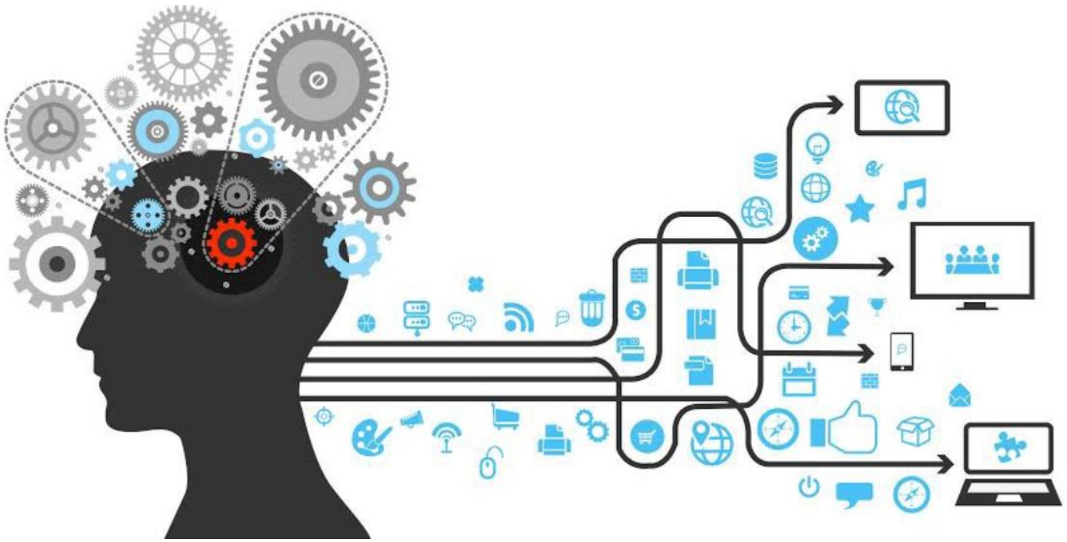
Computer Vision

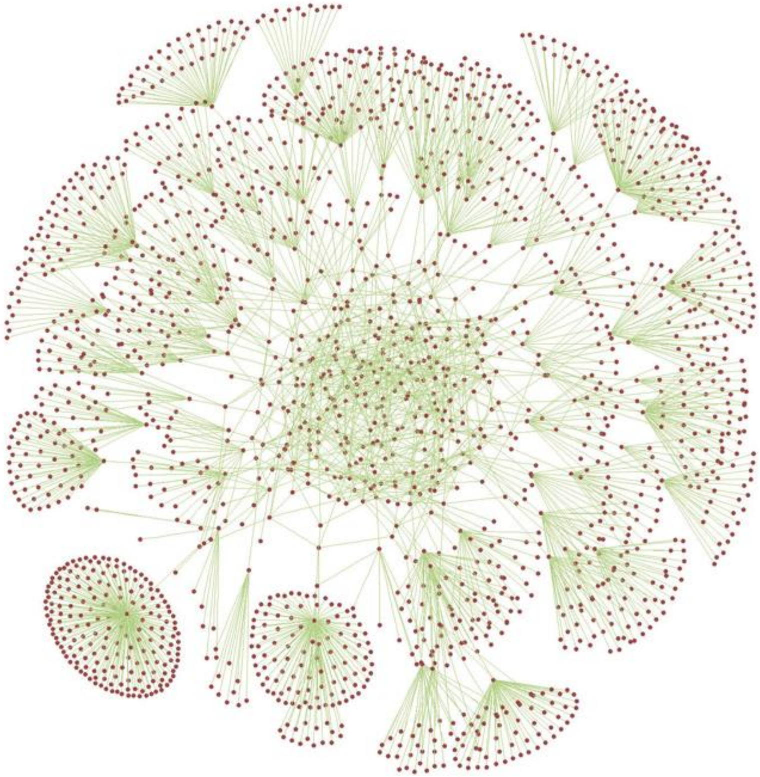
Early vision: Multiple images

School of Electronic &
Electrical Engineering

Sungkyunkwan University

Hyunjin Park

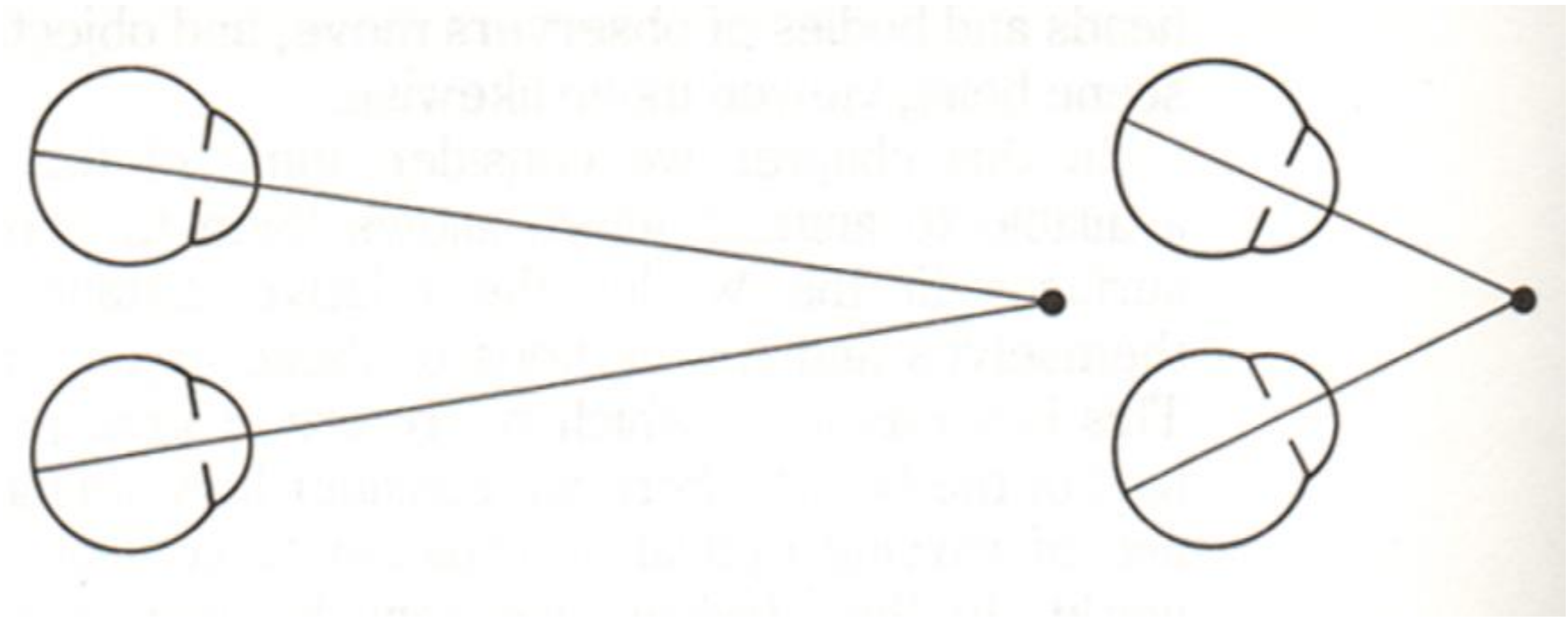




Stereopsis

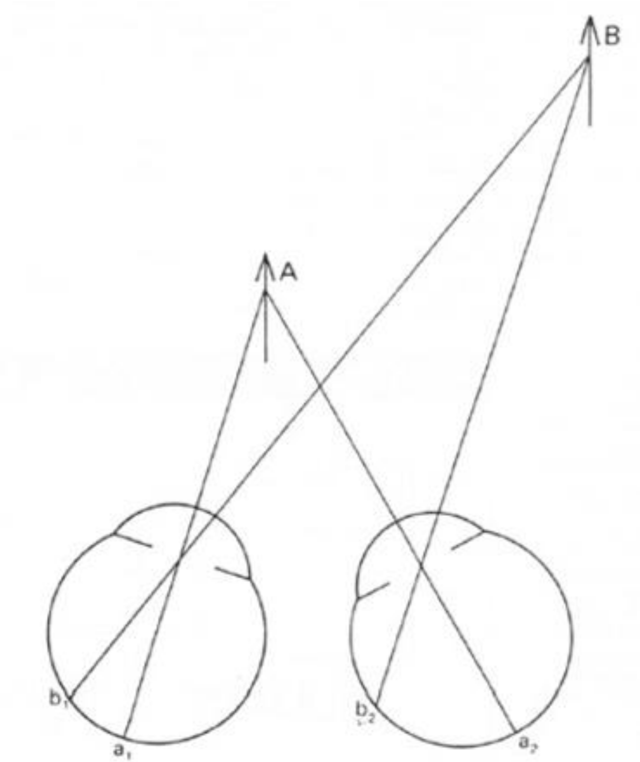
Stereopsis

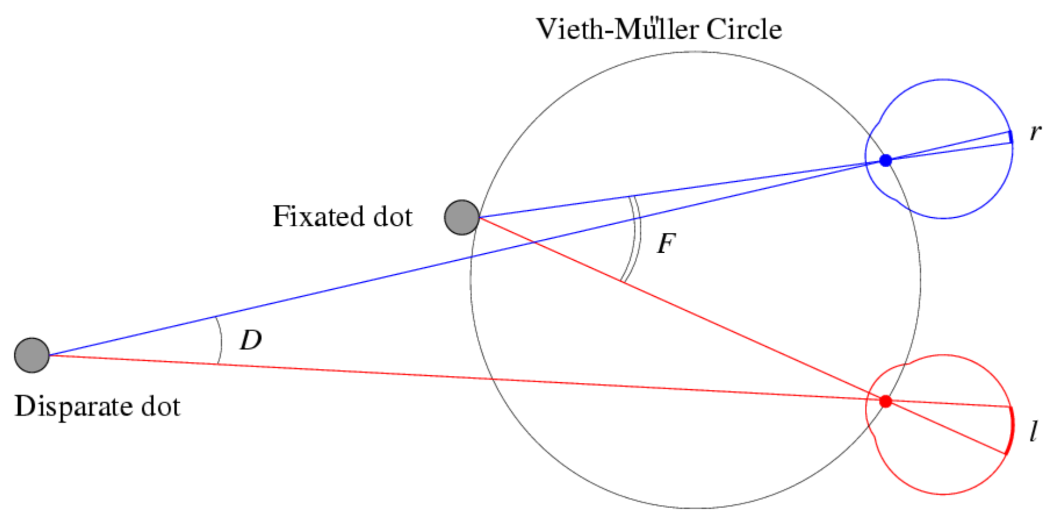
- Concept
  - Fusing the picture recorded by out two eyes
    and exploiting the difference (or disparity) between them
    allows us to gain a strong sense of depth



Stereopsis

- Disparity
  - Occurs when eyes fixate on one object, others appear at different visual angle

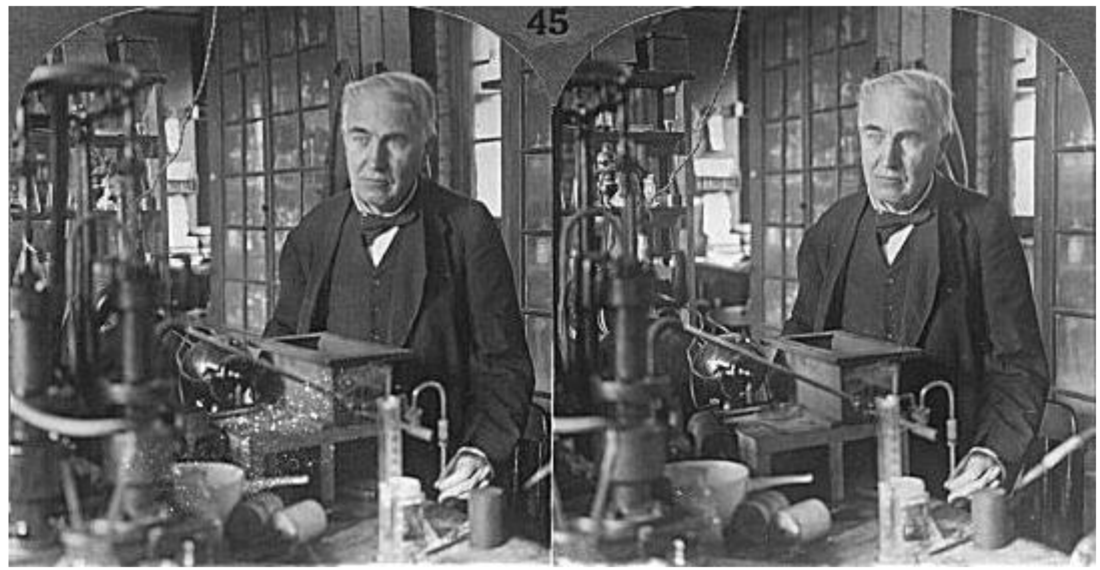




Disparity: $d = r - l = D - F$

Stereopsis

- Two pictures of the same subject from two different viewpoints

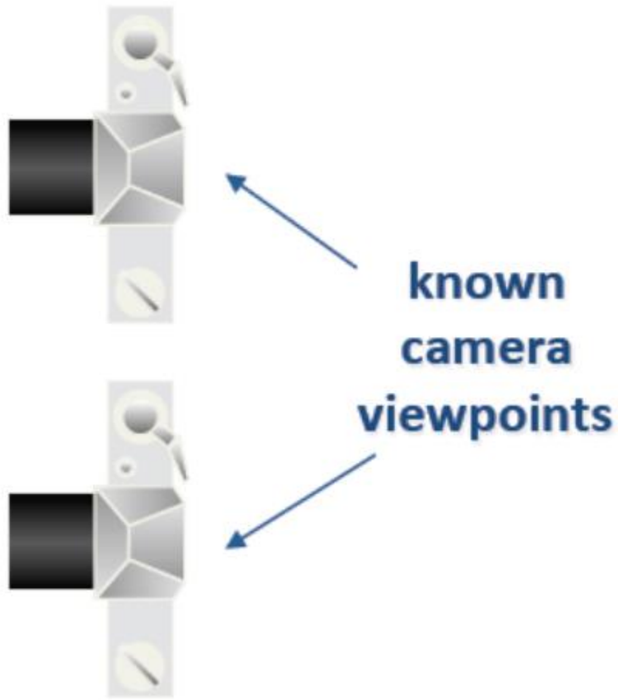


Stereopsis

- Two pictures of the same subject from two different viewpoints

Stereopsis

- The stereopsis problem
  - Fusion and reconstruction



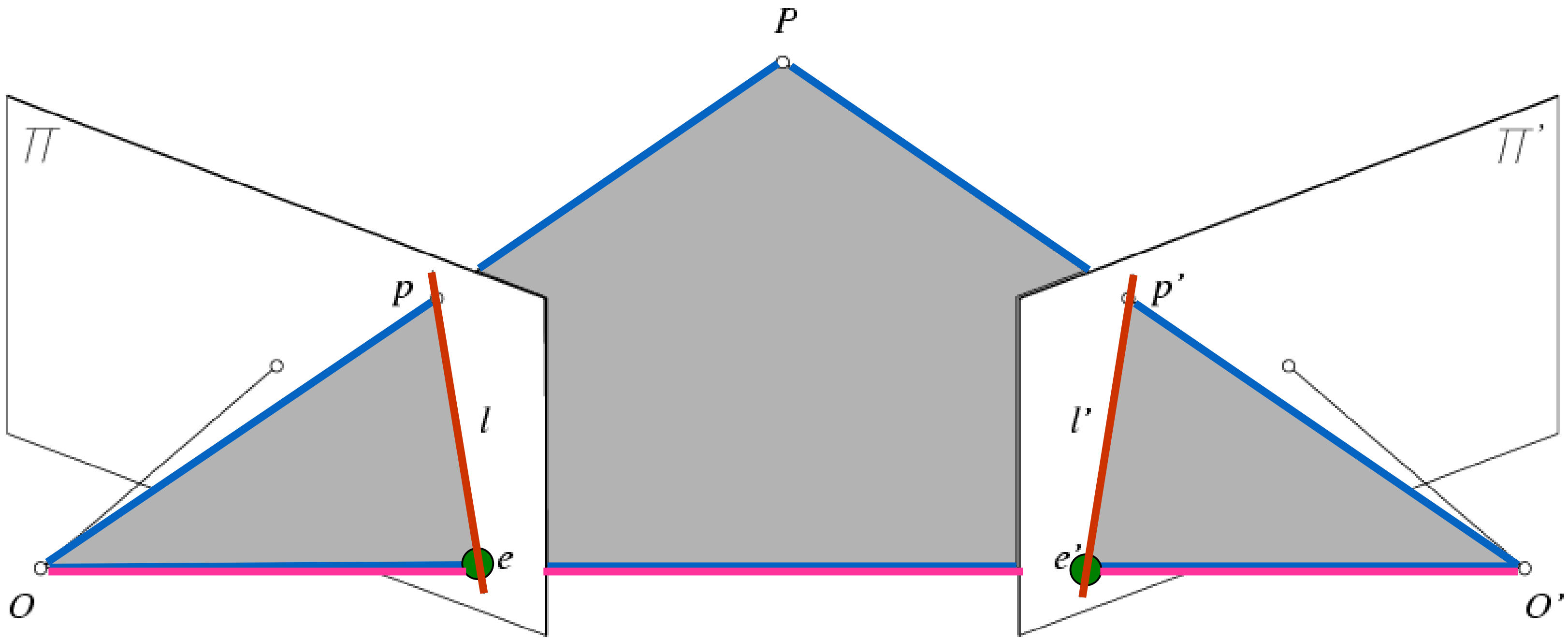
Stereopsis

- Binocular fusion
  - Correlation-based fusion
  - Multi-scale edge matching
  - Dynamic programming

