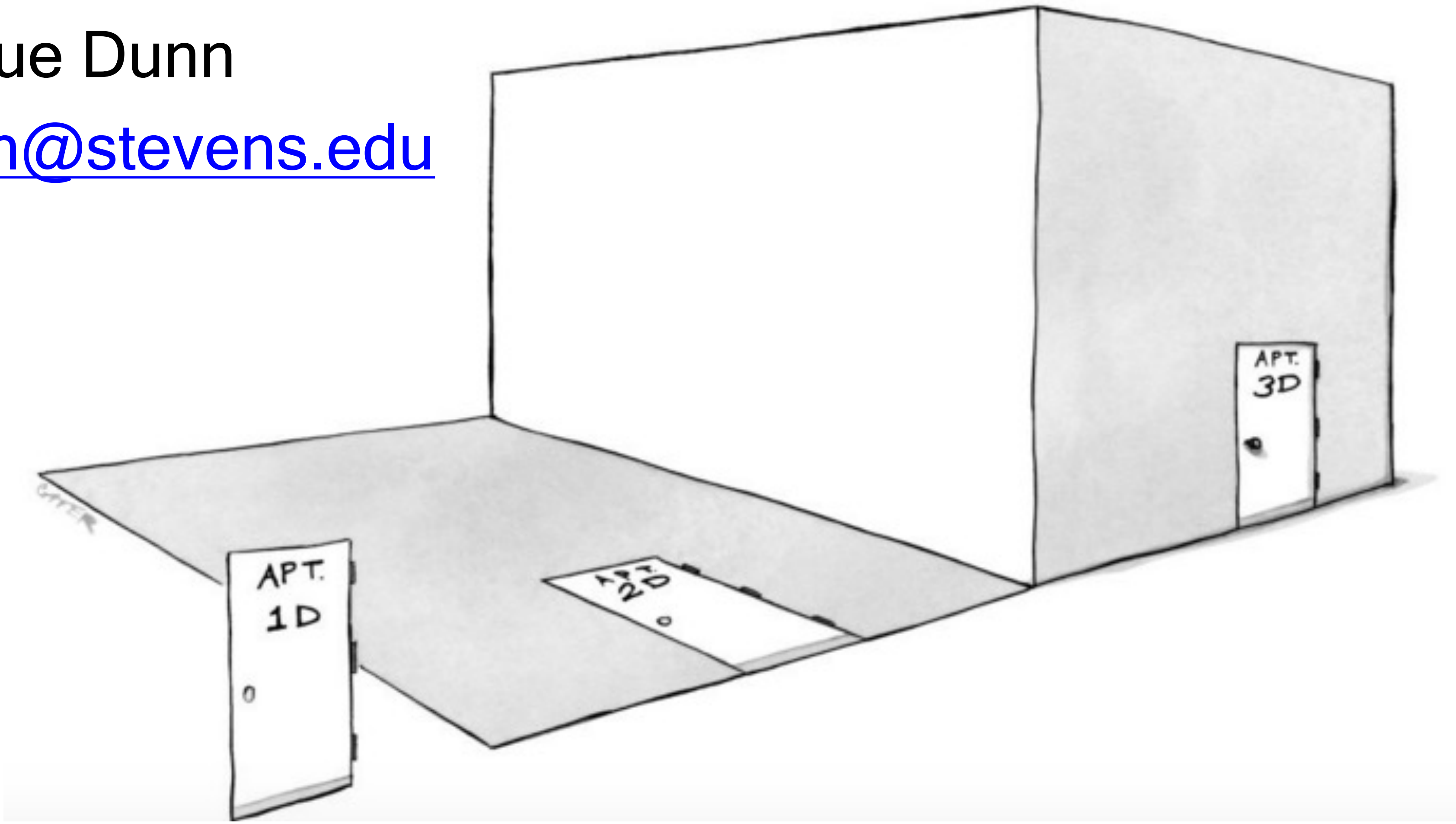
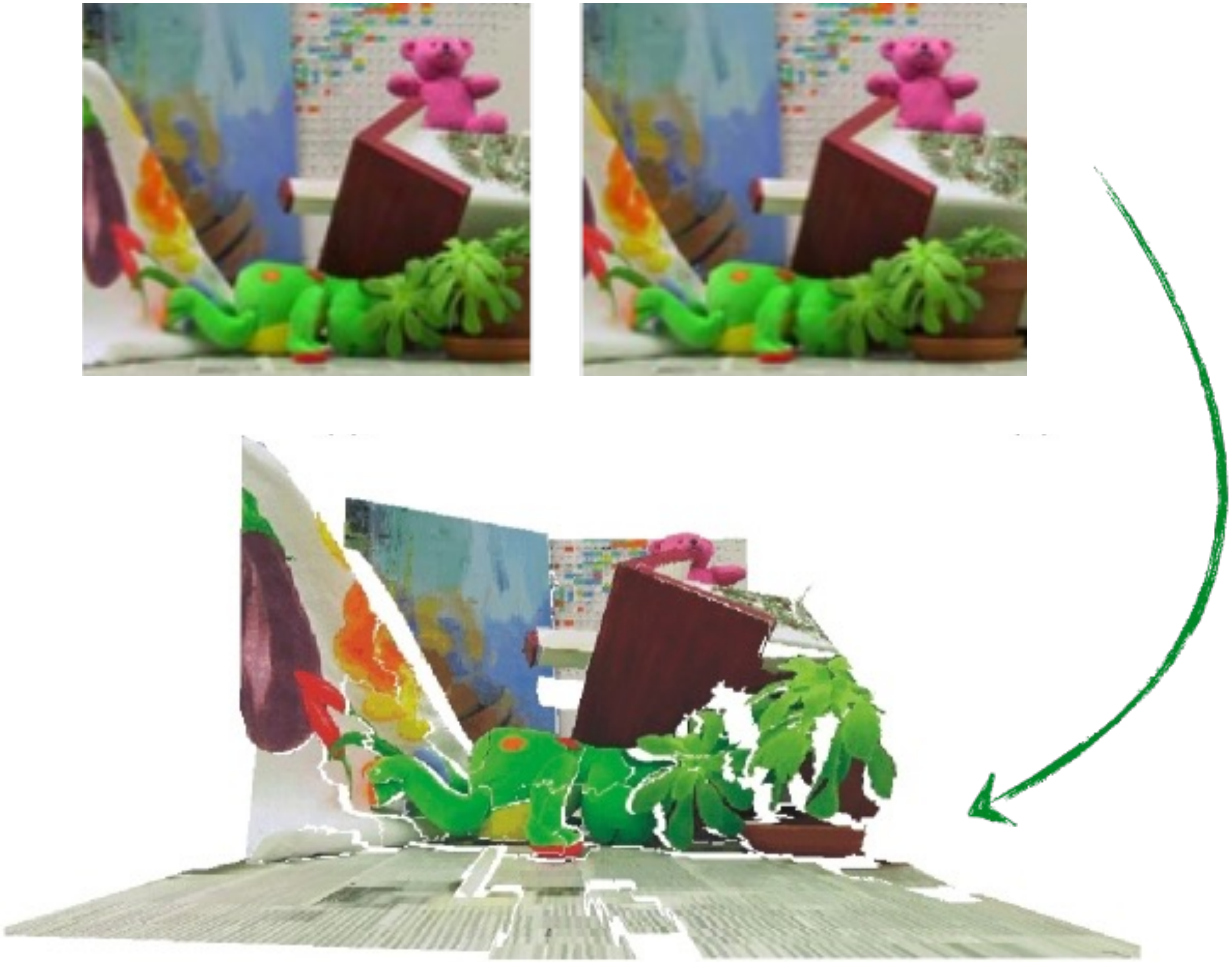
CS 532: 3D Computer Vision
Lecture 4

Enrique Dunn
edunn@stevens.edu



Two-View Stereo



Slide credit: S. Seitz

Stereo

Slide credit: S. Lazebnik

How Two Photographers Unknowingly Shot the Same Millisecond in Time

MAR 07, 2018 RON RISMAN

Slide credit:David Fouhey

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

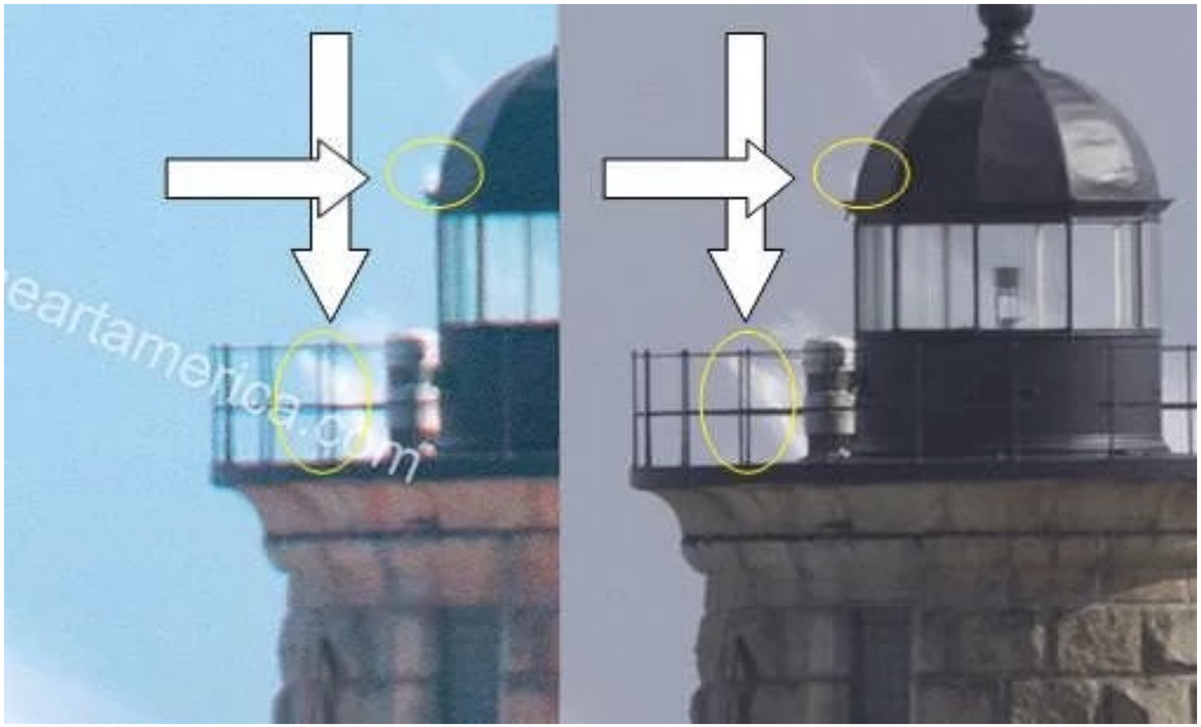
Slide credit:David Fouhey

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

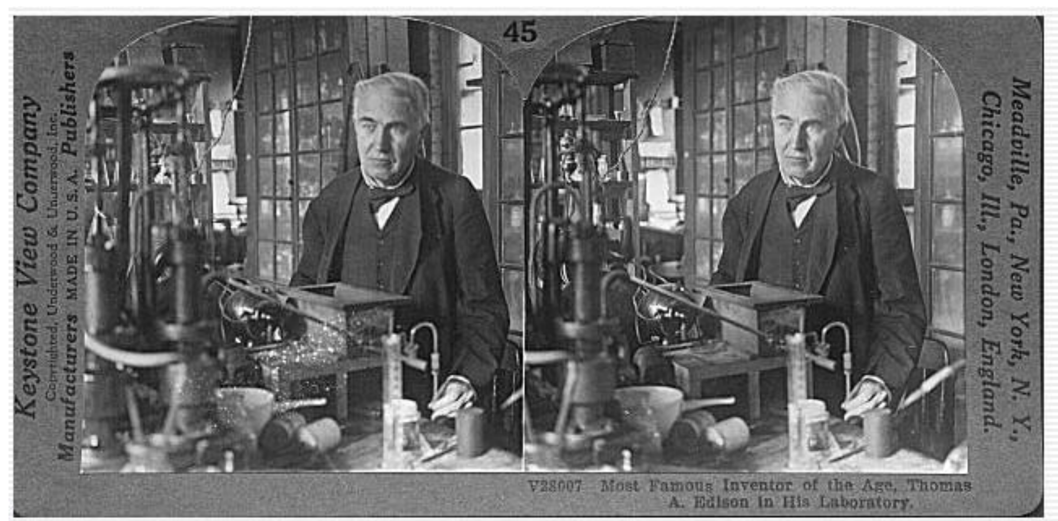


Slide credit:David Fouhey

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

Stereograms

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



Stereograms: Invented by Sir Charles Wheatstone, 1838

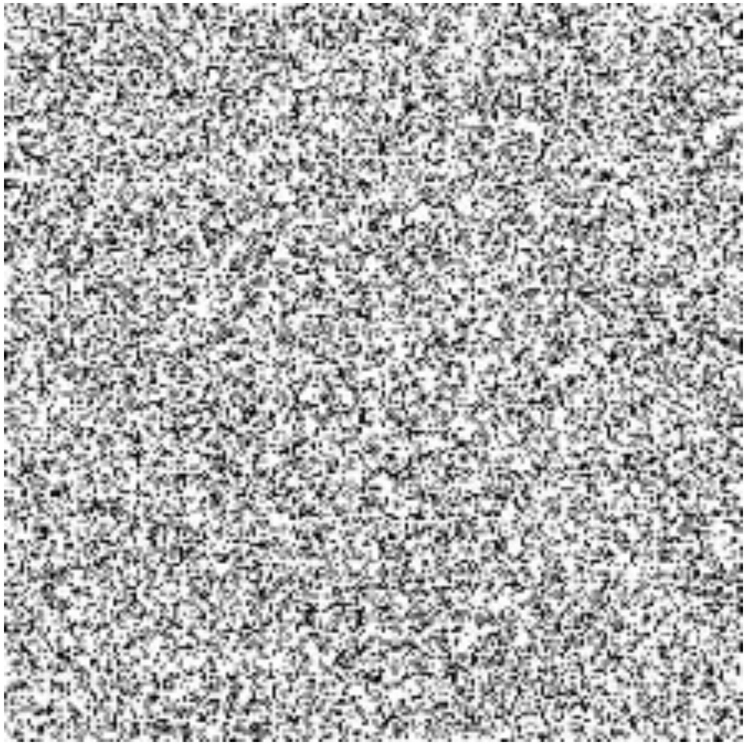
Slide credit: S. Lazebnik

Stereograms

Slide credit: S. Lazebnik

Stereograms

What about this?

