Computer Vision and Image Processing

Gil Ben-Artzi

6

Agenda

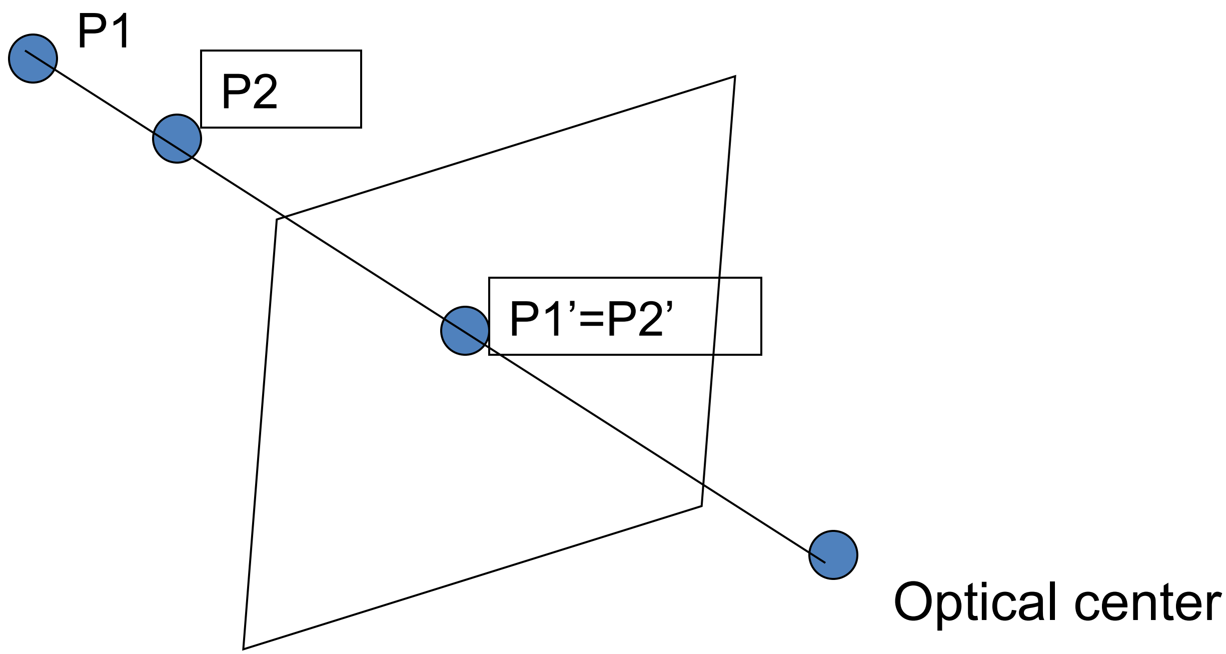
- Topic 1
  - Image Enhancement: histogram, quantization
- Topic 2
  - Filtering: smoothing, median filtering, sharpening
  - Low level detection: Template matching, Edges, Line, Circles
- Topic 3
  - Image Pyramids and Blending, Optical Flow
- Topic 4
  - Geometry: 2D Transformation, Image Warping, Camera Model
- Topic 5
  - Stereo, Homography, Image Stitching (Mosaic/Panorama)
  - Features, RANSAC

Stereo

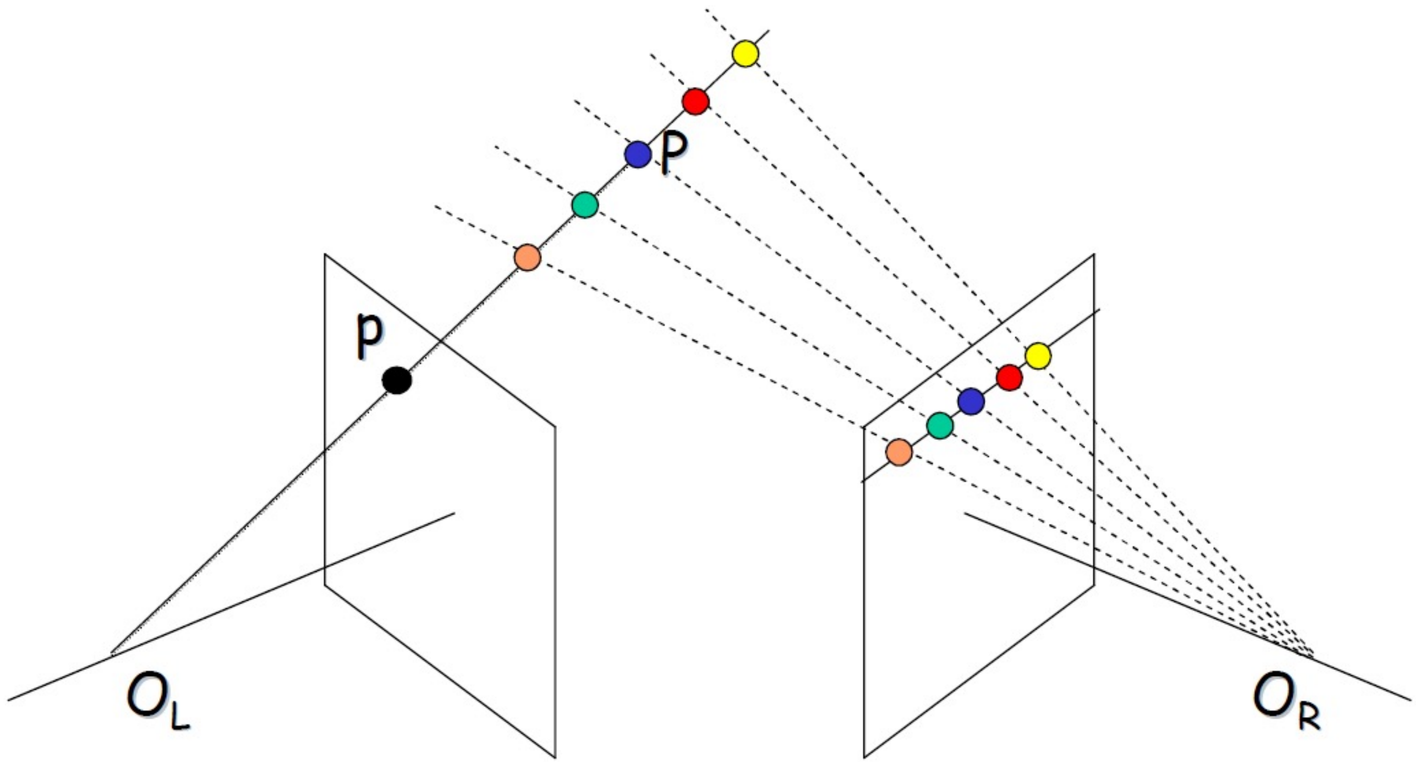

WHY DO WE HAVE
TWO EYES?

Why multiple views?

- Structure and depth are inherently ambiguous from single views.