Left Right

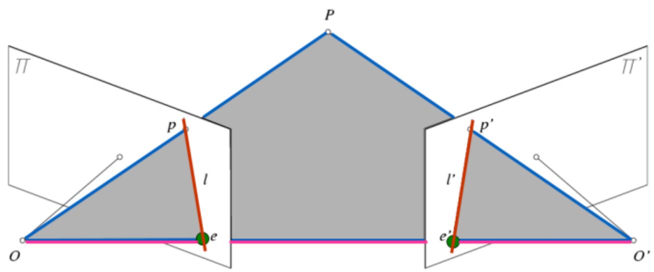
Matching correlation windows across scan lines

Epipolar geometry



- Epipolar Plane
- Epipoles
- Epipolar Lines
- Baseline

Epipolar geometry

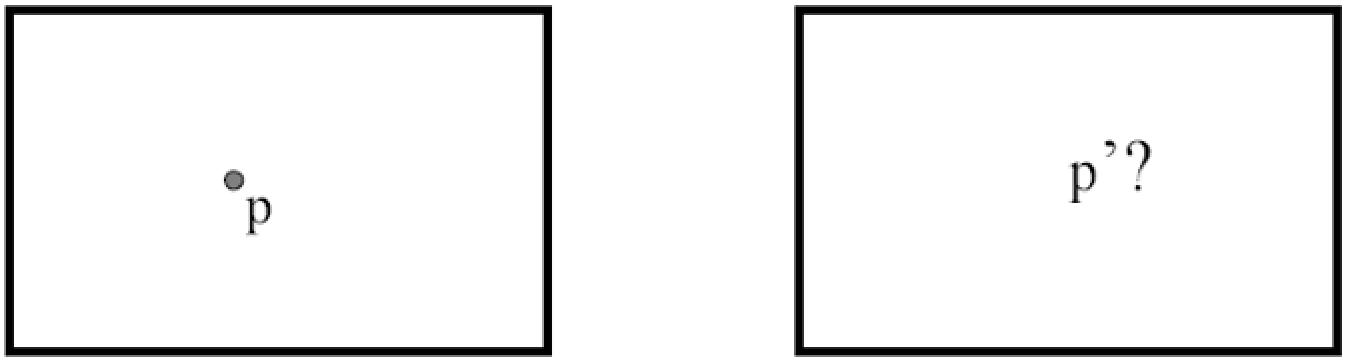


- Baseline
  - Line joining the camera centers
- Epipole
  - Point of intersection of baseline with the image plane
- Epipolar plane
  - Plane containing baseline and world point
- Epipolar line
  - Intersection of epipolar plane with the image plane

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

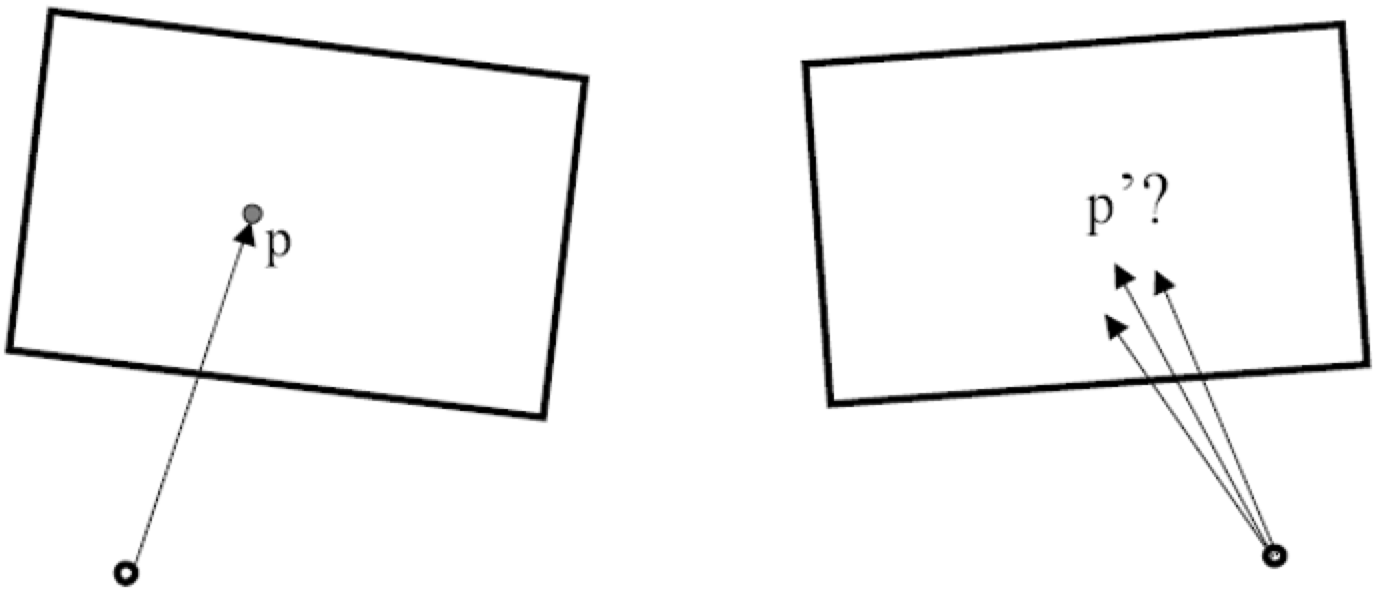
Stereo correspondence constraints

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



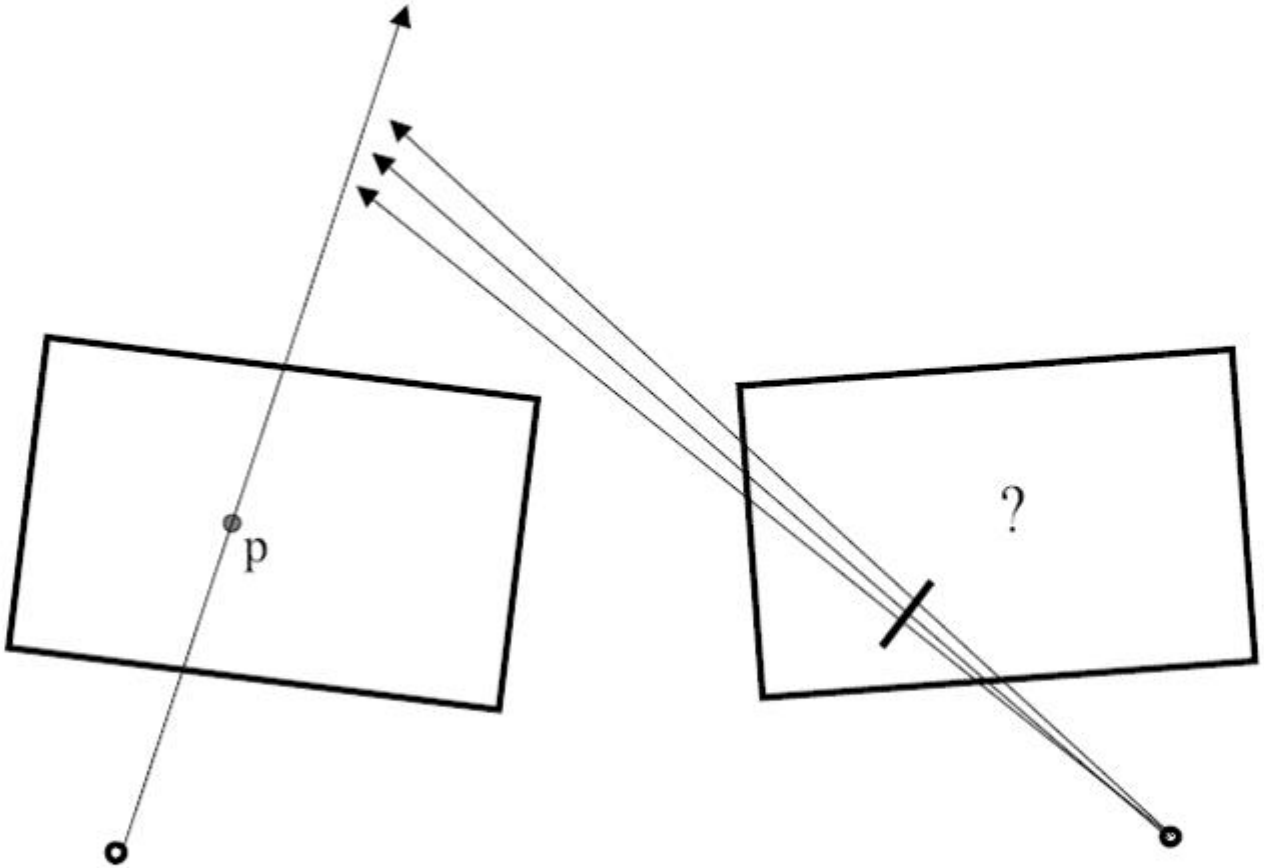
Stereo correspondence constraints

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



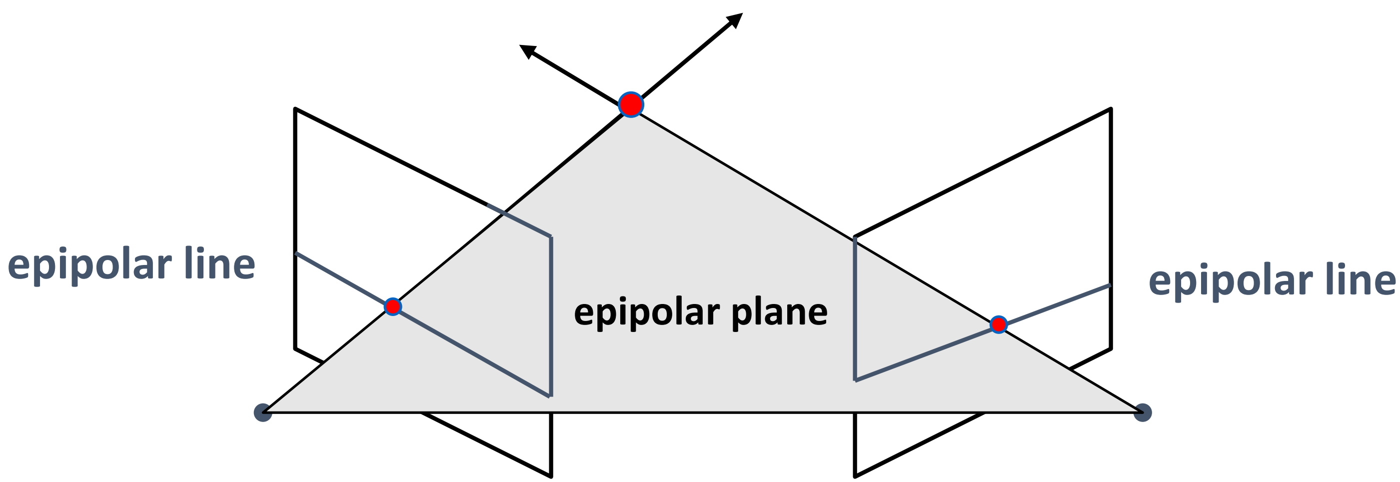
Stereo correspondence constraints

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



Stereo correspondence constraints

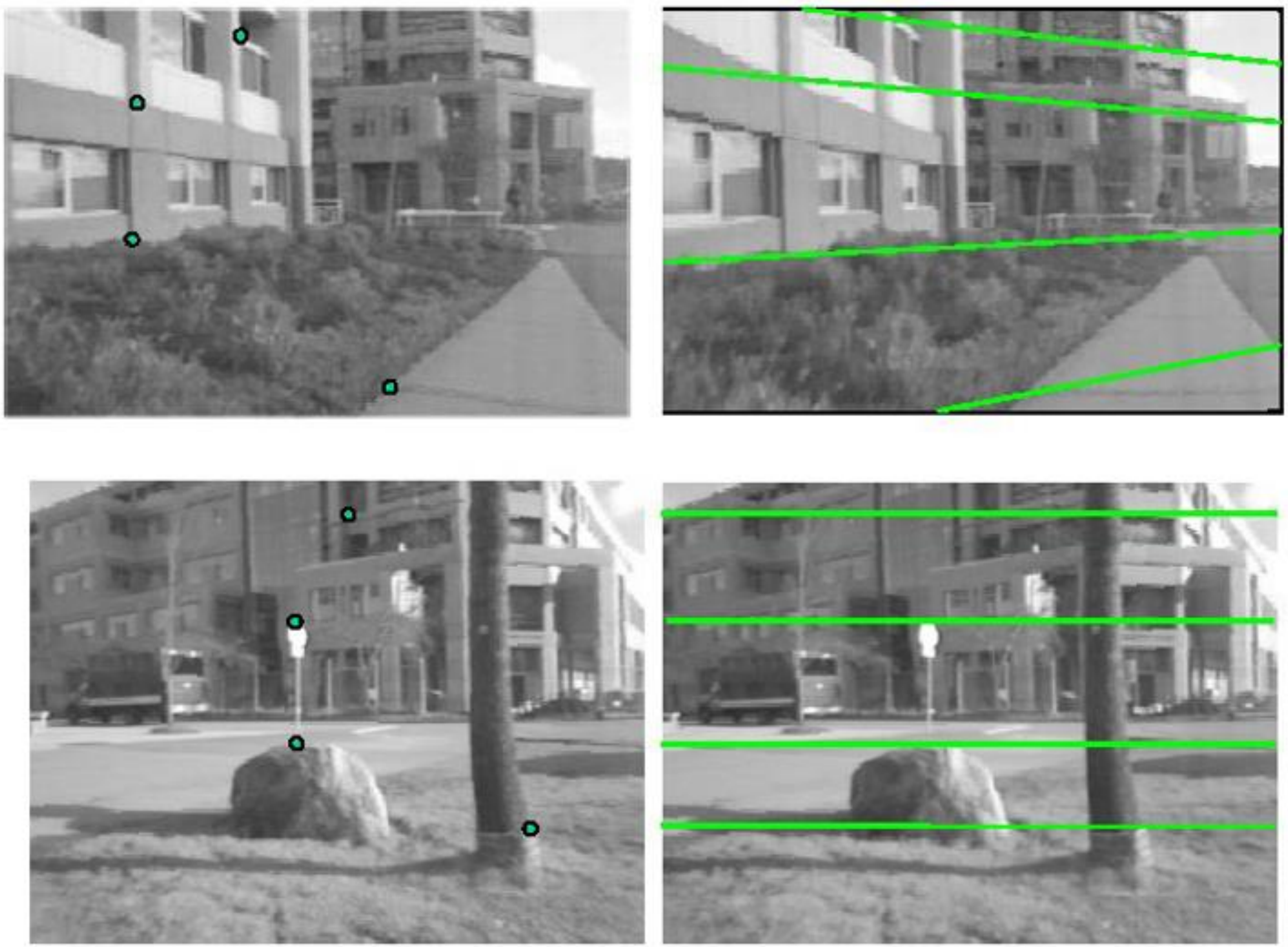
- Geometry of two views allows us to constrain where the corresponding pixel for some image point in the first view must occur in the second view



- Epipolar constraint
  - Reduces correspondence problem to 1D searching along conjugate epipolar lines

