Computer Vision

Bogdan Alexe

bogdan.alexe@fmi.unibuc.ro

University of Bucharest, 2nd semester, 2020-2021

Project 1 - deadline soon

- minor modification regarding the data released in the “train” folder:
  - renamed some files (ground-truth annotations files)
  - added all the data for task 3
    - images 1500 × 1500 generated on the computer
  - fixed some wrong annotations for task 2

- added another folder “evaluation”

- refine the pdf file describing the project
  - please read it carefully

Project 1 - deadline soon

- added another folder “evaluation”

The directory *evaluation* shows how the evaluation will take place on the test data after the deadline. It contains the following subdirectories:

- *fake_test* - this directory exemplifies how the test data will be released, keeping the structure of the previously described *train* directory;
- *submission_files* - this directory exemplifies the format of the results data that we expect from you to submit in the second stage. You will have to send your results in this format, uploading a zip archive of a folder similar with the one called *Alexe_Bogdan_407*;
- *evaluation* - this directory contains code that we will use to evaluate your results using the ground-truth data. Make sure that his code will run on your submitted files. The ground-truth data will be released after you send us your results.

Project 1 – what to submit?

Deadlines: Submit a zip archive containing your code and a pdf file describing your approach until Saturday, 8^{th} of May using the following link https://tinyurl.com/CV-2021-PROJECT1-SUBMISSIONS. Notice that this is a hard deadline, no projects will be accepted after the deadline. Your code should include a README file (see the example in the materials for this project) containing the following information: (i) the libraries required to run the project including the full version of each library; (ii) indications of how to run the solution for each task and where to look for the output file. Students are allowed to submit solution in the format of Jupyter Notebbok files with the code being commented. This can replace the pdf file describing their approach. Students who do not describe their approach (using comments thoughout the code or a pdf file) will incur a penalty of 0.5 points.

On Sunday 9^{th} of May we will make available the test data. You will have to run your system on the test images provided by us and upload your results in the same day as a zip archive using the following link https://tinyurl.com/CV-2021-PROJECT1-RESULTS.

Course structure

1. **Features and filters: low-level vision**

   Linear filters, color, texture, edge detection

2. **Grouping and fitting: mid-level vision**

   Fitting curves and lines, robust fitting, RANSAC, Hough transform, segmentation

3. **Multiple views**

   Local invariant feature and description, epipolar geometry and stereo, object instance recognition

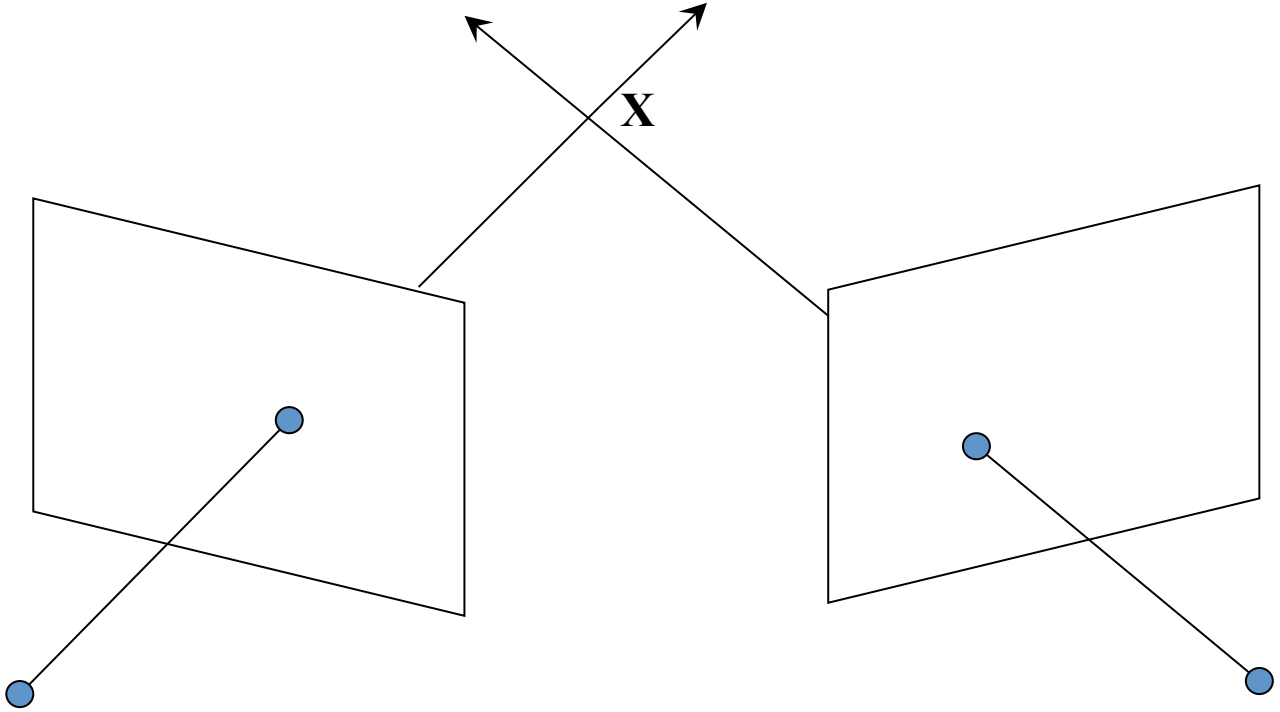
4. **Object Recognition: high – level vision**

   Object classification, object detection, part based models, bovw models

5. **Video understanding**

   Object tracking, background subtraction, motion descriptors, optical flow

Two–View Geometry

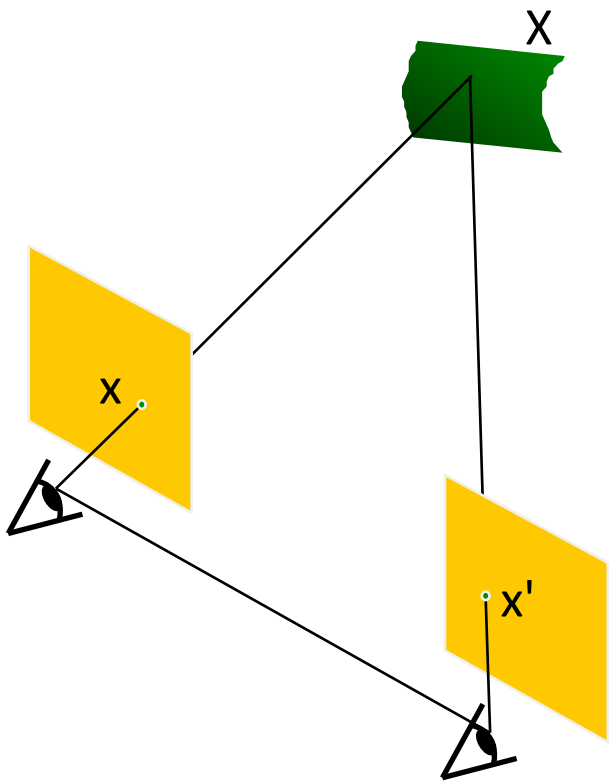


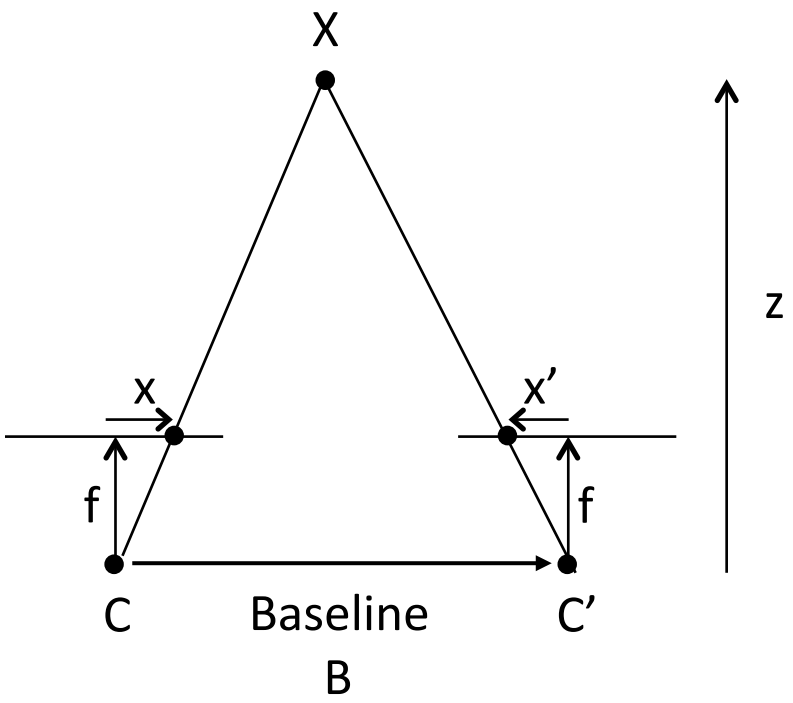
Two-View Geometry

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

Depth from Stereo

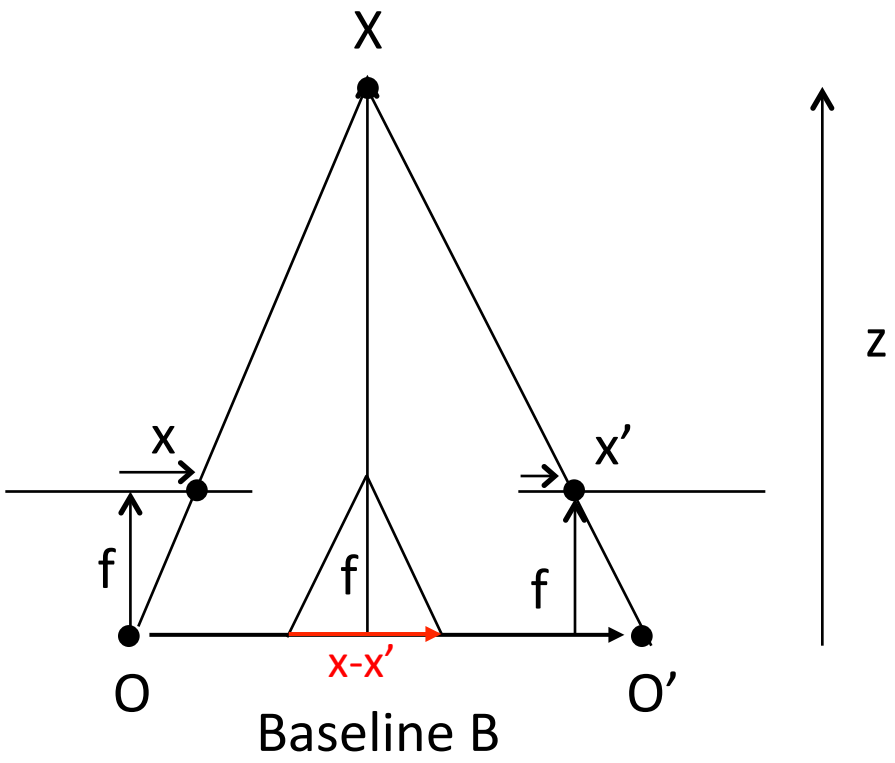
- Goal: recover depth by finding image coordinate x' that corresponds to x





Depth from Stereo

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

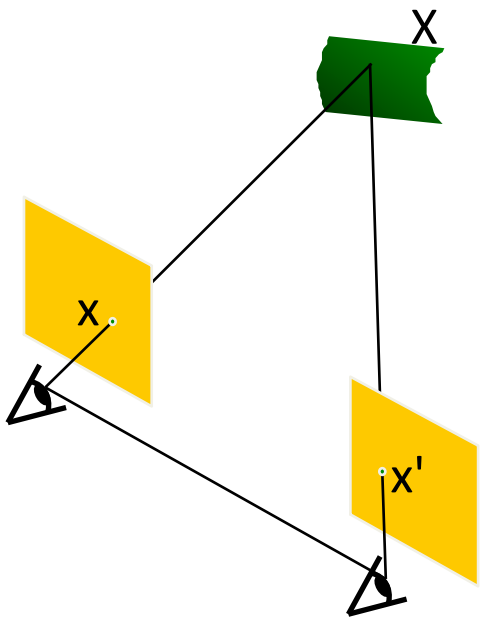


$$disparity = x - x' = \frac{B \cdot f}{z}$$

Disparity is inversely proportional to depth.

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’?

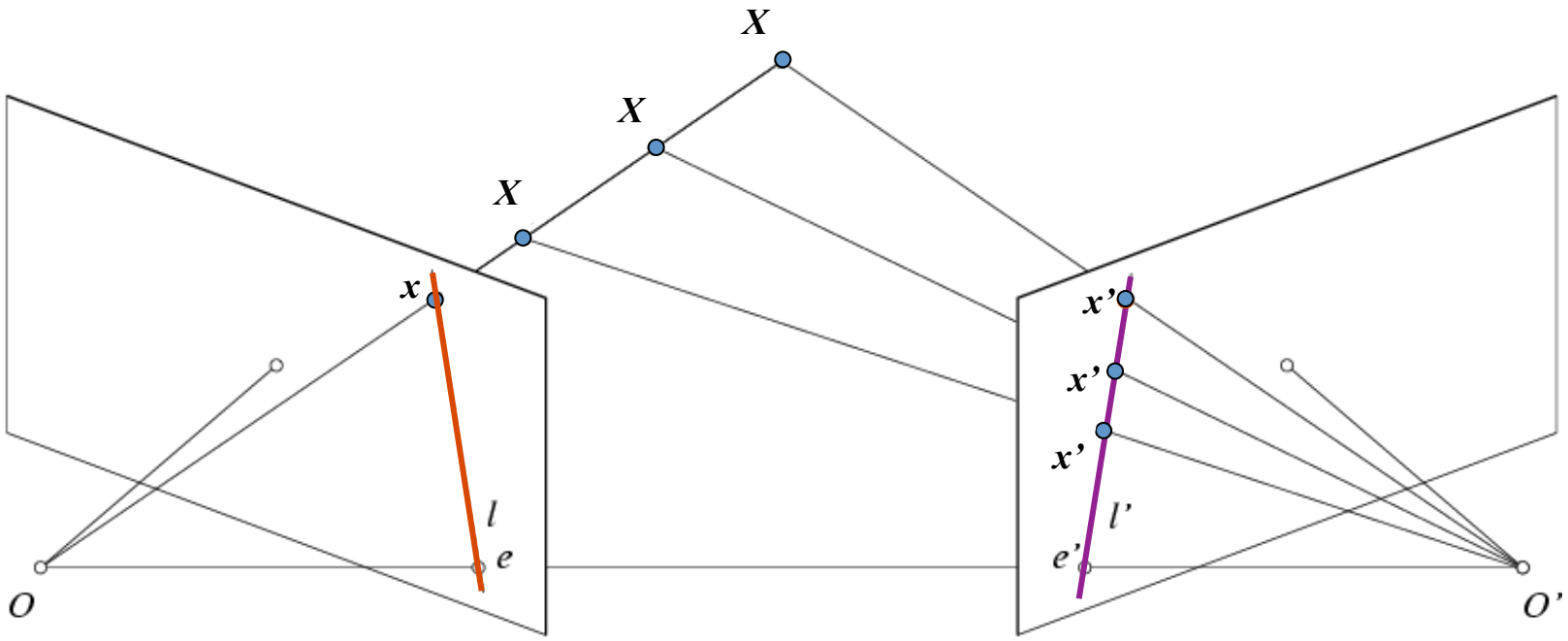


Correspondence Problem

- 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?

