

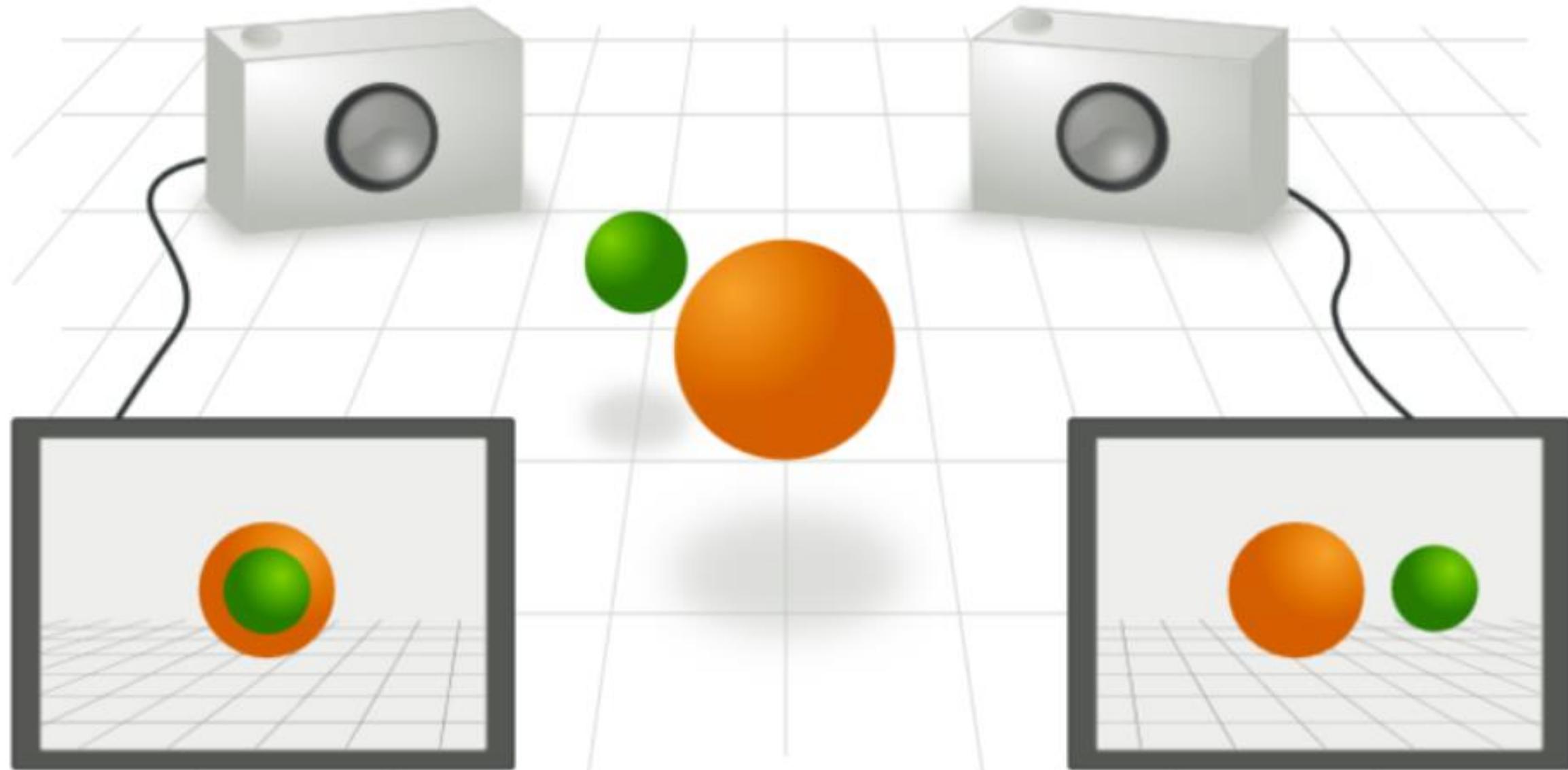
EBU7240

Computer Vision

- Stereo: Epipolar geometry -

Semester 1, 2021

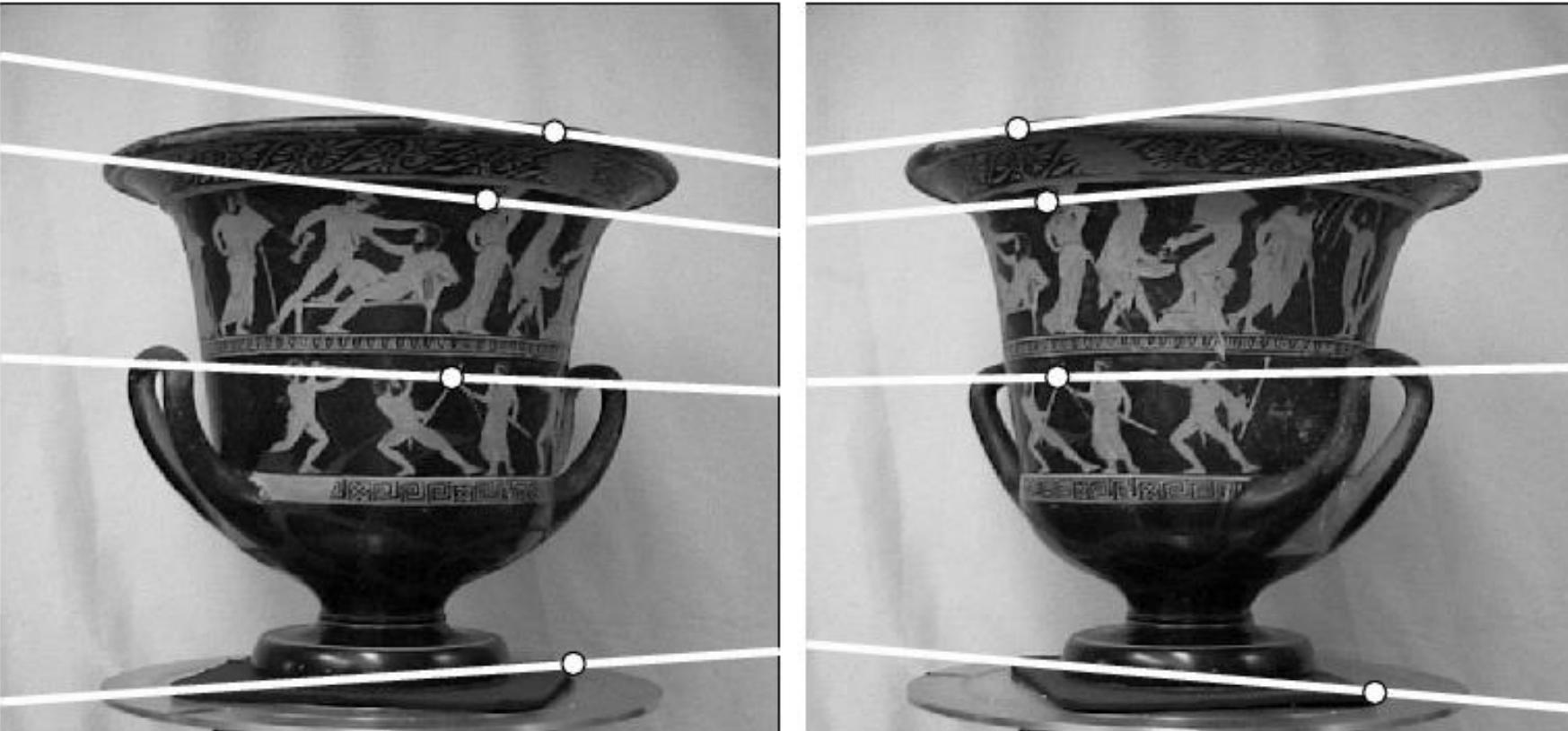
Changjae Oh



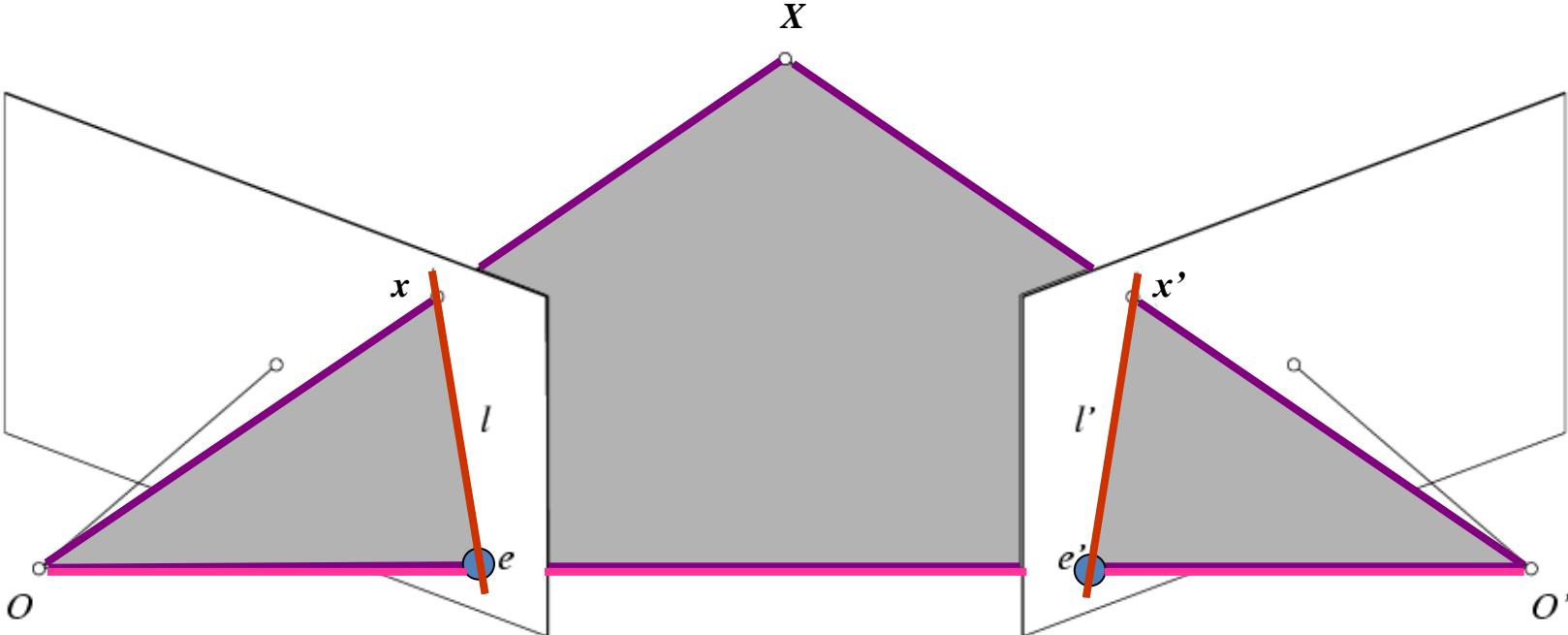
Objectives

- To understand epipolar geometry for stereo vision
- To understand epipolar constraint in calibrated/uncalibrated case

Two-view geometry

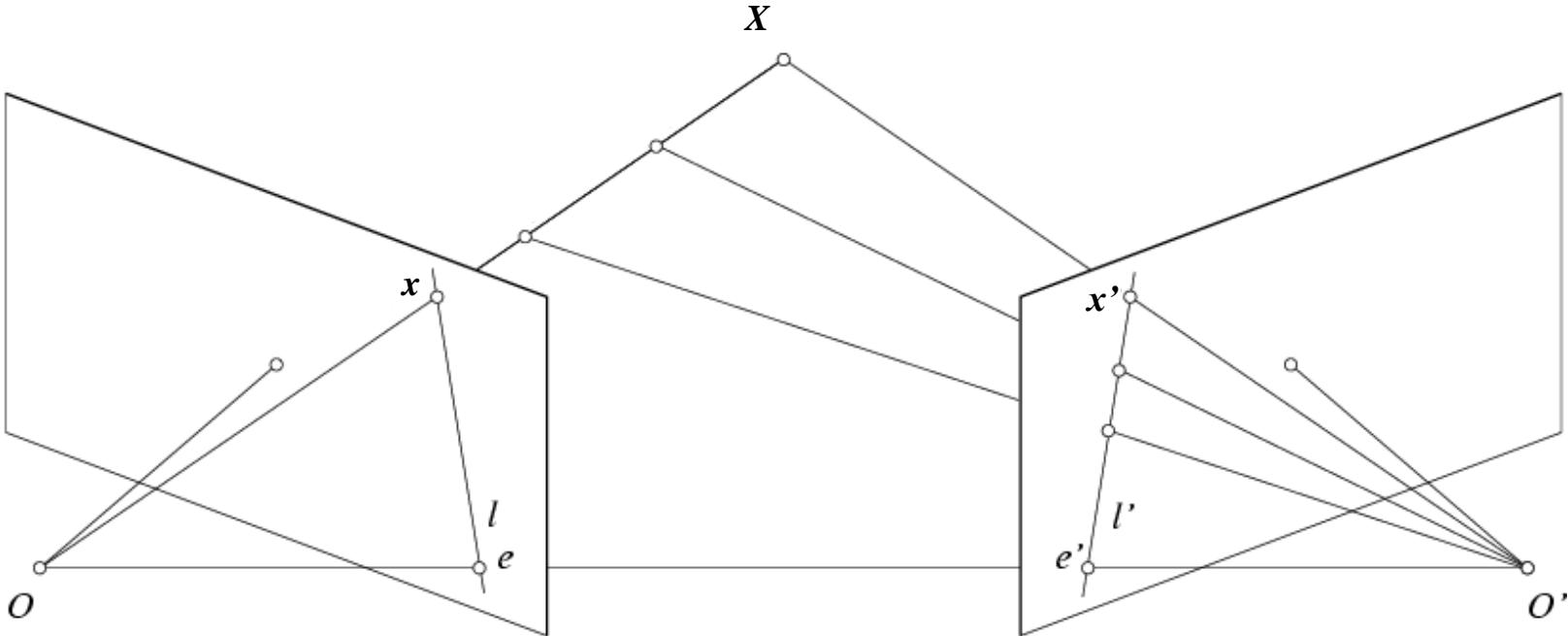


Epipolar geometry



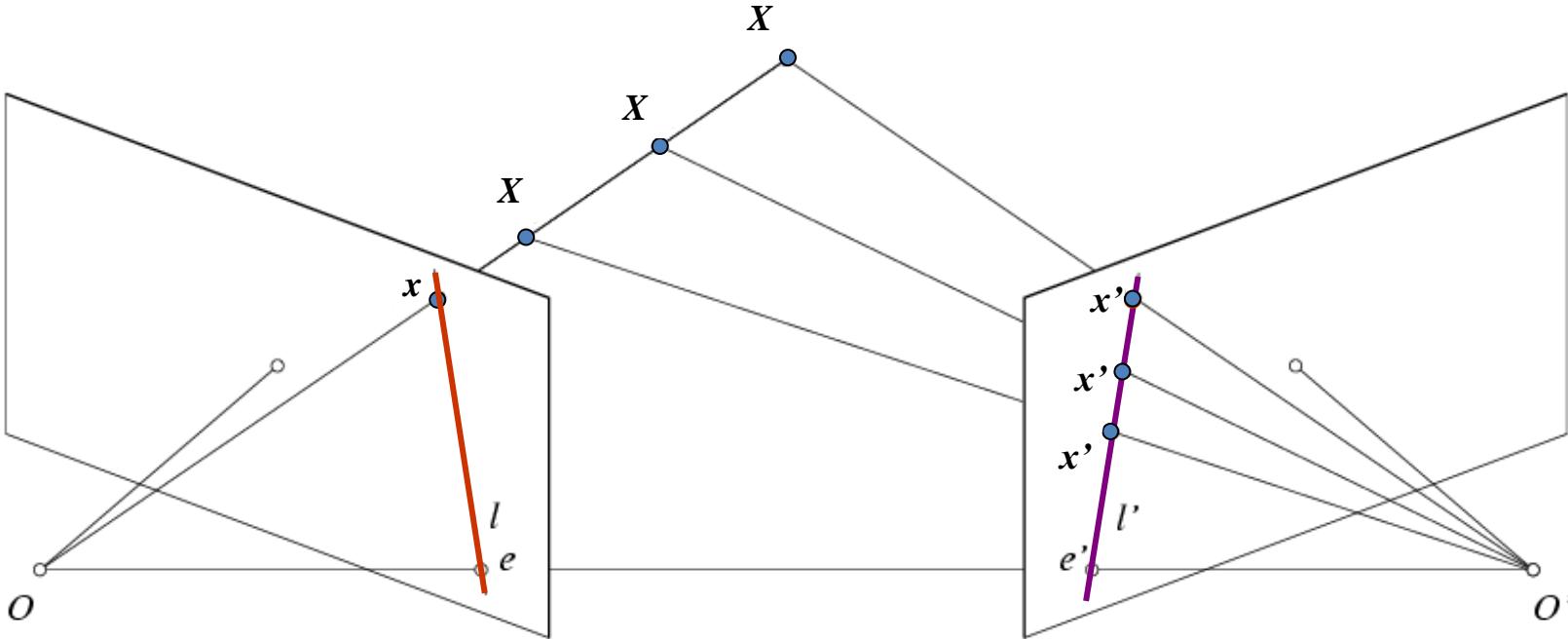
- **Baseline** – line connecting the two camera centers
- **Epipolar Plane** – plane containing baseline (1D family)
- **Epipoles**
 - = intersections of baseline with image planes
 - = projections of the other camera center
 - = vanishing points of the motion direction
- **Epipolar Lines** - intersections of epipolar plane with image planes (always come in corresponding pairs)

Epipolar constraint



- If we observe a point x in one image, where can the corresponding point x' be in the other image?

Epipolar constraint



- Potential matches for x have to lie on the corresponding epipolar line l' .
- Potential matches for x' have to lie on the corresponding epipolar line l .

Epipolar constraint

- An important concept for stereo vision

Task: Match point in the left image to point in the right image



How would you do it?
Block matching to the entire image? (EBU6230)

Epipolar constraint

- An important concept for stereo vision

Task: Match point in the left image to point in the right image



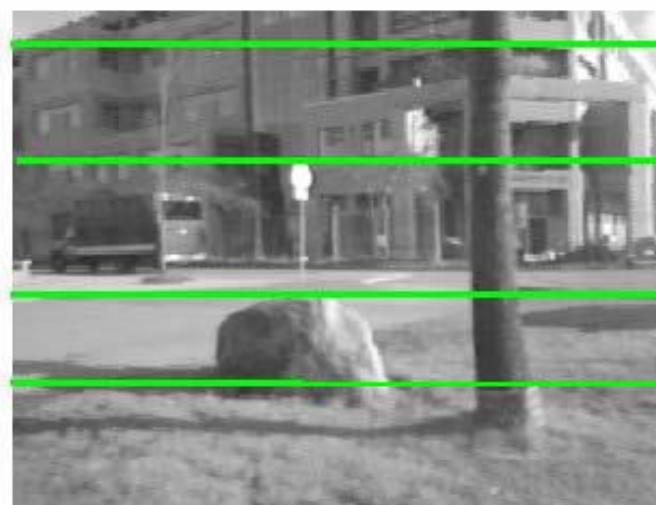
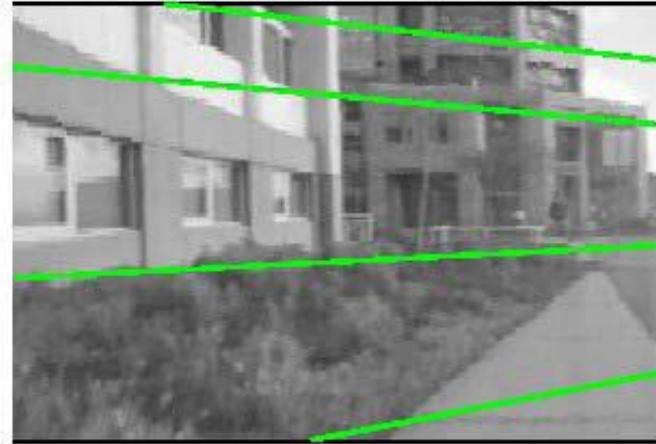
Want to avoid search over entire image

(if the images have been rectified)

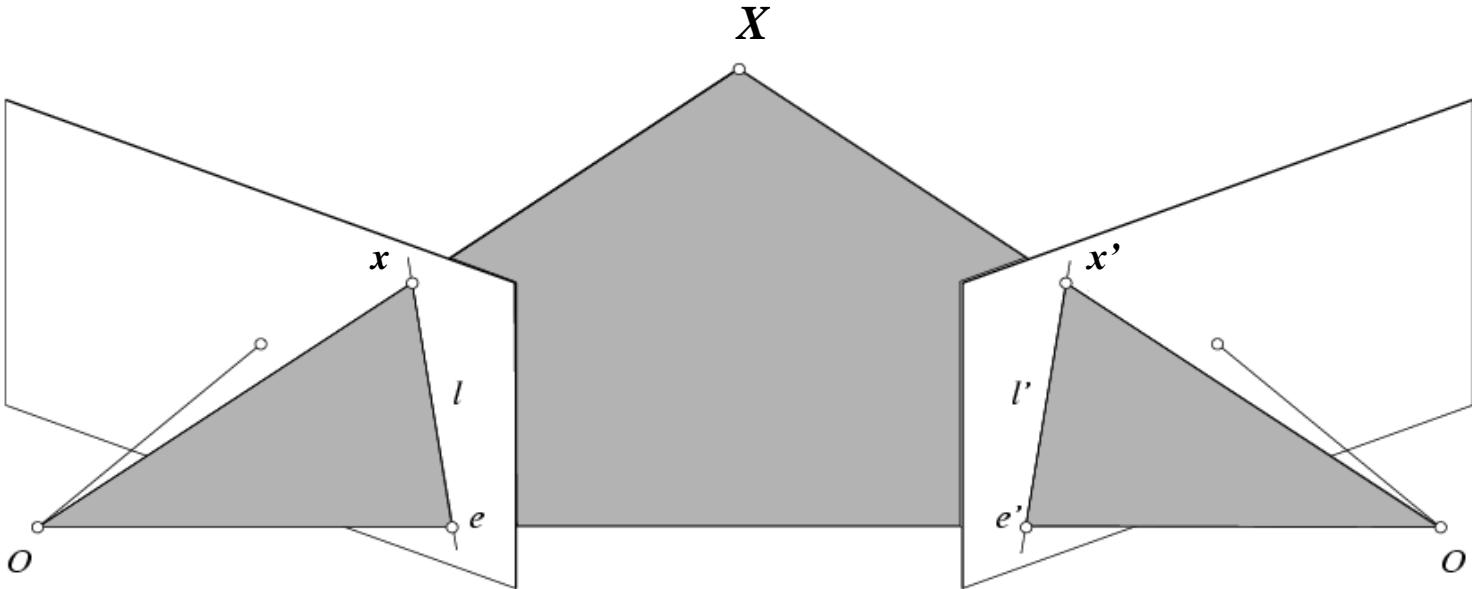
Epipolar constraint reduces search to a single line

