MIT COMPUTER VISION

MIT CSAIL

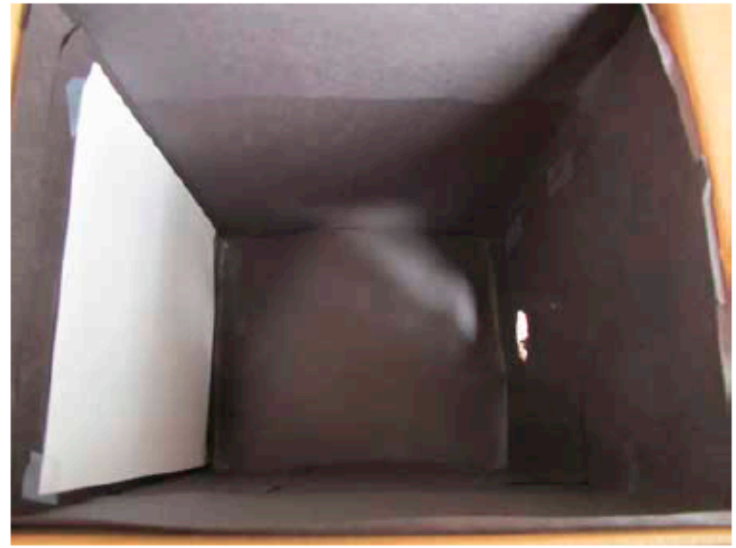
6.869: Advances in Computer Vision

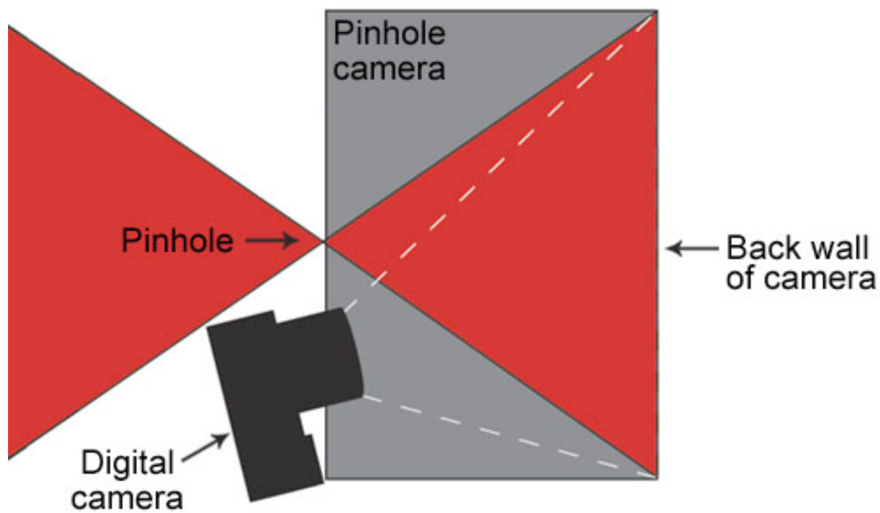
Antonio Torralba, 2012

Lecture 10

Image formation

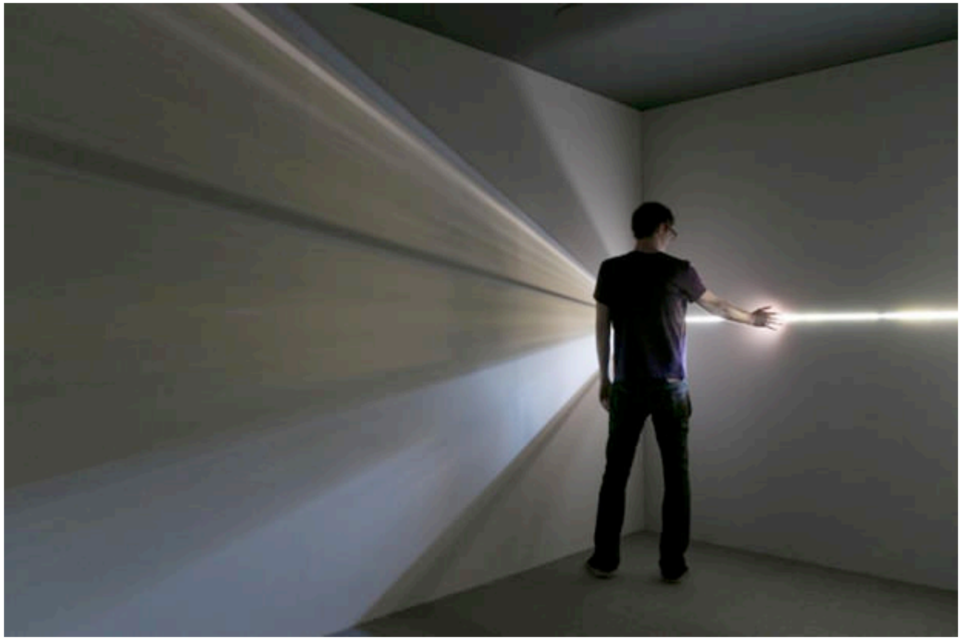
Problem Set 1





http://www.foundphotography.com/PhotoThoughts/archives/2005/04/pinhole_camera_2.html

Source: wikipedia

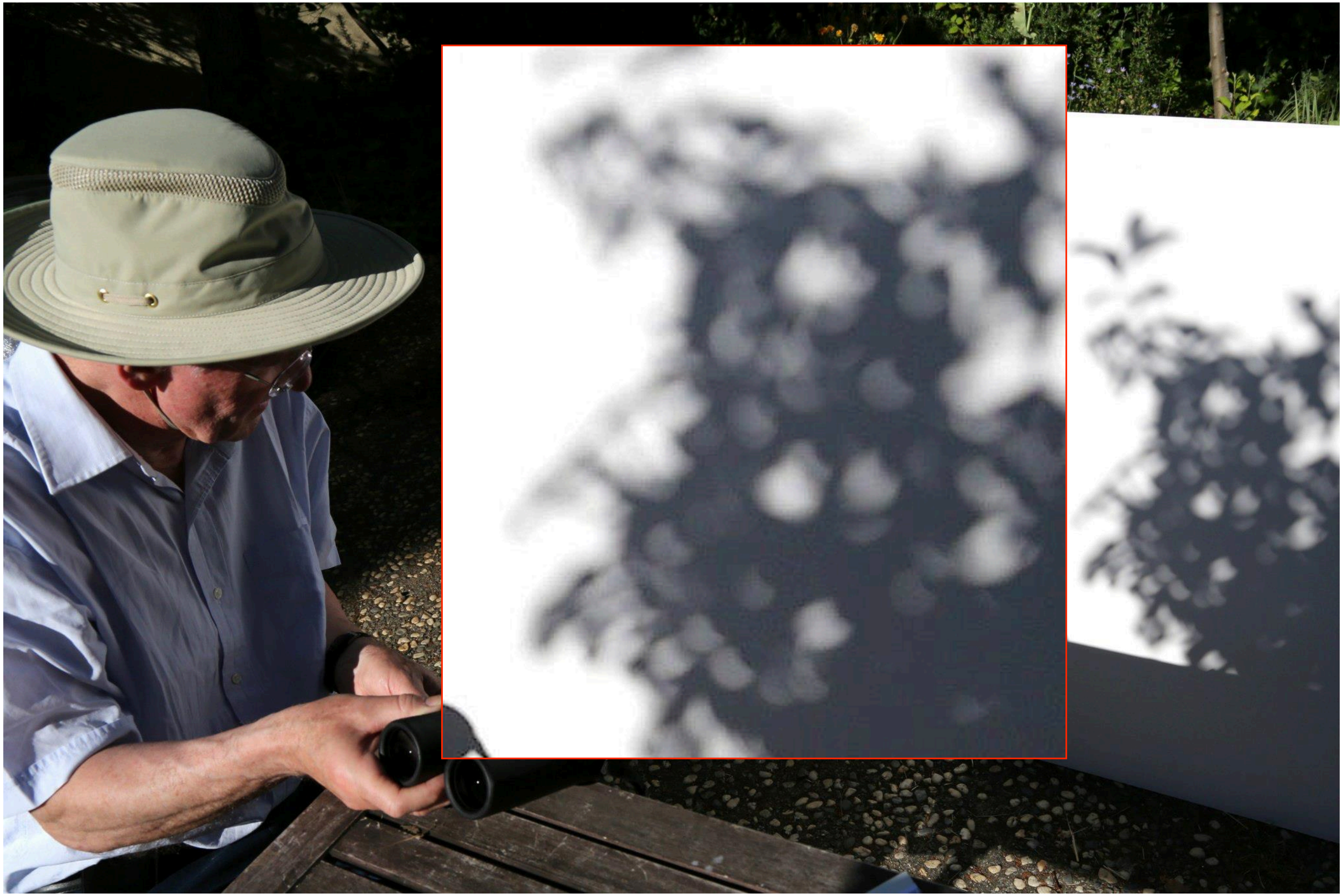


Chris Fraser

”a camera obscura has been used ... to bring images from the outside into a darkened room”

Aberlado Morell

Accidental pinholes in outdoor scenes



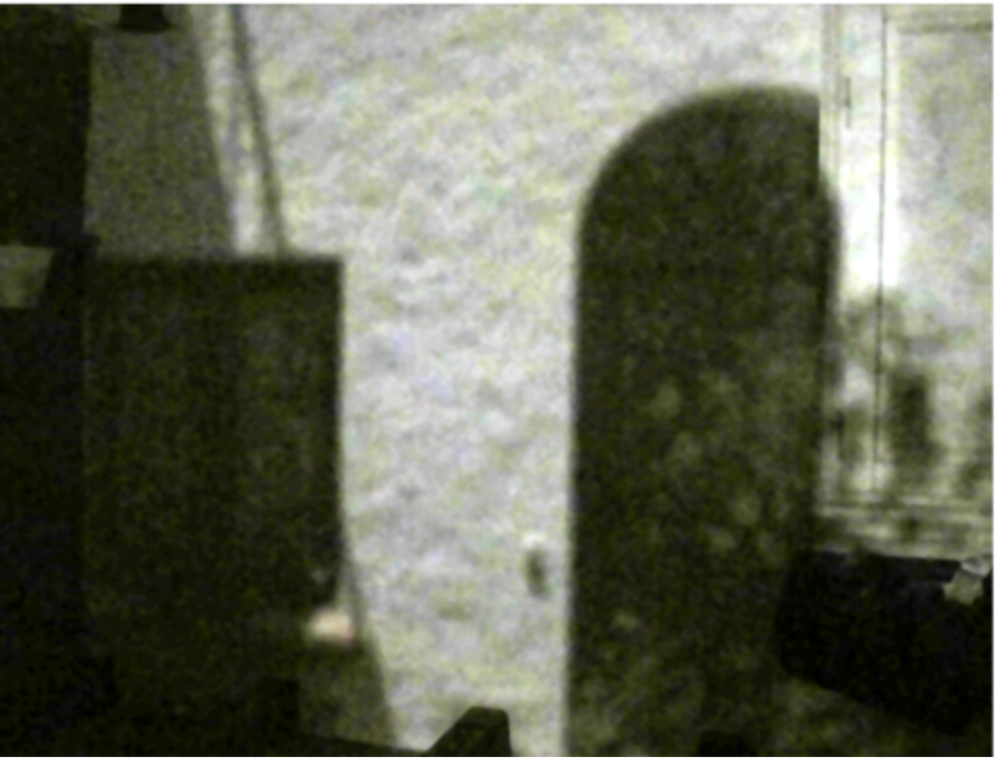
Pierre Moreels father (source: facebook)


Shadows?


SALA
CHANCA

Accidental pinhole camera

Window turned into a pinhole



View outside

Making a pinhole with home materials

Window open

Window turned into a pinhole

Making a pinhole with home materials

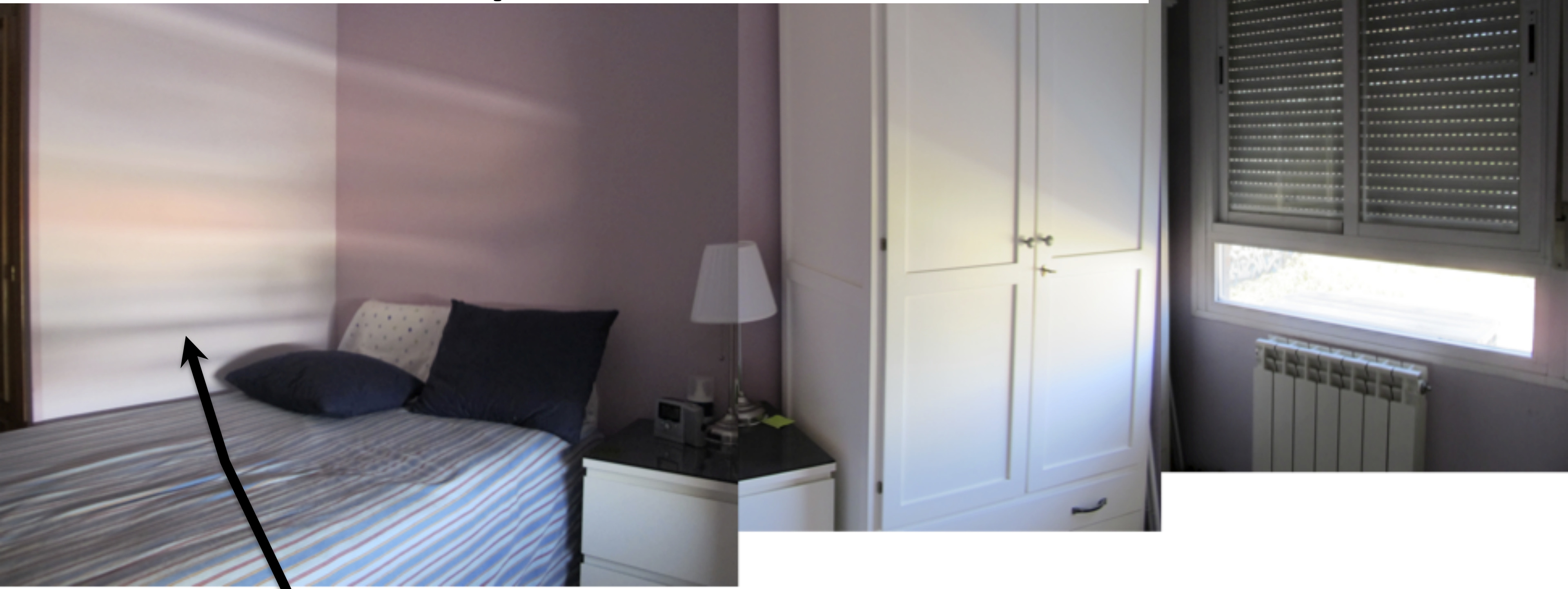
An hotel room,
contrast enhanced.

The view from my window

Accidental pinholes produce images that are unnoticed or misinterpreted as shadows

Another hotel room

Accidental pinhole camera



Outside scene

*

Aperture

See Zomet, A.; Nayar, S.K. CVPR 2006 for a detailed analysis.

Visualizing the convolution

Anti-pinhole or Pinspeck cameras

Adam L. Cohen, 1982

OPTICA ACTA, 1982, VOL. 29, NO. 1, 63–67

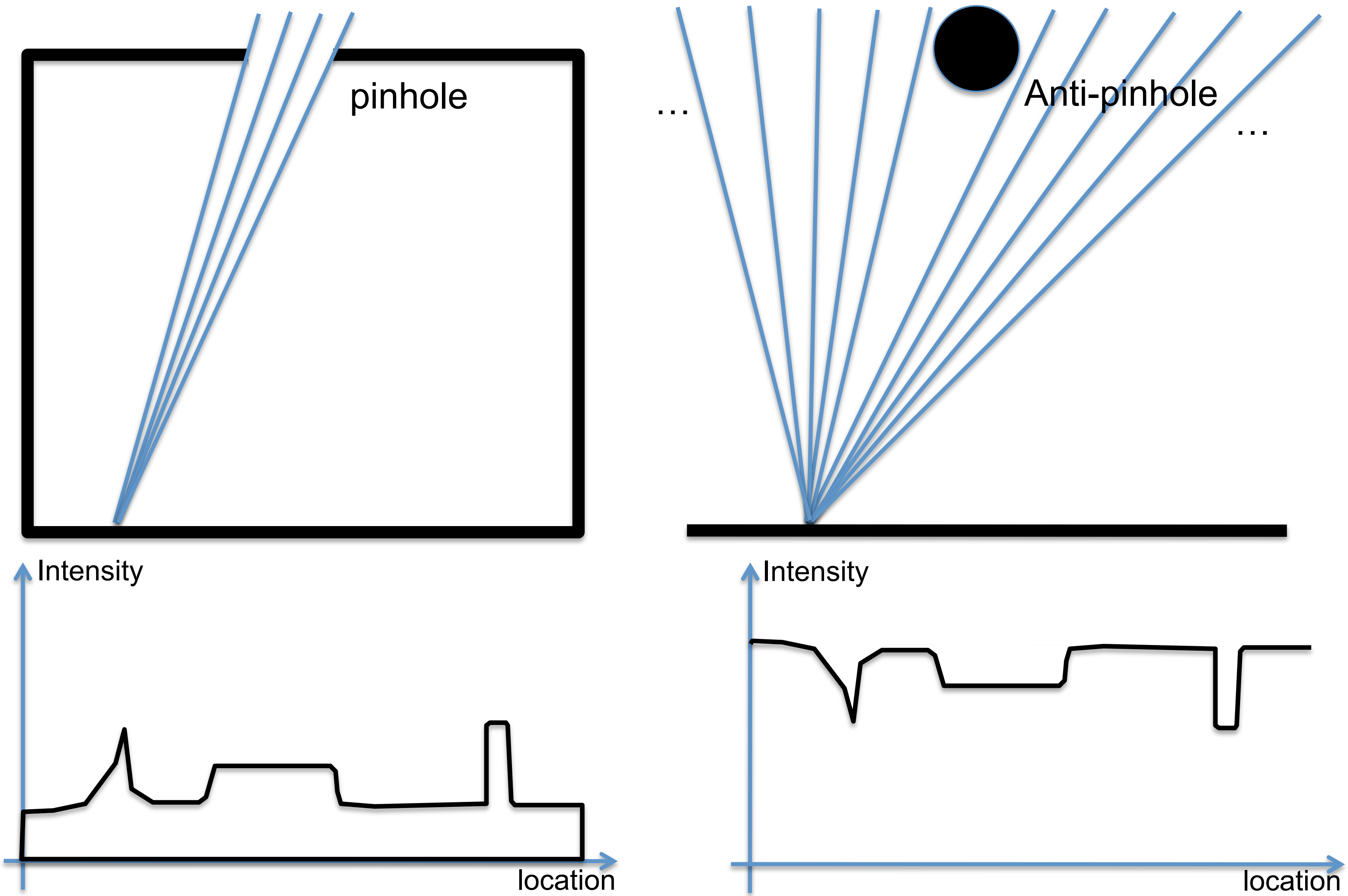
Anti-pinhole imaging

ADAM LLOYD COHEN
Parmly Research Institute, Loyola University of Chicago,
Chicago, Illinois 60626, U.S.A.

(*Received 16 April 1981; revision received 8 July 1981*)

Abstract. By complementing a pinhole to produce an isolated opaque spot, the light ordinarily blocked from the pinhole image is transmitted, and the light ordinarily transmitted is blocked. A negative geometrical image is formed, distinct from the familiar 'bright-spot' diffraction image. Anti-pinhole, or 'pinspeck' images are visible during a solar eclipse, when the shadows of objects appear crescent-shaped. Pinspecks demonstrate unlimited depth of field, freedom from distortion and large angular field. Images of different magnification may be formed simultaneously. Contrast is poor, but is improvable by averaging to remove noise and subtraction of a d.c. bias. Pinspecks may have application in X-ray space optics, and might be employed in the eyes of simple organisms.

Pinhole and Anti-pinhole cameras



Adam L. Cohen, 1982

Natural eyes

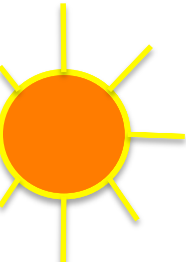
Lenses

Pinholes

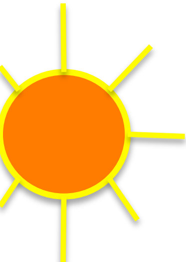
Anti-pinholes

Shadows
Accidental anti-pinhole cameras?

Shadows
Accidental anti-pinhole cameras



Shadows
Accidental anti-pinhole cameras



Background image

-

Input video

=

Negative of the shadow

Background image

-

Input video

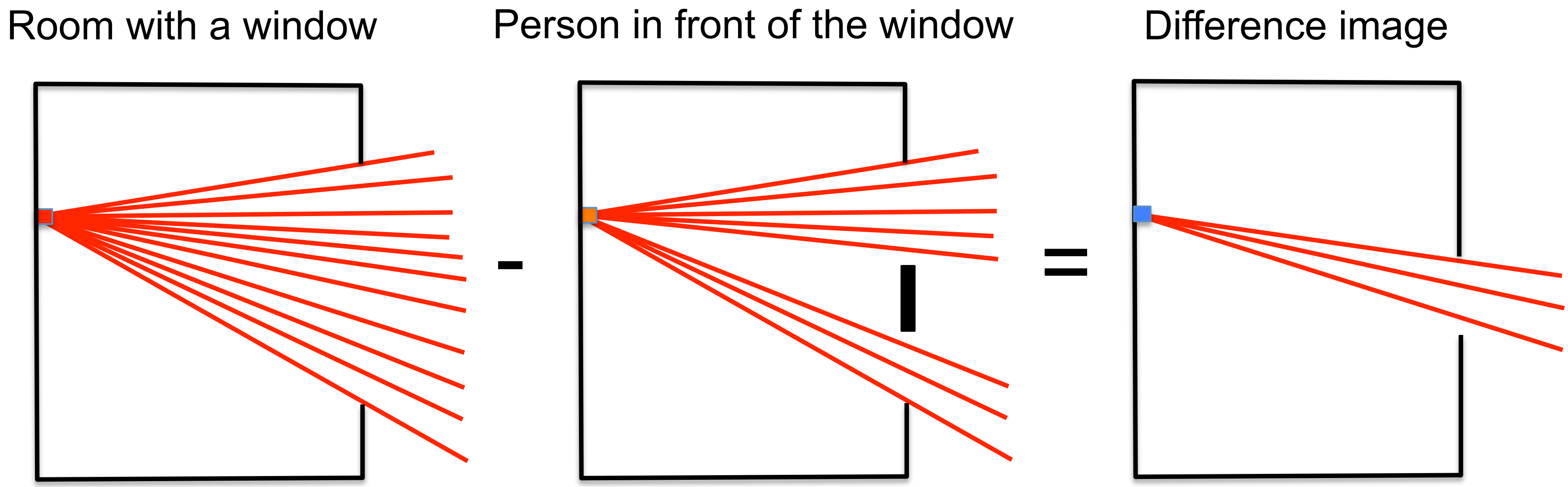
=

Negative of the shadow

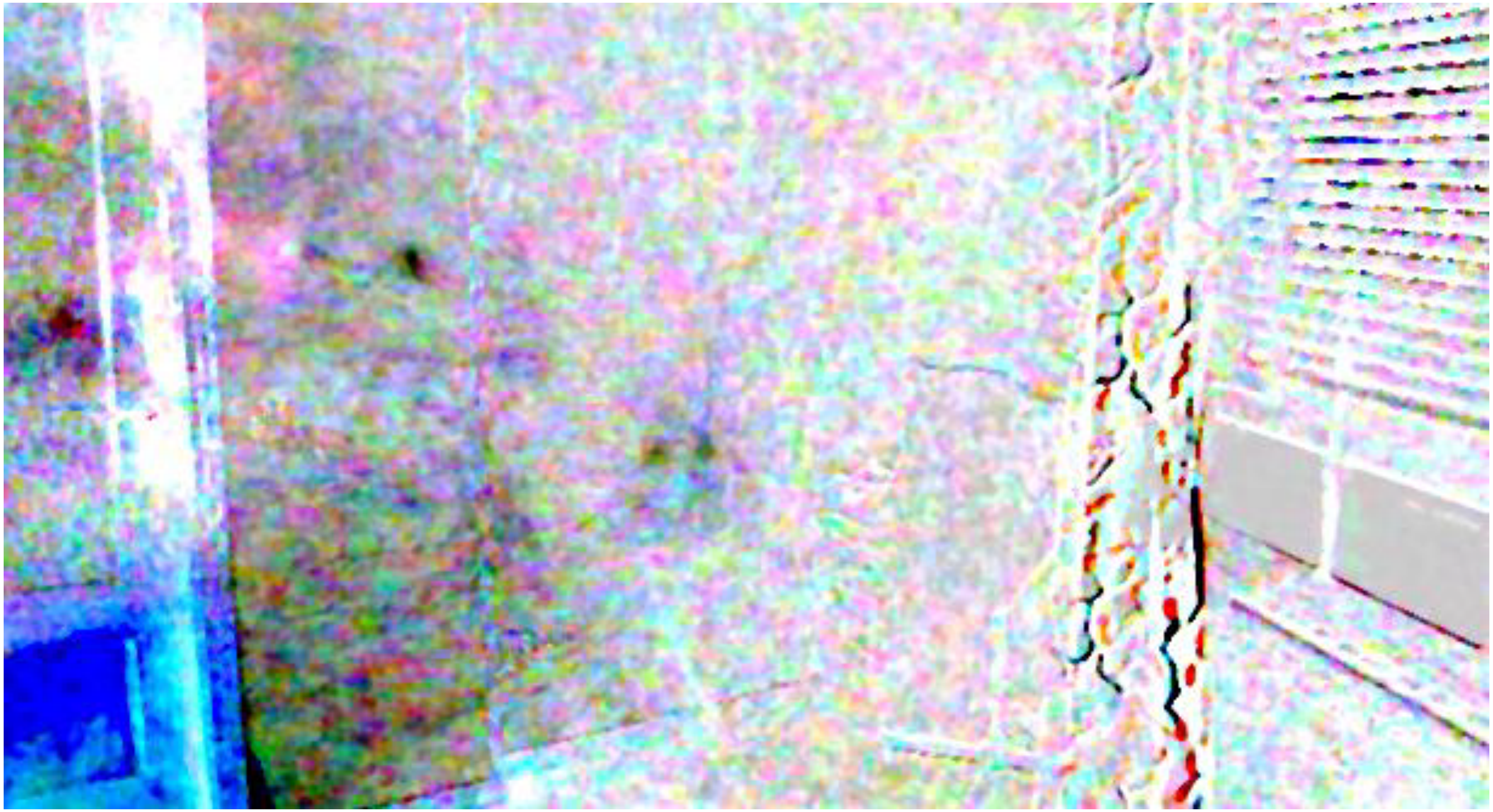
Mixed accidental pinhole and anti-pinhole cameras

