Bill Freeman and Antonio Torralba

Lecture 12
Calibration and Stereo

Szeliski book: section 2.1, section 7.2, chapter 11.

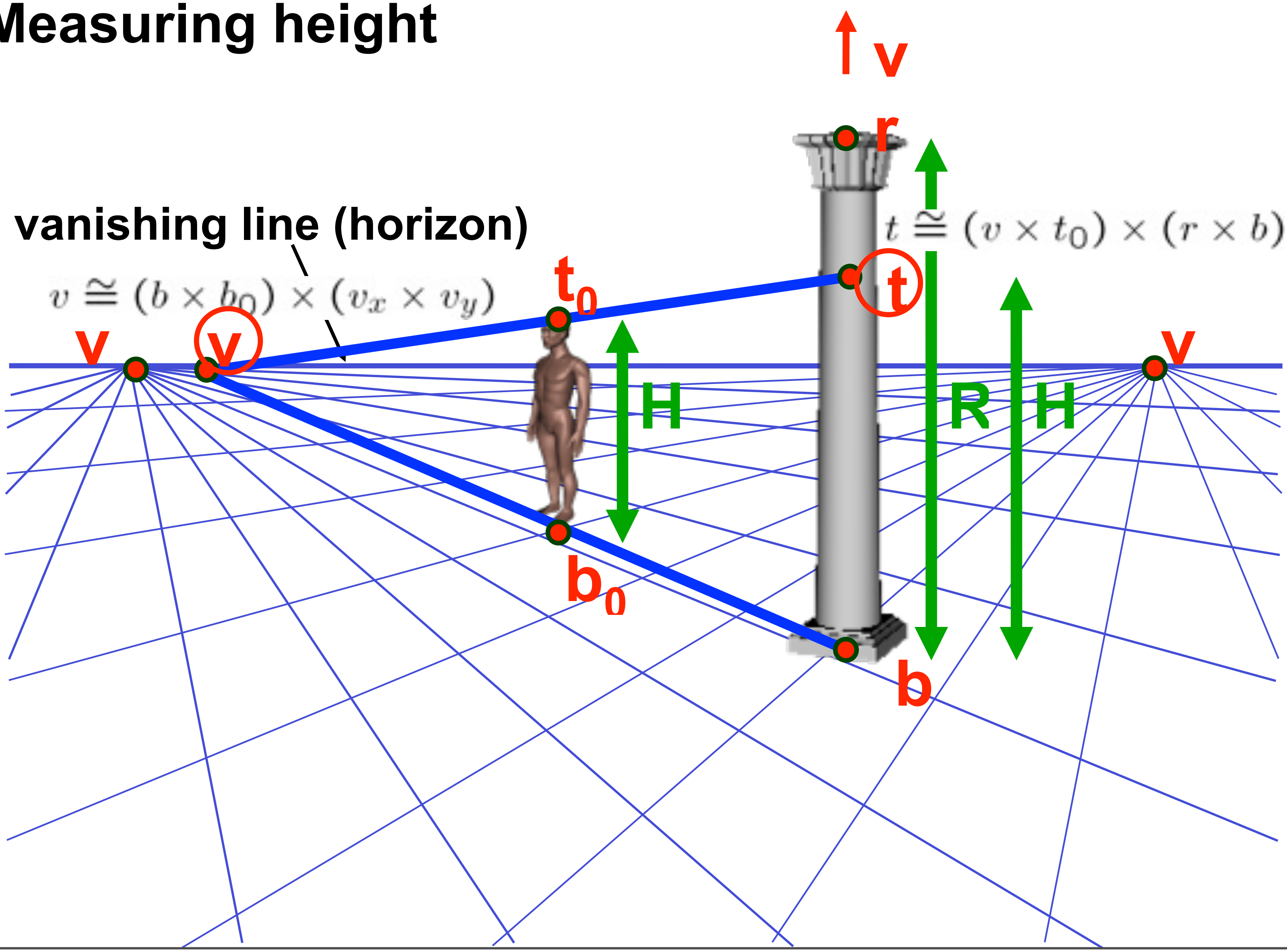
Camera calibration

Use the camera to tell you things about the world:

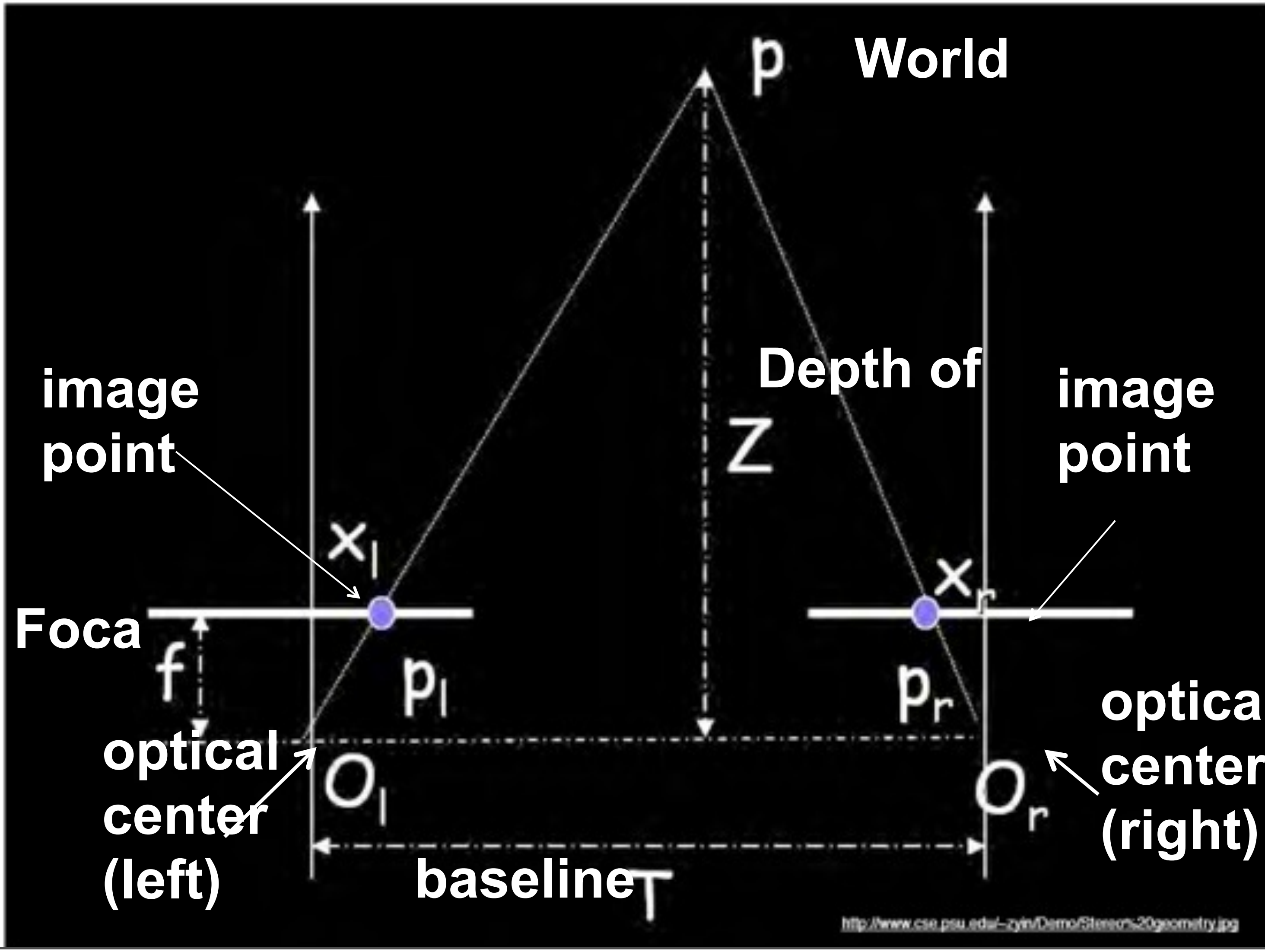
- Relationship between coordinates in the world and coordinates in the image: *geometric camera calibration, see* Szeliski, chapter 6 (see ch. 11 on stereo)

One reason to calibrate a camera

Measuring height



Another reason to calibrate a camera



Three camera projections

3-d point **2-d image position**

(1) Perspective: $(x, y, z) \rightarrow \left(\frac{fx}{z}, \frac{fy}{z}\right)$

(2) Weak perspective: $(x, y, z) \rightarrow \left(\frac{fx}{z_0}, \frac{fy}{z_0}\right)$

(3) Orthographic: $(x, y, z) \rightarrow (x, y)$

Is the perspective projection a linear transformation?

Slide by Steve Seitz

Is the perspective projection a linear transformation?

no—division by z is nonlinear

Slide by Steve Seitz

Homogeneous coordinates

Is the perspective projection a linear transformation?

no—division by z is nonlinear

Slide by Steve Seitz

Homogeneous coordinates

Is the perspective projection a linear transformation?

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

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

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

Slide by Steve Seitz

Perspective Projection

Euclidean coords

2d projection **3d point**

$$\left(\frac{fx}{z}, \frac{fy}{z}\right) \leftarrow (x, y, z)$$

Perspective Projection

Euclidean coords

2d projection

3d point

$$\left(\frac{fx}{z}, \frac{fy}{z}\right) \leftarrow (x, y, z)$$

Homogeneous coords

2d projection

3d point

$$\begin{bmatrix} \frac{fx}{z} \\ \frac{fy}{z} \\ 1 \end{bmatrix} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Perspective Projection

Euclidean coords

2d projection

3d point

$$\left(\frac{fx}{z}, \frac{fy}{z}\right) \leftarrow (x, y, z)$$

Homogeneous coords

2d projection

3d point

$$\begin{bmatrix} \frac{fx}{z} \\ \frac{fy}{z} \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f \end{bmatrix} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Perspective Projection

Euclidean coords

2d projection

3d point

$$\left(\frac{fx}{z}, \frac{fy}{z}\right) \leftarrow (x, y, z)$$

Homogeneous coords

2d projection

3d point

$$\begin{bmatrix} \frac{fx}{z} \\ \frac{fy}{z} \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f \end{bmatrix} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Perspective Projection

Euclidean coords

2d projection — 3d point

$$\left(\frac{fx}{z}, \frac{fy}{z}\right) \leftarrow (x, y, z)$$

Homogeneous coords

2d projection — 3d point

$$\begin{bmatrix} \frac{fx}{z} \\ \frac{fy}{z} \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f \end{bmatrix} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Perspective projection is a matrix multiply using homogeneous coordinates. The matrix is called the projection matrix.

Perspective Projection

How does scaling the projection matrix change the transformation?

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f \end{bmatrix} \Rightarrow \left(f\frac{x}{z}, f\frac{y}{z} \right)$$

Slide by Steve Seitz

Perspective Projection

How does scaling the projection matrix change the transformation?

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f \end{bmatrix} \Rightarrow \left(f\frac{x}{z}, f\frac{y}{z} \right)$$

$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Slide by Steve Seitz

Perspective Projection

How does scaling the projection matrix change the transformation?

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f \end{bmatrix} \Rightarrow \left(f\frac{x}{z}, f\frac{y}{z} \right)$$

$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} fx \\ fy \\ z \end{bmatrix}$$

Slide by Steve Seitz

Perspective Projection

How does scaling the projection matrix change the transformation?

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f \end{bmatrix} \Rightarrow \left(f\frac{x}{z}, f\frac{y}{z} \right)$$

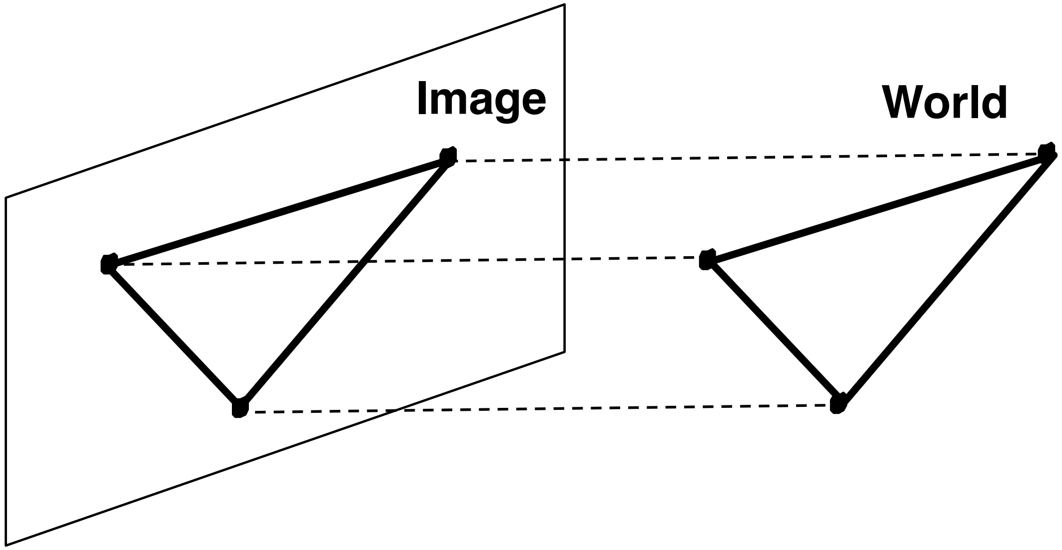
$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} fx \\ fy \\ z \end{bmatrix} \Rightarrow \left(f\frac{x}{z}, f\frac{y}{z} \right)$$

Slide by Steve Seitz

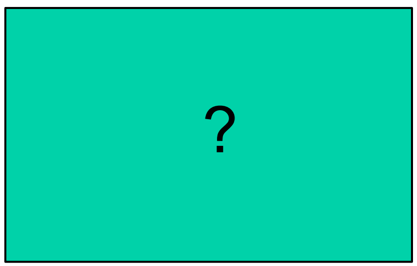
Orthographic Projection

Special case of perspective projection

- Distance from the COP to the PP is infinite



- Also called "parallel projection"
- What's the projection matrix?

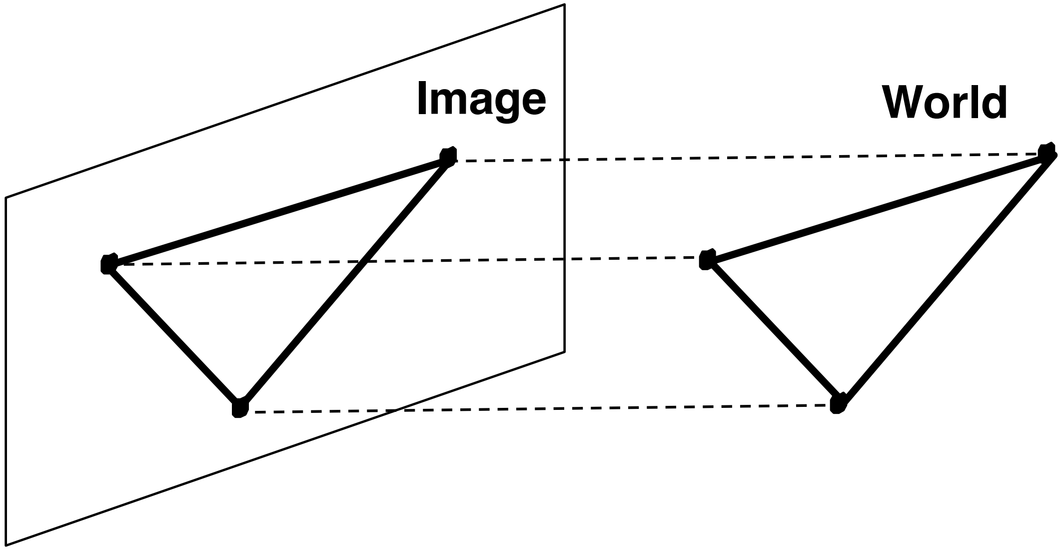


Slide by Steve Seitz

Orthographic Projection

Special case of perspective projection

- Distance from the COP to the PP is infinite



- Also called “parallel projection”
- What’s the projection matrix?

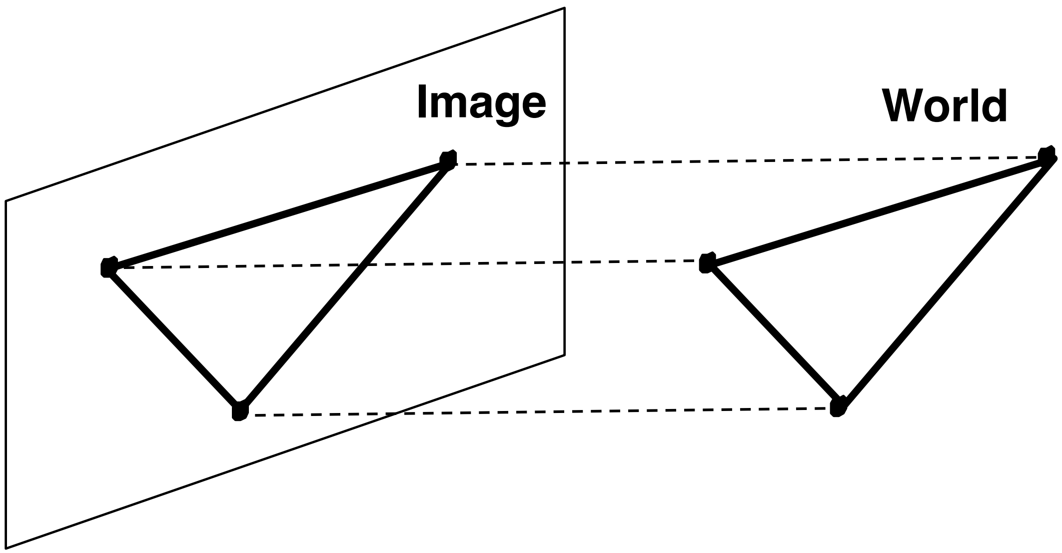
$$\boxed{?} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow (x, y)$$

Slide by Steve Seitz

Orthographic Projection

Special case of perspective projection

- Distance from the COP to the PP is infinite



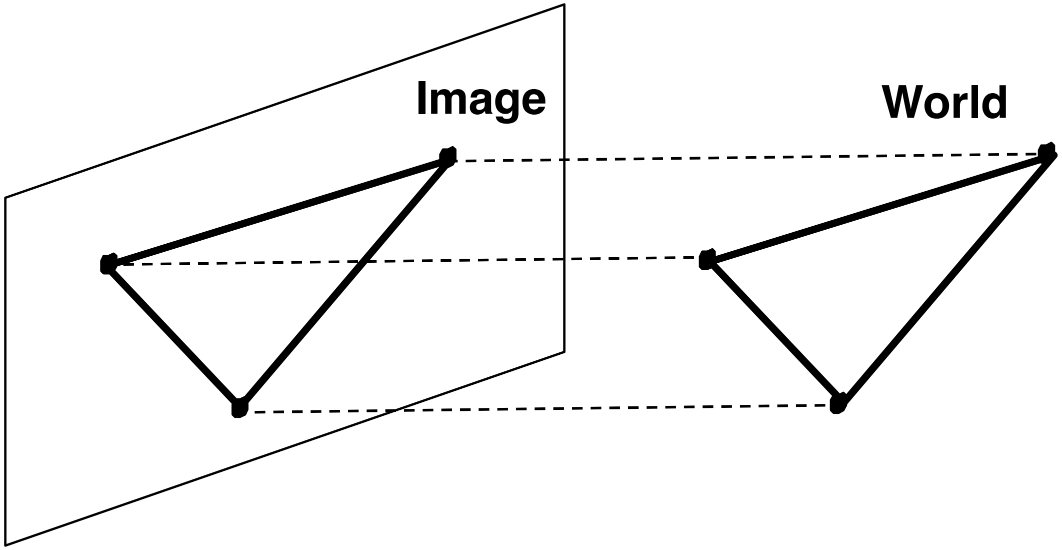
- Also called “parallel projection”
- What’s the projection matrix?

Slide by Steve Seitz

Orthographic Projection

Special case of perspective projection

- Distance from the COP to the PP is infinite

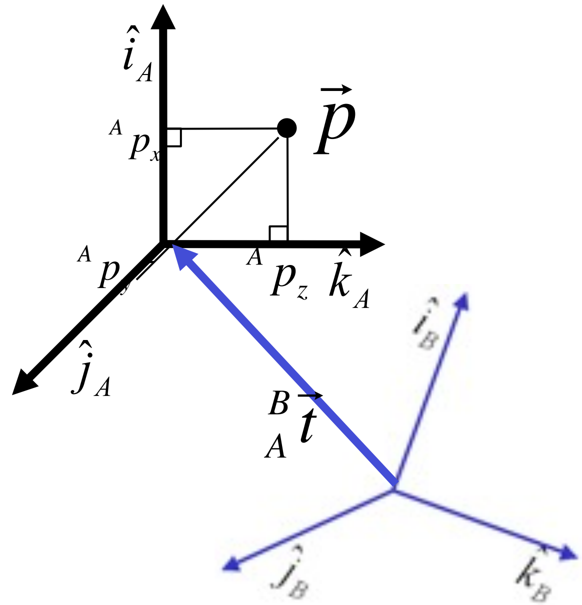


- Also called “parallel projection”
- What’s the projection matrix?

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow (x, y)$$

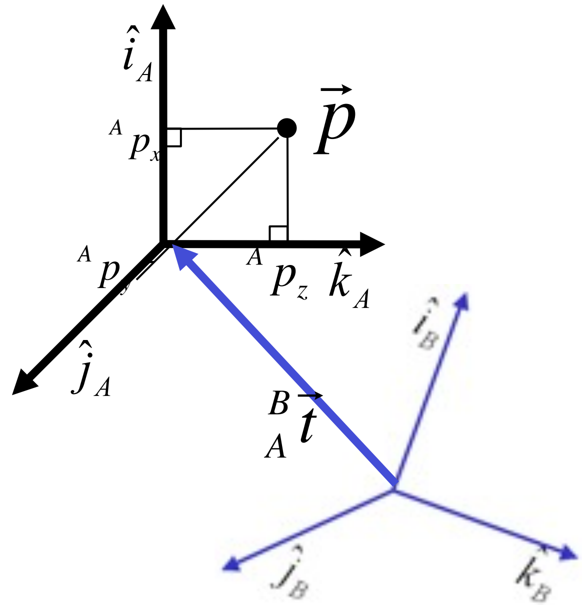
Slide by Steve Seitz

Translation and rotation



Translation and rotation

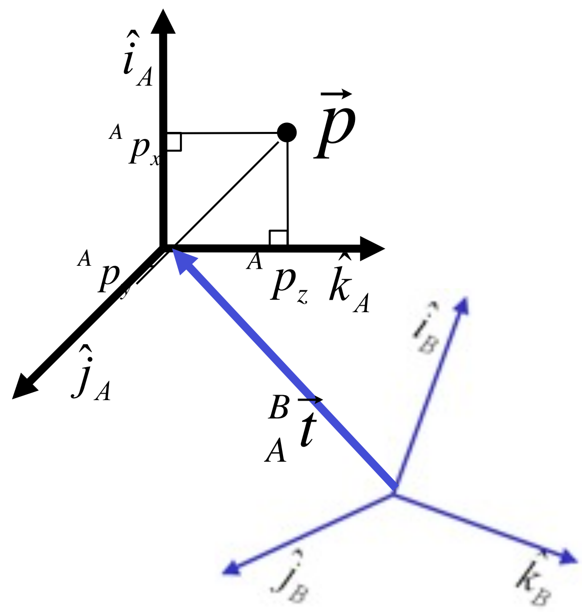
$$ {}^{B}\vec{p} = {}^{B}_{A}R \; {}^{A}\vec{p} + {}^{B}_{A}\vec{t} $$



Translation and rotation

"as described in the coordinates of frame B"

$$ {}^{B}\vec{p} = {}^{B}_{A}R \, {}^{A}\vec{p} + {}^{B}_{A}\vec{t} $$

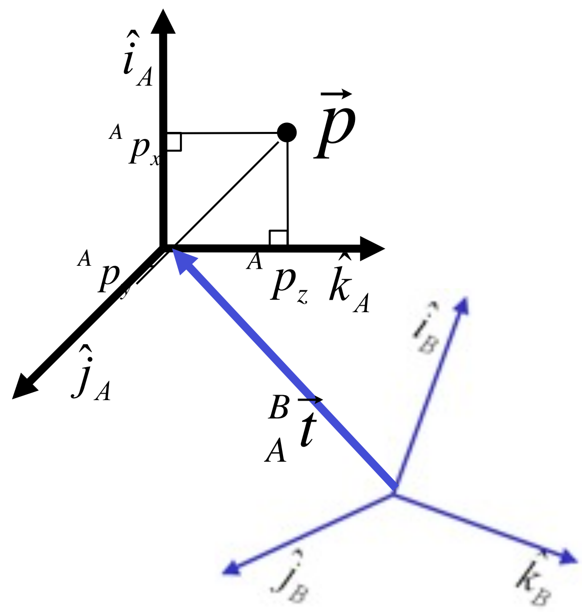


Translation and rotation

“as described in the coordinates of frame B”

Let’s write

$$ {}^{B}\vec{p} = {}^{B}_{A}R \, {}^{A}\vec{p} + {}^{B}_{A}\vec{t} $$



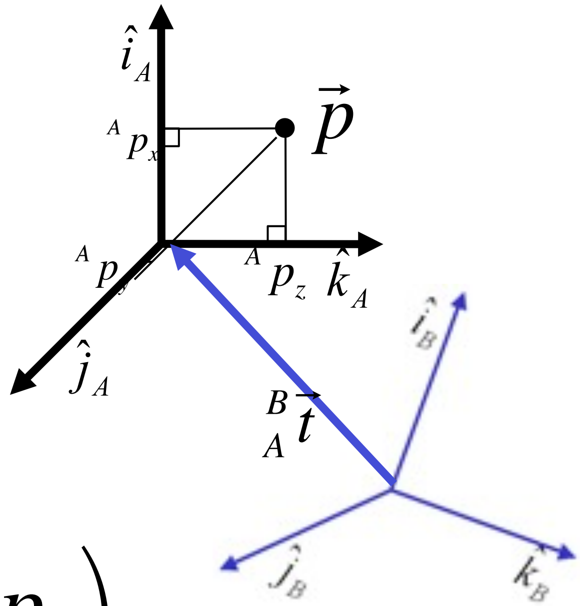
as a single matrix equation:

Translation and rotation

"as described in the coordinates of frame B"

Let's write

$$ {}^{B}\vec{p} = {}^{B}_{A}R \; {}^{A}\vec{p} + {}^{B}_{A}\vec{t} $$



as a single matrix equation:

$$ \begin{pmatrix} {}^{B}p_x \\ {}^{B}p_y \\ {}^{B}p_z \\ 1 \end{pmatrix} = \begin{pmatrix} - & - & - & | \\ - & {}^{B}_{A}R & - & {}^{B}_{A}t \\ - & - & - & | \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{A}p_x \\ {}^{A}p_y \\ {}^{A}p_z \\ 1 \end{pmatrix} $$

Translation and rotation, written in each set of coordinates

Non-homogeneous coordinates

$$^{B}\vec{p} = {}^{B}_{A}R \; {}^{A}\vec{p} + {}^{B}_{A}\vec{t}$$

Translation and rotation, written in each set of coordinates

Non-homogeneous coordinates

$$ {}^{B}\vec{p} = {}^{B}_{A}R \; {}^{A}\vec{p} + {}^{B}_{A}\vec{t} $$

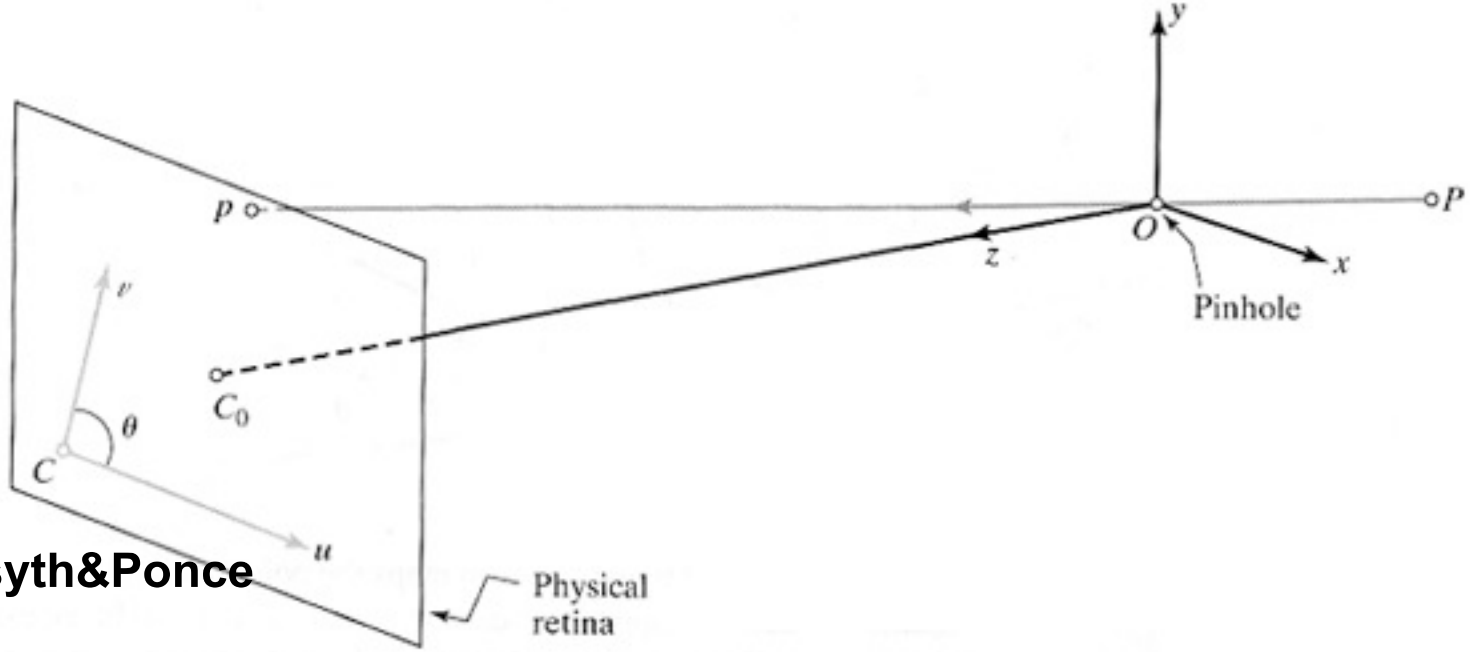
Homogeneous coordinates

$$ {}^{B}\vec{p} = {}^{B}_{A}C \; {}^{A}\vec{p} $$

where

$$ {}^{B}_{A}C = \begin{pmatrix} \boxed{\begin{matrix} - & - & - \\ - & {}^{B}_{A}R & - \\ - & - & - \end{matrix}} & \boxed{\begin{matrix} | \\ {}^{B}_{A}\vec{t} \\ | \end{matrix}} \\ 0 \quad 0 \quad 0 & 1 \end{pmatrix} $$

Intrinsic parameters: from idealized world coordinates to pixel values



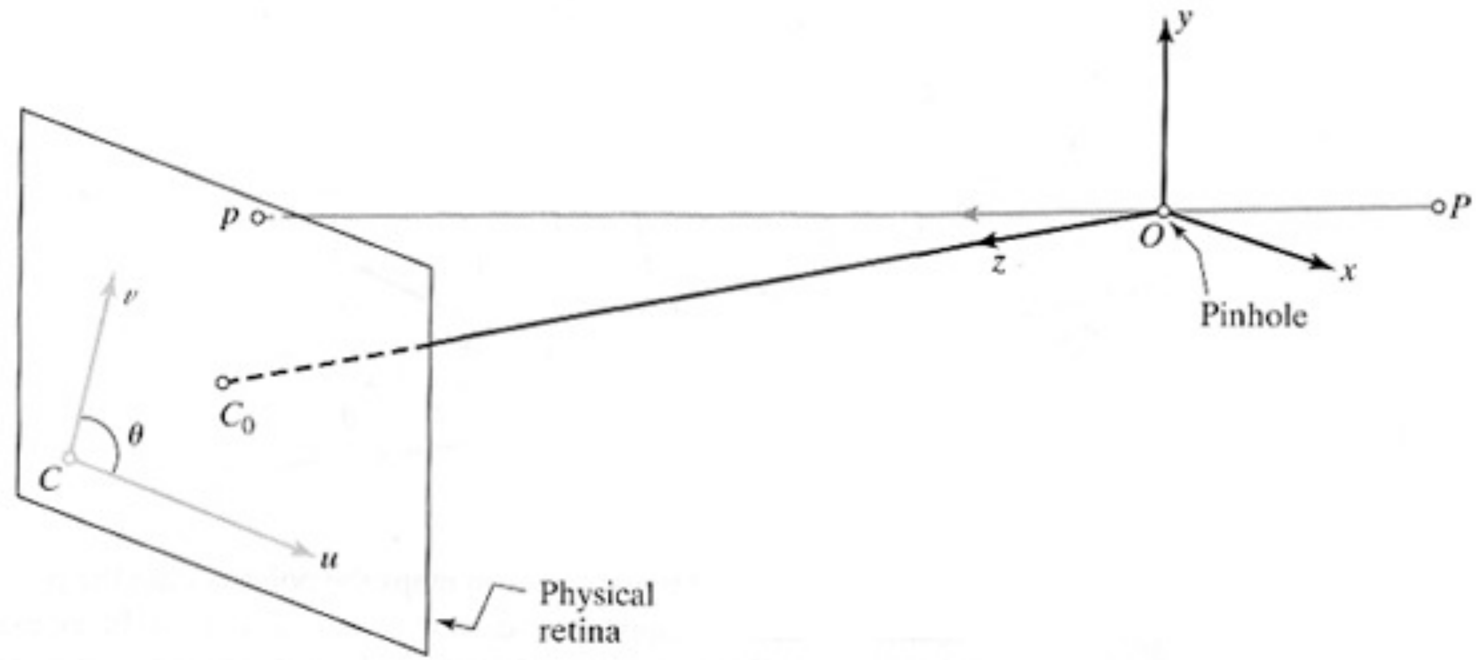
Forsyth&Ponce

Perspective projection

$$u = f\frac{x}{z}$$

$$v = f\frac{y}{z}$$

Intrinsic parameters

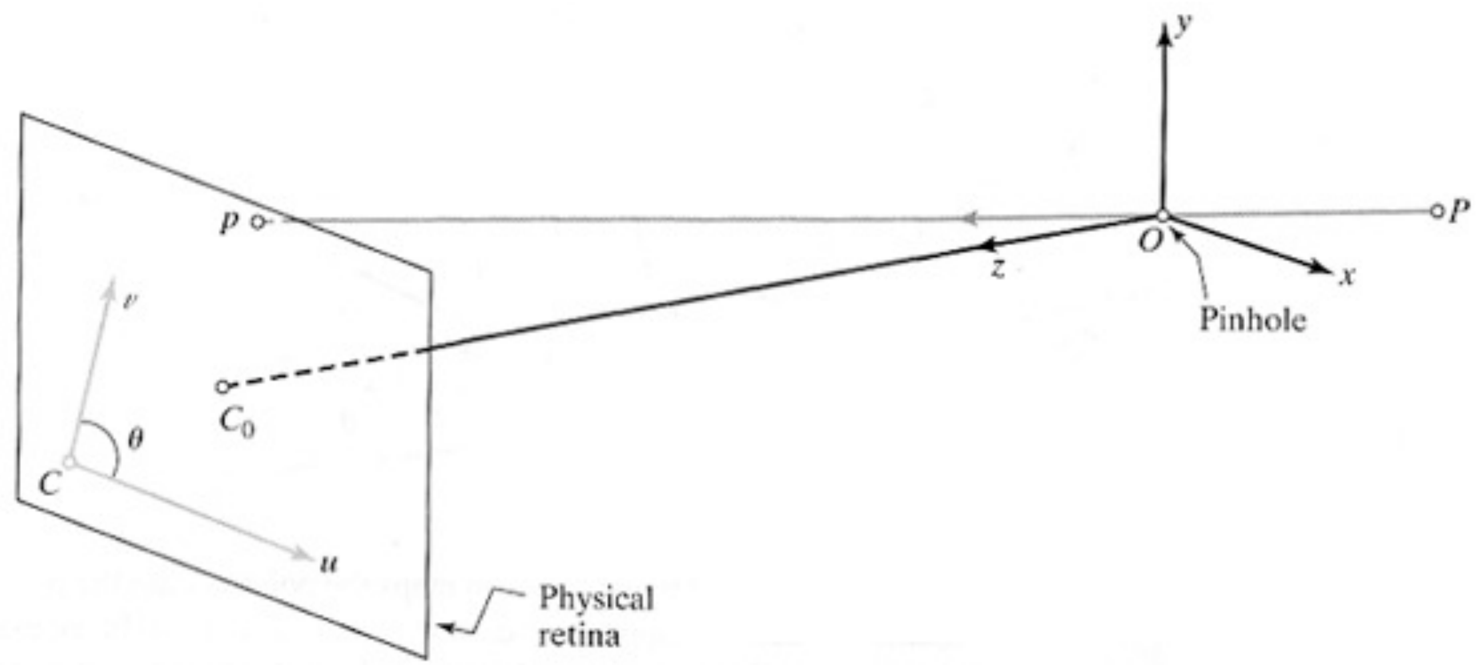


But "pixels" are in some arbitrary spatial units

$$u = \alpha \frac{x}{z}$$

$$v = \alpha \frac{y}{z}$$

Intrinsic parameters

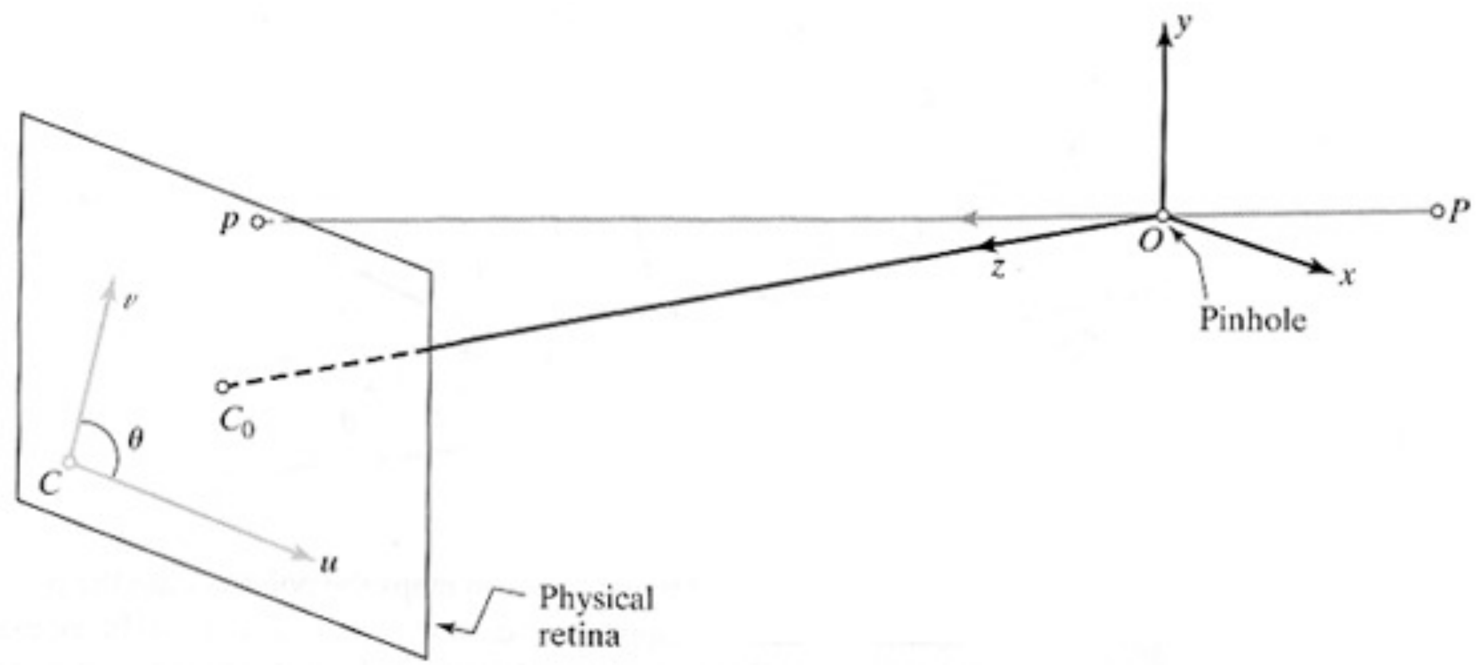


Maybe pixels are not square

$$u = \alpha \frac{x}{z}$$

$$v = \beta \frac{y}{z}$$

Intrinsic parameters

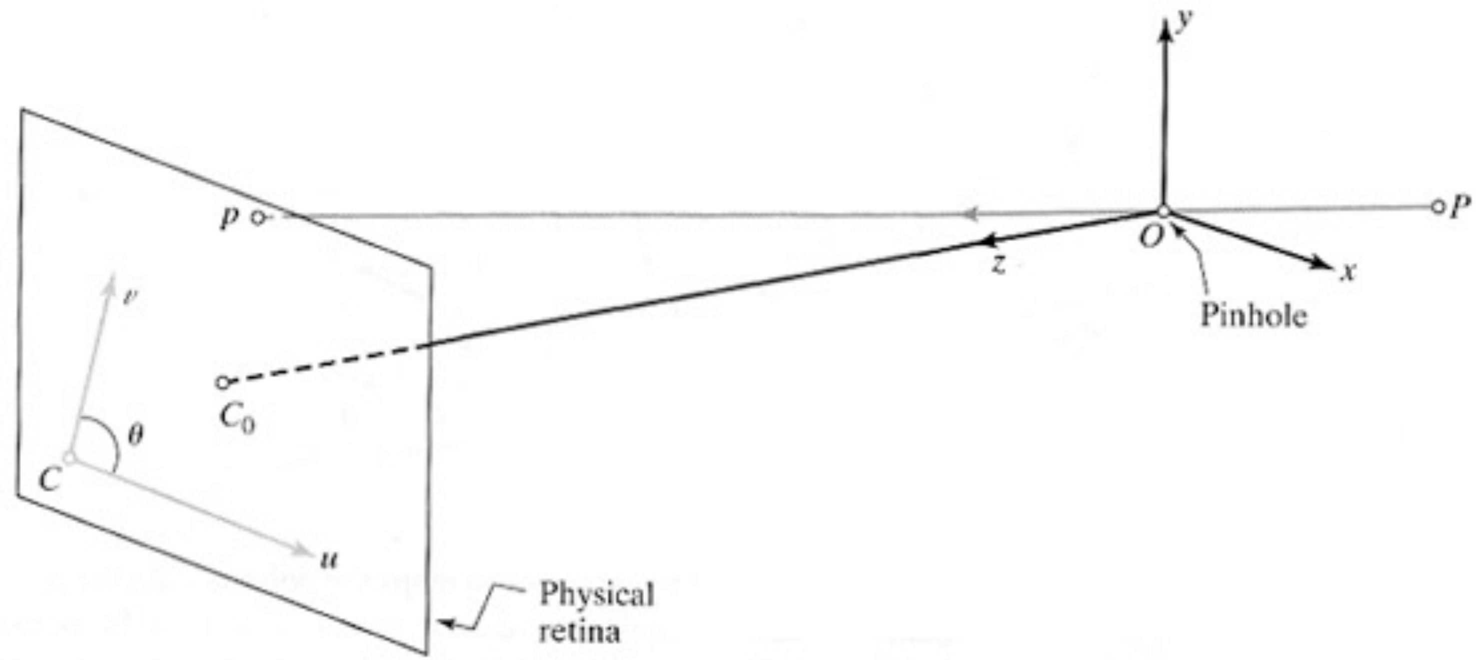


We don't know the origin of our camera pixel coordinates

$$u = \alpha \frac{x}{z} + u_0$$

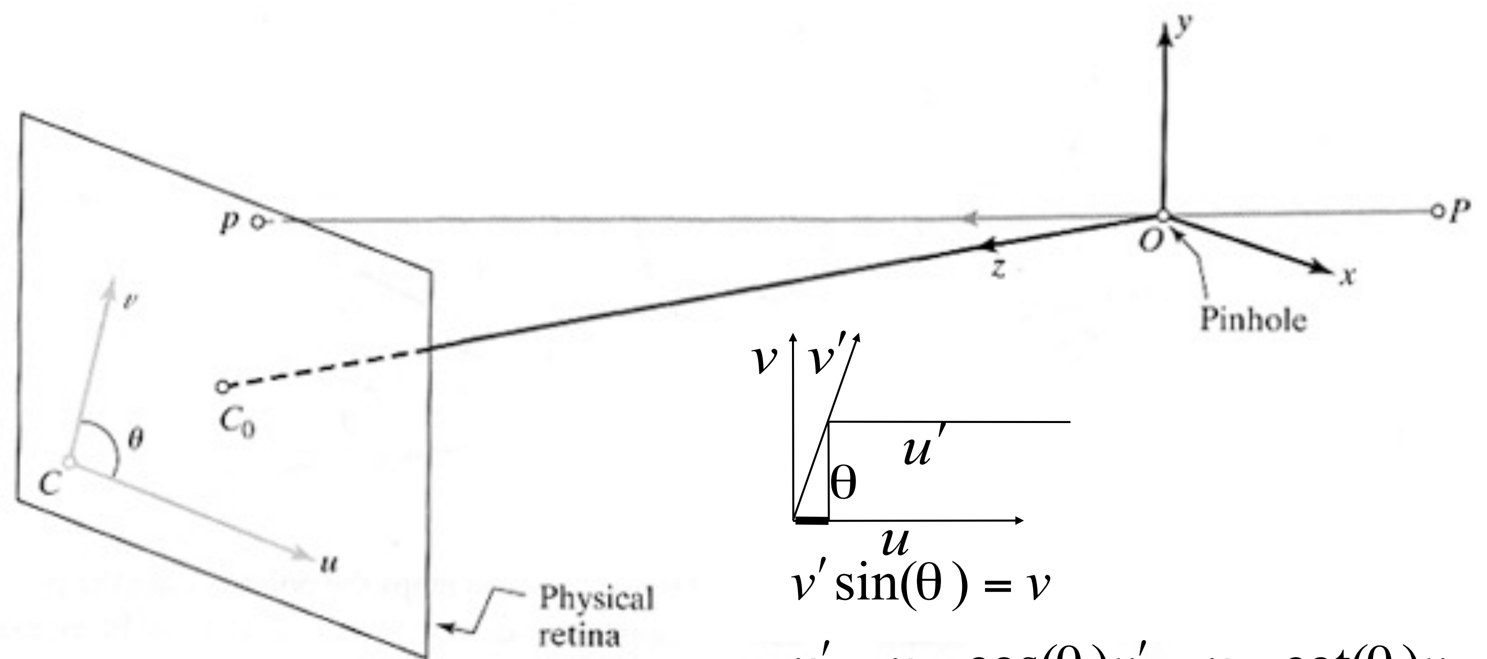
$$v = \beta \frac{y}{z} + v_0$$

Intrinsic parameters



May be skew between camera pixel axes

Intrinsic parameters

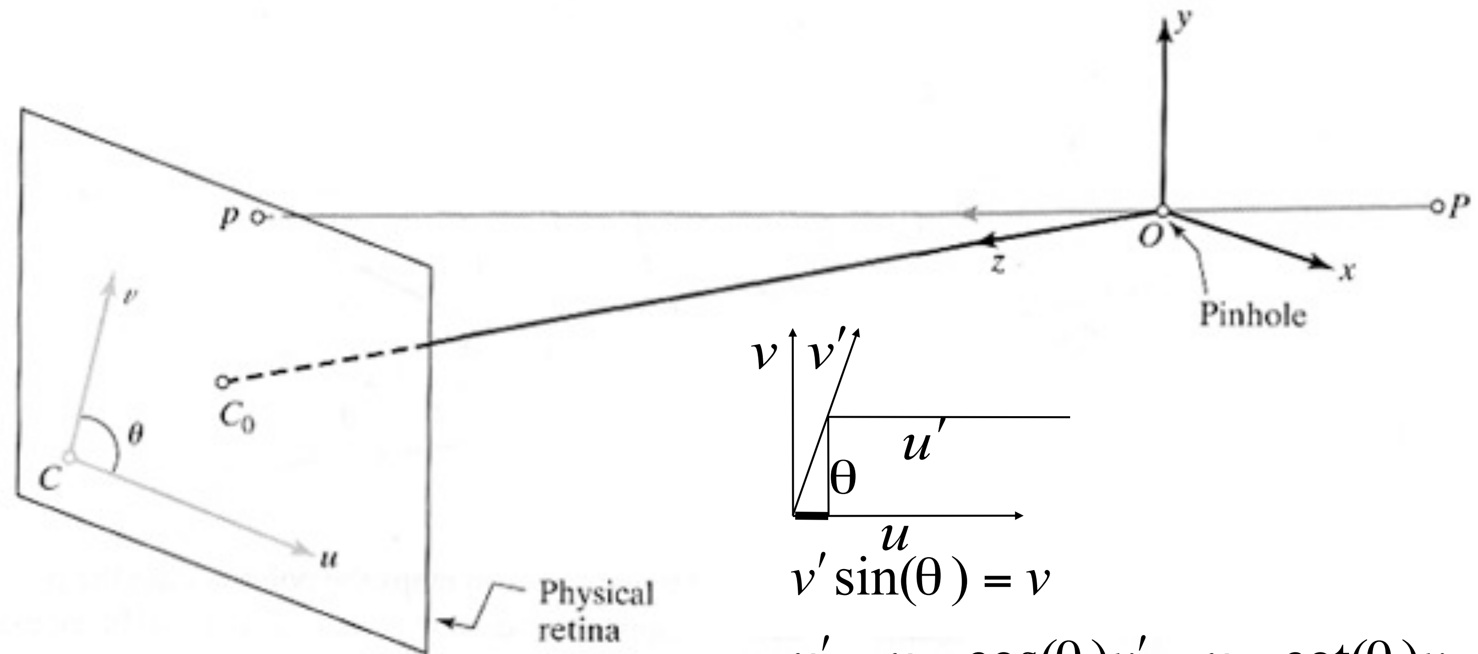


$$v'\sin(\theta) = v$$

$$u' = u - \cos(\theta)v' = u - \cot(\theta)v$$

May be skew between camera pixel axes

Intrinsic parameters



$$v' \sin(\theta) = v$$

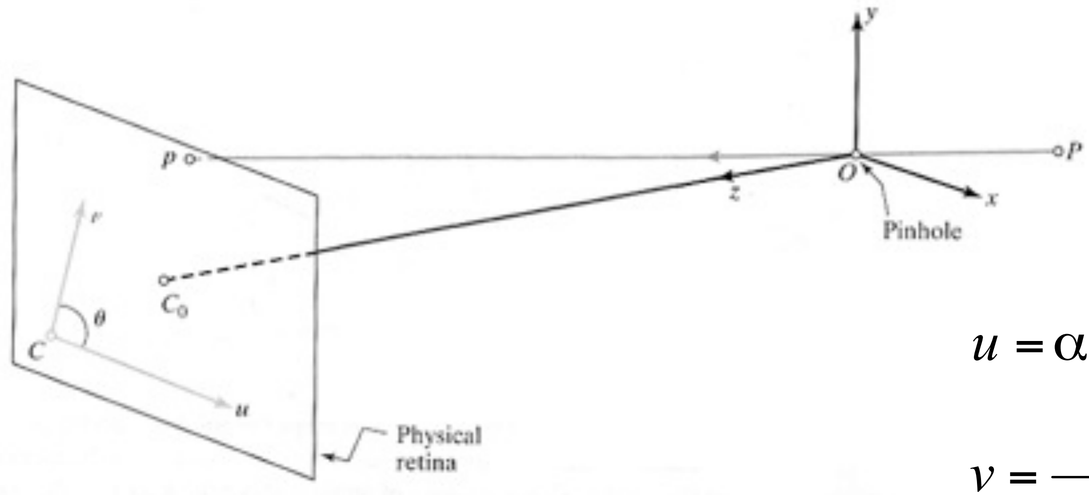
$$u' = u - \cos(\theta) v' = u - \cot(\theta) v$$

May be skew between camera pixel axes

$$u = \alpha \frac{x}{z} - \alpha \cot(\theta) \frac{y}{z} + u_0$$

$$v = \frac{\beta}{\sin(\theta)} \frac{y}{z} + v_0$$

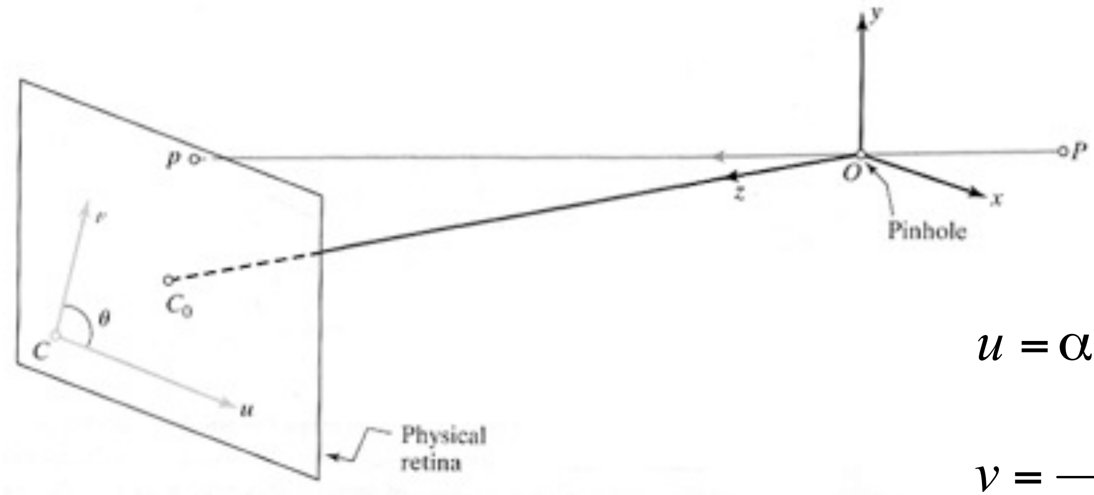
Intrinsic parameters, homogeneous coordinates



$$u = \alpha \frac{x}{z} - \alpha \cot(\theta) \frac{y}{z} + u_0$$

$$v = \frac{\beta}{\sin(\theta)} \frac{y}{z} + v_0$$

Intrinsic parameters, homogeneous coordinates

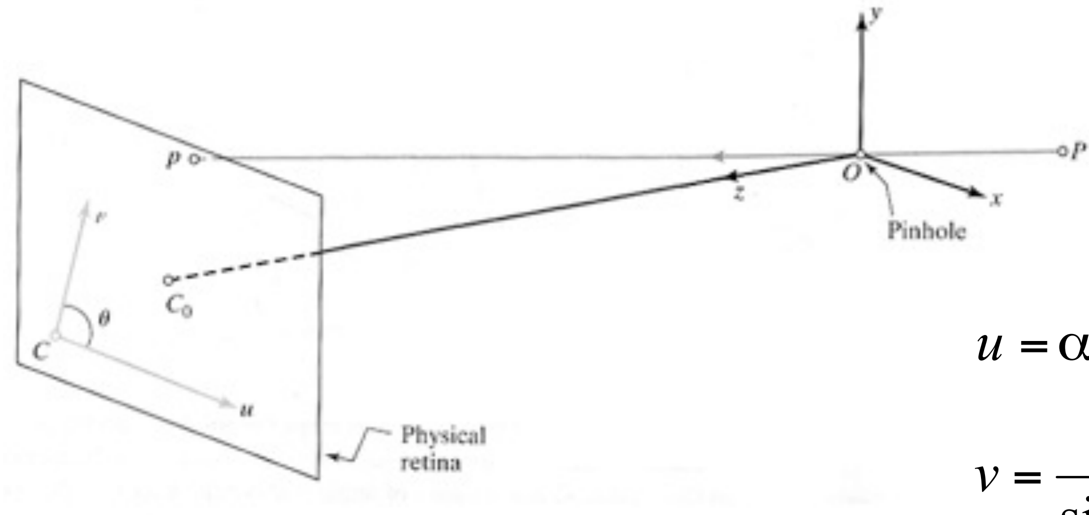


$$u = \alpha \frac{x}{z} - \alpha \cot(\theta) \frac{y}{z} + u_0$$

$$v = \frac{\beta}{\sin(\theta)} \frac{y}{z} + v_0$$

Using homogenous coordinates, we can write this as:

