# Epipolar Geometry and Stereo Vision

Computer Vision

CS 543 / ECE 549

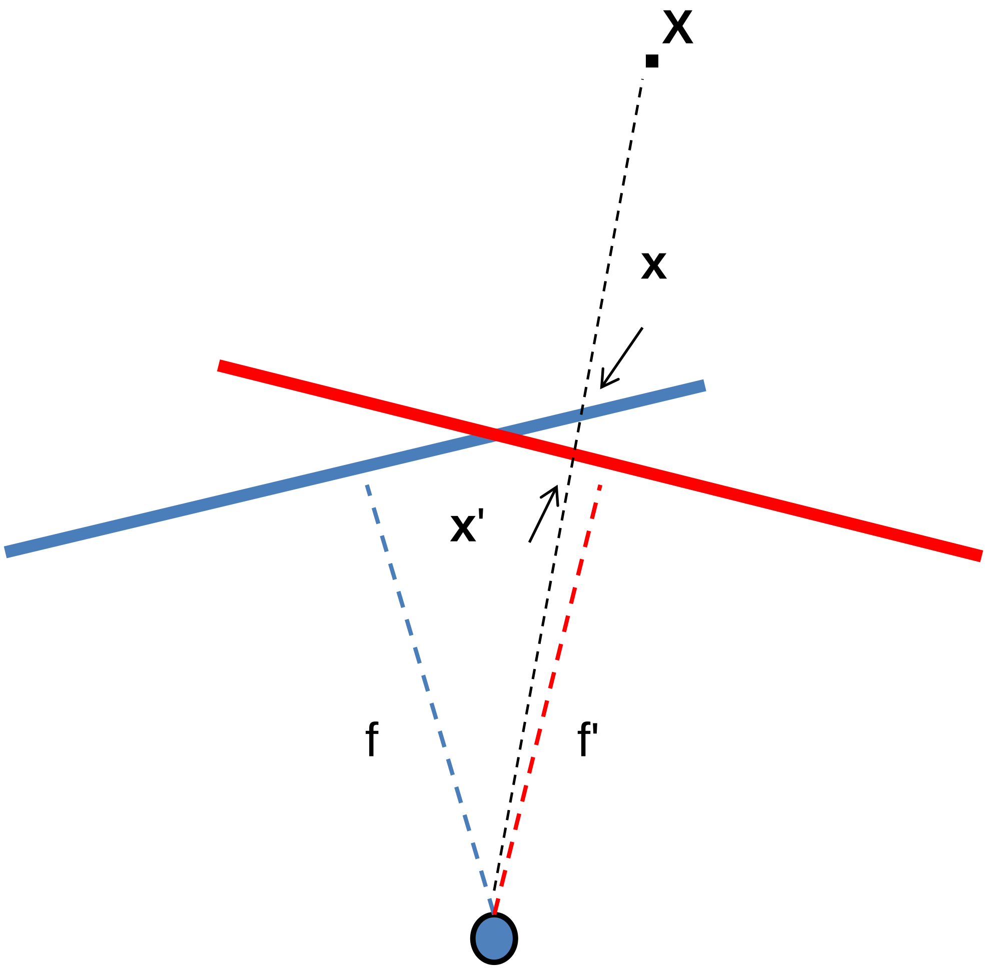
University of Illinois

Derek Hoiem

Many slides adapted from Lana Lazebnik, Silvio Saverese, Steve Seitz, many figures from Hartley & Zisserman

# Last class: Image Stitching

- Two images with rotation/zoom but no translation

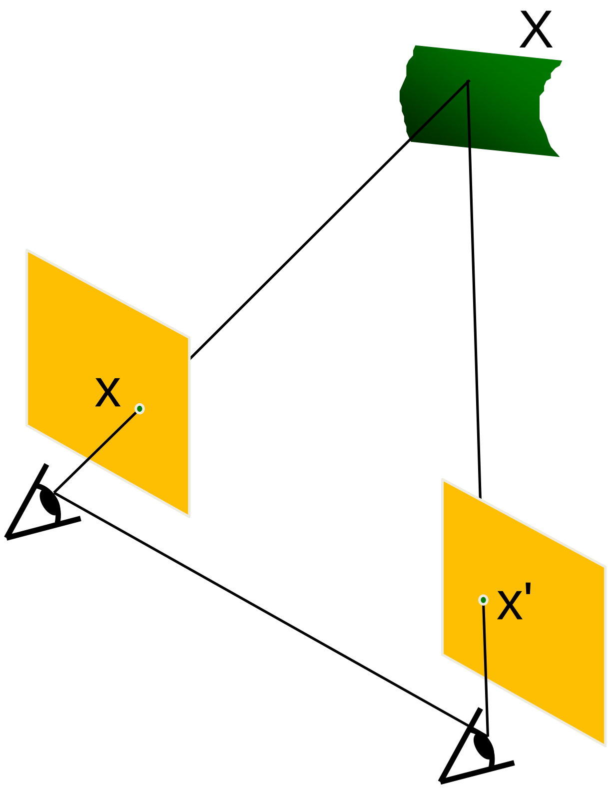


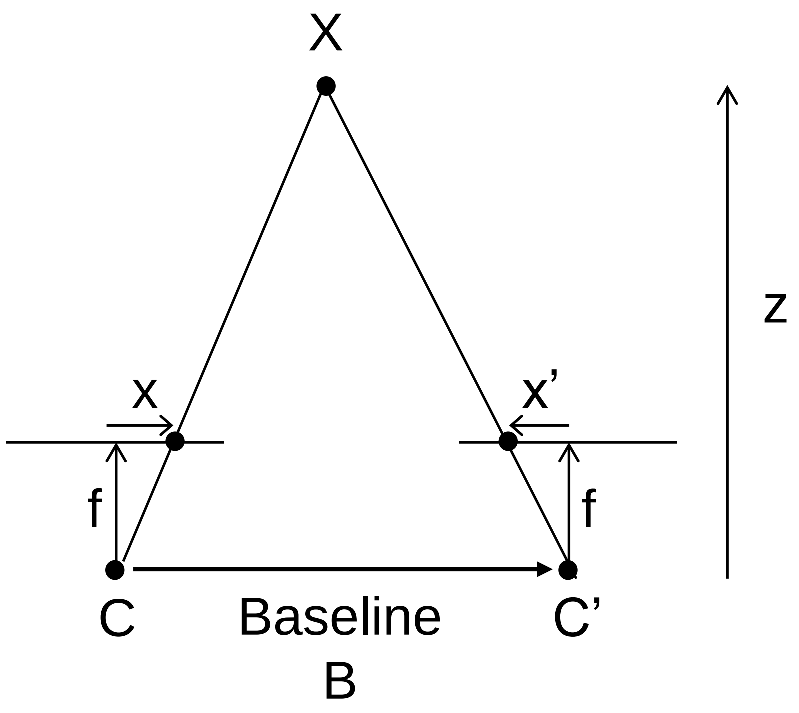
# This class: Two-View Geometry

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

# Depth from Stereo

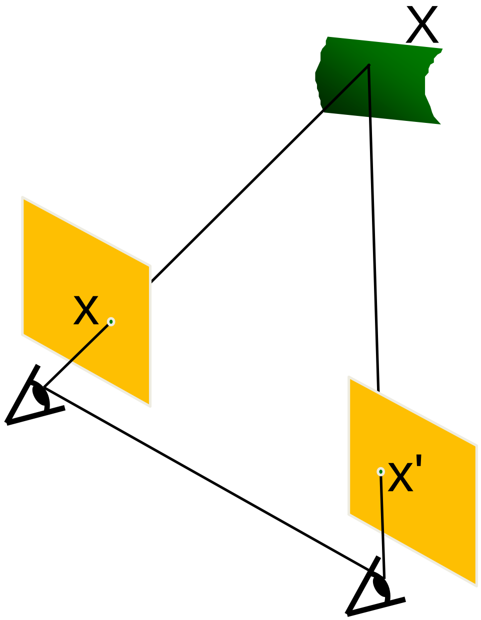
- Goal: recover depth by finding image coordinate x’ that corresponds to x



# Depth from Stereo

- Goal: recover depth by finding image coordinate x' that corresponds to x
- Sub-Problems
  1. Calibration: How do we recover the relation of the cameras (if not already known)?
  2. Correspondence: How do we search for the matching point x'?



# Correspondence Problem

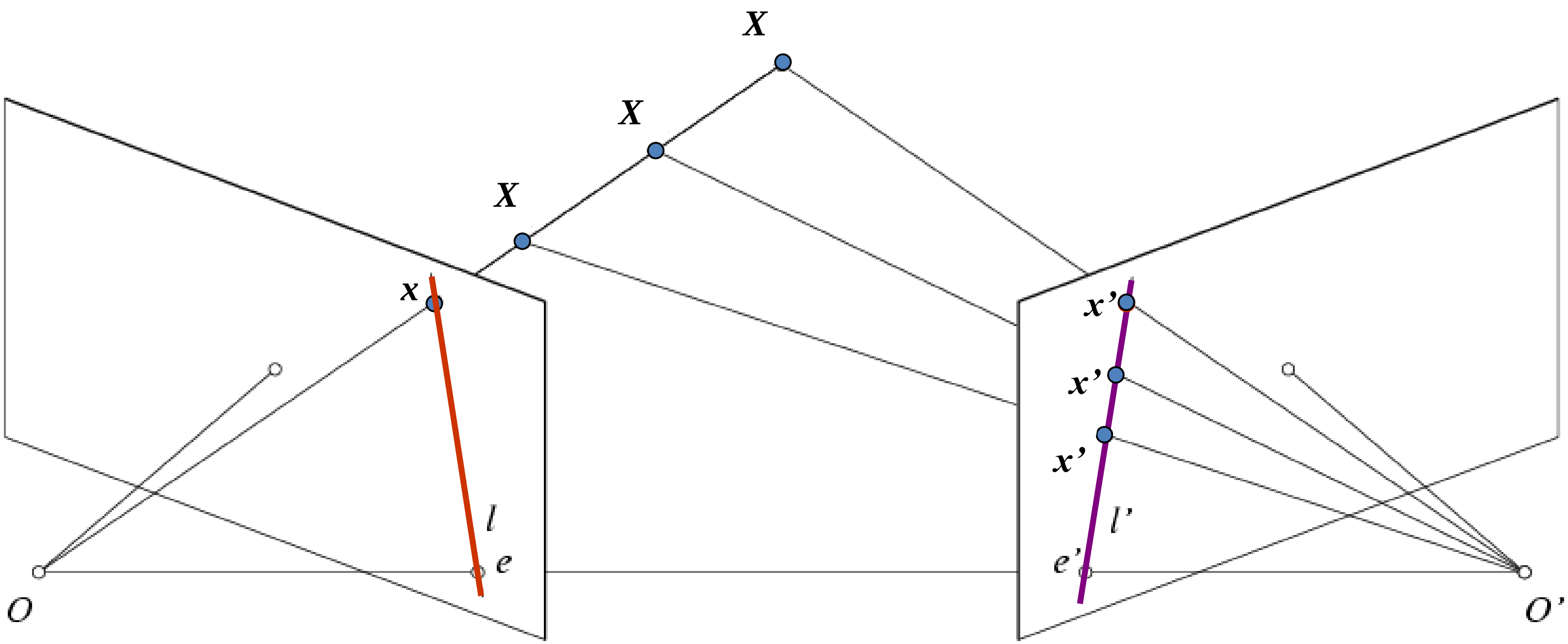


- We have two images taken from cameras with different intrinsic and extrinsic parameters

