3D Data Processing

Stereo Vision

Hyoseok Hwang

Depth Perception

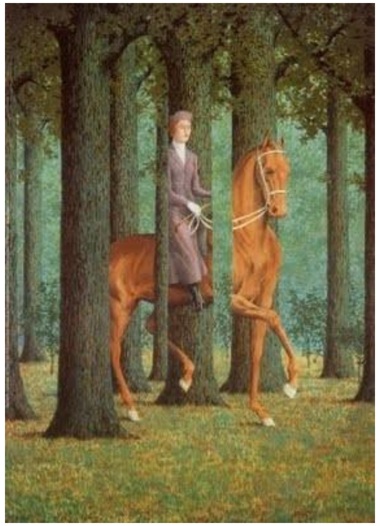
- What cues help us to perceive 3D shape and depth?
  - Monocular cues
  - Binocular cues

What depth cues exist in this 2D image?

Monocular Cues

- Perspective effects
- Occlusion
- Relative Size
- Blur
- Texture Gradients
- Shading

perspective



occlusion

shading

Texture gradient

relative size

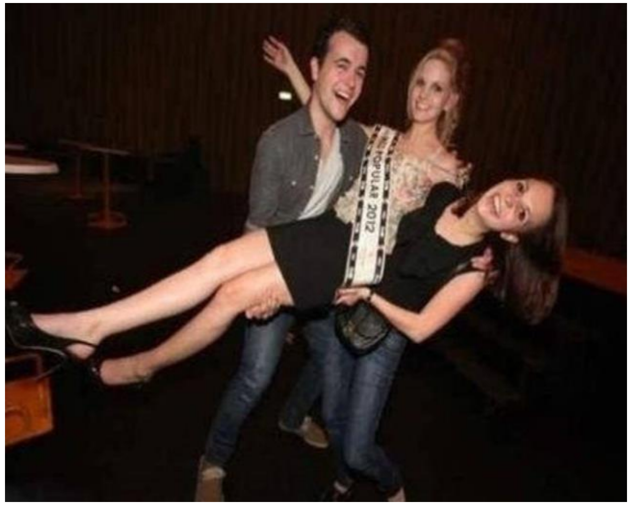
blur

http://rene-magritte-paintings.blogspot.com/2009/11/le-blanc-seing.html

Monocular Cues is insufficient

- There is insufficient evidence in one picture.






- In fact, this is because of your brain.

https://www.dispatch.co.kr/281591

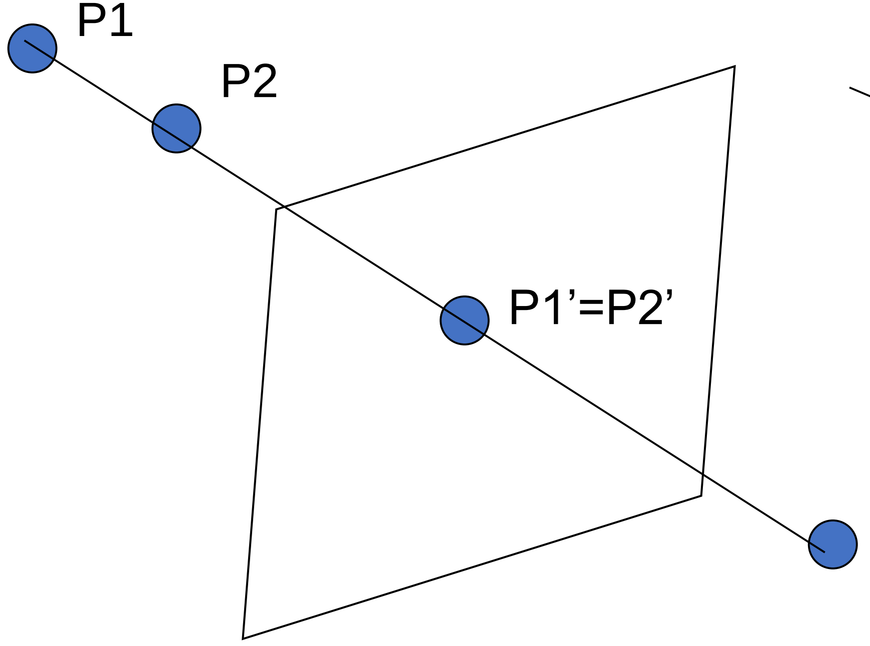
Why multiple views is necessary?

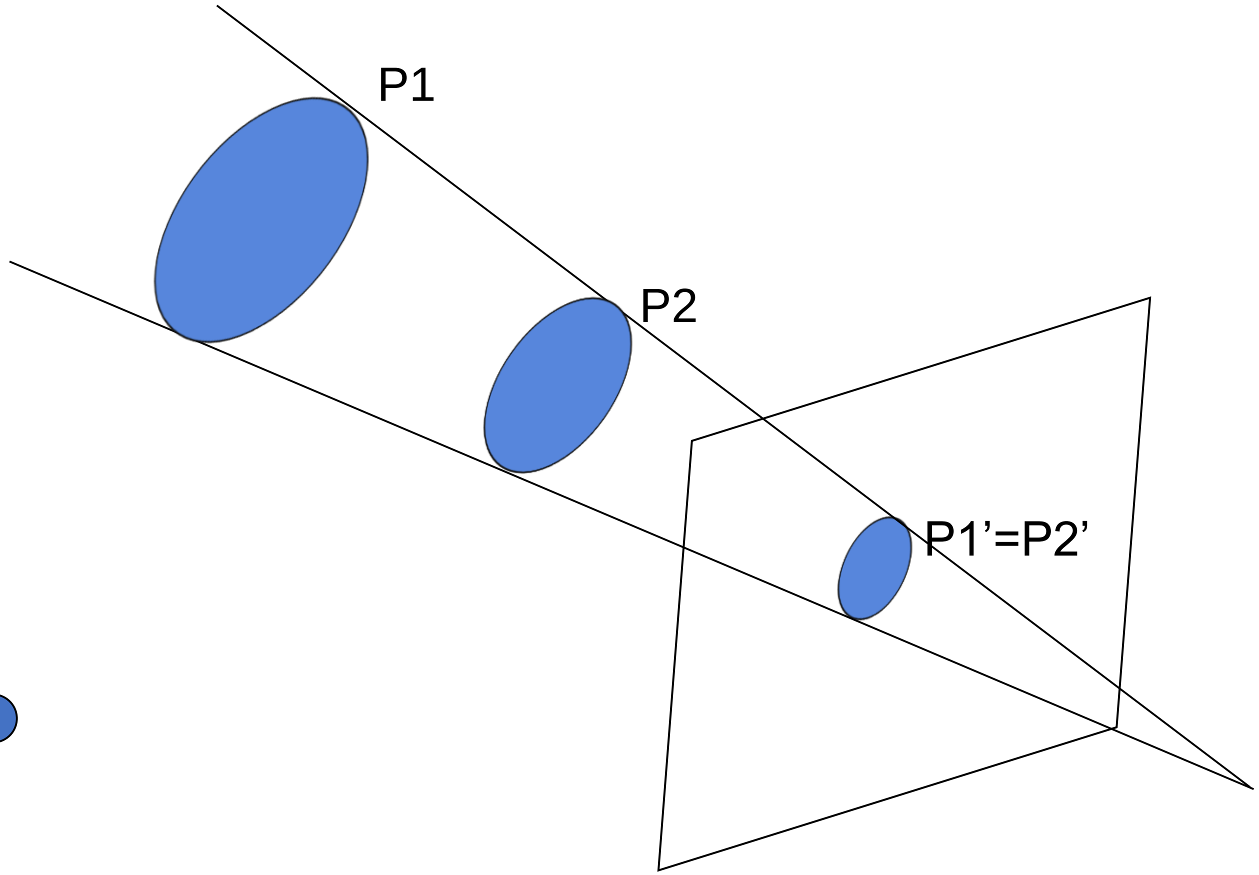
- Structure and depth are inherently ambiguous from single views.



Binocular Cues

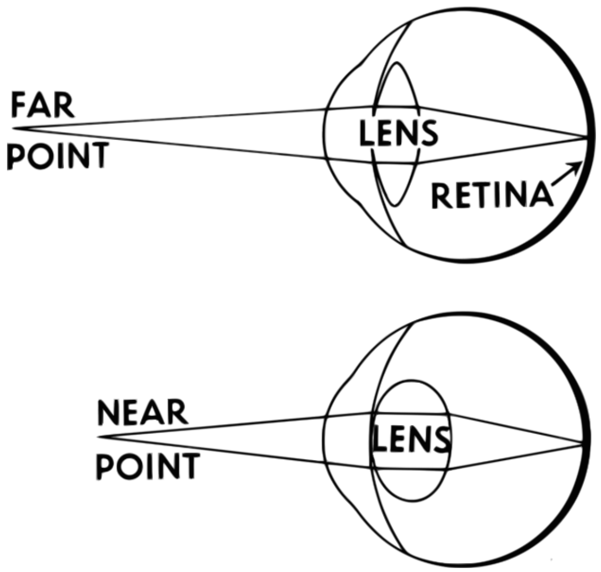
- Ambiguity of monocular cues
  - There are infinite candidates in 3D space for points (areas) on the image.



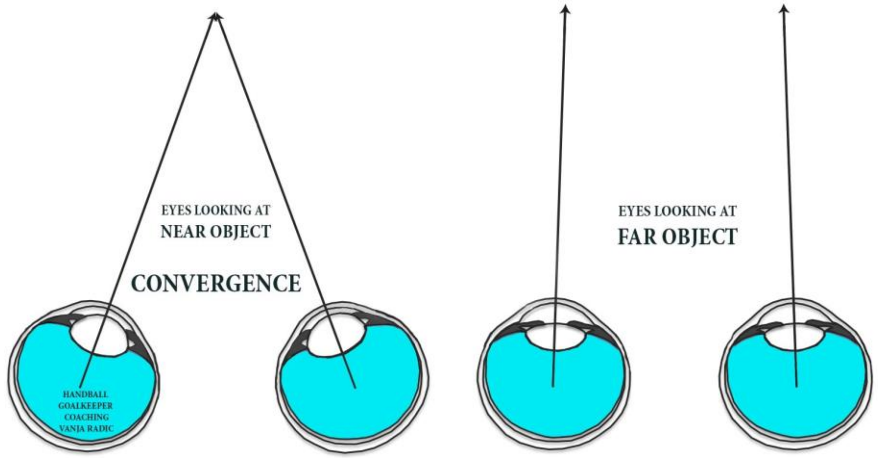


Binocular Cues

- Human stereopsis: vergence and accommodation
  - **Accommodation**: the process by which the vertebrate eye changes optical power to maintain a clear image or focus on an object as its distance varies.
  - **Vergence**: the inward or outward turning movement of the eyes