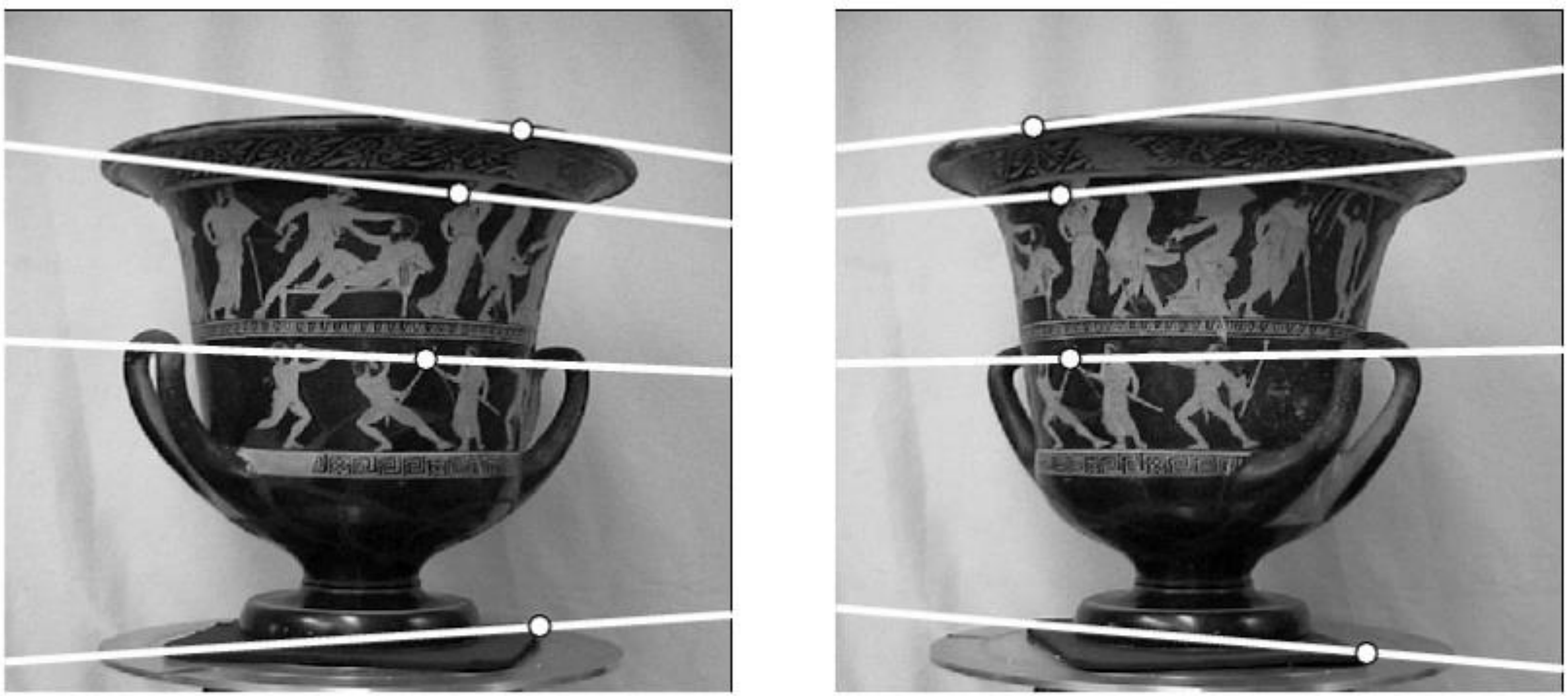
Epipolar constraint

- Example



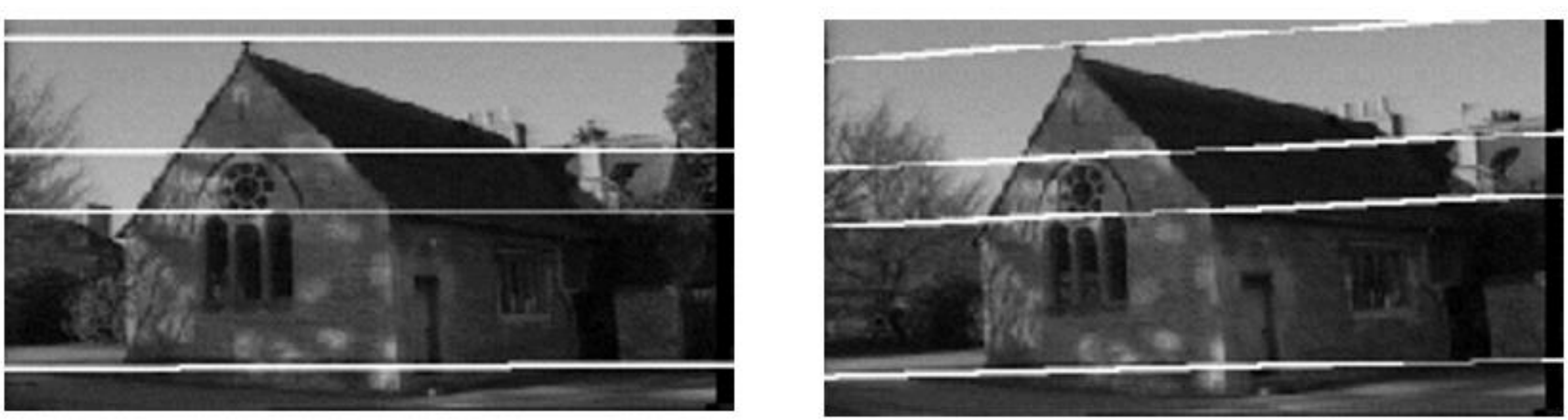
Epipolar constraint

- Example: Converging cameras



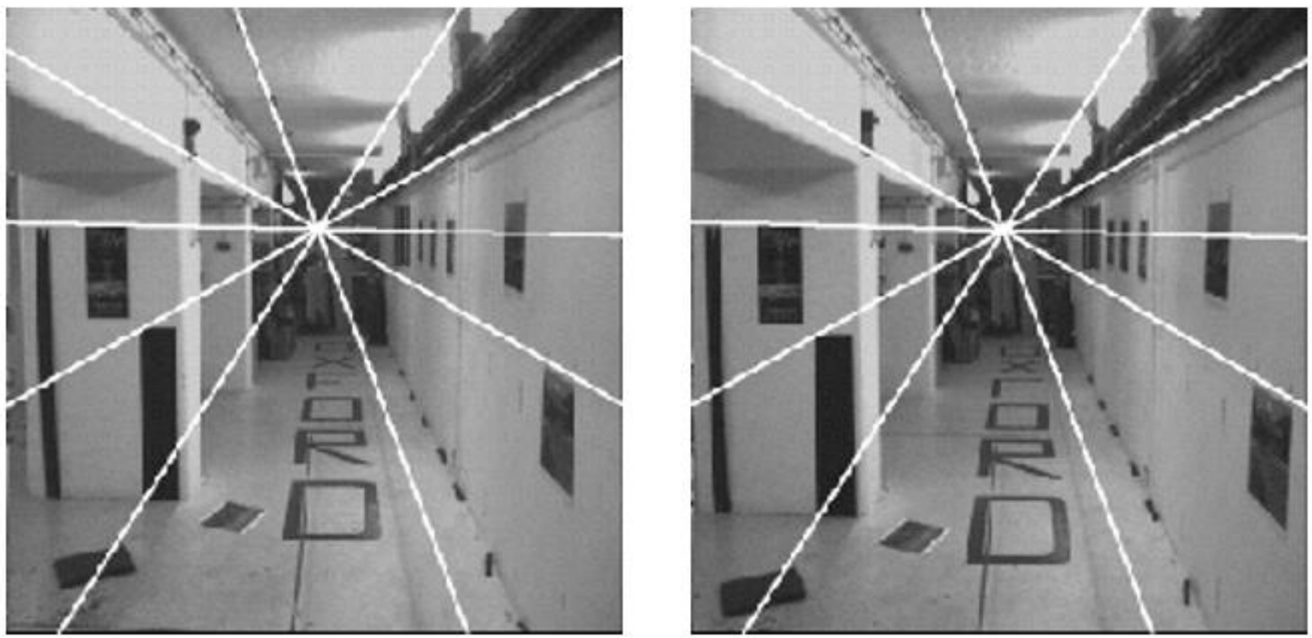
Epipolar constraint

- Example: Motion parallel with image plane



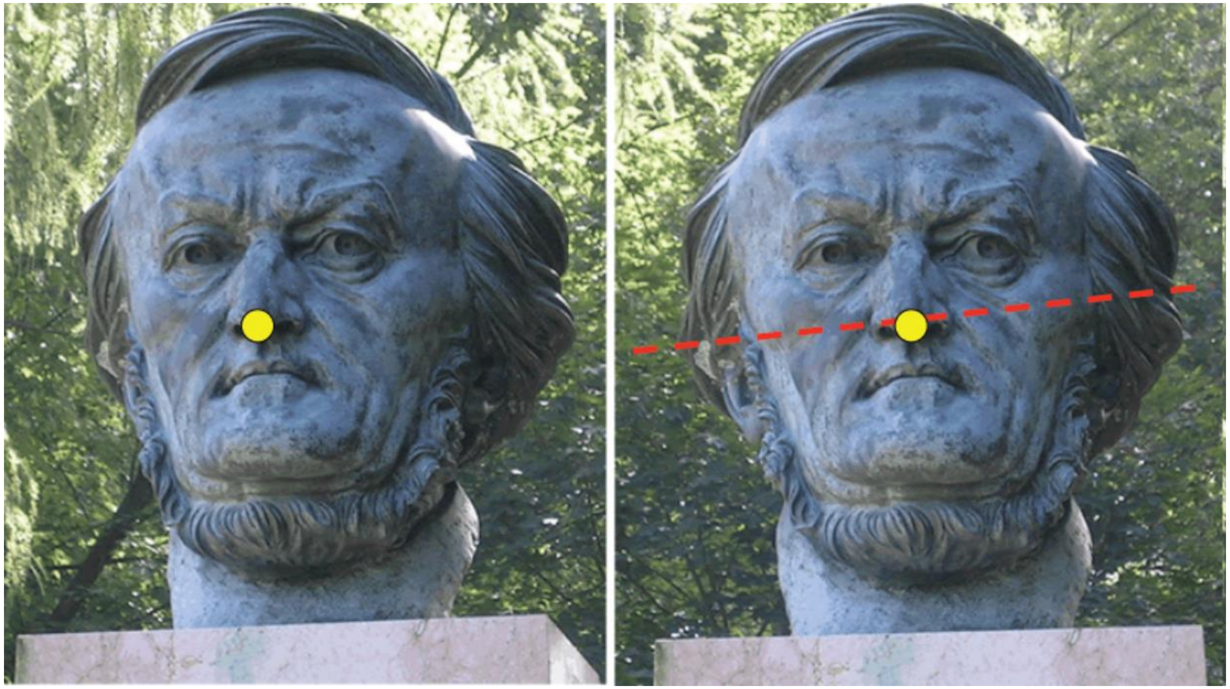
Epipolar constraint

- Example: Forward motion



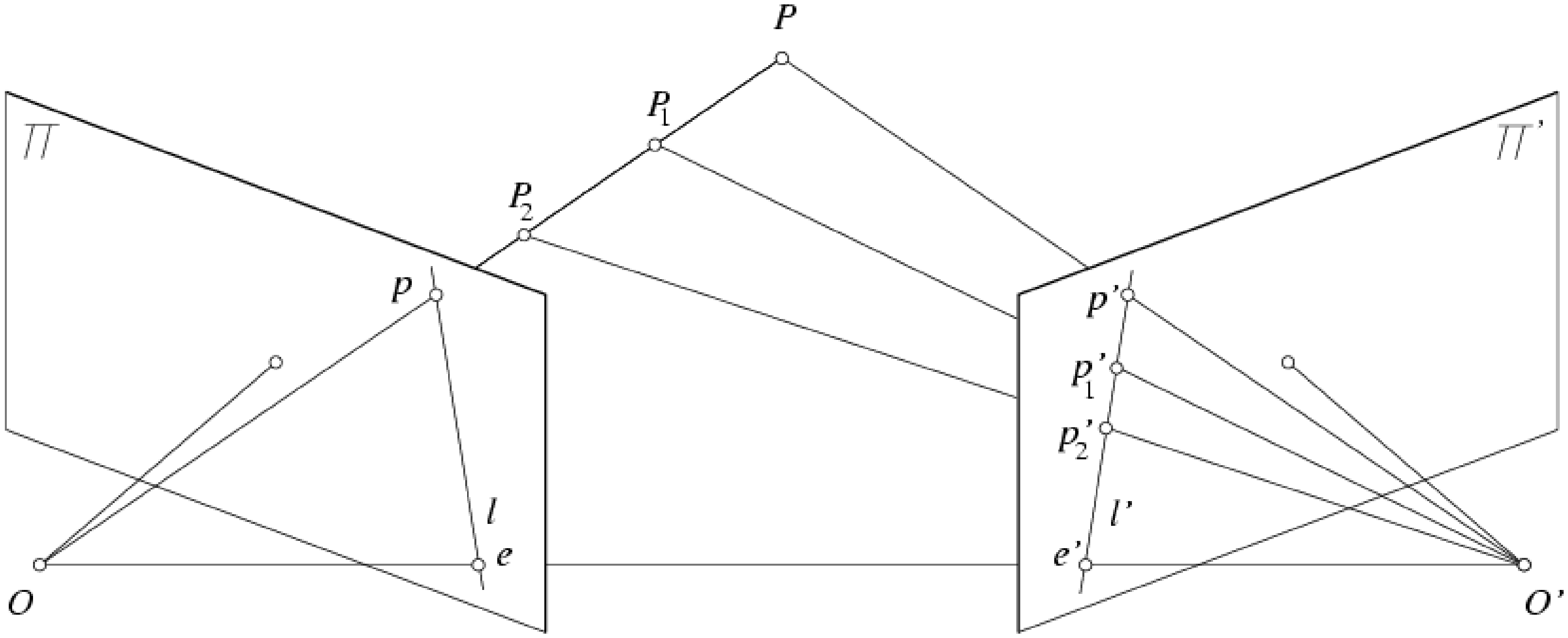
Epipolar constraint

- Two views of the same object
- Suppose I know the camera position and camera matrices
- Given a point on the left image, how can I find the corresponding point on the right image?



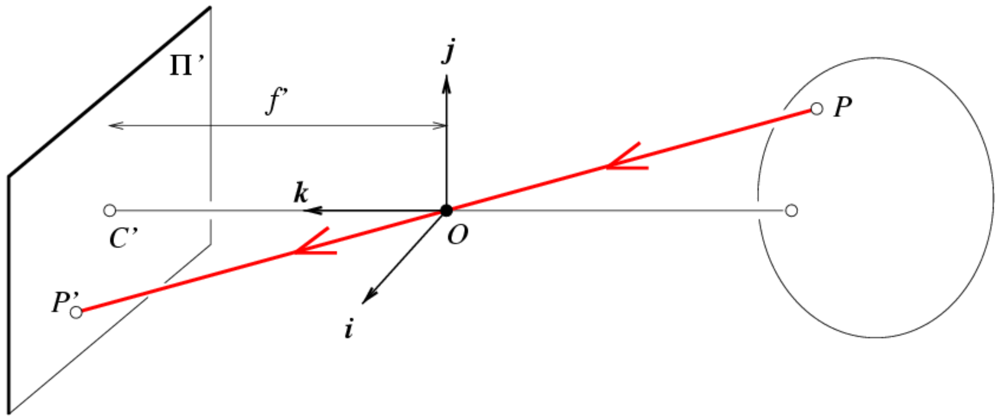
Epipolar constraint

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



Epipolar constraint

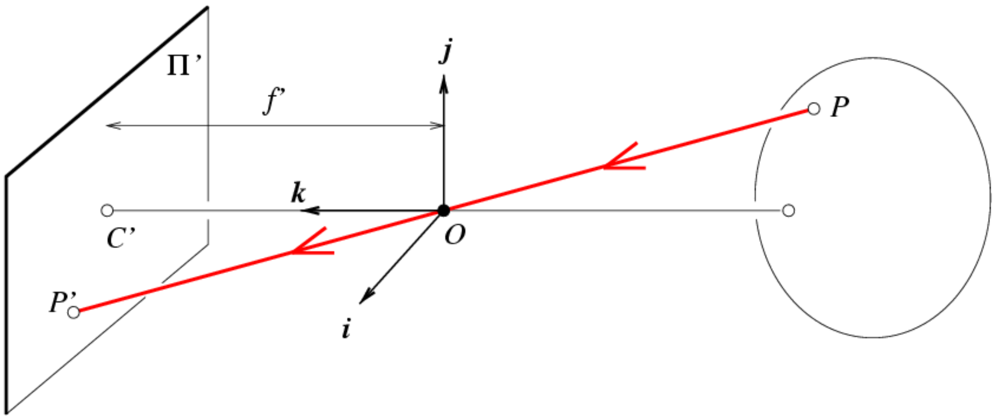
- Real-world point to a point on the camera



$$P' = \begin{bmatrix} fx \\ fy \\ z \end{bmatrix} = \underbrace{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{M} \longleftarrow \textbf{Ideal world}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Epipolar constraint

- Real-world point to a point on the camera



$$P' = \begin{bmatrix} fx \\ fy \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = K[I \quad 0]P$$