Mixed accidental pinhole and anti-pinhole cameras

Mixed accidental pinhole and anti-pinhole cameras



Mixed accidental pinhole and anti-pinhole cameras

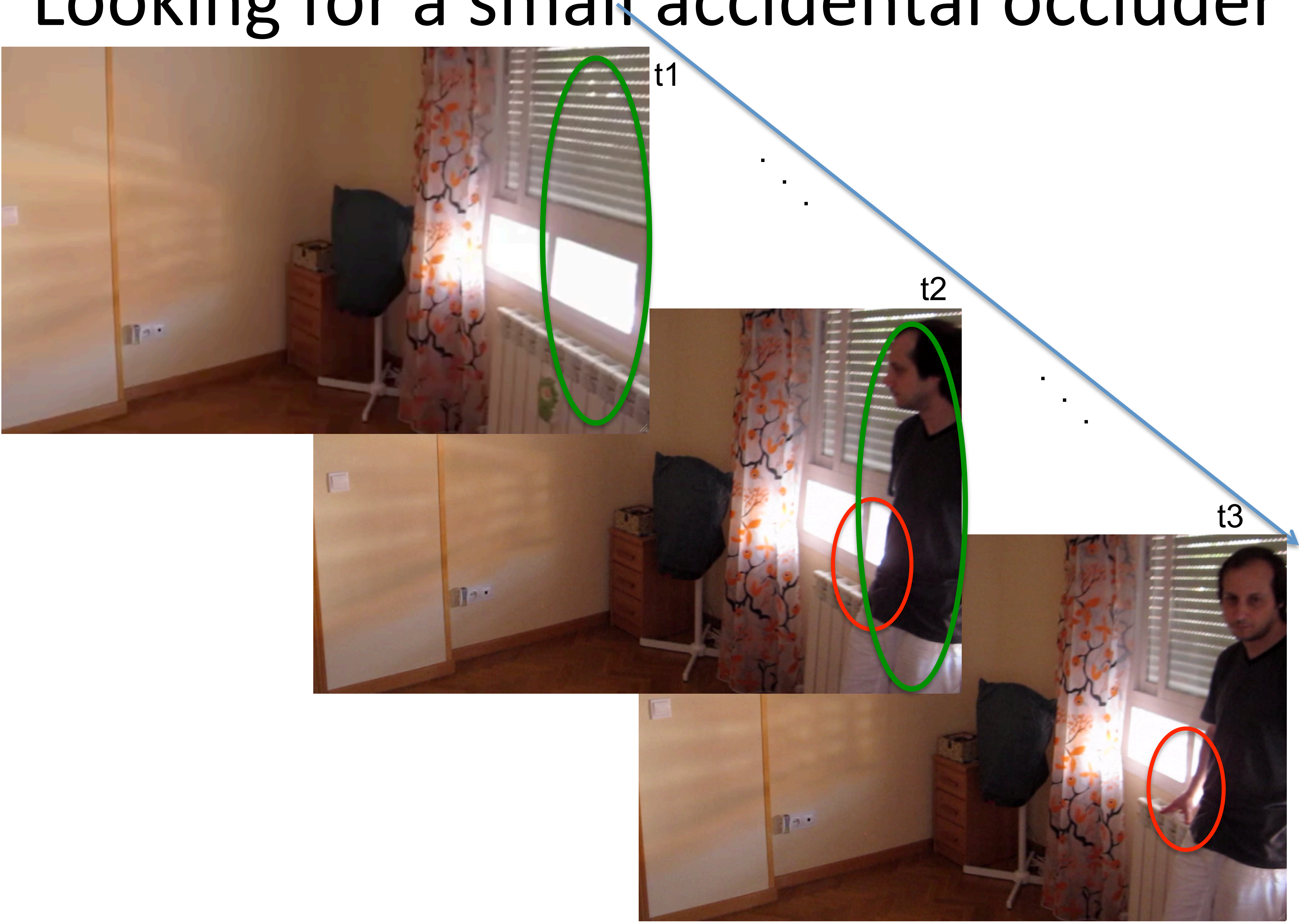


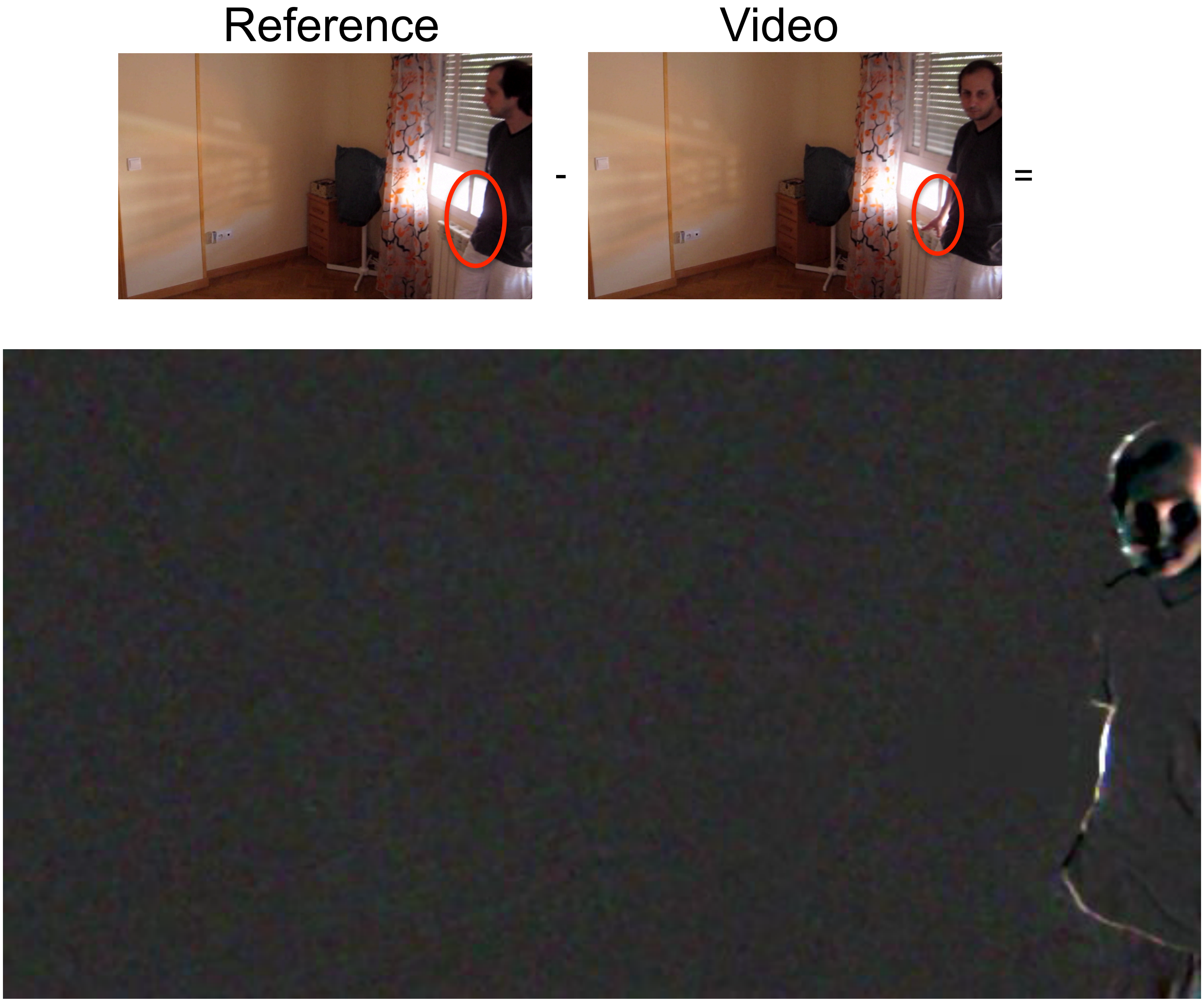
Mixed accidental pinhole and anti-pinhole cameras

Body as the occluder

View outside the window

Looking for a small accidental occluder




Reference
Video
-
=

Looking for a small accidental occluder

Body as the occluder

Hand as the occluder

View outside the window

Venice: The Arsenal
1755-60, Francesco Guardi

http://www.nationalgallery.org.uk/paintings/francesco-guardi-venice-the-arsenal

Notice the cast shadows under the Sun and under the building's shadow

Venice: The Arsenal
1755-60, Francesco Guardi

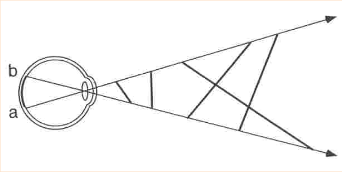
Optional Problem set

Send me pictures of accidental images

Pictures by Julian Straub

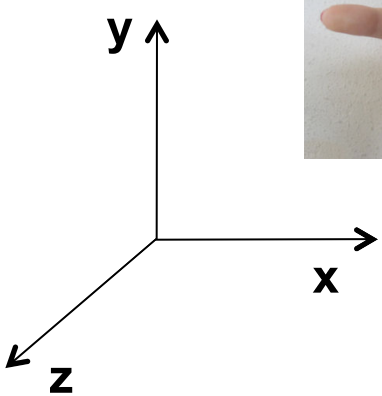
Camera Models

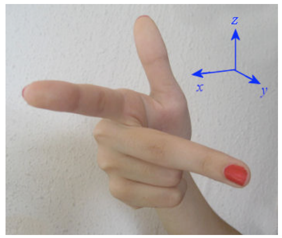
?

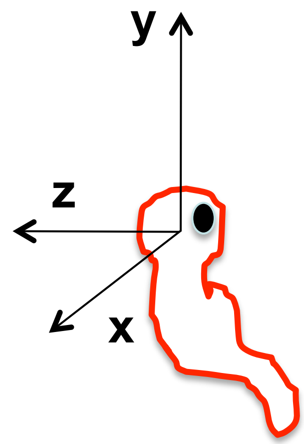


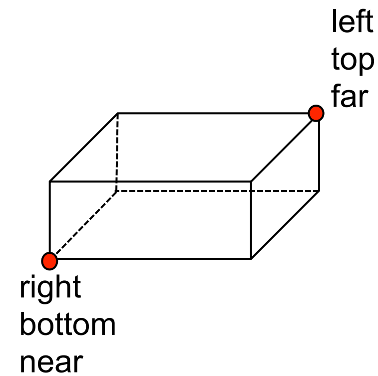
?

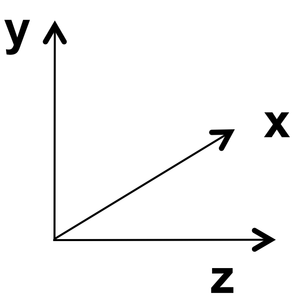
Right - handed system

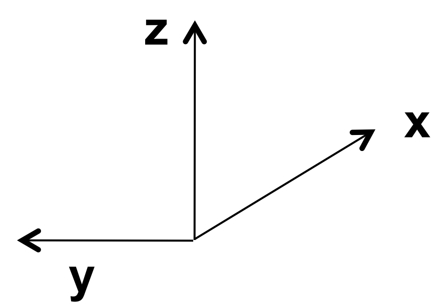




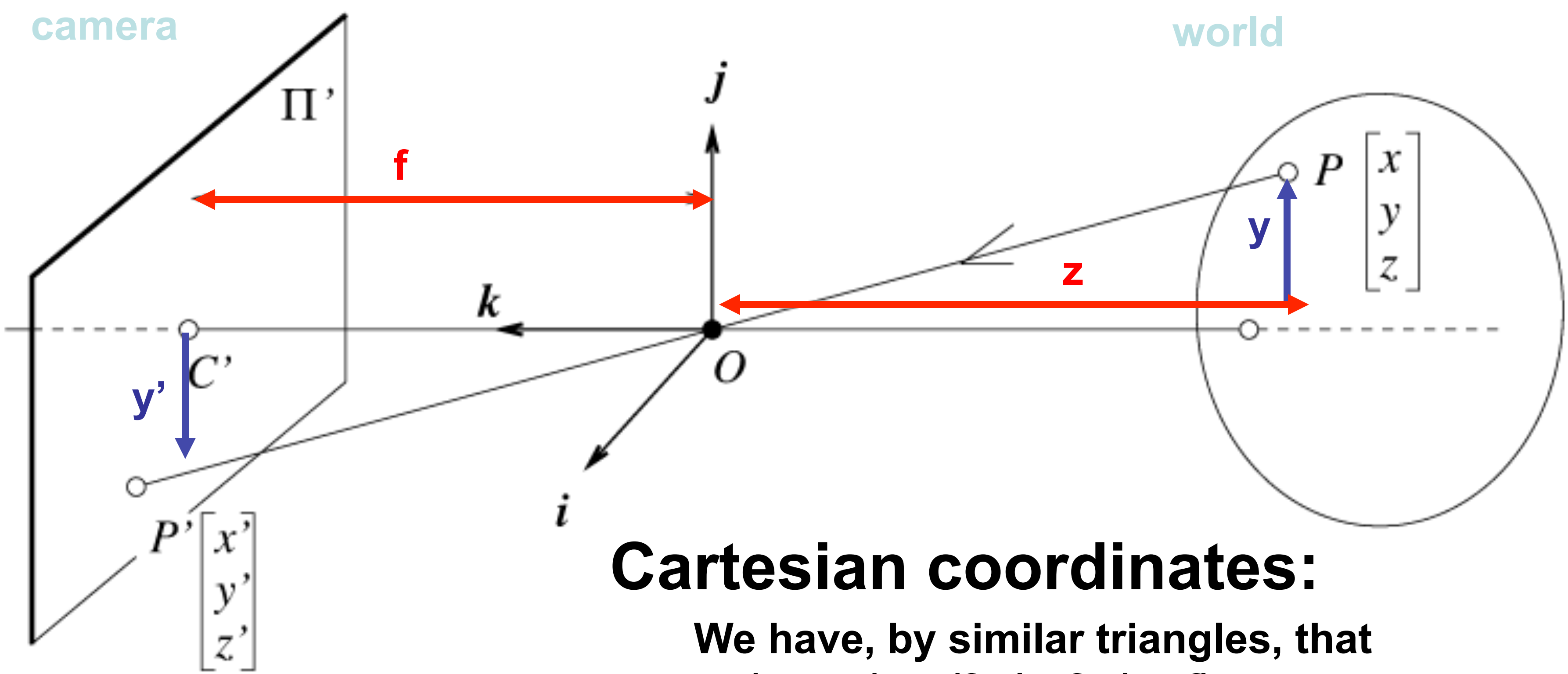








Perspective projection



Cartesian coordinates:

We have, by similar triangles, that

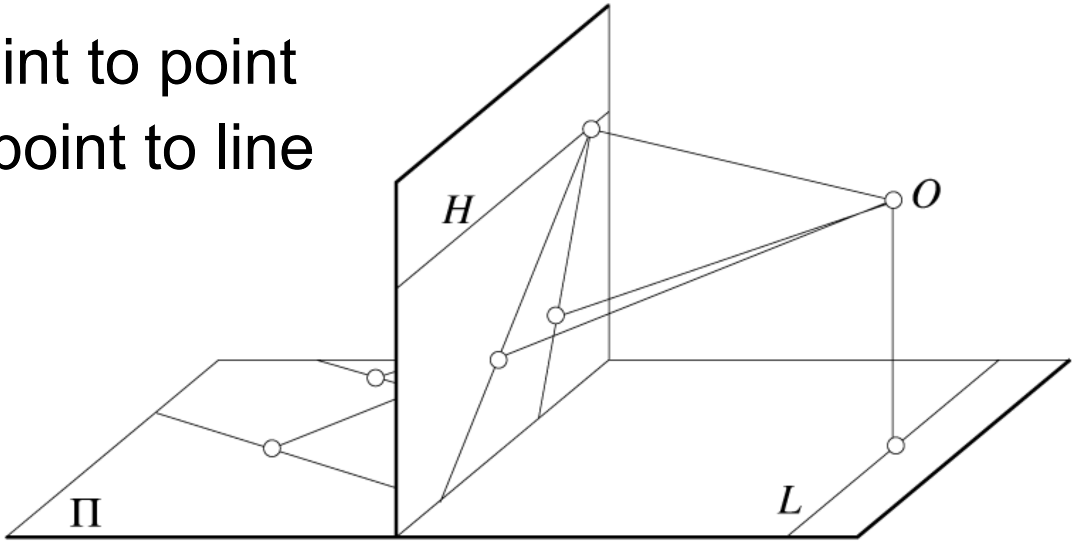
(x, y, z) -> (f x/z, f y/z, -f)

Ignore the third coordinate, and get

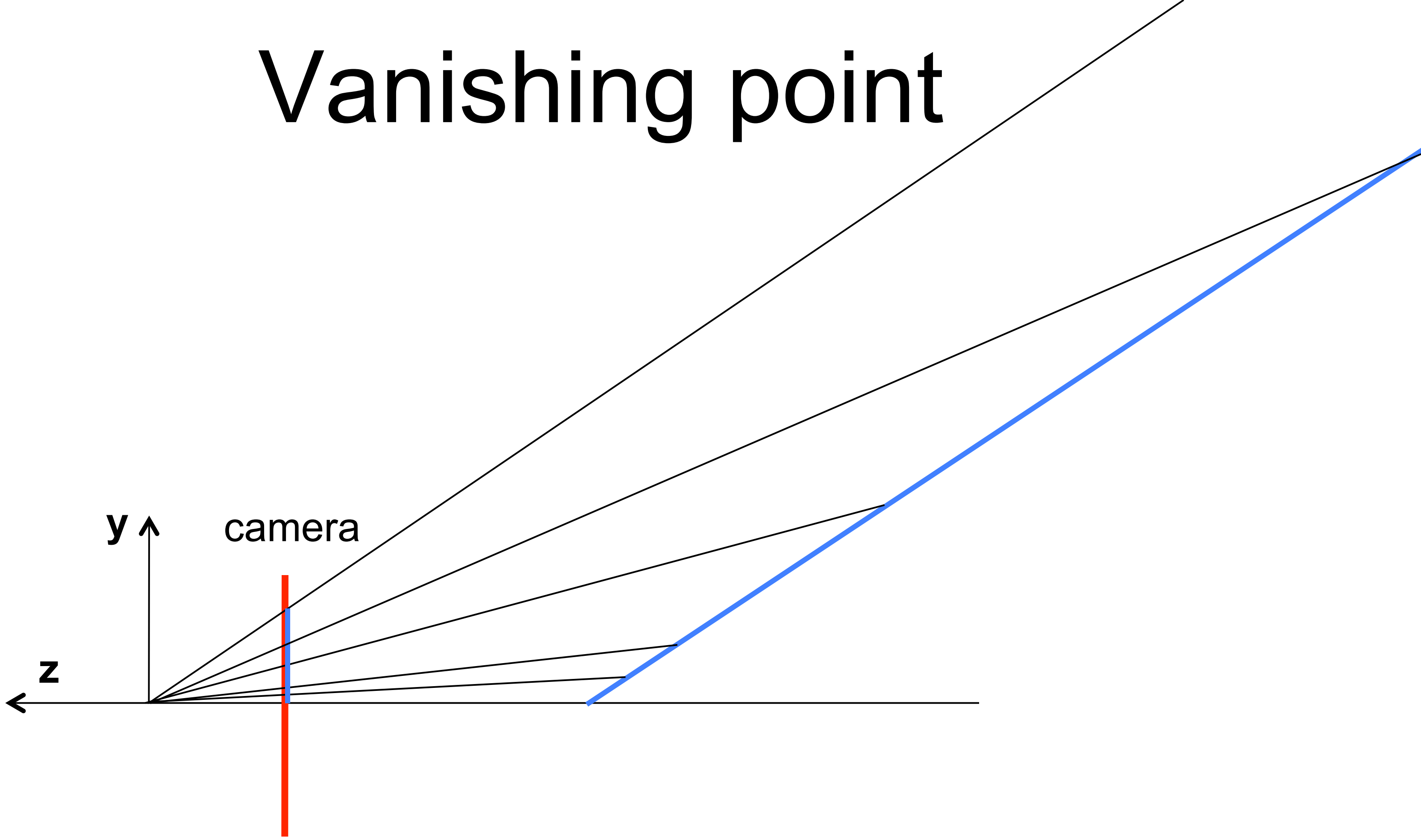
$$(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})$$

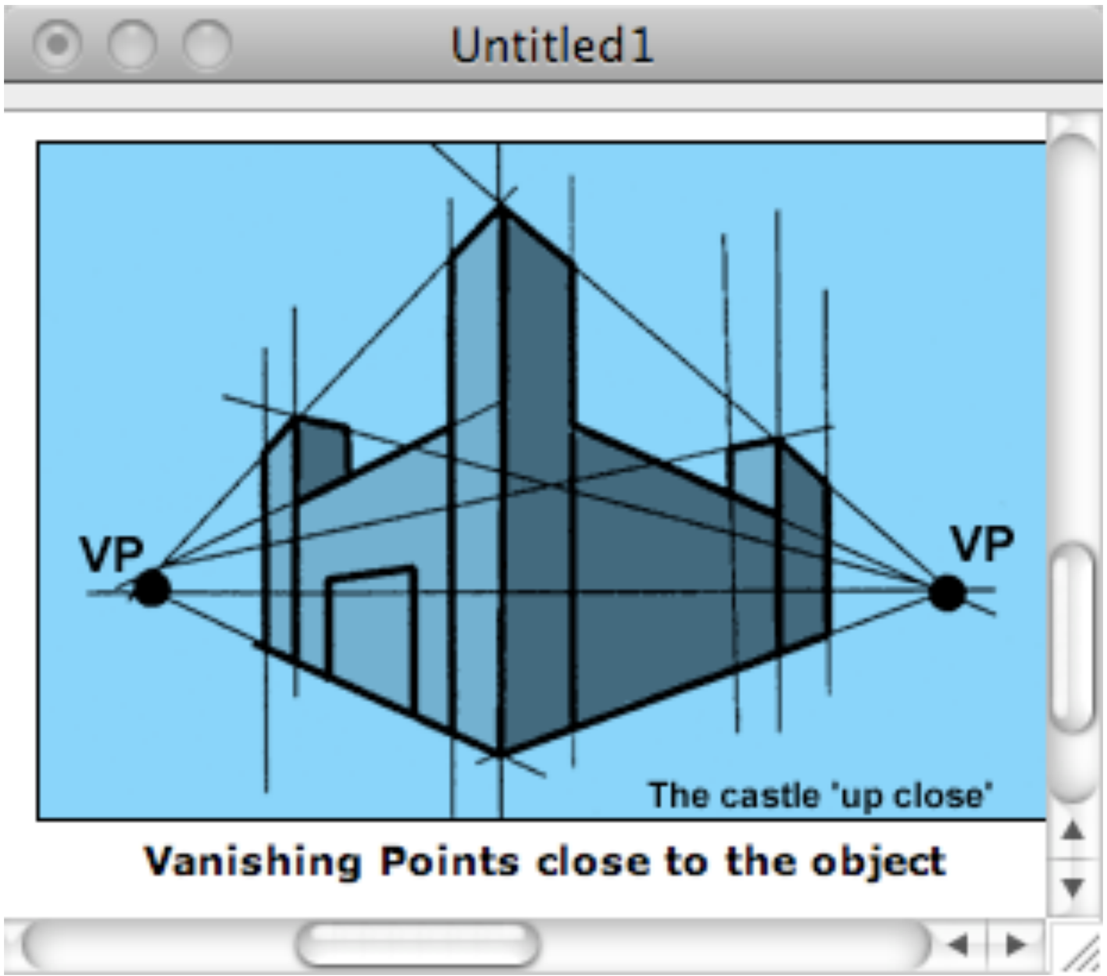
Geometric properties of projection

- Points go to points
- Lines go to lines
- Planes go to whole image or half-planes.
- Polygons go to polygons
- Degenerate cases
  - line through focal point to point
  - plane through focal point to line



Vanishing point

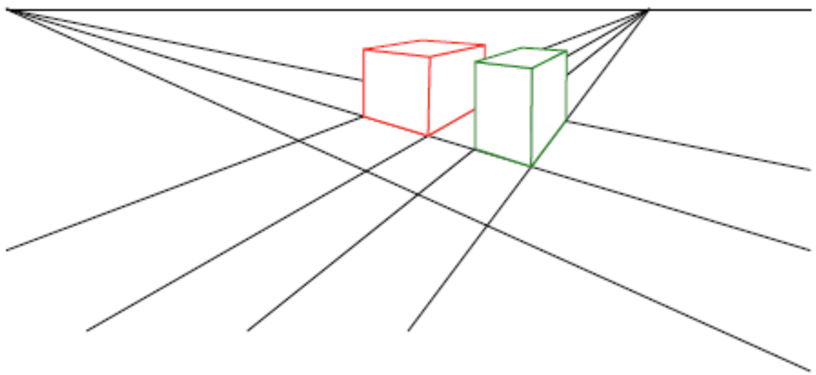




http://www.ider.herts.ac.uk/school/courseware/graphics/two_point_perspective.html

Vanishing points

- Each set of parallel lines (=direction) meets at a different point
  - The *vanishing point* for this direction
- Sets of parallel lines on the same plane lead to *collinear* vanishing points.
  - The line is called the *horizon* for that plane



Line in 3-space

$$x(t) = x_0 + at$$
$$y(t) = y_0 + bt$$
$$z(t) = z_0 + ct$$

Perspective projection of that line

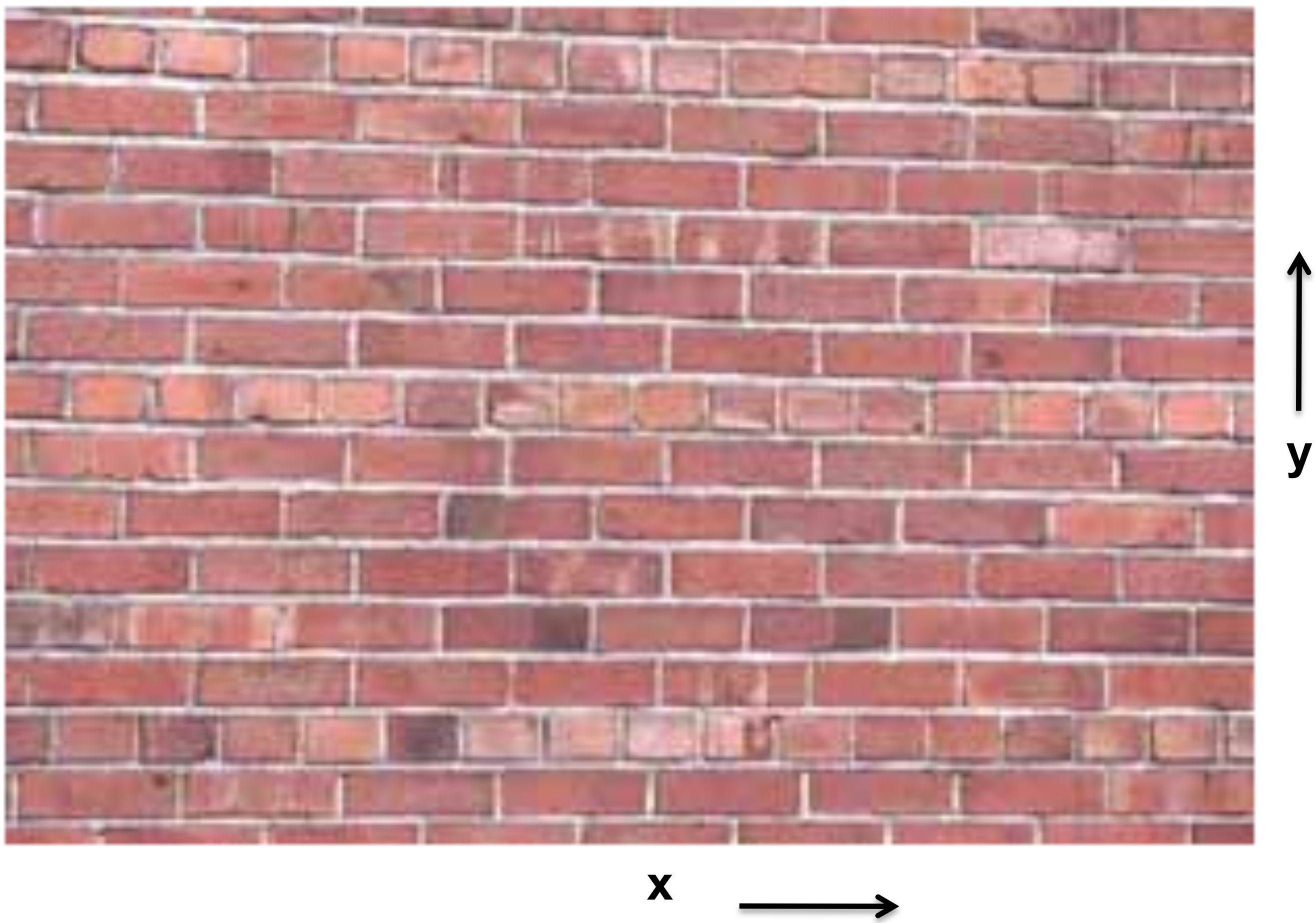
$$x'(t) = \frac{fx}{z} = \frac{f(x_0 + at)}{z_0 + ct}$$
$$y'(t) = \frac{fy}{z} = \frac{f(y_0 + bt)}{z_0 + ct}$$

In the limit as $t \longrightarrow \pm\infty$ **we have (for** $c \neq 0$ **):**

$$x'(t) \longrightarrow \frac{fa}{c}$$
$$y'(t) \longrightarrow \frac{fb}{c}$$

This tells us that any set of parallel lines (same a, b, c parameters) project to the same point (called the vanishing point).

