CS5670: Stereo

Presented by Abe Davis

Most slides from Noah Snavely

Who probably based some of them on slides from other academics, because that's what everyone does…

Coronavirus Updates

- Subsequent lectures will be virtual until otherwise noted, meaning either:
  - Presented live over Zoom, or
  - Posted as a video with scheduled office hours for questions (reverse classroom style)

- Abe's Office hours today:
  - Right after class, shortened
  - Let me know if you have questions but can't make this
  - Will try to schedule office hours over Zoom moving forward

- Please be patient with us as we adapt
  - Clarifications on last lecture will be sent out and/or presented on Monday
  - Some grading may take a bit longer than usual, but we are working on it
  - We hope to be back on pace by the end of next week

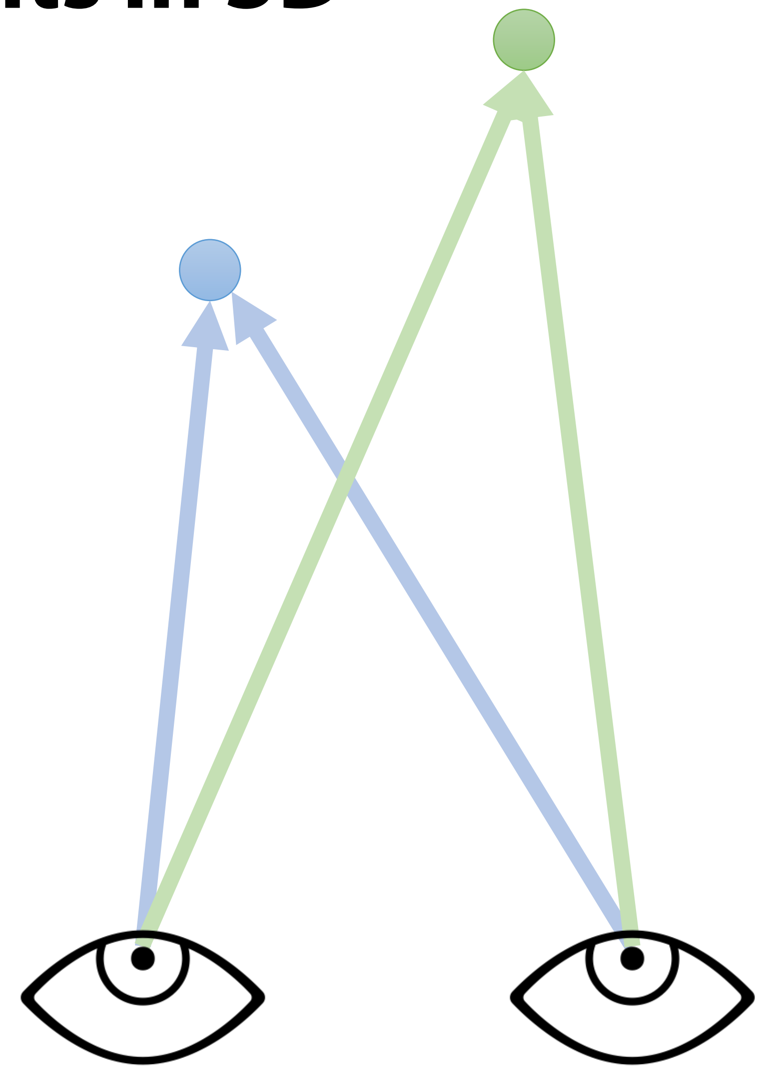
Announcements

- Project 3 is out
  - Code is due Monday, March 23, by 11:59pm
  - Artifact due Wednesday, March 25, by 11:59pm

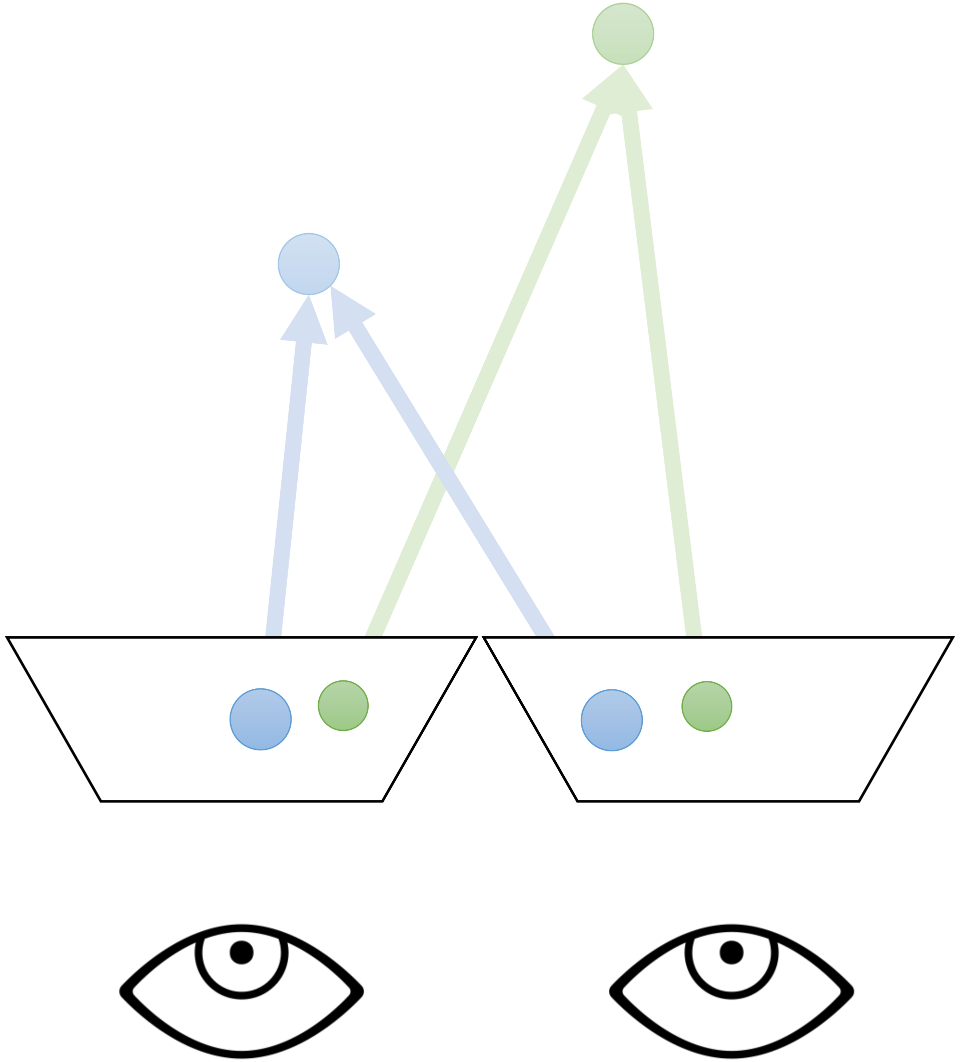
- Default groups are imported from Project 2
  - You can change these if you want

Stereo Vision as Localizing Points in 3D

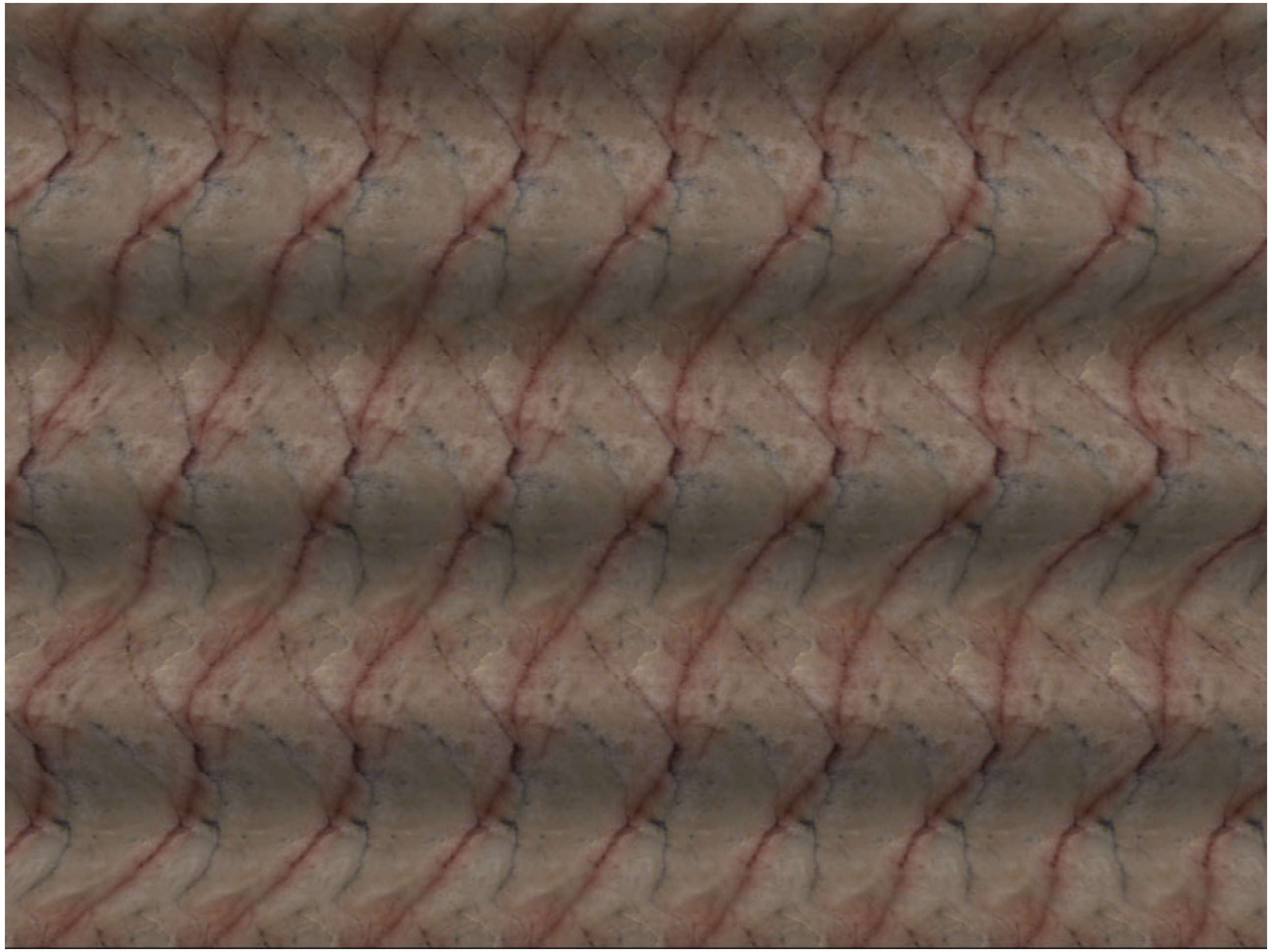
- An object point will project to some point in our image
- That image point corresponds to a ray in the world
- Two rays intersect at a single point, so if we want to localize points in 3D we need 2 eyes



