Stereo

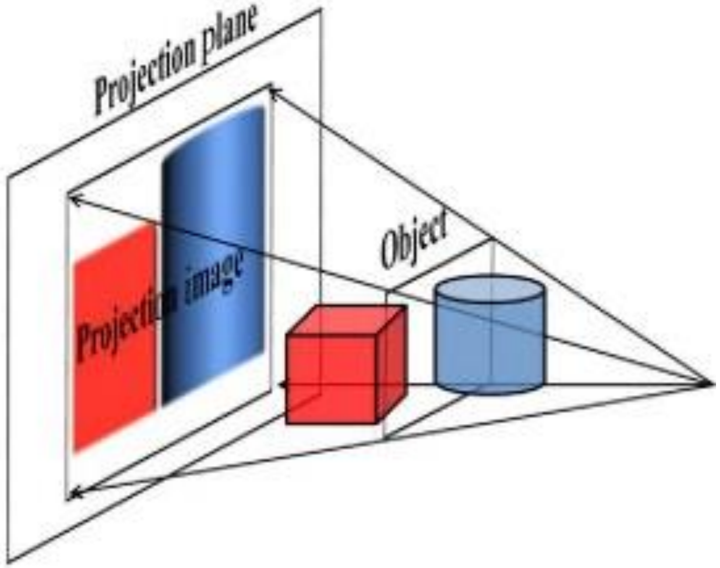
A lot of slides from Noah Snavely +
Shree Nayar’s YT series: First principals of Computer Vision

CS180: Intro to Computer Vision and Comp. Photo
Angjoo Kanazawa & Alexei Efros, UC Berkeley, Fall 2025

Midterm

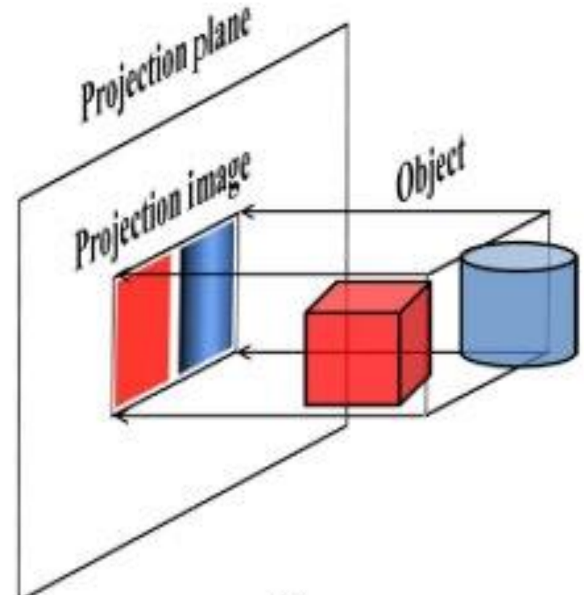
- Everything up to this class
- Focus on the materials of project 0-3

Orthographic Projection



Perspective Projection

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



Scaled Orthographic Projection

$$x = sX$$

- Approximates a very far away object or focal length, s ≅ f/z, with large enough z or f.
- Rays are parallel.
- Used for technical drawings, megical imaging, video games, simple approx in 3D vision

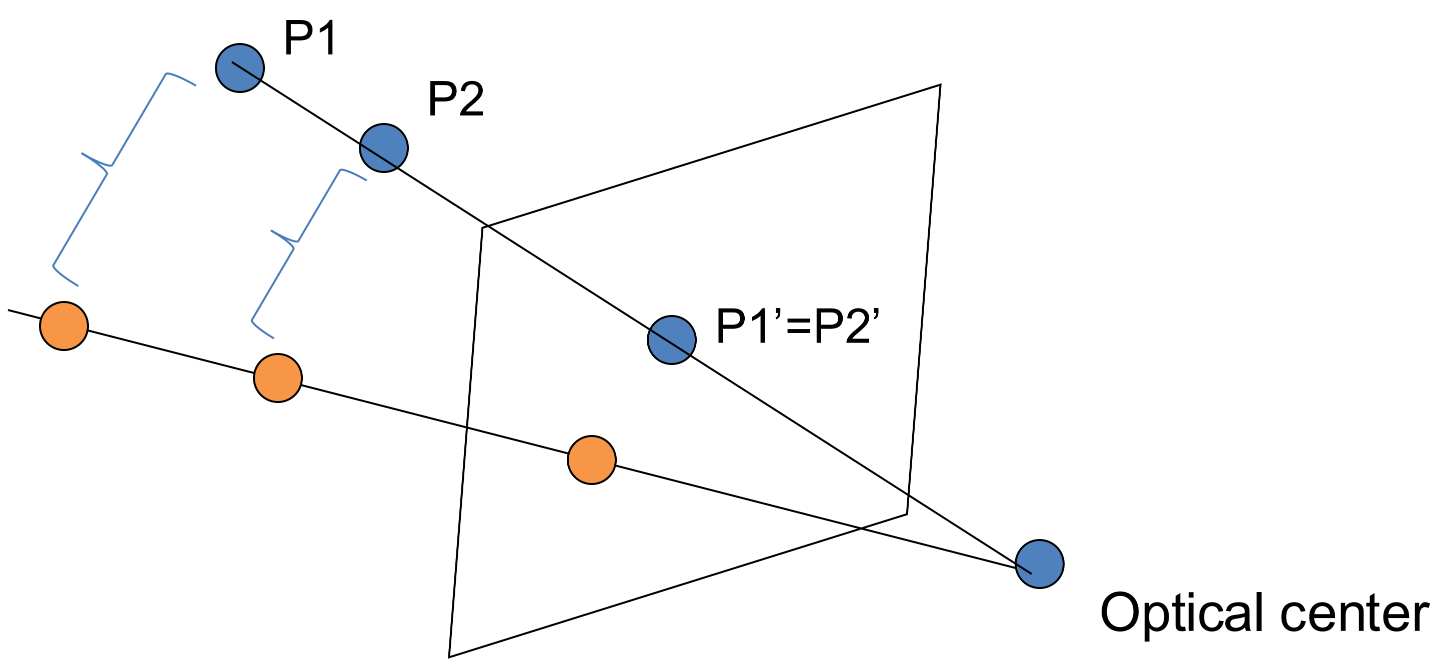
Scale ambiguity

Can you tell exactly how big this diorama is?

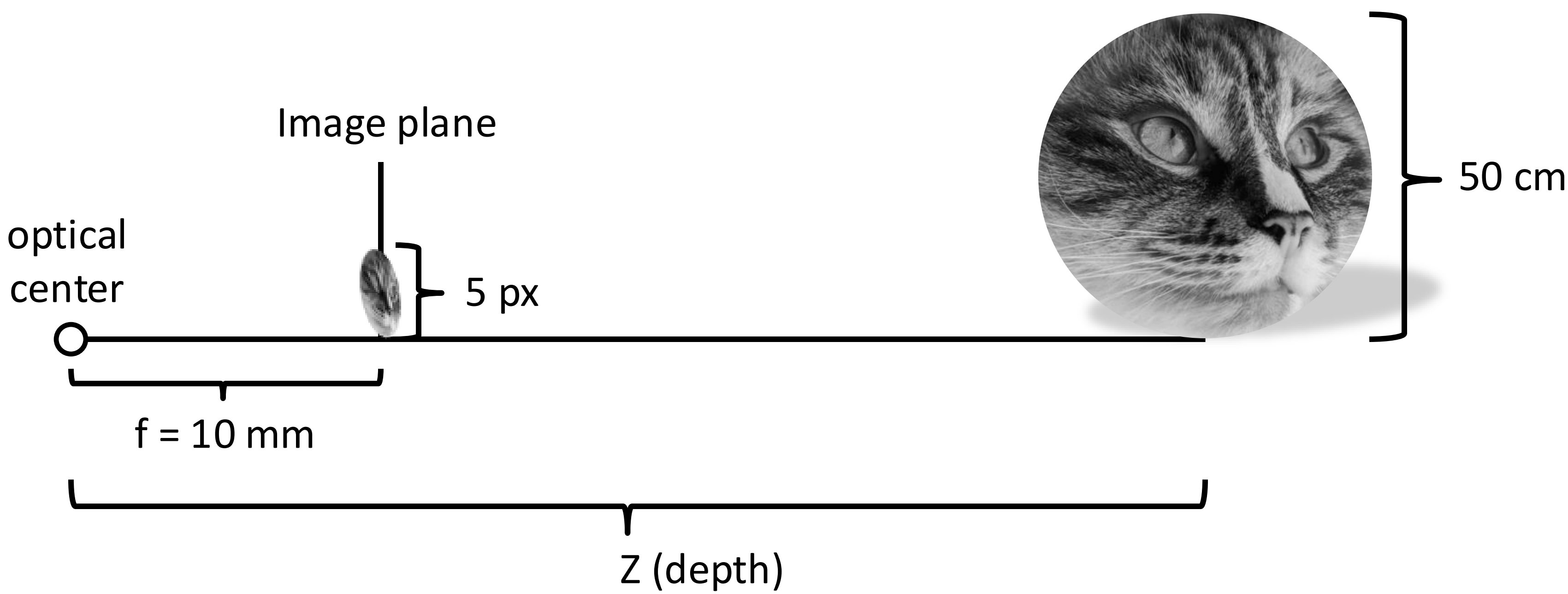
Even if you know the focal length? No, because two different scaled world produce **identical images** even with the same known focal length

We do not know the scale of things even if we know the focal length

- Structure and depth are inherently ambiguous from single views.



You have to know the depths and focal length in order to figure out the size



$$\frac{pixel\ size}{f} = \frac{actual\ size}{Z}$$

You need to know 3 parameters to figure out the 4th ... This is why vision is hard!

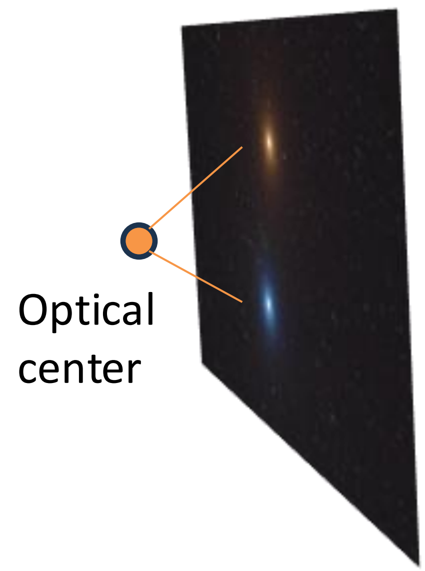
What does focal length give you?

What is the angle between these stars relative to you?

What does focal length give you?

What is the angle between these stars relative to you?

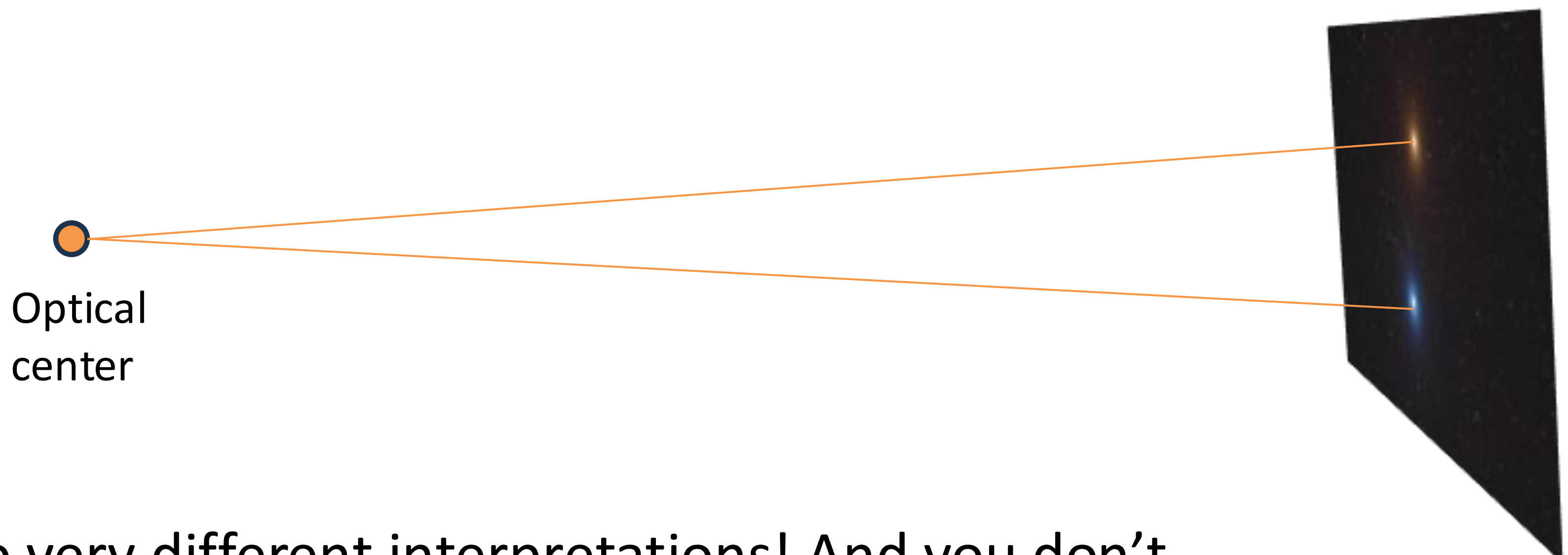
If focal length is very small =
- wide angle camera
- i.e. this could be covering the entire sky
- Angle between the star is wide



What does focal length give you?

What is the angle between these stars relative to you?

