Multi-View Geometry 4

Sreeharsha Paruchuri
Robotics Research Center, IIIT-Hyderabad
venkata.surya@students.iiit.ac.in

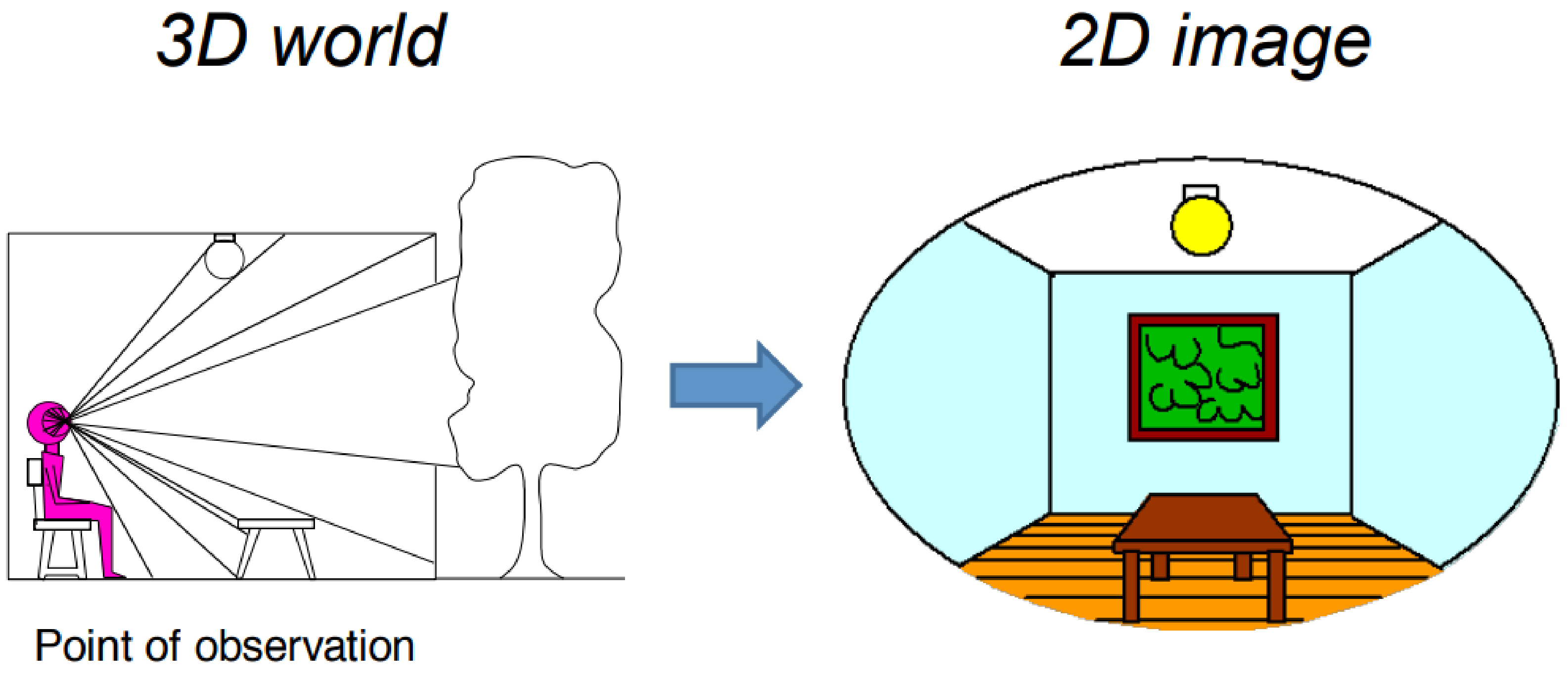
7th June 2021

Slides adapted from CMU 16-385 Computer Vision Course

The role of cameras in perception

- We as humans have 5 sensory organs. Robots?
- How much information can cameras give?
- Recall the properties of a camera.

Camera – A dimensionality reduction machine



What’s the problem with 2D information?

Intrinsic Parameter
(intrinsic matrix)

Extrinsic Parameter
(extrinsic matrix)

$$\lambda \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & x_{c0} \\ 0 & f_y & y_{c0} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r11 & r12 & r13 & t_x \\ r21 & r22 & r23 & t_y \\ r31 & r32 & r33 & t_z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

degree of freedom
(for the depth)

Point in
Captured Iamage

focal length f, center of the image c0

rotaional component r

translate components t

Point in Marker
Coodinate

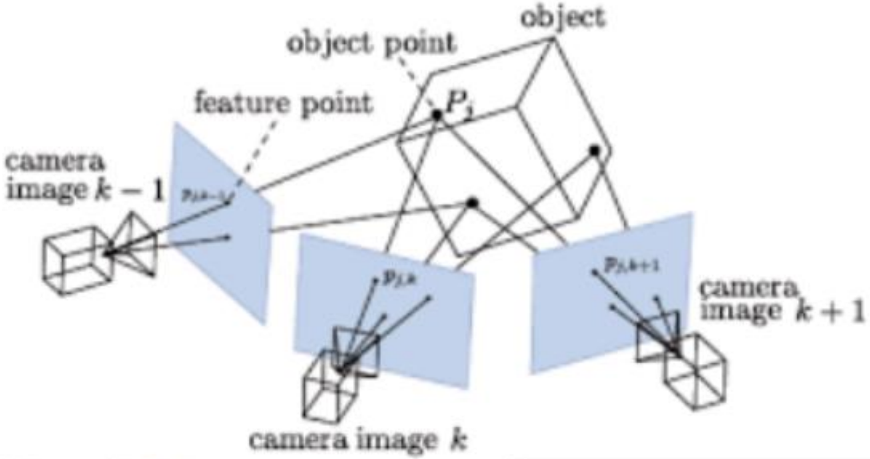
Point in Camera Coordinate

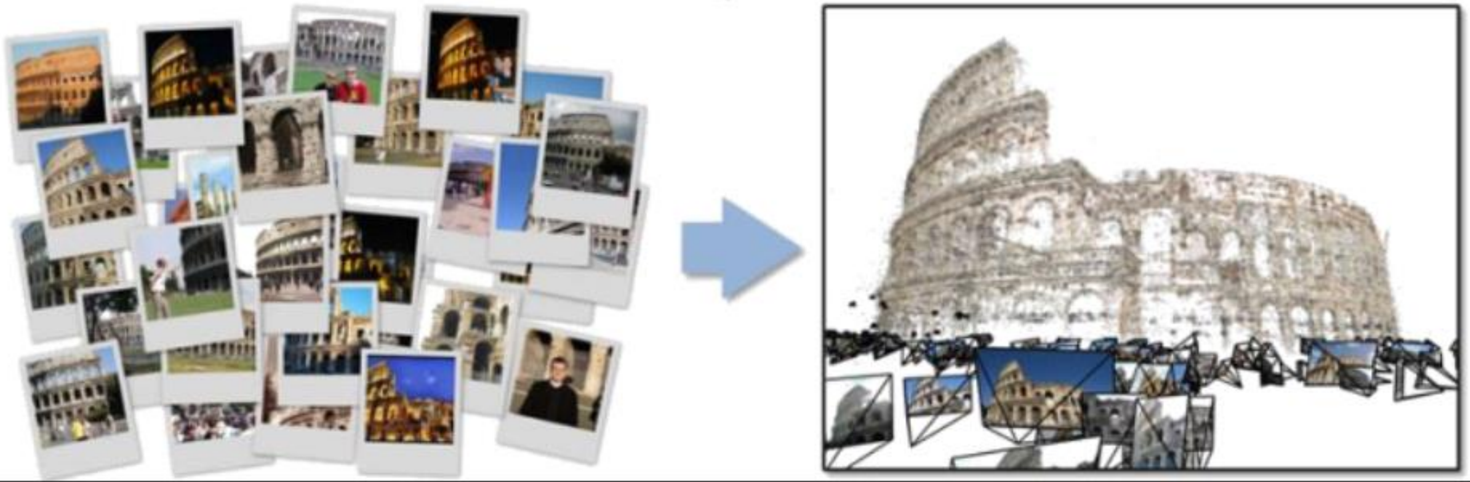
Point in Captured Image coordinate

Why is 3D information important?

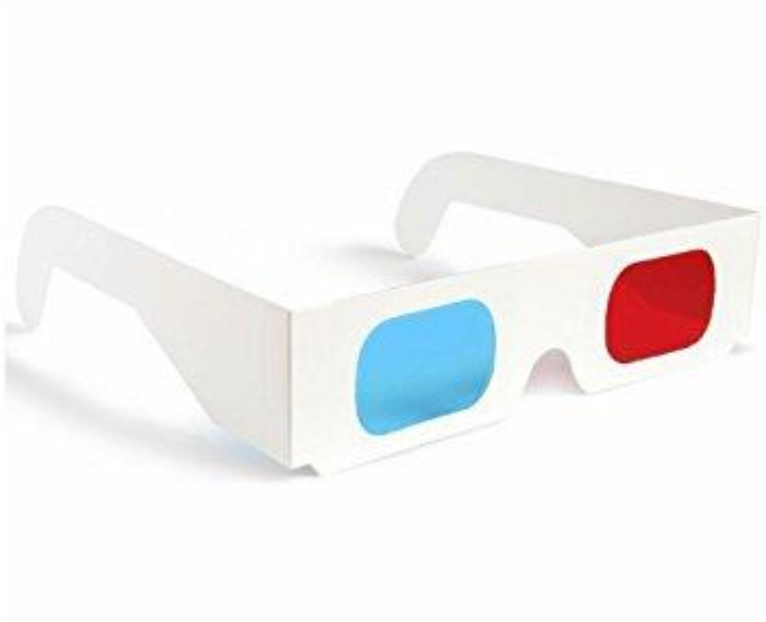
- 3D information -> Depth
- Perception of the environment
- Gauge how close/far away.
- Applications: Self-Driving

If not one then how many?