accommodation

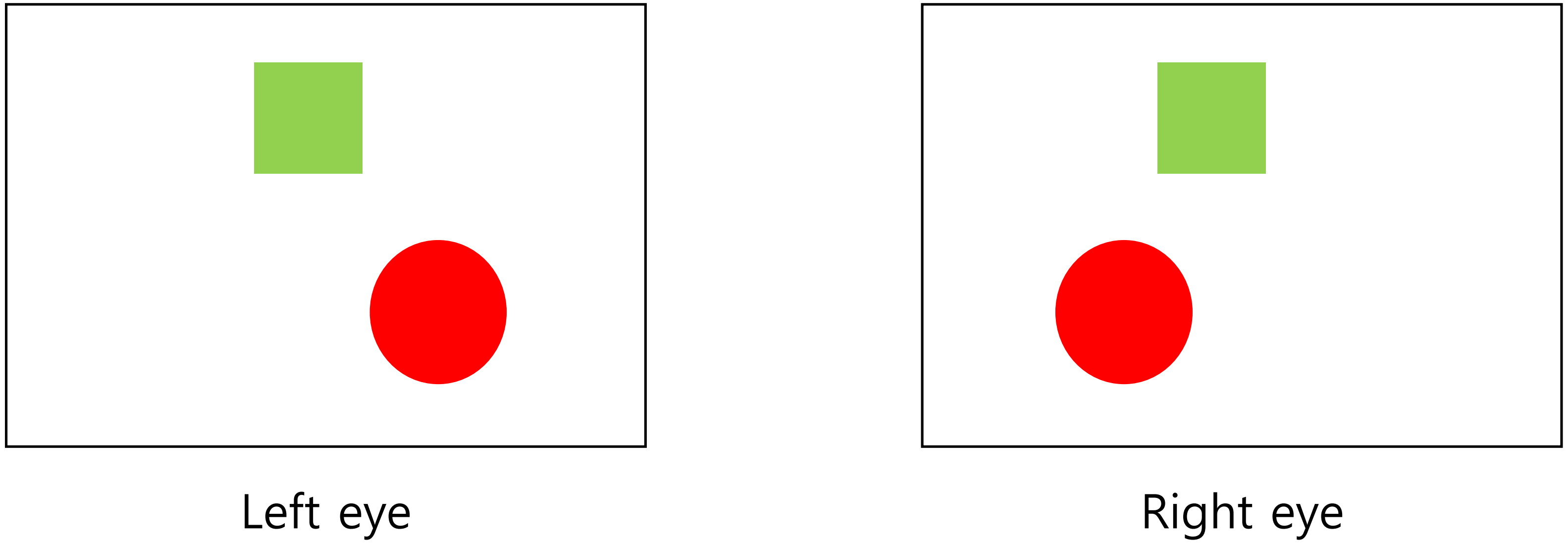


vergence

Binocular Cues

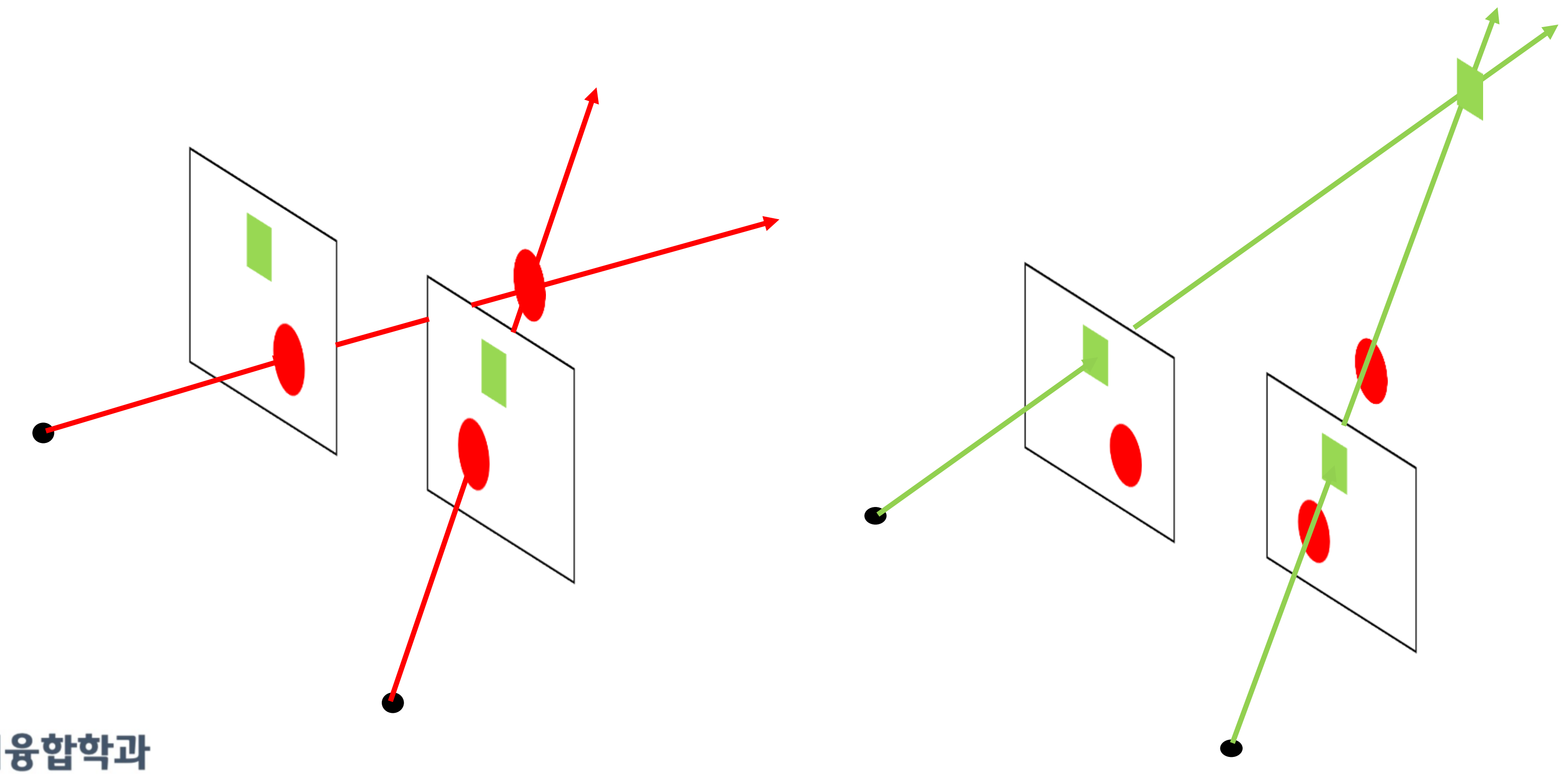
- Human stereopsis: **disparity**
  - Disparity occurs when eyes fixate on one object; others appear at different visual angles

- Example
  - Which is closer?



Binocular Cues

- Human stereopsis: **disparity**
  - In one image, we can know where the pixel came from.
  - When back-projecting light on multiple (at least two or more) identical points, the point where they meet is the actual 3D location.



Binocular Cues

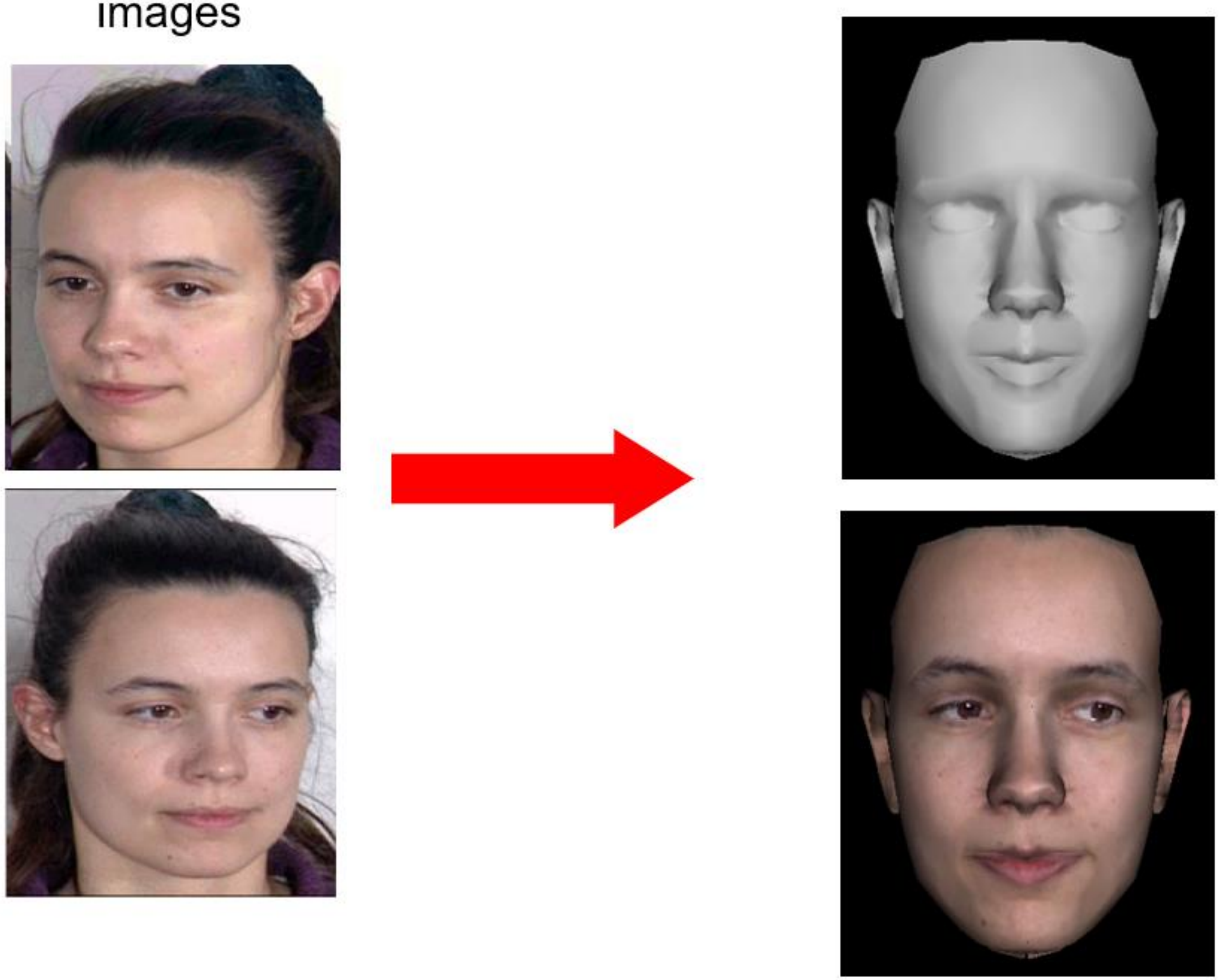
- Human stereopsis: **disparity**
  - In one image, we can know where the pixel came from.
  - When back-projecting light on multiple (at least two or more) identi cal points, the point where they meet is the actual 3D location.

  - To estimate accurate 3D position of corresponding point, we should know
    - Camera intrinsic parameters of each camera
    - Pose(translation, rotation) between origins of camera coordinates

Binocular Cues

- Example

images



shape

surface reflectance

Scenarios

- The two images can arise from

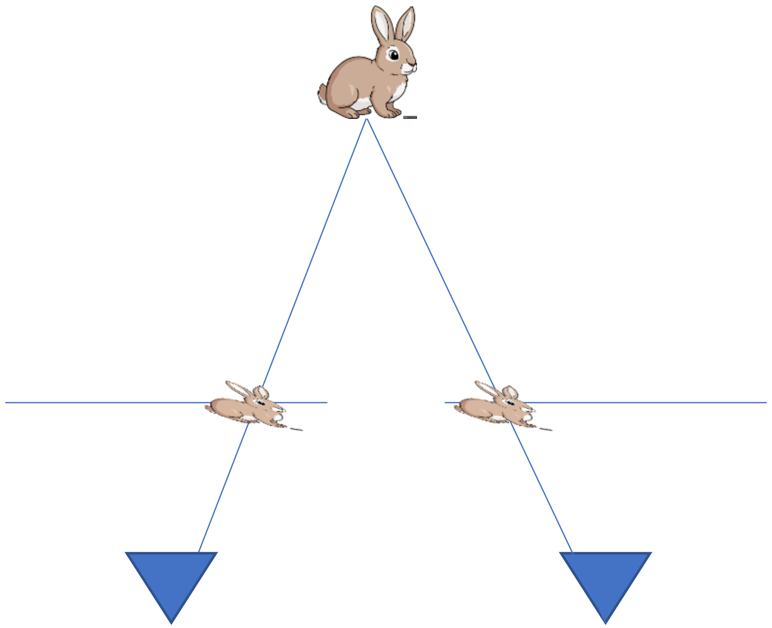
- A stereo rig consisting of two cameras
  - the two images are acquired simultaneously
- or

- A single moving camera (static scene)
  - the two images are acquired sequentially

