Stereo II

CSE 576

Ali Farhadi

Several slides from Larry Zitnick and Steve Seitz

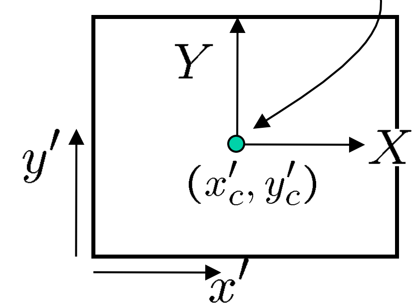
Camera parameters

A camera is described by several parameters

- Translation T of the optical center from the origin of world coords
- Rotation R of the image plane
- focal length f, principle point (x'_c, y'_c), pixel size (s_x, s_y)
- blue parameters are called "extrinsics," red are "intrinsics"

Projection equation

$$\mathbf{x} = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{\Pi X}$$



- The projection matrix models the cumulative effect of all parameters
- Useful to decompose into a series of operations

identity matrix

$$\mathbf{\Pi} = \underbrace{\begin{bmatrix} -fs_x & 0 & x'_c \\ 0 & -fs_y & y'_c \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinsics}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{projection}} \underbrace{\begin{bmatrix} \mathbf{R}_{3x3} & \mathbf{0}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}}_{\text{rotation}} \underbrace{\begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{T}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}}_{\text{translation}}$$

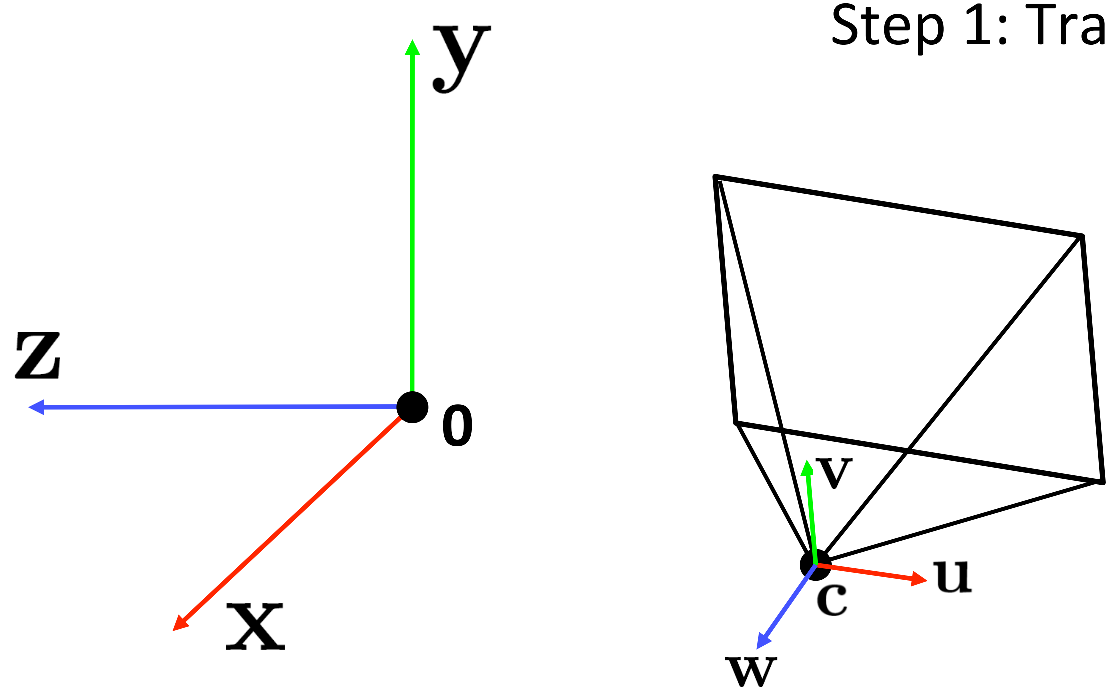
- The definitions of these parameters are not completely standardized
  - especially intrinsics—varies from one book to another

Extrinsics

How do we get the camera to "canonical form"?

- (Center of projection at the origin, x-axis points right, y-axis points up, z-axis points backwards)

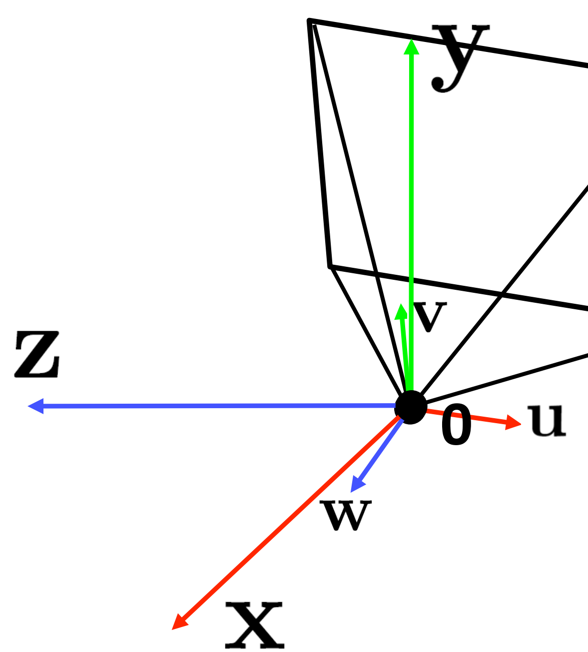
Step 1: Translate by -**c**



Extrinsics

How do we get the camera to "canonical form"?

- (Center of projection at the origin, x-axis points right, y-axis points up, z-axis points backwards)



Step 1: Translate by -**c**

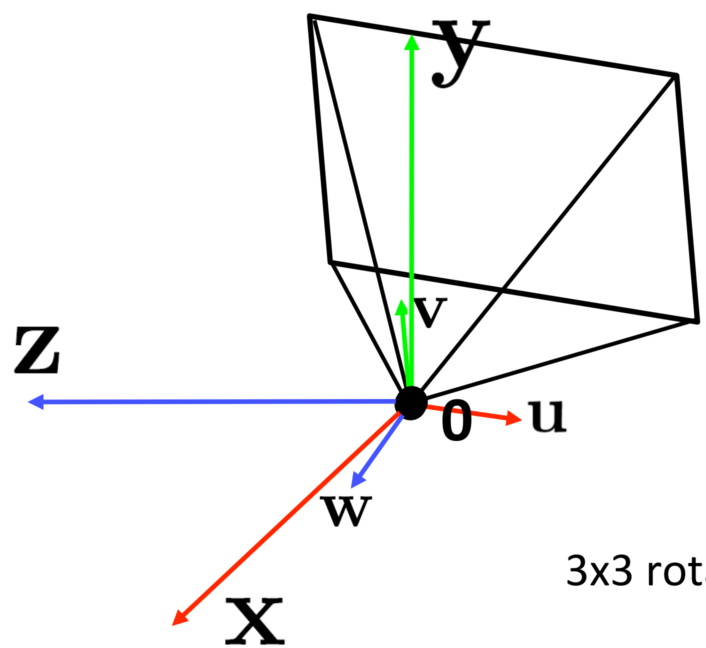
How do we represent translation as a matrix multiplication?

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_{3\times 3} & -\mathbf{c} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}$$

Extrinsics

How do we get the camera to "canonical form"?

- (Center of projection at the origin, x-axis points right, y-axis points up, z-axis points backwards)



Step 1: Translate by -**c**

Step 2: Rotate by **R**

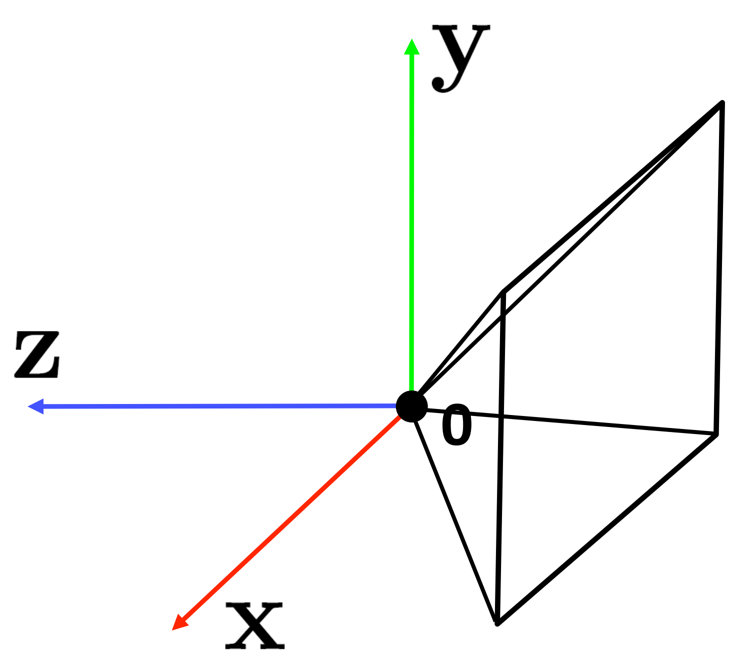
$$\mathbf{R} = \begin{bmatrix} \mathbf{u}^T \\ \mathbf{v}^T \\ \mathbf{w}^T \end{bmatrix}$$

3x3 rotation matrix

Extrinsics

How do we get the camera to "canonical form"?

- (Center of projection at the origin, x-axis points right, y-axis points up, z-axis points backwards)



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

$$\mathbf{R} = \begin{bmatrix} \mathbf{u}^T \\ \mathbf{v}^T \\ \mathbf{w}^T \end{bmatrix}$$

Perspective projection

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

(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)

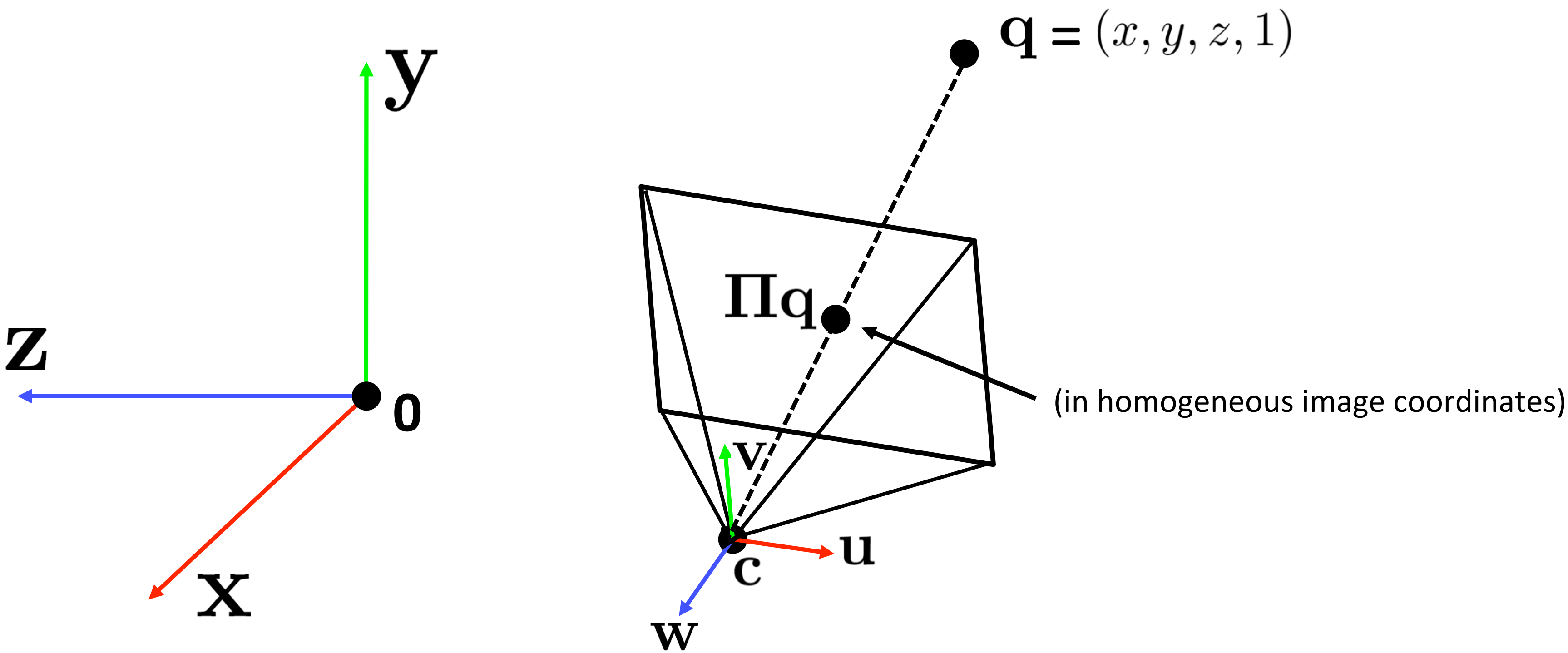
s : **skew** (0 unless pixels are shaped like rhombi/parallelograms)

