Stereo I

CSC420
David Lindell
University of Toronto
cs.toronto.edu/~lindell/teaching/420
Slide credit: Babak Taati ←Ahmed Ashraf ←Sanja Fidler, Yannis Gkioulekas

TCIG Toronto Computational Imaging Group

Logistics

- A4 is out. Due date is March 29
- Final exam April 18th BA1130 9AM – 12 PM
  - multiple choice, short answer, long answer

Overview

- Recap camera matrix and perspective projection
- Two-view geometry

Overview

- Recap camera matrix and perspective projection
- Two-view geometry

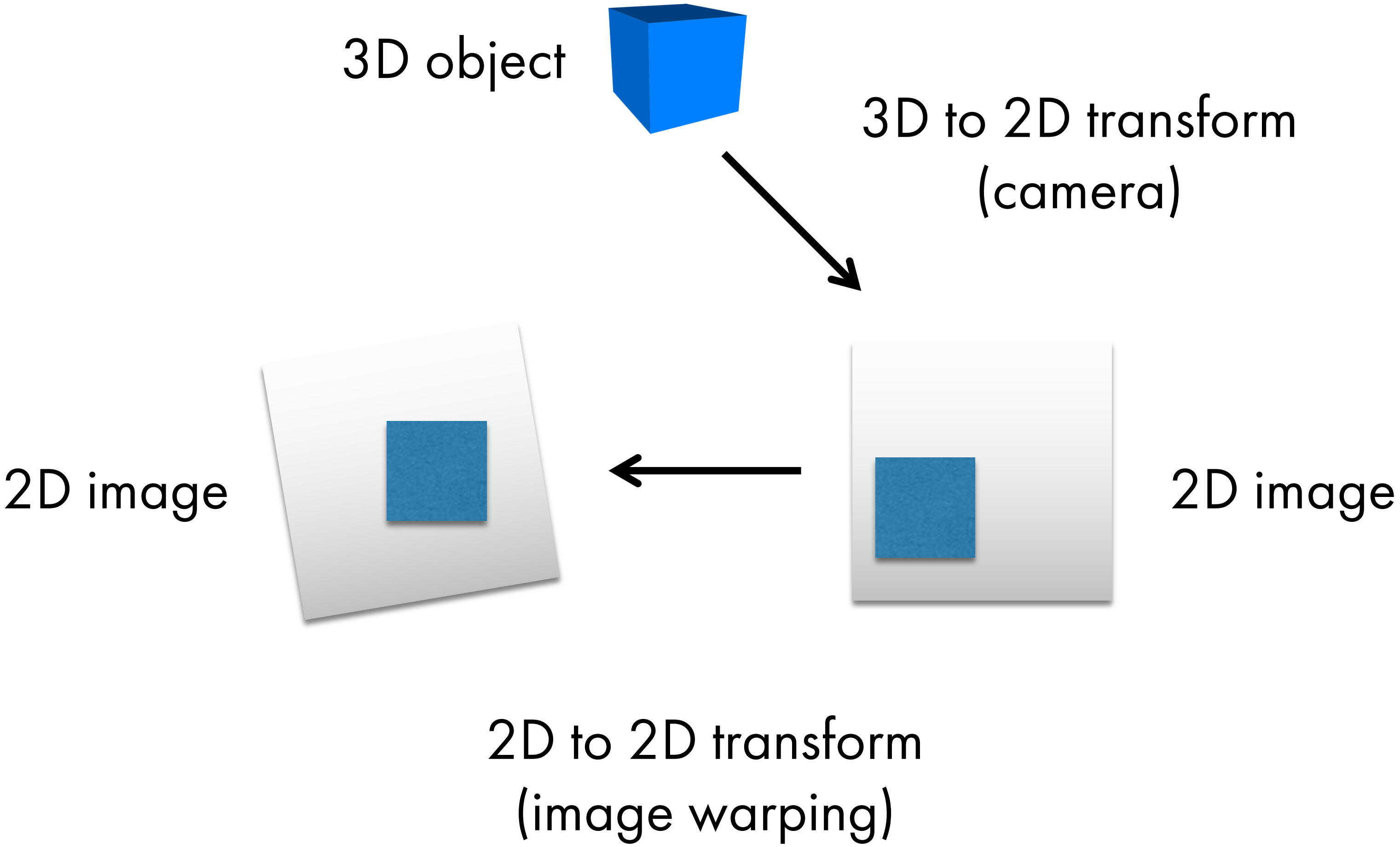
The camera as a coordinate transformation

A camera is a mapping from:

the 3D world

to:

a 2D image



The camera as a coordinate transformation

A camera is a mapping from:

the 3D world

to:

a 2D image

homogeneous coordinates

$$\boldsymbol{x} = \mathbf{P}\mathbf{X}$$

2D image point

camera matrix

3D world point

What are the dimensions of each variable?

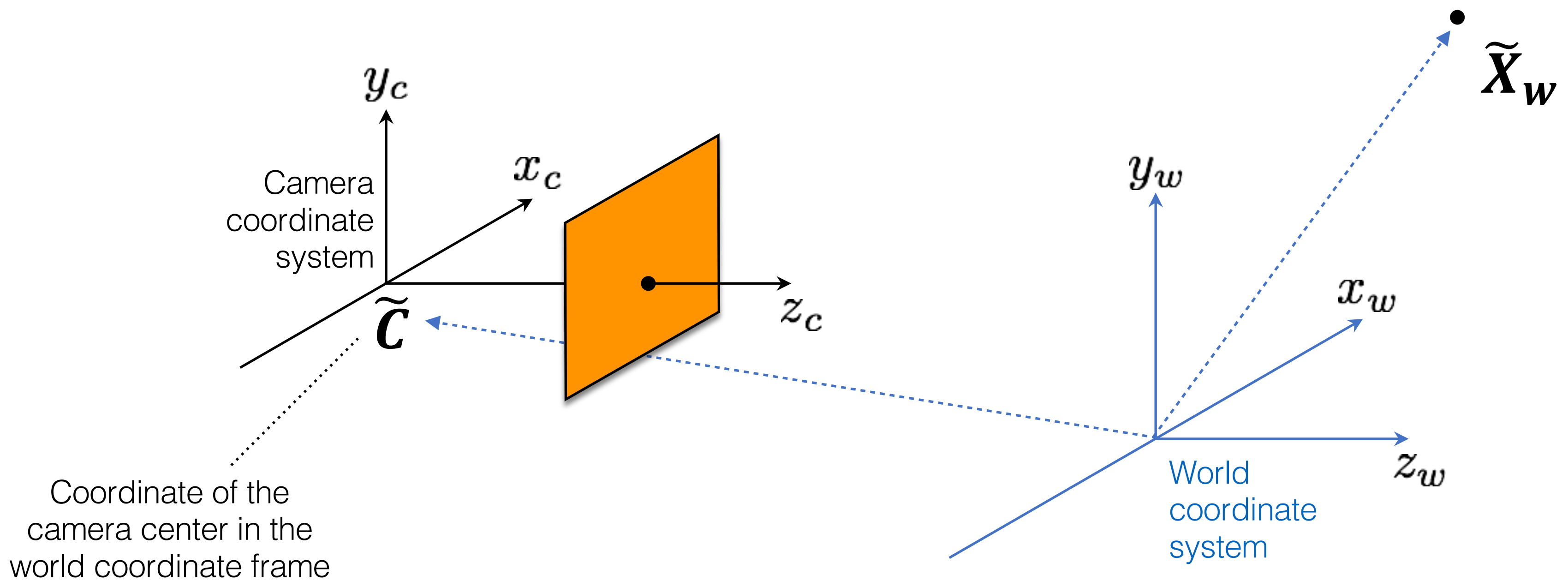
The camera as a coordinate transformation

$$\boldsymbol{x} = \mathbf{PX}$$

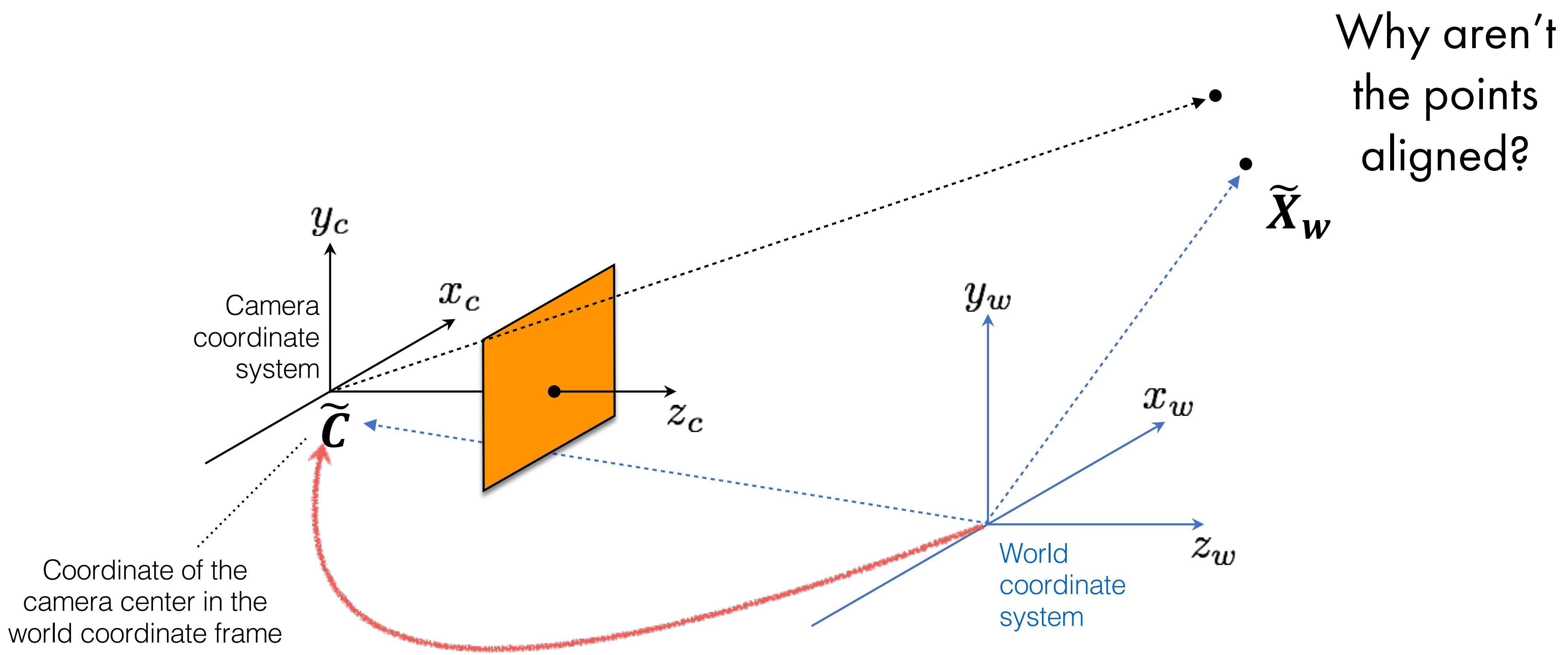
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

homogeneous image coordinates	camera matrix	homogeneous world coordinates
3 x 1	3 x 4	4 x 1

World-to-camera coordinate system transformation



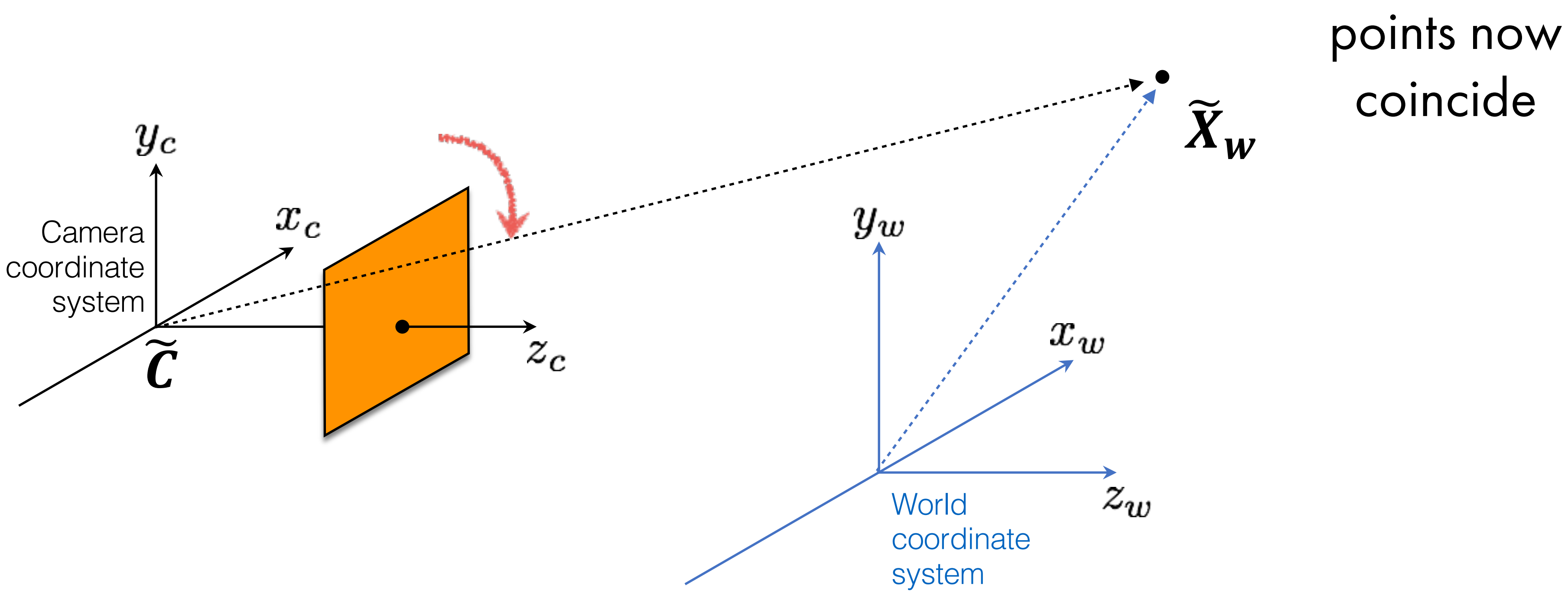
World-to-camera coordinate system transformation



$$(\tilde{X}_w - \tilde{C})$$

translate

World-to-camera coordinate system transformation



$$\boldsymbol{R} \cdot (\tilde{\boldsymbol{X}}_{\boldsymbol{w}} - \tilde{\boldsymbol{C}})$$

rotate translate

Modeling the coordinate system transformation

In heterogeneous coordinates, we have:

$$\widetilde{\mathbf{X}}_{\mathbf{c}} = \mathbf{R} \cdot \left(\widetilde{\mathbf{X}}_{\mathbf{w}} - \tilde{\mathbf{C}}\right)$$

Modeling the coordinate system transformation

In heterogeneous coordinates, we have:

$$\widetilde{\mathbf{X}}_{\mathbf{c}} = \mathbf{R} \cdot \left(\widetilde{\mathbf{X}}_{\mathbf{w}} - \tilde{\mathbf{C}}\right)$$

In homogeneous coordinates, we have:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & -\mathbf{RC} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

or

$$\mathbf{X}_{\mathbf{c}} = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{X}_{\mathbf{w}}$$

