Epipolar Geometry and Stereo Vision

3D Vision

University of Illinois

Derek Hoiem

Some figures from Lana Lazebnik, Steve Seitz, Hartley & Zisserman

This class: Two-View Stereo

- Epipolar geometry
  - Relates cameras from two positions
- Stereo depth estimation
  - Recover depth from two images

3D from Stereo Images

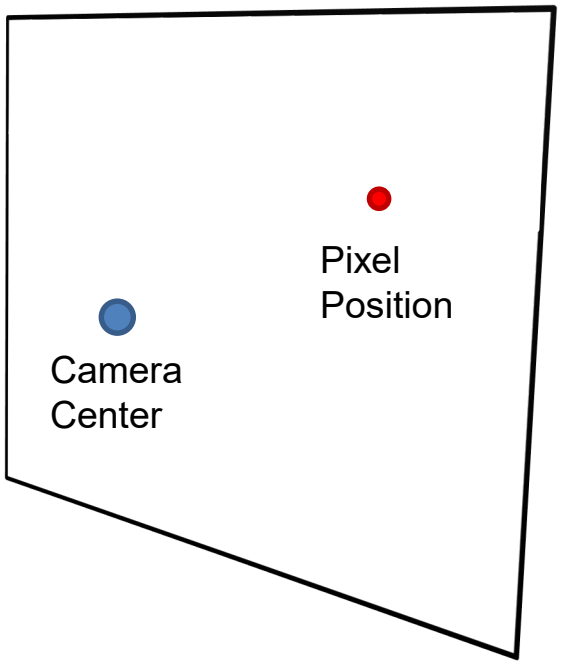
Goal: Produce a depth image from a pair of images from translated cameras

image 1

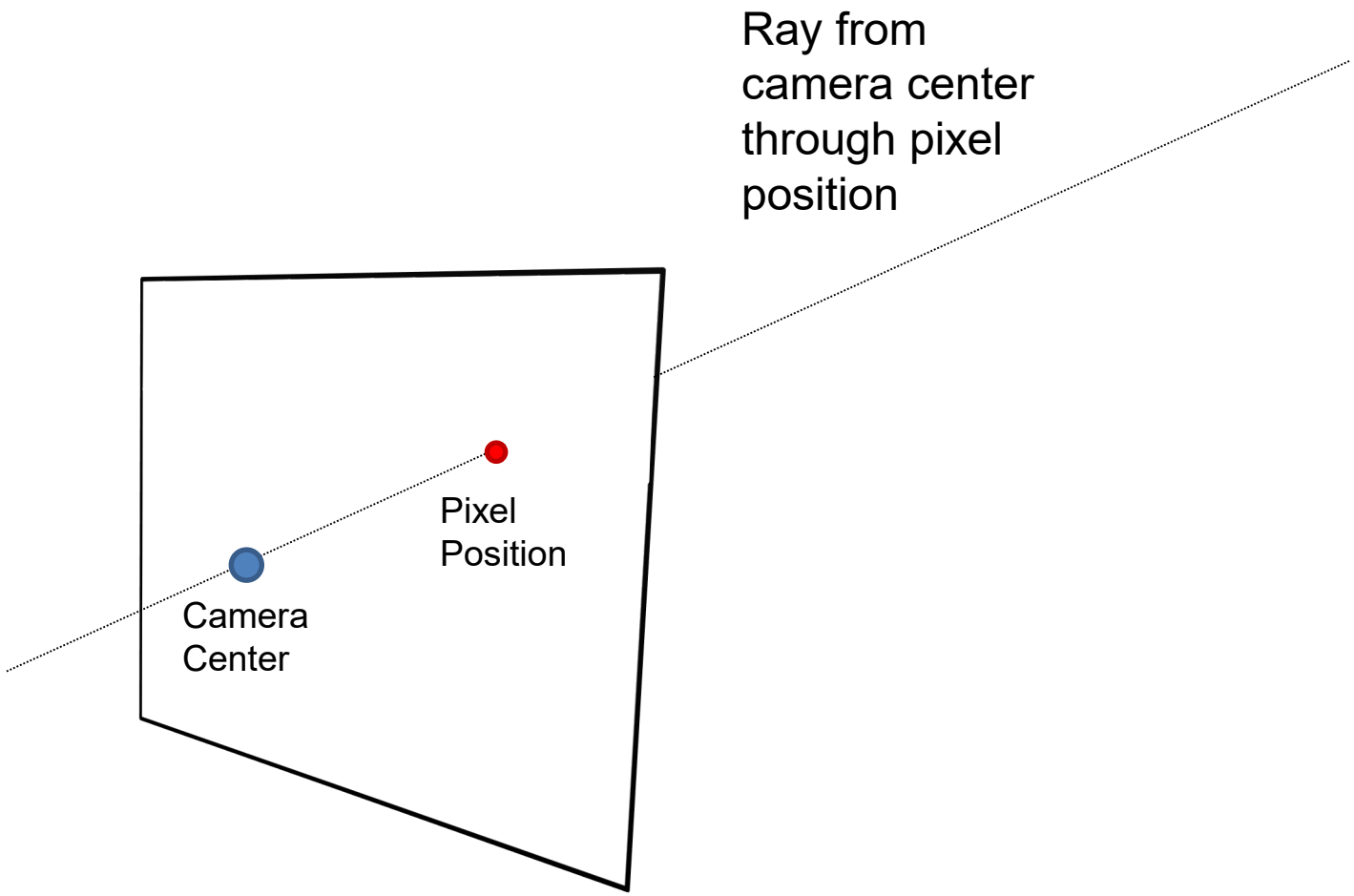
image 2

Dense depth map

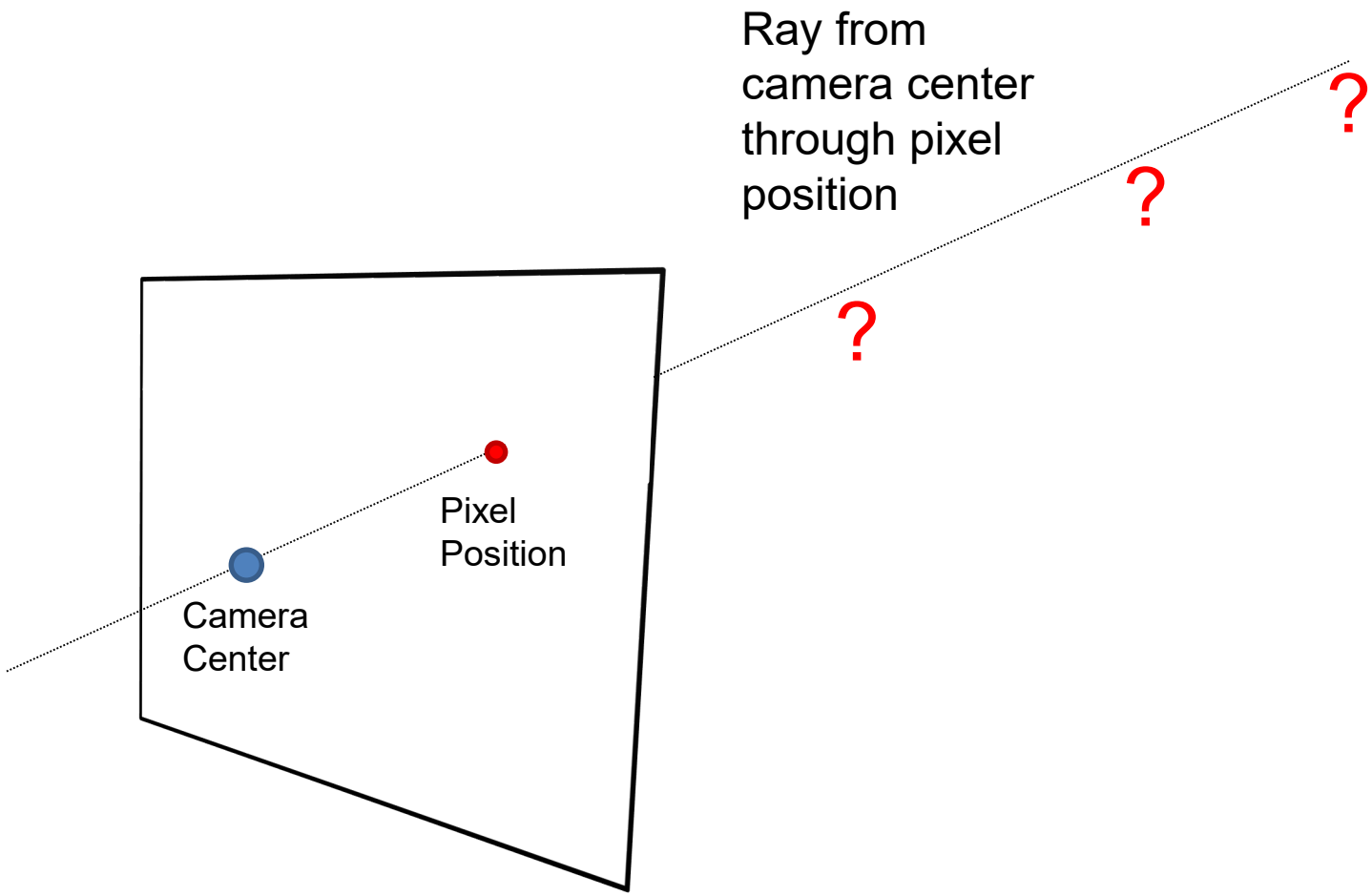
What do we know from a pixel position?



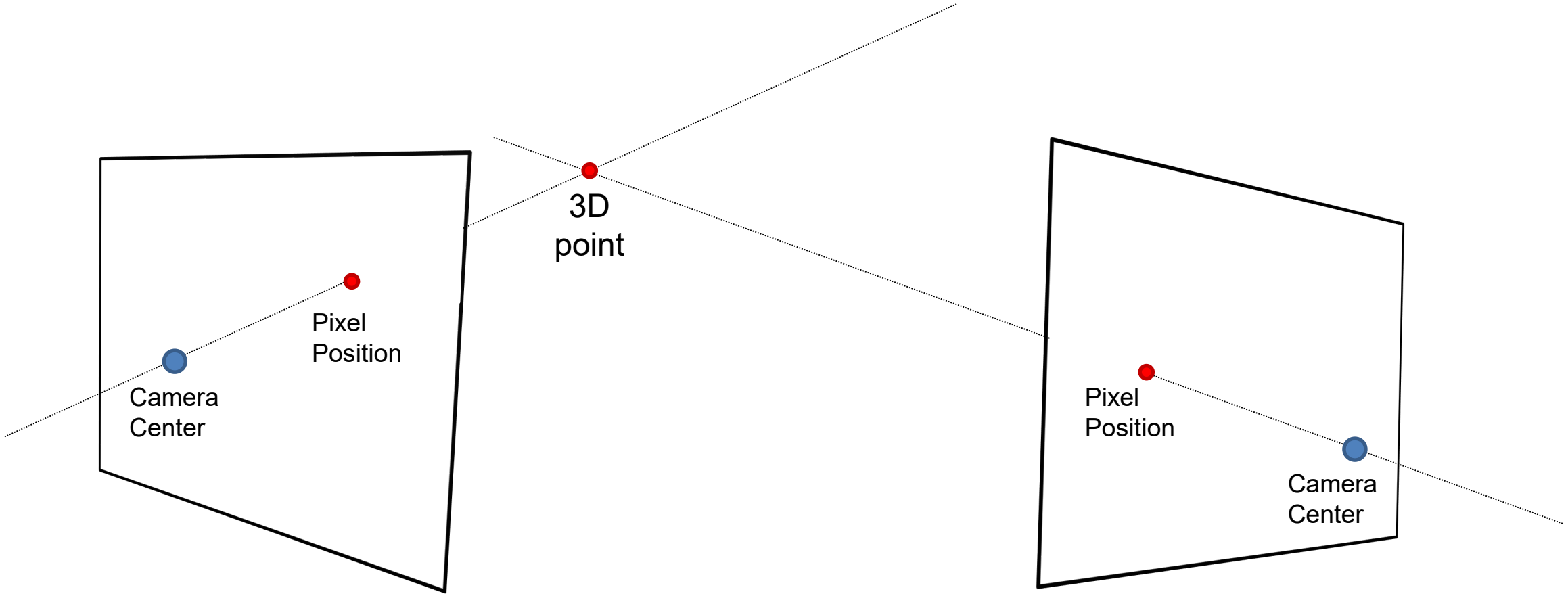
2D Pixel → 3D Ray through camera center and pixel