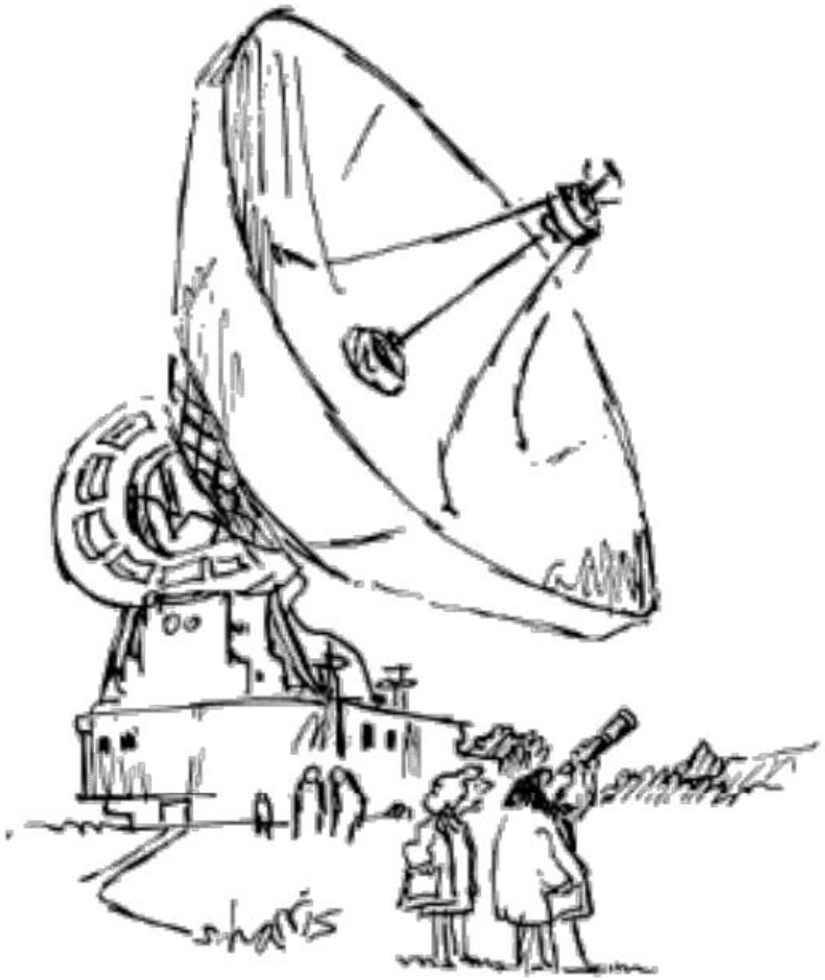




Enter Stereo-Vision!



Two is better than one



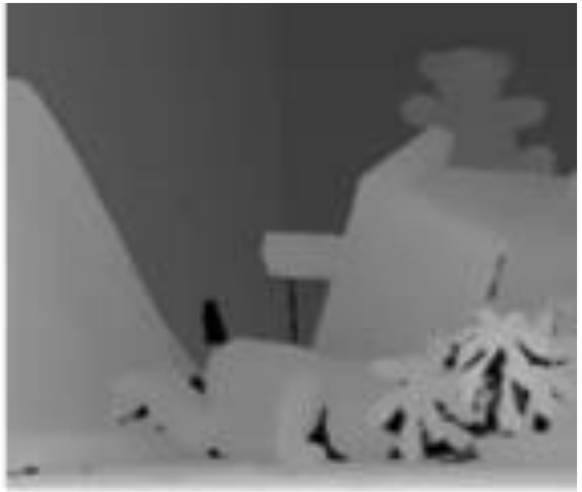
"Just checking."

Biomimicry

- Hold your index finger an arm’s length away.
- Look at it through the left eye keeping the right eye closed.
- Now look at it through the right eye keeping the left one closed.
- You will perceive a shift - this is called as stereo disparity and the brain uses it heavily to infer depth!
- Can we model this problem?

Objects that are close move more or less?

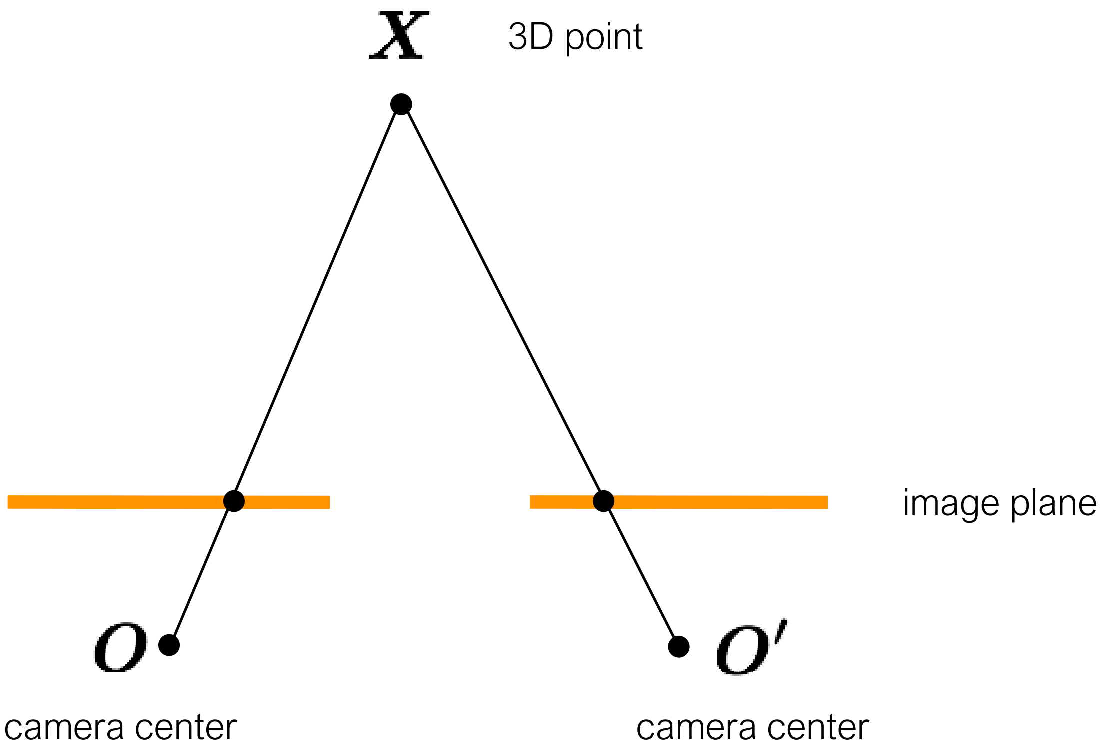
The amount of horizontal movement is inversely proportional to …



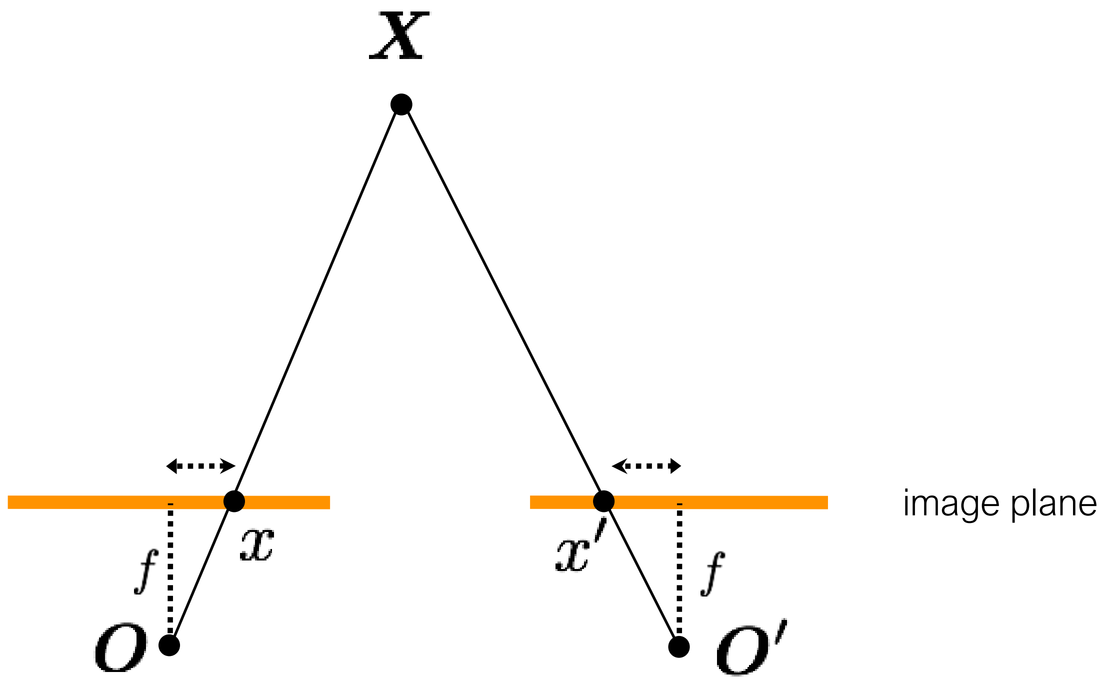
The amount of horizontal movement is inversely proportional to …

… the distance from the camera.