K

Epipolar constraint

Real-world camera

$$P' = MP = K[I \quad 0]P$$

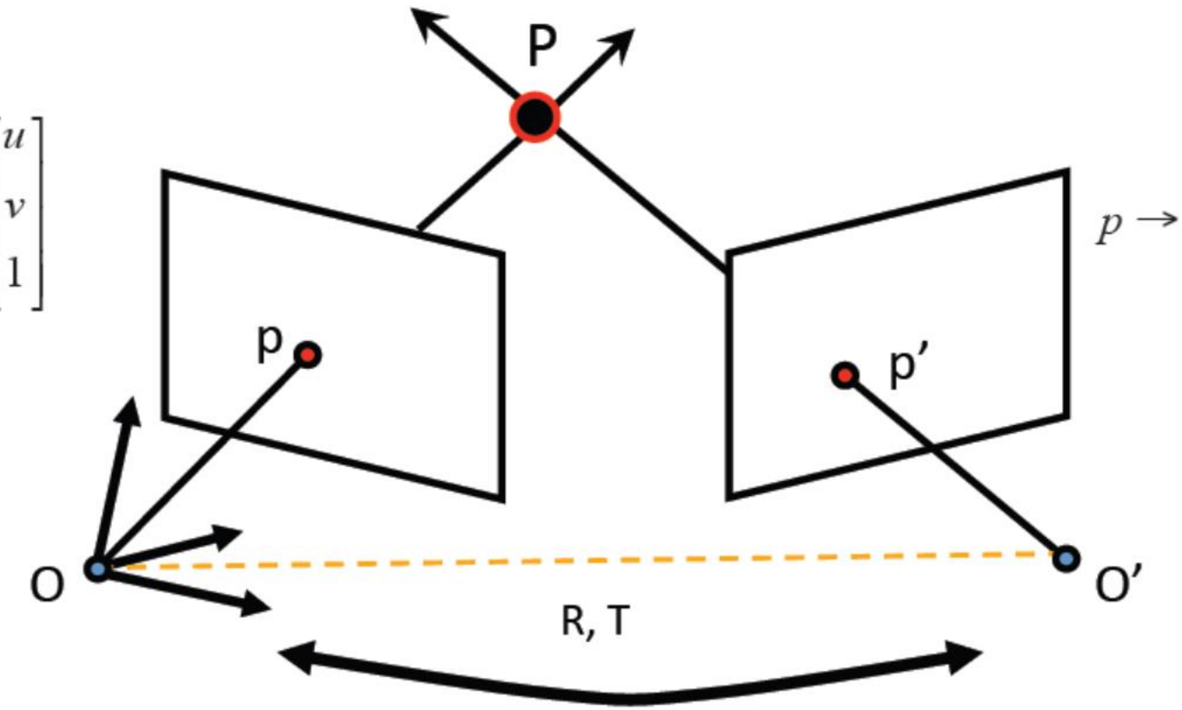
$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & s & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Real-world camera
+
Real-world transformation

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

$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & s & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Epipolar constraint



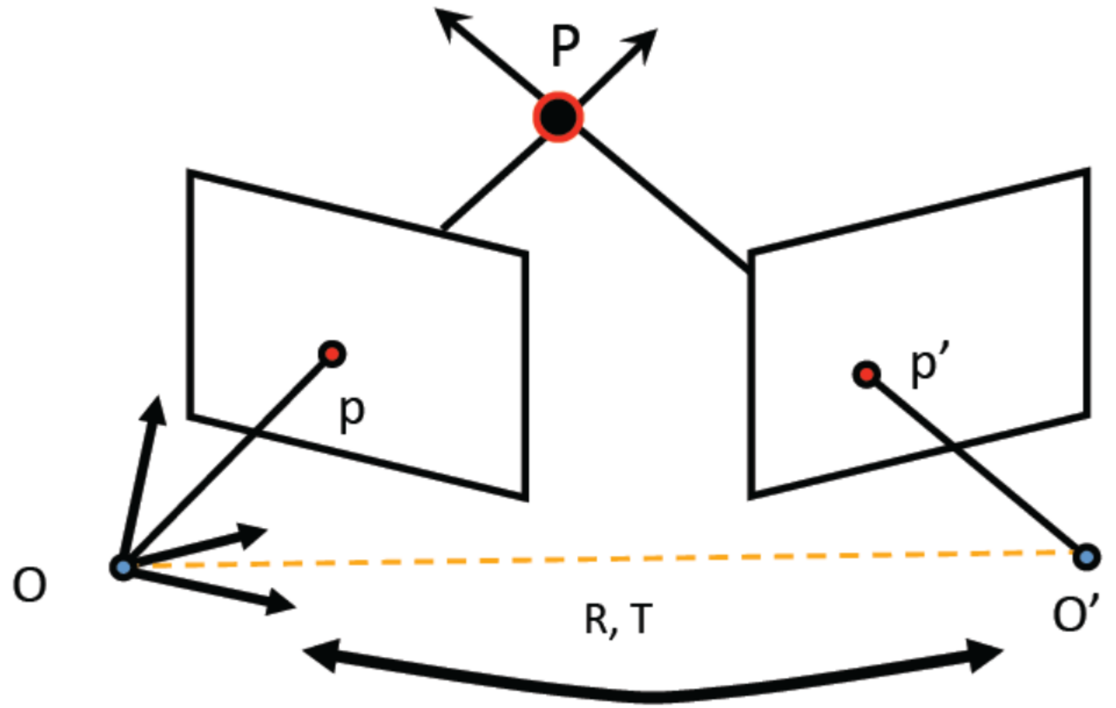
$$p \rightarrow M\,P = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

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

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

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

Epipolar constraint

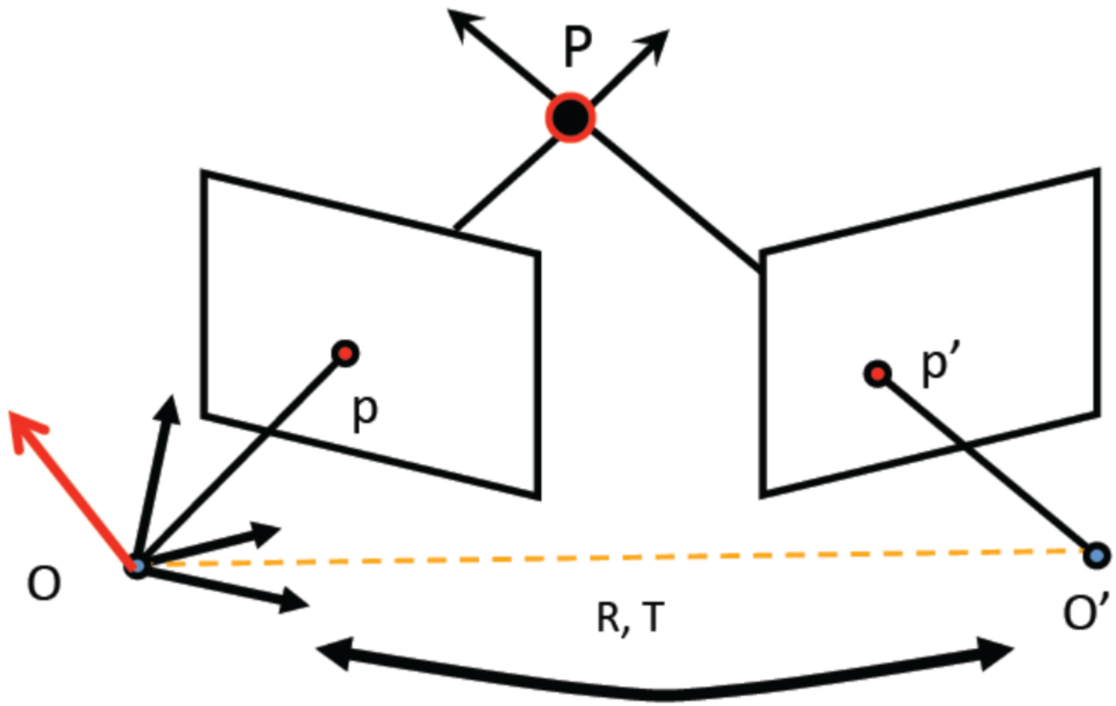


$$M = K\begin{bmatrix} I & 0 \end{bmatrix} \quad\downarrow\quad M = \begin{bmatrix} I & 0 \end{bmatrix}$$

K_1 and K_2 are known
(calibrated cameras)

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

Epipolar constraint



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

Perpendicular to epipolar plane

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

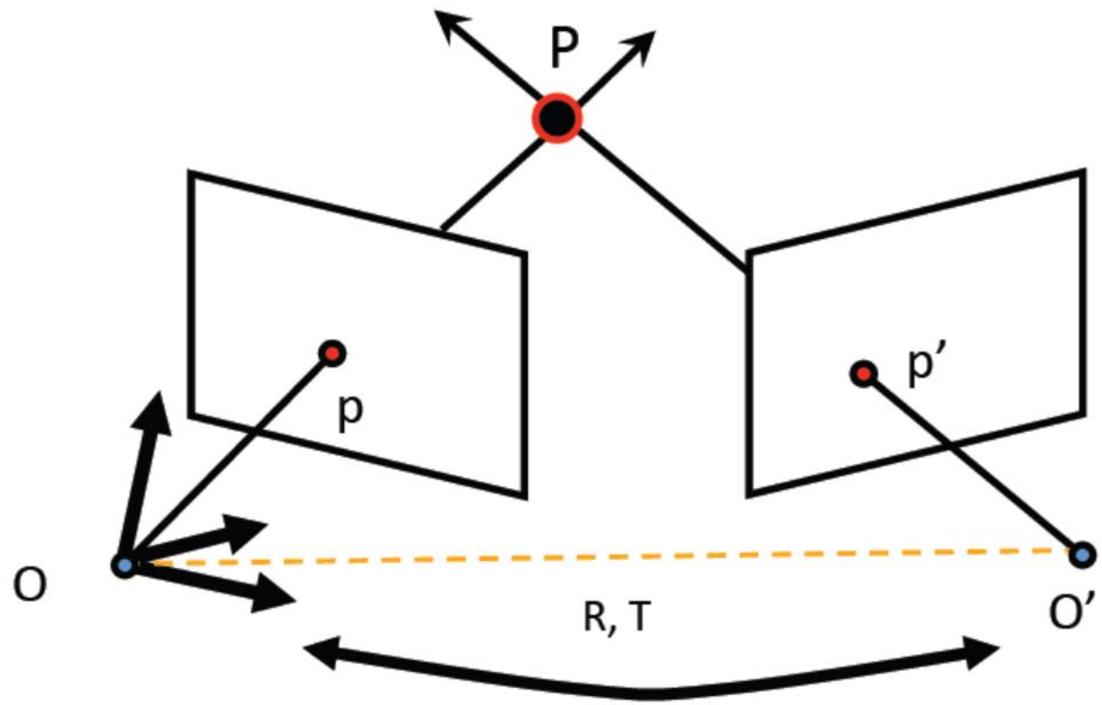
Epipolar constraint

- Cross product as matrix multiplication

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

skew symmetric matrix

Epipolar constraint

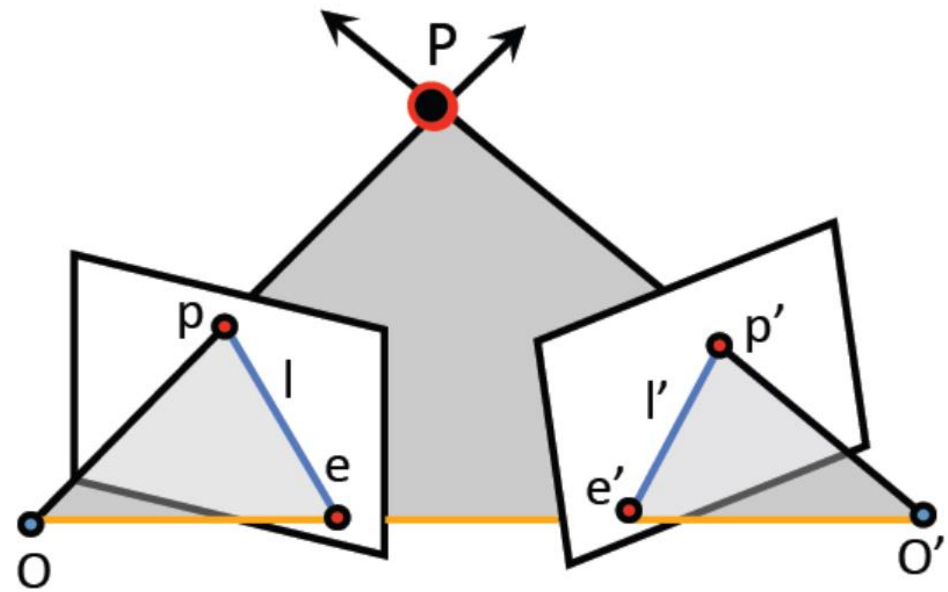


$$p^T \cdot [T \times (R\, p')] = 0 \rightarrow p^T \cdot \boxed{[T_\times] \cdot R}\, p' = 0$$

(Longuet-Higgins, 1981) E = essential matrix

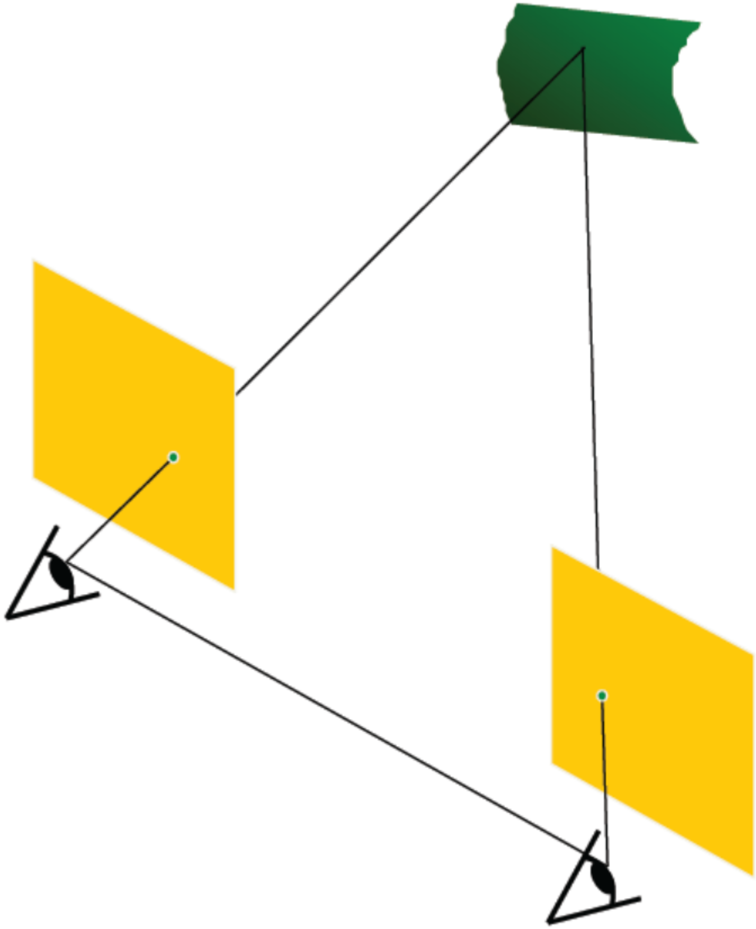
Epipolar constraint

- Ep': Epipolar line associated with p' ($l = Ep'$)
- $E^T p$: Epipolar line associated with p ($l' = E^T p$)
- E is singular (rank two)
- $Ee' = 0$ and $E^T e = 0$
- E is 3×3 matrix, 5 DOF



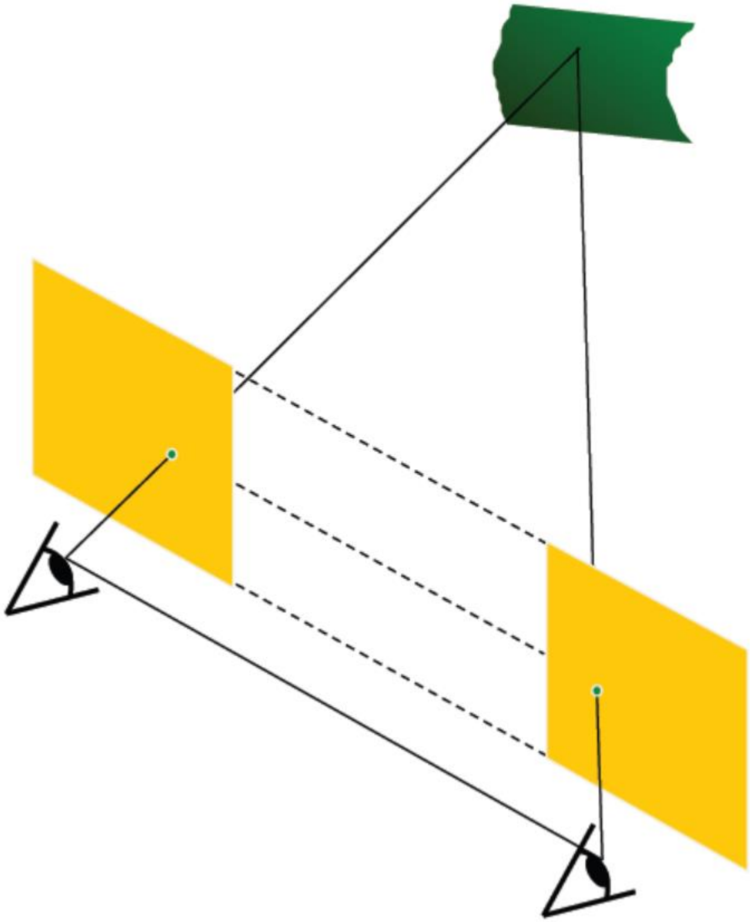
Simple case: Parallel images

- Image planes of cameras are parallel to each other and to the baseline
- Camera centers are at same height
- Focal lengths are the same



Simple case: Parallel images

- Image planes of cameras are parallel to each other and to the baseline
- Camera centers are at same height
- Focal lengths are the same
- Then, epipolar lines fall along the horizontal scan lines of the image