- The two scenarios are geometrically equivalent

Triangulation

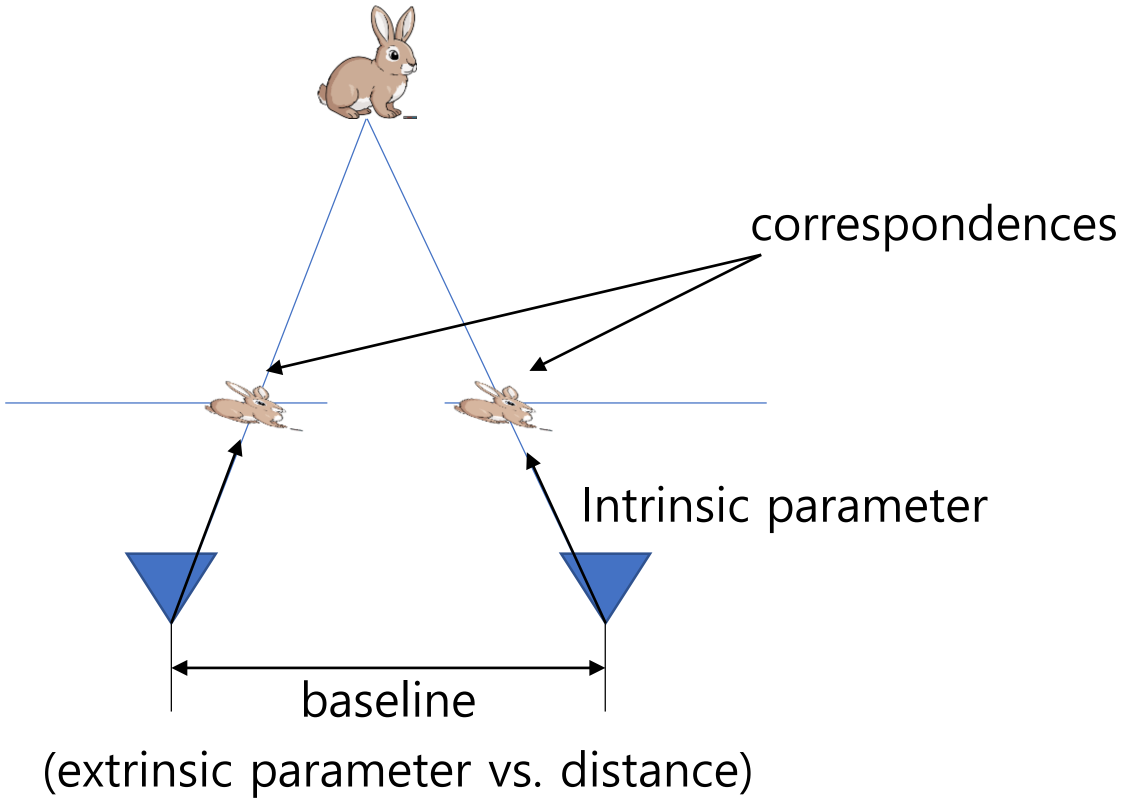
- What are necessary to get 3D information from 2D?



Triangulation

- What are necessary to get 3D information from 2D?

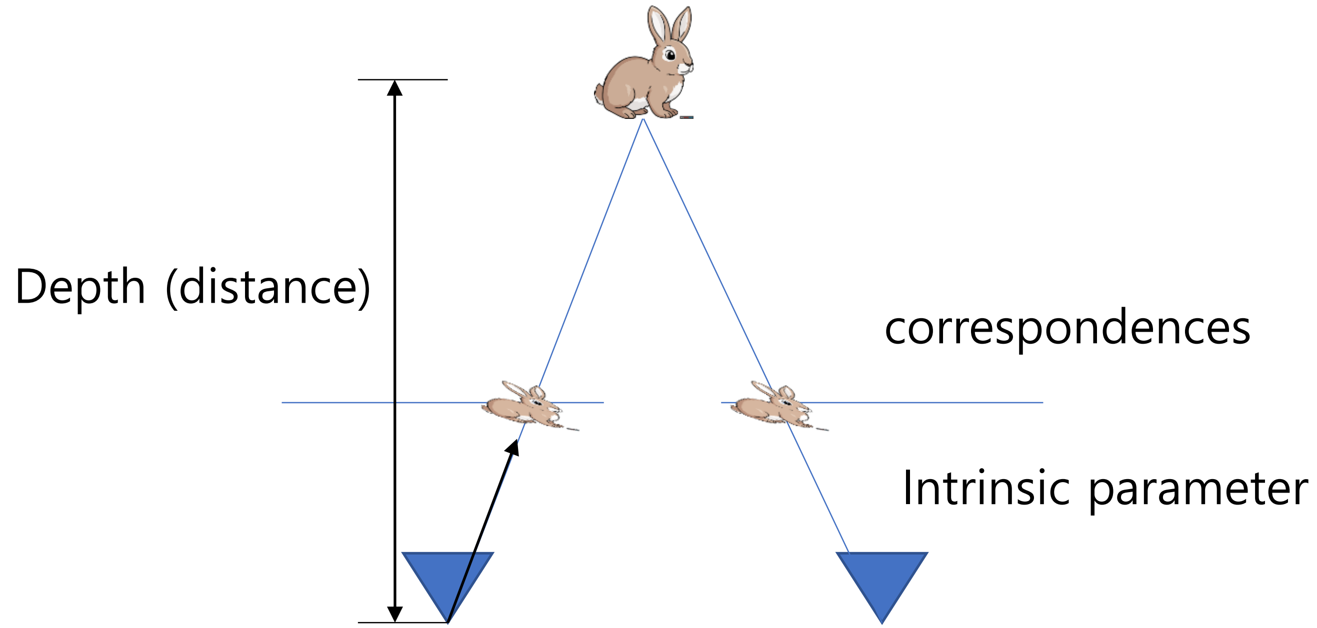
Method #1



Triangulation

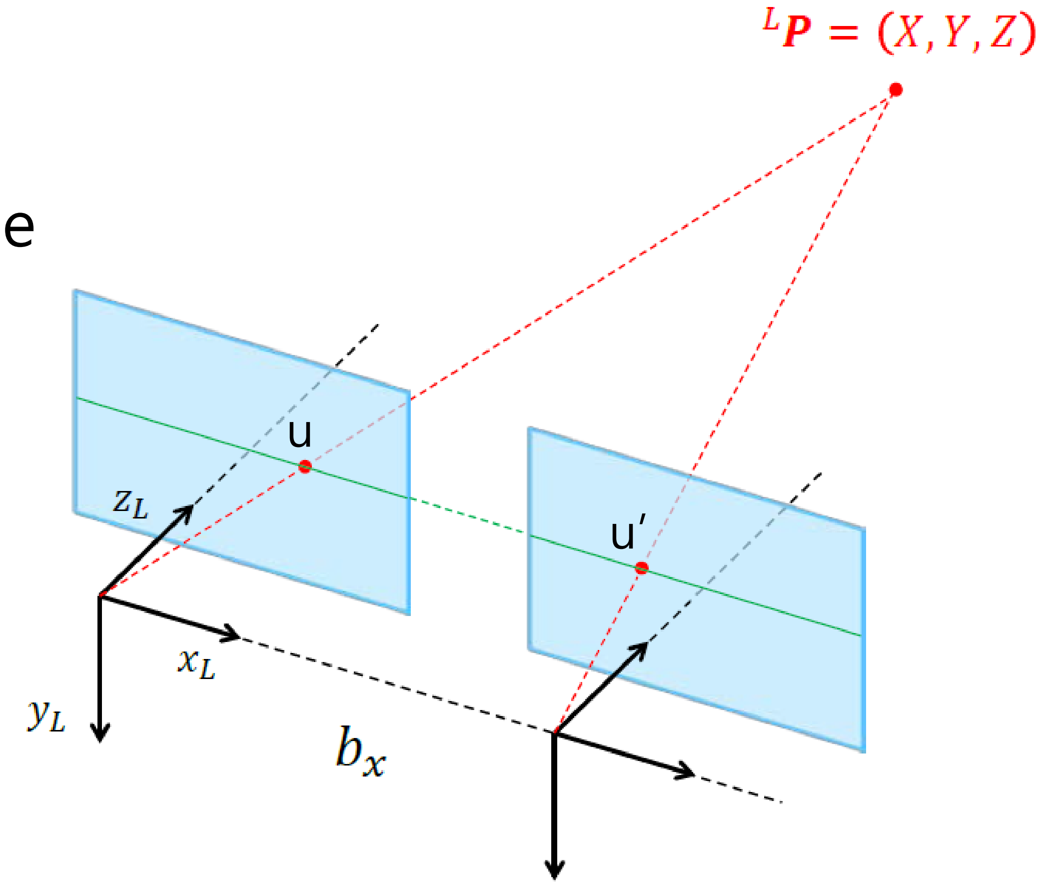
- What are necessary to get 3D information from 2D?

Method #2



Stereo geometry

- Parallel identical cameras
  - Translated along *x*-axis
- Horizontal epipolar lines
  - Corresponding points lie along the same row in the two images
- Make is easy to find corresponding points



Stereo geometry

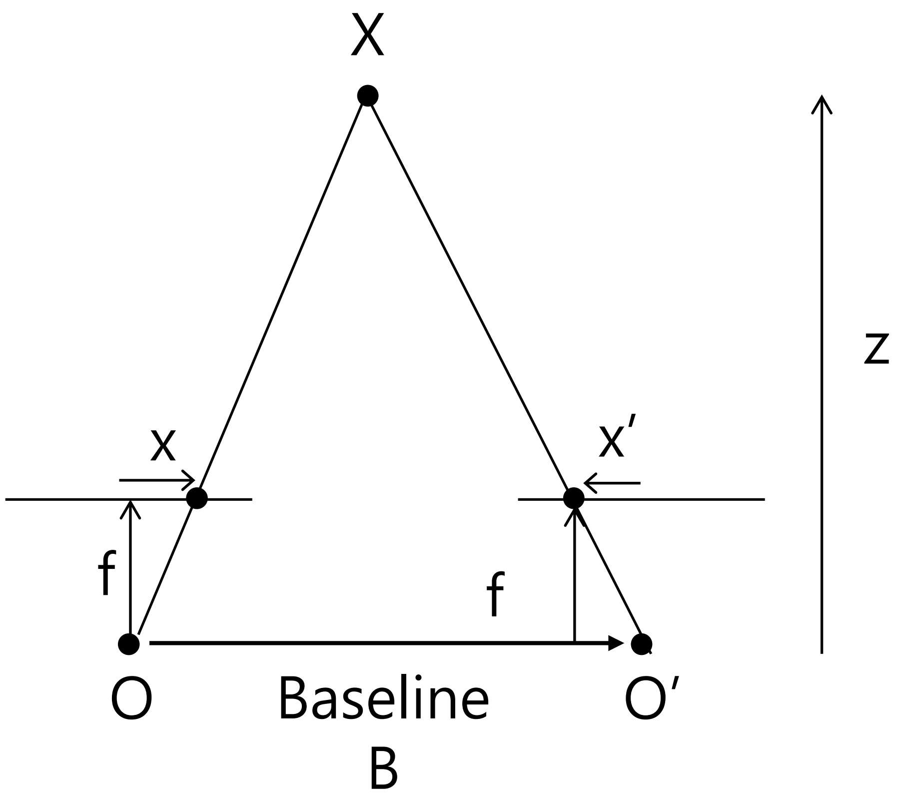
- Triangulation
  - depth from disparity
- Disparity is inversely proportional to depth

$$\frac{O - O' - x + x'}{O - O'} = \frac{z - f}{z}$$

$$1 + \frac{-x + x'}{O - O'} = 1 + \frac{f}{z}$$

$$\frac{x - x'}{O - O'} = \frac{f}{z}$$

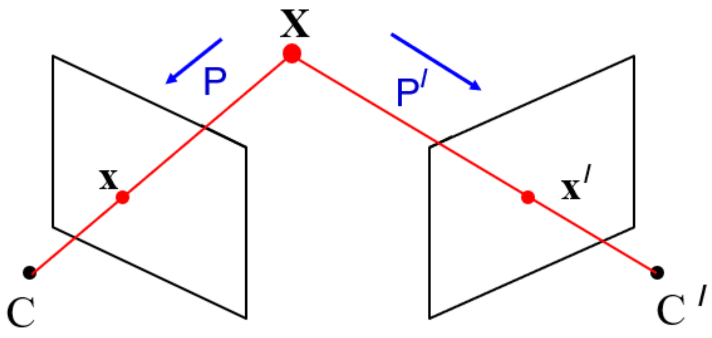
$$disparity = x - x' = \frac{B \cdot f}{z}$$



Stereo geometry

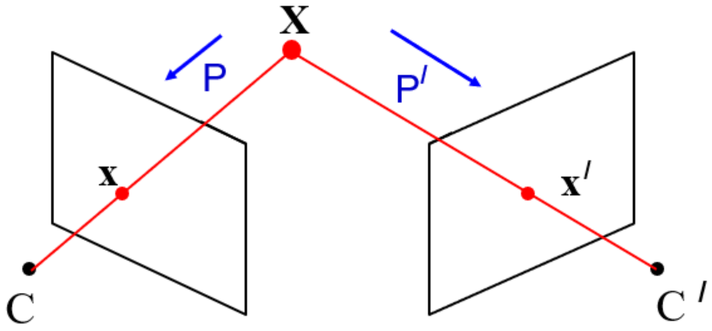
- Usual cases
  - Image planes are not parallel
  - Base line is not parallel to x-axis only
  - The extrinsic parameter $[R|t]$ of the previous case → unusual cases

