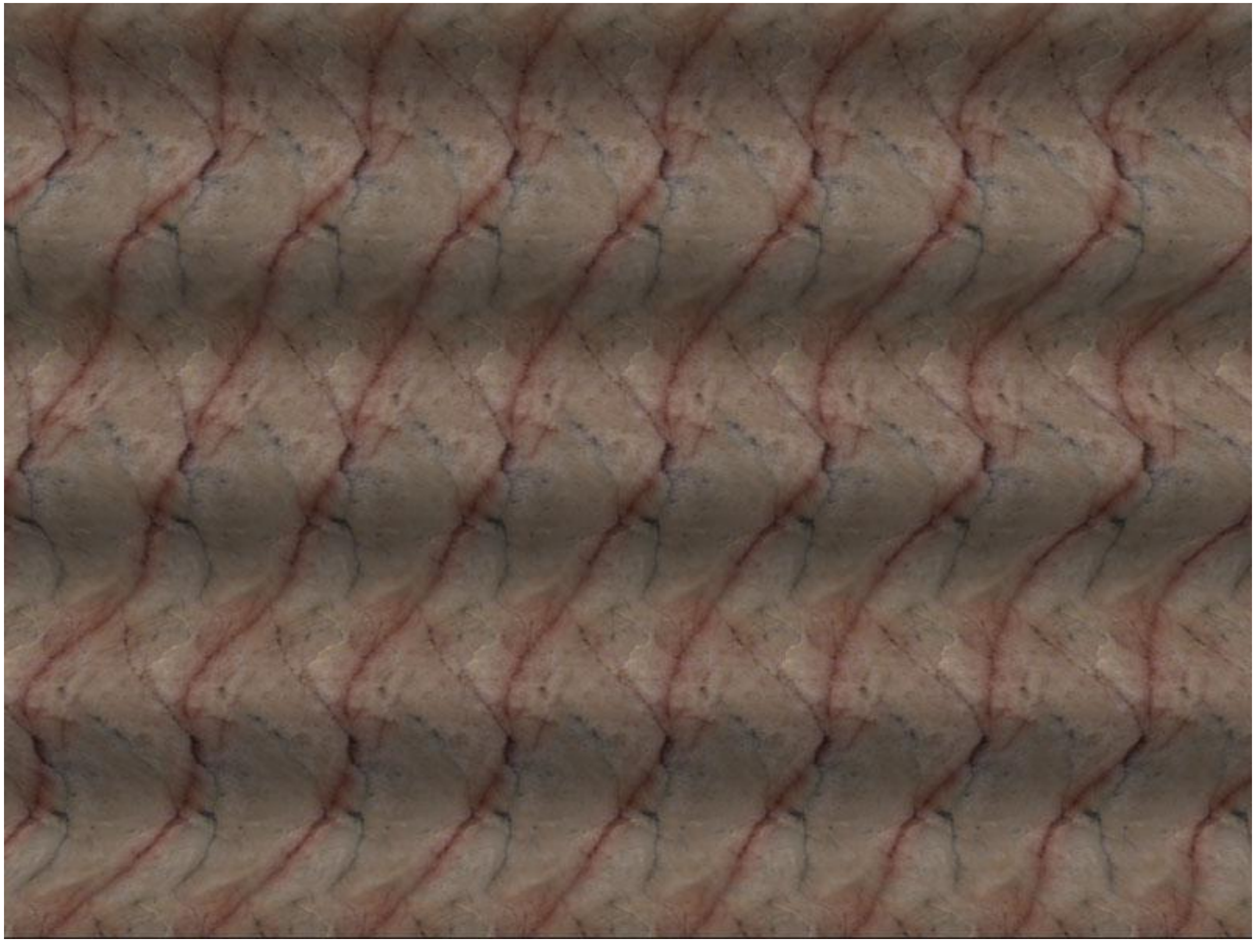
Stereo

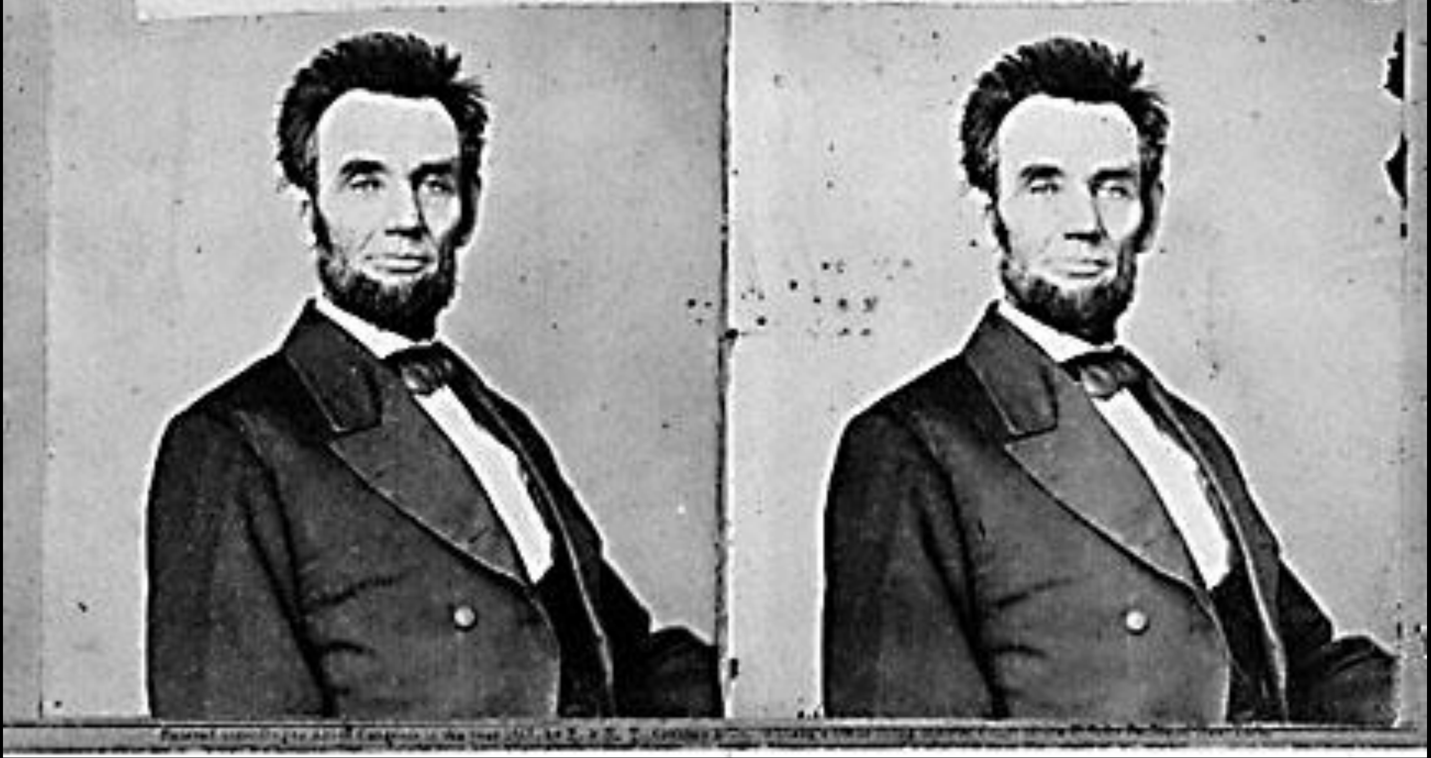


Single image stereogram, by [Niklas Een](#)

Readings

- Trucco & Verri, Chapter 7
  - Read through 7.1, 7.2.1, 7.2.2, 7.3.1, 7.3.2, 7.3.7 and 7.4, 7.4.1.

HON. ABRAHAM LINCOLN, President of United States.



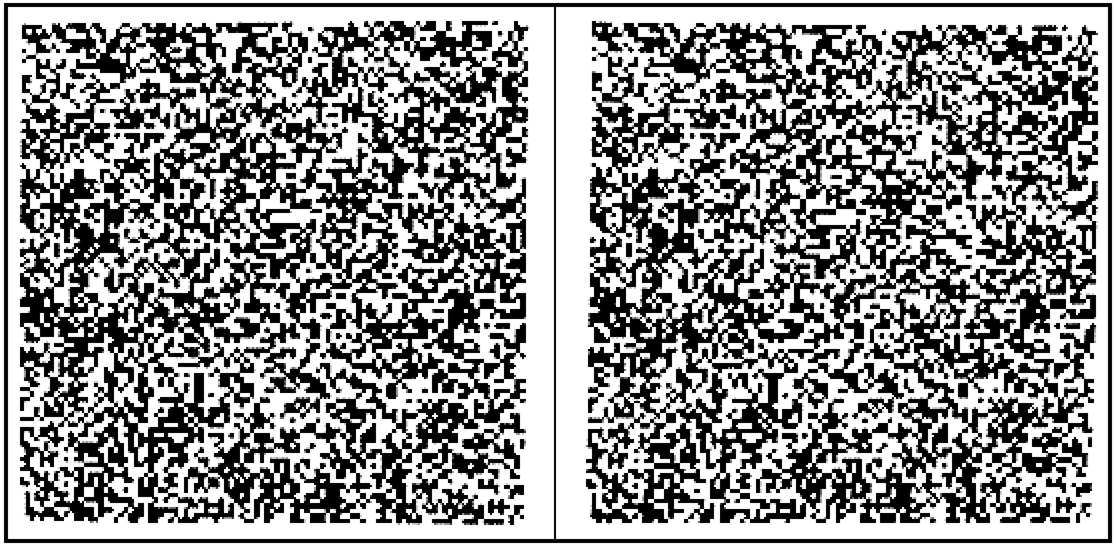
View-Master

present a different image to each eye

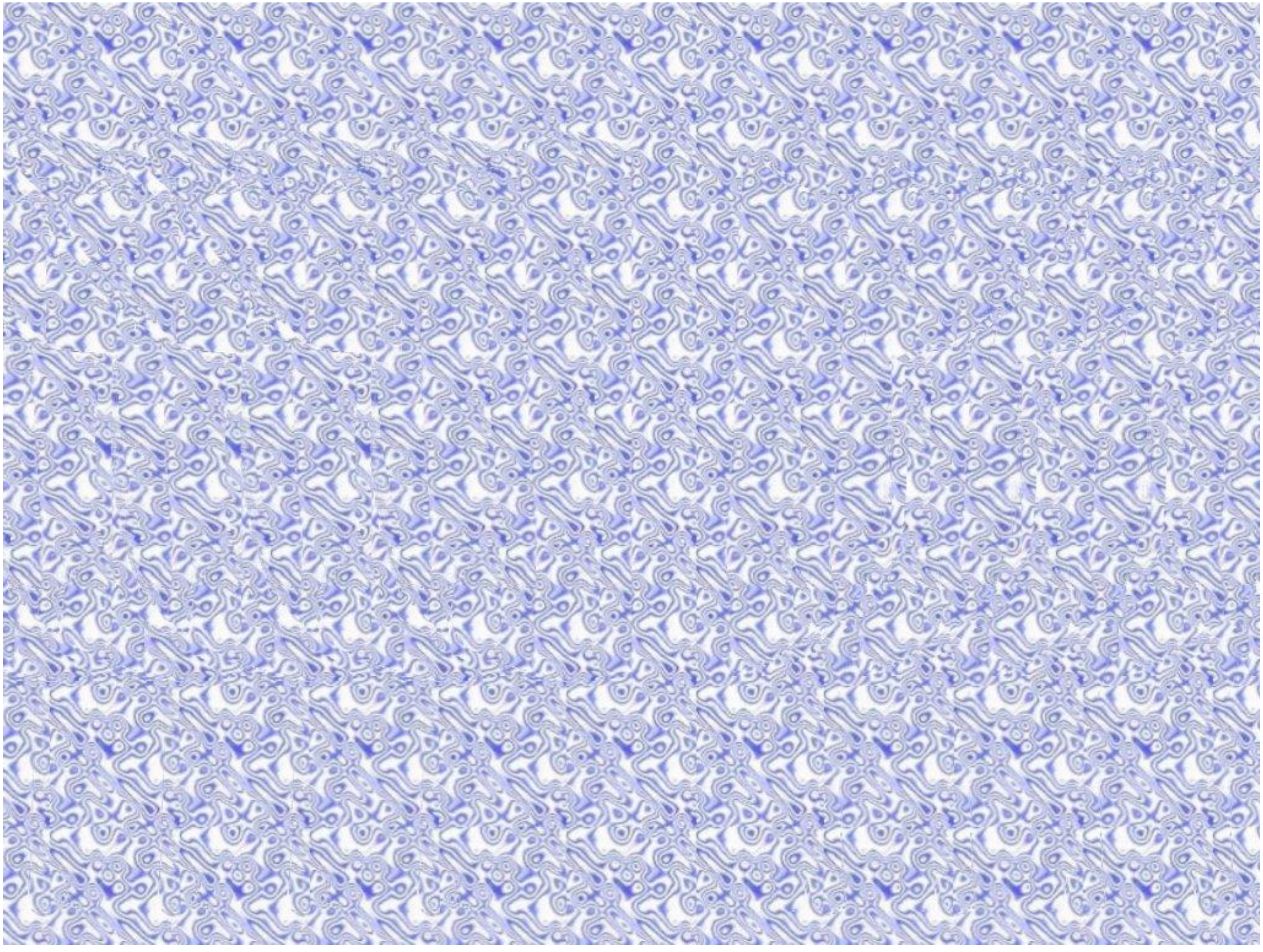
Public Library, Stereoscopic Looking Room, Chicago, by Phillips, 1923

Random Dot Stereograms

Stereo is powerful.