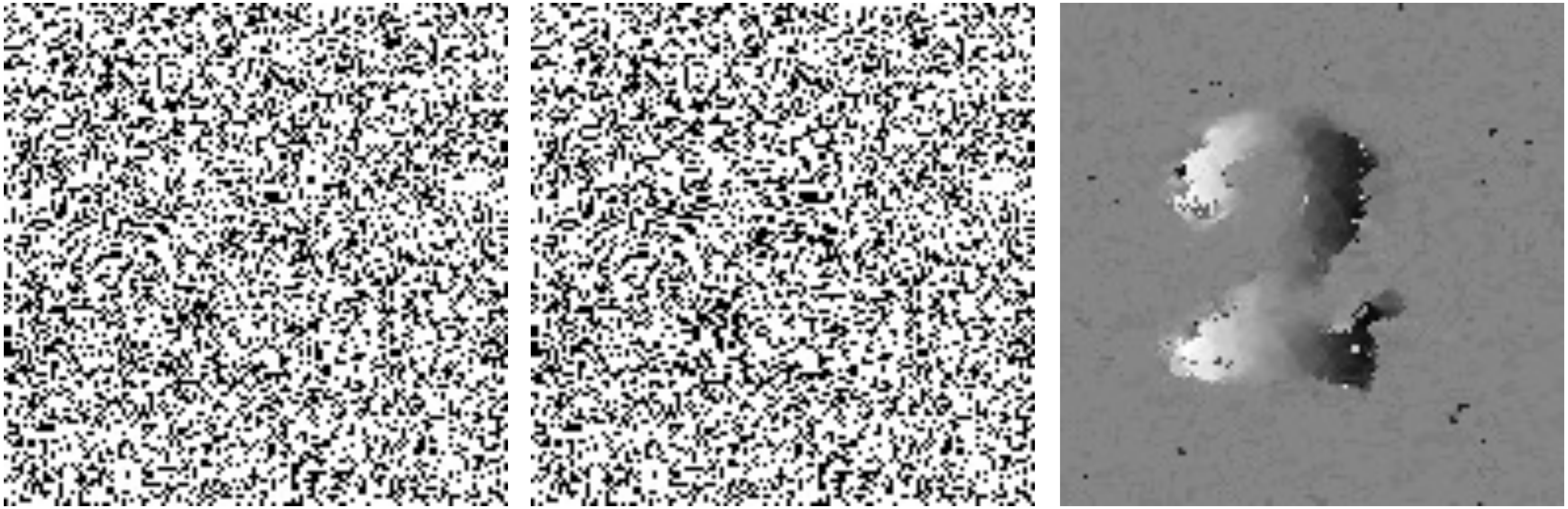


Stereo



Stereo can help dissolve ambiguity

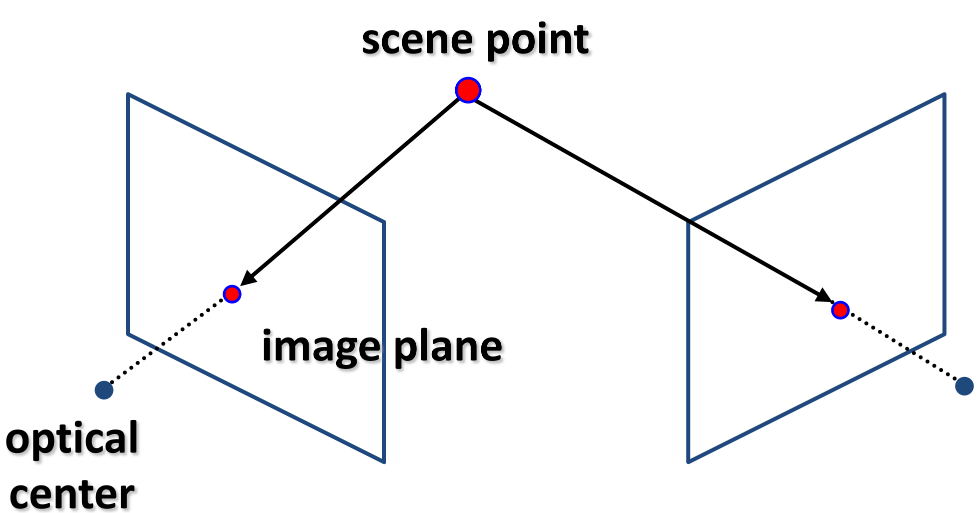
Random Dot Stereograms

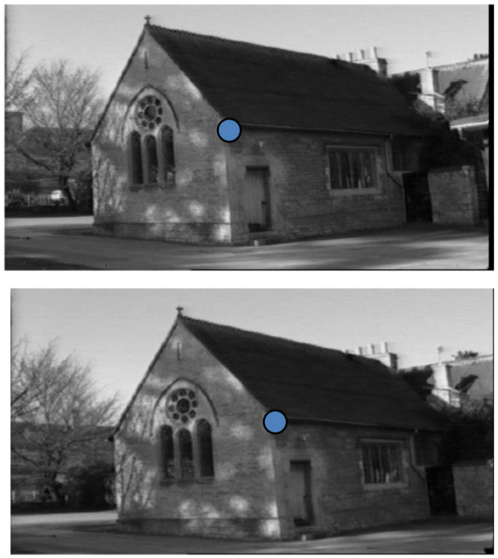


http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/HENKEL/research/stereo/rds/

Estimating depth with stereo

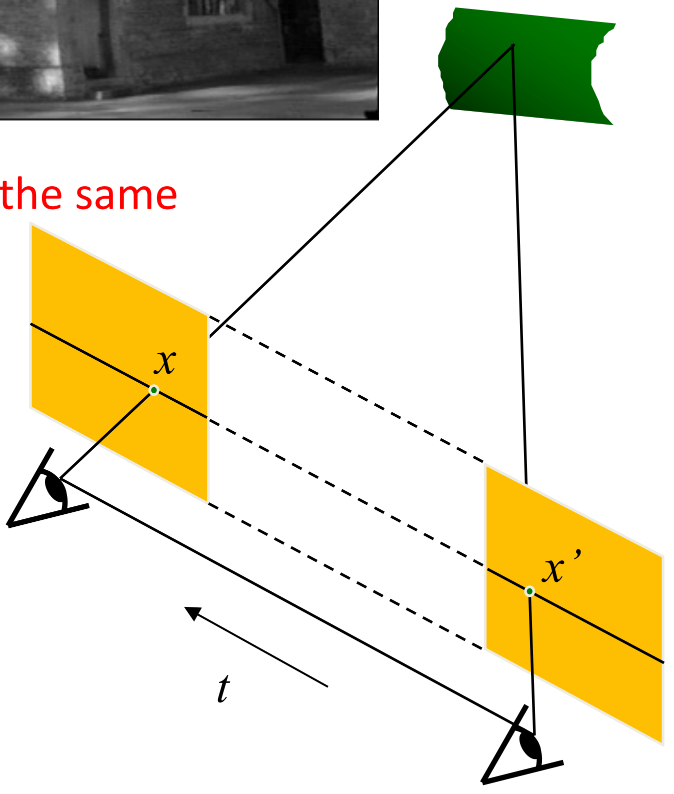
- **Stereo**: shape from "motion" between two views
- We'll need to consider:
  - Info on camera pose ("calibration")
  - Image point correspondences

