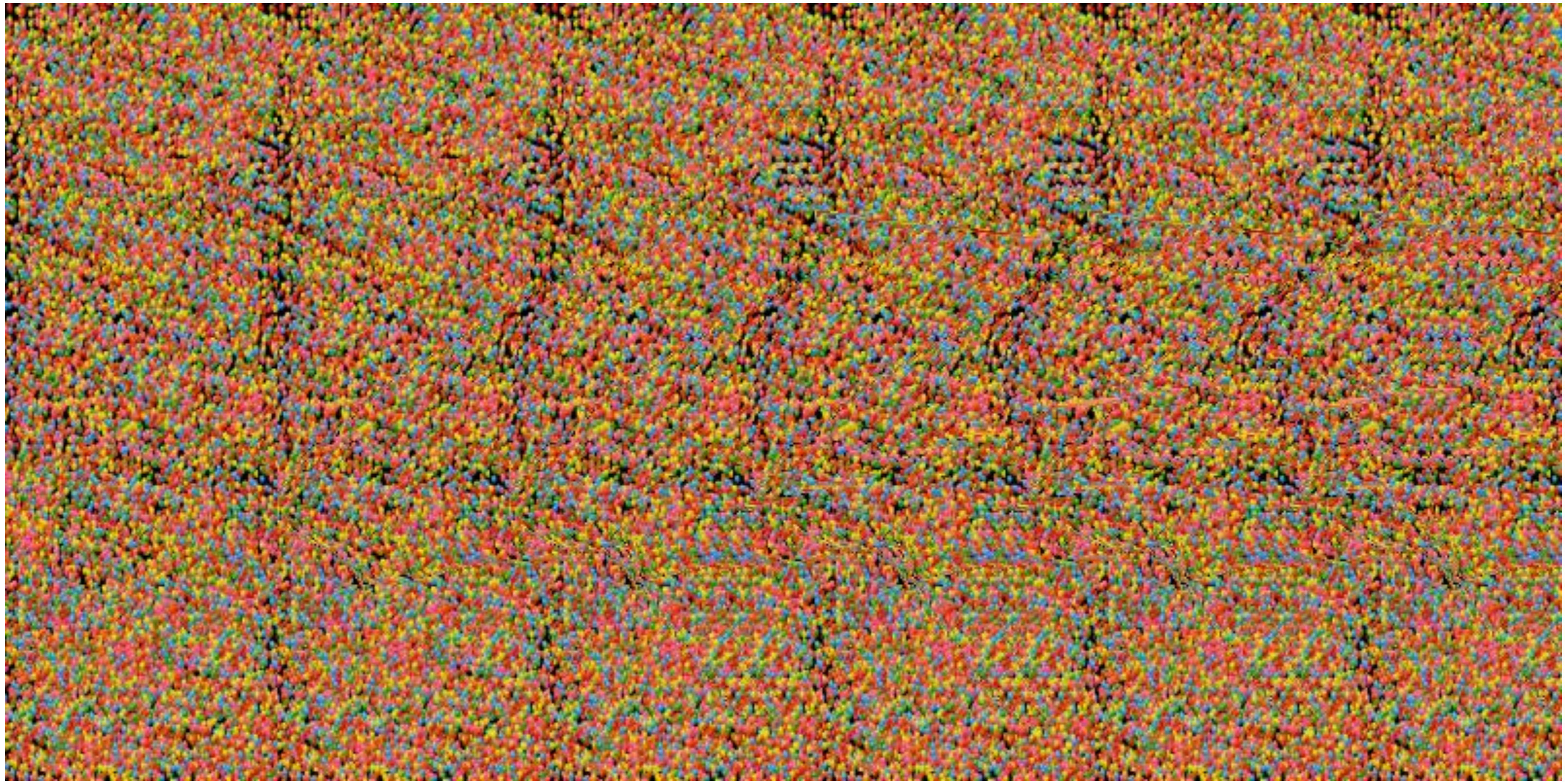
Computer Vision

Binocular Stereo

What is this?



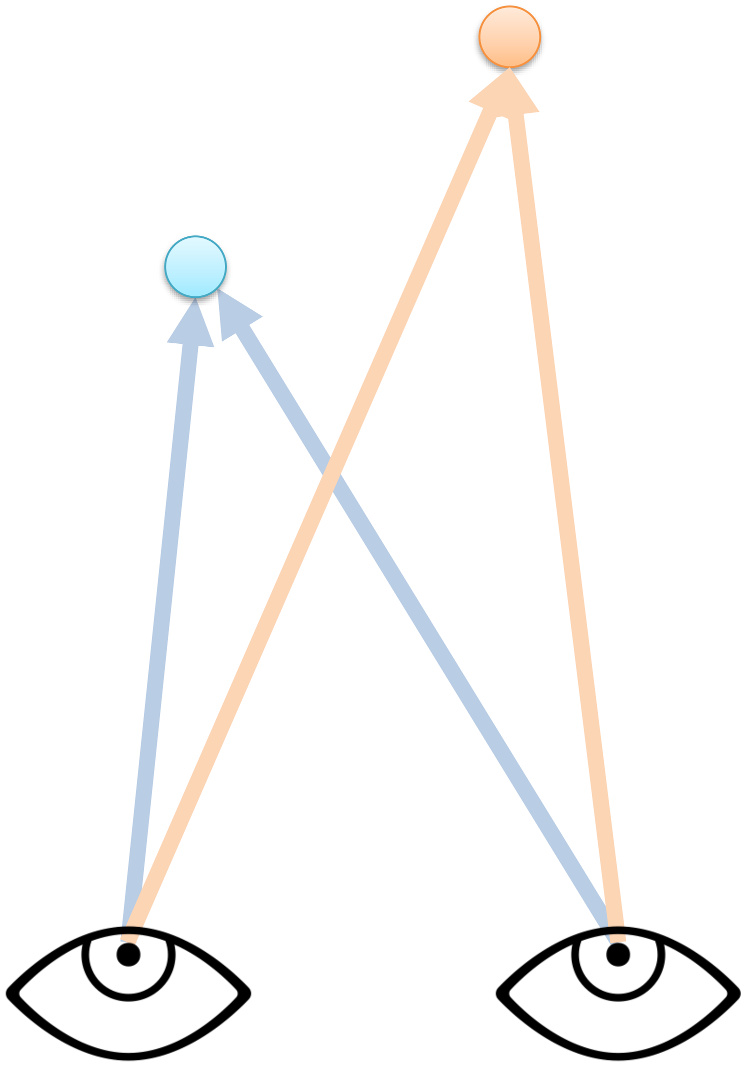
Single image stereogram,
https://en.wikipedia.org/wiki/Autostereogram

“Mark Twain at Pool Table", no date, UCR Museum of Photography

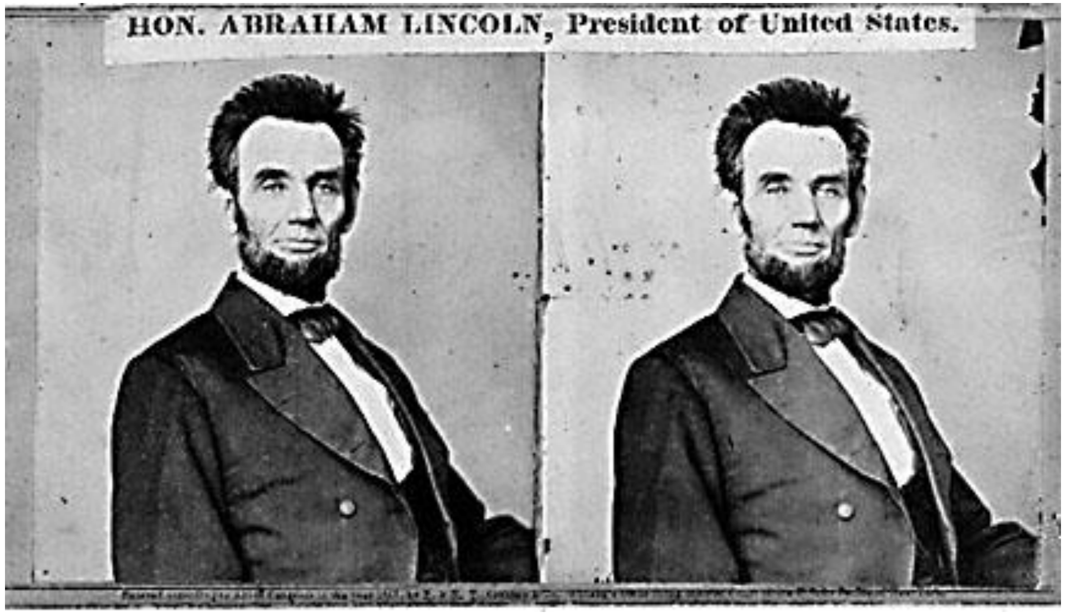
https://giphy.com/gifs/wigglegram-706pNfSKyaDug

Stereo Vision as Localizing Points in 3D

- An object point will project to some point in our image
- That image point corresponds to a ray in the world
- Two rays intersect at a single point, so if we want to localize points in 3D we need 2 eyes

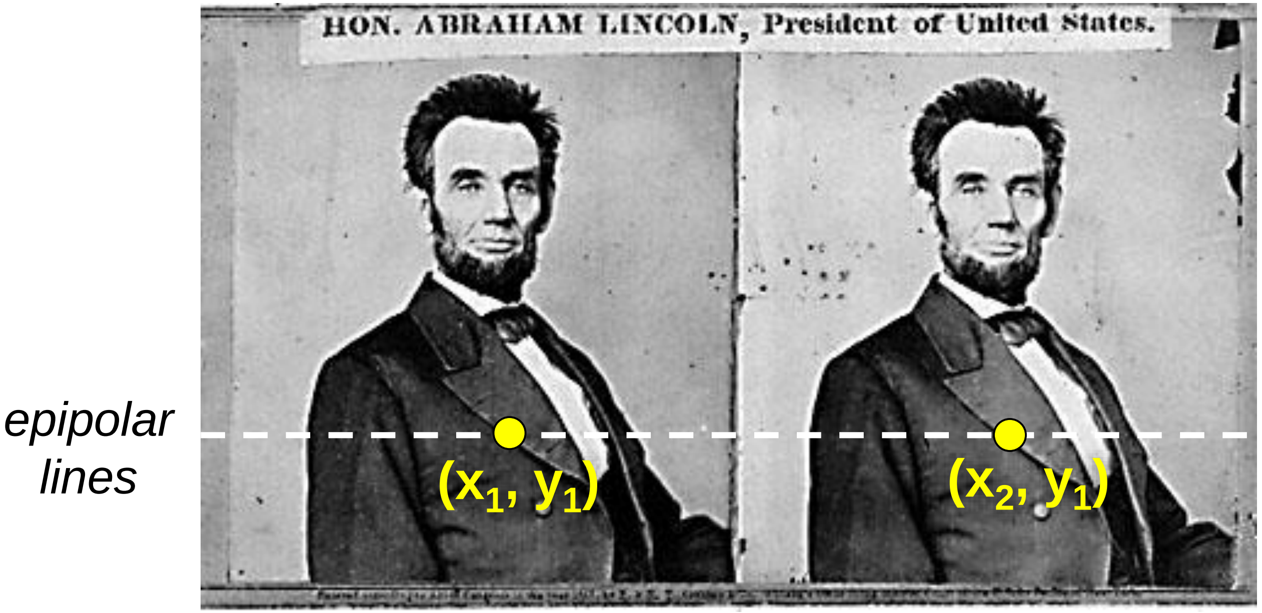


Stereo



- Given two images from different viewpoints
  - How can we compute the depth of each point in the image?
  - Based on *how much each pixel moves* between the two images

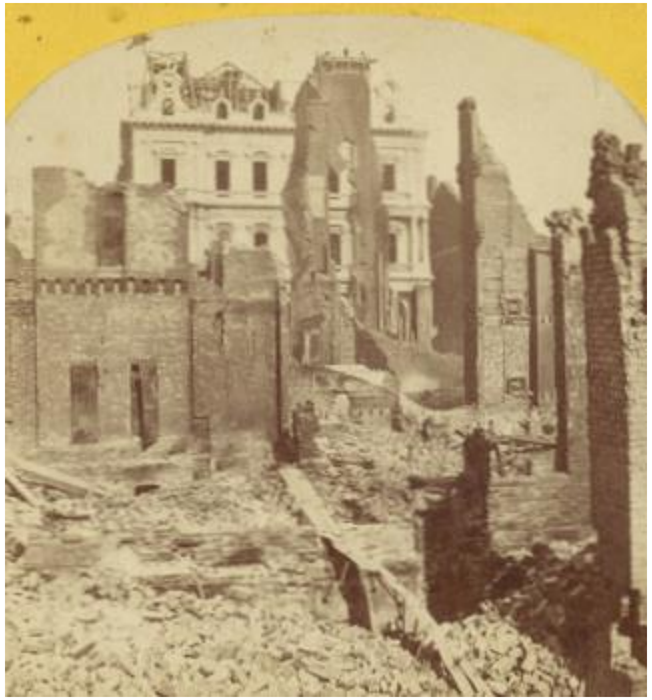
Epipolar geometry



Two images captured by a purely horizontal translating camera
(*rectified* stereo pair)

$x_2 - x_1$ = the *disparity* of pixel (x_1, y_1)

Disparity = inverse depth



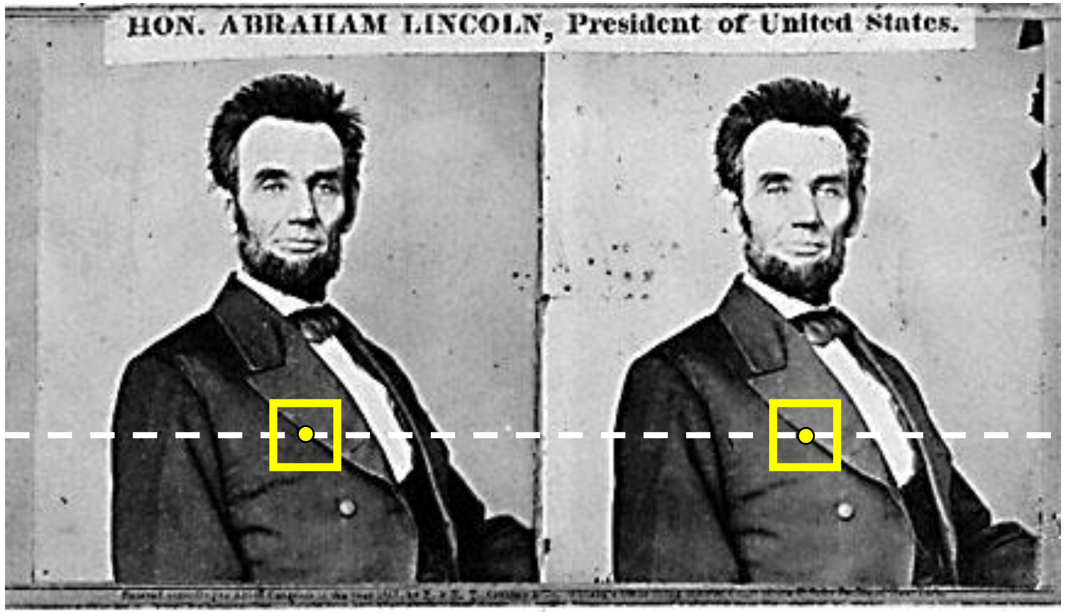
http://stereo.nypl.org/view/41729

(Or, hold a finger in front of your face and wink each eye in succession.)

