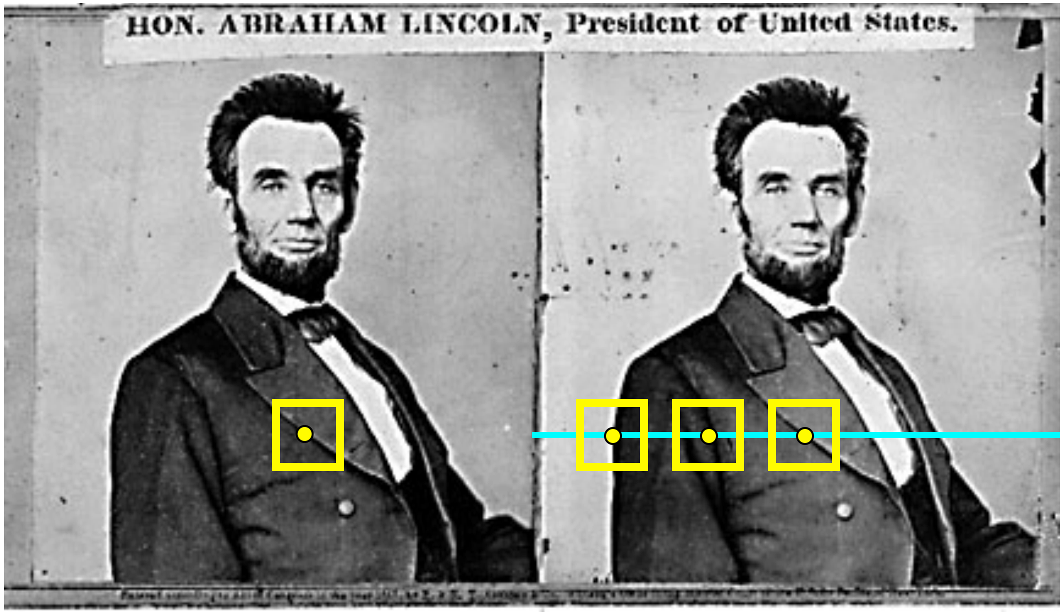
Solving for Stereo Correspondence

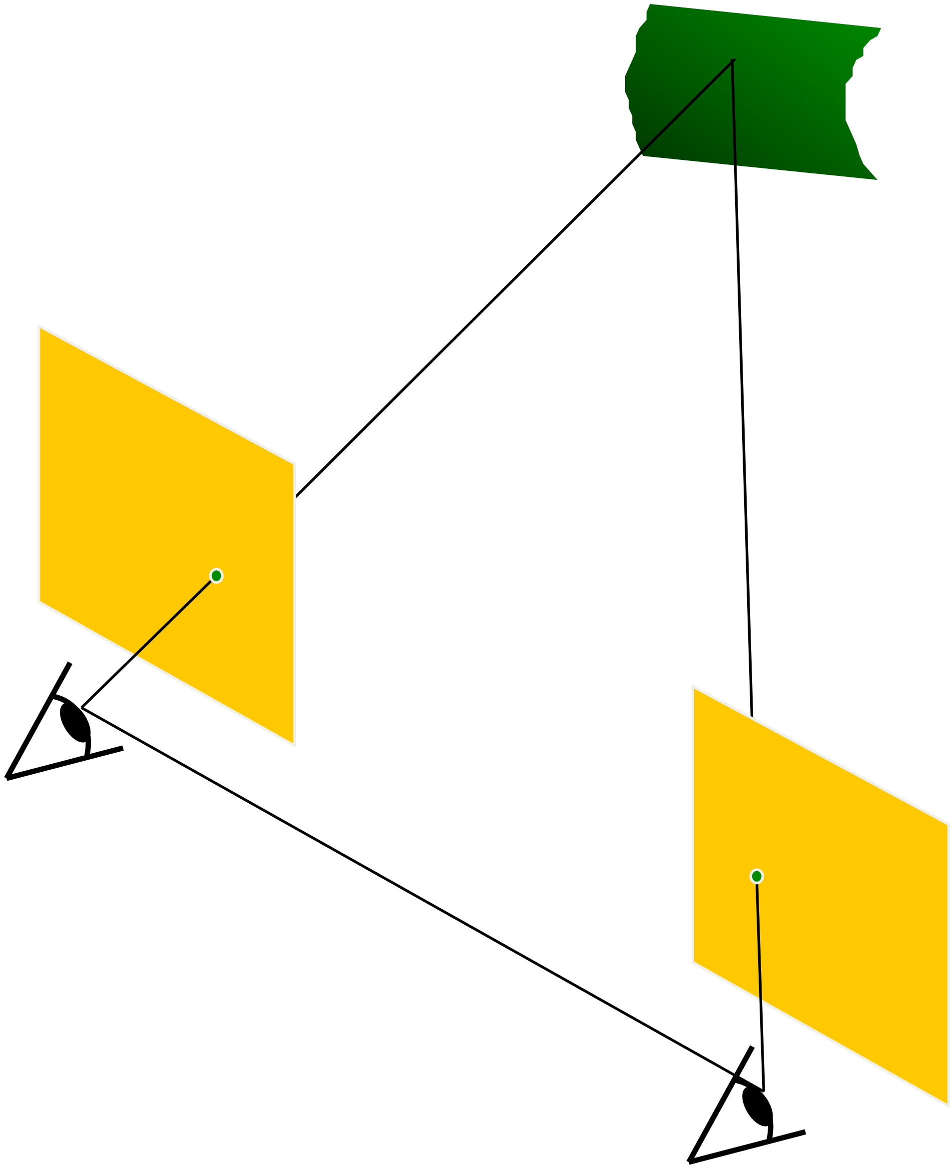
Many slides drawn from Lana Lazebnik, UIUC

