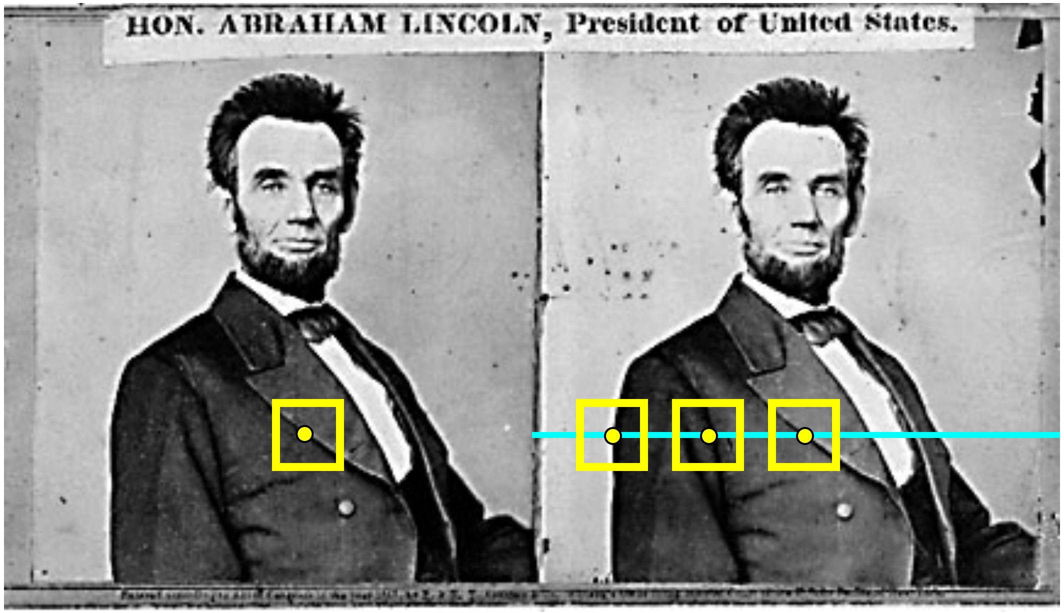
Solving for Stereo Correspondence

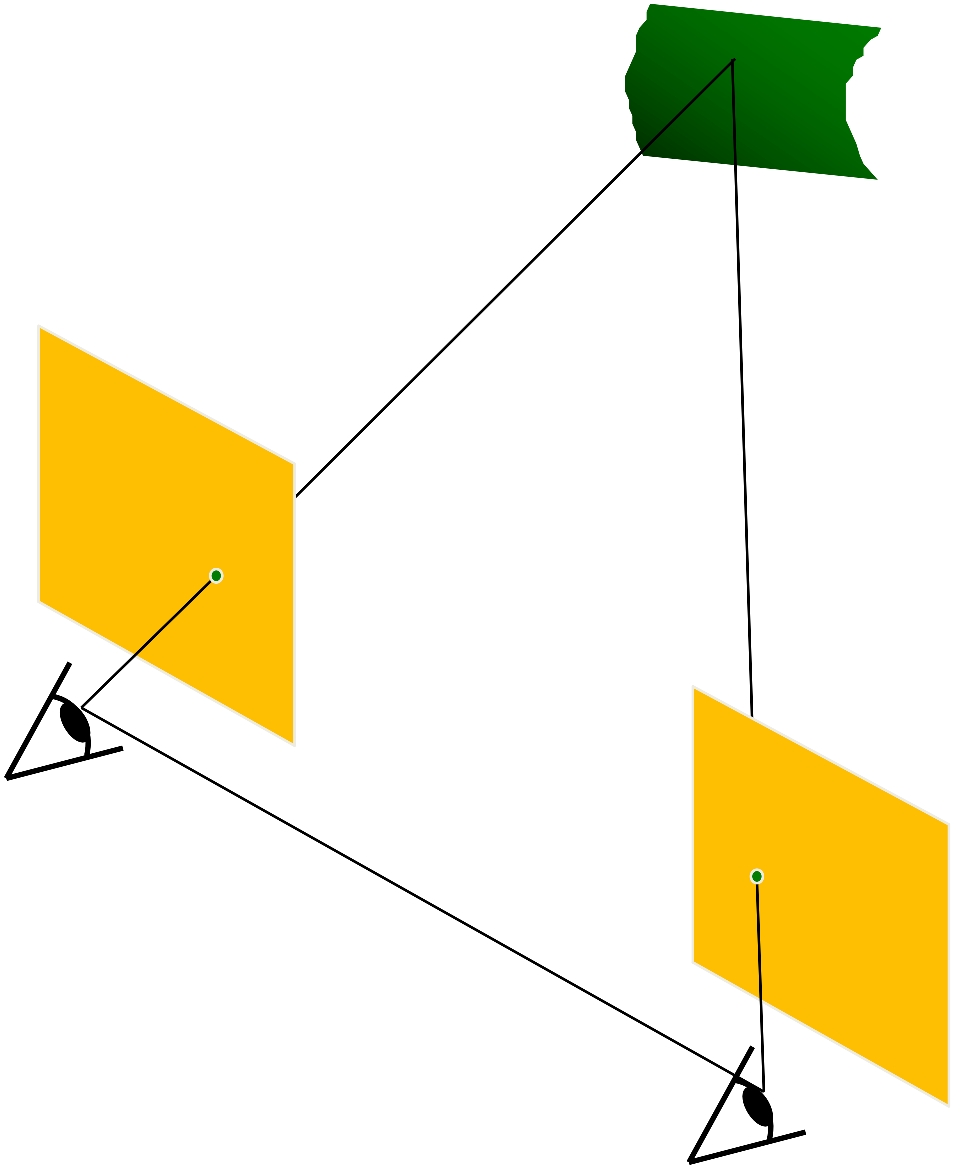
Many slides drawn from Lana Lazebnik, UIUC