Your basic stereo matching algorithm

- **Match Pixels in Conjugate Epipolar Lines**
  - Assume brightness constancy
  - This is a challenging problem
  - Hundreds of approaches
    - A good survey and evaluation: http://www.middlebury.edu/stereo/

Your basic stereo matching algorithm



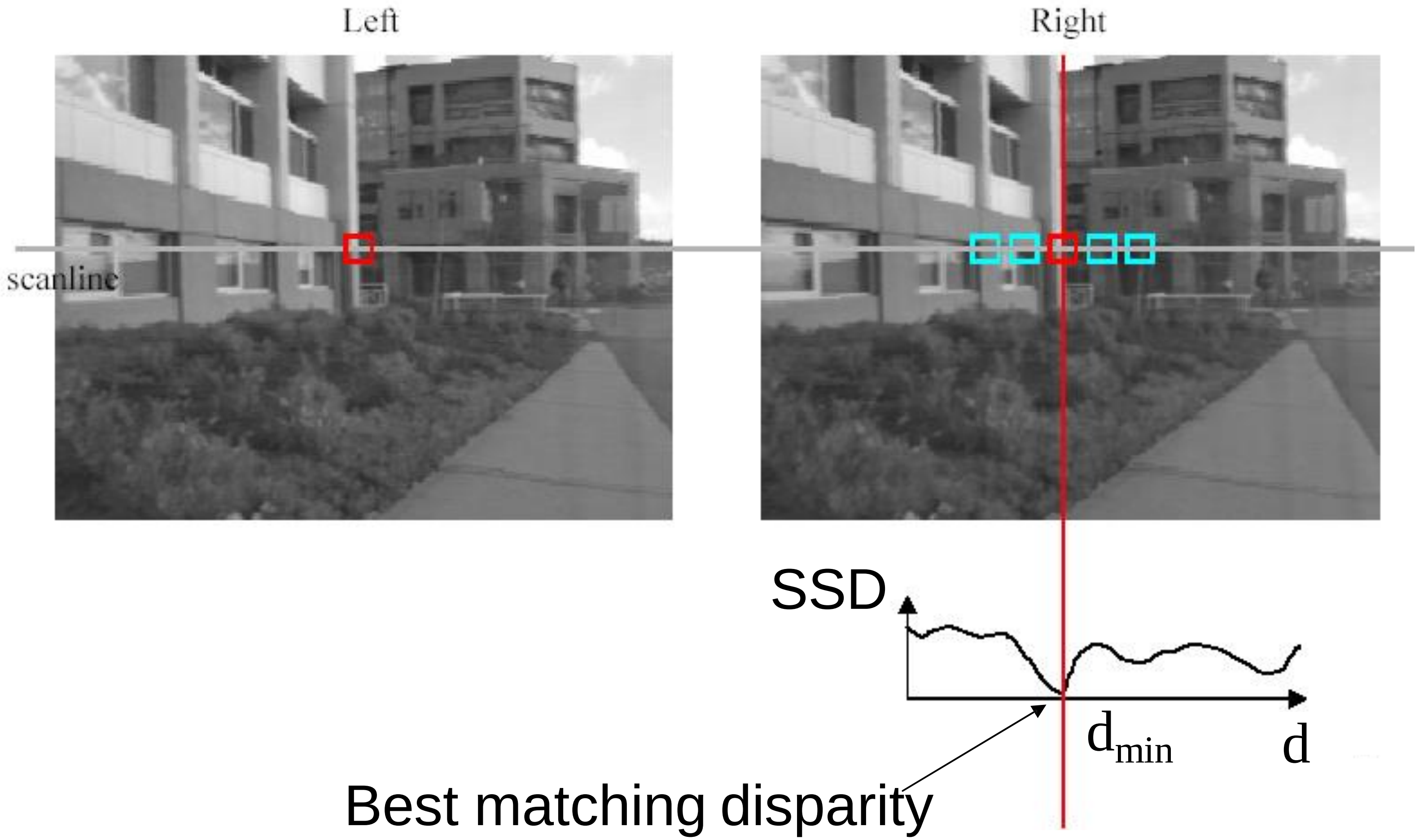
For each epipolar line

For each pixel in the left image

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

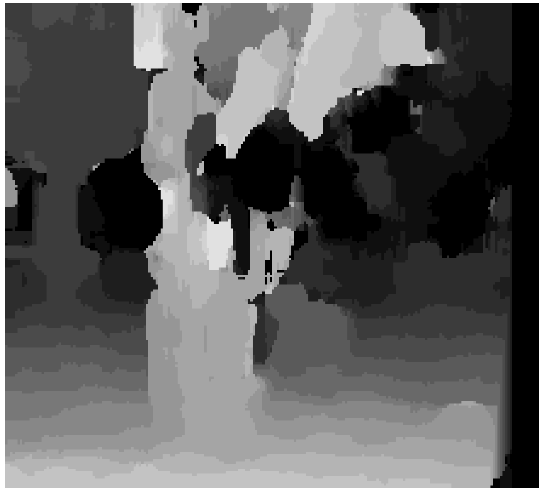
Improvement: match ***windows***

Stereo matching based on SSD



Window size

W = 3



W = 20

Effect of window size

- Smaller window
  - \+ more detail
  - \- more noise
- Larger window
  - \+ less noise
  - \- less detail

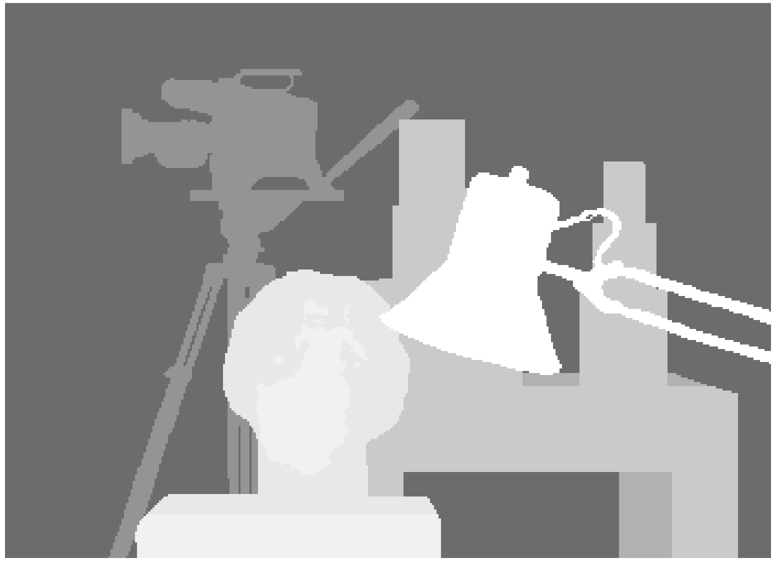
Better results with *adaptive window*

- T. Kanade and M. Okutomi, *A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment*, ICRA 1991.
- D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. IJCV, July 1998

Stereo results

– Data from University of Tsukuba

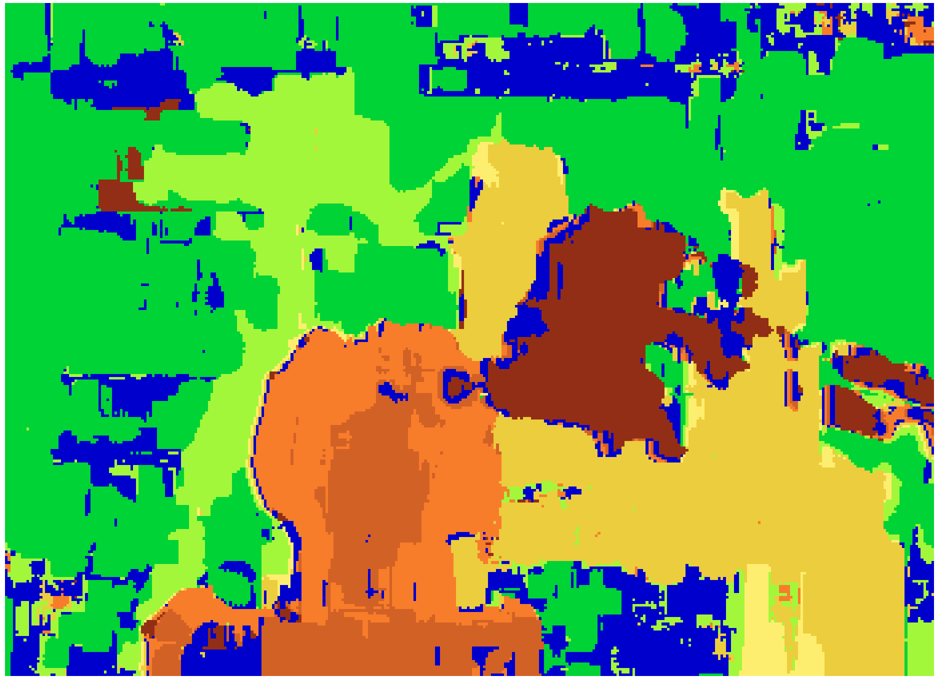
– Similar results on other images without ground truth



Scene

Ground truth

Results with window search



Window-based matching
(best window size)

Ground truth

Better methods exist...

Graph cuts-based method

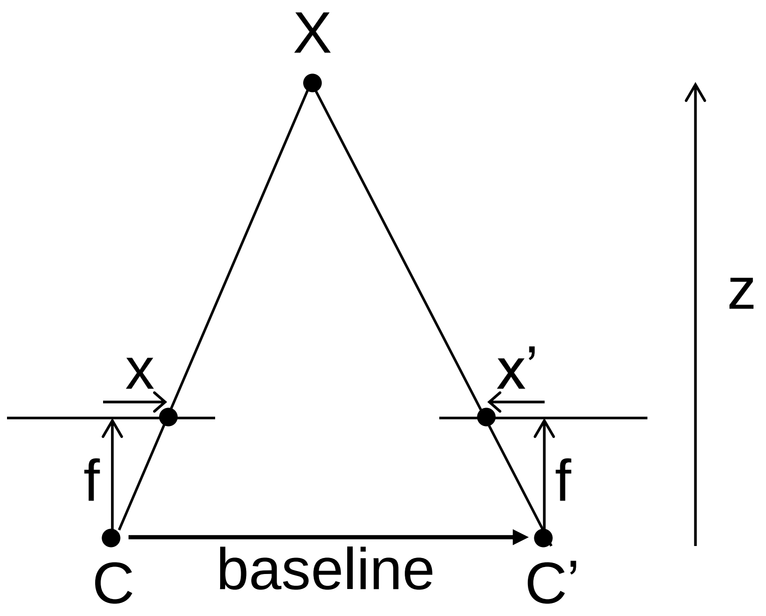
Ground truth

Boykov et al., [Fast Approximate Energy Minimization via Graph Cuts](), International Conference on Computer Vision 1999.

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

Questions?

Depth from disparity



$$disparity = x - x' = \frac{baseline * f}{z}$$

Stereo reconstruction pipeline

- Steps
  - **Calibrate cameras**
  - **Rectify images**
  - **Compute disparity**
  - **Estimate depth**

What will cause errors?

- **Camera calibration errors**
- **Poor image resolution**
- **Occlusions**
- **Violations of brightness constancy (specular reflections)**
- **Large motions**
- **Low-contrast image regions**

Variants of stereo

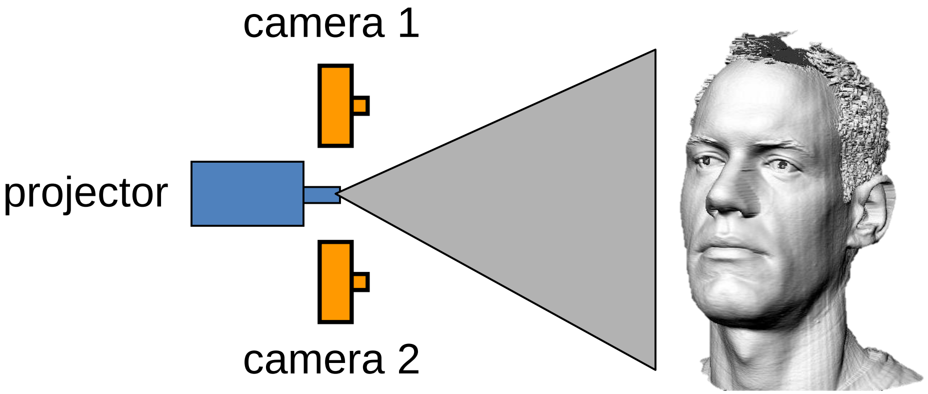
Real-time stereo

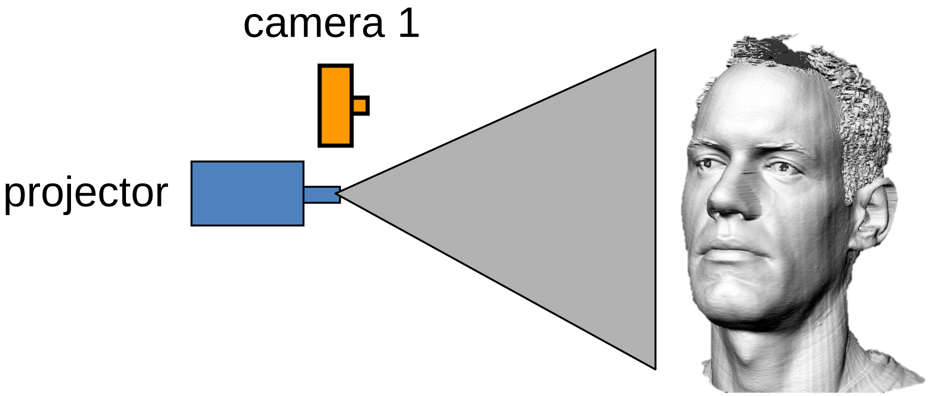
Nomad robot searches for meteorites in Antartica

- Used for robot navigation (and other tasks)
  - Several real-time stereo techniques have been developed (most based on simple discrete search)

