CSCI 497P/597P: Computer Vision

Scott Wehrwein

Stereo Depth Estimation, Matching

CMV: Panorama Stitching is a Solved Problem

Goals

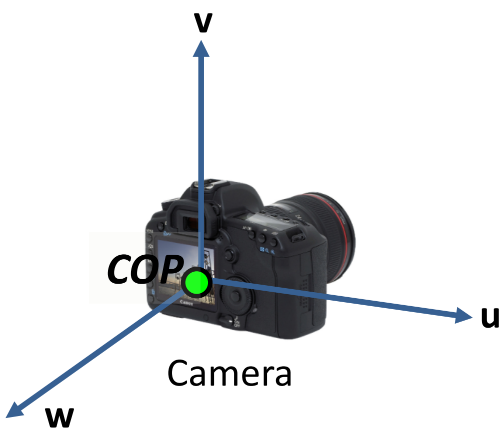
- Understand why stereo matching is the hard part of stereo vision.
- Know the definition and formation of the stereo cost volume.
- Understand the basic metrics used to compare patches (SSD, SAD, NCC)
- Understand the plane sweep stereo algorithm
- Understand the distinction between local and global methods for stereo correspondence.

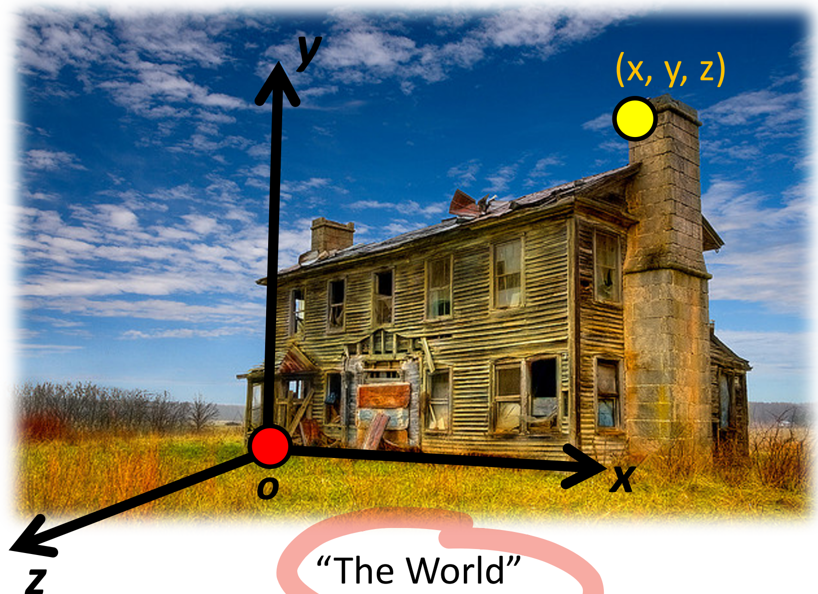
Announcements

- P1 artifact voting results coming soon…

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?



