Stereo

CSE 576

Ali Farhadi

Several slides from Larry Zitnick and Steve Seitz


WHY DO WE HAVE
TWO EYES?

Why do we perceive depth?

What do humans use as depth cues?

Motion

Convergence
When watching an object close to us, our eyes point slightly inward. This difference in the direction of the eyes is called convergence. This depth cue is effective only on short distances (less than 10 meters).

Binocular Parallax
As our eyes see the world from slightly different locations, the images sensed by the eyes are slightly different. This difference in the sensed images is called binocular parallax. Human visual system is very sensitive to these differences, and binocular parallax is the most important depth cue for medium viewing distances. The sense of depth can be achieved using binocular parallax even if all other depth cues are removed.

Monocular Movement Parallax
If we close one of our eyes, we can perceive depth by moving our head. This happens because human visual system can extract depth information in two similar images sensed after each other, in the same way it can combine two images from different eyes.

Focus

Accommodation
Accommodation is the tension of the muscle that changes the focal length of the lens of eye. Thus it brings into focus objects at different distances. This depth cue is quite weak, and it is effective only at short viewing distances (less than 2 meters) and with other cues.

What do humans use as depth cues?

Image cues

Retinal Image Size
When the real size of the object is known, our brain compares the sensed size of the object to this real size, and thus acquires information about the distance of the object.

Linear Perspective
When looking down a straight level road we see the parallel sides of the road meet in the horizon. This effect is often visible in photos and it is an important depth cue. It is called linear perspective.

Texture Gradient
The closer we are to an object the more detail we can see of its surface texture. So objects with smooth textures are usually interpreted being farther away. This is especially true if the surface texture spans all the distance from near to far.

Overlapping
When objects block each other out of our sight, we know that the object that blocks the other one is closer to us. The object whose outline pattern looks more continuous is felt to lie closer.

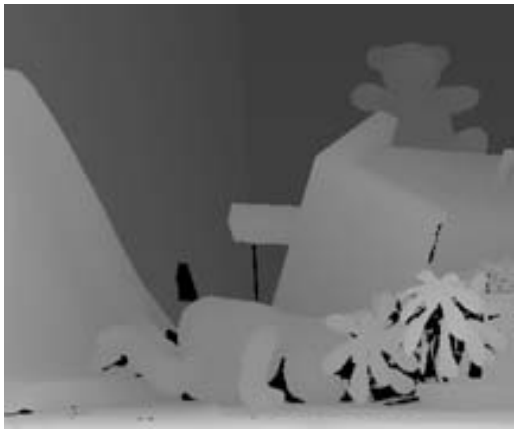
Aerial Haze
The mountains in the horizon look always slightly bluish or hazy. The reason for this are small water and dust particles in the air between the eye and the mountains. The farther the mountains, the hazier they look.

Shades and Shadows
When we know the location of a light source and see objects casting shadows on other objects, we learn that the object shadowing the other is closer to the light source. As most illumination comes downward we tend to resolve ambiguities using this information. The three dimensional looking computer user interfaces are a nice example on this. Also, bright objects seem to be closer to the observer than dark ones.

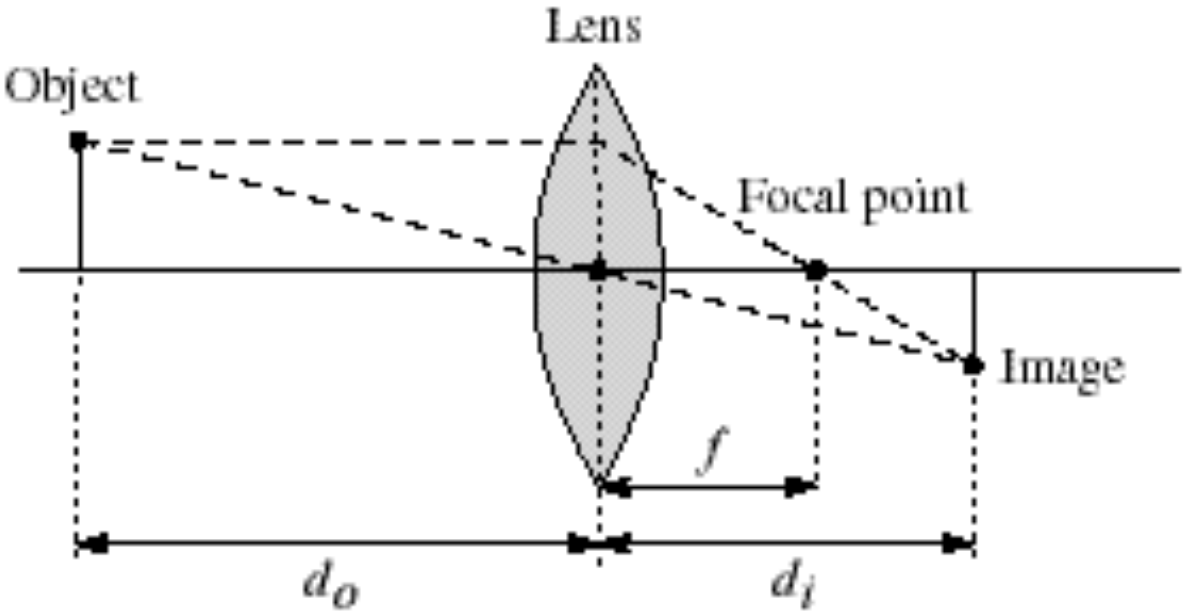
Marko Teittinen http://www.hitl.washington.edu/scivw/EVE/III.A.1.c.DepthCues.html