What if you photograph a brick wall head-on?



Brick wall line in 3-space

$$x(t) = x_0 + at$$

$$y(t) = y_0$$

$$z(t) = z_0$$

Perspective projection of that line

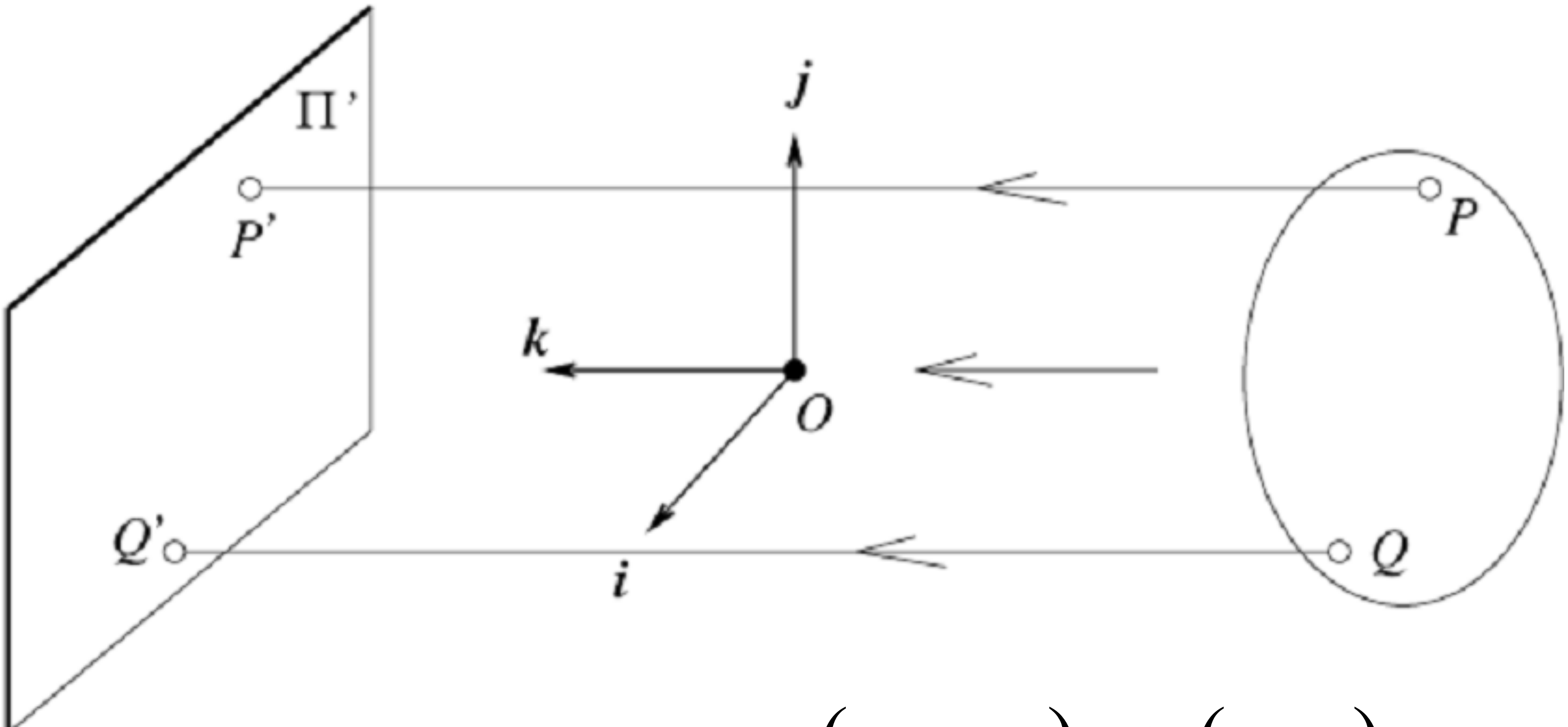
$$x'(t) = \frac{f \cdot (x_0 + at)}{z_0}$$

$$y'(t) = \frac{f \cdot y_0}{z_0}$$

All bricks have same z_0. Those in same row have same y_0

Thus, a brick wall, photographed head-on, gets rendered as set of parallel lines in the image plane.

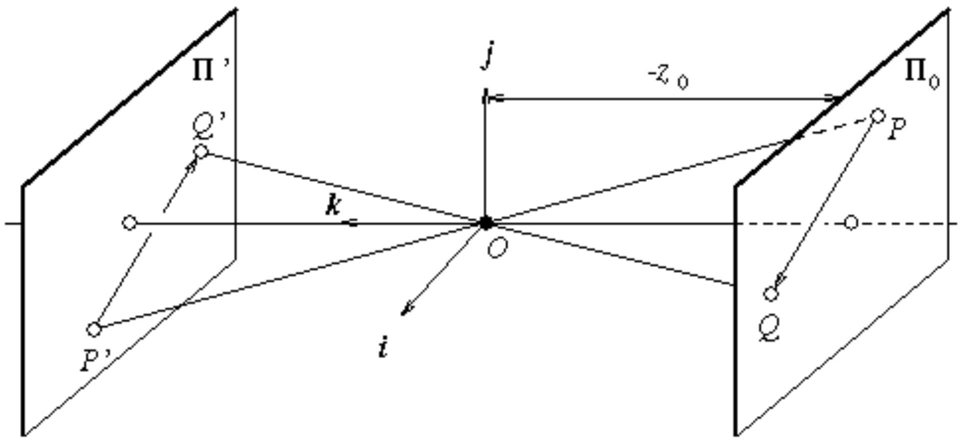
Other projection models: Orthographic projection



$$(x, y, z) \rightarrow (x, y)$$

Other projection models: Weak perspective

- Issue
  - perspective effects, but not over the scale of individual objects
  - collect points into a group at about the same depth, then divide each point by the depth of its group
  - Adv: easy
  - Disadv: only approximate



$$(x, y, z) \to \left(\frac{fx}{z_0}, \frac{fy}{z_0} \right)$$

Three camera projections

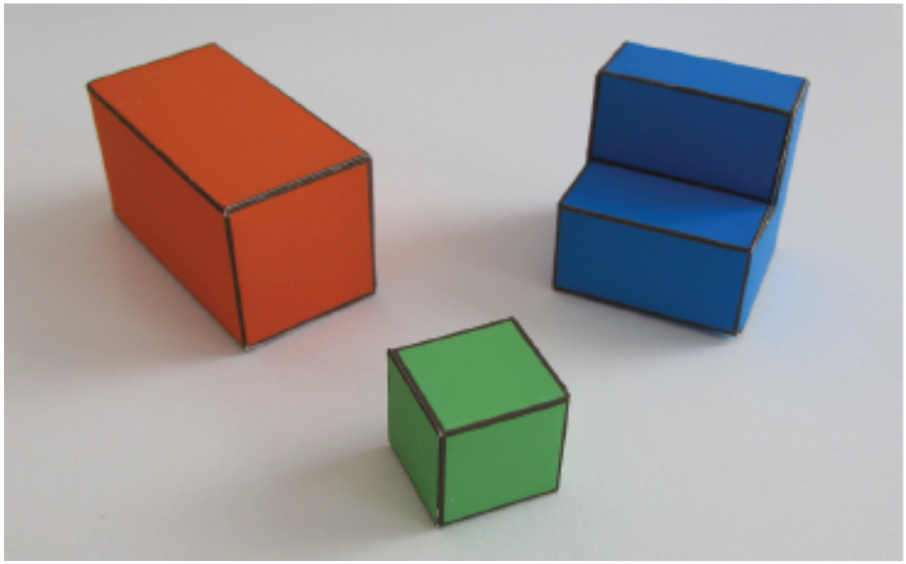
3-d point **2-d image position**

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

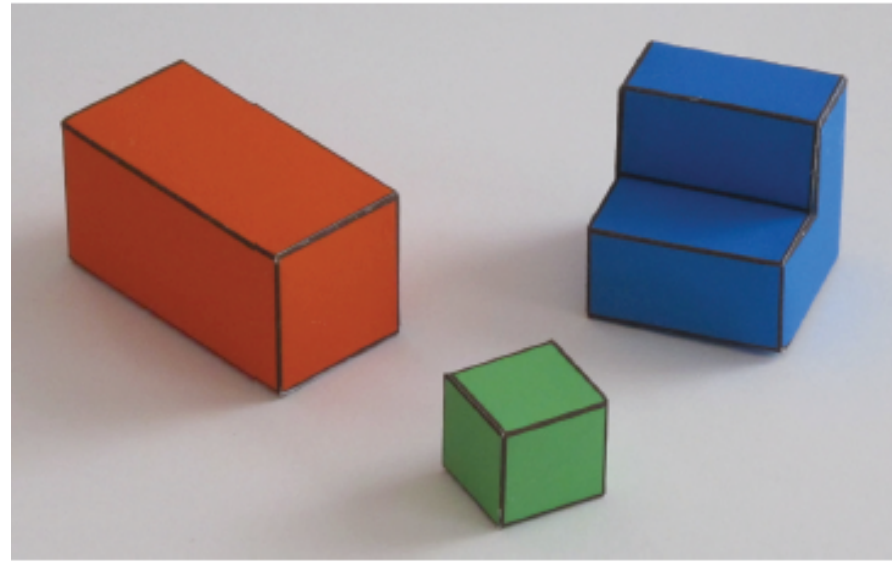
(2) Weak perspective: $$(x, y, z) \rightarrow \left(\frac{fx}{z_0}, \frac{fy}{z_0} \right)$$

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

Three camera projections



Perspective projection



Parallel (orthographic) projection

Weak perspective?

Homogeneous coordinates

- Is this 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

- Projection is a matrix multiply using homogeneous coordinates:

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

This is known as perspective projection

- **The matrix is the projection matrix**

Slide by Steve Seitz

Perspective Projection

How does scaling the projection matrix change the transformation?

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

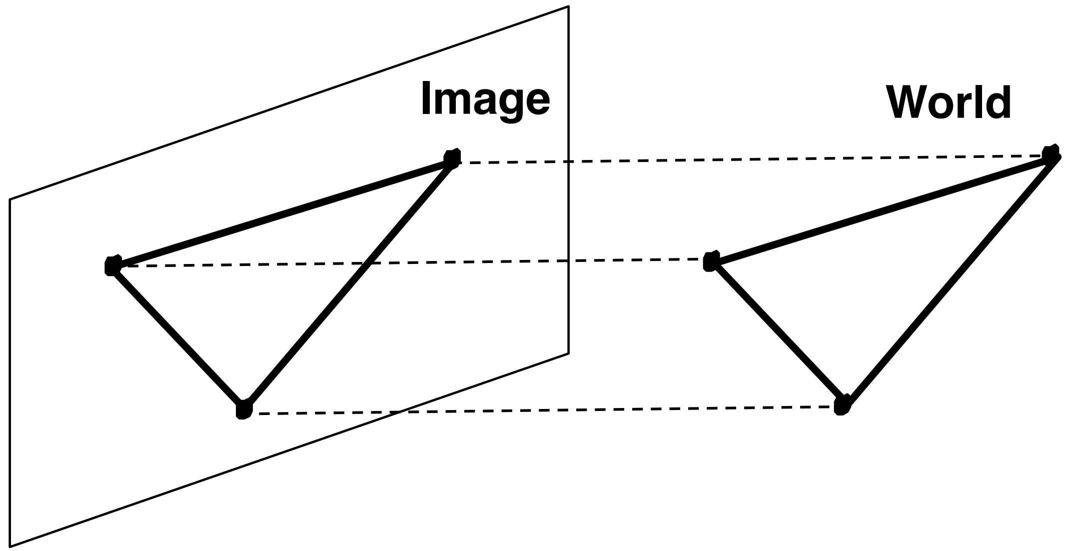
$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} fx \\ fy \\ z \end{bmatrix} \Rightarrow \left(f\frac{x}{z}, f\frac{y}{z} \right)$$

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?

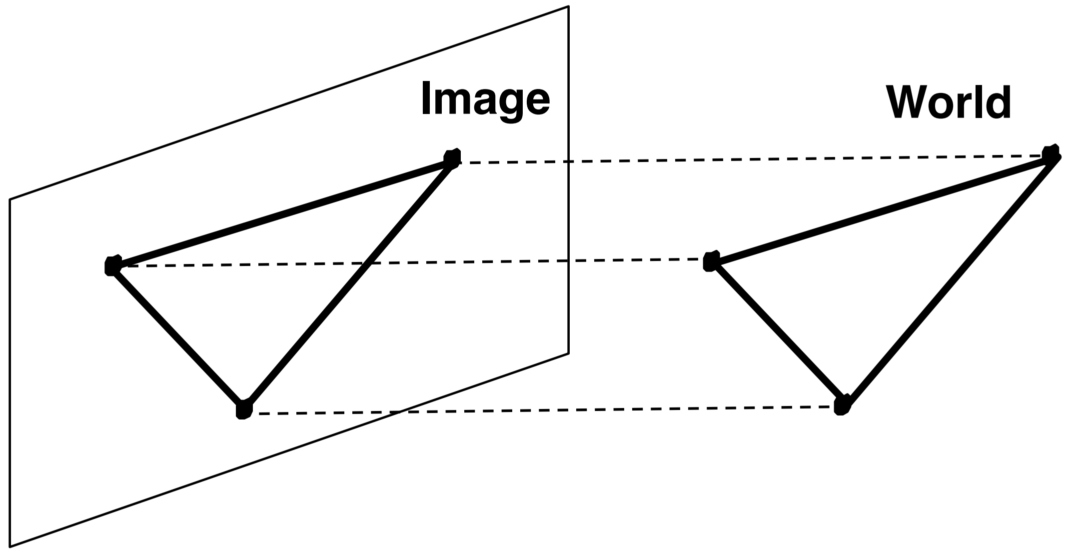
$$\boxed{?}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow (x, y)$$

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?

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

Slide by Steve Seitz

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

Homogeneous coordinates

2D Points:

$$p = \begin{bmatrix} x \\ y \end{bmatrix} \longrightarrow p' = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

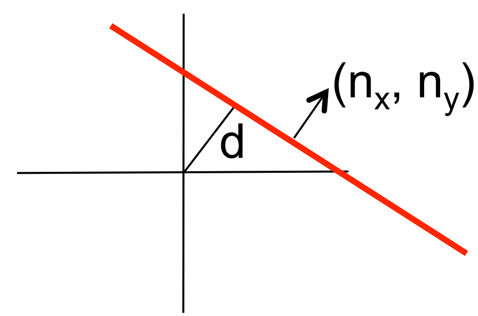
$$p' = \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} \longrightarrow p = \begin{bmatrix} x'/w' \\ y'/w' \end{bmatrix}$$

2D Lines:

$$ax + by + c = 0$$

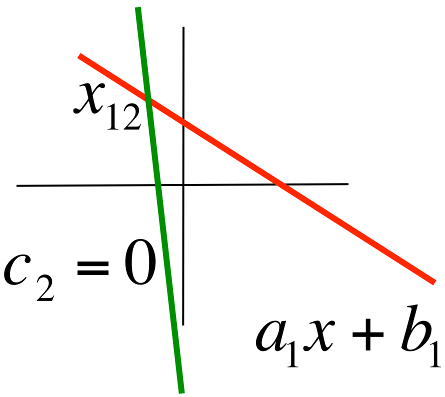
$$\begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

$$l = \begin{bmatrix} a & b & c \end{bmatrix} \Rightarrow \begin{bmatrix} n_x & n_y & d \end{bmatrix}$$



Homogeneous coordinates

Intersection between two lines:



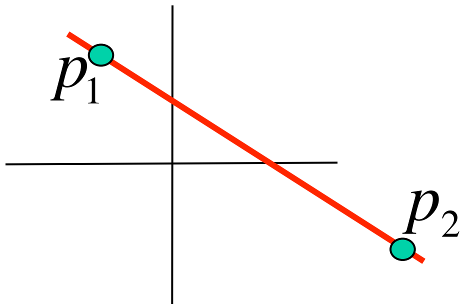
$$a_2 x + b_2 y + c_2 = 0$$

$$a_1 x + b_1 y + c_1 = 0$$

$$\left.\begin{array}{l} l_1 = \begin{bmatrix} a_1 & b_1 & c_1 \end{bmatrix} \\ \\ l_2 = \begin{bmatrix} a_2 & b_2 & c_2 \end{bmatrix} \end{array}\right\} \quad x_{12} = l_1 \times l_2$$

Homogeneous coordinates

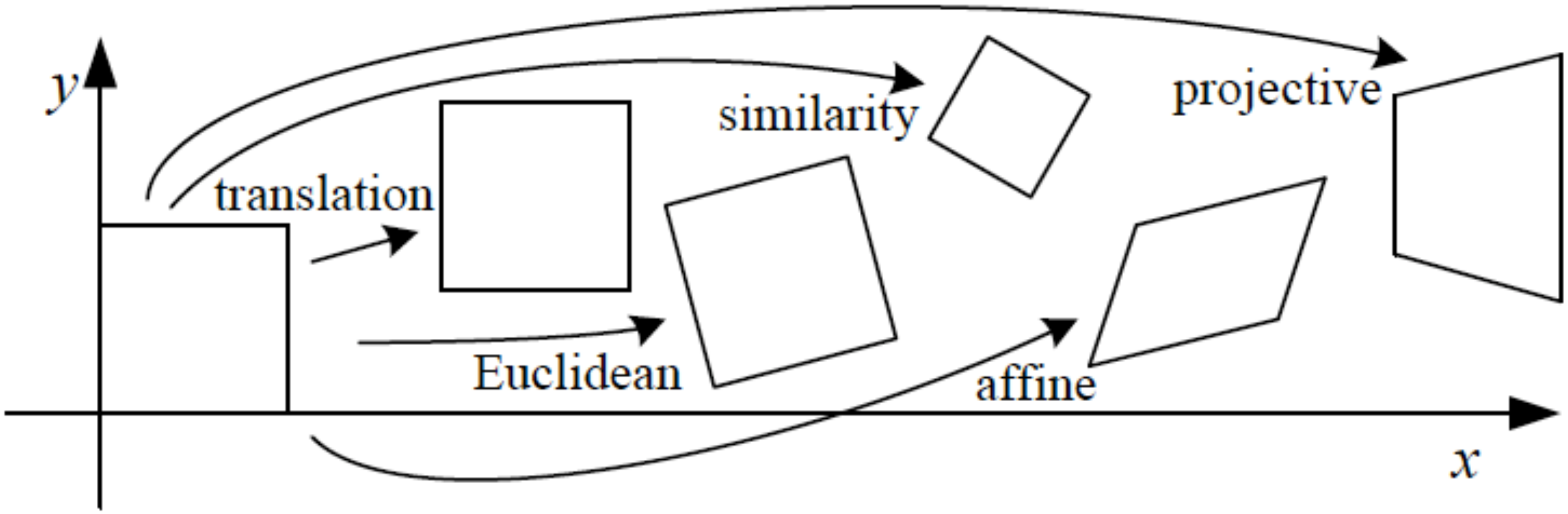
Line joining two points:



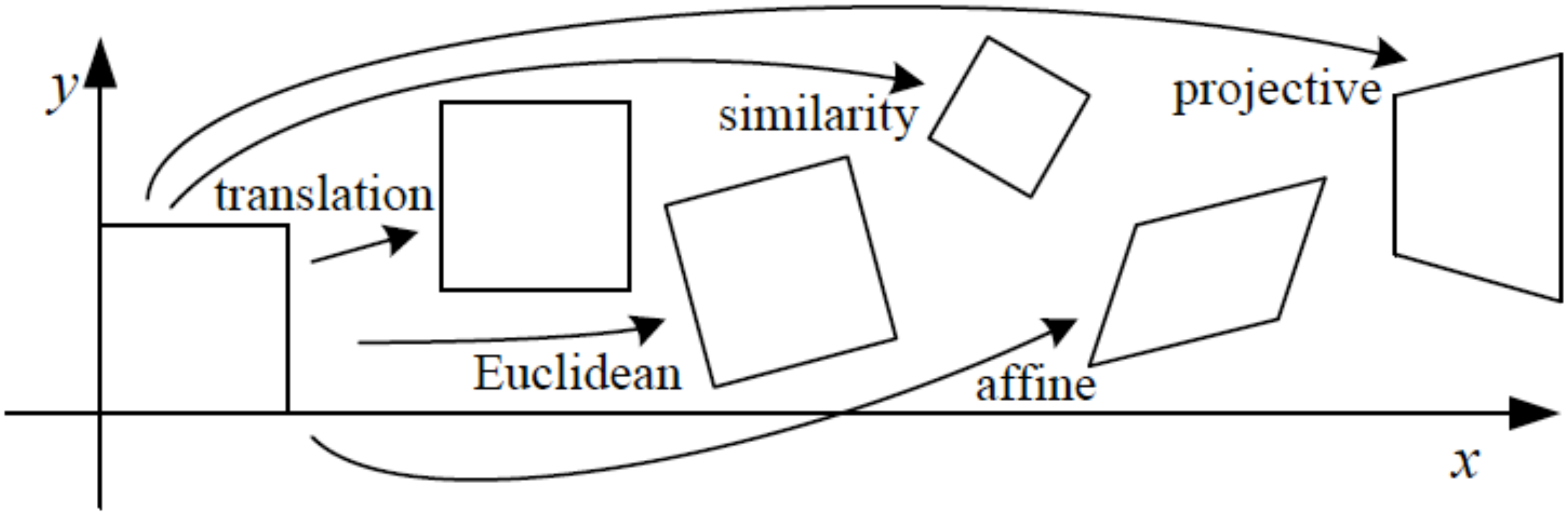
$$ax + by + c = 0$$

$$\left.\begin{array}{l} p_1 = \begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix} \\ \\ p_2 = \begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix} \end{array}\right\} \quad l = p_1 \times p_2$$

2D Transformations

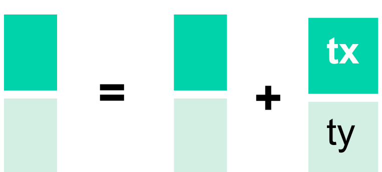


2D Transformations

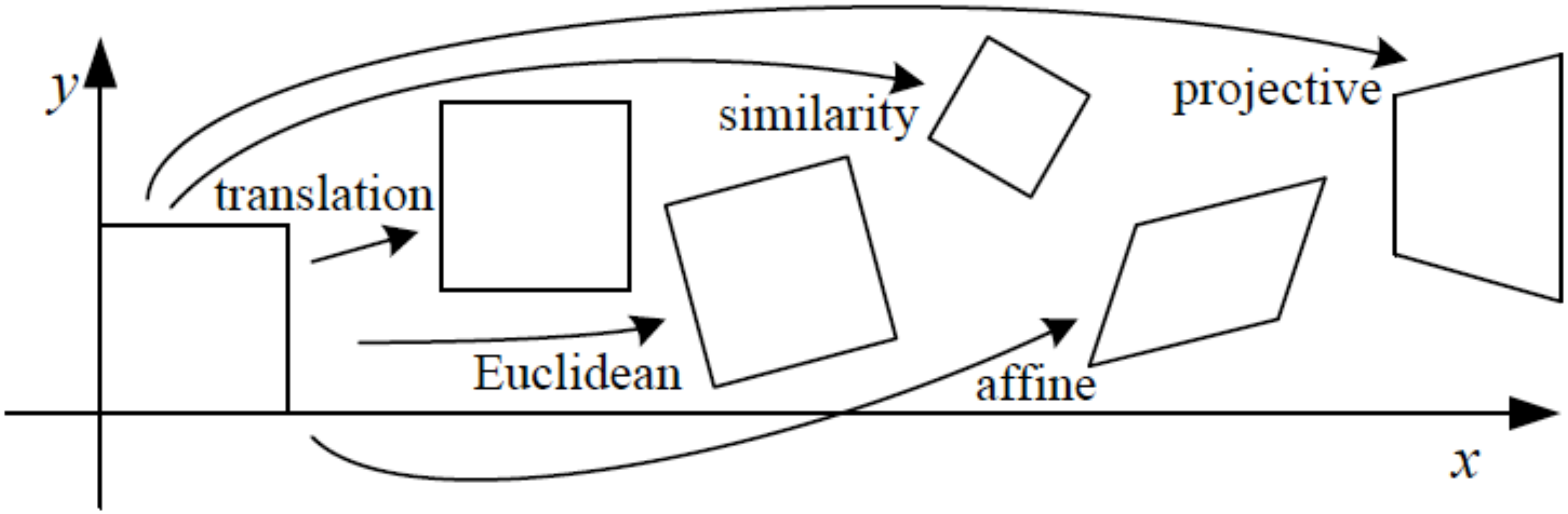


Example: translation

$$\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{t}$$

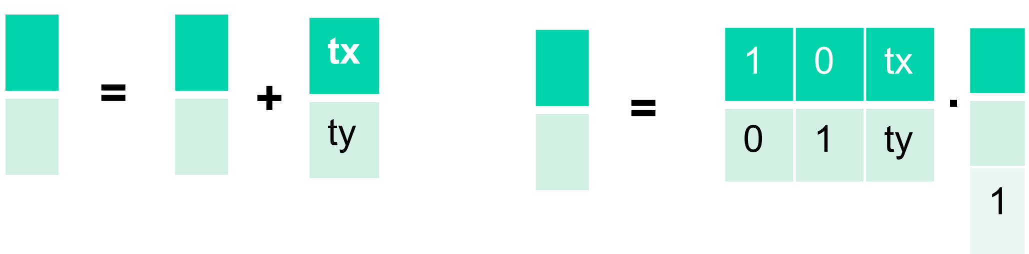


2D Transformations

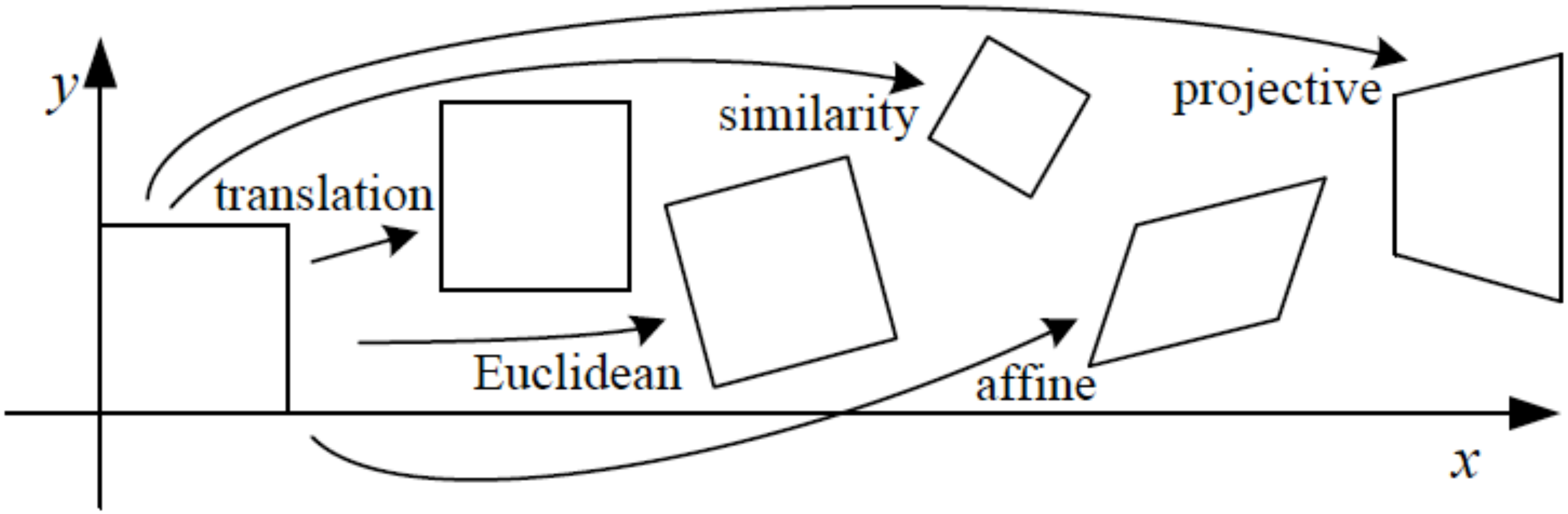


Example: translation

$$x' = x + t \qquad\qquad x' = \begin{bmatrix} I & t \end{bmatrix} \bar{x}$$



2D Transformations



Example: translation

$$x' = x + t \qquad x' = \left[\, I \quad t \,\right] \bar{x} \qquad \bar{x}' = \begin{bmatrix} I & t \\ 0^T & 1 \end{bmatrix} \bar{x}$$

$$\begin{bmatrix} \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot \\ \cdot \end{bmatrix} + \begin{bmatrix} tx \\ ty \end{bmatrix} \qquad \begin{bmatrix} \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \end{bmatrix} \cdot \begin{bmatrix} \cdot \\ \cdot \\ 1 \end{bmatrix} \qquad \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

Now we can chain transformations

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

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

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, section 5.2, 5.3 for references
- (Relationship between intensities in the world and intensities in the image: *photometric image formation*, see Szeliski, sect. 2.2.)