Basic stereo matching algorithm



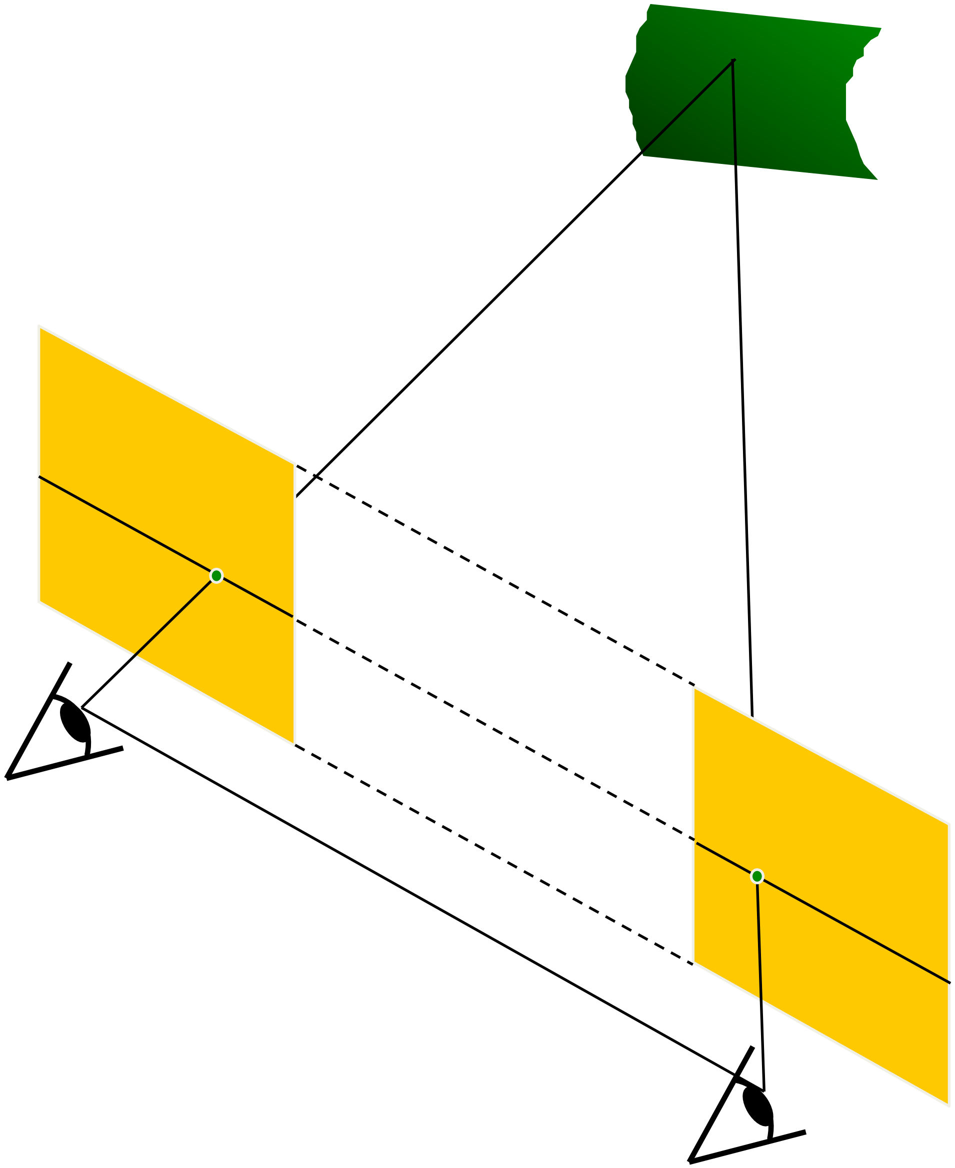
- For each pixel in the first image
  - Find corresponding epipolar line in the right image
  - Examine all pixels on the epipolar line and pick the best match
  - Triangulate the matches to get depth information

Simplest Case: Parallel images



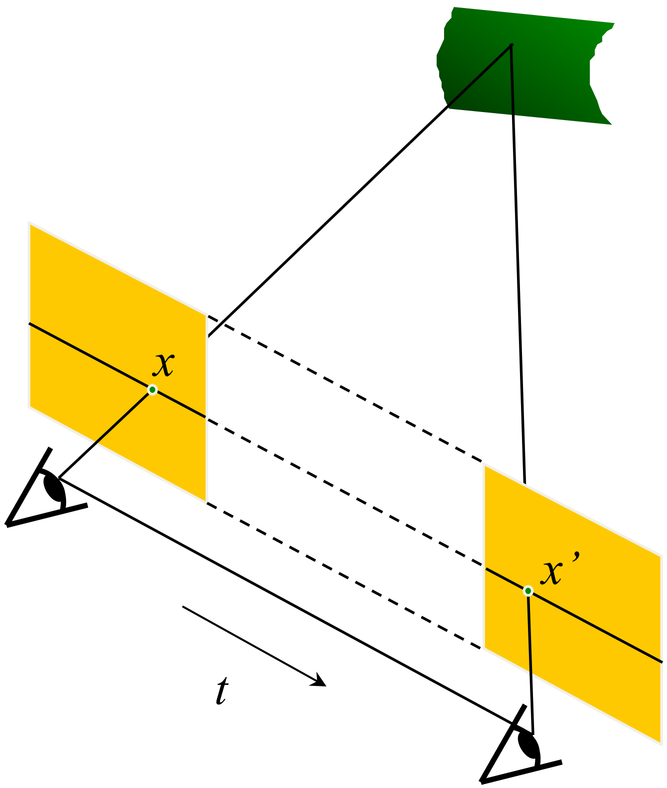
- Image planes of cameras are parallel to each other and to the baseline
- Camera centers are at same height
- Focal lengths are the same