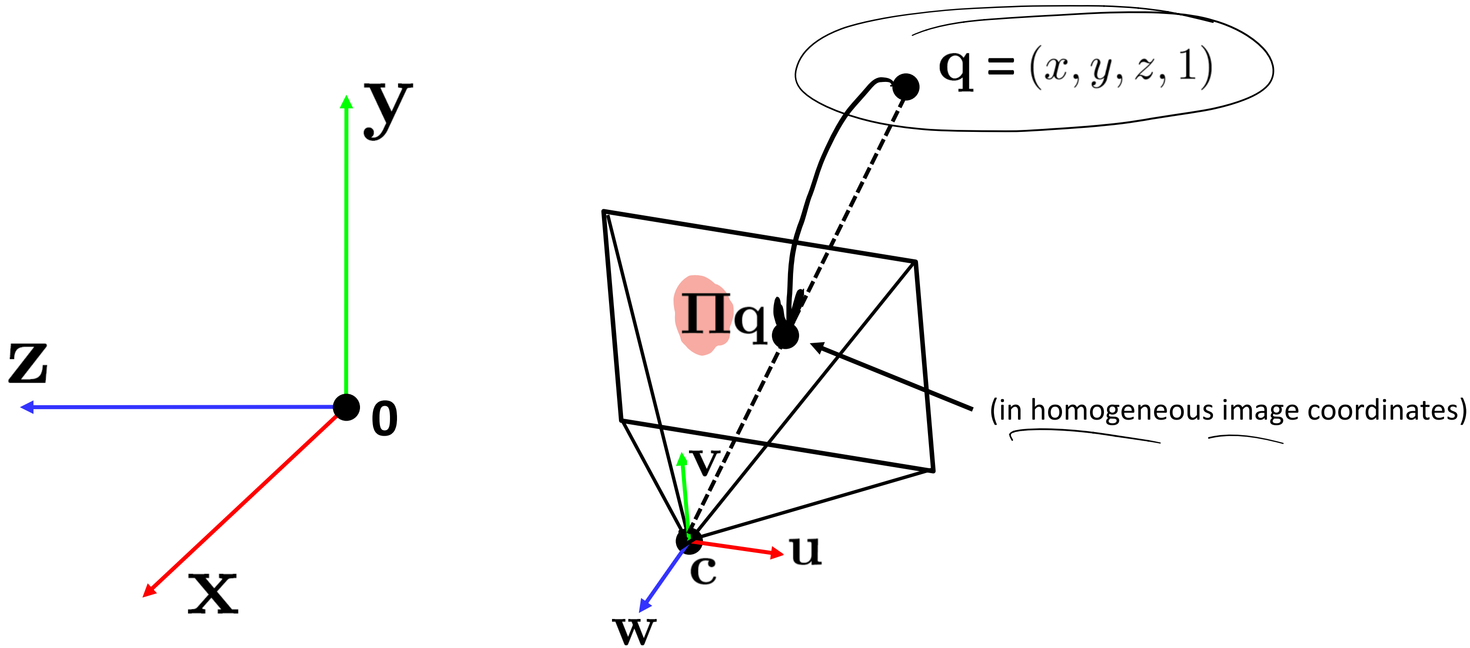


"The World"

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



Intrinsic Camera Parameters

Everything you need to get from **camera** coordinates to **pixel** coordinates:

$$\underbrace{\begin{bmatrix} -f & 0 & 0 \\ 0 & -f & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

cam

img

(intrinsics)

(converts from 3D rays in camera coordinate system to pixel coordinates)

in general, $\mathbf{K} = \begin{bmatrix} -f & s & c_x \\ 0 & -\alpha f & c_y \\ 0 & 0 & 1 \end{bmatrix}$ (upper triangular matrix)

α : **aspect ratio** (1 unless pixels are not square)

s : **skew** (0 unless pixels are shaped like rhombi/parallelograms)

(c_x, c_y) : **principal point** ((0,0) unless optical axis doesn't intersect projection plane at origin)

Extrinsic Camera Parameters

- Everything you need to get from **world** coordinates to **camera** coordinates

$$\begin{bmatrix} & \mathbf{R} & & \begin{matrix}0\\0\\0\end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} & \mathbf{I}_{3\times 3} & & -\mathbf{c} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Step 2: Rotate by **R**   Step 1: Translate by -**c**

Cam

World

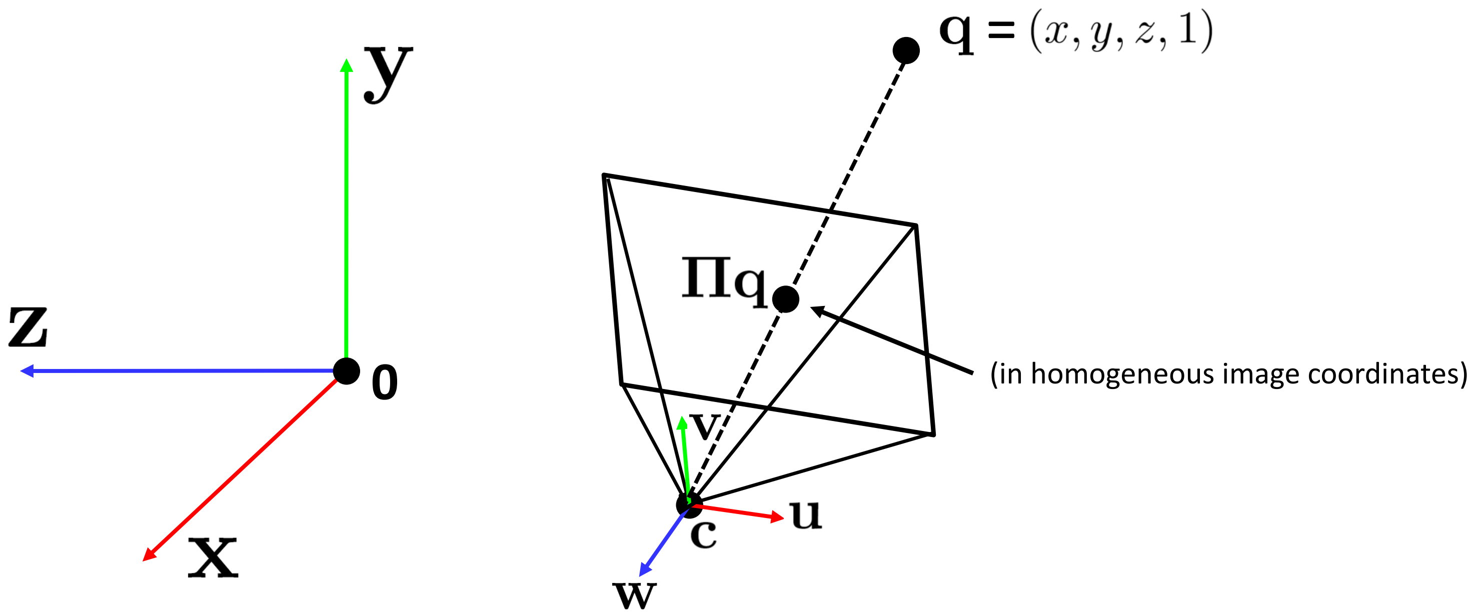
Projection matrix: Putting it all together