Camera calibration

- Intrinsic parameters

Image coordinates relative to camera ←→ Pixel coordinates

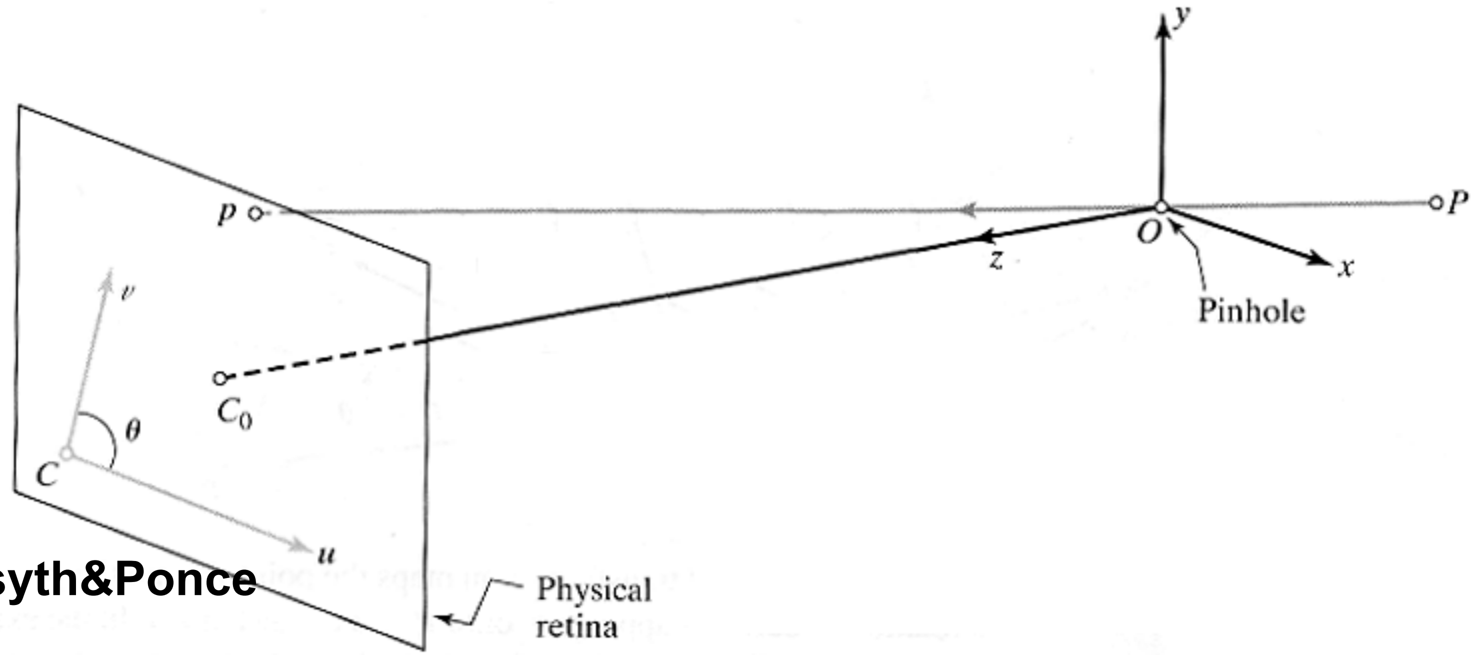
- Extrinsic parameters

Camera frame 1 ←→ Camera frame 2

Camera calibration

- Intrinsic parameters
- Extrinsic parameters

Intrinsic parameters: from idealized world coordinates to pixel values



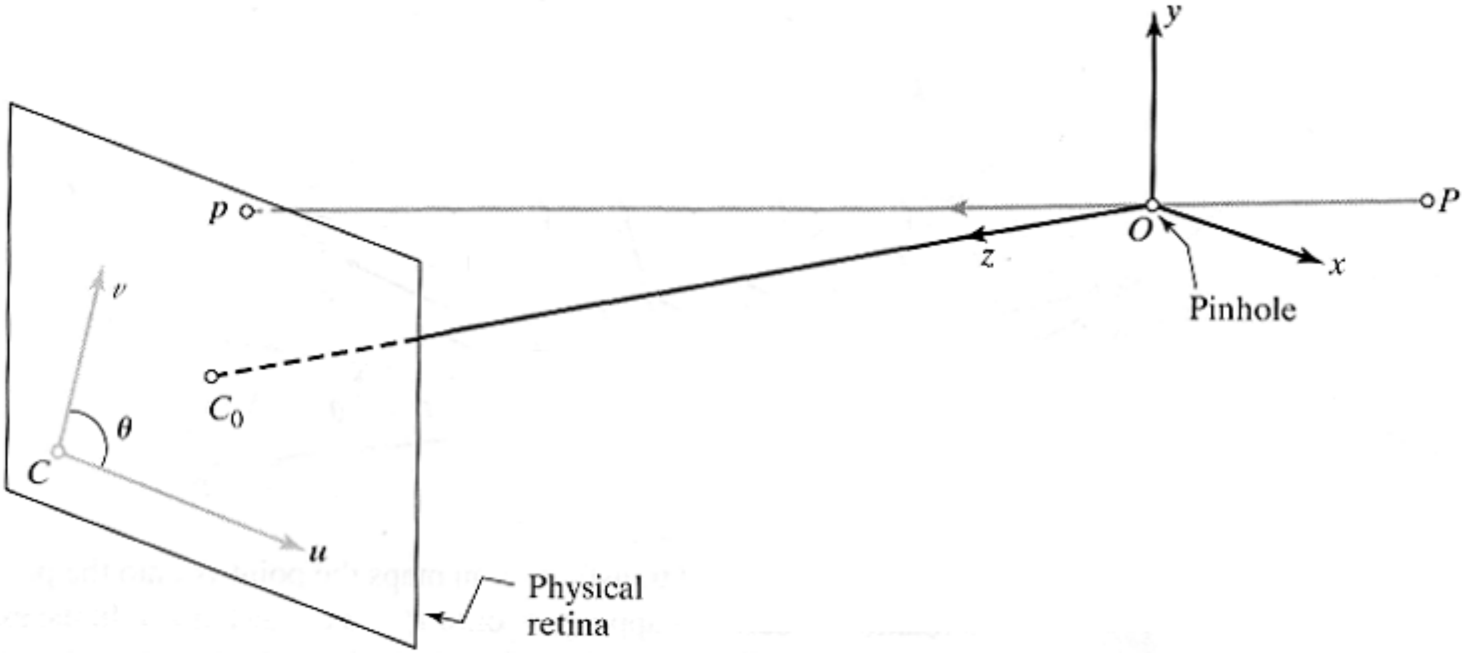
Forsyth&Ponce

Perspective projection

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

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

Intrinsic parameters

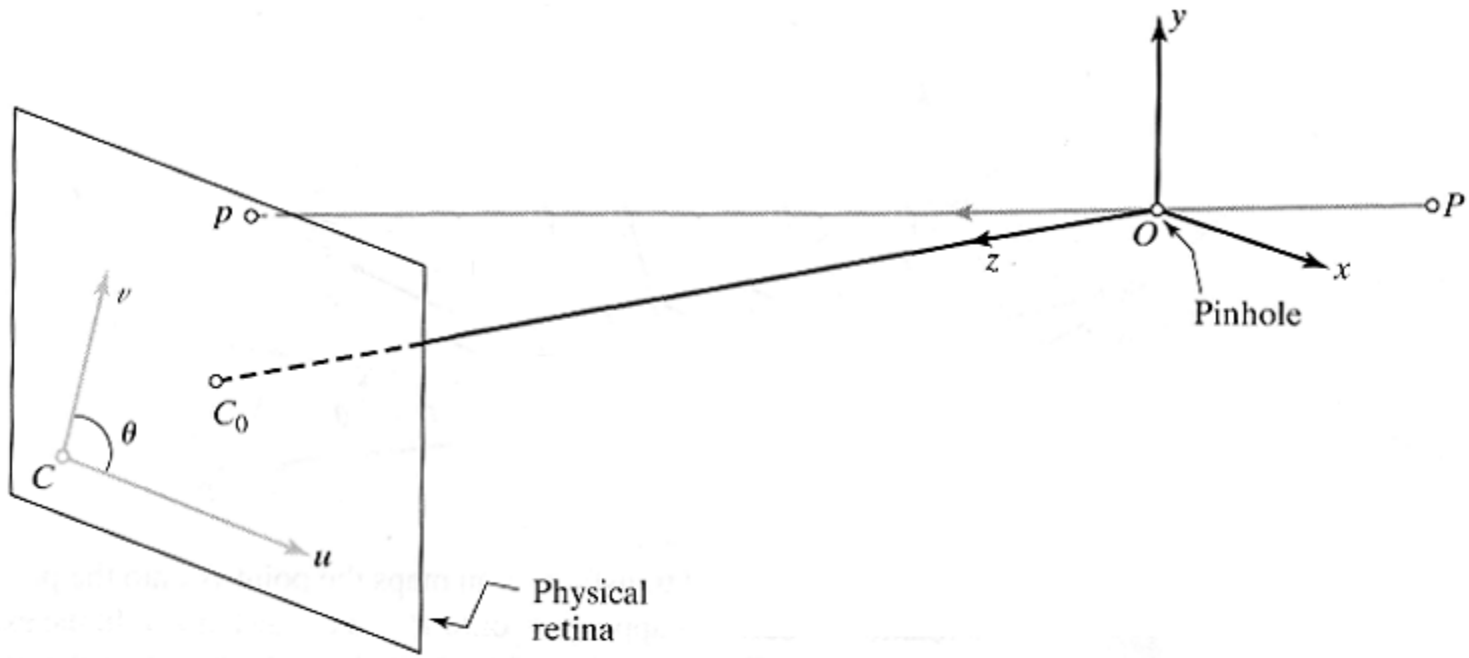


But "pixels" are in some arbitrary spatial units

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

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

Intrinsic parameters

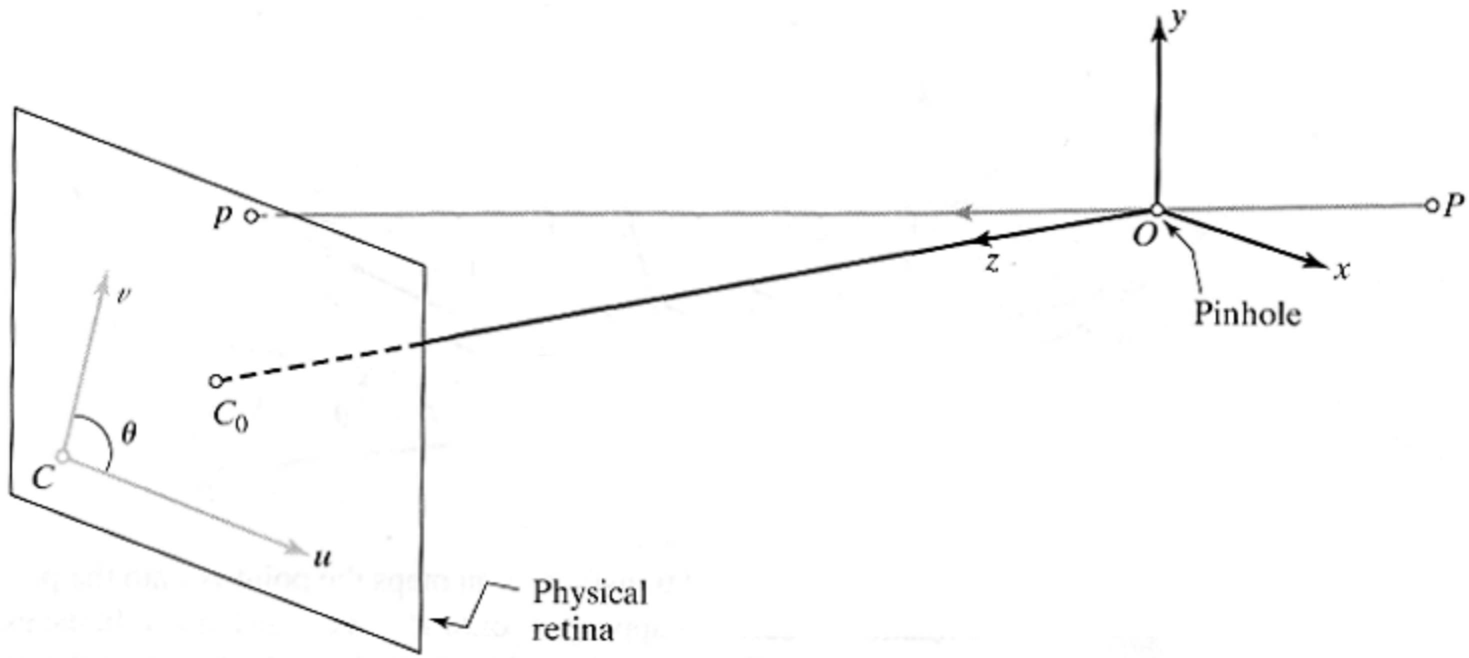


Maybe pixels are not square

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

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

Intrinsic parameters

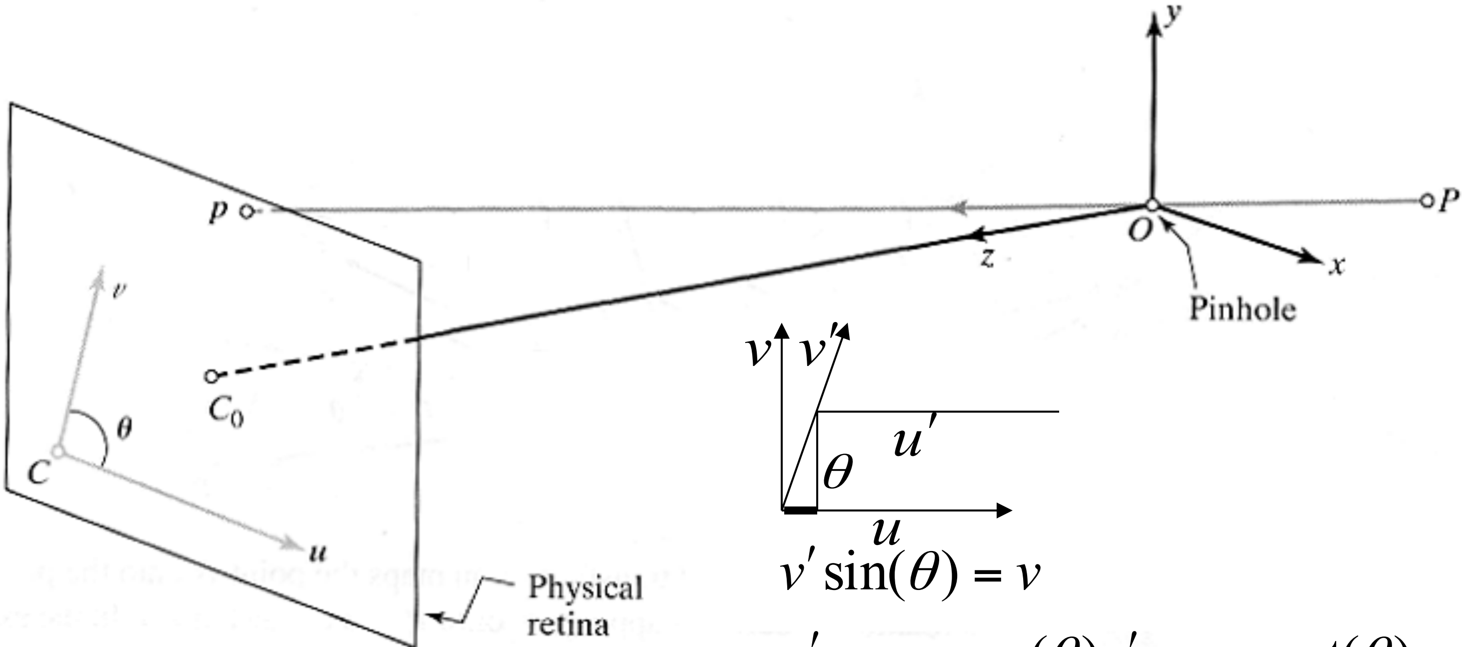


We don't know the origin of our camera pixel coordinates

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

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

Intrinsic parameters



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

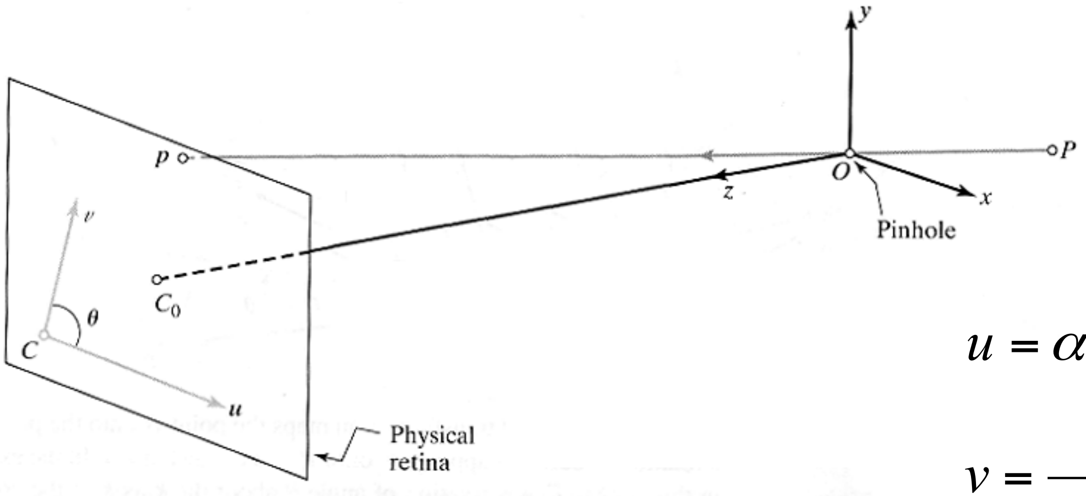
$$u' = u - \cos(\theta) v' = u - \cot(\theta) v$$

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



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

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

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 → $\vec{p} = K \; {}^{C}\vec{p}$

In camera-based coords

Camera calibration

- Intrinsic parameters
- Extrinsic parameters

Extrinsic parameters: translation and rotation of camera frame

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

Non-homogeneous coordinates

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

Homogeneous coordinates

Combining extrinsic and intrinsic calibration parameters, in homogeneous coordinates

pixels

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

Intrinsic

Camera coordinates

World coordinates

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

Extrinsic

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

$$0\;0\;0 \qquad 1$$

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

Forsyth&Ponce

Other ways to write the same equation

pixel coordinates

world coordinates

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

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

Conversion back from homogeneous coordinates leads to:

$$\begin{cases} u = \dfrac{m_1 \cdot \vec{P}}{m_3 \cdot \vec{P}} \\[2ex] v = \dfrac{m_2 \cdot \vec{P}}{m_3 \cdot \vec{P}} \end{cases}$$

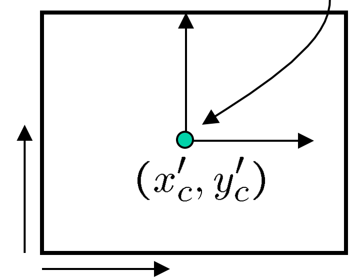
Camera parameters

A camera is described by several parameters

- Translation T of the optical center from the origin of world coords
- Rotation R of the image plane
- focal length f, principle point (x'_c, y'_c), pixel size (s_x, s_y)
- blue parameters are called "extrinsics," red are "intrinsics"

Projection equation

$$\mathbf{X} = \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{\Pi X}$$



- The projection matrix models the cumulative effect of all parameters
- Useful to decompose into a series of operations

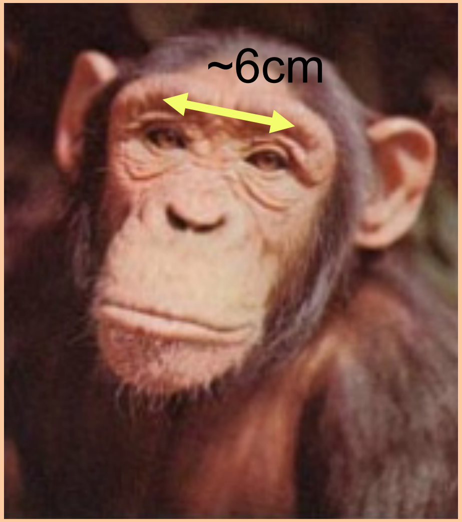
identity matrix

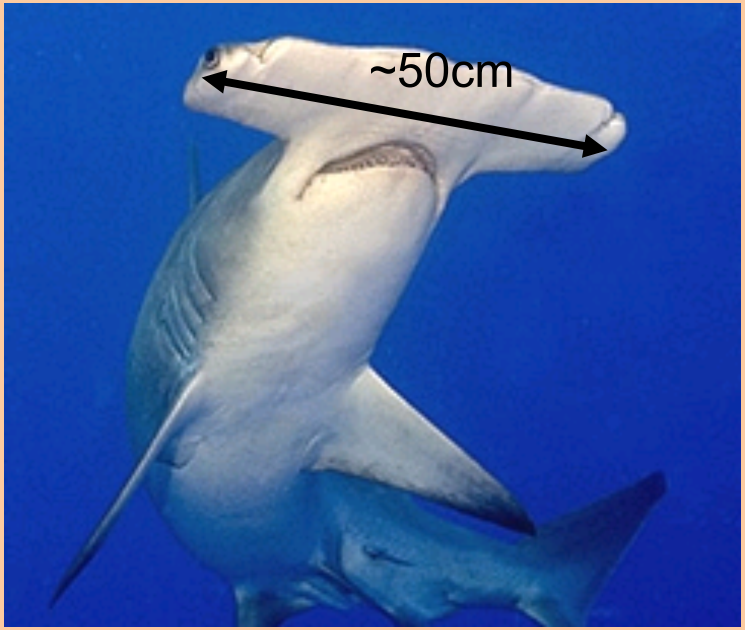
$$\Pi = \begin{bmatrix} -fs_x & 0 & x'_c \\ 0 & -fs_y & y'_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{3x3} & \mathbf{0}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{T}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}$$

intrinsics projection rotation translation

- The definitions of these parameters are not completely standardized
  - especially intrinsics—varies from one book to another

Stereo vision





Depth without objects

Random dot stereograms (Bela Julesz)

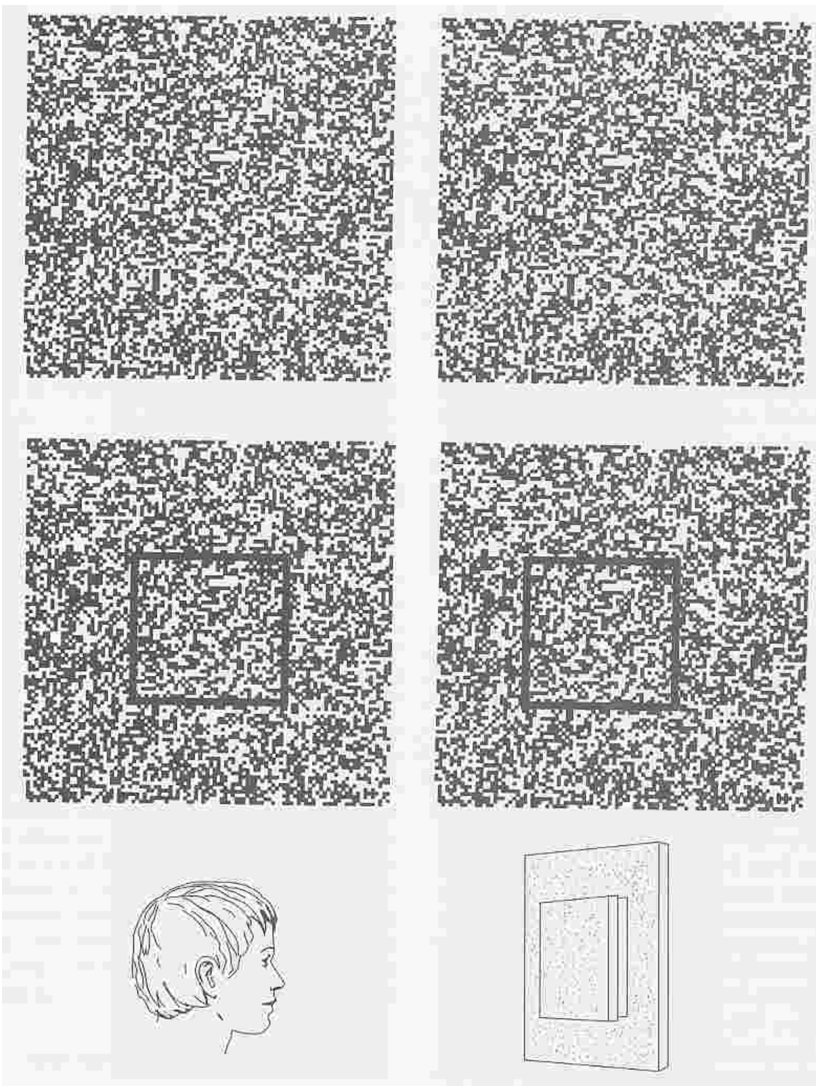


FIGURE 8.13

1	0	1	0	1	0	0	1	0	1
1	0	0	1	0	1	0	1	0	0
0	0	1	1	0	1	1	0	1	0
0	1	0	Y	A	A	B	B	0	1
1	1	1	X	B	A	B	A	0	1
0	0	1	X	A	A	B	A	1	0
1	1	1	Y	B	B	A	B	0	1
1	0	0	1	1	0	1	1	0	1
1	1	0	0	1	1	0	1	1	1
0	1	0	0	0	1	1	1	1	0

1	0	1	0	1	0	0	1	0	1
1	0	0	1	0	1	0	1	0	0
0	0	1	1	0	1	1	0	1	0
0	1	0	A	A	B	B	X	0	1
1	1	1	B	A	B	A	Y	0	1
0	0	1	A	A	B	A	Y	1	0
1	1	1	B	B	A	B	X	0	1
1	0	0	1	1	0	1	1	0	1
1	1	0	0	1	1	0	1	1	1
0	1	0	0	0	1	1	1	1	0

Julesz, 1971

Depth for familiar objects

(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

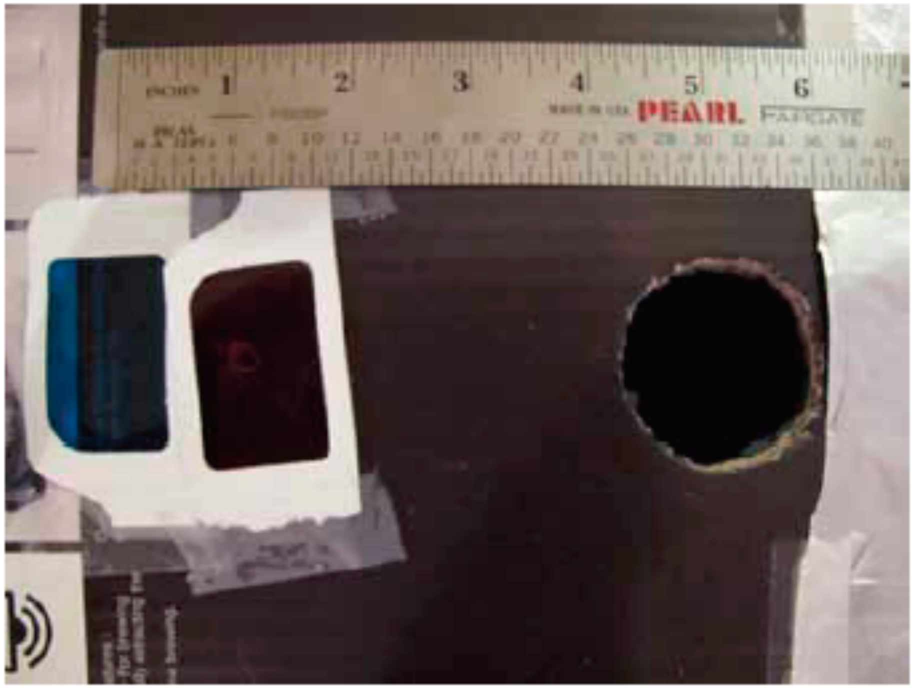
Image courtesy of fisher-price.com

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.