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

# Key idea: Epipolar constraint

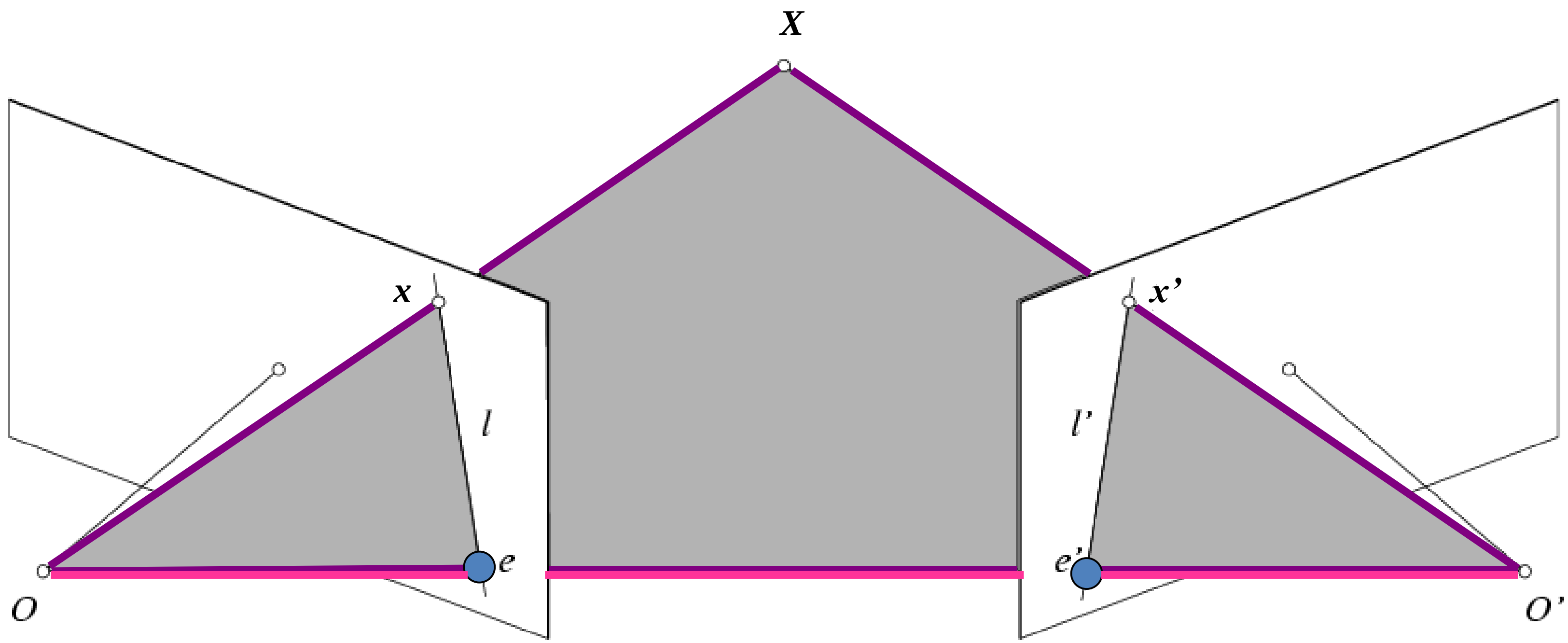
# Key idea: Epipolar constraint




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

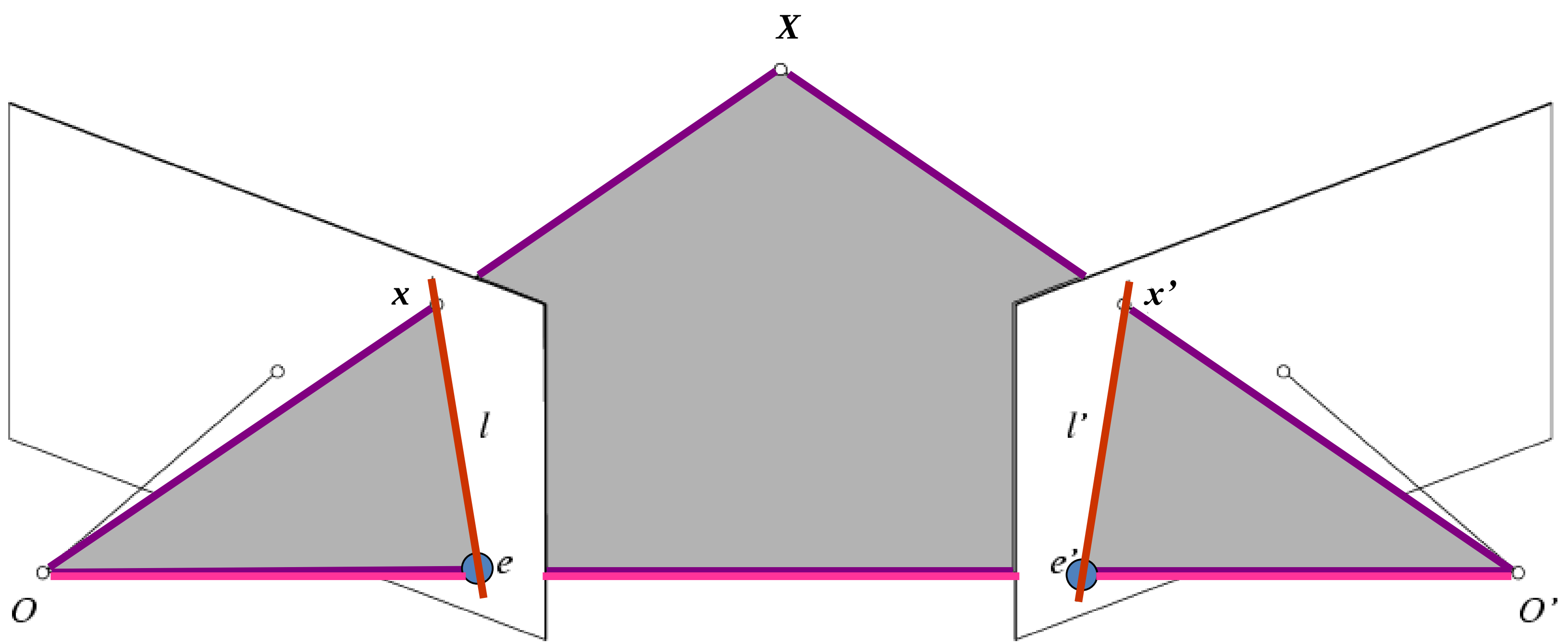
Potential matches for *x'* have to lie on the corresponding line *l*.

# Epipolar geometry: notation




- **Baseline** – line connecting the two camera centers
- **Epipoles**
= intersections of baseline with image planes
= projections of the other camera center
- **Epipolar Plane** – plane containing baseline (1D family)

# Epipolar geometry: notation



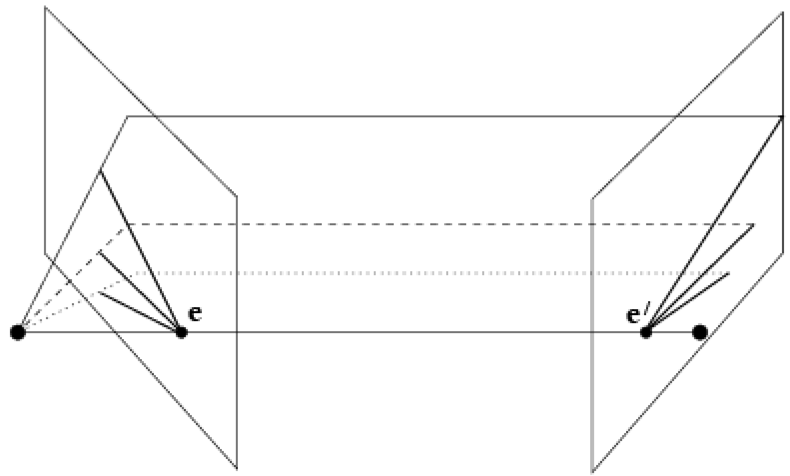

- **Baseline** – line connecting the two camera centers
- **Epipoles**

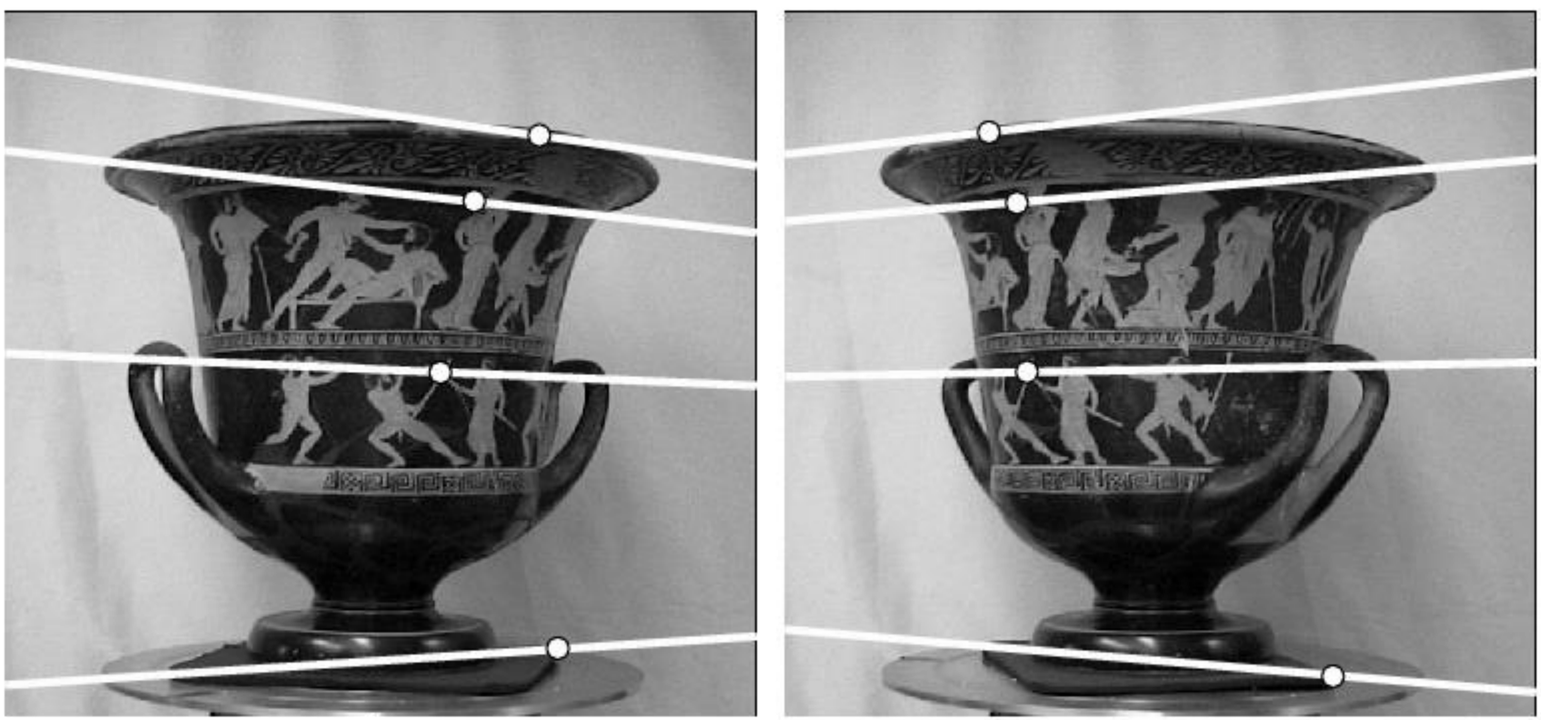
= intersections of baseline with image planes

= projections of the other camera center

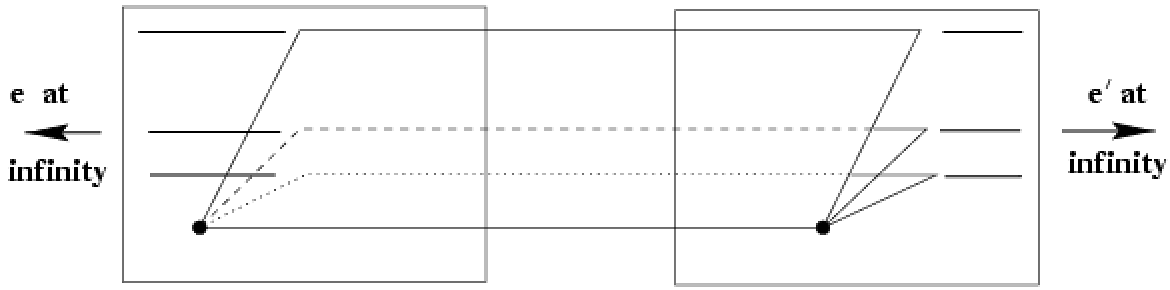
- **Epipolar Plane** – plane containing baseline (1D family)
- **Epipolar Lines** - intersections of epipolar plane with image planes (always come in corresponding pairs)

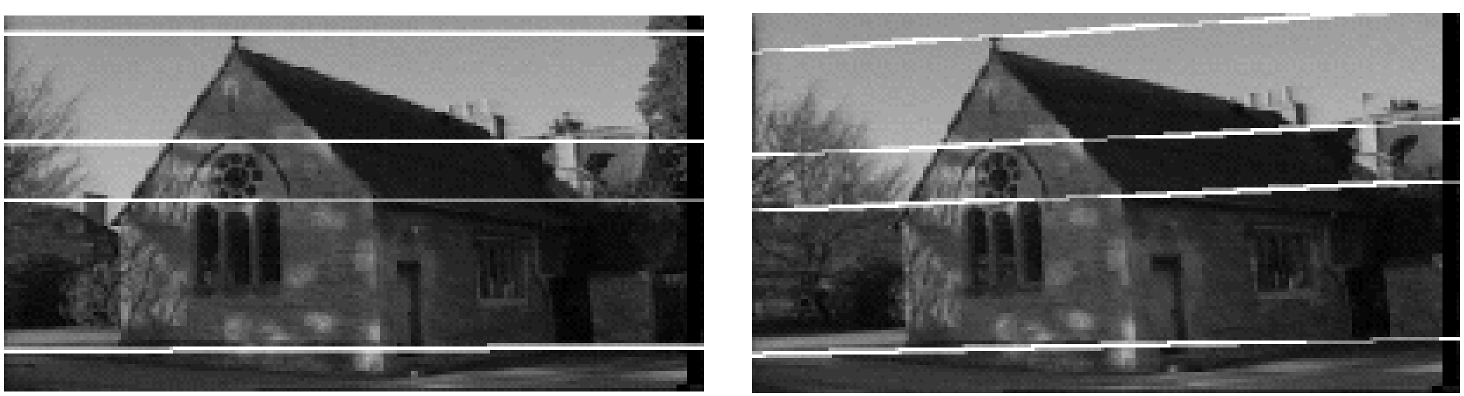
# Example: Converging cameras



# Example: Motion parallel to image plane

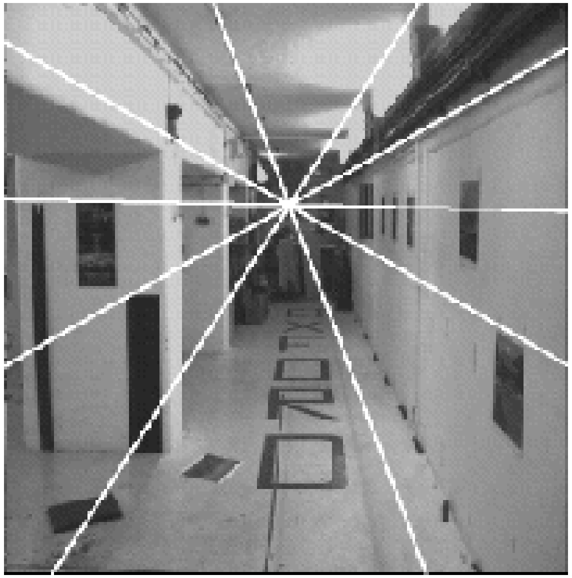


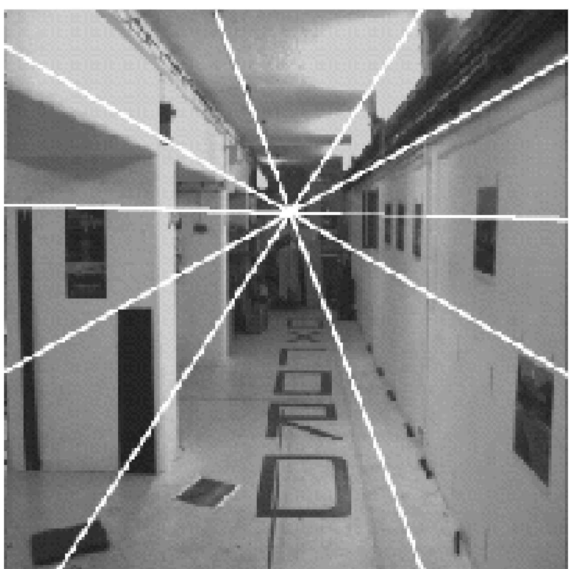



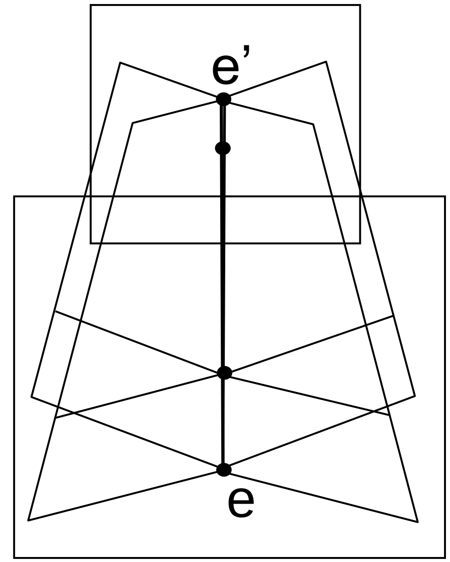
# Example: Forward motion

What would the epipolar lines look like if the camera moves directly forward?