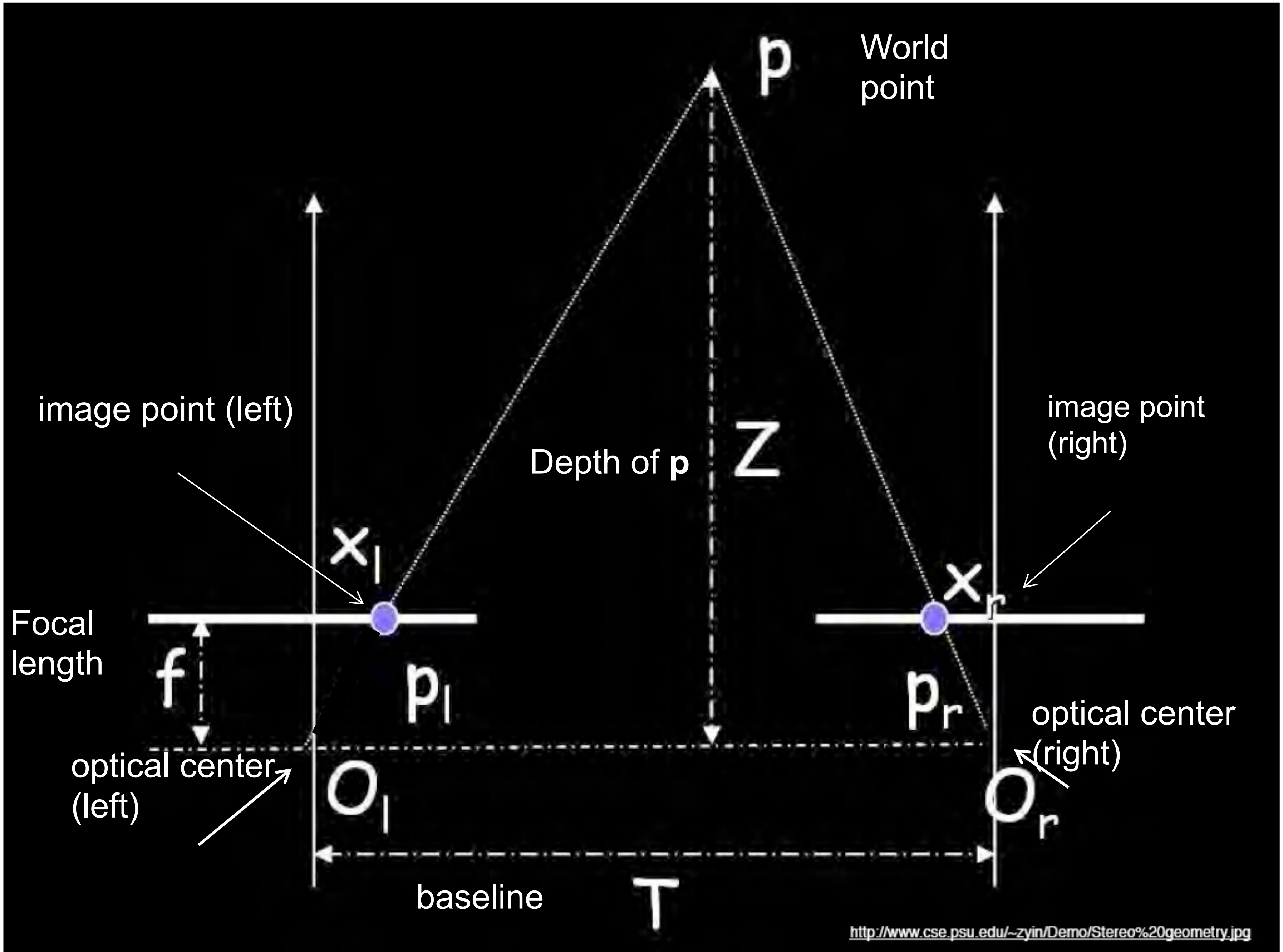




Stereo Configuration

The y-coordinates of corresponding points are the same

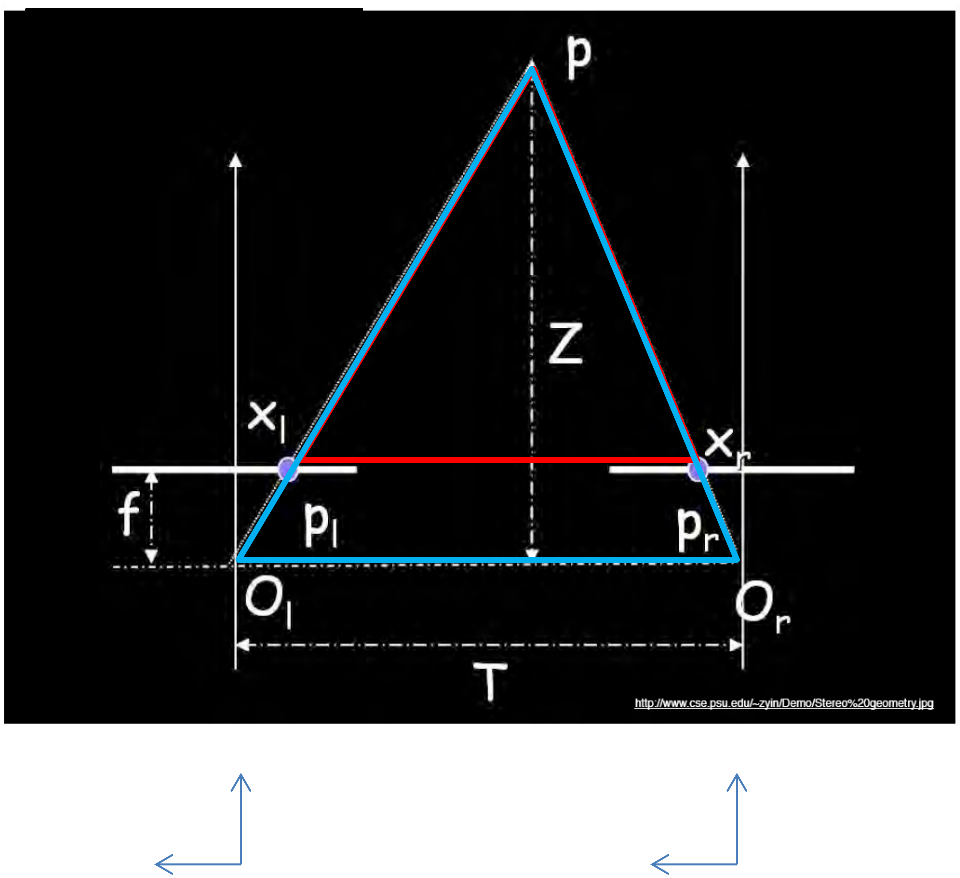




p
World point
image point (left)
Depth of **p**
Z
image point (right)
x_l
x_r
Focal length
f
p_l
p_r
optical center (right)
optical center (left)
O_l
O_r
baseline
T
http://www.cse.psu.edu/~zyin/Demo/Stereo%20geometry.jpg

Geometry for a stereo system

- Assume parallel optical axes, known camera parameters (i.e., calibrated cameras). **What is expression for Z?**



Similar triangles (p_l, P, p_r) and (O_l, P, O_r):

$$\frac{T + x_l - x_r}{Z - f} = \frac{T}{Z}$$

$$Z = f \frac{T}{x_r - x_l}$$

disparity

Depth from disparity

image I(x,y)

Disparity map D(x,y)

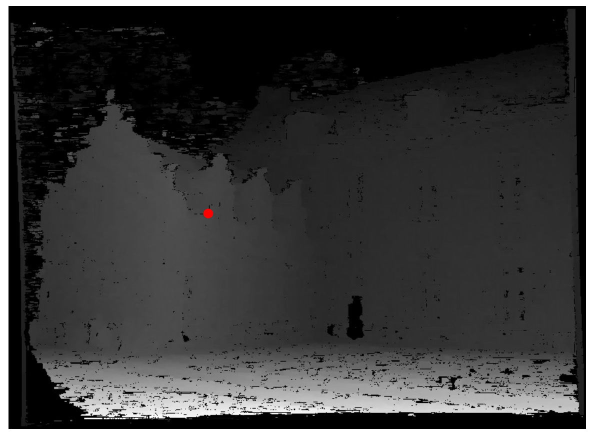
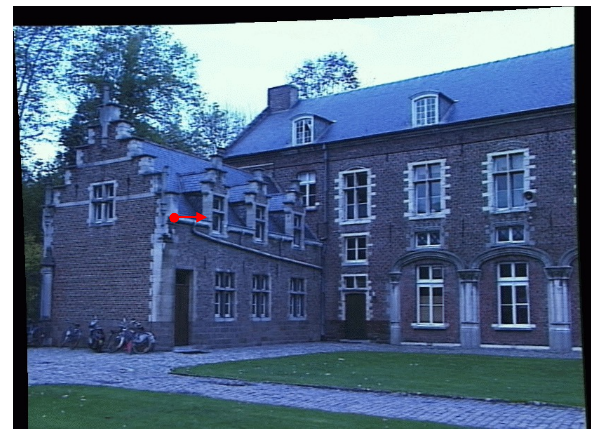


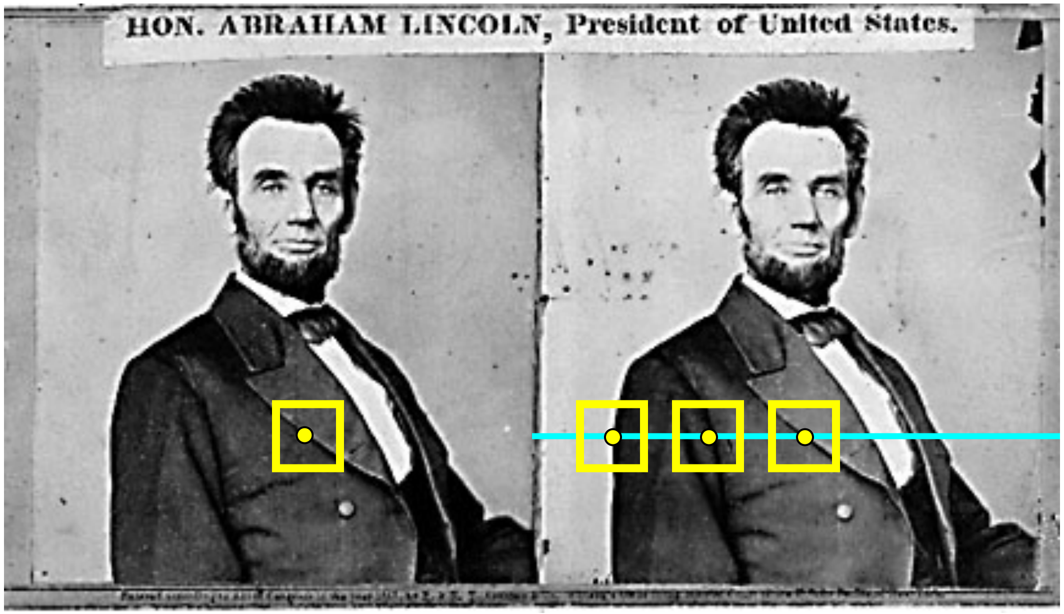
image I´(x´,y´)



