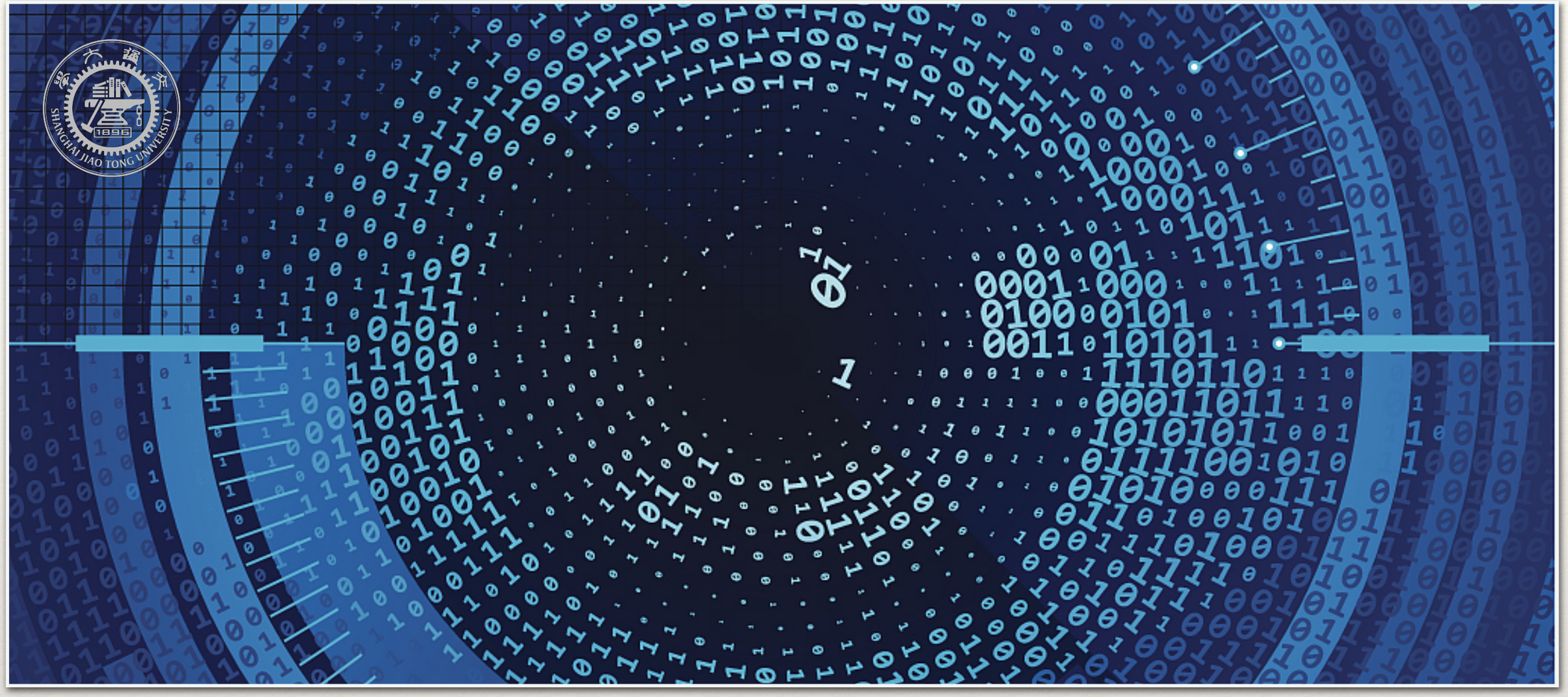


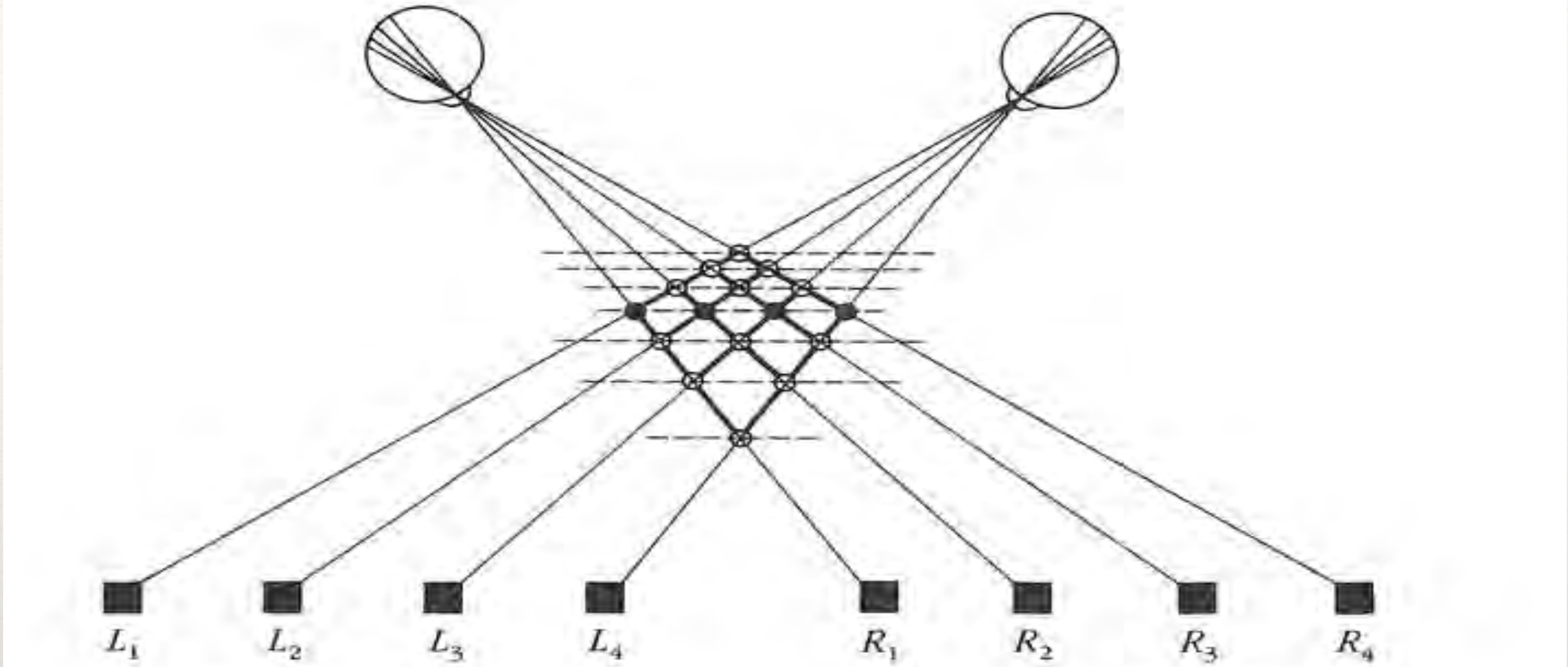
Shanghai Jiao Tong University

Computer Vision

Instructor: Xu Zhao

Class No.: C032703 F032528

Spring 2020



Xu Zhao @ Shanghai Jiao Tong university

Lecture 5-2: Stereopsis

Contents

- Stereo introduction
- Epipolar geometry and the fundamental matrix
- Stereo depth estimation

What visual or physiological cues help us to perceive 3D shape and depth?

- Shape from Shading
  - Photometric stereo: shape from multiple shaded images

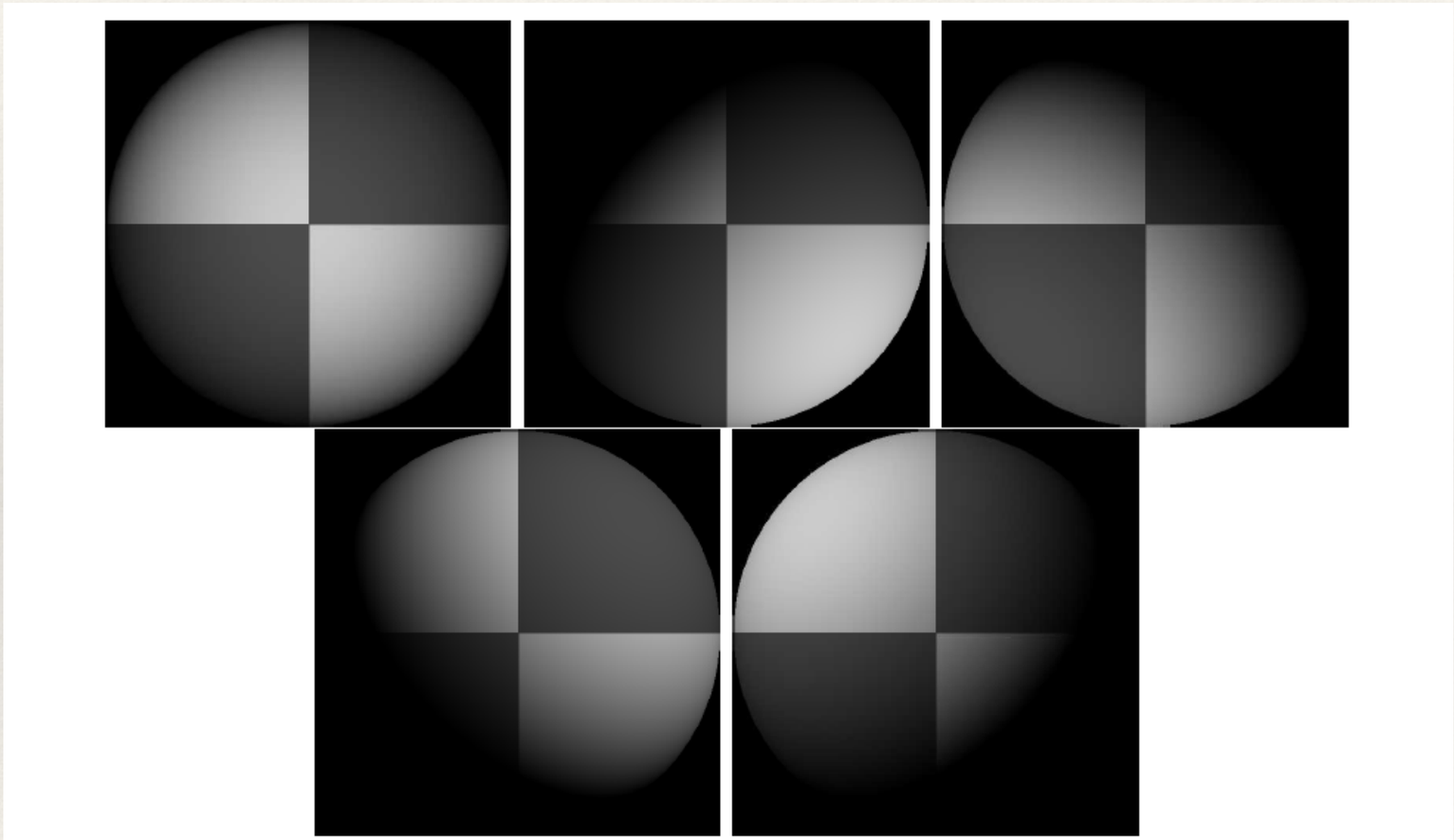
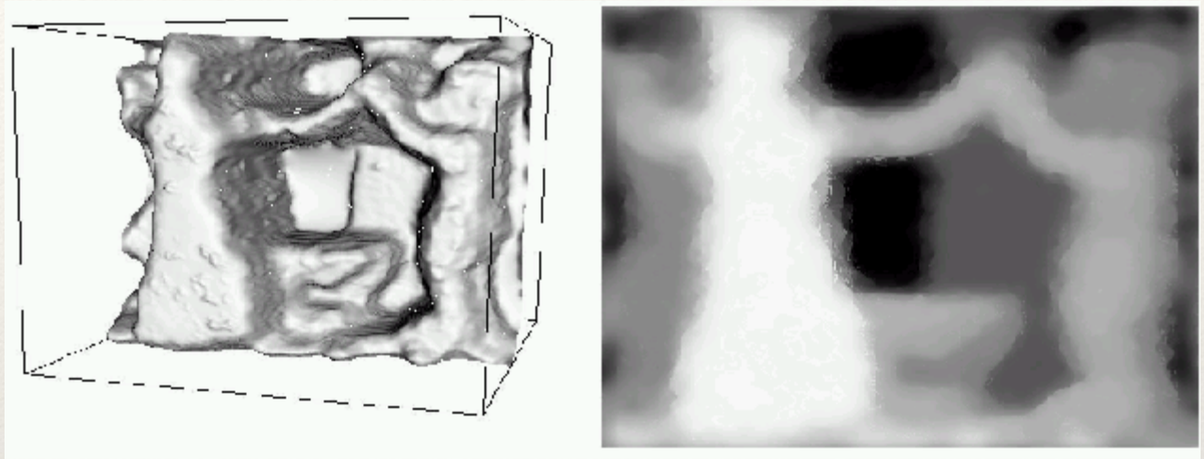
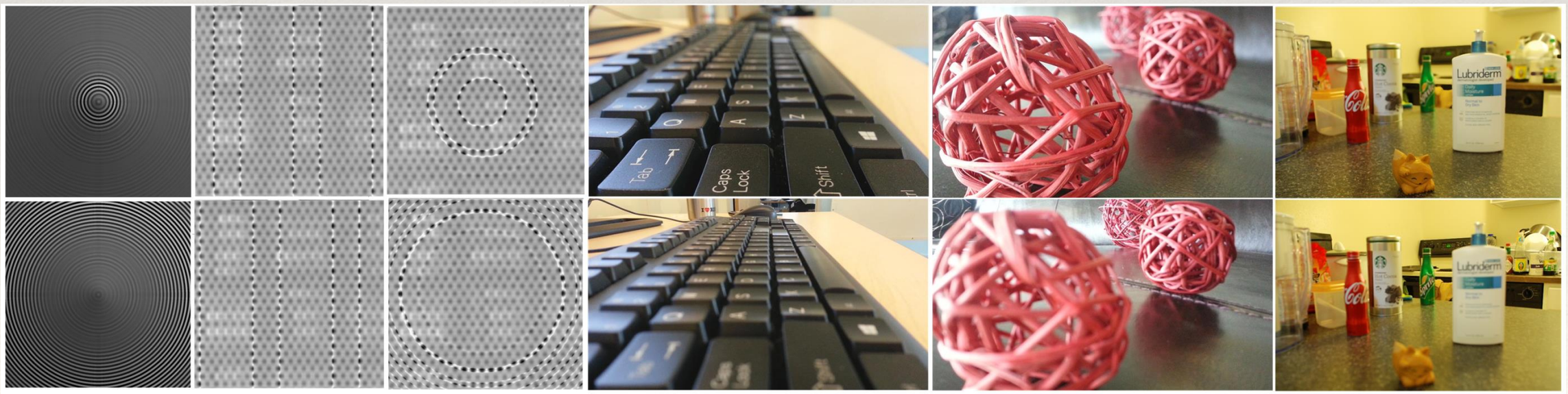


FIGURE 2.11: Five synthetic images of a sphere, all obtained in an orthographic view from the same viewing position. These images are shaded using a local shading model and a distant point source. This is a convex object, so the only view where there is no visible shadow occurs when the source direction is parallel to the viewing direction. The variations in brightness occuring under different sources code the shape of the surface.

What visual or physiological cues help us to perceive 3D shape and depth?

- Shape from focus

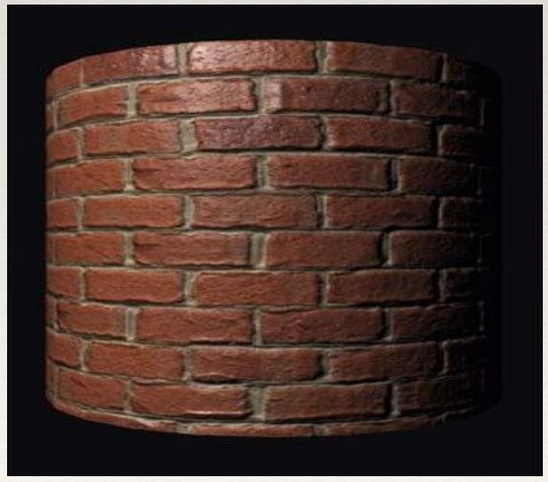


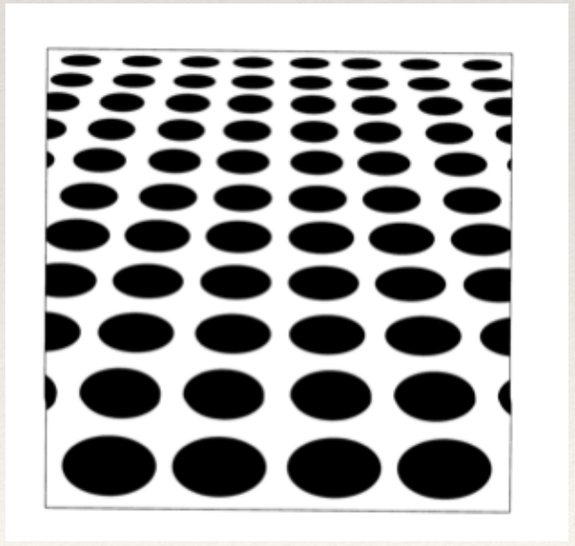


(a) Cone (b) Sinusoidal (c) Cosine (d) Keyboard (e) Balls (f) Kitchen

What visual or physiological cues help us to perceive 3D shape and depth?

- Shape from texture
  - Recover 3D information from 2D image clues.
  - Image clues: variations of size, shape, and density of texture primitives.
  - Yield: surface shape and orientation.





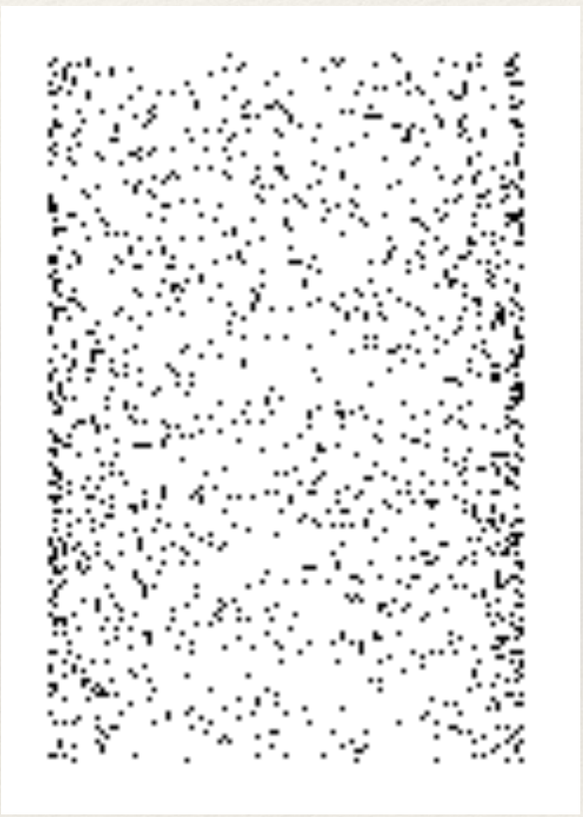
What visual or physiological cues help us to perceive 3D shape and depth?

- Perspective effects

Image credit: S. Seitz

What visual or physiological cues help us to perceive 3D shape and depth?

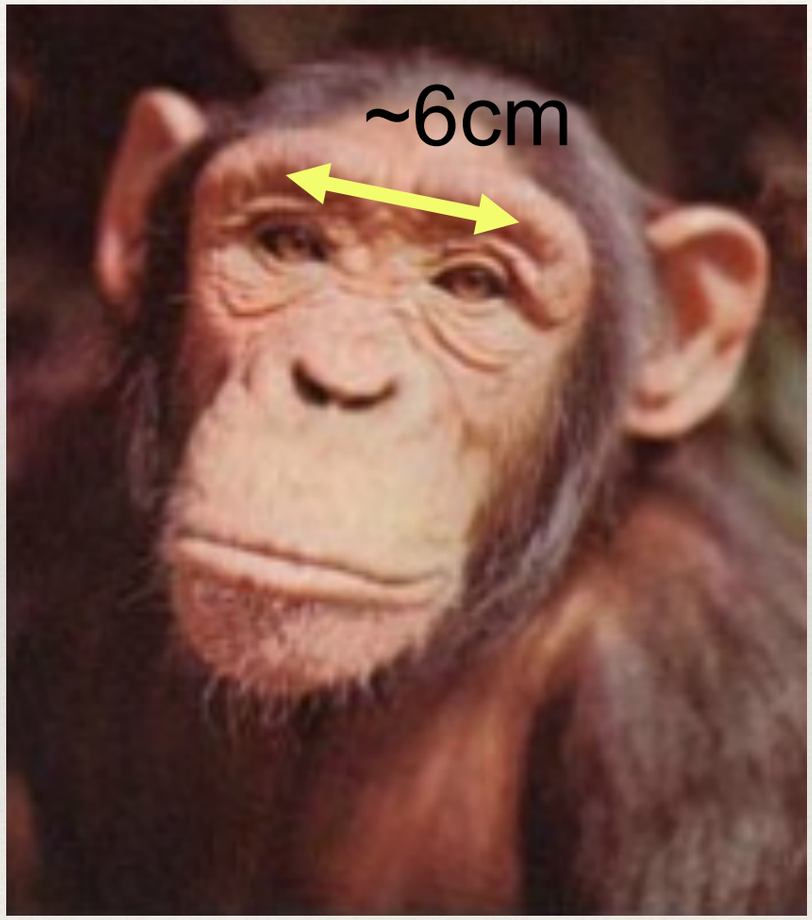
- Motion

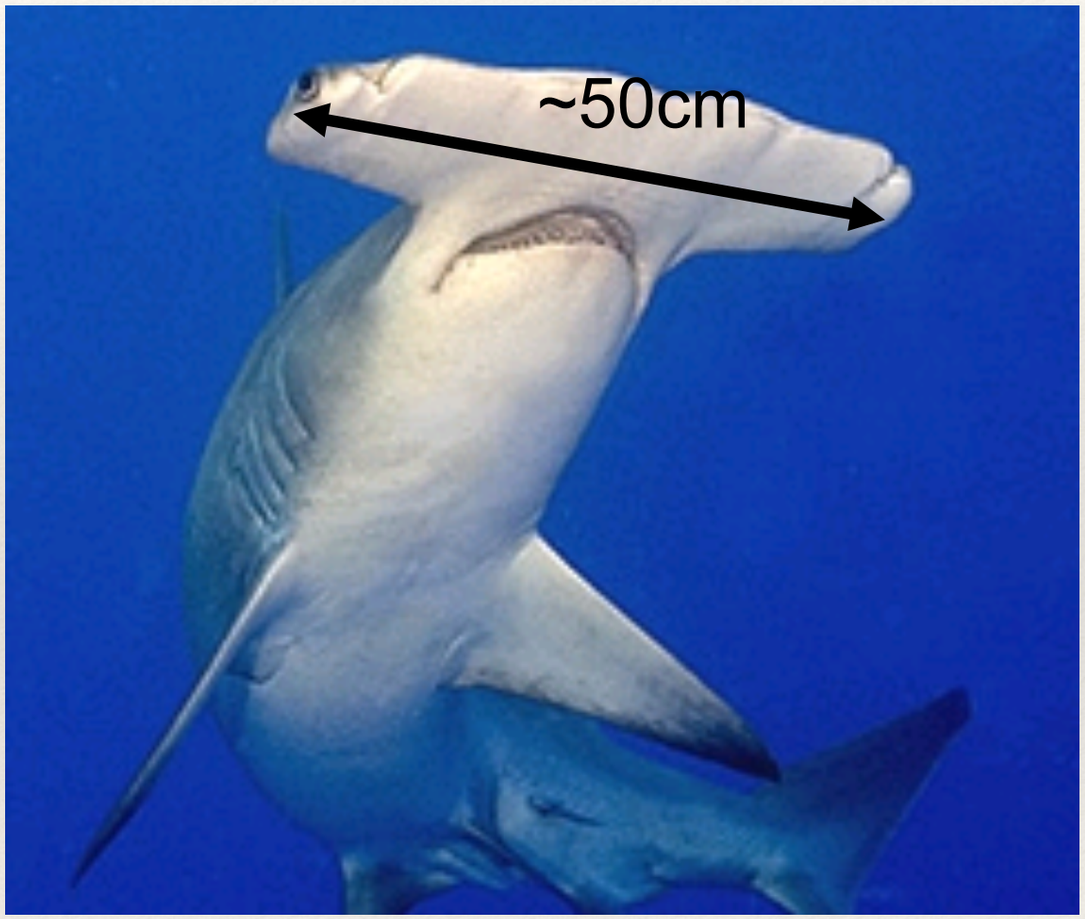


Figures from L. Zhang

http://www.brainconnection.com/teasers/?main=illusion/motion-shape

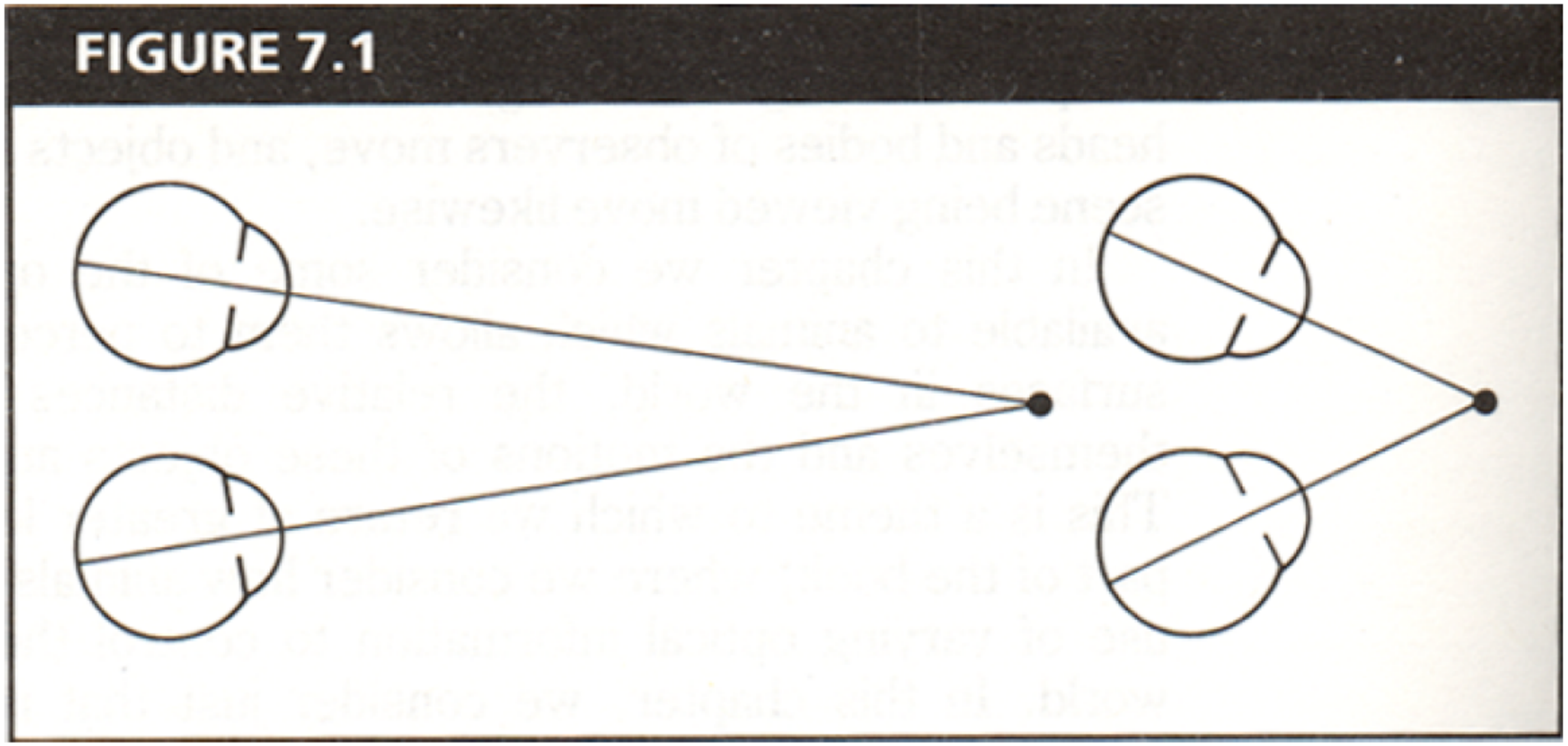
Stereo vision





Human stereopsis

- Human eyes **fixate** on point in space – rotate so that corresponding images form in centers of fovea.