If focal length is very large =
- Almost orthographic
- Angle between the star is narrow



Two very different interpretations! And you don't know unless you know/guess the focal length

Big picture

- We know the projective geometry now
- Now lets use two cameras (stereo) to estimate the geometry!
- Assume the projection matrix is known (K,R,T)
- Goal: Compute depth of every point in each image

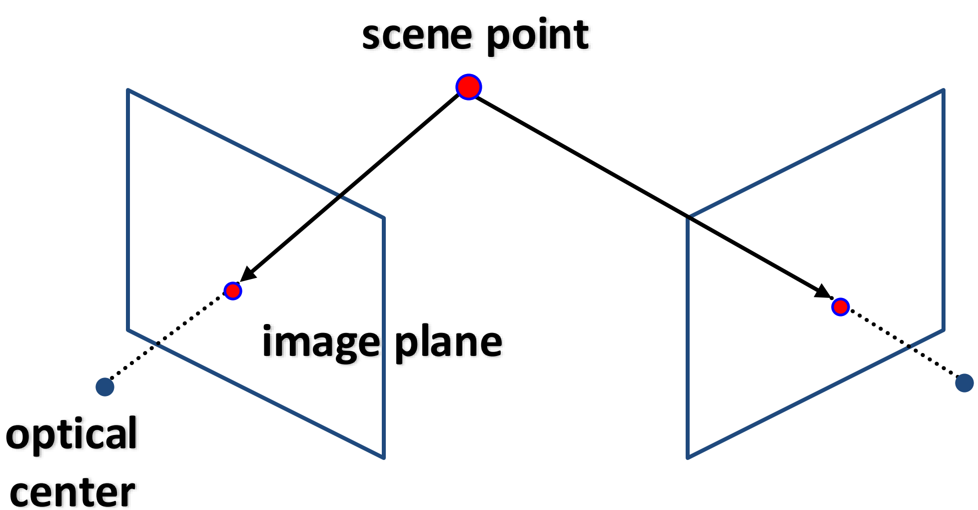
Stereo vision

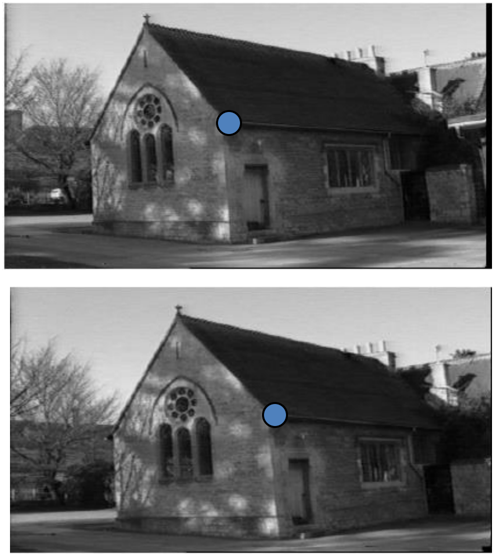
Two cameras, simultaneous views

Single moving camera and static scene

Estimating depth with stereo

- **Stereo**: shape from “motion” between **two** views
- We’ll need to consider:
  - 1. Camera pose (“calibration”) – assume known for now
  - **2. Image point correspondences**





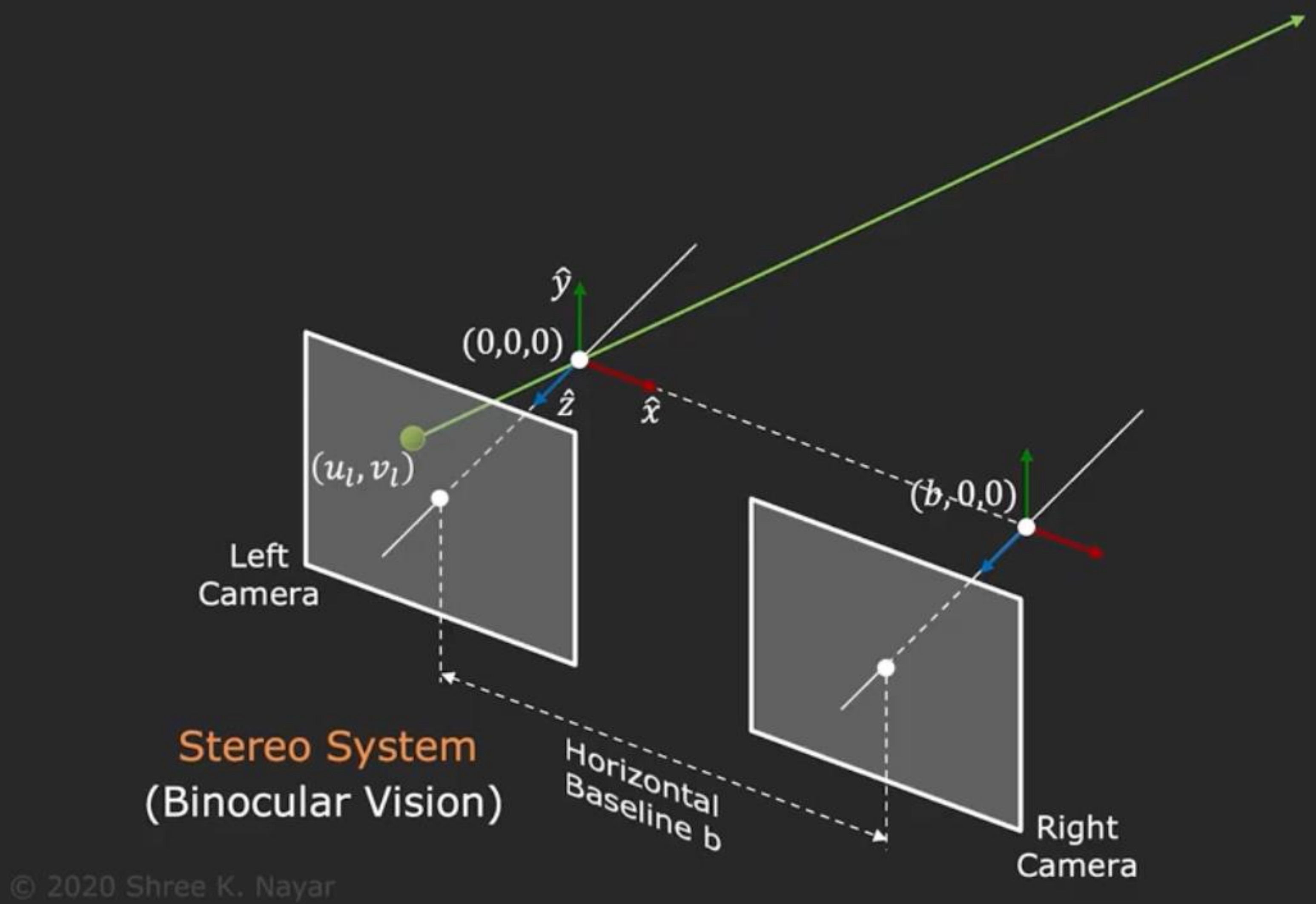
Simple Stereo Setup

- Assume **parallel** optical axes
- Two cameras are calibrated
- Find relative depth

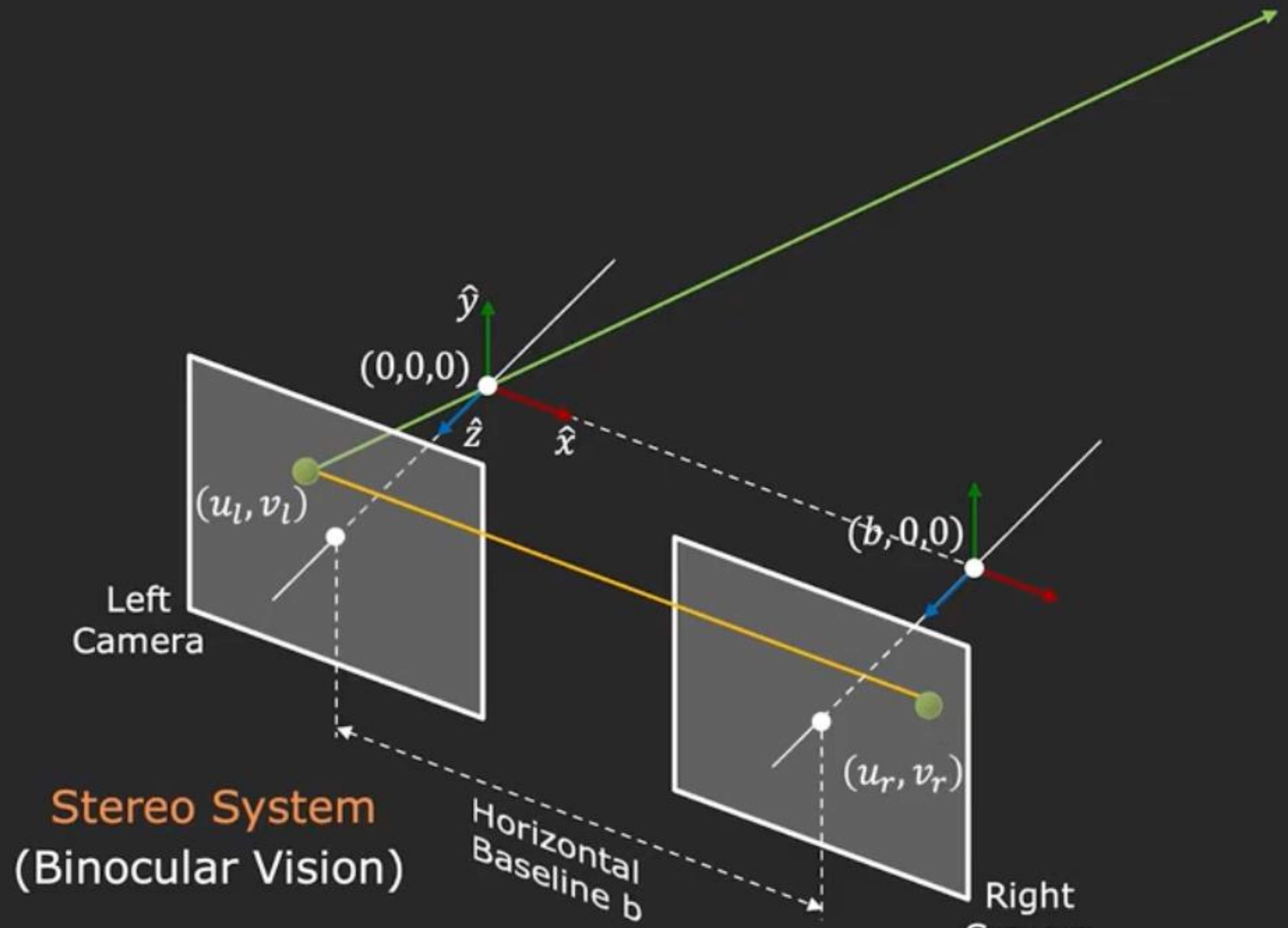
Key Idea: difference in corresponding points to understand shape