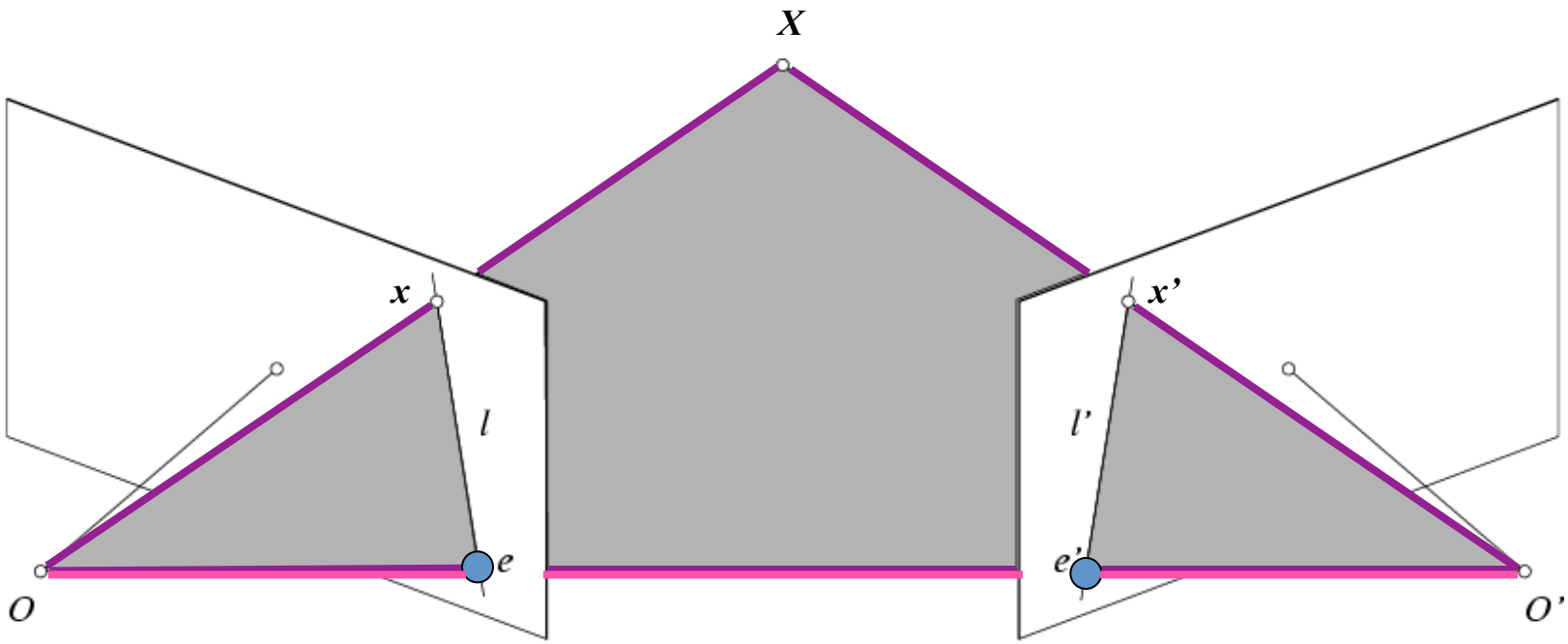
Key idea: Epipolar constraint



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



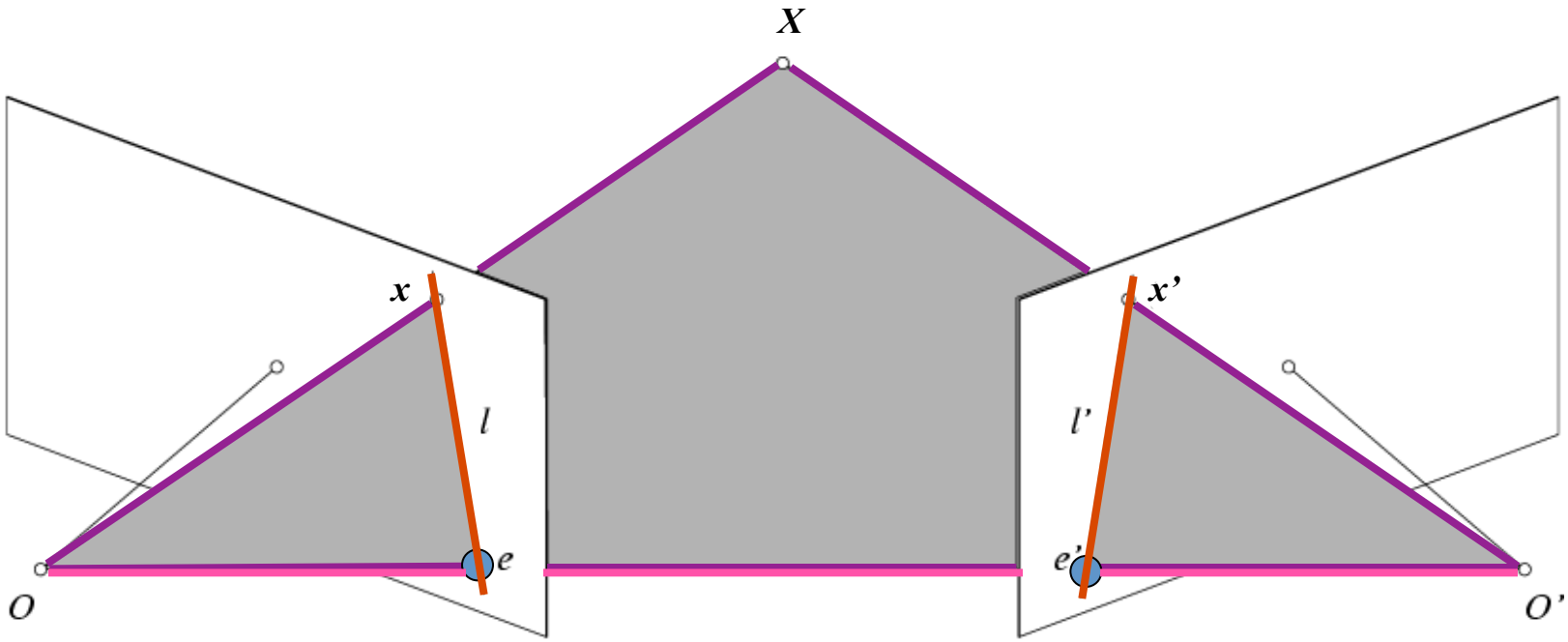
- **Baseline** – line connecting the two camera centers
- **Epipoles**

= intersections of baseline with image planes

= projections of the other camera center

- **Epipolar Plane** – plane containing baseline (1D family)

Epipolar geometry: notation



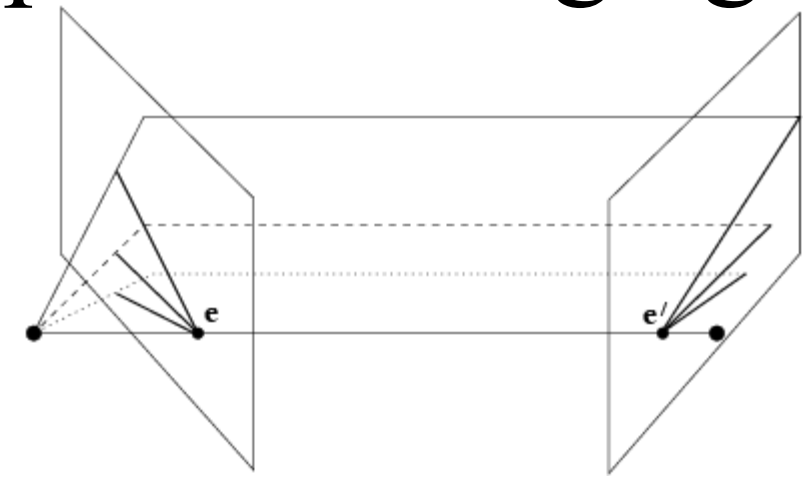
- **Baseline** – line connecting the two camera centers
- **Epipoles**

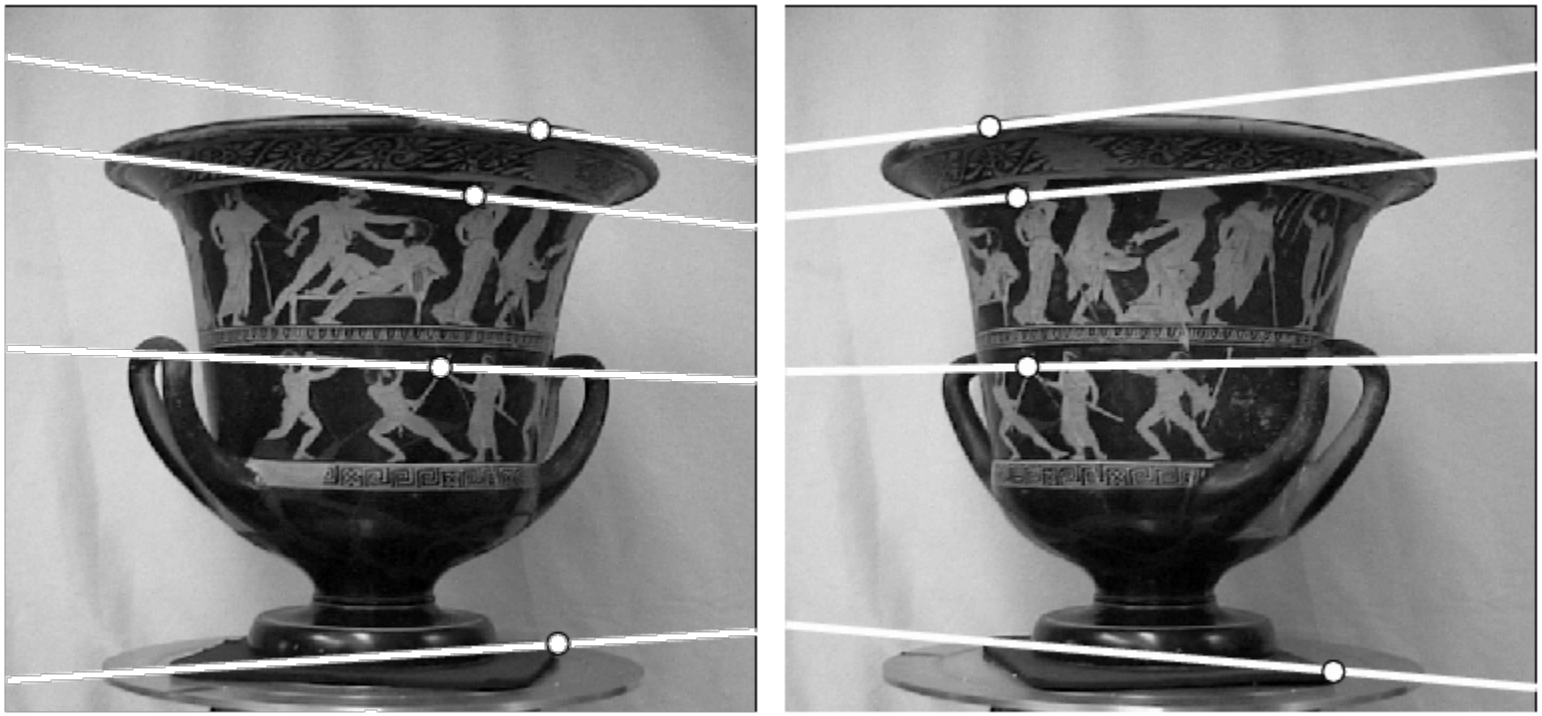
= intersections of baseline with image planes

= projections of the other camera center

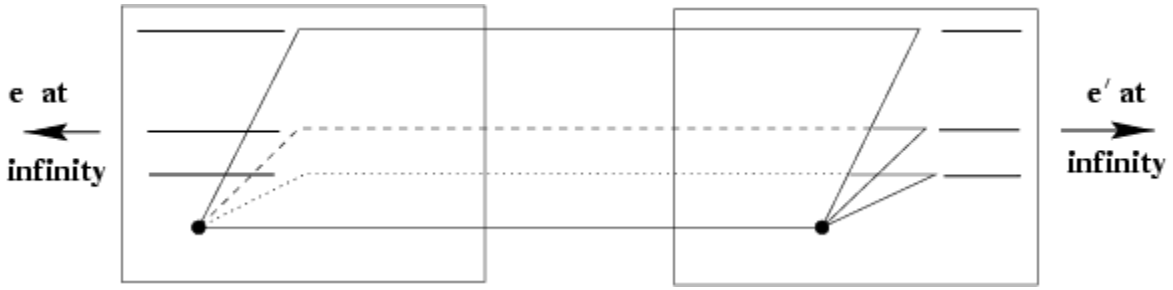
- **Epipolar Plane** – plane containing baseline (1D family)
- **Epipolar Lines** - intersections of epipolar plane with image planes (always come in corresponding pairs)

Example: Converging cameras





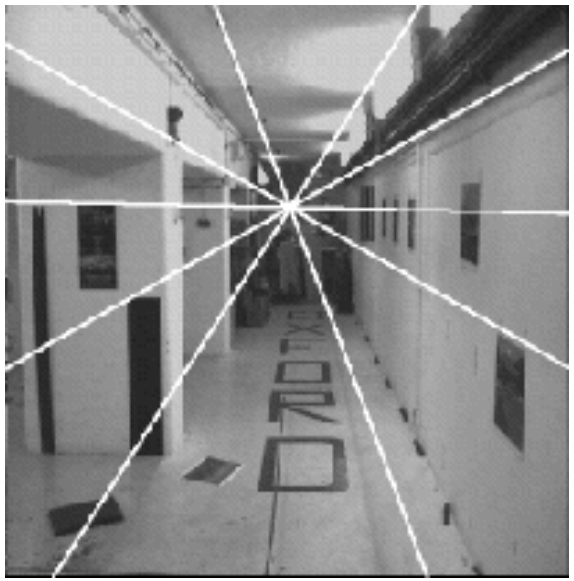
Example: Motion parallel to image plane

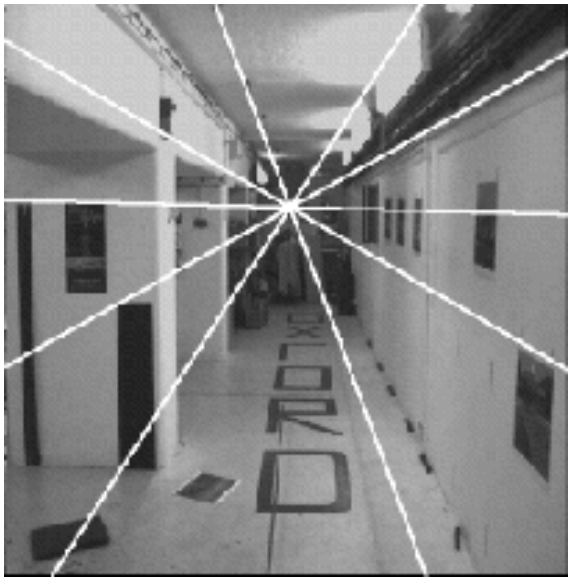


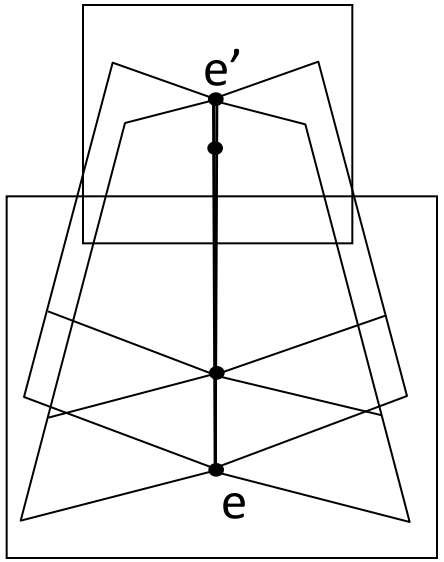
Example: Forward motion

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”

