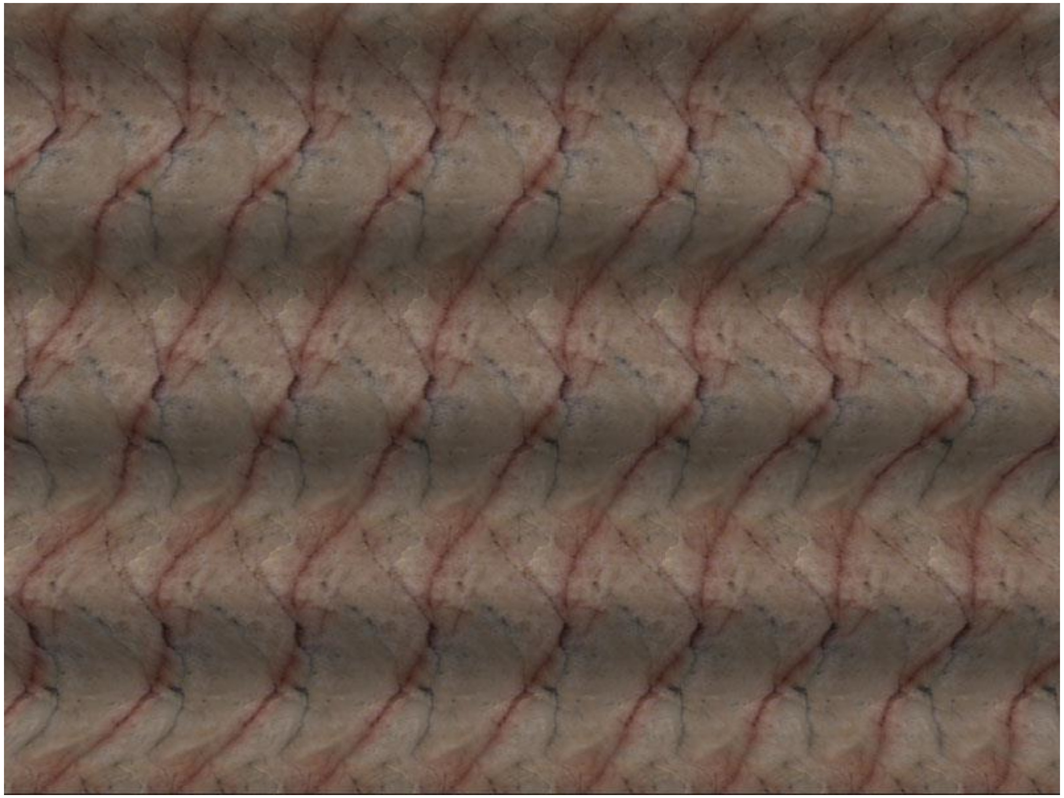
CS5670: Computer Vision

Noah Snavely, Zhengqi Li

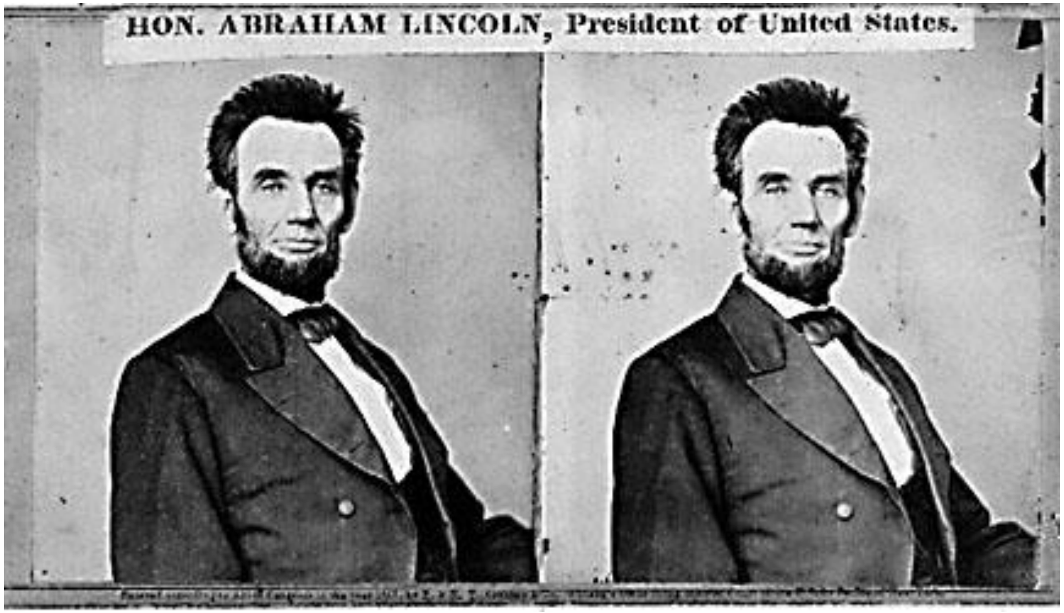
Stereo



Single image stereogram, by Niklas Een

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

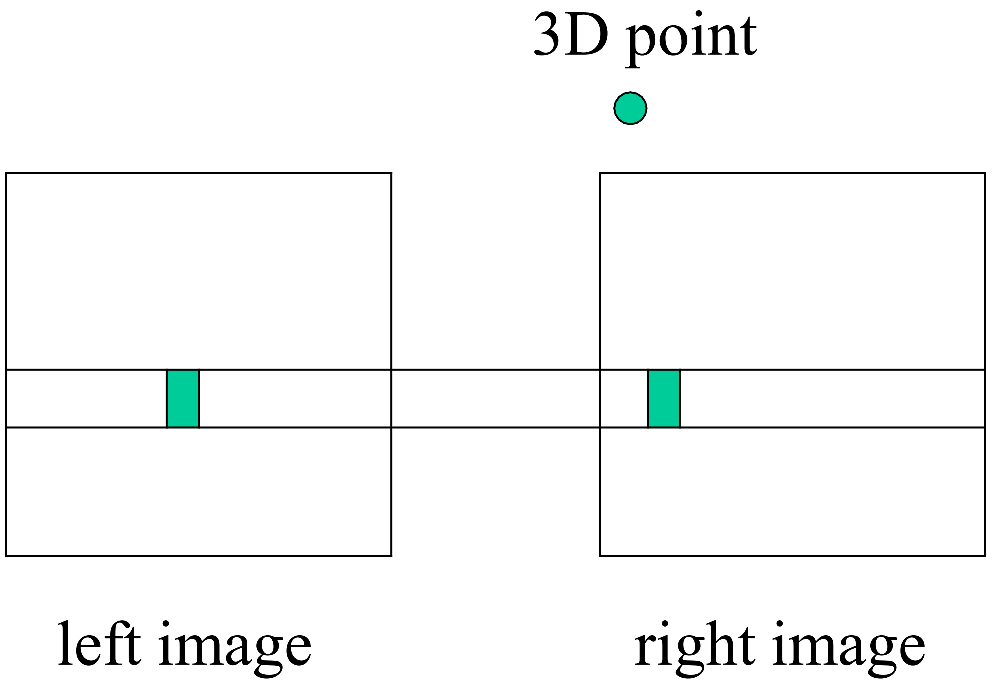
Stereo



- 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

How do we get 3D from Stereo Images?

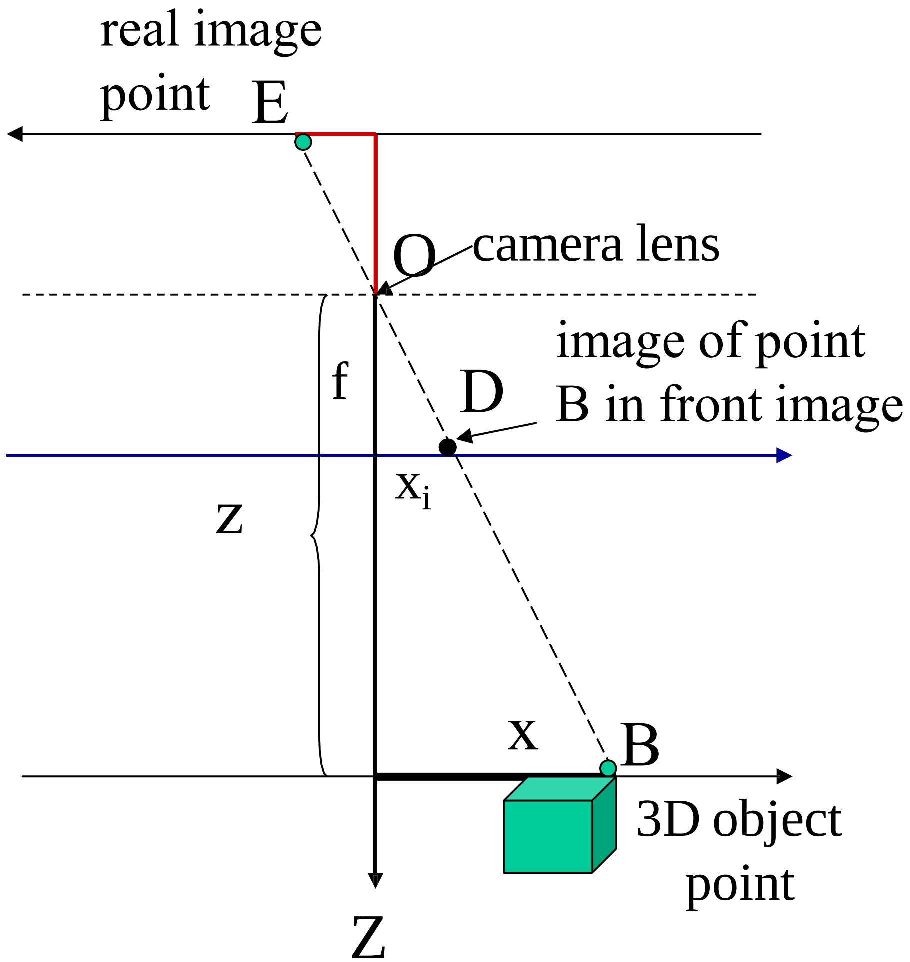
Perception of depth arises from “disparity” of a given 3D point in your right and left retinal images’



disparity: the difference in image location of the *same 3D point* when projected under perspective to two different cameras

$$\boxed{\mathbf{d = xleft\ \ -xright}}$$

Recall :Perspective Projection



This is the axis of the real image plane.