Intrinsic parameters, homogeneous coordinates



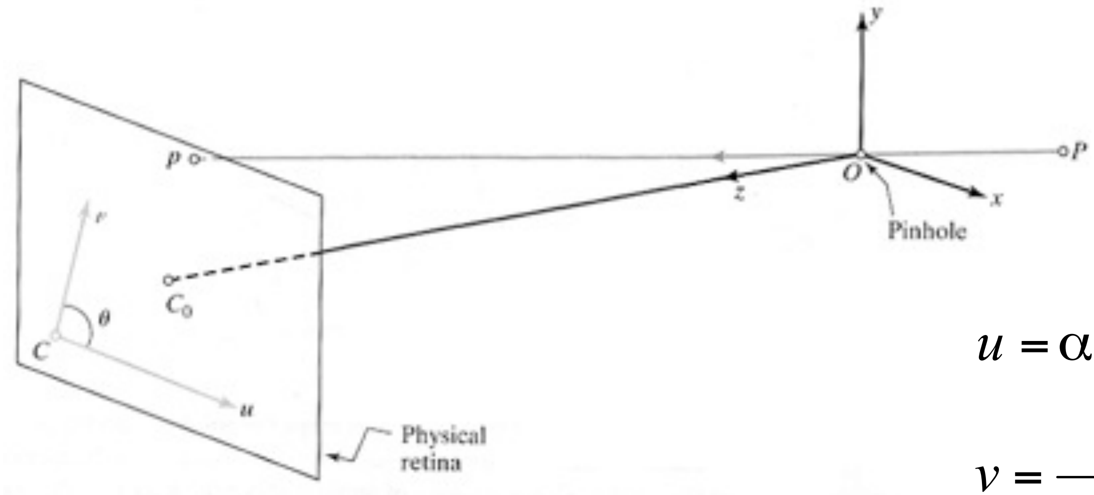
$$u = \alpha \frac{x}{z} - \alpha \cot(\theta) \frac{y}{z} + u_0$$

$$v = \frac{\beta}{\sin(\theta)} \frac{y}{z} + v_0$$

Using homogenous coordinates, we can write this as:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha & -\alpha\cot(\theta) & u_0 & 0 \\ 0 & \dfrac{\beta}{\sin(\theta)} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Intrinsic parameters, homogeneous coordinates



$$u = \alpha \frac{x}{z} - \alpha \cot(\theta) \frac{y}{z} + u_0$$

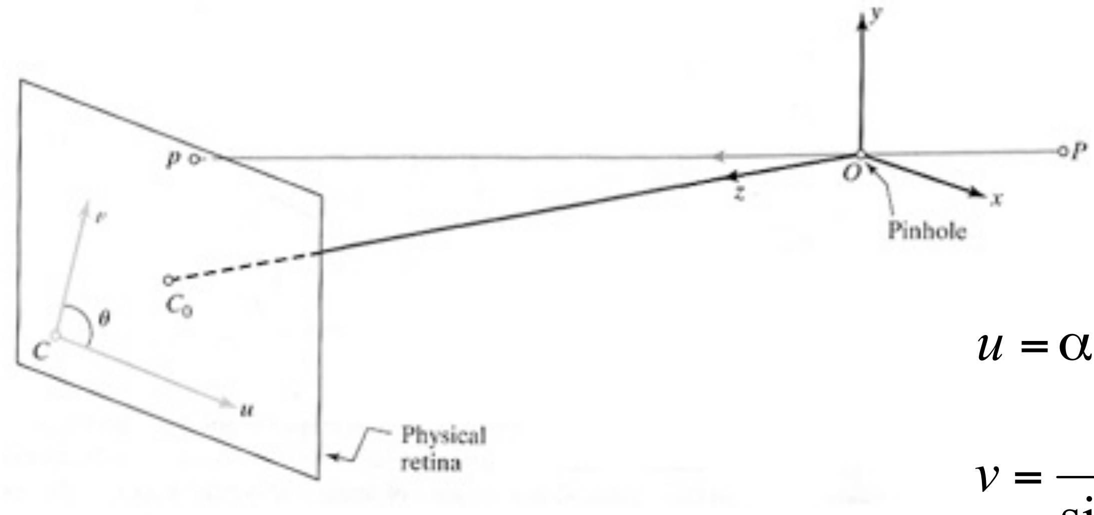
$$v = \frac{\beta}{\sin(\theta)} \frac{y}{z} + v_0$$

Using homogenous coordinates,
we can write this as:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha & -\alpha \cot(\theta) & u_0 & 0 \\ 0 & \dfrac{\beta}{\sin(\theta)} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

or:

Intrinsic parameters, homogeneous coordinates



$$u = \alpha \frac{x}{z} - \alpha \cot(\theta) \frac{y}{z} + u_0$$

$$v = \frac{\beta}{\sin(\theta)} \frac{y}{z} + v_0$$

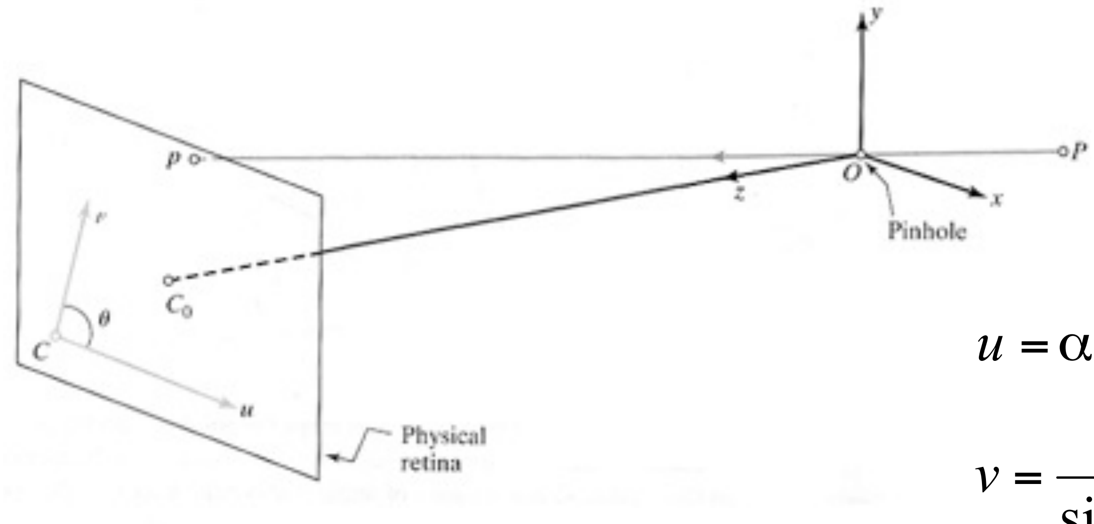
Using homogenous coordinates, we can write this as:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha & -\alpha\cot(\theta) & u_0 & 0 \\ 0 & \dfrac{\beta}{\sin(\theta)} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

or:

$$\vec{p} = \mathrm{K} \; {}^{C}\vec{p}$$

Intrinsic parameters, homogeneous coordinates



$$u = \alpha \frac{x}{z} - \alpha \cot(\theta)\frac{y}{z} + u_0$$

$$v = \frac{\beta}{\sin(\theta)}\frac{y}{z} + v_0$$

Using homogenous coordinates, we can write this as:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha & -\alpha\cot(\theta) & u_0 & 0 \\ 0 & \dfrac{\beta}{\sin(\theta)} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

or:

In pixels $\longrightarrow$ $\vec{p} = \mathrm{K}\ {}^{C}\vec{p}$

In camera-based coords

Extrinsic parameters: translation and rotation of camera frame

Extrinsic parameters: translation and rotation of camera frame

$$ {}^{C}\vec{p} = {}^{C}_{W}R \; {}^{W}\vec{p} + {}^{C}_{W}\vec{t} $$

Non-homogeneous coordinates

Extrinsic parameters: translation and rotation of camera frame

$$ {}^{C}\vec{p} = {}^{C}_{W}R \; {}^{W}\vec{p} + {}^{C}_{W}\vec{t} $$

Non-homogeneous coordinates

$$ \begin{pmatrix} \\ {}^{C}\vec{p} \\ \\ \\ \end{pmatrix} = \begin{pmatrix} \boxed{\begin{matrix} - & - & - \\ - & {}^{C}_{W}R & - \\ - & - & - \end{matrix}} & \boxed{\begin{matrix} | \\ {}^{C}_{W}\vec{t} \\ | \end{matrix}} \\ 0 \quad 0 \quad 0 & 1 \end{pmatrix} \begin{pmatrix} \\ {}^{W}\vec{p} \\ \\ \\ \end{pmatrix} $$

Homogeneous coordinates

Combining extrinsic and intrinsic calibration parameters, in homogeneous coordinates

Intrinsic

$$\vec{p} = \mathrm{K}\ {}^{C}\vec{p}$$

$$\begin{pmatrix} {}^{C}\vec{p} \\ \\ \end{pmatrix} = \begin{pmatrix} - & - & - & | \\ - & {}^{C}_{W}R & - & {}^{C}_{W}\vec{t} \\ - & - & - & | \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{W}\vec{p} \\ \\ \end{pmatrix}$$

Extrinsic

Forsyth&Ponce

Combining extrinsic and intrinsic calibration parameters, in homogeneous coordinates

Intrinsic

$$\vec{p} = \mathrm{K}\ {}^{C}\vec{p}$$

World coordinate

$$\begin{pmatrix} \\ {}^{C}\vec{p} \\ \\ \\ \end{pmatrix} = \begin{pmatrix} - & - & - & | \\ - & {}^{C}_{W}R & - & {}^{C}_{W}\vec{t} \\ - & - & - & | \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \\ {}^{W}\vec{p} \\ \\ \\ \end{pmatrix}$$

Extrinsic

Combining extrinsic and intrinsic calibration parameters, in homogeneous coordinates

Intrinsic

$$\vec{p} = \mathrm{K}\ {}^{C}\vec{p}$$

Camera coordinates

World coordinates

$$\begin{pmatrix} {}^{C}\vec{p} \end{pmatrix} = \begin{pmatrix} \boxed{\begin{matrix} - & - & - \\ - & {}^{C}_{W}R & - \\ - & - & - \end{matrix}} & \boxed{\begin{matrix} | \\ {}^{C}_{W}\vec{t} \\ | \end{matrix}} \\ \begin{matrix} 0 & 0 & 0 \end{matrix} & 1 \end{pmatrix} \begin{pmatrix} {}^{W}\vec{p} \end{pmatrix}$$

Extrinsic

Forsyth&Ponce

Combining extrinsic and intrinsic calibration parameters, in homogeneous coordinates

pixels

Intrinsic

$$\vec{p} = K \; {}^{C}\vec{p}$$

World coordinates

Camera coordinates

$$\begin{pmatrix} {}^{C}\vec{p} \end{pmatrix} = \begin{pmatrix} \boxed{\begin{matrix} - & - & - \\ - & {}^{C}_{W}R & - \\ - & - & - \end{matrix}} & \boxed{\begin{matrix} | \\ {}^{C}_{W}\vec{t} \\ | \end{matrix}} \\ 0 \quad 0 \quad 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{W}\vec{p} \end{pmatrix}$$

Extrinsic

Forsyth&Ponce

Combining extrinsic and intrinsic calibration parameters, in homogeneous coordinates

pixels

Intrinsic

$$\vec{p} = \mathrm{K}\ {}^{C}\vec{p}$$

World coordinates

Camera coordinates

$$\begin{pmatrix} \\ {}^{C}\vec{p} \\ \\ \end{pmatrix} = \begin{pmatrix} \boxed{\begin{matrix} - & - & - \\ - & {}^{C}_{W}R & - \\ - & - & - \end{matrix}} & \boxed{\begin{matrix} | \\ {}^{C}_{W}\vec{t} \\ | \end{matrix}} \\ 0 \quad 0 \quad 0 & 1 \end{pmatrix} \begin{pmatrix} \\ {}^{W}\vec{p} \\ \\ \end{pmatrix}$$

Extrinsic

$$\vec{p} = \underbrace{K \begin{pmatrix} {}^{C}_{W}R & {}^{C}_{W}\vec{t} \\ 0\,0\,0 & 1 \end{pmatrix}}\ {}^{W}\vec{p}$$

$$\vec{p} = M\ {}^{W}\vec{p}$$

Forsyth&Ponce

Other ways to write the same equation

pixel coordinates

world coordinates

$$\vec{p} = M \; {}^{W}\vec{p}$$

notational definition

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} . & m_1^T & . & . \\ . & m_2^T & . & . \\ . & m_3^T & . & . \end{pmatrix} \begin{pmatrix} {}^{W}p_x \\ {}^{W}p_y \\ {}^{W}p_z \\ 1 \end{pmatrix}$$

Other ways to write the same equation

pixel coordinates

world coordinates

$$\vec{p} = M \; {}^{W}\vec{p}$$

notational definition

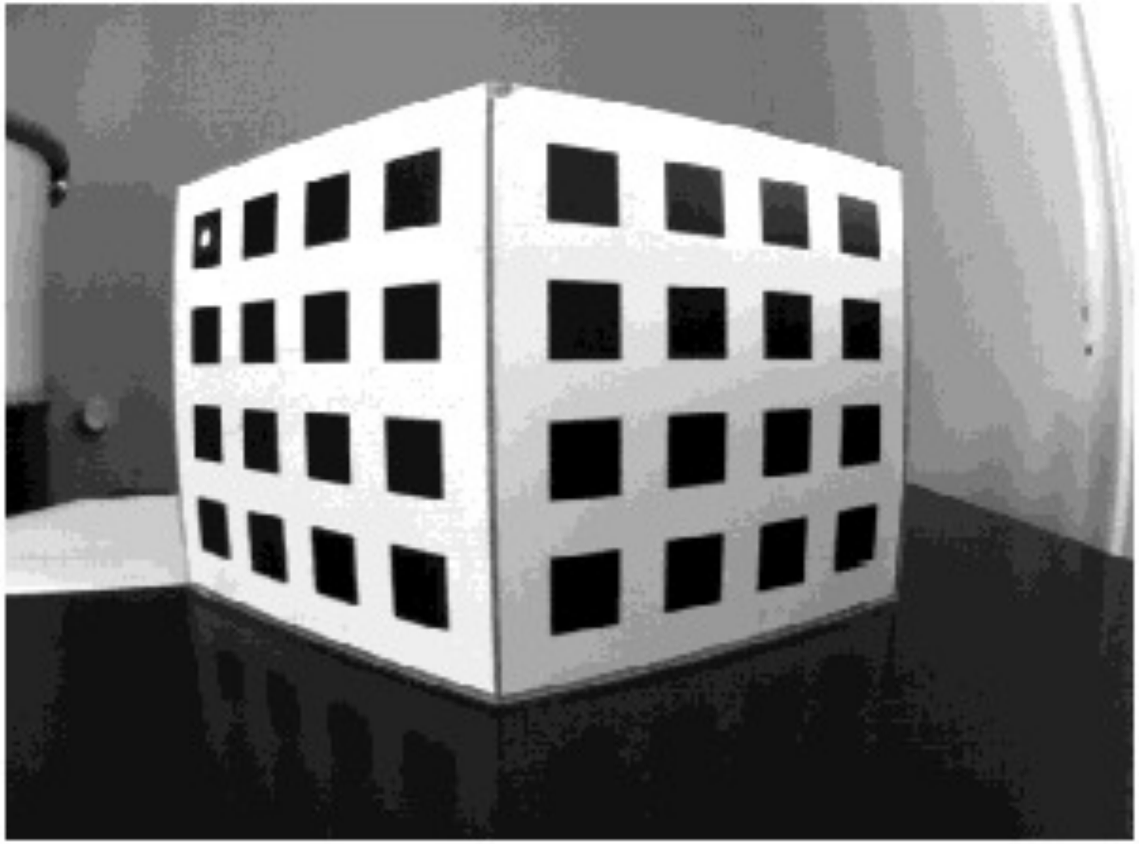
$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} . & m_1^T & . & . \\ . & m_2^T & . & . \\ . & m_3^T & . & . \end{pmatrix} \begin{pmatrix} {}^{W}p_x \\ {}^{W}p_y \\ {}^{W}p_z \\ 1 \end{pmatrix}$$

$\vec{m}_i$ is a vector containing the 4 elements of the ith row of the matrix M

$$\left\{ \begin{aligned} u &= \frac{\vec{m}_1 \cdot \vec{P}}{\vec{m}_3 \cdot \vec{P}} \\ v &= \frac{\vec{m}_2 \cdot \vec{P}}{\vec{m}_3 \cdot \vec{P}} \end{aligned} \right.$$

Conversion of the 2-d position back from homogeneous coordinates leads to:

Calibration target



The Opti-CAL Calibration Target Image

Find the position, u_i and v_i, in pixels,
of each calibration object feature point.

http://www.kinetic.bc.ca/CompVision/opti-CAL.html

Camera calibration

From before, we had these equations relating image positions, u,v, to points at 3-d positions P (in homogeneous coordinates):

$$u = \frac{\vec{m}_1 \cdot \vec{P}}{\vec{m}_3 \cdot \vec{P}}$$

$$v = \frac{\vec{m}_2 \cdot \vec{P}}{\vec{m}_3 \cdot \vec{P}}$$

Camera calibration

From before, we had these equations relating image positions, u,v, to points at 3-d positions P (in homogeneous coordinates):

$$u = \frac{\vec{m}_1 \cdot \vec{P}}{\vec{m}_3 \cdot \vec{P}}$$

$$v = \frac{\vec{m}_2 \cdot \vec{P}}{\vec{m}_3 \cdot \vec{P}}$$

So for each feature point, i, we have:

$$(\vec{m}_1 - u_i \vec{m}_3) \cdot \vec{P}_i = 0$$

$$(\vec{m}_2 - v_i \vec{m}_3) \cdot \vec{P}_i = 0$$

Camera calibration

Stack all these measurements of i=1…n points

$$(\vec{m}_1 - u_i \vec{m}_3) \cdot \vec{P}_i = 0$$

$$(\vec{m}_2 - v_i \vec{m}_3) \cdot \vec{P}_i = 0$$

into a big matrix (cluttering vector arrows omitted from P and m):

$$\begin{pmatrix} P_1^T & 0^T & -u_1 P_1^T \\ 0^T & P_1^T & -v_1 P_1^T \\ \cdots & \cdots & \cdots \\ P_n^T & 0^T & -u_n P_n^T \\ 0^T & P_n^T & -v_n P_n^T \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

Camera calibration

In vector form:
$$\begin{pmatrix} P_1^T & 0^T & -u_1 P_1^T \\ 0^T & P_1^T & -v_1 P_1^T \\ \cdots & \cdots & \cdots \\ P_n^T & 0^T & -u_n P_n^T \\ 0^T & P_n^T & -v_n P_n^T \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

Showing all the elements:
$$\begin{pmatrix} P_{1x} & P_{1y} & P_{1z} & 1 & 0 & 0 & 0 & 0 & -u_1 P_{1x} & -u_1 P_{1y} & -u_1 P_{1z} & -u_1 \\ 0 & 0 & 0 & 0 & P_{1x} & P_{1y} & P_{1z} & 1 & -v_1 P_{1x} & -v_1 P_{1y} & -v_1 P_{1z} & -v_1 \\ & & & & & \cdots & \cdots & \cdots & & & & \\ P_{nx} & P_{ny} & P_{nz} & 1 & 0 & 0 & 0 & 0 & -u_n P_{nx} & -u_n P_{ny} & -u_n P_{nz} & -u_n \\ 0 & 0 & 0 & 0 & P_{nx} & P_{ny} & P_{nz} & 1 & -v_n P_{nx} & -v_n P_{ny} & -v_n P_{nz} & -v_n \end{pmatrix} \begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

Camera calibration

$$\begin{pmatrix} P_{1x} & P_{1y} & P_{1z} & 1 & 0 & 0 & 0 & 0 & -u_1P_{1x} & -u_1P_{1y} & -u_1P_{1z} & -u_1 \\ 0 & 0 & 0 & 0 & P_{1x} & P_{1y} & P_{1z} & 1 & -v_1P_{1x} & -v_1P_{1y} & -v_1P_{1z} & -v_1 \\ & & & & & & \cdots & \cdots & \cdots & & & \\ P_{nx} & P_{ny} & P_{nz} & 1 & 0 & 0 & 0 & 0 & -u_nP_{nx} & -u_nP_{ny} & -u_nP_{nz} & -u_n \\ 0 & 0 & 0 & 0 & P_{nx} & P_{ny} & P_{nz} & 1 & -v_nP_{nx} & -v_nP_{ny} & -v_nP_{nz} & -v_n \end{pmatrix} \begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

Q m = 0

We want to solve for the unit vector m (the stacked one) that minimizes $|Qm|^2$

The minimum eigenvector of the matrix Q^TQ gives us that (see Forsyth&Ponce, 3.1), because it is the unit vector x that minimizes $x^T Q^TQ\ x$.

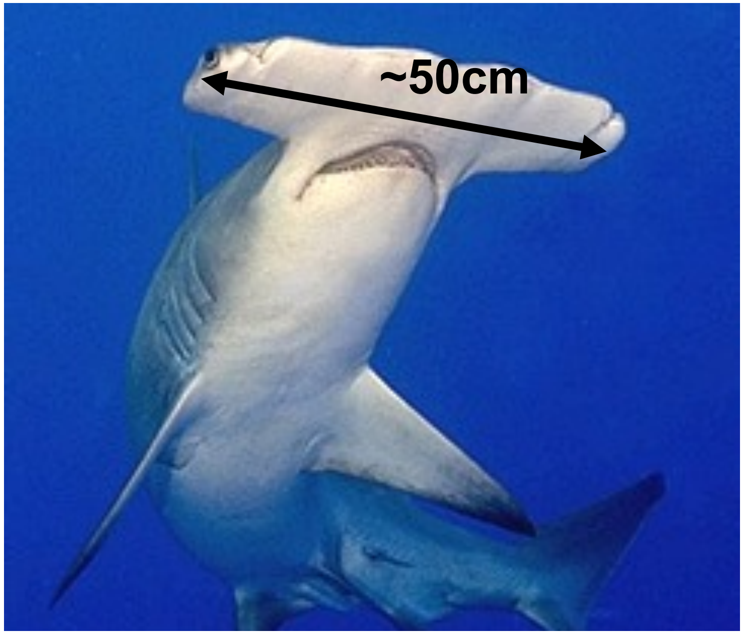
Once you have the M matrix, can recover the intrinsic and extrinsic parameters.

$$\mathcal{M} = \begin{pmatrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T & \alpha t_x - \alpha \cot\theta t_y + u_0 t_z \\ \dfrac{\beta}{\sin\theta} \boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T & \dfrac{\beta}{\sin\theta} t_y + v_0 t_z \\ \boldsymbol{r}_3^T & t_z \end{pmatrix}$$

Stereo vision

Stereo vision

Stereo vision



Depth without objects

Random dot stereograms (Bela Julesz)

Depth without objects

Random dot stereograms (Bela Julesz)

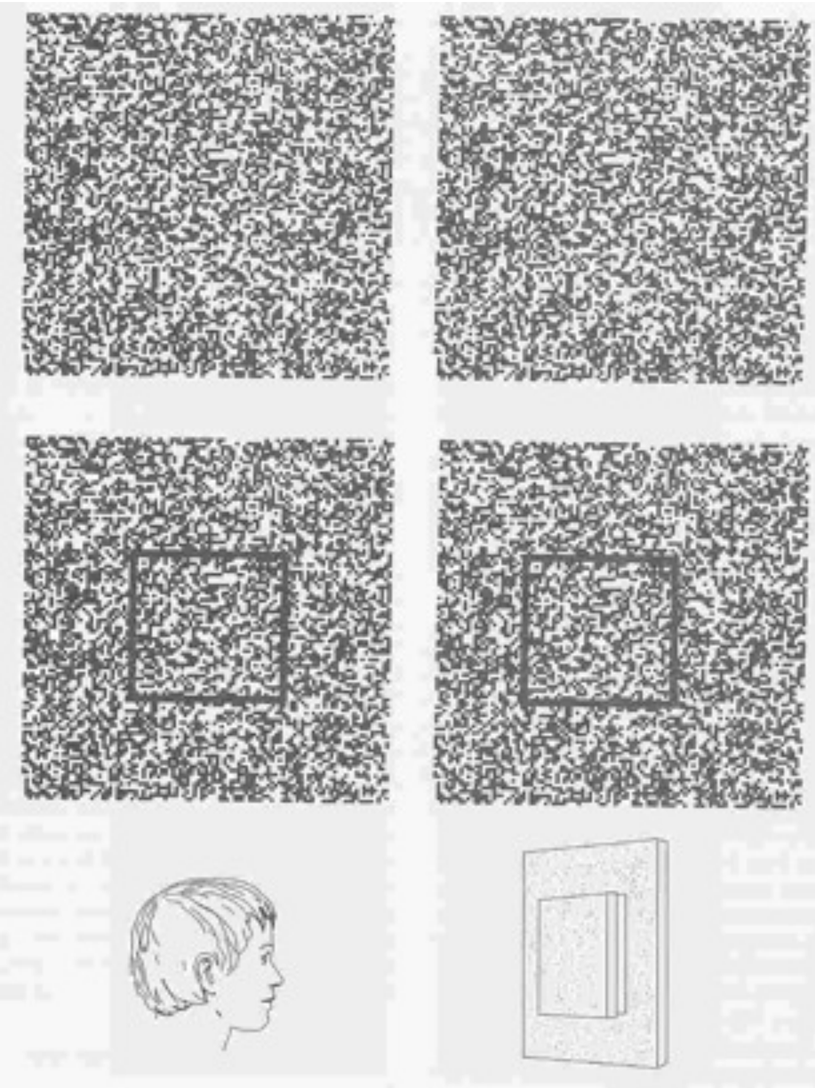


FIGURE 8.13

Depth without objects

Random dot stereograms (Bela Julesz)

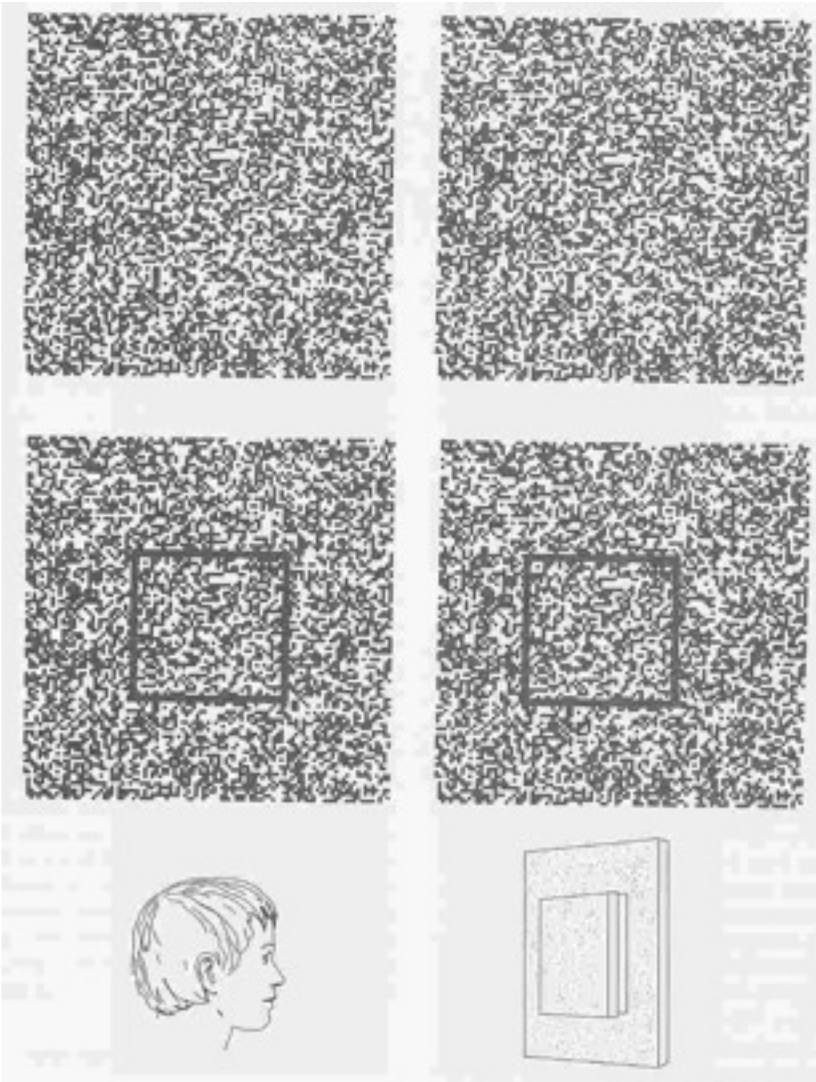
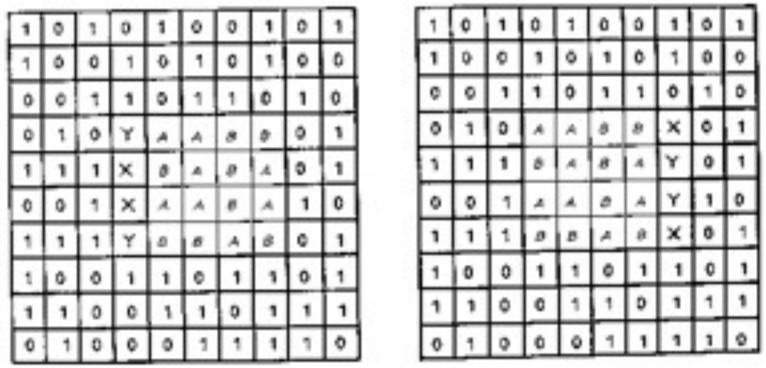


FIGURE 8.13



Julesz, 1971

Depth for familiar objects

http://www.youtube.com/watchv=G_Qwp2GdB1M

(Gregory 1970; Hill and Bruce 1993, 1994; Papathomas and DeCarlo 1999)

Depth for familiar objects

http://www.youtube.com/watchv=G_Qwp2GdB1M

(Gregory 1970; Hill and Bruce 1993, 1994; Papathomas and DeCarlo 1999)

Stereo photography and stereo viewers

Take two pictures of the same subject from two slightly different viewpoints and display so that each eye sees only one of the images.

Invented by Sir Charles Wheatstone, 1838

Slide credit: Kristen Grauman

Stereo photography and stereo viewers

Take two pictures of the same subject from two slightly different viewpoints and display so that each eye sees only one of the images.

Invented by Sir Charles Wheatstone, 1838

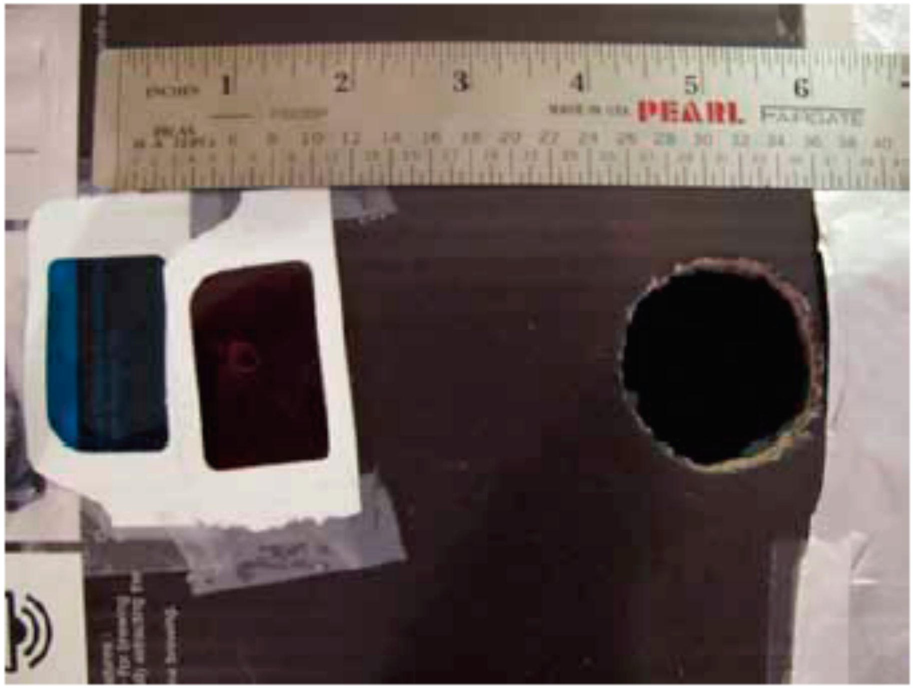
Image courtesy of fisher-price.com

Slide credit: Kristen Grauman

Public Library, Stereoscopic Looking Room, Chicago, by Phillips, 1923

de credit: Kristen Grauman

Anaglyph pinhole camera



Autostereograms

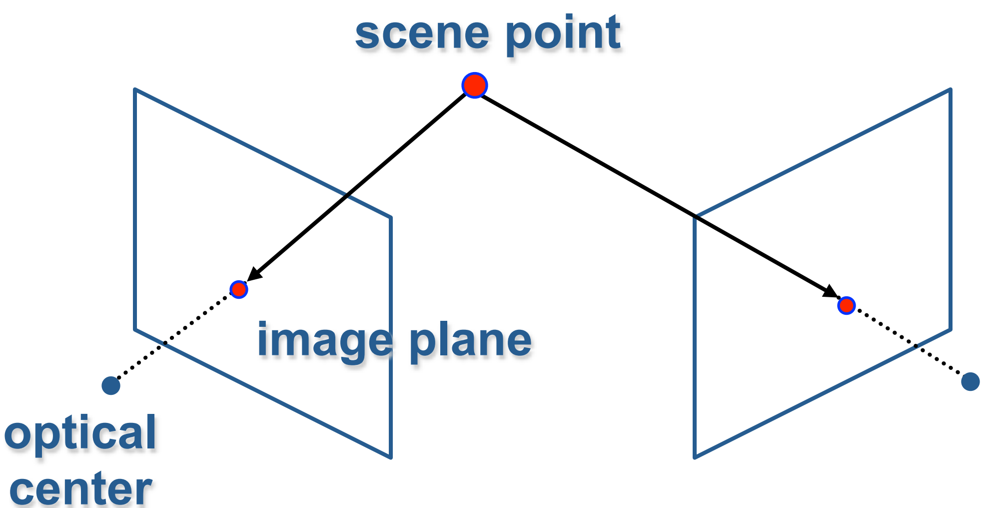
Exploit disparity as depth cue using single image.

(Single image random dot stereogram, Single image stereogram)

Images from magiceye.com

Slide credit: Kristen Grauman

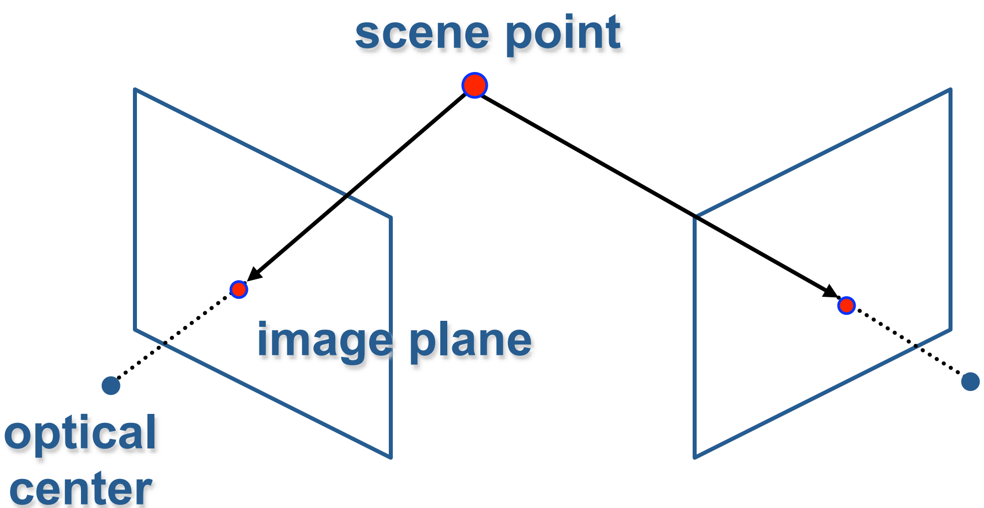
Estimating depth with stereo



Slide credit: Kristen Grauman

Estimating depth with stereo

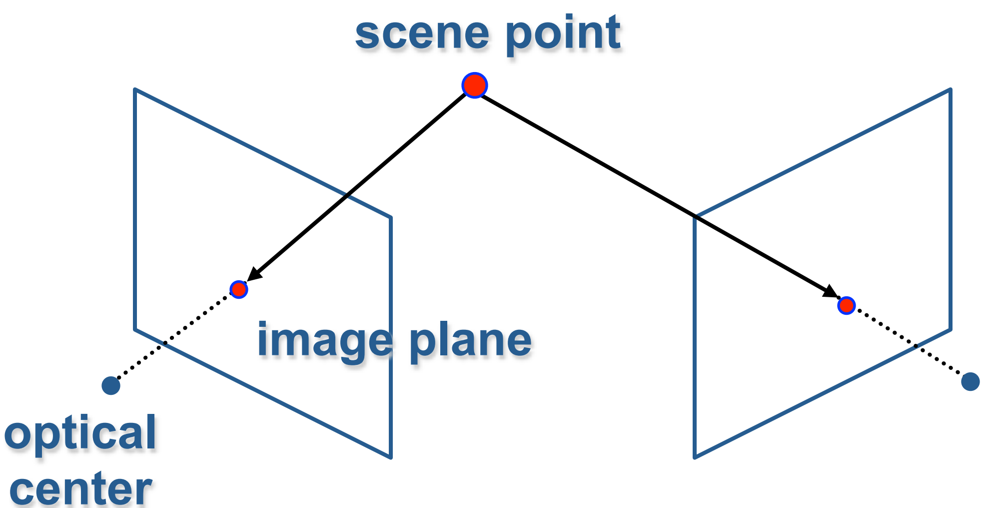
- **Stereo**: shape from disparities between two views



Slide credit: Kristen Grauman

Estimating depth with stereo

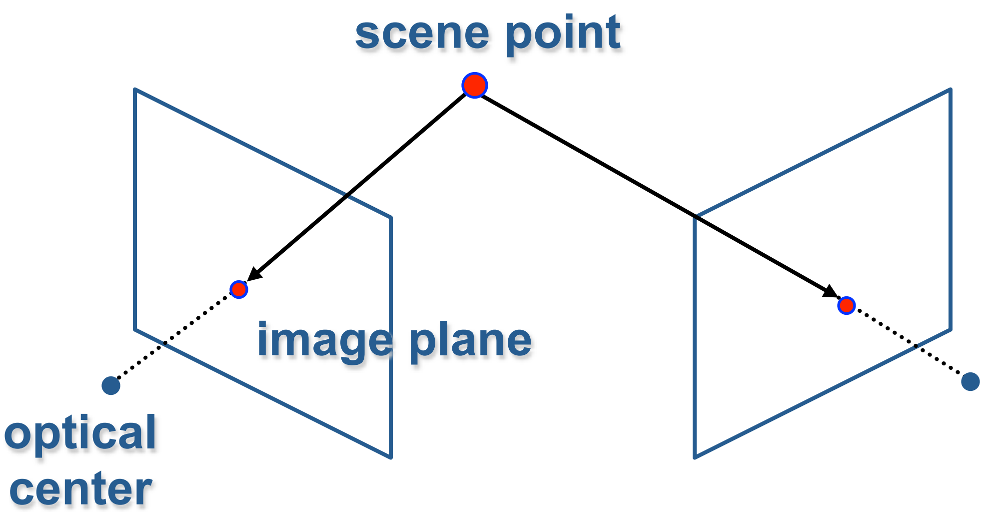
- **Stereo**: shape from disparities between two views
- We'll need to consider:



Slide credit: Kristen Grauman

Estimating depth with stereo

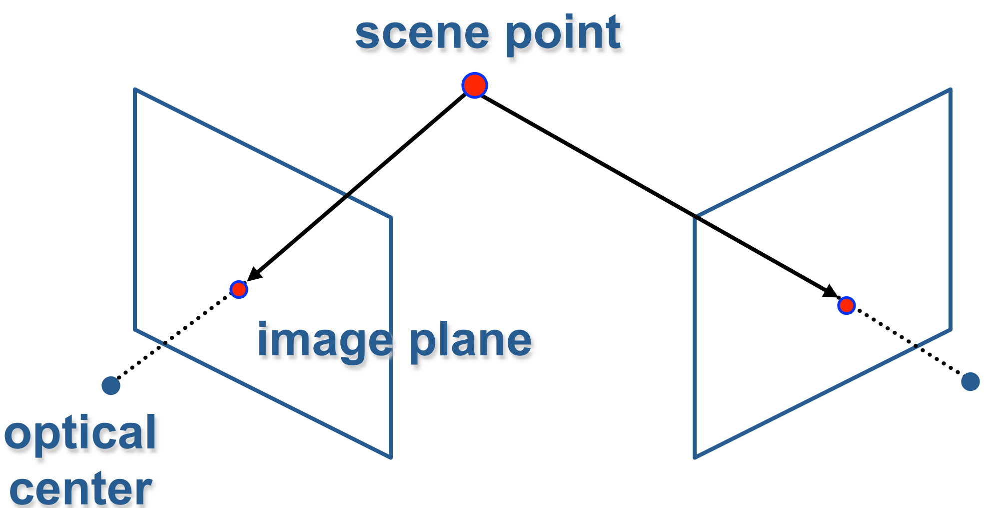
- **Stereo**: shape from disparities between two views
- We'll need to consider:
- Info on camera pose ("calibration")

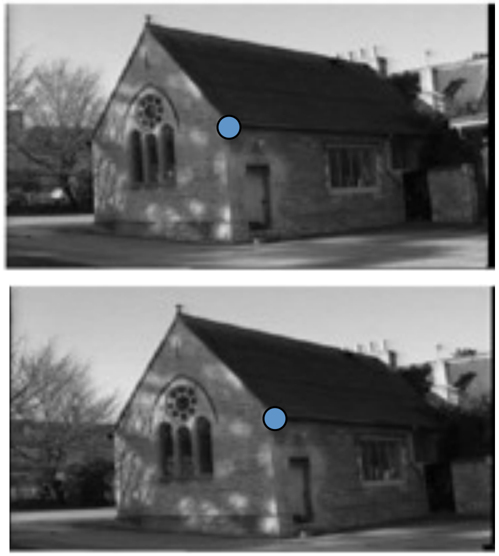


Slide credit: Kristen Grauman

Estimating depth with stereo

- **Stereo**: shape from disparities between two views
- We'll need to consider:
- Info on camera pose ("calibration")
- Image point correspondences





Slide credit: Kristen Grauman

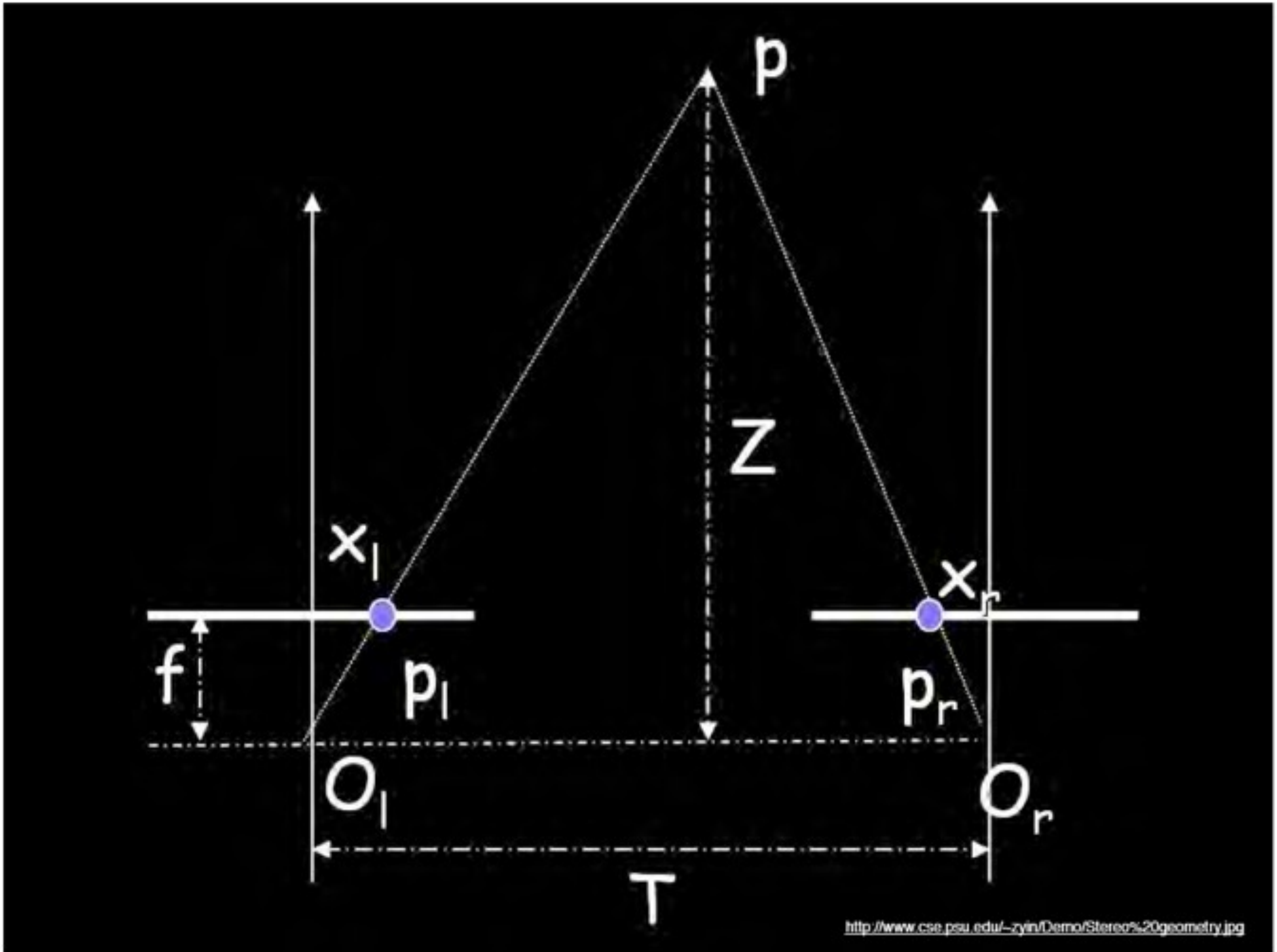
Stereo Topics

- **Special, simple system, main idea**
- More general camera conditions, epipolar constraints
  - epipolar geometry
  - epipolar algebra
- Image rectification
- Stereo matching (likelihood term)
- Stereo regularization (prior term)
- Inference
  - dynamic programming
  - graph cuts
- Structured light

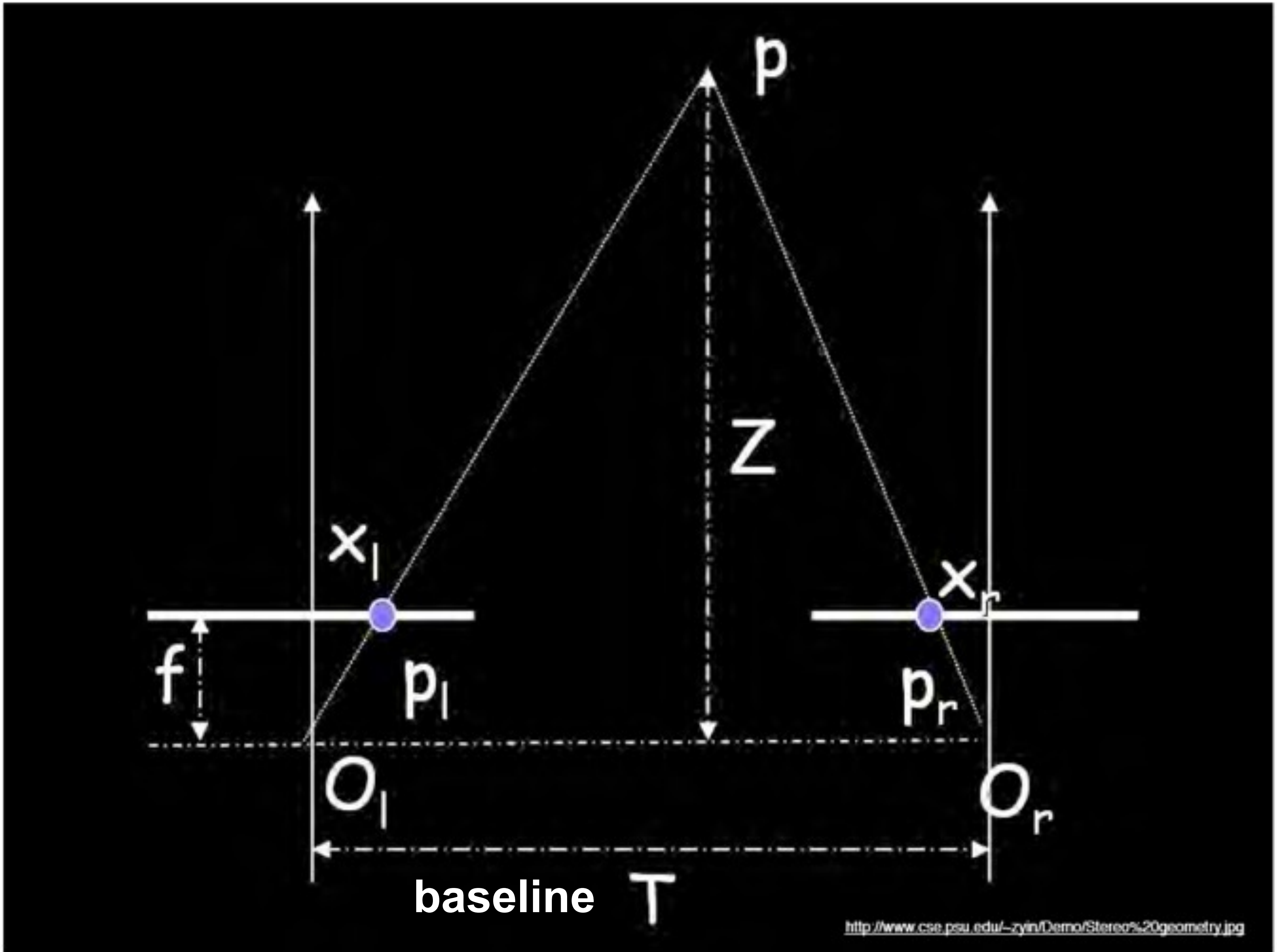
Geometry for a simple stereo system

- First, assuming parallel optical axes, known camera parameters (i.e., calibrated cameras):

