Two View Geometry

CS 600.361/600.461

Instructor: Nicolas Padoy

http://www.cs.jhu.edu/~padoy/cv2011

(Adapted from slides by Kristen Grauman and Fei-Fei Li)

Outline

- 3D transformations
- Triangulation
- Epipolar geometry
- Rectification

Reminders

Central Projection

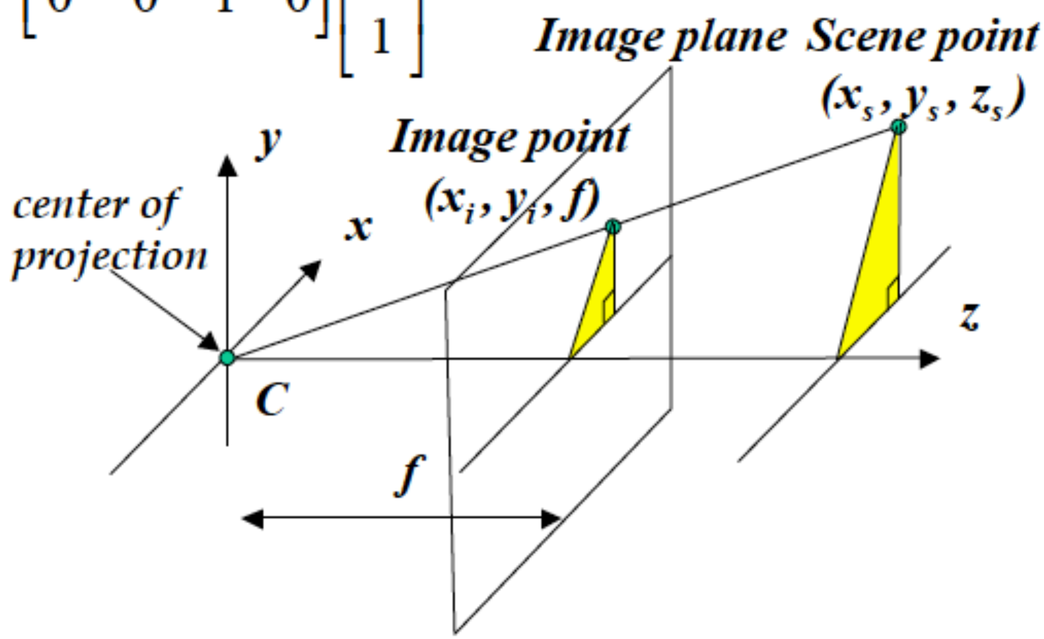
If world and image points are represented by homogeneous vectors, central projection is a linear transformation:

$$x_i = f\frac{x_s}{z_s}$$

$$y_i = f\frac{y_s}{z_s}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix}$$

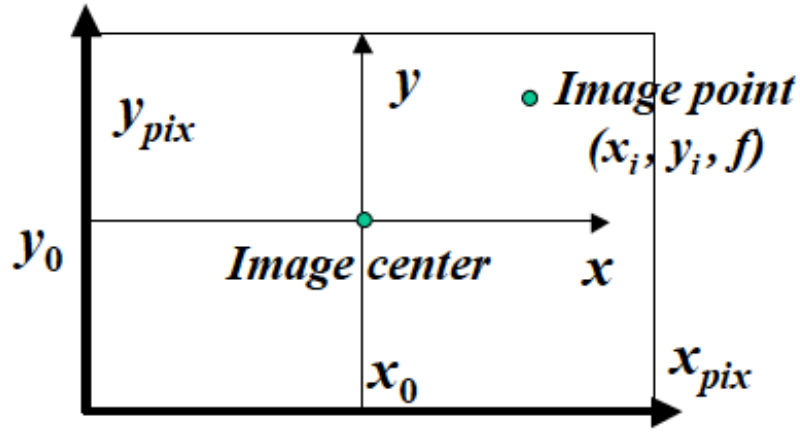
$$x_i = u/w, \quad y_i = v/w$$



Transformation From Lengths to Pixels

Transformation uses:

- image center (x_0, y_0)
- scaling factors k_x and k_y

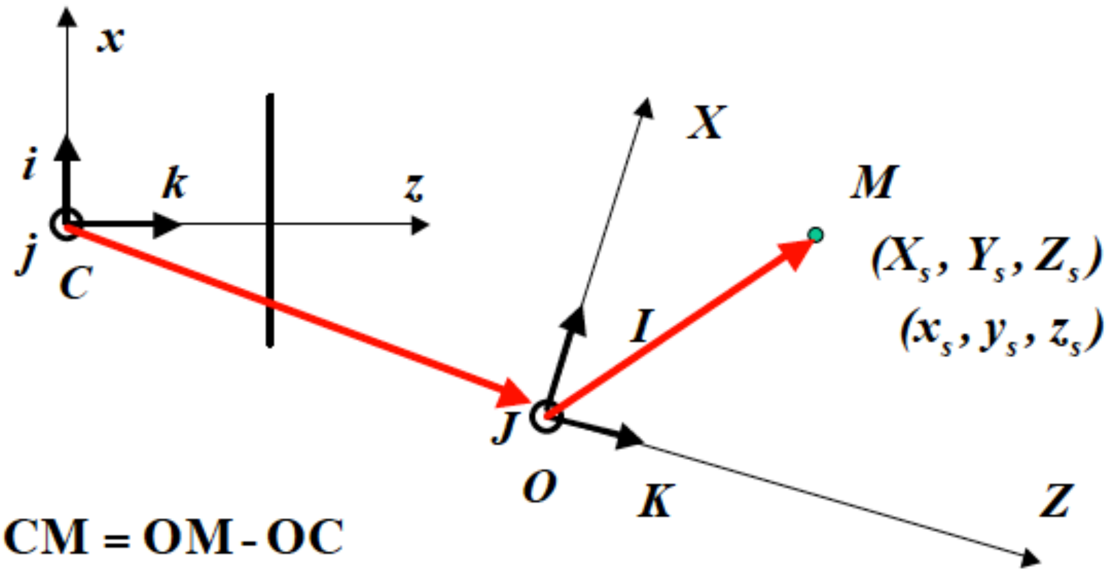


$$x_i = f\,\frac{x_s}{z_s} \qquad x_{pix} = k_x x_i + x_0 = f\,k_x\,\frac{x_s + z_s x_0}{z_s}$$

$$y_i = f\,\frac{y_s}{z_s} \qquad y_{pix} = k_y y_i + y_0 = f\,k_y\,\frac{y_s + z_s y_0}{z_s}$$

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} \alpha_x & 0 & x_0 & 0 \\ 0 & \alpha_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} \quad \text{with} \quad \begin{matrix} \alpha_x = f\,k_x \\ \alpha_y = f\,k_y \end{matrix} \quad \text{then} \quad \begin{matrix} x_{pix} = u'/w' \\ y_{pix} = v'/w' \end{matrix}$$

From Camera Coordinates to World Coordinates 3



$$\mathbf{CM} = \mathbf{OM} - \mathbf{OC}$$

$$x_S\mathbf{i} + y_S\mathbf{j} + z_S\mathbf{k} = (X_S - X_C)\mathbf{I} + (Y_S - Y_C)\mathbf{J} + (Z_S - Z_C)\mathbf{K}$$

$$x_S = (X_S - X_C)\mathbf{I.i} + (Y_S - Y_C)\mathbf{J.i} + (Z_S - Z_C)\mathbf{K.i}$$

$$\mathbf{x}_{\mathbf{cam}} = \mathbf{R}(\mathbf{X} - \tilde{\mathbf{C}}) \qquad (\mathbf{T} = -\mathbf{R}\tilde{\mathbf{C}})$$

$\tilde{\mathrm{C}}$ is vector OC expressed in world coordinate system

Properties of Matrix **P**

$$\boxed{\mathbf{P} = \mathbf{K}\,\mathbf{R}\left[\mathbf{I}_3 \quad | \quad -\tilde{\mathbf{C}}\right]}$$

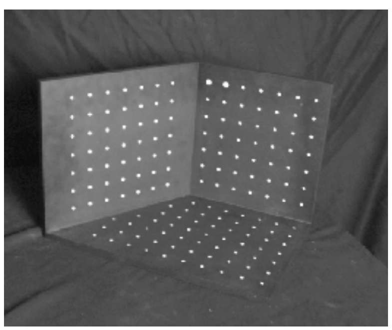
- **P** has 11 degrees of freedom:
  - 5 from triangular calibration matrix **K**, 3 from **R** and 3 from $\tilde{\mathbf{C}}$
- **P** is a fairly general 3 x 4 matrix
  - left 3x3 submatrix **KR** is non-singular

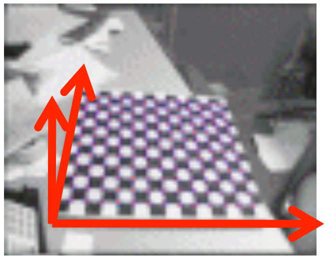
Calibration using a 3D target

- 1. Estimate matrix **P** using scene points and their images
- 2. Estimate the interior parameters and the exterior parameters

$$\boxed{\mathbf{P} = \mathbf{K}\,\mathbf{R}\left[\mathbf{I}_3 \quad | \quad -\tilde{\mathbf{C}}\right]}$$

- Left 3x3 submatrix of **P** is product of upper-triangular matrix and orthogonal matrix





Multi-planar calibration

Equations

$$\begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3\end{bmatrix} = \lambda \mathbf{A} \begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix}$$

yields constraints

$$\mathbf{h}_1^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2 = 0$$
$$\mathbf{h}_1^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_1 = \mathbf{h}_2^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2$$

used to estimate A (matrix of intrinsic parameters).

External parameters obtained for each plane as

$$\mathbf{r}_1 = \lambda \mathbf{A}^{-1} \mathbf{h}_1$$
$$\mathbf{r}_2 = \lambda \mathbf{A}^{-1} \mathbf{h}_2 \qquad \lambda = 1/\|\mathbf{A}^{-1}\mathbf{h}_1\| = 1/\|\mathbf{A}^{-1}\mathbf{h}_2\|$$
$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$
$$\mathbf{t} = \lambda \mathbf{A}^{-1} \mathbf{h}_3$$

3D transformations

3D Transformations

Same idea as 2D transformations

- Homogeneous coordinates: (x,y,z,w)
- 4x4 non-singular transformation matrices

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Basic 3D Transformations

$$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Identity

$$\begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \\ \boldsymbol{z}' \\ \boldsymbol{w} \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_x & 0 & 0 & 0 \\ 0 & \boldsymbol{s}_y & 0 & 0 \\ 0 & 0 & \boldsymbol{s}_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \\ \boldsymbol{w} \end{bmatrix}$$

Scaling

$$\begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \\ \boldsymbol{z}' \\ \boldsymbol{w} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \boldsymbol{t}_x \\ 0 & 1 & 0 & \boldsymbol{t}_y \\ 0 & 0 & 1 & \boldsymbol{t}_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \\ \boldsymbol{w} \end{bmatrix}$$

Translation

$$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Mirror about Y/Z plane

Basic 3D Transformations

Rotation around Z axis: $$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 & 0 \\ \sin\Theta & \cos\Theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Rotation around Y axis: $$\begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \\ \boldsymbol{z}' \\ \boldsymbol{w} \end{bmatrix} = \begin{bmatrix} \cos\Theta & 0 & \sin\Theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\Theta & 0 & \cos\Theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \\ \boldsymbol{w} \end{bmatrix}$$

Rotation around X axis: $$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\Theta & -\sin\Theta & 0 \\ 0 & \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

3D Rotation

- A general 3D rotation R is a 3x3 orthogonal matrix with determinant 1

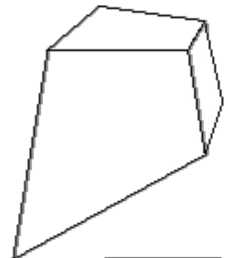
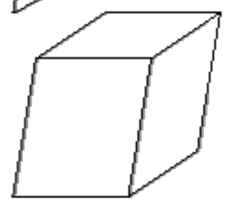
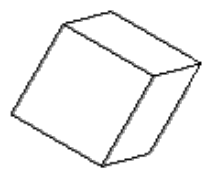
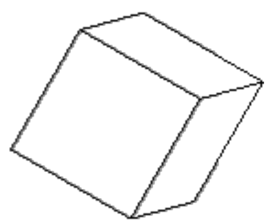
$$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

- 3 degrees of freedom
- Many other possible representations
- Euler angles (successive rotations around each axis)

$$\mathbf{R} = \mathbf{X}_\theta \mathbf{Z}_\psi \mathbf{Y}_\phi$$

- Axis-angle
- Unit quaternions

Hierarchy of (projective) transformations

Projective 15dof	$\begin{bmatrix} A & t \\ v^T & v \end{bmatrix}$		Straight lines, intersection and tangency
Affine 12dof	$\begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix}$		Parallellism of planes and lines, volume ratios, centroids
Similarity 7dof	$\begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix}$		Angles
Euclidean 6dof	$\begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$		Lengths, volume

(A invertible)

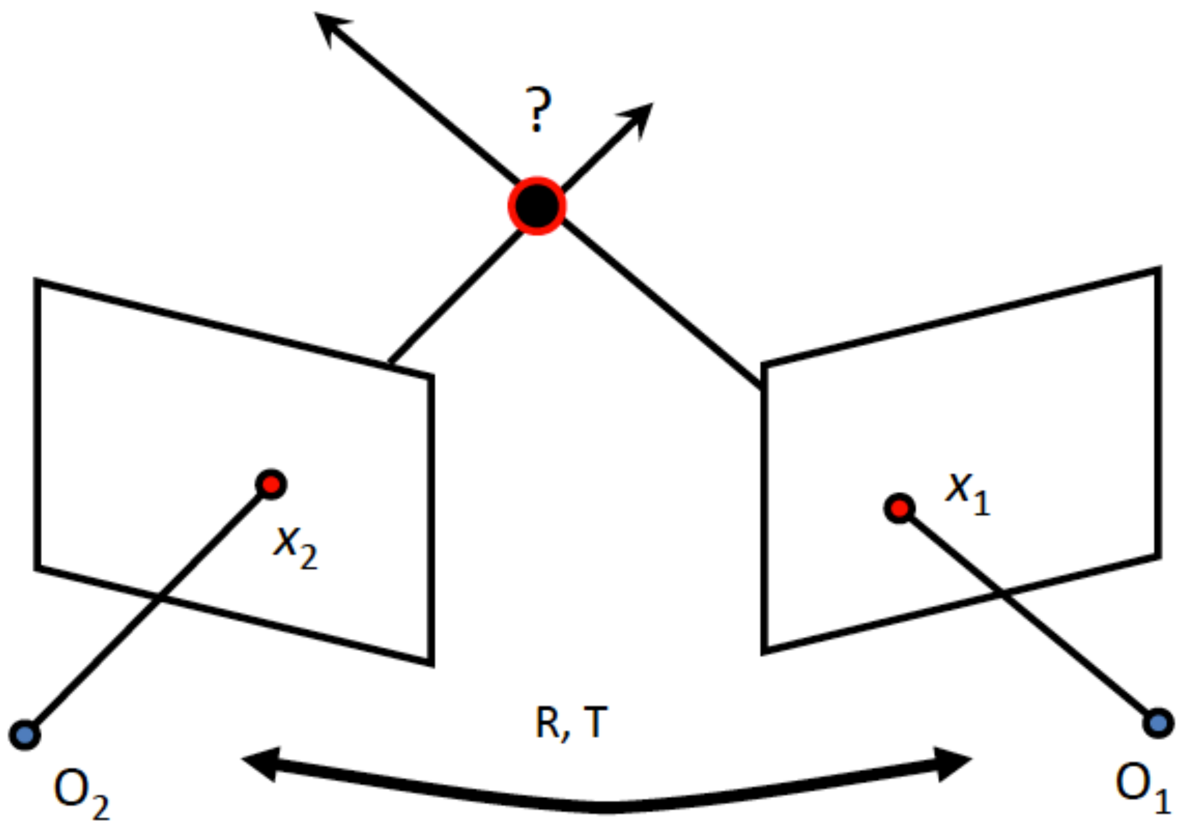
Multiple Views

Why multiple views?

Structure and depth are inherently ambiguous from single views.

Images from Lana Lazebnik

Two images help!



Stereo vision

Two cameras, simultaneous views

Single moving camera and static scene

3D Vision

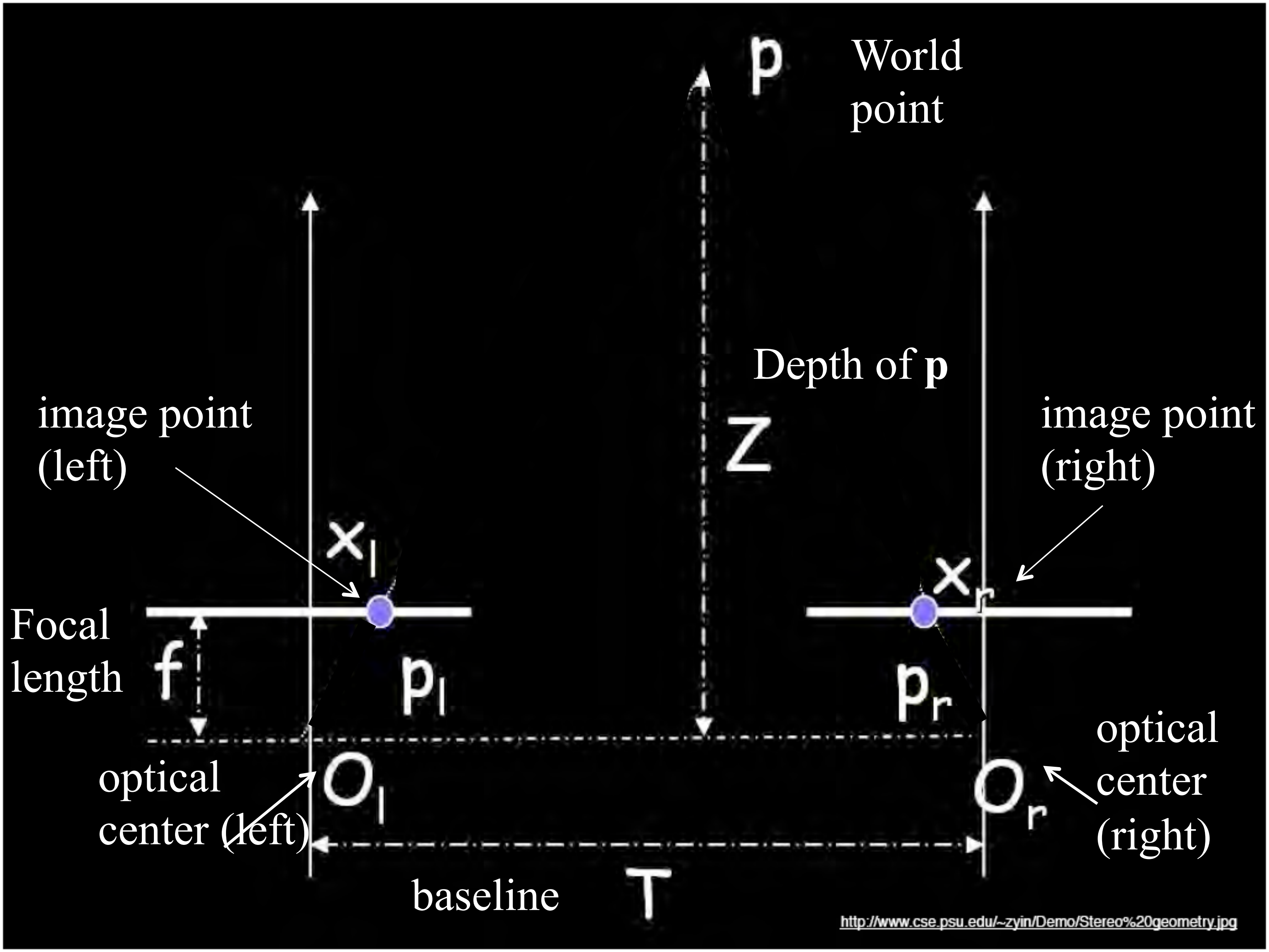
Methods will depend on

- Information available about cameras
- Information available about scene
- Correspondences available between views (points, lines etc)
- Number of views

Video: multi-view reconstruction

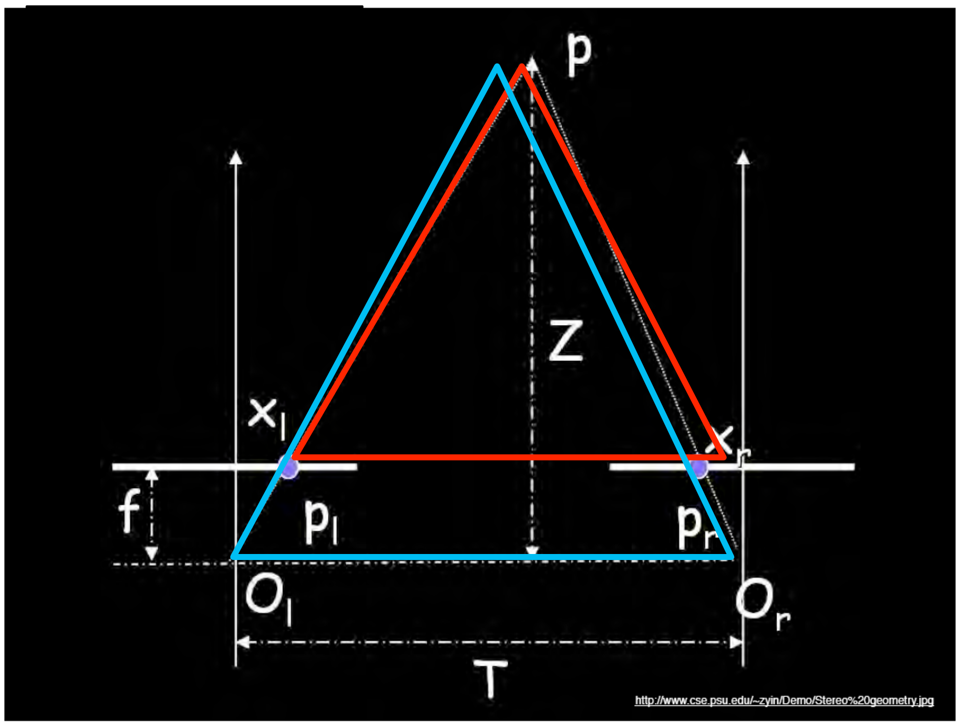
R. Szeliski et al. [Dubrovnik, Croatia]

Geometry for a simple stereo system
(parallel optical axis & calibrated cameras)


p
World point
Depth of **p**
Z
image point (left)
image point (right)
x_l
x_r
Focal length
f
p_l
p_r
optical center (left)
O_l
O_r
optical center (right)
baseline
T
http://www.cse.psu.edu/~zyin/Demo/Stereo%20geometry.jpg

Geometry for a simple stereo system

Assume parallel optical axes, known camera parameters (i.e., calibrated cameras). **What is expression for Z?**



Similar triangles (p_l, P, p_r) and (O_l, P, O_r):

$$\frac{T + x_l - x_r}{Z - f} = \frac{T}{Z}$$

$$Z = f \frac{T}{x_r - x_l}$$

disparity

Depth from disparity

image I(x,y)

Disparity map D(x,y)

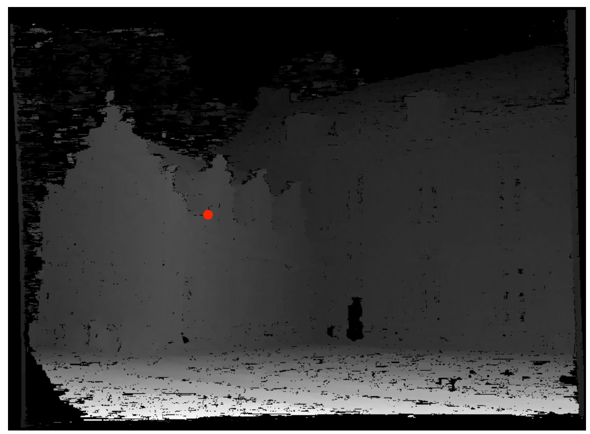


image I´(x´,y´)

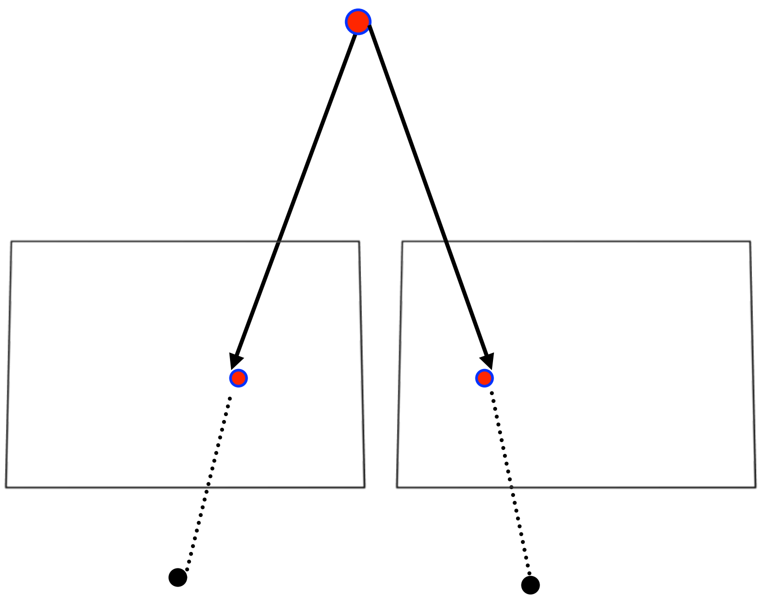
$$(x',y')=(x+D(x,y),\ y)$$

So if we could find the **corresponding points** in two images, we could **estimate relative depth**…

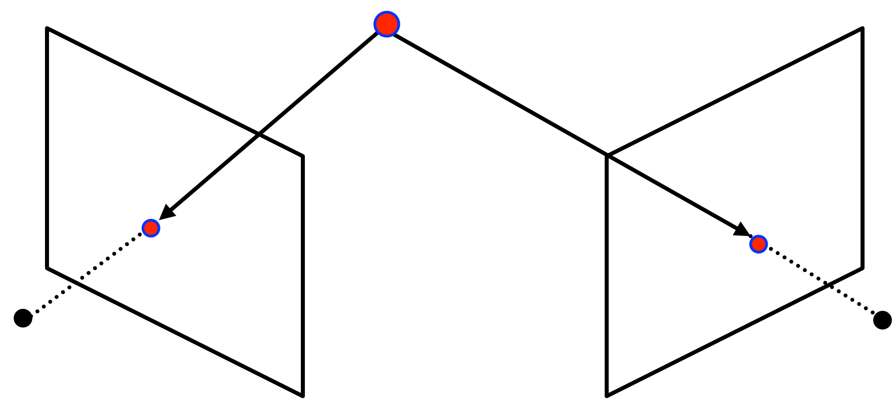
Epipolar geometry

General case, with calibrated cameras

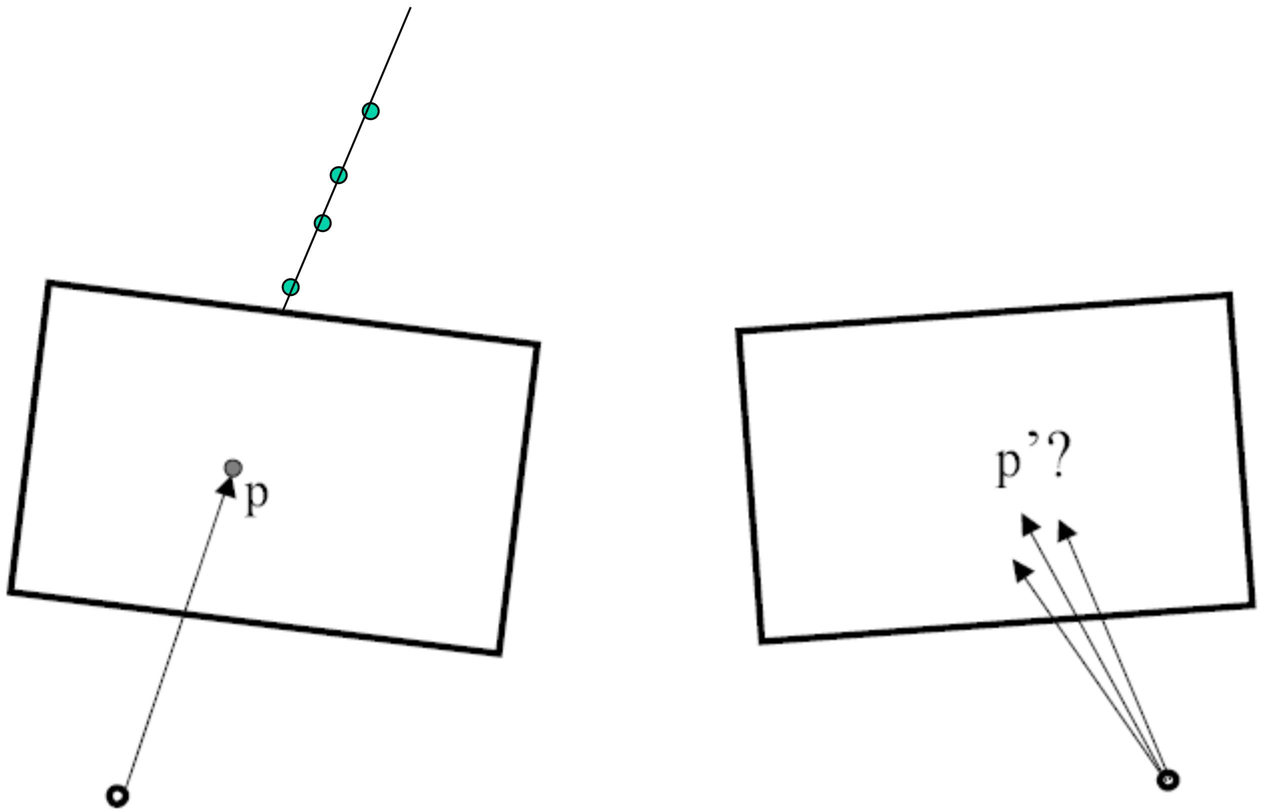
The two cameras need not have parallel optical axes.