Putting it all together

We can write everything into a single projection:

$$\mathbf{x} = \mathbf{P}\mathbf{X}_{\mathbf{w}}$$

Putting it all together

We can write everything into a single projection:

$$\mathbf{x} = \mathbf{P}\mathbf{X}_{\mathbf{w}}$$

The camera matrix now looks like:

$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{I} \quad | \quad \mathbf{0}] \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ \mathbf{0} & 1 \end{bmatrix}$$

Putting it all together

We can write everything into a single projection:

$$\mathbf{x} = \mathbf{P}\mathbf{X_w}$$

The camera matrix now looks like:

$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{I} \quad | \quad \mathbf{0}] \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ \mathbf{0} & 1 \end{bmatrix}$$

intrinsic parameters (3 x 3): correspond to camera internals (image-to-image transformation)

Putting it all together

We can write everything into a single projection:

$$\mathbf{x} = \mathbf{P}\mathbf{X_w}$$

The camera matrix now looks like:

$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{I} \quad | \quad \mathbf{0}] \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ \mathbf{0} & 1 \end{bmatrix}$$

intrinsic parameters (3 x 3): correspond to camera internals (image-to-image transformation)

perspective projection (3 x 4): maps 3D to 2D points (camera-to-image transformation)

Putting it all together

We can write everything into a single projection:

$$\mathbf{x} = \mathbf{P}\mathbf{X_w}$$

The camera matrix now looks like:

$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{I} \quad | \quad \mathbf{0}] \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ \mathbf{0} & 1 \end{bmatrix}$$

intrinsic parameters (3 x 3): correspond to camera internals (image-to-image transformation)

perspective projection (3 x 4): maps 3D to 2D points (camera-to-image transformation)

extrinsic parameters (4 x 4): correspond to camera externals (world-to-camera transformation)

Putting it all together

We can write everything into a single projection:

$$\mathbf{x} = \mathbf{P}\mathbf{X_w}$$

The camera matrix now looks like:

$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\mathbf{R} \;\vdots\; -\mathbf{RC} \right]$$

intrinsic parameters (3 x 3): correspond to camera internals

extrinsic parameters (3 x 4): correspond to camera externals (world-to-image transformation)

General pinhole camera matrix

We can decompose the camera matrix like this:

$$\mathbf{P} = \mathbf{KR}[\mathbf{I}| - \mathbf{C}]$$

(translate first then rotate)

$$\boldsymbol{R} \cdot (\widetilde{\boldsymbol{X}}_{\boldsymbol{w}} - \widetilde{\boldsymbol{C}})$$

rotate translate

General pinhole camera matrix

We can decompose the camera matrix like this:

$$\mathbf{P} = \mathbf{KR}[\mathbf{I}| - \mathbf{C}]$$

(translate first then rotate)

$$\boldsymbol{R} \cdot (\boldsymbol{X_w} - \boldsymbol{C})$$

rotate translate

Another way to write the mapping:

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$$

where $\mathbf{t} = -\mathbf{RC}$

(rotate first then translate)

$$\boldsymbol{R} \cdot \boldsymbol{X_w} - \boldsymbol{R} \cdot \boldsymbol{C}$$

rotate translate

General pinhole camera matrix

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$$

General pinhole camera matrix

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$$

$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc:c} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \end{array} \right]$$

intrinsic parameters

extrinsic parameters

General pinhole camera matrix

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$$

$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \end{array} \right]$$

intrinsic parameters

extrinsic parameters

$$\mathbf{R} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \qquad \mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

3D rotation

3D translation

Recap

What is the size and meaning of each term in the camera matrix?

$$\mathbf{P} = \mathbf{K}\mathbf{R}[\mathbf{I}| - \mathbf{C}]$$

? ? ? ?

Recap

What is the size and meaning of each term in the camera matrix?

$$\mathbf{P} = \mathbf{K}\mathbf{R}[\mathbf{I}| - \mathbf{C}]$$

3x3 intrinsics

?

?

?

Recap

What is the size and meaning of each term in the camera matrix?

$$\mathbf{P} = \mathbf{K}\mathbf{R}[\mathbf{I}| - \mathbf{C}]$$

3x3
intrinsics

3x3
3D rotation

?

?

Recap

What is the size and meaning of each term in the camera matrix?

$$\mathbf{P} = \mathbf{K}\mathbf{R}[\mathbf{I}| - \mathbf{C}]$$

3x3
intrinsics

3x3
3D rotation

3x3
identity

?

Recap

What is the size and meaning of each term in the camera matrix?

$$\mathbf{P} = \mathbf{K}\mathbf{R}[\mathbf{I}| - \mathbf{C}]$$

$\mathbf{K}$	$\mathbf{R}$	$\mathbf{I}$	$\mathbf{C}$
3x3 intrinsics	3x3 3D rotation	3x3 identity	3x1 3D translation

Quiz

The camera matrix relates what two quantities?

$$\boldsymbol{x} = \mathbf{P}\mathbf{X}$$

Quiz

The camera matrix relates what two quantities?

$$\boldsymbol{x} = \mathbf{P}\mathbf{X}$$

homogeneous 3D points to 2D image points

Quiz

The camera matrix relates what two quantities?

$$\boldsymbol{x} = \mathbf{PX}$$

homogeneous 3D points to 2D image points

The camera matrix can be decomposed into?

Quiz

The camera matrix relates what two quantities?

$$\boldsymbol{x} = \mathbf{P}\mathbf{X}$$

homogeneous 3D points to 2D image points

The camera matrix can be decomposed into?

