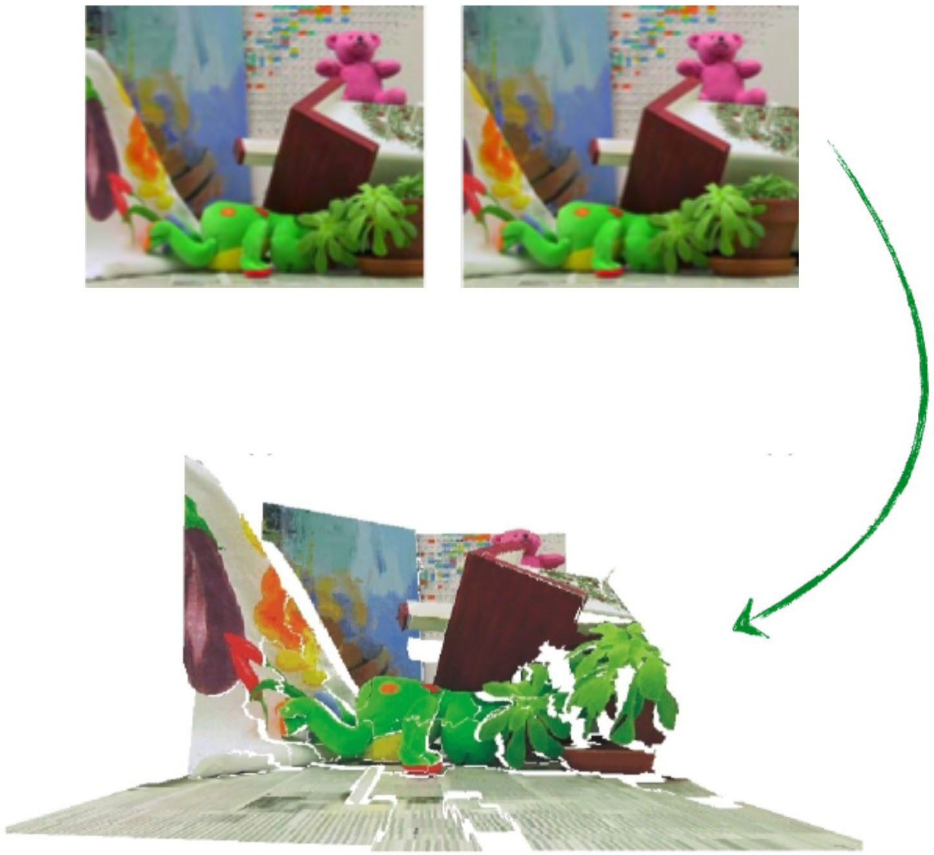
Introduction to Computer Vision

Kaveh Fathian
Assistant Professor
Computer Science Department
Colorado School of Mines

Lecture 13

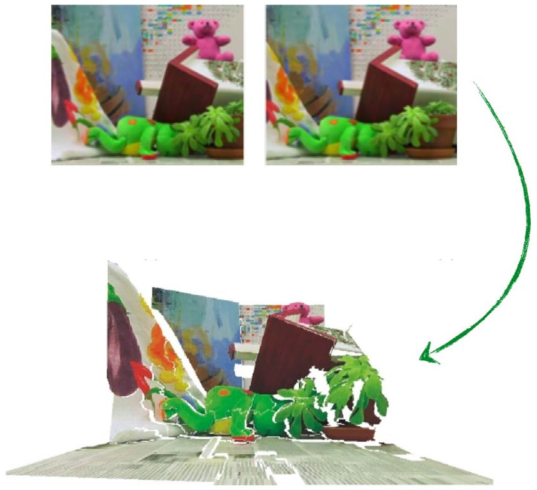
Stereo Vision

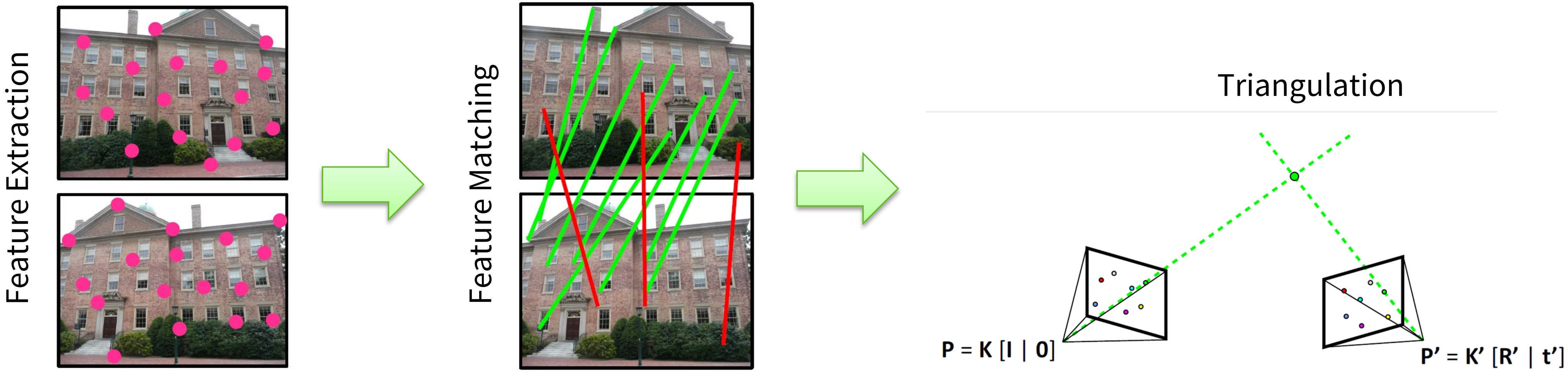


Stereo Vision

Can we reconstruct a 3D depth map from 2 images?

- Use **2-view geometry**:
  1. Extract features
  2. Match features
  3. Find essential/homography matrix (via RANSAC)
  4. Recover camera matrix $(\boldsymbol{R}, \boldsymbol{t})$
  5. Triangulate 3D points





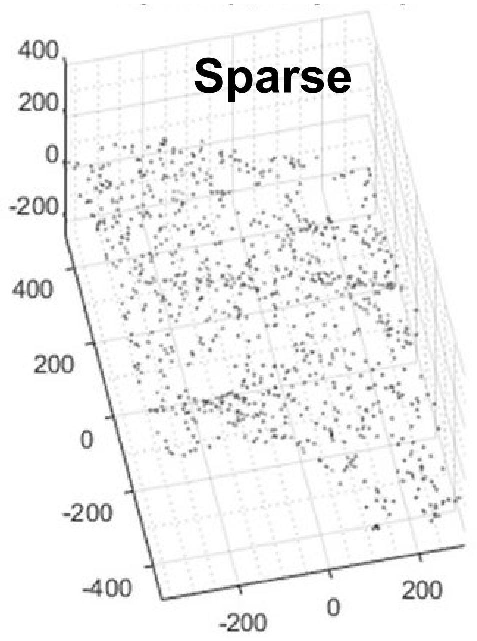
Stereo Vision

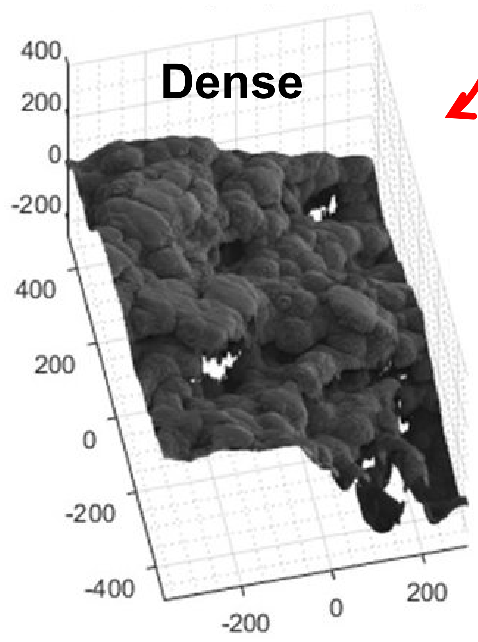
Issues of 2-view geometry:

- Sparse 3D map
- Time consuming to find (correct) corresponding features
- Time consuming to triangulate

Stereo vision: Given camera matrix (calibration, $\boldsymbol{R}, \boldsymbol{t}$), stereo reconstruction solves issues:

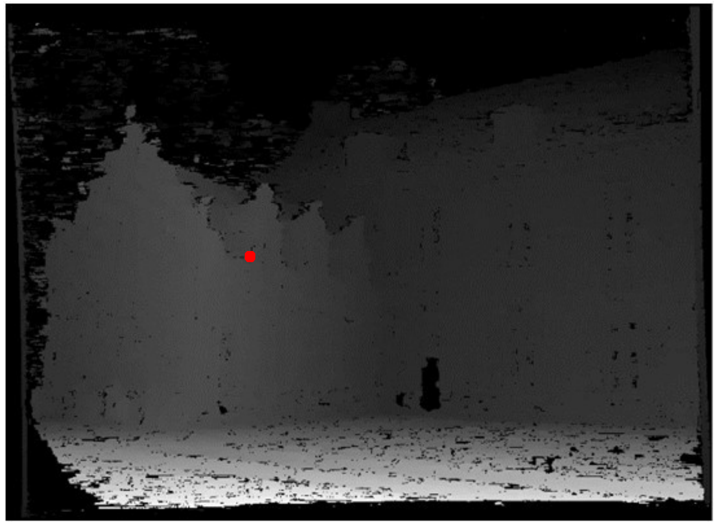
- Dense 3D map
- Fast computation





Depth from Disparity

image I(x,y)



