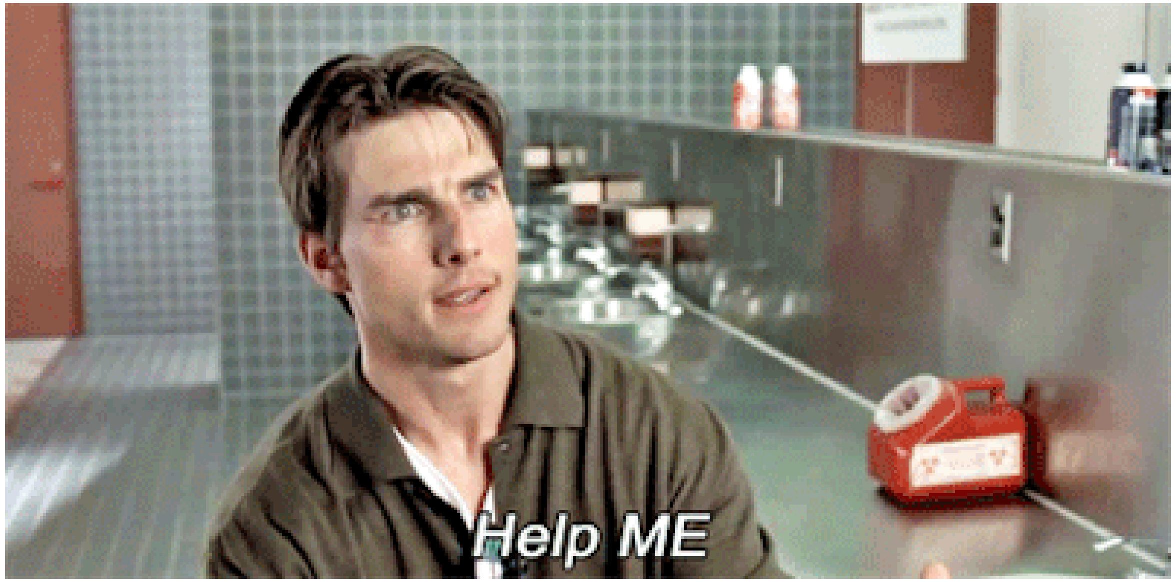
HW 3 Progress



CMSC 491/691

Lecture 15

Stereo Vision

(part of) HW3

- A married couple discovered a photo of themselves from 11 years before they met. Xue and her now-husband Ye were photographed together in 2000 as teenagers, but they only found out about it after getting married!

- In the summer of 2000, they both visited May Fourth Square in Qingdao, China. Several years later, while going through photos of a younger Xue to compare her resemblance to their daughters, Ye stumbled upon the picture.

- As soon as Ye saw the photo, he instantly recognized himself. He recalled, "I remember her mentioning that she had been to Qingdao, and coincidentally, I had also visited Qingdao and taken pictures at the ***May Fourth*** Square. When I saw the photo, I was completely surprised, and I got goosebumps all over my body... it was the exact pose I used for taking photos. I even took a picture from a different angle but in the same posture."

Recap: Camera Matrix : Intrinsic and Extrinsic Parameters

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$$

$$\mathbf{P} = \left[\begin{array}{ccc} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{array}\right] \left[\begin{array}{ccc|c} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \end{array}\right]$$

intrinsic parameters

extrinsic parameters

$$\mathbf{R} = \left[\begin{array}{ccc} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{array}\right] \qquad \mathbf{t} = \left[\begin{array}{c} t_1 \\ t_2 \\ t_3 \end{array}\right]$$

3D rotation

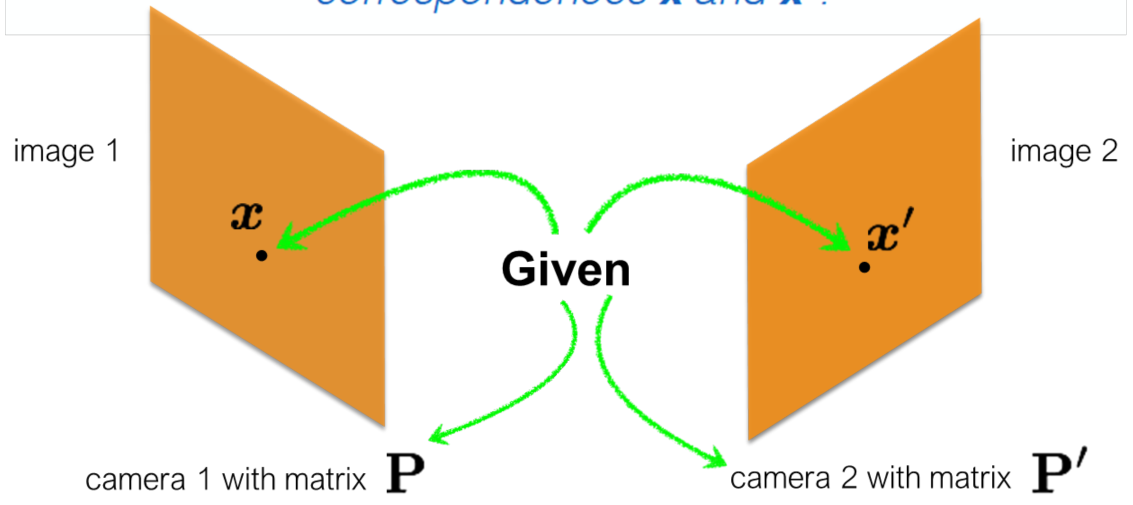
3D translation

Recap: Triangulation and Epipolar Geometry

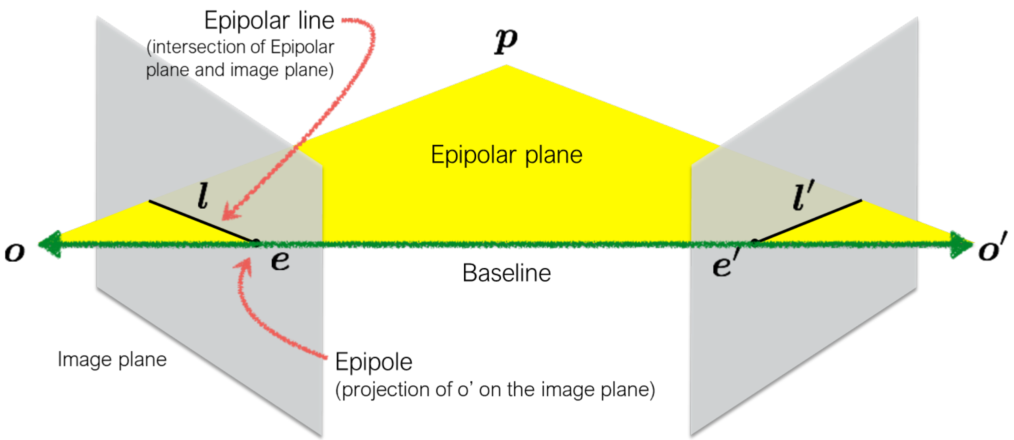
$$\mathbf{x} = \mathbf{P}X$$

known known

*Can we compute **X** from <u>two</u> correspondences **x** and **x'**?*



Epipolar geometry



Longuet-Higgins equation	$\boldsymbol{x}'^\top \mathbf{E}\boldsymbol{x} = 0$	
Epipolar lines	$\boldsymbol{x}^\top \boldsymbol{l} = 0$ $\boldsymbol{l}' = \mathbf{E}\boldsymbol{x}$	$\boldsymbol{x}'^\top \boldsymbol{l}' = 0$ $\boldsymbol{l} = \mathbf{E}^T\boldsymbol{x}'$
Epipoles	$\boldsymbol{e}'^\top \mathbf{E} = \mathbf{0}$	$\mathbf{E}\boldsymbol{e} = \mathbf{0}$

(points in normalized <u>camera</u> coordinates)

Essential Matrix vs Homography

What's the difference between the essential matrix and a homography?

They are both 3 x 3 matrices but …

$$\boldsymbol{l}' = \mathbf{E}\boldsymbol{x}$$

Essential matrix maps a **point** to a **line**

$$\boldsymbol{x}' = \mathbf{H}\boldsymbol{x}$$

Homography maps a **point** to a **point**

The fundamental matrix

The **Fundamental matrix**

is a **generalization**

of the **Essential matrix,**

where the assumption of **calibrated cameras**

is removed

Same equation works in image coordinates!

$$\boldsymbol{x}'^{\top}\mathbf{F}\boldsymbol{x} = 0$$

it maps pixels to epipolar lines

The 8-point algorithm

Assume you have ***M*** matched *image* points

$$\{\boldsymbol{x}_m, \boldsymbol{x}'_m\} \qquad m = 1, \ldots, M$$

Each correspondence should satisfy

$$\boldsymbol{x}'^{\top}_m \mathbf{F} \boldsymbol{x}_m = 0$$

How would you solve for the 3 x 3 ***F*** *matrix?*

$$\boldsymbol{x}_m'^\top \mathbf{F} \boldsymbol{x}_m = 0$$

$$\begin{bmatrix} x_m' & y_m' & 1 \end{bmatrix} \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \begin{bmatrix} x_m \\ y_m \\ 1 \end{bmatrix} = 0$$

How many equation do you get from one correspondence?

$$\begin{bmatrix} x'_m & y'_m & 1 \end{bmatrix} \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \begin{bmatrix} x_m \\ y_m \\ 1 \end{bmatrix} = 0$$

ONE correspondence gives you ONE equation

$$\begin{aligned} x_m x'_m f_1 + x_m y'_m f_2 + x_m f_3 + \\ y_m x'_m f_4 + y_m y'_m f_5 + y_m f_6 + \\ x'_m f_7 + y'_m f_8 + f_9 = 0 \end{aligned}$$

$$\begin{bmatrix} x'_m & y'_m & 1 \end{bmatrix} \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \begin{bmatrix} x_m \\ y_m \\ 1 \end{bmatrix} = 0$$

Set up a homogeneous linear system with 9 unknowns

$$\begin{bmatrix} x_1x'_1 & x_1y'_1 & x_1 & y_1x'_1 & y_1y'_1 & y_1 & x'_1 & y'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_Mx'_M & x_My'_M & x_M & y_Mx'_M & y_My'_M & y_M & x'_M & y'_M & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix} = \mathbf{0}$$

Each point pair (according to epipolar constraint) contributes only one <u>scalar</u> equation

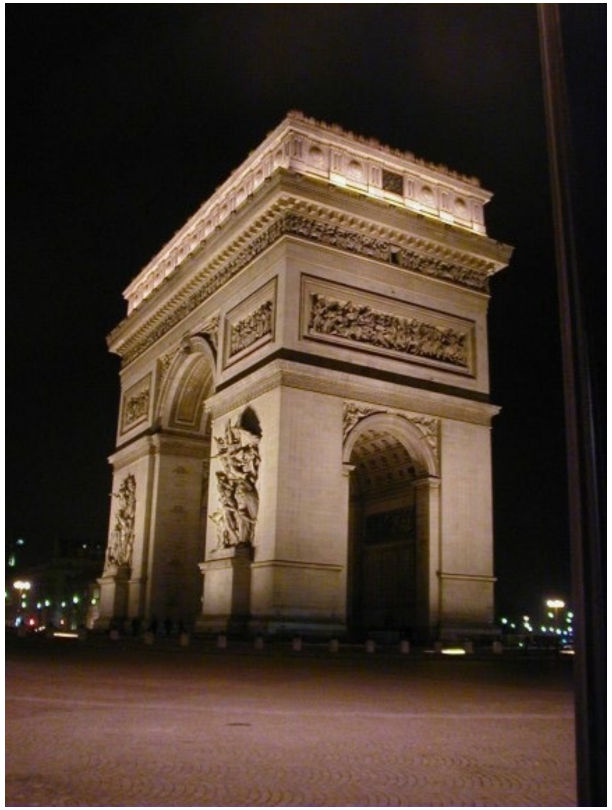
$$\boldsymbol{x}_m'^{\top} \mathbf{F} \boldsymbol{x}_m = 0$$

Note: This is different from the Homography estimation where each point pair contributes 2 equations.

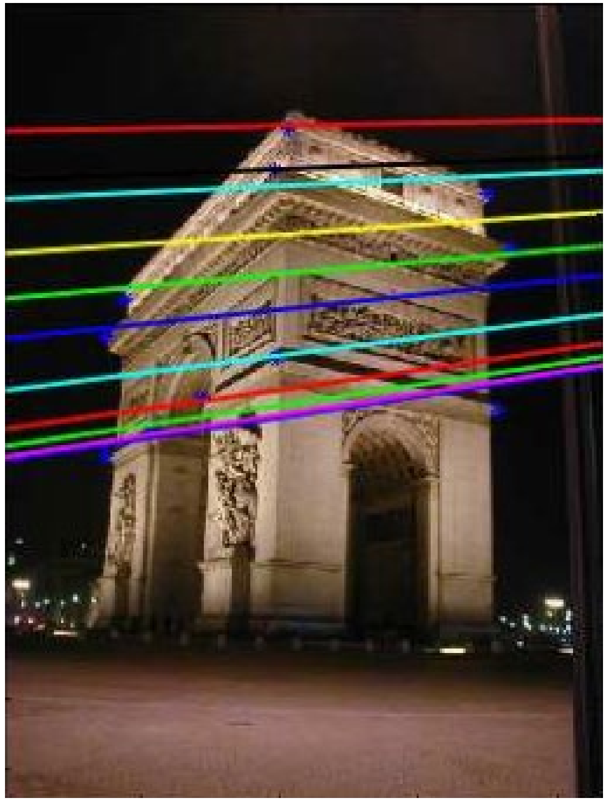
We need at least 8 points

Hence, the 8 point algorithm!

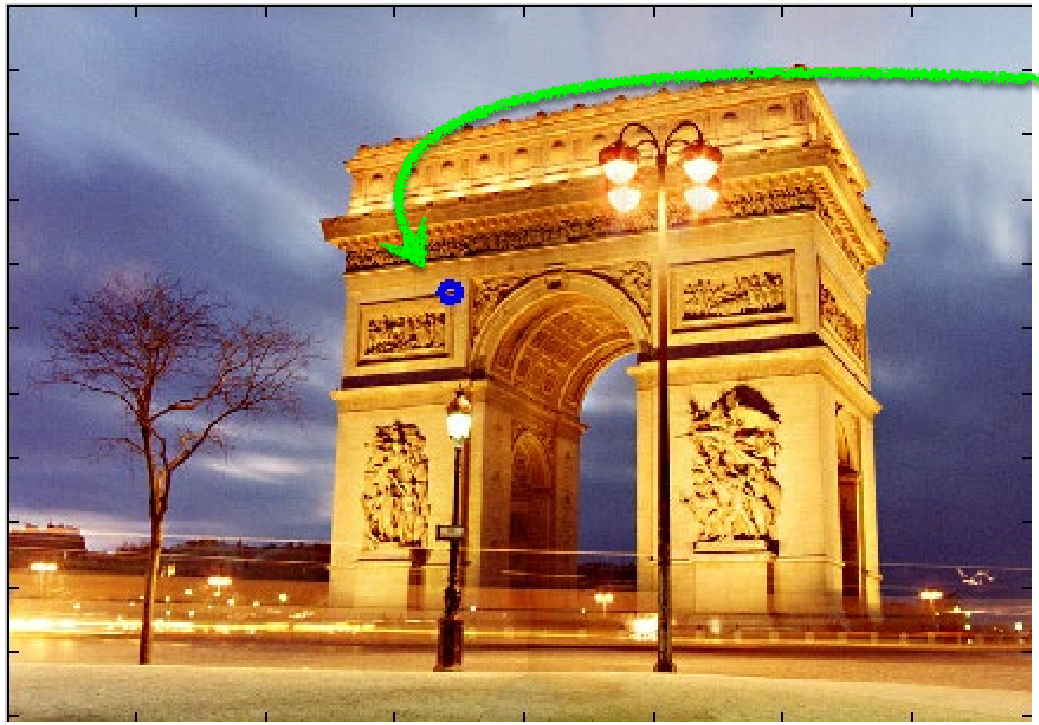
Example



epipolar lines



$$\mathbf{F} = \begin{bmatrix} -0.00310695 & -0.0025646 & 2.96584 \\ -0.028094 & -0.00771621 & 56.3813 \\ 13.1905 & -29.2007 & -9999.79 \end{bmatrix}$$



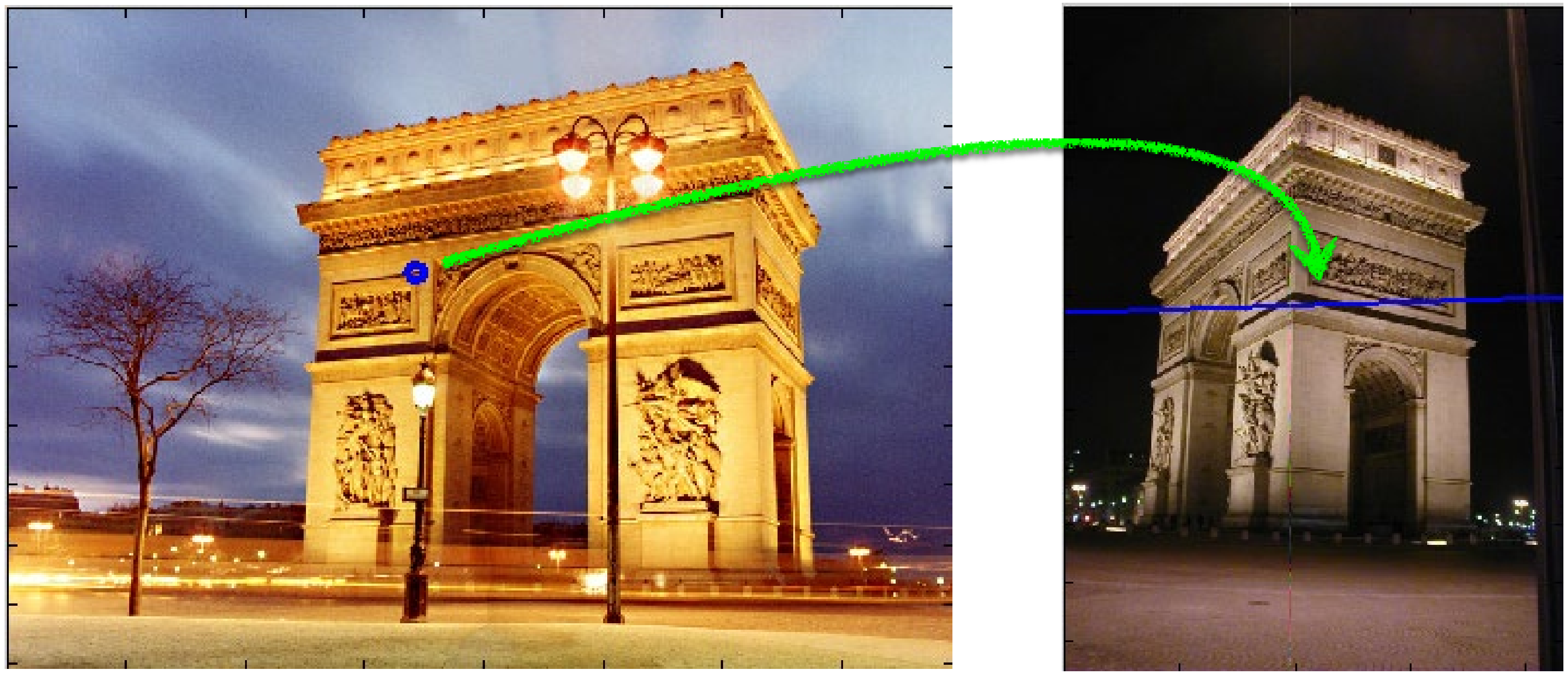
$$\boldsymbol{x} = \begin{bmatrix} 343.53 \\ 221.70 \\ 1.0 \end{bmatrix}$$

$$\boldsymbol{l}' = \mathbf{F}\boldsymbol{x}$$

$$= \begin{bmatrix} 0.0295 \\ 0.9996 \\ -265.1531 \end{bmatrix}$$

$$l' = \mathbf{F}\boldsymbol{x}$$

$$= \begin{bmatrix} 0.0295 \\ 0.9996 \\ -265.1531 \end{bmatrix}$$



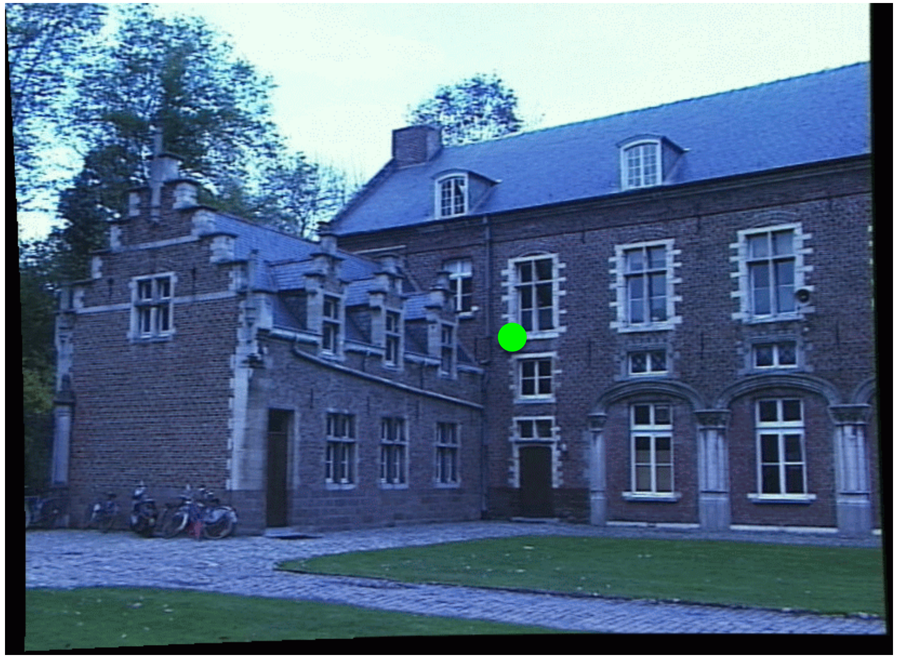
Stereo Imaging

How would you reconstruct 3D points?