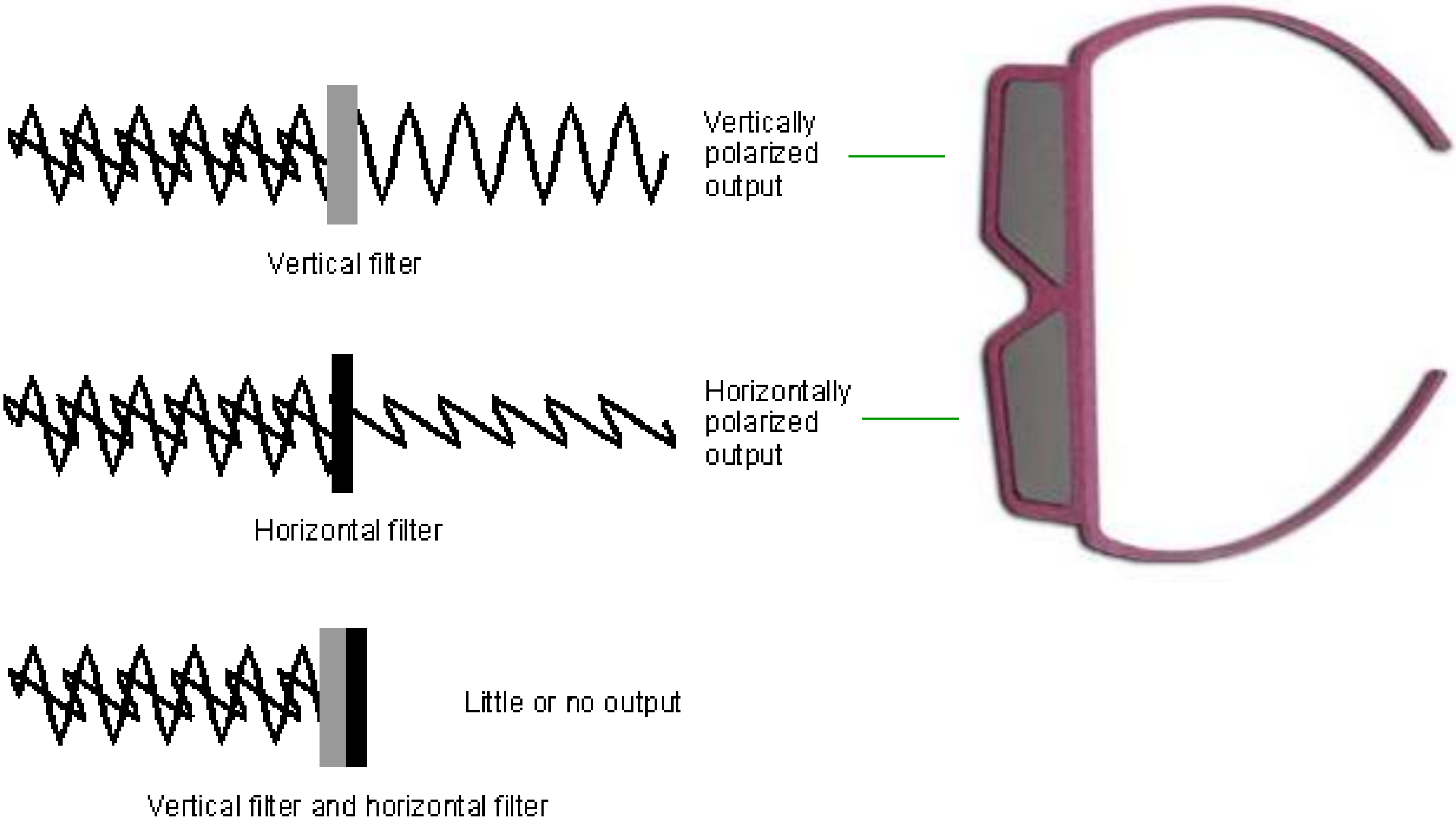


Autostereograms

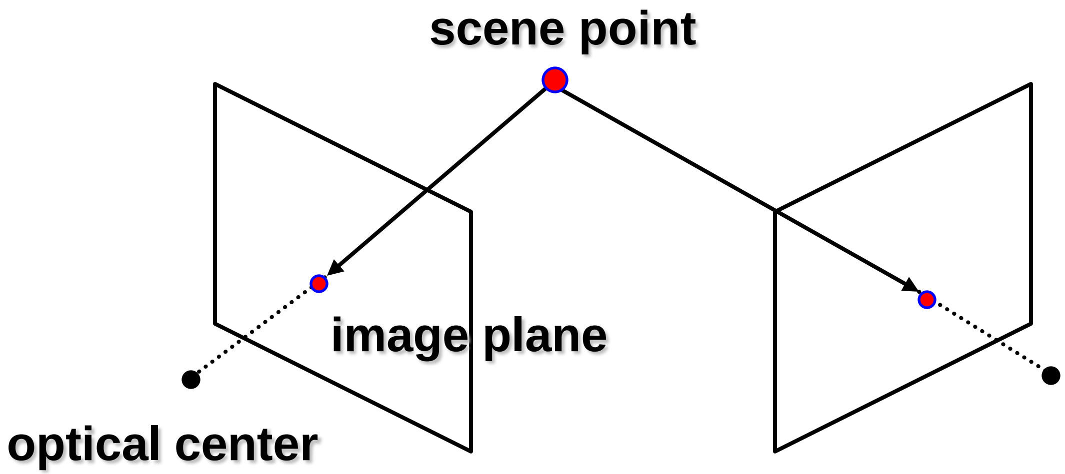


3D Movies

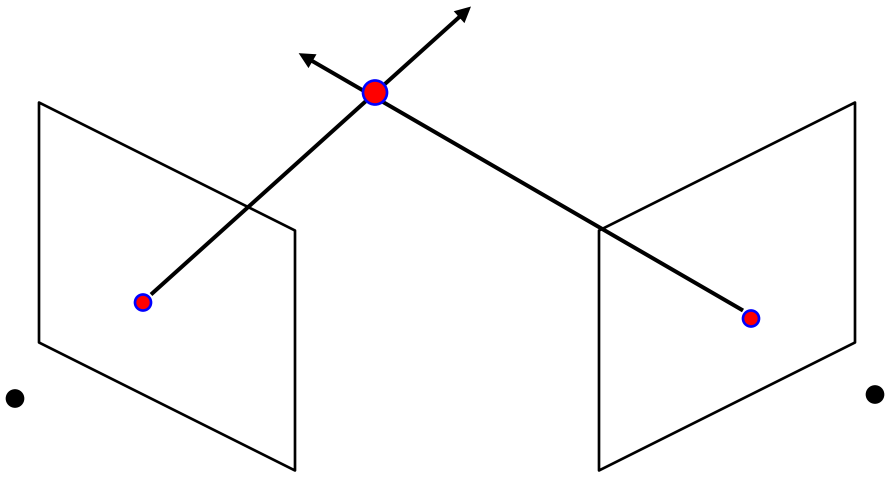
left/right frames differently polarized



Stereo



Stereo



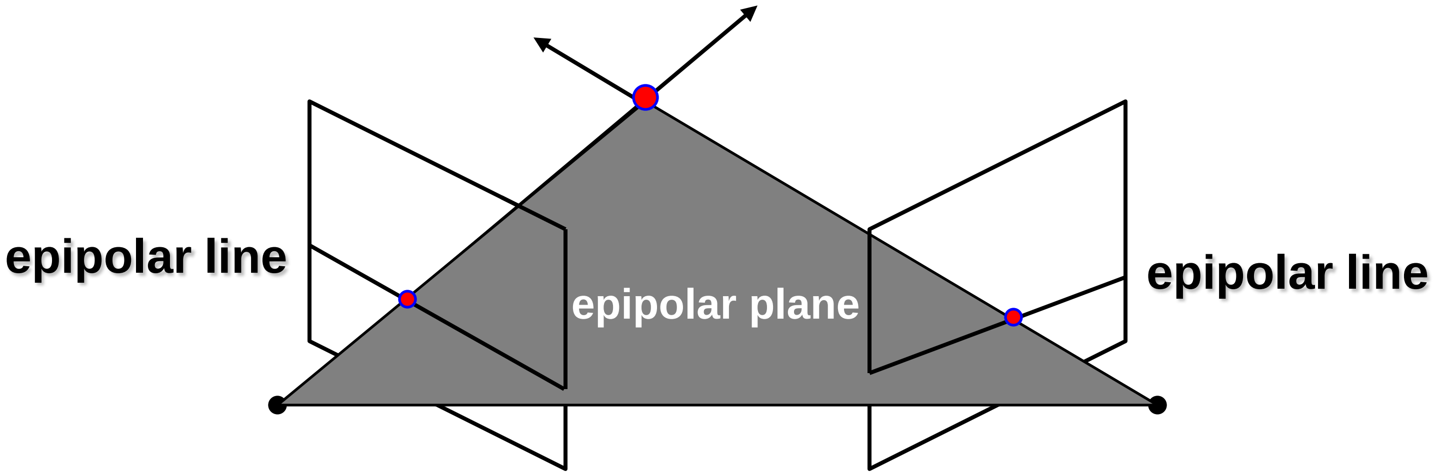
Basic Principle: Triangulation

- Gives reconstruction as intersection of two rays
- Requires
  - camera pose (calibration)
  - ***point correspondence***

Stereo correspondence

Determine Pixel Correspondence

- Pairs of points that correspond to same scene point



Epipolar Constraint

- Reduces correspondence problem to 1D search along *conjugate epipolar lines*
- Java demo: http://www.ai.sri.com/~luong/research/Meta3DViewer/EpipolarGeo.html
- An *epipole* is the projection of one camera center into the other image. (Usually called *e* and *e*'.)