Epipolar constraint example

- Epipolar constraint reduces search range to a single line



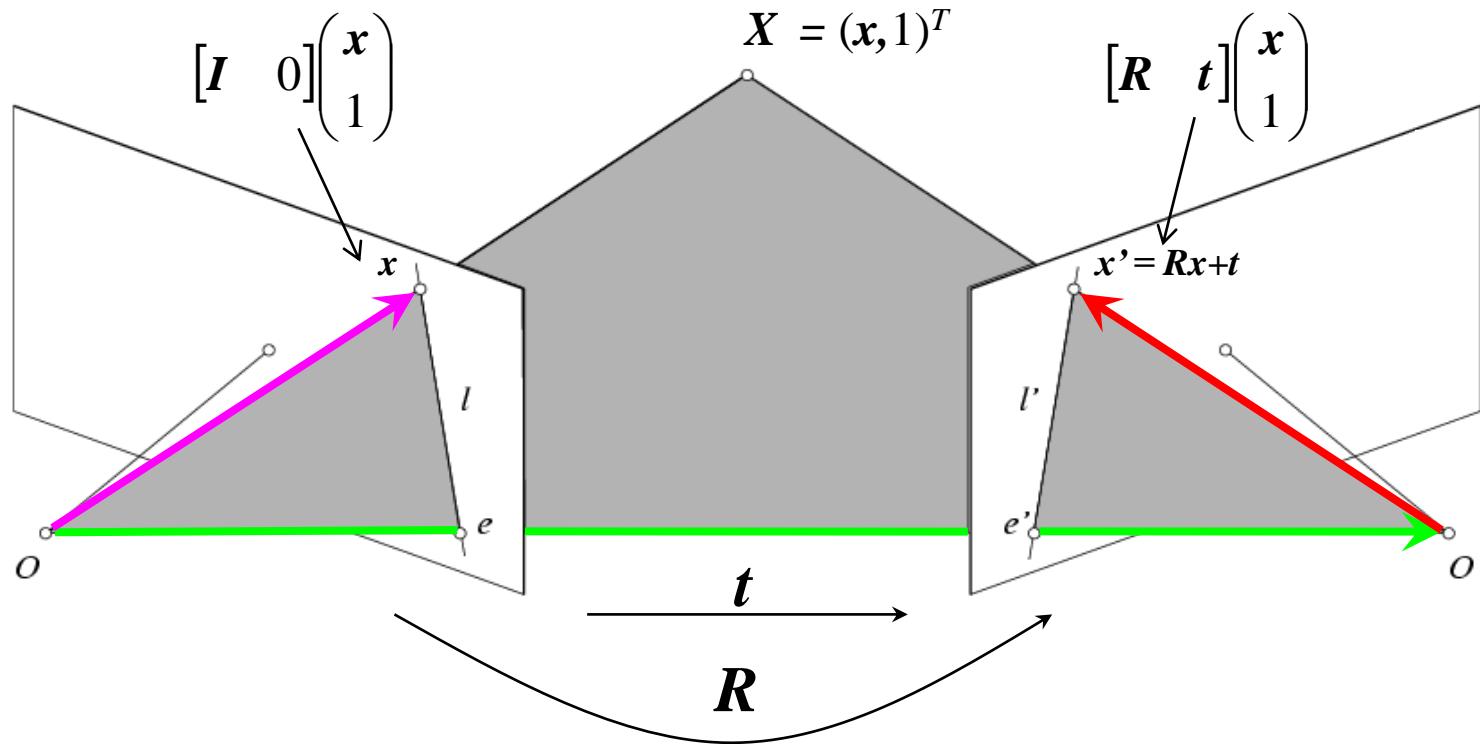
Epipolar constraint: Calibrated case



- Intrinsic and extrinsic parameters of the cameras are known, world coordinate system is set to that of the first camera
- Then the projection matrices are given by $K[I \mid 0]$ and $K'[R \mid t]$
- We can multiply the projection matrices (and the image points) by the inverse of the calibration matrices to get *normalized* image coordinates:

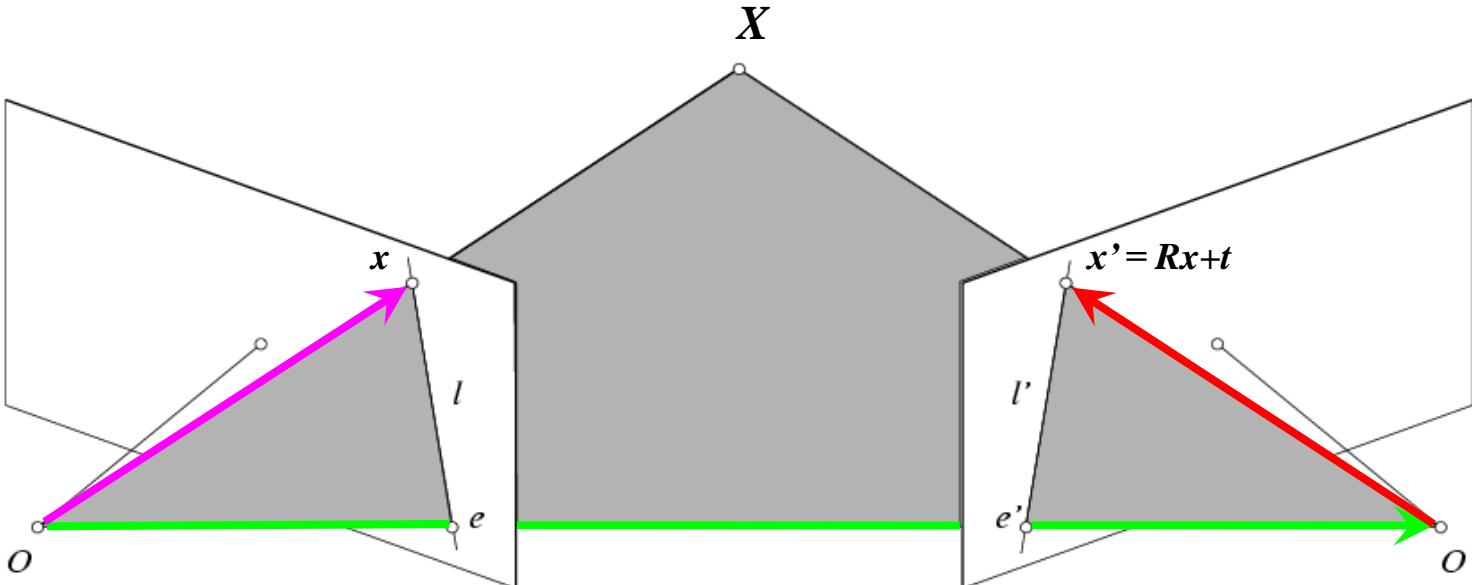
$$\mathbf{x}_{\text{norm}} = \mathbf{K}^{-1} \mathbf{x}_{\text{pixel}} = [\mathbf{I} \ 0] \mathbf{X}, \quad \mathbf{x}'_{\text{norm}} = \mathbf{K}'^{-1} \mathbf{x}'_{\text{pixel}} = [\mathbf{R} \ \mathbf{t}] \mathbf{X}$$

Epipolar constraint: Calibrated case



The vectors Rx , t , and x' are coplanar

Epipolar constraint: Calibrated case

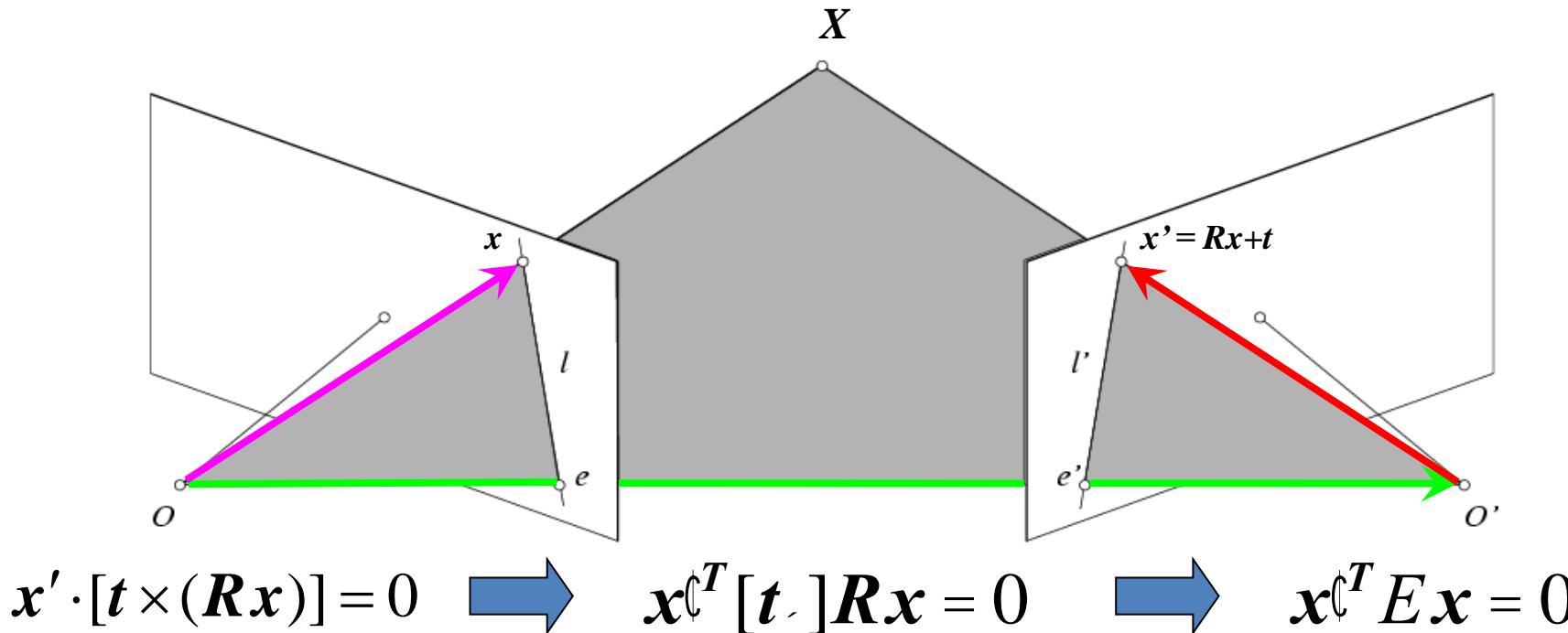


$$x' \cdot [t \times (Rx)] = 0 \quad \Rightarrow \quad x^T [t^T] Rx = 0$$

Recall: $\mathbf{a} \times \mathbf{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = [\mathbf{a}_x] \mathbf{b}$

The vectors Rx , t , and x' are coplanar

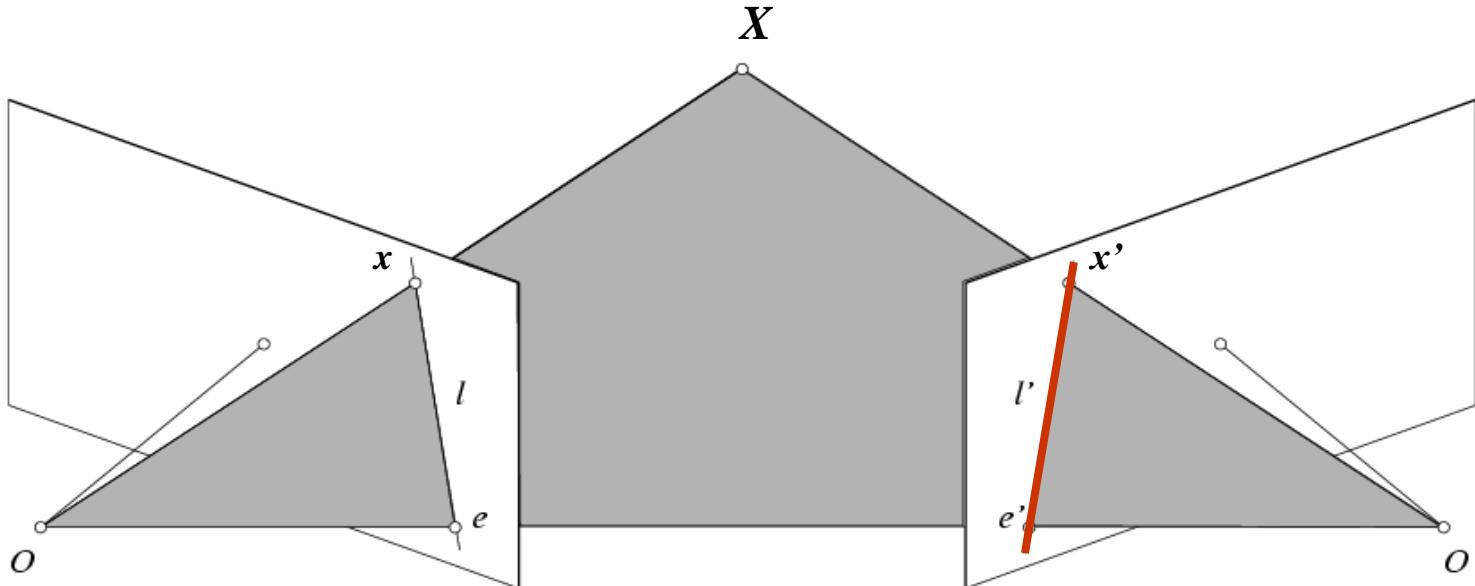
Epipolar constraint: Calibrated case



Essential Matrix
(Longuet-Higgins, 1981)

The vectors Rx , t , and x' are coplanar

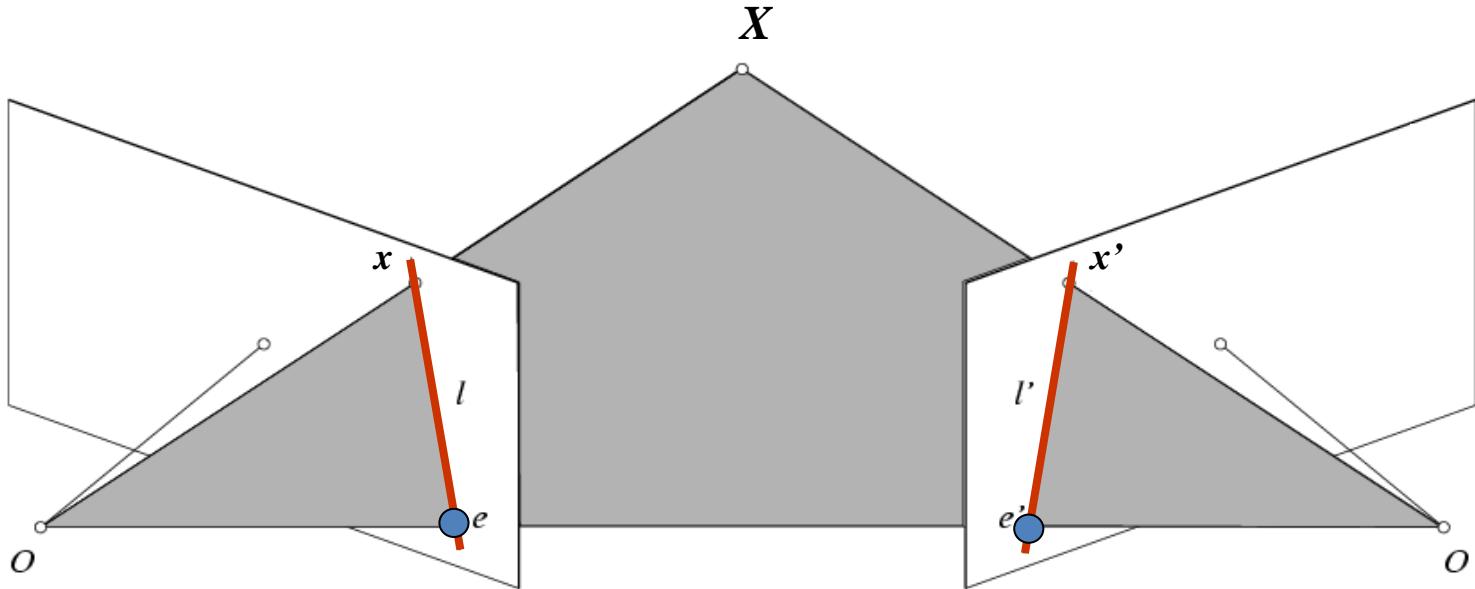
Epipolar constraint: Calibrated case



- Ex is the epipolar line associated with x ($l' = Ex$)
 - Recall: a line is given by $ax + by + c = 0$ or

$$\mathbf{l}^T \mathbf{x} = 0 \quad \text{where} \quad \mathbf{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

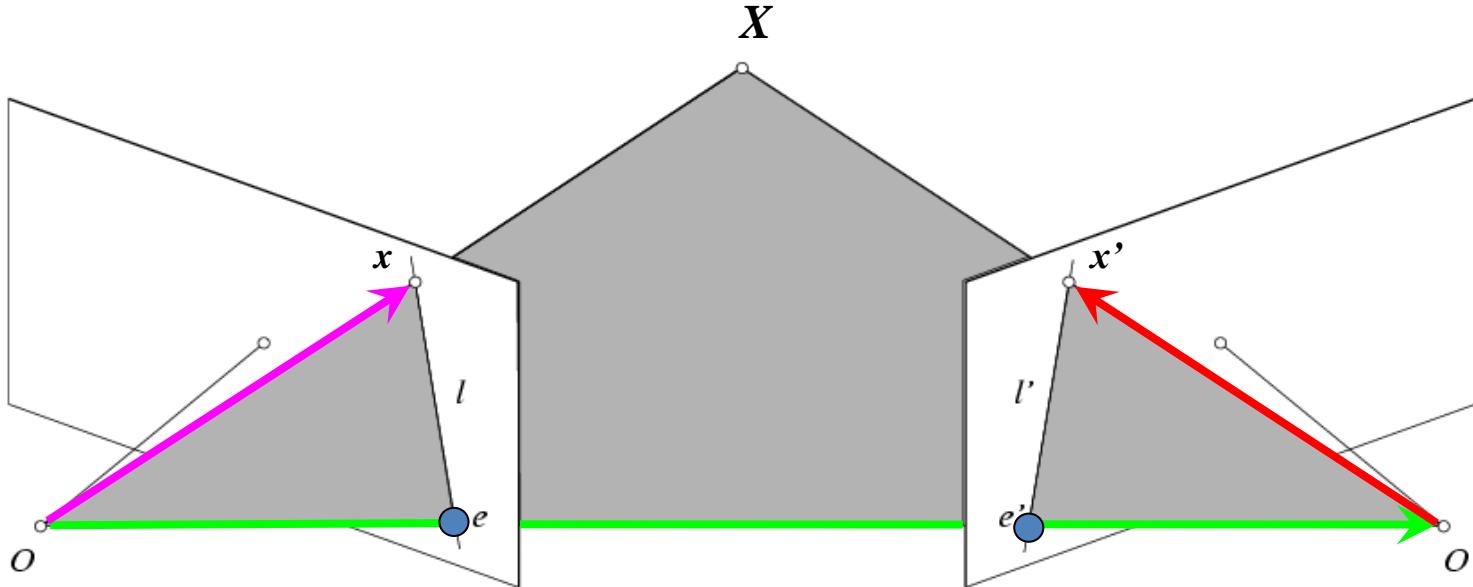
Epipolar constraint: Calibrated case



$$\mathbf{x}^T \mathbf{E} \mathbf{x} = 0$$

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

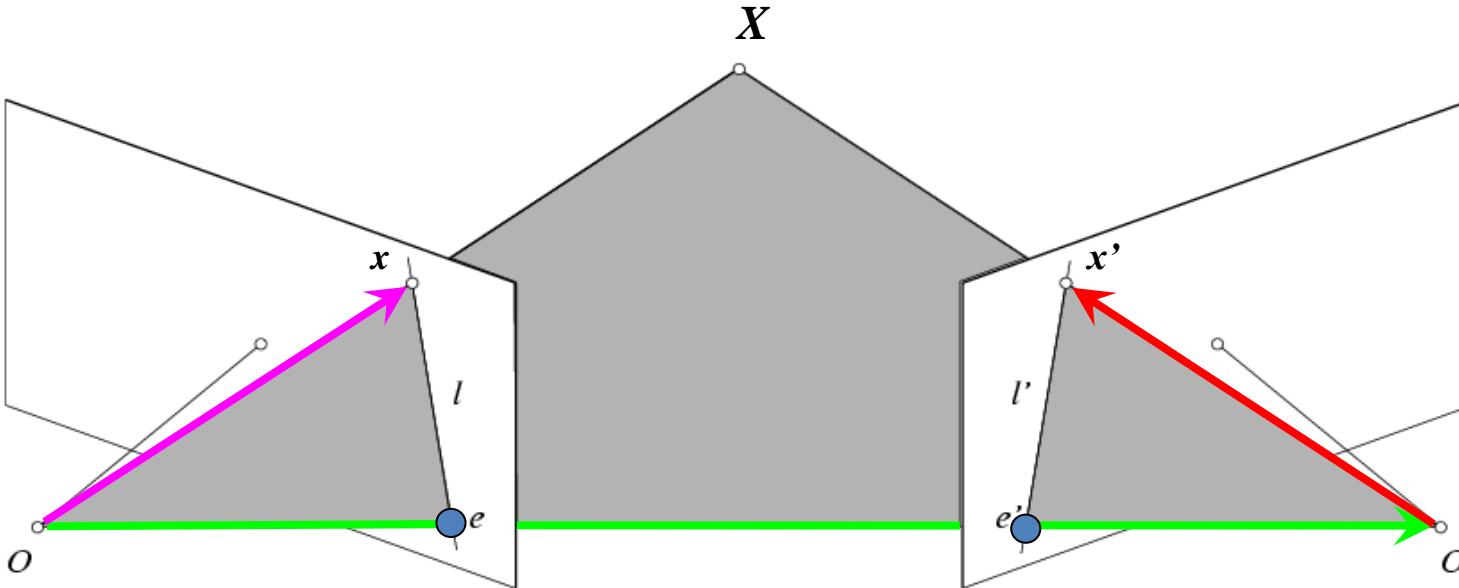
Epipolar constraint: Uncalibrated case



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

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

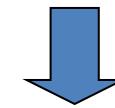
Epipolar constraint: Uncalibrated case



$$\hat{x}'^T E \hat{x} = 0 \quad \xrightarrow{\text{blue arrow}} \quad x'^T F x = 0 \quad \text{with} \quad F = K'^{-T} E K^{-1}$$

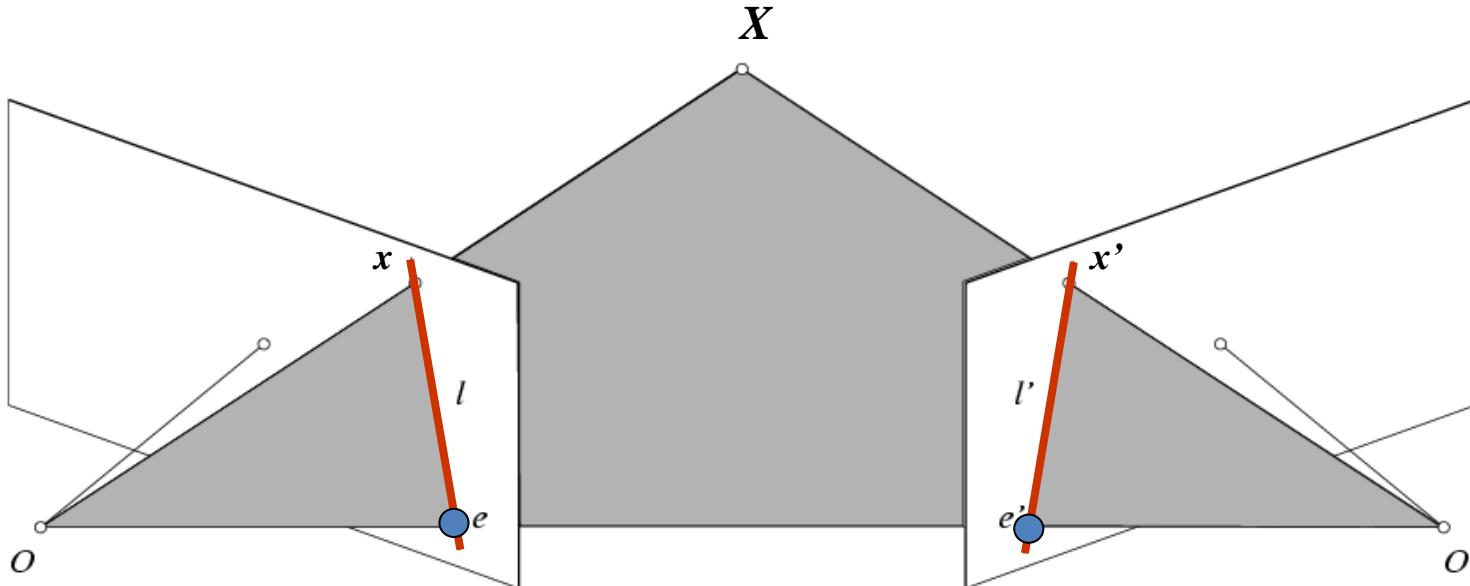
$$\hat{x} = K^{-1} x$$

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



Fundamental Matrix
(Faugeras and Luong, 1992)