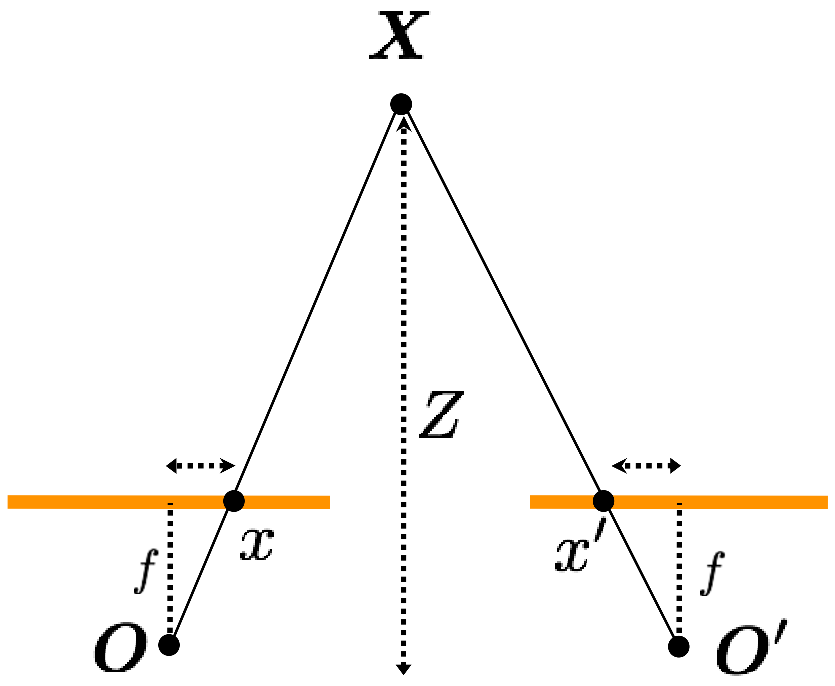
More formally…

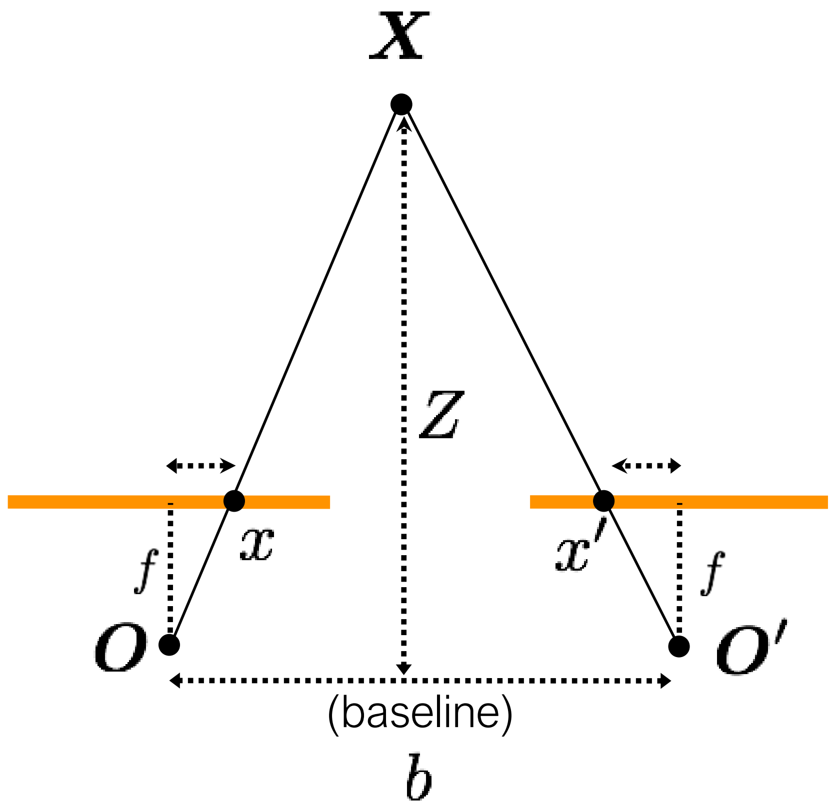


X
3D point
image plane
O
camera center
O'
camera center

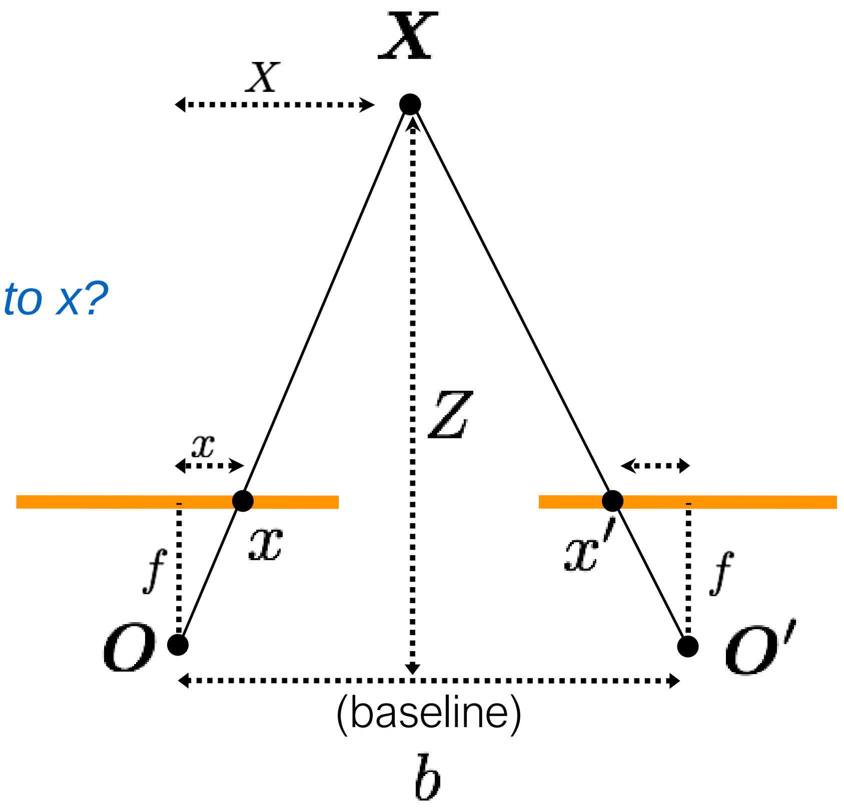


X
x
x'
f
f
O
O'
image plane

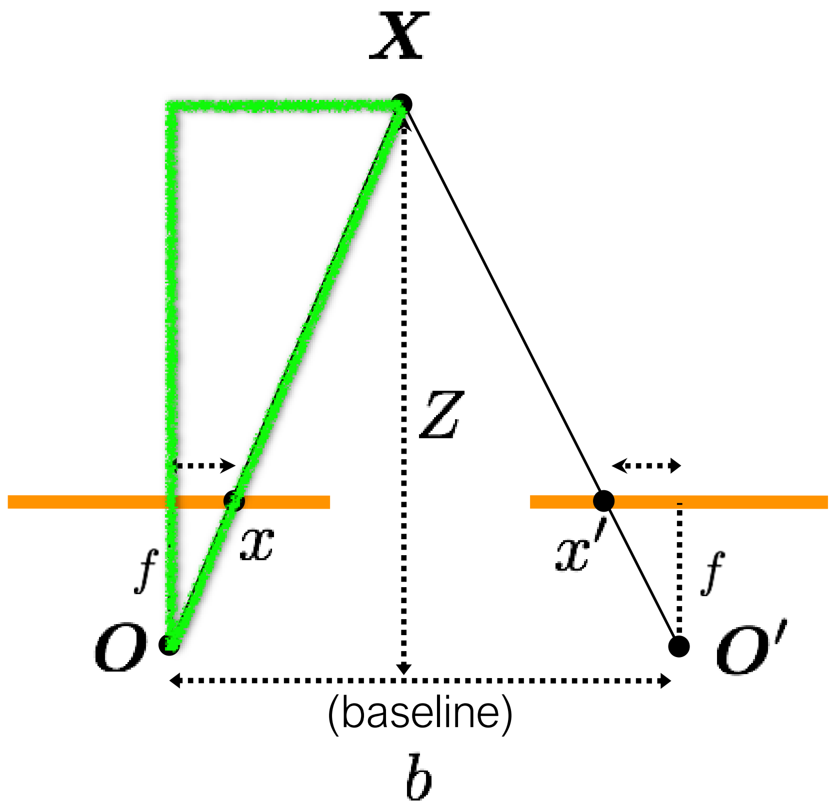

$\boldsymbol{X}$
Z
x
x'
f
f
$\boldsymbol{O}$
$\boldsymbol{O}'$


$\boldsymbol{X}$
Z
x
x'
f
f
$\boldsymbol{O}$
$\boldsymbol{O}'$
(baseline)
b

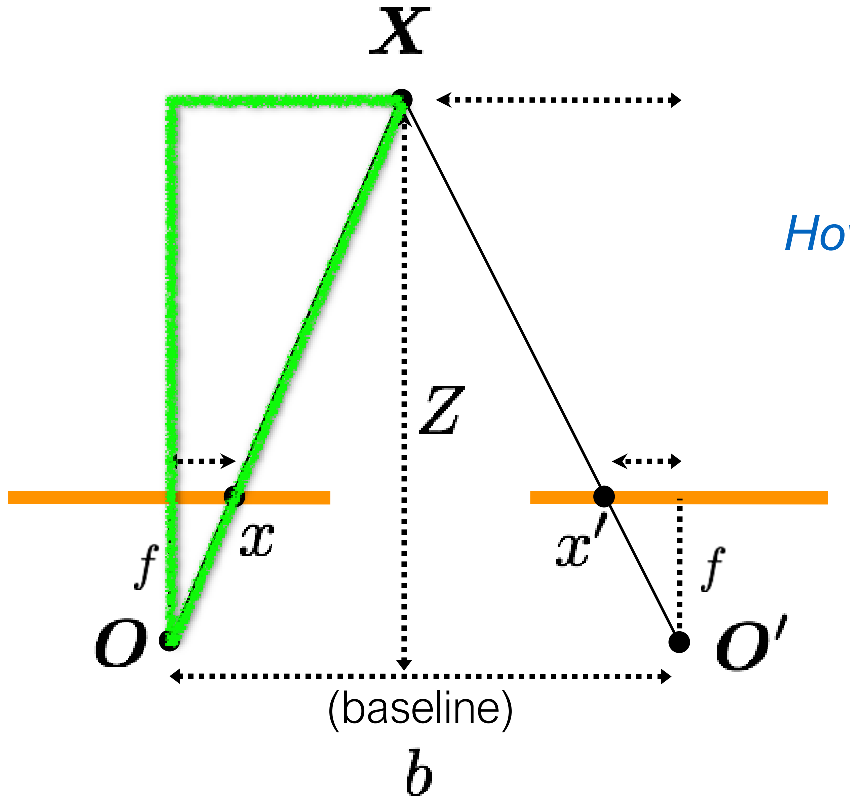
How is X related to x?