Camera parameters $\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$

- Intrinsic parameters
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew (non-rectangular pixels)*
  - *Radial distortion*

$$\mathbf{K} = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & & \beta_x \\ & \alpha_y & \beta_y \\ & & 1 \end{bmatrix}$$

- Extrinsic parameters
  - Rotation and translation relative to world coordinate system
  - What is the projection of the camera center?

$$\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}}\end{bmatrix}$$

coords. of camera center in world frame

$$\mathbf{PC} = \mathbf{K}\begin{bmatrix}\mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}}\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{C}} \\ 1\end{bmatrix} = 0$$

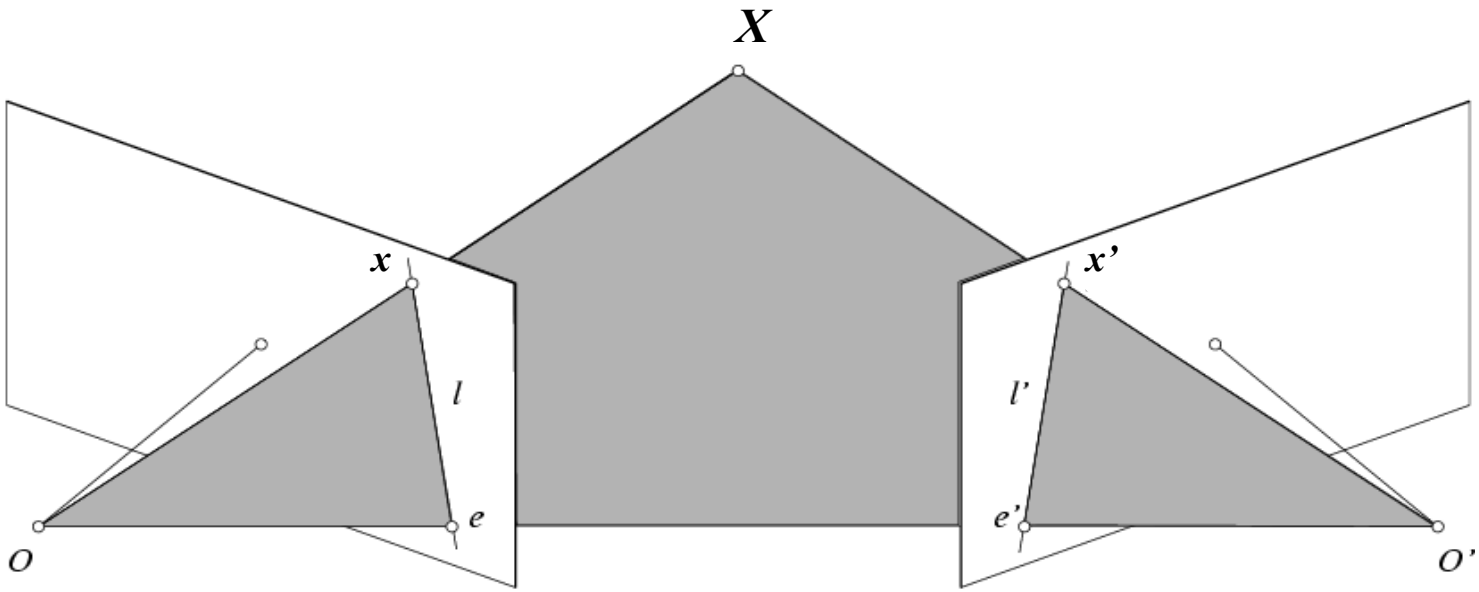
The camera center is the *null space* of the projection matrix!

Camera parameters $\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

$$\begin{bmatrix} \alpha_x & s & \beta_x \\ 0 & \alpha_y & \beta_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}$$

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

for some scale factor

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

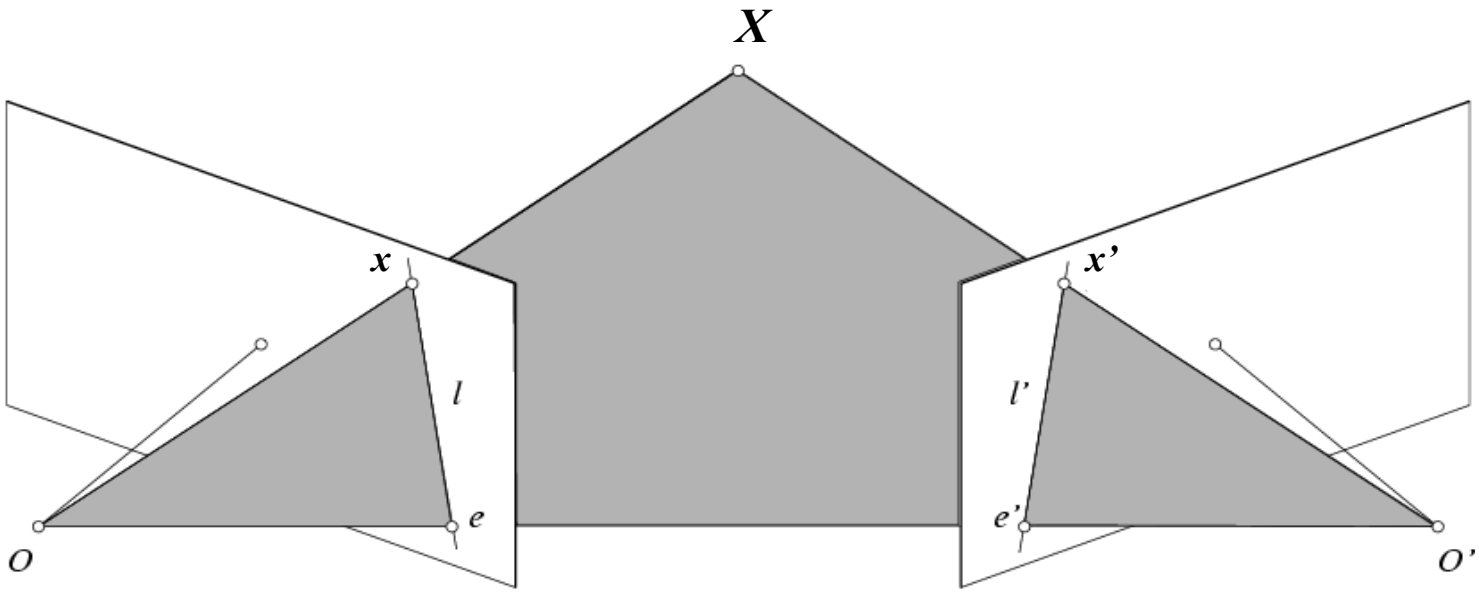
Homogeneous 2d point (3D ray towards X)

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:

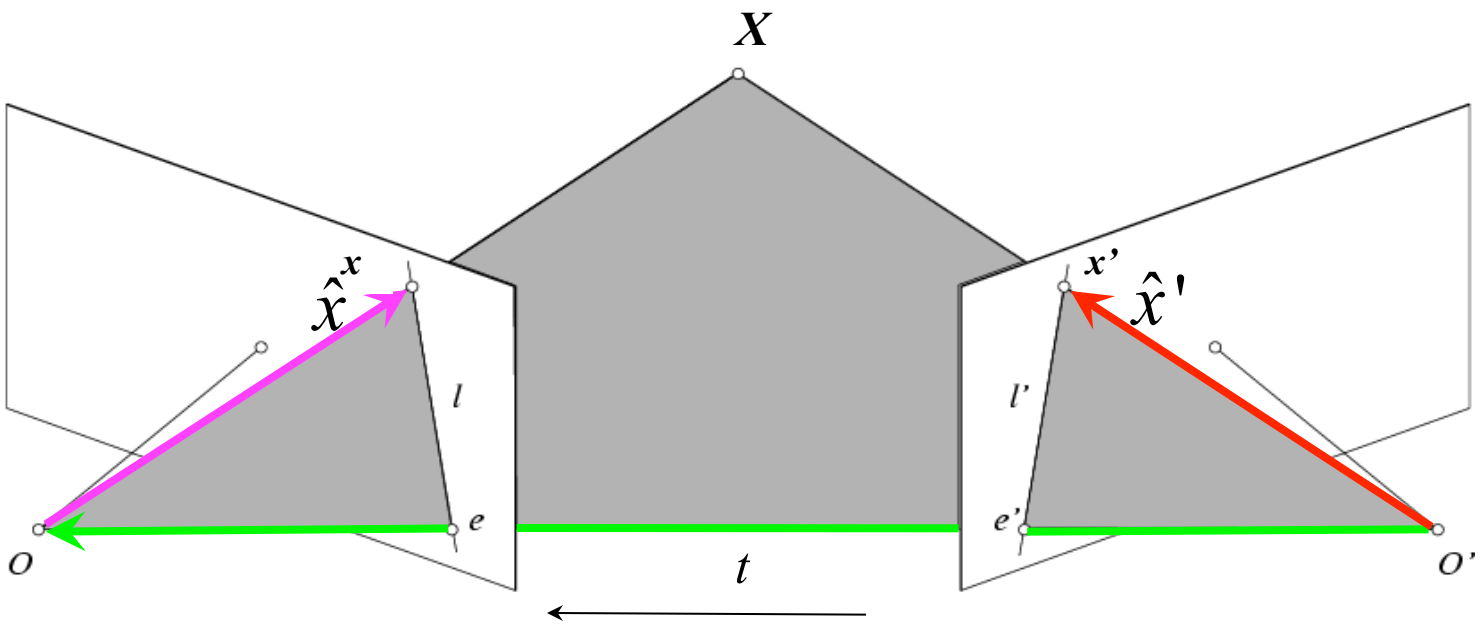
1. Convert to normalized coordinates by pre-multiplying all points with the inverse of the calibration matrix; set first camera's coordinate system to world coordinates
2. Define some R and t that relate X to X' as below

for some scale factor

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

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

Epipolar constraint: Calibrated case

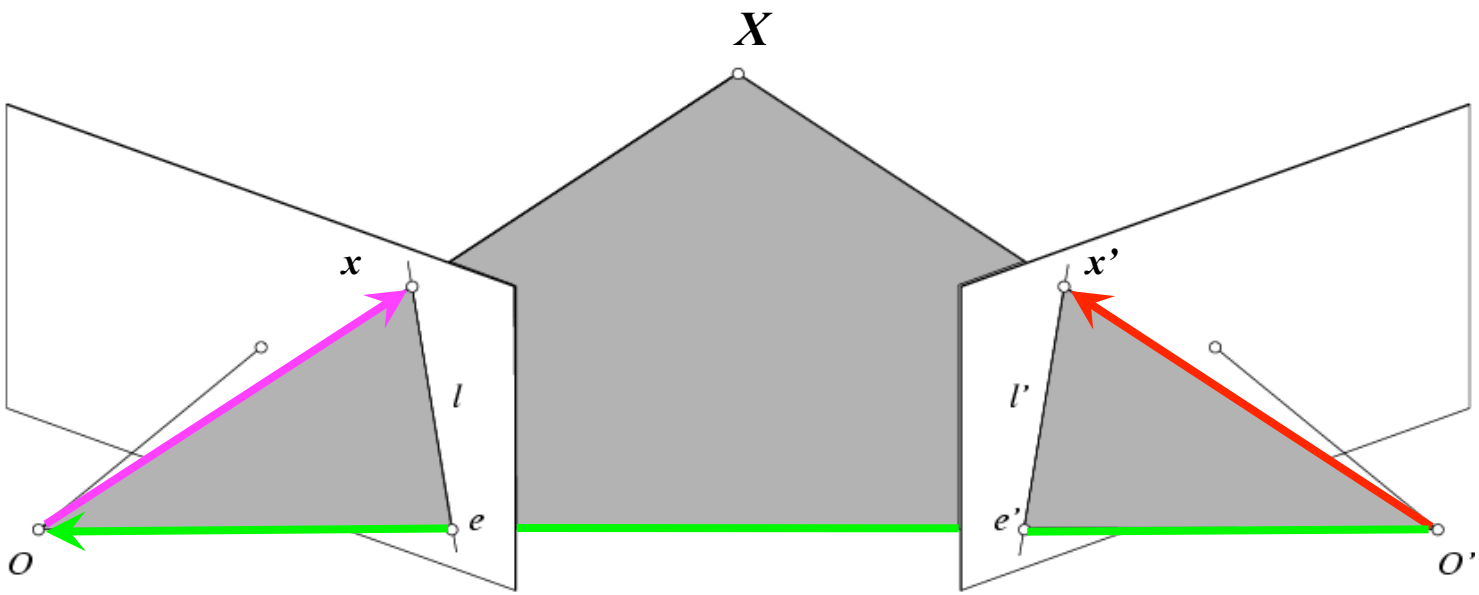


$$\hat{x} = K^{-1}x = X \qquad \hat{x}' = K'^{-1}x' = X'$$

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

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

Essential matrix



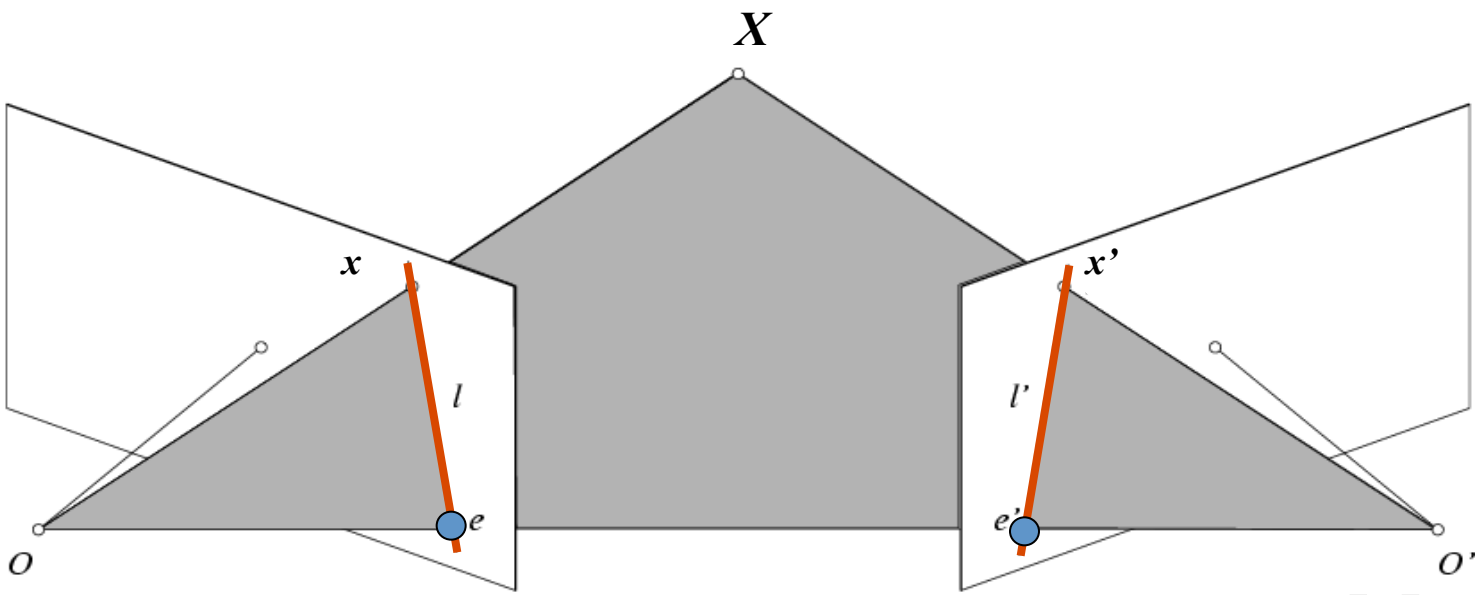
$$\hat{x}\cdot[t\times(R\hat{x}')]=0 \quad \Rightarrow \quad \hat{x}^T E\hat{x}'=0 \quad \text{with} \quad E=[t]_\times R$$

$$[t]_\times=\begin{bmatrix}0 & -t_3 & t_2\\ t_3 & 0 & -t_1\\ -t_2 & t_1 & 0\end{bmatrix}$$