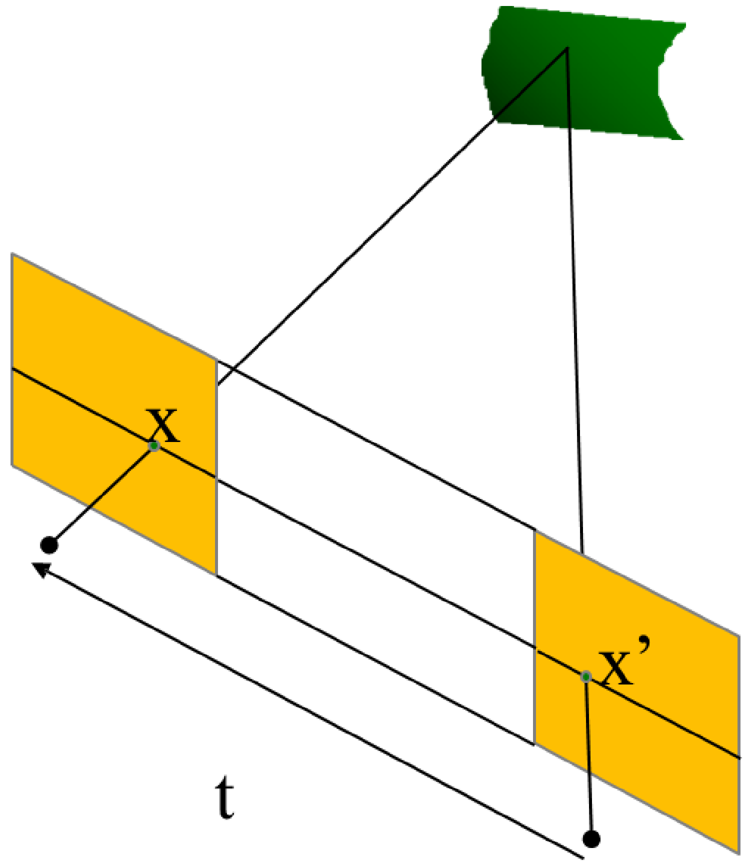
Disparity map D(x,y)

image I´(x´,y´)

$$(x', y') = (x+\mathbf{d},\ y)$$

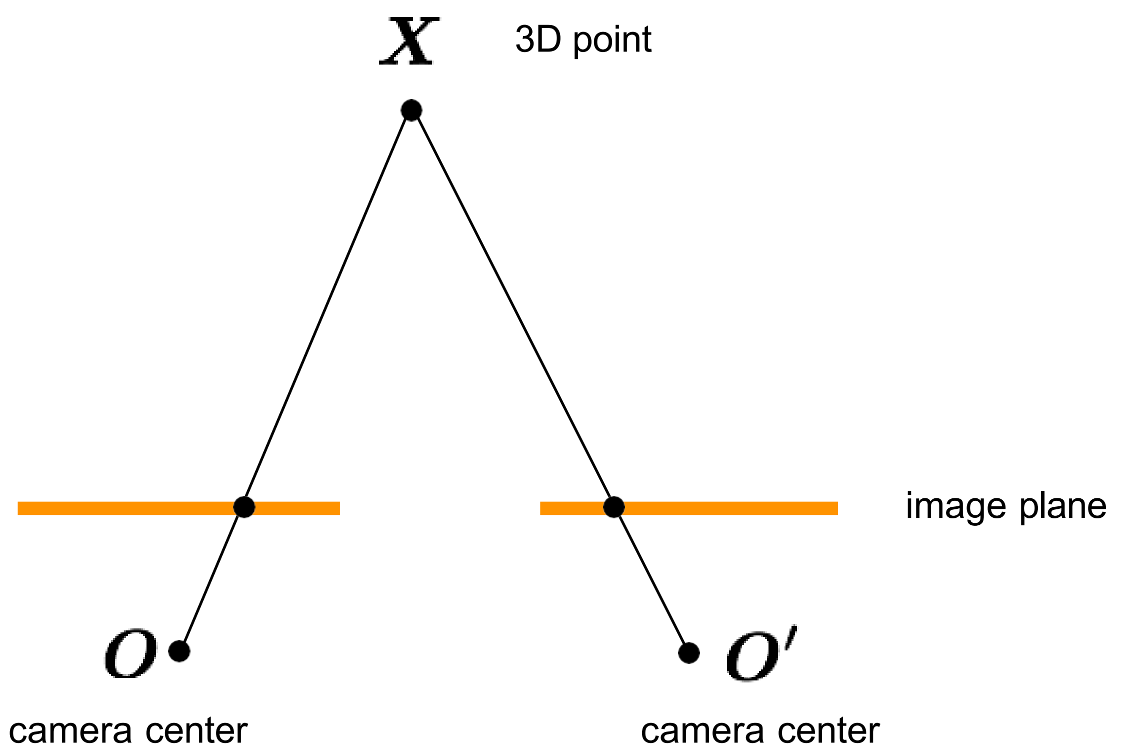
If we could find the **corresponding points** in two images, we could **estimate relative depth**…

Depth from Disparity

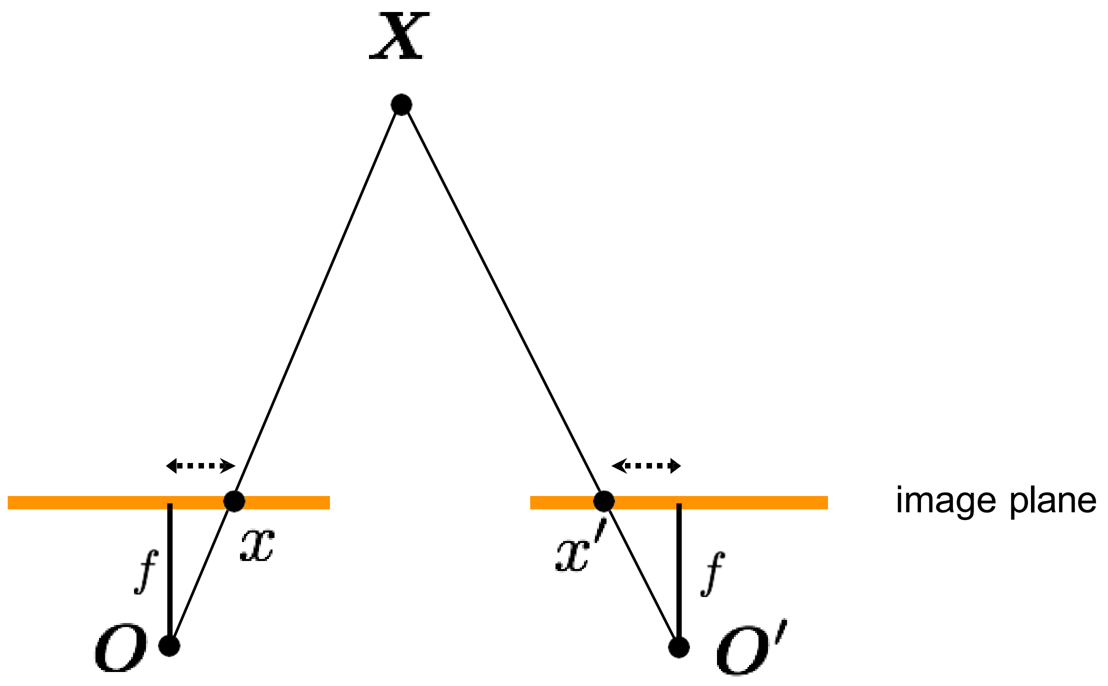


- Assume 2 cameras with CCDs on the **same plane**
- Assume **known correspondence** between images of a 3D point

