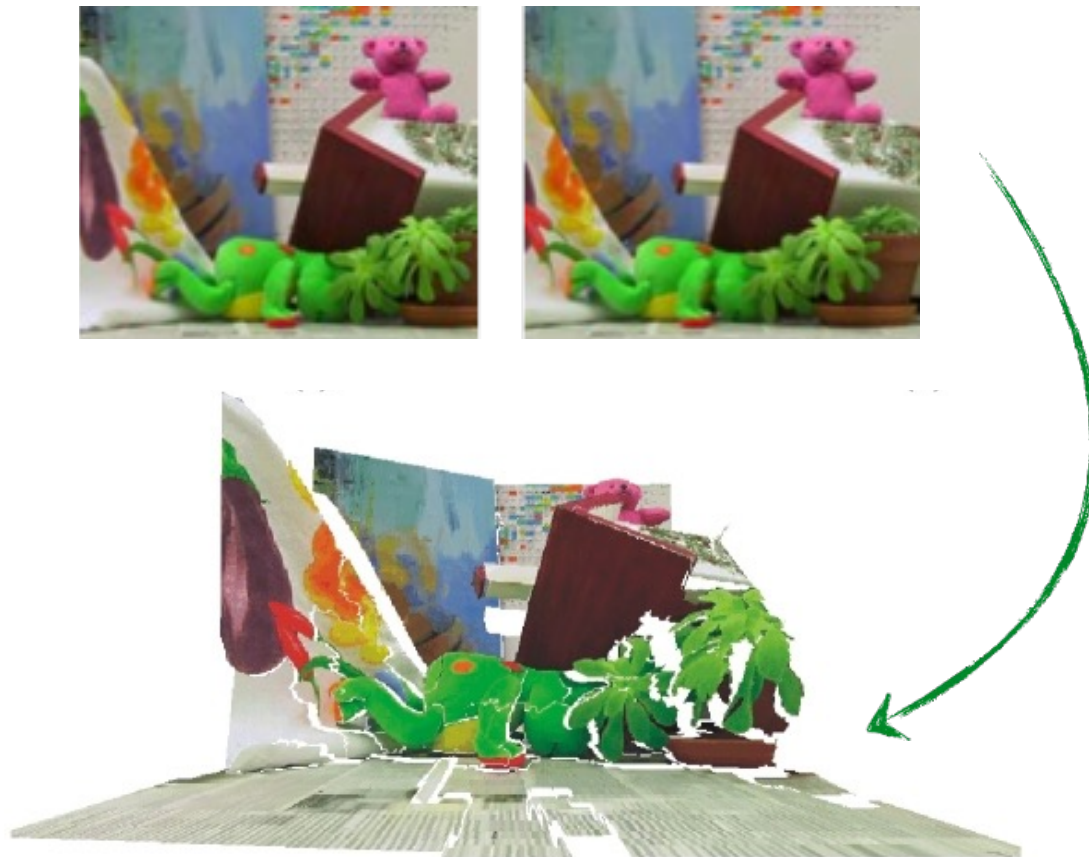


Two-View Stereo



Many slides adapted from Steve Seitz

Problem formulation

- **Given:** stereo pair (assumed calibrated)
- **Wanted:** dense depth map



Outline

- Motivation and history
- Basic two-view stereo setup
- Local stereo matching algorithm
- Beyond local stereo matching
- Active stereo with structured light

Stereo vision and perception of depth

- What cues tell us about scene depth?



How Two Photographers Unknowingly Shot the Same Millisecond in Time

MAR 07, 2018

RON RISMAN

PetaPixel



<https://petapixel.com/2018/03/07/two-photographers-unknowingly-shot-millisecond-time/>

How Two Photographers Unknowingly Shot the Same Millisecond in Time

MAR 07, 2018

RON RISMAN

PetaPixel



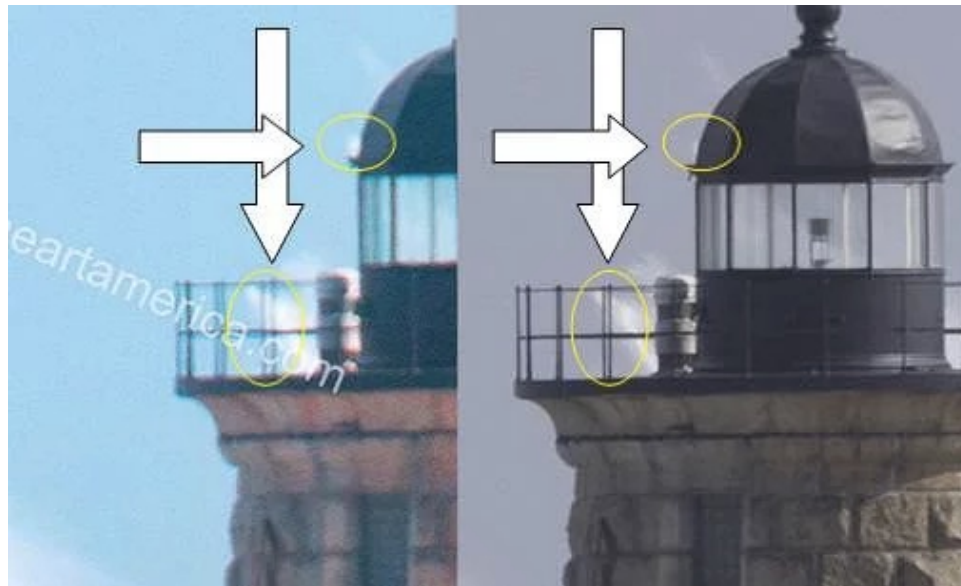
<https://petapixel.com/2018/03/07/two-photographers-unknowingly-shot-millisecond-time/>

How Two Photographers Unknowingly Shot the Same Millisecond in Time

MAR 07, 2018

RON RISMAN

PetaPixel



<https://petapixel.com/2018/03/07/two-photographers-unknowingly-shot-millisecond-time/>

History: Stereograms

- Humans can fuse pairs of images to get a sensation of depth

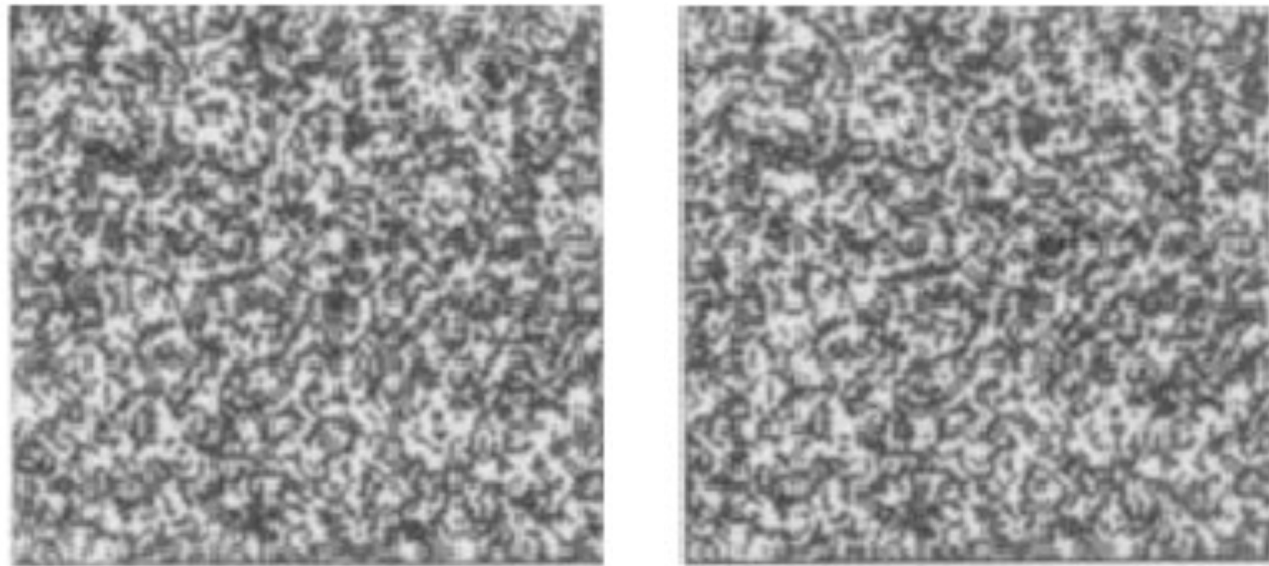


Stereograms: Invented by Sir Charles Wheatstone, 1838

<https://en.wikipedia.org/wiki/Stereoscopy>

History: Random dot stereograms

- Invented by [Bela Julesz](#) in the mid-20th century
- Demonstration that stereo perception can happen without any monocular cues



https://en.wikipedia.org/wiki/Random_dot_stereogram

Outline

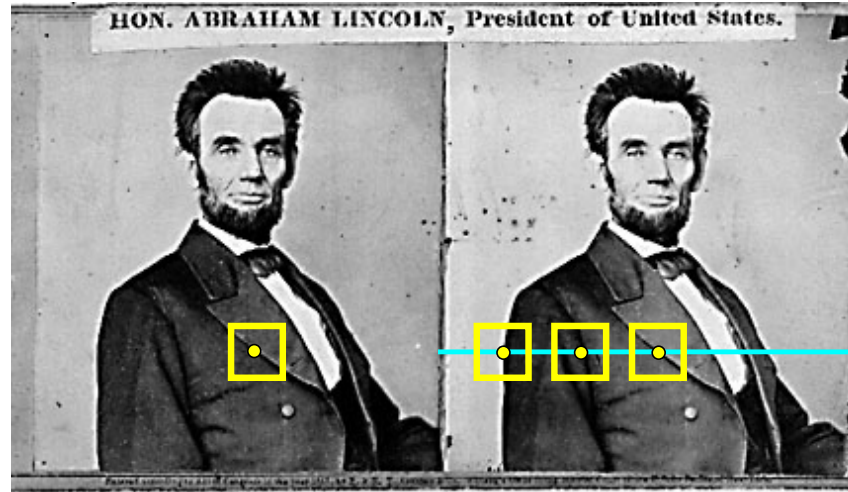
- Motivation and history
- Basic two-view stereo setup

Problem formulation

- **Given:** stereo pair (assumed calibrated)
- **Wanted:** dense depth map

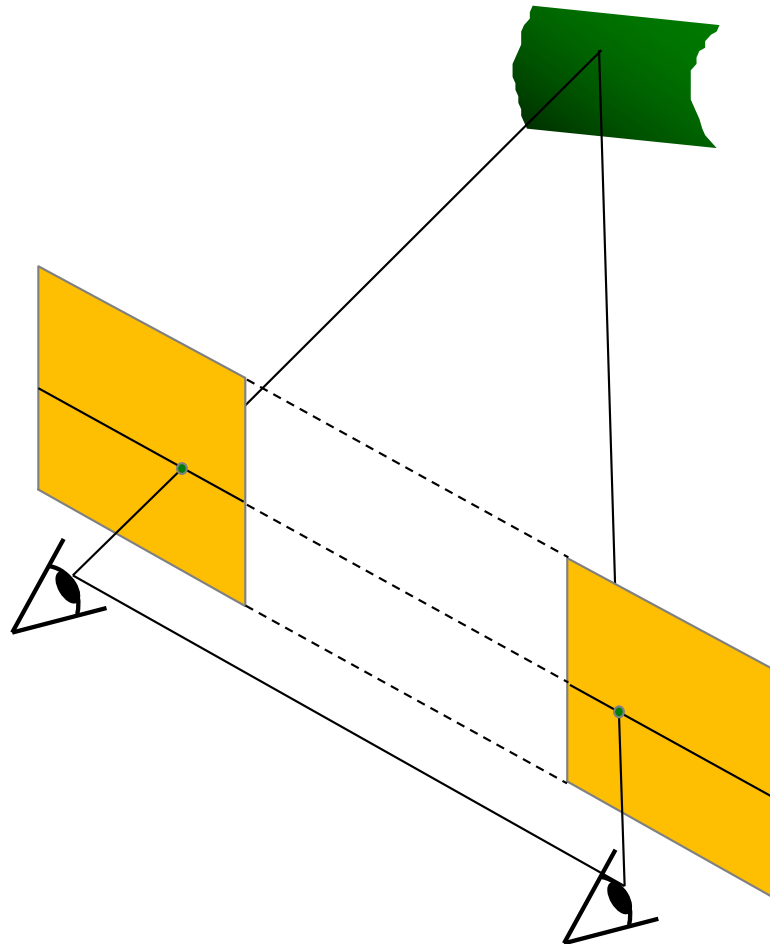


Basic stereo matching algorithm



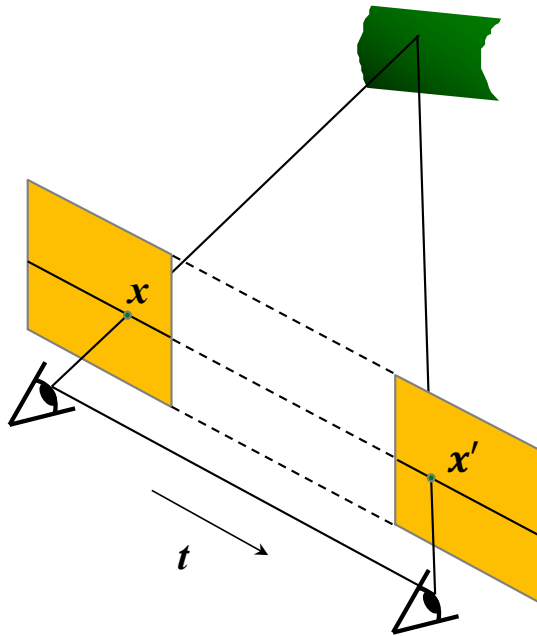
- For each pixel in the first image
 - Find corresponding epipolar line in the right image
 - Examine all pixels on the epipolar line and pick the best match
 - Triangulate the matches to get depth information
- Simplest case: epipolar lines are corresponding scanlines
 - When does this happen?

Parallel images



- Image planes of cameras are parallel to each other and to the baseline
- Camera centers are at the same height
- Focal lengths are the same
- Then epipolar lines fall along horizontal scan lines of the images

Essential matrix for parallel images



Epipolar constraint:

$$\mathbf{x}'^T \mathbf{E} \mathbf{x} = 0, \quad \mathbf{E} = [\mathbf{t}_\times] \mathbf{R}$$

$$\mathbf{R} = \mathbf{I} \quad \mathbf{t} = (t, 0, 0)$$

$$\mathbf{E} = [\mathbf{t}_\times] \mathbf{R} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t \\ 0 & t & 0 \end{bmatrix}$$

$$(u' \ v' \ 1) \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t \\ 0 & t & 0 \end{bmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = 0$$

$$(u' \ v' \ 1) \begin{pmatrix} 0 \\ -t \\ tv \end{pmatrix} = 0$$

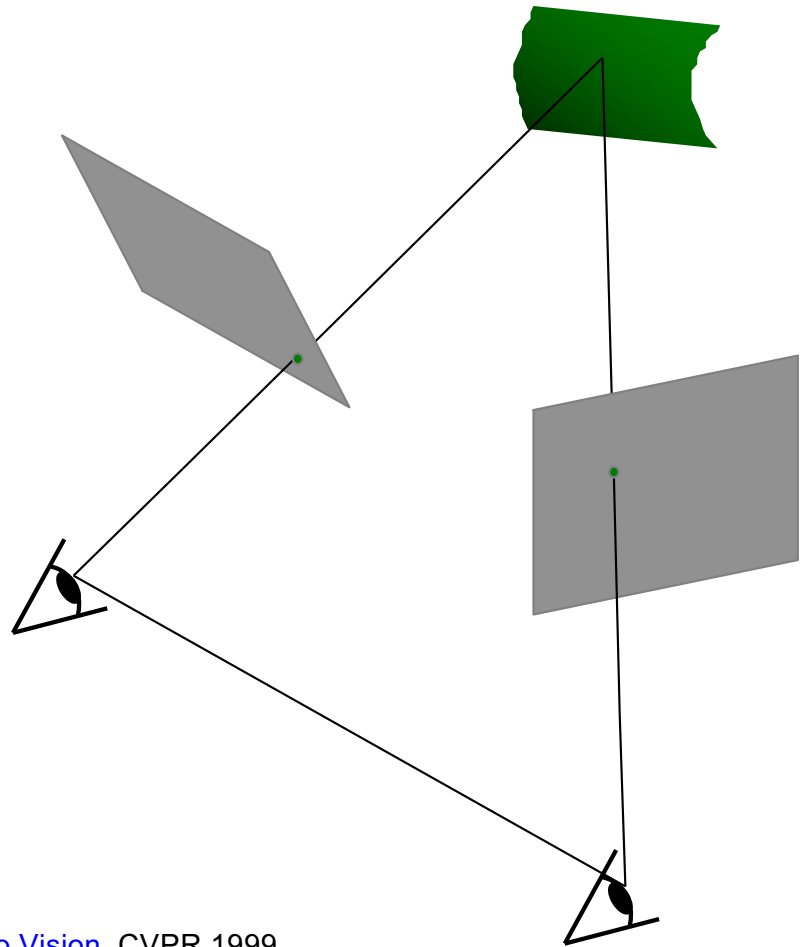
$$-tv + tv' = 0$$

$$v = v'$$

The y-coordinates of corresponding points are the same!