(x´,y´)=(x+D(x,y), y)

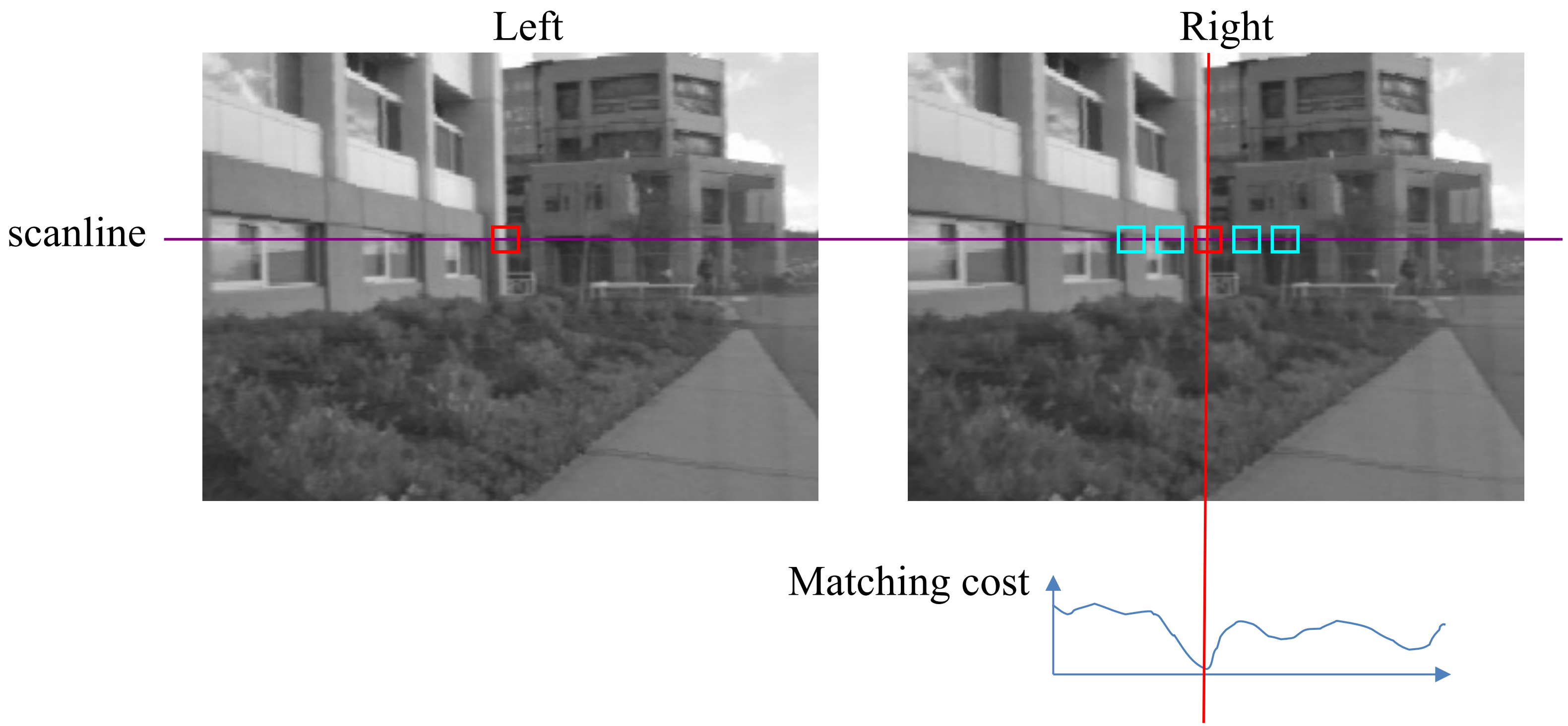
So if we could find the **corresponding points** in two images, we could **estimate relative depth**…

Basic stereo matching algorithm



- Assume each scanline in the left image is the corresponding scanline in the second image
- For each pixel x in the first image
  - Find corresponding scanline in the right image
  - Examine all pixels on the scanline and pick the best match x'
  - Compute disparity x-x' and set depth(x) = f * T/(x-x')

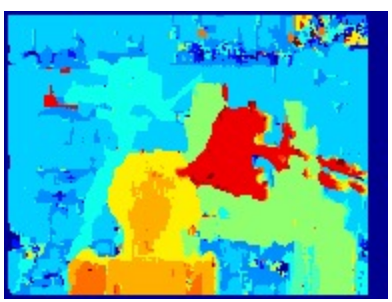
Correspondence search

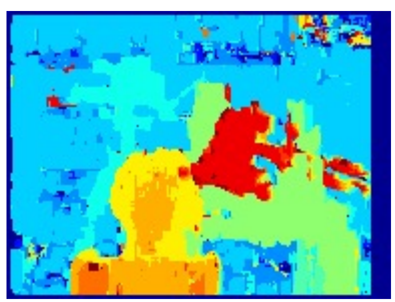


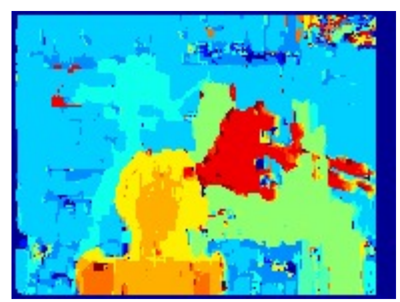
- Slide a window along the right scanline and compare contents of that window with the reference window in the left image
- Matching cost: SSD or normalized correlation

Matching windows:

Similarity Measure	Formula
Sum of Absolute Differences (SAD)	$\sum_{(i,j)\in W} \lvert I_1(i,j) - I_2(x+i,y+j)\rvert$
Sum of Squared Differences (SSD)	$\sum_{(i,j)\in W} \left(I_1(i,j) - I_2(x+i,y+j)\right)^2$
Zero-mean SAD	$\sum_{(i,j)\in W} \lvert I_1(i,j) - \bar{I}_1(i,j) - I_2(x+i,y+j) + \bar{I}_2(x+i,y+j)\rvert$
Locally scaled SAD	$\sum_{(i,j)\in W} \lvert I_1(i,j) - \frac{\bar{I}_1(i,j)}{\bar{I}_2(x+i,y+j)} I_2(x+i,y+j)\rvert$
Normalized Cross Correlation (NCC)	$\frac{\sum_{(i,j)\in W} I_1(i,j).I_2(x+i,y+j)}{\sqrt[2]{\sum_{(i,j)\in W} I_1^2(i,j).\sum_{(i,j)\in W} I_2^2(x+i,y+j)}}$