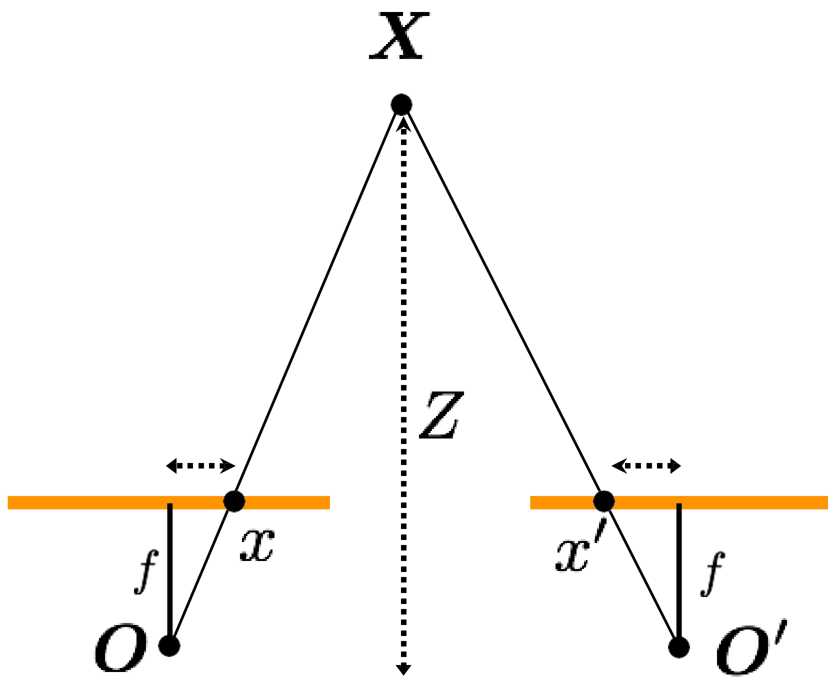
Depth from Disparity



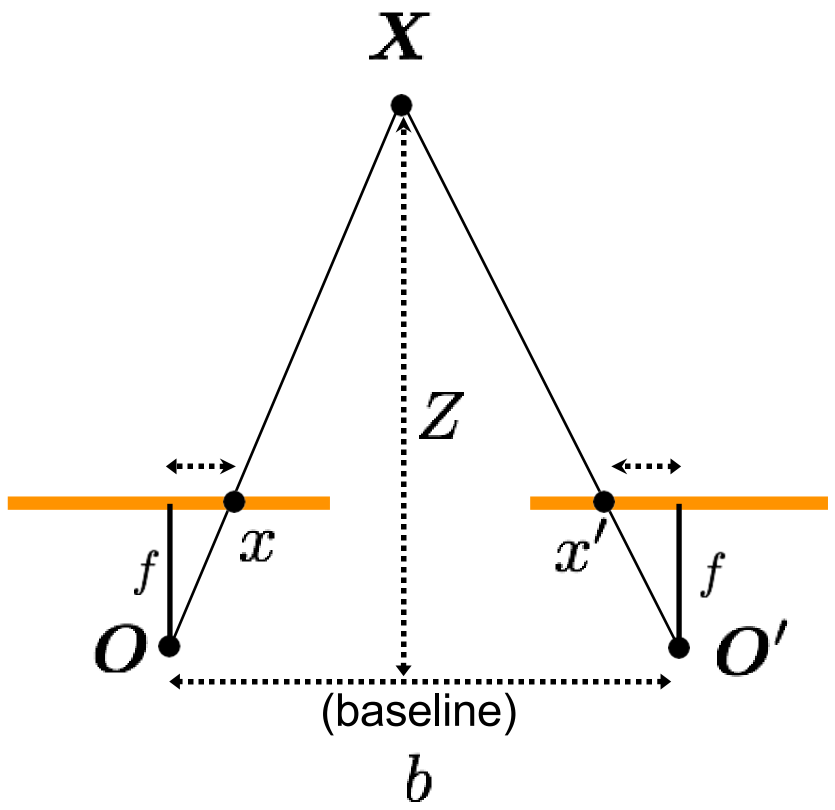
Depth from Disparity



Depth from Disparity

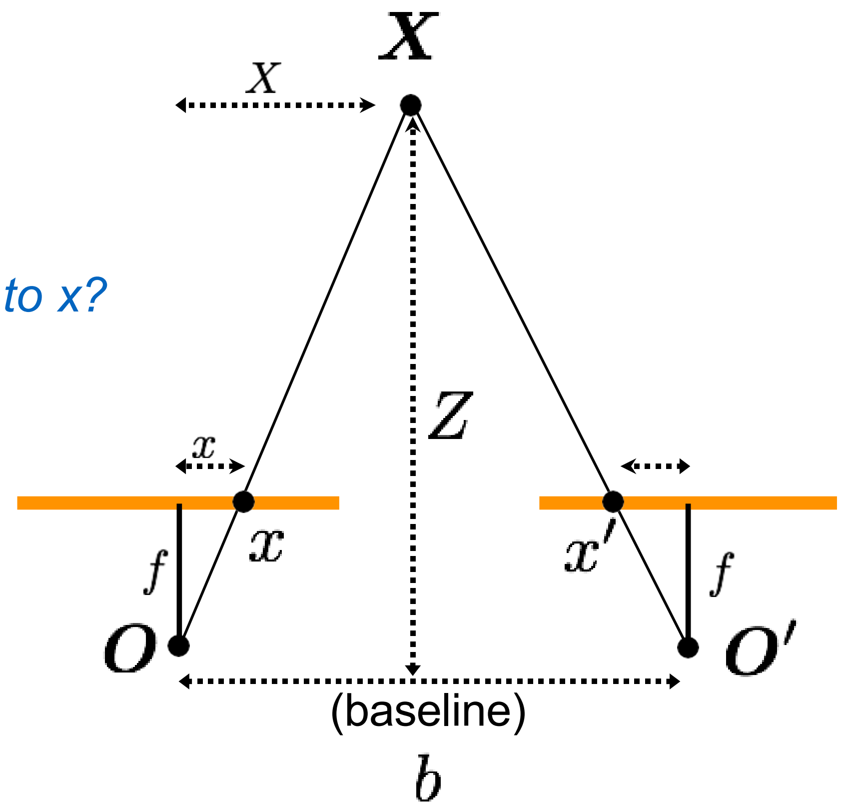


Depth from Disparity



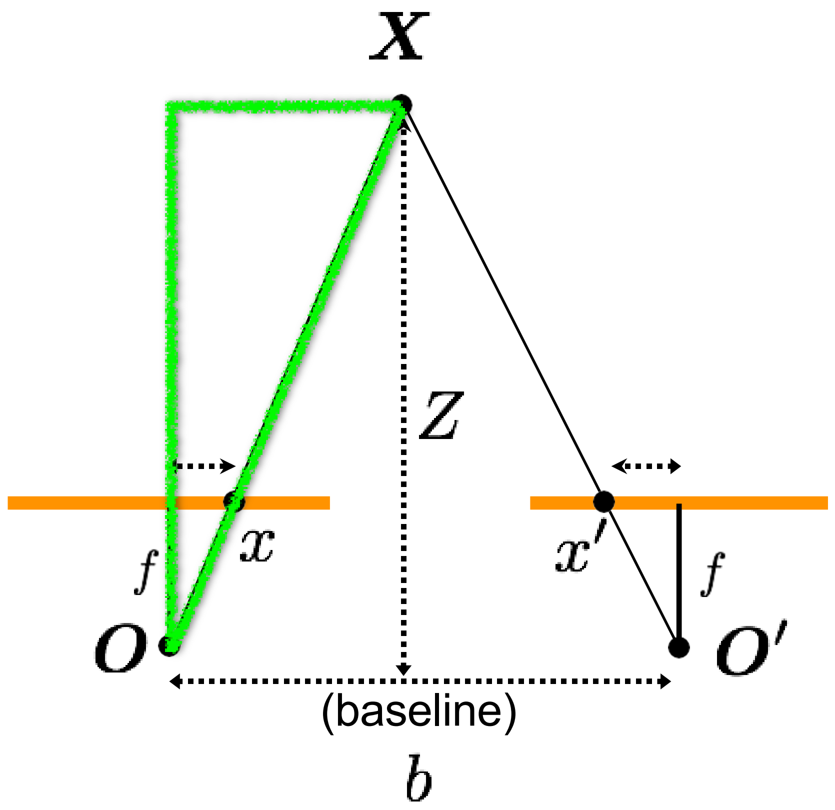
Depth from Disparity

How is X related to x?



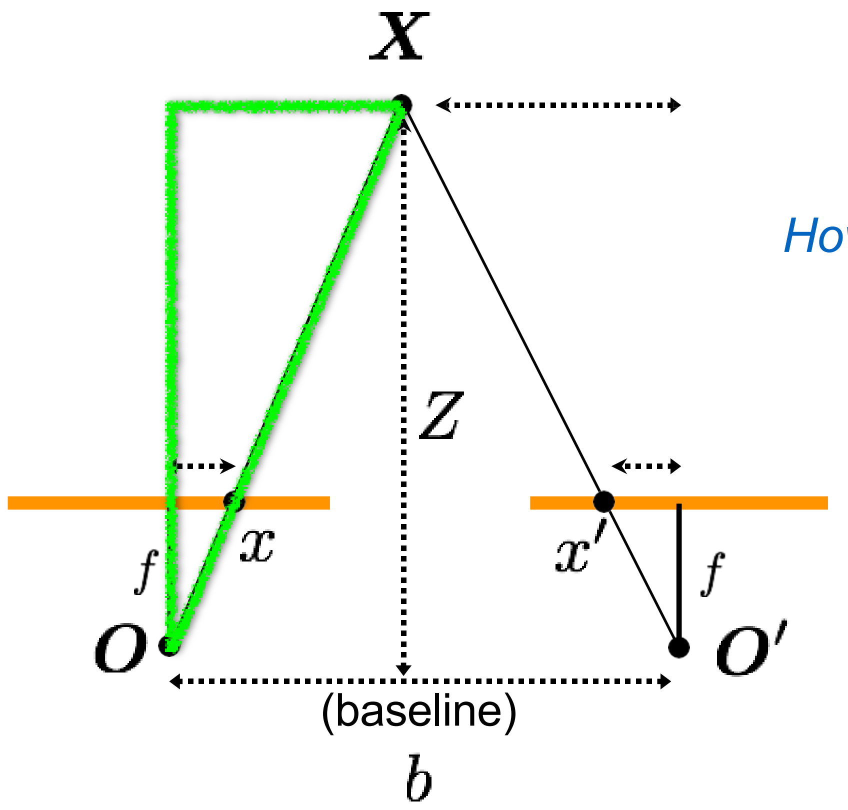
Depth from Disparity

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



Depth from Disparity

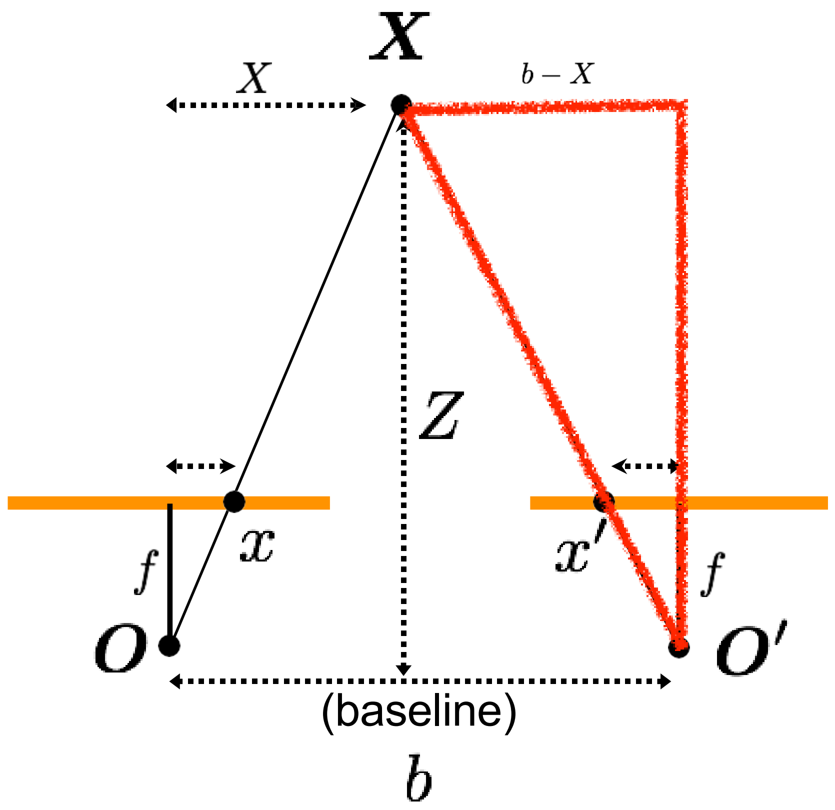
$$\frac{X}{Z} = \frac{x}{f}$$



How is X related to x'?