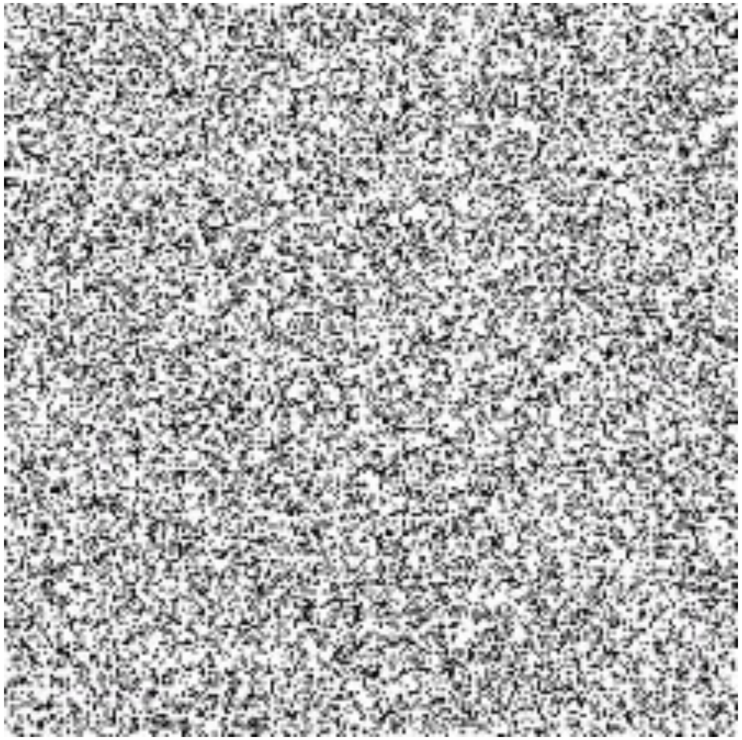


Slide Credit: S. Lazebnik, but idea of random dot sterogram is due to B. Julesz

Stereograms

Bela Julesz: Random Dot Stereogram

Shows that stereo can operate *without* recognition



Slide Credit: S. Lazebnik, but idea of random dot sterogram is due to B. Julesz

Stereograms

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

Autostereograms: www.magiceye.com

Slide credit: S. Lazebnik

Stereograms

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

Autostereograms: www.magiceye.com

Slide credit: S. Lazebnik

Problem formulation

Given a calibrated binocular stereo pair, fuse it to produce a depth image

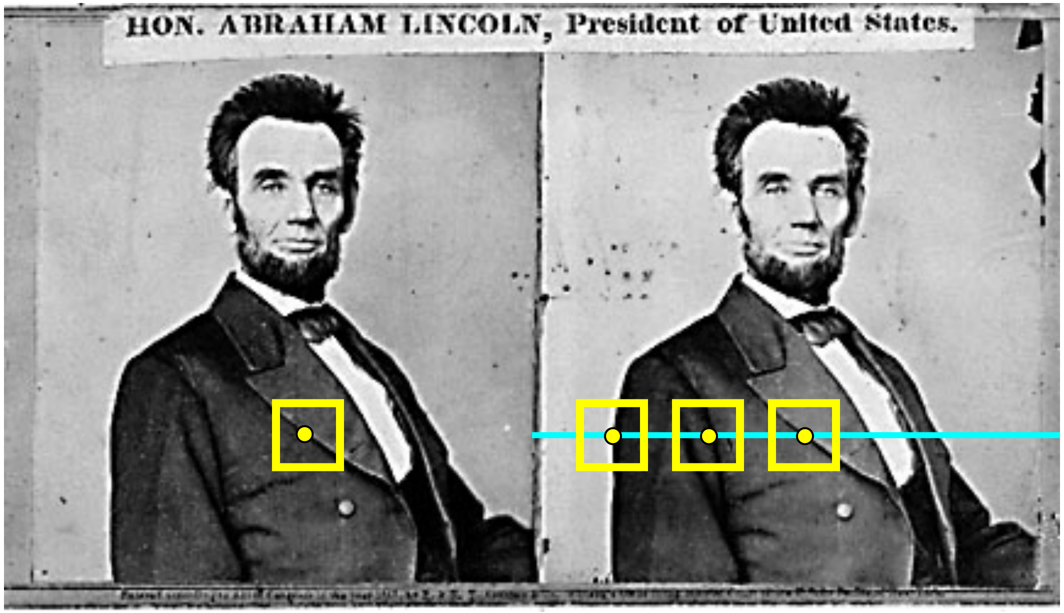
image 1

image 2

Dense depth map

Slide credit: S. Lazebnik

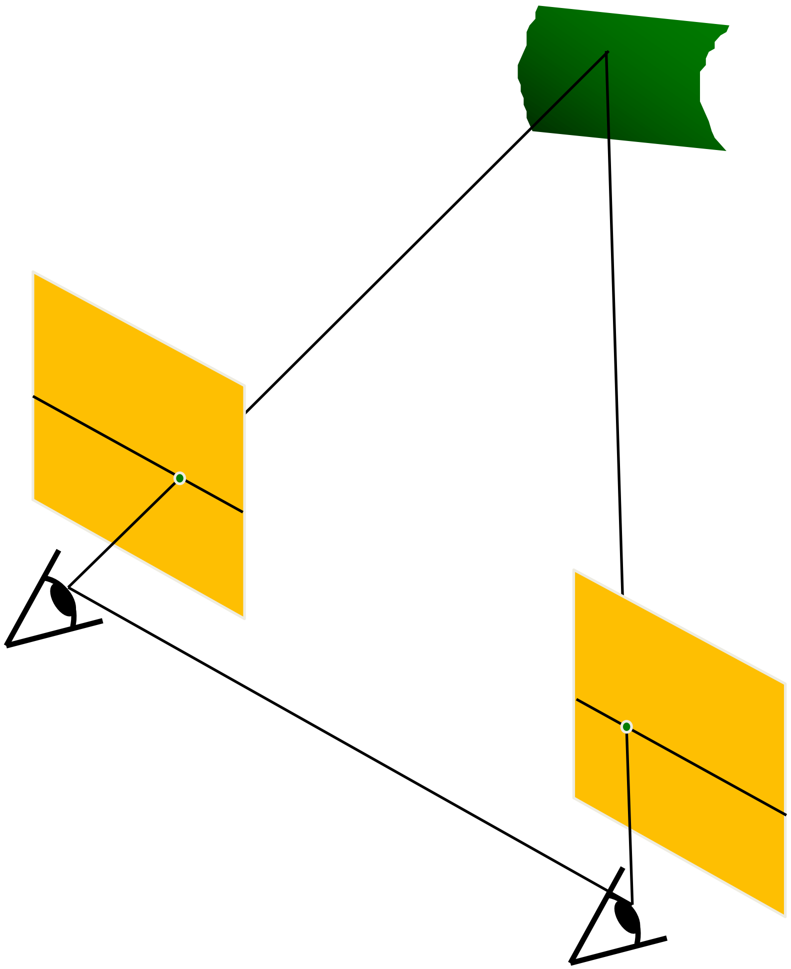
Basic stereo matching algorithm



- 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 = corresponding scanlines
  - When does this happen?

Slide credit: S. Lazebnik

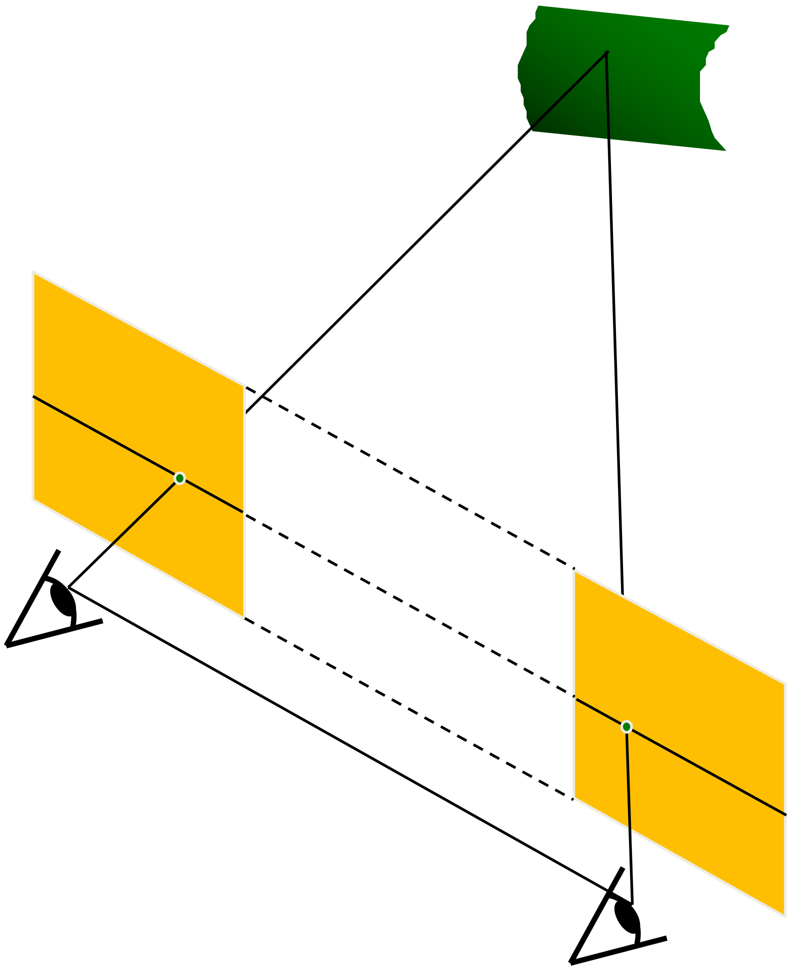
Simplest Case: Parallel images



- Image planes of cameras are parallel to each other and to the baseline
- Camera centers are at same height
- Focal lengths the same

Slide credit: S. Lazebnik

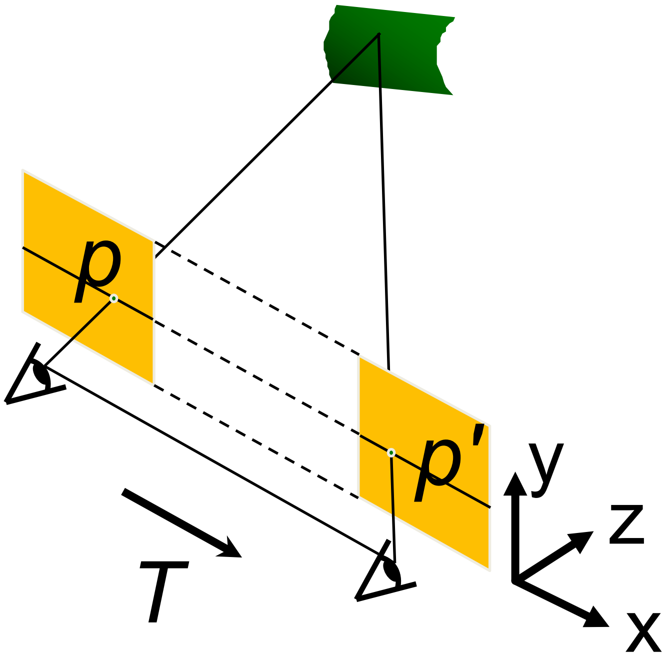
Simplest Case: Parallel images



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

Slide credit: S. Lazebnik

Essential matrix for parallel images



$$\boldsymbol{p}'^{T}\boldsymbol{E}\boldsymbol{p} = 0 \qquad \boldsymbol{E} = [\boldsymbol{t}_x]\boldsymbol{R}$$