SAD

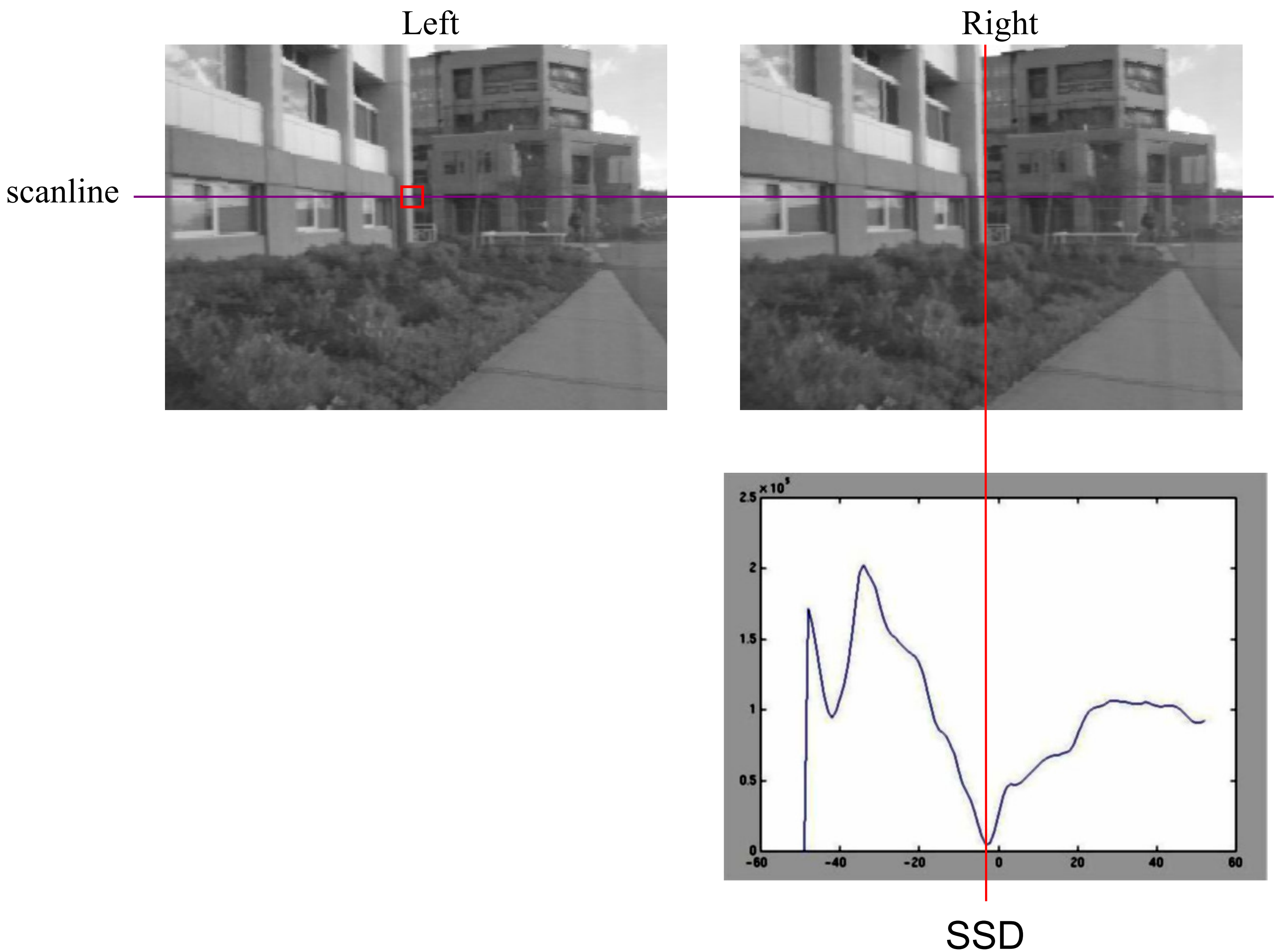

SSD


NCC

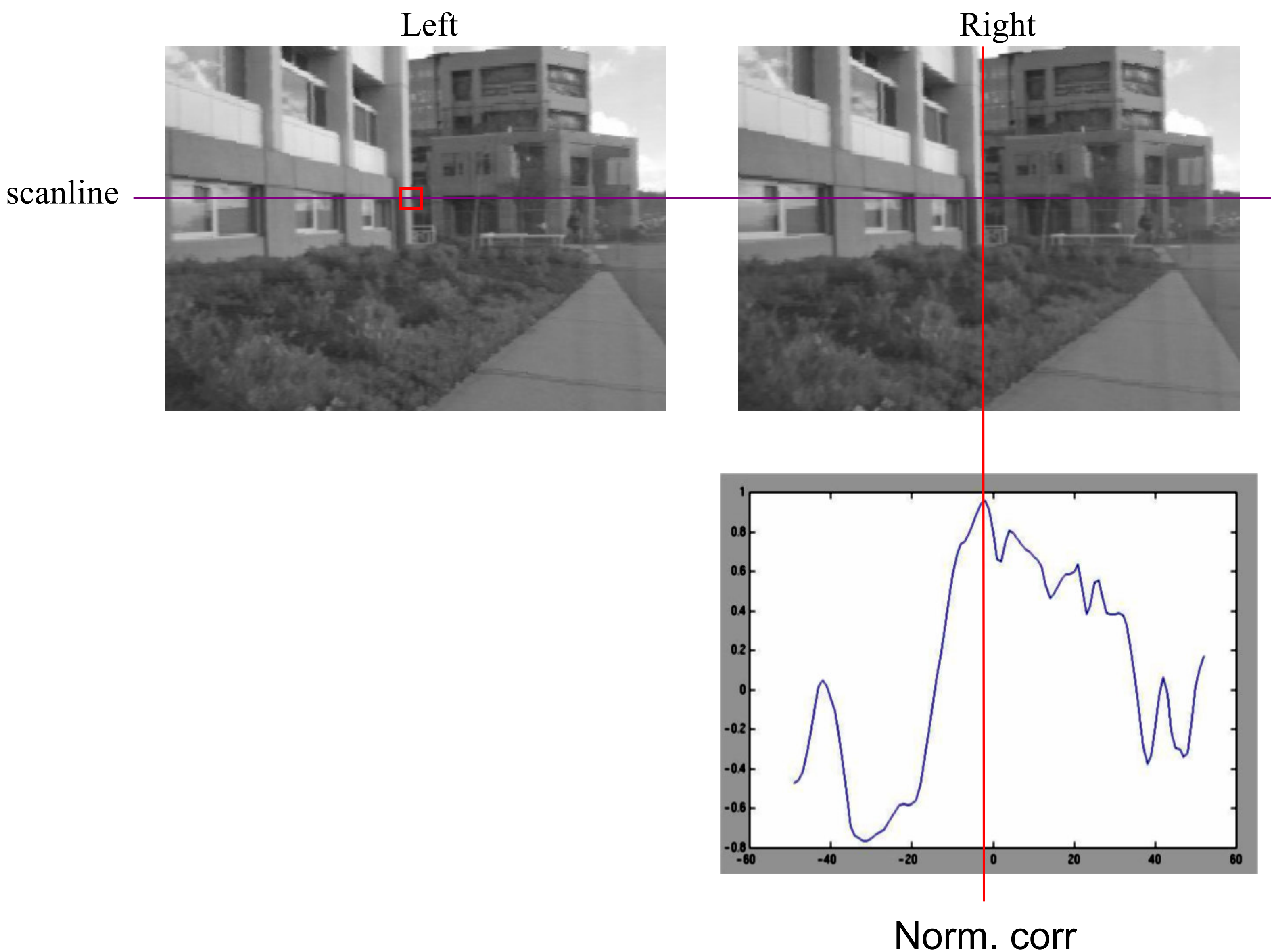

Ground truth

http://siddhantahuja.wordpress.com/category/stereo-vision/

Correspondence search

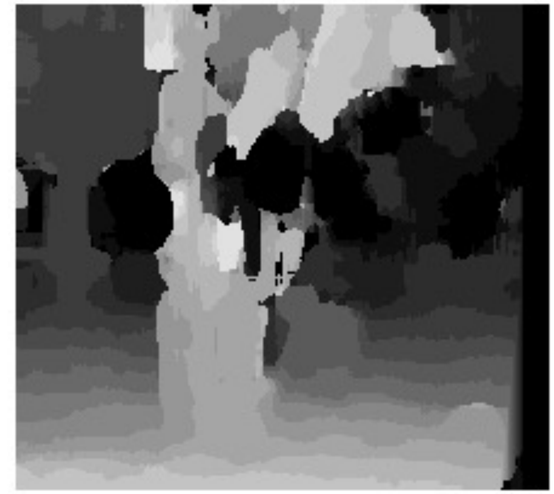


Correspondence search



Effect of window size

W = 3

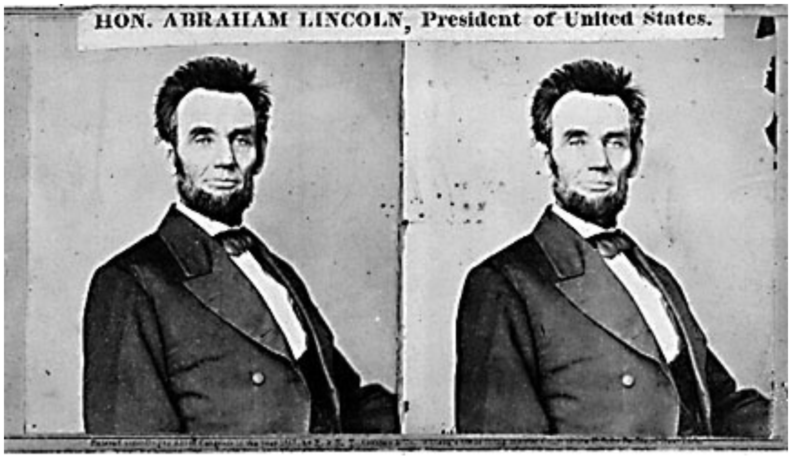


W = 20

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

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

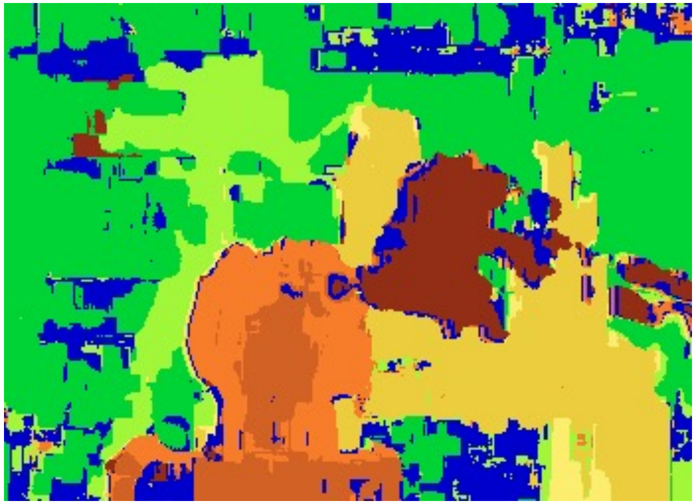
Failures of correspondence search



Results with window search

Data

Window-based matching



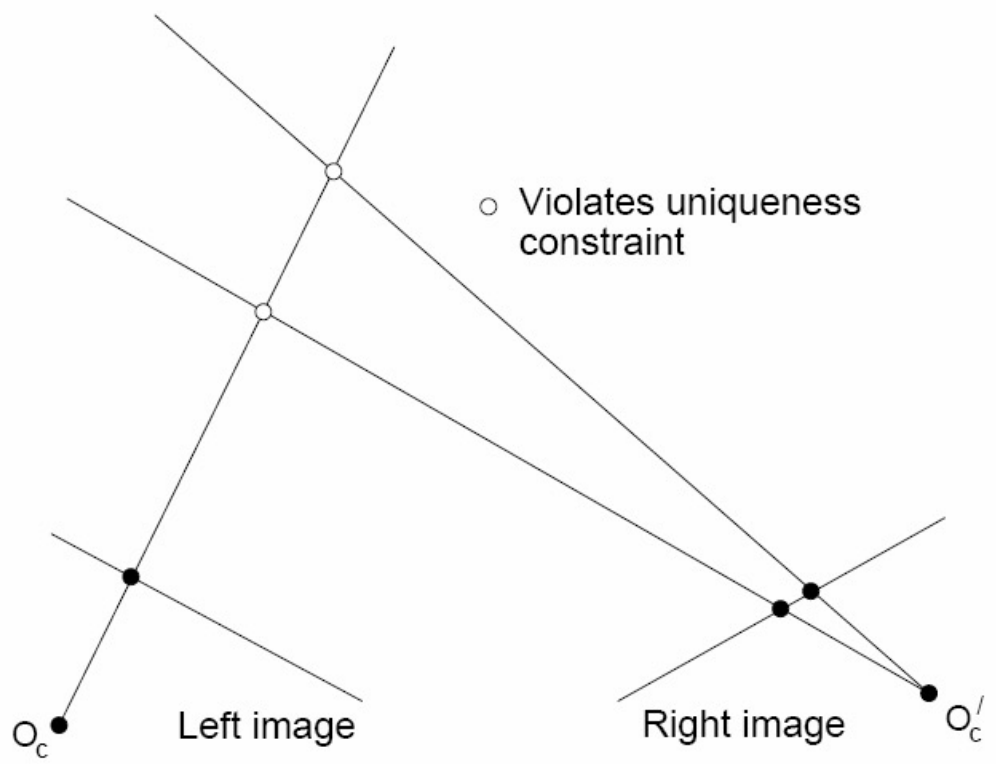
Ground truth