O is the center of projection.

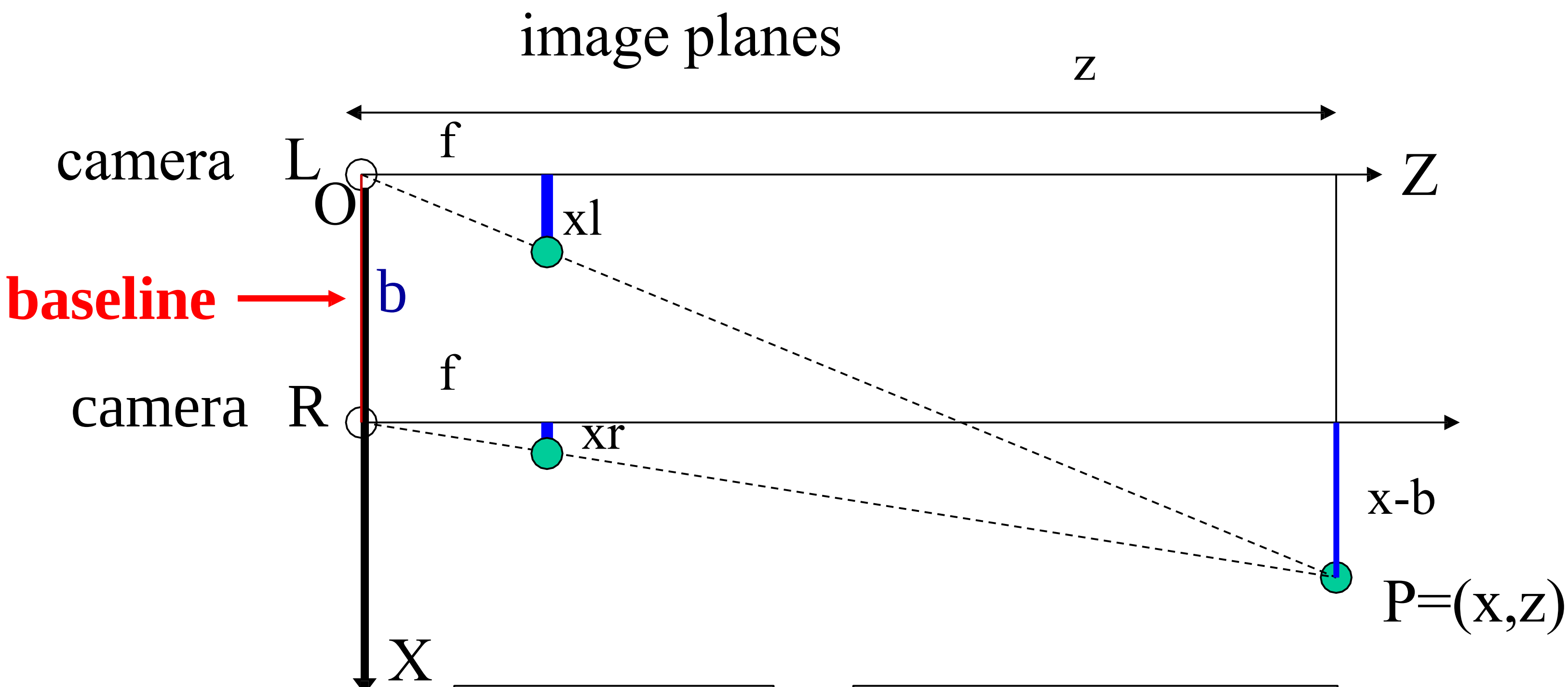
This is the axis of the front image plane, which we use.

$$\boxed{\frac{x_i}{f} = \frac{x}{z}}$$

(from similar triangles)

Projection for Stereo Images

Simple Model: Optic axes of 2 cameras are parallel



$$\frac{z}{f} = \frac{x}{xl} \qquad \frac{z}{f} = \frac{x-b}{xr} \qquad \frac{z}{f} = \frac{y}{yl} = \frac{y}{yr}$$

Y-axis is perpendicular to the page.

(from similar triangles)

3D from Stereo Images: Triangulation

- For stereo cameras with parallel optical axes, focal length f, baseline b, corresponding image points (xl,yl) and (xr,yr), the location of the 3D point can be derived from previous slide's equations:

Depth z = f*b / (xl - xr) = f*b/d

x = xl*z/f or b + xr*z/f

y = yl*z/f or yr*z/f

This method of determining depth from disparity d is called **triangulation.**

Note that **depth is inversely proportional to disparity**

$$\textbf{Depth } z = f*b / (xl - xr) = f*b/d$$

$$x = xl*z/f \quad \text{or} \quad b + xr*z/f$$

$$y = yl*z/f \quad \text{or} \quad yr*z/f$$

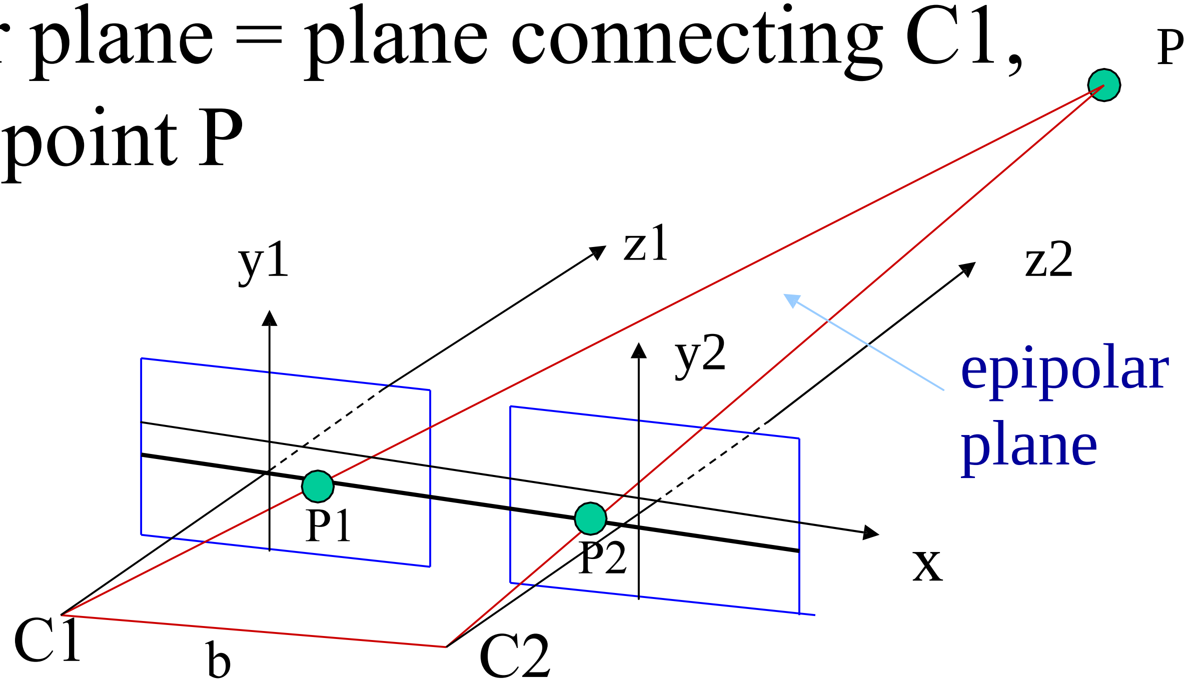
Two main problems:

1. Need to know focal length f, baseline b
   - use prior knowledge or camera calibration (http://www.vision.caltech.edu/bouguetj/calib_doc)
2. Need to find corresponding point (xr,yr) for each (xl,yl) $\Rightarrow$ Correspondence problem

Q: Given a point in the left image, do you need to search the entire right image for the corresponding point?

Epipolar Constraint for Correspondence

Epipolar plane = plane connecting C1, C2, and point P

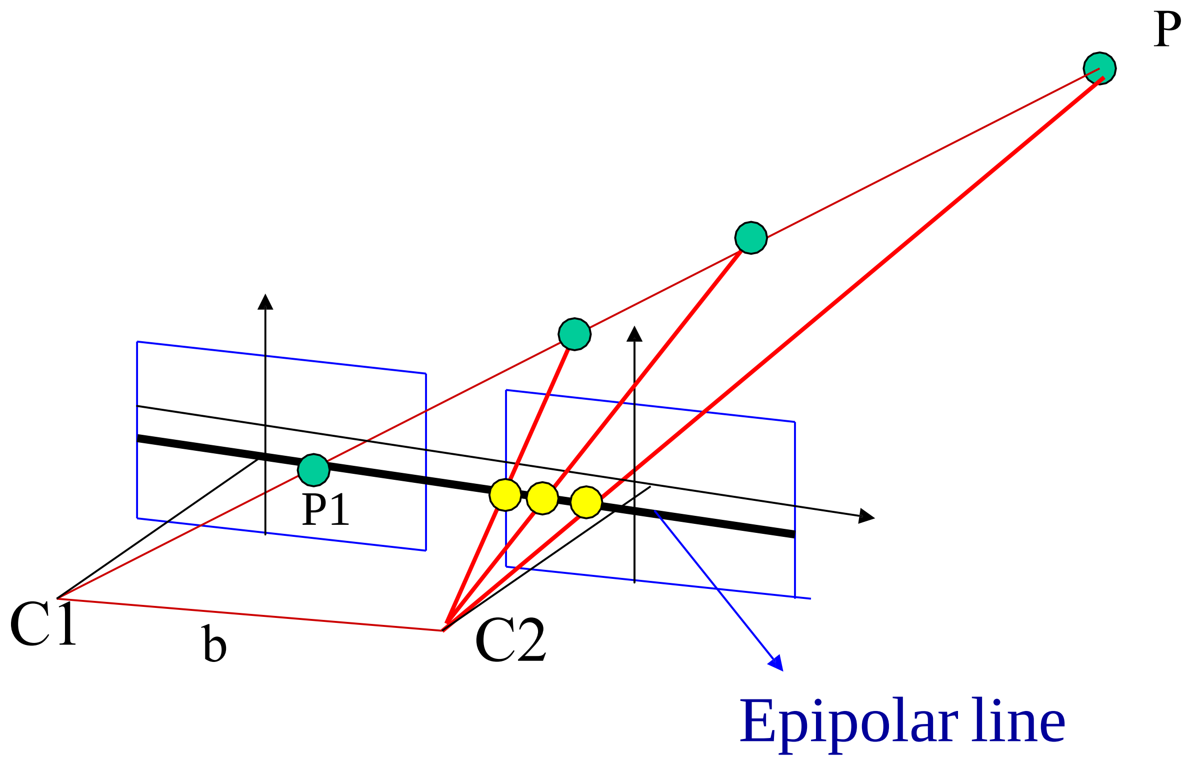


.

