CSCI 497P/597P: Computer Vision

Scott Wehrwein

Stereo

Reading

- Szeliski: Chapter 11.3

Goals

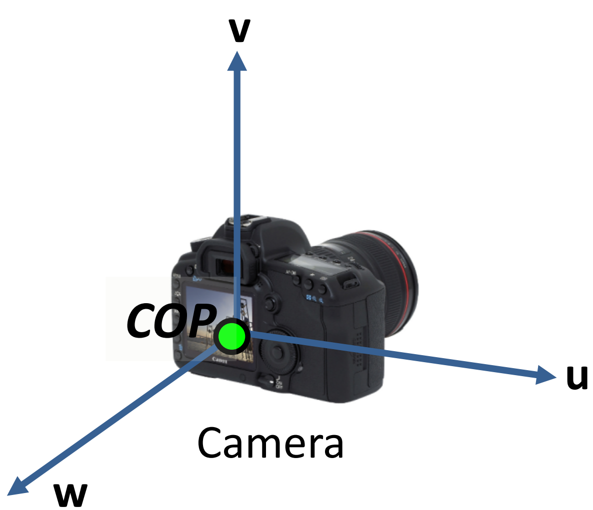
- Understand why stereo matching is the hard part of stereo vision.
- Understand the distinction between local and global methods for stereo correspondence.
- Understand the basic metrics used to compare patches (SSD, SAD, NCC)
- Understand the “energy-based” formulation of some common global stereo matching approaches.

Announcements

- Exam is Wednesday
  - One double-sided sheet of notes
  - Study guide is available as a Page on Canvas
  - The handwritten notes linked on the course webpage may be ugly, but they are usually more pertinent than the slides when available.

Camera(s) without a common COP

- With panoramas, we always assumed a common COP.
- How can we model the geometry of a camera in a separate world coordinate system?



Camera

"The World"

Two important coordinate systems:

1. *World* coordinate system
2. *Camera* coordinate system

How do we project a given point (x, y, z) in world coordinates?

Projection matrix: Putting it all together

Pixel coordinates:
y==i
x==j

"Camera" Coordinates: camera at origin looking down negative z, y points up

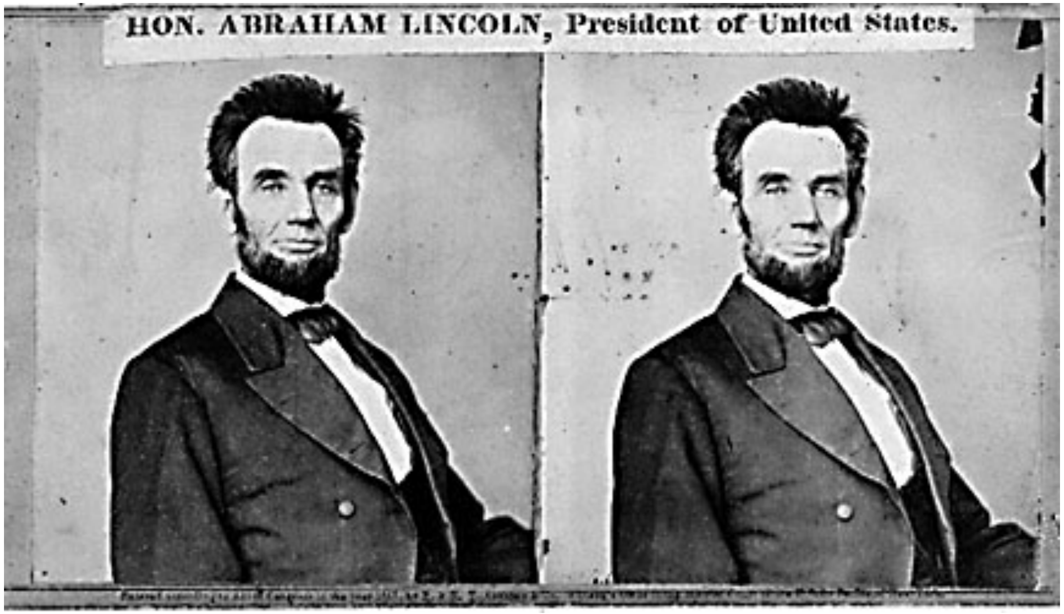
"World" Coordinates

$$\mathbf{\Pi} = \underbrace{\mathbf{K}}_{\text{intrinsics}} \underbrace{\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{projection}} \underbrace{\begin{bmatrix} \mathbf{R} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ 0 \; 0 \; 0 & 1 \end{bmatrix}}_{\text{rotation}} \underbrace{\begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{c} \\ 0 \; 0 \; 0 & 1 \end{bmatrix}}_{\text{translation}}}$$

The **K** matrix converts 3D rays in the camera's coordinate system to 2D image points in image (pixel) coordinates.

This part converts 3D points in world coordinates to 3D rays in the camera's coordinate system. There are 6 parameters represented (3 for position/translation, 3 for rotation).