Fundamental matrix

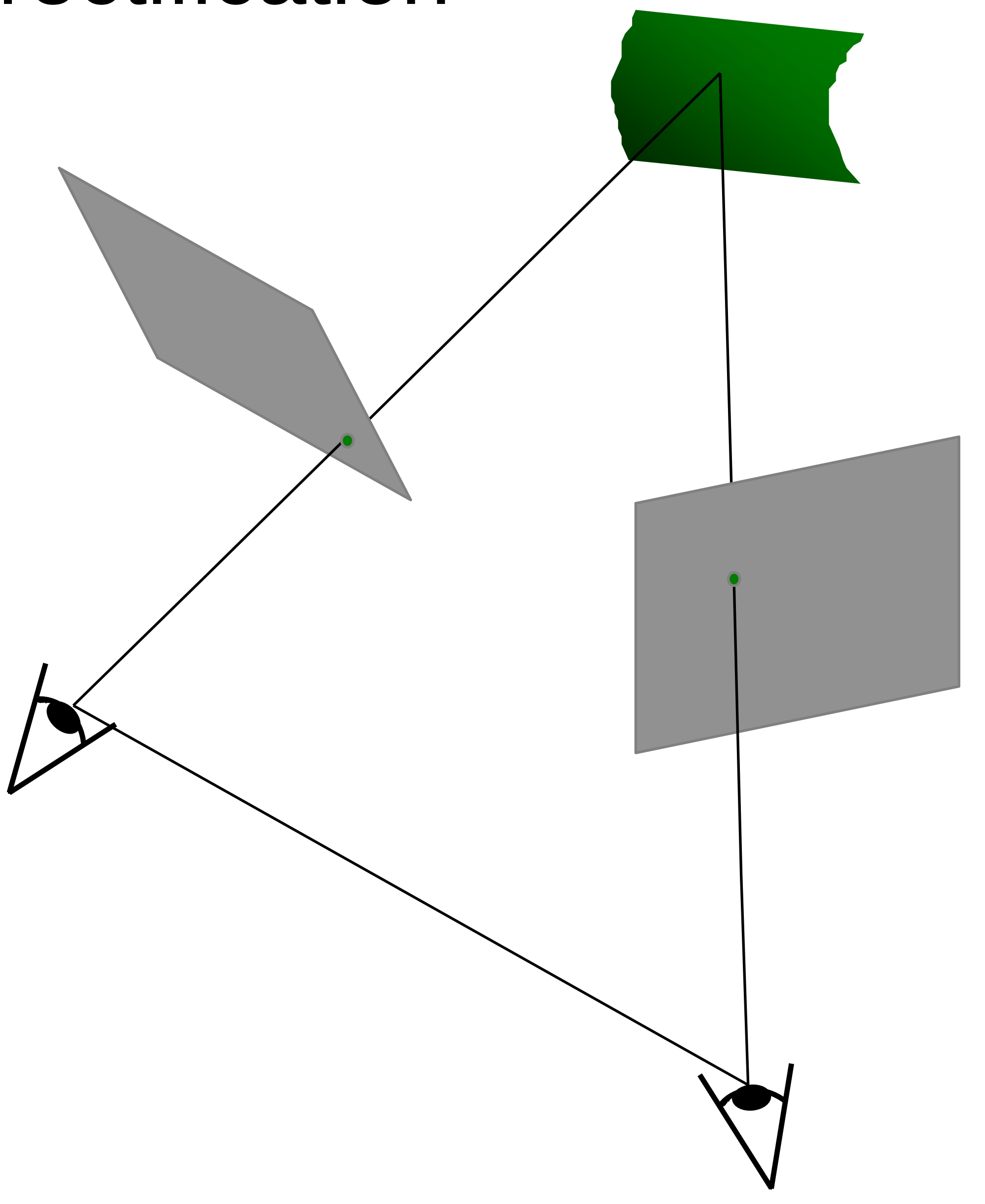
The fundamental matrix F captures the epipolar relationship.

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

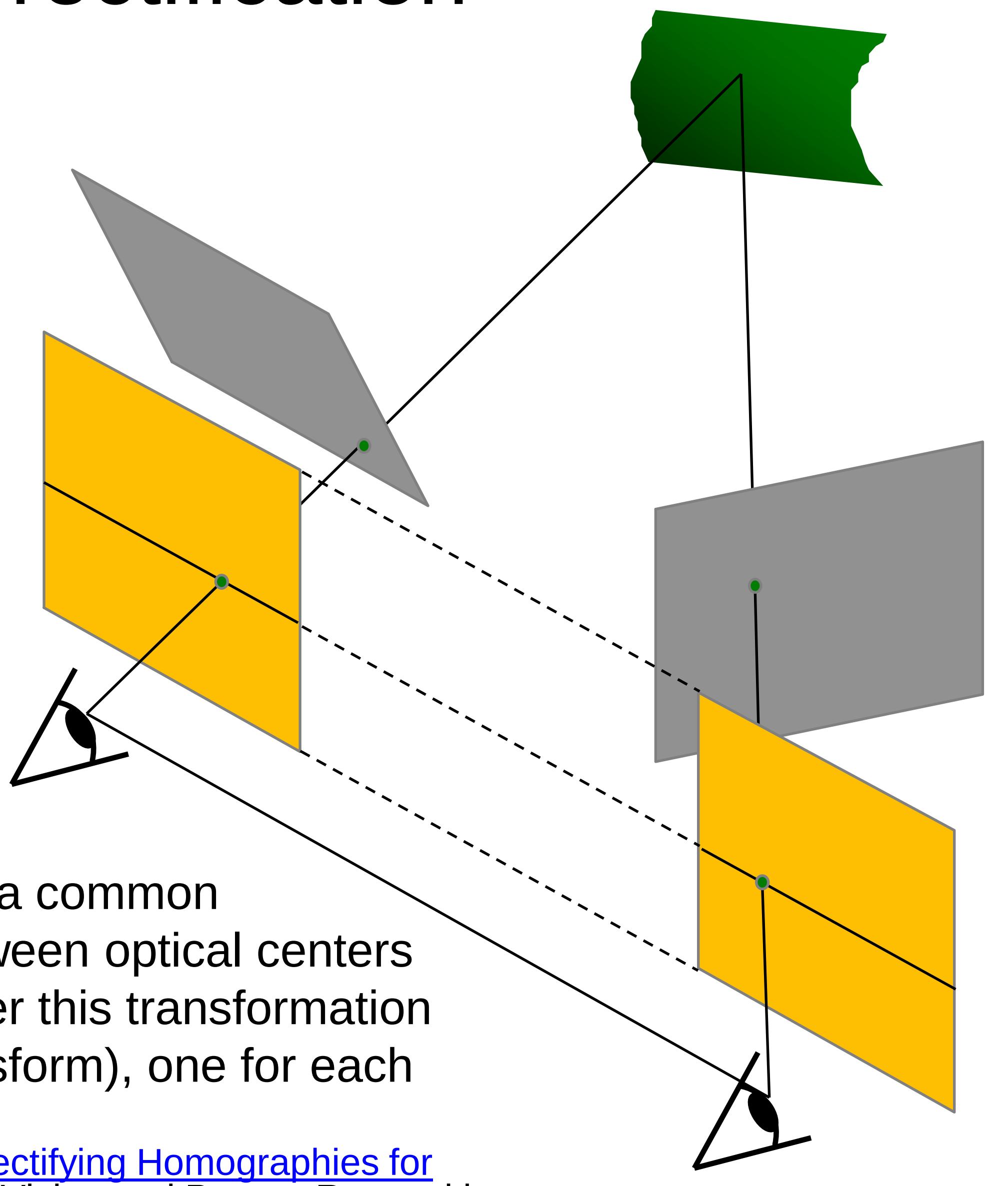
What line in the second image corresponds to point p?

What can we say about Fe or $F^T e'$?

Stereo image rectification



Stereo image rectification



- reproject image planes onto a common plane parallel to the line between optical 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.

Stereo image rectification

Recall the fundamental matrix:

$$p'^T F p = 0$$

What does it look like for rectified images?

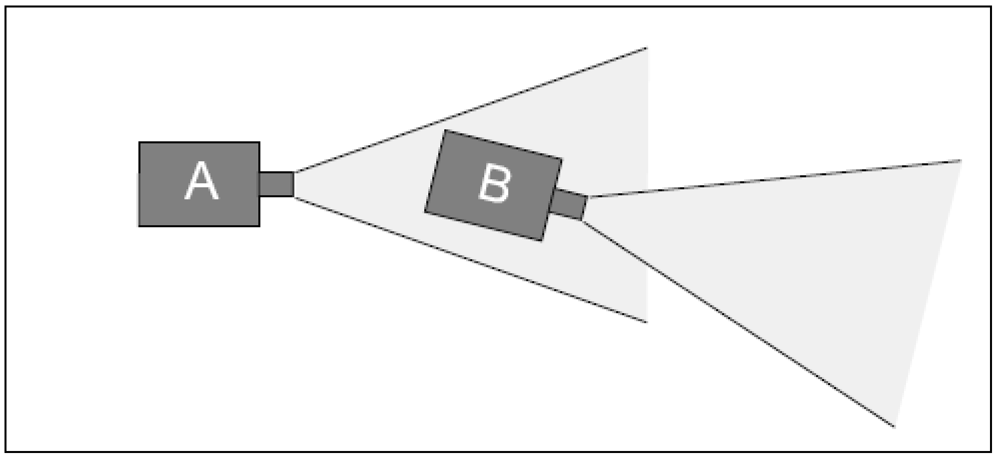
$$F^* = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$p'^T H'^T F^* H p = 0$$

$$F = H'^T F^* H$$

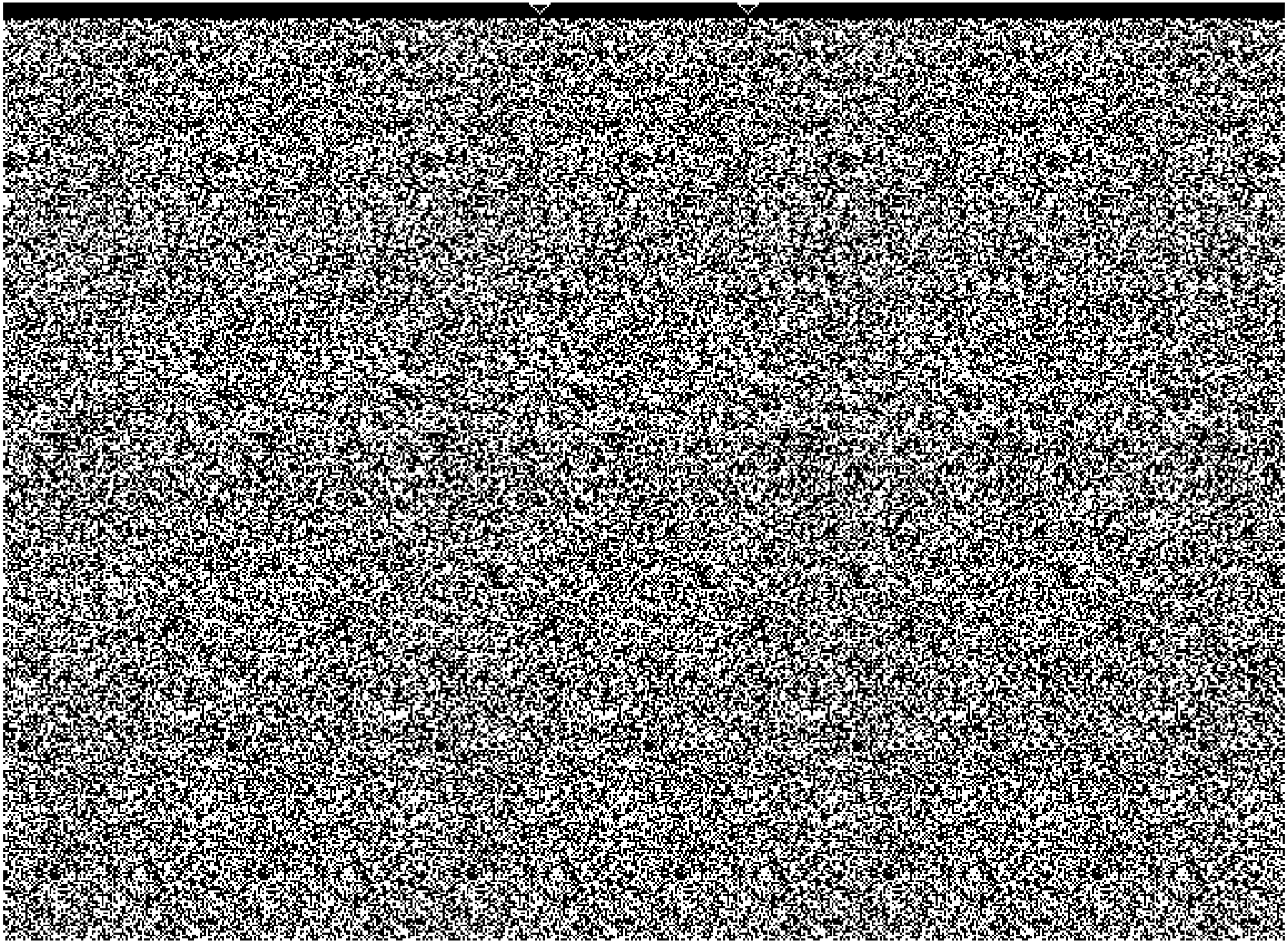
Stereo image rectification

When will this fail?



Actual image pixels end up at infinity!

Stereo Continued



Review: epipolar lines & rectification

Given two calibrated cameras:

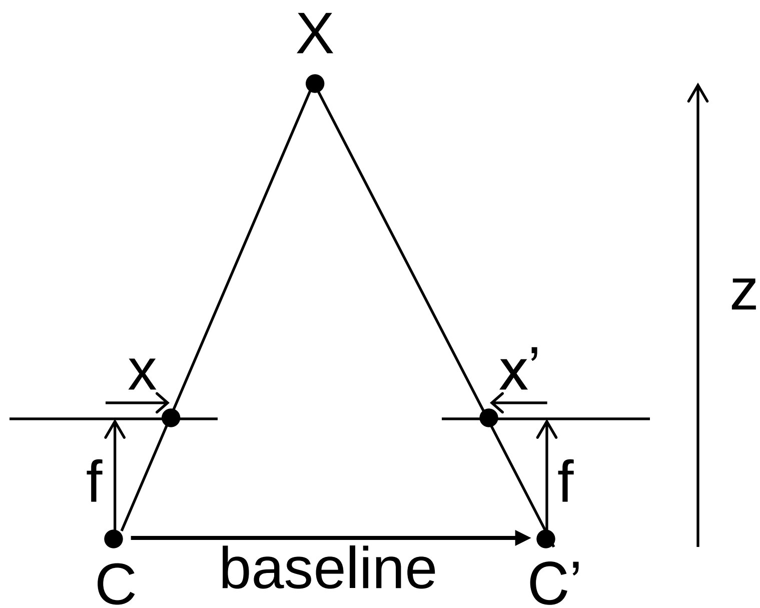
Each point in one image must lie on a line in the other.