***Epipolar plane cuts through image planes forming an epipolar line in each plane**
***Match for P1 (or P2) in the other image must lie on epipolar line**

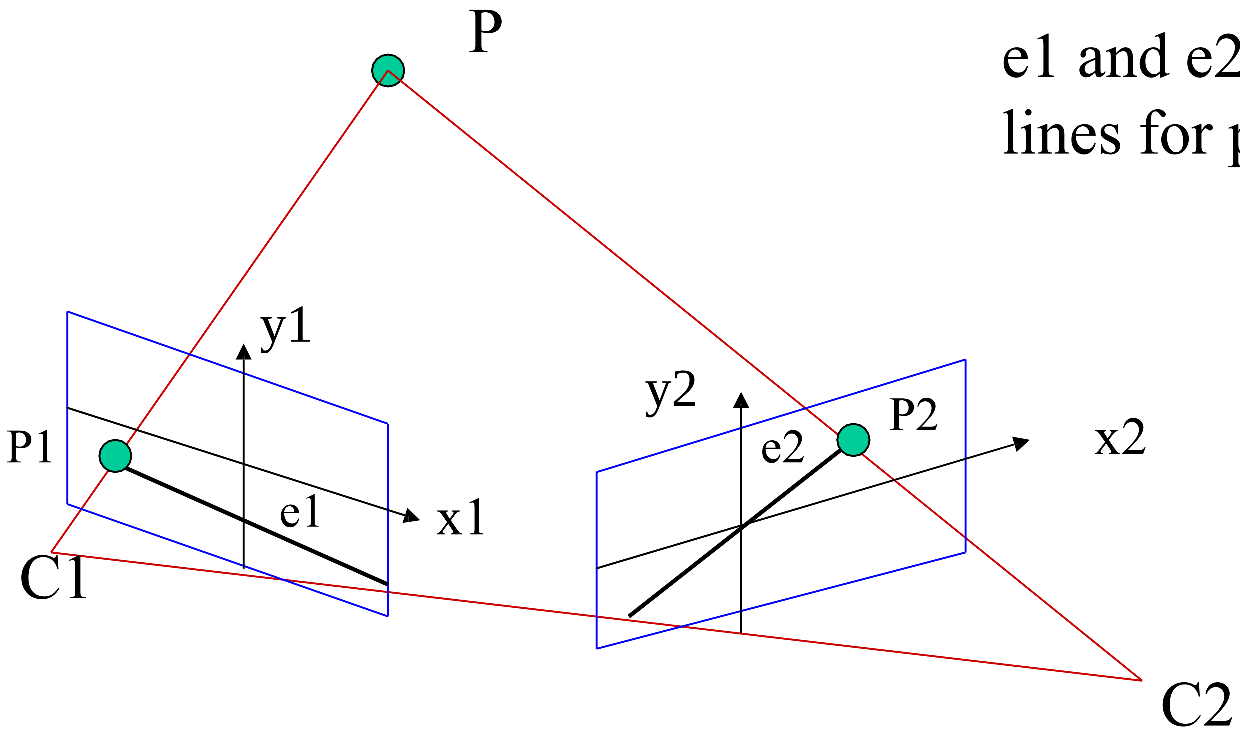
Epipolar Constraint for Correspondence

Match for P1 in the other image must lie on epipolar line
So need search only along this line



What if the optical axes of the 2 cameras are not parallel to each other? Does Epipolar constraint still holds ?

Epipolar constraint still holds…

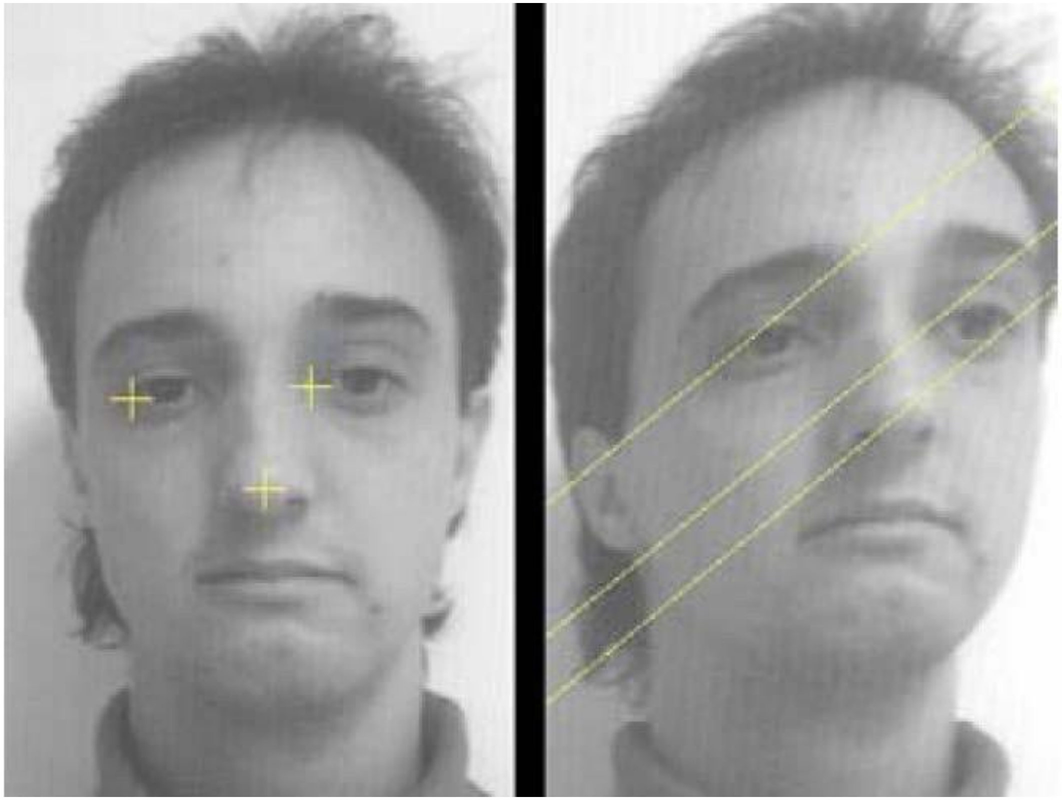


e1 and e2 are the epipolar lines for point P

But the epipolar lines may no longer be horizontal

Java demo: http://www.ai.sri.com/~luong/research/Meta3DViewer/EpipolarGeo.html

Example

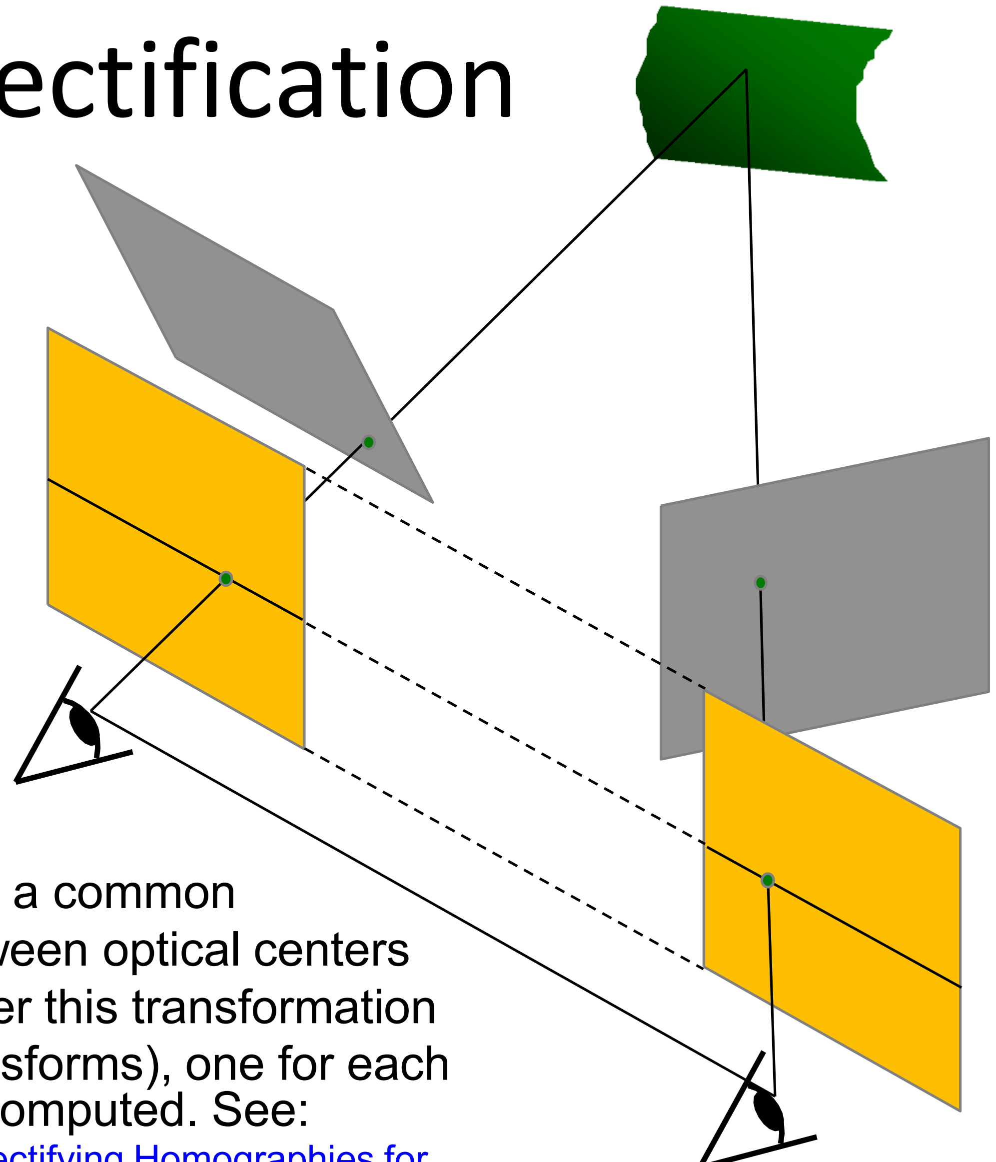


(from a slide by Pascal Fua)