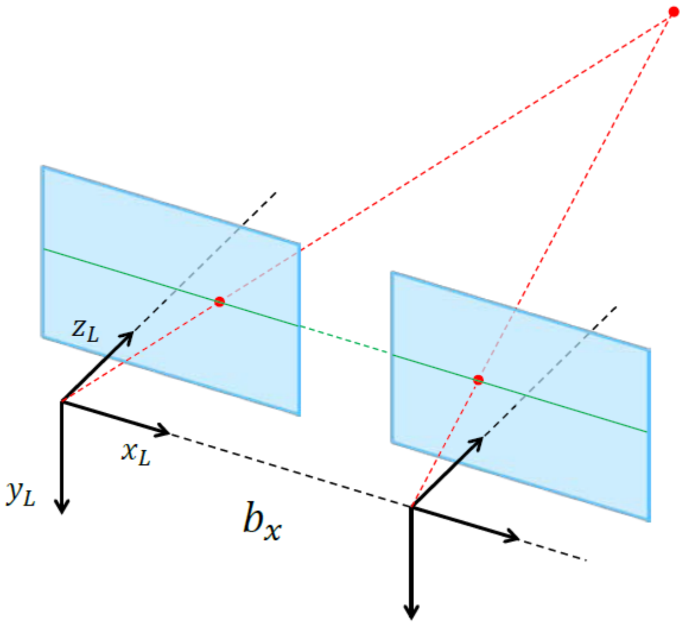
$$\begin{bmatrix} 1 & 0 & 0 & dX \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



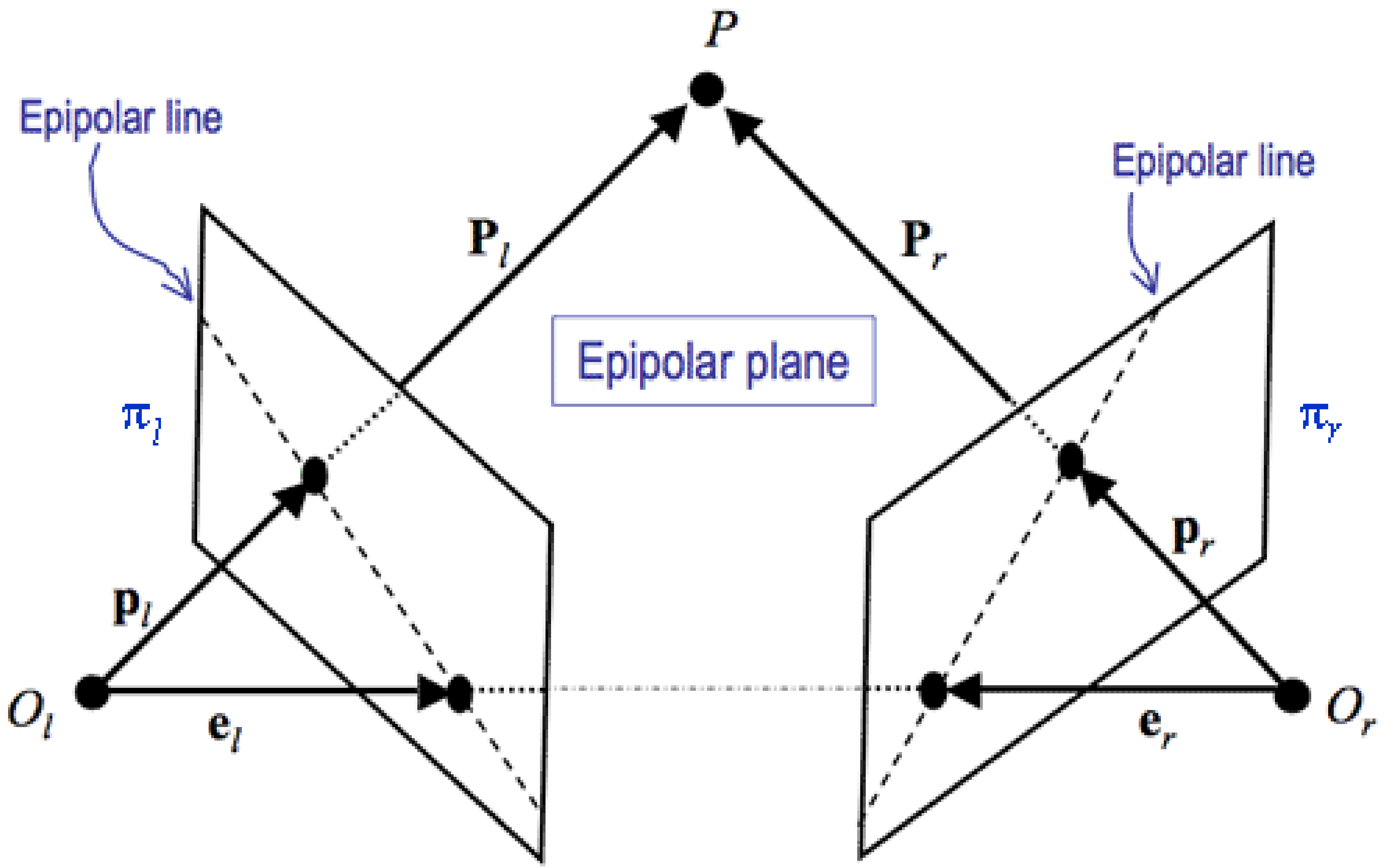
Stereo geometry

- What's the difference between having parallel and non-parallel image sensors?
- OR
- What would we gain if the image sensors were parallel?





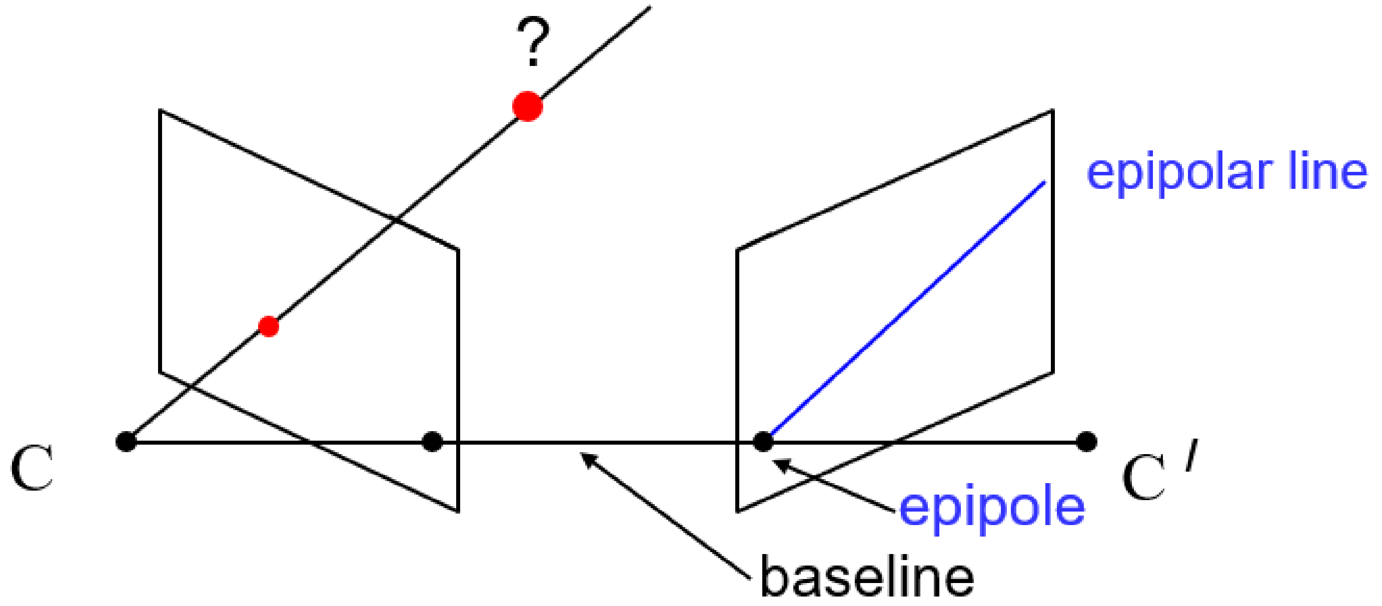
Epipolar geometry



https://www.cs.auckland.ac.nz/courses/compsci773s1t/lectures/773-GG/lectA-773.htm

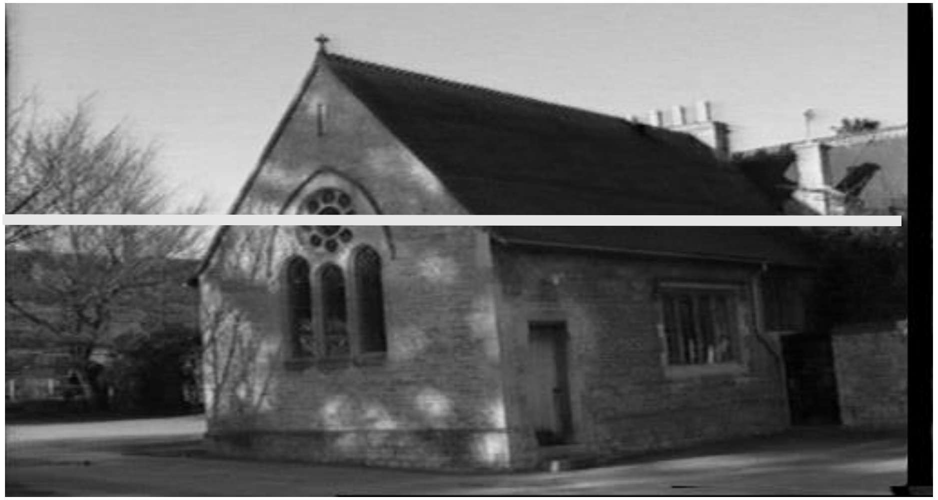
Epipolar geometry

- Given an image point in one view, where is the corresponding point in the other view?
  - A point in one view "generates" an epipolar line in the other view
  - The corresponding point lies on this line



Epipolar line

- Epipolar constraint
  - Reduces correspondence problem to 1D search along an epipolar line

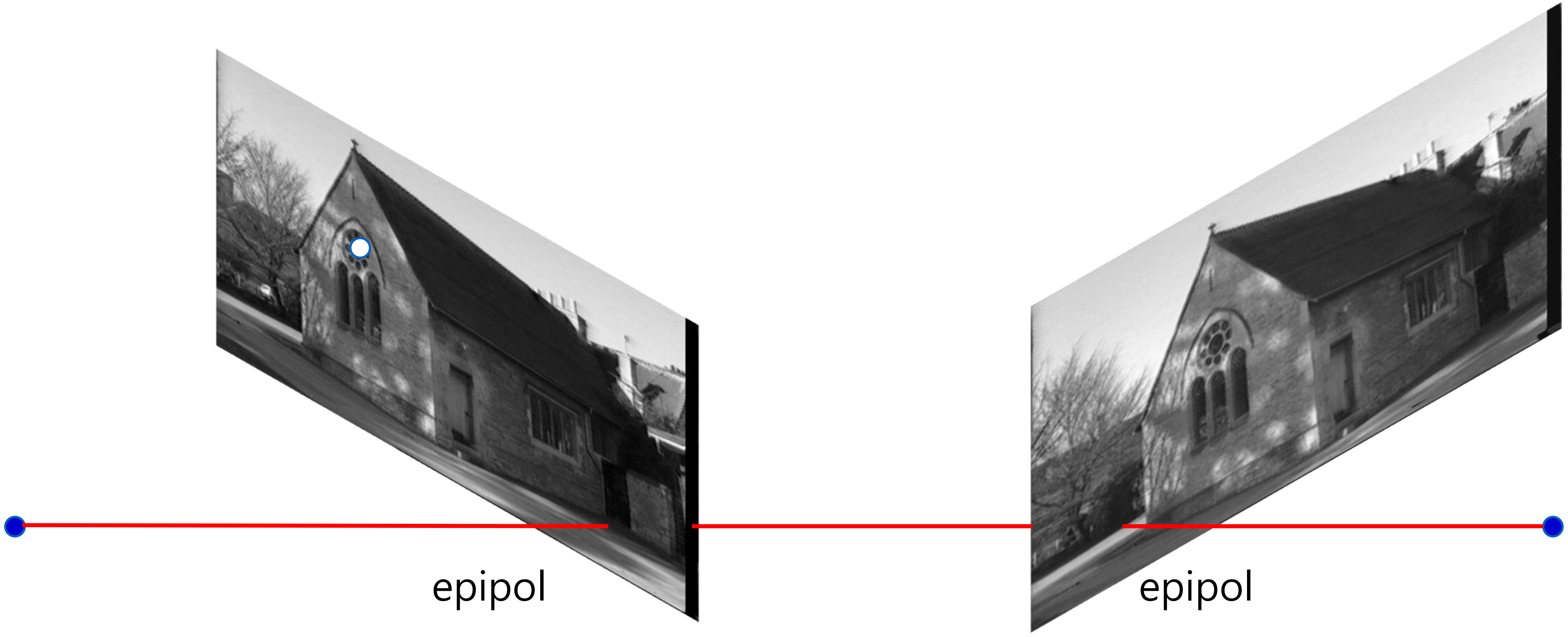


Epipolar line

- Epipolar constraint
  - Reduces correspondence problem to 1D search along an epipolar line