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 are the same
- Then epipolar lines fall along the horizontal scan lines of the images

Essential matrix for parallel images



Epipolar constraint:

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

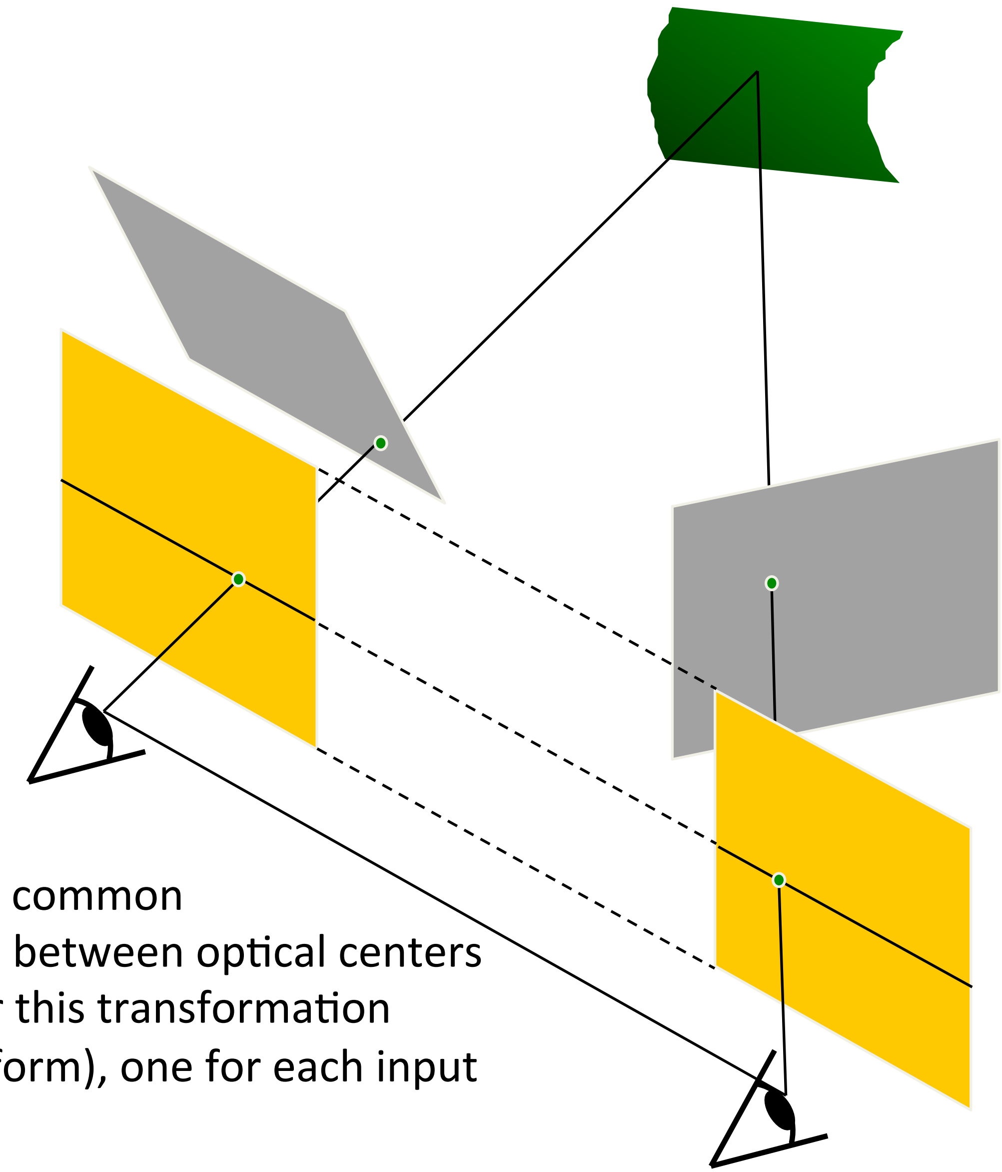
$$\mathbf{R} = \mathbf{I} \qquad \mathbf{t} = (T, 0, 0)$$

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

$$\begin{pmatrix} u' & v' & 1 \end{pmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = 0 \qquad \begin{pmatrix} u' & v' & 1 \end{pmatrix} \begin{pmatrix} 0 \\ -T \\ Tv \end{pmatrix} = 0 \qquad Tv' = Tv$$

The y-coordinates of corresponding points are the same!

