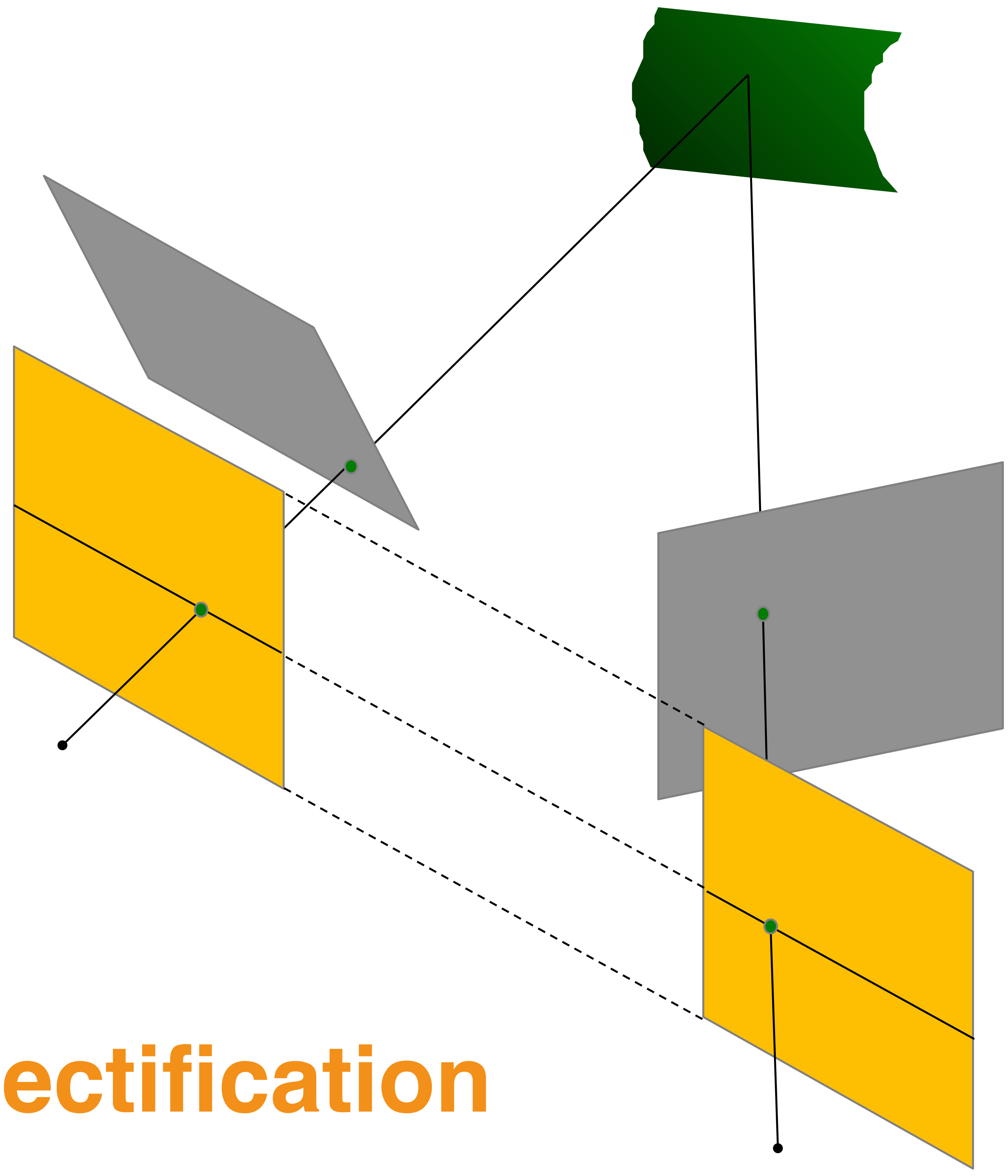
# Stereo Matching

16-385 Computer Vision (Kris Kitani)