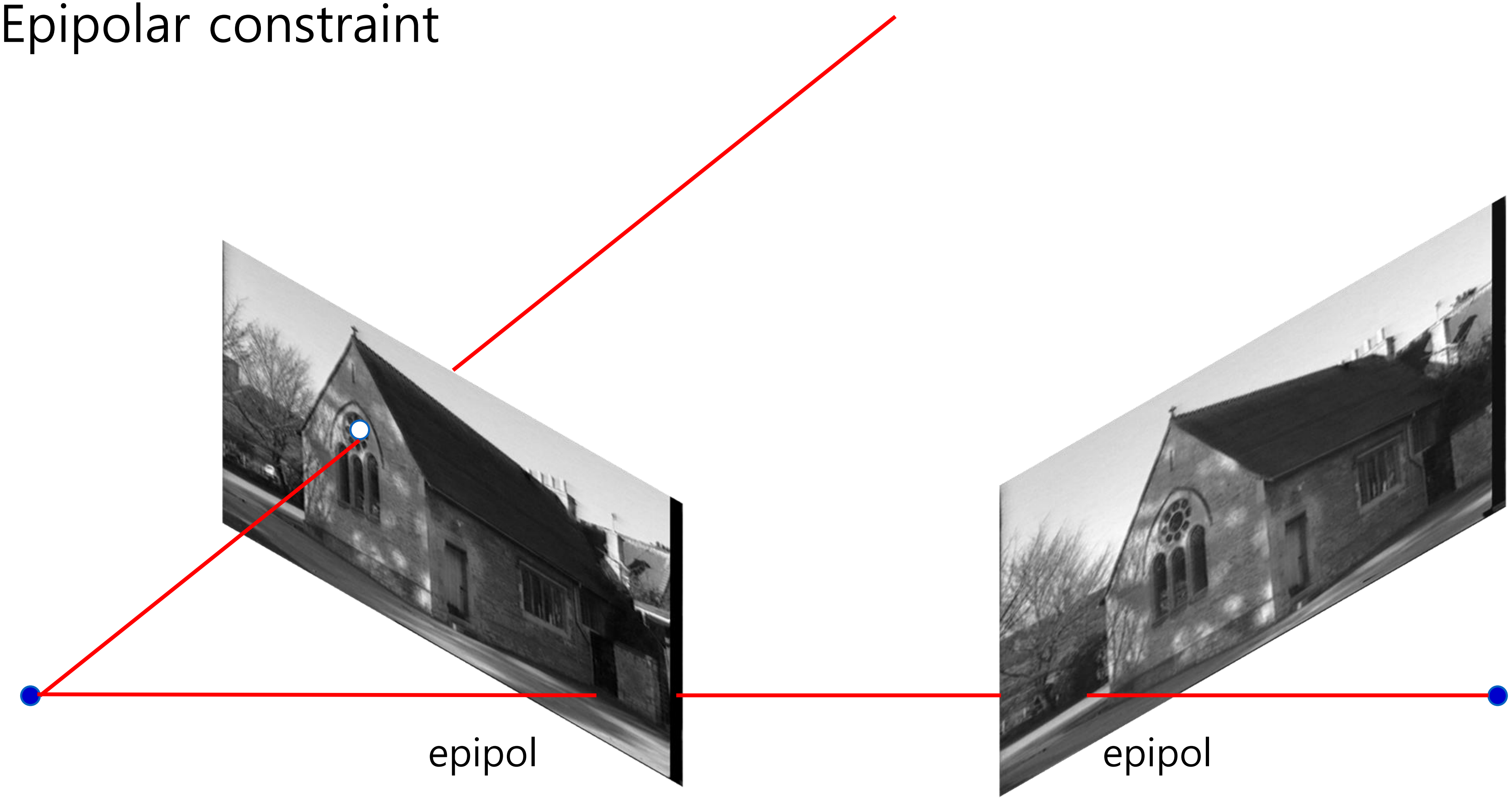
Epipolar line

- Epipolar constraint
  - Reduces correspondence problem to 1D search along an epipolar line