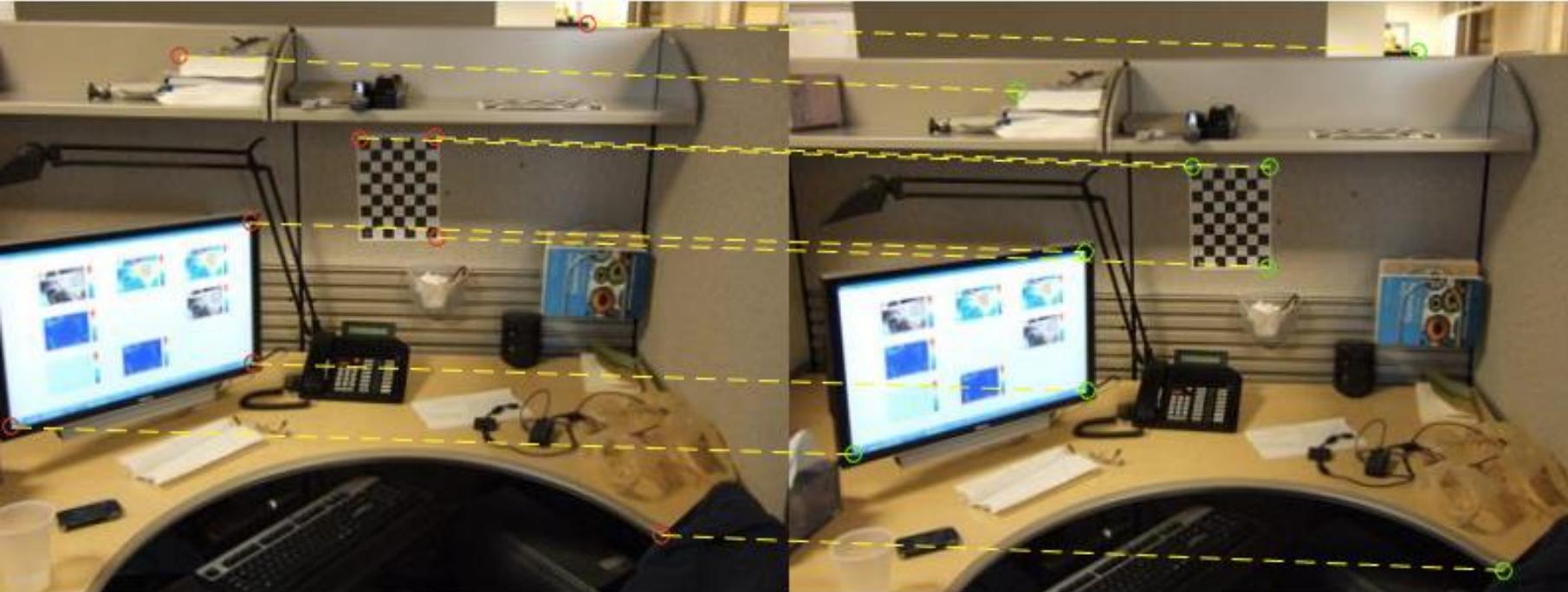
Epipolar constraint: Uncalibrated case



$$\hat{x}'^T E \hat{x} = 0 \quad \xrightarrow{\text{blue arrow}} \quad x'^T F x = 0 \quad \text{with} \quad F = K'^{-T} E K^{-1}$$

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

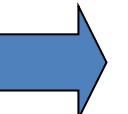
Estimating the fundamental matrix



The eight-point algorithm

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

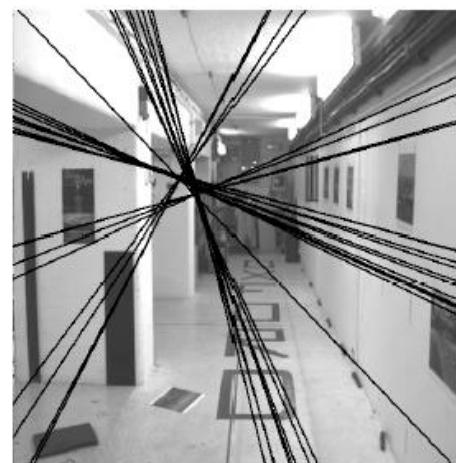
$$\begin{bmatrix} u' & v' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$$



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

Solve homogeneous
linear system using
eight or more matches

Enforce rank-2 constraint
(take SVD of \mathbf{F} and throw out
the smallest singular value)



Problem with eight-point algorithm

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

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

EBU7240

Computer Vision

- Stereo-

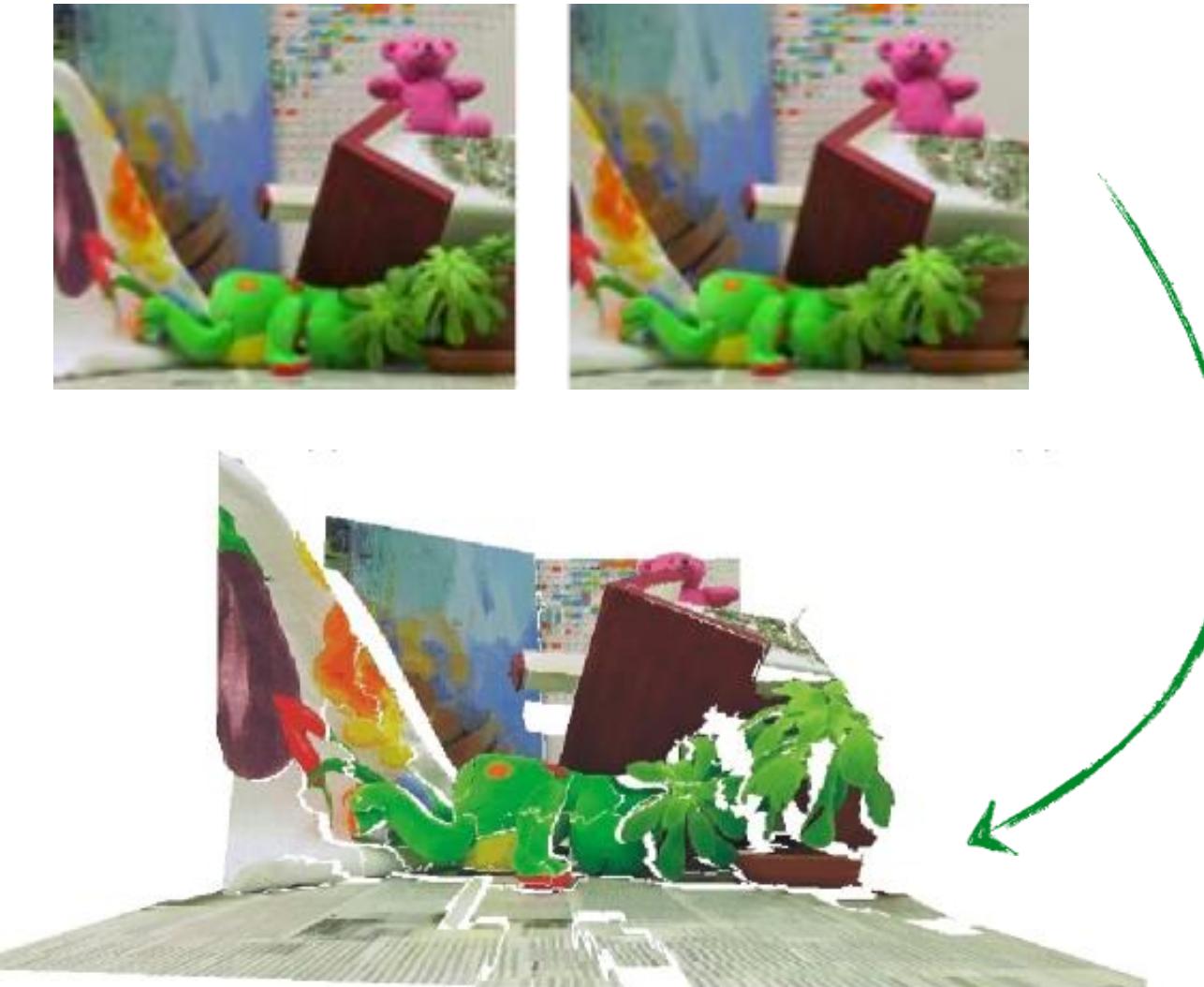
Semester 1, 2021

Changjae Oh

Objectives

- To understand the computational approach to depth estimation from stereo images
- To understand active method for depth estimation

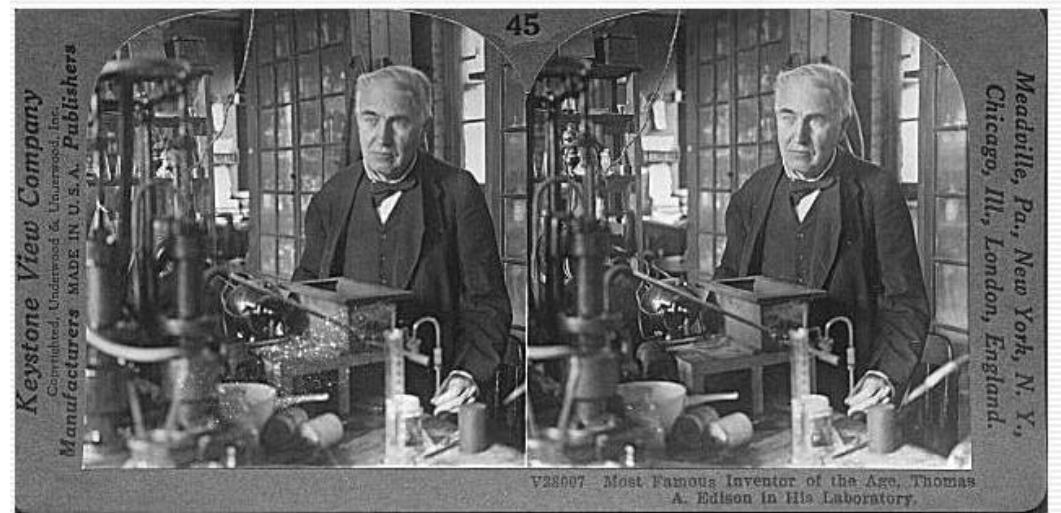
Two-View Stereo



Many slides adapted from Steve Seitz

Stereograms

- Humans can fuse pairs of images to get a sensation of depth



Stereograms: Invented by Sir Charles Wheatstone, 1838

Stereograms



Stereograms

- Humans can fuse pairs of images to get a sensation of depth



Autostereograms: www.magiceye.com

Stereograms

- Humans can fuse pairs of images to get a sensation of depth



Autostereograms: www.magiceye.com

Problem formulation

- Given a calibrated binocular stereo pair, fuse it to produce a depth image

image 1



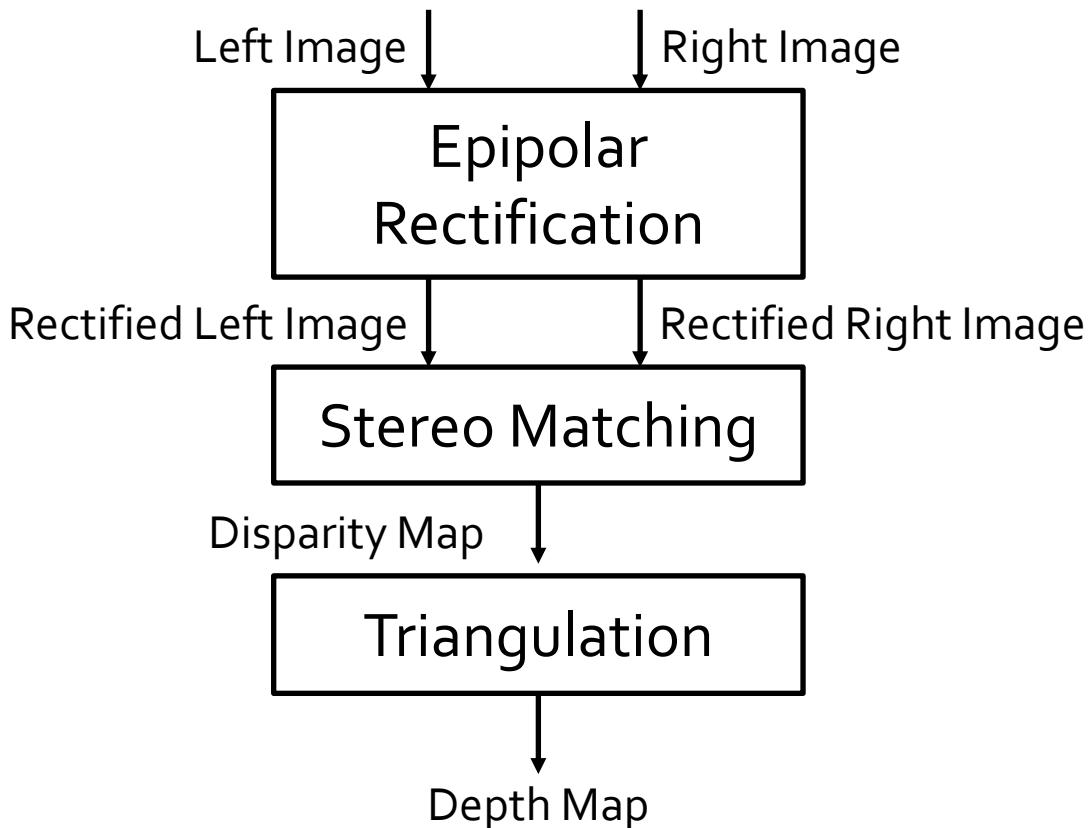
image 2



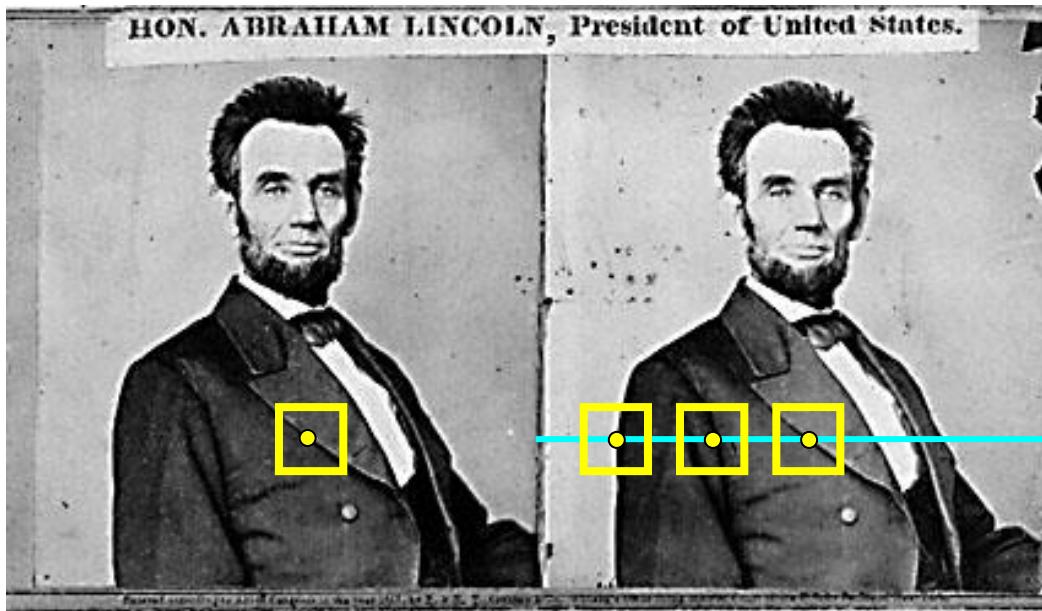
Dense depth map



Computational Stereo Pipeline

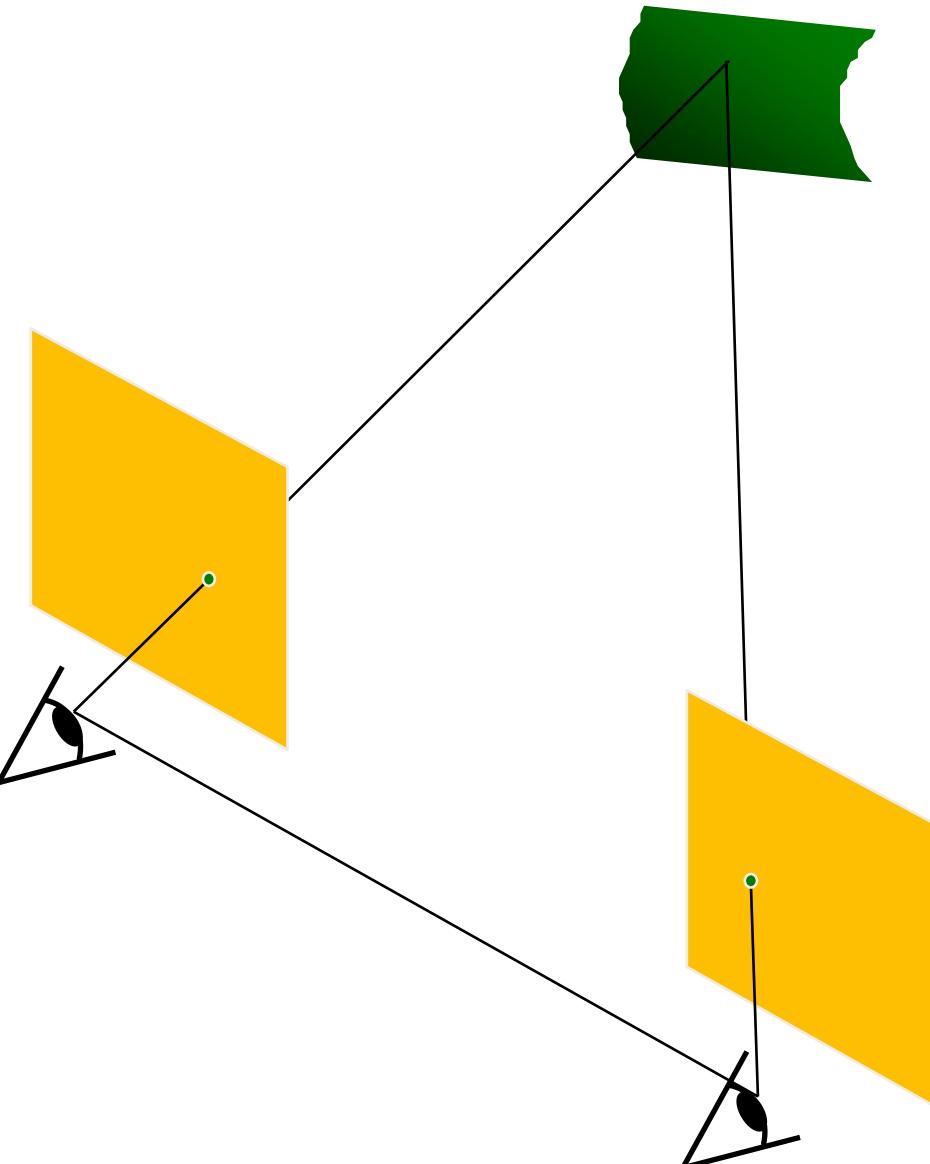


Epipolar rectification – Why needed?



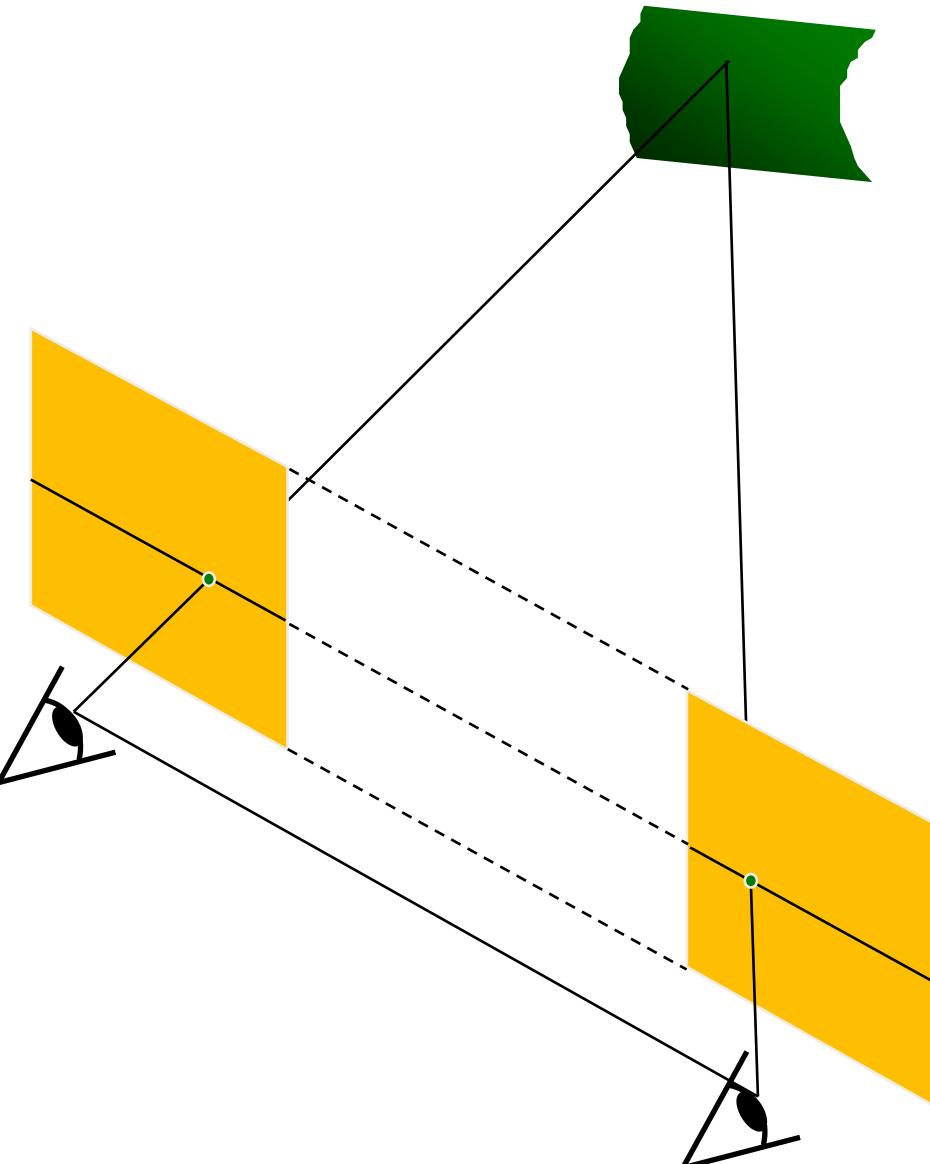
- **For each pixel in the first image**
 - Find corresponding epipolar line in the right image
 - Examine all pixels on the epipolar line and pick the best match
 - Triangulate the matches to get depth information
- **Simplest case: epipolar lines are corresponding scanlines**
 - When does this happen?