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

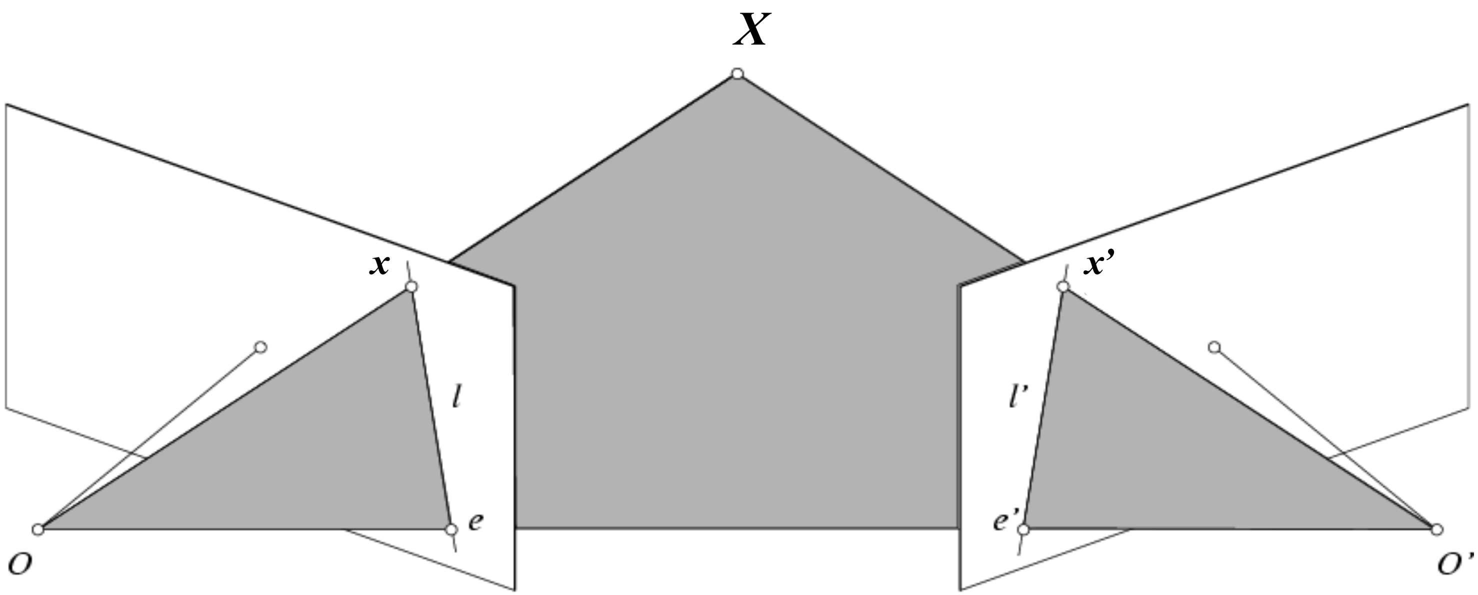
Projection matrix

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

Projection matrix

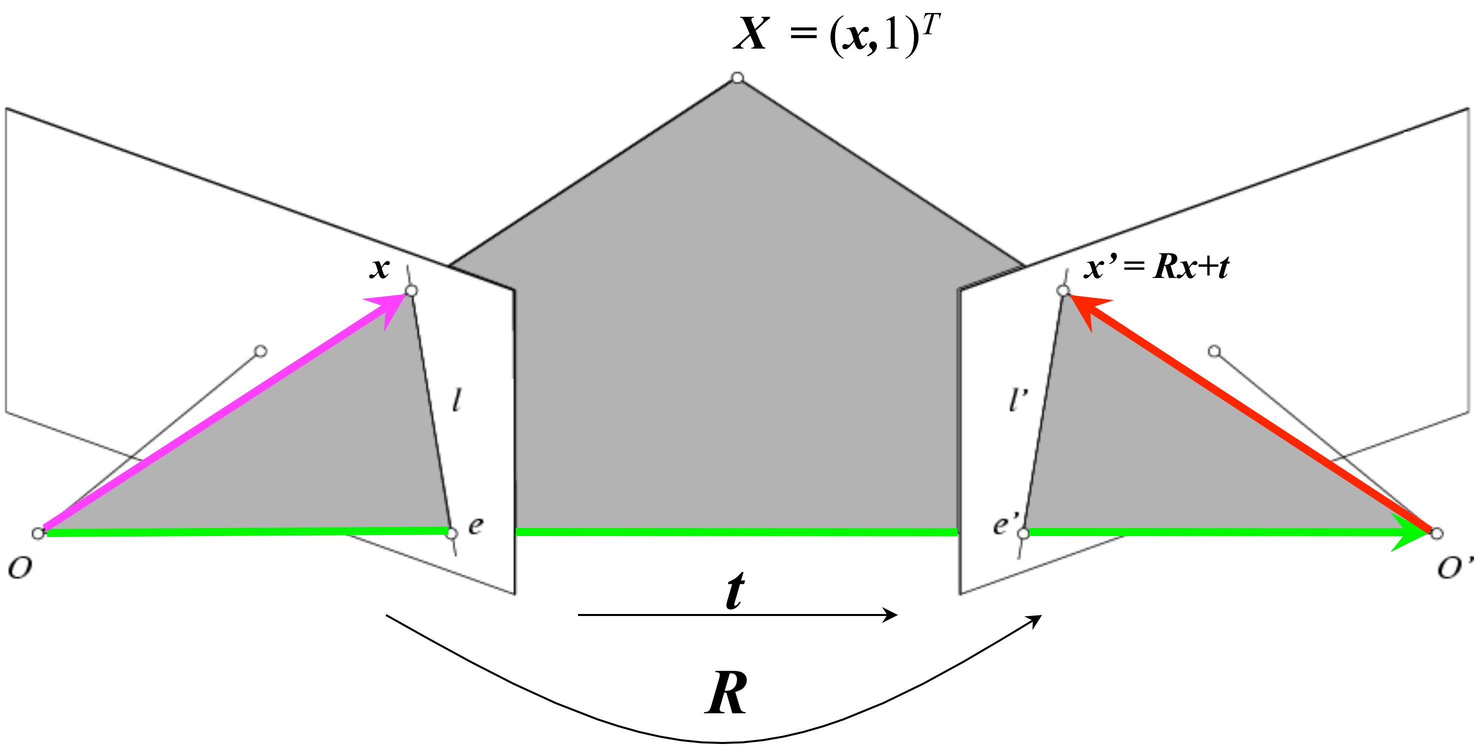


Epipolar constraint: Calibrated case



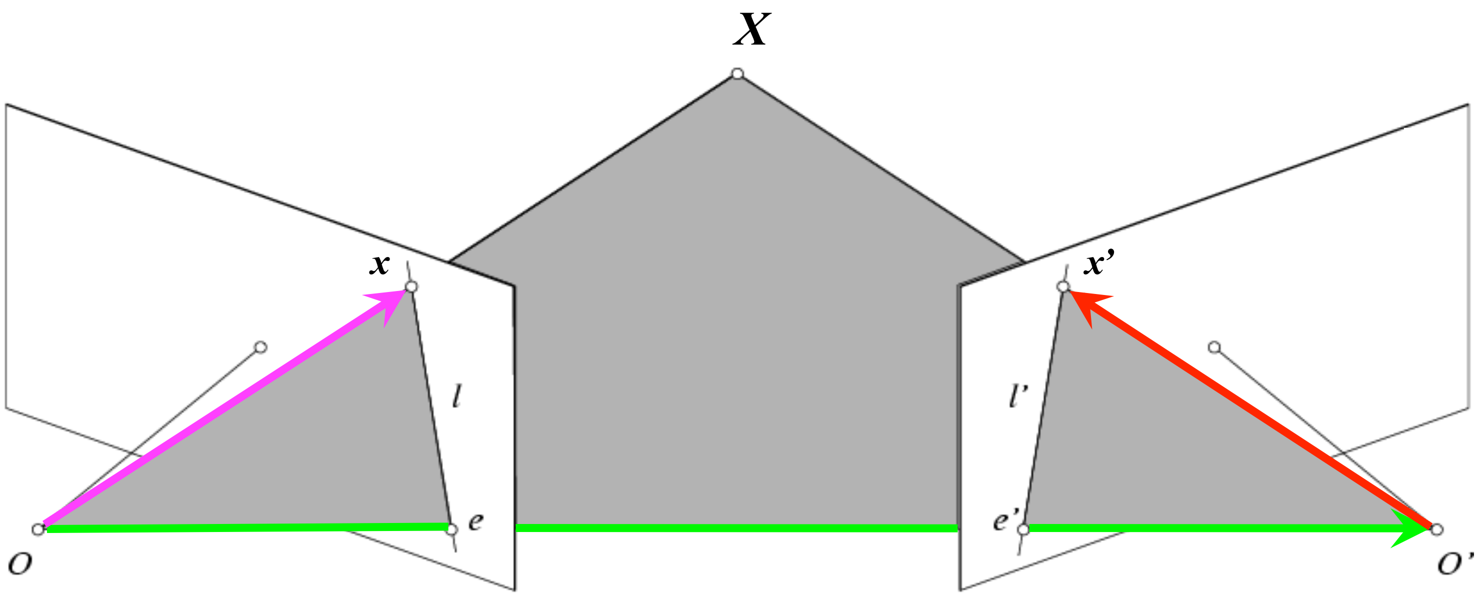
- Assume that the intrinsic and extrinsic parameters of the cameras are known
- We can multiply the projection matrix of each camera (and the image points) by the inverse of the calibration matrix to get *normalized* image coordinates
- We can also set the global coordinate system to the coordinate system of the first camera. Then the projection matrices of the two cameras can be written as $[\mathbf{I} \mid \mathbf{0}]$ and $[\mathbf{R} \mid \mathbf{t}]$

Epipolar constraint: Calibrated case



The vectors ***Rx***, ***t***, and ***x'*** are coplanar

Epipolar constraint: Calibrated case

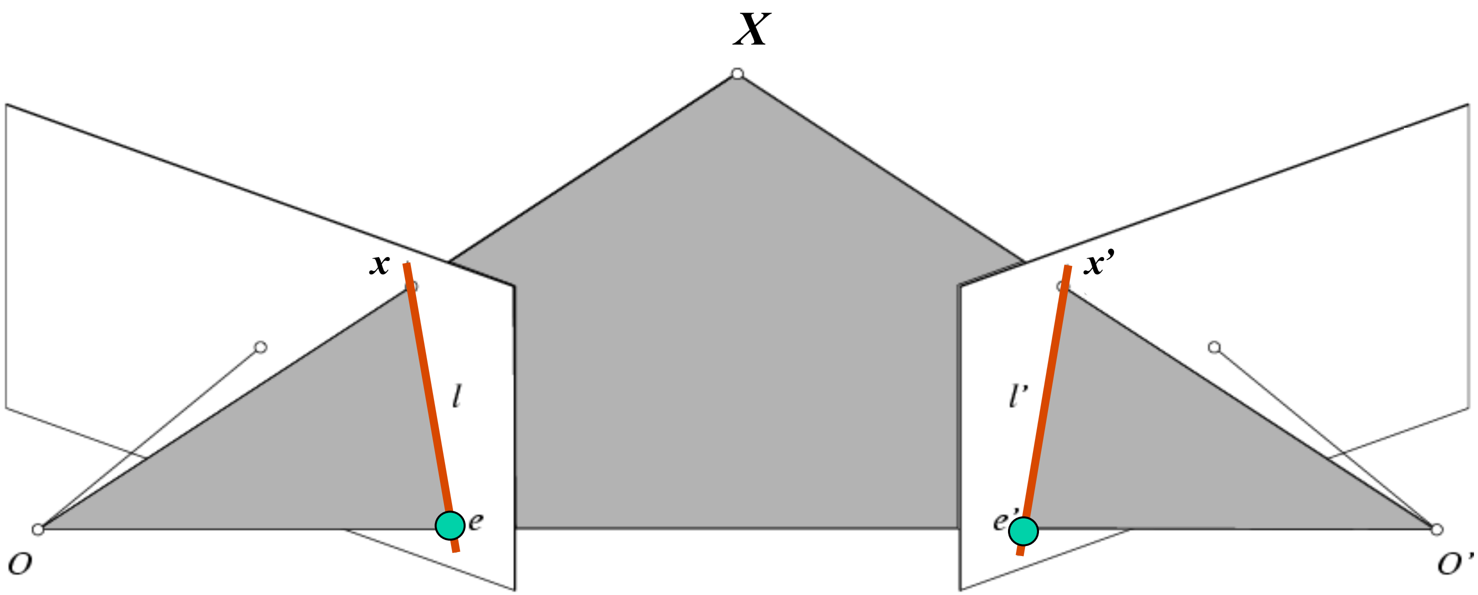


$$\boldsymbol{x}' \cdot [\boldsymbol{t} \times (\boldsymbol{R}\boldsymbol{x})] = 0 \quad \Rightarrow \quad \boldsymbol{x}'^T \boldsymbol{E}\, \boldsymbol{x} = 0 \quad \text{with} \quad \boldsymbol{E} = [\boldsymbol{t}_\times]\boldsymbol{R}$$

Essential Matrix
(Longuet-Higgins, 1981)

The vectors ***Rx***, ***t***, and ***x'*** are coplanar

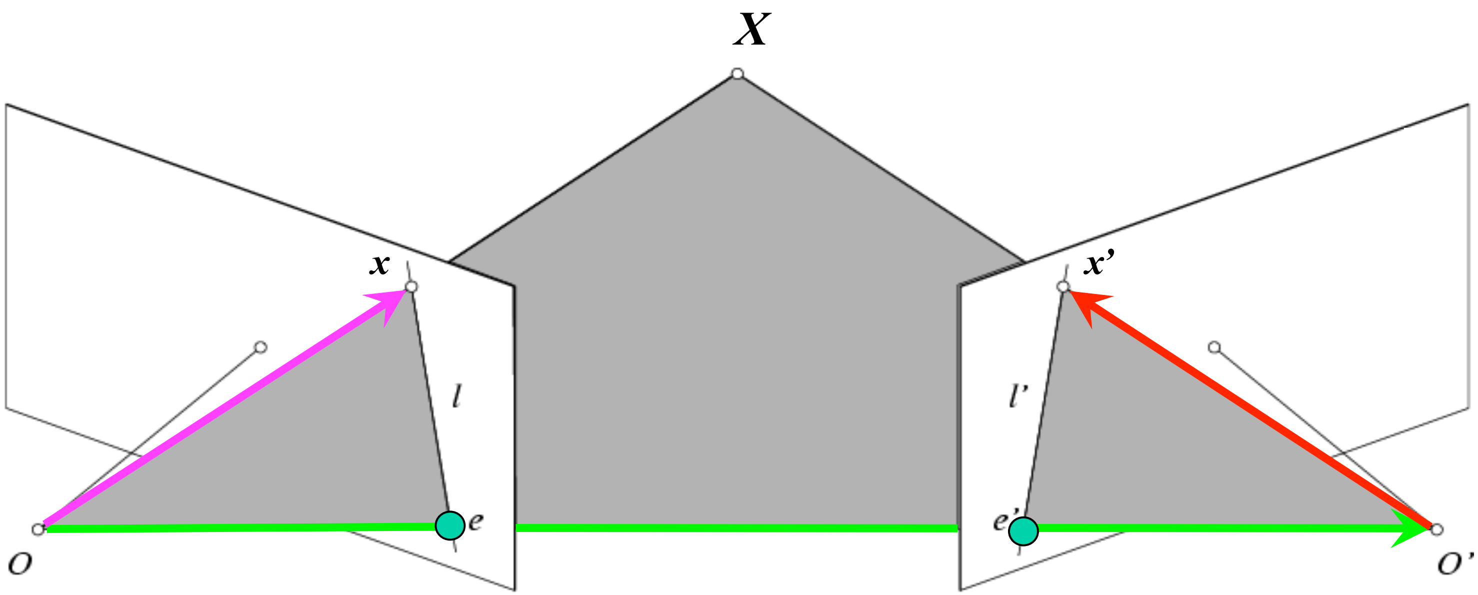
Epipolar constraint: Calibrated case



$$\boldsymbol{x}' \cdot [\boldsymbol{t} \times (\boldsymbol{R}\boldsymbol{x})] = 0 \quad \Rightarrow \quad \boldsymbol{x}'^T \boldsymbol{E}\, \boldsymbol{x} = 0 \quad \text{with} \quad \boldsymbol{E} = [\boldsymbol{t}_\times]\boldsymbol{R}$$