Basic stereo matching algorithm



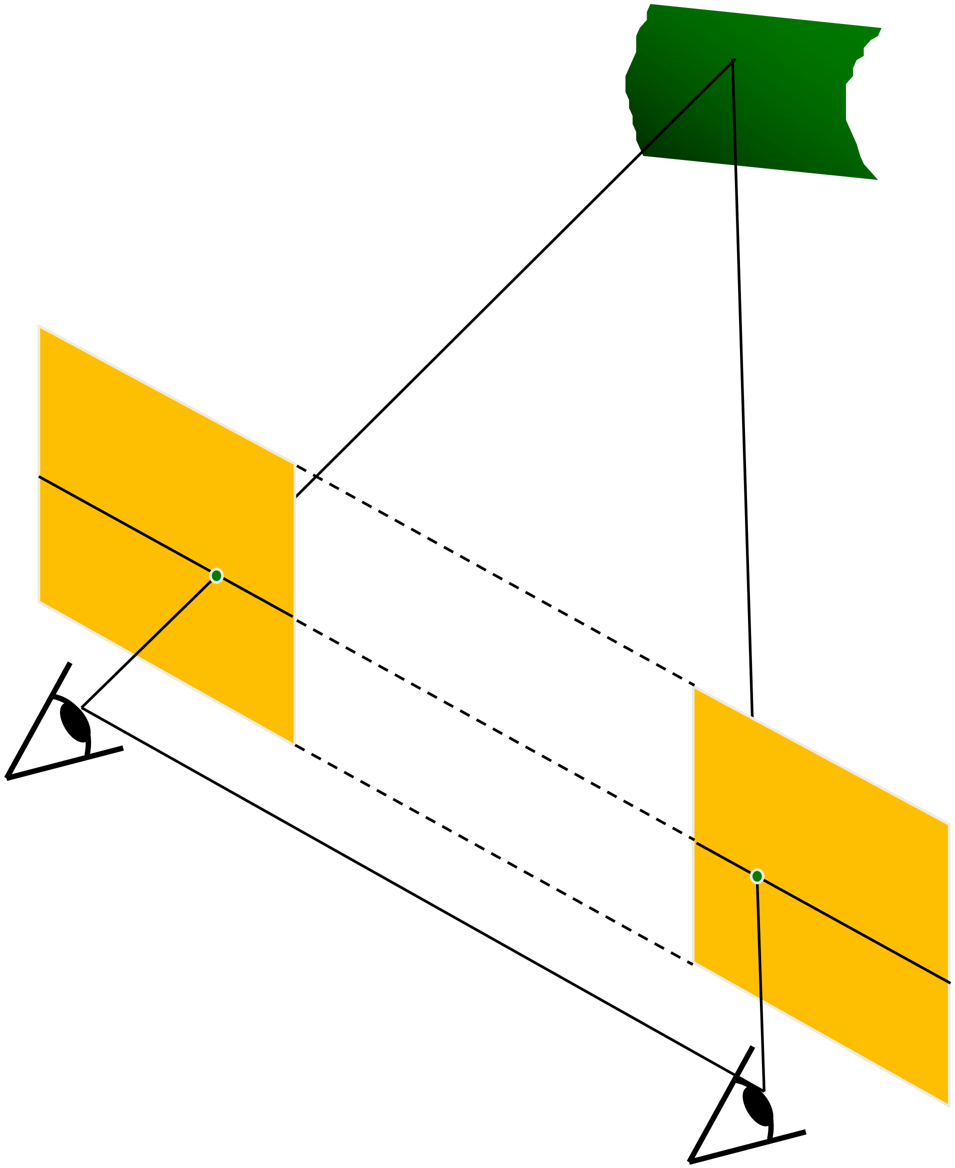
- For each pixel in the first image
  - Find corresponding epipolar line in the right image
  - Examine all pixels on the epipolar line and pick the best match
  - Triangulate the matches to get depth information