# Example: Forward motion

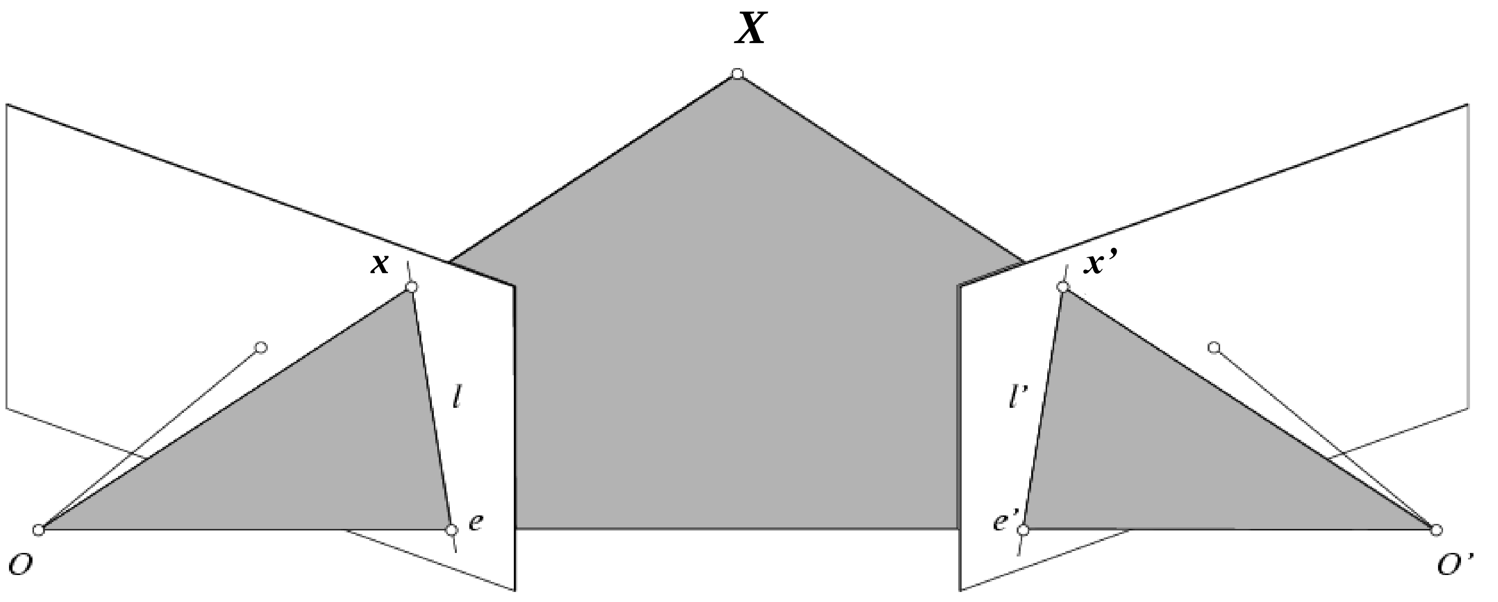







Epipole has same coordinates in both images.
Points move along lines radiating from e: "Focus of expansion"

# 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

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

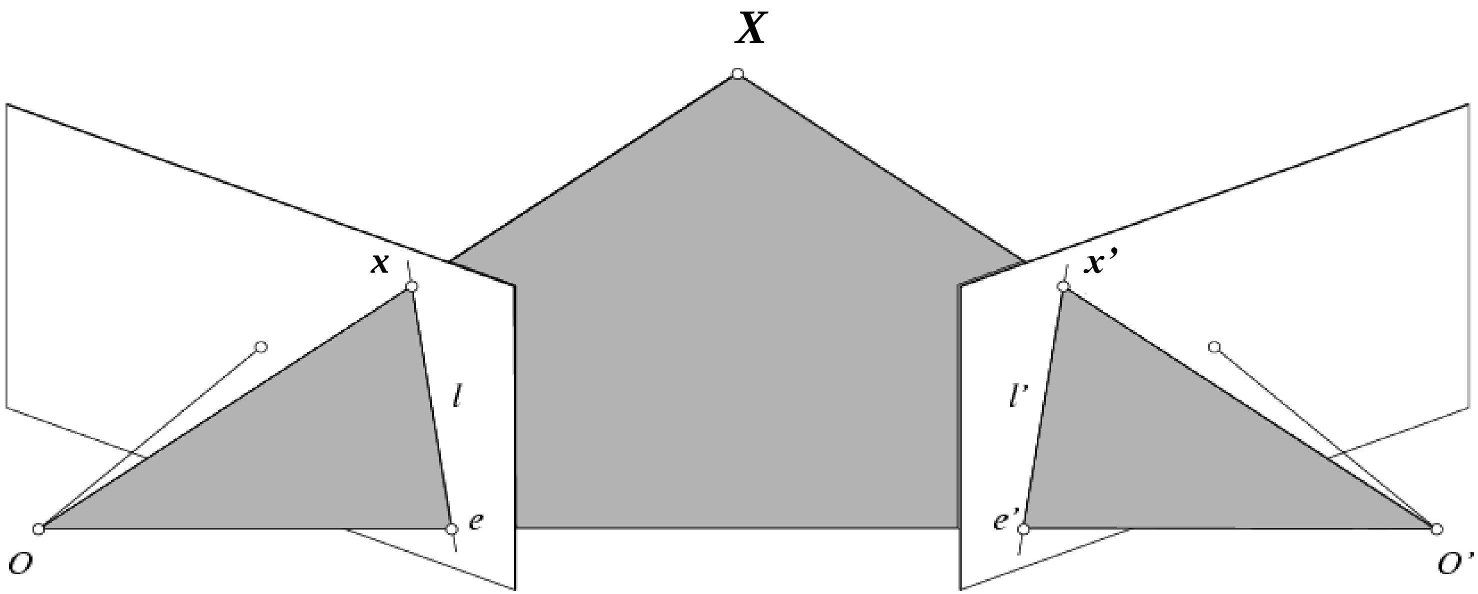
Homogeneous 2d point (3D ray towards X)

2D pixel coordinate (homogeneous)

3D scene point

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

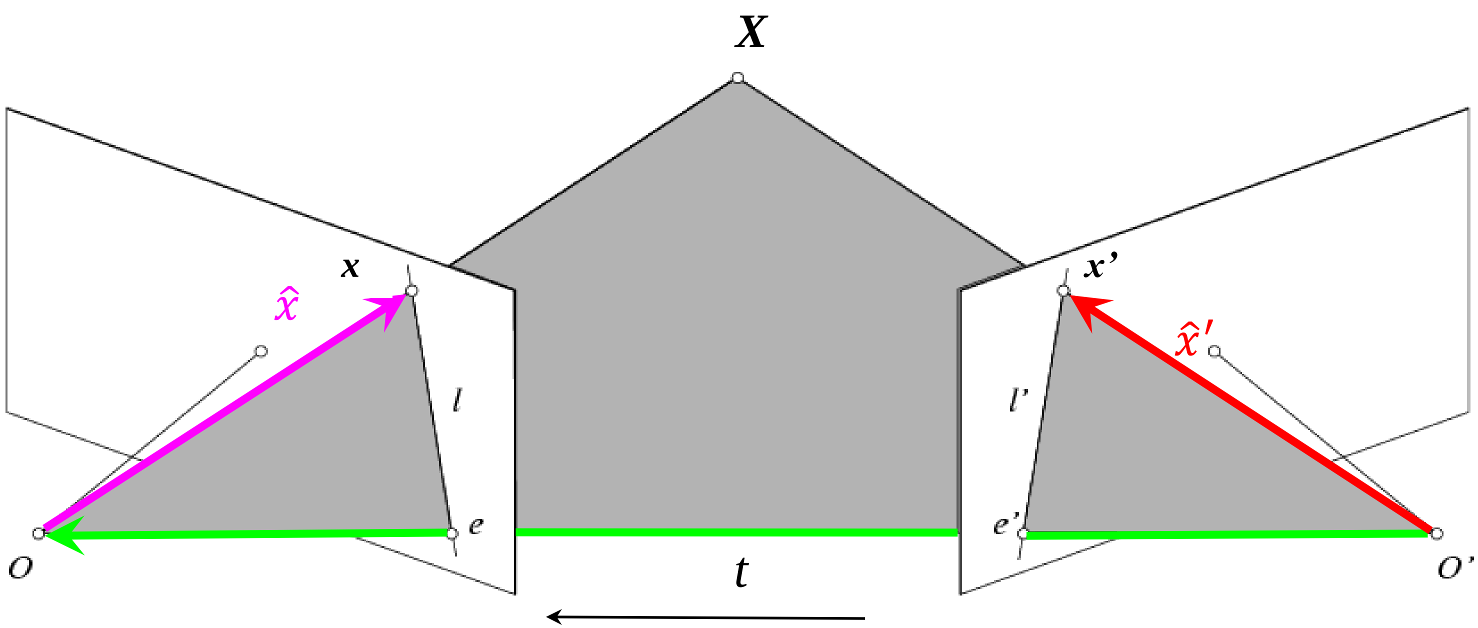
for some scale factor

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

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

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

# Epipolar constraint: Calibrated case

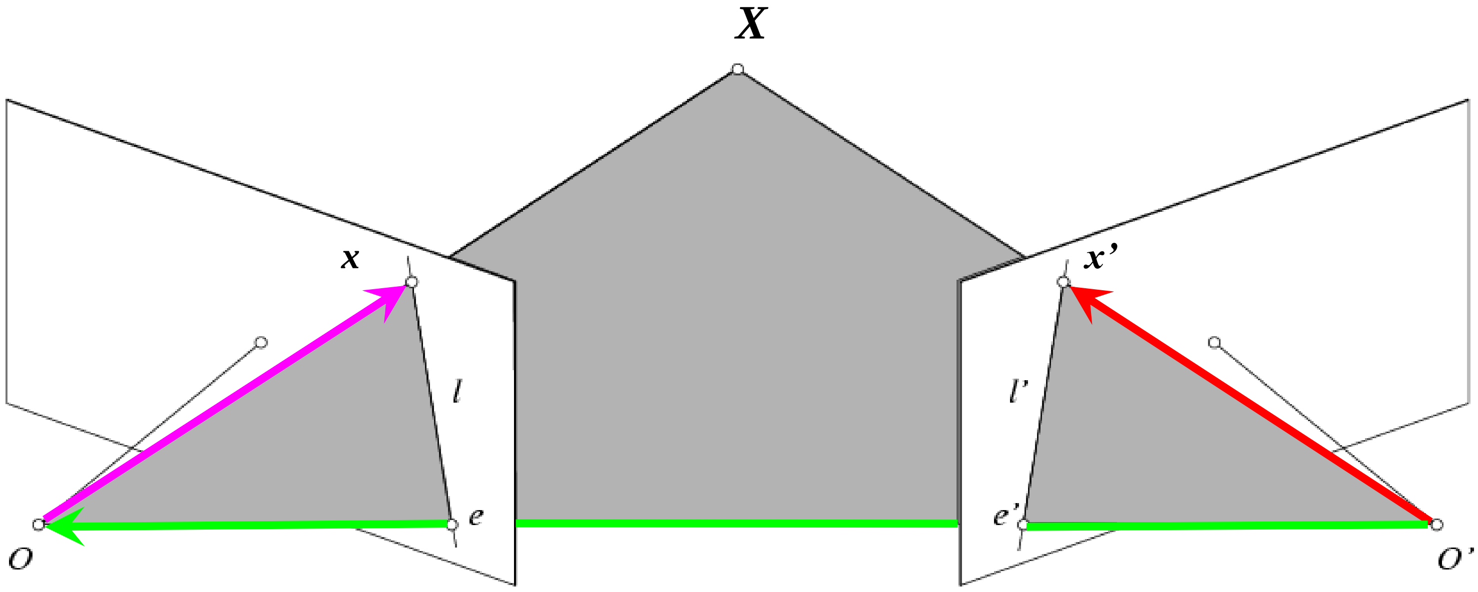



$$\hat{x} = K^{-1}x = X \qquad \hat{x}' = K'^{-1}x' = X'$$

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

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

# Essential matrix

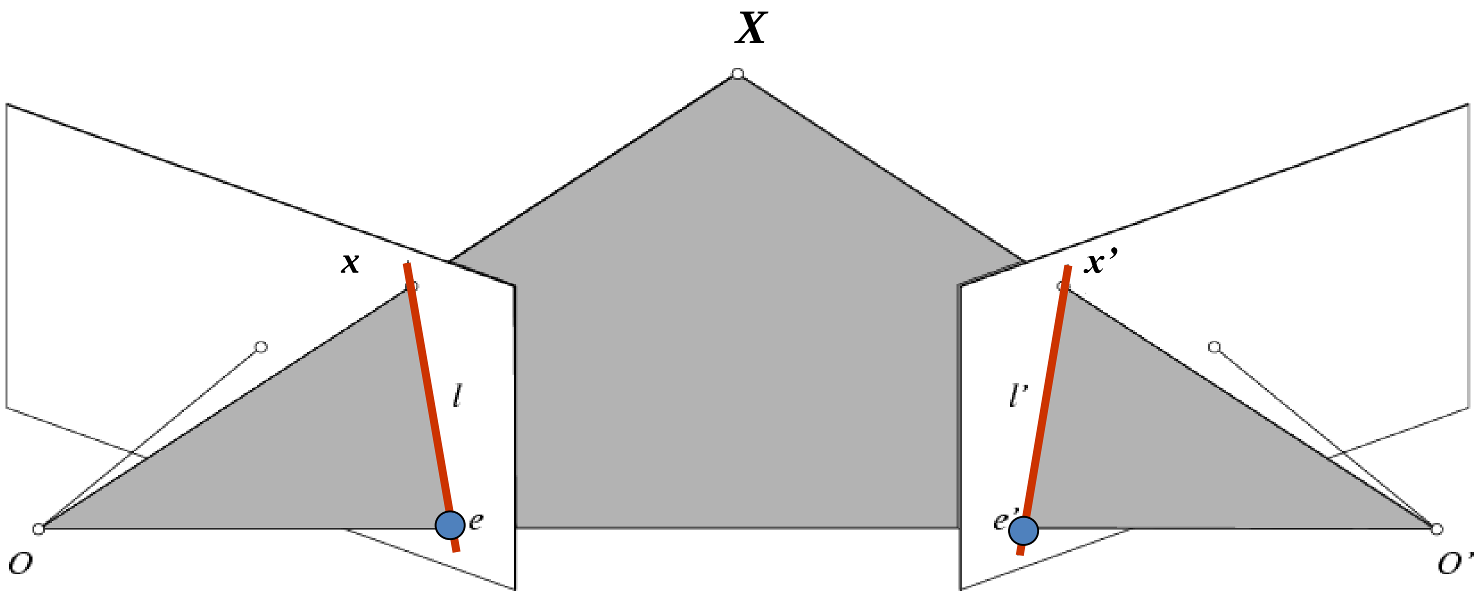



$$\hat{x} \cdot [t \times (R\hat{x}')] = 0$$

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

**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 degrees of freedom
  - (3 for R, 2 for t because it's up to a scale)