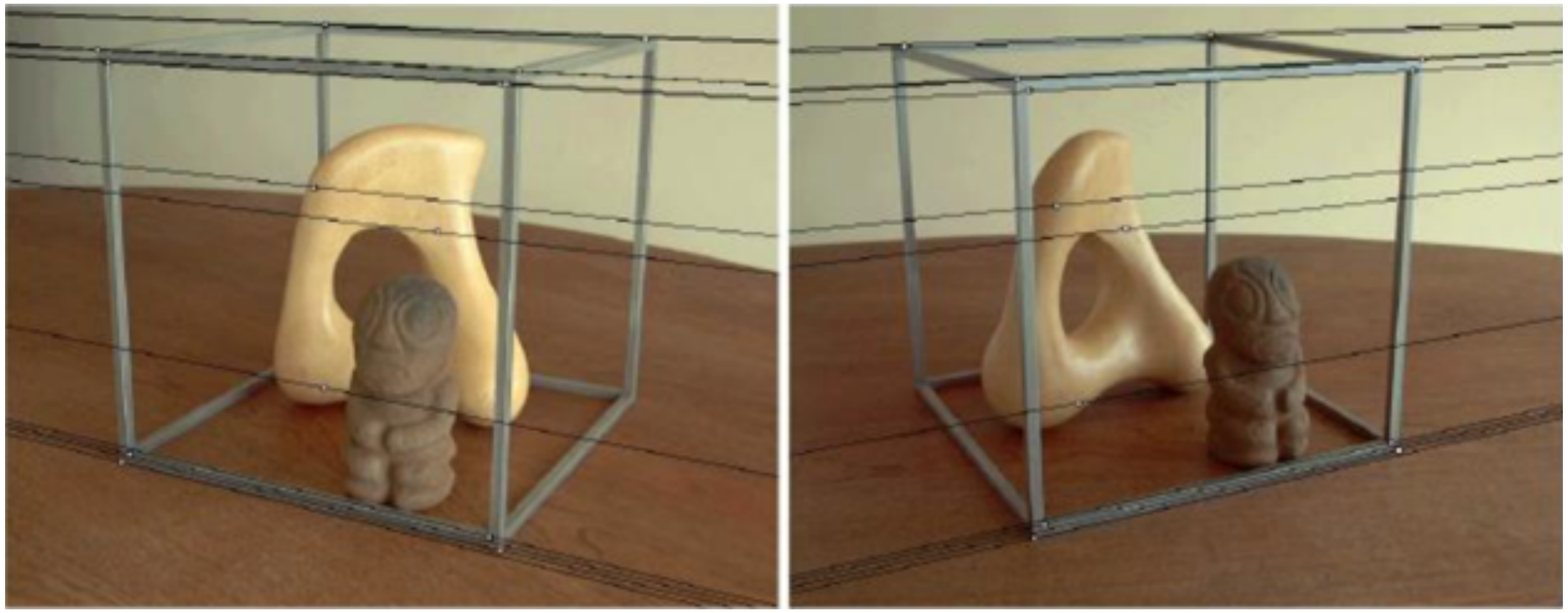
**Carnegie Mellon University**

*What is stereo rectification?*

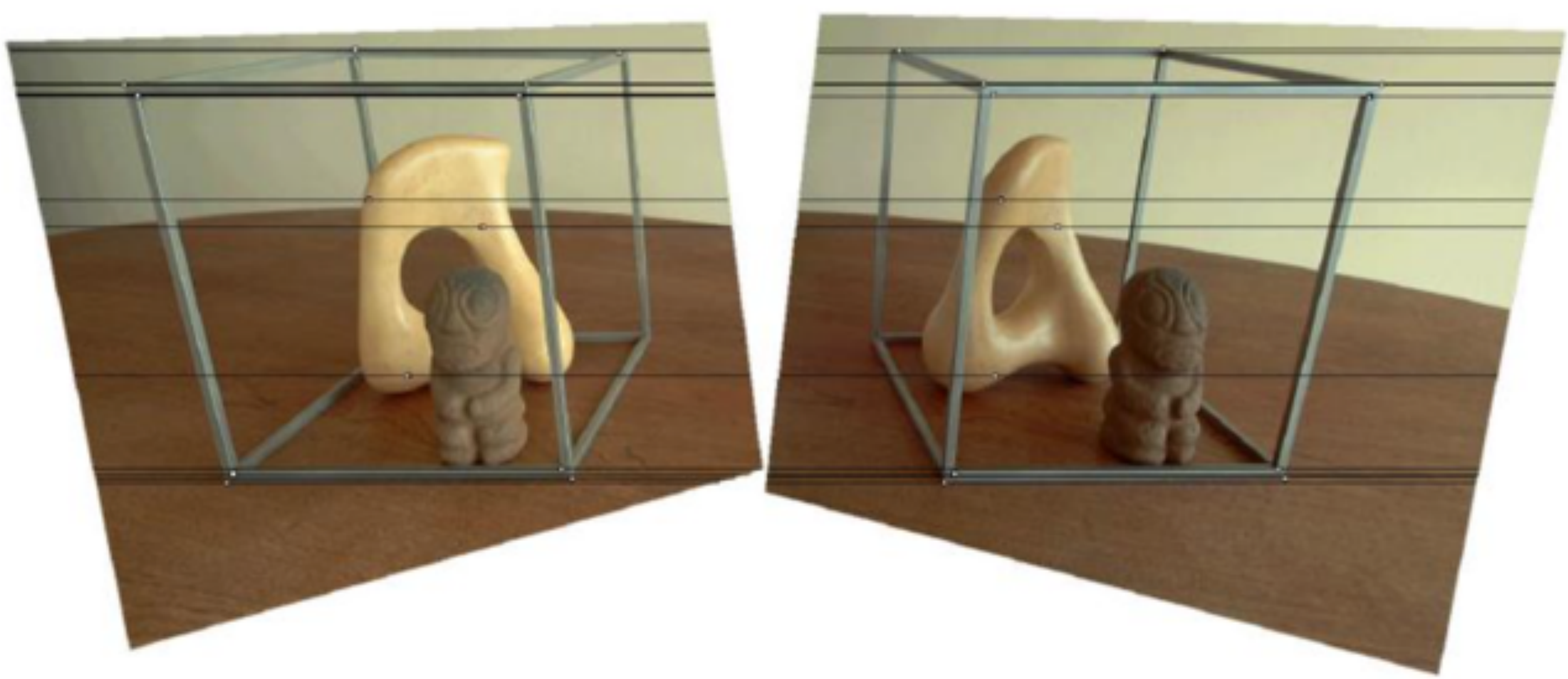
Reproject image planes onto a common plane parallel to the line between camera centers