Left image

Right image

How would you reconstruct 3D points?

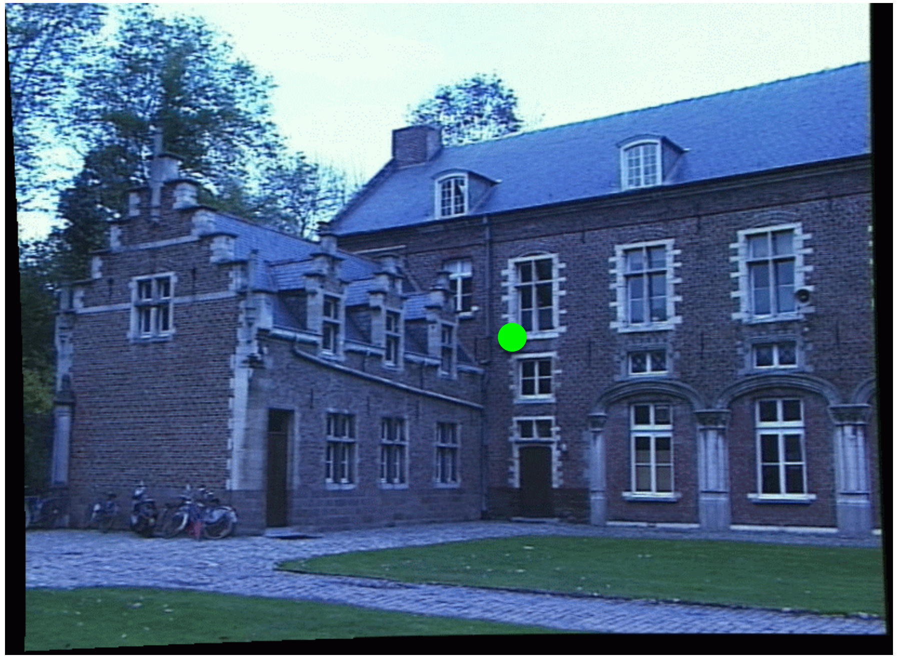


Left image

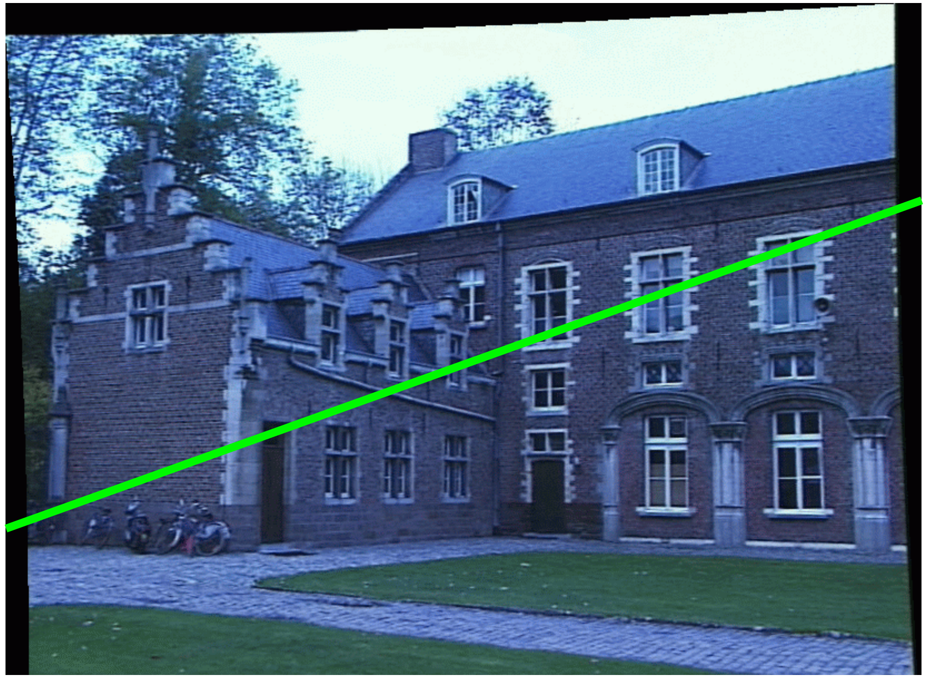
Right image

1. Select point in one image

How would you reconstruct 3D points?



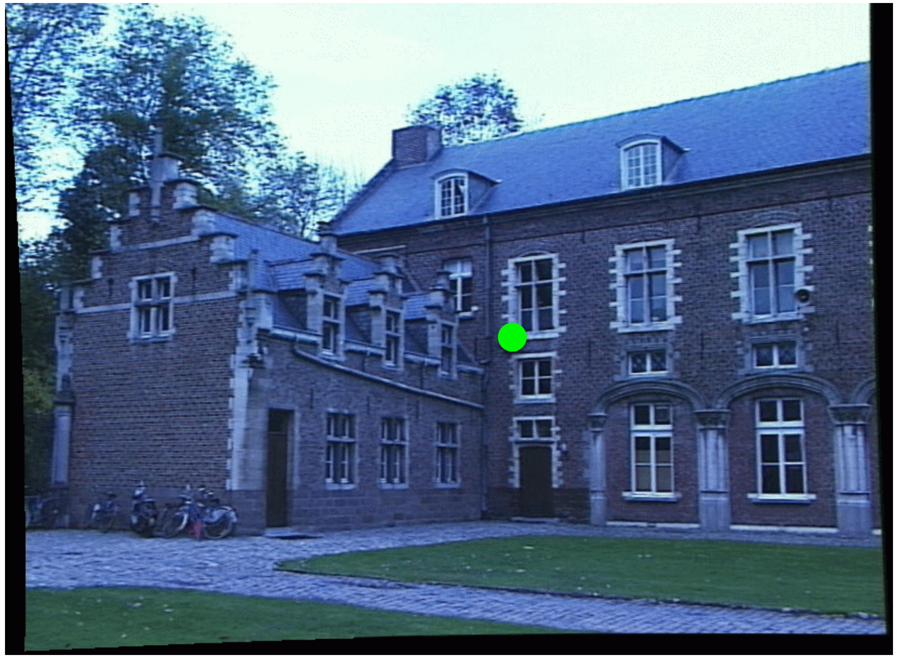
Left image



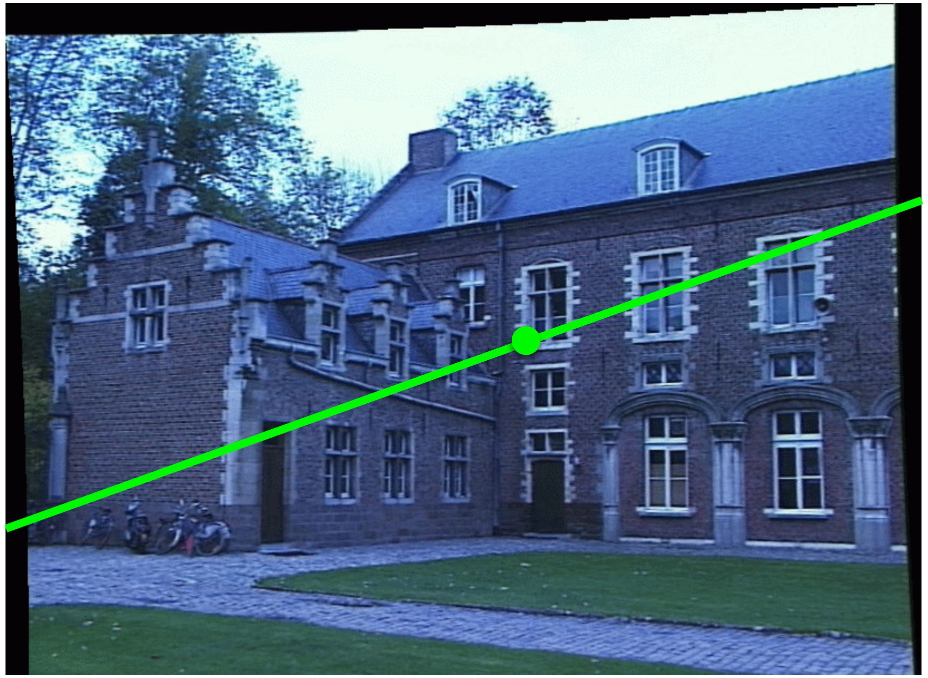
Right image

1. Select point in one image
2. Form the epipolar line for that point in second image

How would you reconstruct 3D points?



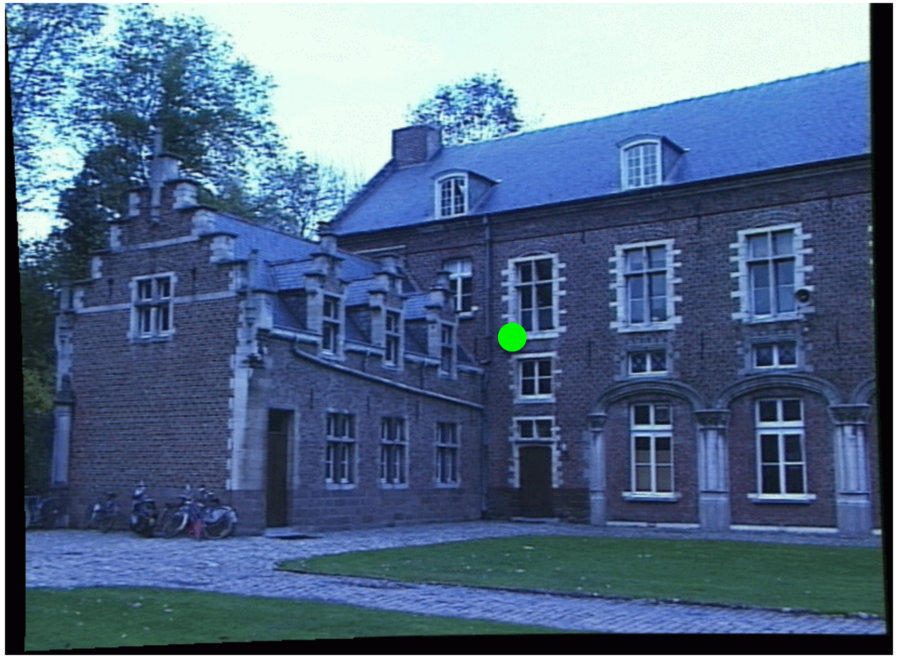
Left image



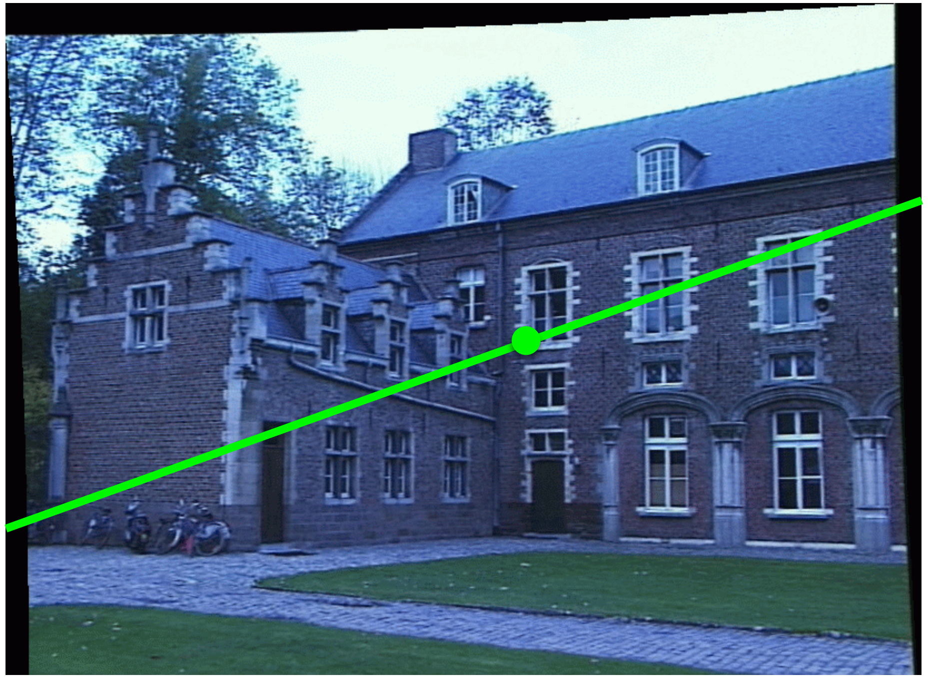
Right image

1. Select point in one image
2. Form the epipolar line for that point in second image
3. Find matching point along line

How would you reconstruct 3D points?



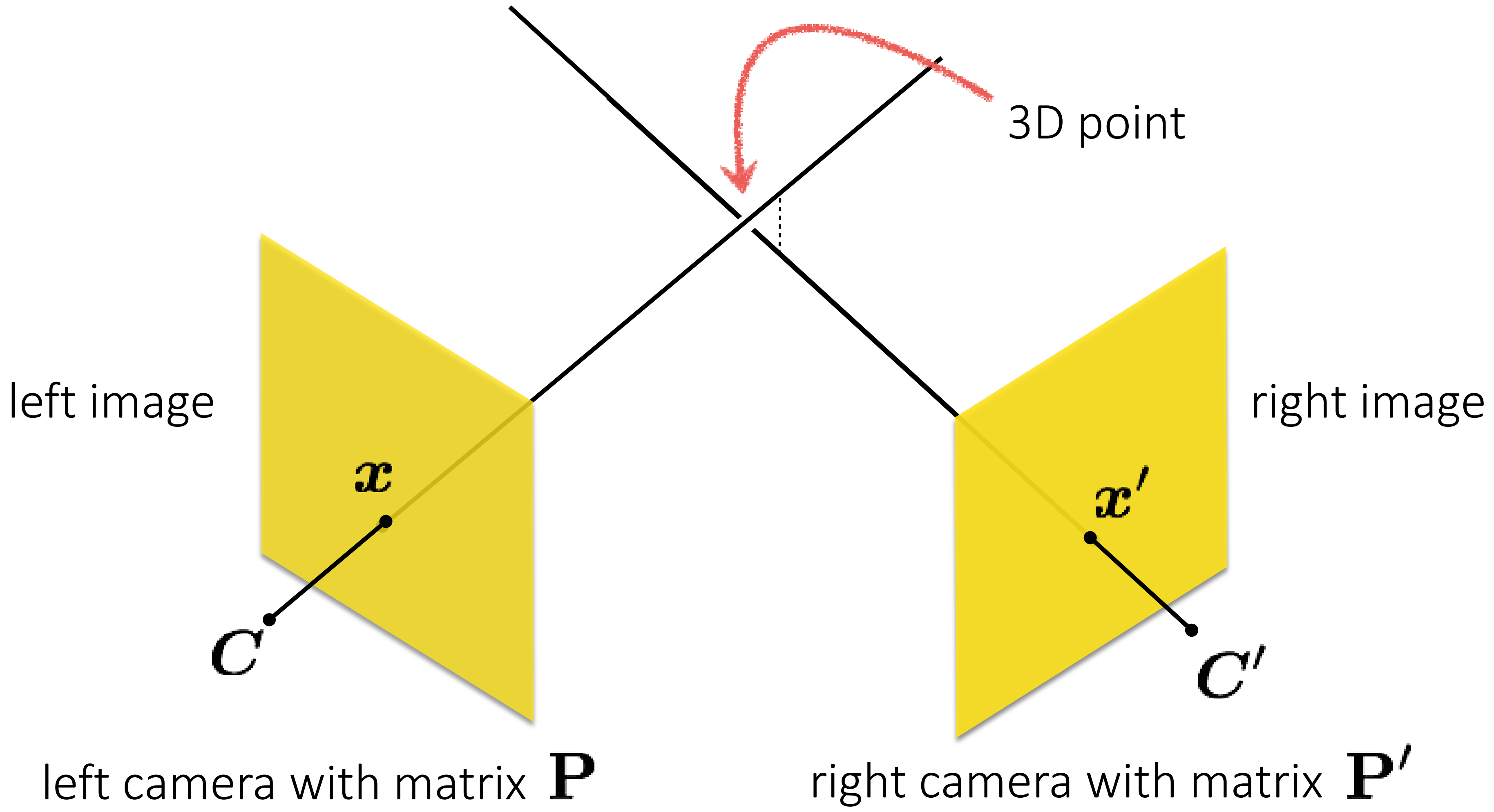
Left image



Right image

1. Select point in one image
2. Form the epipolar line for that point in second image
3. Find matching point along line
4. Perform triangulation

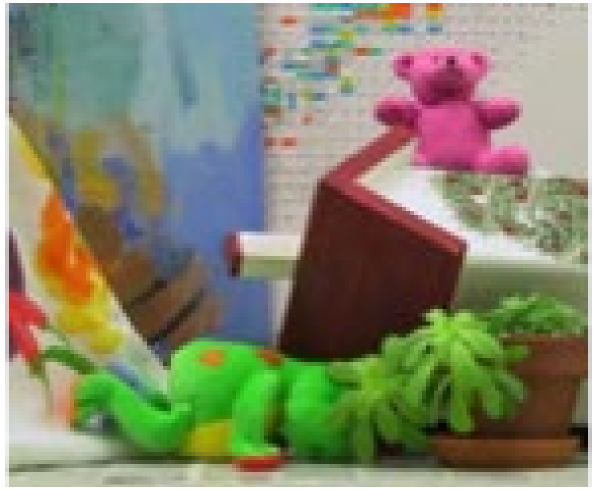
Triangulation

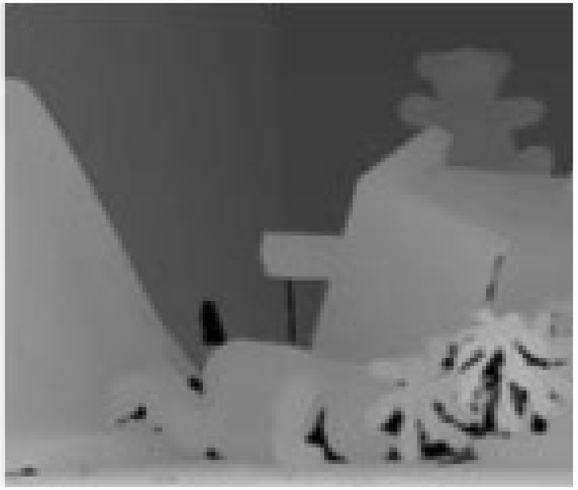


Stereo rectification

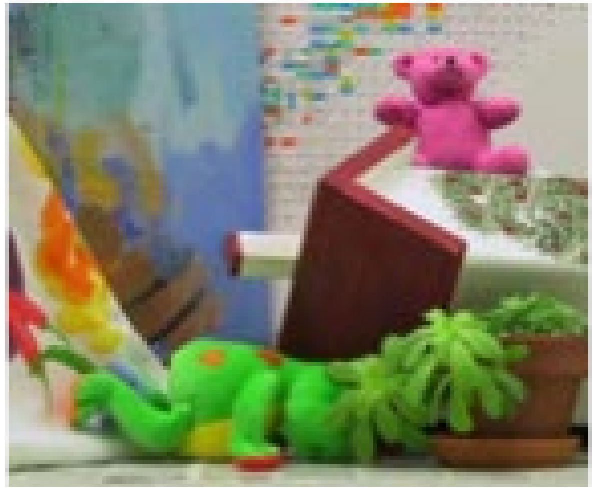
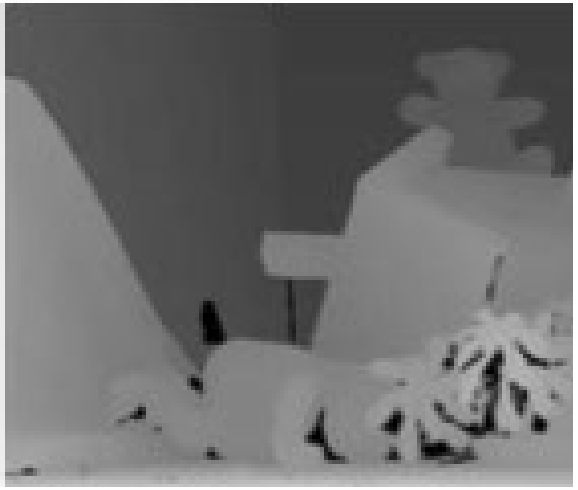
What's different between these two images?