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$$

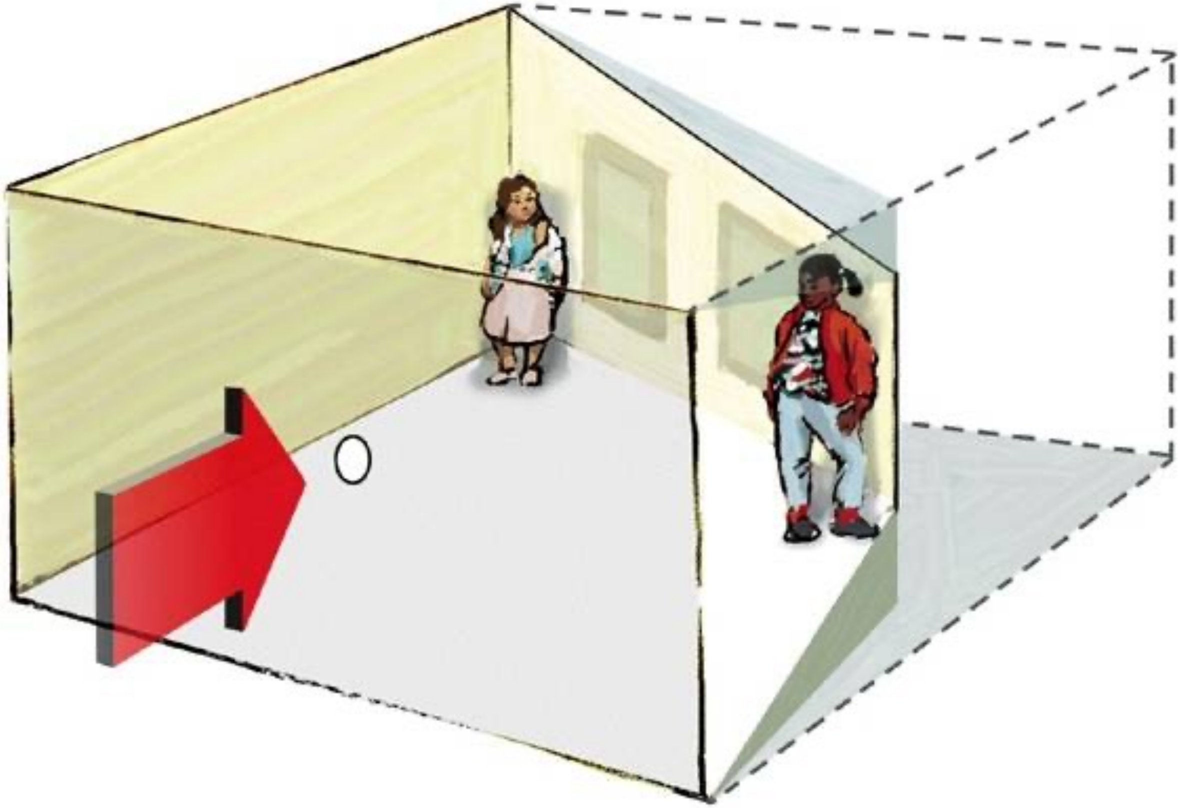
intrinsic and extrinsic parameters

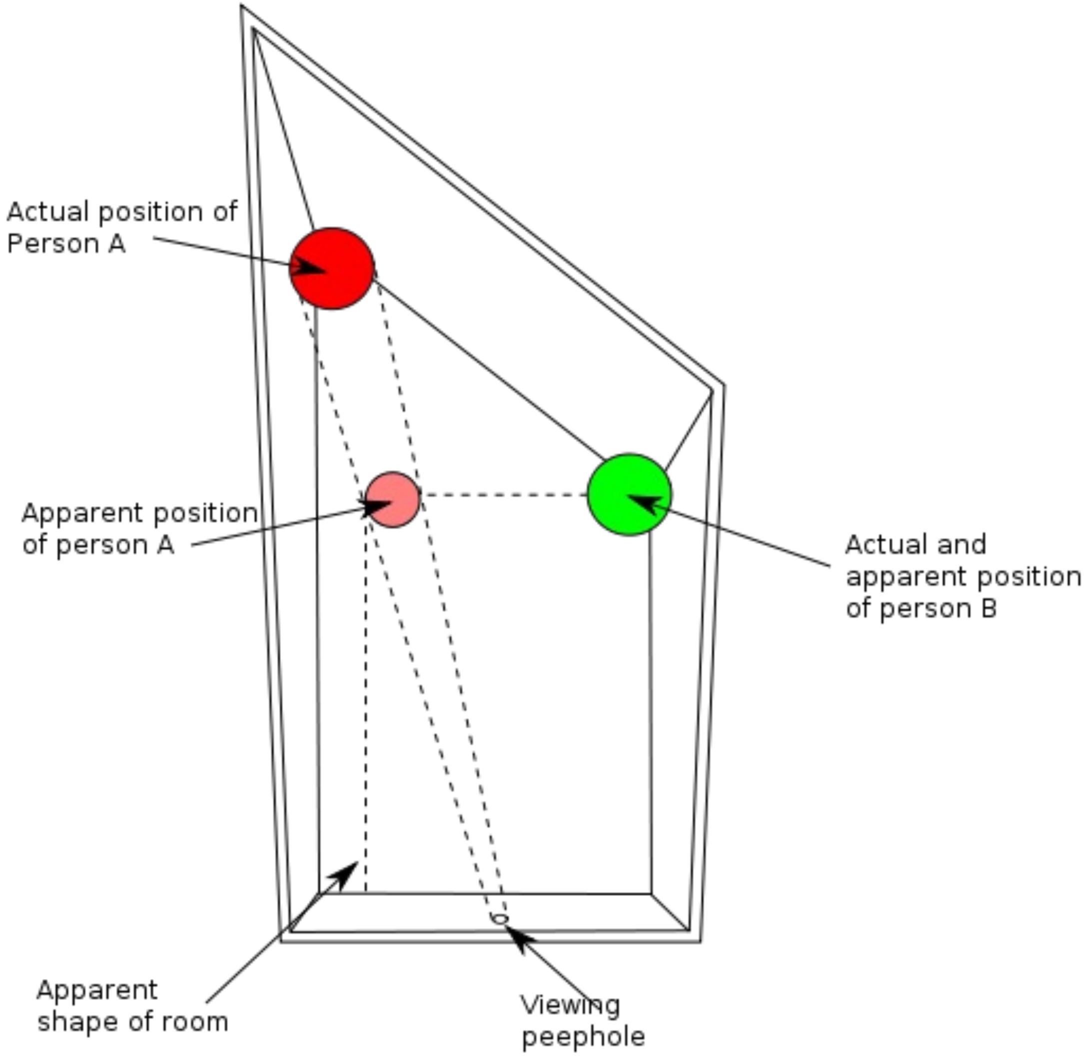
Perspective distortion

Forced perspective

The Ames room illusion

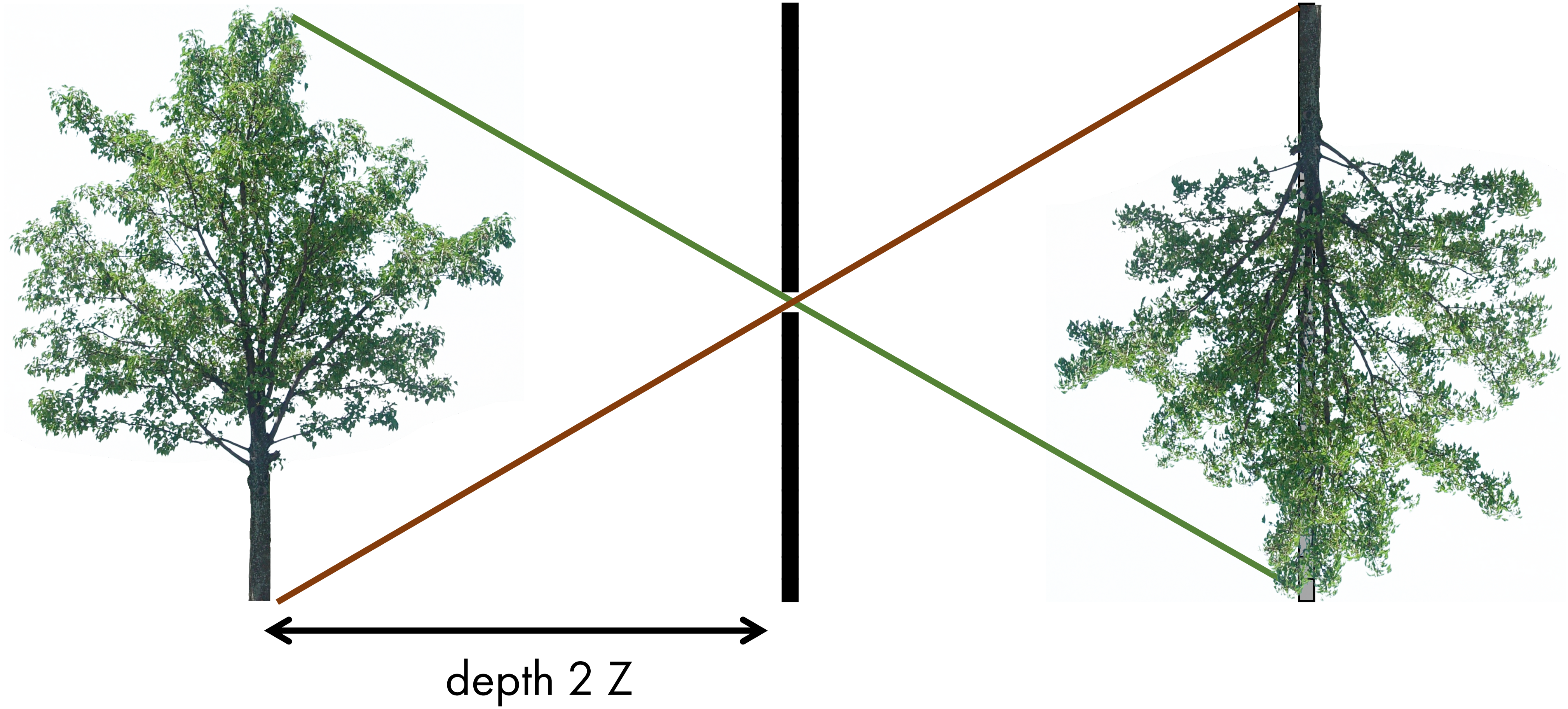
The Ames room illusion



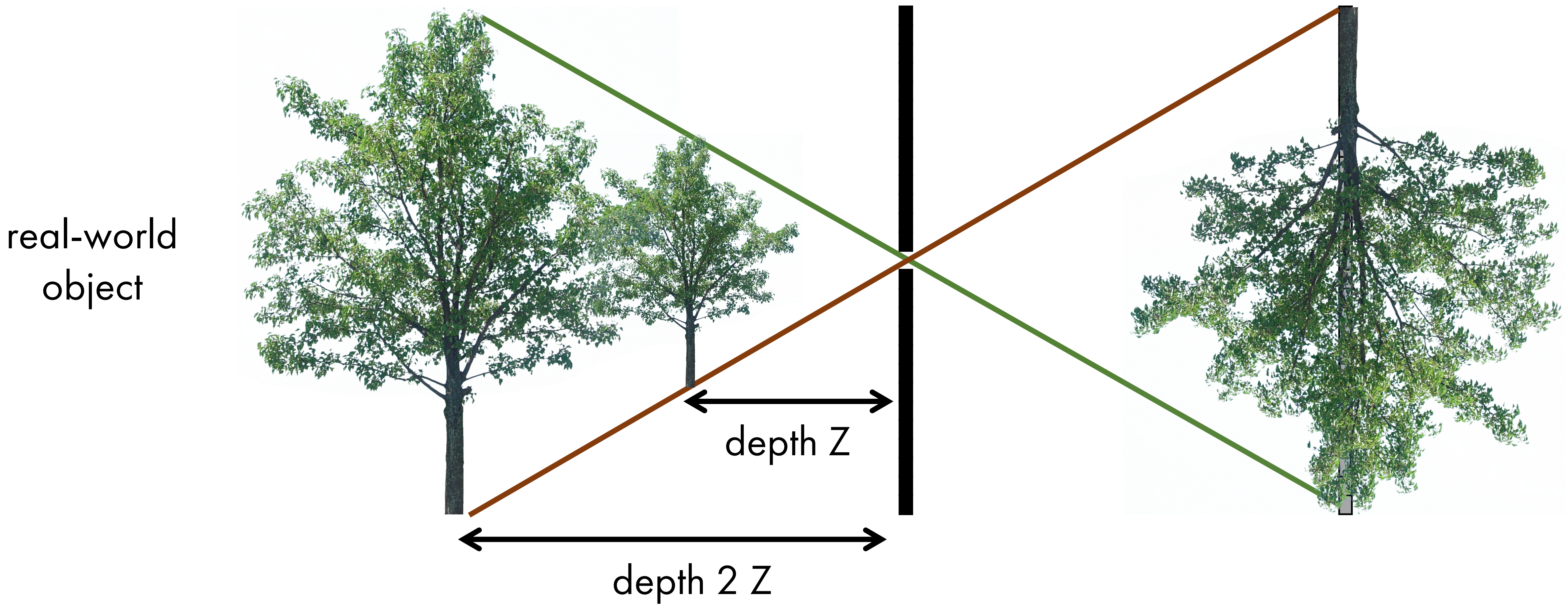


Magnification depends on depth

real-world object



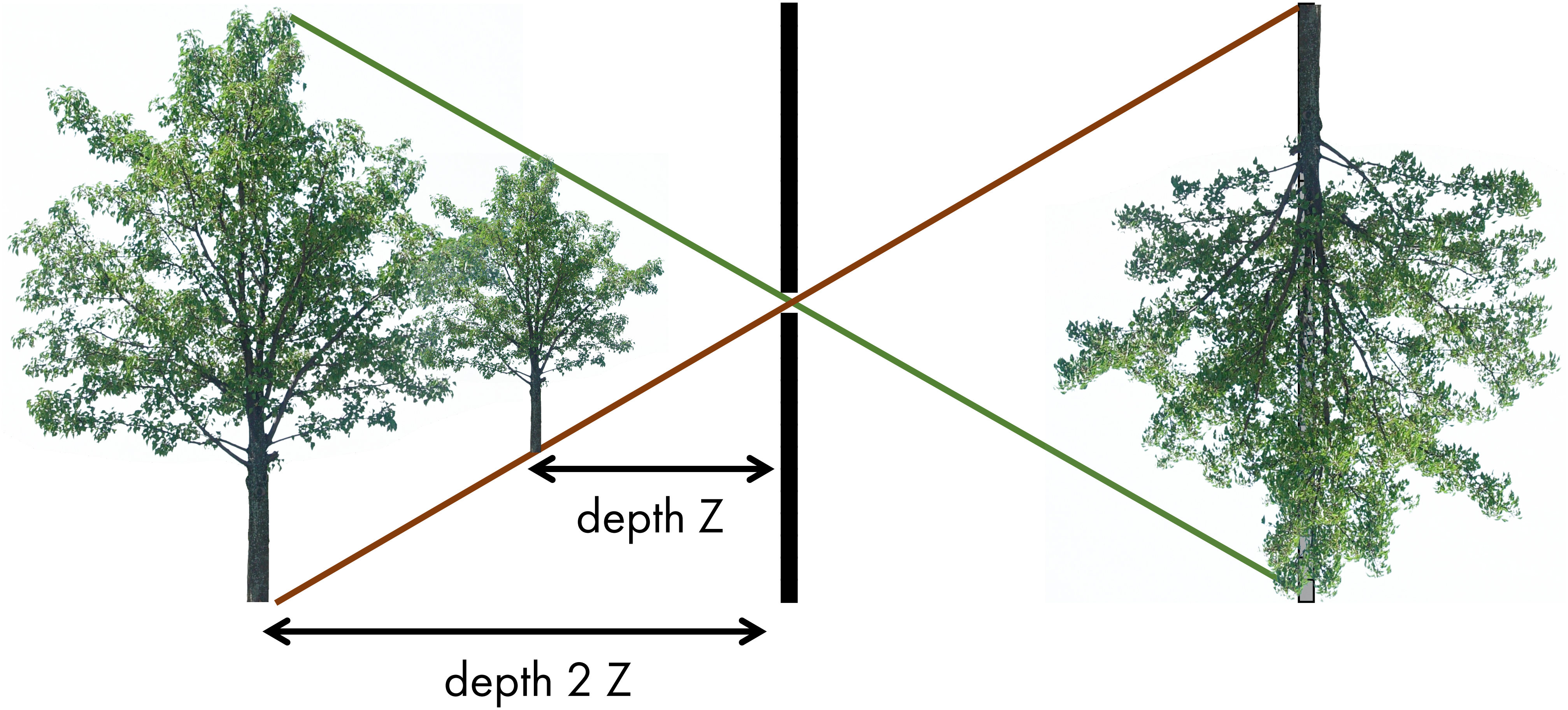
Magnification depends on depth



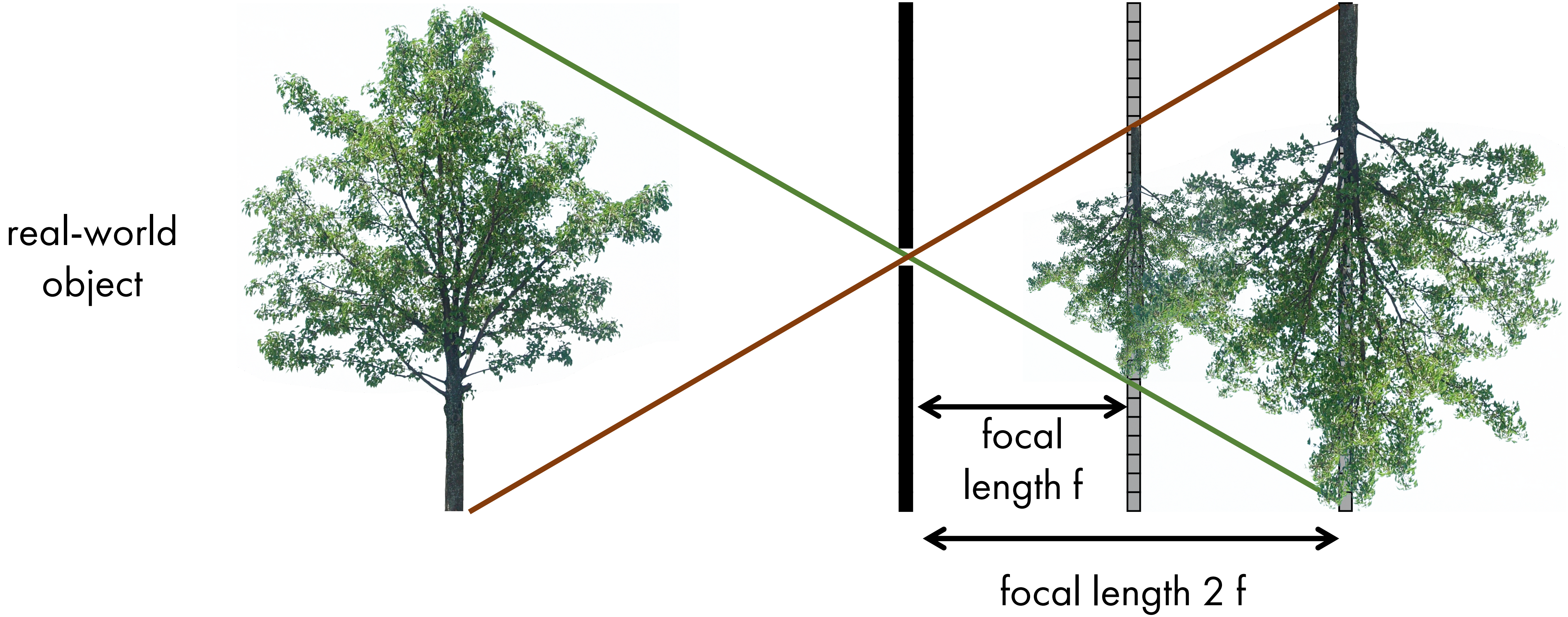
Magnification depends on depth

What happens as we change the focal length?