Essential Matrix
(Longuet-Higgins, 1981)

$[t]_\times$ is the skew symmetric matrix of $t=(t_1,t_2,t_3)$

Properties of the Essential matrix



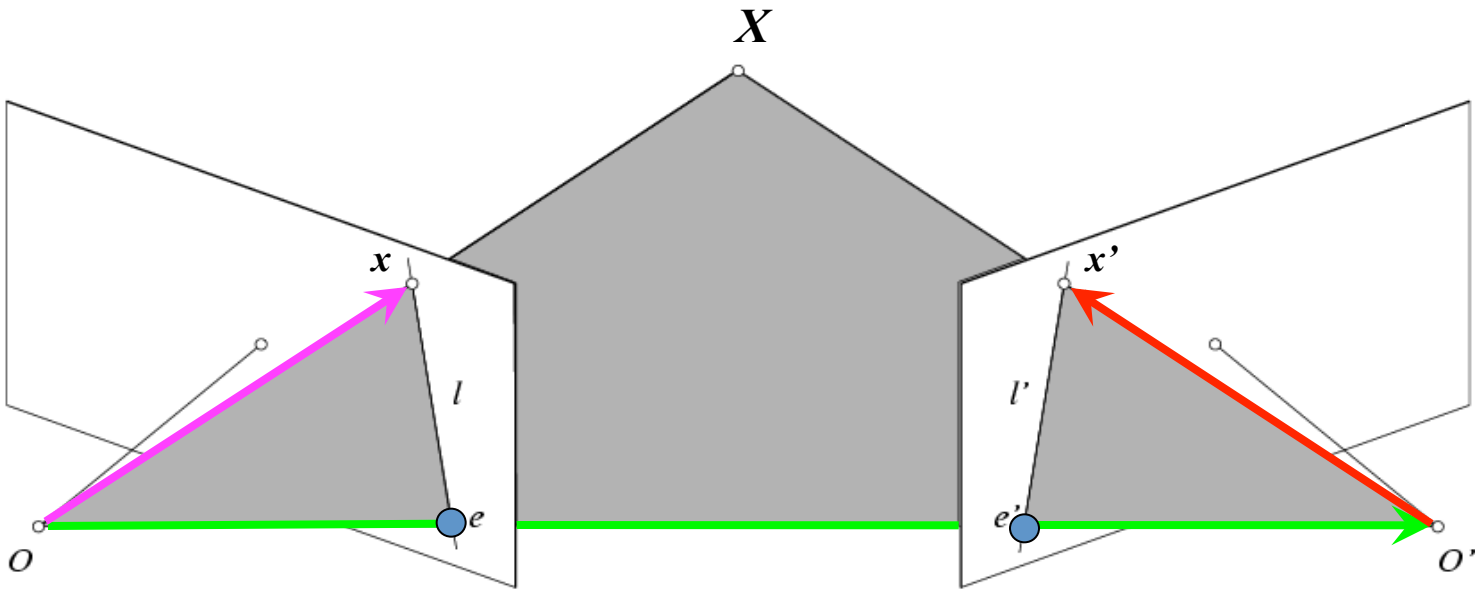
$$\hat{x}\cdot[t\times(R\hat{x}')]=0 \quad \Rightarrow \quad \hat{x}^T E\hat{x}'=0 \quad \text{with} \quad E=[t]_\times R$$

Drop ^ below to simplify notation

- $E\,x'$ 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)

Skew-symmetric matrix

Epipolar constraint: Uncalibrated case



- If we don't know K and K', then we can write the epipolar constraint in terms of *unknown* normalized coordinates:

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

The Fundamental Matrix

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

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

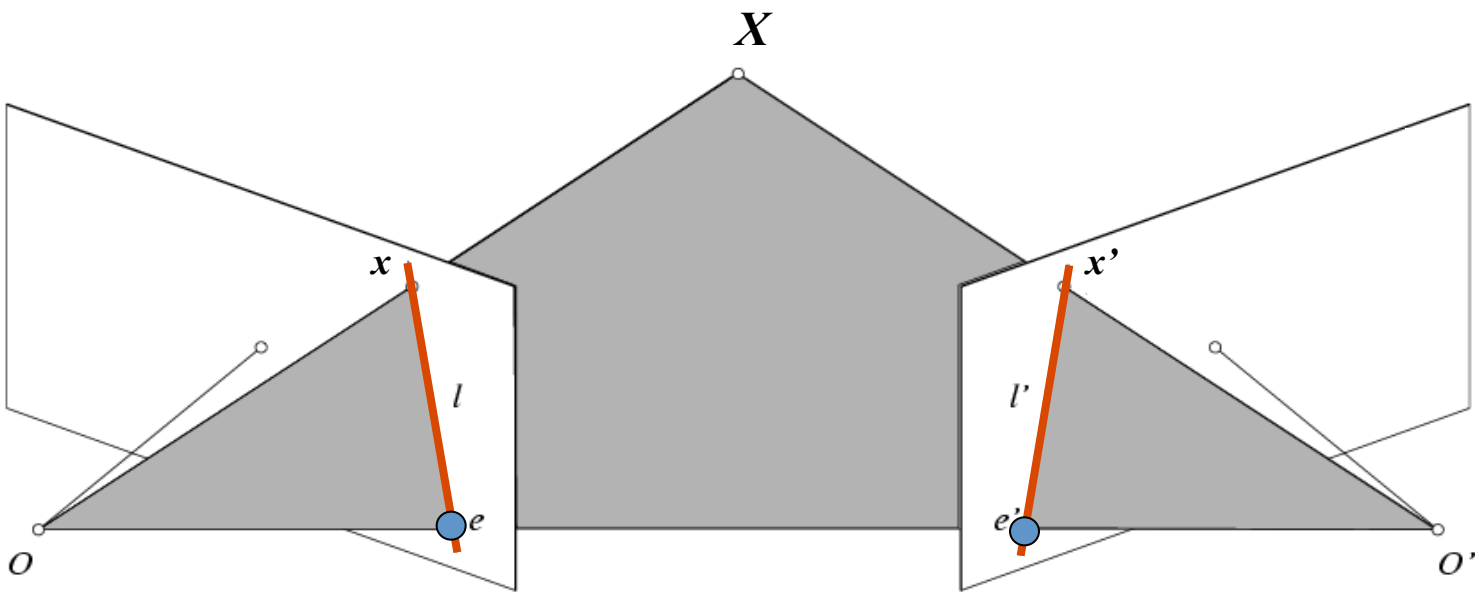
$$\hat{x} = K^{-1} x$$

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

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

Fundamental Matrix
(Faugeras and Luong, 1992)

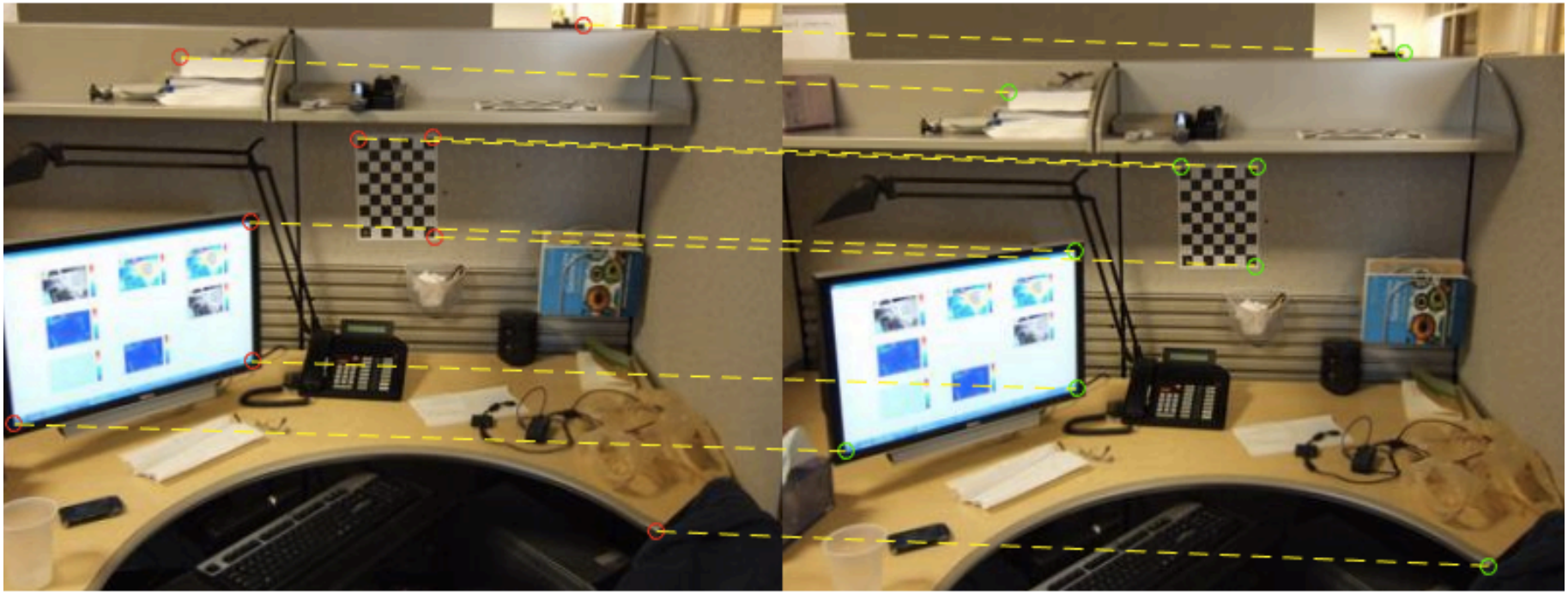
Properties of the Fundamental matrix



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

- $F\,x'$ 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



The eight-point algorithm

$$\boldsymbol{x} = (u, v, 1)^T, \quad \boldsymbol{x}' = (u', v', 1) \qquad x^T F x' = 0$$

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

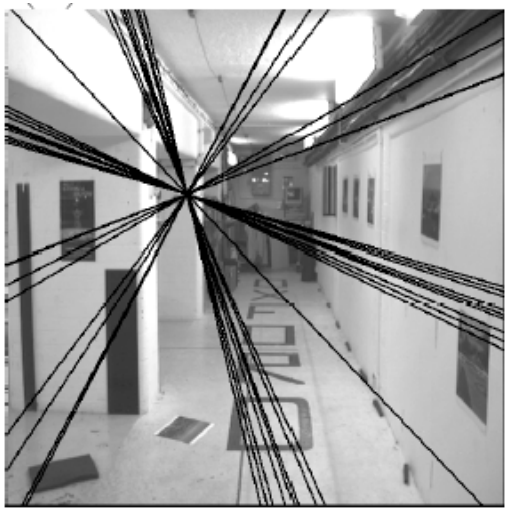
$$\begin{bmatrix} u'u & u'v & u' & v'u & v'v & v' & u & v & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

Solve homogeneous linear system using eight or more matches

Fundamental matrix has rank 2: det(F) = 0

Enforce rank-2 constraint (take SVD of ***F*** and throw out the smallest singular value)

Left: uncorrected F – epipolar lines are not coincident



Right: epipolar lines from corrected F

The 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

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

- 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 $\boldsymbol{F}$ from the normalized points
- Enforce the rank-2 constraint (for example, take SVD of $\boldsymbol{F}$ and throw out the smallest singular value)
- Transform fundamental matrix back to original units: if $\boldsymbol{T}$ and $\boldsymbol{T}'$ are the normalizing transformations in the two images, than the fundamental matrix in original coordinates is $\boldsymbol{T}'^{T}\,\boldsymbol{F}\,\boldsymbol{T}$

The (normalized) eight-point algorithm

1. Solve a system of homogeneous linear equations (for normalized points data)
   a. Write down the system of equations
   b. Solve **f** from A**f**=**0** using SVD

   Matlab:

```
[U, S, V] = svd(A); %A = U*S*V'
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';
```

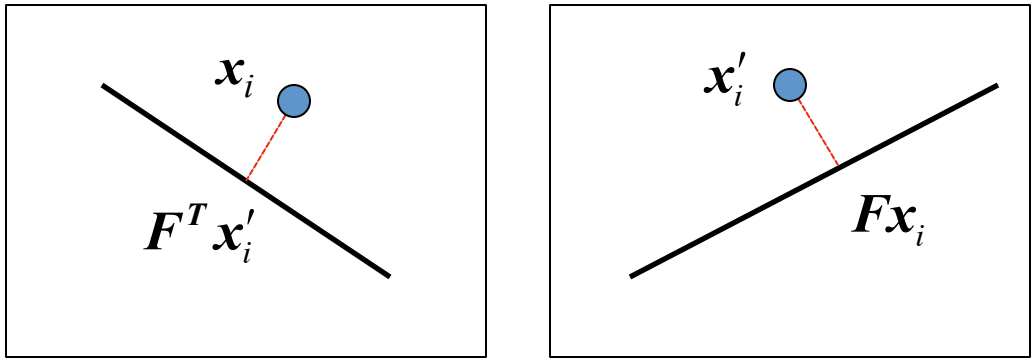
Nonlinear estimation

- Linear estimation minimizes the 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 $\boldsymbol{F}^T\boldsymbol{x}'_i$):

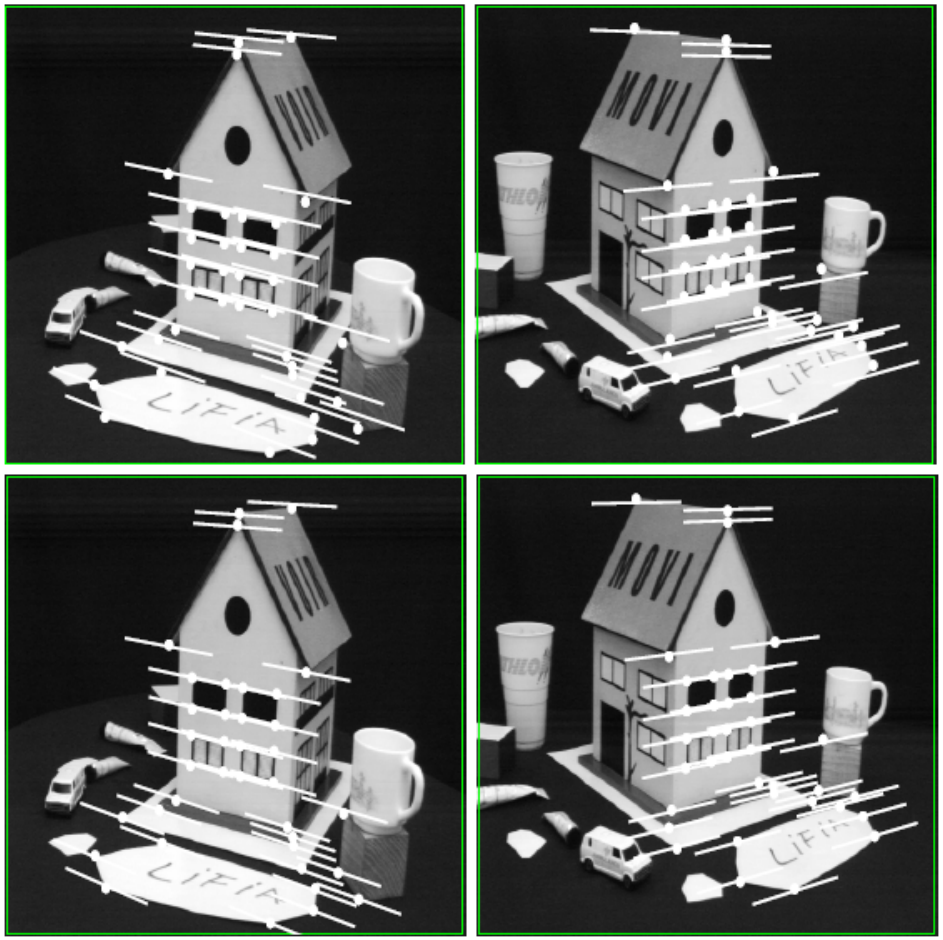
$$\sum_{i=1}^{N} (\boldsymbol{x}_i'^T \boldsymbol{F}\boldsymbol{x}_i)^2$$