The amount of horizontal movement is
inversely proportional to …



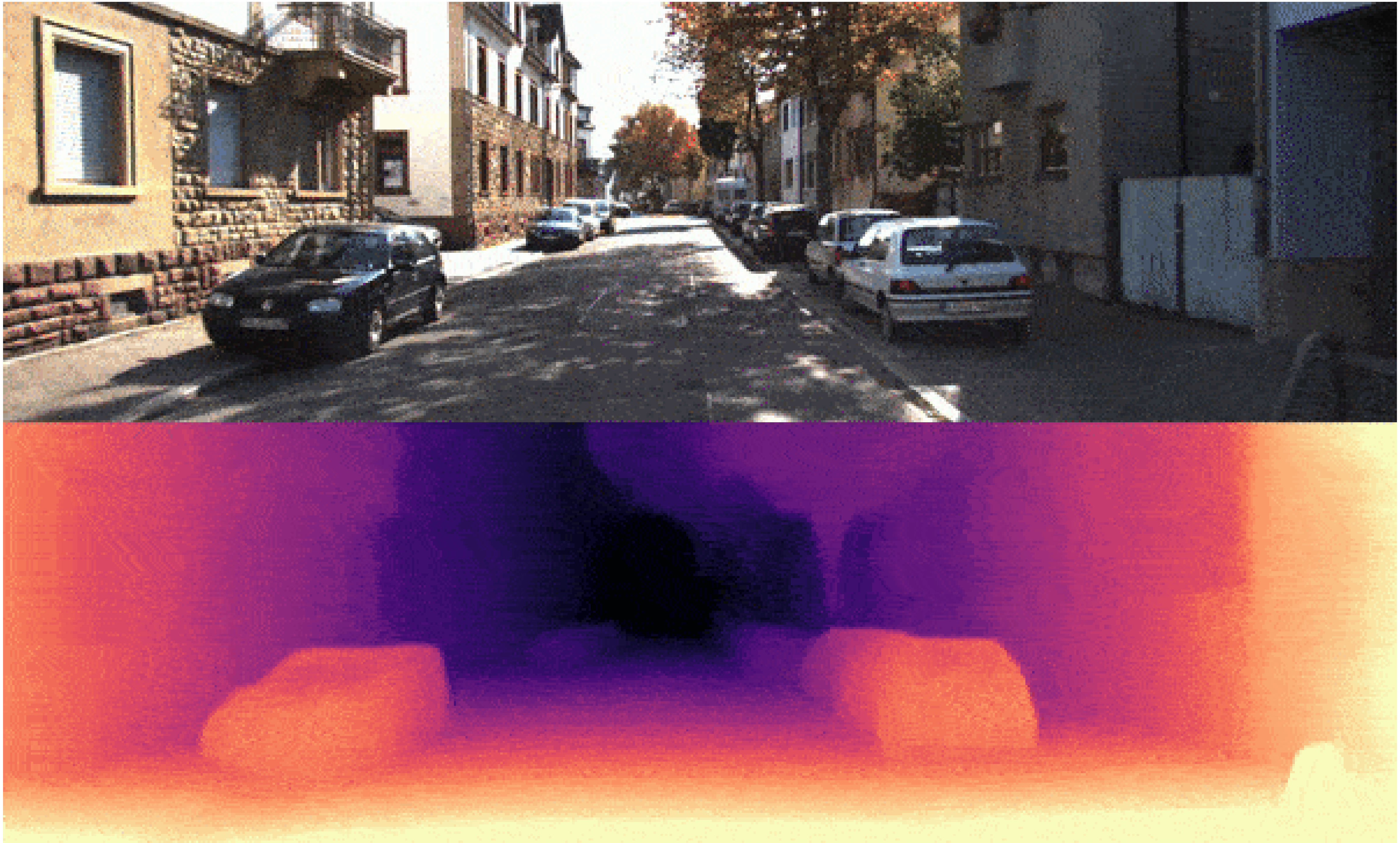


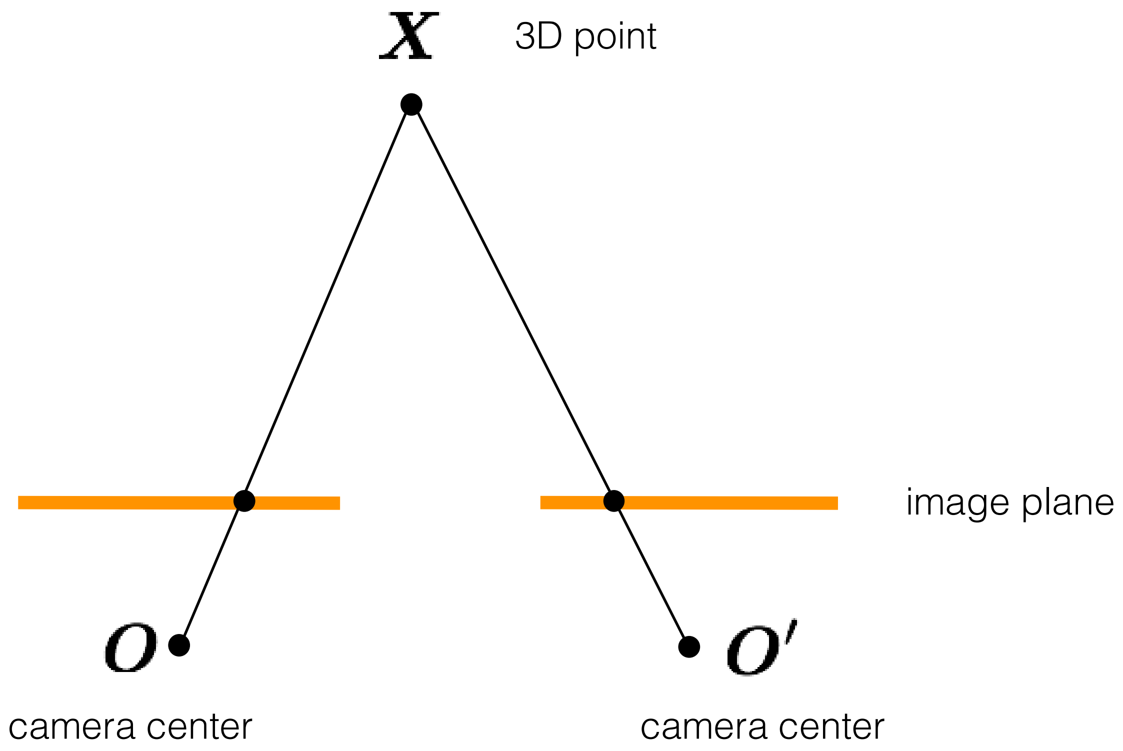
The amount of horizontal movement is inversely proportional to …

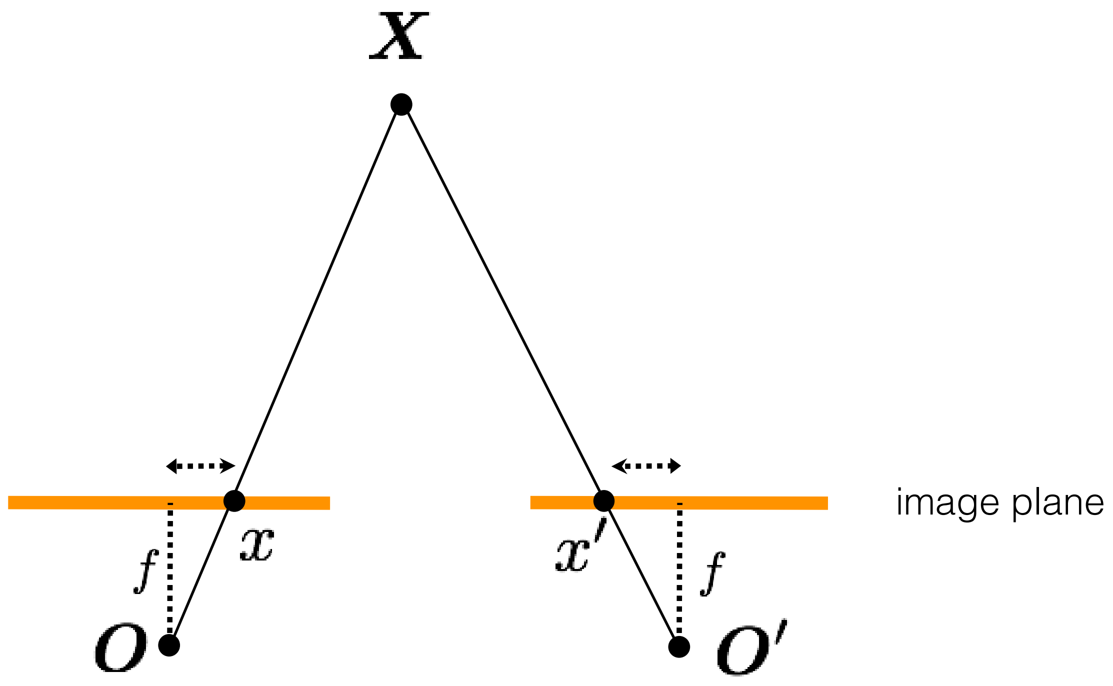




… the distance from the camera.

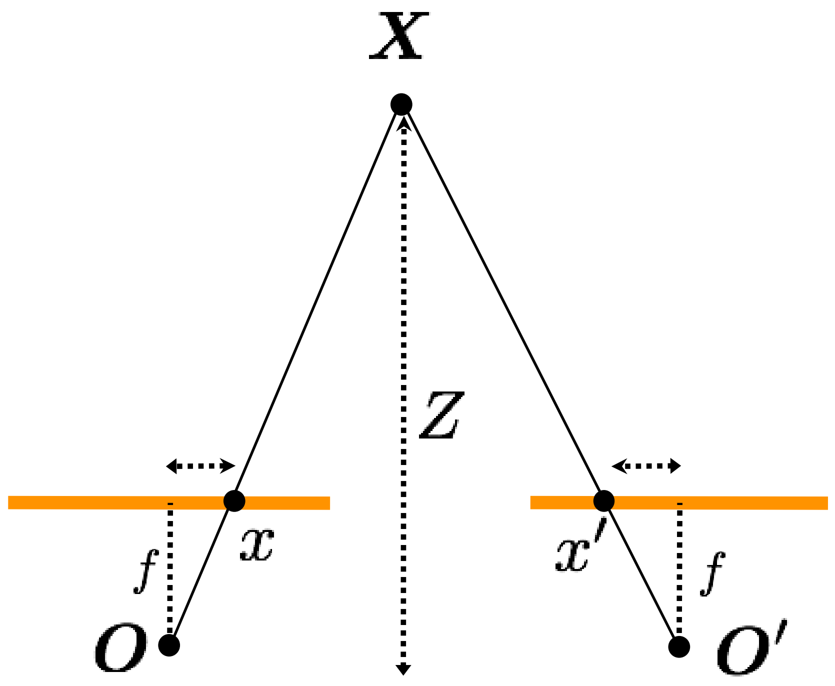
… aka … ***depth***

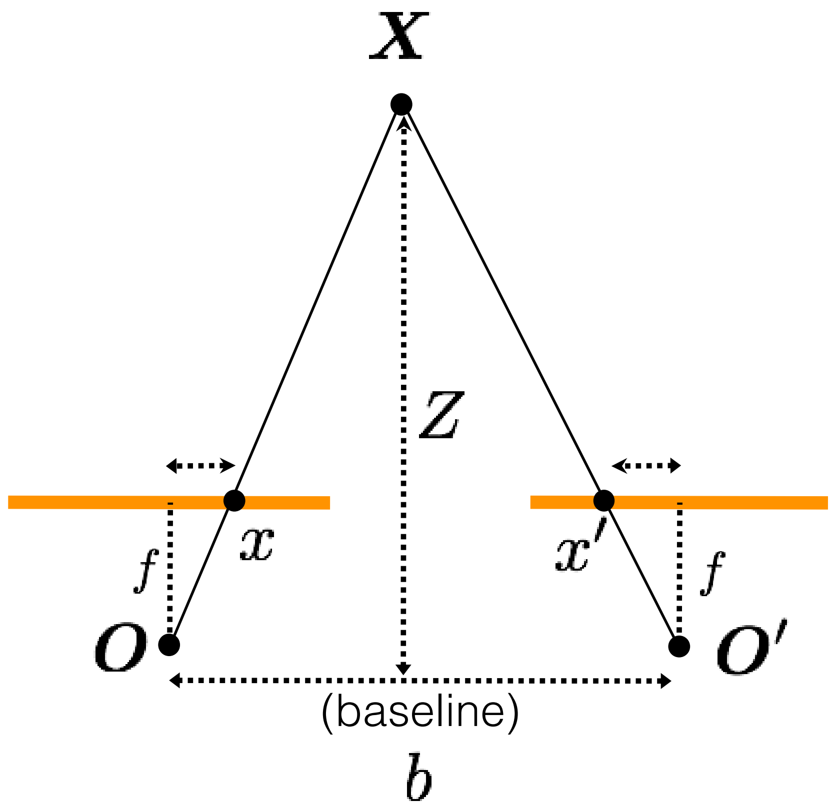



$\boldsymbol{X}$
3D point
image plane
$\boldsymbol{O}$
camera center
$\boldsymbol{O'}$
camera center



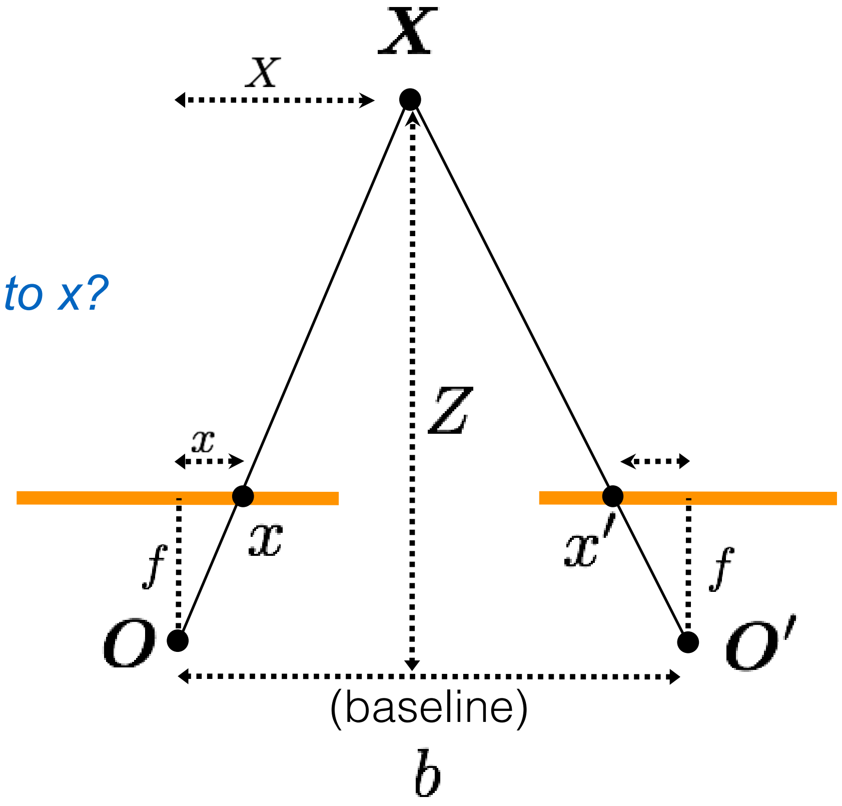
X
x
x'
f
f
O
O'
image plane


X
Z
x
x'
f
f
O
O'

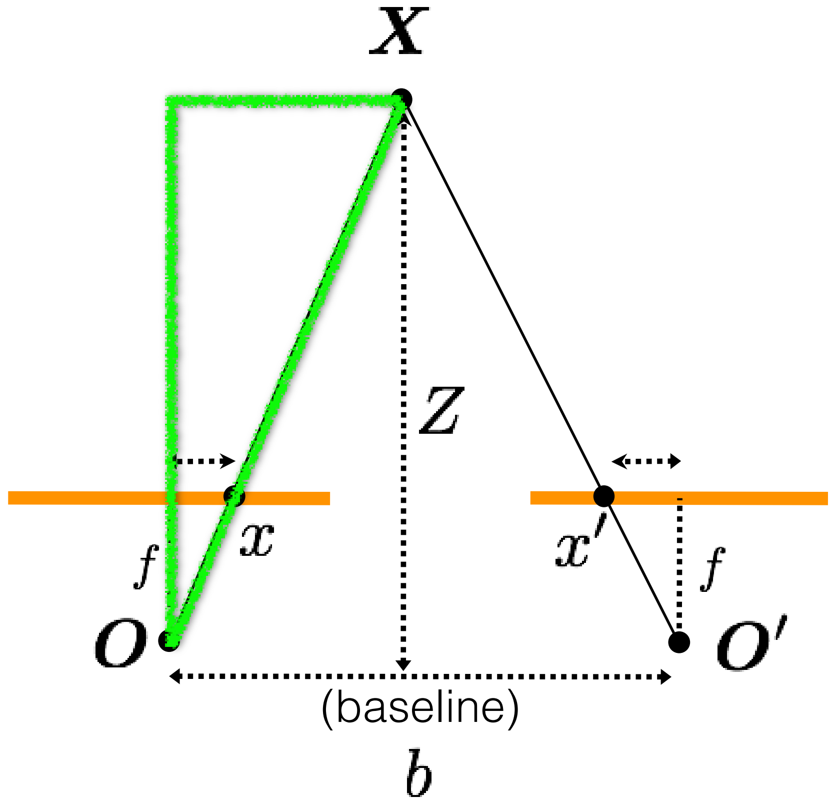


X
Z
x
x'
f
f
O
O'
(baseline)
b

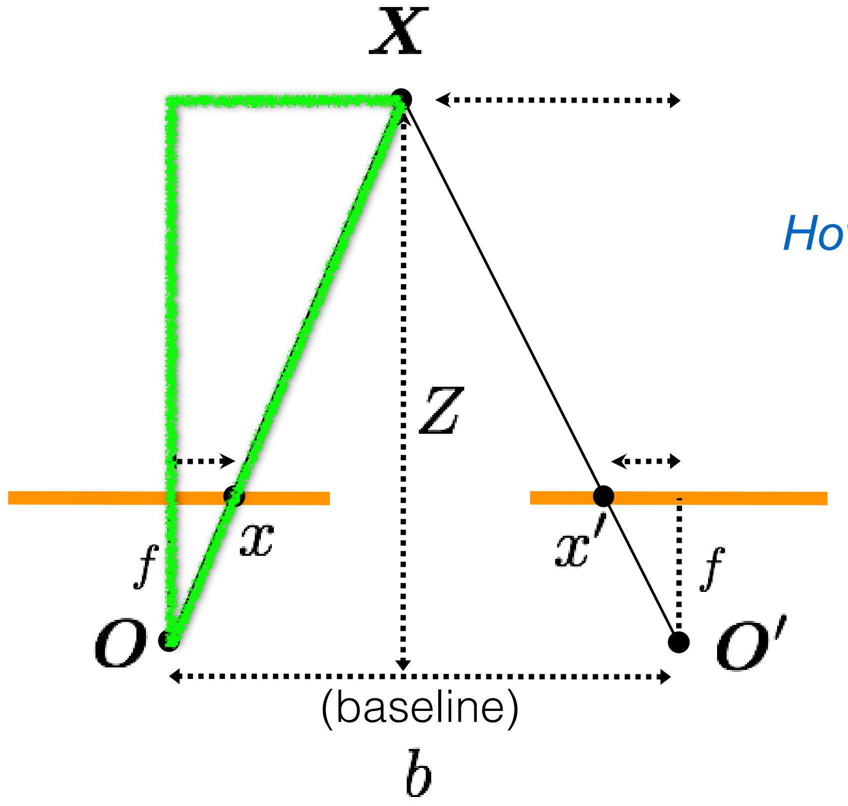
How is X related to x?