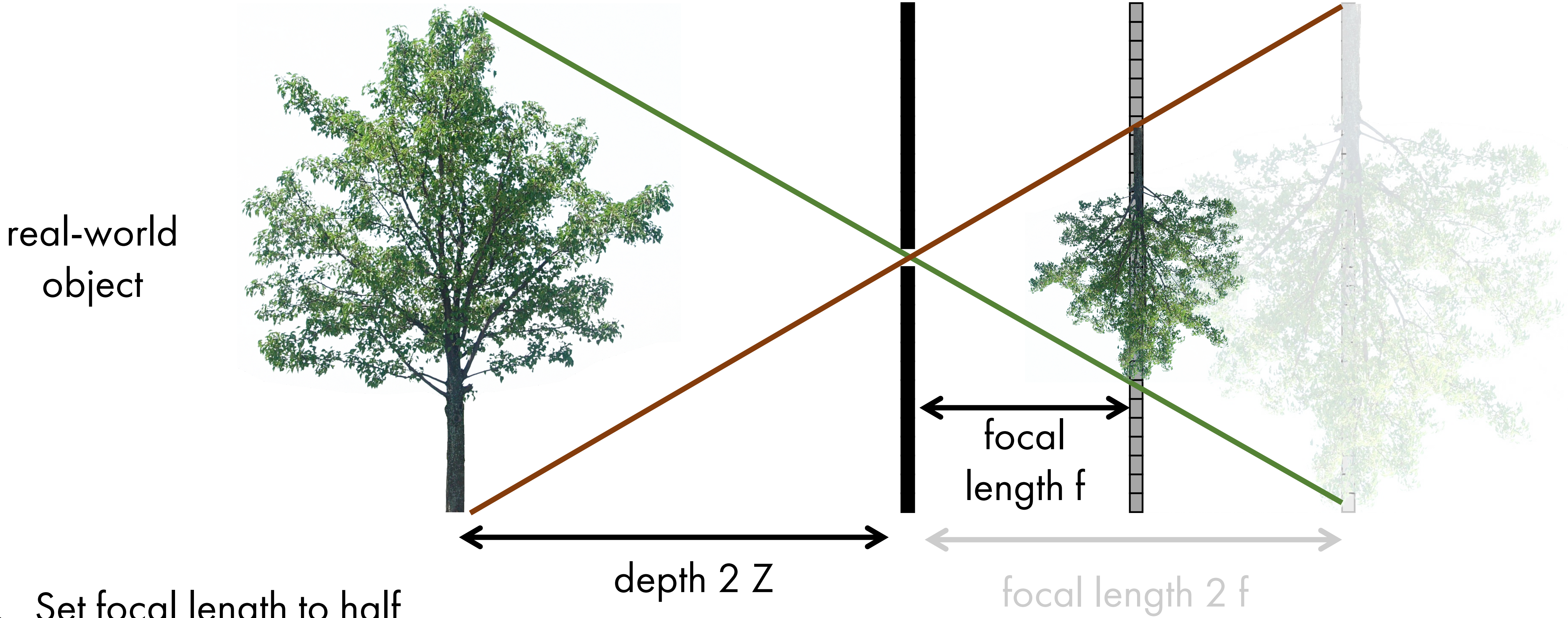
real-world
object



Magnification depends on focal length



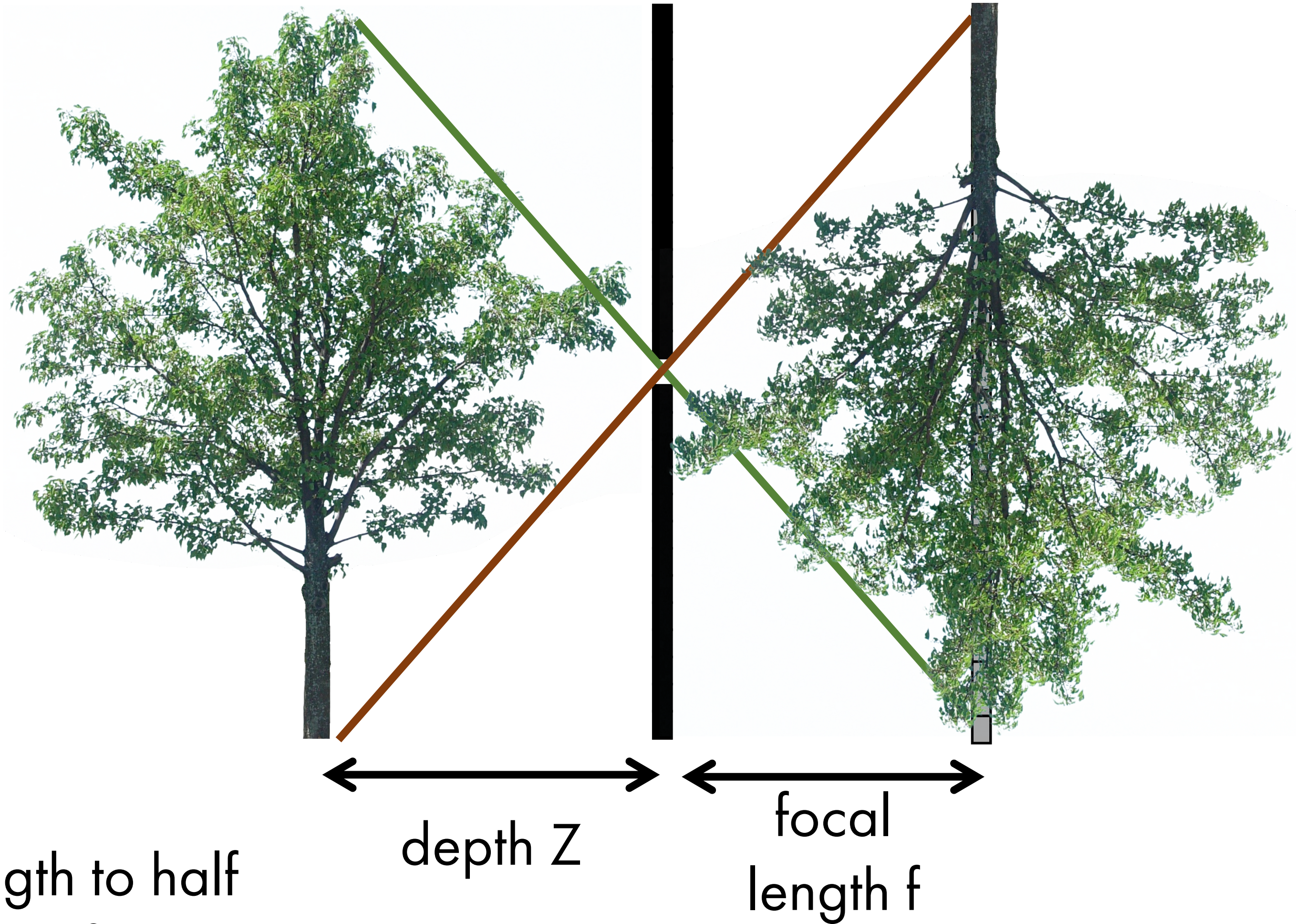
What if...



1. Set focal length to half

What if...

real-world
object



Is this the same image as the one I had at focal length 2f and distance 2Z?

1. Set focal length to half
2. Set depth to half

Perspective distortion

long focal length

mid focal length

short focal length

Perspective distortion

Vertigo effect

Named after Alfred Hitchcock's movie

- also known as "dolly zoom"

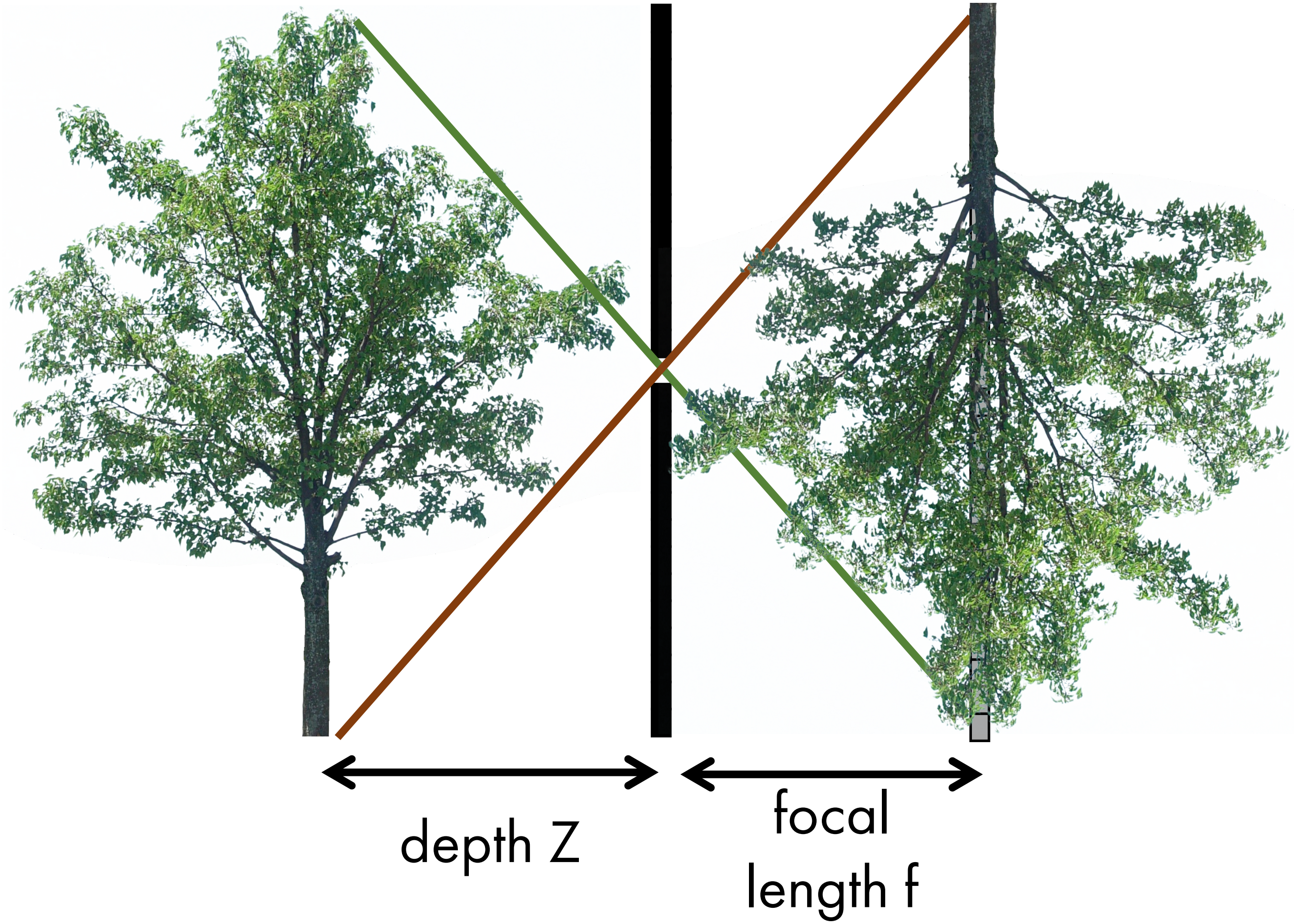
Vertigo effect

How would you create this effect?

Other camera models

What if...

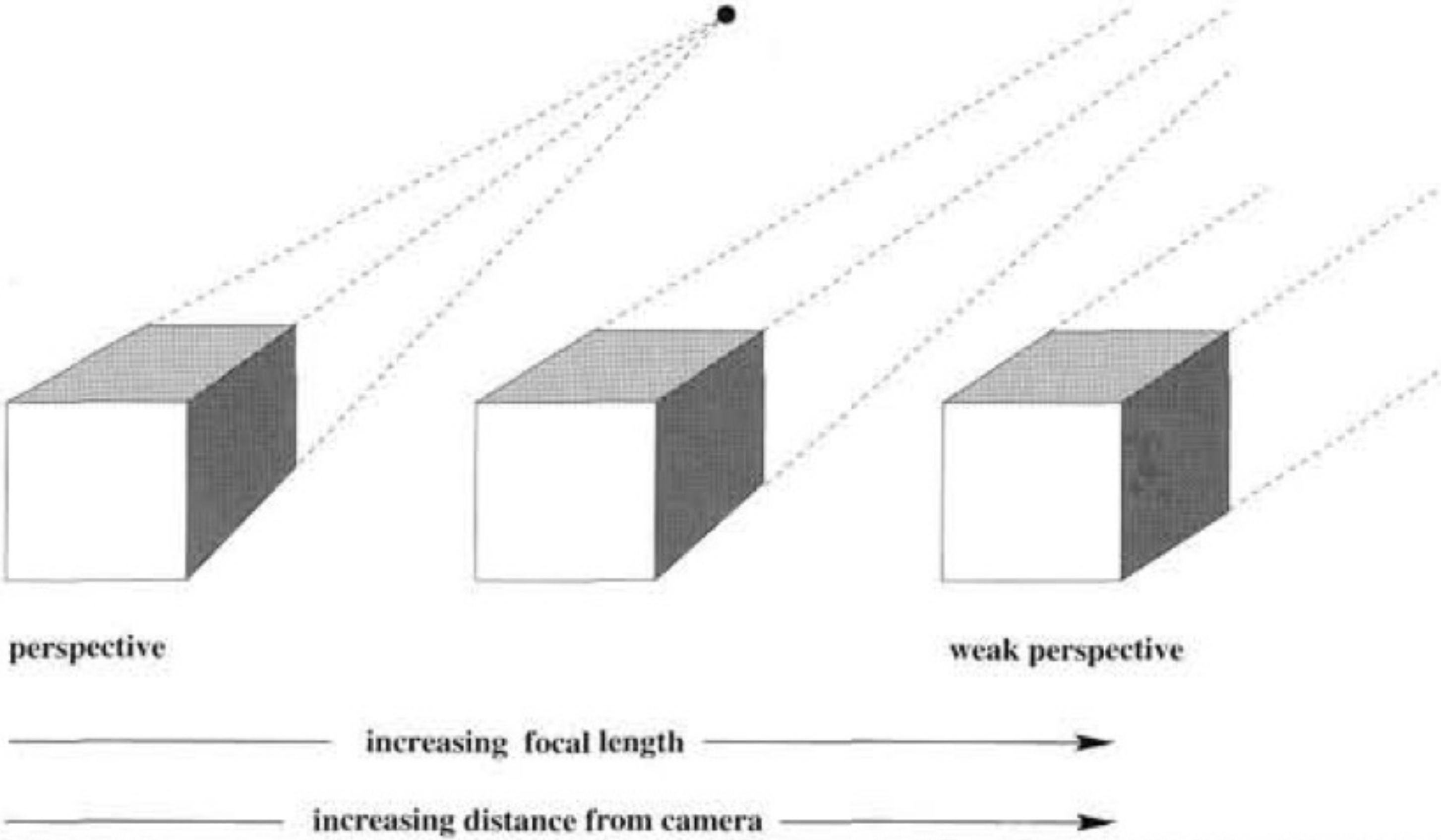
real-world object



... we continue increasing Z and f while maintaining same magnification?

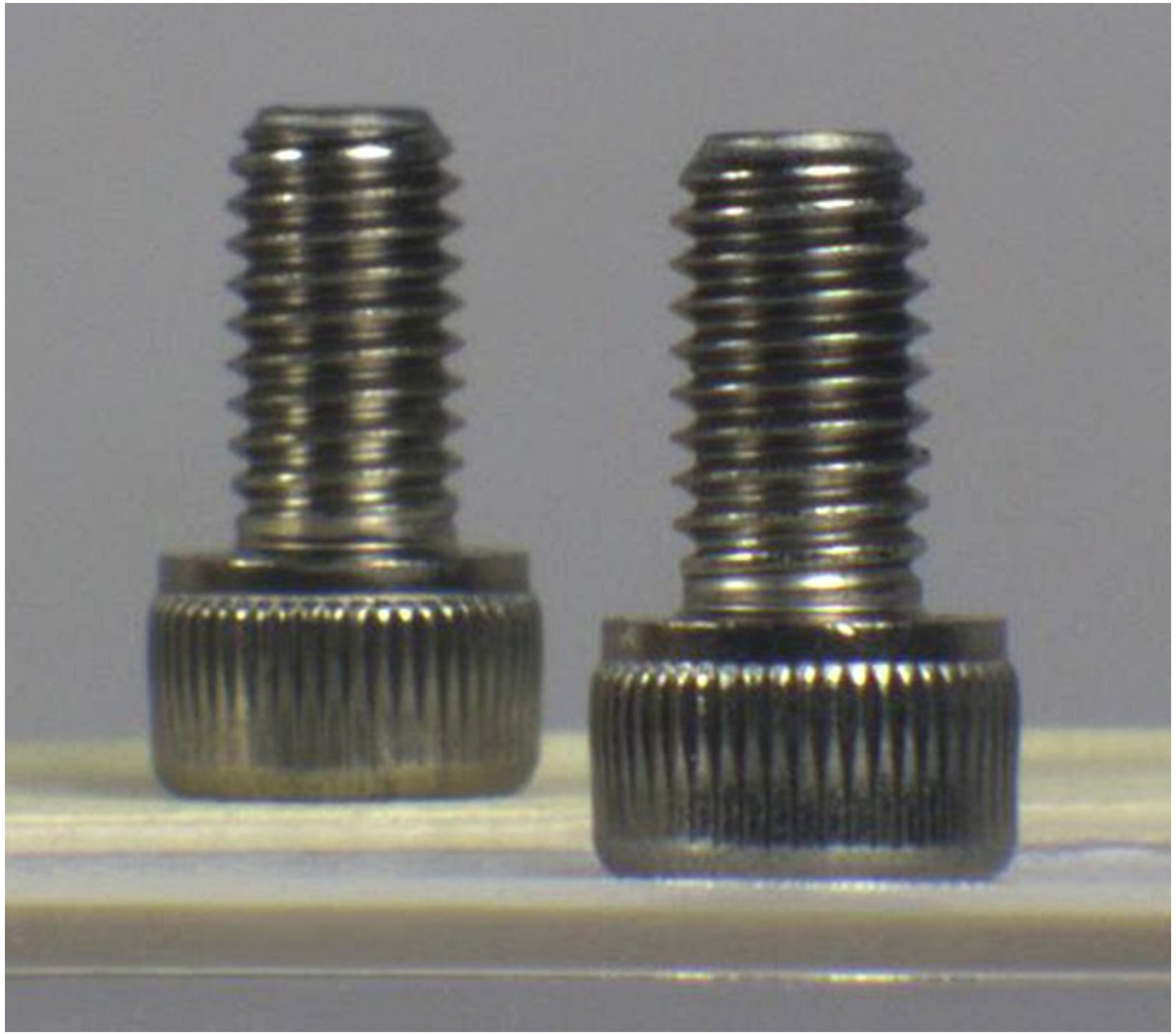
$$f \to \infty \text{ and } \frac{f}{Z} = \text{constant}$$