What's R? **What's t?**

$$\boldsymbol{R} = \boldsymbol{I} \qquad t = [T, 0, 0]$$

$$\boldsymbol{E} = [\boldsymbol{t}_x]\boldsymbol{R} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix}$$

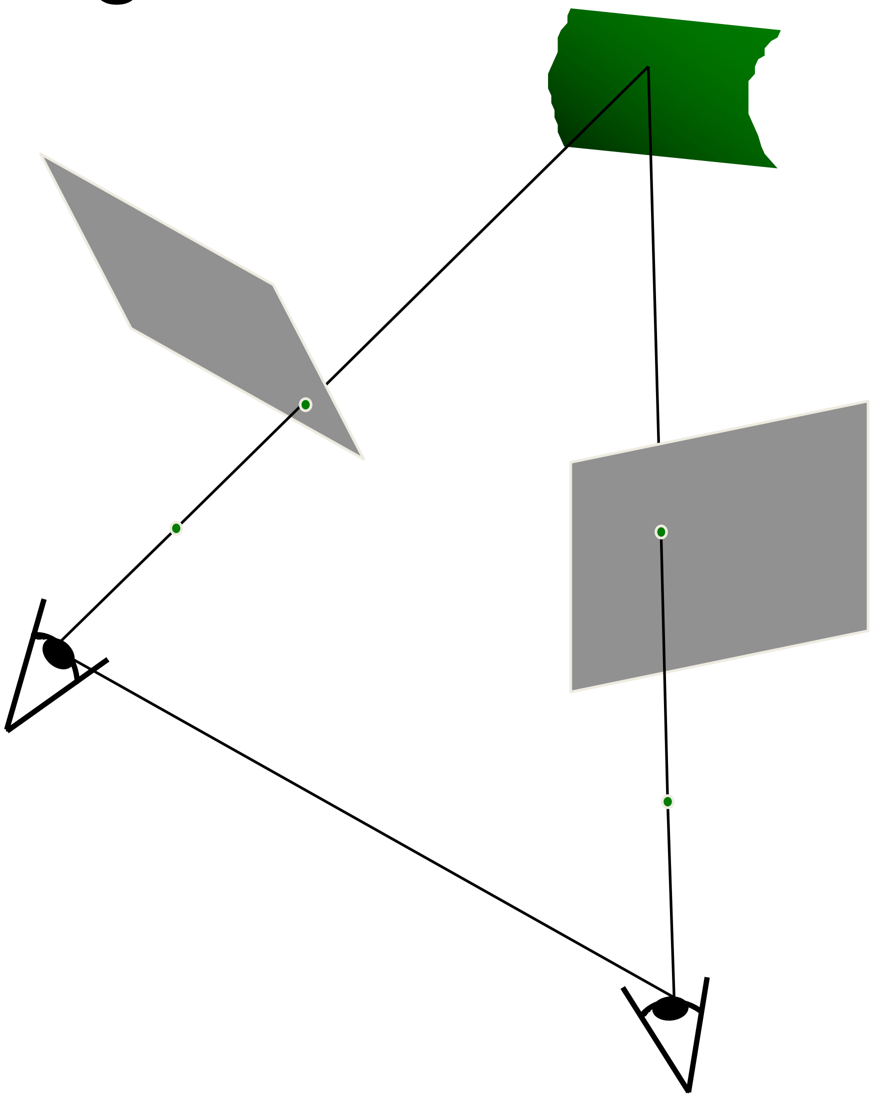
$$[u' \ v' \ 1]\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0 \longrightarrow [u' \ v' \ 1]\begin{bmatrix} 0 \\ -T \\ Tv \end{bmatrix} = 0 \longrightarrow -Tv' + Tv = 0$$

$$Tv = Tv'$$

The y-coordinates of corresponding points are the same!

Slide credit: S. Lazebnik

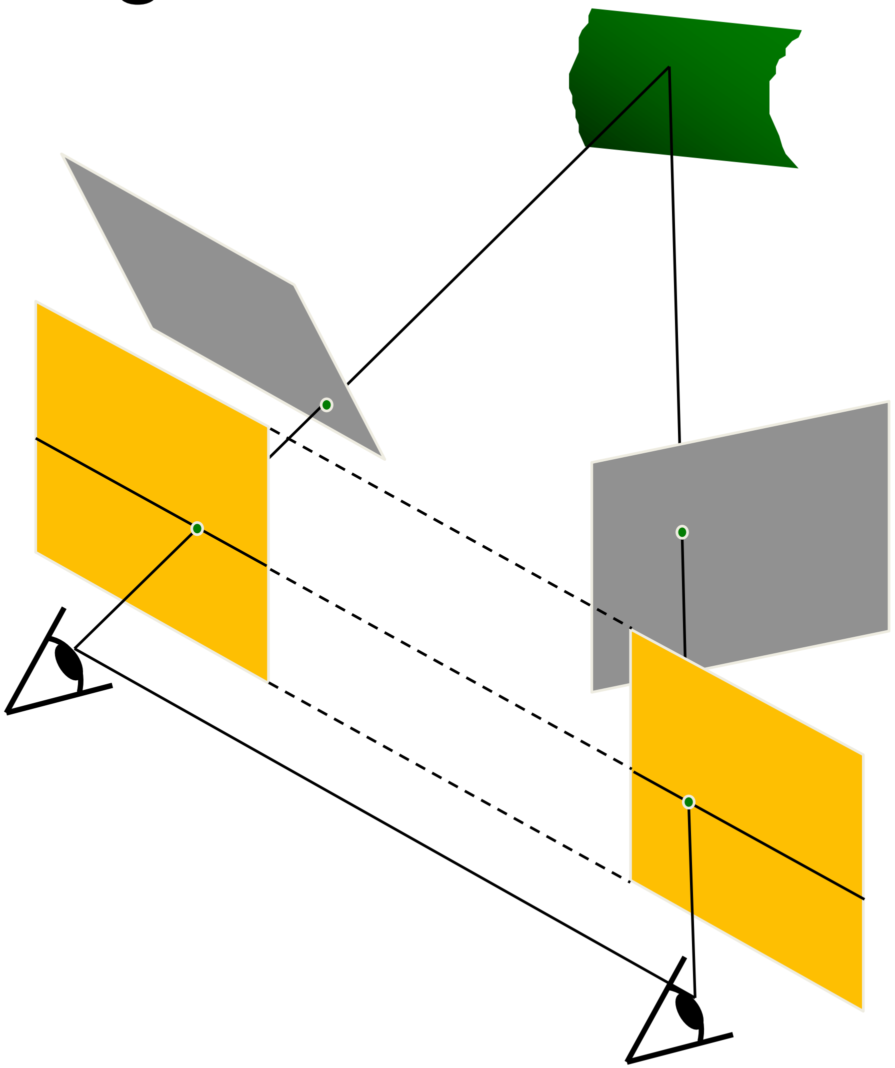
Stereo image rectification



Slide credit: S. Lazebnik

Stereo image rectification

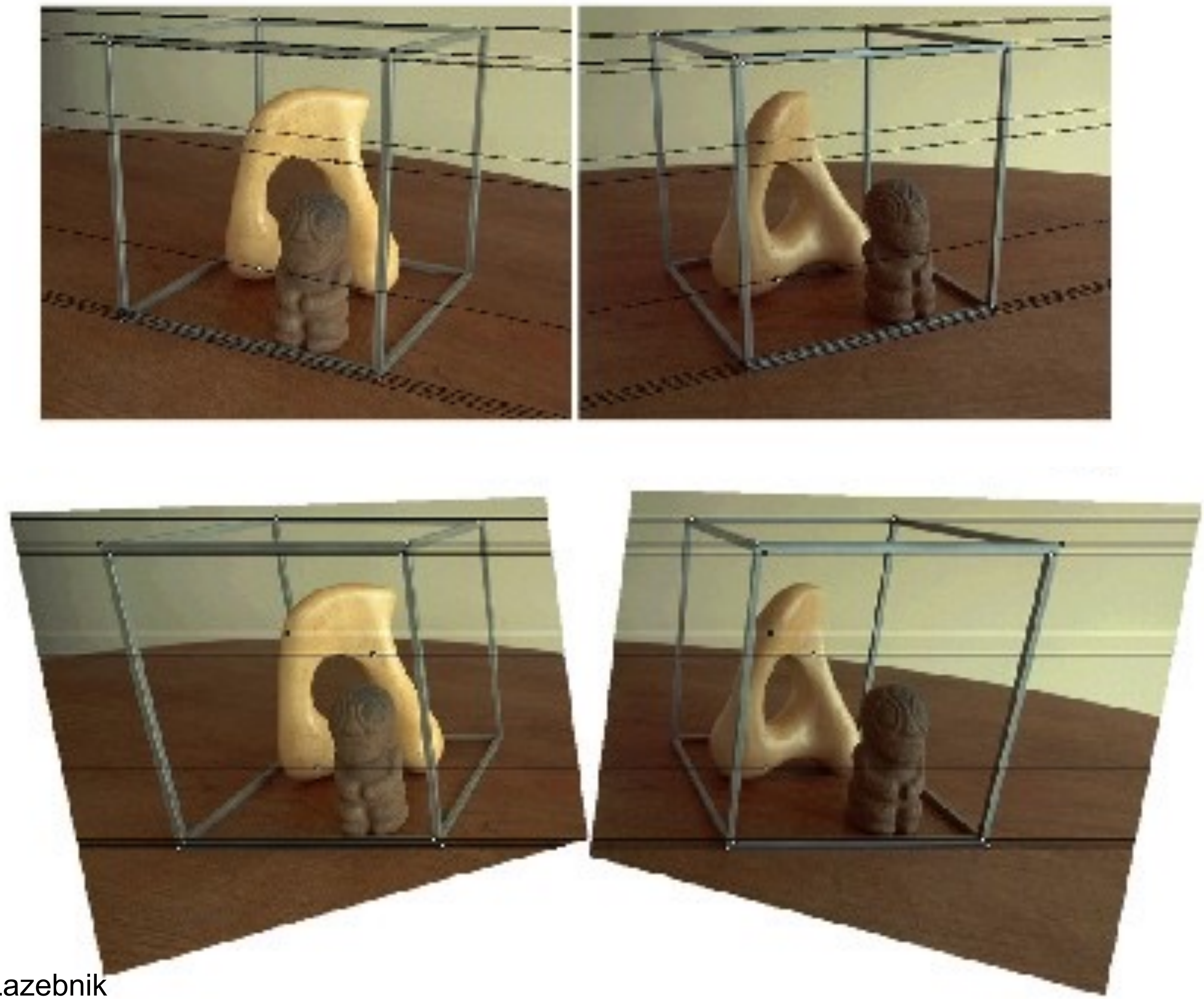
Reproject image planes onto a common plane parallel to the line between optical centers