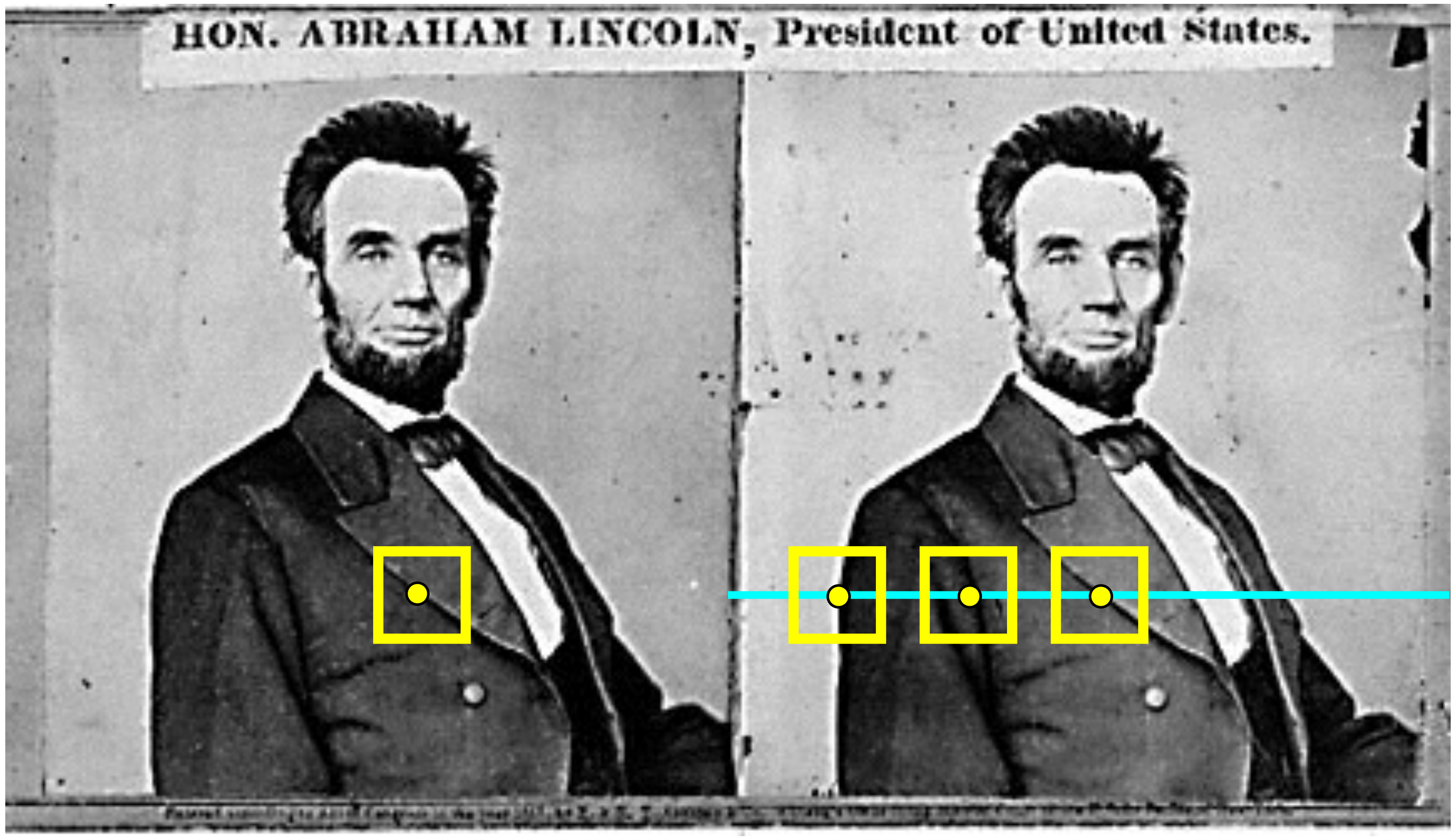
# Recall: Stereo Rectification



# What can we do after rectification?



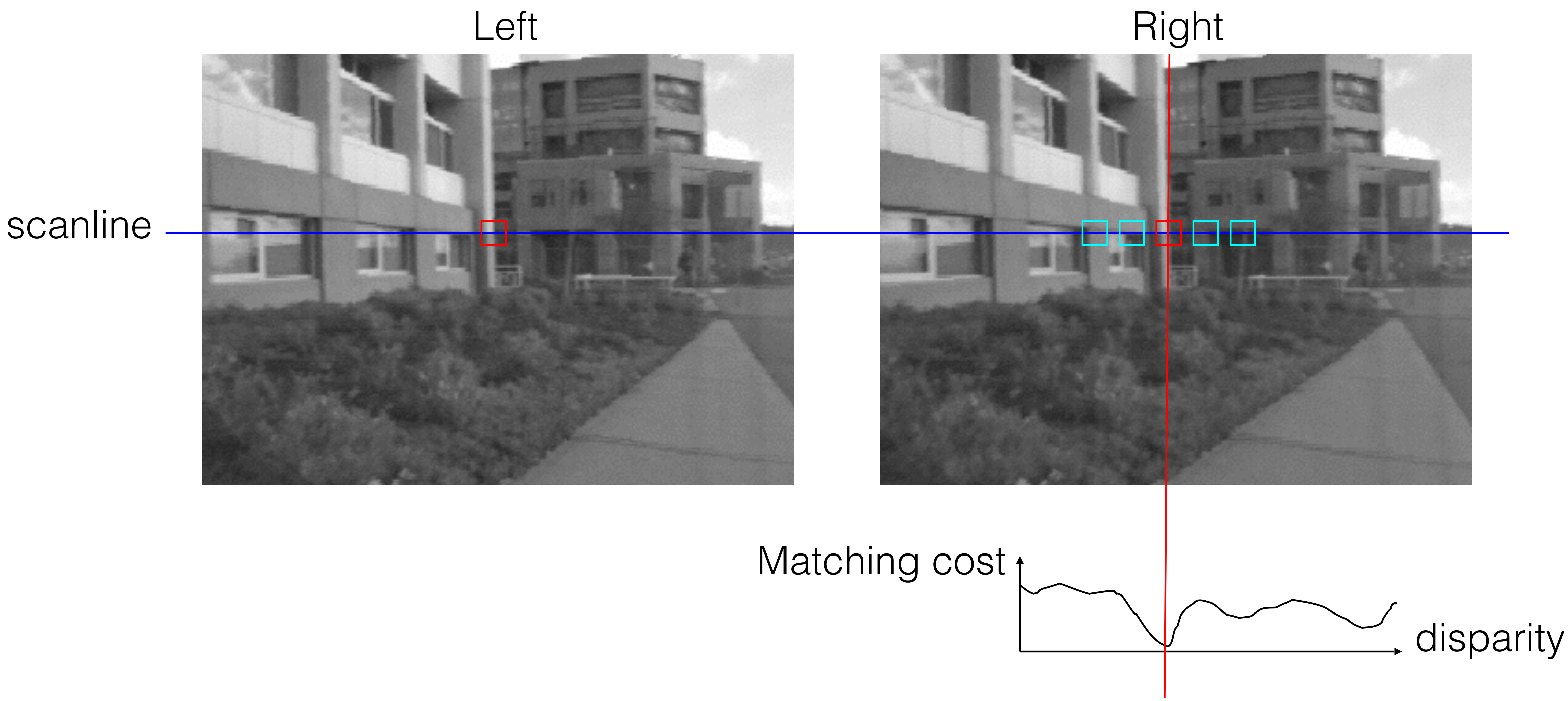
Depth Estimation via  Stereo Matching



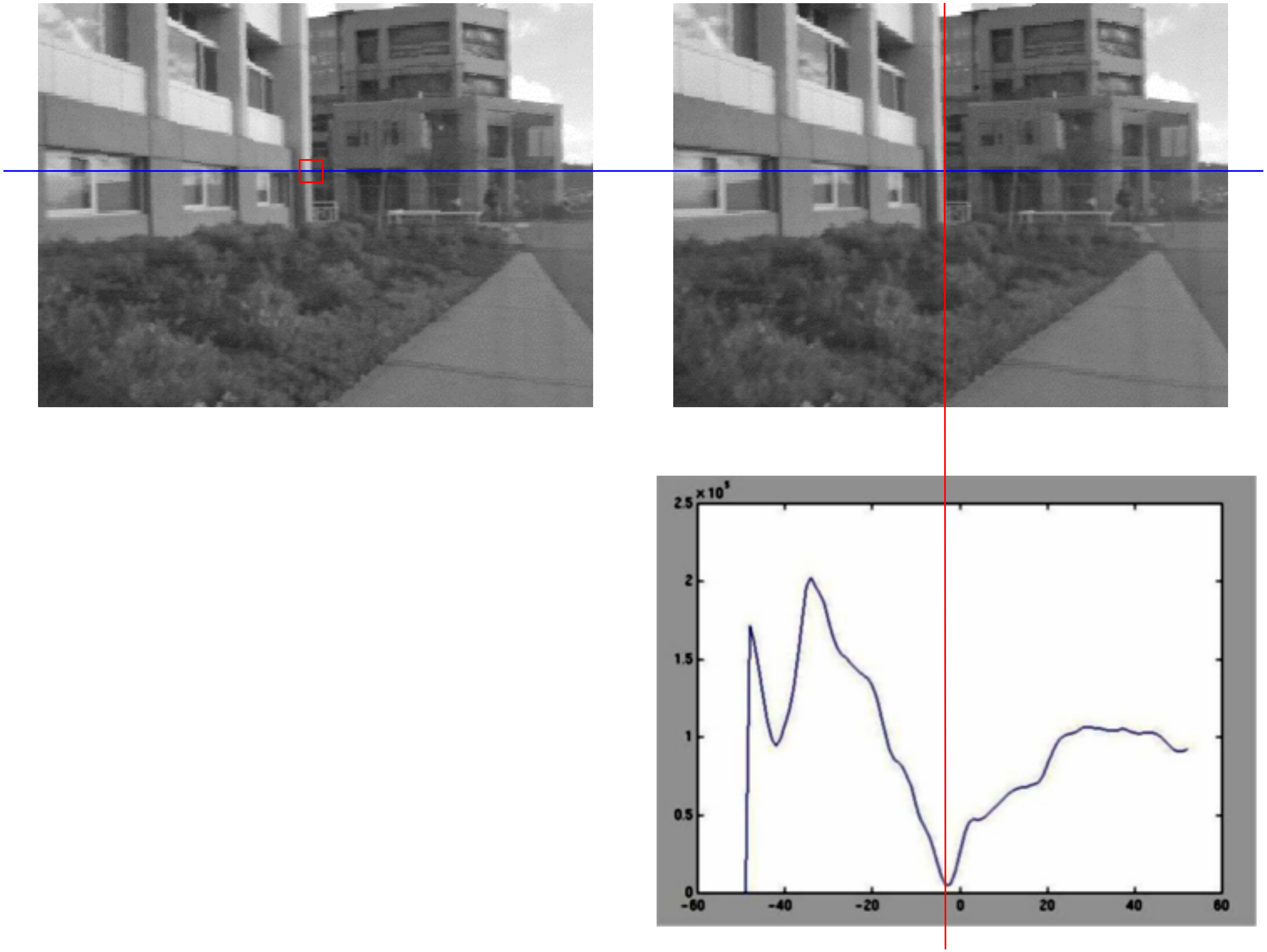

1. Rectify images
   (make epipolar lines horizontal)
2. For each pixel
   a. Find epipolar line
   b. Scan line for best match
   c. Compute depth from disparity