Simplest Case: Parallel images



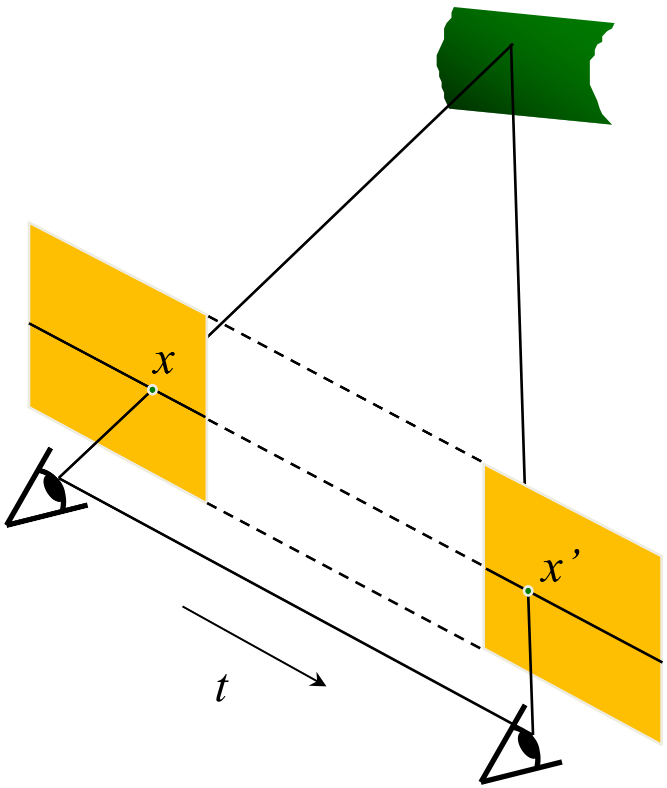
- Image planes of cameras are parallel to each other and to the baseline
- Camera centers are at same height
- Focal lengths are the same

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:

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

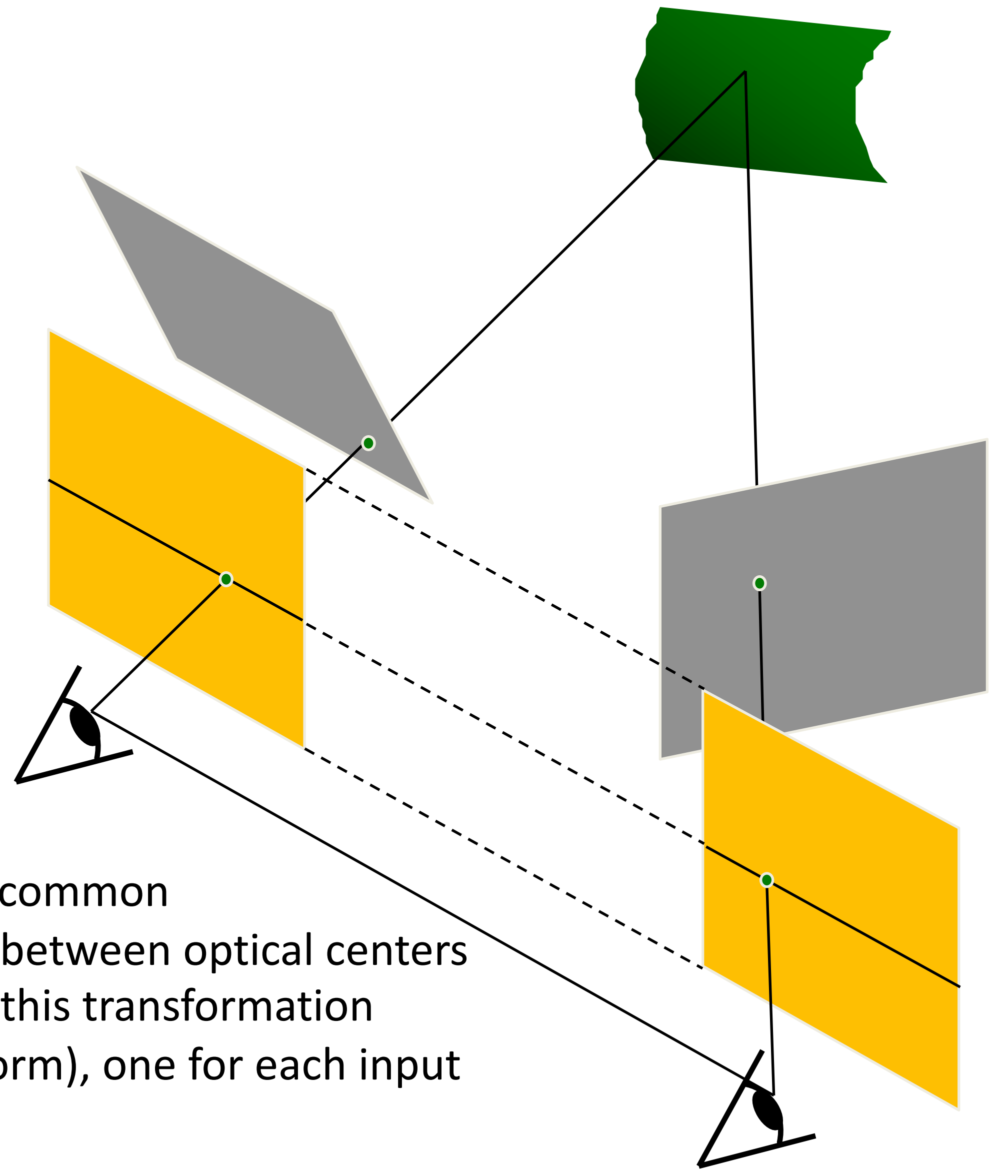
$$\boldsymbol{R} = \boldsymbol{I} \qquad \boldsymbol{t} = (T, 0, 0)$$

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

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

The y-coordinates of corresponding points are the same!