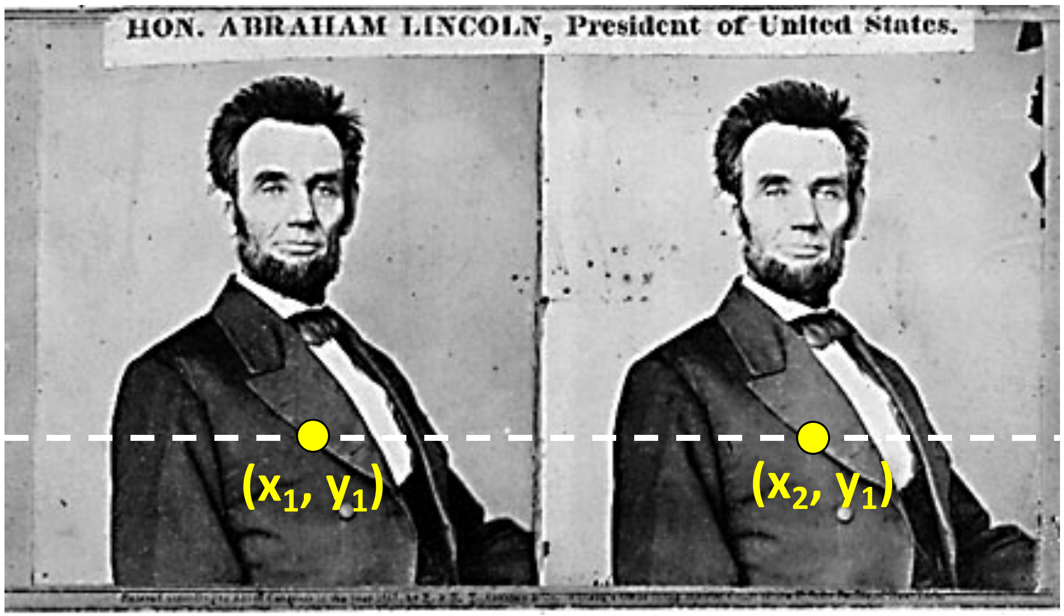
Stereo



- 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

Disparity

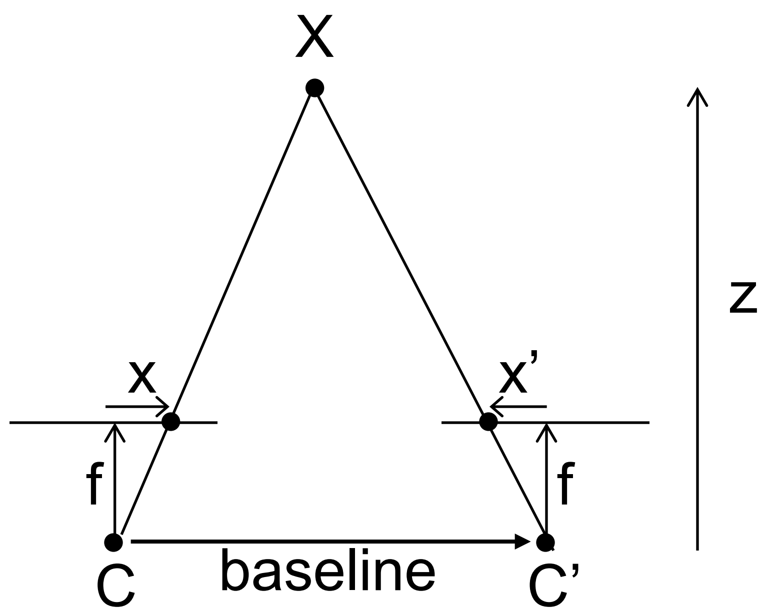
epipolar lines



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

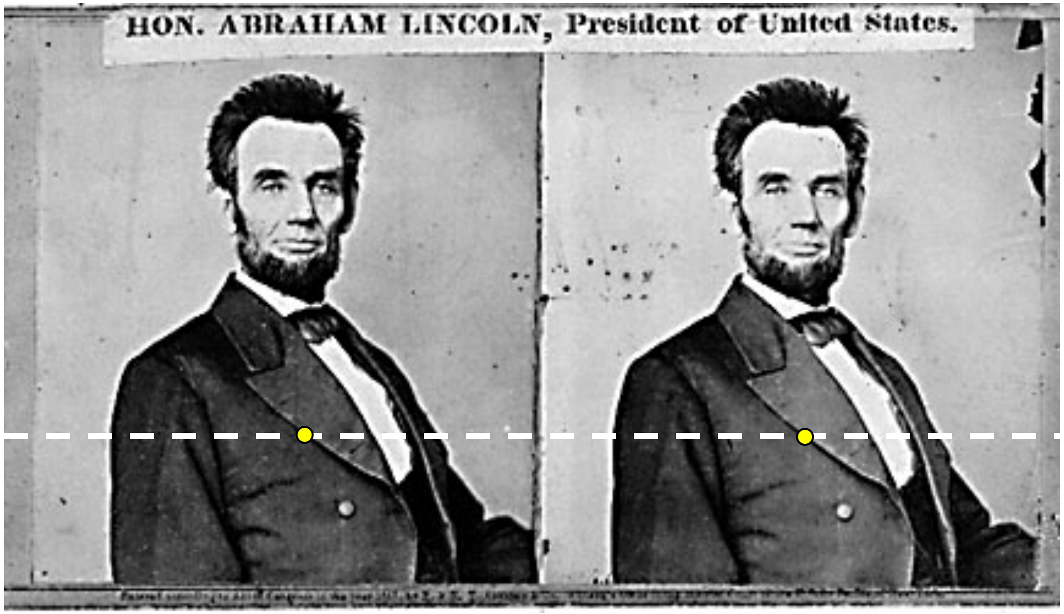
$x_2 - x_1$ = the *disparity* of pixel (x_1, y_1)

Depth from disparity



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

Your basic stereo algorithm



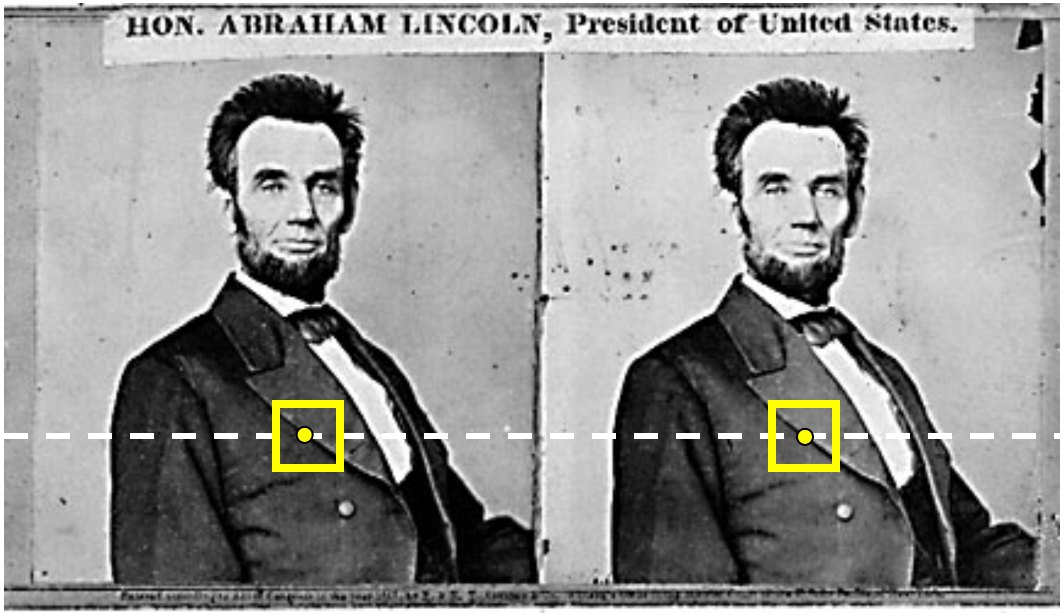
For each row of pixels:

For each pixel in the left row

- Find the matching pixel in the same row of the right image.
- disparity = pixel position – match position
- depth = f*b/disparity

This is the hard part.

Your basic (local) stereo algorithm



For each row of pixels:

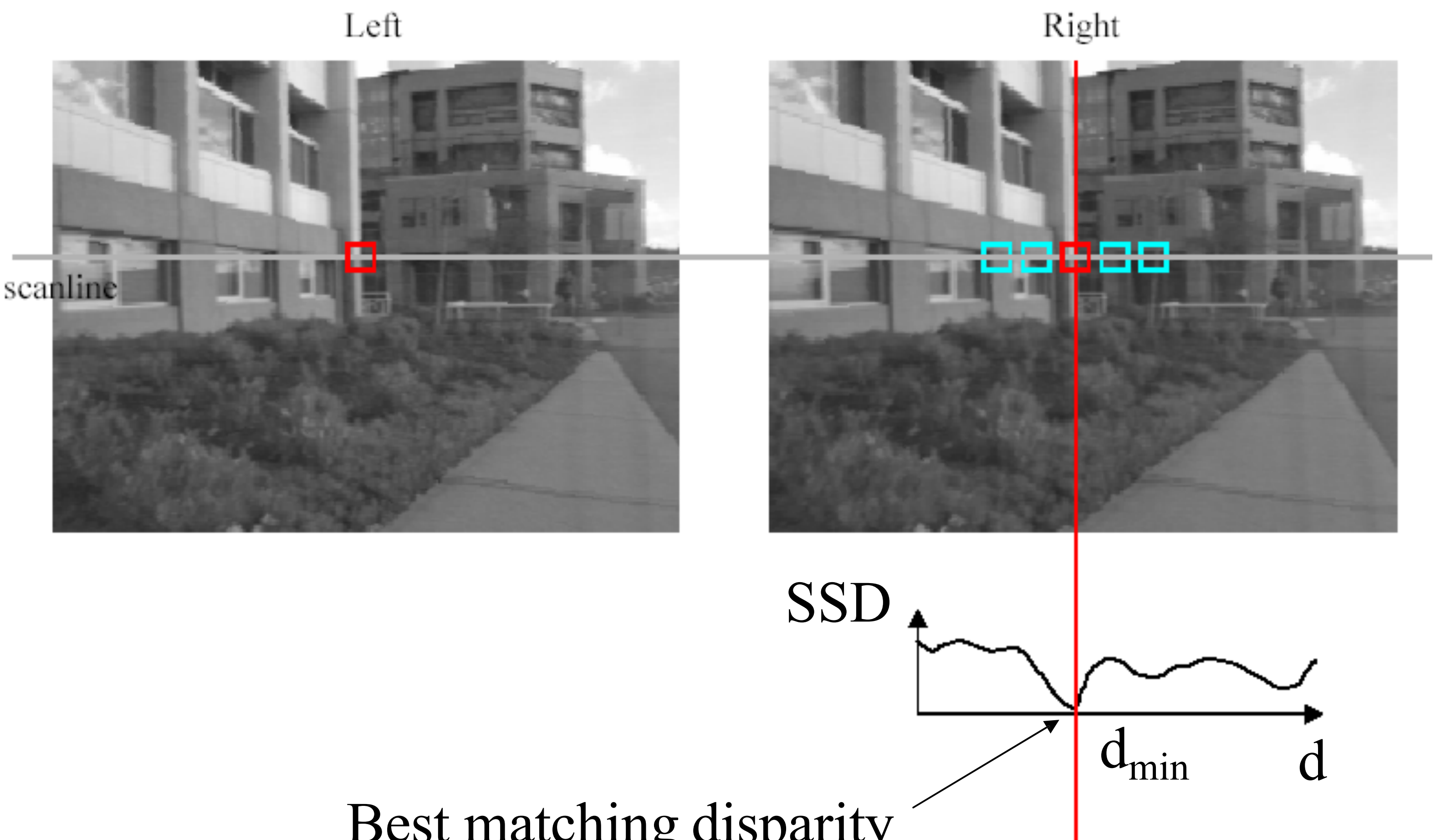
For each pixel in the left row

- Find the matching pixel in the same row of the right image.
- disparity = pixel position – match position
- depth = f*b/disparity

Improvement: match ***windows***

Stereo matching based on SSD

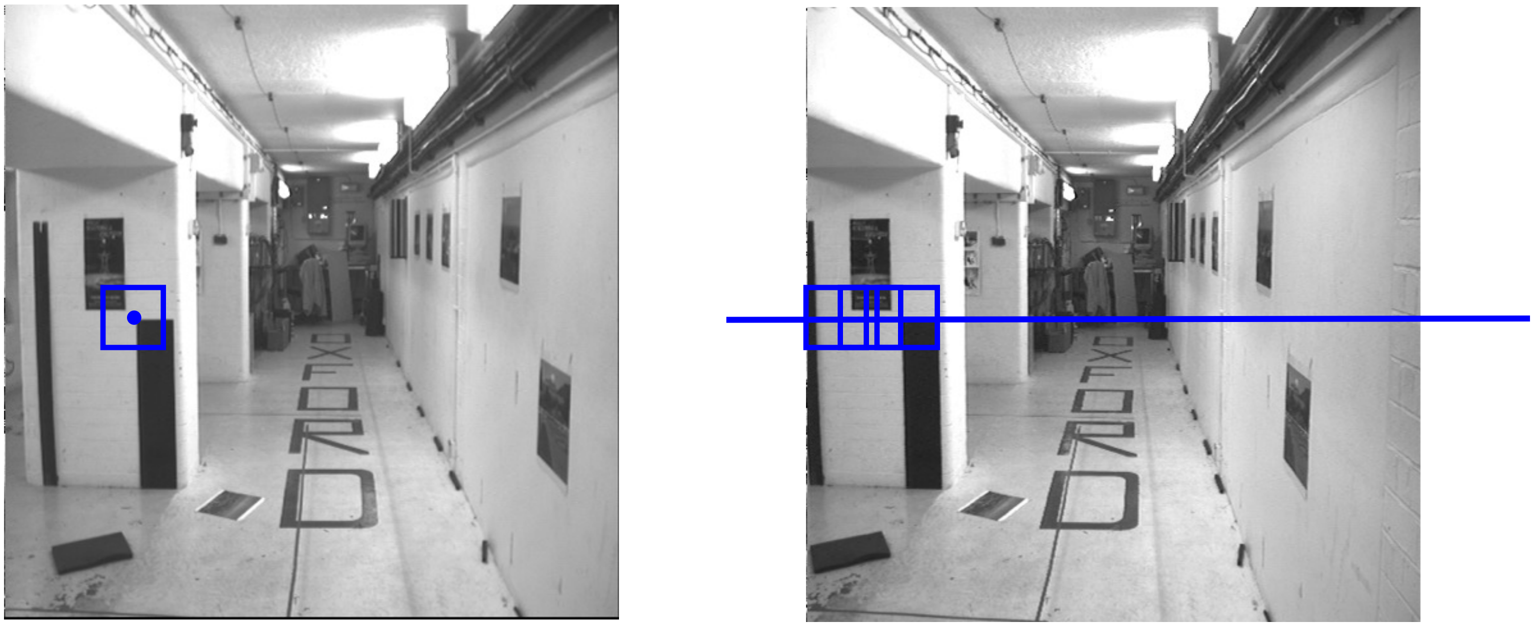
As in Harris detector, but instead of shifted patches, compare patches across stereo pair.



Metrics for correspondence

- SSD = sum of squared differences
- SAD = sum of absolute differences
- NCC = normalized cross-correlation
  - (more ~~convolution~~cross correlation!)

Normalized Cross-Correlation



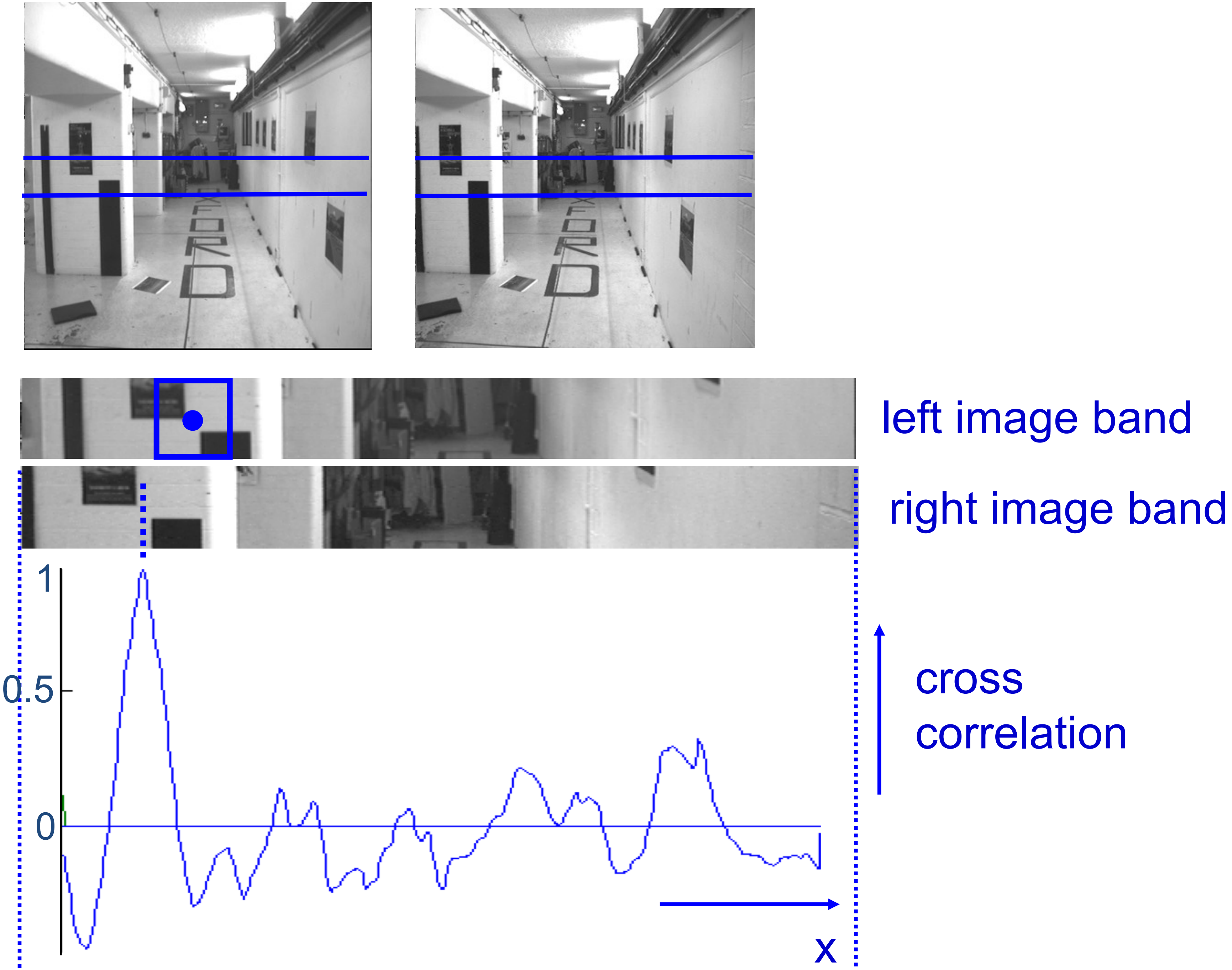
regions $\mathtt{A}, \mathtt{B}$, write as vectors $\mathbf{a}, \mathbf{b}$