Amount of horizontal movement is …

…inversely proportional to the distance from the camera



Cameras

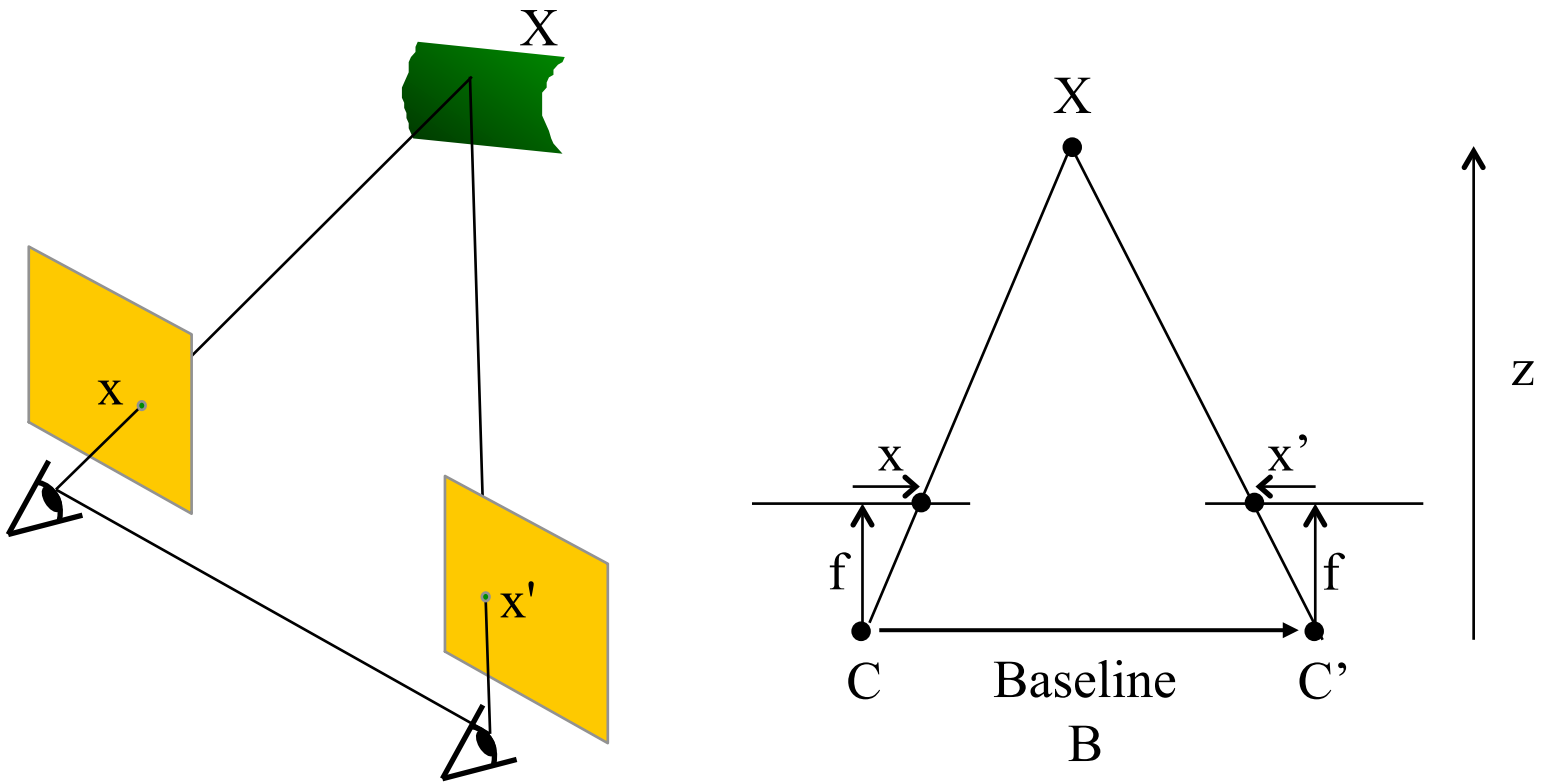


Thin lens equation: $$\frac{1}{d_o} + \frac{1}{d_i} = \frac{1}{f}$$

- Any object point satisfying this equation is in focus

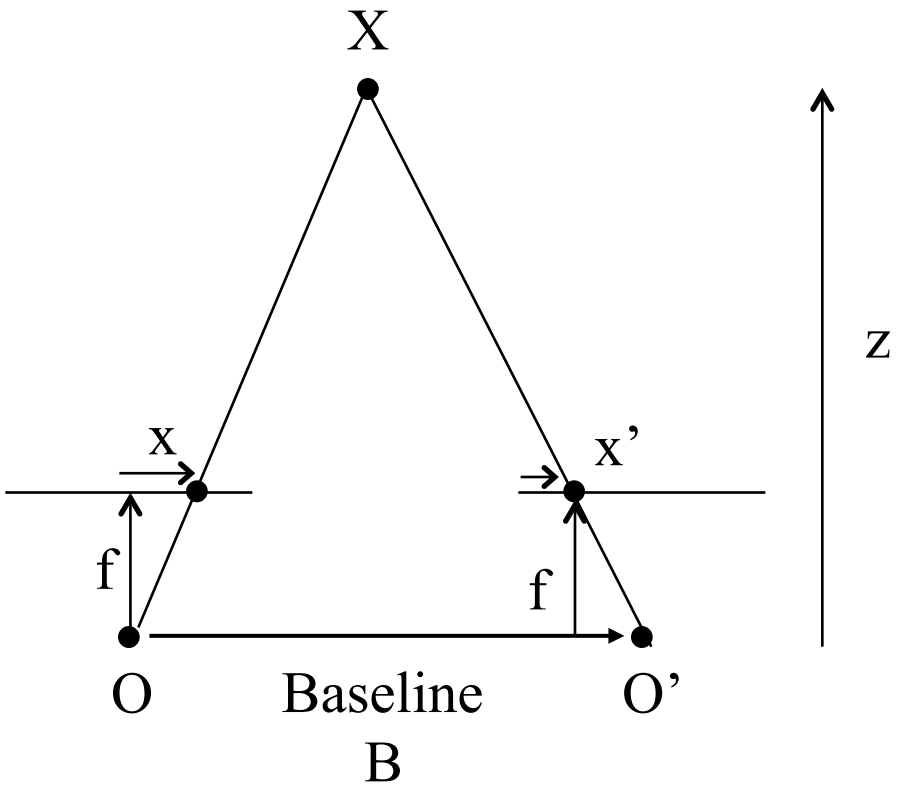
Depth from Stereo

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



Depth from disparity

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



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

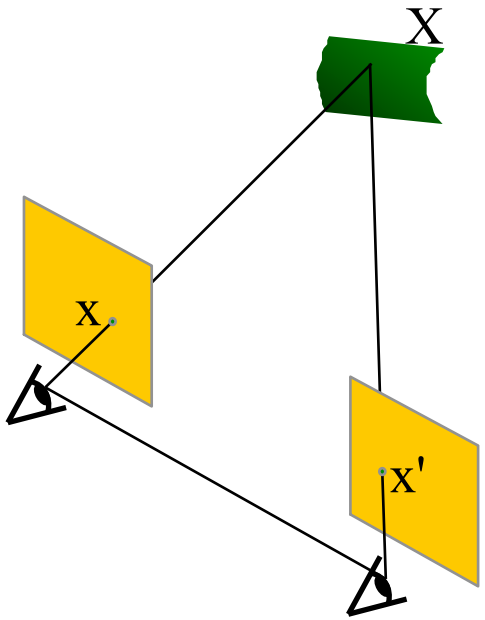
Disparity is inversely proportional to depth.

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’?

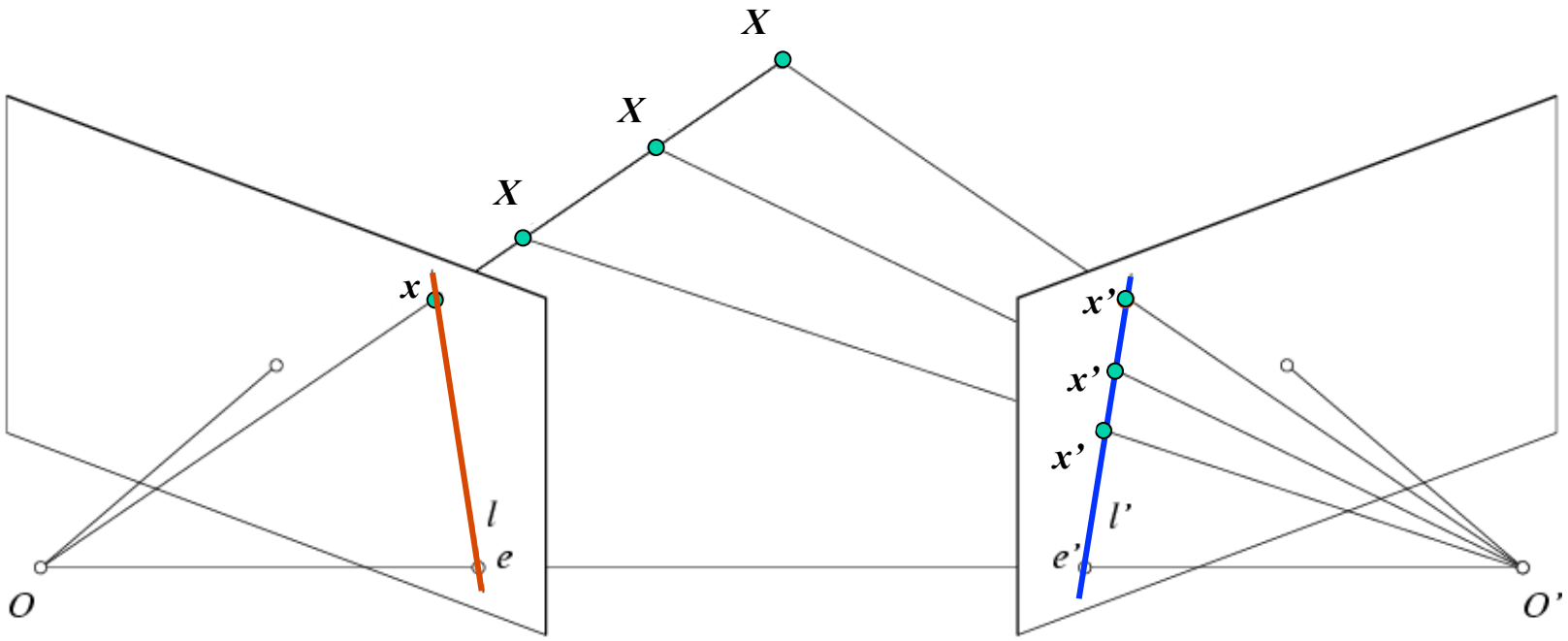


Correspondence Problem

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?