Essential matrix for parallel images

- Epipolar constraint:

$$R = I \qquad t = (T, 0, 0)$$

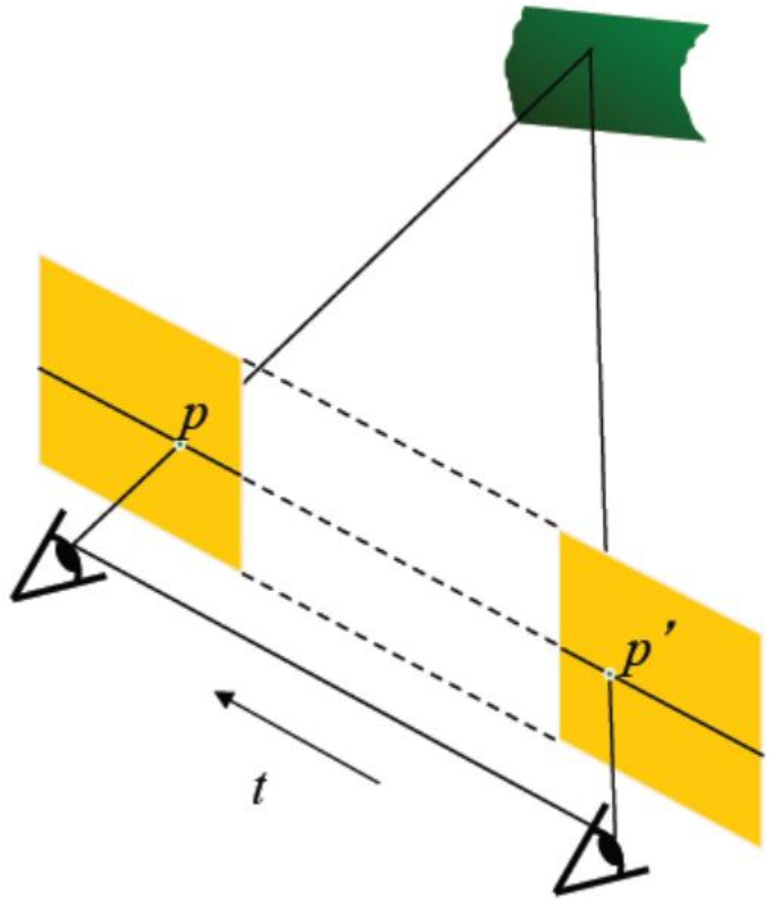
$$p^T E p' = 0, \qquad E = [t_\times]R$$

$$E = [t_\times]R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix}$$

$$\begin{pmatrix} u & v & 1 \end{pmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0$$

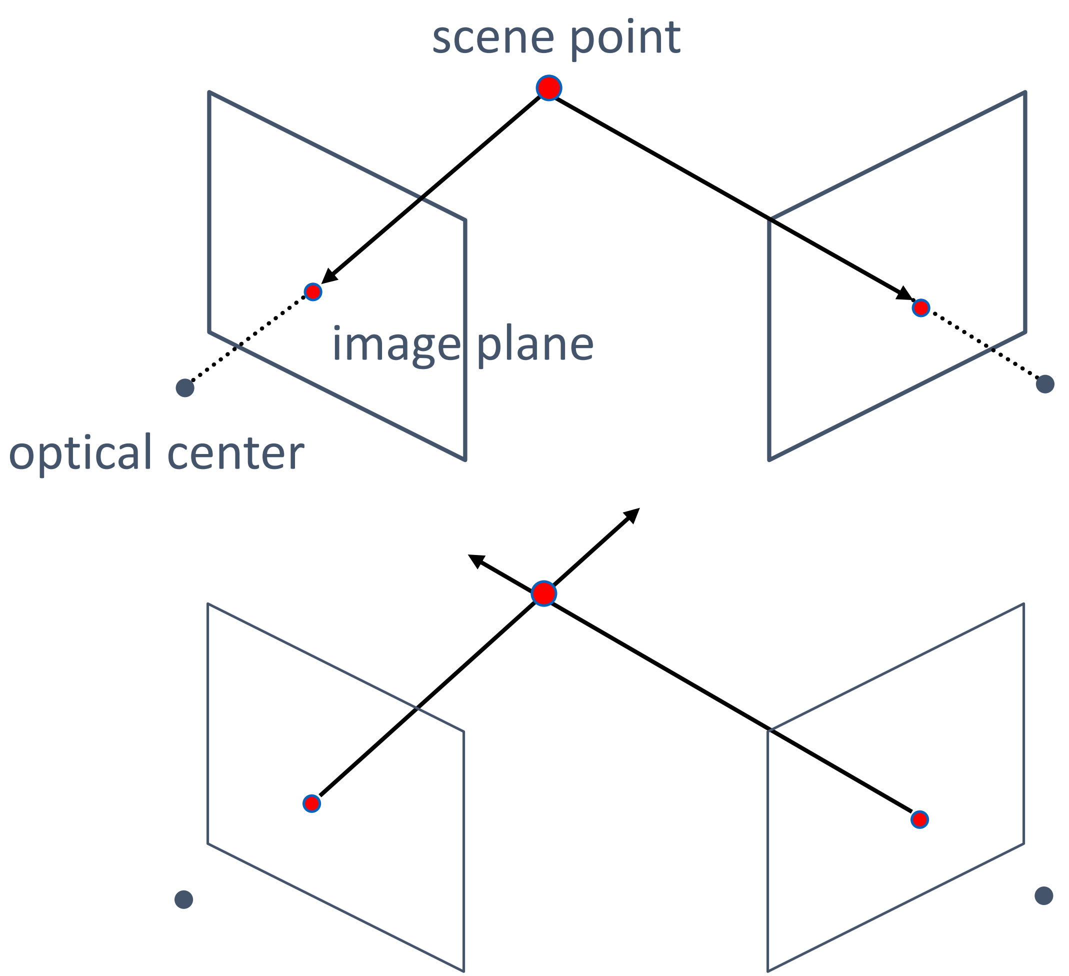
$$\begin{pmatrix} u & v & 1 \end{pmatrix} \begin{pmatrix} 0 \\ -T \\ Tv' \end{pmatrix} = 0$$

$$Tv = Tv'$$



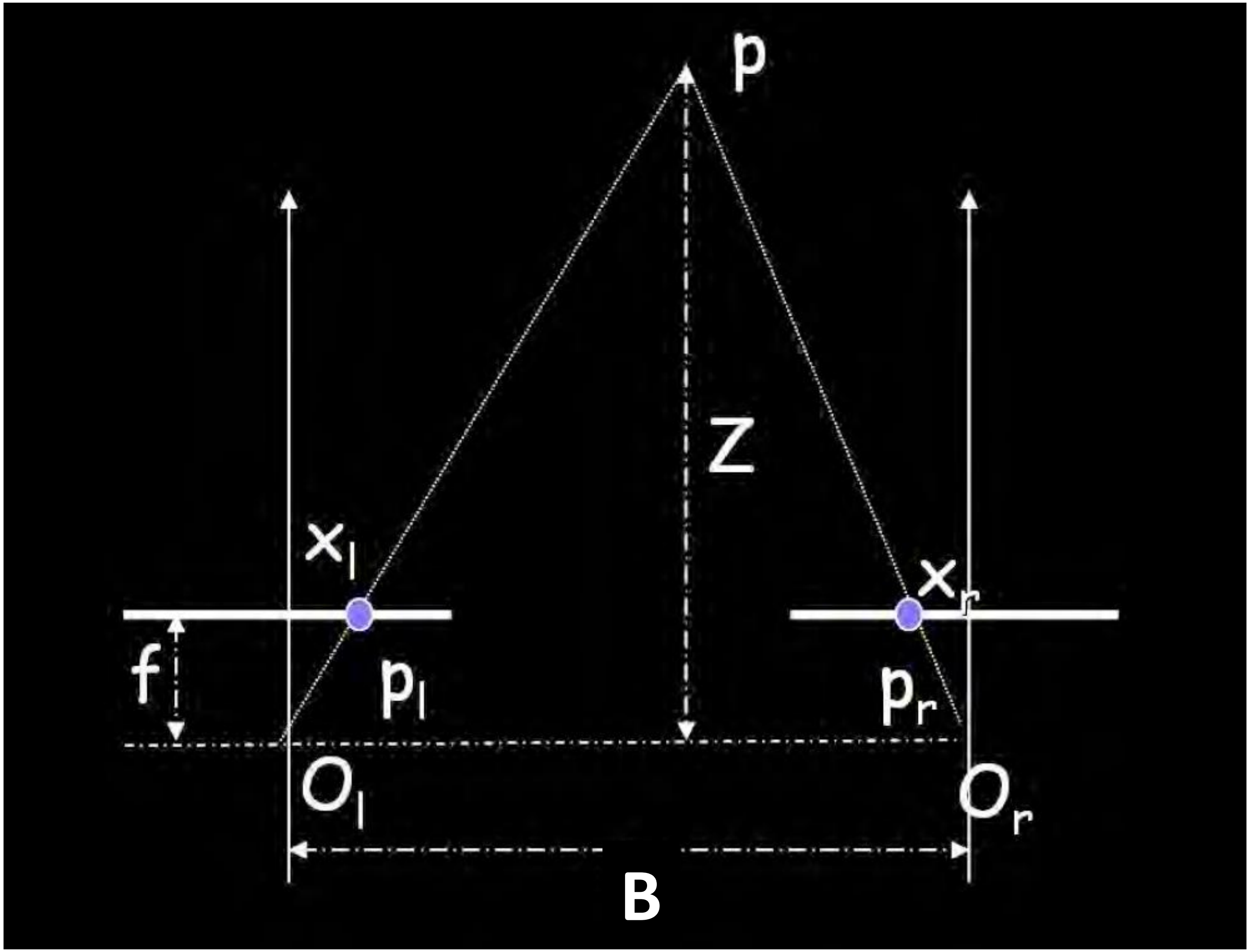
Triangulation

- Basic principle: Triangulation



Triangulation

- Assume parallel optical axes



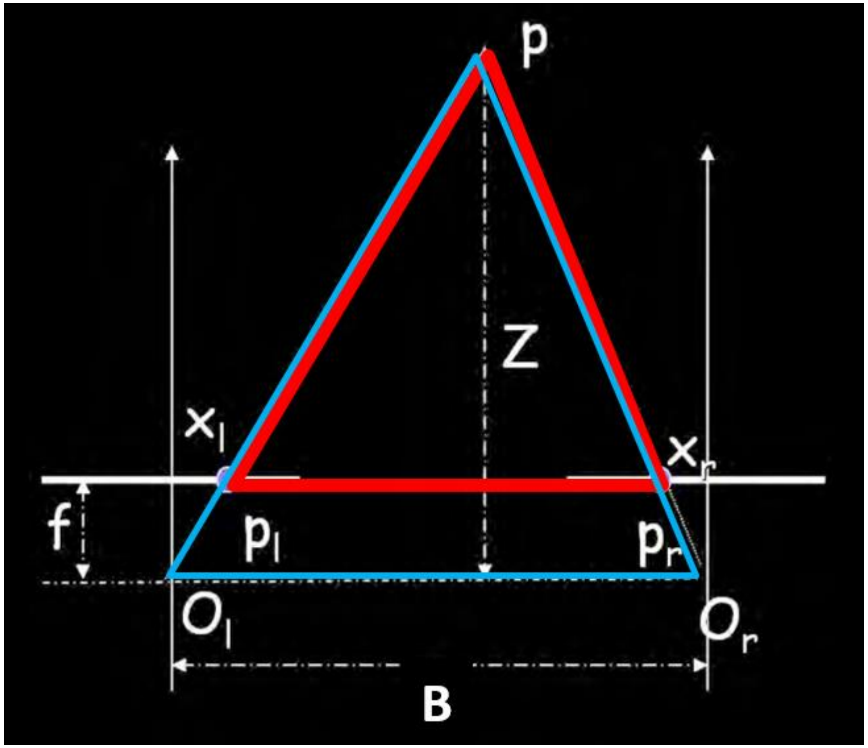
Triangulation

- We can triangulate via:
  - Similar triangles (p_l, P, p_r) and (O_l, P, O_r)

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

$$Z = f \frac{B}{\boxed{x_r - x_l}}$$

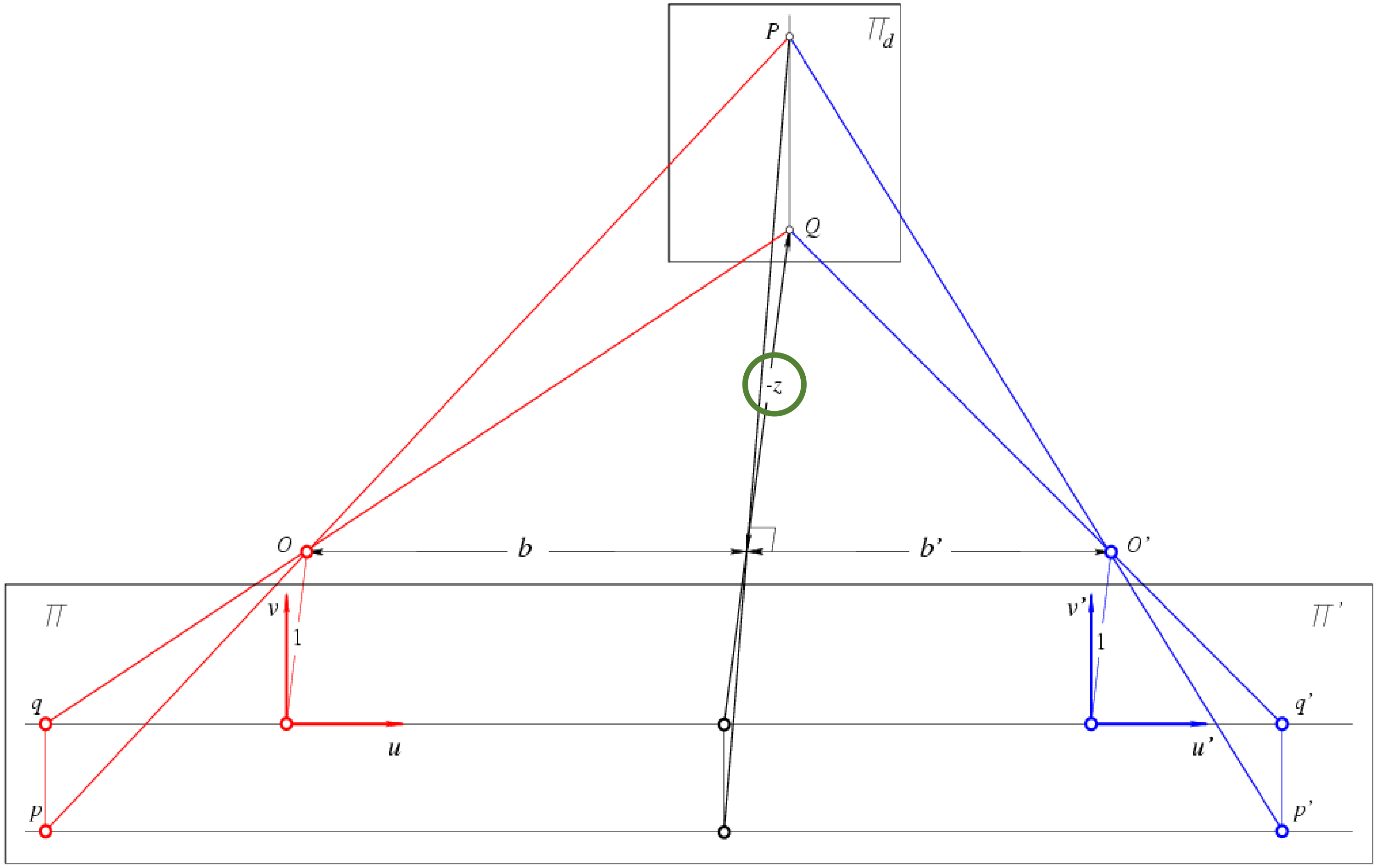
disparity



- Disparity is inversely proportional to depth!