$$\frac{X}{Z} = \frac{x}{f}$$

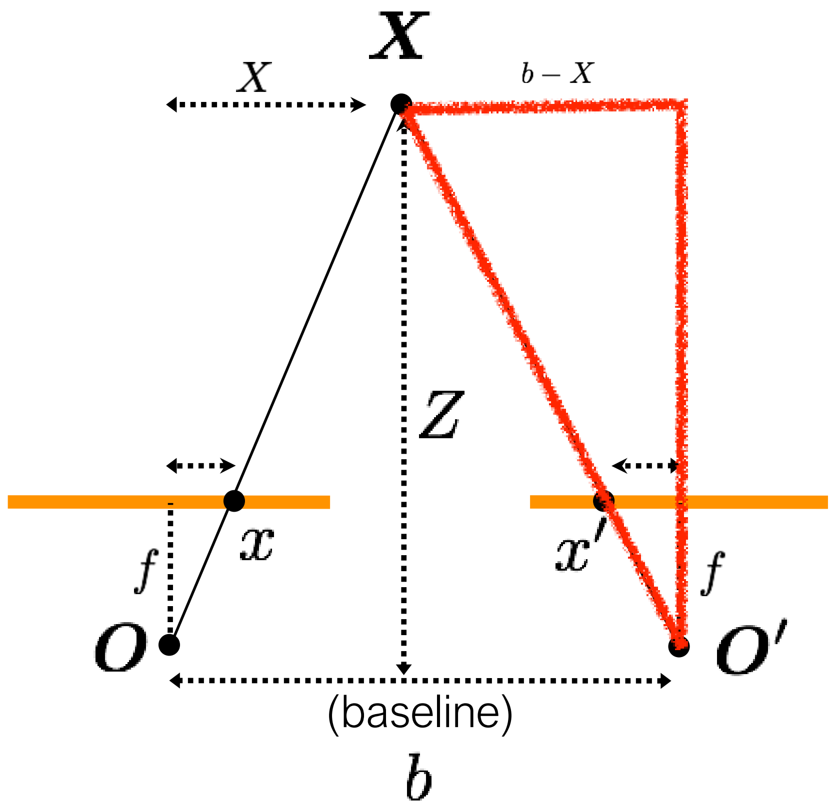


$$\frac{X}{Z} = \frac{x}{f}$$



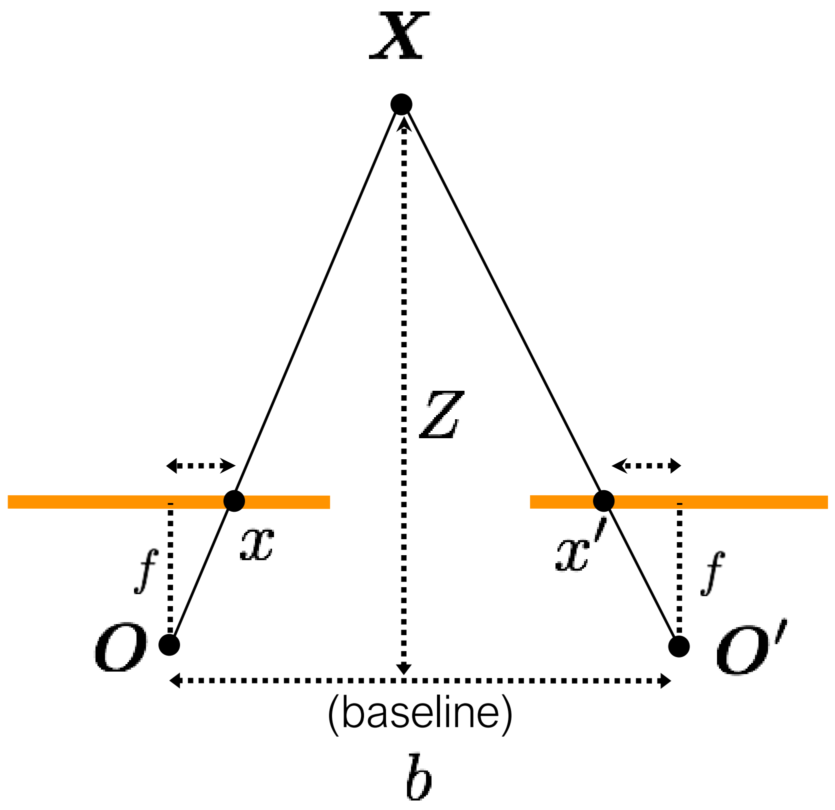
How is X related to x'?

$$\frac{X}{Z} = \frac{x}{f}$$



$$\frac{b - X}{Z} = \frac{x'}{f}$$

$$\frac{X}{Z} = \frac{x}{f}$$



$$\frac{b - X}{Z} = \frac{x'}{f}$$

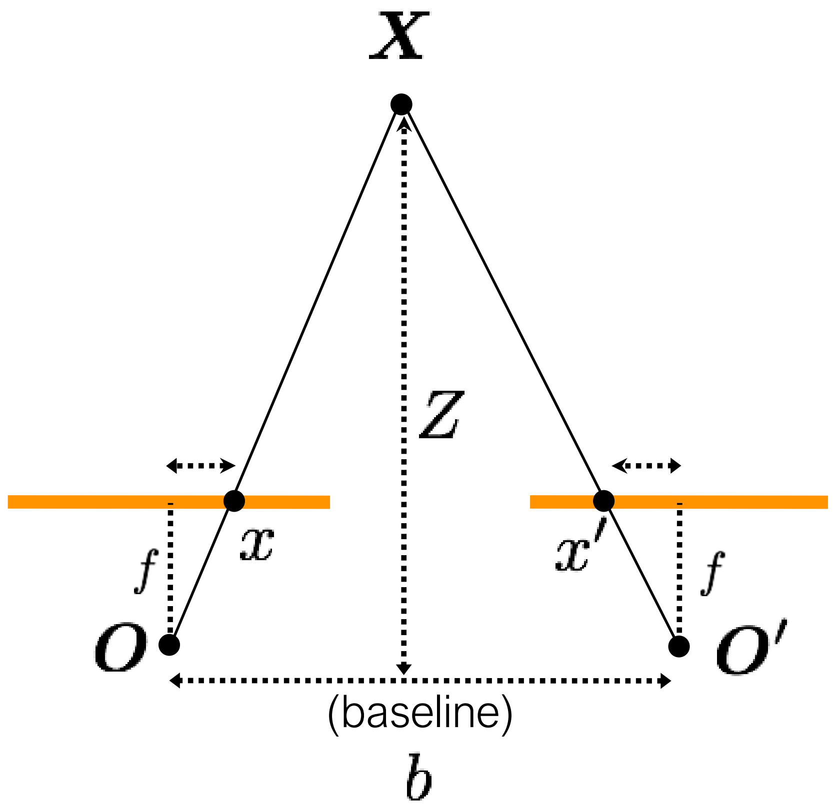
Disparity

$$d = x - x'$$

(wrt to camera origin of image plane)

$$= \frac{bf}{Z}$$

$$\frac{X}{Z} = \frac{x}{f}$$



$$\frac{b - X}{Z} = \frac{x'}{f}$$

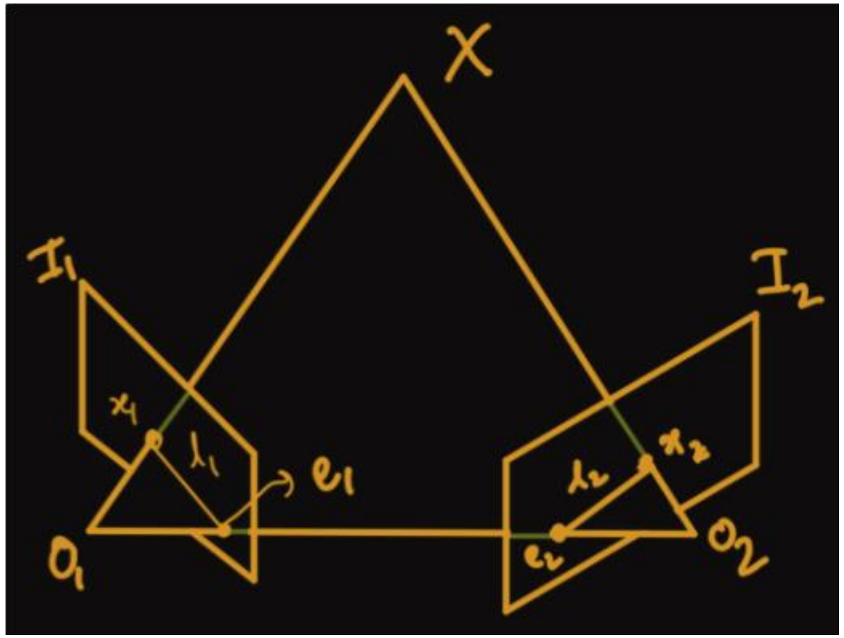
Disparity

$$d = x - x'$$
$$= \frac{bf}{Z}$$

inversely proportional
to depth

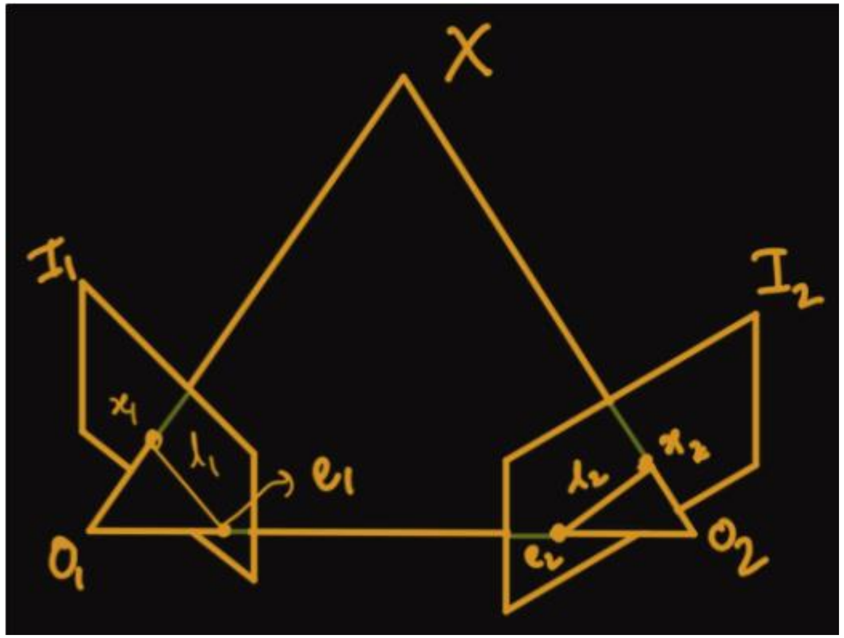
Doesn’t this look familiar?