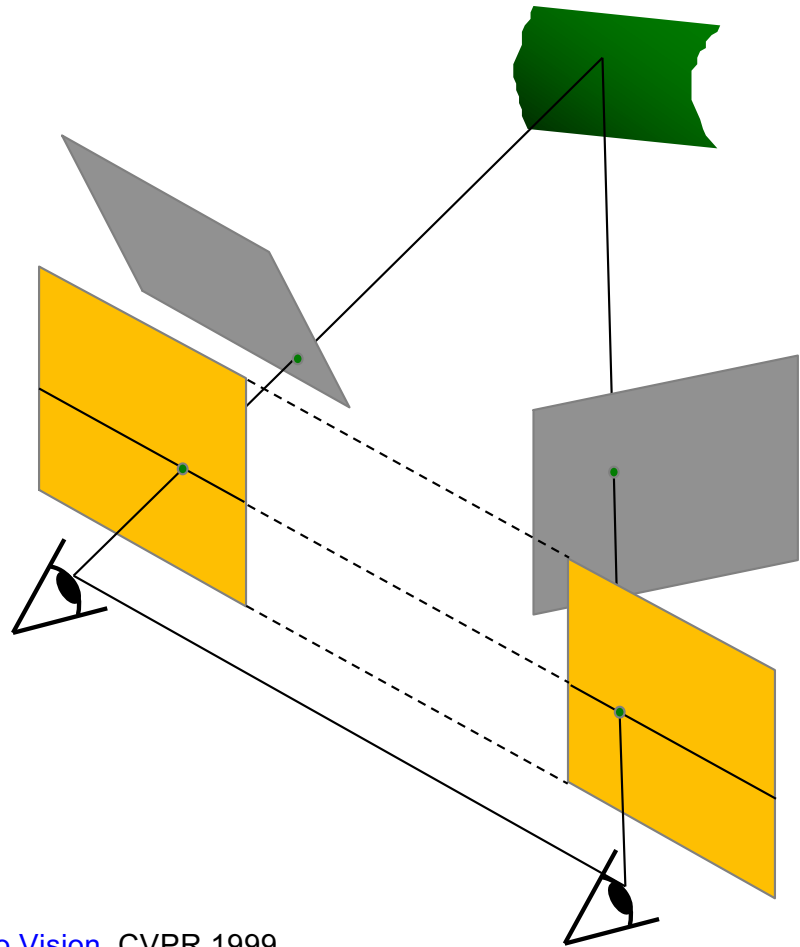
Stereo image rectification

- If the image planes are not parallel, we can find homographies to project each view onto a common plane parallel to the baseline



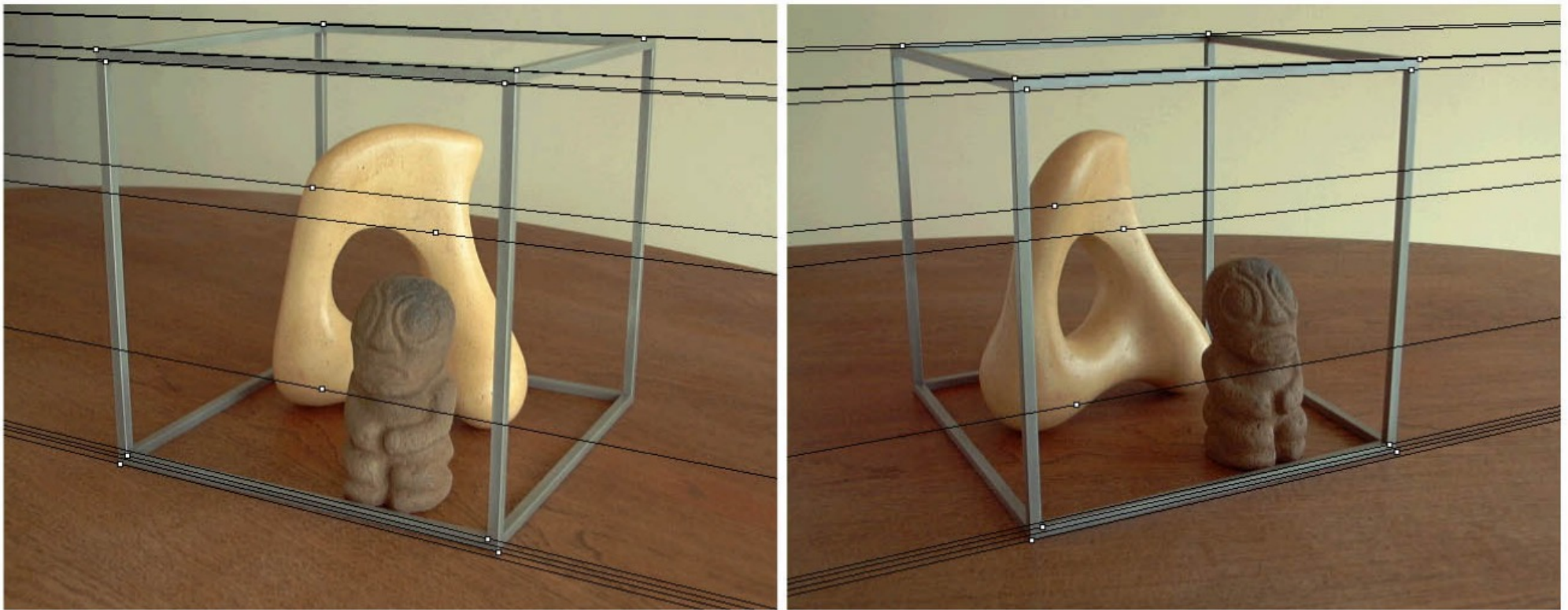
Stereo image rectification

- If the image planes are not parallel, we can find homographies to project each view onto a common plane parallel to the baseline



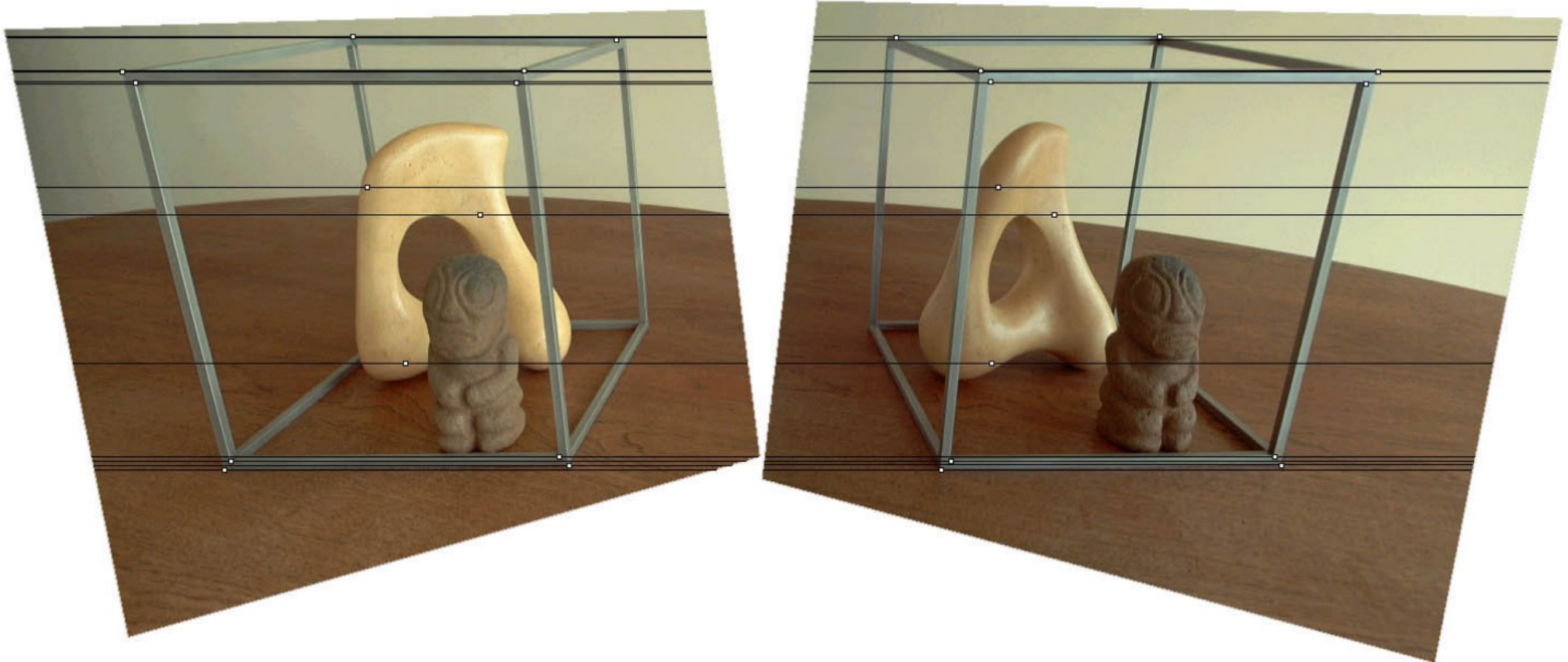
Stereo image rectification

- Before rectification:



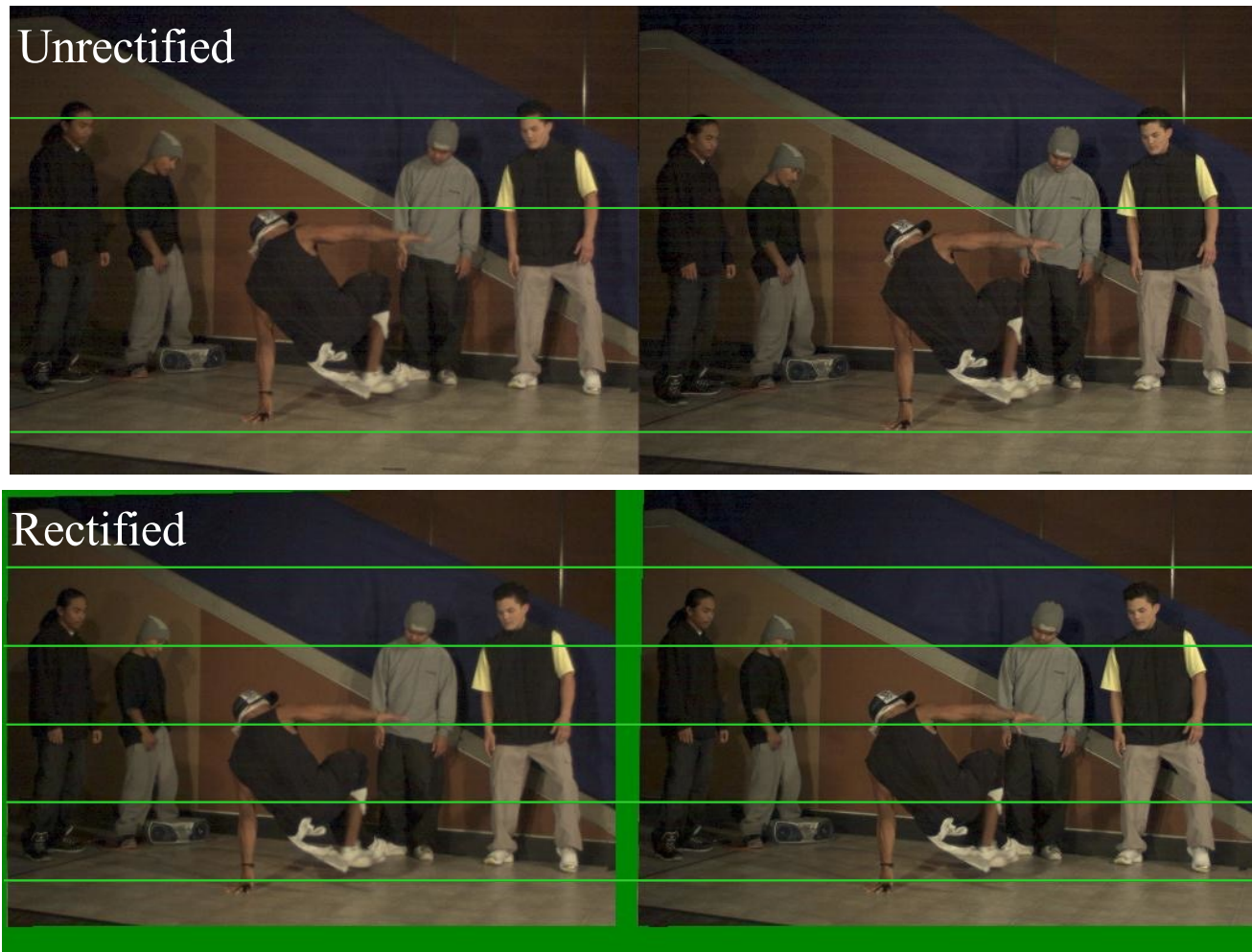
Stereo image rectification

- After rectification:

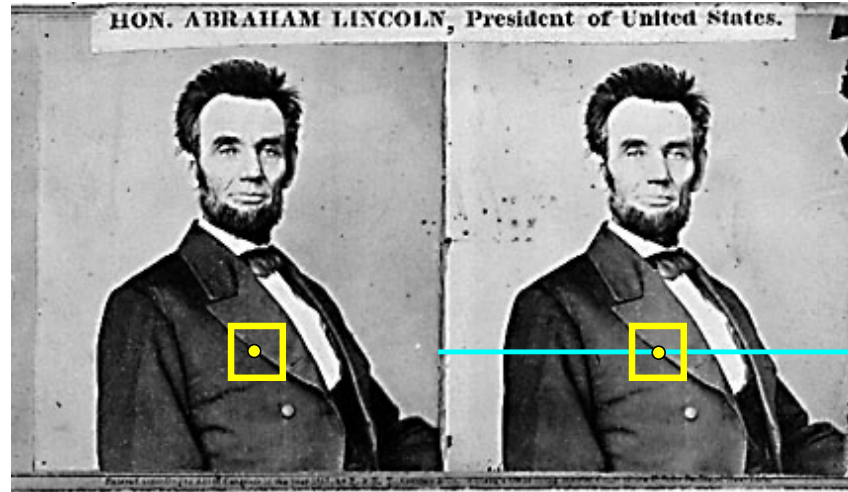


[Image source](#)

Another rectification example

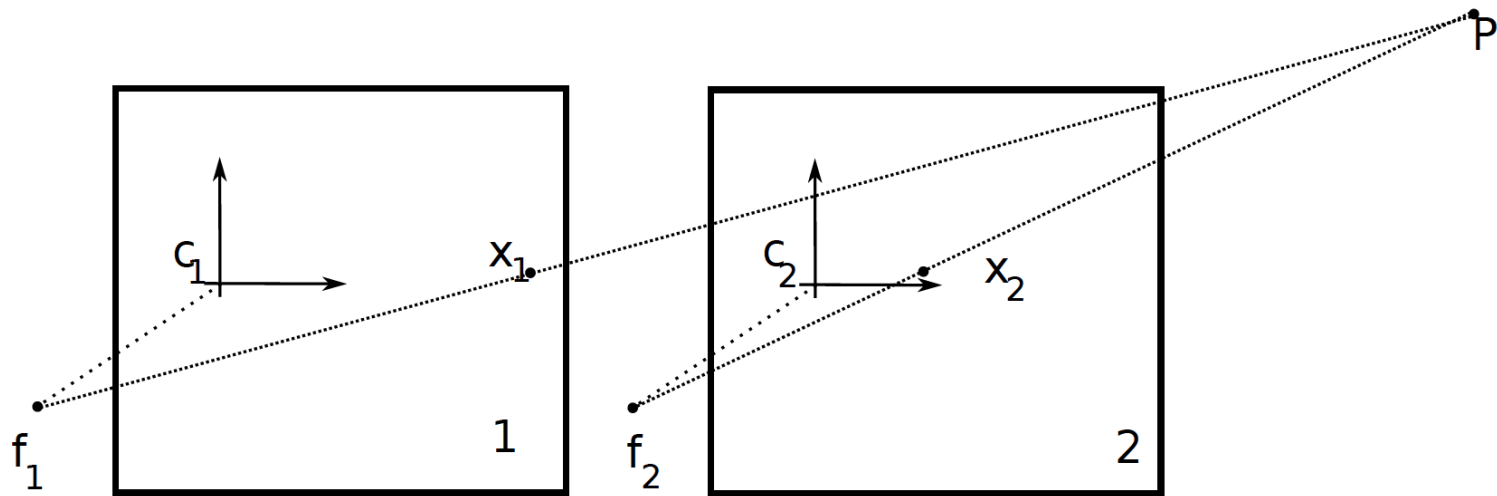


Basic stereo matching algorithm



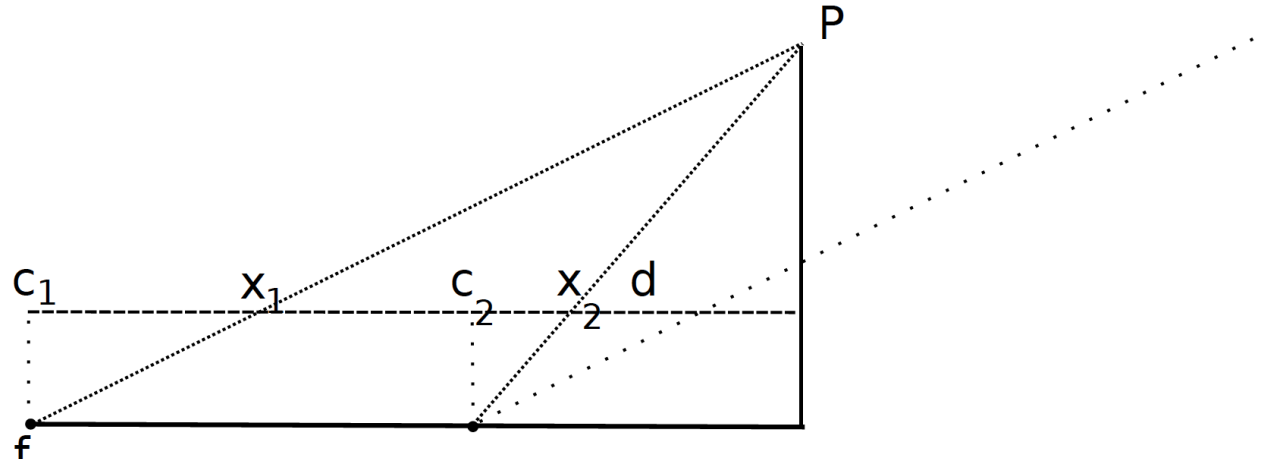
- If necessary, rectify the two stereo images to transform epipolar lines into scanlines
- For each pixel x in the first image
 - Find corresponding epipolar scanline in the right image
 - Examine all pixels on the scanline and pick the best match x'
- Triangulate the matches to get depth information

Two cameras view a point...