From Bruce and Green, Visual Perception, Physiology, Psychology and Ecology

Human stereopsis

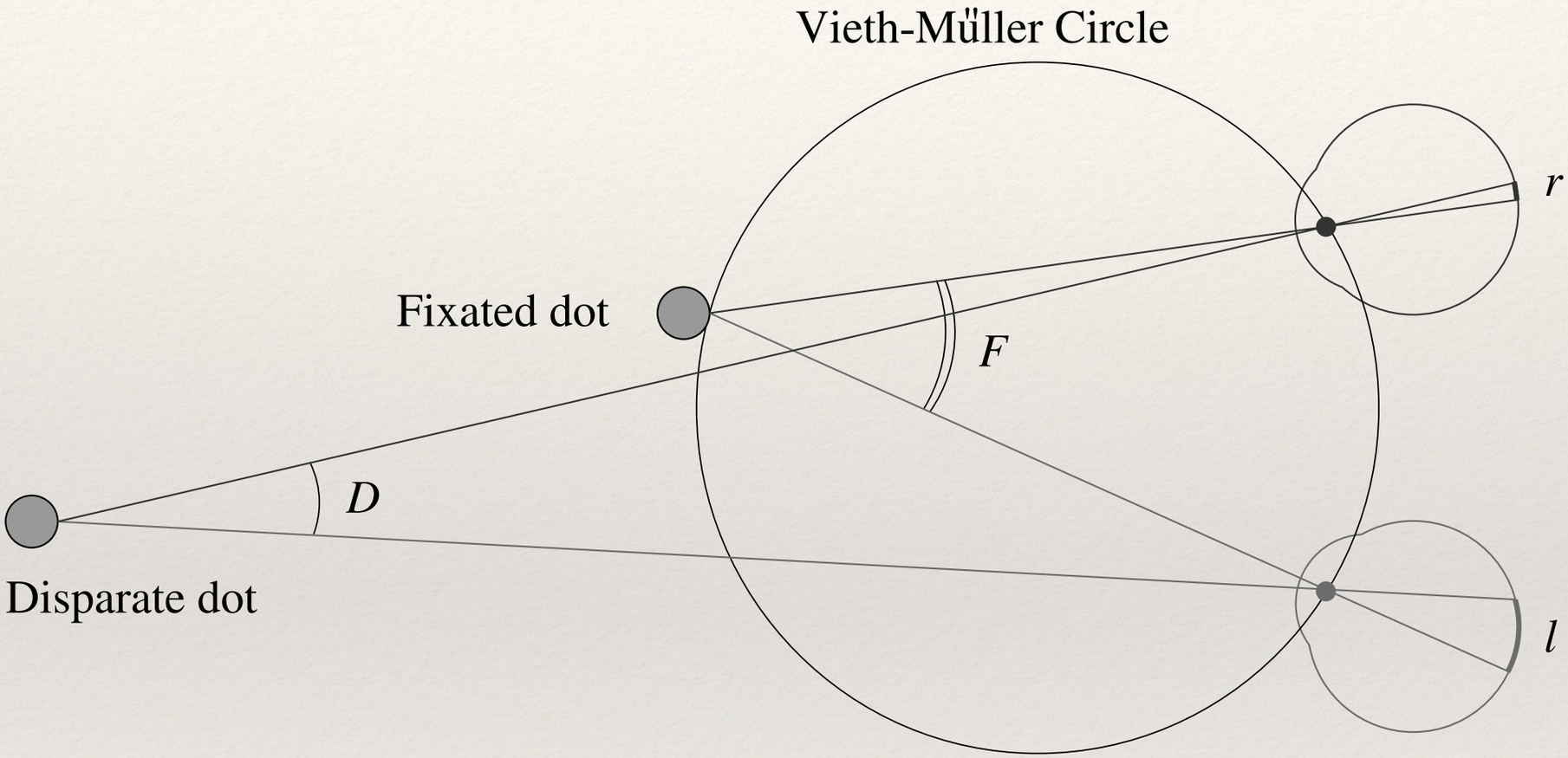
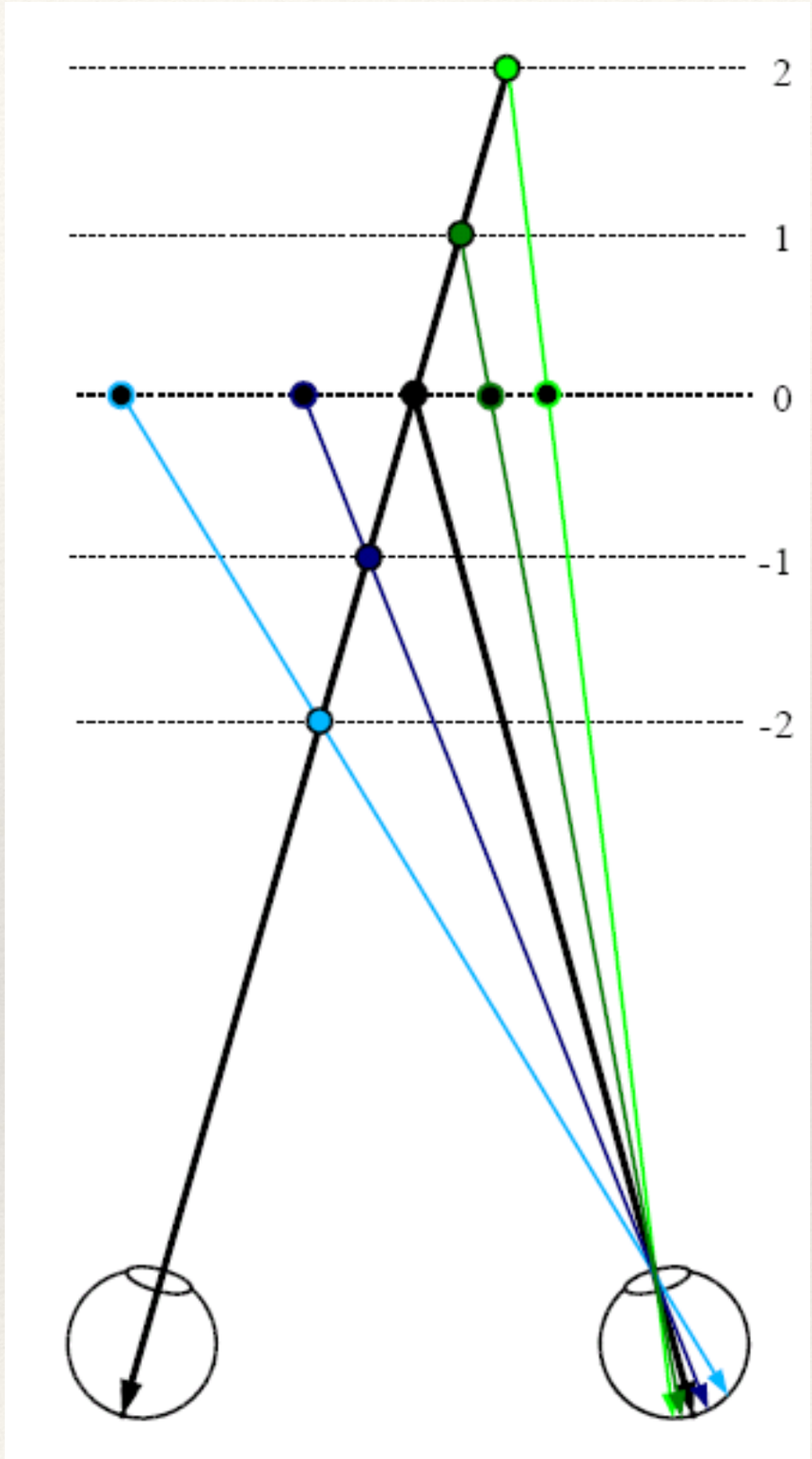


FIGURE 7.7: In this diagram, the close-by dot is fixated by the eyes, and it projects onto the center of their foveas with no disparity. The two images of the far dot deviate from this central position by different amounts, indicating a different depth.


2
1
0
-1
-2

Random dot stereograms (Bela Julesz)

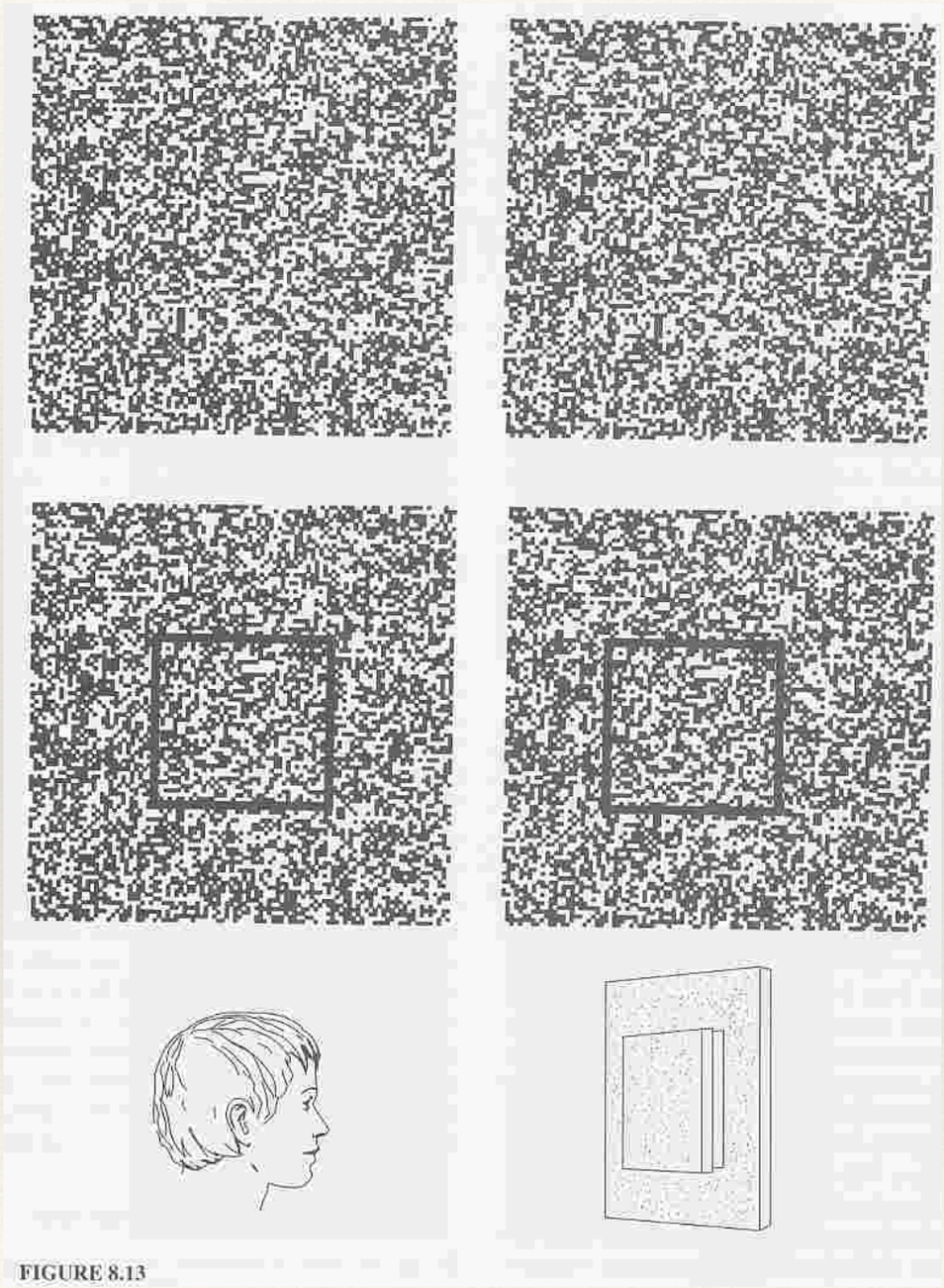
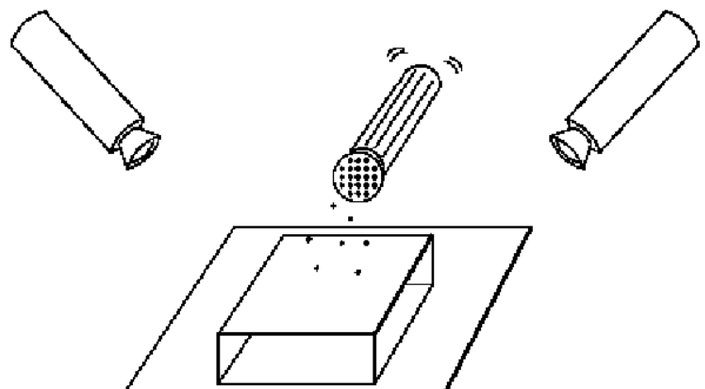
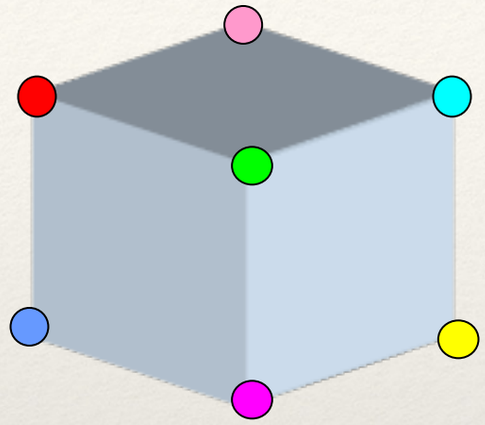


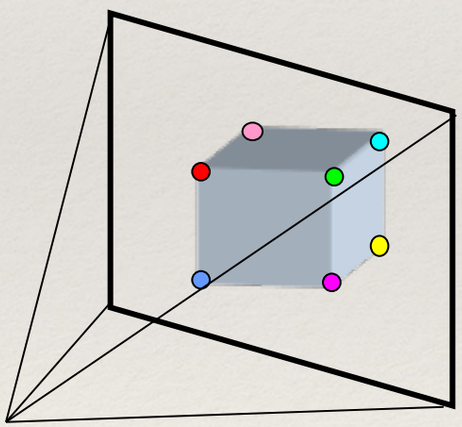
FIGURE 8.13

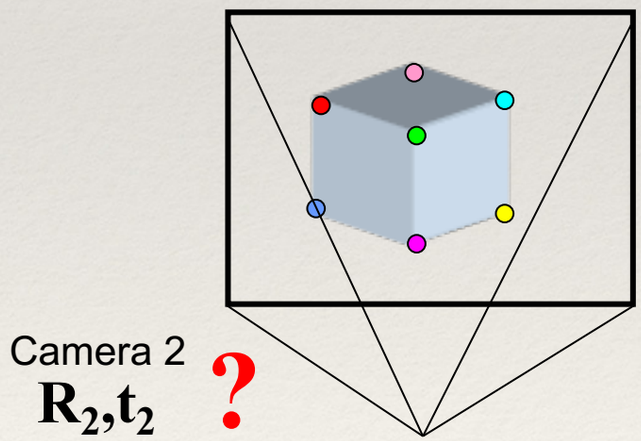


"When viewed monocularly, the images appear completely random. But when viewed stereoscopically, the image pair gives the impression of a square markedly in front of (or behind) the surround."

Two-view geometry problems



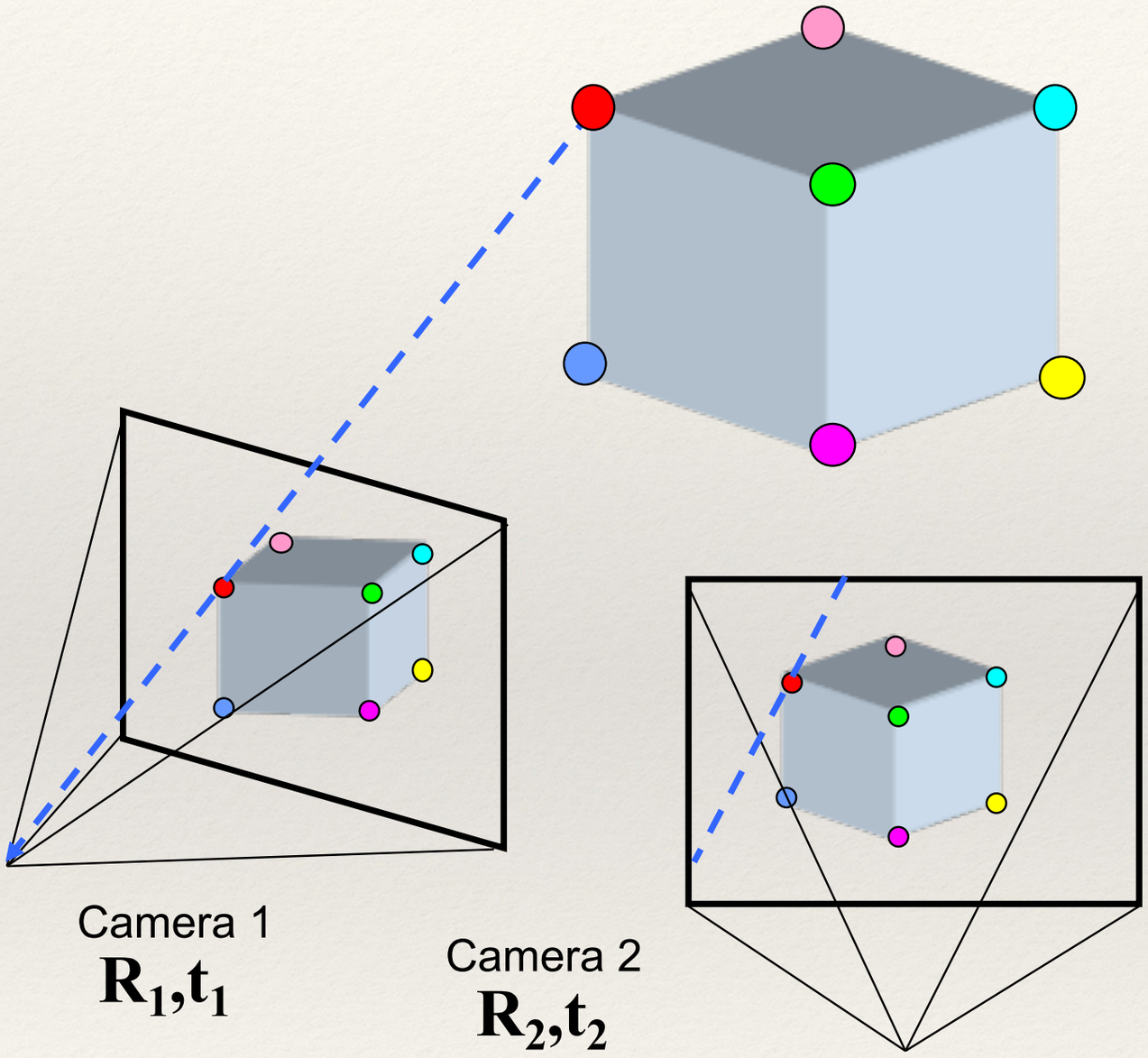




- **Camera 'Motion':** Given a set of corresponding 2D/3D points in two images, compute the camera parameters.

Slide credit: Noah Snavely

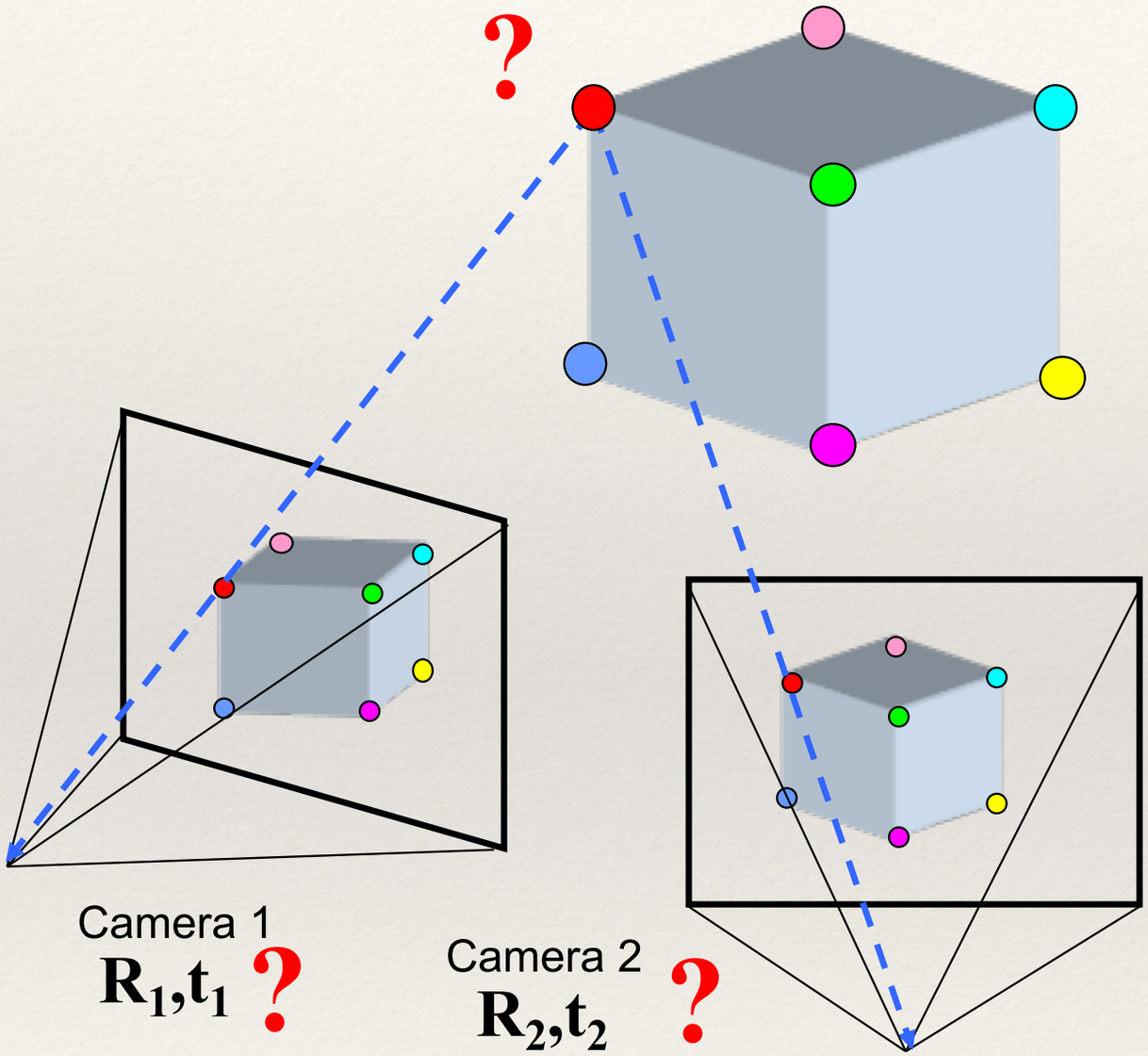
Two-view geometry problems



- **Stereo correspondence:** Given known camera parameters and a point in one of the images, where could its corresponding points be in the other images?