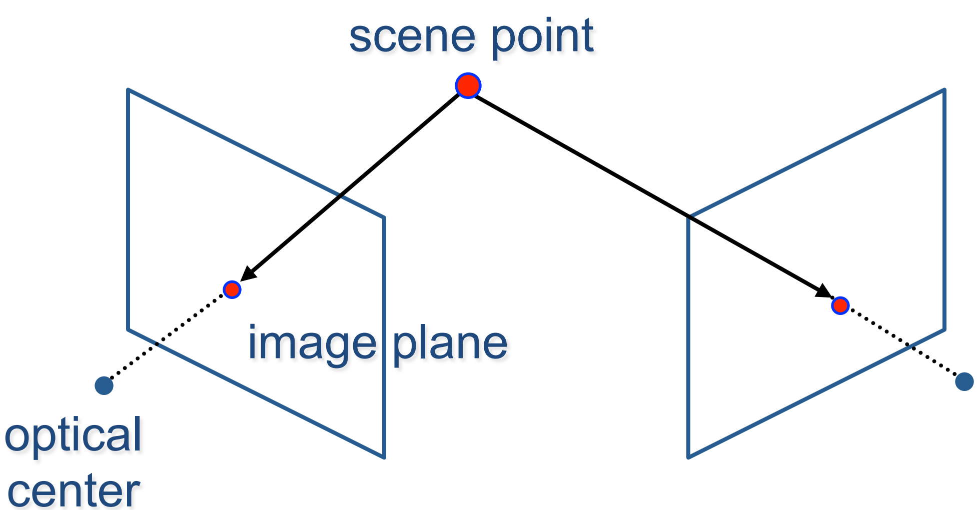
(Single image random dot stereogram, Single image stereogram)

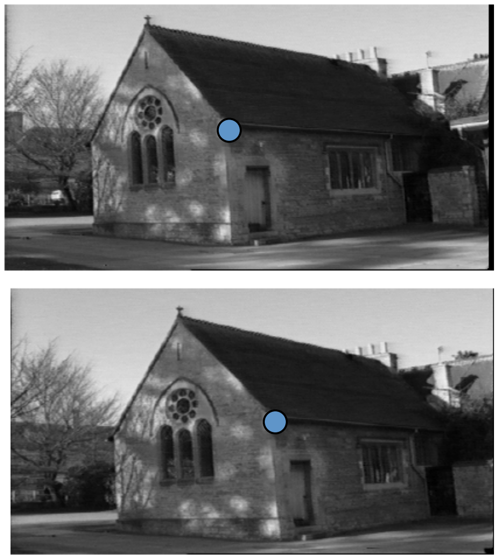
Images from magiceye.com

Slide credit: Kristen Grauman

Estimating depth with stereo

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

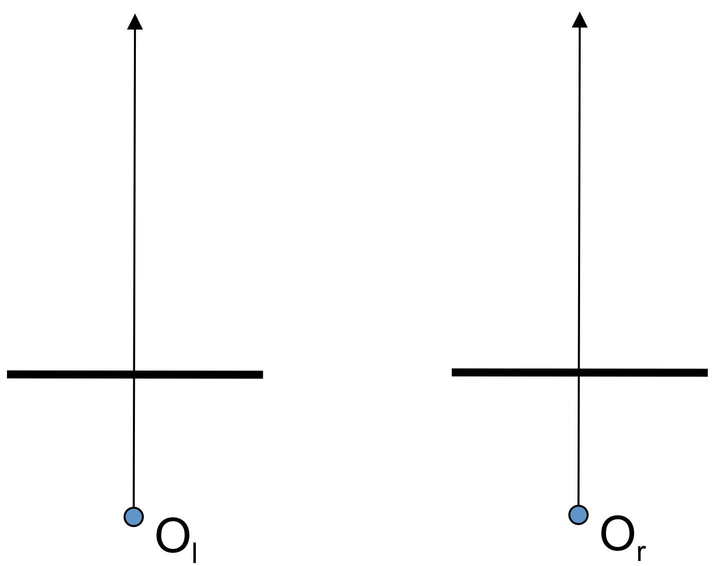


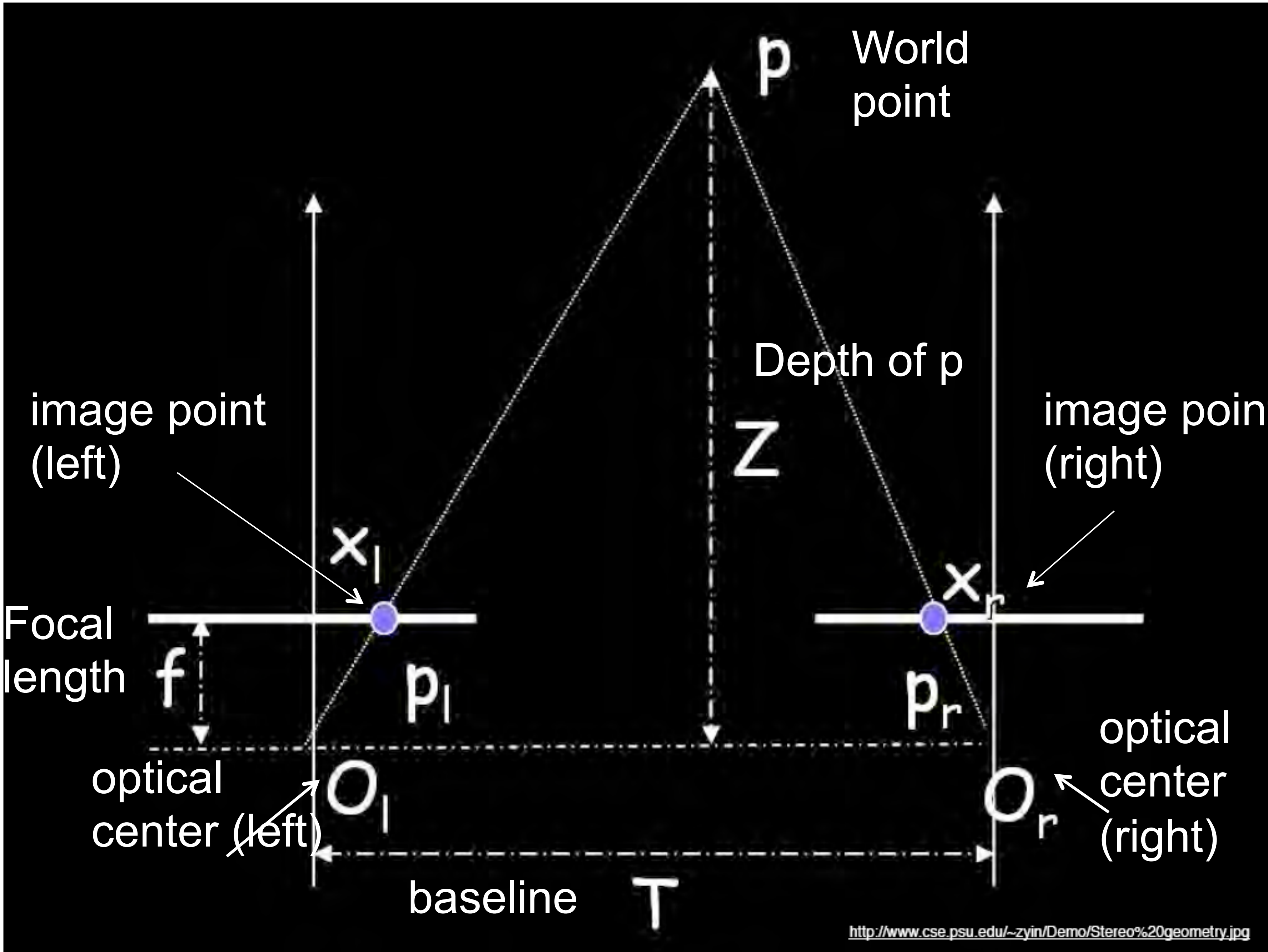


Slide credit: Kristen Grauman

Geometry for a simple stereo system

- Assume a simple setting:
  - Two identical cameras
  - parallel optical axes
  - known camera parameters (i.e., calibrated cameras).

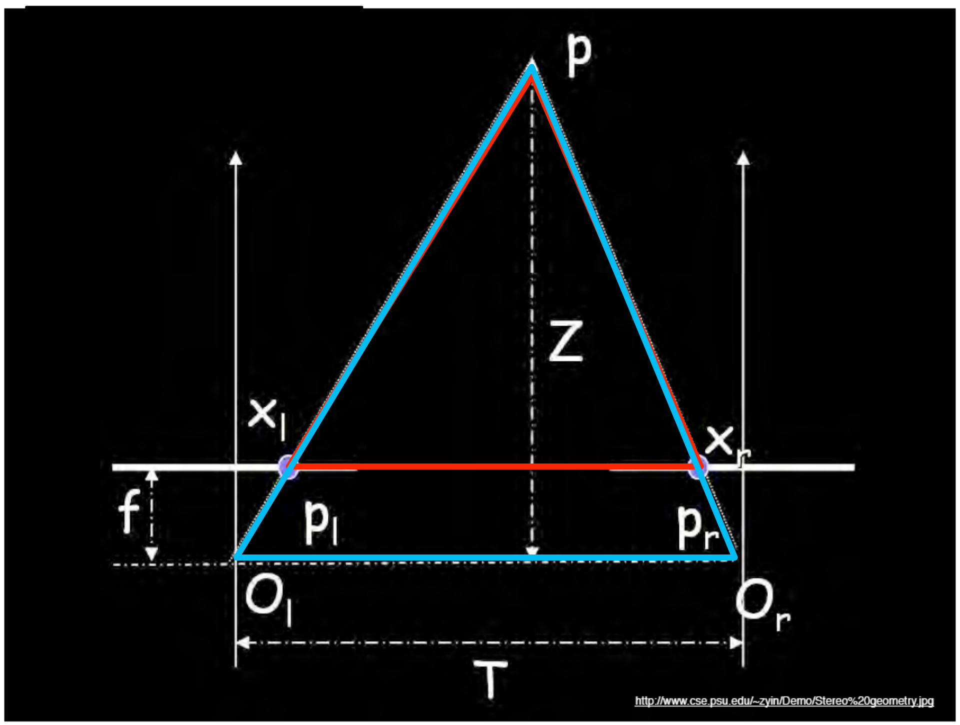




Slide credit: Kristen Grauman

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 → $x_r - x_l$

Slide credit: Kristen Grauman

Depth from disparity

image I(x,y)

Disparity map D(x,y)

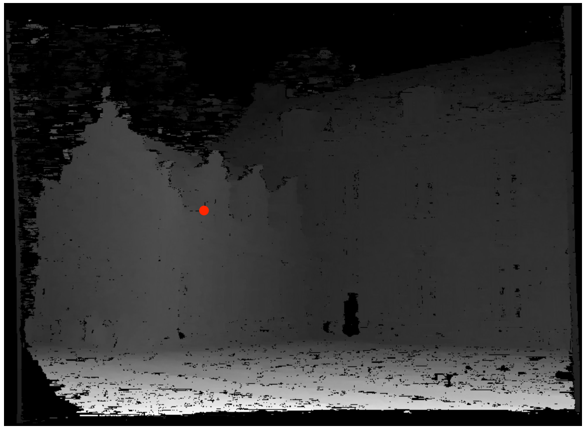


image I´(x´,y´)

$(x´,y´)=(x+D(x,y), y)$

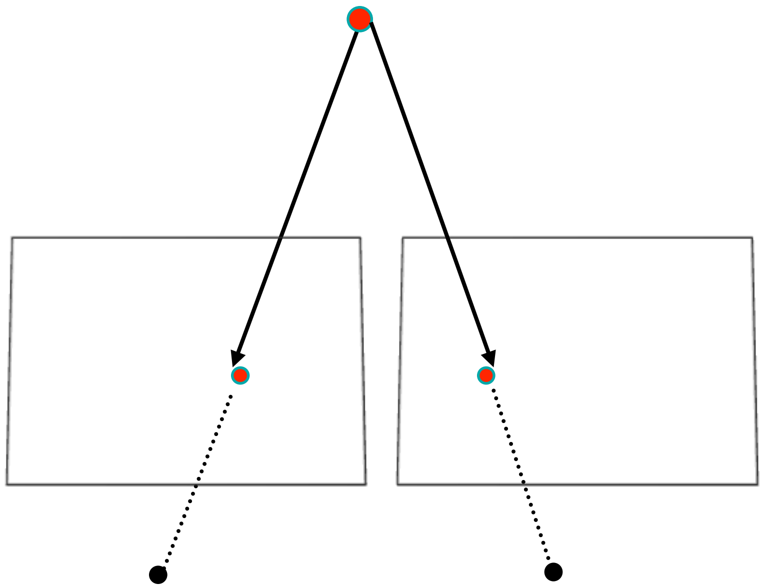
Slide credit: Kristen Grauman

Stereo Topics

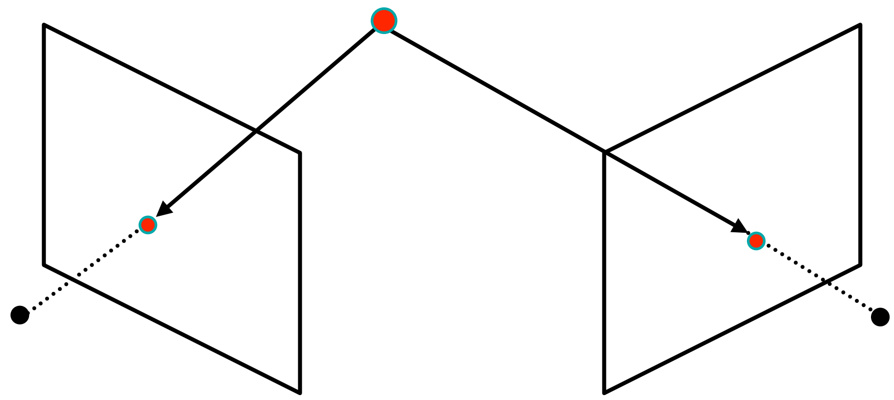
- Special, simple system, main idea
- More general camera conditions, epipolar constraints
  - epipolar geometry
  - epipolar algebra
- Image rectification
- Stereo matching (likelihood term)
- Stereo regularization (prior term)
- Inference
  - dynamic programming
  - graph cuts
- Structured light

General case, with calibrated cameras

- The two cameras need not have parallel optical axes.

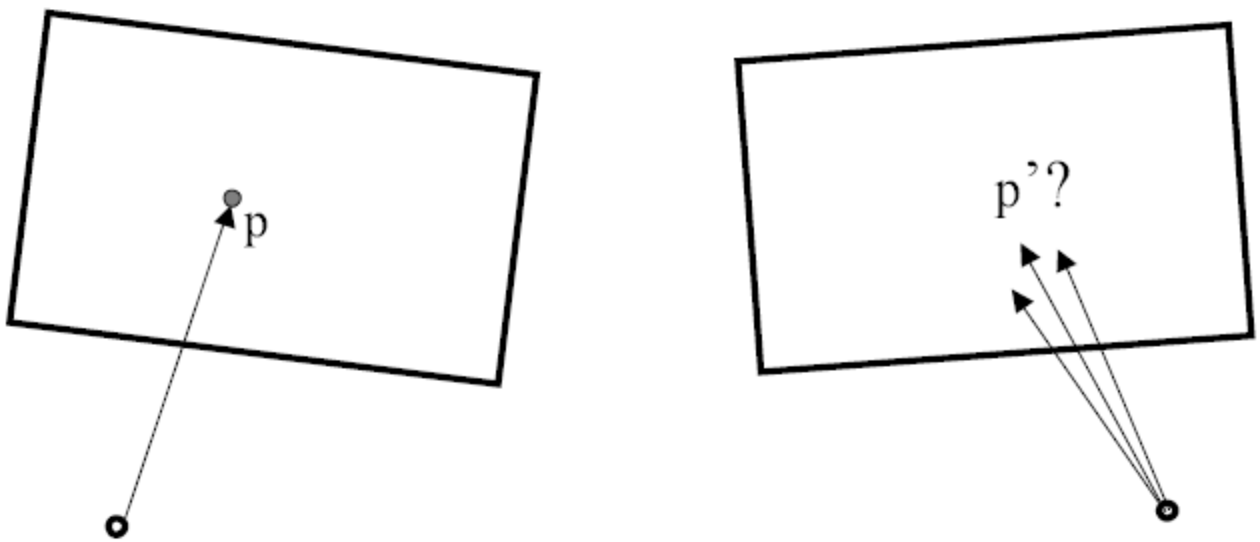


Vs.