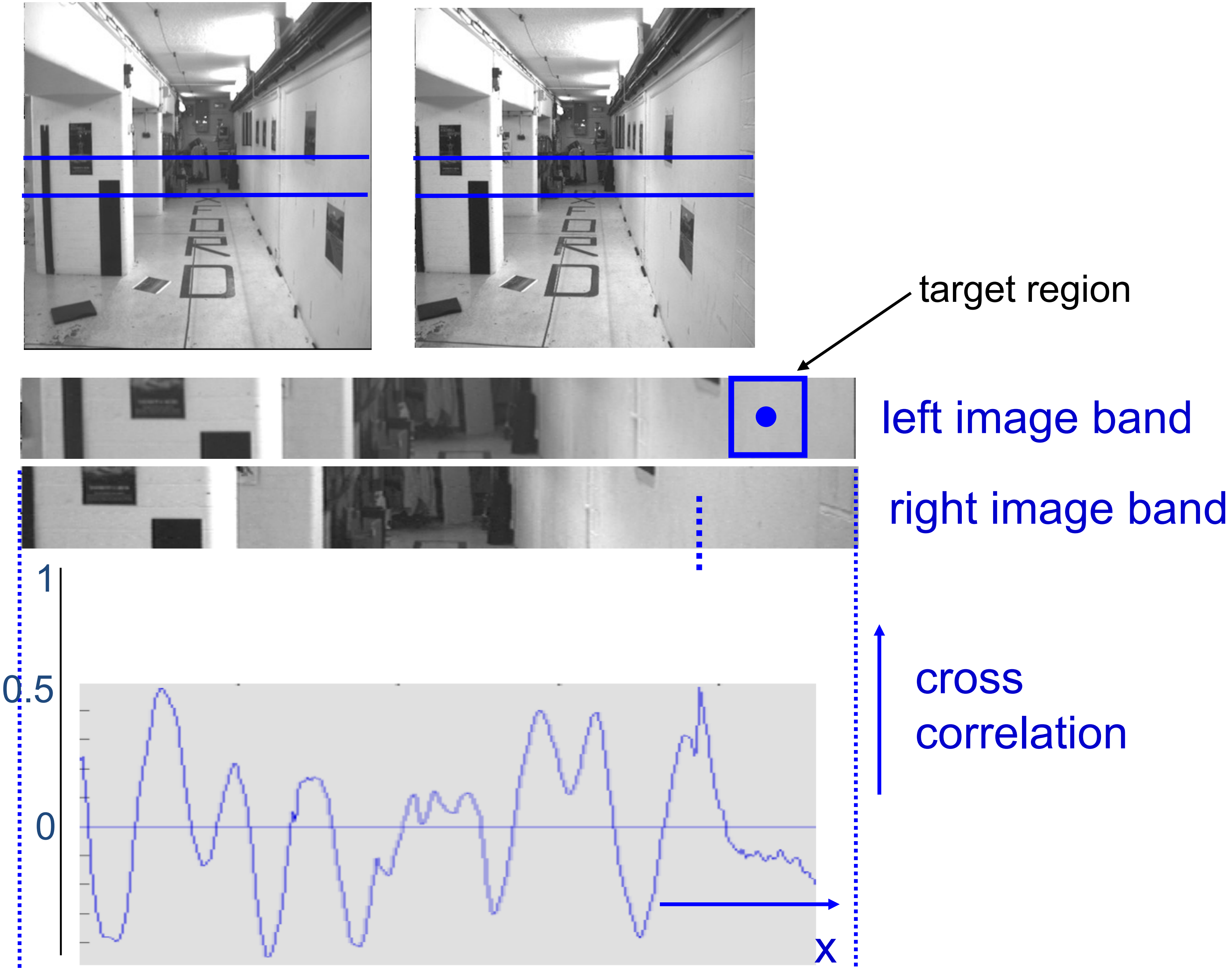
translate so that mean is zero

$$\mathtt{a} \rightarrow \mathtt{a} - \langle \mathbf{a} \rangle, \quad \mathtt{b} \rightarrow \mathtt{b} - \langle \mathbf{b} \rangle$$

$$\text{cross correlation} = \frac{\mathbf{a}.\mathbf{b}}{|\mathbf{a}||\mathbf{b}|}$$

Invariant to $I \rightarrow \alpha I + \beta$

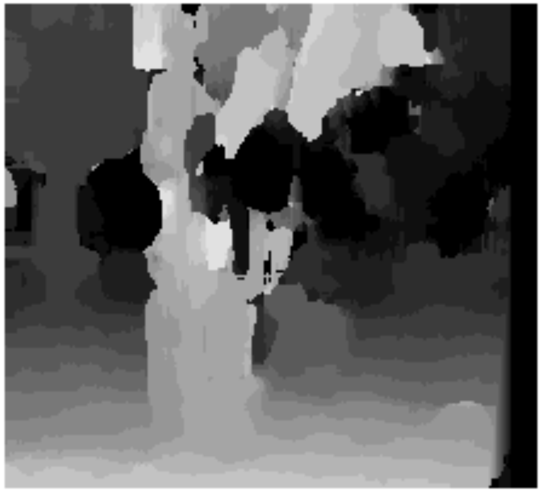

left image band
right image band
1
0.5
0
cross
correlation
x



target region
left image band
right image band
1
0.5
0
cross
correlation
x

Window size

W = 3



W = 20

Effect of window size

- Smaller window
  - \+ More detail
  - • More noise
- Larger window
  - \+ Less noise
  - • Less detail