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

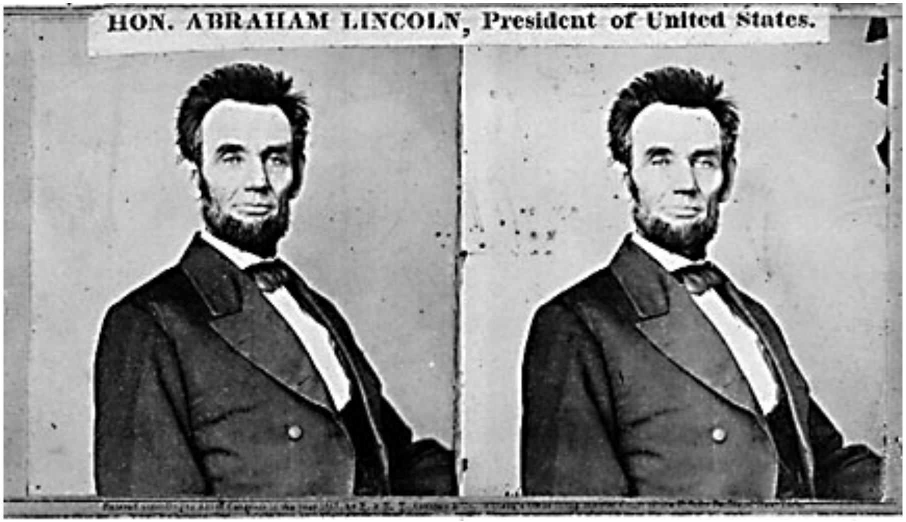
The **K** matrix converts 3D rays in the camera's coordinate system to 2D image points in image (pixel) coordinates.

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).