All such lines go through the *epipole* (the projection of one camera center in the other image).

We can rectify images so that pairs of epipolar lines are horizontal, with the same *y* value.

The epipole for both images is then (1,0,0), a point at infinity.

Depth vs. Disparity



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

Pop Quiz

I go see Avatar 3D with a friend.

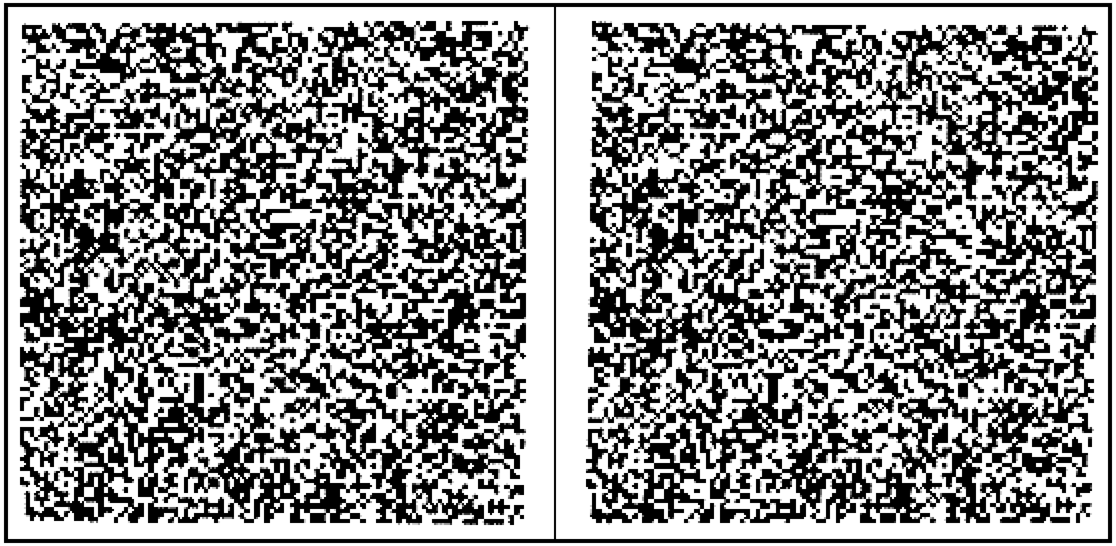
His eyes are further apart than mine.

me

friend

Who sees objects as being closer?

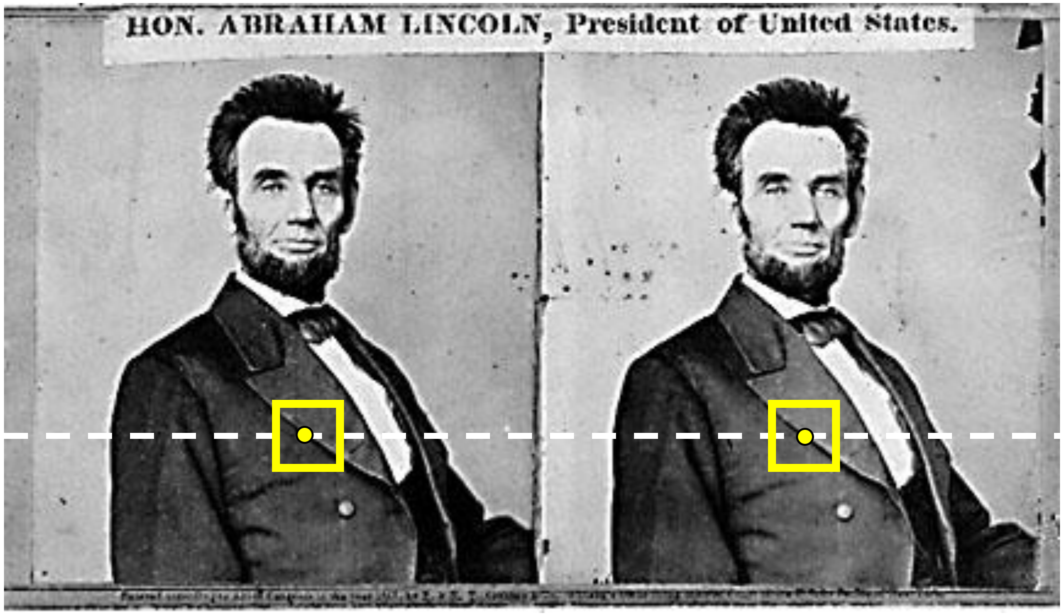
Stereo matching



Match Pixels in Conjugate Epipolar Lines

- Assume brightness constancy
- This is a tough problem
- Numerous approaches
  - A good survey and evaluation: http://vision.middlebury.edu/stereo/

Your basic stereo algorithm



For each epipolar line

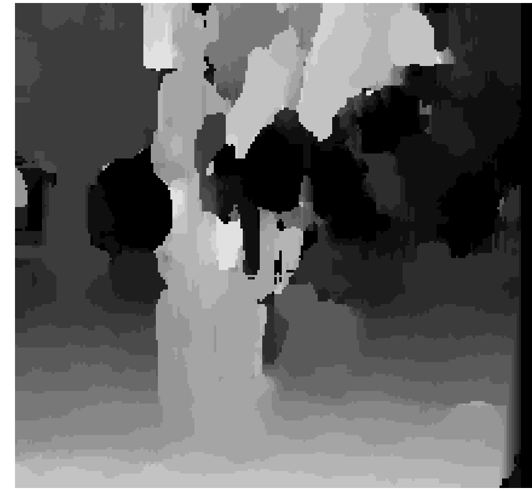
For each pixel in the left image

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

Improvement: match ***windows***

Window size

W = 3



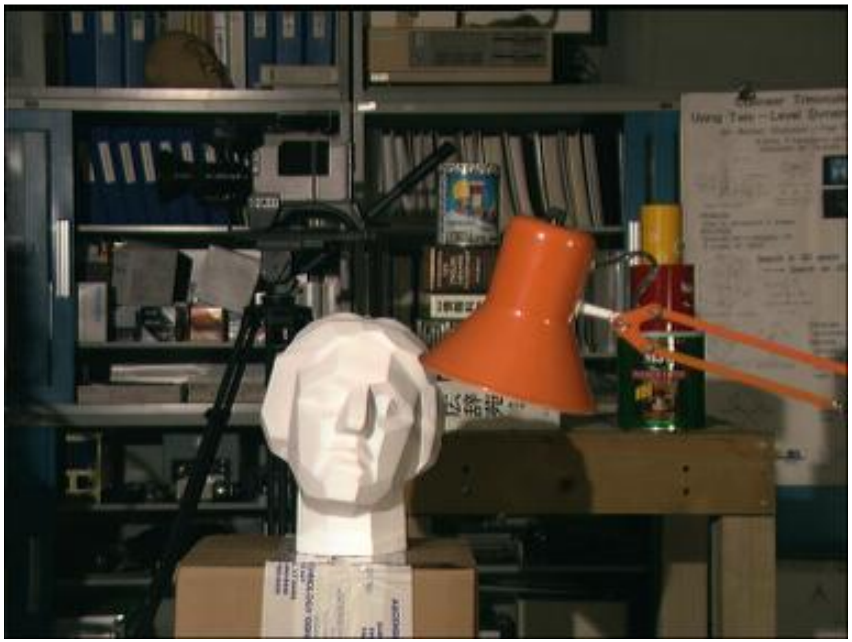
W = 20

Effect of window size

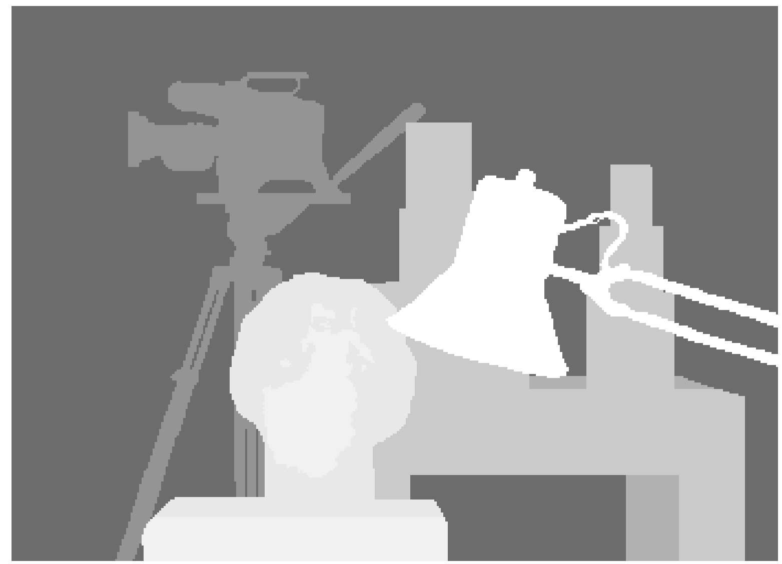
- Smaller window
  - +
  - –
- Larger window
  - +
  - –

Stereo results

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

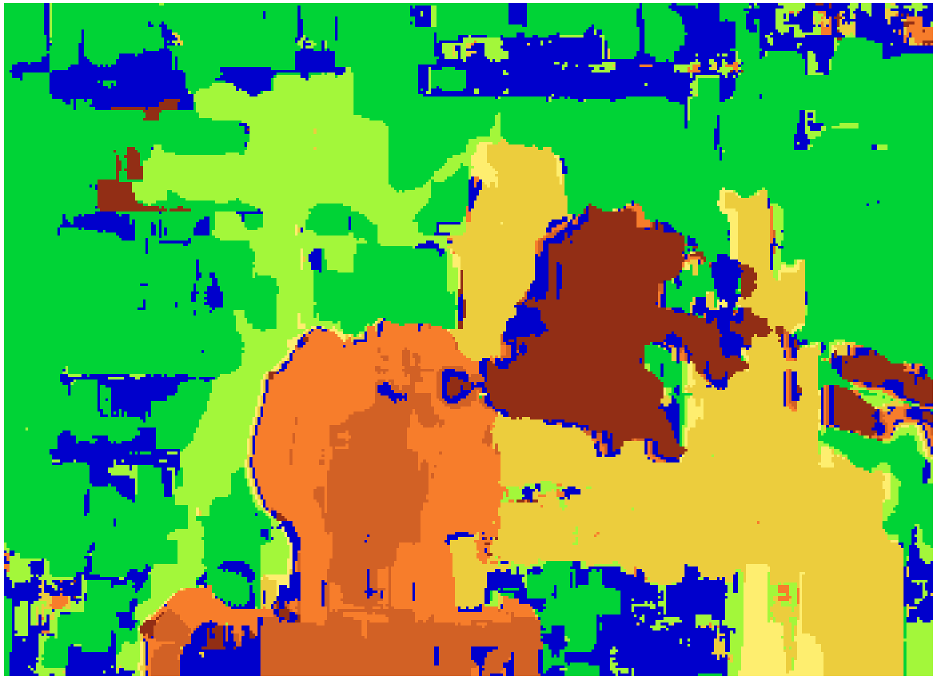


Scene



Ground truth

Results with window search

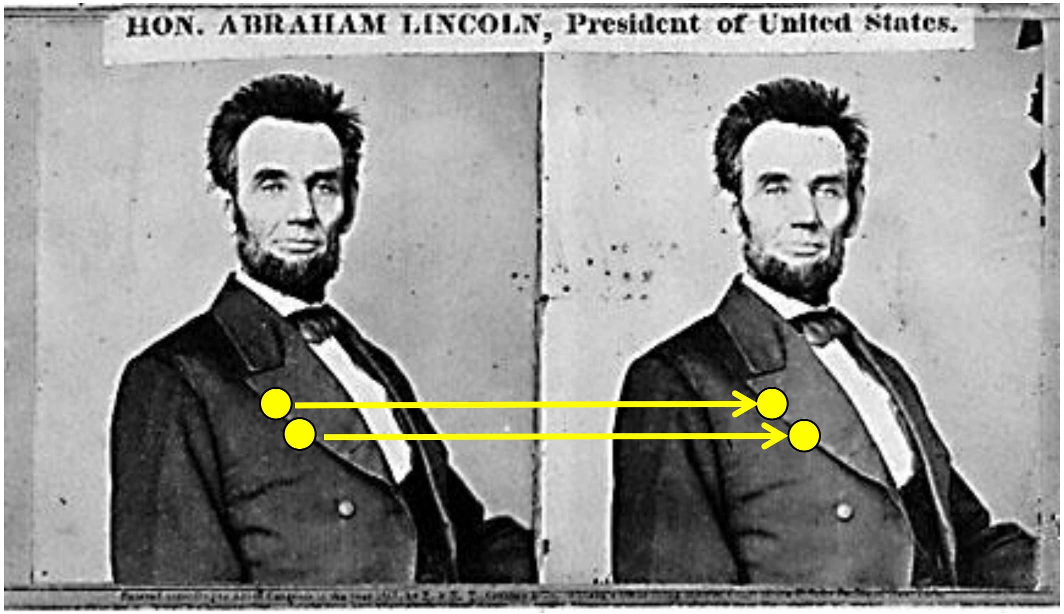


Window-based matching
(best window size)

Ground truth

How could we do better?