Simplest Case: Parallel images



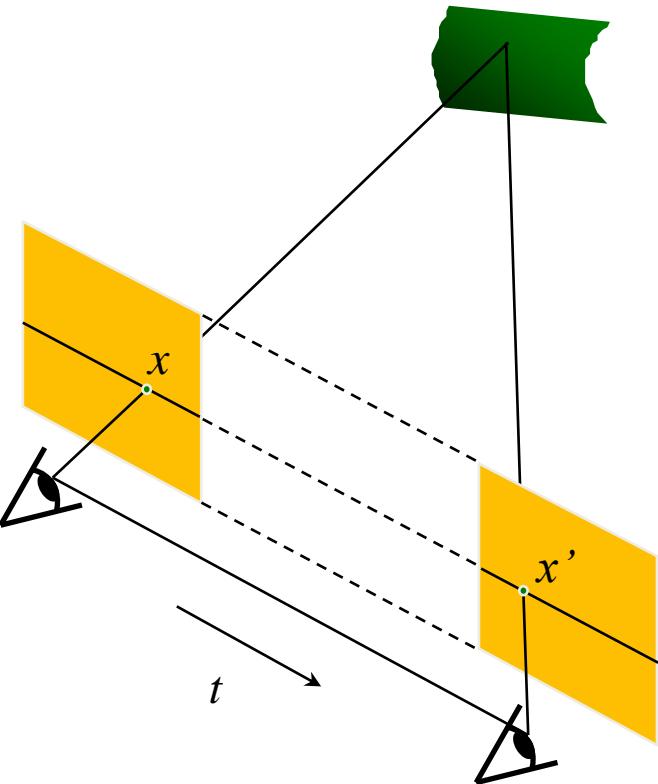
- Image planes of cameras are parallel to each other and to the baseline
- Camera centers are at same height
- Focal lengths are the same

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:

$$x'^T E x = 0, \quad E = [t_x]R$$

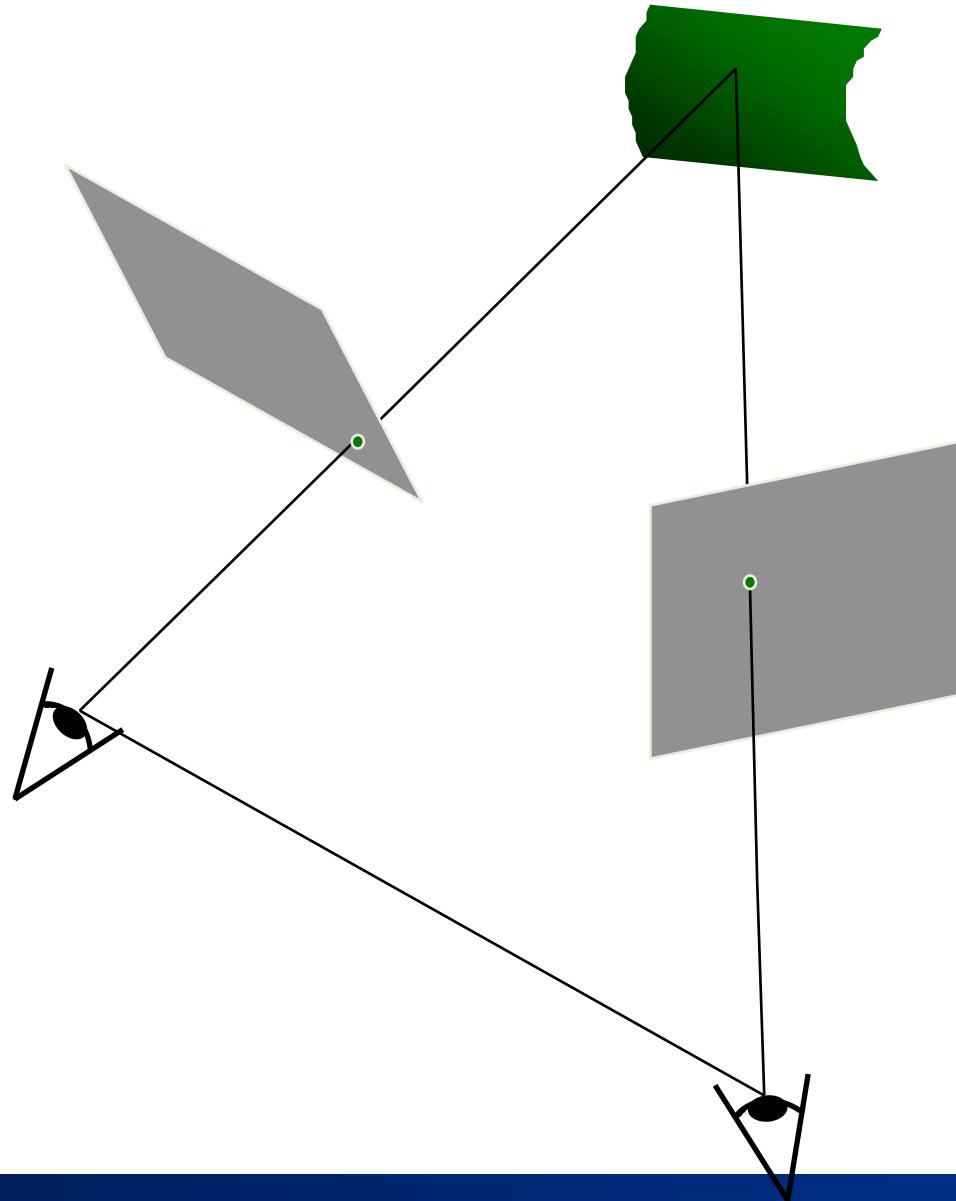
$$R = I \quad t = (T, 0, 0)$$

$$E = [t_x]R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix}$$

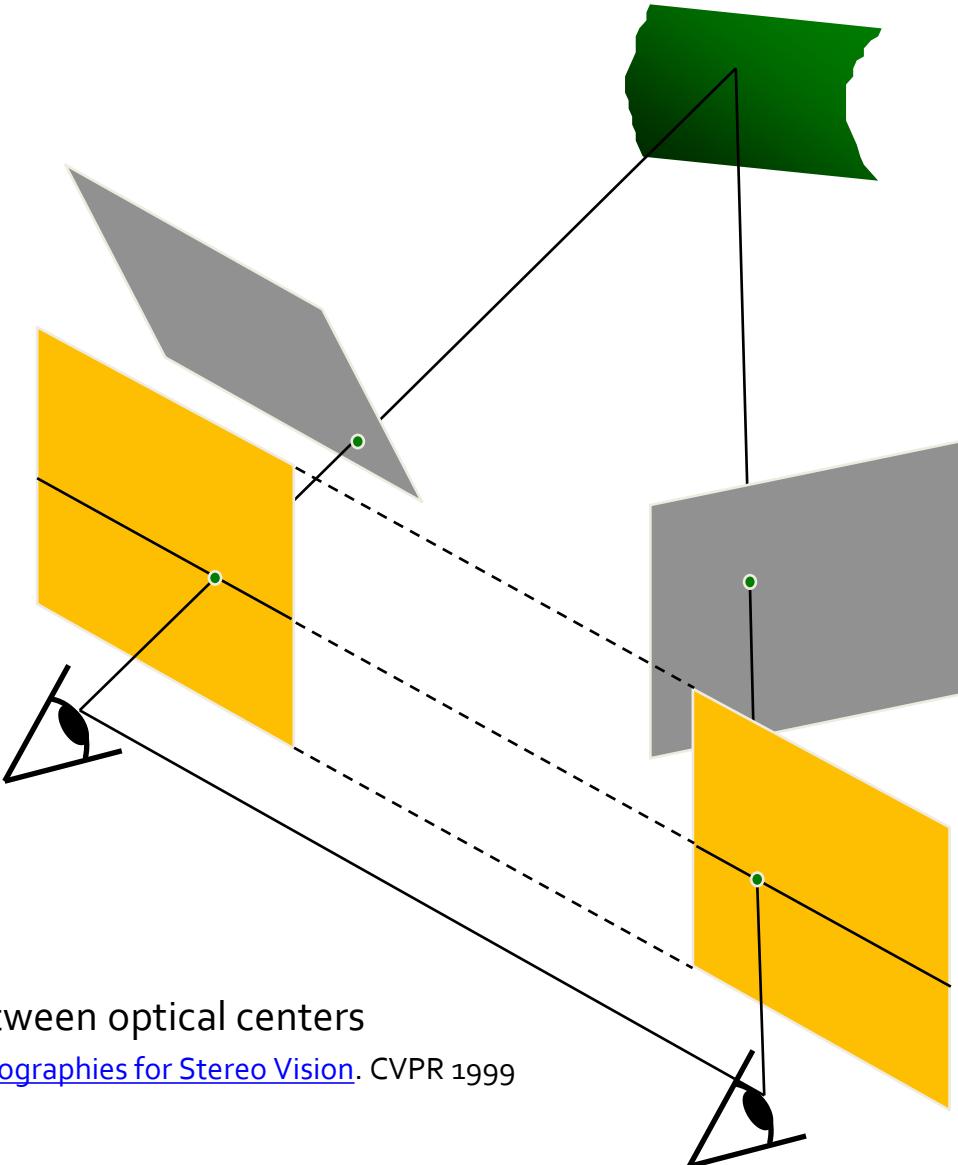
$$(u' \quad v' \quad 1) \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = 0 \quad (u' \quad v' \quad 1) \begin{pmatrix} 0 \\ -T \\ Tv \end{pmatrix} = 0 \quad Tv' = Tv$$

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 optical centers

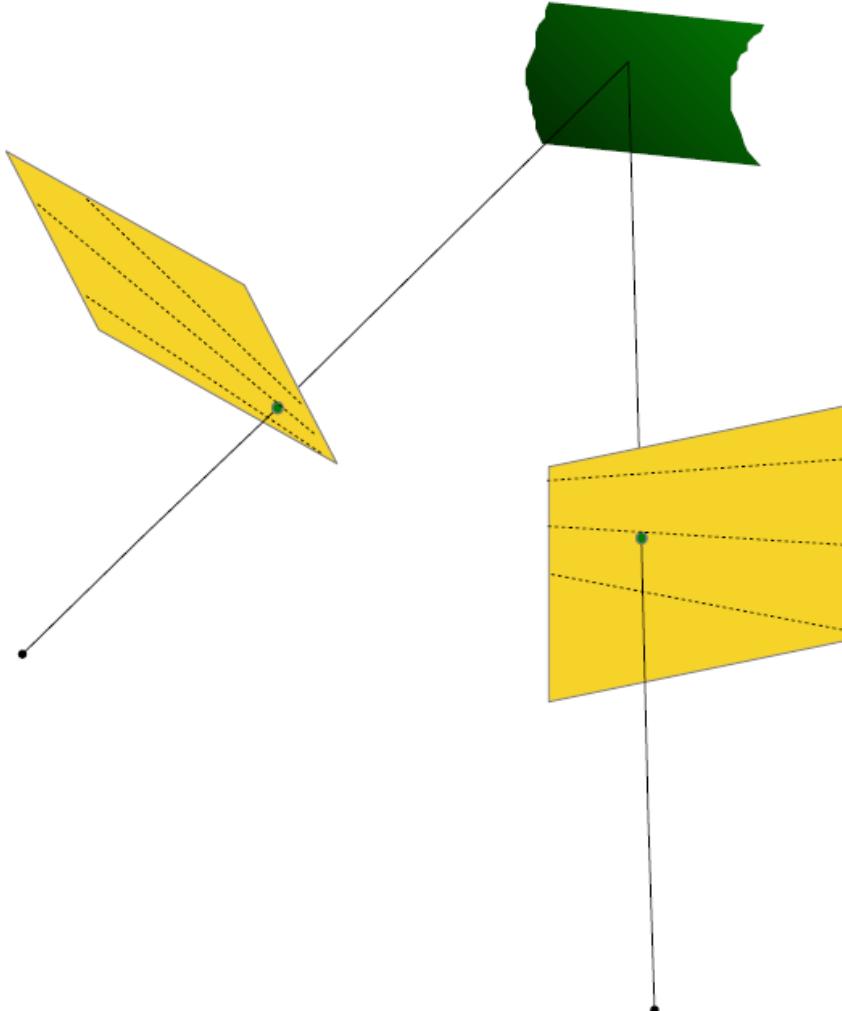
C. Loop and Z. Zhang. [Computing Rectifying Homographies for Stereo Vision](#). CVPR 1999

Stereo image rectification

1. Rotate the right camera by R
(aligns camera coordinate system orientation only)
2. Rotate (rectify) the left camera so that the epipole is at infinity
3. Rotate (rectify) the right camera so that the epipole is at infinity
4. Adjust the scale

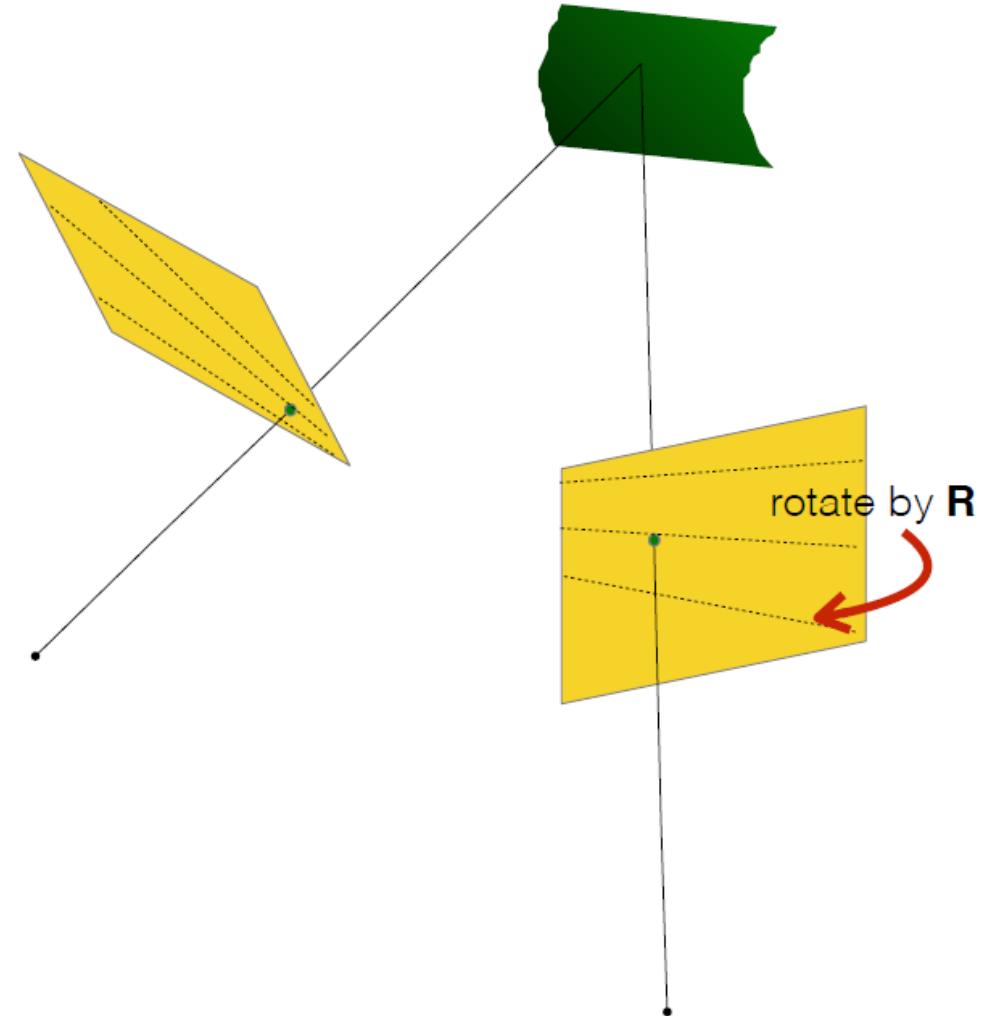
Stereo image rectification

- 1. Compute E to get R**
- 2. Rotate right image by R**
- 3. Rotate both images by R_{rect}**
- 4. Scale both images by H**