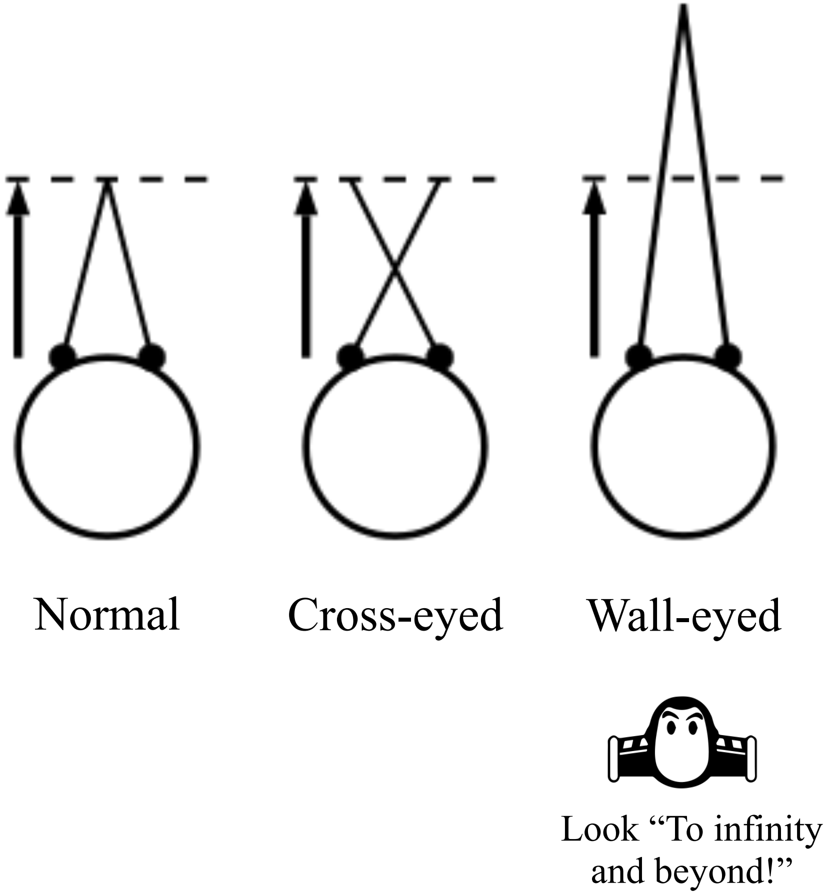
Stereo Trickery: VR



Stereo Trickery: Autostereograms



Single image stereogram, by [Niklas Een]

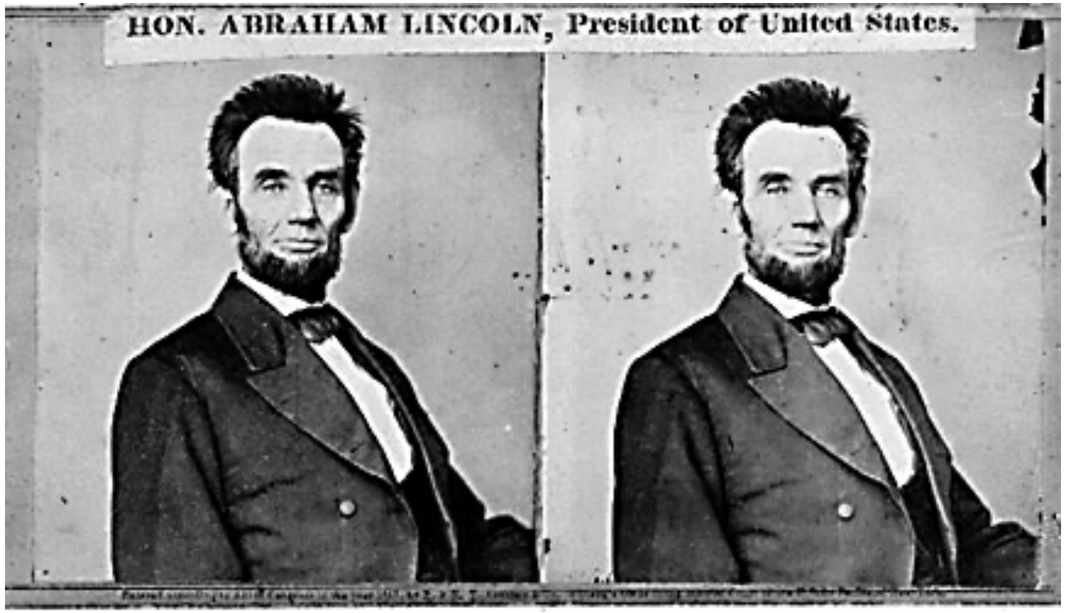


Stereo Trickery: Old-school 3D Glasses

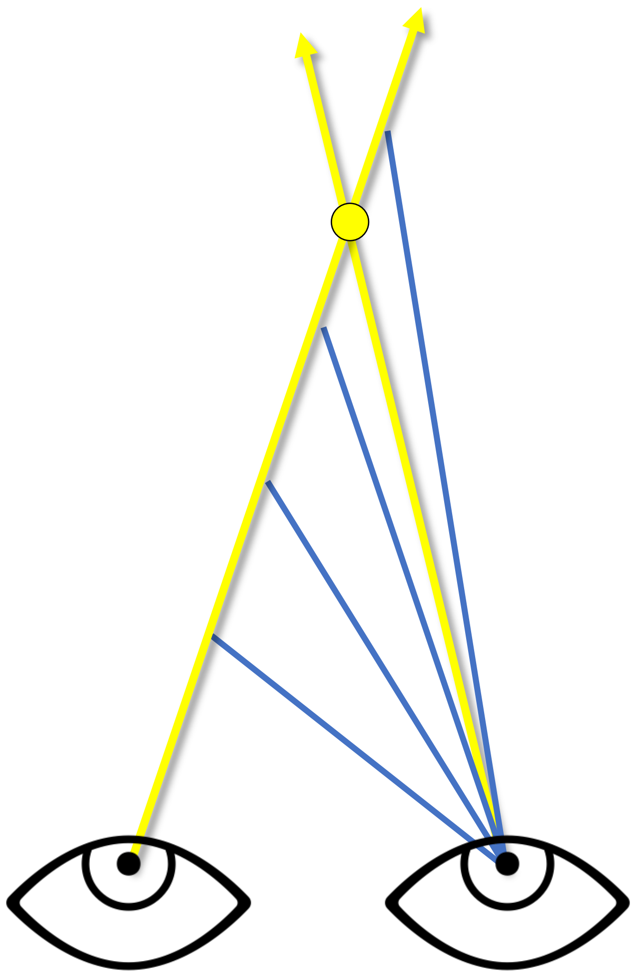
Mark Twain at Pool Table", no date, UCR Museum of Photography