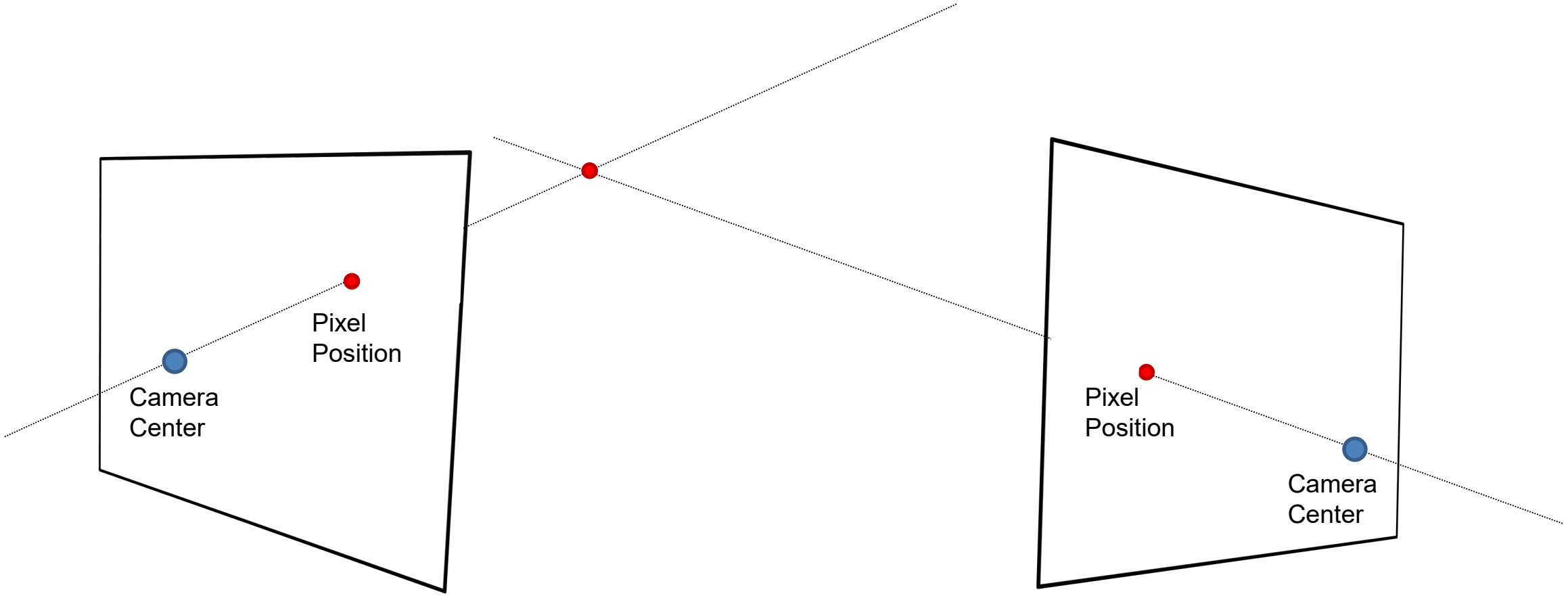
We know the 3D point is along the ray, but cannot determine the depth



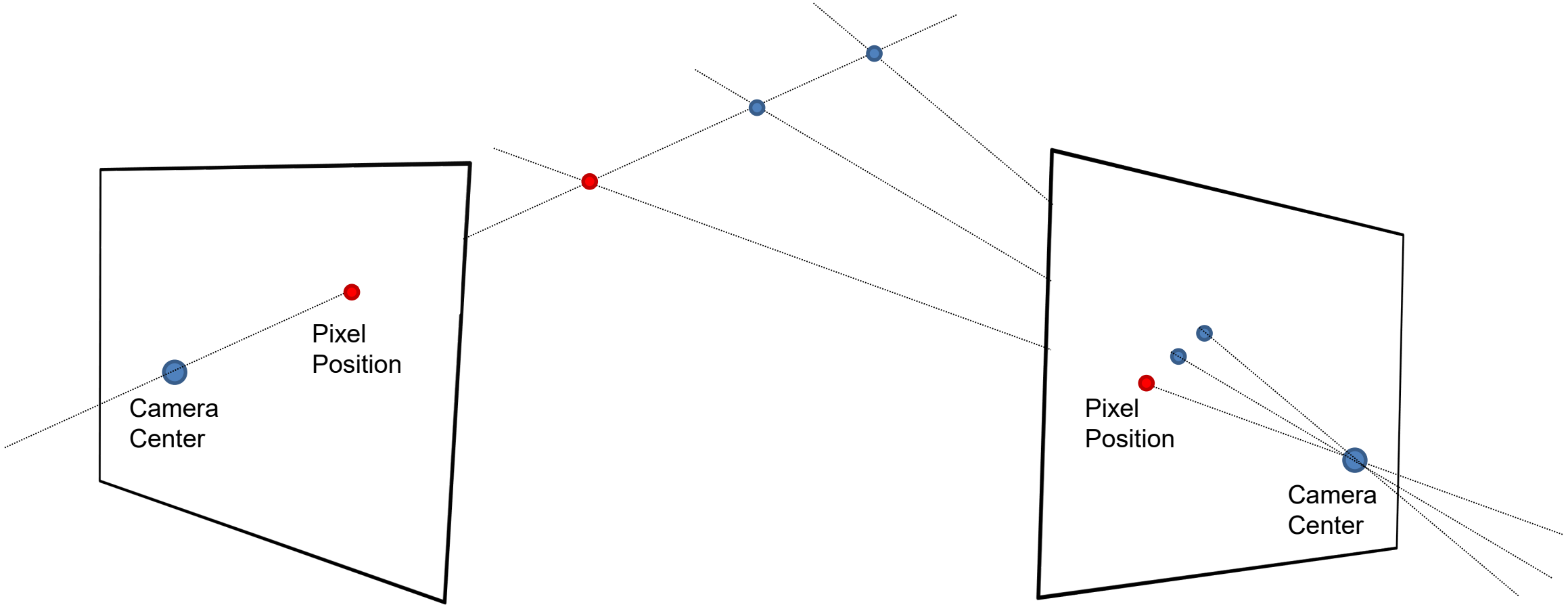
Intersection of two rays determines depth



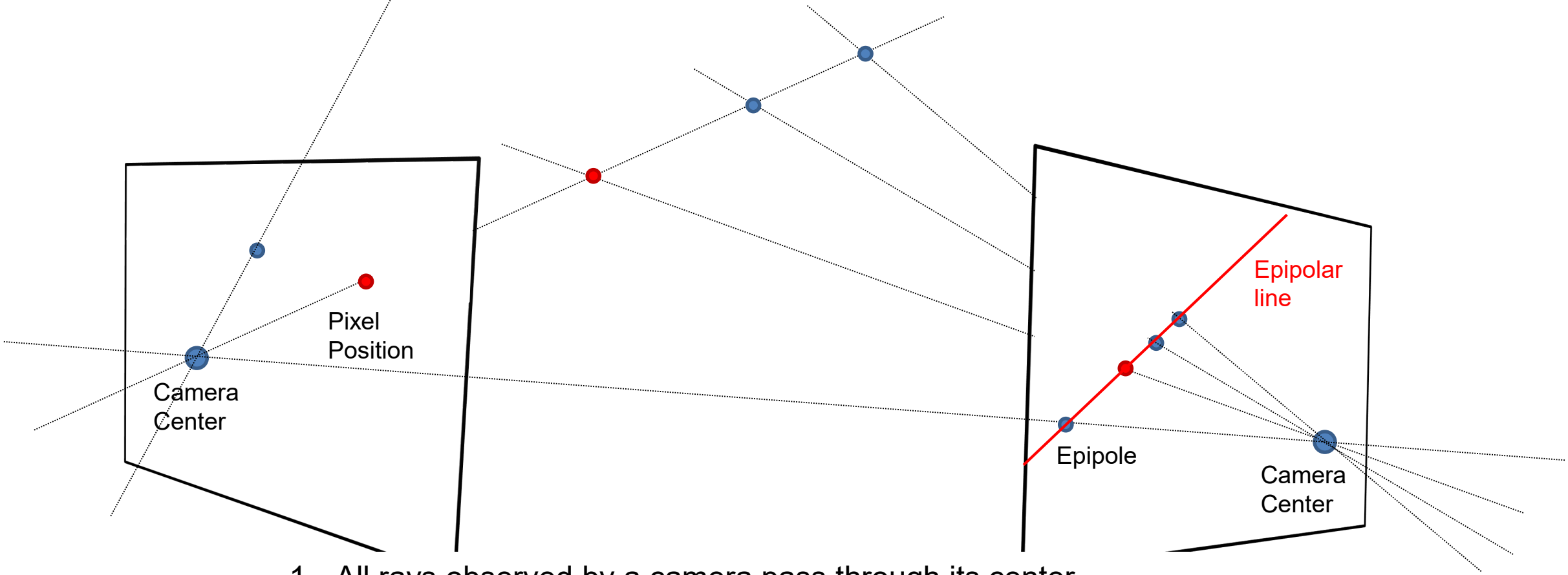
Why do we need two 2D points to get one 3D point?



All points on the ray from first camera project onto line in second camera

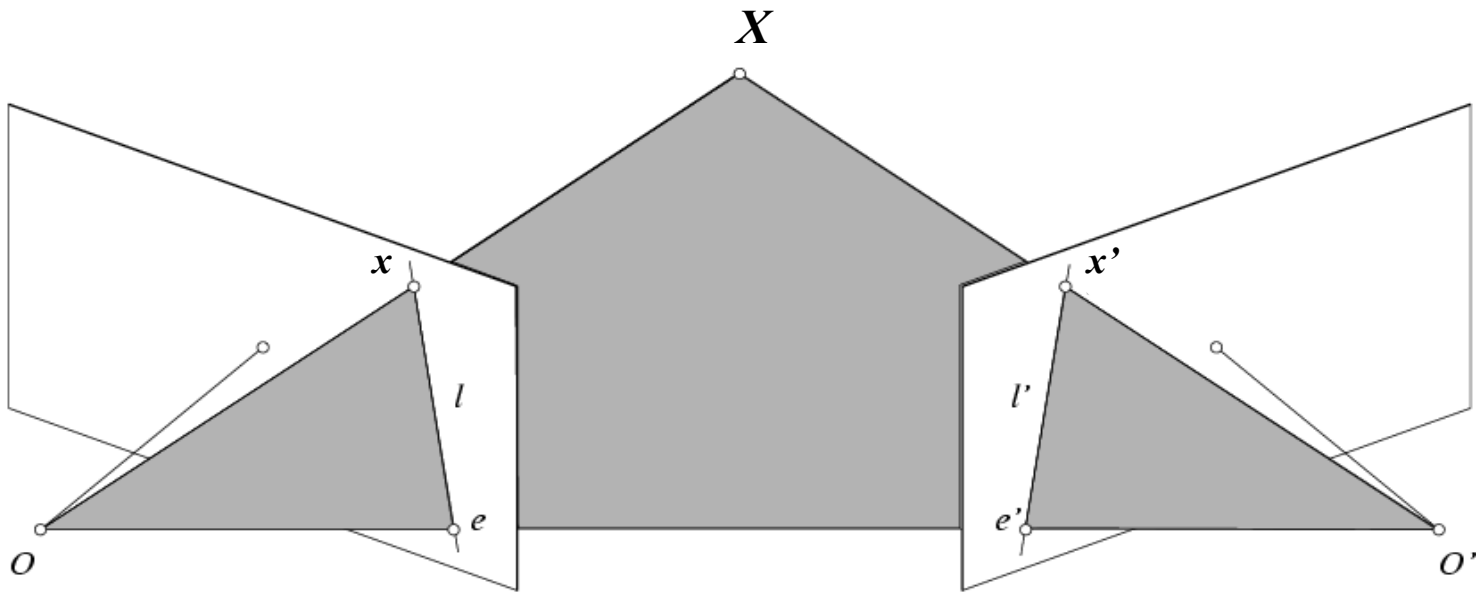


How do we determine this line?



1. All rays observed by a camera pass through its center
2. The center of left projected on right camera is the **epipole**
3. Given the pixel position and epipole from left, we can compute the **epipolar line** in right which must contain the corresponding pixel position

Epipolar constraint: Calibrated case (math version)



Given the intrinsic parameters of the cameras:

1. Convert to normalized coordinates by pre-multiplying all points with the inverse of the calibration matrix; set first camera's coordinate system to world coordinates

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

Homogeneous 2d point (3D ray towards X)

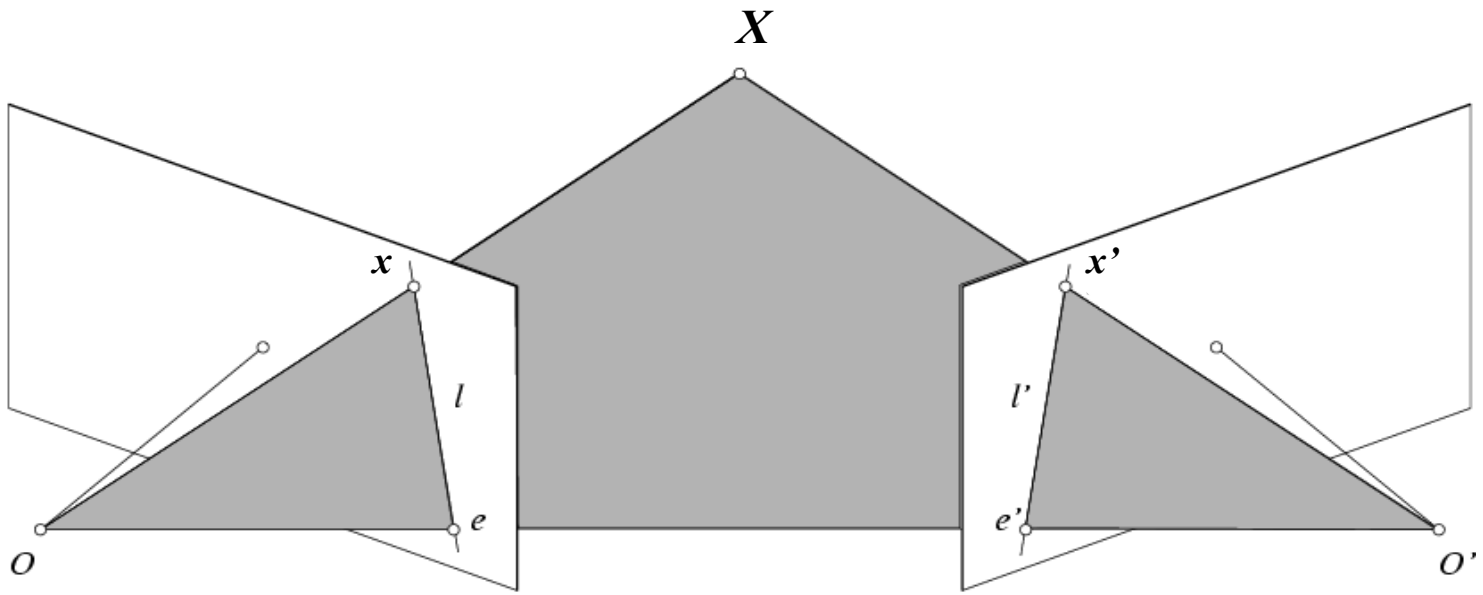
2D pixel coordinate (homogeneous)