- $\boldsymbol{E}\,\boldsymbol{x}$ is the epipolar line associated with $\boldsymbol{x}$ ($\boldsymbol{l}' = \boldsymbol{E}\,\boldsymbol{x}$)
- $\boldsymbol{E}^T\boldsymbol{x}'$ is the epipolar line associated with $\boldsymbol{x}'$ ($\boldsymbol{l} = \boldsymbol{E}^T\boldsymbol{x}'$)
- $\boldsymbol{E}\,\boldsymbol{e} = 0$ and $\boldsymbol{E}^T\boldsymbol{e}' = 0$
- $\boldsymbol{E}$ is singular (rank two)
- $\boldsymbol{E}$ has five degrees of freedom

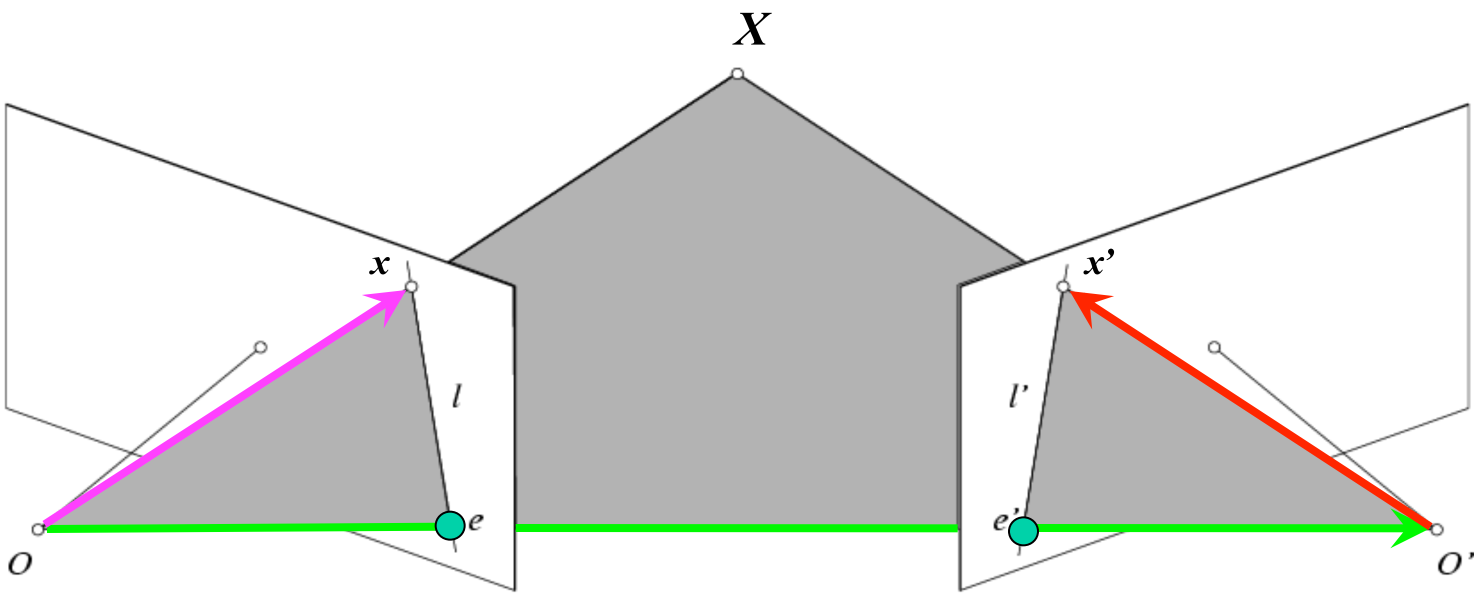
Epipolar constraint: Uncalibrated case



- The calibration matrices ***K*** and ***K'*** of the two cameras are unknown
- We can write the epipolar constraint in terms of *unknown* normalized coordinates:

$$\hat{\mathbf{x}}'^{T} \boldsymbol{E} \hat{\mathbf{x}} = 0 \qquad\qquad \hat{\mathbf{x}} = \boldsymbol{K}^{-1}\mathbf{x}, \quad \hat{\mathbf{x}}' = \boldsymbol{K}'^{-1}\hat{\mathbf{x}}'$$

Epipolar constraint: Uncalibrated case



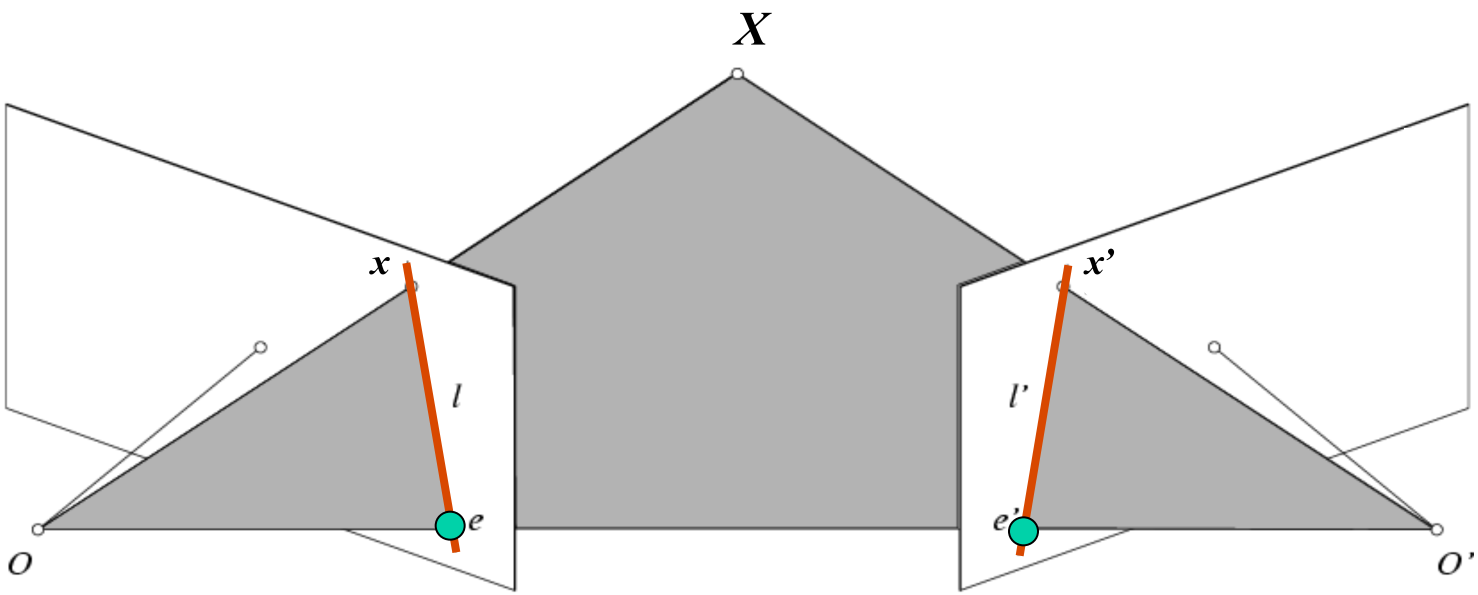
$$\hat{\boldsymbol{x}}'^{T} \boldsymbol{E} \hat{\boldsymbol{x}} = 0 \quad \Rightarrow \quad \boldsymbol{x}'^{T} \boldsymbol{F} \boldsymbol{x} = 0 \quad \text{with} \quad \boldsymbol{F} = \boldsymbol{K}'^{-T} \boldsymbol{E} \boldsymbol{K}^{-1}$$

$$\hat{\boldsymbol{x}} = \boldsymbol{K}^{-1} \boldsymbol{x}$$

$$\hat{\boldsymbol{x}}' = \boldsymbol{K}'^{-1} \boldsymbol{x}'$$

Fundamental Matrix
(Faugeras and Luong, 1992)

Epipolar constraint: Uncalibrated case



$$\hat{\boldsymbol{x}}'^{T} \boldsymbol{E} \hat{\boldsymbol{x}} = 0 \quad \Rightarrow \quad \boldsymbol{x}'^{T} \boldsymbol{F} \boldsymbol{x} = 0 \quad \text{with} \quad \boldsymbol{F} = \boldsymbol{K}'^{-T} \boldsymbol{E} \boldsymbol{K}^{-1}$$

- $\boldsymbol{F}\boldsymbol{x}$ is the epipolar line associated with $\boldsymbol{x}$ ($\boldsymbol{l}' = \boldsymbol{F}\boldsymbol{x}$)
- $\boldsymbol{F}^T\boldsymbol{x}'$ is the epipolar line associated with $\boldsymbol{x}'$ ($\boldsymbol{l}' = \boldsymbol{F}^T\boldsymbol{x}'$)
- $\boldsymbol{F}\boldsymbol{e} = 0$ and $\boldsymbol{F}^T\boldsymbol{e}' = 0$
- $\boldsymbol{F}$ is singular (rank two)
- $\boldsymbol{F}$ has *seven* degrees of freedom

The eight-point algorithm

$$\boldsymbol{x} = (u, v, 1)^T, \quad \boldsymbol{x}' = (u', v', 1)$$

$$\begin{bmatrix} u' & v' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$$

$$\underbrace{\begin{bmatrix} u'u & u'v & u' & v'u & v'v & v' & u & v & 1 \end{bmatrix}}_{\mathrm{A}} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

Minimize:

$$\sum_{i=1}^{N} (\boldsymbol{x}_i'^T \boldsymbol{F} \boldsymbol{x}_i)^2$$

under the constraint

$||\boldsymbol{F}||^2 = 1$

Smallest eigenvalue of $\mathrm{A}^T\mathrm{A}$

The eight-point algorithm

- Meaning of error $\sum_{i=1}^{N} (\boldsymbol{x}_i'^T \boldsymbol{F} \boldsymbol{x}_i)^2$:

  sum of squared *algebraic* distances between points $\boldsymbol{x}'_i$ and epipolar lines $\boldsymbol{F}\boldsymbol{x}_i$ (or points $\boldsymbol{x}_i$ and epipolar lines $\boldsymbol{F}^T\boldsymbol{x}'_i$)
- Nonlinear approach: minimize sum of squared *geometric* distances

$$\sum_{i=1}^{N} \left[\mathrm{d}^2(\boldsymbol{x}'_i, \boldsymbol{F}\boldsymbol{x}_i) + \mathrm{d}^2(\boldsymbol{x}_i, \boldsymbol{F}^T\boldsymbol{x}'_i) \right]$$

Problem with eight-point algorithm

$$\begin{bmatrix} u'u & u'v & u' & v'u & v'v & v' & u & v \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \end{bmatrix} = 0$$

Problem with eight-point algorithm

250906.36	183269.57	921.81	200931.10	146766.13	738.21	272.19	198.81
2692.28	131633.03	176.27	6196.73	302975.59	405.71	15.27	746.79
416374.23	871684.30	935.47	408110.89	854384.92	916.90	445.10	931.81
191183.60	171759.40	410.27	416435.62	374125.90	893.65	465.99	418.65
48988.86	30401.76	57.89	298604.57	185309.58	352.87	846.22	525.15
164786.04	546559.67	813.17	1998.37	6628.15	9.86	202.65	672.14
116407.01	2727.75	138.89	169941.27	3982.21	202.77	838.12	19.64
135384.58	75411.13	198.72	411350.03	229127.78	603.79	681.28	379.48

$$\begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \end{bmatrix} = -1$$

Poor numerical conditioning

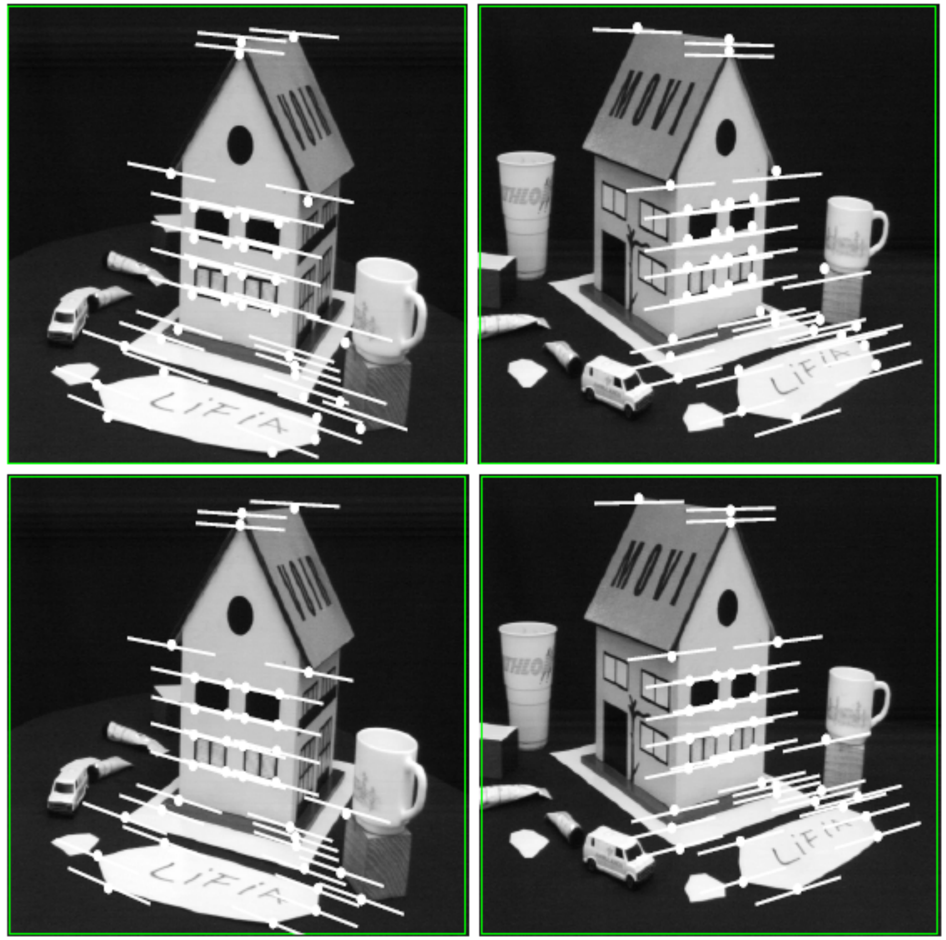
Can be fixed by rescaling the data

The normalized eight-point algorithm

(Hartley, 1995)