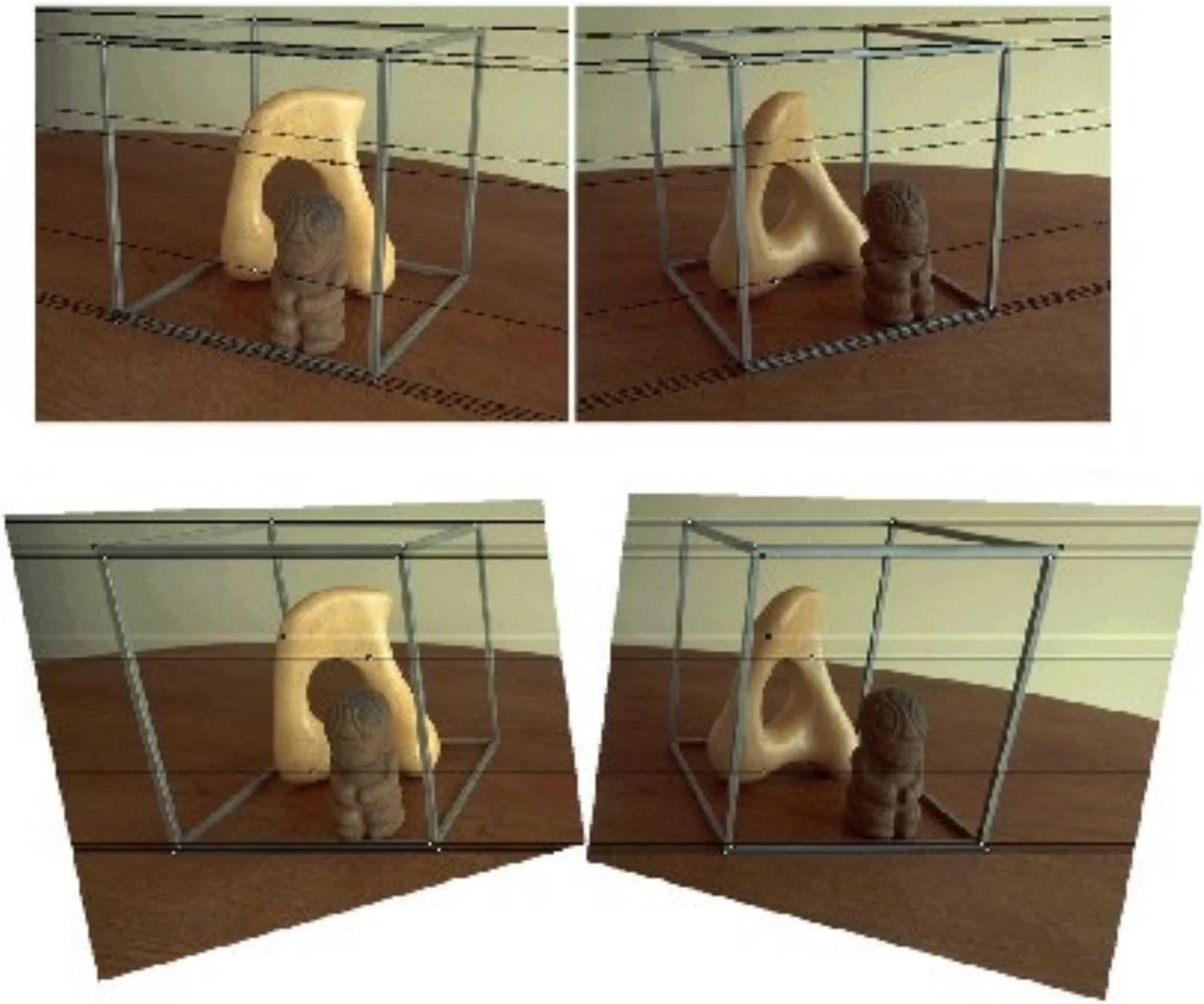
Stereo image rectification



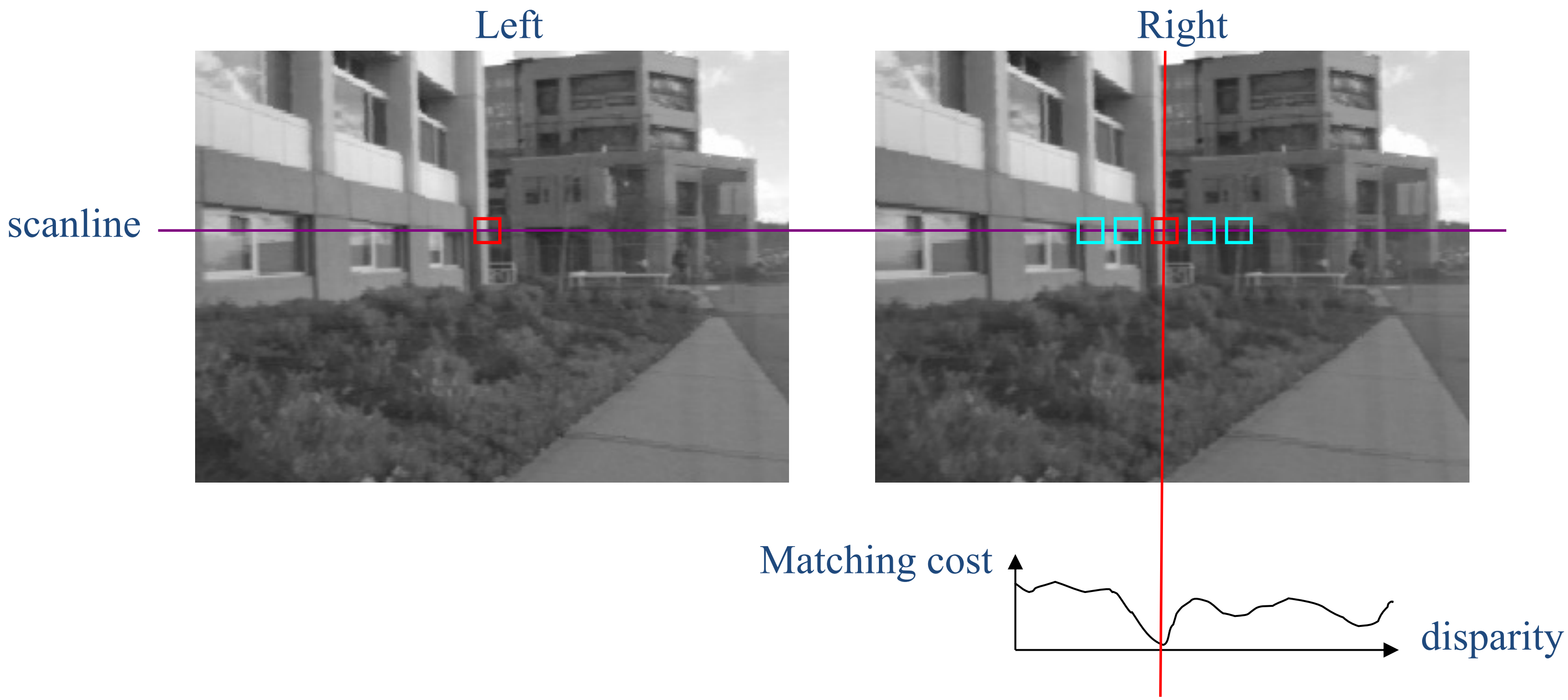
- Reproject image planes onto a common
- plane parallel to the line between optical centers
- Pixel motion is horizontal after this transformation
- Two homographies (3x3 transform), one for each input image reprojection