Active stereo with structured light


Li Zhang's one-shot stereo



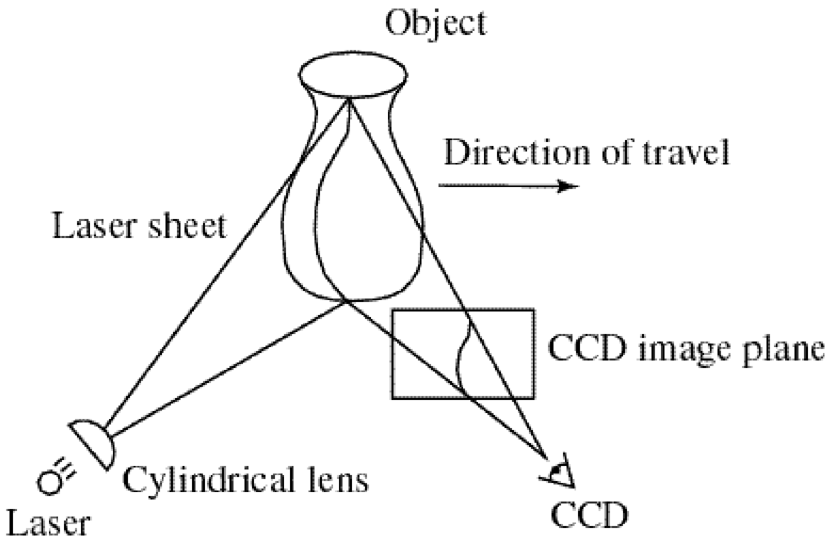


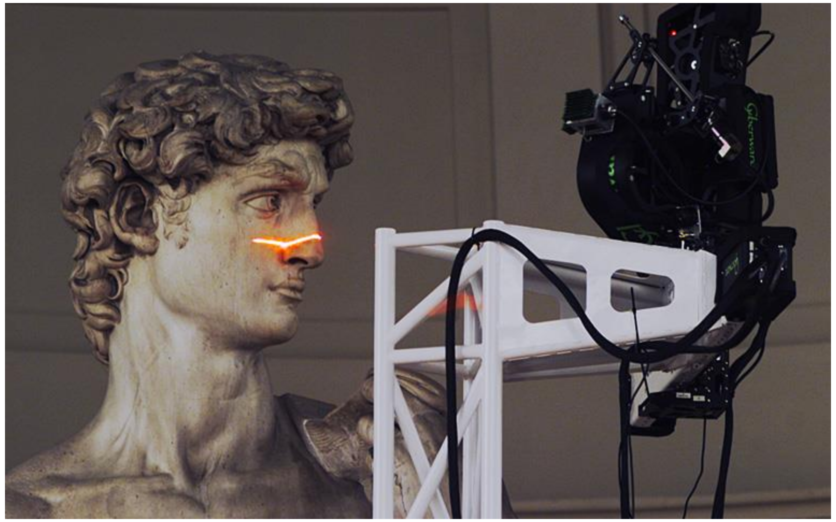
- Project "structured" light patterns onto the object
  - simplifies the correspondence problem
  - basis for active depth sensors, such as Kinect and iPhone X (using IR)

Active stereo with structured light

https://ios.gadgethacks.com/news/watch-iphone-xs-30k-ir-dots-scan-your-face-0180944/

Laser scanning





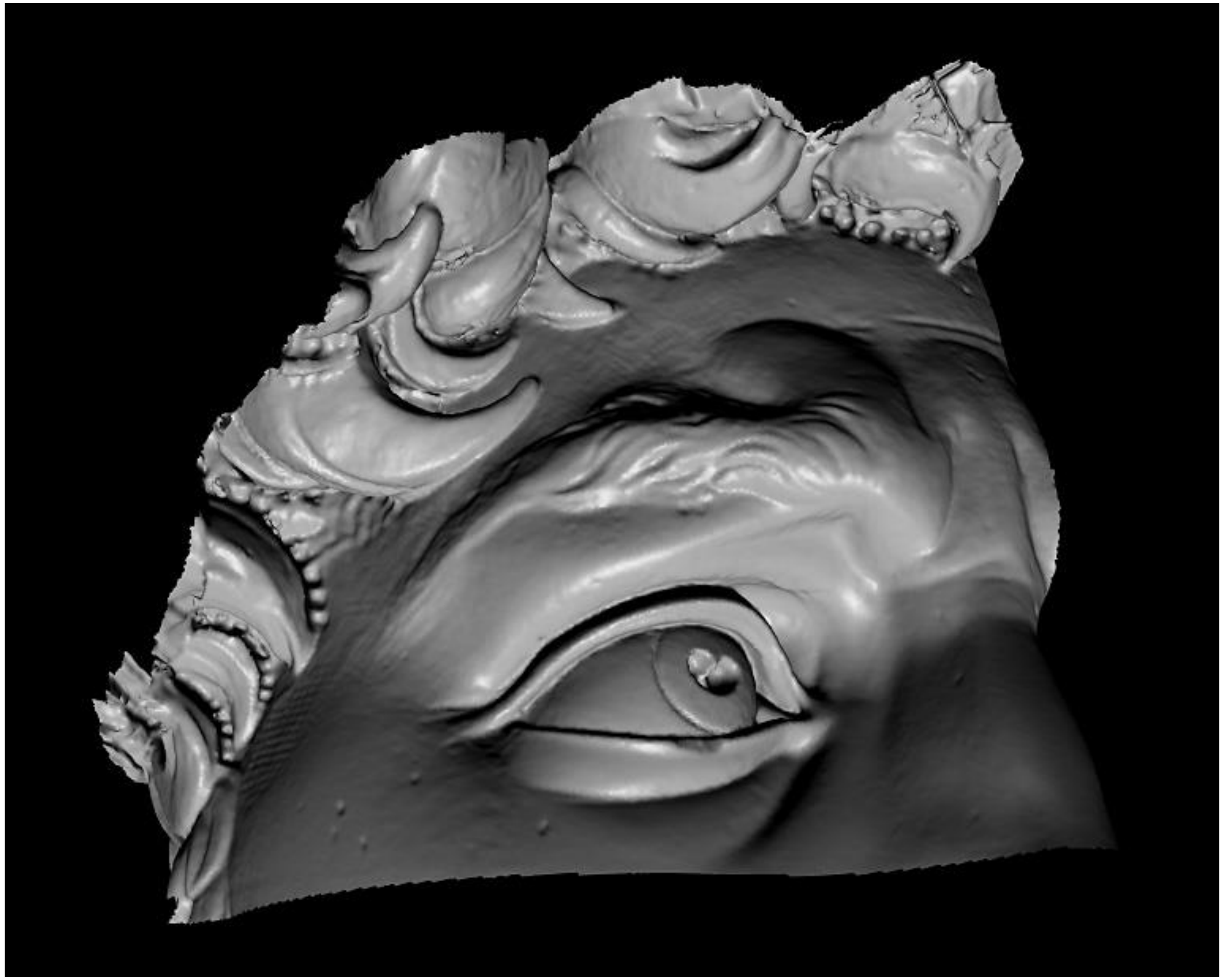
Digital Michelangelo Project
http://graphics.stanford.edu/projects/mich/

- Optical triangulation
  - Project a single stripe of laser light
  - Scan it across the surface of the object
  - This is a very precise version of structured light scanning

Laser scanned models

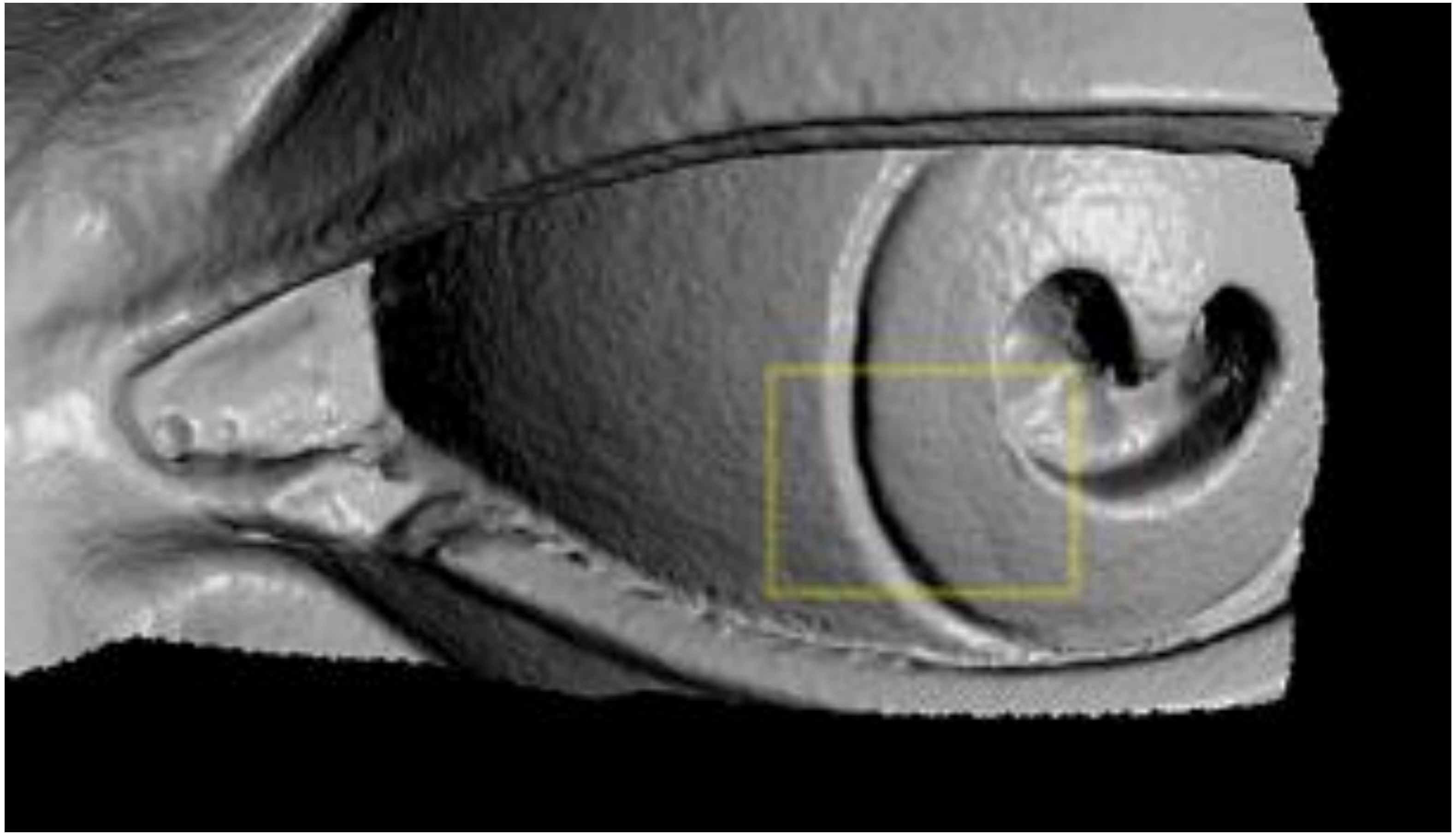
The Digital Michelangelo Project, Levoy et al.

Laser scanned models



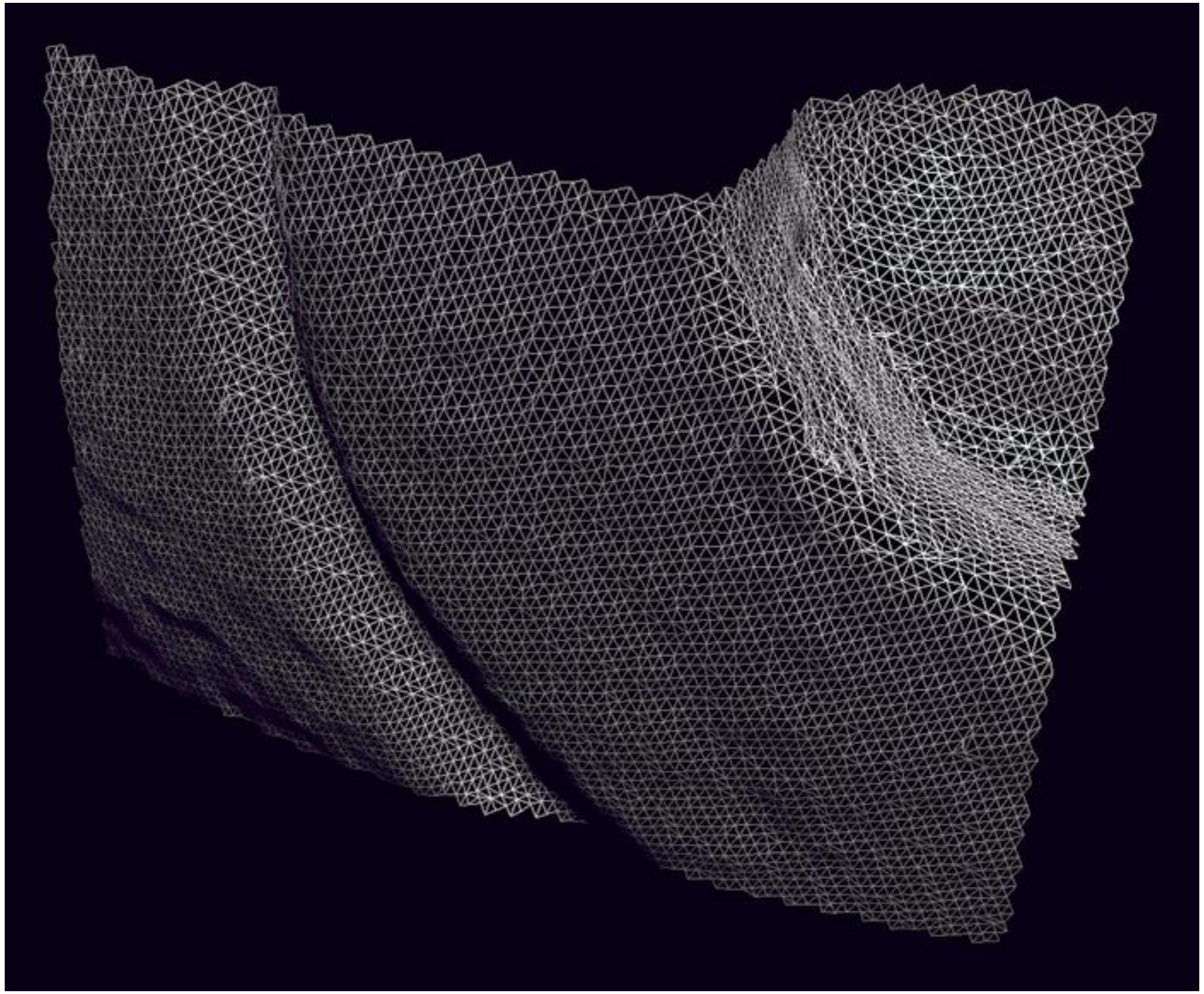
The Digital Michelangelo Project, Levoy et al.

Laser scanned models



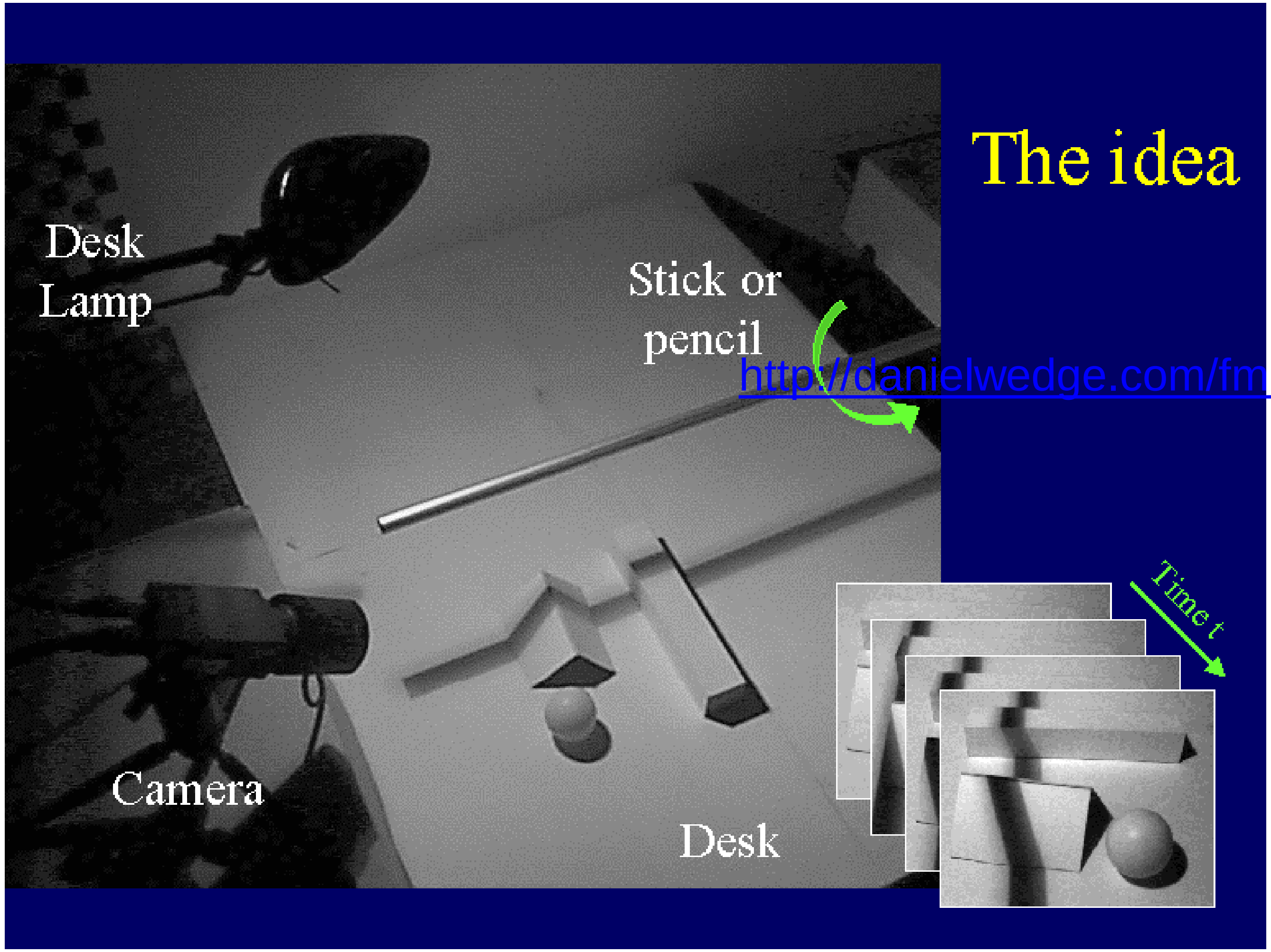
The Digital Michelangelo Project, Levoy et al.