Slide credit: Noah Snavely

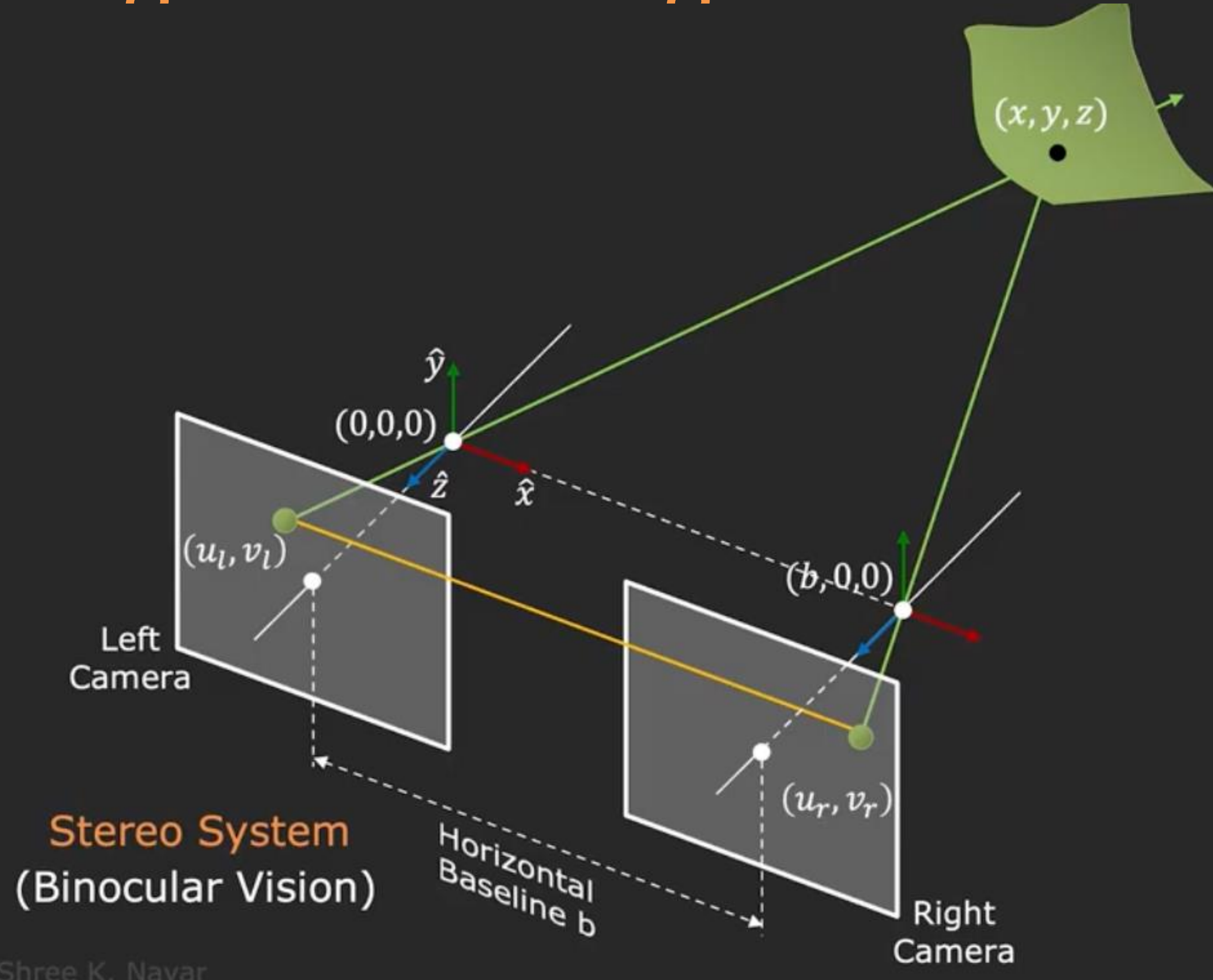
Triangulation using two cameras



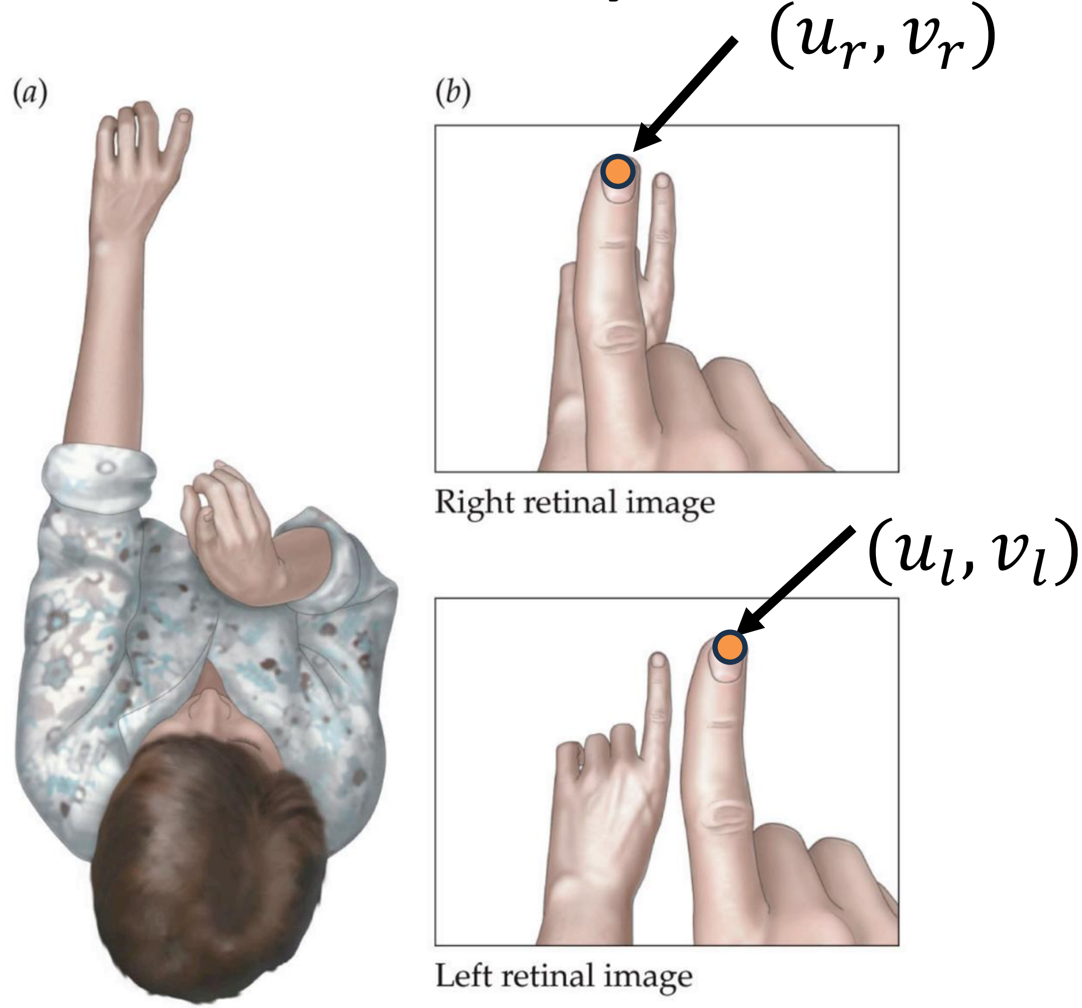
Triangulation using two cameras



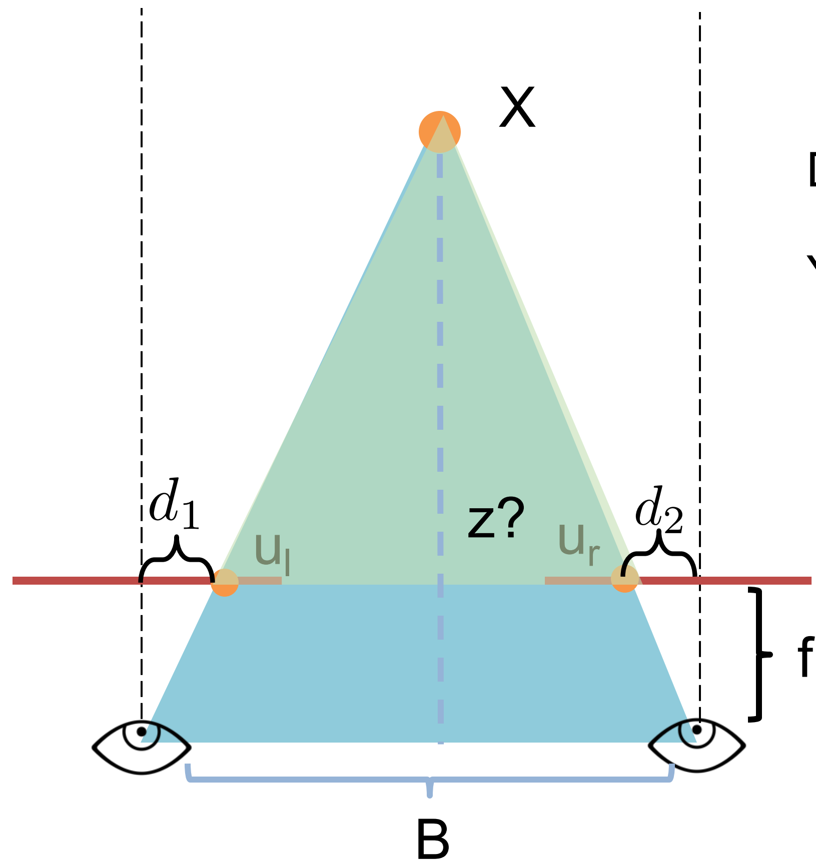
Triangulation using two cameras



We are equipped with binocular vision. Let's try!



Solving for Depth in Simple Stereo



Do we have enough to know what is Z?

Yes, similar triangles!

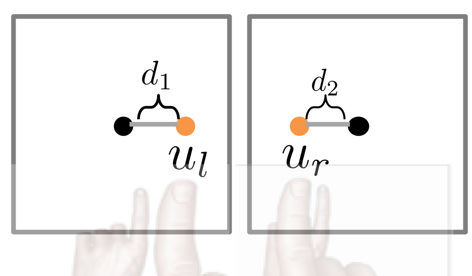
$$\frac{B-(u_l-u_r)}{z-f} = \frac{B}{z}$$

$$z = \frac{fB}{\boxed{u_l-u_r}}$$

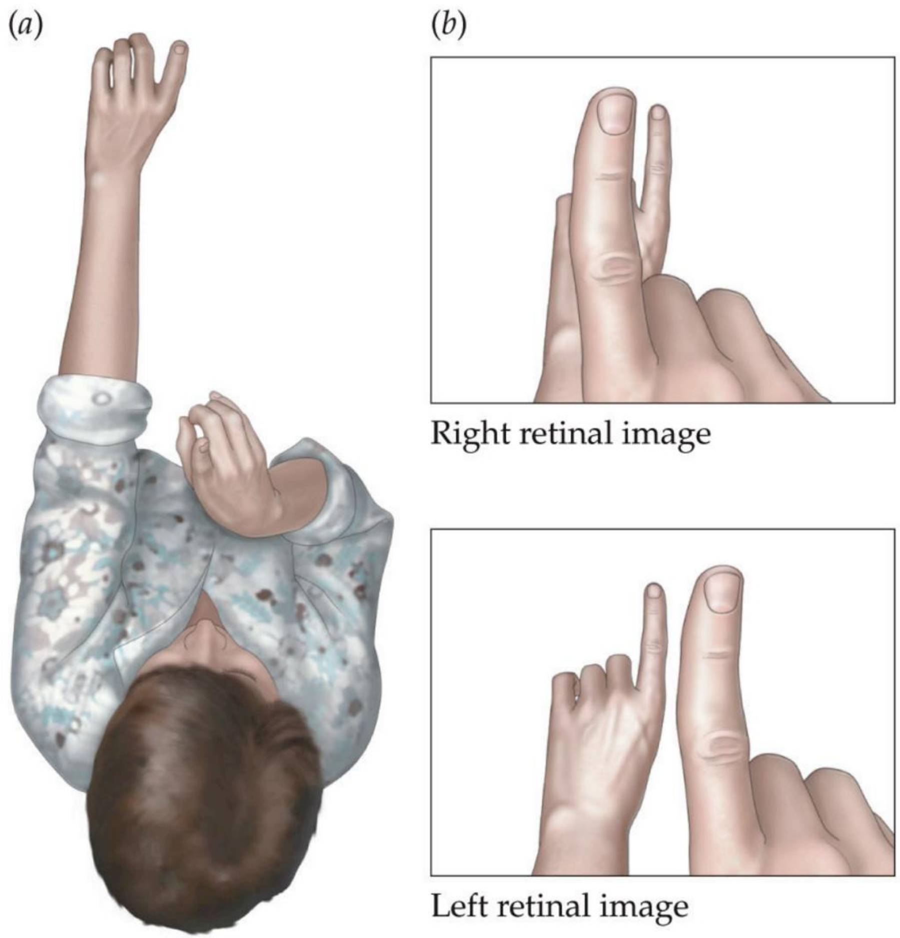
disparity (how much corrsp. pixels move)

Base of : $B-(d1+d_2)$

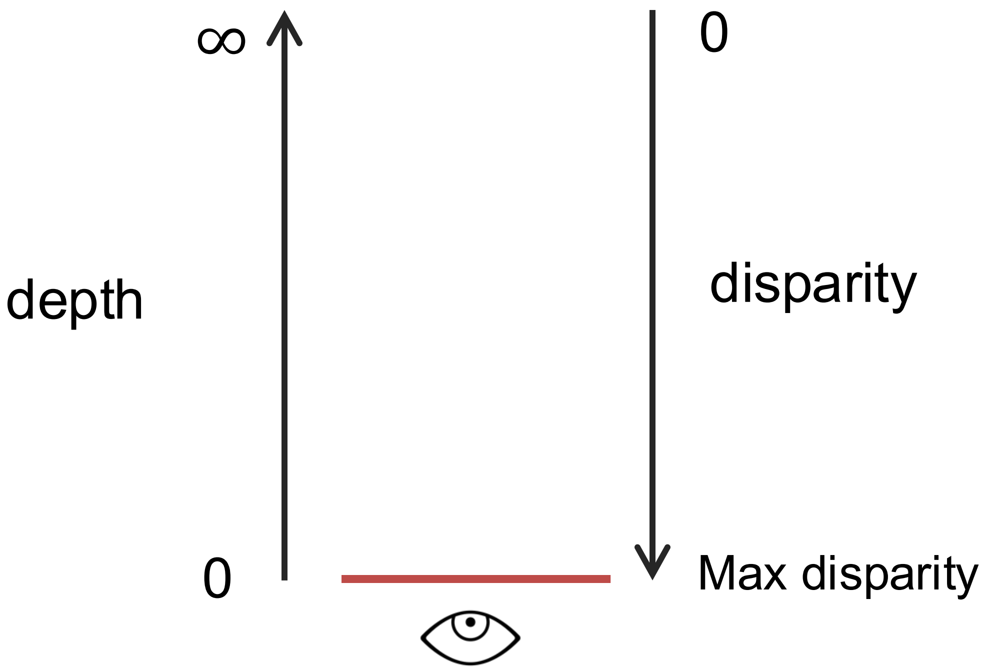
in image coordinates: $= B-(u_l-u_r)$



Try with your hands!



Depth is inversely proportional to disparity



$$z = \frac{fB}{\boxed{u_l - u_r}}$$

disparity

$$z \propto \frac{1}{u_l - u_r} = \frac{1}{d}$$

what is the disparity of the closer point?
what is the disparity of the far away point?

Disparity gives you the depth information!

Try again

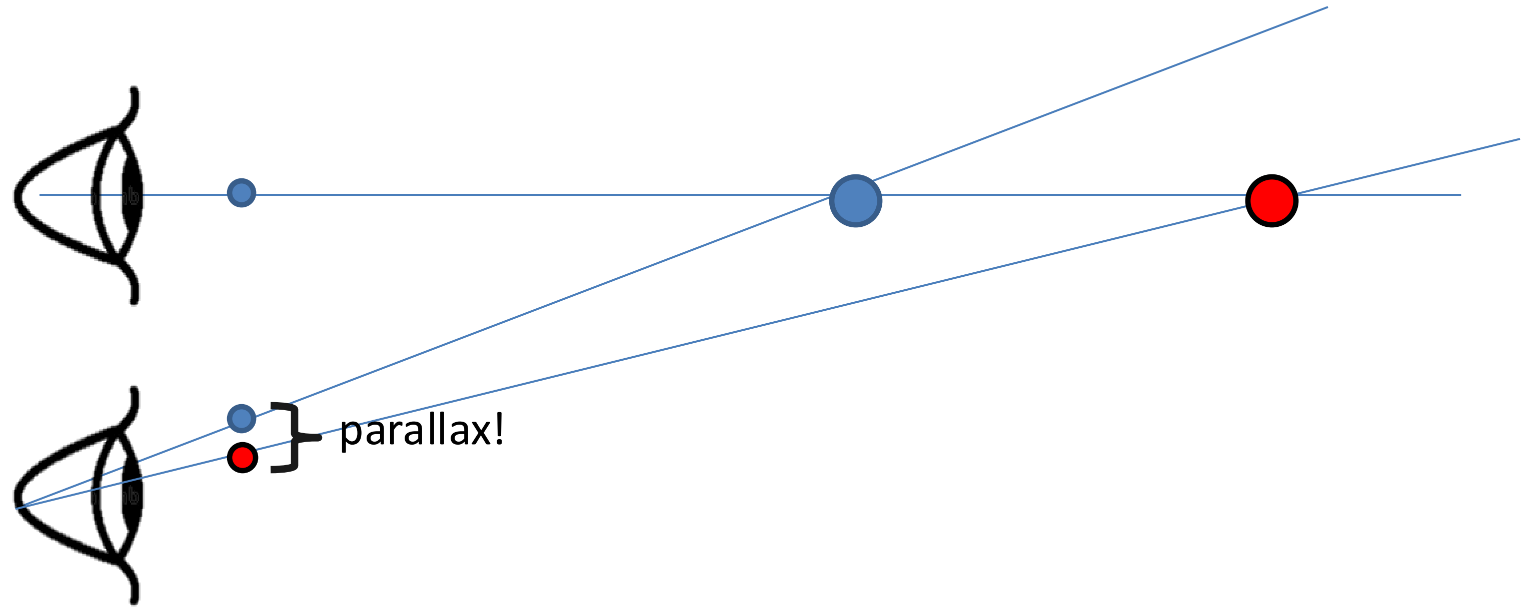
1. Setup so your fingers are on the same line of sight from one eye
2. Now look in the other eye

They move!

Relative displacement is higher as the relative distance grows

== Parallax

Parallax



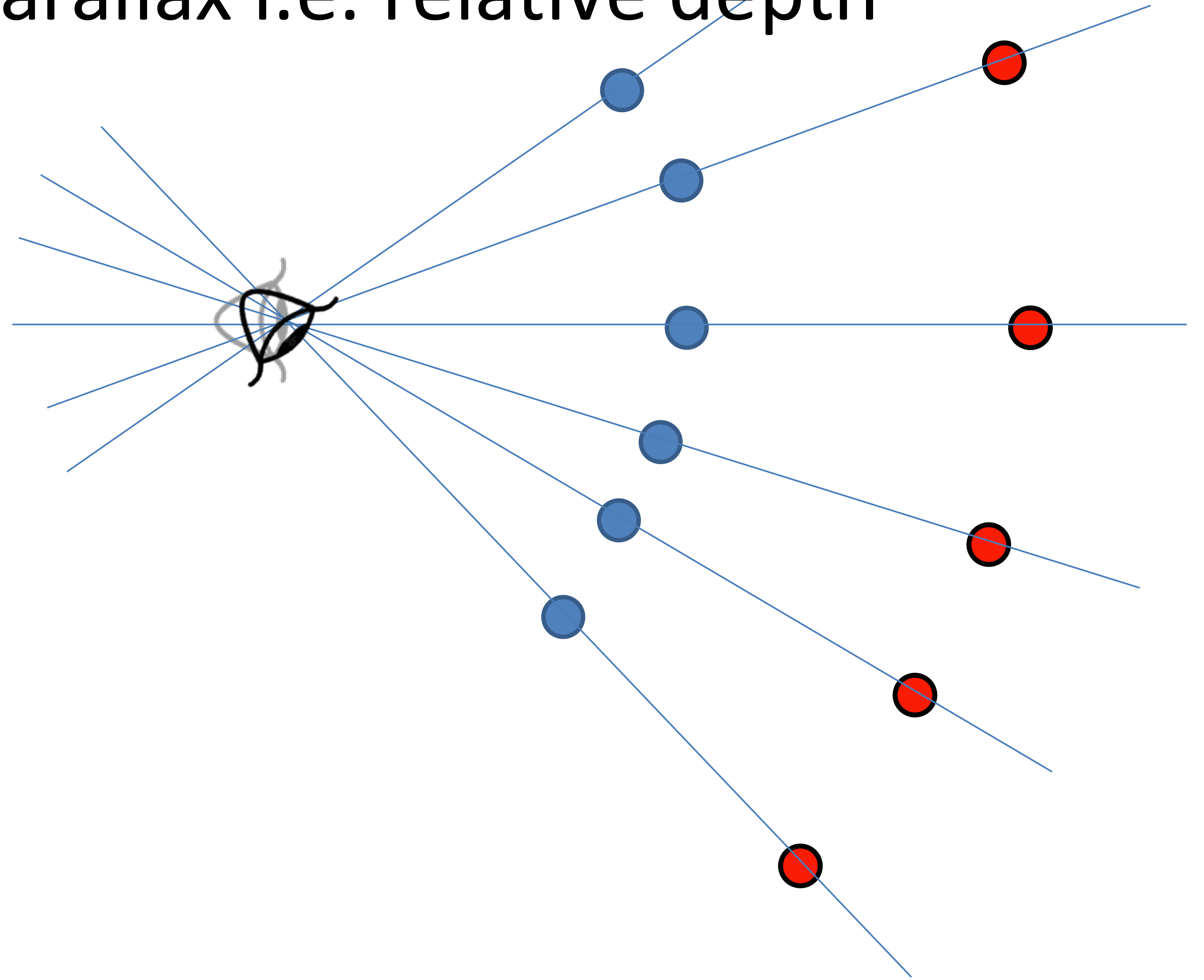
Parallax = *from ancient Greek parállaxis*