Stereo

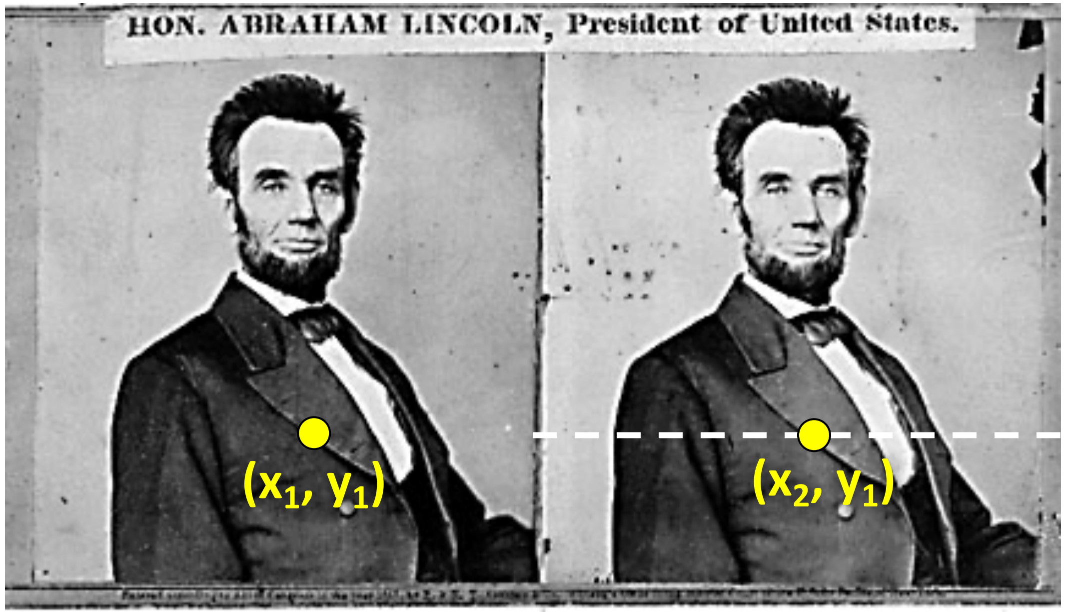
- Given two images from different viewpoints
  - How can we compute the depth of each point in the image?
  - Based on *how much each pixel moves* between the two images



Epipolar geometry



epipolar
lines



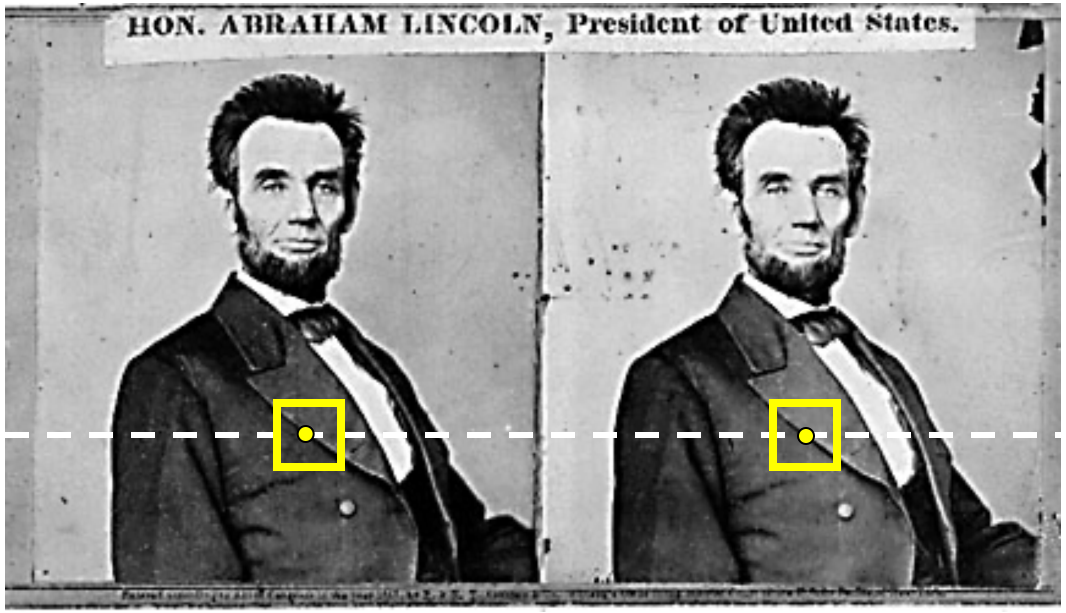
Two images captured by a purely horizontal translating camera
(*rectified* stereo pair)

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

Your basic stereo matching algorithm

- **Match Pixels in Conjugate Epipolar Lines**
  - Assume brightness constancy
  - This is a challenging problem
  - Hundreds of approaches
    - A good survey and evaluation: http://www.middlebury.edu/stereo/