Projection matrix



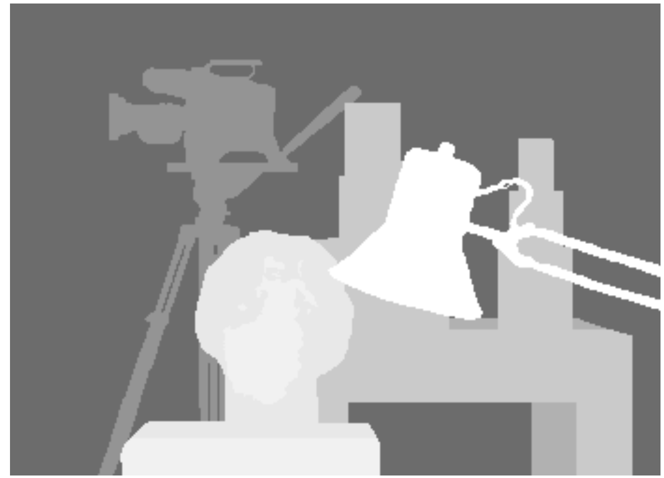
Stereo



- 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

Stereo

Left Image



Ground truth ~~depth~~ map

disparity

- 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

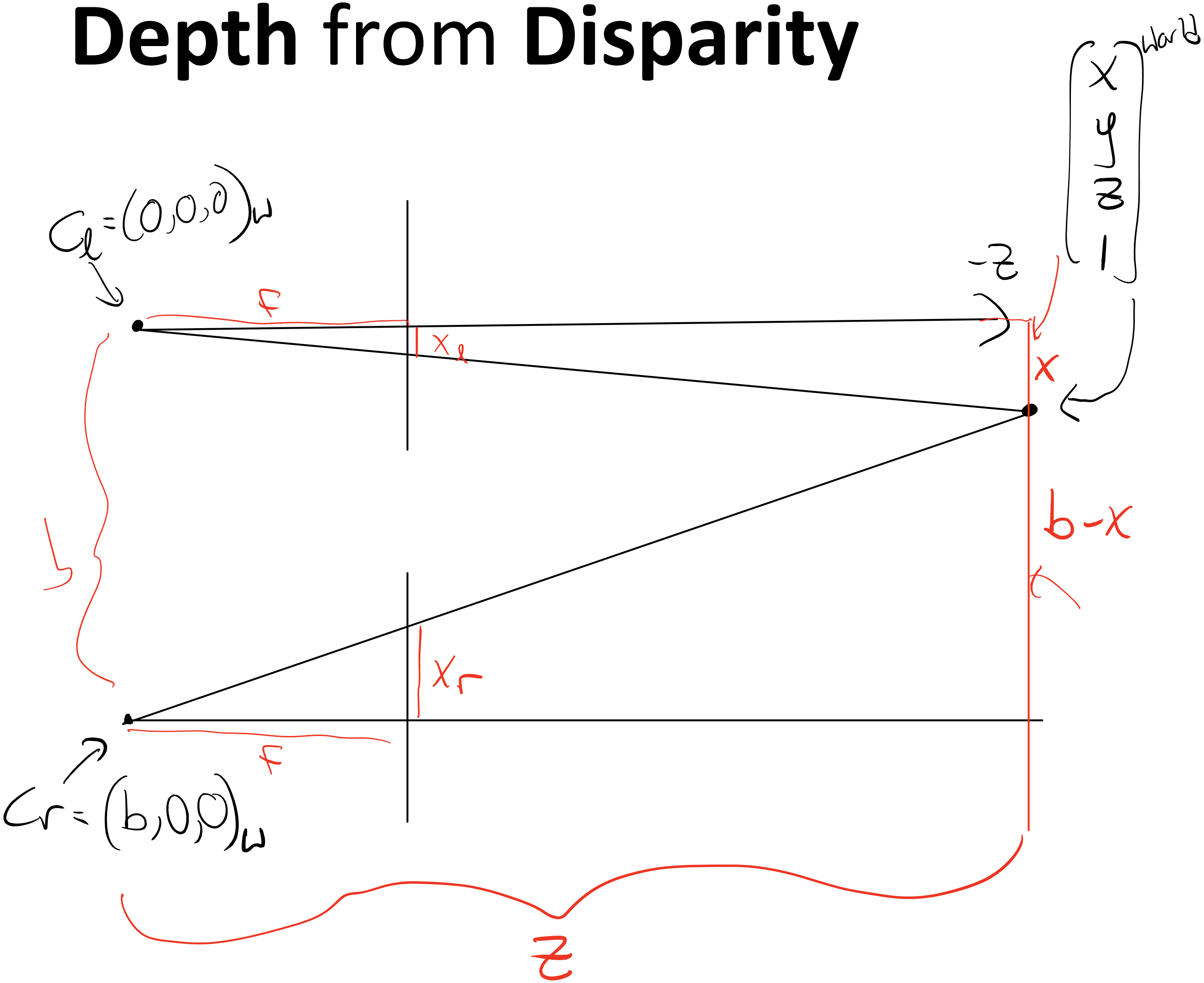
Hypothesis generation time: what relationship do you expect to find between **depth** and **how much a pixel moves**?

$$\boxed{\text{depth} \propto \frac{1}{\text{disparity}}}$$

Depth from Disparity

Assumptions

- 2 cameras
- same f
- same PP
- COP off by b in X
  ↑ known



$$\frac{z}{f} = \frac{x}{x_l} \qquad \frac{b-x}{x_r} = \frac{z}{f}$$ (similar triangles)

$$\frac{z x_l}{f} = x \qquad x = b - \frac{z x_r}{f}$$ (solve for x)

$$\frac{z x_l}{f} = b - \frac{z x_r}{f}$$ (set equal)

$$\frac{z(x_l + x_r)}{f} = b$$ (group z's)

focal length

baseline

$$\boxed{z = \frac{fb}{x_\ell + x_r}}$$

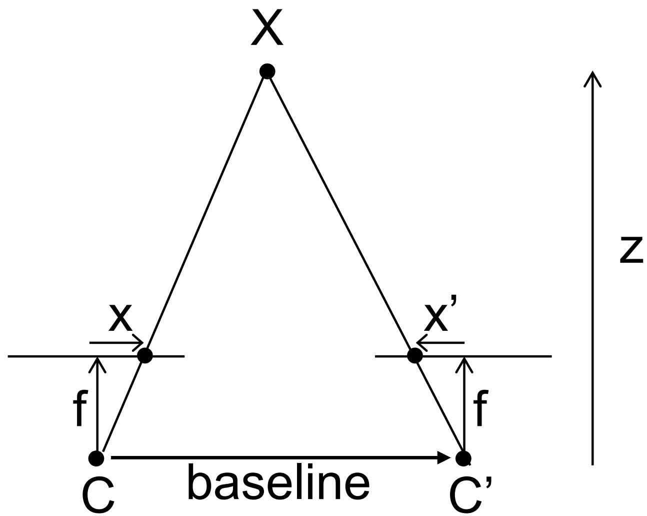
disparity

$$z \propto \frac{1}{\text{disparity}}!$$

Note: x_ℓ, x_r unsigned

If signed, disparity $= x_\ell - x_r$

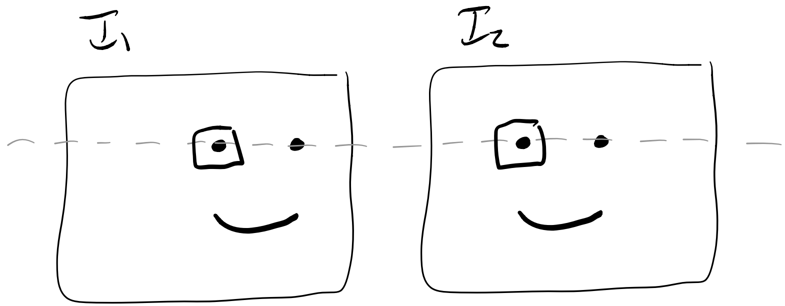
Depth from disparity



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

Stereo Depth Reconstruction: Approach

If I have rectified stereo images, ← same assumptions as before



then I can get depth if I can find correspondence.