# The Fundamental Matrix

Without knowing K and K', we can define a similar relation using *unknown* normalized coordinates

$$\hat{x}^T E \hat{x}' = 0$$

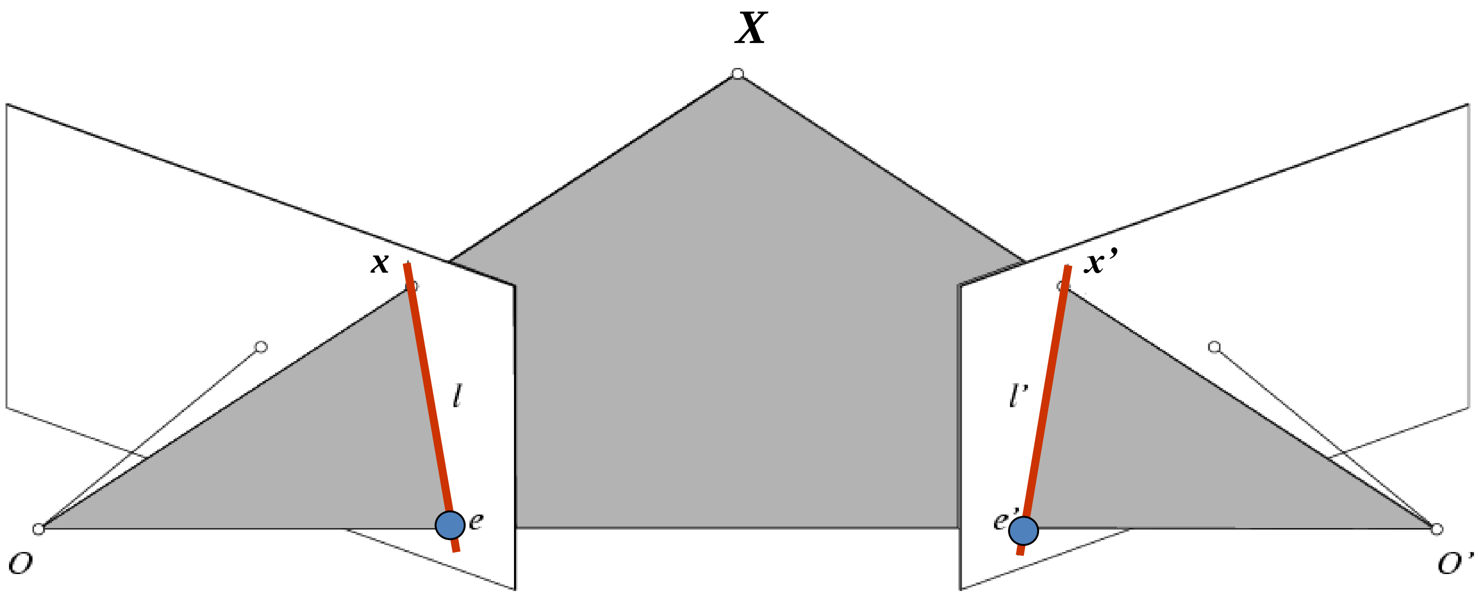
$$\hat{x} = K^{-1} x$$

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

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

**Fundamental Matrix**
(Faugeras and Luong, 1992)

# Properties of the Fundamental matrix




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

- *F x'* is the epipolar line associated with *x'* ($l = F x'$)
- $F^T x$ is the epipolar line associated with *x* ($l' = F^T x$)
- $F e' = 0$ and $F^T e = 0$
- *F* is singular (rank two): det(F)=0
- *F* has seven degrees of freedom: 9 entries but defined up to scale, det(F)=0

# Estimating the Fundamental Matrix

- 8-point algorithm
  - Least squares solution using SVD on equations from 8 pairs of correspondences
  - Enforce det(F)=0 constraint using SVD on F
- 7-point algorithm
  - Use least squares to solve for null space (two vectors) using SVD and 7 pairs of correspondences
  - Solve for linear combination of null space vectors that satisfies det(F)=0
- Minimize reprojection error
  - Non-linear least squares

Note: estimation of F (or E) is degenerate for a planar scene.

# 8-point algorithm

1. Solve a system of homogeneous linear equations
   a. Write down the system of equations

$$\mathbf{x}^T F \mathbf{x}' = 0$$

$$uu'f_{11} + uv'f_{12} + uf_{13} + vu'f_{21} + vv'f_{22} + vf_{23} + u'f_{31} + v'f_{32} + f_{33} = 0$$

$$\mathrm{A}\boldsymbol{f} = \begin{bmatrix} u_1u_1' & u_1v_1' & u_1 & v_1u_1' & v_1v_1' & v_1 & u_1' & v_1' & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu_v' & u_nv_n' & u_n & v_nu_n' & v_nv_n' & v_n & u_n' & v_n' & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ \vdots \\ f_{33} \end{bmatrix} = \mathbf{0}$$

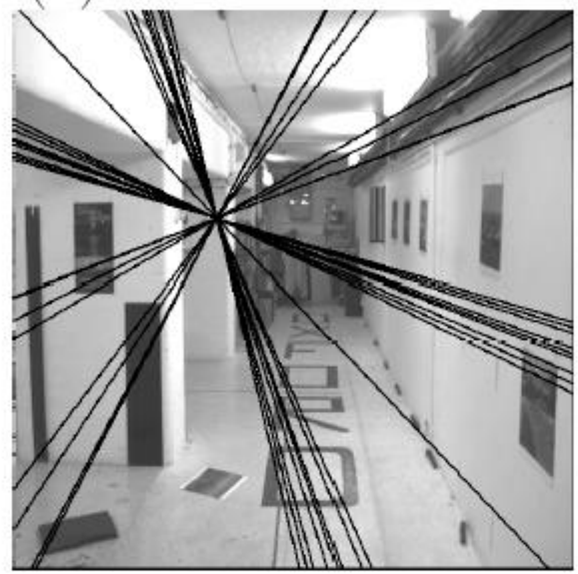
# 8-point algorithm

1. Solve a system of homogeneous linear equations
   a. Write down the system of equations
   b. Solve **f** from A**f**=**0** using SVD

      Matlab:
      ```
      [U, S, V] = svd(A);
      f = V(:, end);
      F = reshape(f, [3 3])';
      ```

# Need to enforce singularity constraint

Fundamental matrix has rank 2 : $\det(\mathtt{F}) = 0$.



**Left :** Uncorrected $\mathtt{F}$ – epipolar lines are not coincident.

**Right :** Epipolar lines from corrected $\mathtt{F}$.

# 8-point algorithm

1. Solve a system of homogeneous linear equations
   a. Write down the system of equations
   b. Solve **f** from A**f**=**0** using SVD

      Matlab:
      ```
      [U, S, V] = svd(A);
      f = V(:, end);
      F = reshape(f, [3 3])';
      ```

2. Resolve det(F) = 0 constraint using SVD

   Matlab:
   ```
   [U, S, V] = svd(F);
   S(3,3) = 0;
   F = U*S*V';
   ```

# 8-point algorithm

1. Solve a system of homogeneous linear equations
   a. Write down the system of equations
   b. Solve **f** from A**f**=**0** using SVD
2. Resolve det(F) = 0 constraint by SVD

Notes:

- Use RANSAC to deal with outliers (sample 8 points)
  – How to test for outliers?
- Solve in normalized coordinates
  – mean=0
  – standard deviation ~= (1,1,1)
  – just like with estimating the homography for stitching

# Comparison of homography estimation and the 8-point algorithm

Assume we have matched points $x \Leftrightarrow x'$ with outliers

**Homography (No Translation)**

**Fundamental Matrix (Translation)**

# Comparison of homography estimation and the 8-point algorithm

Assume we have matched points x ⇔ x' with outliers

**Homography (No Translation)**

**Fundamental Matrix (Translation)**

- Correspondence Relation

$$\mathbf{x}' = \mathbf{H}\mathbf{x} \Rightarrow \mathbf{x}' \times \mathbf{H}\mathbf{x} = \mathbf{0}$$

1. Normalize image coordinates

$$\tilde{\mathbf{x}} = \mathbf{T}\mathbf{x} \quad \tilde{\mathbf{x}}' = \mathbf{T}'\mathbf{x}'$$

2. RANSAC with 4 points
   - Solution via SVD
3. De-normalize: $\mathbf{H} = \mathbf{T}'^{-1}\tilde{\mathbf{H}}\mathbf{T}$

# Comparison of homography estimation and the 8-point algorithm

Assume we have matched points x ⇔ x' with outliers

**Homography (No Translation)**

- Correspondence Relation
  $$\mathbf{x}' = \mathbf{H}\mathbf{x} \Rightarrow \mathbf{x}' \times \mathbf{H}\mathbf{x} = \mathbf{0}$$

1. Normalize image coordinates
   $$\tilde{\mathbf{x}} = \mathbf{T}\mathbf{x} \quad \tilde{\mathbf{x}}' = \mathbf{T}'\mathbf{x}'$$
2. RANSAC with 4 points
   - Solution via SVD
3. De-normalize: $\mathbf{H} = \mathbf{T}'^{-1}\tilde{\mathbf{H}}\mathbf{T}$

**Fundamental Matrix (Translation)**

- Correspondence Relation
  $$\mathbf{x}'^{T}\mathbf{F}\mathbf{x} = 0$$

1. Normalize image coordinates
   $$\tilde{\mathbf{x}} = \mathbf{T}\mathbf{x} \quad \tilde{\mathbf{x}}' = \mathbf{T}'\mathbf{x}'$$
2. RANSAC with 8 points
   - Initial solution via SVD
   - Enforce $\det(\tilde{\mathbf{F}}) = 0$ by SVD
3. De-normalize: $\mathbf{F} = \mathbf{T}'^{T}\tilde{\mathbf{F}}\mathbf{T}$

# 7-point algorithm

## Computation of $\mathtt{F}$ from 7 point correspondences

(i) Form the $7 \times 9$ set of equations $\mathtt{A}\mathbf{f} = 0$.

(ii) System has a 2-dimensional solution set.

(iii) General solution (use SVD) has form

$$\mathbf{f} = \lambda \mathbf{f}_0 + \mu \mathbf{f}_1$$

(iv) In matrix terms

$$\mathtt{F} = \lambda \mathtt{F}_0 + \mu \mathtt{F}_1$$

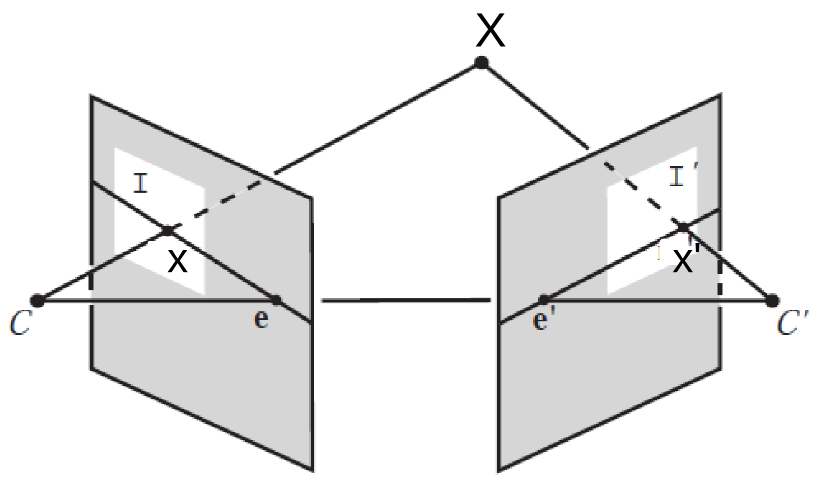
(v) Condition $\det \mathtt{F} = 0$ gives cubic equation in $\lambda$ and $\mu$.

(vi) Either one or three real solutions for ratio $\lambda : \mu$.

Faster (need fewer points) and could be more robust (fewer points), but also need to check for degenerate cases

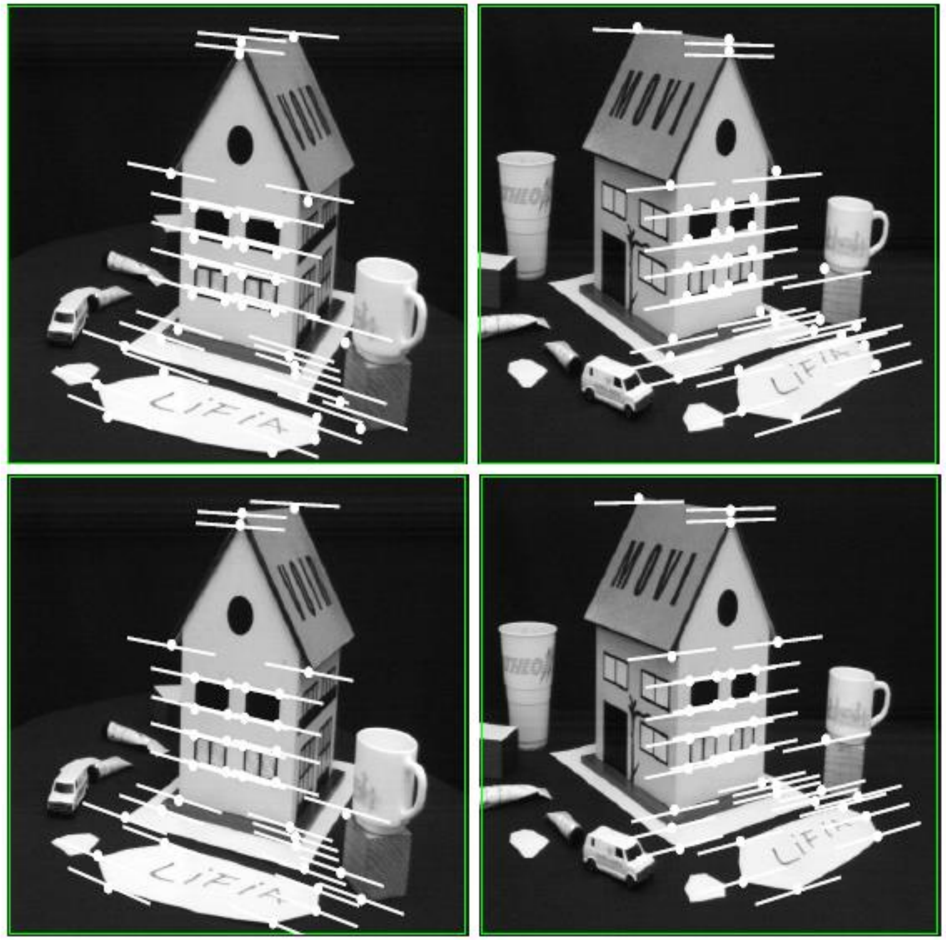
# “Gold standard” algorithm

- Use 8-point algorithm to get initial value of F
- Use F to solve for P and P’ (discussed later)
- Jointly solve for 3d points **X** and **F** that minimize the squared re-projection error