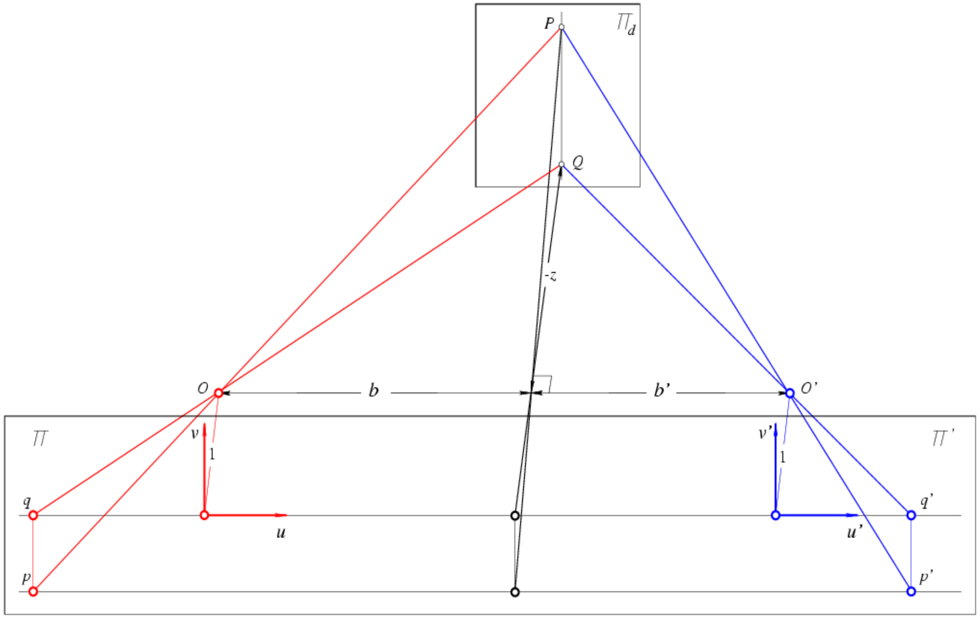
Stereo image reconstruction

- Derive expression for z (depth) as a function of u, u', f, B



Stereo image reconstruction

- Depth of P in the coordinate system attached to the first camera



$$u = f\frac{x_1}{z}$$

$$u' = f\frac{x_2}{z} = f\frac{x_1 - B}{z} = u - f\frac{B}{z}$$

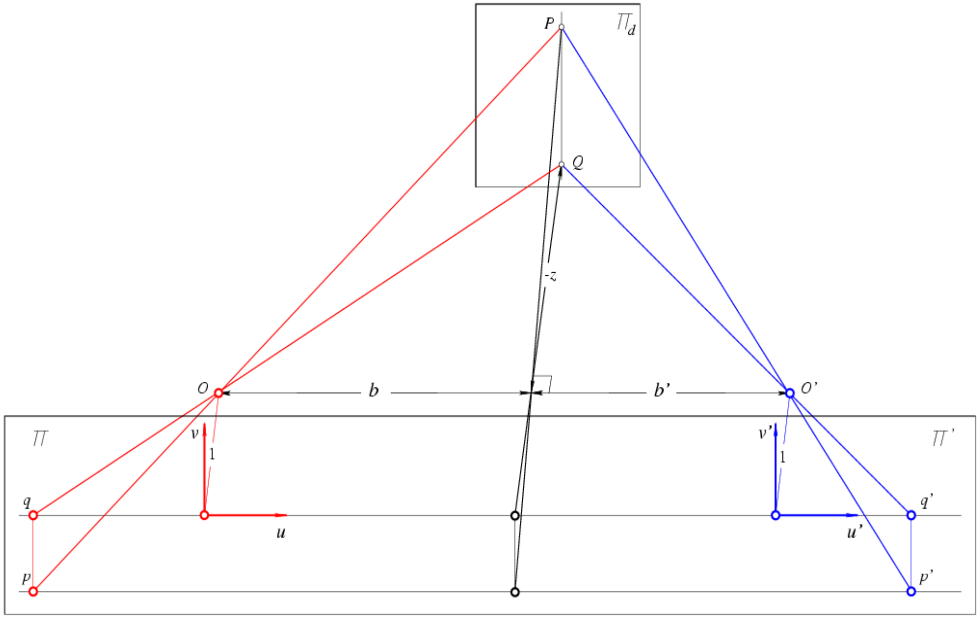
$$\Rightarrow z = -\frac{fB}{u' - u}$$

Disparity: $d = u' - u$, and $f = 1$

→ Depth: $z = -\frac{B}{d}$

Stereo image reconstruction

- Depth of P in the coordinate system attached to the first camera



$$u = f\frac{x}{z}, v = f\frac{y}{z}$$

since $z = -\frac{B}{d}$ and $f = 1$

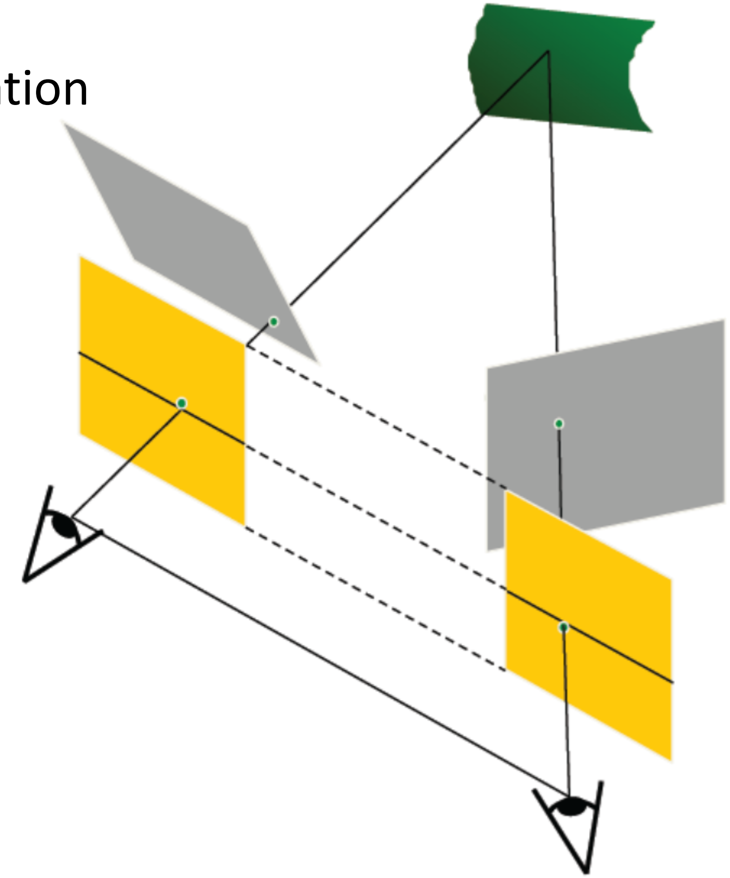
$$x = -\frac{B}{d}u, y = -\frac{B}{d}v$$

i.e., $\boldsymbol{P} = -\left(\frac{B}{d}\right)\boldsymbol{p}$

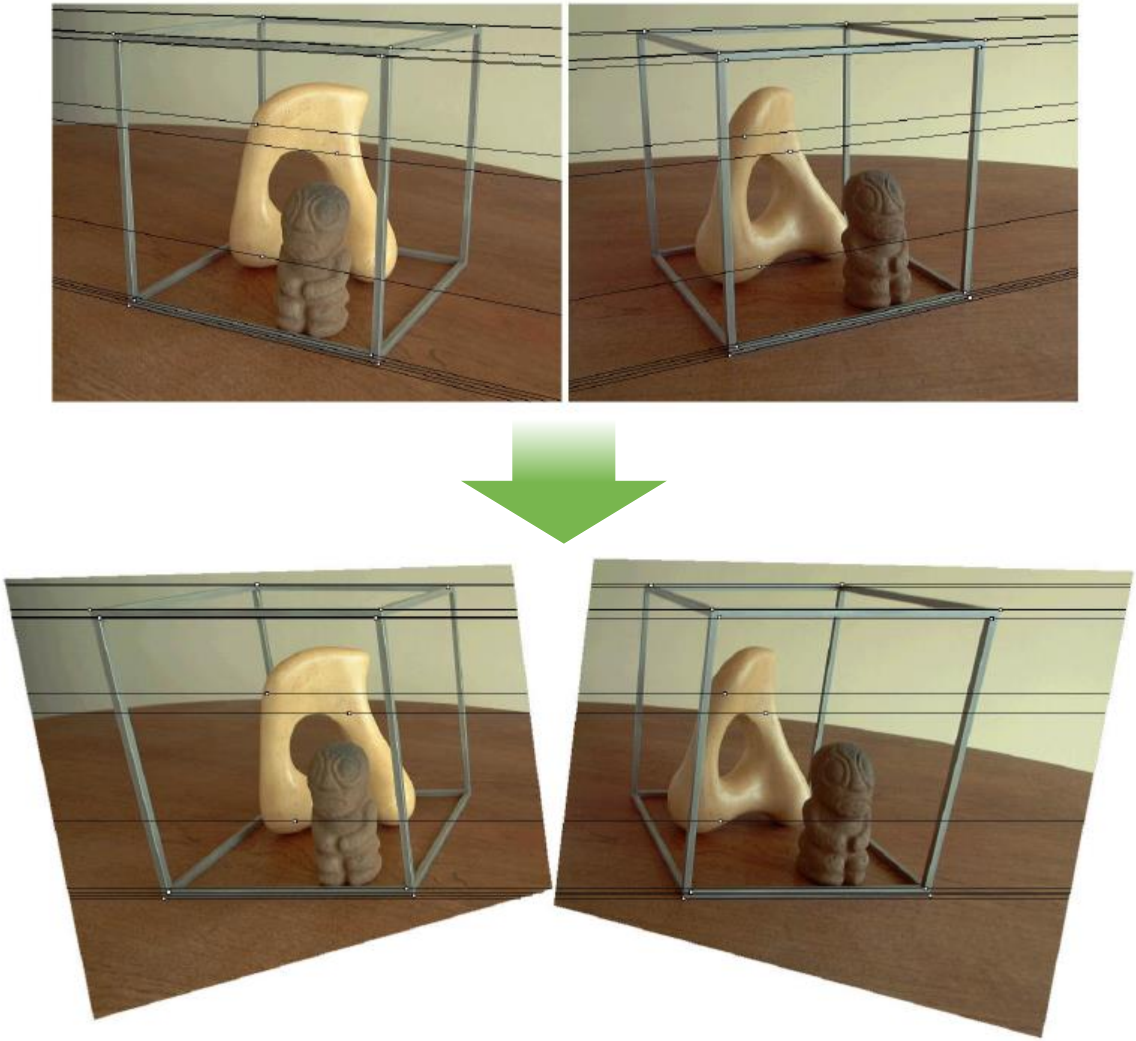
, where $\boldsymbol{P} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ and $\boldsymbol{p} = \begin{pmatrix} \mathrm{u} \\ v \\ 1 \end{pmatrix}$

Stereo image rectification

- Algorithm
  - Re-project image planes onto a common plane parallel to the line between optical centers
  - Pixel motion is horizontal after this transformation
  - Two transformation matrices, one for each input image reprojection

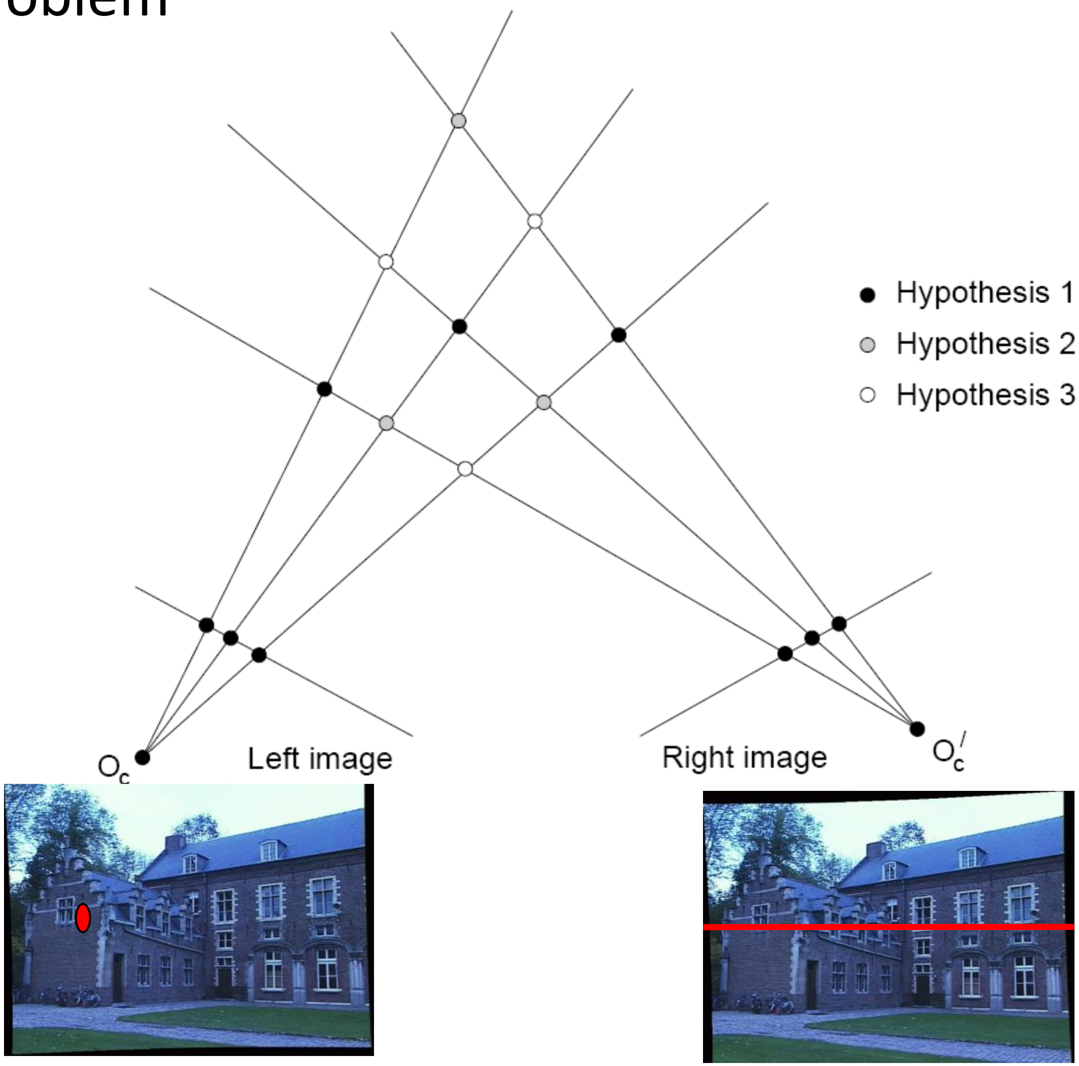


Stereo image rectification



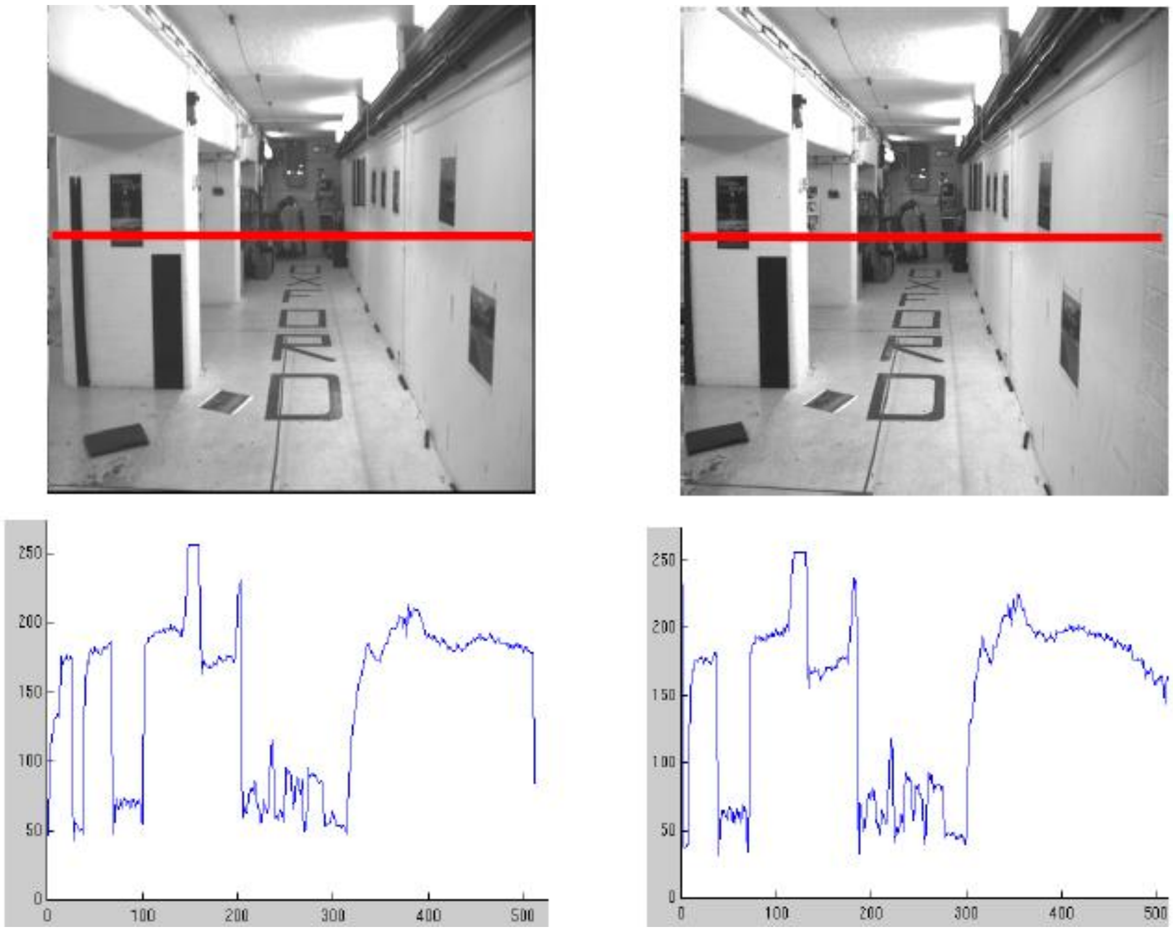
Stereo correspondence

- Stereo correspondence problem
  - Multiple match hypothesis satisfy epipolar constraint
  - Which one is correct?
- Goal
  - Finding matching points between two images