3D scene point

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

3D scene point in 2nd camera's 3D coordinates

Epipolar constraint: Calibrated case



Given the intrinsic parameters of the cameras:

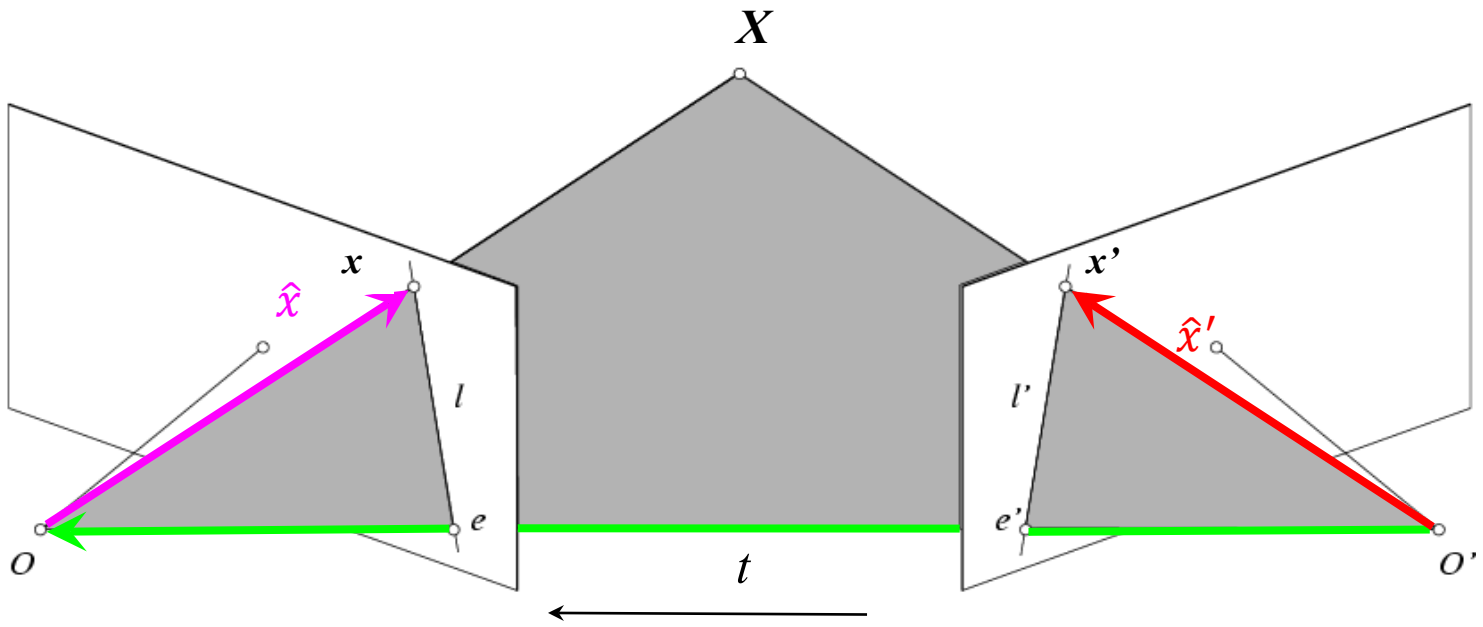
1. Convert to normalized coordinates by pre-multiplying all points with the inverse of the calibration matrix; set first camera's coordinate system to world coordinates
2. Define some *R* and *t* that relate X to X' as below

for some scale factor

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

$$\hat{x} = R\hat{x}' + t$$

Epipolar constraint: Calibrated case

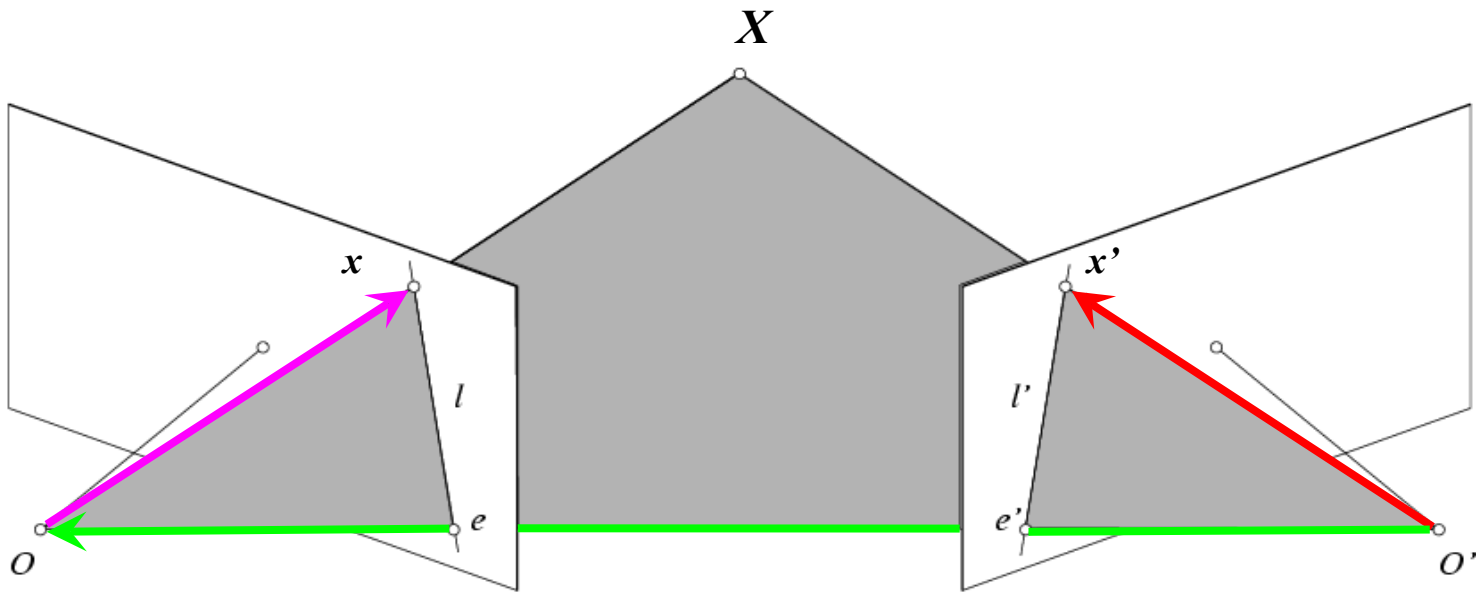


$$\hat{x} = K^{-1}x = X \qquad \hat{x}' = K'^{-1}x' = X'$$

$$\hat{x} = R\hat{x}' + t \quad \Rightarrow \quad \hat{x} \cdot [t \times (R\hat{x}')] = 0$$

(because $\hat{x}, R\hat{x}'$, and t are co-planar)

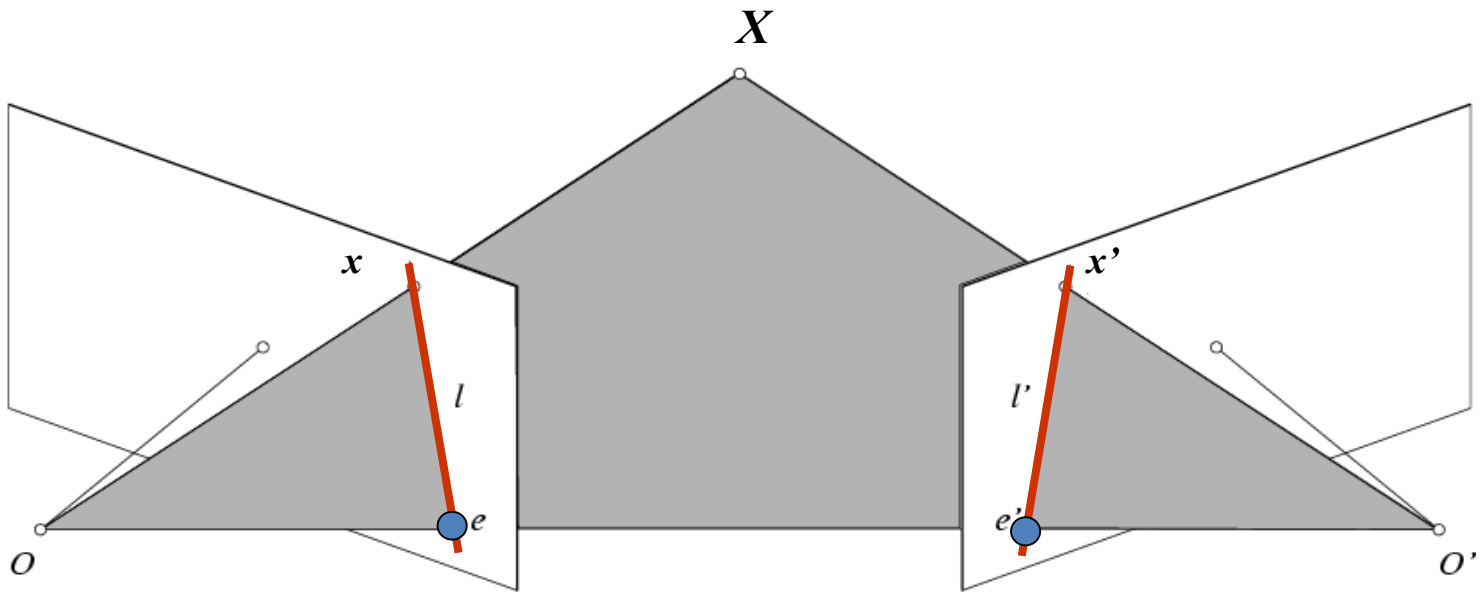
Essential matrix



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

Essential Matrix
(Longuet-Higgins, 1981)

Properties of the Essential matrix



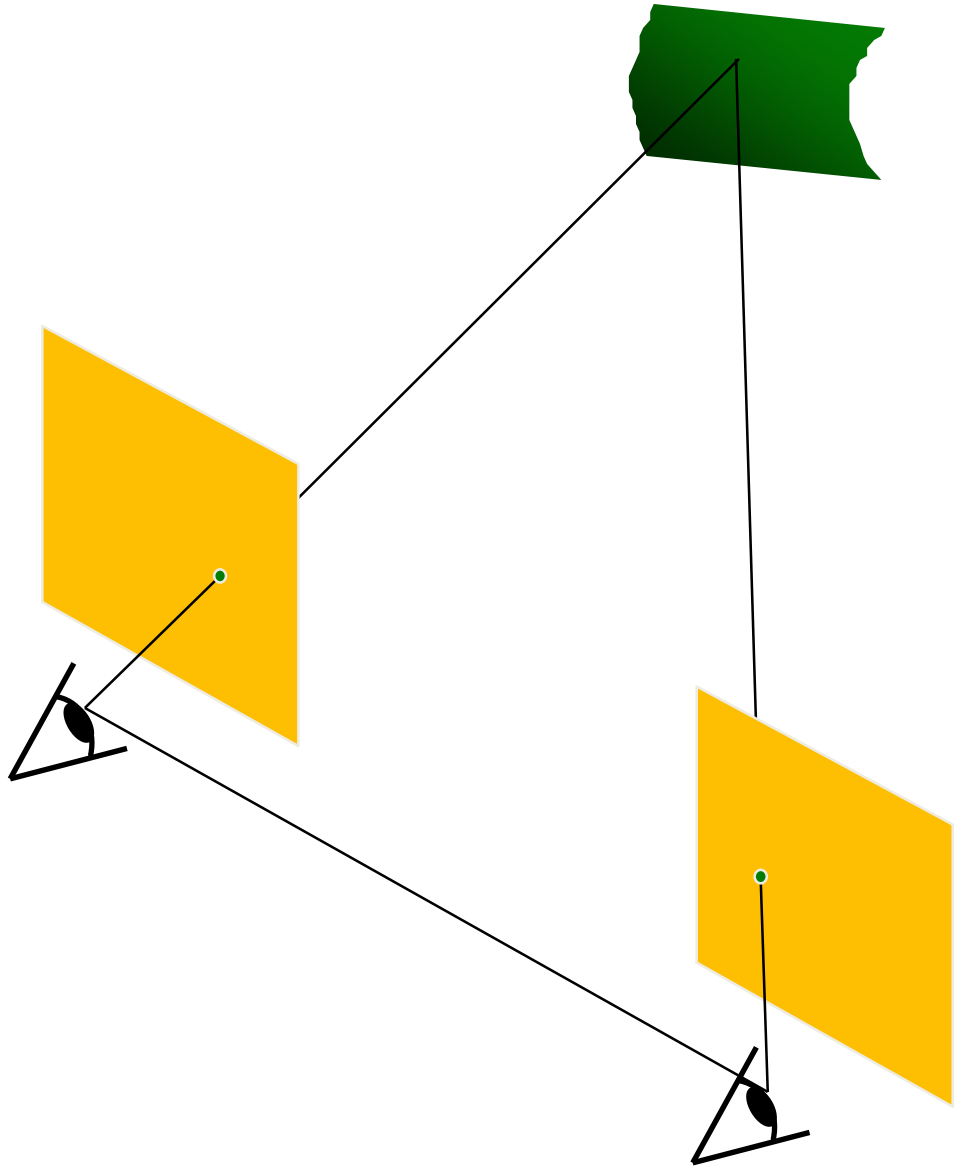
$$\hat{x} \cdot [t \times (R\hat{x}')] = 0 \quad \Rightarrow \quad \hat{x}^T E \hat{x}' = 0 \quad \text{with} \quad E = [t]_\times R$$