RGBD cameras). Here we show a specialized camera geometry, chosen to simplify notation. The second camera is translated with respect to the first, along a direction parallel to the image plane. The second camera is a copy of the first camera, so the image planes are parallel. In this geometry, the point being viewed shifts somewhat to the left in the right camera.

Two cameras view a point...

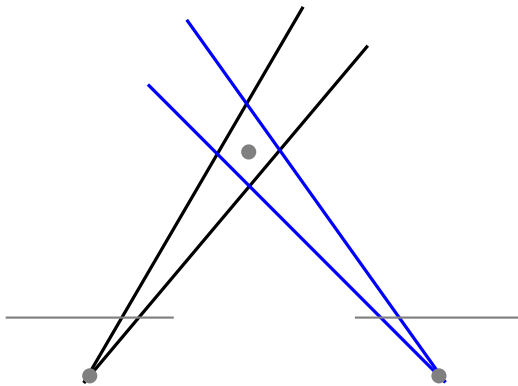


But there are limits to stereopsis. Measuring large depths with two cameras that are close together requires highly accurate estimates of point positions in images. Figure 19.7 shows a simple geometry that illustrates the problem. The point $\mathbf{P}$ projects to $\mathbf{x}_1$ in camera 1, and to $\mathbf{x}_2$ in camera 2. Notice because of the carefully chosen camera geometry, the y -coordinates of $\mathbf{x}_1$ and $\mathbf{x}_2$ are the same; only the x -coordinates differ. Write x_1 for the x -coordinate of $\mathbf{x}_1$; X for the x -coordinate of P , and so on. From the triangles in that figure, we have

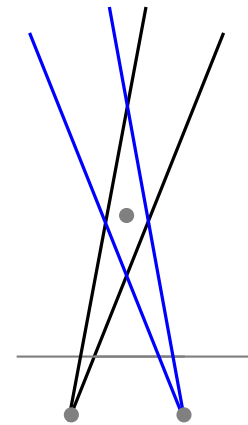
$$d = x_2 - x_1 = f \frac{(X - B) - X}{Z} = -f \frac{B}{Z}$$

meaning that as $\mathbf{P}$ gets further away, the *disparity* (difference between projected positions in left and right cameras) gets smaller, and so gets harder to measure.

Effect of baseline on stereo results

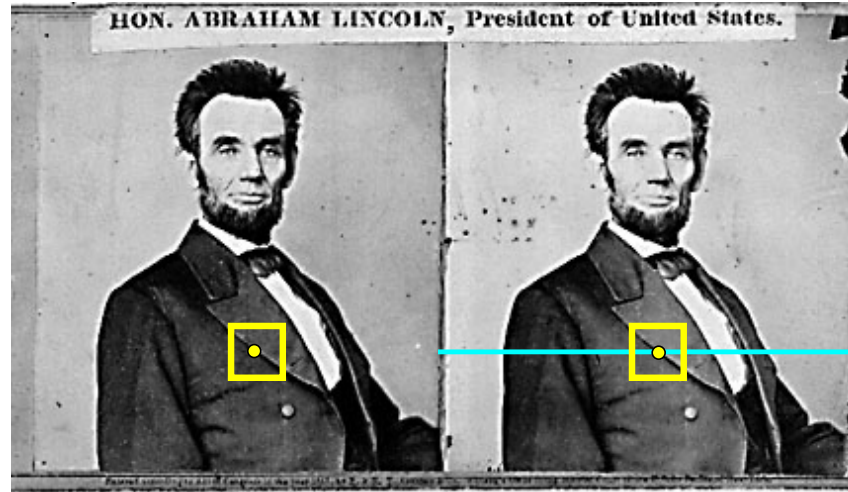


- Larger baseline
 - + Smaller triangulation error
 - Matching is more difficult



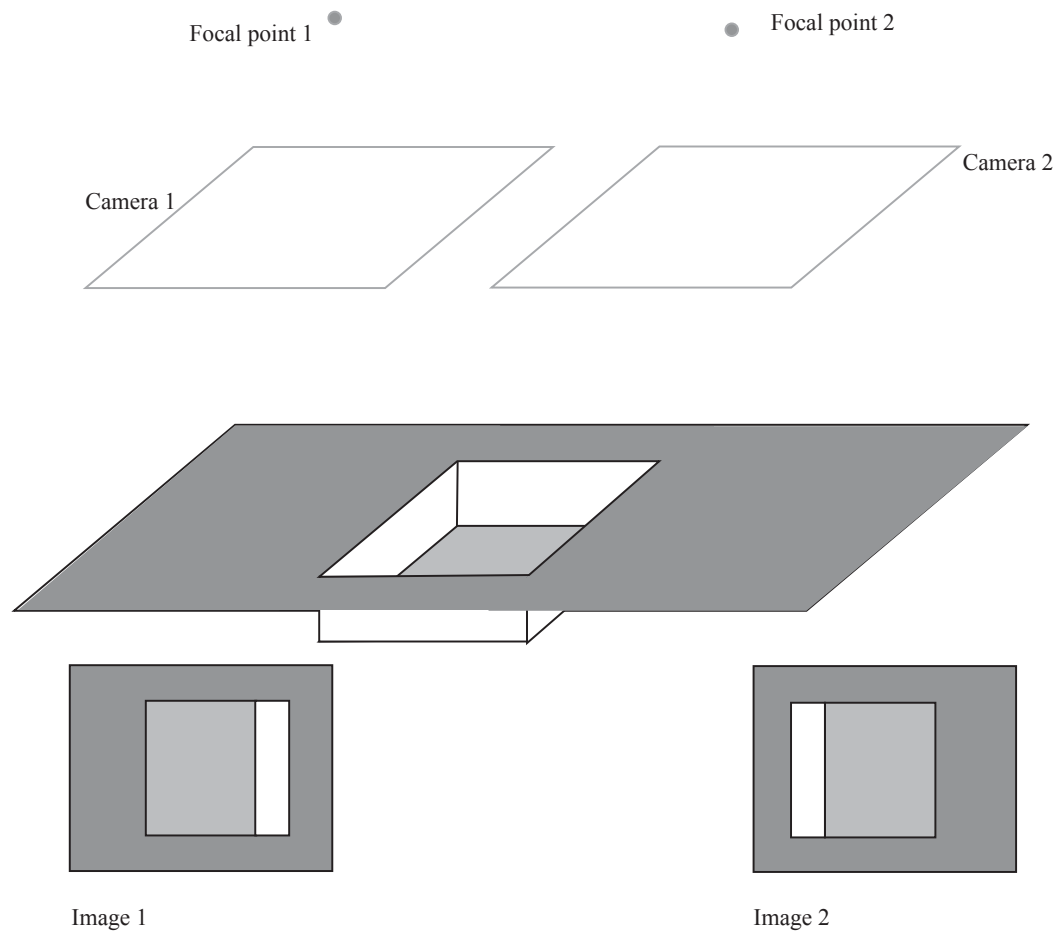
- Smaller baseline
 - Higher triangulation error
 - + Matching is easier

Basic stereo matching algorithm



- If necessary, rectify the two stereo images to transform epipolar lines into scanlines
- For each pixel x in the first image
 - Find corresponding epipolar scanline in the right image
 - Examine all pixels on the scanline and pick the best match x'
 - Compute disparity $x - x'$ and set $\text{depth}(x) = Bf/(x - x')$

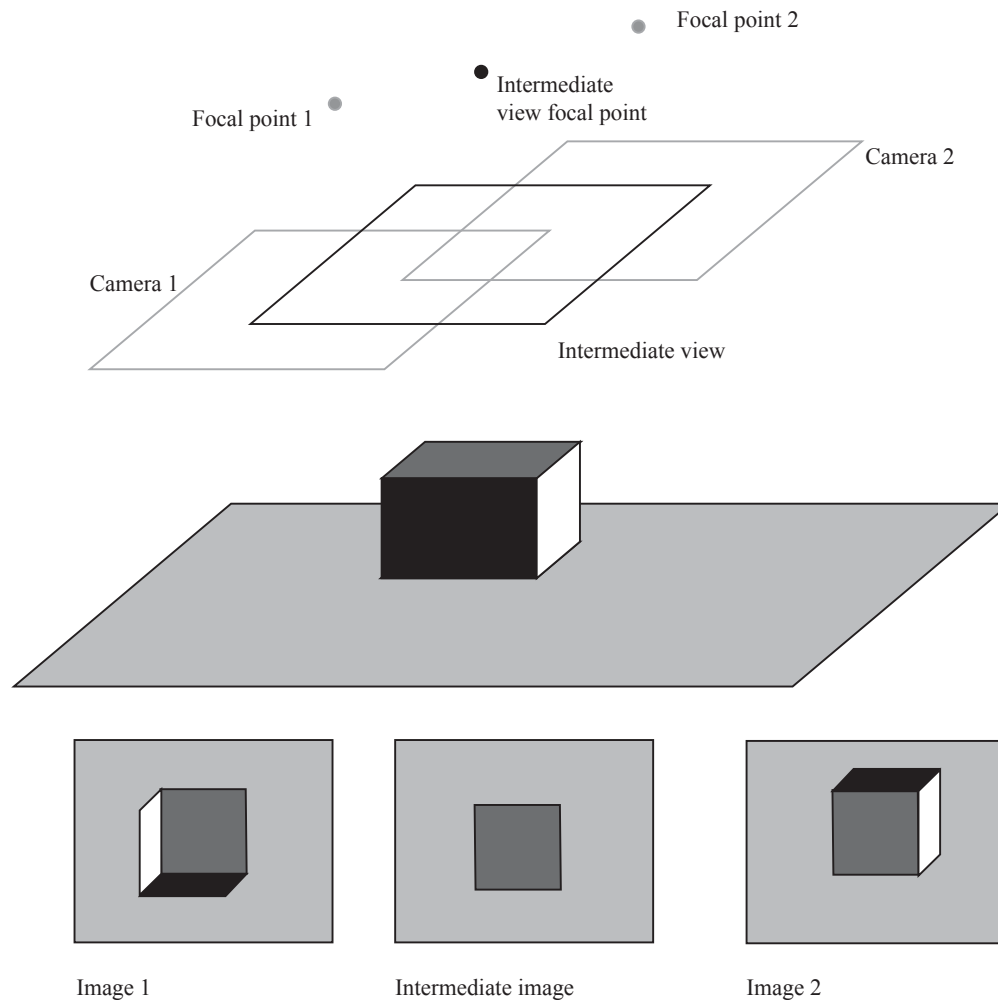
Some points don't have matches - I



This is a depth cue, though not much used directly

Effect sometimes known as Da Vinci stereopsis

Some points don't have matches - II



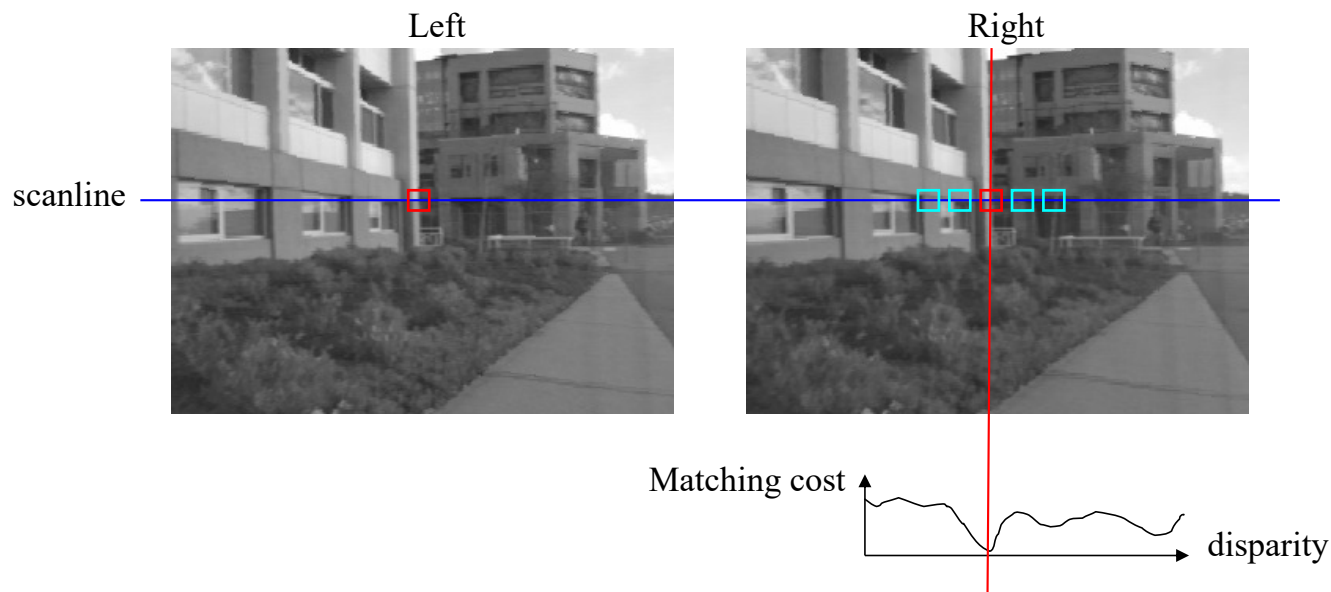
This is a depth cue, though not much used directly