Your basic stereo algorithm



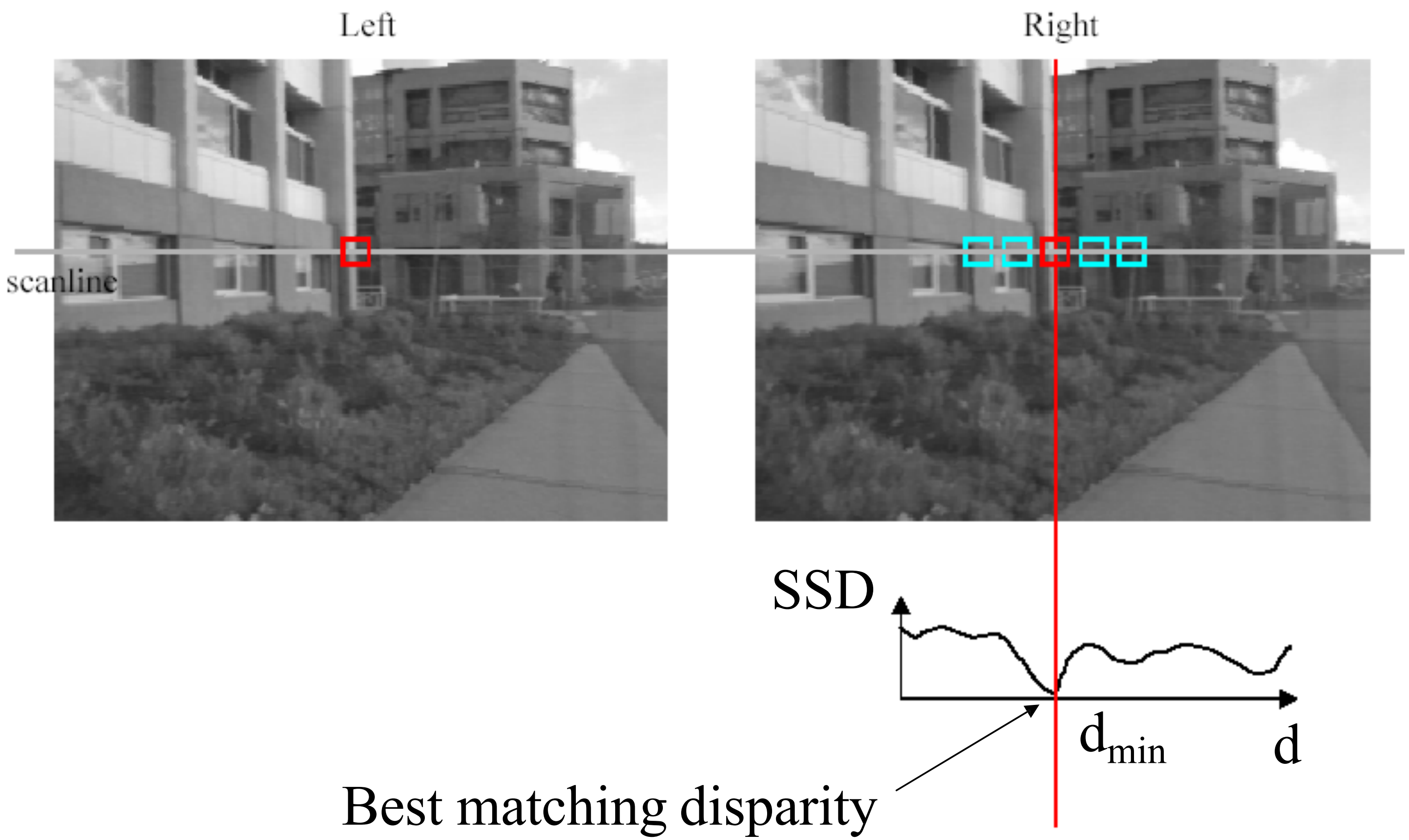
For each epipolar line

For each pixel in the left image

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

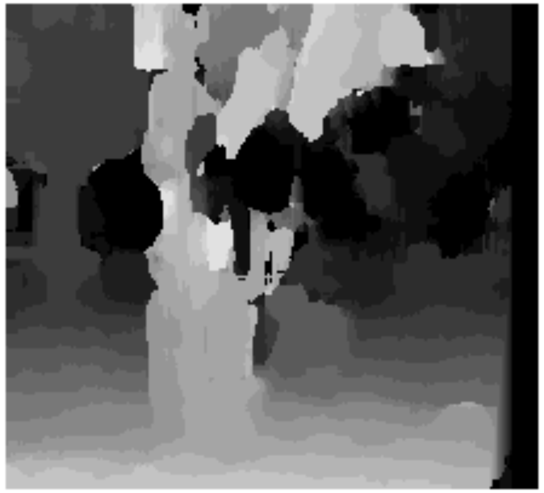
Improvement: match ***windows***

Stereo matching based on SSD



Window size

W = 3



W = 20

Effect of window size

- Smaller window
  - +
  - •
- Larger window
  - +
  - •

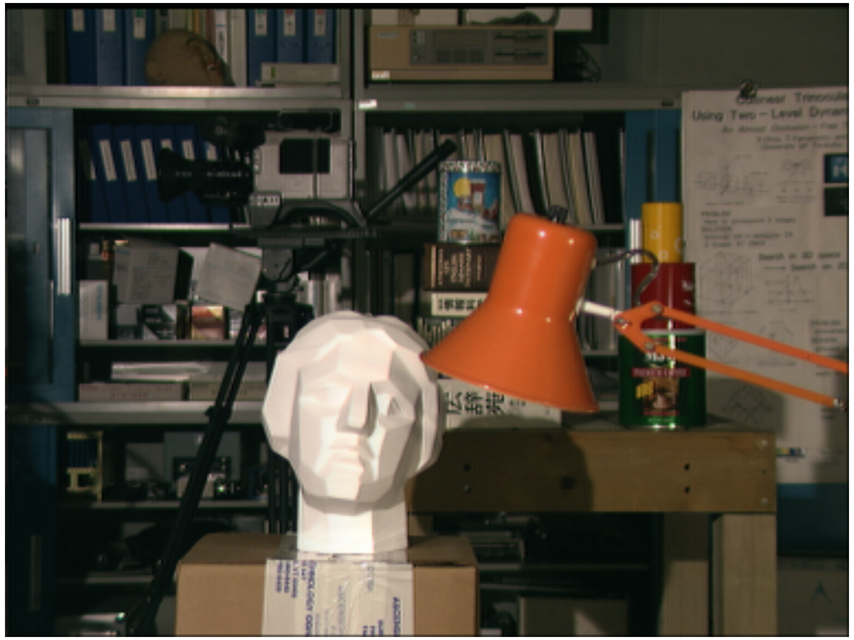
Better results with *adaptive window*

- 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

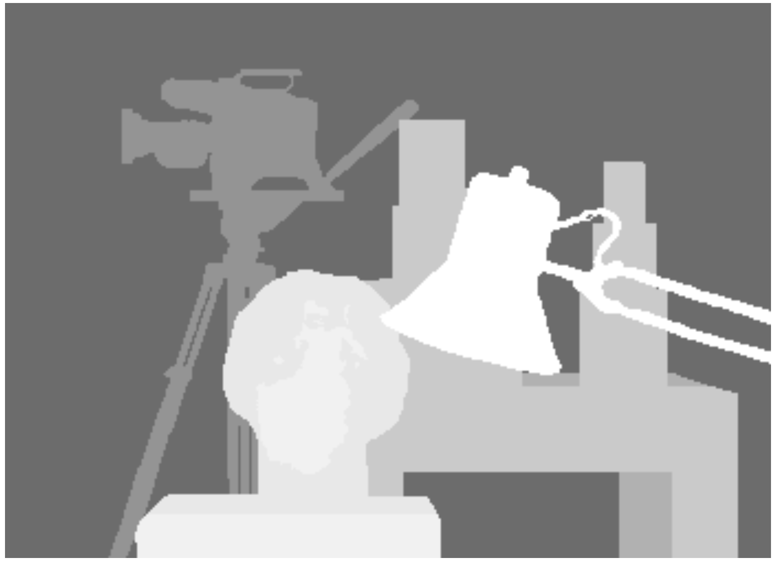
Stereo results

– Data from University of Tsukuba

– Similar results on other images without ground truth

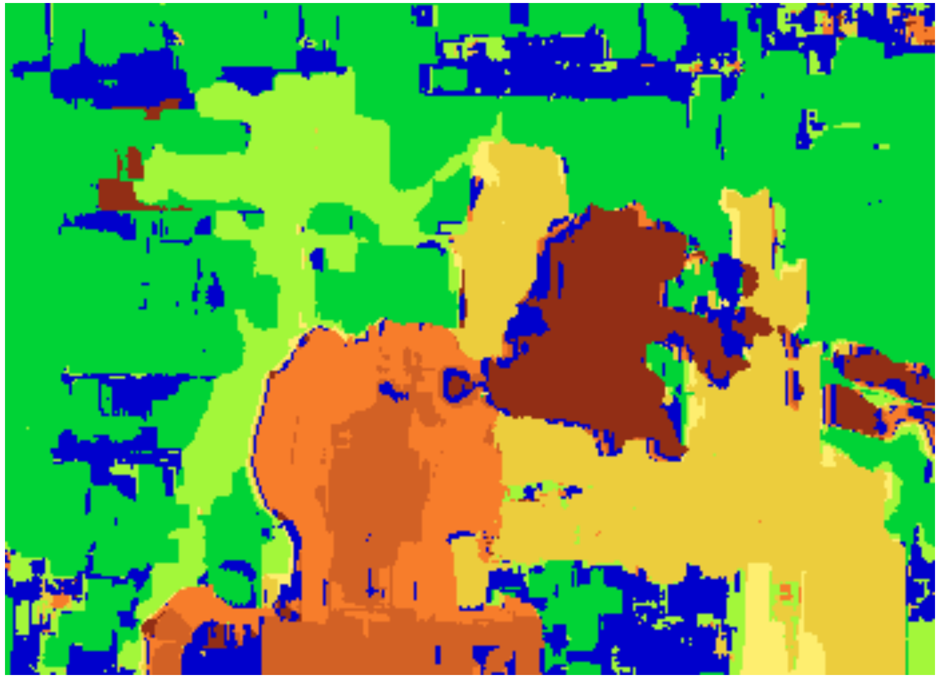


Scene



Ground truth

Results with window search



Window-based matching
(best window size)

Ground truth

Better methods exist...

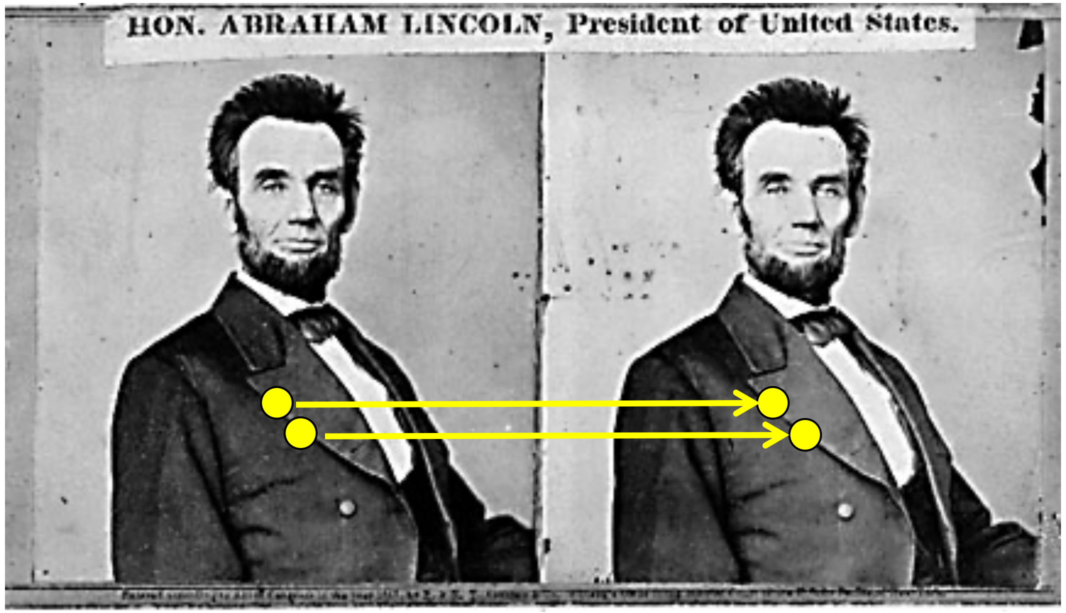
State of the art method (circa 1999)