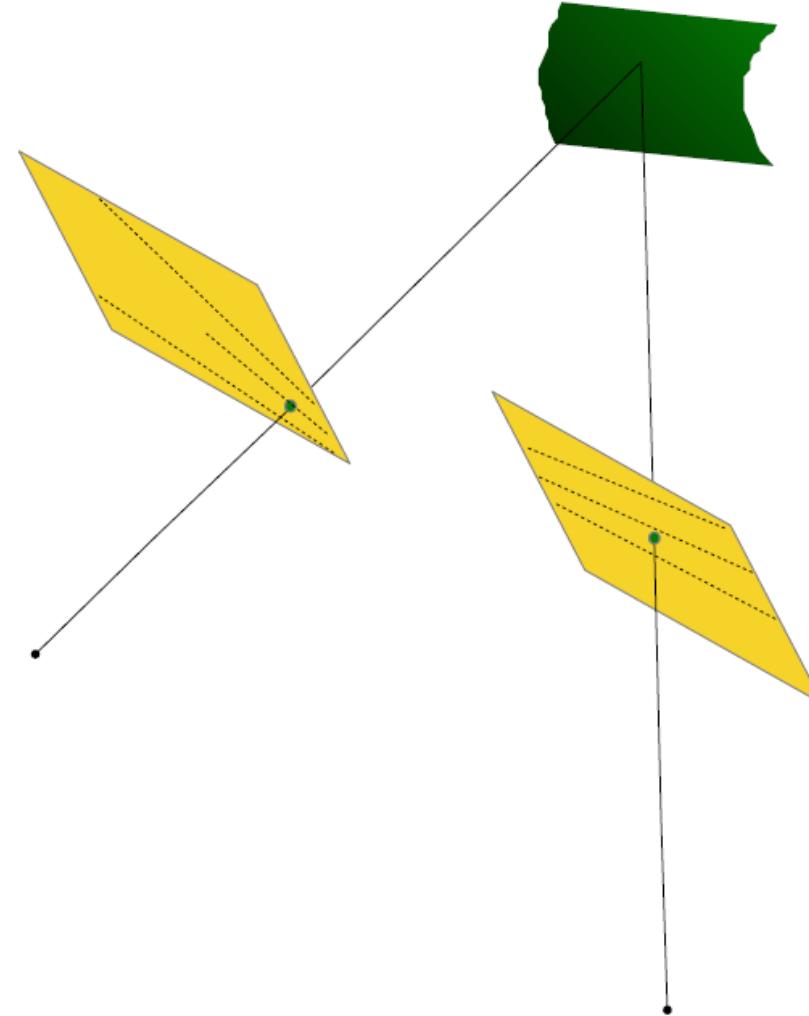
Stereo image rectification

1. Compute E to get R
2. Rotate right image by R
3. Rotate both images by R_{rect}
4. Scale both images by H



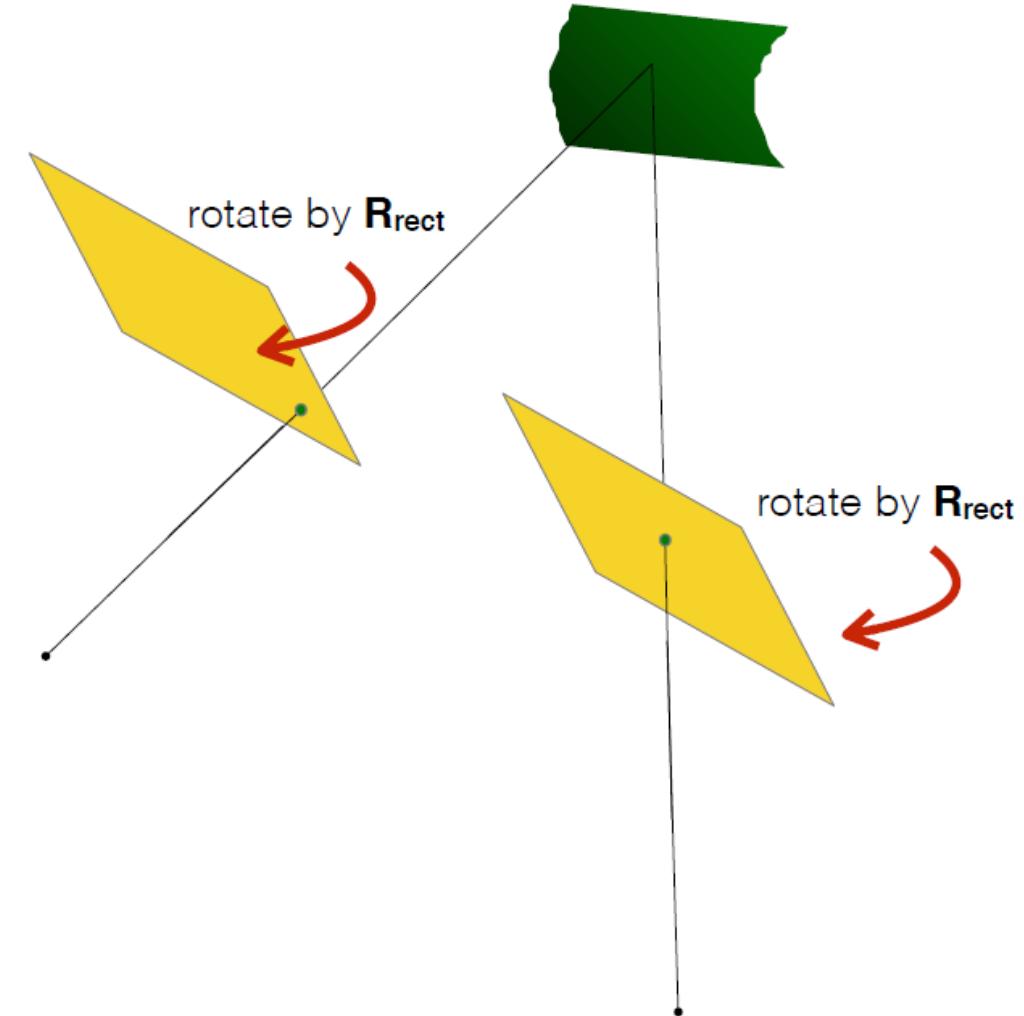
Stereo image rectification

1. Compute E to get R
2. Rotate right image by R
3. Rotate both images by R_{rect}
4. Scale both images by H



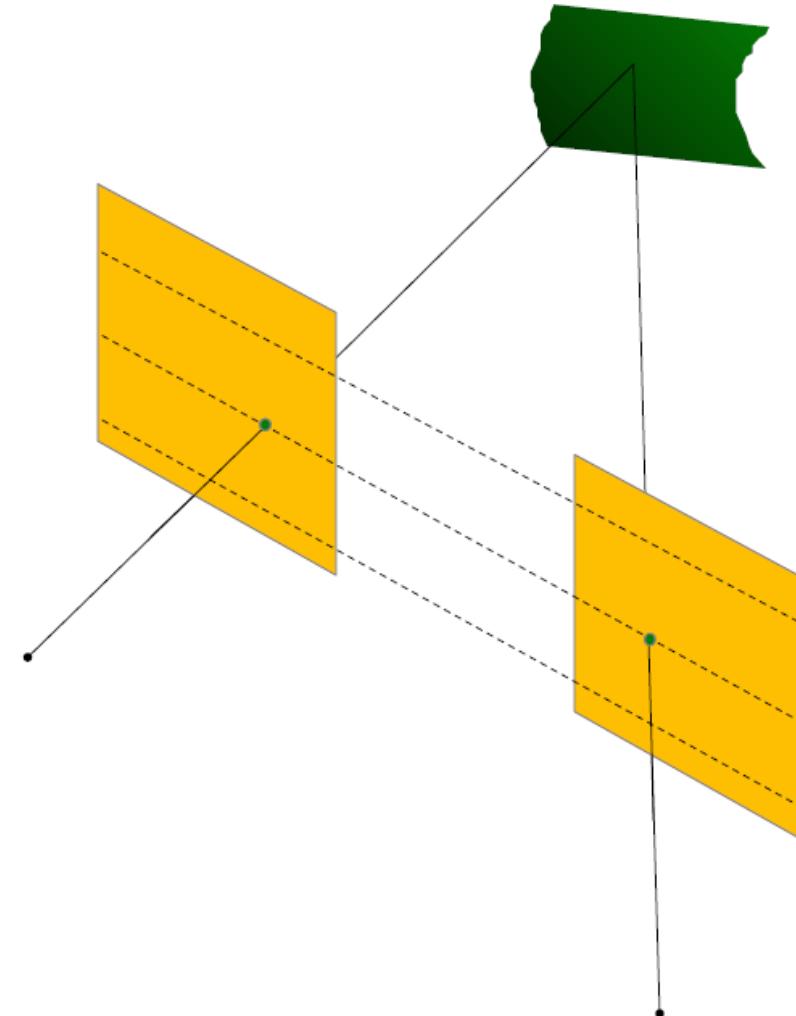
Stereo image rectification

1. Compute E to get R
2. Rotate right image by R
3. Rotate both images by R_{rect}
4. Scale both images by H



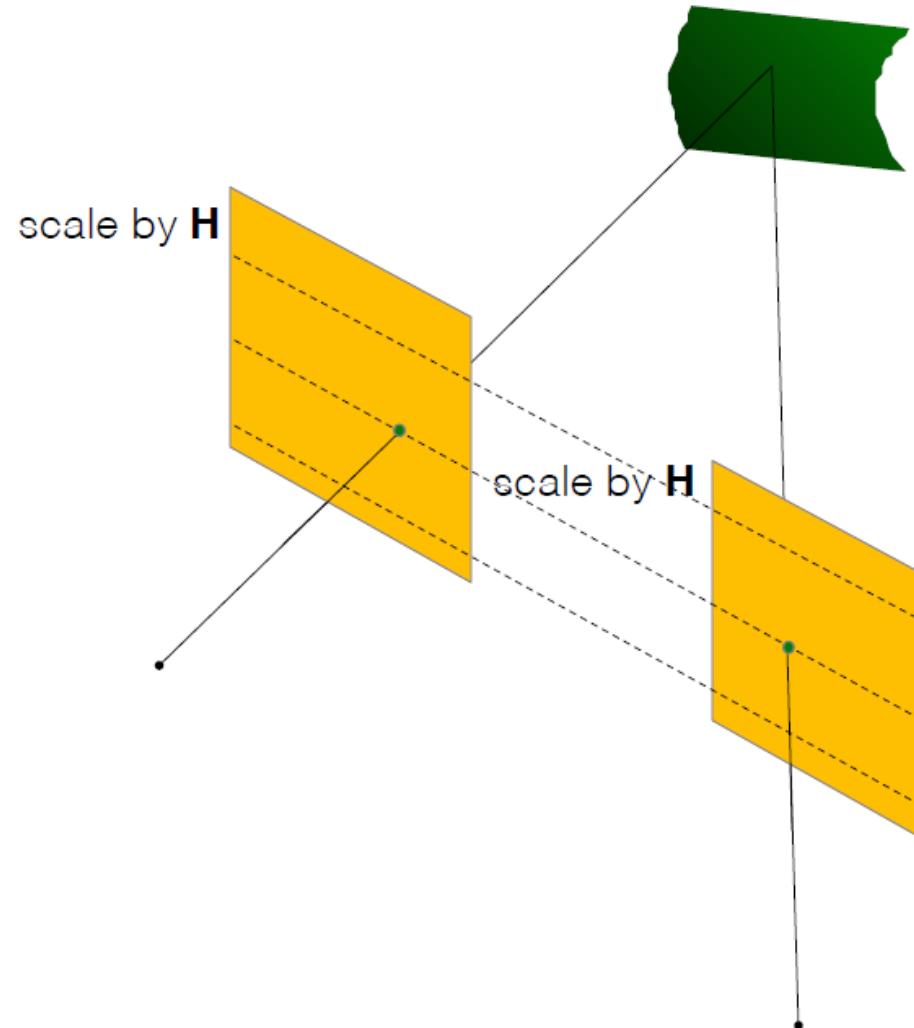
Stereo image rectification

1. Compute E to get R
2. Rotate right image by R
3. Rotate both images by R_{rect}
4. Scale both images by H



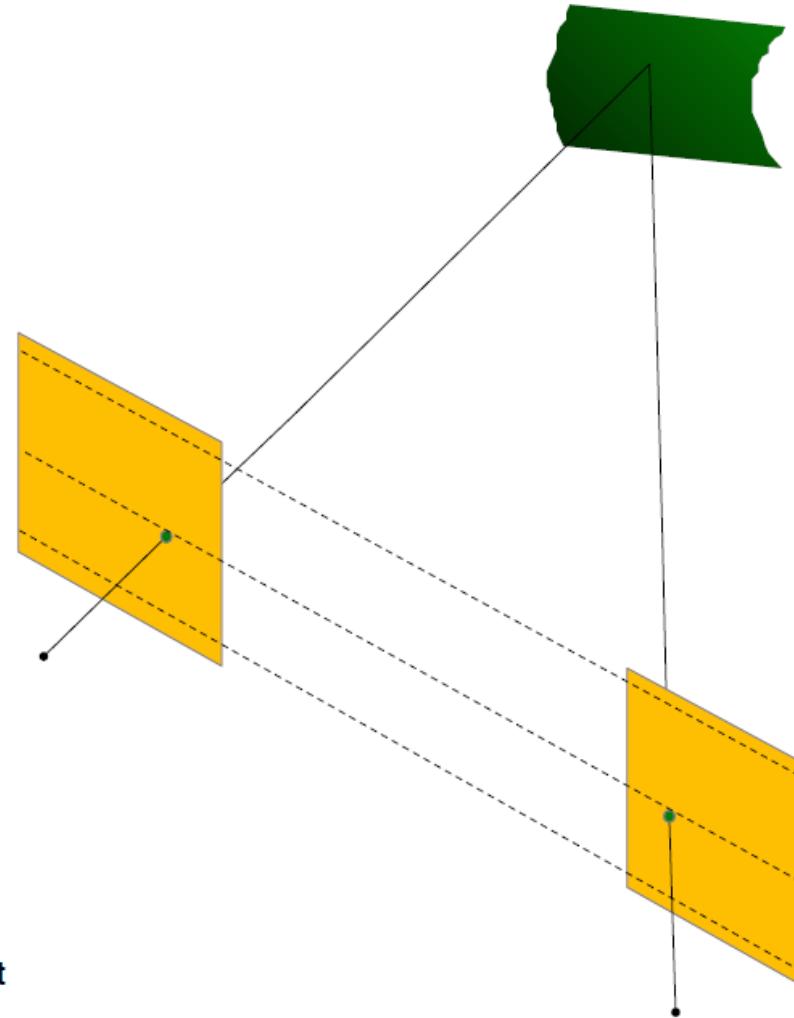
Stereo image rectification

1. Compute E to get R
2. Rotate right image by R
3. Rotate both images by R_{rect}
4. Scale both images by H

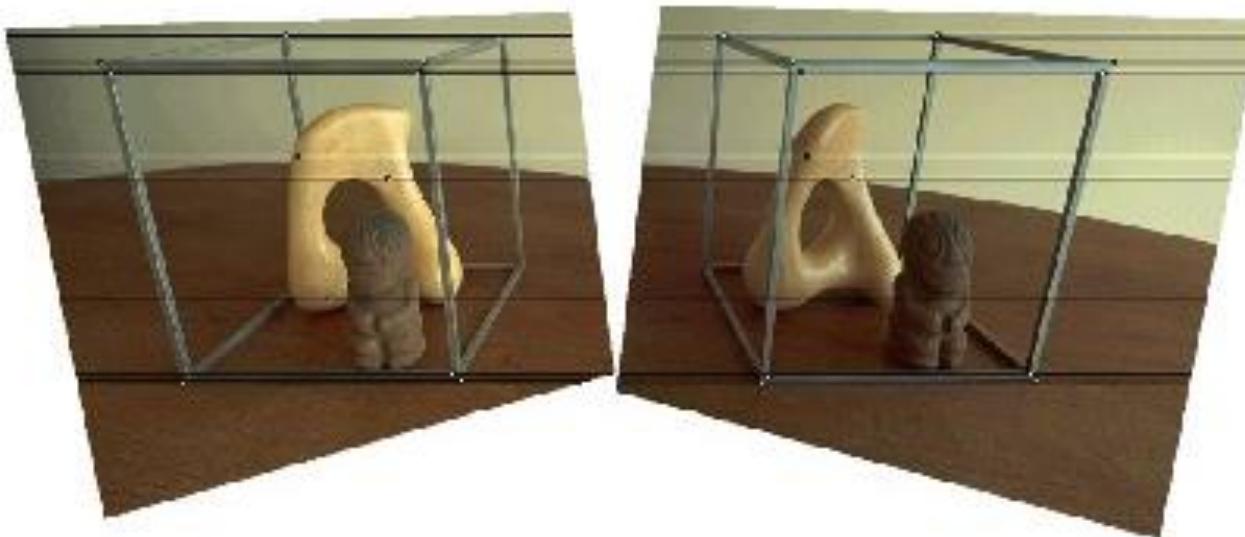


Stereo image rectification

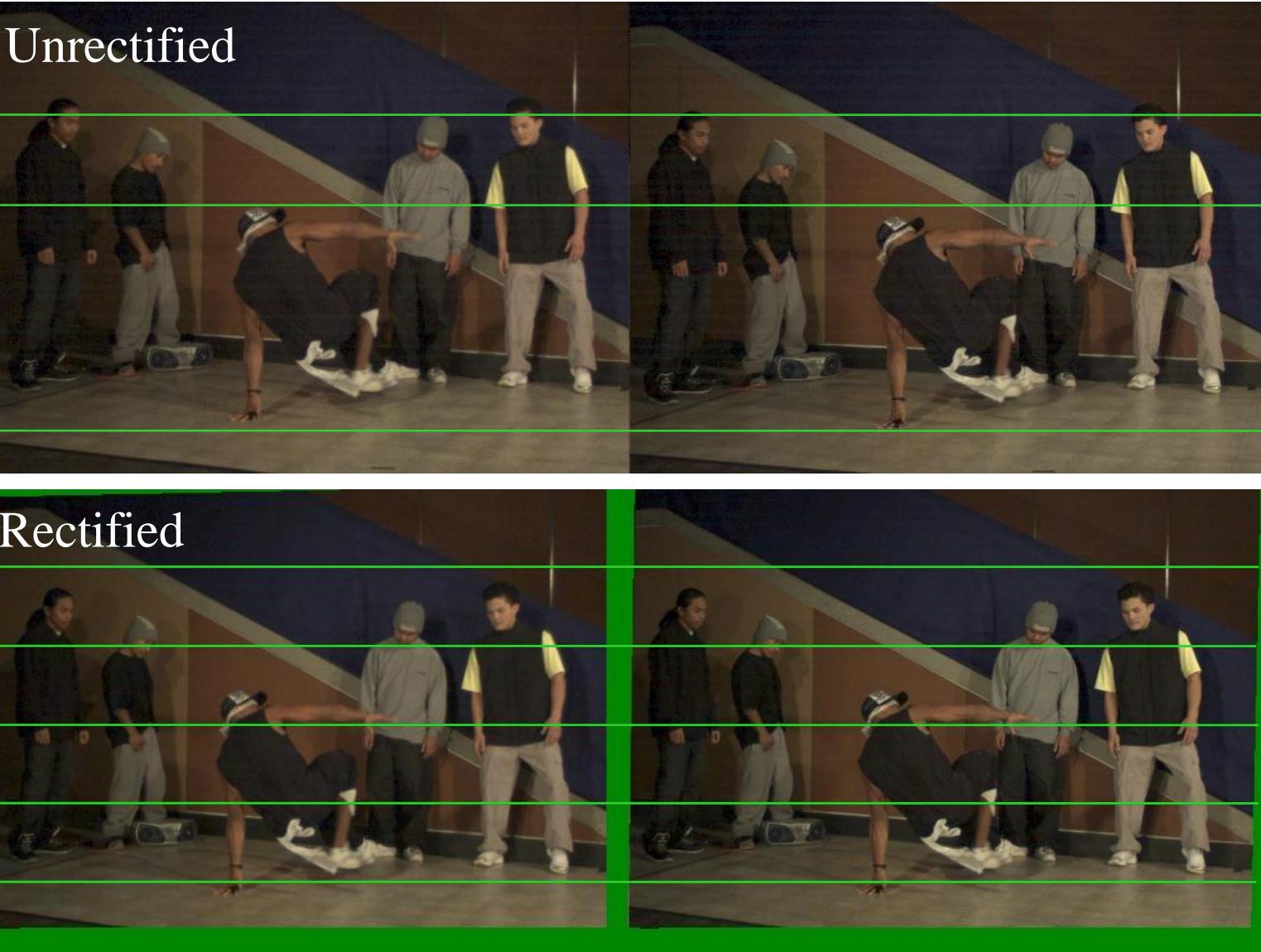
1. Compute E to get R
2. Rotate right image by R
3. Rotate both images by R_{rect}
4. Scale both images by H



Rectification example

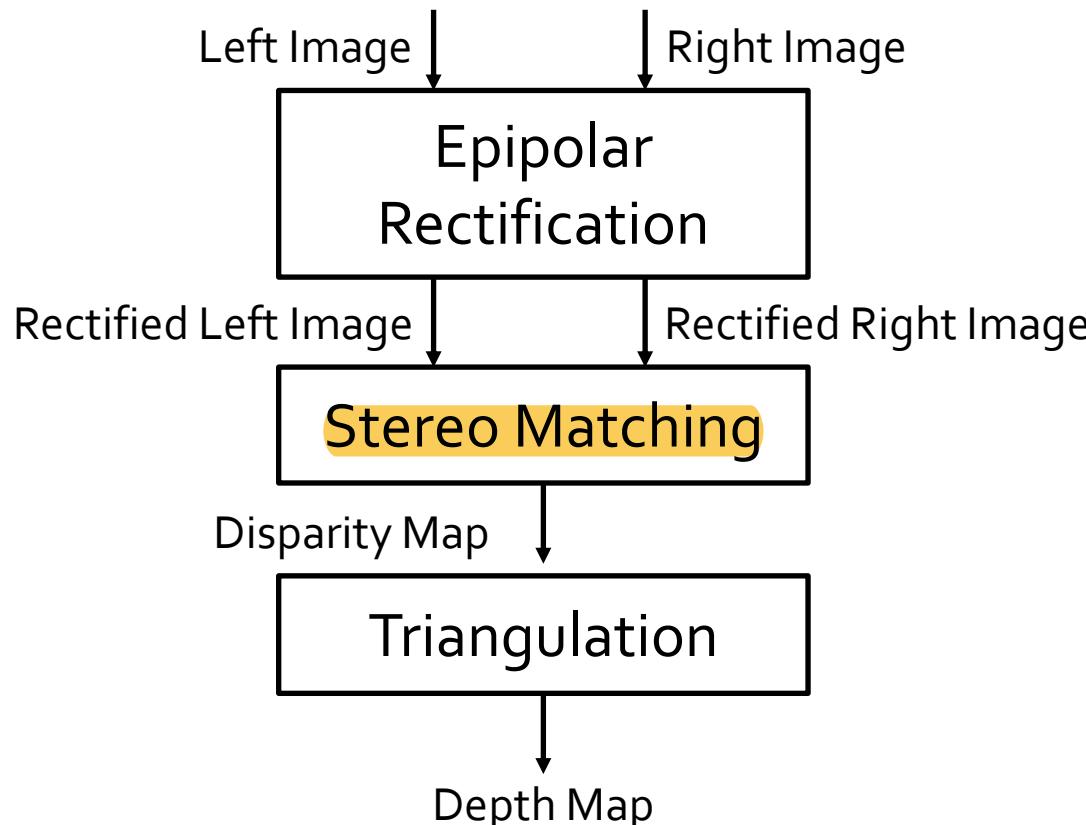


Another rectification example

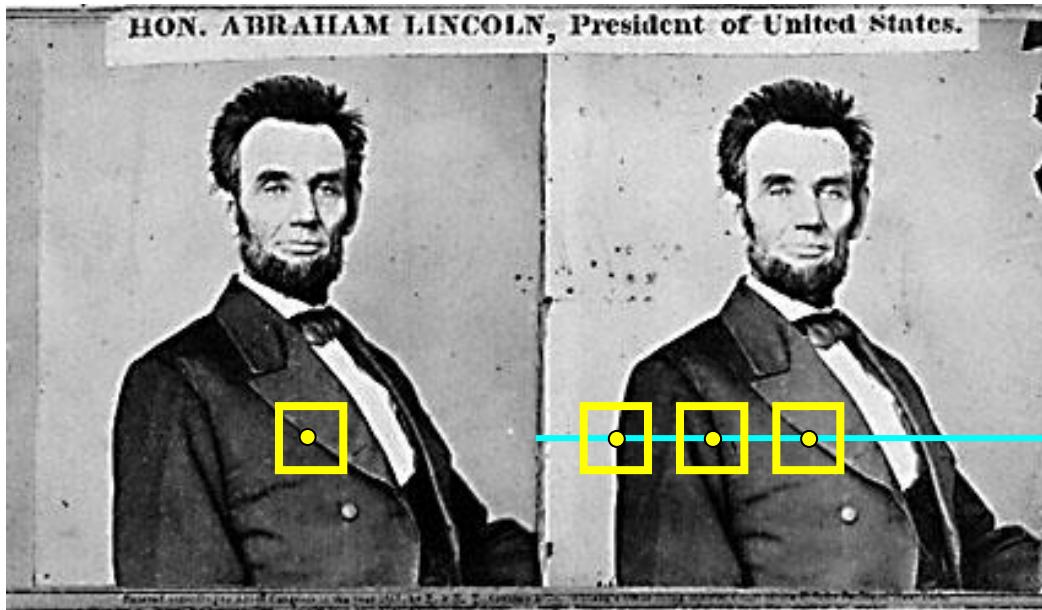


Stereo Matching

- Finding the corresponding points in epipolar line

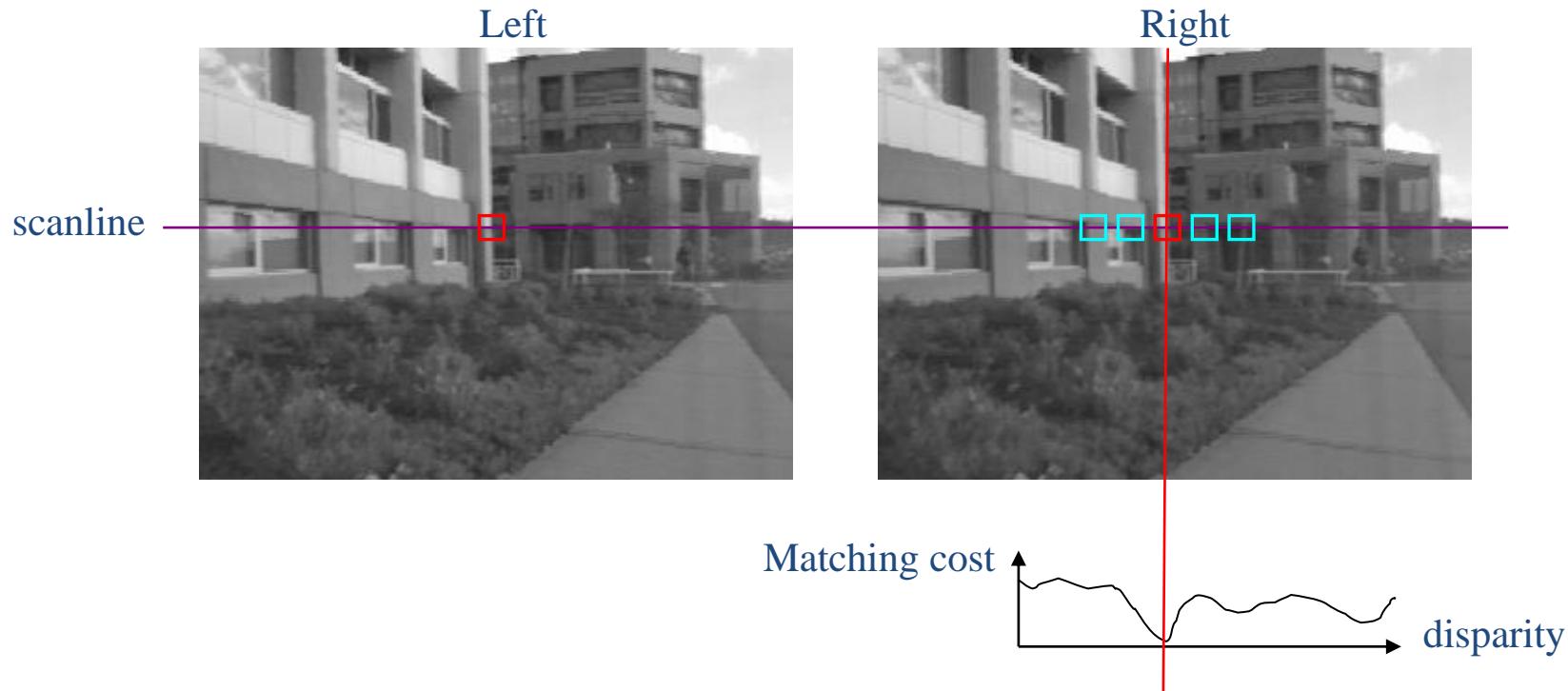


Basic stereo matching algorithm



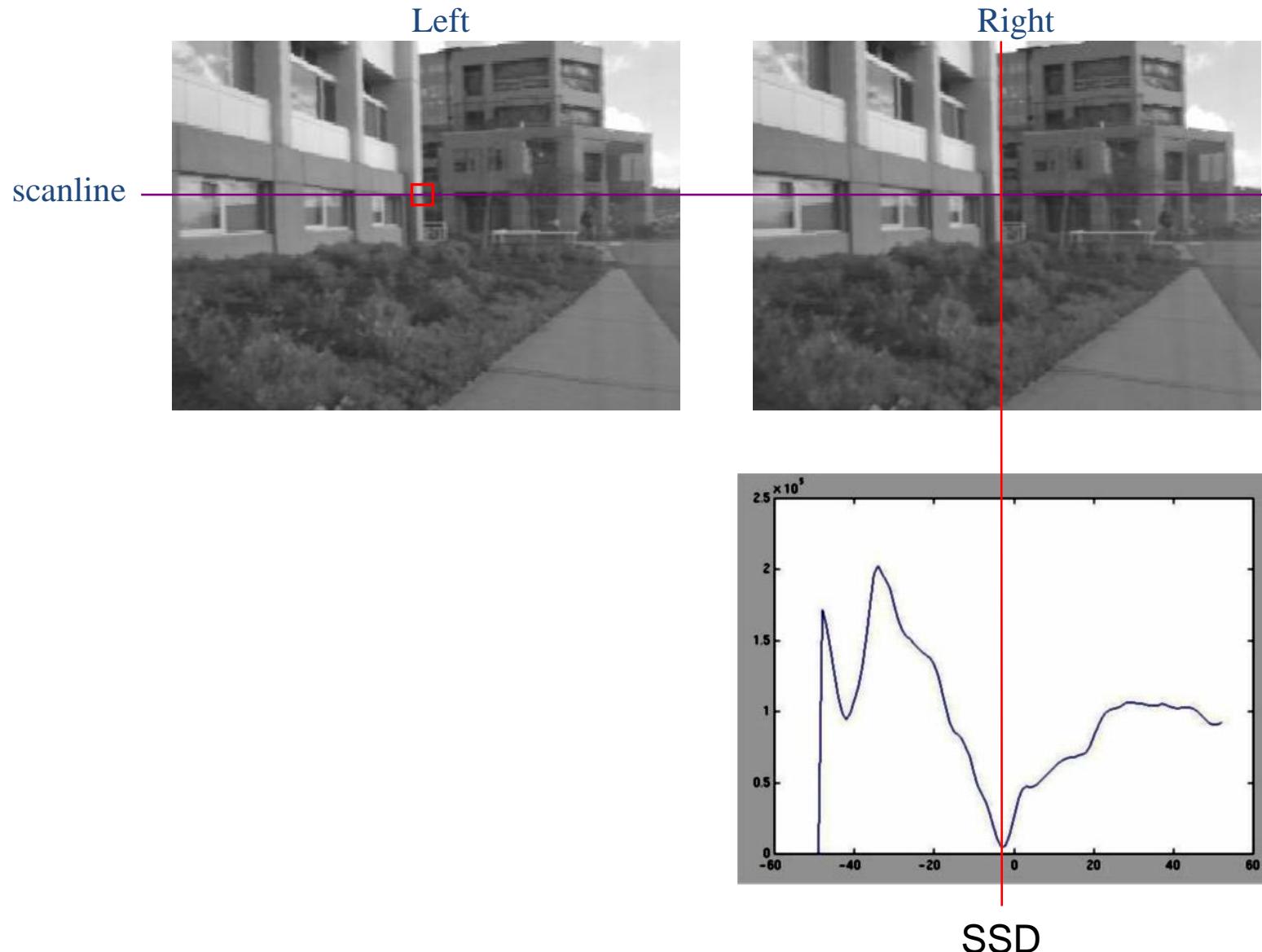
- If necessary, rectify the two stereo images to transform epipolar lines into scanlines
- For each pixel in the first image
 - Find corresponding epipolar line in the right image
 - Examine all pixels on the epipolar line and pick the best match