- Center the image data at the origin, and scale it so the mean squared distance between the origin and the data points is 2 pixels
- Use the eight-point algorithm to compute ***F*** from the normalized points
- Enforce the rank-2 constraint (for example, take SVD of ***F*** and throw out the smallest singular value)
- Transform fundamental matrix back to original units: if ***T*** and ***T'*** are the normalizing transformations in the two images, then the fundamental matrix in original coordinates is $T'^{T} F\, T$

Comparison of estimation algorithms



	8-point	Normalized 8-point	Nonlinear least squares
Av. Dist. 1	2.33 pixels	0.92 pixel	0.86 pixel
Av. Dist. 2	2.18 pixels	0.85 pixel	0.80 pixel

Moving on to stereo…

Fuse a calibrated binocular stereo pair to produce a depth image

image 1

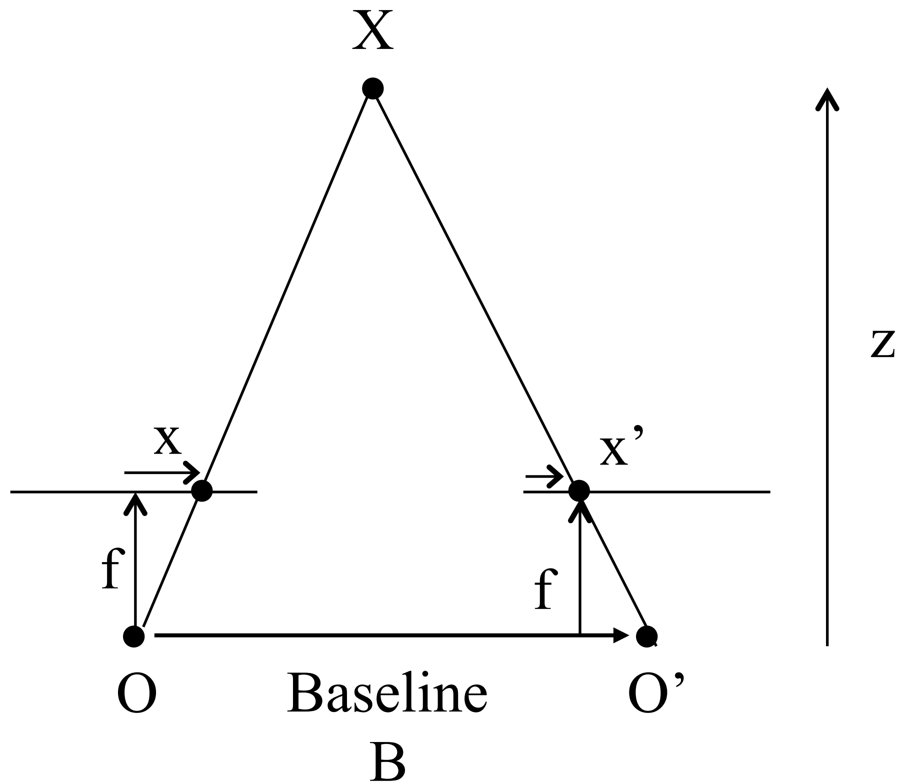
image 2

Dense depth map

Many of these slides adapted from Steve Seitz and Lana Lazebnik

Depth from disparity

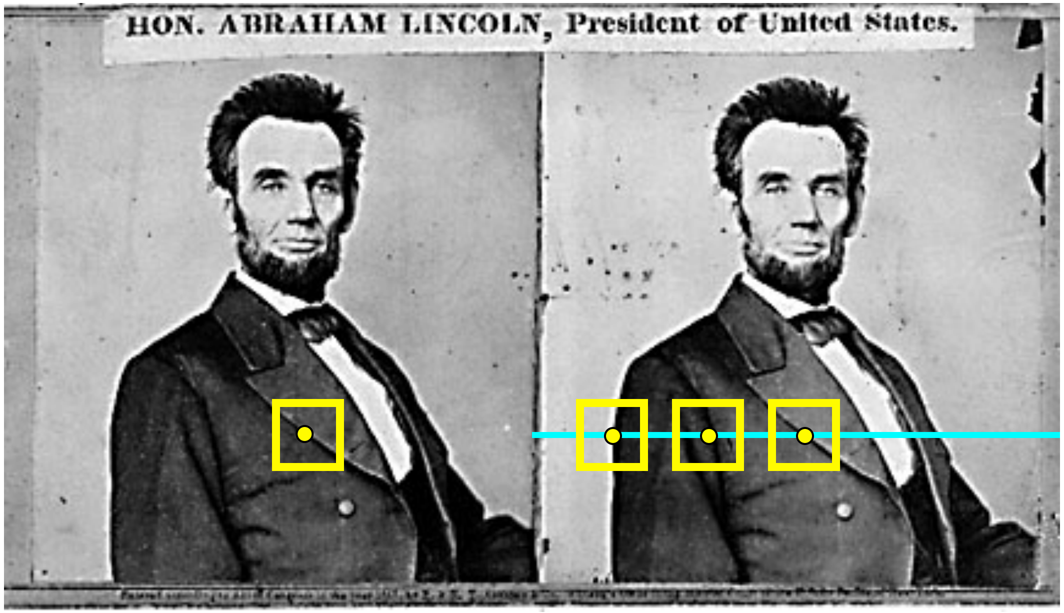
$$\frac{x - x'}{O - O'} = \frac{f}{z}$$



$$disparity = x - x' = \frac{B \cdot f}{z}$$

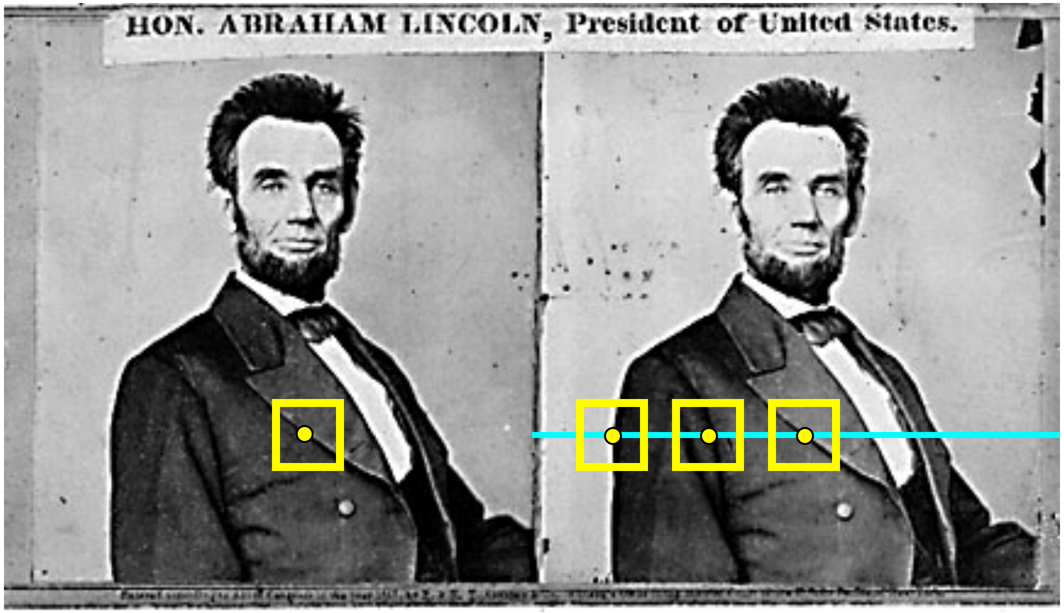
Disparity is inversely proportional to depth.

Basic stereo matching algorithm



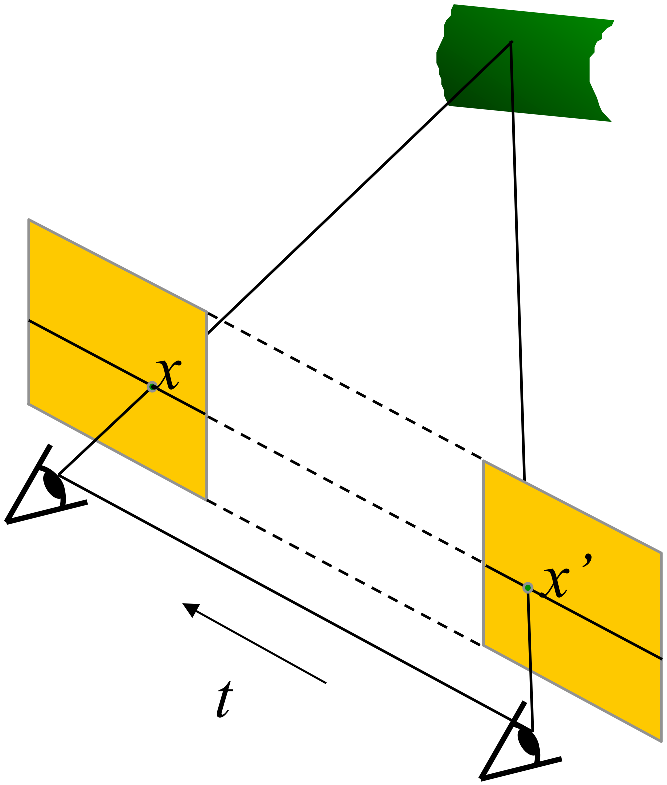
- If necessary, rectify the two stereo images to transform epipolar lines into scanlines
- For each pixel x in the first image
  - Find corresponding epipolar scanline in the right image
  - Search the scanline and pick the best match x’
  - Compute disparity x-x’ and set depth(x) = fB/(x-x’)

Basic stereo matching algorithm



- For each pixel in the first image
  - Find corresponding epipolar line in the right image
  - Search along epipolar line and pick the best match
  - Triangulate the matches to get depth information
- Simplest case: epipolar lines are scanlines
  - When does this happen?

Simplest Case: Parallel images



Epipolar constraint:

$$x^T E x' = 0, \quad E = t \times R$$

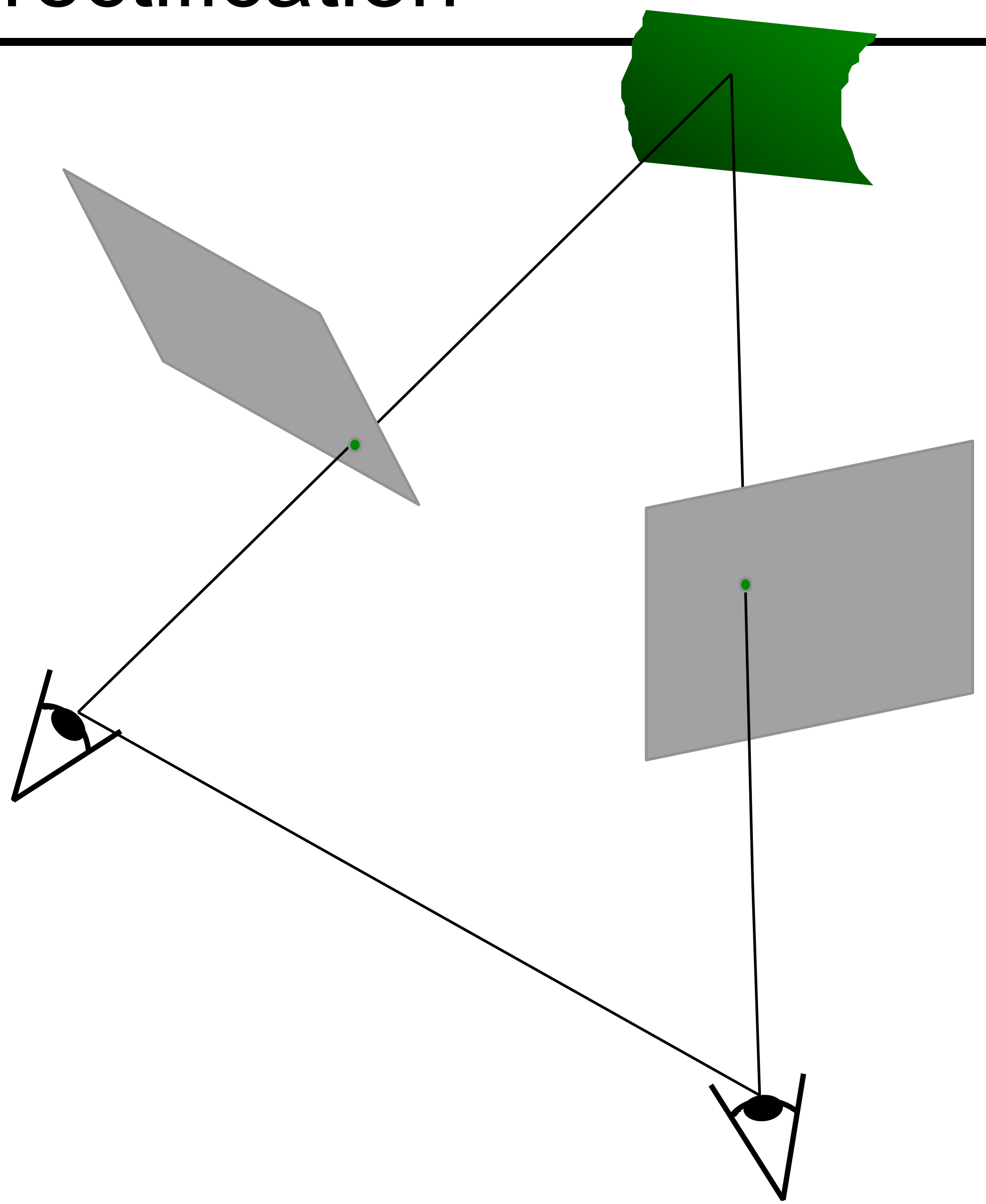
$$R = I \qquad t = (T, 0, 0)$$

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

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

The y-coordinates of corresponding points are the same

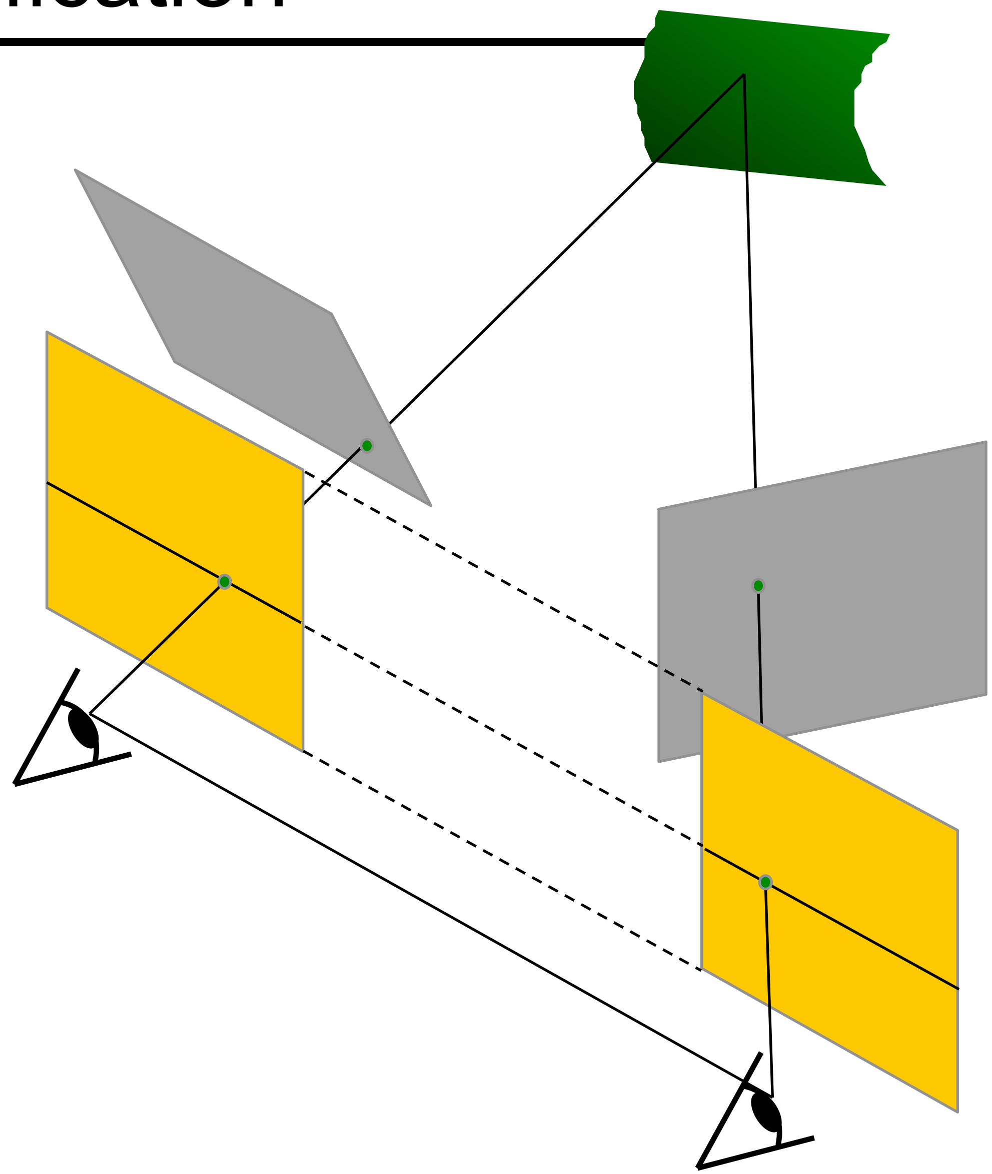
Stereo image rectification



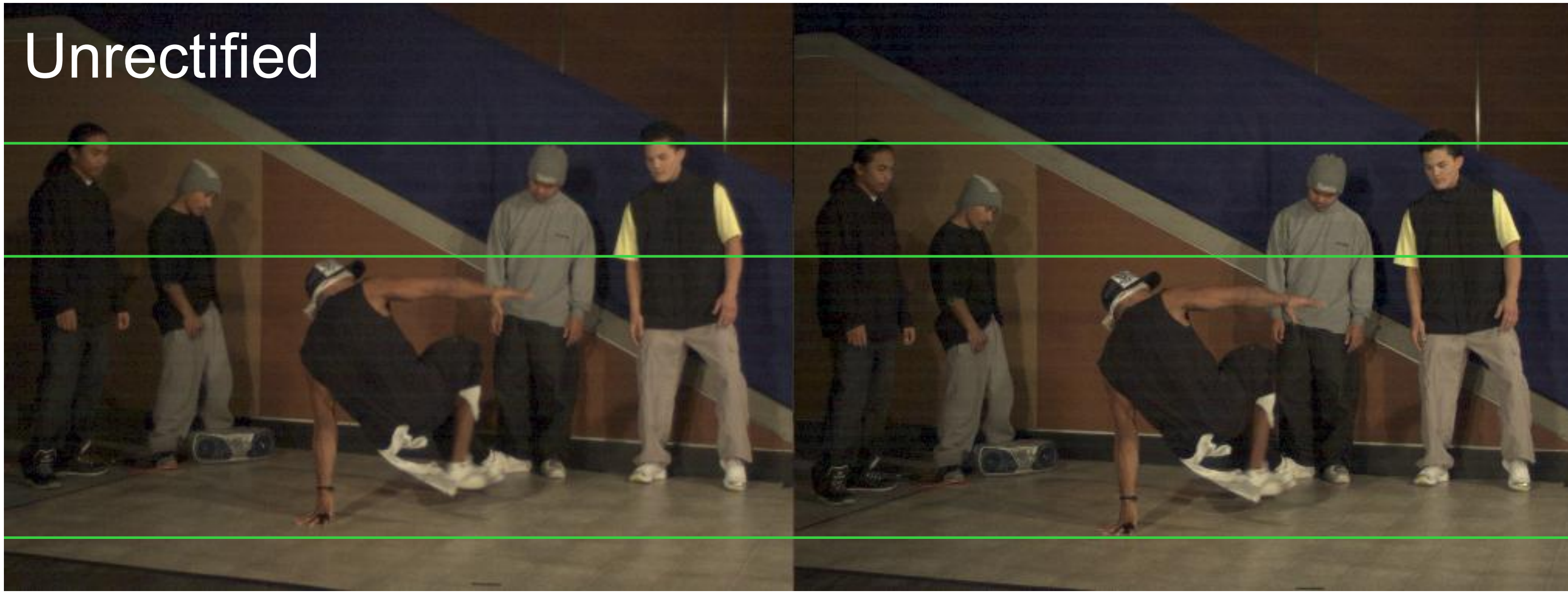
Stereo image rectification

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

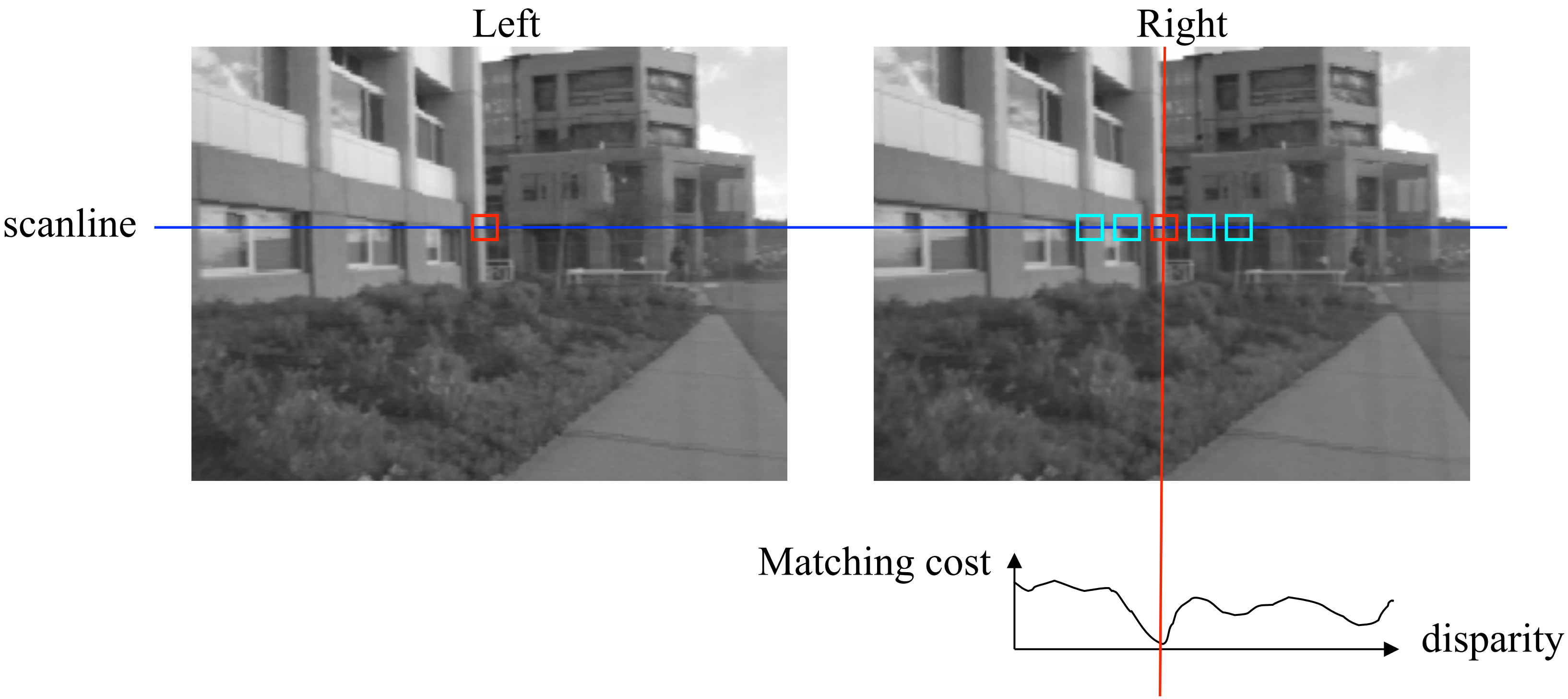
➢ C. Loop and Z. Zhang. [Computing Rectifying Homographies for Stereo Vision](). IEEE Conf. Computer Vision and Pattern Recognition, 1999.