= *Para* (side by side) + *allássō*, (to alter)

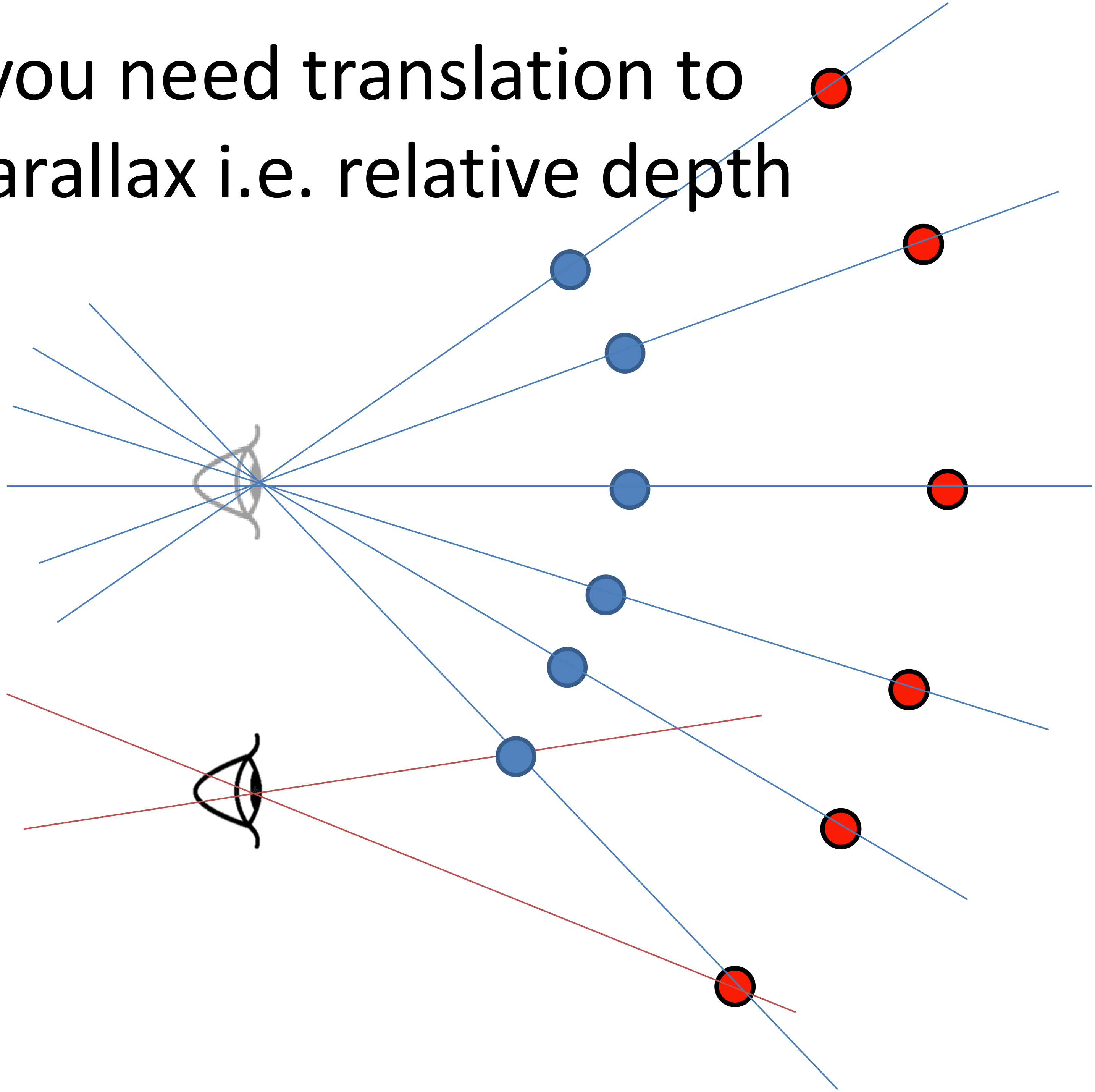
= *Change in position from different view point*

Two eyes give you parallax, you can also move to see more parallax = "Motion Parallax"

Why you need translation to see parallax i.e. relative depth



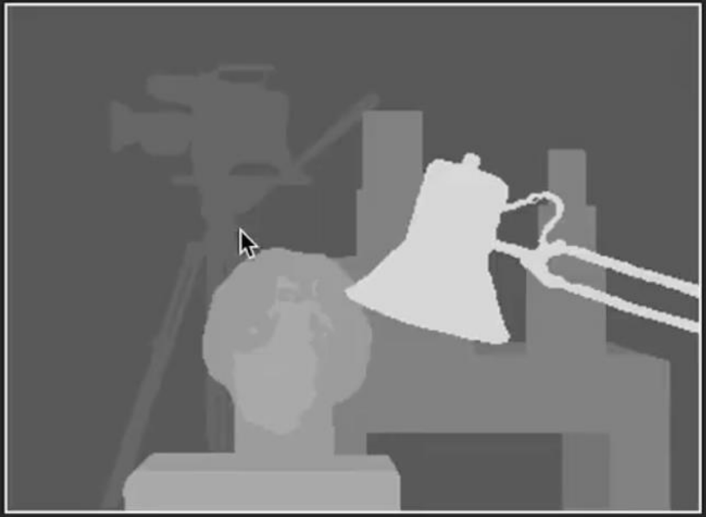
Why you need translation to see parallax i.e. relative depth



Stereo Matching: Finding Disparities

Goal: Find the disparity between left and right stereo pairs.

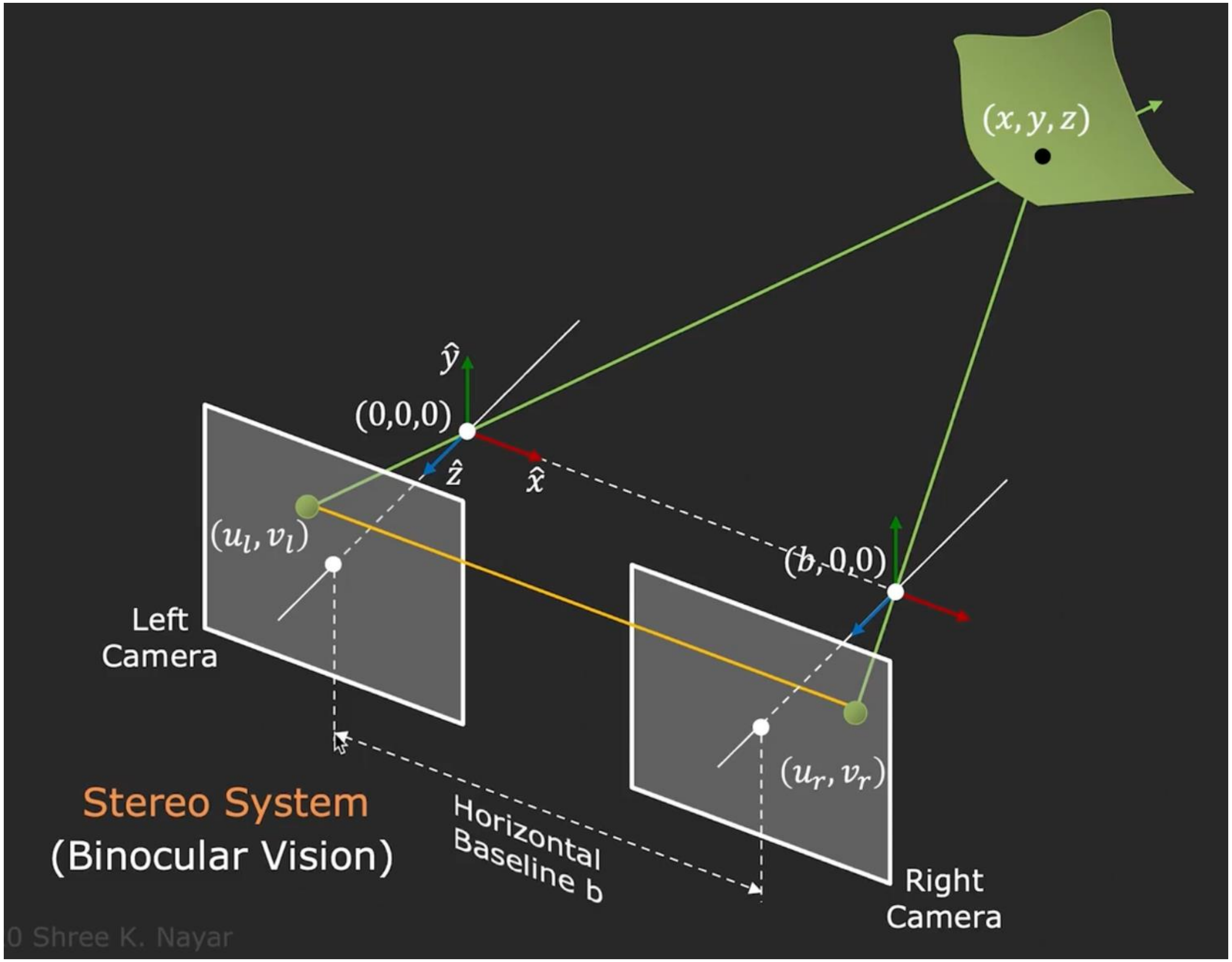
Left/Right Camera Images



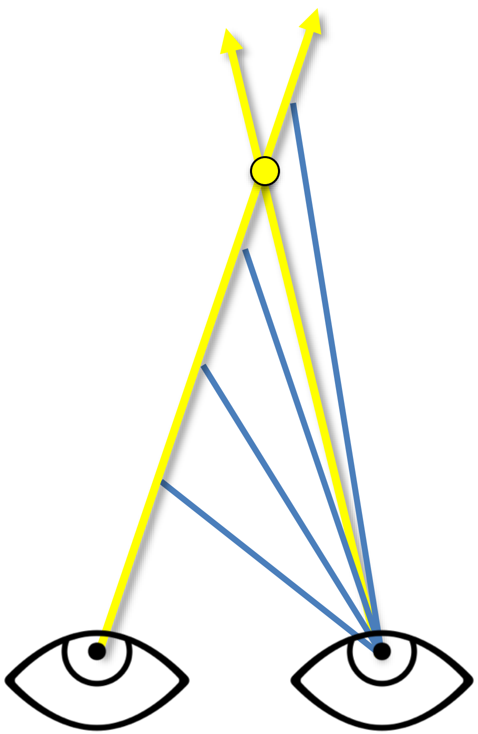
Disparity Map (Ground Truth)

Where is the corresponding point going to be?

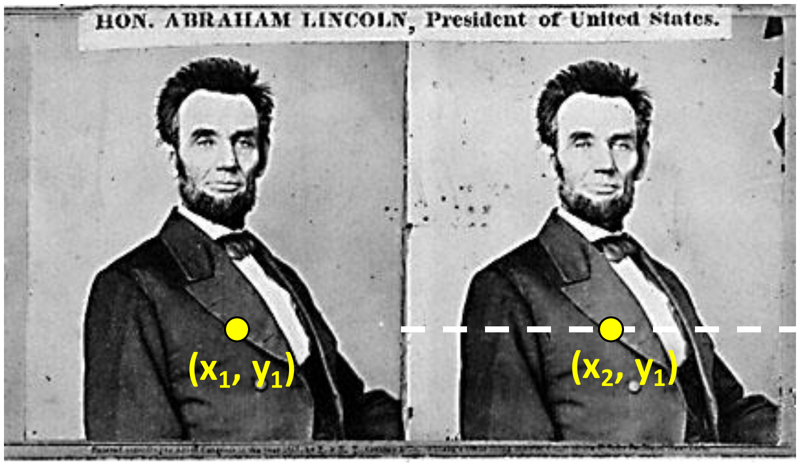
Hint



Epipolar Line



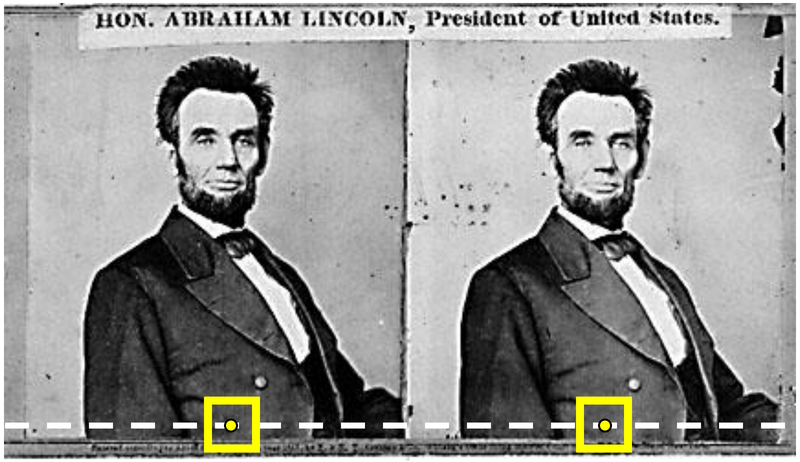
epipolar lines



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

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

Your basic stereo algorithm



For every epipolar line:

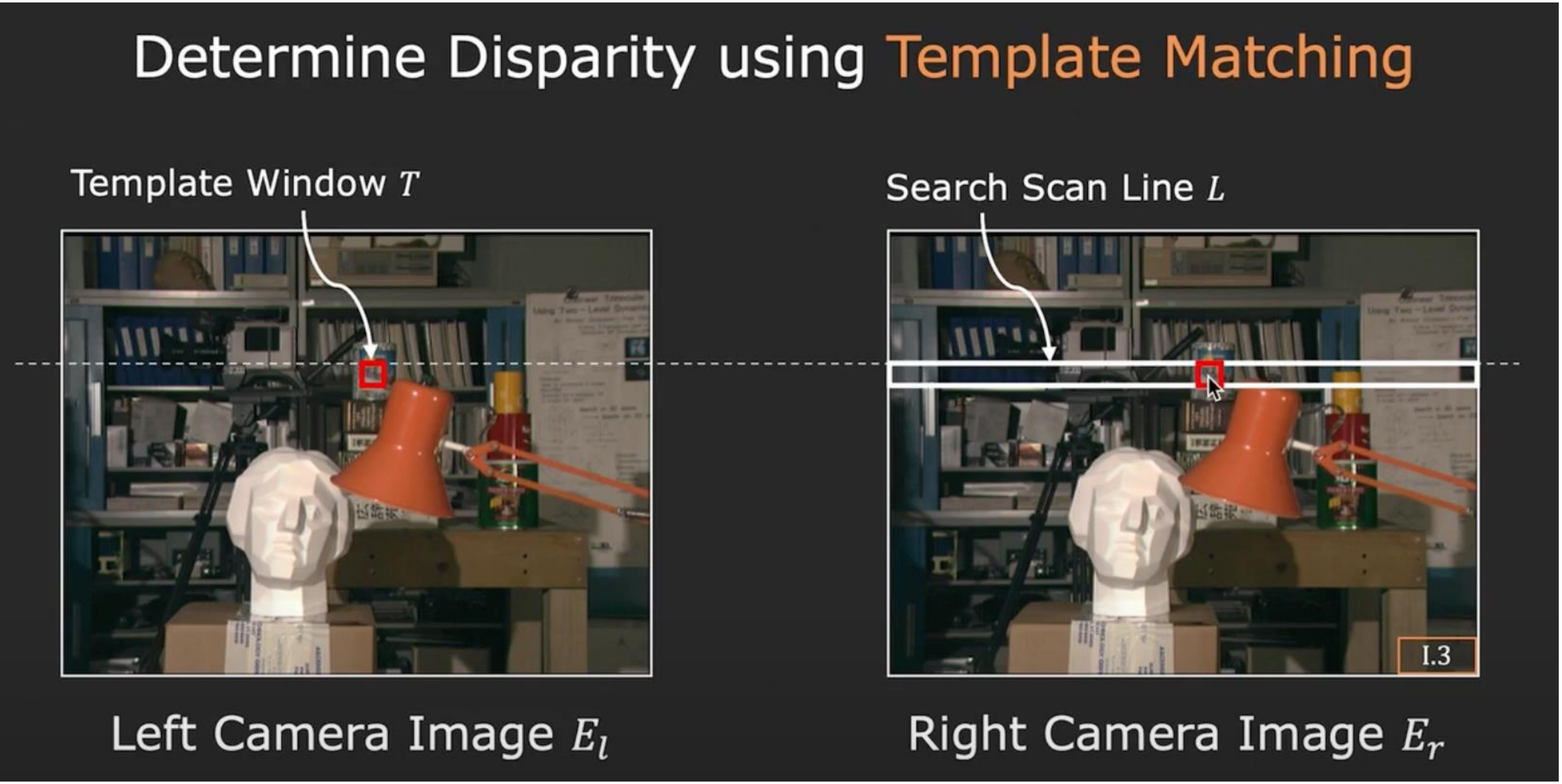
For each pixel in the left image

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

Improvement: match ***windows,*** + *clearly lots of matching strategies*

Slide credit: Noah Snavely

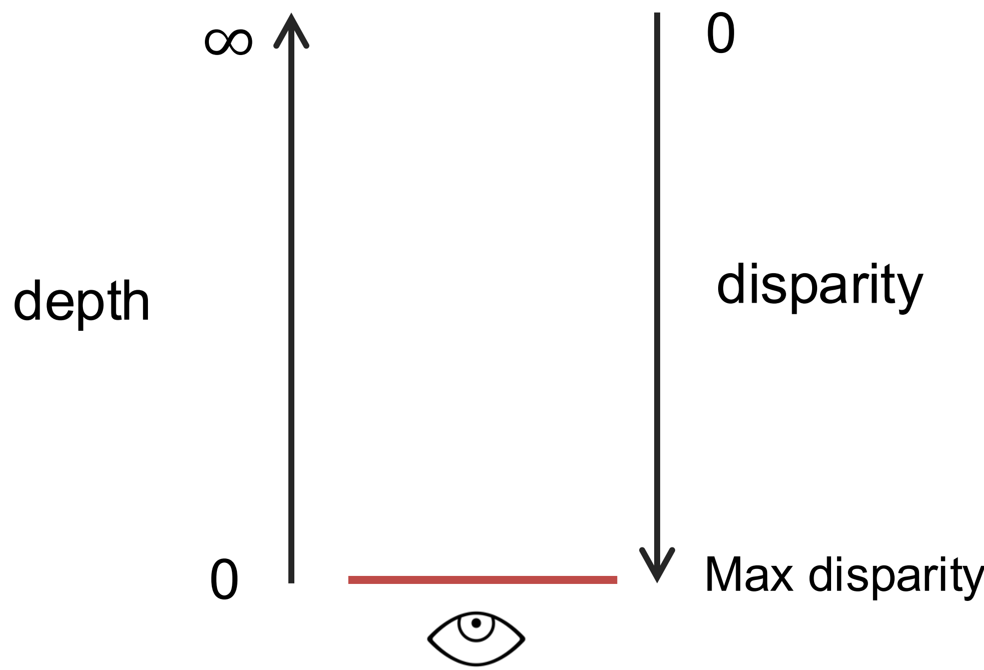
Your basic stereo algorithm



In simple stereo, corresp = 1/depth

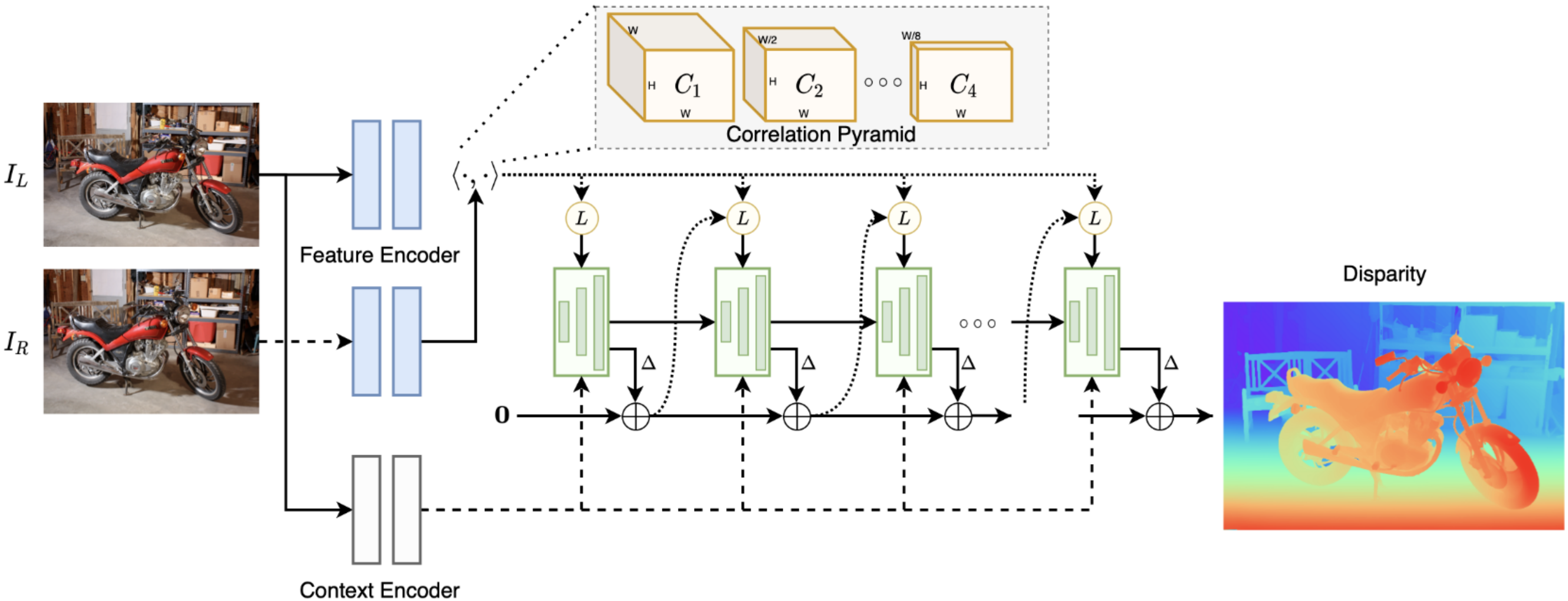
- Once you have correspondence, you know the disparity, so you also have the depth

$$z = \frac{fB}{\underbrace{u_l - u_r}_{\text{disparity}}}$$



A modern learning example: RaftStereo (3DV '21)

- Use network to solve the correspondence along epipolar line



A modern learning example: RaftStereo

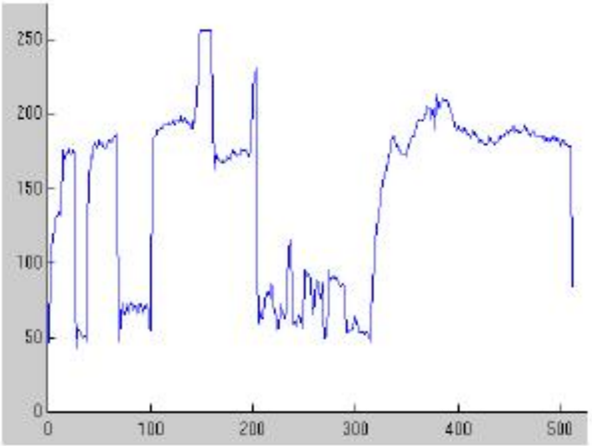
- Just solving correspondence gives you nice – per-pixel depth for an image

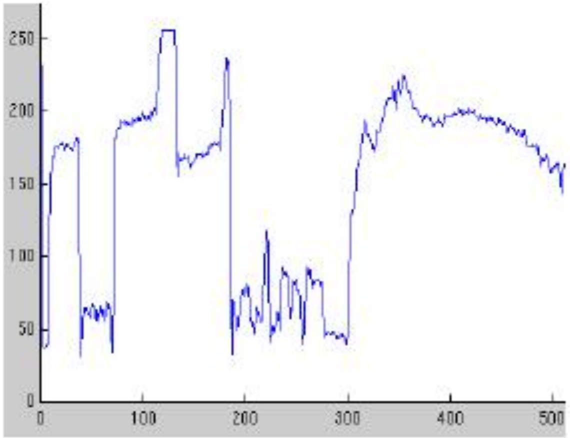

How to solve the Correspondence problem

Parallel camera example – epipolar lines are corresponding rasters

Source: Andrew Zisserman

Intensity profiles

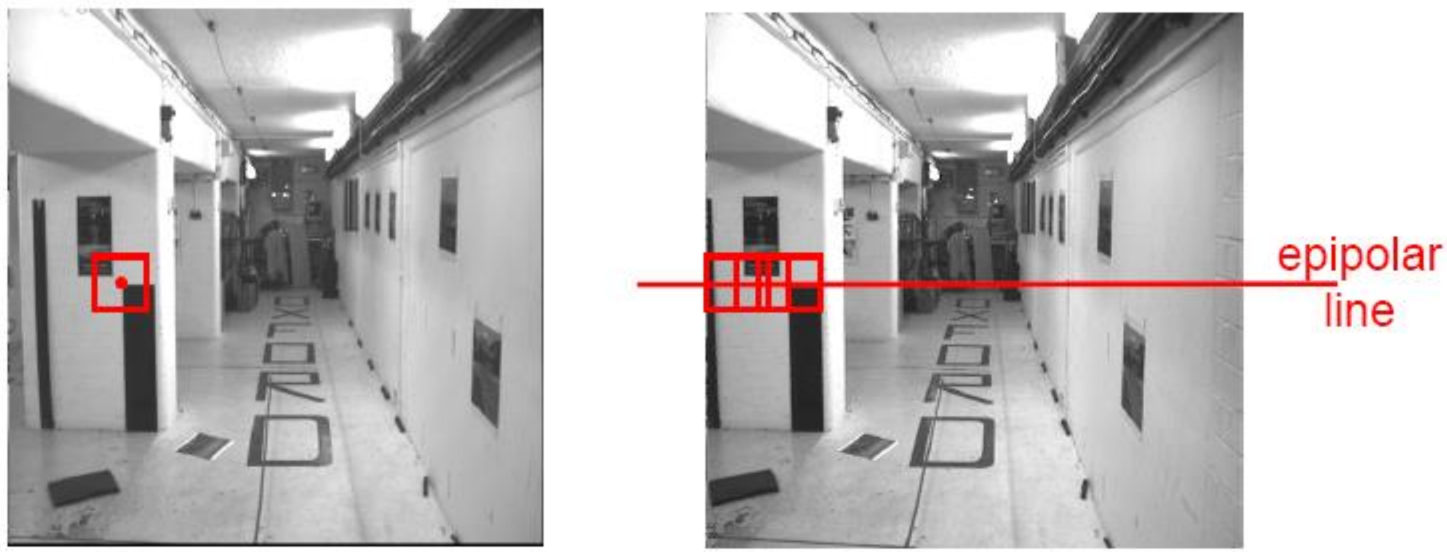




- Clear correspondence between intensities, but also noise and ambiguity

Source: Andrew Zisserman

Correspondence problem



Neighborhood of corresponding points are similar in intensity patterns.