Correspondence search

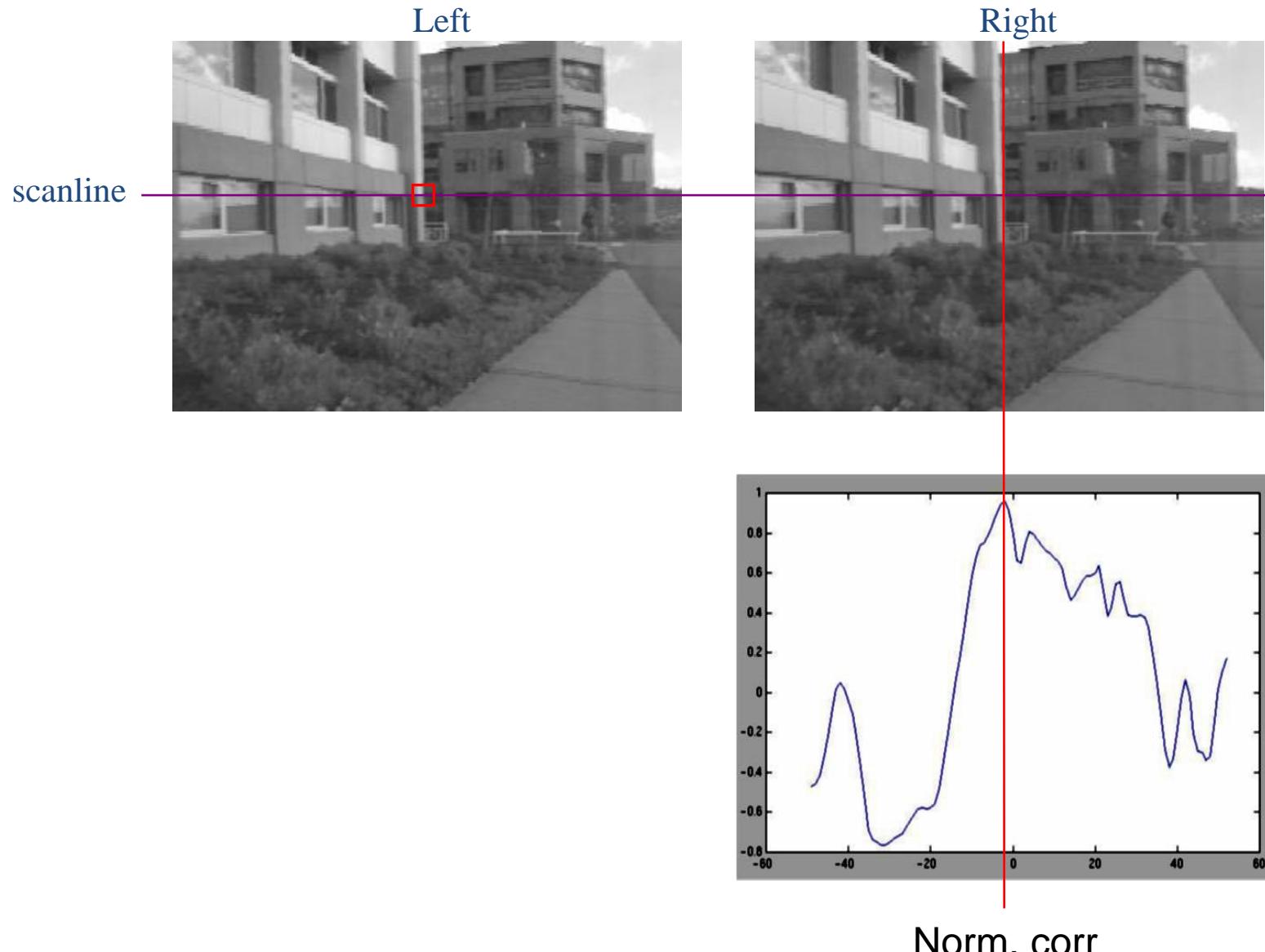


- Slide a window along the right scanline and compare contents of that window with the reference window in the left image
- Matching cost: SSD or normalized correlation

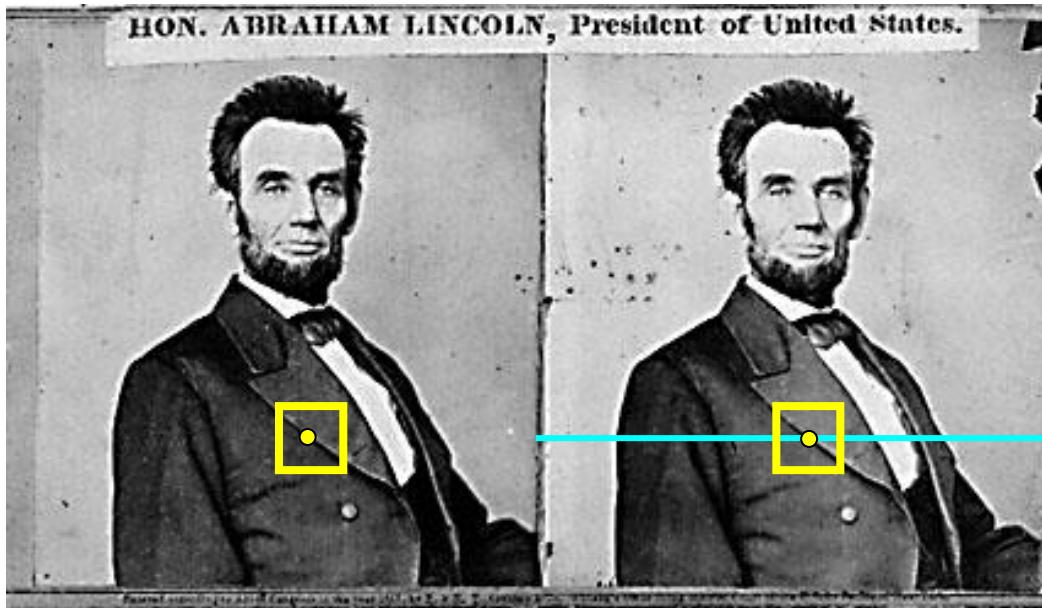
Correspondence search



Correspondence search



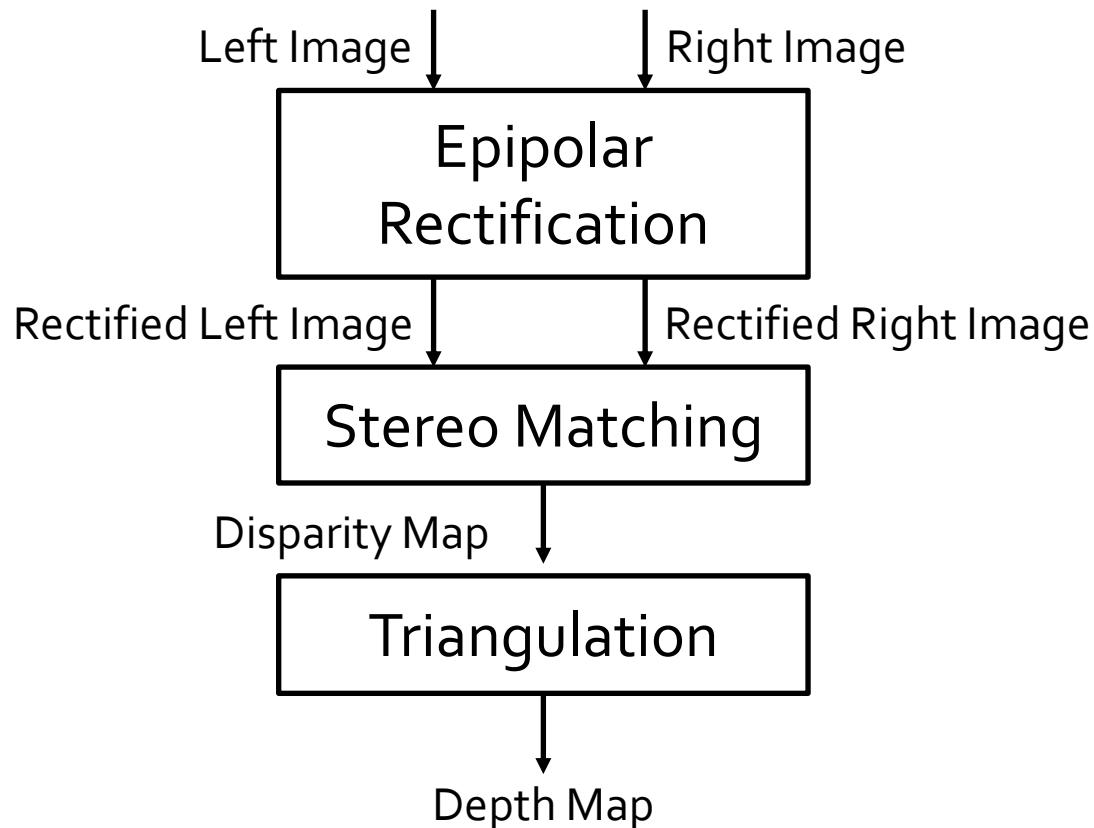
Basic stereo matching algorithm



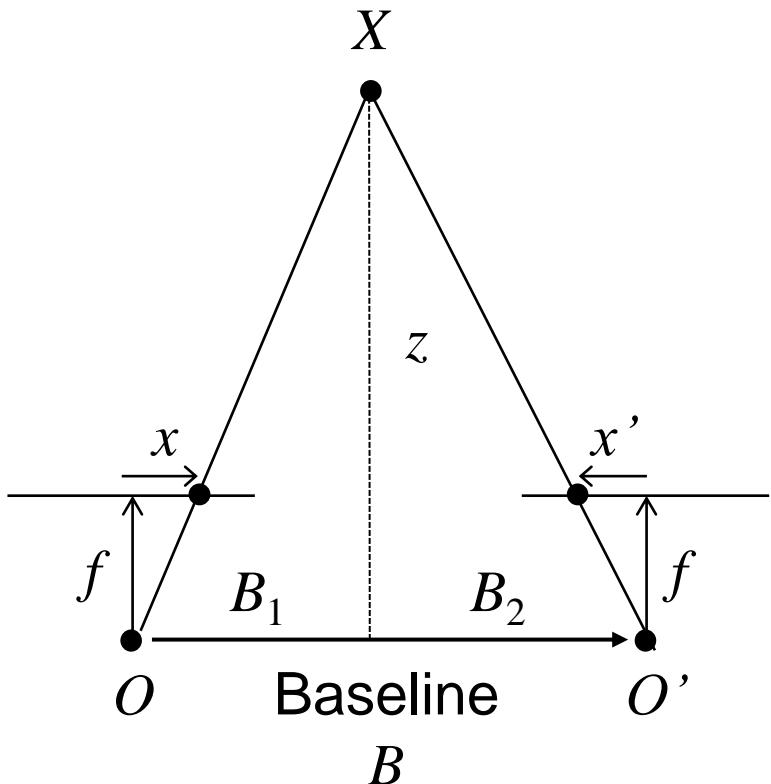
- If necessary, rectify the two stereo images to transform epipolar lines into scanlines
- For each pixel x in the first image
 - Find corresponding epipolar scanline in the right image
 - Examine all pixels on the scanline and pick the best match x'
 - Triangulate the matches to get depth information

Depth from disparity

- Performing triangulation



Depth from disparity



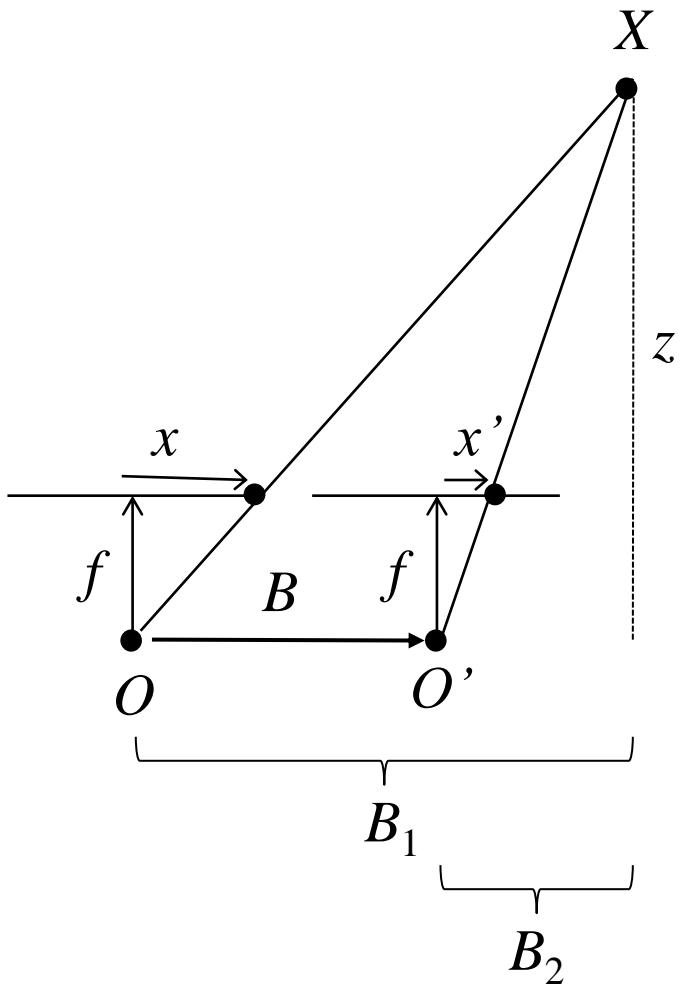
$$\frac{x}{f} = \frac{B_1}{z} \quad \frac{-x'}{f} = \frac{B_2}{z}$$

$$\frac{x - x'}{f} = \frac{B_1 + B_2}{z}$$

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

Disparity is inversely proportional to depth!

Depth from disparity

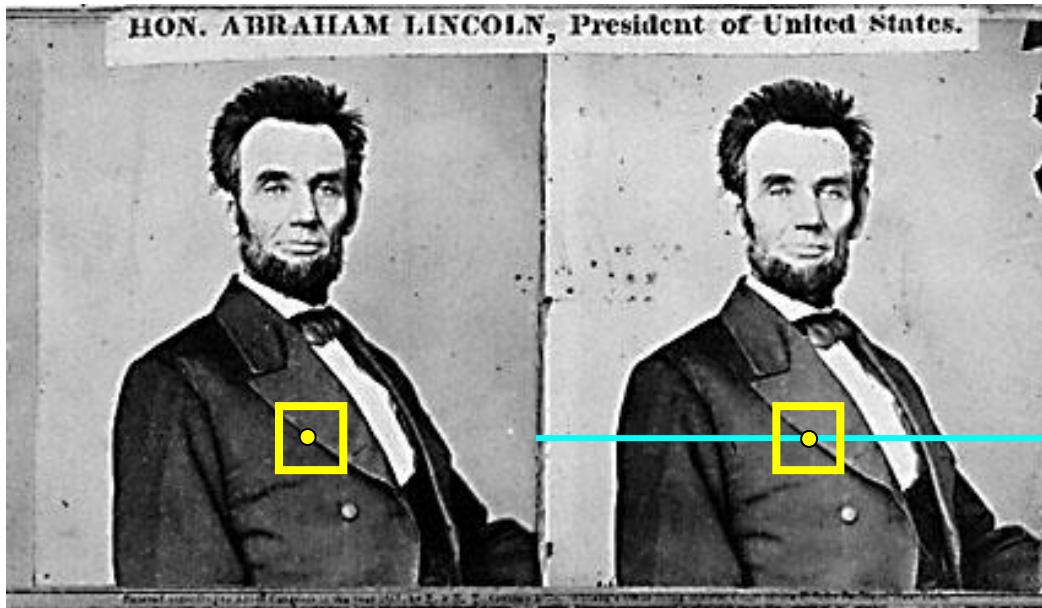


$$\frac{x}{f} = \frac{B_1}{z} \quad \frac{x'}{f} = \frac{B_2}{z}$$

$$\frac{x - x'}{f} = \frac{B_1 - B_2}{z}$$

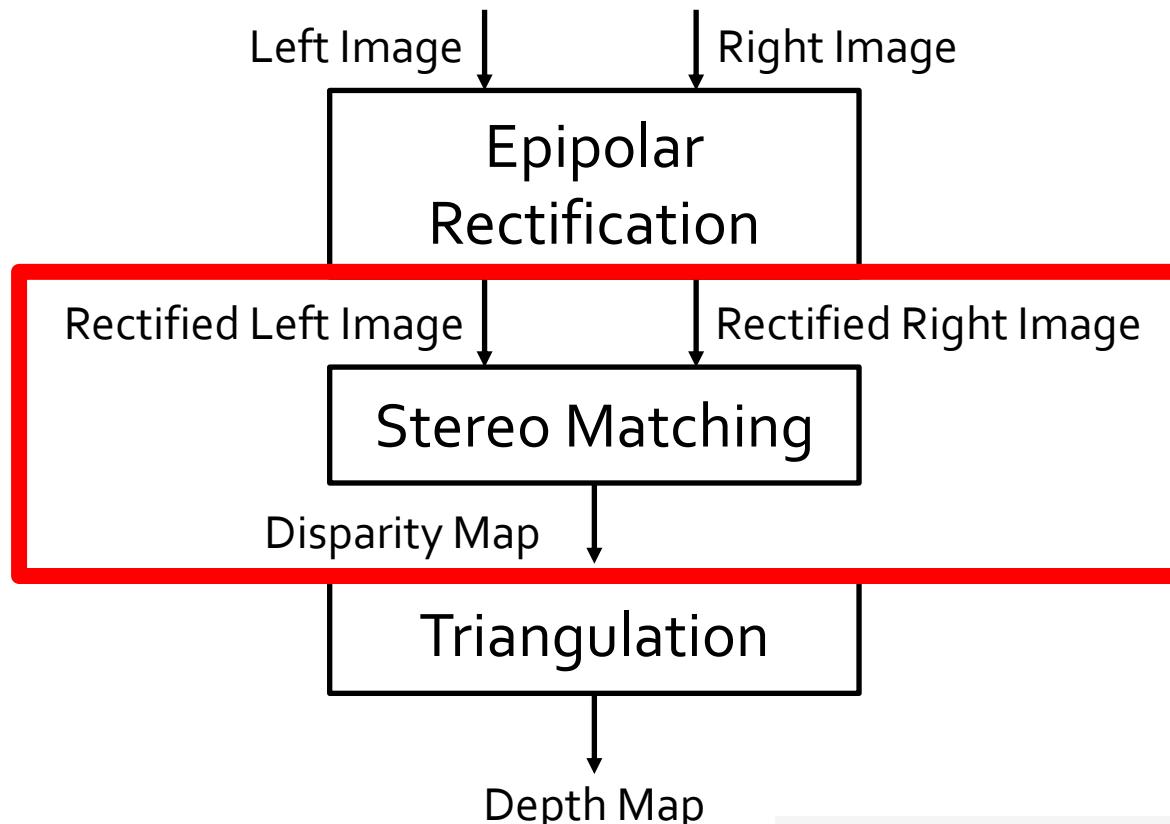
$$disparity = x - x' = \frac{B \cdot f}{z}$$

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

So the key point is..



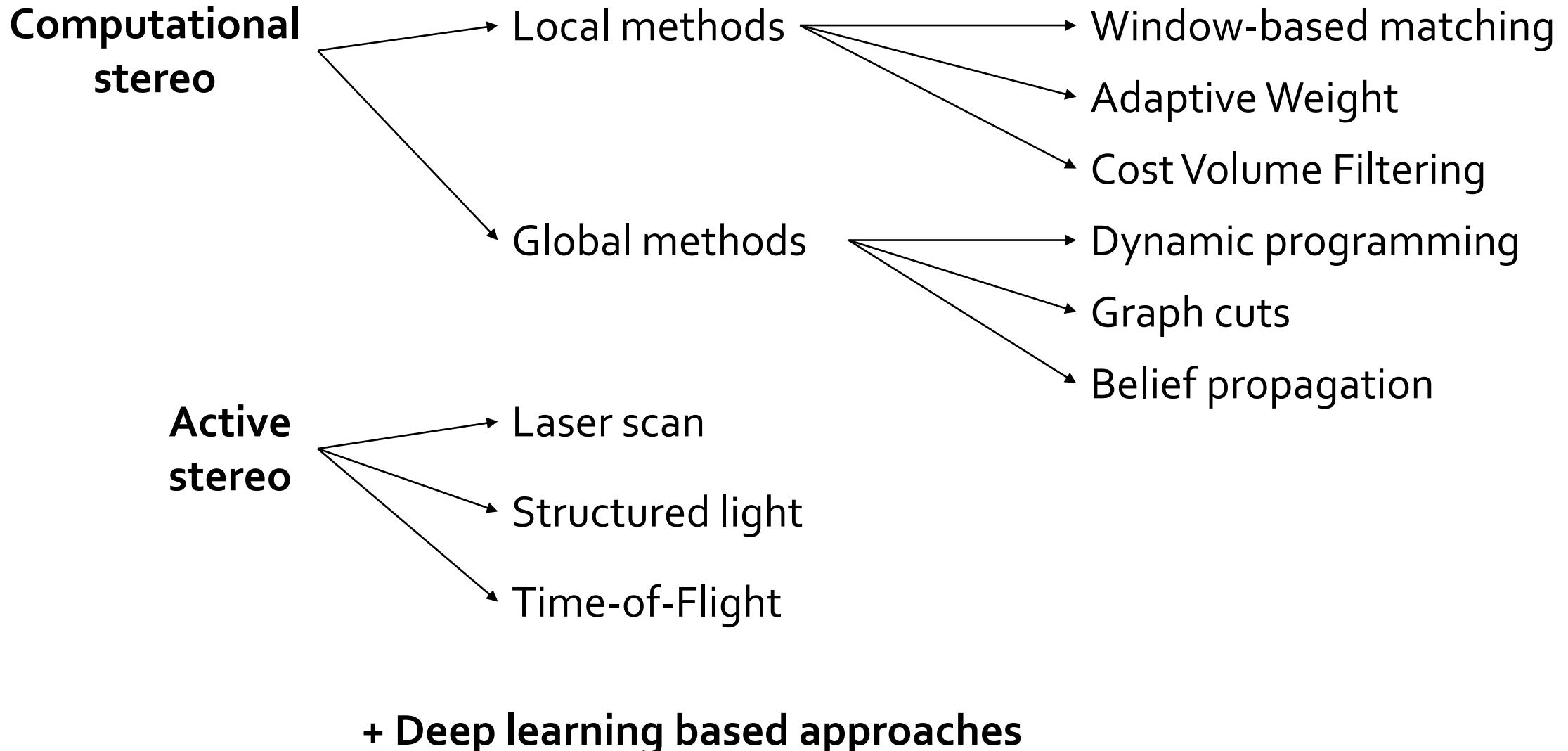
≡ Google Scholar

stereo matching

Articles

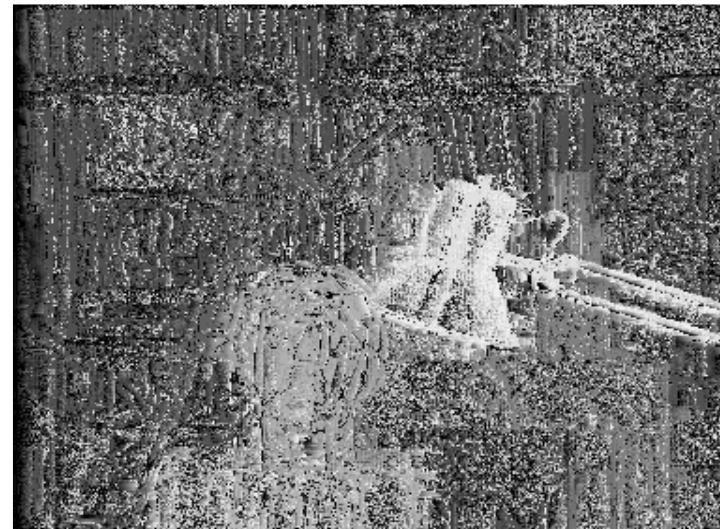
About 585,000 results (0.03 sec)

Taxonomy of stereo matching



A naïve stereo algorithm

- For each pixel x of the left image:
 - Compare color of x against the color of each pixel on the same horizontal scanline in the right image.
 - Select the pixel of most similar color as matching point

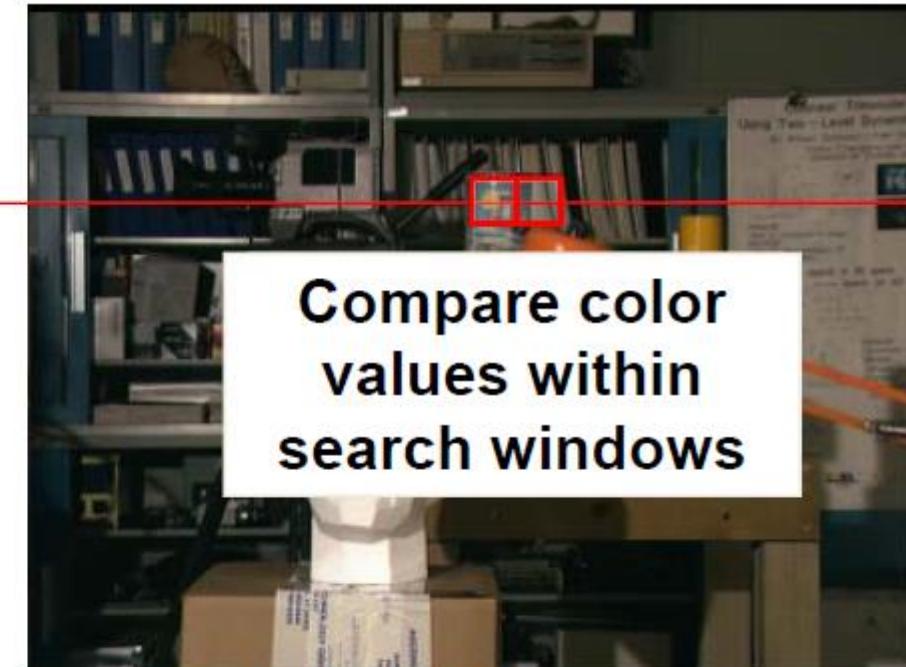


Window-based matching

- Instead of matching single pixel,
 - center a small window on a pixel and match the whole window in the right image.