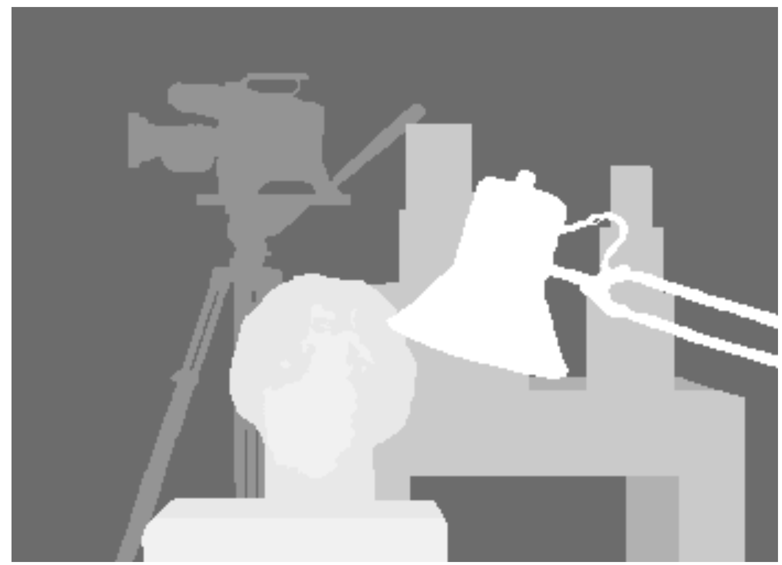
Better results with *adaptive window*

- 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

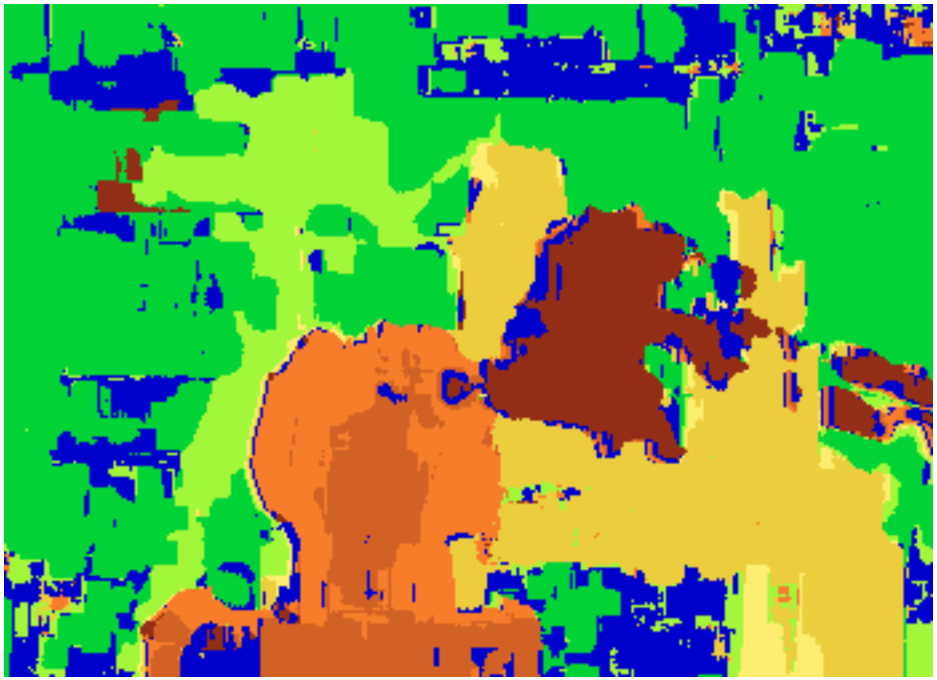
– Data from University of Tsukuba
– Similar results on other images without ground truth

Scene



Ground truth

Results with window search



Window-based matching
(best window size)

Ground truth

Better methods exist...

Fancier method

Ground truth

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

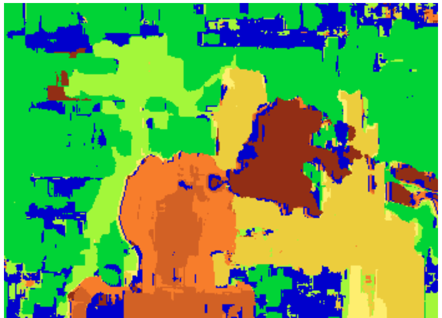
For the latest and greatest: http://www.middlebury.edu/stereo/

Local vs Global methods

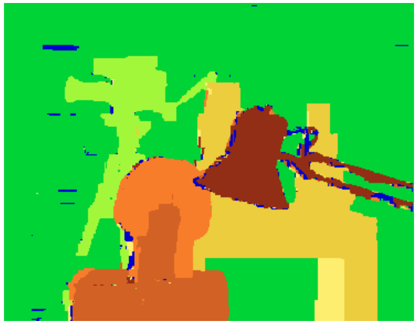
– So far, pixel x's disparity is chosen independent of its neighbors: **local** methods for stereo correspondence.

– Better results are possible using **global** methods that consider constraints like:
  - Nearby pixels should (usually?) have similar disparity
  - Points should not appear "out of order"

Window-based matching:



Fancier method:



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

Better methods exist...

Fancier 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/

The Cost Volume

1. For every pixel (x, y)
   1. For every disparity d
      1. Get patch from image 1 at (x, y)
      2. Get patch from image 2 at (x + d, y)
      3. Compute cost using your metric of choice

```
C = np.array(h,w,d)
for r in range(0,h):
  for c in range(0,w):
    for d in range(-maxd, maxd):
      C[r,c,d] = metric(get_patch(im1,r,c), get_patch(im2,r,c+d)

disp = np.max(C, axis=2)
depth = f * b / disp
```

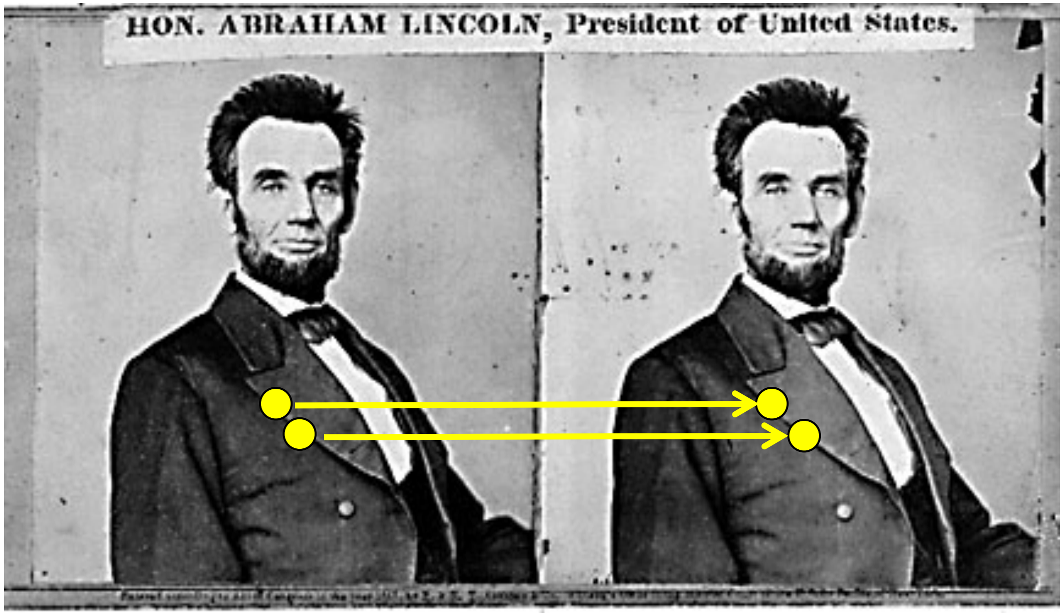
The Cost Volume

$I(x, y)$

$J(x, y)$

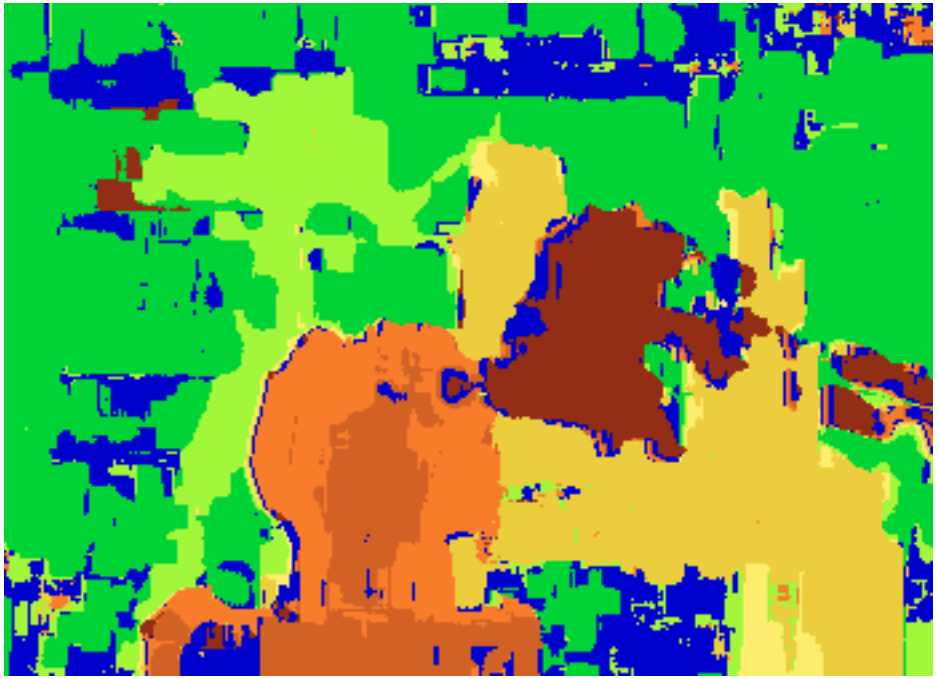
$C(x, y, d)$; the *disparity space image* (DSI)