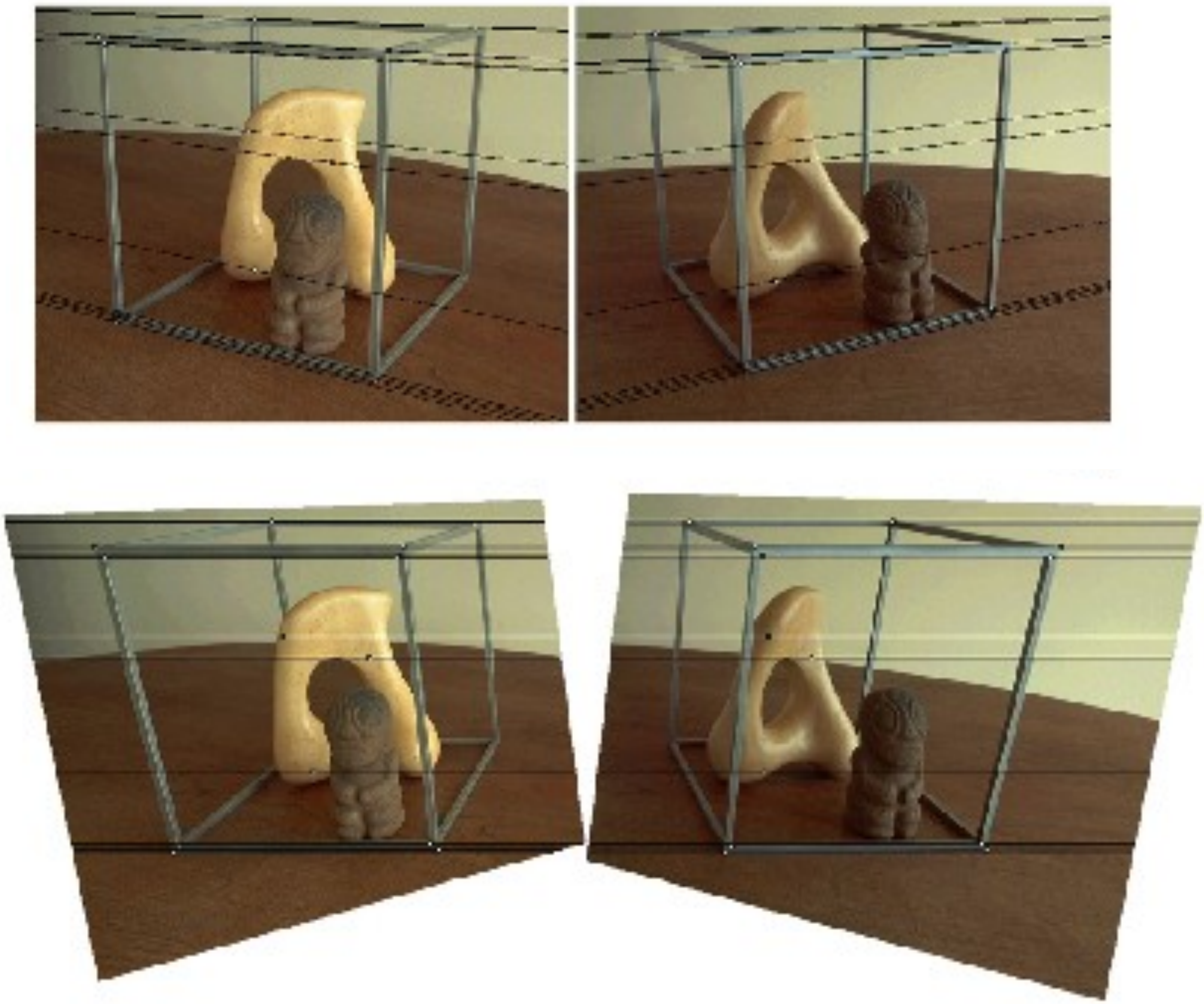
Stereo image rectification



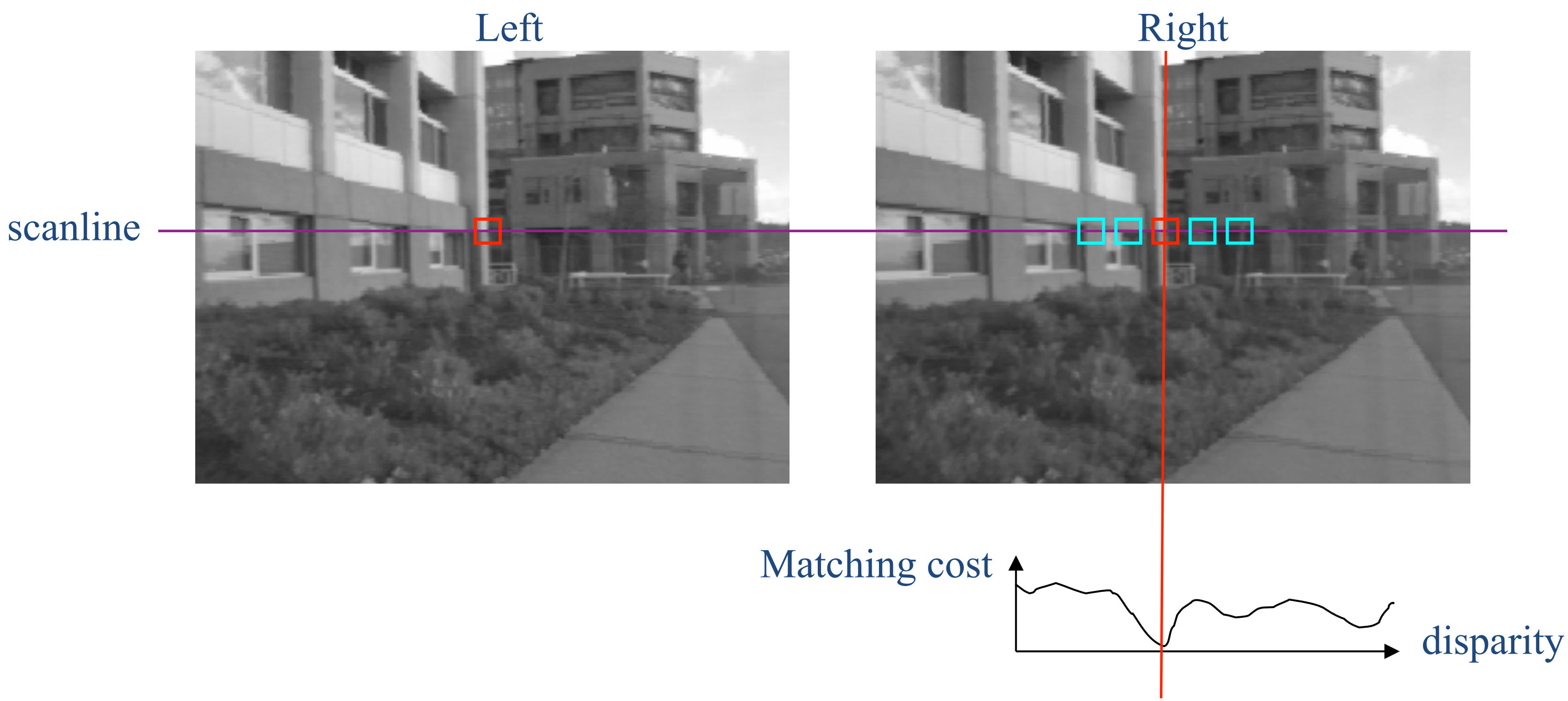
- Reproject image planes onto a common
- plane parallel to the line between optical centers
- Pixel motion is horizontal after this transformation
- Two homographies (3x3 transform), one for each input image reprojection