(a) Left image



(b) Right image

Window-based matching

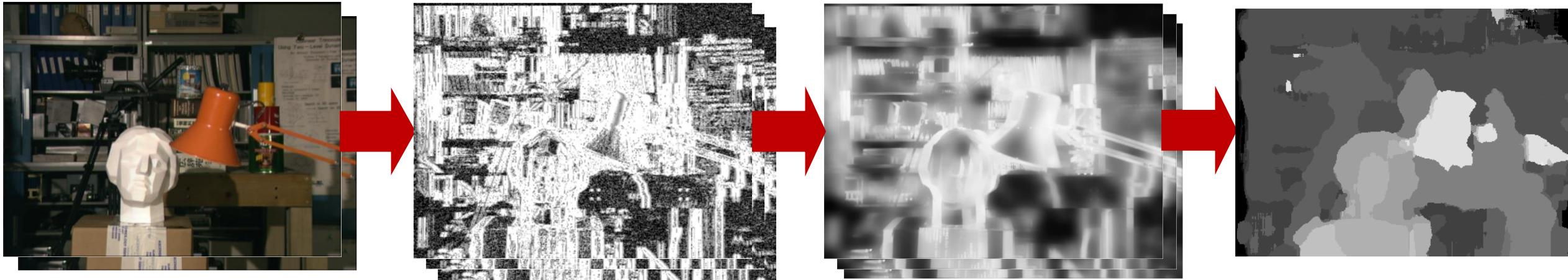
- In a formal way,
 - the disparity d_x of a pixel x in the left image is computed as

$$d_x = \underset{0 \leq d \leq d_{max}}{\operatorname{argmin}} \sum_{q \in W_x} c(q, q - d)$$

- Where
 - argmin returns the value at which the function takes a minimum
 - d_{max} is a parameter defining the maximum disparity (search range).
 - W_x is the set of all pixels inside the window centred on x
 - $c(q, q - d)$ is a function that computes the colour difference between a pixel q of the left image and a pixel $q - d$ of the right image (e.g. Sum of absolute difference in RGB values)

Let's split this equation into three steps!

A common pipeline for local methods



Matching cost
computation

$$c(q, q - d)$$

Cost
aggregation

$$\sum_{q \in W_x} c(q, q - d)$$

Disparity
computation

$$d_x = \operatorname{argmin}_{0 \leq d \leq d_{max}} \sum_{q \in W_x} c(q, q - d)$$

Effect of window size



$W = 3$

- Smaller window
- + More detail
- More noise



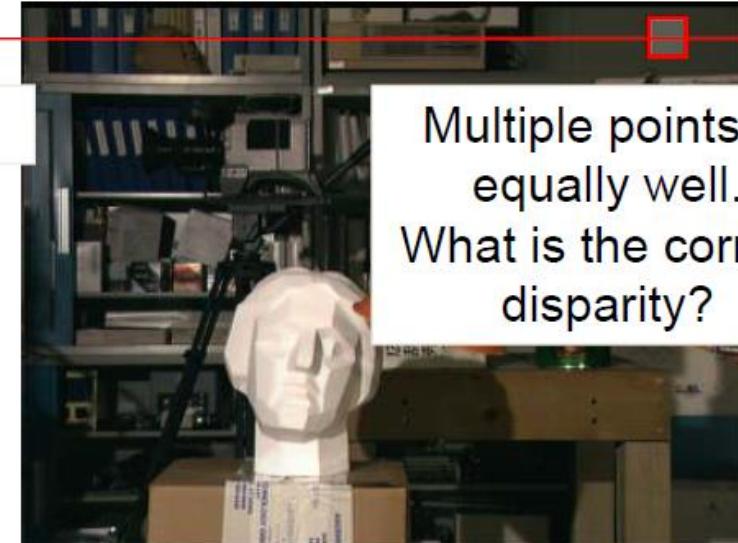
$W = 21$

- Larger window
- + Smoother disparity maps
- Less detail

Problem of Untextured Regions



(a) Left image



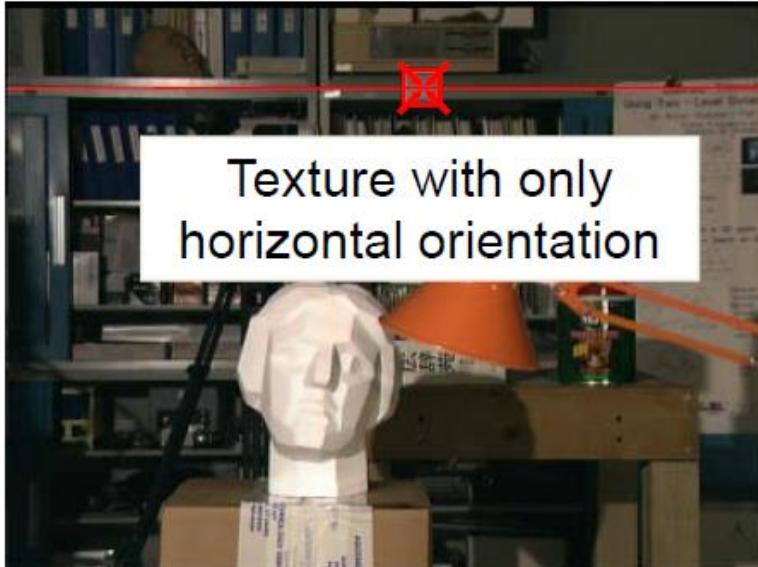
(b) Right image

Untextured region

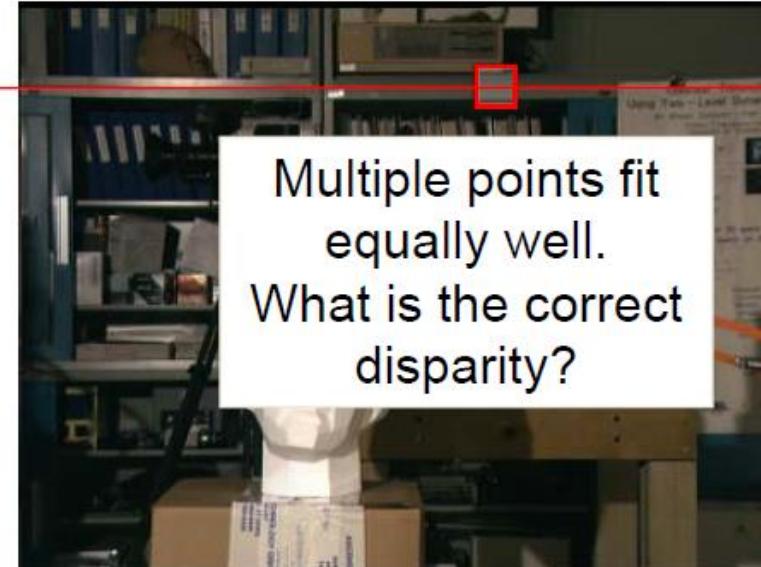
Multiple points fit
equally well.
What is the correct
disparity?

- There has to be a certain amount of colour variation inside the window

Aperture Problem



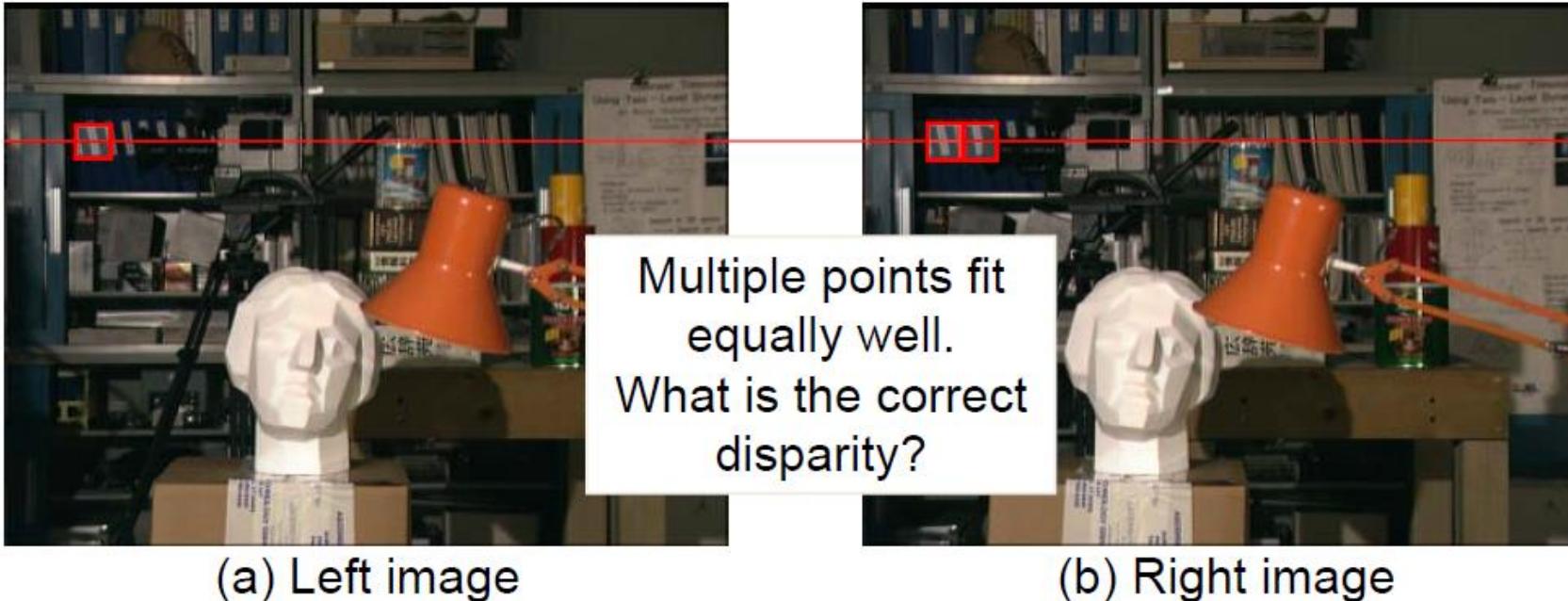
(a) Left image



(b) Right image

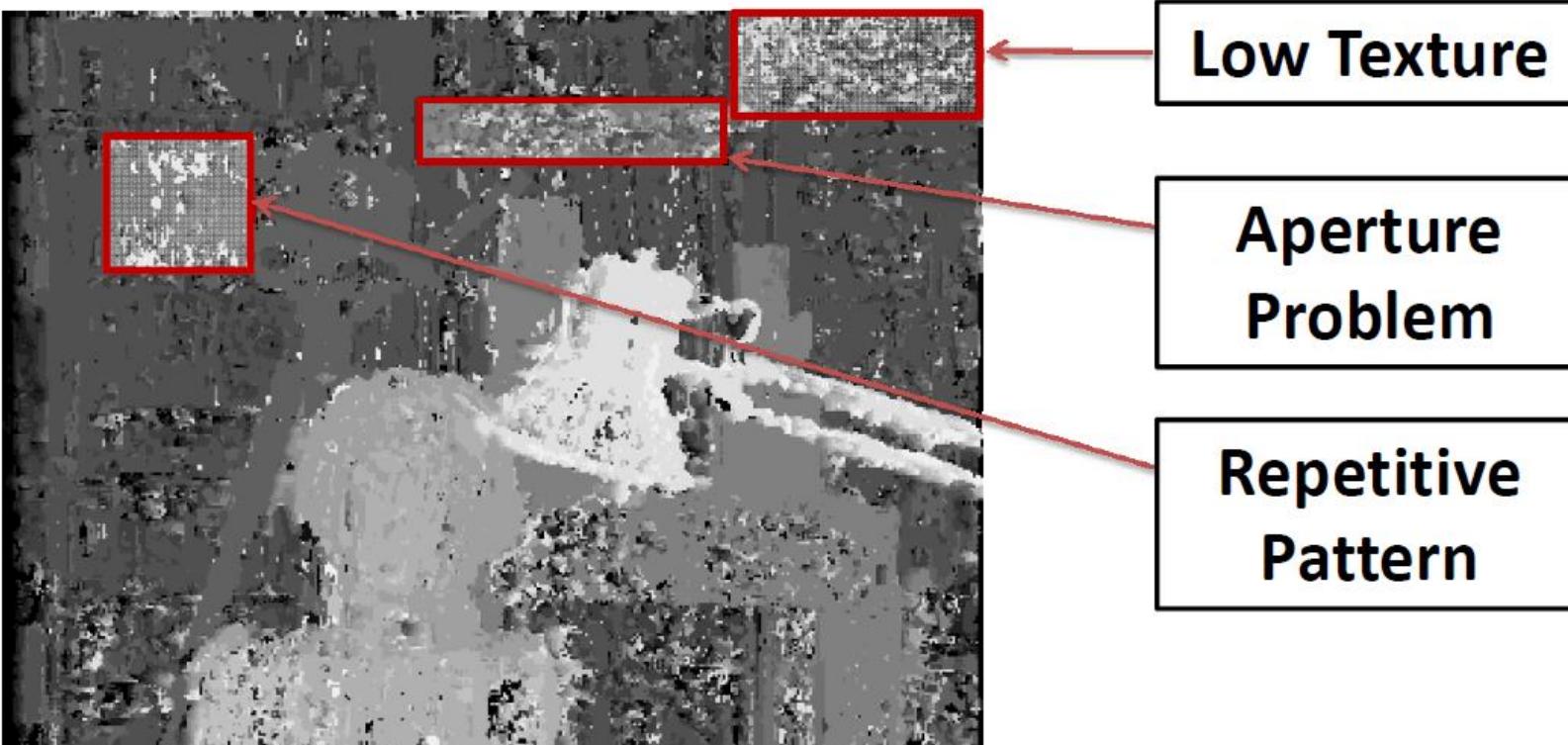
- There needs to be a certain amount of texture with vertical orientation

Problem of Repetitive Patterns



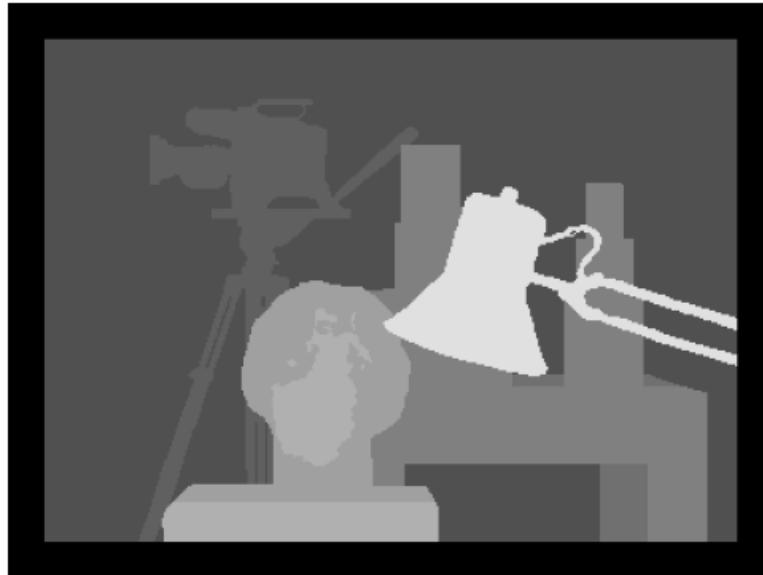
- There needs to be a certain amount of repetitive texture

Effect of these problems



Window size = 3x3 pixels

Foreground fattening problem



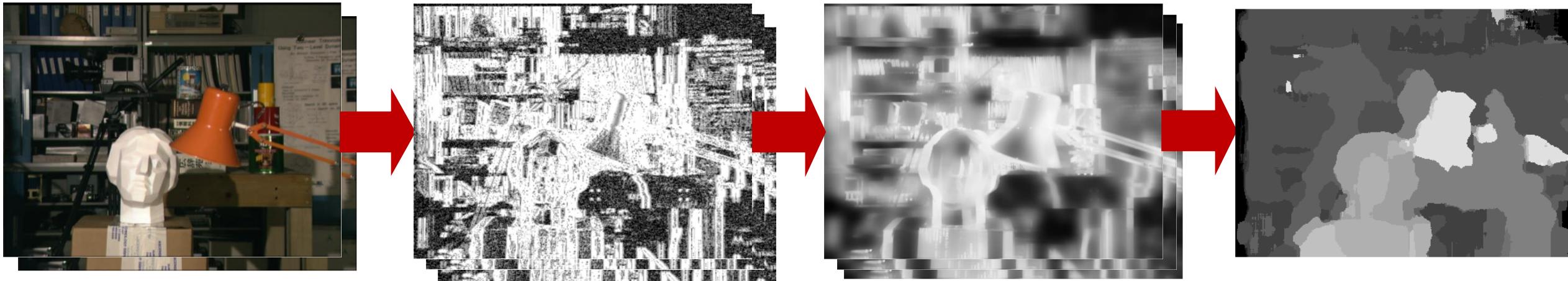
Ground Truth Disparities



Window size = 21x21 pixels

- Foreground objects are clearly enlarged when using large kernels (windows)

Foreground flattening problem: why?



Matching cost computation

$$c(q, q - d)$$

Cost aggregation

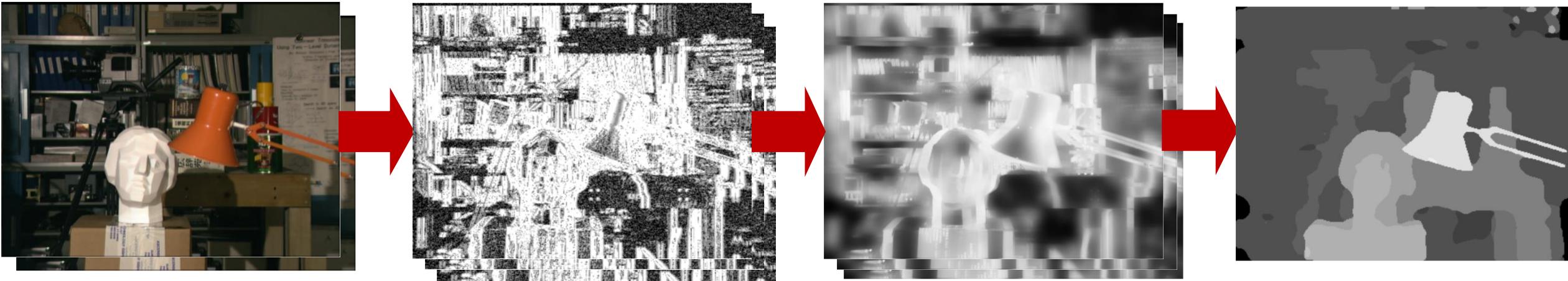
$$\sum_{q \in W_x} c(q, q - d)$$

Disparity computation

$$d_x = \operatorname{argmin}_{0 \leq d \leq d_{max}} \sum_{q \in W_x} c(q, q - d)$$

All pixels in W_x have the same influence on the aggregated costs

Adaptive support weight



Matching cost computation

Cost aggregation

Disparity computation

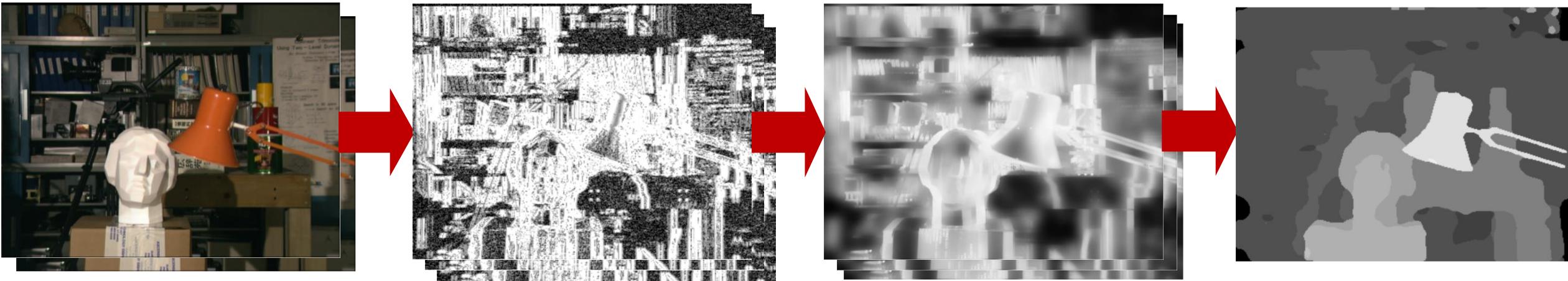
$$c(\mathbf{q}, \mathbf{q} - \mathbf{d})$$

$$\sum_{\mathbf{q} \in W_x} w(\mathbf{x}, \mathbf{q}) c(\mathbf{q}, \mathbf{q} - \mathbf{d})$$

$$d_x = \operatorname{argmin}_{0 \leq d \leq d_{max}} \sum_{\mathbf{q} \in W_x} c(\mathbf{q}, \mathbf{q} - \mathbf{d})$$

1. Assumption: Two points are likely to lie on the same disparity if they are similar in color
2. Only pixels that lie on the same disparity contribute to the aggregated matching costs
-> No foreground flattening

Adaptive support weight



Matching cost computation

Cost aggregation

Disparity computation

$$c(q, q - d)$$

$$\sum_{q \in W_x} w(x, q) c(q, q - d)$$

$$d_x = \operatorname{argmin}_{0 \leq d \leq d_{max}} \sum_{q \in W_x} c(q, q - d)$$

$$w(x, q) = \exp \left(- \left(\frac{\Delta c_{xq}}{\sigma_c} + \frac{\Delta s_{xq}}{\sigma_s} \right) \right)$$

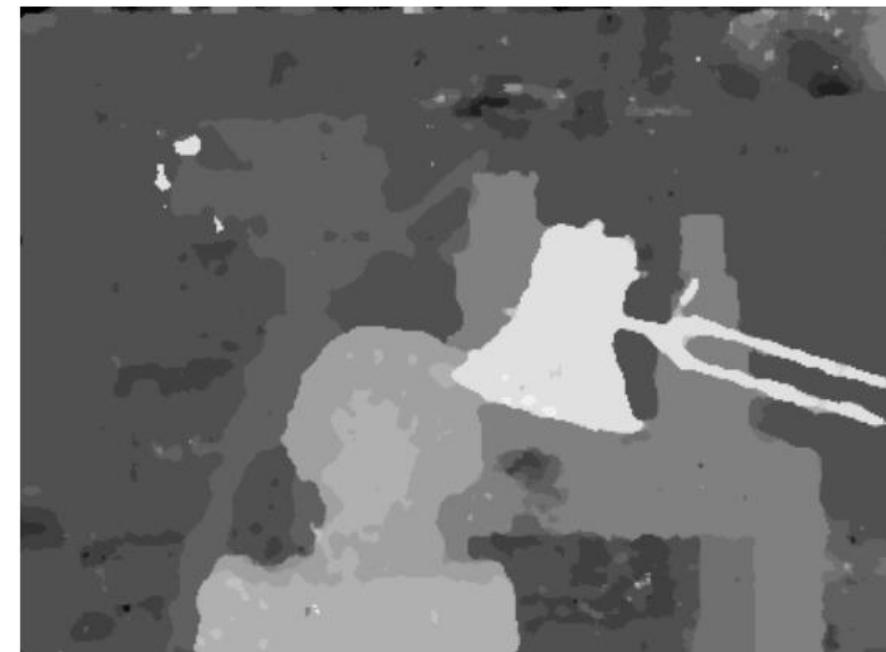
Δc_{xq} : colour distance between x and q

Δs_{xq} : spatial distance between x and q

Results



Window-based



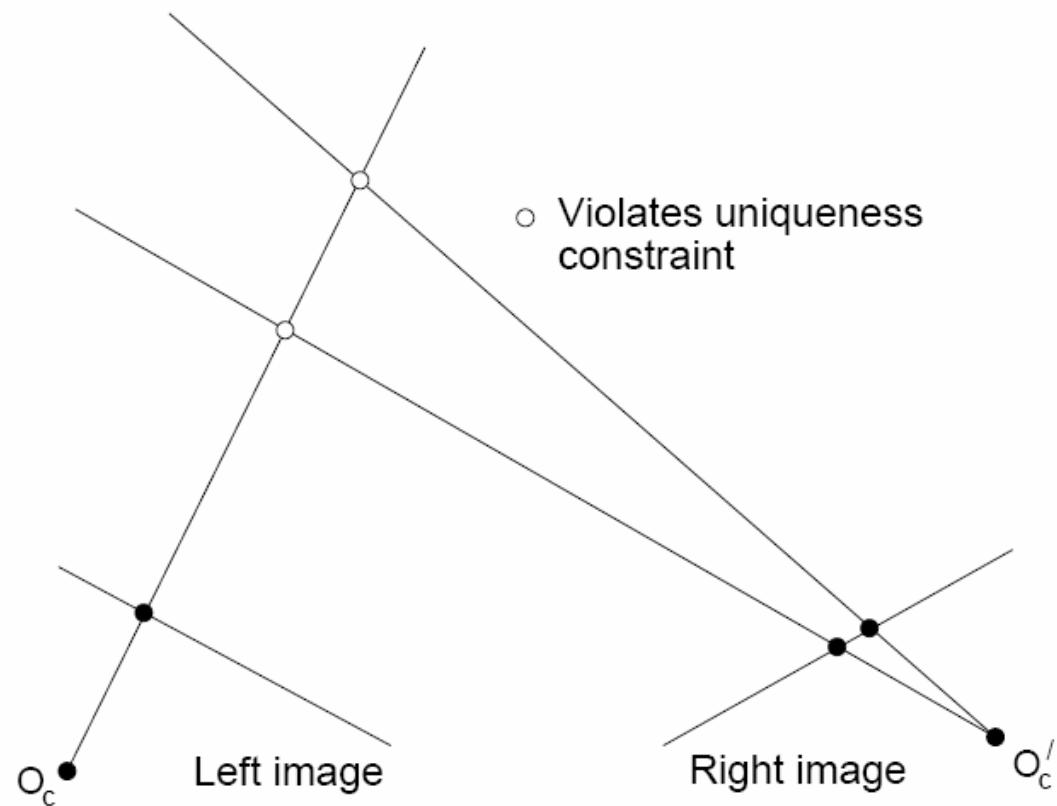
Adaptive support-weight

How can we improve window-based matching?