Global methods: 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

Greedy selection of best match



```
disp = np.max(C, axis=2)
```

Global Methods: Stereo as energy minimization

- Better objective function

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

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

smoothness cost ← Adjacent pixels should (usually) move about the same amount

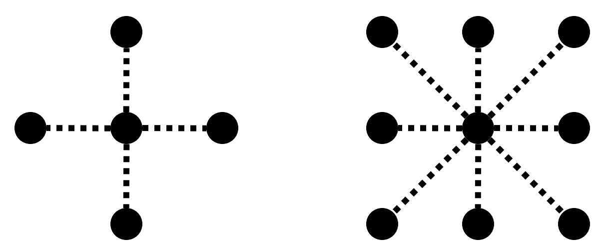
Stereo as energy minimization

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

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

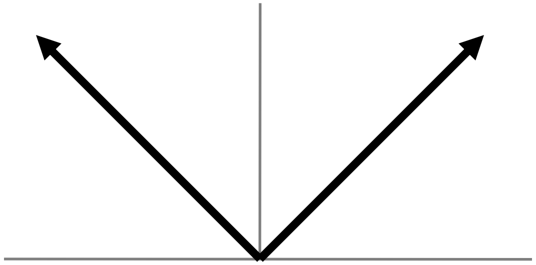
Smoothness cost

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

How do we choose *V*?

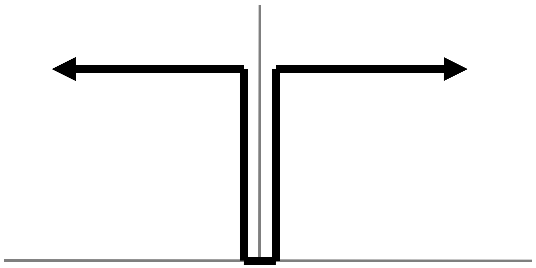
$$V(d_p, d_q) = |d_p - d_q|$$

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"



Global Methods: Minimizing the Energy

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

- For a 1D scanline, E can be minimized exactly using dynamic programming.
  - Basic idea: incrementally build a table of costs one column at a time

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

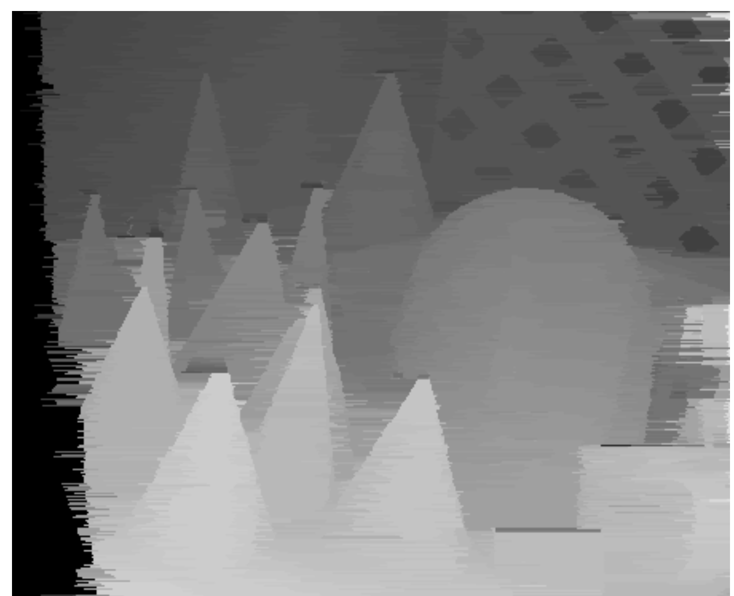
Base case: $D(0, y, i) = C(0, y, i), i = 0, \ldots, L$ (L = max disparity)

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



Global Methods: Minimizing the Energy

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

- For a 2D image, finding the global minimum of E is NP-hard.
- Many local minima so gradient descent doesn't work well.
- Good approximations exist:
  - Map energy onto a Markov Random Field
  - Minimize using graph cuts or belief propagation.

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)

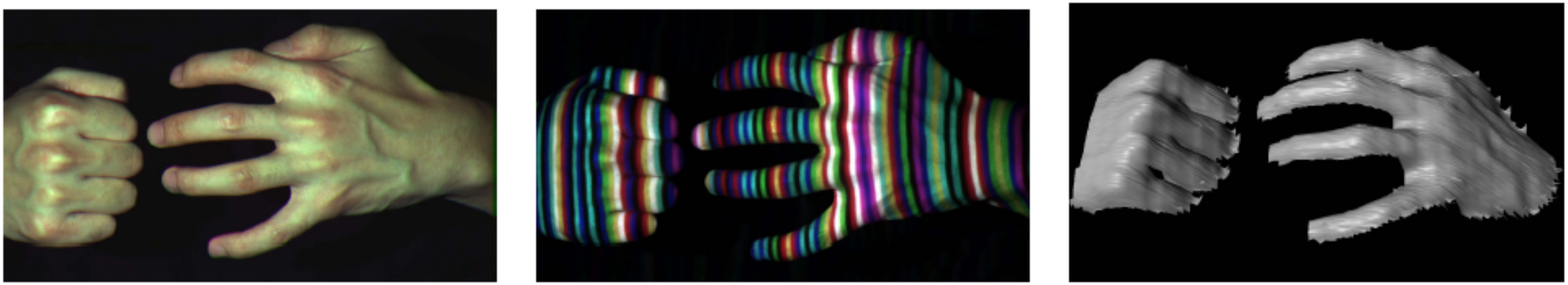
Stereo reconstruction pipeline

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

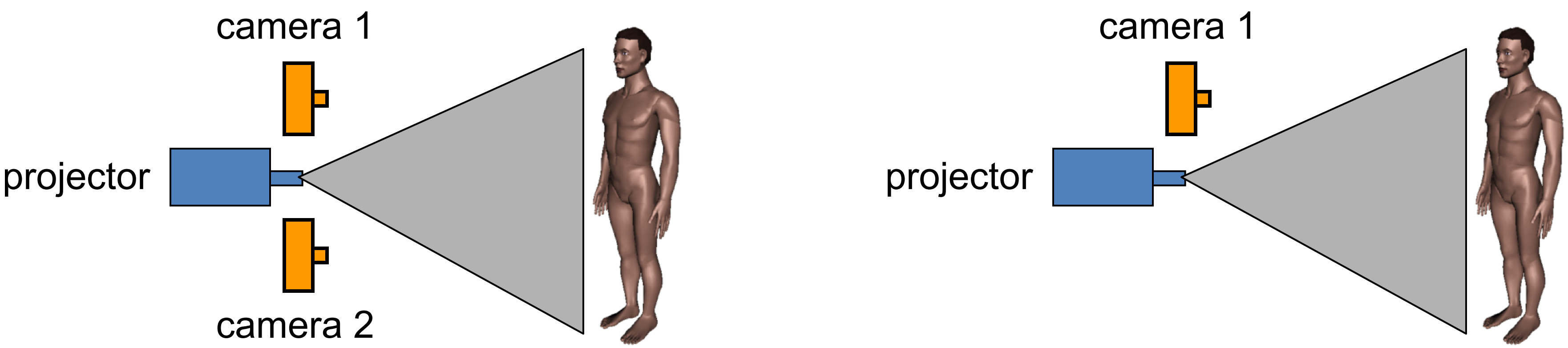
What will cause errors?