- Nonlinear approach: minimize sum of squared *geometric* distances

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

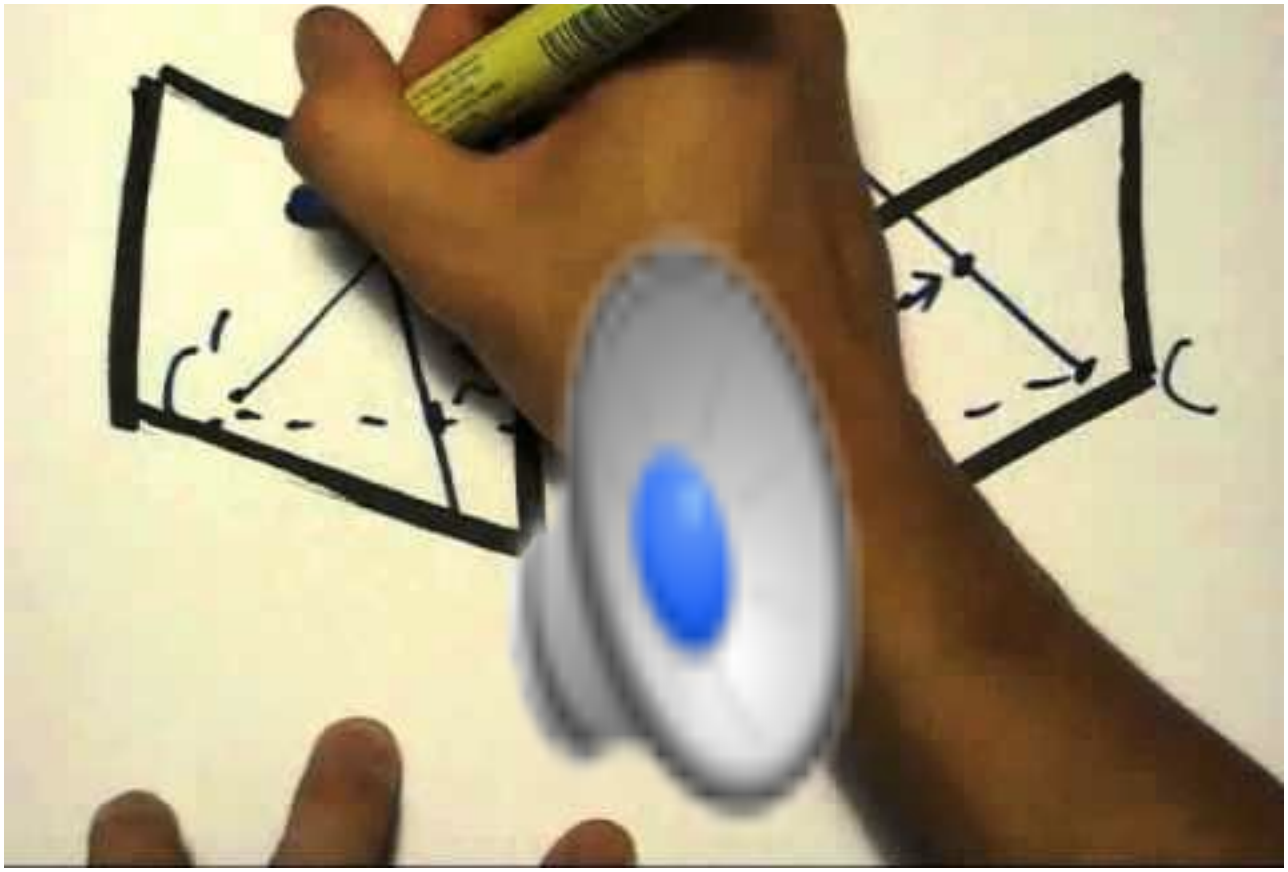


Comparison of estimation algorithms



	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

The Fundamental Matrix Song



http://danielwedge.com/fmatrix/

Moving on to stereo…

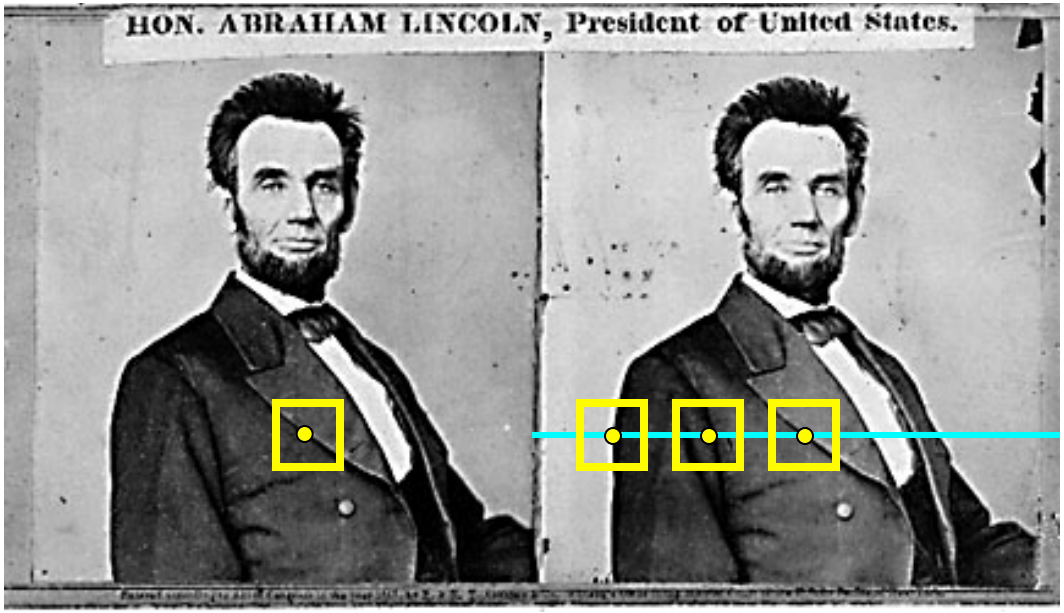
Fuse a calibrated binocular stereo pair to produce a depth image

image 1

image 2

Dense depth map

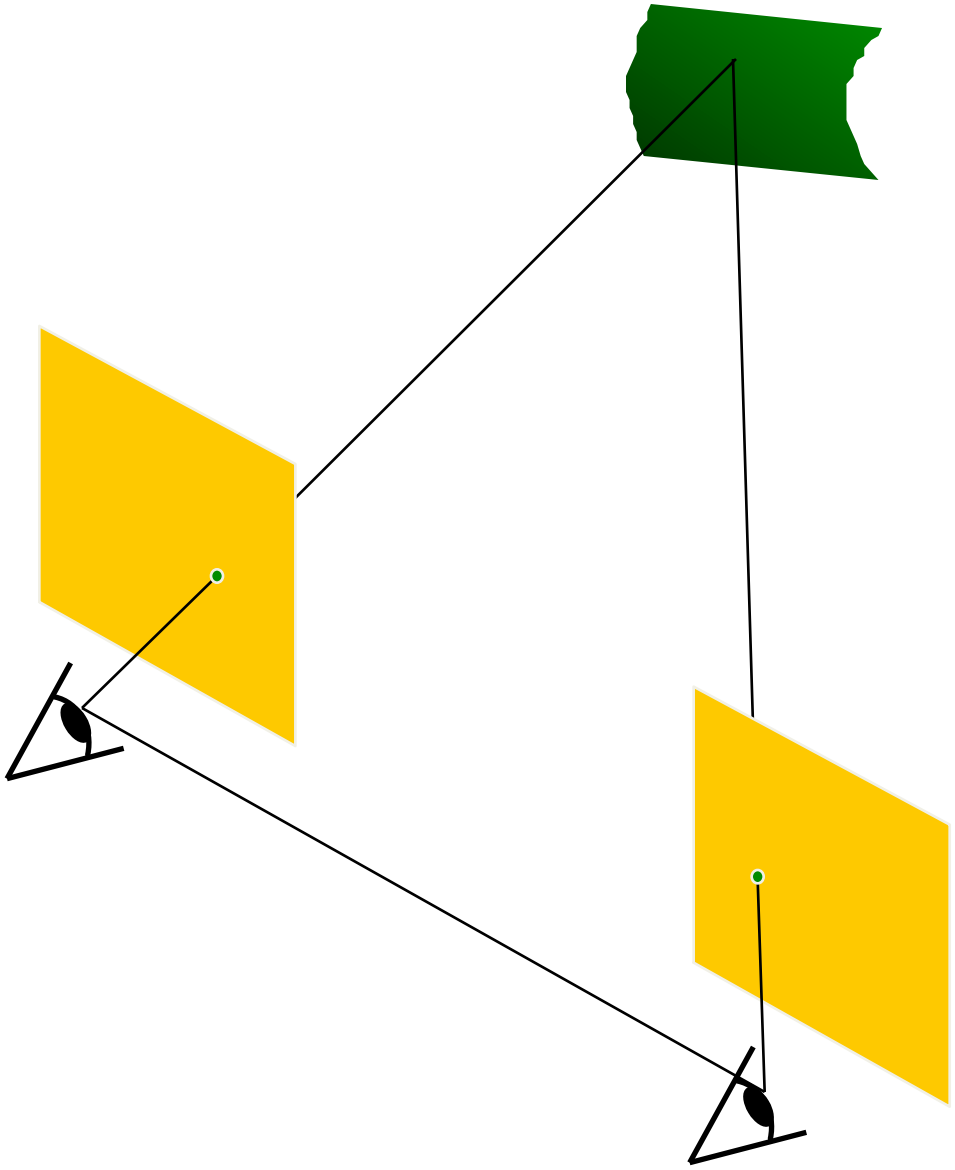
Basic stereo matching algorithm



- 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

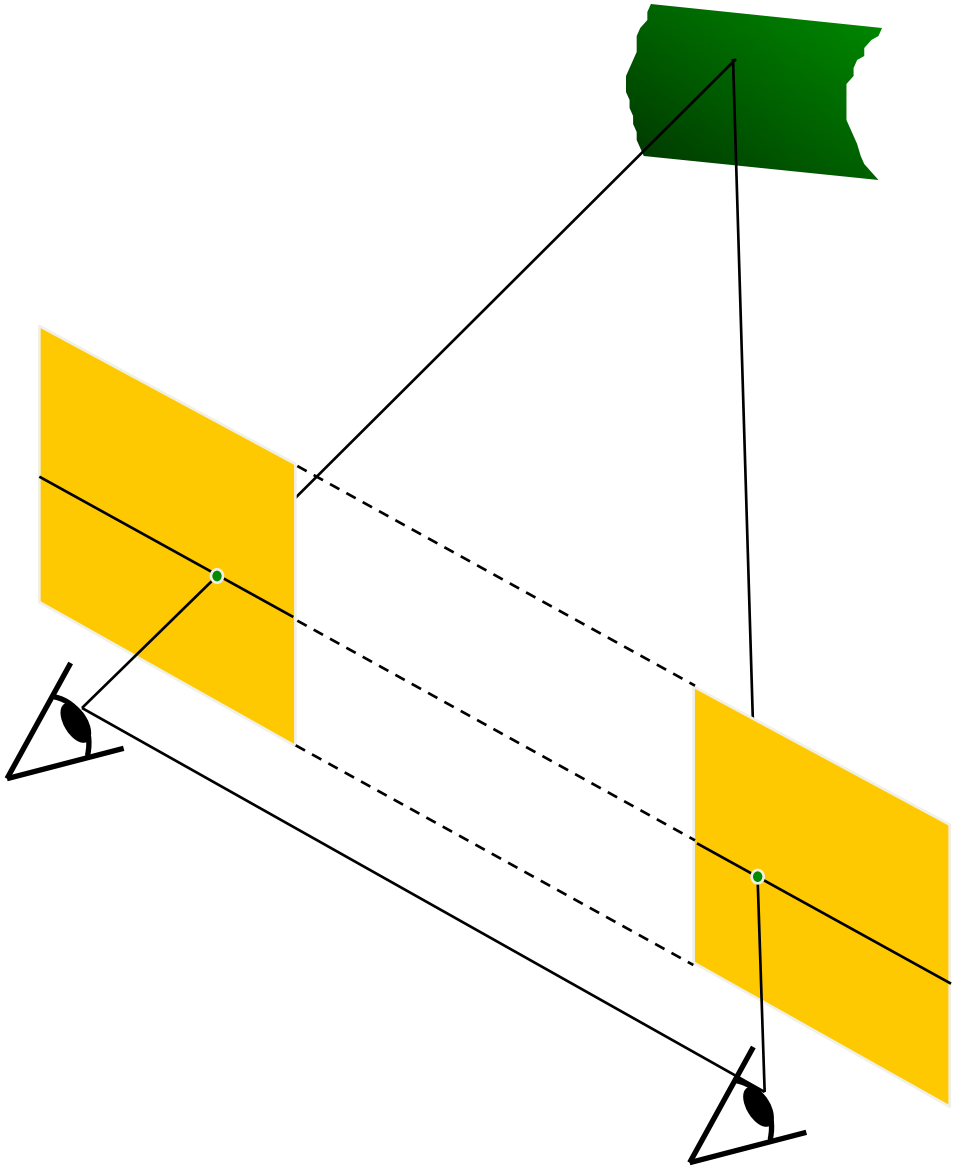
- Simplest case: epipolar lines are scanlines
  - When does this happen?