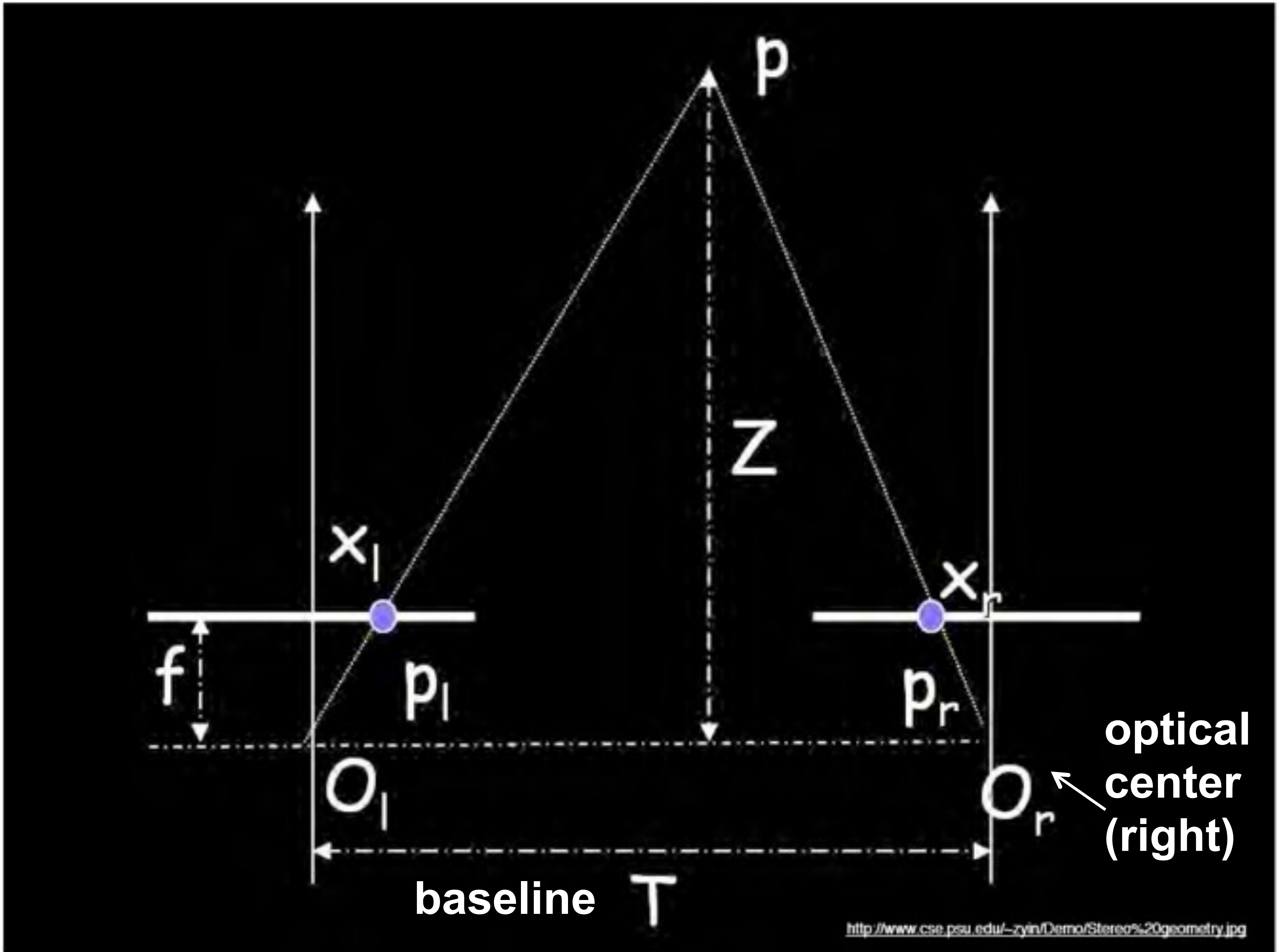
Slide credit: Kristen Grauman



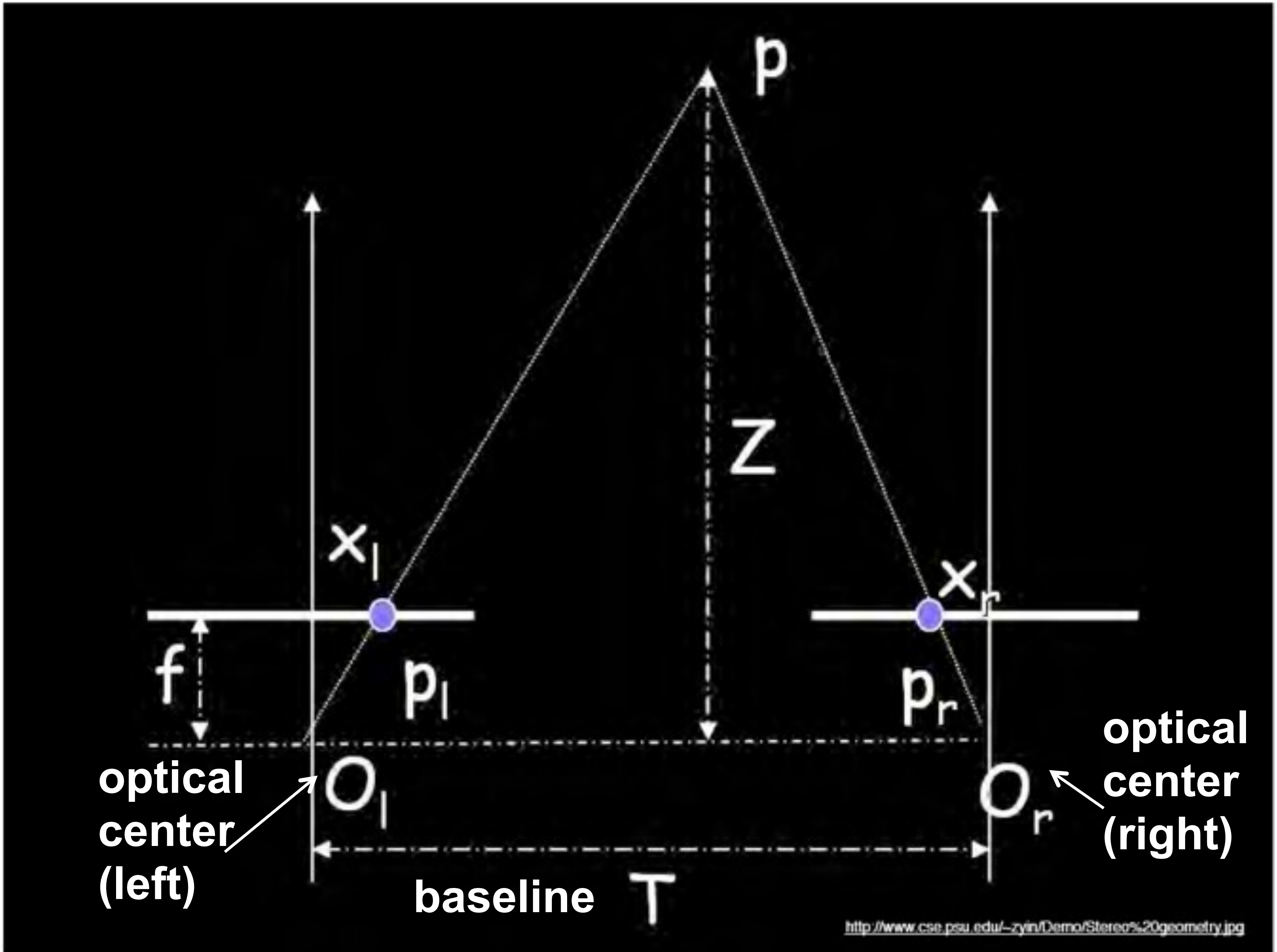
Slide credit: Kristen Grauman



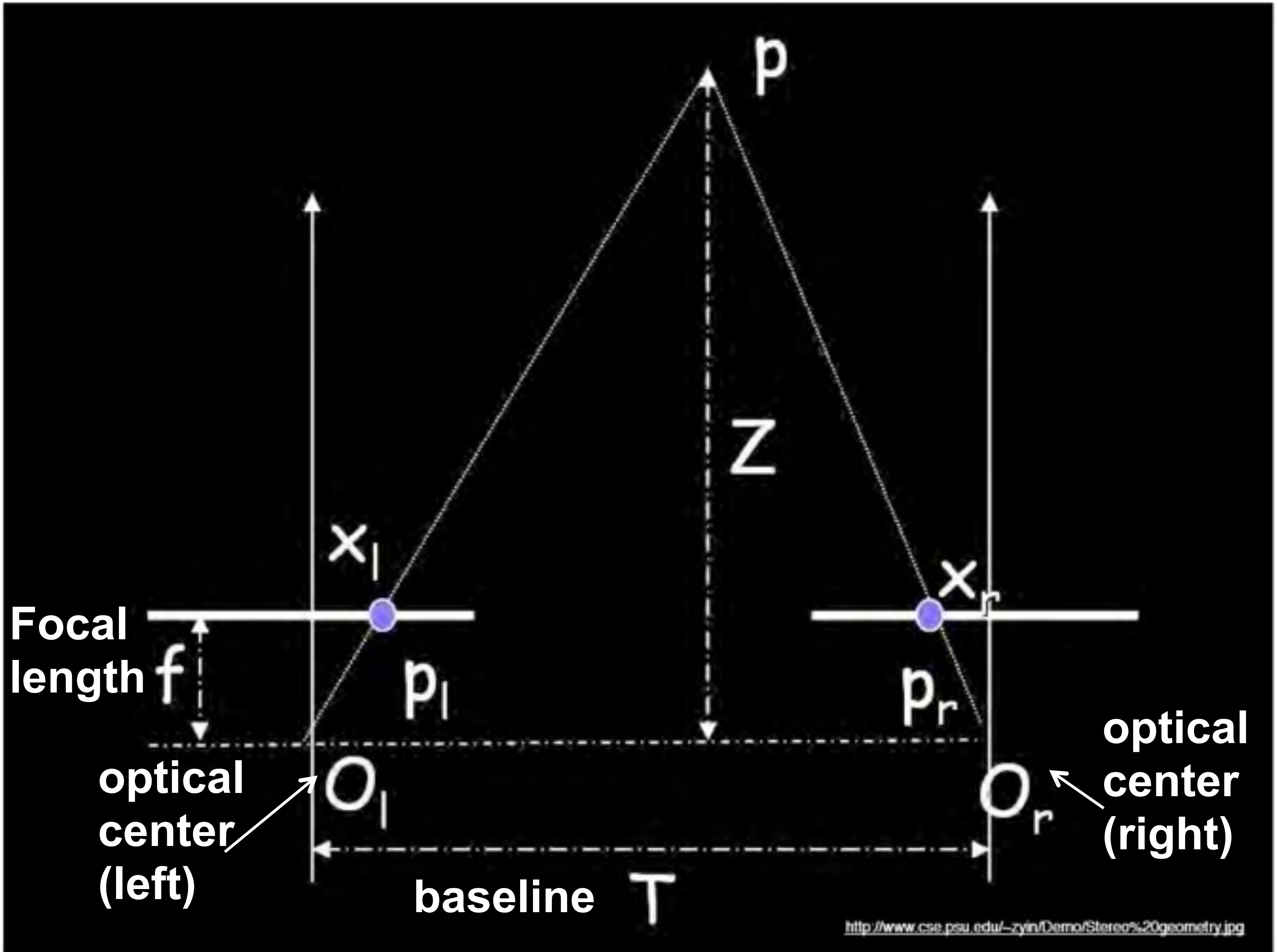
Slide credit: Kristen Grauman



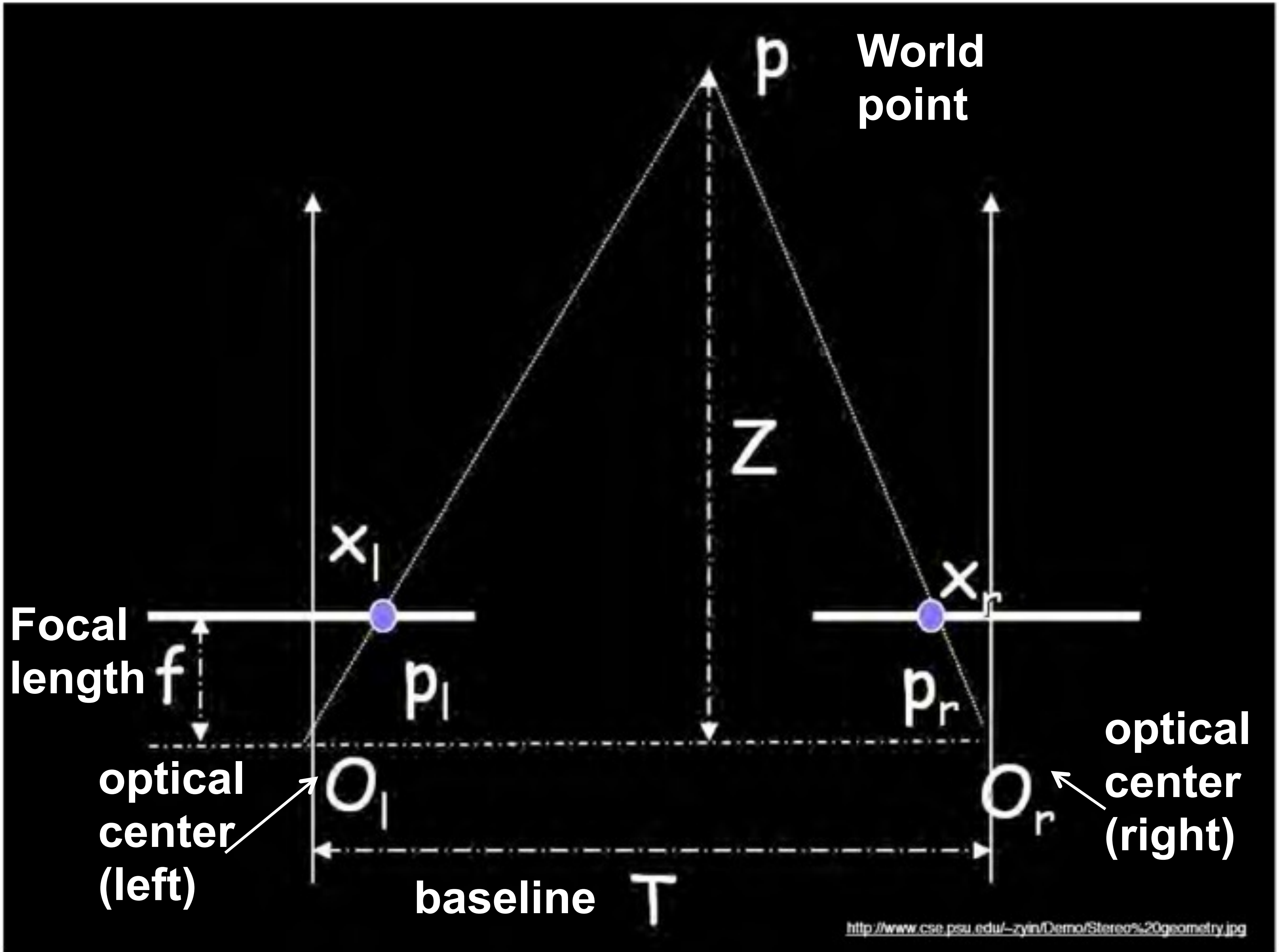
Slide credit: Kristen Grauman



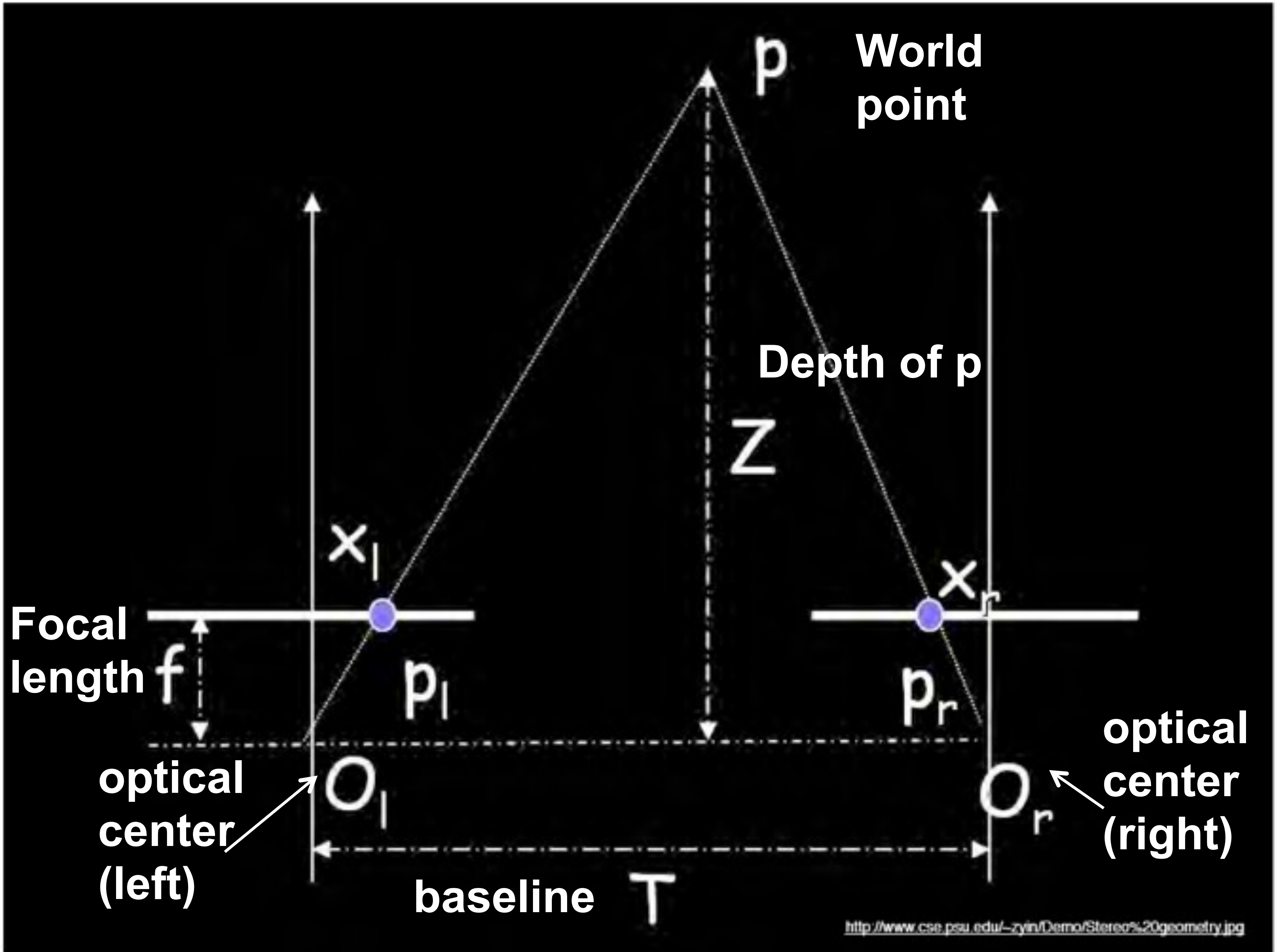
Slide credit: Kristen Grauman



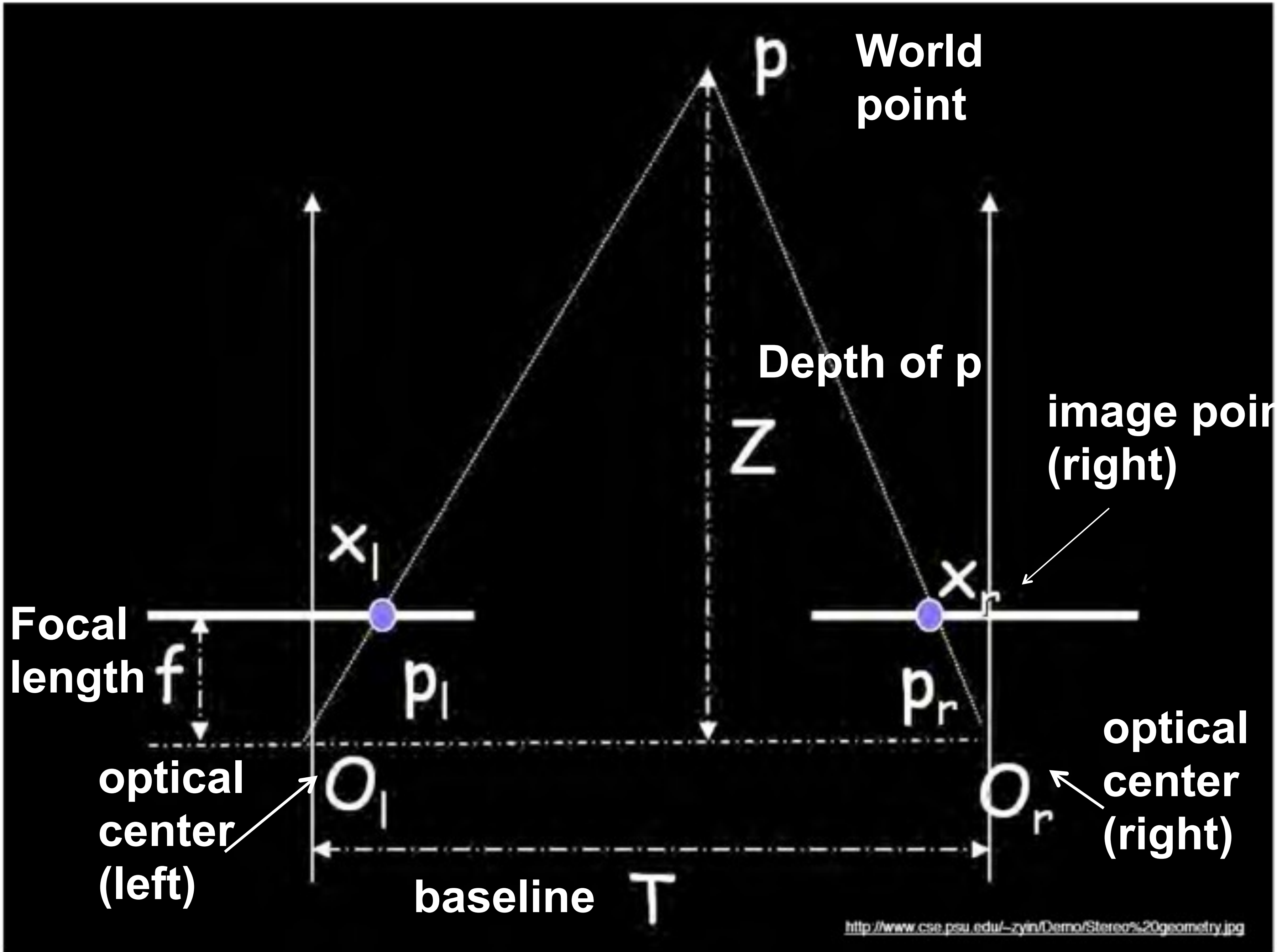
Slide credit: Kristen Grauman



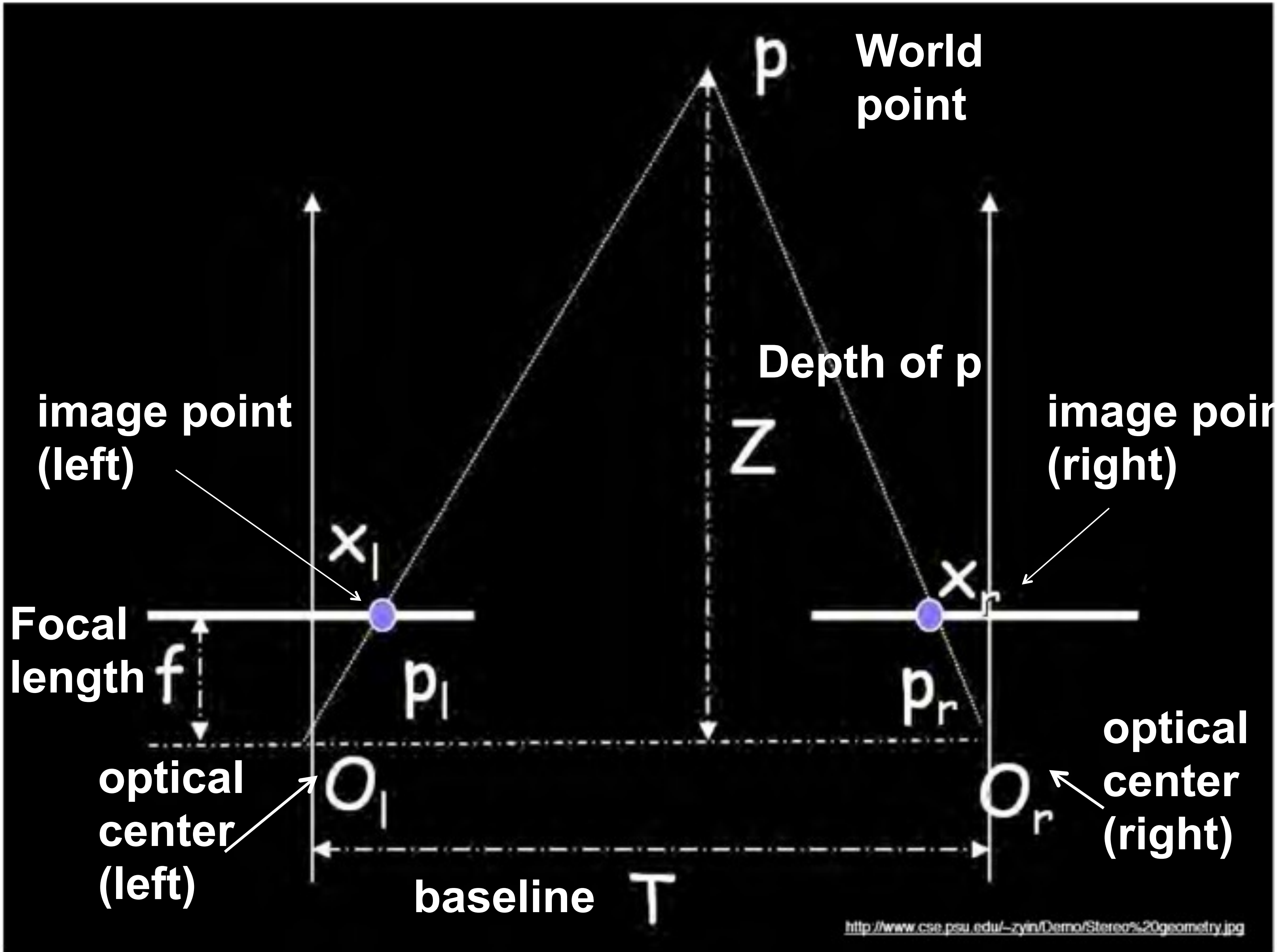
Slide credit: Kristen Grauman



Slide credit: Kristen Grauman



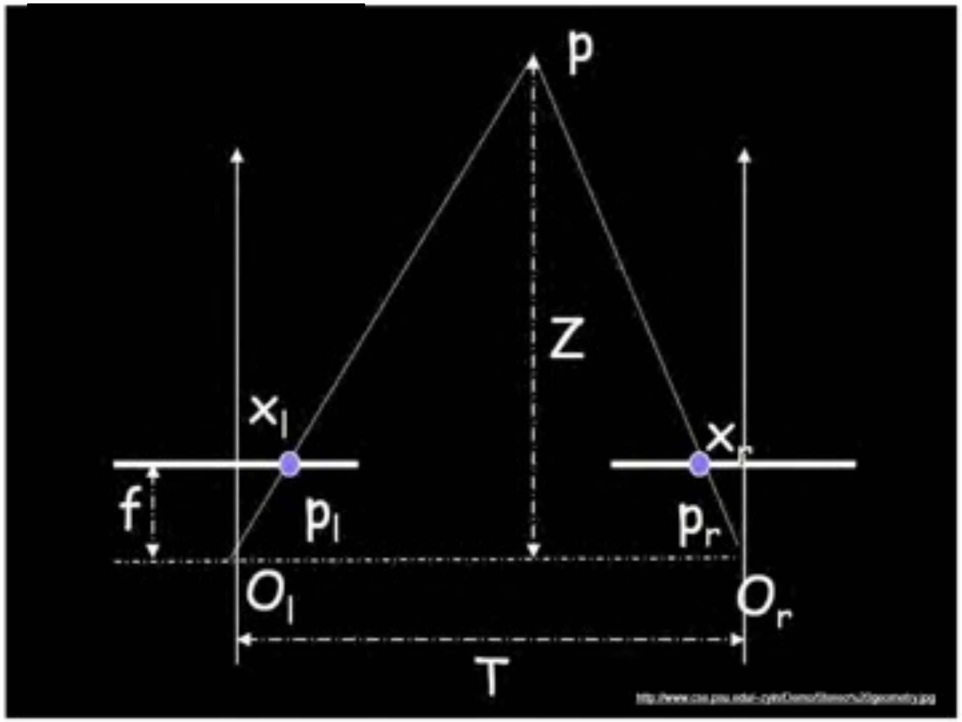
Slide credit: Kristen Grauman



Slide credit: Kristen Grauman

Geometry for a simple stereo system

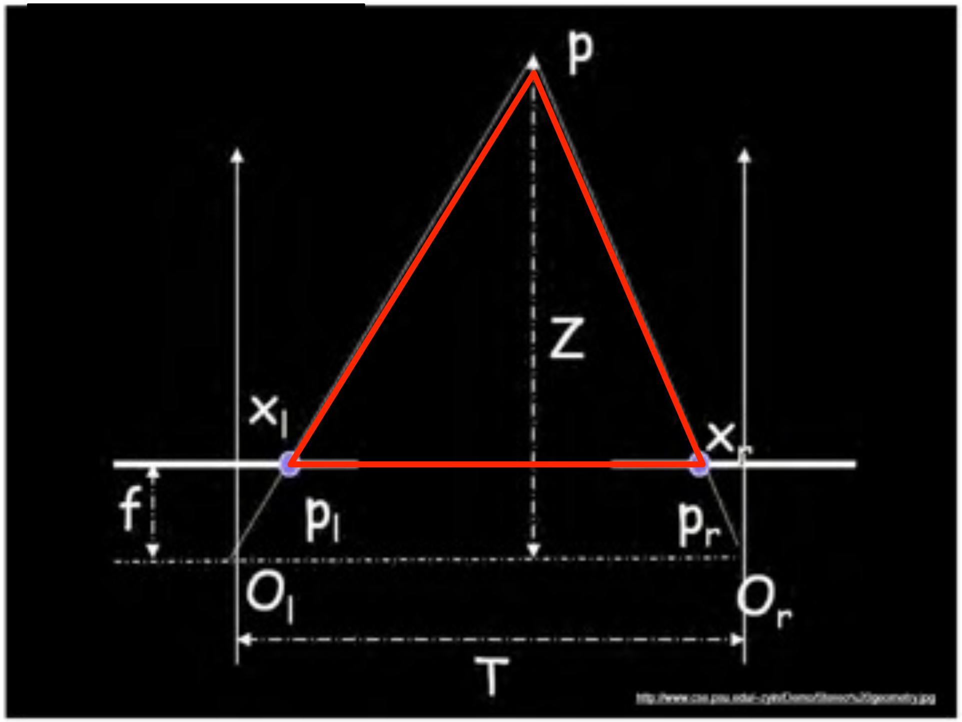
- Assume parallel optical axes, known camera parameters (i.e., calibrated cameras). We can triangulate via:



Slide credit: Kristen Grauman

Geometry for a simple stereo system

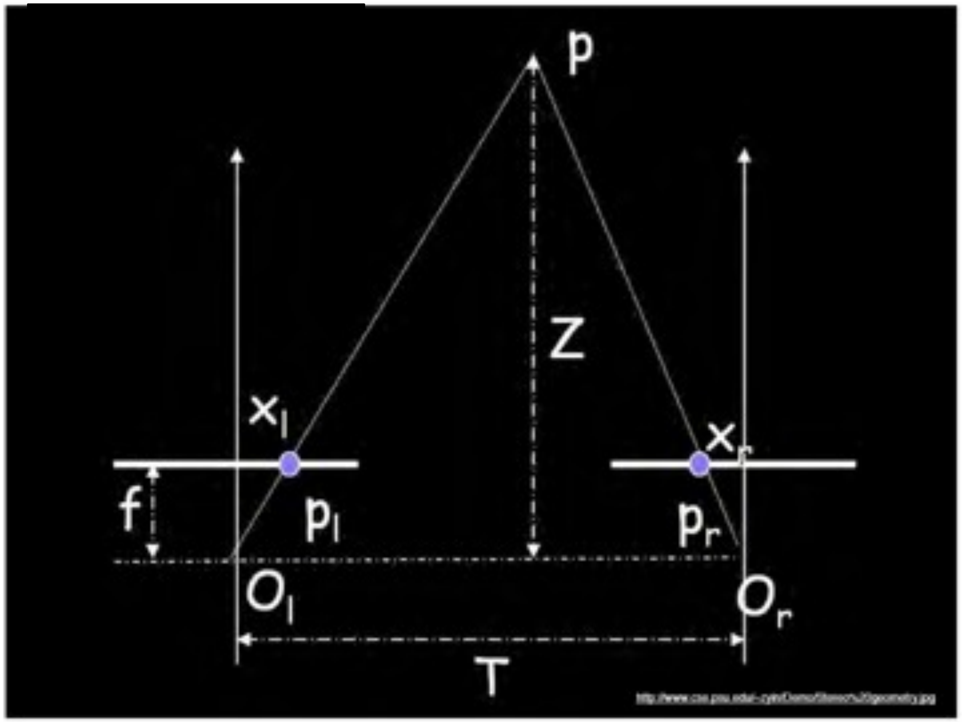
- Assume parallel optical axes, known camera parameters (i.e., calibrated cameras). We can triangulate via:



Slide credit: Kristen Grauman

Geometry for a simple stereo system

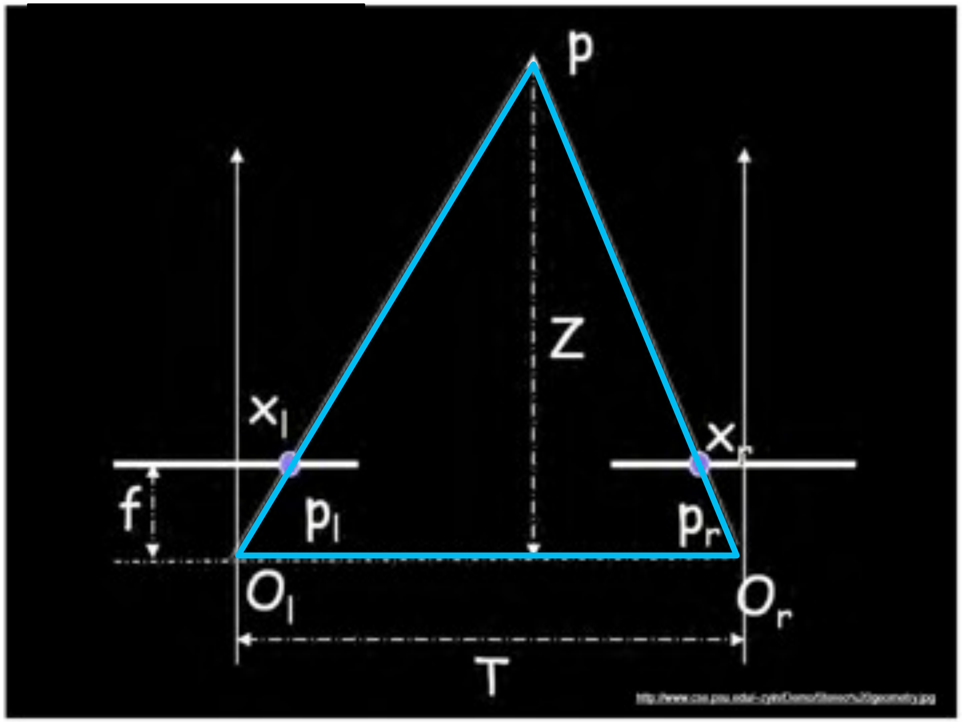
- Assume parallel optical axes, known camera parameters (i.e., calibrated cameras). We can triangulate via:



Slide credit: Kristen Grauman

Geometry for a simple stereo system

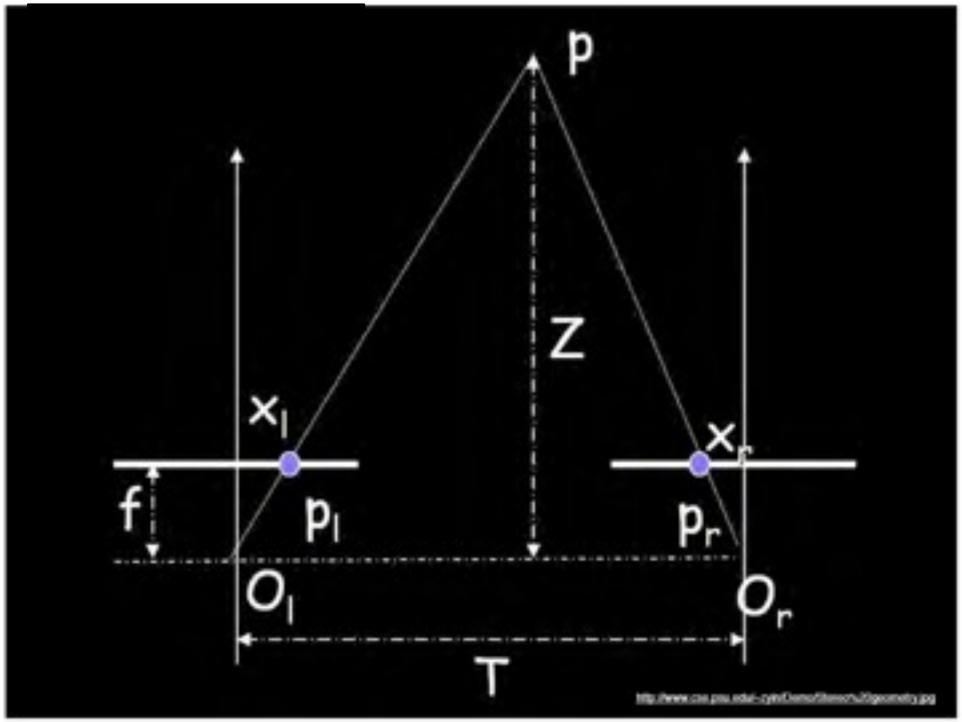
- Assume parallel optical axes, known camera parameters (i.e., calibrated cameras). We can triangulate via:



Slide credit: Kristen Grauman

Geometry for a simple stereo system

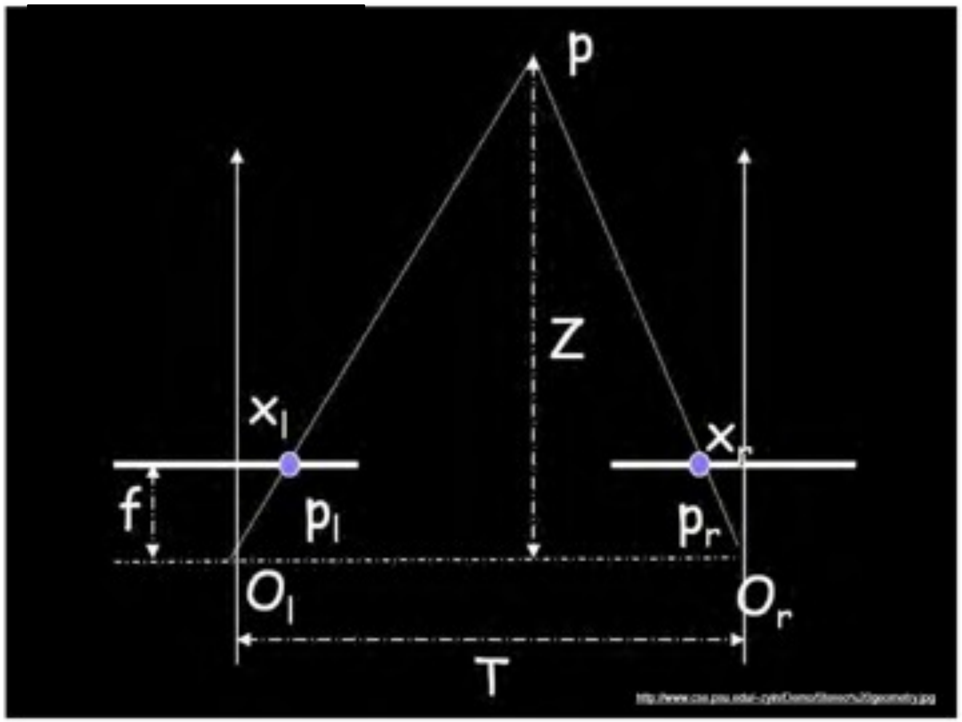
- Assume parallel optical axes, known camera parameters (i.e., calibrated cameras). We can triangulate via:



Slide credit: Kristen Grauman

Geometry for a simple stereo system

- Assume parallel optical axes, known camera parameters (i.e., calibrated cameras). We can triangulate via:

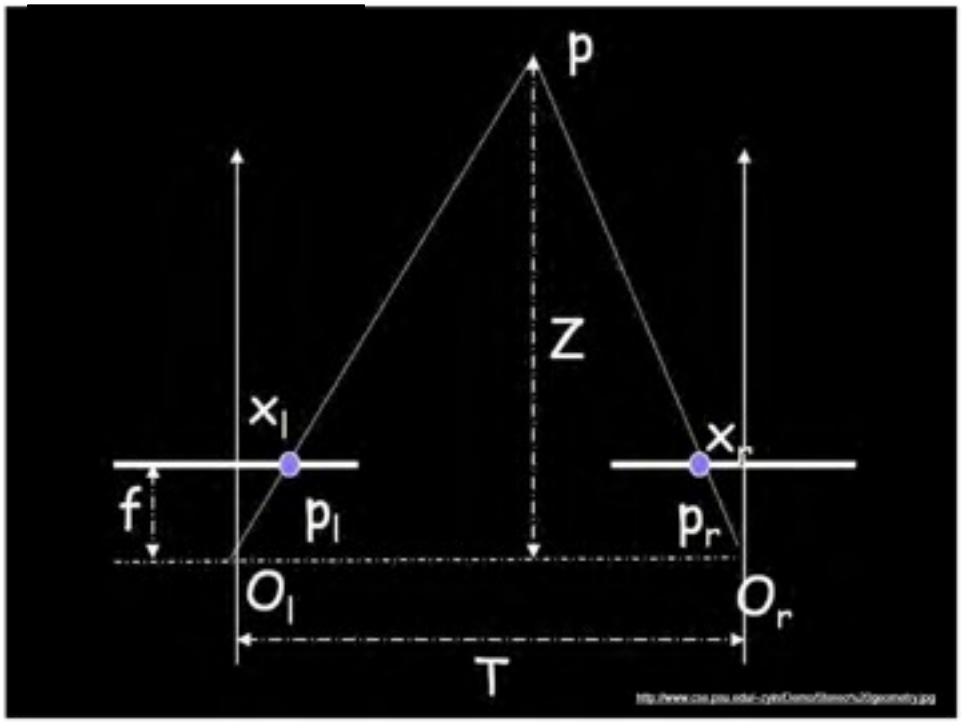


Similar triangles (p_l, P, p_r) and (O_l, P, O_r):

Slide credit: Kristen Grauman

Geometry for a simple stereo system

- Assume parallel optical axes, known camera parameters (i.e., calibrated cameras). We can triangulate via:

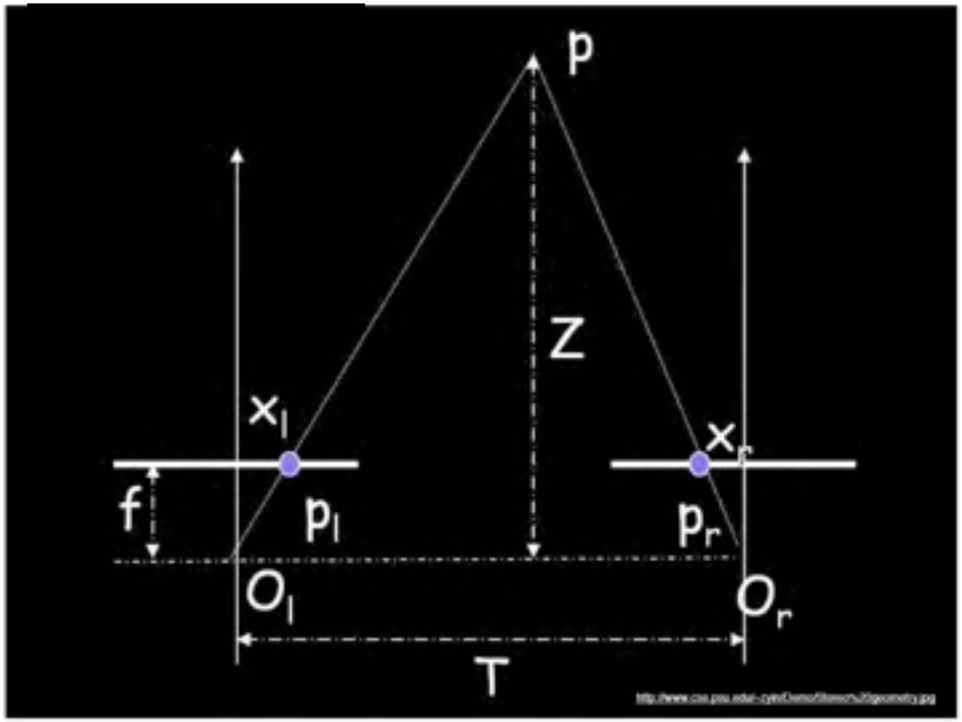


Similar triangles (p_l, P, p_r) and (O_l, P, O_r):

Slide credit: Kristen Grauman

Geometry for a simple stereo system

- Assume parallel optical axes, known camera parameters (i.e., calibrated cameras). We can triangulate via:



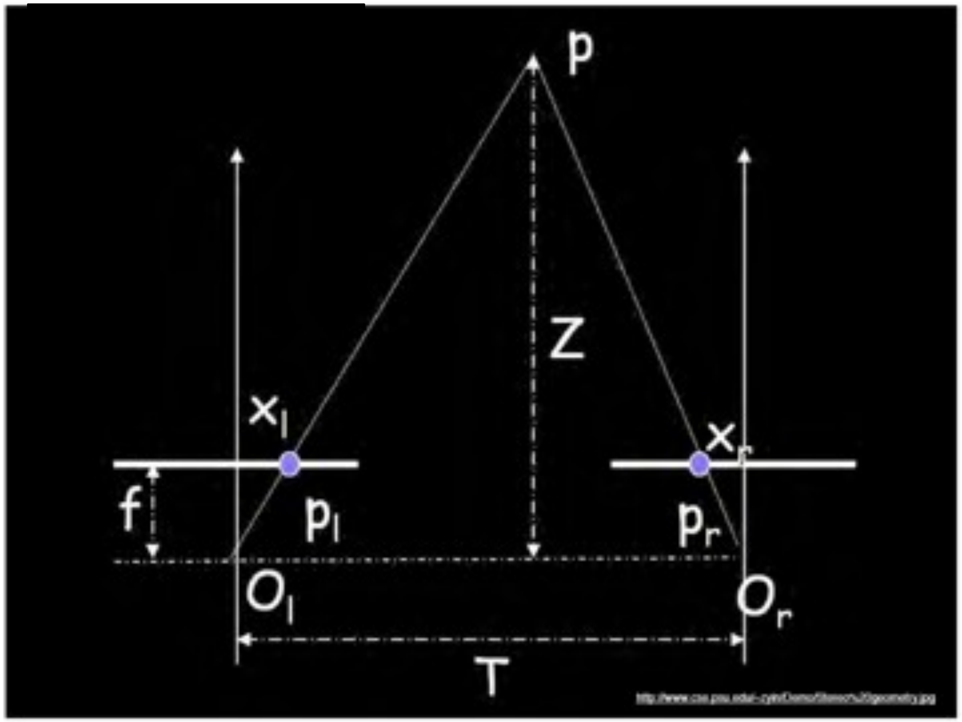
Similar triangles (p_l, P, p_r) and (O_l, P, O_r):

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

Slide credit: Kristen Grauman

Geometry for a simple stereo system

- Assume parallel optical axes, known camera parameters (i.e., calibrated cameras). We can triangulate via:



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

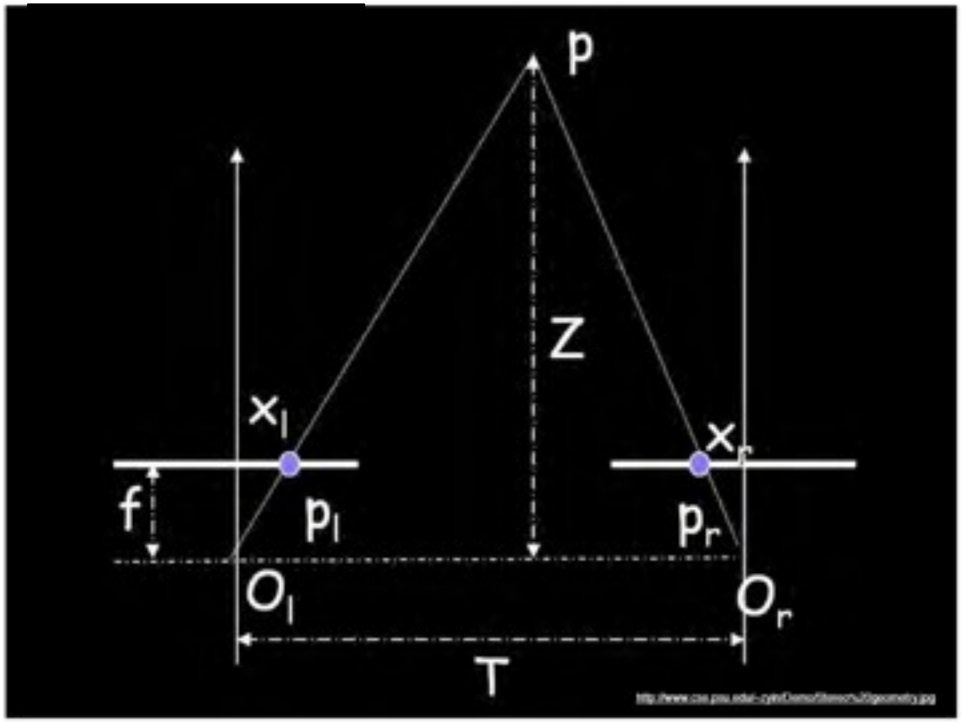
$$\frac{T + x_l - x_r}{Z - f} = \frac{T}{Z}$$

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

Slide credit: Kristen Grauman

Geometry for a simple stereo system

- Assume parallel optical axes, known camera parameters (i.e., calibrated cameras). We can triangulate via:



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

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

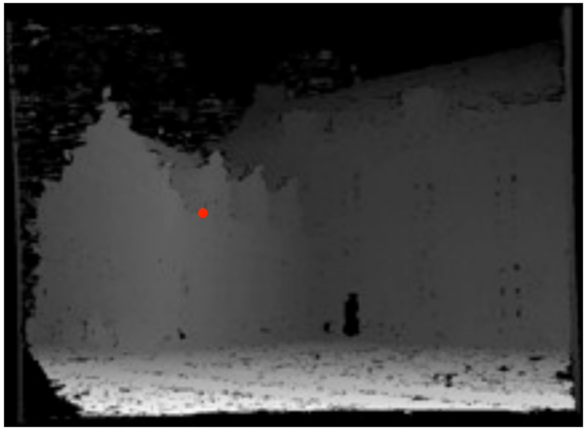
$$Z = f\frac{T}{x_r - x_l}$$

disparity

Slide credit: Kristen Grauman

Depth from disparity

image I(x,y) **Disparity map D(x,y)** **image I´(x´,y´)**



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

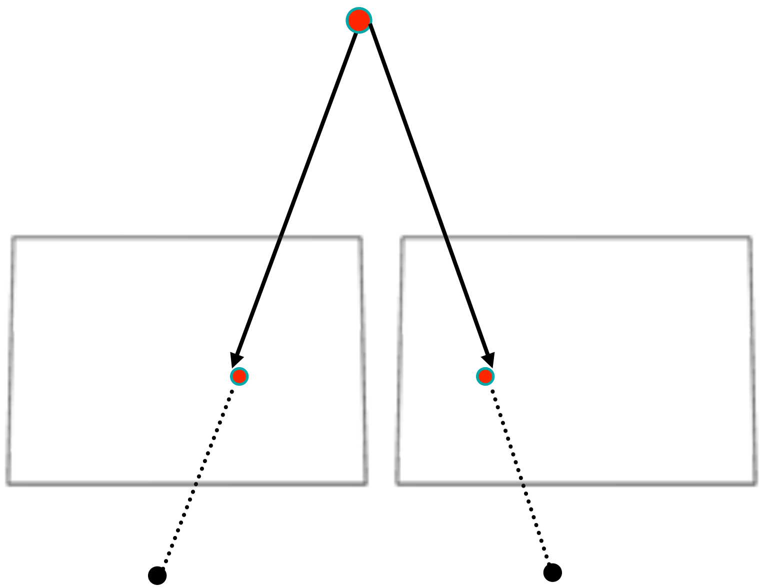
Slide credit: Kristen Grauman

Stereo Topics

- Special, simple system, main idea
- **More general camera conditions, epipolar constraints**
  - **epipolar geometry**
  - epipolar algebra
- Image rectification
- Stereo matching (likelihood term)
- Stereo regularization (prior term)
- Inference
  - dynamic programming
  - graph cuts
- Structured light

General case, with calibrated cameras

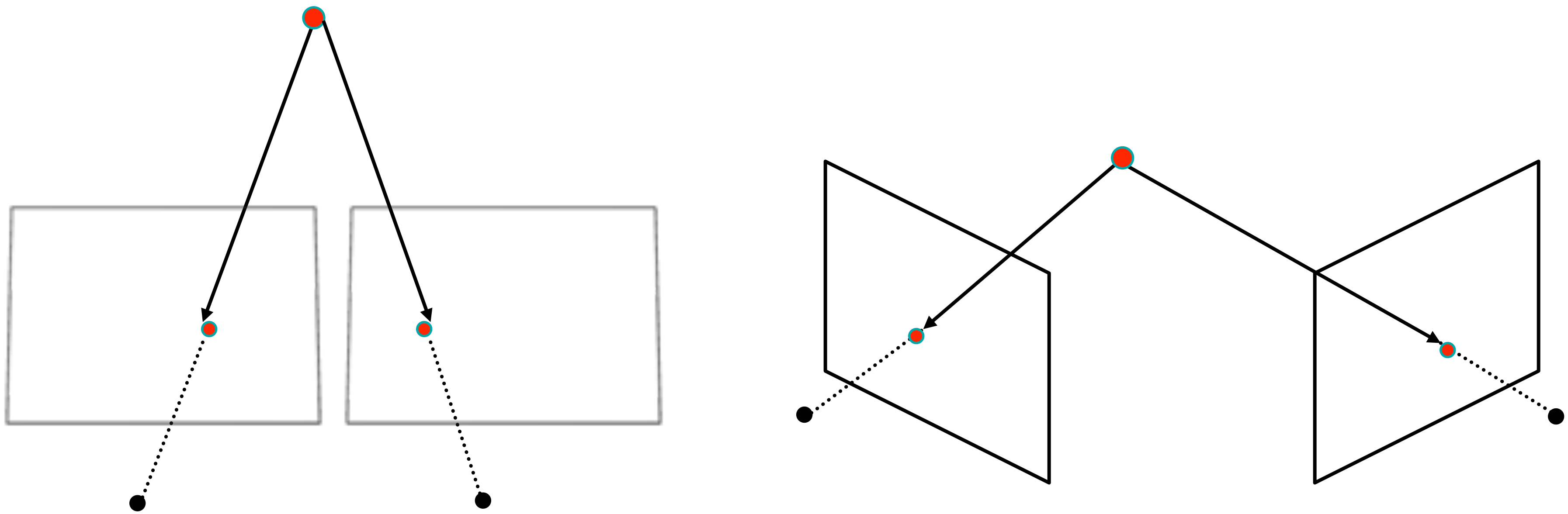
- The two cameras need not have parallel optical axes.



Slide credit: Kristen Grauman

General case, with calibrated cameras

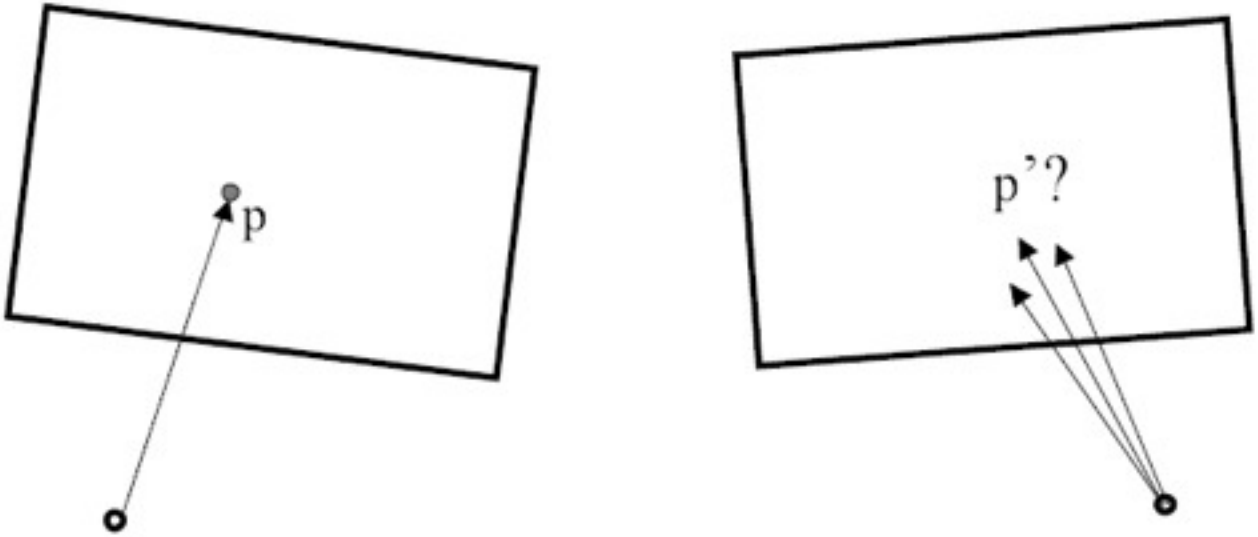
- The two cameras need not have parallel optical axes.