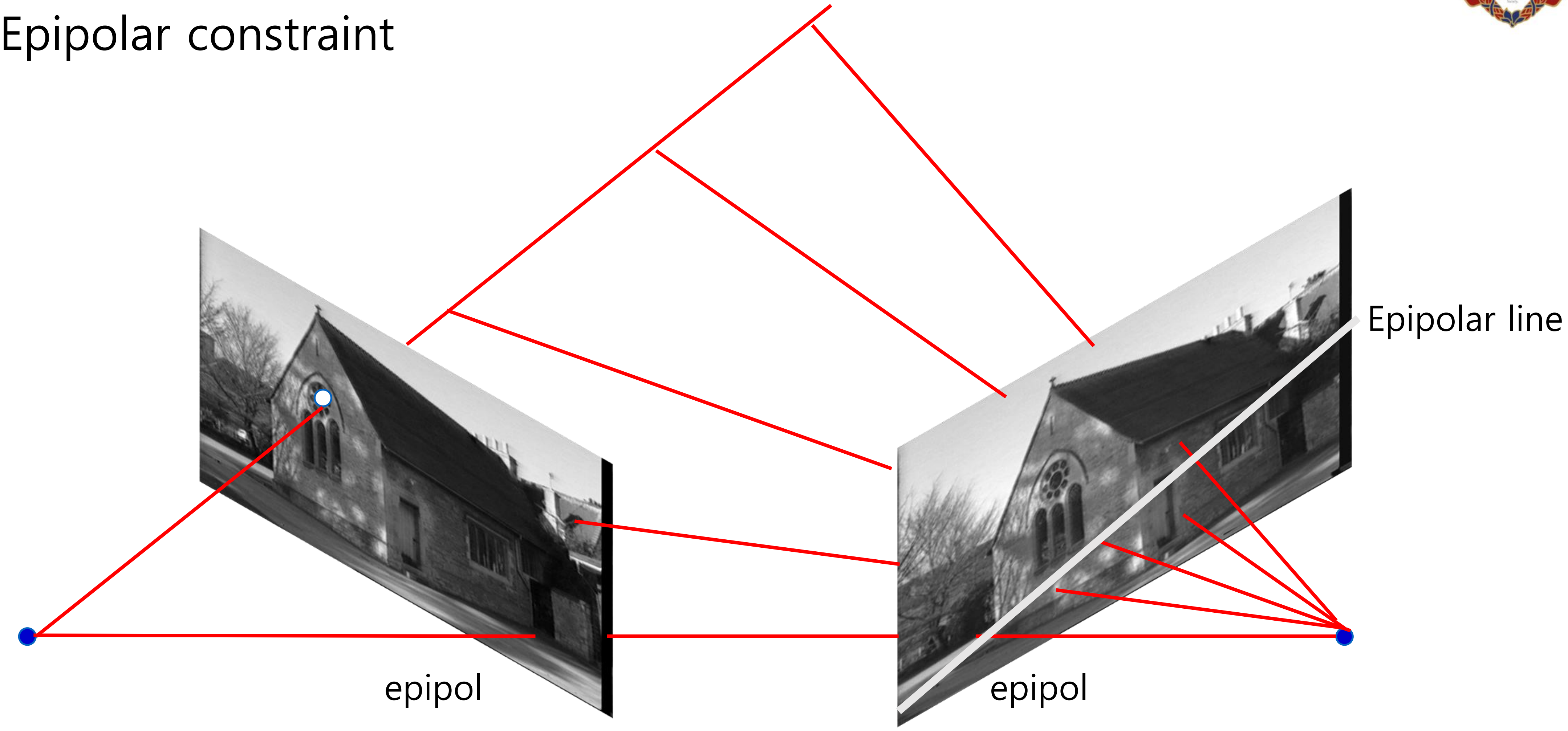
Epipolar line

- Epipolar constraint



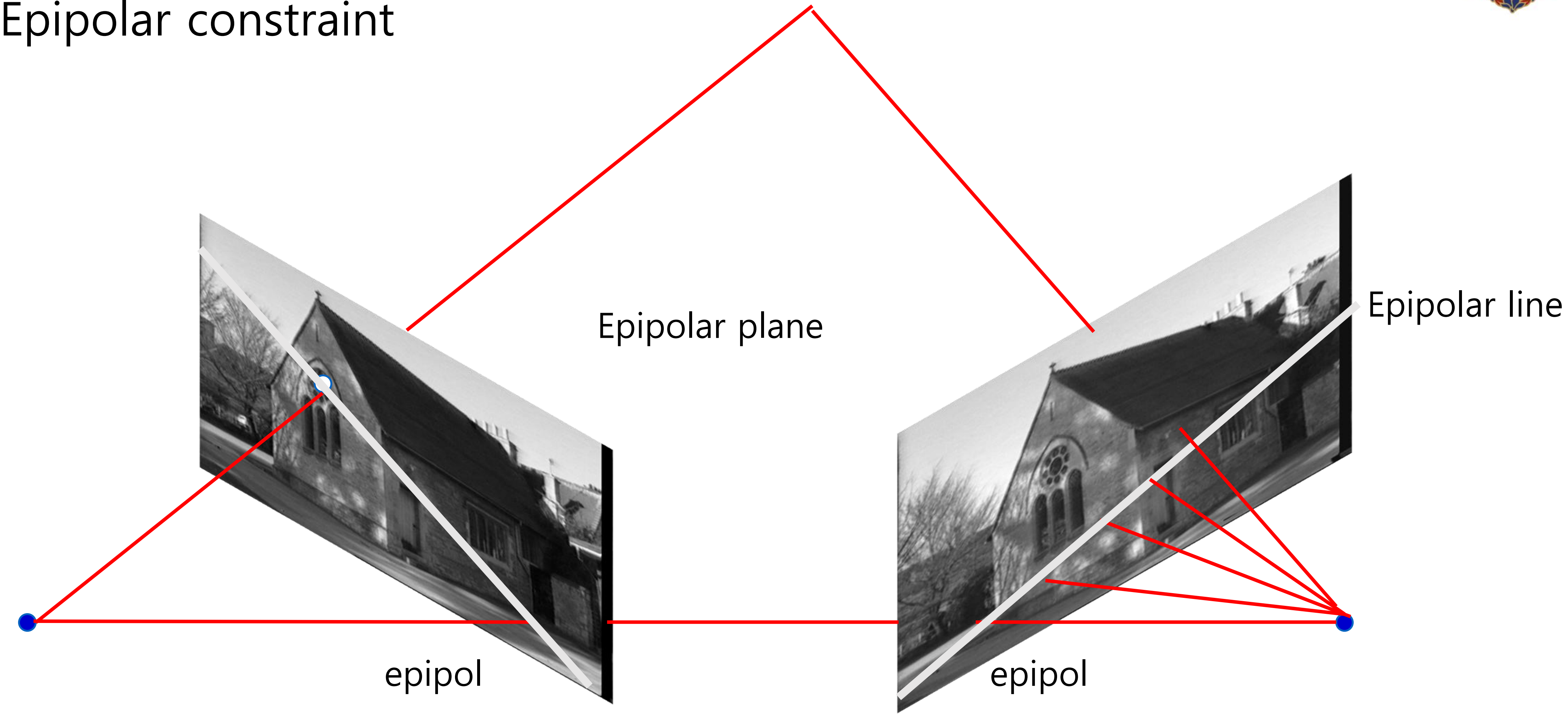
Epipolar line

- Epipolar constraint



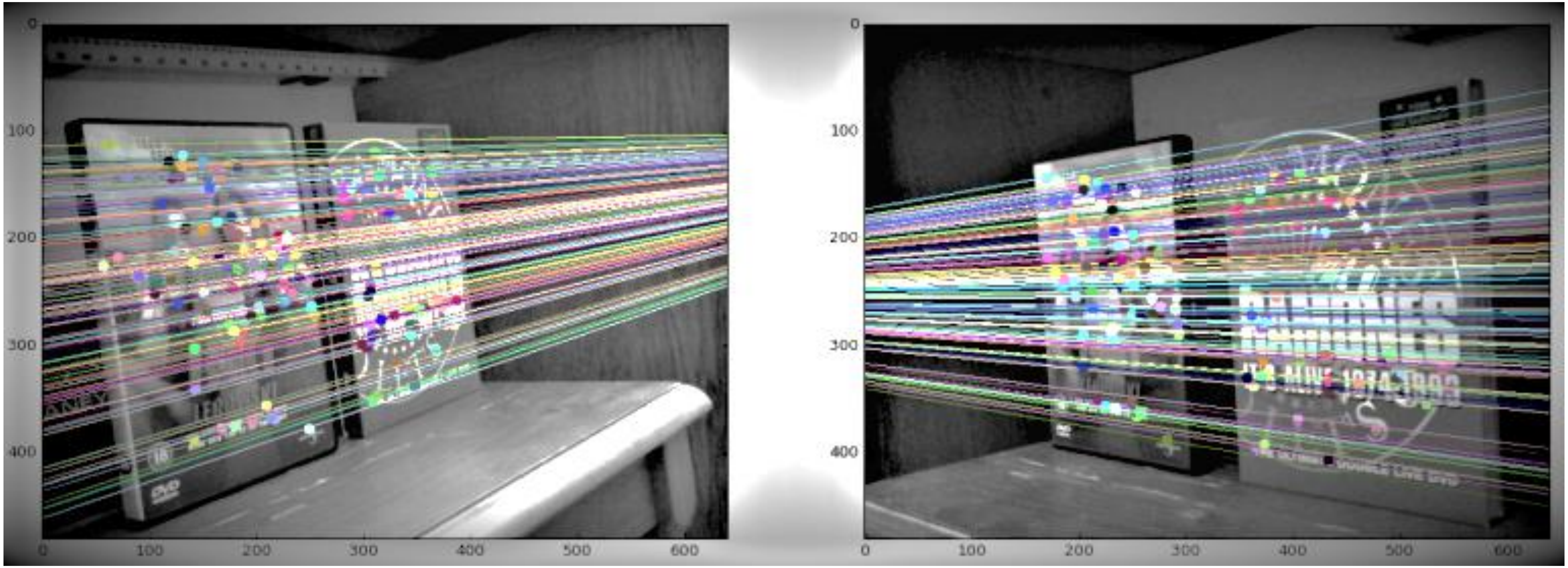
Epipolar line

- Epipolar constraint



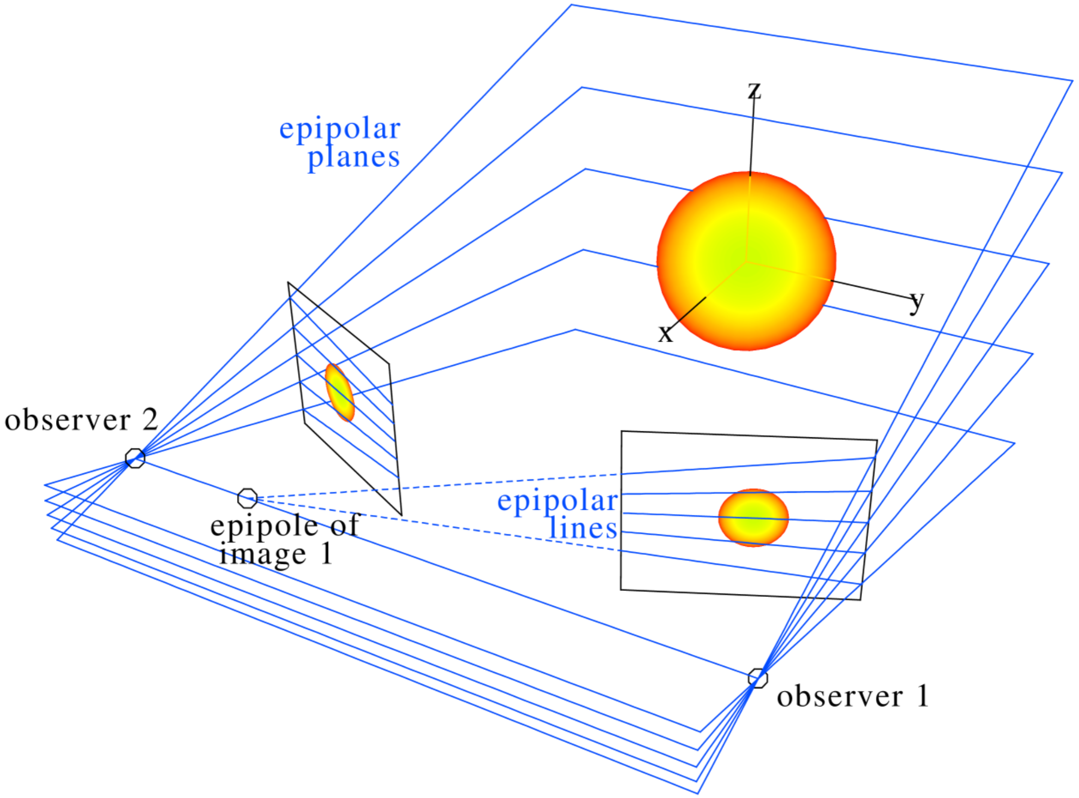
Epipolar line

- Example



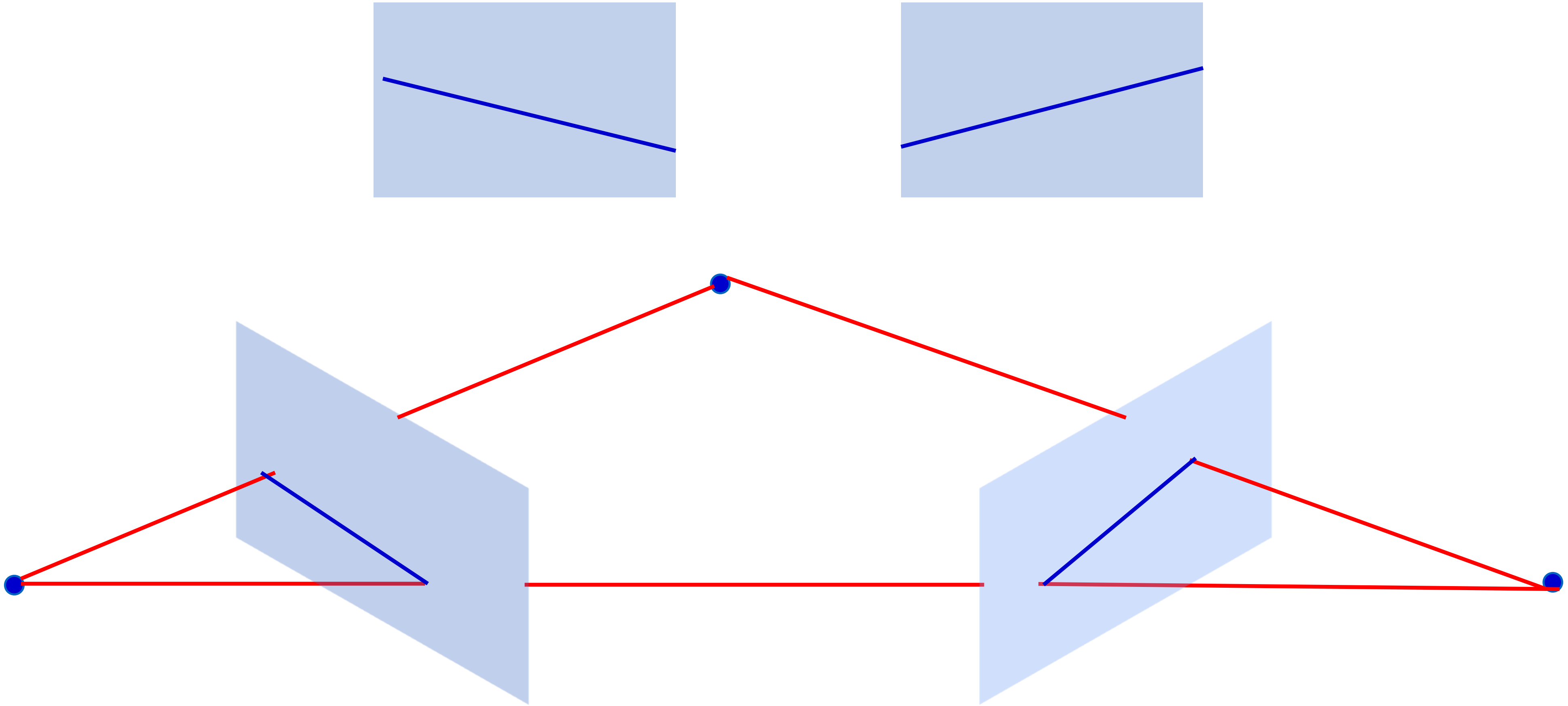
Epipolar geometry

- How can we use epipolar geometry?
  - 3D pose estimation using essential matrix and fundamental matrix
- Make it easy to find corresponding



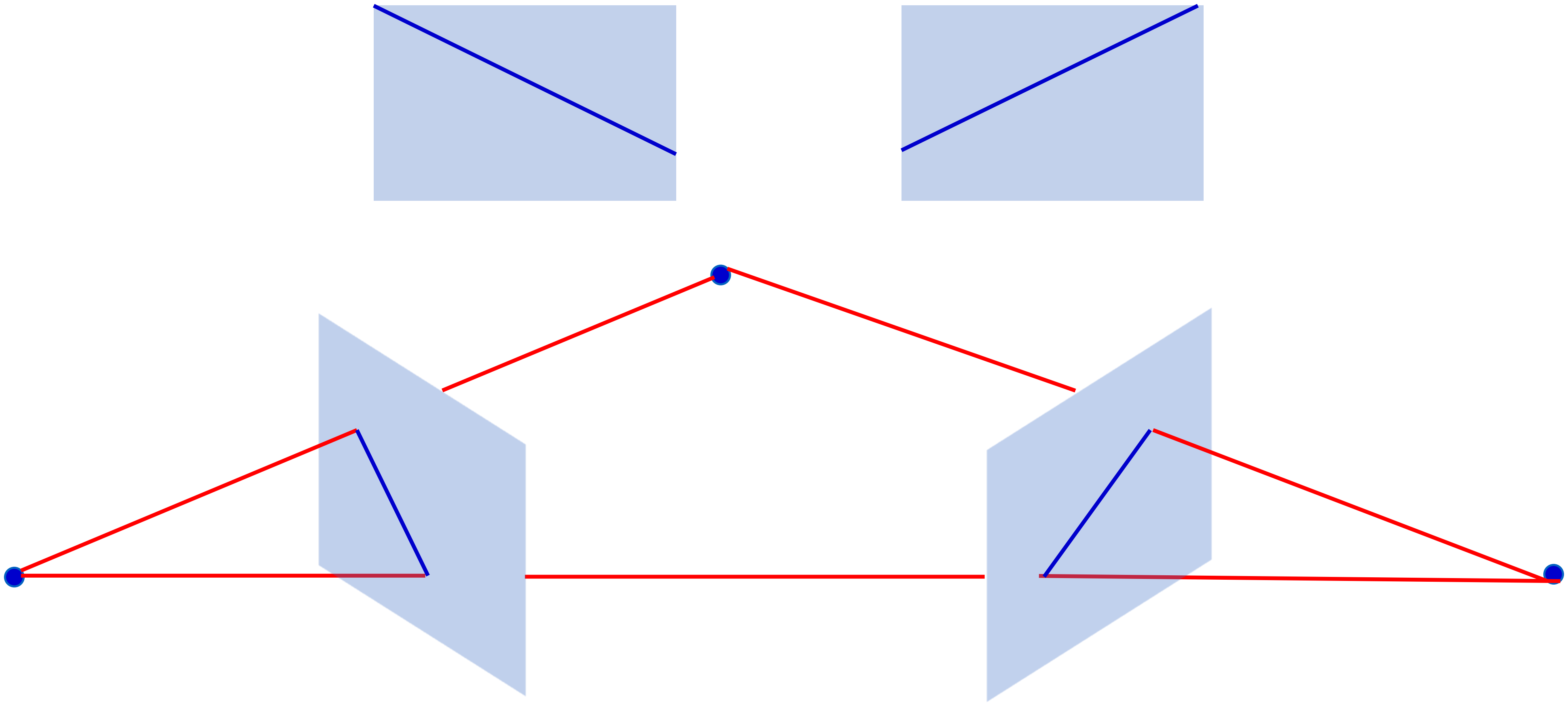
Epipolar geometry

- Parallax and gradient of epipolar lines
  - The corresponding points in the right image on the epipolar line in the left image are on the epipolar line.