See Algorithm 11.2 and Algorithm 11.3 in HZ (pages 284-285) for details

# 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 |

# We can get projection matrices P and P' up to a projective ambiguity

K'*rotation K'*translation

$$\mathbf{P} = [\mathbf{I} \mid \mathbf{0}] \quad \mathbf{P}' = [[\mathbf{e}']_{\times}\mathbf{F} \mid \mathbf{e}'] \quad \mathbf{e}'^{T}\mathbf{F} = 0$$

See HZ p. 255-256

Code:

```
function P = vgg_P_from_F(F)
[U,S,V] = svd(F);
e = U(:,3);
P = [-vgg_contreps(e)*F e];
```

If we know the intrinsic matrices (K and K'), we can resolve the ambiguity

# Let’s recap…

- Fundamental matrix song

# Moving on to stereo…

Fuse a calibrated binocular stereo pair to produce a depth image

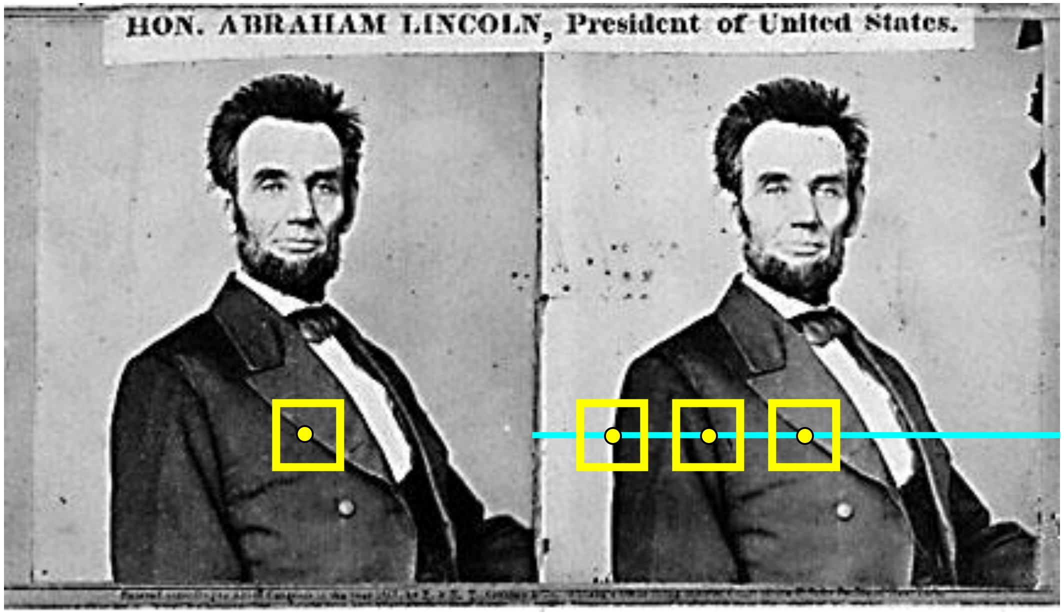
image 1

image 2

Dense depth map

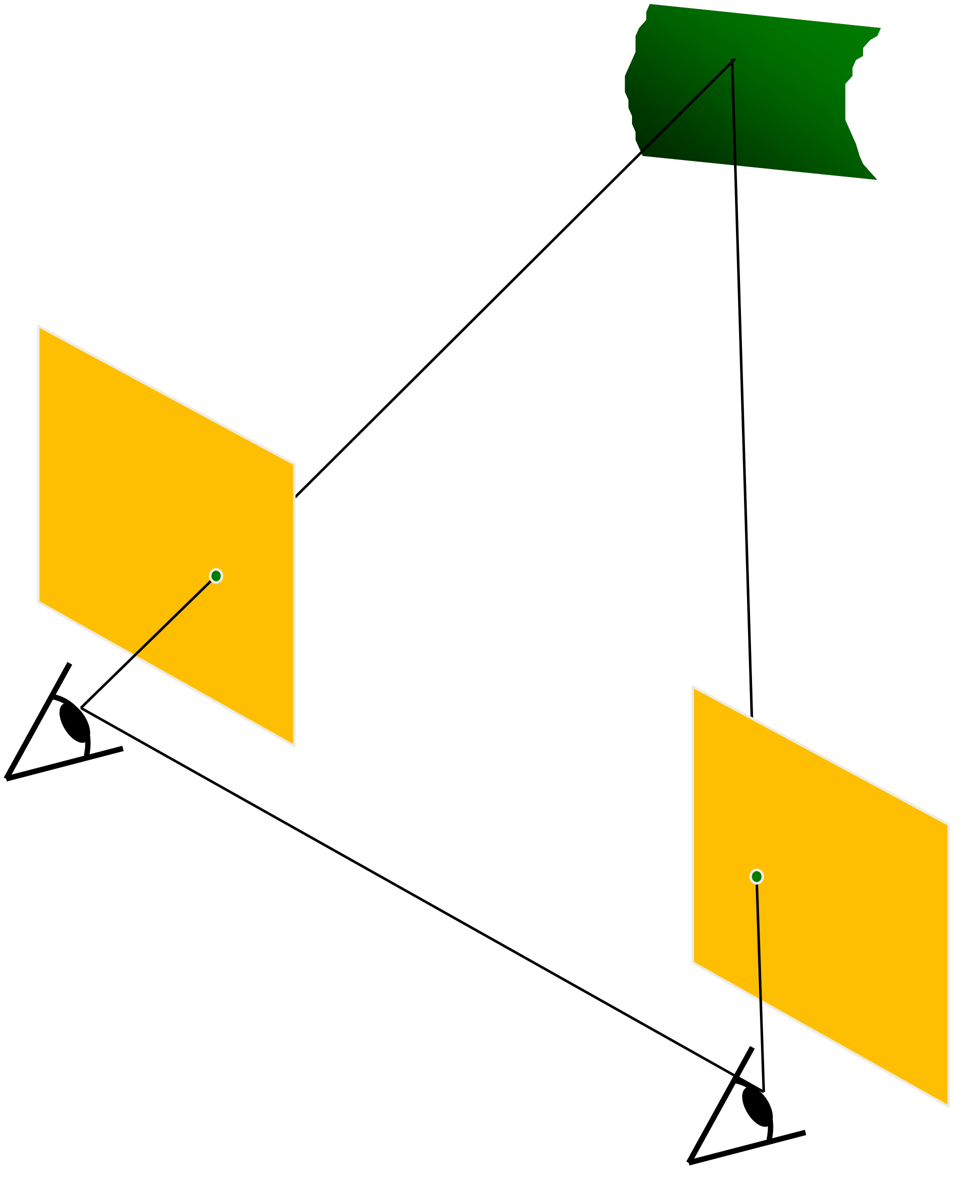
Many of these slides adapted from Steve Seitz and Lana Lazebnik

# Basic stereo matching algorithm



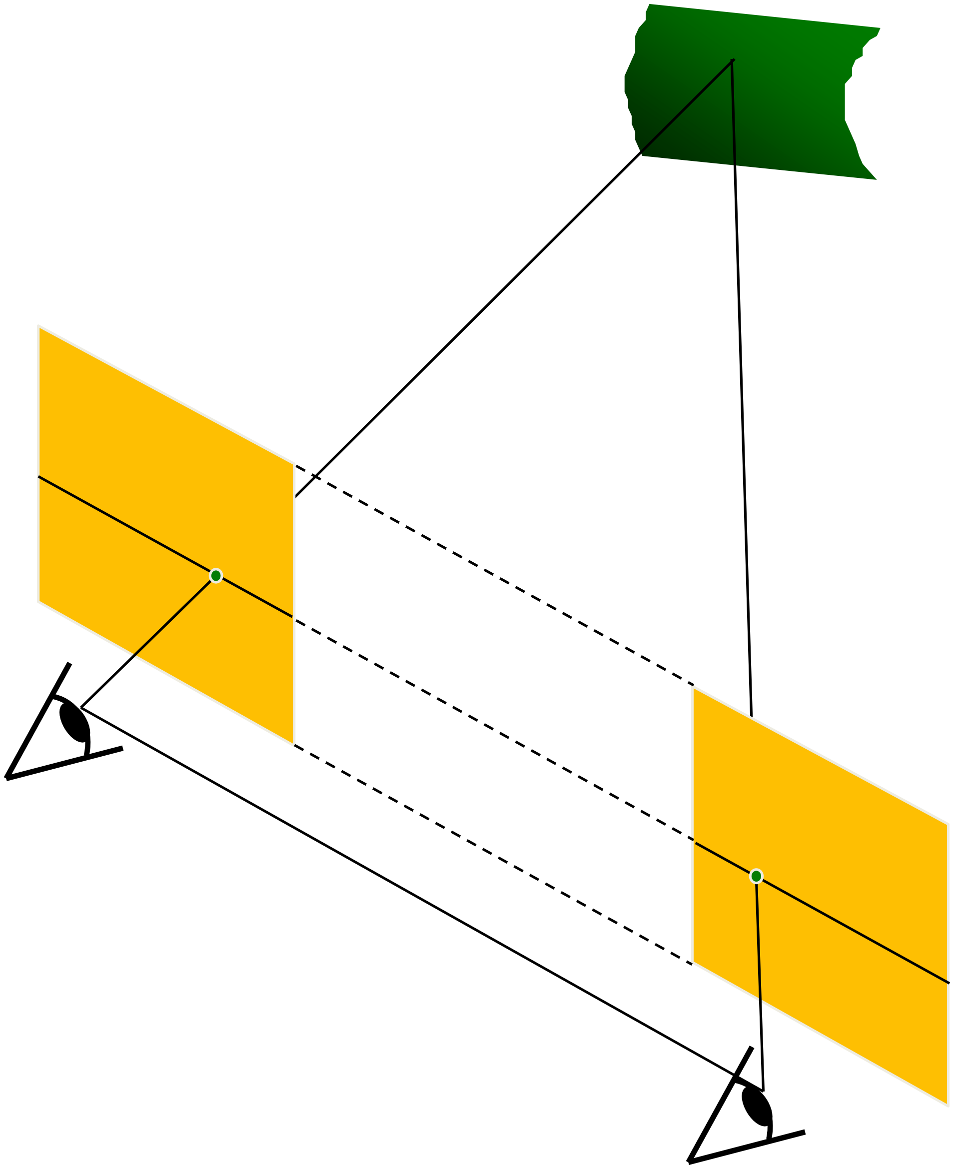

- For each pixel in the first image
  - Find corresponding epipolar line in the right image
  - Search along epipolar line and pick the best match
  - Triangulate the matches to get depth information
- Simplest case: epipolar lines are scanlines
  - When does this happen?

# Simplest Case: Parallel images



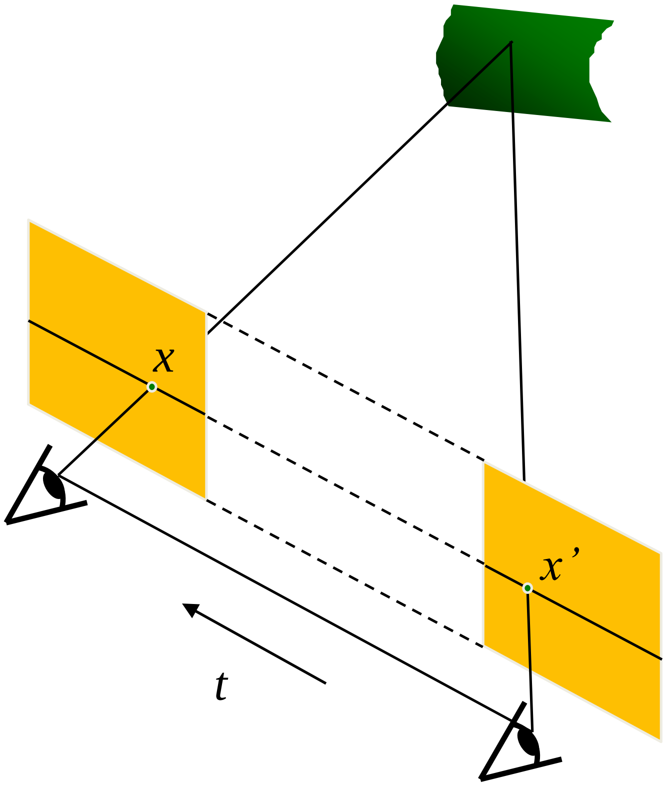
- Image planes of cameras are parallel to each other and to the baseline
- Camera centers are at same height
- Focal lengths are the same

# 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

# Simplest Case: Parallel images




Epipolar constraint:

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

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

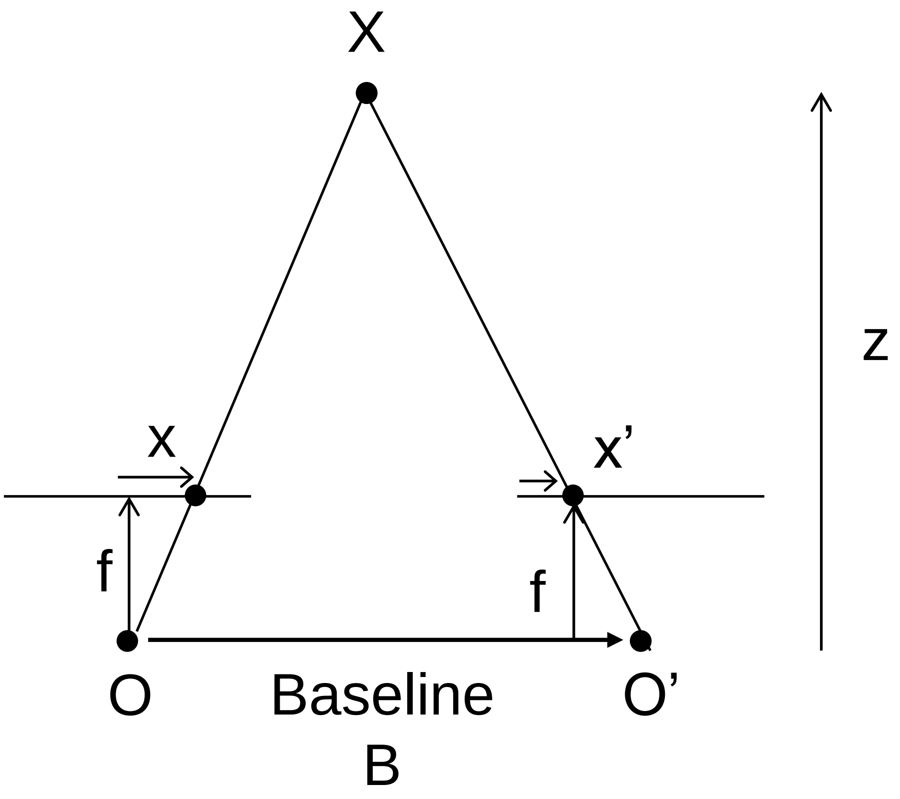
$$E = t \times R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix}$$