Effect sometimes known as Da Vinci stereopsis

Outline

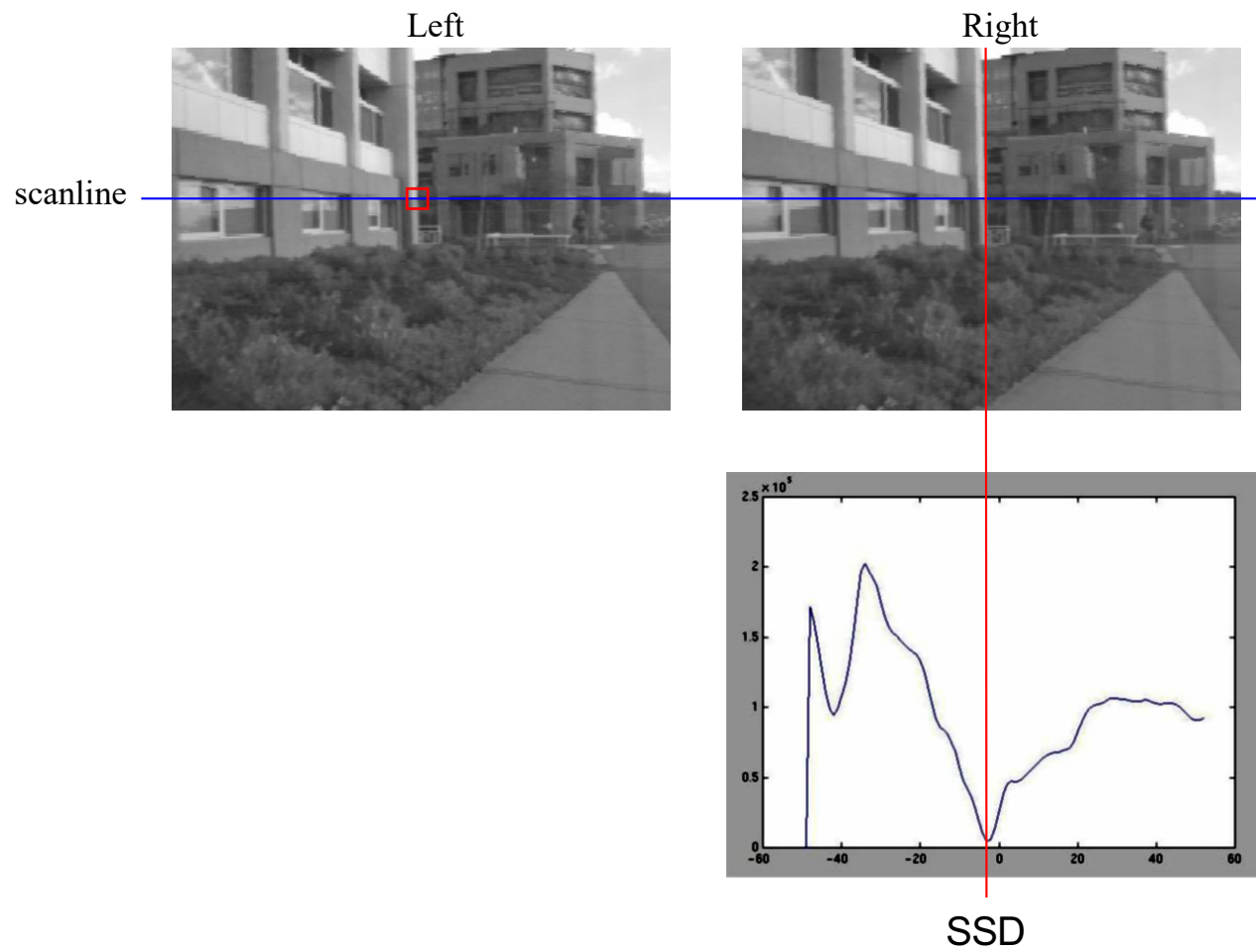
- Motivation and history
- Basic two-view stereo setup
- Local stereo matching algorithm

Correspondence search

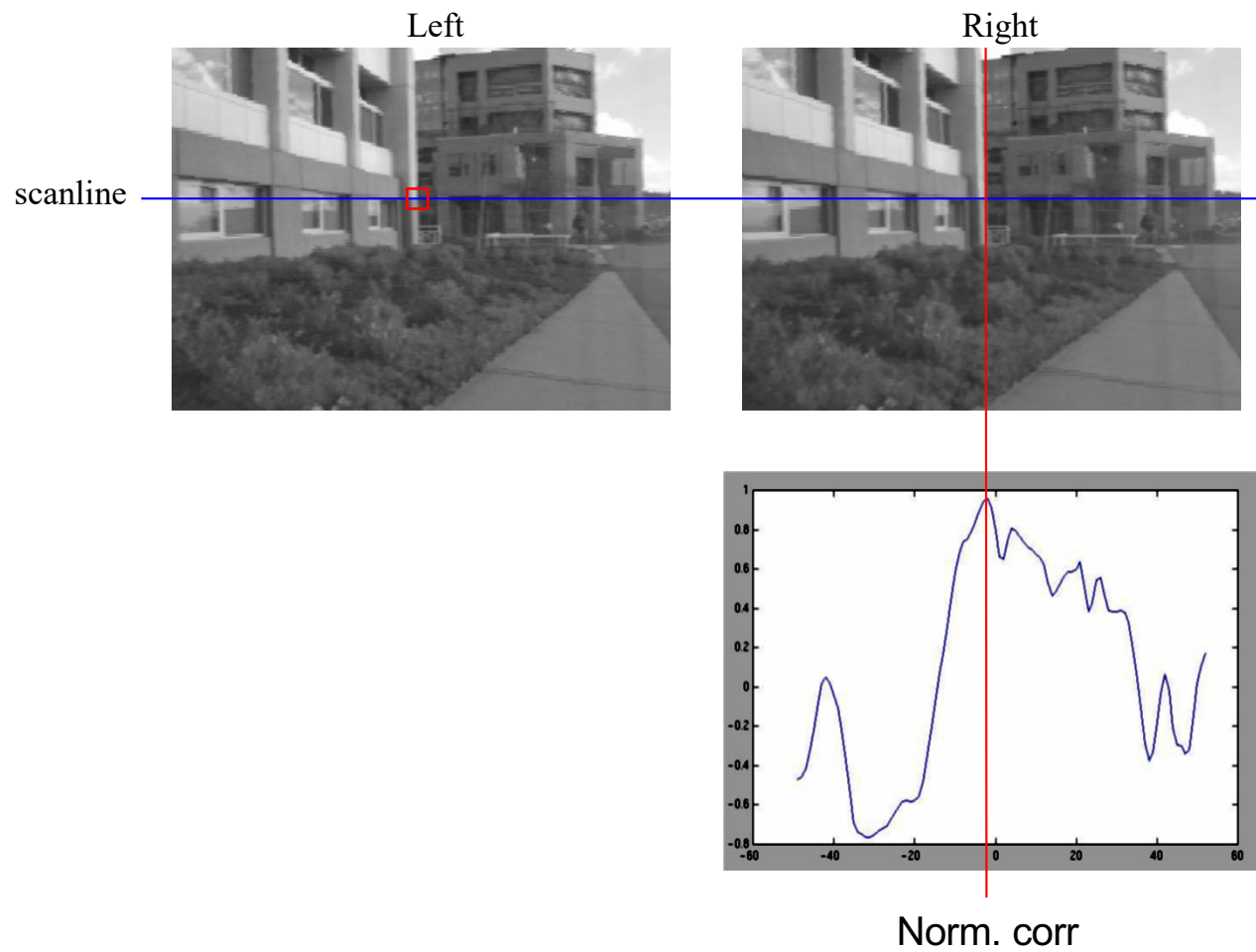


- Slide a window along the right scanline and compare contents of that window with the reference window in the left image
- Matching cost: SSD or normalized correlation

Correspondence search



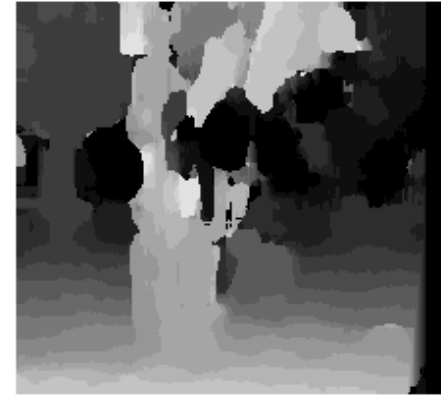
Correspondence search



Effect of window size on correspondence search



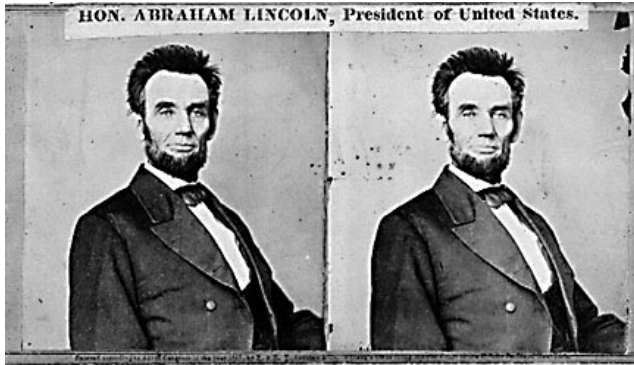
Window size 3



Window size 20

- Smaller window:
 - + More detail
 - More noise
- Larger window:
 - + Smoother disparity maps
 - Less detail

Where will basic window search fail?



Textureless surfaces



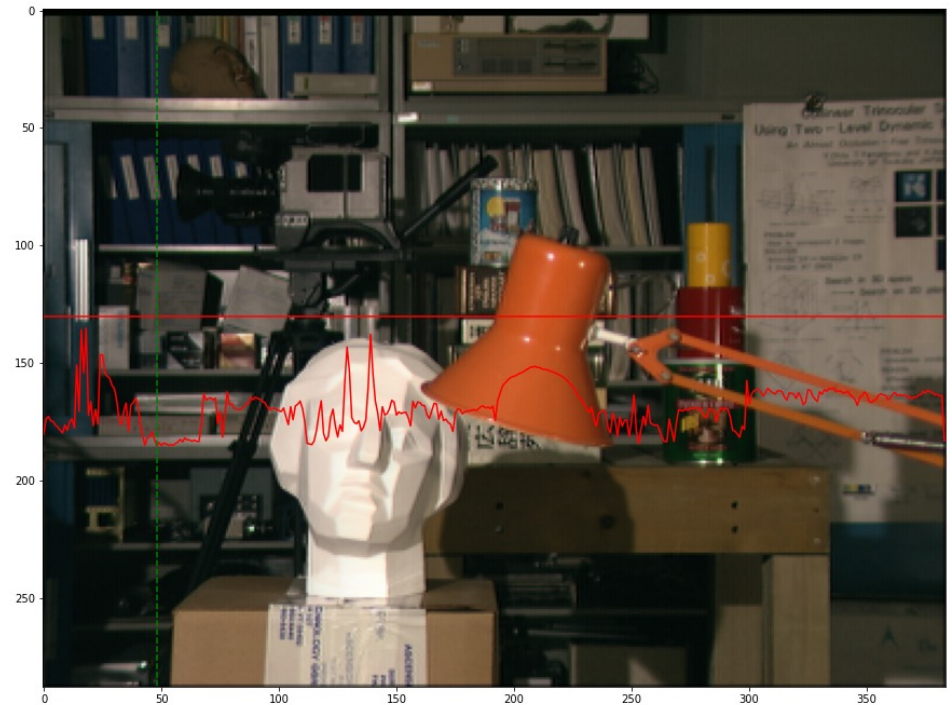
Occlusions, repetition



Non-Lambertian surfaces, specularities

Example: Textured neighborhood

Window size: 1 pixel



Source: [D. Hoiem](#)

Example: Textured neighborhood

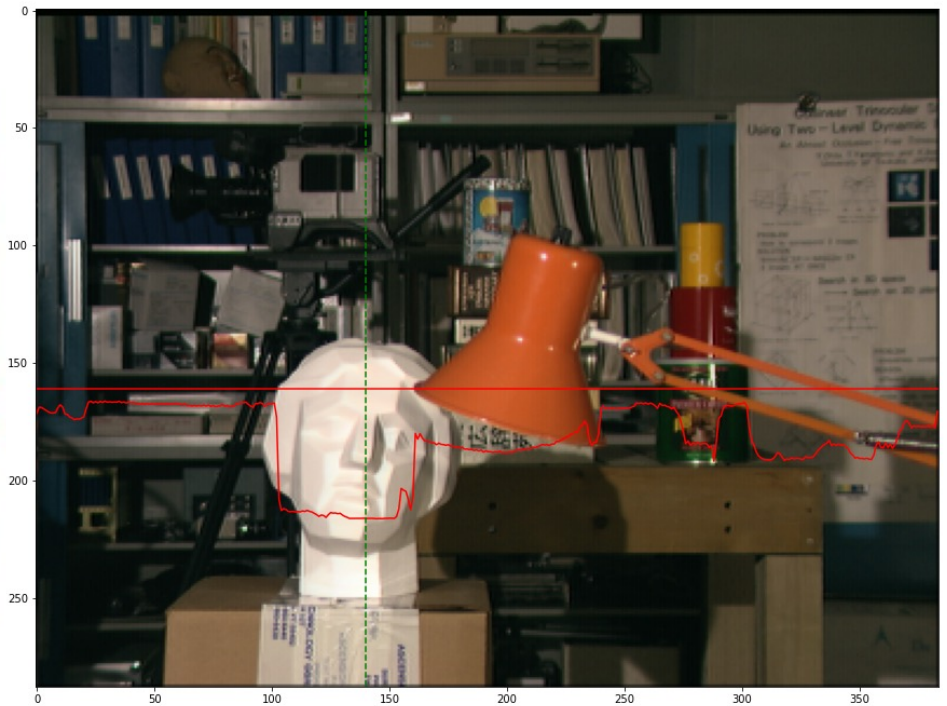
Window size: 7 pixels



Source: [D. Hoiem](#)

Example: Smooth neighborhood

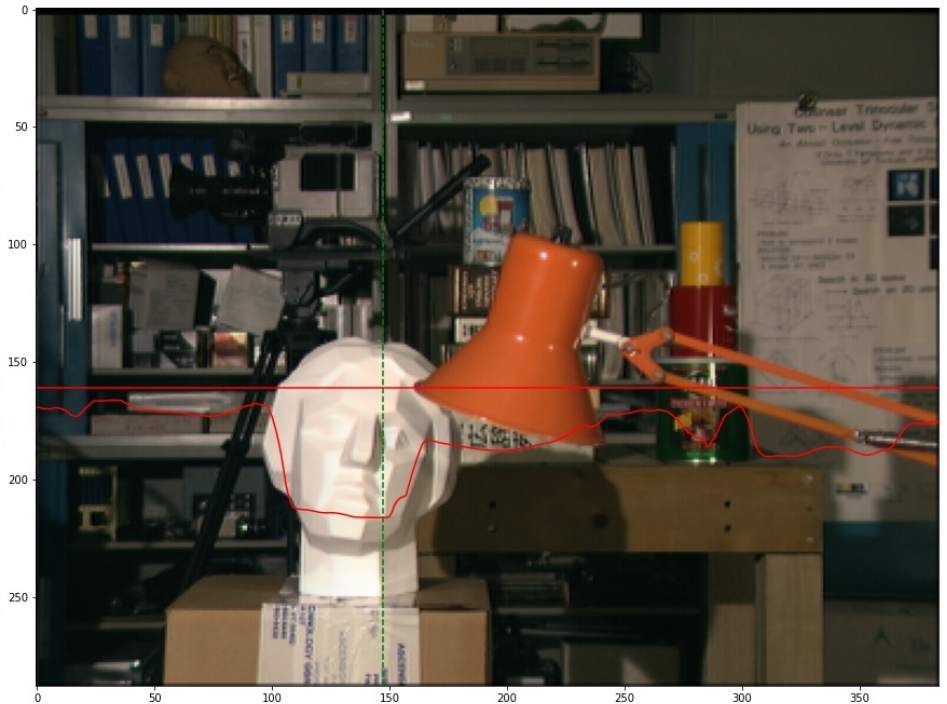
Window size: 1 pixel



Source: [D. Hoiem](#)