Vs.

Slide credit: Kristen Grauman

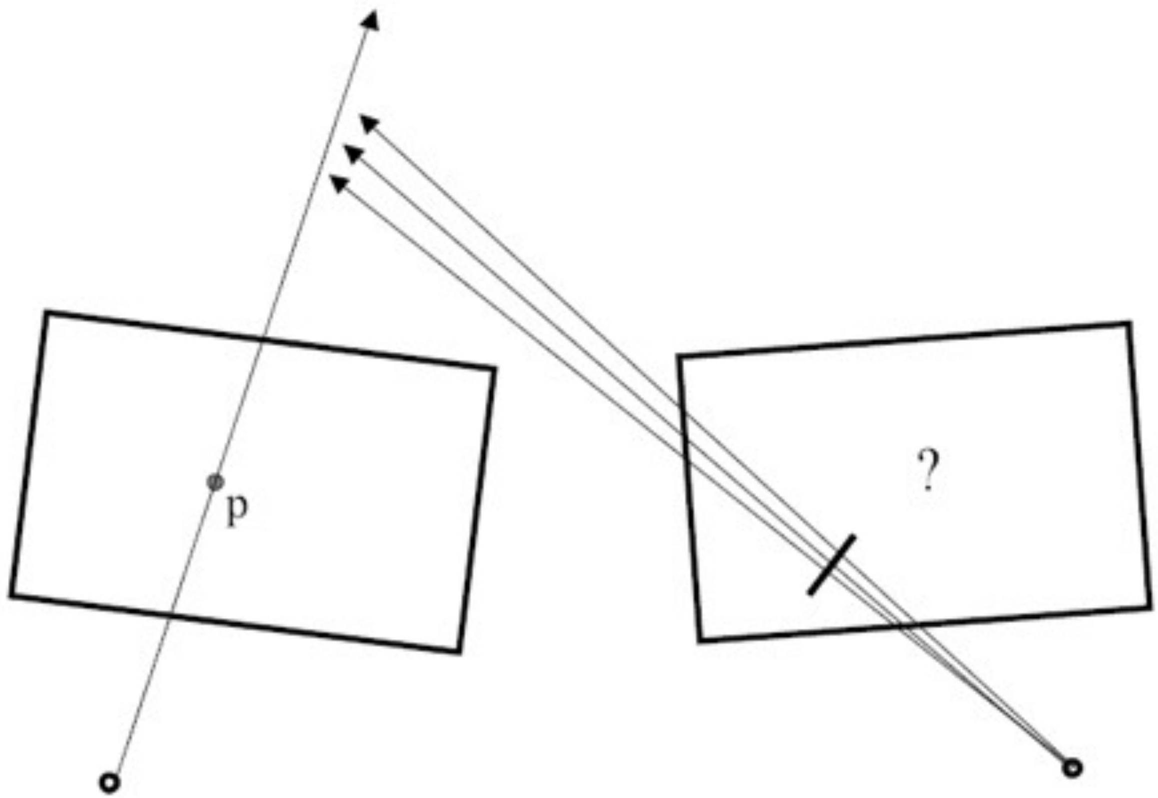
Stereo correspondence constraints



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

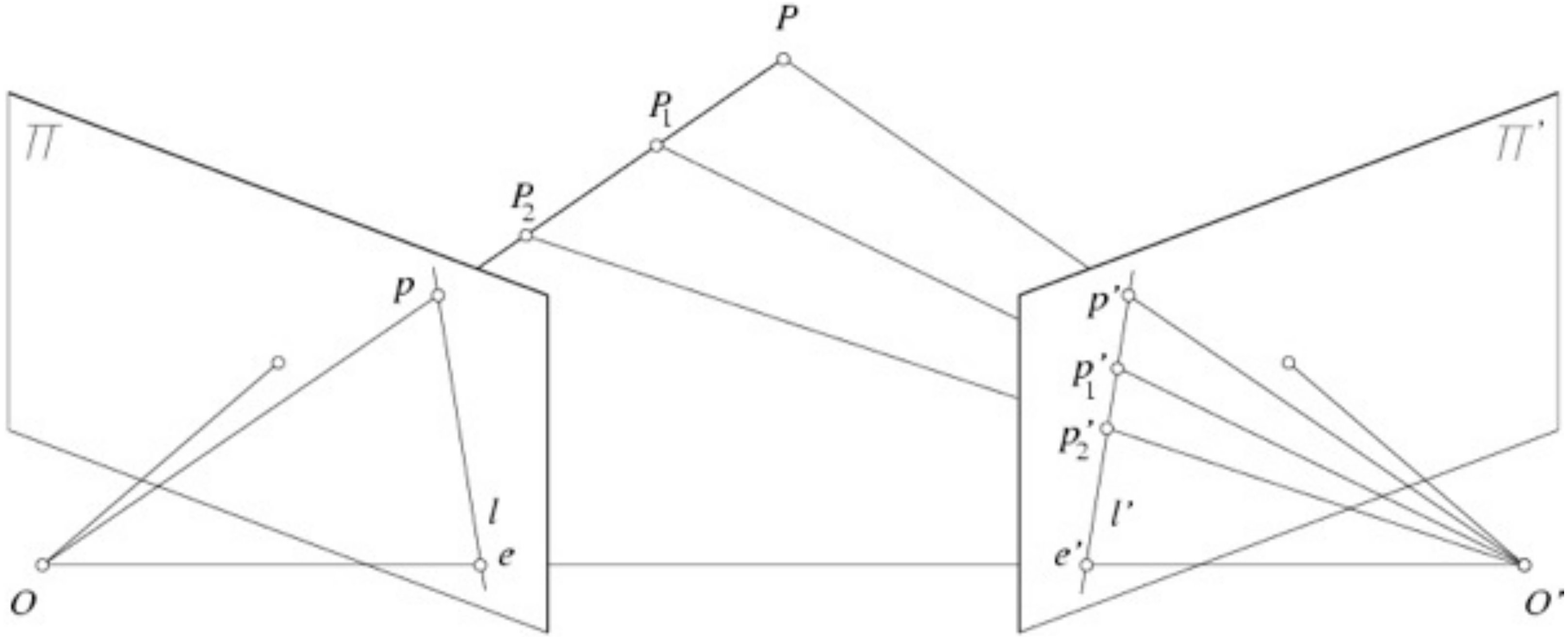
Slide credit: Kristen Grauman

Stereo correspondence constraints



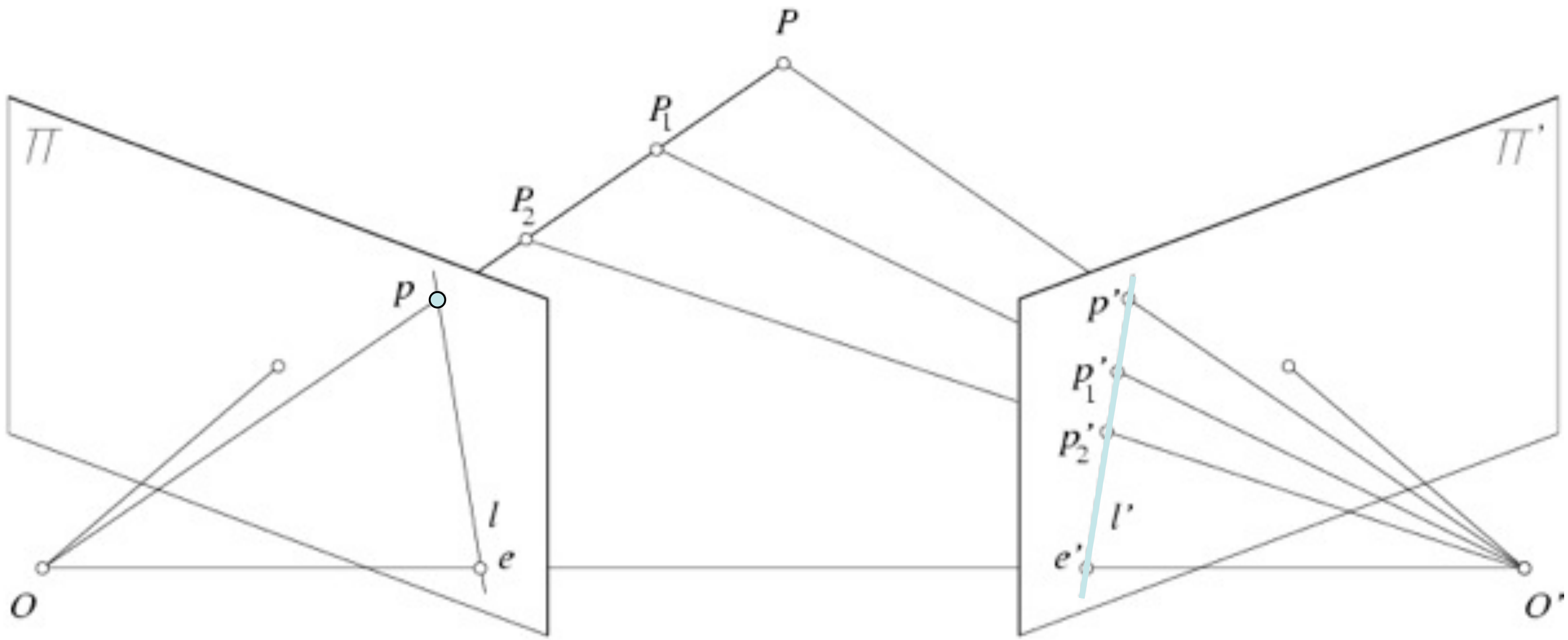
Slide credit: Kristen Grauman

Epipolar constraint



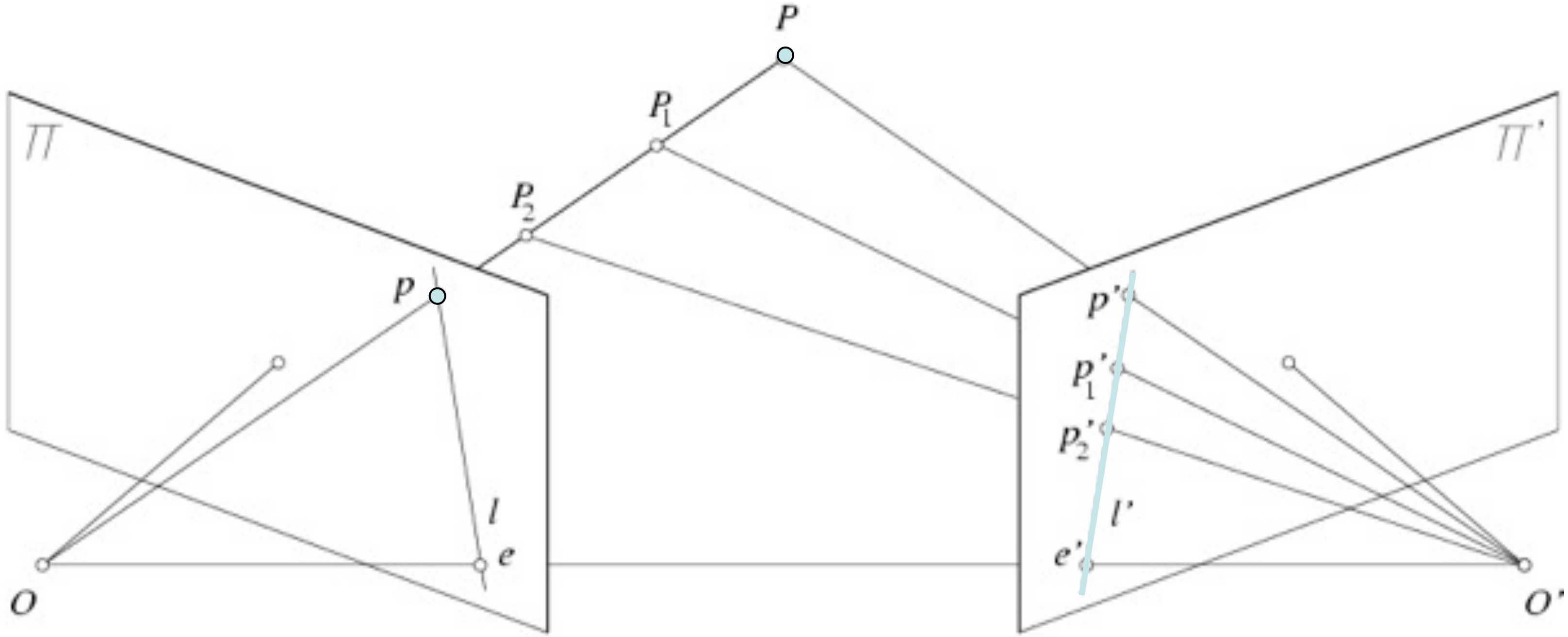
Slide credit: Kristen Grauman

Epipolar constraint



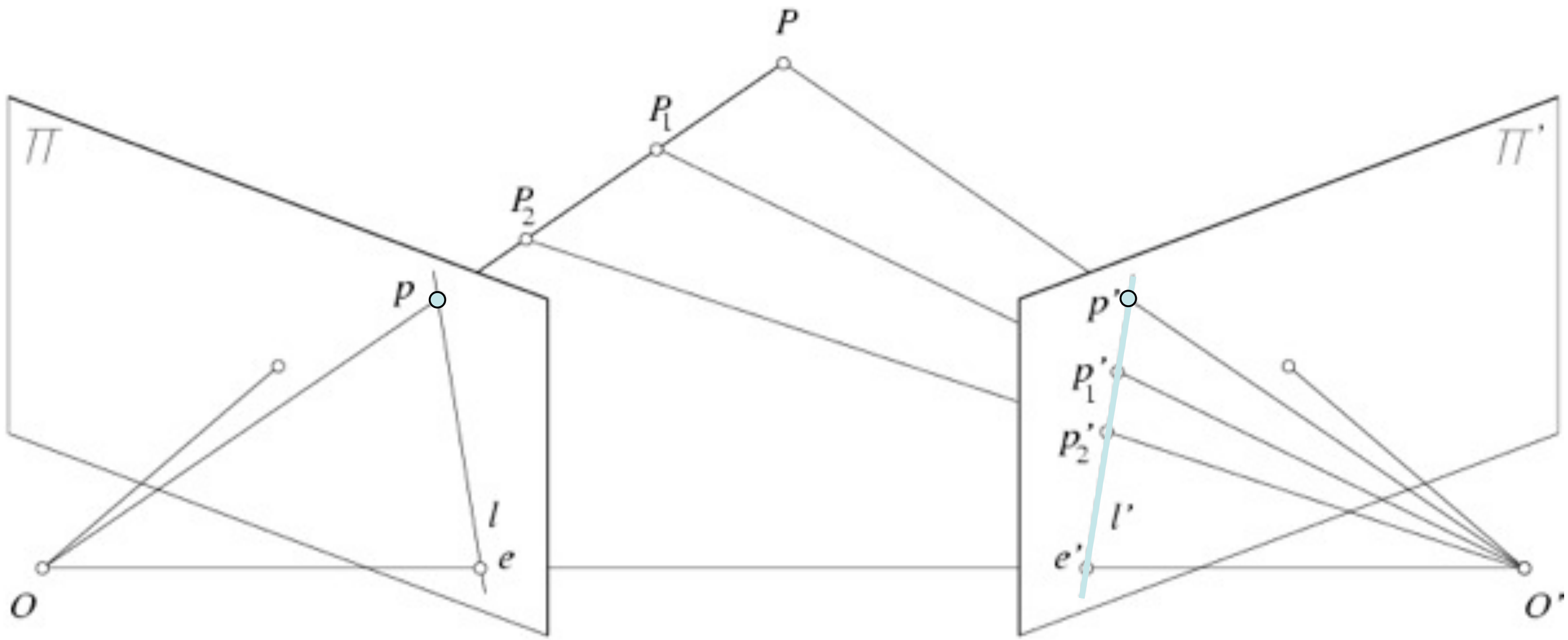
Slide credit: Kristen Grauman

Epipolar constraint



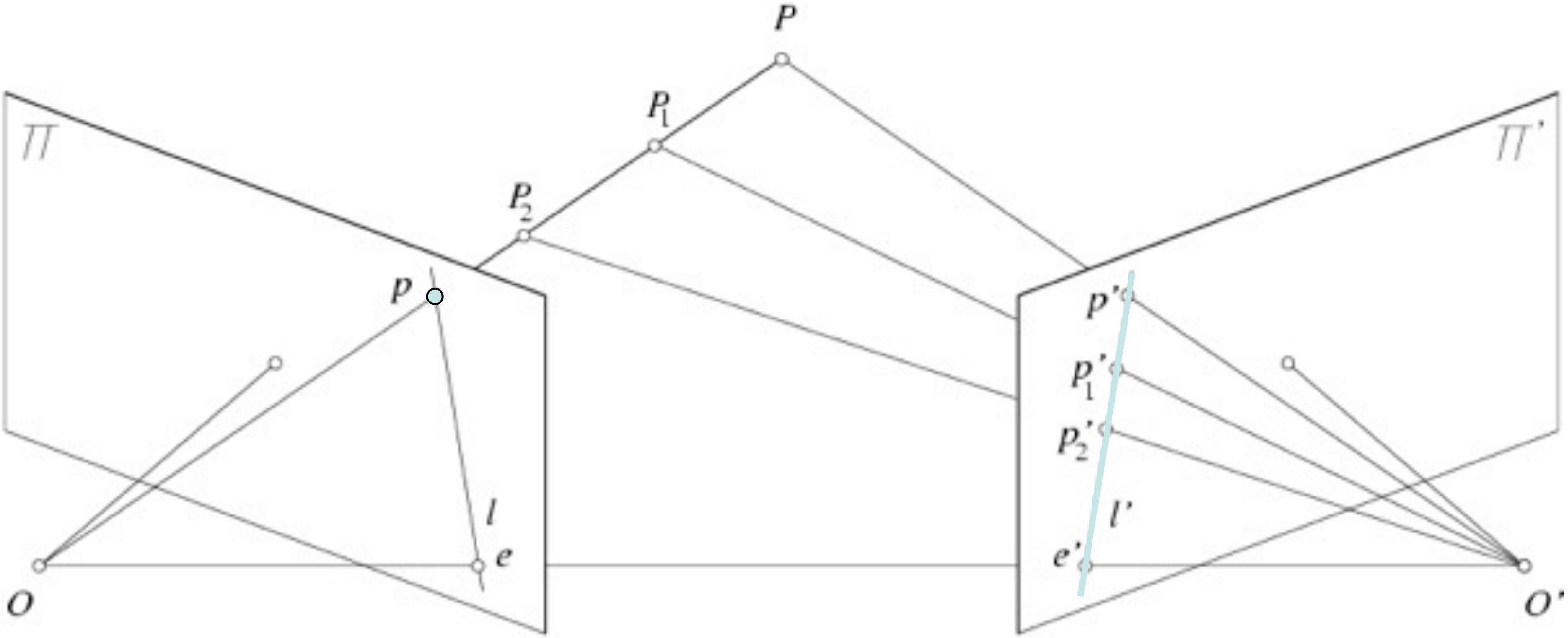
Slide credit: Kristen Grauman

Epipolar constraint



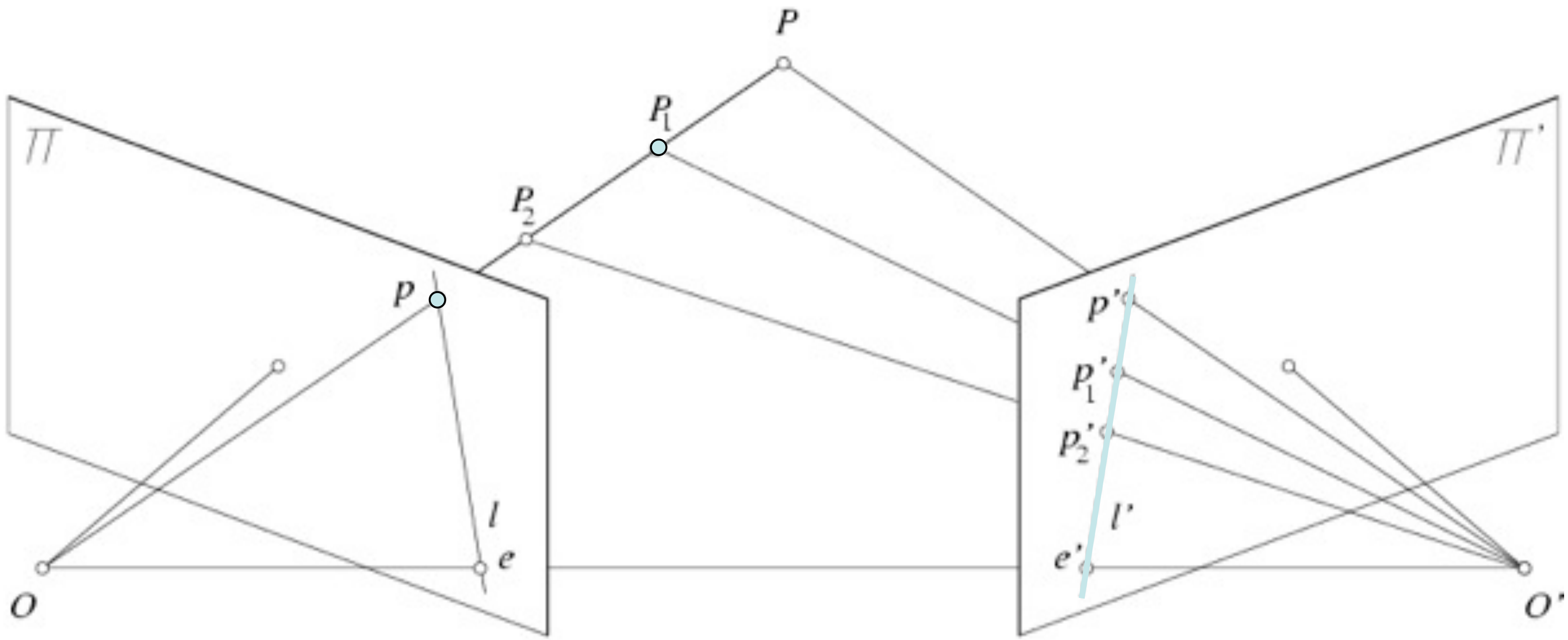
Slide credit: Kristen Grauman

Epipolar constraint



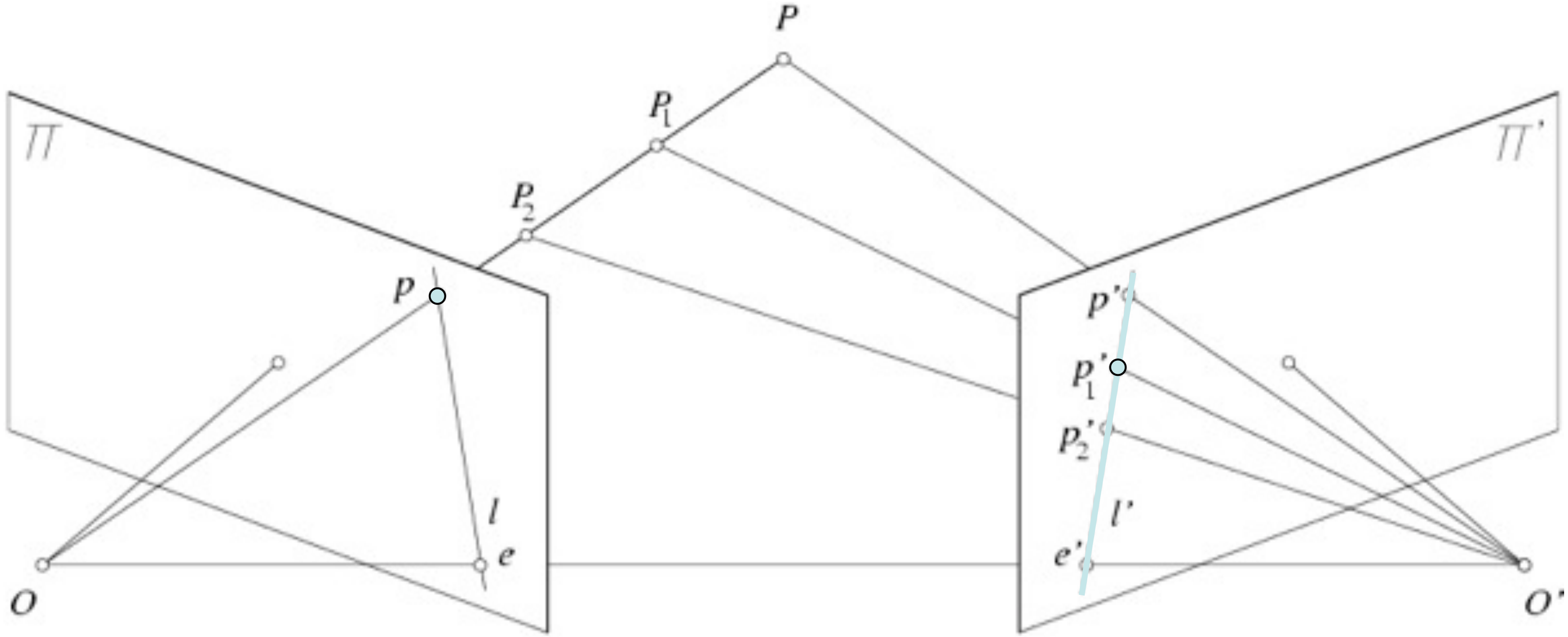
Slide credit: Kristen Grauman

Epipolar constraint



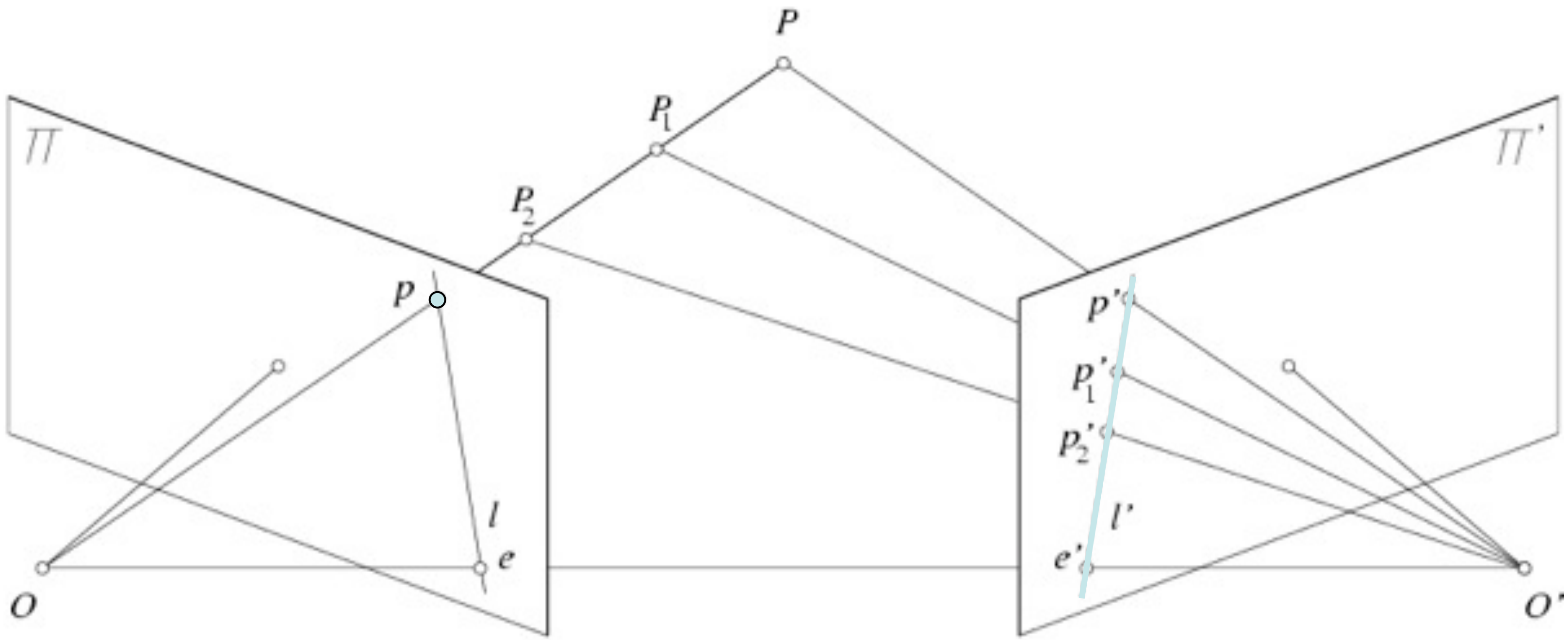
Slide credit: Kristen Grauman

Epipolar constraint



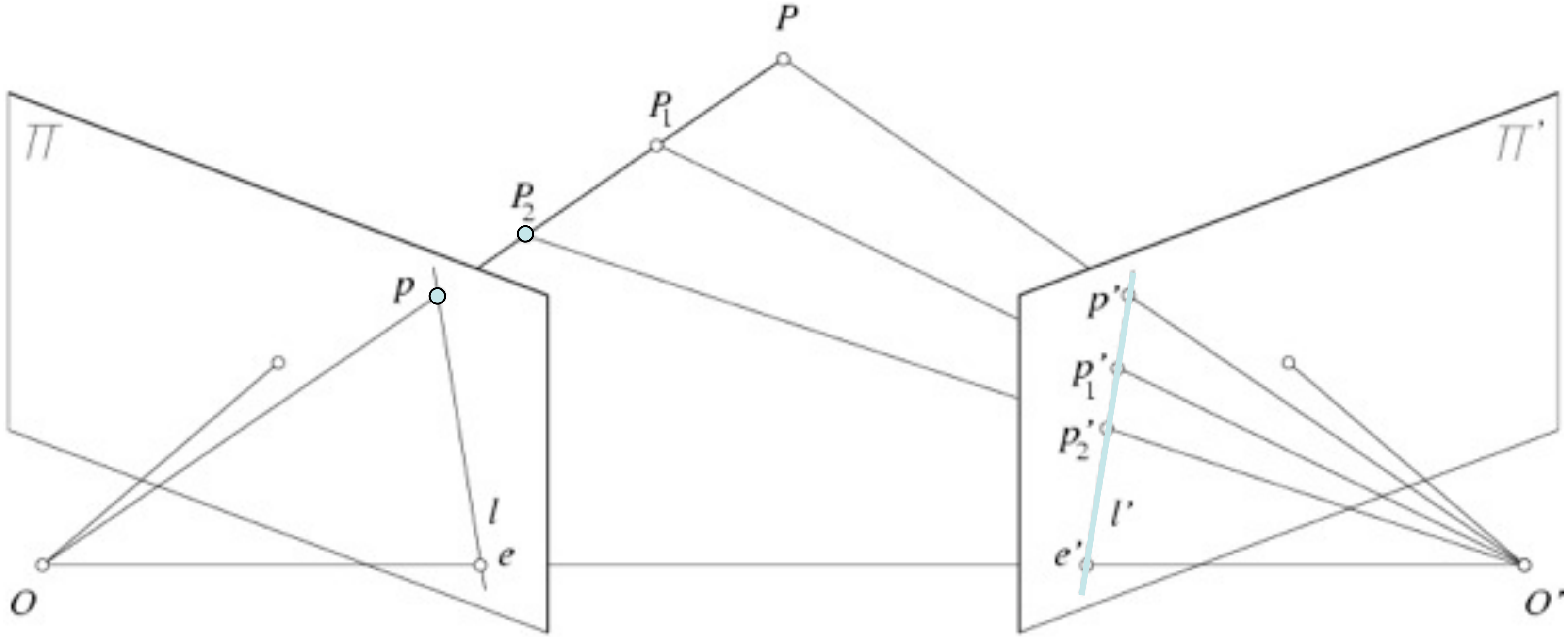
Slide credit: Kristen Grauman

Epipolar constraint



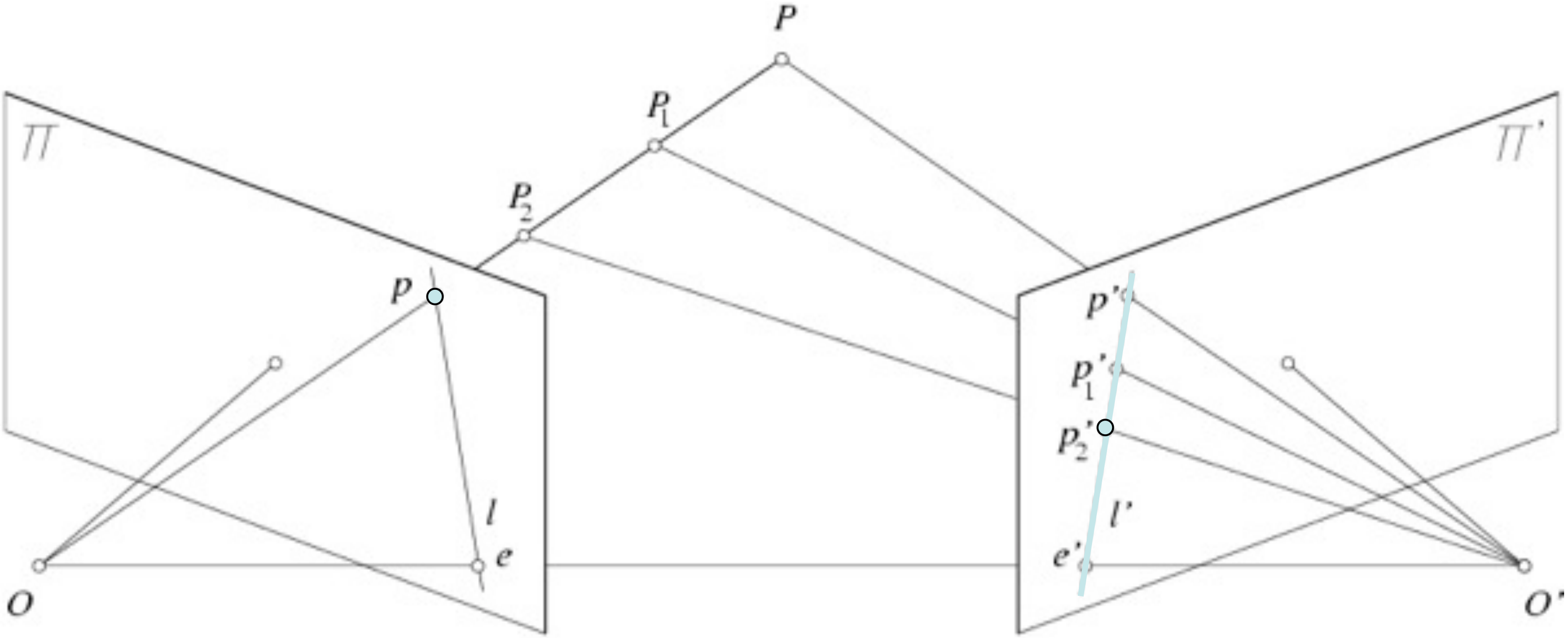
Slide credit: Kristen Grauman

Epipolar constraint



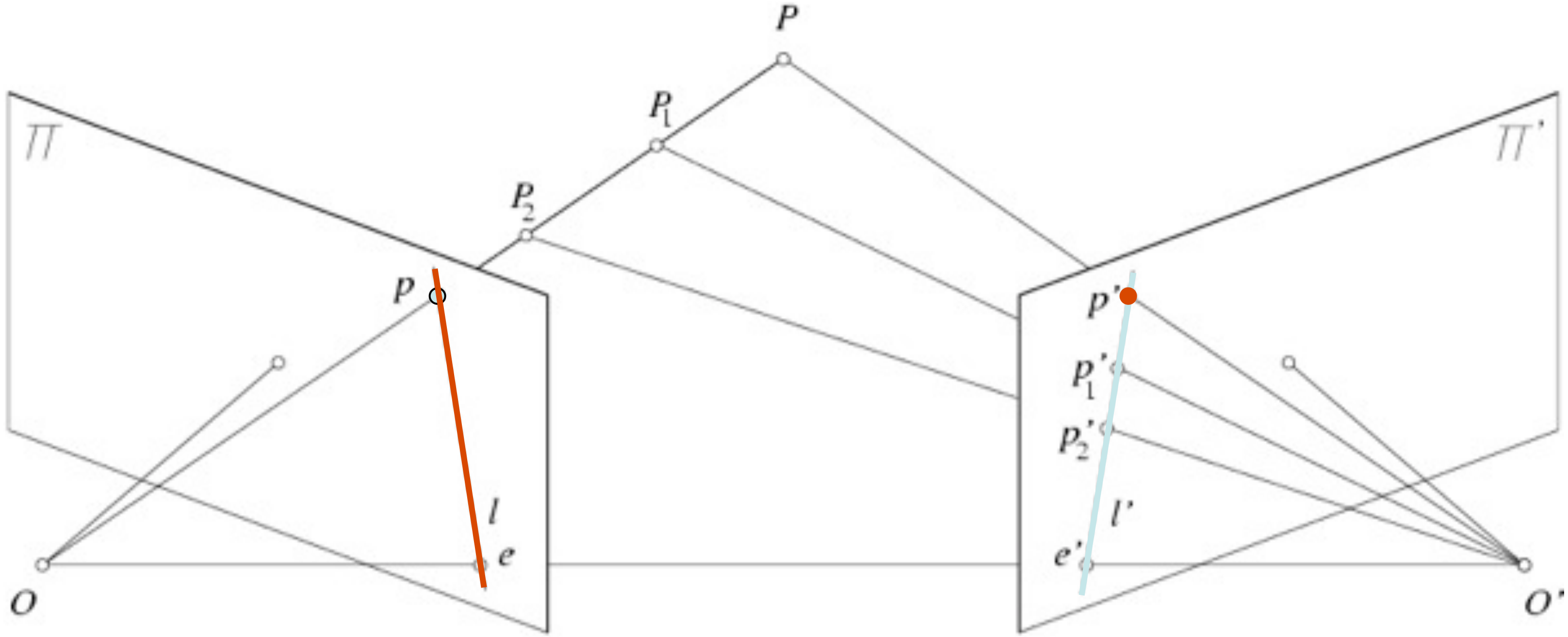
Slide credit: Kristen Grauman

Epipolar constraint



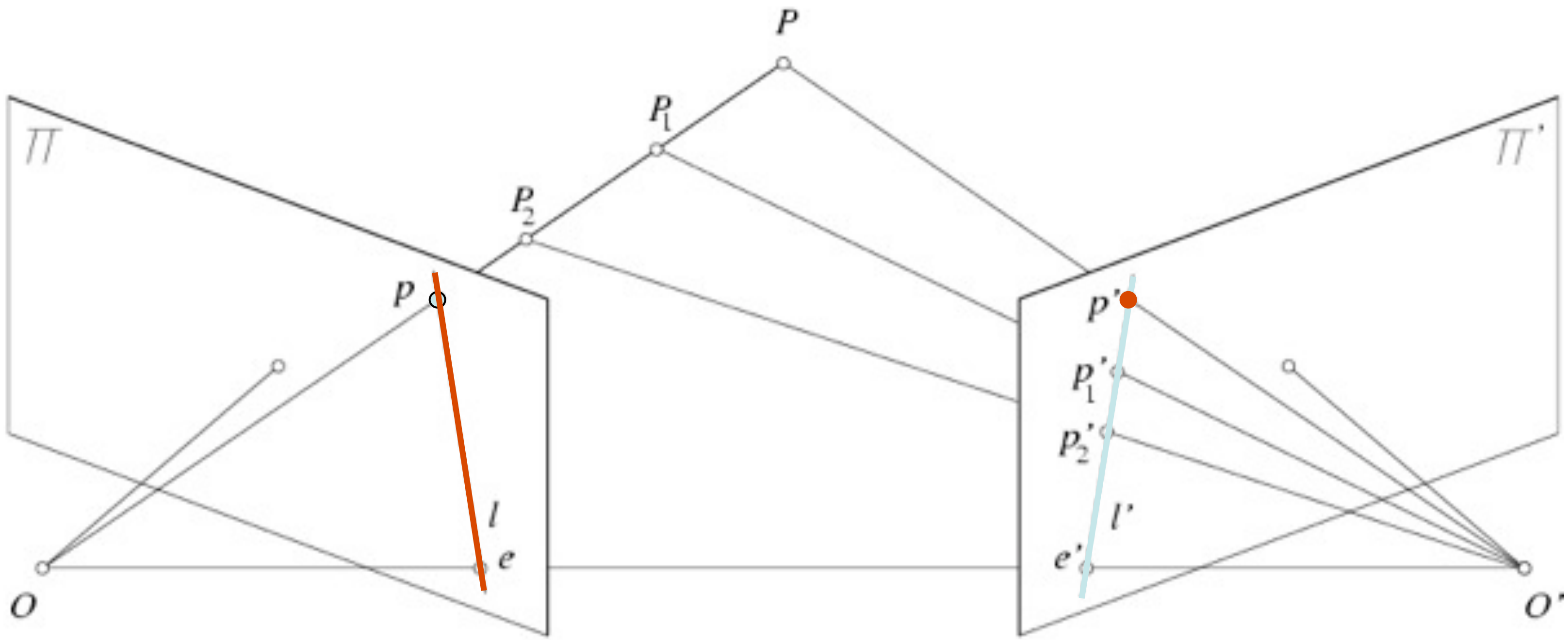
Slide credit: Kristen Grauman

Epipolar constraint



Slide credit: Kristen Grauman

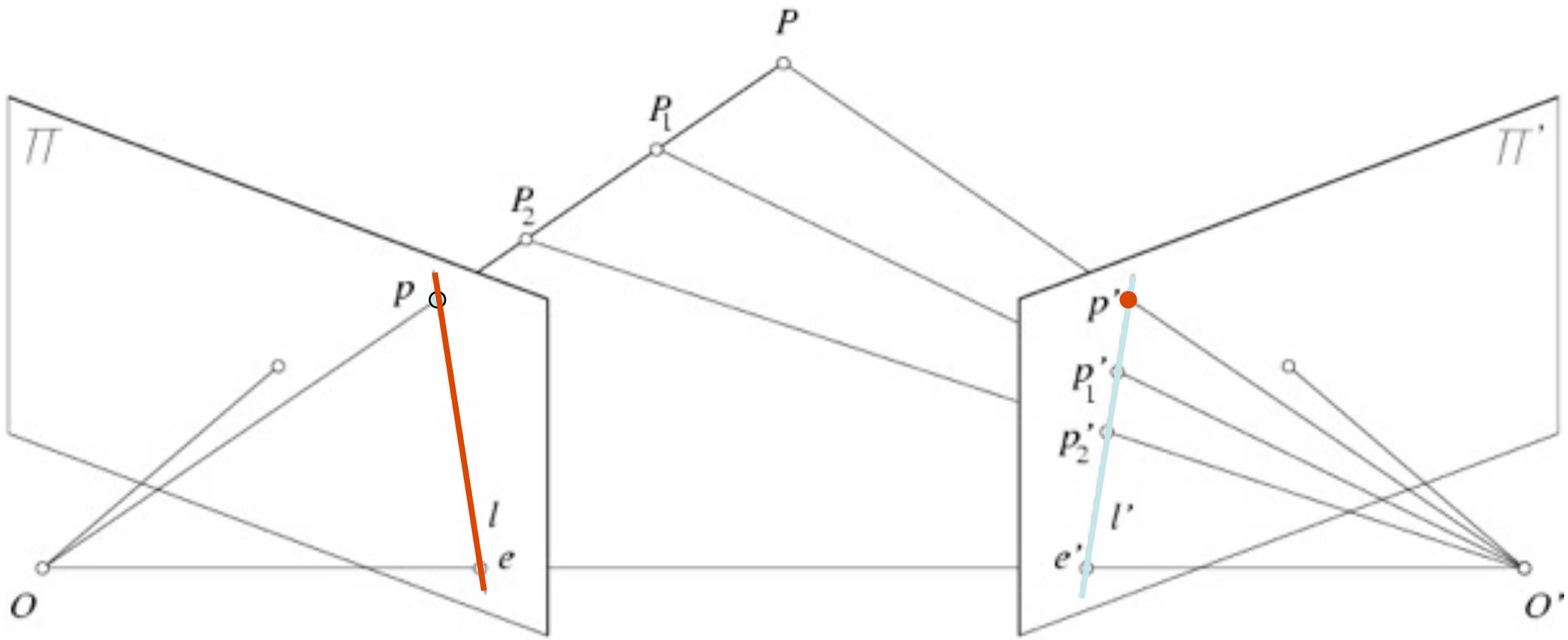
Epipolar constraint



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

Slide credit: Kristen Grauman

Epipolar constraint

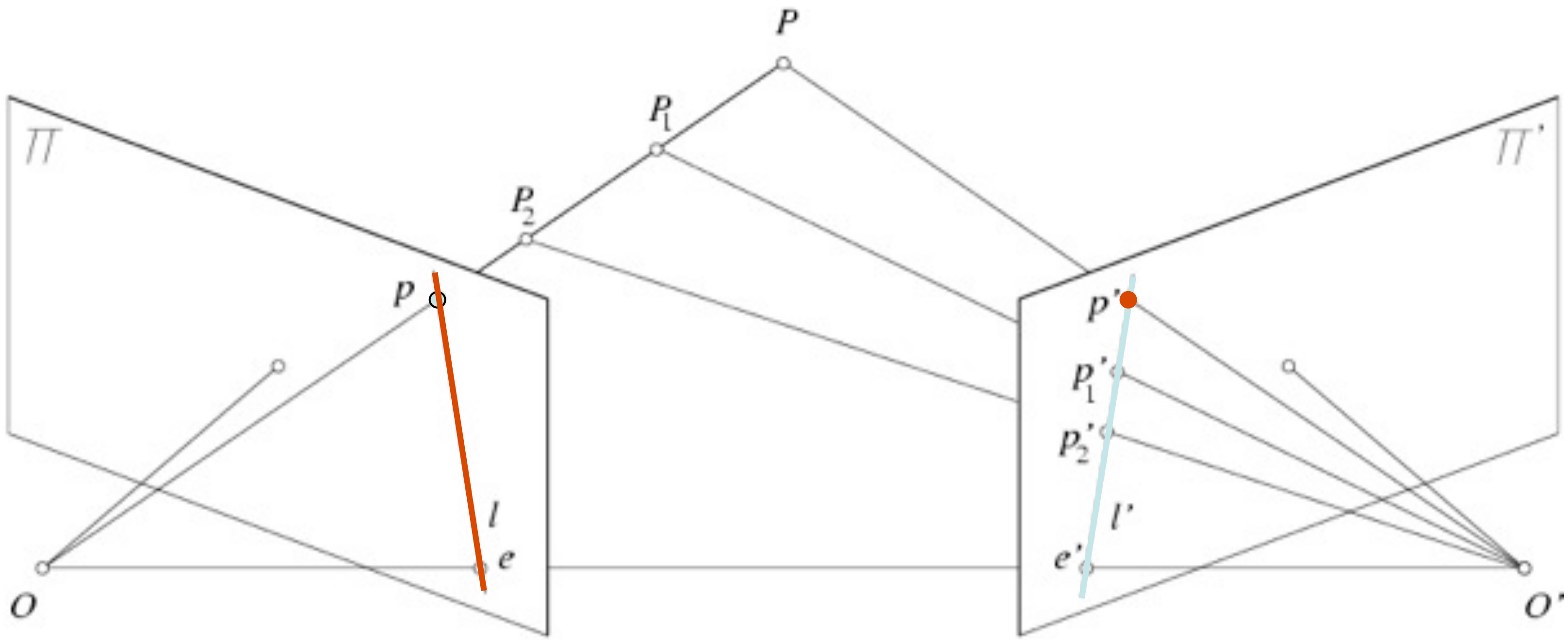


Geometry of two views constrains where the corresponding pixel for some image point in the first view must occur in the second view:

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

Slide credit: Kristen Grauman

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.**

Why is this useful?