Slide credit: Kristen Grauman

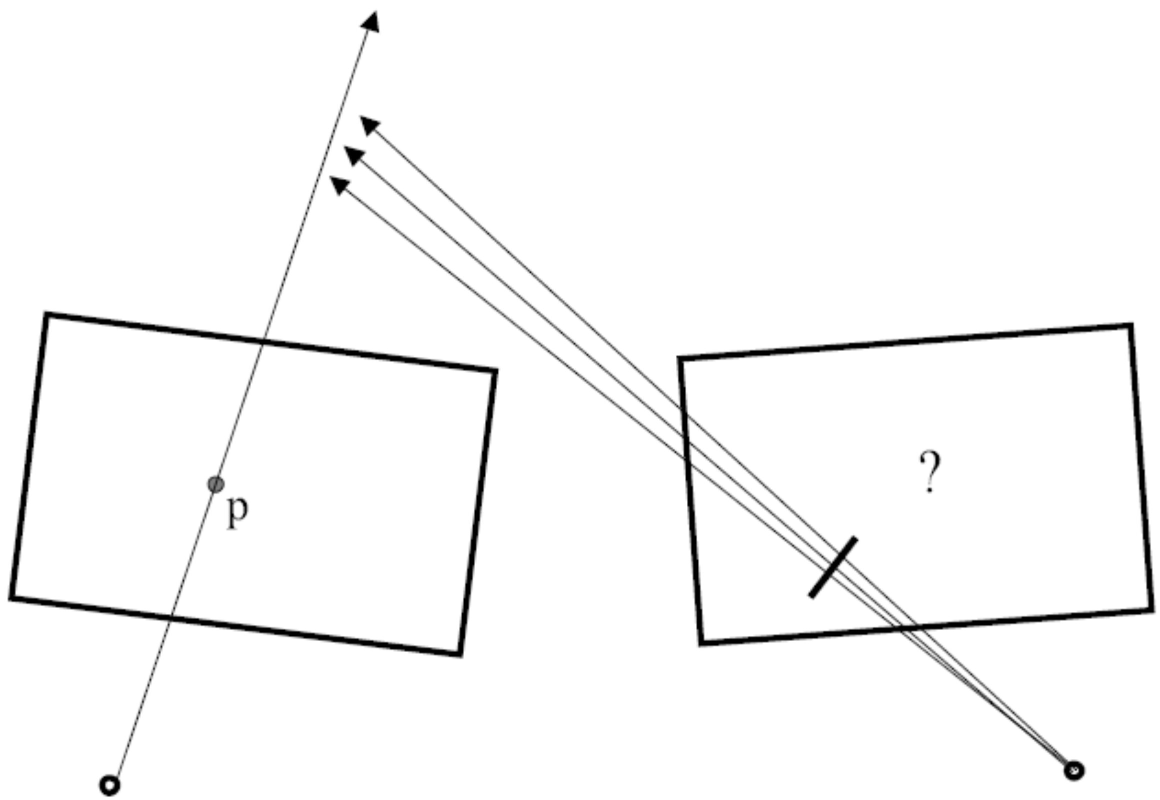
Stereo correspondence constraints



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

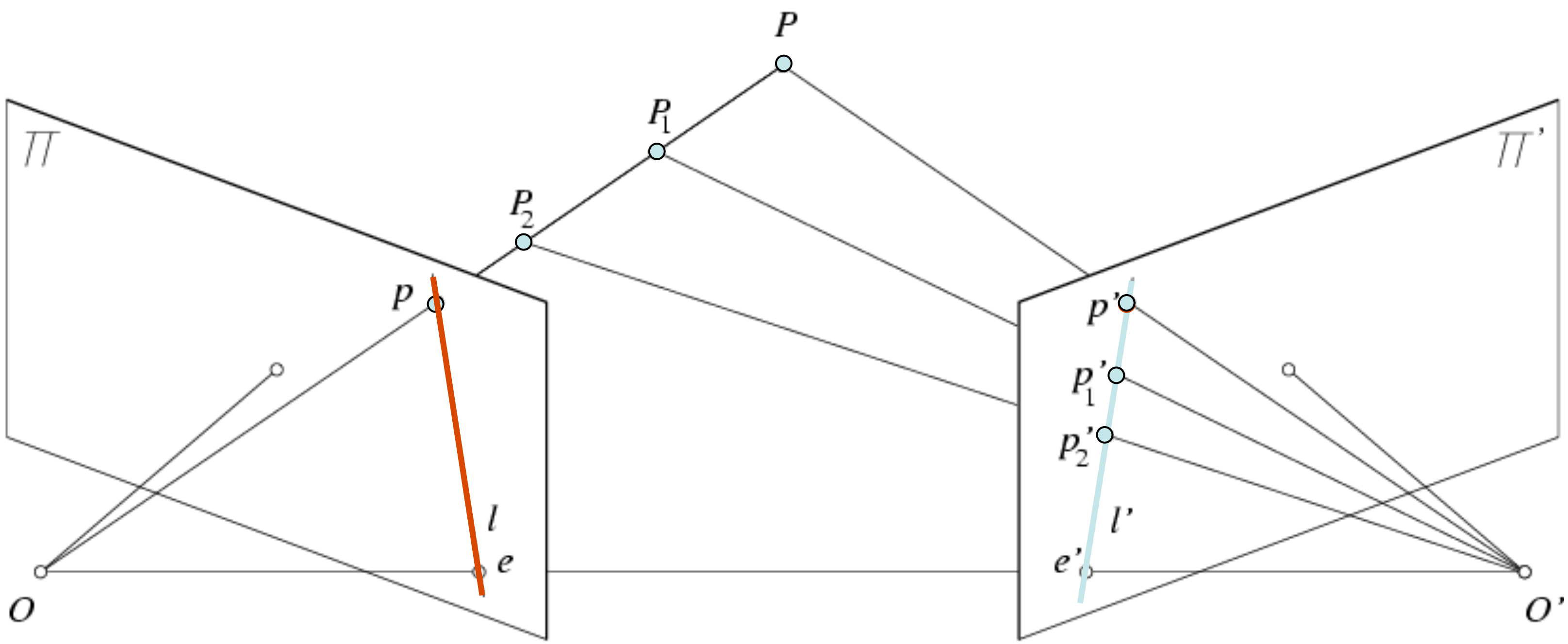
Slide credit: Kristen Grauman

Stereo correspondence constraints



Slide credit: Kristen Grauman

Epipolar constraint



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

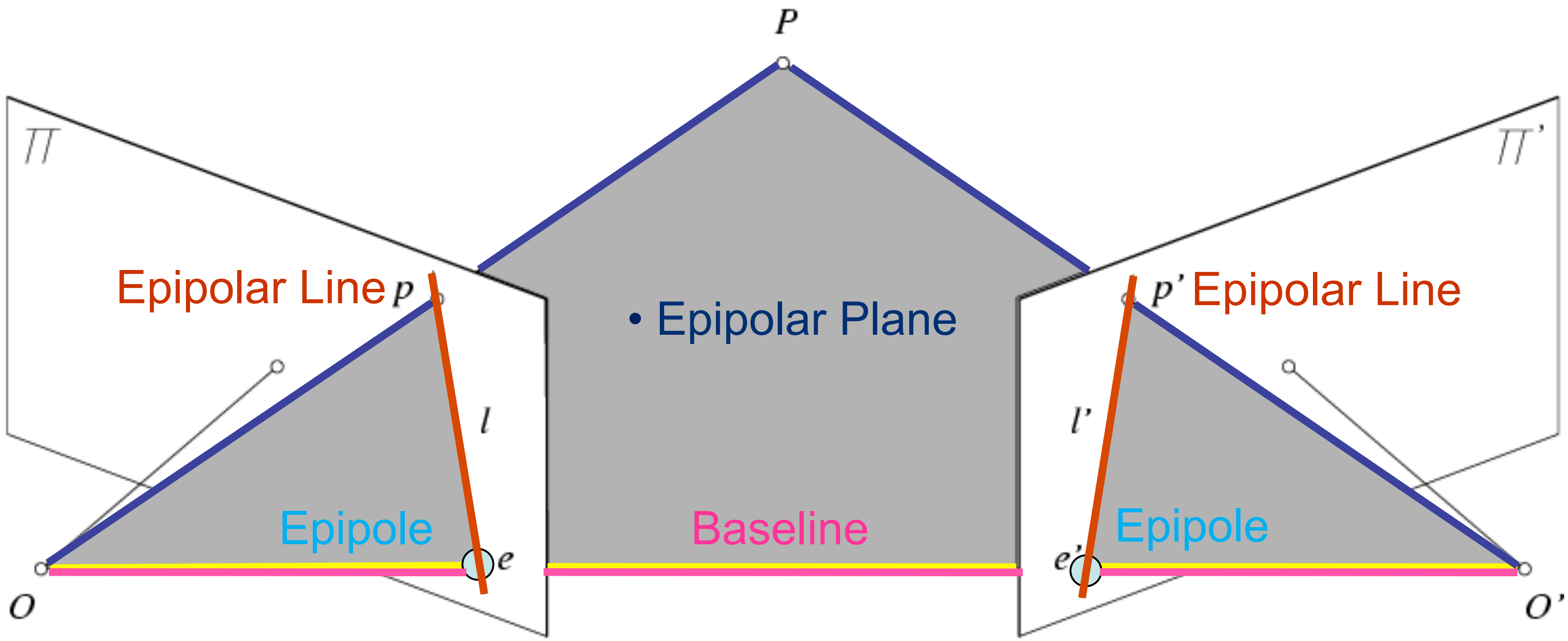
Epipolar constraint

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



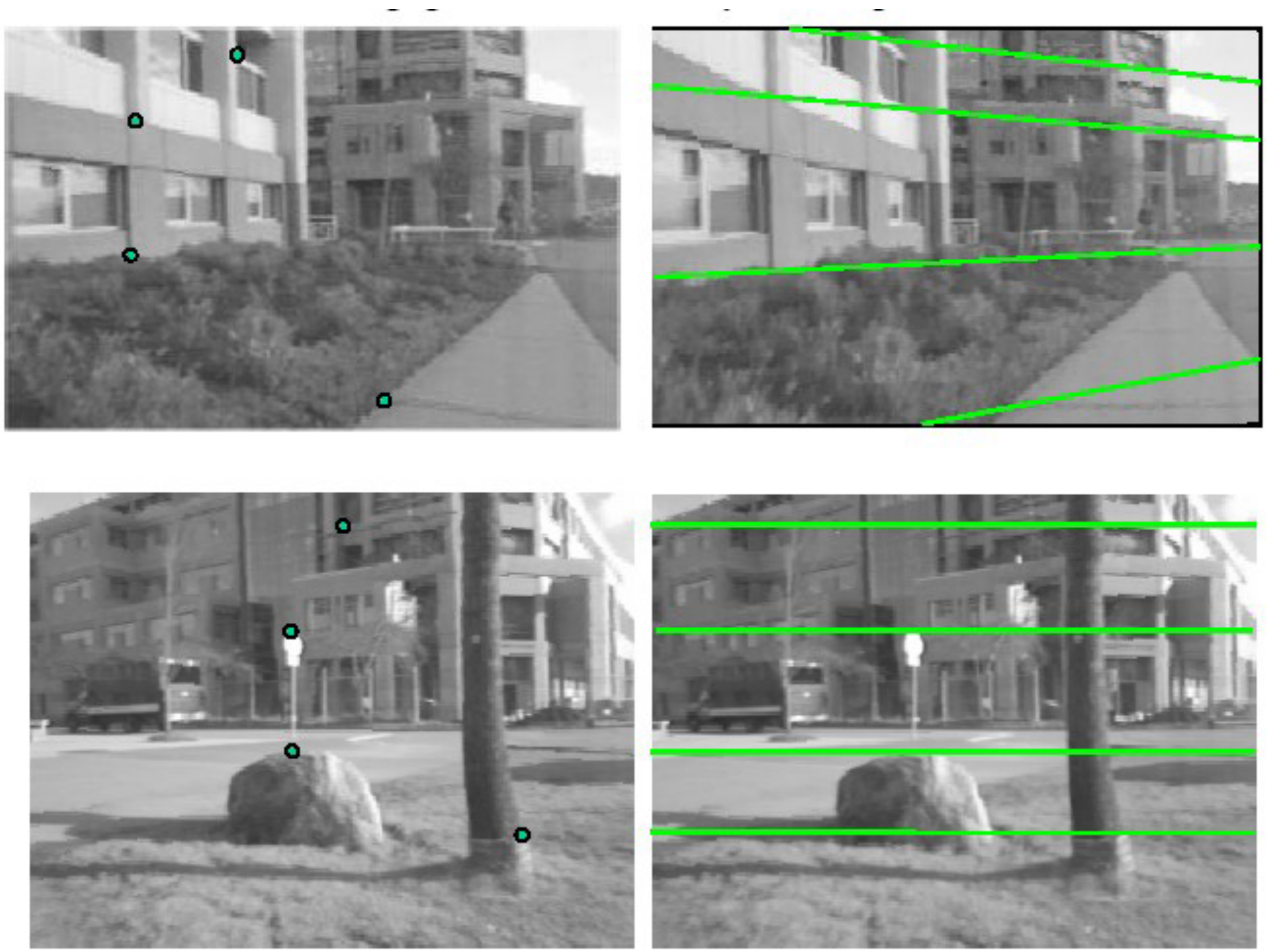
- **Baseline**: line joining the camera centers
- **Epipole**: point of intersection of baseline with the image plane
- **Epipolar plane**: plane containing baseline and world point
- **Epipolar line**: intersection of epipolar plane with the image plane

- All epipolar lines intersect at the epipole
- An epipolar plane intersects the left and right image planes in epipolar lines

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

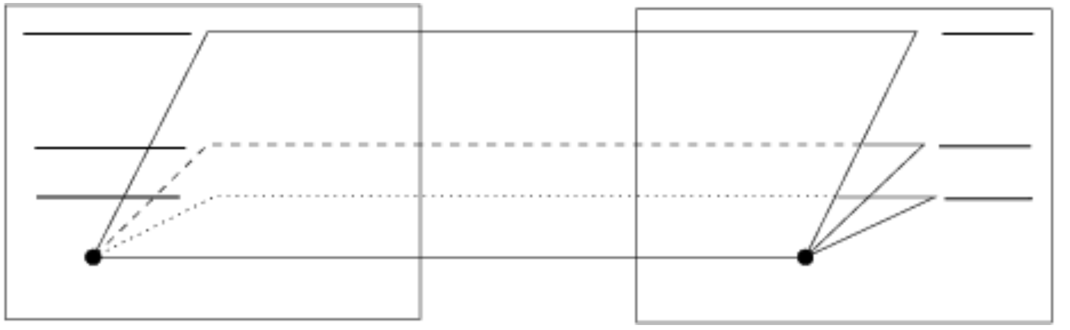
Slide credit: Kristen Grauman

Example



Slide credit: Kristen Grauman

Example: parallel cameras

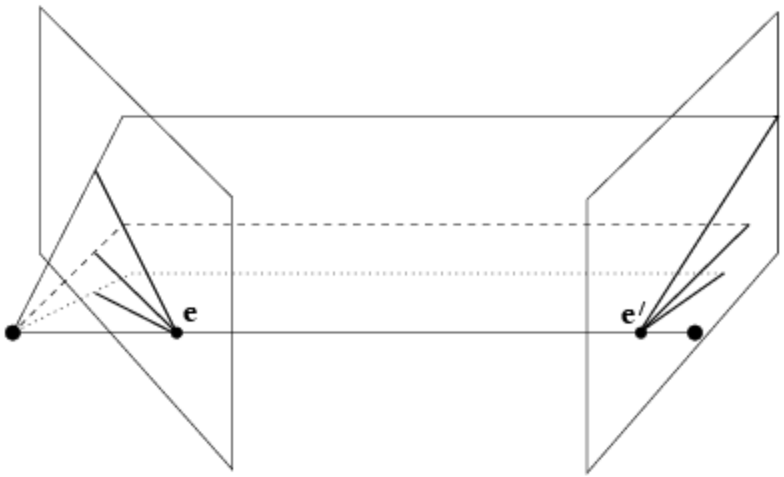


Where are the epipoles?

Figure from Hartley & Zisserman

Slide credit: Kristen Grauman

Example: converging cameras



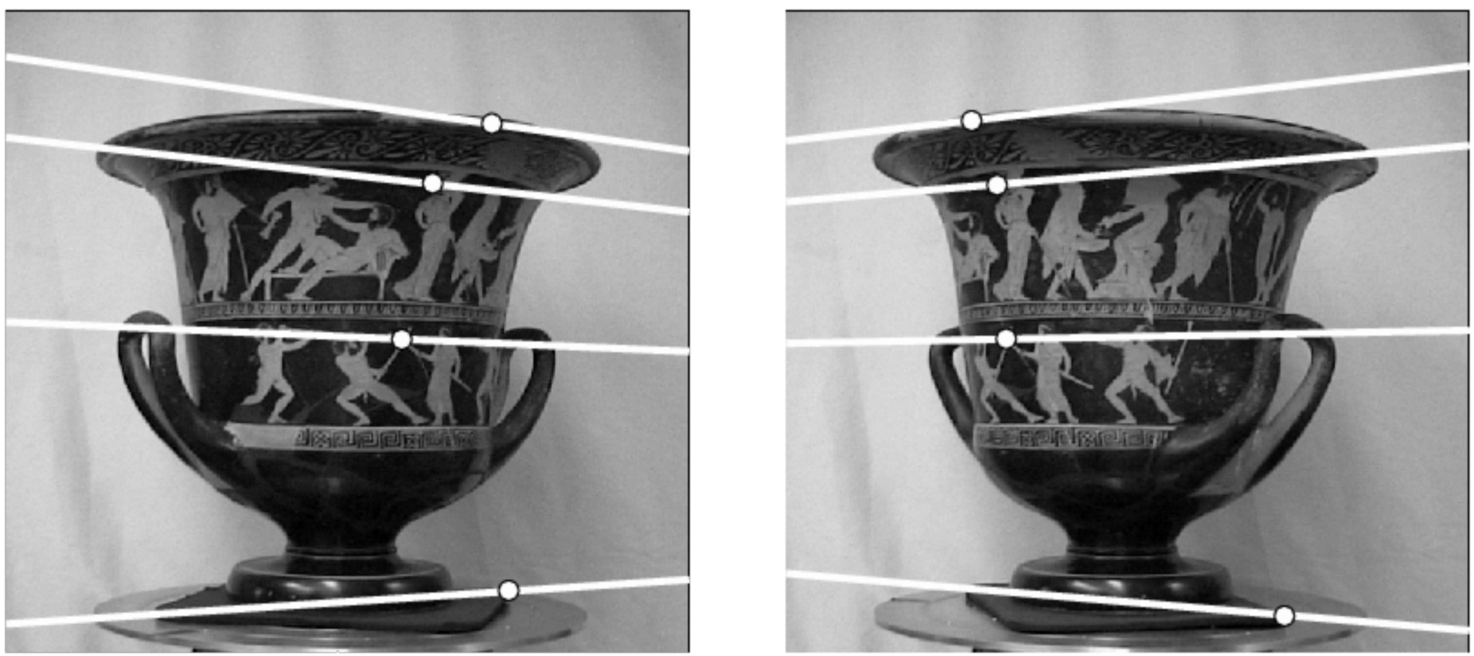


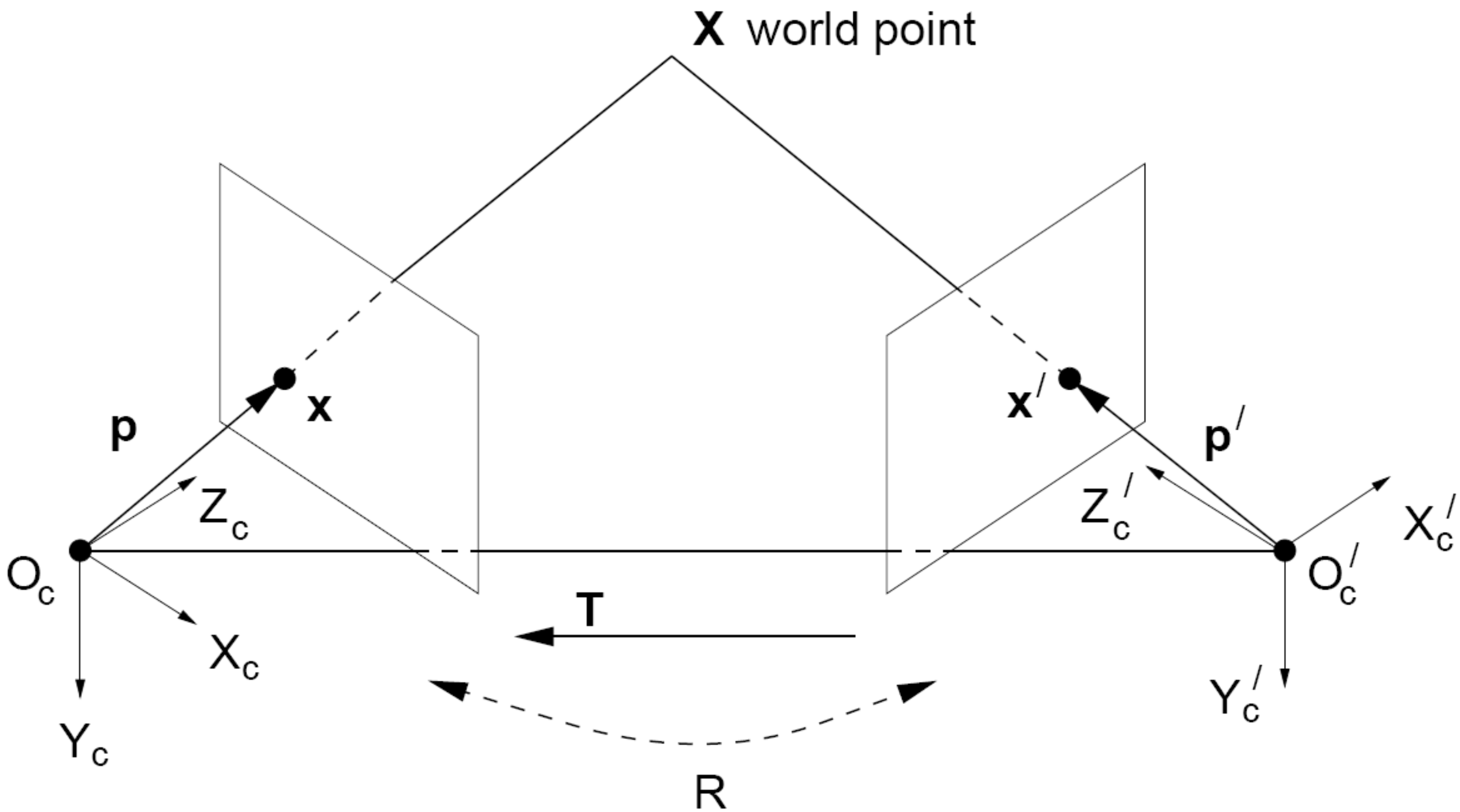
Figure from Hartley & Zisserman

Slide credit: Kristen Grauman

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

Slide credit: Kristen Grauman

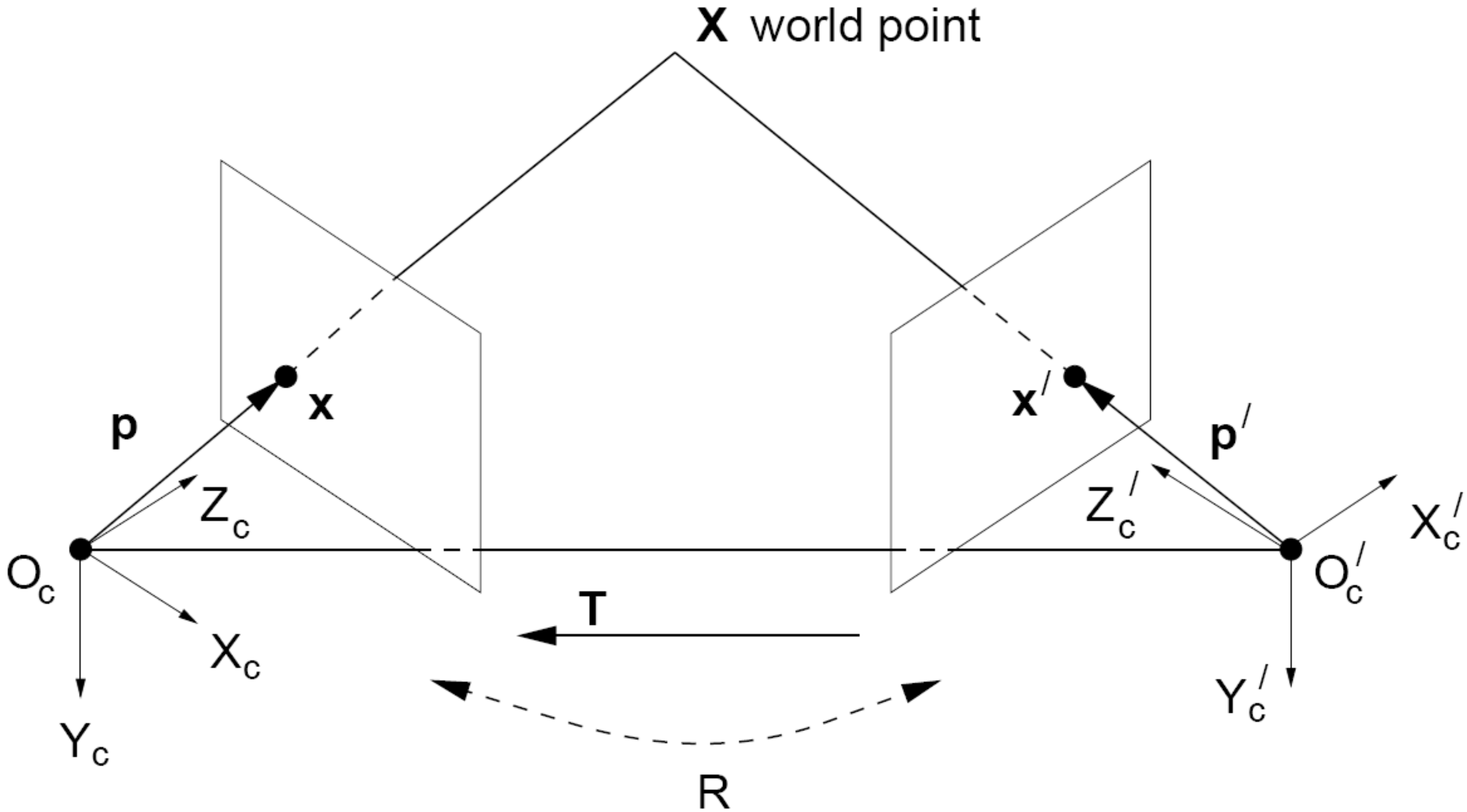
Stereo geometry, with calibrated cameras



Main idea

Slide credit: Kristen Grauman

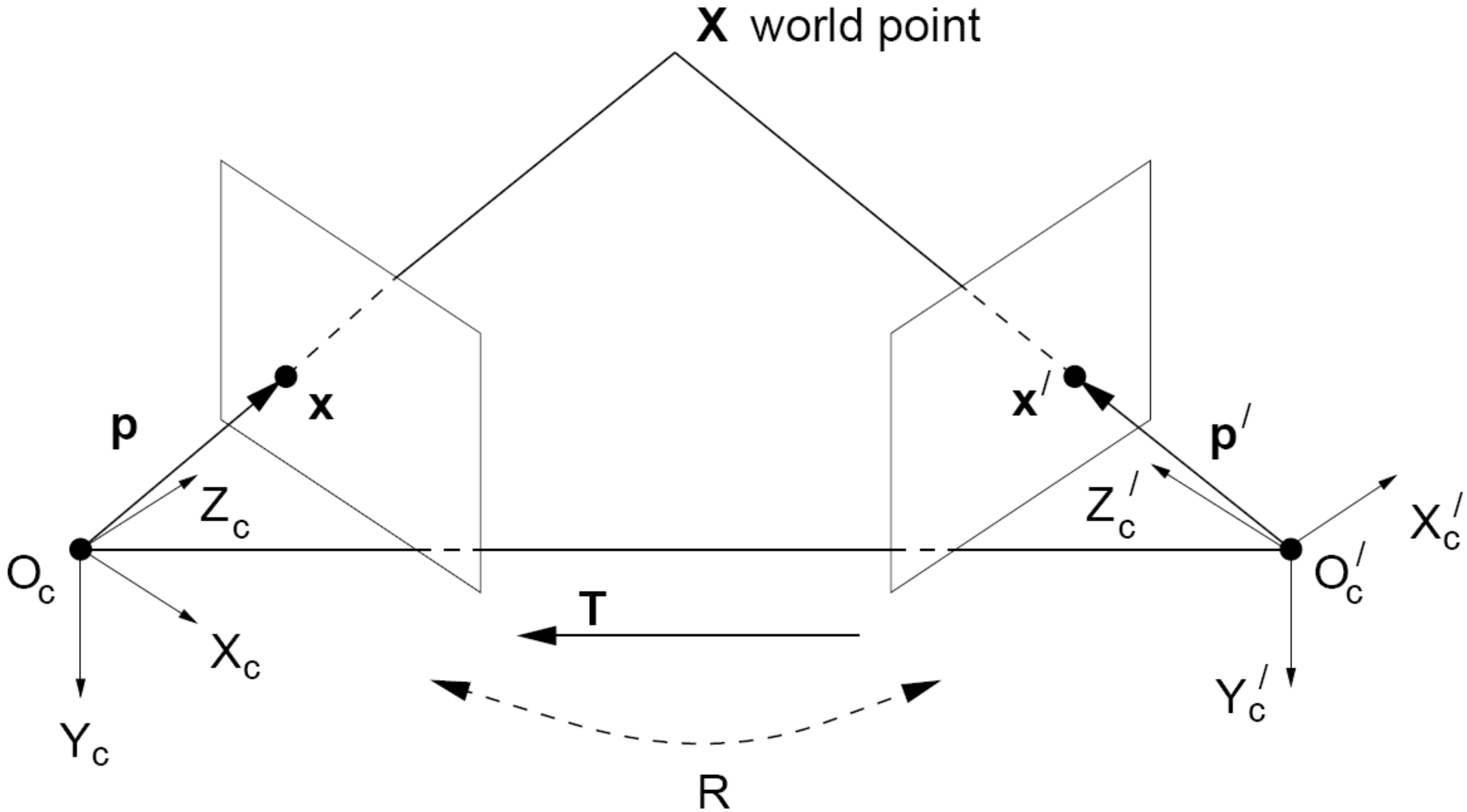
Stereo geometry, with calibrated cameras



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

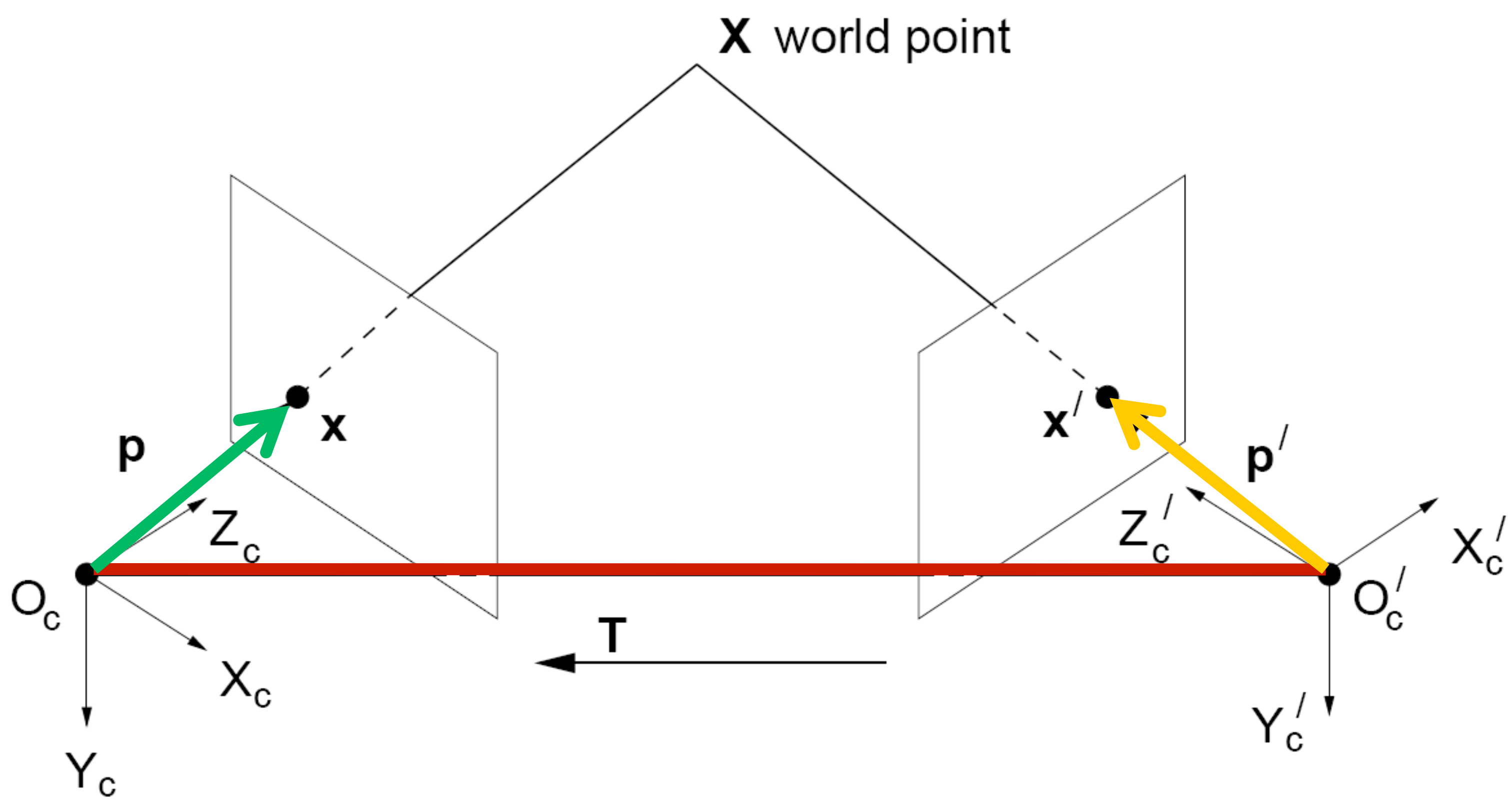


If the stereo rig is calibrated, we know :
how to rotate and translate camera reference frame 1 to get to camera reference frame 2.