Good news: only need to search same row!

Bad news: ambiguity abounds!

Matching is the hard part of Stereo.

Stereo Depth Reconstruction: Algorithm

of possible disparities

```
C = np.array(h, w, D)

for r in range(h):
    for c in range(w):
        for d in range(D):
            C[r, c, d] = Match_cost(Il[r,c], Ir[r,c+d])

depth = np.argmin(C, axis=2)
```

h x w

Notes:

- $\mathbf{C}$ is called the Cost Volume

- compare windows around $I_l[r,c]$, $I_r[r,c+d]$

The Cost Volume

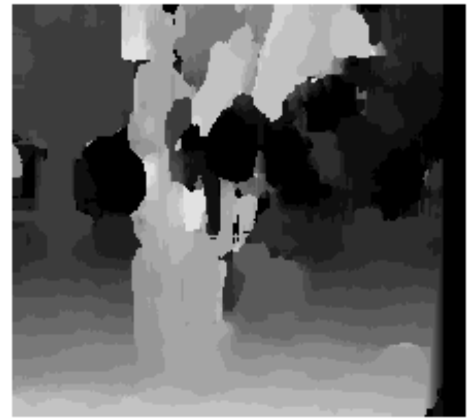
$I(x, y)$

$J(x, y)$

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

Window size

$W = 3$



$W = 20$

Effect of window size

- Smaller window
  - + better detail
  - • more noise
- Larger window
  - + less noise
  - • coarser

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

Metrics for Stereo Matching

- SSD = sum of squared differences

  np.sum((w1 - w2)**2)

- SAD = sum of absolute differences

  np.sum(np.abs(w1 - w2))

- NCC = normalized cross-correlation
  - (more ~~convolution~~cross correlation!)

Normalized Cross Correlation

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

1. subtract the mean of each vector:

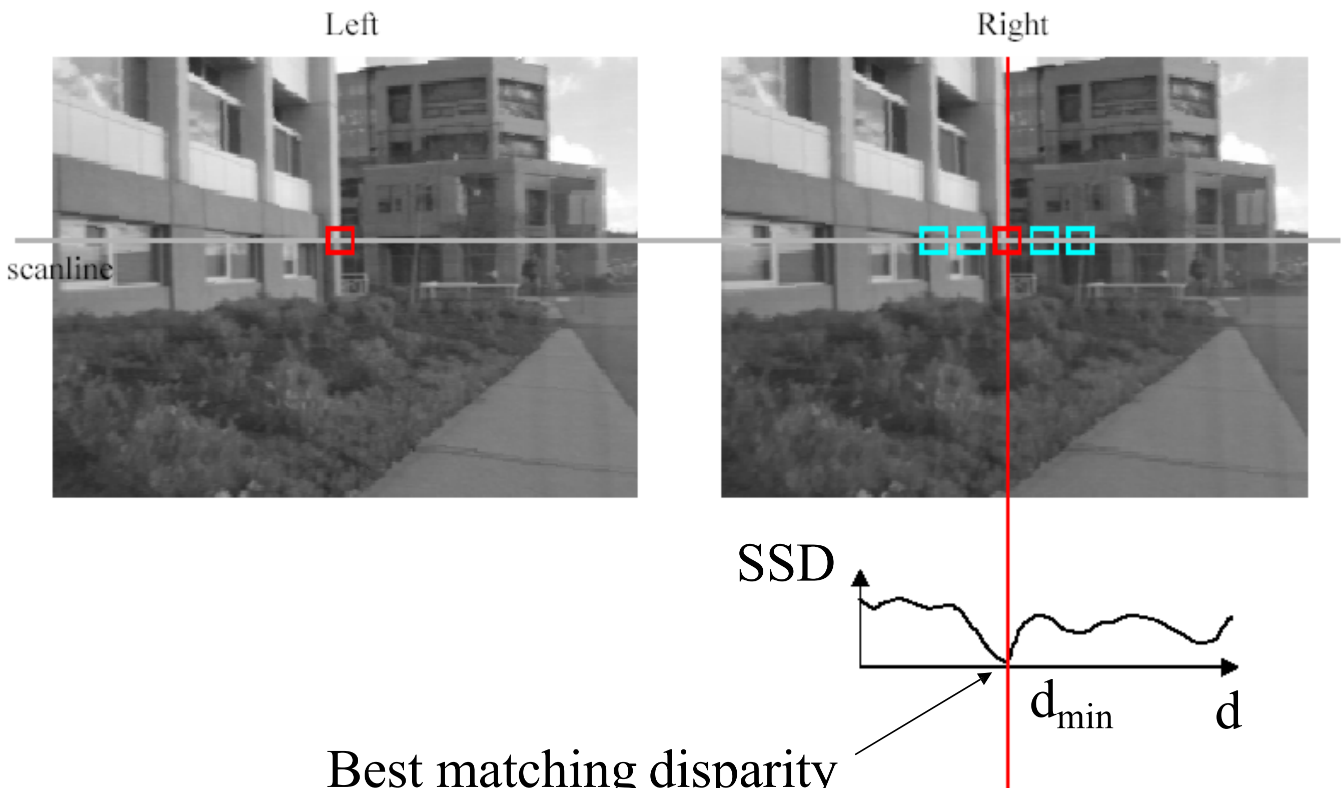
$$\frac{a}{\|a\|} \cdot \frac{b}{\|b\|}$$