How can we improve window-based matching?

- So far, matches are independent for each point
- What constraints or priors can we add?

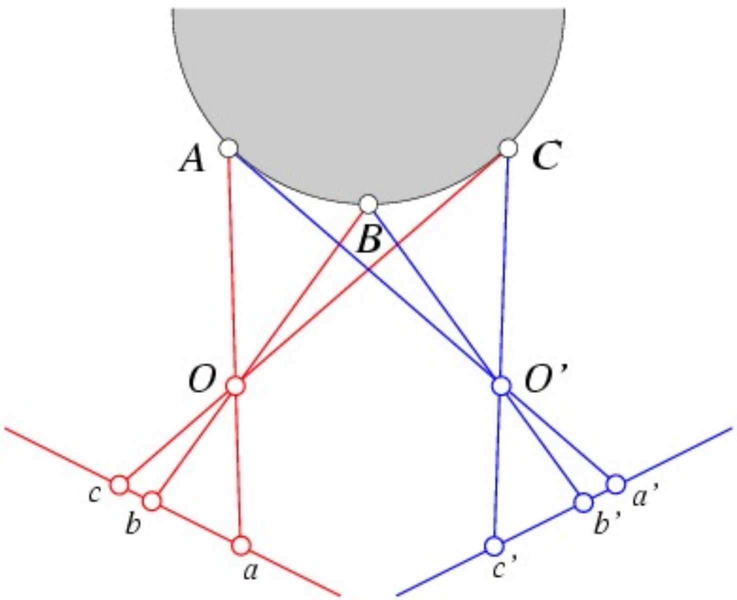
Stereo constraints/priors

- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image



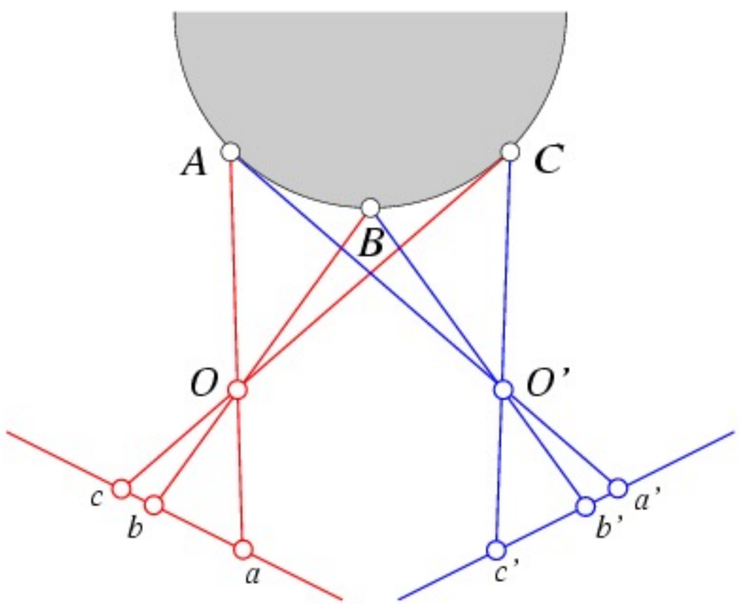
Stereo constraints/priors

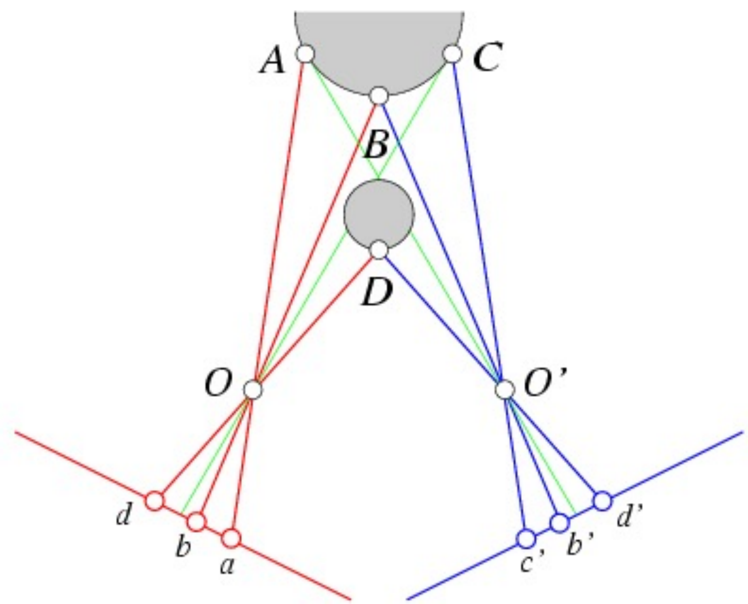
- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image
- Ordering
  - Corresponding points should be in the same order in both views



Stereo constraints/priors

- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image
- Ordering
  - Corresponding points should be in the same order in both views