Vs.

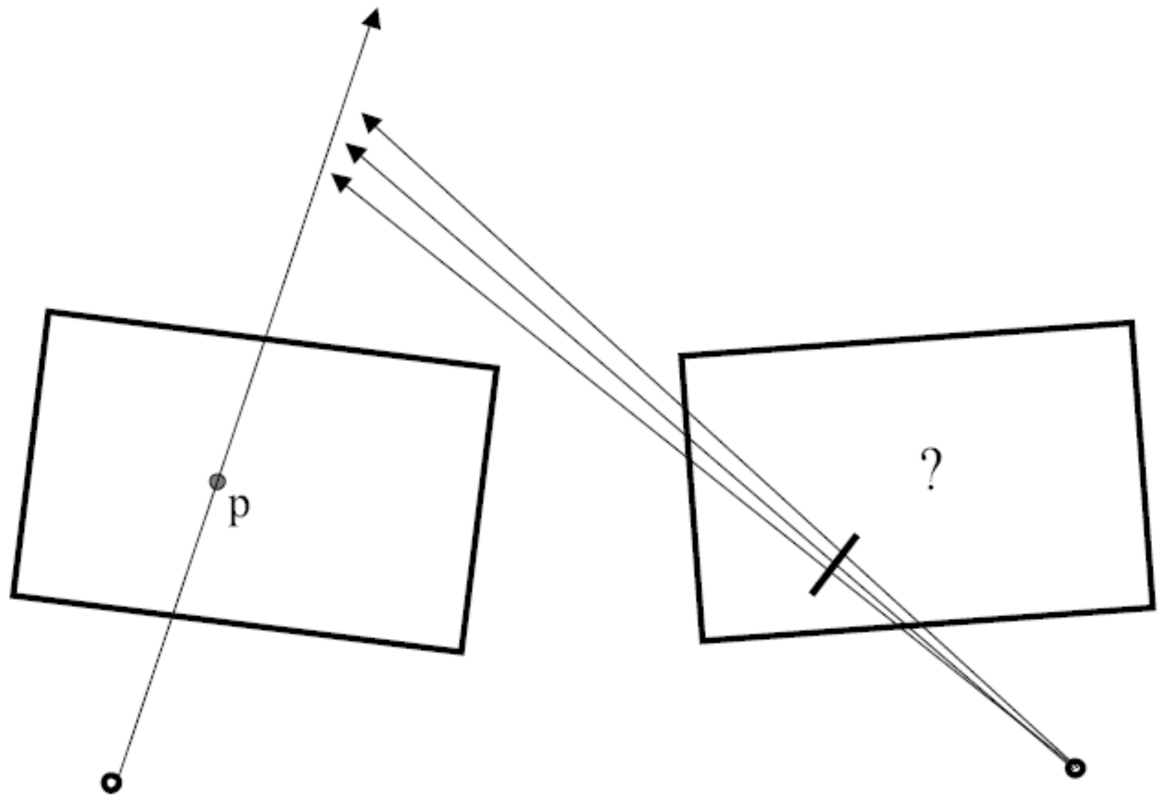


Stereo correspondence constraints

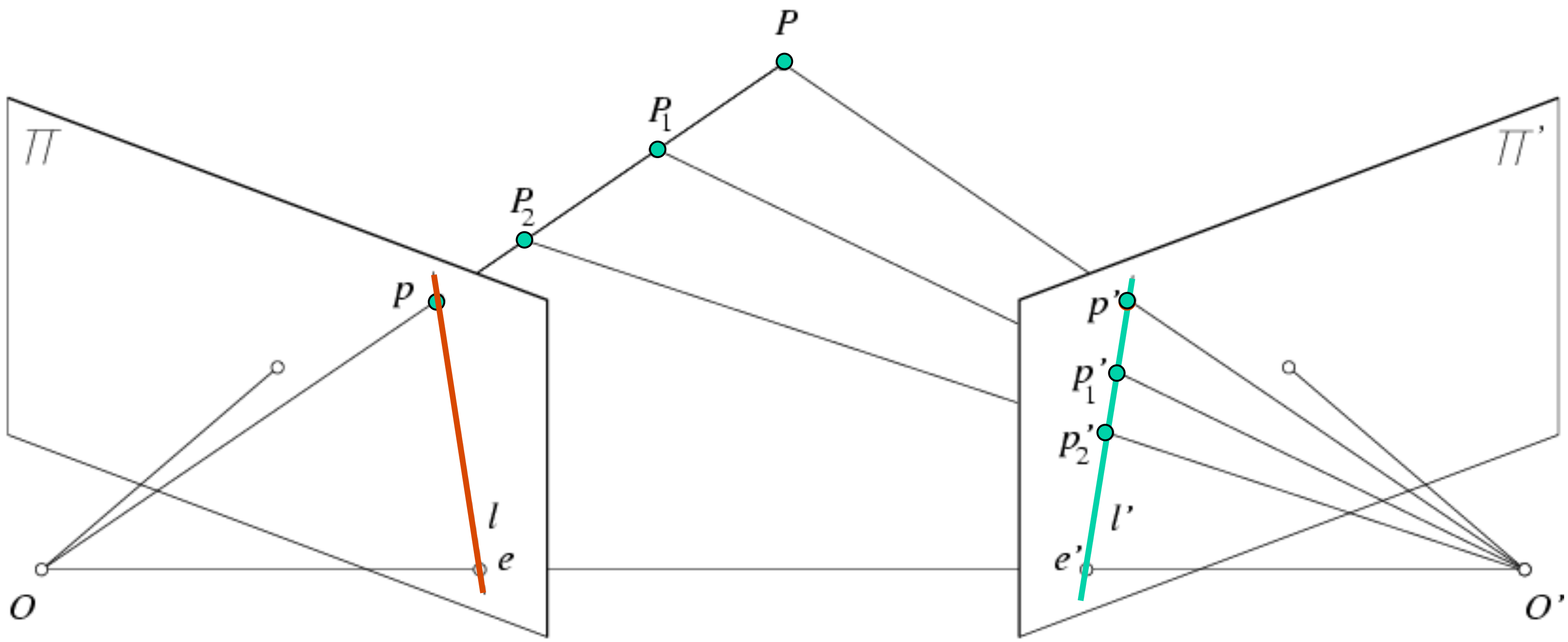


- Given p in left image, where can corresponding point p' be?

Stereo correspondence constraints



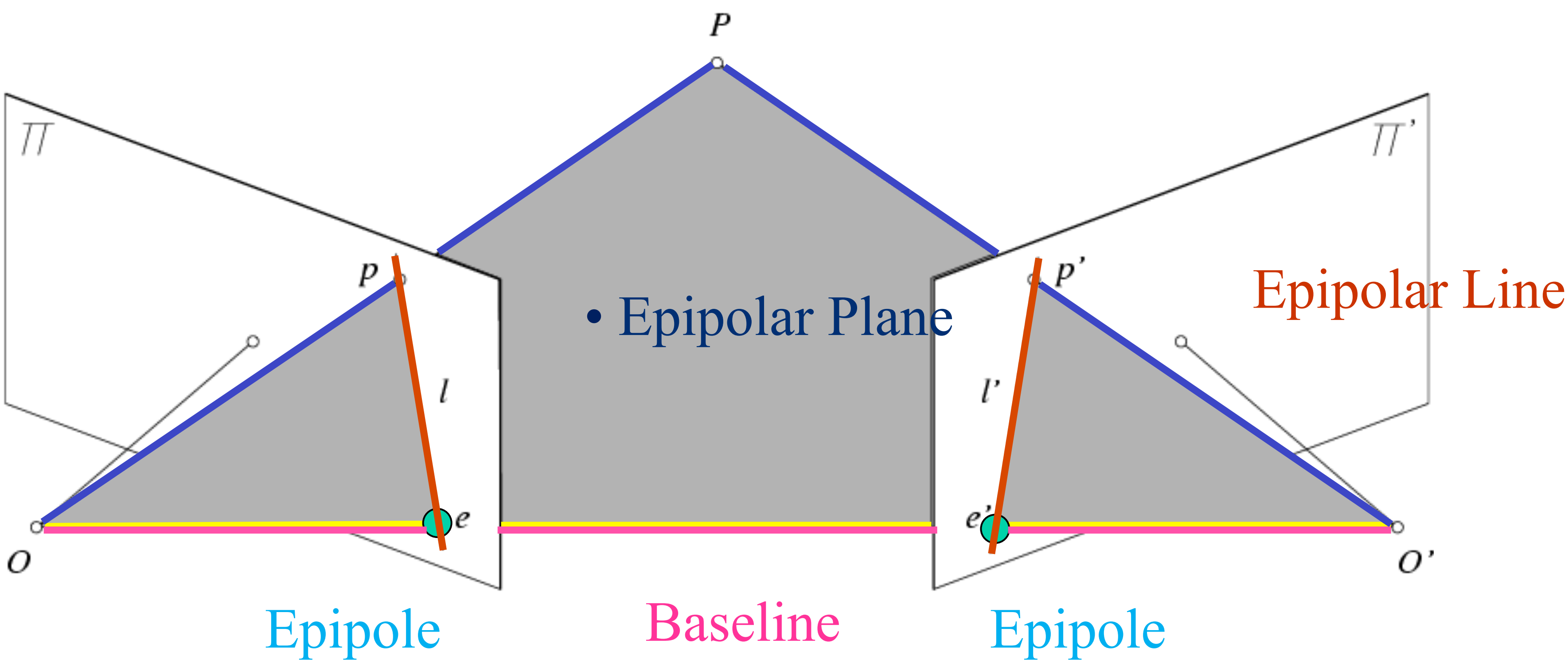
Epipolar constraint



Geometry of two views constrains where the corresponding pixel for some image point in the first view must occur in the second view.

- It must be on the line carved out by a plane connecting the world point and optical centers.

Epipolar geometry



http://www.ai.sri.com/~luong/research/Meta3DViewer/EpipolarGeo.html

Epipolar geometry: terms

Baseline: line joining the camera centers

Epipole: point of intersection of baseline with image plane

Epipolar plane: plane containing baseline and world point

Epipolar line: intersection of epipolar plane with the image plane

All epipolar lines intersect at the epipole

An epipolar plane intersects the left and right image planes in epipolar lines

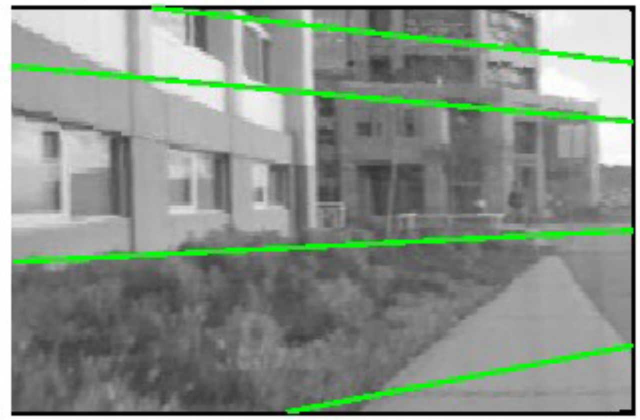
Why is the epipolar constraint useful?

Epipolar constraint

This is useful because it reduces the correspondence problem to a 1D search along an epipolar line.

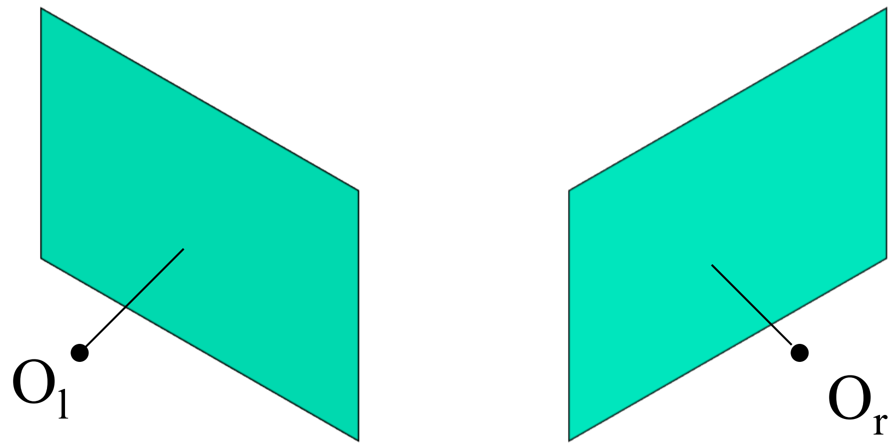
Image from Andrew Zisserman

Example

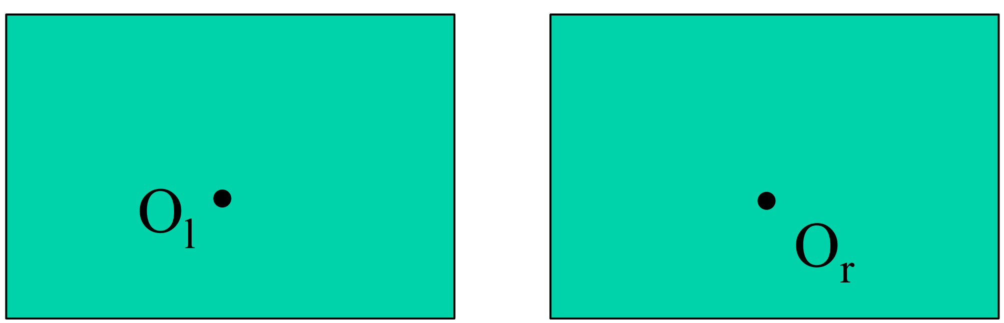


What do the epipolar lines look like?

1.



2.



Example: converging cameras

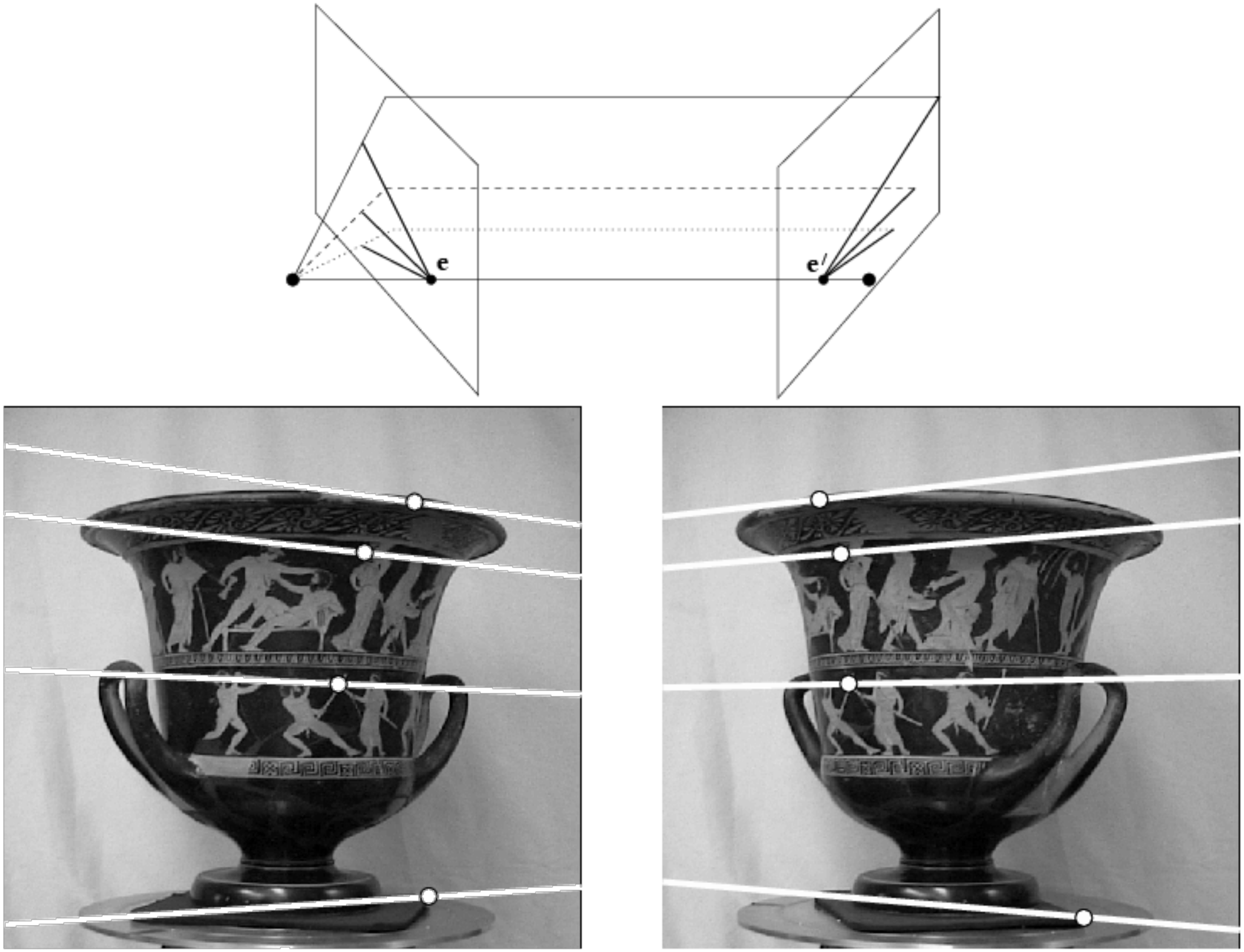
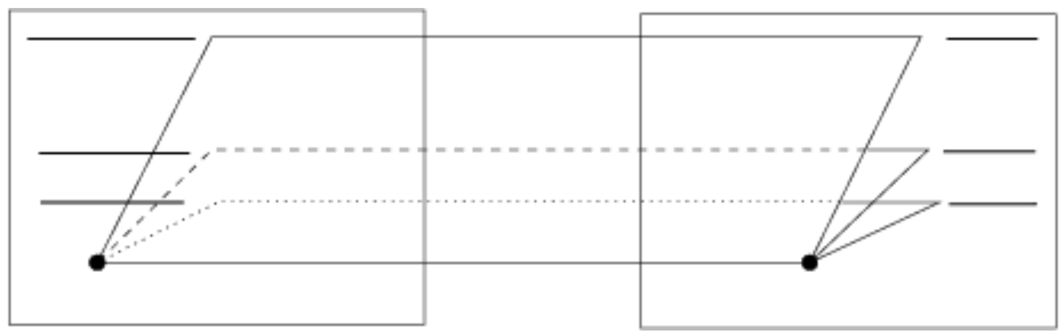


Figure from Hartley & Zisserman

Example: parallel cameras



Where are the epipoles?

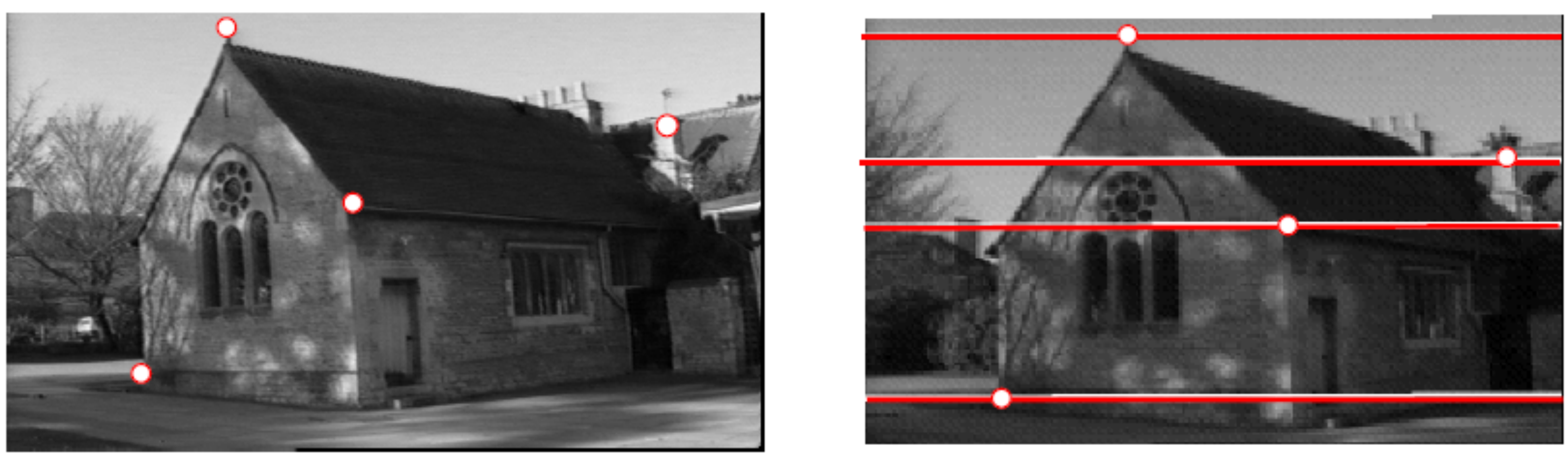
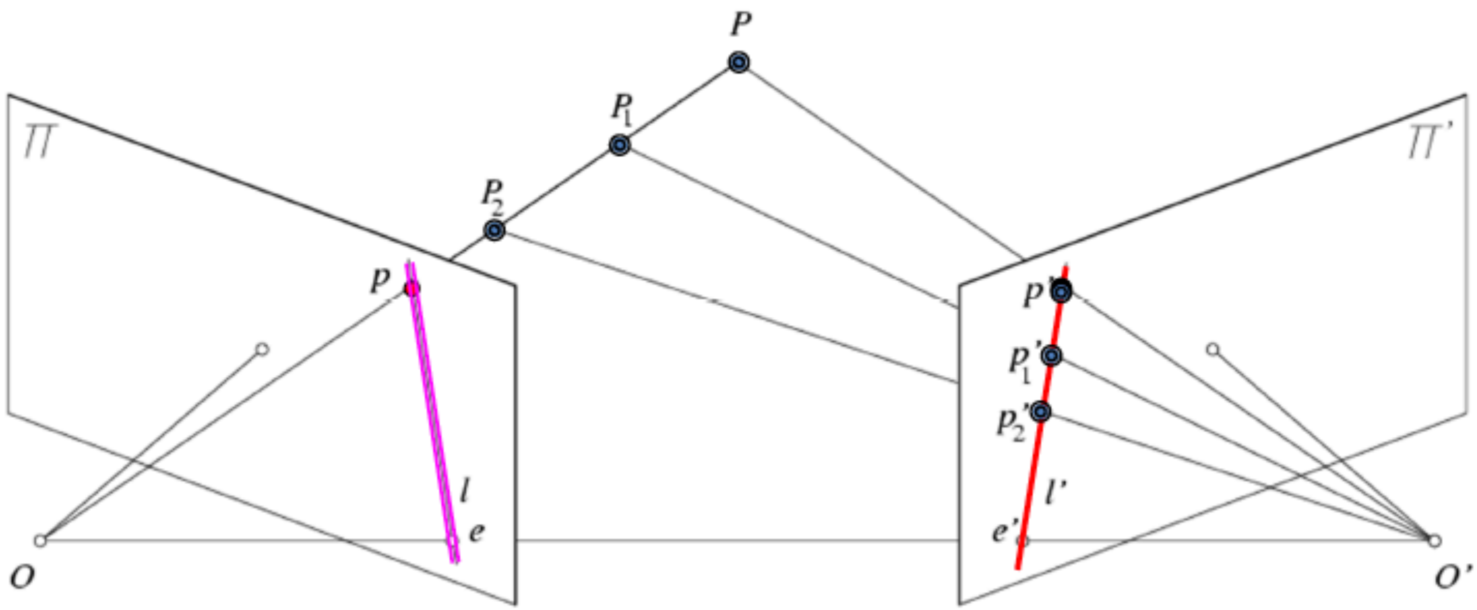


Figure from Hartley & Zisserman

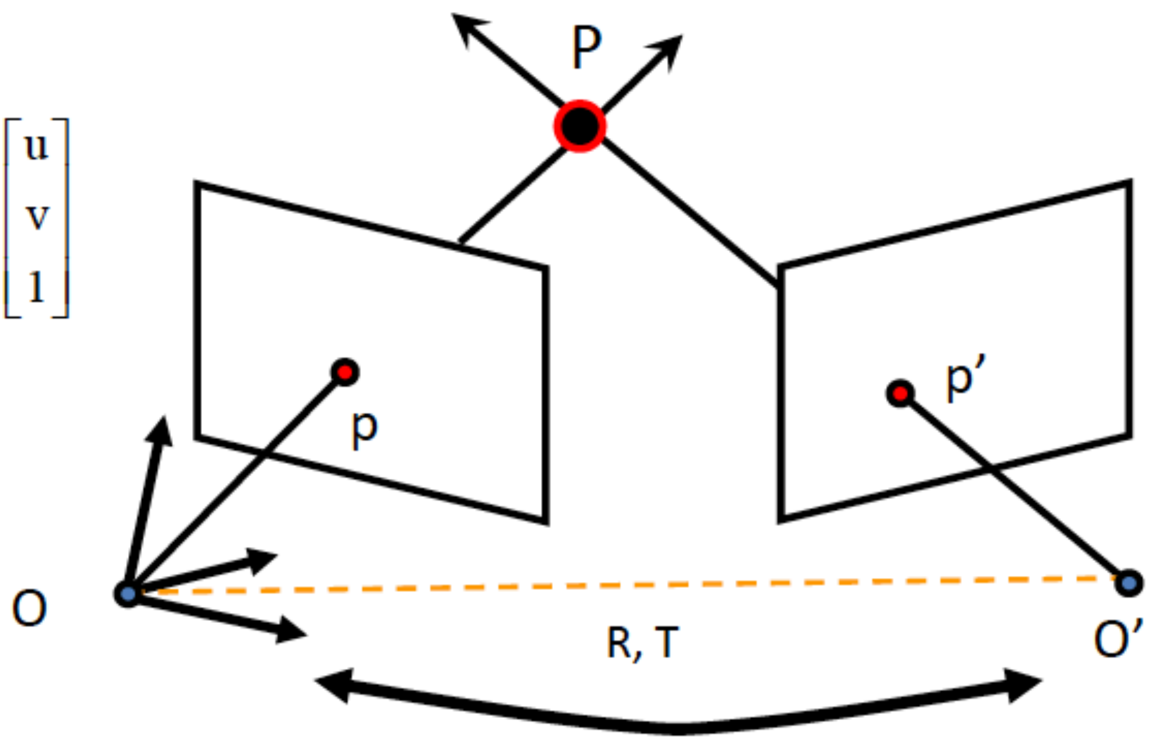
Epipolar constraint



- Potential matches for *p* have to lie on the corresponding epipolar line *l'*.
- Potential matches for *p'* have to lie on the corresponding epipolar line *l*.