Slide credit: Kristen Grauman

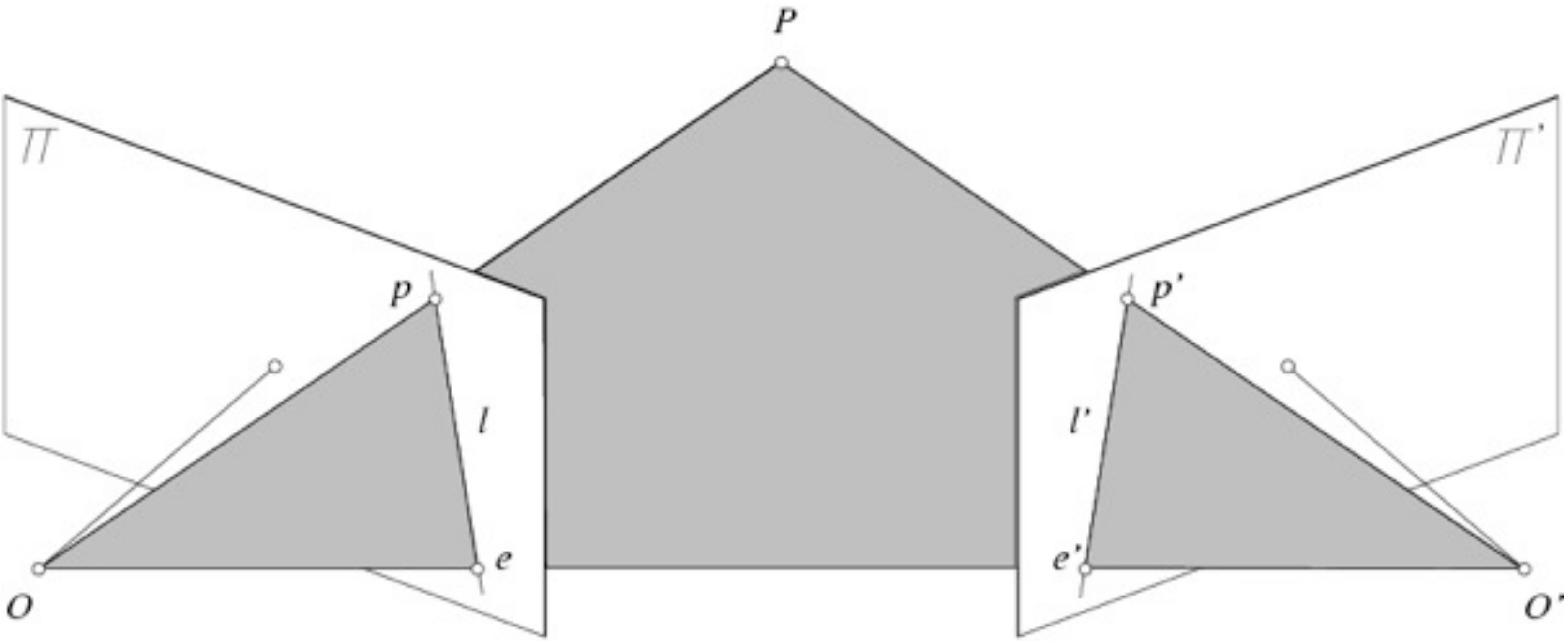
Epipolar constraint

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

Image from Andrew Zisserman

Slide credit: Kristen Grauman

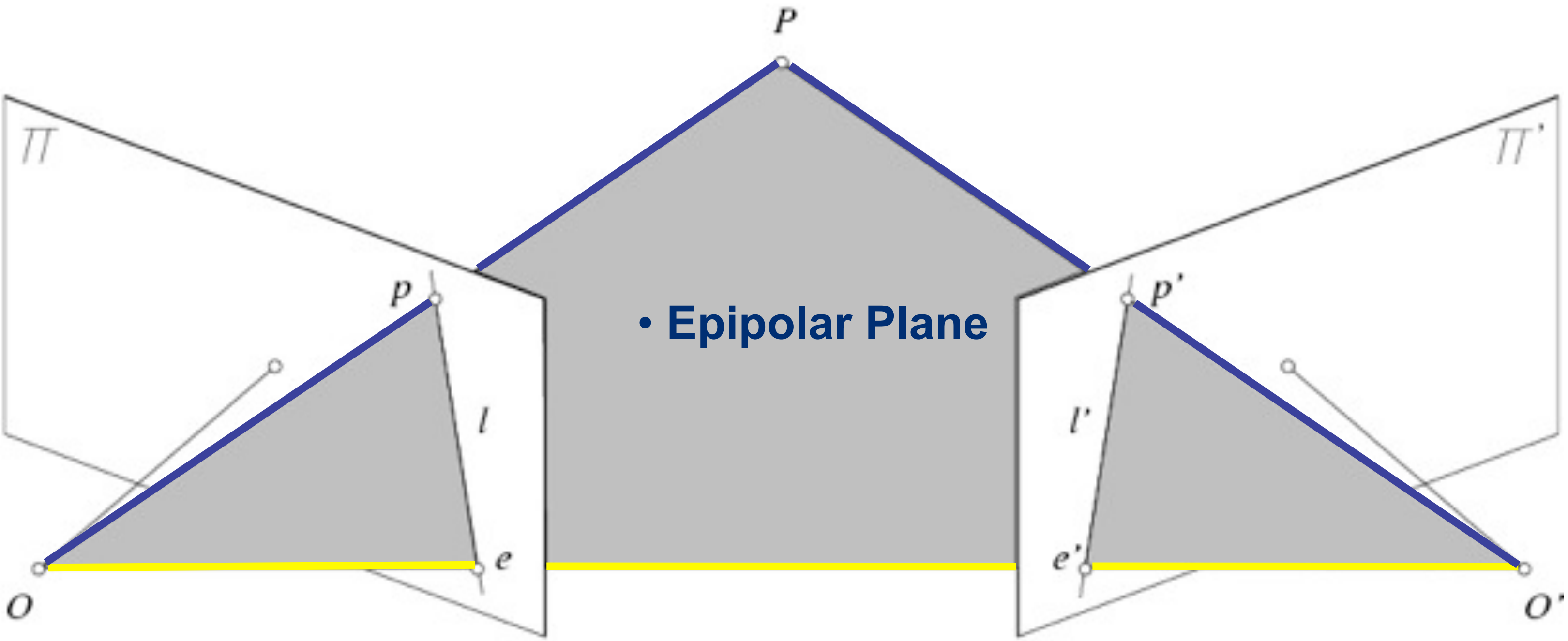
Epipolar geometry



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

Slide credit: Kristen Grauman

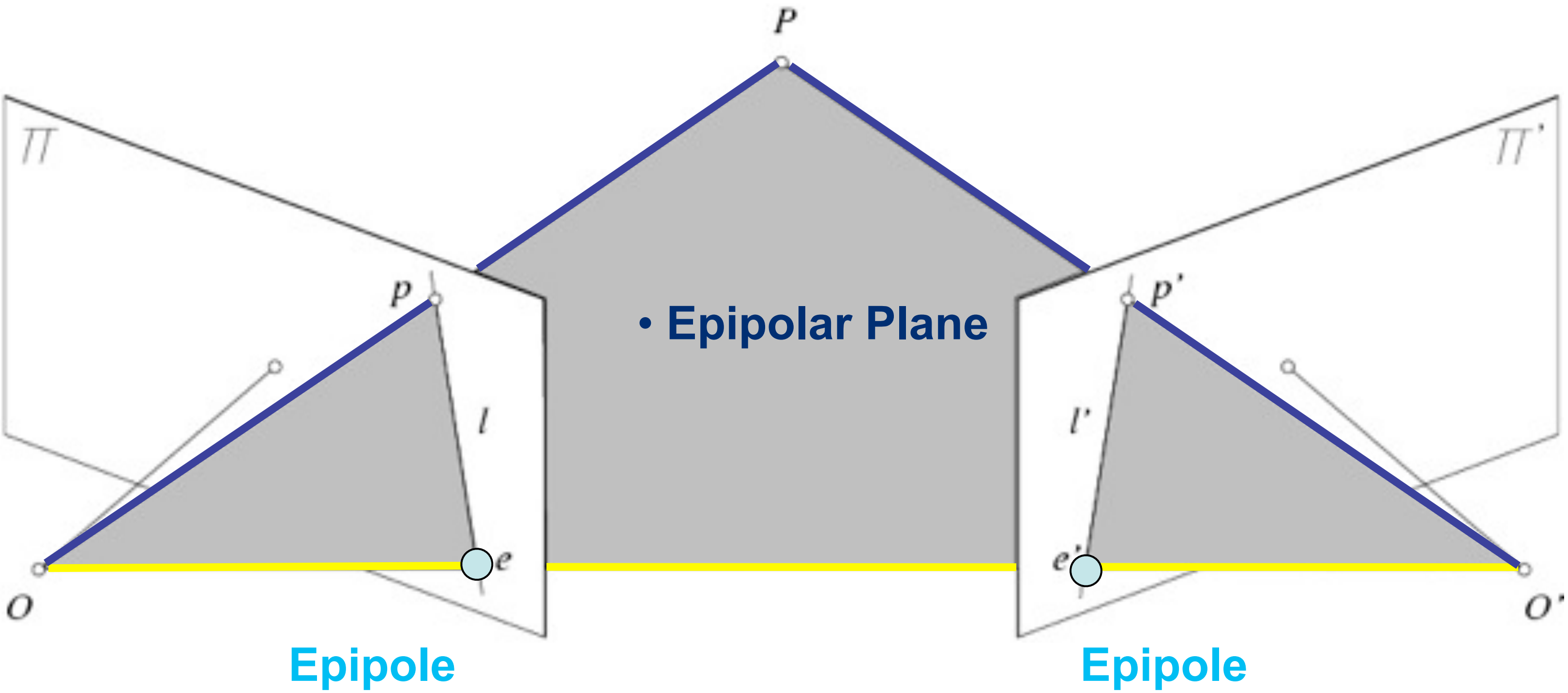
Epipolar geometry



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

Slide credit: Kristen Grauman

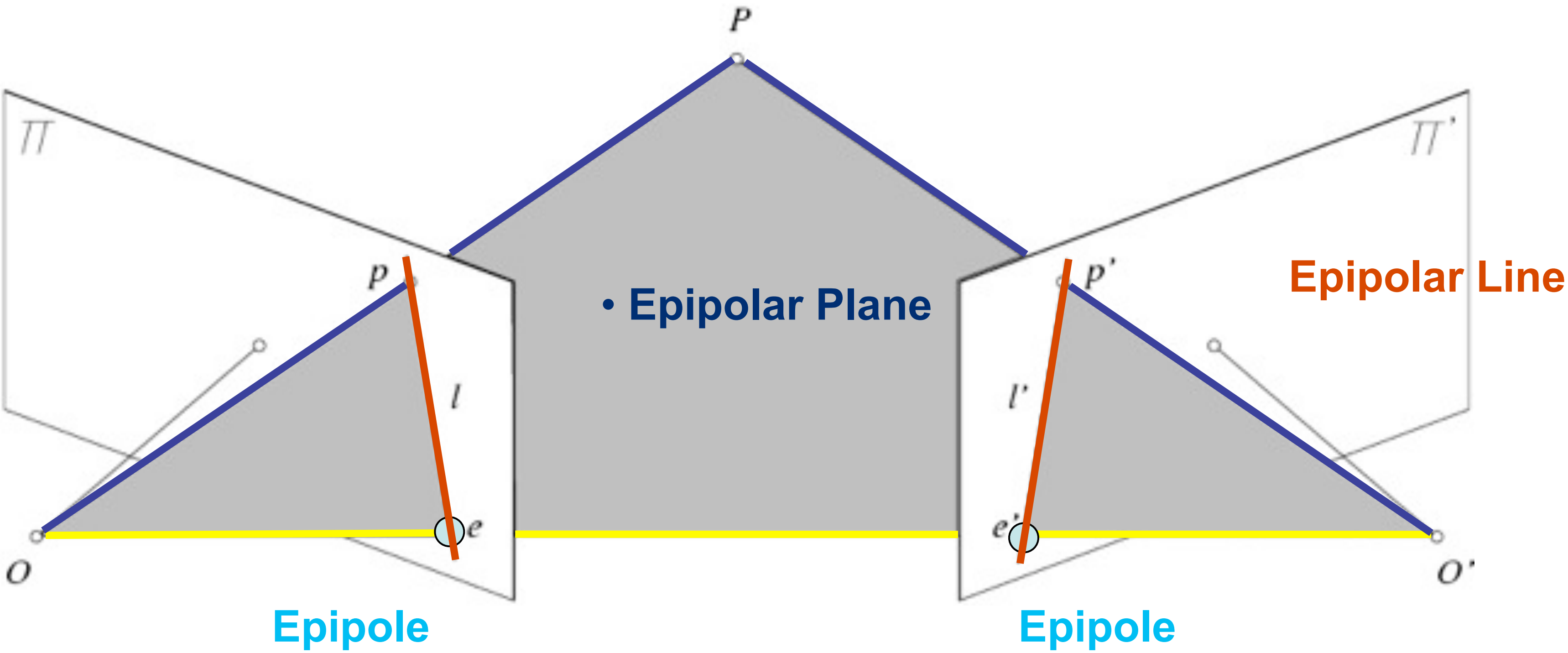
Epipolar geometry



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

Slide credit: Kristen Grauman

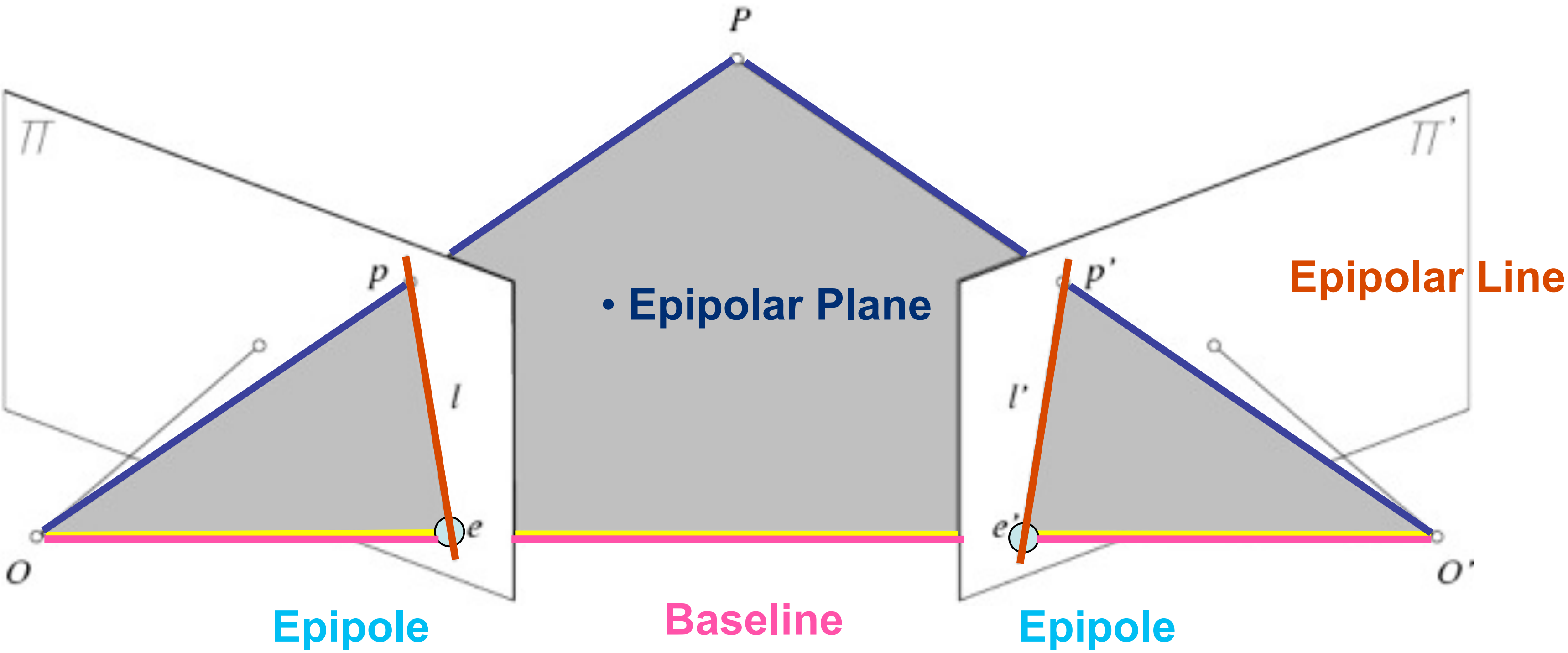
Epipolar geometry



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

Slide credit: Kristen Grauman

Epipolar geometry



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

Slide credit: Kristen Grauman

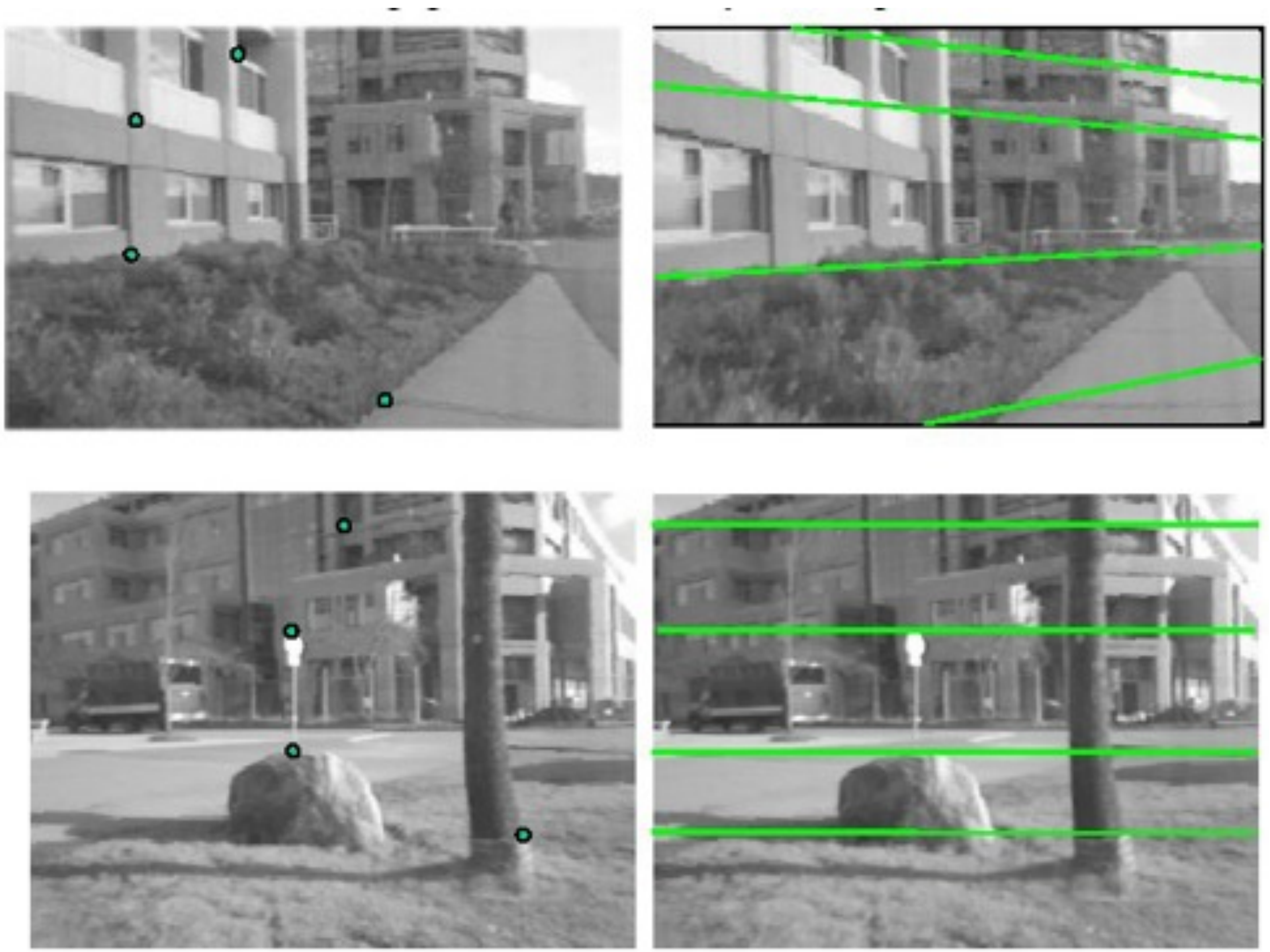
Epipolar geometry: terms

- **Baseline**: line joining the camera centers
- **Epipole**: point of intersection of baseline with the 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

Slide credit: Kristen Grauman

Example



Slide credit: Kristen Grauman

Example: converging cameras

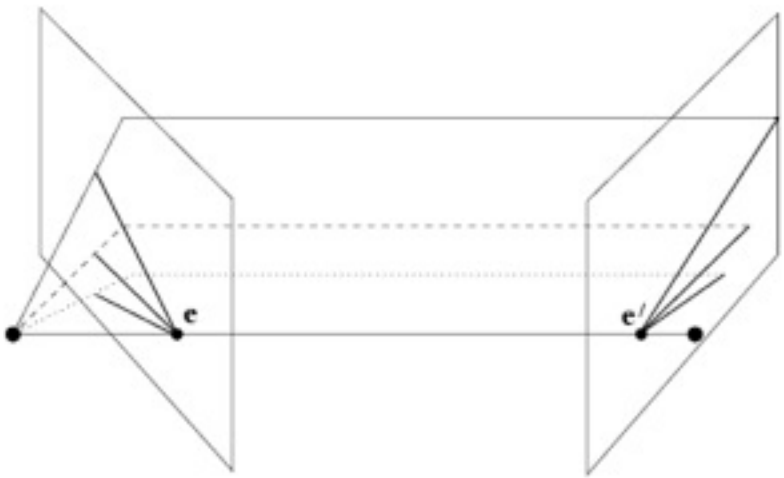
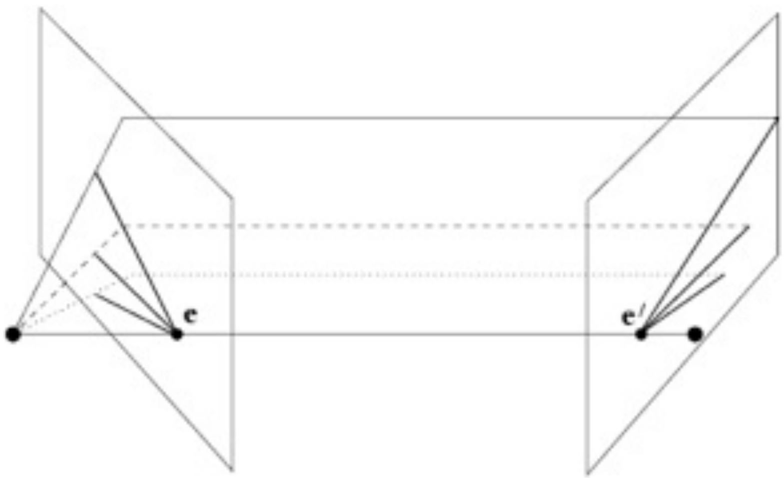


Figure from Hartley & Zisserman

Slide credit: Kristen Grauman

Example: converging cameras



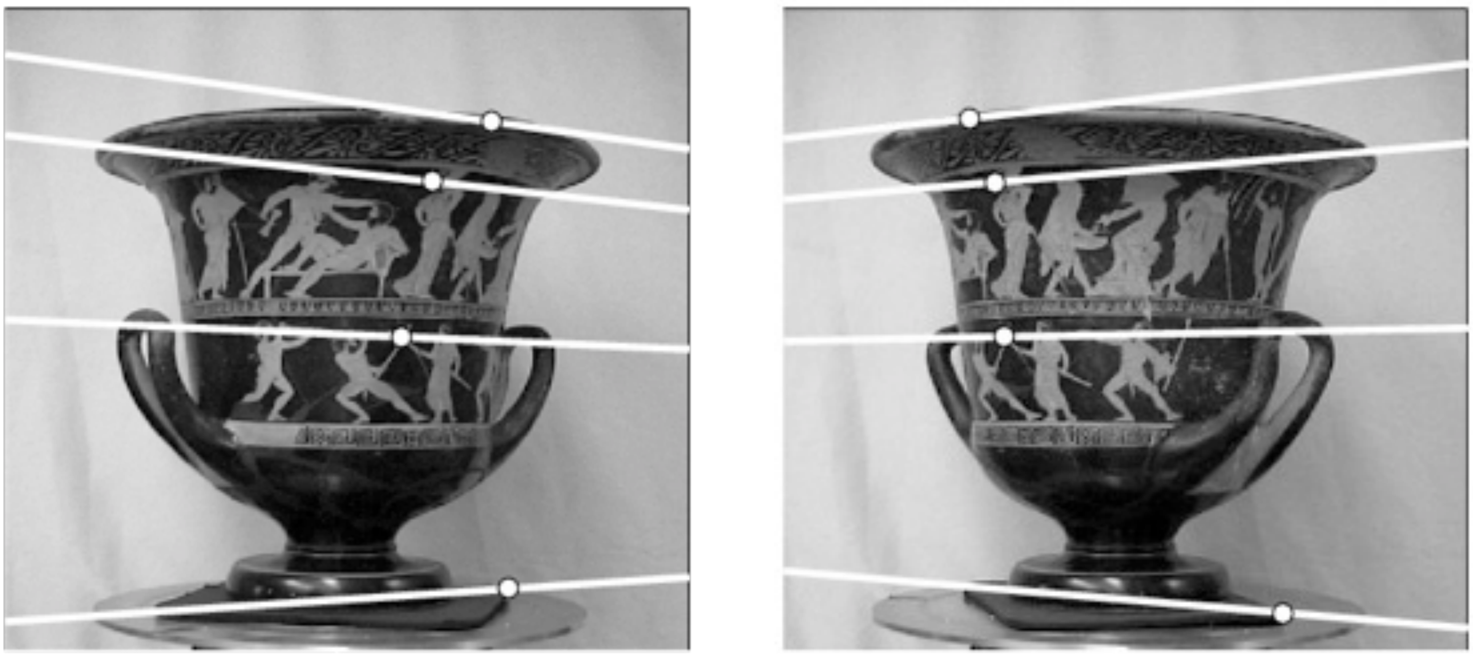
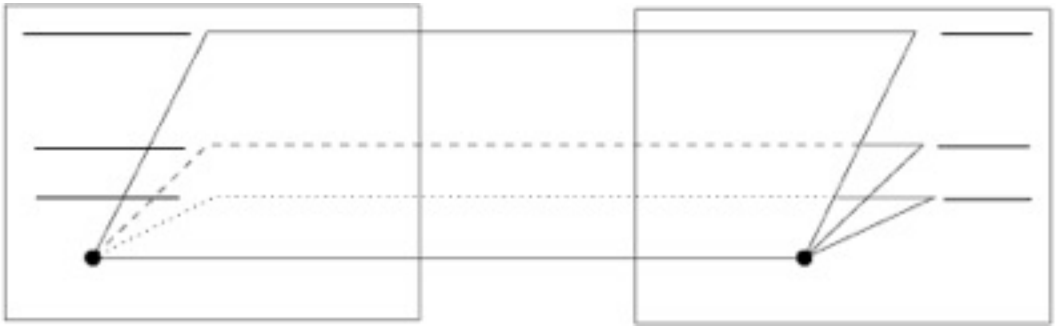


Figure from Hartley & Zisserman

Slide credit: Kristen Grauman

Example: parallel cameras



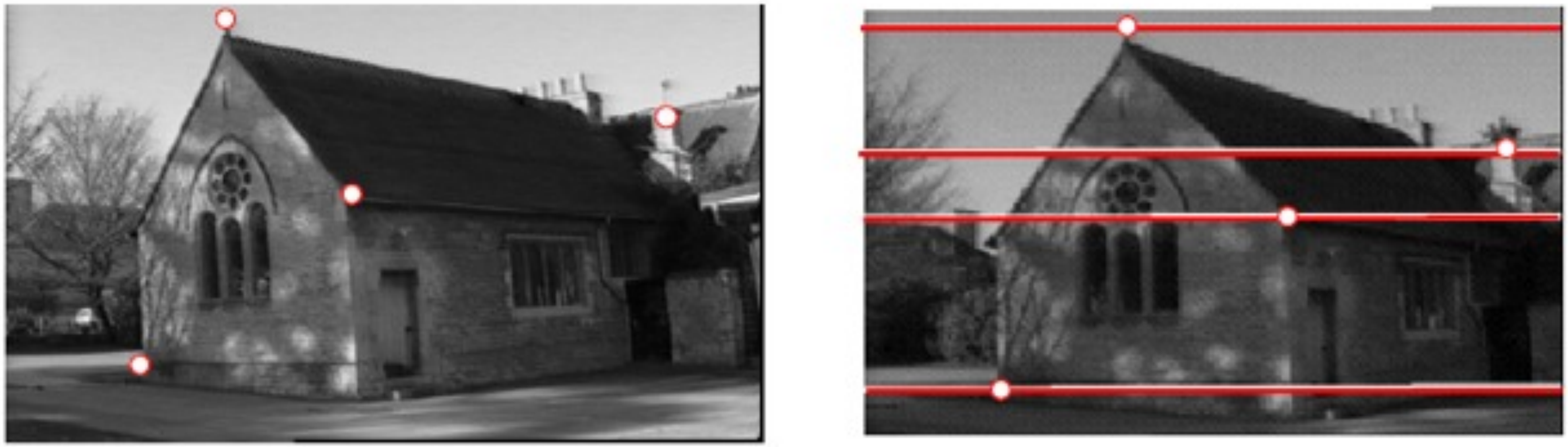
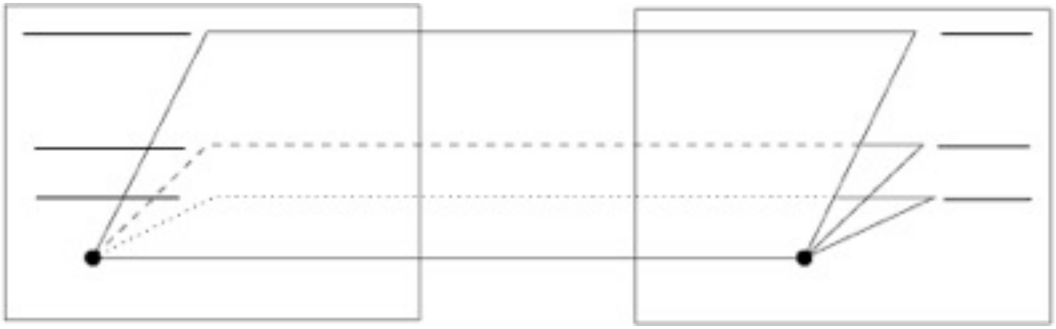


Figure from Hartley & Zisserman

Slide credit: Kristen Grauman

Example: parallel cameras



Where are the epipoles?

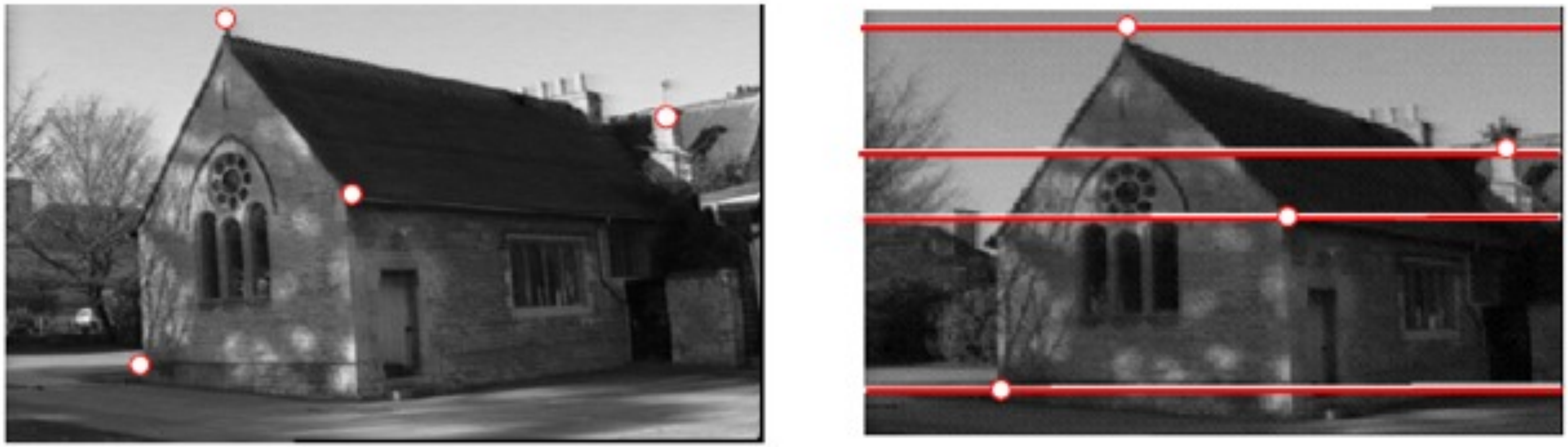


Figure from Hartley & Zisserman

Slide credit: Kristen Grauman

Example: parallel cameras

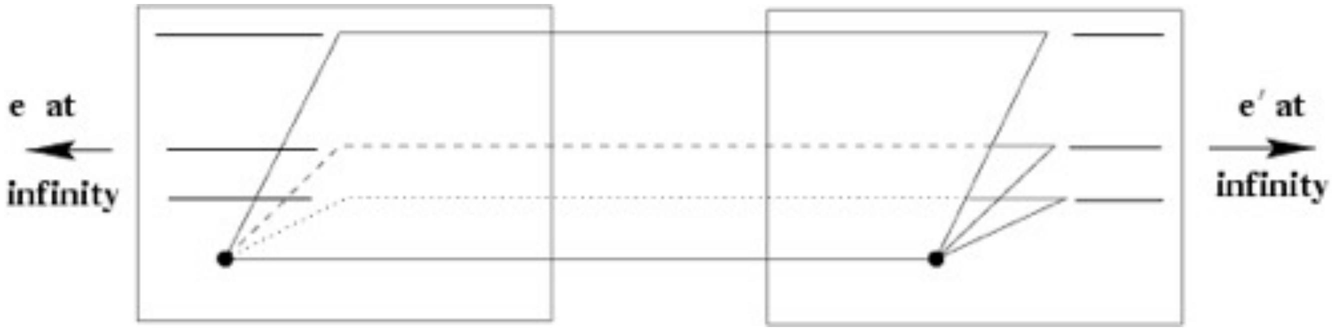


Figure from Hartley & Zisserman

Slide credit: Kristen Grauman

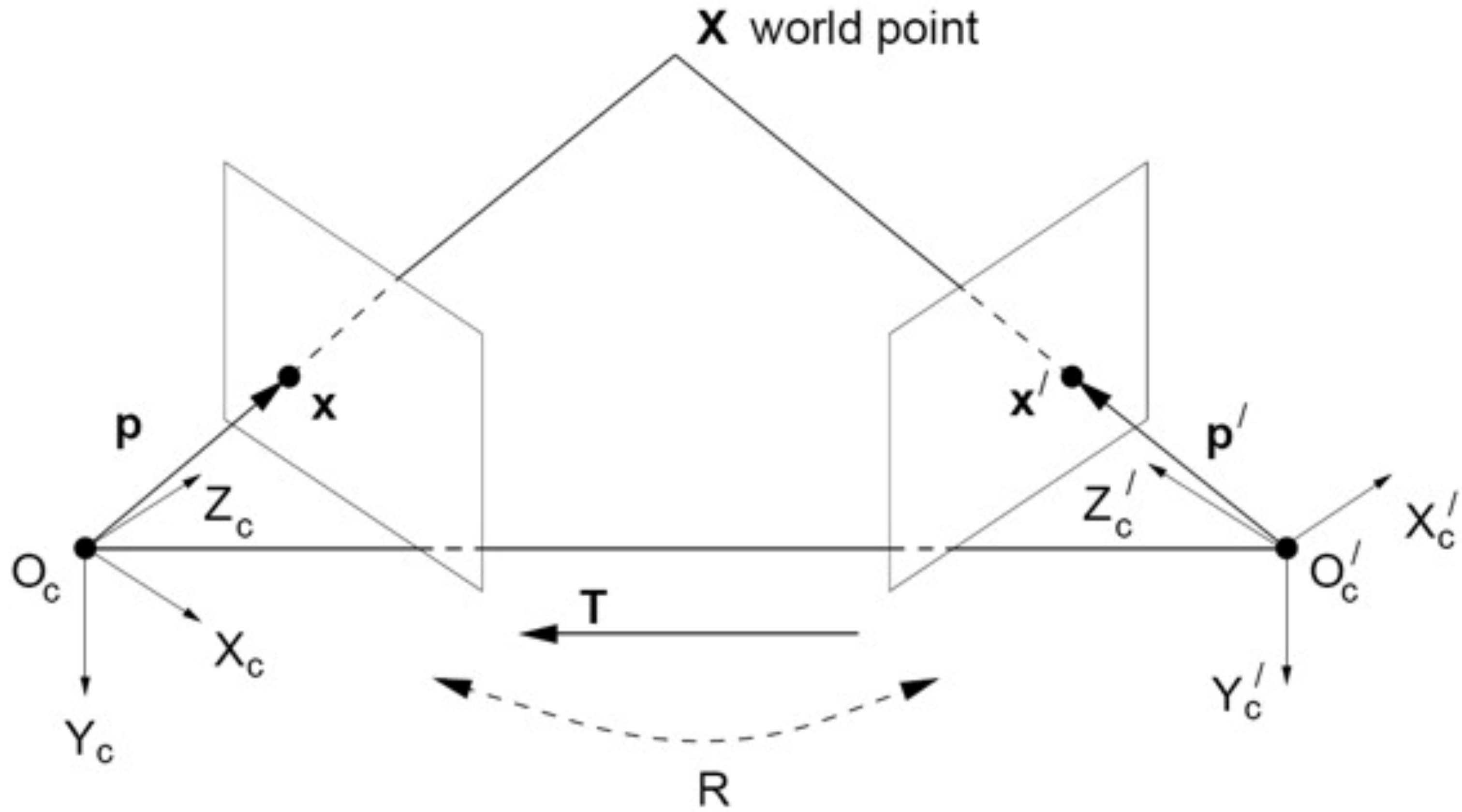
Stereo Topics

- Special, simple system, main idea
- **More general camera conditions, epipolar constraints**
  - epipolar geometry
  - **epipolar algebra**
- Image rectification
- Stereo matching (likelihood term)
- Stereo regularization (prior term)
- Inference
  - dynamic programming
  - graph cuts
- Structured light

- So far, we have the explanation in terms of geometry.
- Now, how to express the epipolar constraints algebraically?

Slide credit: Kristen Grauman

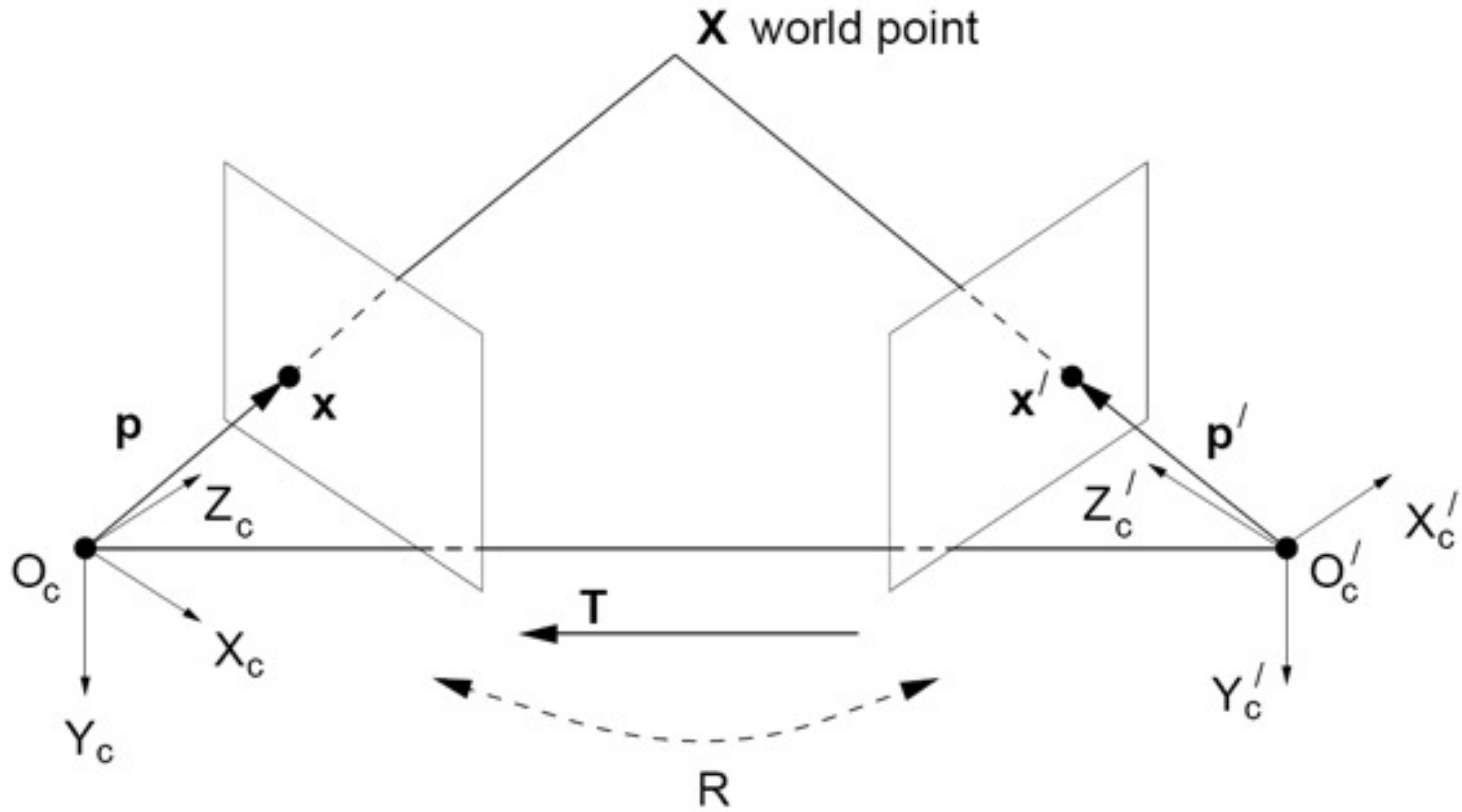
Stereo geometry, with calibrated cameras



Main idea

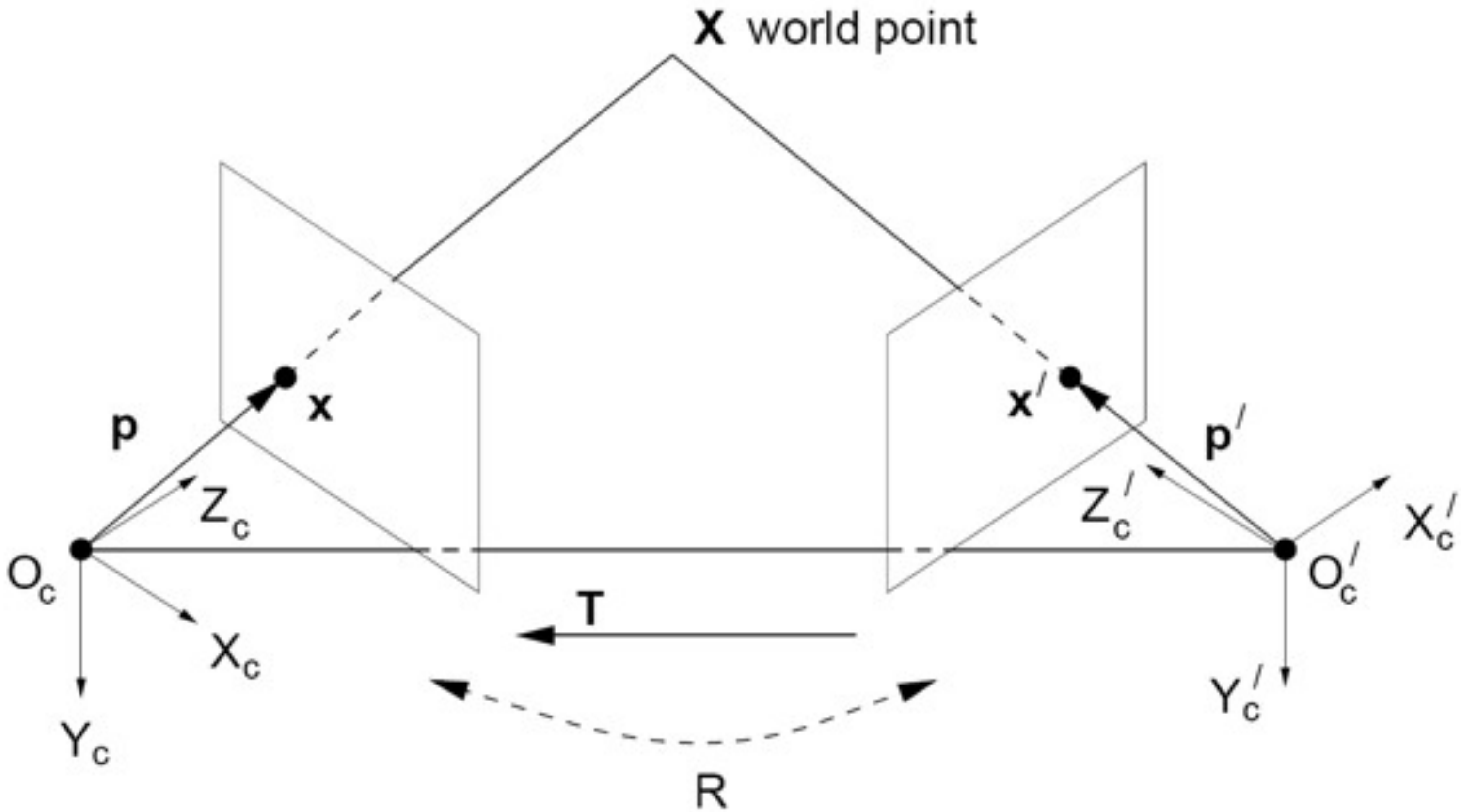
Slide credit: Kristen Grauman

Stereo geometry, with calibrated cameras



Slide credit: Kristen Grauman

Stereo geometry, with calibrated cameras

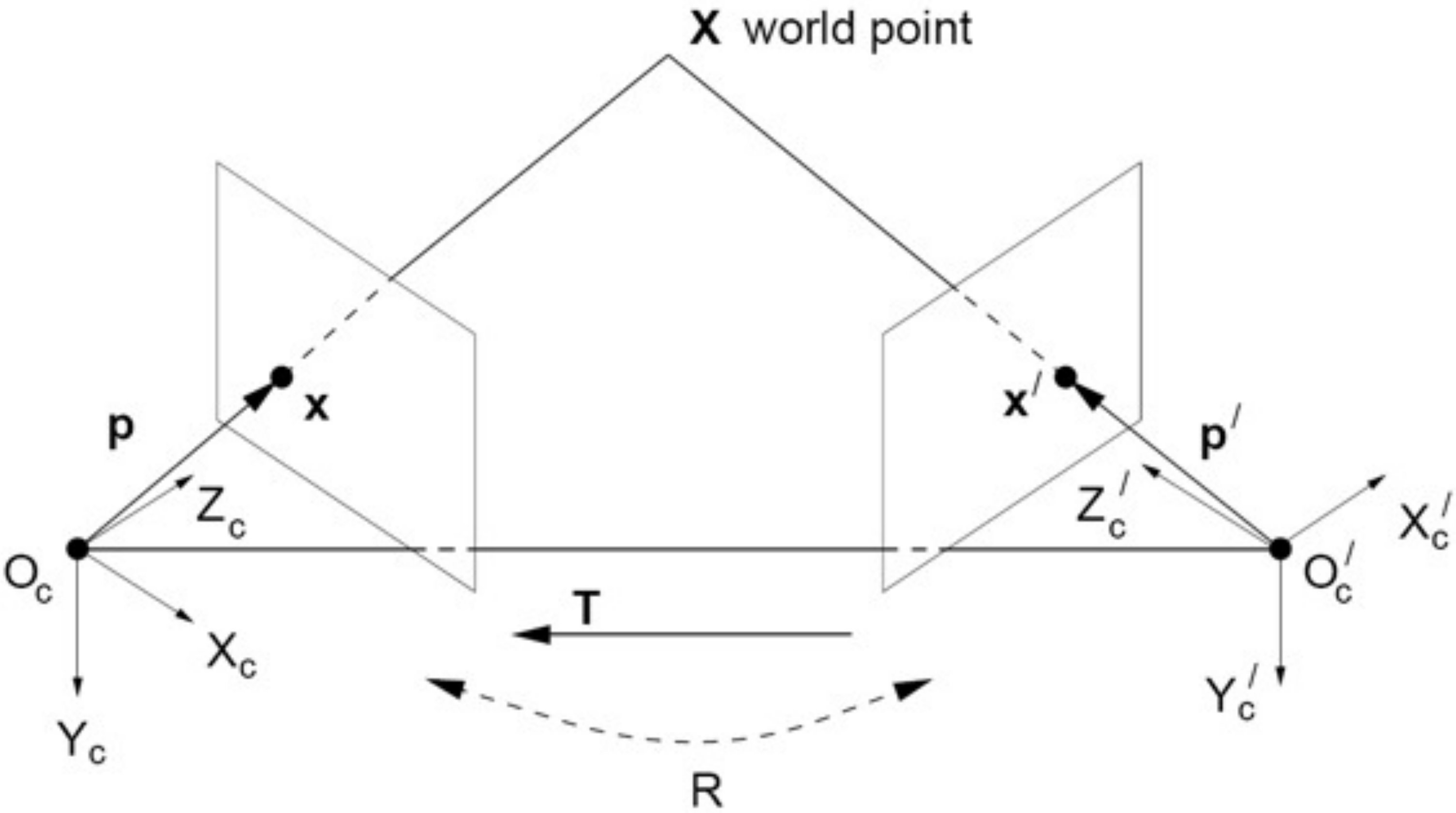


If the stereo rig is calibrated, we know :

how to rotate and translate camera reference frame 1 to get to camera reference frame 2.

Slide credit: Kristen Grauman

Stereo geometry, with calibrated cameras



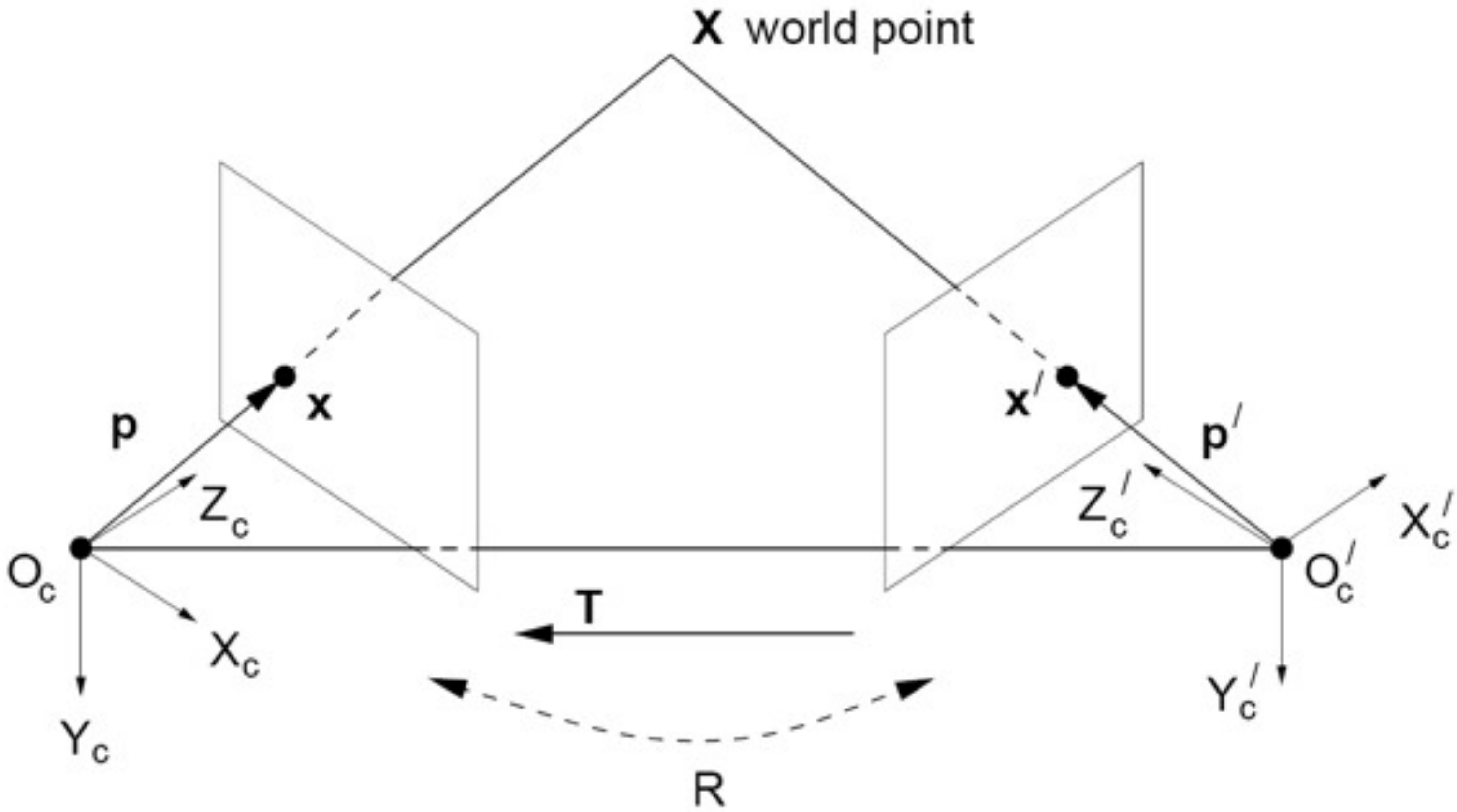
If the stereo rig is calibrated, we know :

how to rotate and translate camera reference frame 1 to get to camera reference frame 2.

Rotation: 3 x 3 matrix R; translation: 3 vector T.