- 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

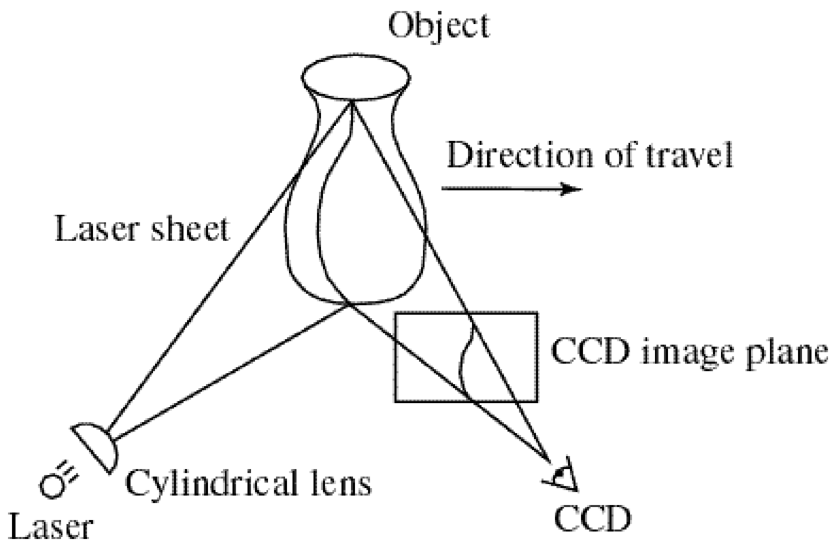


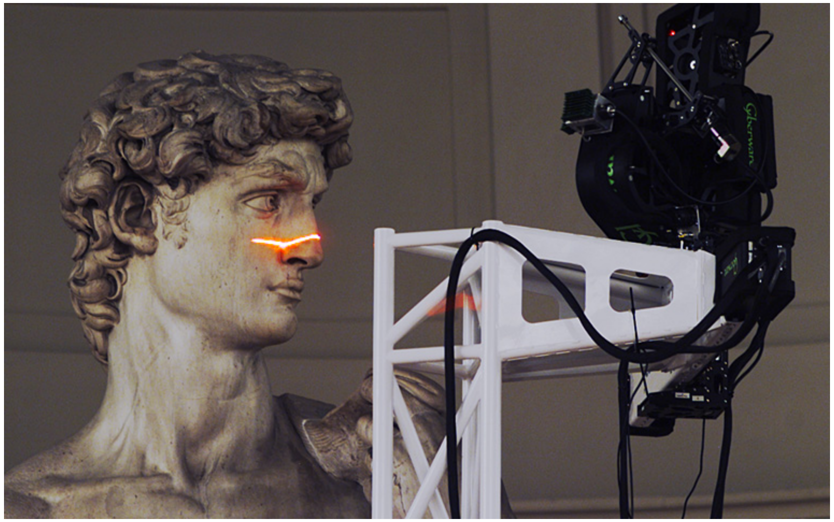
- Project "structured" light patterns onto the object
  - simplifies the correspondence problem
  - basis for active depth sensors, such as Kinect and iPhone X (using IR)

Active stereo with structured light

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

Laser scanning





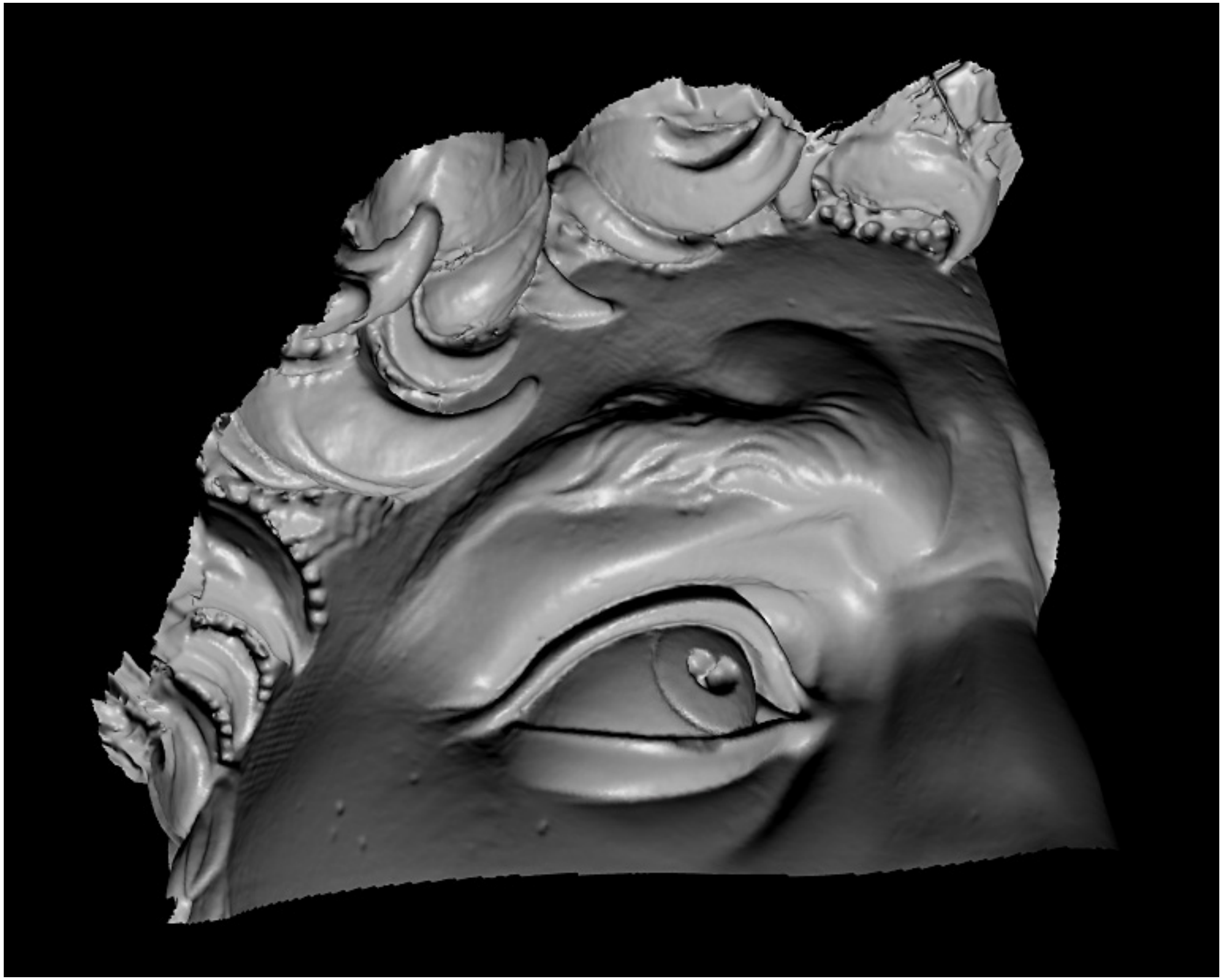
Digital Michelangelo Project
http://graphics.stanford.edu/projects/mich/