Epipolar constraint

$$P \rightarrow M\,P = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

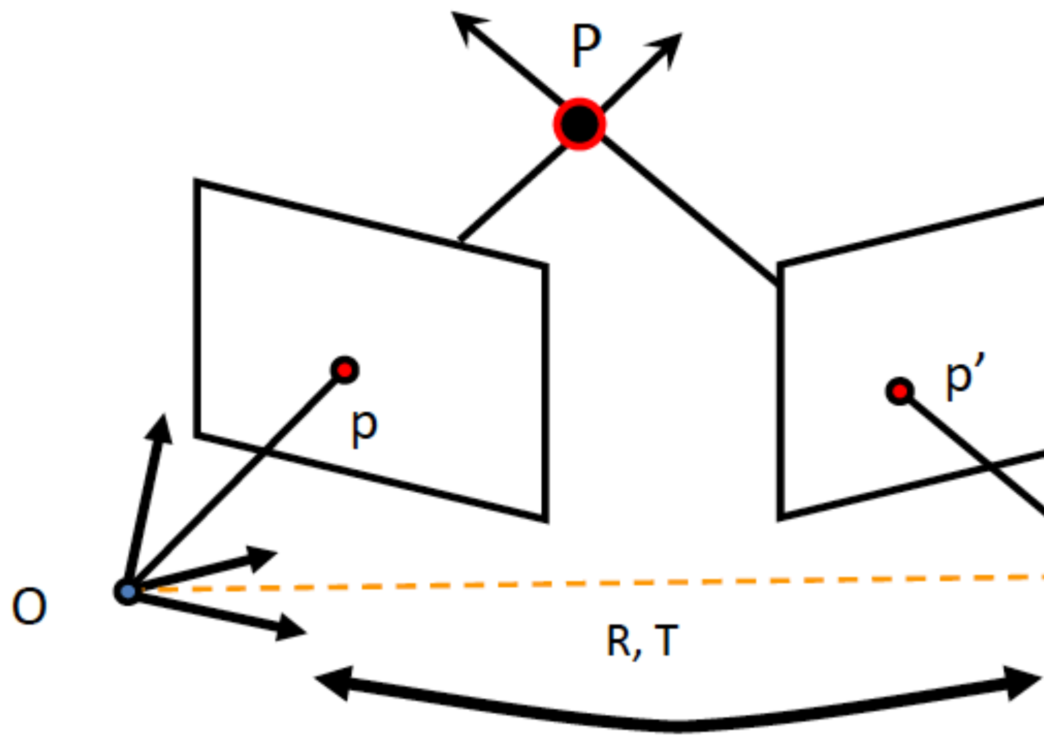


$$P \rightarrow M'\,P = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$$

$$M = K\begin{bmatrix} I & 0 \end{bmatrix}$$

$$M' = K\begin{bmatrix} R & T \end{bmatrix}$$

Epipolar constraint



$$M = K[I \quad 0]$$

$$\downarrow$$

$$M = [I \quad 0]$$

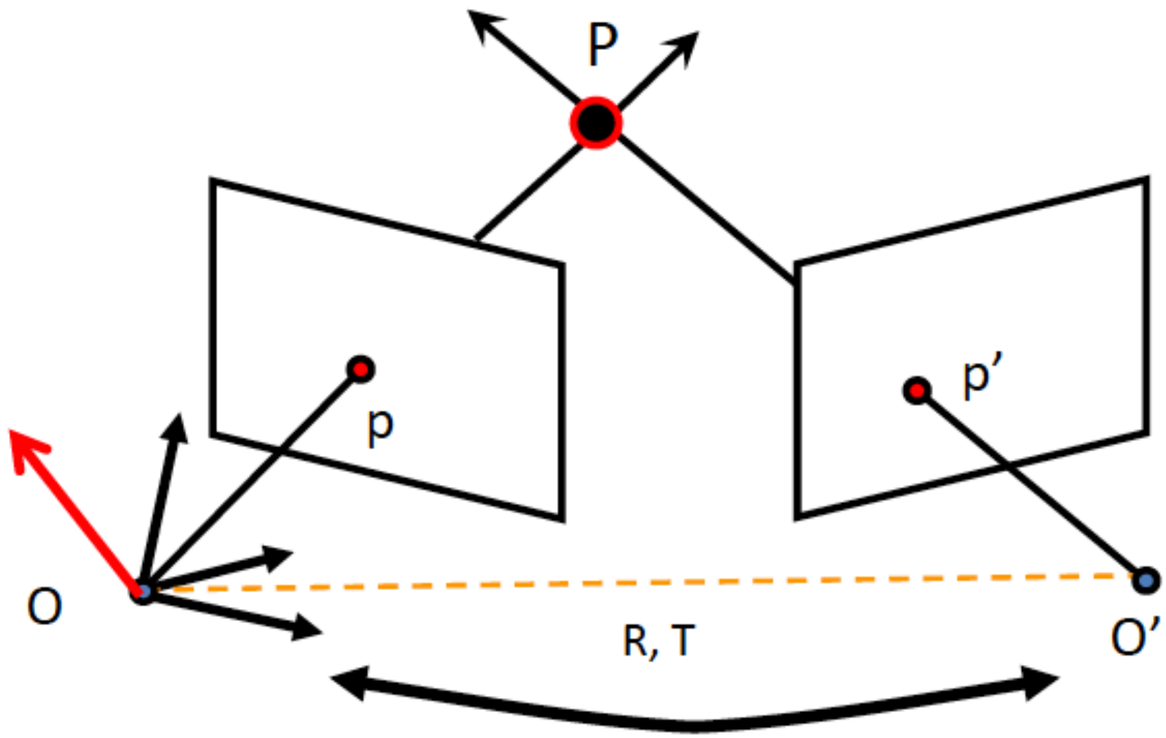
K_1 and K_2 are known
(calibrated cameras)

$$M' = K[R \quad T]$$

$$\downarrow$$

$$M' = [R \quad T]$$

Epipolar constraint



$$T \times (R\,p')$$

Perpendicular to epipolar plane

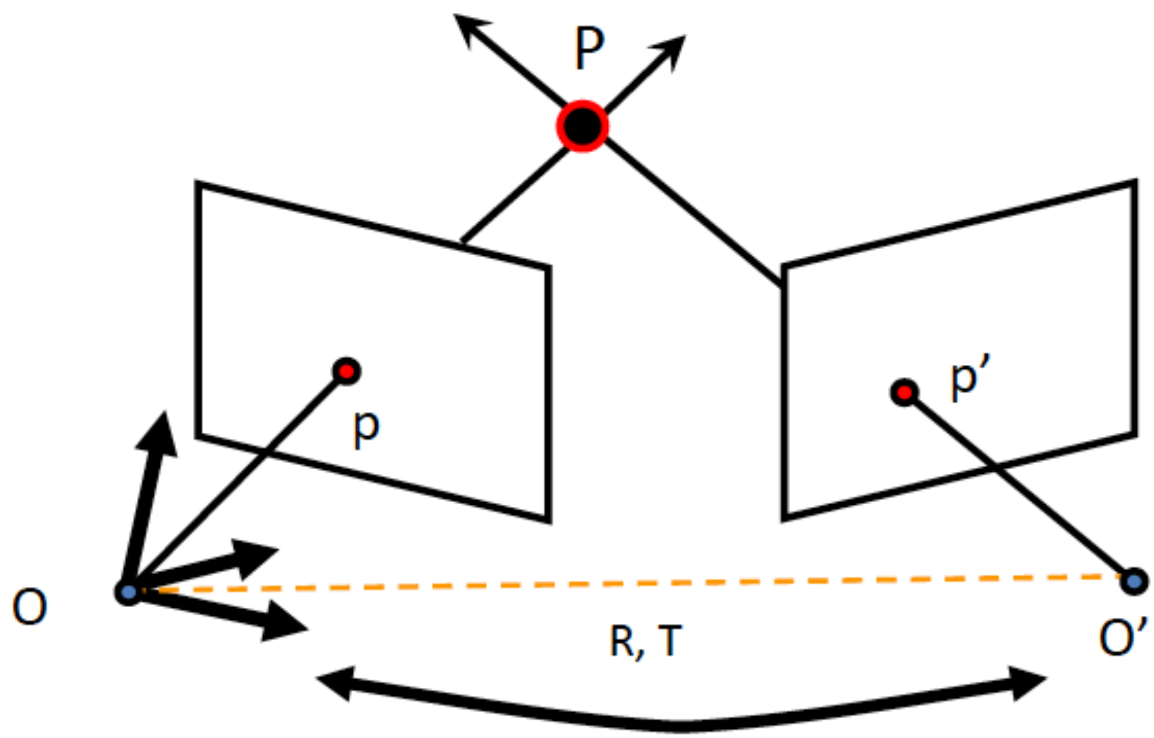
$$\mathrm{p}^{\mathrm{T}} \cdot [\mathrm{T} \times (\mathrm{R}\,\mathrm{p}')] = 0$$

Cross product as matrix multiplication

$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = [\mathbf{a}_\times]\mathbf{b}$$

"skew symmetric matrix"

Epipolar constraint



$$p^T \cdot [T \times (R\,p')] = 0 \rightarrow p^T \cdot \boxed{[T_\times] \cdot R}\, p' = 0$$

(Longuet-Higgins, 1981) E = essential matrix

Essential matrix

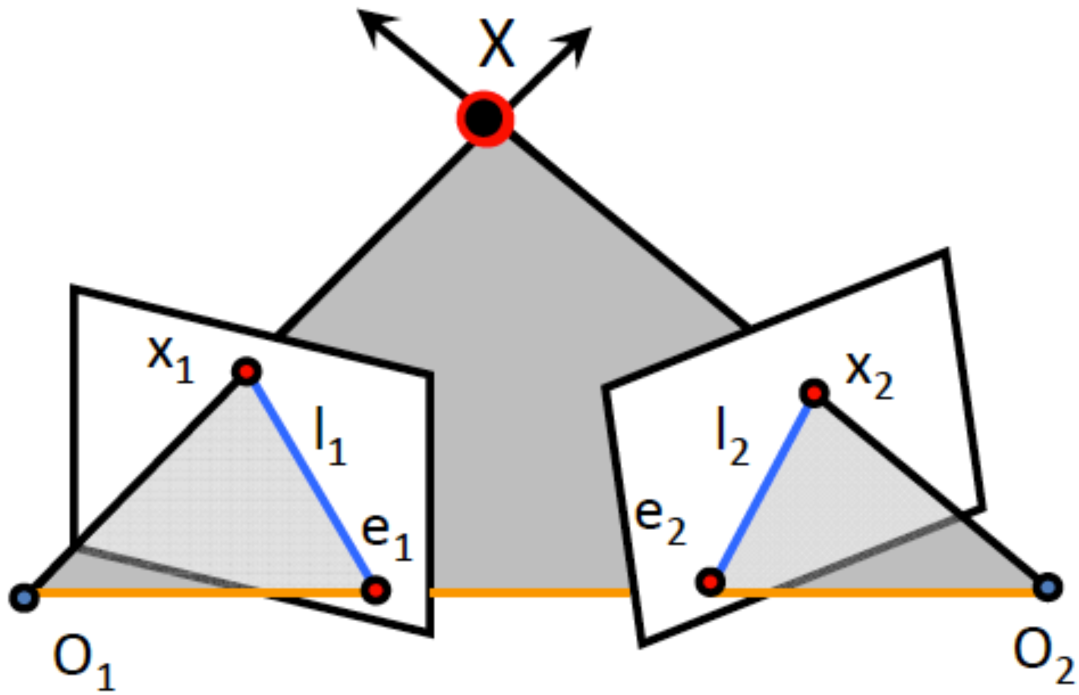
$$\mathbf{E} = [\mathrm{T}_x]\mathbf{R}$$

E is called the **essential matrix**, and it relates corresponding image points between both cameras, given the rotation and translation.

If we observe a point in one image, its position in other image is constrained to lie on line defined by above.

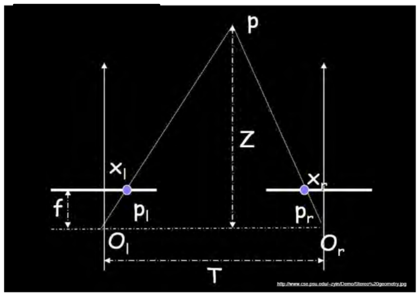
Note: these points are in **camera coordinate systems**.

Essential matrix



- $E\,x_2$ is the epipolar line associated with x_2 ($l_1 = E\,x_2$)
- $E^T x_1$ is the epipolar line associated with x_1 ($l_2 = E^T x_1$)
- E is singular (rank two)
- $E\,e_2 = 0$ and $E^T e_1 = 0$
- E is 3x3 matrix; 5 DOF

Essential matrix example: parallel cameras



$$\mathbf{R} = \mathbf{I}$$

$$\mathbf{T} = [-d, 0, 0]^{\mathrm{T}}$$

$$\mathbf{E} = [\mathrm{T}_{\mathbf{x}}]\mathbf{R} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{pmatrix}$$

$$\mathbf{p} = [x, y, f]$$

$$\mathbf{p'} = [x', y', f]$$

$$\mathbf{p}'^{\mathrm{T}}\mathbf{E}\mathbf{p} = 0$$

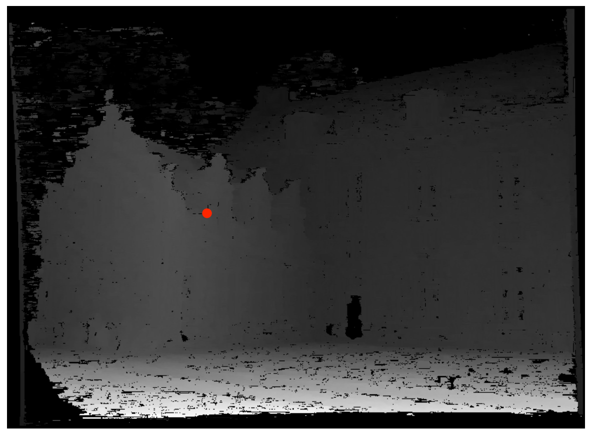
$$\begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ f \end{bmatrix} = 0$$

$$\Leftrightarrow \begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 \\ df \\ -dy \end{bmatrix} = 0$$

$$\boxed{\Leftrightarrow y = y'}$$

For the parallel cameras, image of any point must lie on same horizontal line in each image plane.

image $I(x,y)$ Disparity map $D(x,y)$ image $I'(x',y')$

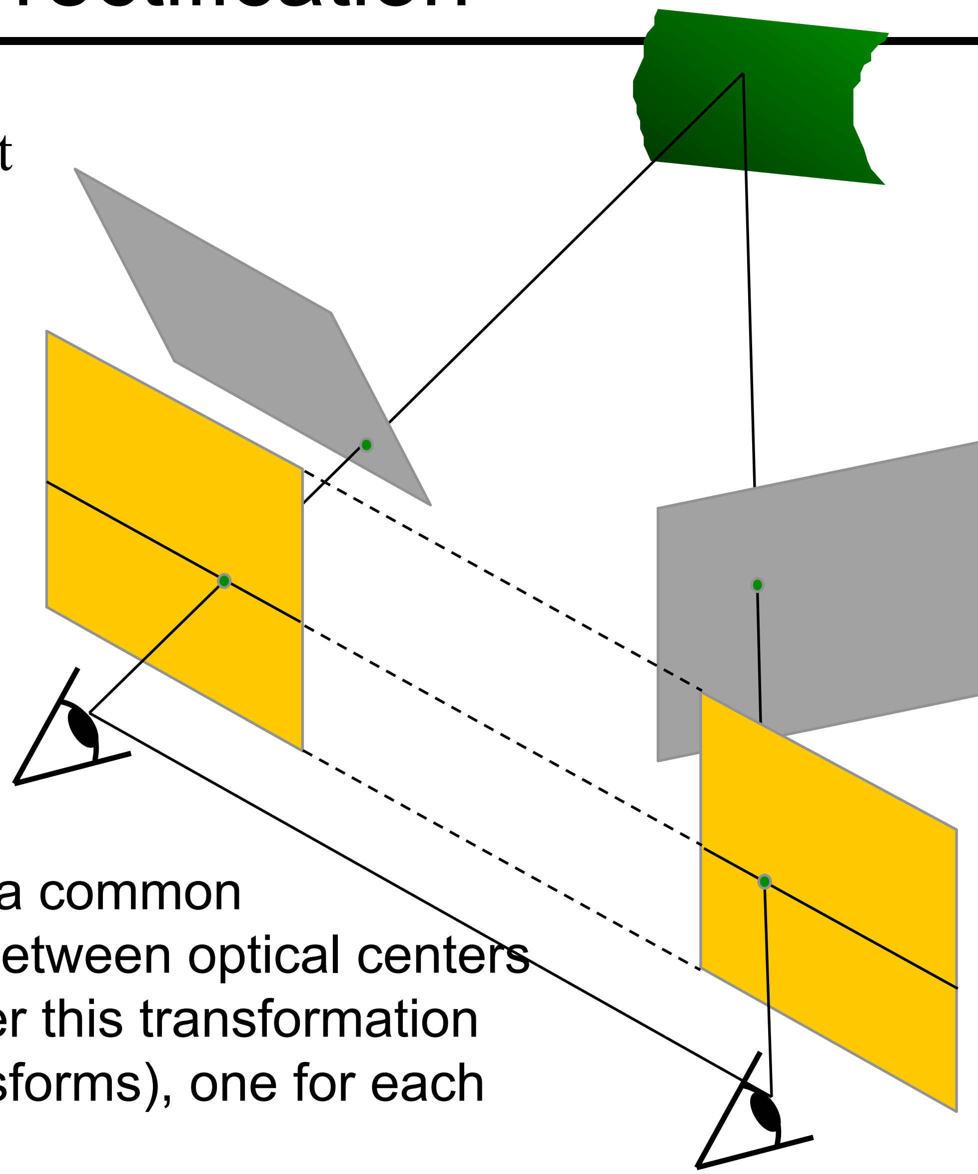


$$(x', \boxed{y'}) = (x + D(x,y), \boxed{y})$$

What about when cameras' optical axes are not parallel?

Stereo image rectification

In practice, it is convenient if image scanlines (rows) are the epipolar lines.



- reproject image planes onto a common plane parallel to the line between optical centers
- pixel motion is horizontal after this transformation
- two homographies (3x3 transforms), one for each input image reprojection

Slide credit: Li Zhang

Projection and Planar Homographies

- First Fundamental Theorem of Projective Geometry:
  - There exists a unique homography that performs a change of basis between two projective spaces of the same dimension.

$$
\begin{aligned}
s[u \quad v \quad 1]^T &= A[r_1 \quad r_2 \quad r_3 \quad t][X \quad Y \quad Z \quad 1]^T \\
s[u \quad v \quad 1]^T &= A[r_1 \quad r_2 \quad r_3 \quad t][X \quad Y \quad 0 \quad 1]^T \\
s[u \quad v \quad 1]^T &= A[r_1 \quad r_2 \quad t][X \quad Y \quad 1]^T \\
s[u \quad v \quad 1]^T &= H[X \quad Y \quad 1]^T
\end{aligned}
$$

  - Projection Becomes

$$s\tilde{m} = H\tilde{M}$$

  - Notice that the homography H is defined up to scale (s).

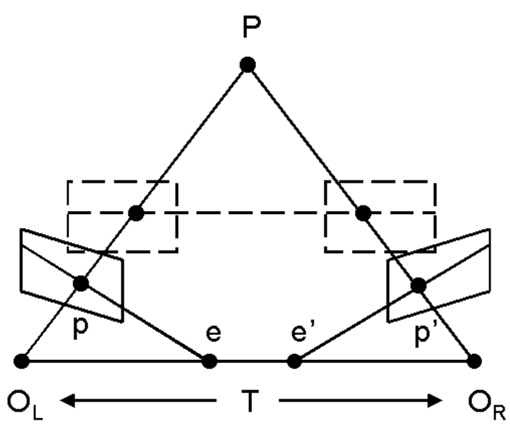
How to Change Viewing Geometry

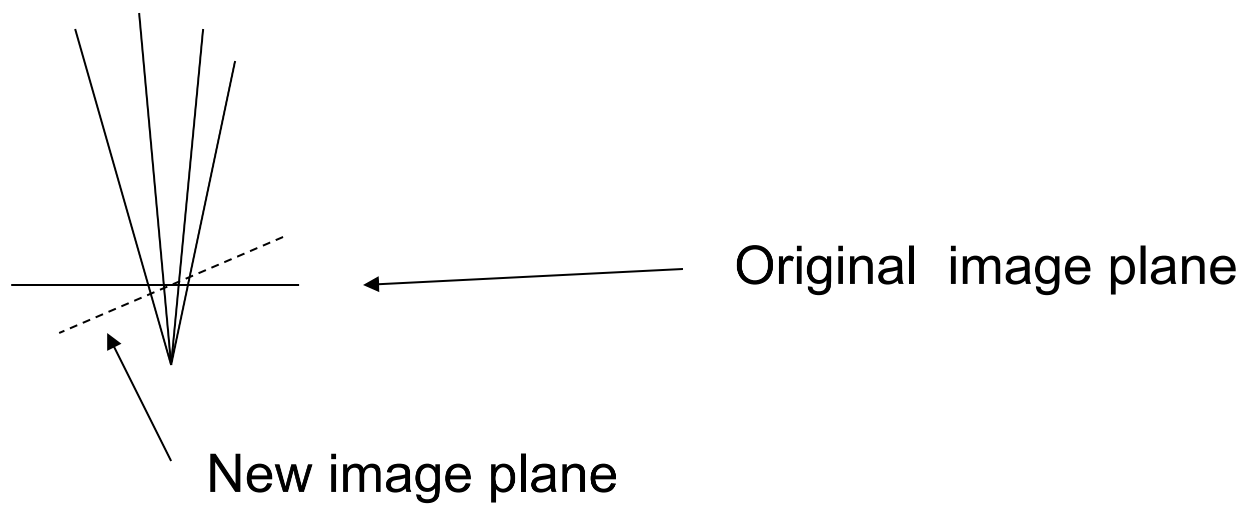
Homographies can be used to compute an image as seen by a rotated camera

Consider a point p which is imaged as $q = K(\lambda p) = \lambda K p$
with $q = (u, v, 1)'$

Now, compute $p' = R p$, or $p' = \lambda R K^{-1} q$

Finally, reproject: $q' = \lambda' K p' = \lambda'' K R K^{-1} q = \lambda'' H q$ --- H is a homography!





$$\mathbf{p_r}^{\mathbf{t}}\, \mathbf{E}\ \mathbf{p_l} = \mathbf{0}$$

There are two poses relative poses consistent with E up to sign

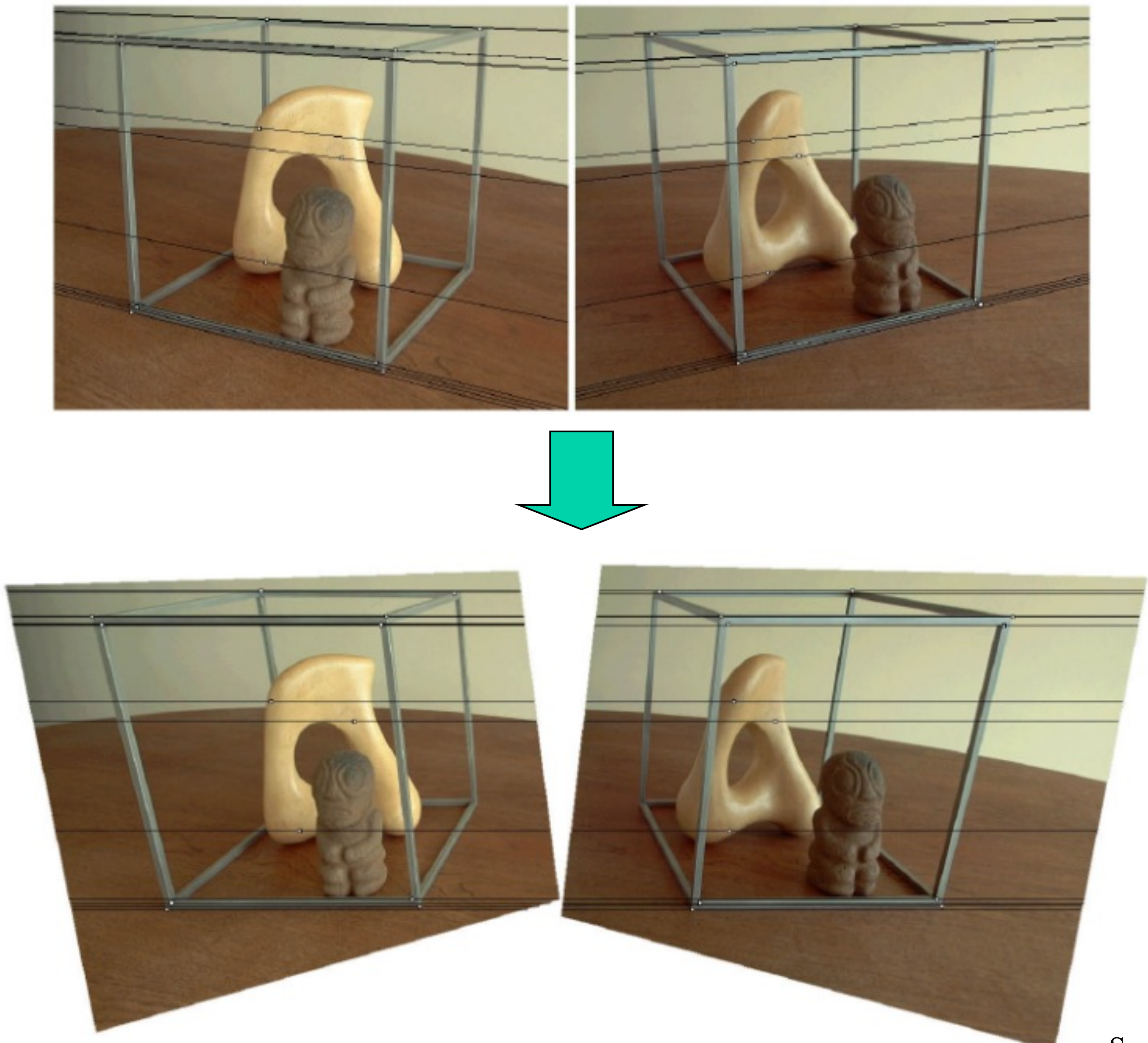
$E = U\ D\ V^t$

$(sk(T_1), R_1) = (U\ R_Z(+\pi/2)\ D\ U^t\ ,\ V\ R_Z(+\pi/2)\ V^t)$
$(sk(T_2), R_2) = (U\ R_Z(-\pi/2)\ D\ U^t\ ,\ V\ R_Z(-\pi/2)\ V^t)$

$D = \mathrm{diag}(1, 1, 0)$

The four possible solutions (including sign of T) can be disambiguated by ensuring all points are in front of the cameras