Source: Andrew Zisserman

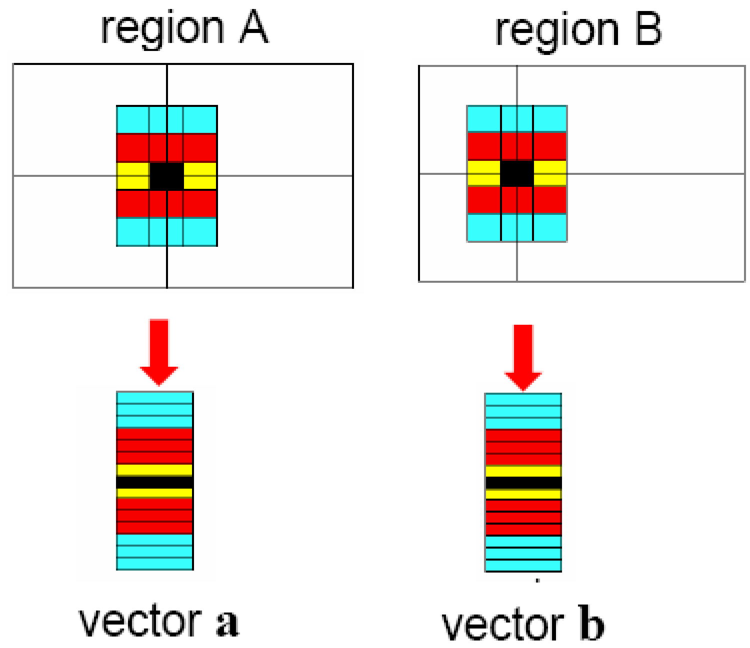
Normalized cross correlation

subtract mean: $A \leftarrow A - <A>, B \leftarrow B - <B>$

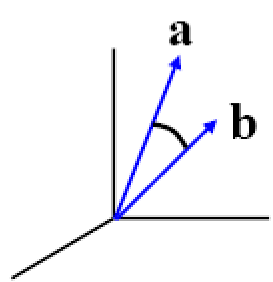
$$\mathrm{NCC} = \frac{\sum_i \sum_j A(i,j)B(i,j)}{\sqrt{\sum_i \sum_j A(i,j)^2}\sqrt{\sum_i \sum_j B(i,j)^2}}$$

Write regions as vectors

$$\mathtt{A} \rightarrow \mathbf{a}, \;\; \mathtt{B} \rightarrow \mathbf{b}$$



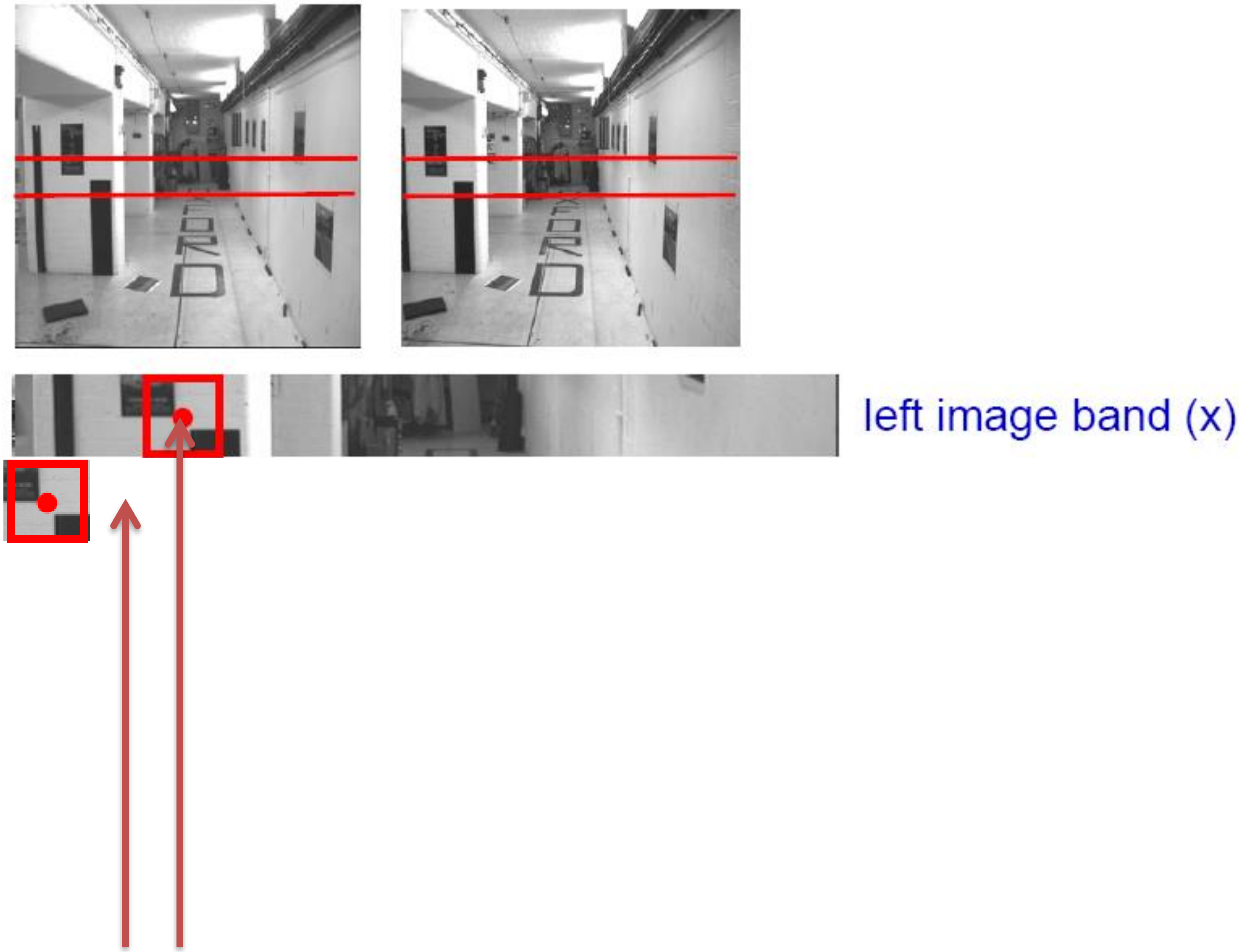
$$\mathrm{NCC} = \frac{\mathbf{a}.\mathbf{b}}{|\mathbf{a}||\mathbf{b}|}$$



$$-1 \leq \mathrm{NCC} \leq 1$$

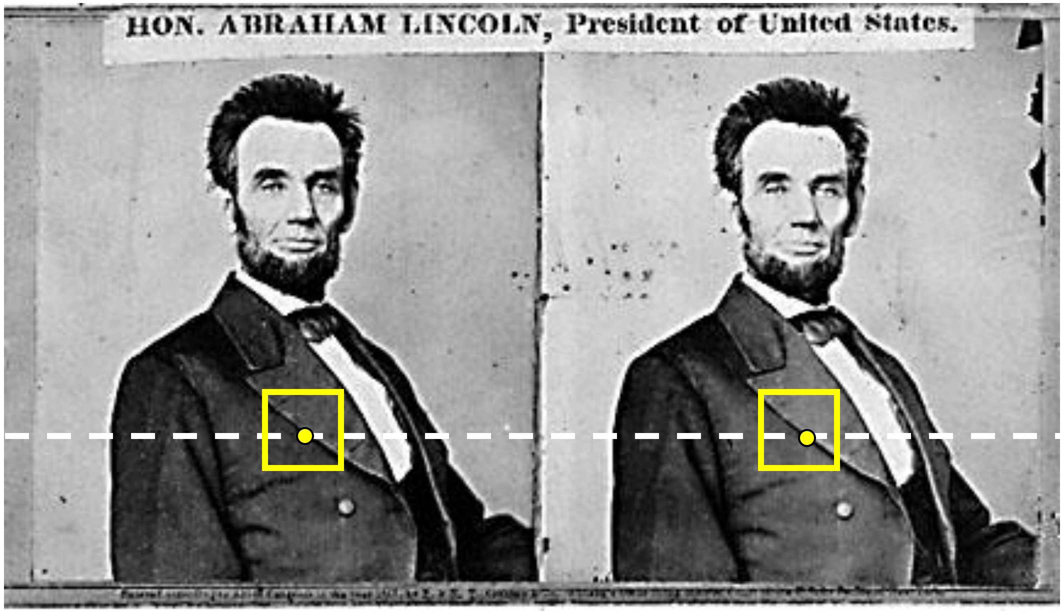
Source: Andrew Zisserman

Correlation-based window matching



Source: Andrew Zisserman

Dense correspondence search



For each epipolar line

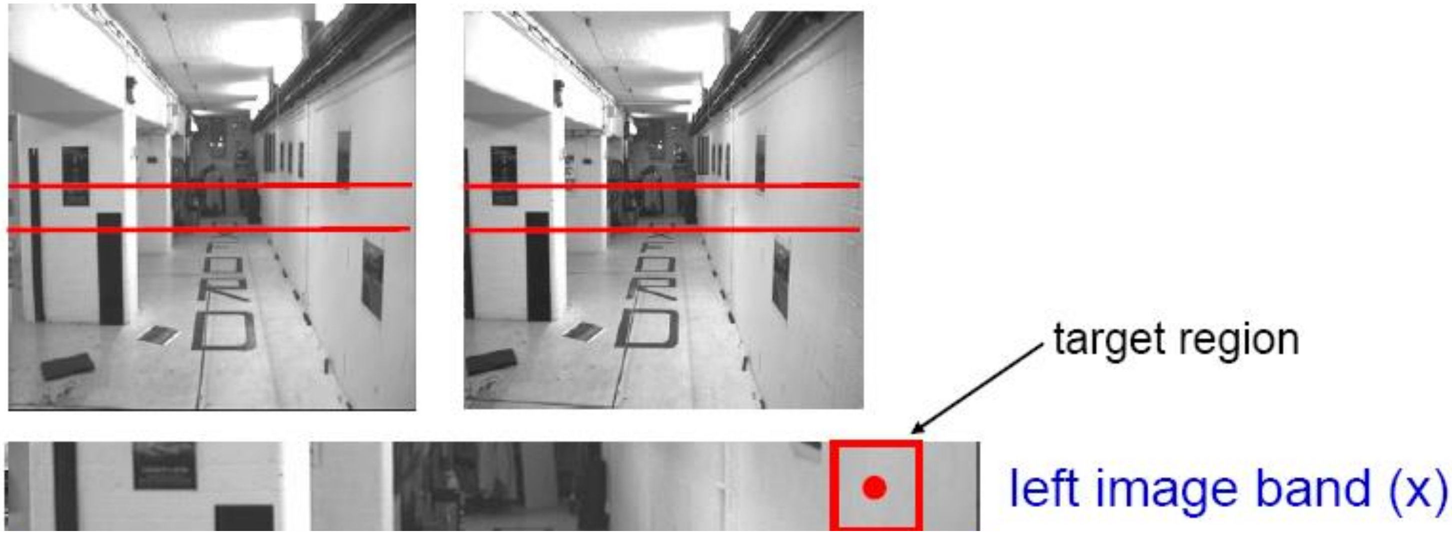
For each pixel / window in the left image

- compare with every pixel / window on same epipolar line in right image
- pick position with minimum match cost (e.g., SSD, correlation)

Adapted from Li Zhang

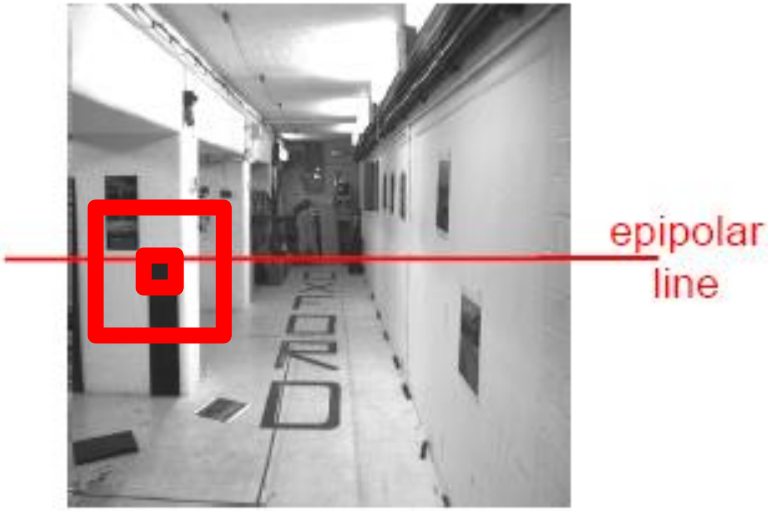
Grauman

Textureless regions



Source: Andrew Zisserman

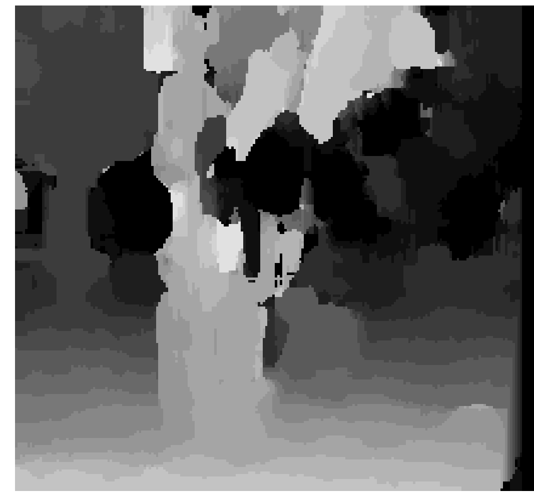
Effect of window size



Source: Andrew Zisserman

Effect of window size

W = 3



W = 20

Want window large enough to have sufficient intensity variation, yet small enough to contain only pixels with about the same disparity.

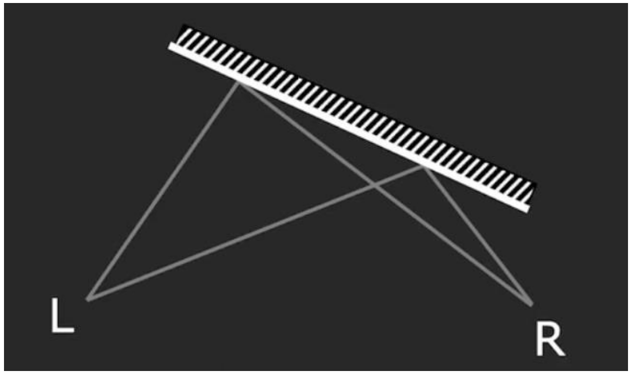
Figures from Li Zhang

Grauman

Issues with Stereo

- Surface must have non-repetitive texture

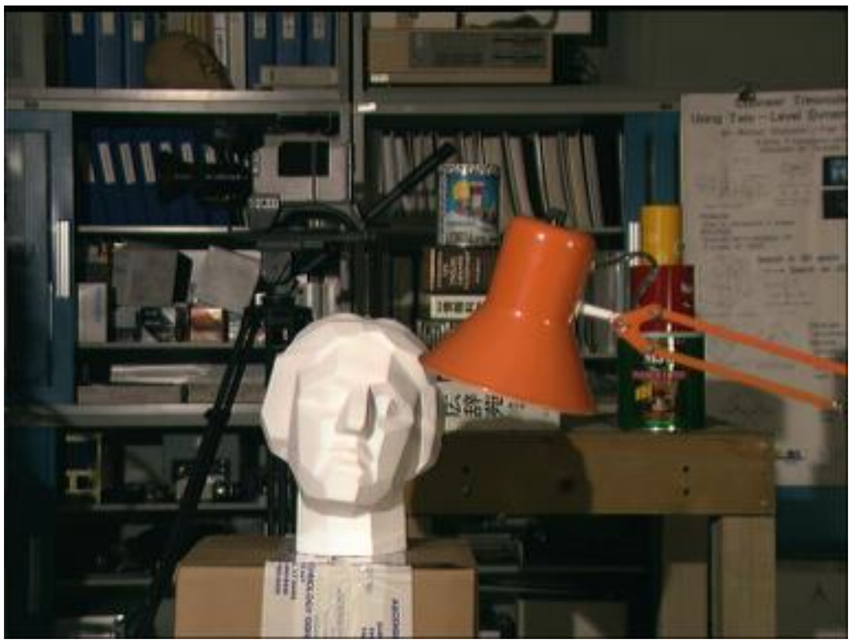
- Foreshortening effect makes matching a challenge