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

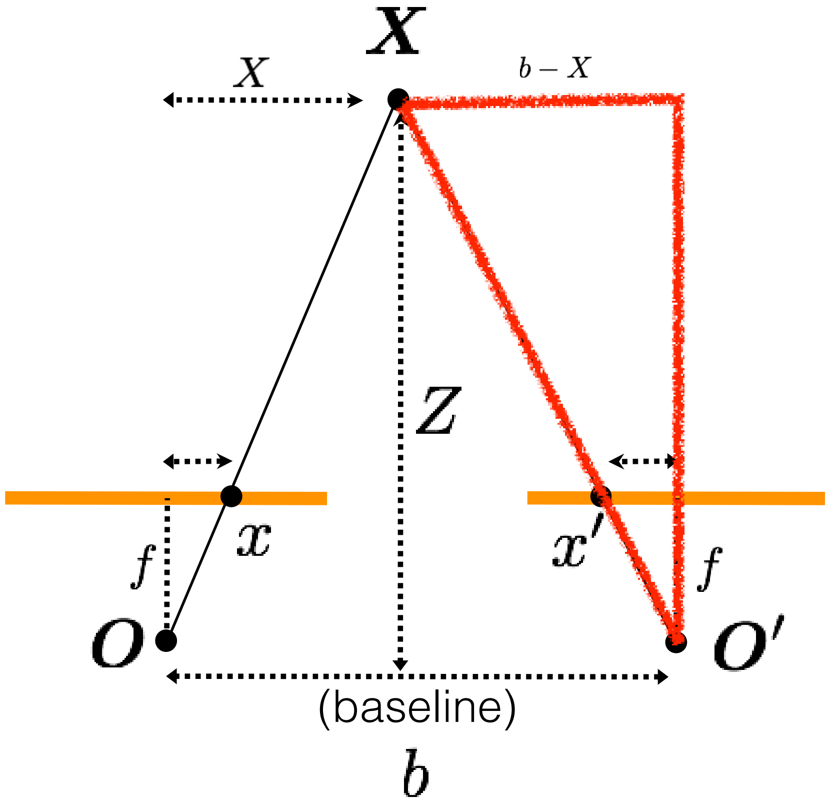


$$\frac{X}{Z} = \frac{x}{f}$$



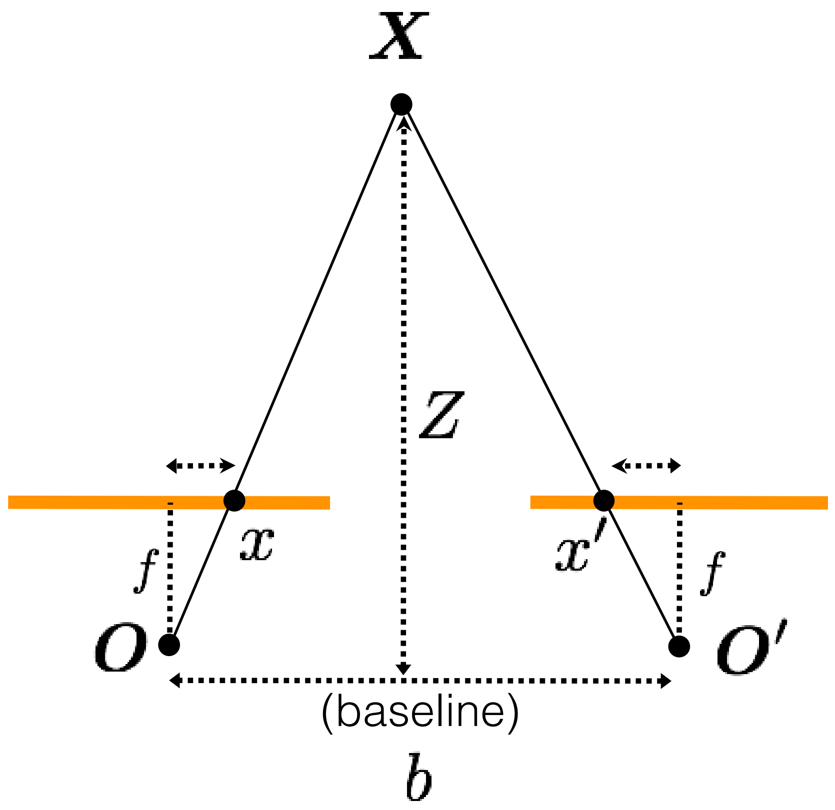
How is X related to x'?

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



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

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



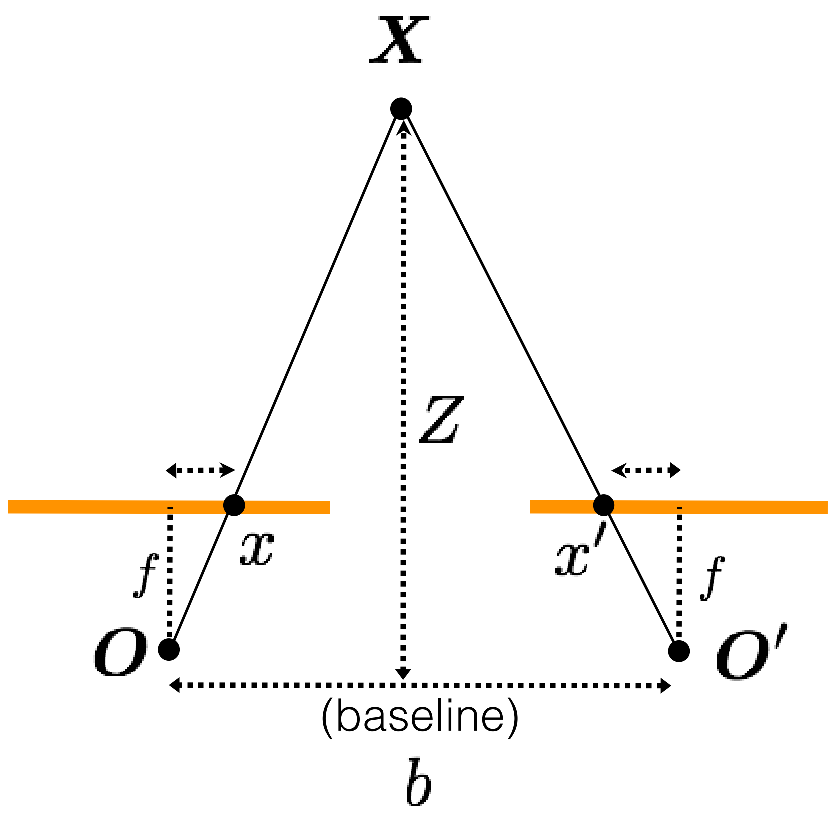
$$\frac{b - X}{Z} = \frac{x'}{f}$$

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

Stereoscopes: A 19th Century Pastime

Old **Zeiss** pocket stereoscope with original test image

A **stereoscope** is a device for viewing a stereoscopic pair of separate images, depicting left-eye and right-eye views of the same scene, as a single three-dimensional image.

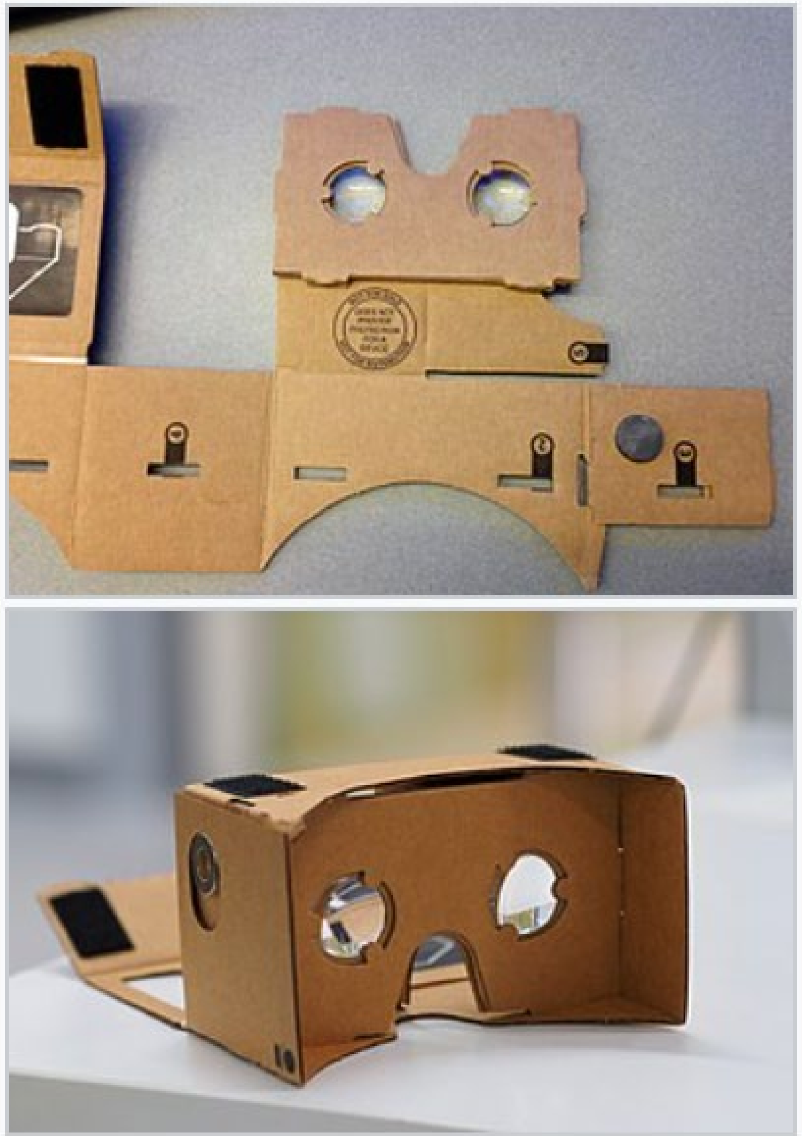
A typical stereoscope provides each eye with a lens that makes the image seen through it appear larger and more distant and usually also shifts its apparent horizontal position, so that for a person with normal binocular depth perception the edges of the two images seemingly fuse into one "stereo window".

https://en.wikipedia.org/wiki/Stereoscope

Google Cardboard

Second-generation Google Cardboard viewer

Developer	Google
Manufacturer	Google, third-party companies
Type	Virtual reality platform
Release date	June 25, 2014; 9 years ago
Discontinued	March 3, 2021; 2 years ago (Official viewer, Google Store)
Units shipped	15 million



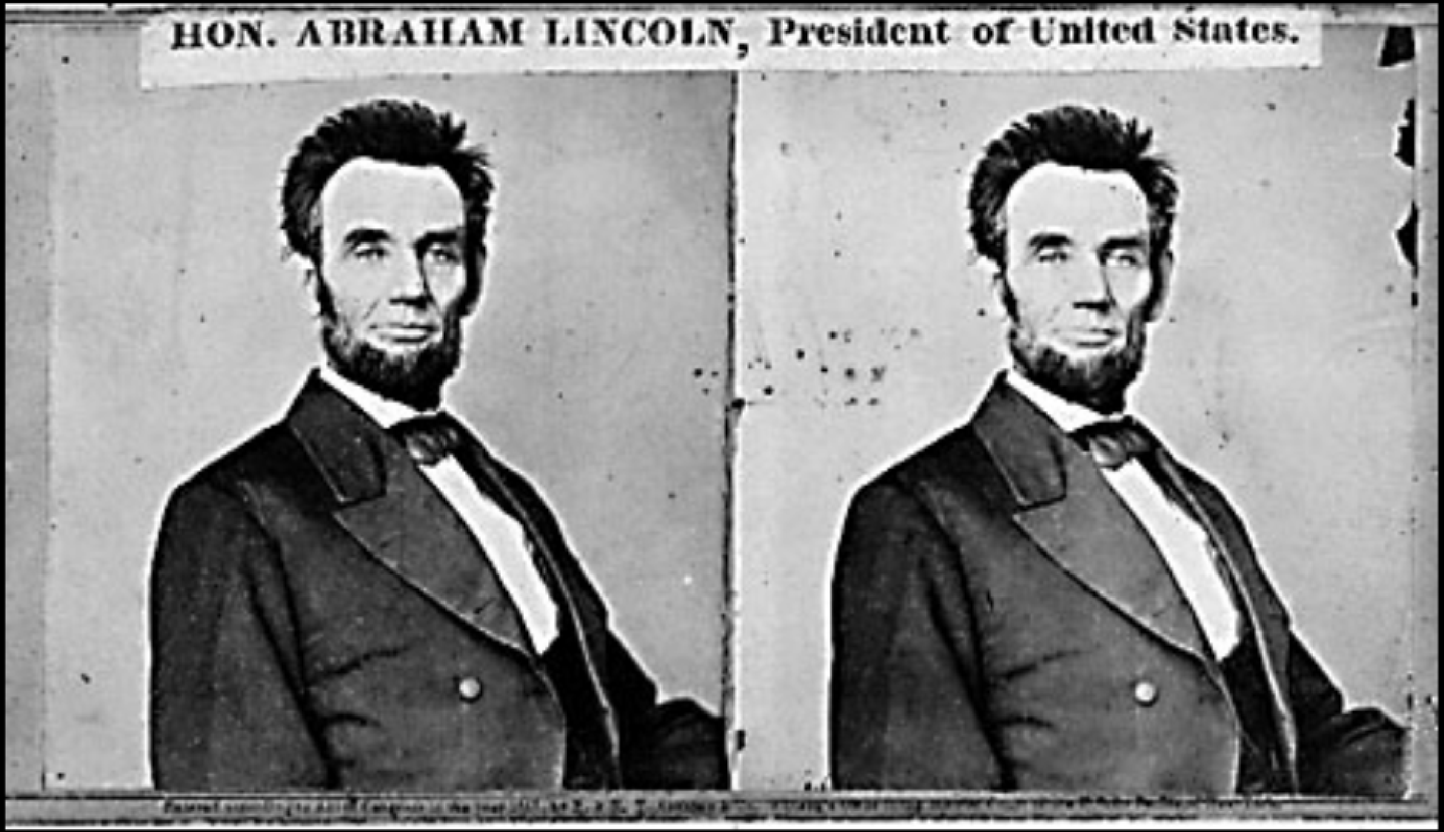
A Cardboard viewer unassembled *(top)* and assembled *(bottom)*

Once the kit is assembled, a smartphone is inserted in the back of the device and held in place by the selected fastening device. A Google Cardboard–compatible app splits the smartphone display image into two, one for each eye,

Apps on the mobile phone substitute for stereo cards; these apps can also sense rotation and expand the stereoscope's capacity into that of a full-fledged virtual reality device.

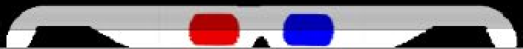
The underlying technology is otherwise unchanged from earlier stereoscopes.

https://en.wikipedia.org/wiki/Google_Cardboard

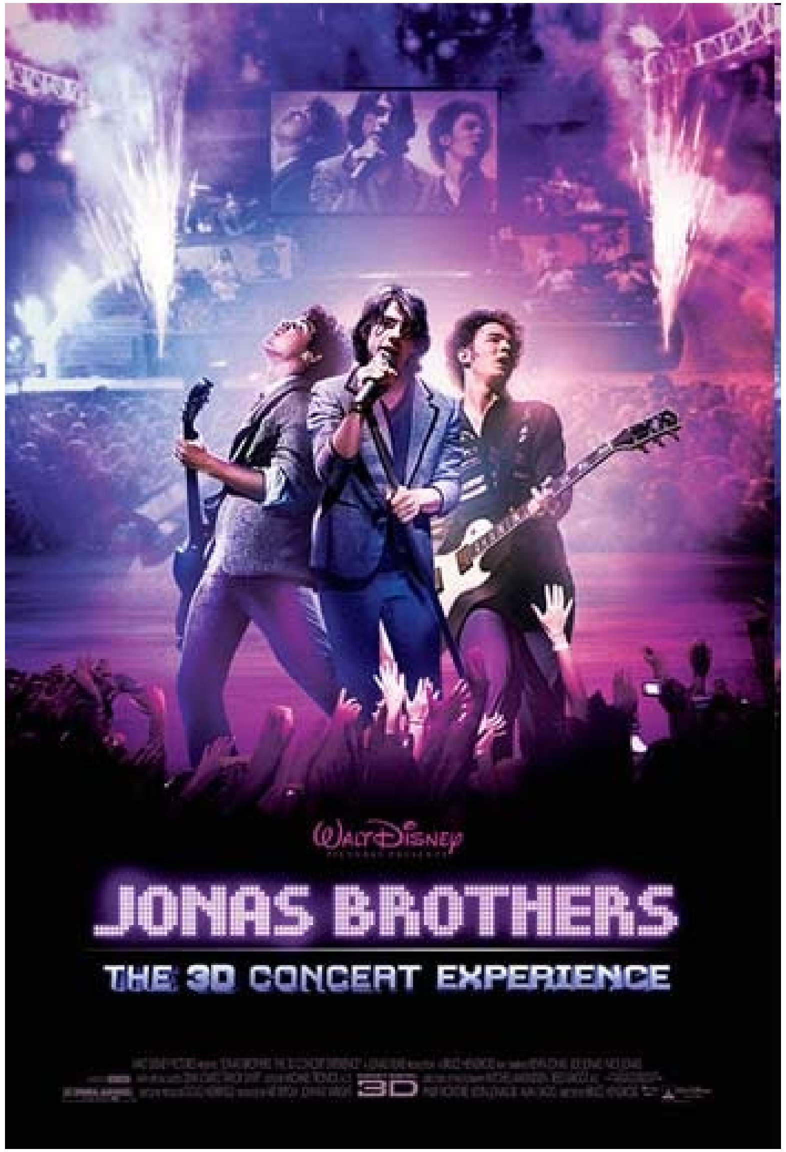

HON. ABRAHAM LINCOLN, President of United States.

Public Library, Stereoscopic Looking Room, Chicago, by Phillips, 1923

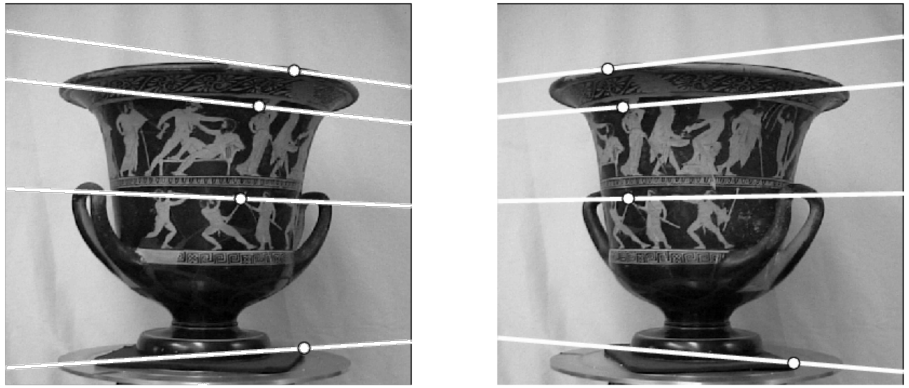
Mark Twain at Pool Table", no date, UCR Museum of Photography



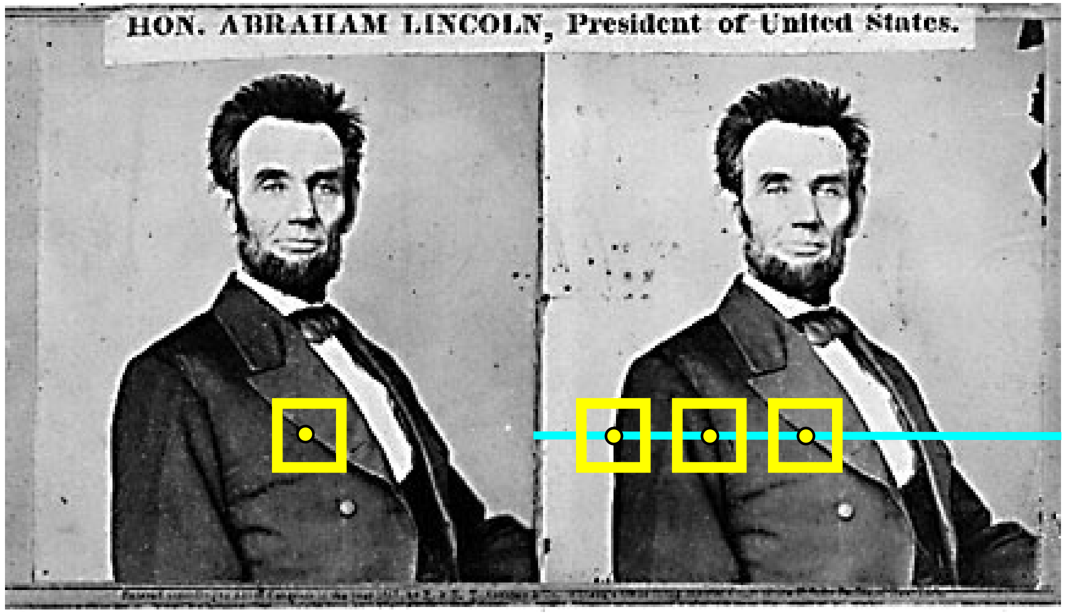
This is how 3D movies work



So can I compute depth from any two images of the same object?