$$Z = \frac{bf}{d}$$

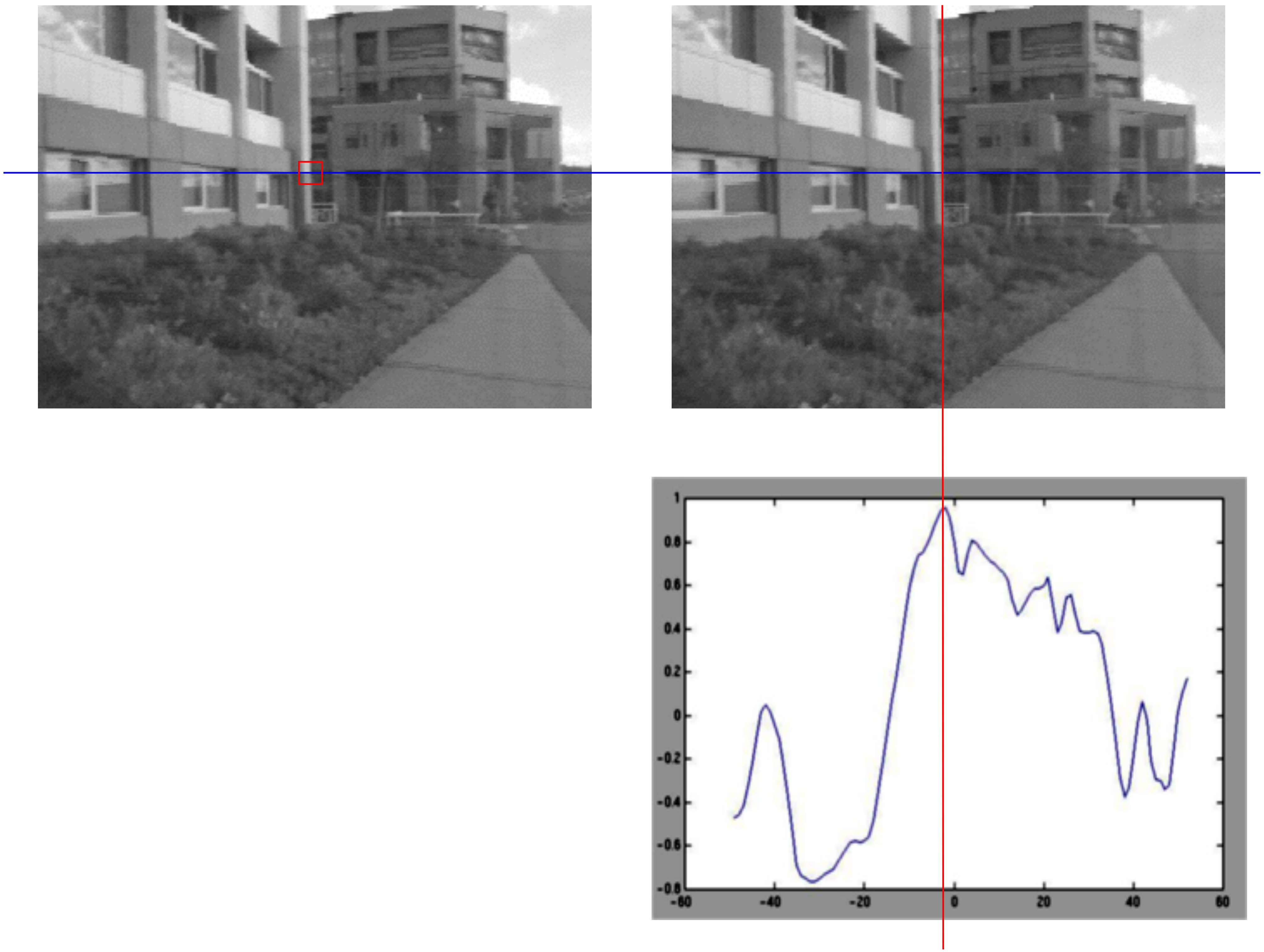
# Stereo Block Matching




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

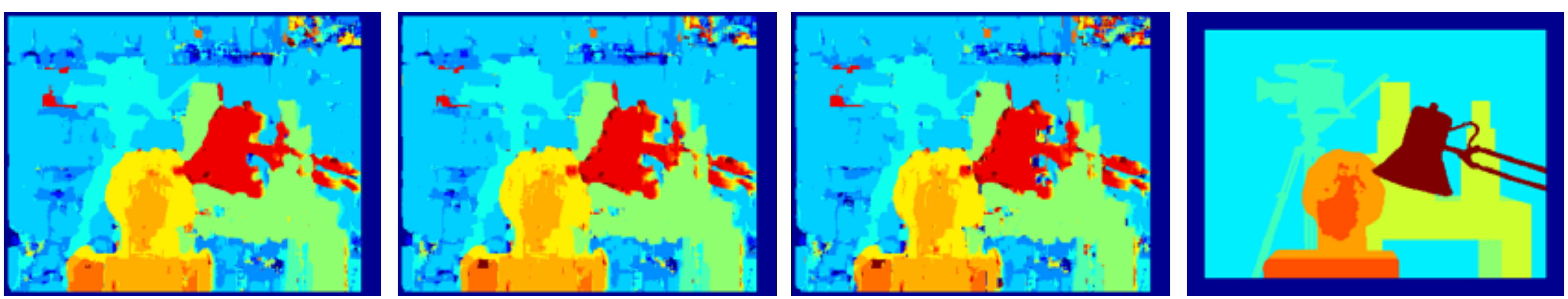



SSD




Normalized cross-correlation

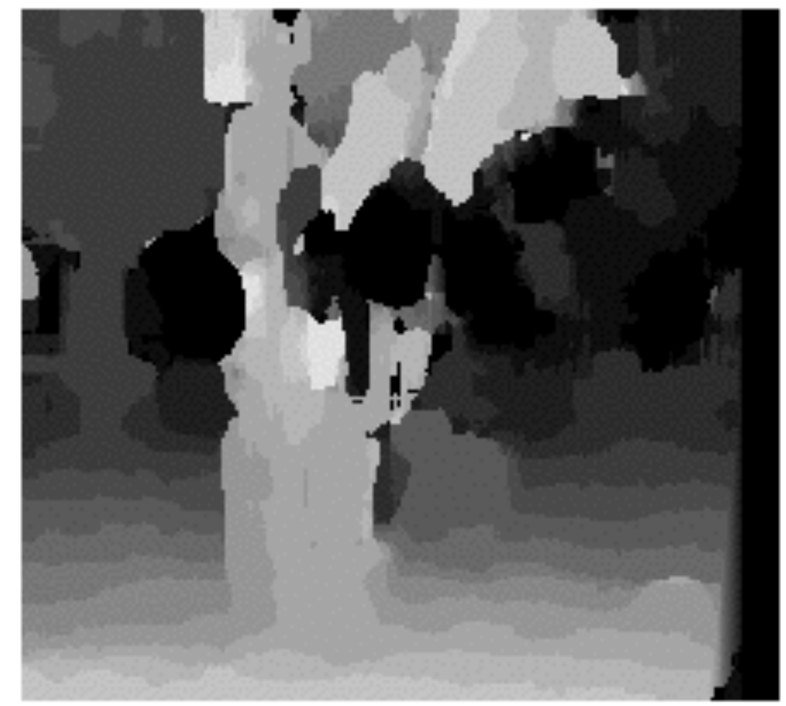
| Similarity Measure | Formula |
|---|---|
| Sum of Absolute Differences (SAD) | $\sum_{(i,j)\in W} \lvert I_1(i,j) - I_2(x+i,y+j) \rvert$ |
| Sum of Squared Differences (SSD) | $\sum_{(i,j)\in W} \left(I_1(i,j) - I_2(x+i,y+j)\right)^2$ |
| Zero-mean SAD | $\sum_{(i,j)\in W} \lvert I_1(i,j) - \bar{I}_1(i,j) - I_2(x+i,y+j) + \bar{I}_2(x+i,y+j) \rvert$ |
| 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