Yellow epipolar lines for the three points shown on the left image

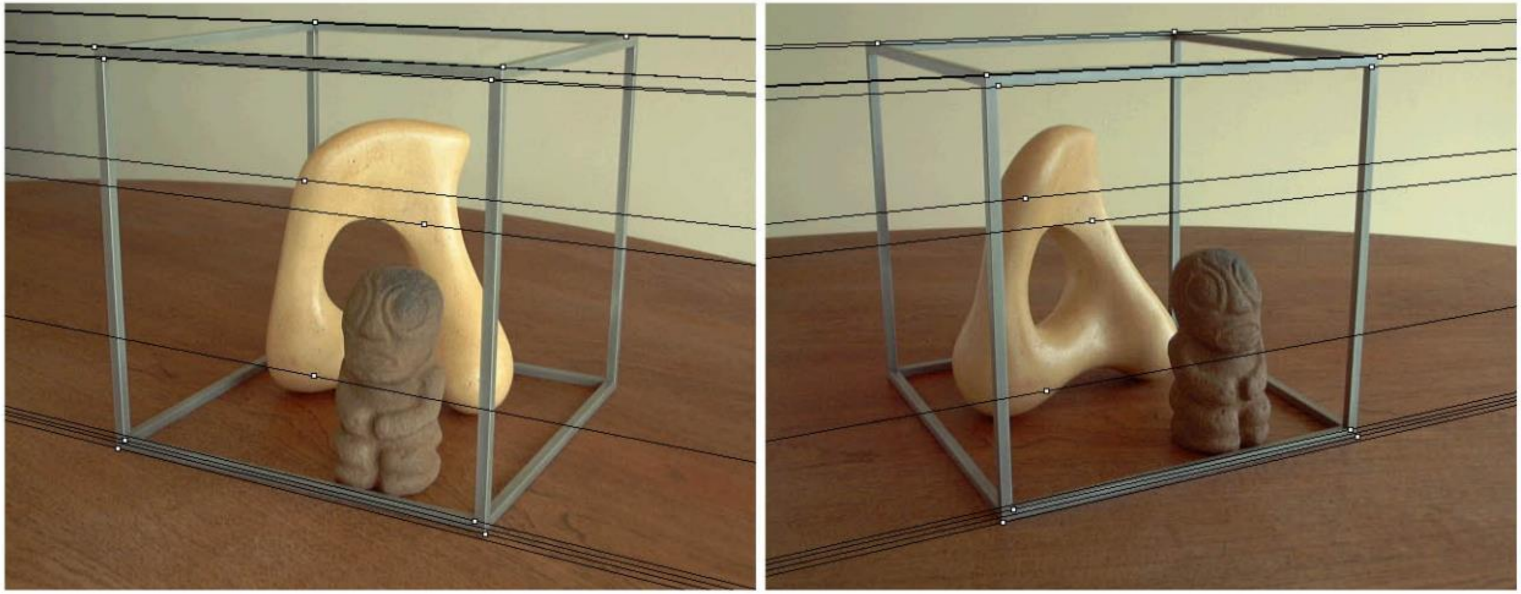
Given a point P1 in left image on epipolar line e1, can find epipolar line e2 provided we know relative orientations of cameras $\Rightarrow$ Requires camera calibration (see lecture 5)

Alternate approach: Stereo image rectification



- Reproject image planes onto a common plane parallel to the line between optical centers
- Epipolar line is horizontal after this transformation
- Two homographies (3x3 transforms), one for each input image reprojection, is computed. See:

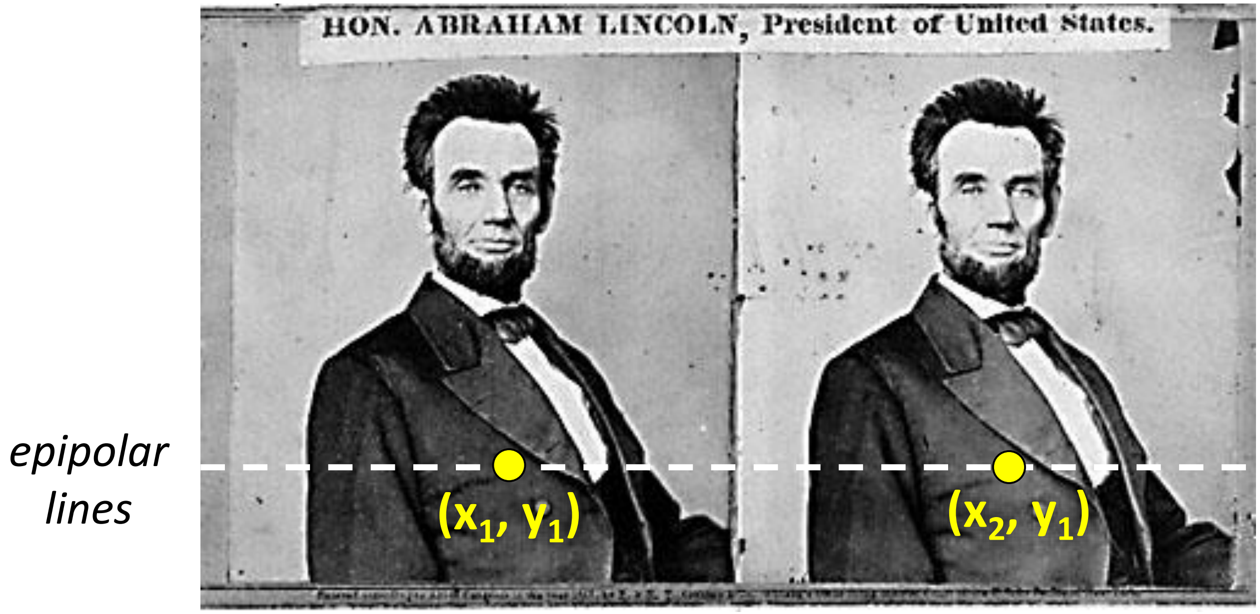
➢ C. Loop and Z. Zhang. Computing Rectifying Homographies for Stereo Vision. IEEE Conf. Computer Vision and Pattern Recognition, 1999.



Original stereo pair

After rectification

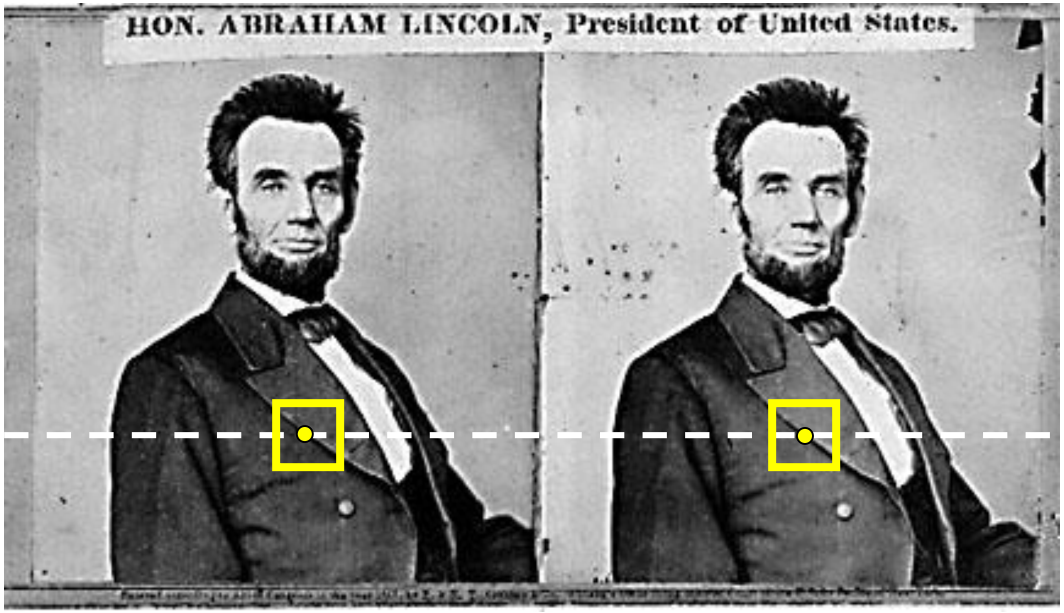
Epipolar geometry



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



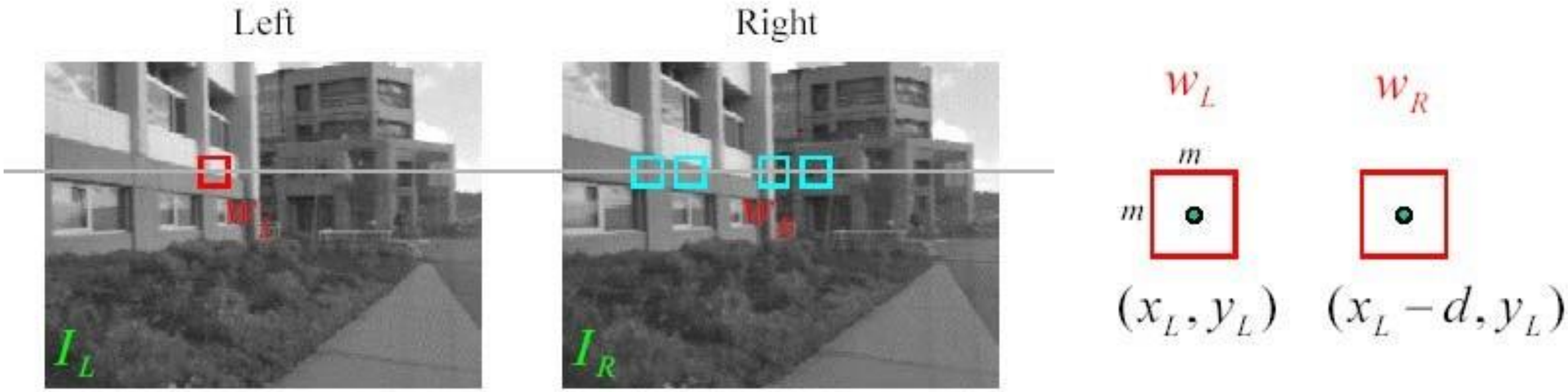
For each epipolar line

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***

Matching using Sum of Squared Differences (SSD)



w_L and w_R are corresponding m by m windows of pixels.