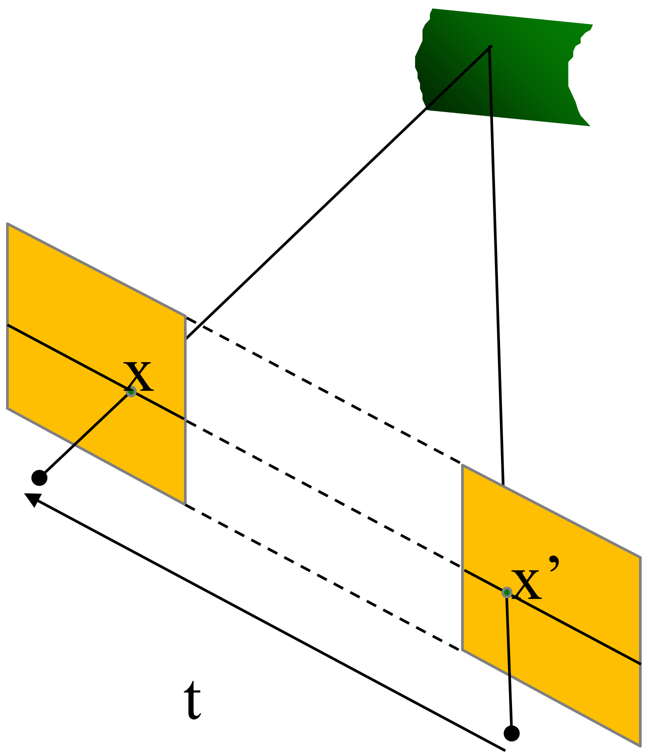
Yes if you can “rectify” them
i.e. make epipolar lines horizontal



1. Rectify images
   (make epipolar lines horizontal)
2. For each pixel
   a. Find epipolar line
   b. Scan line for best match
   c. Compute depth from disparity

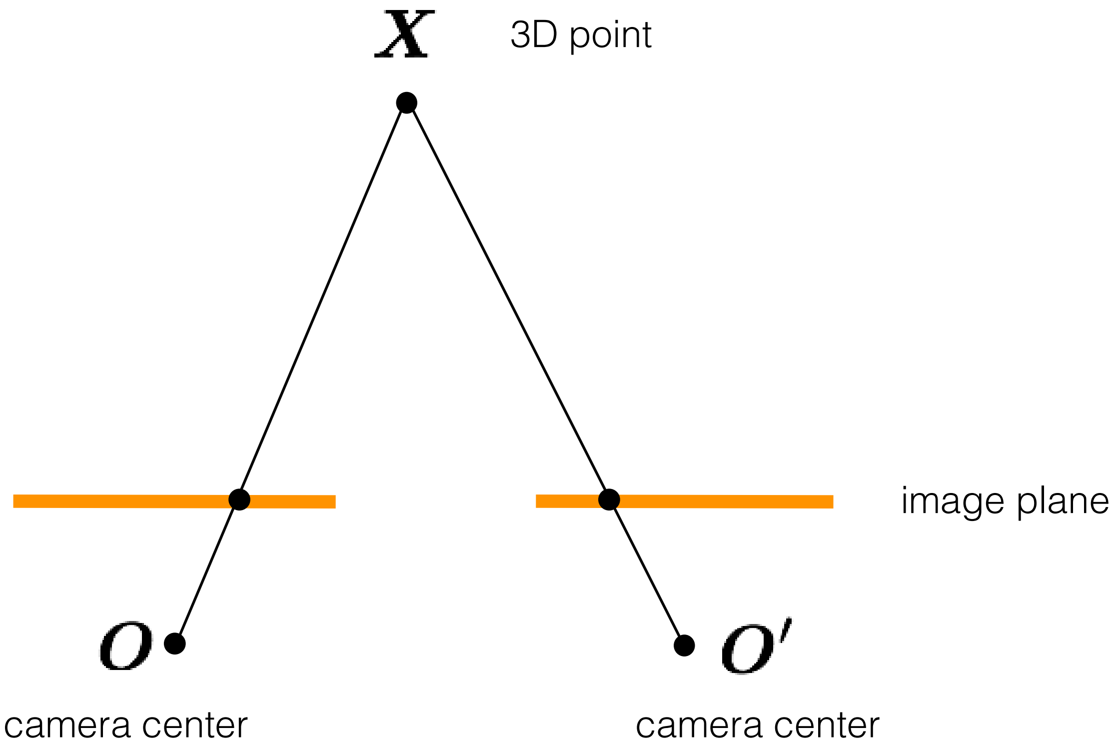
$$Z = \frac{bf}{d}$$

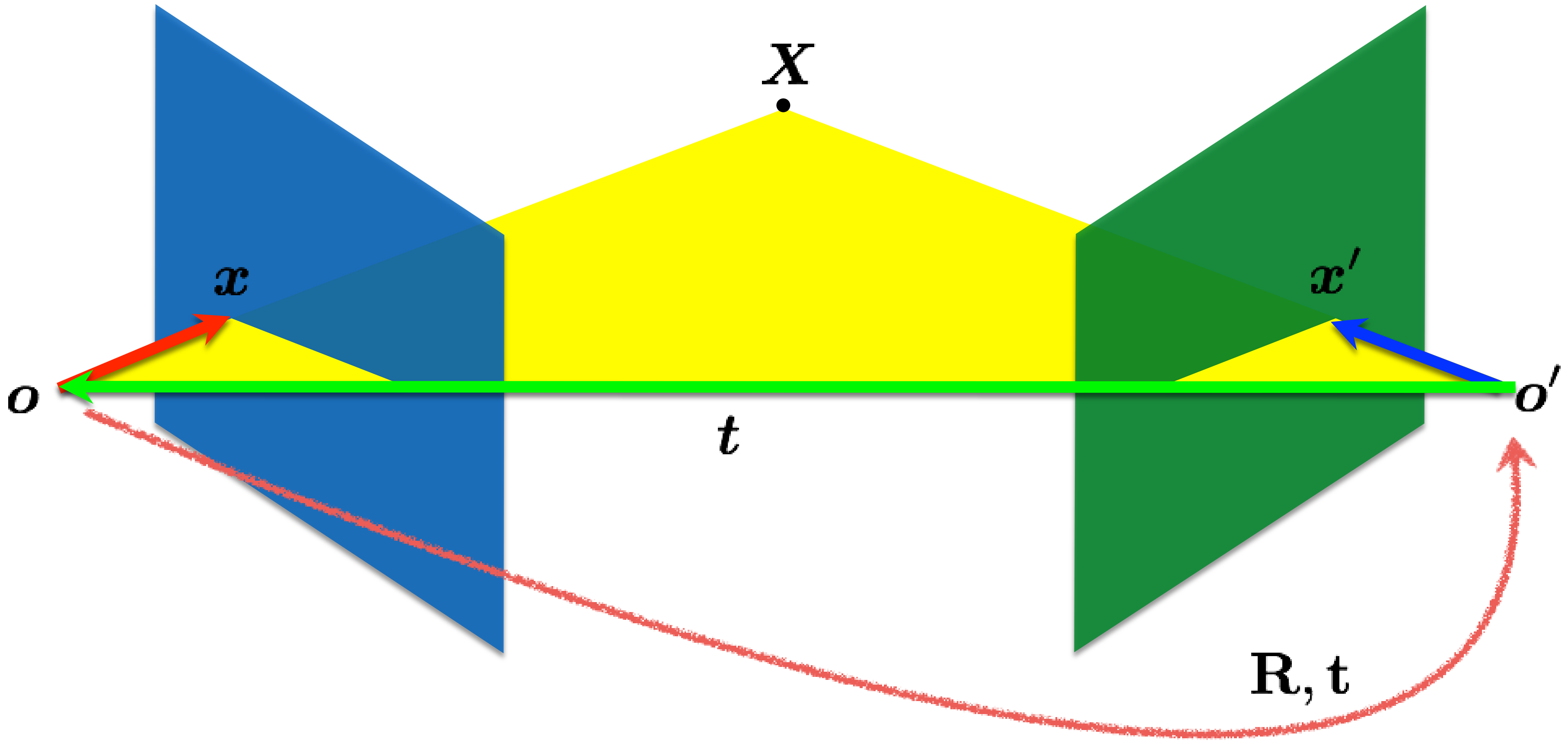
When are epipolar lines horizontal?



When this relationship holds:

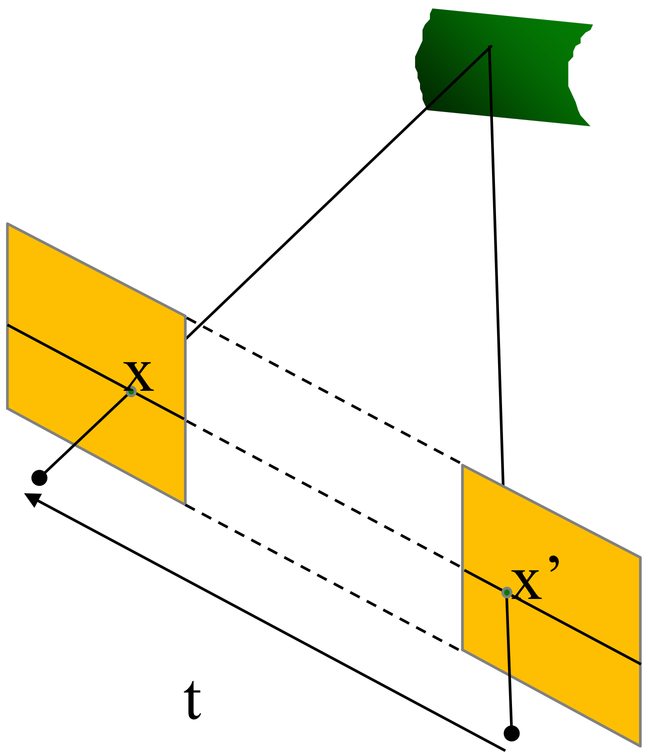
$$\boxed{R = I \qquad t = (T, 0, 0)}$$





$$\boldsymbol{x}' = \mathbf{R}(\boldsymbol{x} - \boldsymbol{t})$$

When are epipolar lines horizontal?



When this relationship holds:

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

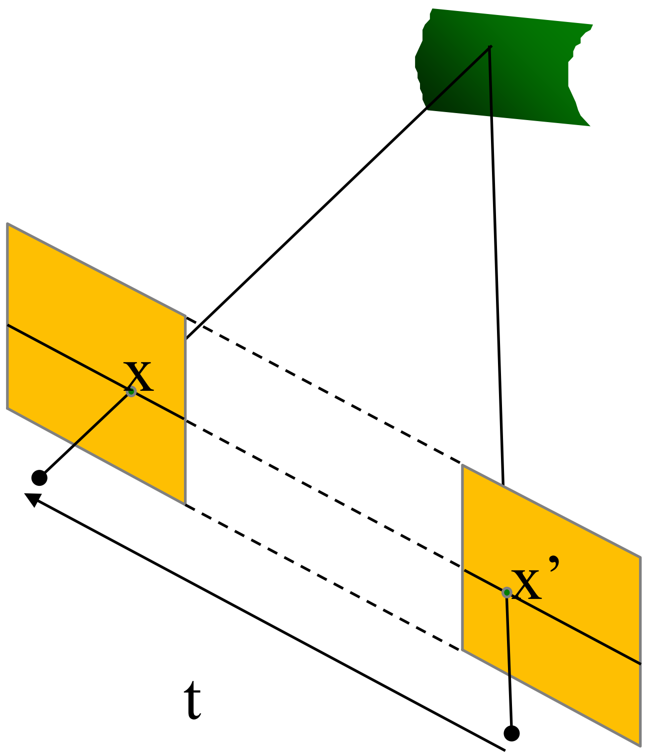
Let's try this out…

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

This always has to hold for rectified images

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

When are epipolar lines horizontal?



When this relationship holds:

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

Let's try this out…

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

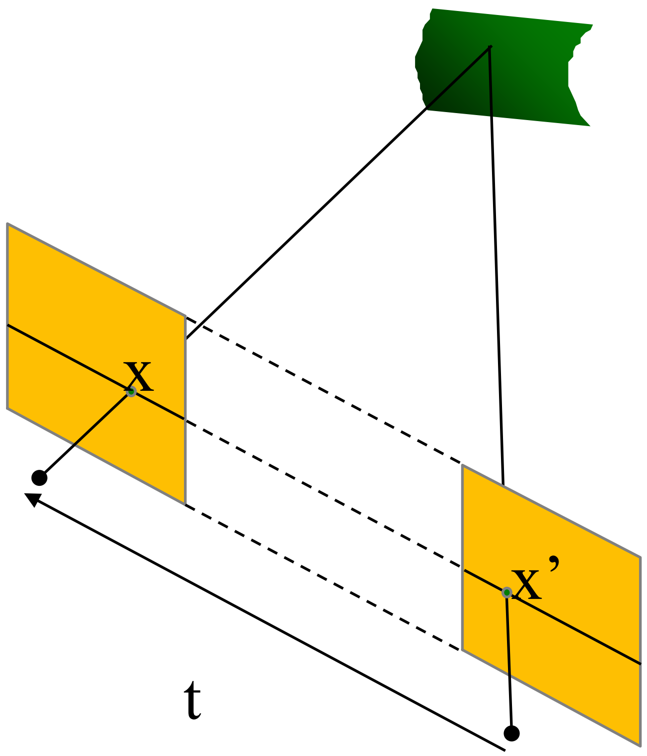
This always has to hold for rectified images

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

Write out the constraint

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

When are 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

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

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

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

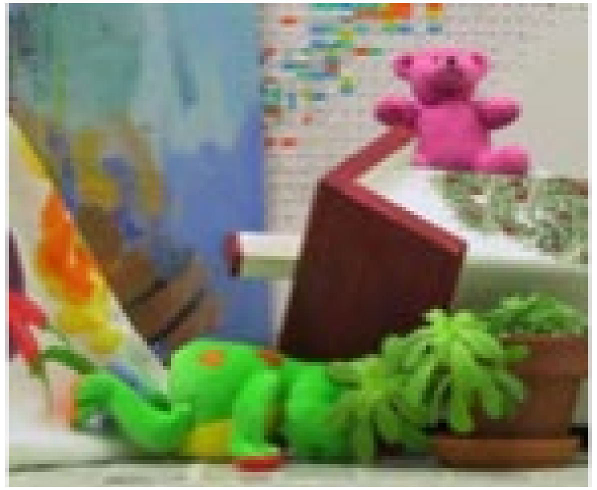
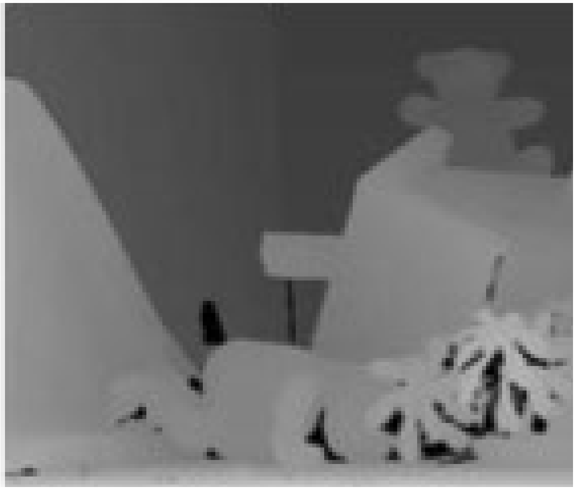
y coordinate is always the same!

Stereo rectification

2023-10-30

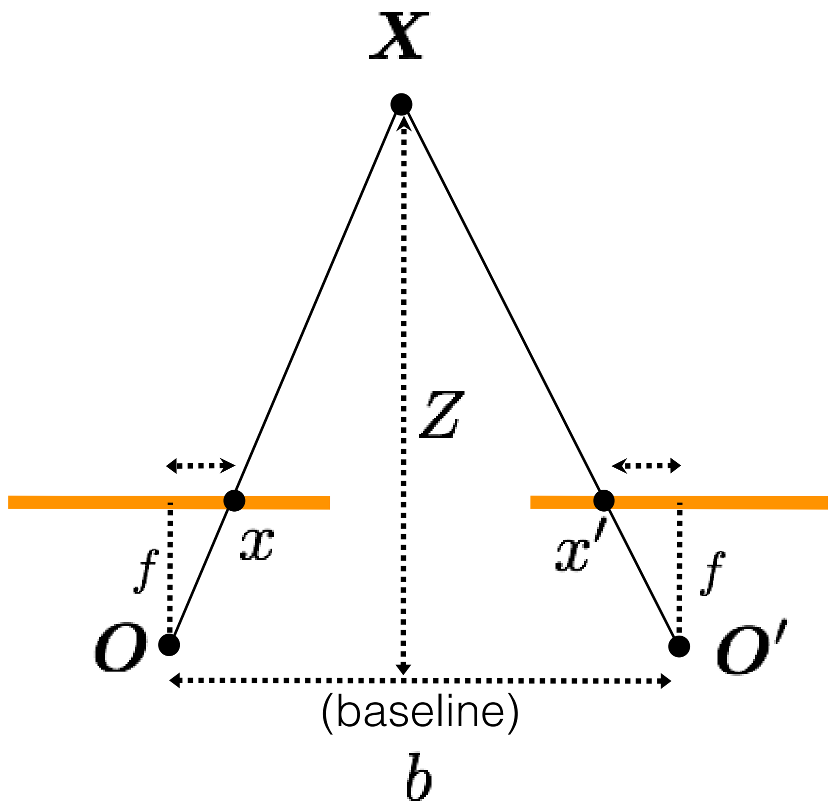
What's different between these two images?

The amount of horizontal movement is inversely proportional to …

… the distance from the camera.