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

Rectification example



Correspondence search



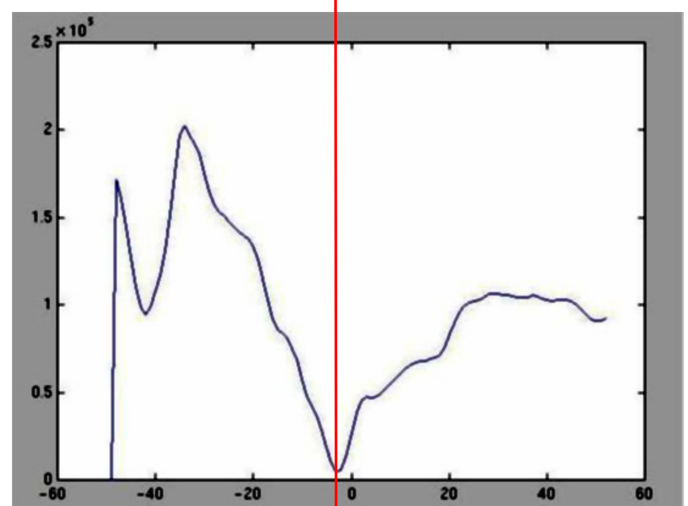
- Slide a window along the right scanline and compare contents of that window with the reference window in the left image
- Matching cost: SSD or normalized correlation