$$\mathrm{a} \to \mathrm{a} - \langle \mathbf{a} \rangle, \quad \mathrm{b} \to \mathrm{b} - \langle \mathbf{b} \rangle$$

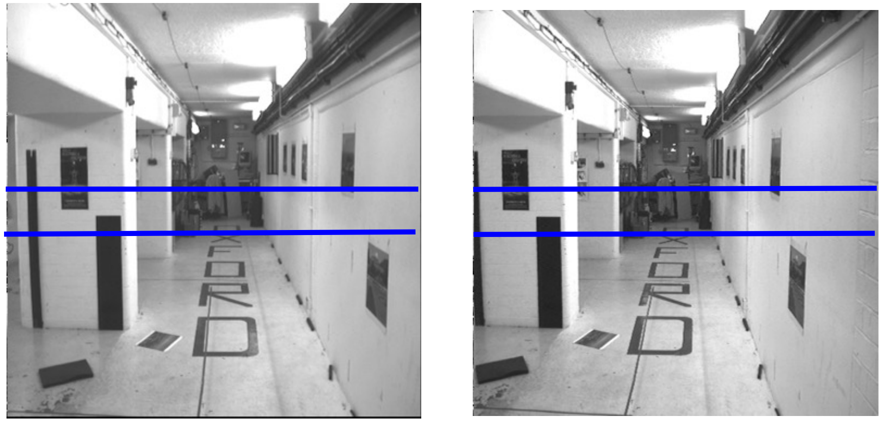
$$\text{cross correlation} = \frac{\mathbf{a}.\mathbf{b}}{|\mathbf{a}||\mathbf{b}|}$$

