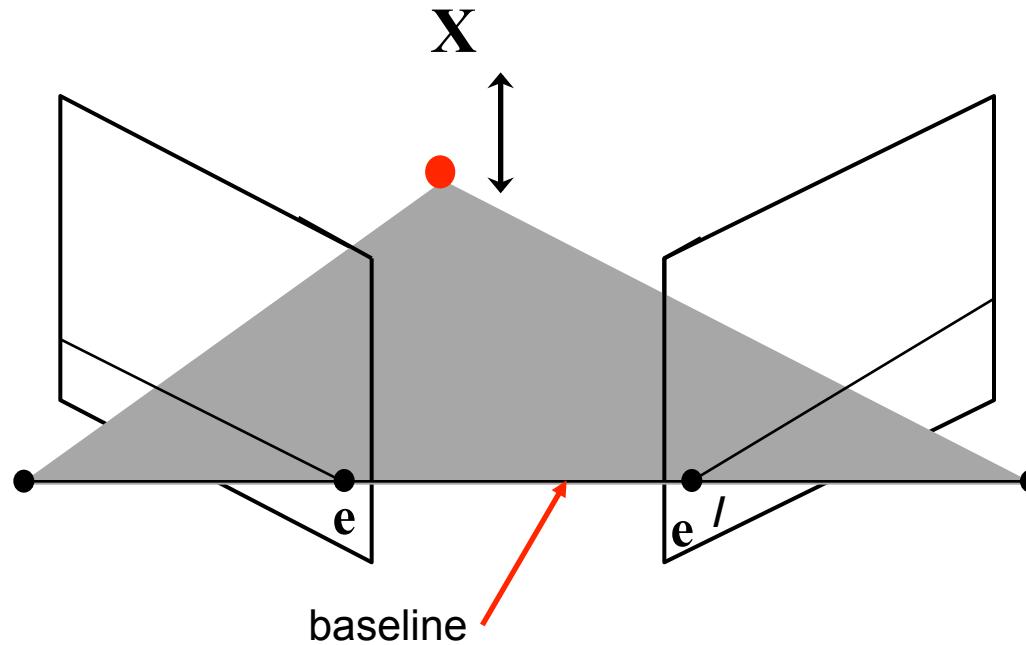
---



- The **epipolar line  $I'$**  is the image of the ray through  $x$
- The **epipole  $e$**  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:  $e = PC'$ ,  $e' = P'C$

# The epipolar pencil

---

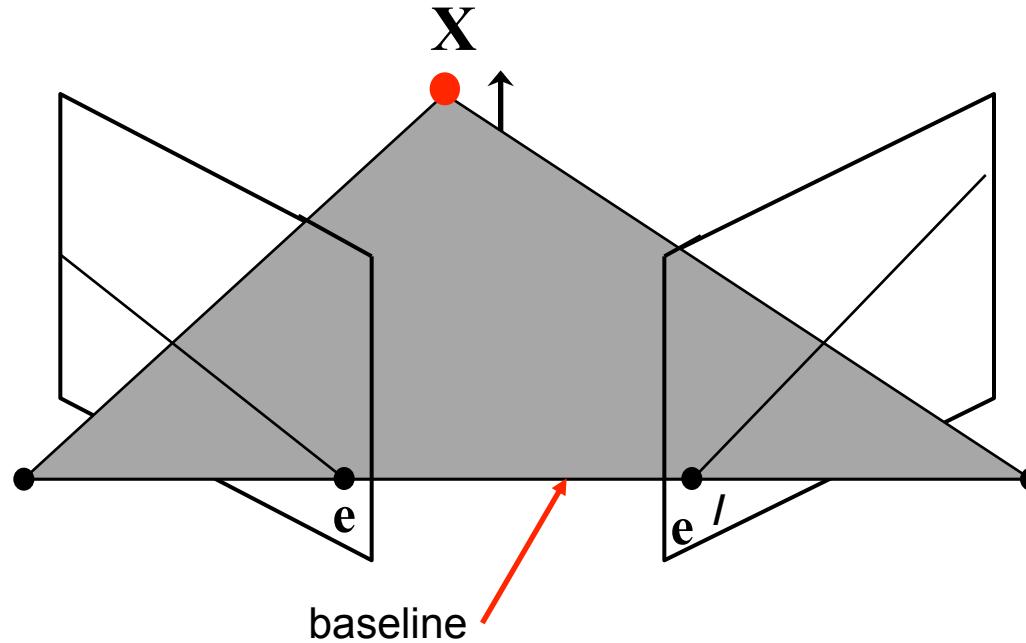


As the position of the 3D point  $\mathbf{X}$  varies, the epipolar planes “rotate” about the baseline. This family of planes is known as an **epipolar pencil**. All epipolar lines intersect at the epipole.

(a pencil is a one parameter family)

# The epipolar pencil

---

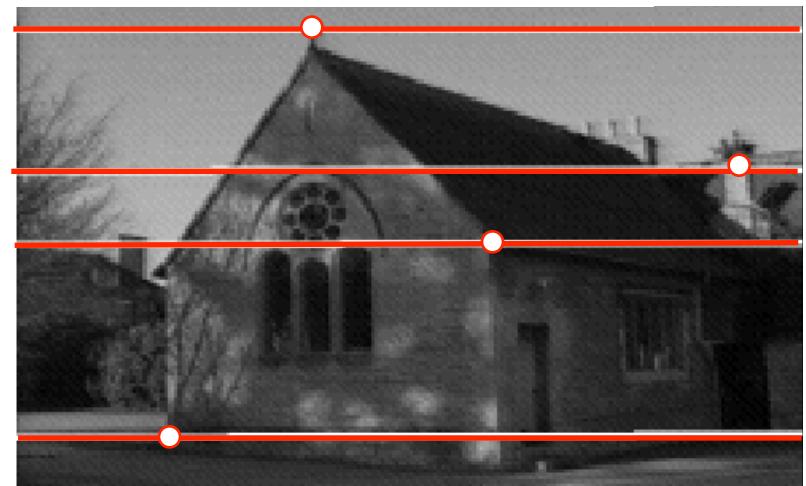
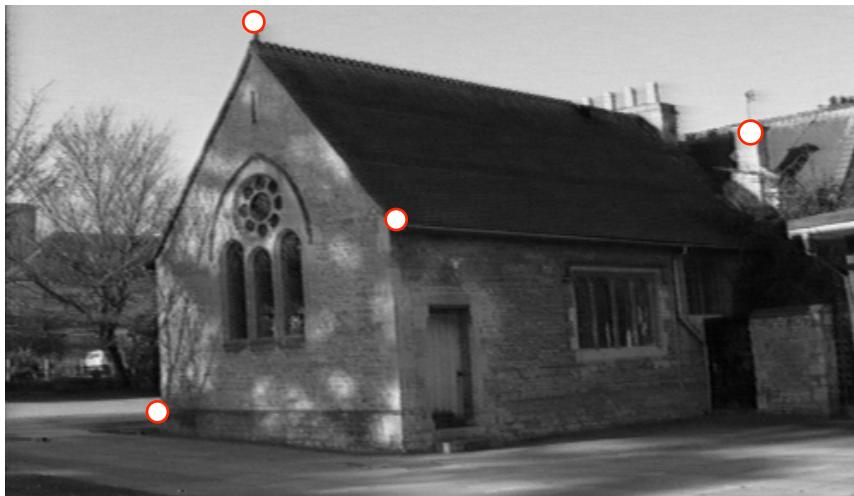
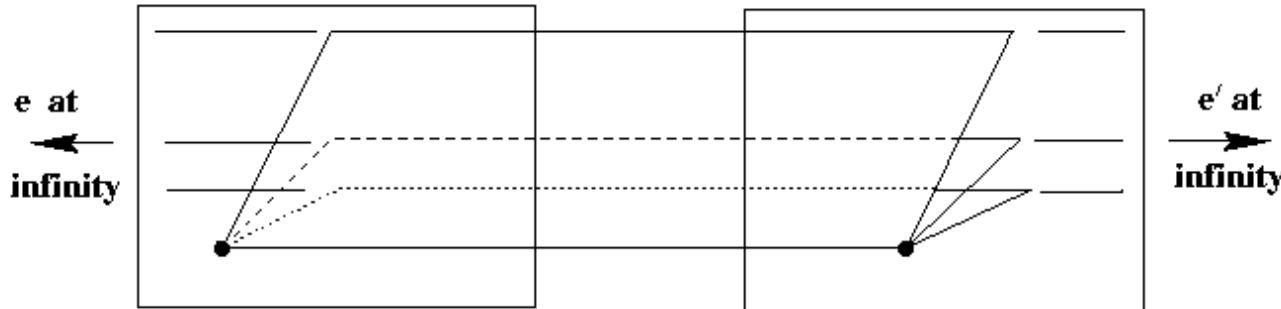


As the position of the 3D point  $\mathbf{X}$  varies, the epipolar planes “rotate” about the baseline. This family of planes is known as an **epipolar pencil**. All epipolar lines intersect at the epipole.

(a pencil is a one parameter family)

# Epipolar geometry example I: parallel cameras

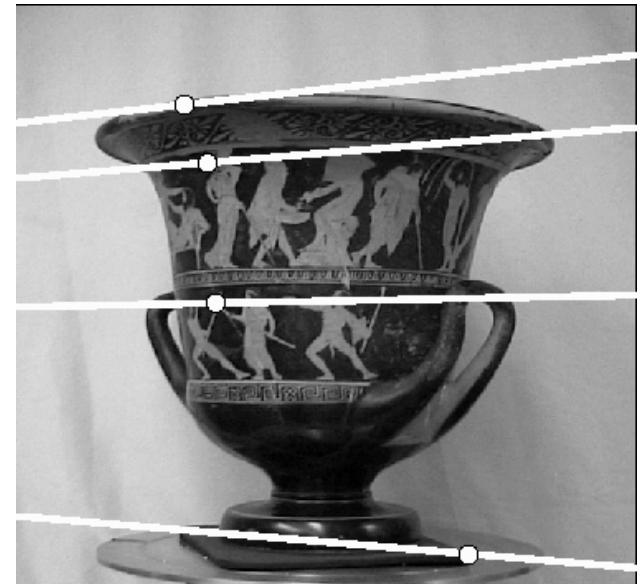
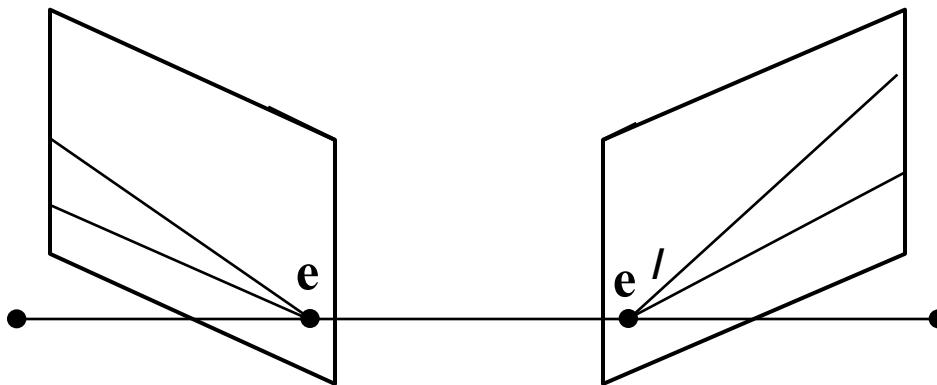
---



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.

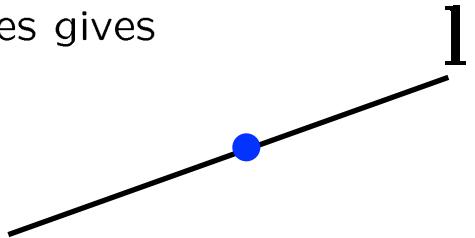
Write the line as  $-y + 1 = 0$  then  $l_1 = 0, l_2 = -1, l_3 = 1$ , and  $\mathbf{l} = (0, -1, 1)^\top$ .

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

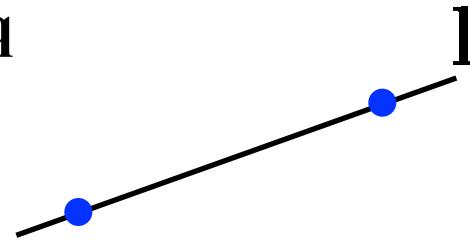
Writing both the point and line in homogeneous coordinates gives

$$l_1x_1 + l_2x_2 + l_3x_3 = 0$$

- [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

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

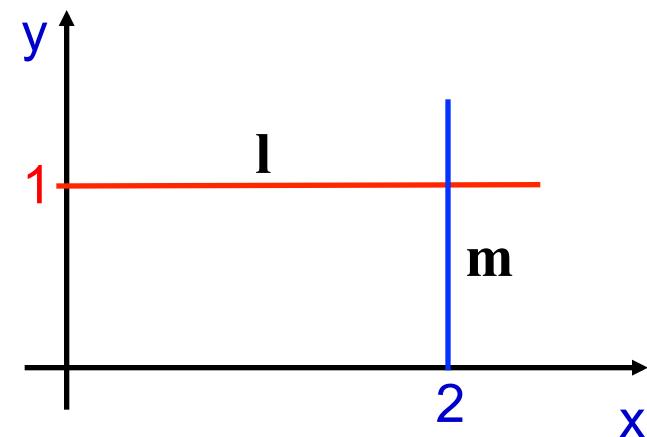
- The intersection of two lines  $\mathbf{l}$  and  $\mathbf{m}$  is the point  $\mathbf{x} = \mathbf{l} \times \mathbf{m}$

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  $3 \times 3$  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  $\mathbf{l}$  and  $\mathbf{m}$

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

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

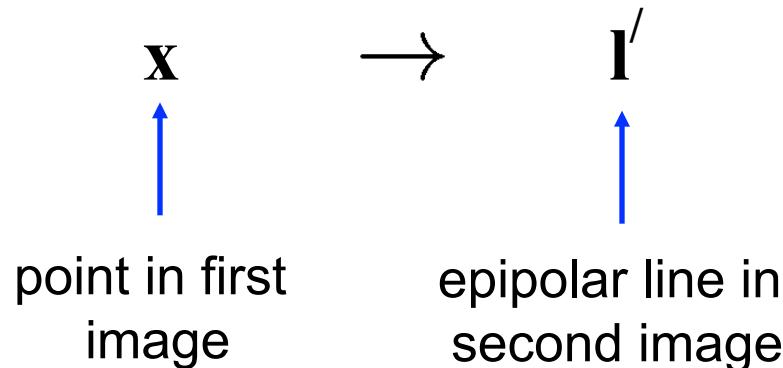
Note

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

# Algebraic representation of epipolar geometry

---

We know that the epipolar geometry defines a mapping



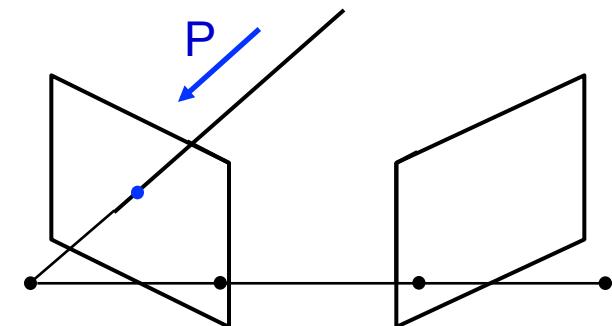
- the map only depends on the cameras  $P, P'$  (not on structure)
- it will be shown that the map is linear and can be written as  $\mathbf{l}' = F\mathbf{x}$ , where  $F$  is a  $3 \times 3$  matrix called the fundamental matrix

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

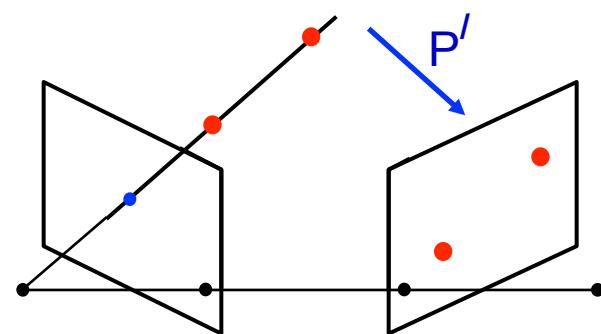
---

## Outline

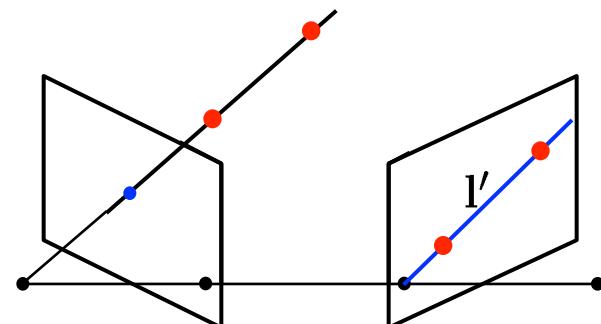
Step 1: for a point  $x$  in the first image  
back project a ray with camera  $P$