camera is close to object and has small focal length

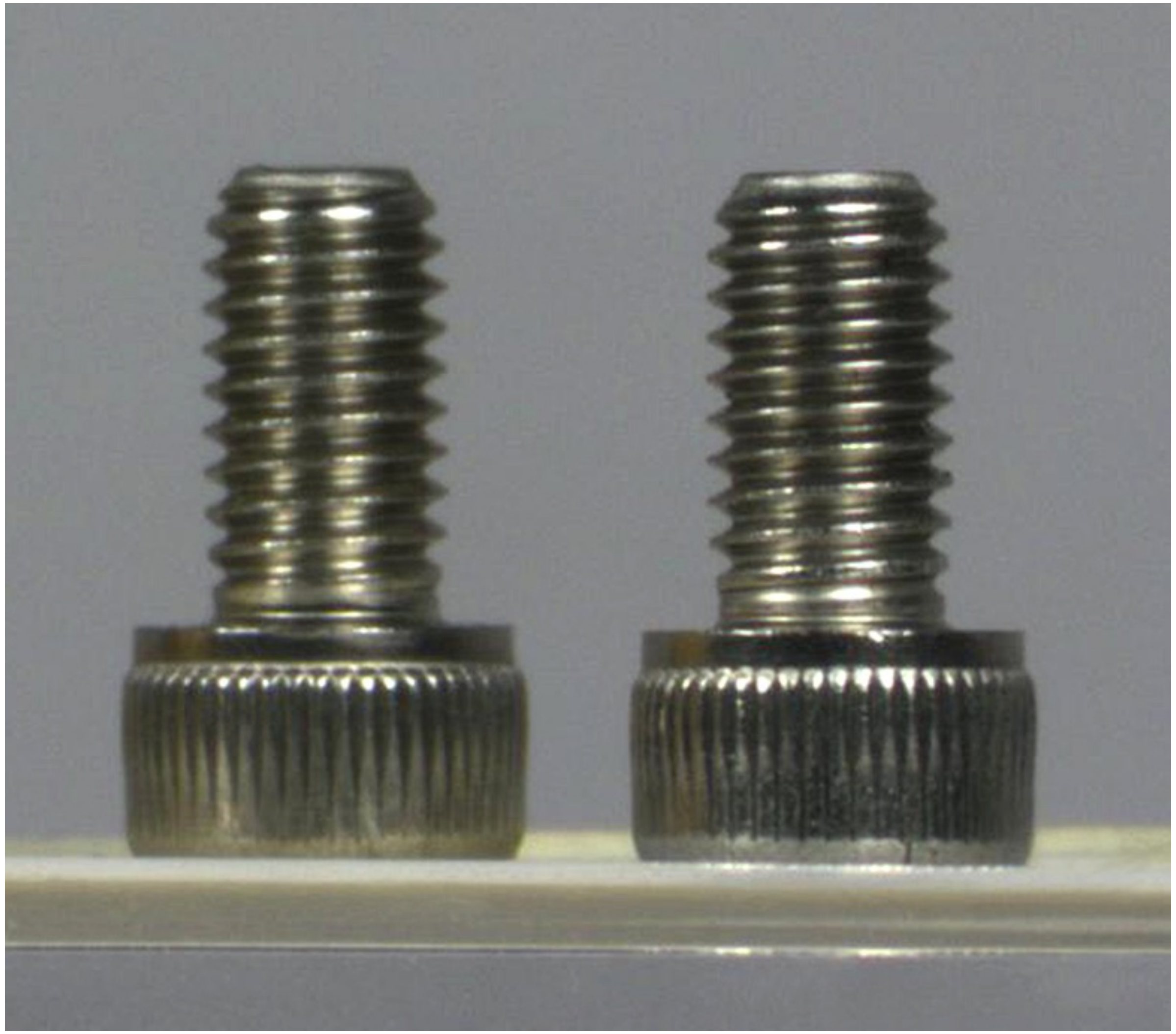


camera is far from object and has large focal length

Different cameras



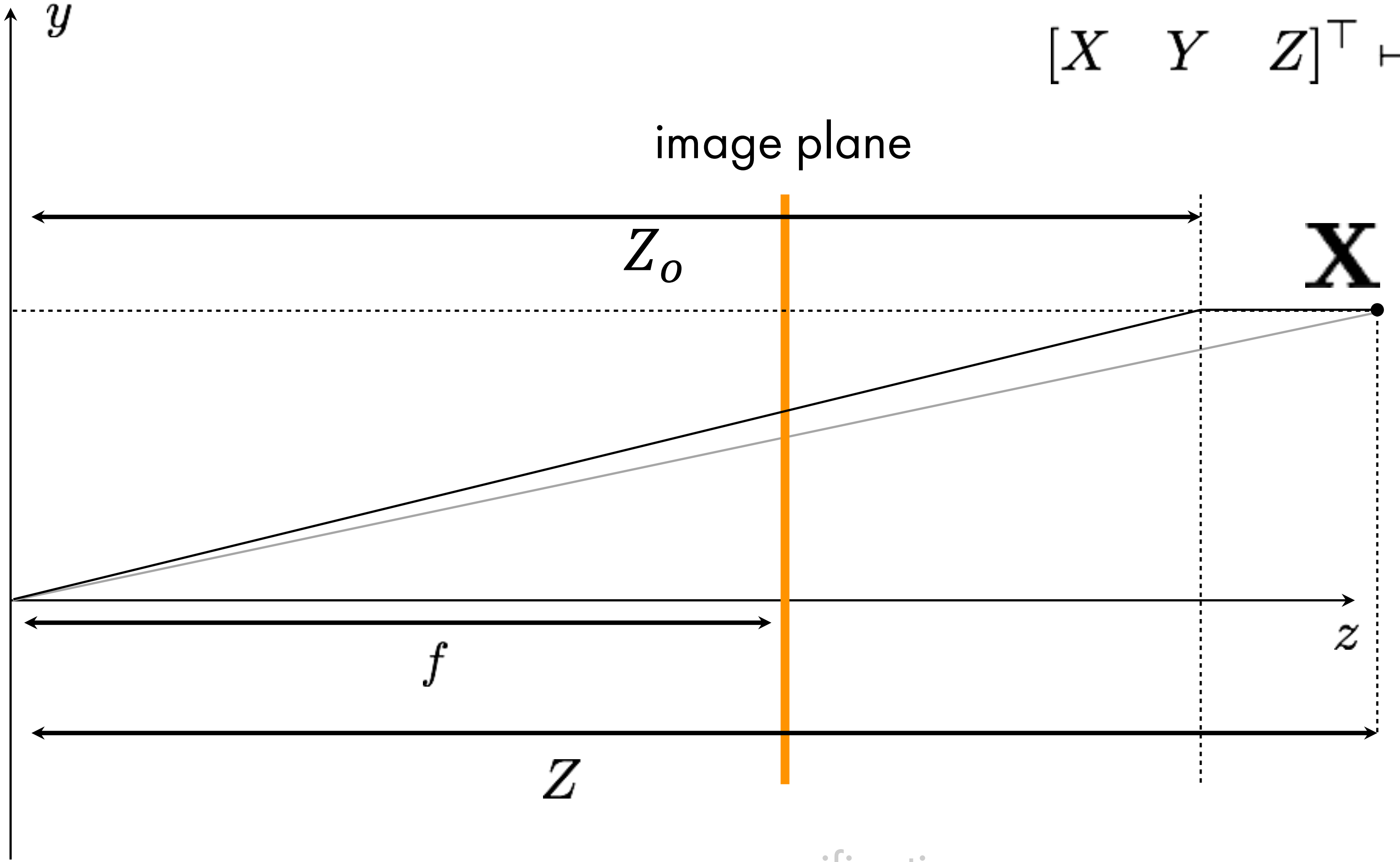
perspective camera



weak perspective camera

Weak perspective vs perspective camera

$$[X \quad Y \quad Z]^\top \mapsto [fX/Z_o \quad fY/Z_o]^\top$$



- magnification does not change with depth
- constant magnification depending on f and Z_o

magnification changes with depth

$$[X \quad Y \quad Z]^\top \mapsto [fX/Z \quad fY/Z]^\top$$

When can we assume a weak perspective camera?

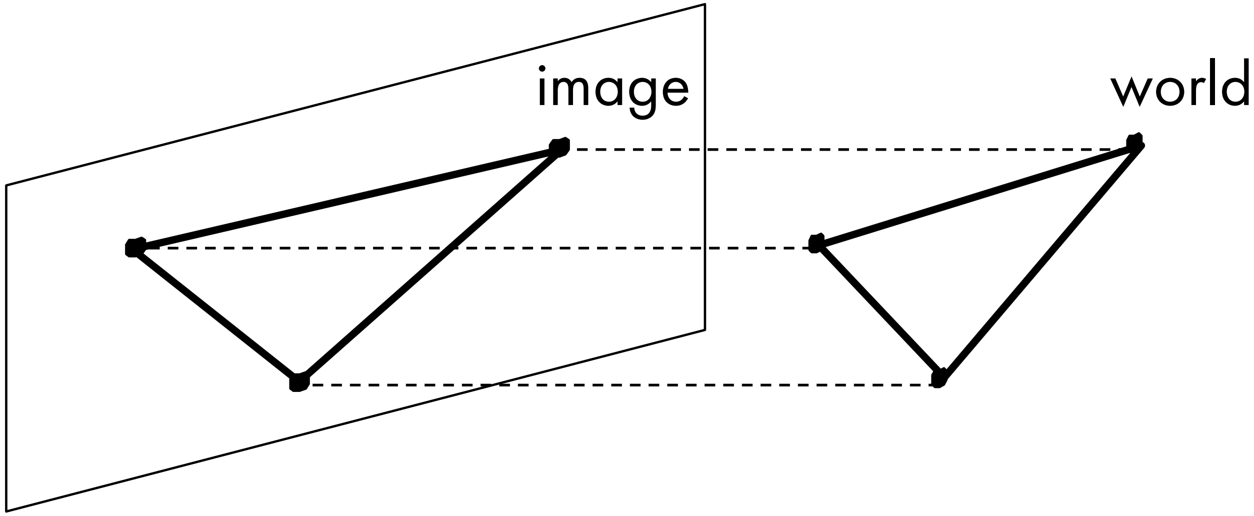
1. When the scene (or parts of it) is very far away.

Weak perspective projection applies to the mountains.

Orthographic camera

Special case of weak perspective camera where:

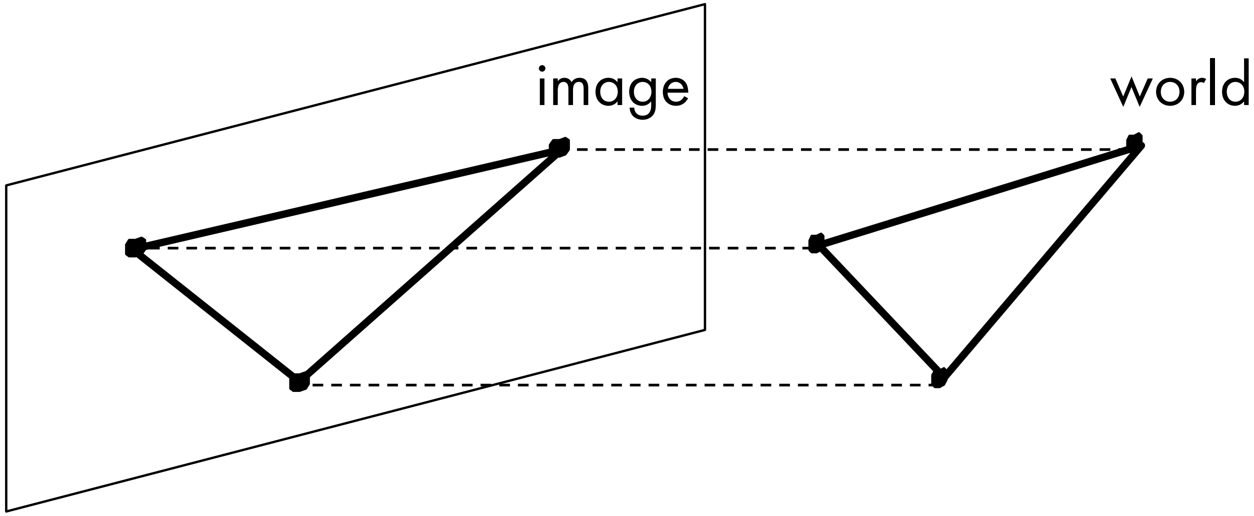
- constant magnification is equal to 1.



Orthographic camera

Special case of weak perspective camera where:

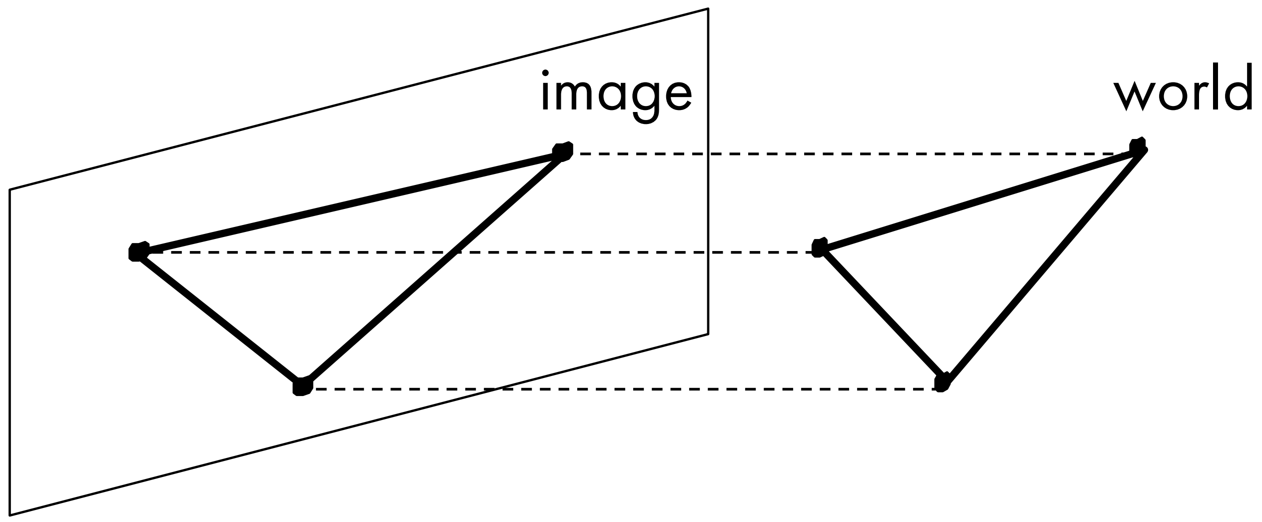
- constant magnification is equal to 1.
- there is no shift between camera and image origins.



Orthographic camera

Special case of weak perspective camera where:

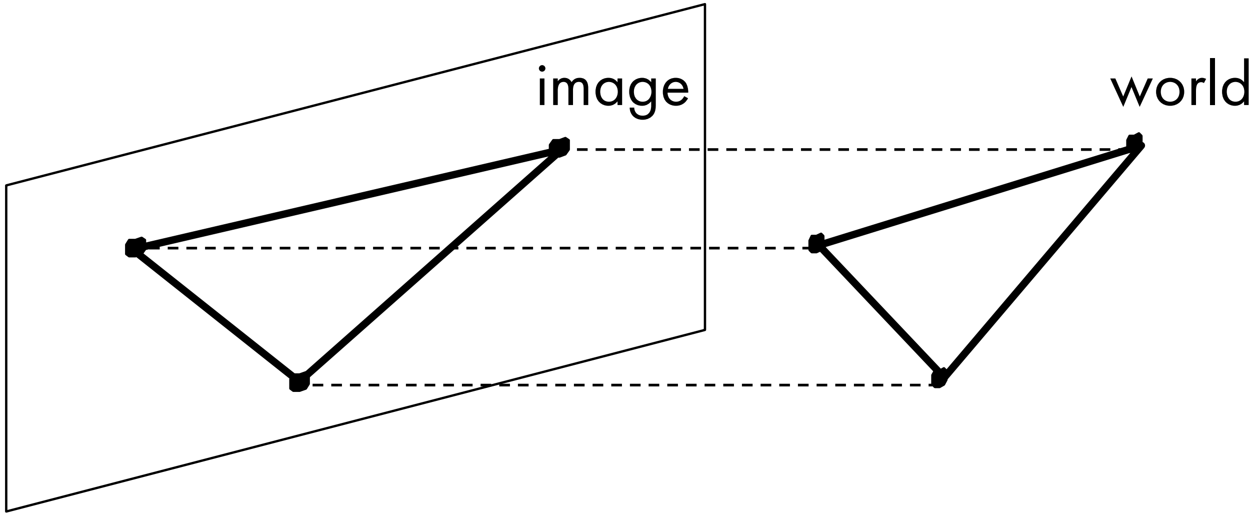
- constant magnification is equal to 1.
- there is no shift between camera and image origins.
- the world and camera coordinate systems are the same.