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

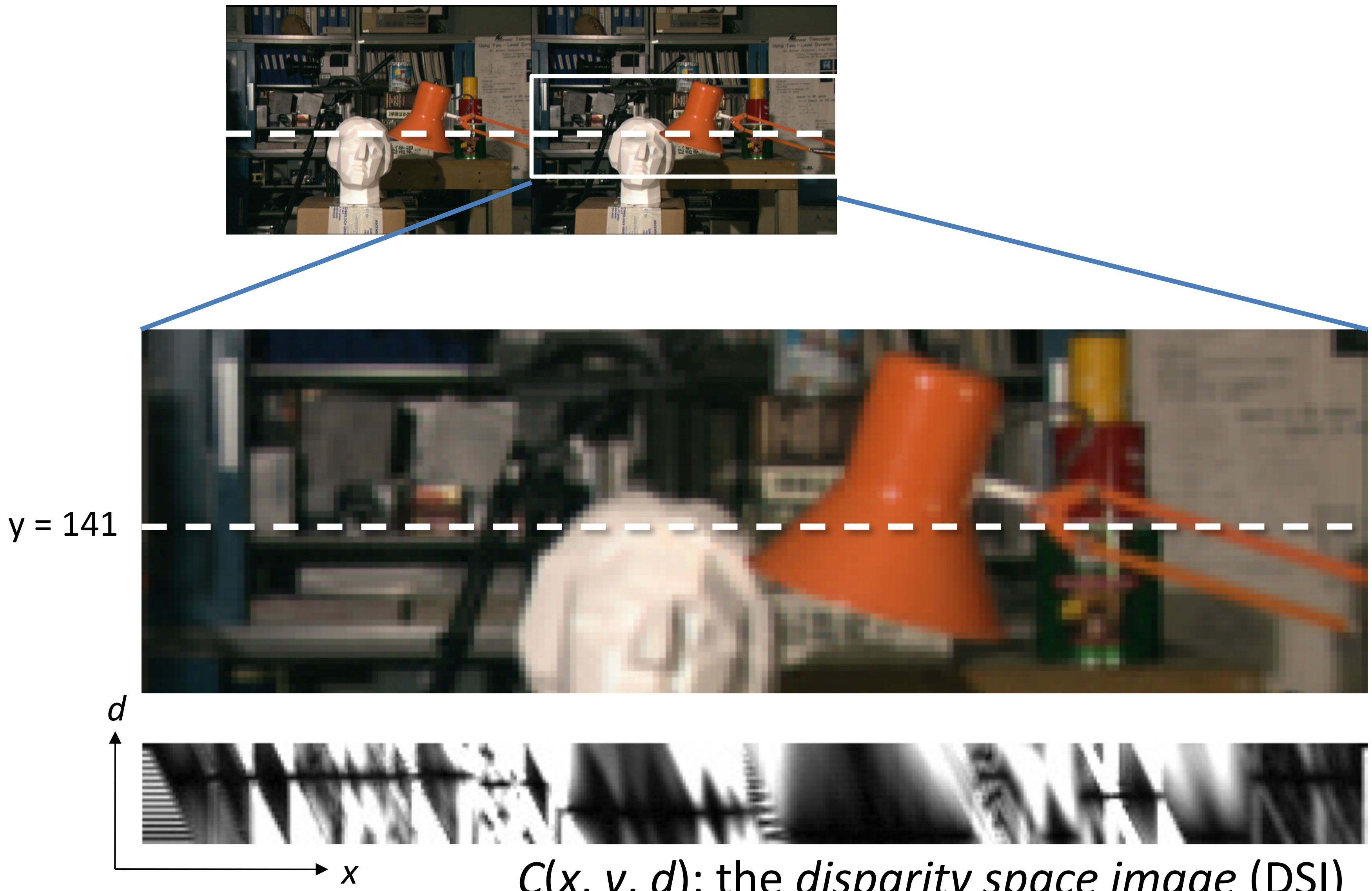
- Find disparity map *d* that minimizes an energy function $E(d)$

- 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



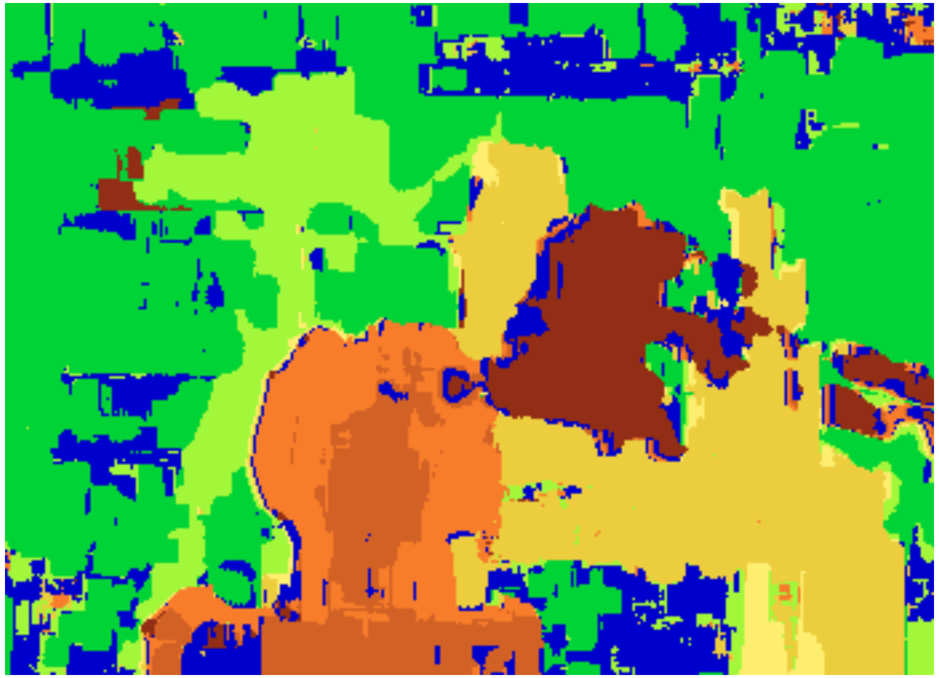
$C(x, y, d)$; the *disparity space image* (DSI)

Stereo as energy minimization

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

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

Greedy selection of best match



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

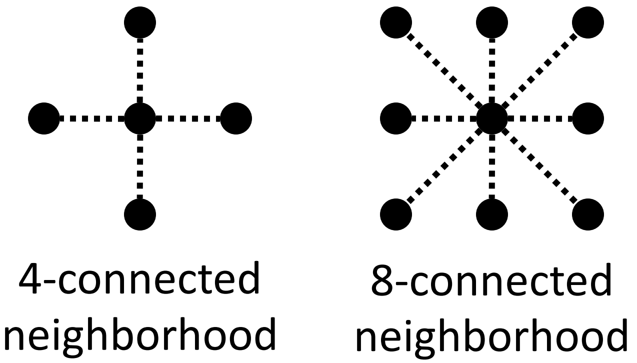
Stereo as energy minimization

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

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



Smoothness cost

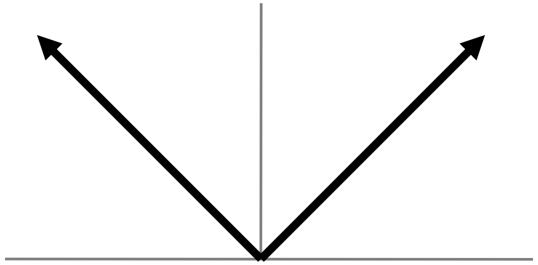
Now the energy contributed by one pixel depends on the disparity assigned to its neighbors!

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

How do we choose *V*?

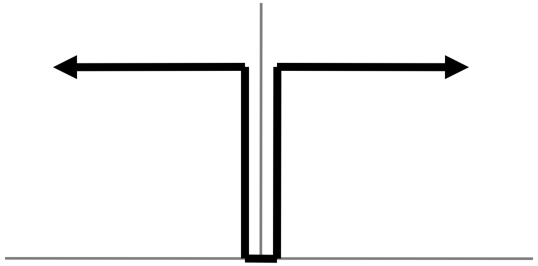
$$V(d_p, d_q) = |d_p - d_q|$$

L_1 distance



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

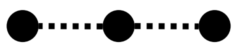
"Potts model"



Dynamic programming

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

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



- Basic idea: incrementally build a table of costs *D* one column at a time

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

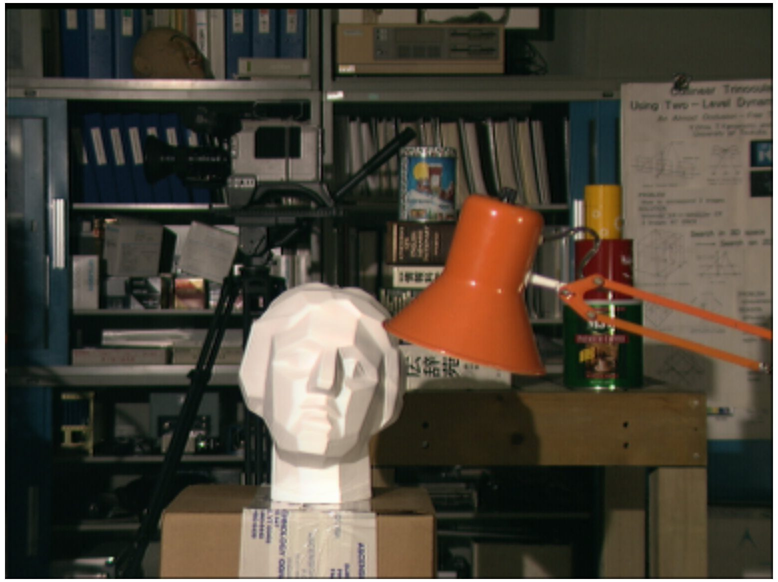
Base case: $D(0, y, i) = C(0, y, i), i = 0, \ldots, L$ (L = max disparity)