# Effect of window size

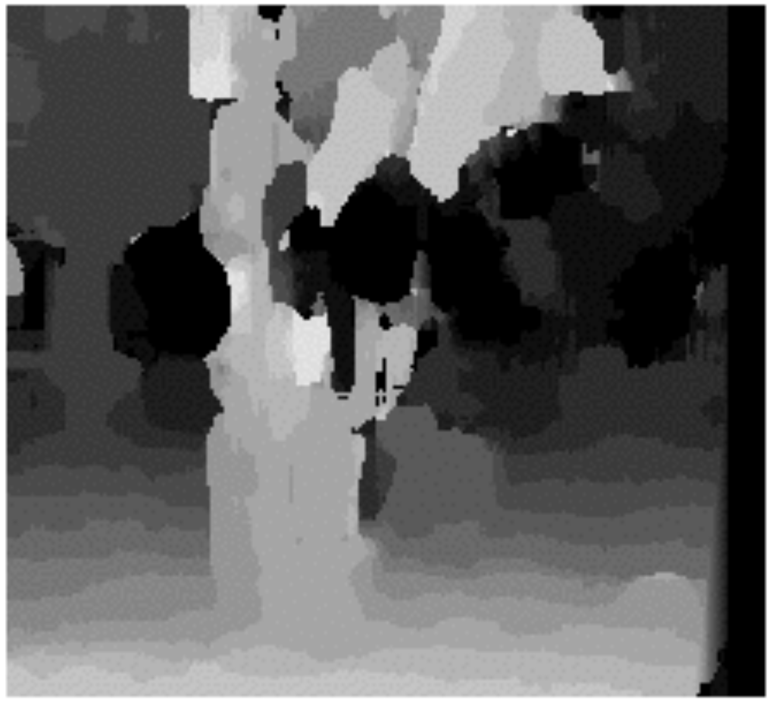
W = 3



W = 20

# Effect of window size

W = 3



W = 20

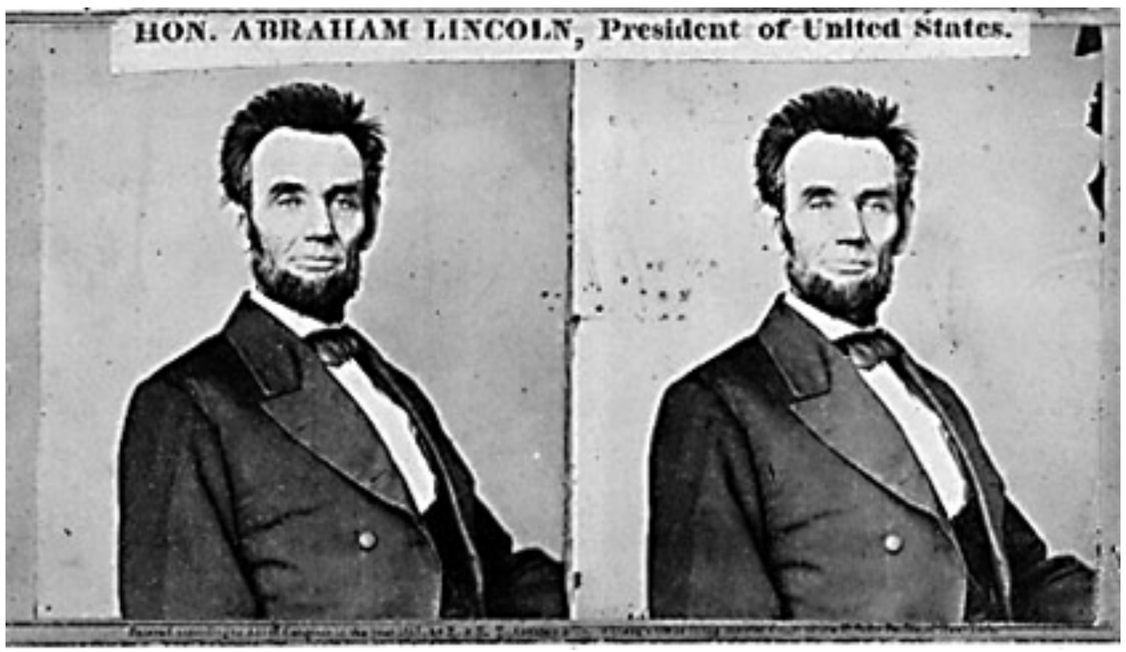
**Smaller window**

- \+ More detail
- \- More noise

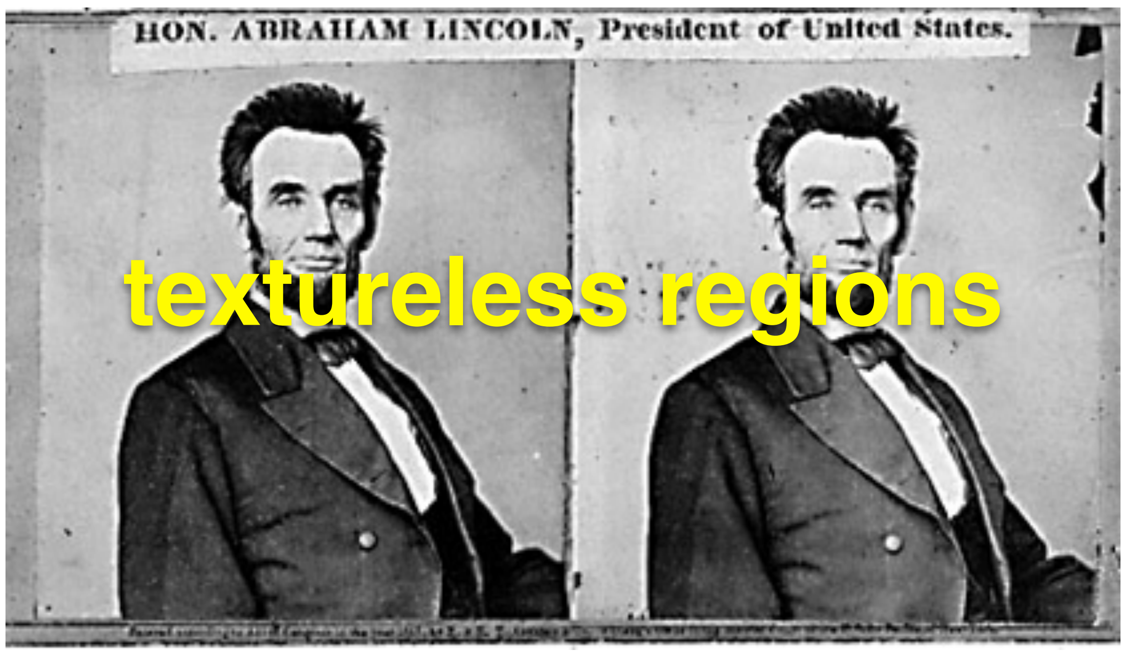
**Larger window**

- \+ Smoother disparity maps
- \- Less detail
- \- Fails near boundaries

# *When will stereo block matching fail?*



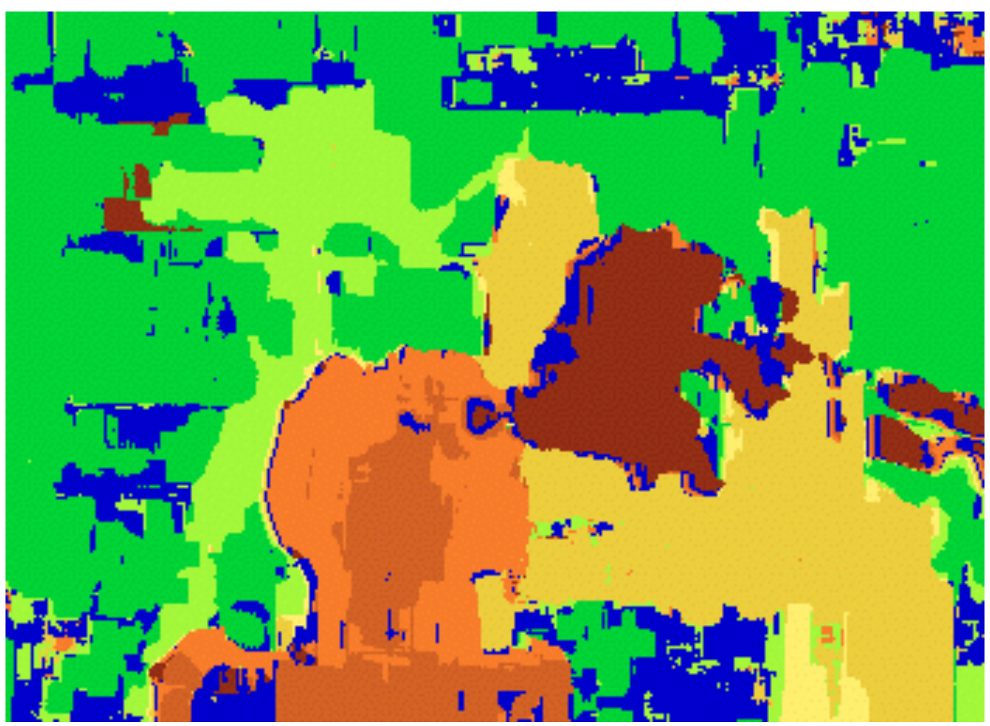

# *When will stereo block matching fail?*





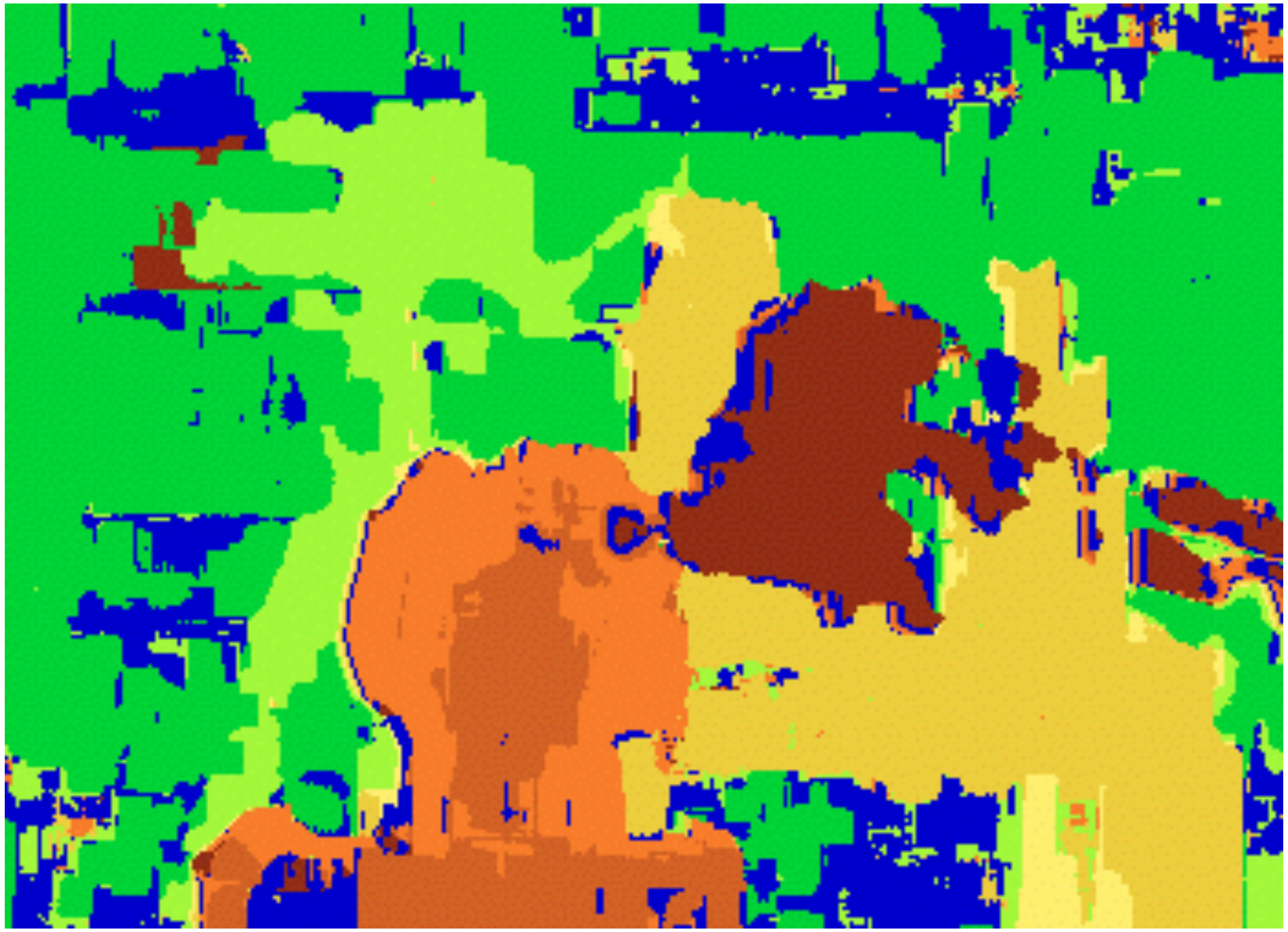
# Improving Stereo Block Matching

Block matching