Laser scanned models

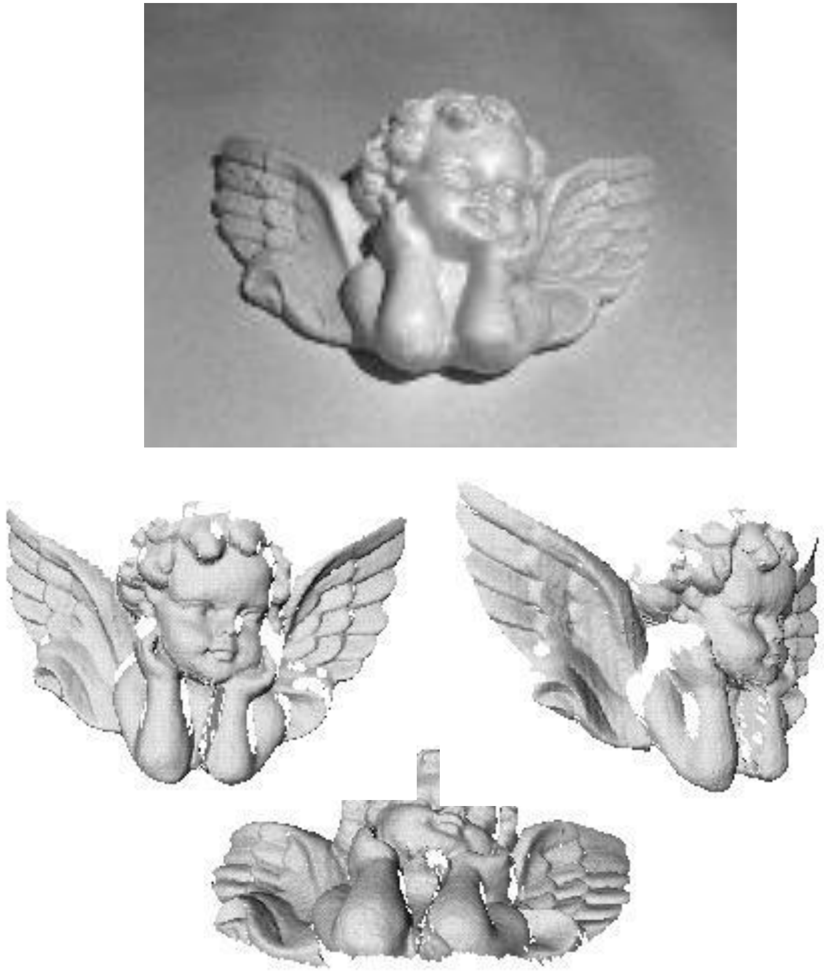


The Digital Michelangelo Project, Levoy et al.

3D Photography on your Desk



http://danielwedge.com/fmatrix/

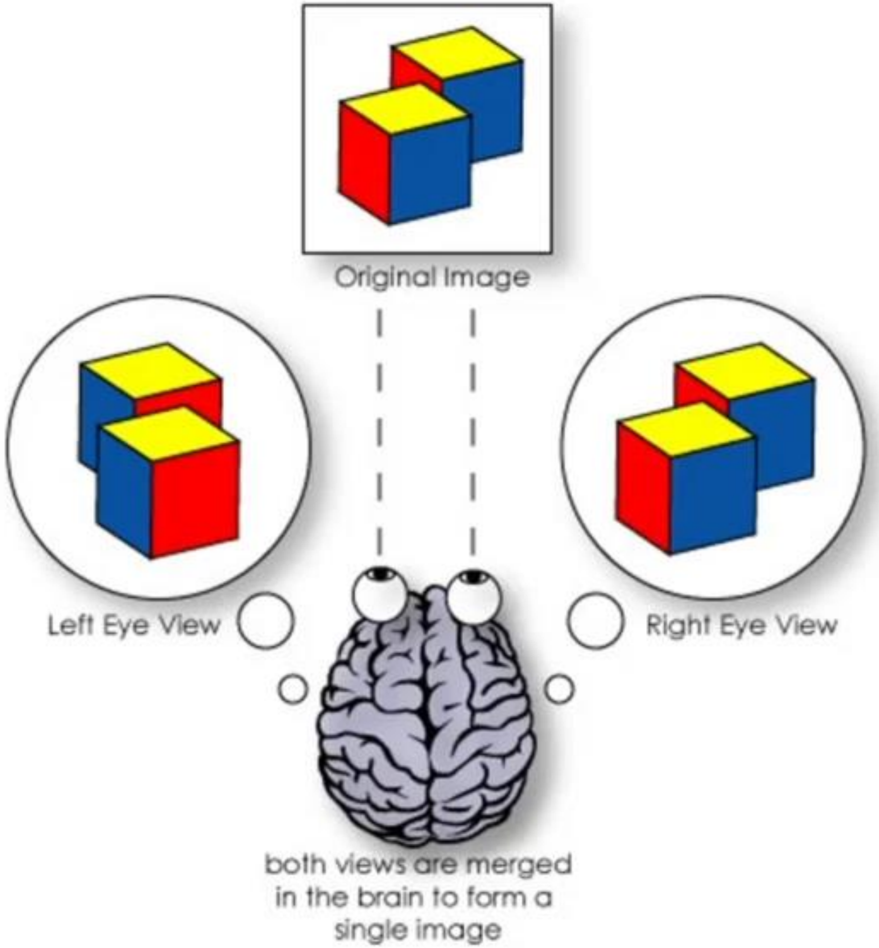


http://www.vision.caltech.edu/bouguetj/ICCV98/

Questions?

Human Has Stereo Vision!

- Our pair of eyes gives us the ability to sense depth.



Two-View Stereo

- The goal is to get dense points in 3D from two image pairs with known baseline $(\mathrm{R}, \mathbf{t})$.

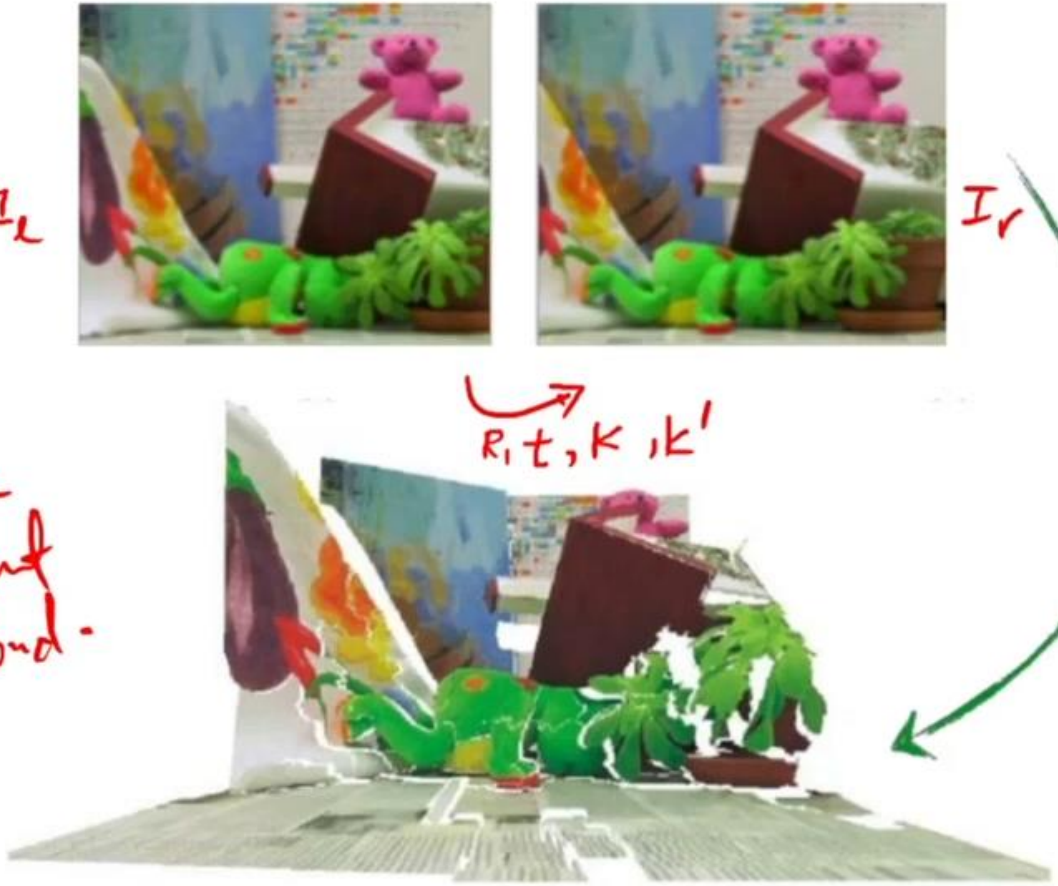


Image source: http://slazebni.cs.illinois.edu/spring19/lec18_stereo.pdf

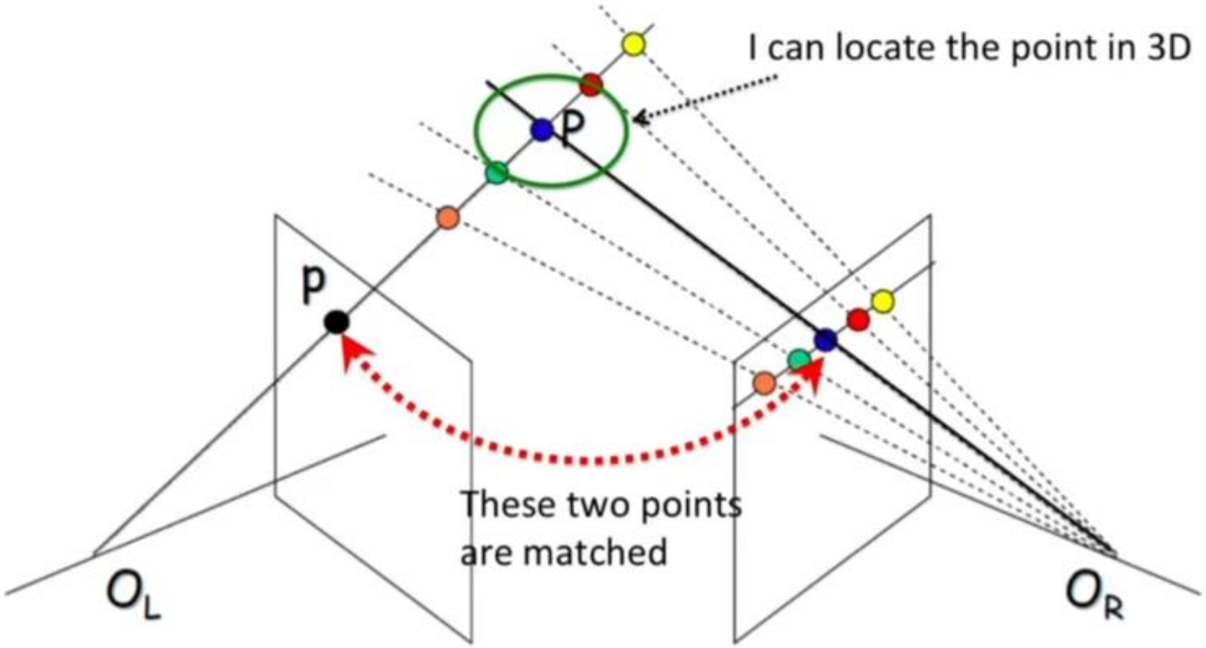
Stereograms

- This is how 3D movies are made!

Image source: http://slazebni.cs.illinois.edu/spring19/lec18_stereo.pdf

Two-View Stereo

- What cues tell us about scene depth?



Epipolar geometry and triangulation from image correspondences give us depth!