Slide credit: Noah Snavely

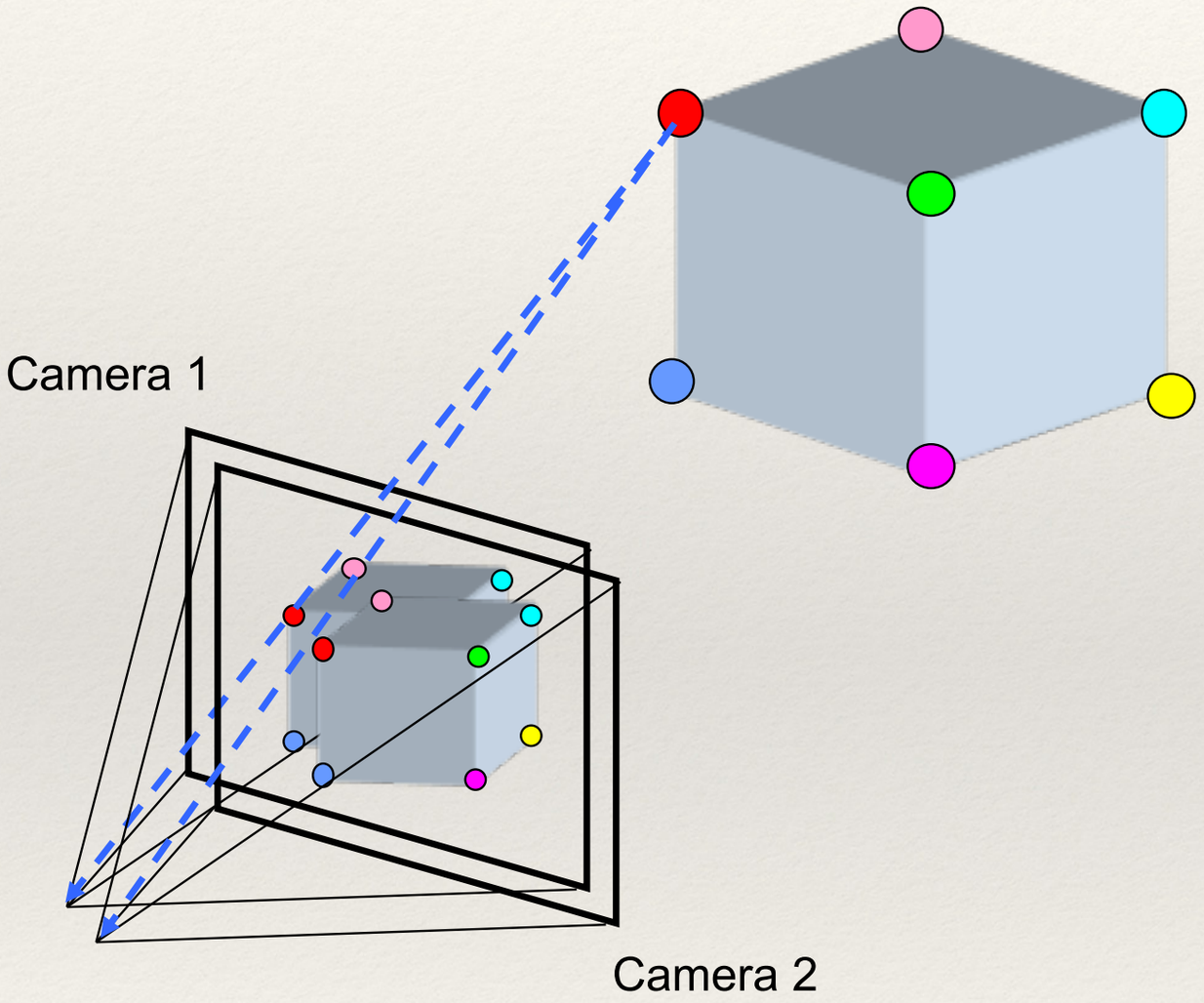
Two-view geometry problems



- **Structure from Motion:** Given projections of the same 3D point in two or more images, compute the 3D coordinates of that point

Slide credit: Noah Snavely

Two-view geometry problems



- **Optical flow:** Given two images, find the location of a world point in a second close-by image with no camera info.

Slide credit: Noah Snavely

Estimating depth with stereo

- **Depth**: The distance from camera center to a scene point, or the z-coordinate of scene point, is an important information to understand the 3-dimentional scene from its images.
- **Stereo**: A vision technique to compute the depth of scene by the position difference of a scene point in the two camera's image planes. ***shape from disparities between two views.***
- We'll need to consider:
  - Info on camera pose ("calibration")
  - Image point correspondences

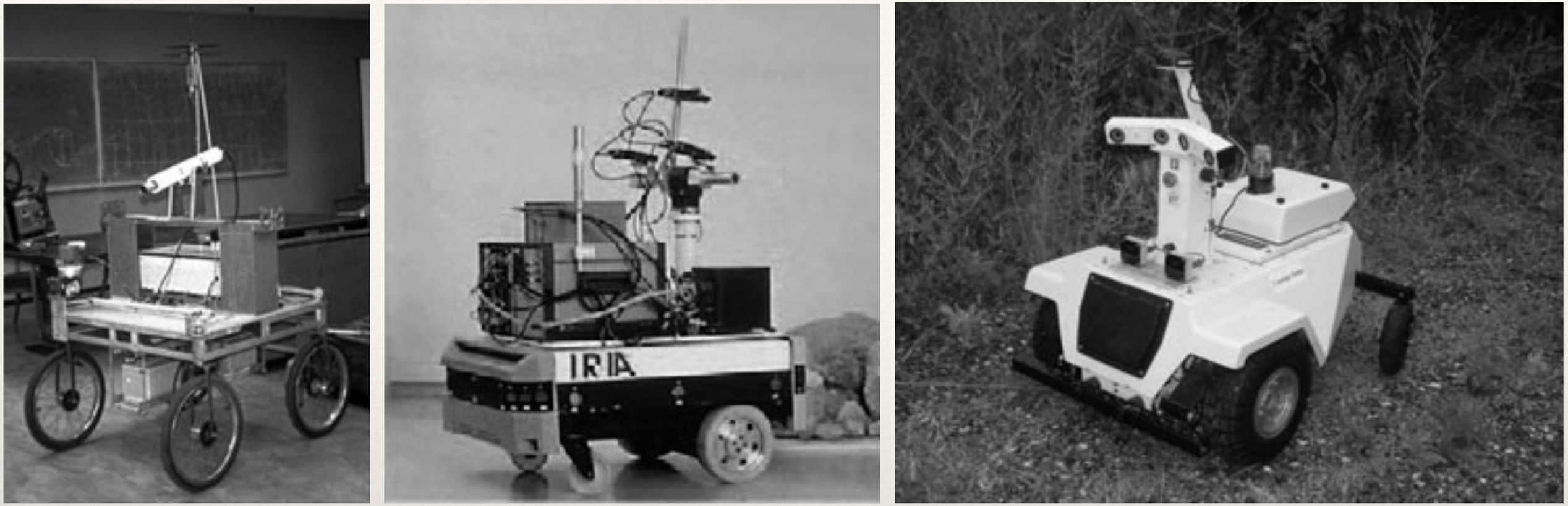
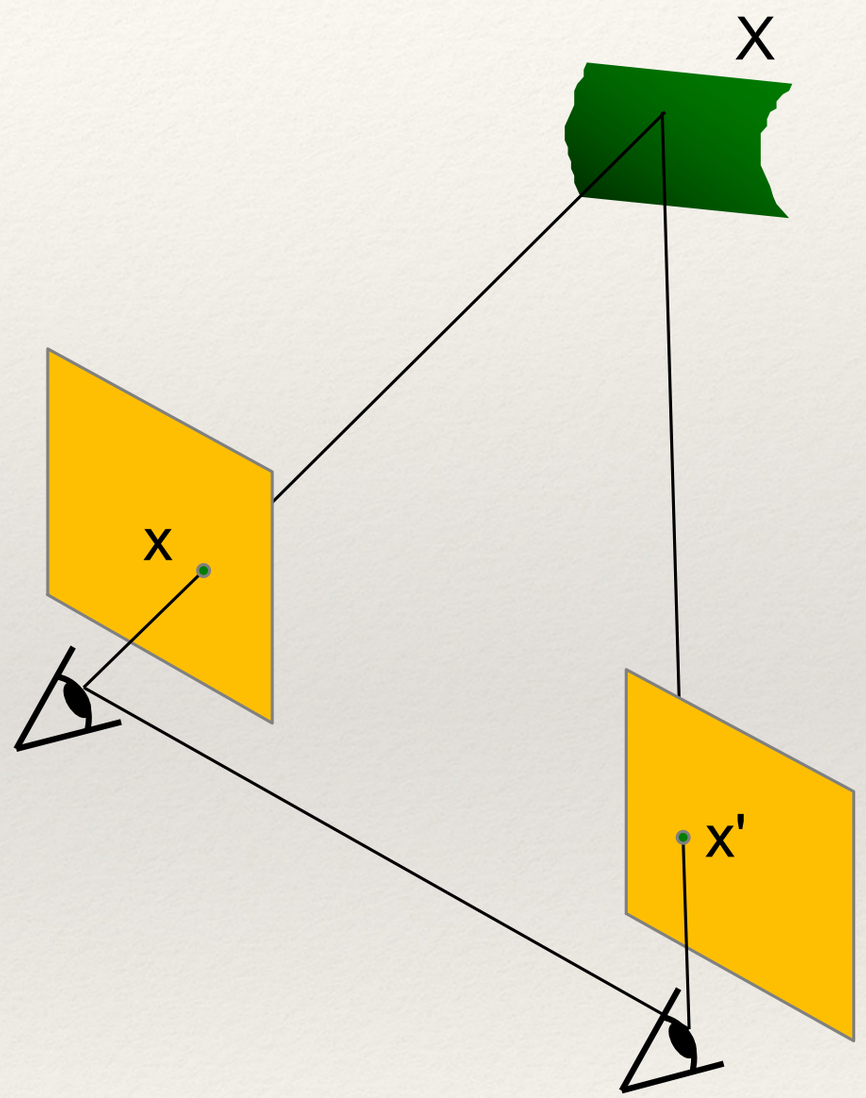
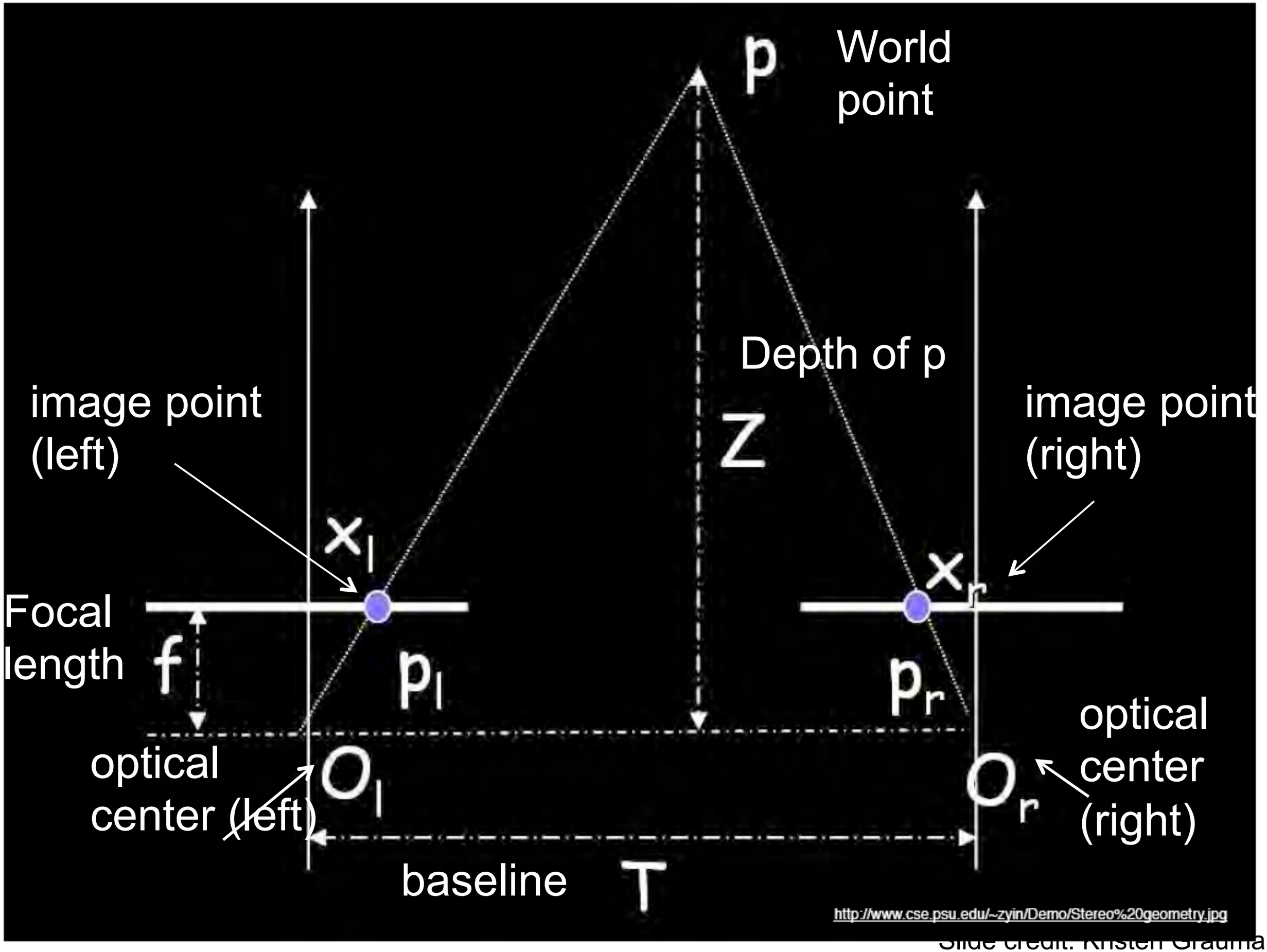


FIGURE 7.1: **Left:** The Stanford cart sports a single camera moving in discrete increments along a straight line and providing multiple snapshots of outdoor scenes. **Center:** The INRIA mobile robot uses three cameras to map its environment. **Right:** The NYU mobile robot uses two stereo cameras, each capable of delivering an image pair. As shown by these examples, although two eyes are sufficient for stereo fusion, mobile robots are sometimes equipped with three (or more) cameras. The bulk of this chapter is concerned with binocular perception but stereo algorithms using multiple cameras are discussed in Section 7.6. *Photos courtesy of Hans Moravec, Olivier Faugeras, and Yann LeCun.*

Geometry for a simple stereo system

- Assume:
  - parallel optical axes
  - known camera parameters (i.e., calibrated cameras):
- Goal: recover depth of X by finding image coordinate x’ that corresponds to x

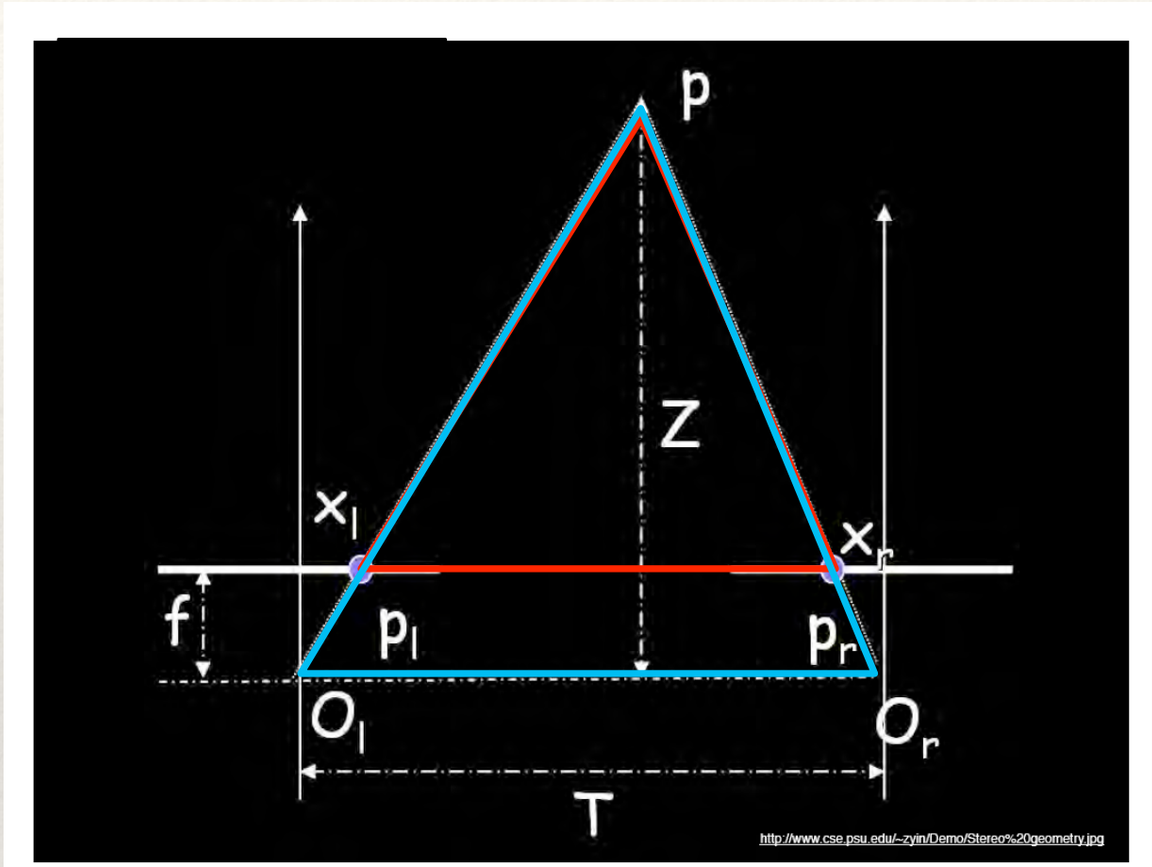




Slide credit: Kristen Grauman

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):



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

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

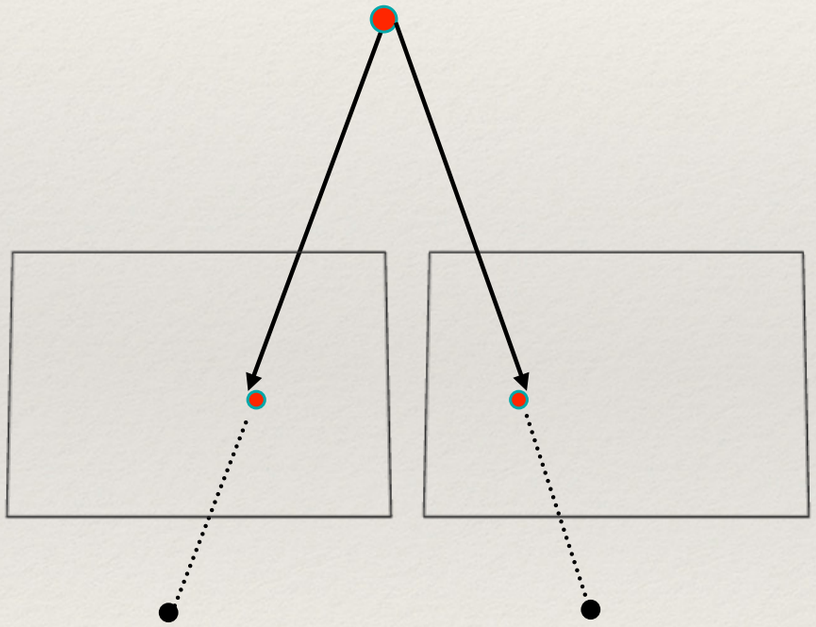
$$Z = f\frac{T}{x_r - x_l}$$

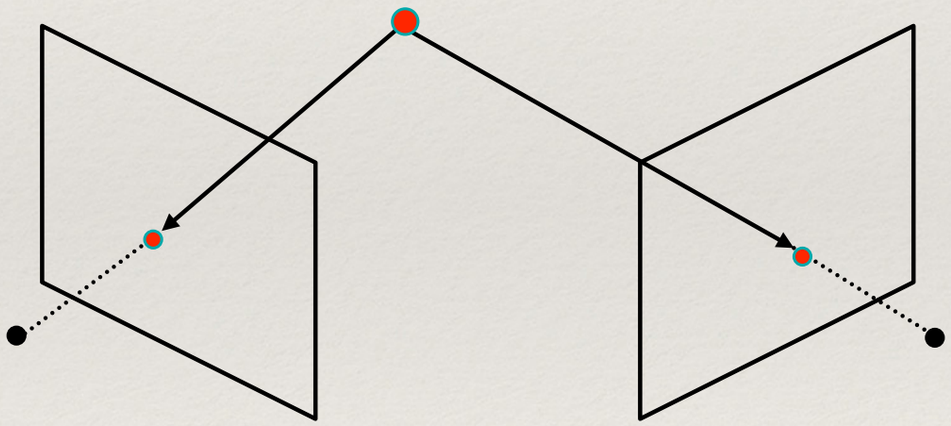
disparity

Slide credit: Kristen Grauman

Stereo of arbitrary camera arrangement

- The two cameras need not have parallel optical axes.

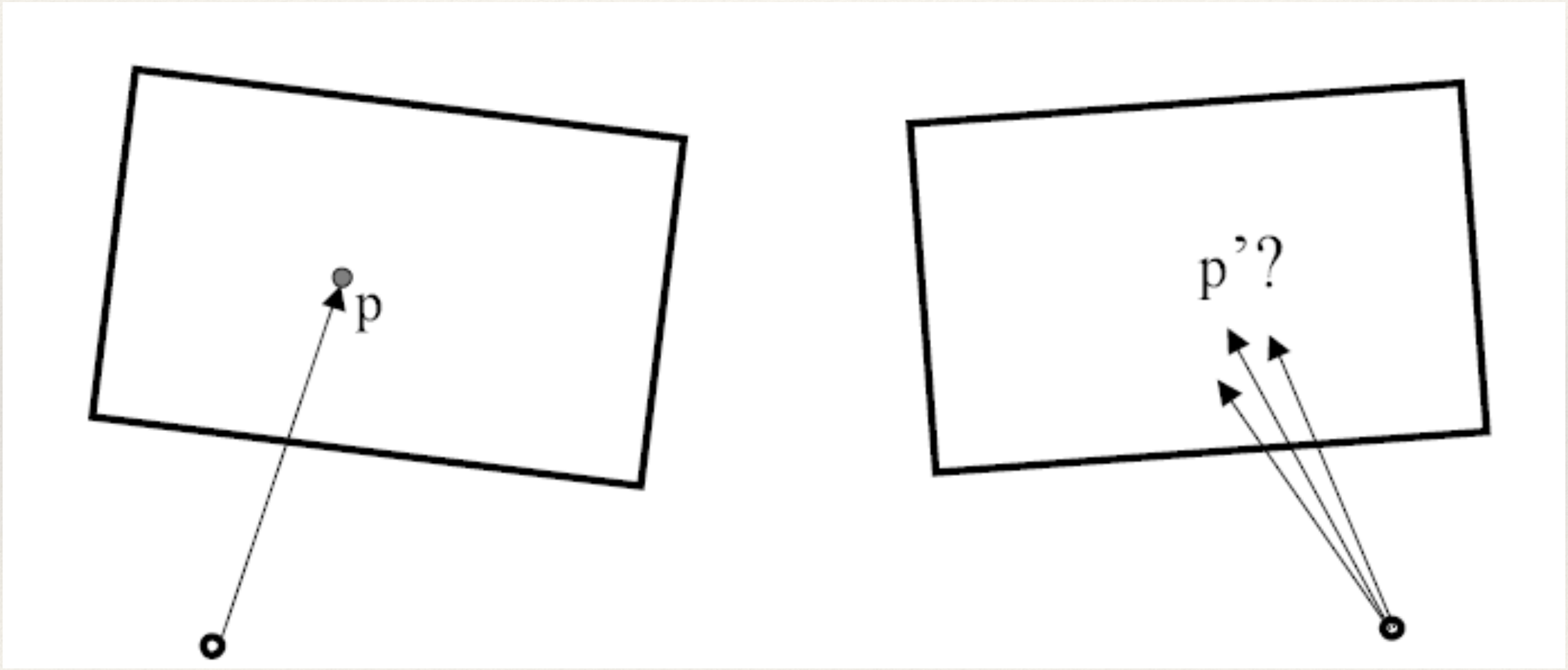


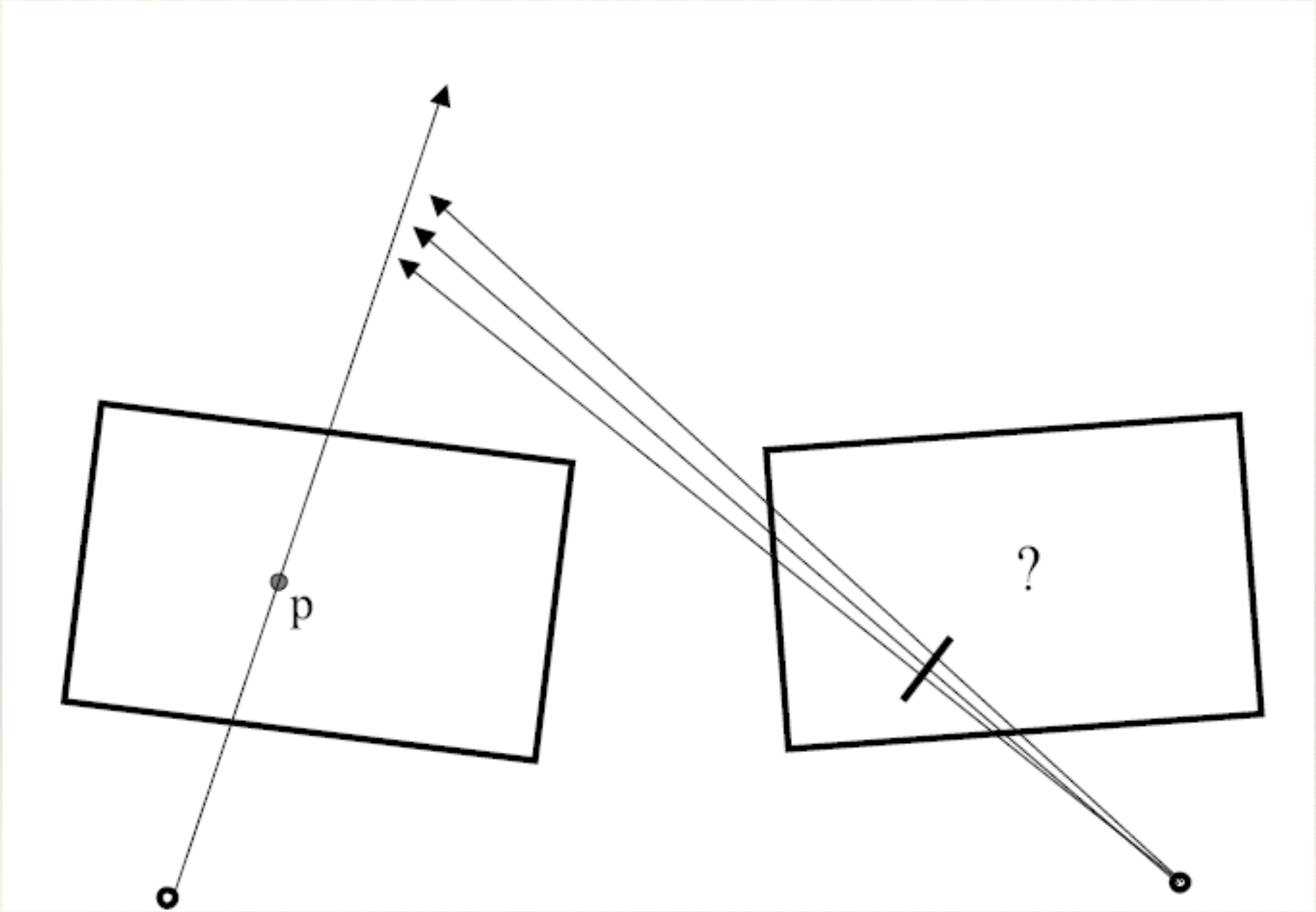


Slide credit: Kristen Grauman

Stereo correspondence constraints

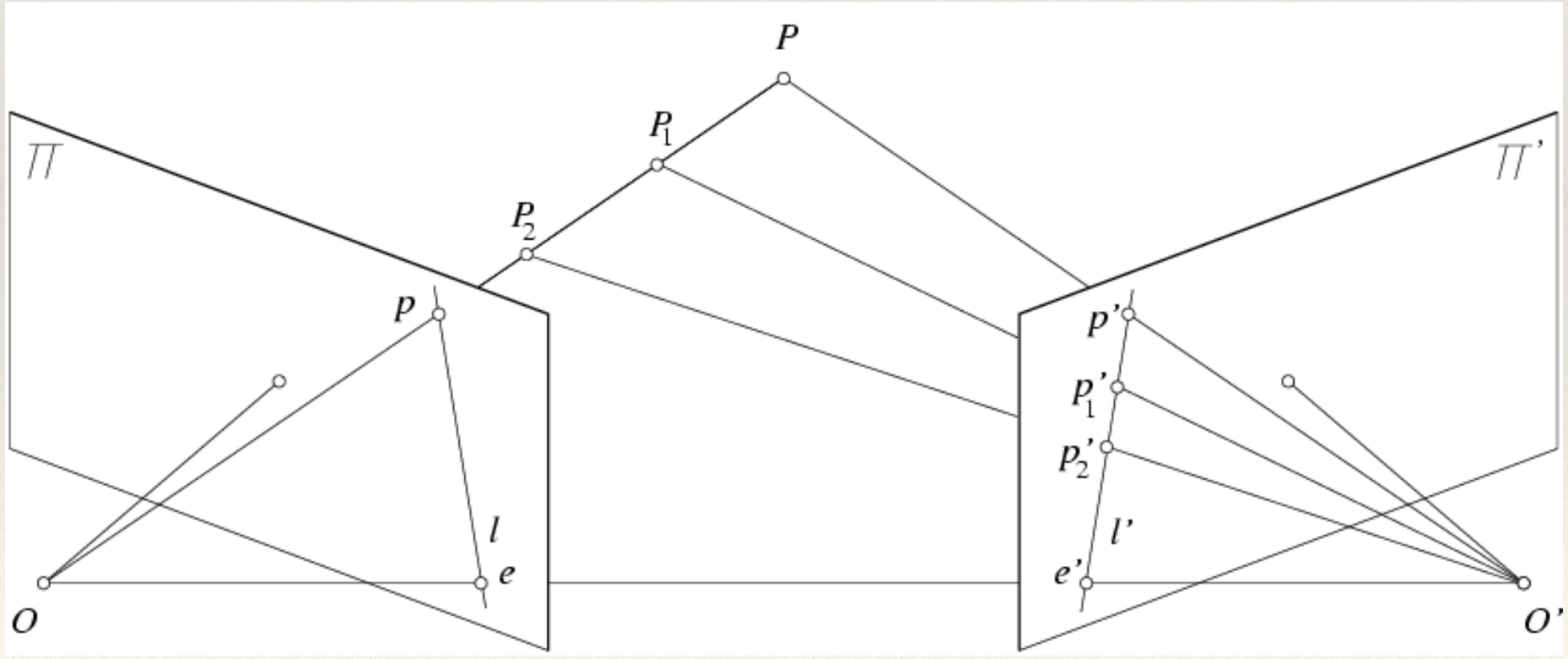
- Given p in left image, where can corresponding point p’ be?




p
?