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

Expressing this mathematically

1. Match quality
   – Want each pixel to find a good match in the other image

$$matchCost = \sum_{x,y} \|I(x,y) - J(x + d_{xy}, y)\|$$

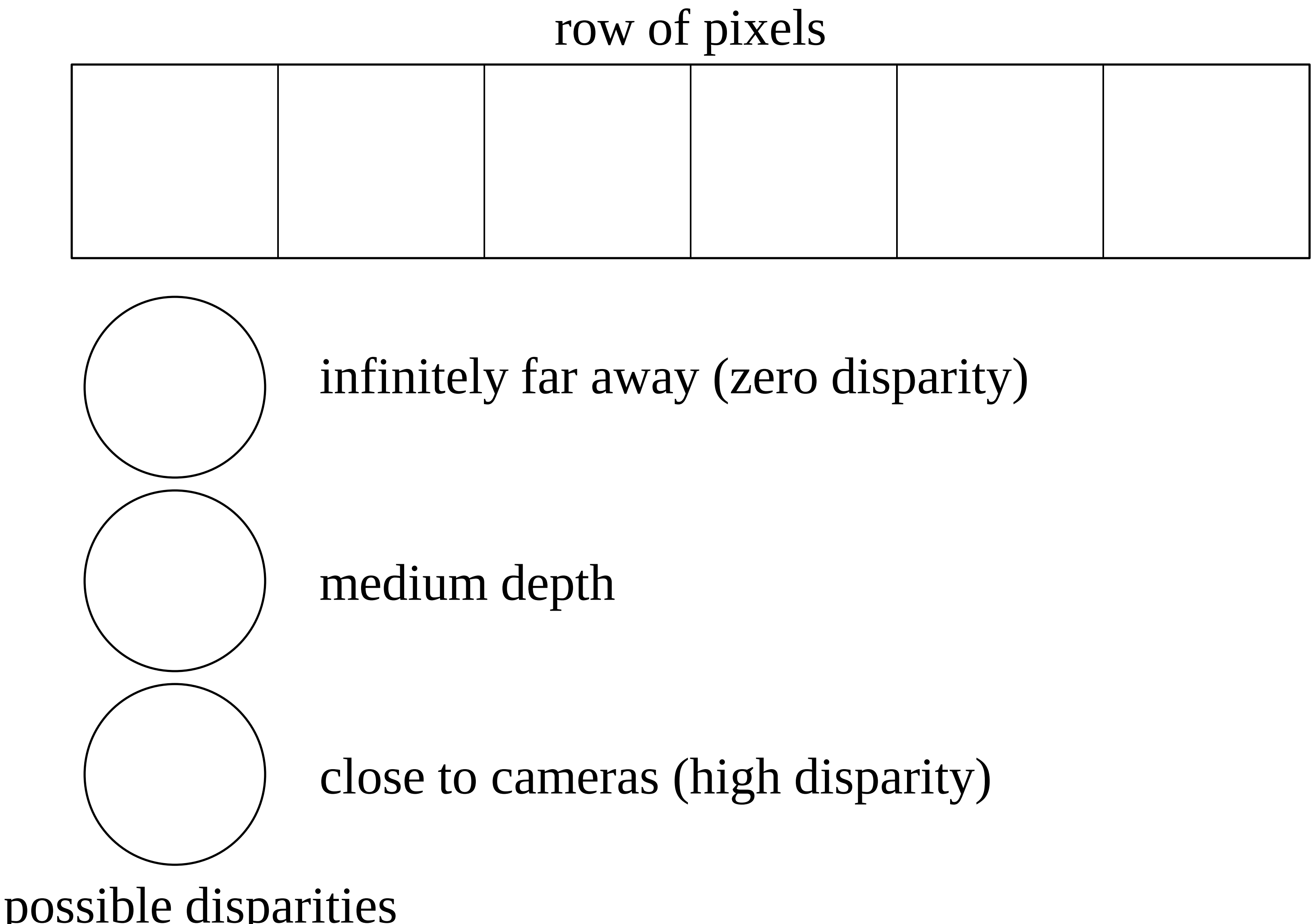
2. Smoothness
   – If two pixels are adjacent, they should (usually) move about the same amount

$$smoothnessCost = \sum_{neighbor\ pixels\ p,q} |d_p - d_q|$$

We want to minimize $Energy = matchCost + smoothnessCost$

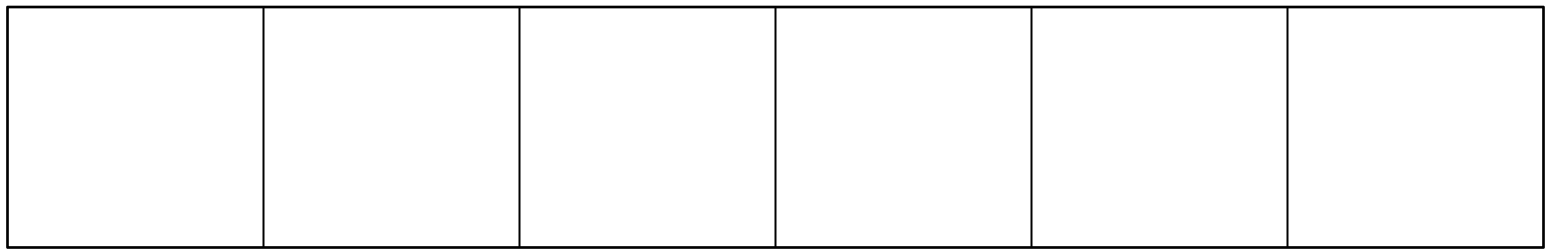
For now, let's consider horizontal smoothness only.