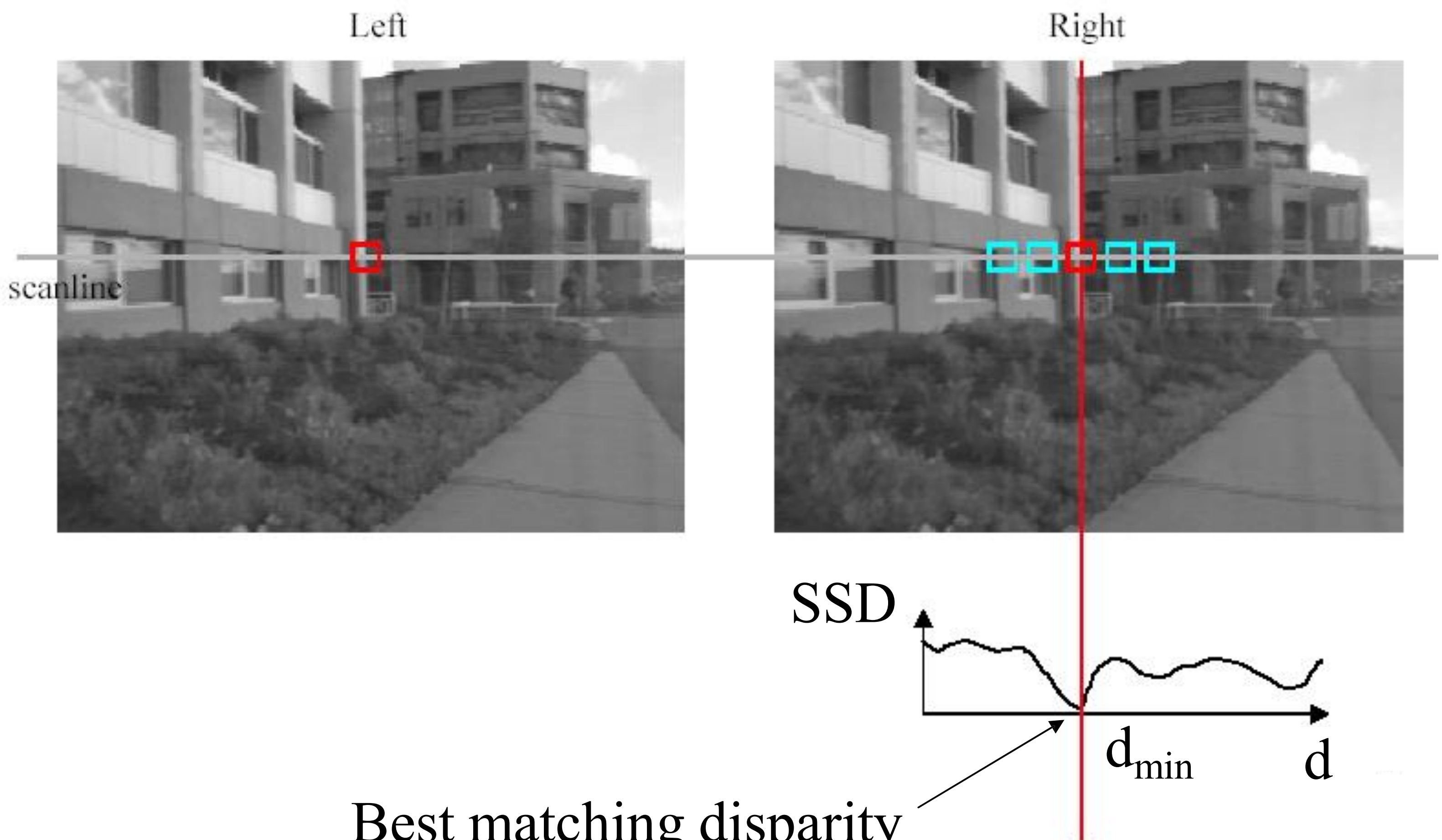
We define the window function :

$$W_m(x, y) = \{u, v \mid x - \tfrac{m}{2} \le u \le x + \tfrac{m}{2}, y - \tfrac{m}{2} \le v \le y + \tfrac{m}{2}\}$$

The SSD cost measures the intensity difference as a function of disparity :

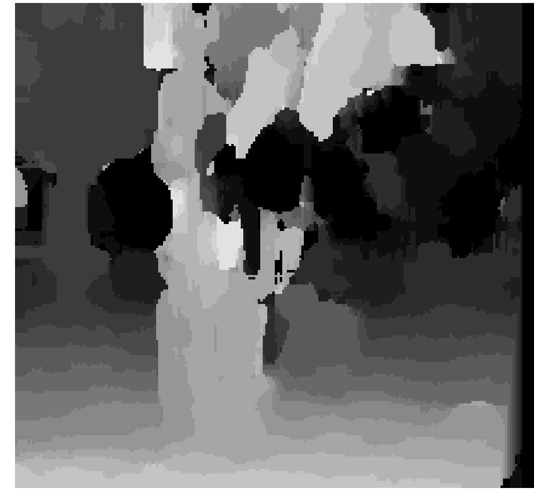
$$C_r(x, y, d) = \sum_{(u,v)\in W_m(x,y)} [I_L(u, v) - I_R(u - d, v)]^2$$

Stereo matching based on SSD



Window size

W = 3



W = 21

Effect of window size

- Smaller window
  - +
  - –
- Larger window
  - +
  - –

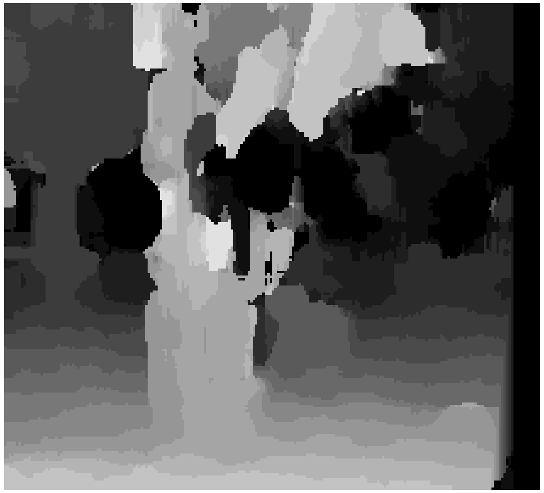
Better results with *adaptive window*

- T. Kanade and M. Okutomi, *A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment*,, Proc. International Conference on Robotics and Automation, 1991.
- D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. International Journal of Computer Vision, 28(2):155-174, July 1998

Problems with window size

Input stereo pair

W = 3



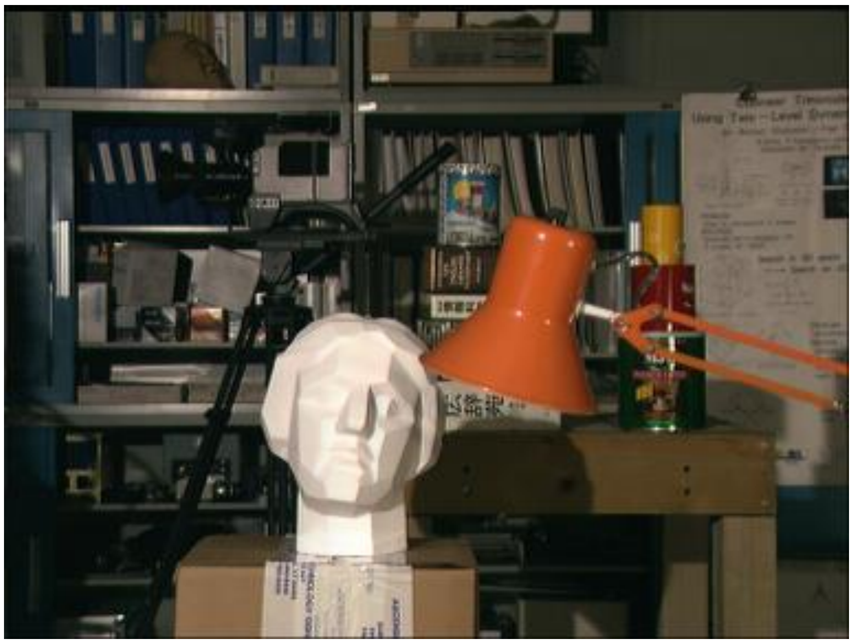
W = 21

Effect of window size W

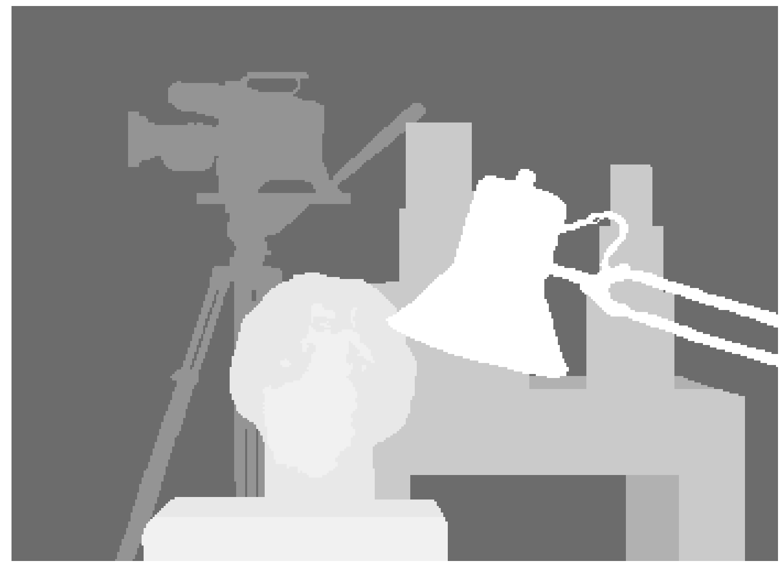
- Smaller window
  - \+ Good precision, more detail
  - – Sensitive to noise
- Larger window
  - \+ Robust to noise
  - – Reduced precision, less detail

Stereo results

- Data from University of Tsukuba
- Similar results on other images without ground truth