Orthographic camera

Special case of weak perspective camera where:

- constant magnification is equal to 1.
- there is no shift between camera and image origins.
- the world and camera coordinate systems are the same.

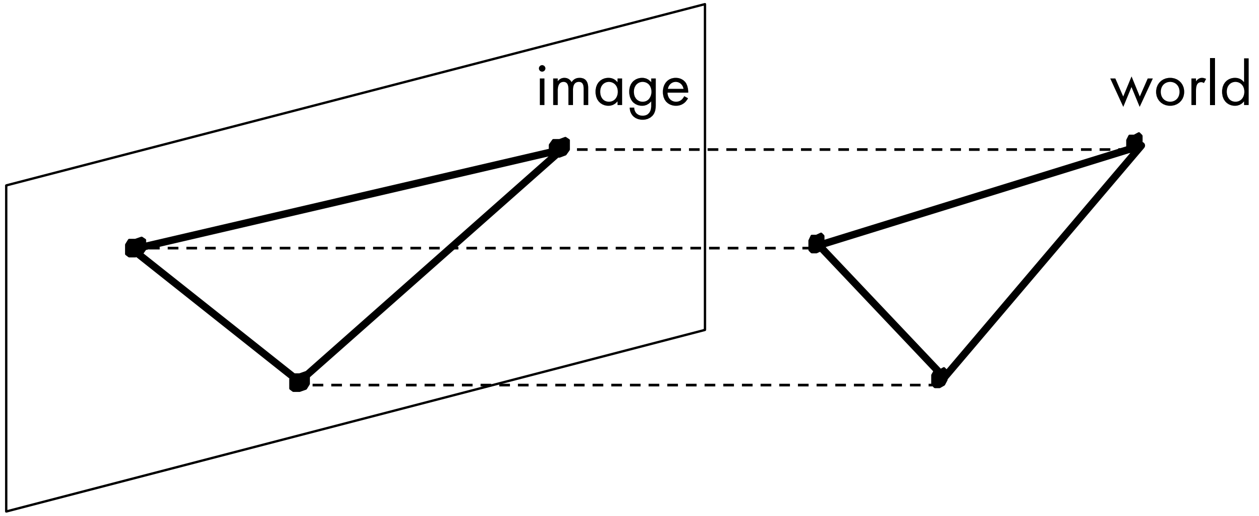


What is the camera matrix in this case?

Orthographic camera

Special case of weak perspective camera where:

- constant magnification is equal to 1.
- there is no shift between camera and image origins.
- the world and camera coordinate systems are the same.



$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

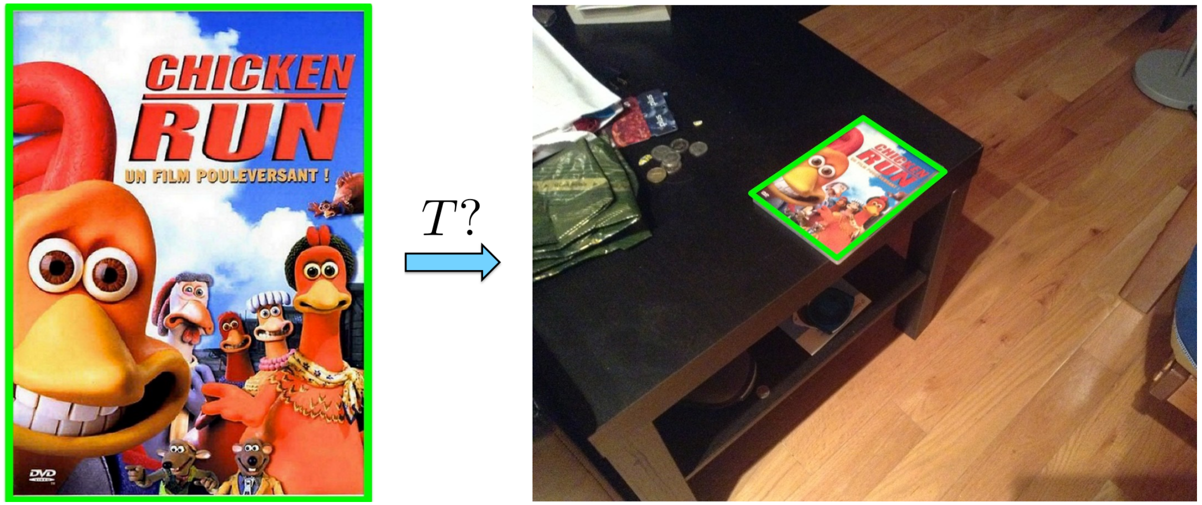
Overview

- Recap camera matrix and perspective projection
- Two-view geometry

Homography

- In Lecture 8 we said that a homography is a transformation that maps a projective plane to another projective plane.
- Defined by the following:

$$w\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



Homography

- Let’s revisit our transformation in the (new) light of perspective projection.

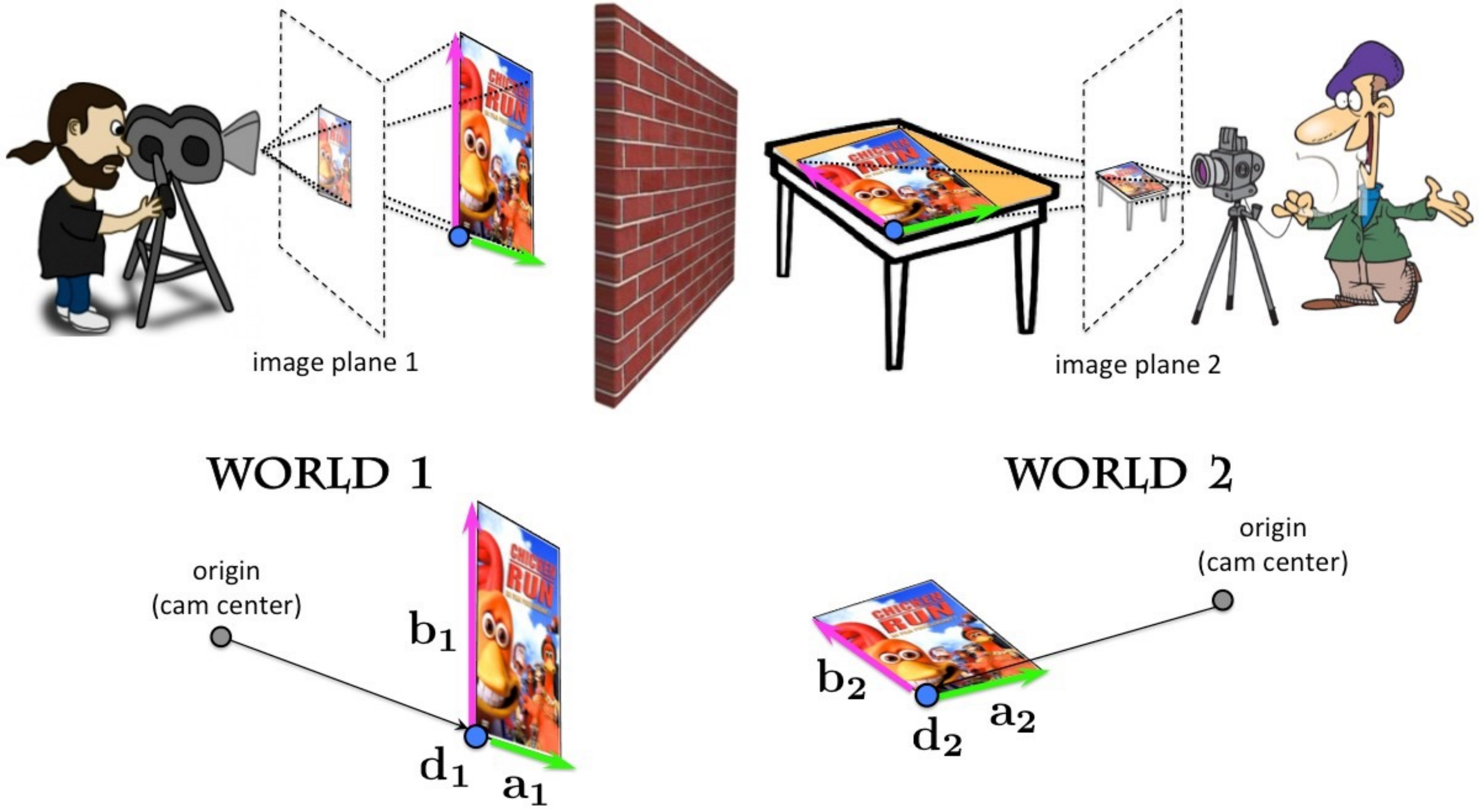
Homography

- Let's revisit our transformation in the (new) light of perspective projection.

We have our object in two different worlds, in two different poses relative to camera, two different photographers, and two different cameras.

Homography

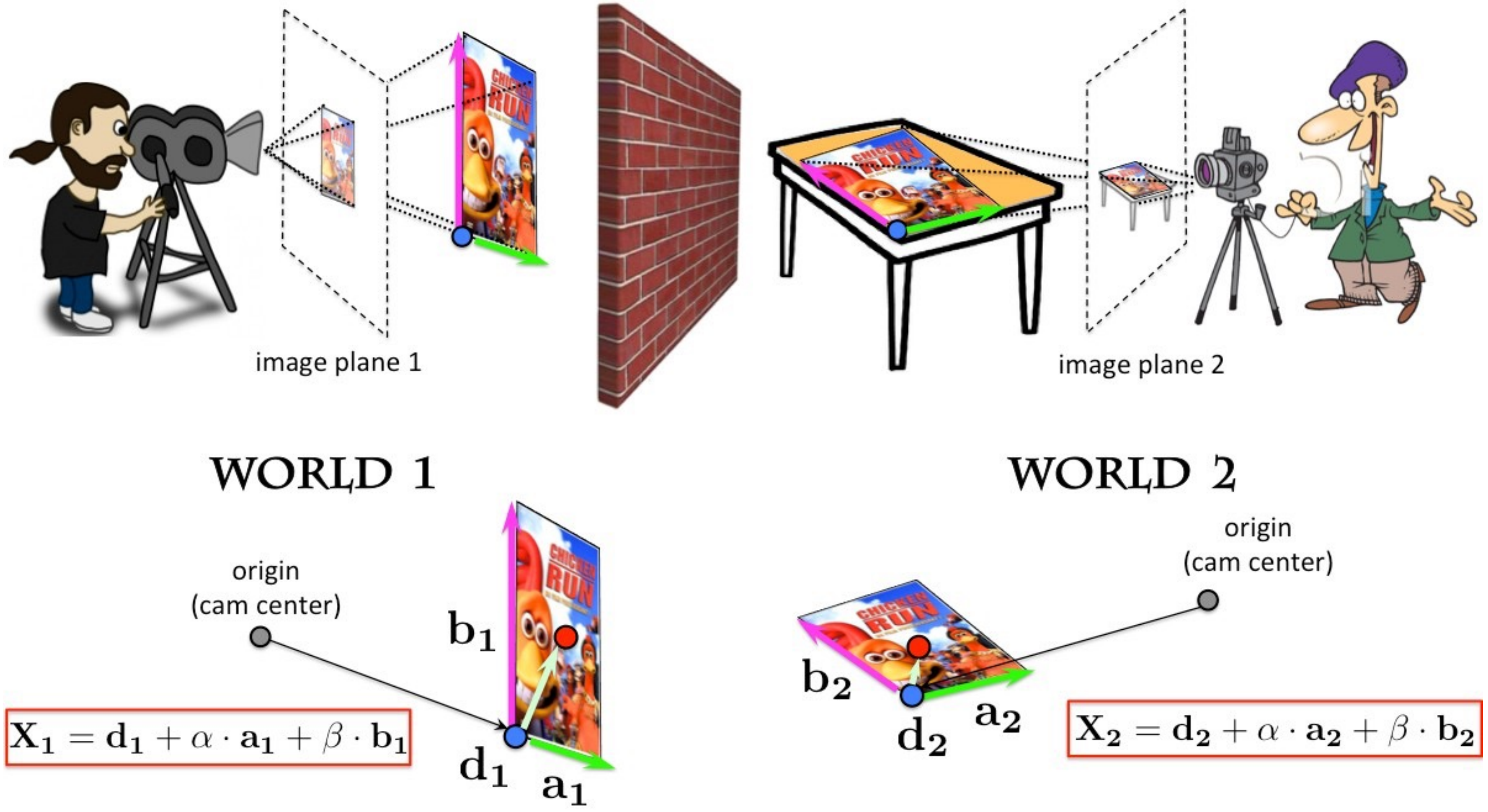
- Let's revisit our transformation in the (new) light of perspective projection.



Our object is a plane. Each plane is characterized by one point d on the plane and two independent vectors a and b on the plane.