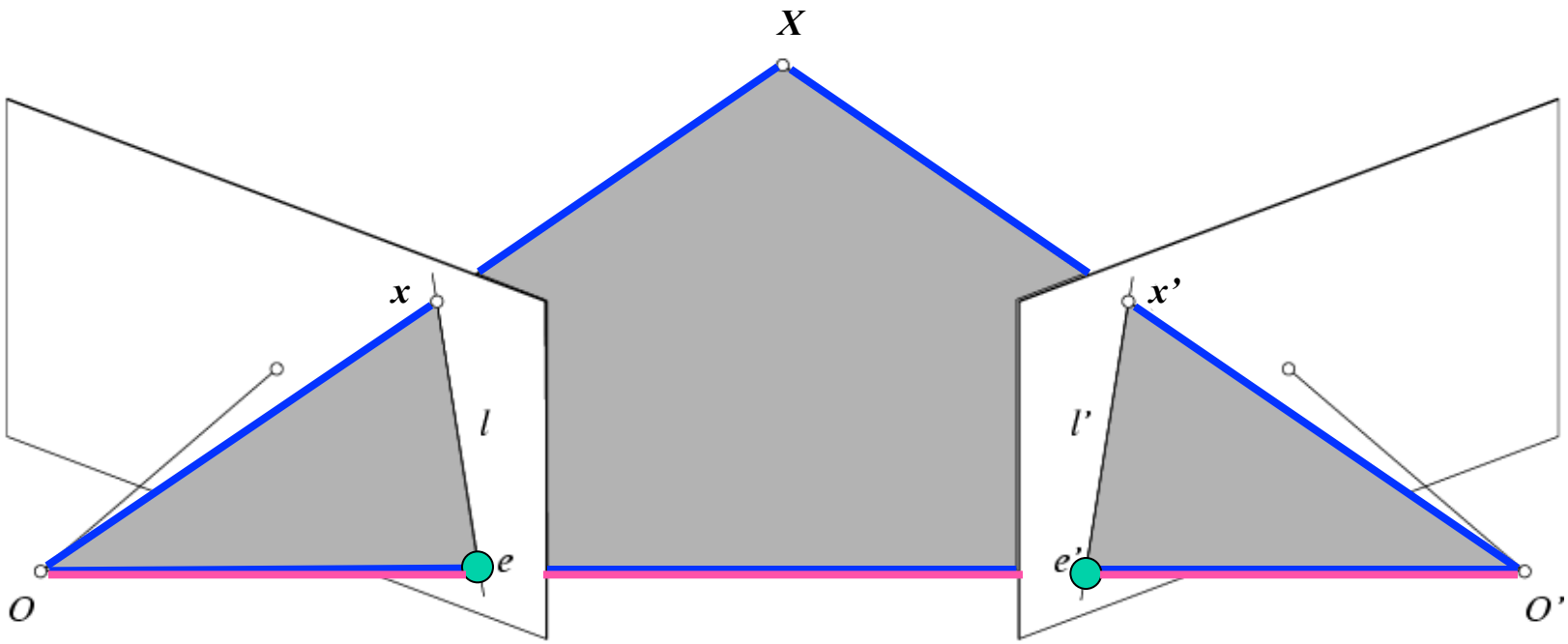
Key idea: Epipolar constraint



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



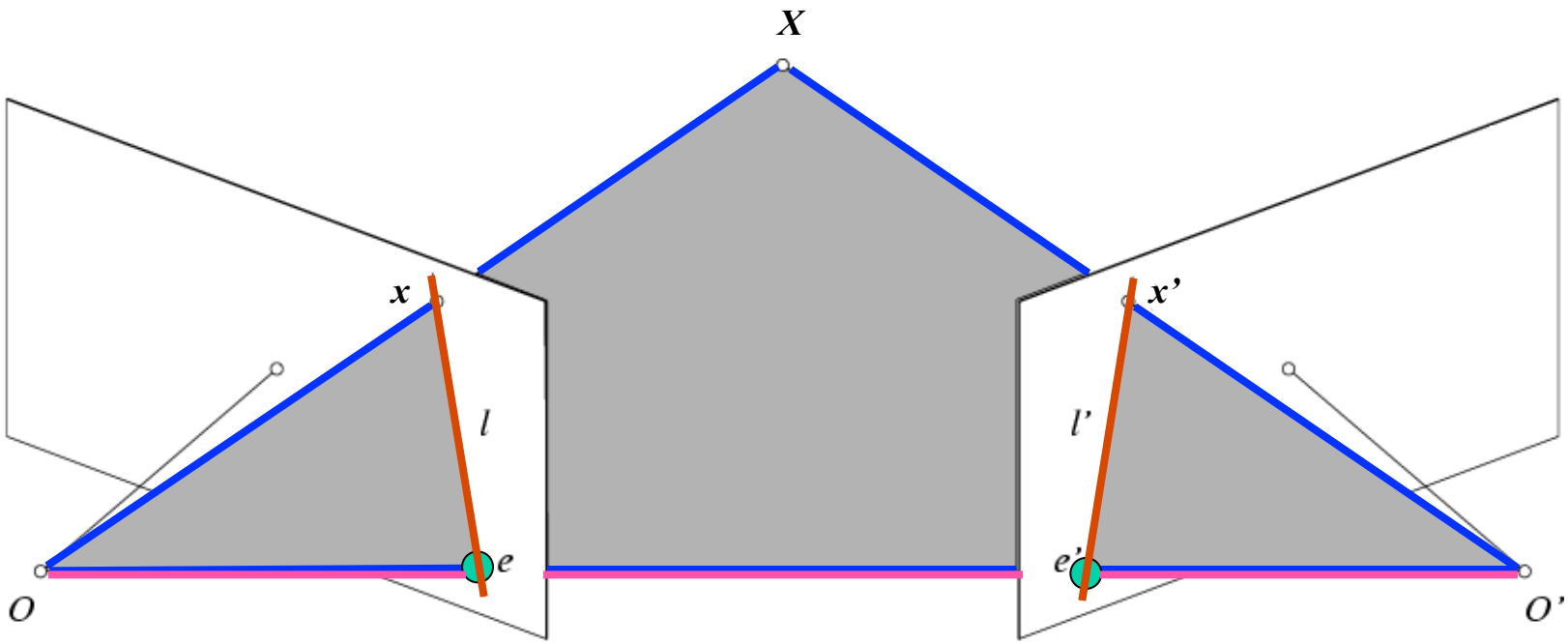
- **Baseline** – line connecting the two camera centers
- **Epipoles**

= intersections of baseline with image planes

= projections of the other camera center

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

Epipolar geometry: notation



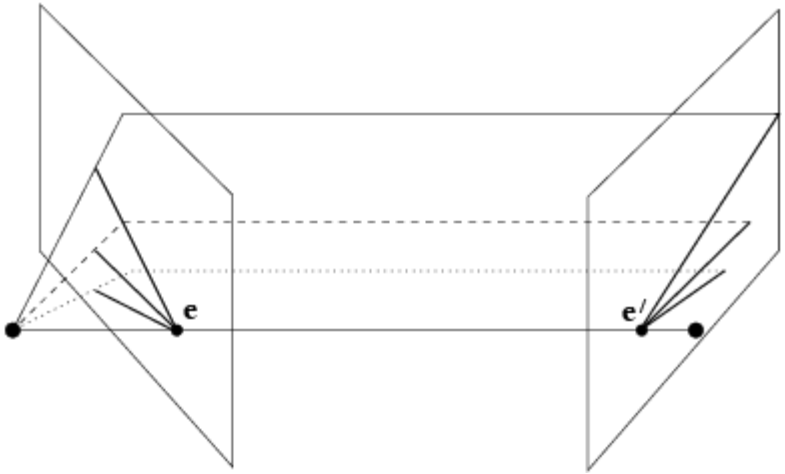
- **Baseline** – line connecting the two camera centers
- **Epipoles**

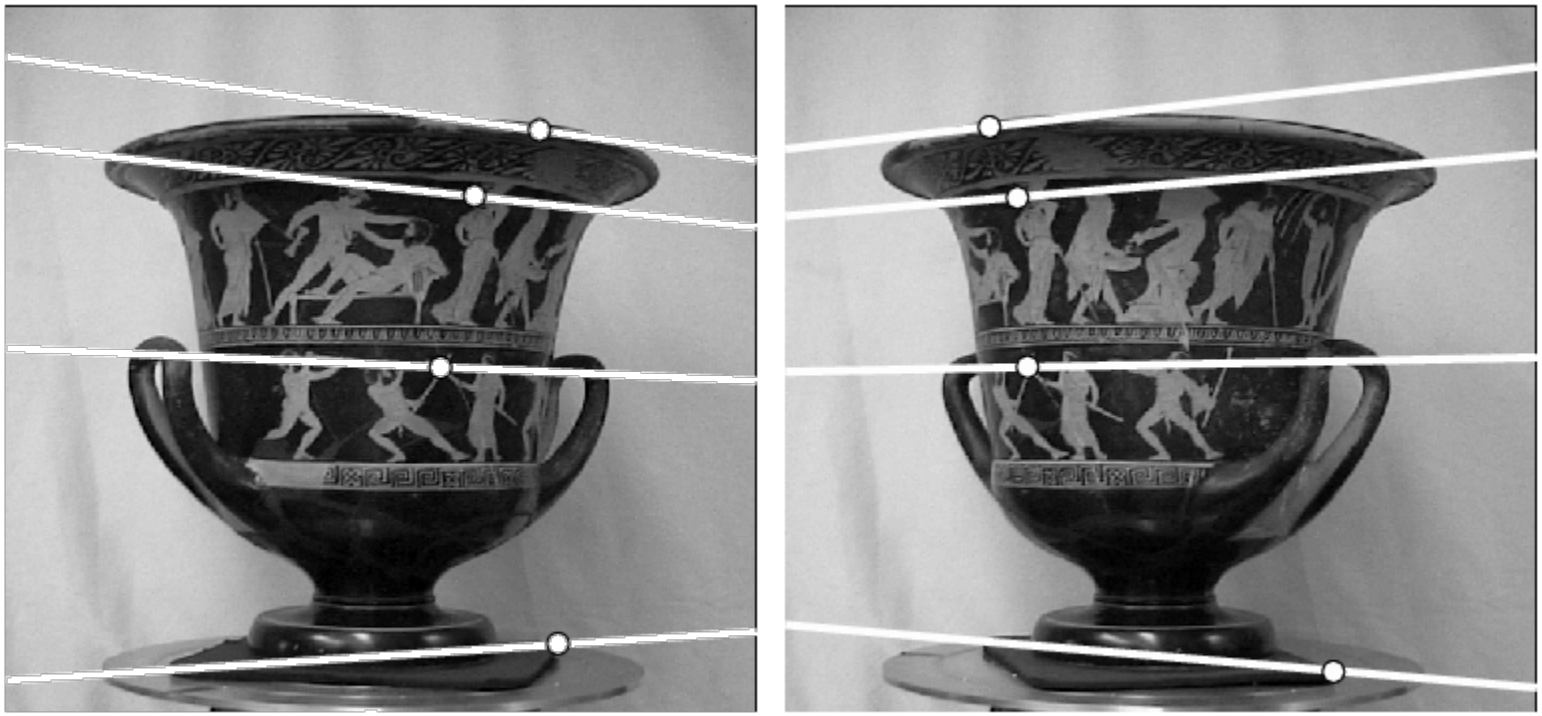
= intersections of baseline with image planes

= projections of the other camera center

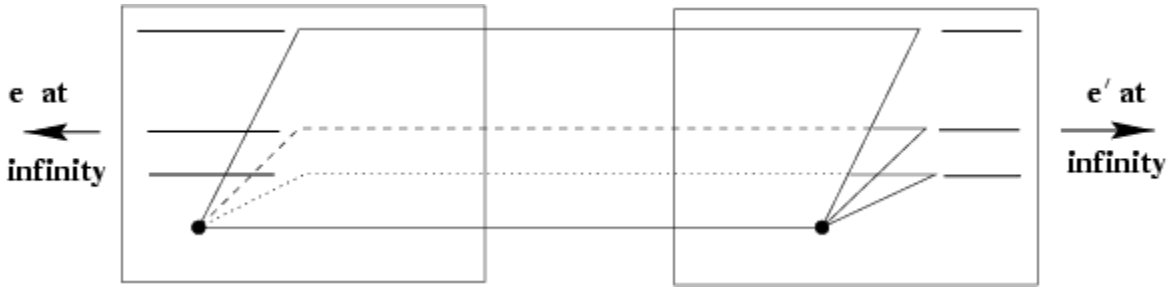
- **Epipolar Plane** – plane containing baseline (1D family)
- **Epipolar Lines** - intersections of epipolar plane with image planes (always come in corresponding pairs)

Example: Converging cameras





Example: Motion parallel to image plane



Example: Forward motion

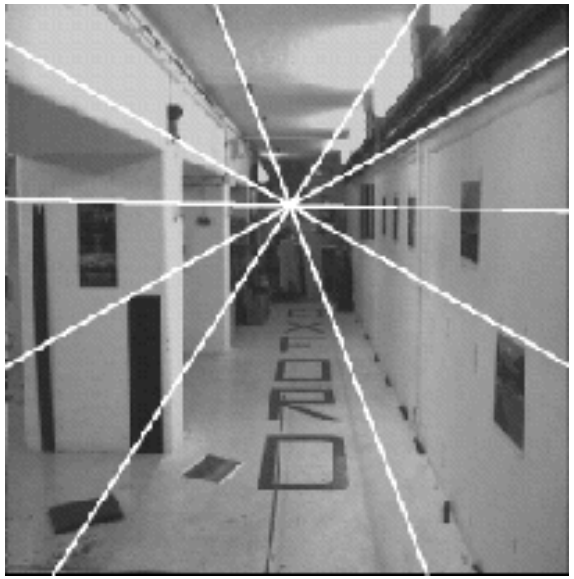
What would the epipolar lines look like if the camera moves directly forward?