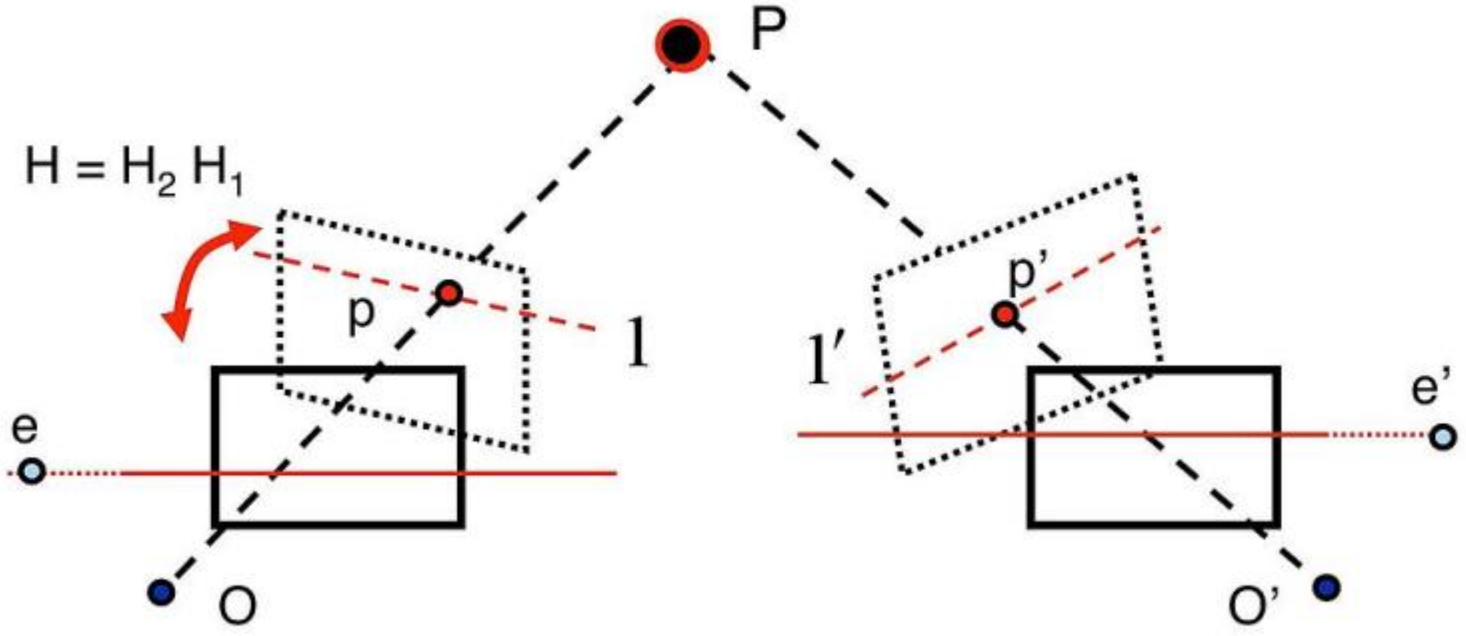
Epipolar geometry

- Parallax and gradient of epipolar lines
  - How do I find the epipolar line for every pixel?



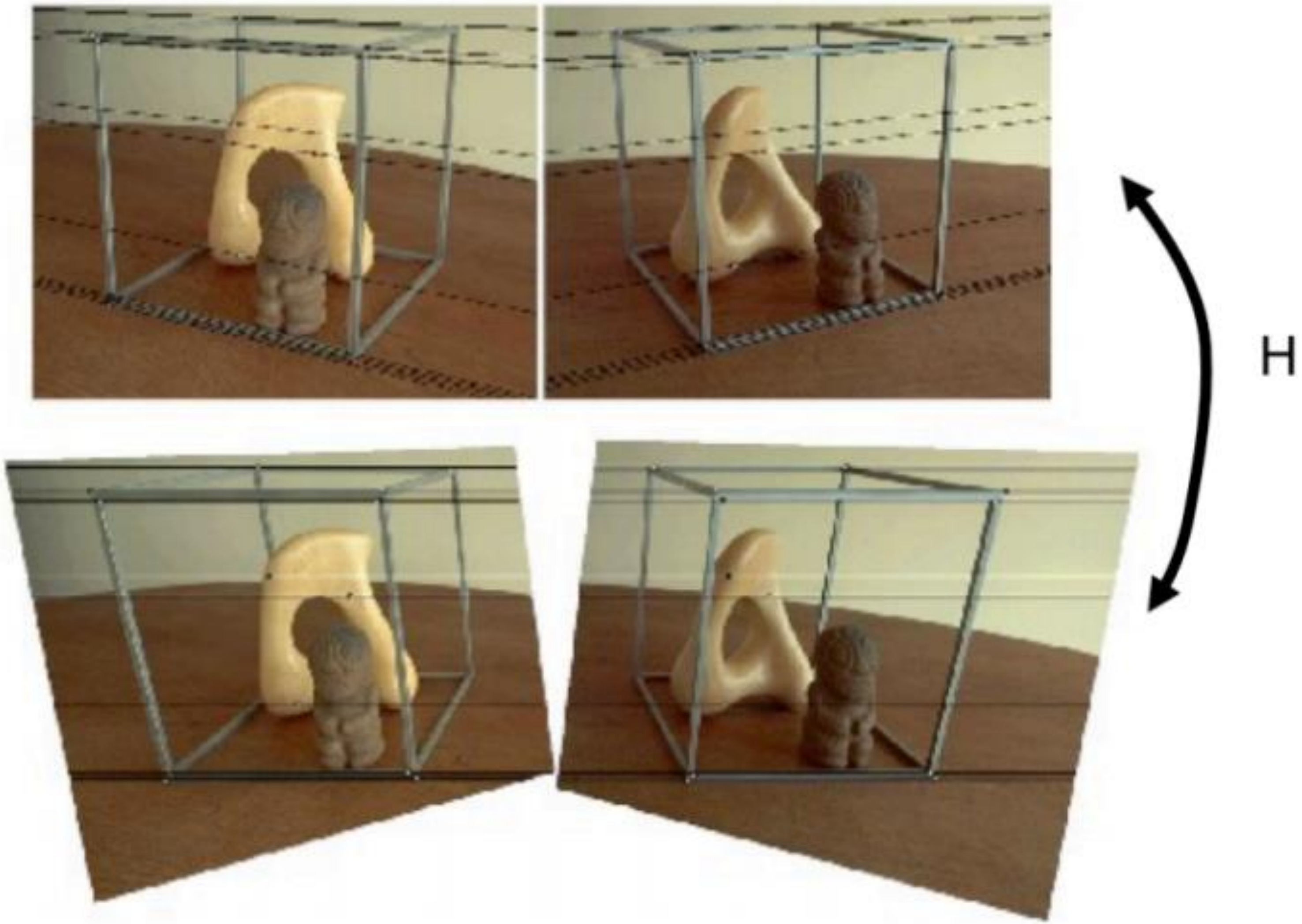
Rectification

- Rectification
  - Find each homography that move <u>epipole to the infinity</u>
  - This is equivalent to parallelizing the image sensor to the baseline.



Stereo matching

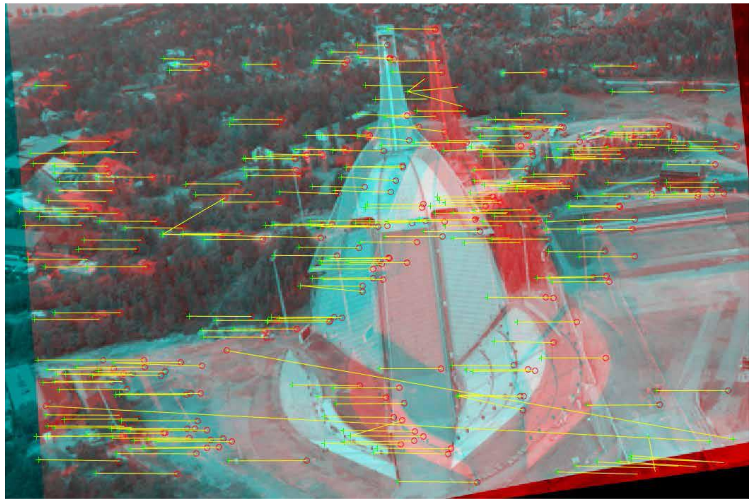
- Capture images
- Rectification
- L-R Matching
- Depth map



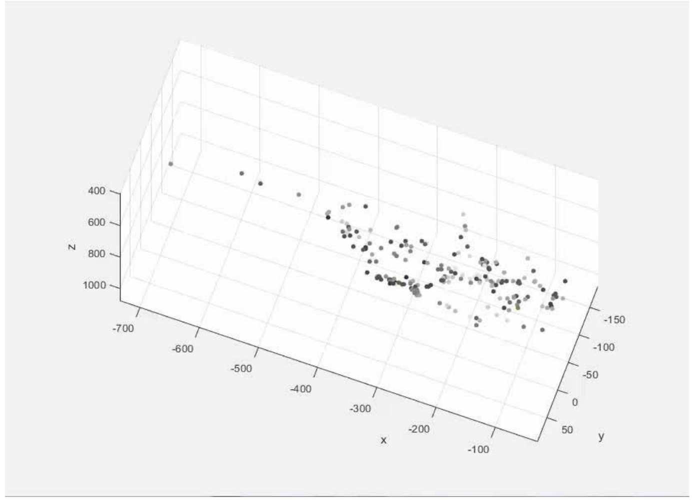
Stereo matching

- Sparse stereo
  - Extract feature points (corner, DOG, etc.)
  - Match feature points along the same row
  - Compute 3D from disparity

L, R images



Feature matching & disparity(line)



3D representation

Stereo matching

- Dense matching
  - For a patch in the left image
  - Compare with patches along the same row in the right image → patch matching
  - Matching method : template matching (SSD, SAD, NCC)

Review: template matching

- Sum of Absolute Differences (SAD)

$$SAD(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \left| I(i+m, j+n) - T(m,n) \right|$$

- Sum of Squared Differences (SSD)

$$SSD(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \left(I(i+m, j+n) - T(m,n) \right)^2$$

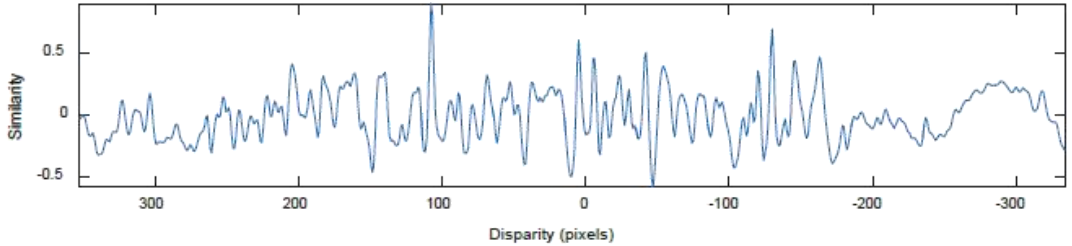
- Normalized Cross Correlation (NCC)

$$NCC(i,j) = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(i+m, j+n) \cdot T(m,n)}{\left(\sqrt{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(i+m, j+n)^2}\right) \cdot \left(\sqrt{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} T(m,n)^2}\right)}$$