Homography

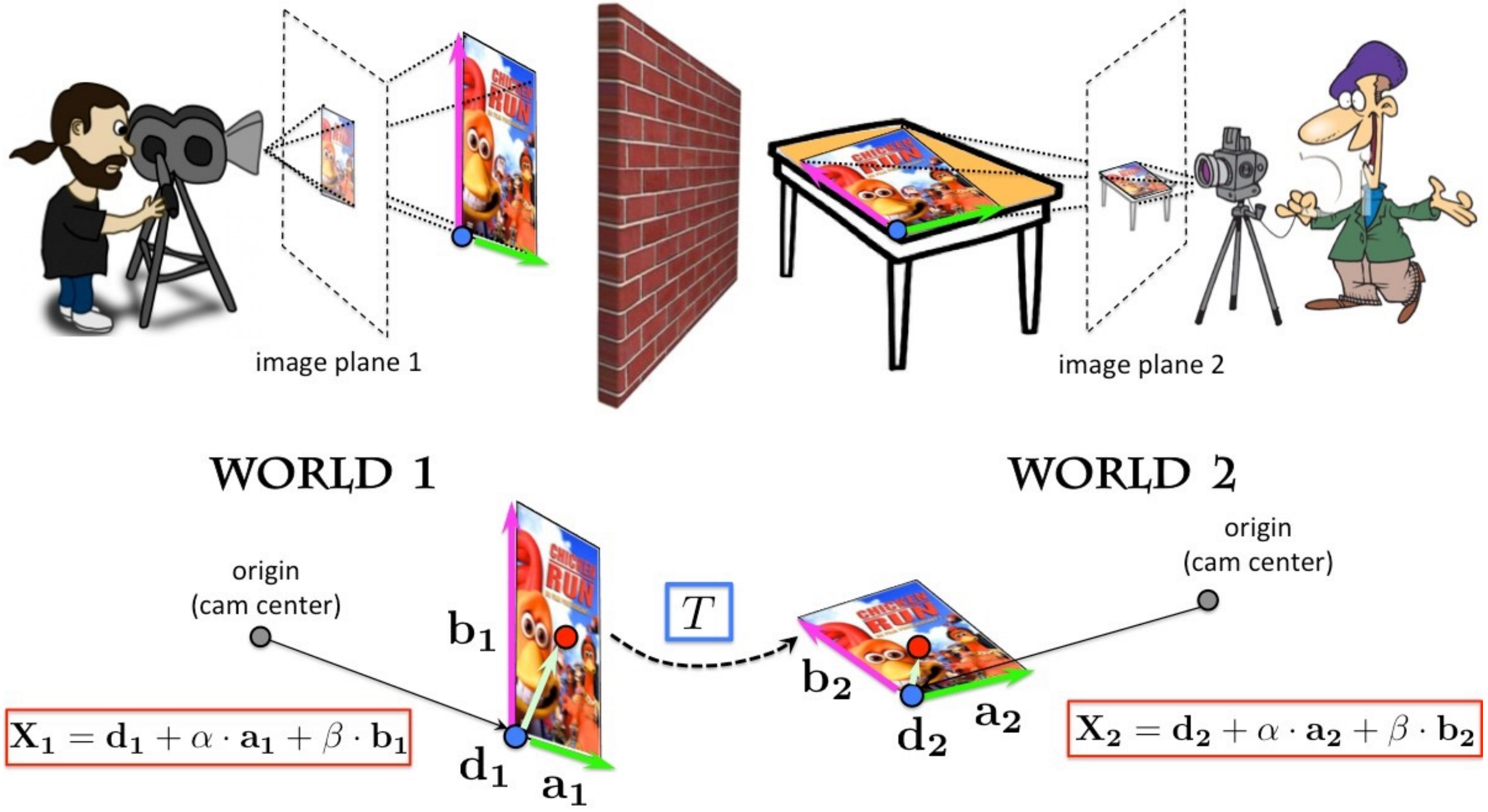
- Let's revisit our transformation in the (new) light of perspective projection.



Then any other point X on the plane can be written as: $X = d + \alpha a + \beta b$; where α and β are in the DVD's coordinate system defined by its basis vectors and origin.

Homography

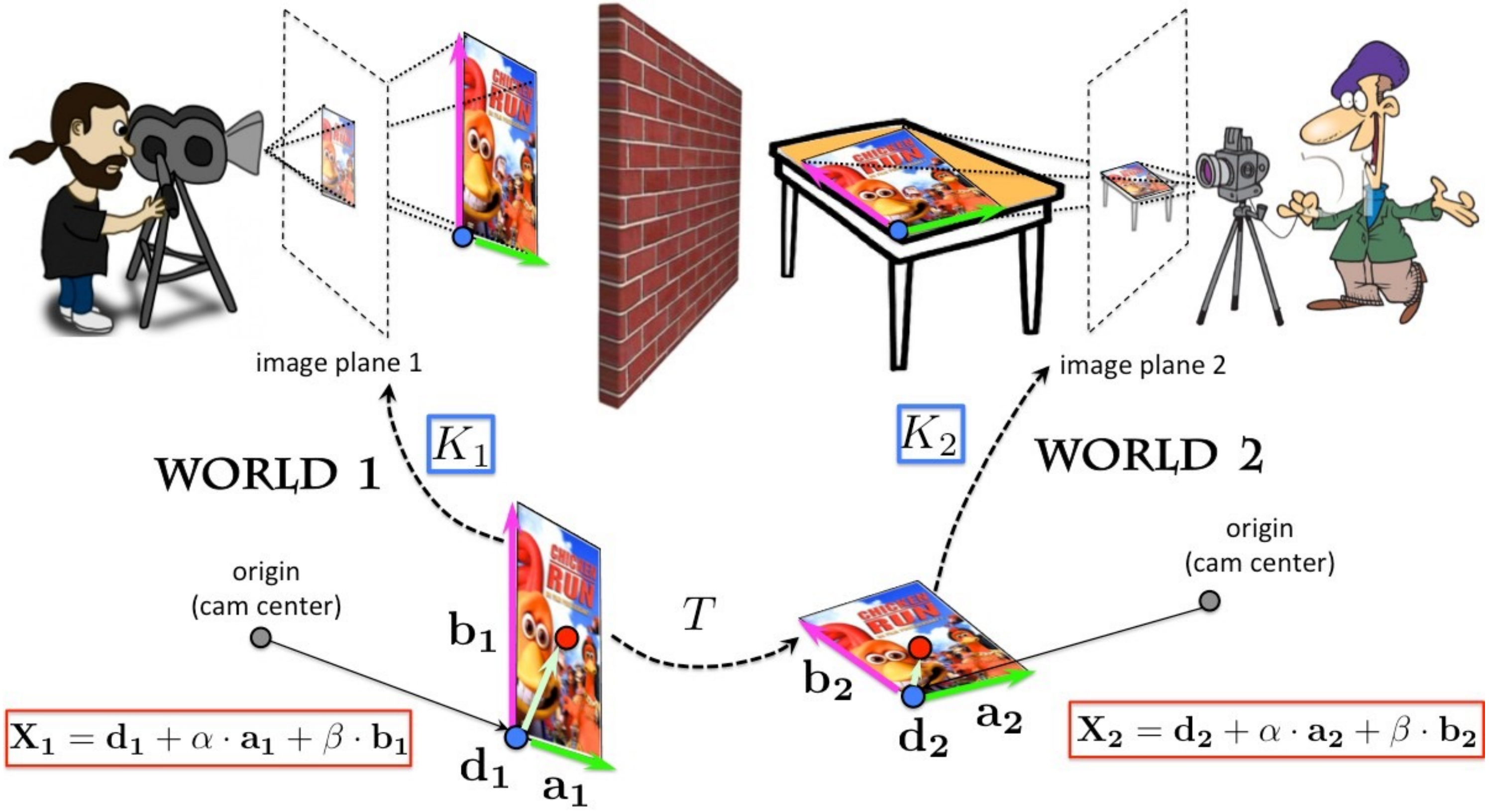
- Let's revisit our transformation in the (new) light of perspective projection.



Any two Chicken Run DVDs on our planet are related by some transformation T. We'll compute it, don't worry.

Homography

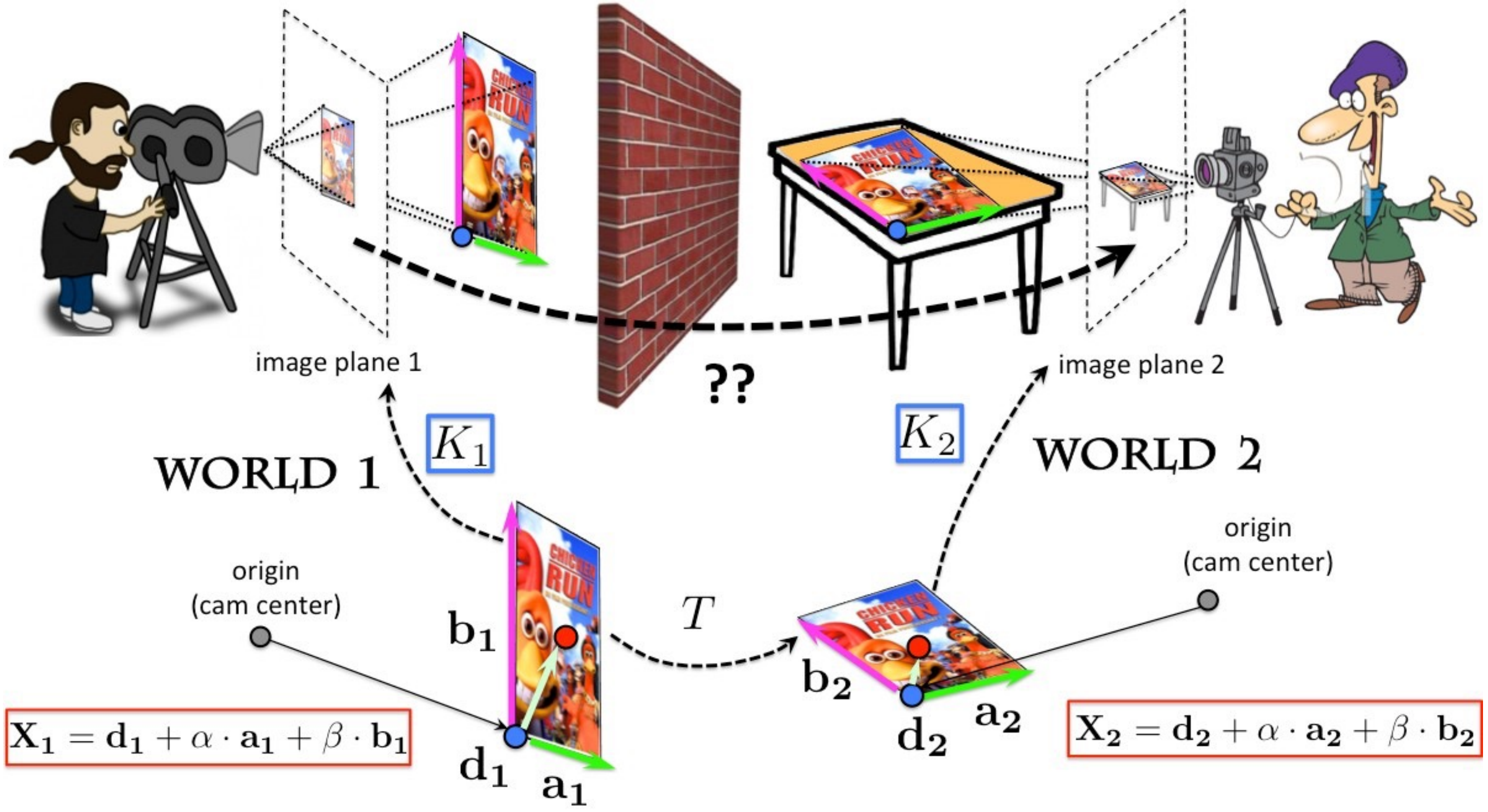
- Let's revisit our transformation in the (new) light of perspective projection.



Each object is seen by a different camera and thus projects to the corresponding image plane with different camera intrinsics.

Homography

- Let's revisit our transformation in the (new) light of perspective projection.



Given this, the question is what's the transformation that maps the DVD on the first image to the DVD in the second image?

Homography

- Each point on a plane can be written as: $X = d + \alpha \cdot a + \beta \cdot b$, where d is a point, and a and b are two independent directions on the plane.

Homography

- Each point on a plane can be written as: $X = d + \alpha \cdot a + \beta \cdot b$, where d is a point, and a and b are two independent directions on the plane.
- Let's have two different planes in 3D:

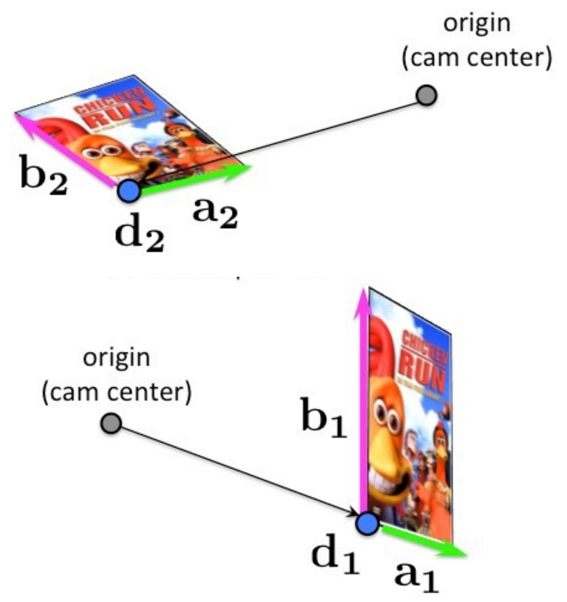
$$\text{First plane}: \; \mathbf{X}_1 = \mathbf{d}_1 + \alpha \cdot \mathbf{a}_1 + \beta \cdot \mathbf{b}_1$$
$$\text{Second plane}: \; \mathbf{X}_2 = \mathbf{d}_2 + \alpha \cdot \mathbf{a}_2 + \beta \cdot \mathbf{b}_2$$

Homography

- Each point on a plane can be written as: $X = d + \alpha \cdot a + \beta \cdot b$, where d is a point, and a and b are two independent directions on the plane.
- Let's have two different planes in 3D:

$$\text{First plane}: \ \mathbf{X}_1 = \mathbf{d}_1 + \alpha \cdot \mathbf{a}_1 + \beta \cdot \mathbf{b}_1$$

$$\text{Second plane}: \ \mathbf{X}_2 = \mathbf{d}_2 + \alpha \cdot \mathbf{a}_2 + \beta \cdot \mathbf{b}_2$$



Homography

- Each point on a plane can be written as: $X = d + \alpha \cdot a + \beta \cdot b$, where d is a point, and a and b are two independent directions on the plane.
- Let's have two different planes in 3D:

$$\text{First plane}: \mathbf{X}_1 = \mathbf{d}_1 + \alpha \cdot \mathbf{a}_1 + \beta \cdot \mathbf{b}_1$$
$$\text{Second plane}: \mathbf{X}_2 = \mathbf{d}_2 + \alpha \cdot \mathbf{a}_2 + \beta \cdot \mathbf{b}_2$$

- Via α and β, the two points X1 and X2 are in the same location relative to each plane (correspondences!)