- 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

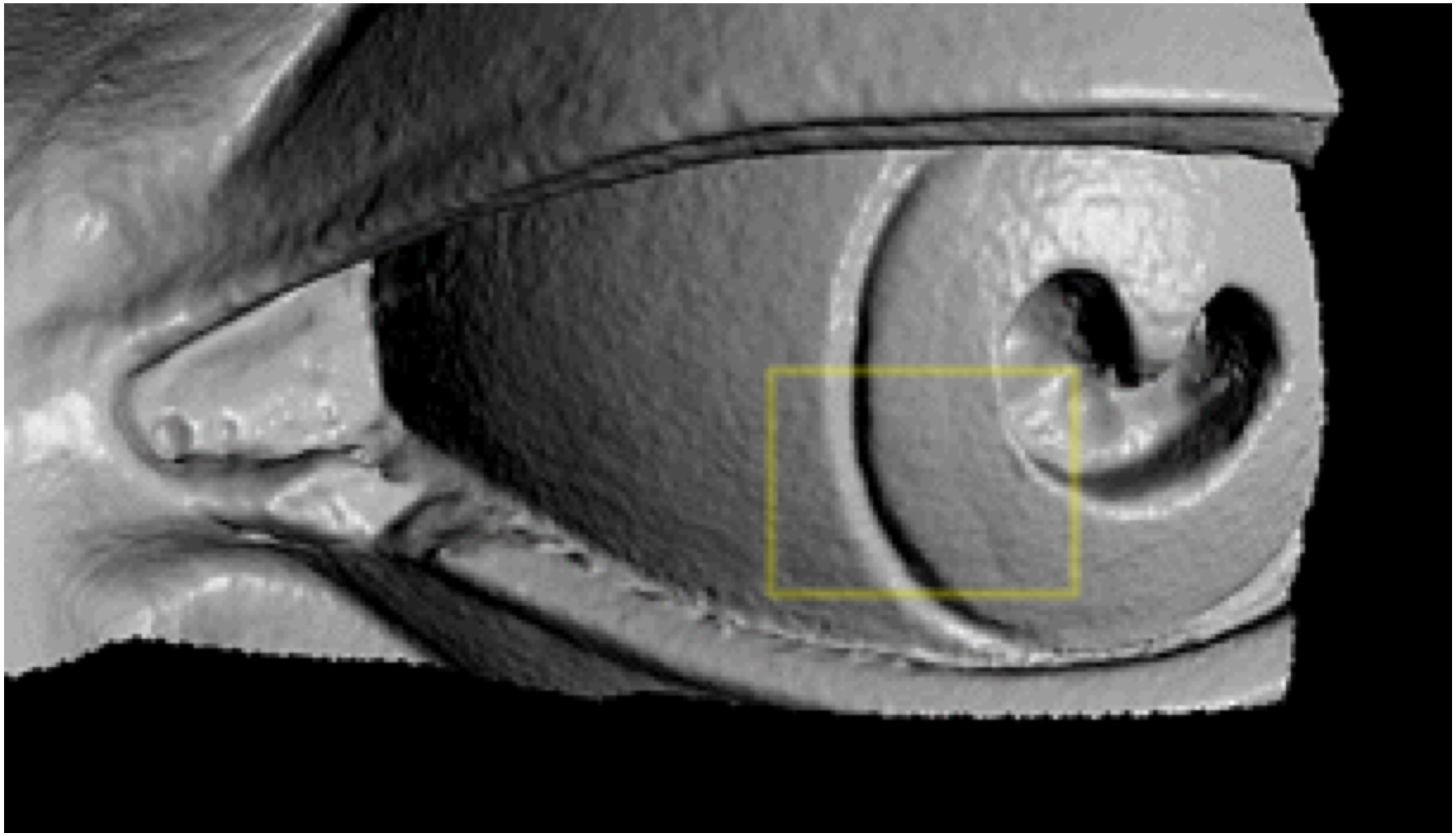
The Digital Michelangelo Project, Levoy et al.

Laser scanned models



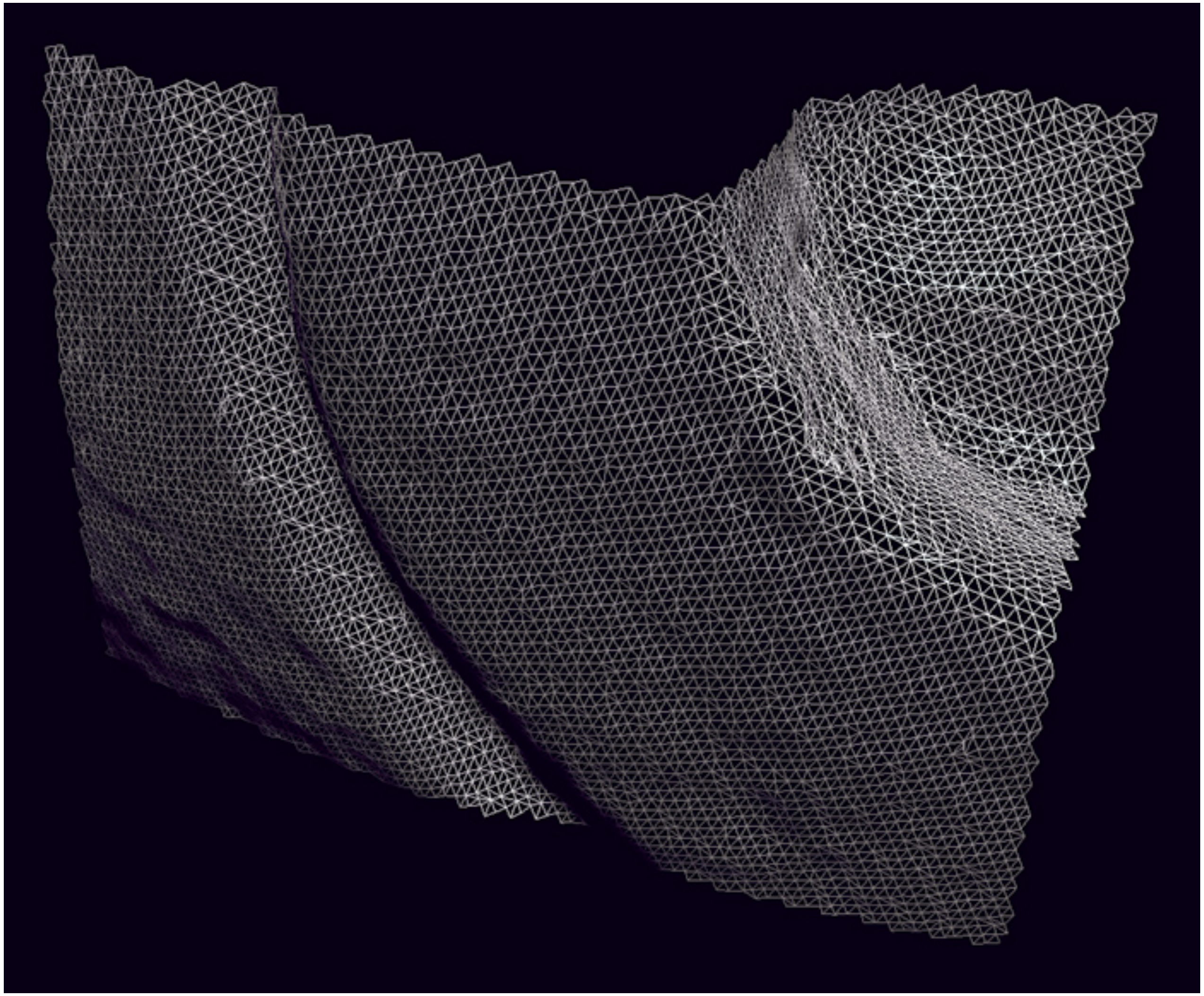
The Digital Michelangelo Project, Levoy et al.

Laser scanned models



The Digital Michelangelo Project, Levoy et al.

Laser scanned models



The Digital Michelangelo Project, Levoy et al.

Microsoft Kinect