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

The y-coordinates of corresponding points are the same

# Depth from disparity

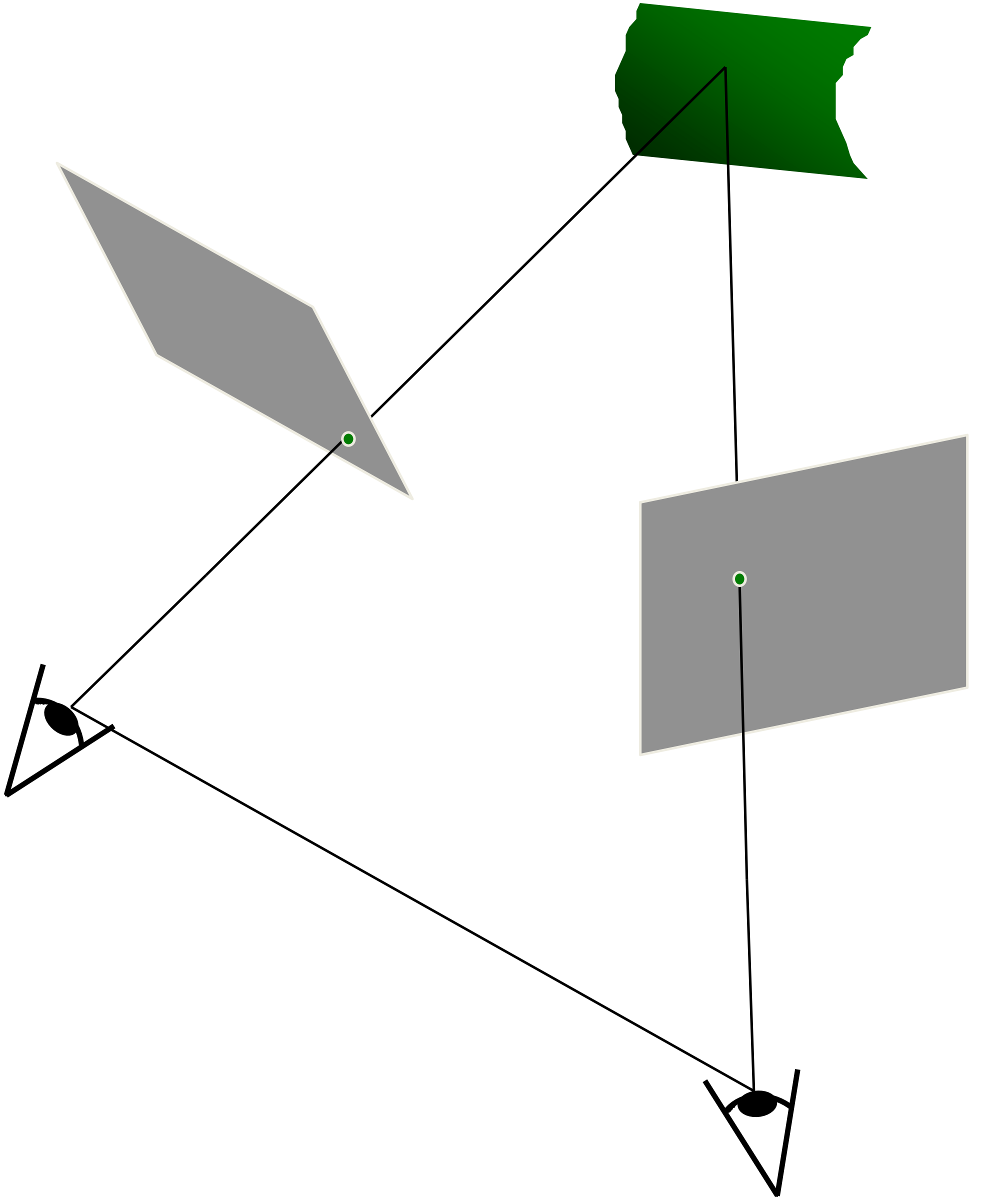
$$\frac{x - x'}{O - O'} = \frac{f}{z}$$




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

Disparity is inversely proportional to depth.

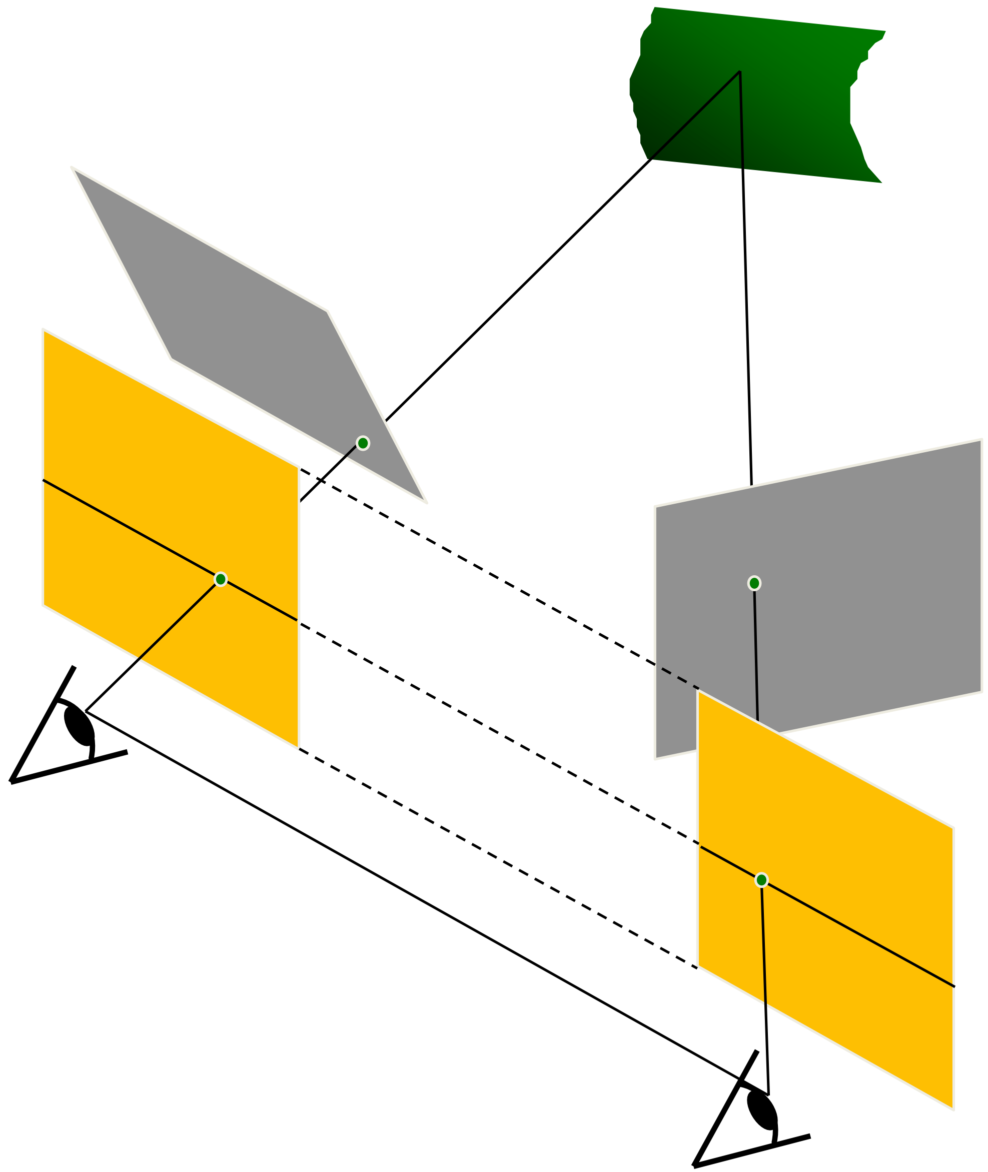
# Stereo image rectification



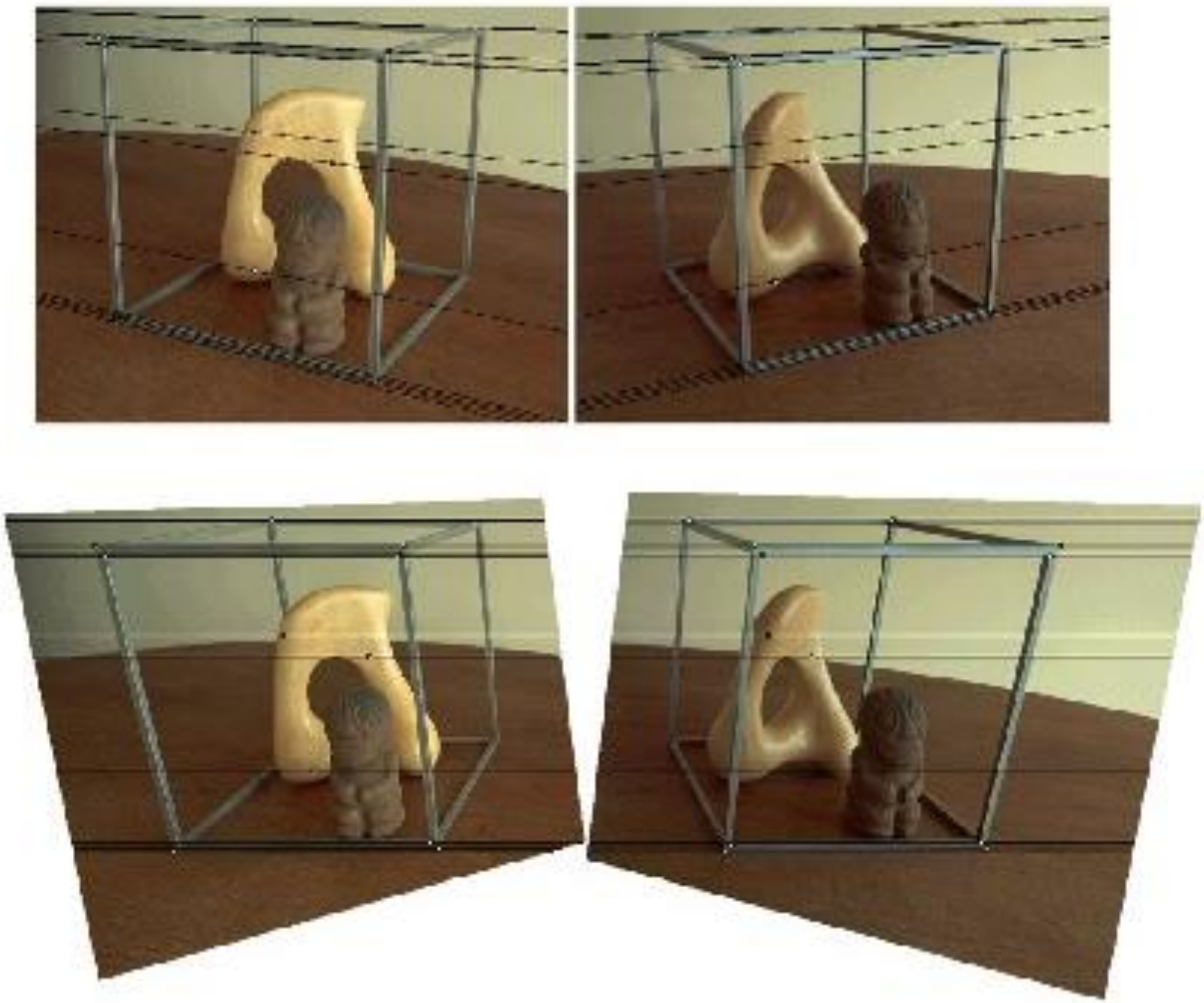
# Stereo image rectification

- 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

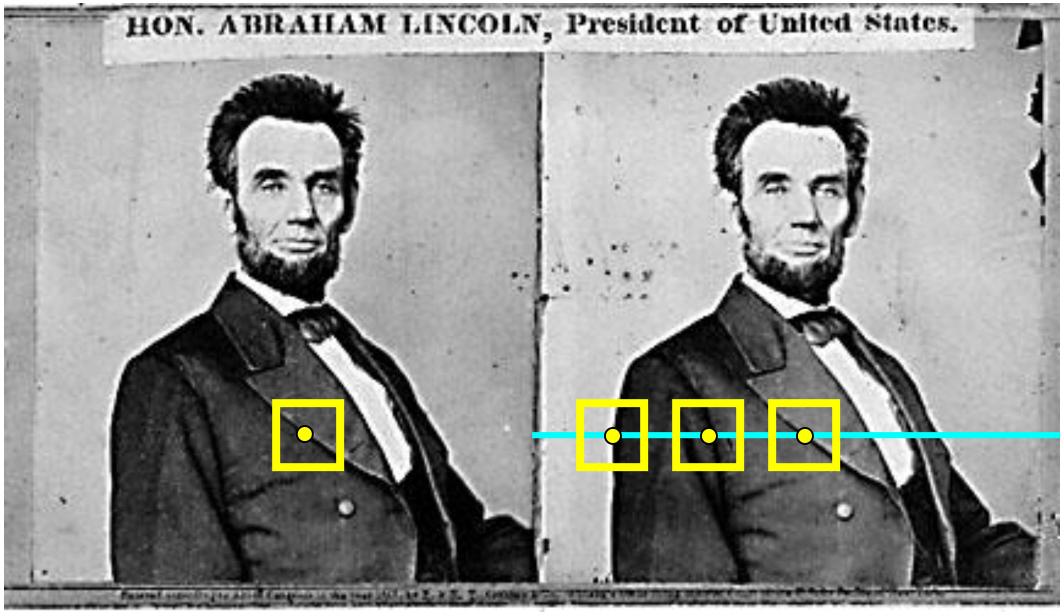
➢ C. Loop and Z. Zhang. Computing Rectifying Homographies for Stereo Vision. IEEE Conf. Computer Vision and Pattern Recognition, 1999.