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



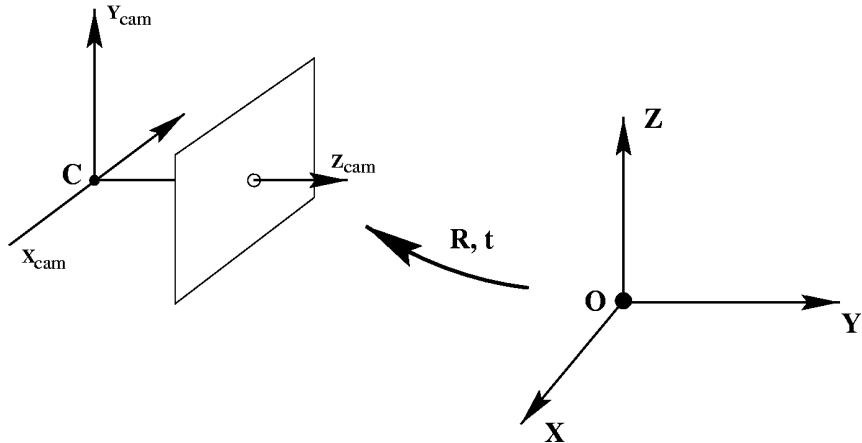
Step 3: compute the line through the two  
image points using the relation  $\mathbf{l}' = \mathbf{p} \times \mathbf{q}$



- choose camera matrices

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

internal calibration      rotation      translation  
 from world to camera coordinate frame

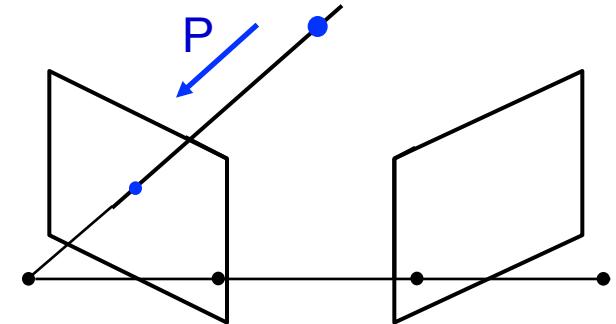


- first camera  $P = K [ I \mid 0 ]$

world coordinate frame aligned with first camera

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

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



A point  $x$  back projects to a ray

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = zK^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = zK^{-1}x$$

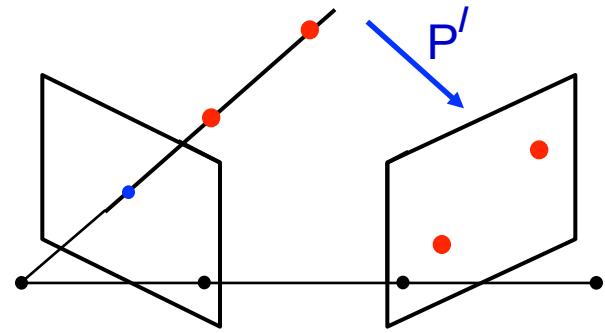
where  $Z$  is the point's depth, since

$$x(z) = \begin{pmatrix} zK^{-1}x \\ 1 \end{pmatrix}$$

satisfies

$$PX(z) = K[I \mid 0]x(z) = x$$

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



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

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

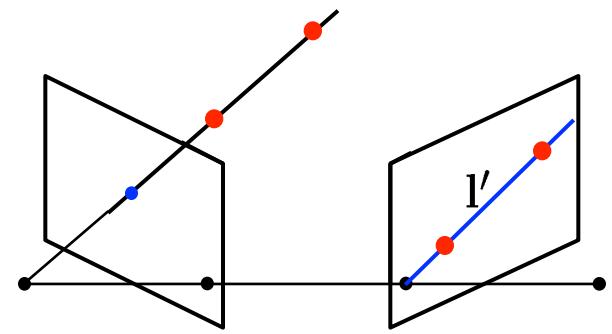
- $Z = \infty$  is the point at infinity  $\begin{pmatrix} K^{-1}\mathbf{x} \\ 0 \end{pmatrix}$

Project these two points into the second view

$$P' \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} = K'[R \mid t] \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} = K'\mathbf{t}$$

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

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  $\mathbf{l}' = (\mathbf{K}'\mathbf{t}) \times (\mathbf{K}'\mathbf{R}\mathbf{K}^{-1}\mathbf{x})$

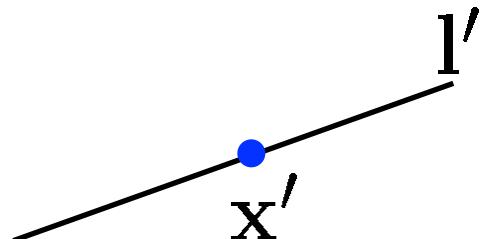
Using the identity  $(\mathbf{M}\mathbf{a}) \times (\mathbf{M}\mathbf{b}) = \mathbf{M}^{-\top}(\mathbf{a} \times \mathbf{b})$  where  $\mathbf{M}^{-\top} = (\mathbf{M}^{-1})^\top = (\mathbf{M}^\top)^{-1}$

$$\mathbf{l}' = \mathbf{K}'^{-\top} \left( \mathbf{t} \times (\mathbf{R}\mathbf{K}^{-1}\mathbf{x}) \right) = \underbrace{\mathbf{K}'^{-\top} [\mathbf{t}]_\times \mathbf{R}\mathbf{K}^{-1}}_{\mathbf{F}} \mathbf{x} \quad \mathbf{F} \text{ is the fundamental matrix}$$

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

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

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



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

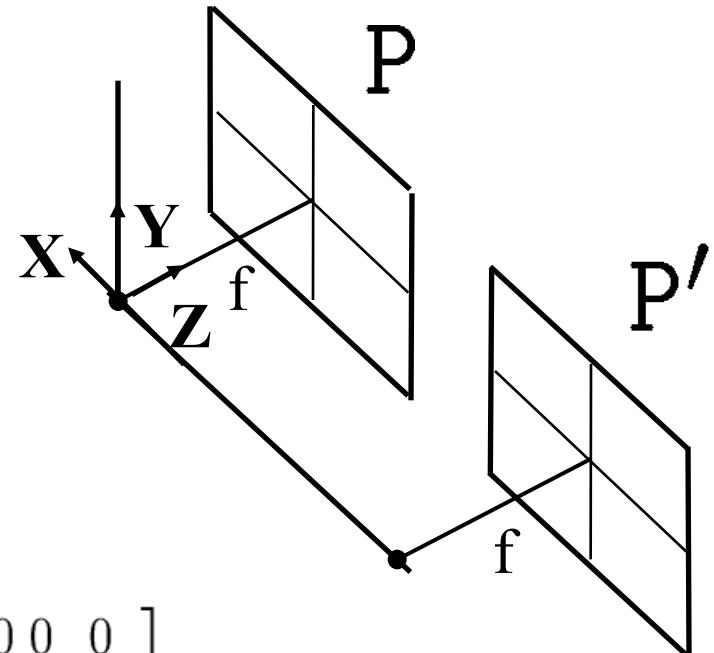
$$P = K[I \mid \mathbf{0}] \quad P' = K'[R \mid t]$$

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

$$F = K'^{-\top}[t]_x R K^{-1}$$

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

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



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

$F$  is a rank 2 matrix

The epipole  $e$  is the null-space vector (kernel) of  $F$  (exercise), i.e.  $Fe = 0$

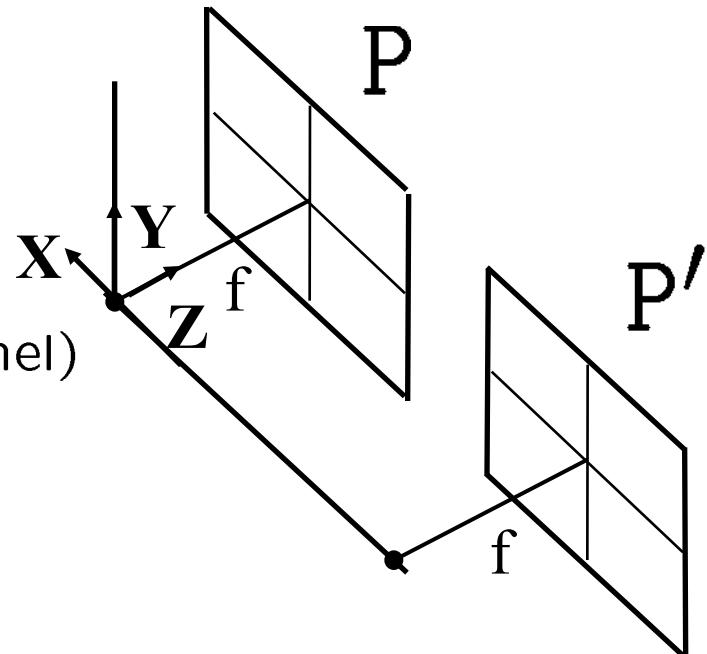
In this case

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

so that

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

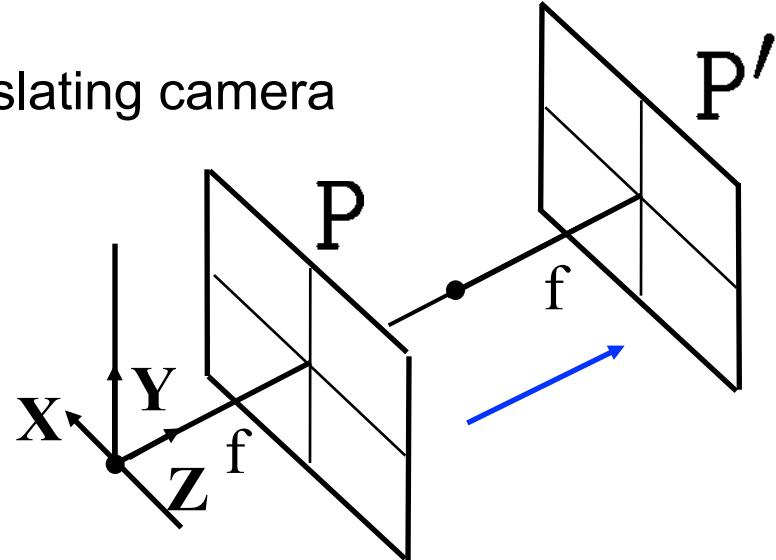
Geometric interpretation ?



## Example II: compute F for a forward translating camera

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

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

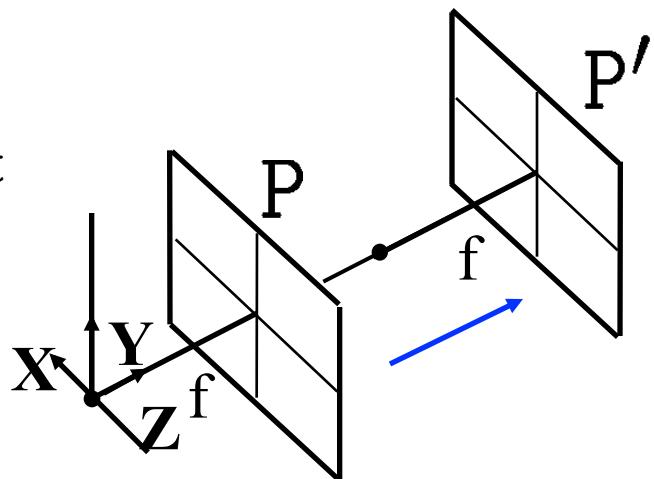


$$F = K'^{-T}[t] \times R K^{-1}$$

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

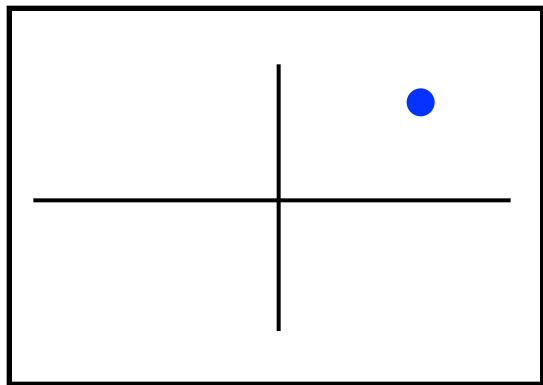
From  $\mathbf{l}' = \mathbf{F}\mathbf{x}$  the epipolar line for the point  $\mathbf{x} = (x, y, 1)^\top$  is

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

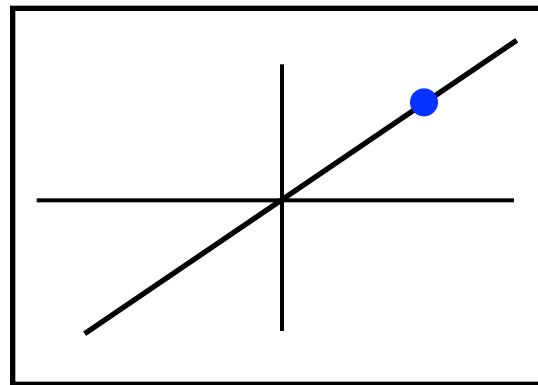


The points  $(x, y, 1)^\top$  and  $(0, 0, 1)^\top$  lie on this line

first image



second image







# Summary: Properties of the Fundamental matrix

---

- $F$  is a rank 2 homogeneous matrix with 7 degrees of freedom.
- Point correspondence:  
if  $x$  and  $x'$  are corresponding image points, then  
 $x'^T F x = 0$ .
- Epipolar lines:
  - ◊  $l' = Fx$  is the epipolar line corresponding to  $x$ .
  - ◊  $l = F^T x'$  is the epipolar line corresponding to  $x'$ .
- Epipoles:
  - ◊  $F\mathbf{e} = \mathbf{0}$ .
  - ◊  $F^T \mathbf{e}' = \mathbf{0}$ .
- Computation from camera matrices  $P, P'$ :  
 $P = K[I \mid \mathbf{0}], P' = K'[R \mid t], F = K'^{-T}[t]_x R K^{-1}$

## Admin Interlude

---

- Assignment 1 due this time next week
  - So come to office hours right after class!
- Meet Chaitanya Rudra ([cr1512@nyu.edu](mailto:cr1512@nyu.edu)) who is the class TA

# Stereo correspondence algorithms

# Problem statement

---

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

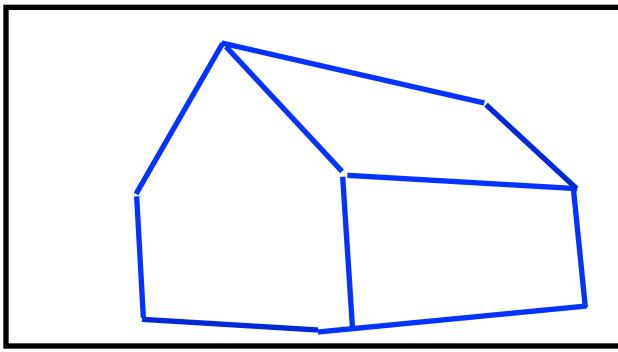
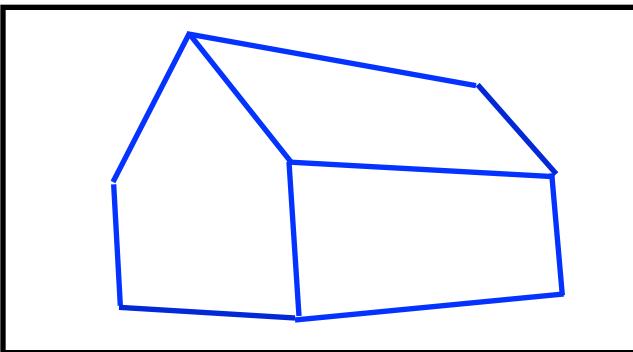
Algorithms may be classified into two types:

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

---

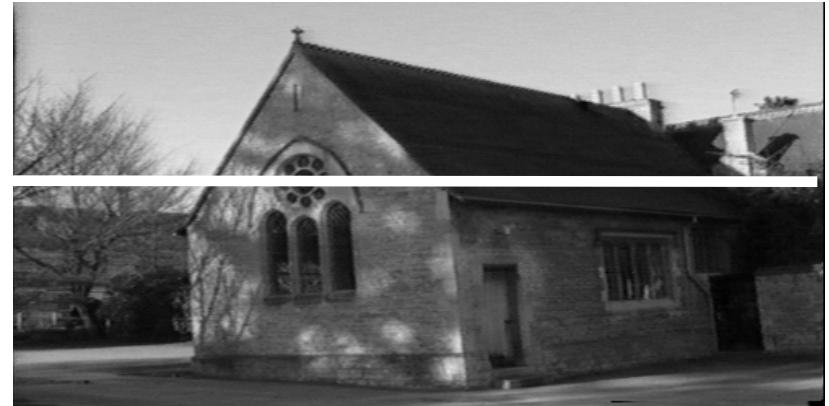


1. Group model (house, windows, etc) independently in each image
2. Match points (vertices) between images

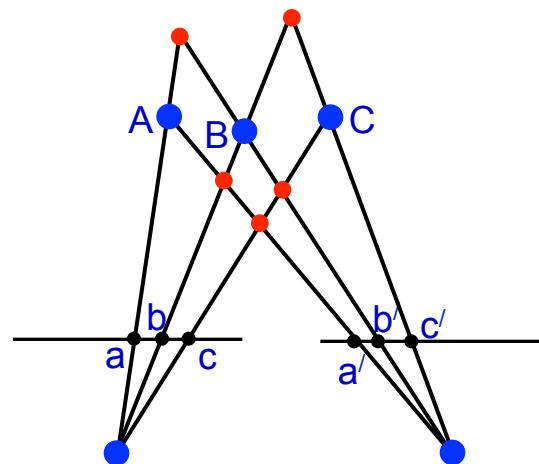
# Bottom up matching

---

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



- 1D correspondence problem

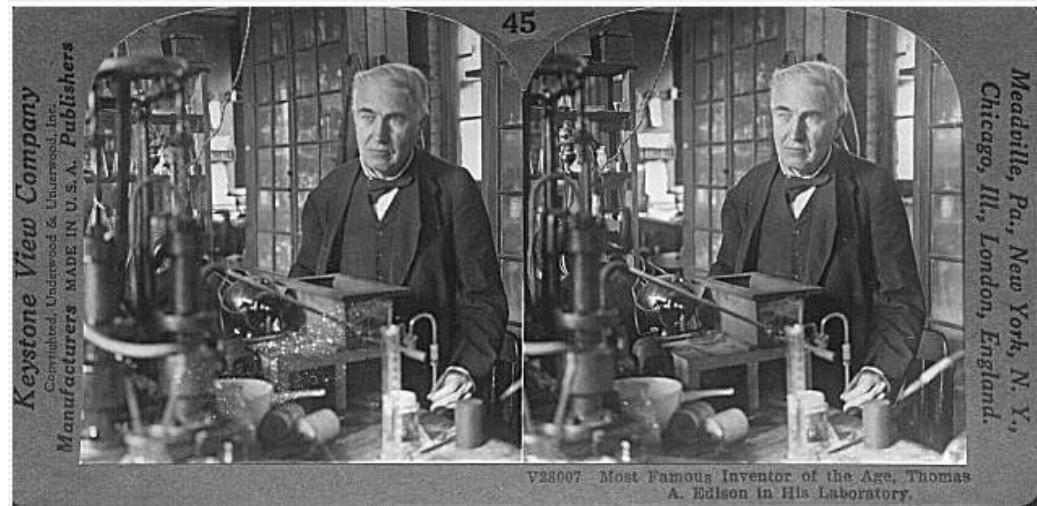




# Stereograms

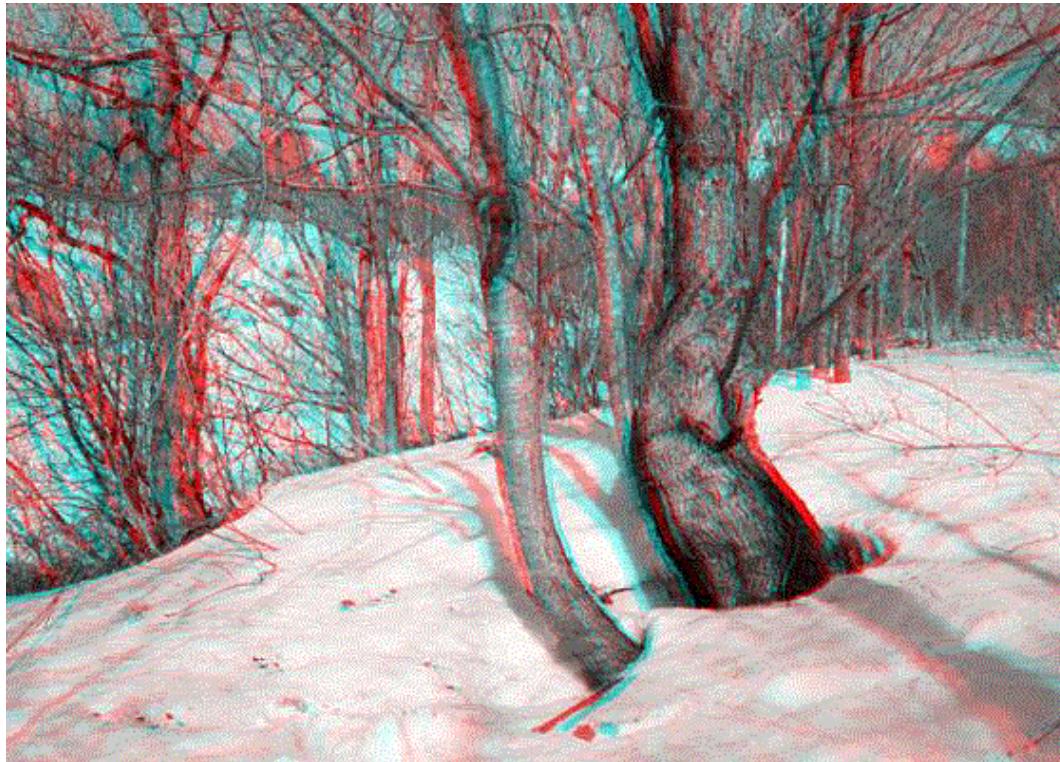
---

- Invented by Sir Charles Wheatstone, 1838



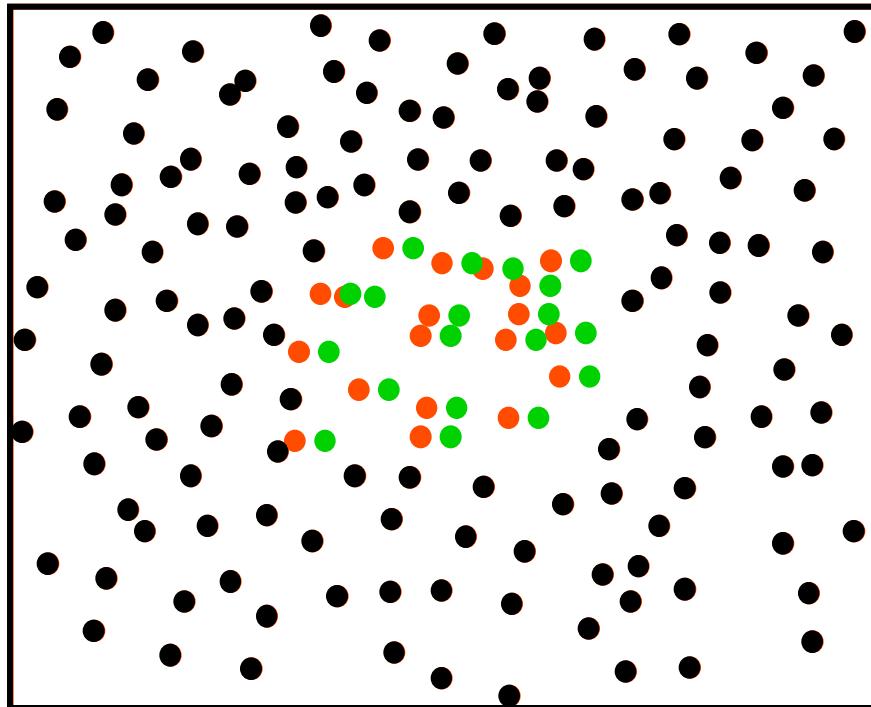
# Red/green stereograms

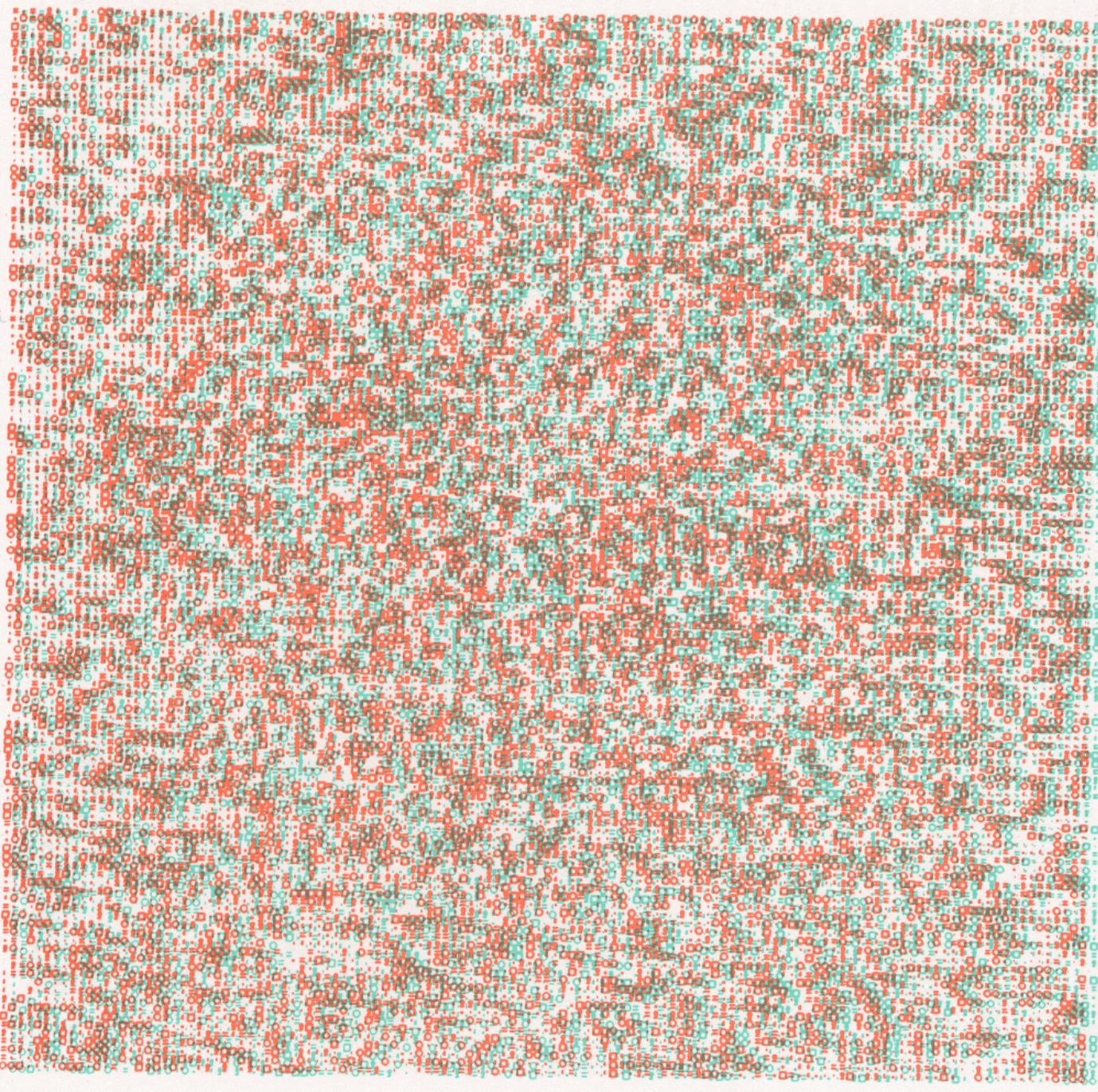
---



# Random dot stereograms

---





# Autostereograms

---



Autostereograms: [www.magiceye.com](http://www.magiceye.com)

# Autostereograms

---



Autostereograms: [www.magiceye.com](http://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

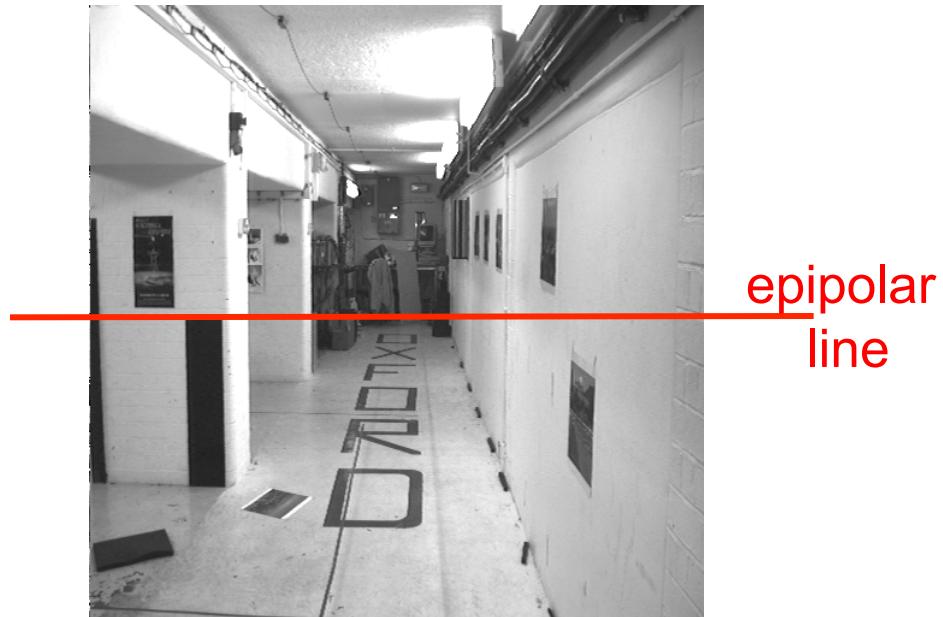
---



# Dense correspondence algorithm

---

Parallel camera example – epipolar lines are corresponding rasters



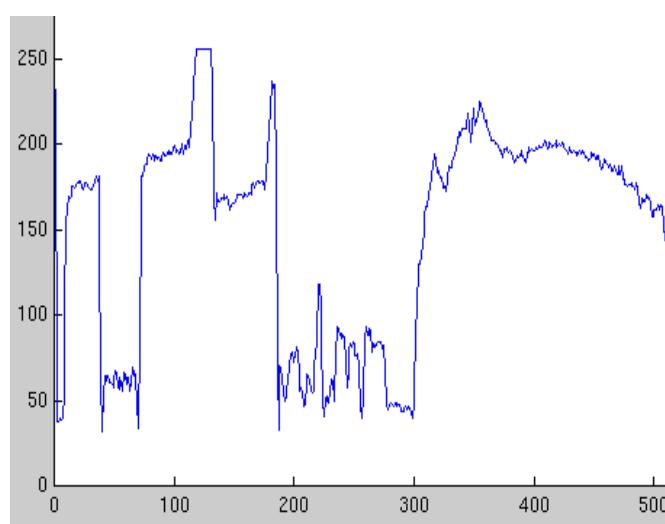
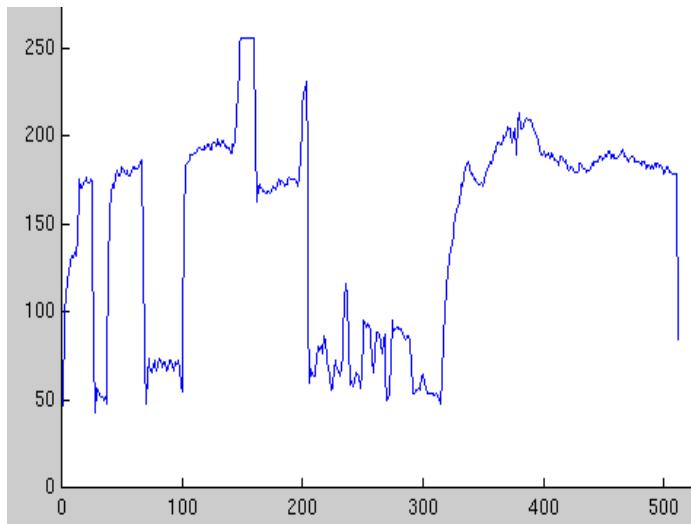
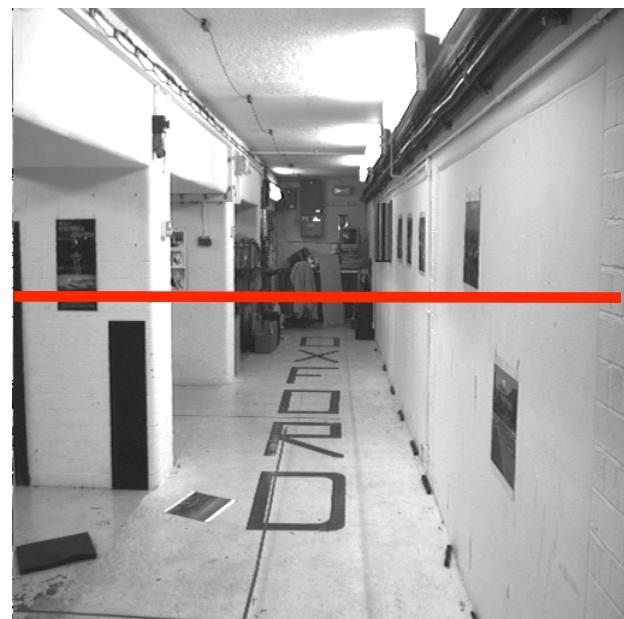
epipolar  
line