- 2D-2D correspondences.
- Epipolar Geometry.
- Stereo Vision is a special case.
- Correspondences between images?

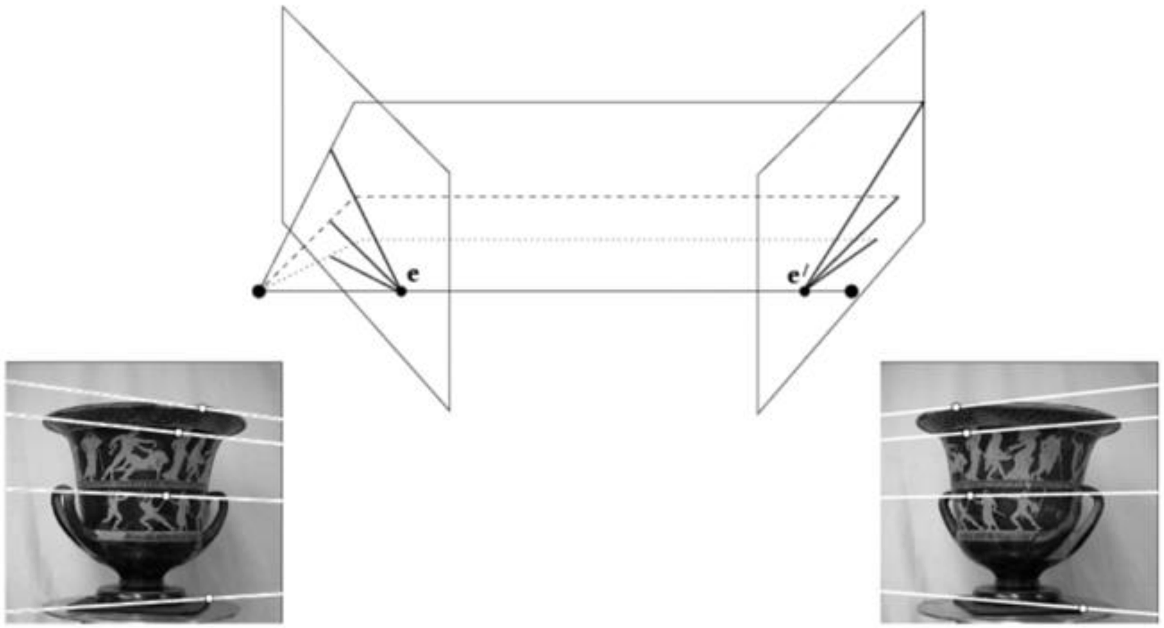


Recall the epipolar constraints

- O1, O2 and X lie on a plane – epipolar plane
- x1, x2 are projections of X - correspondences
- e1, e2 are the epipoles – images of O2, O1
- l1, l2 are the epipolar lines – 1D search
- Epipolar lines pass through the epipoles
- Essential matrix – relationship between planes

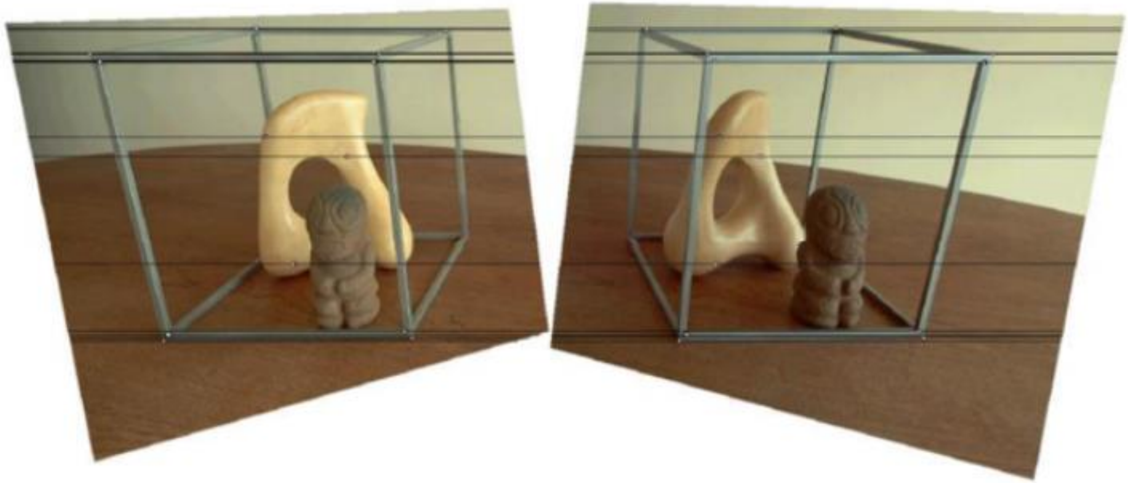


Visualising the epipolar lines

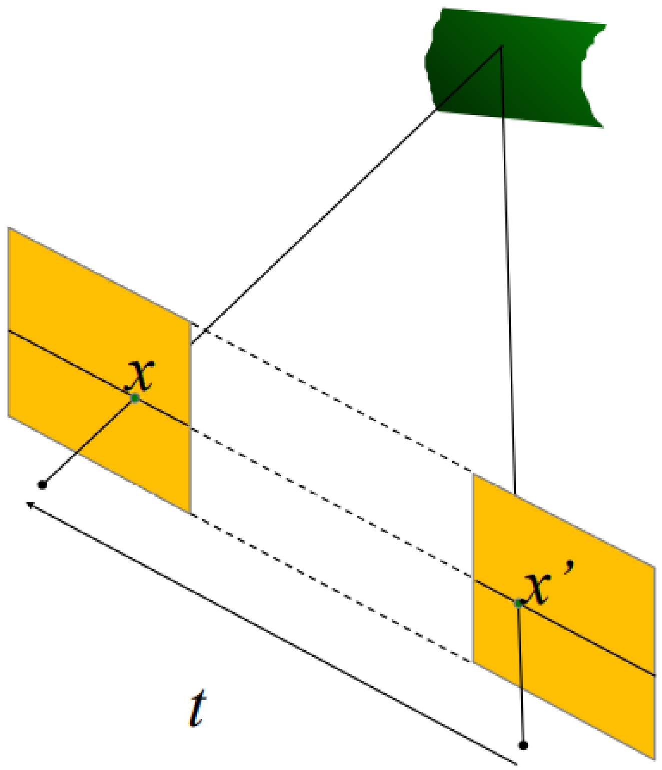


What have we achieved? Problems?

How can you make the epipolar lines horizontal?



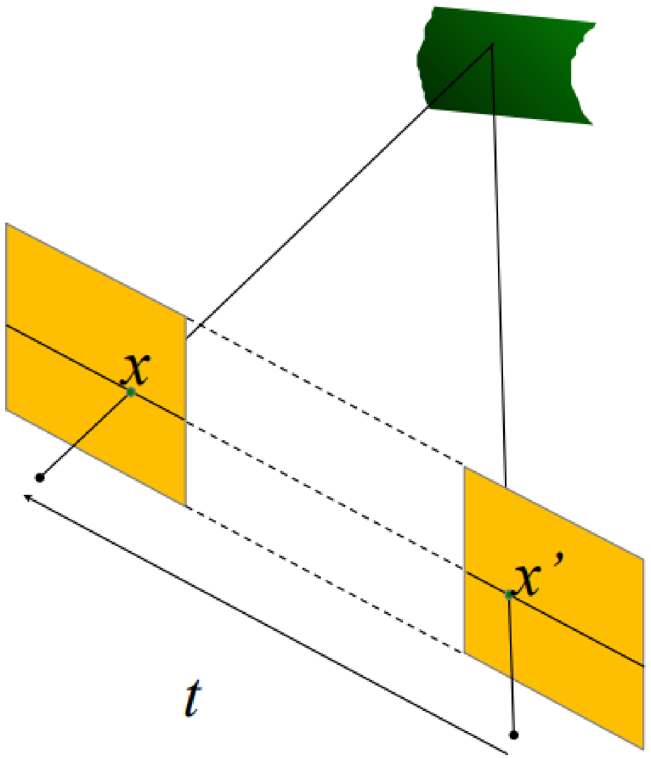
How can you make the epipolar lines horizontal?



When this relationship holds:

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

How can you make the epipolar lines horizontal?



When this relationship holds:

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

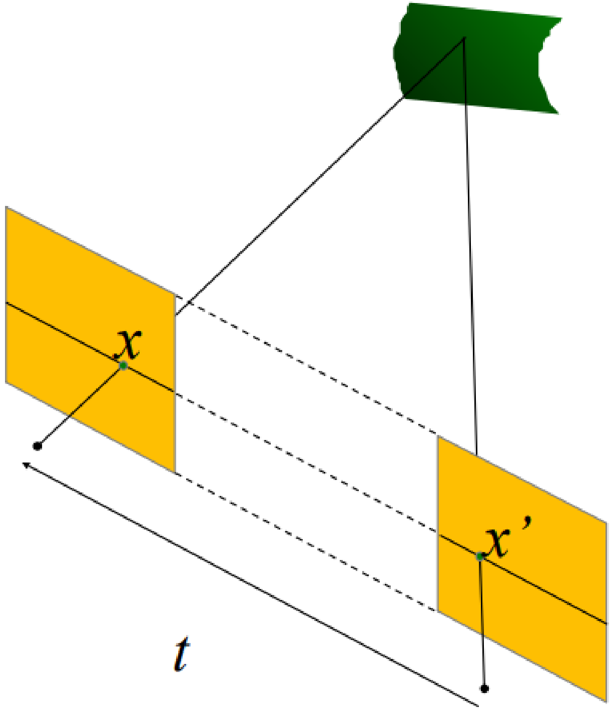
Let's try this out…

$$E = t \times R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix}$$

This always has to hold

$$x^T E x' = 0$$

How can you make the epipolar lines horizontal?