Slide credit: Kristen Grauman

Stereo geometry, with calibrated cameras



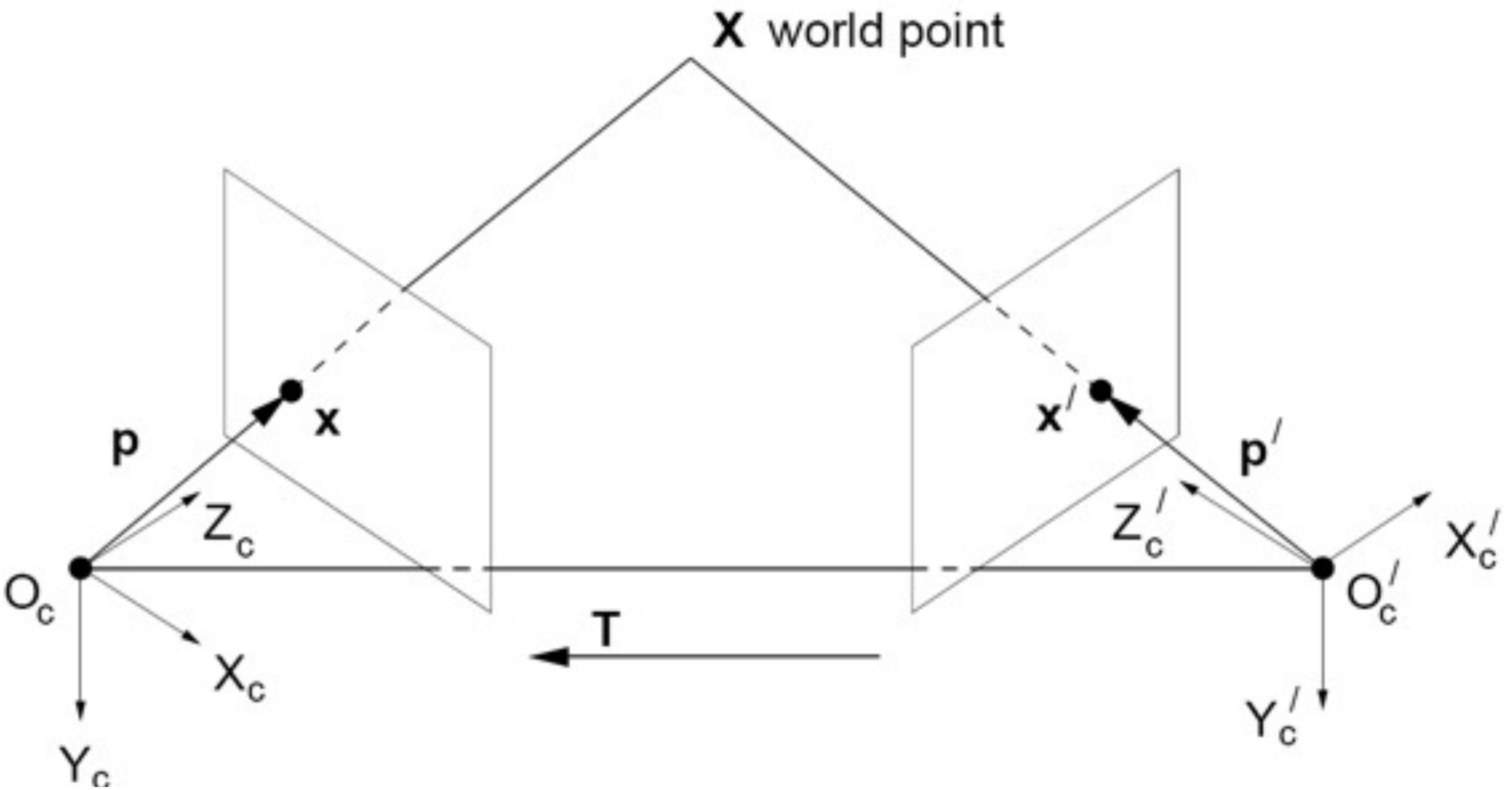
If the stereo rig is calibrated, we know :

how to rotate and translate camera reference frame 1 to get to camera reference frame 2.

$$X'_c = RX_c + T'$$

Slide credit: Kristen Grauman

From geometry to algebra



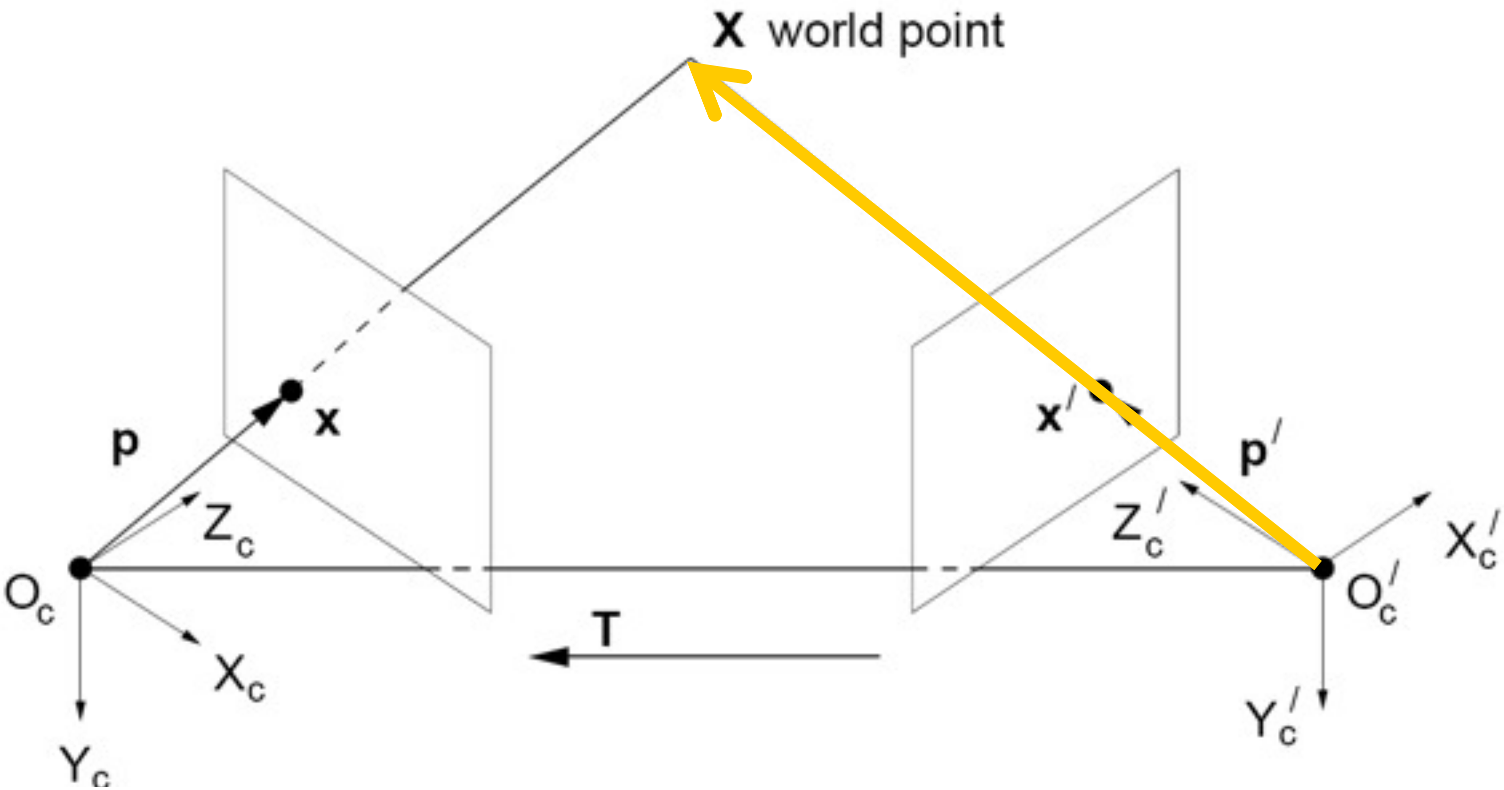
$$X' = RX + T'$$ From unprimed to primed coordinate system

$$T' \times X' = T' \times RX + T' \times T'$$ Cross with T' on both sides

$$= T' \times RX$$ Simplify

$$X' \cdot (T' \times X') = X' \cdot (T' \times RX) = 0$$ Projecting onto X', in the epipolar plane, gives 0

From geometry to algebra



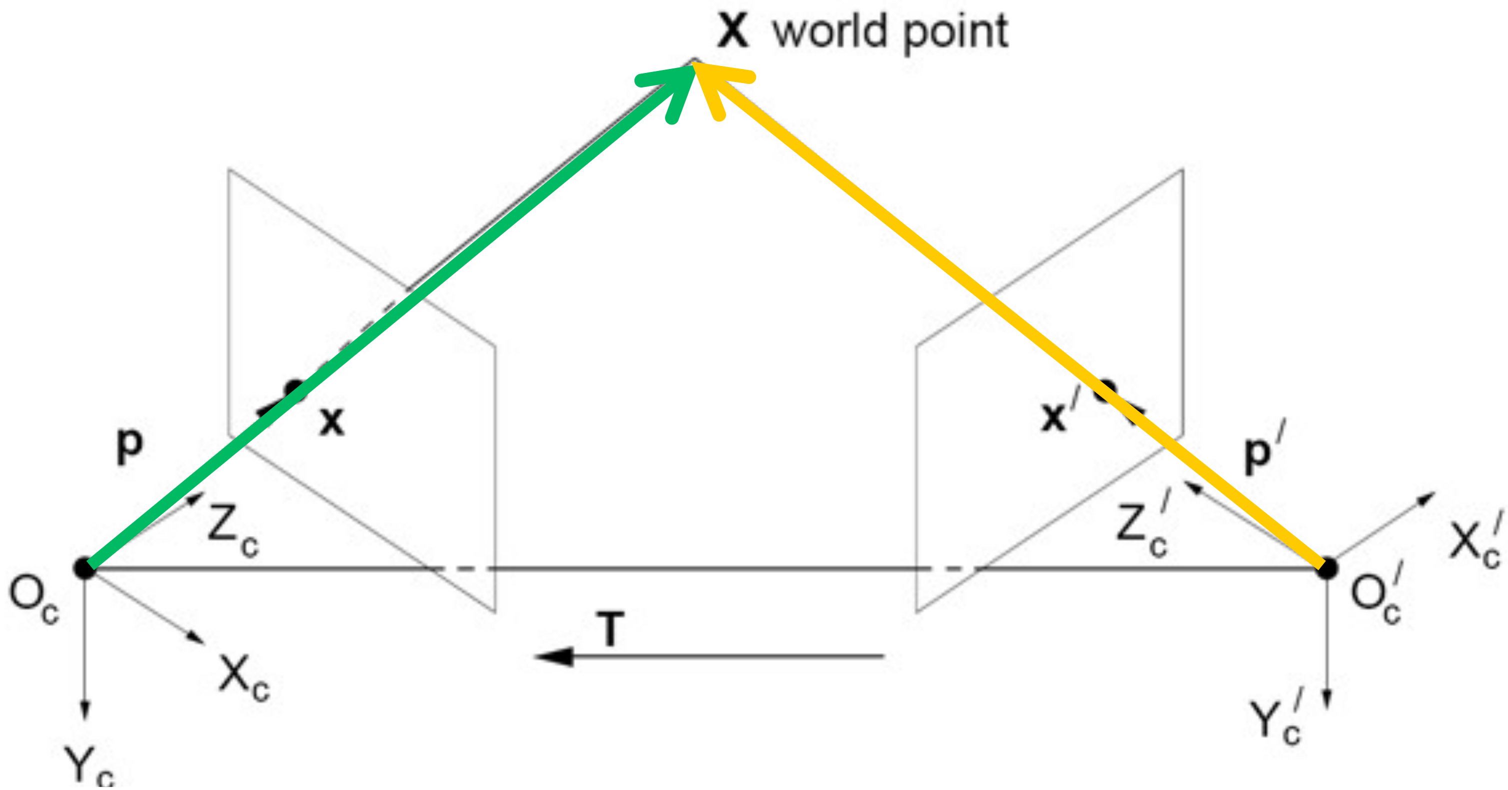
$$X' = RX + T'$$ From unprimed to primed coordinate system

$$T' \times X' = T' \times RX + T' \times T'$$ Cross with T' on both sides

$$= T' \times RX$$ Simplify

$$X' \cdot (T' \times X') = X' \cdot (T' \times RX) = 0$$ Projecting onto X', in the epipolar plane, gives 0

From geometry to algebra



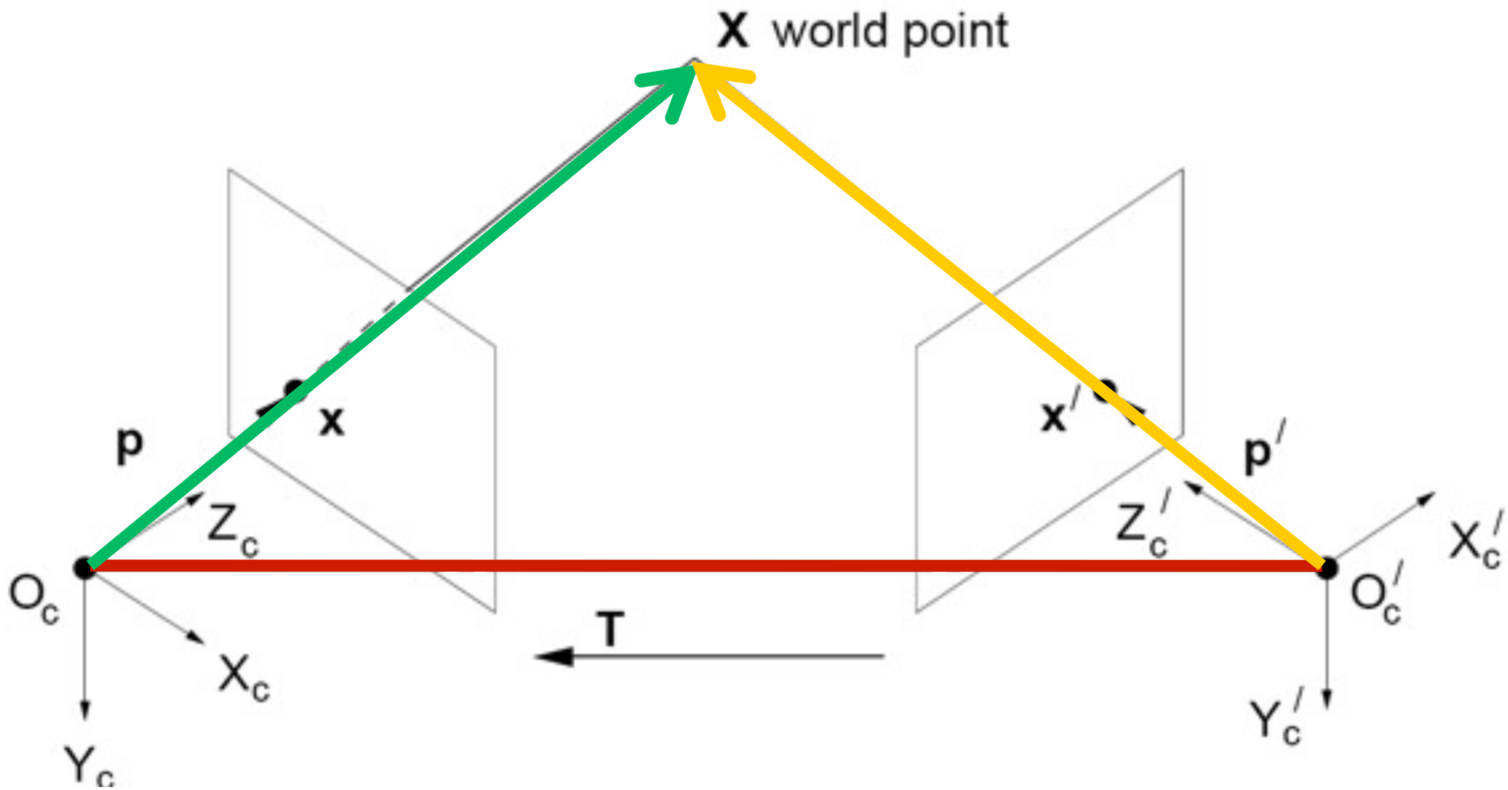
$$X' = RX + T'$$ From unprimed to primed coordinate system

$$T' \times X' = T' \times RX + T' \times T'$$ Cross with T’ on both sides

$$= T' \times RX$$ Simplify

$$X' \cdot (T' \times X') = X' \cdot (T' \times RX) = 0$$ Projecting onto X’, in the epipolar plane, gives 0

From geometry to algebra



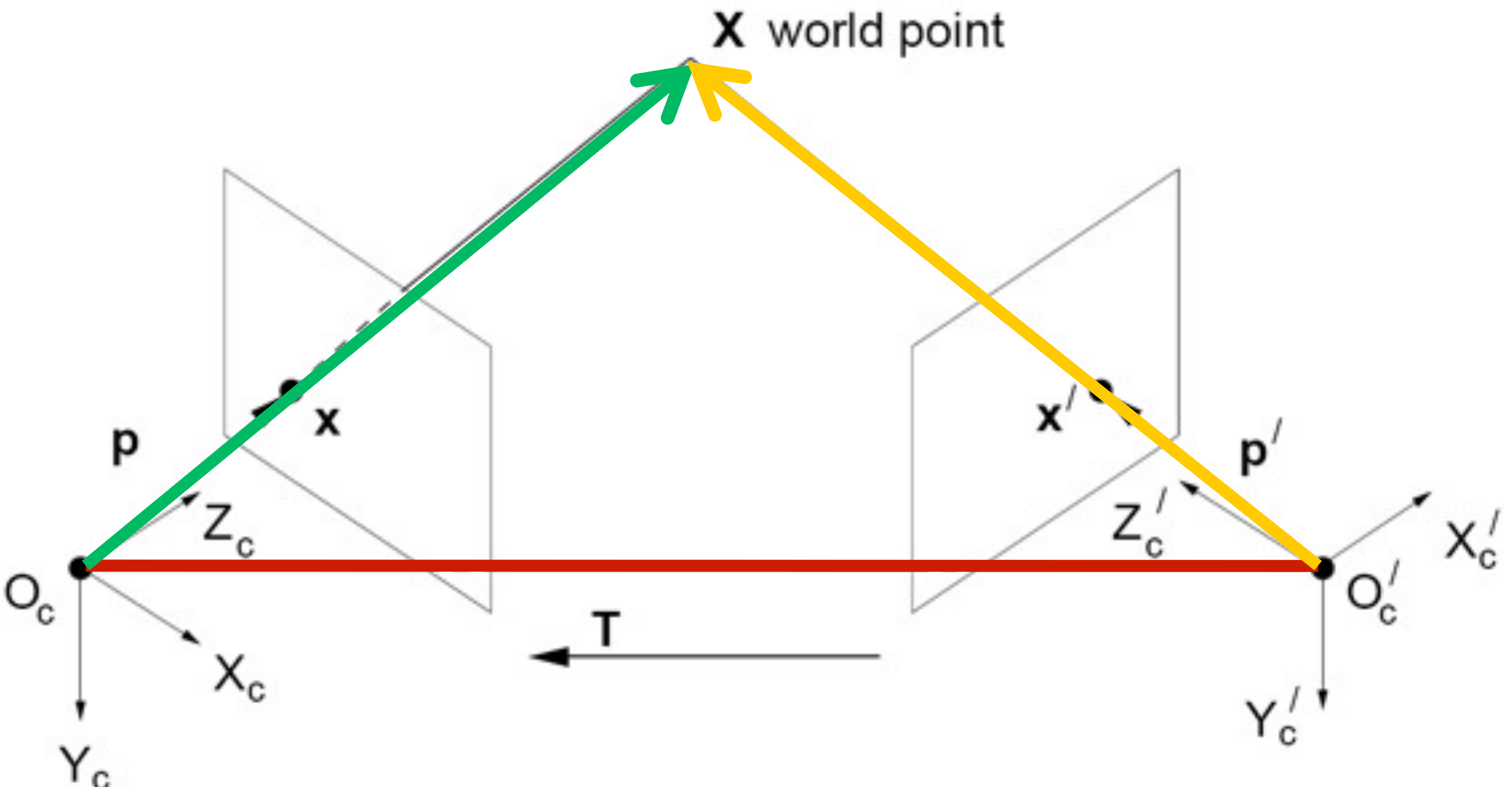
$$X' = RX + T'$$ From unprimed to primed coordinate system

$$T' \times X' = T' \times RX + T' \times T'$$ Cross with T' on both sides

$$= T' \times RX$$ Simplify

$$X' \cdot (T' \times X') = X' \cdot (T' \times RX) = 0$$ Projecting onto X', in the epipolar plane, gives 0

From geometry to algebra



$$X' = RX + T'$$

From unprimed to primed coordinate system

$$T' \times X' = T' \times RX + T' \times T'$$

Cross with T’ on both sides

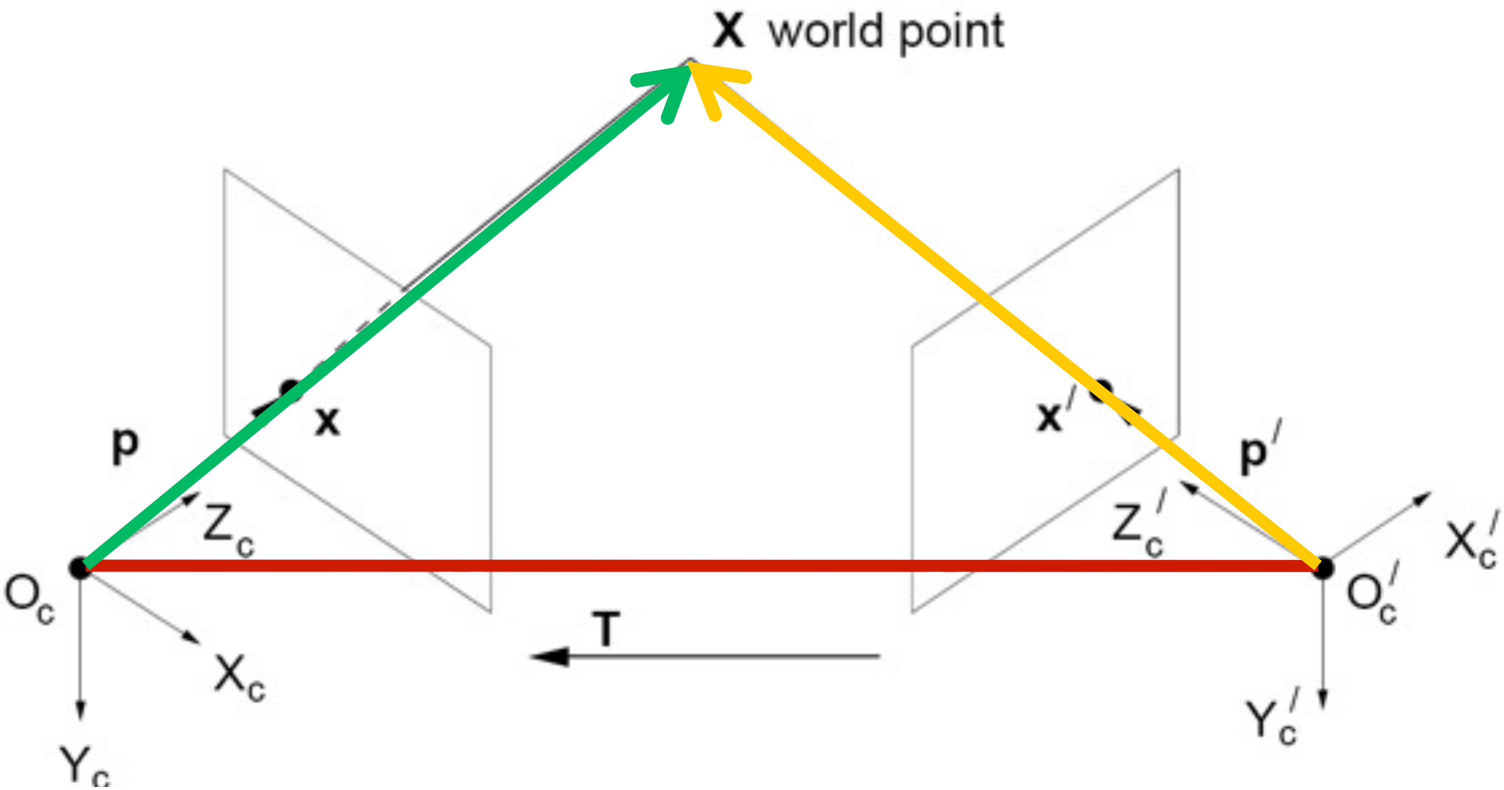
$$= T' \times RX$$

Simplify

$$X' \cdot (T' \times X') = X' \cdot (T' \times RX) = 0$$

Projecting onto X’, in the epipolar plane, gives 0

From geometry to algebra



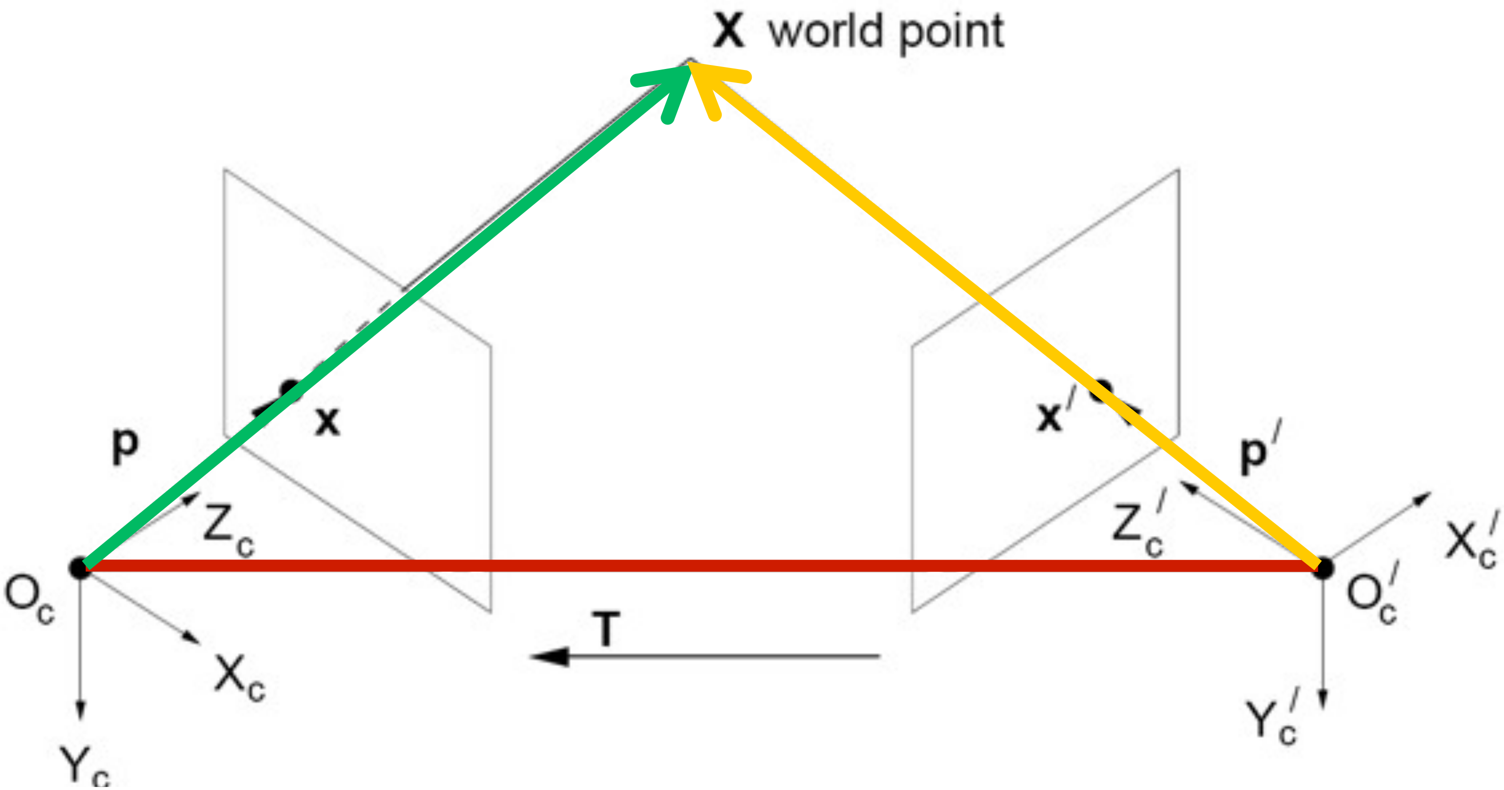
$$X' = RX + T'$$
From unprimed to primed coordinate system

$$T' \times X' = T' \times RX + T' \times T'$$
Cross with T' on both sides

$$= T' \times RX$$
Simplify

$$X' \cdot (T' \times X') = X' \cdot (T' \times RX) = 0$$
Projecting onto X', in the epipolar plane, gives 0

From geometry to algebra



$$X' = RX + T'$$

From unprimed to primed coordinate system

$$T' \times X' = T' \times RX + T' \times T'$$

Cross with T' on both sides

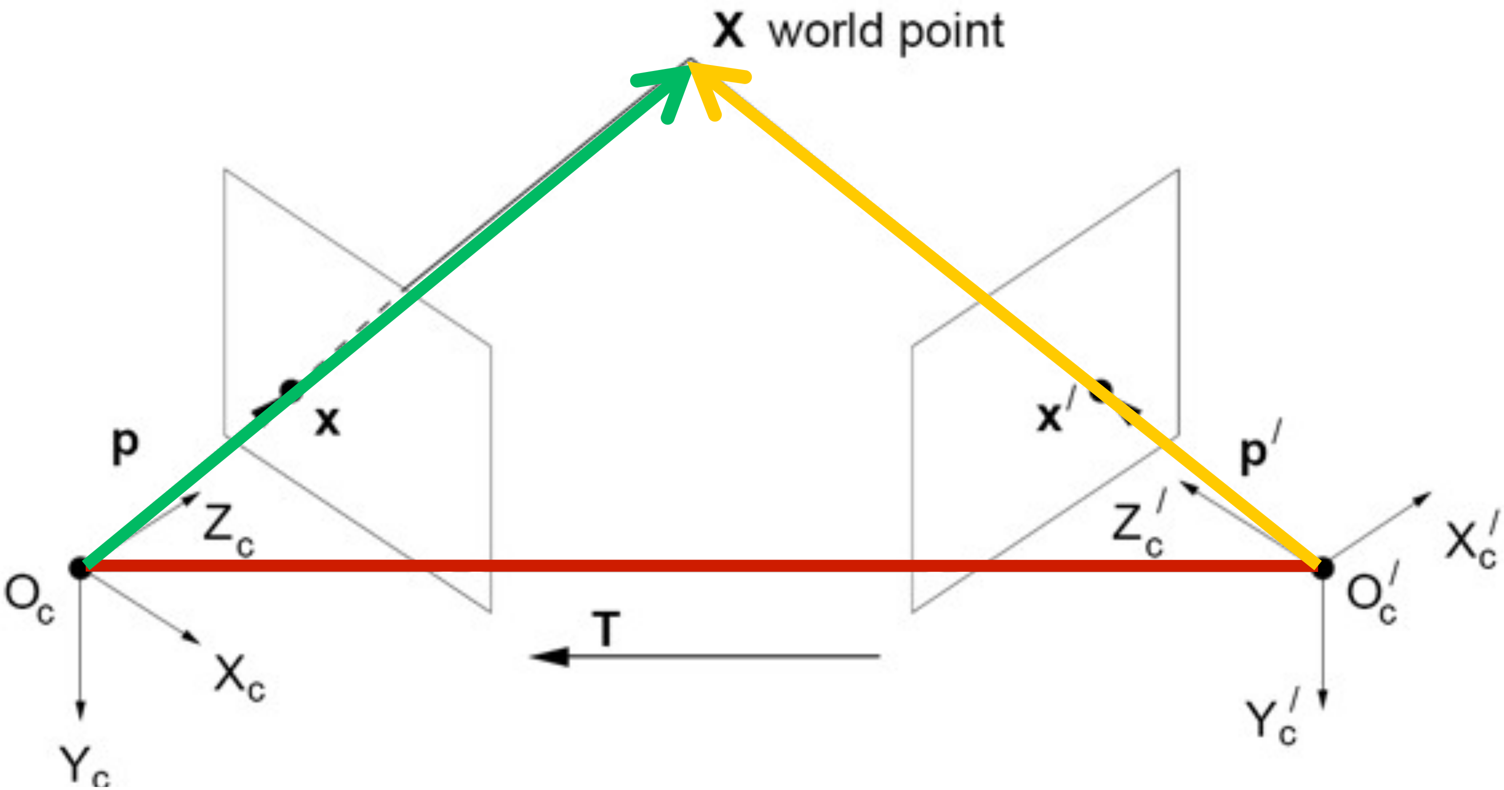
$$= T' \times RX$$

Simplify

$$X' \cdot (T' \times X') = X' \cdot (T' \times RX) = 0$$

Projecting onto X', in the epipolar plane, gives 0

From geometry to algebra



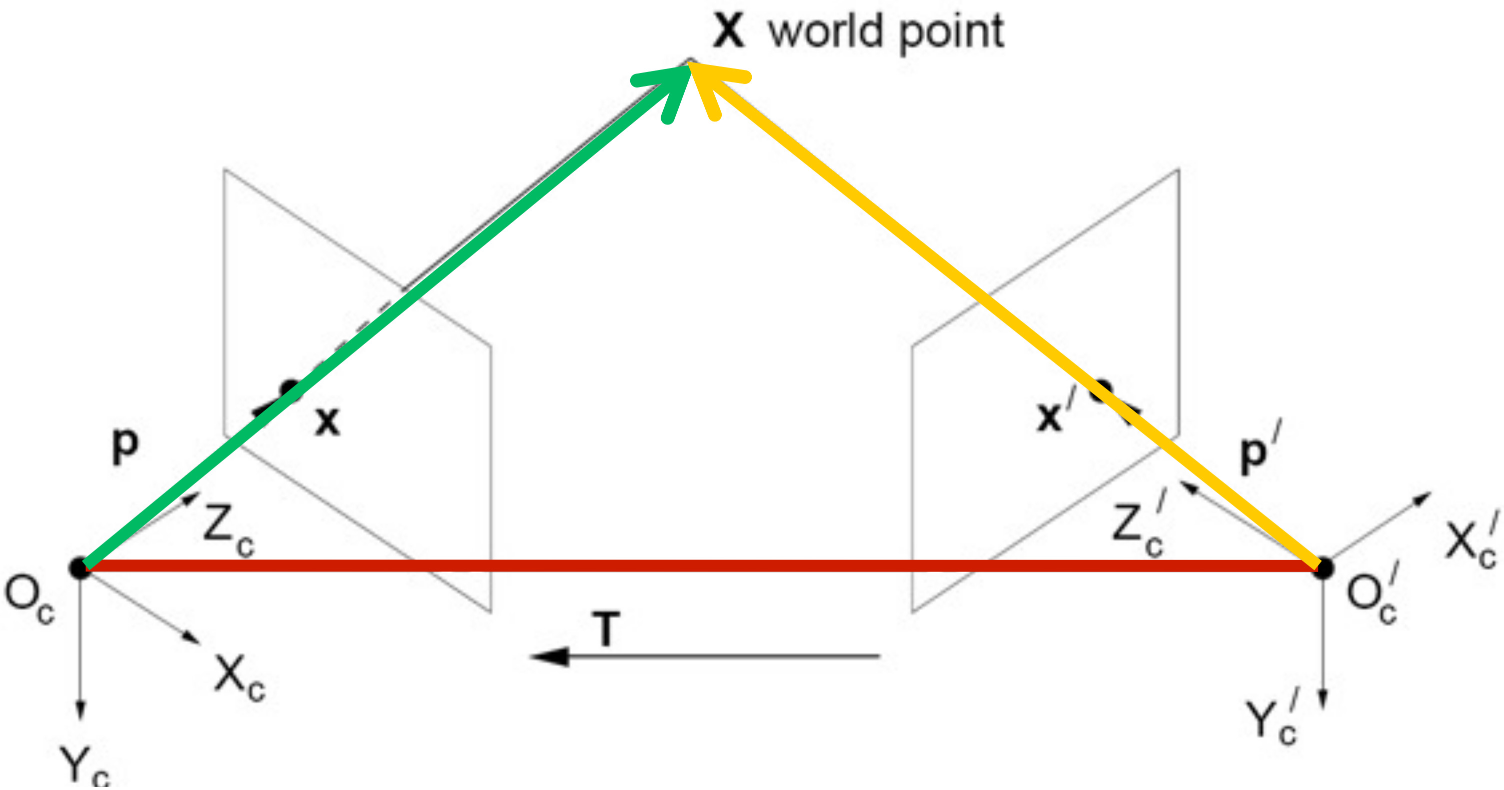
$$X' = RX + T'$$ From unprimed to primed coordinate system

$$T' \times X' = T' \times RX + T' \times T'$$ Cross with T' on both sides

$$= T' \times RX$$ Simplify

$$X' \cdot (T' \times X') = X' \cdot (T' \times RX) = 0$$ Projecting onto X', in the epipolar plane, gives 0

From geometry to algebra



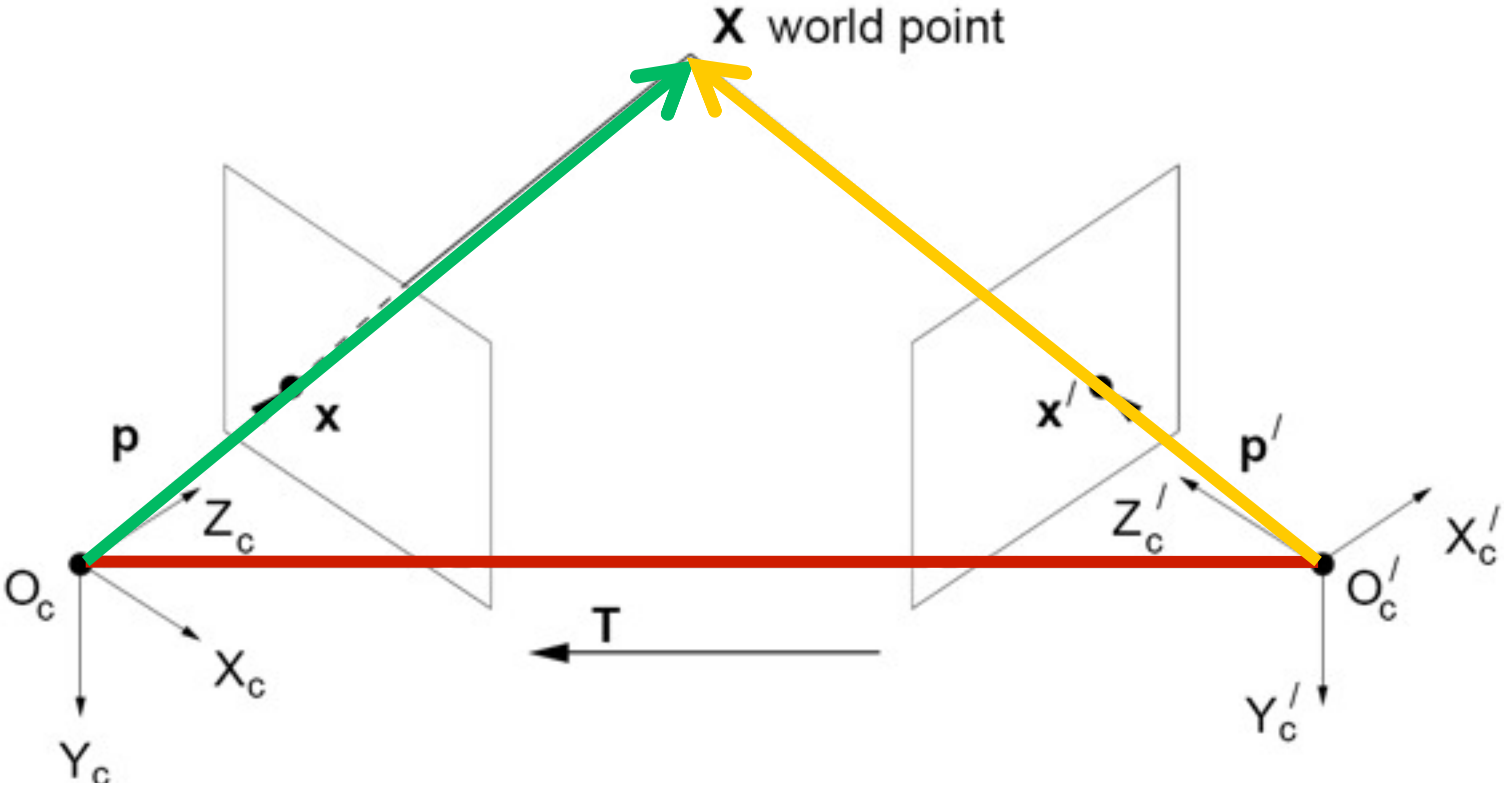
$$X' = RX + T'$$ From unprimed to primed coordinate system

$$T' \times X' = T' \times RX + T' \times T'$$ Cross with T' on both sides

$$= T' \times RX$$ Simplify

$$X' \cdot (T' \times X') = X' \cdot (T' \times RX) = 0$$ Projecting onto X', in the epipolar plane, gives 0

From geometry to algebra



$$X' = RX + T'$$
From unprimed to primed coordinate system

$$T' \times X' = T' \times RX + T' \times T'$$
Cross with T' on both sides

$$= T' \times RX$$
Simplify

$$X' \cdot (T' \times X') = X' \cdot (T' \times RX) = 0$$
Projecting onto X', in the epipolar plane, gives 0

Aside: cross product

$$\vec{a} \times \vec{b} = \vec{c}$$

Slide credit: Kristen Grauman

Aside: cross product

$$\vec{a} \times \vec{b} = \vec{c}$$

$$\vec{a} \cdot \vec{c} = 0$$
$$\vec{b} \cdot \vec{c} = 0$$

Vector cross product takes two vectors and returns a third vector that's perpendicular to both inputs.

So here, c is perpendicular to both a and b, which means the dot product = 0.

Slide credit: Kristen Grauman

Matrix form of cross product

$$\vec{a} \times \vec{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \vec{c}$$

$$\vec{a} \cdot \vec{c} = 0$$
$$\vec{b} \cdot \vec{c} = 0$$

Slide credit: Kristen Grauman

Matrix form of cross product

$$\vec{a} \times \vec{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \vec{c}$$

$$\vec{a} \cdot \vec{c} = 0$$
$$\vec{b} \cdot \vec{c} = 0$$

Can be expressed as a matrix multiplication.

Slide credit: Kristen Grauman

Matrix form of cross product

$$\vec{a} \times \vec{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \vec{c}$$

$$\bar{a} \cdot \bar{c} = 0$$
$$\bar{b} \cdot \bar{c} = 0$$

Can be expressed as a matrix multiplication.

$$\left[a_x\right] = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

Slide credit: Kristen Grauman

Matrix form of cross product

$$\vec{a} \times \vec{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \vec{c}$$

$$\vec{a} \cdot \vec{c} = 0$$
$$\vec{b} \cdot \vec{c} = 0$$

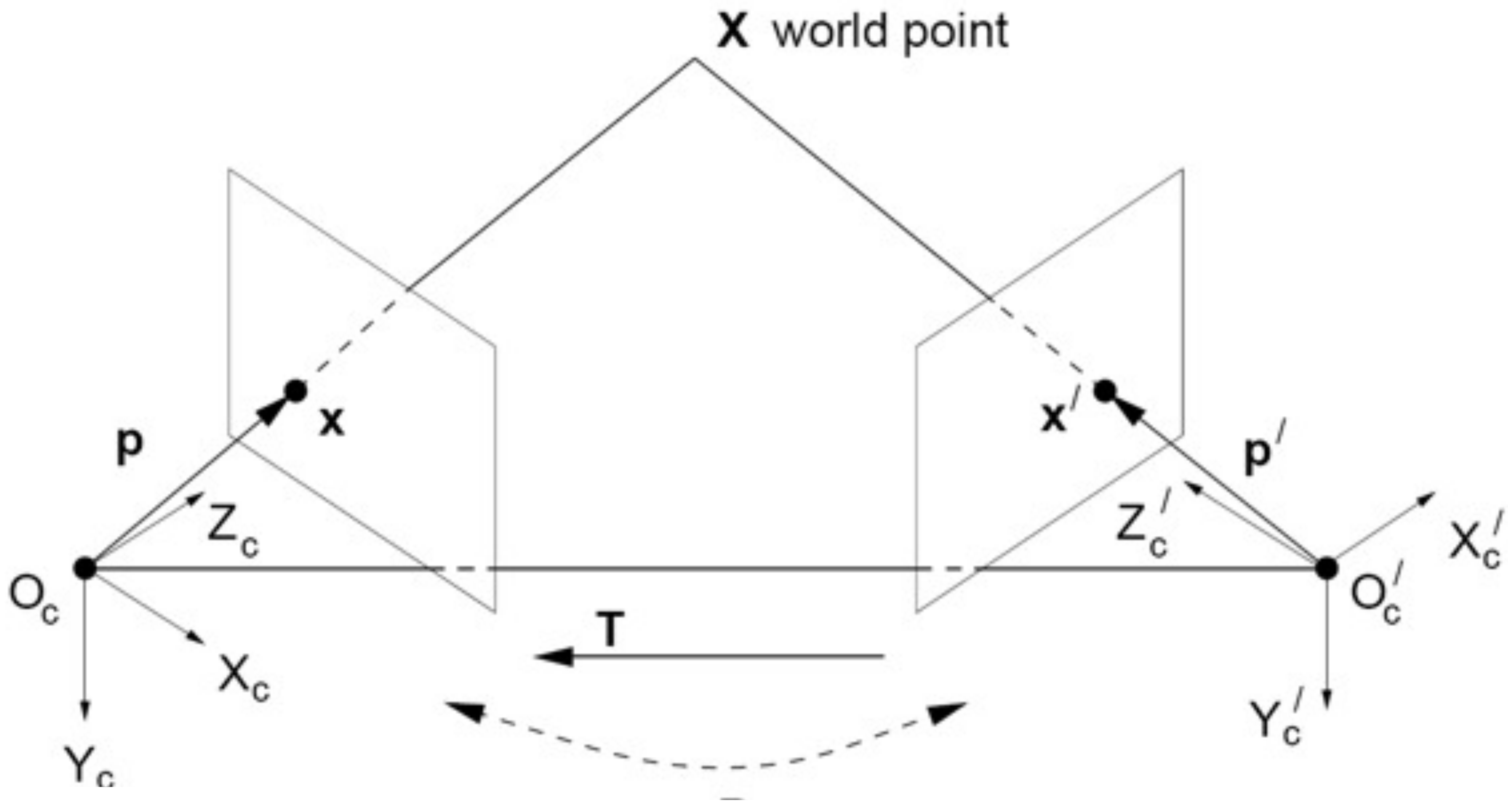
Can be expressed as a matrix multiplication.

$$[a_x] = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

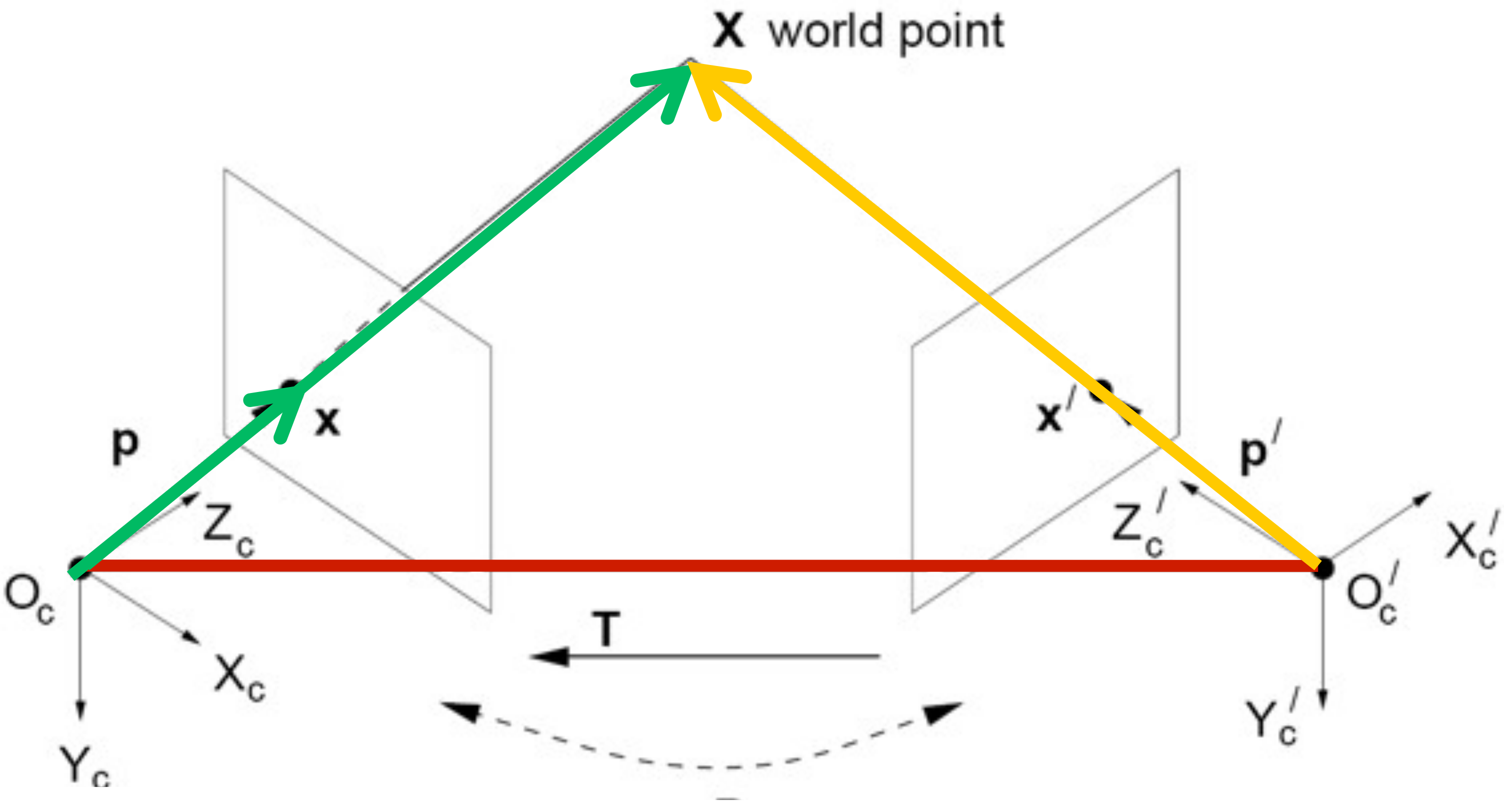
$$\boxed{\vec{a} \times \vec{b} = [a_x]\vec{b}}$$

Slide credit: Kristen Grauman

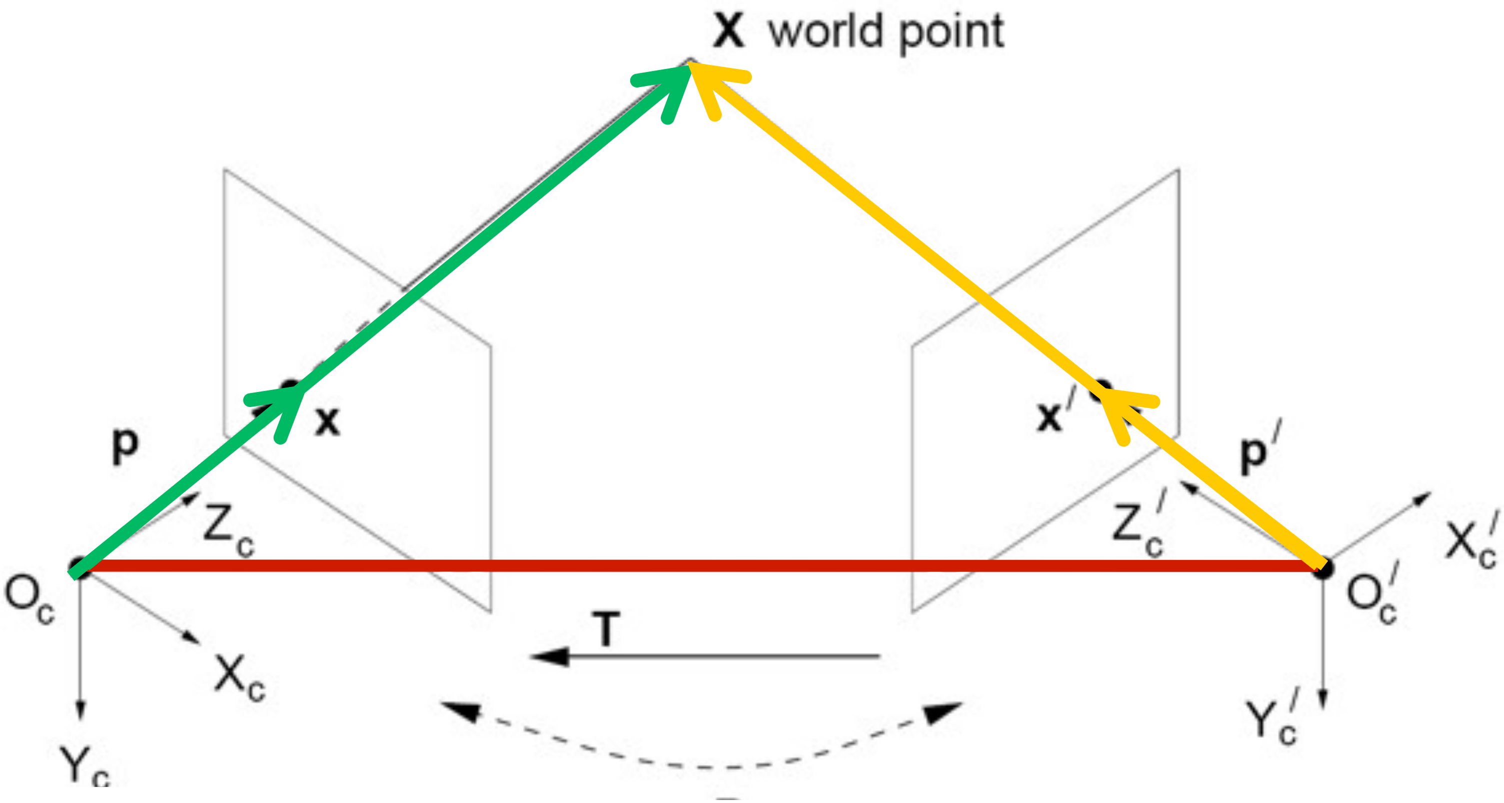
x and x' are scaled versions of X and X'



x and x' are scaled versions of X and X'



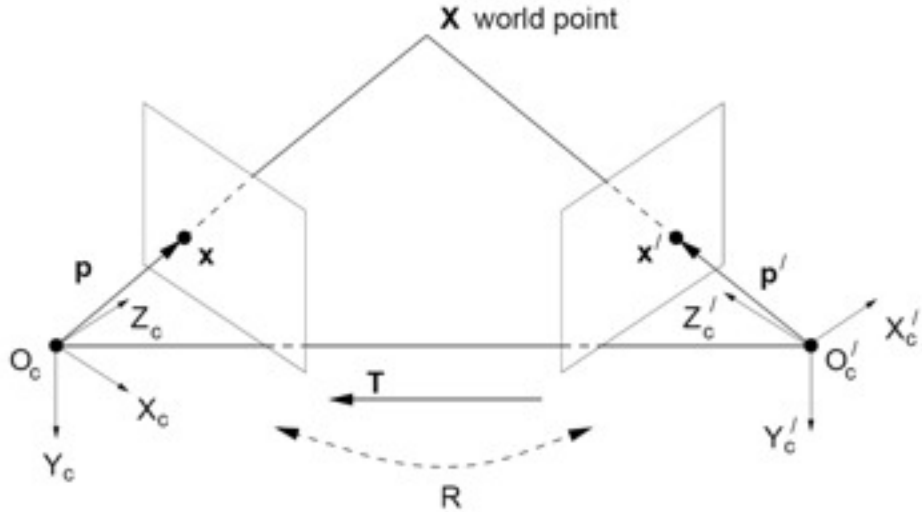
x and x' are scaled versions of X and X'



$$X' \cdot (T' \times RX) = 0$$

$$X' \cdot (T'_x RX) = 0$$

$$E = T'_x R$$

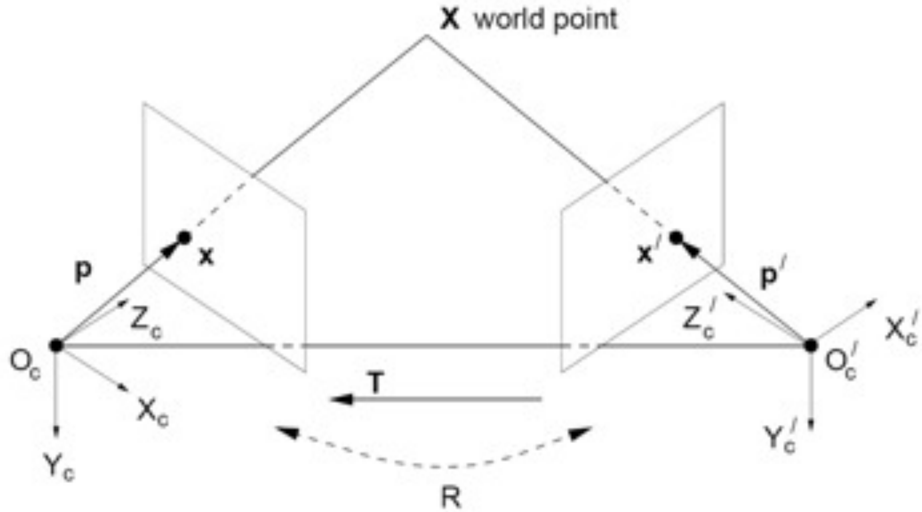


$$x'^T E x = 0$$ pts x and x' in the image planes are scaled versions of X and X'.

$$X' \cdot (T' \times RX) = 0$$

$$X' \cdot (T'_x RX) = 0$$

Let $E = T'_x R$

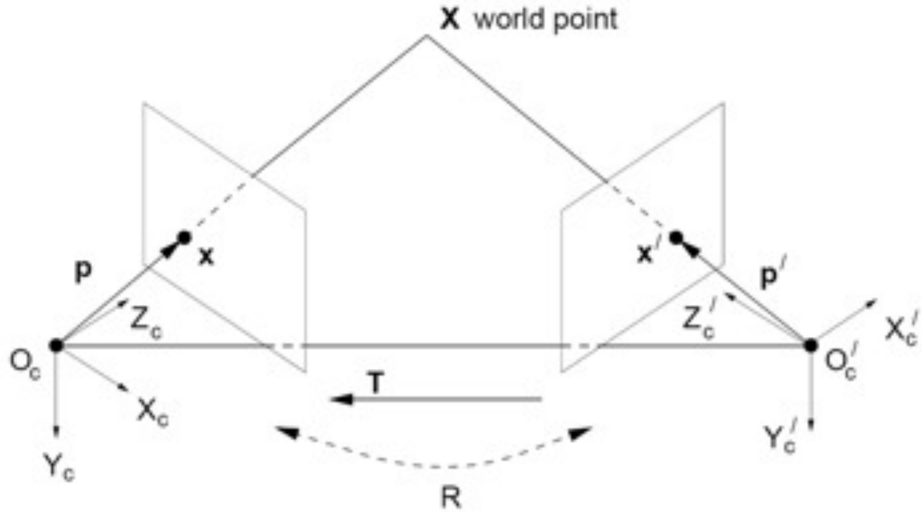


$$x'^T E x = 0$$ pts x and x' in the image planes are scaled versions of X and X'.

$$X' \cdot (T' \times RX) = 0$$

$$X' \cdot (T'_x RX) = 0$$

Let $E = T'_x R$



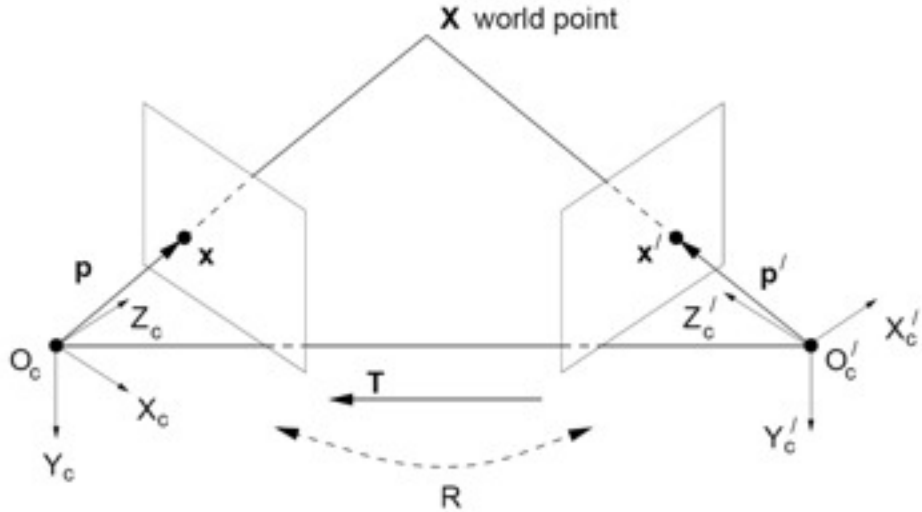
$$\mathbf{X}'^T \mathbf{E} \mathbf{X} = 0$$

$$x'^T E x = 0$$ pts x and x' in the image planes are scaled versions of X and X'.

$$X' \cdot (T' \times RX) = 0$$

$$X' \cdot (T'_x RX) = 0$$

Let $$E = T'_x R$$



$$\mathbf{X}'^T \mathbf{E} \mathbf{X} = 0$$

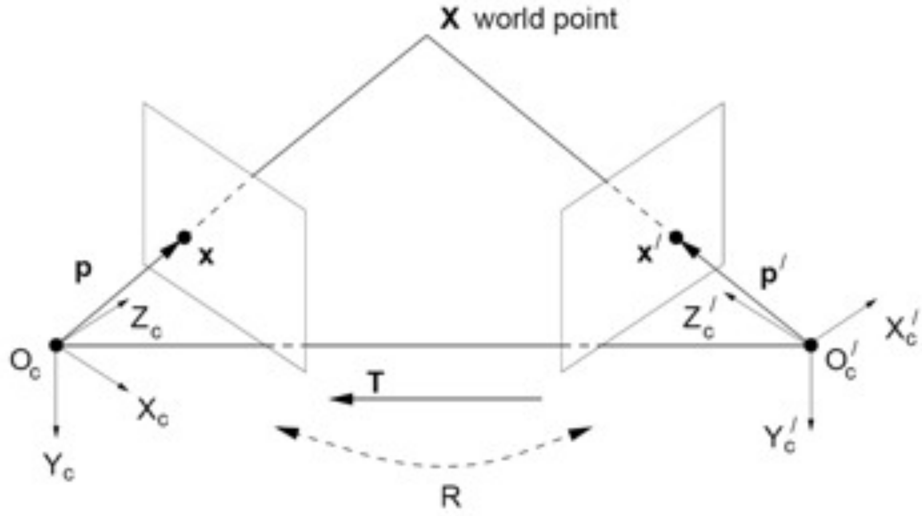
$$x'^T E x = 0$$ pts x and x' in the image planes are scaled versions of X and X'.