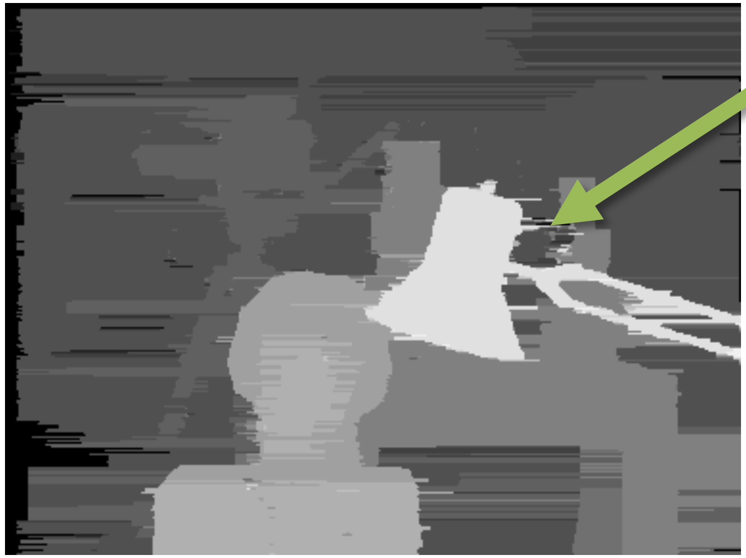
Recurrence: $D(x, y, i) = C(x, y, i) + \min_{j \in \{0,1,\ldots,L\}} D(x-1, y, j) + \lambda |i - j|$

Dynamic programming

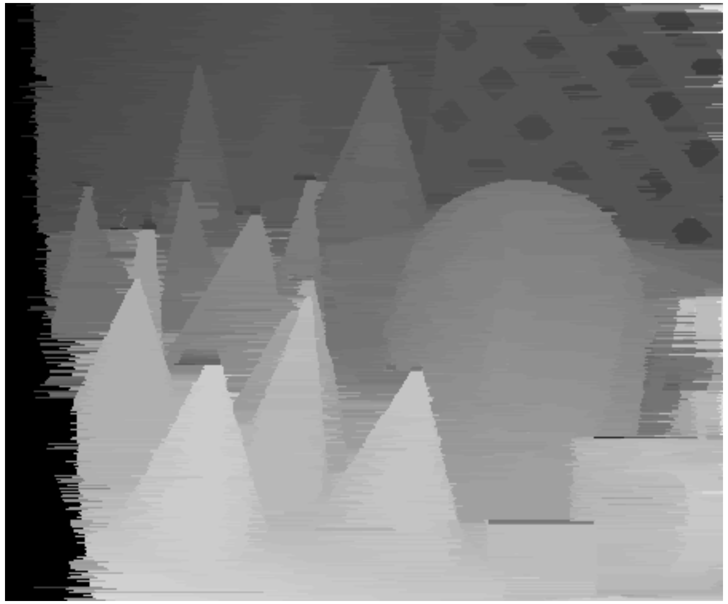
- Finds “smooth”, low-cost path through DPI from left to right

Dynamic Programming



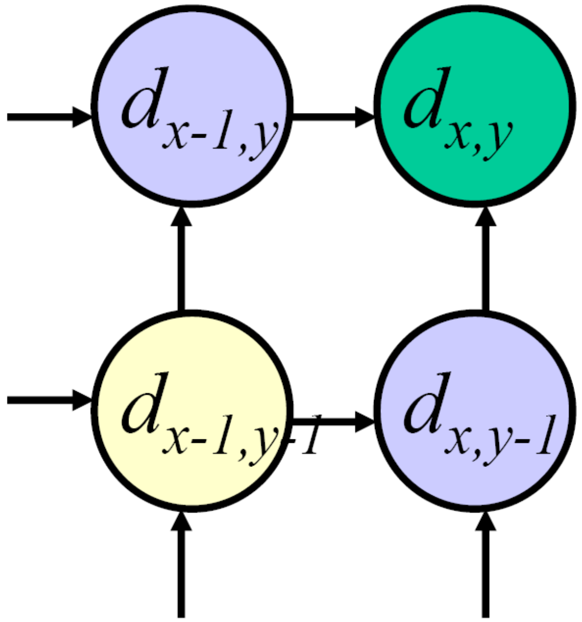


What's with the streaks?



Dynamic programming

- Can we apply this trick in 2D as well?



- No: $d_{x,y-1}$ and $d_{x-1,y}$ may depend on different values of $d_{x-1,y-1}$

Slide credit: D. Huttenlocher

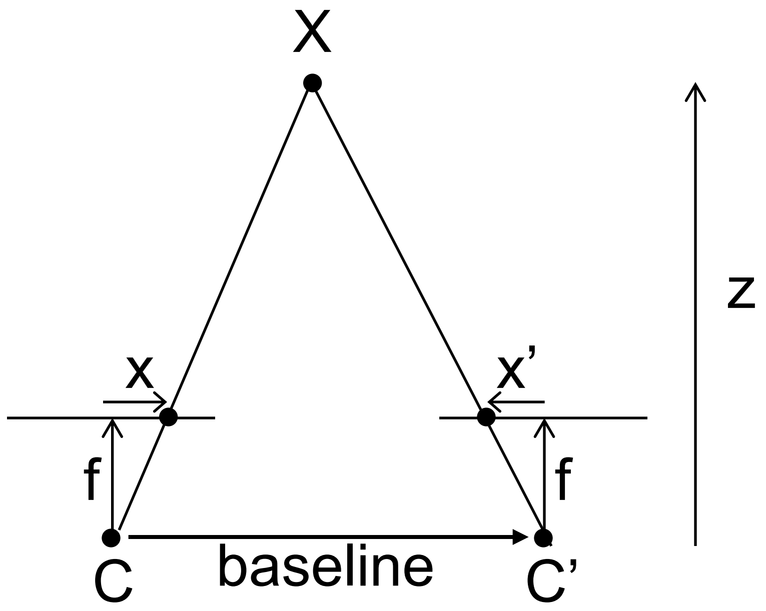
Stereo as a minimization problem

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

- The 2D problem has many local minima
  - Gradient descent doesn't work well
- And a large search space
  - n x m image w/ k disparities has k^{nm} possible solutions
  - Finding the global minimum is NP-hard in general
- Good approximations exist… we'll see this soon

Questions?

Depth from disparity



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

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

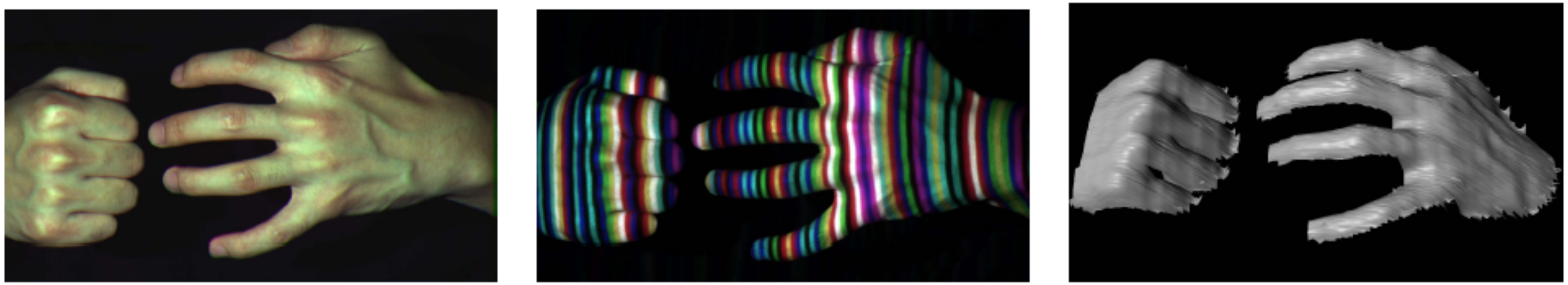
Stereo reconstruction pipeline

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

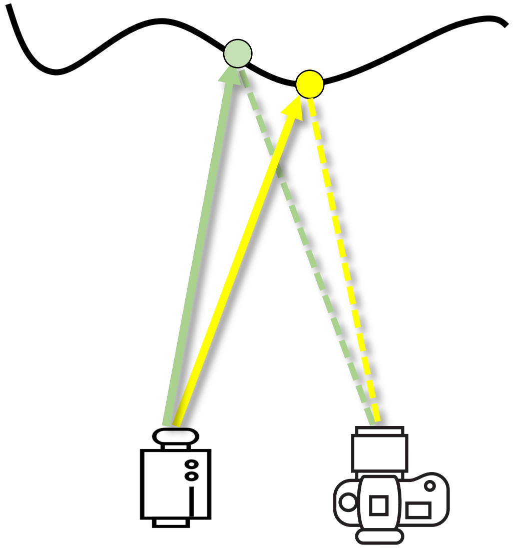
What will cause errors?