Stereo correspondence

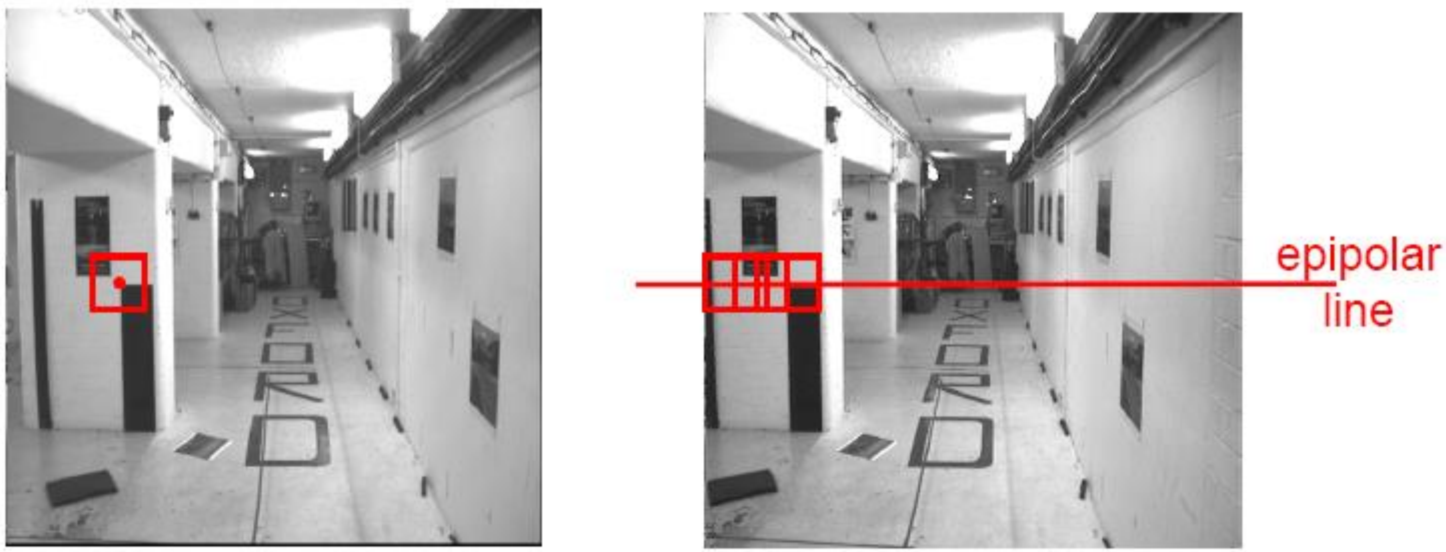
- Parallel camera example



Clear correspondence between intensities, but also noise and ambiguity

Stereo correspondence

- Parallel camera example



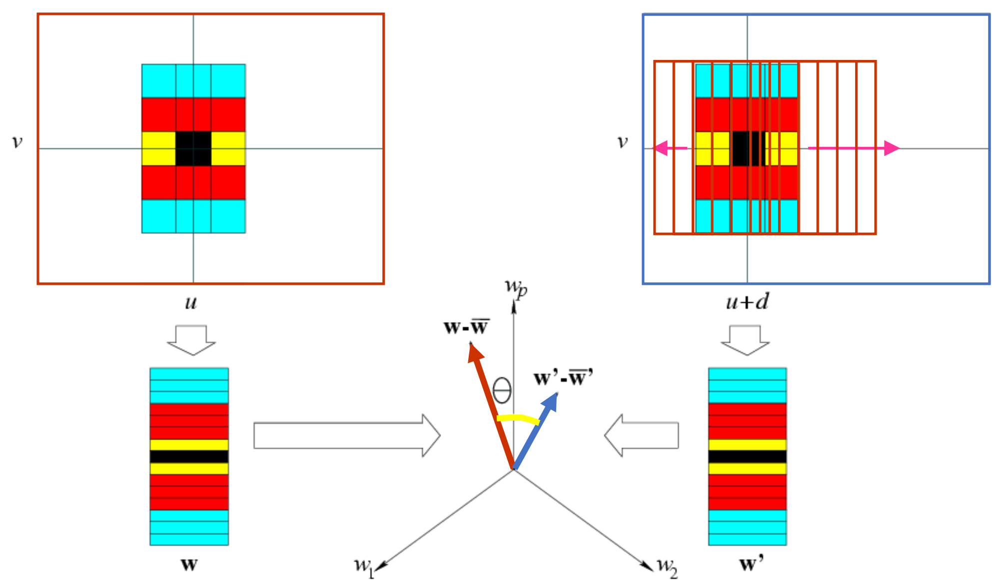
Neighborhood of corresponding points are similar in intensity patterns

Correspondence using correlation

- Subtract mean from images: $A \leftarrow A - \bar{A}$, $B \leftarrow B - \bar{B}$

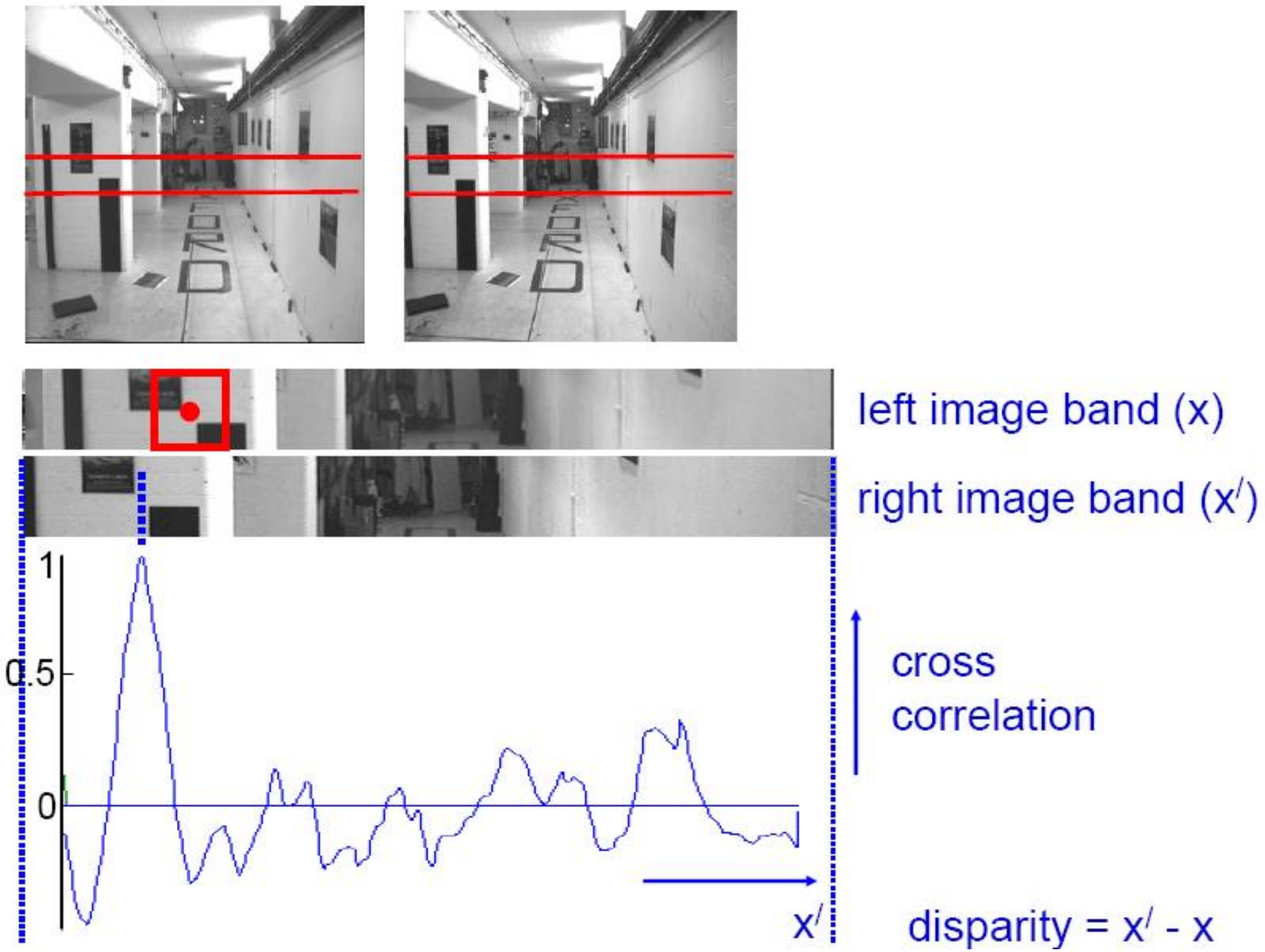
- Normalized cross-correlation: $NCC = \dfrac{\sum_i \sum_j A(i,j)B(i,j)}{\sqrt{\sum_i \sum_j A(i,j)^2}\sqrt{\sum_i \sum_j B(i,j)^2}}$

- Correlation-based window matching



Correspondence using correlation

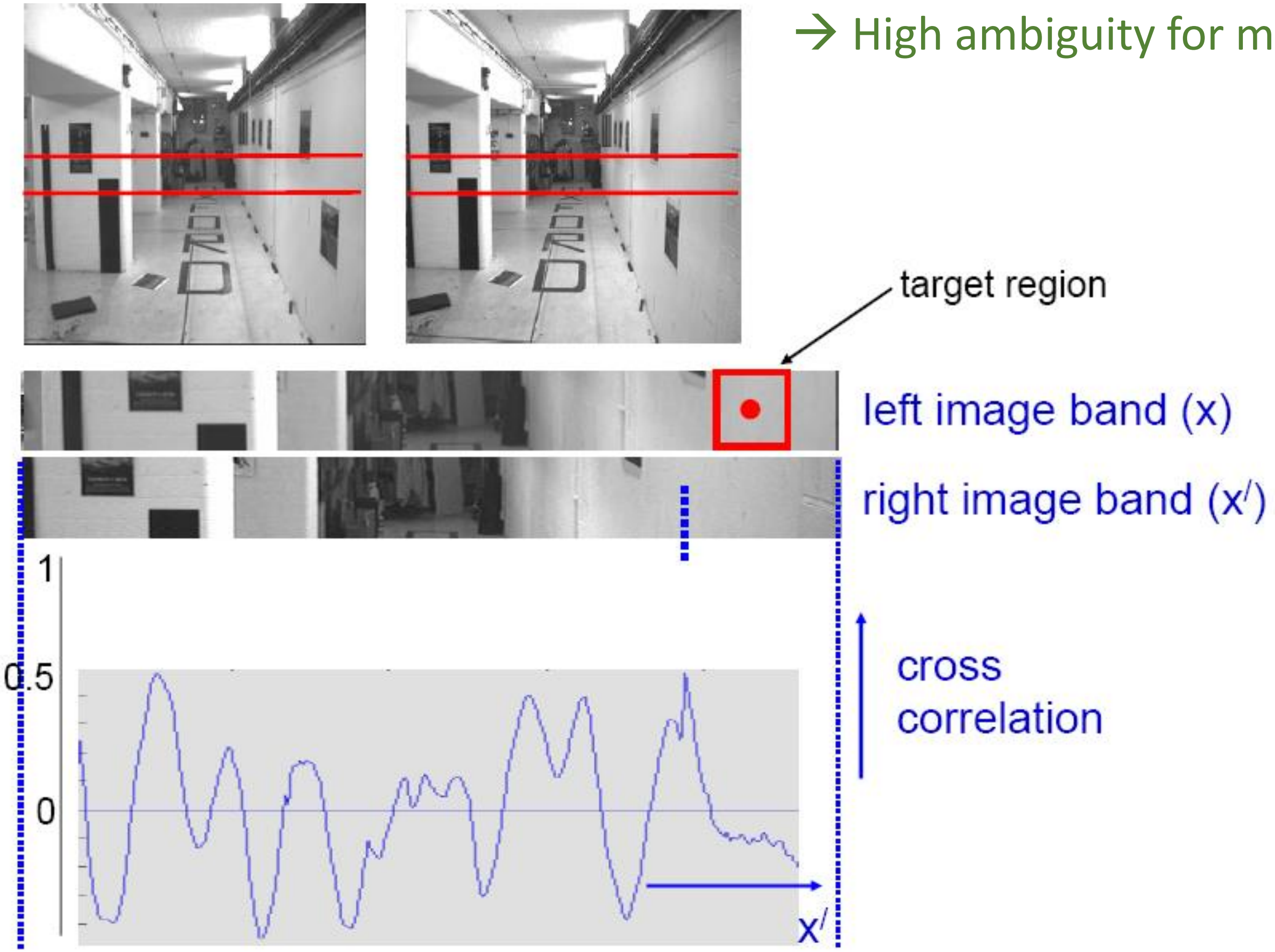
- Correlation-based window matching



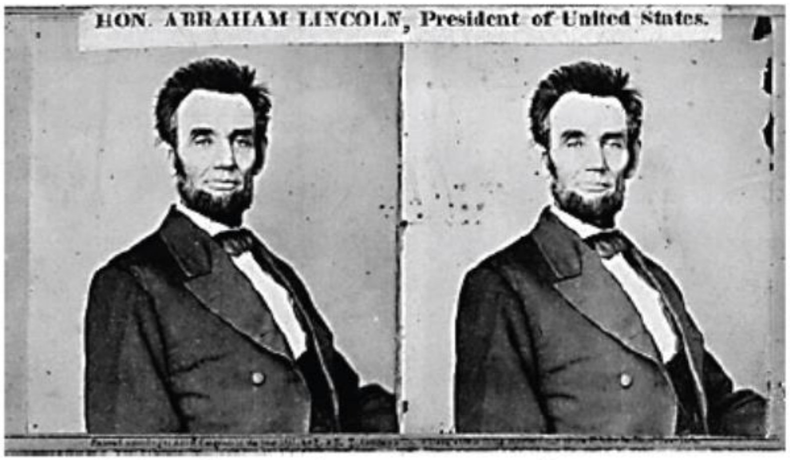
Limitations of similarity constraint

- Textureless regions

Textureless regions are non-distinct
→ High ambiguity for matches



Limitations of similarity constraint



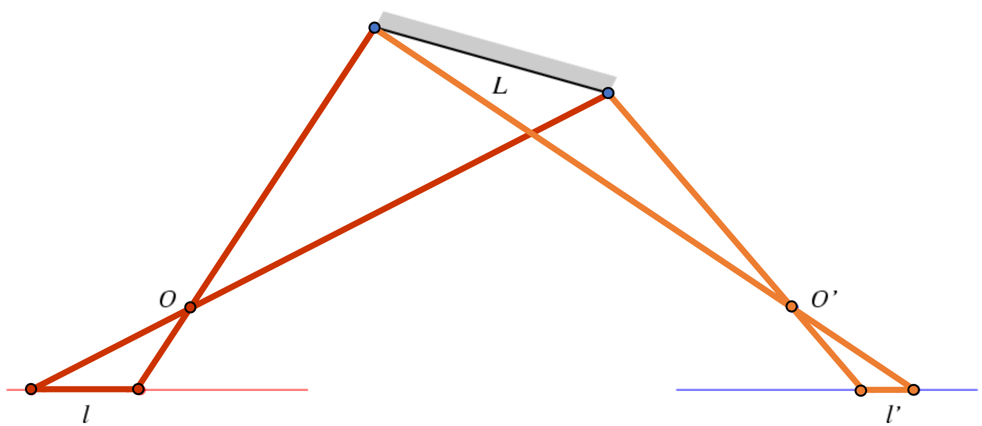
Textureless surfaces

Occlusions, repetition

Specular surfaces

Limitations of similarity constraint

- Foreshortening problem
  - Correlation-based technique assume that the observed surface is parallel to the two image planes
  - The foreshortening of oblique surfaces depends on the position of the cameras observing them