Example: Smooth neighborhood

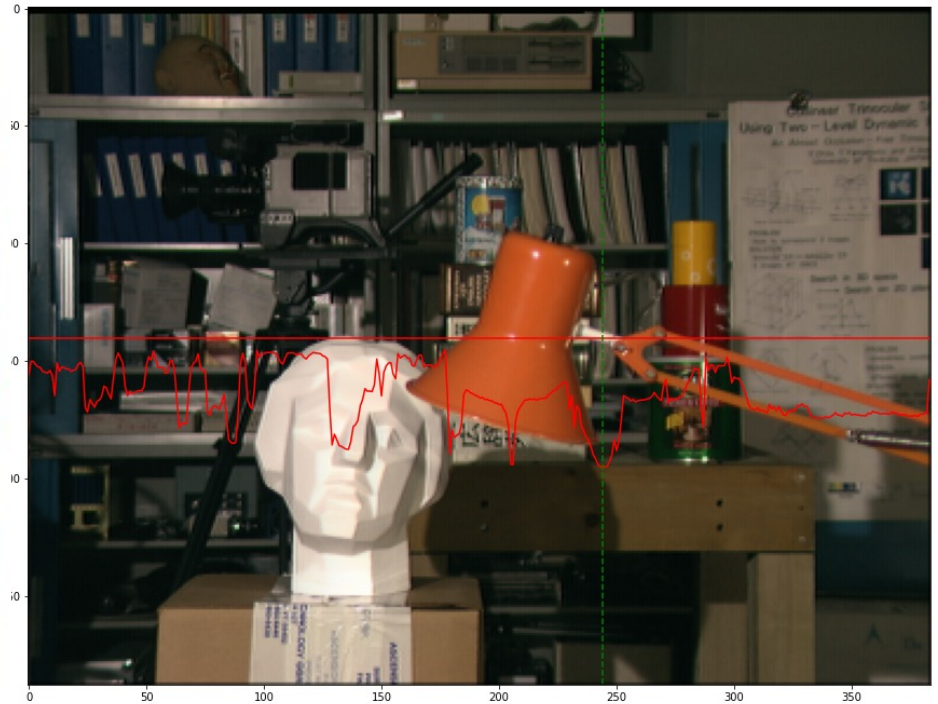
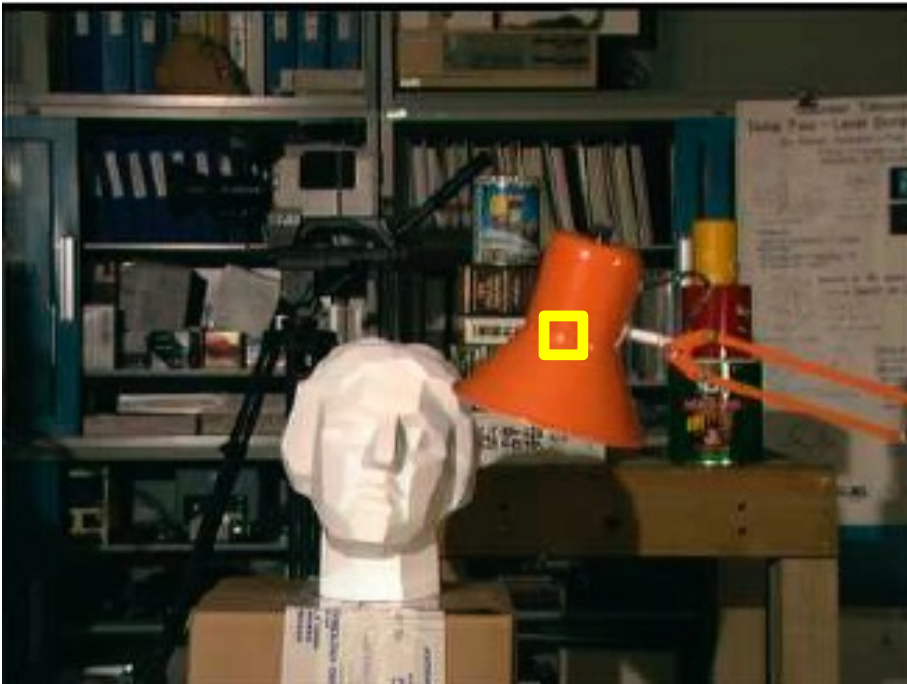
Window size: 7 pixels



Source: [D. Hoiem](#)

Example: Specular highlight

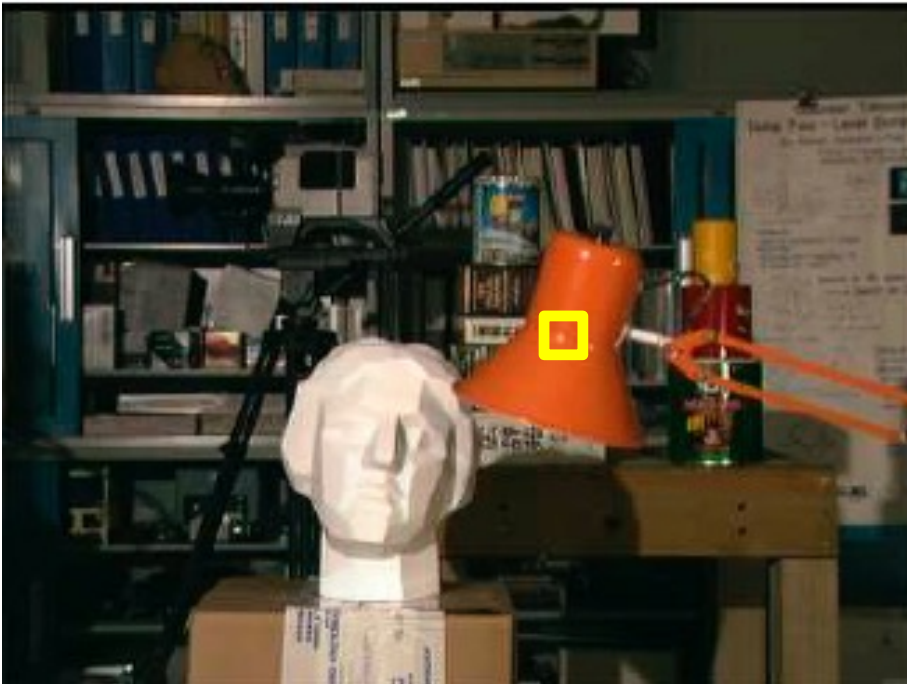
Window size: 1 pixel



Source: [D. Hoiem](#)

Example: Specular highlight

Window size: 7 pixels



Source: [D. Hoiem](#)

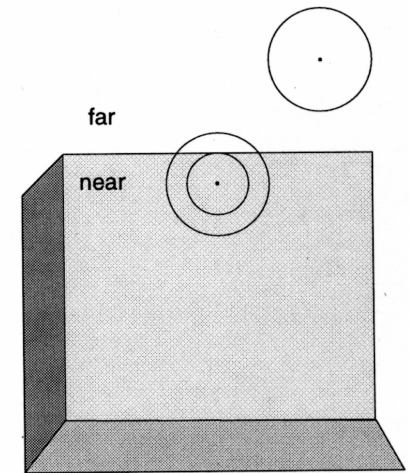
More interesting matching costs

Instead of SSD of pixel window, compute filter outputs for many filters

SSD those

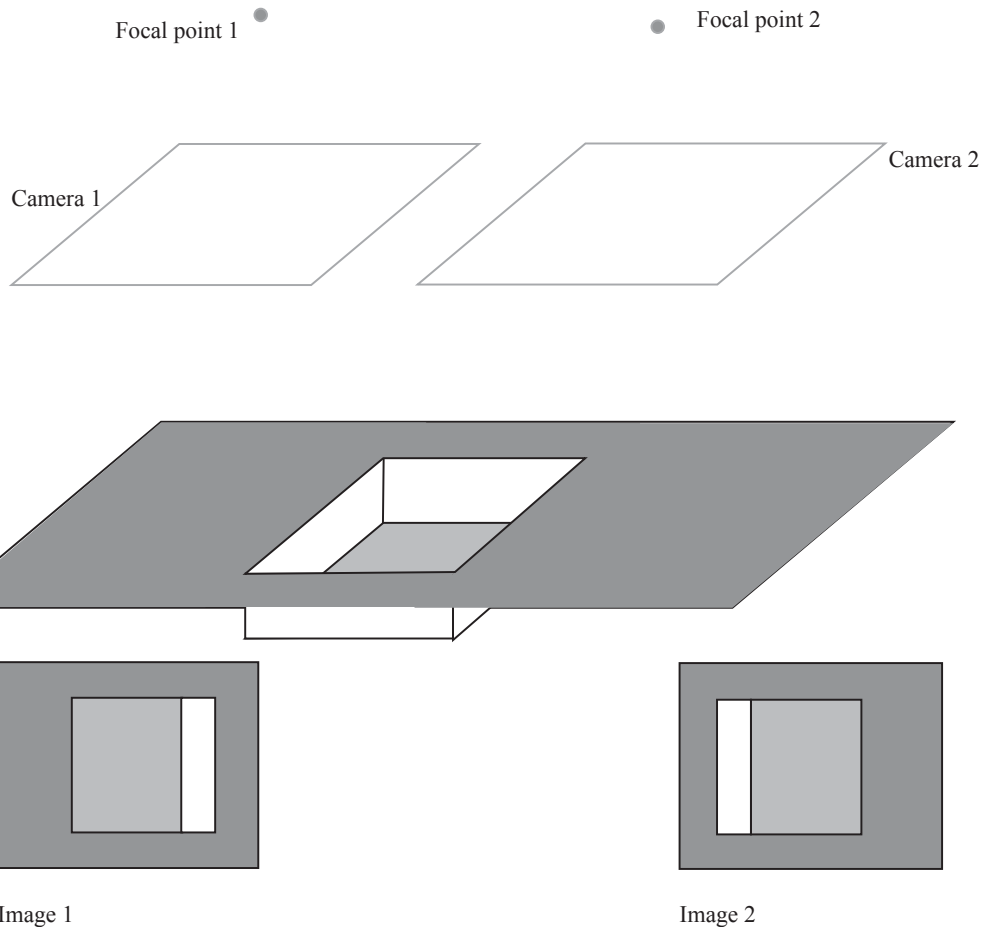
Advantage: more detailed description of points

Disadvantage: Filter support interacts badly with fast changes in depth



From Jones and Malik, "A computational framework for determining Stereo correspondences from a set of linear spatial filters"

Dealing with Da Vinci



Consequence:

some points in L (R) image do not have matches in R (L) image

Note asymmetry in previous algorithm
(find best matching point in other side - what if best matches aren't consistent?)

Strategy:

match L to R

match R to L

use only consistent matches

Outline

- Motivation and history
- Basic two-view stereo setup
- Local stereo matching algorithm
- **Beyond local stereo matching**

Stereo as optimization with non-local constraints

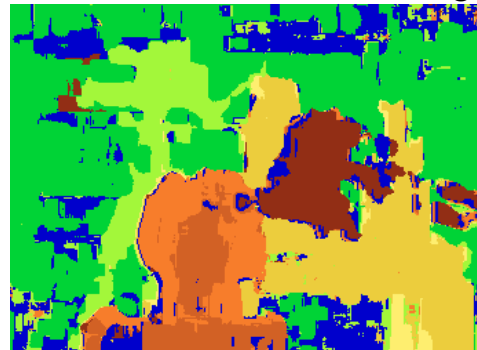
Data



Ground truth



Window-based matching



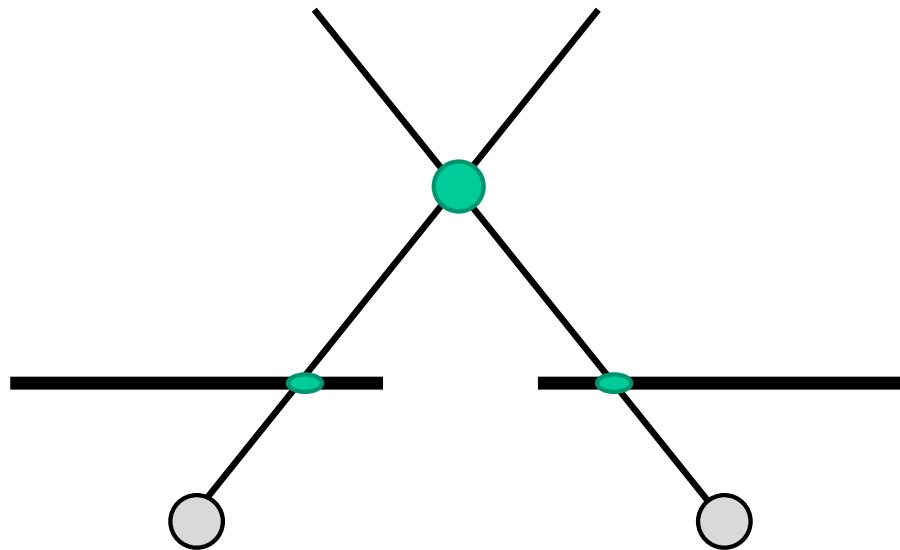
Global optimization method (graph cuts)



Y. Boykov, O. Veksler, and R. Zabih, [Fast Approximate Energy Minimization via Graph Cuts](#), PAMI 2001

Non-local constraint: Uniqueness

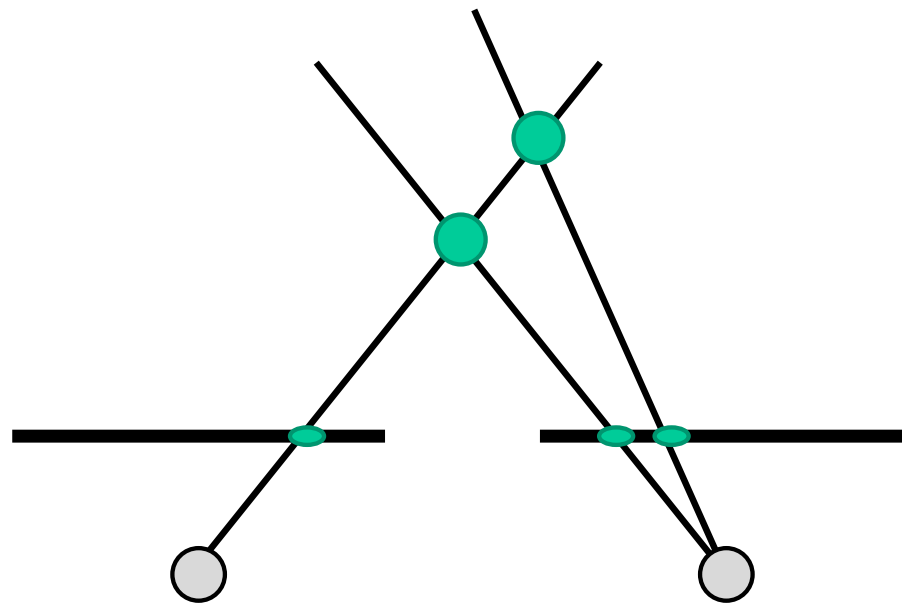
- Each point in one image should match at most one point in the other image
- Does uniqueness always hold in real life?



Source: [J. Johnson and D. Fouhey](#)

Non-local constraint: Uniqueness

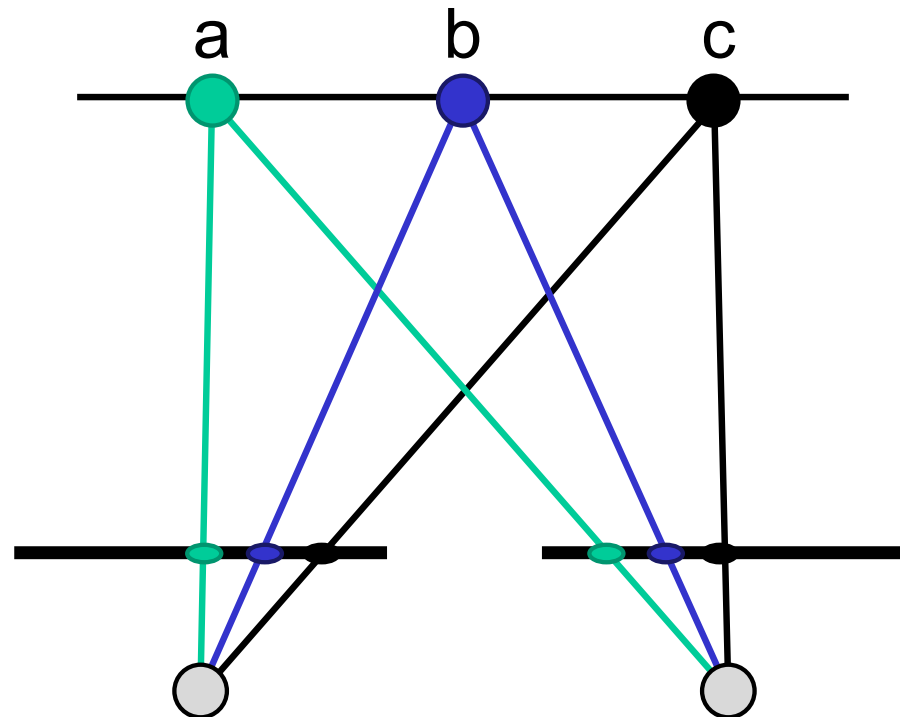
- Each point in one image should match at most one point in the other image
- Does uniqueness always hold in real life?



Source: [J. Johnson and D. Fouhey](#)

Non-local constraint: Ordering

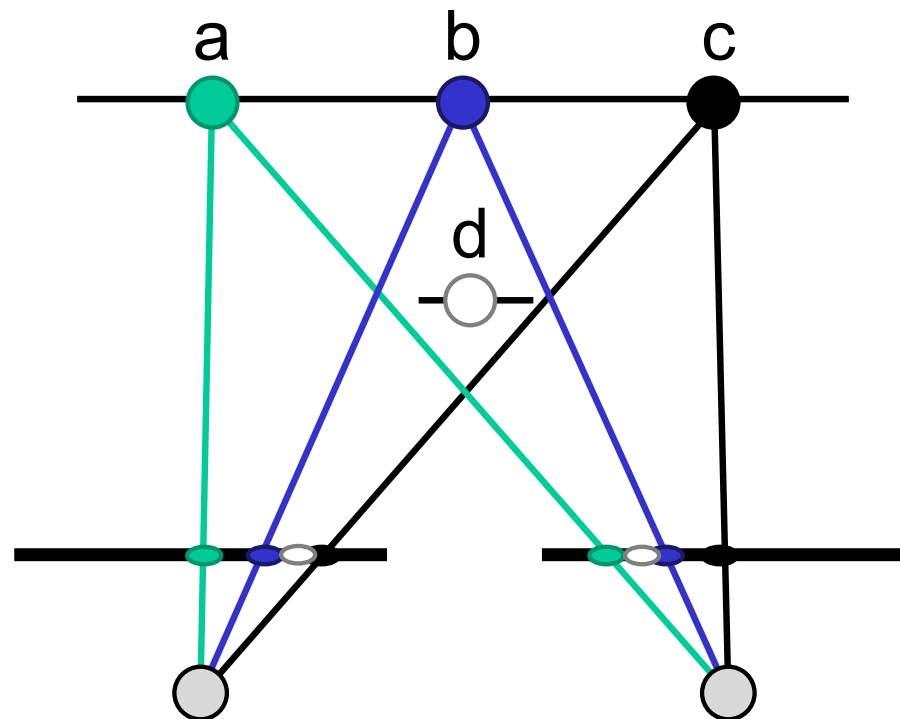
- Corresponding points should appear in the same order
- Is ordering always preserved in real life?