Depth from Disparity

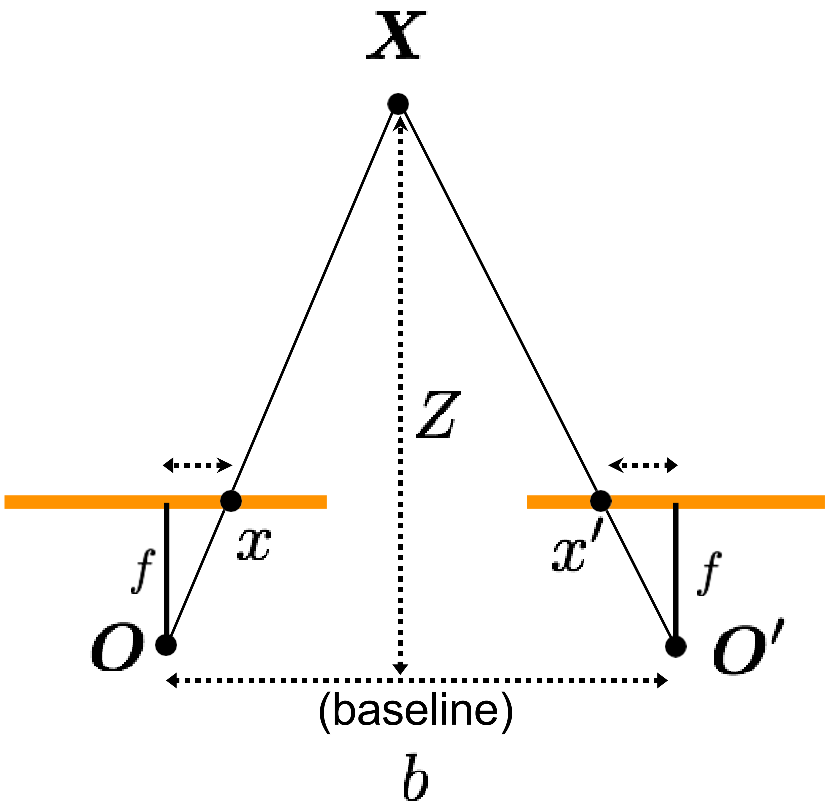
$$\frac{X}{Z} = \frac{x}{f}$$



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

Depth from Disparity

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



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

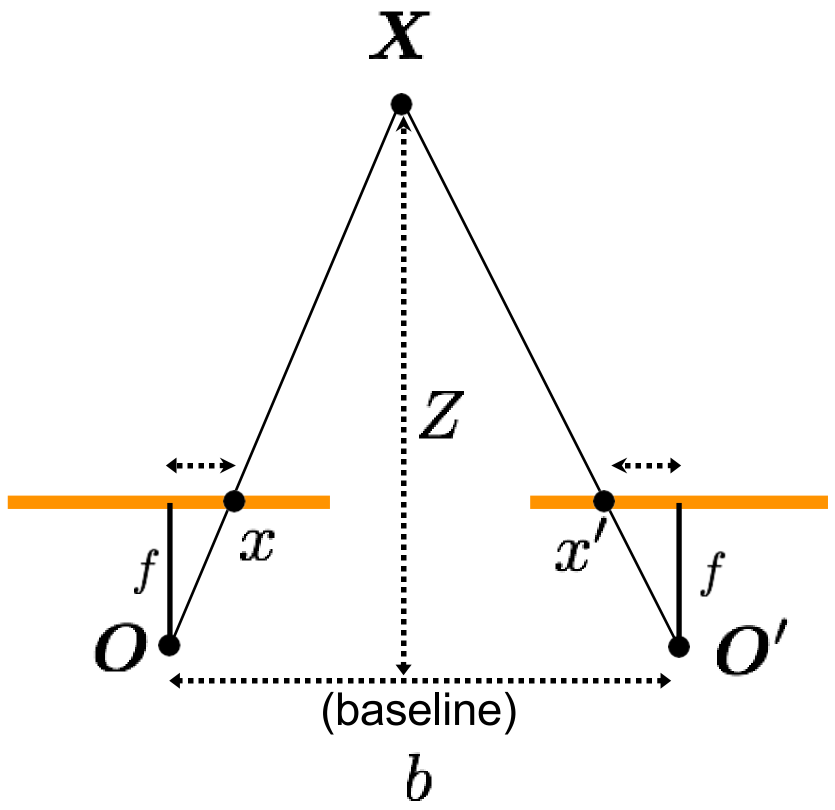
Disparity

$$d = x - x'$$ (wrt to camera origin of image plane)

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

Depth from Disparity

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



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

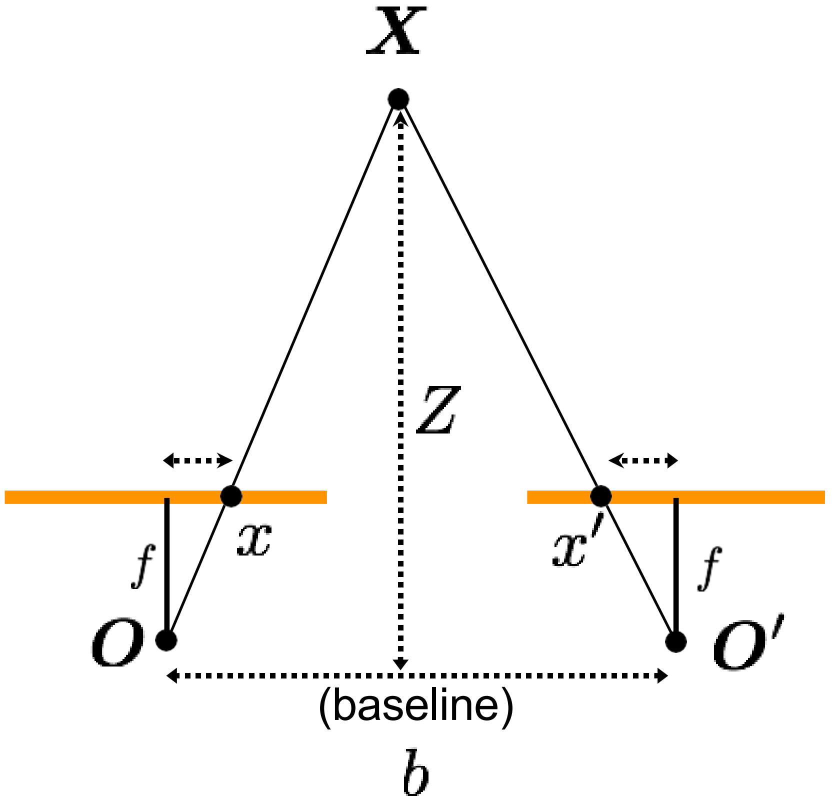
Disparity

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

inversely proportional
to depth

Depth from Disparity

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



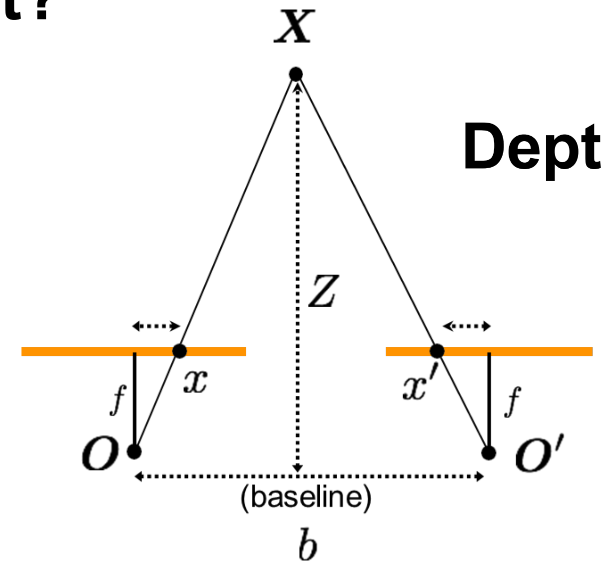
$$\frac{b - X}{Z} = \frac{x'}{f}$$

Depth

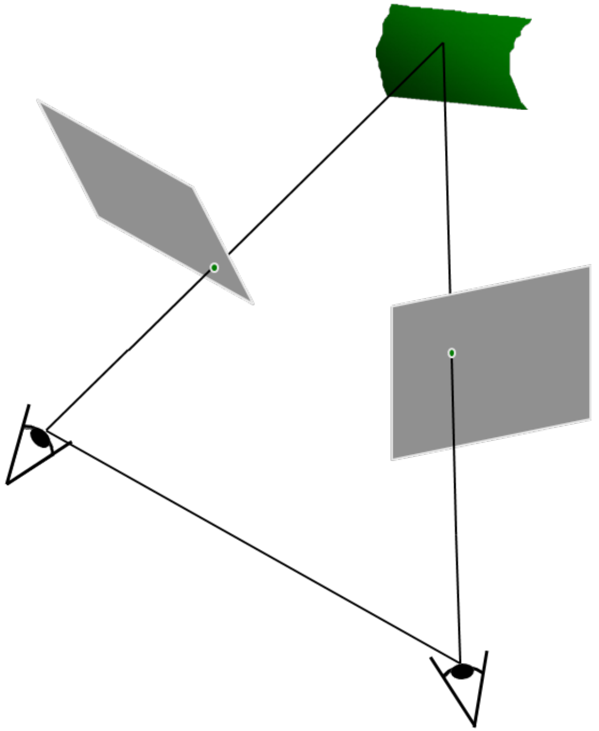
$$Z = \frac{bf}{d}$$

Stereo Vision

- We assume 2 cameras with CCDs on the **same plane**
  - **What if they are not?**