Drop ^ below to simplify notation

Skew-symmetric matrix

- $E\,x'$ is the epipolar line associated with x' ($l = E\,x'$)
- $E^T x$ is the epipolar line associated with x ($l' = E^T x$)
- $E\,e' = 0$ and $E^T e = 0$
- E is singular (rank two)
- E has five or six degrees of freedom
  - (3 for R; 2 for t because it's up to a scale, or 3 for t if scale is known)

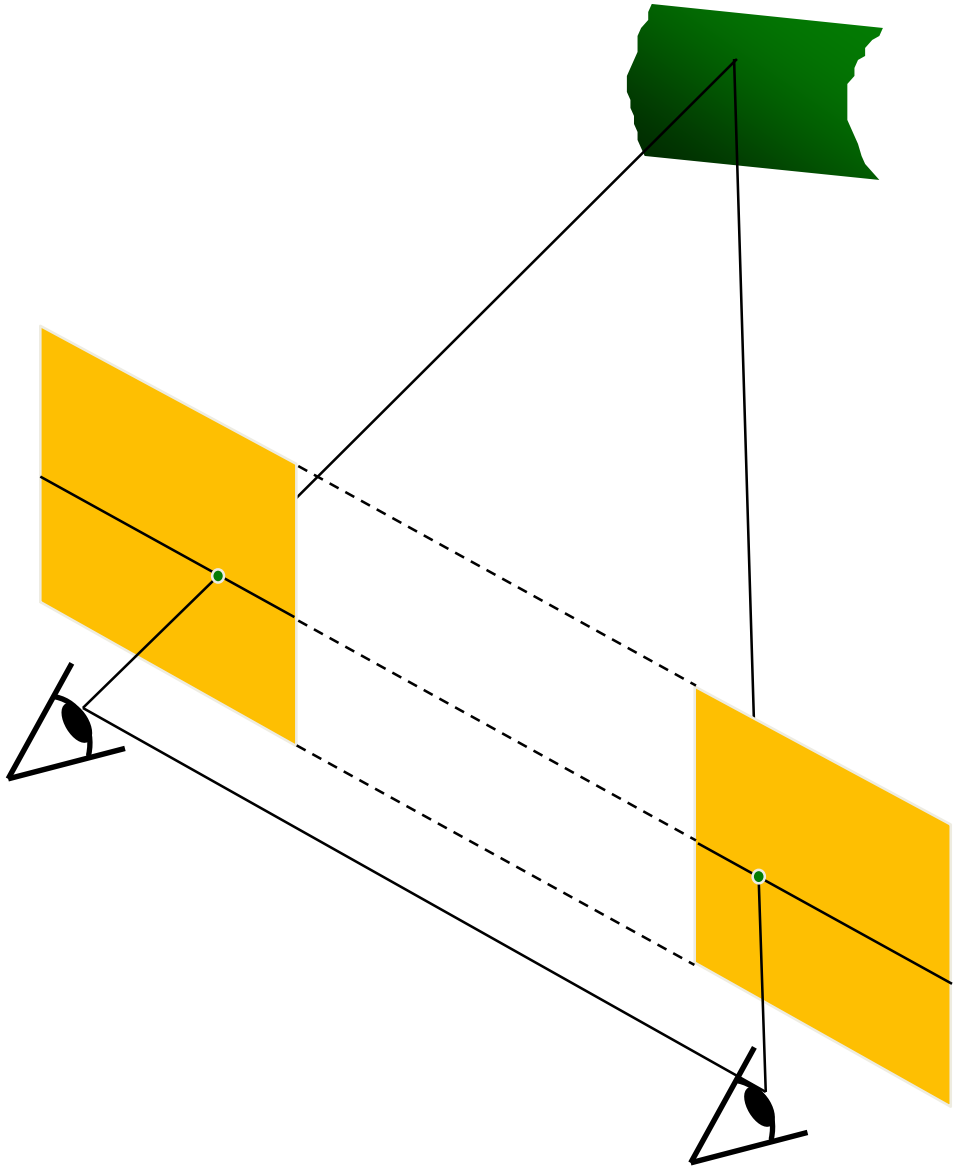
Simplest Case: Parallel images



- Cameras have no relative rotation
- Translation is purely in horizontal direction
- Intrinsic parameters are the same

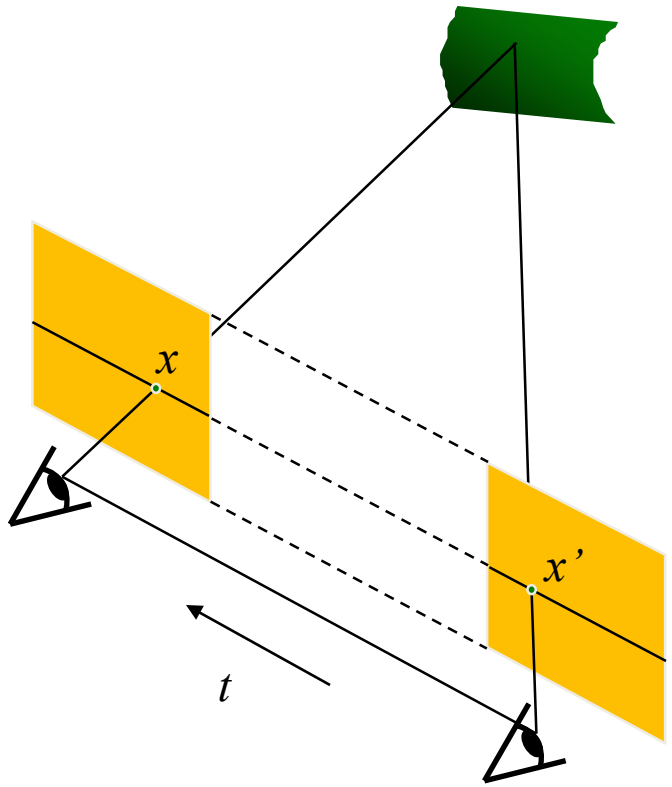
When this is only approximately true, the images can be “rectified” to make it true, assuming they are calibrated

Simplest Case: Parallel images



- Cameras have no relative rotation
- Translation is purely in horizontal direction
- Intrinsic parameters are the same
- Then, epipolar lines fall along the horizontal scan lines of the images

Simplest Case: Parallel images



Epipolar constraint:

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

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

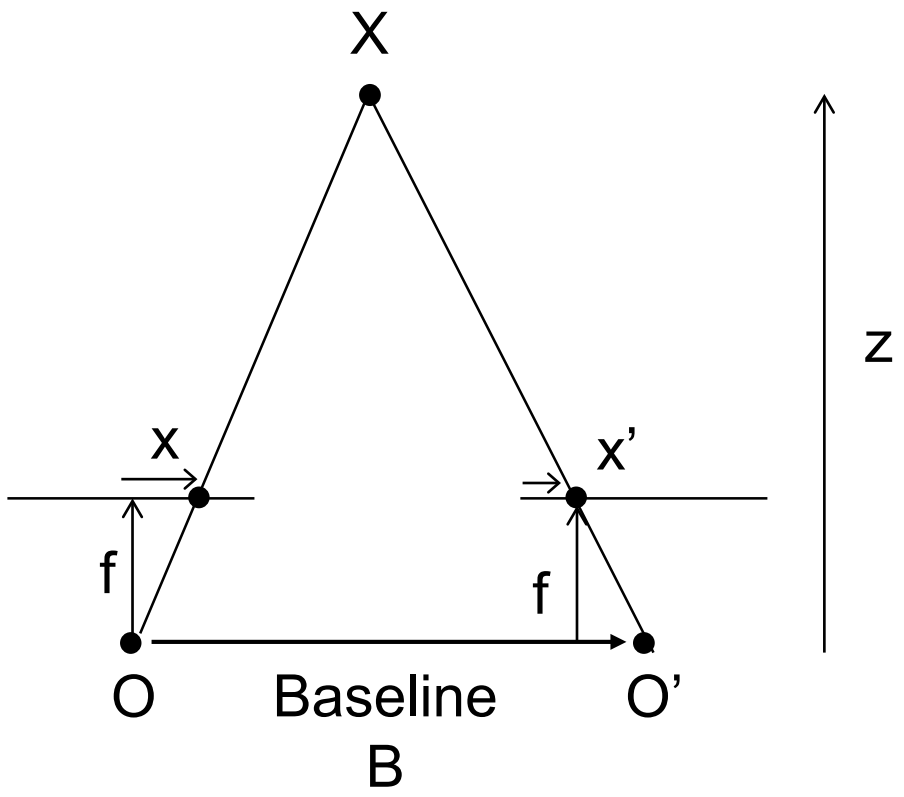
$$E = t \times R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix}$$

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

The v-coordinates (rows) of corresponding points are the same

Depth from disparity

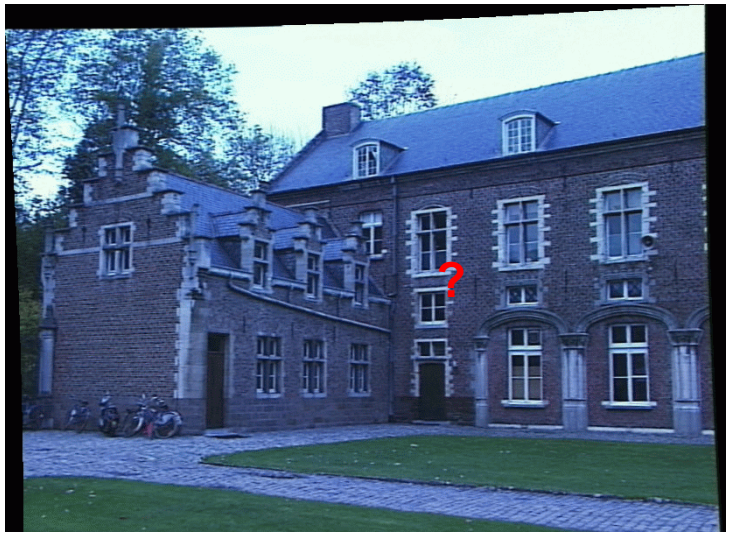
$$\frac{x - x'}{O - O'} = \frac{f}{z}$$



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

Disparity is inversely proportional to depth.