Source: [J. Johnson and D. Fouhey](#)

Non-local constraint: Ordering

- Corresponding points should appear in the same order
- Is ordering always preserved in real life?



Source: [J. Johnson and D. Fouhey](#)

Non-local constraint: Smoothness

- We expect disparity values to change slowly (for the most part)

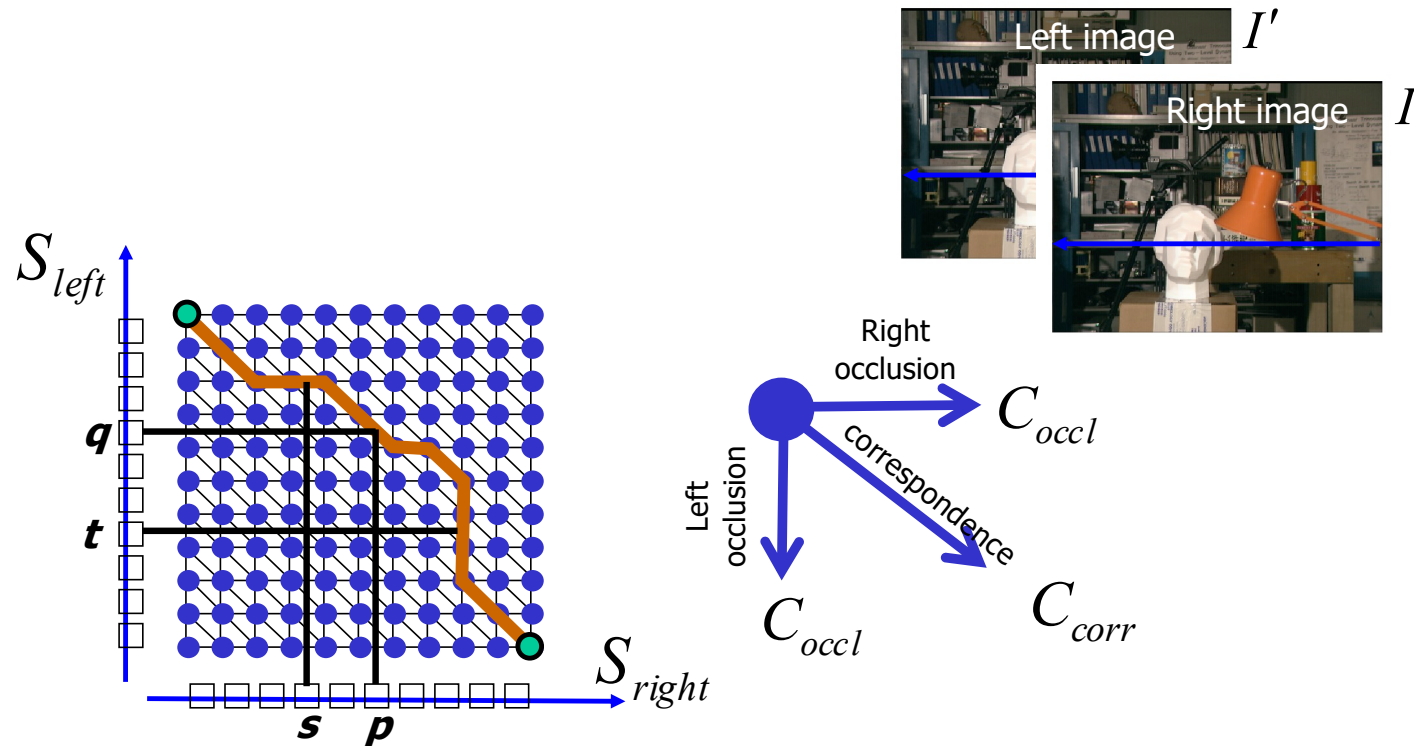


Scanline stereo by dynamic programming

- Match pixels along the entire scanline while preserving uniqueness and ordering
- Different scanlines are still optimized independently



Scanline stereo by dynamic programming

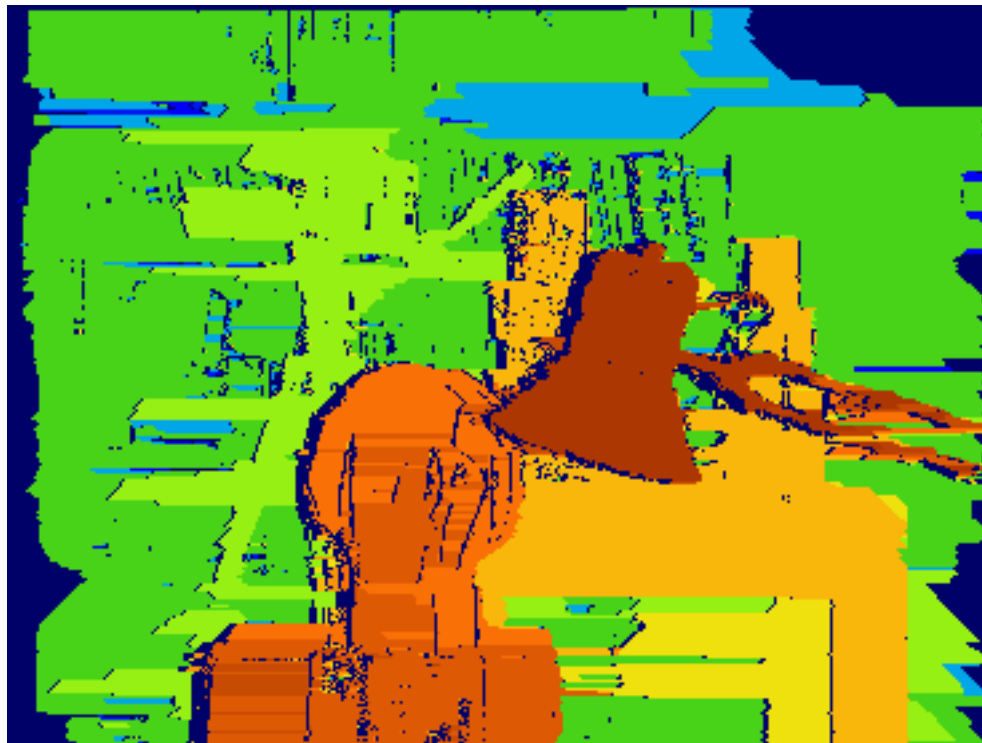


Source: Y. Boykov

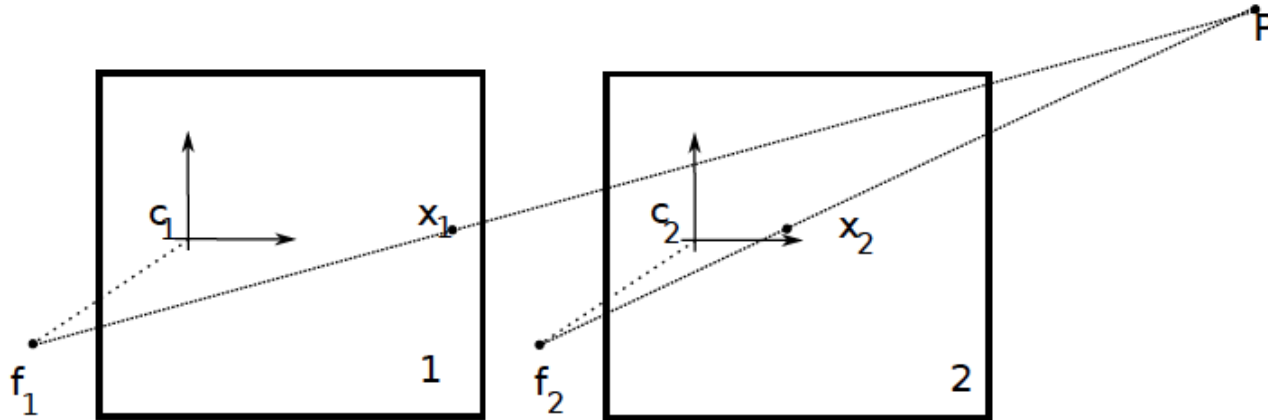
Y. Ohta and T. Kanade. [Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming](#). IEEE Trans. PAMI, 1985

Scanline stereo by dynamic programming

- Generates streaking artifacts!



Stereo as an optimization problem



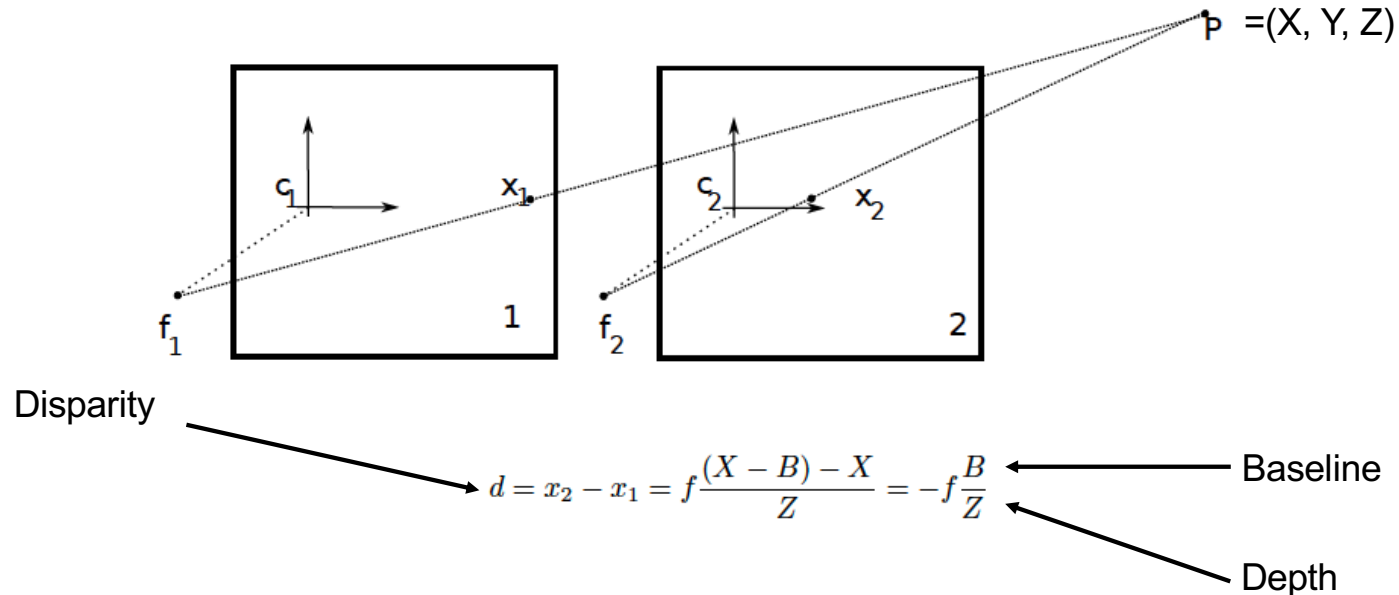
Originally:

find x_1 , x_2 that match, compute depth

Now:

choose depth at x_1 that causes x_2 to be best match

Stereo as an optimization problem



Assume camera 2 is translated wrt camera 1, both are calibrated

IF you choose the right depth for x_1 , then:

you know disparity, so you know x_2

and you can optimize $\|color(x_1) - color(x_2(d))\|^2 + smoothness(depth)$
or something like it

Stereo as an optimization problem (sketch)

Quantize depth to a fixed number of levels (say, 256)

Encode depth at every pixel with a one-hot vector (say, 256 d)

$$\mathbf{w}_1 = \begin{pmatrix} \|\mathbf{c}(x_1) - \mathbf{c}(x_1 + \delta_1)\|^2 \\ \|\mathbf{c}(x_1) - \mathbf{c}(x_1 + \delta_2)\|^2 \\ \|\mathbf{c}(x_1) - \mathbf{c}(x_1 + \delta_3)\|^2 \\ \dots \\ \|\mathbf{c}(x_1) - \mathbf{c}(x_1 + \delta_{256})\|^2 \end{pmatrix}$$

$$\mathbf{d}(x_1) = \mathbf{d}_1 = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 1 \\ 0 \\ 0 \\ \dots \\ 0 \\ 0 \end{pmatrix}$$

$$\text{Cost}(\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \dots) = \mathbf{w}_1^T \mathbf{d}_1 + \mathbf{w}_2 \mathbf{d}_2 + \dots + \text{smoothness term}$$

Stereo as an optimization problem (sketch)

Quantize depth to a fixed number of levels (say, 256)

Encode depth at every pixel with a one-hot vector (say, 256 d)

$$\text{Cost}(\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \dots) = \mathbf{w}_1^T \mathbf{d}_1 + \mathbf{w}_2 \mathbf{d}_2 + \dots + \text{smoothness term}$$

Typically: cheap if $\text{depth}(x_1) = \text{depth}(\text{neighbors})$
expensive otherwise

EG

$$N - \sum_{i \in \text{neighbors}} \mathbf{d}_1^T \mathbf{d}_j$$

Stereo as an optimization problem (sketch)

Result:

large discrete optimization problem, at least NP-hard

There are excellent approximation algorithms (below)

For a small set of cases, true optimum is known (by good luck)

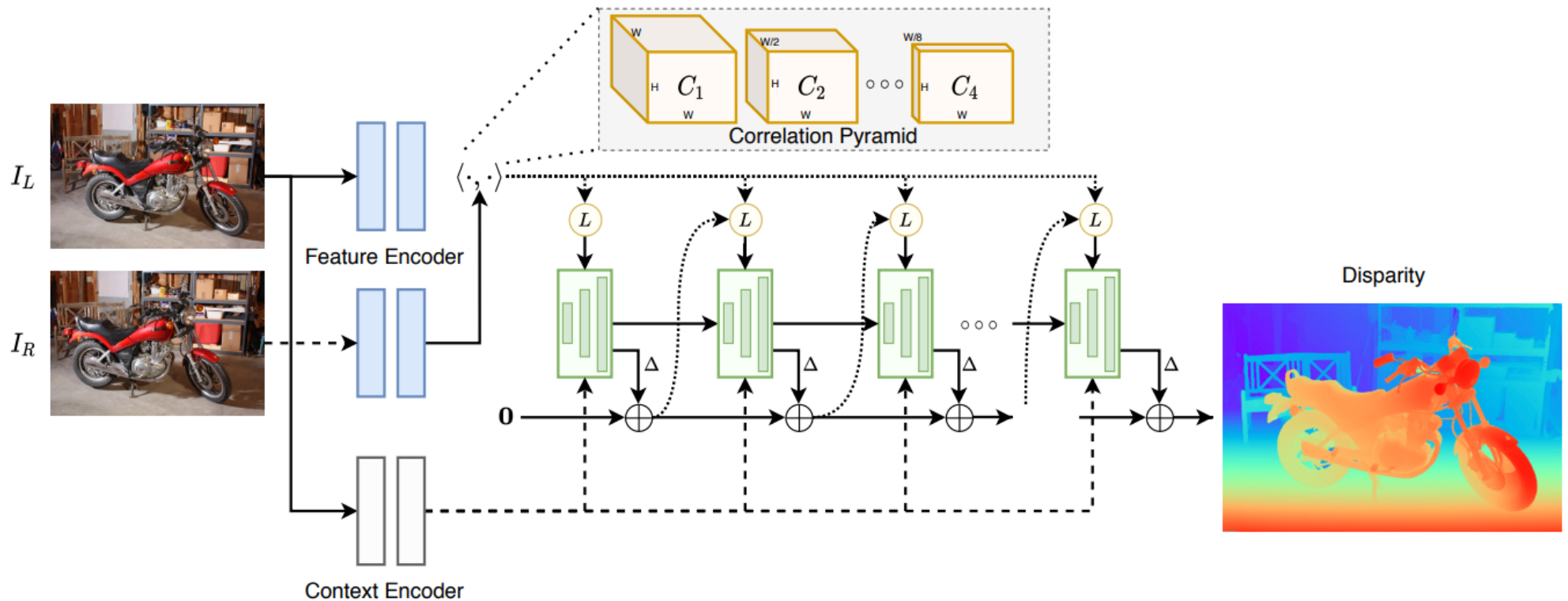
Yield:

excellent stereo algorithms

Stereo as an optimization problem (sketch)