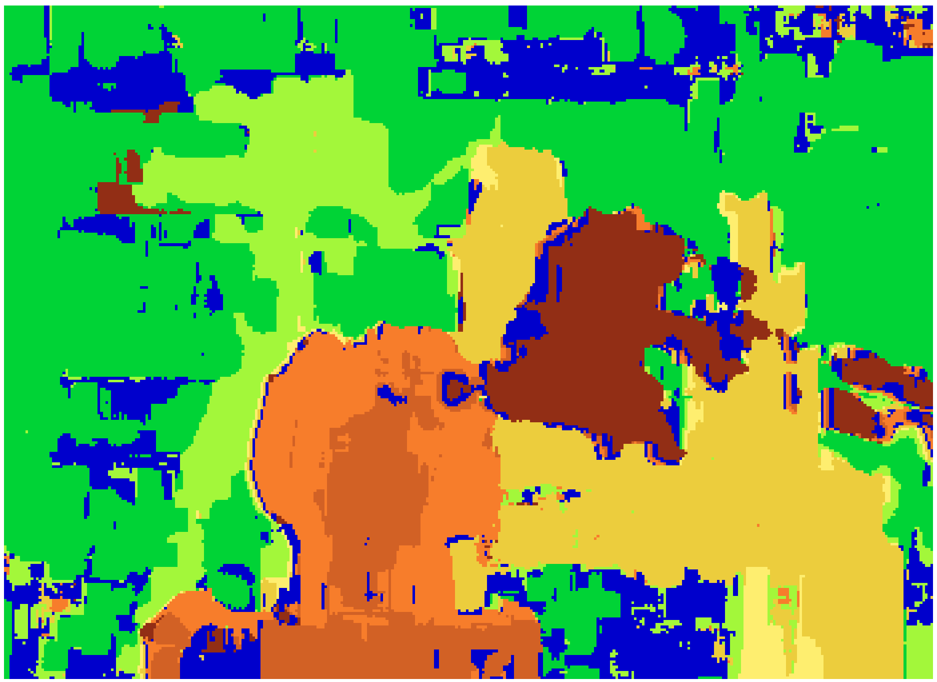


Scene



Ground truth

Results with window search



Window-based matching
(best window size)

Ground truth

Better methods exist...

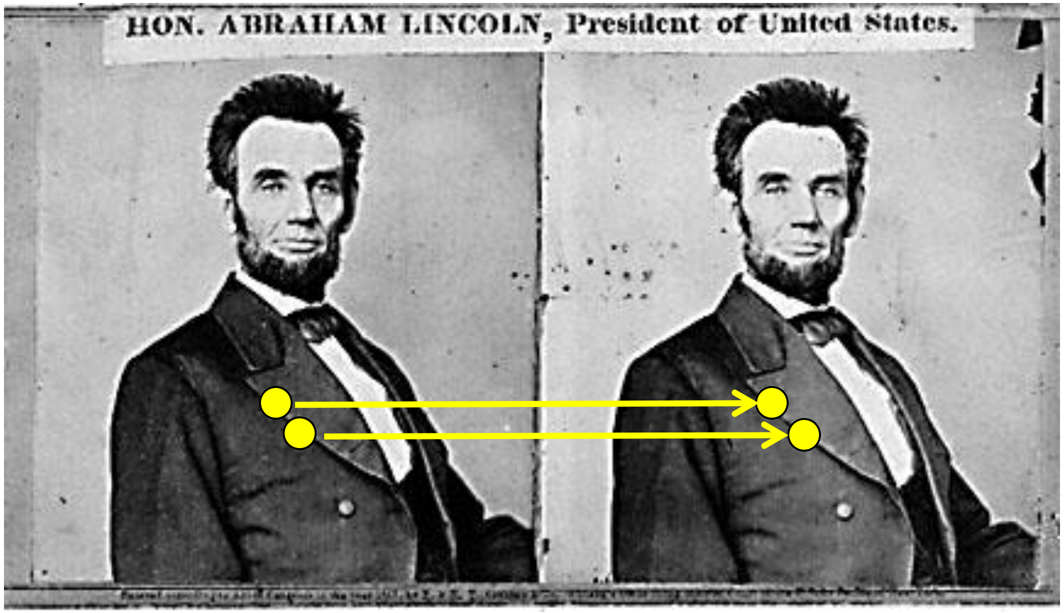
State of the Art method

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

Ground truth

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

Stereo as energy minimization



What defines a good stereo correspondence?

1. Match quality
   – Want each pixel to find a good match in the other image
2. Smoothness
   – If two pixels are adjacent, they should (usually) move about the same amount

Stereo as energy minimization

- Find disparity map *d* that minimizes an energy function $E(d)$

- Simple pixel / window matching

$$E(d) = \sum_{(x,y) \in I} C(x, y, d(x, y))$$

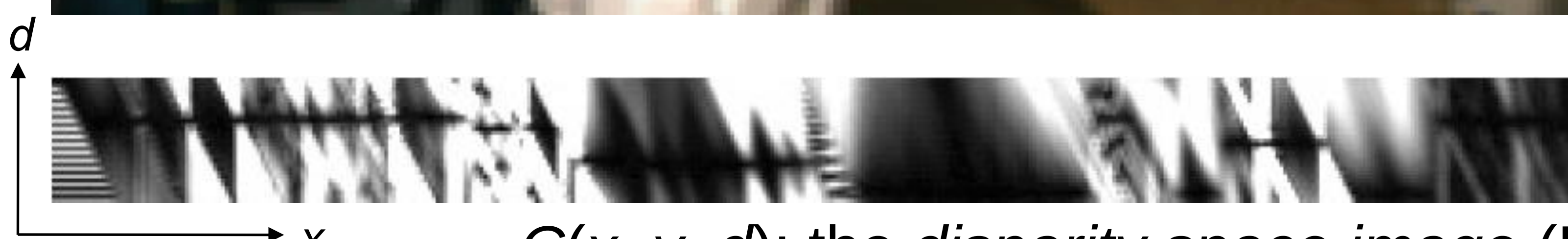
$C(x, y, d(x, y))$ = SSD distance between windows *I*(*x*, *y*) and *J*(*x* + *d*(*x*,*y*), *y*)

Stereo as energy minimization

$I(x, y)$

$J(x, y)$

y = 141



$C(x, y, d)$; the *disparity space image* (DSI)

Stereo as energy minimization

Simple pixel / window matching: choose the minimum of each column in the DSI independently:

$$d(x,y) = \arg\min_{d'} \; C(x,y,d')$$

Stereo as energy minimization

Better objective function

$$E(d) = \underbrace{E_d(d)}_{\text{match cost}} + \lambda \underbrace{E_s(d)}_{\text{smoothness cost}}$$

Want each pixel to find a good match in the other image

Adjacent pixels should (usually) move about the same amount

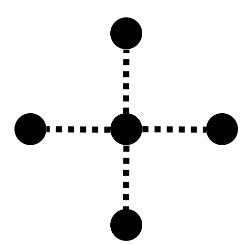
Stereo as energy minimization

$$E(d) = E_d(d) + \lambda E_s(d)$$

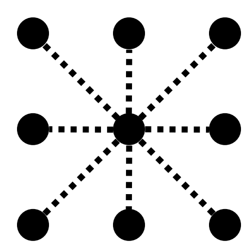
match cost: $$E_d(d) = \sum_{(x,y) \in I} C(x, y, d(x, y))$$

smoothness cost: $$E_s(d) = \sum_{(p,q) \in \mathcal{E}} V(d_p, d_q)$$

$\mathcal{E}$: set of neighboring pixels



4-connected neighborhood



8-connected neighborhood

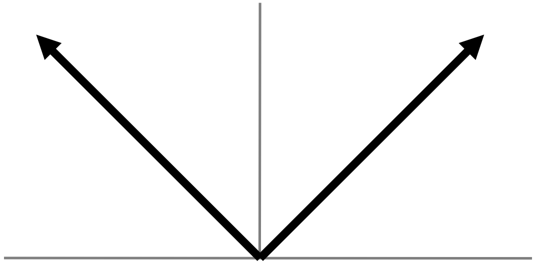
Smoothness cost

$$E_s(d) = \sum_{(p,q)\in\mathcal{E}} V(d_p, d_q)$$

How do we choose *V*?

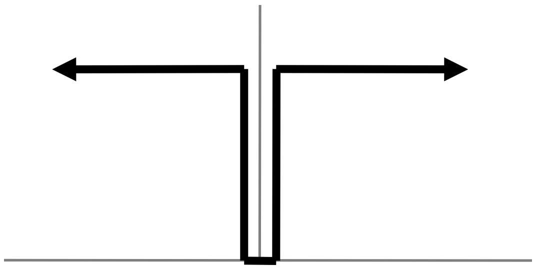
$$V(d_p, d_q) = |d_p - d_q|$$

L_1 distance



$$V(d_p, d_q) = \begin{cases} 0 & \text{if } d_p = d_q \\ 1 & \text{if } d_p \neq d_q \end{cases}$$

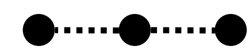
"Potts model"



Dynamic programming

$$E(d) = E_d(d) + \lambda E_s(d)$$

Can minimize this independently per scanline using dynamic programming (DP)



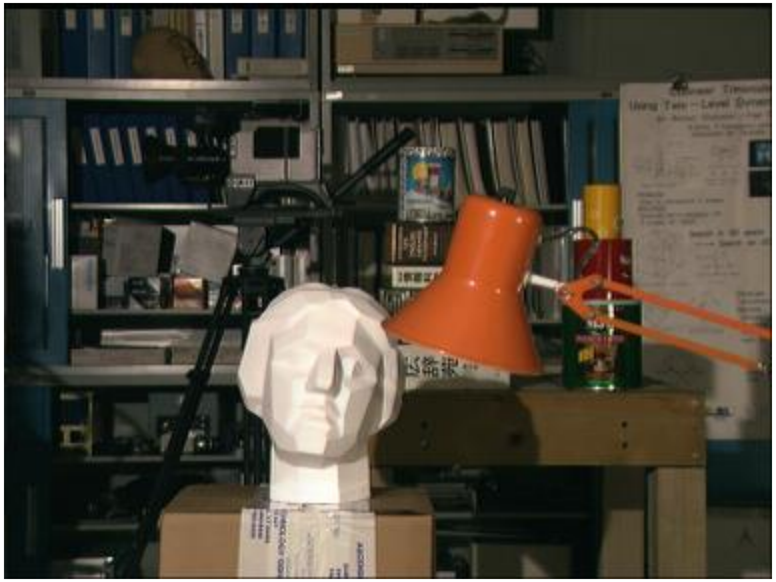
$D(x, y, d)$ **: minimum cost of solution such that *d*(*x*,*y*) = *d***