Simplest Case: Parallel images



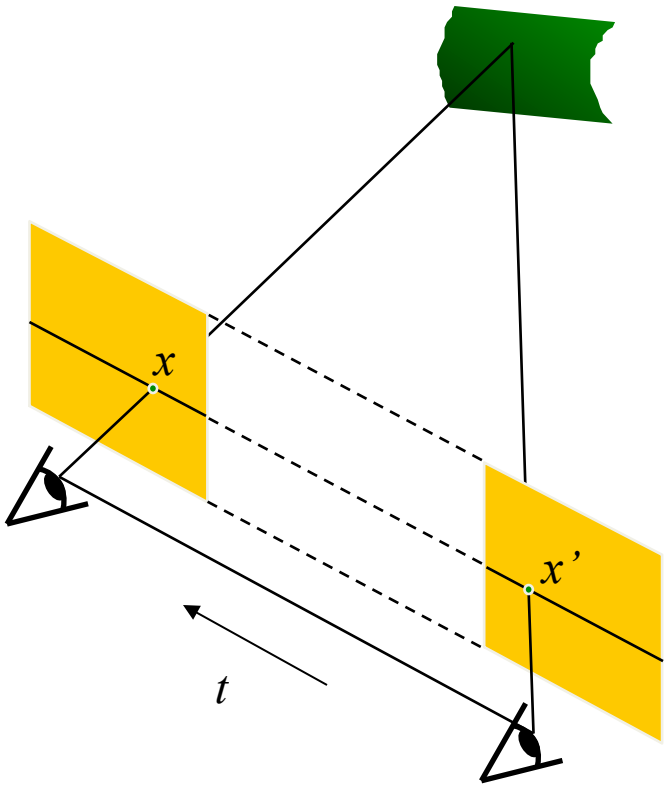
- Image planes of cameras are parallel to each other and to the baseline
- Camera centers are at same height
- Focal lengths are the same

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:

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

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

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

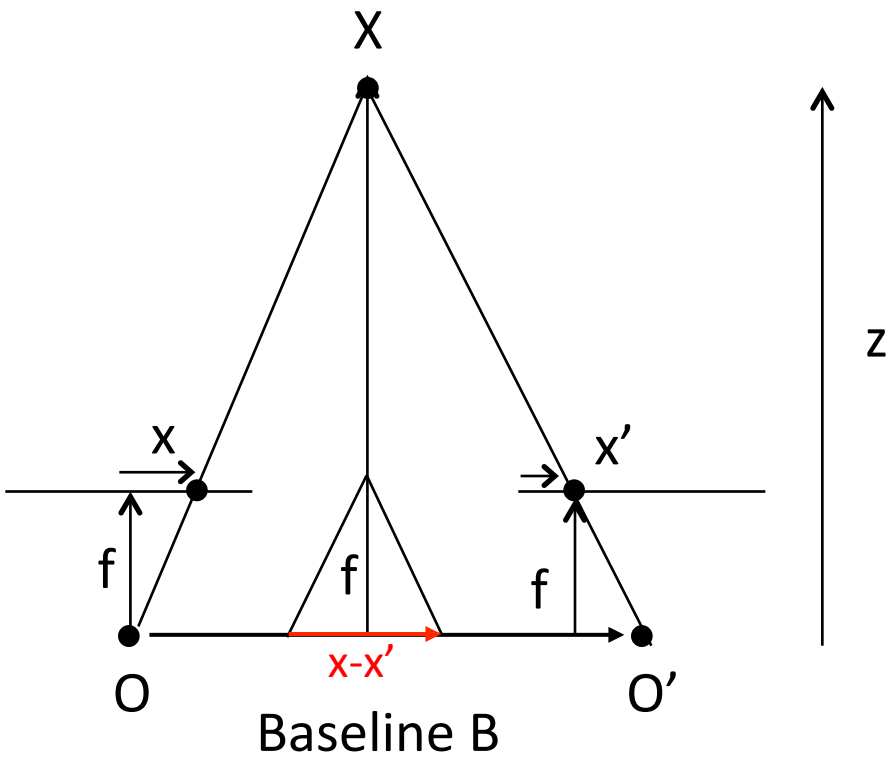
$$\begin{pmatrix} u & v & 1 \end{pmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0$$

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

The y-coordinates of corresponding points are the same

Depth from disparity

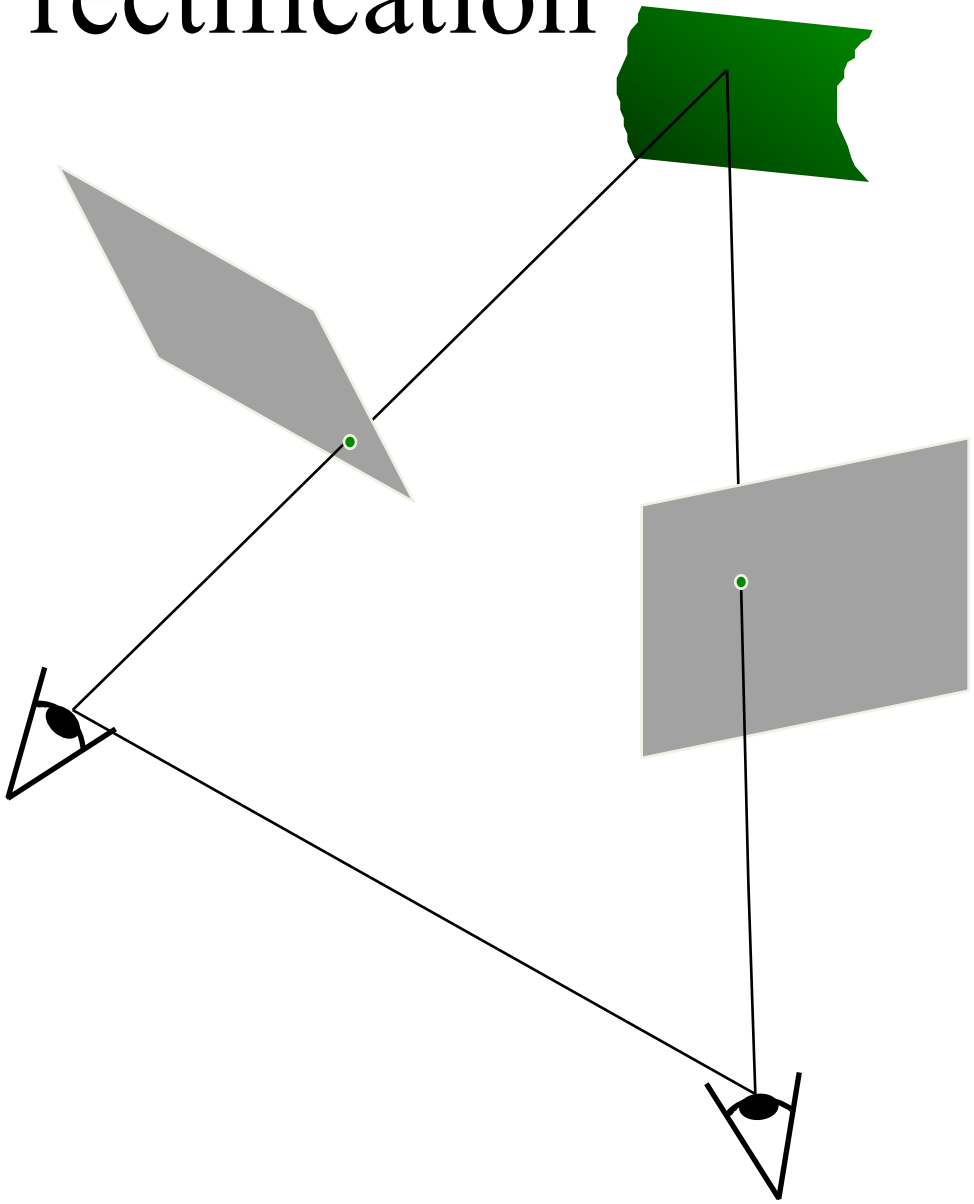
$$\frac{x - x'}{O - O'} = \frac{f}{z}$$



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

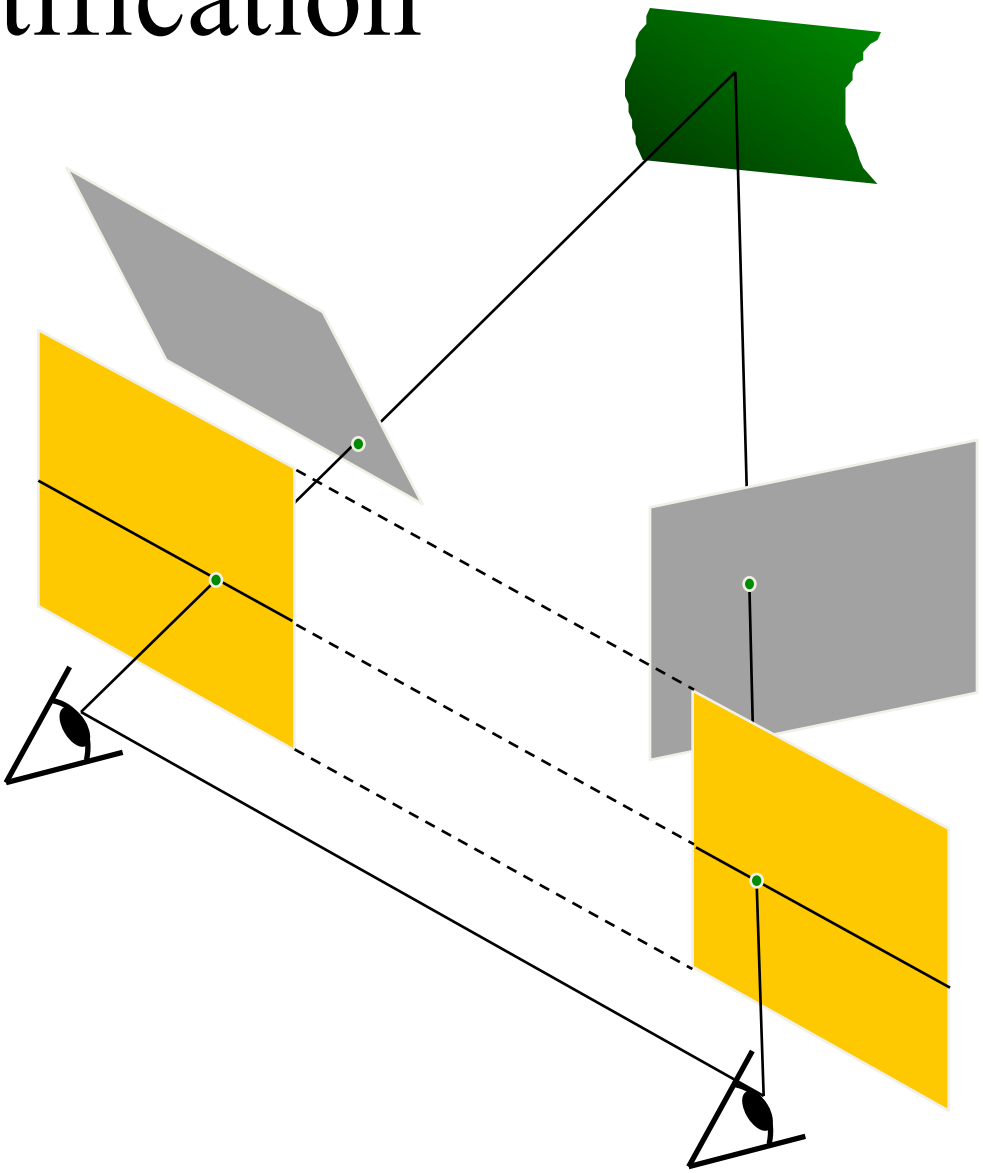
Disparity is inversely proportional to depth.

Stereo image rectification



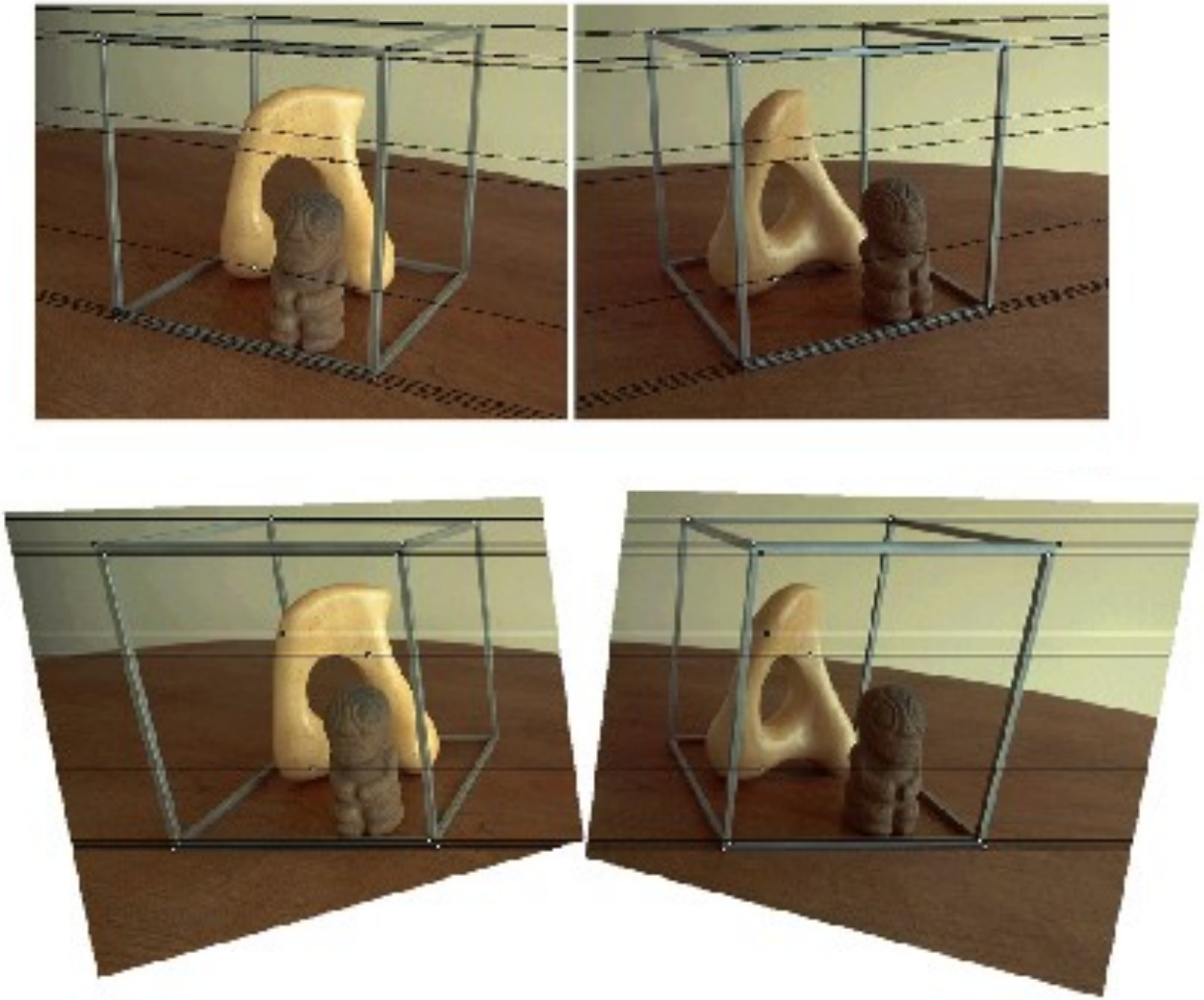
Stereo image rectification

- 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

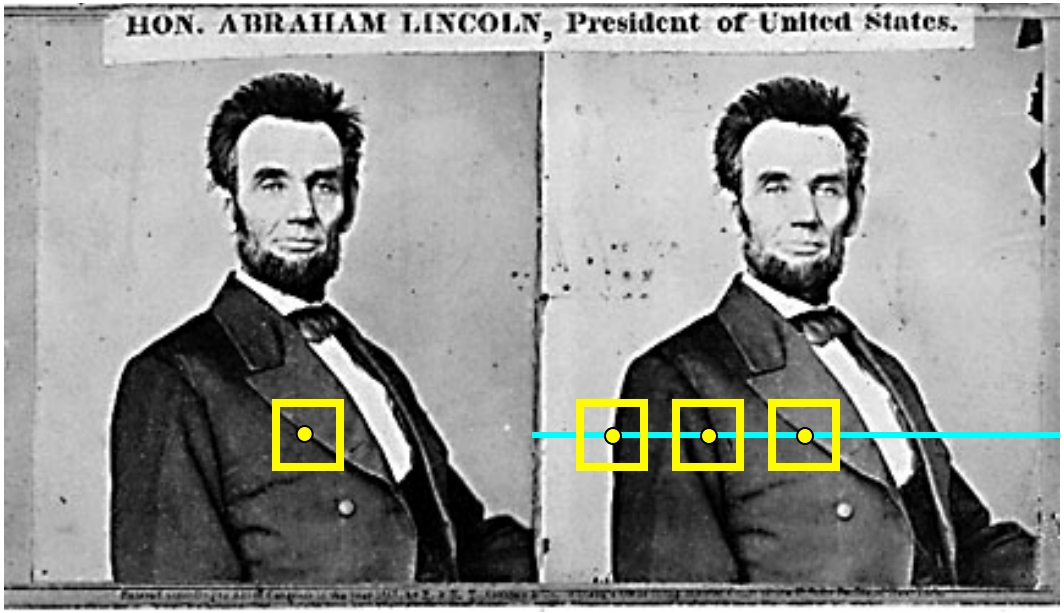


➢ C. Loop and Z. Zhang. Computing Rectifying Homographies for Stereo Vision. IEEE Conf. Computer Vision and Pattern Recognition, 1999.