Image source: http://www.cs.toronto.edu/~fidler/slides/2015/CSC420/lecture12_hres.pdf

Two-View Stereo: Problem formulation

- **Given:** Two cameras with known baseline $(\mathrm{R}, \mathbf{t})$ and rigidly fixed onto a rig.
- **Find:** The depth map, which gives the dense 3D points of the scene.

image 1

image 2

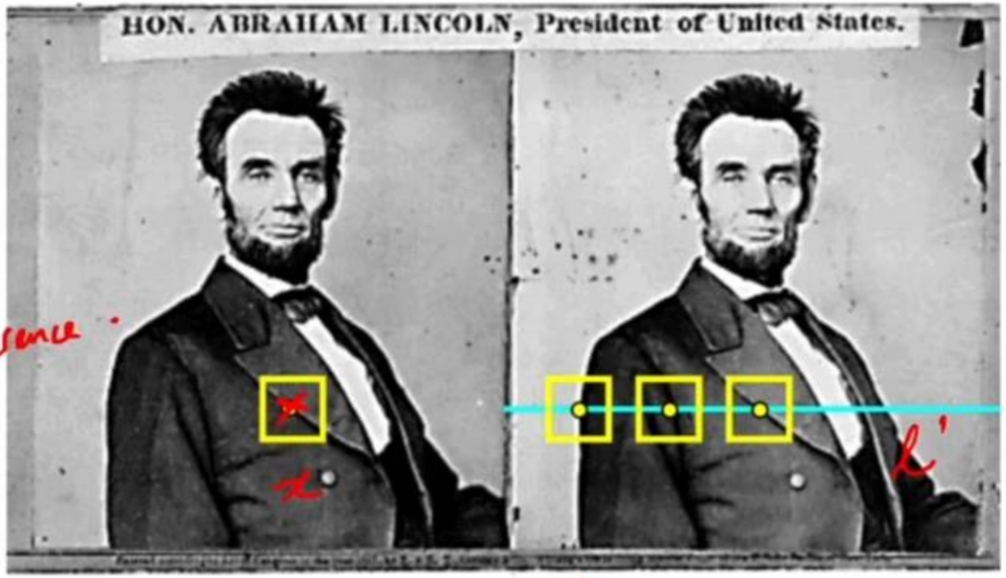
Dense depth map

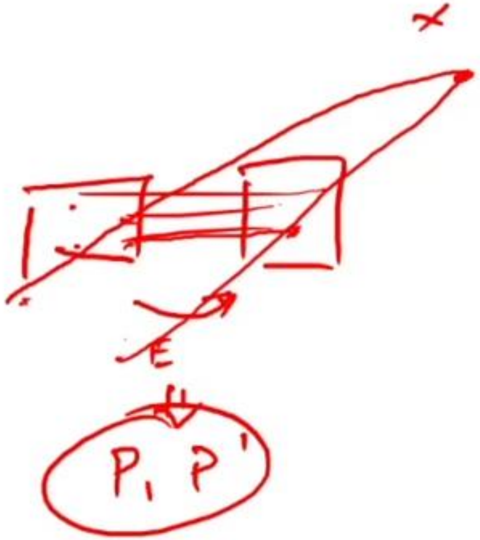
Bumblebee®2 FireWire

Image source: https://www.flir.com/support/products/bumblebee2-firewire#Overview

Basic Stereo Matching Algorithm

Reference





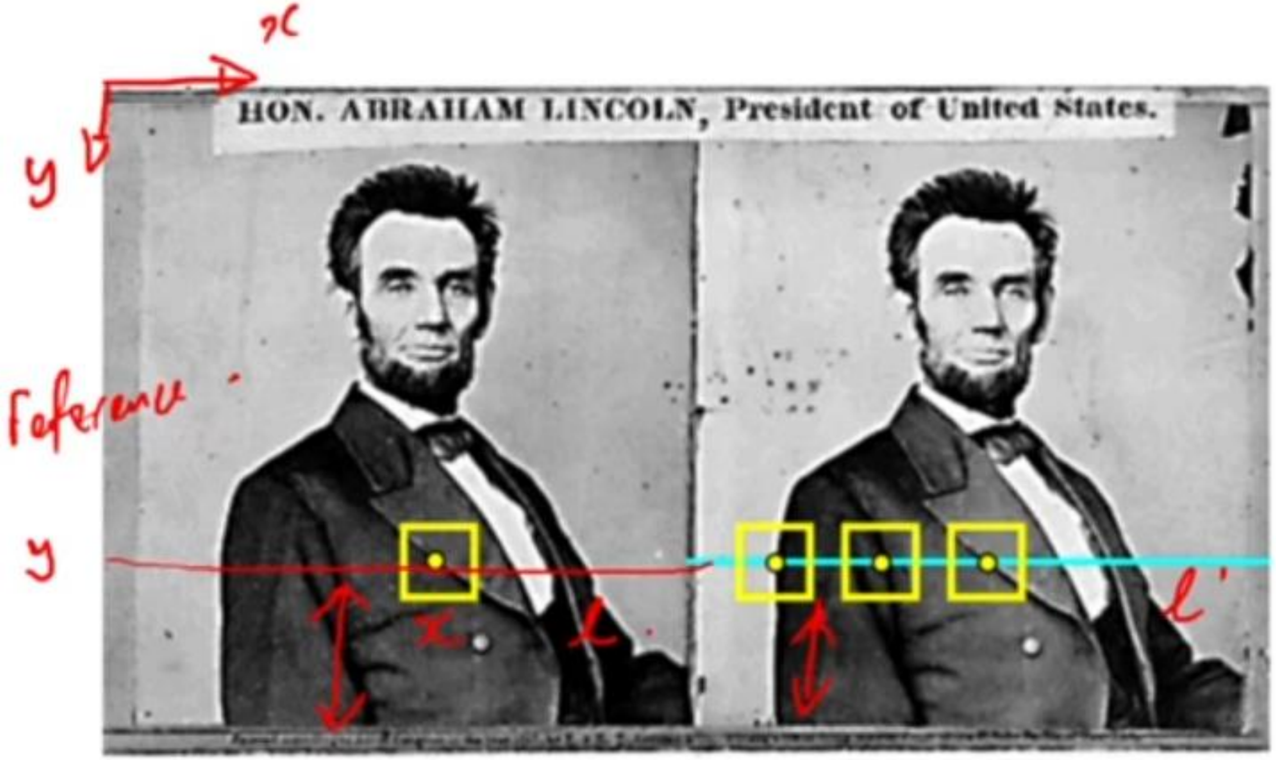
For each pixel in the first image:

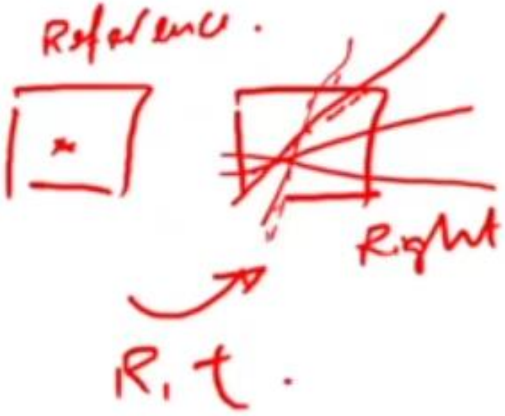
$K, K' : \underbrace{R, t}_{E} \Rightarrow F \Rightarrow l' = Fx$

1. Find corresponding epipolar line in the right image.
2. Examine all pixels on the epipolar line and pick the best match.
3. Triangulate the matches to get depth information.

Image source: http://slazebni.cs.illinois.edu/spring18/lec18_stereo.pdf

Basic Stereo Matching Algorithm

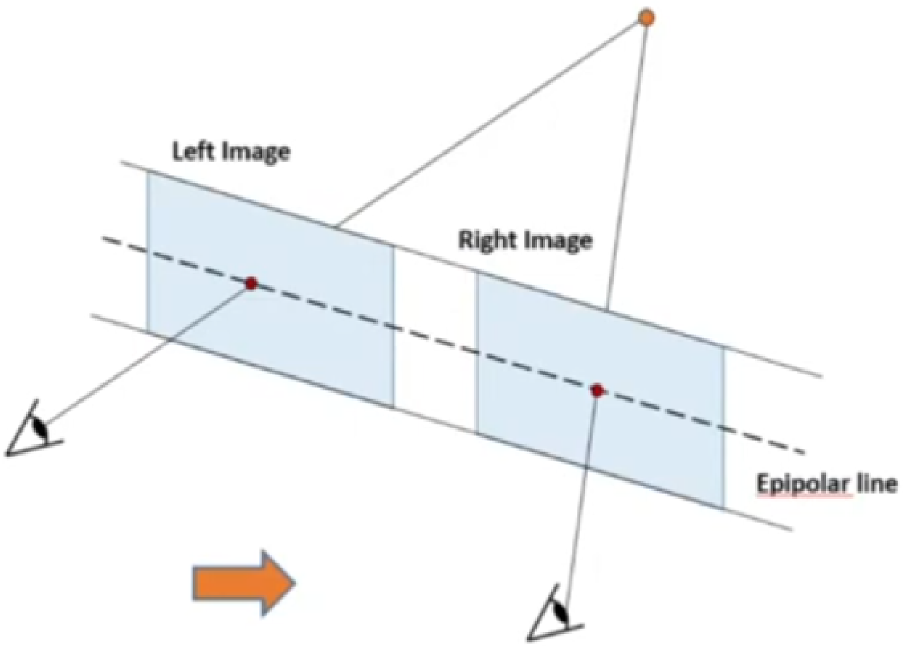




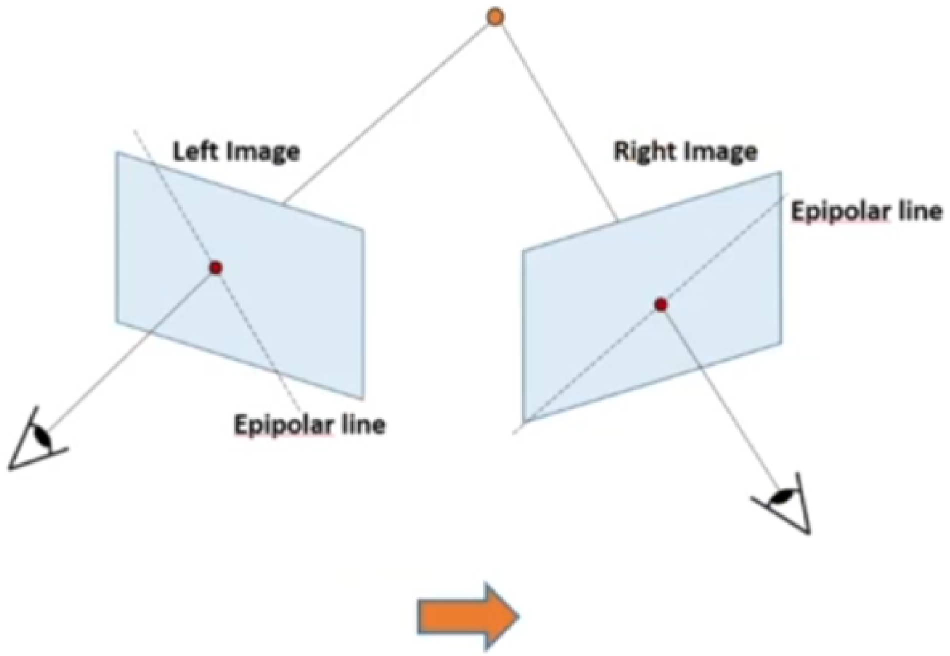
- Simplest case: epipolar lines are corresponding scanlines.
- When does this happen?