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



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

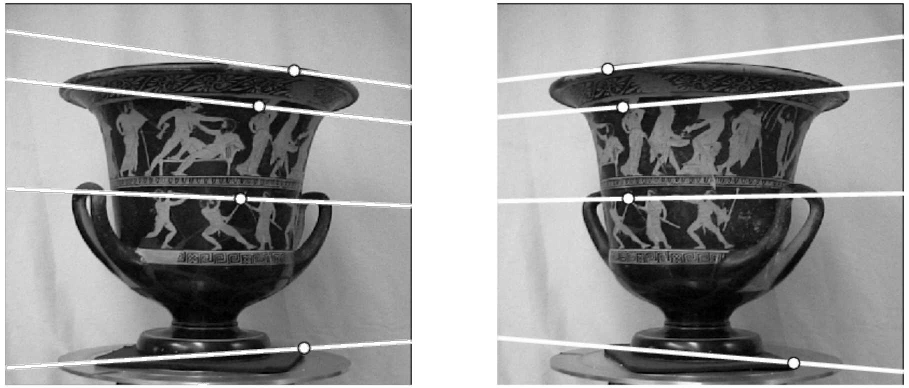
Disparity

$$d = x - x'$$

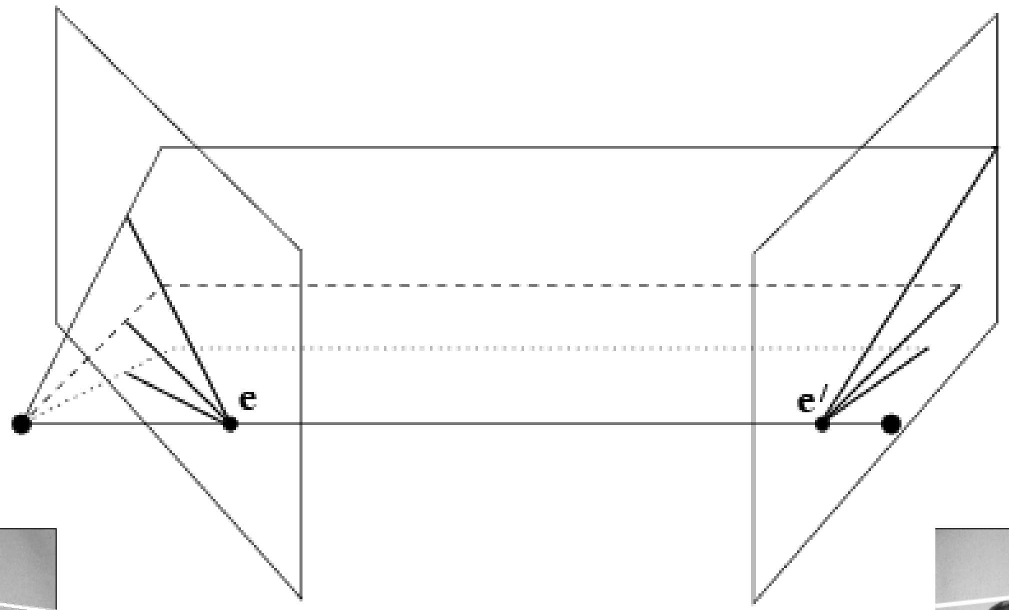
$$= \frac{bf}{Z}$$

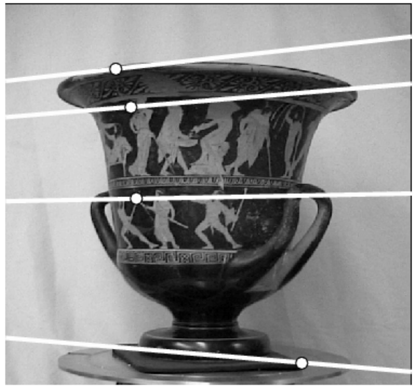
inversely proportional
to depth

So can I compute depth from any two images of the same object?



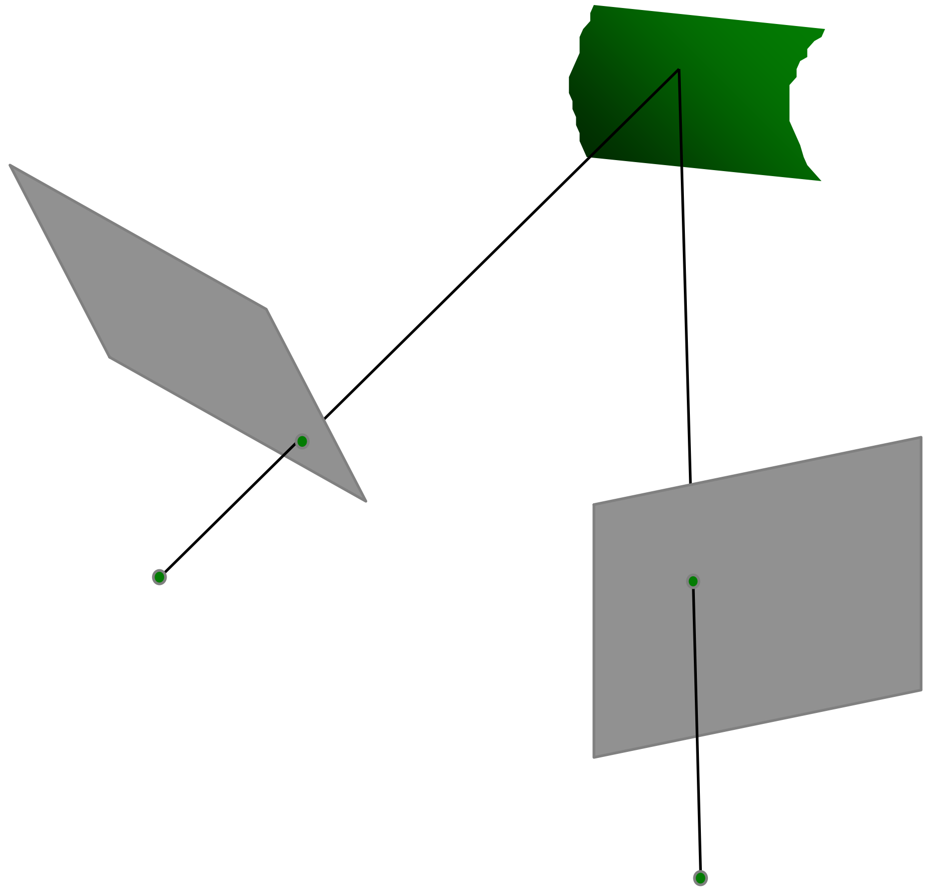
Yes if you can “rectify” them
i.e. make epipolar lines horizontal





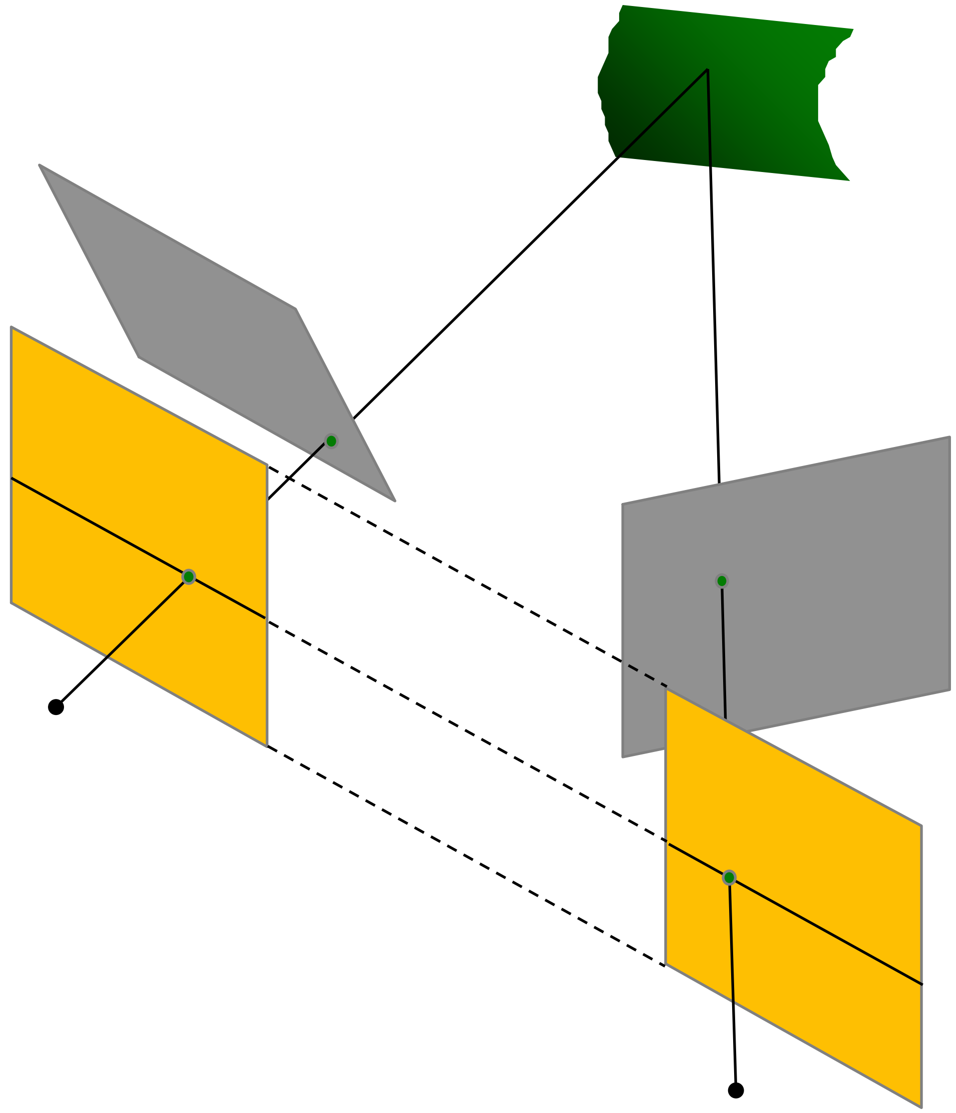
It’s hard to make the image planes exactly parallel

Stereo Rectification



Stereo Rectification

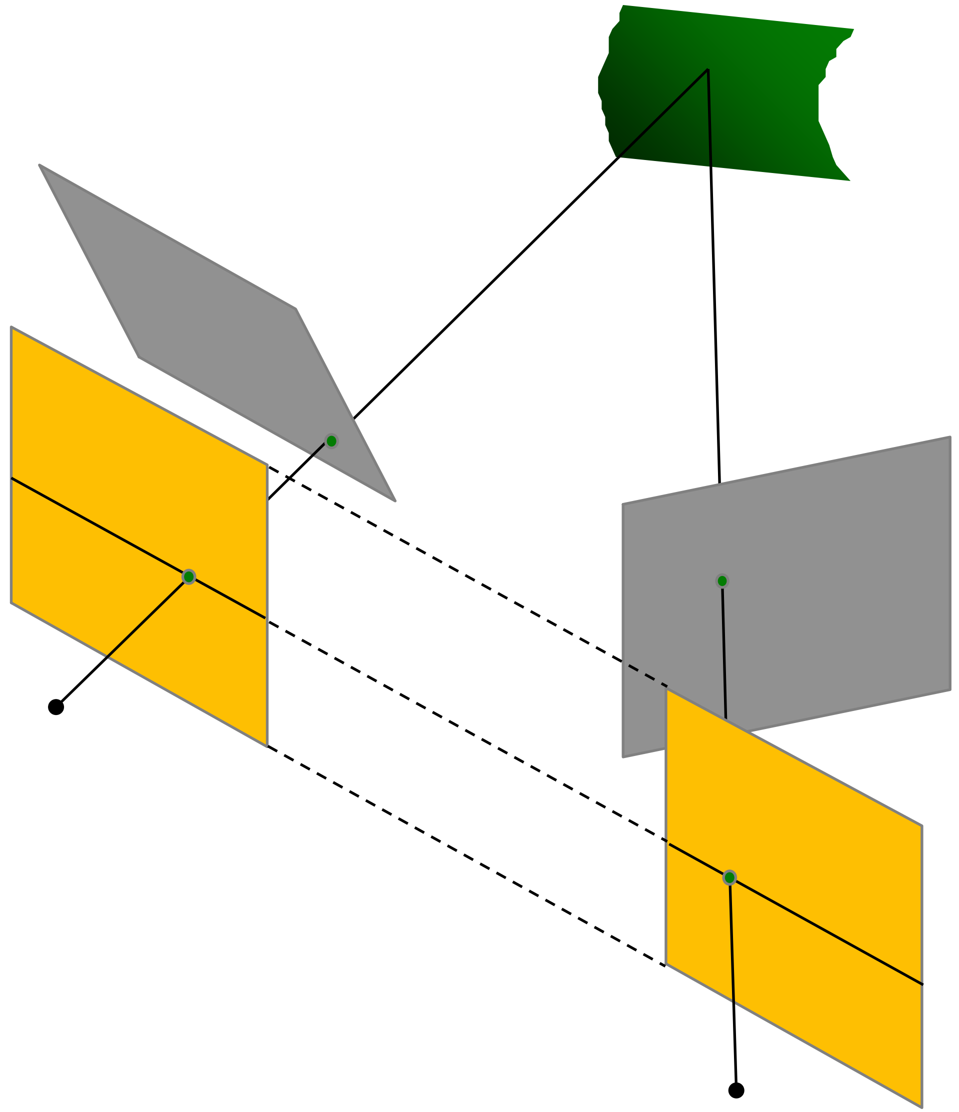
Reproject image planes onto a common plane parallel to the line between camera centers



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

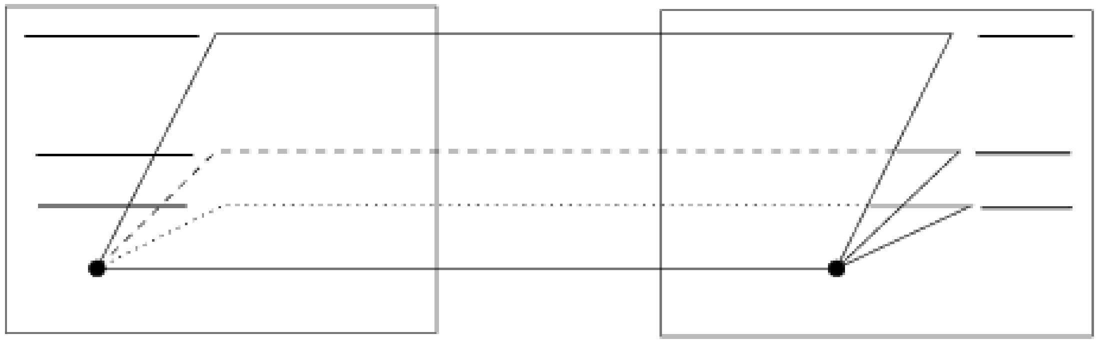
1. **Rotate** the right camera by **R**
   (aligns camera coordinate system orientation only)

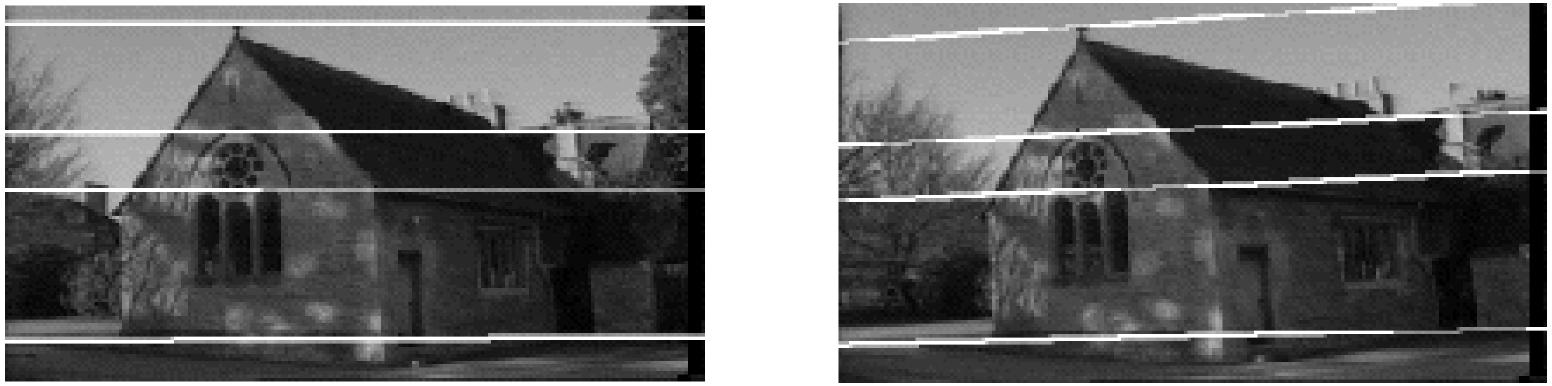
2. Rotate (**rectify**) the left camera so that the epipole is at infinity

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

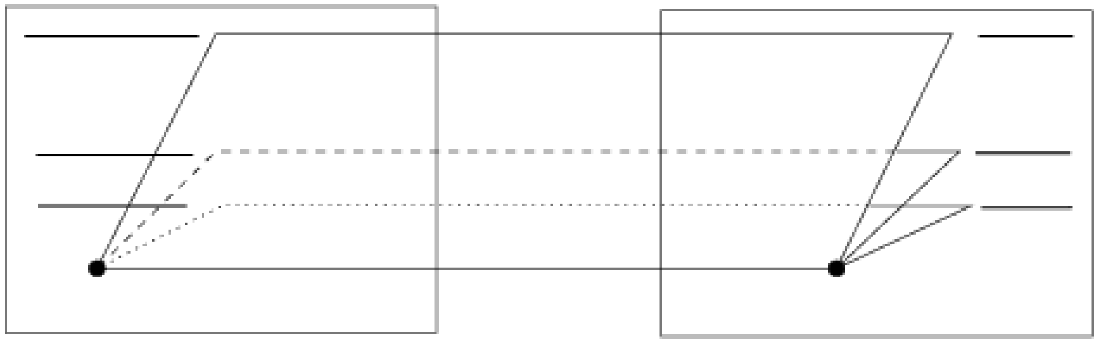
4. Adjust the **scale**

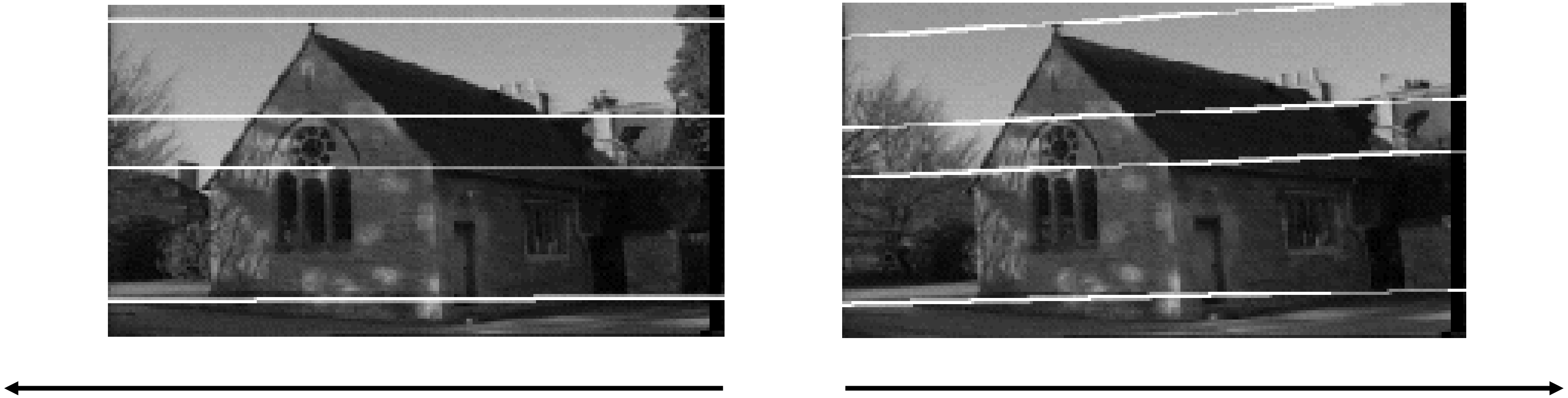
Parallel cameras





Parallel cameras





epipole at infinity

Setting the epipole to infinity

(Building $\mathbf{R}_{\text{rect}}$ from **e**)

Let $R_{\text{rect}} = \begin{bmatrix} \boldsymbol{r}_1^\top \\ \boldsymbol{r}_2^\top \\ \boldsymbol{r}_3^\top \end{bmatrix}$

Given:

epipole **e**
(using SVD on E)
(translation from **E)**

$$\boldsymbol{r}_1 = \boldsymbol{e}_1 = \frac{T}{||T||}$$

epipole coincides with translation vector

$$\boldsymbol{r}_2 = \frac{1}{\sqrt{T_x^2 + T_y^2}} \begin{bmatrix} -T_y & T_x & 0 \end{bmatrix}$$

cross product of e and the direction vector of the optical axis

$$\boldsymbol{r}_3 = \boldsymbol{r}_1 \times \boldsymbol{r}_2$$

orthogonal vector

If $\quad \boldsymbol{r}_1 = \boldsymbol{e}_1 = \frac{T}{||T||} \quad$ and $\quad \boldsymbol{r}_2 \quad \boldsymbol{r}_3 \quad$ orthogonal

then $\quad R_{\text{rect}} \boldsymbol{e}_1 = \begin{bmatrix} \boldsymbol{r}_1^\top \boldsymbol{e}_1 \\ \boldsymbol{r}_2^\top \boldsymbol{e}_1 \\ \boldsymbol{r}_3^\top \boldsymbol{e}_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