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

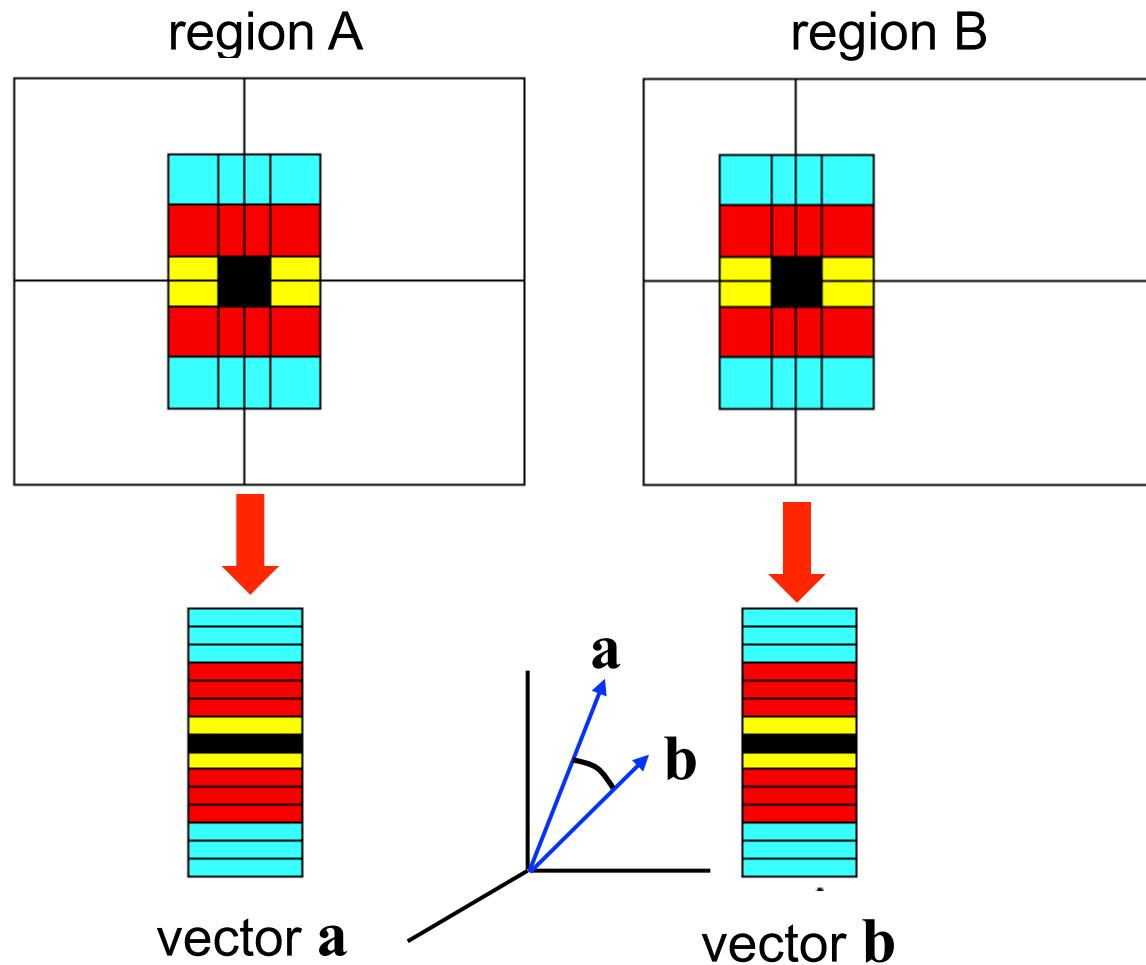
$$NCC = \frac{\sum_i \sum_j A(i, j)B(i, j)}{\sqrt{\sum_i \sum_j A(i, j)^2} \sqrt{\sum_i \sum_j B(i, j)^2}}$$

write regions as vectors

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

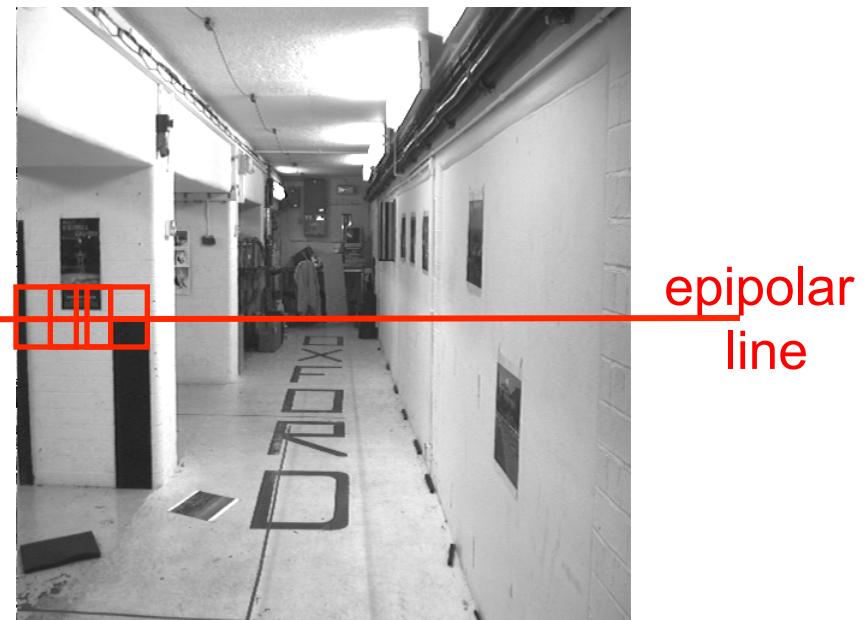
$$NCC = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| |\mathbf{b}|}$$

$$-1 \leq NCC \leq 1$$



# Cross-correlation of neighbourhood regions

---



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

translate so that mean is zero

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

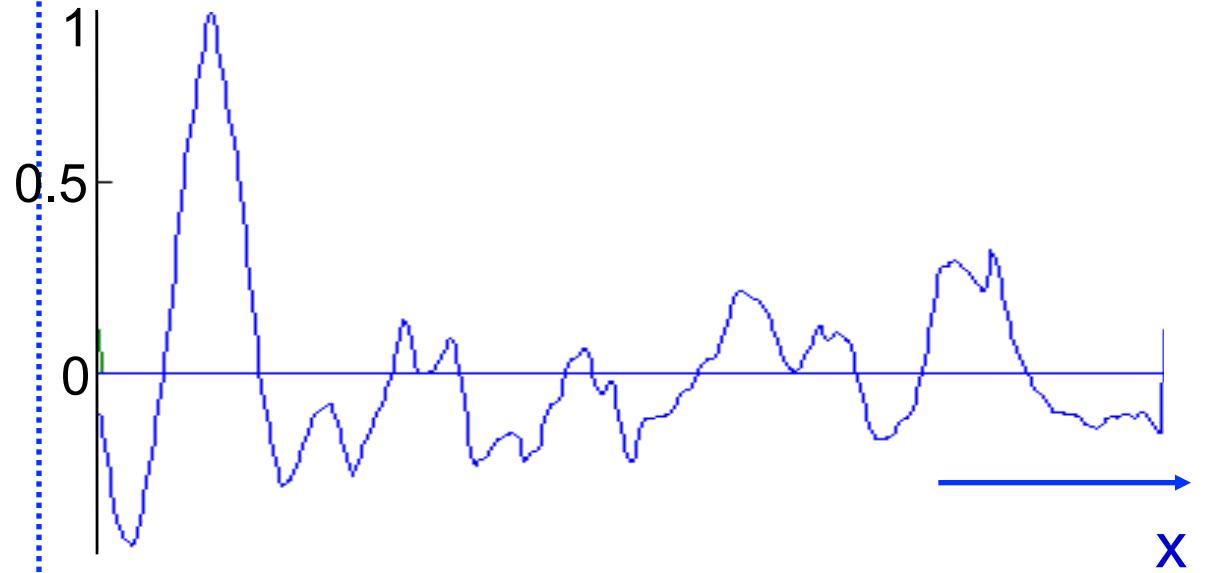
$$\text{cross correlation} = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| |\mathbf{b}|}$$

Invariant to  $I \rightarrow \alpha I + \beta$   
(exercise)

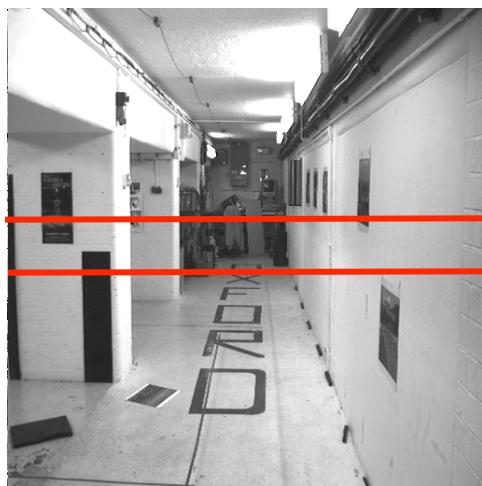


left image band

right image band



cross  
correlation



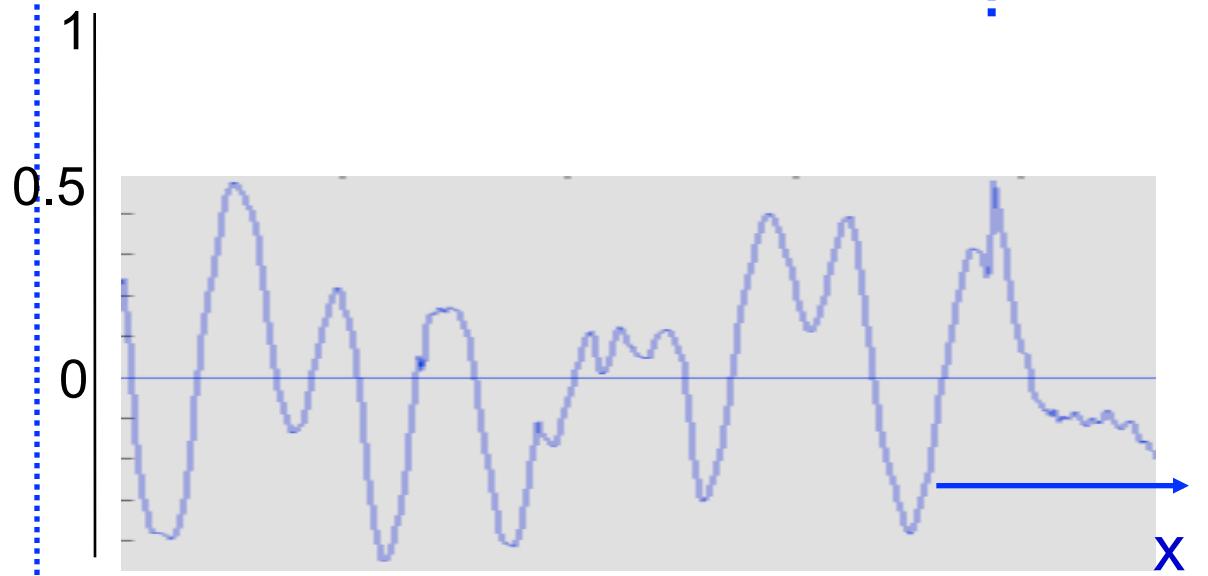
target region



left image band



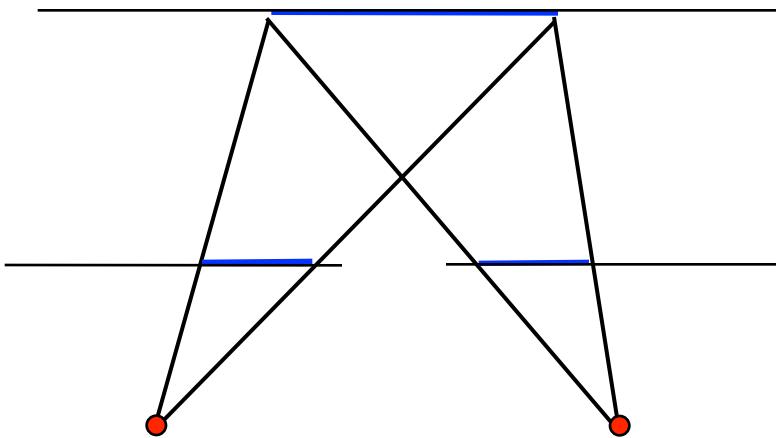
right image band



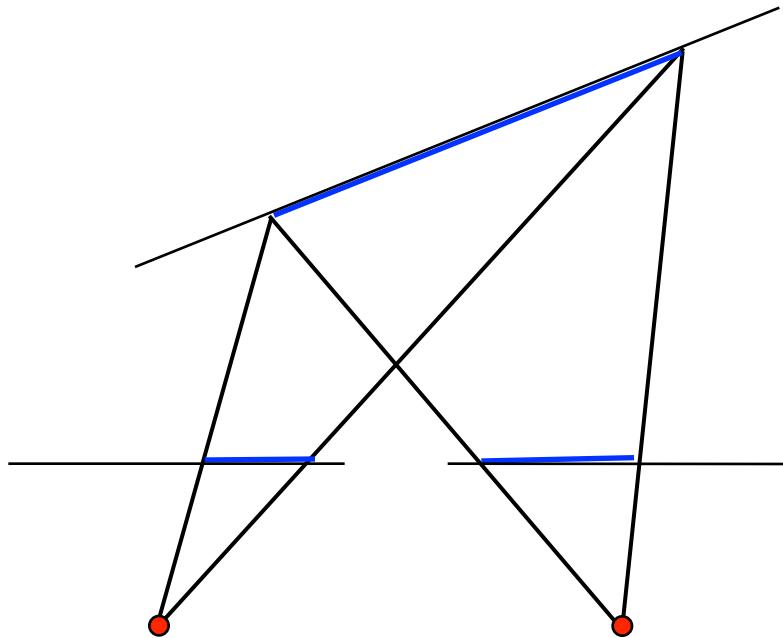
cross  
correlation

## 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



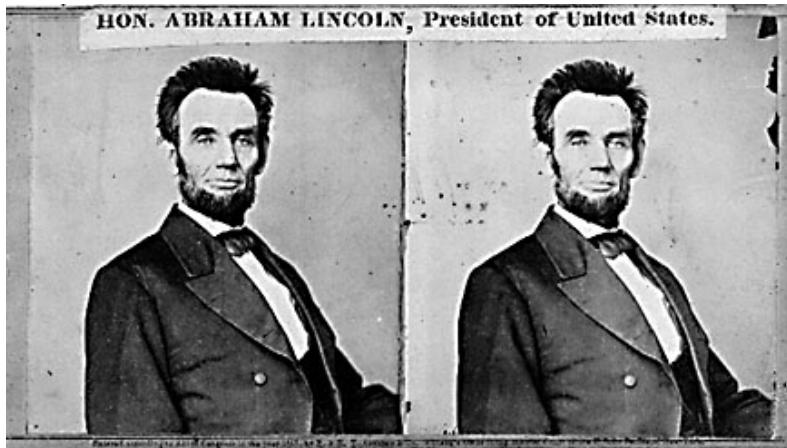
fronto-parallel surface  
imaged length the same



slanting surface  
imaged lengths differ

# Limitations of similarity constraint

---



Textureless surfaces



Occlusions, repetition



Non-Lambertian surfaces, specularities

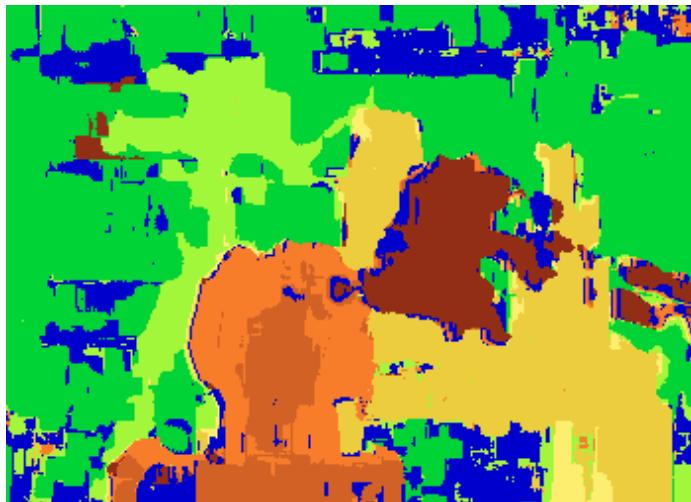
# Results with window search

---

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

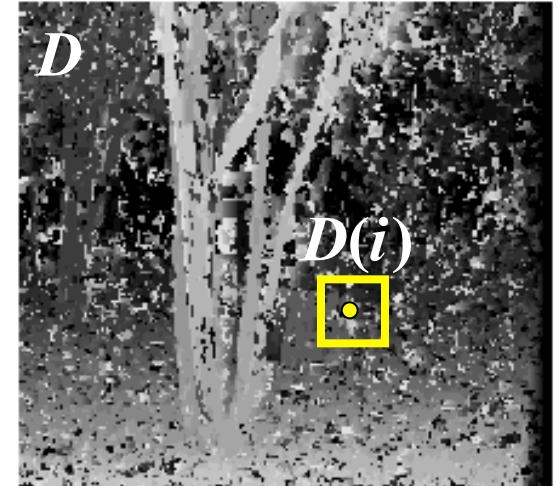
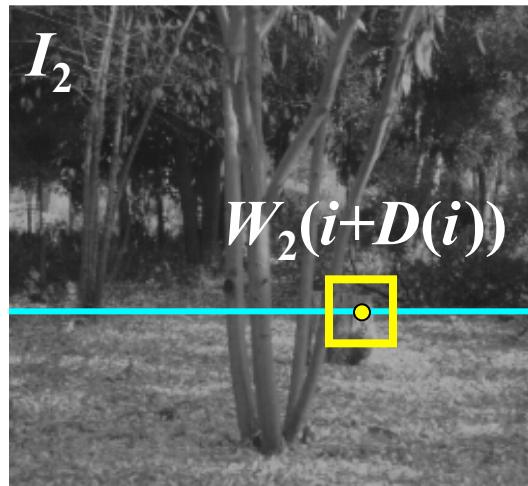
## Other constraints