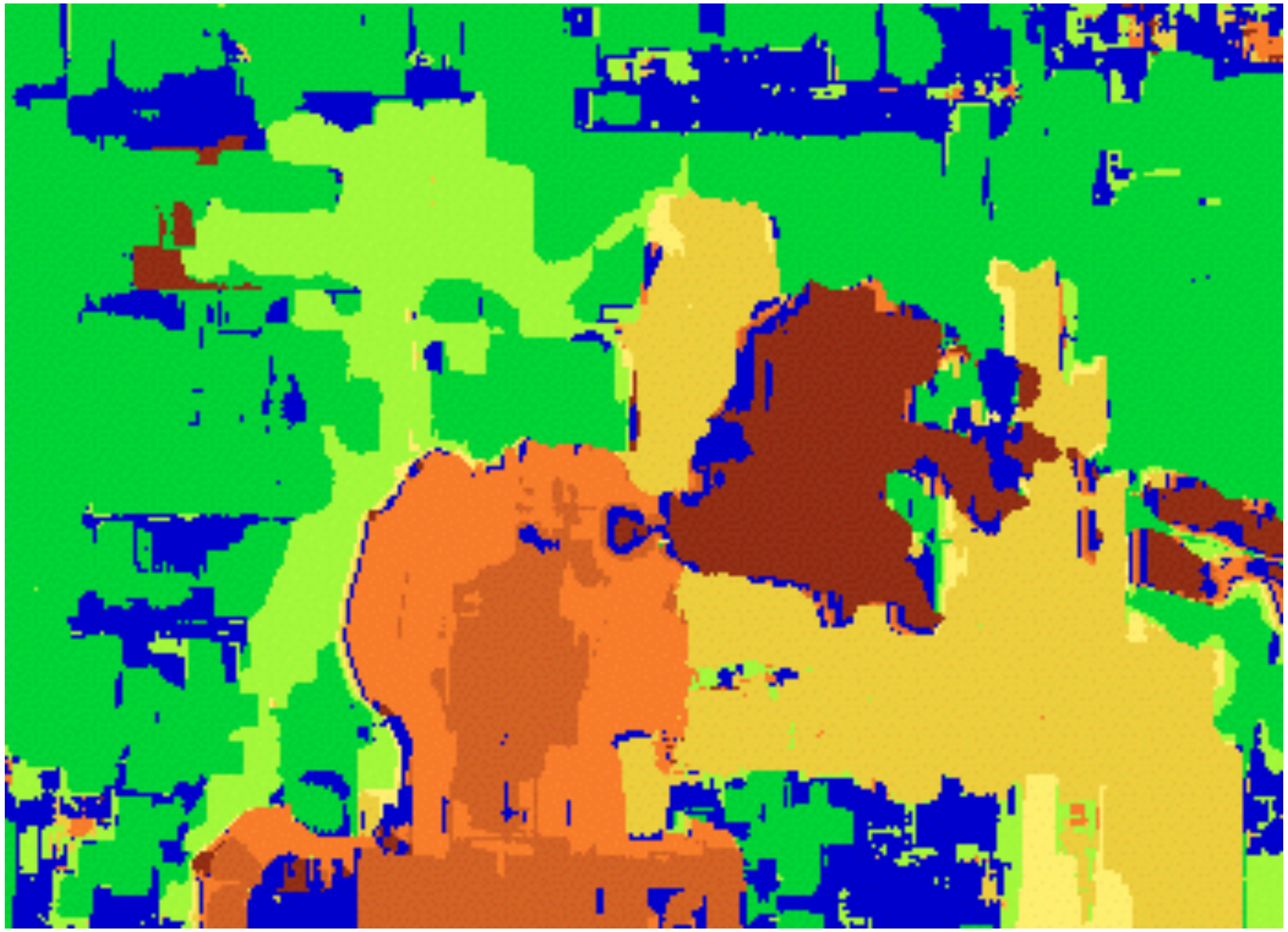


Ground truth

*What are some problems with the result?*



*How can we improve depth estimation?*



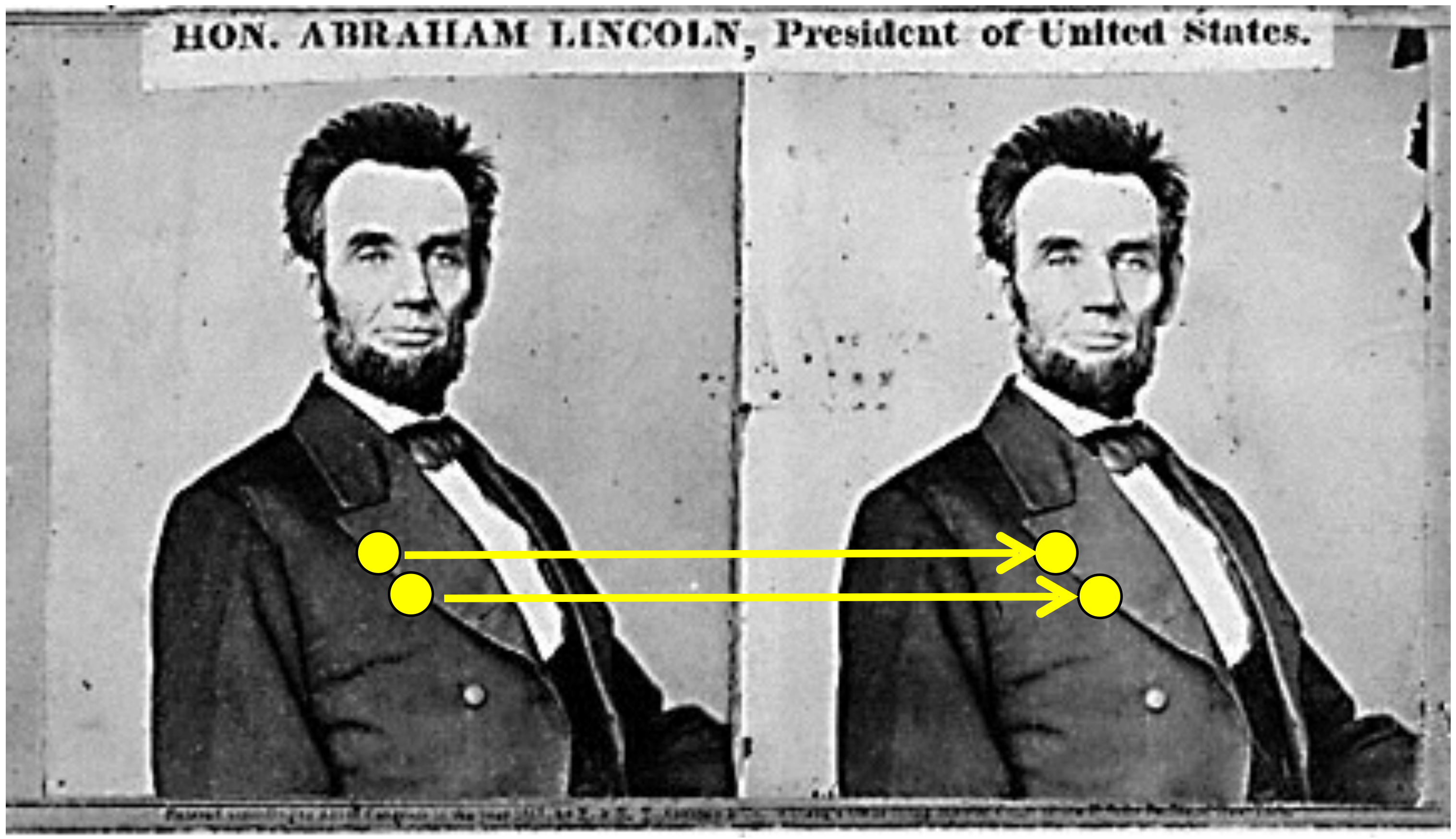
*How can we improve depth estimation?*

Too many discontinuities.
We expect disparity values to change slowly.

Let's make an assumption:
**depth should change smoothly**

Stereo matching 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

energy function
(for one pixel)

$$E(d) = \underbrace{E_d(d)}_{\text{data term}} + \underbrace{\lambda E_s(d)}_{\text{smoothness term}}$$

energy function
(for one pixel)

$$E(d) = \underbrace{E_d(d)}_{\text{data term}} + \lambda \underbrace{E_s(d)}_{\text{smoothness term}}$$

Want each pixel to find a good match in the other image

(block matching result)

Adjacent pixels should (usually) move about the same amount

(smoothness function)

$$E(d) = E_d(d) + \lambda E_s(d)$$

$$E_d(d) = \sum_{(x,y)\in I} C(x, y, d(x, y))$$

data term

SSD distance between windows centered at I(x, y) and J(x+ d(x,y), y)

$$E(d) = E_d(d) + \lambda E_s(d)$$

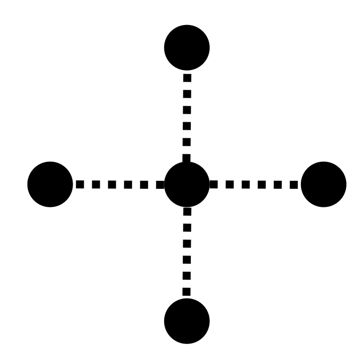
$$E_d(d) = \sum_{(x,y)\in I} C(x, y, d(x, y))$$