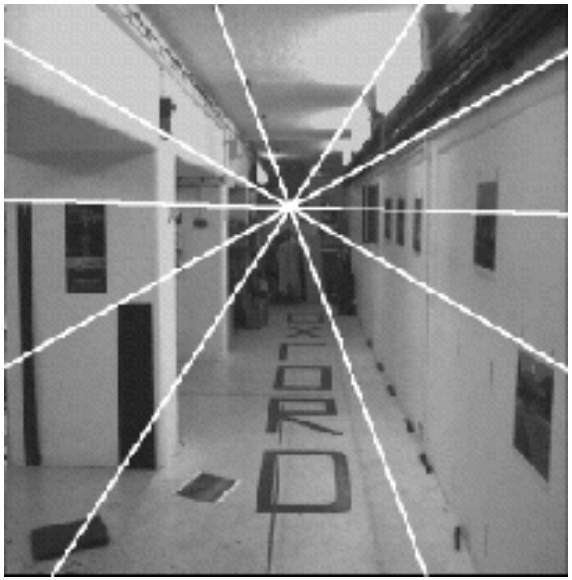
Example: Motion perpendicular to image plane

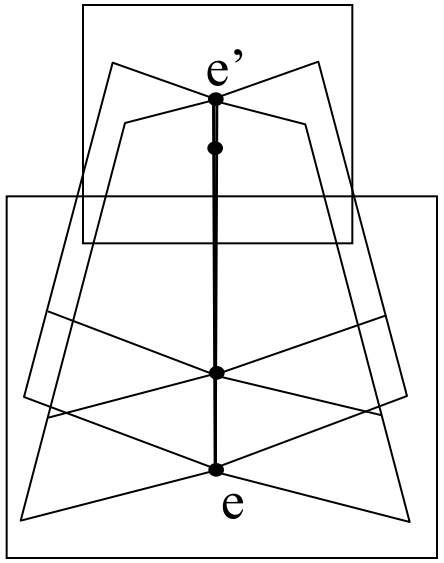
Example: Motion perpendicular to image plane

- Points move along lines radiating from the epipole: “focus of expansion”
- Epipole is the principal point

Example: Forward motion



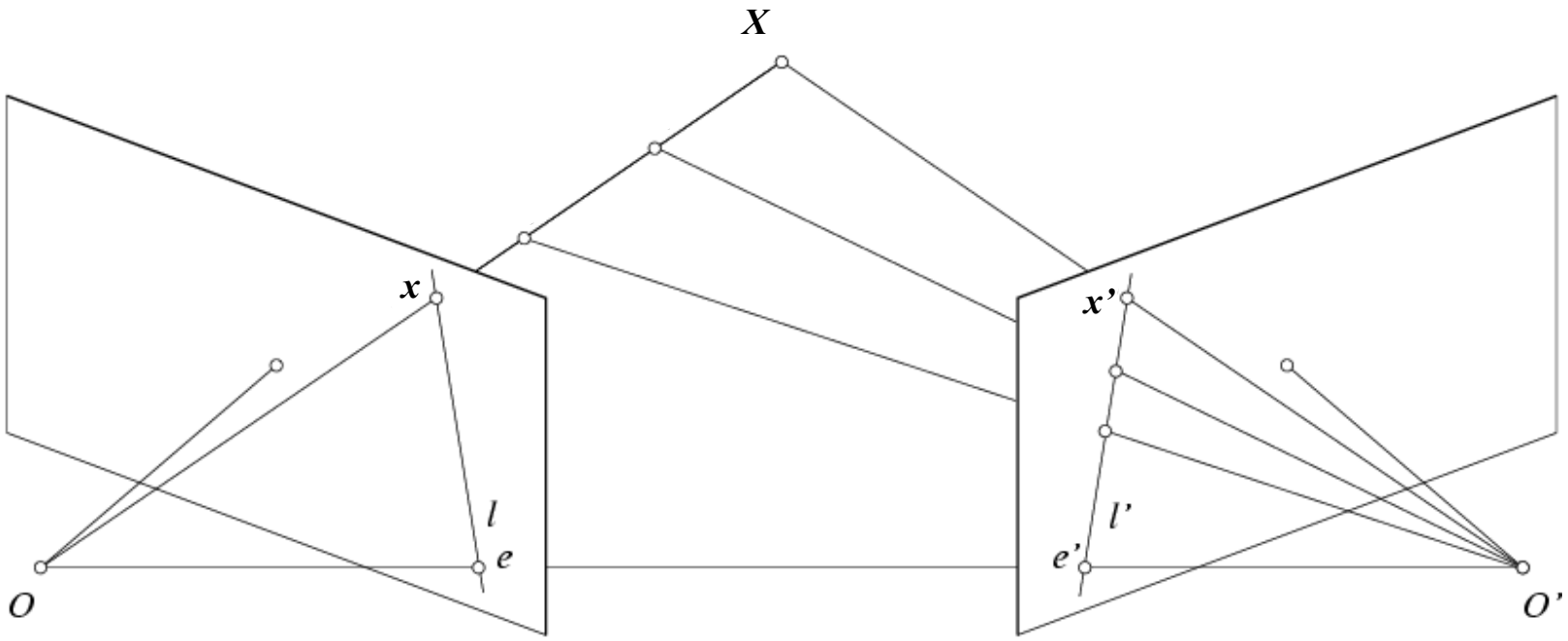




Epipole has same coordinates in both images.

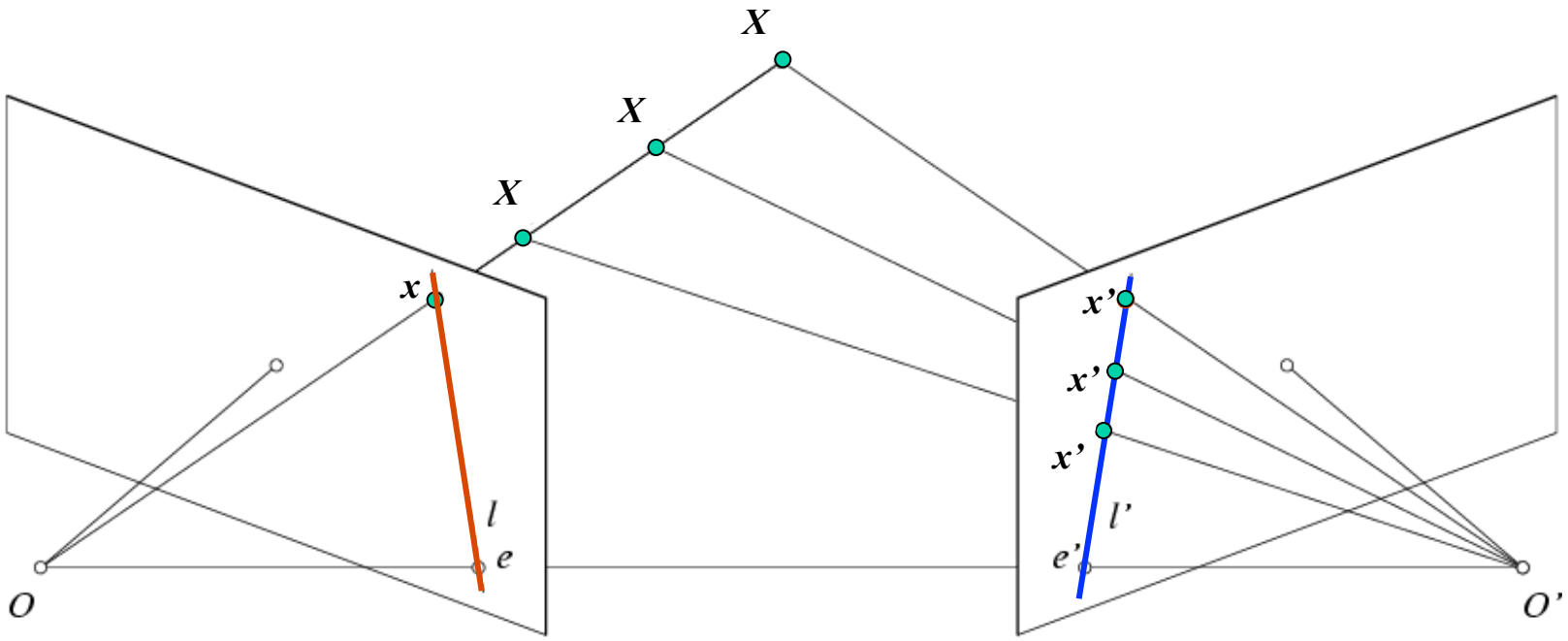
Points move along lines radiating from “Focus of expansion”