- uniqueness
- ordering
- smoothness of disparity field

## Applicability

- textured scene, largely fronto-parallel

# Stereo matching as energy minimization



MAP estimate of disparity image  $D$ :  $P(D | I_1, I_2) \propto P(I_1, I_2 | D)P(D)$

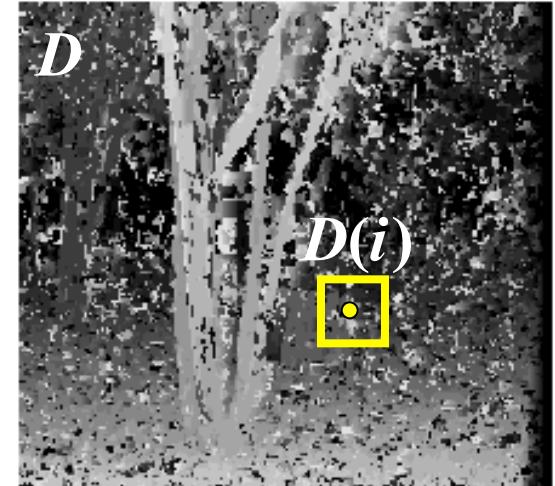
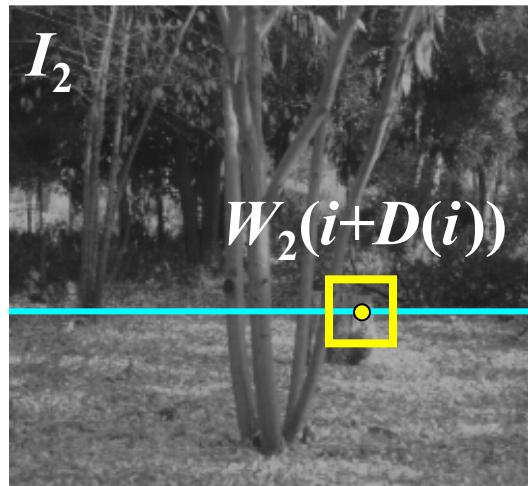
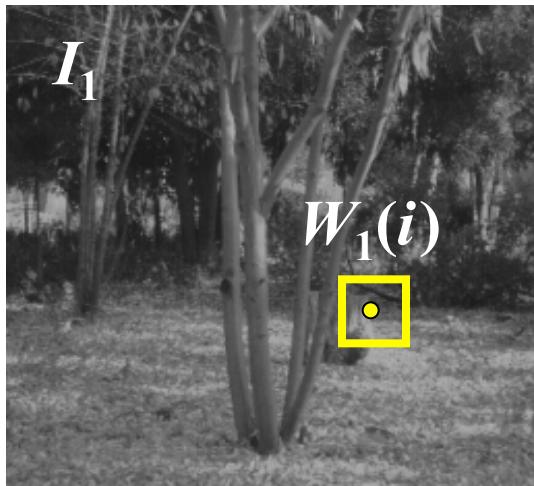
$$-\log P(D | I_1, I_2) \propto -\log P(I_1, I_2 | D) - \log P(D)$$

$$E = \alpha E_{\text{data}}(I_1, I_2, D) + \beta E_{\text{smooth}}(D)$$

$$E_{\text{data}} = \sum_i (W_1(i) - W_2(i + D(i)))^2$$

$$E_{\text{smooth}} = \sum_{\text{neighbors } i, j} \rho(D(i) - D(j))$$

# Stereo matching as energy minimization



$$E = \alpha E_{\text{data}}(I_1, I_2, D) + \beta E_{\text{smooth}}(D)$$

$$E_{\text{data}} = \sum_i (W_1(i) - W_2(i + D(i)))^2$$

$$E_{\text{smooth}} = \sum_{\text{neighbors } i, j} \rho(D(i) - D(j))$$

- Energy functions of this form can be minimized using *graph cuts*

[Y. Boykov, O. Veksler, and R. Zabih,  
Fast Approximate Energy Minimization via Graph Cuts, PAMI 2001](#)

# Graph cuts solution

---



Graph cuts



Ground truth

Y. Boykov, O. Veksler, and R. Zabih,

Fast Approximate Energy Minimization via Graph Cuts, PAMI 2001

For the latest and greatest: <http://www.middlebury.edu/stereo/>

## Example dense correspondence algorithm

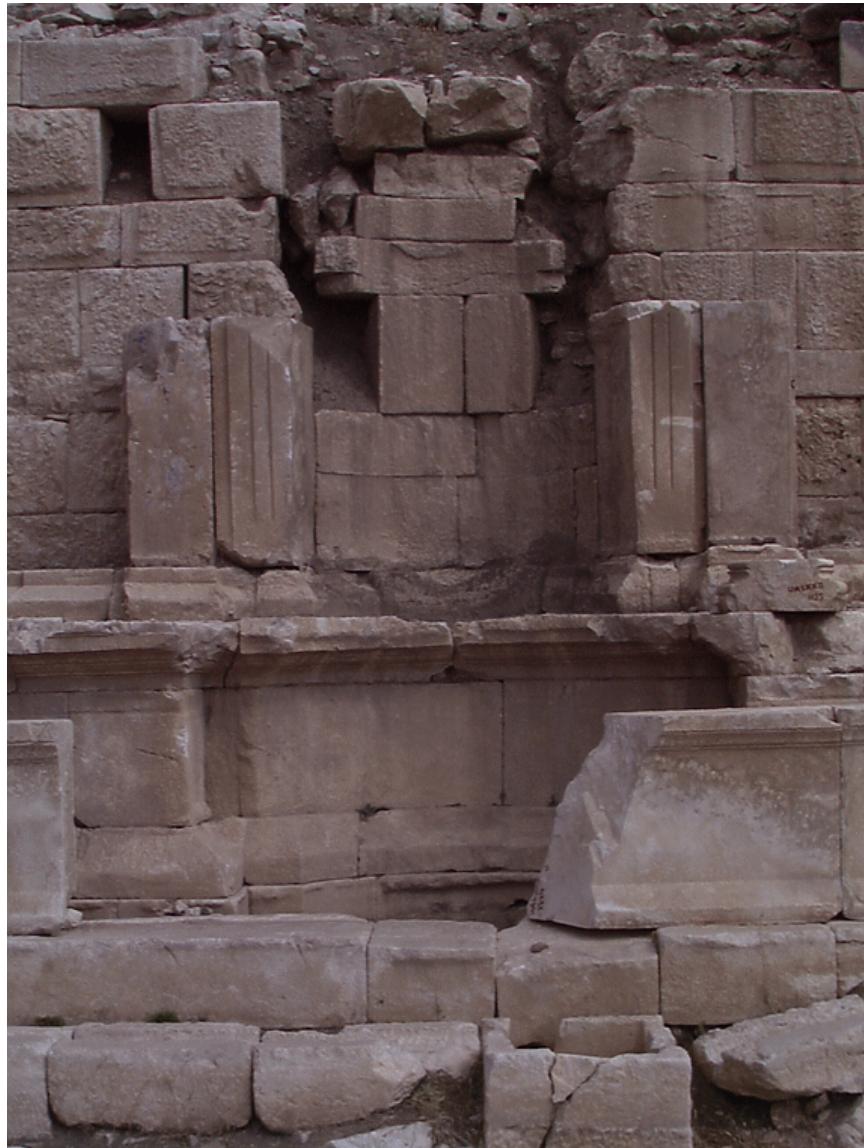


left image



right image

## 3D reconstruction

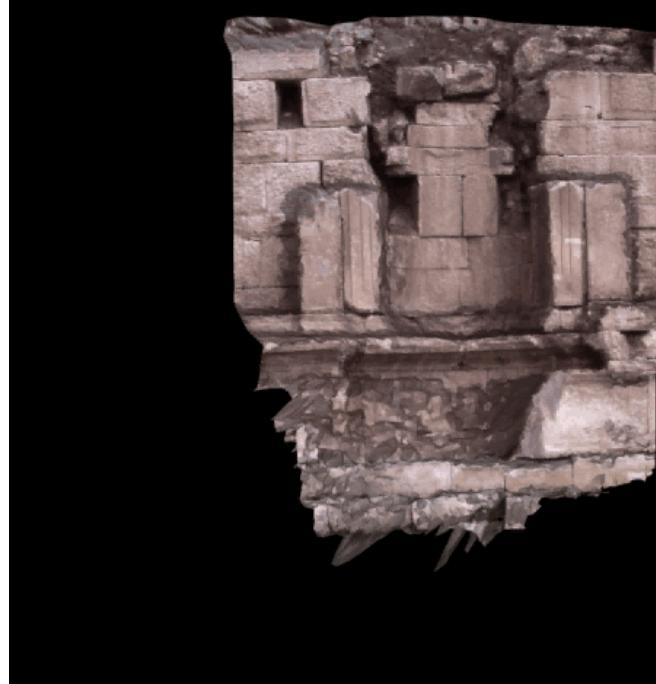


right image



depth map  
intensity = depth

# Texture mapped 3D triangulation



# Pentagon example

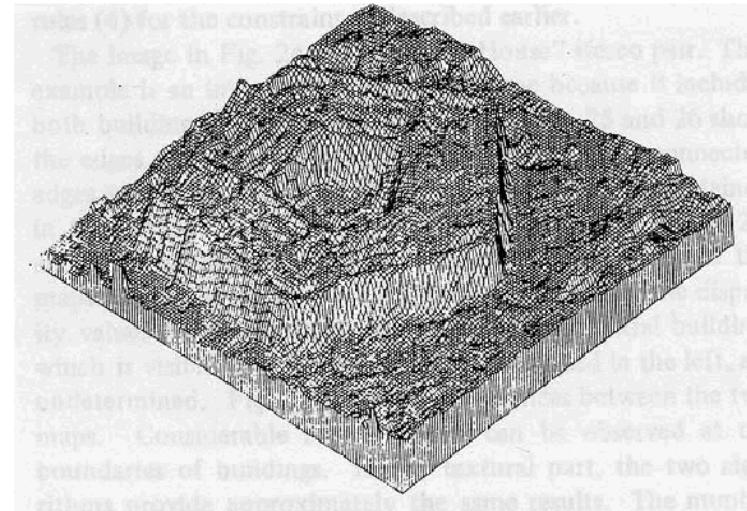
left image



right image



range map

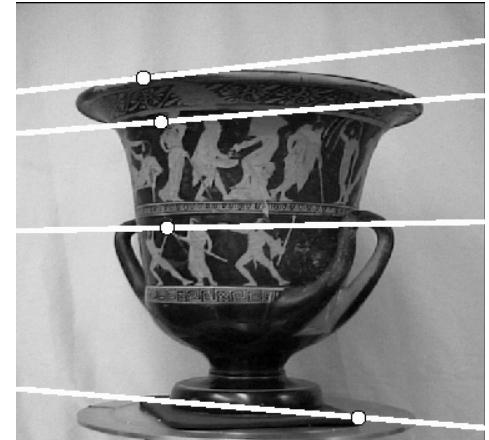
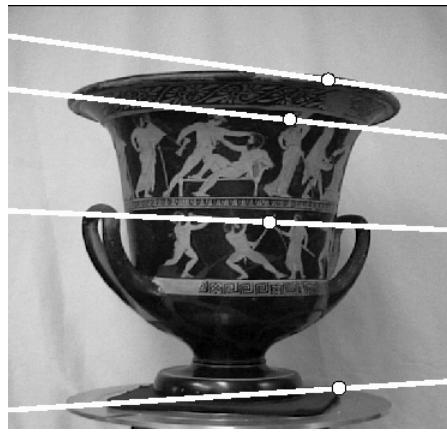
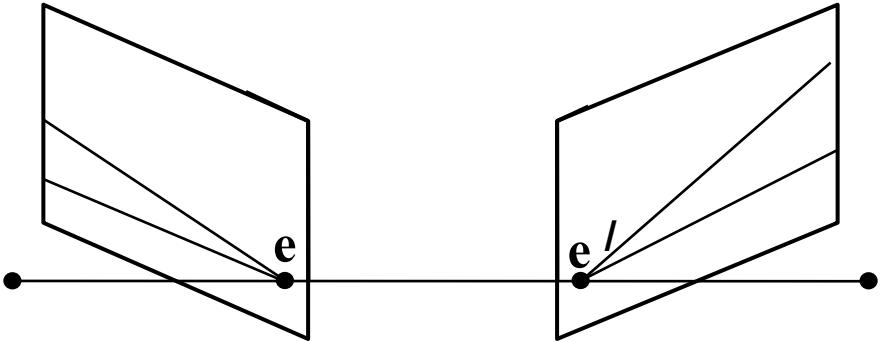


# Rectification

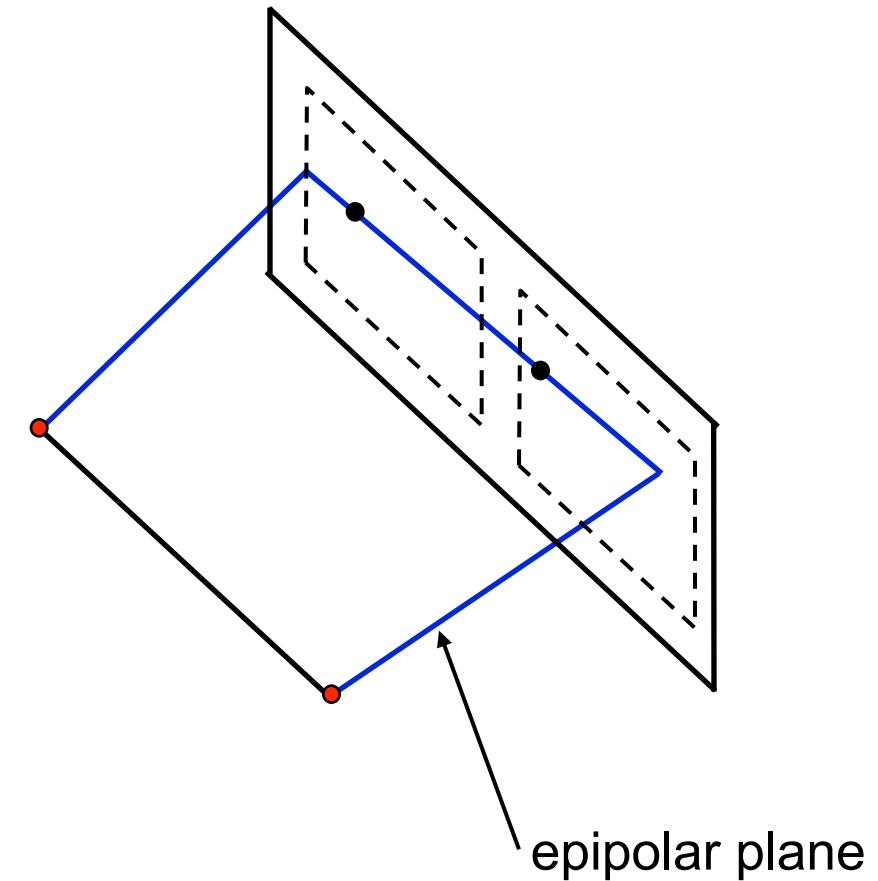
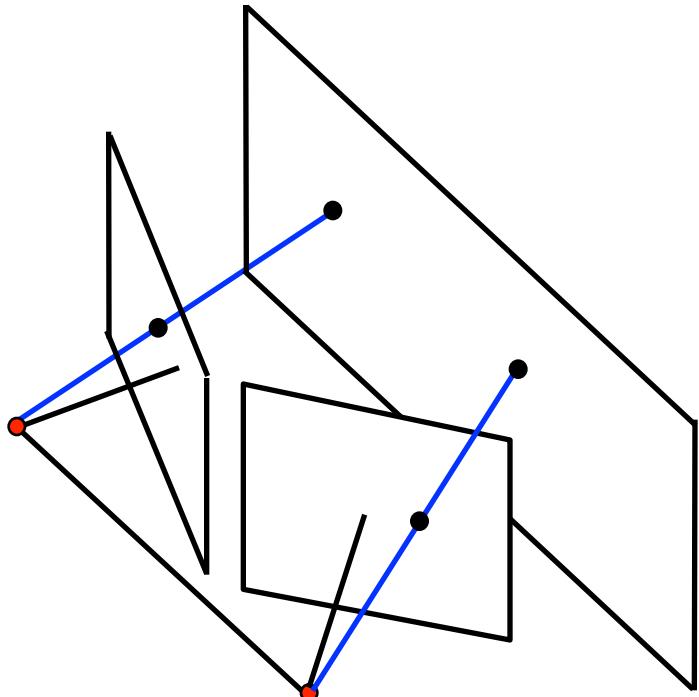
---