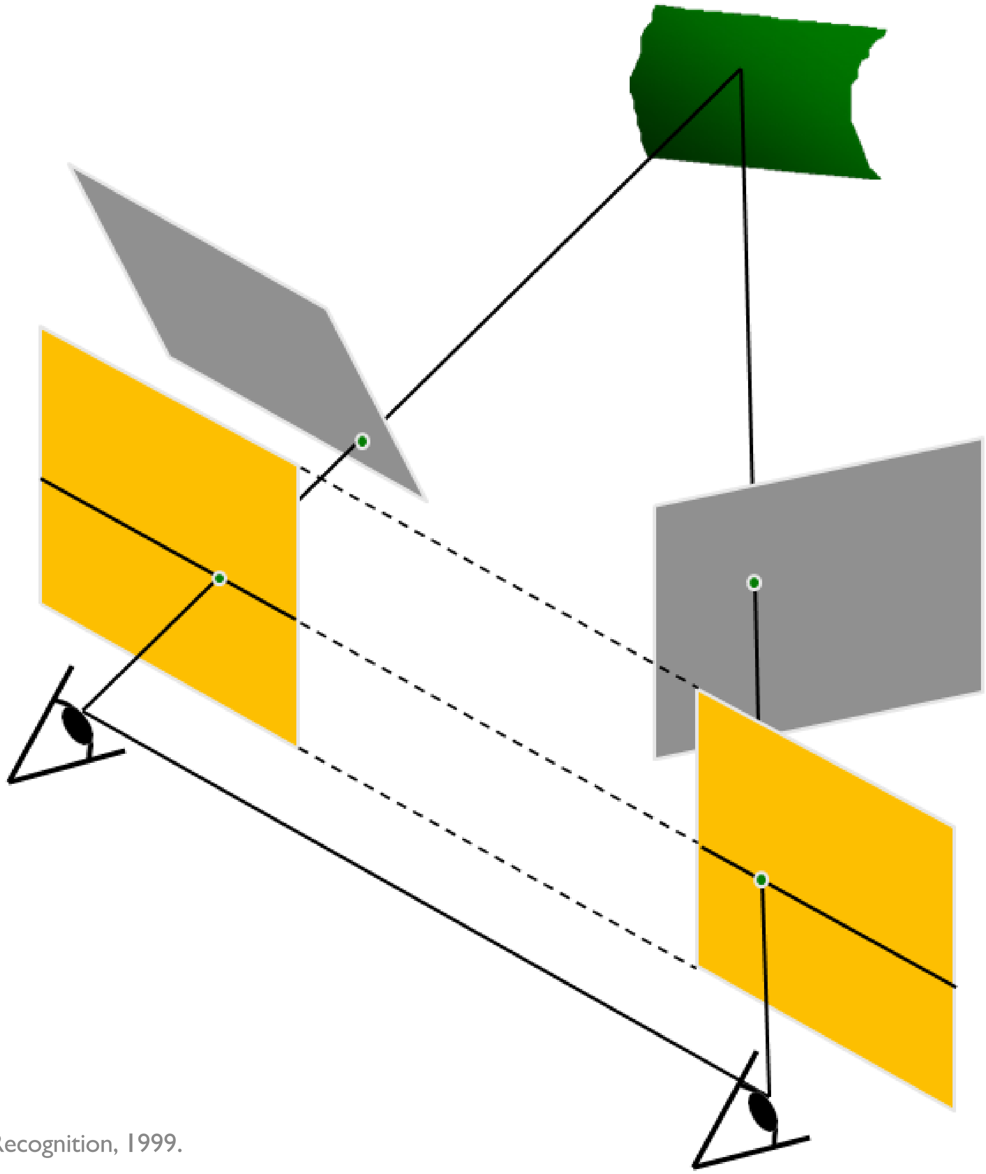
Depth: $Z = \frac{bf}{d}$



- We assumed **known correspondence** between images of 3D points
  - **How to find (correct) correspondences?**

Stereo Image Rectification

- Reproject image planes onto a common plane parallel to the line between camera centers
- Pixel motion is horizontal after this transformation
- Two homographies (3x3 transform), one for each input image reprojection

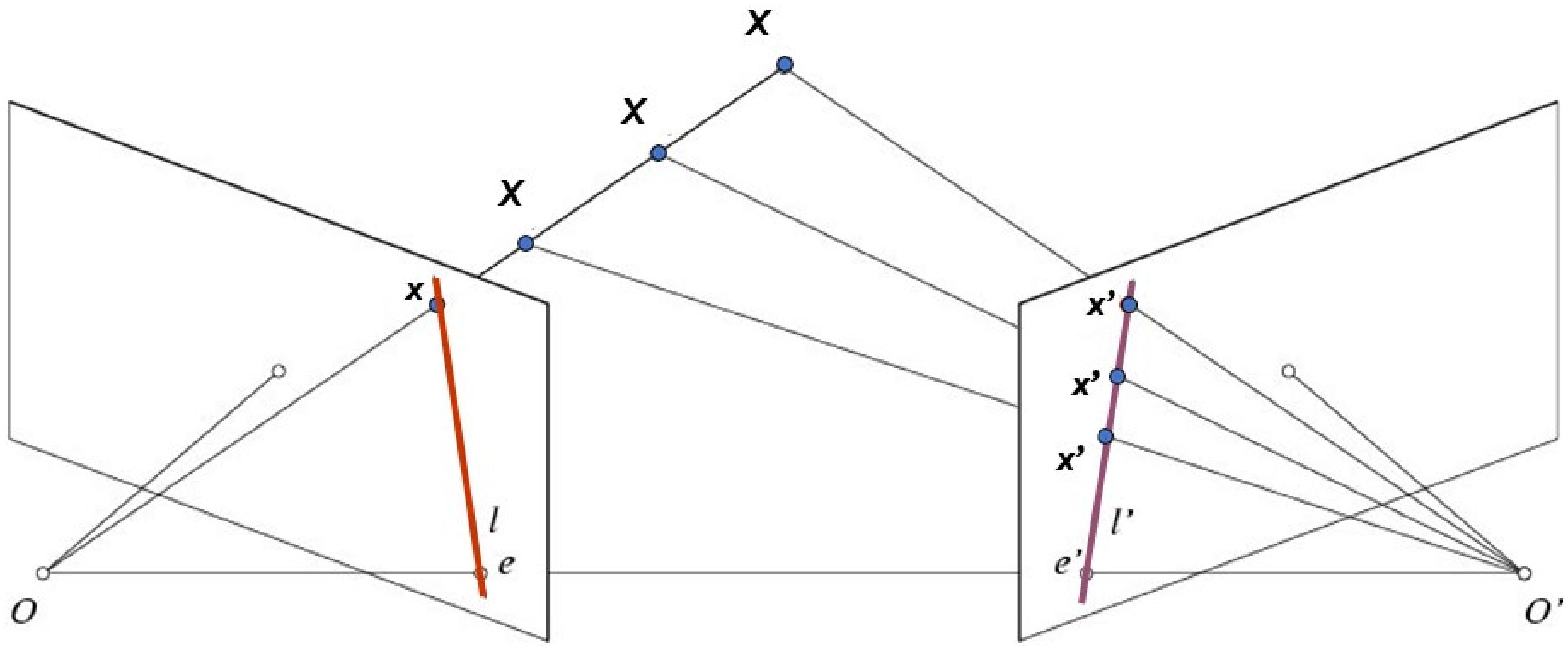


C. Loop and Z. Zhang. [Computing Rectifying Homographies for Stereo Vision](). IEEE Conf. Computer Vision and Pattern Recognition, 1999.

Finding Correspondences using Epipolar Constraint

How to **efficiently** find correspondences?

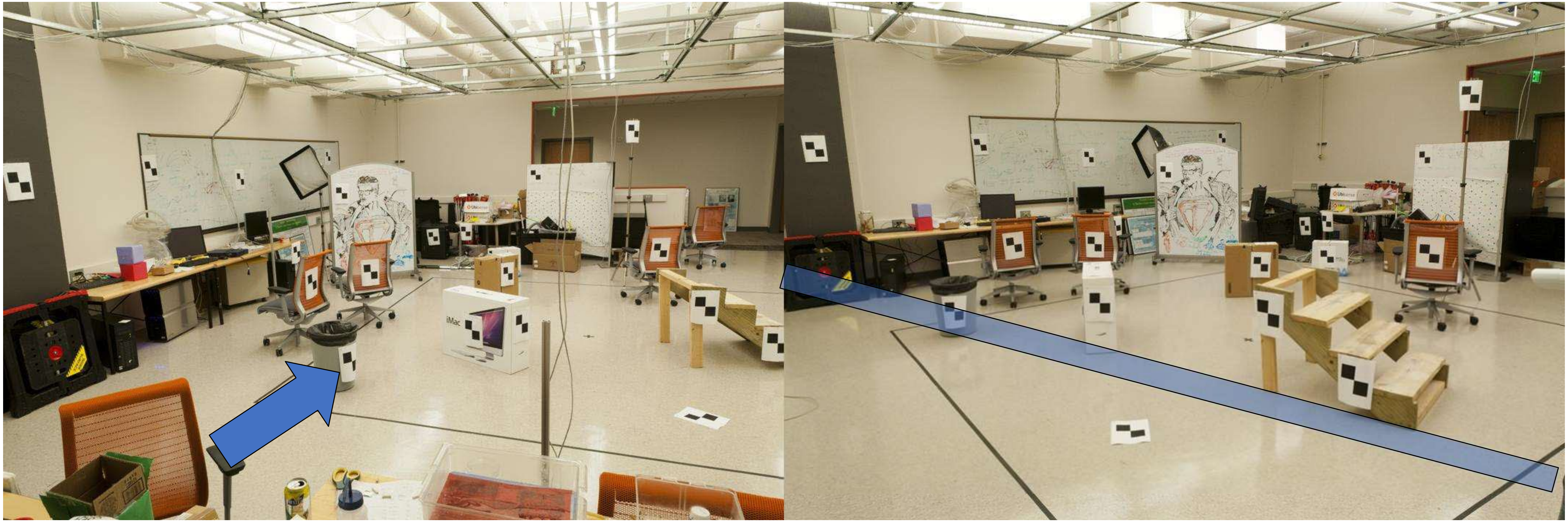
- **Key Idea**: Epipolar Constraint



Potential matches for x' have to lie on the corresponding line l.