Example

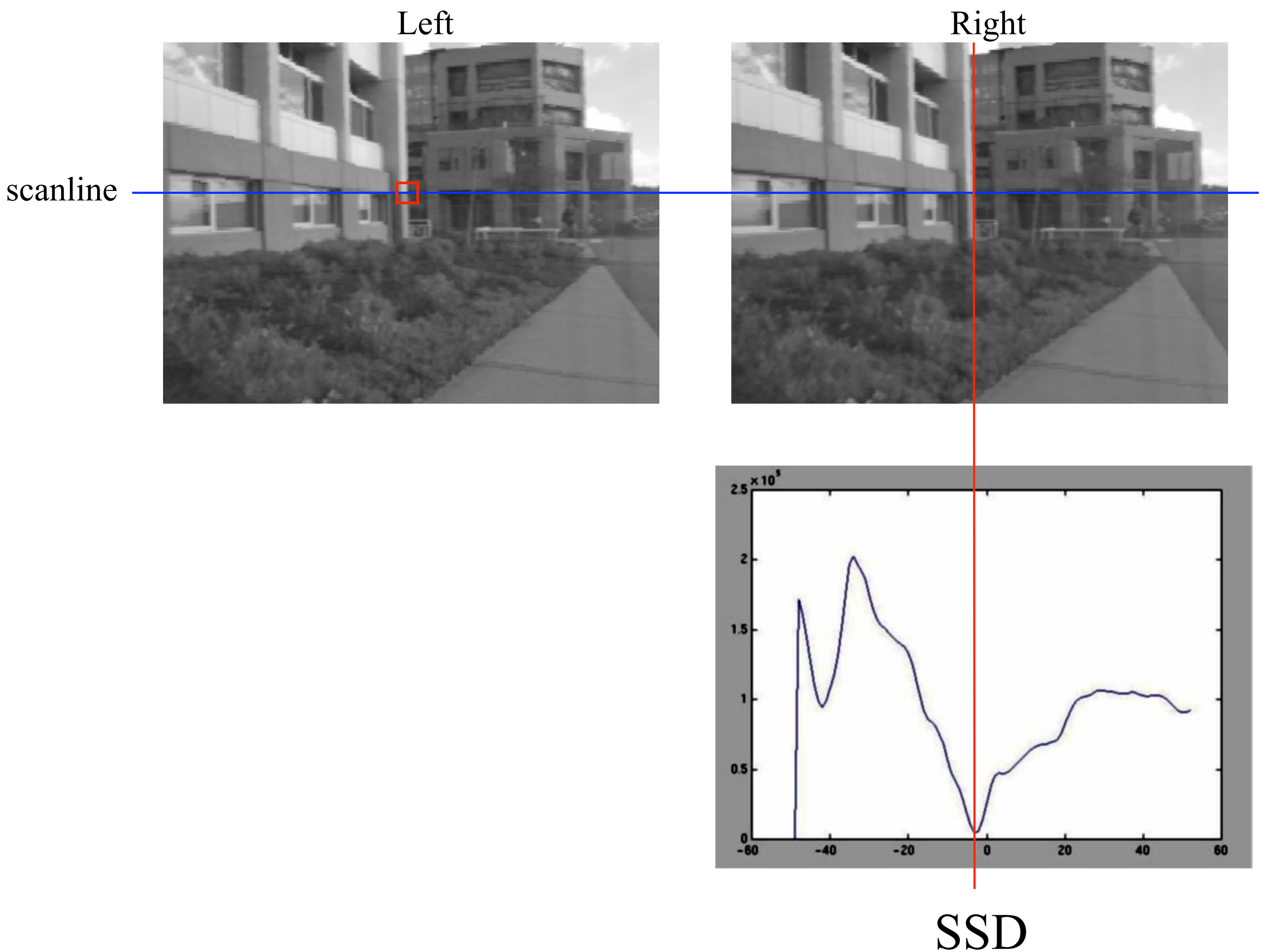


Correspondence search

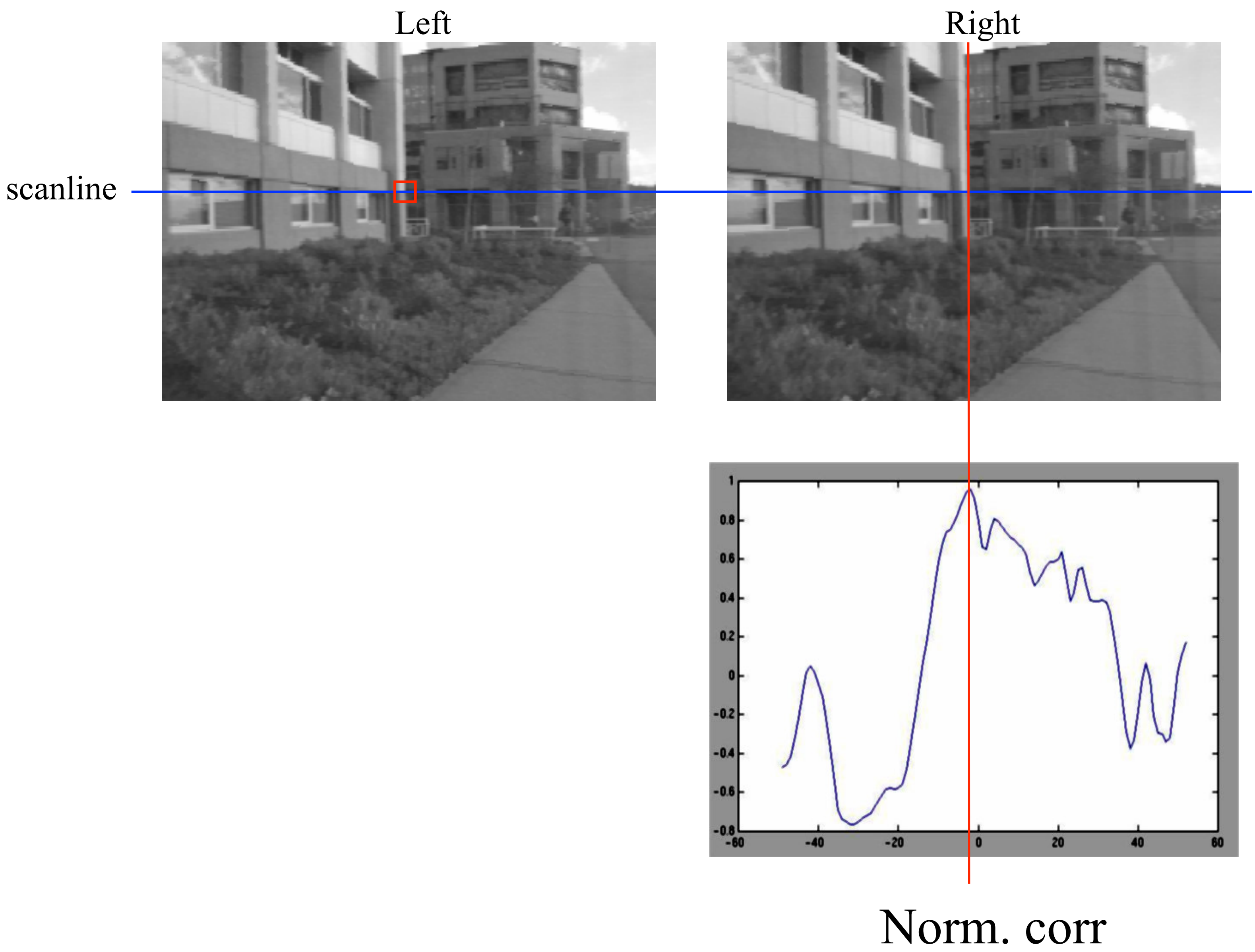


- Slide a window along the right scanline and compare contents of that window with the reference window in the left image
- Matching cost: SSD or normalized correlation

Correspondence search

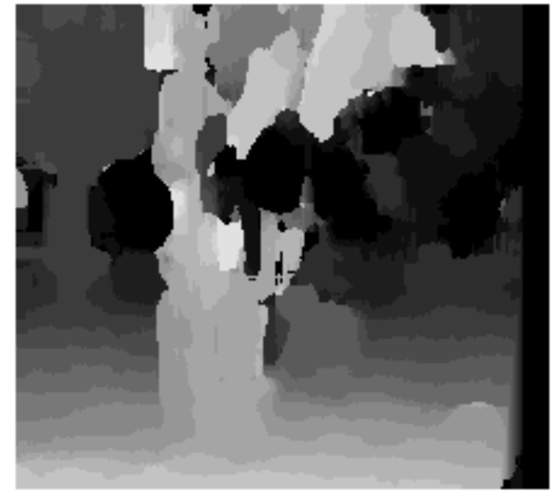


Correspondence search



Effect of window size

W = 3

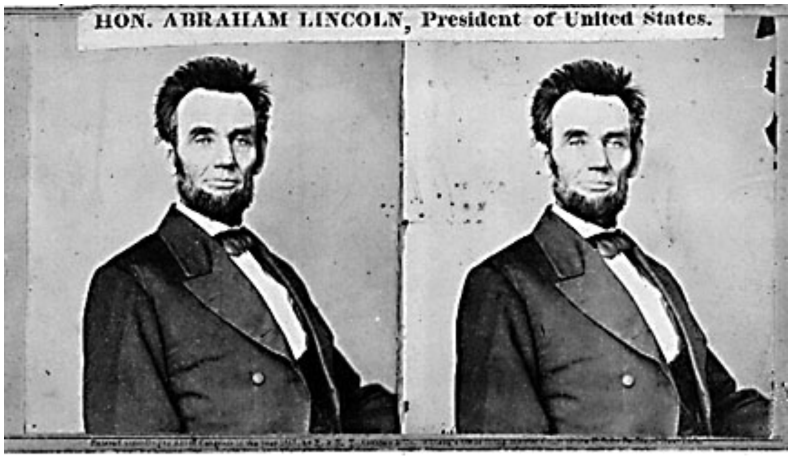


W = 20

- Smaller window
  + More detail
  – More noise

- Larger window
  + Smoother disparity maps
  – Less detail
  – Fails near boundaries

Failures of correspondence search



Textureless surfaces

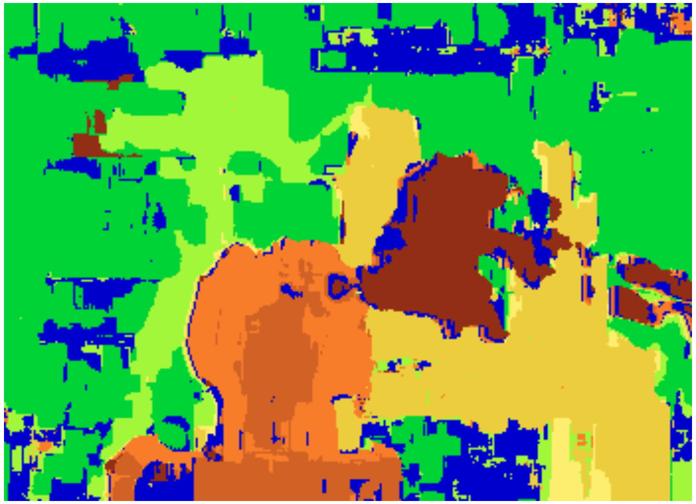
Occlusions, repetition

Non-Lambertian surfaces, specularities

Results with window search

Data

Window-based matching



Ground truth

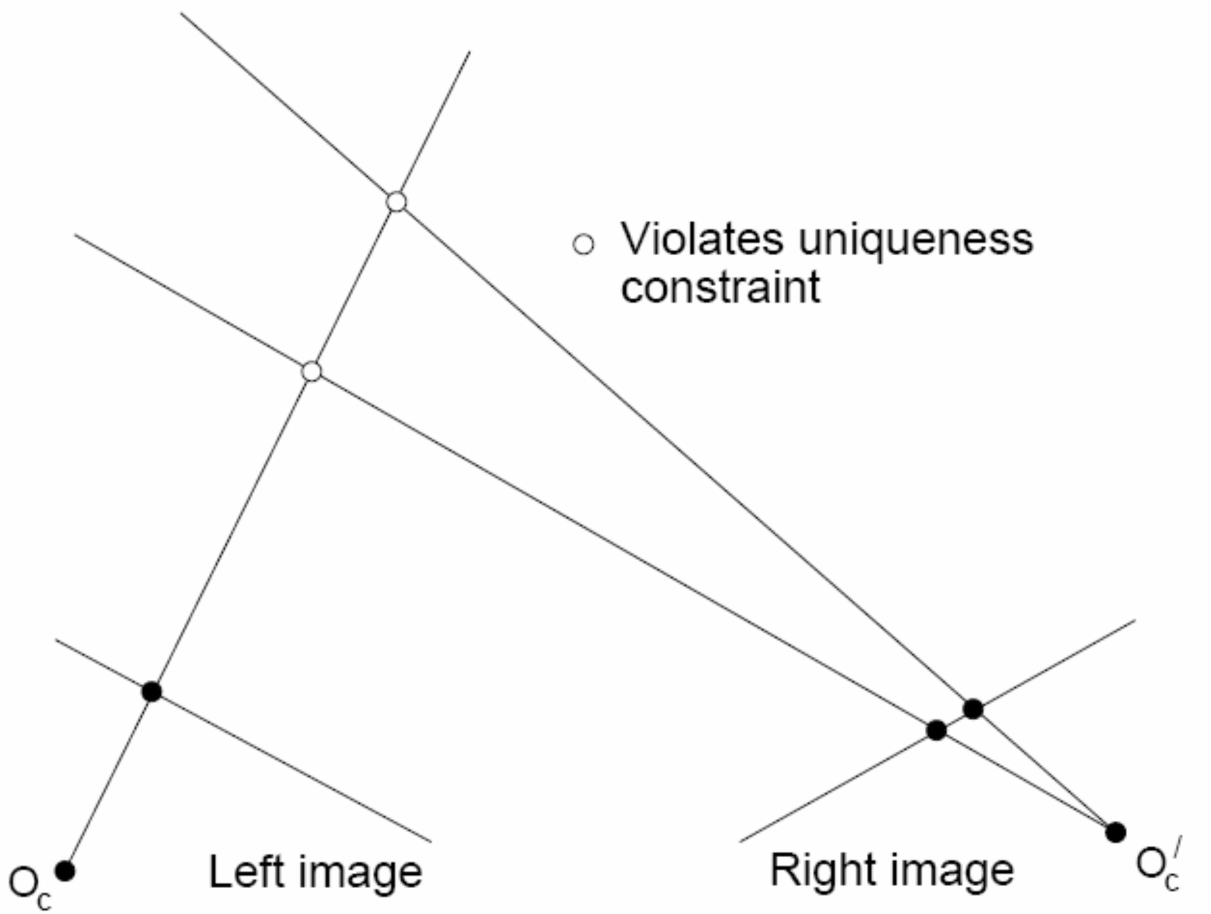
How can we improve window-based matching?

So far, matches are independent for each point

What constraints or priors can we add?

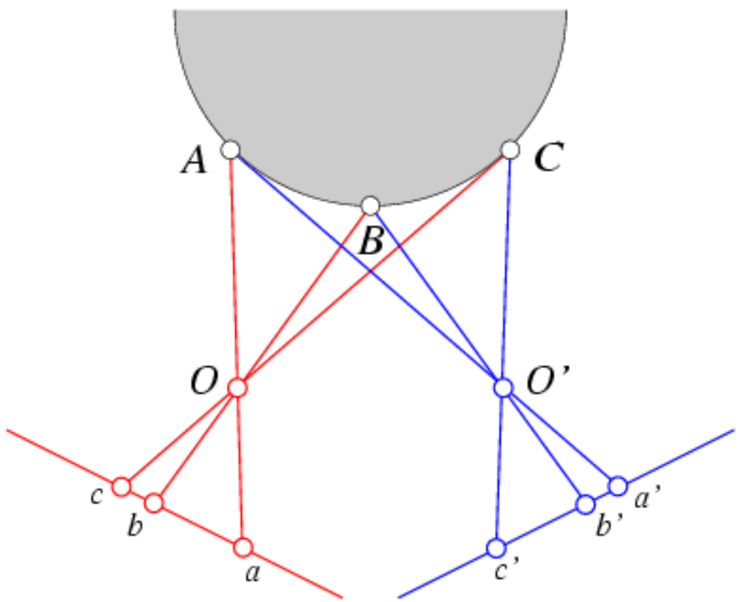
Stereo constraints/priors

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



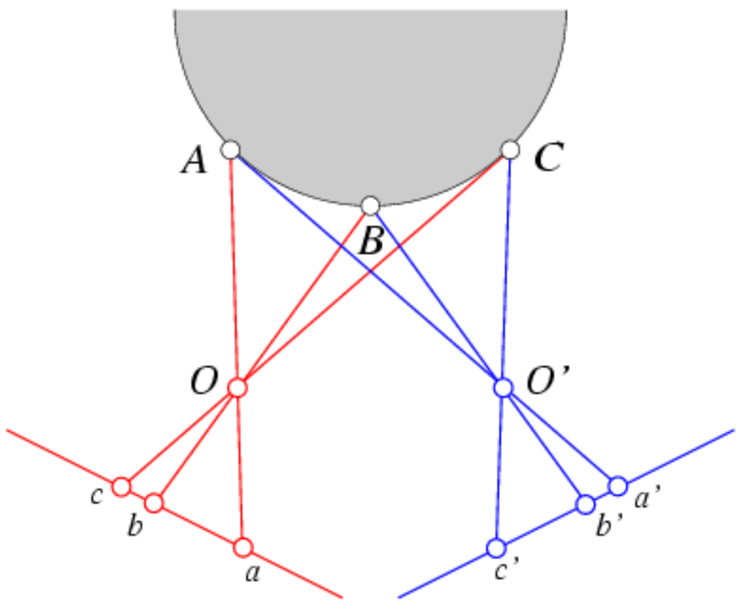
Stereo constraints/priors

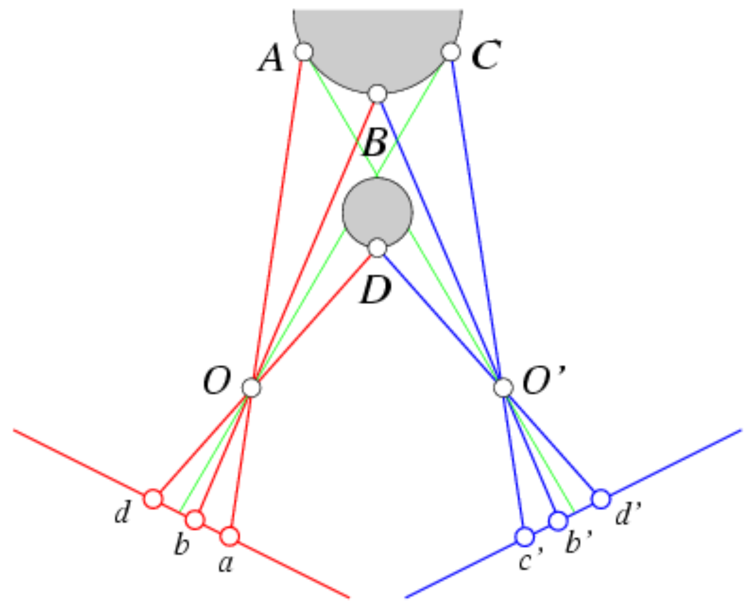
- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image
- Ordering
  - Corresponding points should be in the same order in both views