•C. Loop and Z. Zhang. Computing Rectifying Homographies for Stereo Vision. IEEE Conf. Computer Vision and Pattern Recognition, 1999.

Rectification example



Correspondence search



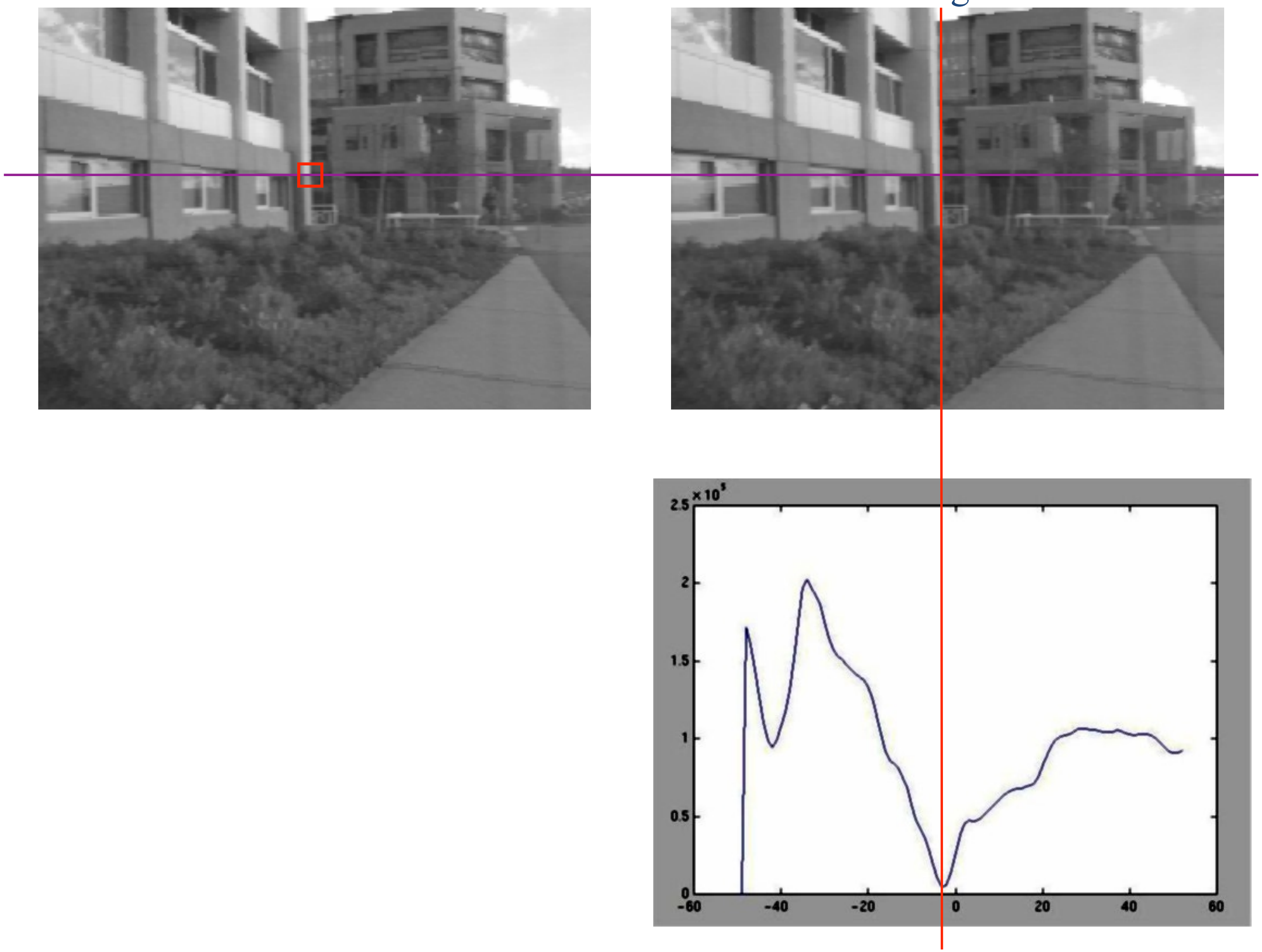
- Slide a window along the right scanline and compare contents of that window with the reference window in the left image
- Matching cost: SSD or normalized correlation