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

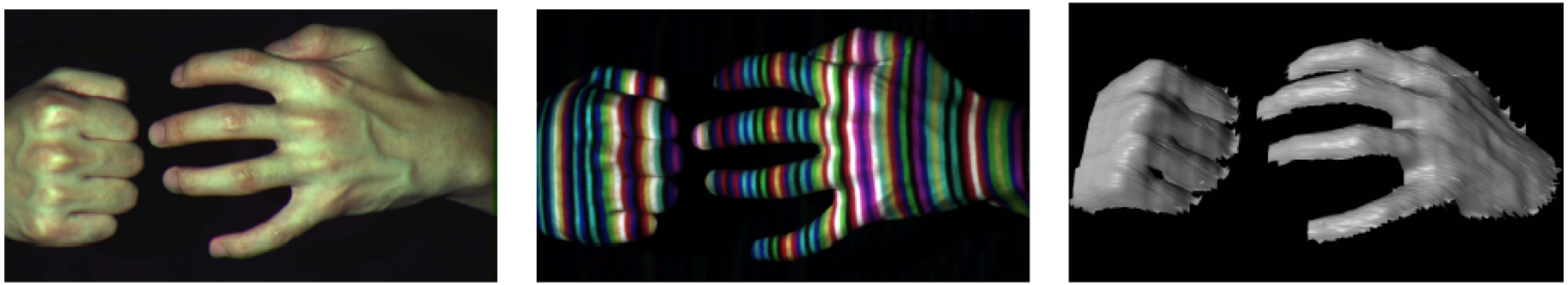
Active stereo with structured light



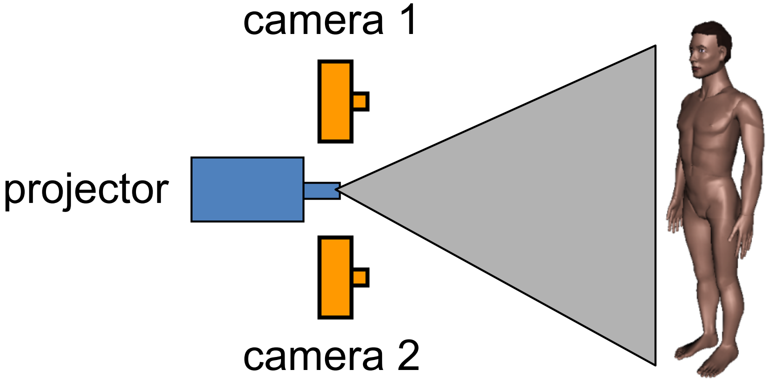
Li Zhang's one-shot stereo

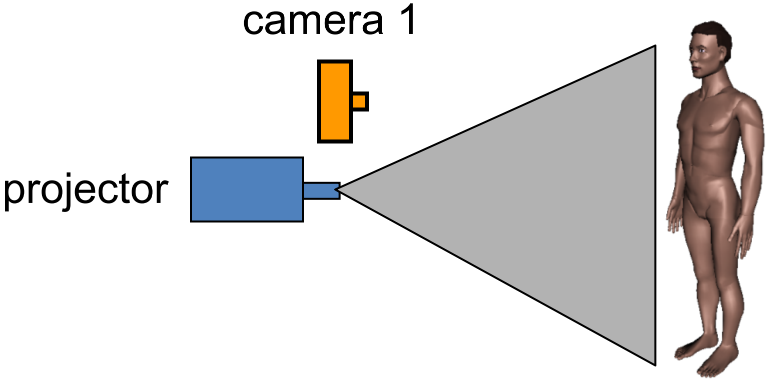


Active stereo with structured light



Li Zhang's one-shot stereo



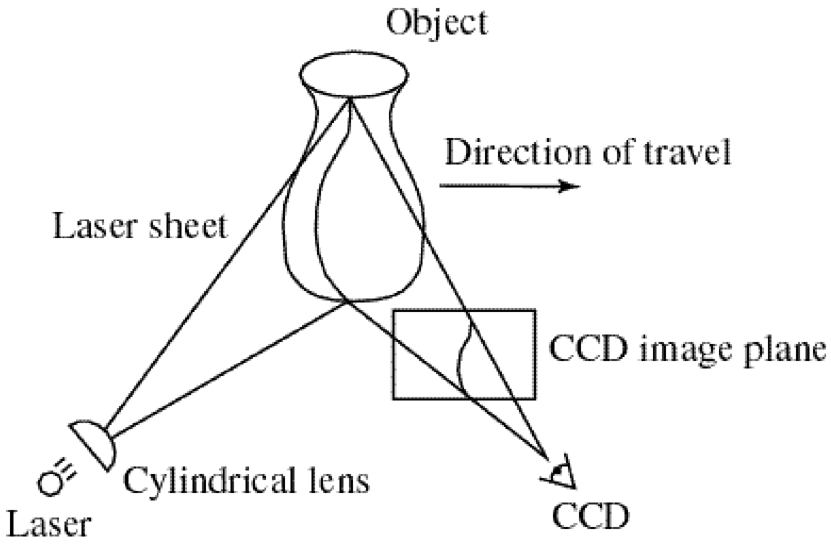


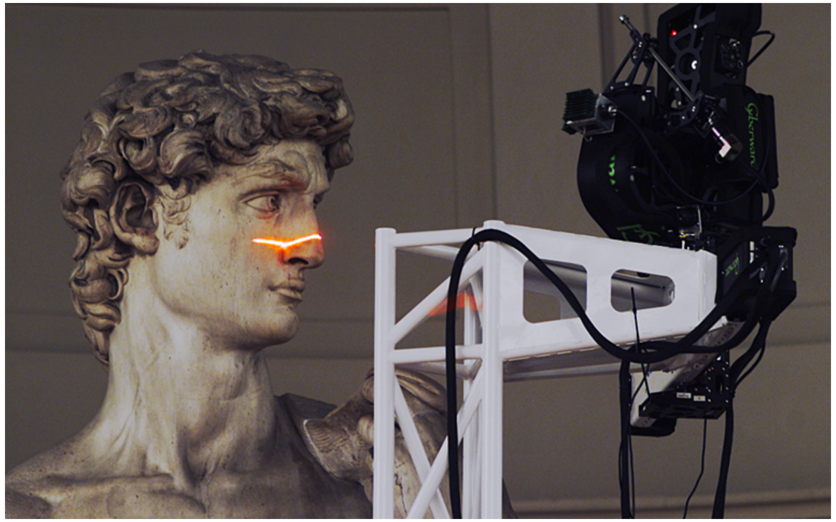
- Project "structured" light patterns onto the object
  - simplifies the correspondence problem
  - basis for active depth sensors, such as Kinect and iPhone X (using IR)

Active stereo with structured light

https://ios.gadgethacks.com/news/watch-iphone-xs-30k-ir-dots-scan-your-face-0180944/

Laser scanning





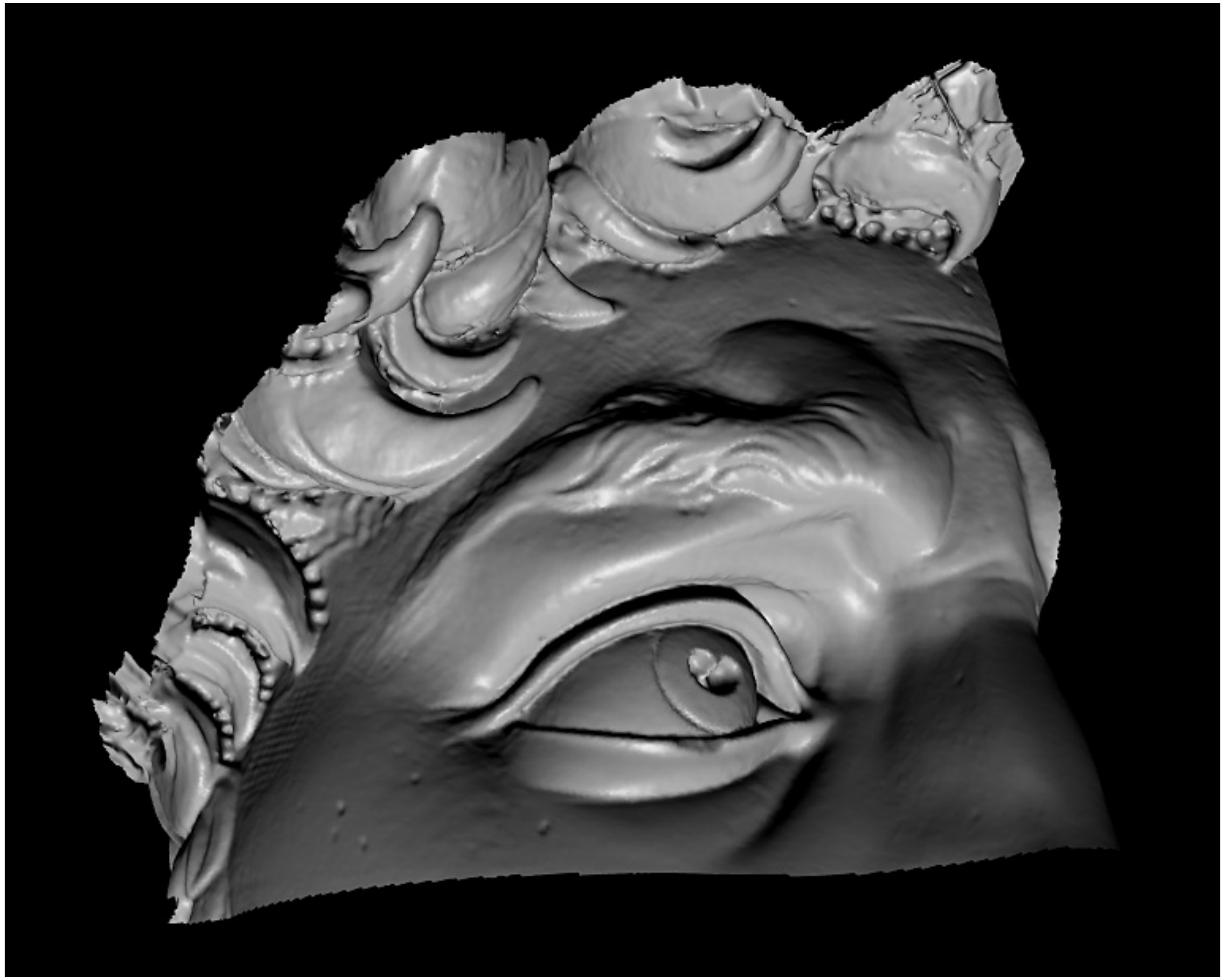
Digital Michelangelo Project
http://graphics.stanford.edu/projects/mich/