SSD distance between windows
centered at I(x, y) and J(x+ d(x,y), y)

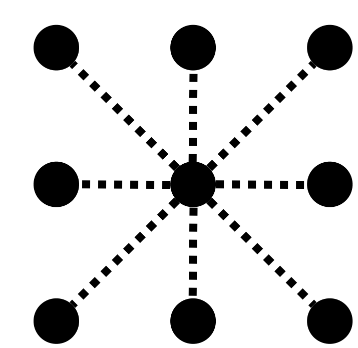
$$E_s(d) = \sum_{(p,q)\in \mathcal{E}} V(d_p, d_q)$$

smoothness term

$\mathcal{E}$ : set of neighboring pixels



4-connected
neighborhood



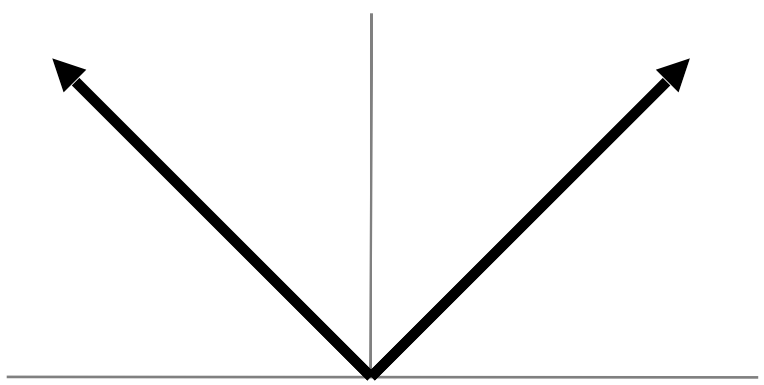
8-connected
neighborhood

$$E_s(d) = \sum_{(p,q)\in\mathcal{E}} V(d_p, d_q)$$

smoothness term

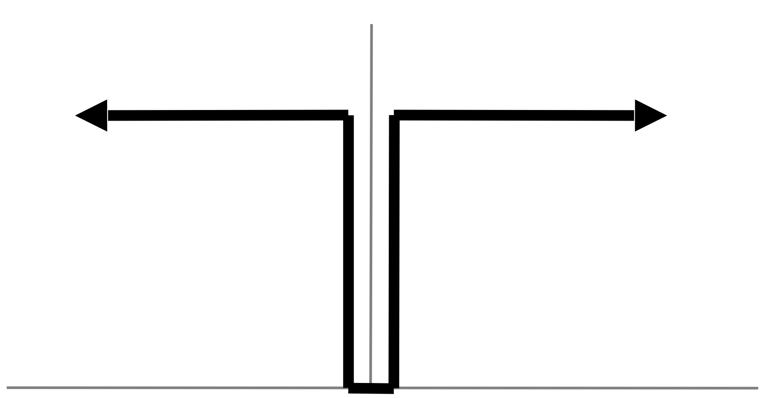
$$V(d_p, d_q) = |d_p - d_q|$$

$L_1$ distance



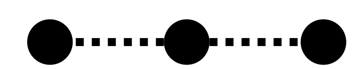
$$V(d_p, d_q) = \begin{cases} 0 & \text{if } d_p = d_q \\ 1 & \text{if } d_p \neq d_q \end{cases}$$

“Potts model”



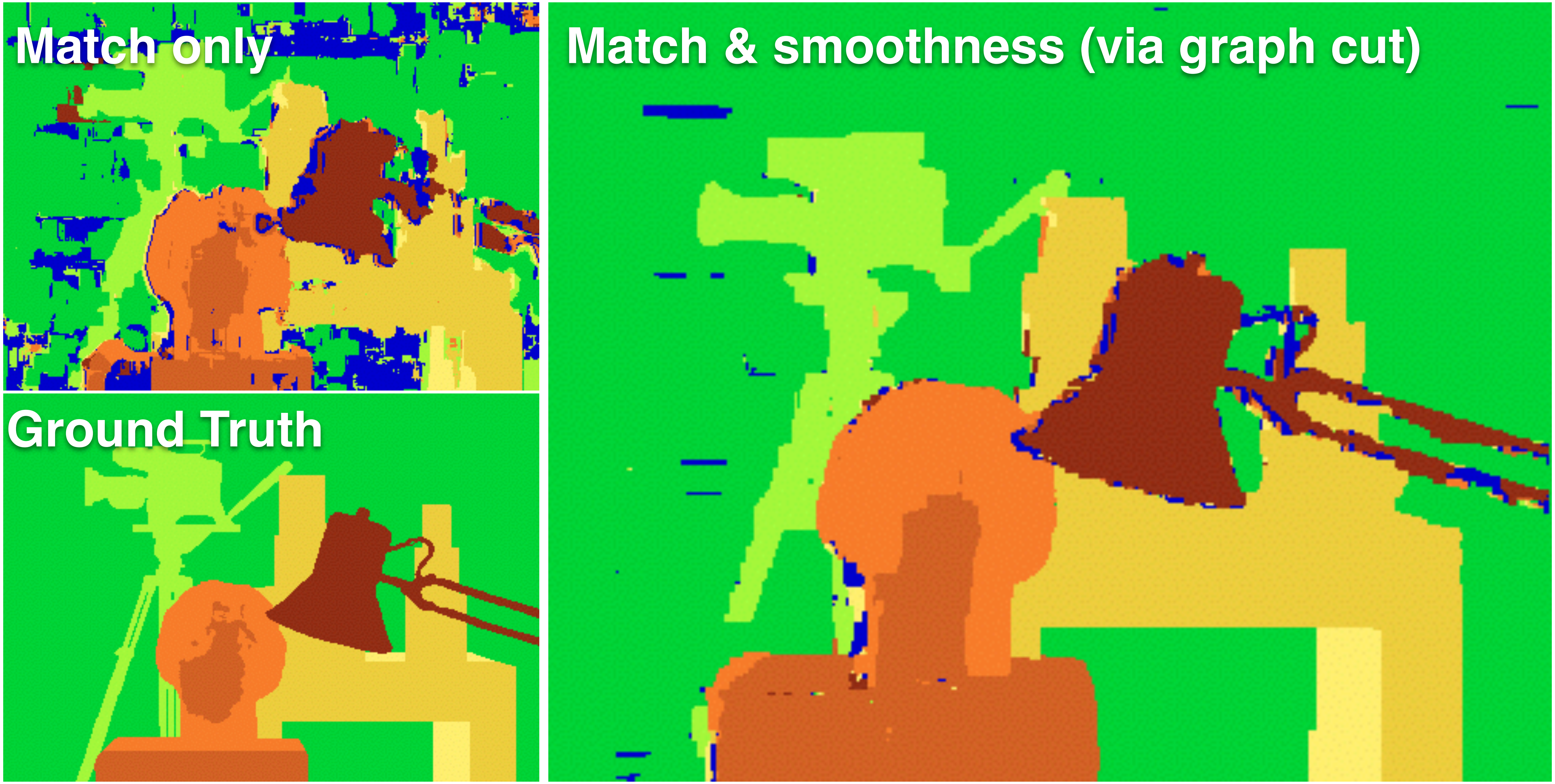
# Dynamic Programming

$$E(d) = E_d(d) + \lambda E_s(d)$$

Can minimize this independently per scanline
using dynamic programming (DP) 

$D(x, y, d)$ : minimum cost of solution such that d(x,y) = d

$$D(x, y, d) = C(x, y, d) + \min_{d'} \{D(x-1, y, d') + \lambda \left|d - d'\right|\}$$




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