# Rectification example

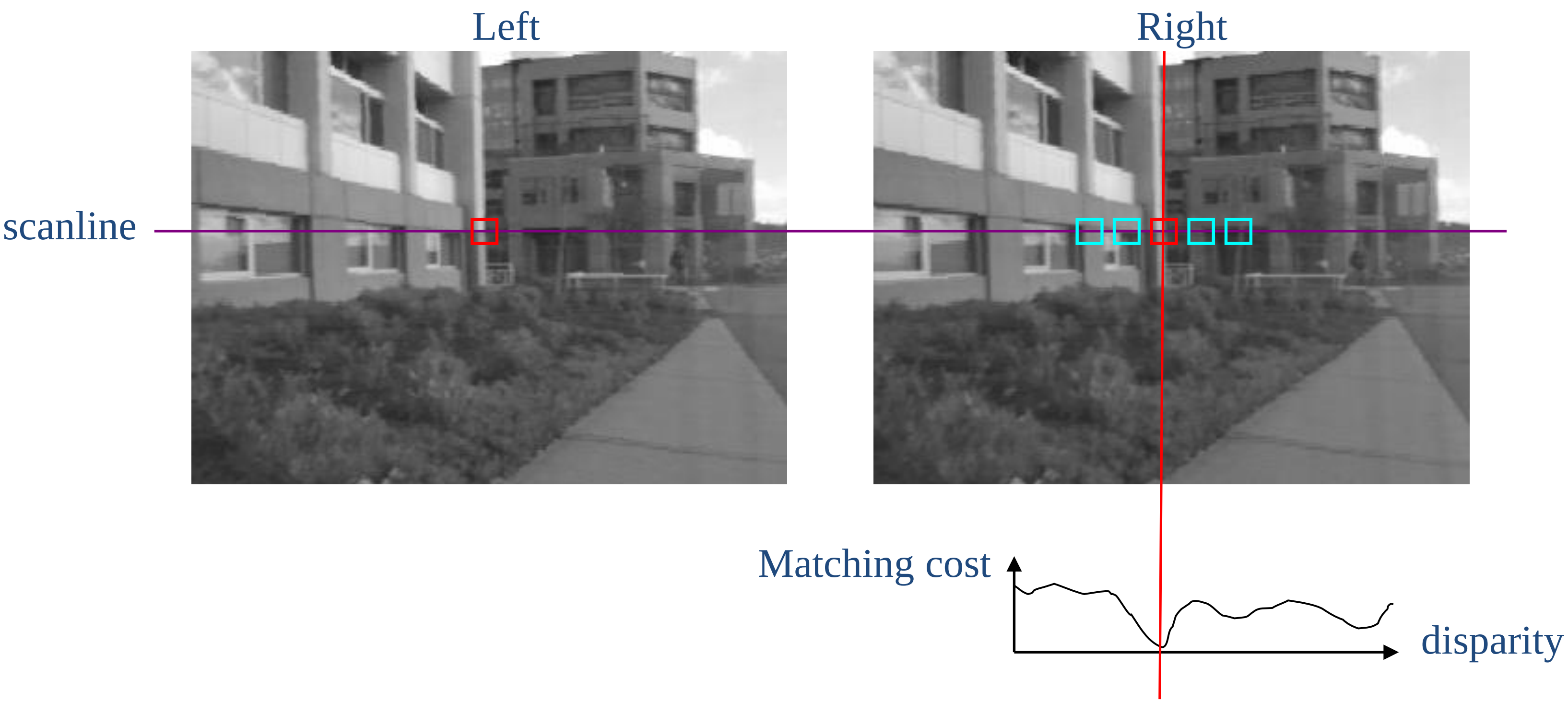


# Basic stereo matching algorithm



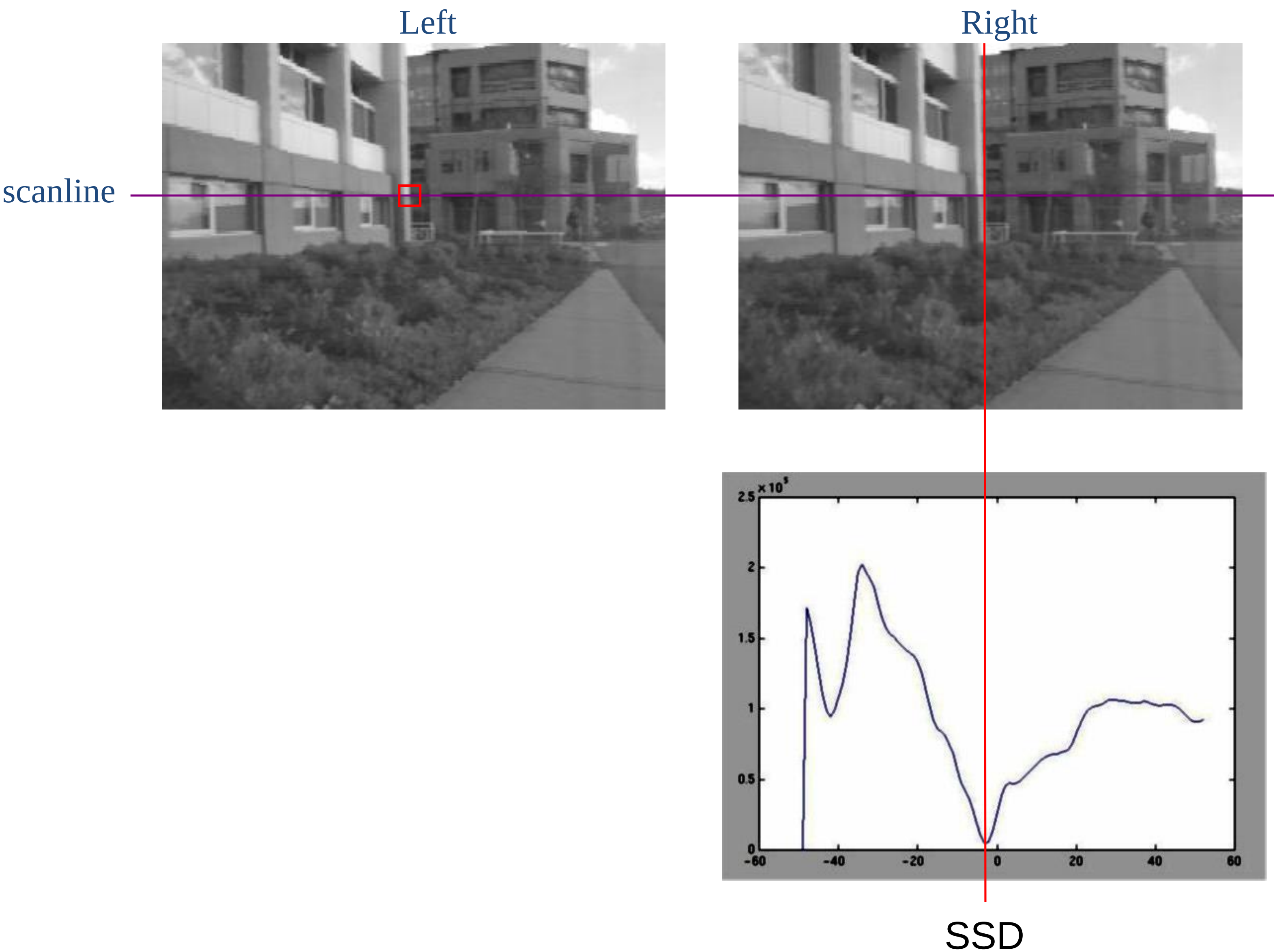

- If necessary, rectify the two stereo images to transform epipolar lines into scanlines
- For each pixel x in the first image
  - Find corresponding epipolar scanline in the right image
  - Search the scanline and pick the best match x’
  - Compute disparity x-x’ and set depth(x) = fB/(x-x’)

# Correspondence search

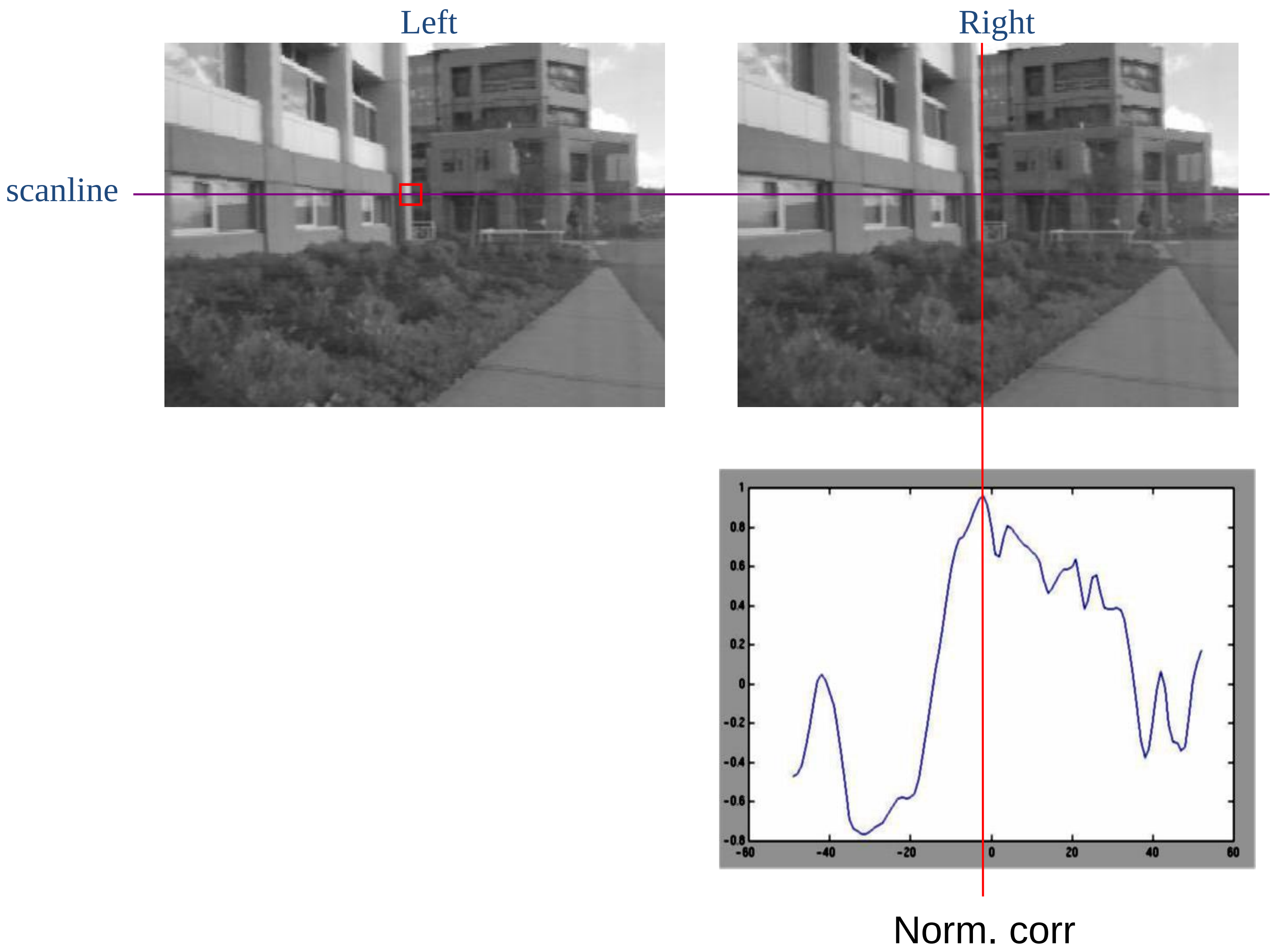



- Slide a window along the right scanline and compare contents of that window with the reference window in the left image
- Matching cost: SSD or normalized correlation