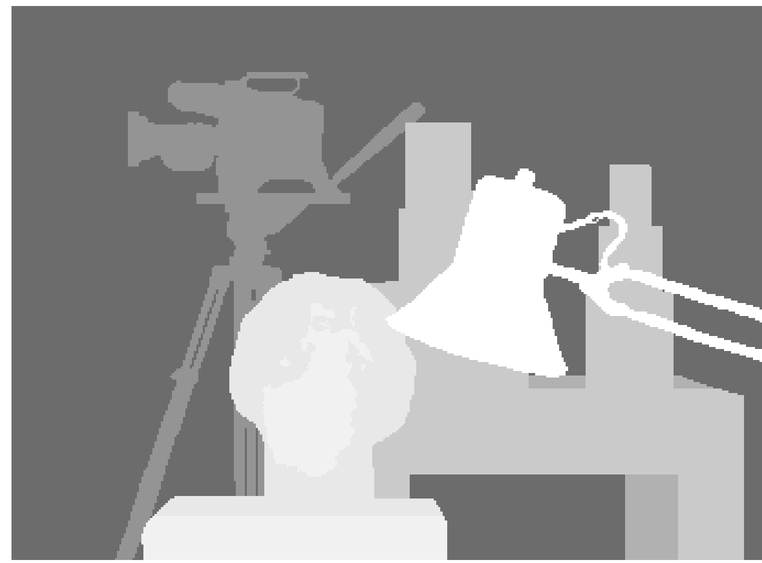
Slide Credit: Shree Nayar

Stereo Results

– Data from University of Tsukuba

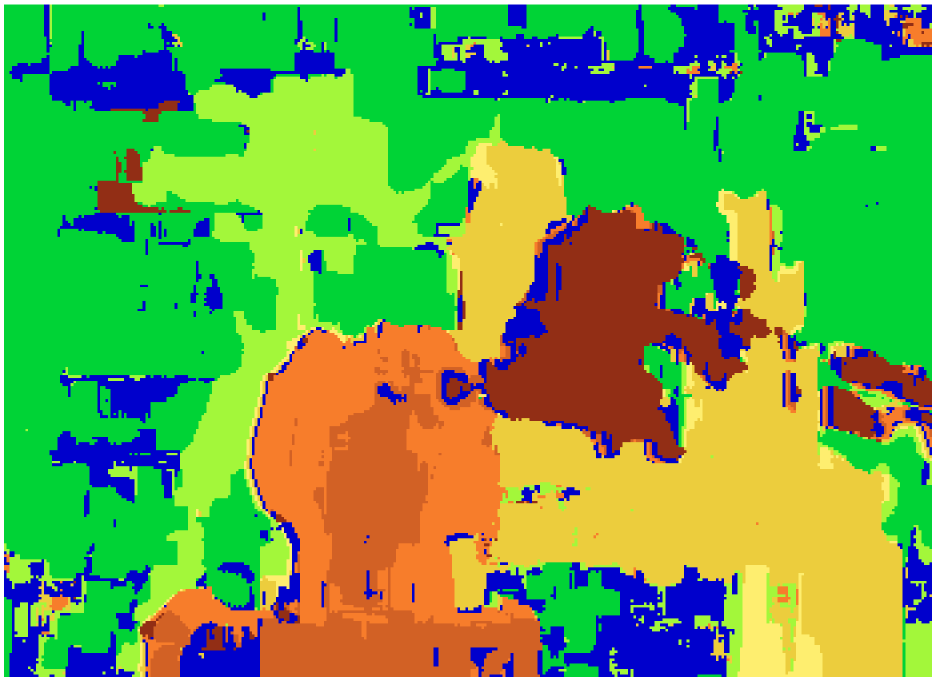


Scene



Ground truth

Results with Window Search



Window-based matching
(best window size)

Ground truth

Better methods exist...

Energy Minimization

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

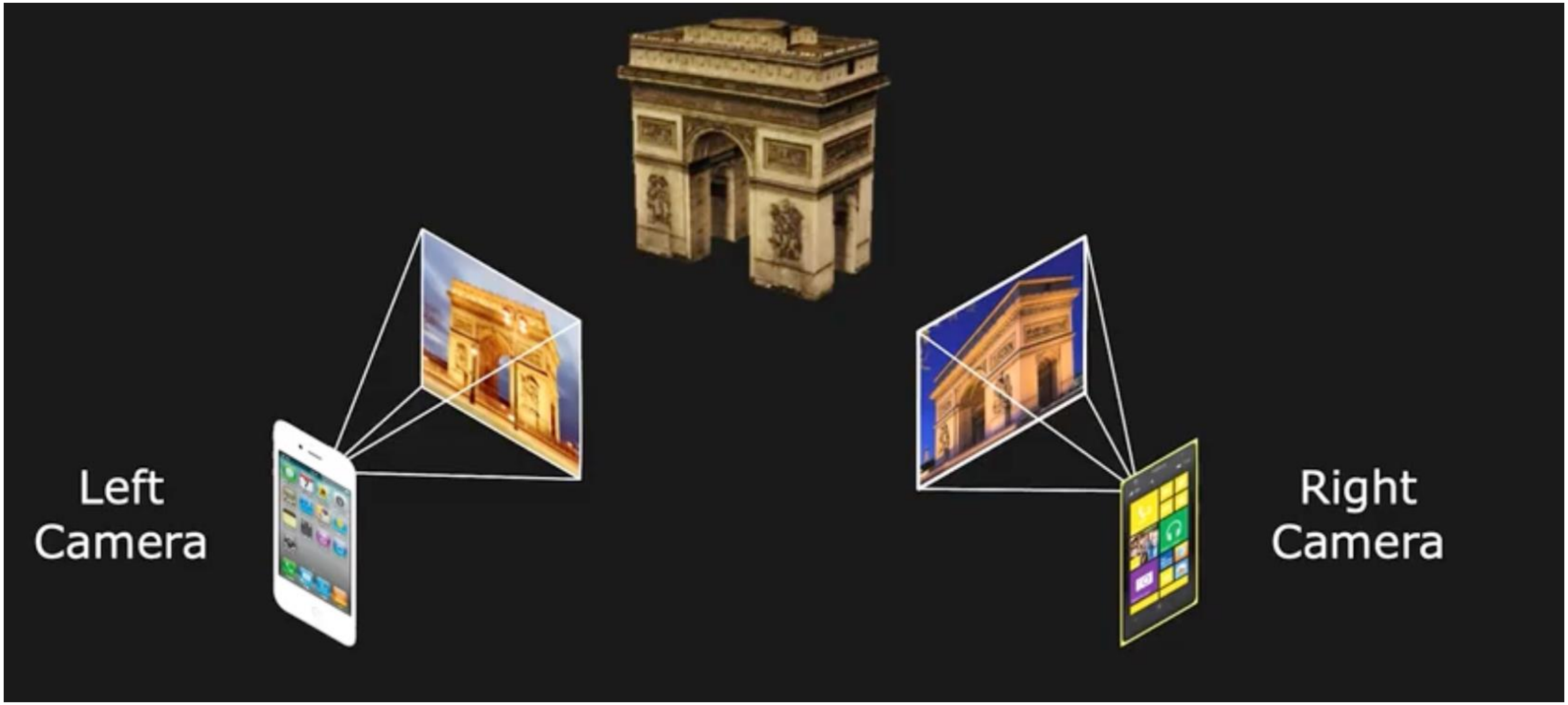
Ground truth

Summary

- With a simple stereo system, **how much pixels move, or “disparity”** give information about the depth
- Correspondences to measure the pixel disparity

Next: Uncalibrated Stereo

- From two arbitrary views

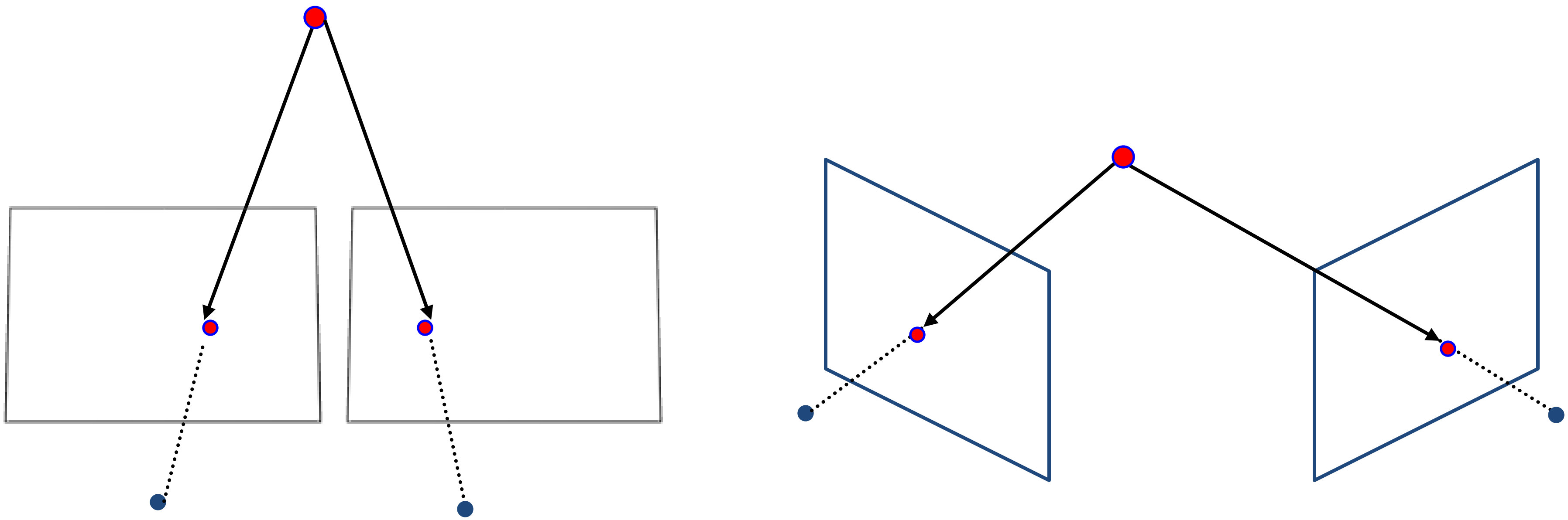


- Assume intrinsics are known (fx, fy, ox, oy)

Slide Credit: Shree Nayar

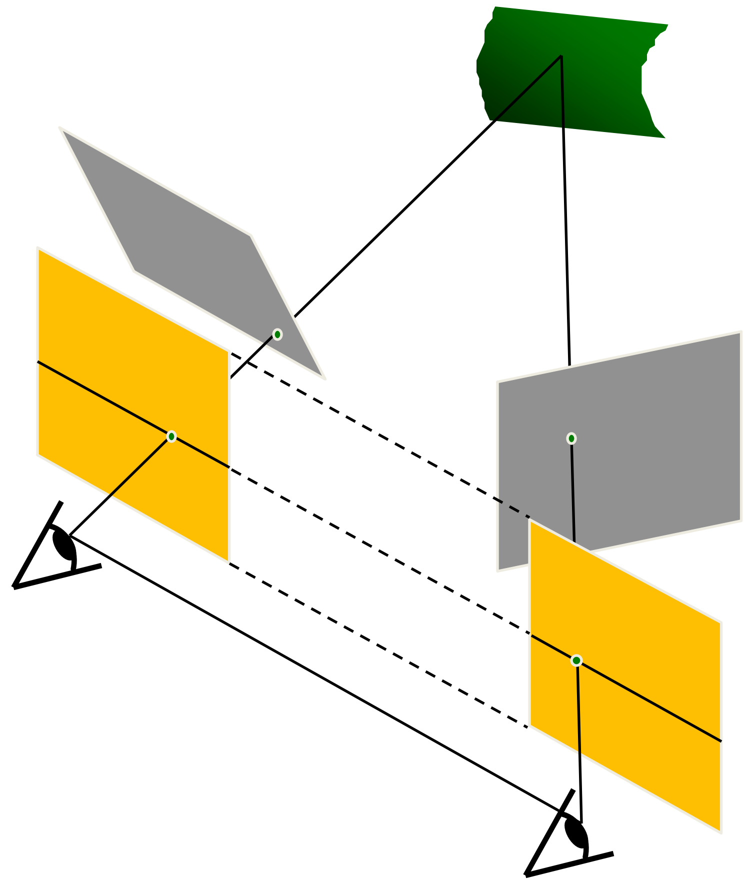
General case, with calibrated cameras

- The two cameras need not have parallel optical axes.



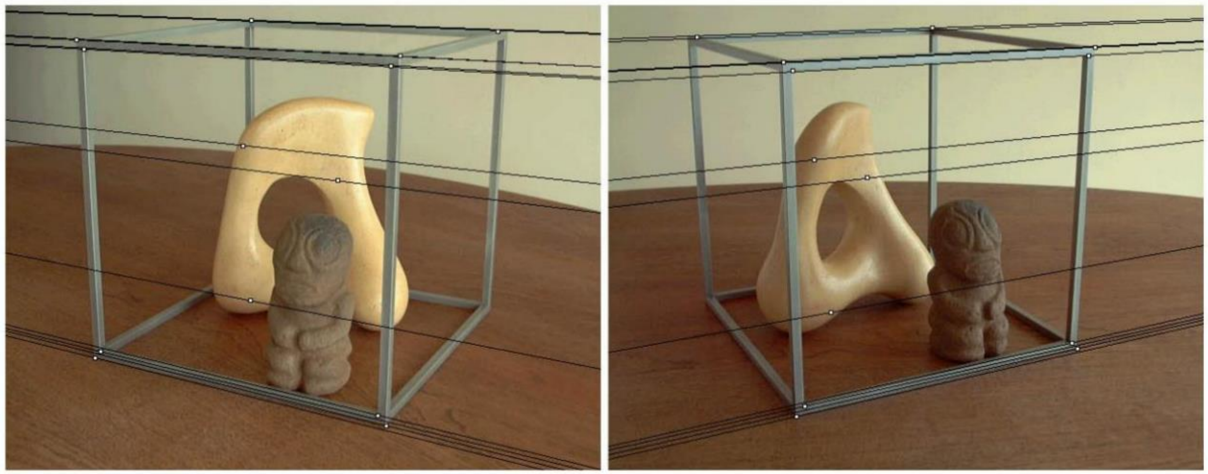
Option 1: Rectify via homography

- reproject image planes onto a common plane
  - plane parallel to the line between optical centers
- pixel motion is horizontal after this transformation
- two homographies, one for each input image reprojection
  - C. Loop and Z. Zhang. [Computing Rectifying Homographies for Stereo Vision](). CVPR 1999.