Stereo image rectification: example

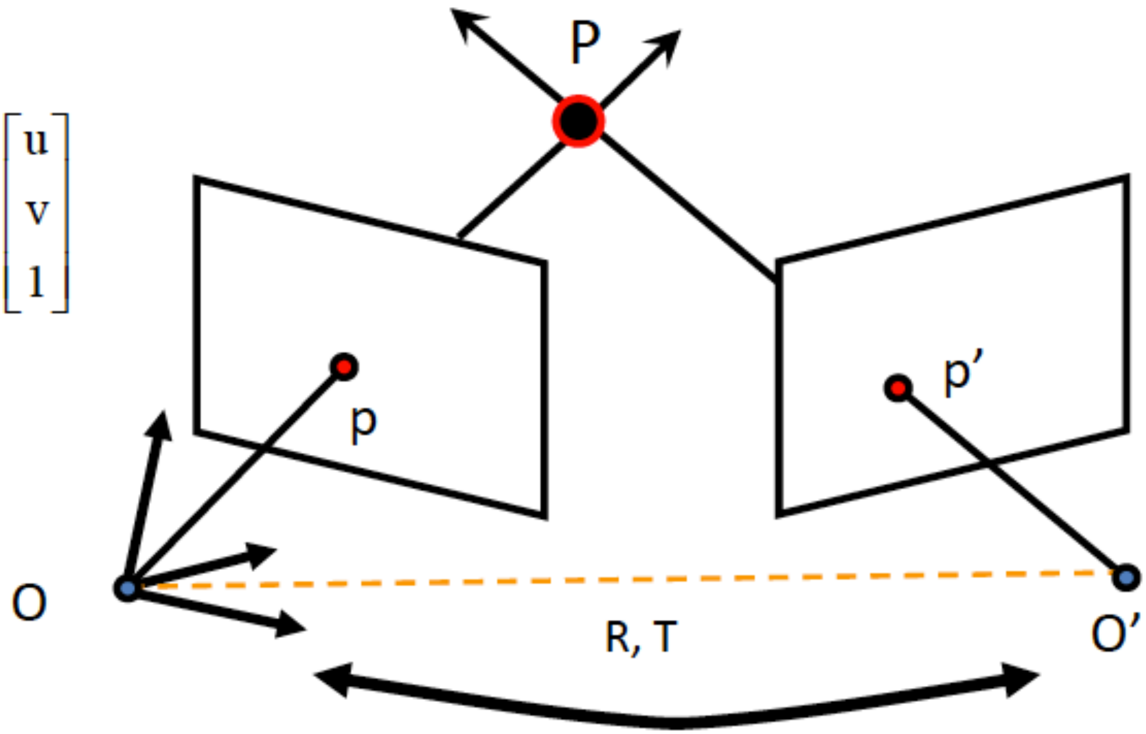


Source: Alyosha Efros

Epipolar geometry
(uncalibrated cameras)

Epipolar constraint

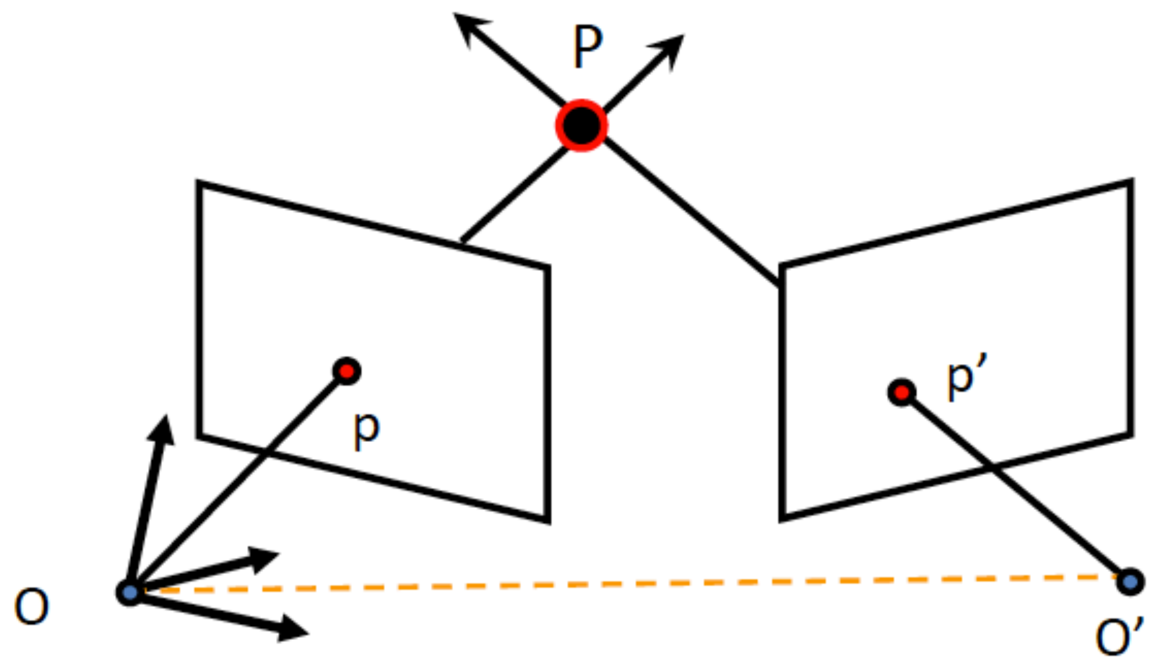
$$P \rightarrow M\,P = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$



$$P \rightarrow M'\,P = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$$

$$M = K\begin{bmatrix} I & 0 \end{bmatrix} \qquad M' = K\begin{bmatrix} R & T \end{bmatrix}$$

Epipolar constraint

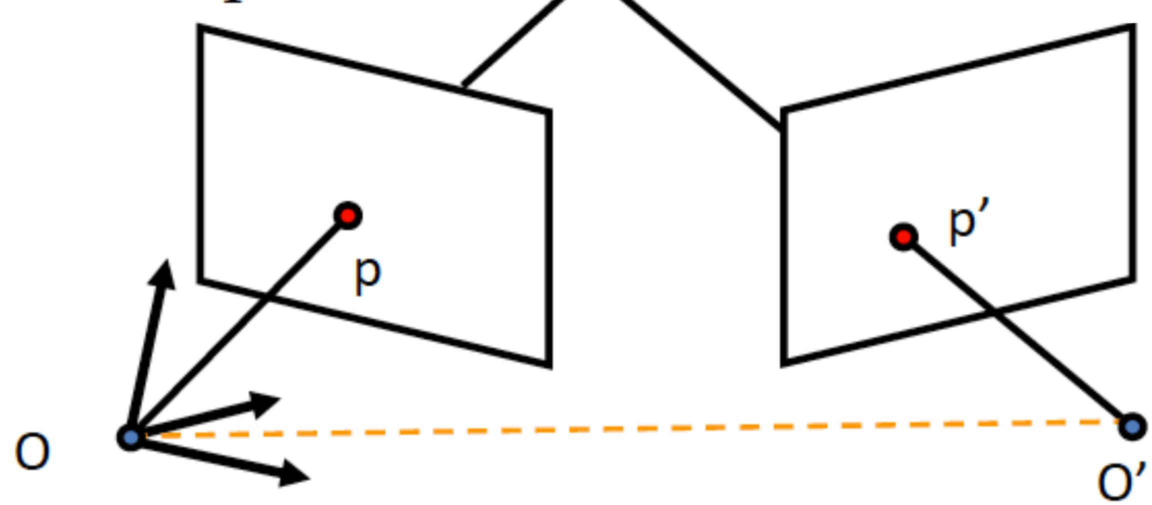


$$P \rightarrow M\,P \rightarrow p = \begin{bmatrix} u \\ v \end{bmatrix} \qquad M = \boxed{K}\begin{bmatrix} I & 0 \end{bmatrix}$$

unknown

Epipolar constraint

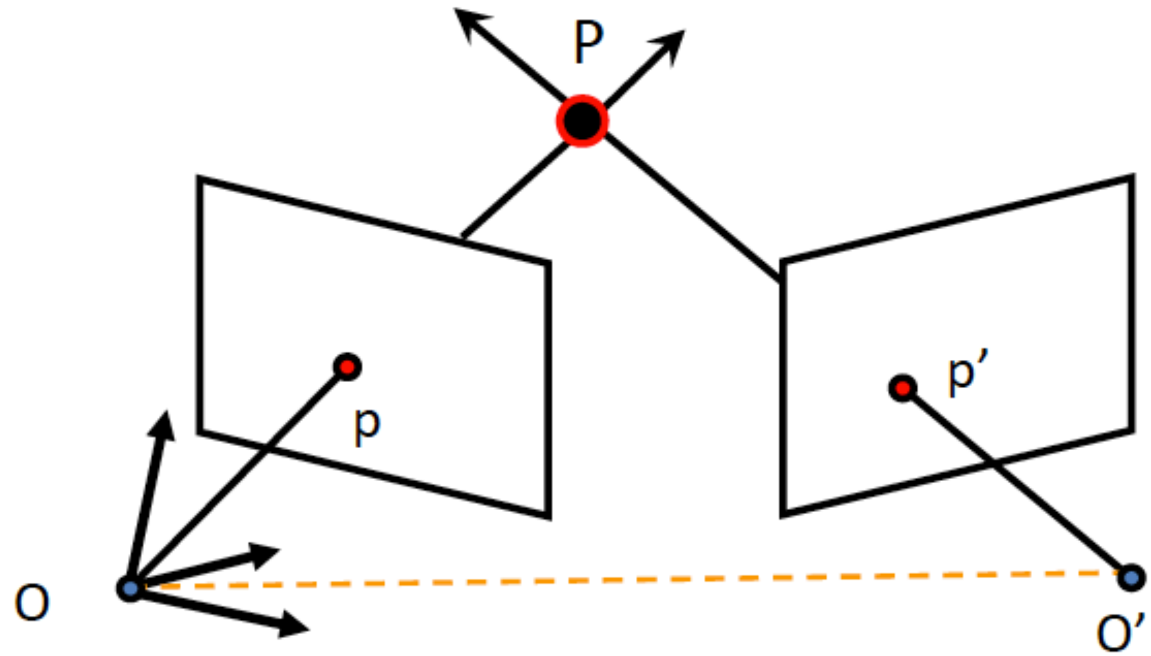
$$p \rightarrow K^{-1}\, p$$



$$\mathrm{p}^{T} \cdot [T_{\times}] \cdot R\, \mathrm{p}' = 0 \rightarrow (K^{-1}\, \mathrm{p})^{T} \cdot [T_{\times}] \cdot R\, K'^{-1}\, \mathrm{p}' = 0$$

$$\mathrm{p}^{T} \boxed{K^{-T} \cdot [T_{\times}] \cdot R\, K'^{-1}}\, \mathrm{p}' = 0 \rightarrow \mathrm{p}^{T} \boxed{F}\, \mathrm{p}' = 0$$

Fundamental matrix

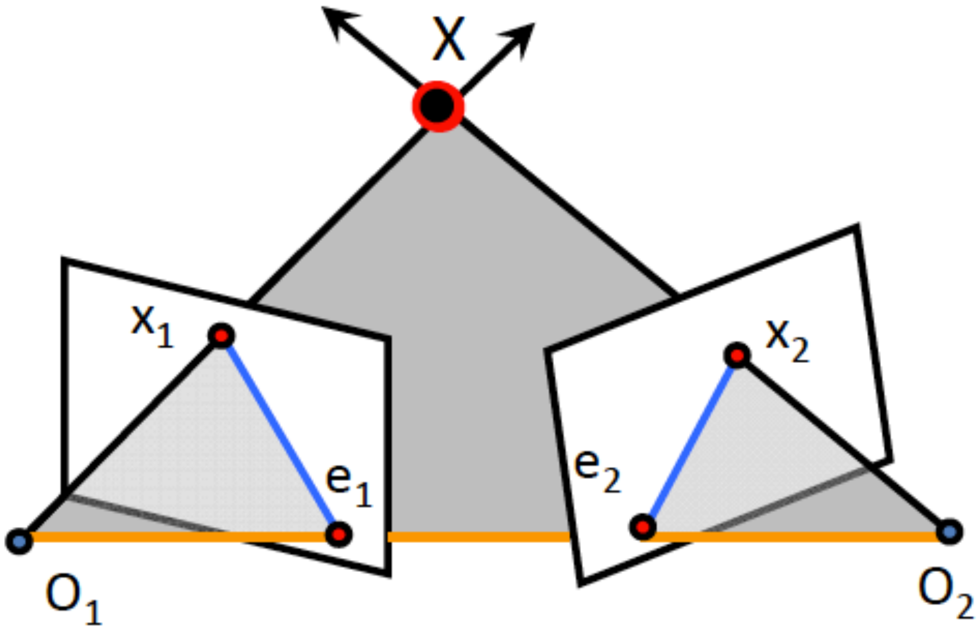


$$\boxed{p^T F p' = 0}$$

F = Fundamental Matrix

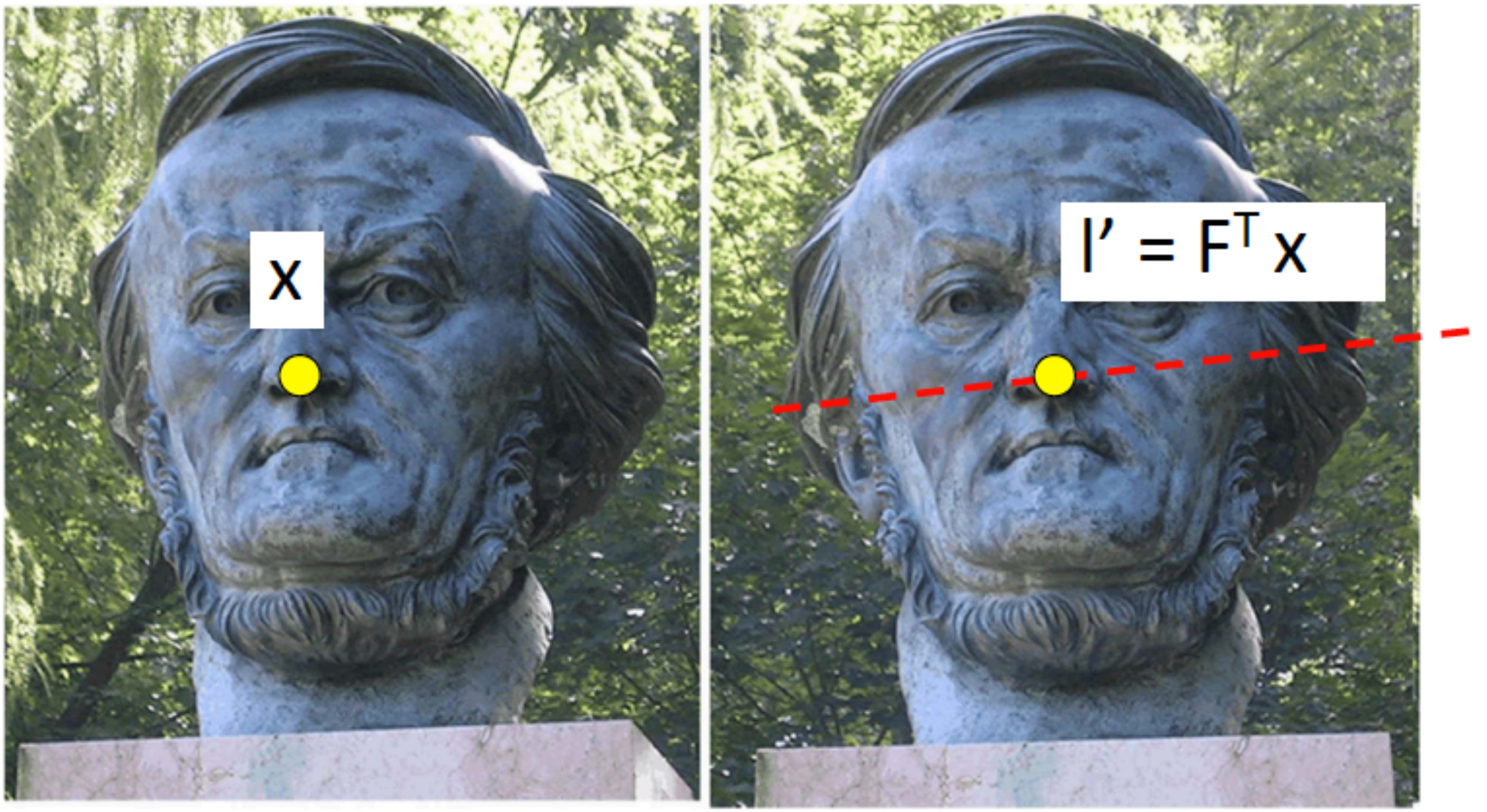
(Faugeras and Luong, 1992)

Fundamental matrix



- $F x_2$ is the epipolar line associated with x_2 ($l_1 = F x_2$)
- $F^T x_1$ is the epipolar line associated with x_1 ($l_2 = F^T x_1$)
- F is singular (rank two)
- $F e_2 = 0$ and $F^T e_1 = 0$
- F is 3x3 matrix; 7 DOF

Why is F useful ?



- Suppose F is known
- No additional information about the scene and camera is given
- Given a point on left image, how can I find the corresponding point on right image?

Why is F useful ?

- F captures information about the epipolar geometry of 2 views + camera parameters
- MORE IMPORTANTLY: F gives constraints on how the scene changes under view point transformation (without reconstructing the scene!)
- Powerful tool in:
  - 3D reconstruction
  - Multi-view object/scene matching

Computation of F from point correspondences

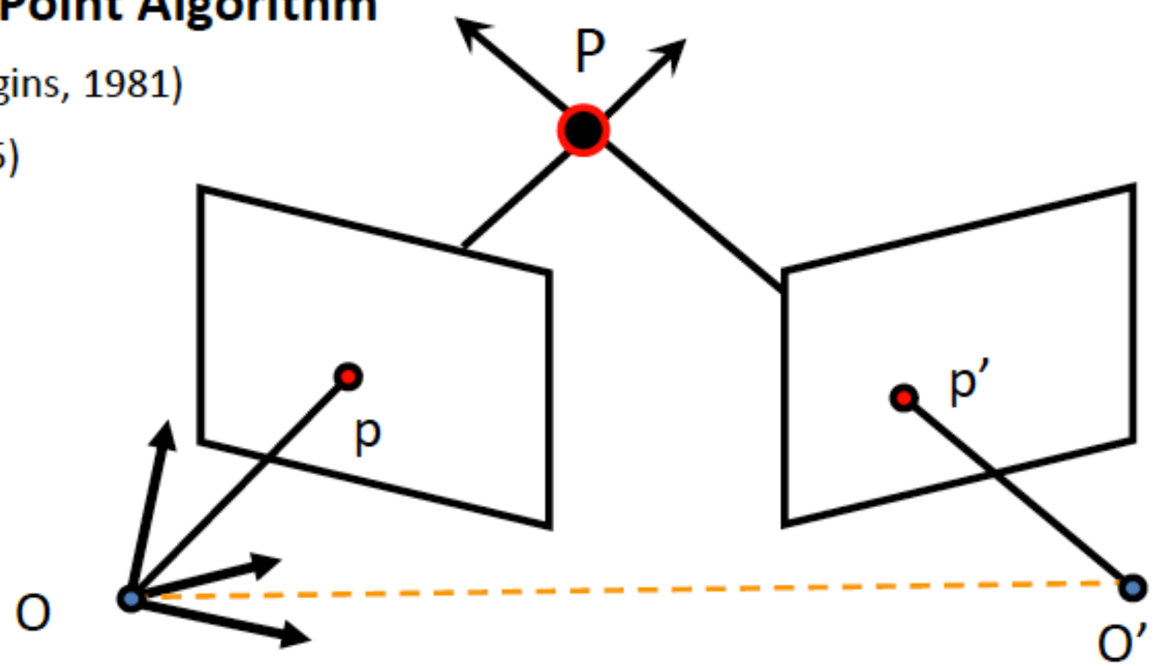
(similar approach for E if intrinsics are known)

Estimating F

The Eight-Point Algorithm

(Longuet-Higgins, 1981)

(Hartley, 1995)



$$P \rightarrow p = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad P \rightarrow p' = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \qquad p^T\, F\, p' = 0$$

Estimating F

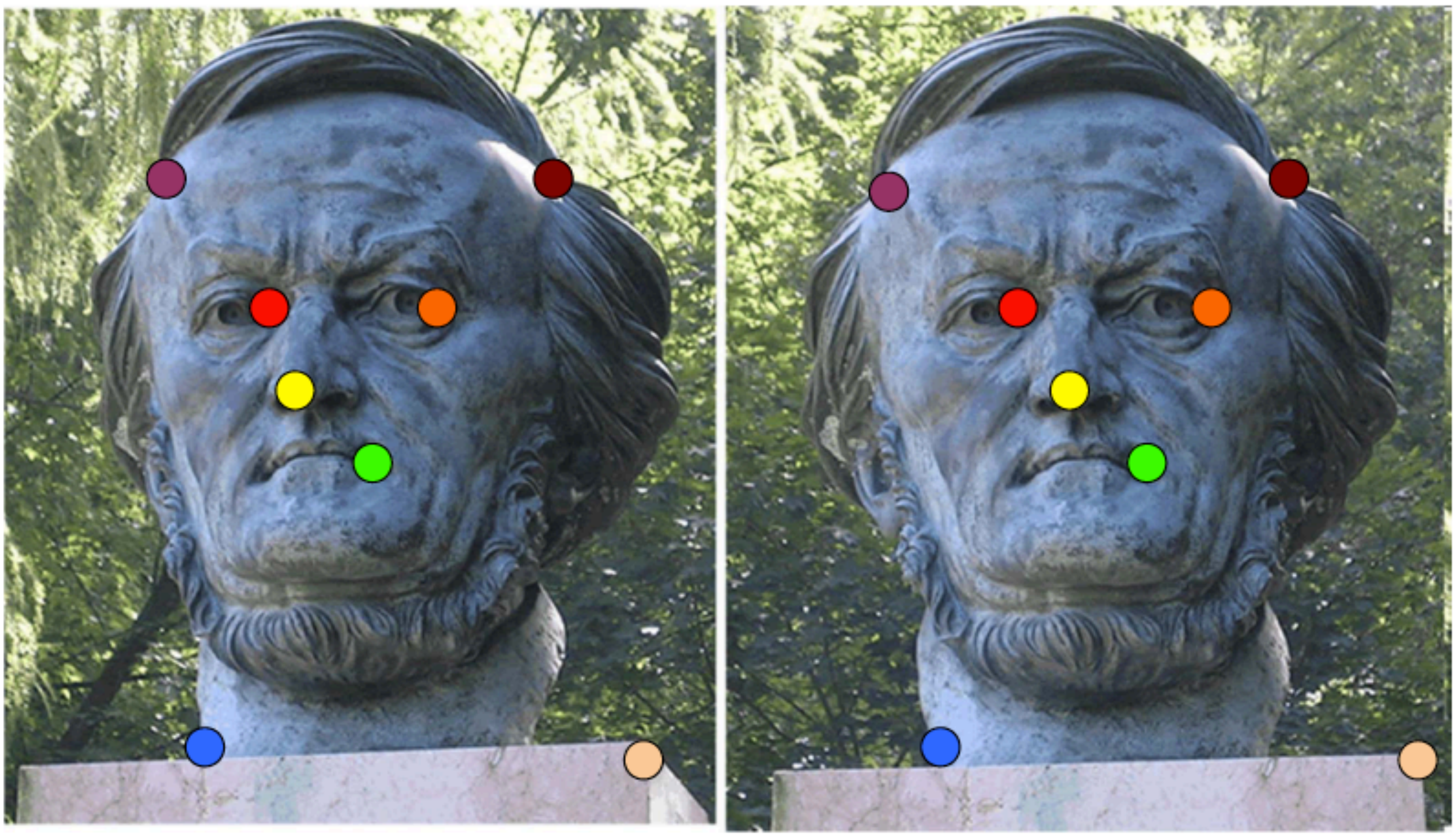
$$\mathrm{p}^{\mathrm{T}}\ \mathrm{F}\,\mathrm{p}' = 0 \quad \Rightarrow$$

$$(u, v, 1)\begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0$$

$$\Rightarrow \quad (uu', uv', u, vu', vv', v, u', v', 1)\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

Let’s take 8 corresponding points

Estimating F



Estimating F

$$\underbrace{\begin{pmatrix} u_1u_1' & u_1v_1' & u_1 & v_1u_1' & v_1v_1' & v_1 & u_1' & v_1' & 1 \\ u_2u_2' & u_2v_2' & u_2 & v_2u_2' & v_2v_2' & v_2 & u_2' & v_2' & 1 \\ u_3u_3' & u_3v_3' & u_3 & v_3u_3' & v_3v_3' & v_3 & u_3' & v_3' & 1 \\ u_4u_4' & u_4v_4' & u_4 & v_4u_4' & v_4v_4' & v_4 & u_4' & v_4' & 1 \\ u_5u_5' & u_5v_5' & u_5 & v_5u_5' & v_5v_5' & v_5 & u_5' & v_5' & 1 \\ u_6u_6' & u_6v_6' & u_6 & v_6u_6' & v_6v_6' & v_6 & u_6' & v_6' & 1 \\ u_7u_7' & u_7v_7' & u_7 & v_7u_7' & v_7v_7' & v_7 & u_7' & v_7' & 1 \\ u_8u_8' & u_8v_8' & u_8 & v_8u_8' & v_8v_8' & v_8 & u_8' & v_8' & 1 \end{pmatrix}}_{\mathbf{W}} \underbrace{\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix}}_{\mathbf{f}} = 0$$

- Homogeneous system $\mathbf{W}\,\mathbf{f} = 0$
- Rank 8 ⟶ A non-zero solution exists (unique)
- If N>8 ⟶ Lsq. solution by SVD! ⟶ $\hat{\mathrm{F}}$ $\|\mathbf{f}\| = 1$

Estimating F

$$p^T \hat{F} p' = 0$$

The estimated $\hat{F}$ may have full rank ($\det(\hat{F}) \neq 0$)
(F should have rank=2 instead)

Find F that minimizes $\|F - \hat{F}\| = 0$

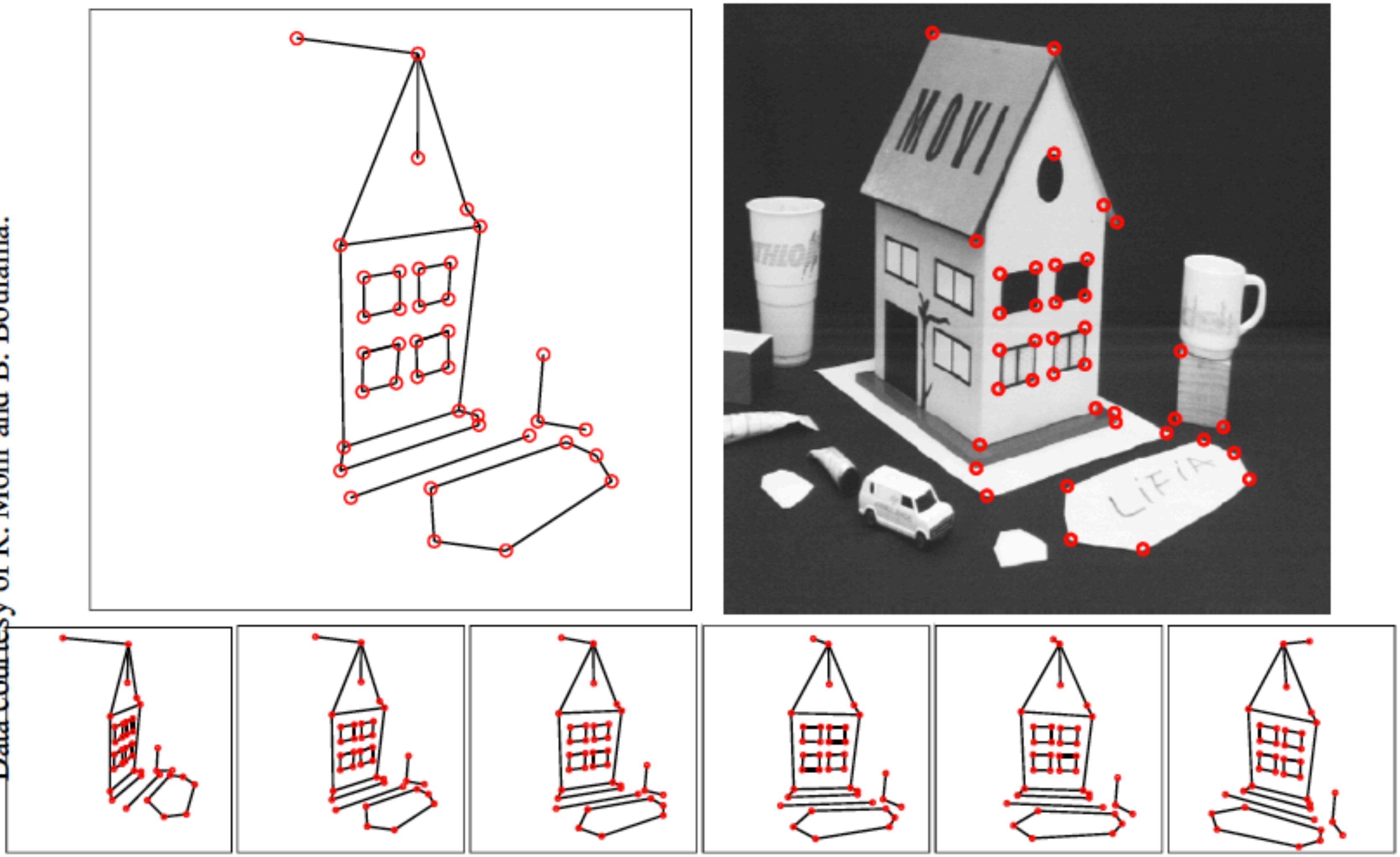
Frobenius norm (*)