Stereo matching

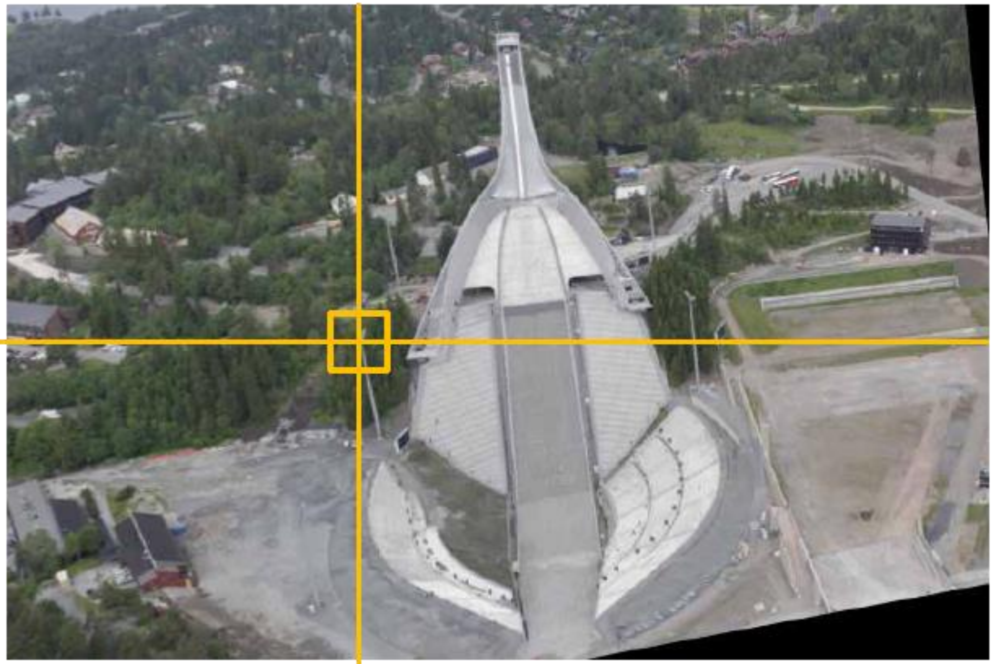
• Dense matching

- For a patch in the left image
  - Compare with patches along the same row in the right image

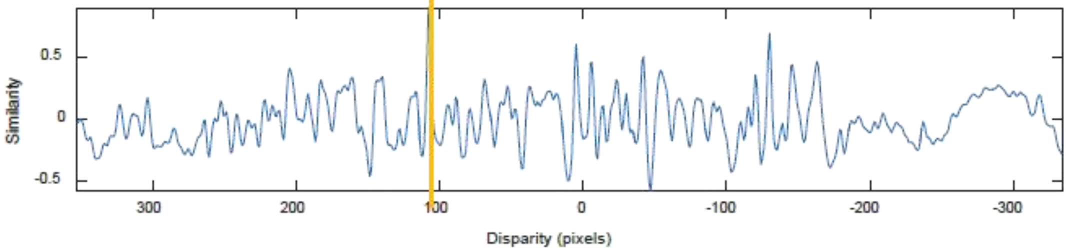


Stereo matching

- Dense matching (Repeat for all pixels in the left image)



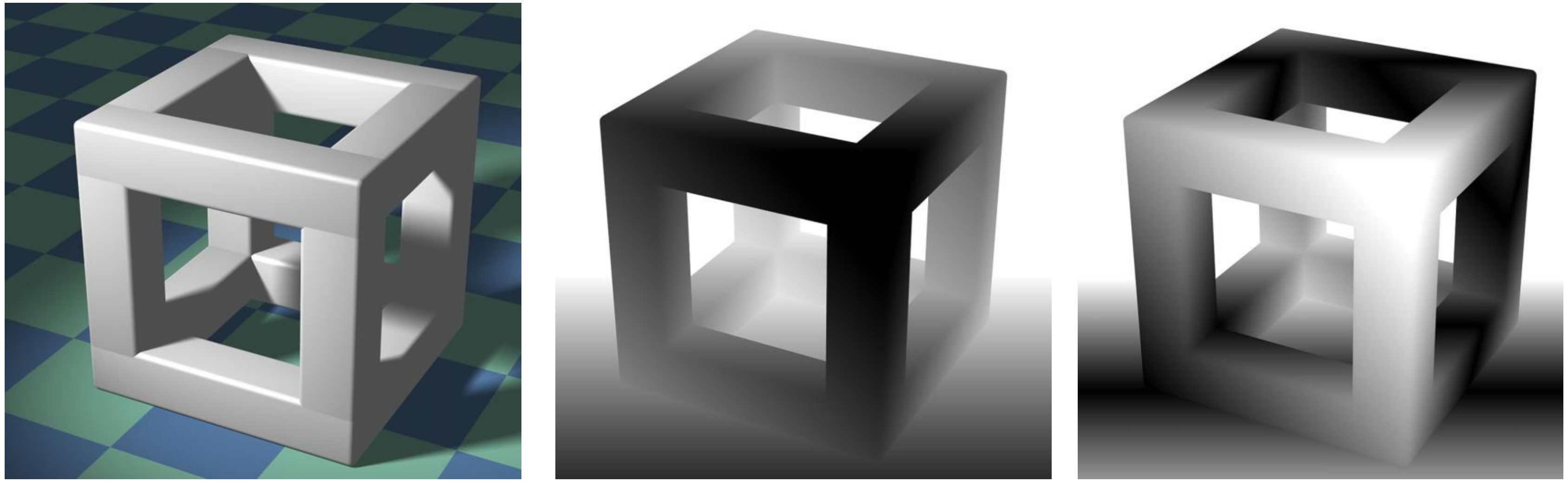
- For a patch in the left image
  - Compare with patches along the same row in the right image
  - Select patch with highest score



Depth representation

- Depth map (depth image)
  - an image or image channel that contains information relating to the distance of the surfaces of scene objects from a viewpoint.
  - Single channel image, of which pixel values are
    - Distance
    - Disparity

$$disparity = x - x' = \frac{B \cdot f}{z}$$



3D Depth (distance) Disparity (inverse distance)

Depth representation

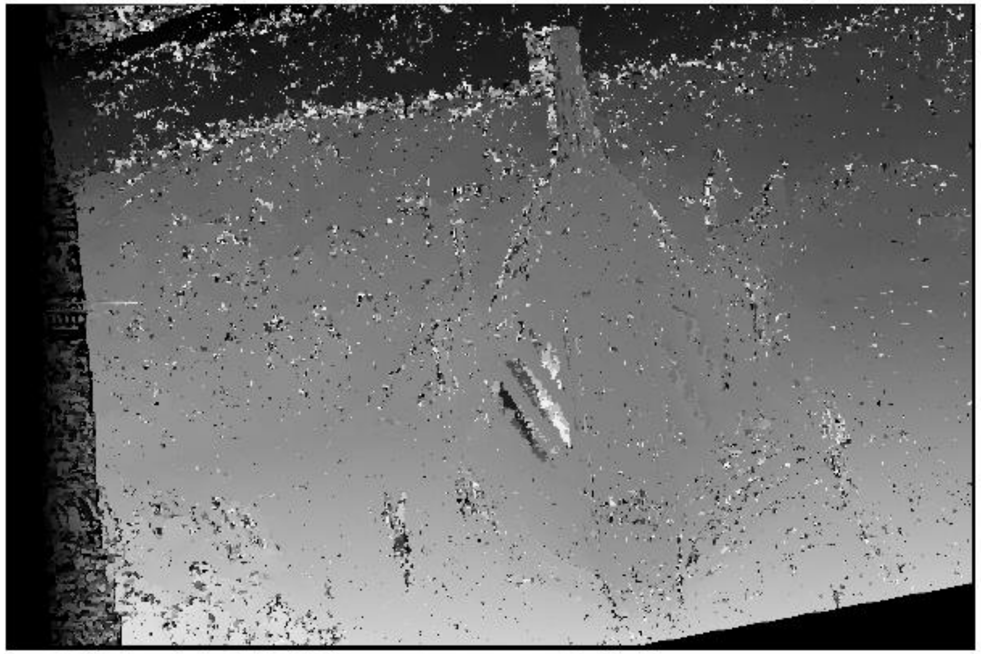
- Depth map
  - Each pixel value represents a distance d .
    - The distance can be derived from disparity
  - Possible to compute ray direction of each pixel
  - Find k that makes zk equal to d
    - Set $k[x, y, z]^T$ in 3D

(a) Real-image

(b) Depth-map

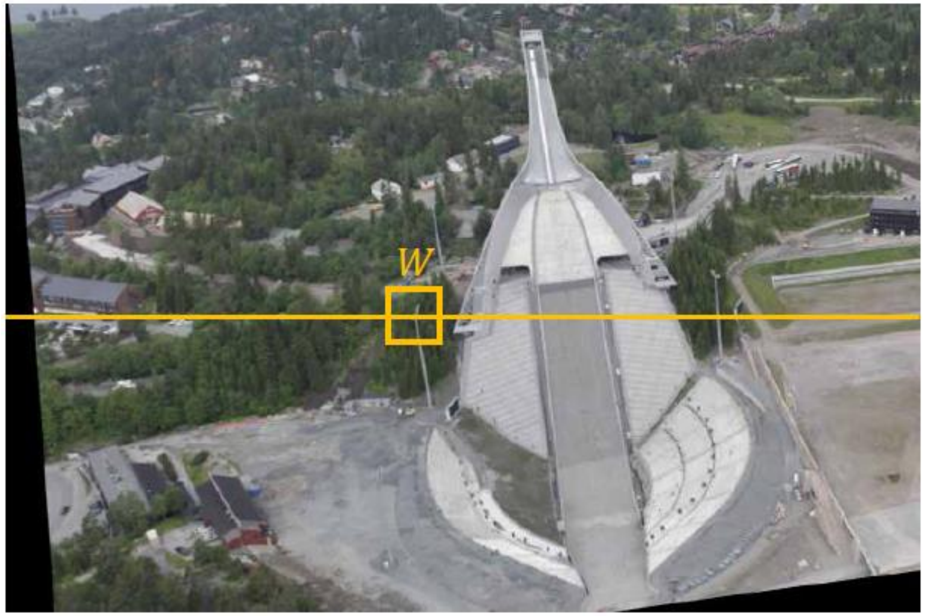
Depth representation

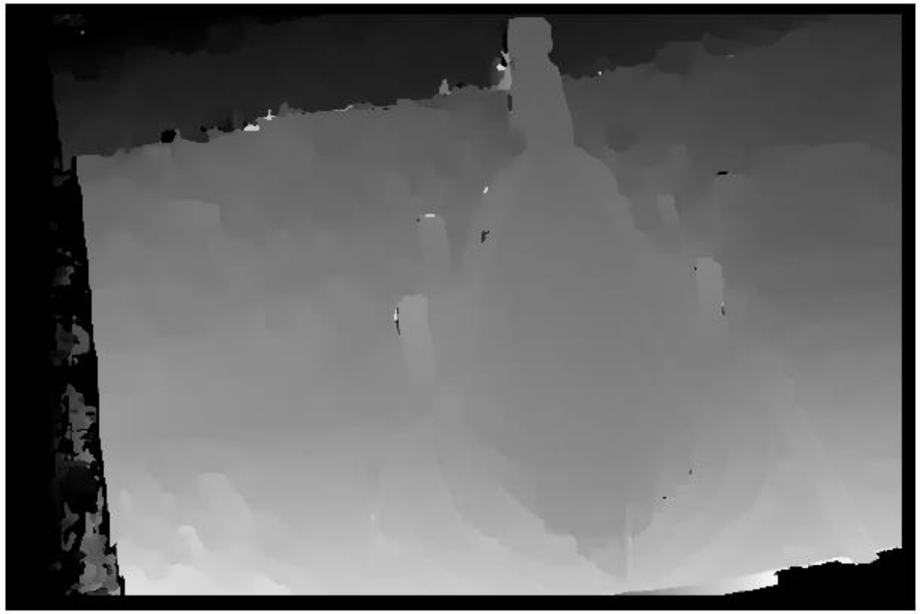
- Depth map with small size of window
  - More difficult to match
  - Noisy
  - Fine structure



Depth representation

- Depth map with large size of window
  - Loss of detail
  - Smooth
  - slow





3D reconstruction

- 3D reconstruction using stereo images
  - Rectification
  - Get disparity map
  - 3D back projection with Left(or right) image and disparity map
    - $d = D(u, v)$ when D is depth map
    - Or
    - $d = \frac{Bf}{D(u,v)}$ when D is disparity map
    - $k\begin{bmatrix} x \\ y \\ z \end{bmatrix} = K^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$ for each pixel
    - Compute k that makes kz equals to d
    - Set $k\begin{bmatrix} x \\ y \\ z \end{bmatrix}$ in 3D space

Thank you