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



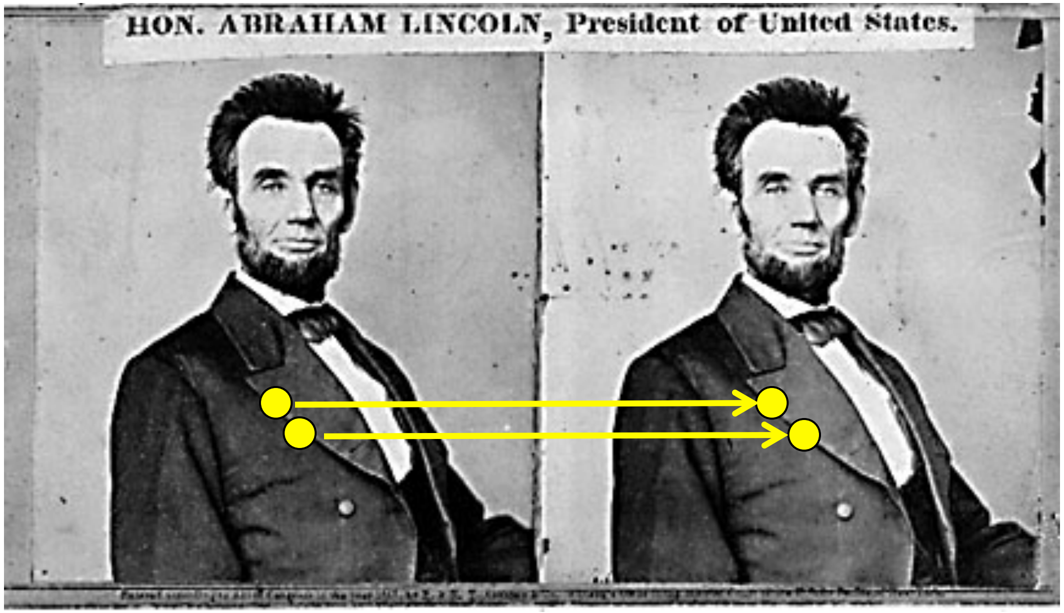


Ordering constraint doesn't hold

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 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

Better objective function

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

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

Adjacent pixels should (usually) move about the same amount

Stereo as energy minimization

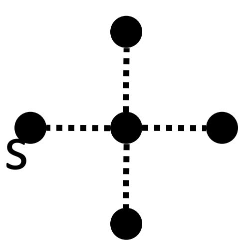
$$E(d) = E_d(d) + \lambda E_s(d)$$

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

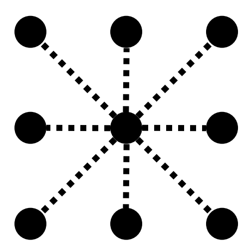
smoothness cost: $$E_s(d) = \sum_{(p,q)\in\mathcal{E}} V(d_p, d_q)$$

$$C(x, y, d(x, y)) =$$ SSD distance between windows *I*(*x*, *y*) and *J*(*x*, *y* + *d*(*x*,*y*))

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



4-connected neighborhood



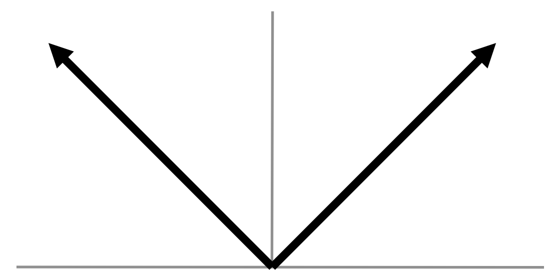
8-connected neighborhood

Smoothness cost

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

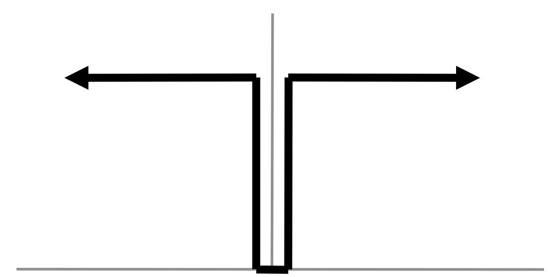
$$V(d_p, d_q) = |d_p - d_q|$$

L_1 distance



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

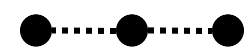
"Potts model"



Dynamic programming

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

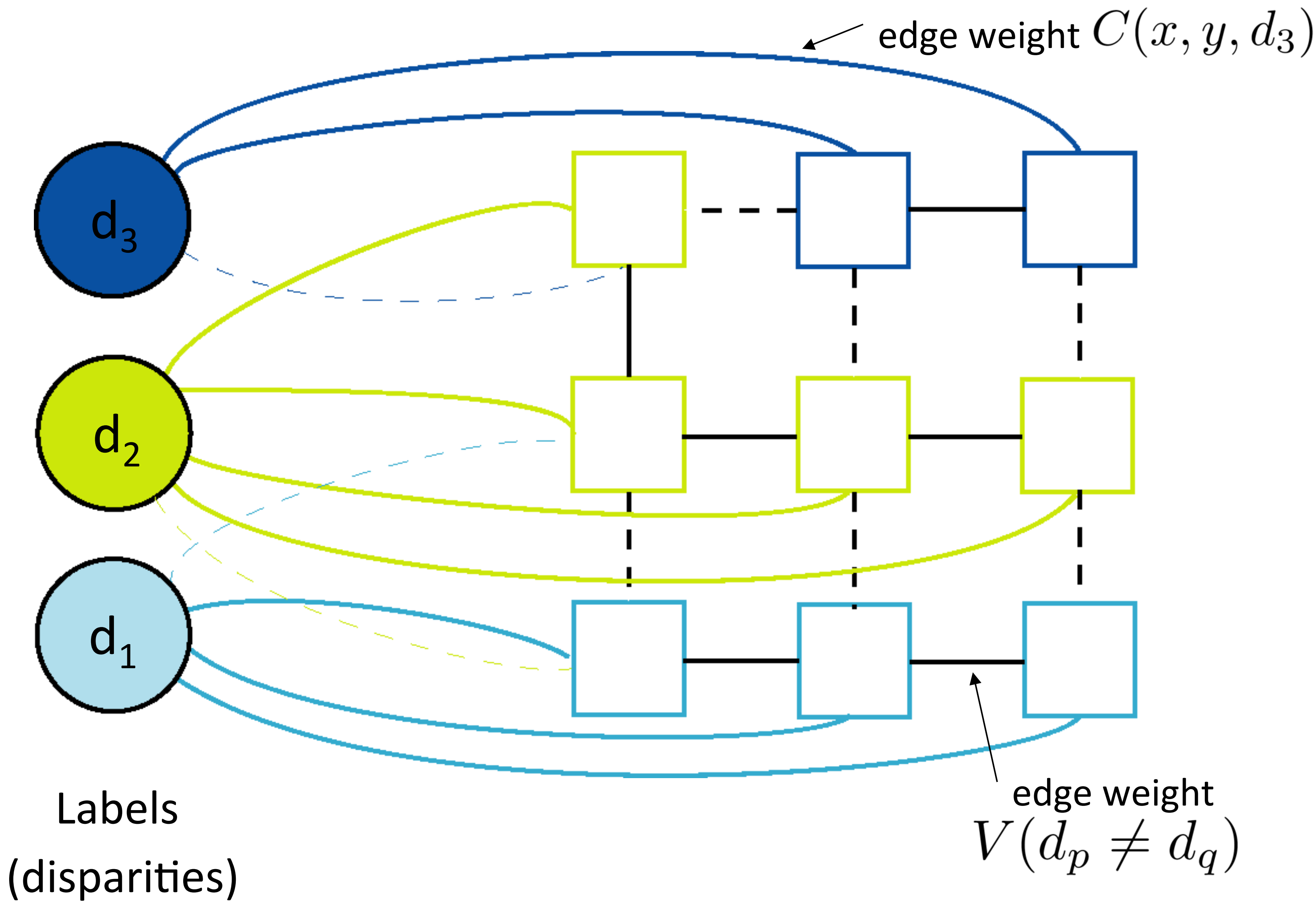
Can minimize this independently per scanline using dynamic programming (DP)



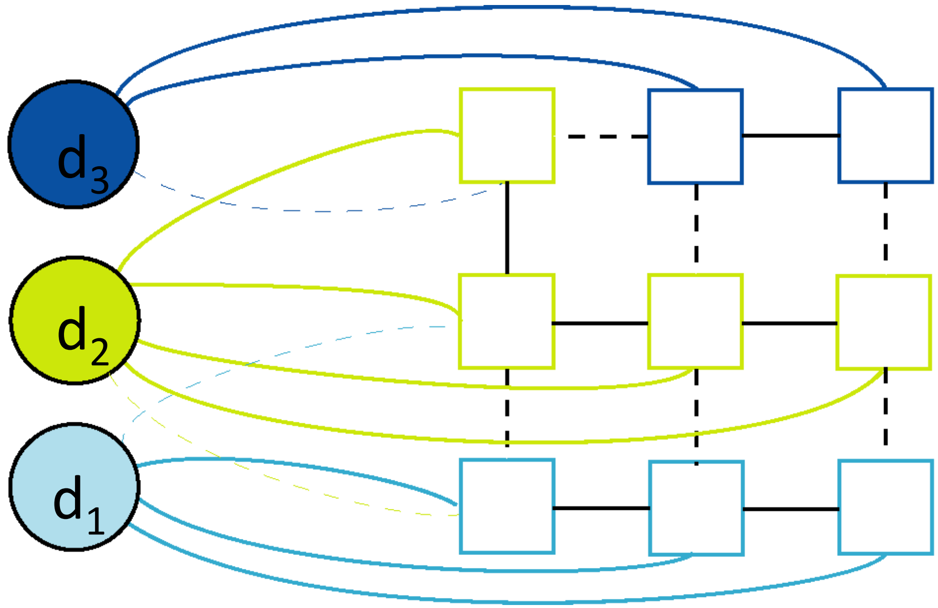
$D(x, y, d)$: minimum cost of solution such that *d*(*x*,*y*) = *d*

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

Energy minimization via graph cuts



Energy minimization via graph cuts



- Graph Cut
  - Delete enough edges so that
    - each pixel is connected to exactly one label node
  - Cost of a cut: sum of deleted edge weights
  - Finding min cost cut equivalent to finding global minimum of energy function

Stereo as energy minimization


$I(x, y)$ $J(x, y)$

y = 141

d

x

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

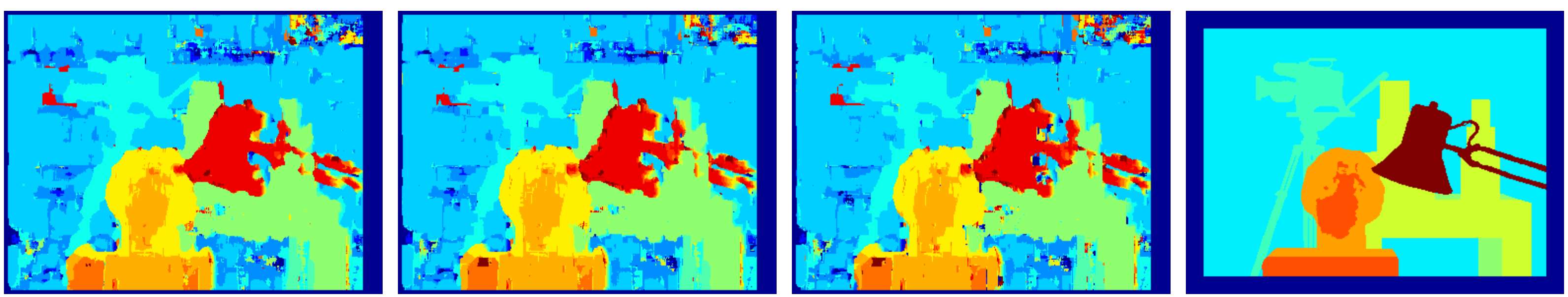
Stereo as energy minimization

Simple pixel / window matching: choose the minimum of each column in the DSI independently:

$$d(x, y) = \arg\min_{d'} \; C(x, y, d')$$

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)	$\frac{\sum_{(i,j)\in W} I_1(i,j).I_2(x+i,y+j)}{\sqrt[2]{\sum_{(i,j)\in W} I_1^2(i,j).\sum_{(i,j)\in W} I_2^2(x+i,y+j)}}$

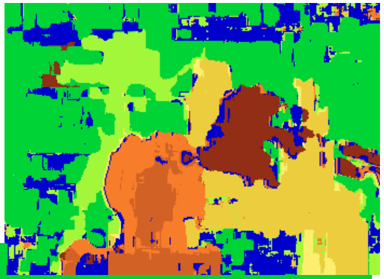


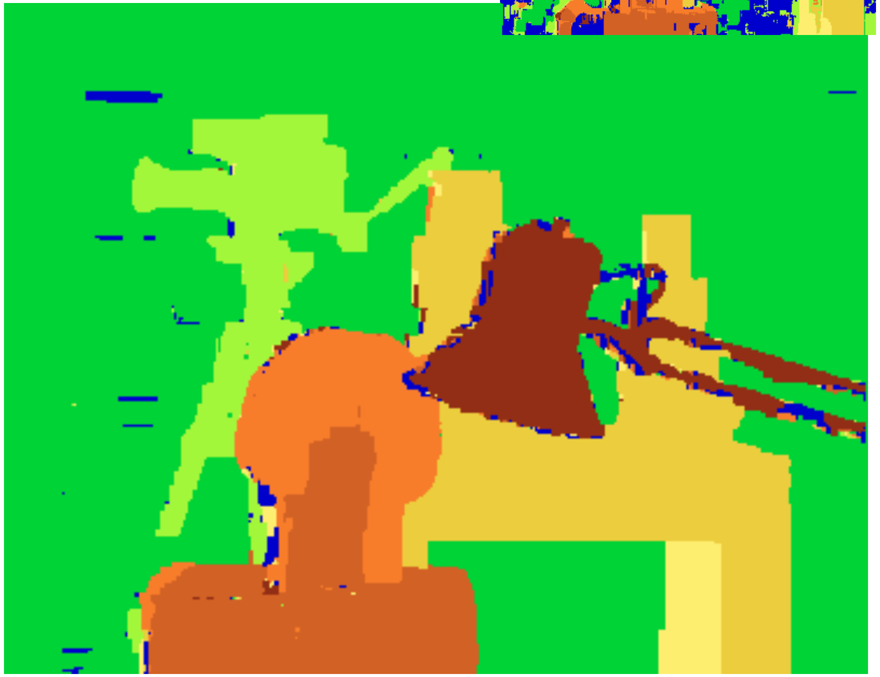
SAD SSD NCC Ground truth

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

Before & After

Before





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/

Real-time stereo

Nomad robot searches for meteorites in Antartica
http://www.frc.ri.cmu.edu/projects/meteorobot/index.html

Used for robot navigation (and other tasks)

- Several software-based real-time stereo techniques have been developed (most based on simple discrete search)

Why does stereo fail?