Correspondence search

Left

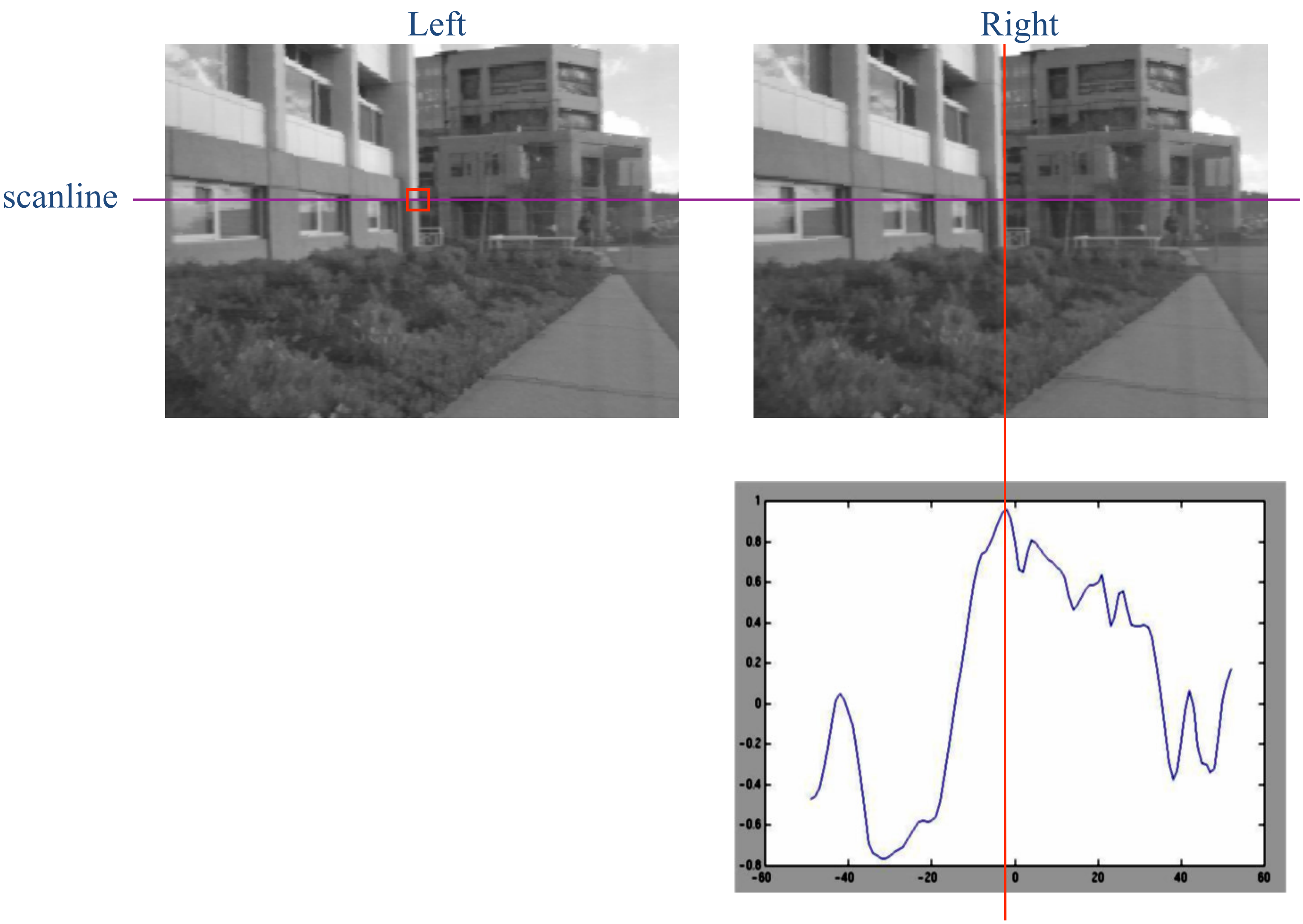
Right

scanline



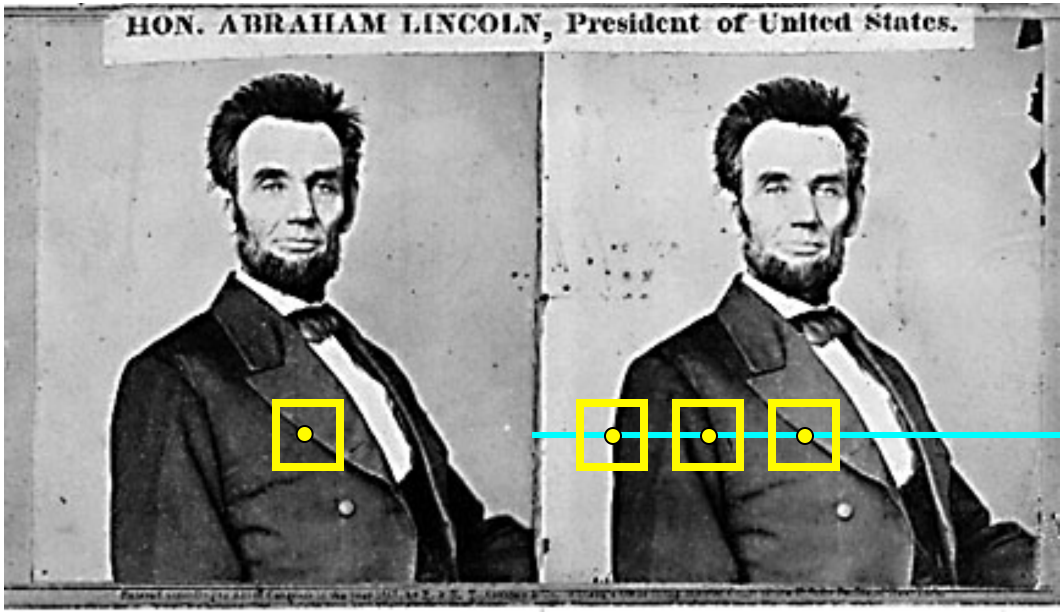
SSD

Correspondence search



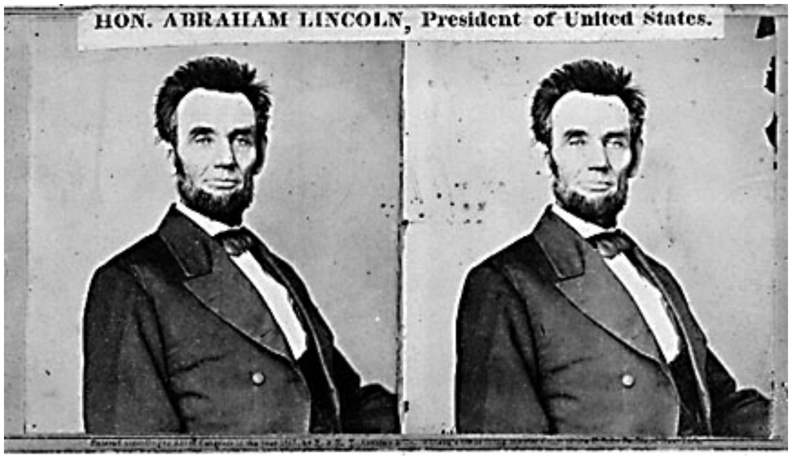
Norm. corr

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 $\text{depth}(x) = B*f/(x-x')$

Failures of correspondence search



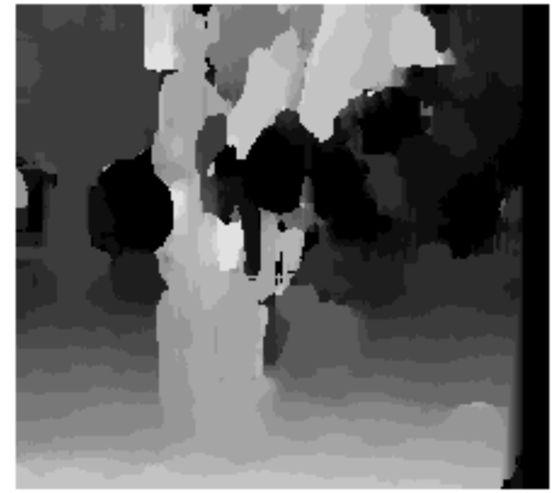
Textureless surfaces

Occlusions, repetition

Non-Lambertian surfaces, specularities

Effect of window size

W = 3



W = 20

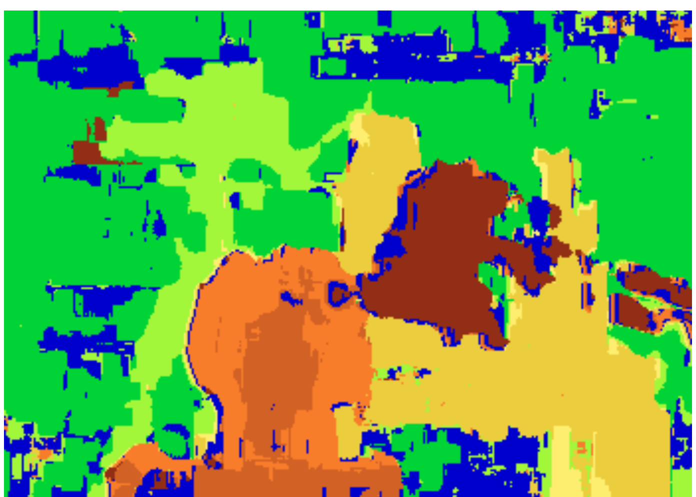
- Smaller window
  - + More detail
  - • More noise

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

Results with window search

Data

Window-based matching



Ground truth

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/

Labeling improved through label swap and expansion macro-moves

BOYKOV ET AL.: FAST APPROXIMATE ENERGY MINIMIZATION VIA GRAPH CUTS

```
1.   Start with an arbitrary labeling f
2.   Set success := 0
3.   For each pair of labels {α, β} ⊂ 𝓛
   3.1.   Find f̂ = arg min E(f′) among f′ within one α-β swap of f
   3.2.   If E(f̂) < E(f), set f := f̂ and success := 1
4.   If success = 1 goto 2
5.   Return f
```

```
1.   Start with an arbitrary labeling f
2.   Set success := 0
3.   For each label α ∈ 𝓛
   3.1.   Find f̂ = arg min E(f′) among f′ within one α-expansion of f
   3.2.   If E(f̂) < E(f), set f := f̂ and success := 1
4.   If success = 1 goto 2
5.   Return f
```

Fig. 3. Our swap algorithm (top) and expansion algorithm (bottom).