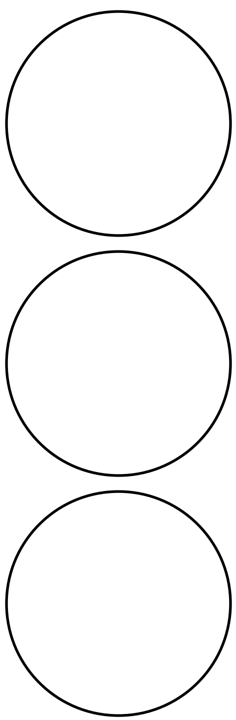
Stereo via Dynamic Programming



Stereo via Dynamic Programming

row of pixels

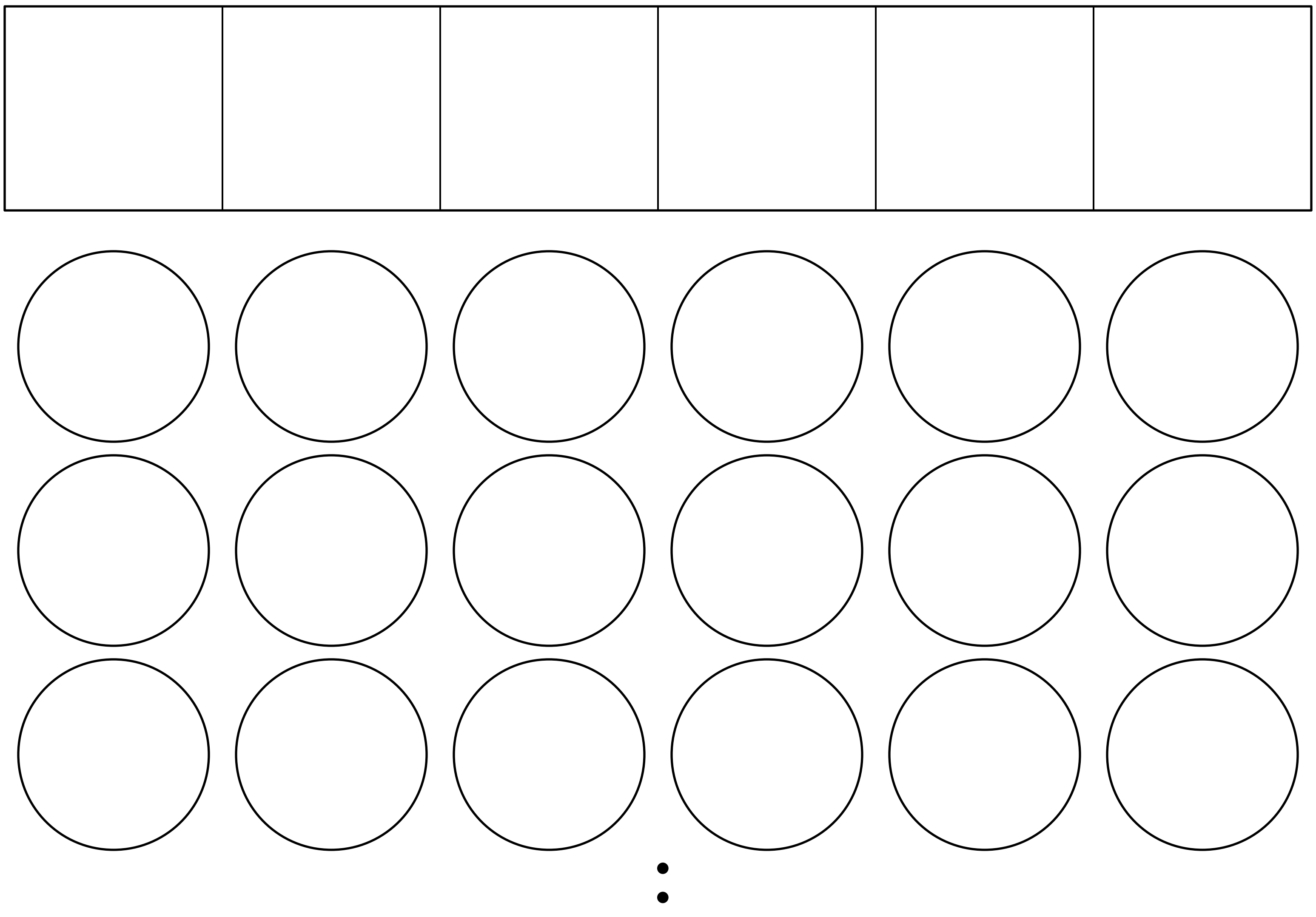




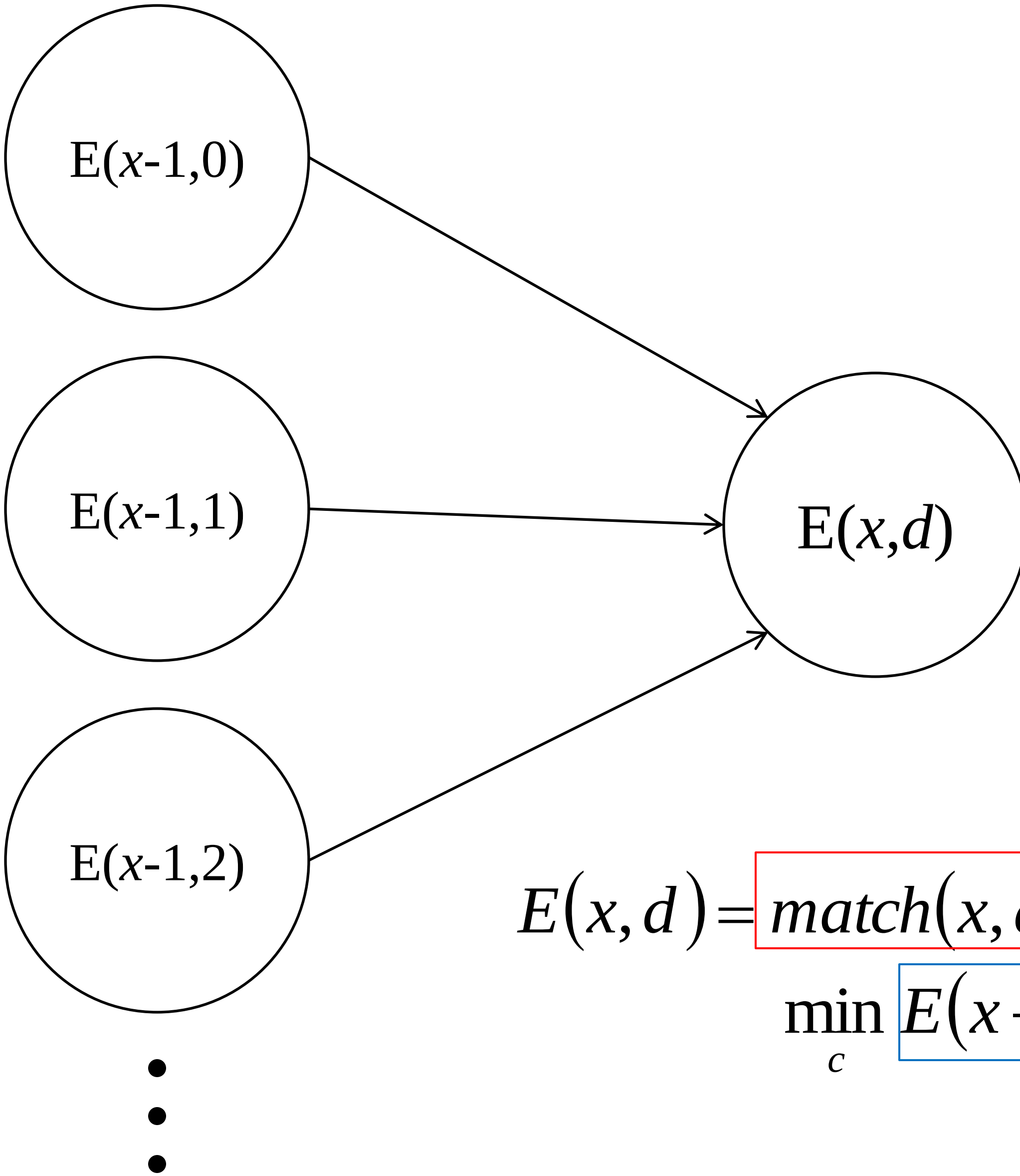
possible disparities

E(*x*,*d*) = minimum cost solution from pixels 1 to *x* where pixel *x* has disparity *d*

Stereo via Dynamic Programming



Stereo via Dynamic Programming



match cost
$|I(x) - J(x+d)|$

smoothness cost
$|c - d|$

best solution up to previous pixel

$$E(x,d) = \boxed{match(x,d)} + \min_{c} \boxed{E(x-1,c)} + \boxed{smooth(c,d)}$$

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://vision.middlebury.edu/stereo/

Stereo as energy minimization

Expressing this mathematically

1. Match quality
   - Want each pixel to find a good match in the other image

$$matchCost = \sum_{x,y} \|I(x,y) - J(x + d_{xy}, y)\|$$

2. Smoothness
   - If two pixels are adjacent, they should (usually) move about the same amount

$$smoothnessCost = \sum_{neighbor\ pixels\ p,q} |d_p - d_q|$$

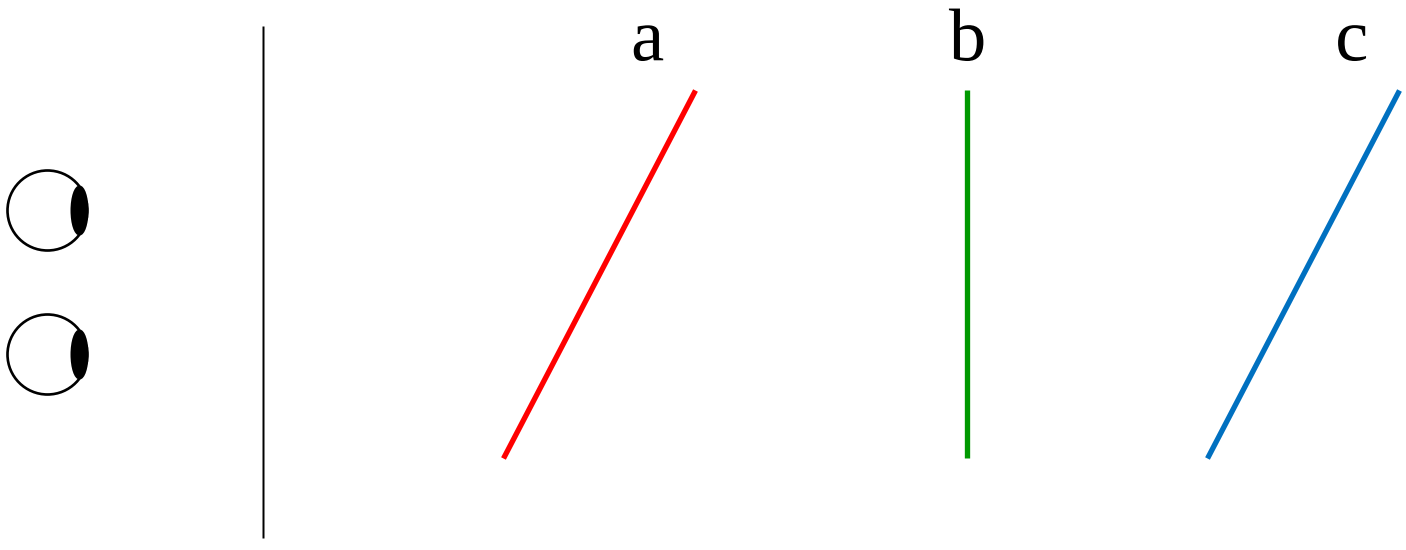
We want to minimize $Energy = matchCost + smoothnessCost$

- With 2D smoothness, this is a special type of energy function known as an MRF (Markov Random Field)
  - Effective and fast algorithms have been recently developed:
    - » Graph cuts, belief propagation….
    - » for more details (and code): http://vision.middlebury.edu/MRF/

What does this smoothness do?

Penalizes surfaces not parallel to the image plane

Which surface incurs the least cost? The most cost?

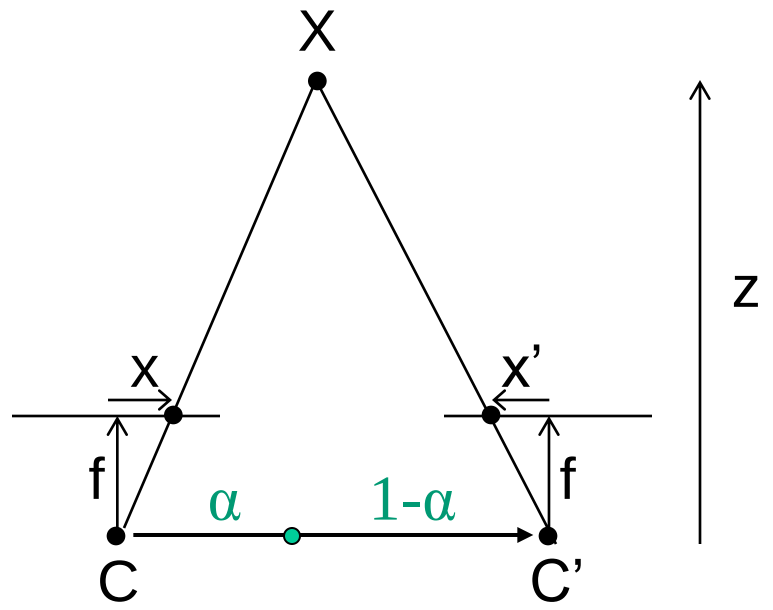


Stereo Applications

View Interpolation

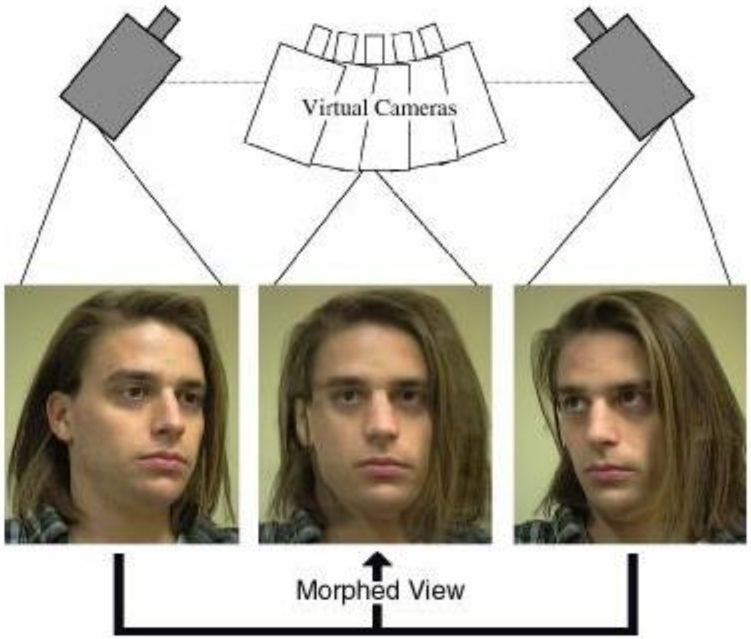
Suppose we put a camera at (1-α)C + αC'.

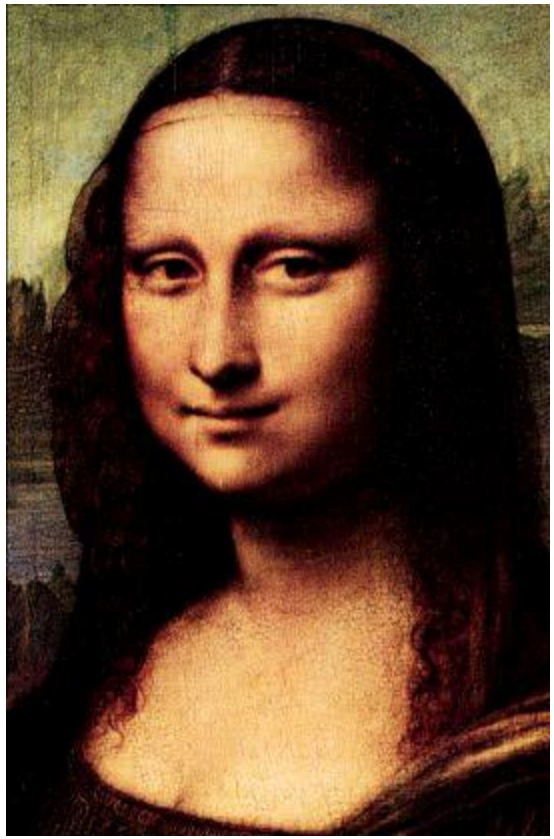
Where would X project to this image?