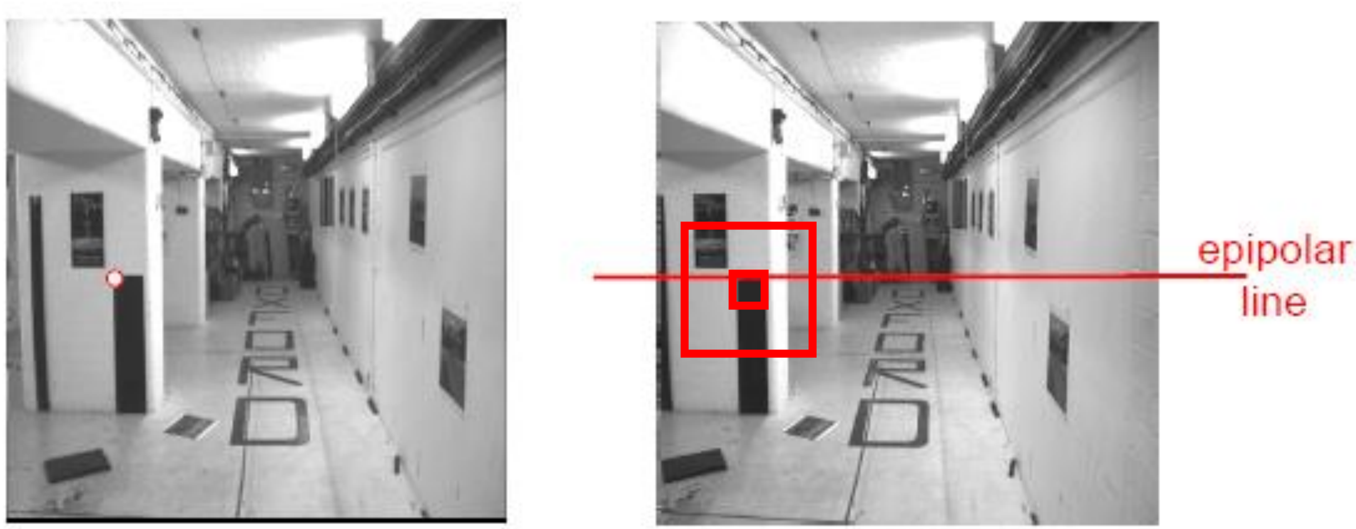


Limitations of similarity constraint

- Foreshortening problem: Solution
  - Add a second pass using disparity estimates to warp the correlation windows to compensate for inequal amounts of foreshortening in the two pictures

Effect of window size

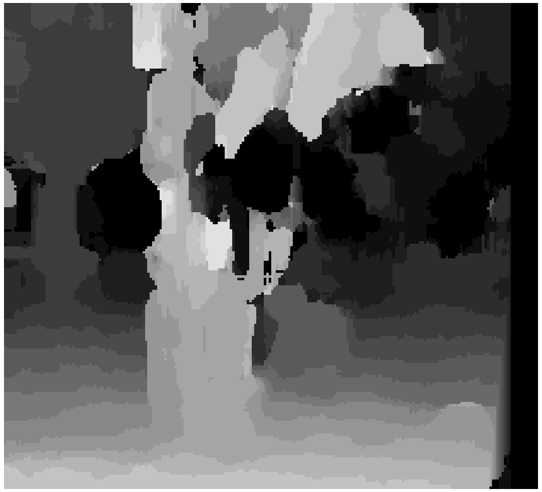
- Effect of window size



Effect of window size

- Smaller window
  - More detail but more noise
- Larger window
  - Smoother disparity maps but less detail

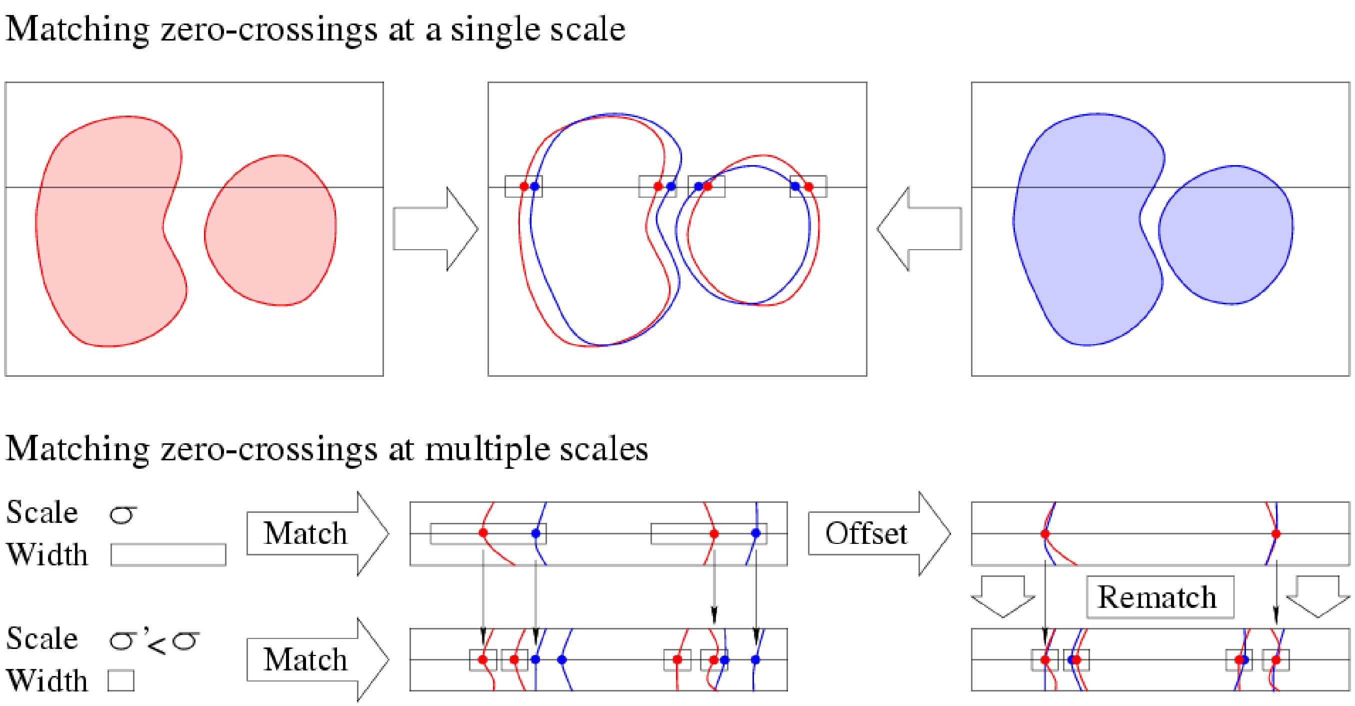
W = 3



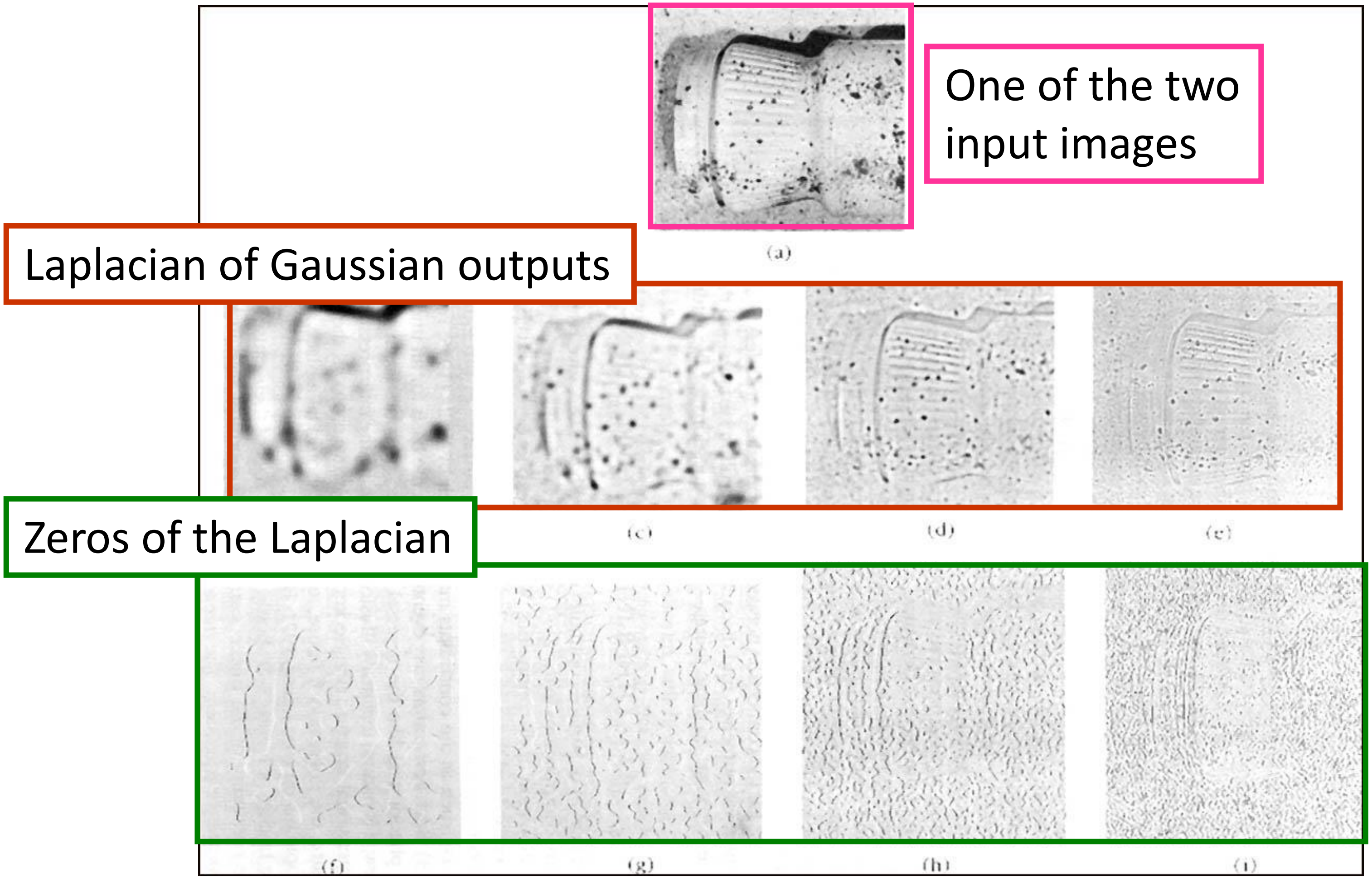
W = 20

Multi-scale edge matching

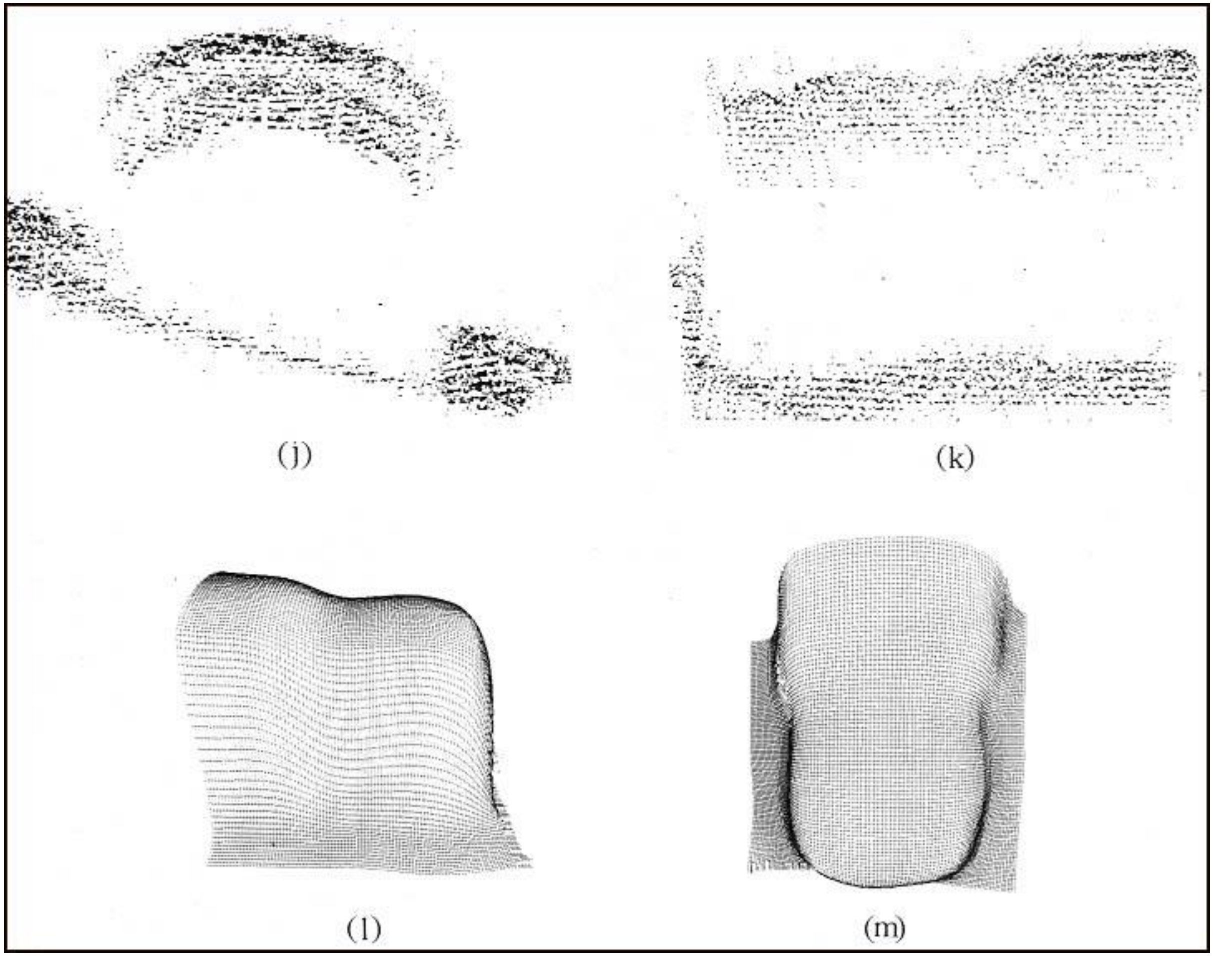
- Edges are found by repeatedly smoothing the image and detecting the zero crossings of the second derivative (Laplacian)
- Matches at coarse scales are used to offset the search for matches at fine scales



Multi-scale edge matching



Multi-scale edge matching

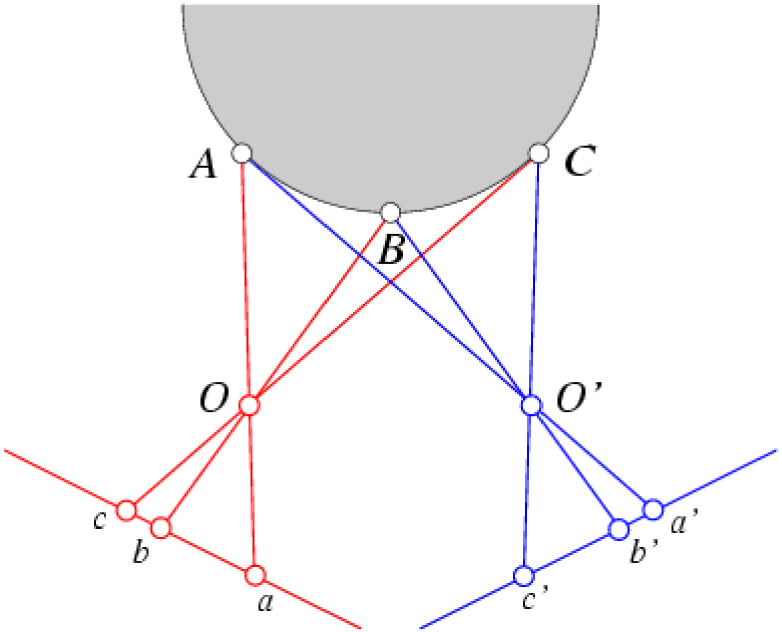


Two views of depth map

Two views of surface obtained by interpolating the reconstructed points

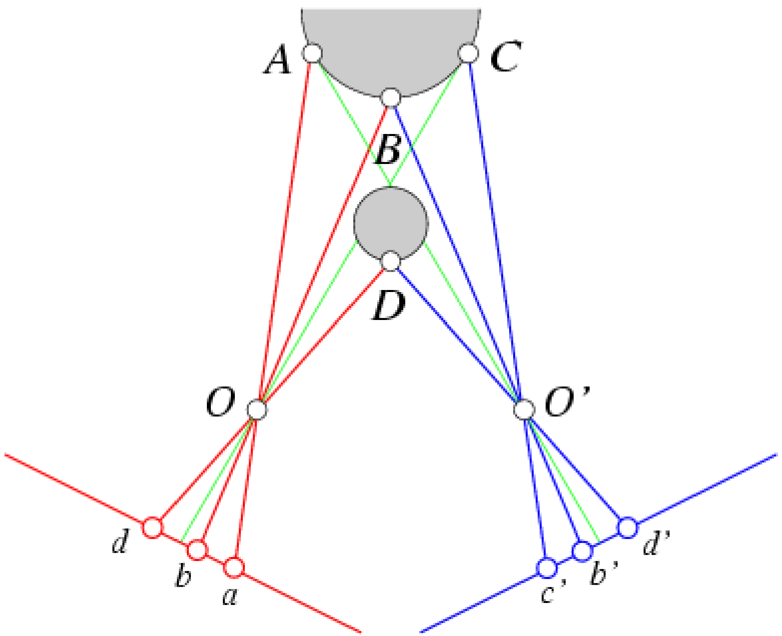
Ordering constraint

- In general, the points are in the same order on both epipolar lines



Ordering constraint

- But it is not always the case



Approaches to find correspondences

- Intensity correlation-based approaches
  - (+) Dense disparity (disparity at each pixel)
  - (-) Foreshortening
    - Solution: Warp windows?
- Edge/feature matching approaches
  - (+) Solve the foreshortening problem
  - (-) Sparse disparity
    - Solution: Interpolate intermediate disparities
  - (-) Requires feature detection
- Dynamic programming
  - (+) Use both feature and intensities