When this relationship holds:

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

Let's try this out...

$$E = t \times R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix}$$

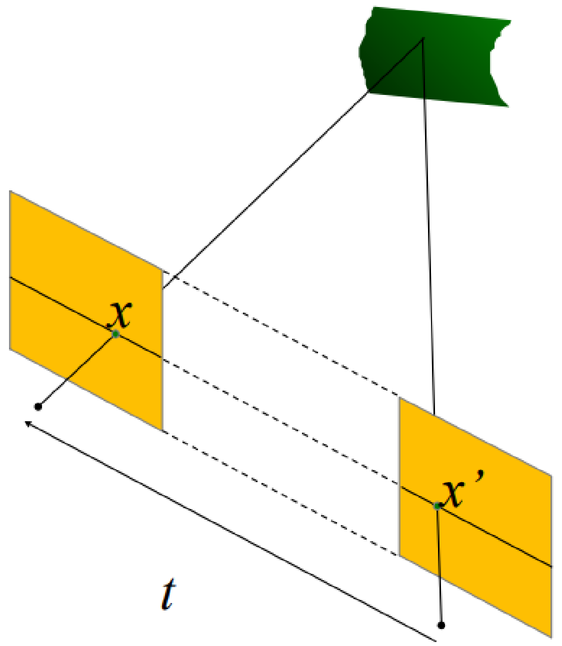
This always has to hold

$$x^T E x' = 0$$

Write out the constraint

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

How can you make the epipolar lines horizontal?



When this relationship holds:

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

Let's try this out…

$$E = t \times R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix}$$

This always has to hold

$$x^T E x' = 0$$

Write out the constraint

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

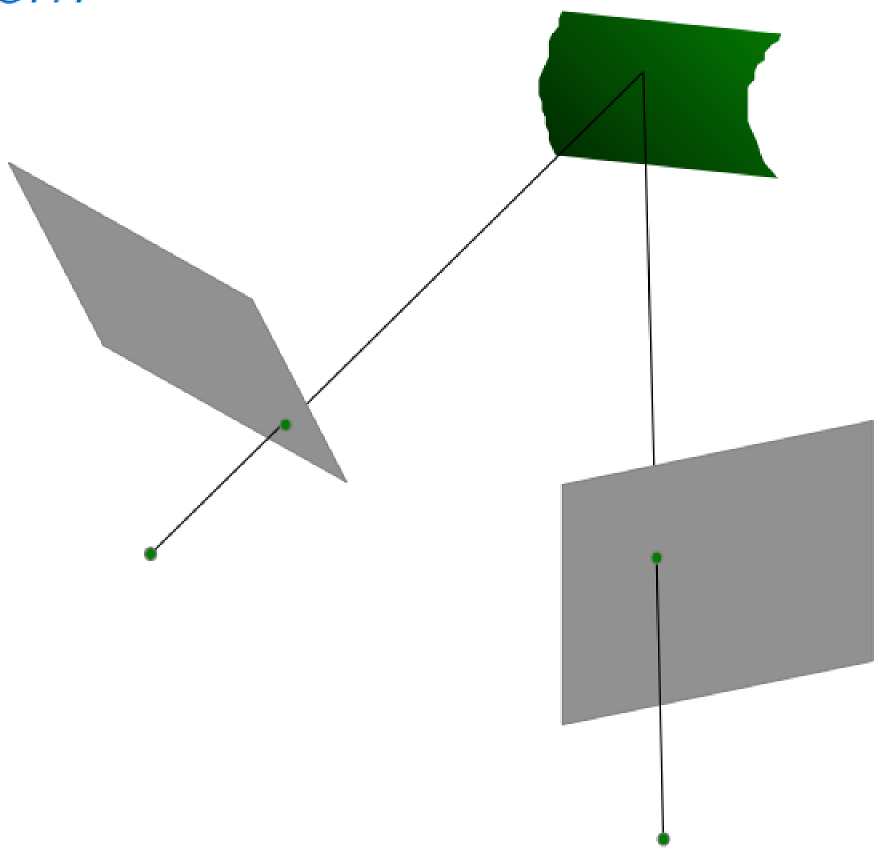
The image of a 3D point will always be on the same horizontal line

$$Tv = Tv'$$

y coordinate is always the same!

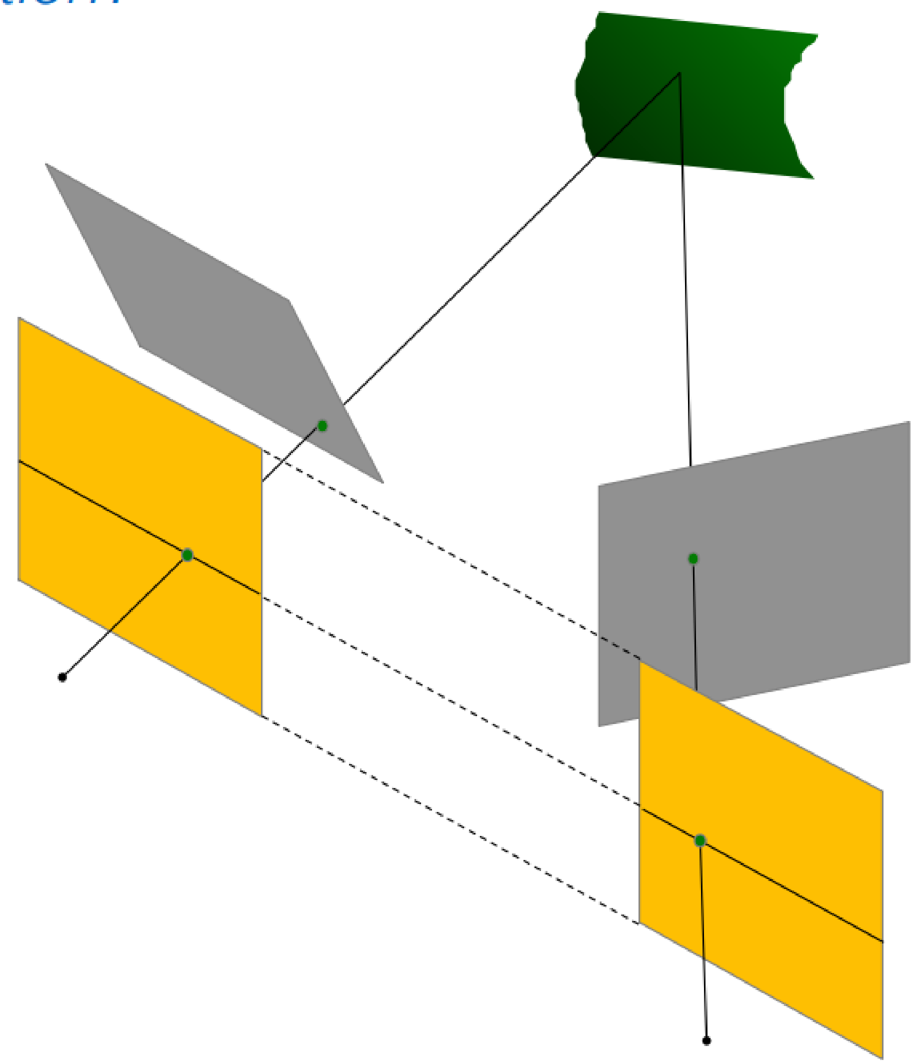
Implications?

What is stereo rectification?



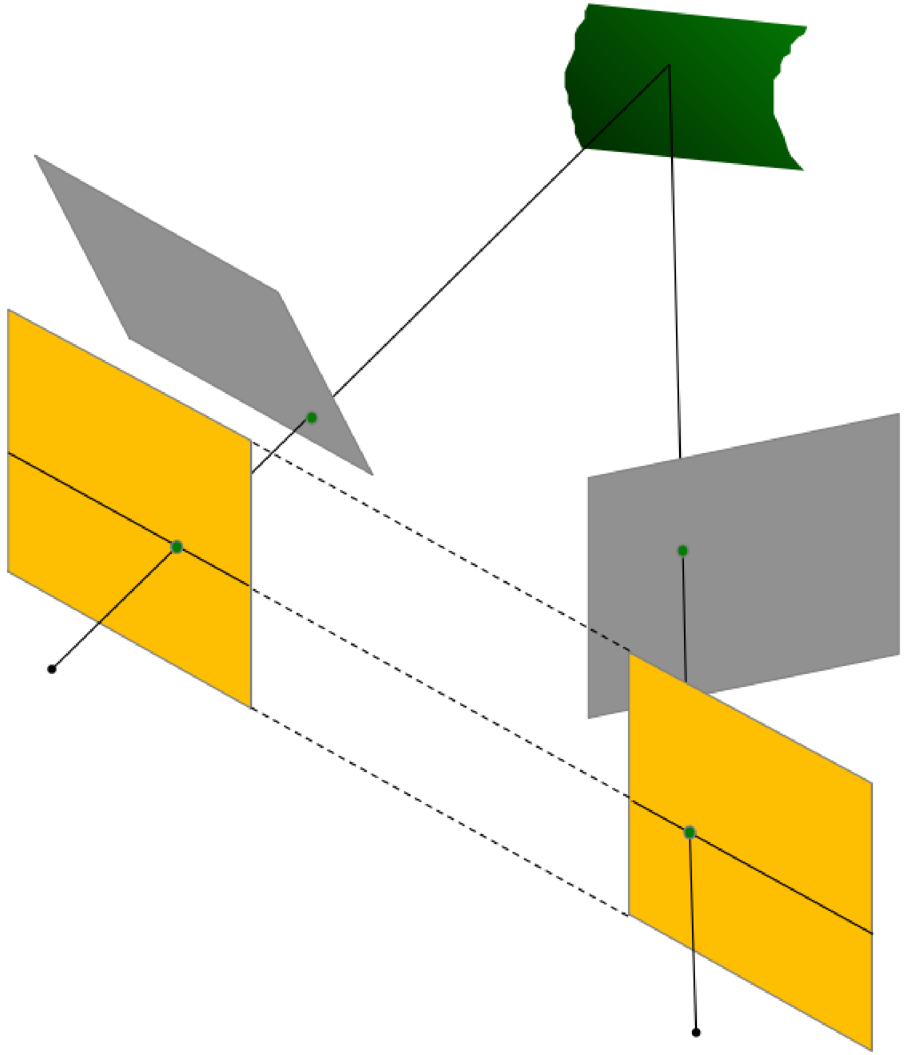
What is stereo rectification?

Reproject image planes onto a common plane parallel to the line between camera centers



What is stereo rectification?