Subject to det(F)=0

SVD (again!) can be used to solve this problem

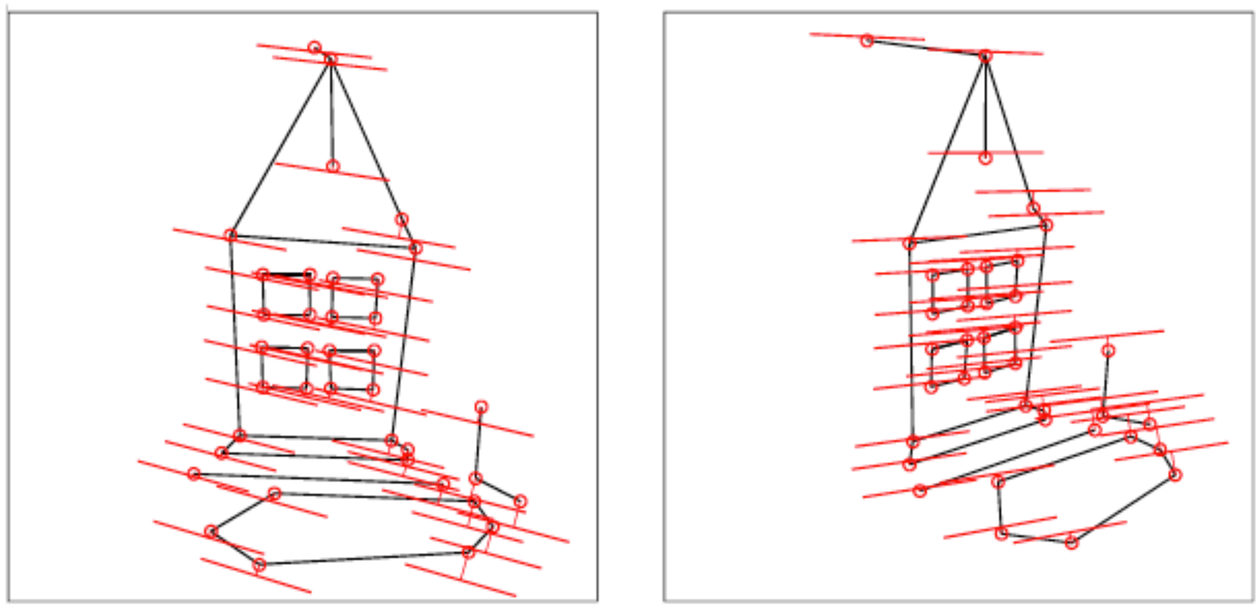
(*) Sqrt root of the sum pf squares of all entries

Example



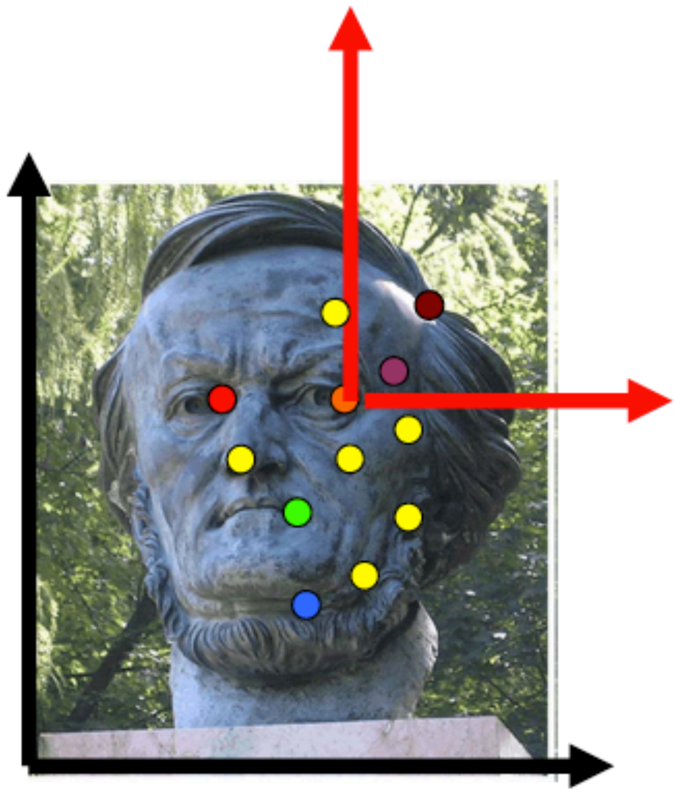
Data courtesy of R. Mohr and B. Boufama.

Example



Mean errors:
10.0pixel
9.1pixel

Normalization



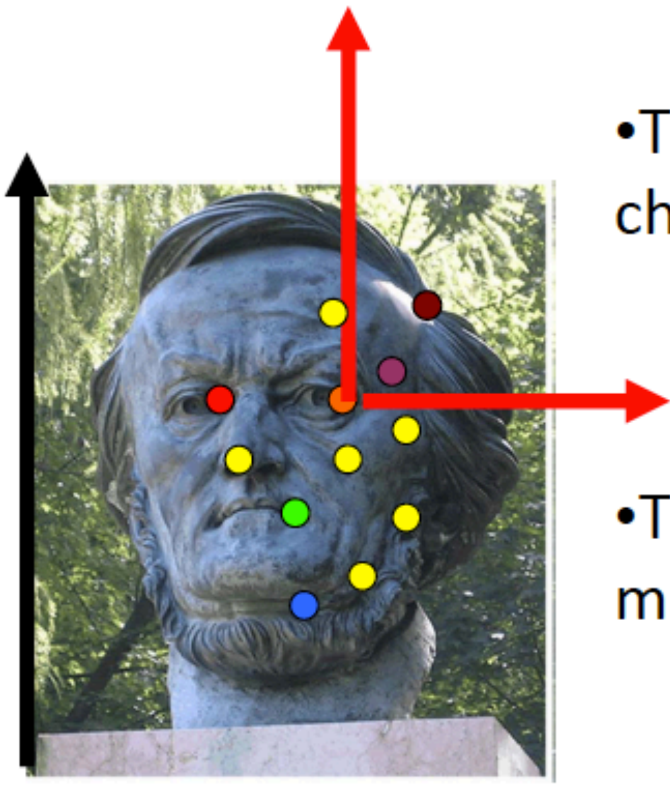
Is the accuracy in estimating F function of the ref. system in the image plane?

E.g. under similarity transformation (T = scale + translation):

$$q_i = T_i \; p_i \qquad q_i' = T_i' \; p_i'$$

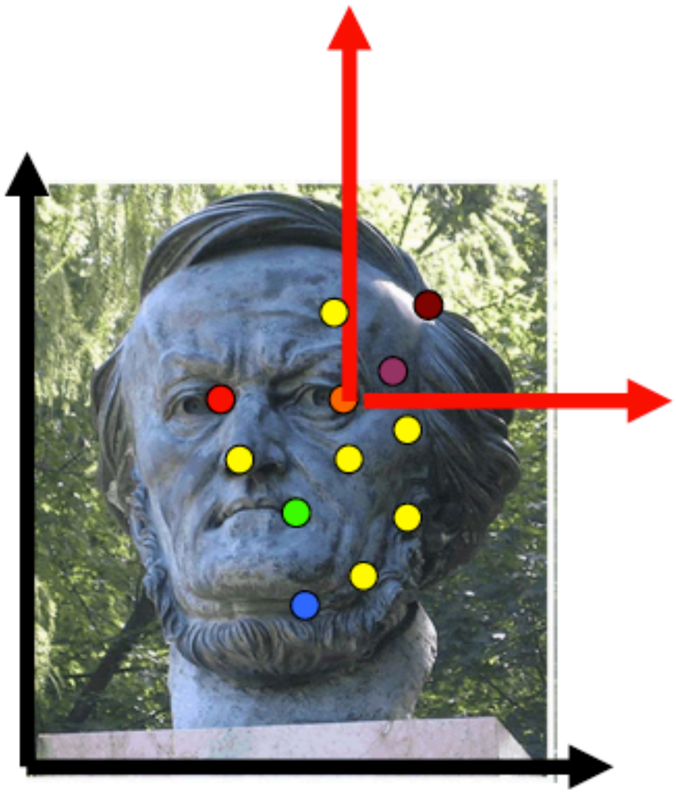
Does the accuracy in estimating F change if a transformation T is applied?

Normalization



- The accuracy in estimating F does change if a transformation T is applied
- There exists a T for which accuracy is maximized

Normalization



$$\mathbf{W}\,\mathbf{f} = 0, \quad \|\mathbf{f}\| = 1 \xrightarrow{\text{Lsq solution by SVD}} \mathrm{F}$$

- SVD enforces Rank(W)=8

- Recall the structure of W:
  Highly un-balance
  (not well conditioned)

-Values of W must have similar magnitude

More details HZ pag 108

The Normalized Eight-Point Algorithm

0. Compute T_i and T_i'

1. Normalize coordinates:

$$q_i = T_i \, p_i \qquad q_i' = T_i' \, p_i'$$

2. Use the eight-point algorithm to compute F'_q from the points q_i and q'_i.

1. Enforce the rank-2 constraint.

$$\rightarrow F_q \qquad \begin{cases} q^T F_q \, q' = 0 \\ \det(F_q) = 0 \end{cases}$$

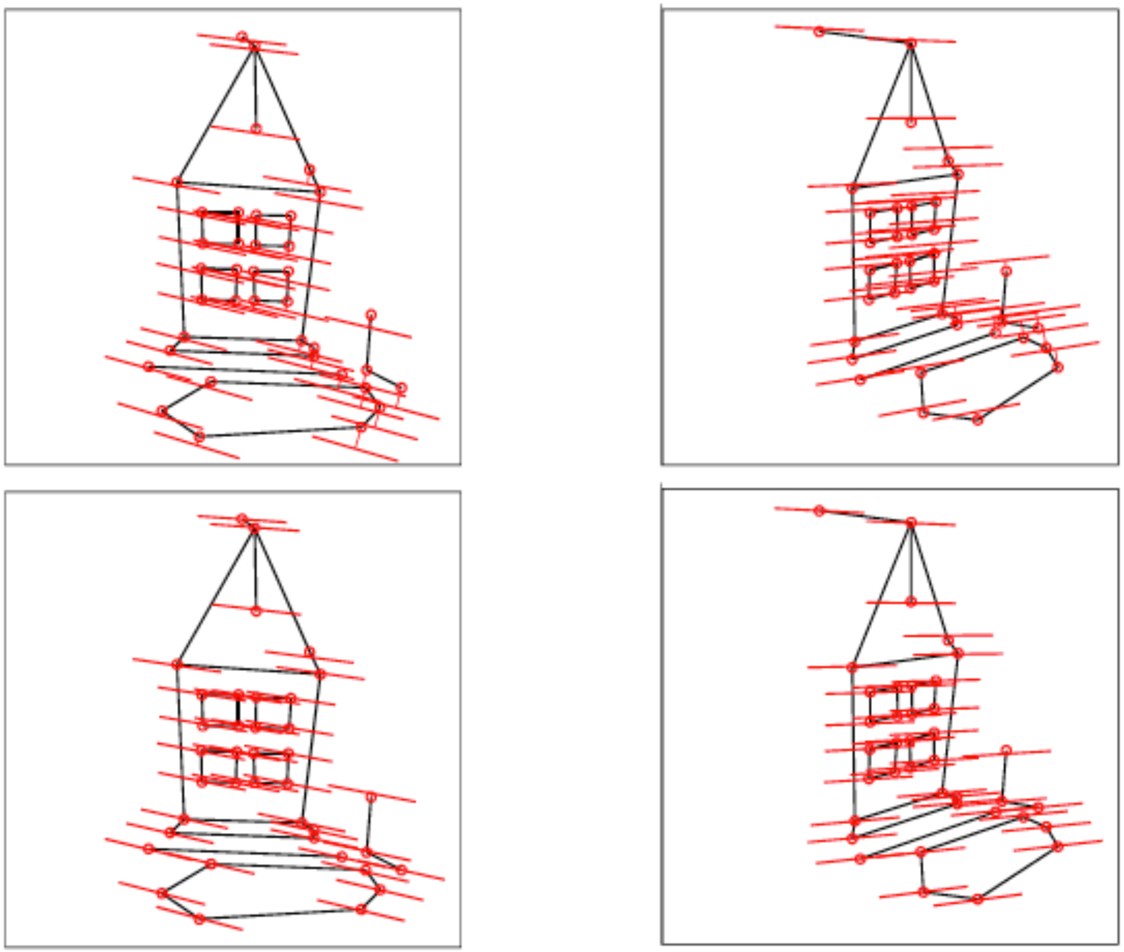
2. De-normalize F_q:

$$F = T'^T F_q T$$

Example

Without transformation

With transformation



Mean errors:
10.0pixel
9.1pixel

Mean errors:
1.0pixel
0.9pixel

RANSAC

Similar approach to homography estimation

Step 1. Extract features

Step 2. Compute a set of potential matches

Step 3. do

- Step 3.1 select minimal sample (i.e. 8 matches)
- Step 3.2 compute solution(s) for F

(generate hypothesis)

- Step 3.3 determine inliers (verify hypothesis)

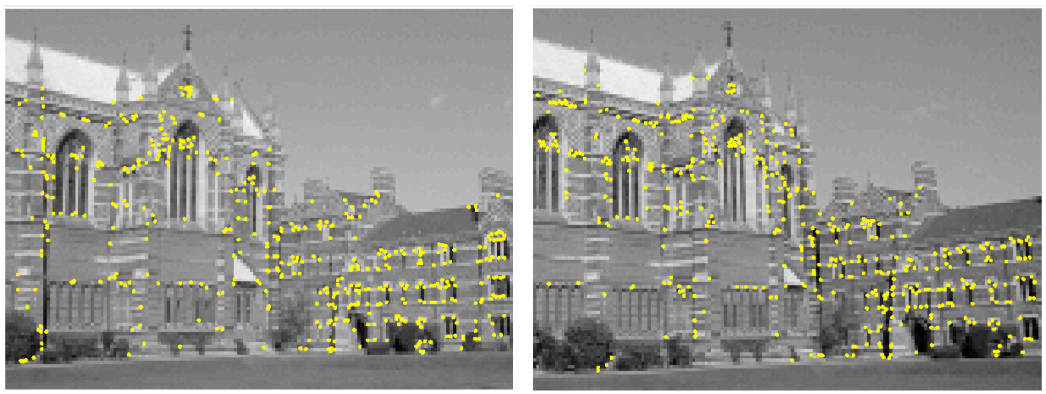
until (enough inliers)

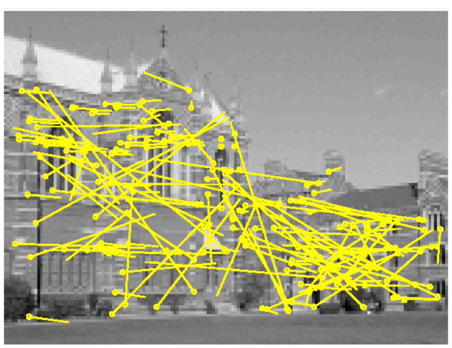
Step 4. Compute F based on all inliers

Step 5. Look for additional matches

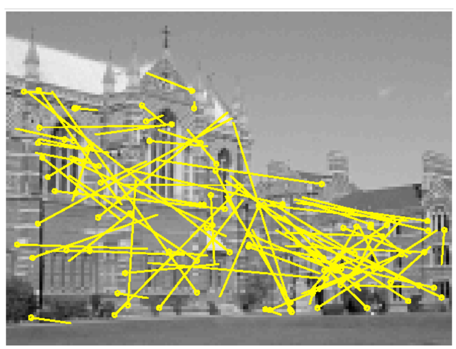
Step 6. Refine F based on all correct matches

Example

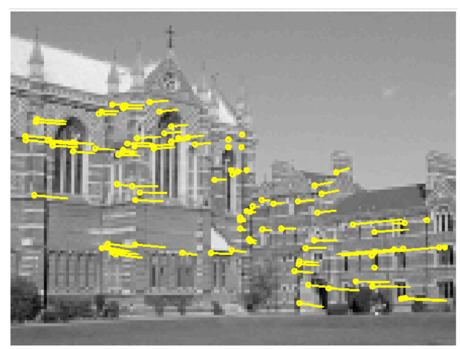




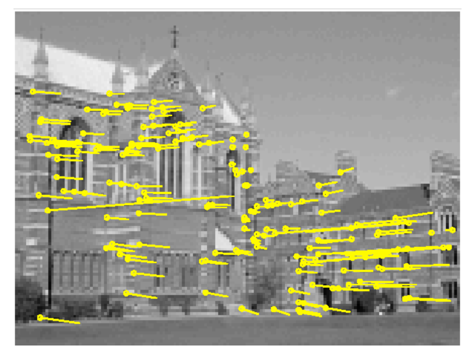
All matches



Outliers



Inliers



Re-estimation

Back to triangulation

(using a pair of corresponding points)

A few cases

- Fully calibrated pair of cameras
  - use e.g. DLT as done earlier in class
- Intrinsics (K) are known
  - Compute E (from F or directly from point correspondences)
  - Compute (R,T) from E [see RS or HZ]
  - Triangulate

Some Facts about E and F

E has ?? degrees of freedom?

E q = l → a projective line (also q' E)

The vector e s.t. Ee=0 is an epipole; likewise e'E = 0

SVD(E) = U D V^t where D= [s, s, 0] for any s > 0

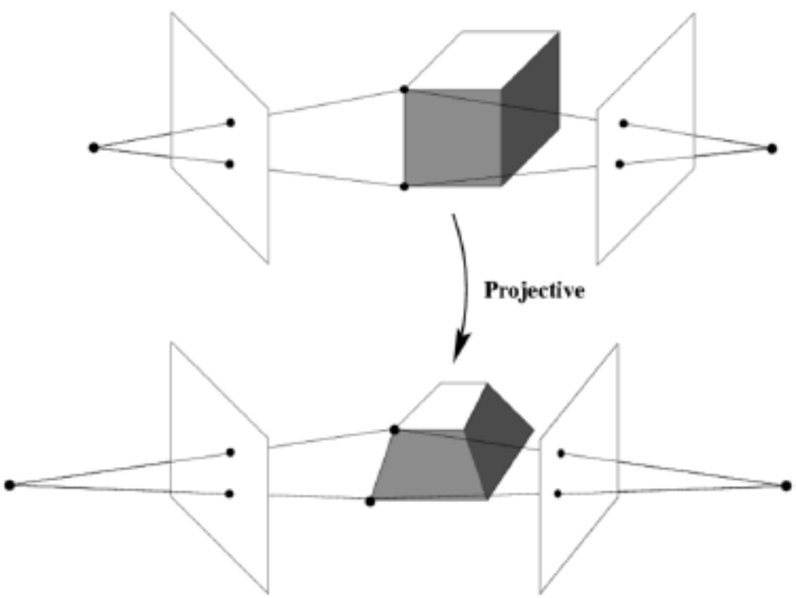
Both E and F are singular (why?) so det(E) = det(F) = 0

Both are only defined up to scale

F has ?? degrees of freedom

Uncalibrated cameras

- Compute F from correspondences (=weak calibration)
    - F provides constraints only up to a 3D projective transformation
    - More information about scene necessary for affine/metric reconstruction
        - known correspondences between 3D reconstructed points and world
        - or scene/camera constraints (e.g. parallel lines)
    - [See later in class: stratified reconstruction]



$$x_j = M_i \boxed{X_j} \qquad \boxed{M_i} = K_i\begin{bmatrix} R_i & T_i \end{bmatrix}$$

$$\boxed{H\,X_j} \qquad \boxed{M_j\,H^{-1}}$$

$$x_j = M_i\,X_j = \left(M_i\,H^{-1}\right)\left(H\,X_j\right)$$

Summary

Depth from stereo: main idea is to triangulate from corresponding image points.

Epipolar geometry defined by two cameras

- Epipolar constraint limits where points from one view will be imaged in the other
- Makes search for correspondences quicker

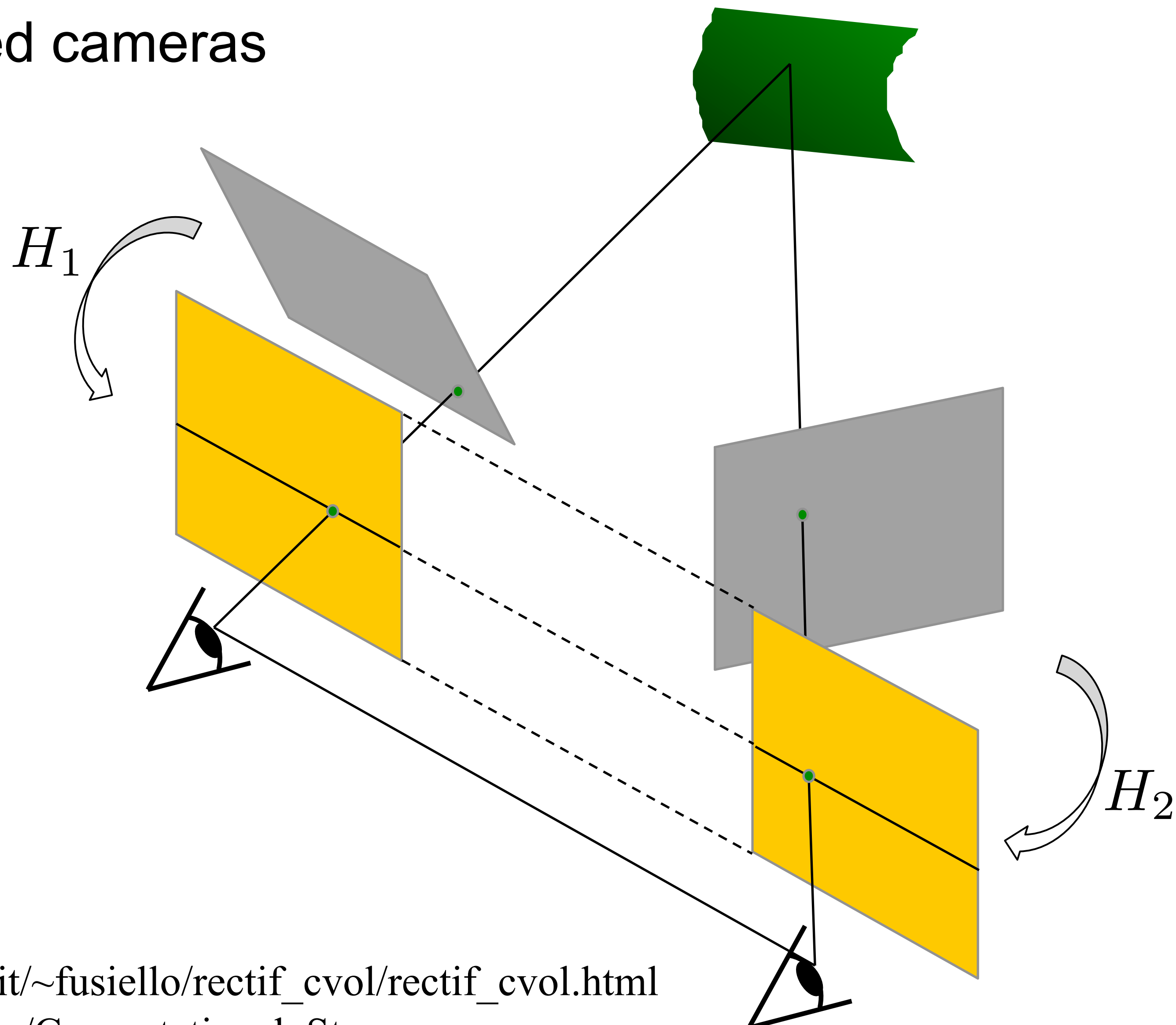
Terms: epipole, epipolar plane / lines, disparity, rectification, intrinsic/extrinsic parameters, essential matrix, fundamental matrix, baseline

Stereo Matching

Rectification

Using calibrated cameras

-> Board



References:
http://profs.sci.univr.it/~fusiello/rectif_cvol/rectif_cvol.html
https://cirl.lcsr.jhu.edu/Computational_Stereo

$$\mathbf{p_r}^{\mathbf{t}}\, \mathbf{E}\, \mathbf{p_l} = \mathbf{0}$$

There are four relative poses consistent with E up to sign

$E = U\ D\ V^t$

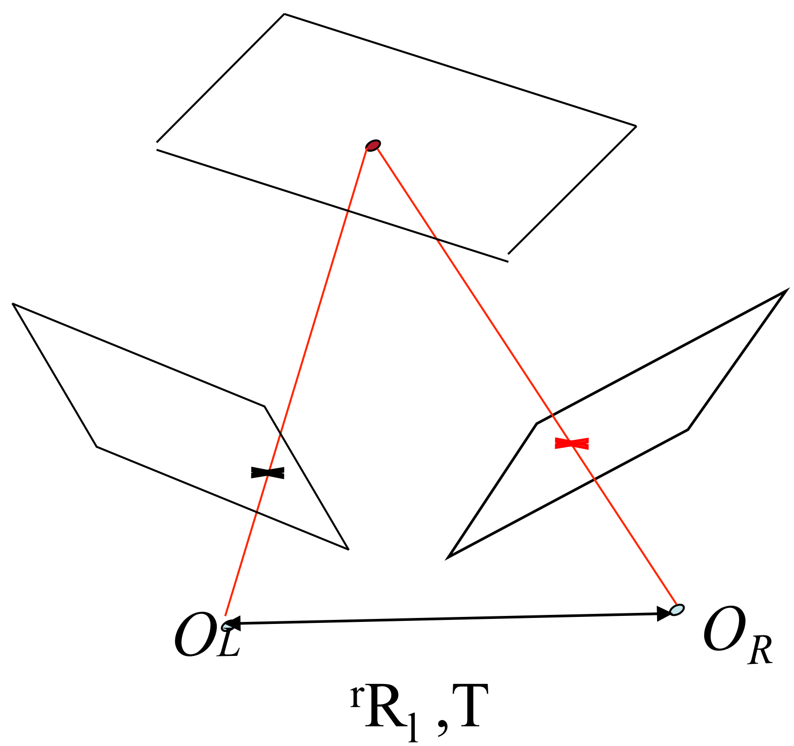
$(sk(T_1), R_1) = (U\ R_Z(+\pi/2)\ D\ U^t\ ,\ V\ R_Z(+\pi/2)\ V^t)$
$(sk(T_2), R_2) = (U\ R_Z(-\pi/2)\ D\ U^t\ ,\ V\ R_Z(-\pi/2)\ V^t)$

$D = diag(1, 1, 0)$

The four possible solutions (including sign of T) can be disambiguated by ensuring all points are in front of the cameras

Rectification: How?

- Goal: Turn a verged camera system into a non-verged system with the same focal lengths.
- Assumption: We know the calibration of the 2 camera setup,ie: intrinsic parameters and the extrinsics (${}^{r}R_{l}$,T).