C. Loop and Z. Zhang. [Computing Rectifying Homographies for Stereo Vision](). CVPR 1999

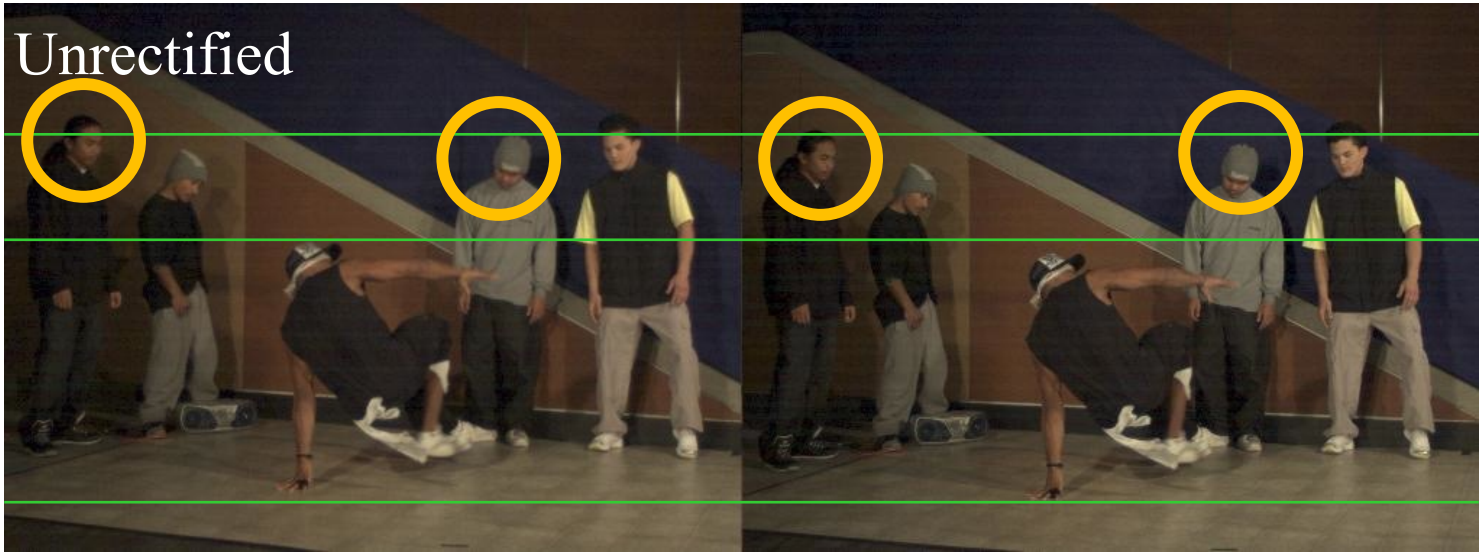
Slide credit: S. Lazebnik

Rectification example



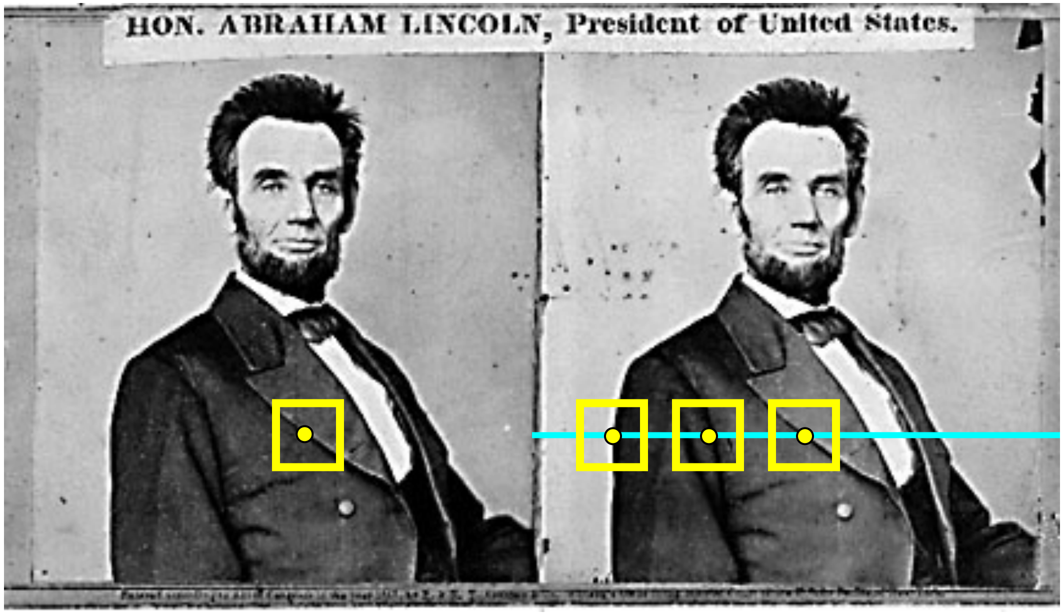
Slide credit: S. Lazebnik

Another rectification example



Slide credit: S. Lazebnik

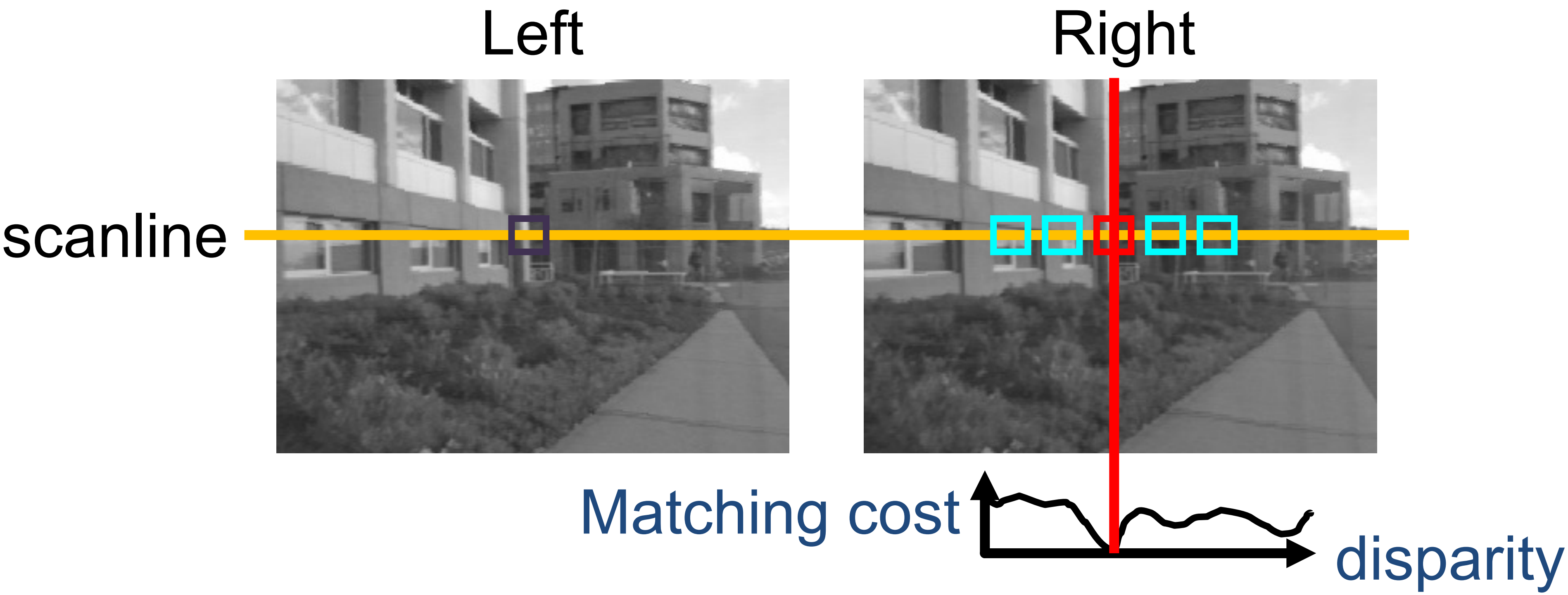
Basic stereo matching algorithm



- If necessary, rectify the two stereo images to transform epipolar lines into scanlines
- 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

Slide credit: S. Lazebnik

Correspondence Search



Slide window along the right scanline, compare contents of that window with reference window on left

Matching cost: SSD or normalized correlation

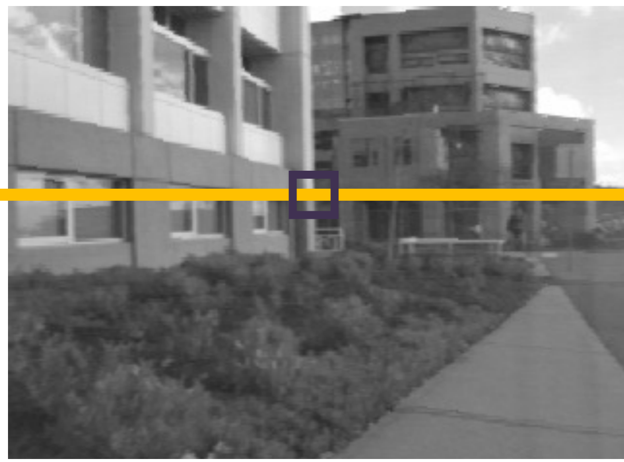
Slide credit: S. Lazebnik

Correspondence Search

Left

Right

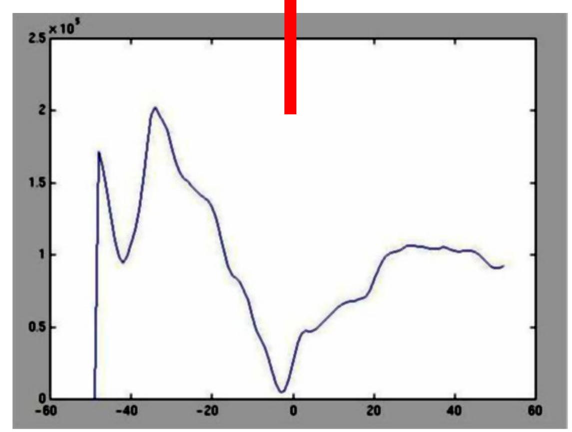
scanline



Matching cost

Sum of squared differences

$$\sum_i (l_i - r_i)^2$$



Disparity

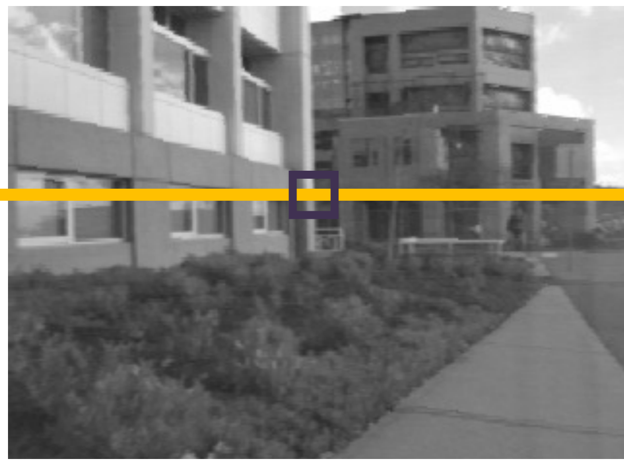
Slide credit: S. Lazebnik

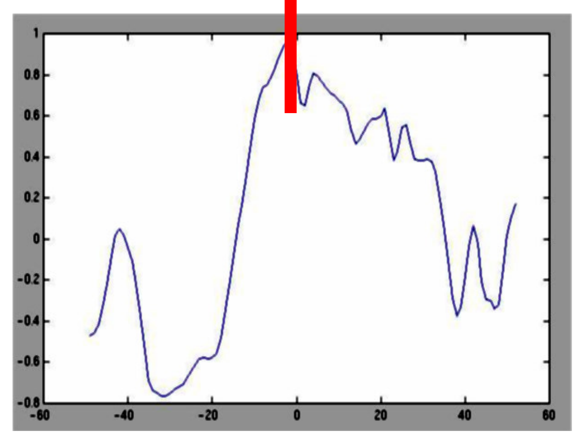
Correspondence Search

Left

Right

scanline





Matching cost

Normalized correlation

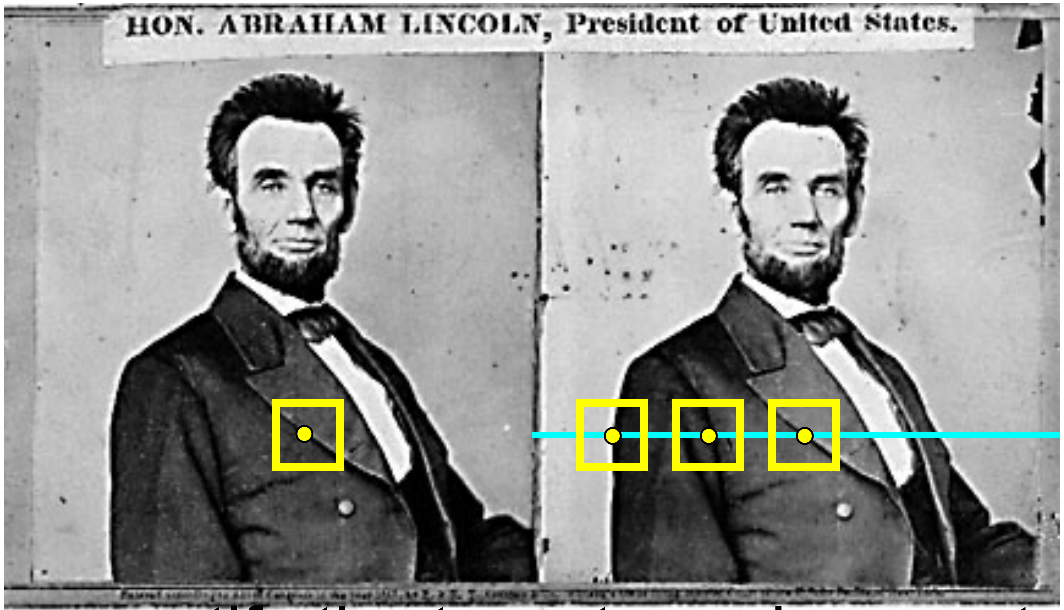
$$\hat{x}_i = \frac{x_i - \mathrm{mean}(x)}{\mathrm{std}(x)}$$