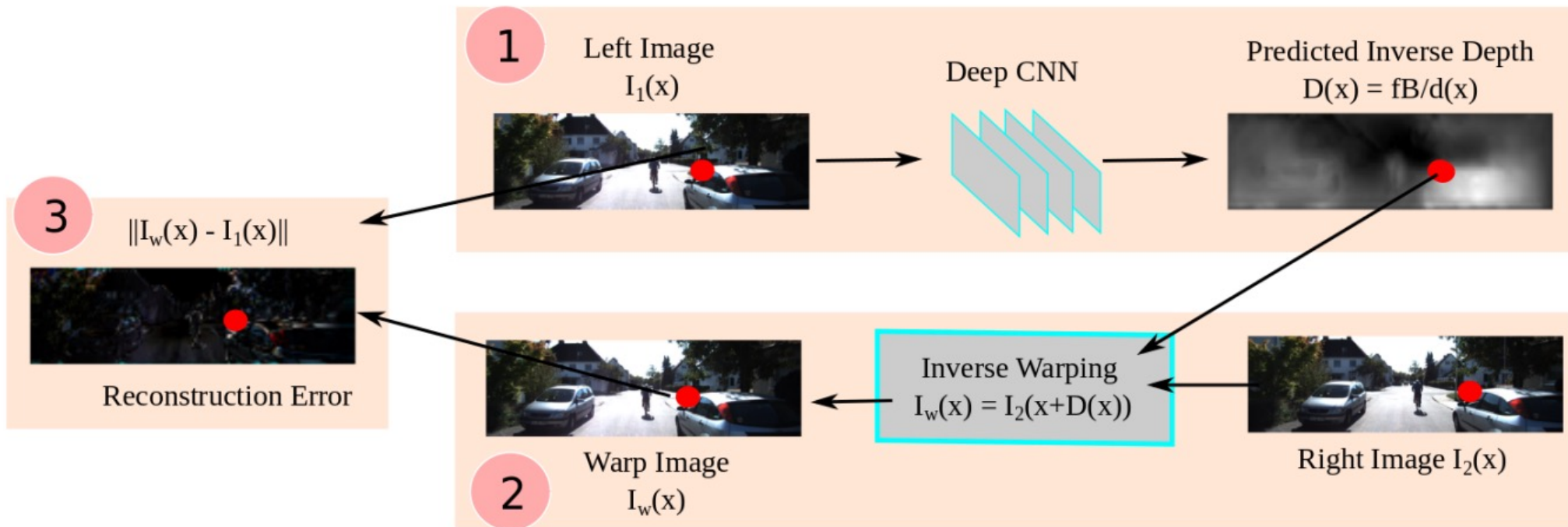


Stereo matching with deep networks



L. Lipson et al. [RAFT-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching](#). arXiv 2021

Self-supervised depth estimation



Stereo datasets

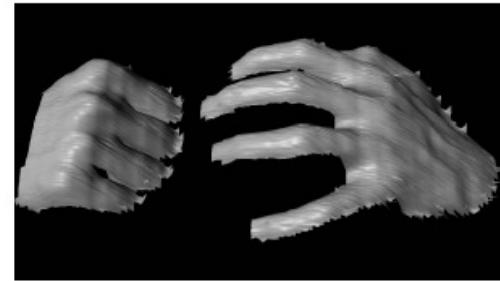
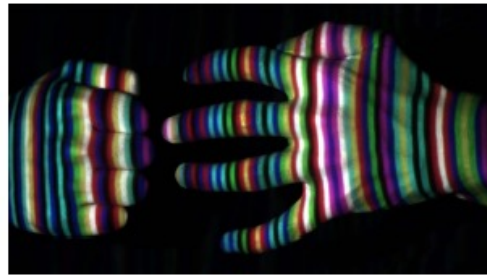
- [Middlebury stereo datasets](#)
- [KITTI](#)
- [Synthetic data](#)



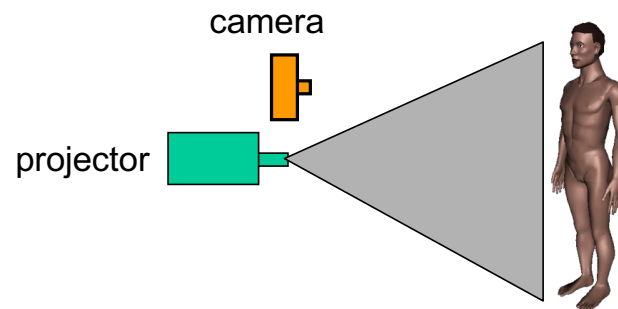
Outline

- Motivation and history
- Basic two-view stereo setup
- Local stereo matching algorithm
- Stereo with non-local optimization
- Active stereo with structured light

Active stereo with structured light

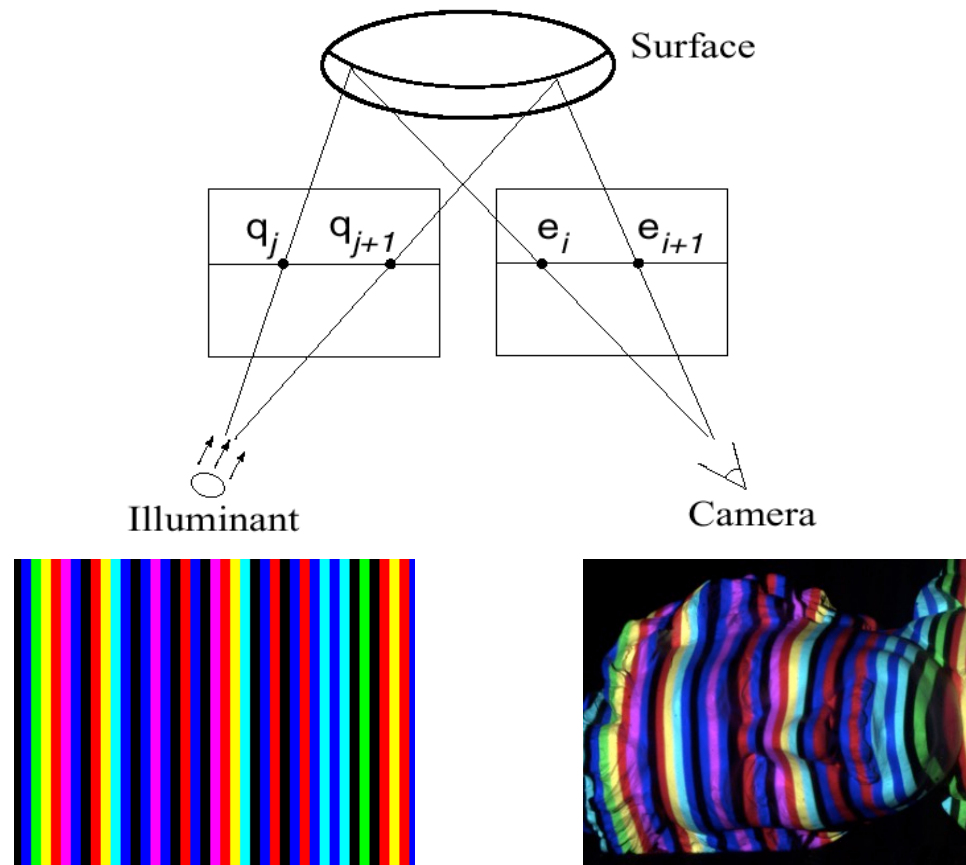


- Project “structured” light patterns onto the object
 - Simplifies the correspondence problem
 - Allows us to use only one camera



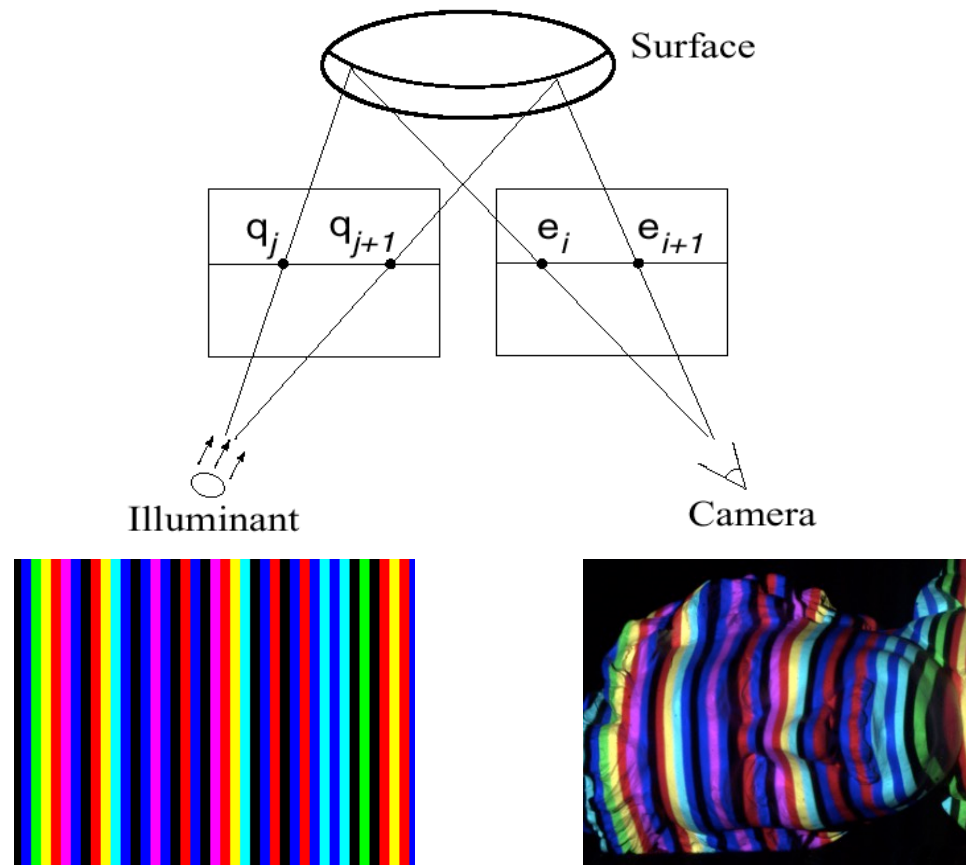
L. Zhang, B. Curless, and S. M. Seitz. [Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming](#). 3DPVT 2002

Active stereo with structured light



L. Zhang, B. Curless, and S. M. Seitz. [Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming](#). *3DPVT* 2002

Active stereo with structured light



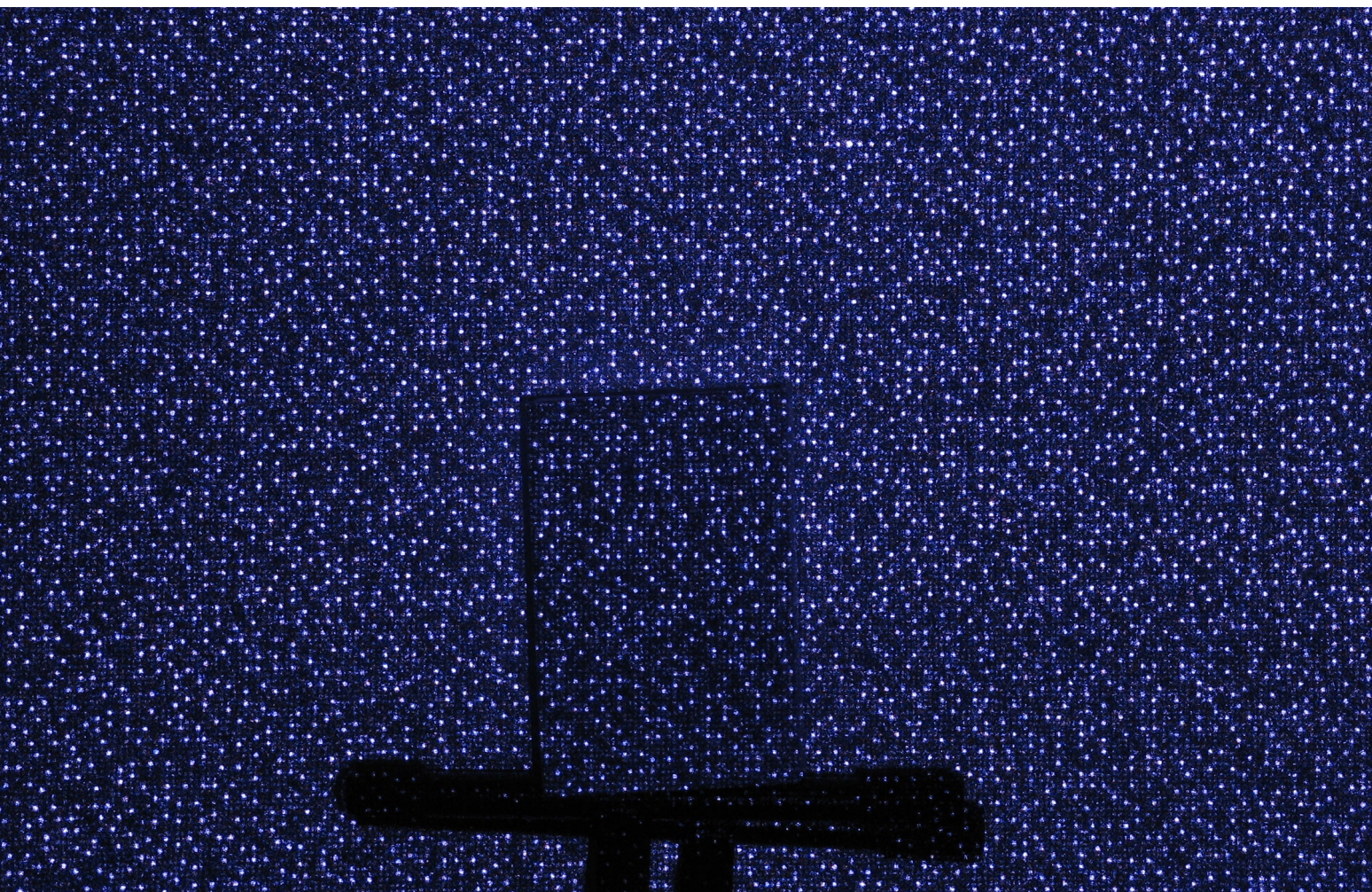
http://en.wikipedia.org/wiki/Structured-light_3D_scanner

Kinect: Structured infrared light



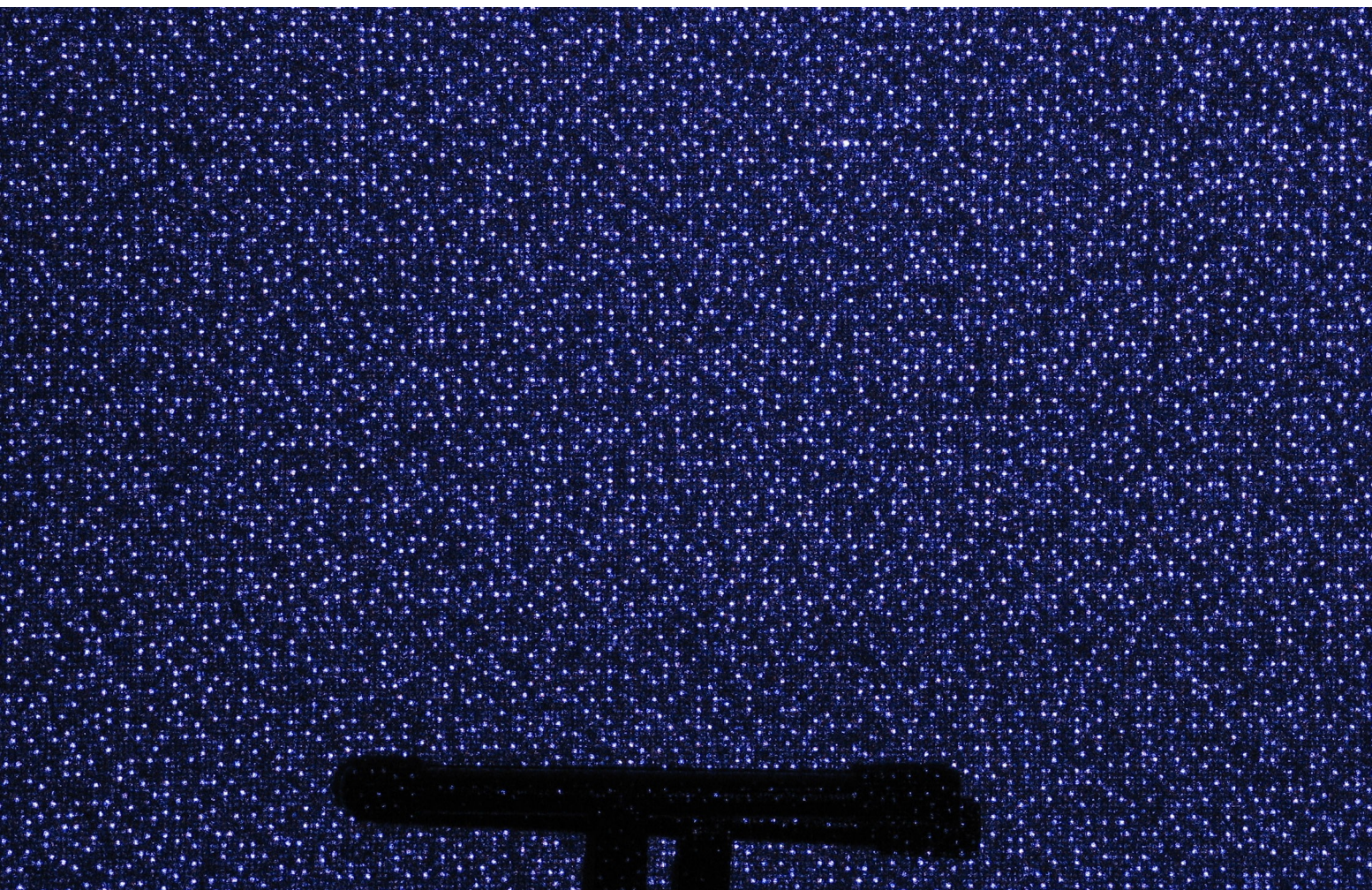
<http://bbzippo.wordpress.com/2010/11/28/kinect-in-infrared/>