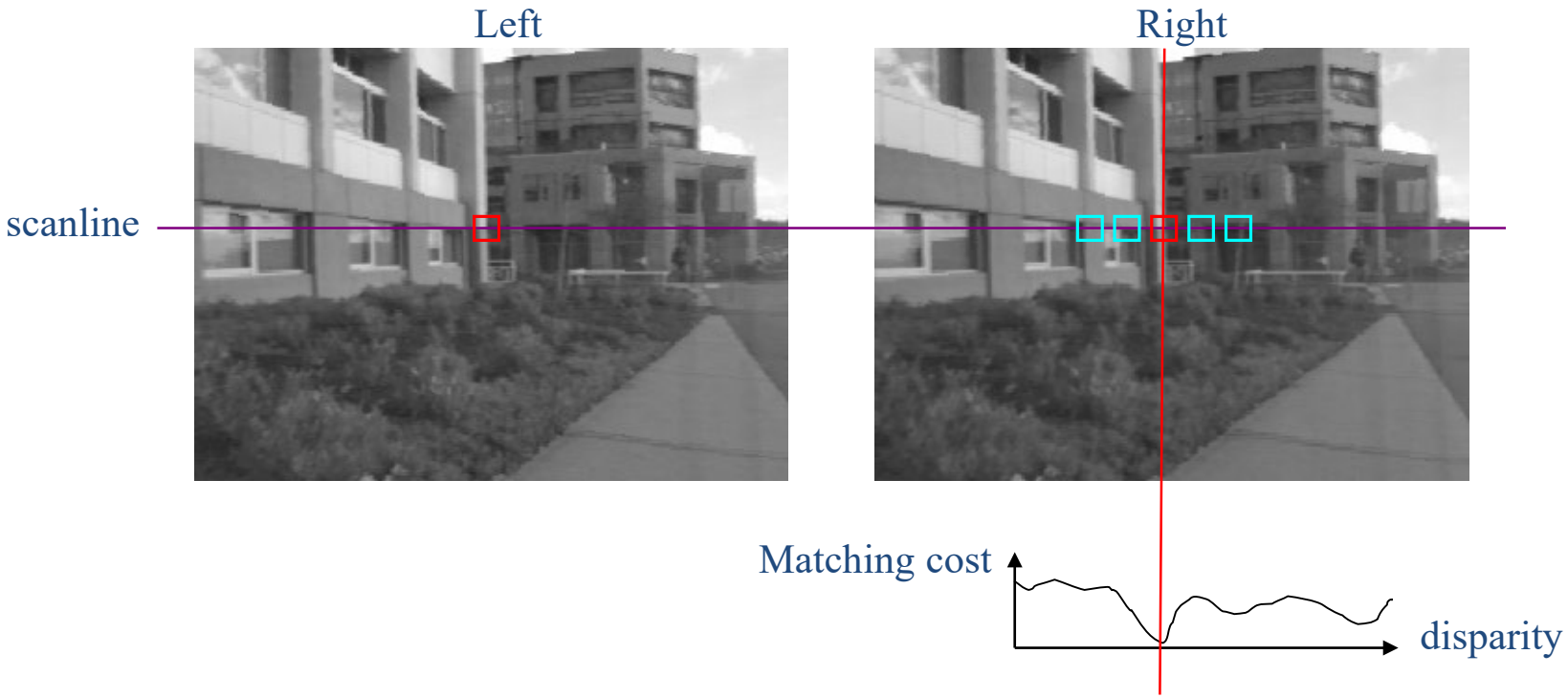
Correspondence Problem



- We have two images taken from calibrated cameras with different positions

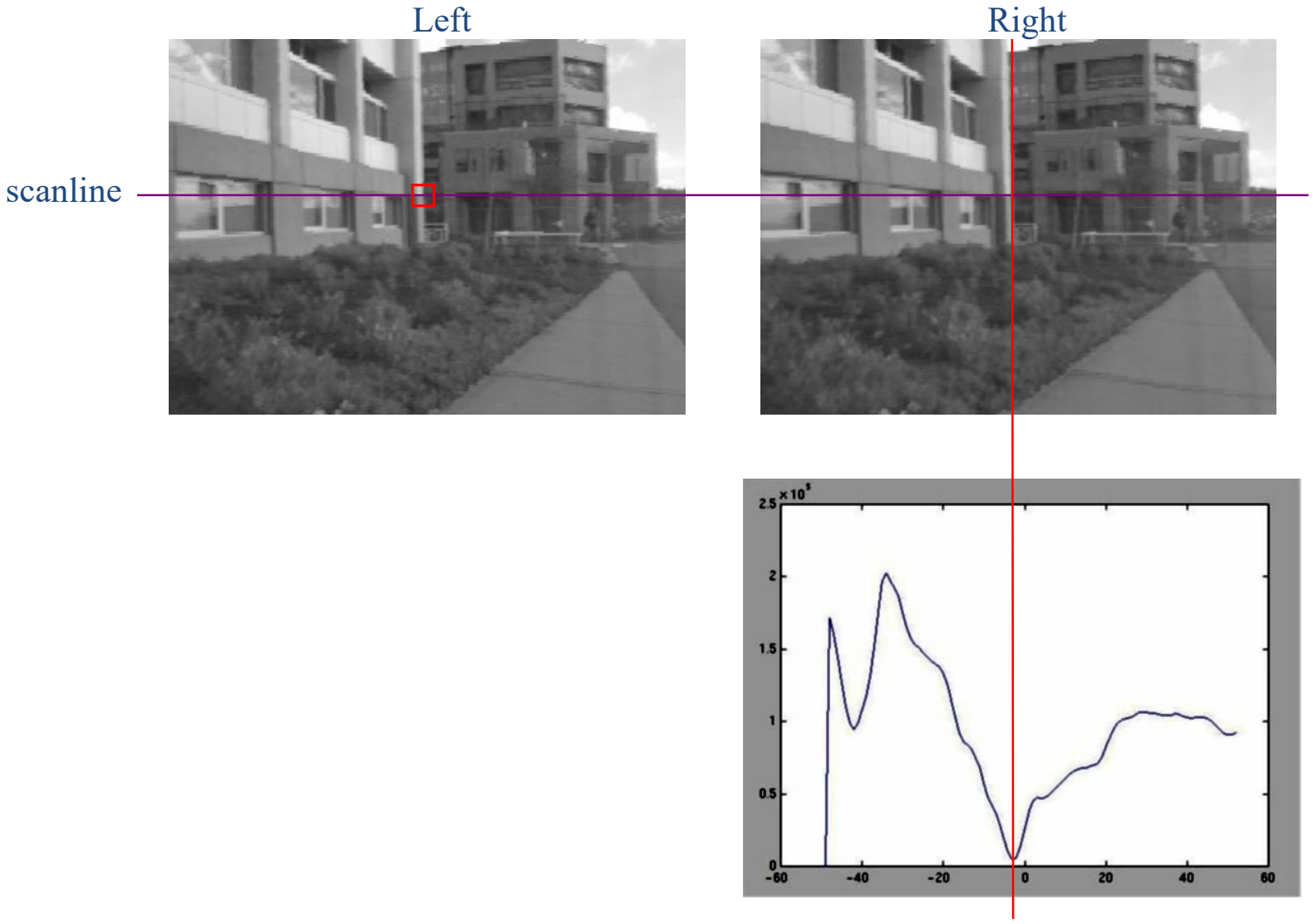
- How do we match a point in the first image to a point in the second? How can we constrain our search?

Correspondence search

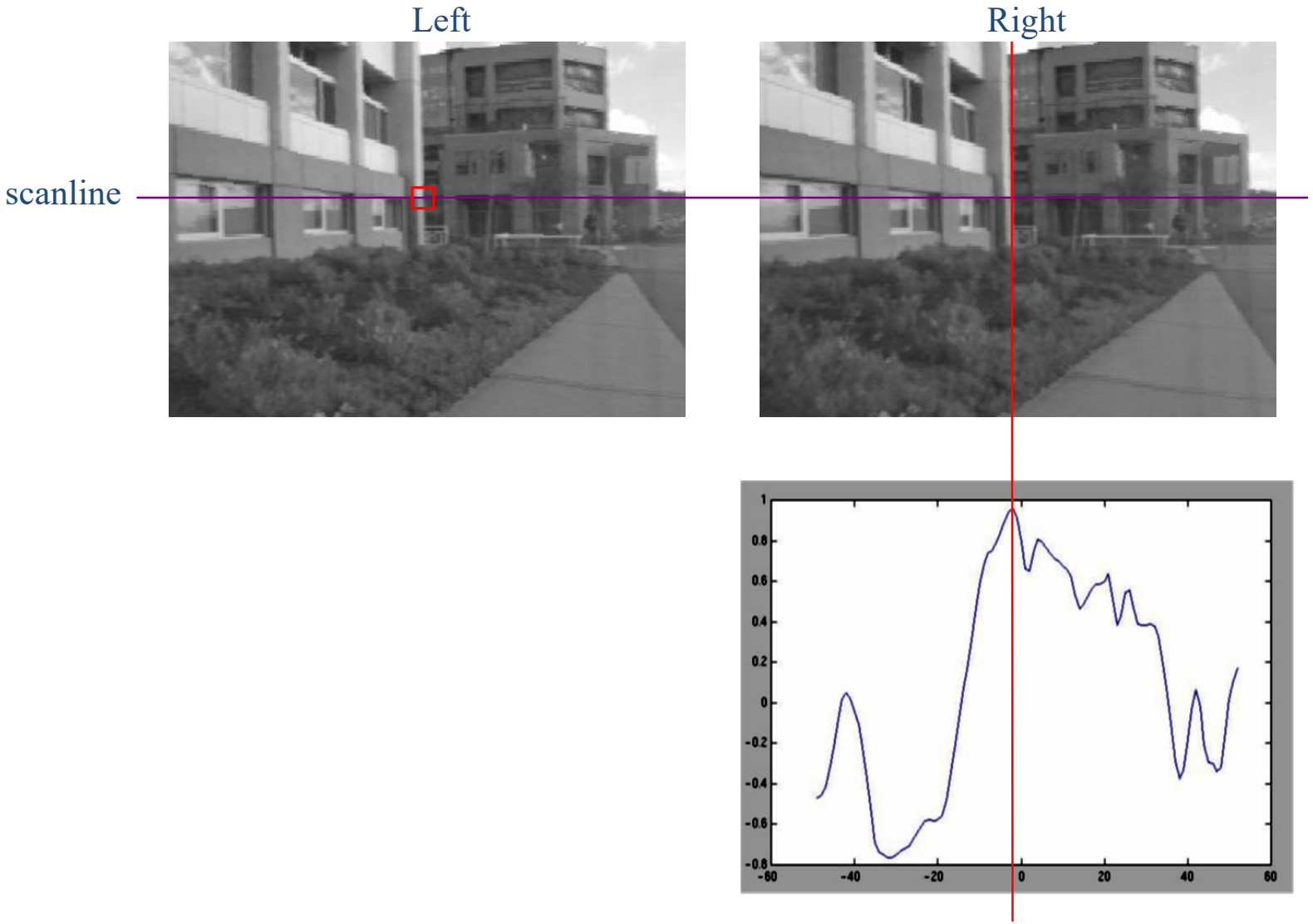


- Slide a window along the right scanline and compare contents of that window with the reference window in the left image
- Matching cost
  - SSD: sum squared distance
  - SAD: sum absolute distance
  - NCC: normalized cross correlation
  - SSIM: structural similarity index measure (compromise of SSD and NCC)

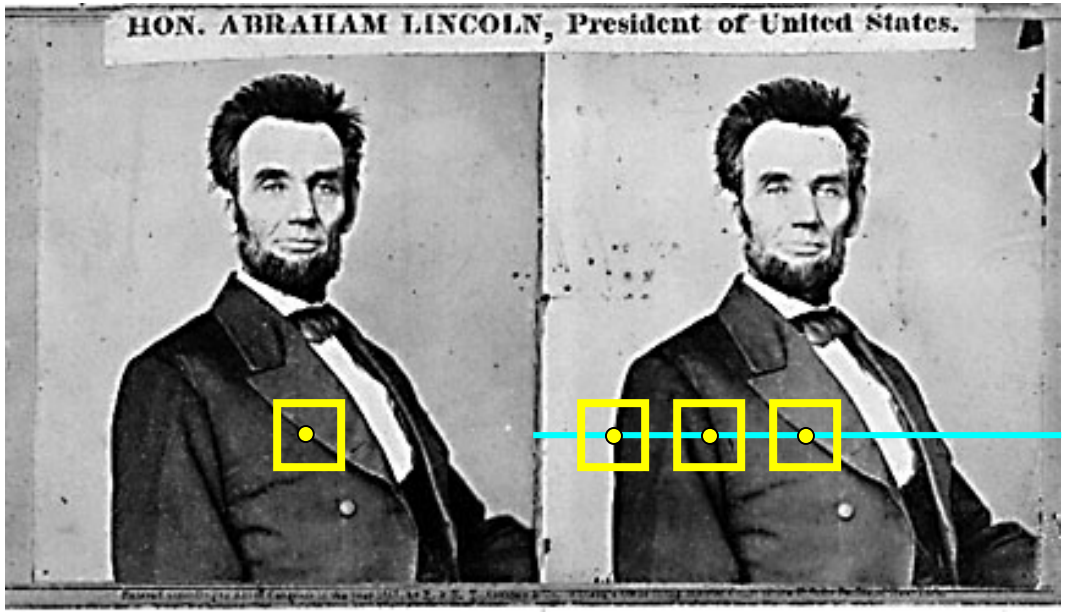
Correspondence search



Correspondence search



Basic stereo matching algorithm

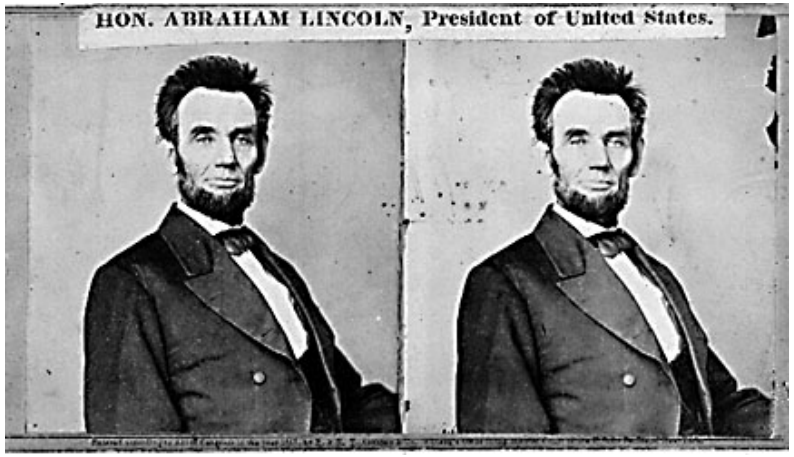


- If necessary, rectify the two stereo images to transform epipolar lines into scanlines
- For each pixel x in the first image
  - Find corresponding epipolar scanline in the right image
  - Search the scanline and pick the best match x’
  - Compute disparity x-x’ and set depth(x) = fB/(x-x’)

What are the assumptions in detecting correspondences?