$$\hat{l} \cdot \hat{r}$$

Disparity

Slide credit: S. Lazebnik

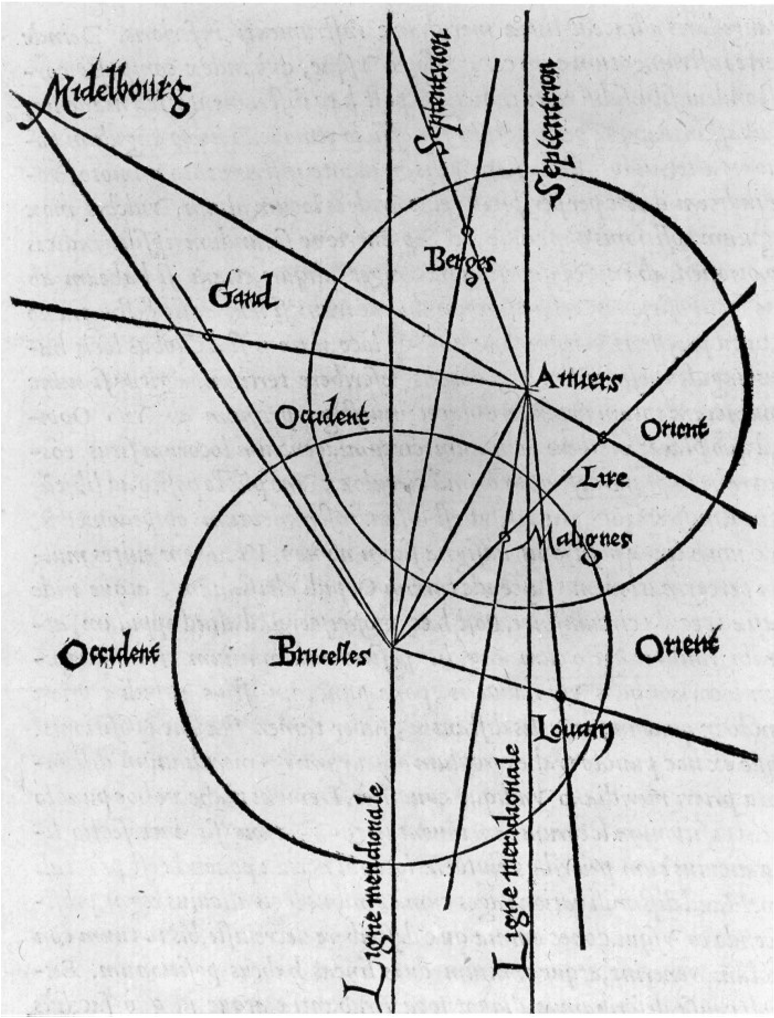
Basic stereo matching algorithm



- 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

Slide credit: S. Lazebnik

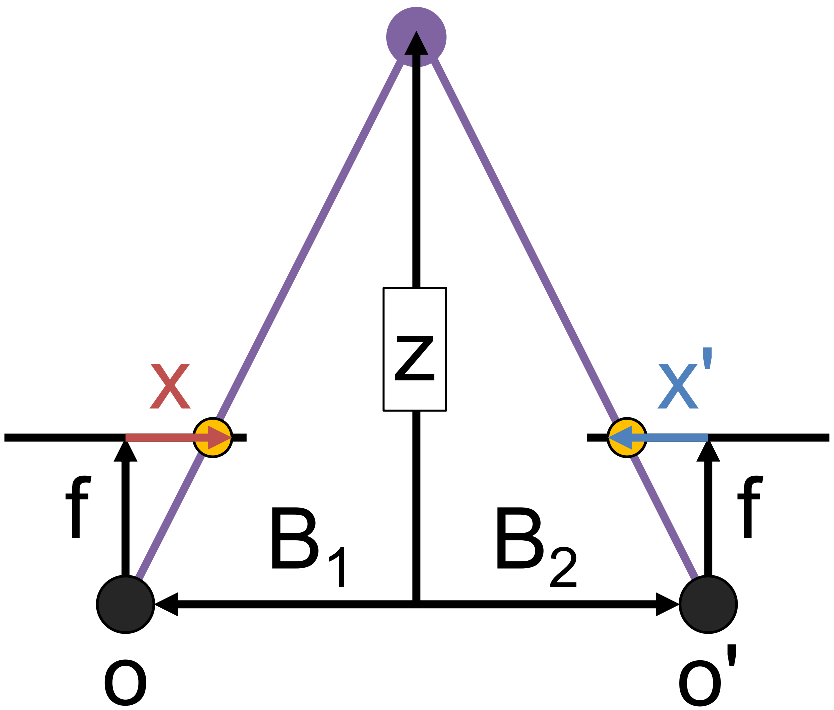
Triangulation: Older History



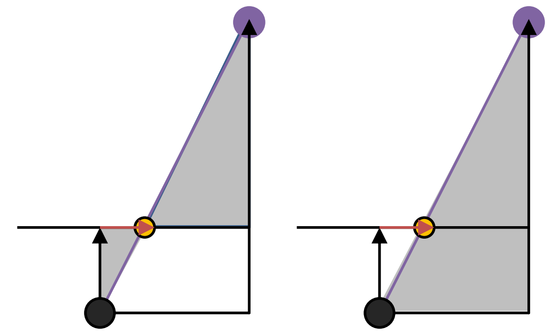
- From [Wikipedia](https://en.wikipedia.org/wiki/Triangulation): *Gemma Frisius's 1533 diagram introducing the idea of triangulation into the science of surveying. Having established a baseline, e.g. the cities of Brussels and Antwerp, the location of other cities, e.g. Middelburg, Ghent etc., can be found by taking a compass direction from each end of the baseline, and plotting where the two directions cross. This was only a theoretical presentation of the concept — due to topographical restrictions, it is impossible to see Middelburg from either Brussels or Antwerp. Nevertheless, the figure soon became well known all across Europe.*

Slide credit: S. Lazebnik

Depth from disparity



$$\frac{x}{f} = \frac{B_1}{z}$$



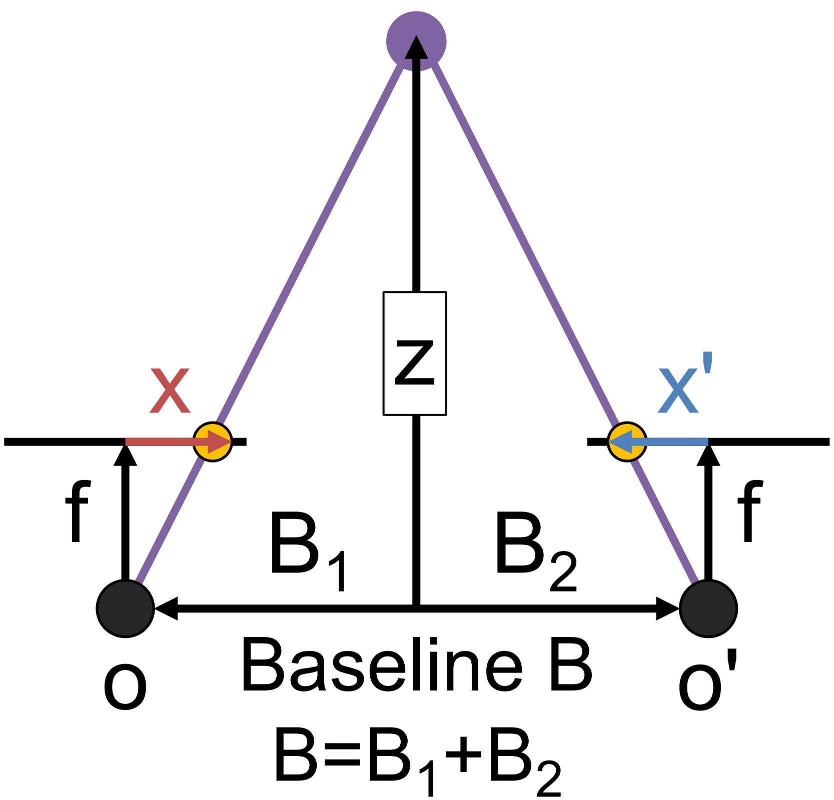
By similar triangles

$$\frac{-x'}{f} = \frac{B_2}{z}$$

Similarly by similar triangles

Diagram adapted from S. Lazebnik

Depth from disparity



$$\frac{x}{f} = \frac{B_1}{z} \qquad \frac{-x'}{f} = \frac{B_2}{z}$$

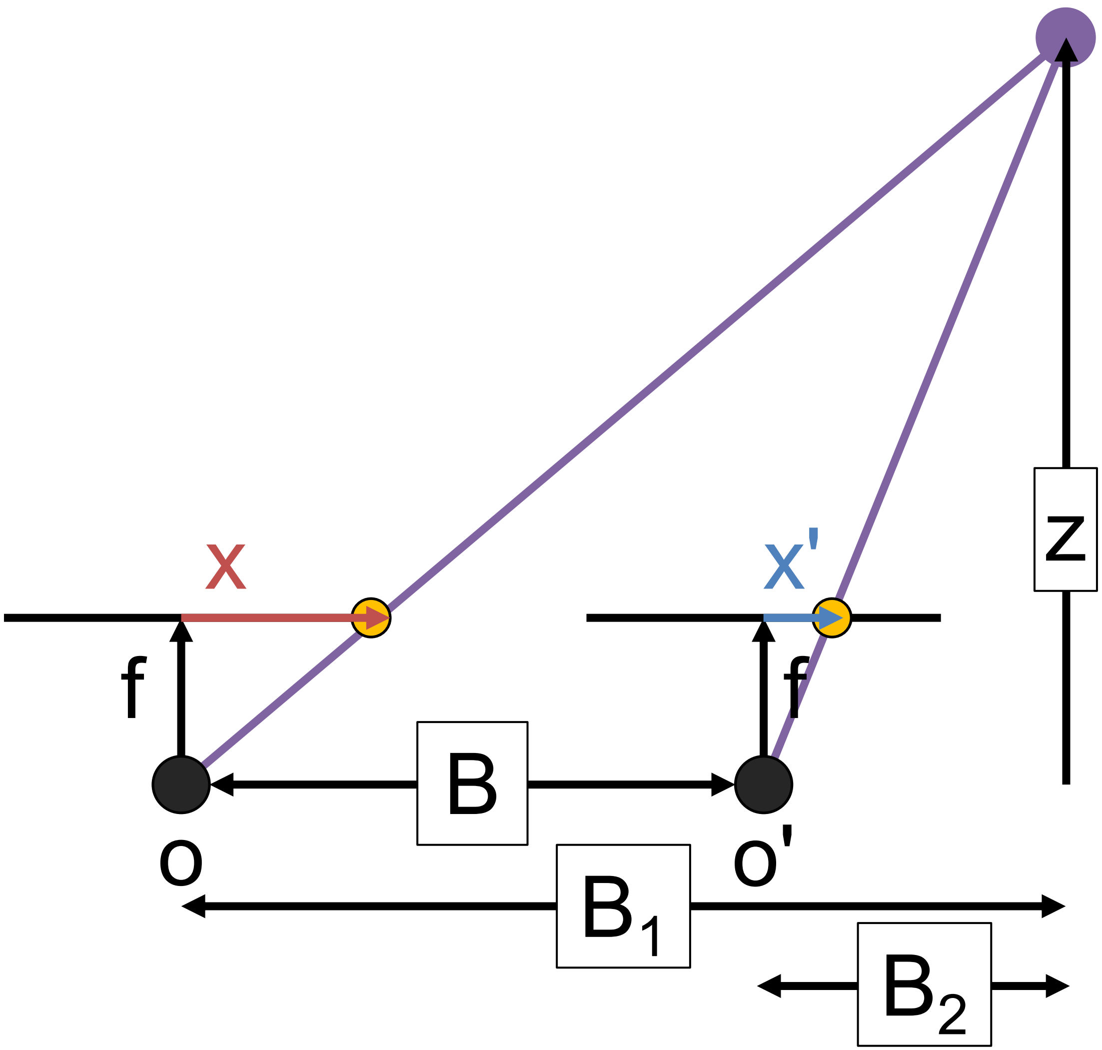
Add them

$$\frac{x - x'}{f} = \frac{B_1 + B_2}{z}$$

$$\underbrace{x - x'}_{\text{Disparity}} = \frac{fB}{z}$$

Diagram adapted from S. Lazebnik

Depth from disparity



$$\frac{x}{f} = \frac{B_1}{z} \qquad \frac{x'}{f} = \frac{B_2}{z}$$

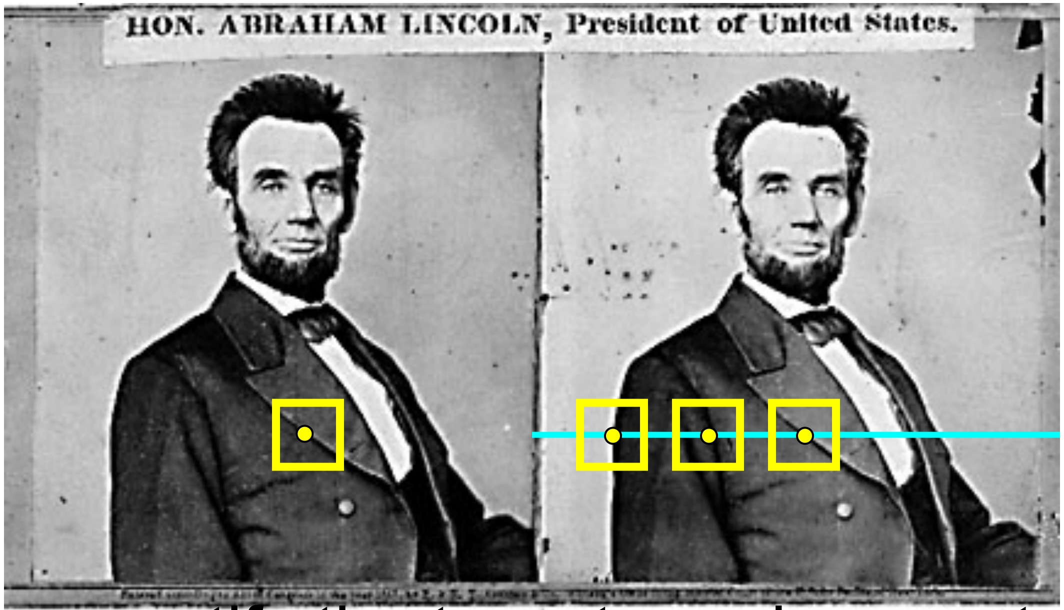
Subtract them

$$\frac{x - x'}{f} = \frac{B_1 - B_2}{z}$$

$$x - x' = \frac{fB}{z}$$

Diagram adapted from S. Lazebnik

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 $\text{depth}(x) = B*f/(x–x')$