View Morphing

[Seitz & Dyer 96]





Stereo reconstruction pipeline

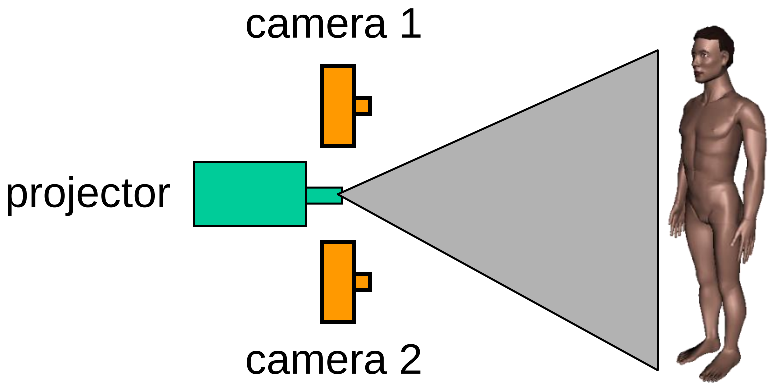
Steps

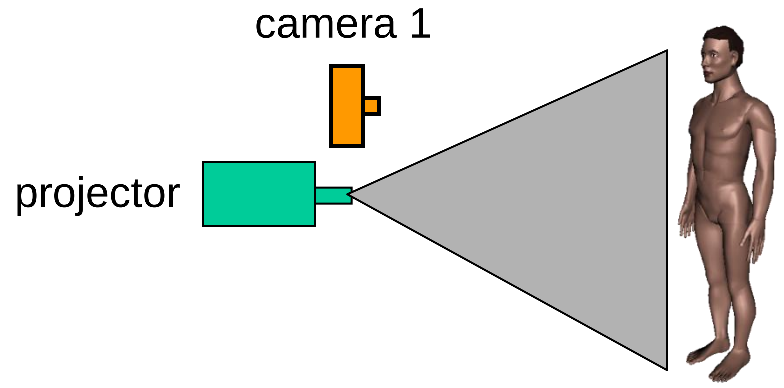
- Calibrate cameras
- Rectify images
- Compute disparity
- Estimate depth

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

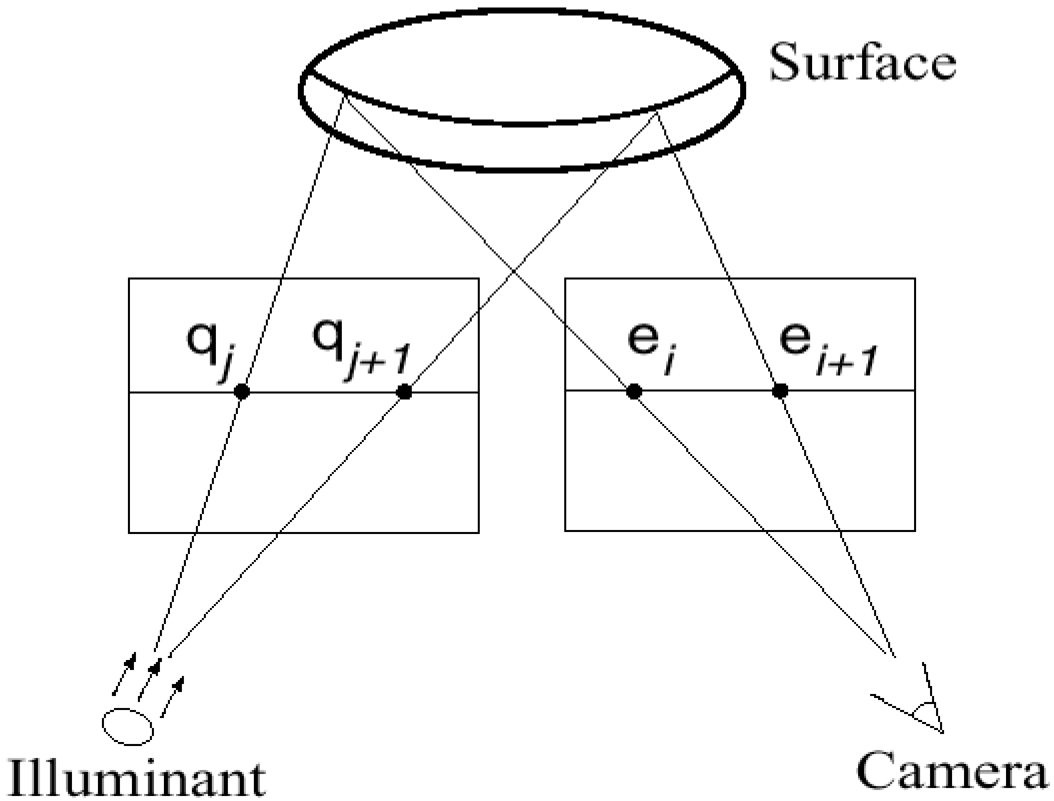




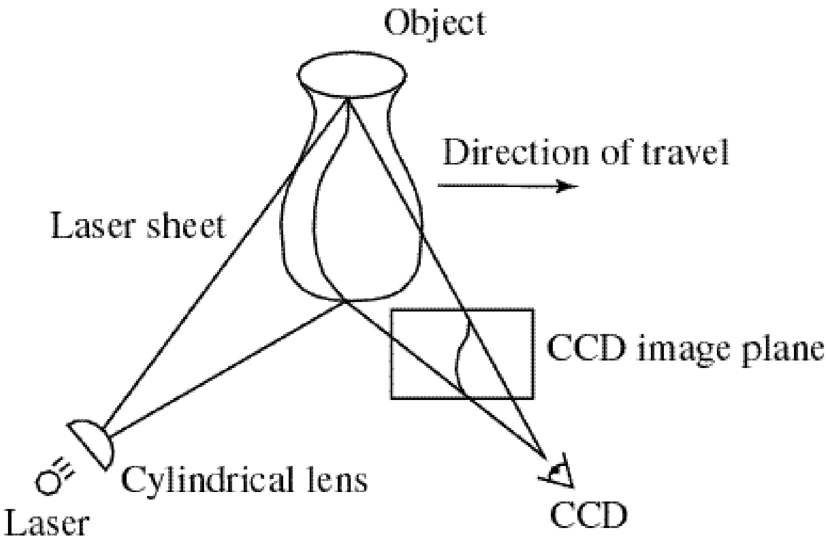
Project “structured” light patterns onto the object

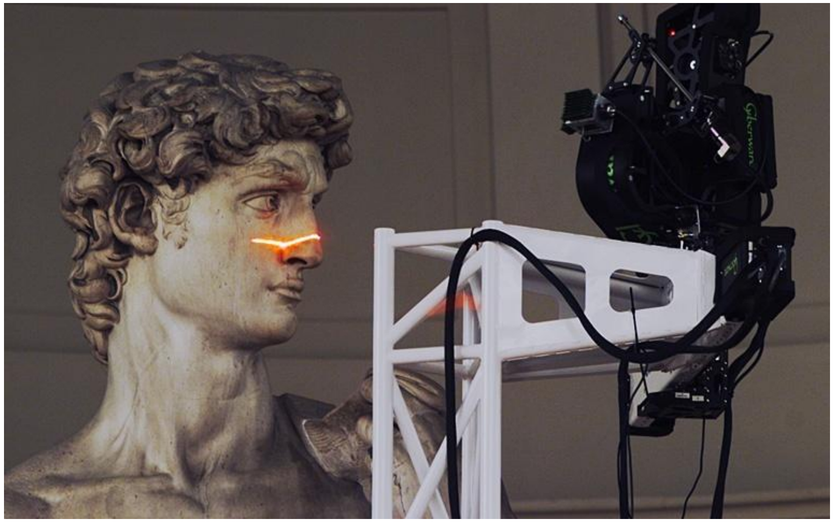
- simplifies the correspondence problem

One light, one camera



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

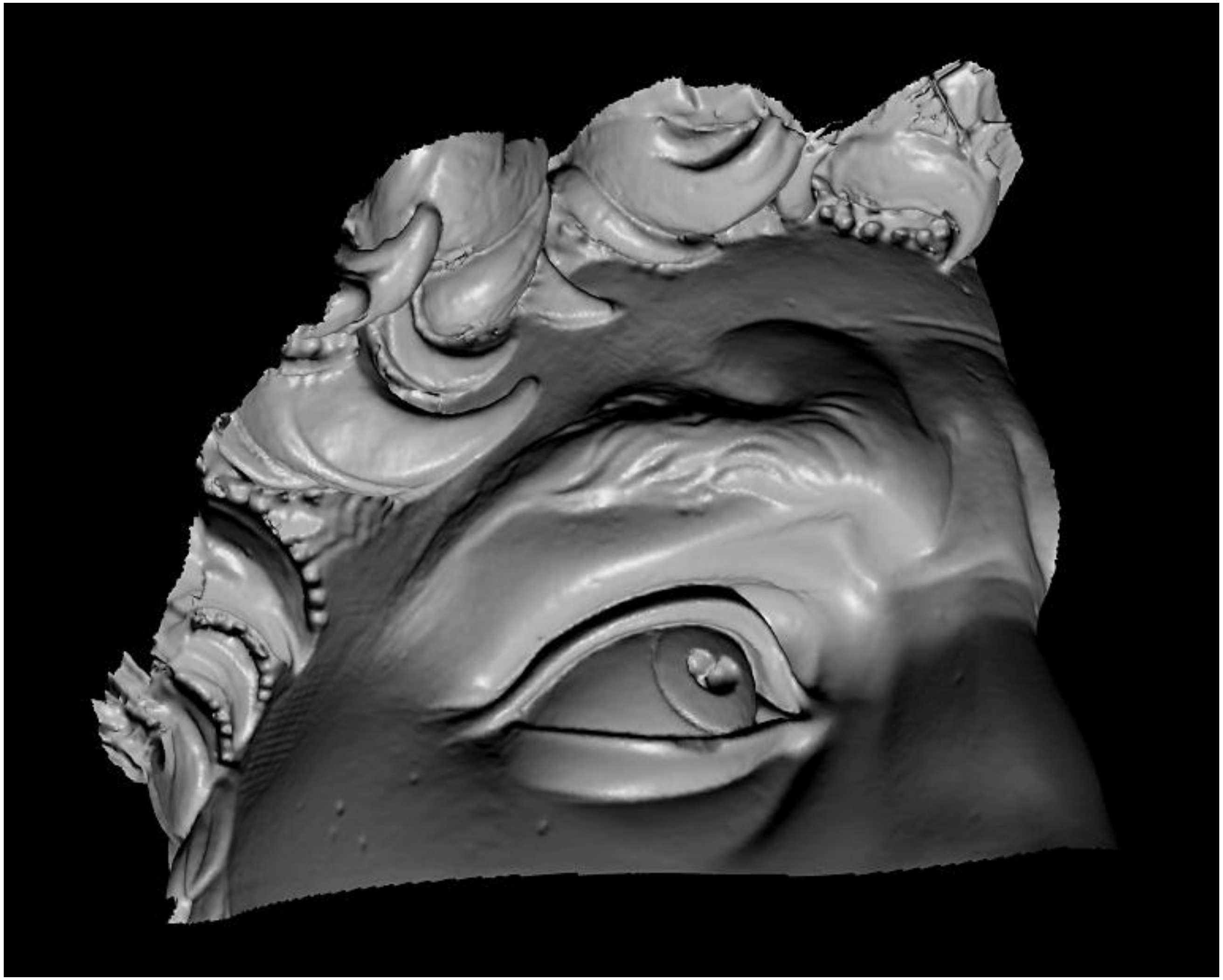
Laser scanned models

The Digital Michelangelo Project, Levoy et al.

Laser scanned models

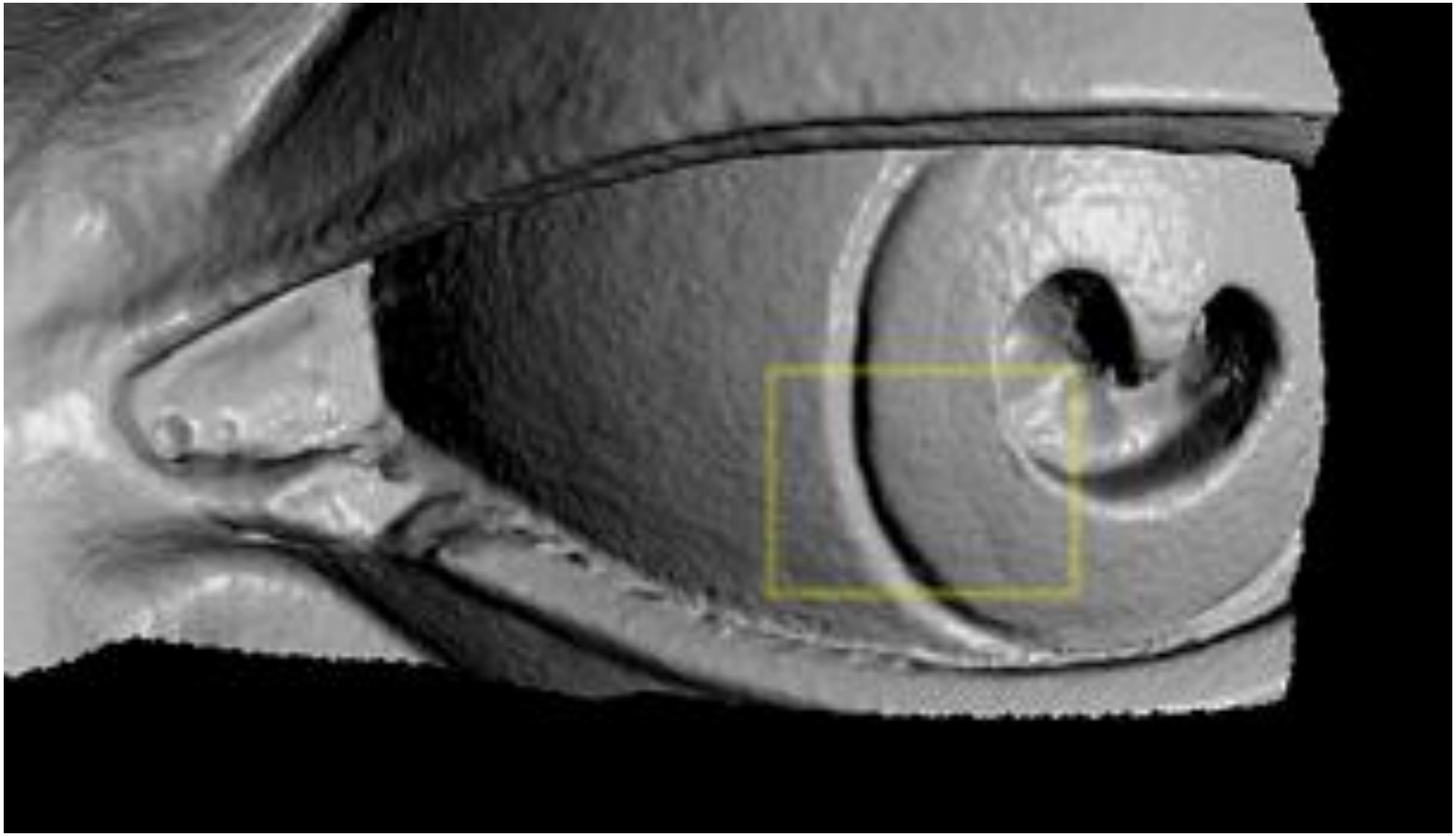
The Digital Michelangelo Project, Levoy et al.