The two basic macro-moves

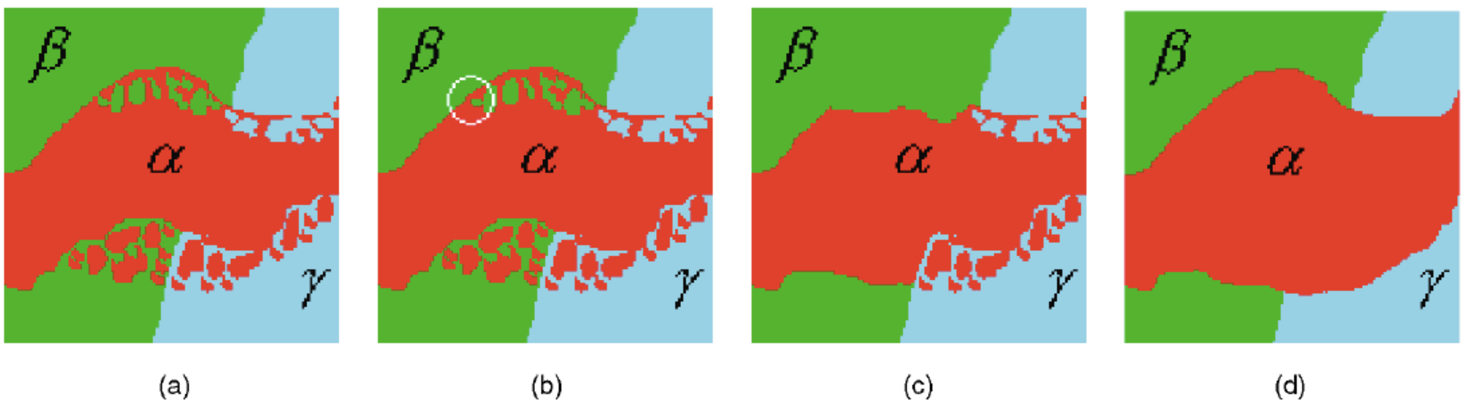


Fig. 2. Examples of standard and large moves from a given initial labeling (a). The number of labels is $|\mathcal{L}| = 3$. A standard move, (a) → (b), changes the label of a single pixel (in the circled area). Strong moves, α-β-swap (a) → (c) and α-expansion (a) → (d), allow large number of pixels to change their labels simultaneously.

Each move is done by solving a graph cut problem

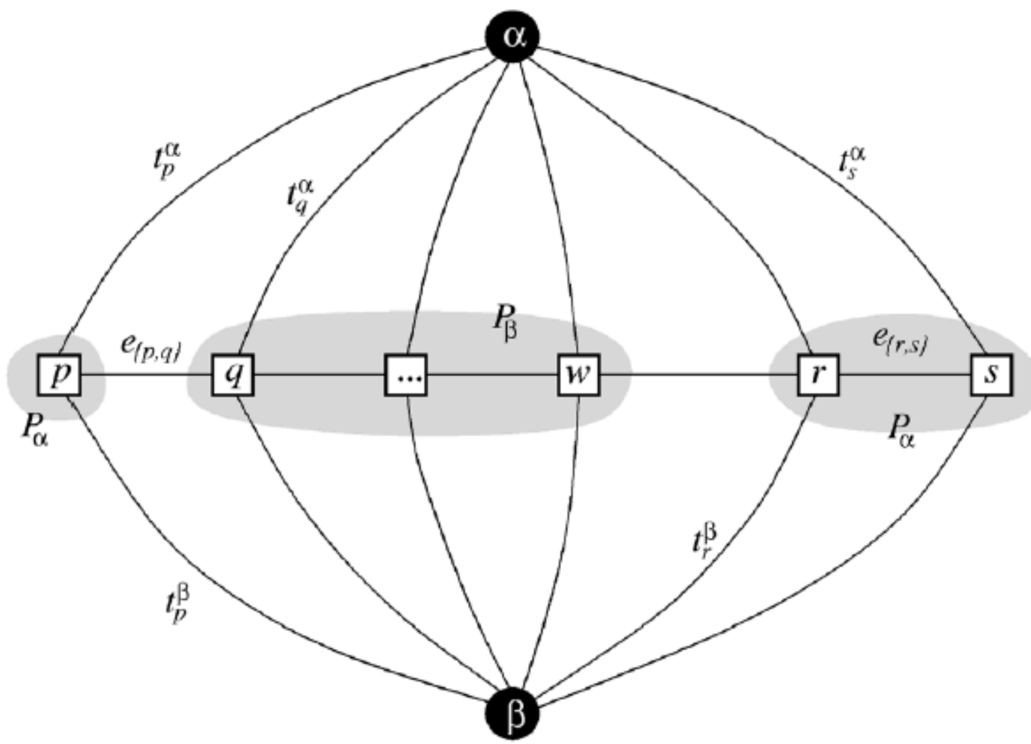


Fig. 4. An example of the graph $\mathcal{G}_{\alpha\beta}$ for a 1D image. The set of pixels in the image is $\mathcal{P}_{\alpha\beta} = \mathcal{P}_\alpha \cup \mathcal{P}_\beta$, where $\mathcal{P}_\alpha = \{p, r, s\}$ and $\mathcal{P}_\beta = \{q, \ldots, w\}$.