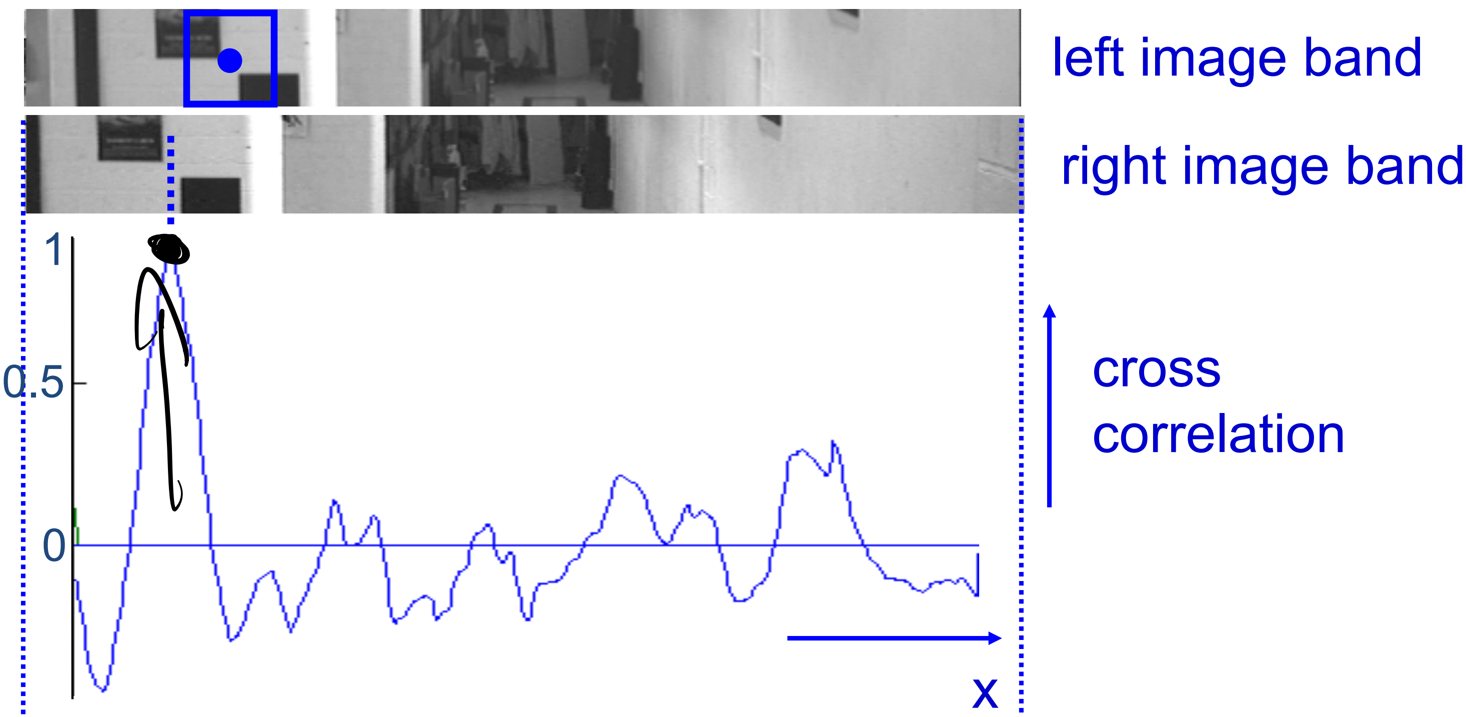
Invariant to $I \to \alpha I + \beta$

Stereo matching based on SSD

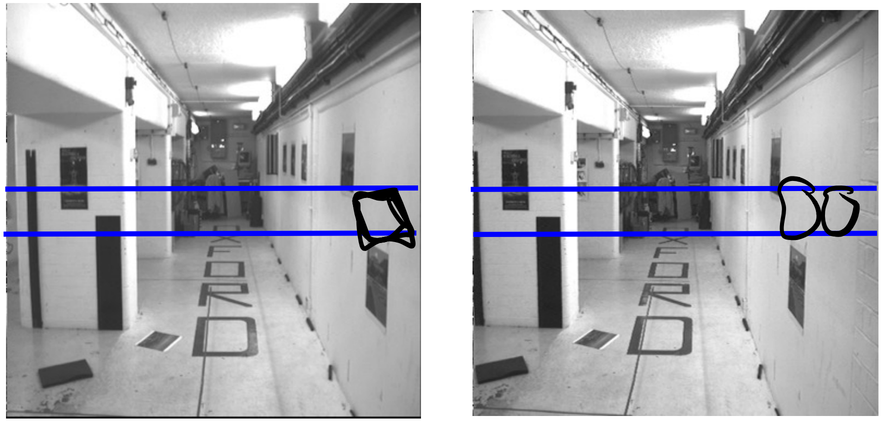


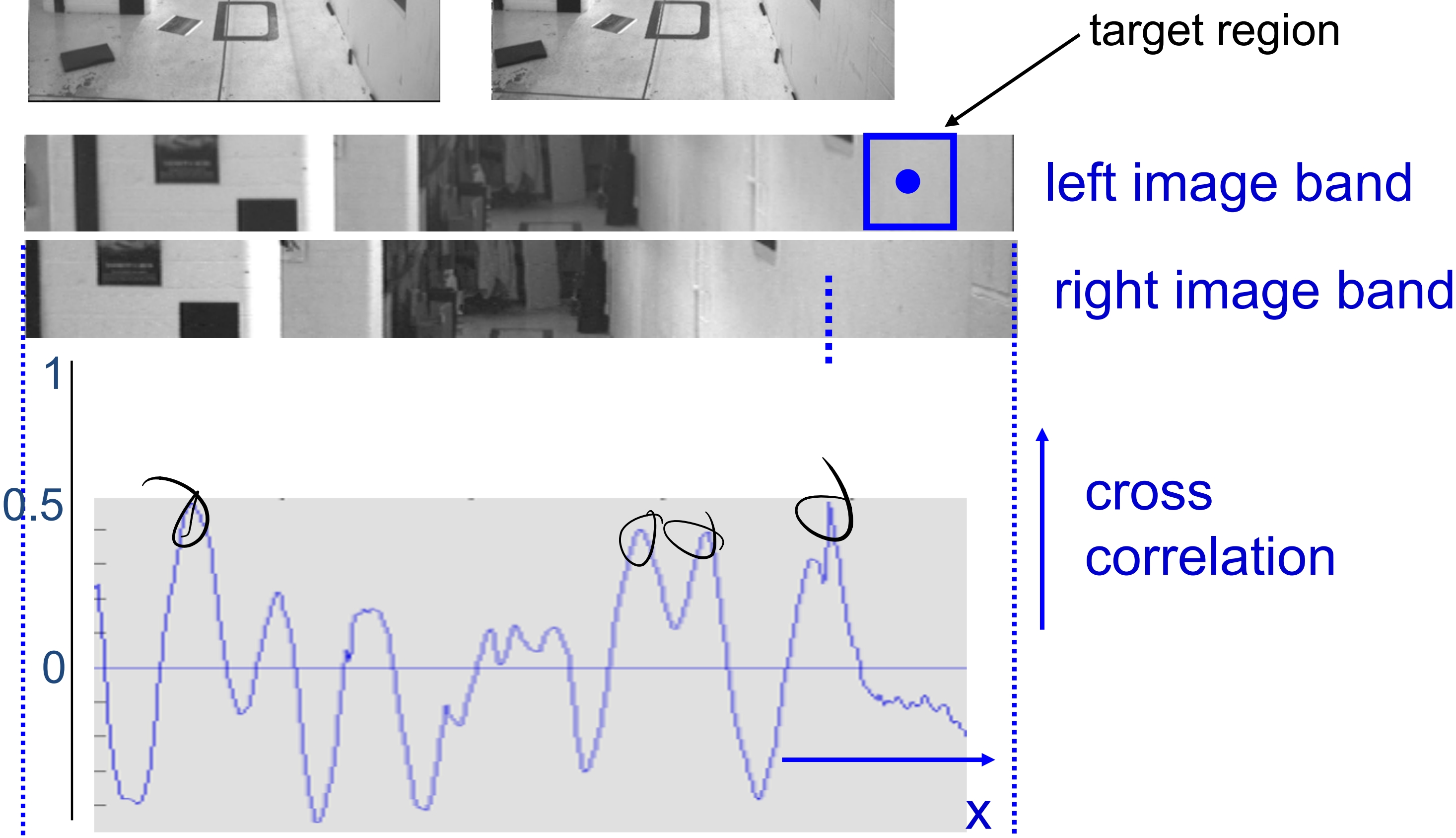
Stereo with NCC: The Good Case





Stereo with NCC: The Bad Case





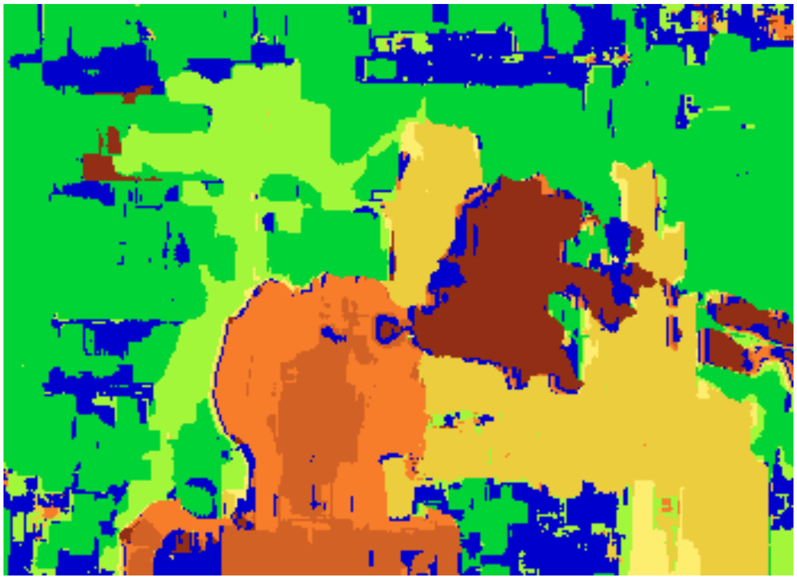
Stereo results