Laser scanned models



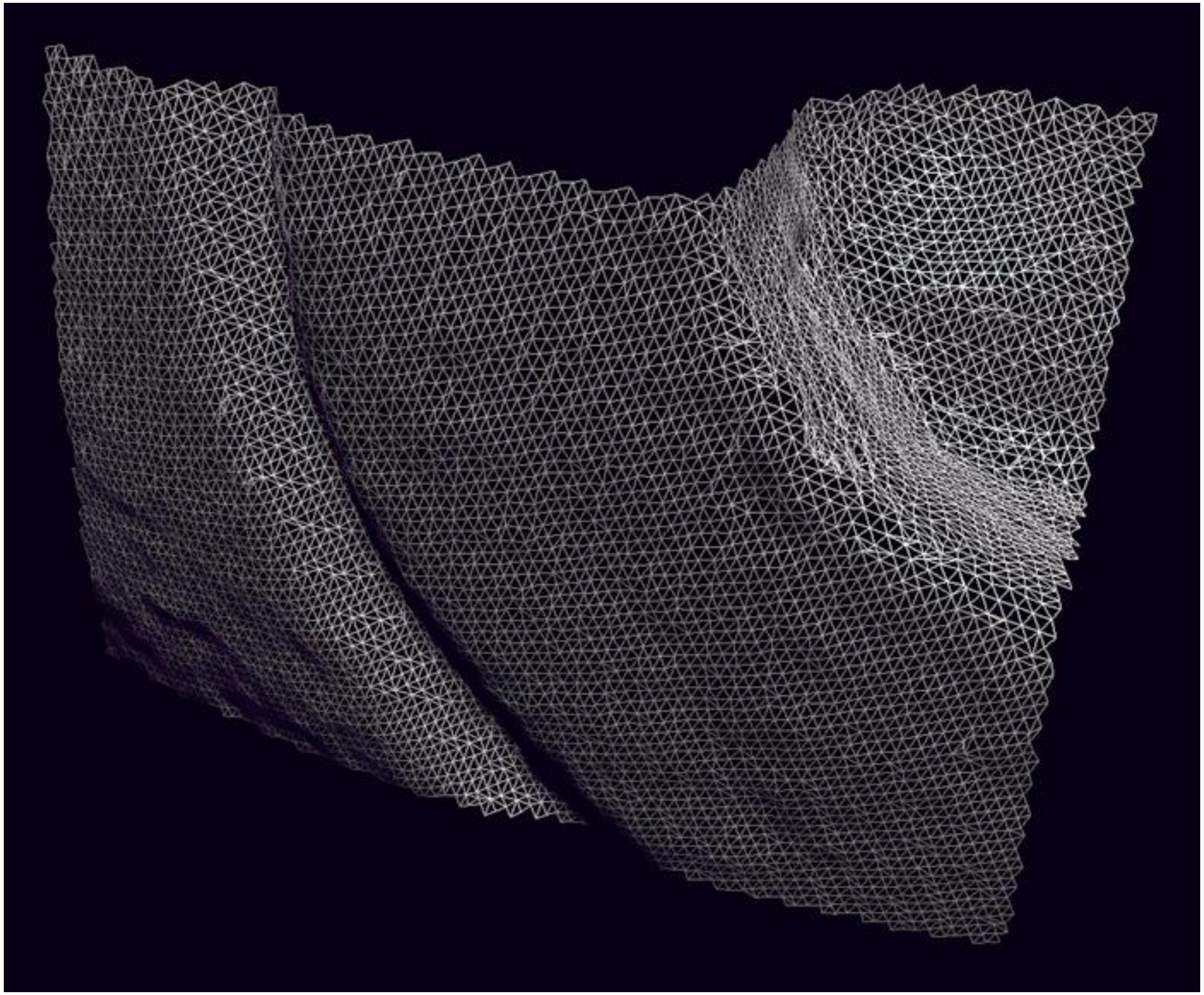
The Digital Michelangelo Project, Levoy et al.

Laser scanned models



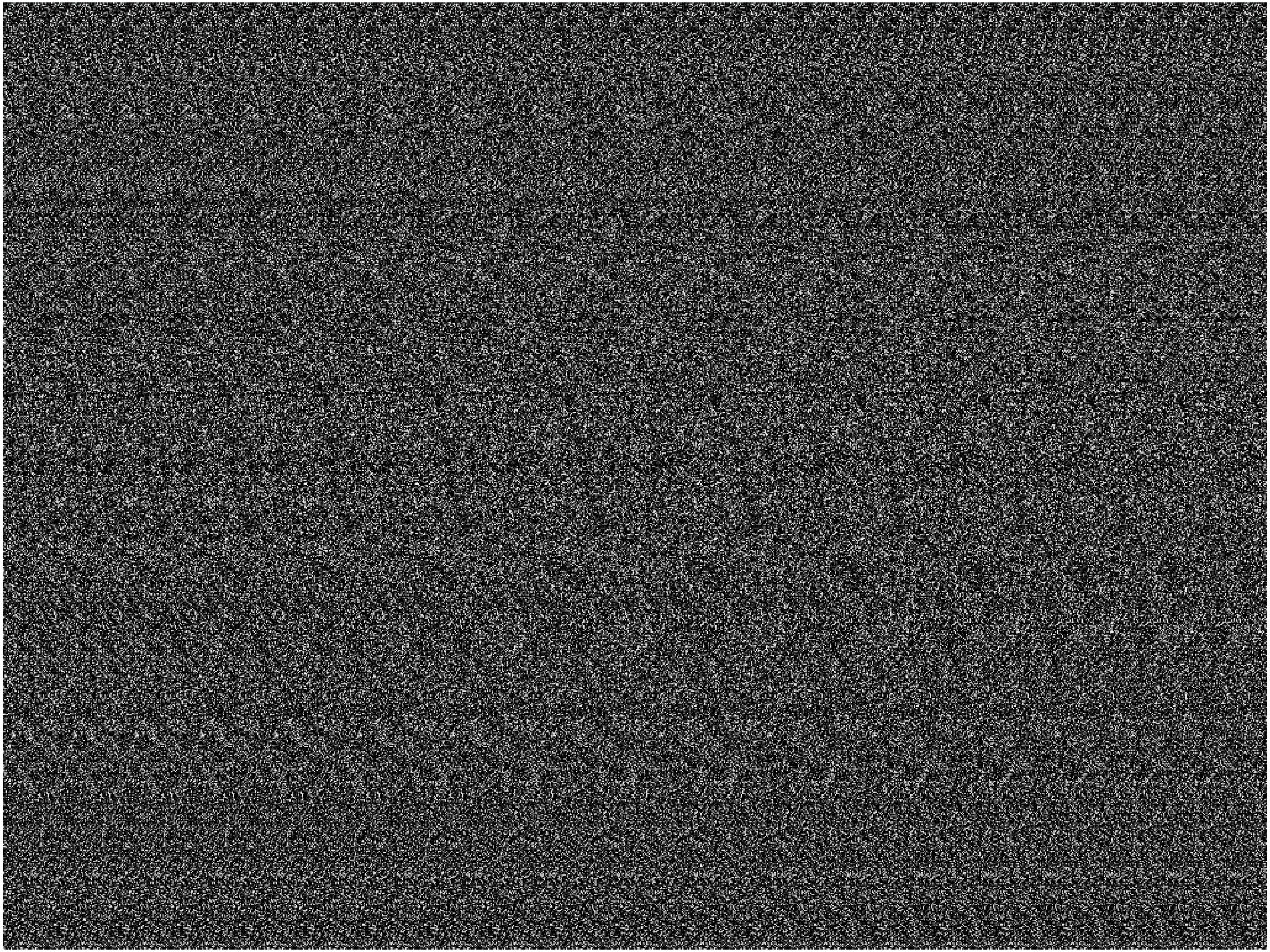
The Digital Michelangelo Project, Levoy et al.

Laser scanned models



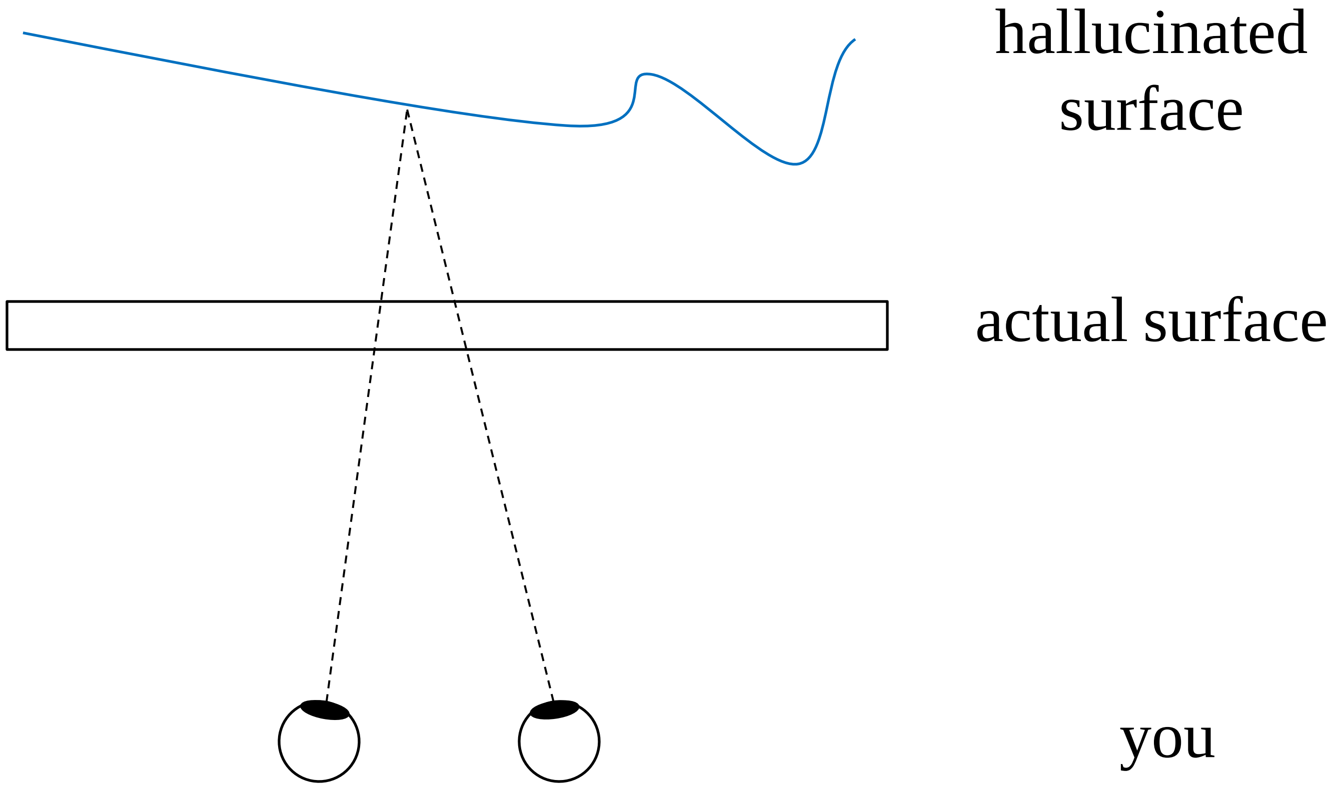
The Digital Michelangelo Project, Levoy et al.

Autostereograms



Autostereograms

How do they work?



Autostereograms

How do they work?