Correspondence search

Left

Right

scanline



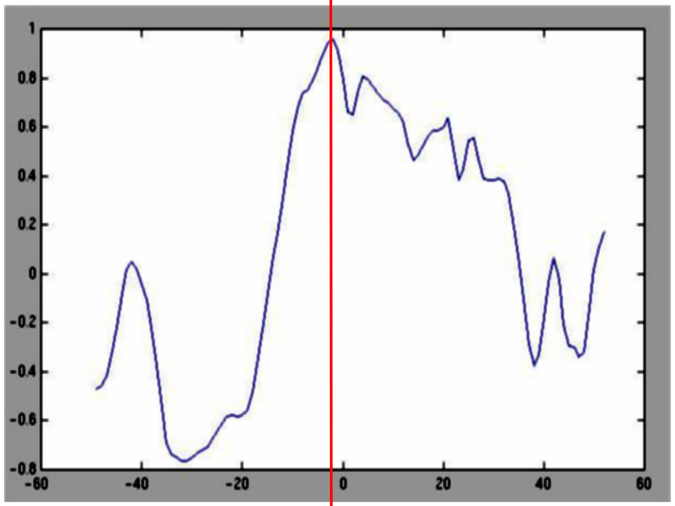
SSD

Correspondence search

Left

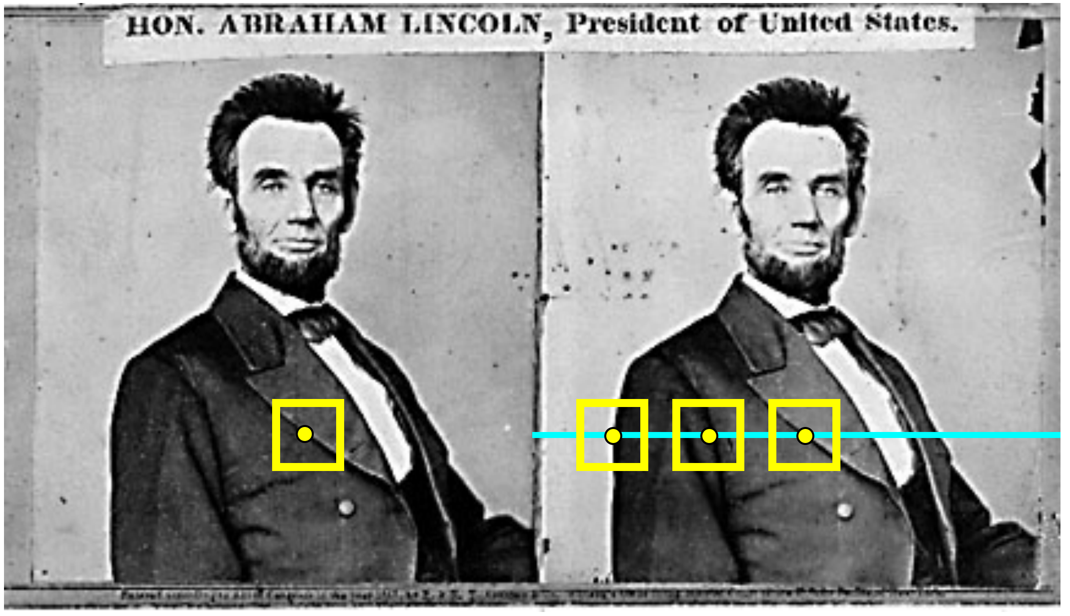
Right

scanline



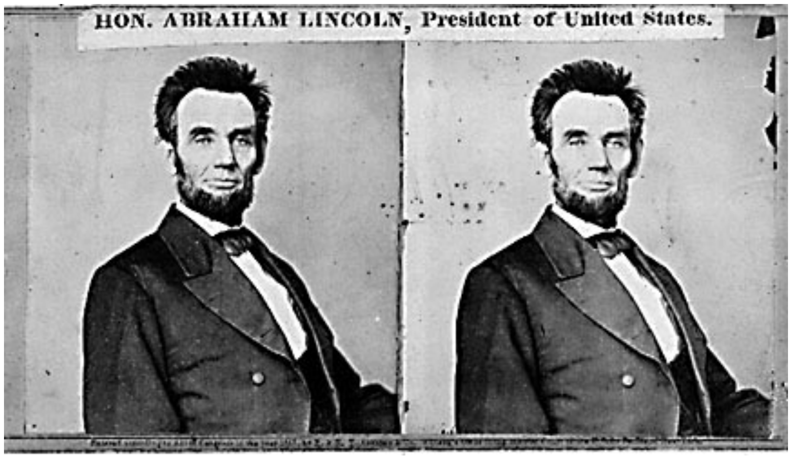
Norm. corr

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



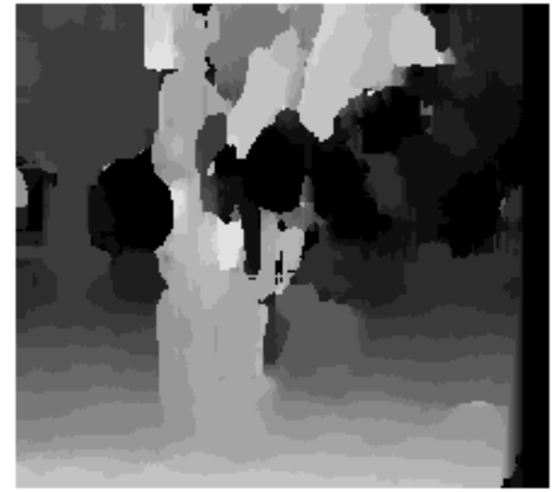
Textureless surfaces

Occlusions, repetition

Non-Lambertian surfaces, specularities

Effect of window size

W = 3



W = 20

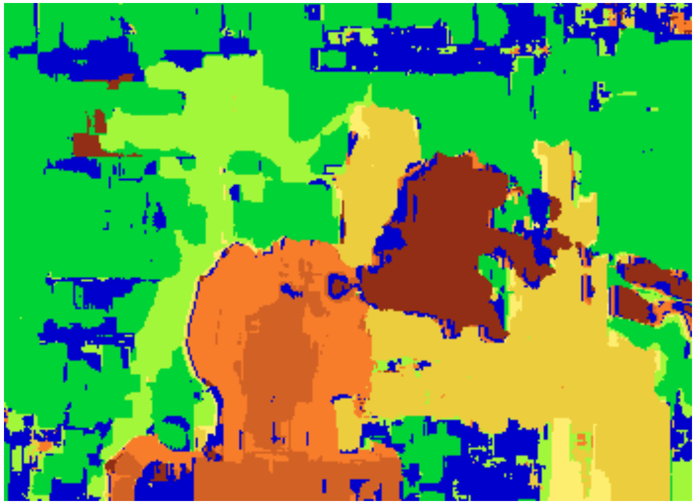
- Smaller window
  + More detail
  - More noise

- 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/

SuperGlue: Learning Feature Matching with Graph Neural Networks

Paul-Edouard Sarlin[1*] Daniel DeTone[2] Tomasz Malisiewicz[2] Andrew Rabinovich[2]
[1] ETH Zurich [2] Magic Leap, Inc.