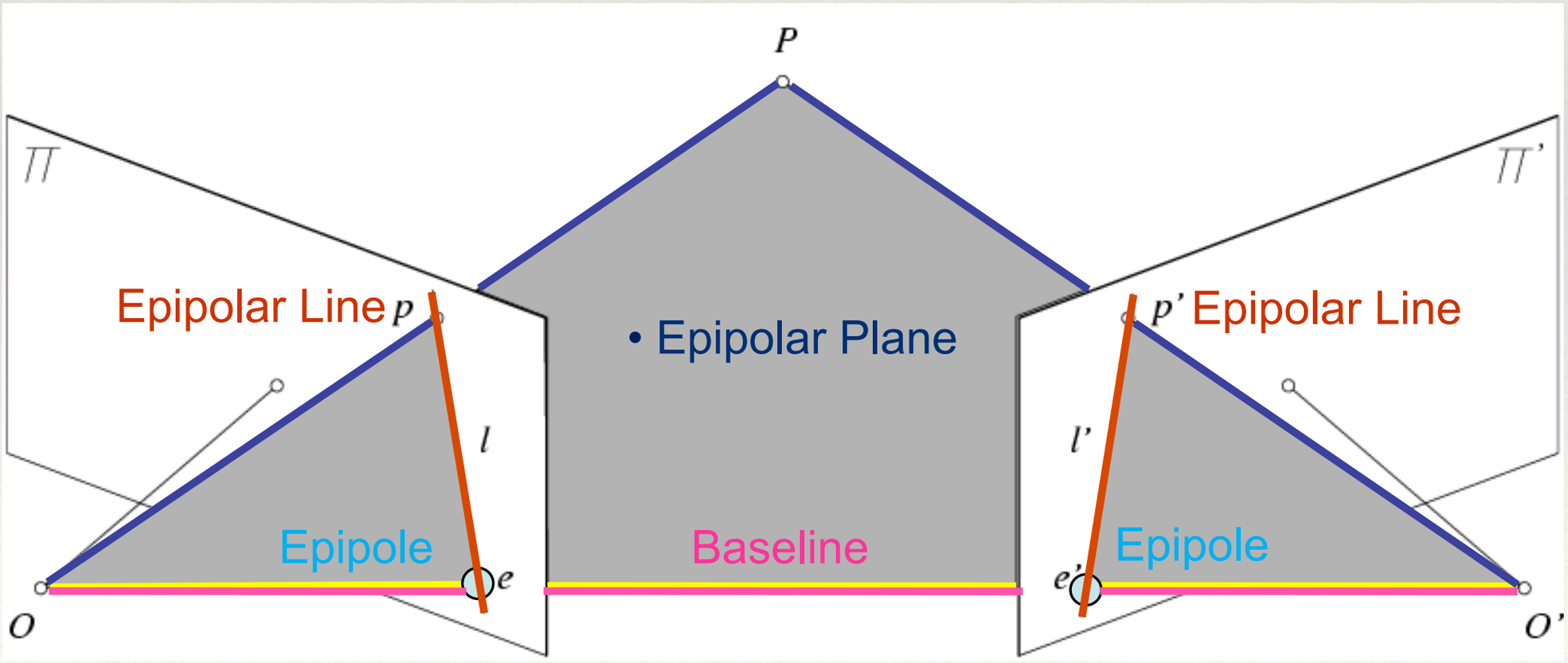
Epipolar constraint

- Geometry of two views constrains where the corresponding pixel for some image point in the first view must occur in the second view: It must be on the line carved out by a plane connecting the world point and optical centers.



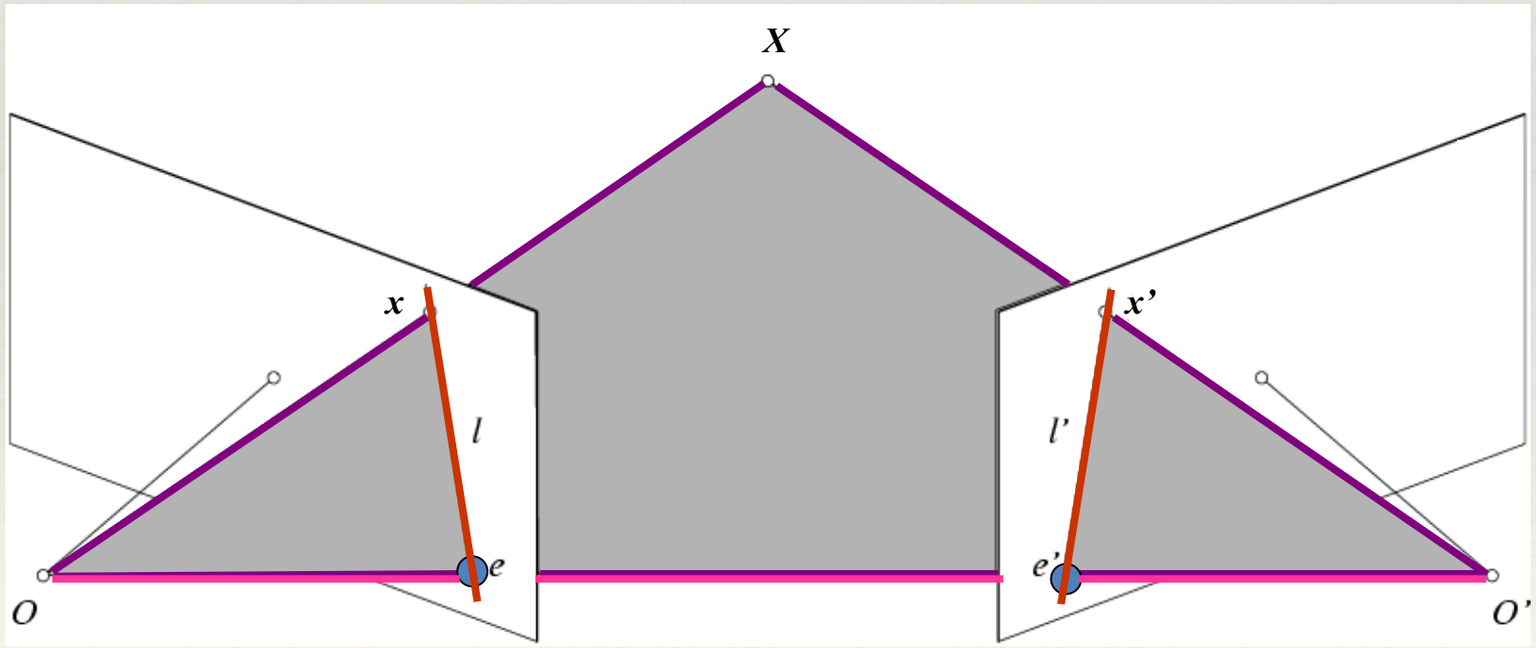
Epipolar geometry

- **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

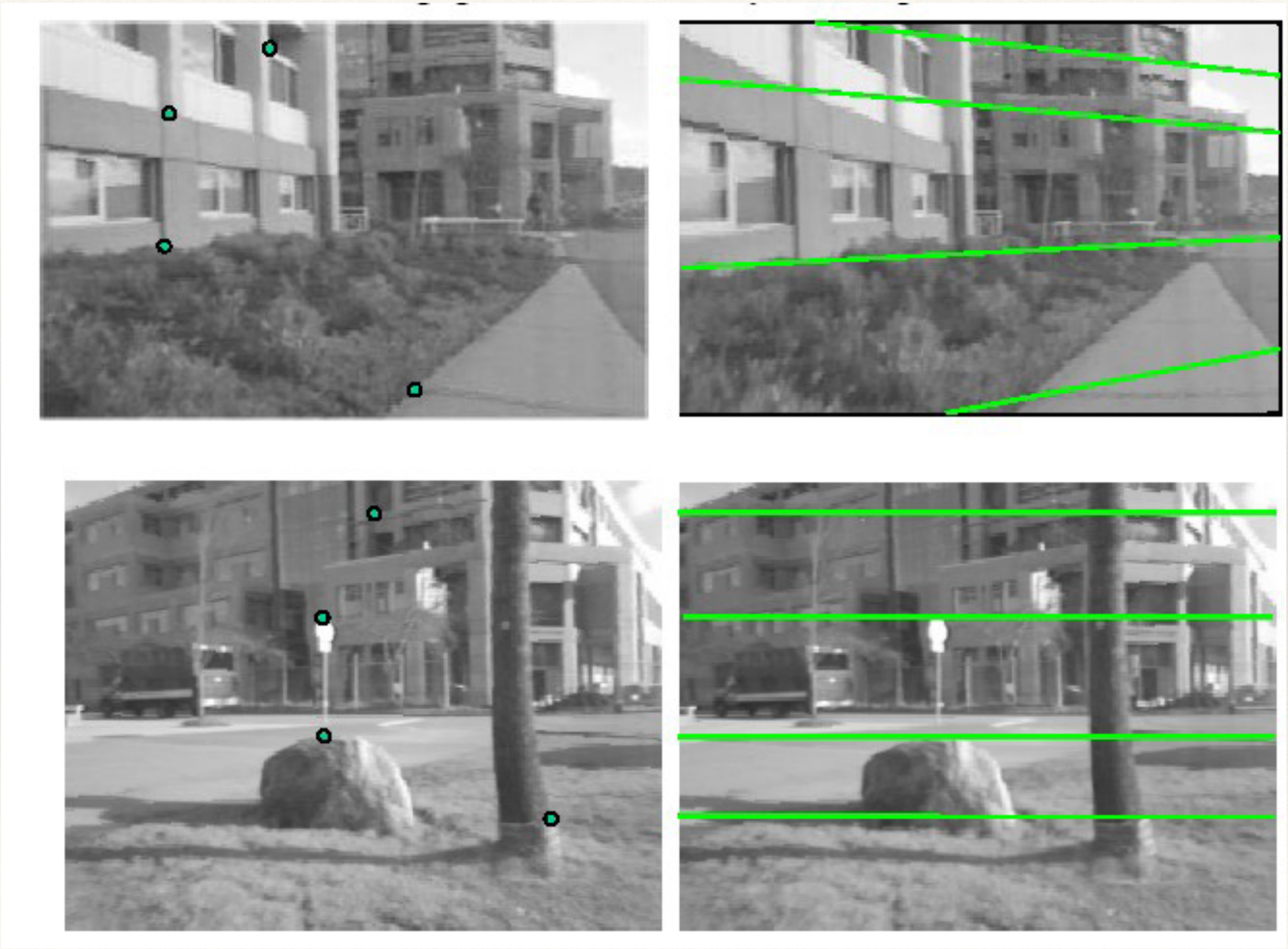


Why epipolar constraint is useful?

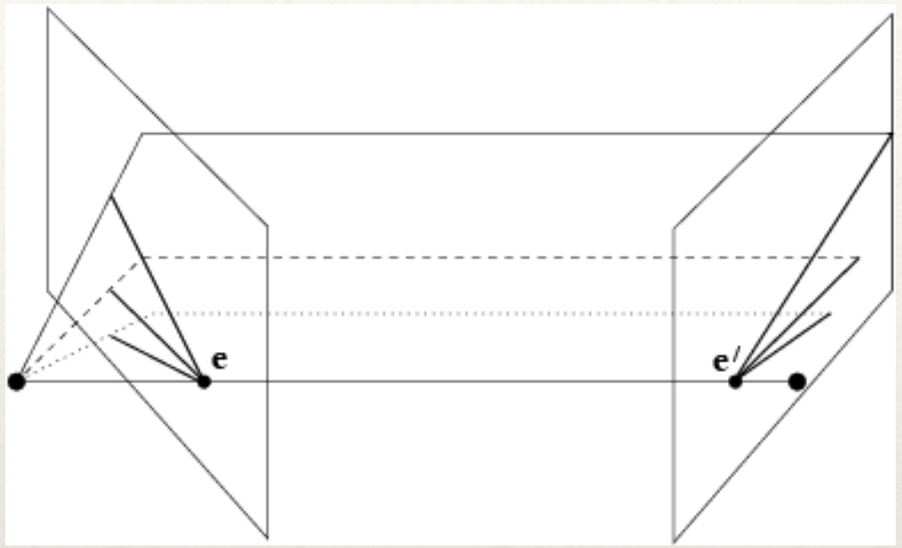
- Reduce search space for stereo disparity estimation.
- Help find *x'*: If I know *x*, and have calibrated cameras (known intrinsics K,K' and extrinsic relationship), I can restrict *x'* to be along *l'*.

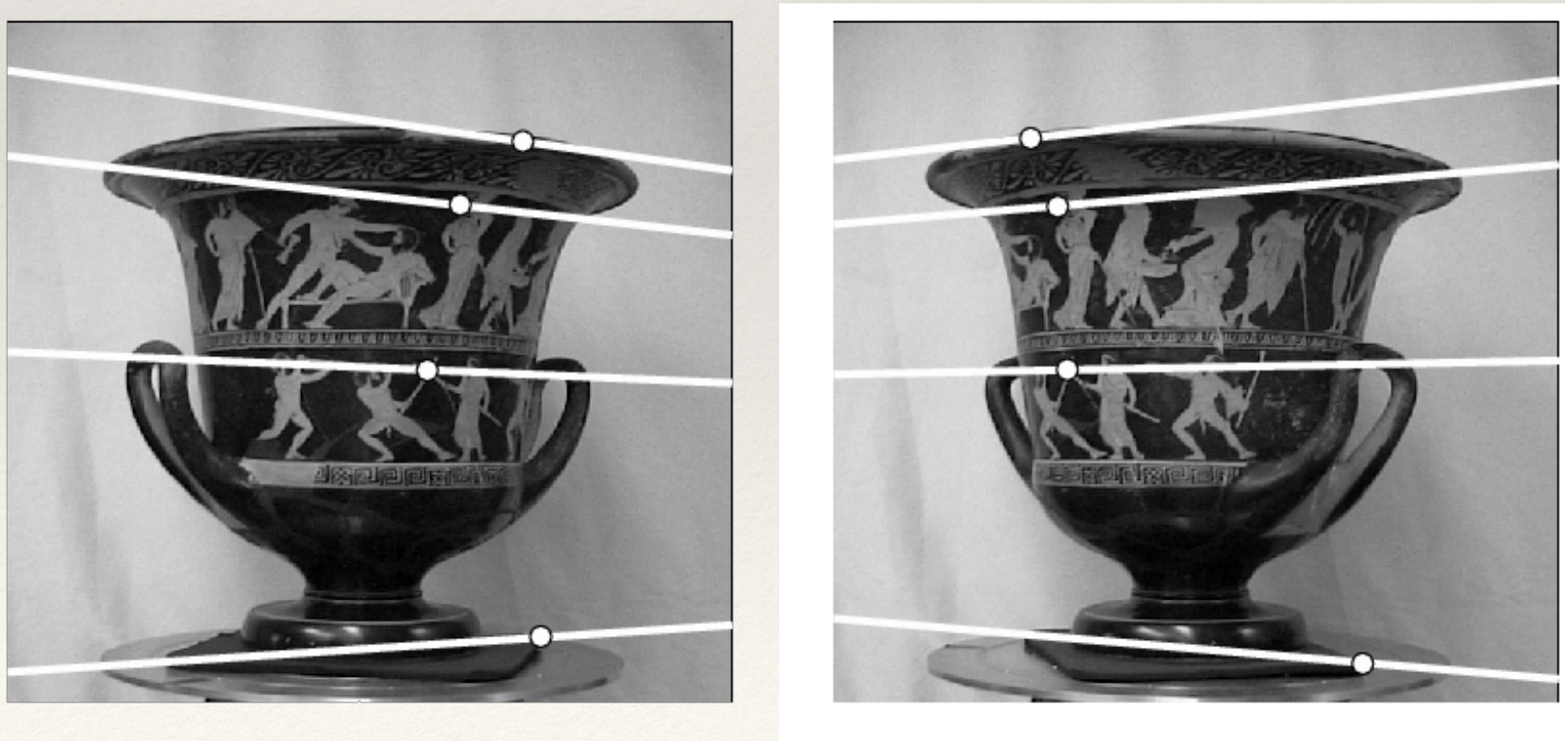


Example



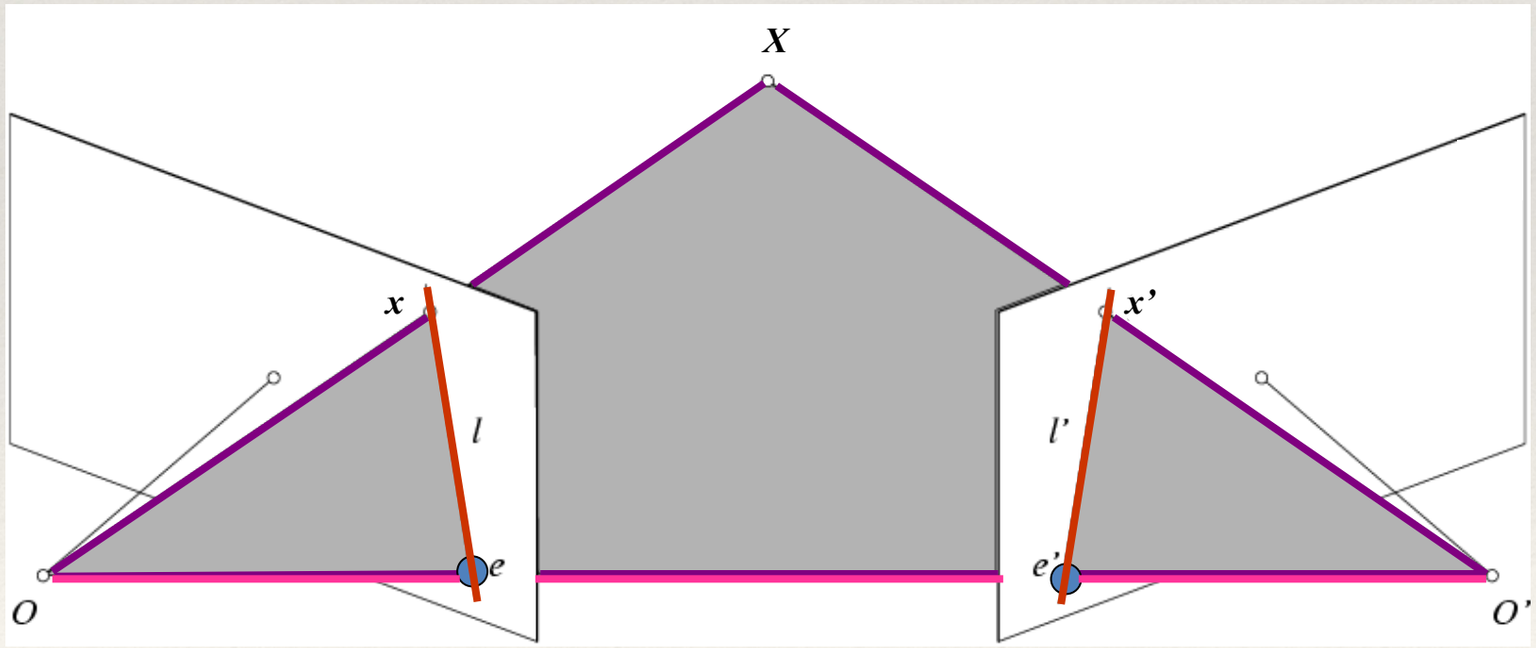
Example: converging cameras



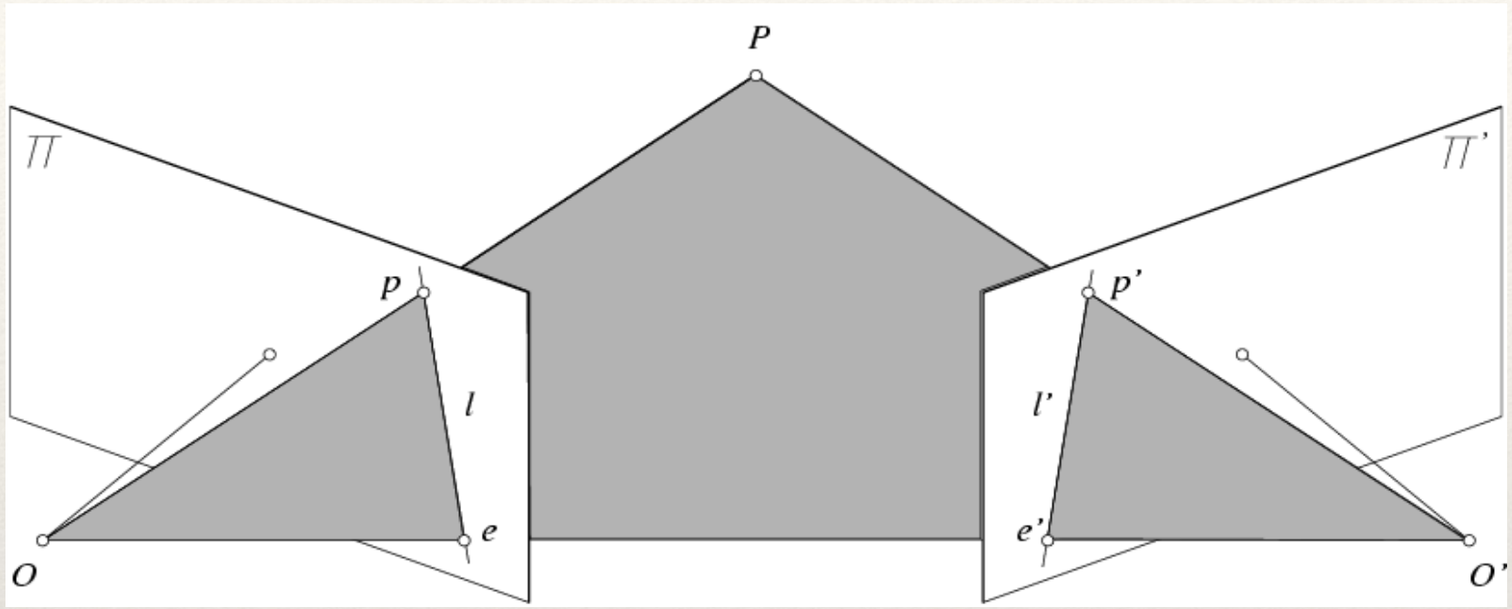


Why epipolar constraint is useful?

- If we have enough *x*, *x'* correspondences, we can estimate relative position and orientation between the cameras and the 3D position of corresponding image points.