For converging cameras

- epipolar lines are not parallel



# Project images onto plane parallel to baseline



# Rectification continued

---

Convert converging cameras to parallel camera geometry by an image mapping

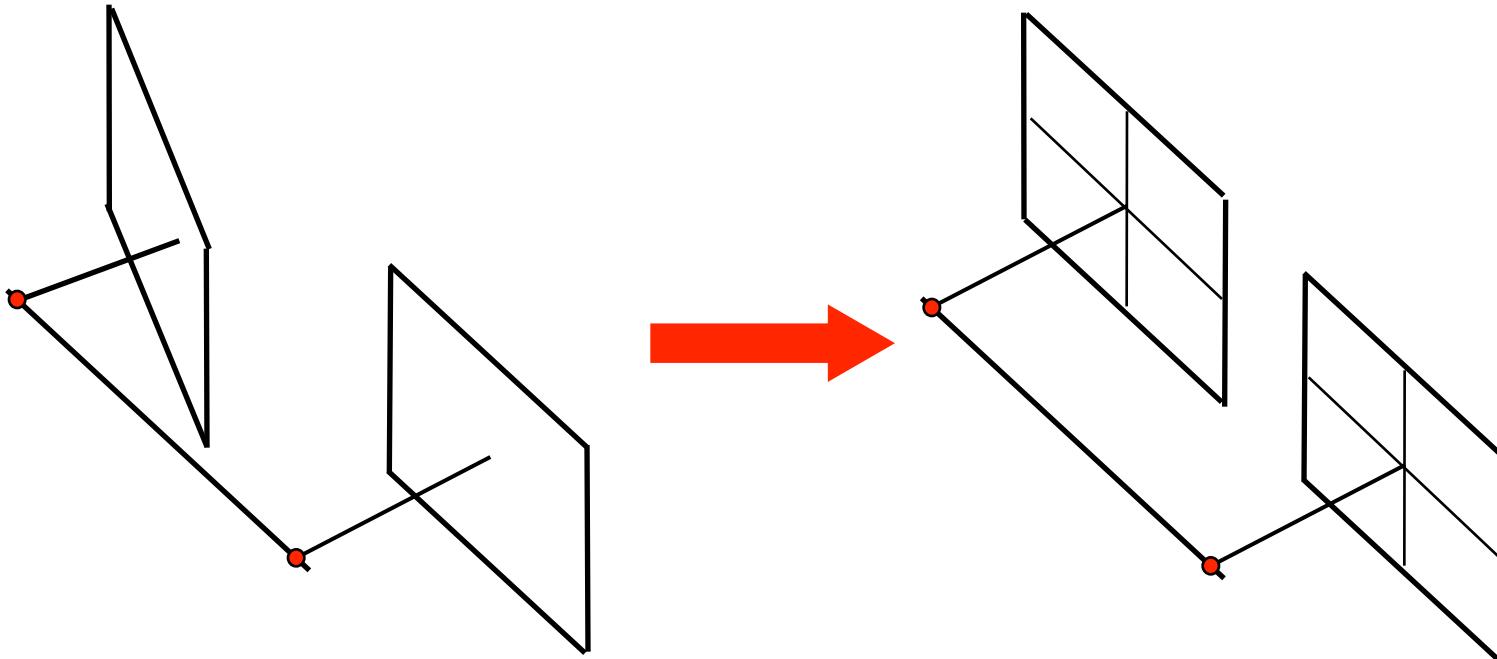


Image mapping is a 2D homography (projective transformation)

$$H = KRK^{-1} \quad (\text{exercise})$$

# Rectification continued

---

Convert converging cameras to parallel camera geometry by an image mapping

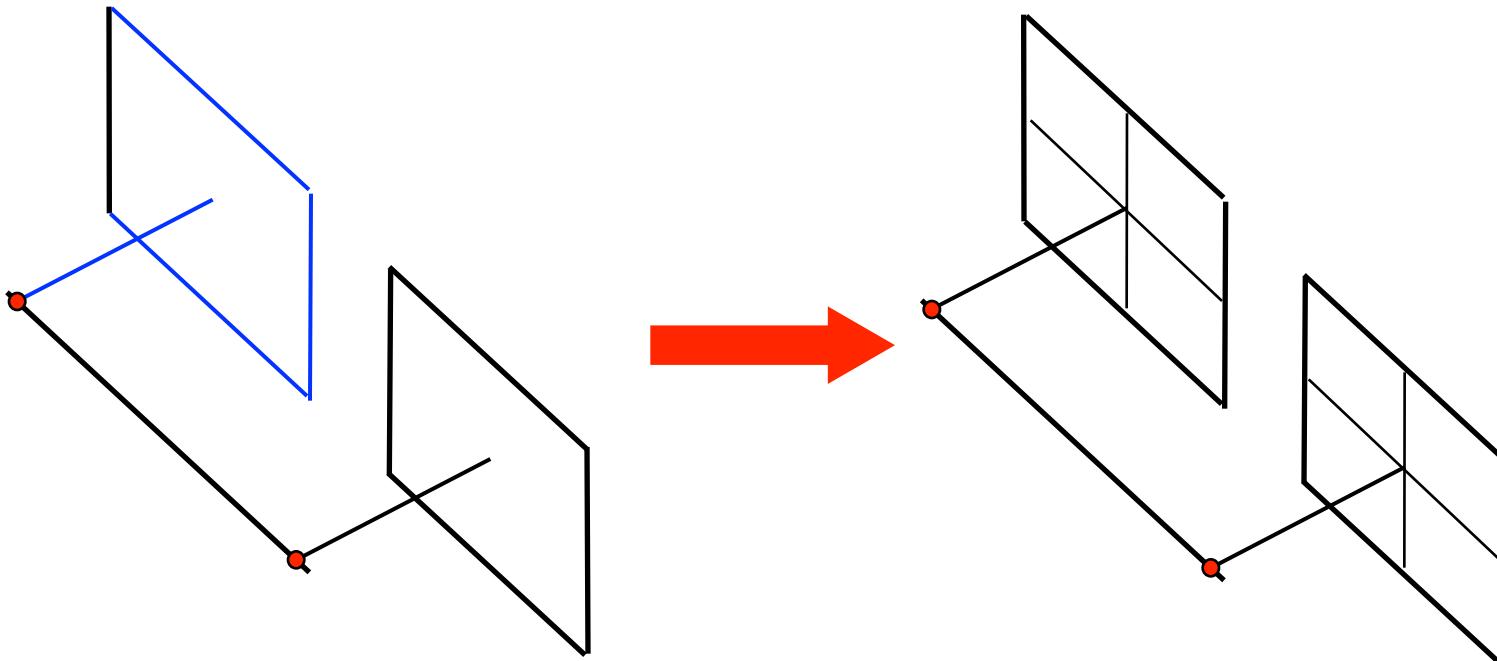
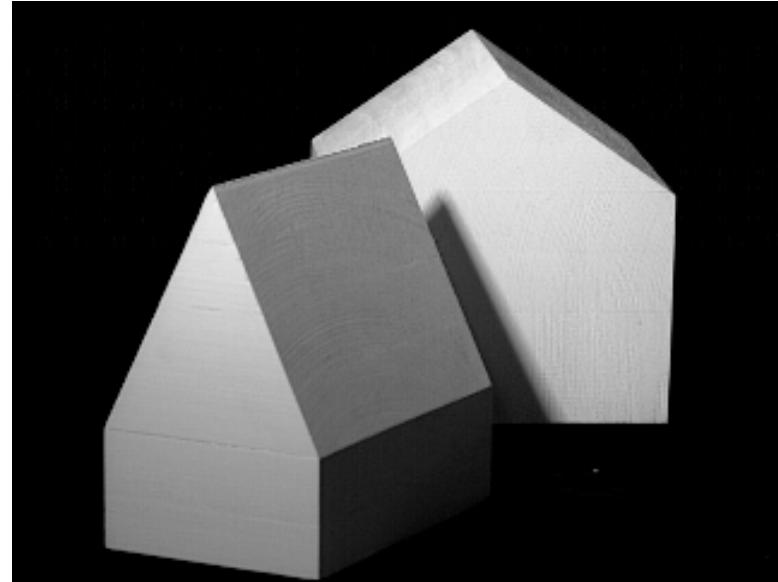
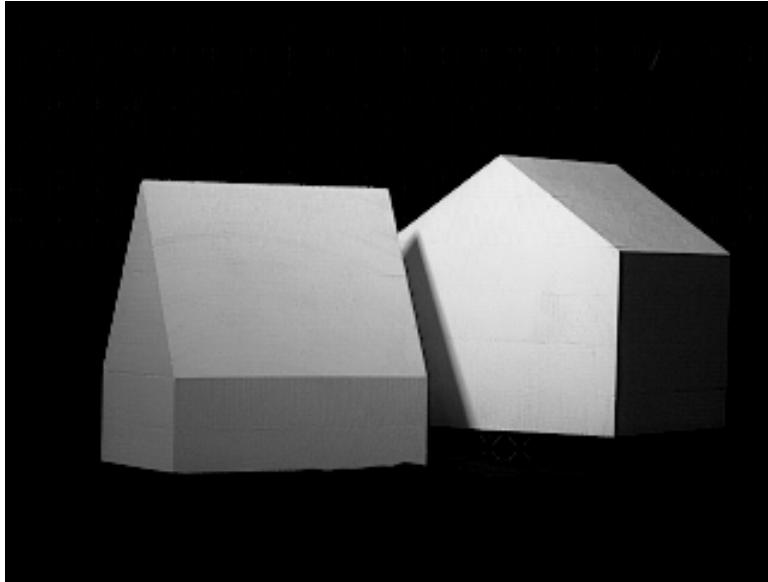


Image mapping is a 2D homography (projective transformation)

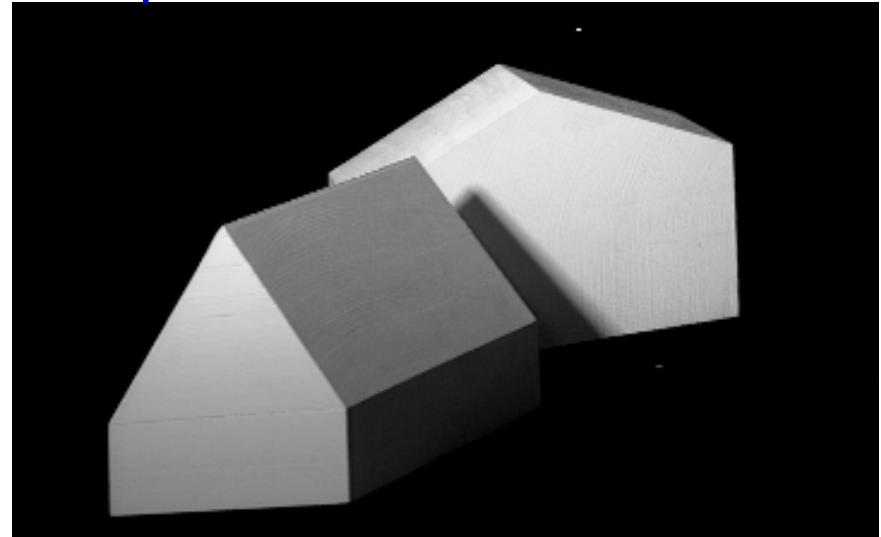
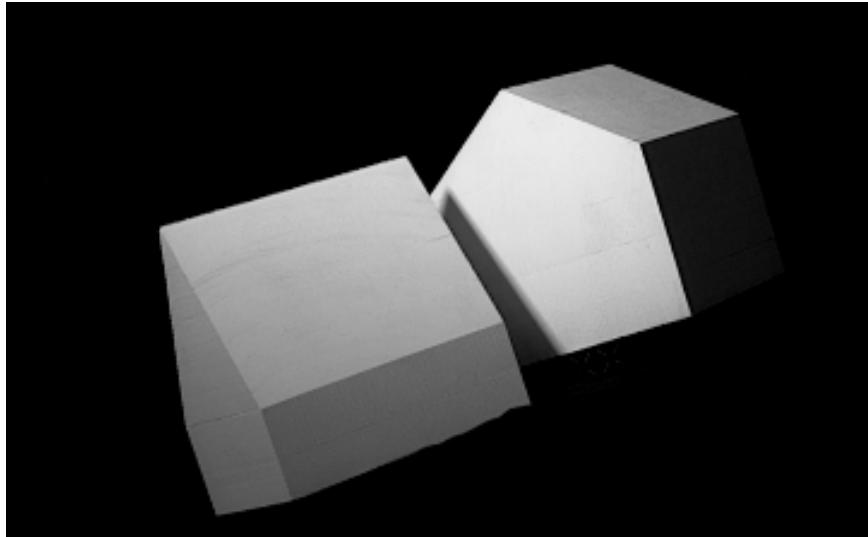
$$H = KRK^{-1} \quad (\text{exercise})$$

## Example

original stereo pair



rectified stereo pair



## Example: depth and disparity for a parallel camera stereo rig

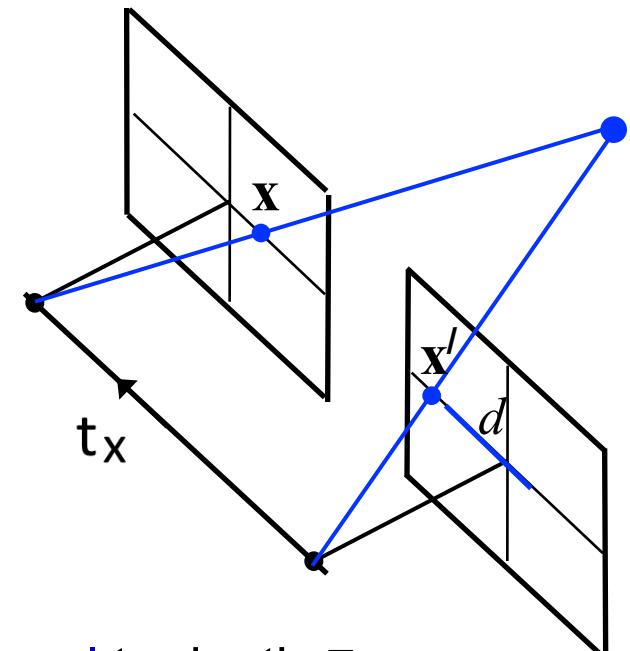
$$K = K' = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R = I \quad t = \begin{pmatrix} t_x \\ 0 \\ 0 \end{pmatrix}$$

Then,  $y' = y$ , and the **disparity**  $d = x' - x = \frac{ft_x}{Z}$

### Derivation

$$\frac{x}{f} = \frac{X}{Z} \quad \frac{x'}{f} = \frac{X + t_x}{Z}$$

$$\frac{x'}{f} = \frac{x}{f} + \frac{t_x}{Z}$$



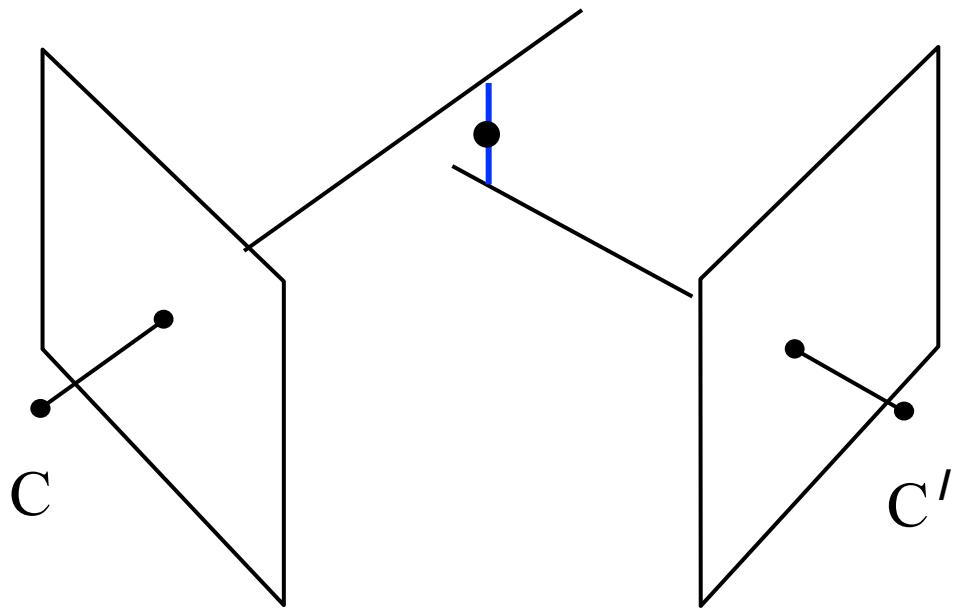
### Note

- image movement (disparity) is **inversely proportional** to depth  $Z$   
as  $z \rightarrow \infty$ ,  $d \rightarrow 0$
- depth is inversely proportional to disparity