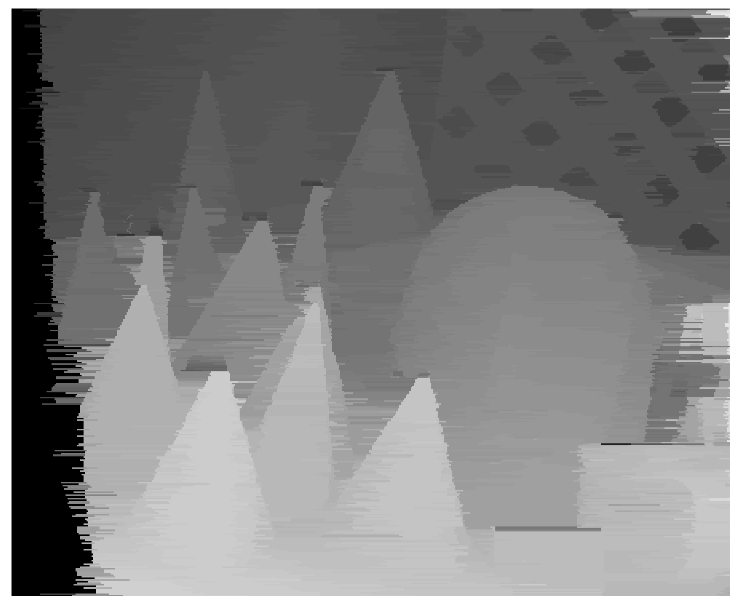
$$D(x, y, d) = C(x, y, d) + \min_{d'} \{D(x-1, y, d') + \lambda \left| d - d' \right|\}$$

Dynamic programming

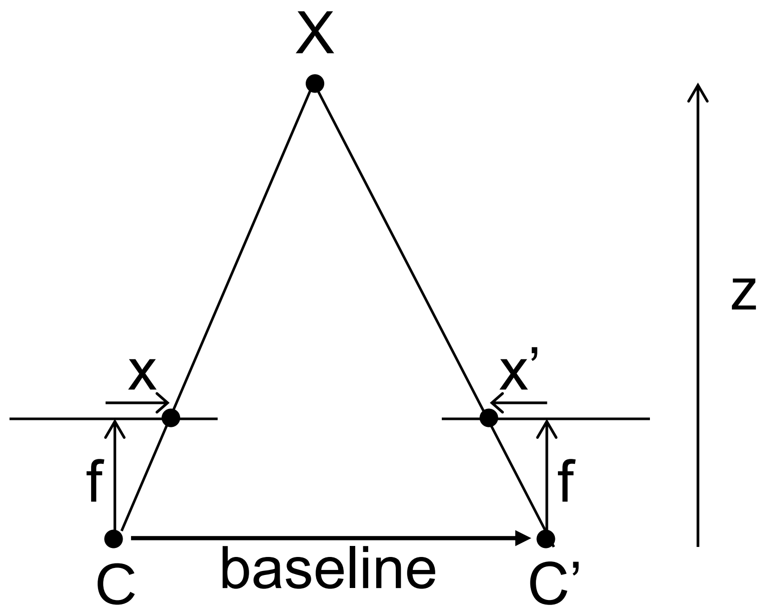
Finds “smooth” path through DPI from left to right

Dynamic Programming





Depth from disparity



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

Questions?

Real-time stereo

Nomad robot searches for meteorites in Antartica
http://www.frc.ri.cmu.edu/projects/meteorobot/index.html

Used for robot navigation (and other tasks)

- Several software-based real-time stereo techniques have been developed (most based on simple discrete search)

Stereo reconstruction pipeline

Steps

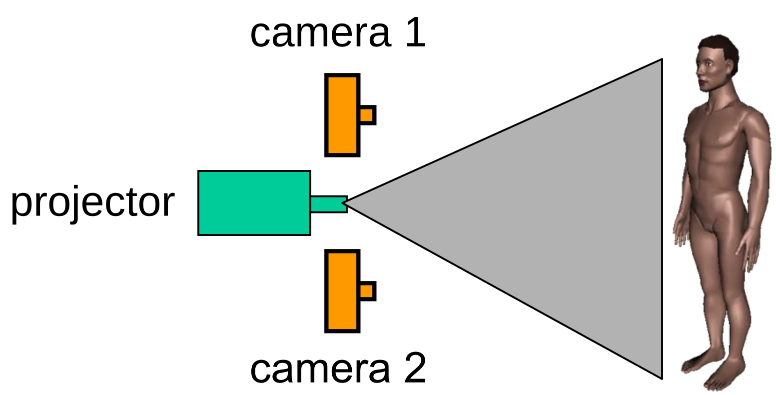
- Calibrate cameras
- Rectify images
- Compute disparity
- Estimate depth

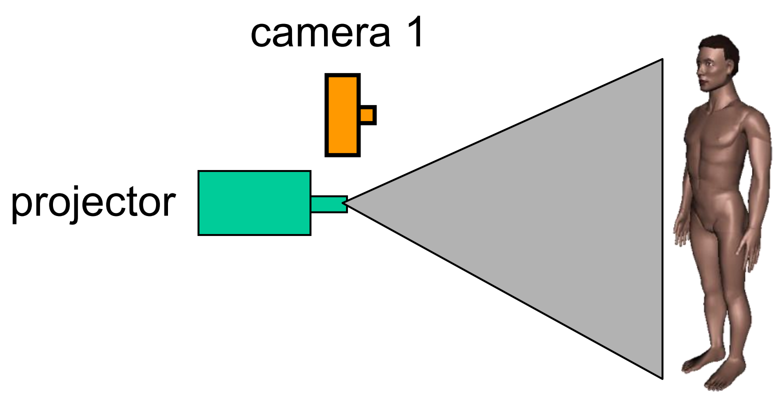
What will cause errors?