Epipolar constraint: calibrated case



- Given the intrinsic parameters of the cameras:

1. Convert to **normalized coordinates** by pre-multiplying all points with the inverse of the calibration matrix; set first camera's coordinate system to world coordinates

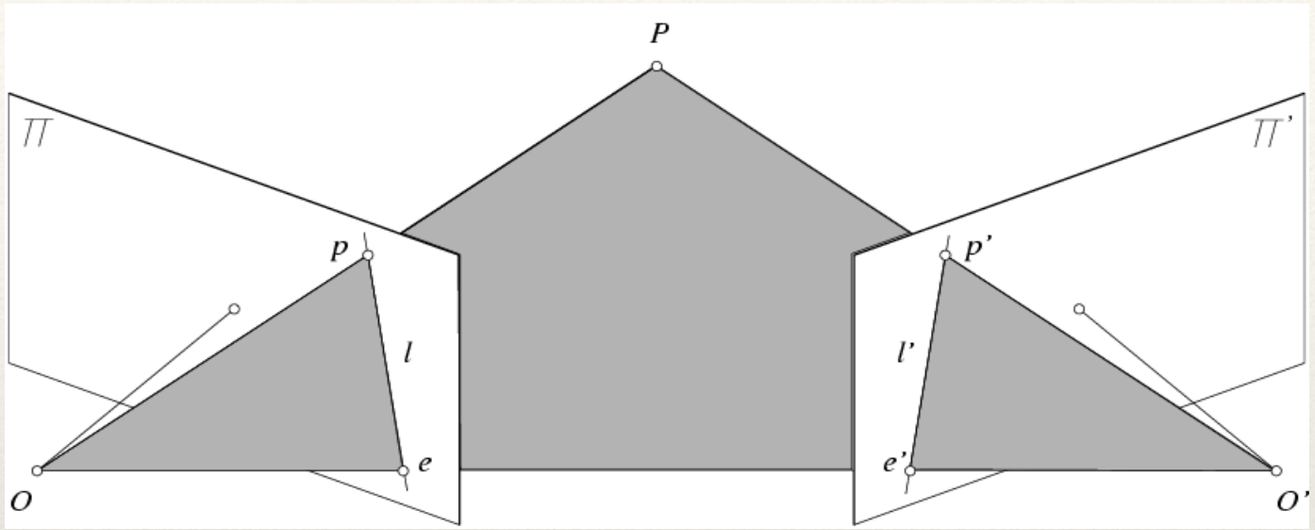
$$\hat{x} = K^{-1}x = X$$

- $\hat{x}$: Homogeneous 2d point (3D ray towards X)
- x: 2D pixel coordinate (homogeneous)
- X: 3D scene point

$$\hat{x}' = K'^{-1}x' = X'$$

- X': 3D scene point in 2nd camera's 3D coordinates

Epipolar constraint: calibrated case



- Given the intrinsic parameters of the cameras:
  1. Convert to **normalized coordinates** by pre-multiplying all points with the inverse of the calibration matrix; set first camera's coordinate system to world coordinates1.
  2. Define some *R* and *t* that relate X to X′ as below

$$\hat{x} = R\hat{x}' + t$$

$$\hat{x} = K^{-1}x = X$$

$\hat{x}$: Homogeneous 2d point (3D ray towards X)

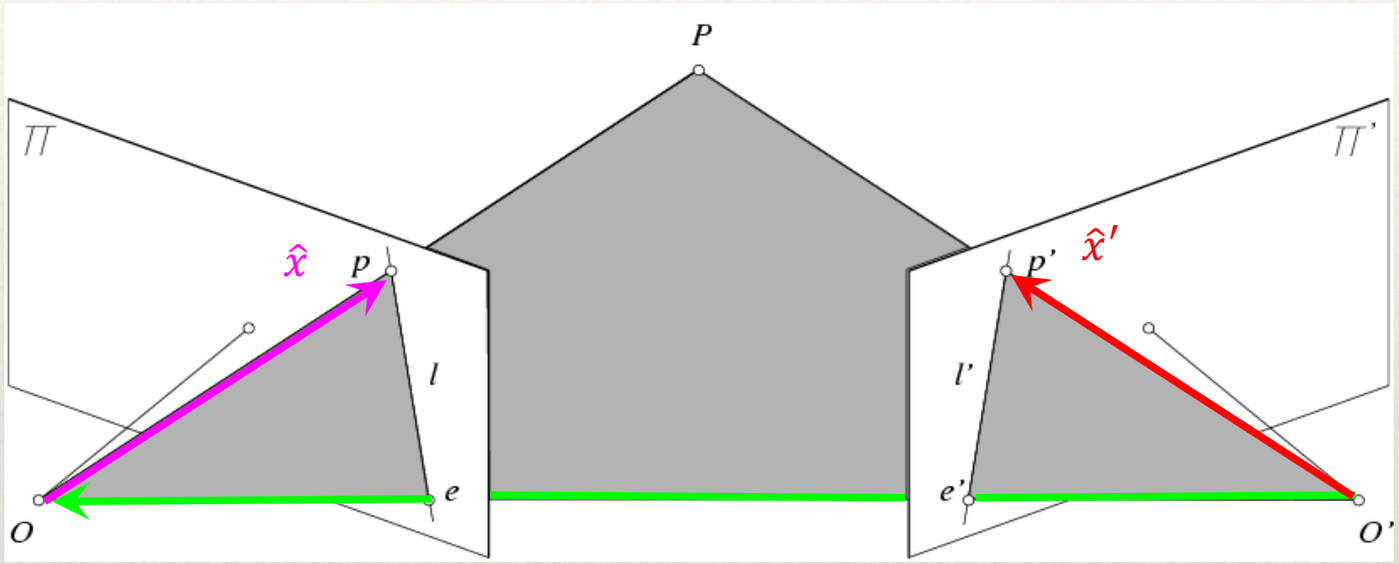
x: 2D pixel coordinate (homogeneous)

X: 3D scene point

$$\hat{x}' = K'^{-1}x' = X'$$

X': 3D scene point in 2nd camera's 3D coordinates

Epipolar constraint: calibrated case



Intrinsics *K*

t

Intrinsics *K'*

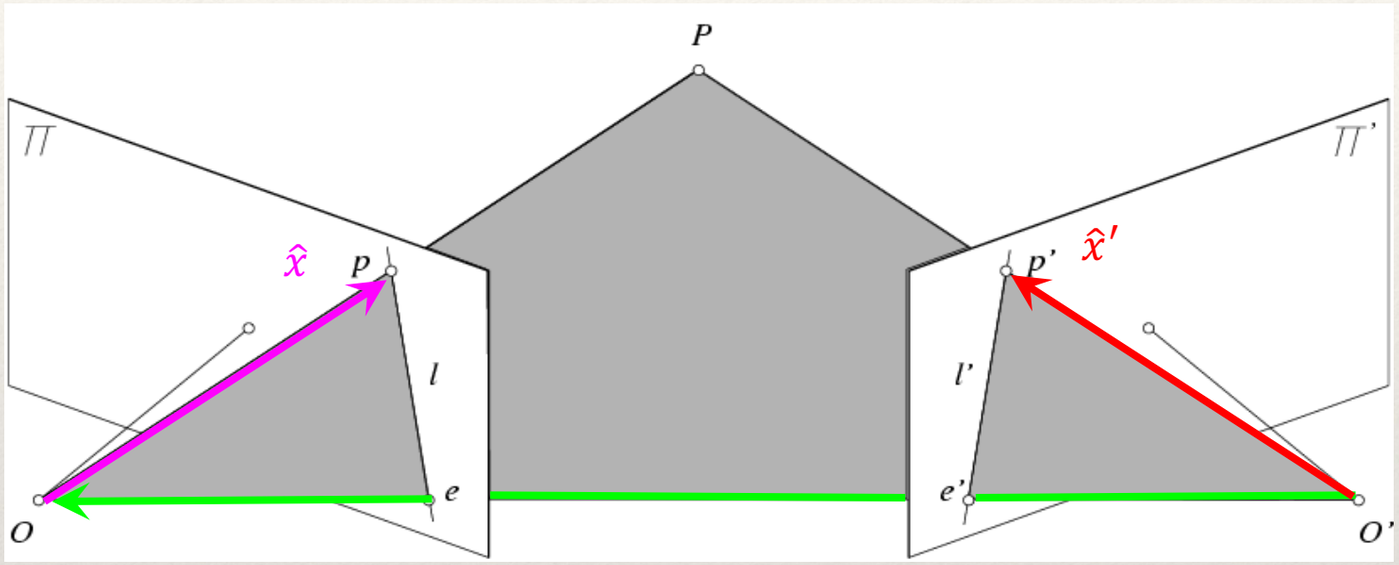
$$\hat{x} = K^{-1}x = X$$

$$\hat{x}' = K'^{-1}x' = X'$$

$$\hat{x} = R\hat{x}' + t \quad \Rightarrow \quad \hat{x} \cdot [t \times (R\hat{x}')] = 0$$

(because $\hat{x}, R\hat{x}', t$ are co-planar)

Essential matrix



$$\hat{x} \cdot [t \times (R\hat{x}')] = 0 \quad \Rightarrow \quad \hat{x}^T E \hat{x}' = 0 \quad \text{with} \quad E = [t]_\times R$$

- E is a 3x3 matrix which relates corresponding pairs of normalized homogeneous image points across pairs of images – for intrinsic *K* calibrated cameras.

Essential Matrix
(Longuet-Higgins, 1981)

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 matrix multiplication

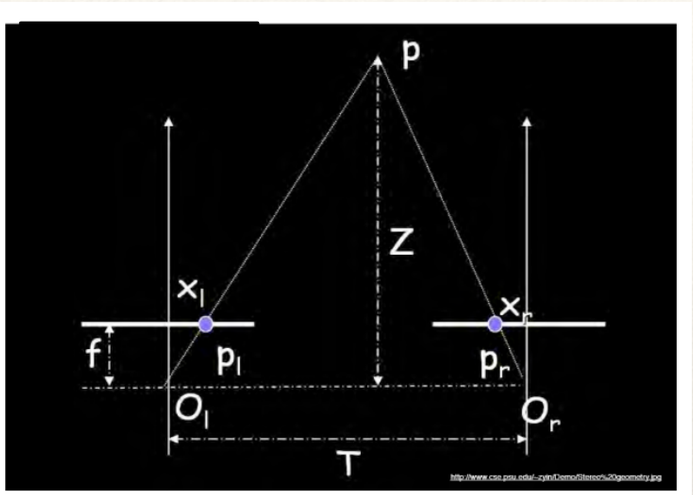
$$[a_x]=\begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

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

Properties of the essential matrix

- $E\,x'$ is the epipolar line associated with x' ($l = E\,x'$)
- $E^T x$ is the epipolar line associated with x ($l' = E^T x$)
- $E\,e' = 0$ and $E^T e = 0$
- E is singular (rank two)
- E has five degrees of freedom
  - (3 for R, 2 for t because it's up to a scale)

Essential matrix example: parallel cameras



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

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

$$\mathbf{E} = [\mathbf{T}_x]\mathbf{R} = \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{matrix}$$

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

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

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

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

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

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

$$\Leftrightarrow y = y'$$

The fundamental matrix

- Without knowing K and K’, we can define a similar relation using *unknown* normalized coordinates

$$\hat{x}^T E \hat{x}' = 0$$

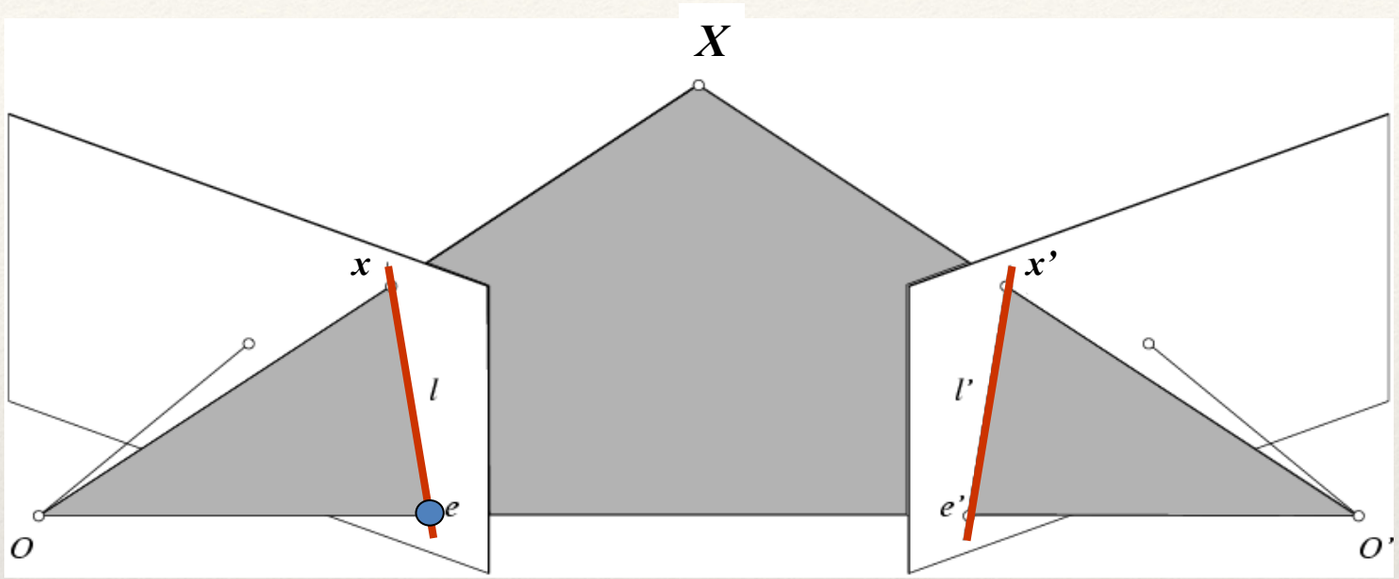
$$\hat{x} = K^{-1} x$$

$$\hat{x}' = K'^{-1} x'$$

$$\Rightarrow \quad x^T F x' = 0 \quad \text{with} \quad F = K^{-T} E K'^{-1}$$

Fundamental Matrix
(Faugeras and Luong, 1992)

Properties of the fundamental matrix



- $F\,x' = 0$ is the epipolar line l associated with x'
- $F^T x = 0$ is the epipolar line l' associated with x
- F is singular (rank two): det(F)=0
- $F\,e' = 0$ and $F^T e = 0$ (nullspaces of F = e'; nullspace of F^T = e')
- F has seven degrees of freedom: 9 entries but defined up to scale, det(F)=0

More details of F

- F is a 3x3 matrix
- Rank 2 -> projection; one column is a linear combination of the other two.
- Determined up to scale.
- 7 degrees of freedom

$$\begin{matrix} a & b & \alpha a + \beta b \\ c & d & \alpha c + \beta d \\ e & f & \alpha e + \beta f \end{matrix}$$

where α is scalar; e.g., can normalize out

- Given x projected from X into image 1, F constrains the projection of x′ into image 2 to an epipolar line.

Why epipolar constraint is useful?

- Reduce search space for stereo disparity estimation.
- Help find *x'*: If I know *x*, and have calibrated cameras (known intrinsics K,K' and extrinsic relationship), I can restrict *x'* to be along *l'*.

Image from Andrew Zisserman

What about when cameras' optical axes are not parallel?

image I(x,y)

Disparity map D(x,y)

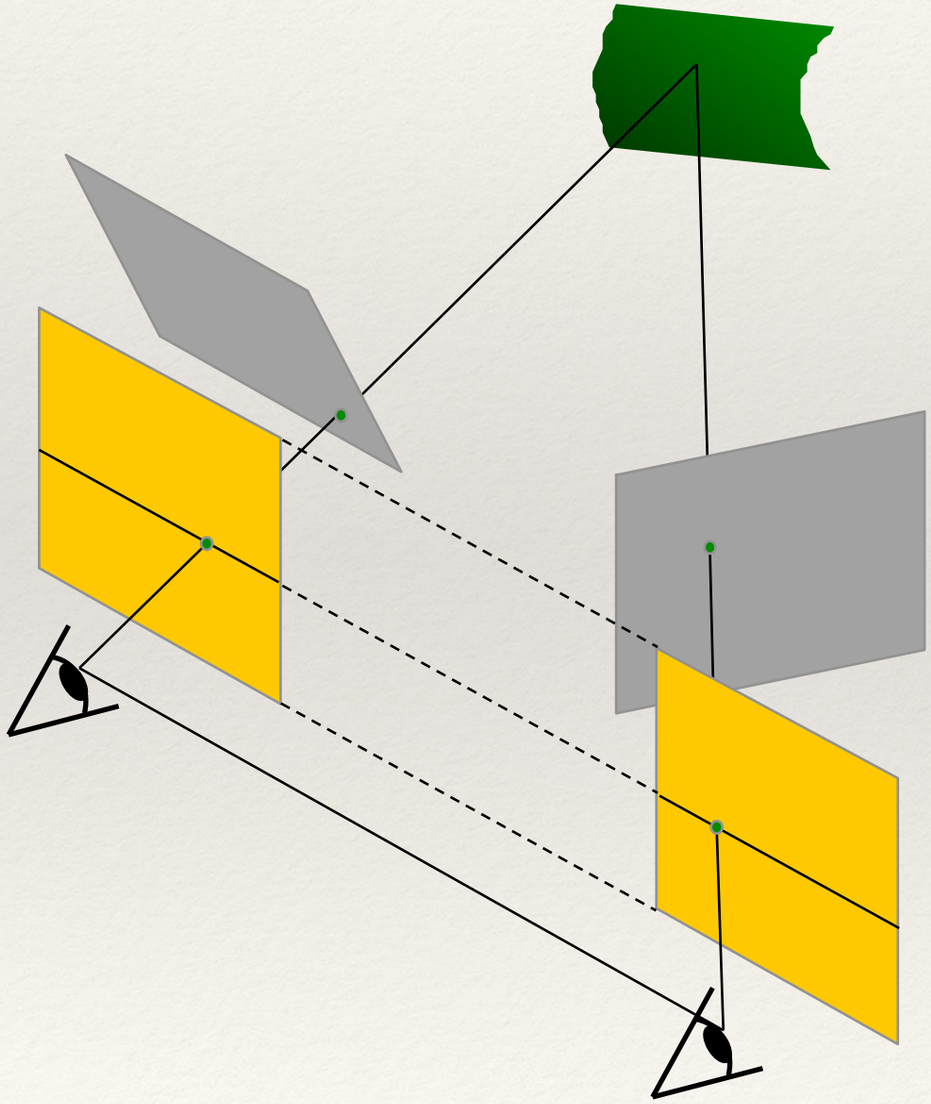
image I´(x´,y´)

$(x´,y´)=(x+D(x,y),y)$

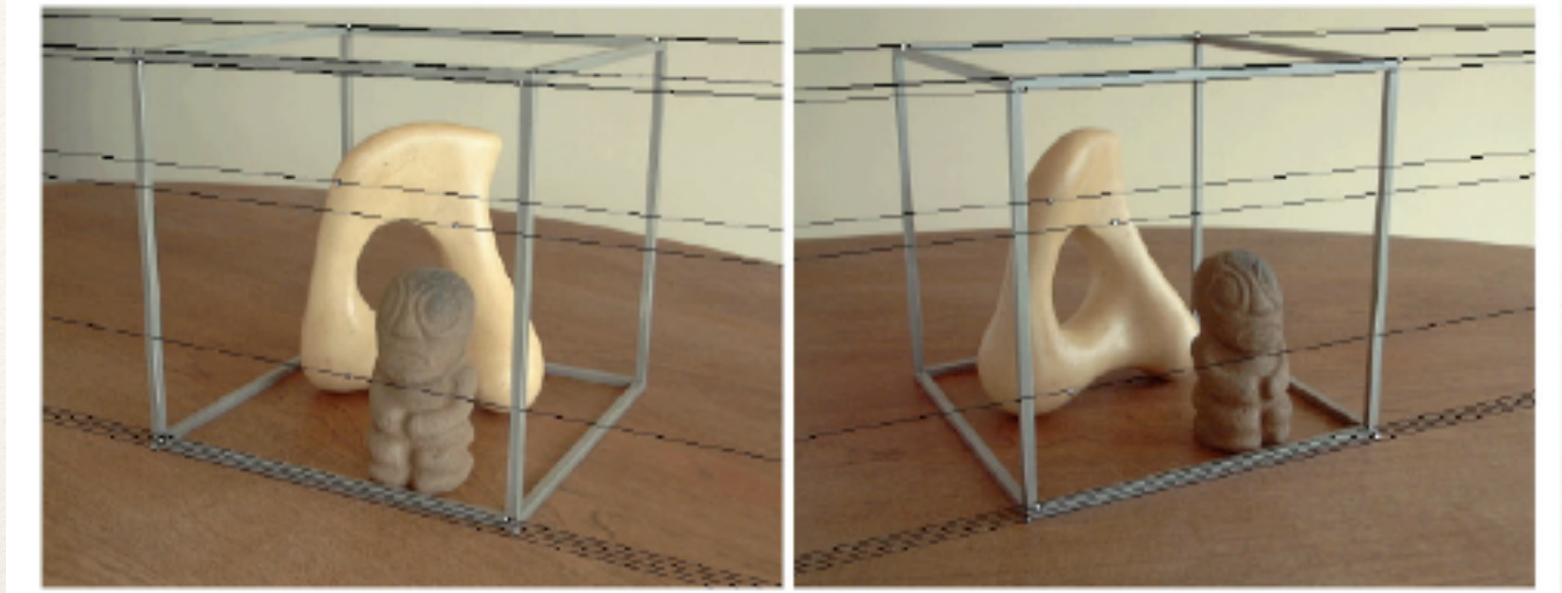
Slide credit: Kristen Grauman

Stereo 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



Stereo matching

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

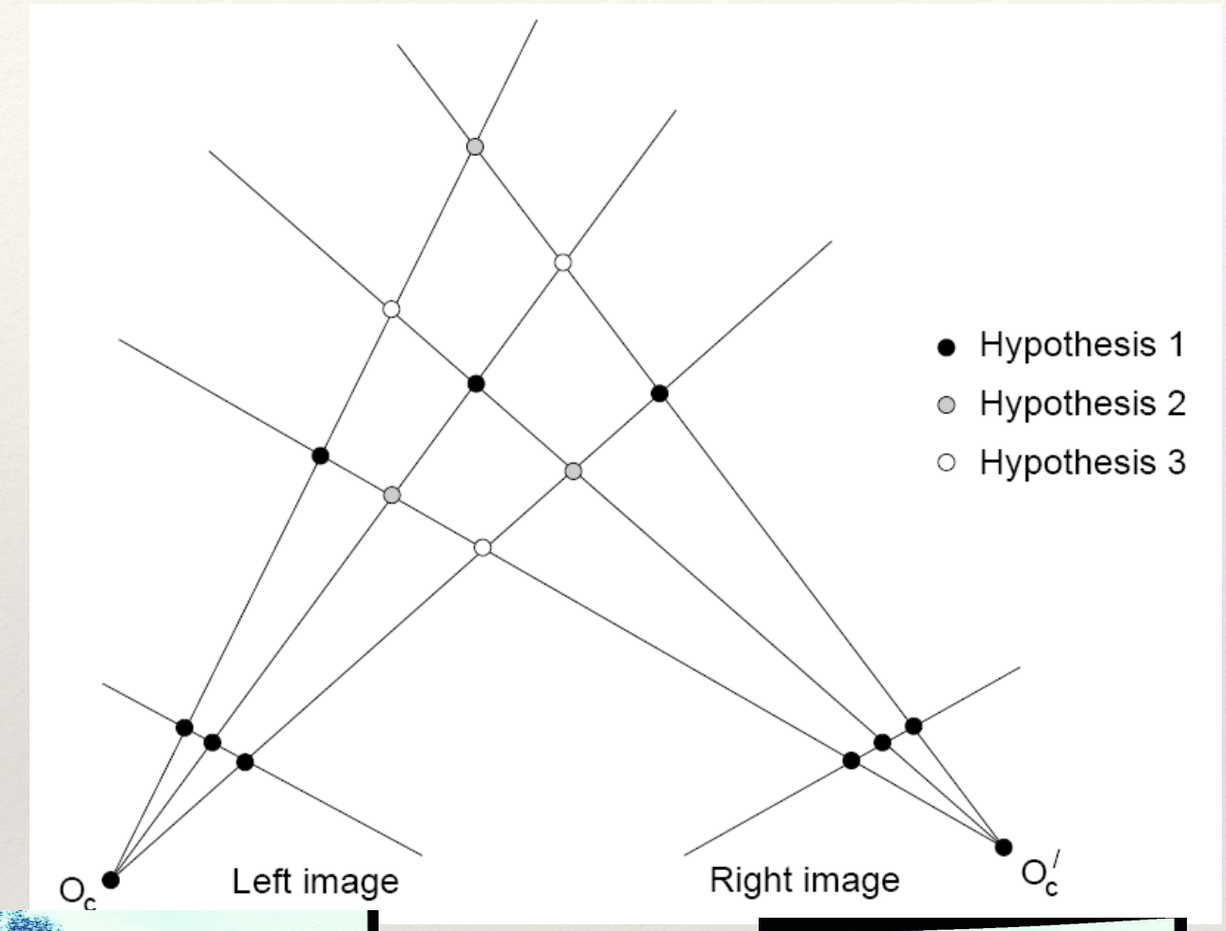
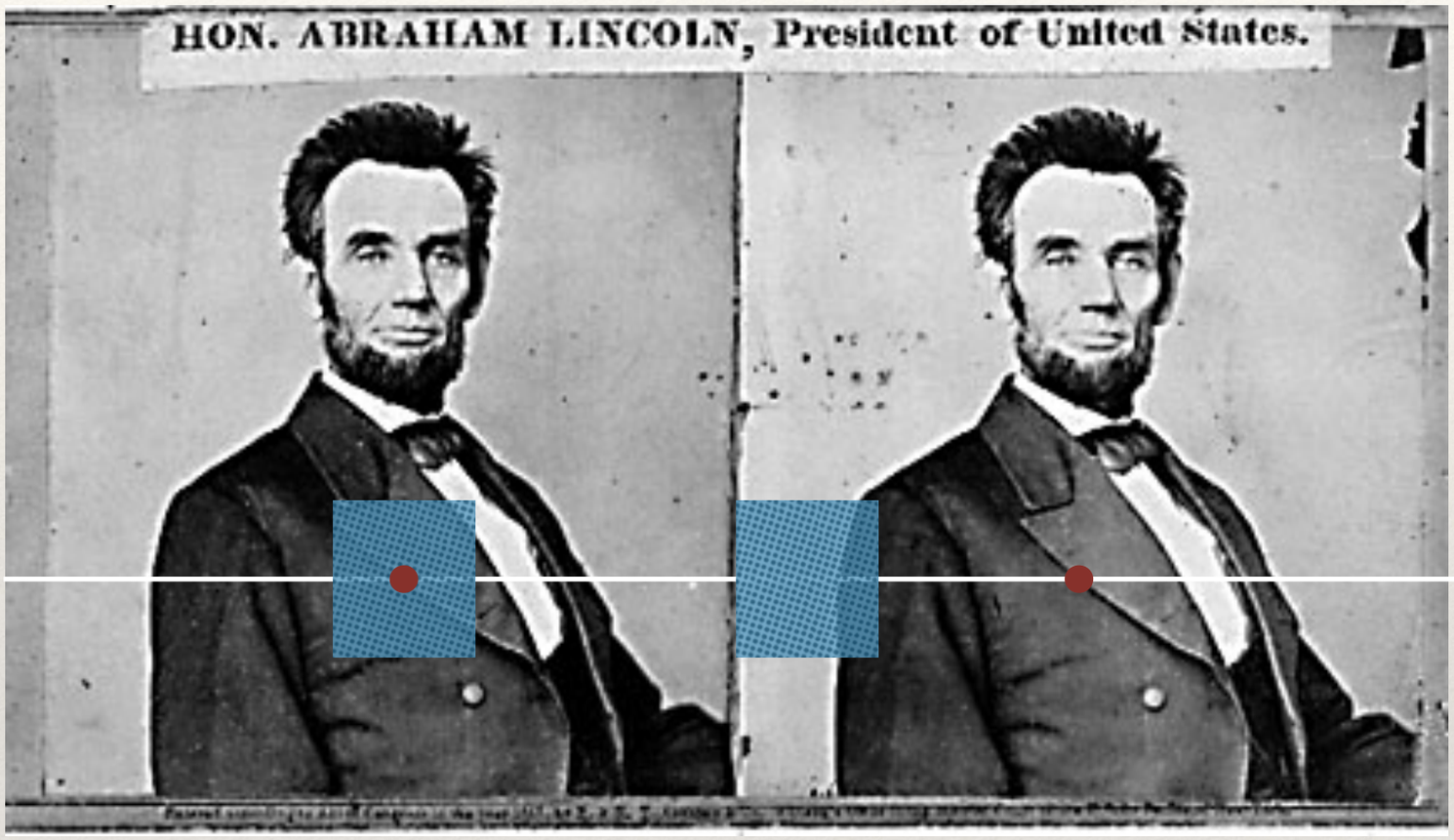