Slide courtesy of Noah Snavely

Option 1: Rectify via homography



Original stereo pair

After rectification

Slide courtesy of Noah Snavely

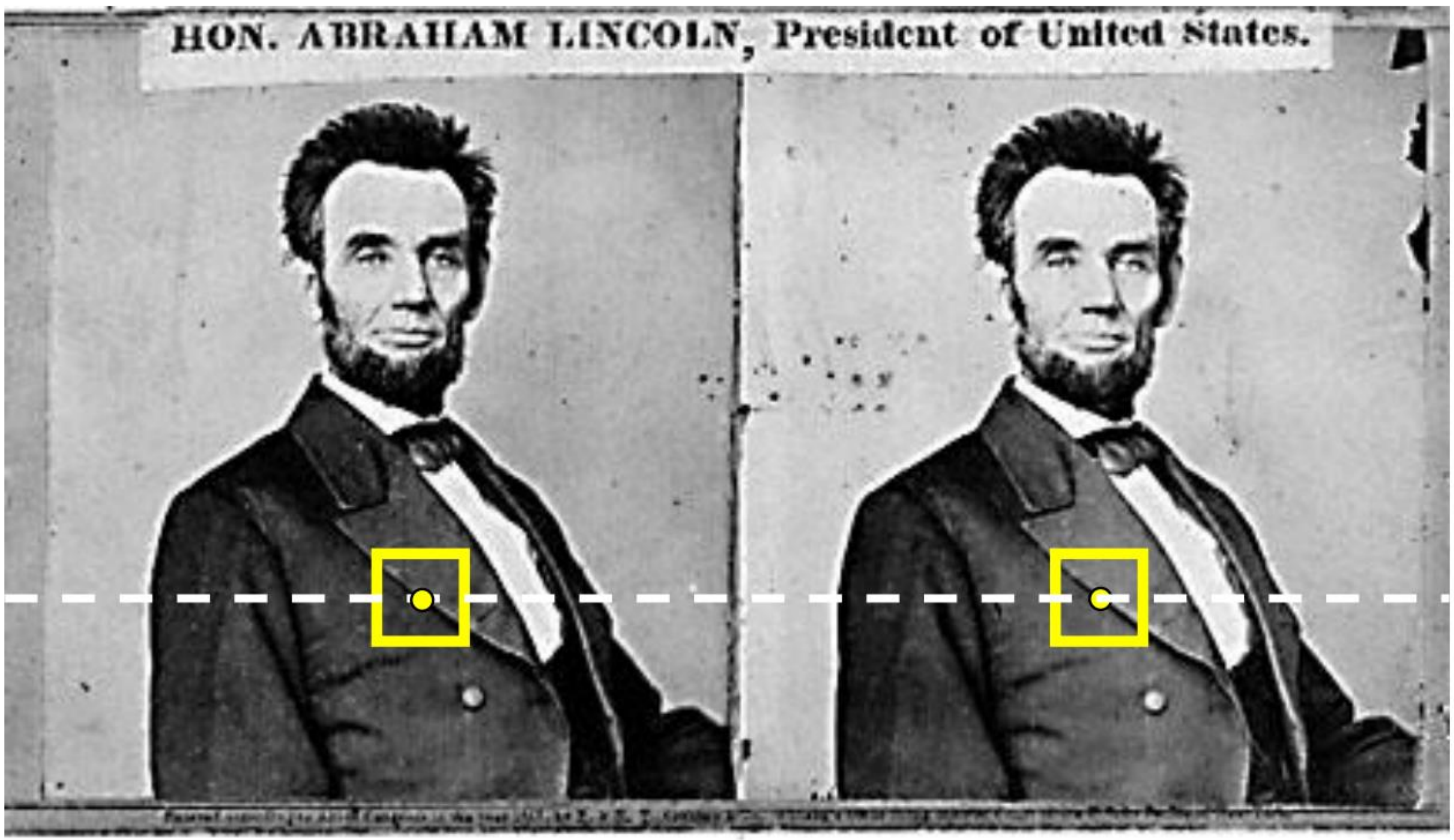
Option 2

1. Solve for correspondences
2. Estimate camera
3. Triangulate

Option 2

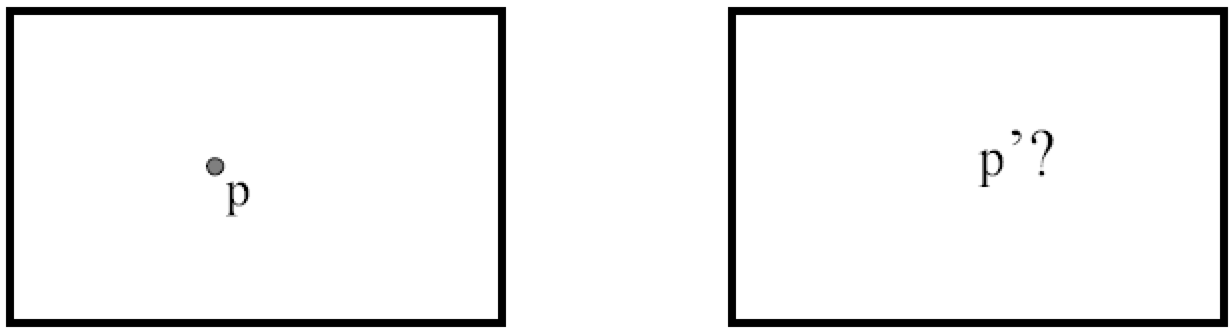
1. **Solve for correspondences**
2. **Estimate camera**
   - **What is the relationship between the camera + correspondences?**
3. Triangulate

Where do epipolar lines come from?



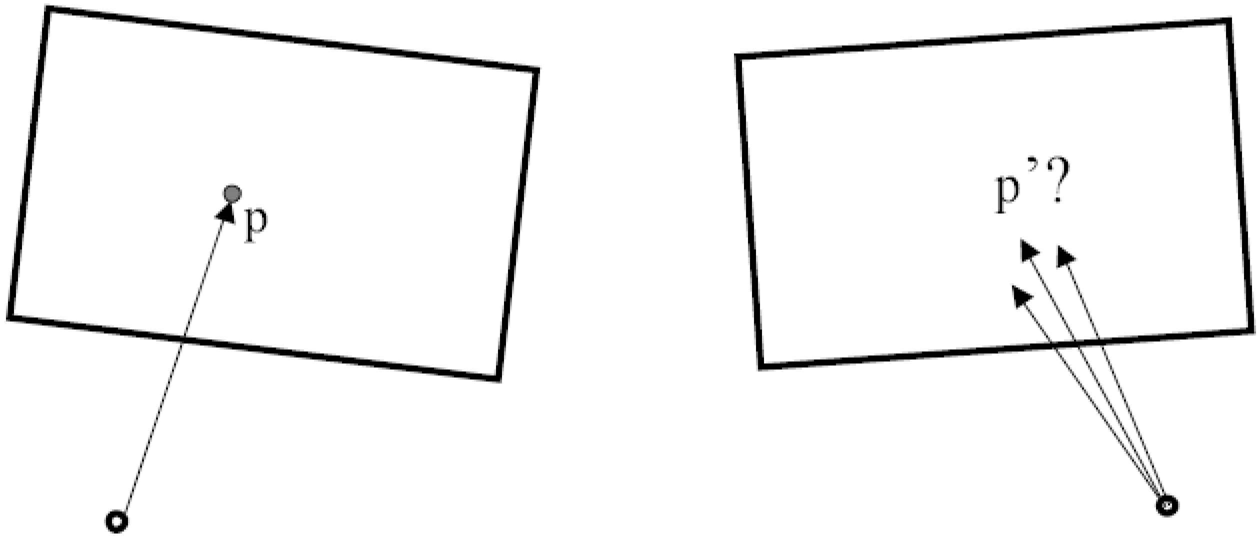
Slide courtesy of Noah Snavely

Stereo correspondence constraints



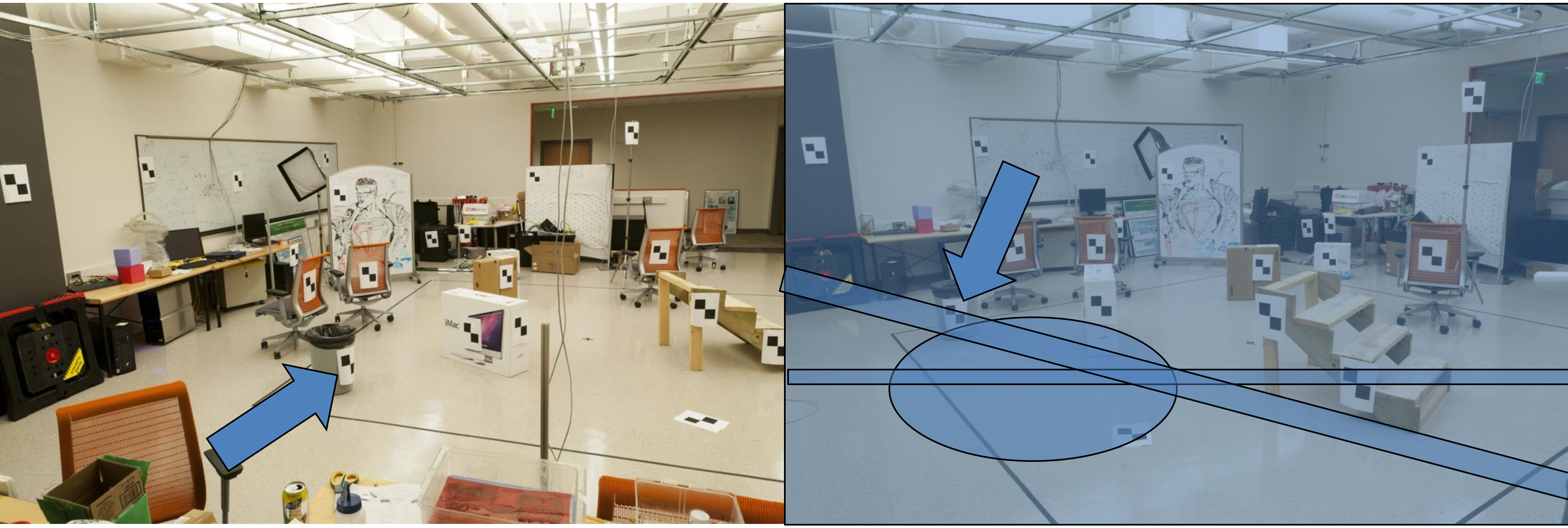
- Given p in left image, where can corresponding point p' be?

Stereo correspondence constraints



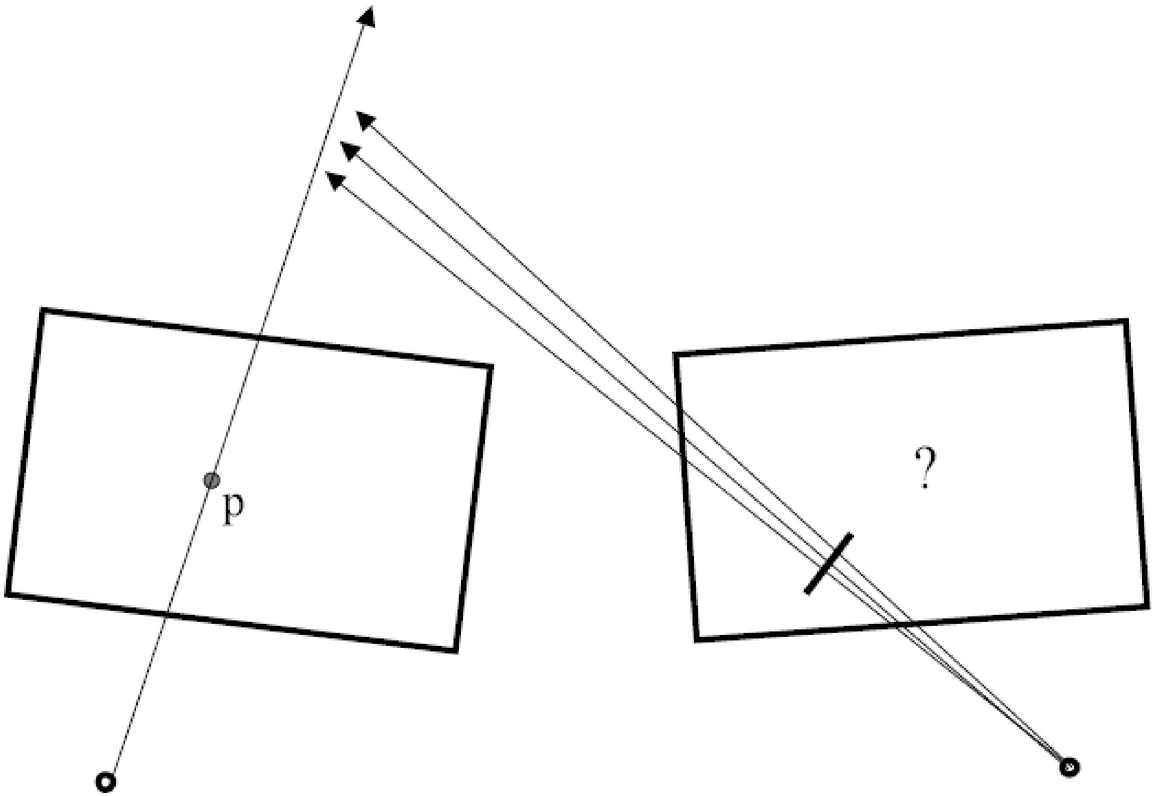
- Given p in left image, where can corresponding point p' be?

Where do we need to search?

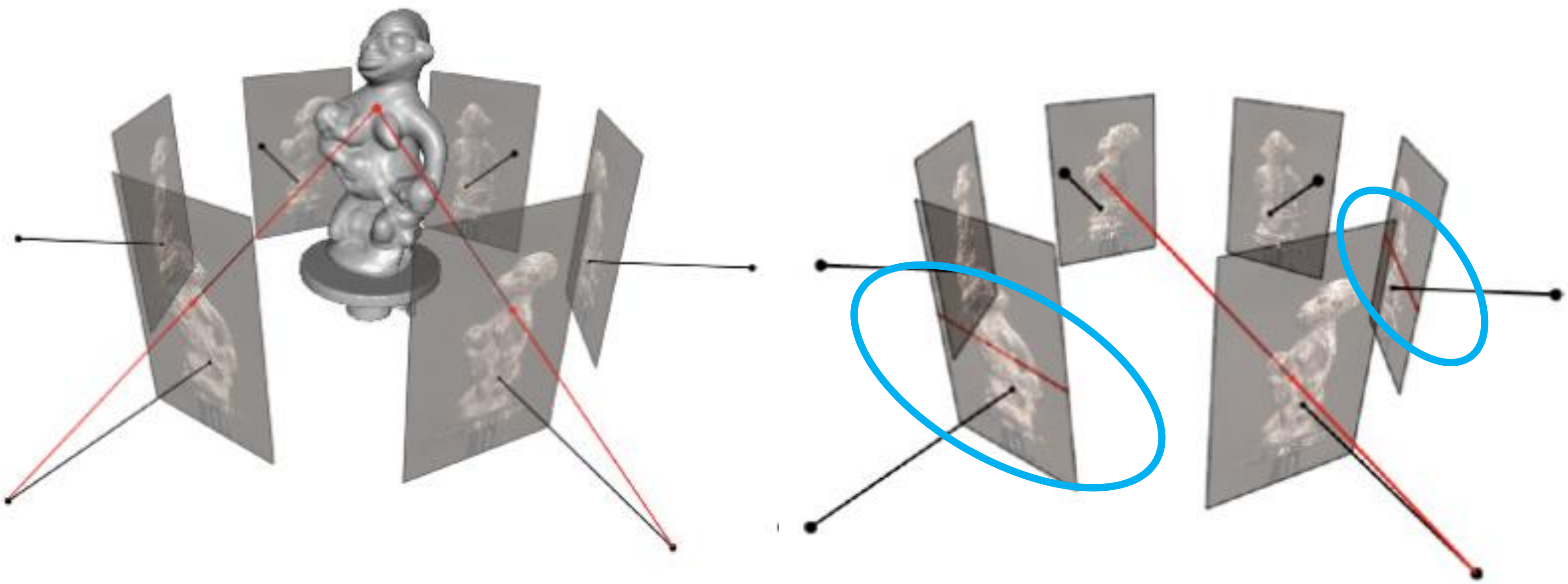


Slide by James Hays

Stereo correspondence constraints



Epipolar Geometry



Figures by Carlos Hernandez