Epipolar constraint



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

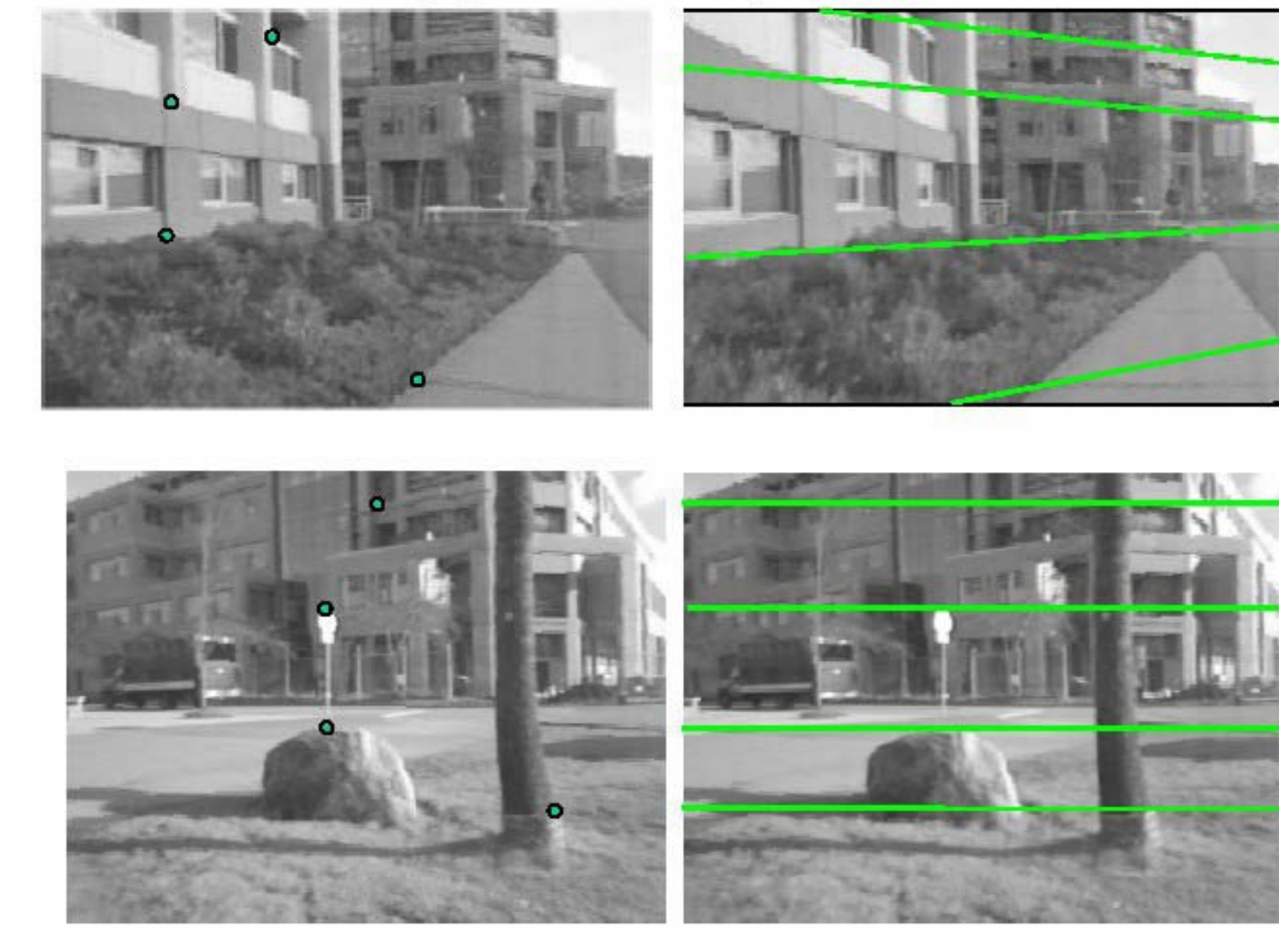
Epipolar constraint



- 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 example

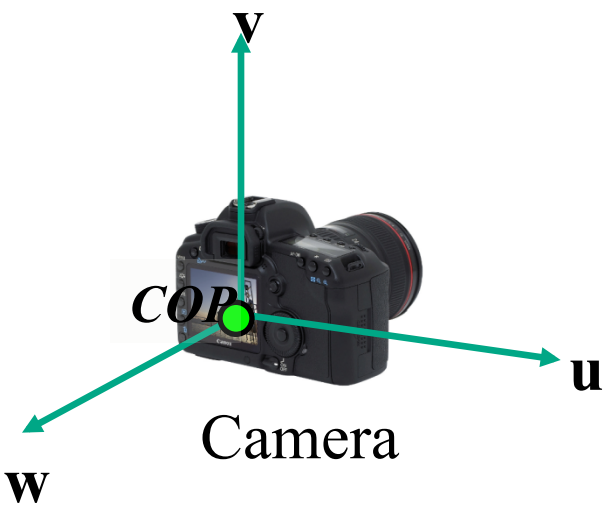


Camera parameters

How many numbers do we need to describe a camera?

- We need to describe its *pose* in the world
- We need to describe its internal *parameters*

A Tale of Two Coordinate Systems



Camera

Two important coordinate systems:

1. *World* coordinate system
2. *Camera* coordinate system

"The World"

Camera parameters

- To project a point (x,y,z) in *world* coordinates into a camera
- First transform (x,y,z) into *camera* coordinates
- Need to know
  - Camera position (in world coordinates)
  - Camera orientation (in world coordinates)
- Then project into the image plane
  - Need to know camera *intrinsics*
- These can all be described with matrices

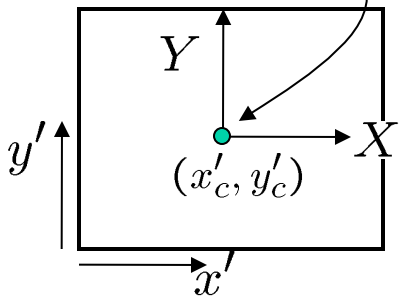
Camera parameters

A camera is described by several parameters

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

Projection equation

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



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

identity matrix

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

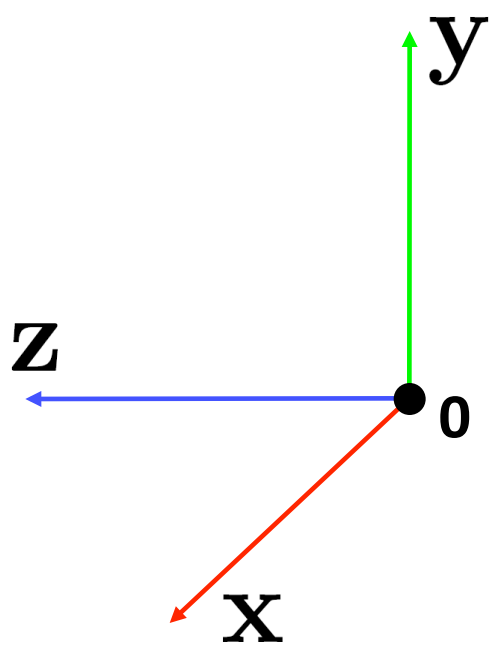
intrinsics projection rotation translation

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

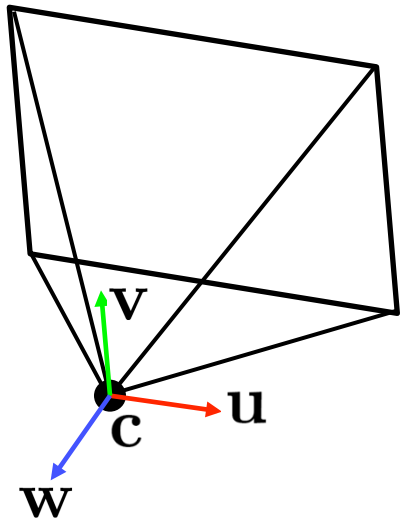
Extrinsics

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

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



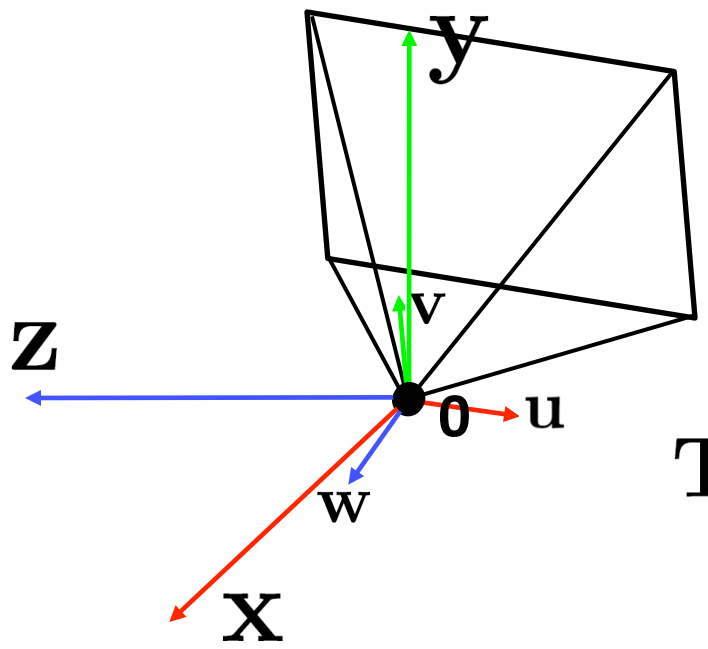
Step 1: Translate by -**c**



Extrinsics

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

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



Step 1: Translate by -**c**

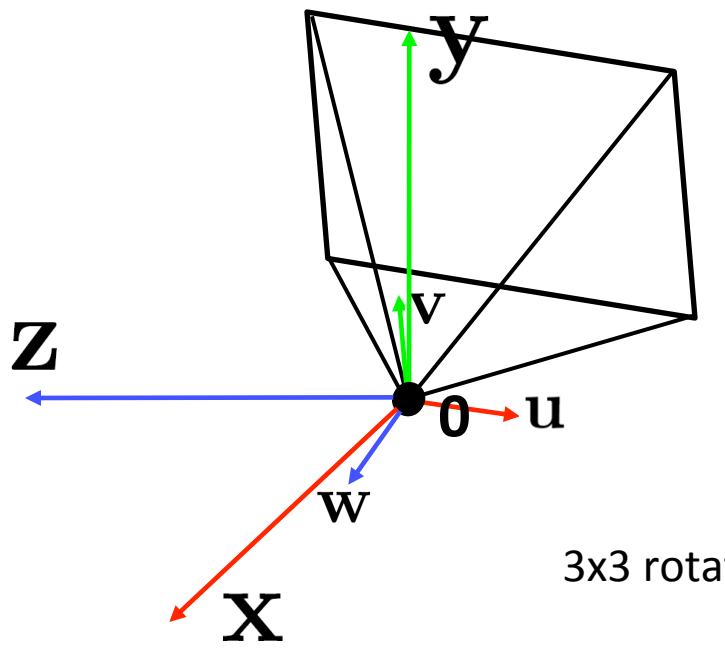
How do we represent translation as a matrix multiplication?