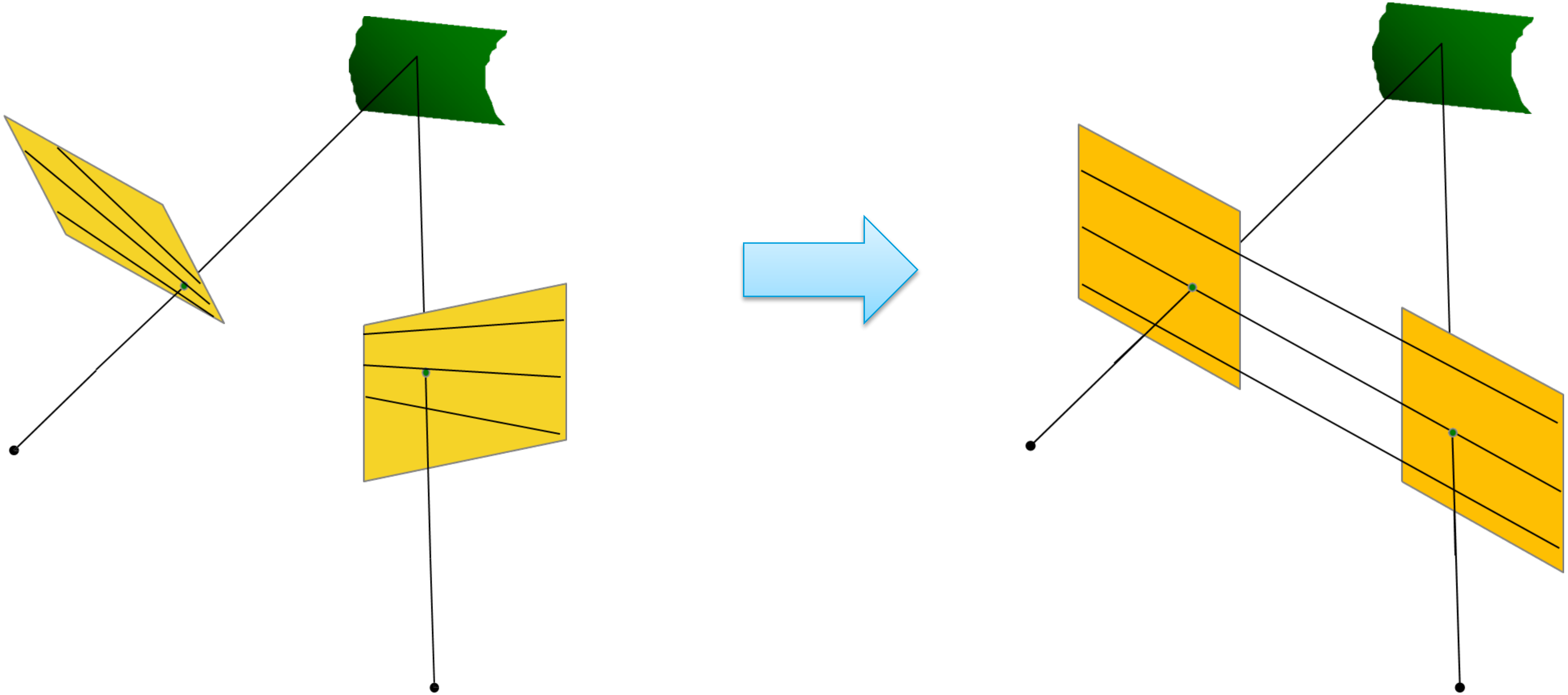
Potential matches for x have to lie on the corresponding line l'.

Finding Correspondences using Epipolar Constraint

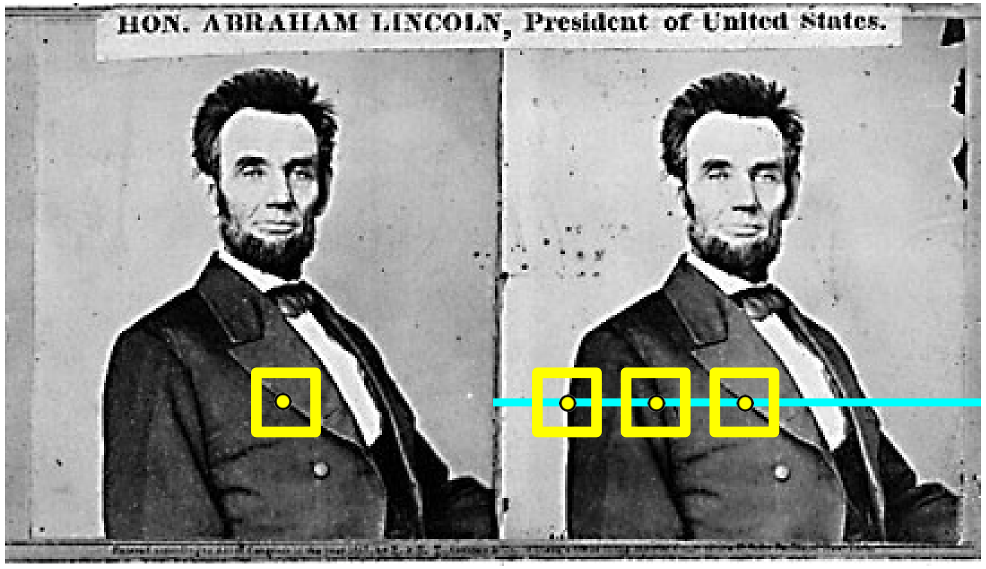


Finding Correspondences using Epipolar Constraint

After stereo rectification, epipolar lines become horizontal:



Stereo Vision



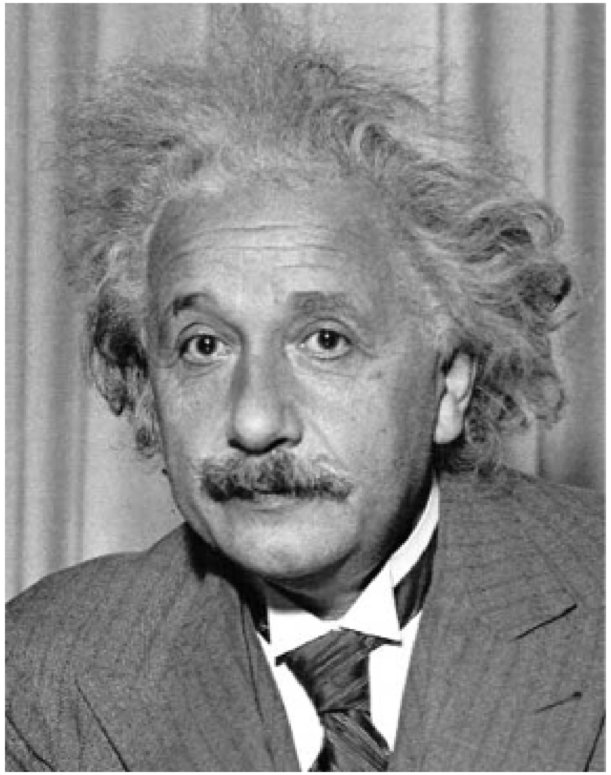
1. Rectify images
   (make epipolar lines horizontal)
2. For each $\boxed{\text{pixel}}$ (or window)
   a. Find epipolar line
   b. Scan line for best match
   c. Compute depth from disparity

How would you do this?

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

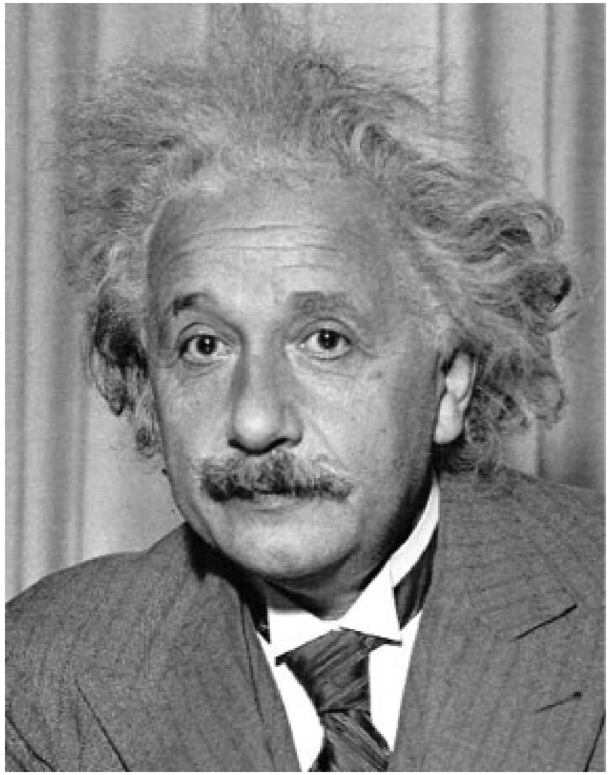
Template Matching

How do we detect the template  in he following image?



Template Matching

How do we detect the template  in he following image?



filter 

output

$$h[m,n] = \sum_{k,l} (g[k,l] - f[m+k, n+l])^2$$

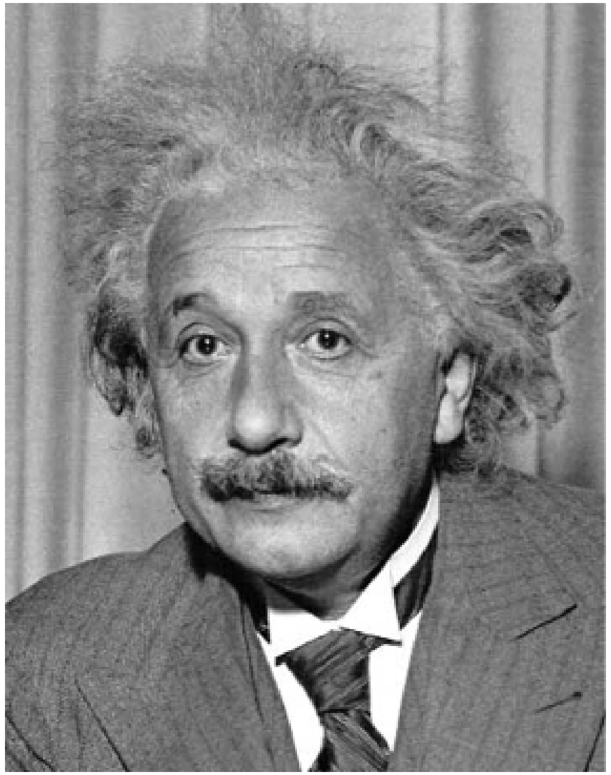
image

What will the output look like?