Figure from Gee & Cipolla 1999

Stereo matching

- Beyond the hard constraint of epipolar geometry, there are "soft" constraints to help identify corresponding points
  - Similarity
  - Uniqueness
  - Ordering
  - Disparity gradient
- To find matches in the image pair, we will assume
  - Most scene points visible from both views
  - Image regions for the matches are similar in appearance

Dense correspondence search



- For each epipolar line
  - For each pixel / window in the left image:
    - Compare with every pixel / window on same epipolar line in right image
    - Pick position with minimum match cost (e.g., SSD, normalized correlation)

Normalized correlation

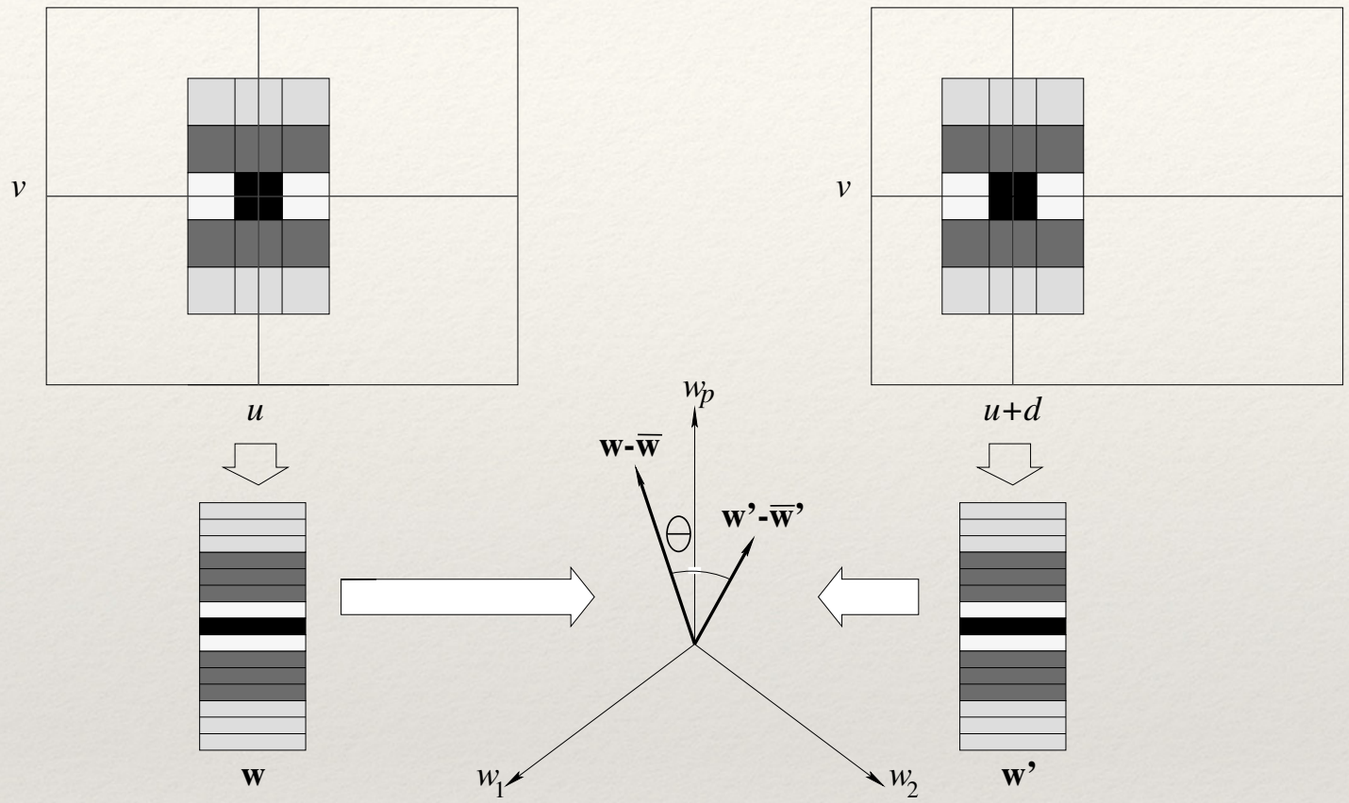
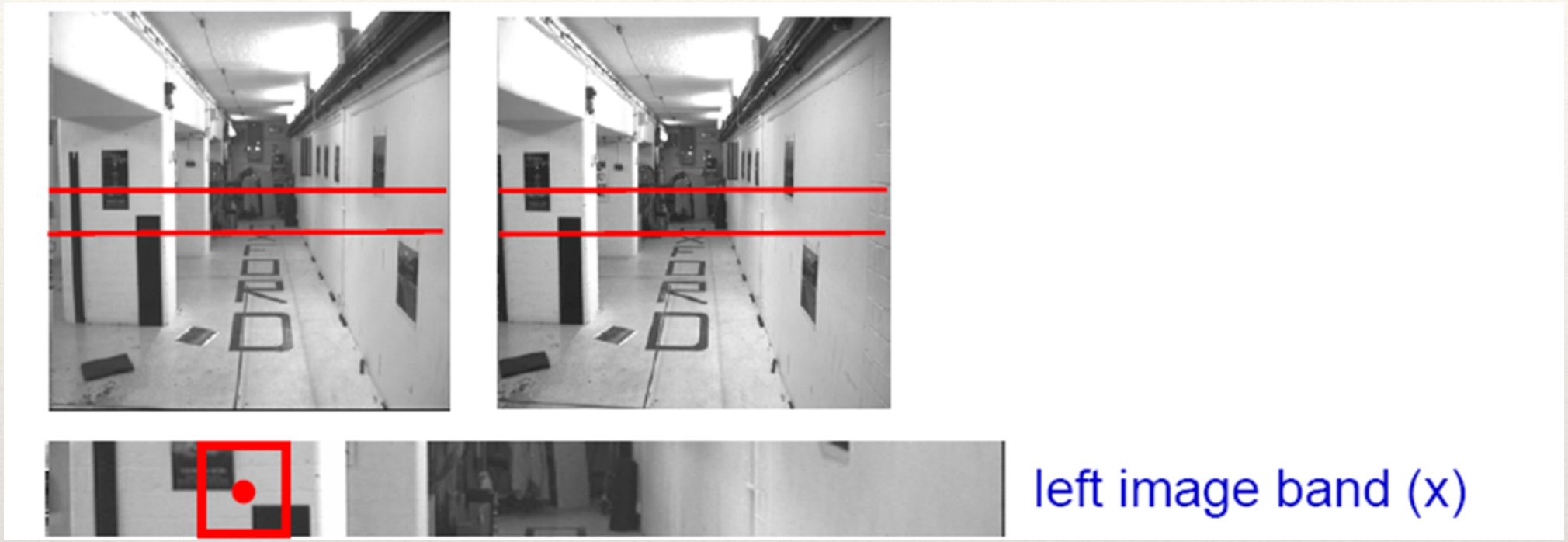


FIGURE 7.9: Correlation of two 3×5 windows along corresponding epipolar lines. The second window position is separated from the first one by an offset d. The two windows are encoded by vectors $\boldsymbol{w}$ and $\boldsymbol{w}'$ in $\mathbb{R}^{15}$, and the correlation function measures the cosine of the angle θ between the vectors $\boldsymbol{w} - \bar{\boldsymbol{w}}$ and $\boldsymbol{w}' - \bar{\boldsymbol{w}}'$ obtained by subtracting from the components of $\boldsymbol{w}$ and $\boldsymbol{w}'$ the average intensity in the corresponding windows.

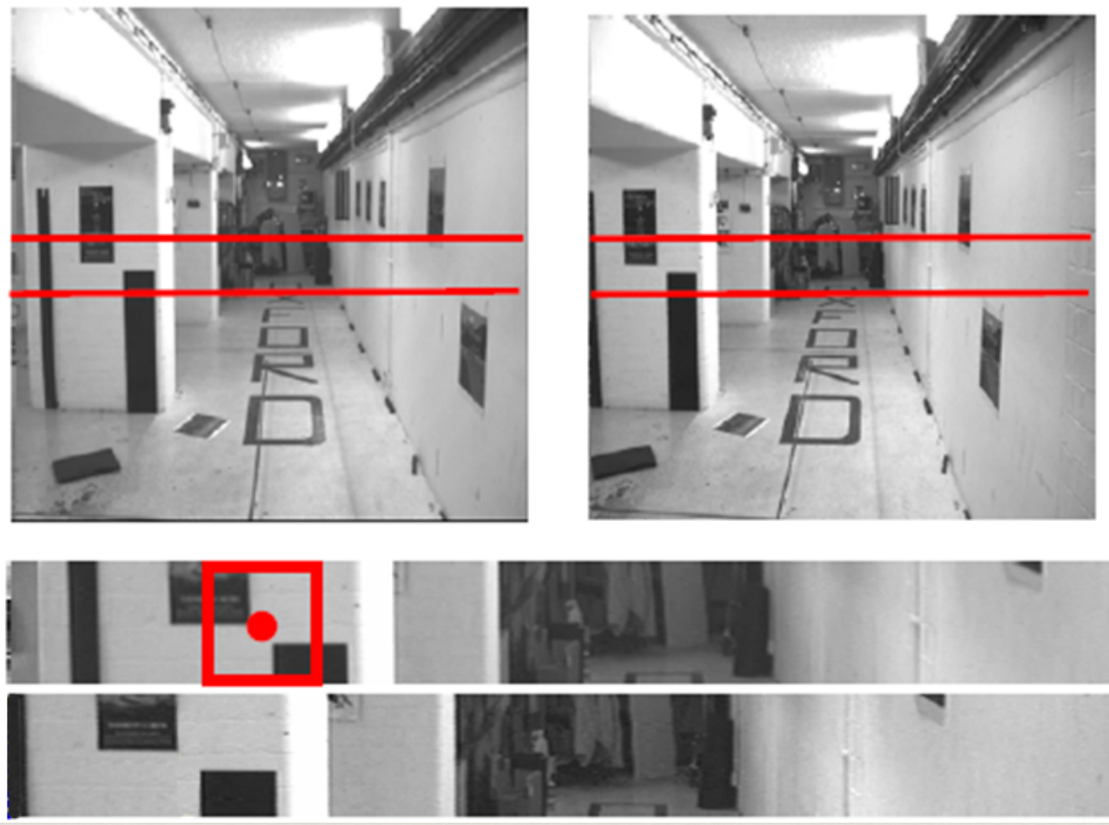
$$C(d) = \frac{1}{||\boldsymbol{w} - \bar{\boldsymbol{w}}||} \frac{1}{||\boldsymbol{w}' - \bar{\boldsymbol{w}}'||} [(\boldsymbol{w} - \bar{\boldsymbol{w}}) \cdot (\boldsymbol{w}' - \bar{\boldsymbol{w}}')],$$

Correlation-based window matching



Source: Andrew Zisserman

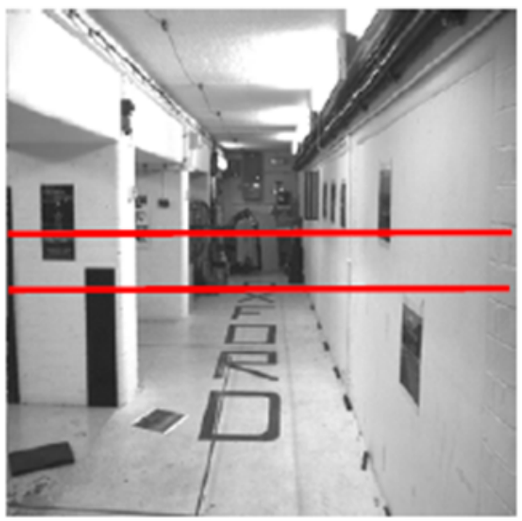
Correlation-based window matching



left image band (x)

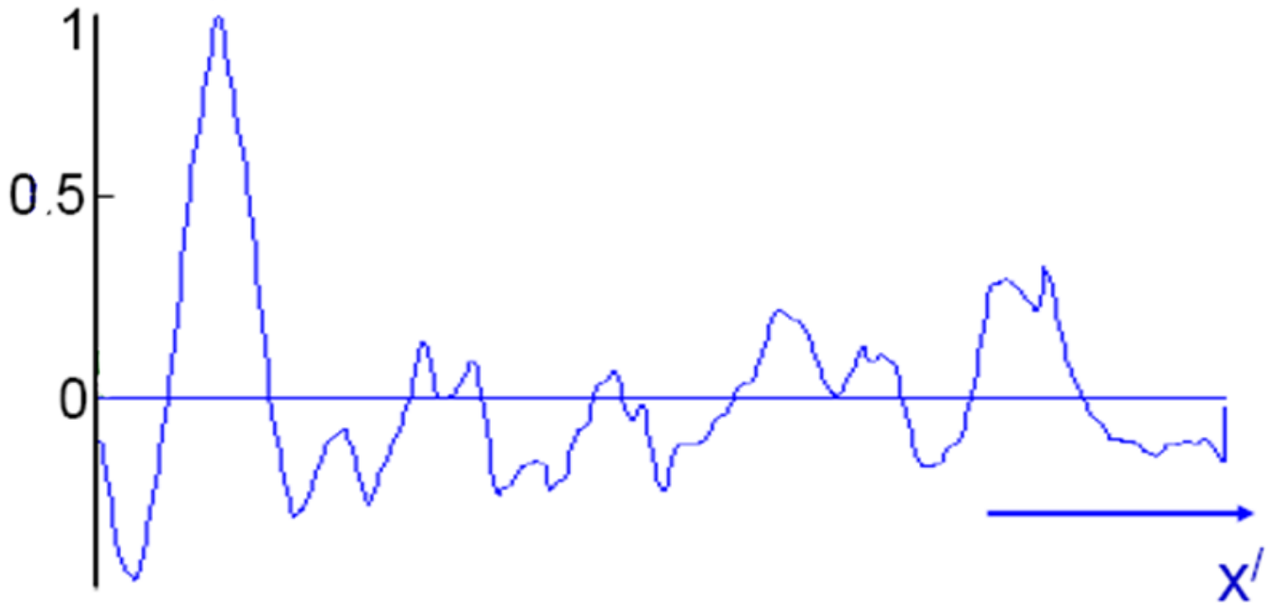
right image band ($x^{/}$)

Correlation-based window matching



left image band (x)

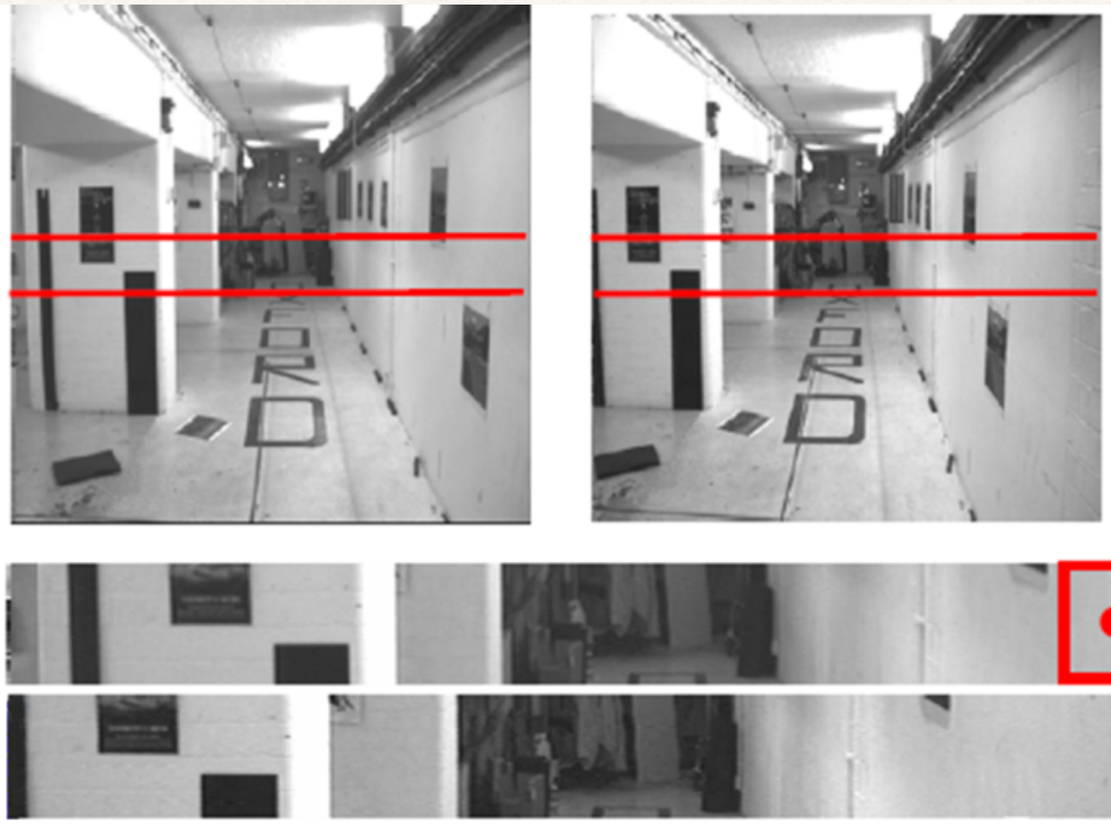
right image band (x')



cross correlation

disparity = $x' - x$

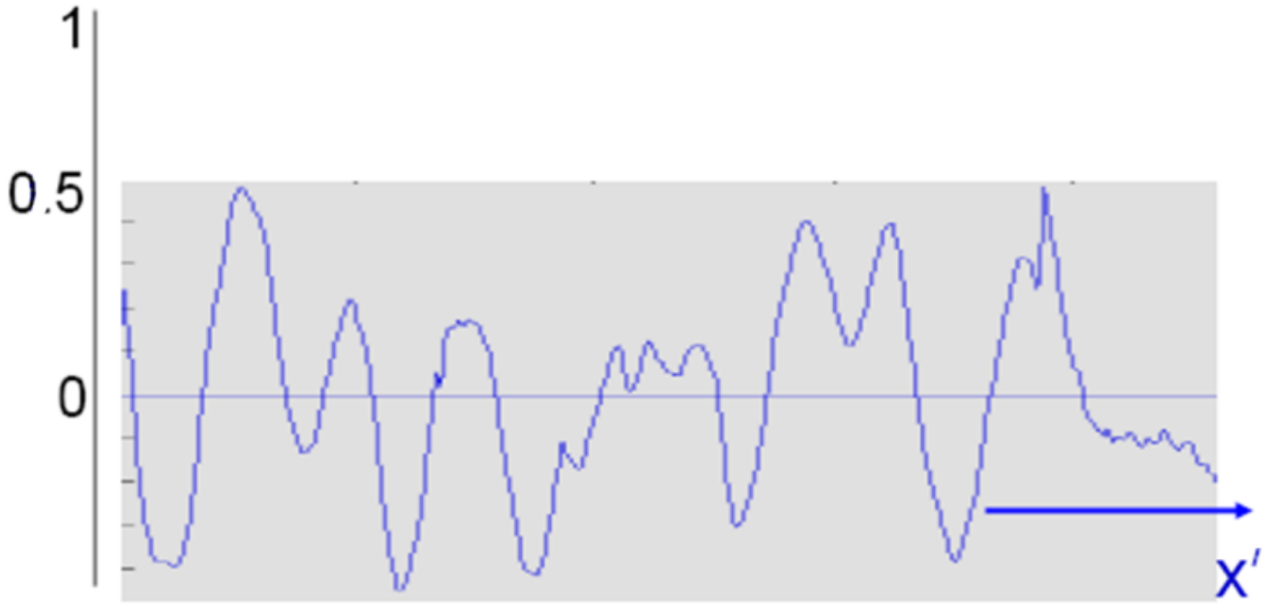
Correlation-based window matching



target region

left image band (x)

right image band (x')

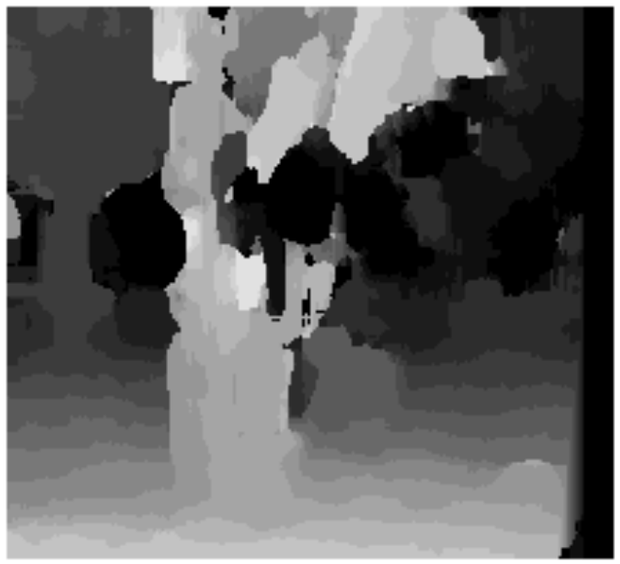


cross correlation

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

Effect of window size

W = 3

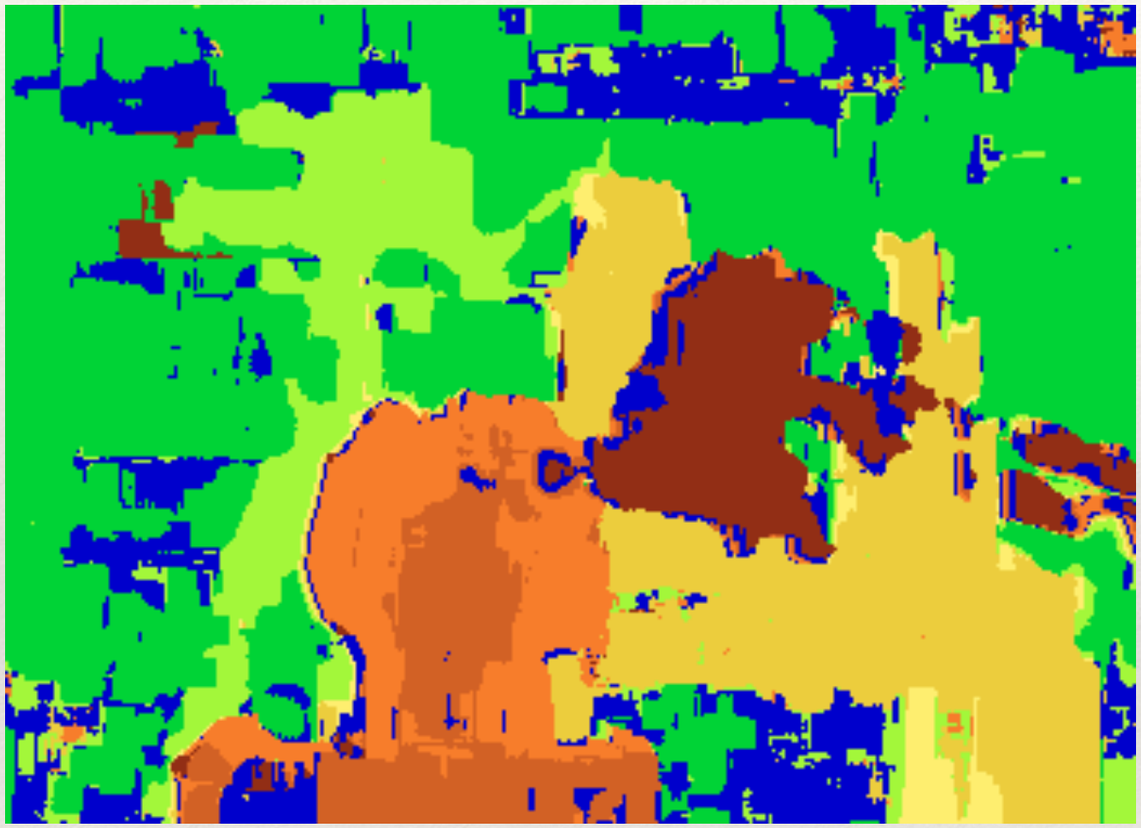


W = 20

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

Stereo – Tsukuba test scene

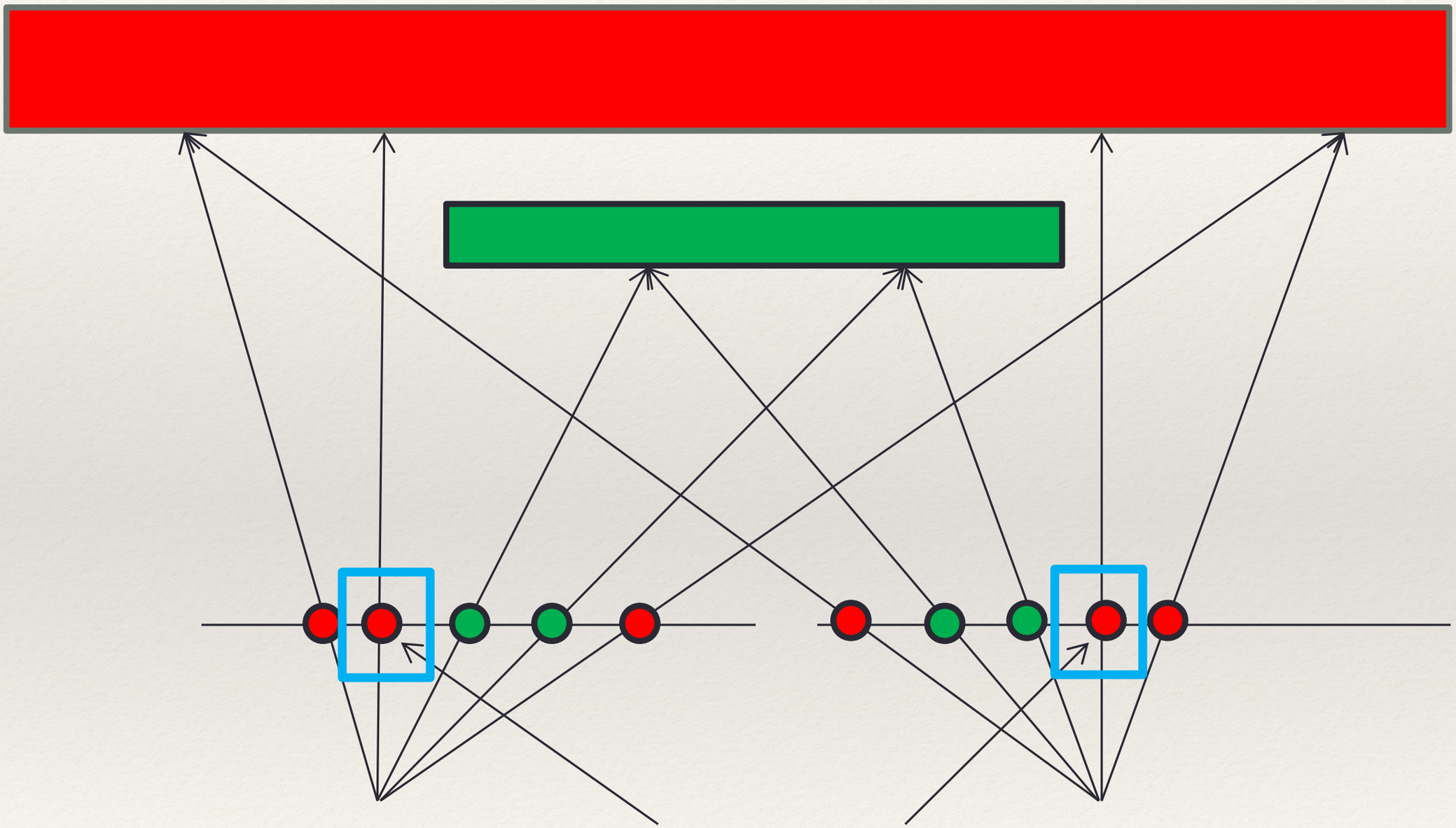
Results with window search



Window-based matching
(best window size)