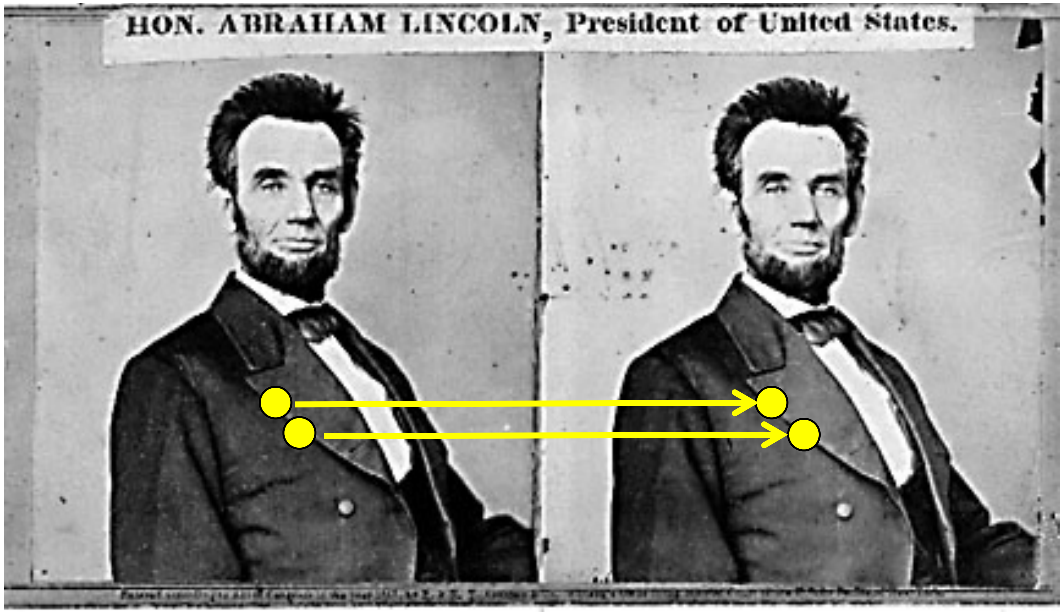


Ordering constraint doesn't hold

Priors and constraints

- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image
- Ordering
  - Corresponding points should be in the same order in both views
- Smoothness
  - We expect disparity values to change slowly (for the most part)

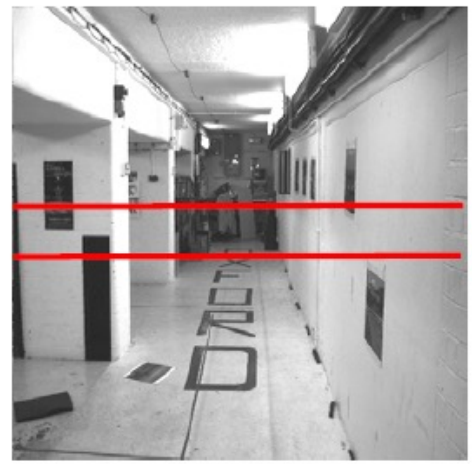
Stereo Cost Function



- 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 Cost Function

Objective: compute horizontal displacement for matches between left and right images