Two cases of Epipolar Geometry

- Case with two cameras with parallel optical axes
- General case



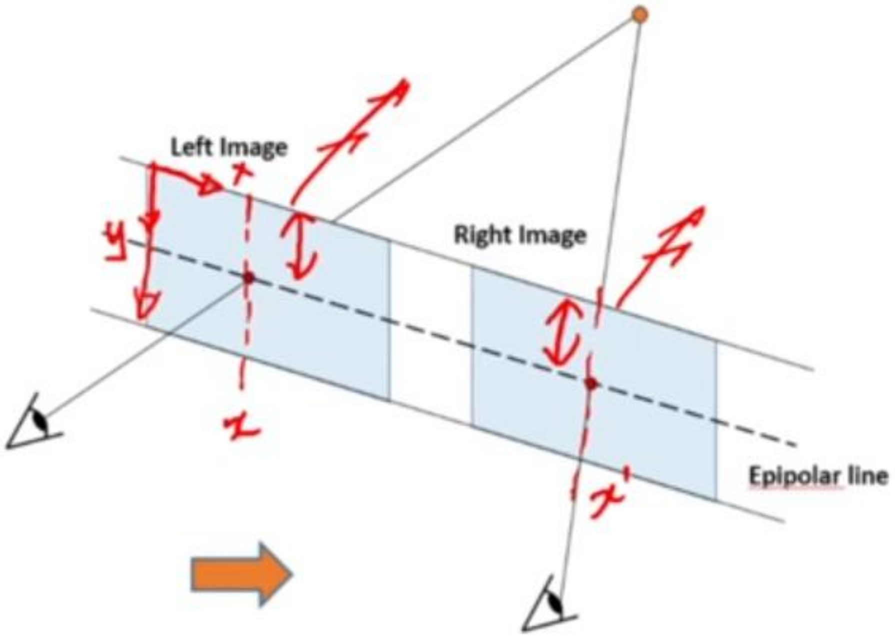
$R = I, \mathbf{t} = (T, 0\ 0)^T$



$[R \mid \mathbf{t}]$

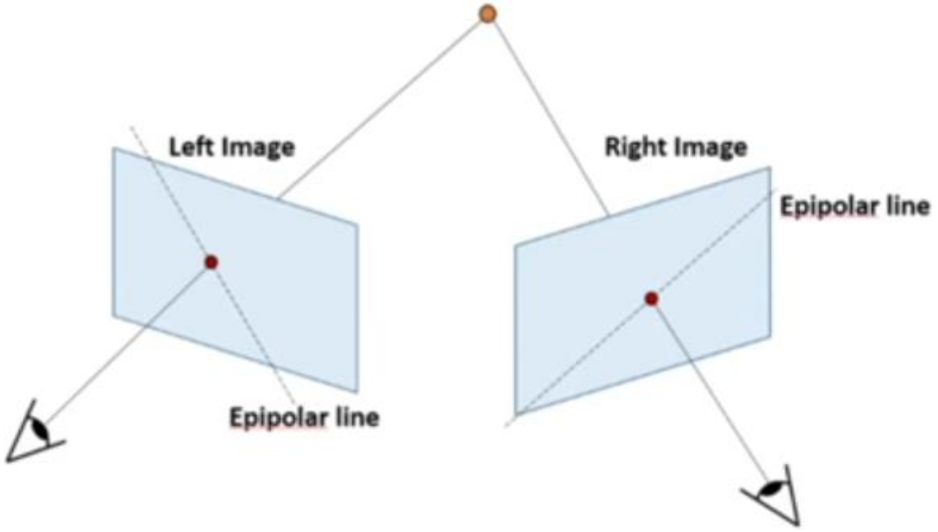
Two cases of Epipolar Geometry

- Case with two cameras with parallel optical axes
- General case



$\mathrm{R} = \mathrm{I}, \mathbf{t} = (T, 0\ 0)^{\mathrm{T}}$

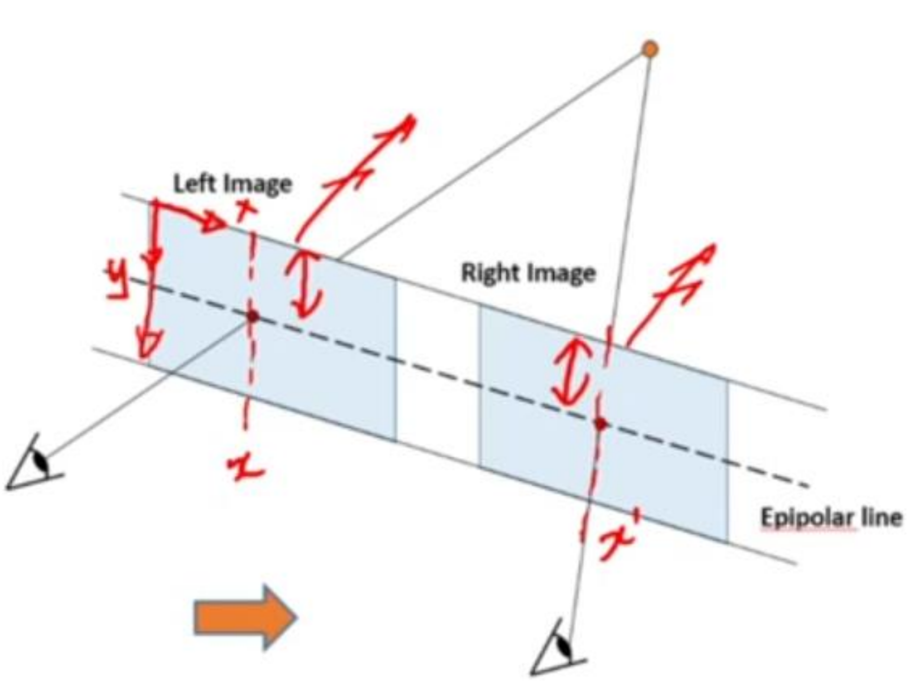
$x - x' = T$



$[\mathrm{R} \mid \mathbf{t}]$

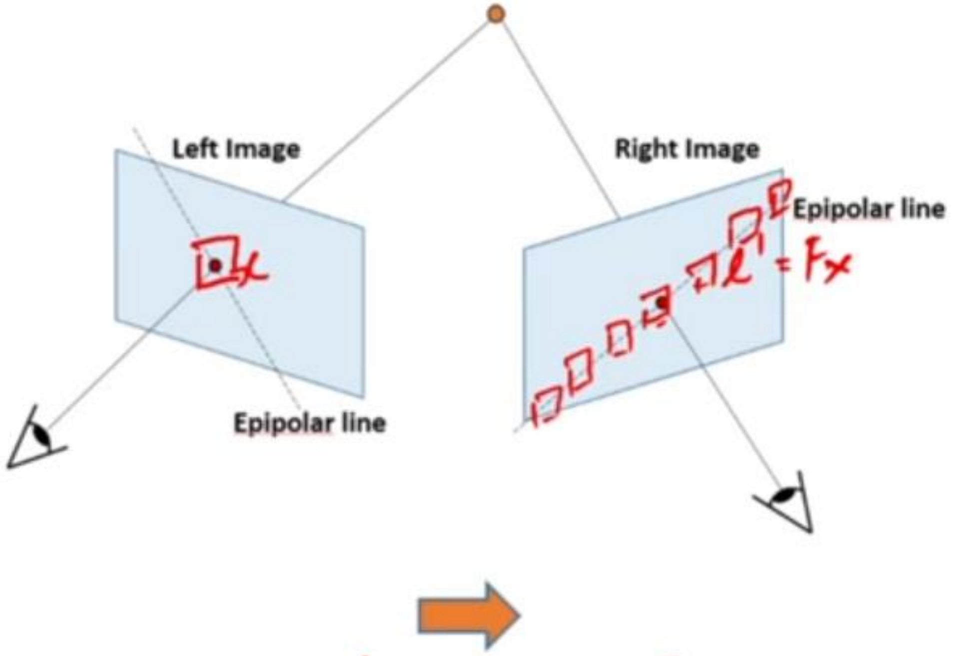
Two cases of Epipolar Geometry

- Case with two cameras with parallel optical axes
- General case



$\mathrm{R} = \mathrm{I}, \mathbf{t} = (T, 0\ 0)^{\mathrm{T}}$

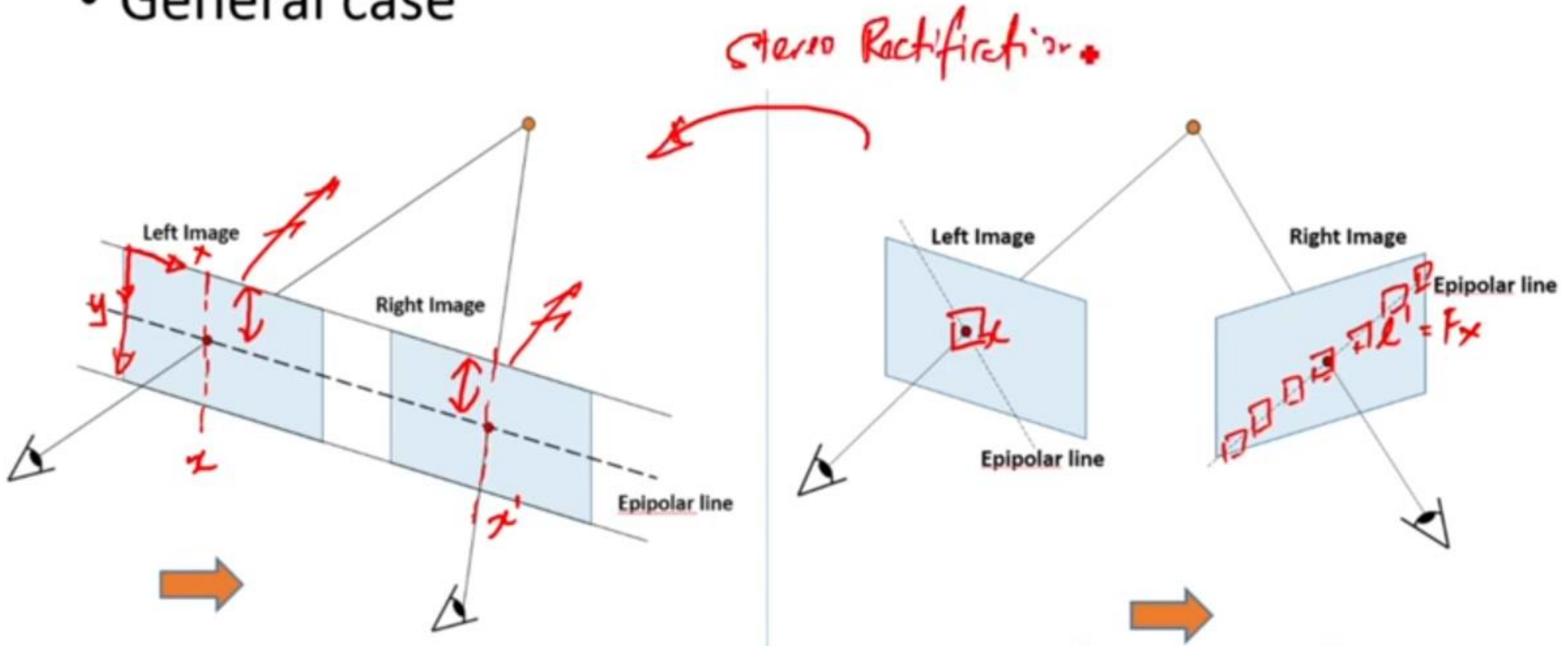
$x - x' = T$



$K, K'\ \underbrace{[\mathrm{R} \mid \mathbf{t}]}_{E} \Rightarrow F.$

Two cases of Epipolar Geometry

- Case with two cameras with parallel optical axes
- General case



$\mathrm{R} = \mathrm{I}, \mathbf{t} = (T, 0\ 0)^{\mathrm{T}}$

$x - x' = T$

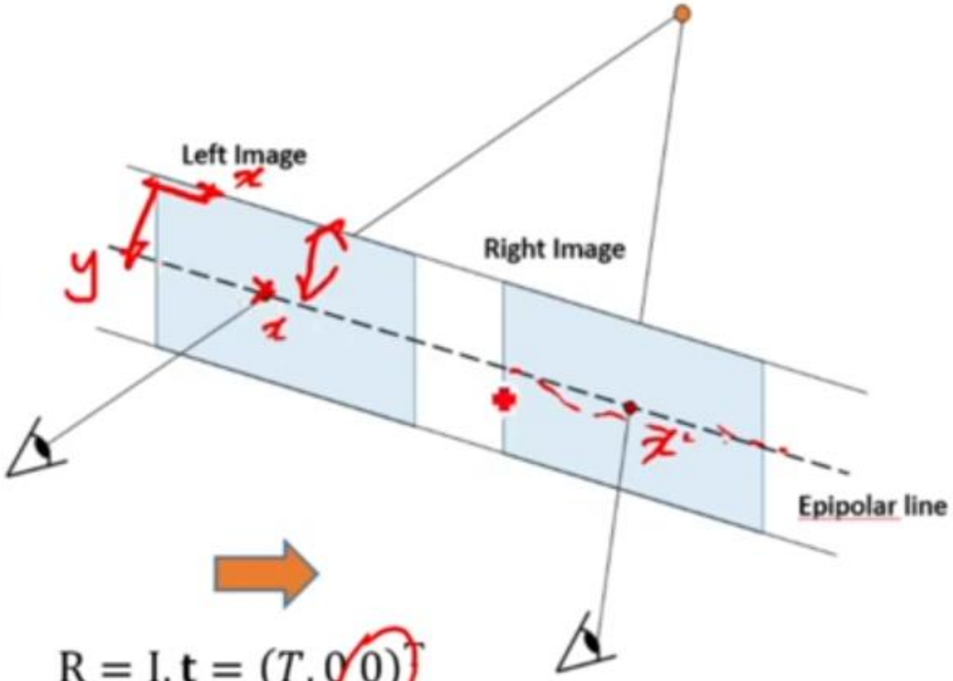
$K, K'\ \underbrace{[\mathrm{R} \mid \mathbf{t}]}_{E} \Rightarrow F.$

Simplest Case: Parallel Images

- Image planes of cameras are parallel to each other and to the baseline.

- Camera centers are at same height and focal lengths are the same.

- Then epipolar lines fall along the horizontal scan lines of the images.



$$\mathrm{R} = \mathrm{I}, \mathbf{t} = (T, 0, 0)^T$$

Essential Matrix for Parallel Images

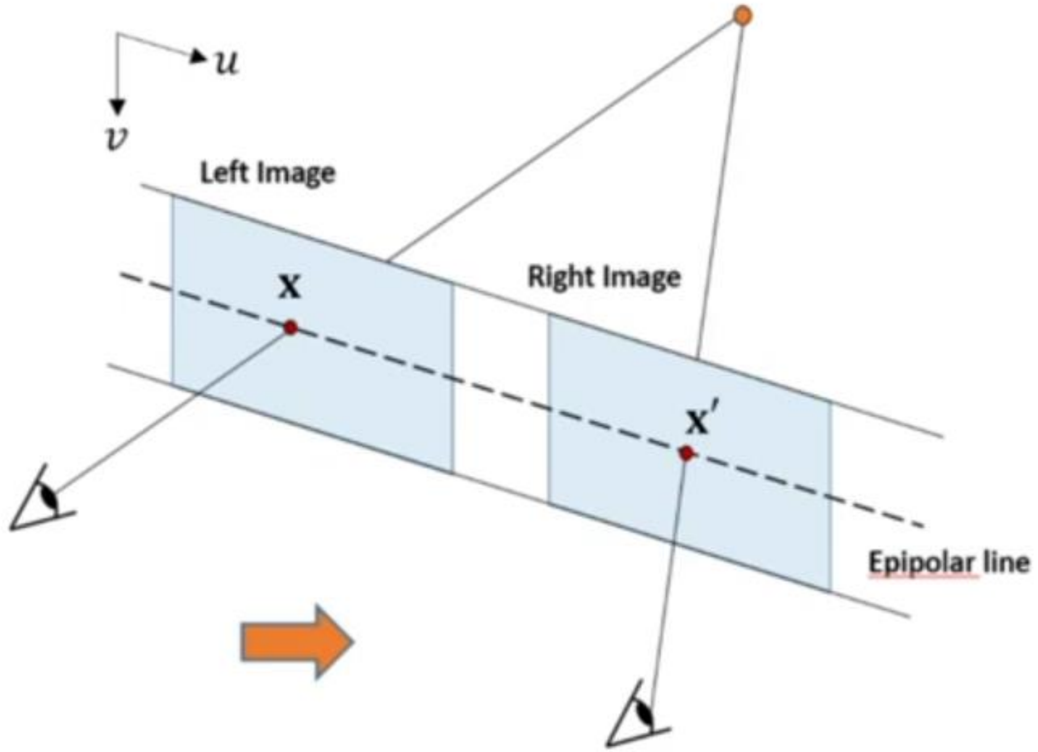
Epipolar constraint:

$$\mathbf{x}'^T \mathrm{E}\mathbf{x} = 0, \qquad \mathrm{E} = [\mathbf{t}_\times]\mathrm{R}.$$

Since

$$\mathrm{R} = \mathrm{I} \text{ and } \mathbf{t} = (T, 0\ 0)^{\mathrm{T}},$$

$$\Rightarrow \mathrm{E} = [\mathbf{t}_\times]\mathrm{R} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix}.$$



Now we have:

$$(u' \quad v' \quad 1)\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix}\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = 0 \Rightarrow (u' \quad v' \quad 1)\begin{pmatrix} 0 \\ -T \\ Tv \end{pmatrix} = 0 \Rightarrow Tv' = Tv.$$

The y-coordinates of corresponding points are the same!

Essential Matrix for Parallel Images

Epipolar constraint:

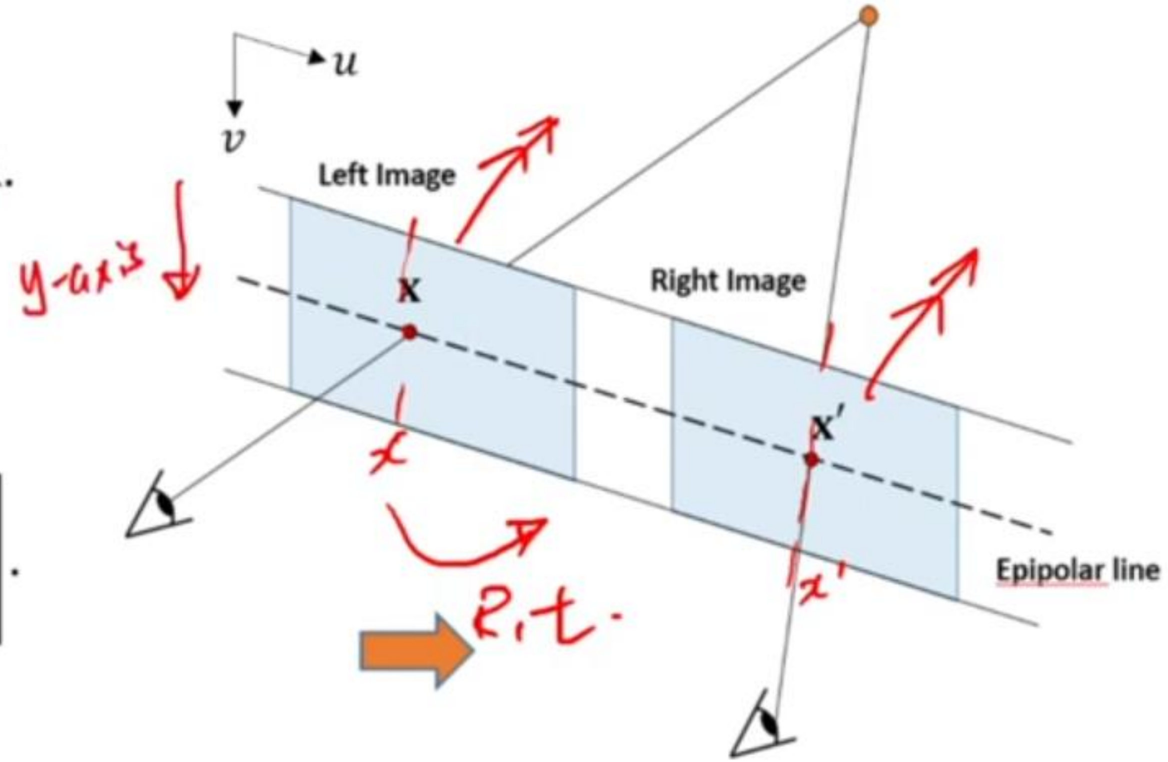
$\Rightarrow$ $\mathbf{x}'^T \mathrm{E}\mathbf{x} = 0, \qquad \mathrm{E} = [\mathbf{t}_\times]\mathrm{R}.$

Since

$x - x'$ $\Downarrow$

$$\mathrm{R} = \mathrm{I} \text{ and } \mathbf{t} = (T, 0\ 0)^{\mathrm{T}},$$

$$\Rightarrow \mathrm{E} = [\mathbf{t}_\times]\mathrm{R} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix}.$$



Now we have:

$$(u' \quad v' \quad 1)\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix}\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = 0 \Rightarrow (u' \quad v' \quad 1)\begin{pmatrix} 0 \\ -T \\ Tv \end{pmatrix} = 0 \Rightarrow \cancel{T}v' = \cancel{T}v.$$

$v' = v$

The y-coordinates of corresponding points are the same!