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

Extrinsics

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

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



Step 1: Translate by -**c**

Step 2: Rotate by **R**

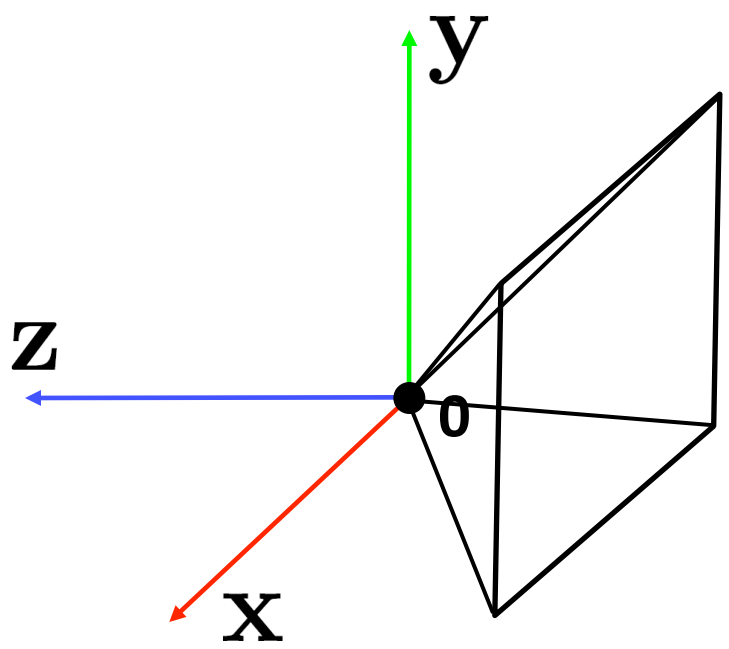
$$\mathbf{R} = \begin{bmatrix} \mathbf{u}^T \\ \mathbf{v}^T \\ \mathbf{w}^T \end{bmatrix}$$

3x3 rotation matrix

Extrinsics

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

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



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

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

Perspective projection

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

(intrinsics)

(converts from 3D rays in camera coordinate system to pixel coordinates)

in general, $\mathbf{K} = \begin{bmatrix} -f & s & c_x \\ 0 & -\alpha f & c_y \\ 0 & 0 & 1 \end{bmatrix}$ (upper triangular matrix)

α : **aspect ratio** (1 unless pixels are not square)

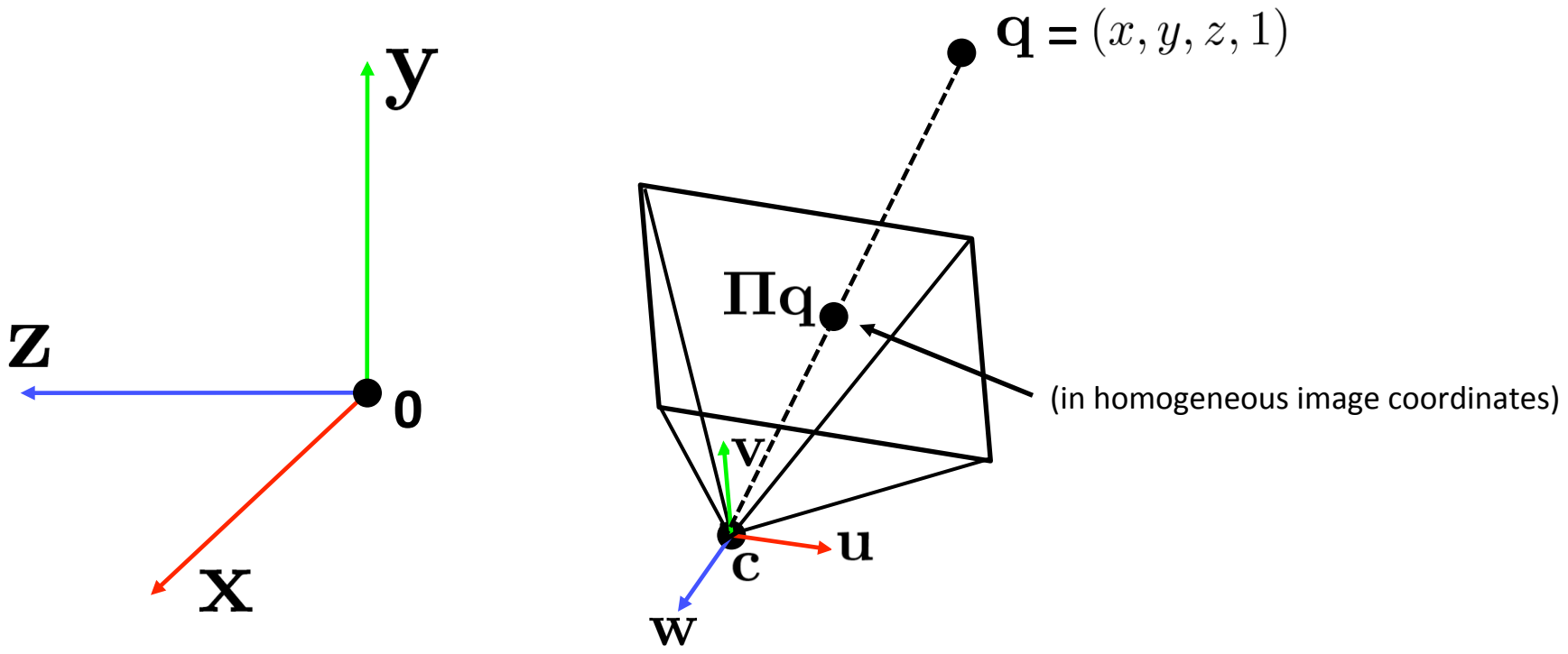
s : **skew** (0 unless pixels are shaped like rhombi/parallelograms)