# Triangulation

# 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  $\mathbf{x} \times (\mathbf{P}\mathbf{X}) = \mathbf{0}$

$$x(\mathbf{p}^{3\top}\mathbf{X}) - (\mathbf{p}^{1\top}\mathbf{X}) = 0$$

$$y(\mathbf{p}^{3\top}\mathbf{X}) - (\mathbf{p}^{2\top}\mathbf{X}) = 0$$

$$x(\mathbf{p}^{2\top}\mathbf{X}) - y(\mathbf{p}^{1\top}\mathbf{X}) = 0$$

- rearrange as (first two equations only)

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

Similarly for the second camera:

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

Collecting together gives

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

where  $\mathbf{A}$  is the  $4 \times 4$  matrix

$$\mathbf{A} = \begin{bmatrix} x\mathbf{p}^{\top} - \mathbf{p}^{\top} \\ y\mathbf{p}^{\top} - \mathbf{p}^{\top} \\ x' \mathbf{p}'^{\top} - \mathbf{p}'^{\top} \\ y' \mathbf{p}'^{\top} - \mathbf{p}'^{\top} \end{bmatrix}$$

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

**Problem:** does not minimize anything meaningful

**Advantage:** extends to more than two views

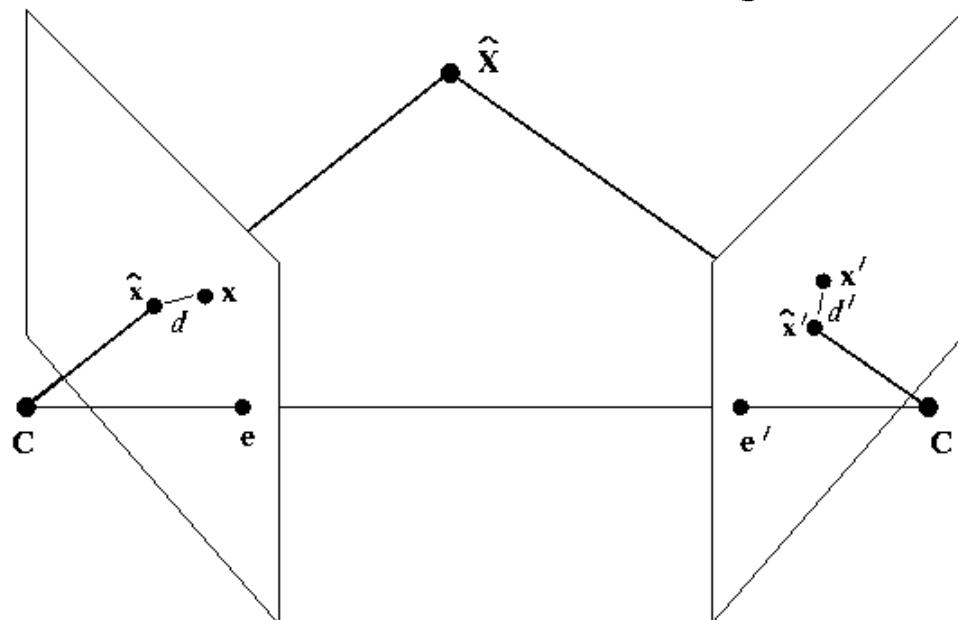
### 3. Minimizing a geometric/statistical error

---

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

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

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



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

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 X
- The minimization appears to be over three parameters (the position X), but the problem can be reduced to a minimization over one parameter

# Different formulation of the problem

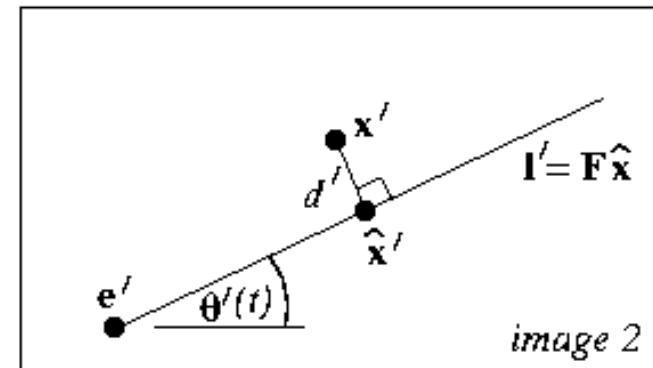
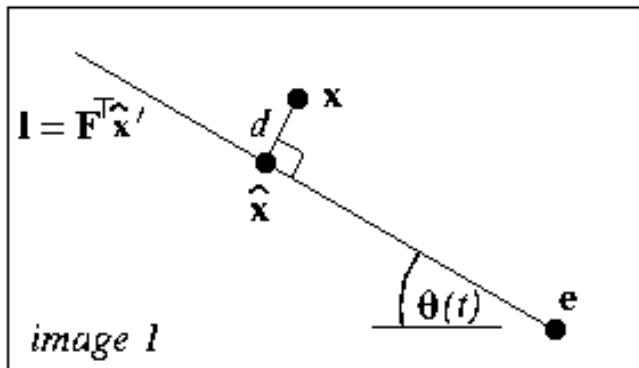
---

The minimization problem may be formulated differently:

- Minimize

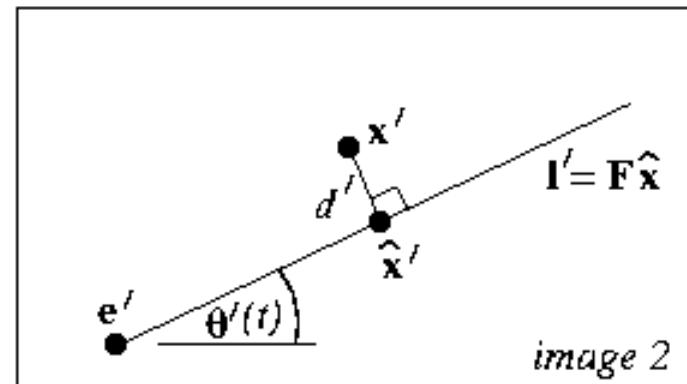
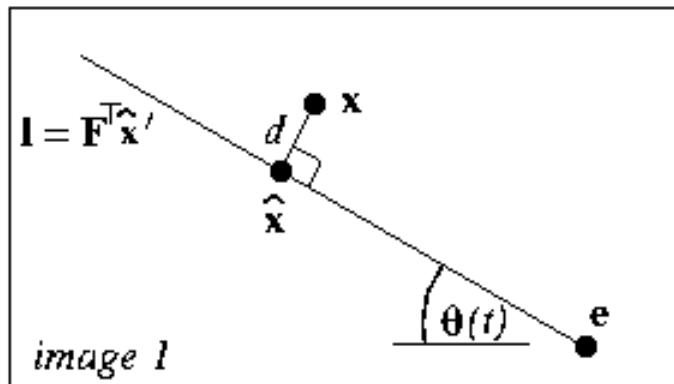
$$d(\mathbf{x}, \mathbf{l})^2 + d(\mathbf{x}', \mathbf{l}')^2$$

- $\mathbf{l}$  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  $\hat{\mathbf{x}}'$ .



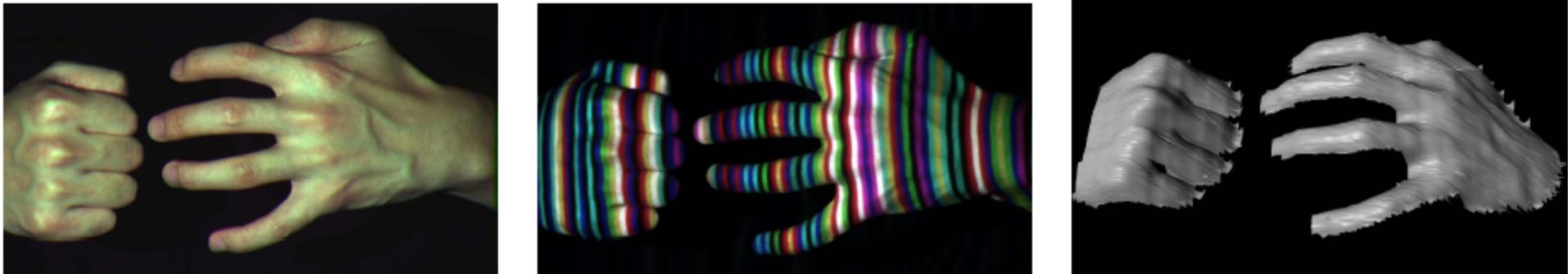
## Minimization method

- Parametrize the pencil of epipolar lines in the first image by  $t$ , such that the epipolar line is  $\mathbf{l}(t)$
- Using  $\mathbf{F}$  compute the corresponding epipolar line in the second image  $\mathbf{l}'(t)$
- Express the distance function  $d(\mathbf{x}, \mathbf{l})^2 + d(\mathbf{x}', \mathbf{l}')^2$  explicitly as a function of  $t$
- Find the value of  $t$  that minimizes the distance function
- Solution is a 6<sup>th</sup> degree polynomial in  $t$

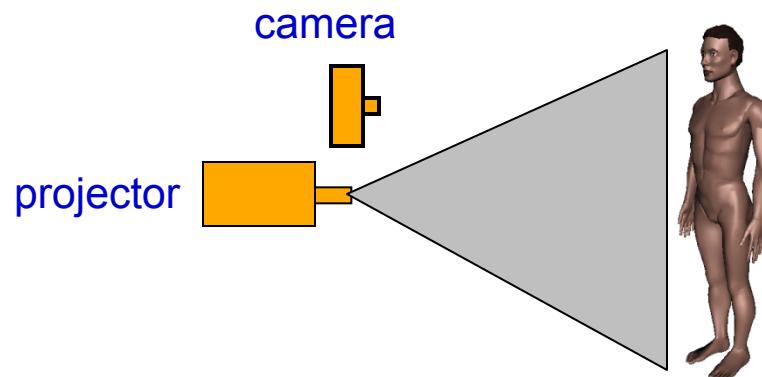


Other approaches  
to obtaining 3D  
structure

# Active stereo with structured light



- Project “structured” light patterns onto the object
  - simplifies the correspondence problem
  - Allows us to use only one camera

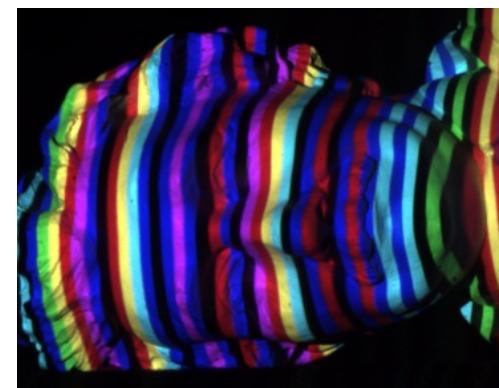
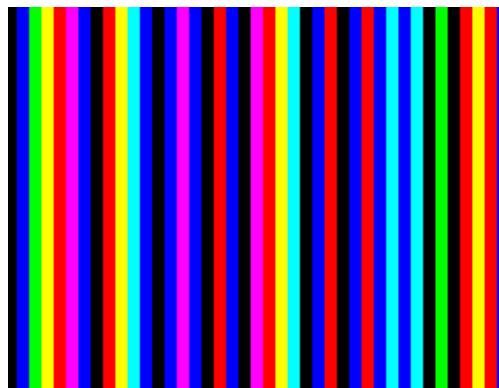
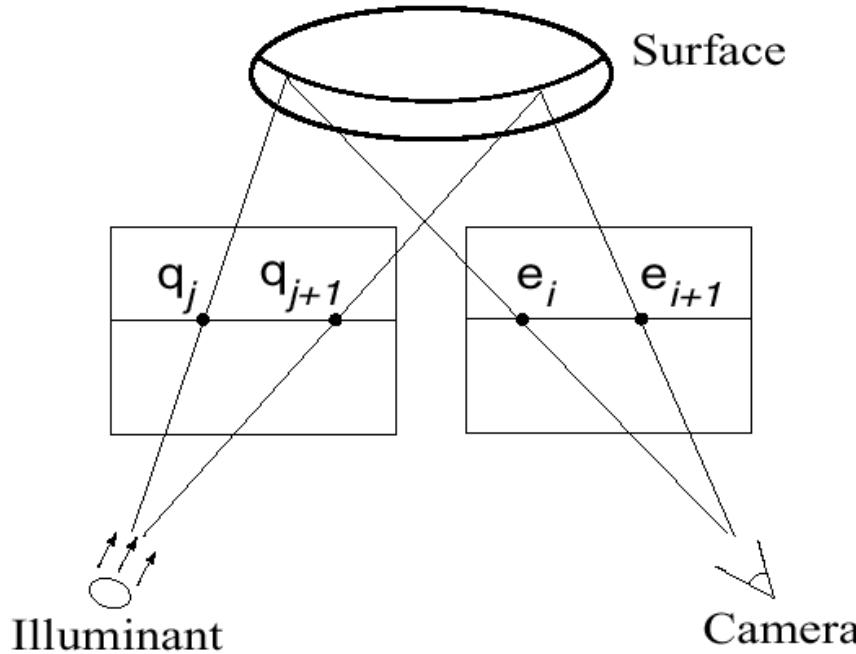