Simplest Case: Parallel Images

- So all points on the projective line span by the left camera center $\mathbf{C}$ and image point $\mathbf{x}$ project to a horizontal line with $v' = v$ on the right image.

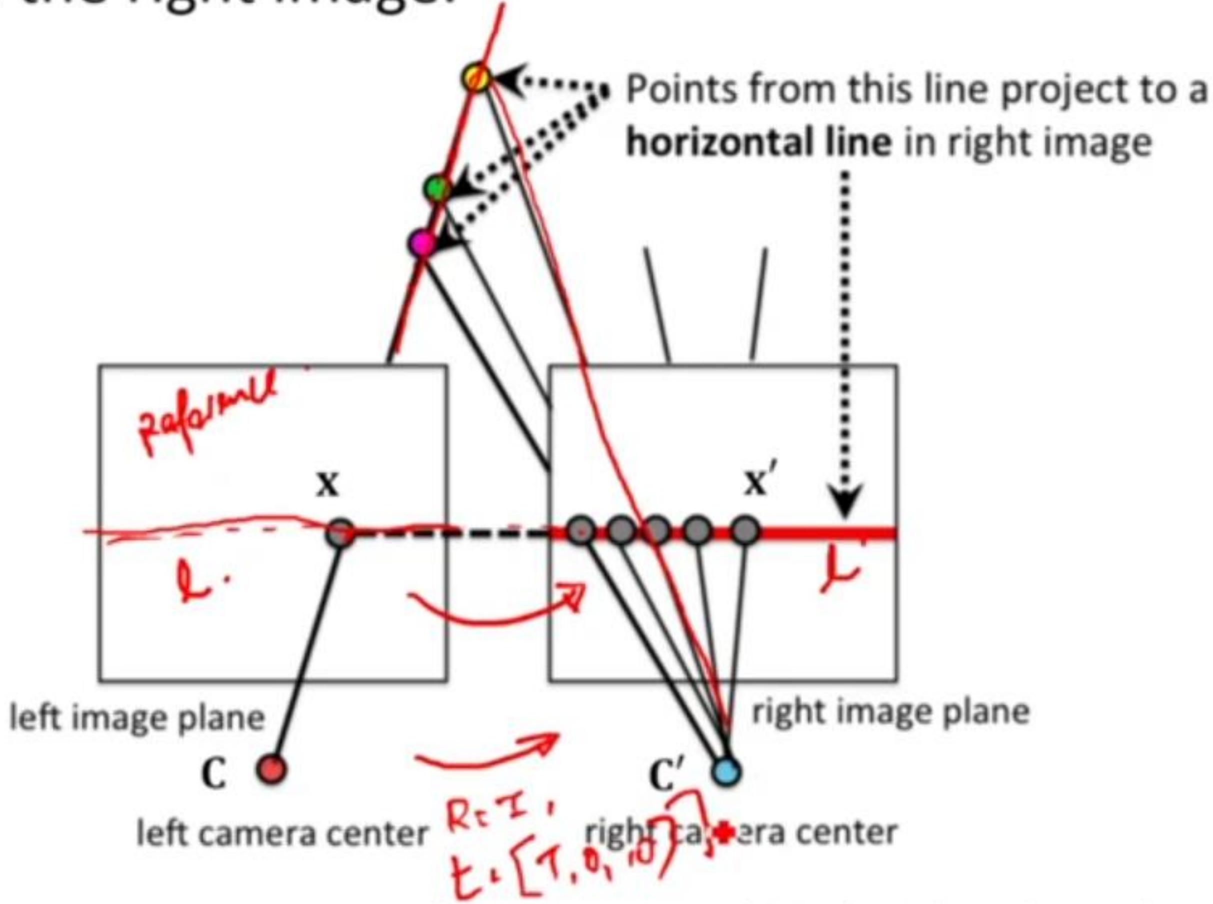
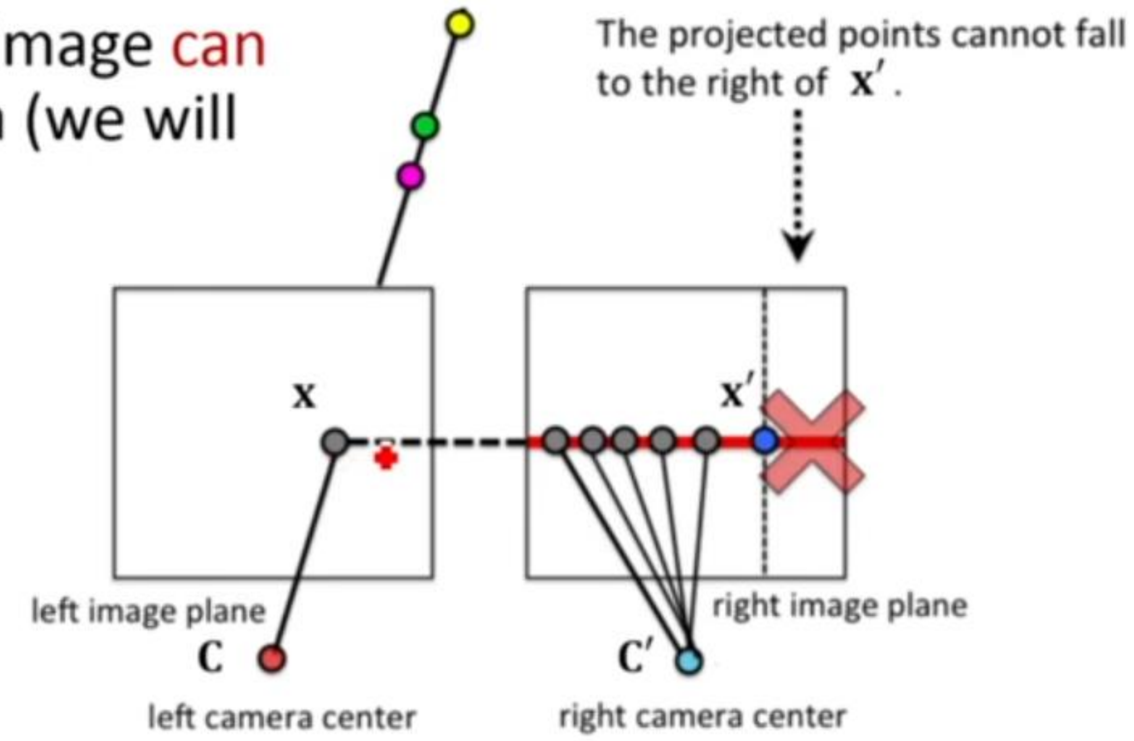


Image source: http://www.cs.toronto.edu/~fidler/slides/2015/CSC420/lecture12_hres.pdf

Simplest Case: Parallel Images

- **Another observation:** No point from the projective line of the left image can project to right of $\mathbf{x}'$ on the right image.

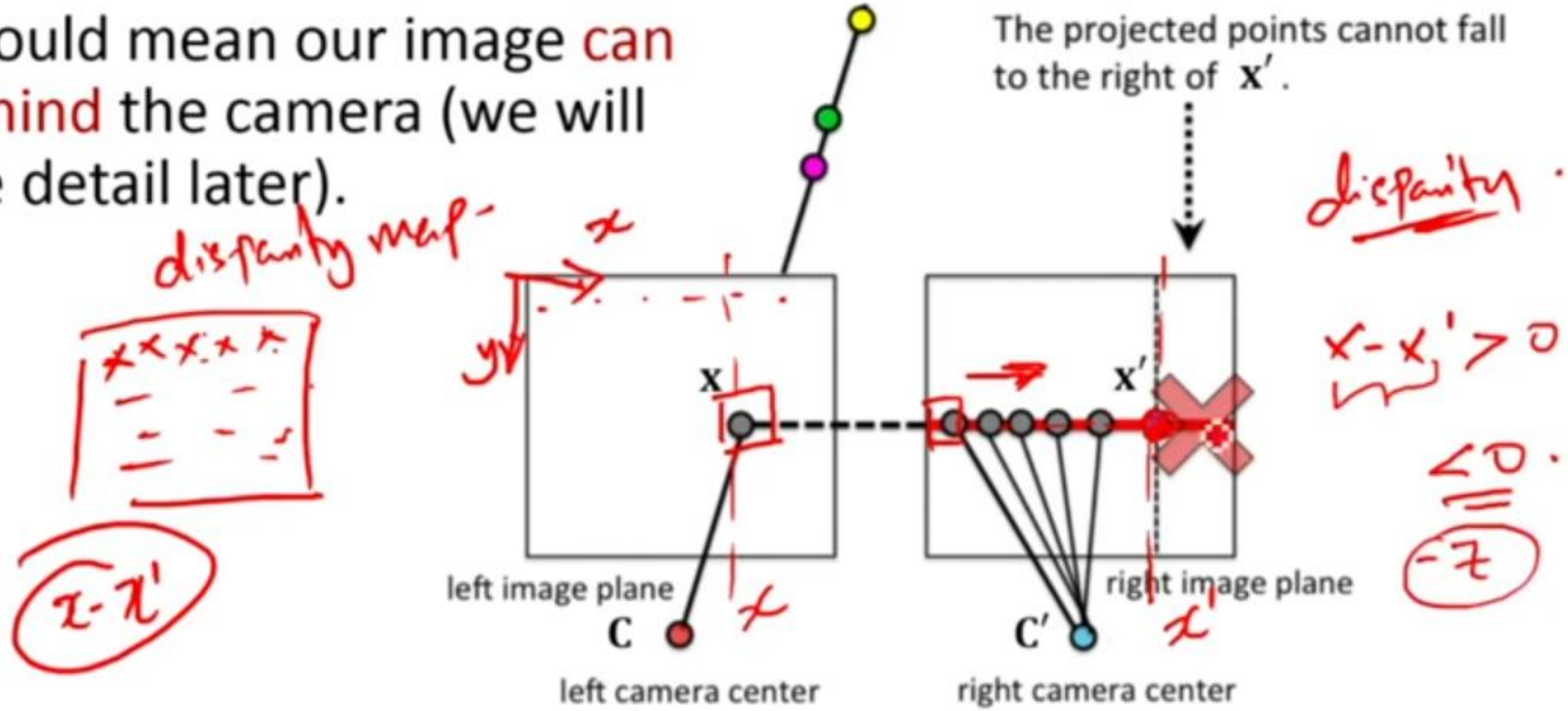
- That would mean our image can see behind the camera (we will see the detail later).



Simplest Case: Parallel Images

- **Another observation:** No point from the projective line of the left image can project to right of $\mathbf{x}'$ on the right image.

- That would mean our image can see behind the camera (we will see the detail later).



General Case: Non-Parallel Images

This is a two-step process:

1. **Stereo calibration**:

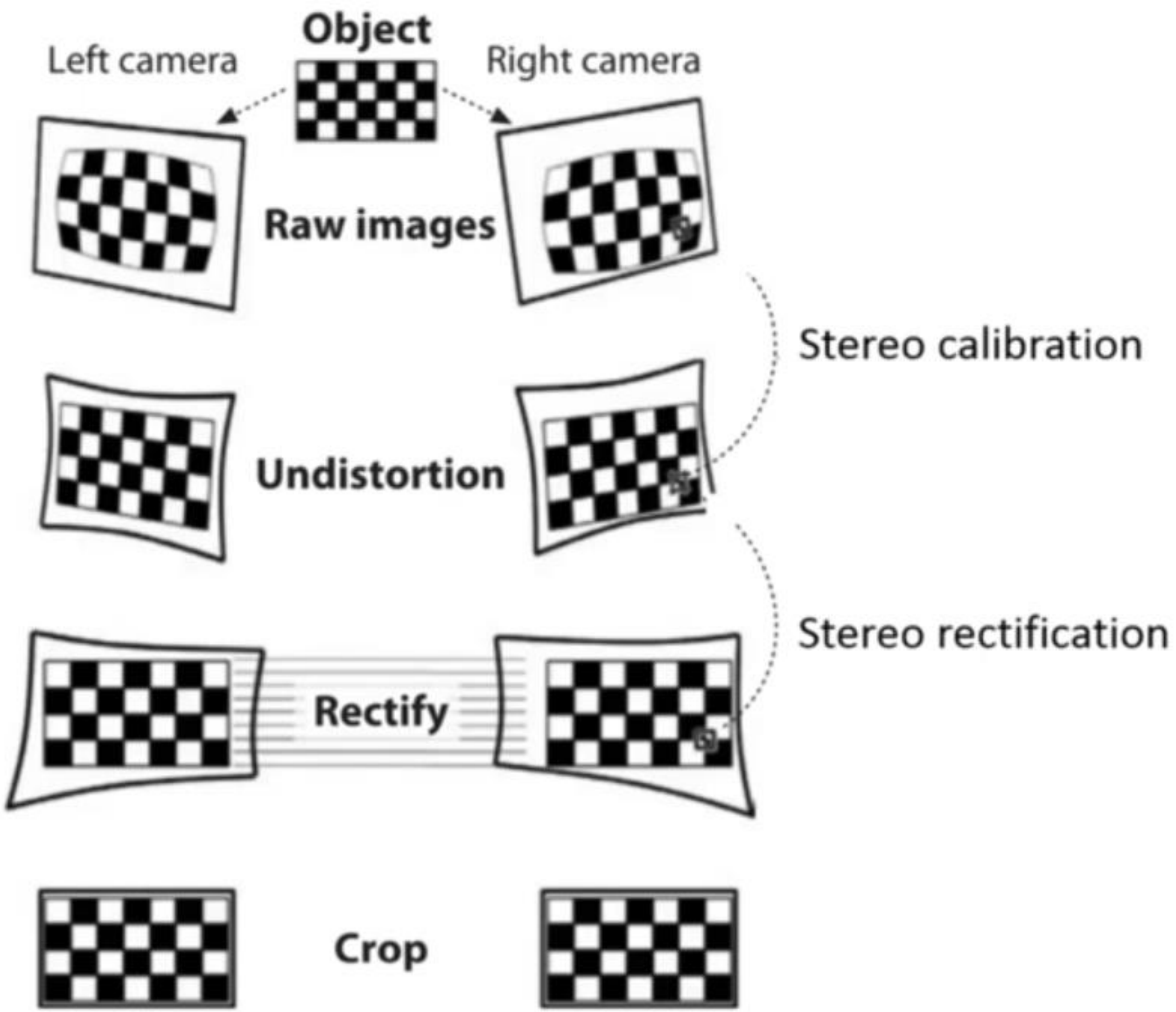
   a. Use the checkerboard pattern to find the intrinsics parameters K and K', and distortion parameters of each camera.

   b. Undistort images and compute the essential matrix E and then decompose to get the relative pose $(\mathrm{R}, \mathbf{t})$ of the cameras.