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

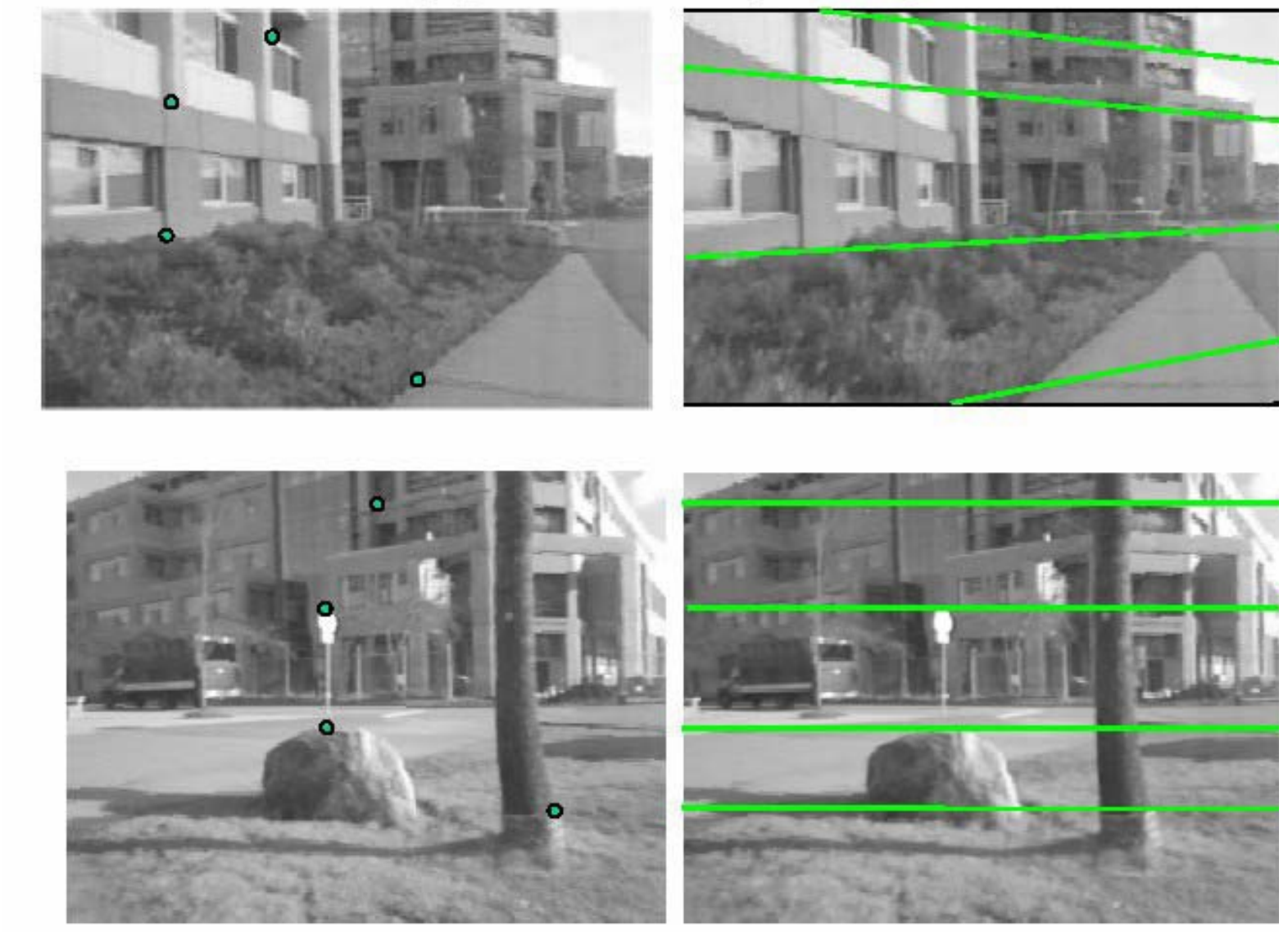
Projection matrix

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

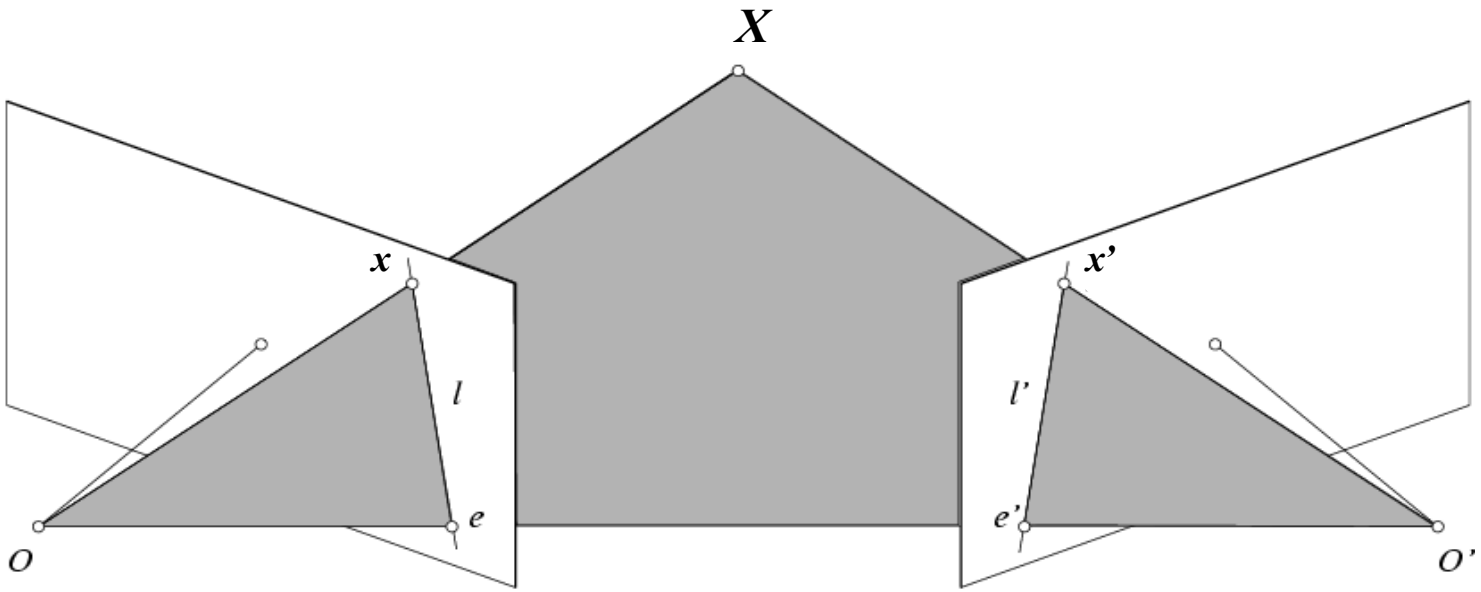
Projection matrix



Epipolar constraint example

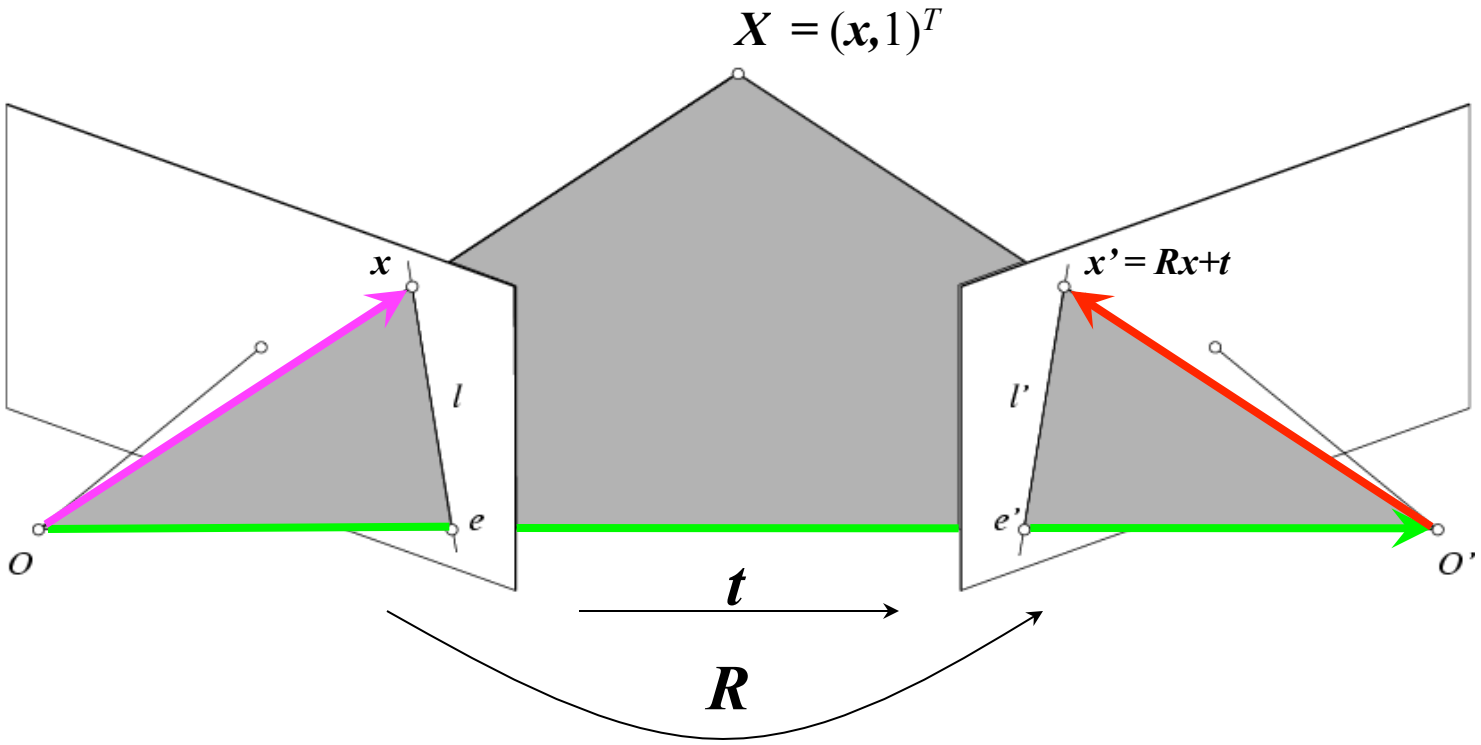


Epipolar constraint: Calibrated case



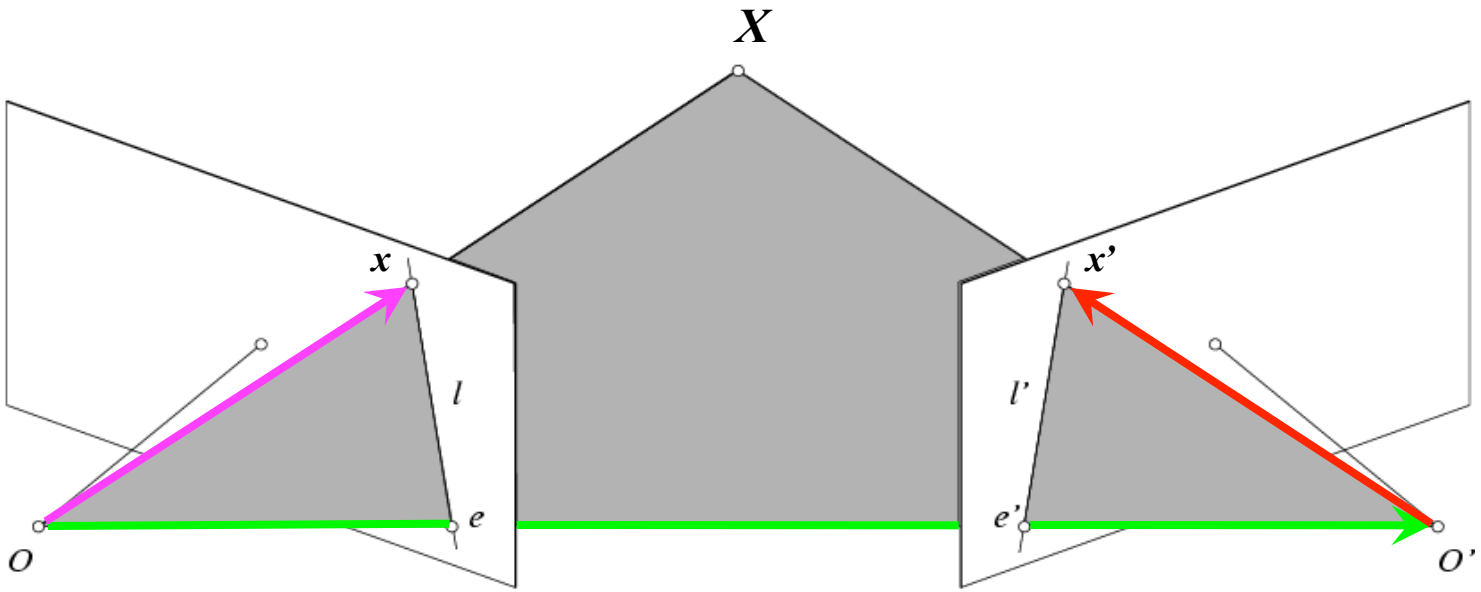
- Assume that the intrinsic and extrinsic parameters of the cameras are known
- We can multiply the projection matrix of each camera (and the image points) by the inverse of the calibration matrix to get *normalized* image coordinates
- We can also set the global coordinate system to the coordinate system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

Epipolar constraint: Calibrated case



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

Epipolar constraint: Calibrated case

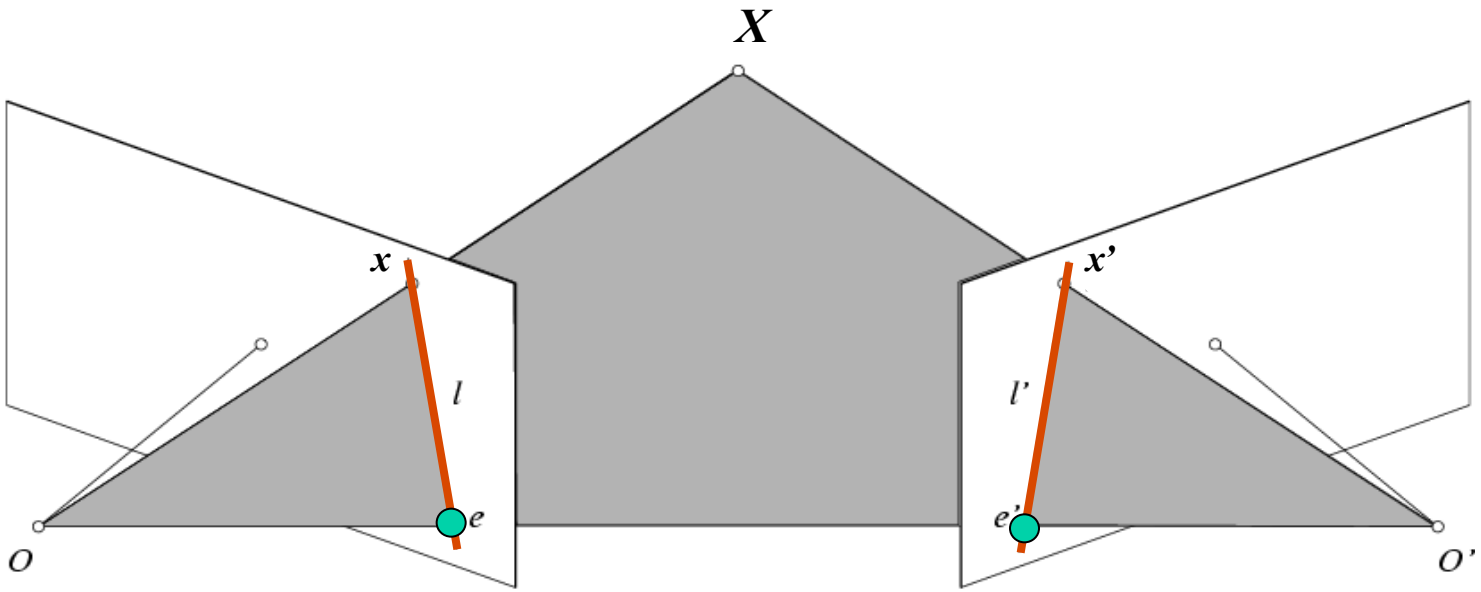


$$\boldsymbol{x}' \cdot [\boldsymbol{t} \times (\boldsymbol{R}\boldsymbol{x})] = 0 \quad \Rightarrow \quad \boldsymbol{x}'^{T} \boldsymbol{E}\, \boldsymbol{x} = 0 \quad \text{with} \quad \boldsymbol{E} = [\boldsymbol{t}_{\times}]\boldsymbol{R}$$