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

Slide credit: Kristen Grauman

From geometry to algebra



$$\mathbf{X'} = \mathbf{R}\mathbf{X} + \mathbf{T}$$

$$\underbrace{\mathbf{T} \times \mathbf{X'}}_{\text{Normal to the plane}} = \mathbf{T} \times \mathbf{R}\mathbf{X}$$

$$\mathbf{X'} \cdot (\mathbf{T} \times \mathbf{X'}) = \mathbf{X'} \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X})$$

$$= 0$$

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

Another aside: Matrix form of cross product

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

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

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

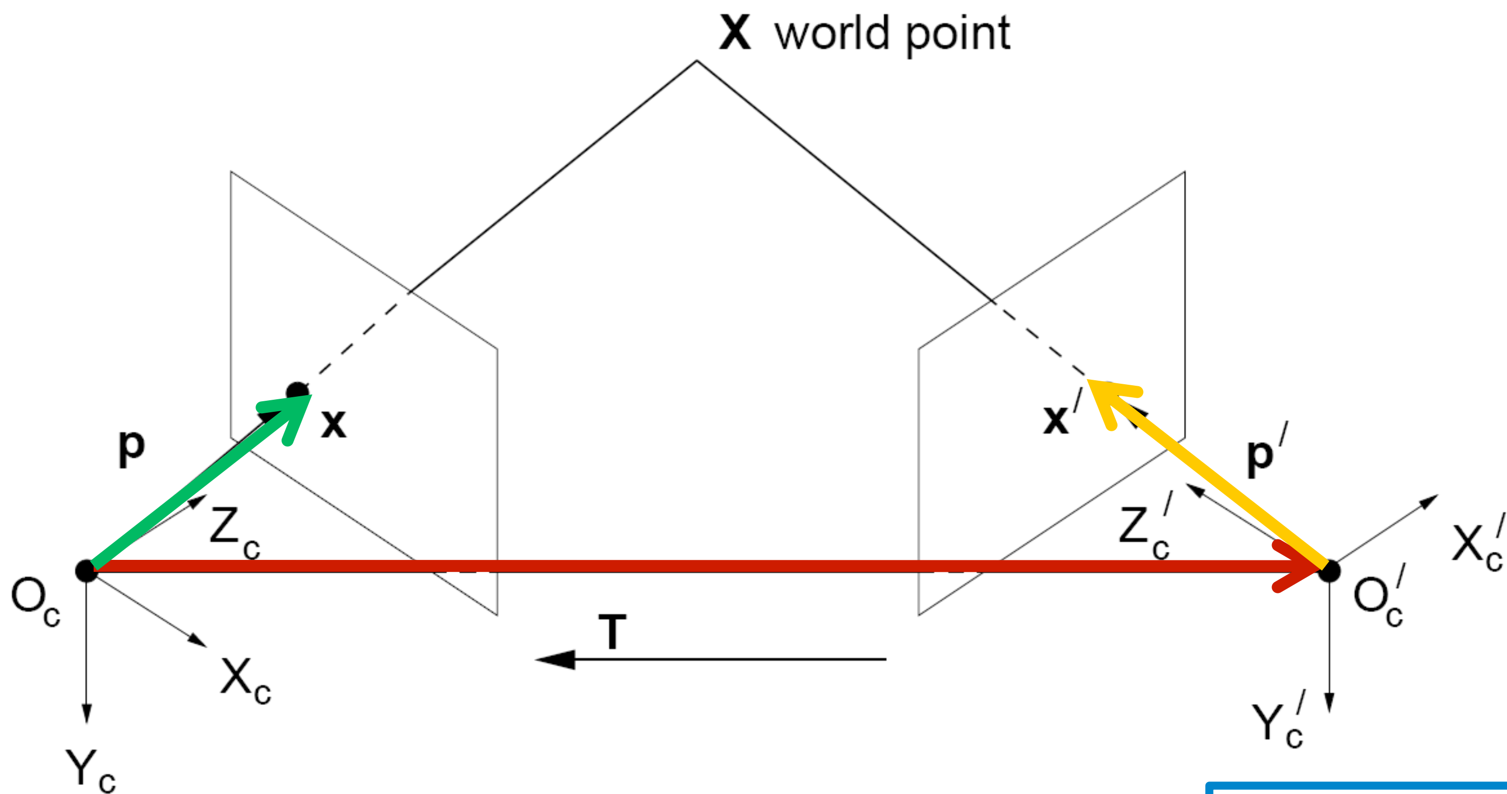
Can be expressed as a matrix multiplication.

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

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

Slide credit: Kristen Grauman

From geometry to algebra



$$\mathbf{X}' = \mathbf{R}\mathbf{X} + \mathbf{T}$$

$$\underbrace{\mathbf{T} \times \mathbf{X}'}_{\text{Normal to the plane}} = \mathbf{T} \times \mathbf{R}\mathbf{X} + \mathbf{T} \times \mathbf{T}$$

$$= \mathbf{T} \times \mathbf{R}\mathbf{X}$$

$$\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{X}') = \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X})$$

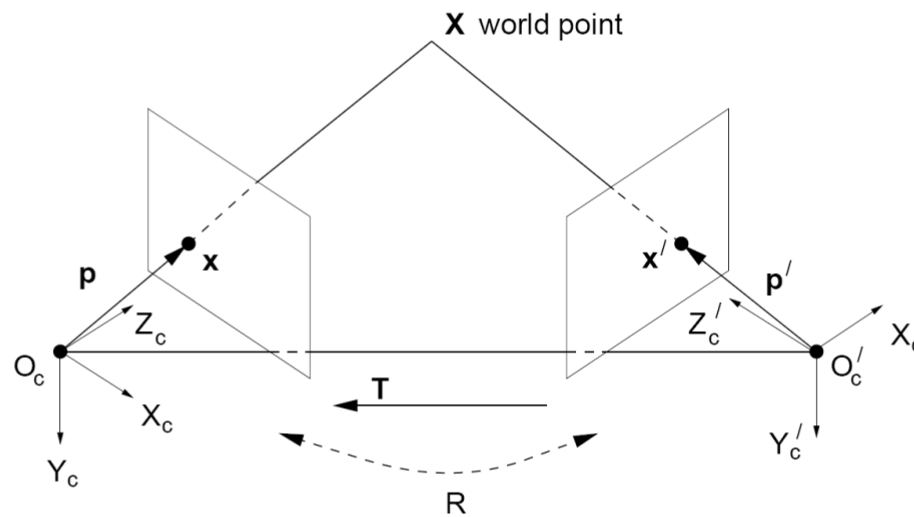
$$= 0$$

Slide credit: Kristen Grauman

Essential matrix

$$\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X}) = 0$$

$$\mathbf{X}' \cdot (\mathbf{T}_x \; \mathbf{R}\mathbf{X}) = 0$$



Let $\mathbf{E} = \mathbf{T}_x\mathbf{R}$

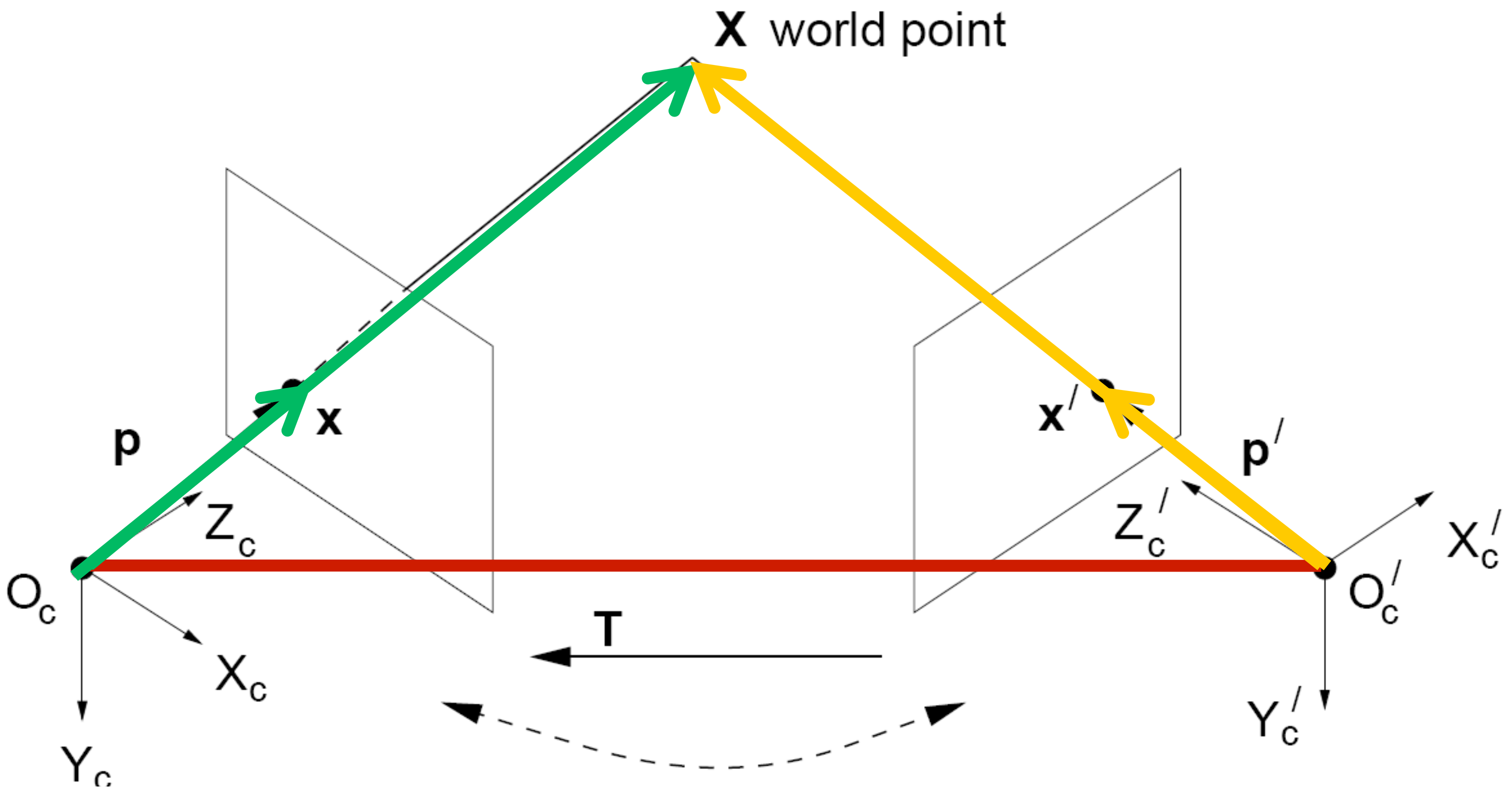
$$\mathbf{X}'^{T}\mathbf{E}\mathbf{X} = 0$$

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

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

Note: these points are in camera coordinate systems.

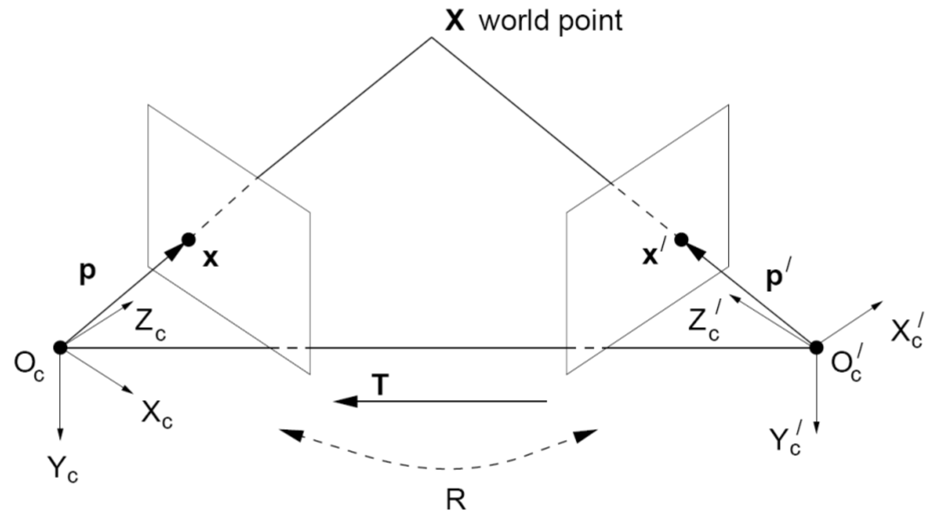
x and x’   are scaled versions of X and X’



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

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

Let $E = T'_x R$



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

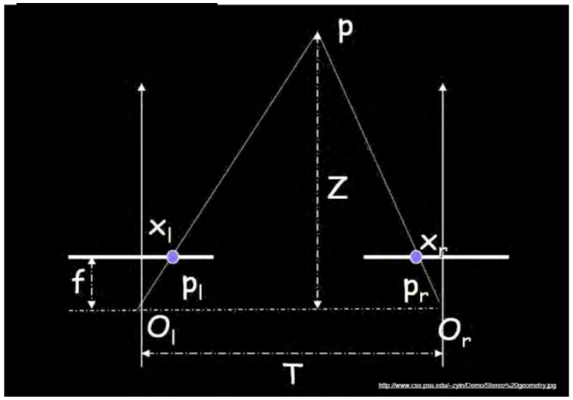
$$x'^T E x = 0$$ pts x and x' in the image planes are scaled versions of X and X'.

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

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



$\mathbf{R} =$

$\mathbf{T} =$

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

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

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

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

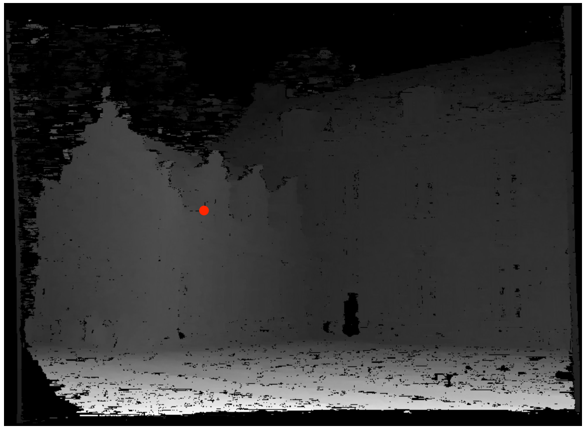
For the parallel cameras, image of any point must lie on same horizontal line in each image plane.

Slide credit: Kristen Grauman

image I(x,y)

Disparity map D(x,y)

image I´(x´,y´)



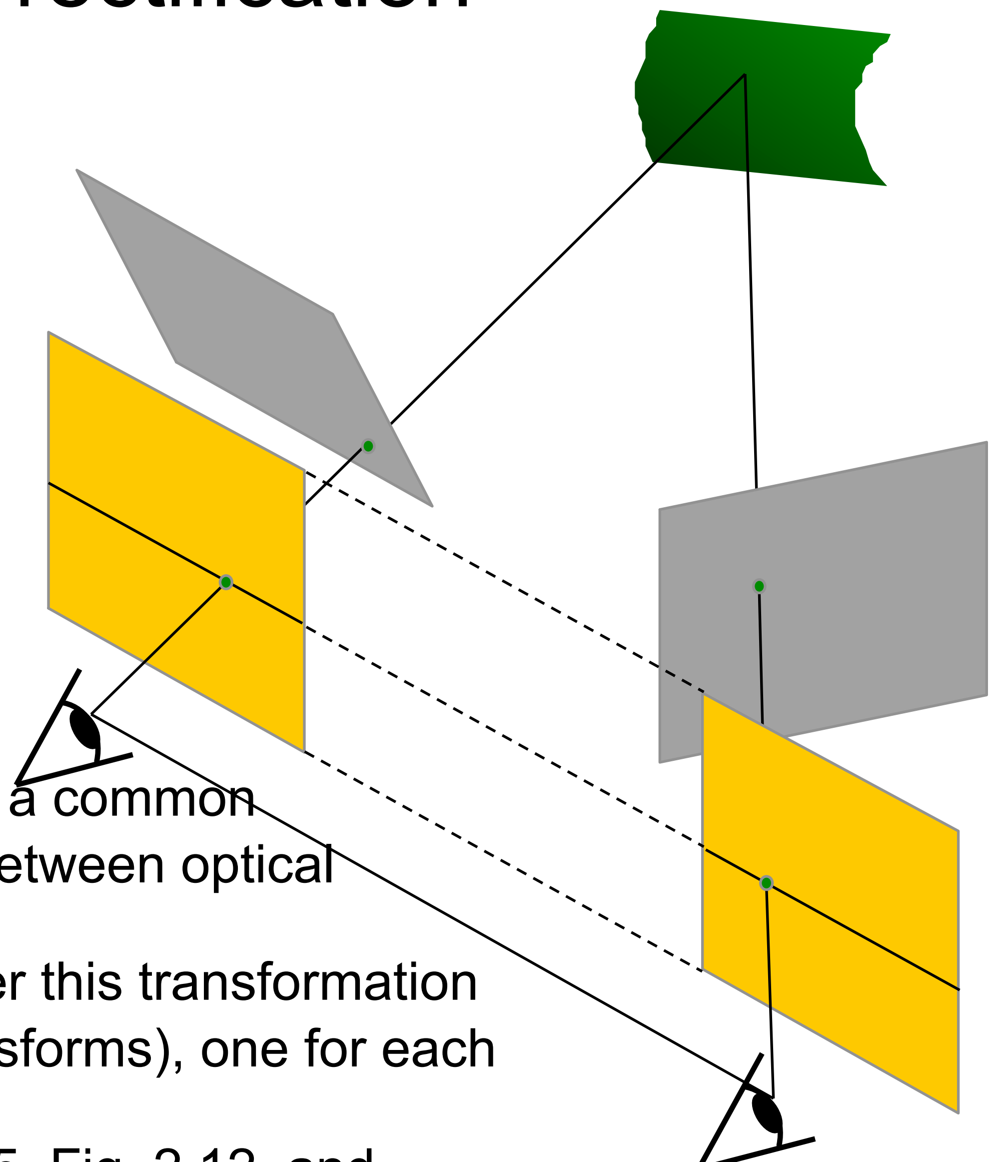
$$(x', y') = (x + D(x,y), y)$$

What about when cameras' optical axes are not parallel?

Slide credit: Kristen Grauman

Stereo image rectification

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

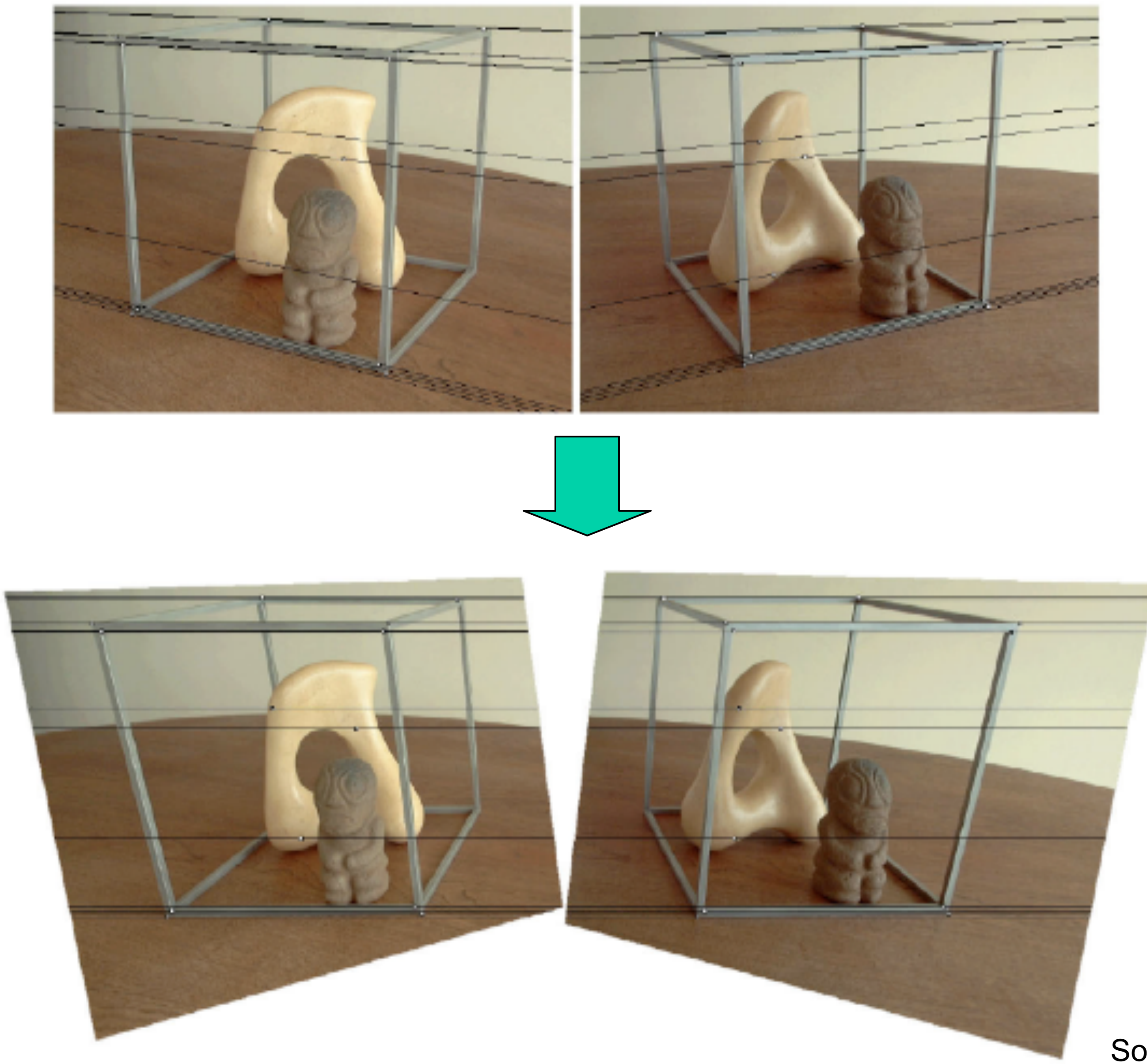


- Reproject image planes onto a common plane parallel to the line between optical centers
- Pixel motion is horizontal after this transformation
- Two homographies (3x3 transforms), one for each input image reprojection
- See Szeliski book, Sect. 2.1.5, Fig. 2.12, and "Mapping from one camera to another"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

Adapted from Li Zhang

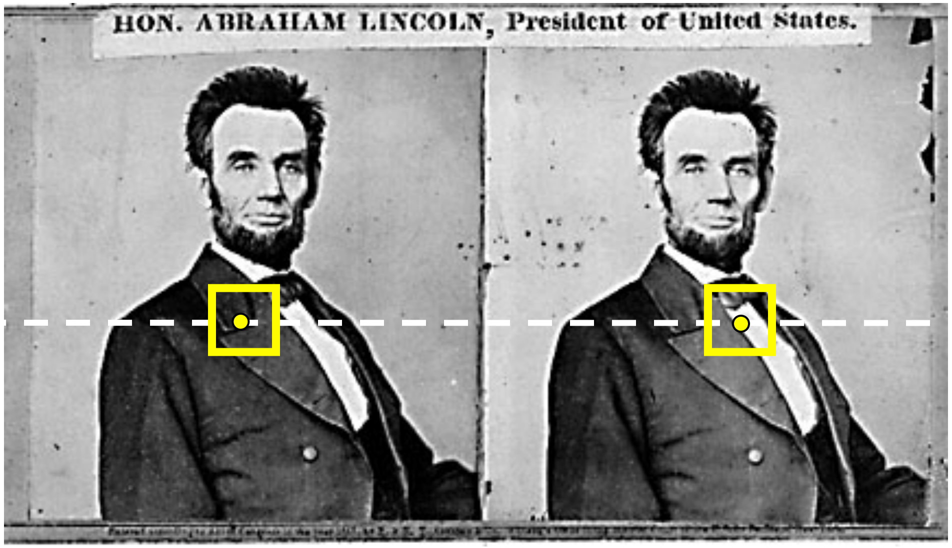
Slide credit: Kristen Grauman

Stereo image rectification: example



Source: Alyosha Efros

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)

Slide credit: Rick Szeliski

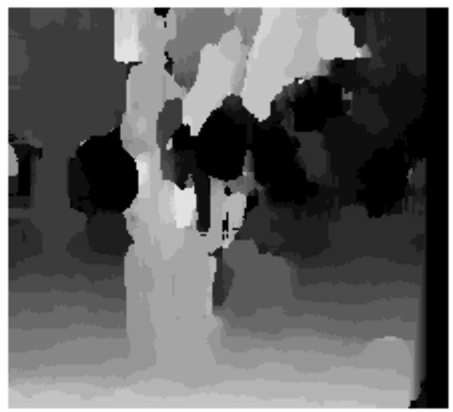
How big should the neighborhood be?