- The similarity constraint is local
 - Each reference window is matched independently
- Need to enforce non-local correspondence constraints

Non-local constraints

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



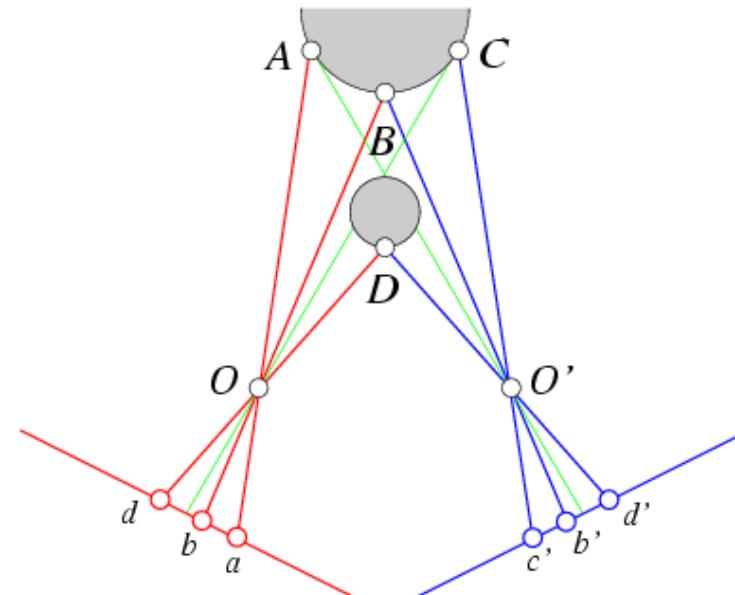
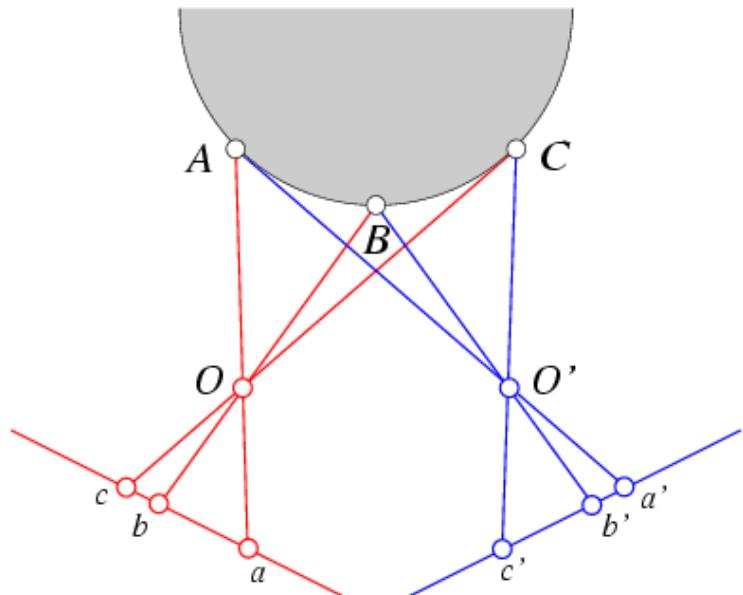
Non-local constraints

- **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



Ordering constraint doesn't hold

Non-local constraints

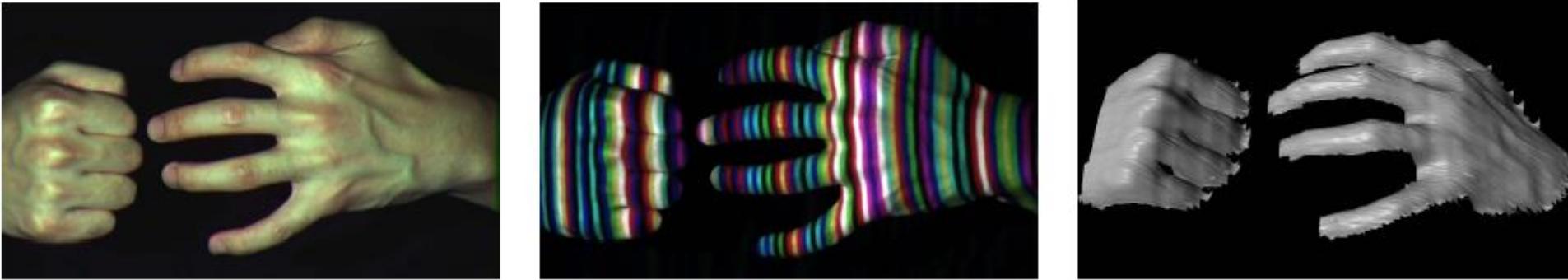
- **Uniqueness**
 - For any point in one image, there should be **at most one matching point** in the other image
- **Ordering**
 - **Corresponding points should be in the same order** in both views
- **Smoothness**
 - We expect **disparity values to change slowly** (for the most part)

Stereo datasets

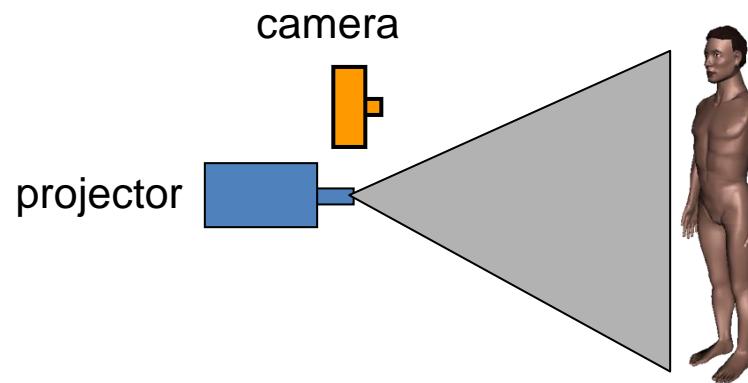
- Middlebury stereo datasets
- KITTI
- Synthetic data



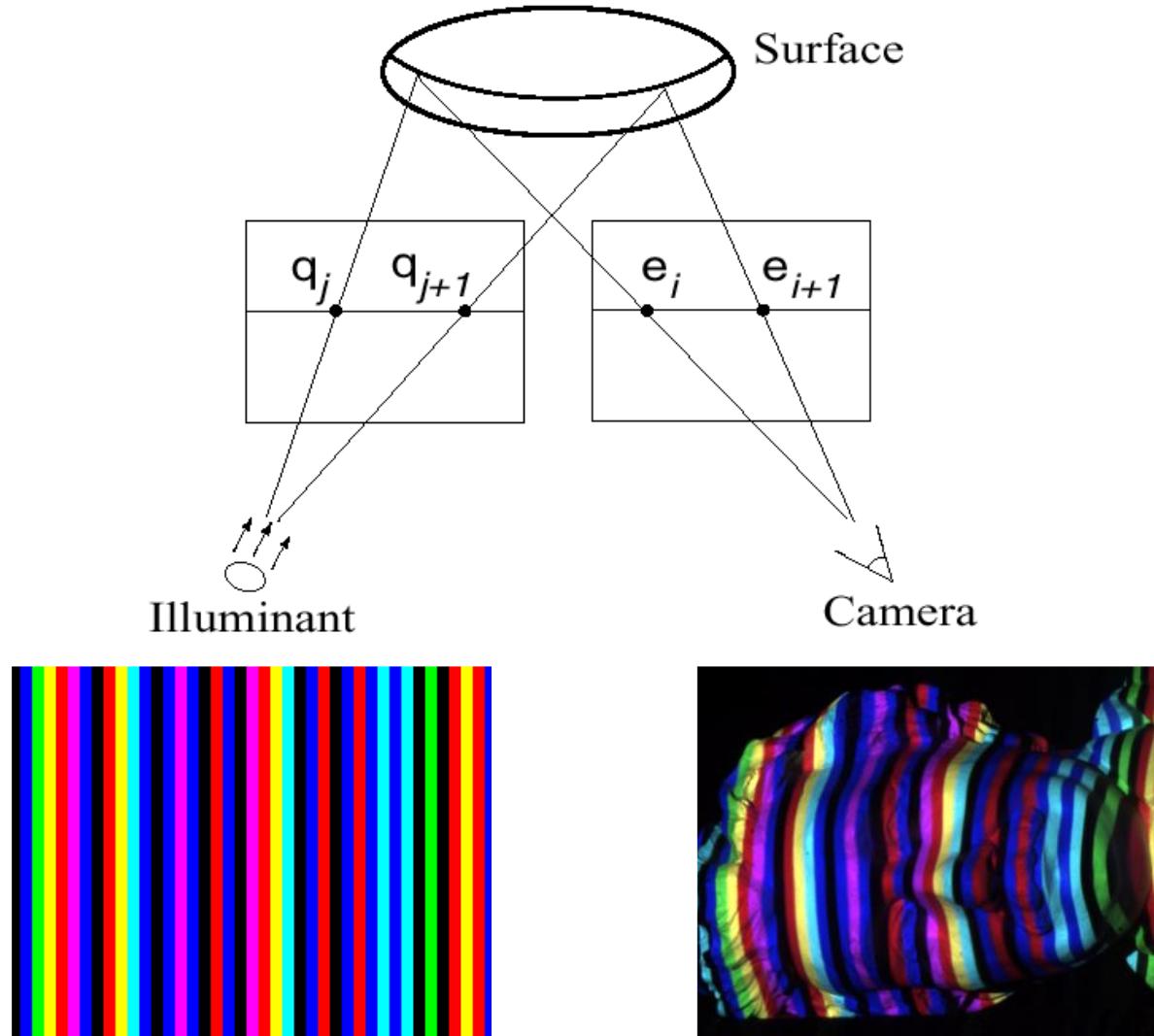
Active stereo with structured light



- Project “structured” light patterns onto the object
 - Simplifies the correspondence problem
 - Allows us to use only one camera

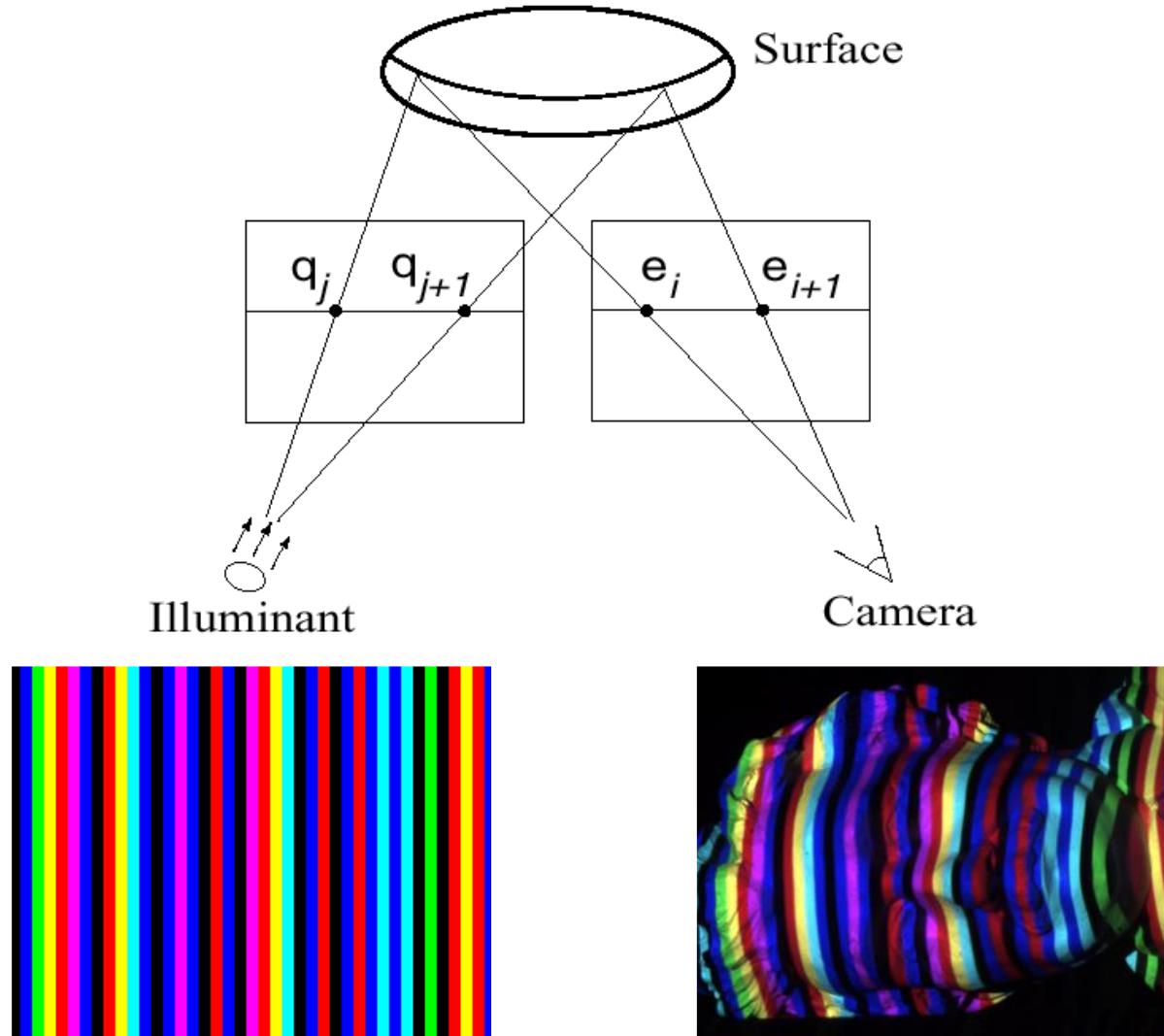


Active stereo with structured light



L. Zhang, B. Curless, and S. M. Seitz. [Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming](#). 3DPVT 2002

Active stereo with structured light



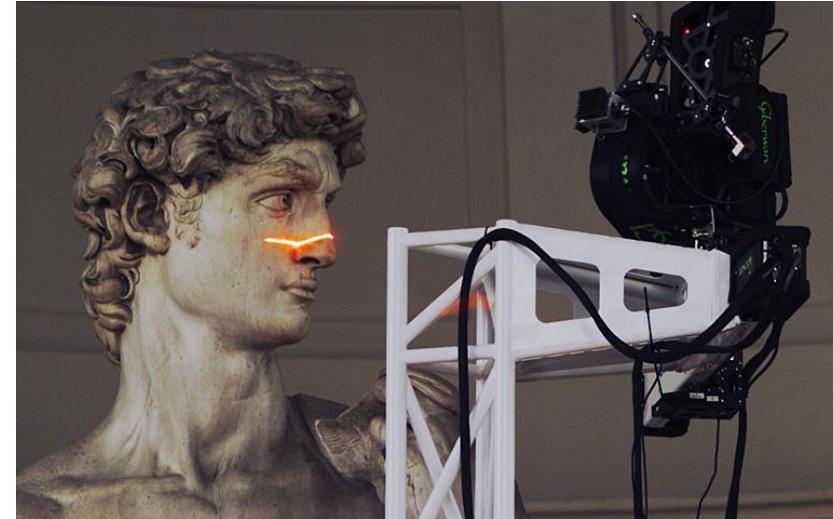
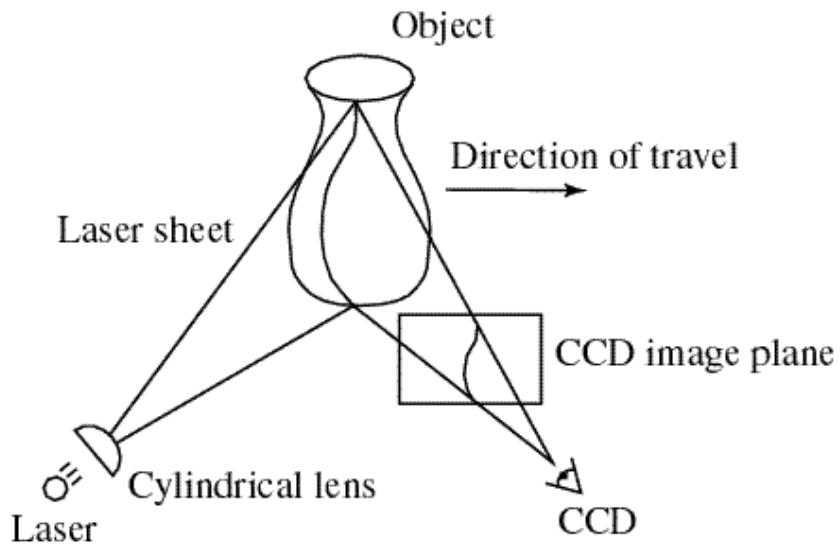
http://en.wikipedia.org/wiki/Structured-light_3D_scanner

Kinect: Structured infrared light



<http://bbzippo.wordpress.com/2010/11/28/kinect-in-infrared/>

Laser scanning



Digital Michelangelo Project
Levoy et al.

<http://graphics.stanford.edu/projects/mich/>

- **Optical triangulation**
 - Project a single stripe of laser light
 - Scan it across the surface of the object
 - This is a very precise version of structured light scanning

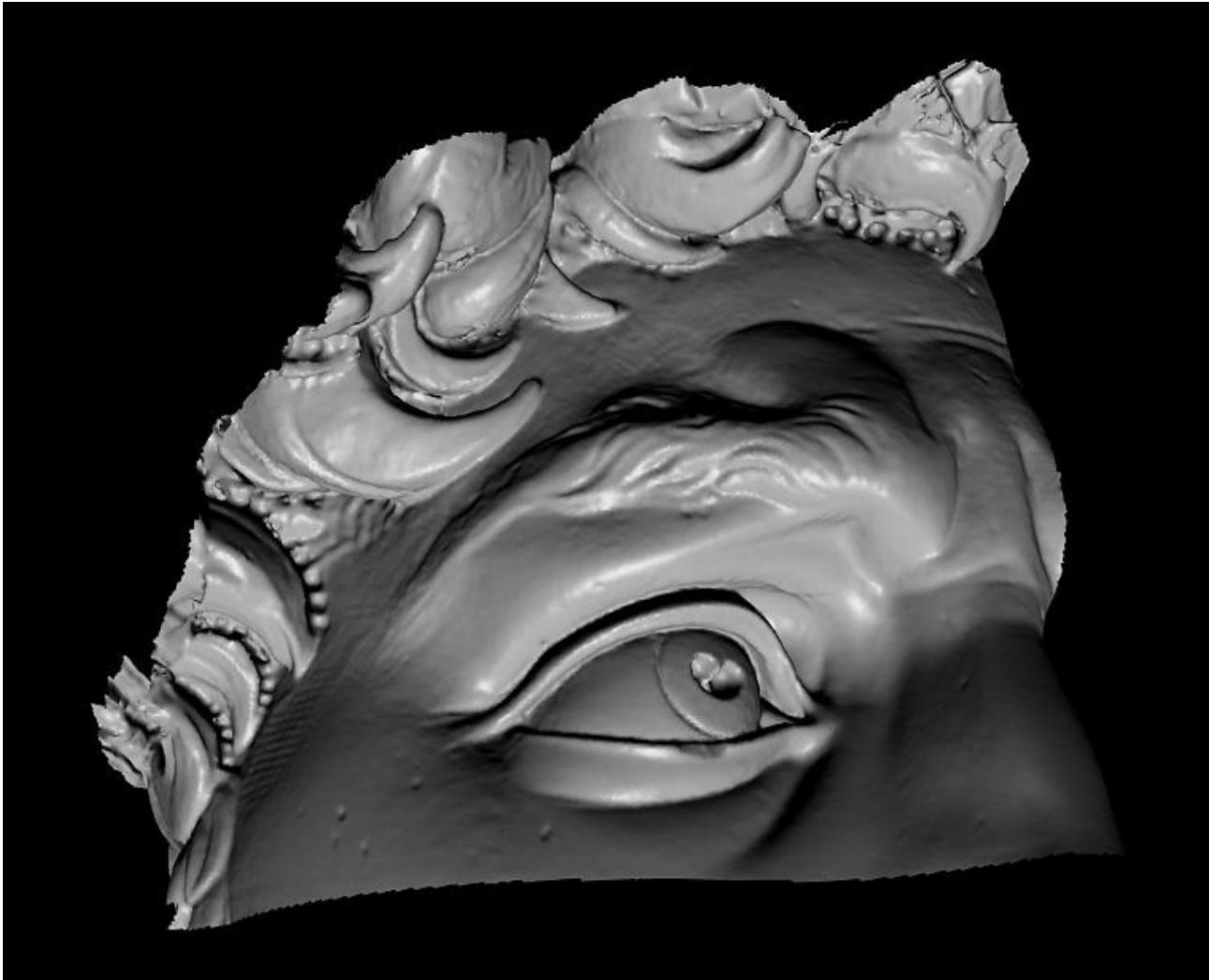
Laser scanned models



The Digital Michelangelo Project, Levoy et al.

Source: S. Seitz

Laser scanned models



The Digital Michelangelo Project, Levoy et al.

Source: S. Seitz