Solution 3: Use sum of squared differences (SSD).

Template Matching

How do we detect the template  in he following image?

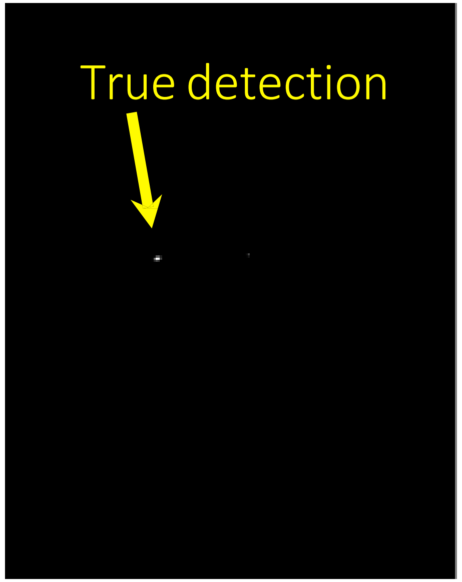


filter 

output

$$h[m,n] = \sum_{k,l} (g[k,l] - f[m+k, n+l])^2$$

image

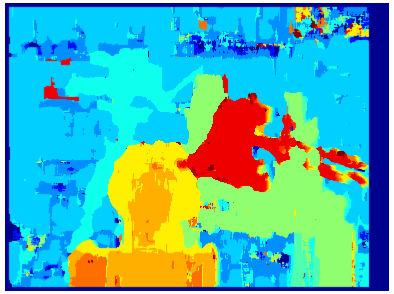


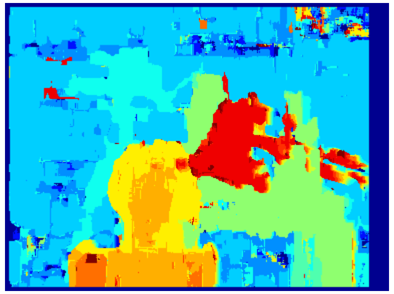
Solution 3: Use sum of squared differences (SSD).

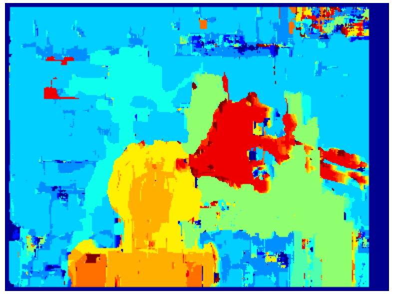
What could go wrong?

Similarity Metrics

Similarity Measure	Formula
Sum of Absolute Differences (SAD)	$\sum_{(i,j)\in W} \lvert I_1(i,j) - I_2(x+i, y+j)\rvert$
Sum of Squared Differences (SSD)	$\sum_{(i,j)\in W} \left(I_1(i,j) - I_2(x+i, y+j)\right)^2$
Zero-mean SAD	$\sum_{(i,j)\in W} \lvert I_1(i,j) - \bar{I}_1(i,j) - I_2(x+i,y+j) + \bar{I}_2(x+i,y+j)\rvert$
Locally scaled SAD	$\sum_{(i,j)\in W} \lvert I_1(i,j) - \frac{\bar{I}_1(i,j)}{\bar{I}_2(x+i,y+j)} I_2(x+i,y+j)\rvert$
Normalized Cross Correlation (NCC)	$\frac{\sum_{(i,j)\in W} I_1(i,j).I_2(x+i,y+j)}{\sqrt[2]{\sum_{(i,j)\in W} I_1^2(i,j).\sum_{(i,j)\in W} I_2^2(x+i,y+j)}}$