L. Zhang, B. Curless, and S. M. Seitz.

Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. 3DPVT 2002

# Active stereo with structured light

---

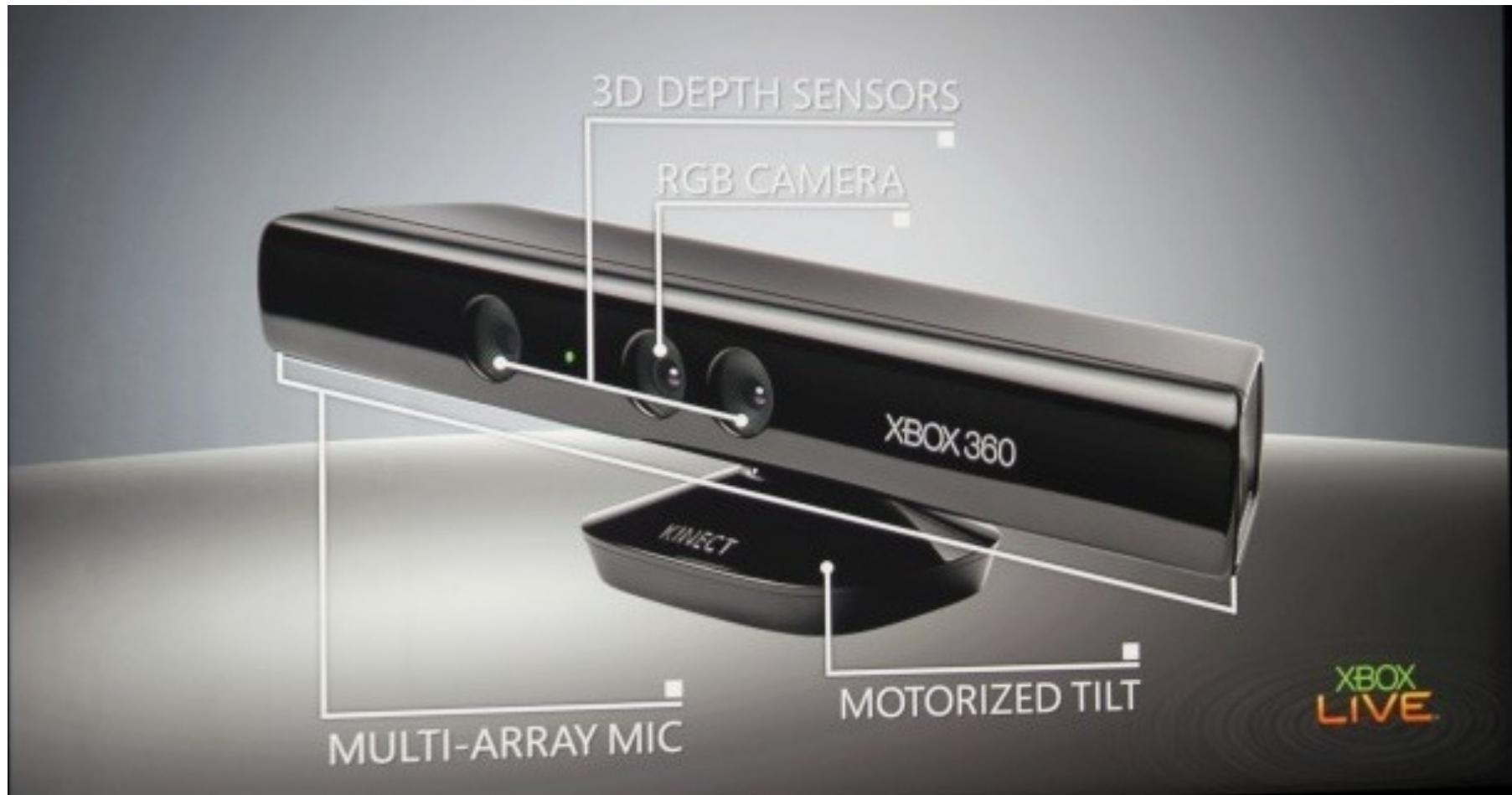


L. Zhang, B. Curless, and S. M. Seitz.

Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. 3DPVT 2002

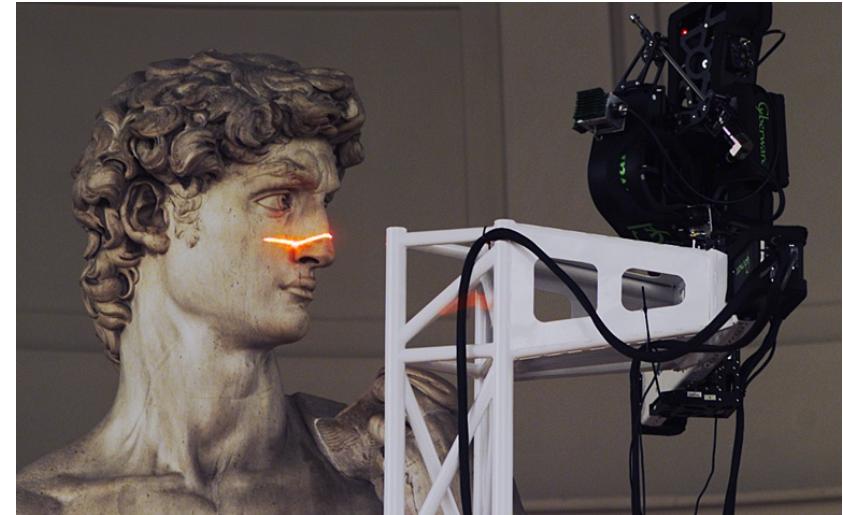
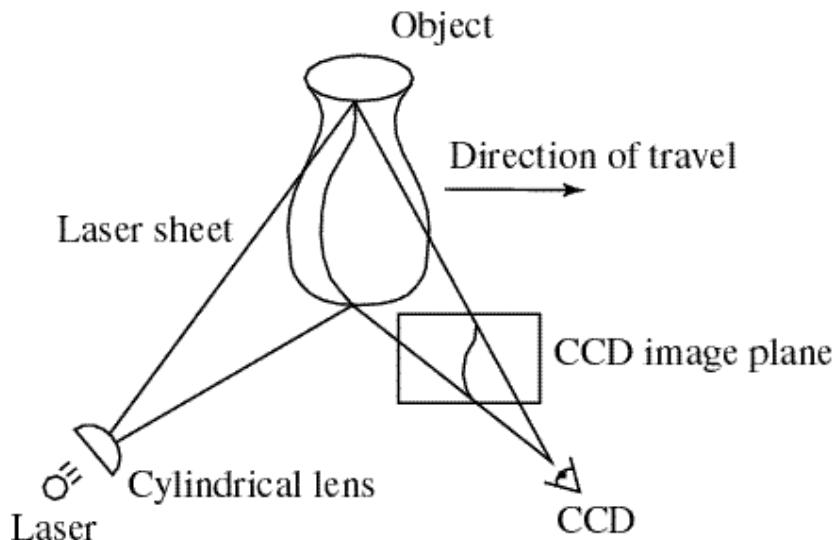
# Microsoft Kinect

---



# Laser scanning

---



Digital Michelangelo Project  
<http://graphics.stanford.edu/projects/mich/>

- Optical triangulation
  - Project a single stripe of laser light
  - Scan it across the surface of the object
  - This is a very precise version of structured light scanning

# Laser scanned models

---



*The Digital Michelangelo Project, Levoy et al.*

Source: S. Seitz

# Laser scanned models

---

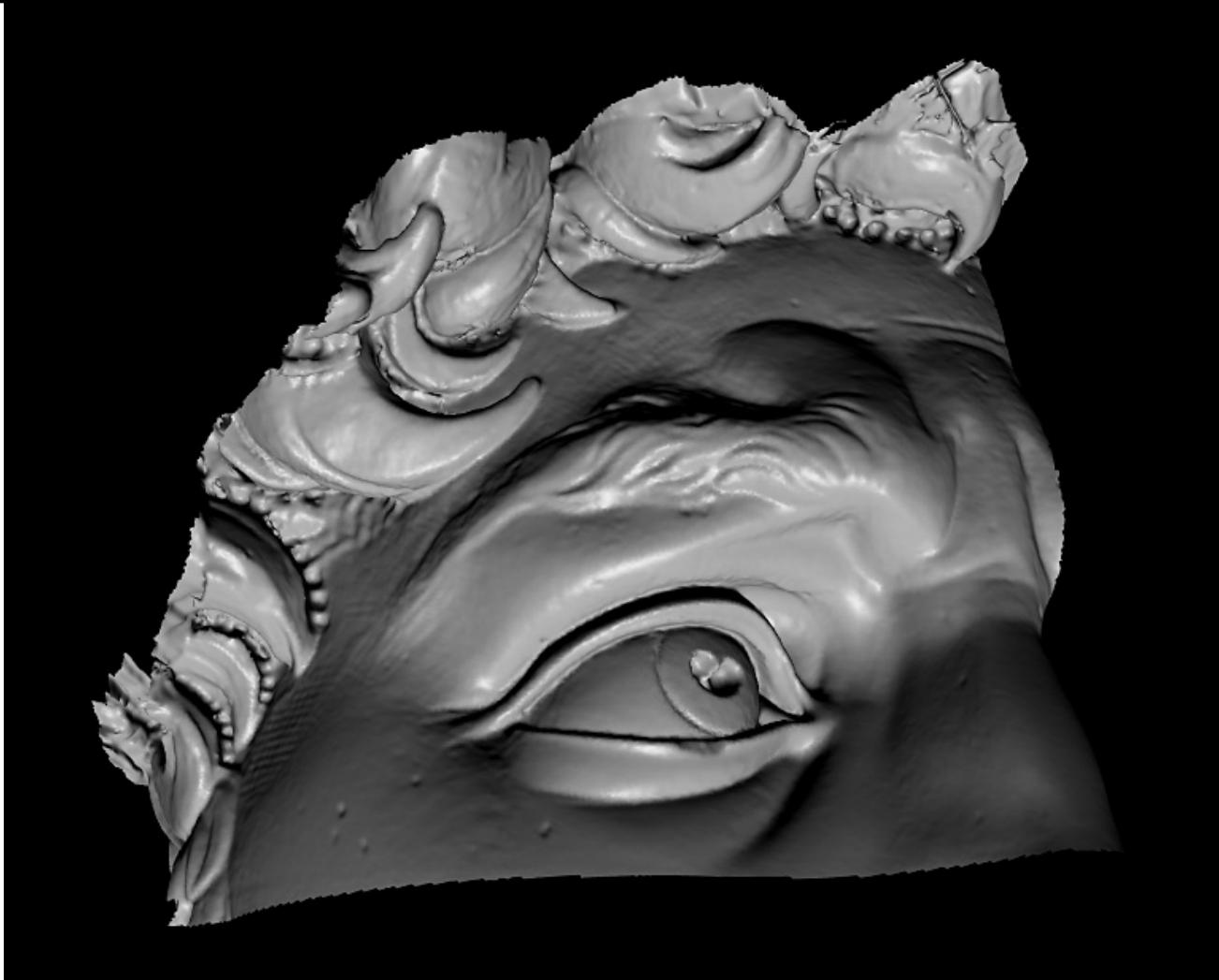


*The Digital Michelangelo Project, Levoy et al.*

Source: S. Seitz

# Laser scanned models

---

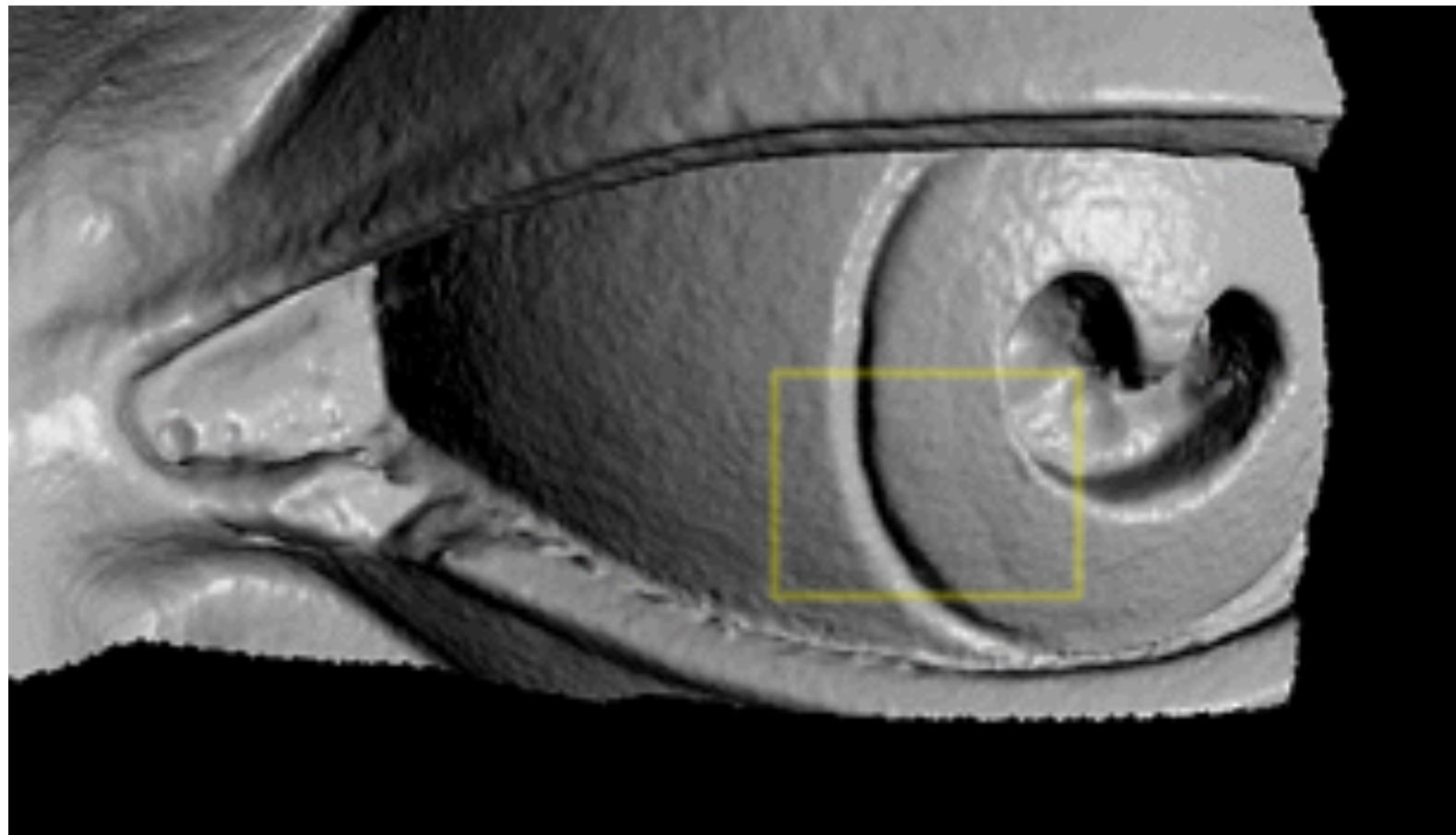


*The Digital Michelangelo Project, Levoy et al.*

Source: S. Seitz

# Laser scanned models

---

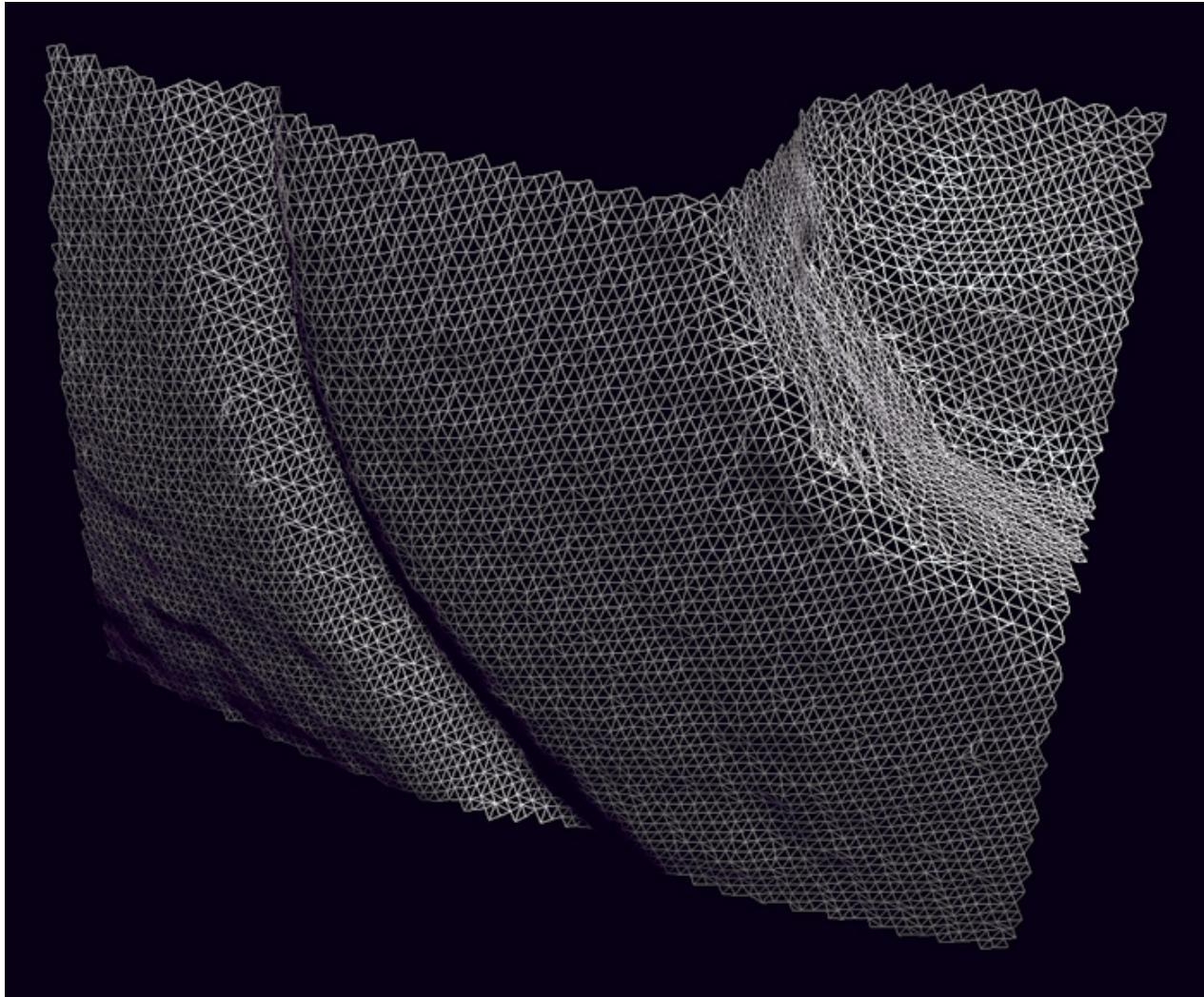


*The Digital Michelangelo Project, Levoy et al.*

Source: S. Seitz

# Laser scanned models

---



*The Digital Michelangelo Project, Levoy et al.*

Source: S. Seitz

# Aligning range images

---

- A single range scan is not sufficient to describe a complex surface
- Multiple range images can be aligned to build a model



B. Curless and M. Levoy,  
A Volumetric Method for Building Complex Models from Range Images, SIGGRAPH  
1996

# Aligning range images

---

- A single range scan is not sufficient to describe a complex surface
- Need techniques to register multiple range images
  - ... which brings us to *multi-view stereo*