At x-infinity

Stereo Rectification Algorithm

1. Estimate **E** using the 8 point algorithm (SVD)
2. Estimate the epipole **e** (SVD of **E**)
3. Build $\mathbf{R}_{rect}$ from **e**
4. Decompose **E** into **R** and **T**
5. Set $\mathbf{R}_1 = \mathbf{R}_{rect}$ and $\mathbf{R}_2 = \mathbf{R}\mathbf{R}_{rect}$
6. Rotate each left camera point (warp image)
   $[x' \; y' \; z'] = \mathbf{R}_1 [x \; y \; z]$
7. Rectified points as $\mathbf{p} = f/z'[x' \; y' \; z']$
8. Repeat 6 and 7 for right camera points using $\mathbf{R}_2$

Use built-in OpenCV functions for this

stereoRectifyUncalibrated()

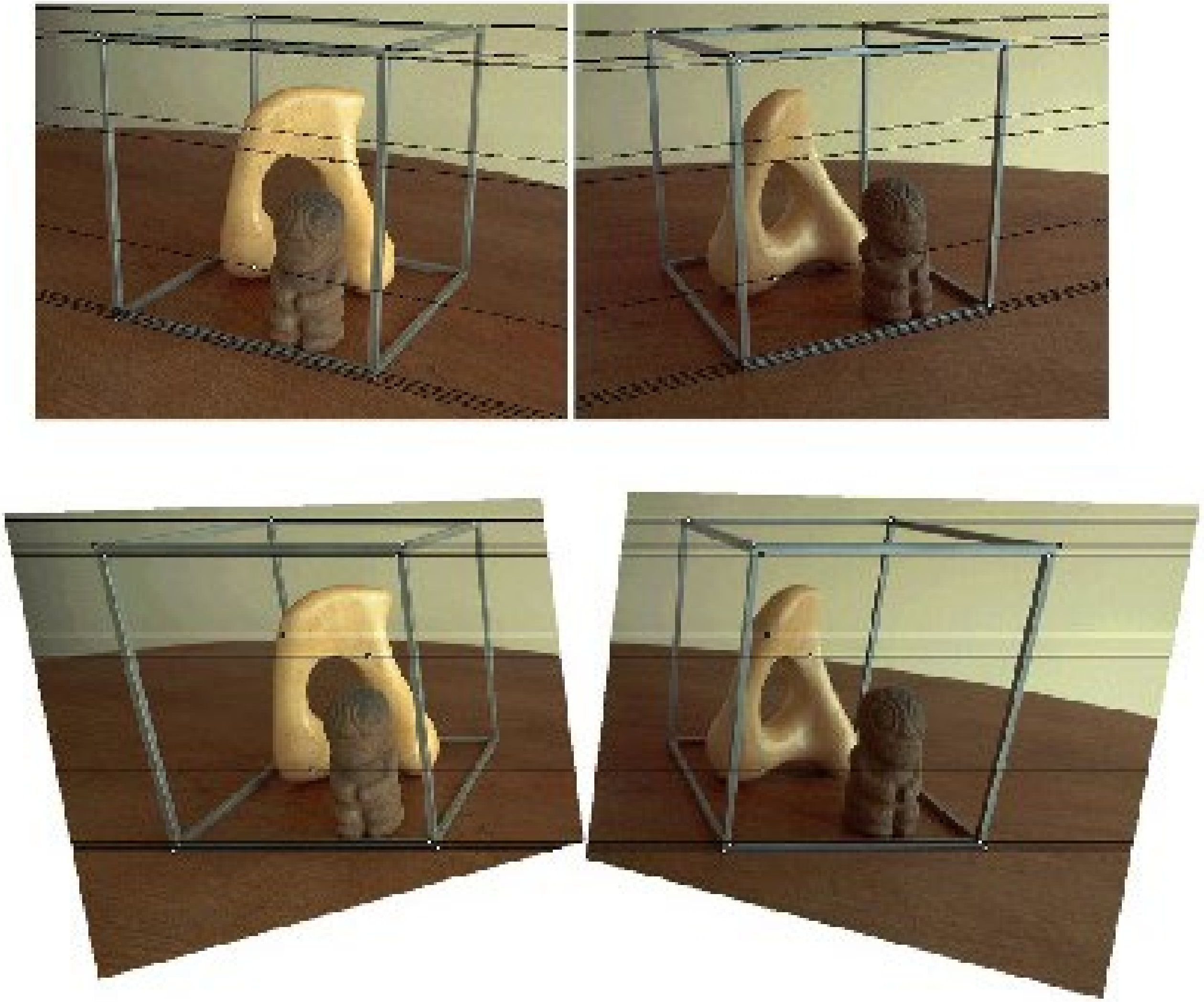
Computes a rectification transform for an uncalibrated stereo camera.

Parameters

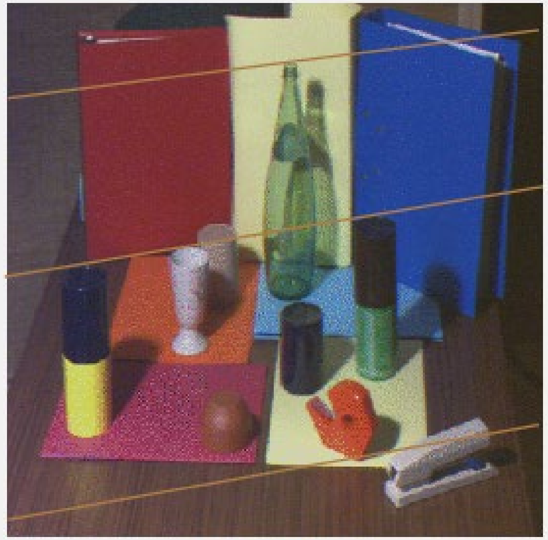
points1	Array of feature points in the first image.
points2	The corresponding points in the second image. The same formats as in findFundamentalMat are supported.
F	Input fundamental matrix. It can be computed from the same set of point pairs using findFundamentalMat .
imgSize	Size of the image.
H1	Output rectification homography matrix for the first image.
H2	Output rectification homography matrix for the second image.
threshold	Optional threshold used to filter out the outliers. If the parameter is greater than zero, all the point pairs that do not comply with the epipolar geometry (that is, the points for which $\lvert \texttt{points2[i]}^T * \mathrm{F} * \texttt{points1[i]} \rvert > \texttt{threshold}$) are rejected prior to computing the homographies. Otherwise, all the points are considered inliers.

The function computes the rectification transformations without knowing intrinsic parameters of the cameras and their relative position in the space, which explains the suffix "uncalibrated". Another related difference from stereoRectify is that the function outputs not the rectification transformations in the object (3D) space, but the planar perspective transformations encoded by the homography matrices H1 and H2 . The function implements the algorithm [88] .

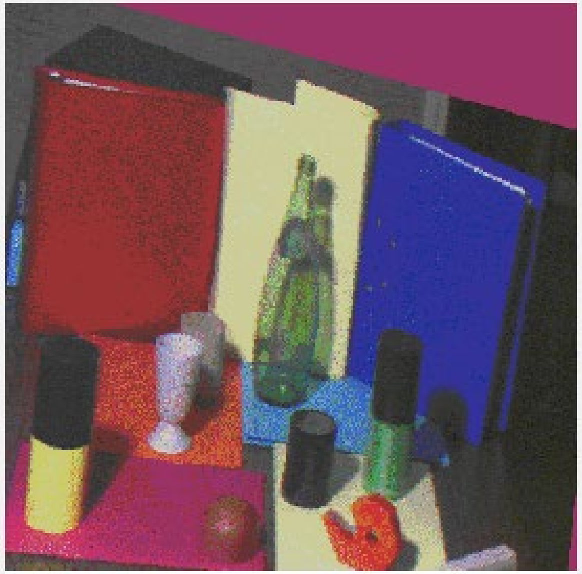
Rectification example



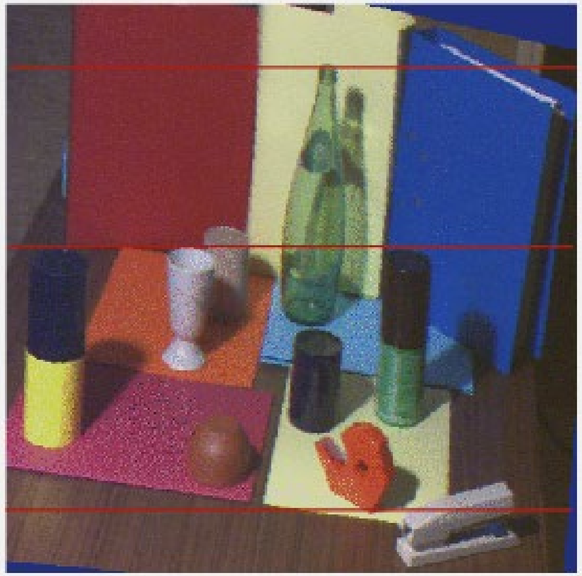
Left



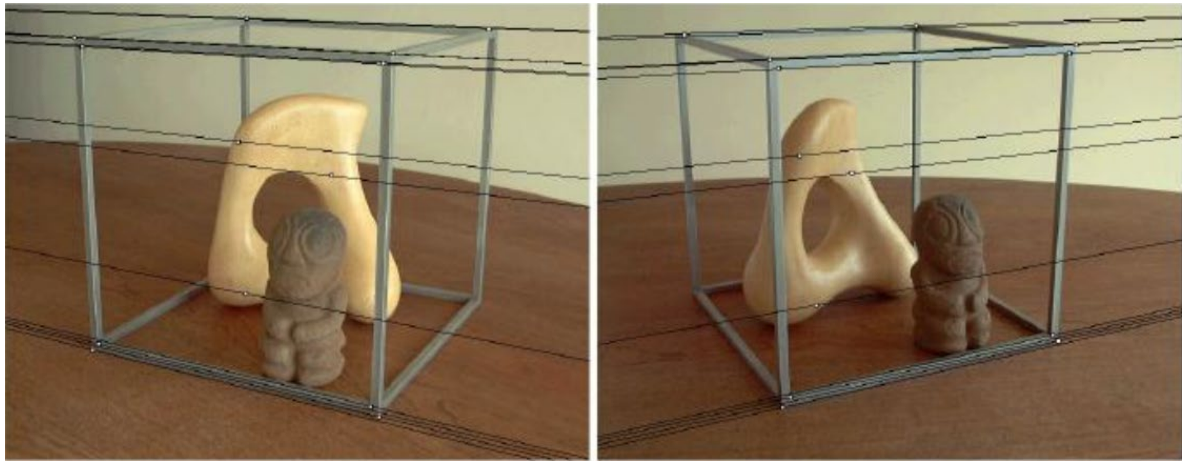
Right



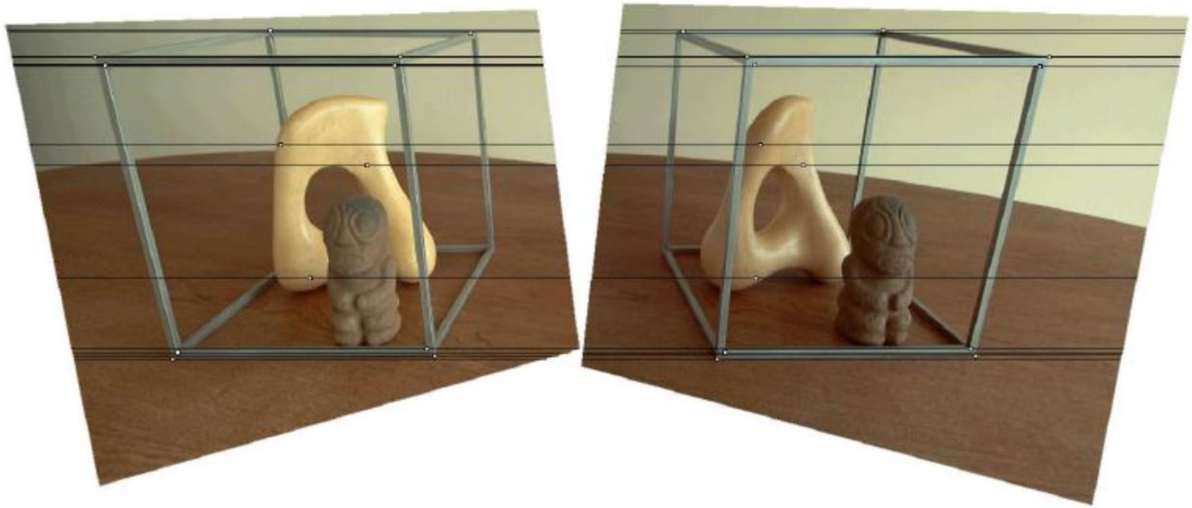
Rectified Left



Rectified Right



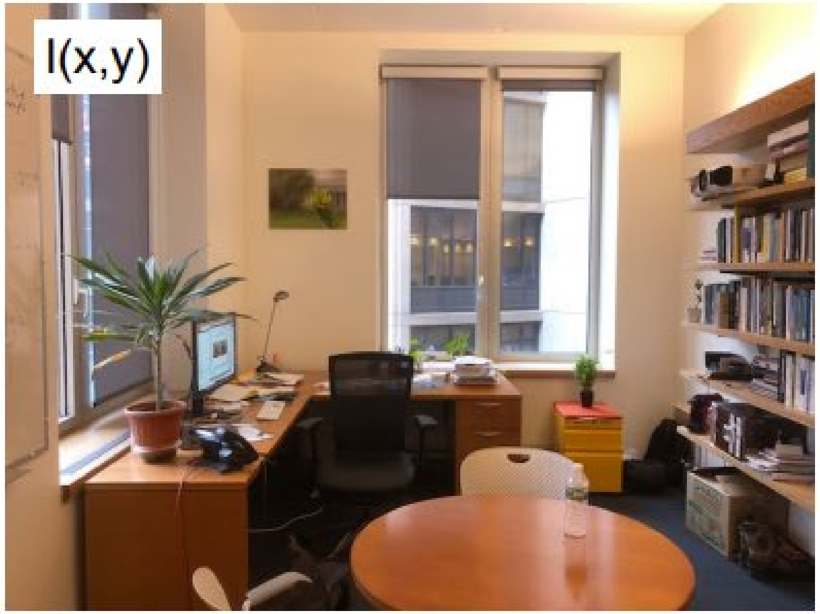
What can we do after rectification?

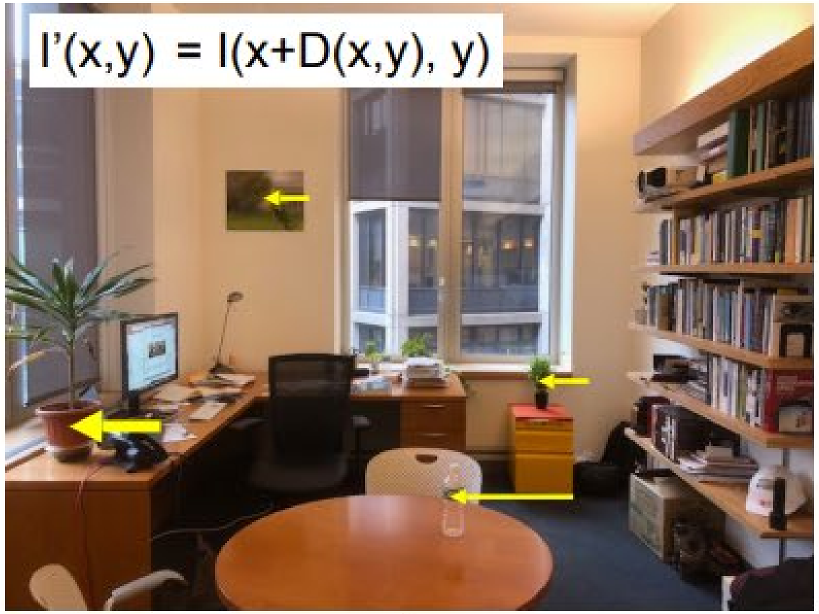


Depth Estimation

Depth Estimation via Stereo Matching

Disparity map



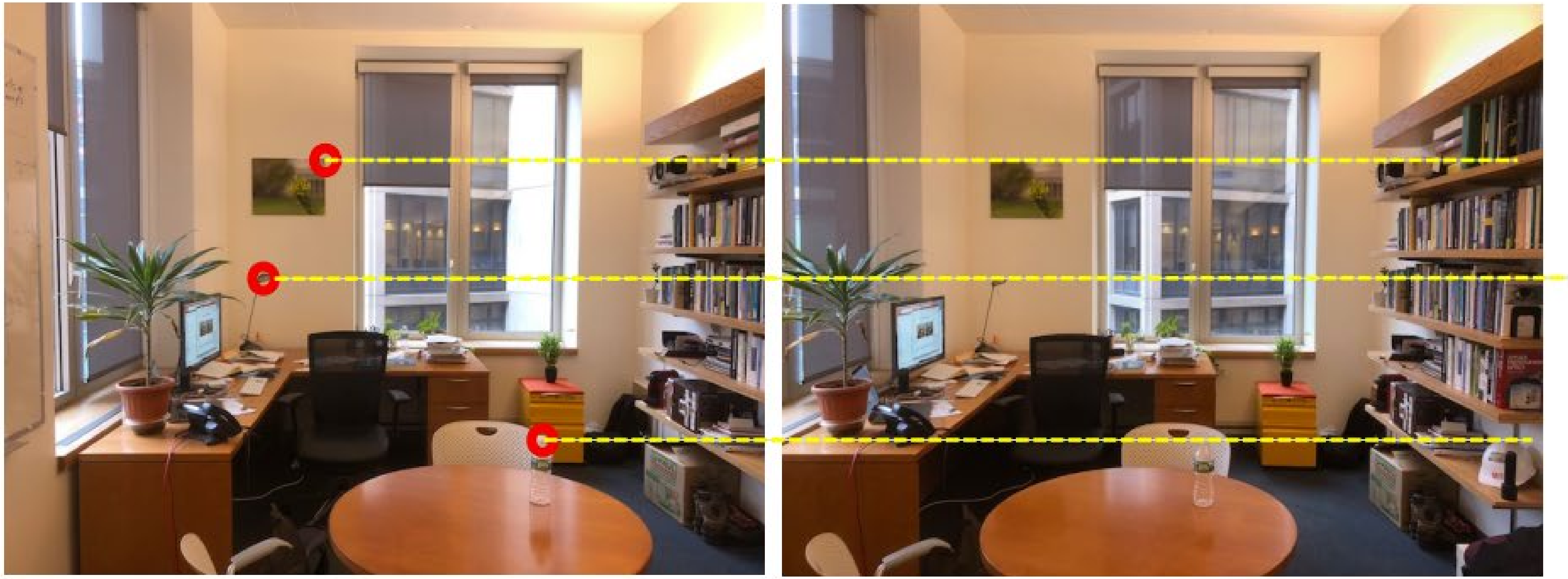


D(x,y)

$$Z(x,y) = \frac{f}{D(x,y)}$$

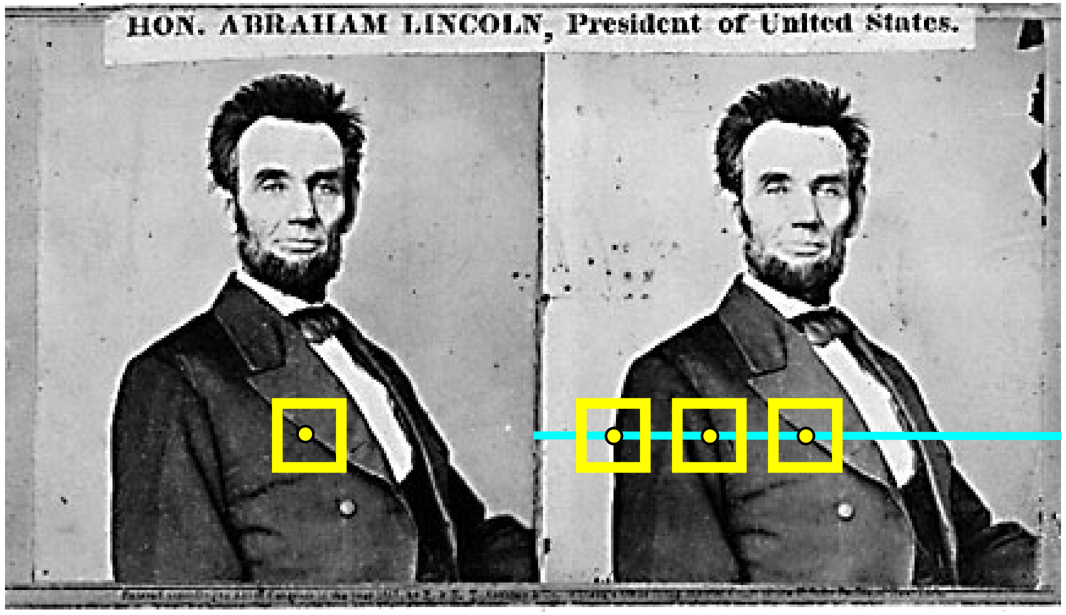
Source: Torralba, Isola, Freeman

Finding correspondences



We only need to search for matches along horizontal lines.