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

$$X' \cdot (T' \times RX) = 0$$

$$X' \cdot (T'_x RX) = 0$$

Let $E = T'_x R$



$$\mathbf{X}'^T \mathbf{E} \mathbf{X} = 0$$

$$x'^T E x = 0$$ pts x and x' in the image planes are scaled versions of X and X'.

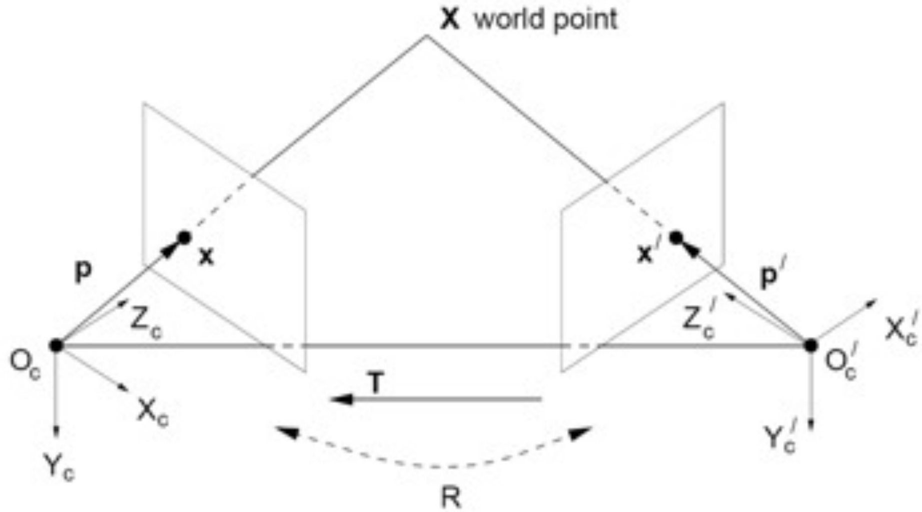
E is called the essential matrix, and it relates corresponding image points between both cameras, given the rotation and translation.

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

$$X' \cdot (T' \times RX) = 0$$

$$X' \cdot (T'_x RX) = 0$$

Let $E = T'_x R$



$$\mathbf{X}'^T \mathbf{E} \mathbf{X} = 0$$

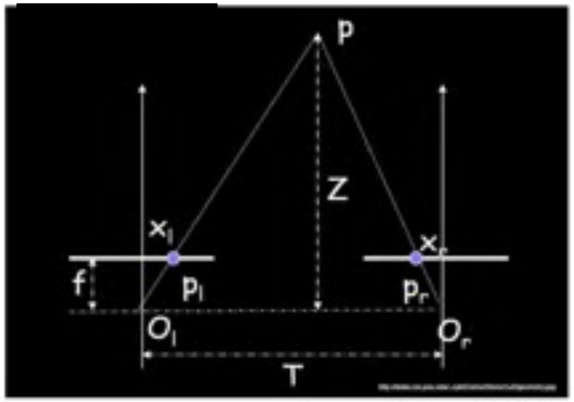
$$x'^T E x = 0$$ pts x and x' in the image planes are scaled versions of X and X'.

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 the other image is constrained to lie on line defined by above.

Note: these points are in camera coordinate systems.

Essential matrix example: parallel cameras



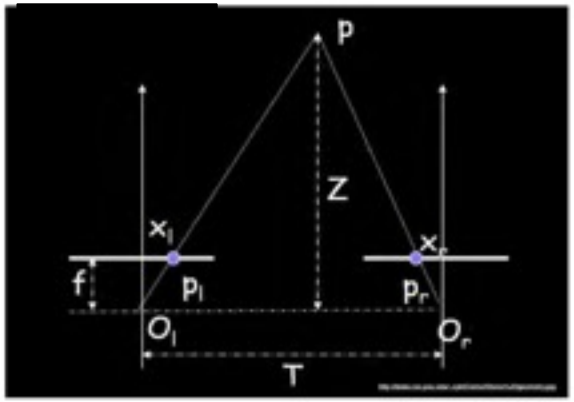
$\mathbf{R} =$

$\mathbf{T} =$

$\mathbf{E} = [\mathbf{T_x}]\mathbf{R} =$

Slide credit: Kristen Grauman

Essential matrix example: parallel cameras



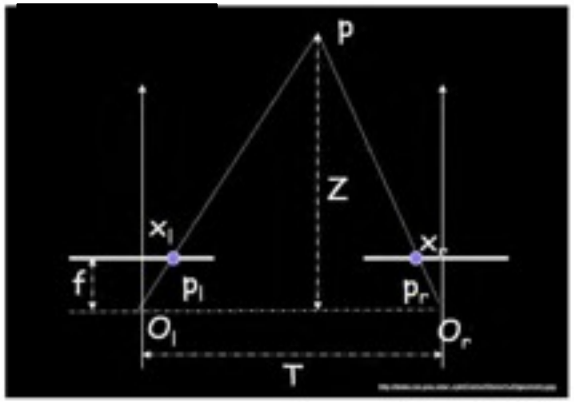
$\mathbf{R} = \mathbf{I}$

$\mathbf{T} =$

$\mathbf{E} = [\mathbf{T}_\mathbf{x}]\mathbf{R} =$

Slide credit: Kristen Grauman

Essential matrix example: parallel cameras



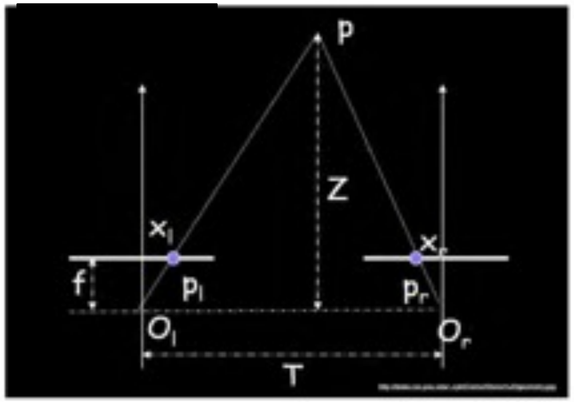
$$\mathbf{R} = \mathbf{I}$$

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

$$\mathbf{E} = [\mathbf{T}_{\mathbf{x}}]\mathbf{R} =$$

Slide credit: Kristen Grauman

Essential matrix example: parallel cameras



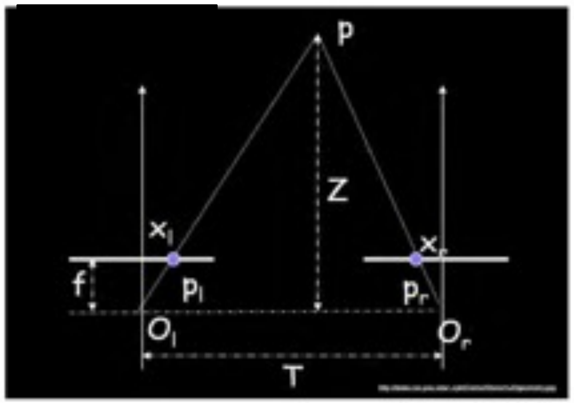
$$\mathbf{R} = \mathbf{I}$$

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

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

Slide credit: Kristen Grauman

Essential matrix example: parallel cameras



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

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

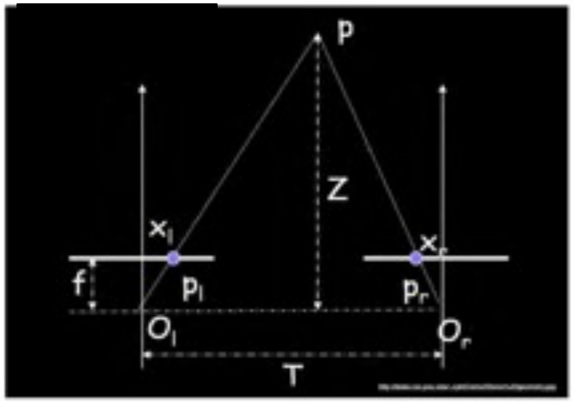
$$\mathbf{E} = [\mathbf{T}_{\mathbf{x}}]\mathbf{R} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{d} \\ \mathbf{0} & -\boldsymbol{d} & \mathbf{0} \end{pmatrix}$$

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

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

Slide credit: Kristen Grauman

Essential matrix example: parallel cameras



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

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

$$\mathbf{E} = [\mathbf{T_x}]\mathbf{R} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{pmatrix}$$

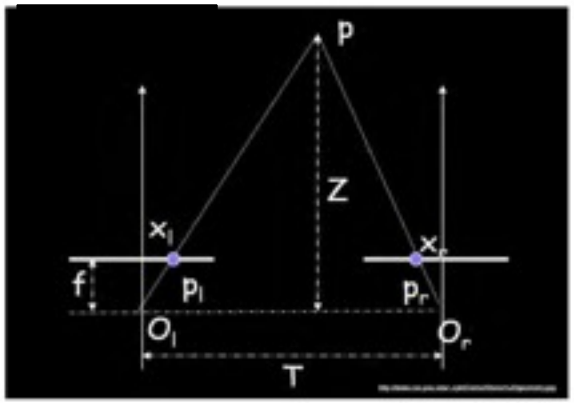
$$\mathbf{p} = [x, y, f]$$

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

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

Slide credit: Kristen Grauman

Essential matrix example: parallel cameras



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

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

$$\mathbf{E} = [\mathbf{T_x}]\mathbf{R} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{pmatrix}$$

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

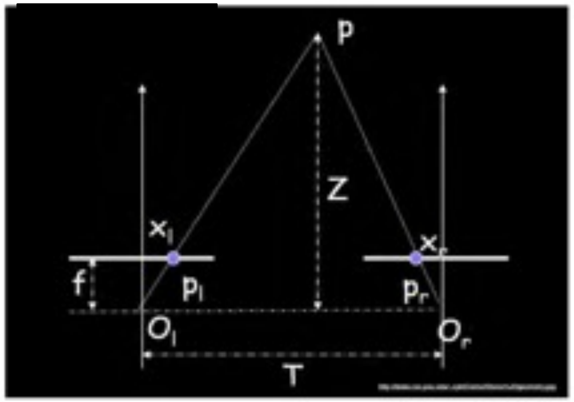
$$\mathbf{p'} = [x', y', f]$$

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

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

Slide credit: Kristen Grauman

Essential matrix example: parallel cameras



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

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

$$\mathbf{E} = [\mathbf{T_x}]\mathbf{R} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{pmatrix}$$

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

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

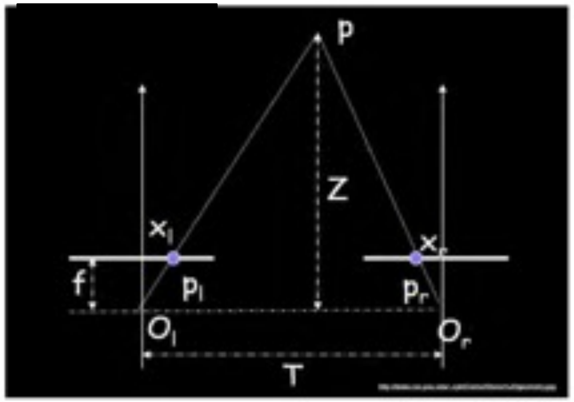
$$\mathbf{p}'^{\mathrm{T}}\mathbf{E}\mathbf{p} = 0$$

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

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

Slide credit: Kristen Grauman

Essential matrix example: parallel cameras



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

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

$$\mathbf{E} = [\mathbf{T_x}]\mathbf{R} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{pmatrix}$$

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

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

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

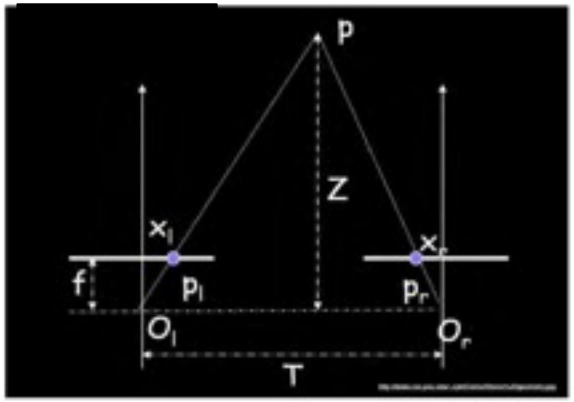
$$\begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ f \end{bmatrix} = 0$$

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

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

Slide credit: Kristen Grauman

Essential matrix example: parallel cameras



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

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

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

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

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

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

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

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

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

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

Slide credit: Kristen Grauman

image I(x,y)

Disparity map D(x,y)

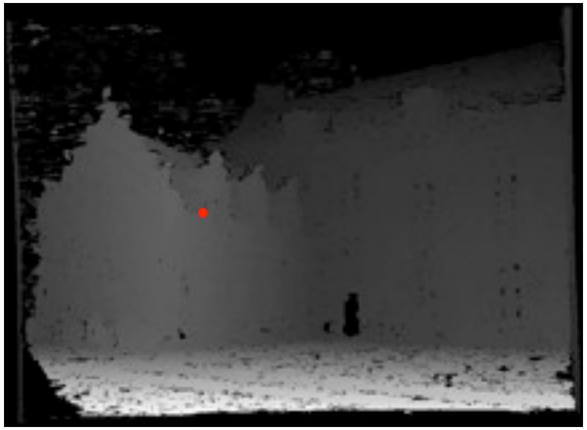


image I´(x´,y´)

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

Slide credit: Kristen Grauman

image I(x,y)

Disparity map D(x,y)

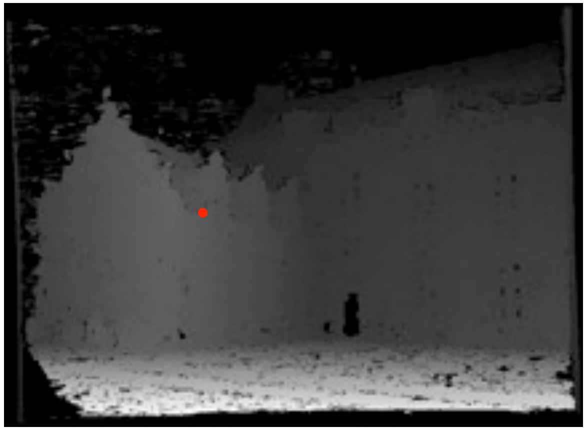


image I´(x´,y´)

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

Slide credit: Kristen Grauman

image I(x,y)

Disparity map D(x,y)

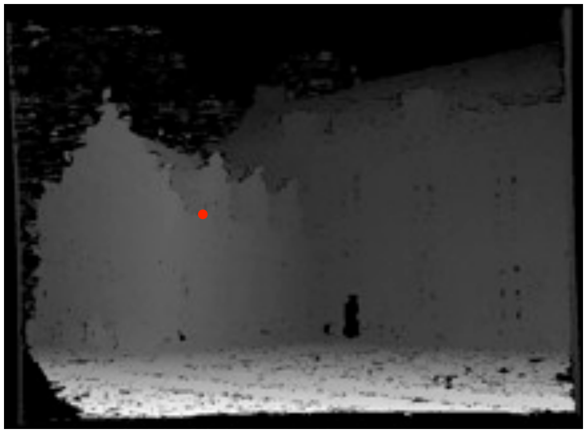


image I´(x´,y´)

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

What about when cameras' optical axes are not parallel?

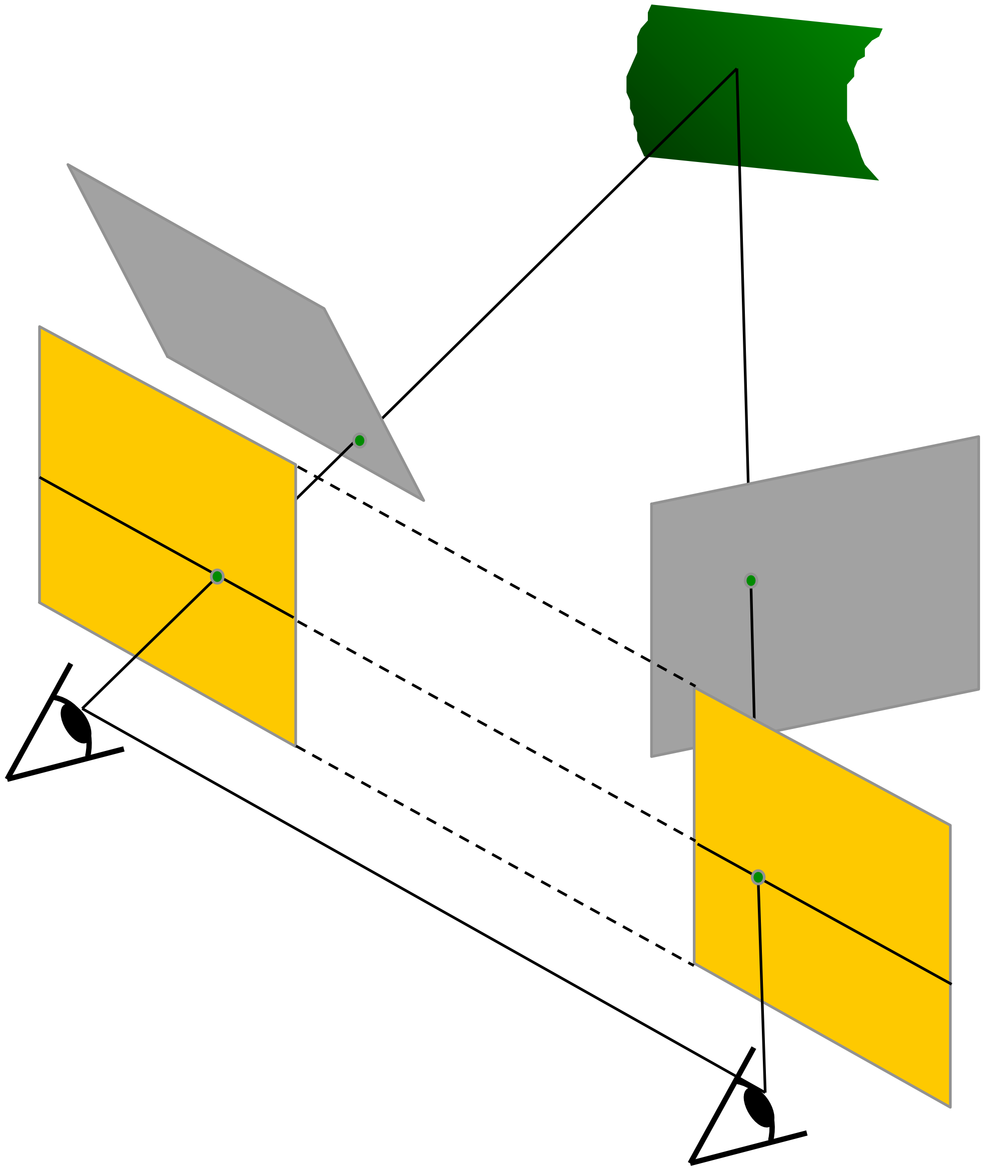
Slide credit: Kristen Grauman

Stereo Topics

- Special, simple system, main idea
- More general camera conditions, epipolar constraints
  - epipolar geometry
  - epipolar algebra
- **Image rectification**
- Stereo matching (likelihood term)
- Stereo regularization (prior term)
- Inference
  - dynamic programming
  - graph cuts
- Structured light

Stereo image rectification

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

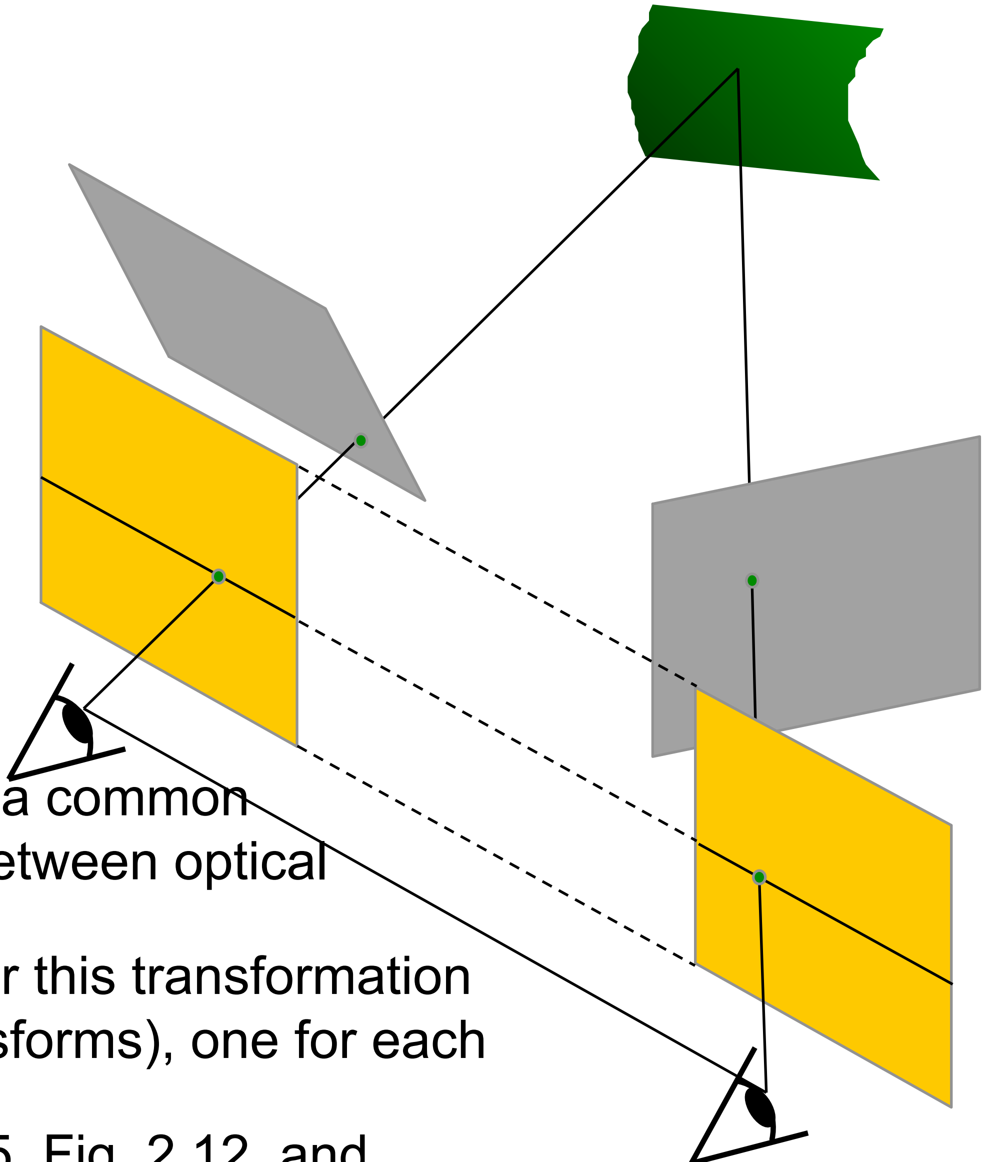


Adapted from Li Zhang

Slide credit: Kristen Grauman

Stereo image rectification

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

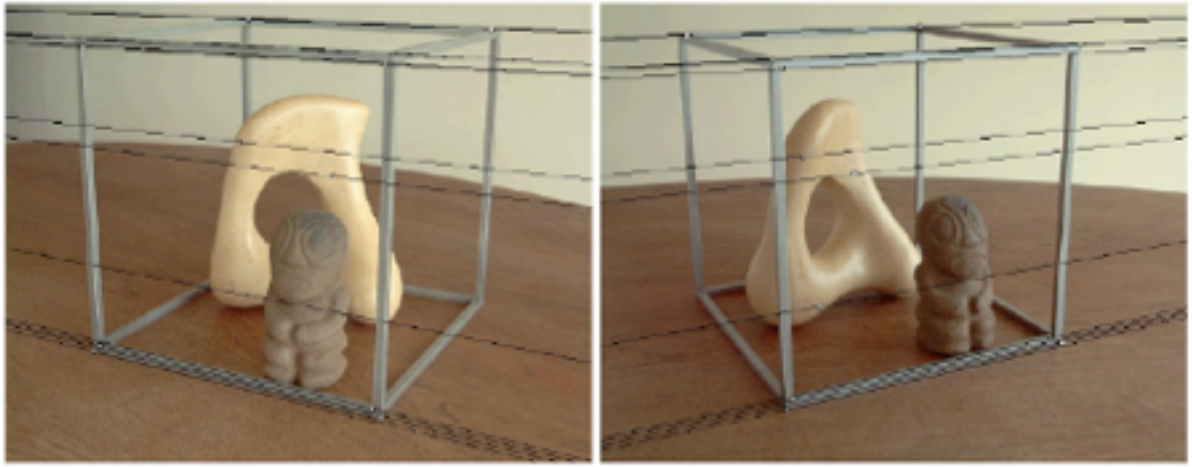


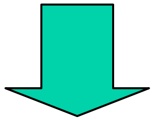
- Reproject image planes onto a common plane parallel to the line between optical centers
- Pixel motion is horizontal after this transformation
- Two homographies (3x3 transforms), one for each input image reprojection
- See Szeliski book, Sect. 2.1.5, Fig. 2.12, and "Mapping from one camera to another" p. 56

Adapted from Li Zhang

Slide credit: Kristen Grauman

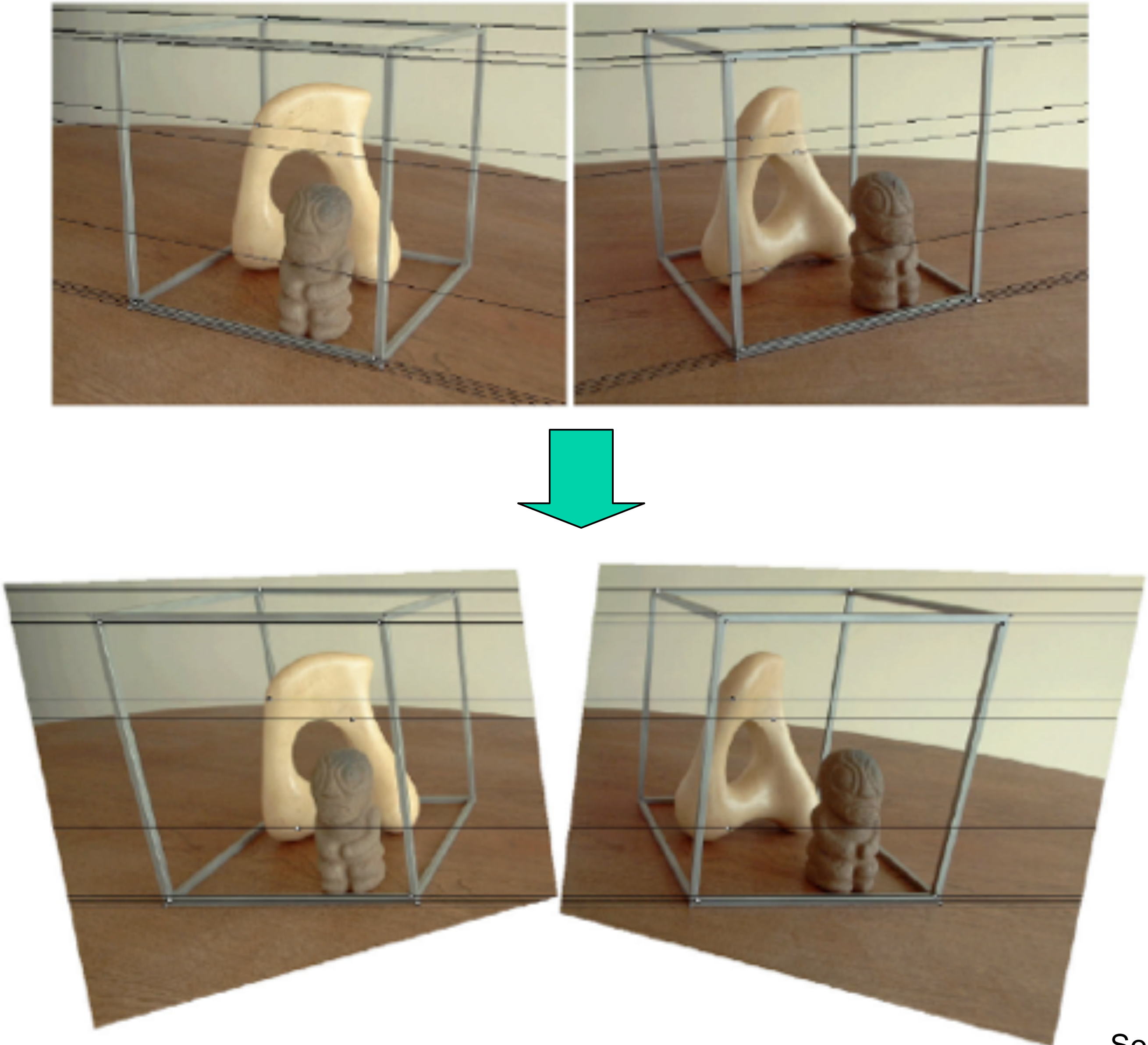
Stereo image rectification: example





Source: Alyosha Efros

Stereo image rectification: example

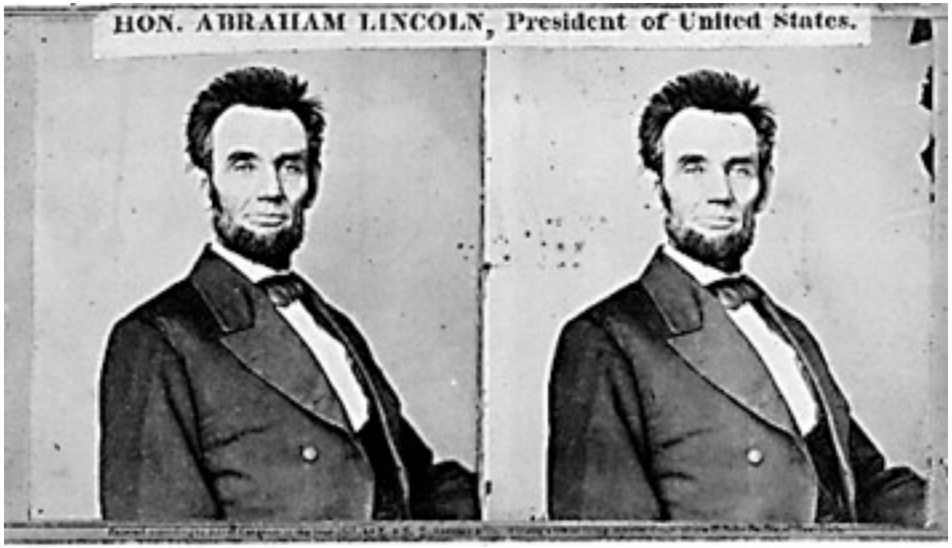


Source: Alyosha Efros

Stereo Topics

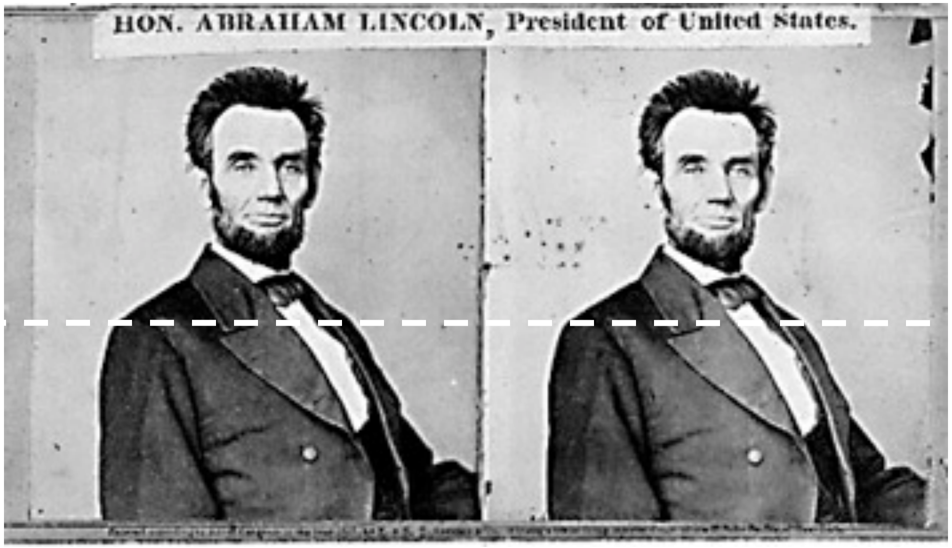
- Special, simple system, main idea
- More general camera conditions, epipolar constraints
  - epipolar geometry
  - epipolar algebra
- Image rectification
- **Stereo matching (likelihood term)**
- Stereo regularization (prior term)
- Inference
  - dynamic programming
  - graph cuts
- Structured light

Your basic stereo algorithm



Slide credit: Rick Szeliski

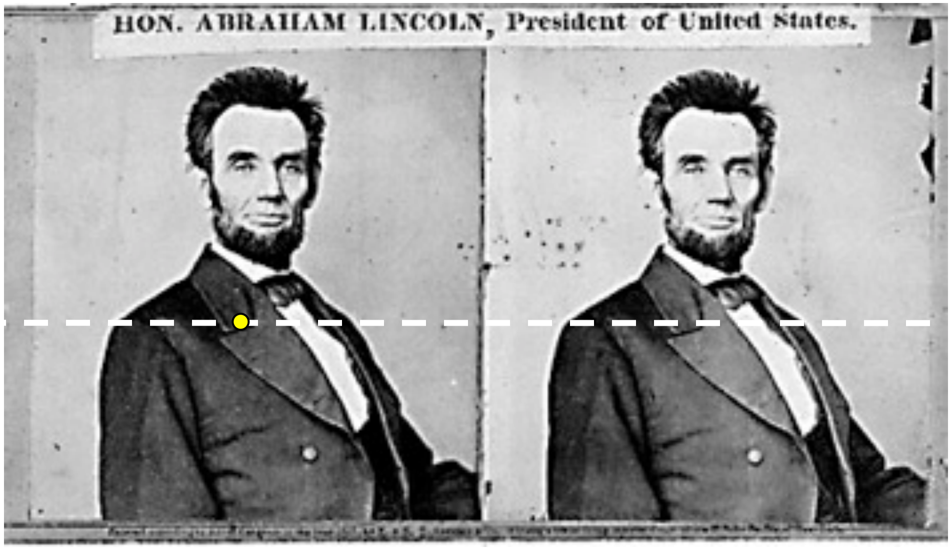
Your basic stereo algorithm



For each epipolar line

Slide credit: Rick Szeliski

Your basic stereo algorithm

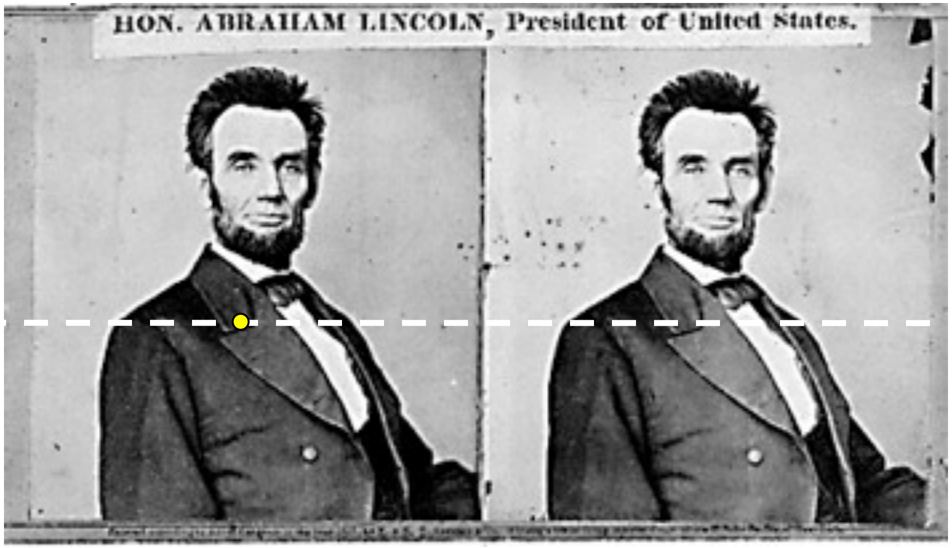


For each epipolar line

- For each pixel in the left image

Slide credit: Rick Szeliski

Your basic stereo algorithm



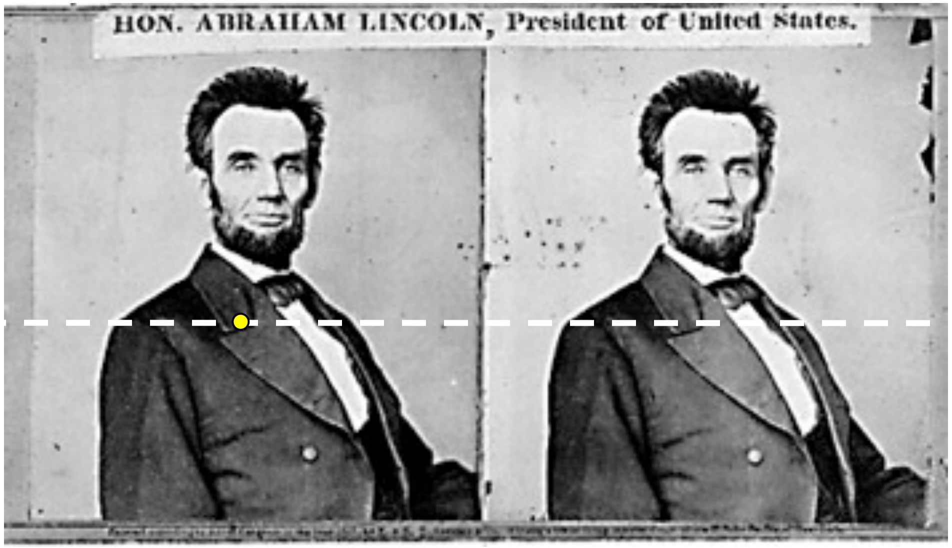
For each epipolar line

For each pixel in the left image

- compare with every pixel on same epipolar line in right image

Slide credit: Rick Szeliski

Your basic stereo algorithm



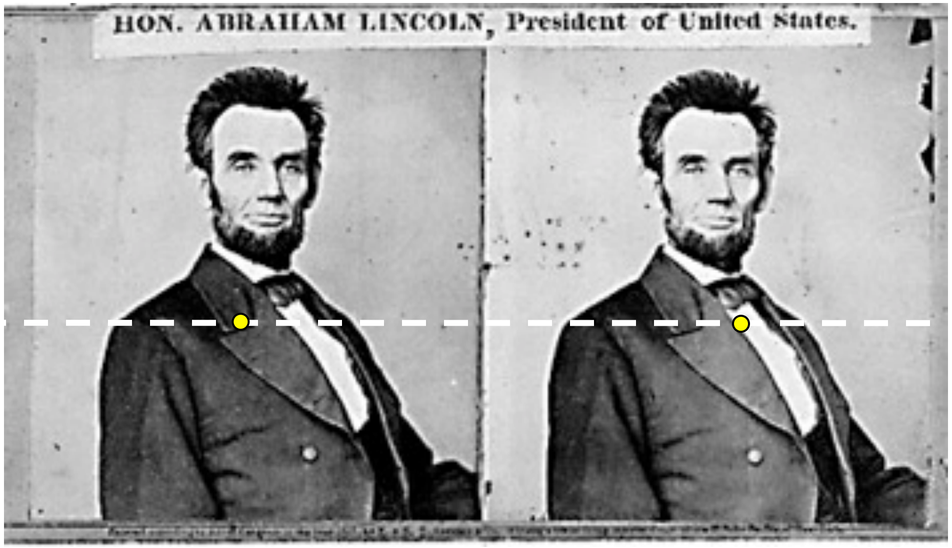
For each epipolar line

For each pixel in the left image

- compare with every pixel on same epipolar line in right image
- pick pixel with minimum match cost

Slide credit: Rick Szeliski

Your basic stereo algorithm



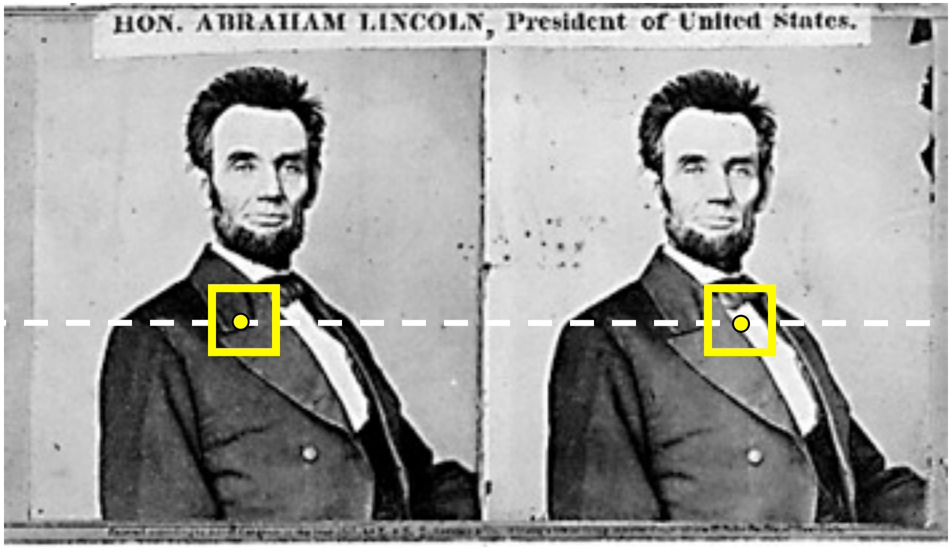
For each epipolar line

For each pixel in the left image

- compare with every pixel on same epipolar line in right image
- pick pixel with minimum match cost

Slide credit: Rick Szeliski

Your basic stereo algorithm



For each epipolar line

For each pixel in the left image

- compare with every pixel on same epipolar line in right image
- pick pixel with minimum match cost

Improvement: match ***windows***

Slide credit: Rick Szeliski

Image block matching

How do we determine correspondences?

- *block matching* or *SSD* (sum squared differences)

$$E(x,y;d) = \sum_{(x',y')\in N(x,y)} [I_L(x'+d,y') - I_R(x',y')]^2$$

d is the *disparity* (horizontal motion)

How big should the neighborhood be?

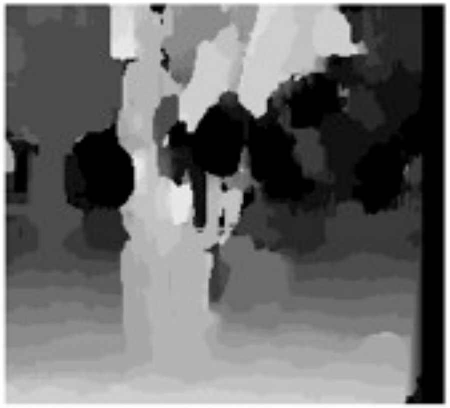
Slide credit: Rick Szeliski

Neighborhood size

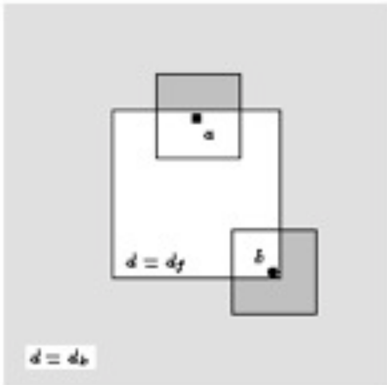
Smaller neighborhood: more details

Larger neighborhood: fewer isolated mistakes

w = 3



w = 20



Slide credit: Rick Szeliski

Matching criteria

Slide credit: Rick Szeliski

Matching criteria

Raw pixel values (correlation)

Slide credit: Rick Szeliski

Matching criteria

Raw pixel values (correlation)

Band-pass filtered images [Jones & Malik 92]

Slide credit: Rick Szeliski

Matching criteria

Raw pixel values (correlation)

Band-pass filtered images [Jones & Malik 92]

“Corner” like features [Zhang, …]

Slide credit: Rick Szeliski

Matching criteria

Raw pixel values (correlation)

Band-pass filtered images [Jones & Malik 92]

“Corner” like features [Zhang, …]

Edges [many people…]

Slide credit: Rick Szeliski

Matching criteria

Raw pixel values (correlation)

Band-pass filtered images [Jones & Malik 92]

“Corner” like features [Zhang, …]

Edges [many people…]

Gradients [Seitz 89; Scharstein 94]

Slide credit: Rick Szeliski

Matching criteria

Raw pixel values (correlation)

Band-pass filtered images [Jones & Malik 92]

“Corner” like features [Zhang, …]

Edges [many people…]

Gradients [Seitz 89; Scharstein 94]

Rank statistics [Zabih & Woodfill 94]

Slide credit: Rick Szeliski

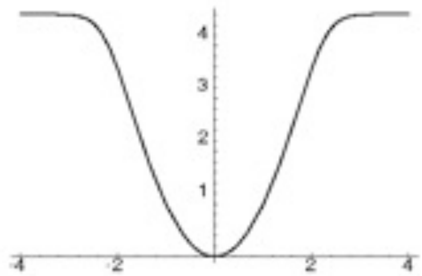
Local evidence framework

1. For every disparity, compute *raw* matching costs

$$E_0(x, y; d) = \rho(I_L(x' + d, y') - I_R(x', y'))$$

Why use a robust function?

- occlusions, other outliers

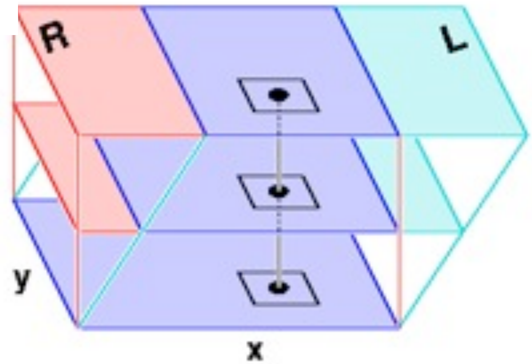


Can also use alternative match criteria

Local evidence framework

2. Aggregate costs spatially

$$E(x,y;d) = \sum_{(x',y') \in N(x,y)} E_0(x',y',d)$$



- Here, we are using a *box filter* (efficient moving average implementation)
- Can also use weighted average, [non-linear] diffusion…

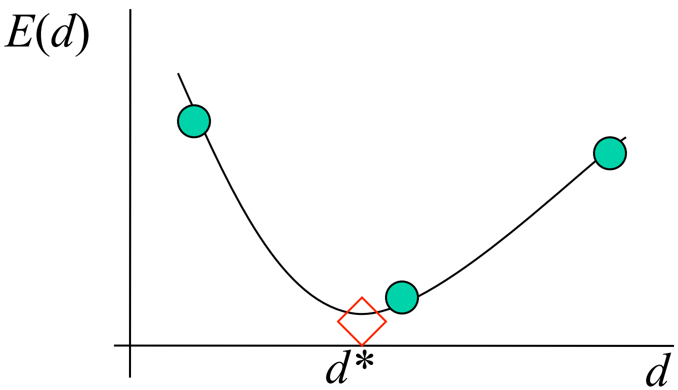
Slide credit: Rick Szeliski

Local evidence framework

3. Choose winning disparity at each pixel

$$d(x,y) = \arg\min_d E(x,y;d)$$

4. Interpolate to *sub-pixel* accuracy



Slide credit: Rick Szeliski

Local evidence framework

Advantages:

- gives detailed surface estimates
- fast algorithms based on moving averages
- sub-pixel disparity estimates and confidence

Limitations:

- narrow baseline $\Rightarrow$ noisy estimates
- fails in textureless areas
- gets confused near occlusion boundaries

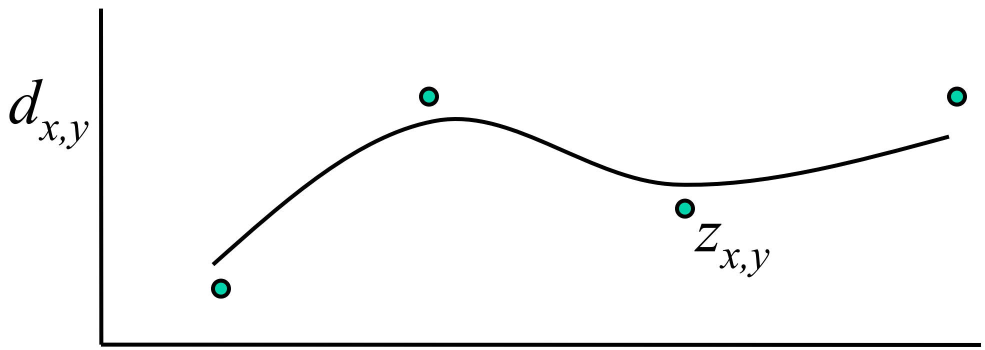
Slide credit: Rick Szeliski

Stereo Topics

- Special, simple system, main idea
- More general camera conditions, epipolar constraints
  - epipolar geometry
  - epipolar algebra
- Image rectification
- Stereo matching (likelihood term)
- **Stereo regularization (prior term)**
- Inference
  - dynamic programming
  - graph cuts
- Structured light

Energy minimization

1-D example: approximating splines

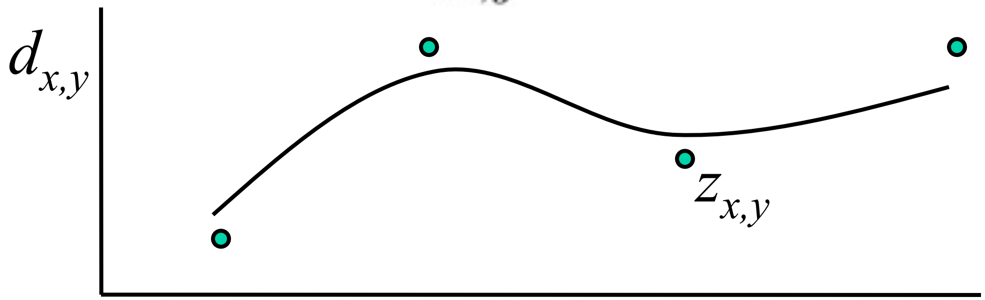


Slide credit: Rick Szeliski

Energy minimization

1-D example: approximating splines

$$
\begin{aligned}
E_{\text{total}}(\mathbf{d}) &= E_{\text{data}}(\mathbf{d}) + \lambda E_{\text{smoothness}}(\mathbf{d}) \\
E_{\text{data}}(\mathbf{d}) &= \sum_{x,y} (d_{x,y} - z_{x,y})^2 \\
E_{\text{membrane}}(\mathbf{d}) &= \sum_{x,y} (d_{x,y} - d_{x-1,y})^2 \\
E_{\text{thin plate}}(\mathbf{d}) &= \sum_{x,y} (2d_{x,y} - d_{x-1,y} - d_{x+1,y})^2
\end{aligned}
$$



Slide credit: Rick Szeliski

Stereo Topics

- Special, simple system, main idea
- More general camera conditions, epipolar constraints
  - epipolar geometry
  - epipolar algebra
- Image rectification
- Stereo matching (likelihood term)
- Stereo regularization (prior term)
- **Inference**
  - **dynamic programming**
  - graph cuts
- Structured light

Dynamic programming

Evaluate best cumulative cost at each pixel

$$
\begin{aligned}
E_{\text{total}}(\mathbf{d}) &= E_{\text{data}}(\mathbf{d}) + \lambda E_{\text{smoothness}}(\mathbf{d}) \\
E_{\text{data}}(\mathbf{d}) &= \sum_{x,y} (d_{x,y} - z_{x,y})^2 \\
E_{\text{smoothness}}(\mathbf{d}) &= \sum_{x,y} |d_{x,y} - d_{x-1,y}|
\end{aligned}
$$

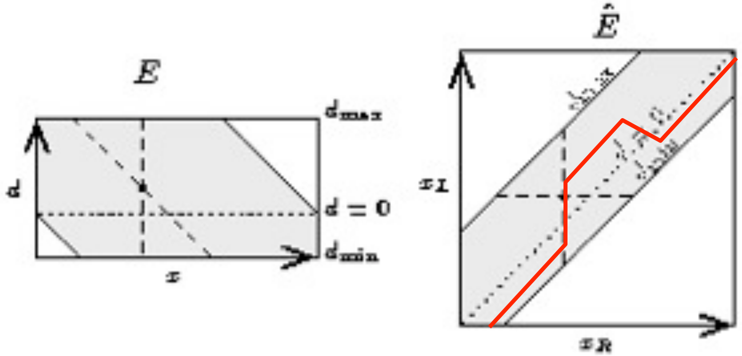
Slide credit: Rick Szeliski

Dynamic programming

1-D cost function

$$E(\mathbf{d}) = \sum_{x,y} \rho_P(d_{x+1,y} - d_{x,y}) + \sum_{x,y} E_0(x,y;d)$$

$$\tilde{E}(x,y,d) = E_0(x,y;d) + \min_{d'} \left(\tilde{E}(x-1,y,d') + \rho_P(d_{x,y} - d'_{x-1,y})\right)$$



Dynamic programming

Disparity space image and min. cost path

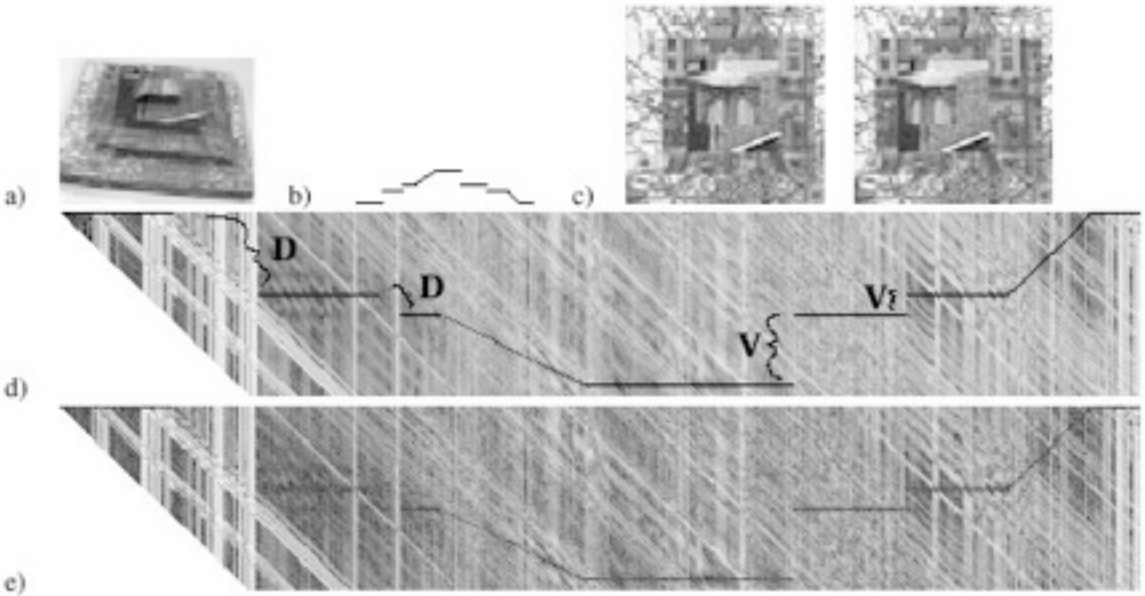


Fig. 4. This figure shows (a) a model of the stereo sloping wedding cake that we will use as a test example, (b) a depth profile through the center of the sloping wedding cake, (c) a simulated, noise-free image pair of the cake, (d) the enhanced, cropped, correlation $DSI-$ representation for the image pair in (c), and (e) the enhanced, cropped, correlation DSI for a noisy sloping wedding cake (SNR = 18 dB). In (d), the regions labeled "D" mark diagonal gaps in the matching path caused by regions occluded in the left image. The regions labeled "V" mark vertical jumps in the path caused by regions occluded in the right image.