Rectification example

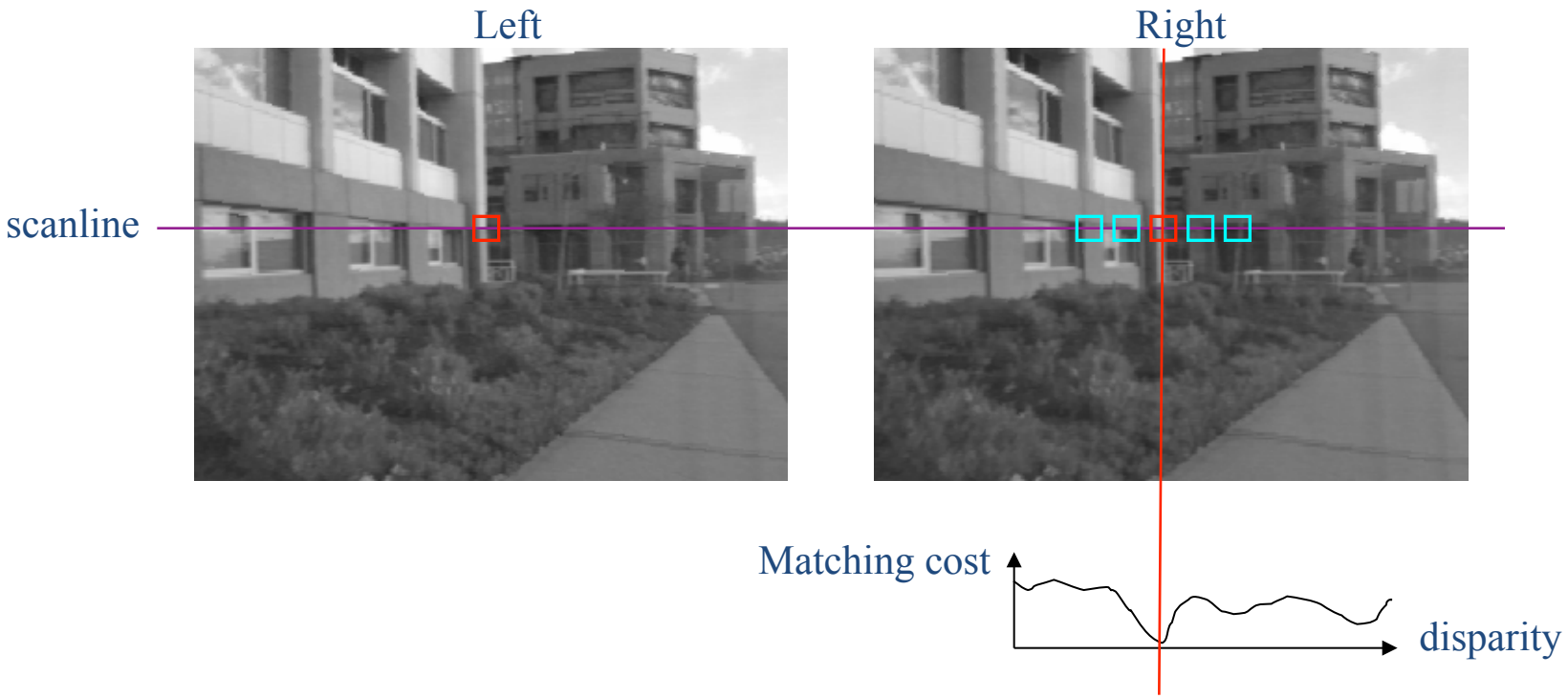


Basic stereo matching algorithm



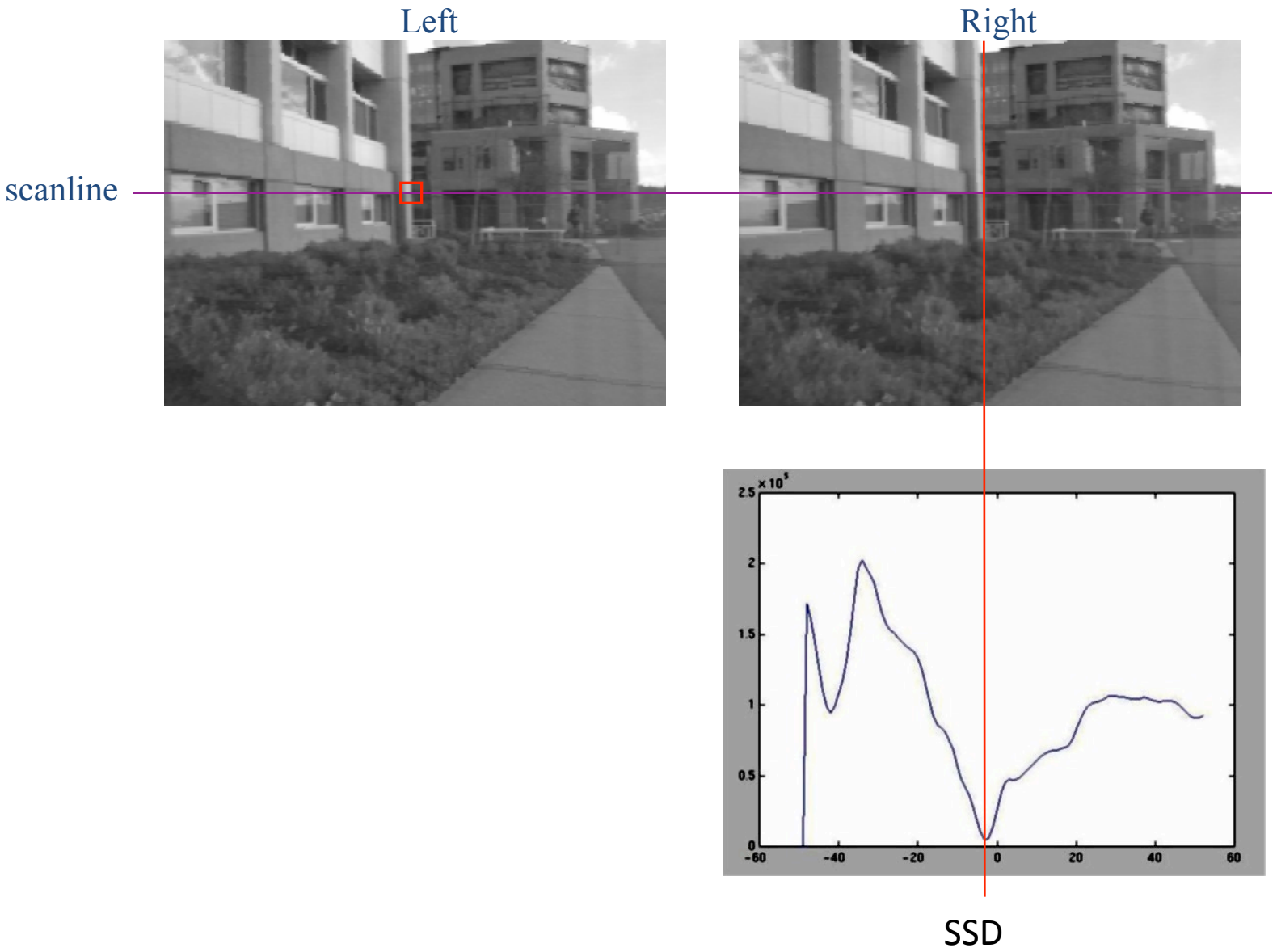
- If necessary, rectify the two stereo images to transform epipolar lines into scanlines
- For each pixel x in the first image
  - Find corresponding epipolar scanline in the right image
  - Search the scanline and pick the best match x'
  - Compute disparity x-x' and set depth(x) = fB/(x-x')

Correspondence search

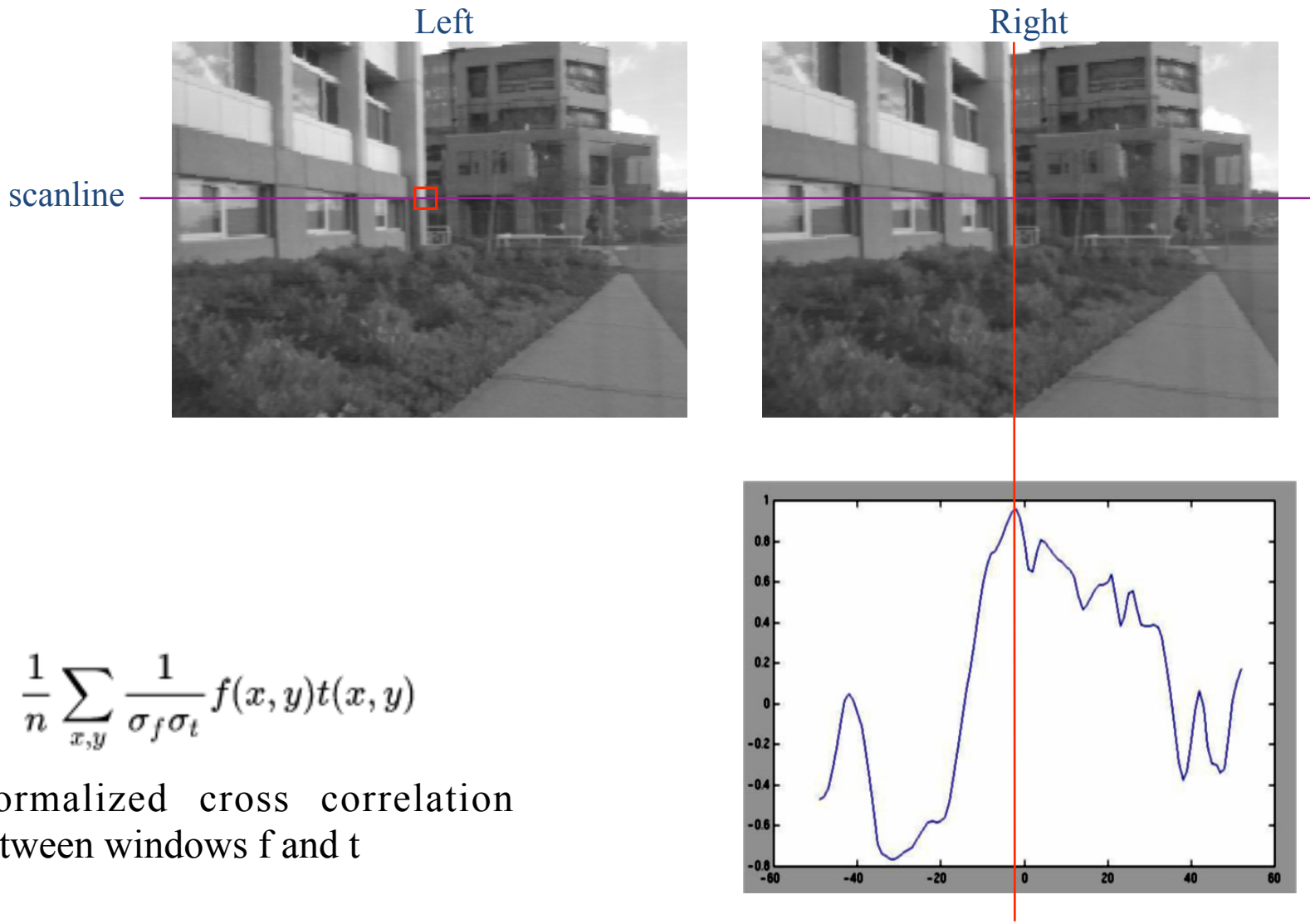


- Slide a window along the right scanline and compare contents of that window with the reference window in the left image
- Matching cost: SSD or normalized correlation