Slide credit: S. Lazebnik

Failures of Correspondence Search

Textureless regions. **Why?**

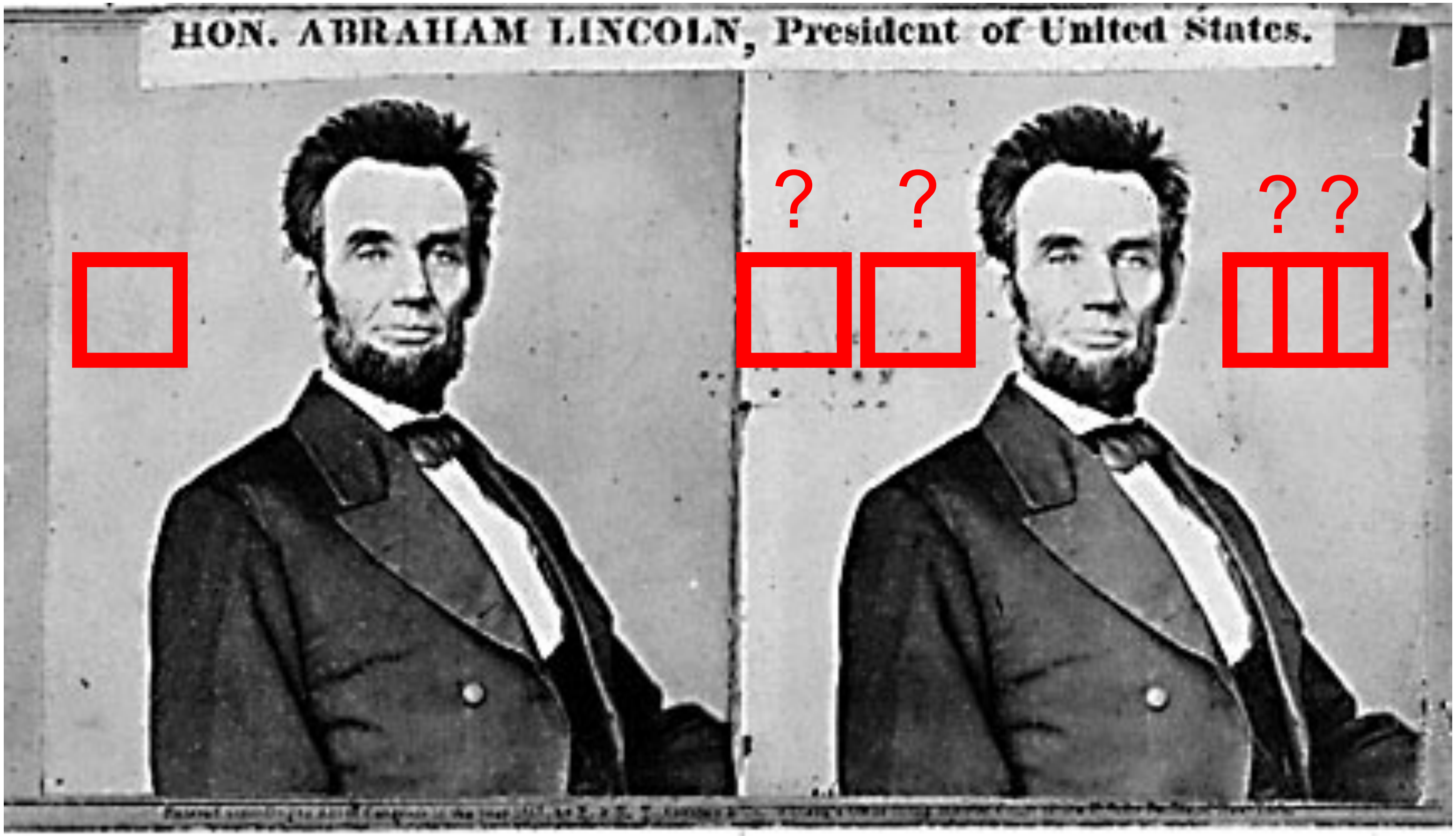


Image credit: S. Lazebnik

Failures of Correspondence Search

Repeated Patterns. **Why?**

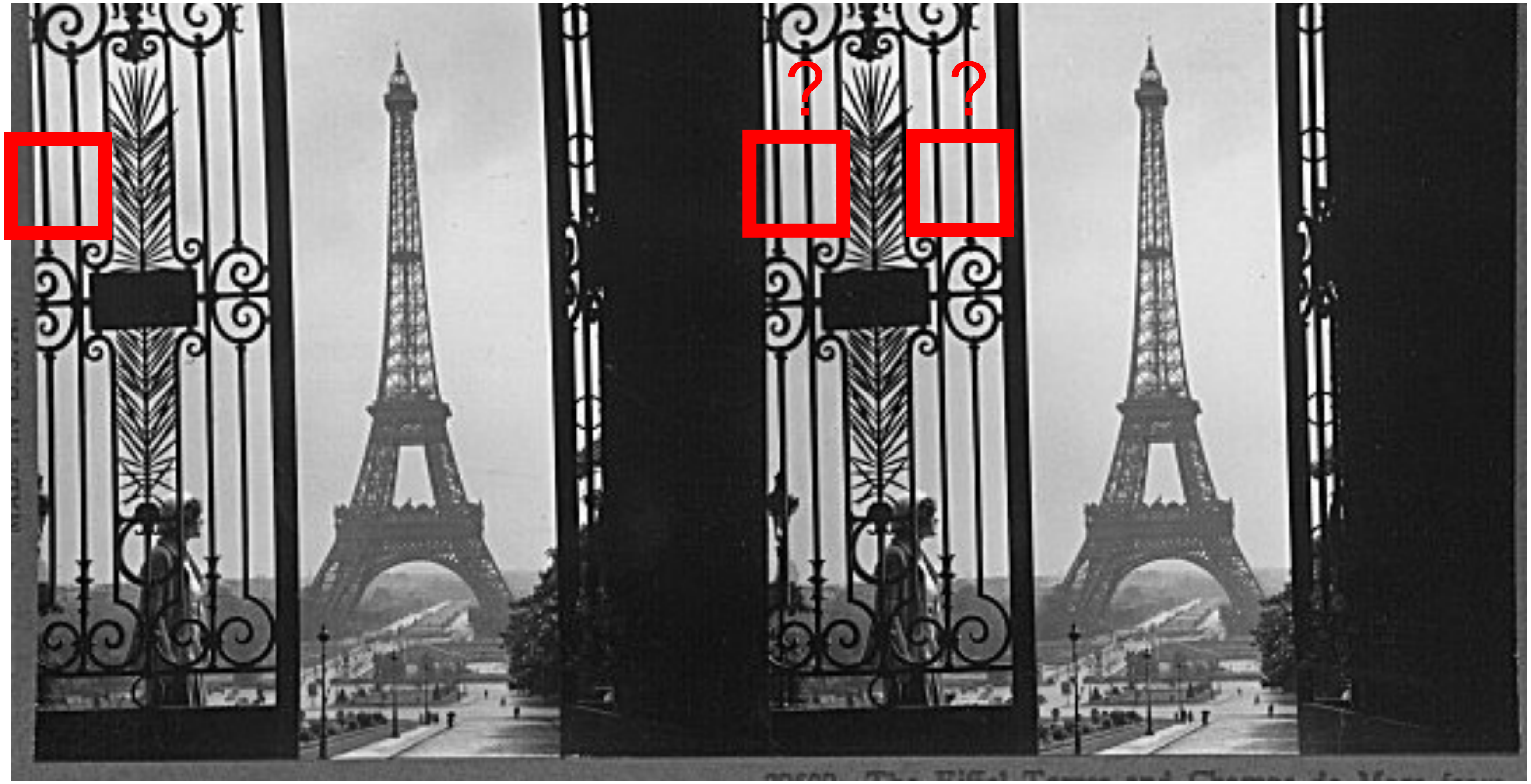


Image credit: S. Lazebnik

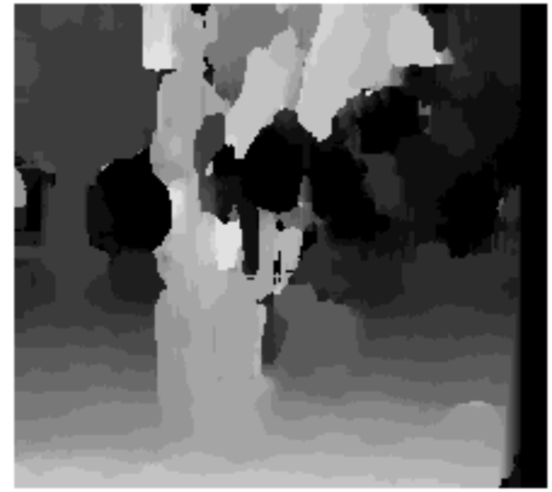
Failures of Correspondence Search

Specular Surfaces. **Why?**

Image credit: S. Lazebnik

Effect of window size

W = 3



W = 20

- Smaller window
  - \+ More detail
  - \- More noise

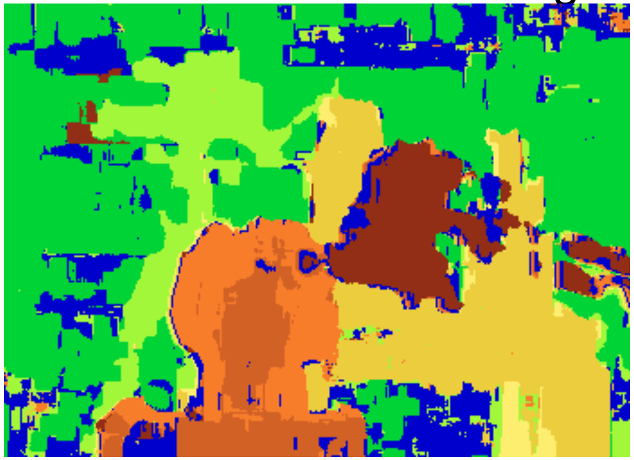
- Larger window
  - \+ Smoother disparity maps
  - \- Less detail

Image credit: S. Lazebnik

Results with window search

Data

Window-based matching



Ground truth

Image credit: S. Lazebnik

Better methods exist...