- Surfaces have constant appearance across viewpoints
  - Lambertian material: non-reflective, non-transparent
  - Same lighting and camera gain, or measures like NCC that factor out mean and variance within patch
- Appearance is distinctive enough to identify correspondences
  - Textured surfaces
- Depth is uniform within the patch
  - Can extend to locally planar if surface normal also estimated
  - Violated at object edges and non-frontal or non-smooth surfaces
- Depth in nearby pixels is similar or co-planar (for global optimization)

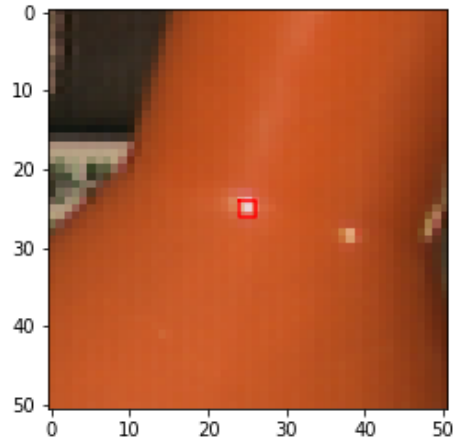
Failures of correspondence search



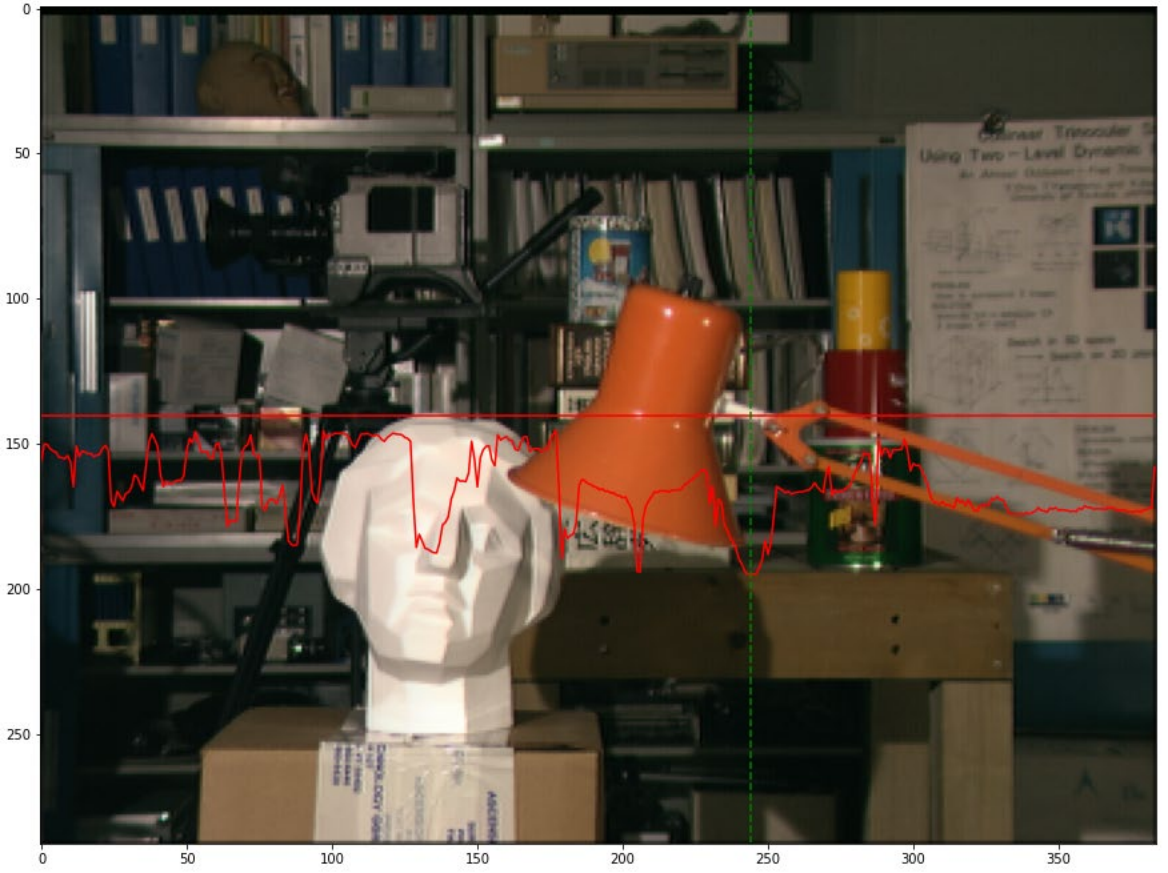
Textureless surfaces

Occlusions, repetition

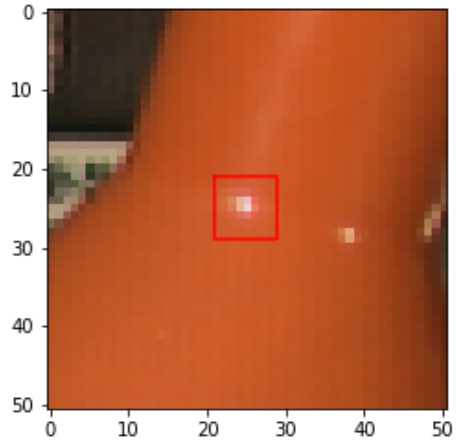
Non-Lambertian surfaces, specularities



Reflective Surface

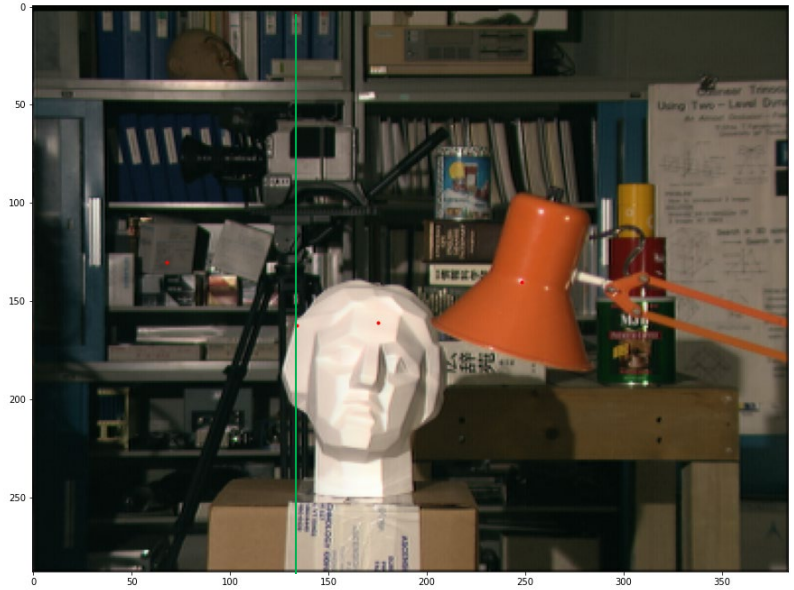


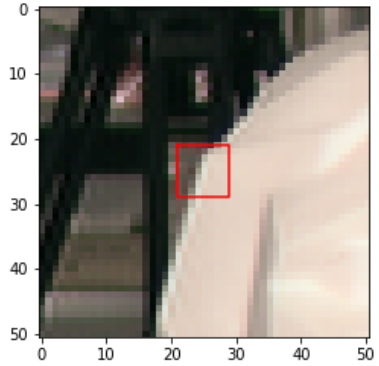
Patch Size = 1 pixel



Reflective Surface

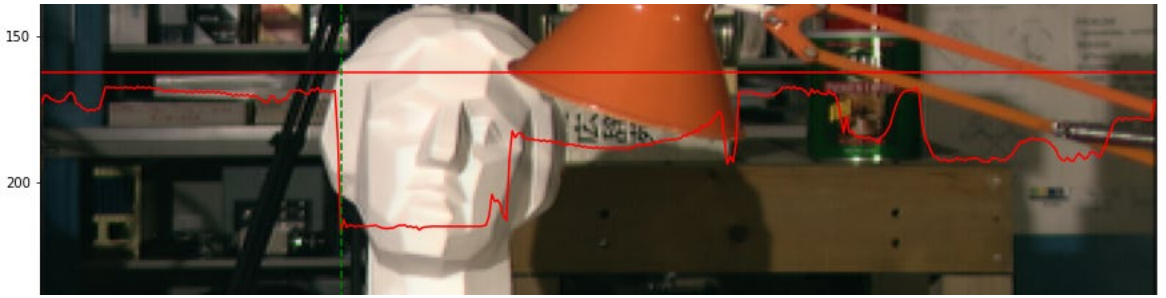
Patch Size = 7 pixels



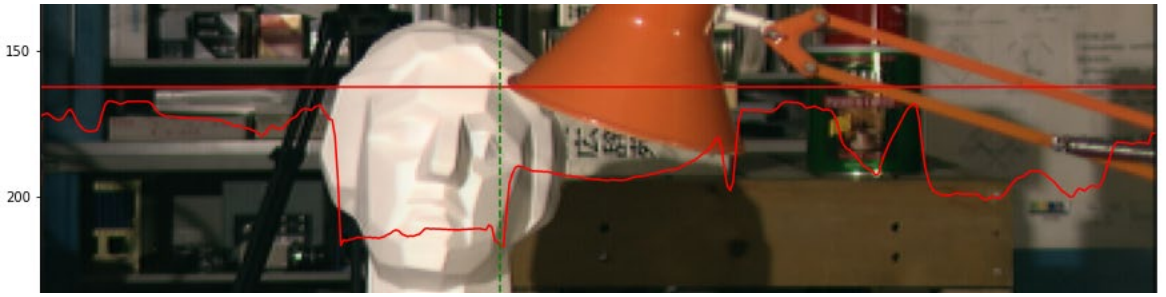


Edge

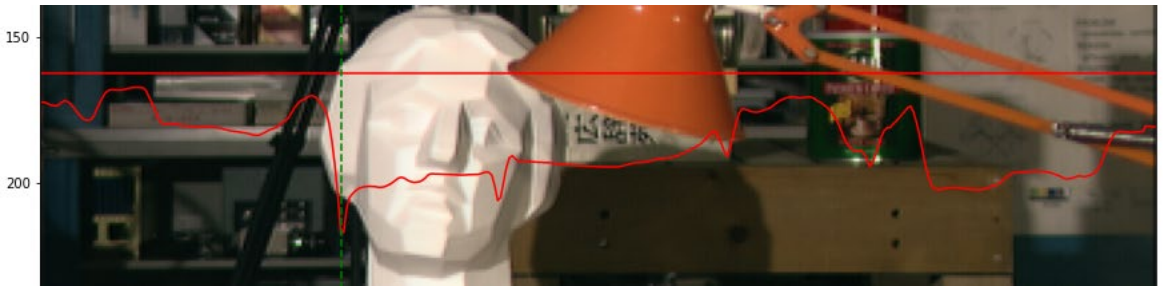
Patch Size 1

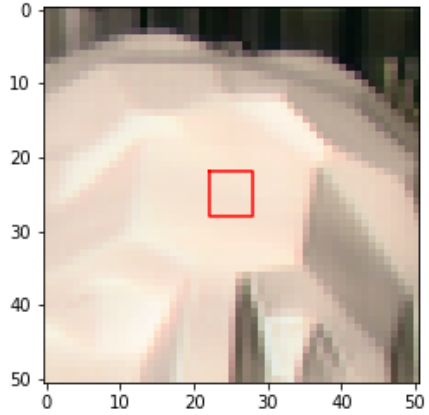


Patch Size 3



Patch Size 7

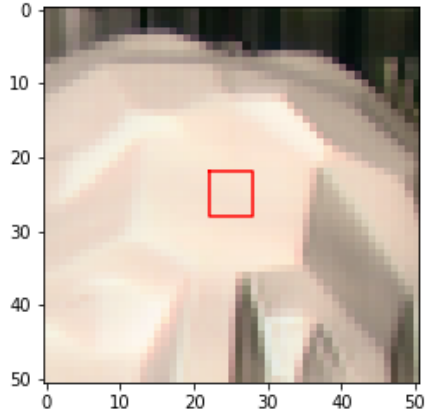




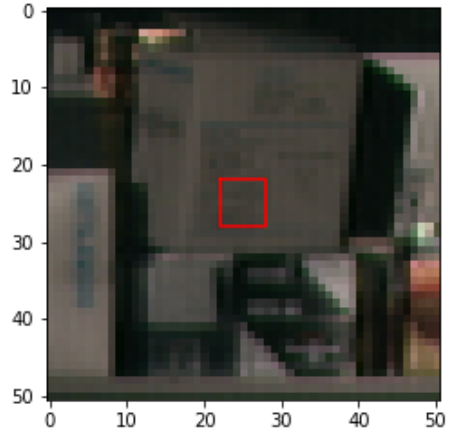
Smooth Surface

Patch Size = 1 pixels

Smooth Surface

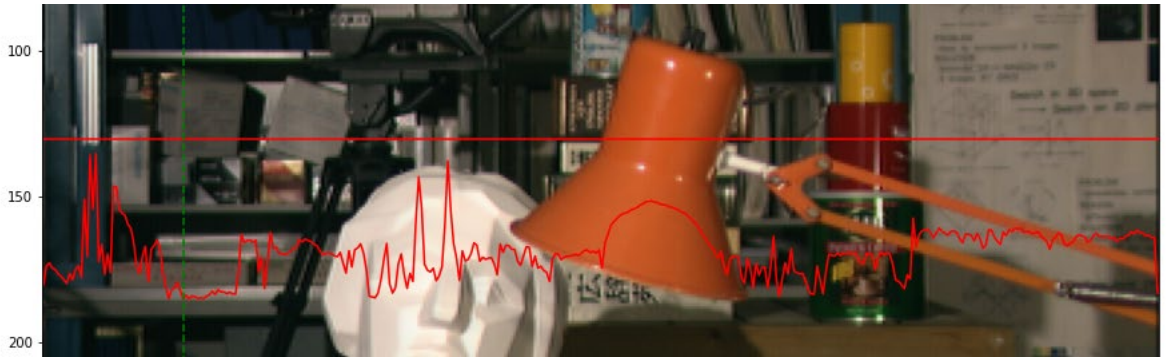


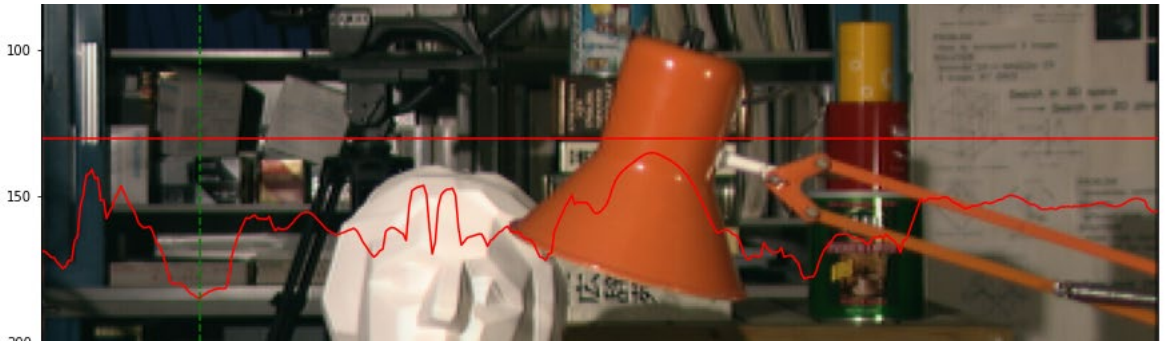
Patch Size = 7 pixels



Textured Surface

Patch Size = 1 pixel



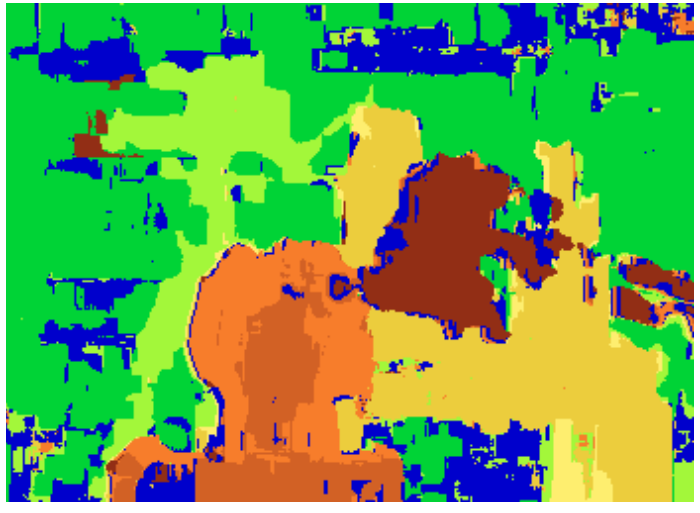


Patch Size = 7 pixels

Results with window search

Data

Window-based matching

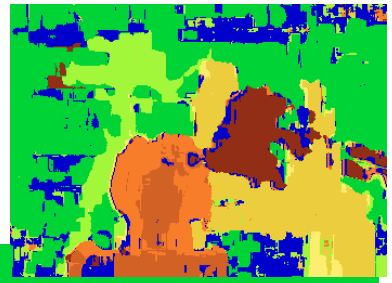


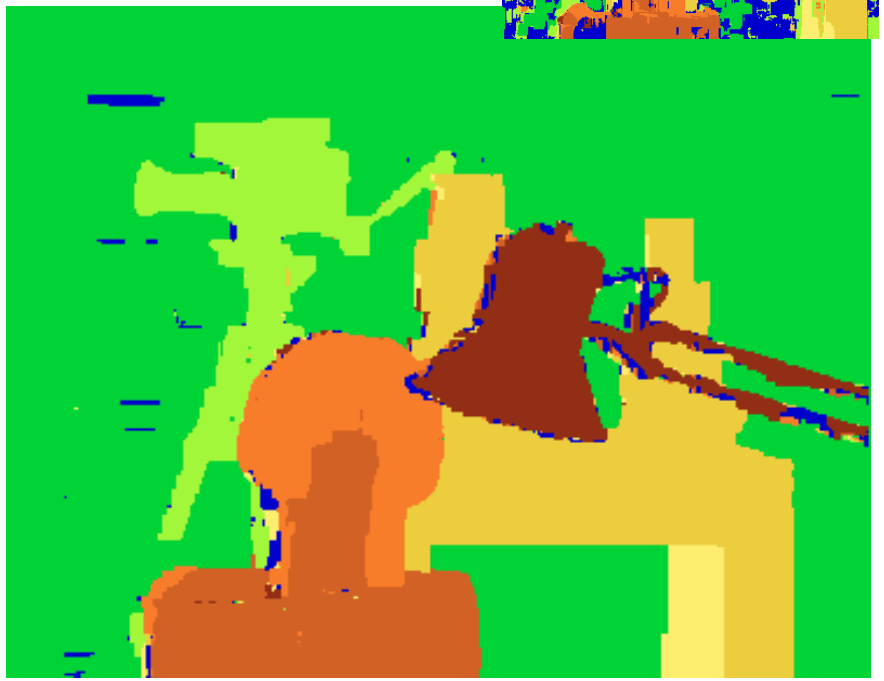
Ground truth

2 min break

Add constraints and solve with graph cuts

Before





Graph cuts

Ground truth

*

V. Kolmogorov and R. Zabih, Computing Visual Correspondence with Occlusions via Graph Cuts, ICCV 2001

For the latest and greatest: http://www.middlebury.edu/stereo/

Graph cut optimization

- Minimize $E(f) = E_{data}(f) + E_{occ}(f) + E_{smooth}(f)$
  - Data: disparity per pixel should have low photometric cost
  - Occlusion: cost for not assigning a depth value
  - Smooth: cost for assigning neighboring pixels to different depths

- Optimization is over “cost volume”, i.e. there is a unary cost for each disparity for each pixel

- Use graph cuts with label expansion to get good solution
  - Iteratively assign all labels to a particular disparity that improve the global cost
  - Optimal assignment within each step but may be globally suboptimal
  - Relatively slow

Taxonomy of solutions and designs

- Cost: SSD, NCC, SAD, …
- Aggregation: pixel, fixed window, adaptive window, …
- Optimization: winner-take-all, graph cuts, dynamic program
- Priors: smoothness (improve speed/quality), piecewise-planar