Correspondence search



Correspondence search

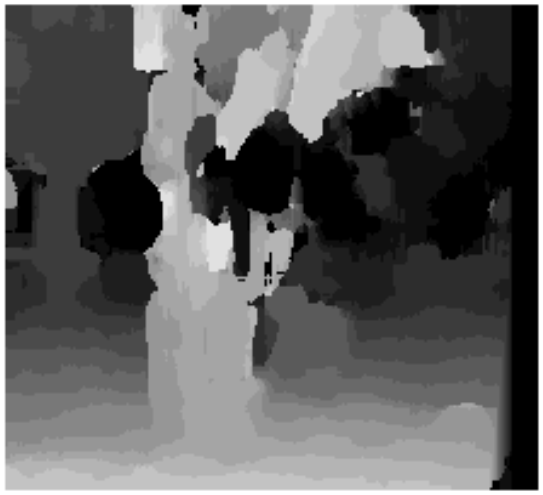


$$\frac{1}{n}\sum_{x,y}\frac{1}{\sigma_f\sigma_t}f(x,y)t(x,y)$$

Normalized cross correlation between windows f and t

Effect of window size

W = 3

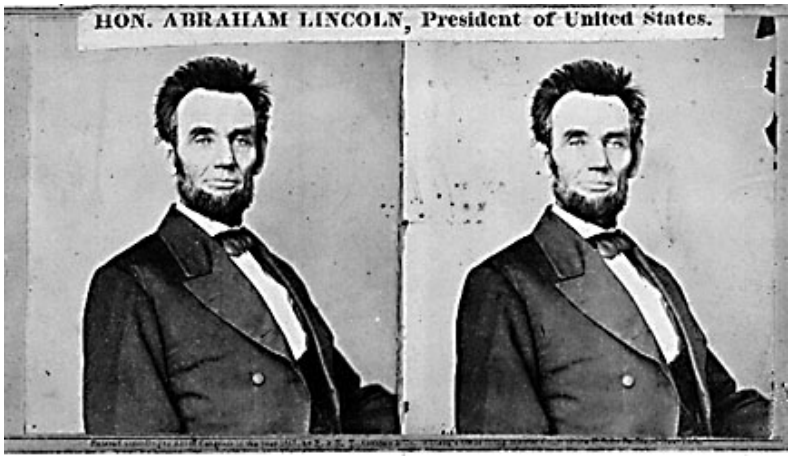


W = 20

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

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

Failures of correspondence search



Textureless surfaces

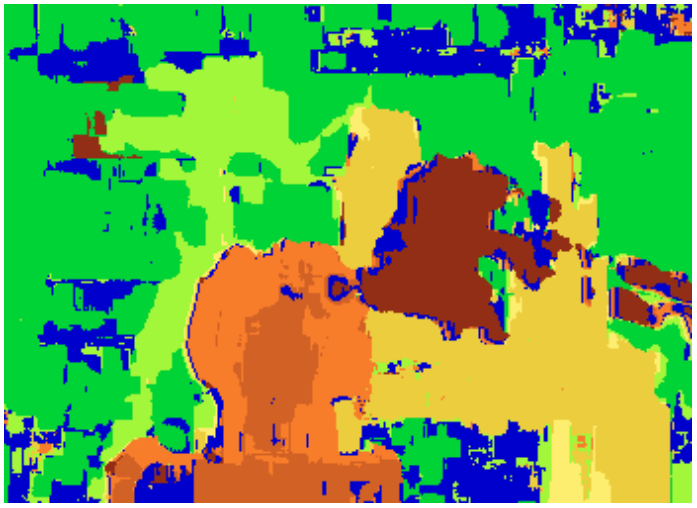
Occlusions, repetition

Non-Lambertian surfaces, specularities

Results with window search

Data

Window-based matching



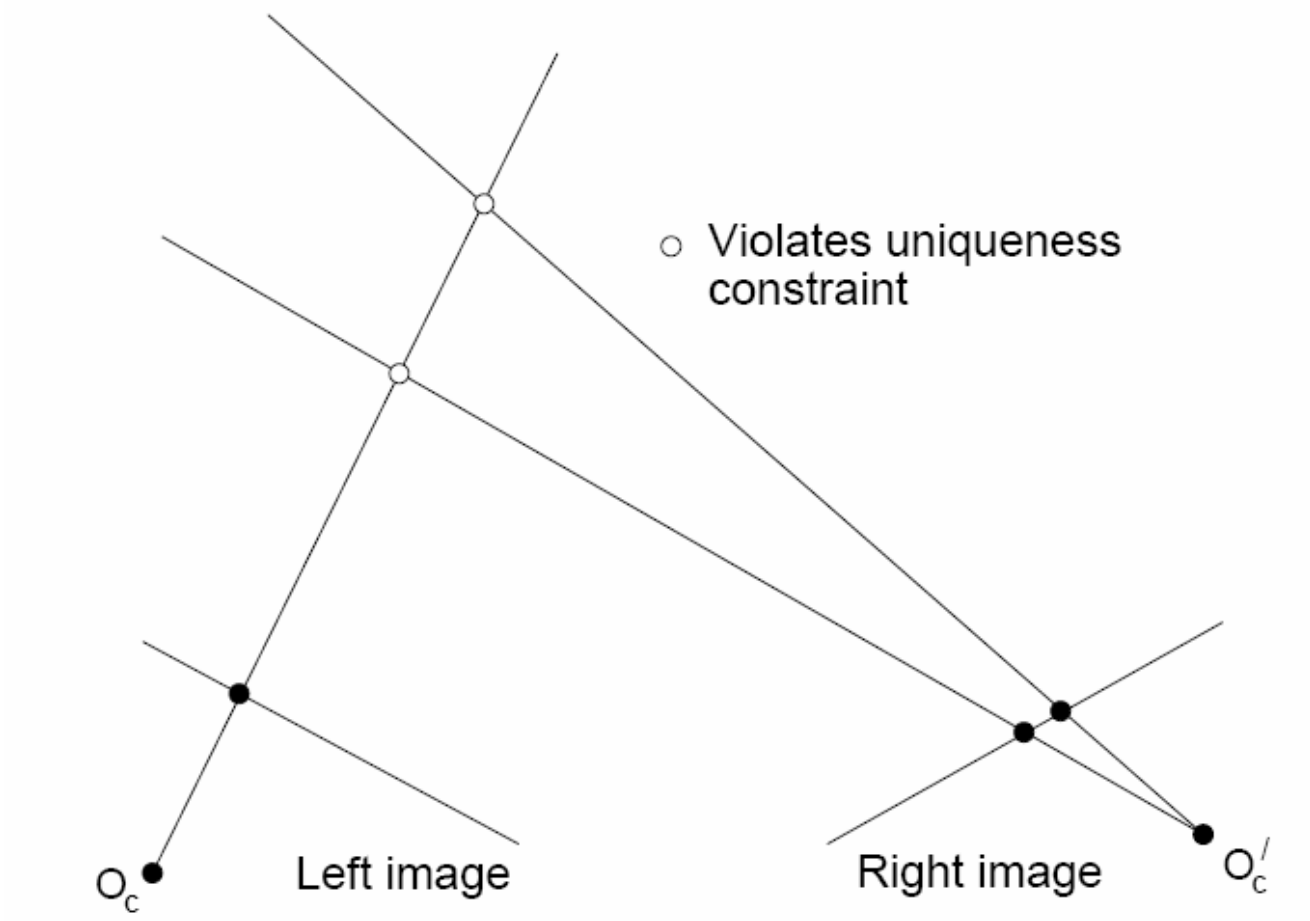
Ground truth

How can we improve window-based matching?

- So far, matches are independent for each point
- What constraints or priors can we add?

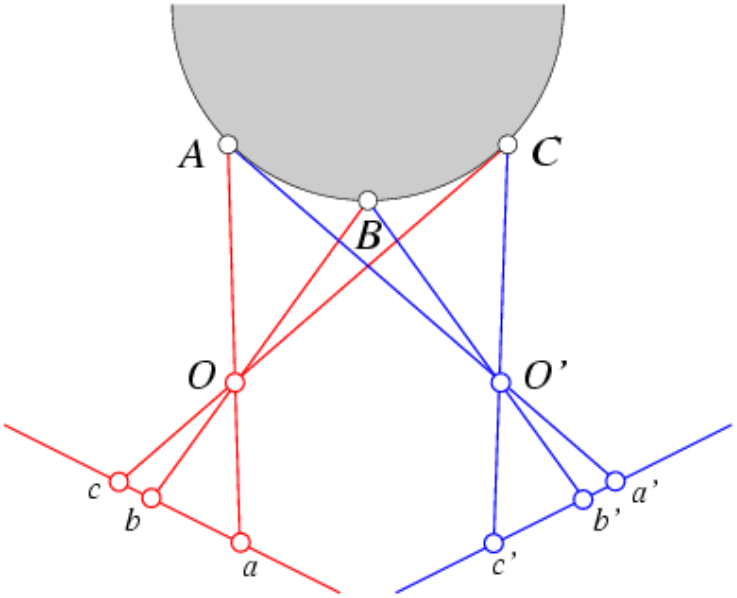
Stereo constraints/priors

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



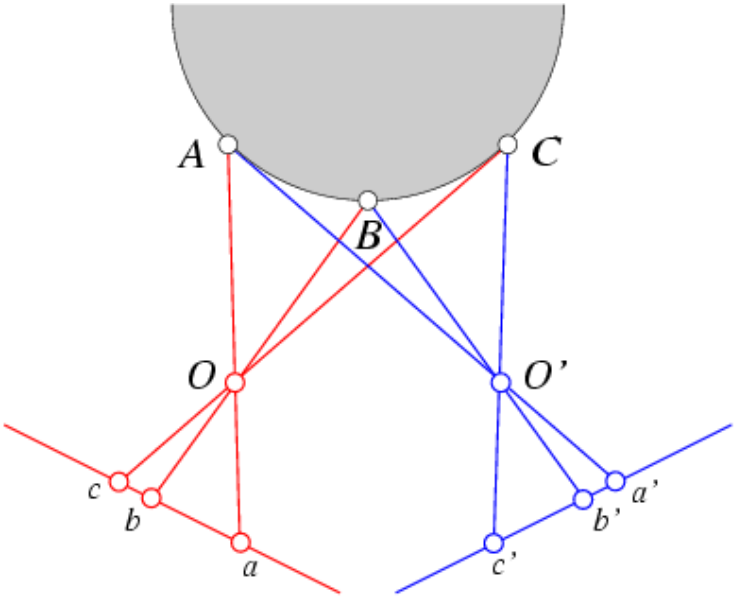
Stereo constraints/priors

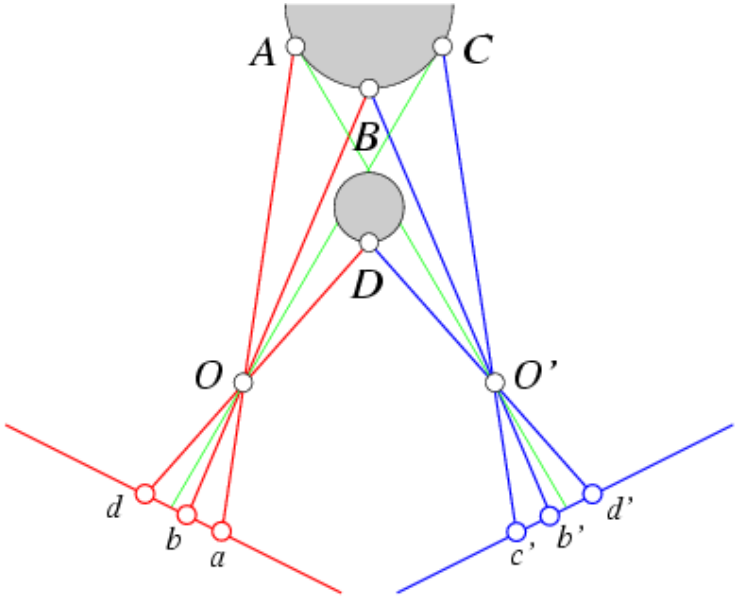
- Uniqueness
  - For any point in one image, there should be at most one matching point in the other image
- Ordering
  - Corresponding points should be in the same order in both views



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



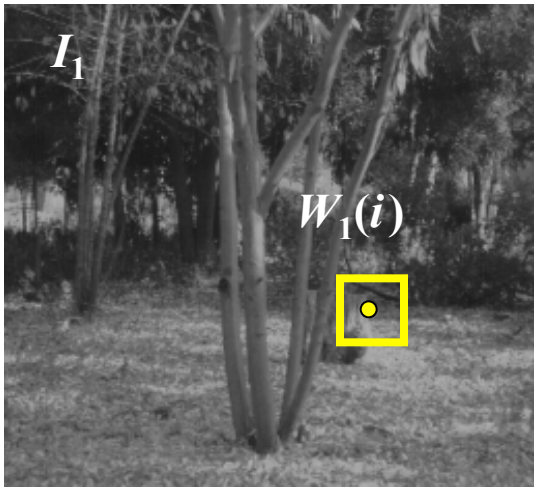


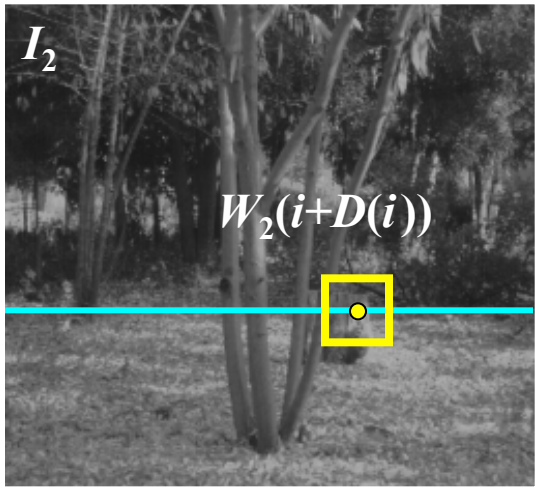
Ordering constraint doesn't hold

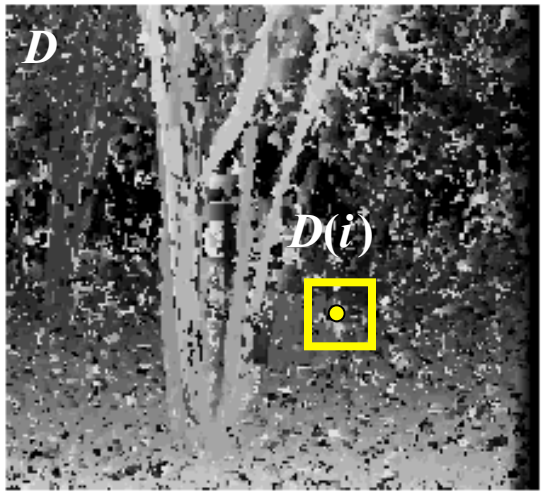
Priors and constraints

- 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 usually change slowly

Stereo matching as energy minimization







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

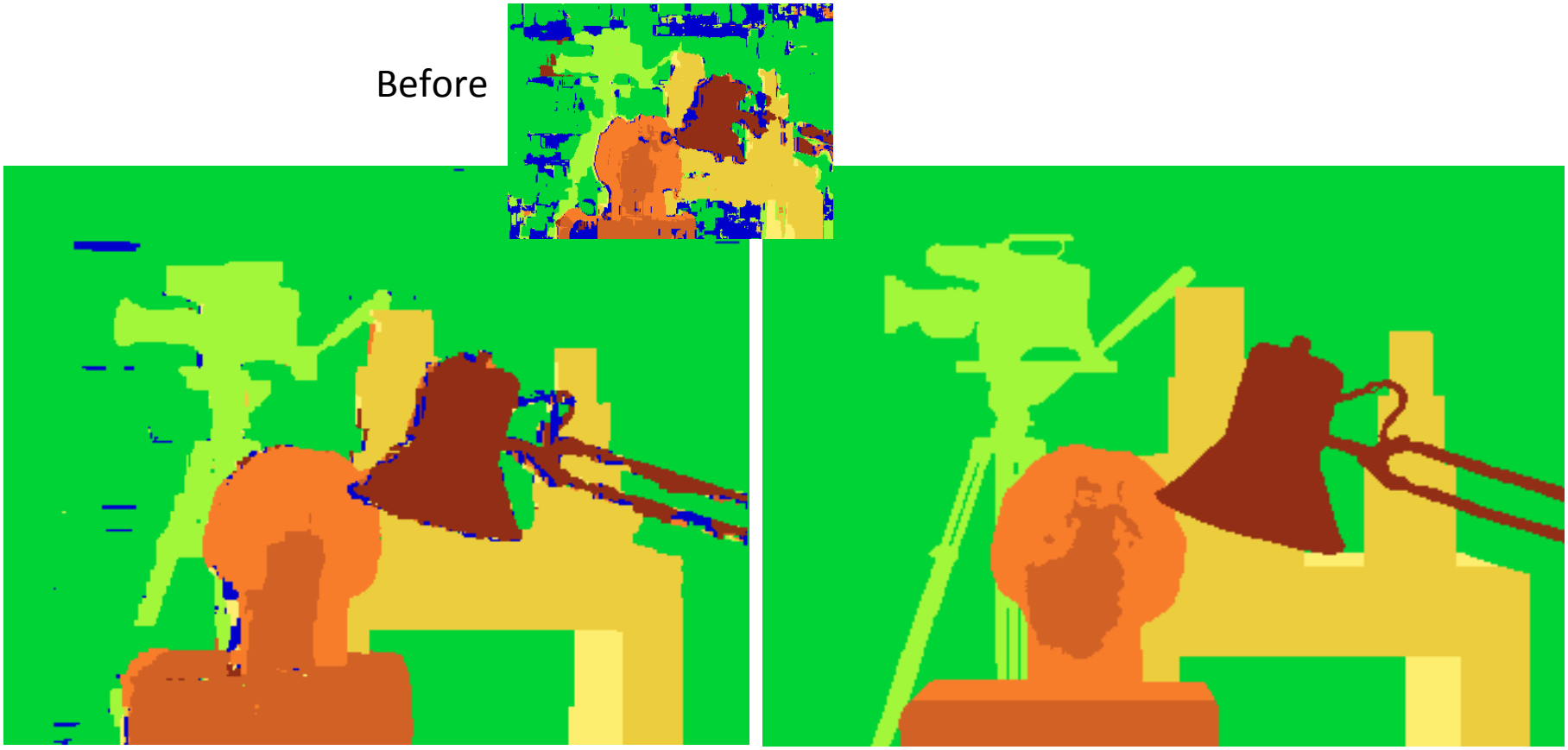
$$E_{\text{data}} = \sum_i \left(W_1(i) - W_2(i + D(i))\right)^2 \qquad E_{\text{smooth}} = \sum_{\text{neighbors } (i,j)} \left\|D(i) - D(j)\right\|^2$$

- 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