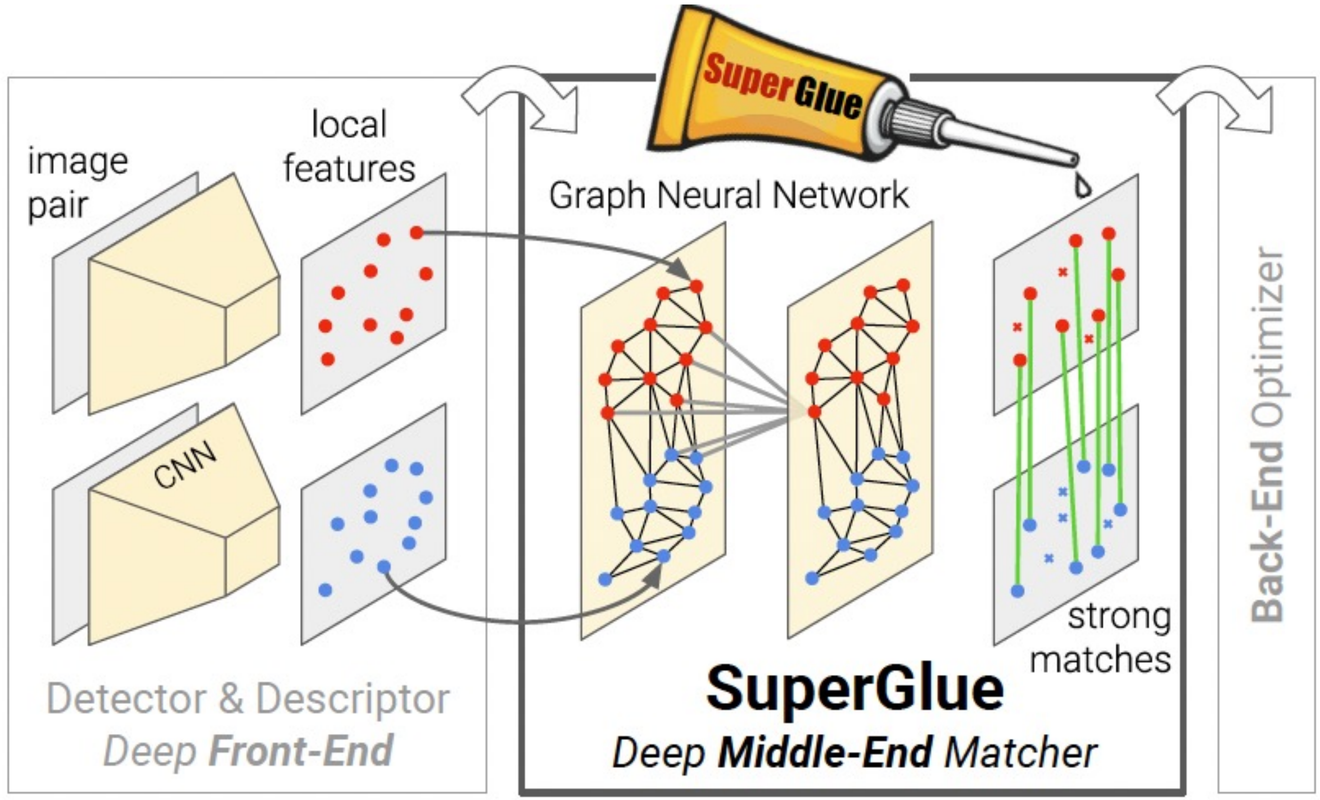


Figure 1: **Feature matching with SuperGlue.** Our approach establishes pointwise correspondences from off-the-shelf local features: it acts as a middle-end between handcrafted or learned front-end and back-end. SuperGlue uses a graph neural network and attention to solve an assignment optimization problem, and handles partial point visibility and occlusion elegantly, producing a partial assignment.