Source: Torralba, Isola, Freeman



1. Rectify images
   (make epipolar lines horizontal)
2. For each pixel
   a. Find epipolar line
   b. Scan line for best match
   c. Compute depth from disparity

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

How would you do this?

Computing disparity

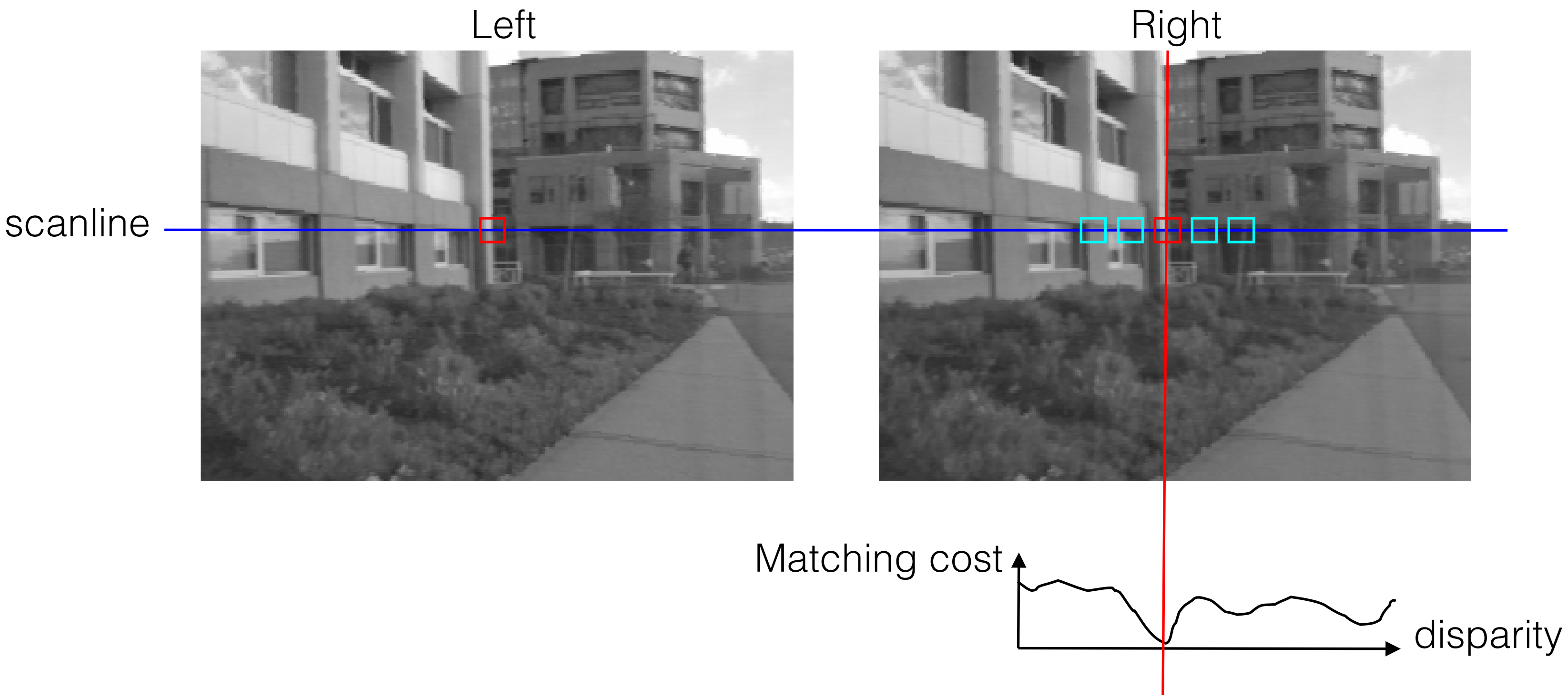
Source: Torralba, Isola, Freeman

Computing disparity

Semi-global matching [Hirschmüller 2008]

Source: Torralba, Isola, Freeman

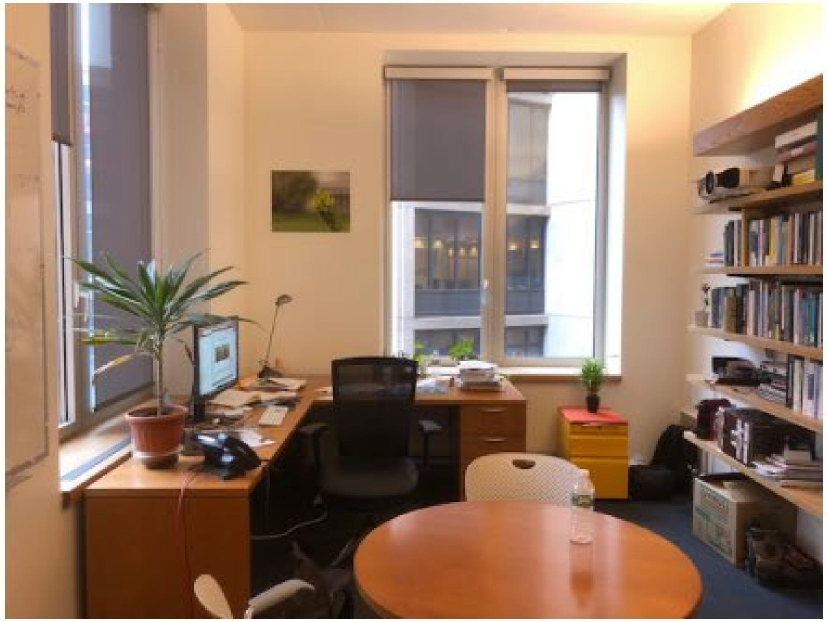
Stereo Block Matching

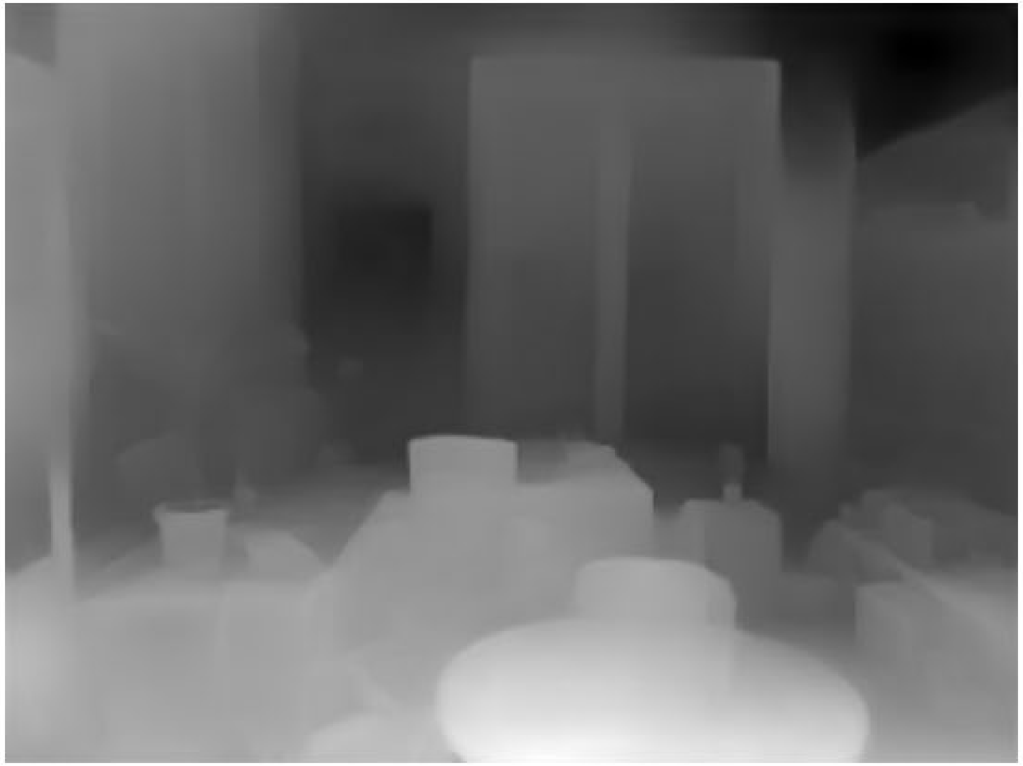


- Slide a window along the epipolar line and compare contents of that window with the reference window in the left image
- Matching cost: SSD or normalized correlation

Depth from Single Image

Can also learn depth from a single image





MegaDepth: Learning Single-View Depth Prediction from Internet Photos

Zhengqi Li   Noah Snavely
Department of Computer Science & Cornell Tech, Cornell University

Source: Torralba, Isola, Freeman

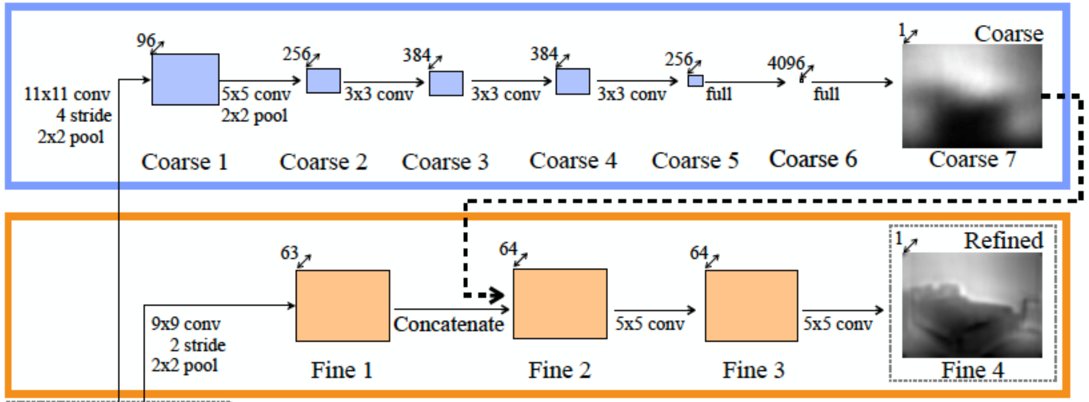
Use inference power of deep learning to regress depth directly from single image

Not as accurate as stereo methods, but still solves ambiguity issues through semantic cues

Depth Map Prediction from a Single Image using a Multi-Scale Deep Network

David Eigen deigen@cs.nyu.edu
Christian Puhrsch cpuhrsch@nyu.edu
Rob Fergus fergus@cs.nyu.edu

Dept. of Computer Science, Courant Institute, New York University



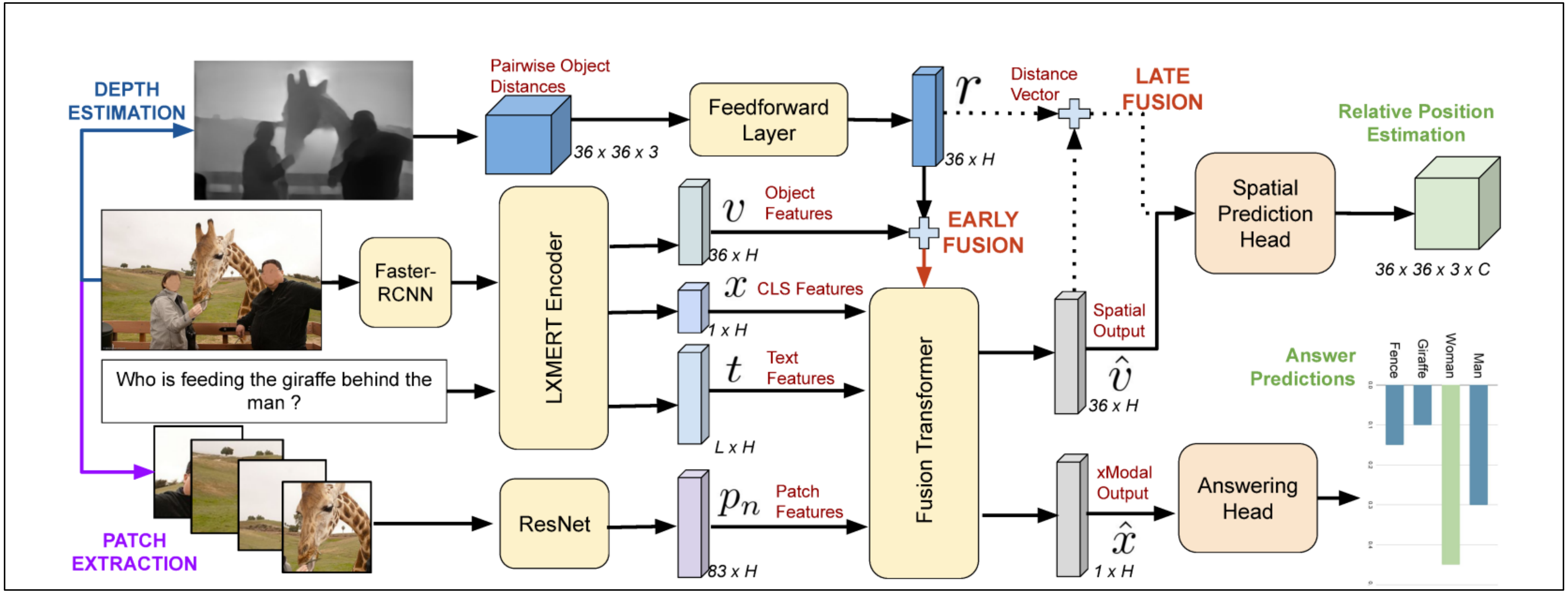
		Coarse					Fine
Layer	input	1	2,3,4	5	6	7	1,2,3,4
Size (NYUDepth)	304x228	37x27	18x13	8x6	1x1	74x55	74x55
Size (KITTI)	576x172	71x20	35x9	17x4	1x1	142x27	142x27
Ratio to input	/1	/8	/16	/32	–	/4	/4

Figure 1: Model architecture.

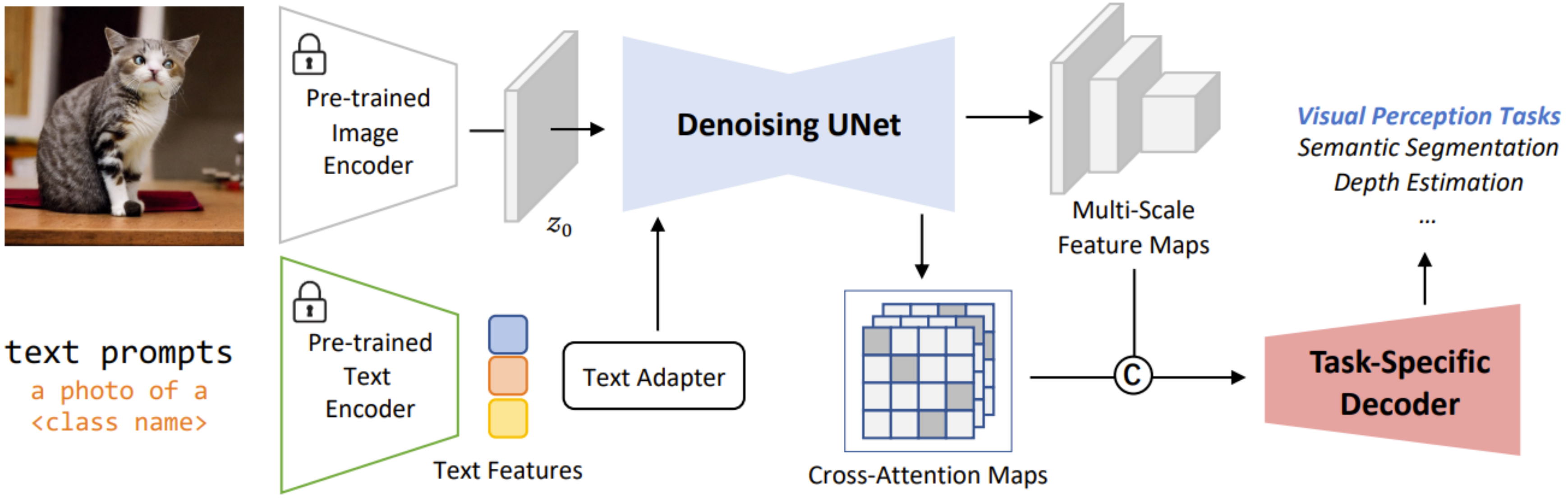
My Research: ICCV 2021: Answering Questions about Images using Depth Information

Weakly Supervised Relative Spatial Reasoning for Visual Question Answering

Pratyay Banerjee Tejas Gokhale Yezhou Yang Chitta Baral
Arizona State University
{pbanerj6, tgokhale, yz.yang, chitta}@asu.edu



VPD: Language-Guided Depth Estimation



Unleashing Text-to-Image Diffusion Models for Visual Perception

Wenliang Zhao[1*] Yongming Rao[1*] Zuyan Liu[1*] Benlin Liu[2] Jie Zhou[1] Jiwen Lu[1†]

[1]Tsinghua University [2]University of Washington

My Research: CVPR 2024: Quantifying Efficacy of Language-Guided Depth Estimation

On the Robustness of Language Guidance for Low-Level Vision Tasks: Findings from Depth Estimation

Agneet Chatterjee[◇] Tejas Gokhale[♠] Chitta Baral[◇] Yezhou Yang[◇]

[◇]Arizona State University [♠]University of Maryland, Baltimore County

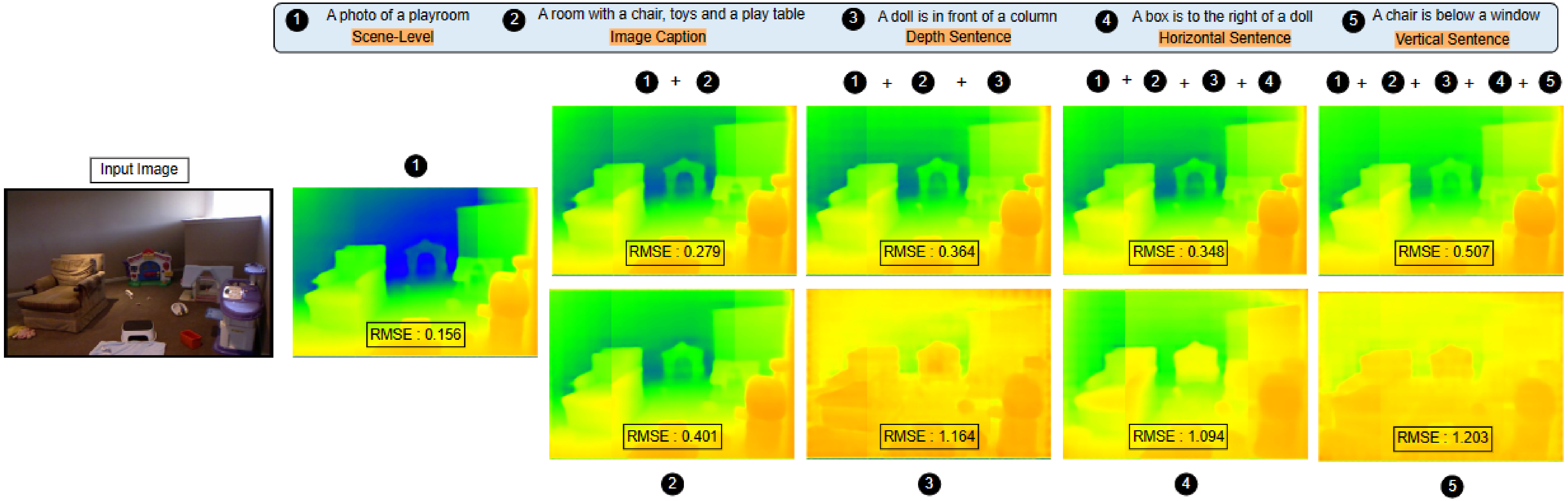


Figure 2. An illustration of depth maps generated by language-guided depth estimation methods such as VPD (**zero-shot**) when prompted with various sentence inputs that we use as part of our study. The first row shows the effect of progressively adding descriptions as input, while the second row shows depth maps generated by single sentence inputs.