Fronto-Parallel Surfaces: Depth is constant within the region of local support

Why does stereo fail?

Monotonic Ordering - Points along an epipolar scanline appear in the same order in both stereo images
Occlusion – All points are visible in each image

Why does stereo fail?

Image Brightness Constancy: Assuming Lambertian surfaces, the brightness of corresponding points in stereo images are the same.

Why does stereo fail?

Match Uniqueness: For every point in one stereo image, there is at most one corresponding point in the other image.

Stereo reconstruction pipeline

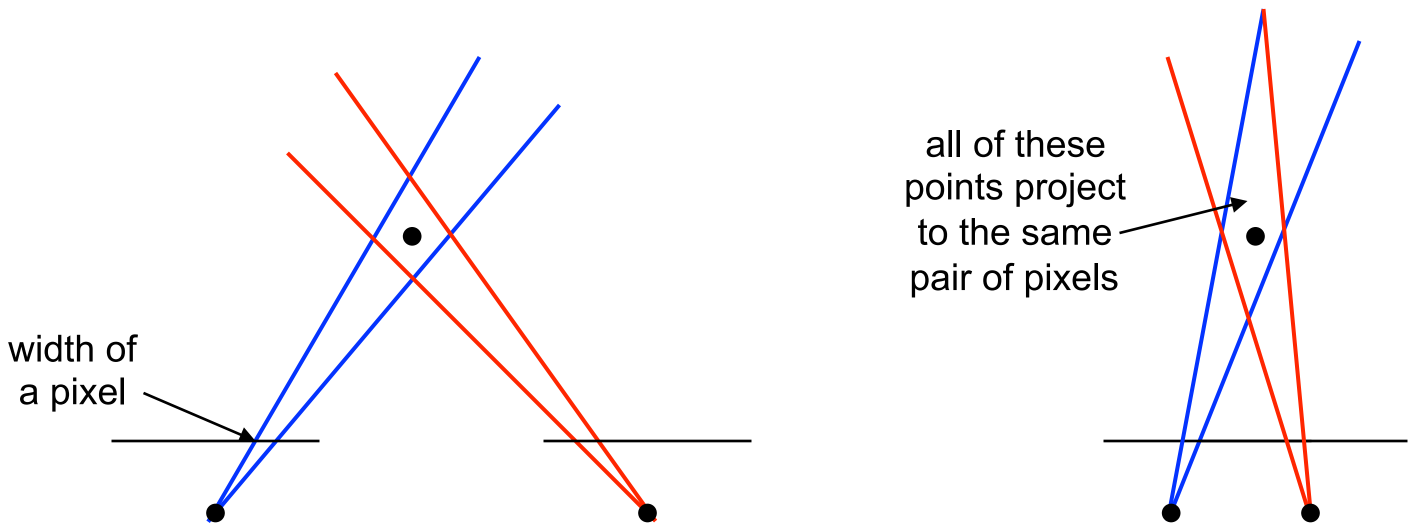
Steps

- Calibrate cameras
- Rectify images
- Compute disparity
- Estimate depth

What will cause errors?

- Camera calibration errors
- Poor image resolution
- Occlusions
- Violations of brightness constancy (specular reflections)
- Large motions
- Low-contrast image regions

Choosing the stereo baseline

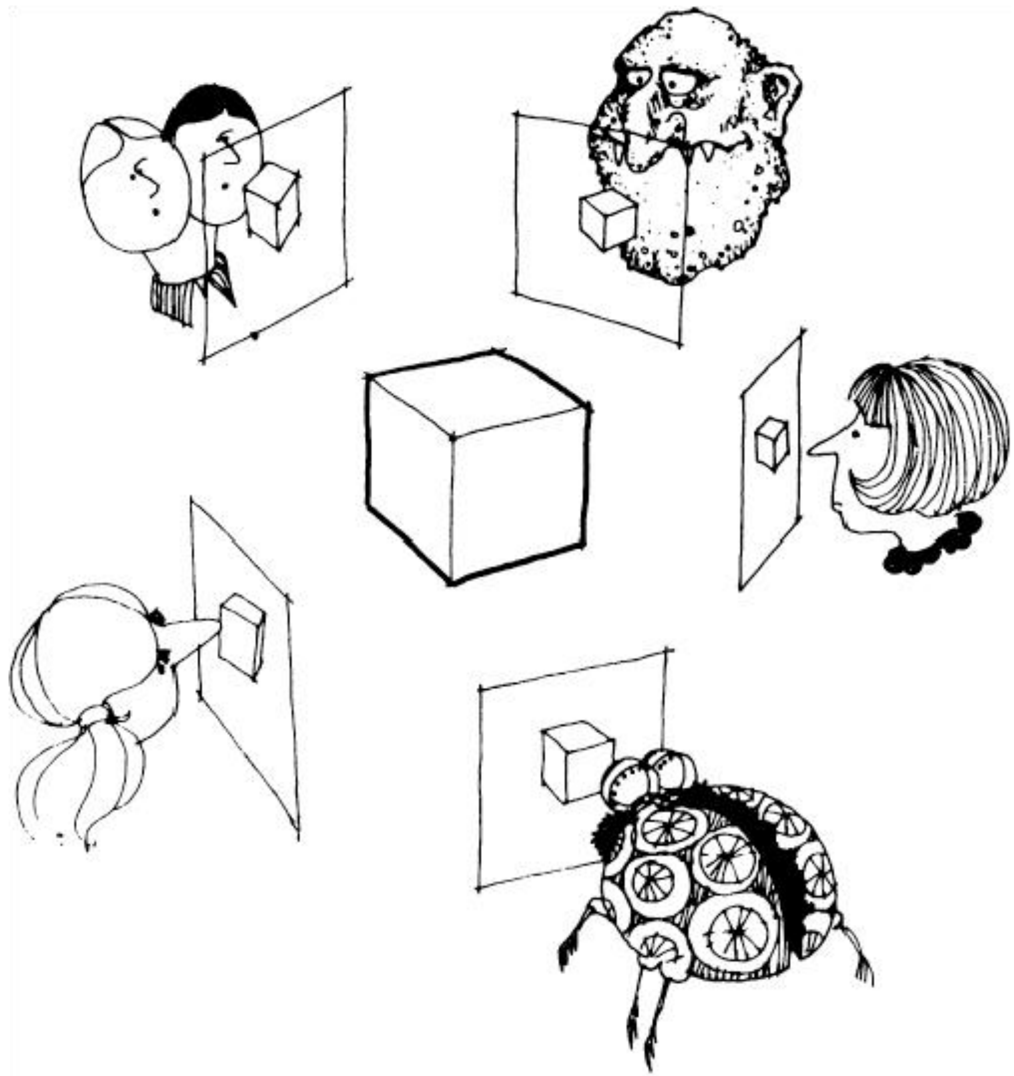


Large Baseline **Small Baseline**

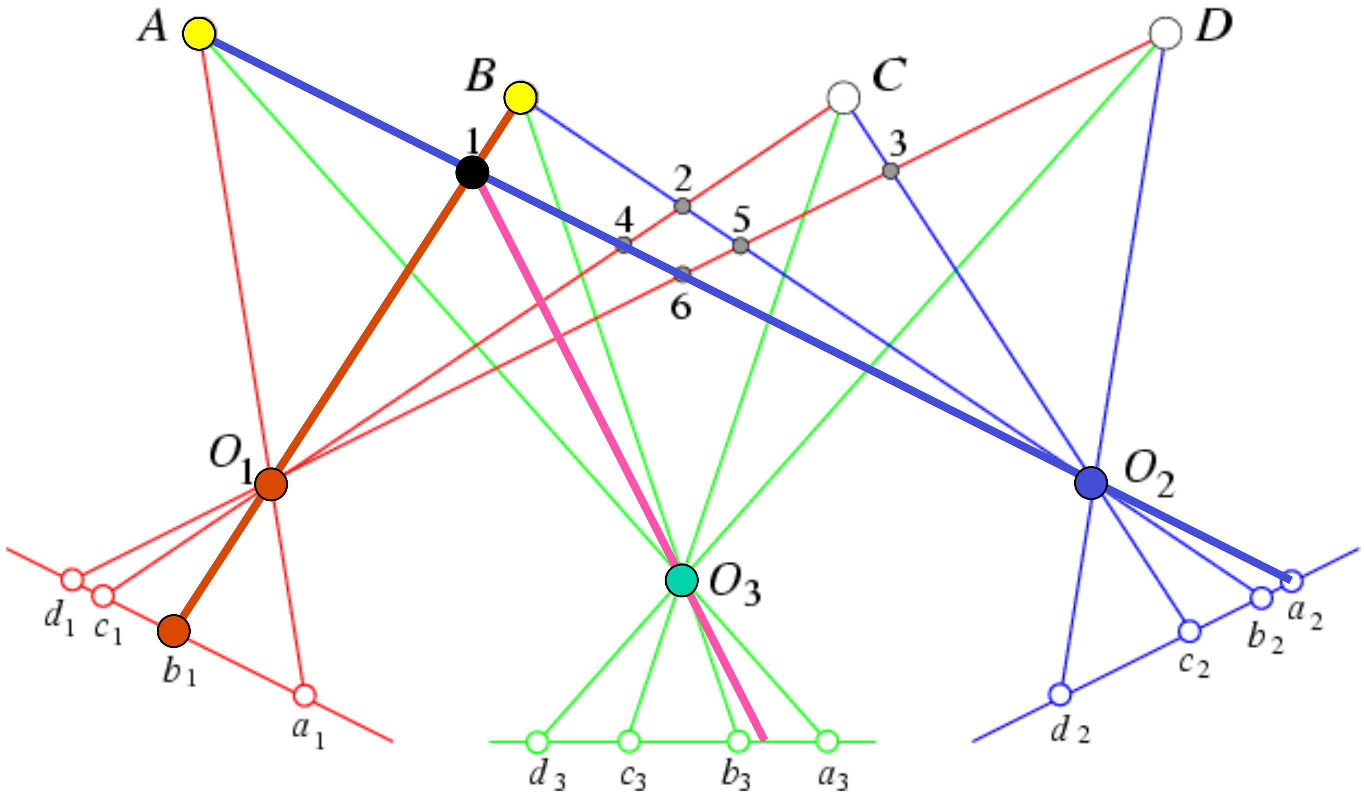
What's the optimal baseline?

- Too small: large depth error
- Too large: difficult search problem

Multi-view stereo ?

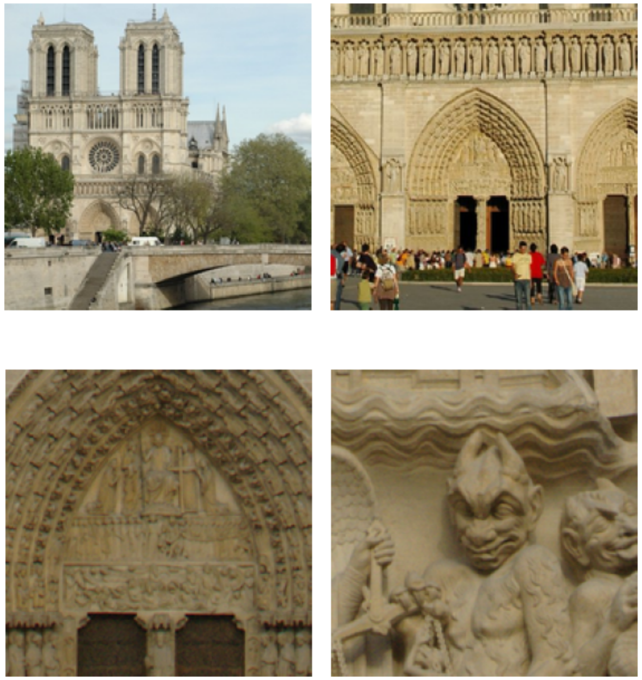


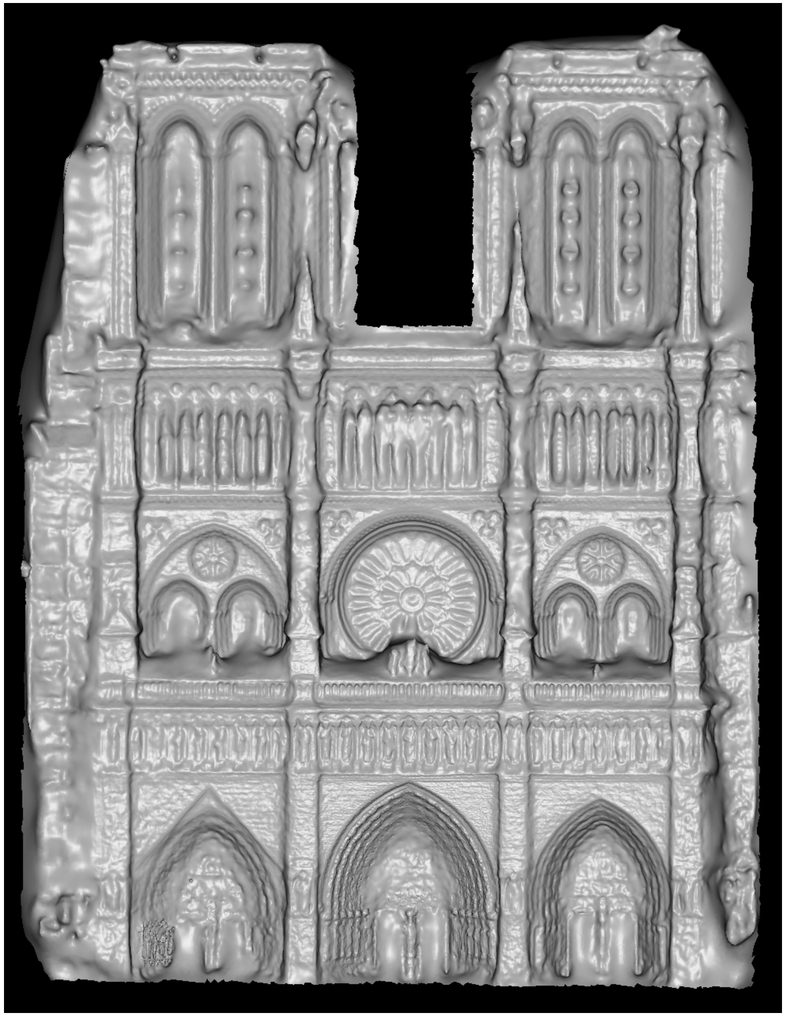
Beyond two-view stereo



The third view can be used for verification

Using more than two images





[Multi-View Stereo for Community Photo Collections](#)
M. Goesele, N. Snavely, B. Curless, H. Hoppe, S. Seitz
Proceedings of [ICCV 2007](#),