Graph cuts

Ground truth

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

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

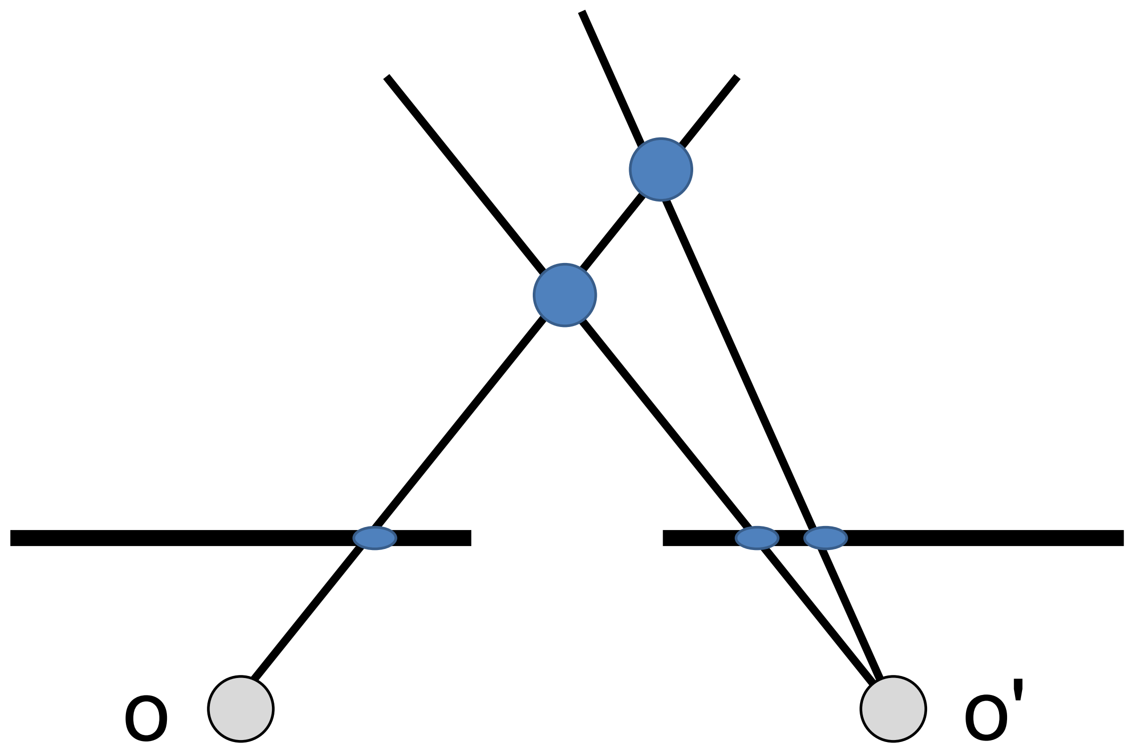
Image credit: S. Lazebnik

Improving Window-based Matching

- Similarity is **local** (each window independent)
- Need non-local correspondence constraints / cues.

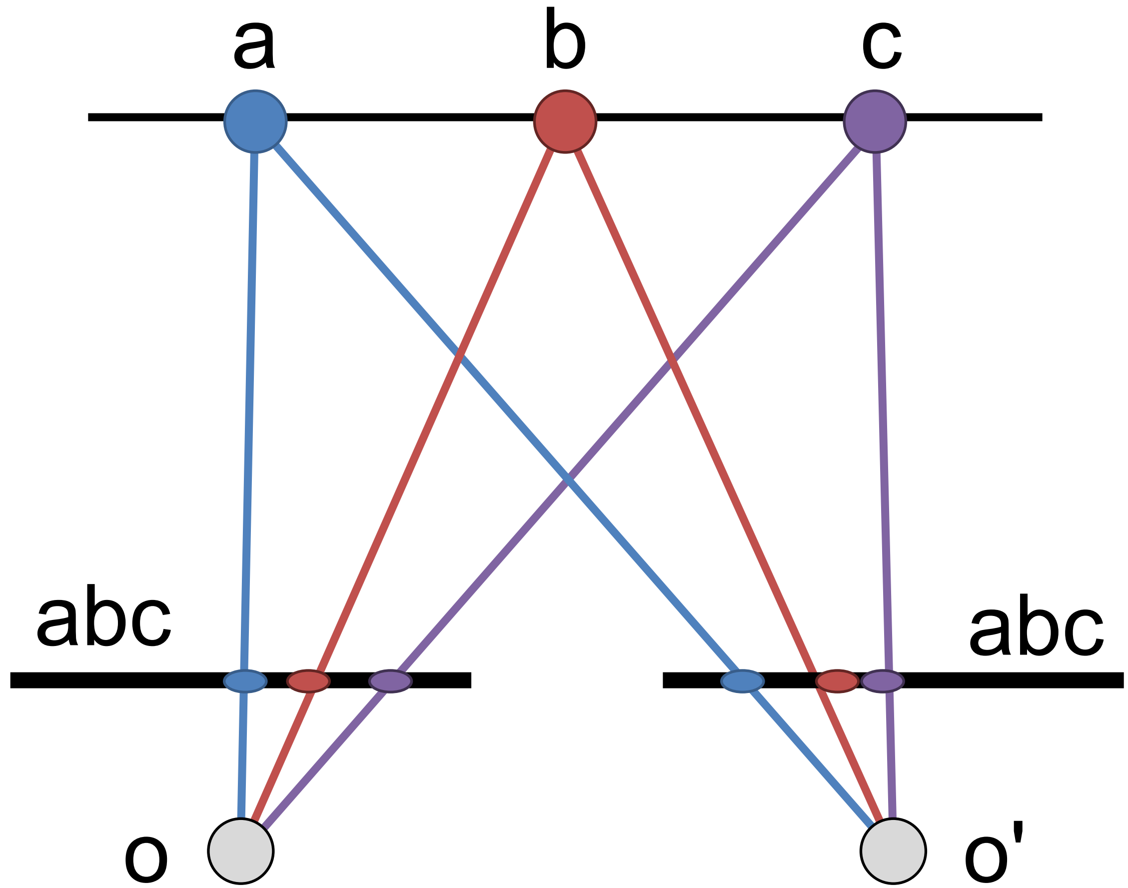
Uniqueness

- Each point in one image should match at most one point in other image.
- **When might this not be true?**



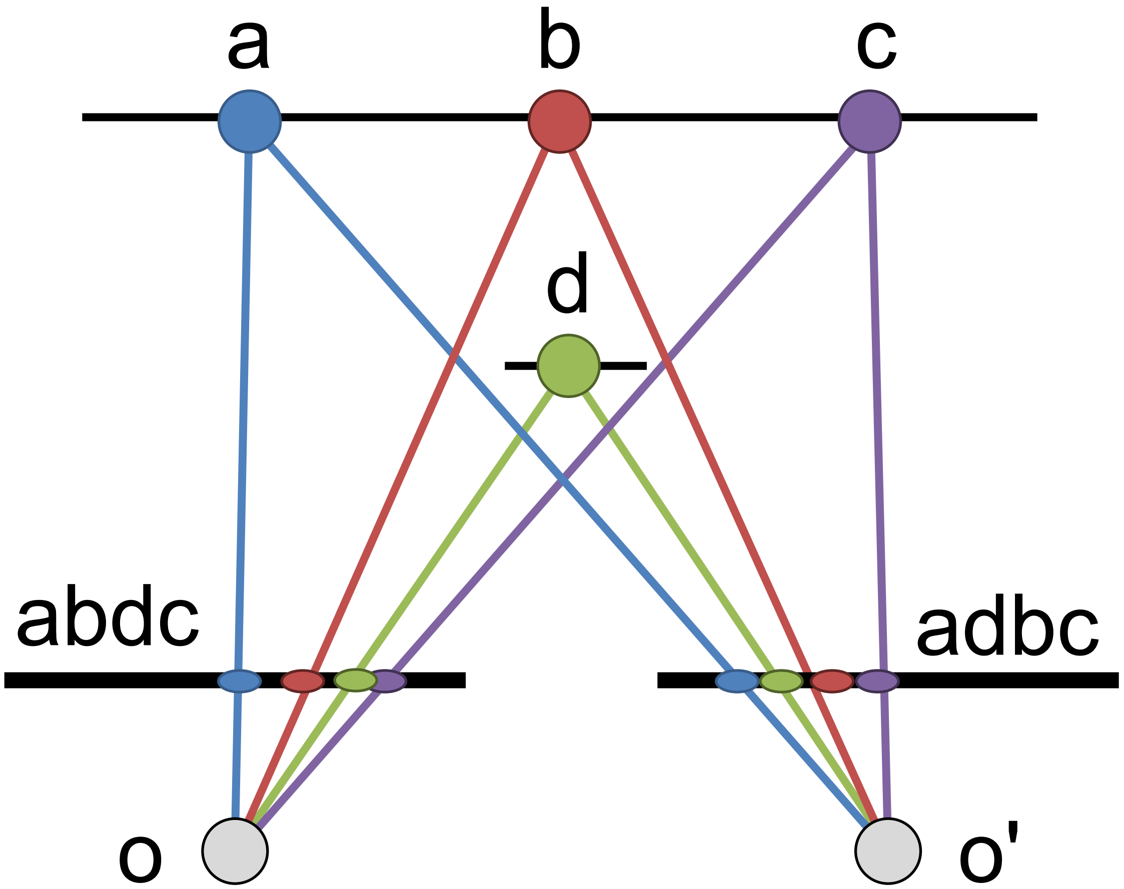
Ordering

- Corresponding points should be in same order



Ordering

- Not always true!

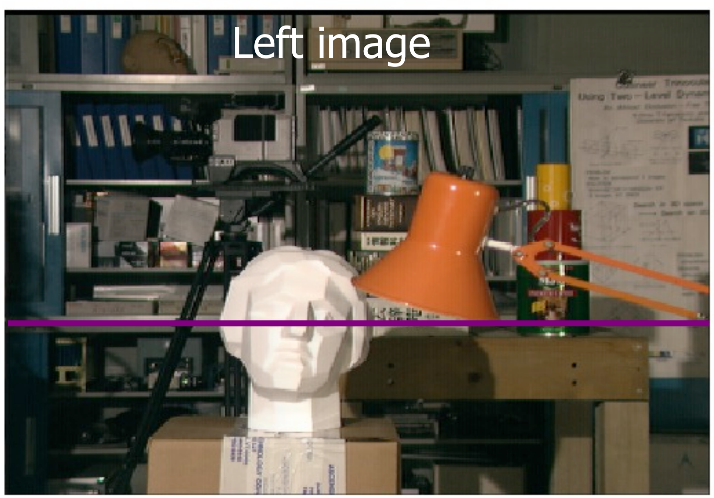


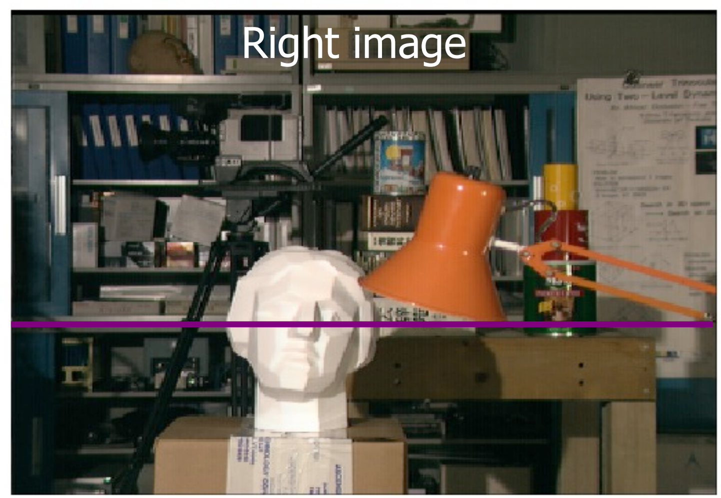
Smoothness

- We expect disparity values to change slowly (for the most part)
- **When is this not true?**

Scanline Stereo

- Try to coherently match pixels on the entire scanline
- Different scanlines are optimized (by dynamic programming) independently