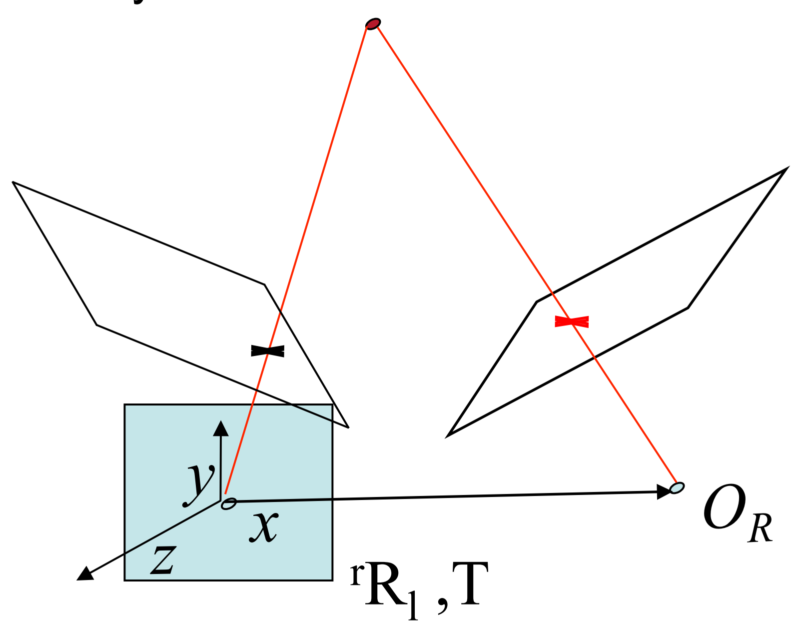
Rectification: How?

- New coordinate system for rectified image planes:
  - X axis: baseline (translation between two optical centers)

    x = T/||T||
  - y axis is orthogonal to x ([0,0,1] is the z axis of the current left camera)

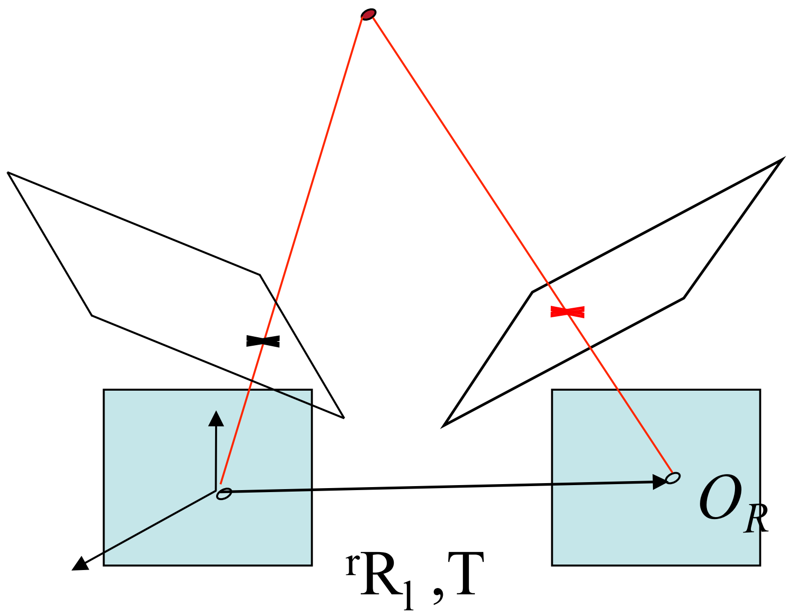
y = [0,0,1] cross x/|| [0,0,1] cross x ||

  - z = x cross y



Rectification: How?

- Both the left and right camera can now be rotated to be parallel to this frame
  - ${}^{l}R_u$ = [x y z] is the rotation from unverged to verged frame for left camera
  - ${}^{r}R_u = {}^{r}R_l\,{}^{l}R_u$

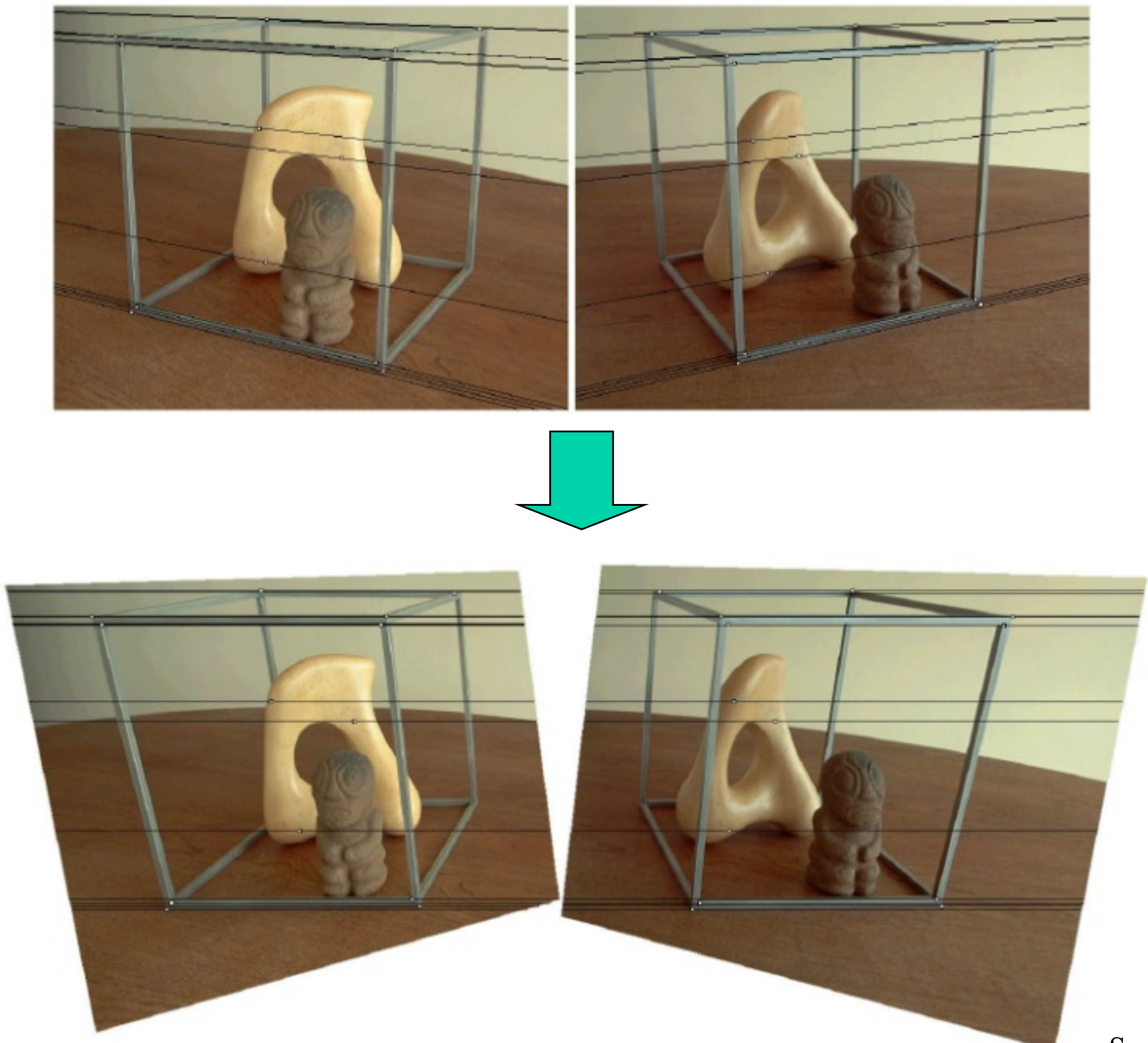


Rectification: Basic Algorithm

1. Create a mesh of pixel coordinates for the rectified image
2. Turn the mesh into a list of homogeneous points
3. Project *backwards* through the intrinsic parameters to get unit focal length values
   a) note that we can choose some of these parameters if we want
   b) typical to choose scale factors to be the same (simpler)
   c) typical to choose the image centers to allow “centering” the rectified images
   d) note that both y centers must be identical!
4. Rotate these values back to the current camera coordinate system.
5. Project them *forward* through the intrinsic parameters to get pixel coordinates again.
6. Sample at these points to populate the rectified image.

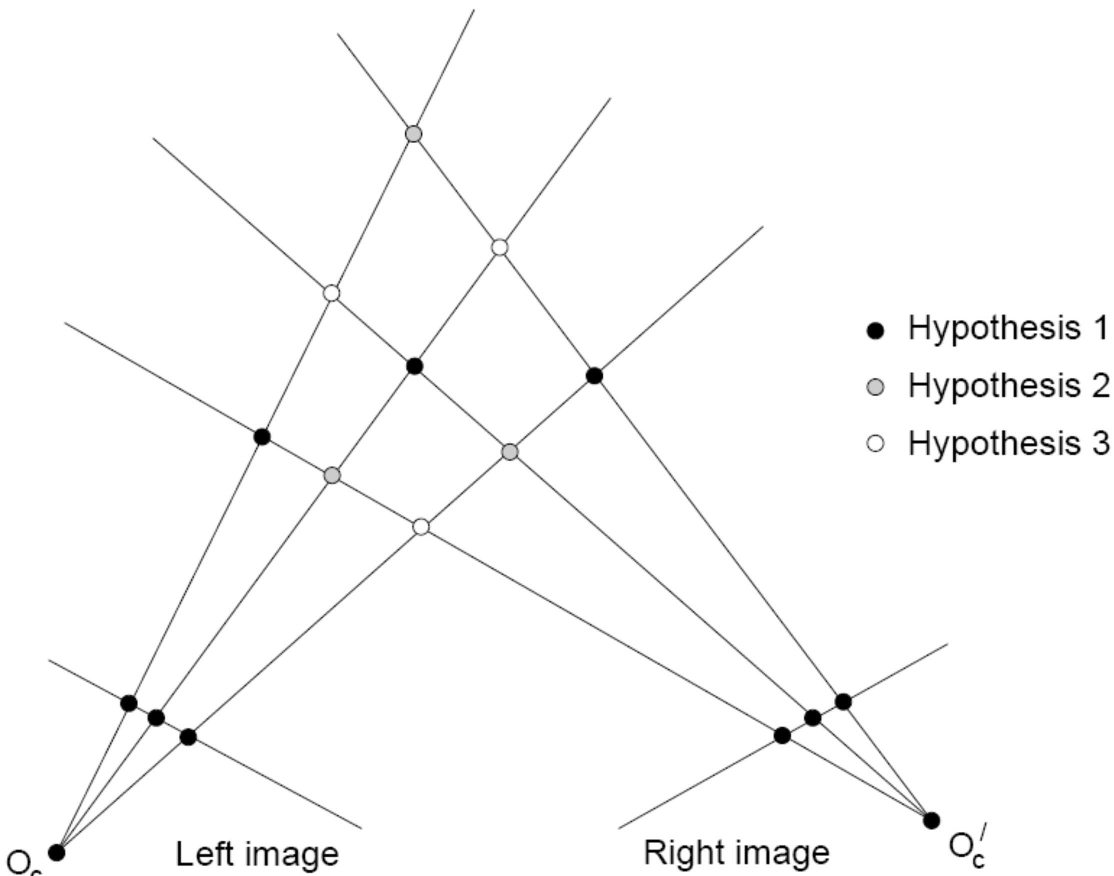
Depth from Stereo

Stereo image rectification: example



Source: Alyosha Efros

Correspondence problem



Multiple match hypotheses satisfy epipolar constraint, but which is correct?

Figure from Gee & Cipolla 1999

Correspondence problem

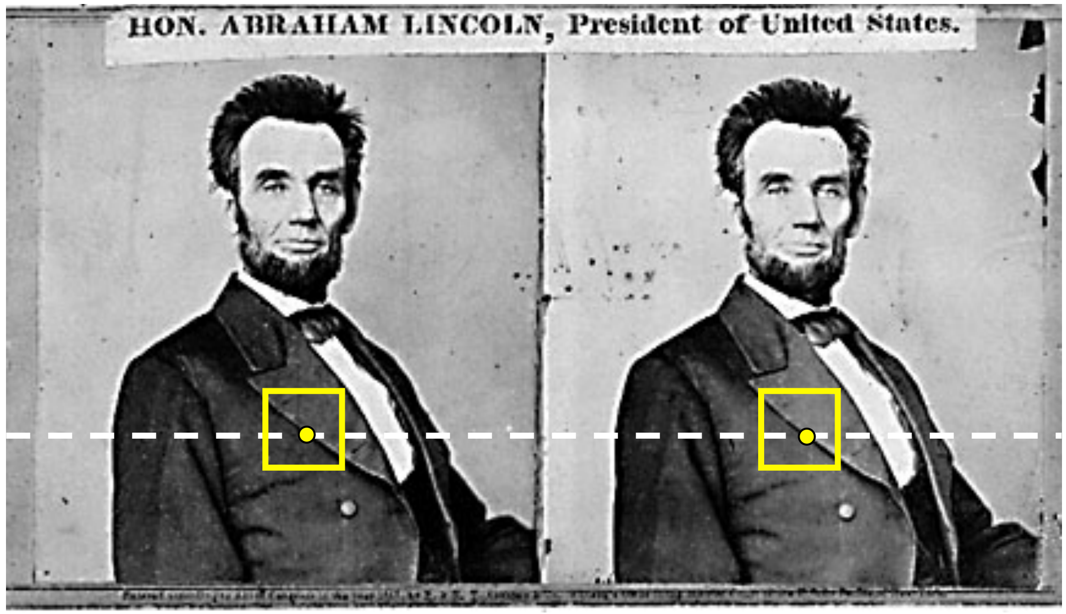
Beyond the hard constraint of epipolar geometry, there are "soft" constraints to help identify corresponding points

- Similarity
- Uniqueness
- Ordering
- Disparity gradient

To find matches in the image pair, we will assume

- Most scene points visible from both views
- Image regions for the matches are similar in appearance

Dense correspondence search



For each epipolar line

For each pixel / window in the left image

- compare with every pixel / window on same epipolar line in right image
- pick position with minimum match cost (e.g., SSD, correlation)

Adapted from Li Zhang

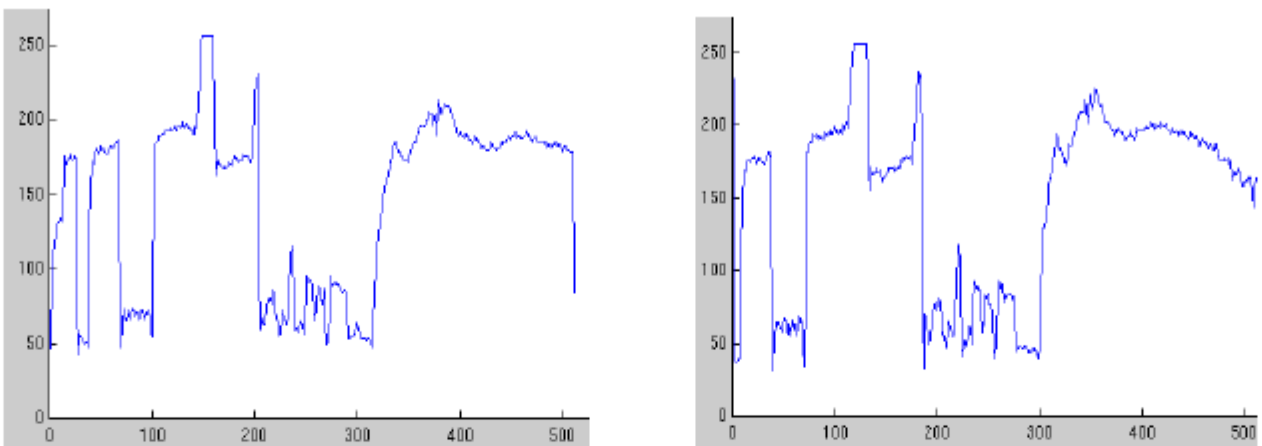
Correspondence problem

Parallel camera example: epipolar lines are corresponding image scanlines

Source: Andrew Zisserman

Correspondence problem

Intensity profiles



- Clear correspondence between intensities, but also noise and ambiguity

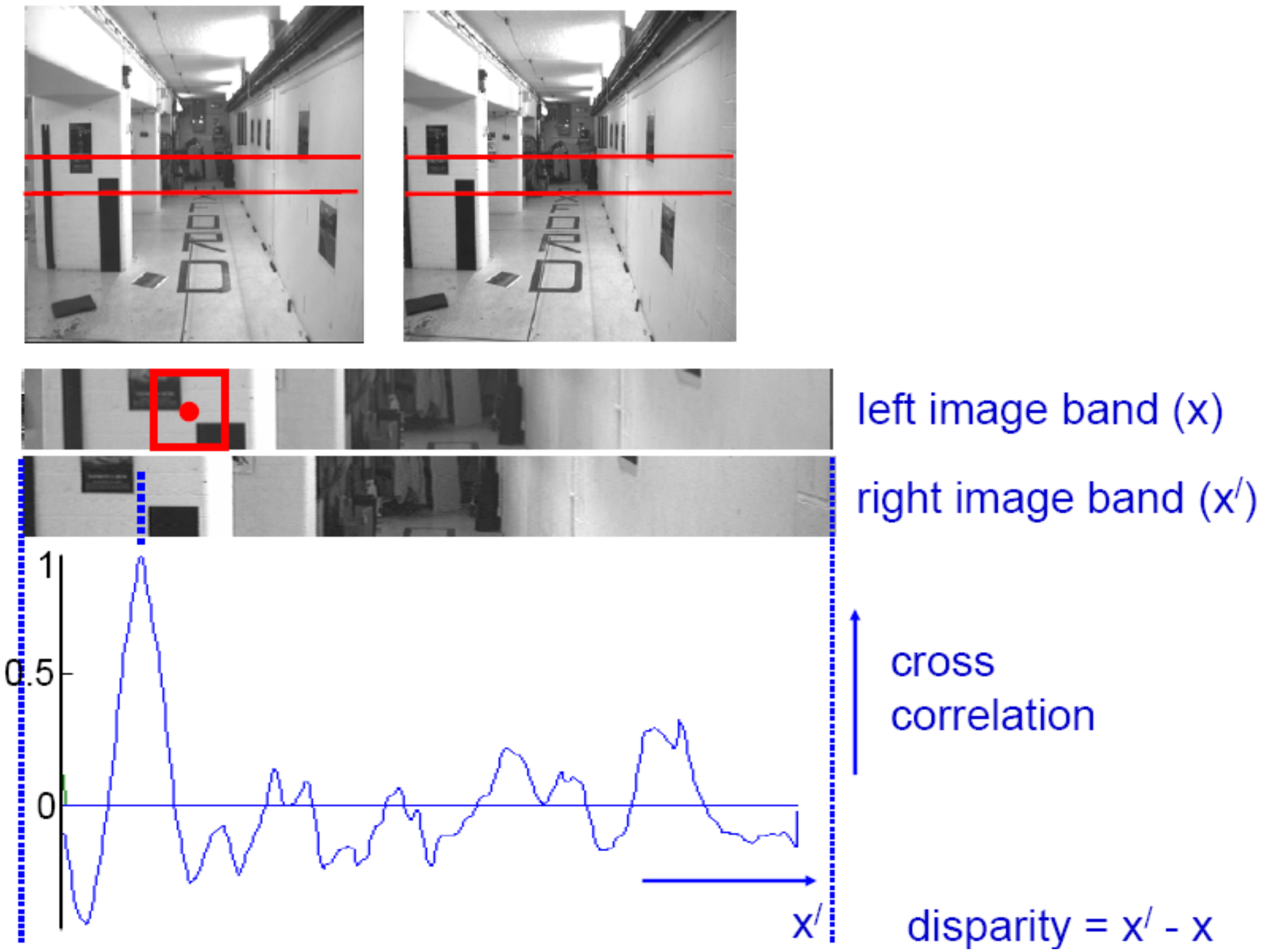
Source: Andrew Zisserman

Correspondence problem

Neighborhoods of corresponding points are similar in intensity patterns.

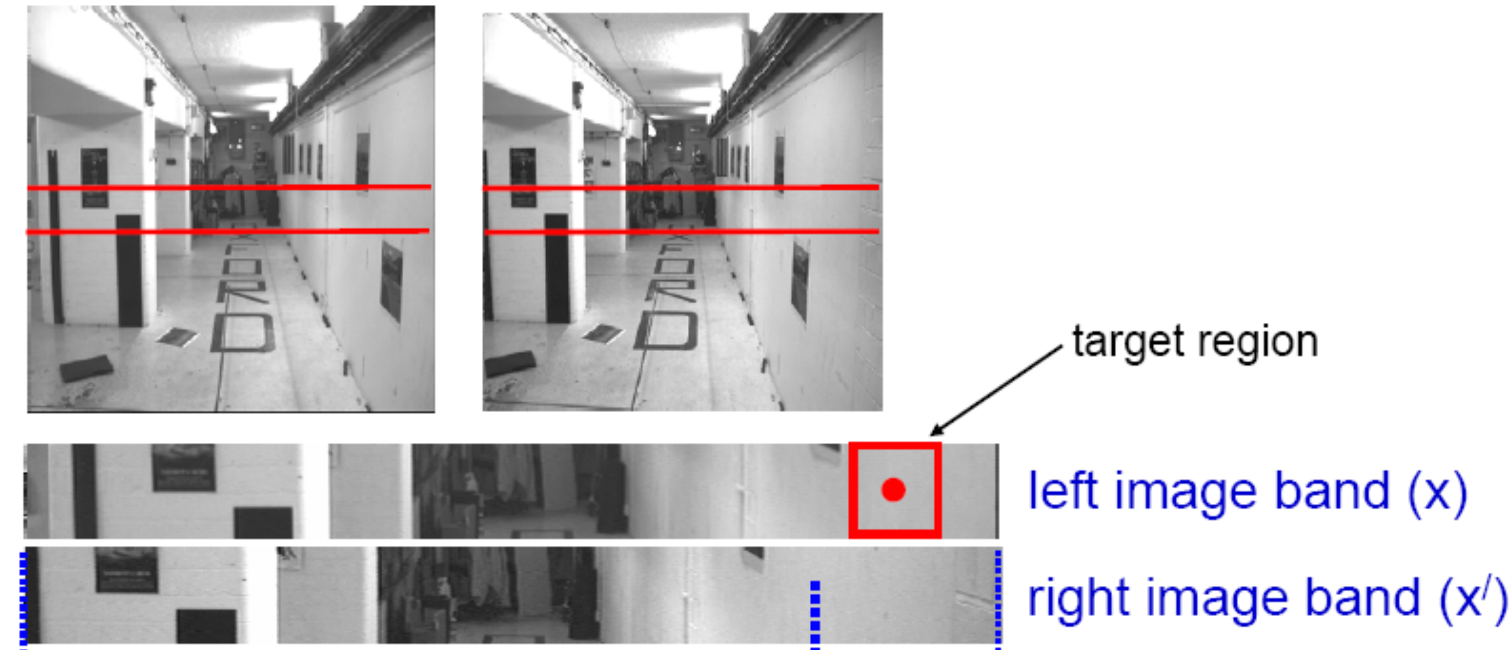
Source: Andrew Zisserman

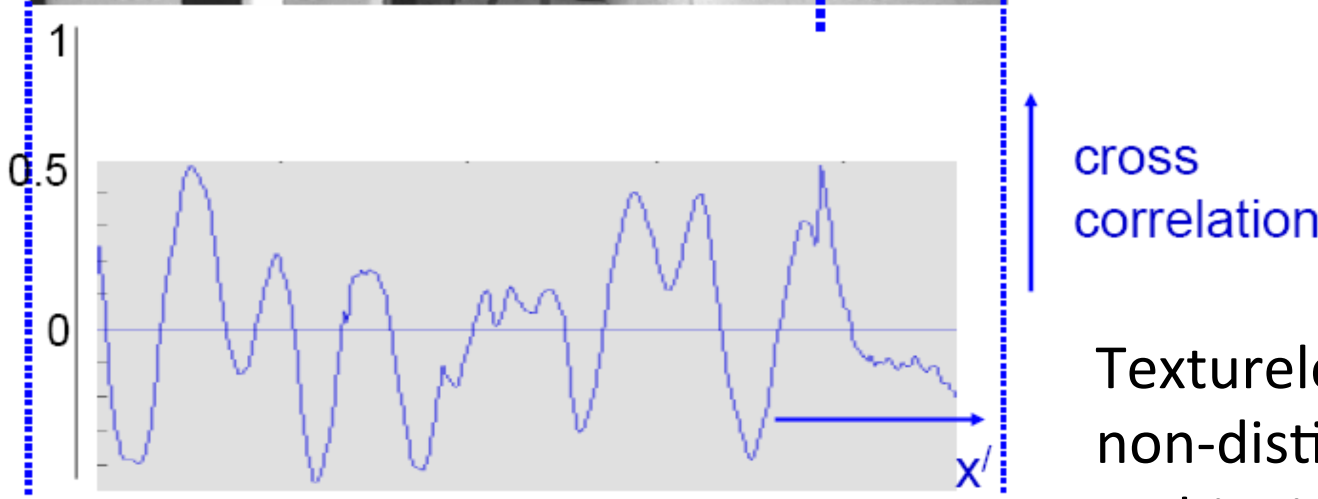
Correlation-based window matching



Source: Andrew Zisserman

Textureless regions

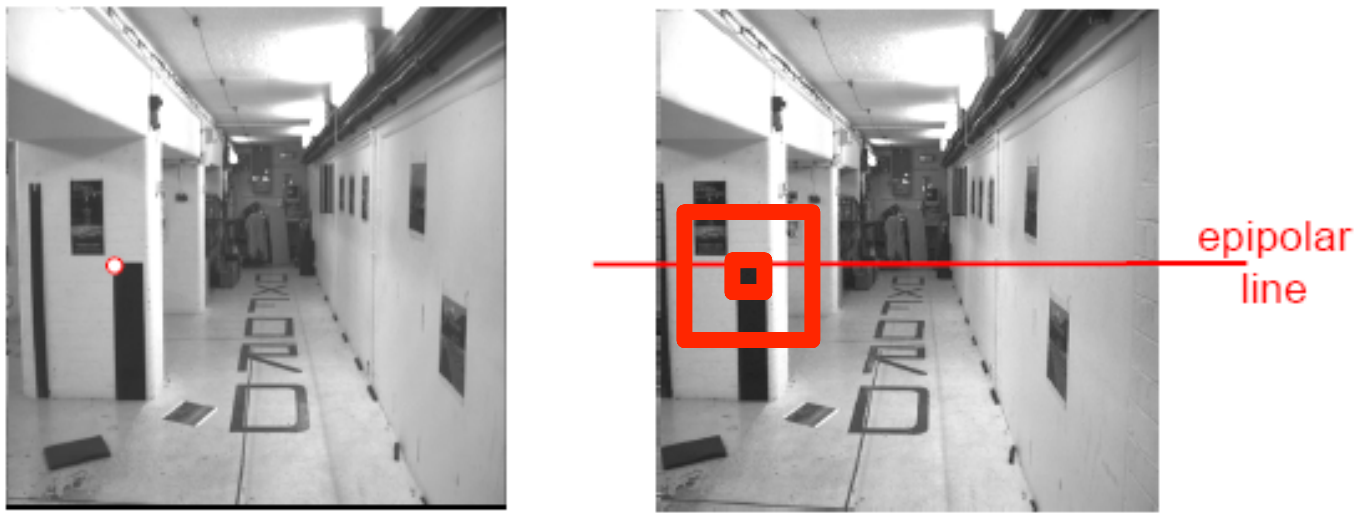




Textureless regions are non-distinct; high ambiguity for matches.

Source: Andrew Zisserman

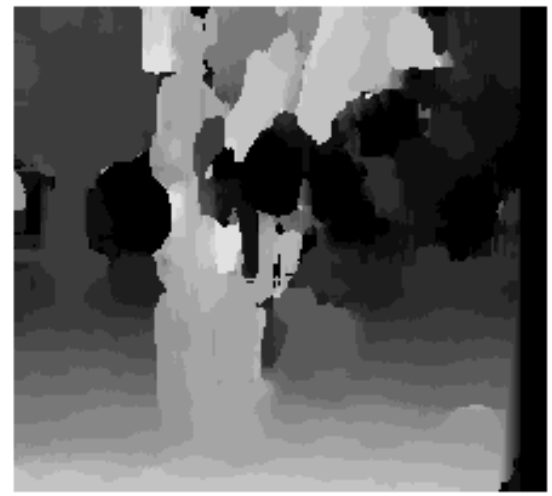
Effect of window size



Source: Andrew Zisserman

Effect of window size

W = 3



W = 20

Want window large enough to have sufficient intensity variation, yet small enough to contain only pixels with about the same disparity.