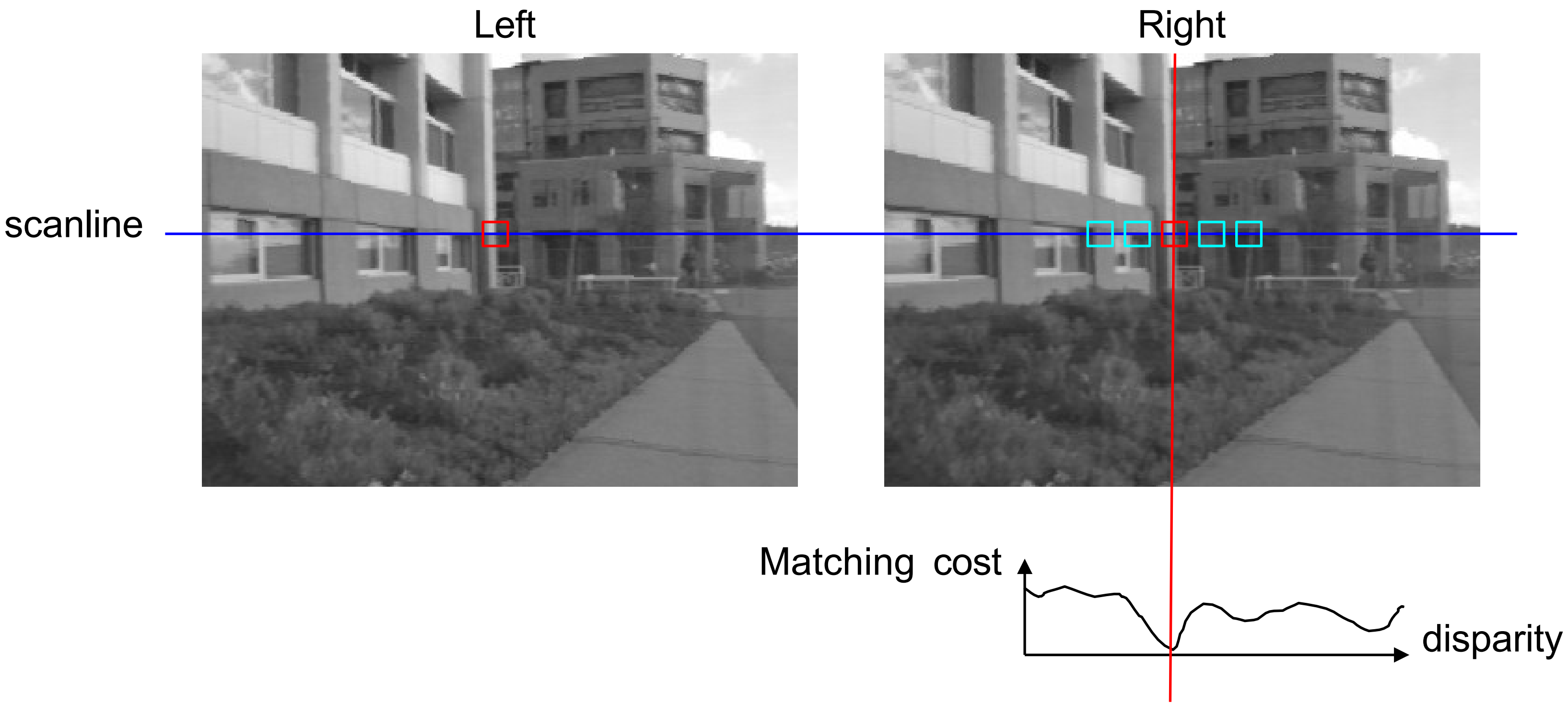

SAD


SSD


NCC

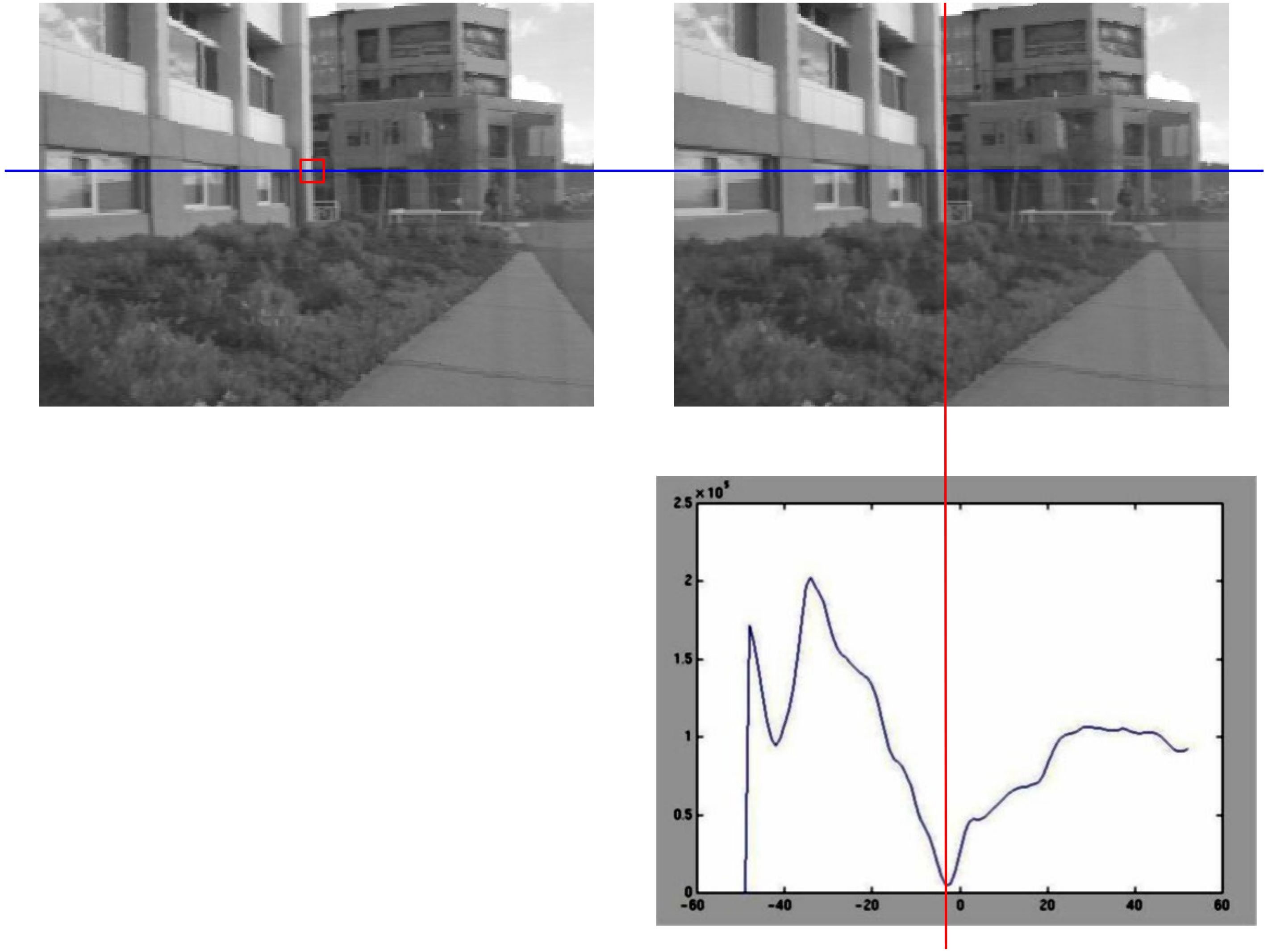

Ground truth

Stereo Block Matching



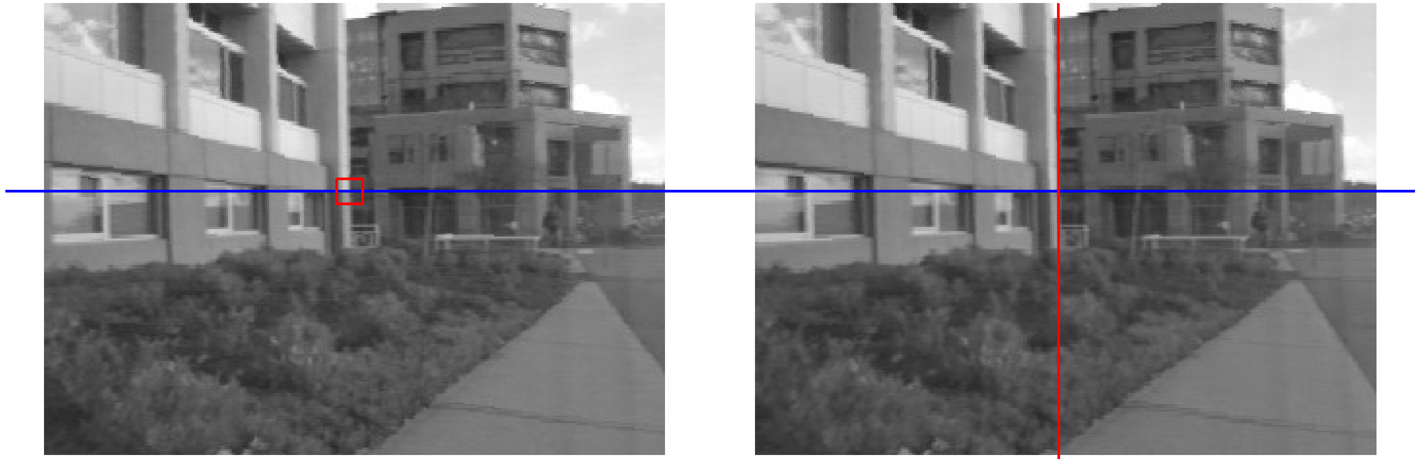
- Slide a window along the epipolar line and compare contents of that window with the reference window in the left image
- Matching cost: SSD or normalized correlation

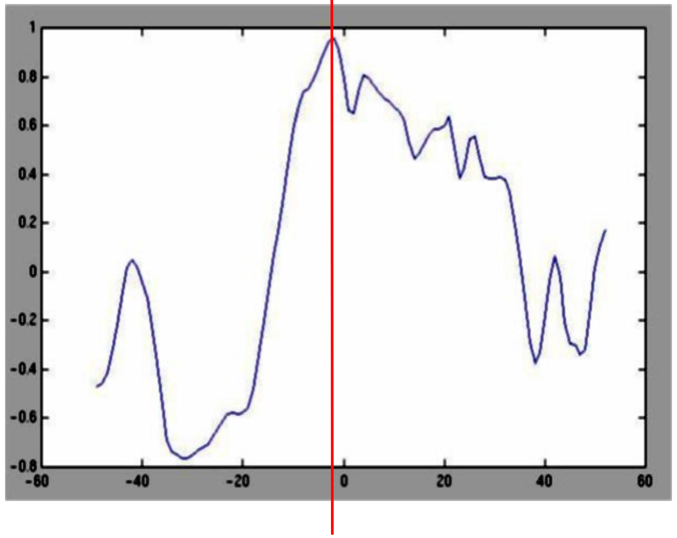
Sum of Squared Differences



SSD

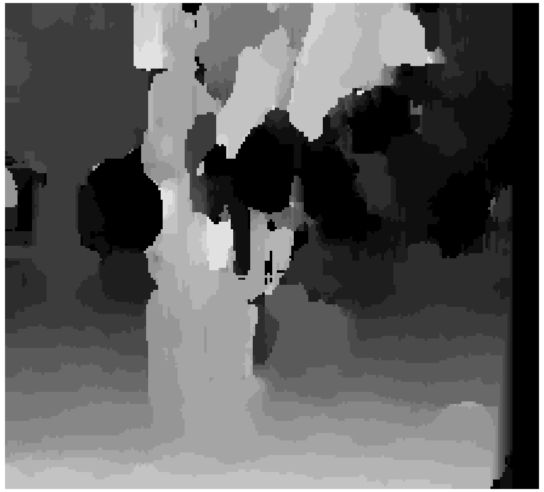
Normalized cross-correlation





Effect of window size

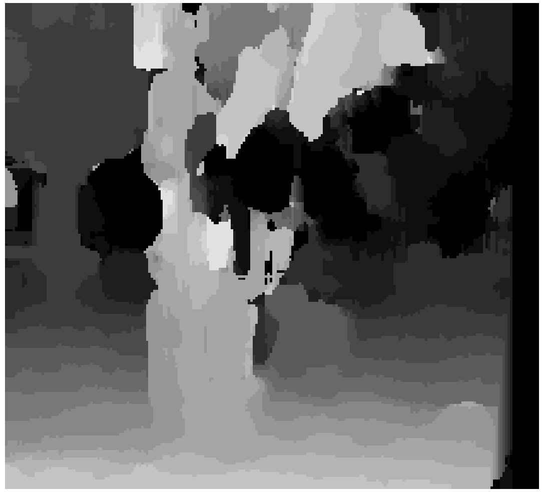
W = 3



W = 20

Effect of window size

W = 3

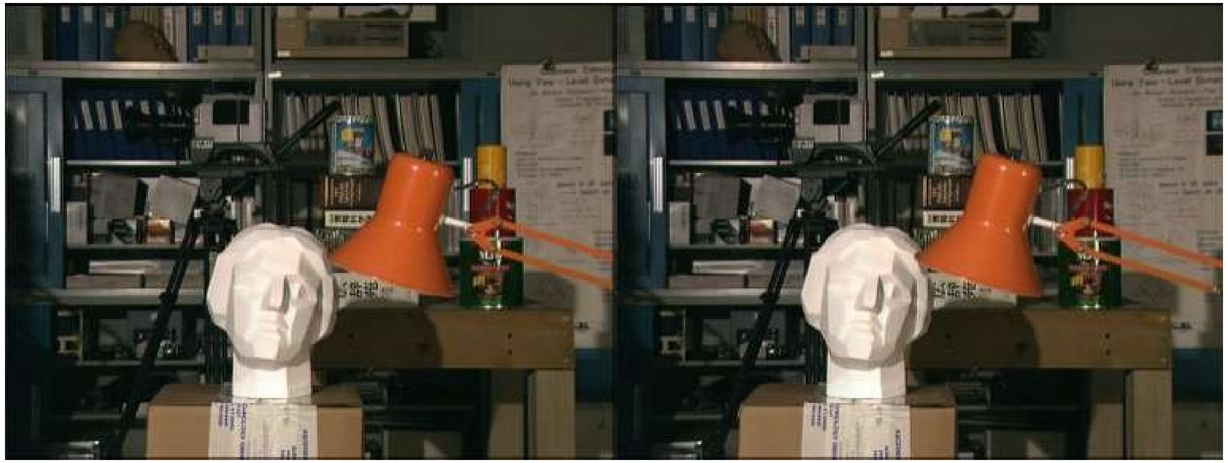


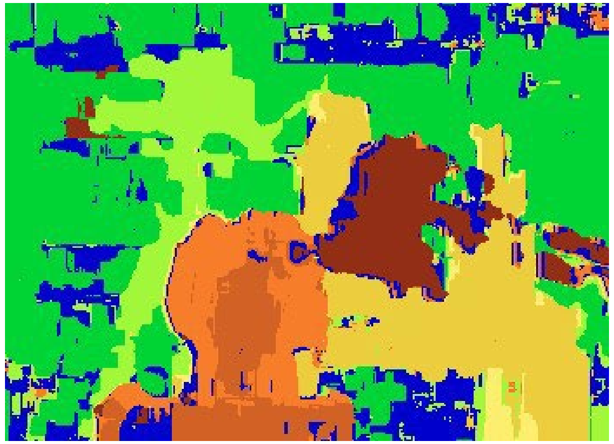
W = 20

Smaller window
- + More detail
- - More noise

Larger window
- + Smoother disparity maps
- - Less detail
- - Fails near boundaries

Tsukuba Test Scene





Window-based matching
(best window size)

'Ground truth'

For the latest and greatest: https://vision.middlebury.edu/stereo/eval3/