# Correspondence search

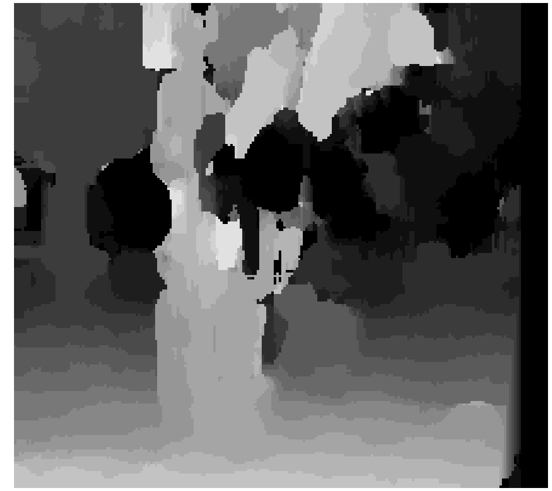


# Correspondence search



# Effect of window size

W = 3

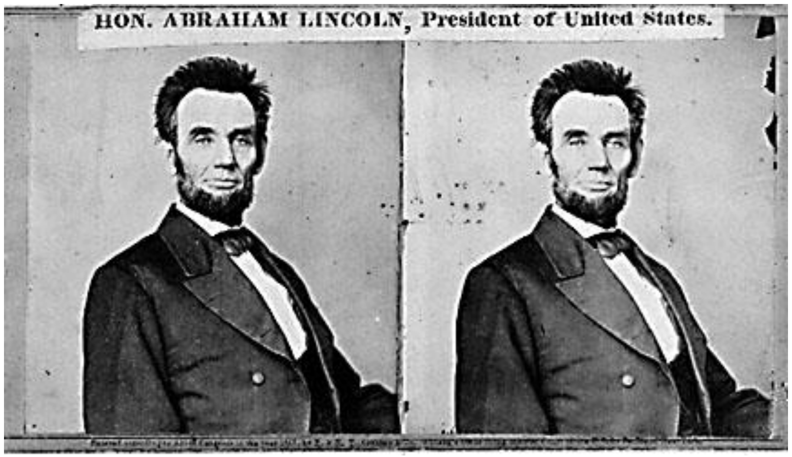


W = 20

- Smaller window
  - \+ More detail
  - – More noise

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

# Failures of correspondence search




Textureless surfaces

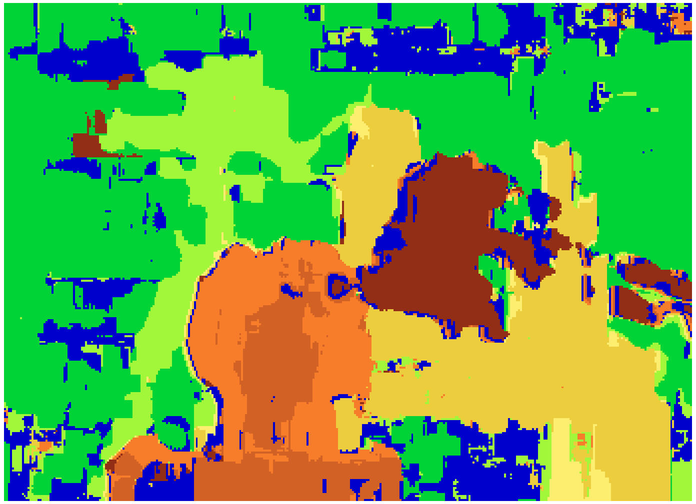
Occlusions, repetition

Non-Lambertian surfaces, specularities

# Results with window search

Data

Window-based matching



Ground truth

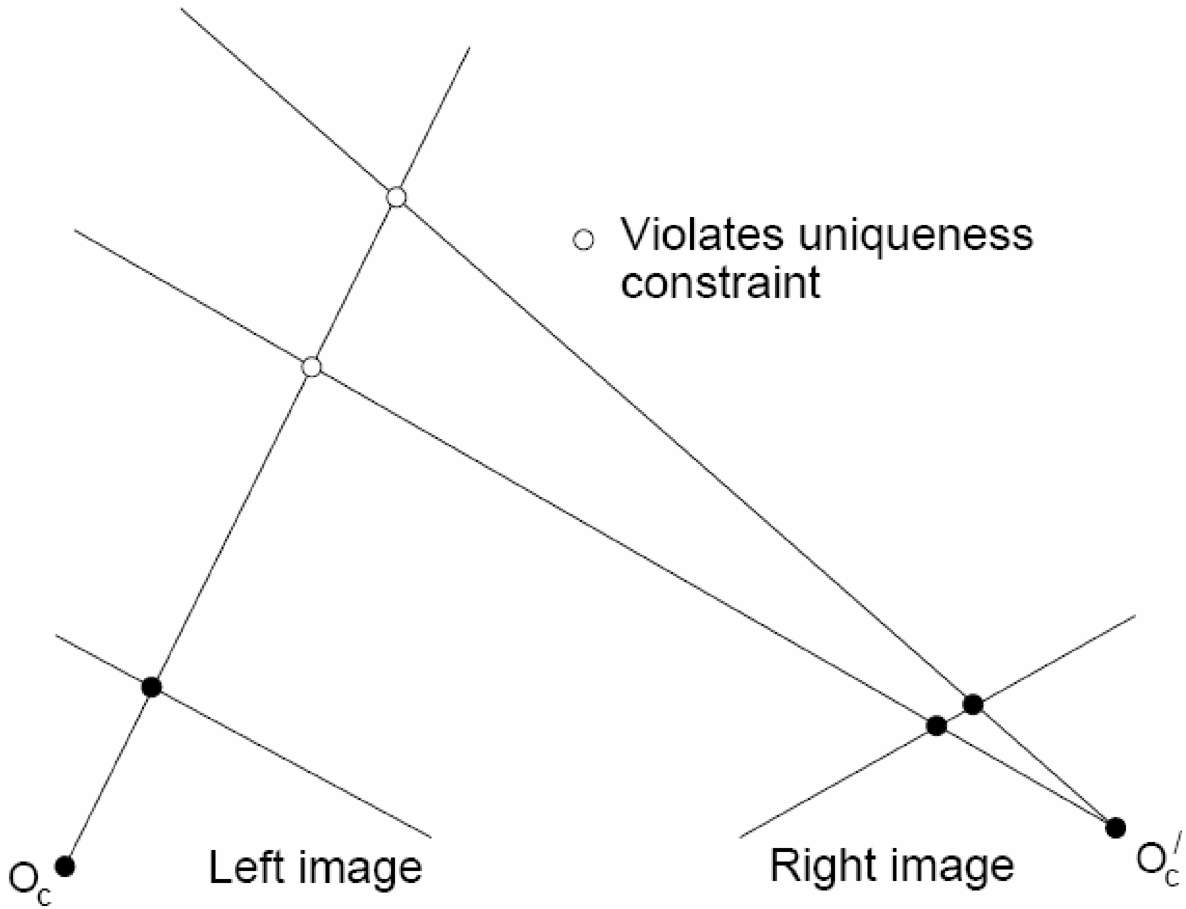
# How can we improve window-based matching?

- So far, matches are independent for each point

- What constraints or priors can we add?

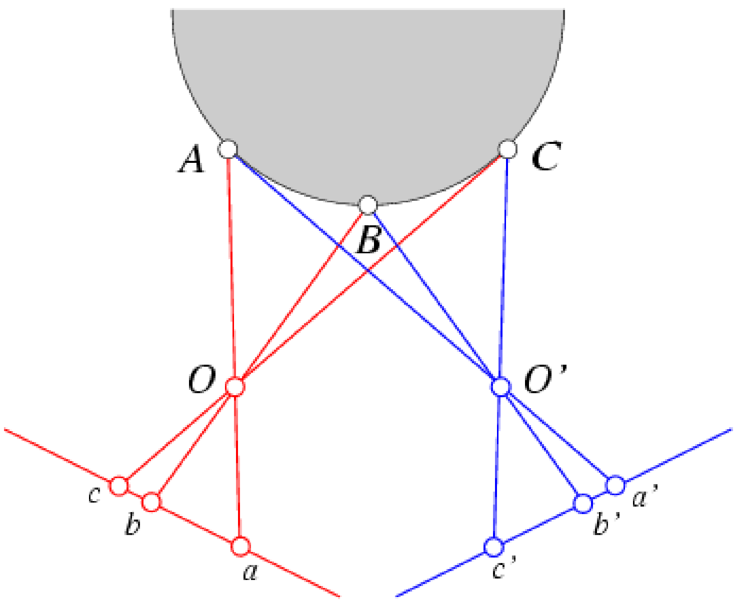
# Stereo constraints/priors

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



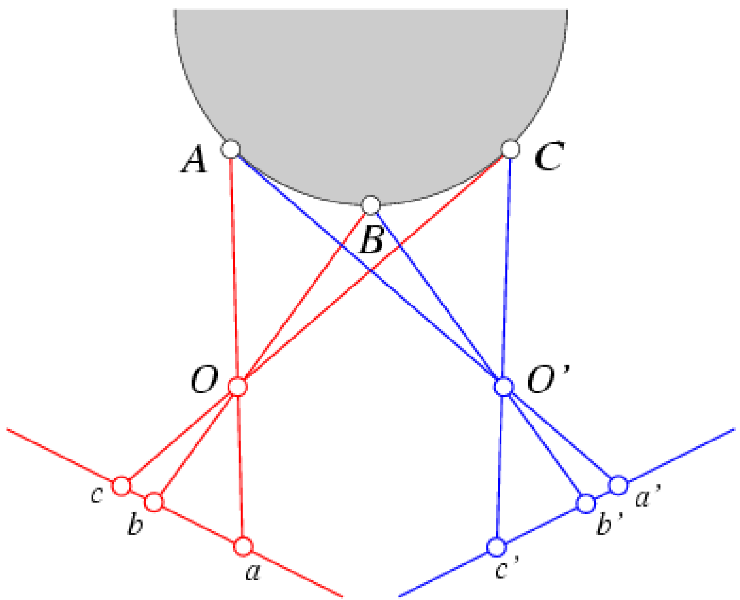
# Stereo constraints/priors

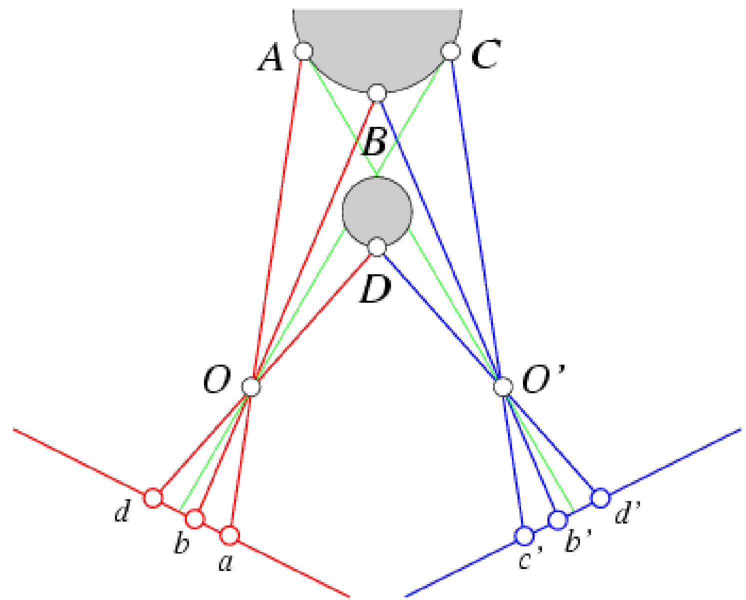
- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image
- Ordering
  - Corresponding points should be in the same order in both views



# 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



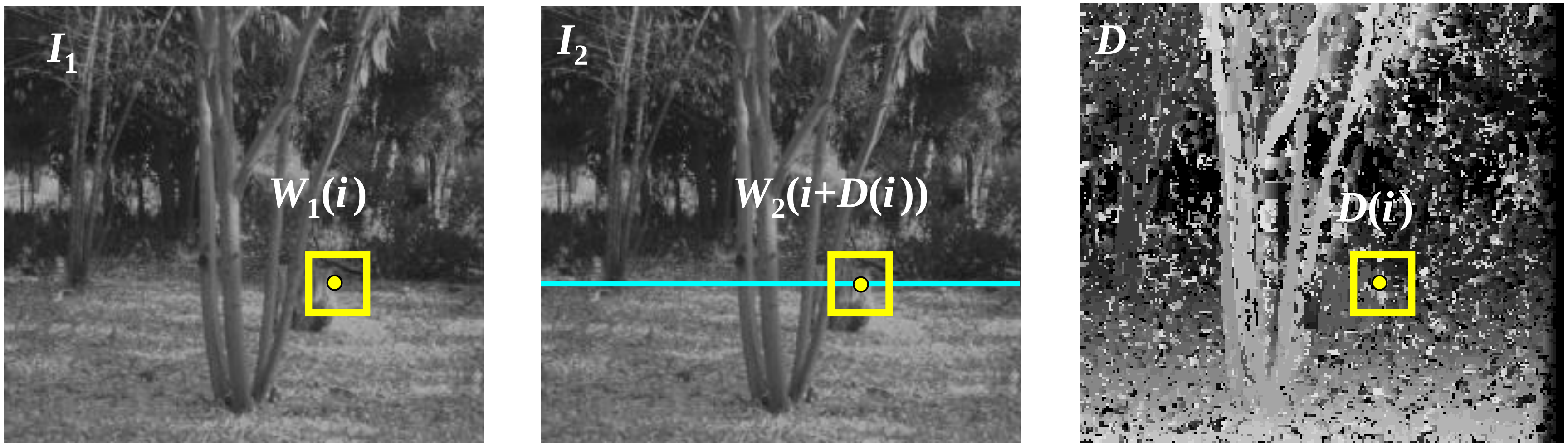




Ordering constraint doesn't hold

# 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 matching as energy minimization




$$E = E_{\text{data}}(D; I_1, I_2) + \beta E_{\text{smooth}}(D)$$

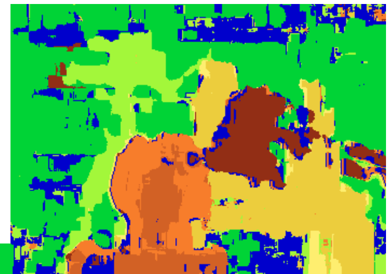
$$E_{\text{data}} = \sum_i \left( W_1(i) - W_2(i + D(i)) \right)^2 \qquad E_{\text{smooth}} = \sum_{\text{neighbors } i,j} \left\| D(i) - D(j) \right\|^2$$

- Energy functions of this form can be minimized using *graph cuts*

Y. Boykov, O. Veksler, and R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, PAMI 2001

Many of these constraints can be encoded in an energy function and solved using graph cuts

Before



Graph cuts

Ground truth

Y. Boykov, O. Veksler, and R. Zabih, [Fast Approximate Energy Minimization via Graph Cuts](), PAMI 2001

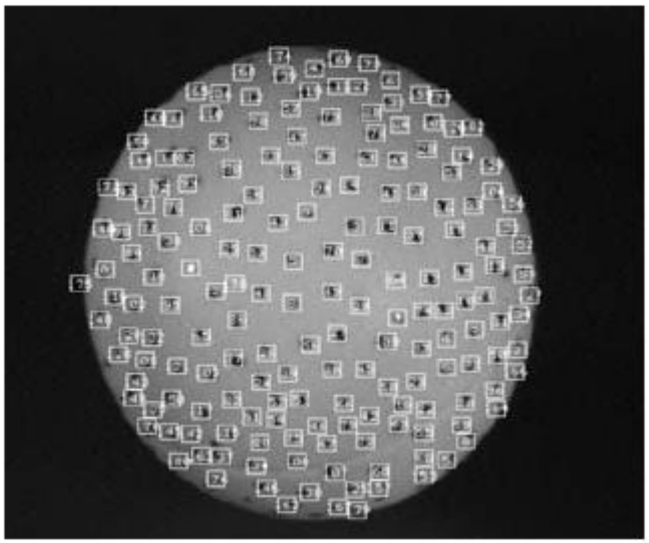
For the latest and greatest: http://www.middlebury.edu/stereo/

# Summary

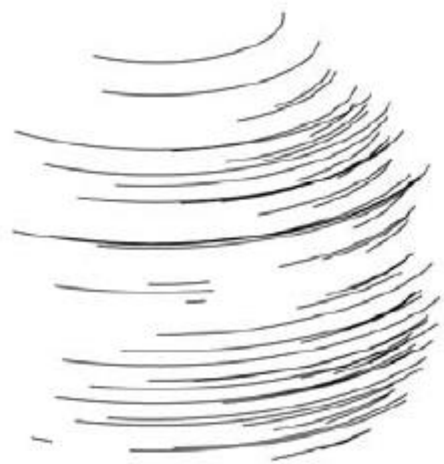
- Epipolar geometry
  - Epipoles are intersection of baseline with image planes
  - Matching point in second image is on a line passing through its epipole
  - Fundamental matrix maps from a point in one image to a line (its epipolar line) in the other
  - Can solve for F given corresponding points (e.g., interest points)
  - Can recover canonical camera matrices from F (with projective ambiguity)

- Stereo depth estimation
  - Estimate disparity by finding corresponding points along scanlines
  - Depth is inverse to disparity

# Next class: structure from motion



(a)



(b)

(c)