- Refinement: parabolic fit, Lucas-Kanade

A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms

Table 1. Summary taxonomy of several dense two-frame stereo correspondence methods. The methods are grouped to contrast different matching costs (top), aggregation methods (middle), and optimization techniques (third section). The last section lists some papers outside our framework. Key to abbreviations: hier.—hierarchical (coarse-to-fine), WTA—winner-take-all, DP—dynamic programming, SA—simulated annealing, GC—graph cut.

Method	Matching cost	Aggregation	Optimization
SSD (traditional)	Squared difference	Square window	WTA
Hannah (1974)	Cross-correlation	(Square window)	WTA
Nishihara (1984)	Binarized filters	Square window	WTA
Kass (1988)	Filter banks	-None-	WTA
Fleet et al. (1991)	Phase	-None-	Phase-matching
Jones and Malik (1992)	Filter banks	-None-	WTA
Kanade (1994)	Absolute difference	Square window	WTA
Scharstein (1994)	Gradient-based	Gaussian	WTA
Zabih and Woodfill (1994)	Rank transform	(Square window)	WTA
Cox et al. (1995)	Histogram eq.	-None-	DP
Frohlinghaus and Buhmann (1996)	Wavelet phase	-None-	Phase-matching
Birchfield and Tomasi (1998a)	Shifted abs. diff	-None-	DP
Marr and Poggio (1976)	Binary images	Iterative aggregation	WTA
Prazdny (1985)	Binary images	3D aggregation	WTA
Szeliski and Hinton (1985)	Binary images	Iterative 3D aggregation	WTA
Okutomi and Kanade (1992)	Squared difference	Adaptive window	WTA
Yang et al. (1993)	Cross-correlation	Non-linear filtering	Hier. WTA
Shah (1993)	Squared difference	Non-linear diffusion	Regularization
Boykov et al. (1998)	Thresh. abs. diff.	Connected-component	WTA
Scharstein and Szeliski (1998)	Robust sq. diff.	Iterative 3D aggregation	Mean-field
Zitnick and Kanade (2000)	Squared difference	Iterative aggregation	WTA
Veksler (2001)	Abs. diff-avg.	Adaptive window	WTA
Quam (1984)	Cross-correlation	-None-	Hier. Warp
Barnard (1989)	Squared difference	-None-	SA
Geiger et al. (1992)	Squared difference	Shiftable window	DP
Belhumeur (1996)	Squared difference	-None-	DP
Cox et al. (1996)	Squared difference	-None-	DP
Ishikawa and Geiger (1998)	Squared difference	-None-	Graph cut
Roy and Cox (1998)	Squared difference	-None-	Graph cut
Bobick and Intille (1999)	Absolute difference	Shiftable window	DP
Boykov et al. (2001)	Squared difference	-None-	Graph cut
Kolmogorov and Zabih (2001)	Squared difference	-None-	Graph cut
Birchfield and Tomasi (1999)	Shifted abs. diff.	-None-	GC + planes
Tao et al. (2001)	Squared difference	(Color segmentation)	WTA + regions

How does deep learning apply?

- Replace intensity-based photometric cost with learned features

- Regress disparity based on cost volume

Benchmarks

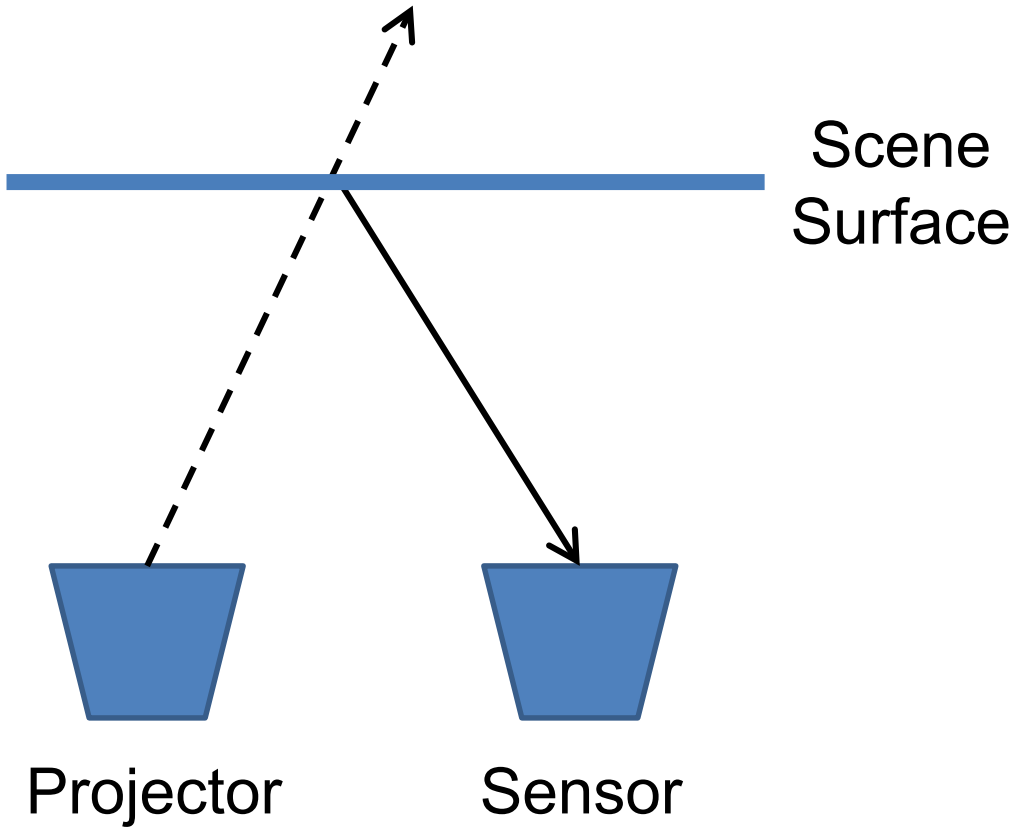
- Middlebury: https://vision.middlebury.edu/stereo/data/