- 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

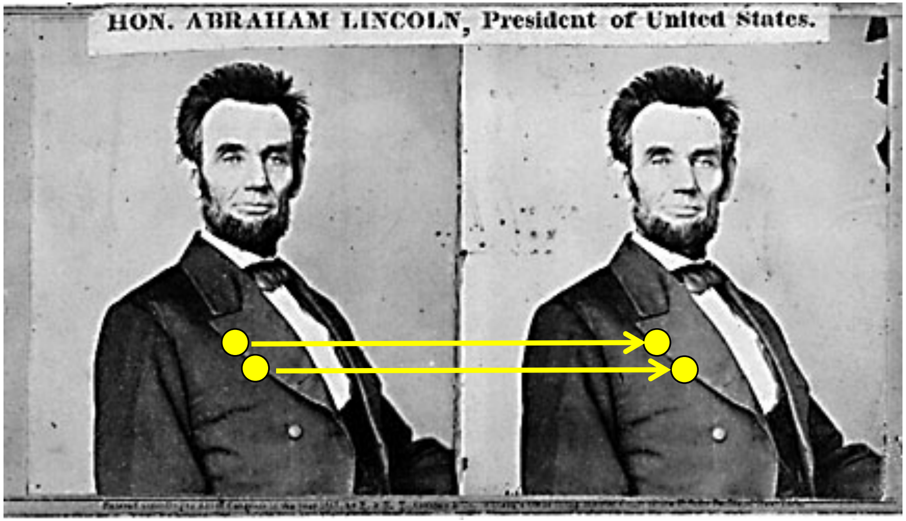
Better methods exist...

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/

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

$I(x, y)$

$J(x, y)$

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

Greedy selection of best match