Slide credit: Rick Szeliski

Dynamic programming

Sample result
(note horizontal streaks)

[Intille & Bobick]

a)
b)
c)

Fig. 12. Results of two stereo algorithms on Figure 1. (a) Original left image. (b) Cox *et al.* algorithm[14], and (c) the algorithm described in this paper.

Slide credit: Rick Szeliski

Stereo Topics

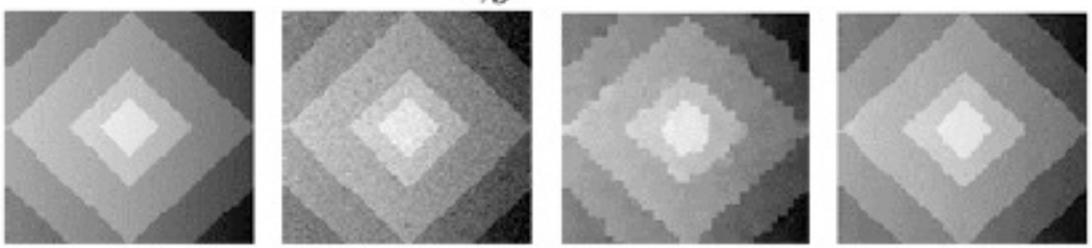
- Special, simple system, main idea
- More general camera conditions, epipolar constraints
  - epipolar geometry
  - epipolar algebra
- Image rectification
- Stereo matching (likelihood term)
- Stereo regularization (prior term)
- **Inference**
  - dynamic programming
  - **graph cuts**
- Structured light

graph cuts home page: http://www.cs.cornell.edu/~rdz/graphcuts.html

Graph cuts

Solution technique for general 2D problem

$$
\begin{aligned}
E_{\text{total}}(\mathbf{d}) &= E_{\text{data}}(\mathbf{d}) + \lambda E_{\text{smoothness}}(\mathbf{d}) \\
E_{\text{data}}(\mathbf{d}) &= \sum_{x,y} f_{x,y}(d_{x,y}) \\
E_{\text{smoothness}}(\mathbf{d}) &= \sum_{x,y} \rho(d_{x,y} - d_{x-1,y}) \\
&\quad + \sum_{x,y} \rho(d_{x,y} - d_{x,y-1})
\end{aligned}
$$



(a) original image (b) observed image (c) local min w.r.t. standard moves (d) local min w.r.t. α-expansion moves

Slide credit: Rick Szeliski

Graph cuts

α-β swap

α expansion

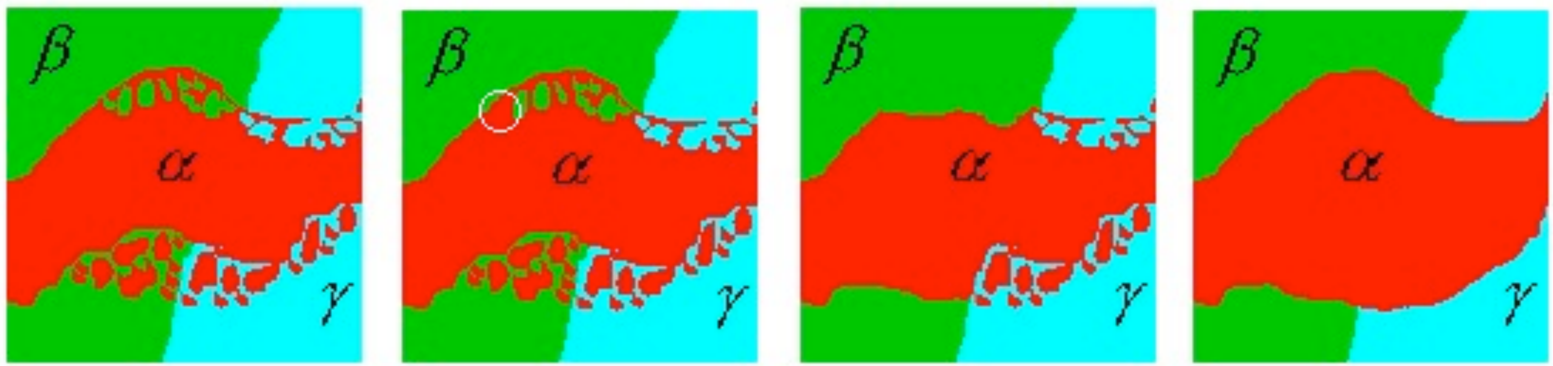
modify smoothness penalty based on edges

compute best possible match within integer disparity

Slide credit: Rick Szeliski

Graph cuts

Two different kinds of moves:



(a) initial labeling (b) standard move (c) α-β-swap (d) α-expansion

Slide credit: Rick Szeliski

Bayesian inference

Formulate as statistical inference problem

Prior model $p_P(\boldsymbol{d})$

Measurement model $p_M(I_L, I_R | \boldsymbol{d})$

Posterior model

$$p_M(\boldsymbol{d} \mid I_L, I_R) \propto p_P(\boldsymbol{d})\, p_M(I_L, I_R | \boldsymbol{d})$$

Maximum a Posteriori (MAP estimate):

maximize $p_M(\boldsymbol{d} \mid I_L, I_R)$

Slide credit: Rick Szeliski

Markov Random Field

Probability distribution on disparity field *d*(*x*,*y*)

$$p_P(d_{x,y}|\mathbf{d}) = p_P(d_{x,y}|\{d_{x',y'}, (x',y') \in \mathcal{N}(x,y)\})$$

$$p_P(\mathbf{d}) = \frac{1}{Z_P} e^{-E_P(\mathbf{d})}$$

$$E_P(\mathbf{d}) = \sum_{x,y} \rho_P(d_{x+1,y} - d_{x,y}) + \rho_P(d_{x,y+1} - d_{x,y})$$

Enforces *smoothness* or *coherence* on field

Slide credit: Rick Szeliski

Measurement model

Likelihood of intensity correspondence

$$p_M(I_L, I_R|\mathbf{d}) = \frac{1}{Z_M} e^{-E_0(x,y;d)}$$

$$E_0(x, y; d) = \rho(I_L(x' + d, y') - I_R(x', y'))$$

Corresponds to Gaussian noise for quadratic ρ

Slide credit: Rick Szeliski

MAP estimate

Maximize posterior likelihood

$$\begin{aligned} E(\mathbf{d}) &= -\log p(\mathbf{d}|I_L, I_R) \\ &= \sum_{x,y} \rho_P(d_{x+1,y} - d_{x,y}) + \rho_P(d_{x,y+1} - d_{x,y}) \\ &\quad + \sum_{x,y} \rho_M(I_L(x + d_{x,y}, y) - I_R(x, y)) \end{aligned}$$

Equivalent to *regularization* (energy minimization with smoothness constraints)

Slide credit: Rick Szeliski

Why Bayesian estimation?

Slide credit: Rick Szeliski

Why Bayesian estimation?

Principled way of determining cost function

Slide credit: Rick Szeliski

Why Bayesian estimation?

Principled way of determining cost function

Explicit model of noise and prior knowledge

Why Bayesian estimation?

Principled way of determining cost function

Explicit model of noise and prior knowledge

Admits a wide variety of optimization algorithms:

Slide credit: Rick Szeliski

Why Bayesian estimation?

Principled way of determining cost function

Explicit model of noise and prior knowledge

Admits a wide variety of optimization algorithms:

- gradient descent (local minimization)

Why Bayesian estimation?

Principled way of determining cost function

Explicit model of noise and prior knowledge

Admits a wide variety of optimization algorithms:

- gradient descent (local minimization)
- stochastic optimization (Gibbs Sampler)

Slide credit: Rick Szeliski

Why Bayesian estimation?

Principled way of determining cost function

Explicit model of noise and prior knowledge

Admits a wide variety of optimization algorithms:

- gradient descent (local minimization)
- stochastic optimization (Gibbs Sampler)
- mean-field optimization

Slide credit: Rick Szeliski

Why Bayesian estimation?

Principled way of determining cost function

Explicit model of noise and prior knowledge

Admits a wide variety of optimization algorithms:

- gradient descent (local minimization)
- stochastic optimization (Gibbs Sampler)
- mean-field optimization
- graph theoretic (actually deterministic) [Zabih]

Slide credit: Rick Szeliski

Why Bayesian estimation?

Principled way of determining cost function

Explicit model of noise and prior knowledge

Admits a wide variety of optimization algorithms:

- gradient descent (local minimization)
- stochastic optimization (Gibbs Sampler)
- mean-field optimization
- graph theoretic (actually deterministic) [Zabih]
- [loopy] belief propagation

Slide credit: Rick Szeliski

Why Bayesian estimation?

Principled way of determining cost function

Explicit model of noise and prior knowledge

Admits a wide variety of optimization algorithms:

- gradient descent (local minimization)
- stochastic optimization (Gibbs Sampler)
- mean-field optimization
- graph theoretic (actually deterministic) [Zabih]
- [loopy] belief propagation
- large stochastic flips [Swendsen-Wang]

Slide credit: Rick Szeliski

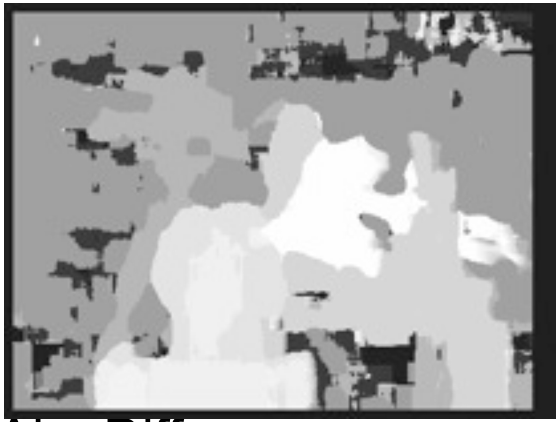
Depth Map Results

Input image   Sum Abs Diff

Mean field   Graph cuts

Slide credit: Rick Szeliski

Depth Map Results

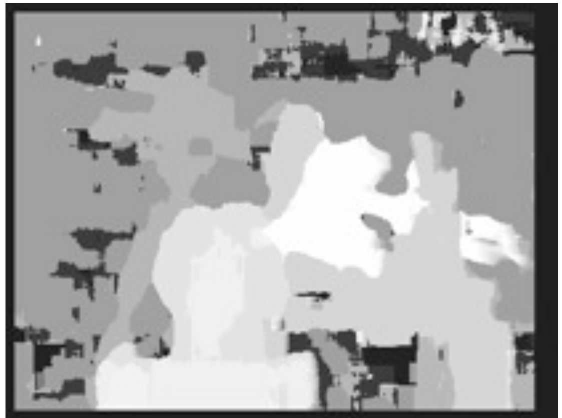


Input image Sum Abs Diff

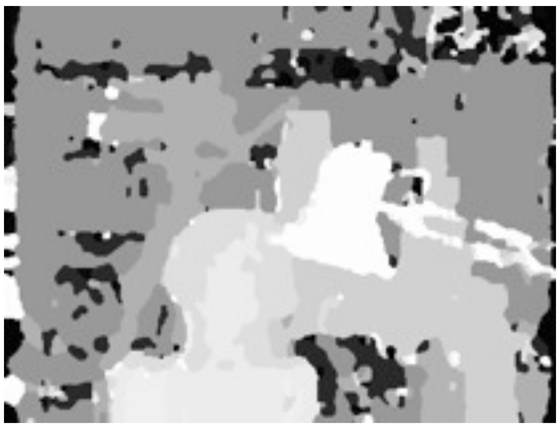
Mean field Graph cuts

Slide credit: Rick Szeliski

Depth Map Results



Input image   Sum Abs Diff

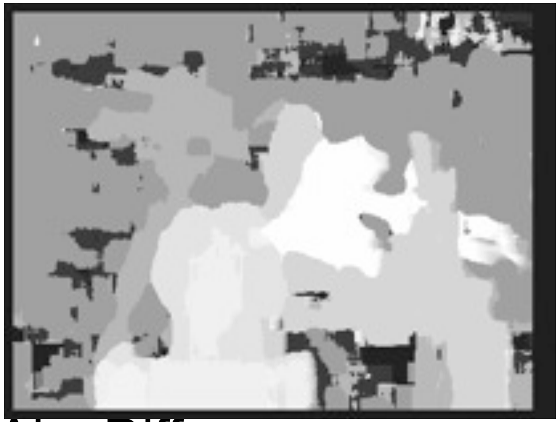


Mean field   Graph cuts

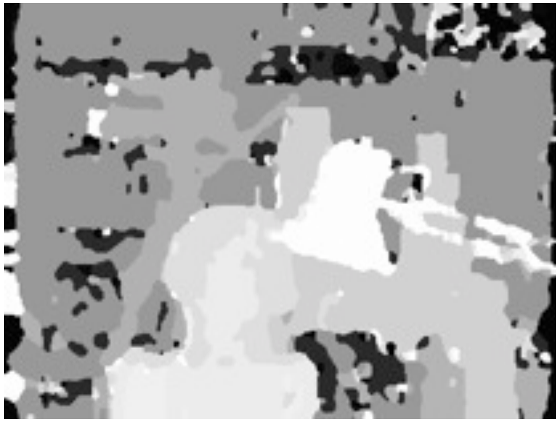
Slide credit: Rick Szeliski

Depth Map Results

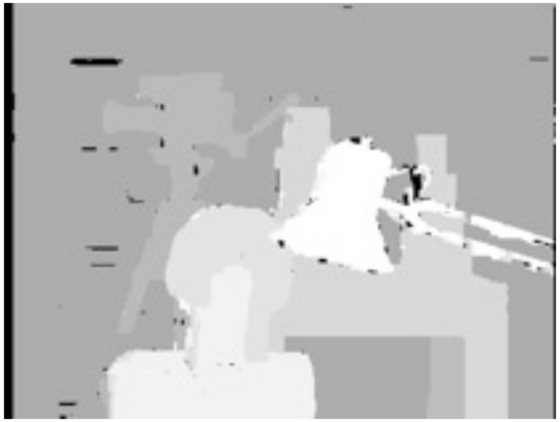
Input image



Sum Abs Diff



Mean field



Graph cuts

Slide credit: Rick Szeliski

Stereo evaluation

vision.middlebury.edu

stereo • mview • MRF • flow

Stereo Evaluation • Datasets • Code • Submit

Daniel Scharstein • Richard Szeliski

Welcome to the Middlebury Stereo Vision Page, formerly located at www.middlebury.edu/stereo. This website accompanies our taxonomy and comparison of two-frame stereo correspondence algorithms [1]. It contains:

- An on-line evaluation of current algorithms
- Many stereo datasets with ground-truth disparities
- Our stereo correspondence software
- An on-line submission script that allows you to evaluate your stereo algorithm in our framework

How to cite the materials on this website:
We grant permission to use and publish all images and numerical results on this website. If you report performance results, we request that you cite our paper [1]. Instructions on how to cite our datasets are listed on the datasets page. If you want to cite this website, please use the URL "**vision.middlebury.edu/stereo/**".

References:

[1] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1/2/3):7-42, April-June 2002. Microsoft Research Technical Report MSR-TR-2001-81, November 2001.

Slide credit: Rick Szeliski

Stereo—best algorithms

Error Threshold = 1

Error Threshold...

Sort by nonocc | Sort by all | Sort by disc

Algorithm	Avg. Rank	Tsukuba (ground truth) nonocc	Tsukuba all	Tsukuba disc	Venus (ground truth) nonocc	Venus all	Venus disc	Teddy (ground truth) nonocc	Teddy all	Teddy disc	Cones (ground truth) nonocc	Cones all	Cones disc
AdaptingBP [17]	2.8	1.11 6	1.37 3	5.79 7	**0.10** 1	0.21 2	**1.44** 1	4.22 4	7.06 2	11.8 4	**2.48** 1	7.92 2	**7.32** 1
DoubleBP2 [35]	2.9	**0.88** 1	**1.29** 1	**4.76** 1	0.13 3	0.45 5	1.87 5	3.53 2	8.30 3	**9.63** 1	2.90 3	8.78 8	7.79 2
DoubleBP [15]	4.9	0.88 2	1.29 2	4.76 2	0.14 5	0.60 13	2.00 7	3.55 3	8.71 5	9.70 2	2.90 4	9.24 11	7.80 3
SubPixDoubleBP [30]	5.6	1.24 10	1.76 13	5.98 8	0.12 2	0.46 6	1.74 4	**3.45** 1	8.38 4	10.0 3	2.93 5	8.73 7	7.91 4
AdaptOvrSegBP [33]	9.9	1.69 22	2.04 21	5.64 6	0.14 4	**0.20** 1	1.47 2	7.04 14	11.1 7	16.4 11	3.60 11	8.96 10	8.84 10
SymBP+occ [7]	10.8	0.97 4	1.75 12	5.09 4	0.16 6	0.33 3	2.19 8	6.47 8	10.7 6	17.0 14	4.79 24	10.7 21	10.9 20
PlaneFitBP [32]	10.8	0.97 5	1.83 14	5.26 5	0.17 7	0.51 8	1.71 3	6.65 9	12.1 13	14.7 7	4.17 20	10.7 20	10.6 19
AdaptDispCalib [36]	11.8	1.19 8	1.42 4	6.15 9	0.23 9	0.34 4	2.50 11	7.80 19	13.6 21	17.3 17	3.62 12	9.33 12	9.72 15
Segm+visib [4]	12.2	1.30 15	1.57 5	6.92 18	0.79 21	1.06 18	6.76 22	5.00 5	**6.54** 1	12.3 5	3.72 13	8.62 6	10.2 17
C-SemiGlob [19]	12.3	2.61 29	3.29 24	9.89 27	0.25 12	0.57 10	3.24 15	5.14 6	11.8 8	13.0 6	2.77 2	8.35 4	8.20 5
SO+borders [29]	12.8	1.29 14	1.71 9	6.83 15	0.25 13	0.53 9	2.26 9	7.02 13	12.2 14	16.3 9	3.90 15	9.85 16	10.2 18
DistinctSM [27]	14.1	1.21 9	1.75 11	6.39 11	0.35 14	0.69 16	2.63 13	7.45 18	13.0 17	18.1 19	3.91 16	9.91 18	8.32 7
CostAggr+occ [39]	14.3	1.38 17	1.96 17	7.14 19	0.44 16	1.13 19	4.87 19	6.80 11	11.9 10	17.3 16	3.60 10	8.57 5	9.36 13

S

[12] K.-J. Yoon and I.-S. Kweon. Adaptive support-weight approach for correspondence search. PAMI 28(4):650-656, 2006.

[13] M. Gong and Y.-H. Yang. Near real-time reliable stereo matching using programmable graphics hardware. CVPR 2005.

[14] L. Wang, M. Liao, M. Gong, R. Yang, and D. Nistér. High-quality real-time stereo using adaptive cost aggregation and dynamic programming. 3DPVT 2006.

[15] Q. Yáng, L. Wang, R. Yang, H. Stewénius, and D. Nistér. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. CVPR 2006.

[16] H. Audirac, A. Beloiarov, F. Núñez, and J. Villegas. Dense disparity map based on STICA algorithm. Expo Forestal, Mexico, 2005.

[17] A. Klaus, M. Sormann and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. ICPR 2006.

[18] C. Lei, J. Selzer, and Y. Yang. Region-tree based stereo using dynamic programming optimization. CVPR 2006.

[19] H. Hirschmüller. Stereo vision in structured environments by consistent semi-global matching. CVPR 2006, PAMI 30 (2):328-341, 2008.

[20] C. Strecha, R. Fransens, and L. Van Gool. Combined depth and outlier estimation in multi-view stereo. CVPR 2006.

[21] Q. Yang, L. Wang, R. Yang, S. Wang, M. Liao, and D. Nistér. Real-time global stereo matching using hierarchical belief propagation. BMVC 2006.

[22] Y. Deng and X. Lin. A fast line segment based dense stereo algorithm using tree dynamic programming. ECCV 2006.

[23] S. El-Etriby, A. Al-Hamadi, and B. Michaelis. Dense depth map reconstruction by phase difference-based algorithm under influence of perspective distortion. ICCVG 2006 / J. Machine Graphics and Vision.

[24] S. Larsen, P. Mordohai, M. Pollefeys, and H. Fuchs. Temporally consistent reconstruction from multiple video streams using enhanced belief propagation. ICCV 2007.

[25] S. Gehrig and U. Franke. Improving sub-pixel accuracy for long range stereo. ICCV VRML workshop 2007.

[26] L. Zitnick and S.B. Kang. Stereo for image-based rendering using image over-segmentation. IJCV 2007.

[27] K.-J. Yoon and I. S. Kweon. Stereo matching with the distinctive similarity measure. ICCV 2007.

[28] F. Tombari, S. Mattoccia, and L. Di Stefano. Segmentation-based adaptive support for accurate stereo correspondence. PSIVT 2007.

[29] S. Mattoccia, F. Tombari, and L. Di Stefano. Stereo vision enabling precise border localization within a scanline optimization framework. ACCV 2007.

[30] Q. Yang, R. Yang, J. Davis, and D. Nistér. Spatial-depth super resolution for range images. CVPR 2007.

[31] S. El-Etriby, A. Al-Hamadi, and B. Michaelis. Dense stereo correspondence with slanted surface using phase-based algorithm. IEEE ISIE 2007.

[32] Anonymous. Near real-time stereo for weakly-textured scenes. Submitted to CVPR 2008.

[33] Y. Taguchi, B. Wilburn, and L. Zitnick. Stereo reconstruction with mixed pixels using adaptive over-segmentation. CVPR 2008.

[34] A. Bhusnurmath and C.J. Taylor. Stereo matching using interior point methods. 3DPVT 2008.

[35] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. Submitted to PAMI.

[36] Z.Gu, X.Su, Y.Liu, and Q.Zhang. Local stereo matching with adaptive support-weight, rank transform and disparity calibration. Pattern Recognition Letters, 2008.

CSE

Stereo Topics

- Special, simple system, main idea
- More general camera conditions, epipolar constraints
  - epipolar geometry
  - epipolar algebra
- Image rectification
- Stereo matching (likelihood term)
- Stereo regularization (prior term)
- Inference
  - dynamic programming
  - graph cuts
- **Structured light**

Active stereo with structured light

Li Zhang's one-shot stereo

Project "structured" light patterns onto the object

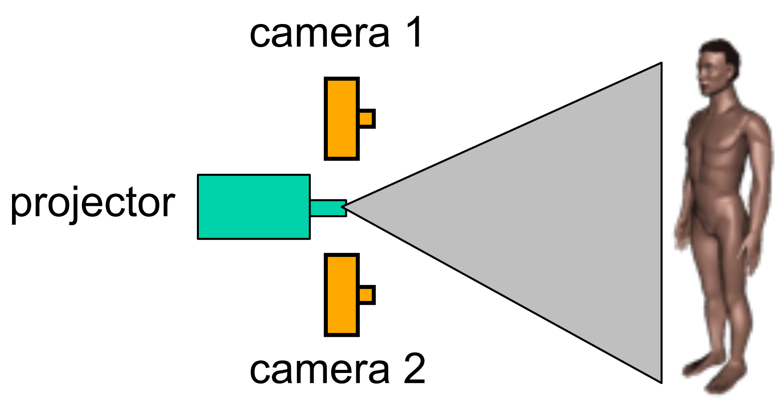
- simplifies the correspondence problem

Li Zhang, Brian Curless, and Steven M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. In *Proceedings of the 1st International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*, Padova, Italy, June 19-21, 2002, pp. 24-36.

Slide credit: Rick Szeliski

Active stereo with structured light

Li Zhang's one-shot stereo



Project "structured" light patterns onto the object

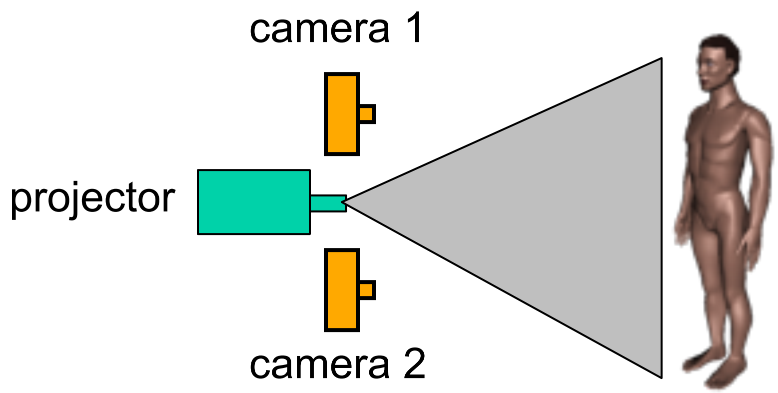
- simplifies the correspondence problem

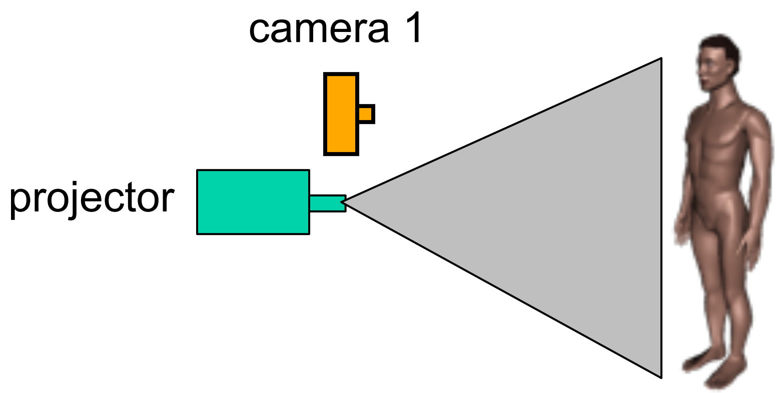
Li Zhang, Brian Curless, and Steven M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. In *Proceedings of the 1st International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*, Padova, Italy, June 19-21, 2002, pp. 24-36.

Slide credit: Rick Szeliski

Active stereo with structured light

Li Zhang's one-shot stereo



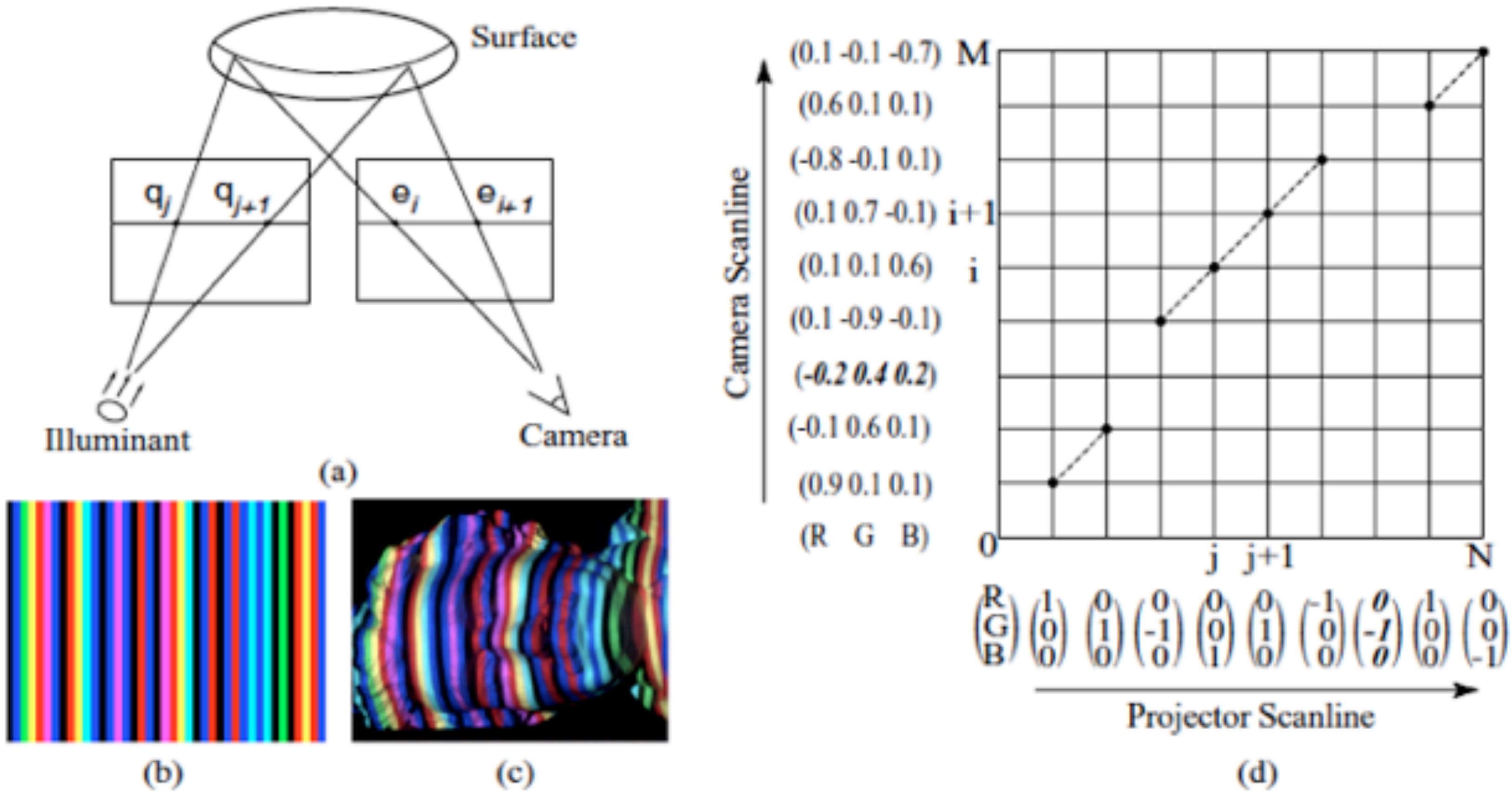


Project "structured" light patterns onto the object

- simplifies the correspondence problem

Li Zhang, Brian Curless, and Steven M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. In *Proceedings of the 1st International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*, Padova, Italy, June 19-21, 2002, pp. 24-36.

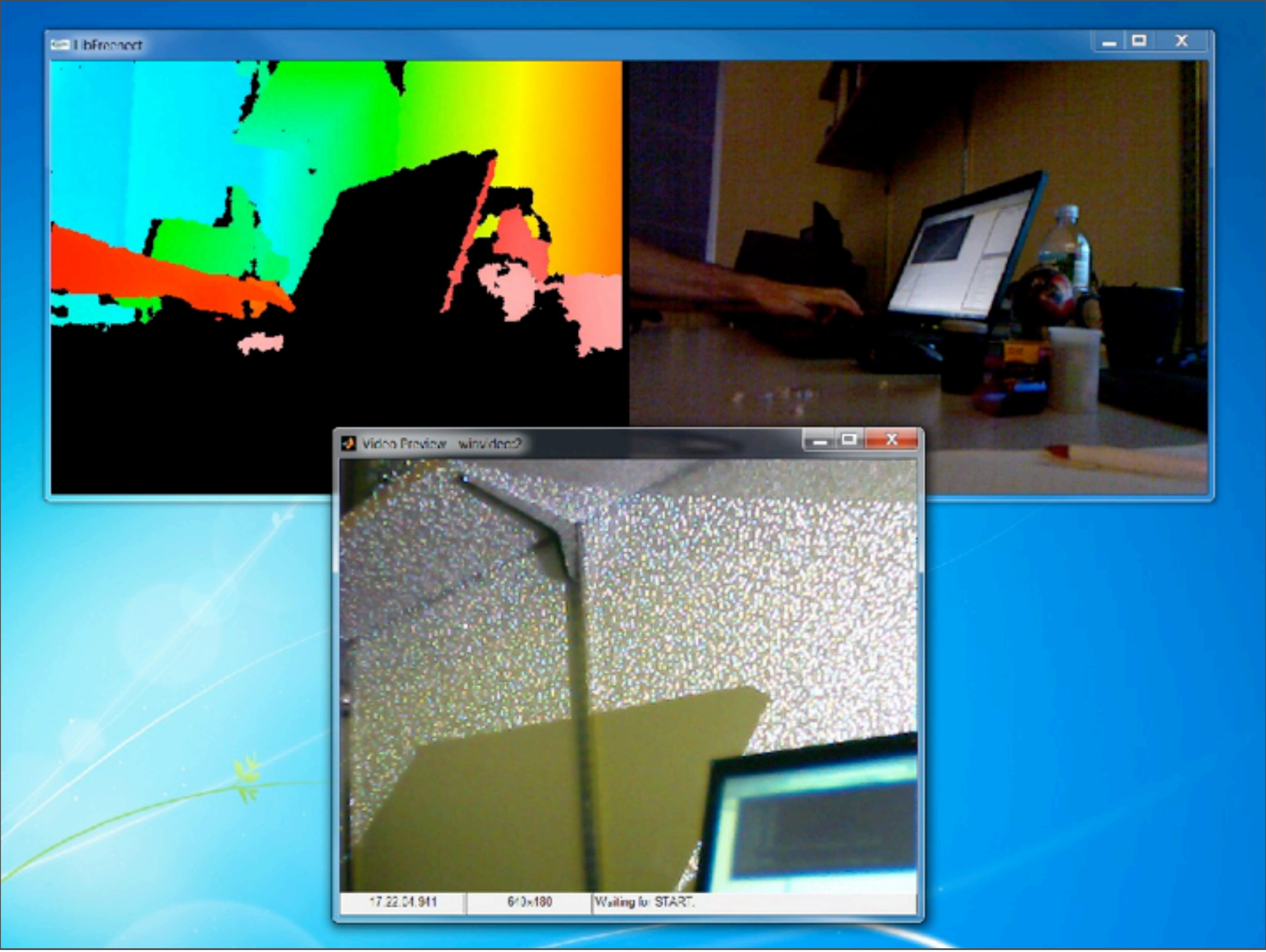
Slide credit: Rick Szeliski

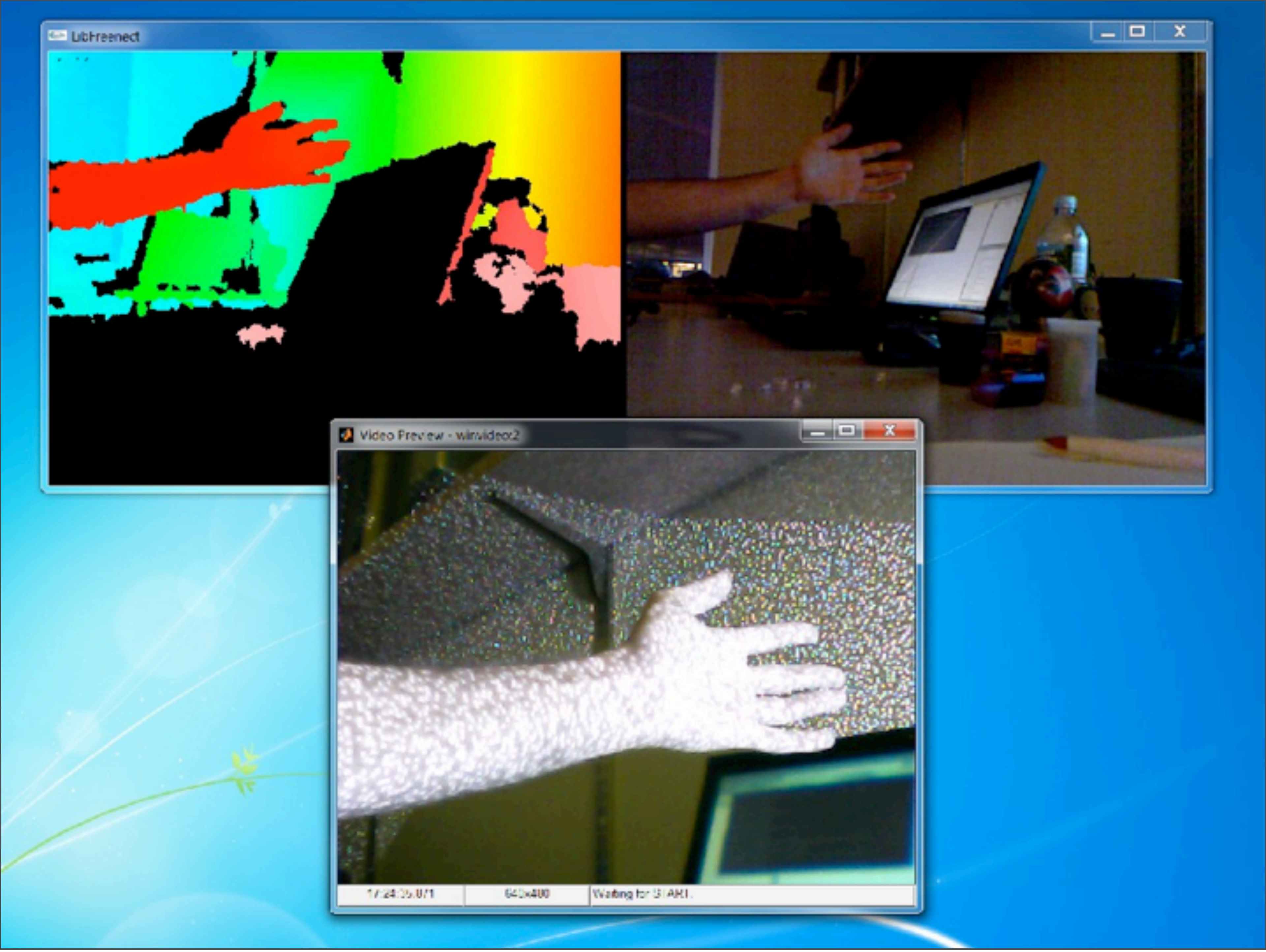


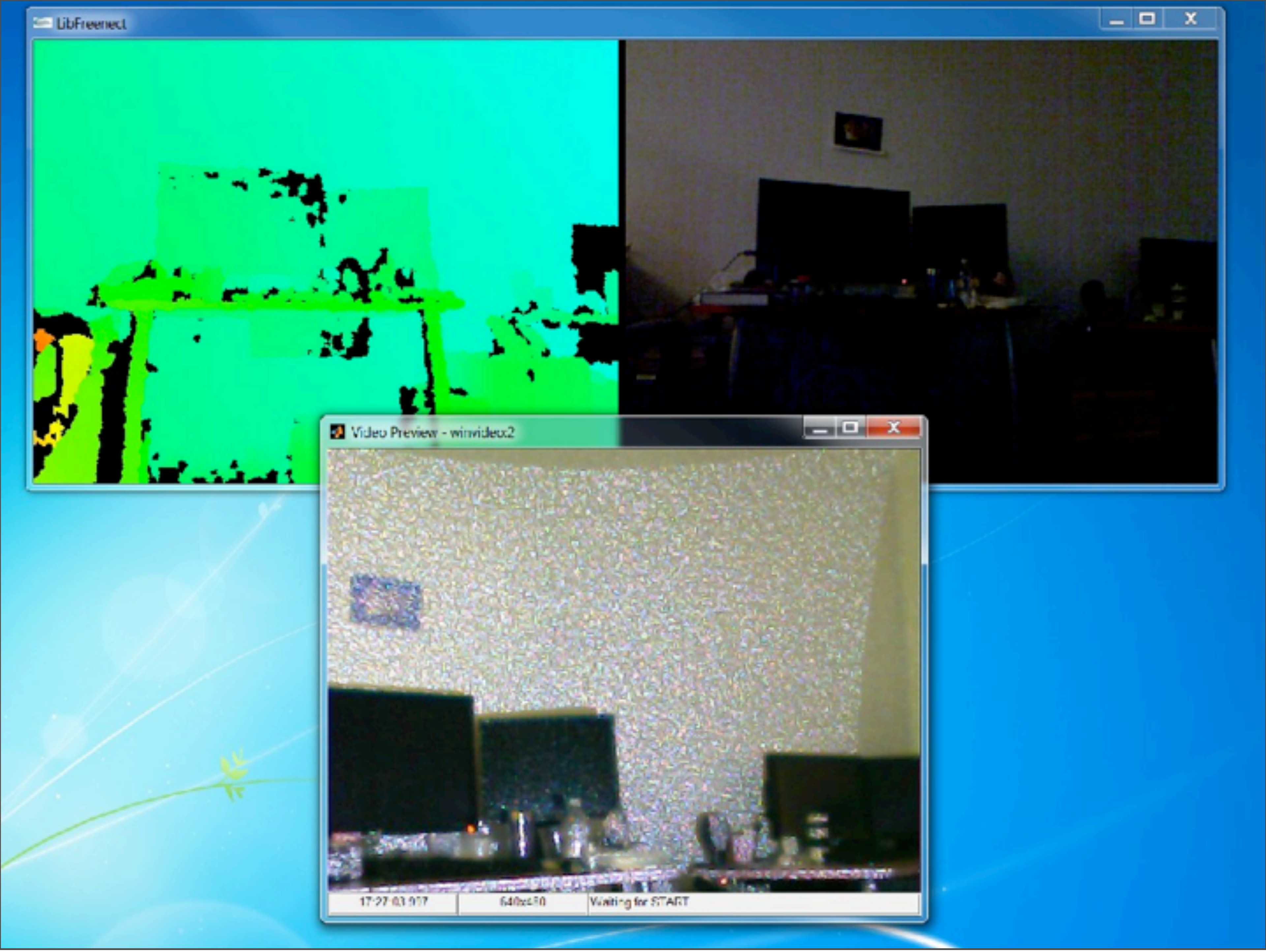
igure 2. Summary of the one-shot method. (a) In optical triangulation, an illumination pattern is projected onto an object and the eflected light is captured by a camera. The 3D point is reconstructed from the relative displacement of a point in the pattern and mage. If the image planes are rectified as shown, the displacement is purely horizontal (one-dimensional). (b) An example of he projected stripe pattern and (c) an image captured by the camera. (d) The grid used for multi-hypothesis code matching. The orizontal axis represents the projected color transition sequence and the vertical axis represents the detected edge sequence, oth taken for one projector and rectified camera scanline pair. A match represents a path from left to right in the grid. Each ertex (j, i) has a score, measuring the consistency of the correspondence between e_i, the color gradient vectors shown by the ertical axis, and q_j, the color transition vectors shown below the horizontal axis. The score for the entire match is the summation f scores along its path. We use dynamic programming to find the optimal path. In the illustration, the camera edge in bold italics orresponds to a false detection, and the projector edge in bold italics is missed due to, e.g., occlusion.

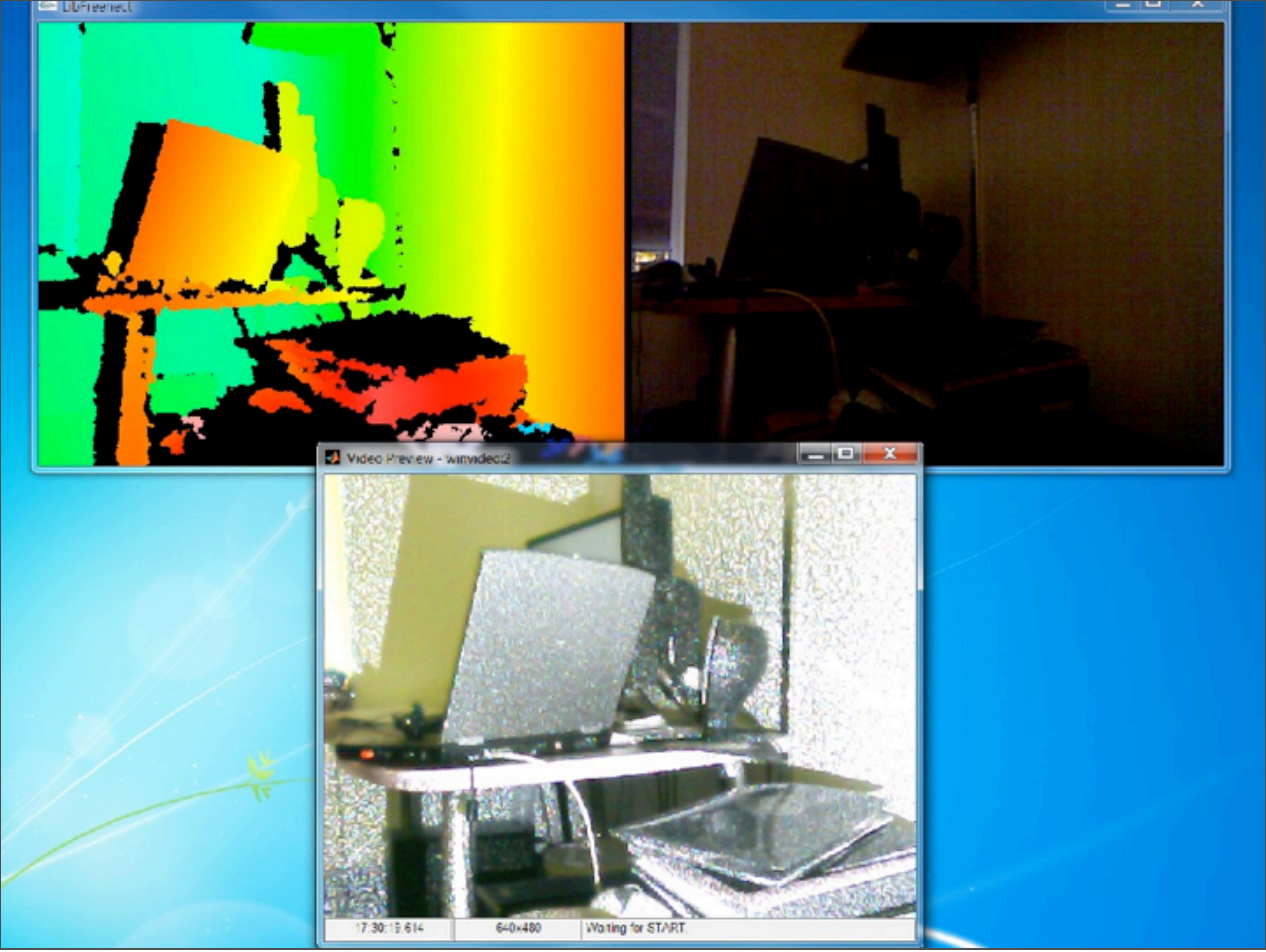
Li Zhang, Brian Curless, and Steven M. Seitz


Megapixel
10X Digital Zoom
f=3.85mm

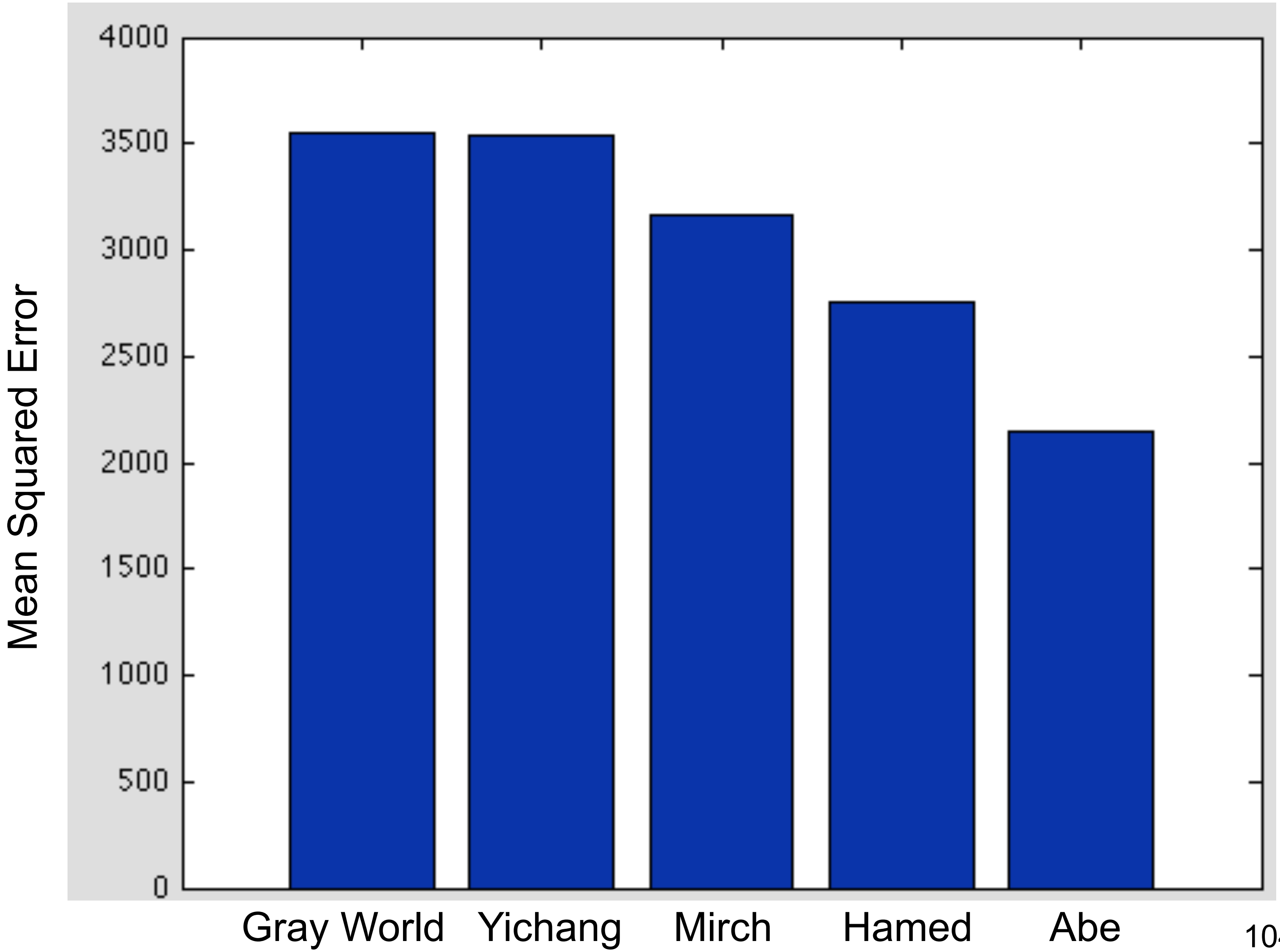

Video Preview - winvideo2
Waiting for START.


LibFreenect
Video Preview - winvideo2


LibFreenect
Video Preview - winvideo2
Waiting for START


LibFreenect
Video Preview - winvideo:2
17:30:19.614
640x480
Waiting for START.

OSA Illumination Spectrum Estimation Contest Leaders, March 13, 2011



Bibliography

D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision, 47(1):7-42, May 2002.

R. Szeliski. Stereo algorithms and representations for image-based rendering. In British Machine Vision Conference (BMVC'99), volume 2, pages 314-328, Nottingham, England, September 1999.

G. M. Nielson, Scattered Data Modeling, IEEE Computer Graphics and Applications, 13(1), January 1993, pp. 60-70.

S. B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In CVPR'2001, vol. I, pages 103-110, December 2001.

Y. Boykov, O. Veksler, and Ramin Zabih, Fast Approximate Energy Minimization via Graph Cuts, Unpublished manuscript, 2000.

A.F. Bobick and S.S. Intille. Large occlusion stereo. International Journal of Computer Vision, 33(3), September 1999. pp. 181-200

D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. International Journal of Computer Vision, 28(2):155-174, July 1998

Slide credit: Rick Szeliski

Bibliography

Volume Intersection

- Martin & Aggarwal, “Volumetric description of objects from multiple views”, Trans. Pattern Analysis and Machine Intelligence, 5(2), 1991, pp. 150-158.
- Szeliski, “Rapid Octree Construction from Image Sequences”, Computer Vision, Graphics, and Image Processing: Image Understanding, 58(1), 1993, pp. 23-32.

Voxel Coloring and Space Carving

- Seitz & Dyer, “Photorealistic Scene Reconstruction by Voxel Coloring”, Proc. Computer Vision and Pattern Recognition (CVPR), 1997, pp. 1067-1073.
- Seitz & Kutulakos, “Plenoptic Image Editing”, Proc. Int. Conf. on Computer Vision (ICCV), 1998, pp. 17-24.
- Kutulakos & Seitz, “A Theory of Shape by Space Carving”, Proc. ICCV, 1998, pp. 307-314.

Slide credit: Rick Szeliski

Bibliography

Related References

- Bolles, Baker, and Marimont, “Epipolar-Plane Image Analysis: An Approach to Determining Structure from Motion”, International Journal of Computer Vision, vol 1, no 1, 1987, pp. 7-55.
- Faugeras & Keriven, “Variational principles, surface evolution, PDE's, level set methods and the stereo problem", IEEE Trans. on Image Processing, 7(3), 1998, pp. 336-344.
- Szeliski & Golland, “Stereo Matching with Transparency and Matting”, Proc. Int. Conf. on Computer Vision (ICCV), 1998, 517-524.
- Roy & Cox, “A Maximum-Flow Formulation of the N-camera Stereo Correspondence Problem”, Proc. ICCV, 1998, pp. 492-499.
- Fua & Leclerc, “Object-centered surface reconstruction: Combining multi-image stereo and shading", International Journal of Computer Vision, 16, 1995, pp. 35-56.
- Narayanan, Rander, & Kanade, “Constructing Virtual Worlds Using Dense Stereo”, Proc. ICCV, 1998, pp. 3-10.

Slide credit: Rick Szeliski