- KITTI: http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=stereo

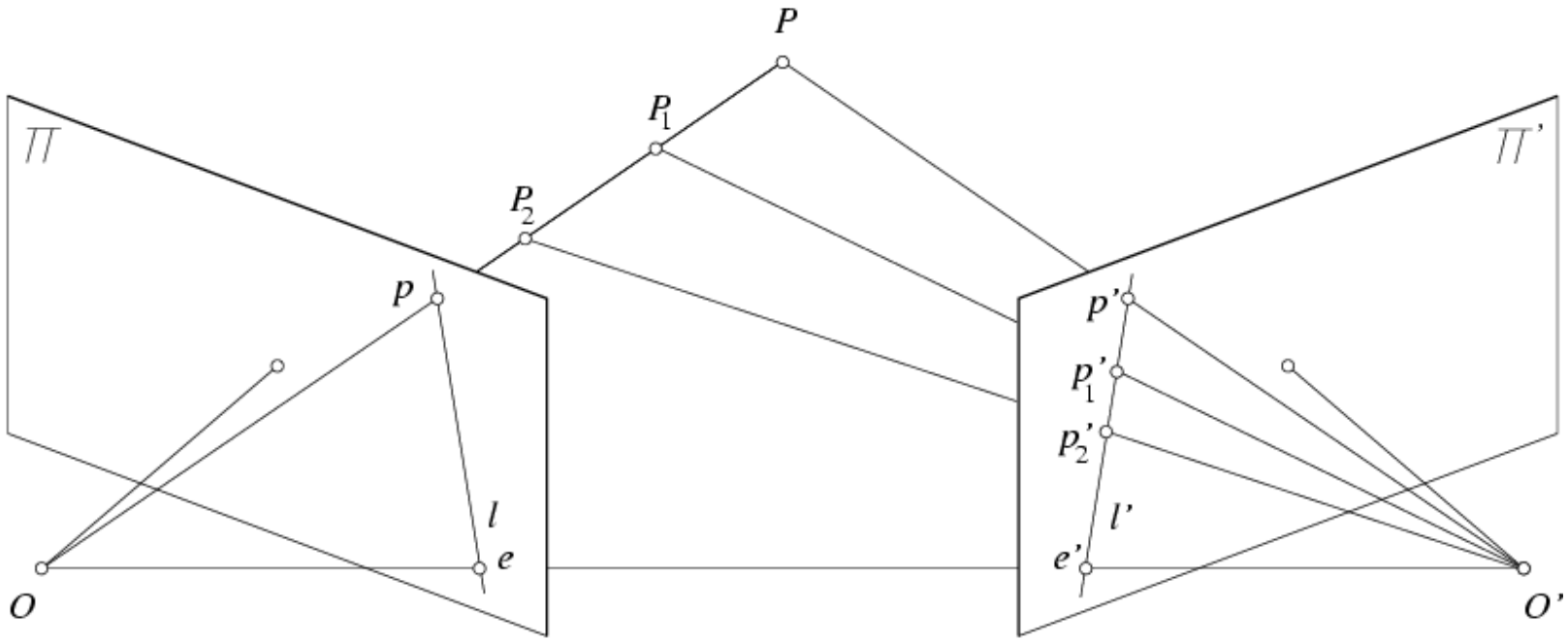
- Papers with code:
  https://paperswithcode.com/datasets?q=Stereo&v=lst&o=match&mod=stereo&page=1

Kinect: Depth from Projector-Sensor

Only one image: How is it possible to get depth?



Same stereo algorithms apply

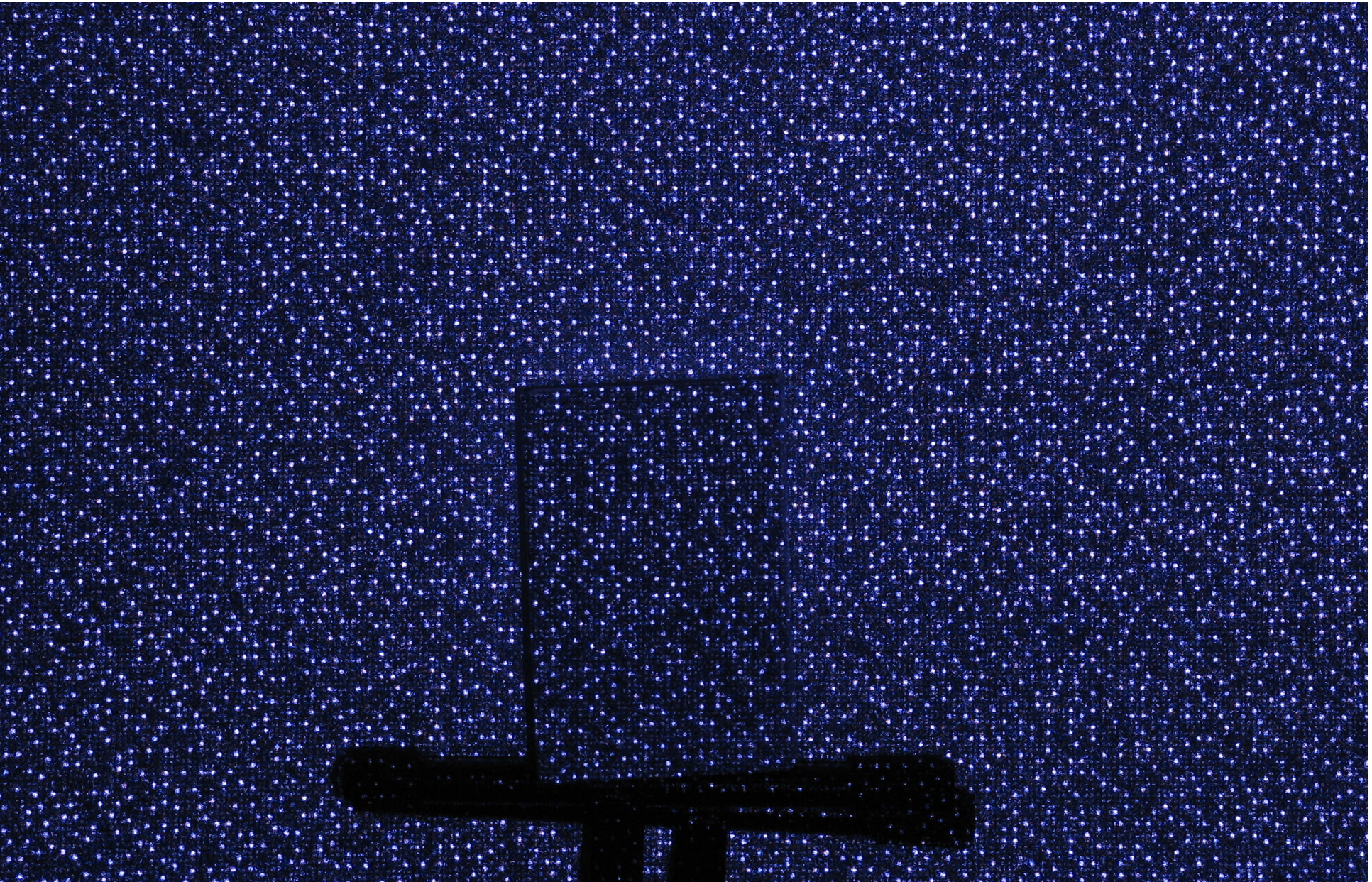


Projector

Sensor

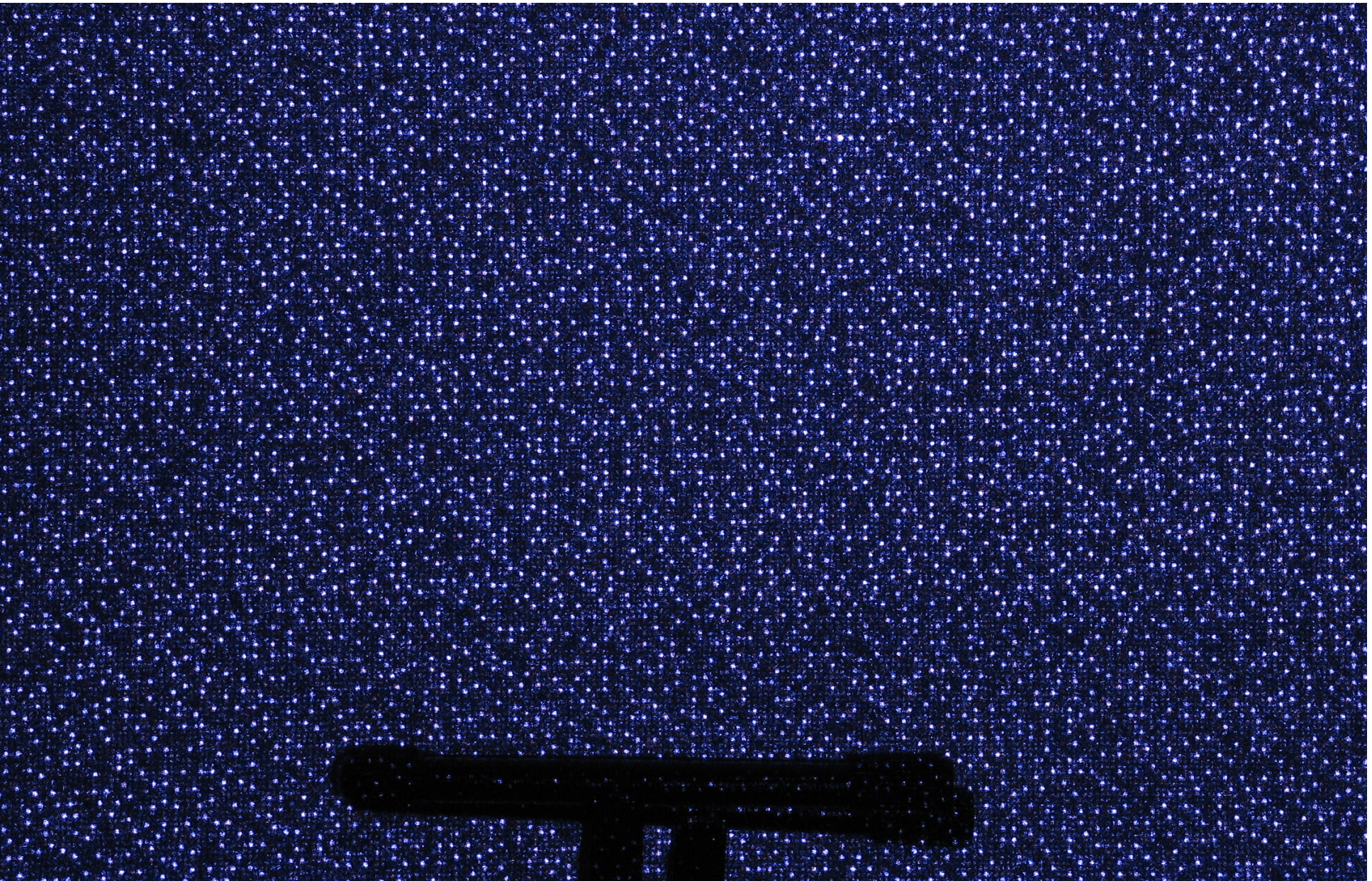
Source: http://www.futurepicture.org/?p=97

Example: Book vs. No Book



Source: http://www.futurepicture.org/?p=97

Example: Book vs. No Book



Region-growing Random Dot Matching

1. Detect dots (“speckles”) and label them unknown
2. Randomly select a region anchor, a dot with unknown depth
   a. Windowed search via normalized cross correlation along scanline
      – Check that best match score is greater than threshold; if not, mark as “invalid” and go to 2
   b. Region growing (dynamic program)
      1. Neighboring pixels are added to a queue
      2. For each pixel in queue, initialize by anchor’s shift; then search small local neighborhood; if matched, add neighbors to queue
      3. Stop when no pixels are left in the queue
3. Repeat until all dots have known depth or are marked “invalid”

http://www.wipo.int/patentscope/search/en/WO2007043036

Projected IR vs. Natural Light Stereo

- What are the advantages of IR?
  - Works in low light conditions
  - Does not rely on having textured objects
  - Not confused by repeated scene textures
  - Can tailor algorithm to produced pattern

- What are advantages of natural light?
  - Works outside, anywhere with sufficient light
  - Uses less energy
  - Resolution limited only by sensors, not projector

- Difficulties with both
  - Very dark surfaces may not reflect enough light
  - Specular reflection in mirrors or metal causes trouble

Summary

- 3D points can be triangulated from corresponding pixels in two images, constrained by epipolar lines (usually scanlines)

- Stereo involves solving for disparity of each pixel by
  - Matching: Comparing pixel intensities, patches, deep features
  - Optimization: WTA, smoothness, and other priors
  - Refinement: subpixel optimization via parabolic fit or gradient

- Stereo works best on the interior of highly textured, non-reflective surfaces

- Stereo algorithms also apply when light is projected with a known pattern and other cases like varying light sources

Next class

- Choose paper with your group today
- Read paper and submit individual review. Can discuss with group, but write your review on your own.
- Discuss with your group in class Tues and make summary slide – 20 minutes
- Groups present their slides in order of publication date (oldest to newest) – 30-50 minutes