Figures from Li Zhang

Match Metric Summary

MATCH METRIC	DEFINITION
Normalized Cross-Correlation (NCC)	$\dfrac{\sum_{u,v}\left(I_1(u,v)-\bar{I}_1\right)\cdot\left(I_2(u+d,v)-\bar{I}_2\right)}{\sqrt{\sum_{u,v}\left(I_1(u,v)-\bar{I}_1\right)^2\cdot\sum_{u,v}\left(I_2(u+d,v)-\bar{I}_2\right)^2}}$
Normalized SSD	$\sum_{u,v}\left(\dfrac{\left(I_1(u,v)-\bar{I}_1\right)}{\sqrt{\sum_{u,v}\left(I_1(u,v)-\bar{I}_1\right)^2}}-\dfrac{\left(I_2(u+d,v)-\bar{I}_2\right)}{\sqrt{\sum_{u,v}\left(I_2(u+d,v)-\bar{I}_2\right)^2}}\right)^2$
Sum of Squared Differences (SSD)	$\sum_{u,v}\left(I_1(u,v)-I_2(u+d,v)\right)^2$
Sum of Absolute Differences (SAD) Zero Mean SAD	$\sum_{u,v}\left\|I_1(u,v)-I_2(u+d,v)\right\|$ $\sum_{u,v}\left\|\left(I_1(u,v)-\bar{I}_1\right)-\left(I_2(u+d,v)-\bar{I}_2\right)\right\|$
Rank	$I'_k(u,v)=\sum_{m,n}I_k(m,n)<I_k(u,v)$ $\sum_{u,v}\left(I'_1(u,v)-I'_2(u+d,v)\right)$
Census	$I'_k(u,v)=BITSTRING_{m,n}\left(I_k(m,n)<I_k(u,v)\right)$ $\sum_{u,v}HAMMING\left(I'_1(u,v),I'_2(u+d,v)\right)$

Remember, these two are actually the same

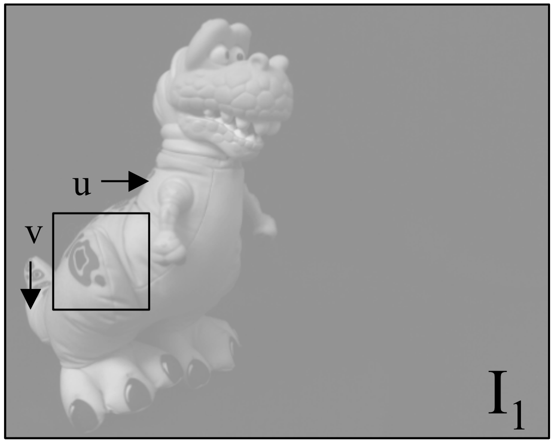
Implementing Region Based Matching

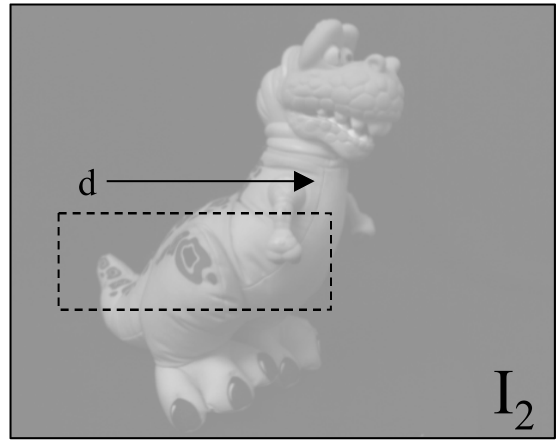
- **Deciding the Range**: The first step in correspondence search is to compute the range of disparities to search
  - Horopter
- Assume a non-verged (rectified) system. Therefore, we have
  - $d = f\, b/z$
  - given a range z_{min} to z_{max}, we calculate
    - $d_{min} = f\, b / z_{max}$
    - $d_{max} = f\, b / z_{min}$
- Thus, for each point u_l in the left image, we will search points $u_l + d_{min}$ to $u_l + d_{max}$ in the right.
- Note we can turn this around and start at a point u_r and search from $u_r - d_{max}$ to $u_r - d_{min}$

Region Based Algorithm

- For each pixel (i,j) of the left image and offset o in disparity range
  - Compute $d(i,j,o) = \sum_{k,l} \psi(I_l(i+k,j+l),I_r(i+k,j+l+o))$
  - The disparity at (i,j) is the value o that minimizes d
- The result of performing this search over every pixel is the *disparity map.*

Correspondence Search Algorithm (simple version for CC)






```
For  i = 1:nrows
   for j=1:ncols
                best(i,j) = -1
                for k = mindisparity:maxdisparity
                   c = CC(I1(i,j),I2(i,j+k),winsize)
                   if (c > best(i,j))
                                best(i,j) = c
                                disparities(i,j) = k
                 end
                 end
   end
end
```

multiplies: O(nrows * ncols * disparities * winx * winy)

Correspondence Search Algorithm (efficient version for CC)

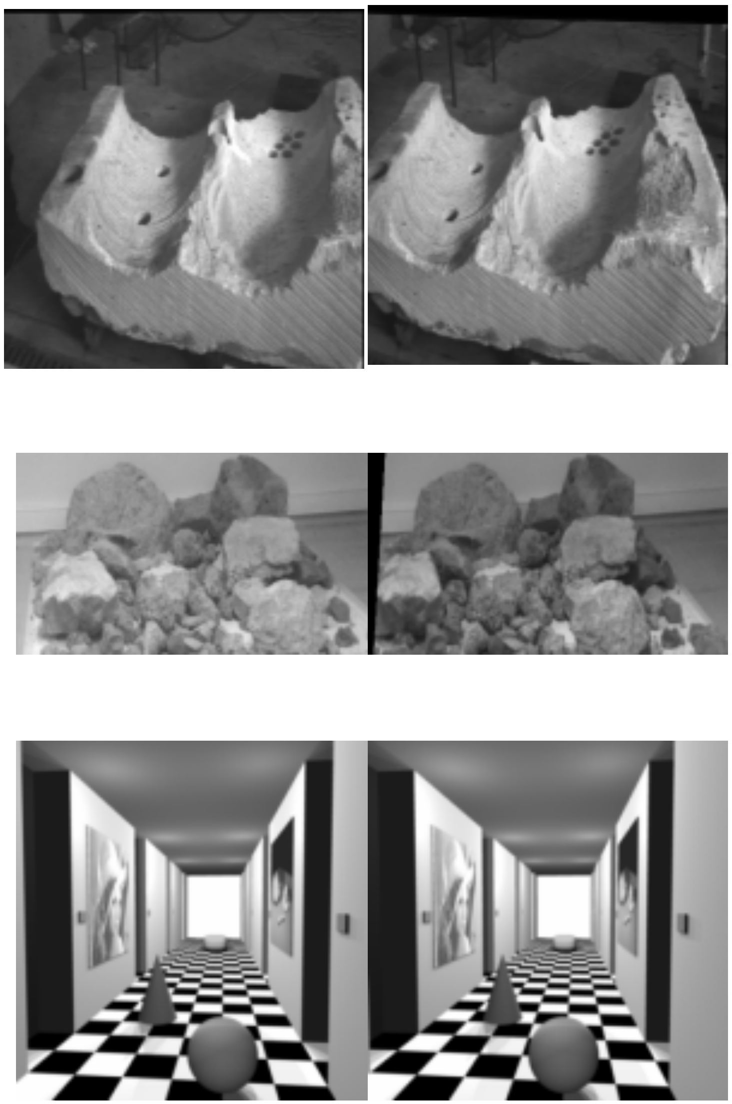
```
best = -ones(size(im))
disp = zeros(size(im))
for k = mindisparity:maxdisparity
        prod = I1(:,overlap) .* I2(:,k+overlap)
        CC = conv2(prod,fspecial('average',winsize))
        better = CC > best;
        disp = better .* k + (1-better).*disp;
        best = better .*CC + (1-better).*best;
end
```

multiplies: O(disparities * nrows * ncols)

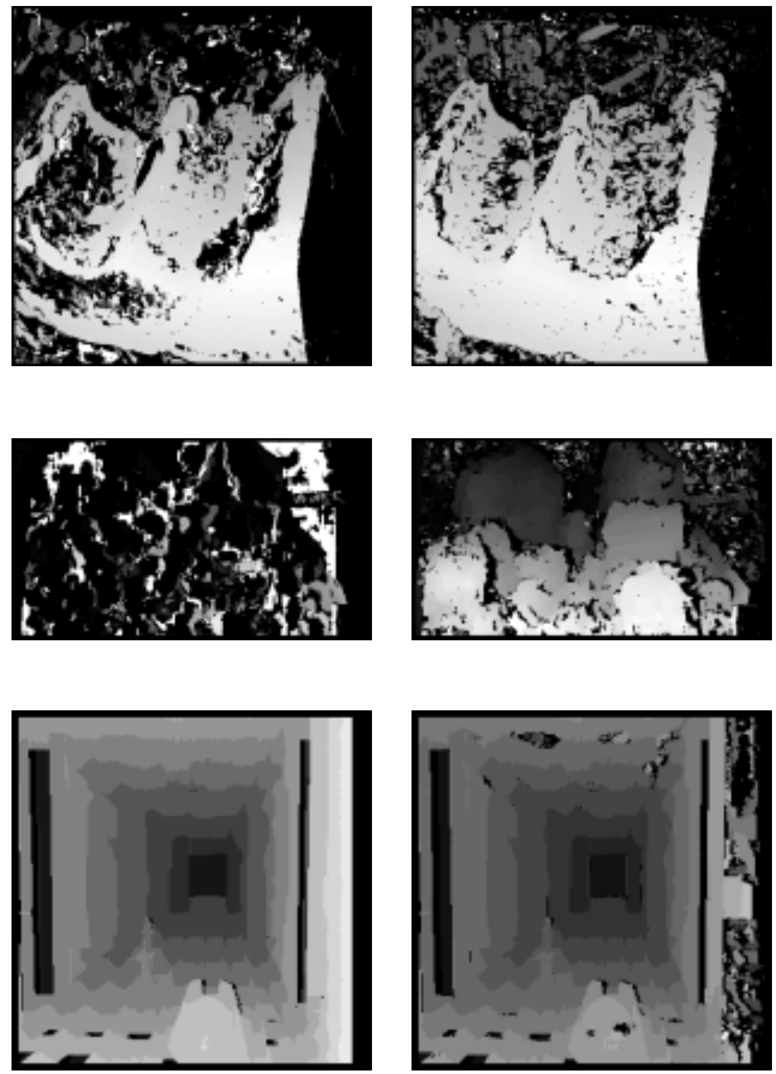
An Additional Twist

- Note that searching from left to right *is not the same* as searching from right to left.
- As a result, we can obtain a somewhat independent disparity map by flipping the images around.
- The results should be the same map up to sign.
- LRCheck: $disp_{lr}(i,j) = - disp_{rl}(i,j+disp_{lr}(i,j))$

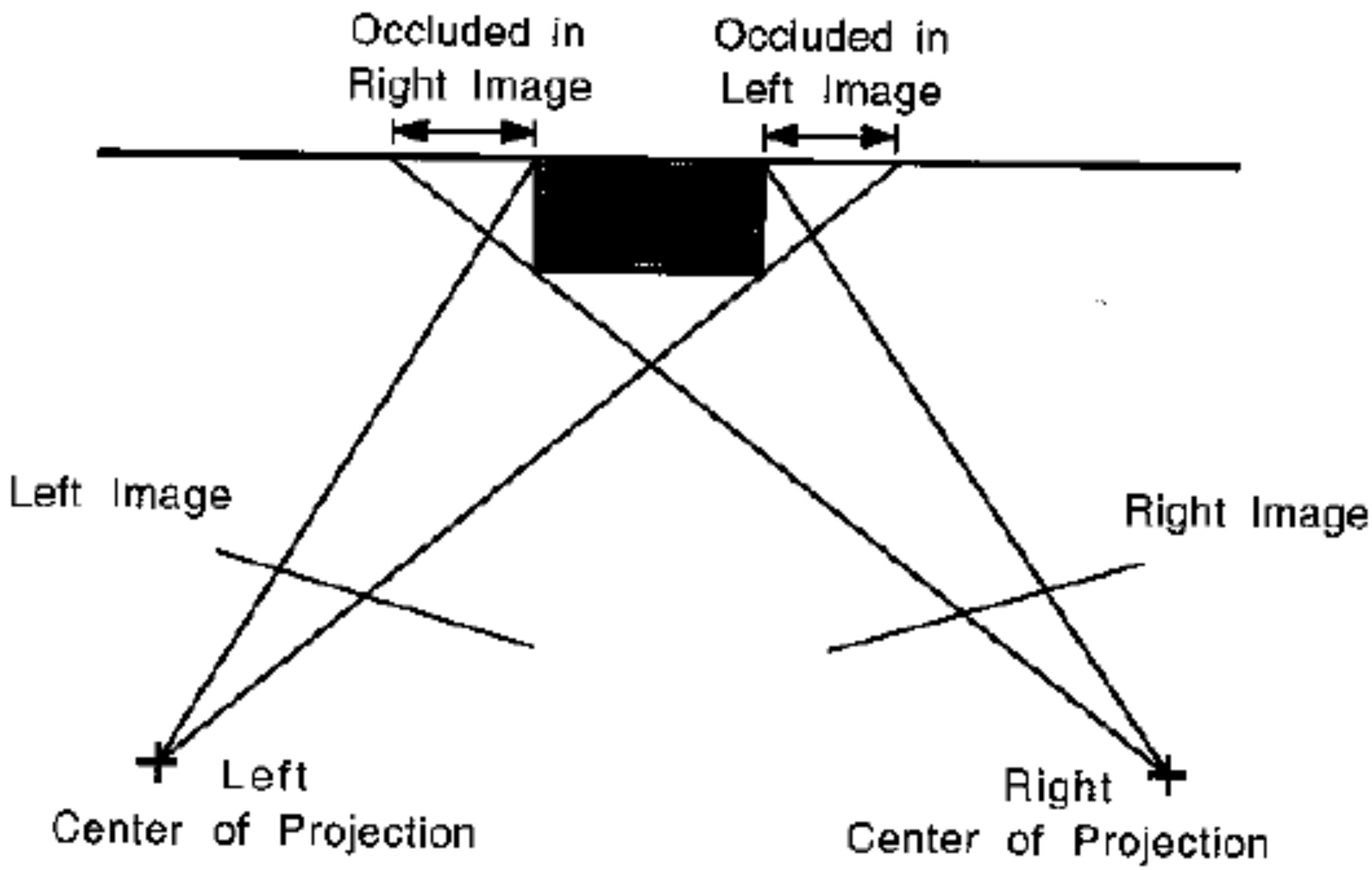
Example Disparity Maps



SSD ZNNC

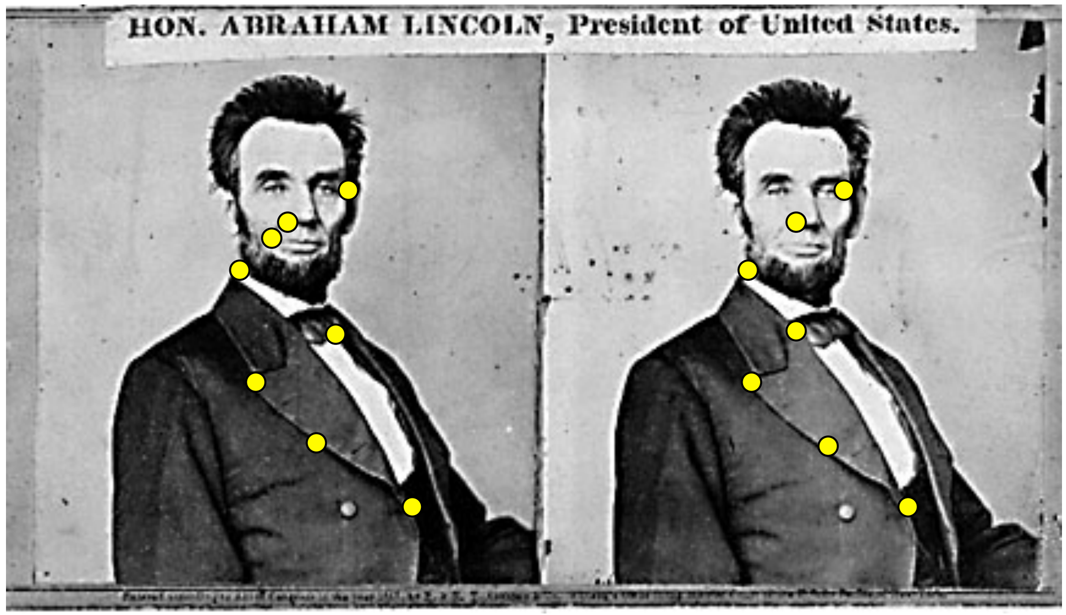


Occlusion



Slide credit: David Kriegman

Sparse correspondence search



Restrict search to sparse set of **detected features** (e.g., corners)

Rather than pixel values (or lists of pixel values) use *feature descriptor* and an associated *feature distance*

Still narrow search further by epipolar geometry

Tradeoffs between dense and sparse search?

Correspondence problem

Beyond the hard constraint of epipolar geometry, there are "soft" constraints to help identify corresponding points

- Similarity
- Uniqueness
- Disparity gradient
- Ordering

Uniqueness constraint

Up to one match in right image for every point in left image

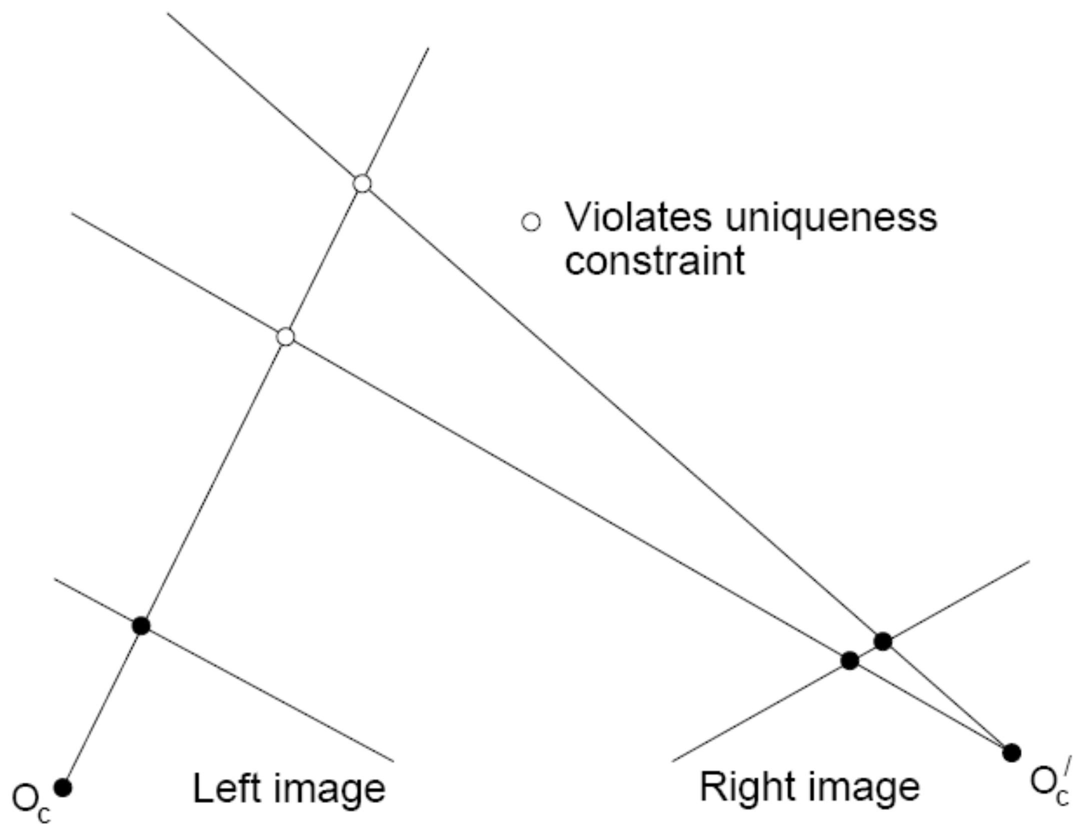
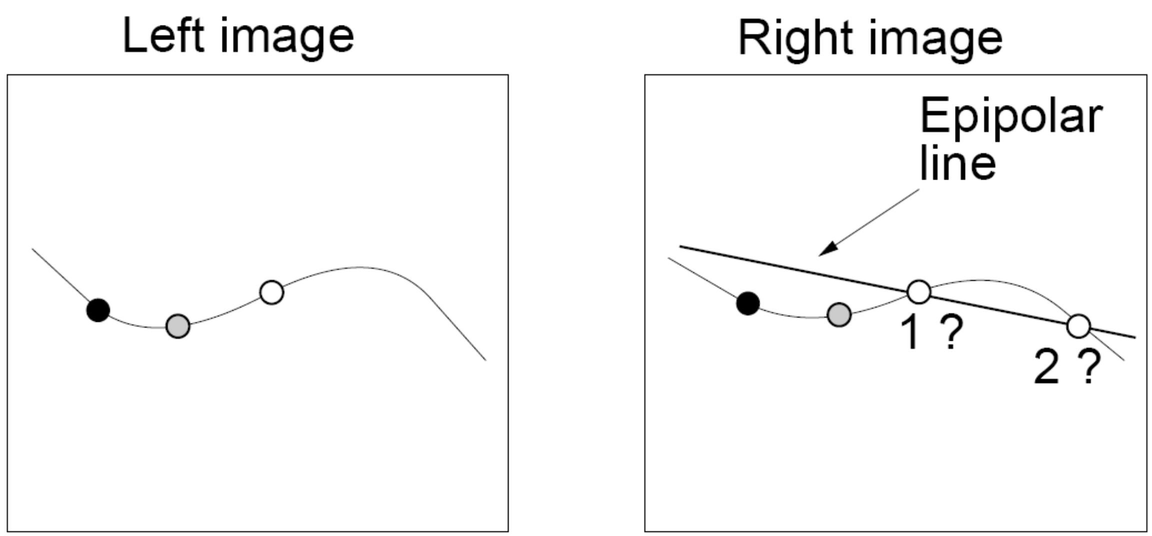


Figure from Gee & Cipolla 1999

Disparity gradient constraint

Assume piecewise continuous surface, so want disparity estimates to be locally smooth



Given matches ● and ◍, point ○ in the left image must match point 1 in the right image. Point 2 would exceed the disparity gradient limit.

Figure from Gee & Cipolla 1999

Ordering constraint

Points on **same surface** (opaque object) will be in same order in both views

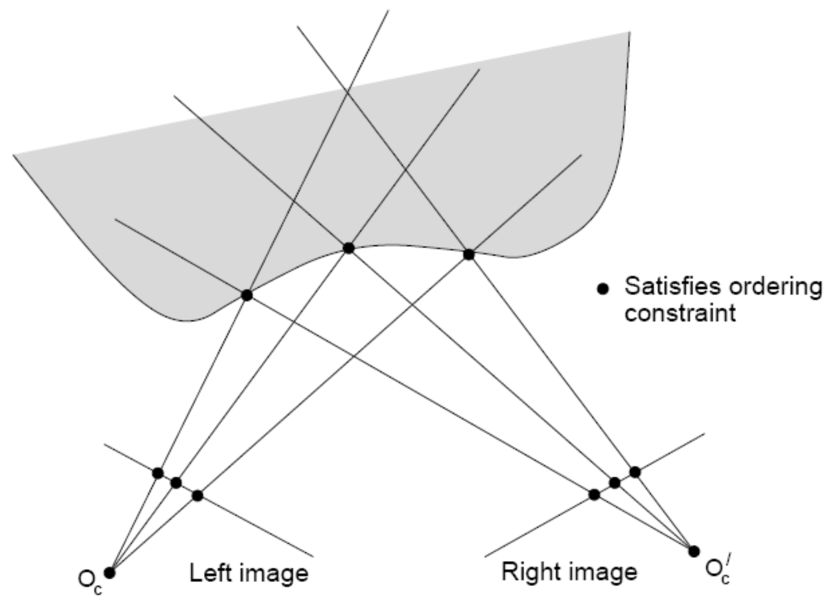
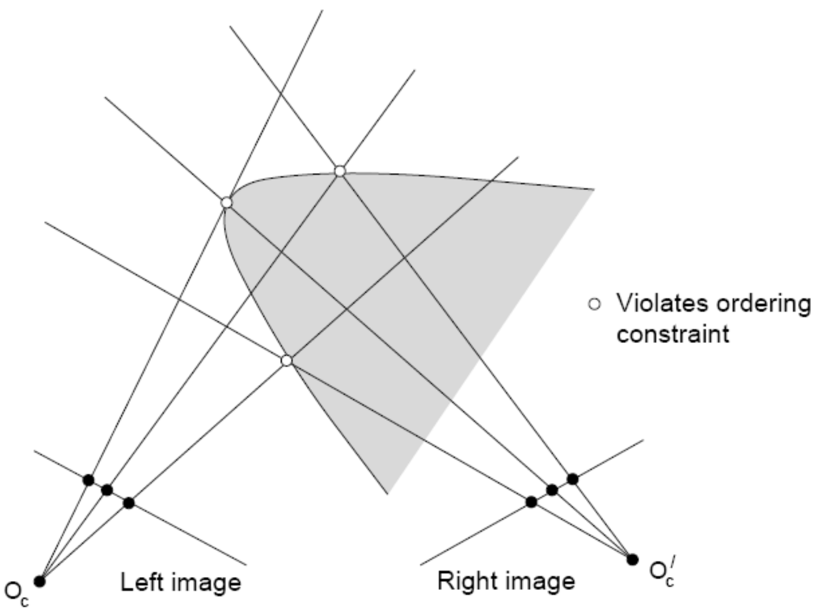
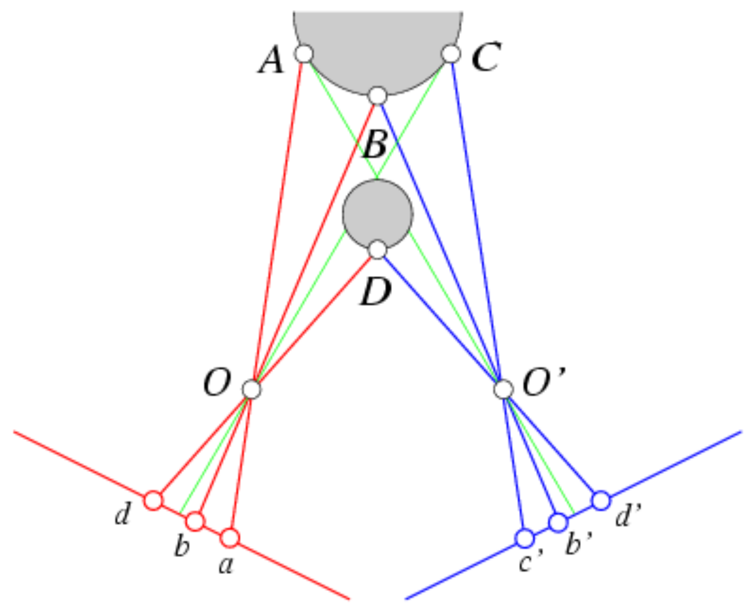


Figure from Gee & Cipolla 1999

Ordering constraint

- Won't always hold, e.g. consider transparent object, or an occluding surface





Figures from Forsyth & Ponce

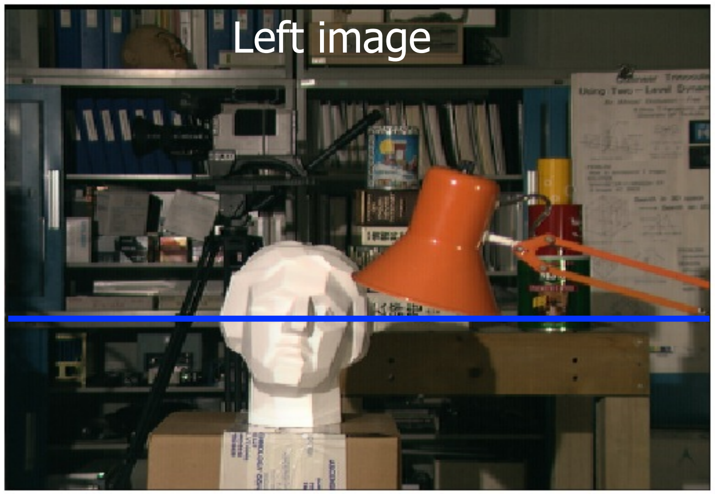
Beyond individual correspondences to estimate disparities:

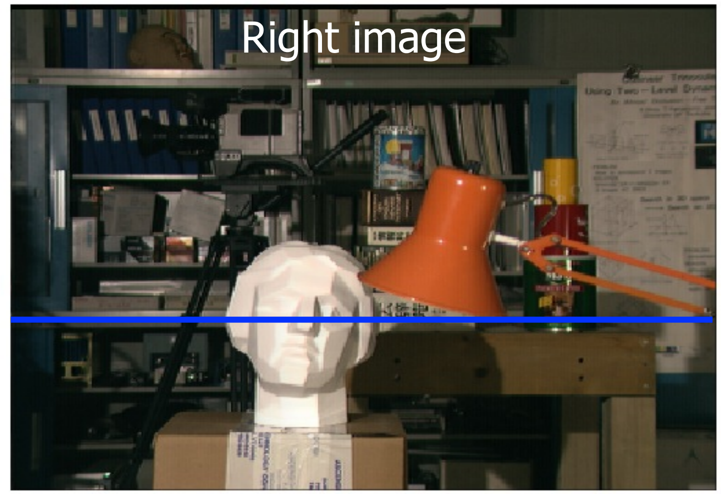
Optimize correspondence assignments jointly

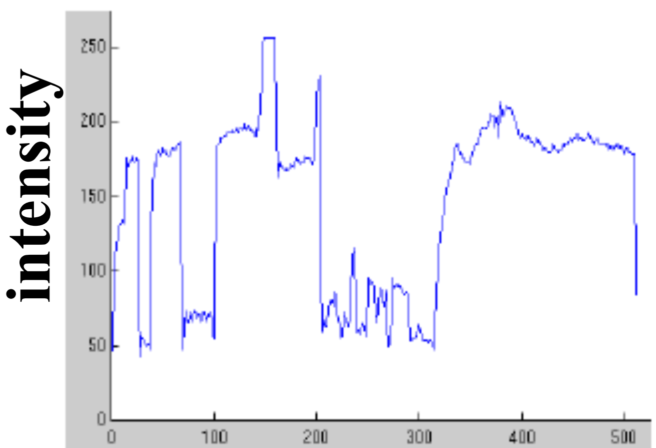
- Scanline at a time (DP)
- Full 2D grid (graph cuts)

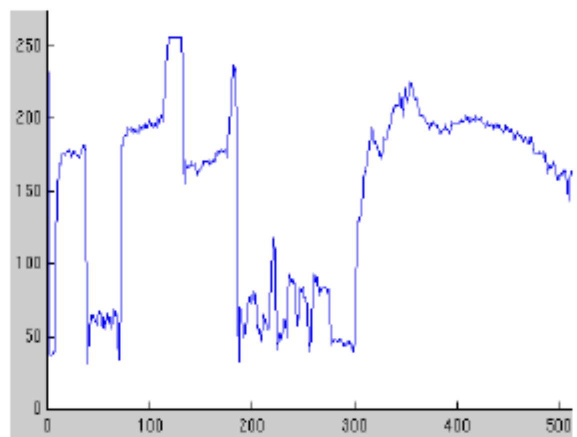
Scanline stereo

- Try to coherently match pixels on the entire scanline
- Different scanlines are still optimized independently

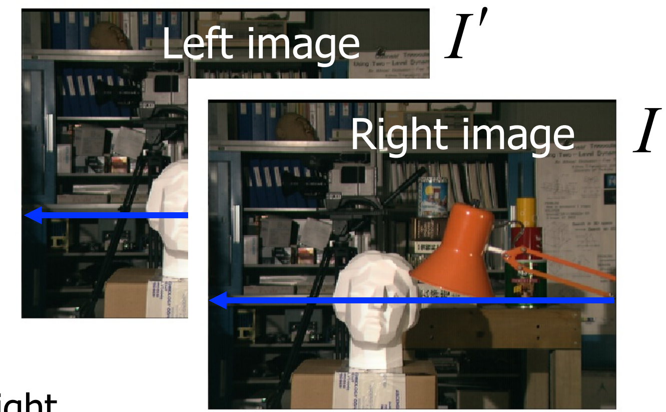


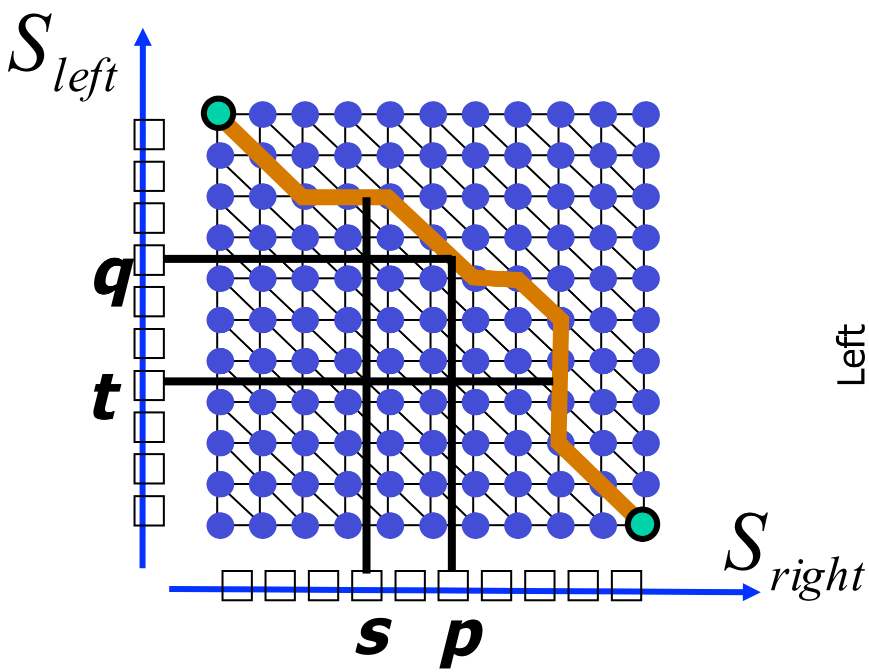


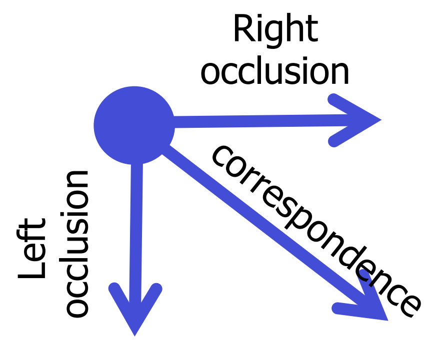




“Shortest paths” for scan-line stereo







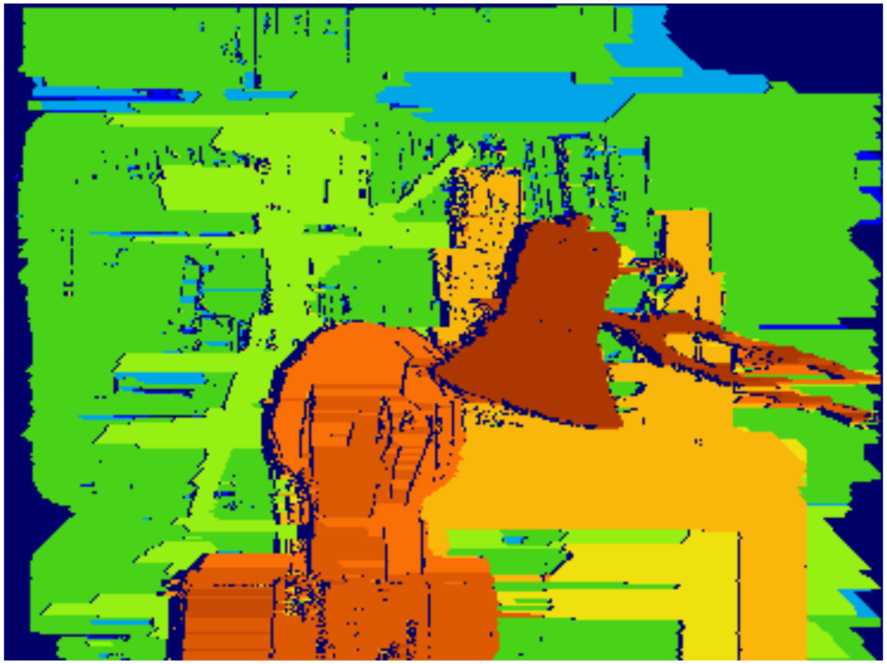
Can be implemented with dynamic programming

Ohta & Kanade ’85, Cox et al. ‘96

Slide credit: Y. Boykov

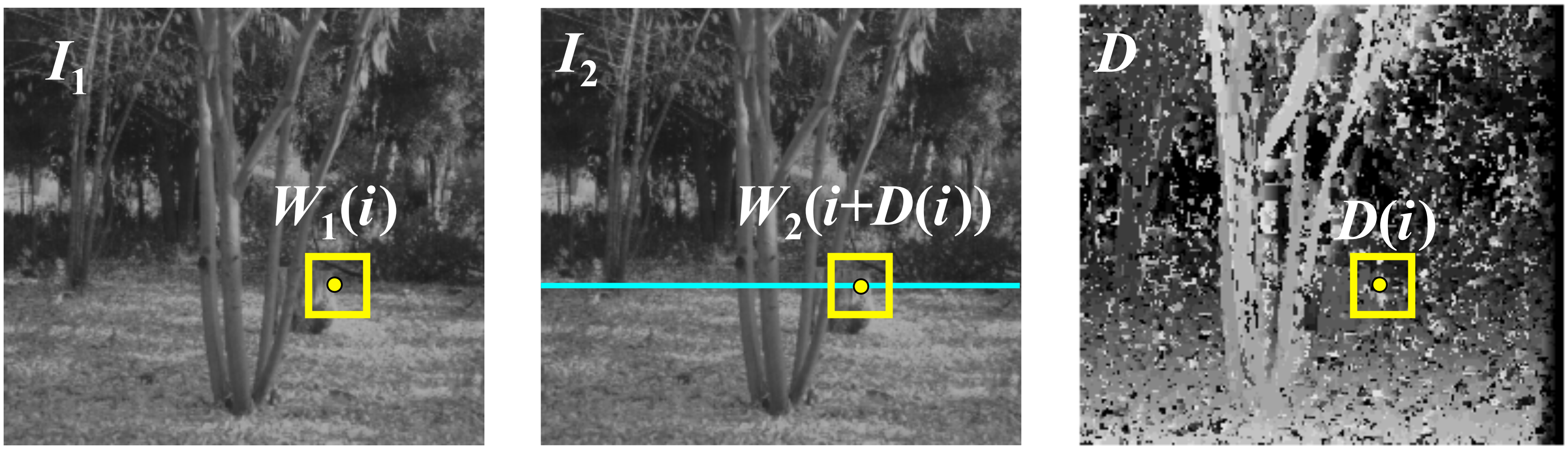
Coherent stereo on 2D grid

- Scanline stereo generates streaking artifacts



- Can't use dynamic programming to find spatially coherent disparities/ correspondences on a 2D grid

Stereo matching as energy minimization



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

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

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

- 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

Source: Steve Seitz

Recap: stereo with calibrated cameras

Given image pair, **K**, **R, T**

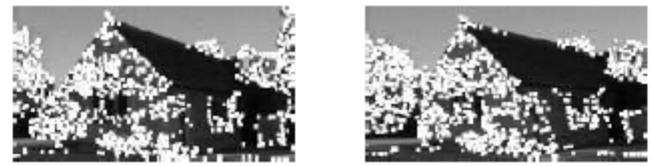
Detect some features

Compute essential matrix **E**

Match features using the epipolar and other constraints

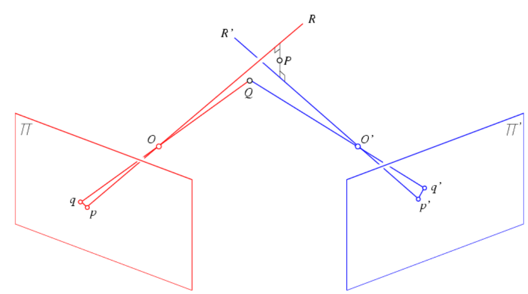
Triangulate for 3d structure

Note: (R,T) can e.g. be computed from E (see book HZ/RS) or from a calibration pattern



Left Right

Left Right



Multi-view reconstruction
Un-calibrated cameras

(Slides adapted from Svetlana Lazebnik)

Multiple-view geometry questions

- **Scene geometry (structure):** Given 2D point matches in two or more images, where are the corresponding points in 3D?

- **Correspondence (stereo matching):** Given a point in just one image, how does it constrain the position of the corresponding point in another image?

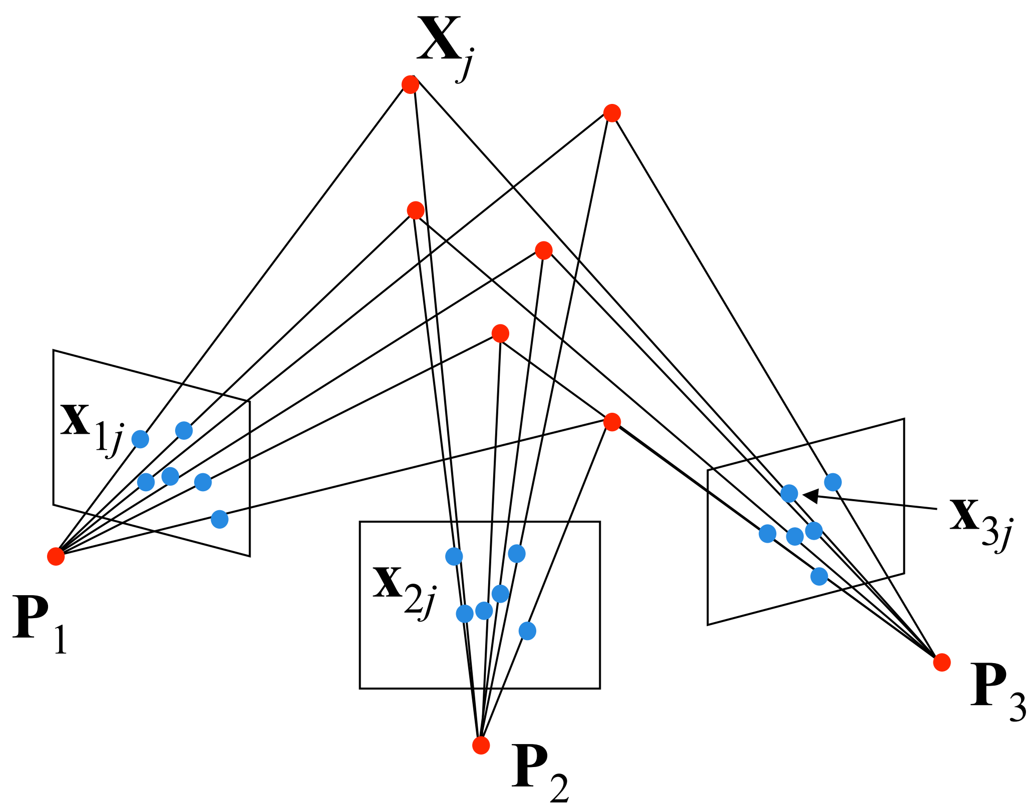
- **Camera geometry (motion):** Given a set of corresponding points in two or more images, what are the camera matrices for these views?

Structure from motion

- Given: *m* images of *n* fixed 3D points

$$\mathbf{x}_{ij} = \mathbf{P}_i \mathbf{X}_j, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n$$

- Problem: estimate *m* projection matrices $\mathbf{P}_i$ and *n* 3D points $\mathbf{X}_j$ from the *mn* correspondences $\mathbf{x}_{ij}$



Structure from motion ambiguity

- If we scale the entire scene by some factor *k* and, at the same time, scale the camera matrices by the factor of 1/*k*, the projections of the scene points in the image remain exactly the same:

$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\frac{1}{k}\mathbf{P}\right)(k\mathbf{X})$$

It is impossible to recover the absolute scale of the scene!

Structure from motion ambiguity

- If we scale the entire scene by some factor *k* and, at the same time, scale the camera matrices by the factor of 1/*k*, the projections of the scene points in the image remain exactly the same

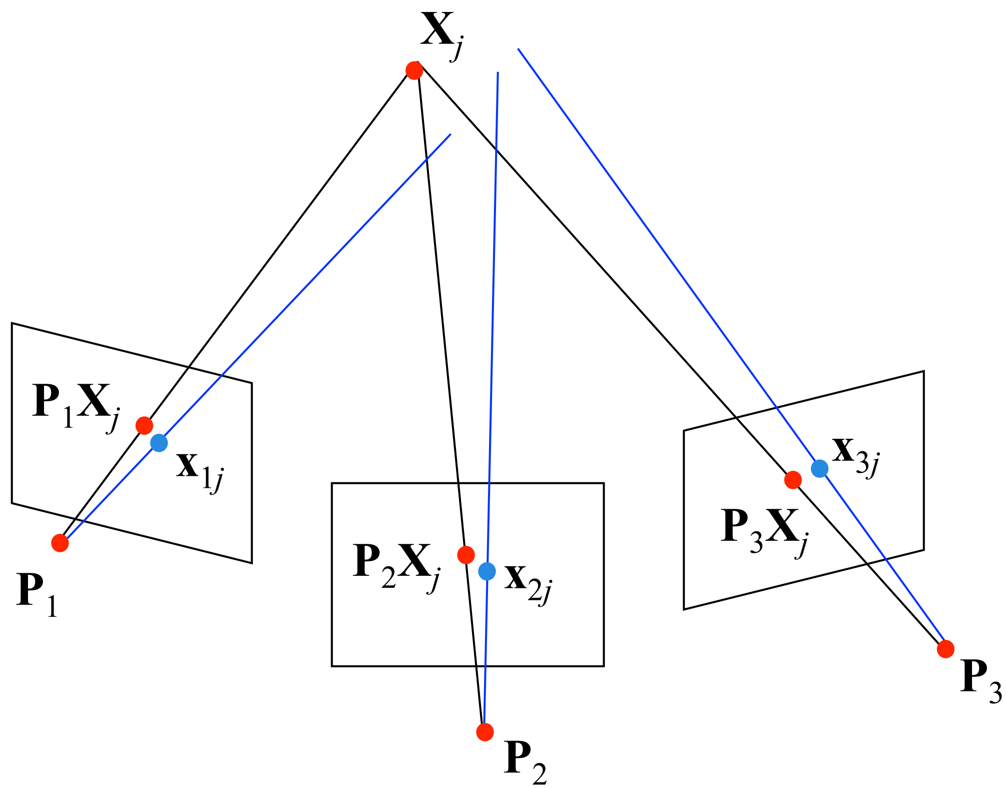
- More generally: if we transform the scene using a transformation **Q** and apply the inverse transformation to the camera matrices, then the images do not change

$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{Q}^{-1}\right)\left(\mathbf{Q}\mathbf{X}\right)$$

Bundle adjustment

- Non-linear method for refining structure and motion
- Minimizing reprojection error

$$E(\mathbf{P},\mathbf{X}) = \sum_{i=1}^{m}\sum_{j=1}^{n} D\left(\mathbf{x}_{ij}, \mathbf{P}_i\mathbf{X}_j\right)^2$$

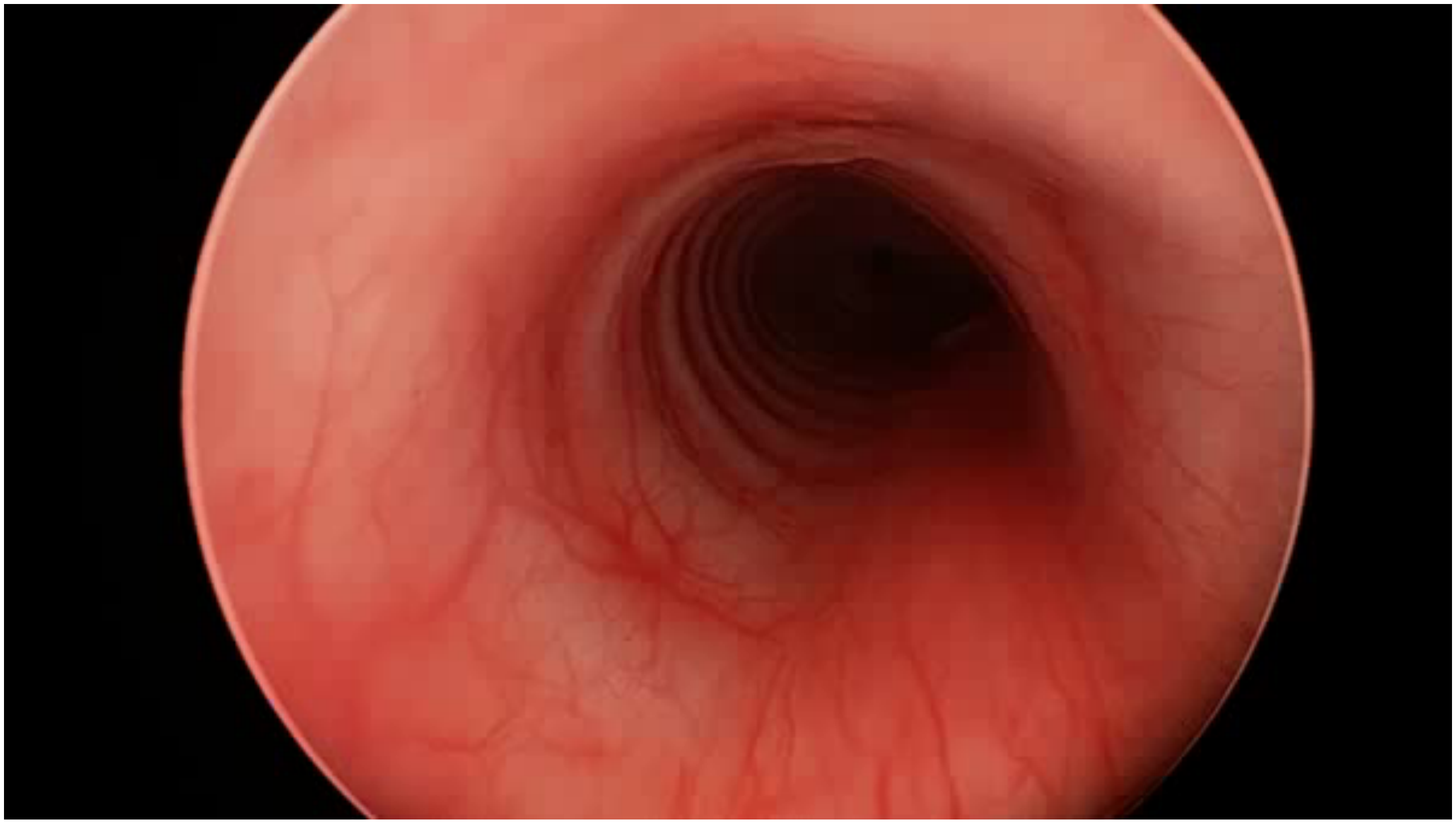


Typical Bundle Adjustment Algorithm

1. Detect matches in all images
2. Pick pairs of images and compute epipolar geometry
3. Find connected components and perform bundle adjustment
4. Detect additional matches if possible based on (now known) epipolar geometry
5. Repeat
6. Optionally perform dense surface reconstruction

Quantitative Endoscopy

(with Masaru Ishii, MD, Eric Meisner, Haluk Tokgozoglu)



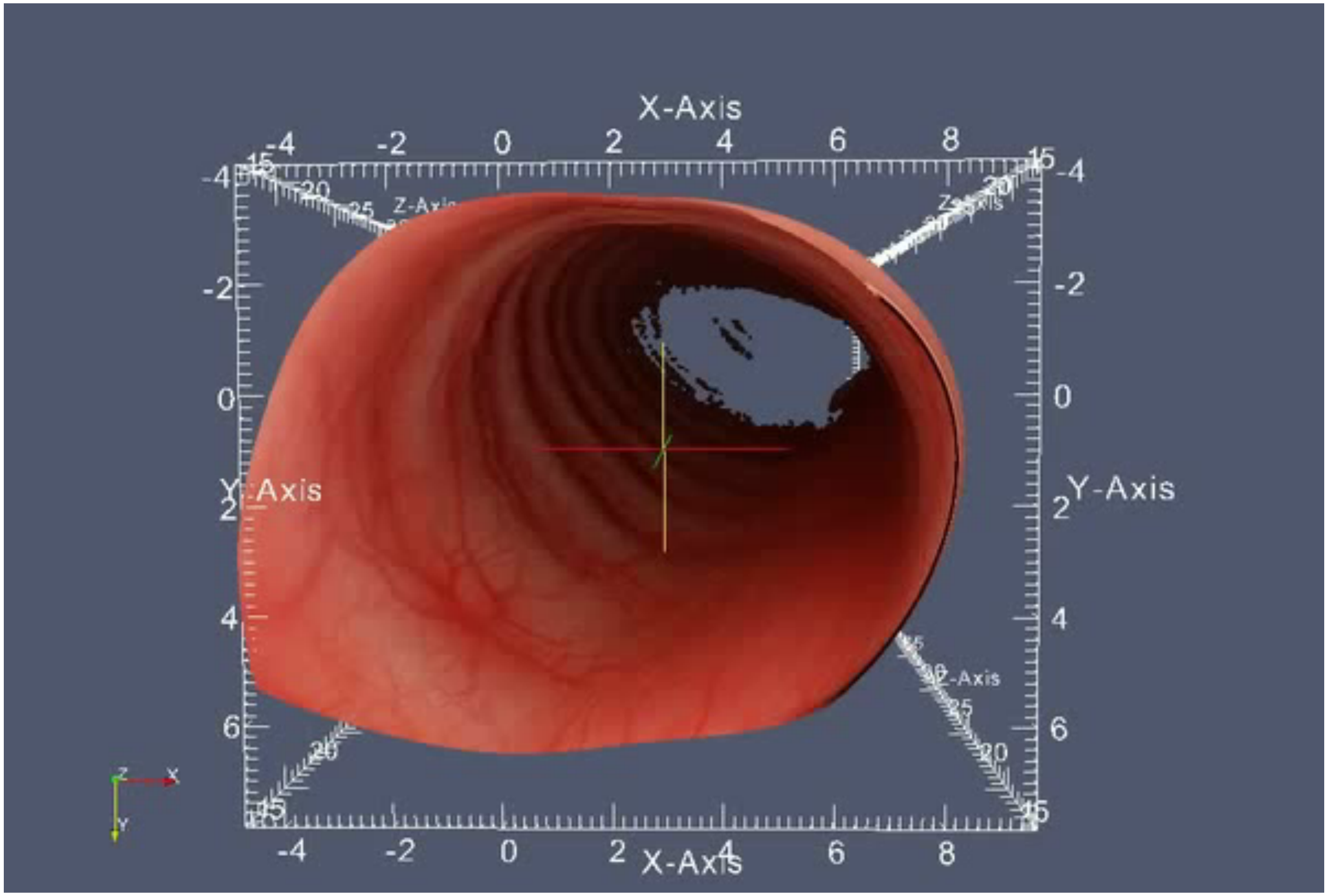
Can we measure and evaluate noninvasively?

LABORATORY FOR Computational Sensing + Robotics THE JOHNS HOPKINS UNIVERSITY

ERC | CISST

Quantitative Endoscopy

(with Masaru Ishii, MD, Eric Meisner, Haluk Tokgozoglu)



Can we measure and evaluate noninvasively?

Laboratory for Computational Sensing + Robotics
The Johns Hopkins University

ERC | CISST

Conclusion

- 3D transformations
- Epipolar geometry
- Essential matrix
- Fundamental matrix
- Triangulation
- Algorithms for stereo matching