Essential Matrix
(Longuet-Higgins, 1981)

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

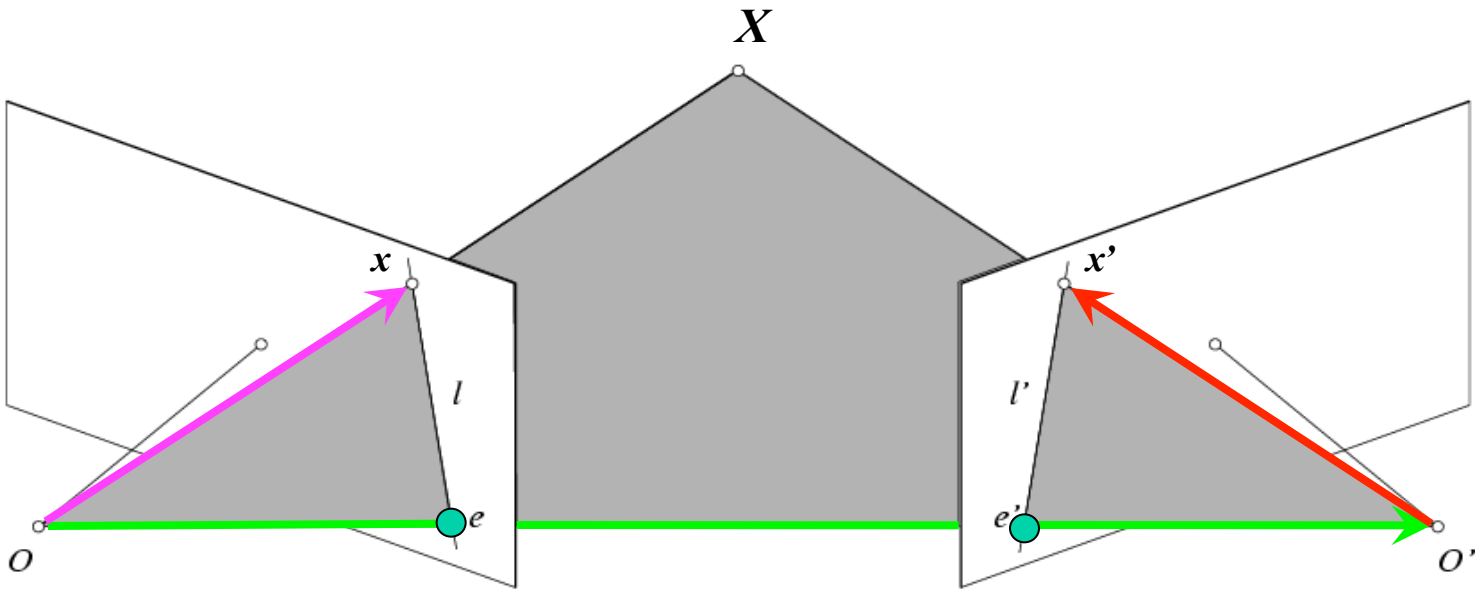
Epipolar constraint: Calibrated case



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

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

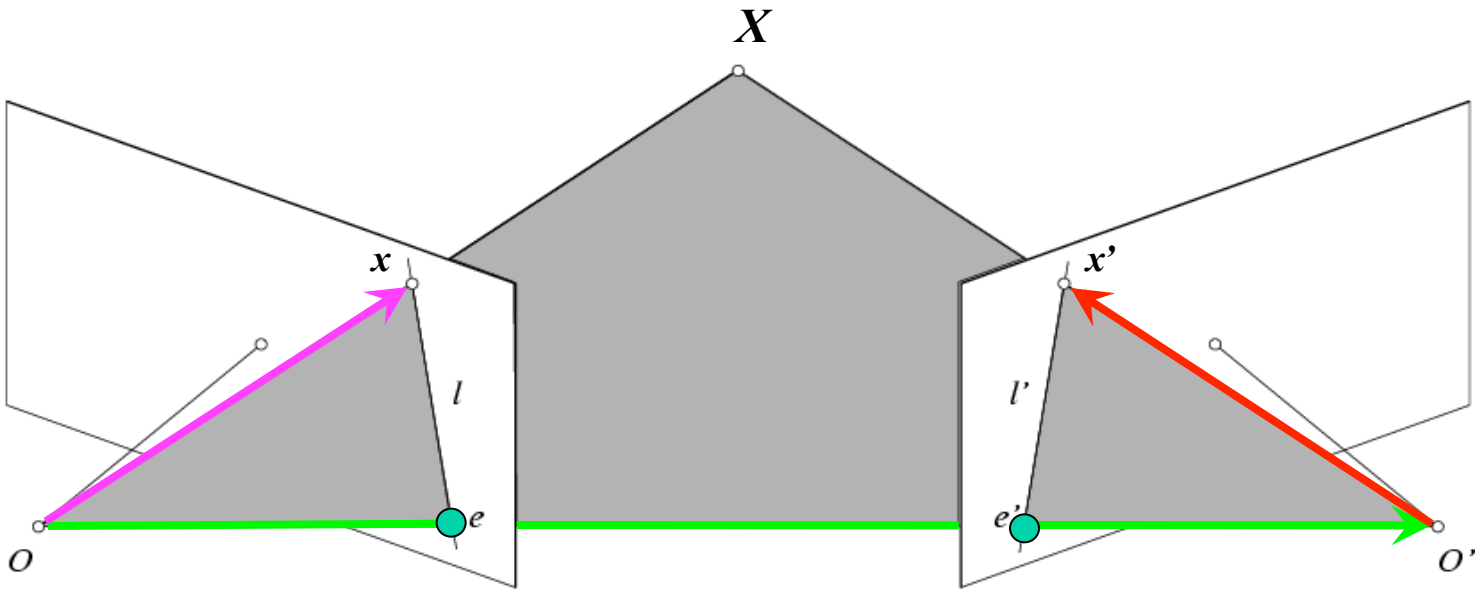
Epipolar constraint: Uncalibrated case



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

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

Epipolar constraint: Uncalibrated case



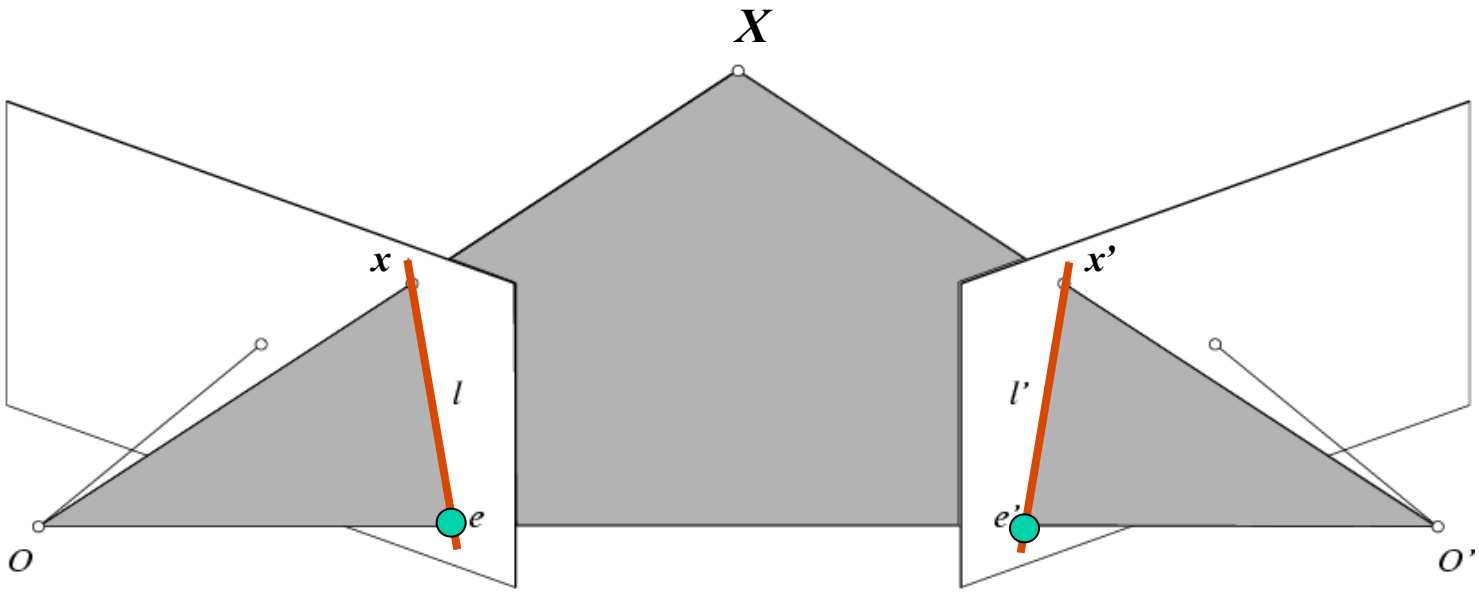
$$\hat{x}'^T E \hat{x} = 0 \quad \Rightarrow \quad x'^T F x = 0 \quad \text{with} \quad F = K'^{-T} E K^{-1}$$

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

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

Fundamental Matrix
(Faugeras and Luong, 1992)

Epipolar constraint: Uncalibrated case



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

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

The eight-point algorithm

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

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

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

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

Minimize:

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

under the constraint $\|\boldsymbol{F}\|^2=1$