‘Ground truth’

Problem: occlusion



Occluded pixels

Disparity gradient constraint

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

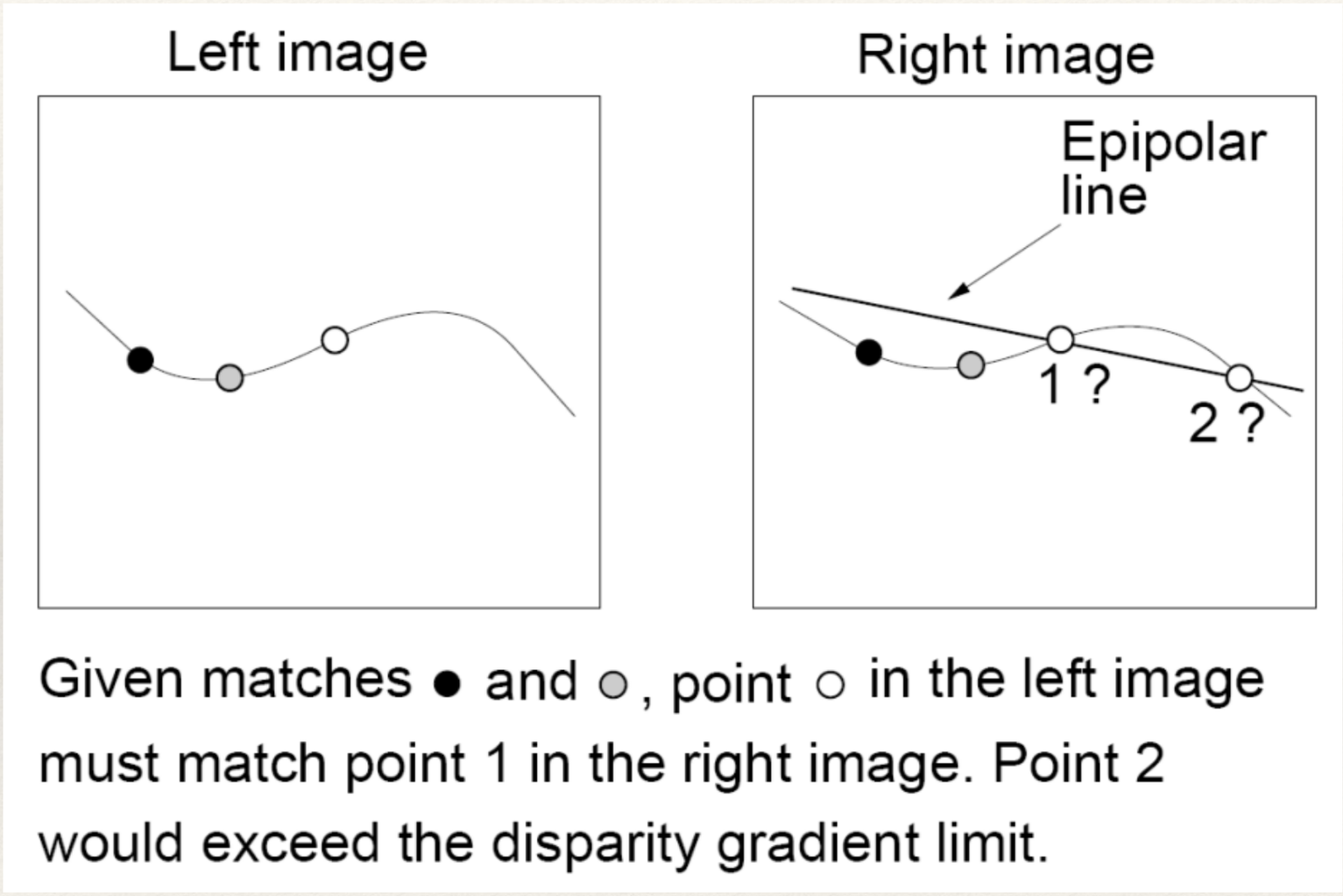


Figure from Gee & Cipolla

Ordering constraint

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

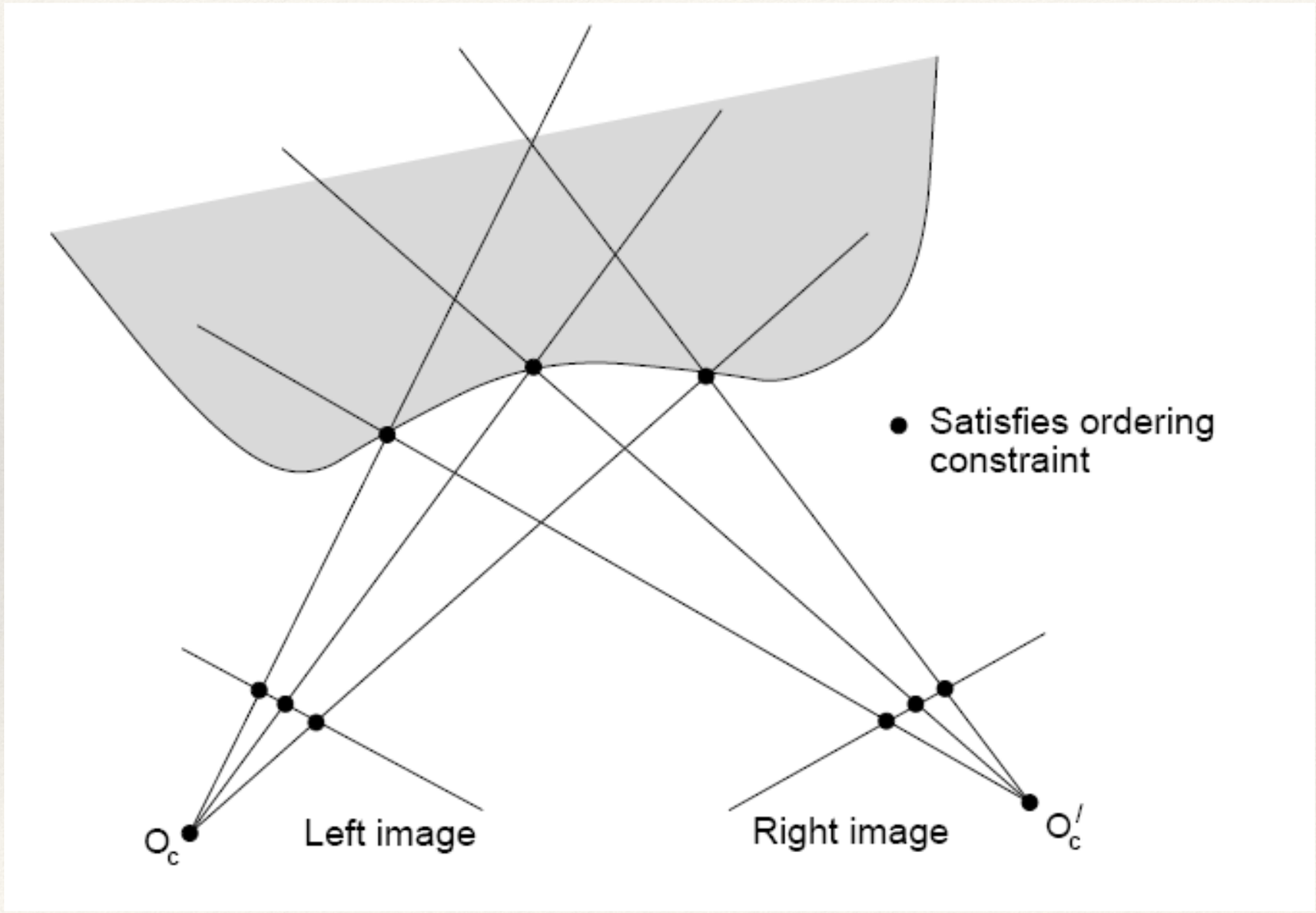
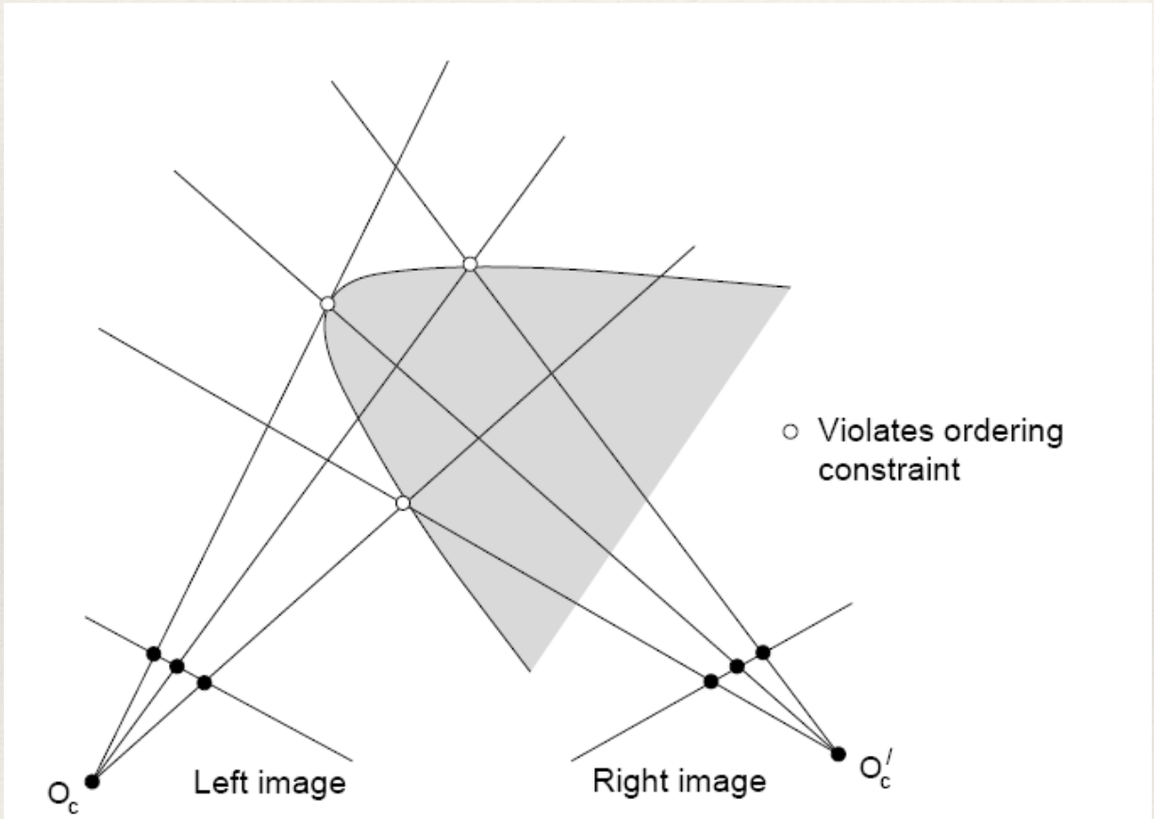
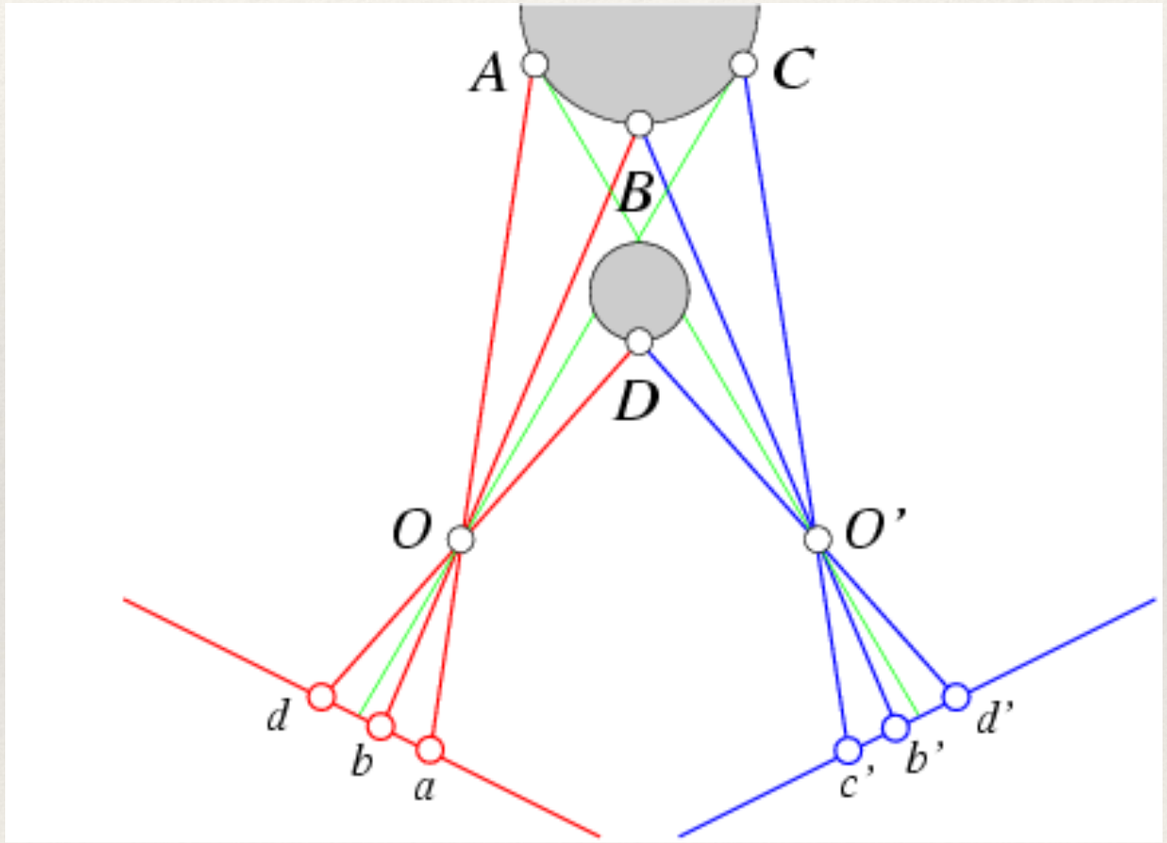


Figure from Gee & Cipolla

Ordering constraint

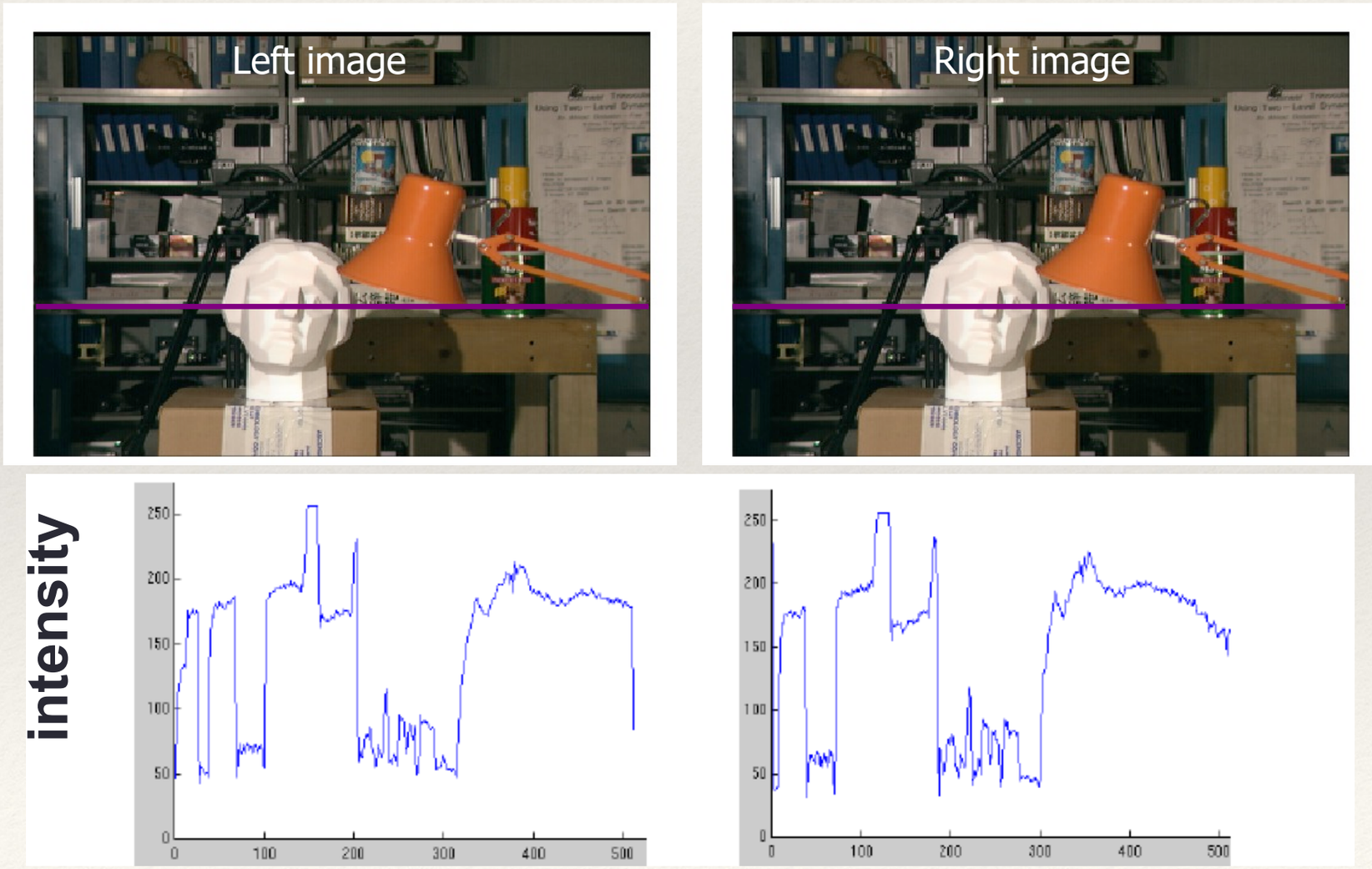
- Won’t always hold, e.g. consider transparent object, or an occluding surface



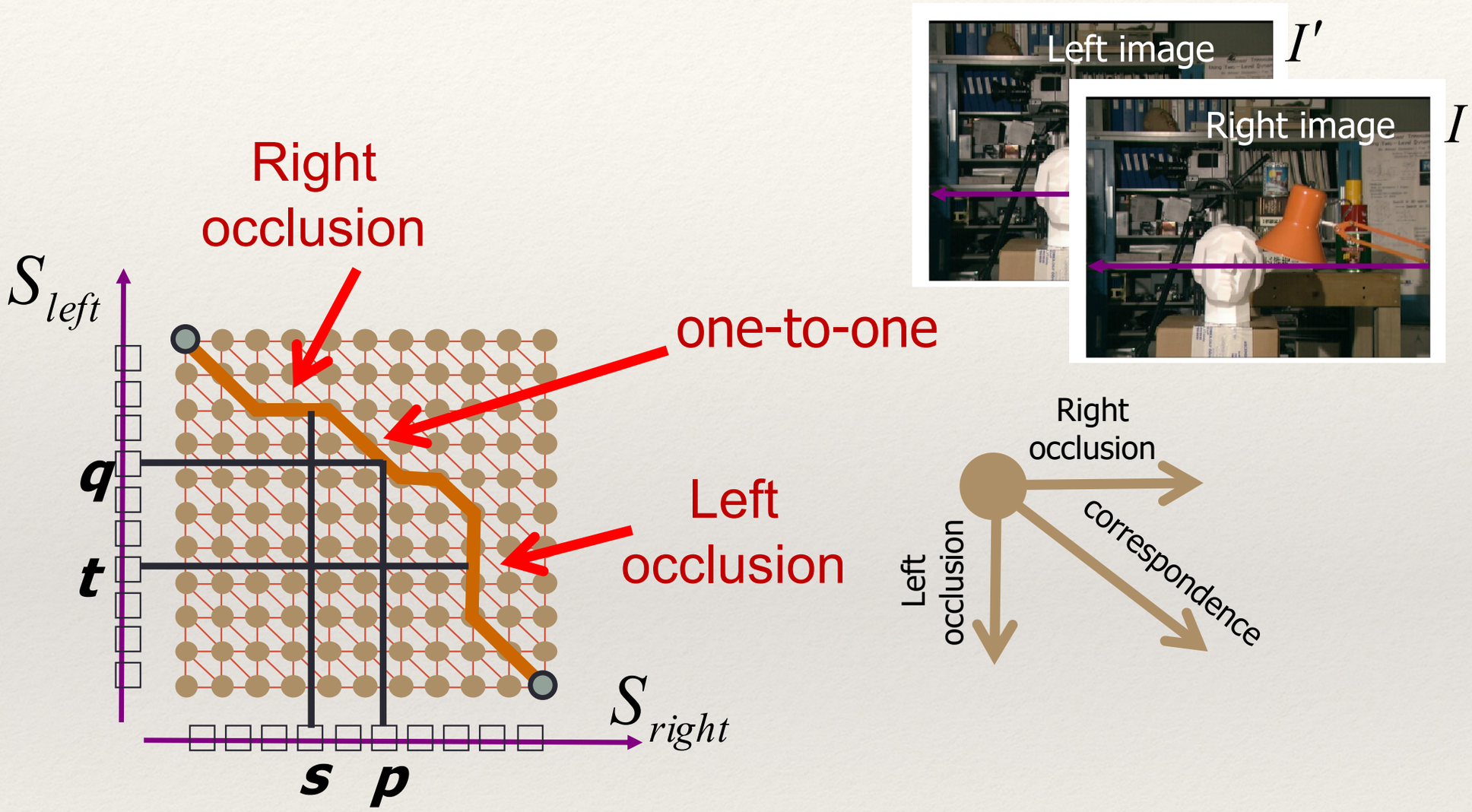


Global method - dynamical programing

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



"Shortest paths" for scan-line stereo



Can be implemented with dynamic programming

Ohta & Kanade '85, Cox et al. '96, Intille & Bobick, '01

Slide credit: Y. Boykov

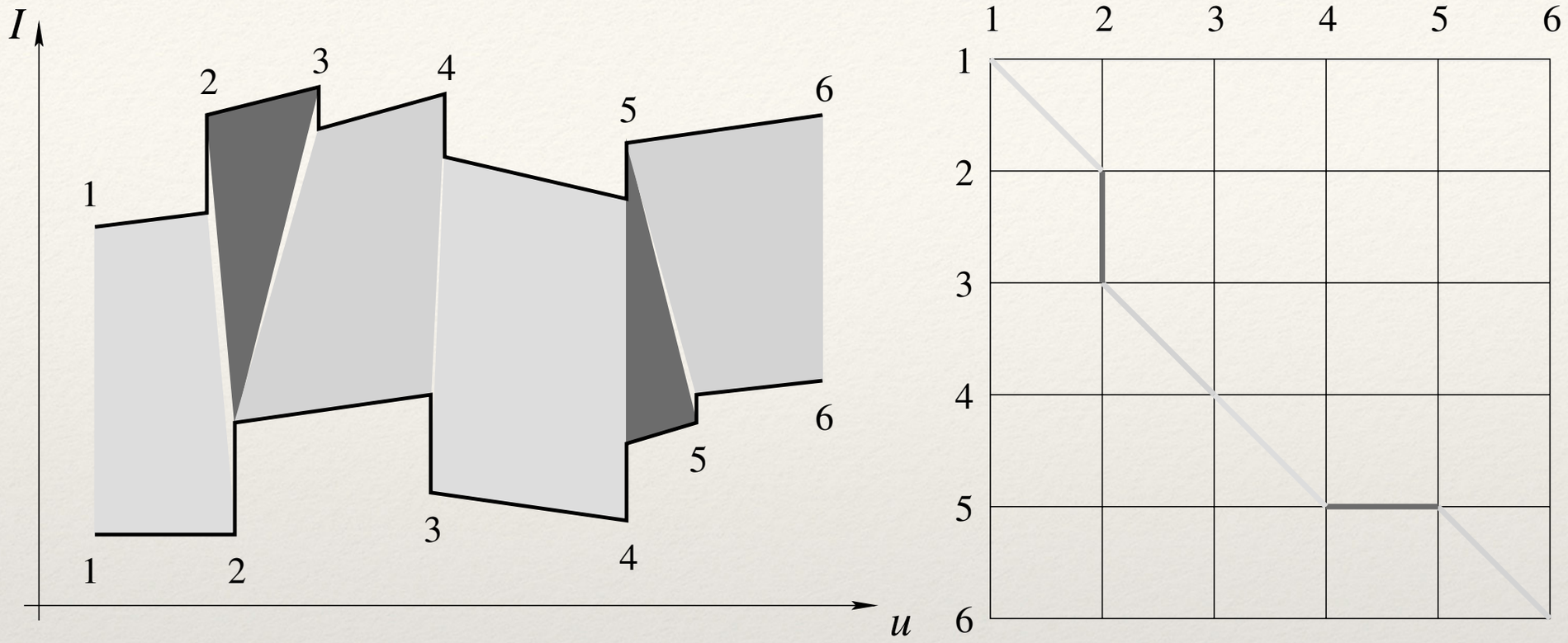


FIGURE 7.14: Dynamic programming and stereopsis: The **left** part of the figure shows two intensity profiles along matching epipolar lines. The polygons joining the two profiles indicate matches between successive intervals (some of the matched intervals may have zero length). The **right** part of the diagram represents the same information in graphical form: an arc (thick line segment) joins two nodes (i, i') and (j, j') when the intervals (i, j) and (i', j') of the intensity profiles match each other.