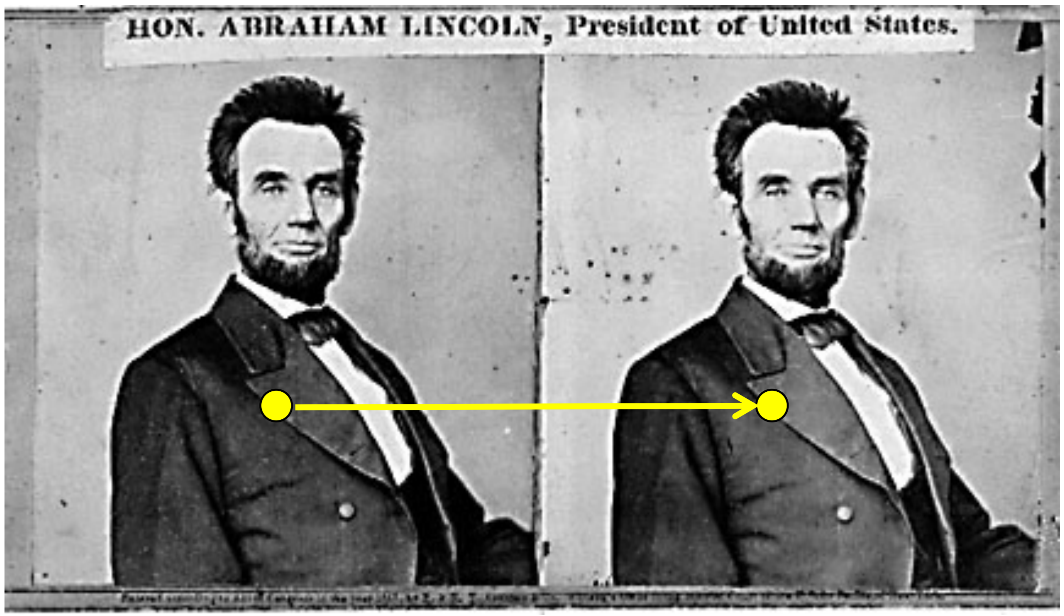
x_i is spatial shift of *i* th pixel

$$f(\mathbf{x}) = \sum_{i=1}^{n} m_i(x_i) + \sum_{i=2}^{n} \phi(x_{i-1}, x_i)$$

quality of match

uniqueness, smoothness

Stereo Cost Function

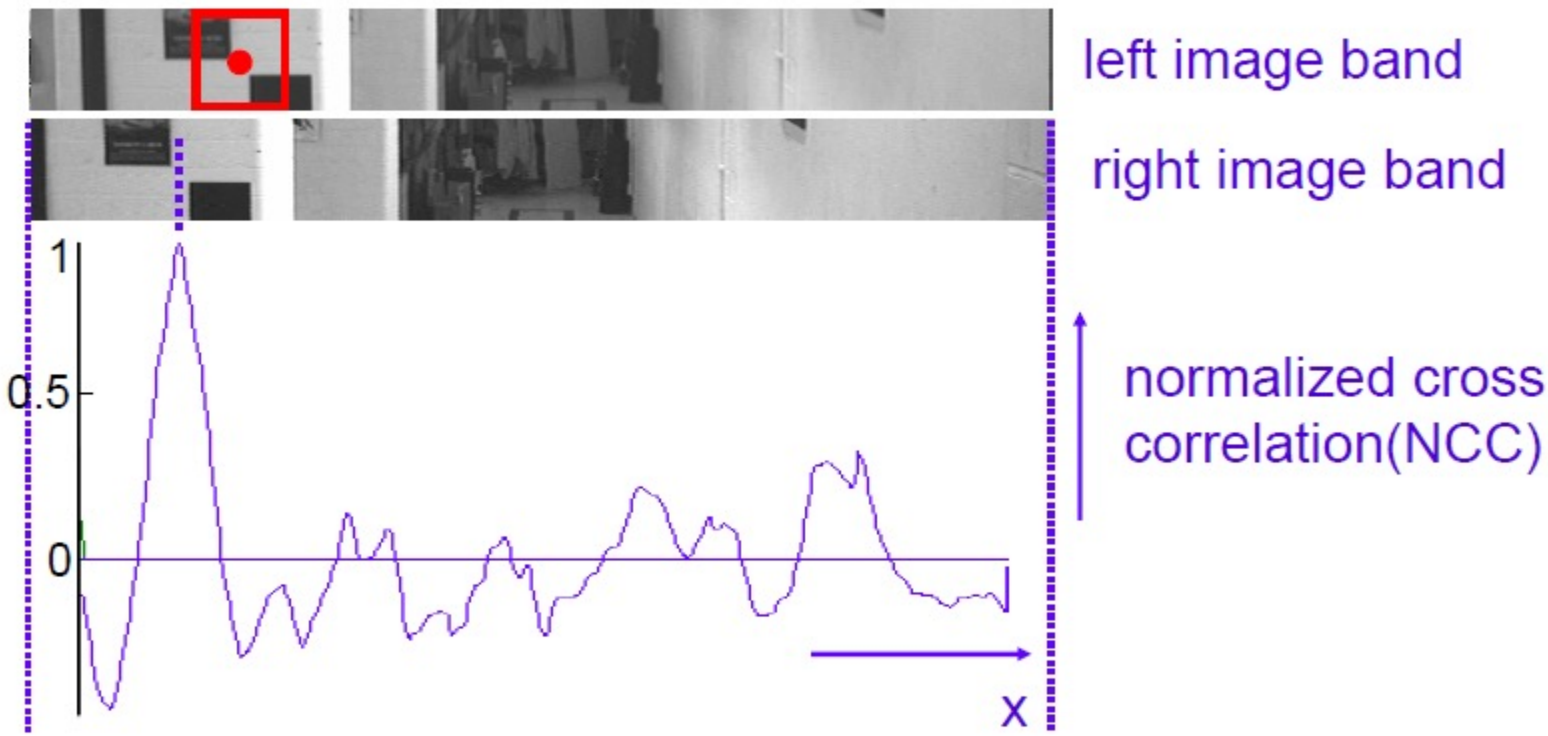


$$m_i(x_i) = I_{left}(i) - I_{right}(i + x_i)$$

$$\phi(x_{i-1}, x_i) = x_{i-1} - x_i$$

Back to Stereo Correspondence

$$m(x) = \alpha(1 - \text{NCC})^2$$



NCC of square image regions at offset (disparity) x

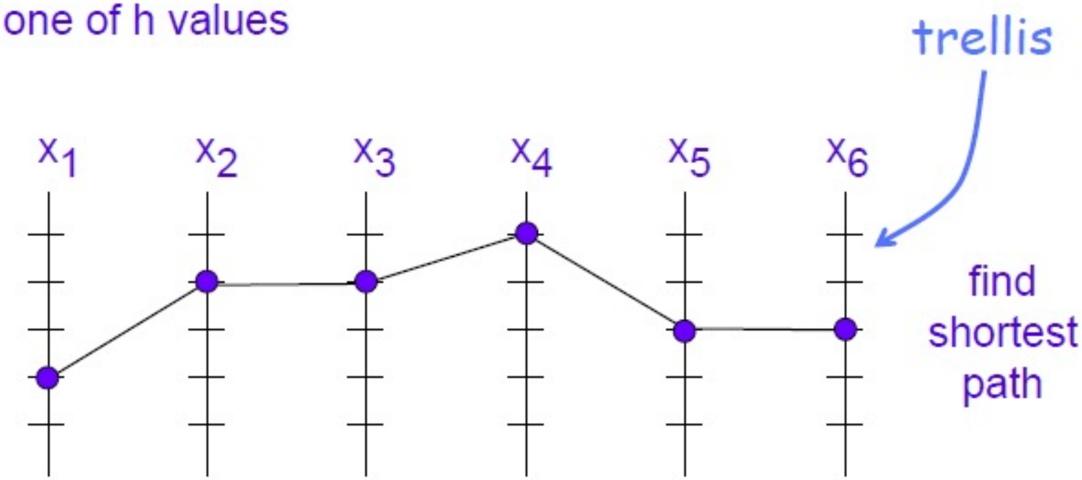
Dynamic Programming

Consider a cost function $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ of the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} m_i(x_i) + \sum_{i=2}^{n} \phi_i(x_{i-1}, x_i)$$

where x_i can take one of h values

e.g. h=5, n=6



$$f(\mathbf{x}) = \left\{ \begin{array}{c} m_1(x_1) + m_2(x_2) + m_3(x_3) + m_4(x_4) + m_5(x_5) + m_6(x_6) \\ \phi(x_1, x_2) + \phi(x_2, x_3) + \phi(x_3, x_4) + \phi(x_4, x_5) + \phi(x_5, x_6) \end{array} \right.$$

Complexity of minimization:

- exhaustive search $O(h^n)$
- dynamic programming $O(nh^2)$

Complexity of Our Stereo Correspondence

Objective: compute horizontal displacement for matches between left and right images

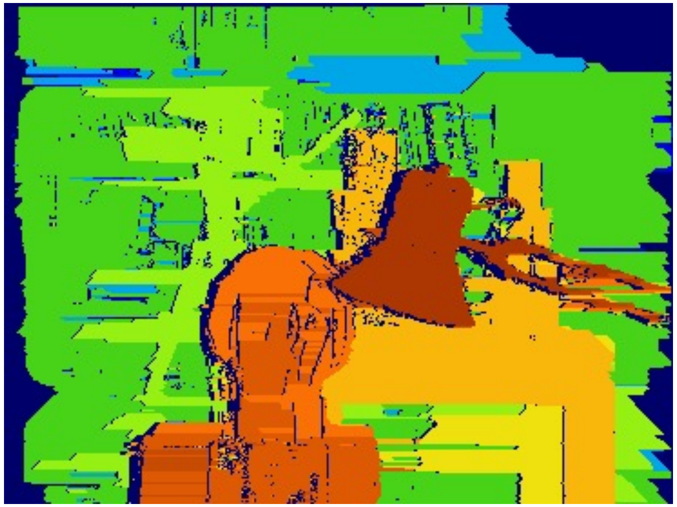
x_i is spatial shift of i'th pixel $\rightarrow h = 40$

$\mathbf{x}$ is all pixels in row $\rightarrow n = 256$

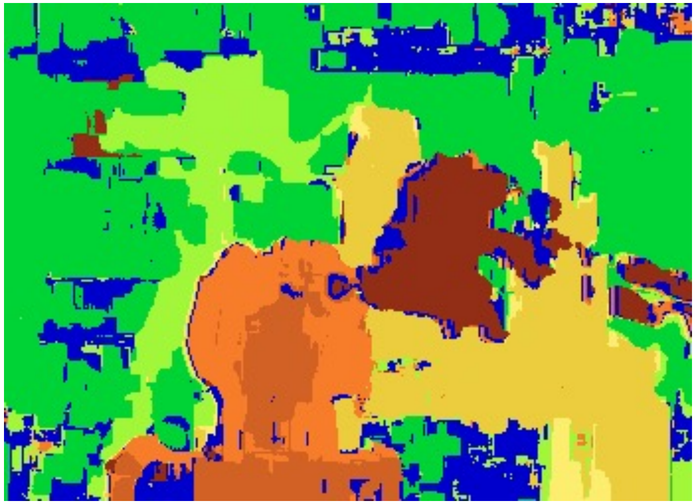
Complexity $O(40^{256})$ vs $O(256 \times 40^2)$

Coherent stereo on 2D grid

- Scanline stereo generates streaking artifacts



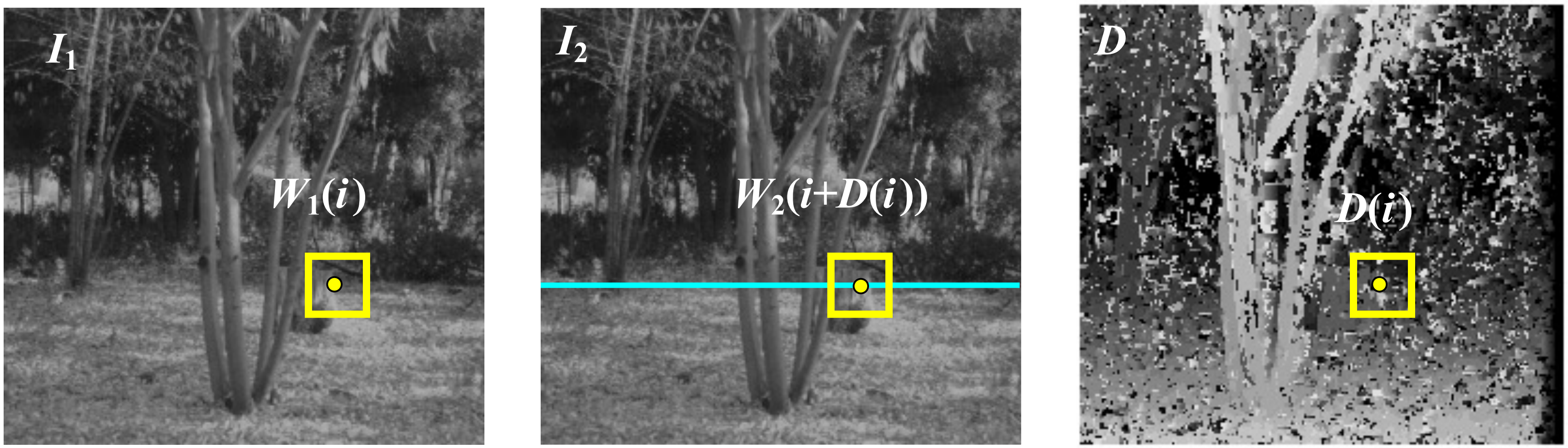
Scanline (Dynamic Programming)



Window Based

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

Stereo matching as energy minimization



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

- Random field interpretation
- Energy functions of this form can be minimized using *graph cuts*

Graph Cut Results

[Fast Approximate Energy Minimization via Graph Cuts]

http://www.middlebury.edu/stereo/