- 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

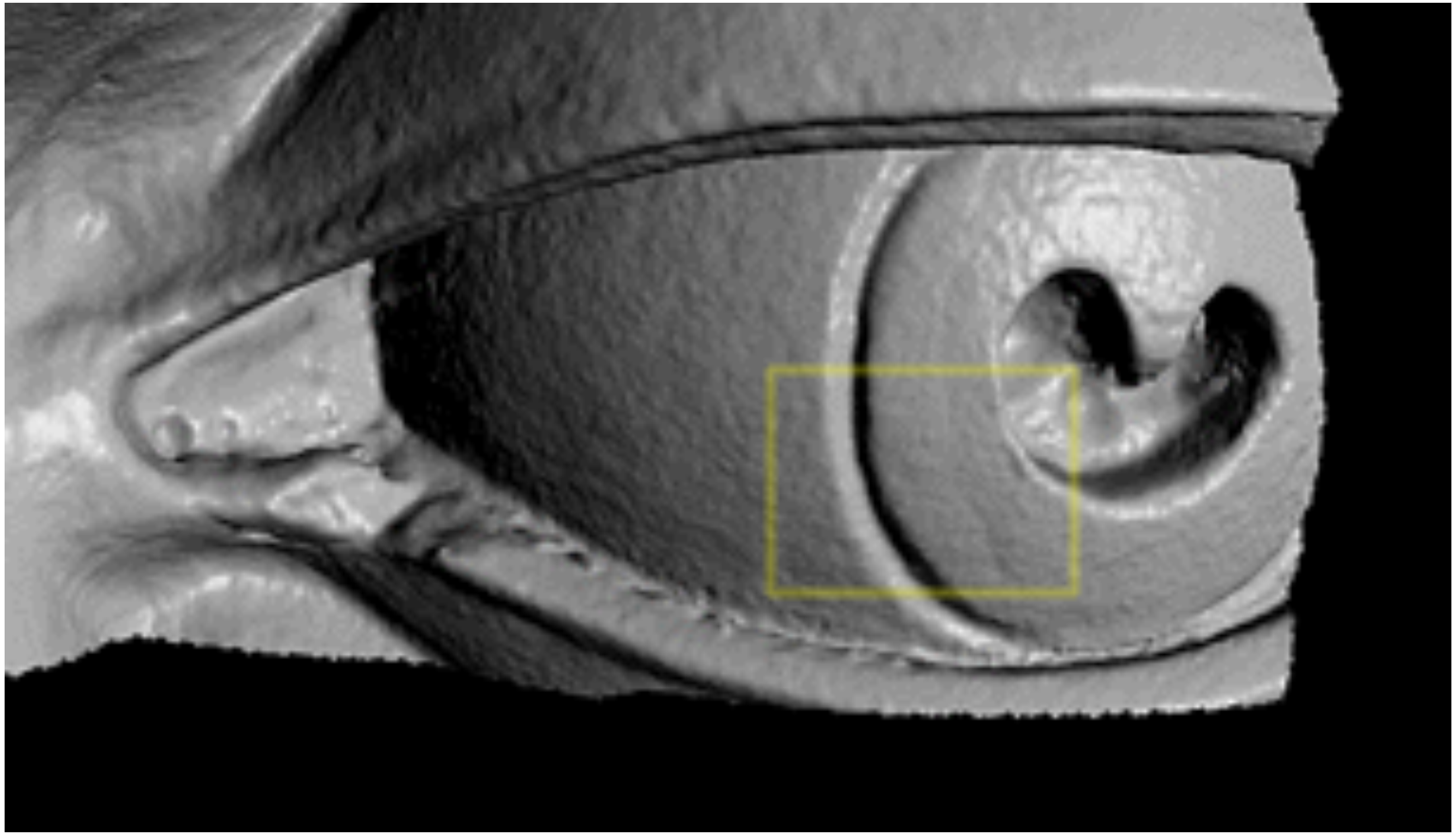
The Digital Michelangelo Project, Levoy et al.

Laser scanned models



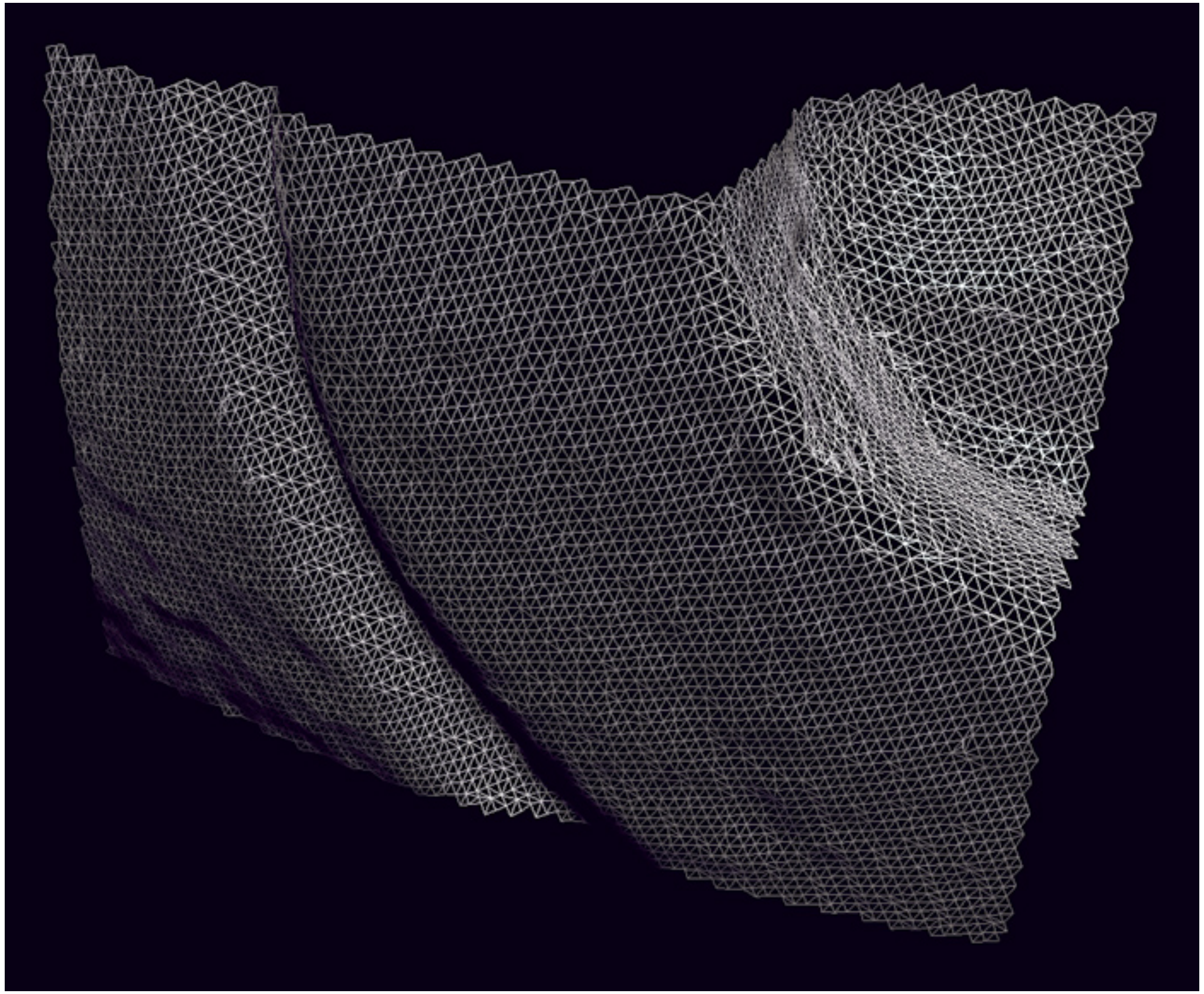
The Digital Michelangelo Project, Levoy et al.

Laser scanned models



The Digital Michelangelo Project, Levoy et al.

Laser scanned models



The Digital Michelangelo Project, Levoy et al.

Questions?