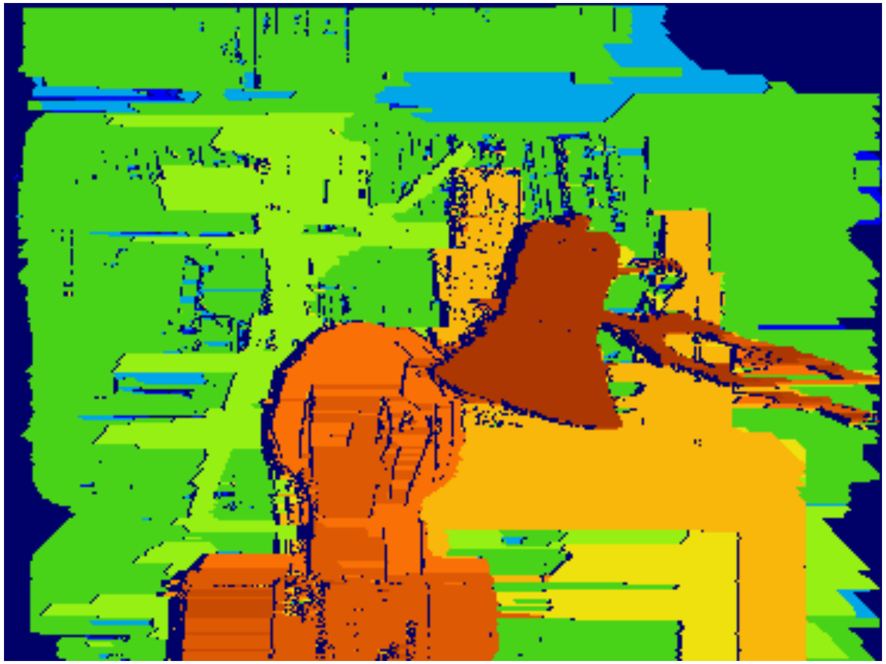




Slide credit: S. Lazebnik

Coherent Stereo on 2D Grid

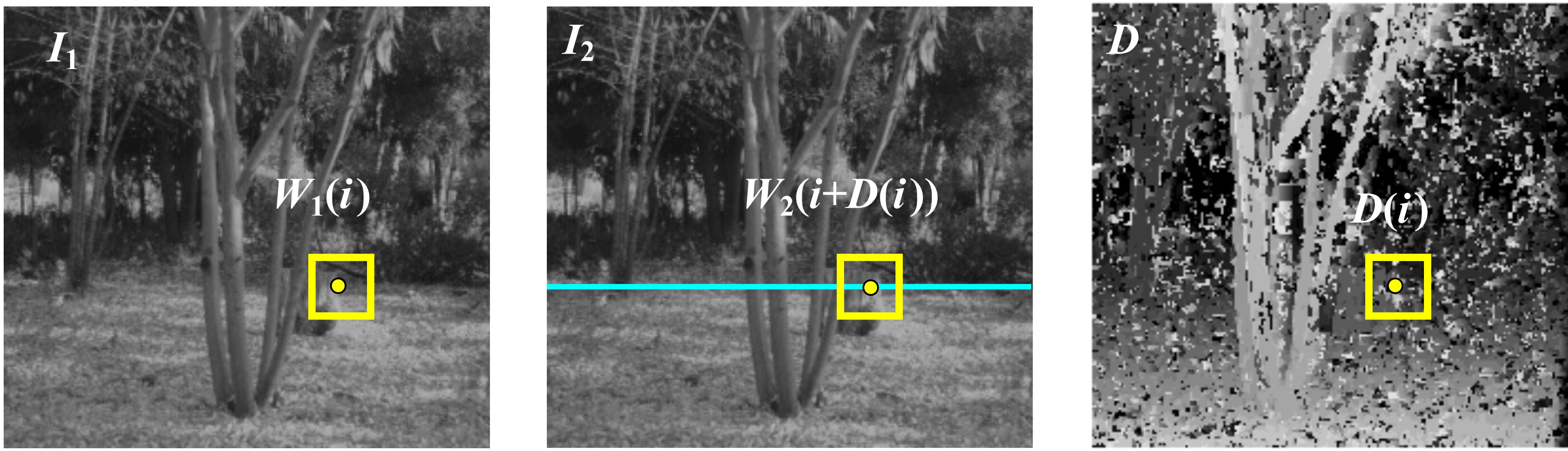
- Scanline stereo generates streaking artifacts



- Can't use dynamic programming to find spatially coherent disparities on a 2D grid

Slide credit: S. Lazebnik

Stereo Matching as Optimization



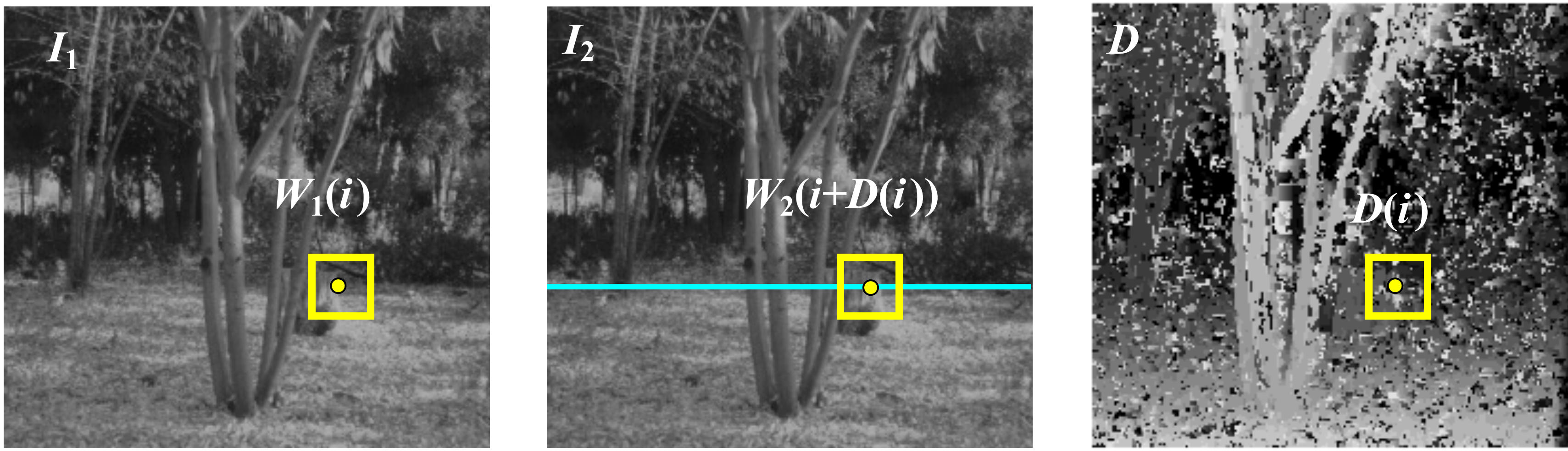
$$E(D) = \underbrace{\sum_i \left(W_1(i) - W_2\left(i + D(i)\right)\right)^2}_{\text{Data term}} + \lambda \underbrace{\sum_{\text{neighbors } i,j} \rho\left(D(i) - D(j)\right)}_{\text{Smoothness term}}$$

Solvable by graph cuts for certain smoothnesses ρ

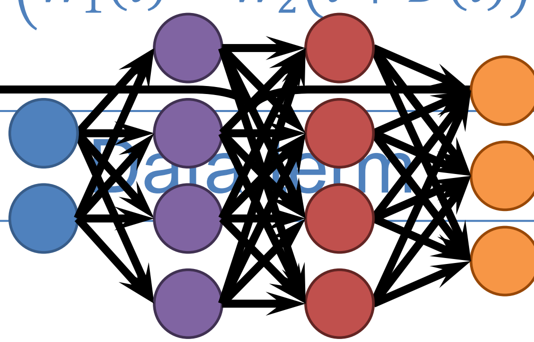
Y. Boykov, O. Veksler, and R. Zabih, [Fast Approximate Energy Minimization via Graph Cuts](), PAMI 2001

Slide credit: S. Lazebnik

Is This Doable by Deep Network?



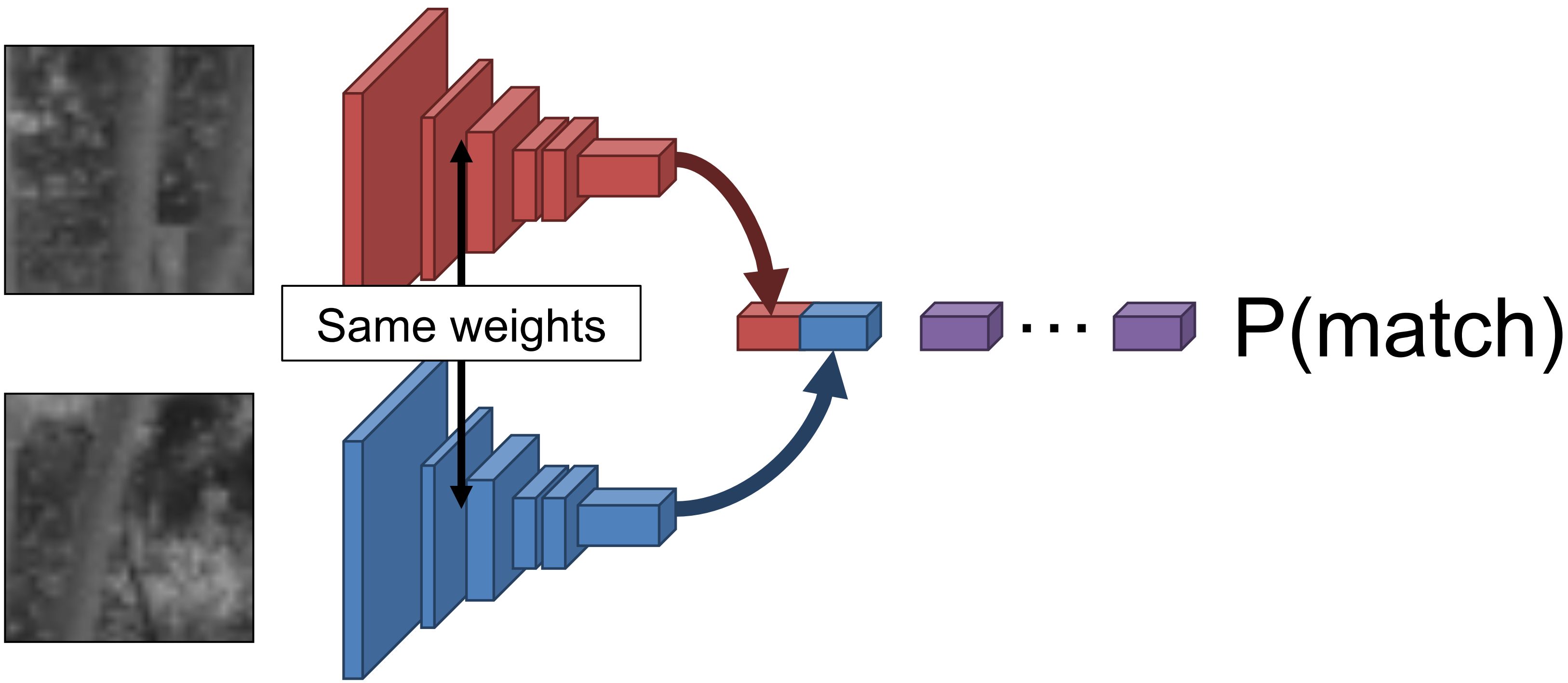
$$E(D) = \underbrace{\sum_i \left(W_1(i) - W_2\big(i + D(i)\big)\right)^2}_{\text{Data term}} + \lambda \underbrace{\sum_{\text{neighbors } i,j} \rho\big(D(i) - D(j)\big)}_{\text{Smoothness term}}$$



Easy solution: replace the data term with a network

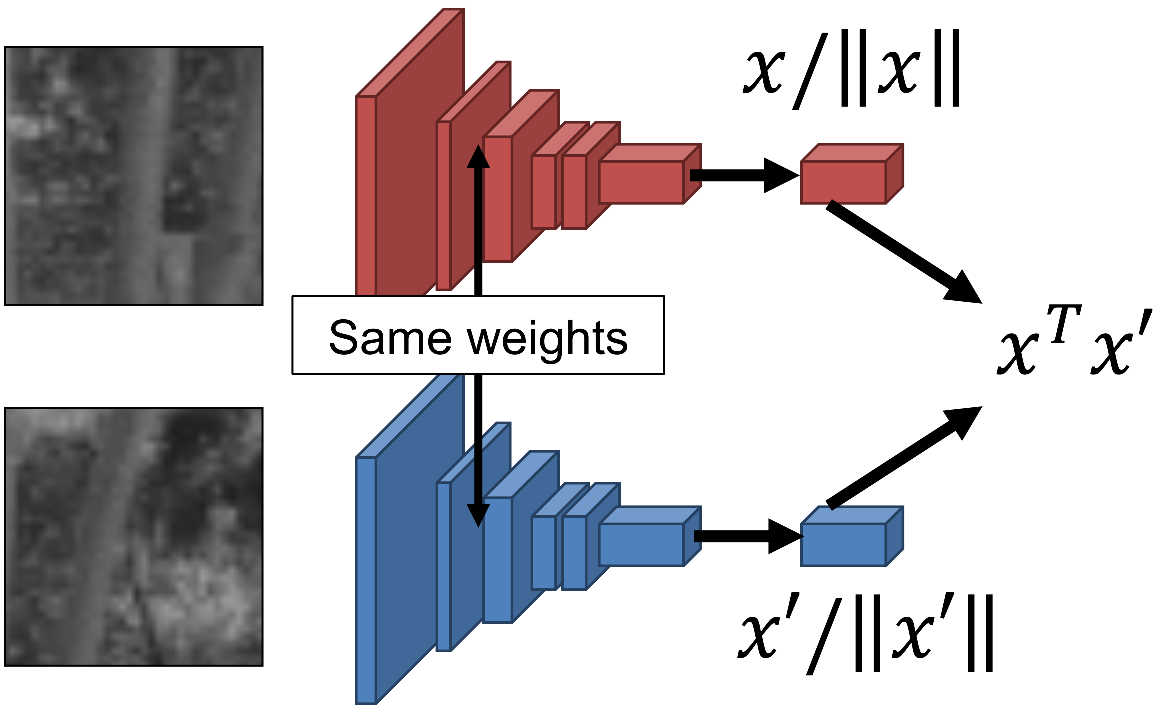
Deep Learning For Stereo

- Feed in two images to identical networks, concatenate outputs, learn multilayer perceptron
- Slow: **why?**



Deep Learning For Stereo

- Normalize outputs; treat dot product as prediction of match/no match
- Fast: **why?**



svm/hinge loss

Stereo datasets

- [Middlebury stereo datasets]
- [KITTI]
- [Synthetic data?]

Slide credit: S. Lazebnik