Neighborhood size

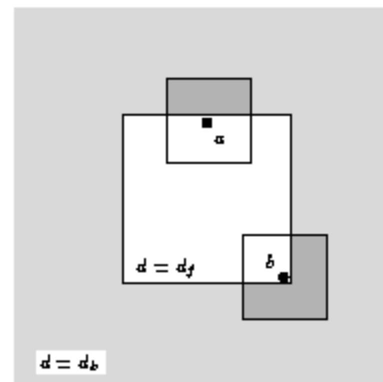
Smaller neighborhood: more details

Larger neighborhood: fewer isolated mistakes

w = 3



w = 20



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

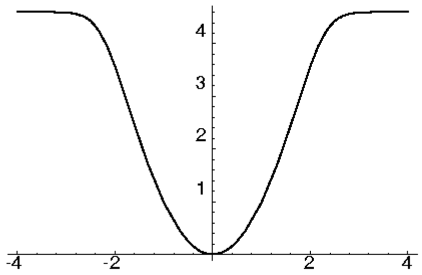
Local evidence framework

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



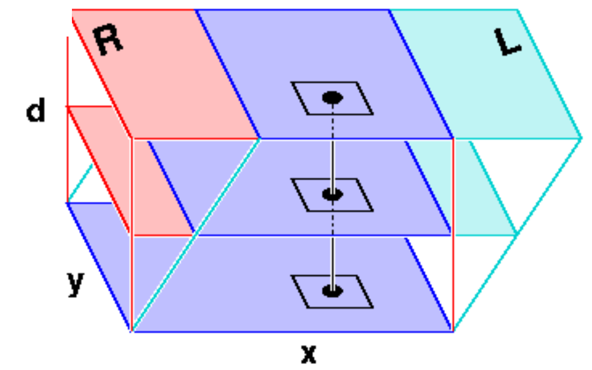
Can also use alternative match criteria

Slide credit: Rick Szeliski

Local evidence framework

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…

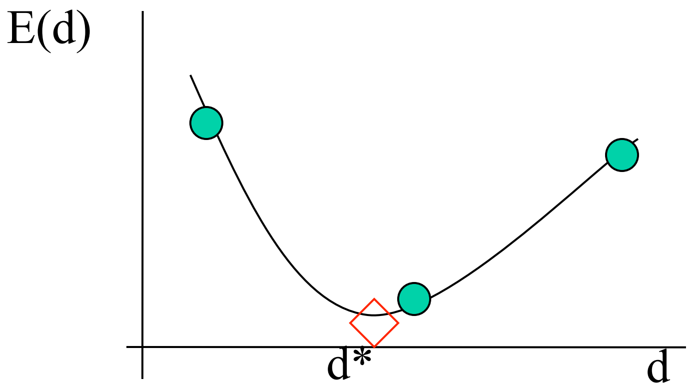
Slide credit: Rick Szeliski

Local evidence framework

Choose winning disparity at each pixel

$$d(x, y) = \arg\min_{d} E(x, y; d)$$

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:

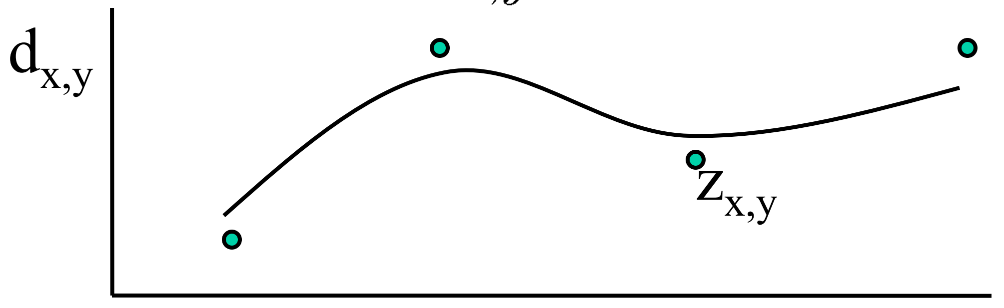
- narrow baseline $\Rightarrow$ noisy estimates
- fails in textureless areas
- gets confused near occlusion boundaries

Slide credit: Rick Szeliski

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

Dynamic programming

Evaluate best cumulative cost at each pixel

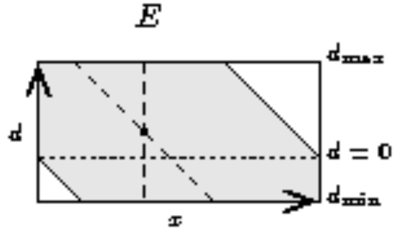
$$
\begin{aligned}
E_{\text{total}}(\mathbf{d}) &= E_{\text{data}}(\mathbf{d}) + \lambda E_{\text{smoothness}}(\mathbf{d}) \\
E_{\text{data}}(\mathbf{d}) &= \sum_{x,y} (d_{x,y} - z_{x,y})^2 \\
E_{\text{smoothness}}(\mathbf{d}) &= \sum_{x,y} |d_{x,y} - d_{x-1,y}|
\end{aligned}
$$

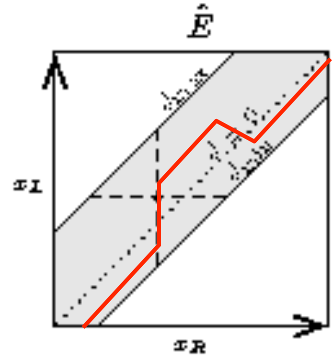
Slide credit: Rick Szeliski

Dynamic programming

1-D cost function

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





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

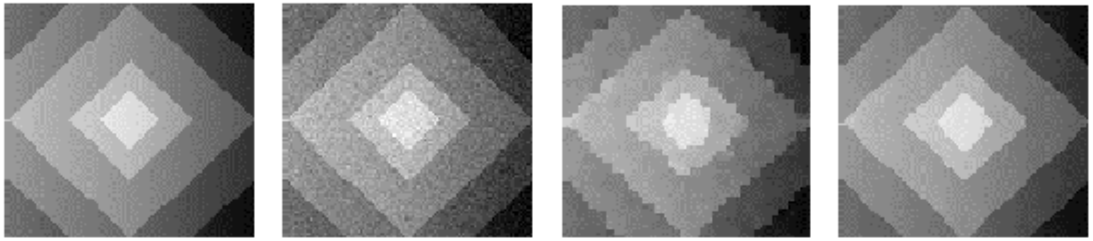
Graph cuts

Solution technique for general 2D problem

$$E_{\text{total}}(\mathbf{d}) = E_{\text{data}}(\mathbf{d}) + \lambda E_{\text{smoothness}}(\mathbf{d})$$

$$E_{\text{data}}(\mathbf{d}) = \sum_{x,y} f_{x,y}(d_{x,y})$$

$$E_{\text{smoothness}}(\mathbf{d}) = \sum_{x,y} \rho(d_{x,y} - d_{x-1,y}) + \sum_{x,y} \rho(d_{x,y} - d_{x,y-1})$$



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

Graph cuts

α-β swap

expansion

modify smoothness penalty based on edges

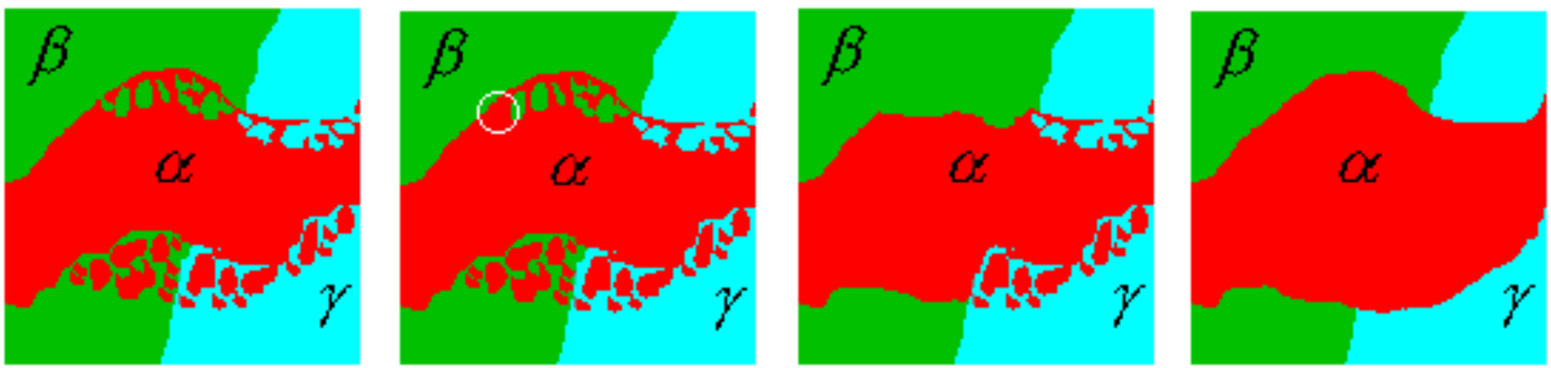
compute best possible match within integer disparity

Slide credit: Rick Szeliski

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

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 $p_P(d)$

Measurement model $p_M(I_L, I_R | d)$

Posterior model

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

Maximum a Posteriori (MAP estimate):

maximize $p_M(d \mid I_L, I_R)$

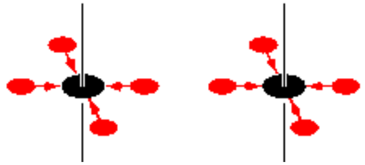
Slide credit: Rick Szeliski

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?

Principled way of determining cost function

Explicit model of noise and prior knowledge

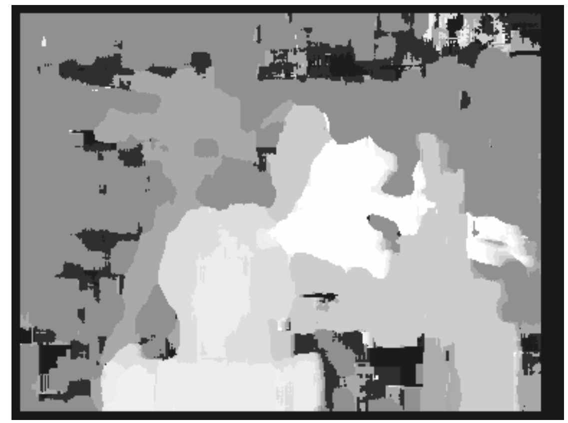
Admits a wide variety of optimization algorithms:

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

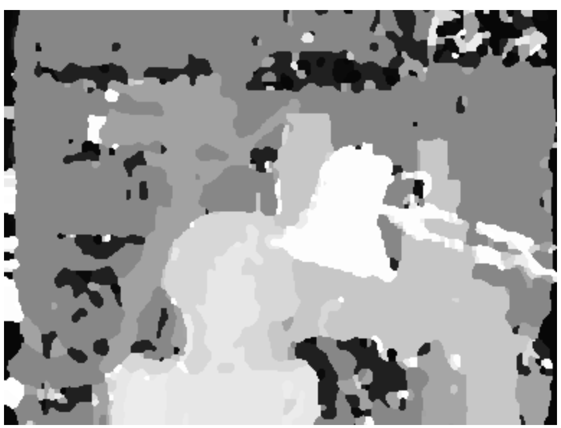
Slide credit: Rick Szeliski

Depth Map Results

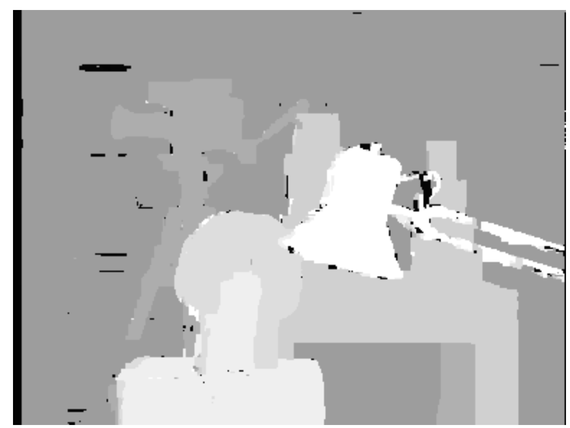
Input image



Sum Abs Diff



Mean field



Graph cuts

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...

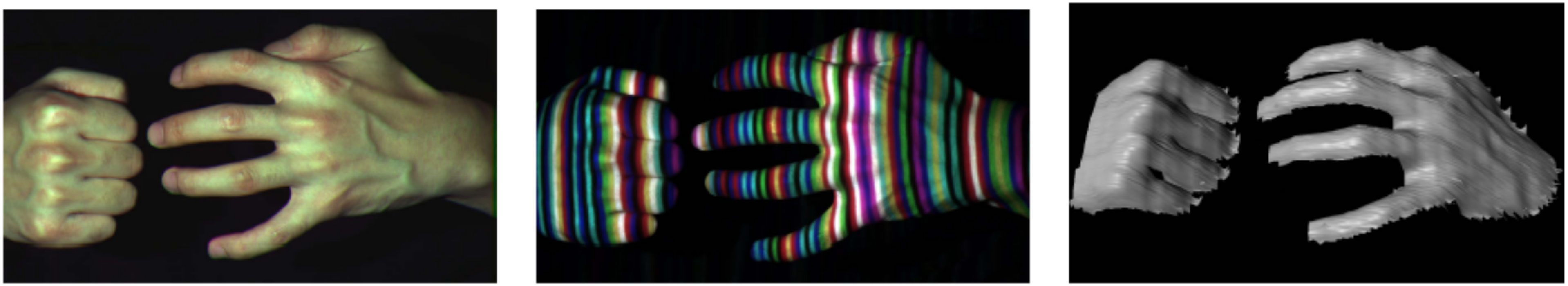
Sort by nonocc | Sort by all | Sort by disc

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

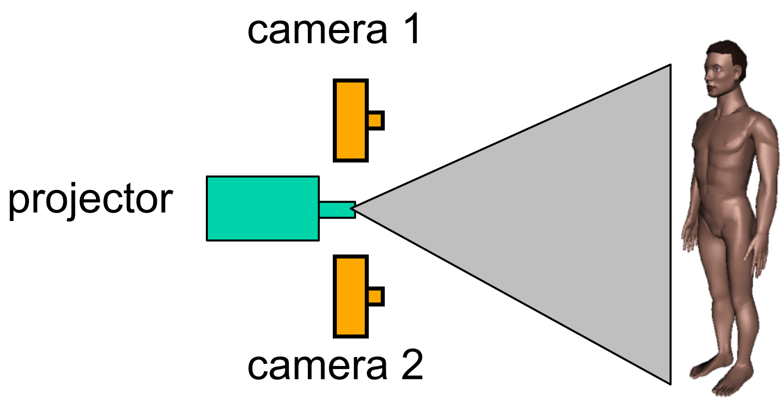
Stereo Topics

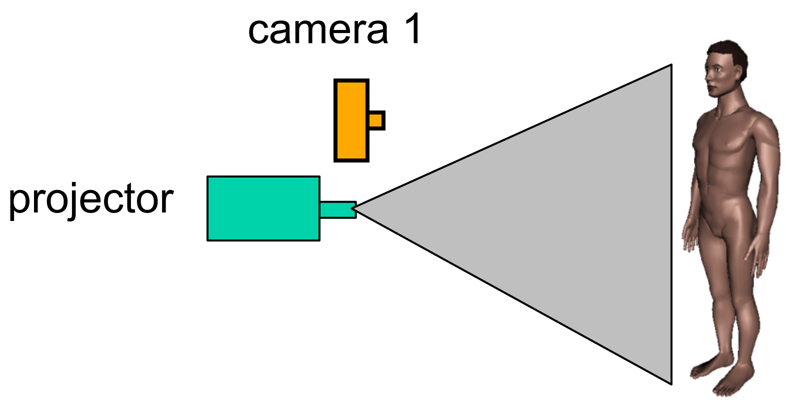
- Special, simple system, main idea
- More general camera conditions, epipolar constraints
  - epipolar geometry
  - epipolar algebra
- Image rectification
- Stereo matching (likelihood term)
- Stereo regularization (prior term)
- Inference
  - dynamic programming
  - graph cuts
- Structured light

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

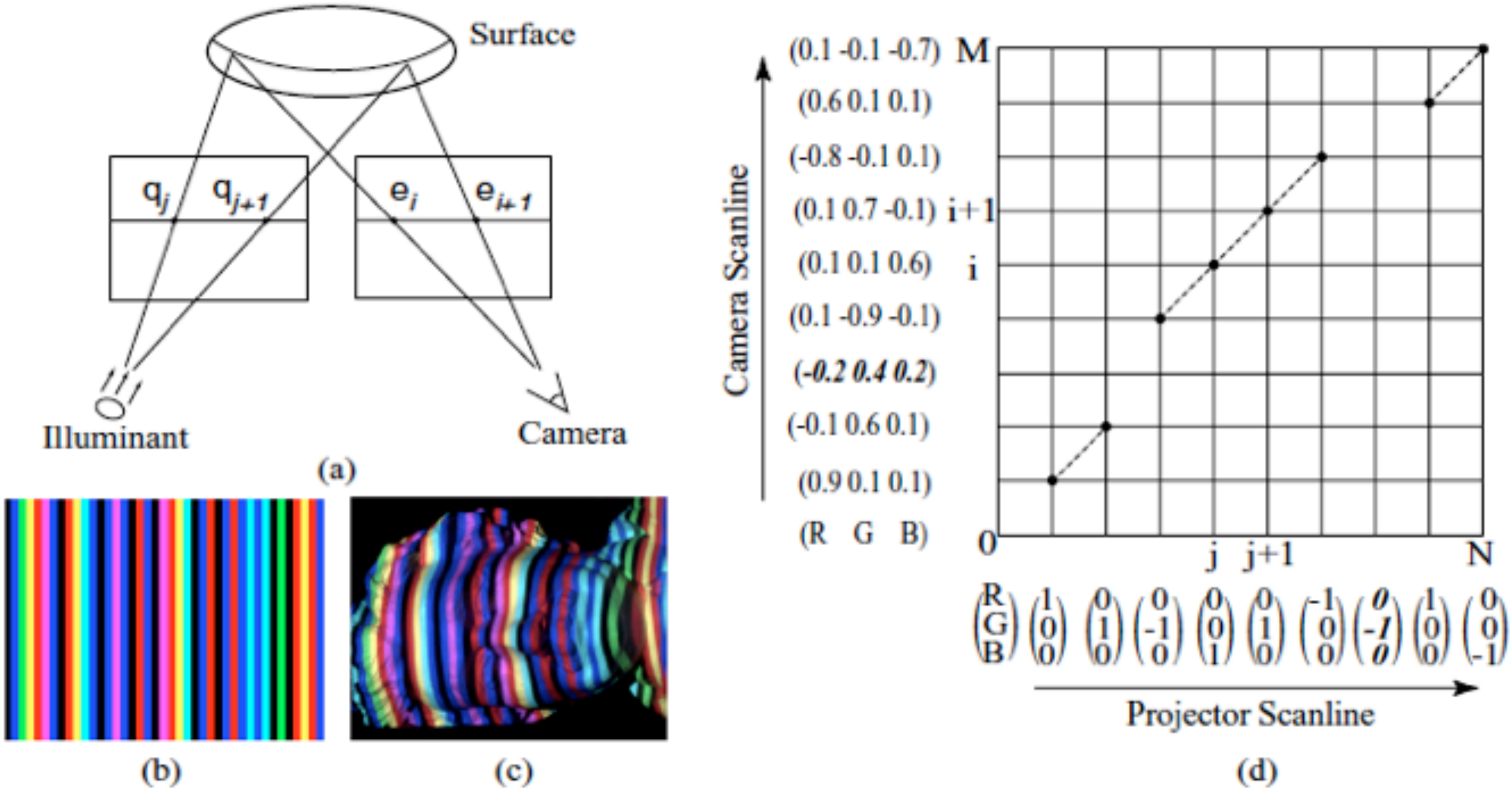
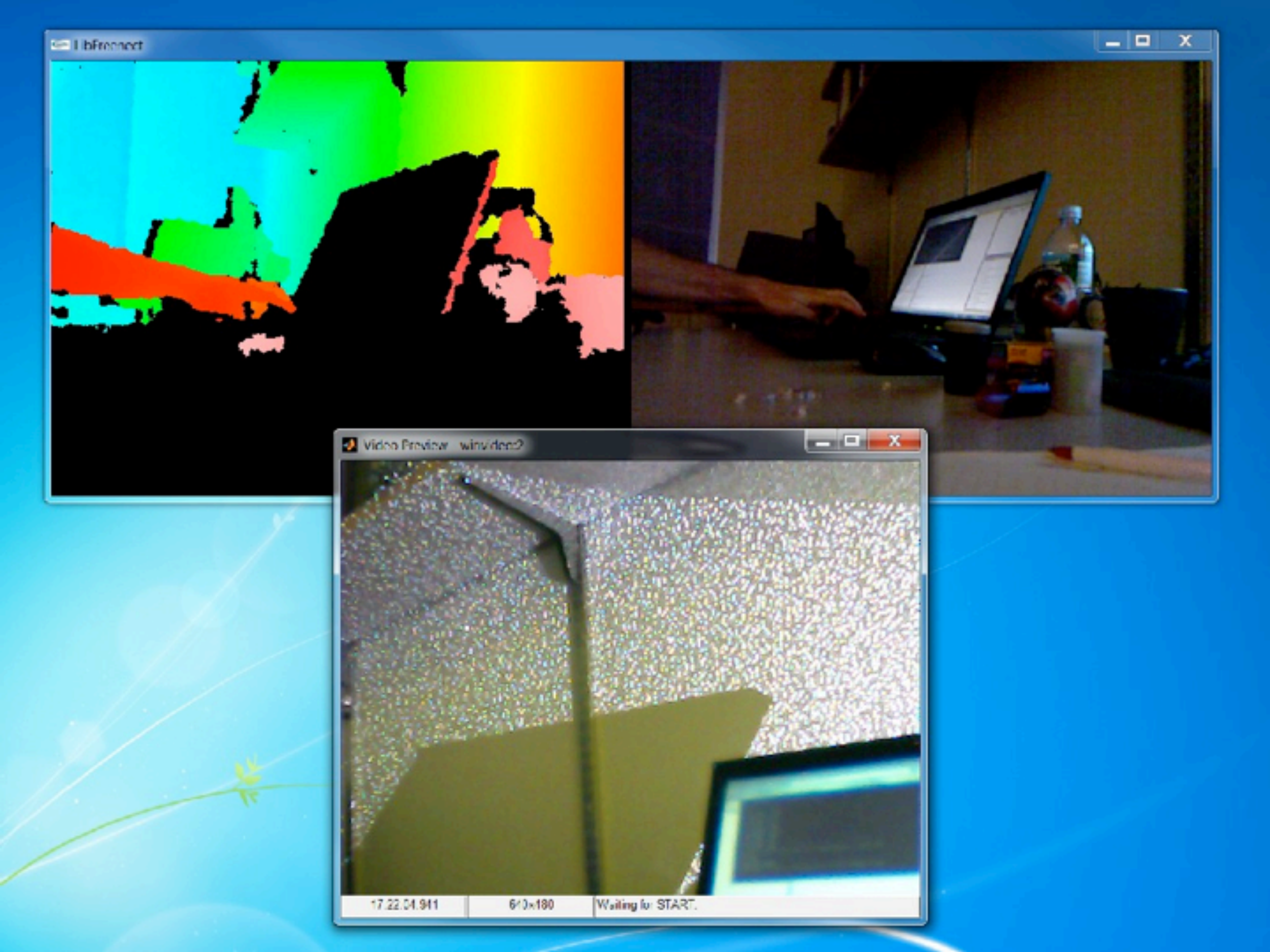


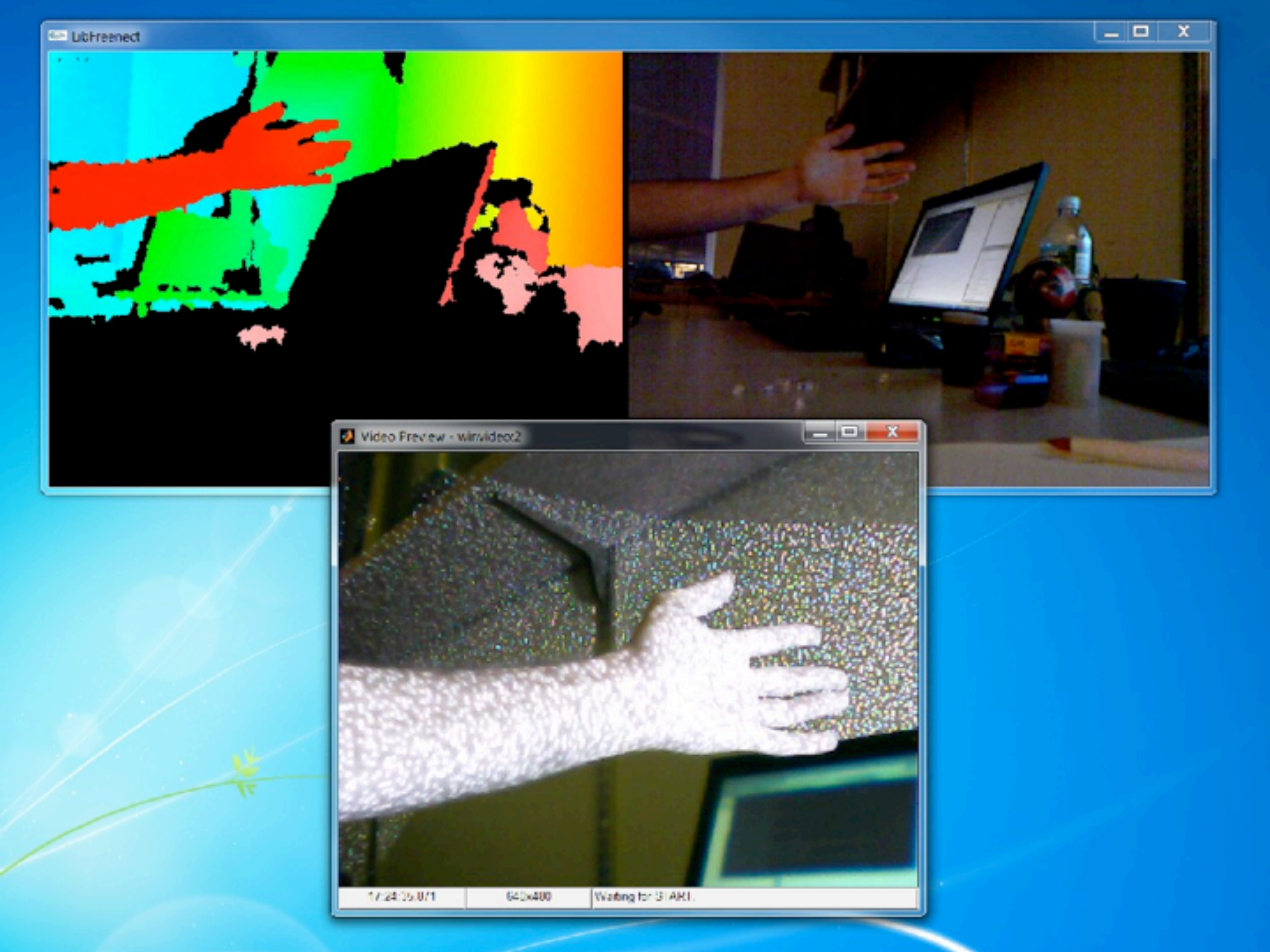
Figure 2. Summary of the one-shot method. (a) In optical triangulation, an illumination pattern is projected onto an object and the reflected light is captured by a camera. The 3D point is reconstructed from the relative displacement of a point in the pattern and image. If the image planes are rectified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

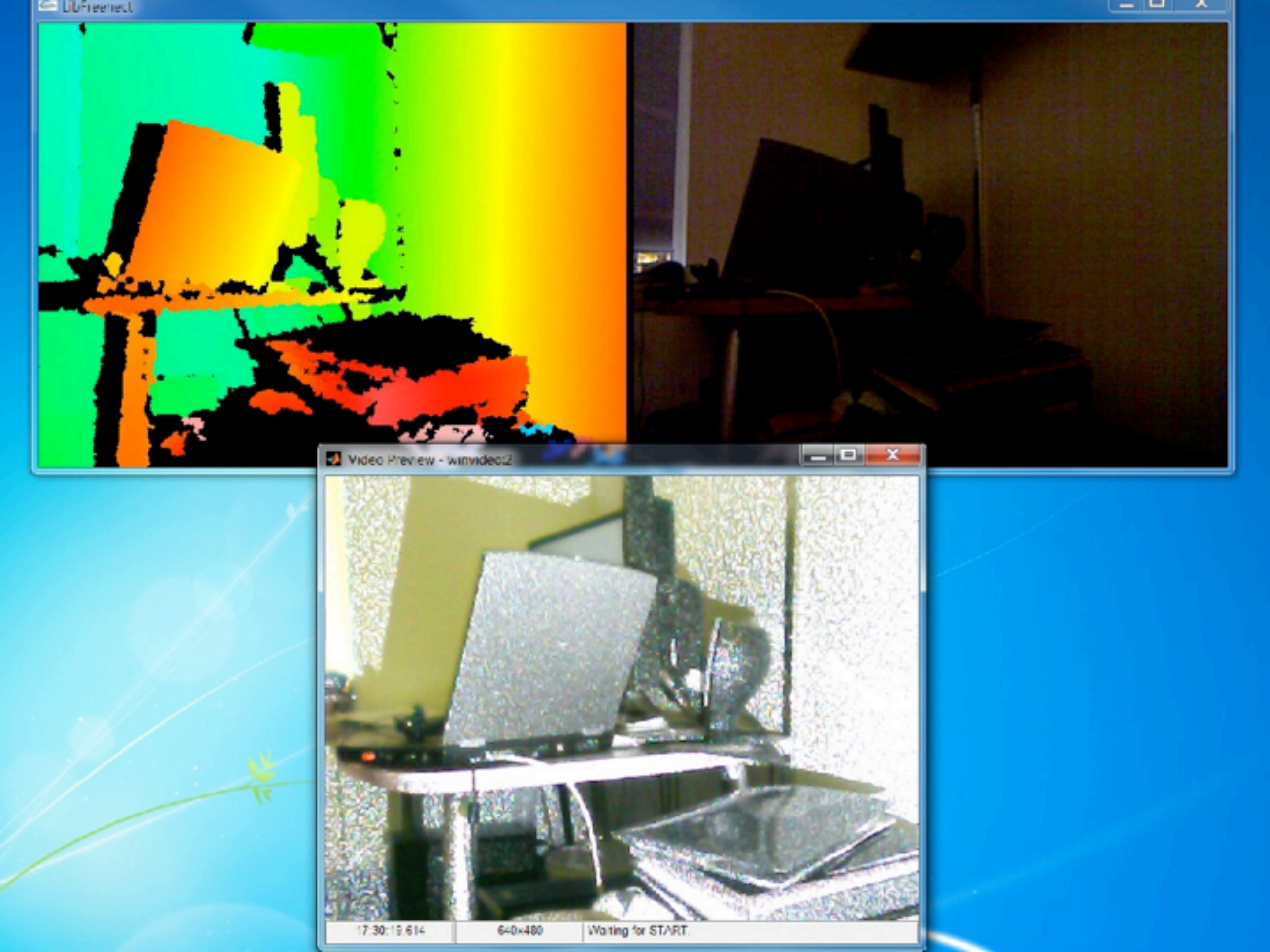

RC1
Q2
R11
C1
5V DM DP GND


Megapixel 10X Digital Zoom f=3.85mm




LibFreenect
Video Preview - winvideo2

LibFreenect
Video Preview - winvideo2
Waiting for START


LibFreenect
Video Preview - winvideo2
17:30:19.614
640x480
Waiting for START.

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