2. **Stereo rectification**:

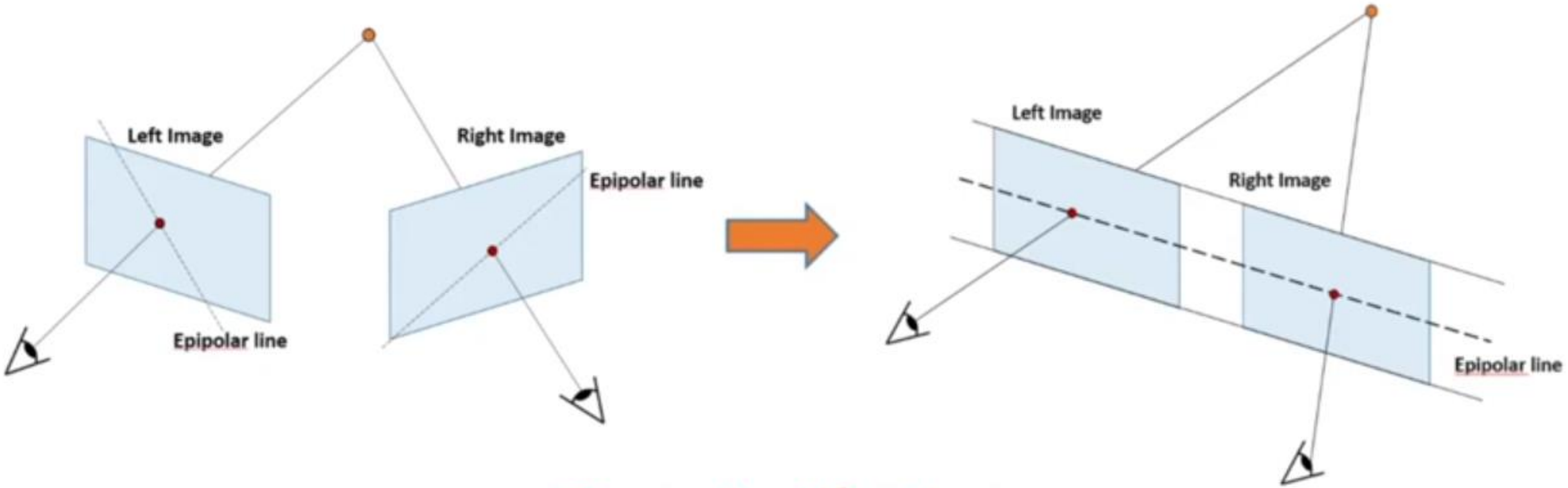
   a. Correct the individual images so that they appear as if they had been taken by two cameras with row-aligned image planes.

General Case: Non-Parallel Images



Stereo Rectification

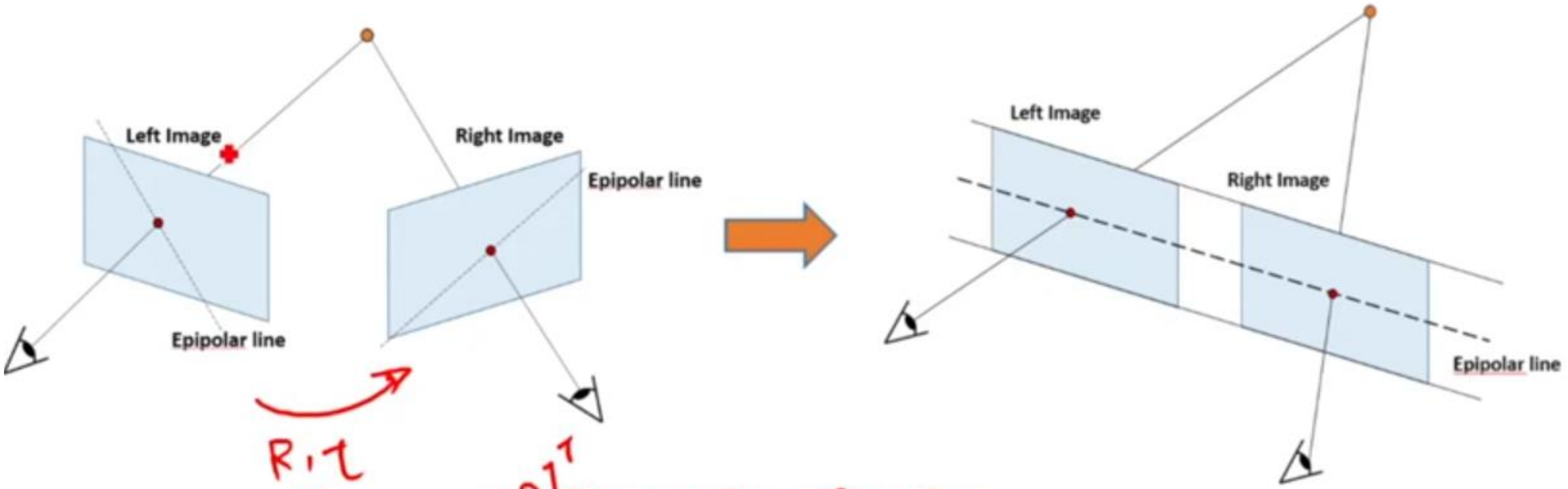
- Our goal is to mathematically align the two cameras into one viewing plane so that pixel rows between the cameras are exactly aligned with each other.



Stereo Rectification

Stereo Rectification

- Our goal is to mathematically align the two cameras into one viewing plane so that pixel rows between the cameras are exactly aligned with each other.



R, T

$\Downarrow$

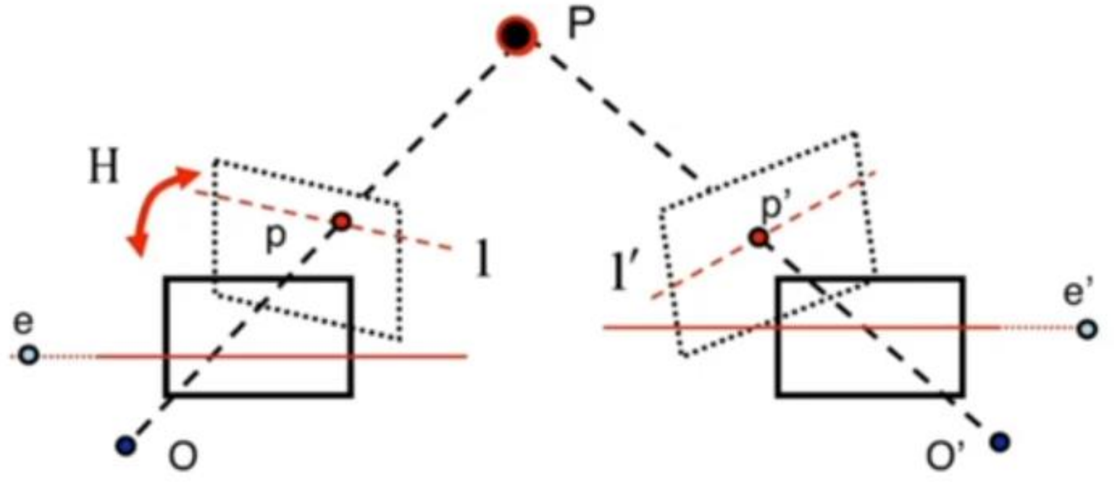
$R = I,\ t = [T, 0, 0]^T$

Stereo Rectification

Stereo Rectification

- All rectified images satisfy the following two properties:

1. All epipolar lines are parallel to the horizontal axis.
2. Corresponding points have identical vertical coordinates.



Stereo Rectification

- **Goal:** Find a projective transformation H such that the epipoles $\boldsymbol{e}$ and $\boldsymbol{e}'$ in the two images are mapped to the infinite point $[1,0,0]^T$.

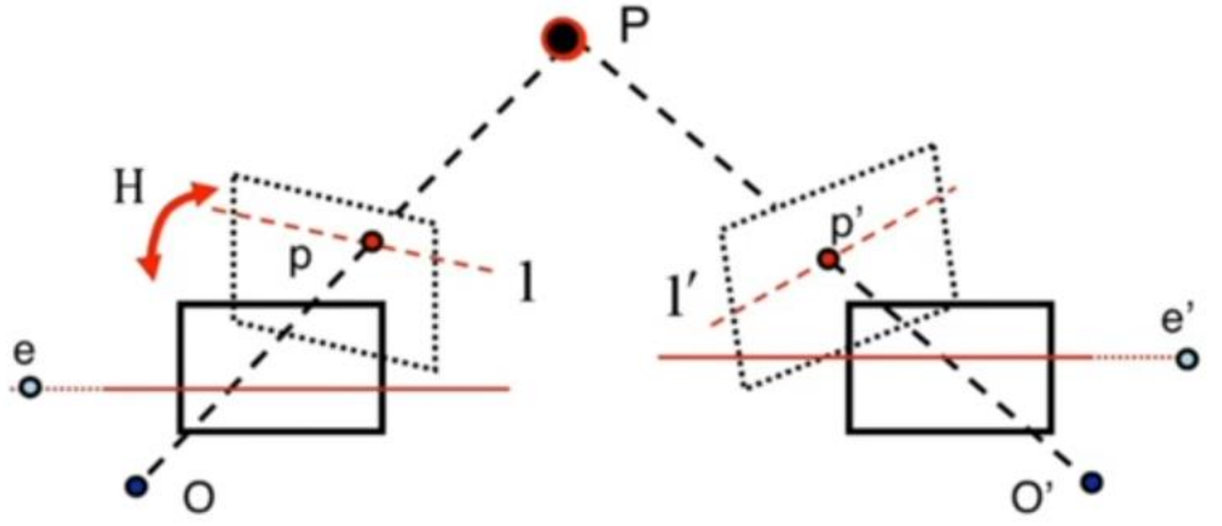


Image source: https://en.wikipedia.org/wiki/Image_rectification

Stereo Rectification

- **Goal:** Find a projective transformation H such that the epipoles $\boldsymbol{e}$ and $\boldsymbol{e}'$ in the two images are mapped to the infinite point $[1,0,0]^T$.

= infinite.

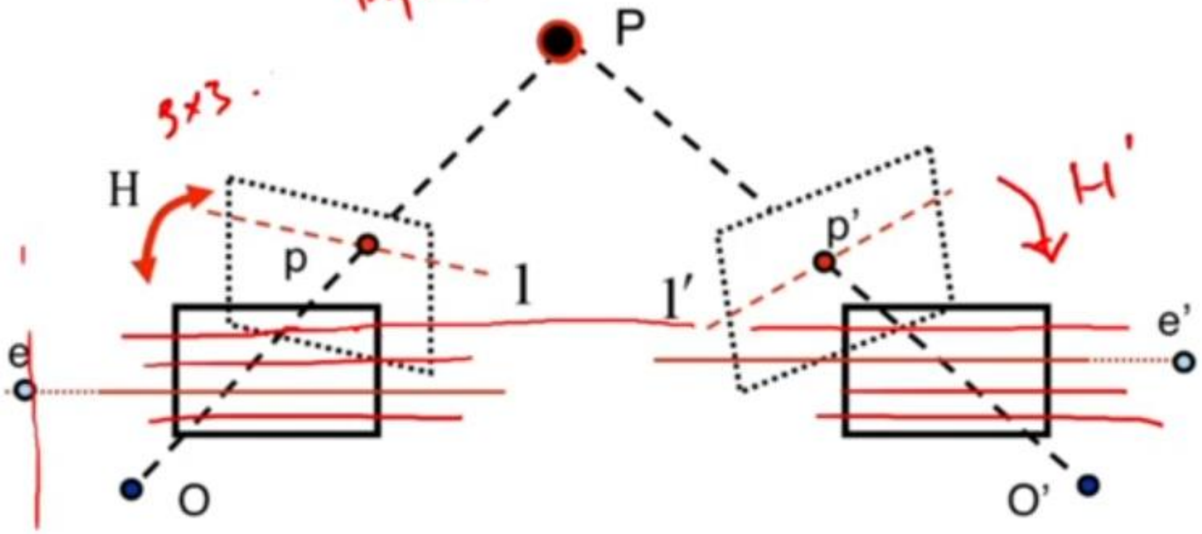
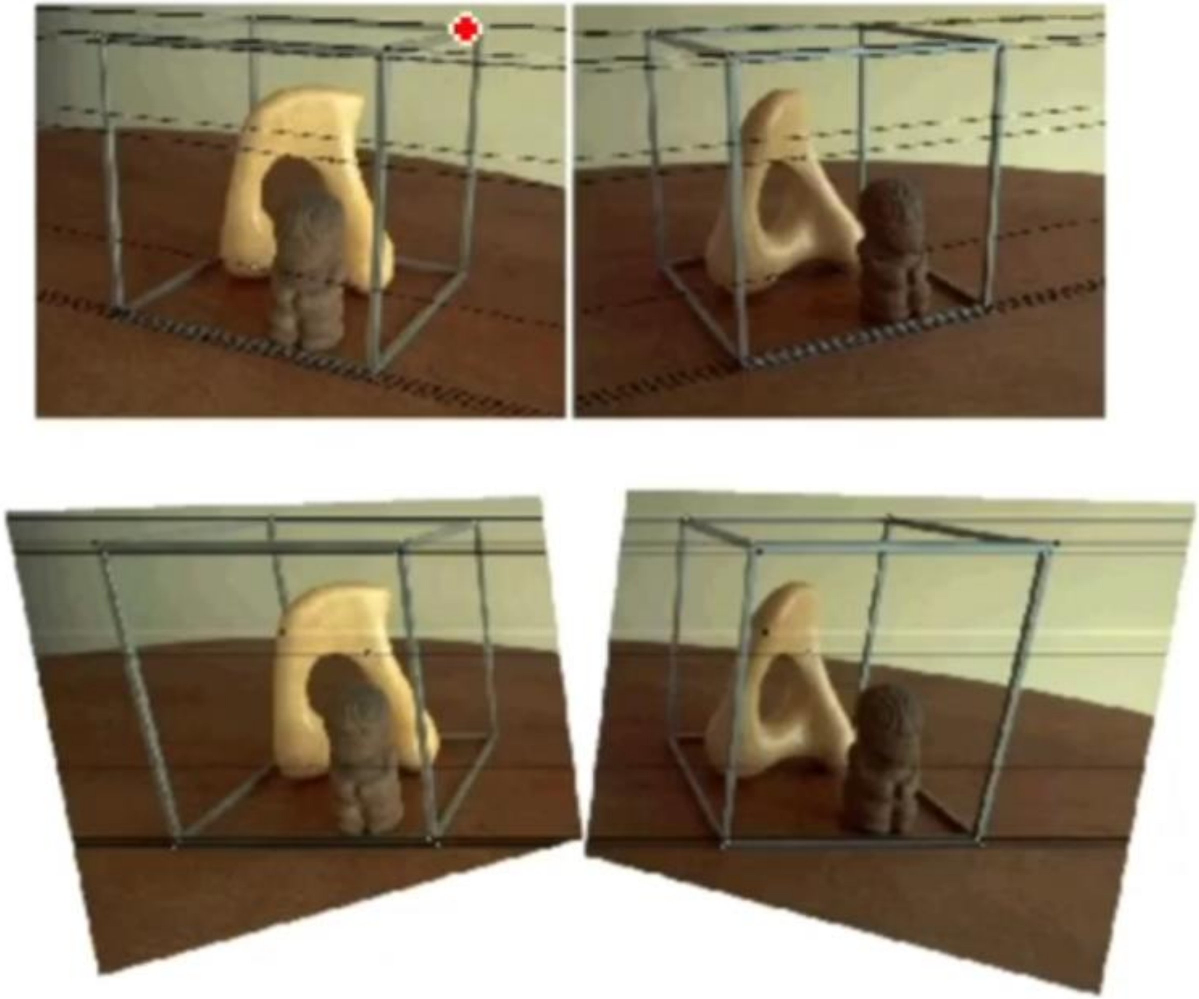


Image source: https://en.wikipedia.org/wiki/Image_rectification

Stereo Rectification



Questions?