Example: Book vs. No Book



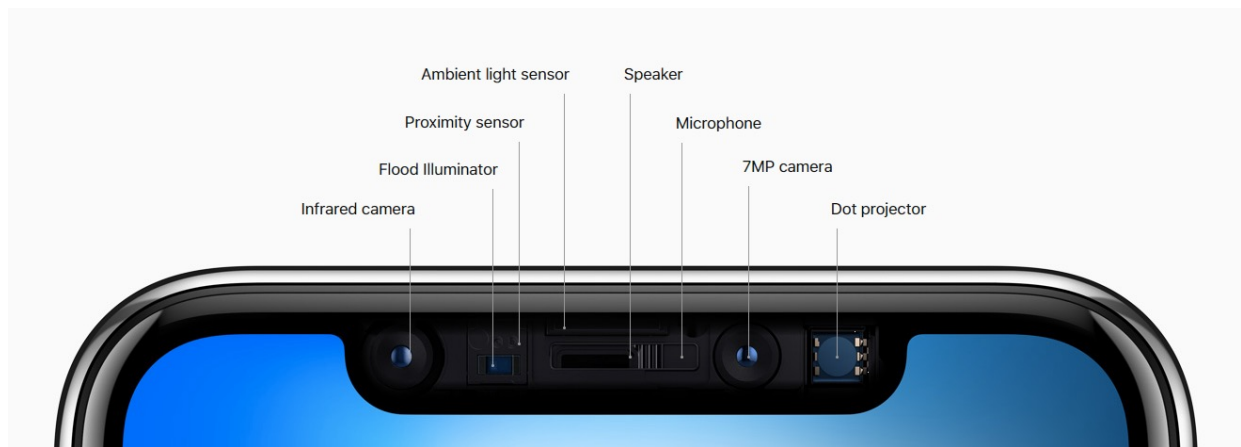
[Source](#) (via D. Hoiem)

Example: Book vs. No Book



[Source](#) (via D. Hoiem)

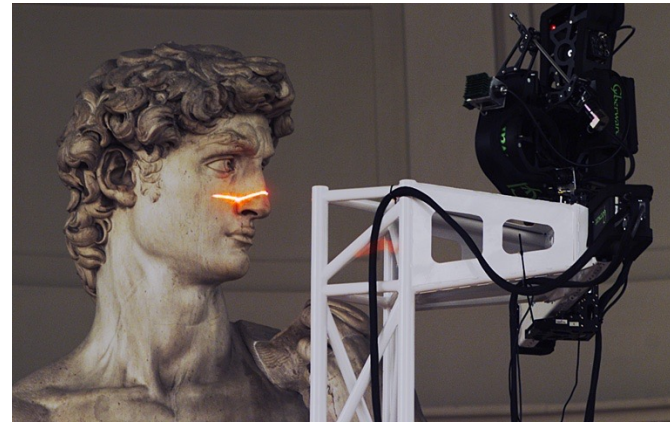
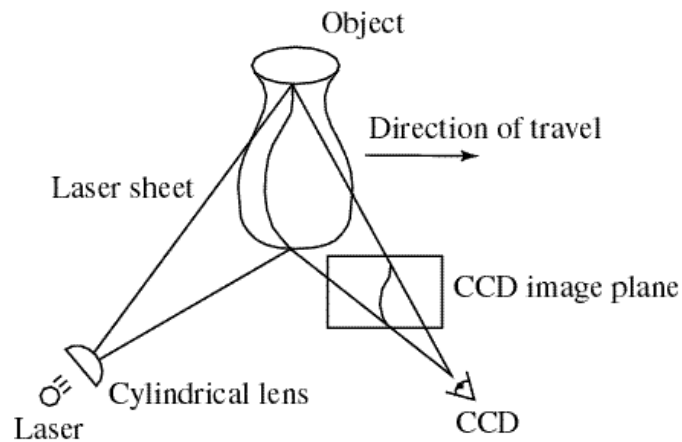
Apple TrueDepth



<https://www.cnet.com/news/apple-face-id-truedepth-how-it-works/>



Laser scanning



Digital Michelangelo Project
Levoy et al.

<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

Source: S. Seitz

Laser scanned models



The Digital Michelangelo Project, Levoy et al.

Source: S. Seitz

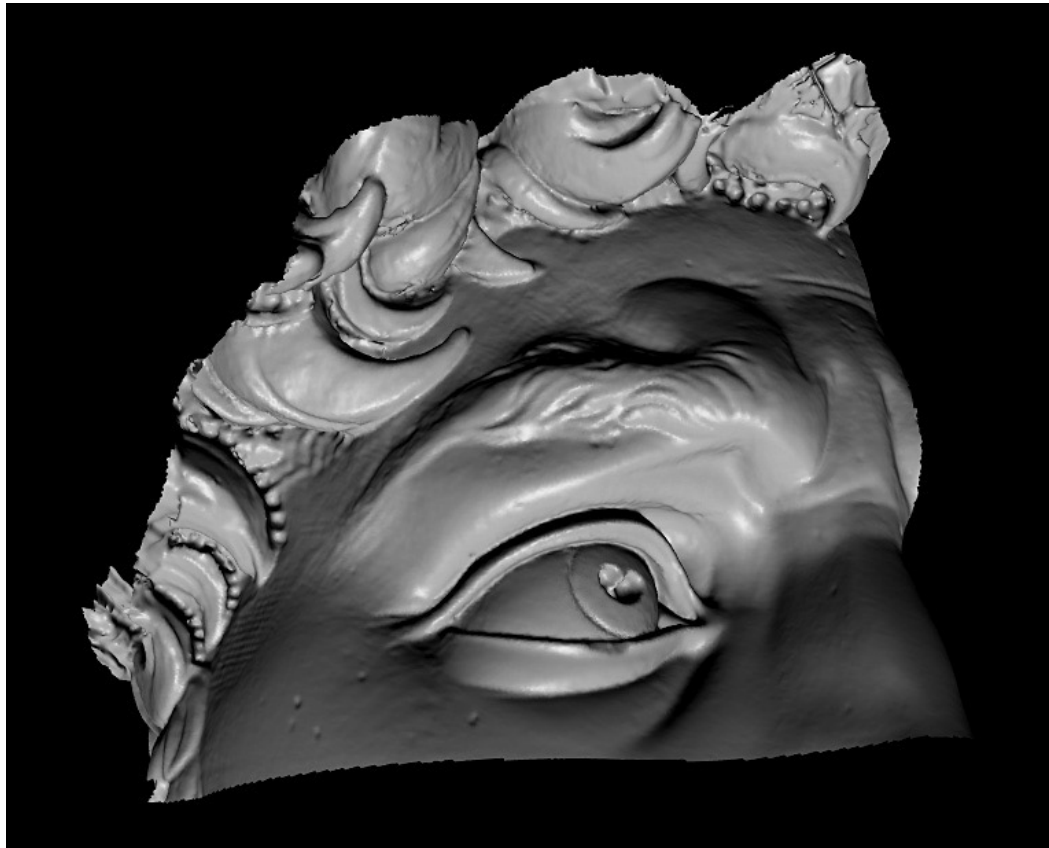
Laser scanned models



The Digital Michelangelo Project, Levoy et al.

Source: S. Seitz

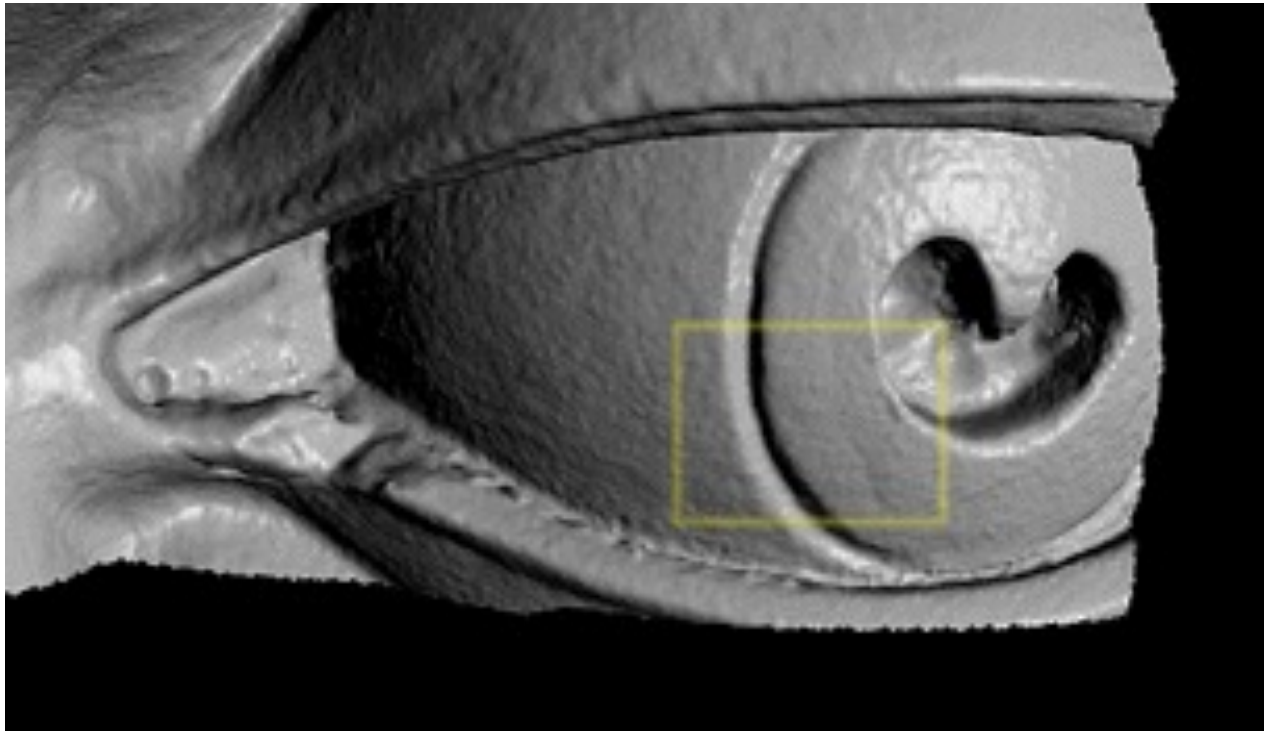
Laser scanned models



The Digital Michelangelo Project, Levoy et al.

Source: S. Seitz

Laser scanned models

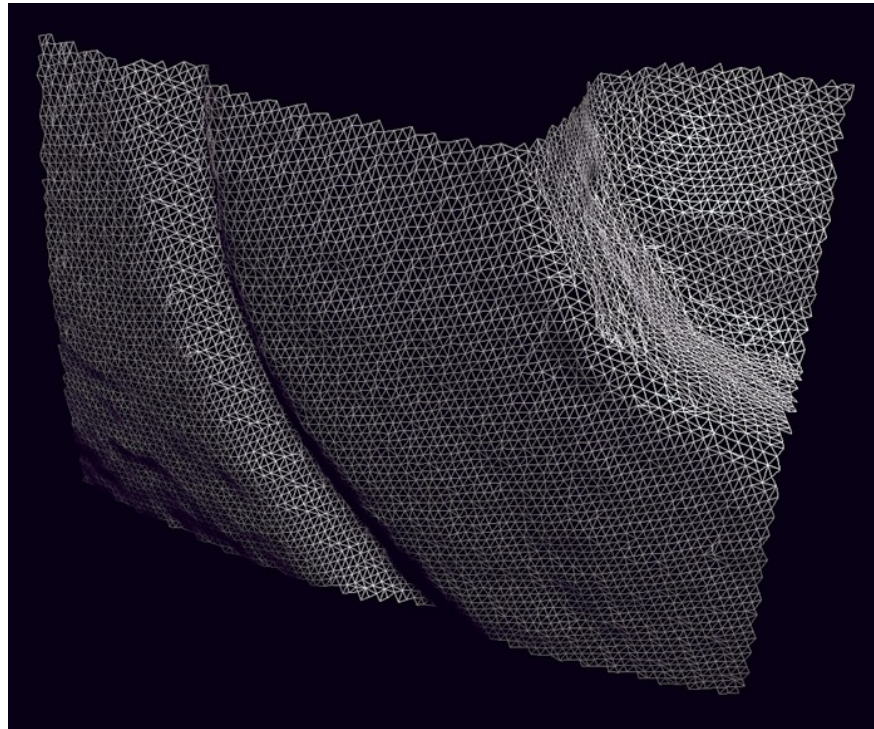


The Digital Michelangelo Project, Levoy et al.

Source: S. Seitz

Laser scanned models

1.0 mm resolution (56 million triangles)

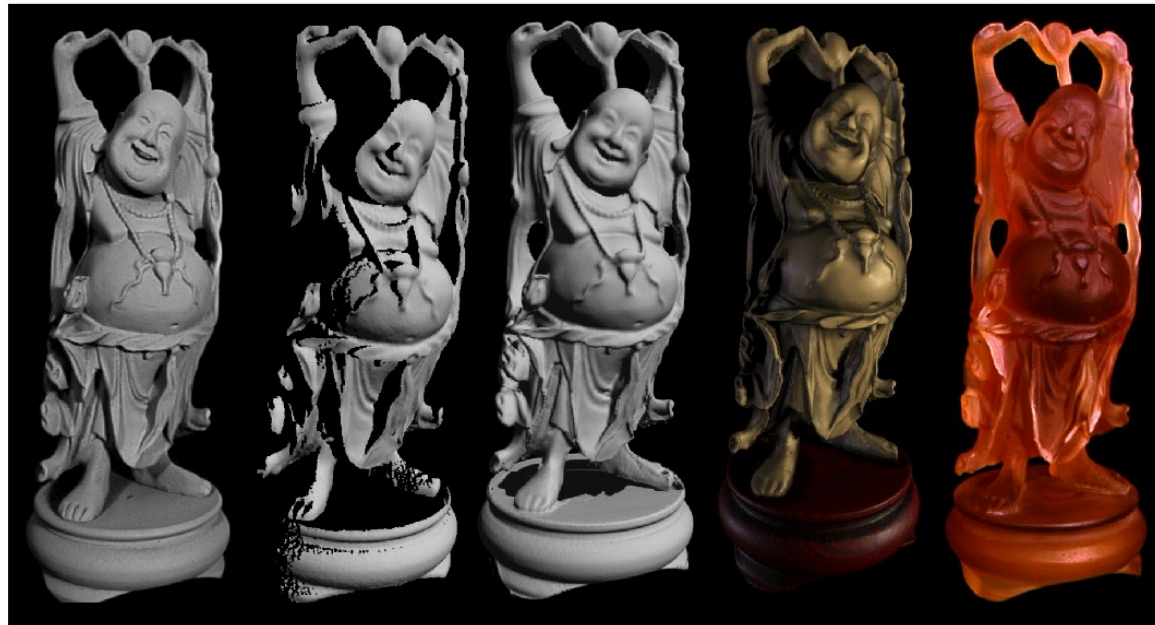


The Digital Michelangelo Project, Levoy et al.

Source: S. Seitz

Aligning range images

- A single range scan is not sufficient to capture a complex surface
- Need techniques to register multiple range images



B. Curless and M. Levoy, [A Volumetric Method for Building Complex Models from Range Images](#),
SIGGRAPH 1996

Aligning range images

- A single range scan is not sufficient to capture a complex surface
- Need techniques to register multiple range images

... which brings us to *multi-view stereo*