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

Active stereo with structured light

Li Zhang's one-shot stereo

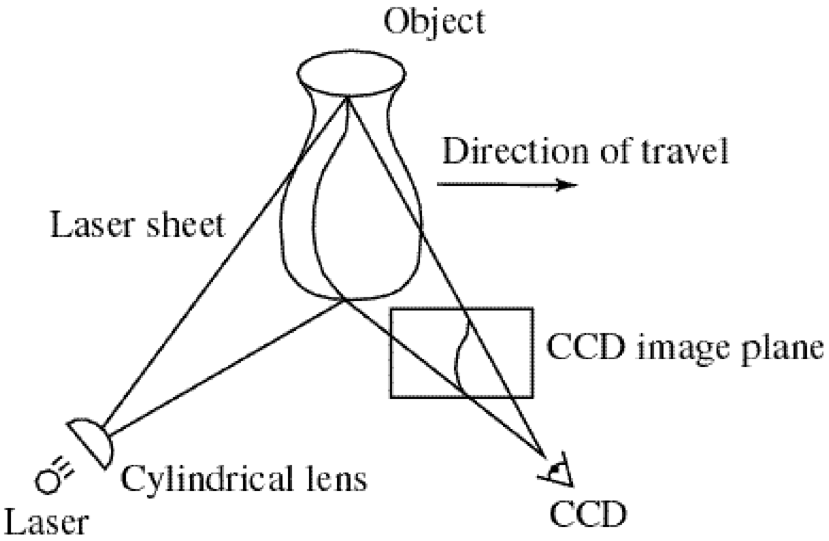


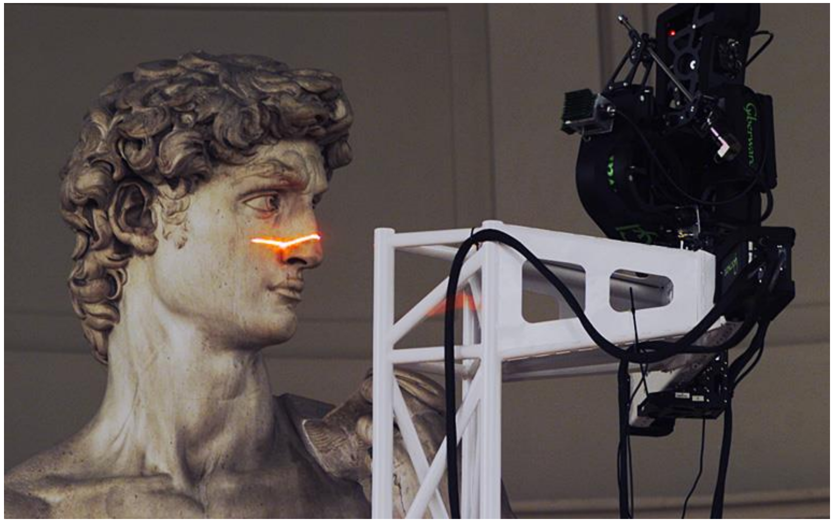


Project "structured" light patterns onto the object

- simplifies the correspondence problem

Laser scanning





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

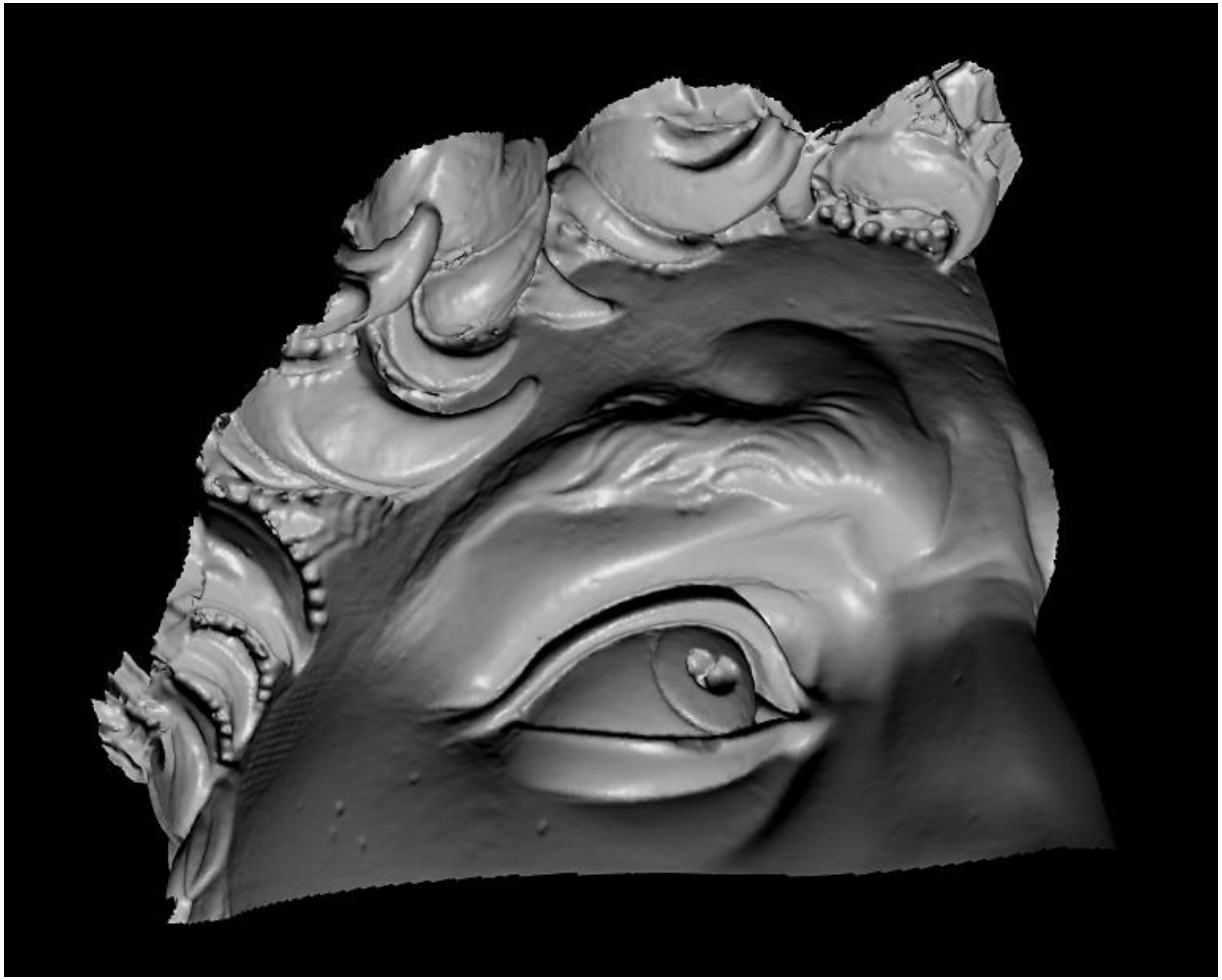
Optical triangulation

- 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

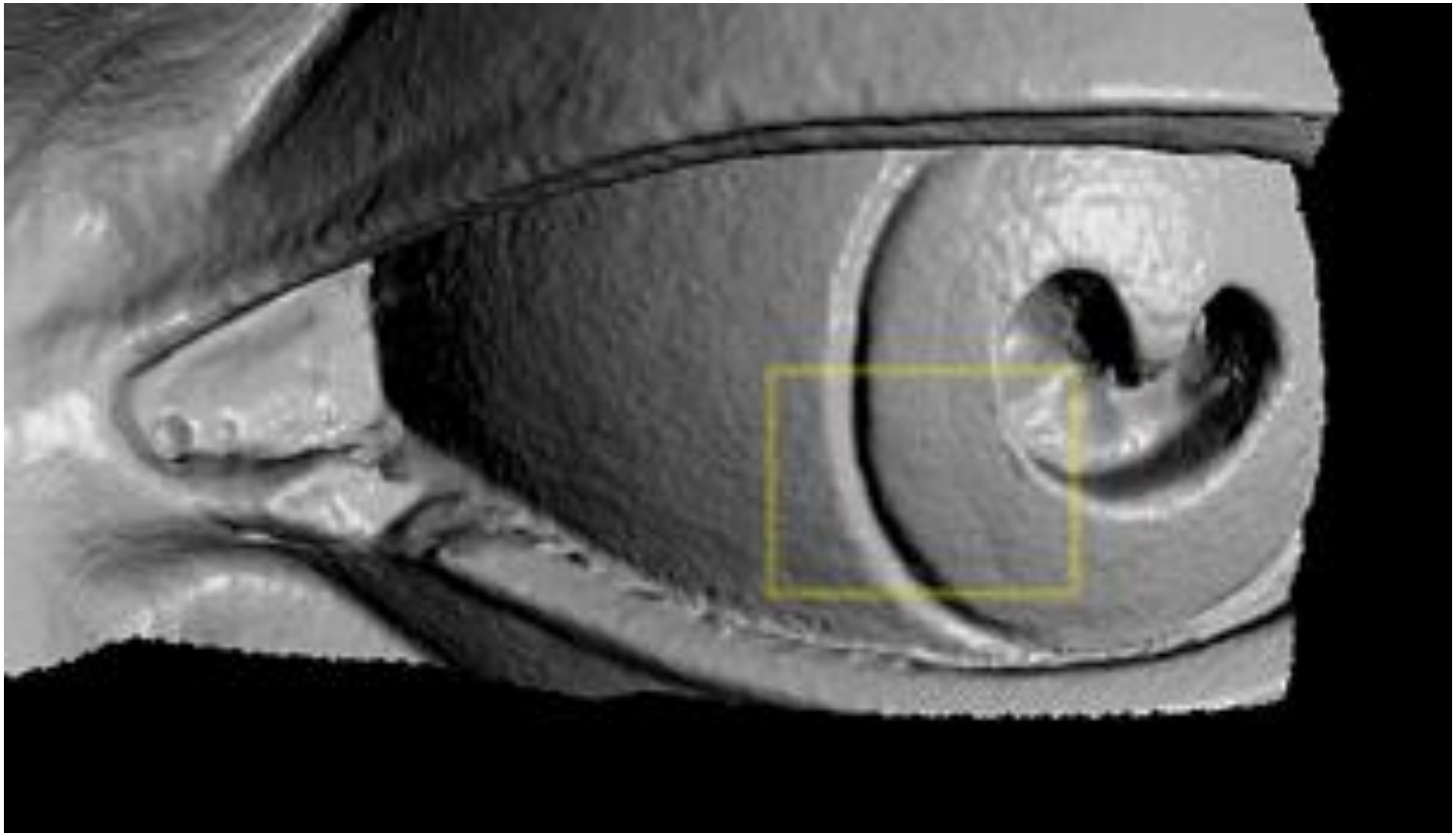
The Digital Michelangelo Project, Levoy et al.

Laser scanned models



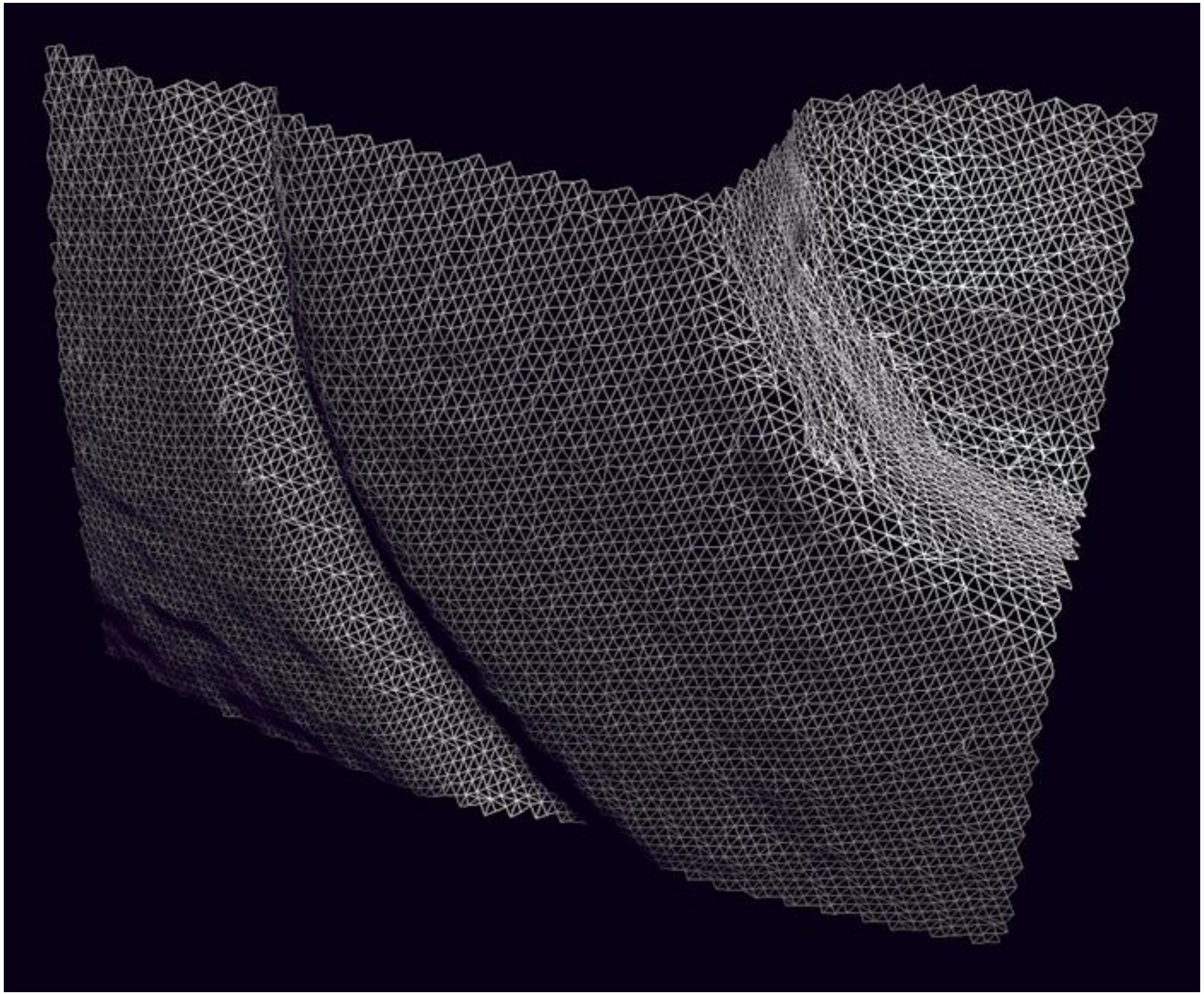
The Digital Michelangelo Project, Levoy et al.

Laser scanned models



The Digital Michelangelo Project, Levoy et al.

Laser scanned models



The Digital Michelangelo Project, Levoy et al.