Reproject image planes onto a common plane parallel to the line between camera centers



Need two homographies (3x3 transform), one for each input image reprojection

C. Loop and Z. Zhang. Computing Rectifying Homographies for Stereo Vision.Computer Vision and Pattern Recognition, 1999.

Stereo Rectification

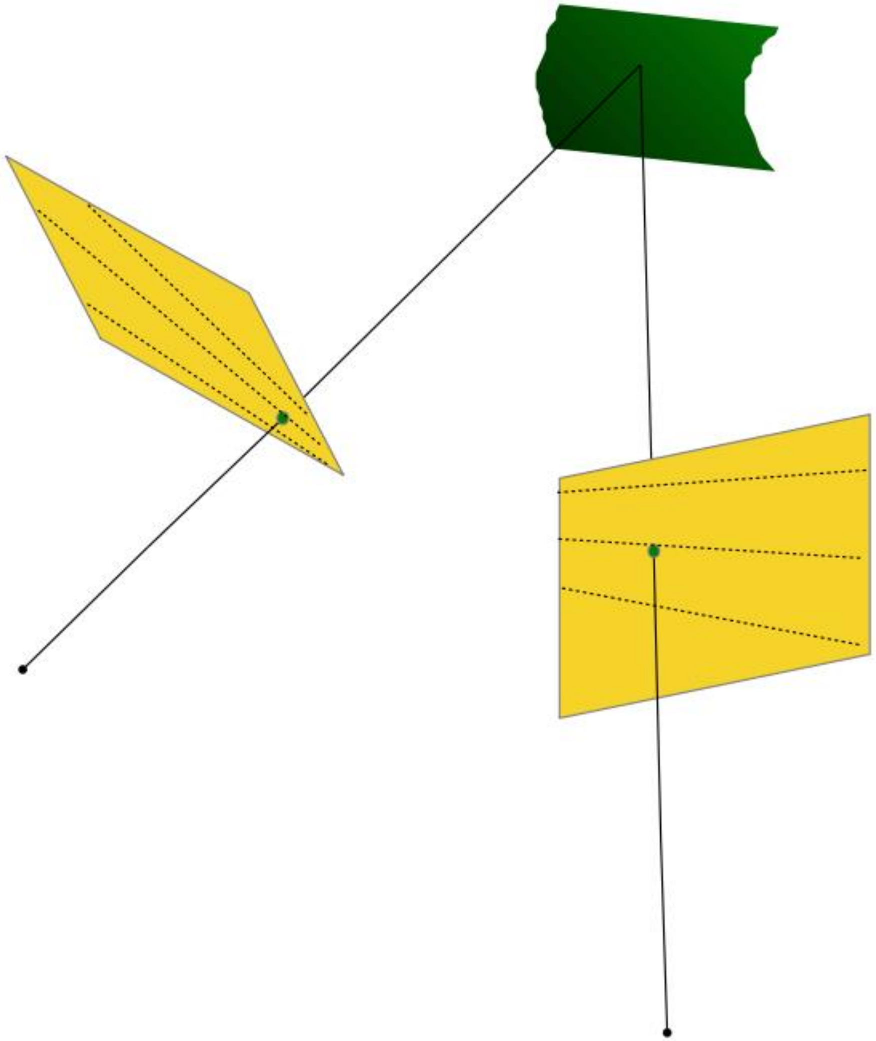
1. **Rotate** the right camera by **R**
   (aligns camera coordinate system orientation only)

2. Rotate (**rectify**) the left camera so that the epipole is at infinity

3. Rotate (**rectify**) the right camera so that the epipole is at infinity

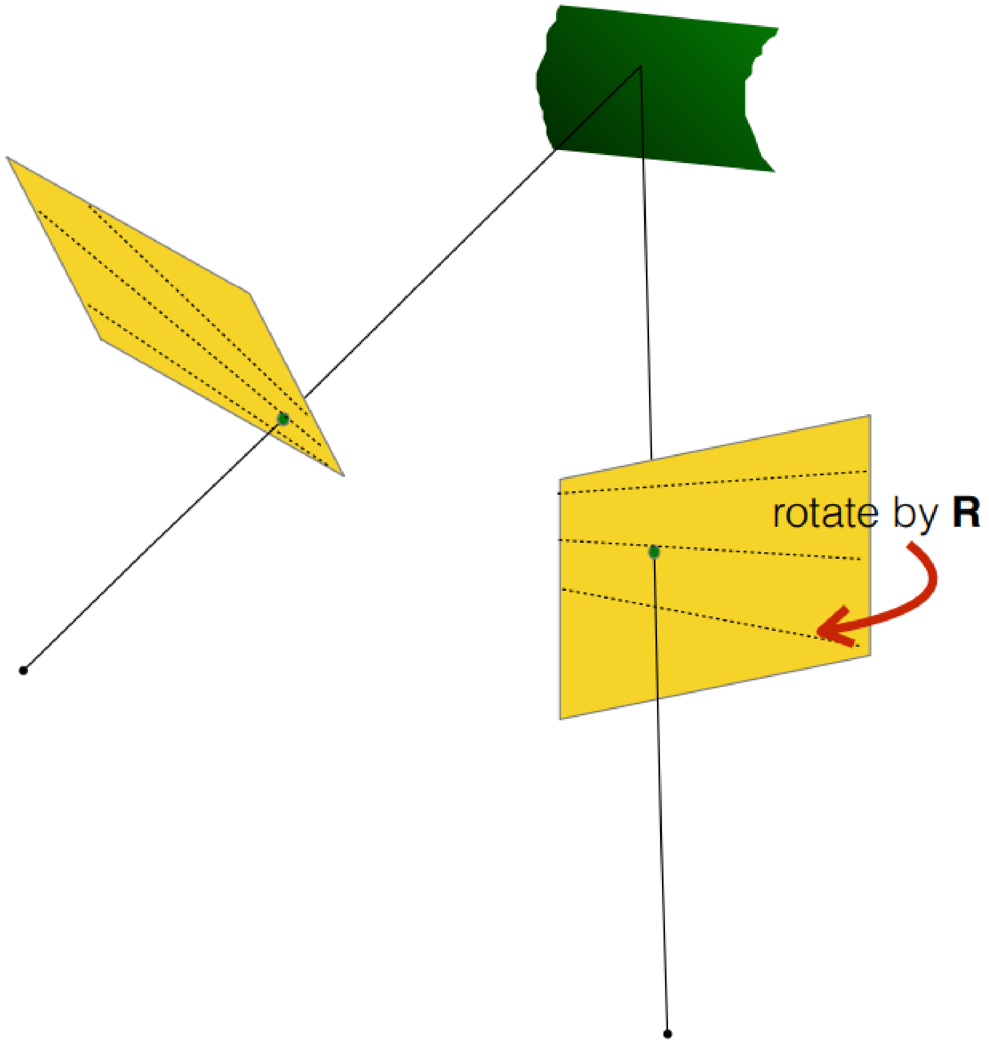
4. Adjust the **scale**

Stereo Rectification:



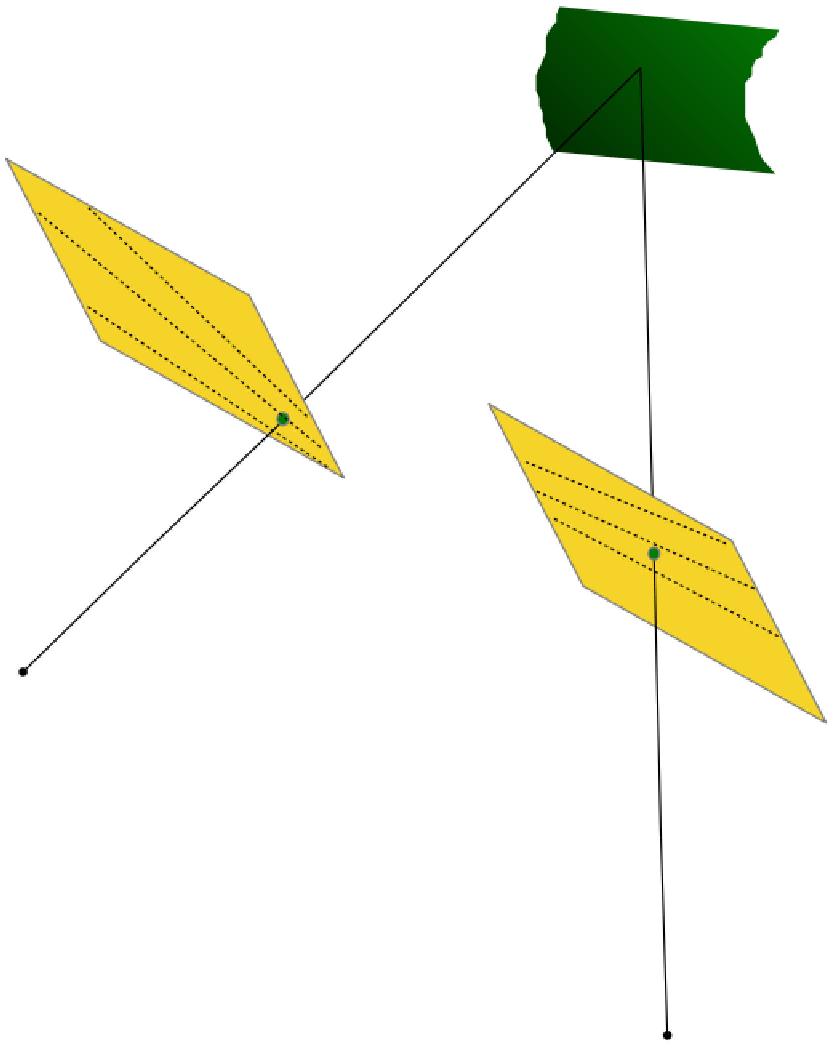
1. Compute **E** to get **R**
2. Rotate right image by **R**
3. Rotate both images by $\mathbf{R}_{\mathbf{rect}}$
4. Scale both images by **H**

Stereo Rectification:



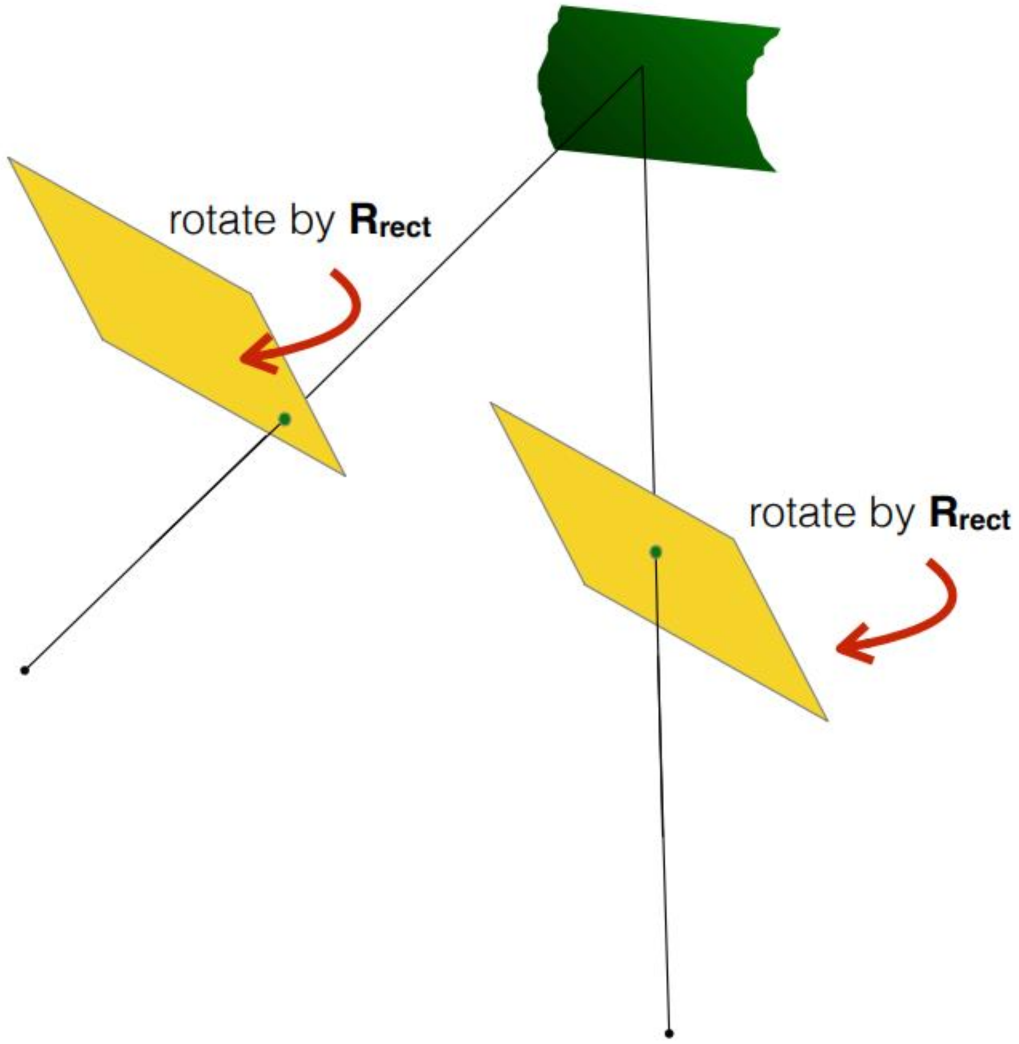
1. Compute **E** to get **R**
2. Rotate right image by **R**
3. Rotate both images by $\mathbf{R}_{\mathbf{rect}}$
4. Scale both images by **H**

Stereo Rectification:



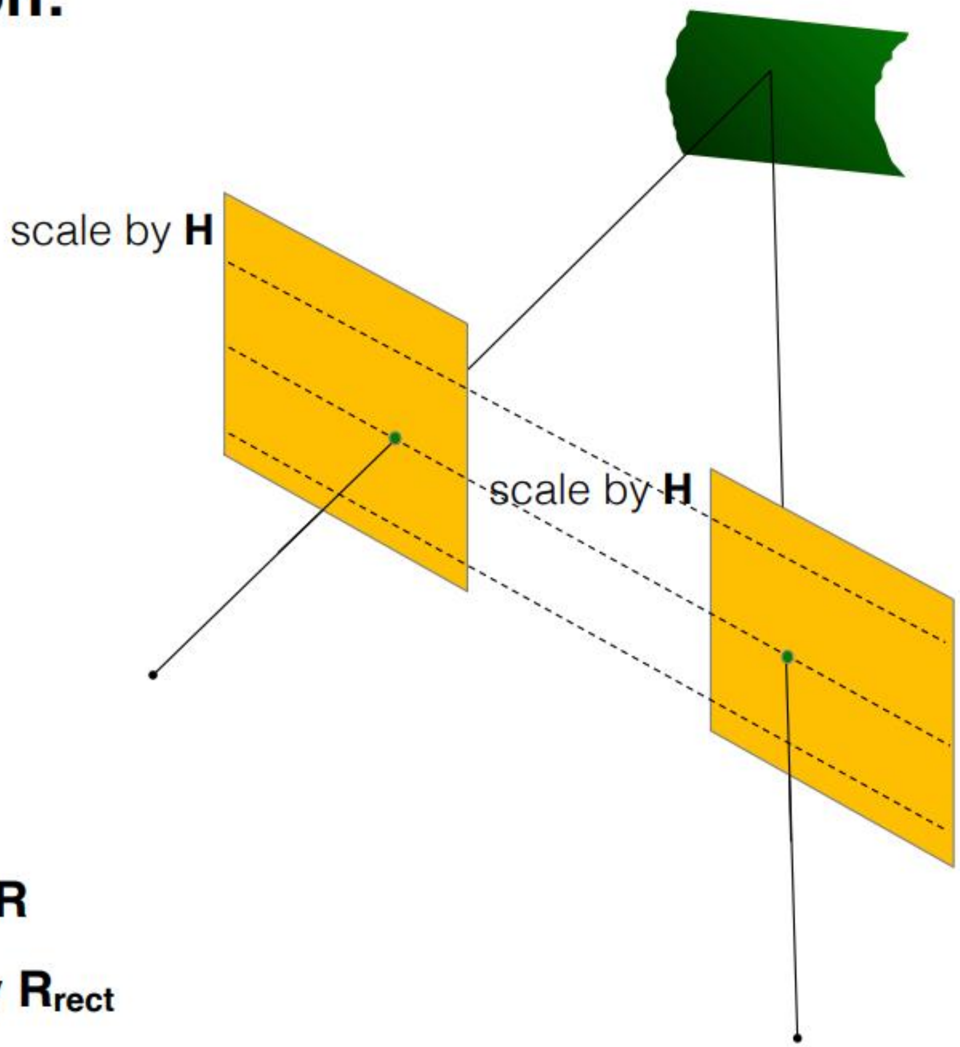
1. Compute **E** to get **R**
2. Rotate right image by **R**
3. Rotate both images by $\mathbf{R}_{\text{rect}}$
4. Scale both images by **H**

Stereo Rectification:



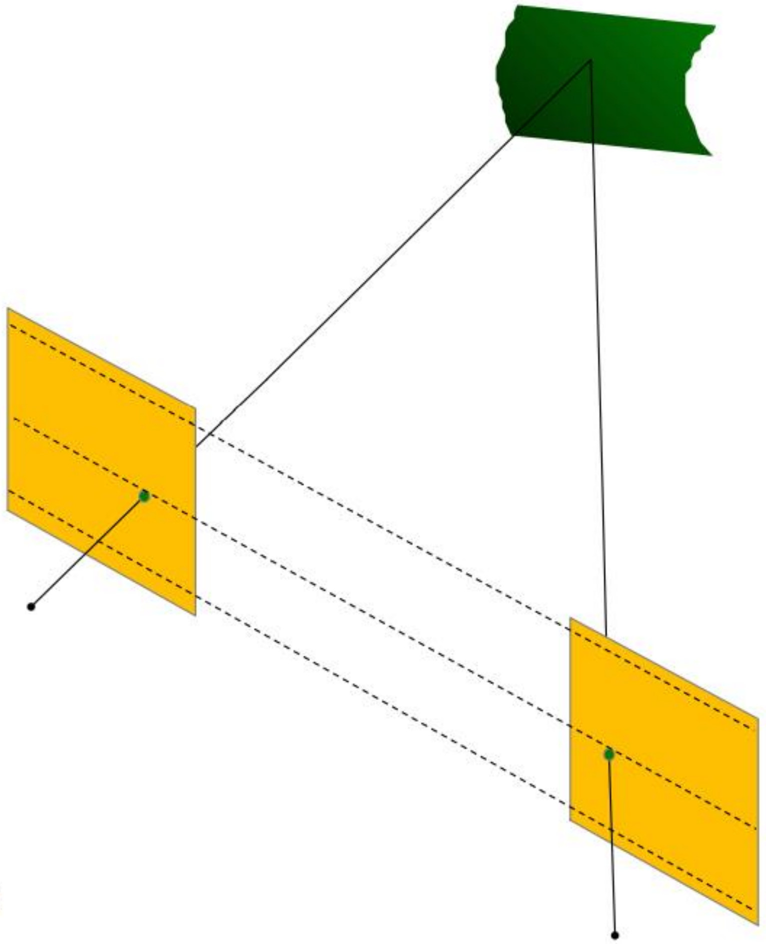
1. Compute **E** to get **R**
2. Rotate right image by **R**
3. Rotate both images by $\mathbf{R}_{\mathbf{rect}}$
4. Scale both images by **H**

Stereo Rectification:



1. Compute **E** to get **R**
2. Rotate right image by **R**
3. Rotate both images by $\mathbf{R}_{\mathbf{rect}}$
4. Scale both images by **H**

Stereo Rectification:



1. Compute **E** to get **R**
2. Rotate right image by **R**
3. Rotate both images by $\mathbf{R}_{\mathbf{rect}}$
4. Scale both images by **H**

We know the equation of the 1D search space

We match nxn patches between images

Drawbacks?

- Cameras need sufficient light
- Textureless regions in images (car door)
- We can use Textured light, infrared light
- Limited Range, Poor outdoor performance
- Relatively low range compared to LiDAR/RADAR

But wait… have I been lying?

- Portrait Mode – masking foreground
- How? Using depth!
- Stereo setup to extract depth - HTC One M8
- Pixel 2 with one lens able to achieve this bokeh

Enter Deep Learning! Ref: [Learning Single Camera Depth Estimation Using Dual-Pixels (thecvf.com)]