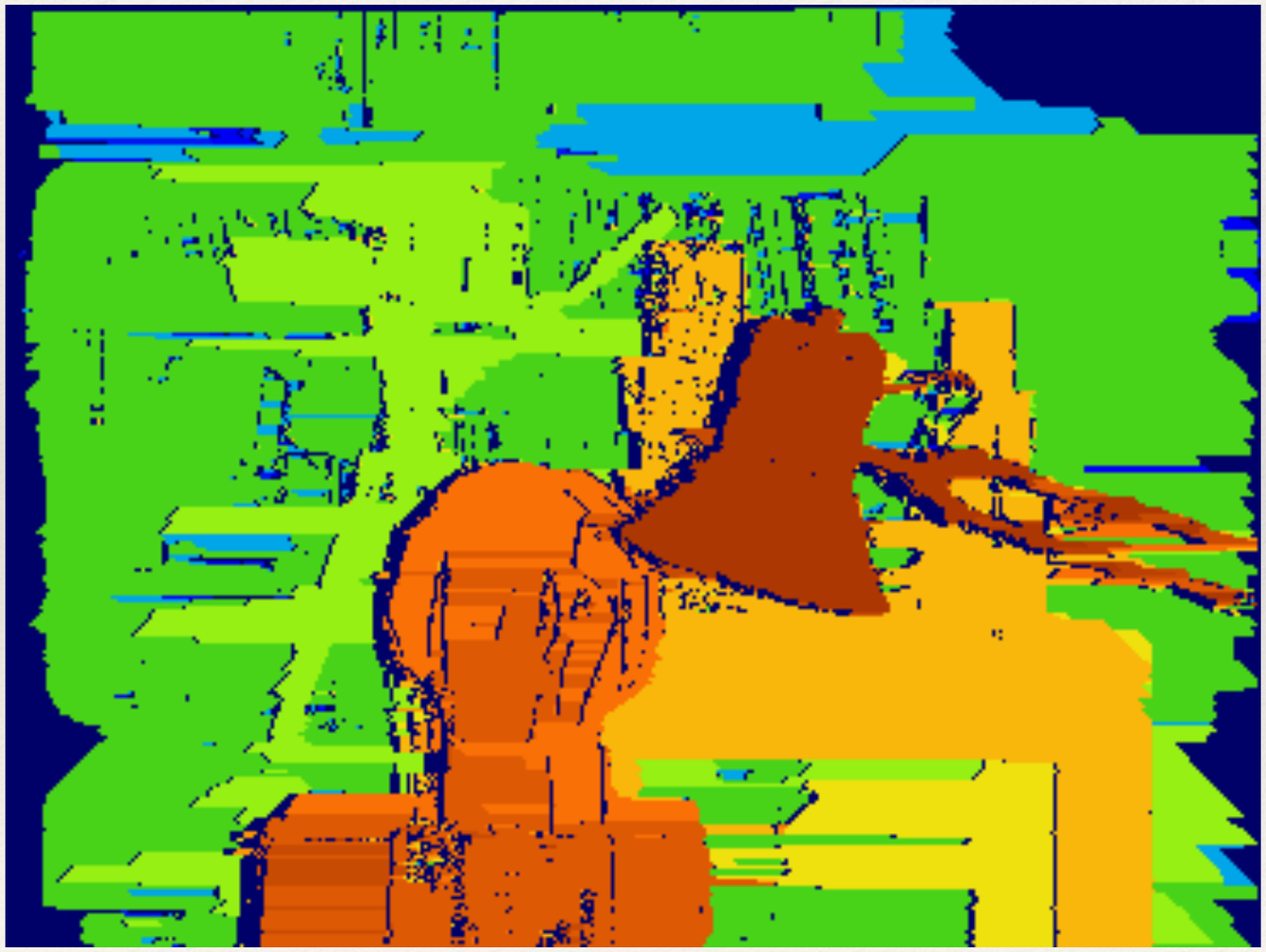
We assume the scanlines have m and n edge points, respectively (the endpoints of the scanlines are included for convenience). Two auxiliary functions are used: Inferior-Neighbors(k, l) returns the list of neighbors (i, j) of the node (k, l) such that $i \leq k$ and $j \leq l$, and Arc-Cost(i, j, k, l) evaluates and returns the cost of matching the intervals (i, k) and (j, l). For correctness, $C(1, 1)$ should be initialized with a value of zero.

% Loop over all nodes (k, l) in ascending order.
for $k = 1$ to m do
  for $l = 1$ to n do
    % Initialize optimal cost $C(k, l)$ and backward pointer $B(k, l)$.
    $C(k, l) \leftarrow +\infty$; $B(k, l) \leftarrow$ nil;
    % Loop over all inferior neighbors (i, j) of (k, l).
    for $(i, j) \in$ Inferior-Neighbors(k, l) do
      % Compute new path cost and update backward pointer if necessary.
      $d \leftarrow C(i, j) +$ Arc-Cost(i, j, k, l);
      if $d < C(k, l)$ then $C(k, l) \leftarrow d$; $B(k, l) \leftarrow (i, j)$ endif;
      endfor;
    endfor;
  endfor;
% Construct optimal path by following backward pointers from (m, n).
$P \leftarrow \{(m, n)\}$; $(i, j) \leftarrow (m, n)$;
while $B(i, j) \neq$ nil do $(i, j) \leftarrow B(i, j)$; $P \leftarrow \{(i, j)\} \cup P$ endwhile.

Algorithm 7.2: A Dynamic-Programming Algorithm for Establishing Stereo Correspondences Between Two Corresponding Scanlines.

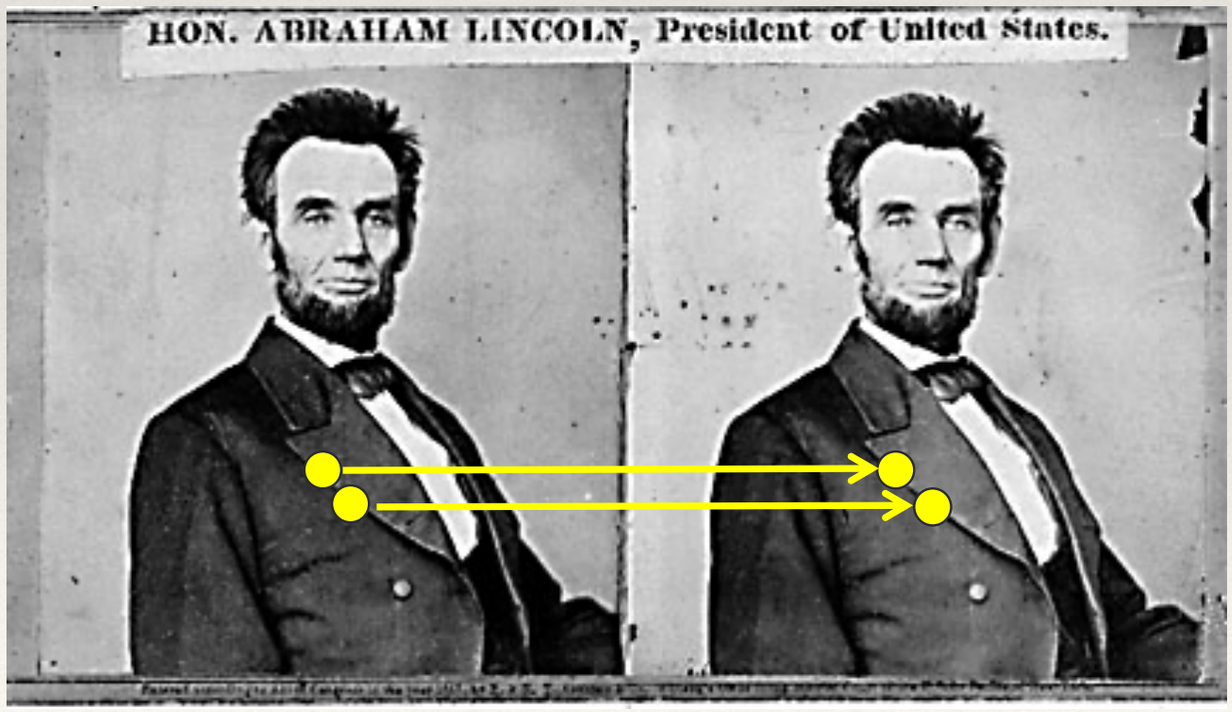
Scanline stereo generates streaking artifacts

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

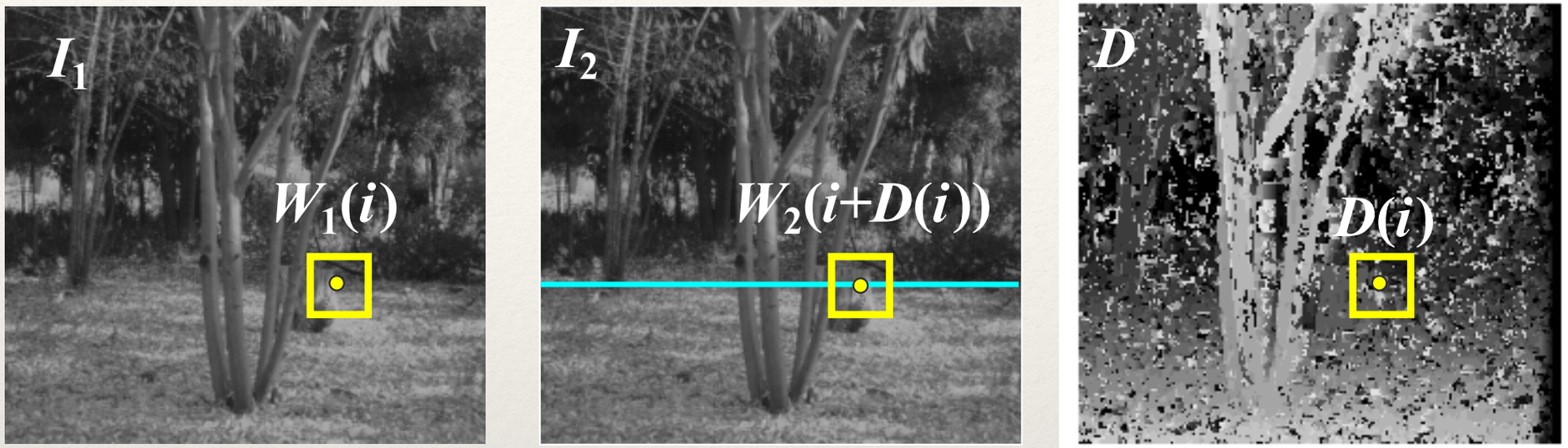


Global method - energy minimization

- What defines a good stereo correspondence?
  - Match quality
    - Want each pixel to find a good match in the other image
  - Smoothness
    - If two pixels are adjacent, they should (usually) move about the same amount



Global method - energy minimization



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

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

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

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

Y. Boykov, O. Veksler, and R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, PAMI 2001

Source: Steve Seitz

Better results

Graph cut method

Ground truth

Boykov et al., Fast Approximate Energy Minimization via Graph Cuts, International Conference on Computer Vision, September 1999.

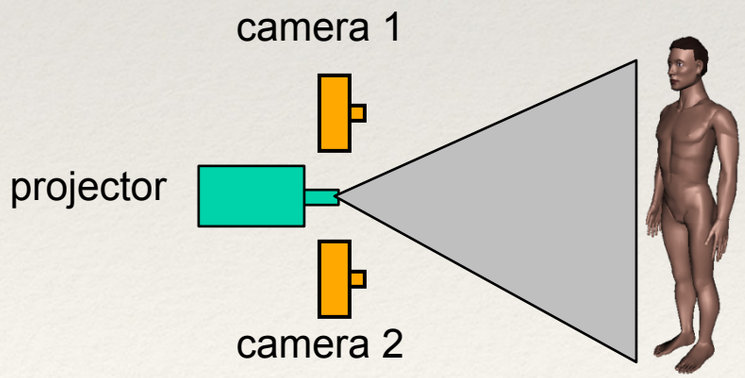
For the latest and greatest: http://www.middlebury.edu/stereo/

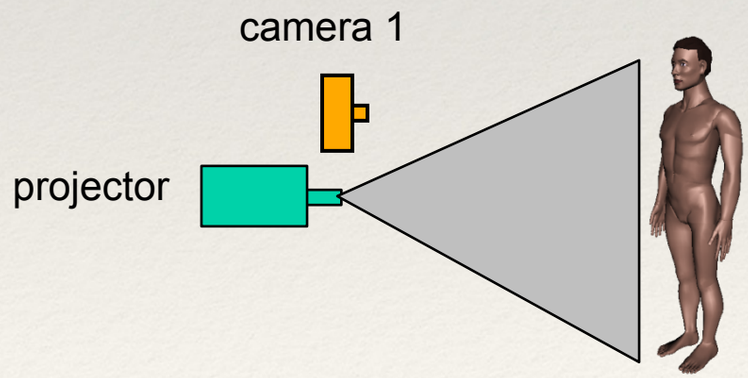
Challenges

- Low-contrast ‘textureless’ image regions
- Occlusions
- Violations of brightness constancy
- Specular reflections
- Really large baselines
- Foreshortening and appearance change
- Camera calibration errors

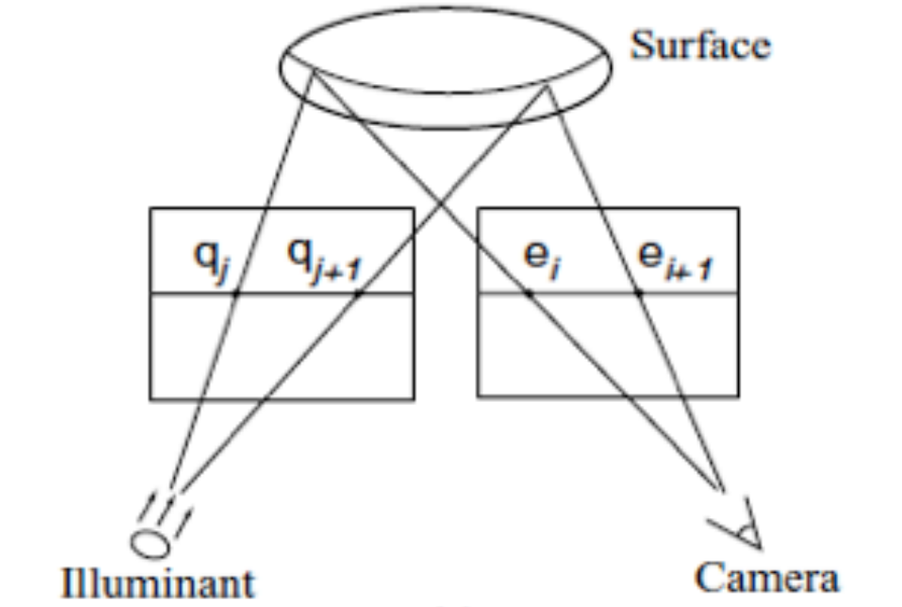
Active stereo: structured light

Project “structured” light patterns onto the object simplifies the correspondence problem



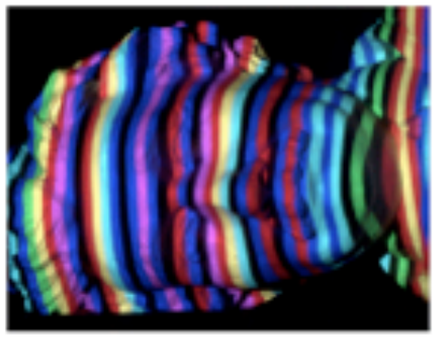


Slide credit: Rick Szeliski

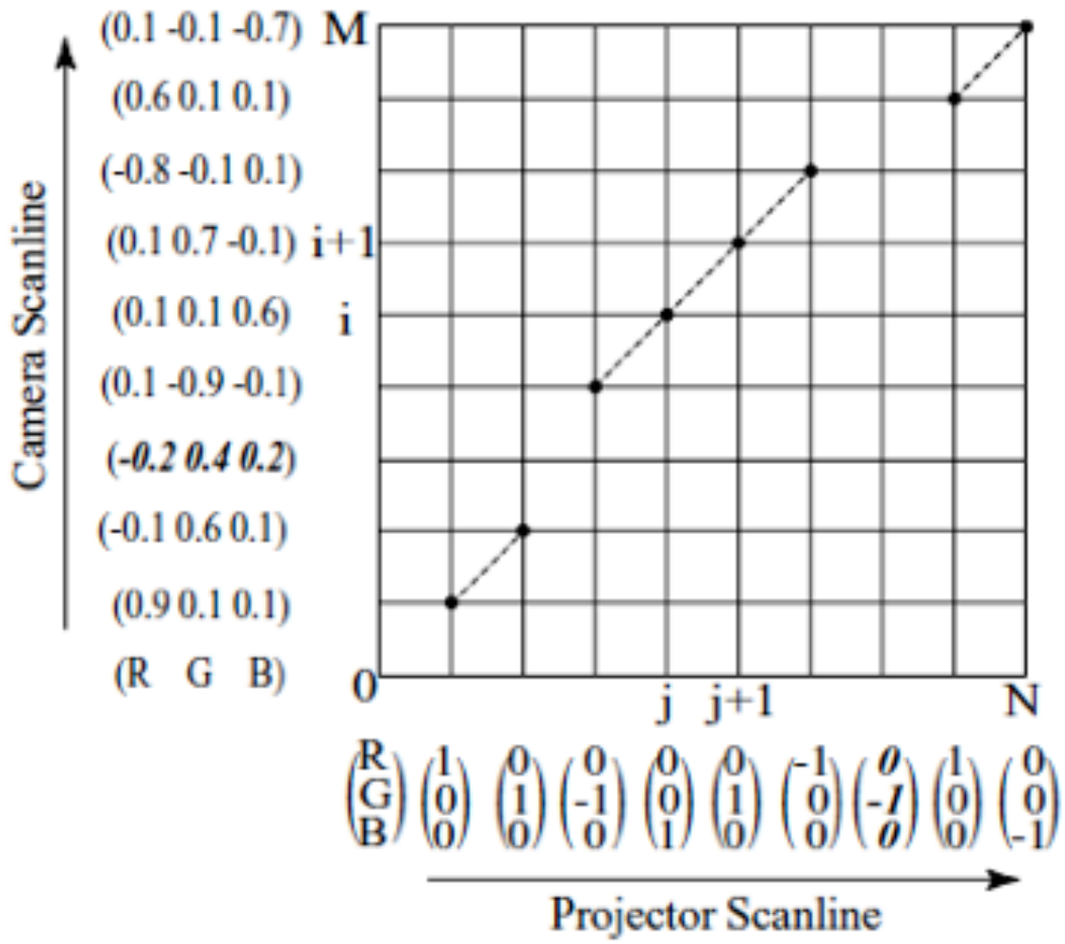


(a)

(b)



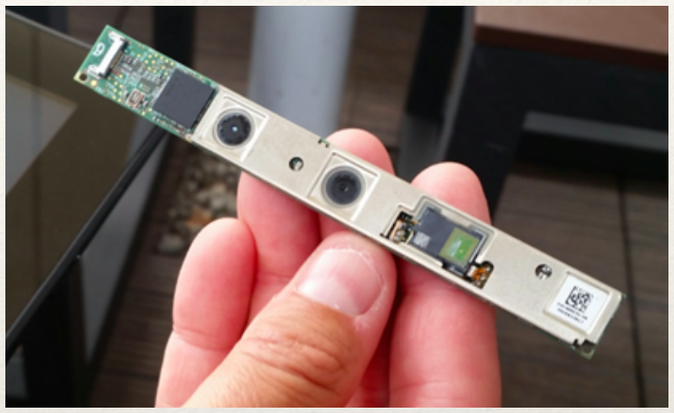
(c)



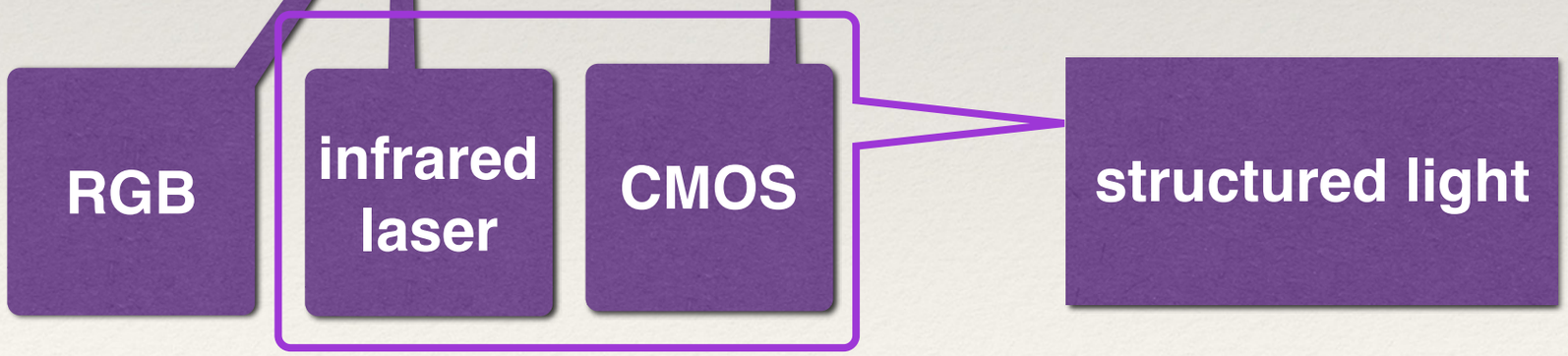
(d)

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.

Kinect-v1: structured infrared light

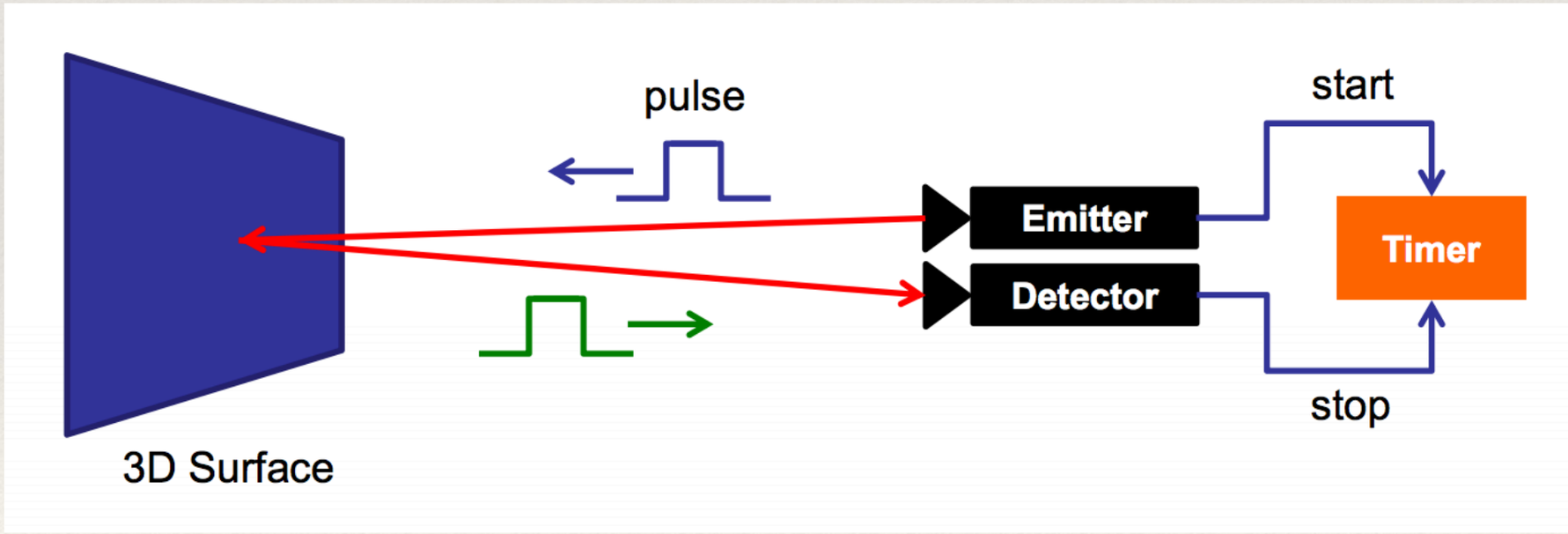


Intel laptop depth camera



Time of flight (Kinect - v2)

- Depth cameras in HoloLens use *time of flight*
  - "SONAR for light"
  - Emit light of a known wavelength, and time how long it takes for it to come back



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