Homography

- Each point on a plane can be written as: $X = d + \alpha \cdot a + \beta \cdot b$, where d is a point, and a and b are two independent directions on the plane.
- Let's have two different planes in 3D:

$$\text{First plane}: \ \mathbf{X}_1 = \mathbf{d}_1 + \alpha \cdot \mathbf{a}_1 + \beta \cdot \mathbf{b}_1$$
$$\text{Second plane}: \ \mathbf{X}_2 = \mathbf{d}_2 + \alpha \cdot \mathbf{a}_2 + \beta \cdot \mathbf{b}_2$$

- Via α and β, the two points X1 and X2 are in the same location relative to each plane.
- We can rewrite this using homogeneous coordinates:

$$\text{First plane}: \ \mathbf{X}_1 = \begin{bmatrix} \mathbf{a}_1 & \mathbf{b}_1 & \mathbf{d}_1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ 1 \end{bmatrix} = A_1 \begin{bmatrix} \alpha \\ \beta \\ 1 \end{bmatrix}$$

Homography

- Each point on a plane can be written as: $X = d + \alpha \cdot a + \beta \cdot b$, where d is a point, and a and b are two independent directions on the plane.
- Let's have two different planes in 3D:

$$\text{First plane}: \mathbf{X}_1 = \mathbf{d}_1 + \alpha \cdot \mathbf{a}_1 + \beta \cdot \mathbf{b}_1$$
$$\text{Second plane}: \mathbf{X}_2 = \mathbf{d}_2 + \alpha \cdot \mathbf{a}_2 + \beta \cdot \mathbf{b}_2$$

- Via α and β, the two points X1 and X2 are in the same location relative to each plane.
- We can rewrite this using homogeneous coordinates:

$$\text{First plane}: \mathbf{X}_1 = \begin{bmatrix}\mathbf{a}_1 & \mathbf{b}_1 & \mathbf{d}_1\end{bmatrix}\begin{bmatrix}\alpha \\ \beta \\ 1\end{bmatrix} = A_1\begin{bmatrix}\alpha \\ \beta \\ 1\end{bmatrix}$$
$$\text{Second plane}: \mathbf{X}_2 = \begin{bmatrix}\mathbf{a}_2 & \mathbf{b}_2 & \mathbf{d}_2\end{bmatrix}\begin{bmatrix}\alpha \\ \beta \\ 1\end{bmatrix} = A_2\begin{bmatrix}\alpha \\ \beta \\ 1\end{bmatrix}$$

Homography

- Each point on a plane can be written as: $X = d + \alpha \cdot a + \beta \cdot b$, where d is a point, and a and b are two independent directions on the plane.
- Let's have two different planes in 3D:

$$\text{First plane}: \ \mathbf{X}_1 = \mathbf{d}_1 + \alpha \cdot \mathbf{a}_1 + \beta \cdot \mathbf{b}_1$$
$$\text{Second plane}: \ \mathbf{X}_2 = \mathbf{d}_2 + \alpha \cdot \mathbf{a}_2 + \beta \cdot \mathbf{b}_2$$

- Via α and β, the two points X1 and X2 are in the same location relative to each plane.
- We can rewrite this using homogeneous coordinates:

$$\text{First plane}: \ \mathbf{X}_1 = \begin{bmatrix}\mathbf{a}_1 & \mathbf{b}_1 & \mathbf{d}_1\end{bmatrix}\begin{bmatrix}\alpha \\ \beta \\ 1\end{bmatrix} = A_1\begin{bmatrix}\alpha \\ \beta \\ 1\end{bmatrix}$$
$$\text{Second plane}: \ \mathbf{X}_2 = \begin{bmatrix}\mathbf{a}_2 & \mathbf{b}_2 & \mathbf{d}_2\end{bmatrix}\begin{bmatrix}\alpha \\ \beta \\ 1\end{bmatrix} = A_2\begin{bmatrix}\alpha \\ \beta \\ 1\end{bmatrix}$$

- $A_1 = [a_1 \quad b_1 \quad d_1]$ and $A_2 = [a_2 \quad b_2 \quad d_2]$ are 3 x 3 matrices.

Homography

- In 3D, a transformation between the planes is given by:

$$X_2 = T\,X_1$$

There is one transformation T between every pair of points X_1 and X_2.

Homography

- In 3D, a transformation between the planes is given by:

$$\mathsf{X}_2 = T\,\mathsf{X}_1$$

There is one transformation T between every pair of points X_1 and X_2.

- Expand it:

$$A_2 \begin{bmatrix} \alpha \\ \beta \\ 1 \end{bmatrix} = T\,A_1 \begin{bmatrix} \alpha \\ \beta \\ 1 \end{bmatrix} \qquad \text{for every } \alpha, \beta$$

Homography

- In 3D, a transformation between the planes is given by:

$$\mathsf{X}_2 = T\,\mathsf{X}_1$$

There is one transformation T between every pair of points X_1 and X_2.

- Expand it:

$$A_2 \begin{bmatrix} \alpha \\ \beta \\ 1 \end{bmatrix} = T\,A_1 \begin{bmatrix} \alpha \\ \beta \\ 1 \end{bmatrix} \qquad \text{for every } \alpha, \beta$$

- Then it follows: $T = A_2 A_1^{-1}$, with T a 3×3 matrix.

Homography

- Let's look at what happens in projective (image) plane. Note that we have each plane in a separate image and the two images may not have the same camera intrinsic parameters. Denote them with K_1 and K_2.

$$w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = K_1 \mathsf{X}_1 \qquad \text{and} \qquad w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K_2 \mathsf{X}_2$$

Homography

- Let's look at what happens in projective (image) plane. Note that we have each plane in a separate image and the two images may not have the same camera intrinsic parameters. Denote them with K_1 and K_2.

$$w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = K_1 \mathbf{X}_1 \quad \text{and} \quad w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K_2 \mathbf{X}_2$$

- Insert $X_2 = T X_1$ into the equality on the right

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K_2 T \mathbf{X}_1$$

Homography

- Let's look at what happens in projective (image) plane. Note that we have each plane in a separate image and the two images may not have the same camera intrinsic parameters. Denote them with K_1 and K_2.

$$w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = K_1 \mathsf{X}_1 \qquad \text{and} \qquad w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K_2 \mathsf{X}_2$$

- Insert X_2 = T X into the equality on the right

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K_2 T \mathsf{X}_1 = K_2 T \,(K_1^{-1} K_1) \mathsf{X}_1$$

Homography

- Let's look at what happens in projective (image) plane. Note that we have each plane in a separate image and the two images may not have the same camera intrinsic parameters. Denote them with K_1 and K_2.

$$w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = K_1 \mathbf{X}_1 \qquad \text{and} \qquad w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K_2 \mathbf{X}_2$$

- Insert $X_2 = T X$ into the equality on the right

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K_2 T \mathbf{X}_1 = K_2 T (K_1^{-1} \underbrace{K_1) \mathbf{X}_1}_{w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}}$$

Homography

- Let's look at what happens in projective (image) plane. Note that we have each plane in a separate image and the two images may not have the same camera intrinsic parameters. Denote them with K_1 and K_2.

$$w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = K_1 \mathbf{X}_1 \qquad \text{and} \qquad w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K_2 \mathbf{X}_2$$

- Insert $X_2 = T\,X$ into the equality on the right

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K_2 \, T \, \mathbf{X}_1 = K_2 \, T \, (K_1^{-1} K_1) \mathbf{X}_1 = w_1 K_2 \, T \, K_1^{-1} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Homography

- Let's look at what happens in projective (image) plane. Note that we have each plane in a separate image and the two images may not have the same camera intrinsic parameters. Denote them with K_1 and K_2.

$$w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = K_1 \mathbf{X}_1 \qquad \text{and} \qquad w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K_2 \mathbf{X}_2$$

- Insert $X_2 = T\,X$ into the equality on the right

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K_2\, T\, \mathbf{X}_1 = K_2\, T\, (K_1^{-1} K_1) \mathbf{X}_1 = w_1 \underbrace{K_2\, T\, K_1^{-1}}_{3\times 3 \text{ matrix}} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Homography

- Let's look at what happens in projective (image) plane. Note that we have each plane in a separate image and the two images may not have the same camera intrinsic parameters. Denote them with K_1 and K_2.

$$w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = K_1 \mathbf{X}_1 \qquad \text{and} \qquad w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K_2 \mathbf{X}_2$$

- Insert $X_2 = T\,X$ into the equality on the right

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K_2 T \mathbf{X}_1 = K_2 T (K_1^{-1} K_1) \mathbf{X}_1 = w_1 \underbrace{K_2 T K_1^{-1}}_{3\times 3 \text{ matrix}} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

- Finally, divide through by w_1

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Homography

- Let's look at what happens in projective (image) plane. Note that we have each plane in a separate image and the two images may not have the same camera intrinsic parameters. Denote them with K_1 and K_2.

$$w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = K_1 \mathbf{X}_1 \quad \text{and} \quad w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K_2 \mathbf{X}_2$$

- Insert $X_2 = T\,X$ into the equality on the right

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K_2 T \mathbf{X}_1 = K_2 T (K_1^{-1} K_1) \mathbf{X}_1 = w_1 \underbrace{K_2 T K_1^{-1}}_{3\times3 \text{ matrix}} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

- Finally, divide through by w_1

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

what is this?

Homography

- If we want to compute correspondences between images and we have the homography, what else do we need?
  - 3D positions?
  - Camera intrinsics?

Homography

- If we want to compute correspondences between images and we have the homography, what else do we need?
  - 3D positions?
  - Camera intrinsics?

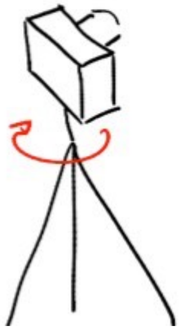
- Still one more loose end from lecture 8 to recap...

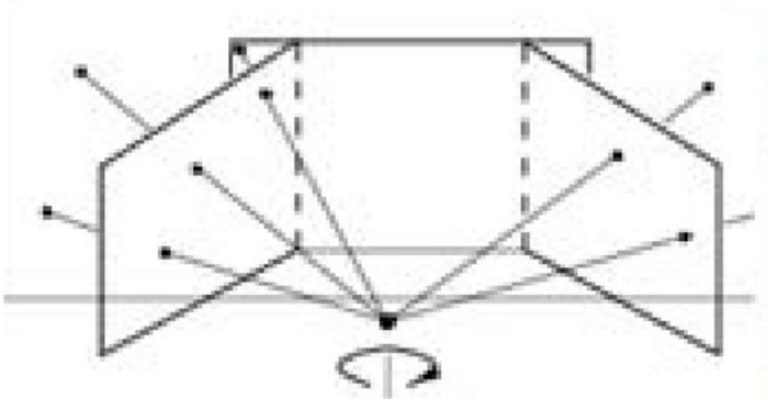
Remember Panorama Stitching from Lecture 9?

Take a tripod, rotate camera and take pictures

[Source: Fernando Flores-Mangas]

Remember Panorama Stitching from Lecture 9?





- Each pair of images is related by homography. Why?

[Source: Fernando Flores-Mangas]

Rotating the Camera

- Rotating my camera with R is the same as rotating the 3D points with R^T (inverse of R):

$$\mathbf{X}_2 = R^T \mathbf{X}_1$$

- where X_1 is a 3D point in the coordinate system of the first camera and X_2 the 3D point in the coordinate system of the rotated camera.

Rotating the Camera

- Rotating my camera with R is the same as rotating the 3D points with R^T (inverse of R):

$$\mathbf{X}_2 = R^T \mathbf{X}_1$$

- where X_1 is a 3D point in the coordinate system of the first camera and X_2 the 3D point in the coordinate system of the rotated camera.
- We can use the same trick as before, where we have $T = R^T$:

$$w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = K\mathbf{X}_1 \qquad \text{and} \qquad w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K\mathbf{X}_2$$

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = w_1 \underbrace{K\, R^T\, K^{-1}}_{3\times 3 \text{ matrix}} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Rotating the Camera

- Rotating my camera with R is the same as rotating the 3D points with R^T (inverse of R):

$$\mathbf{X}_2 = R^T \mathbf{X}_1$$

- where X_1 is a 3D point in the coordinate system of the first camera and X_2 the 3D point in the coordinate system of the rotated camera.
- We can use the same trick as before, where we have $T = R^T$:

$$w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = K\mathbf{X}_1 \quad \text{and} \quad w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K\mathbf{X}_2$$

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = w_1 \underbrace{K\, R^T\, K^{-1}}_{3\times 3 \text{ matrix}} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

what is this?

What If I Move The Camera?

- So if I take a picture, rotate the camera, and take a second picture…

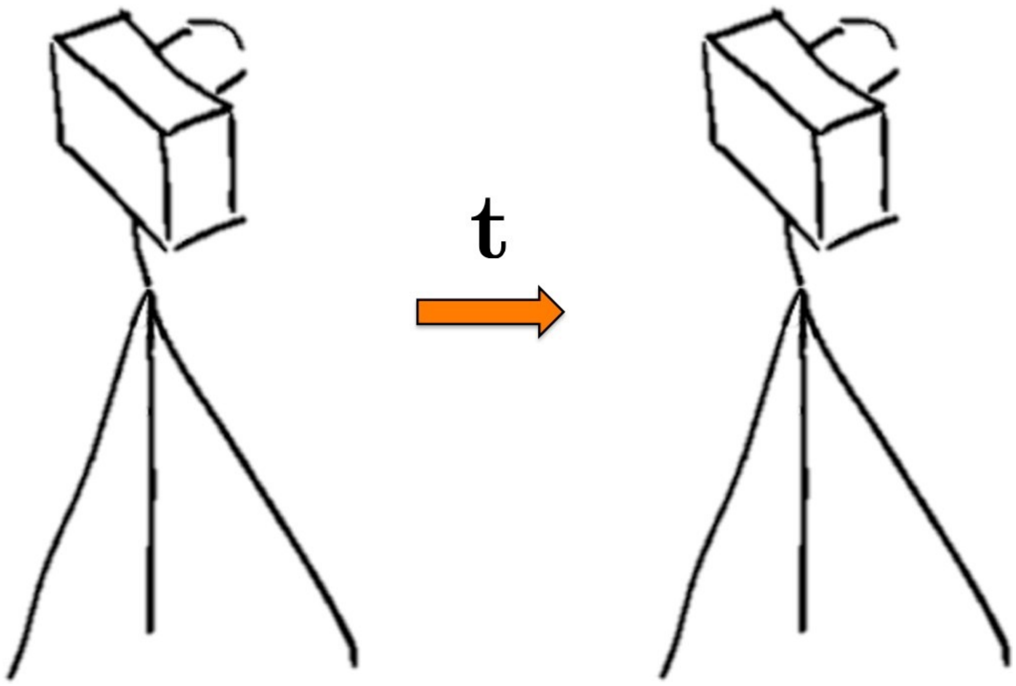
- How are the first and second images related?

What If I Move The Camera?

- So if I take a picture, rotate the camera, and take a second picture…
- How are the first and second images related?
- by a Homography (assuming the scene didn’t change)

What If I Move The Camera?

- So if I take a picture, rotate the camera, and take a second picture…
- How are the first and second images related?
- by a Homography (assuming the scene didn't change)
- What if I move the camera?



What If I Move The Camera?

- If I move the camera by t, then: $X_2 = X_1 - t$. Let's try the same trick again:

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K\, \mathbf{X_2}$$

What If I Move The Camera?

- If I move the camera by t, then: $X_2 = X_1 - t$. Let's try the same trick again:

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K\,\mathbf{X}_2 = K\,(\mathbf{X}_1 - \mathbf{t})$$

What If I Move The Camera?

- If I move the camera by t, then: $X_2 = X_1 - t$. Let's try the same trick again:

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K\,\mathbf{X_2} = K\,(\underbrace{\mathbf{X_1}}_{w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}} - \mathbf{t})$$

What If I Move The Camera?

- If I move the camera by t, then: $X_2 = X_1 - t$. Let's try the same trick again:

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K\,\mathbf{X}_2 = K\,(\mathbf{X}_1 - \mathbf{t}) = w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} - K\mathbf{t}$$

- What's the problem here?

What If I Move The Camera?

- If I move the camera by t, then: $X_2 = X_1 - t$. Let's try the same trick again:

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K\,\mathbf{X}_2 = K\,(\mathbf{X}_1 - \mathbf{t}) = w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} - K\mathbf{t}$$

- Now, different values of w_1 give different points in the second image!

What If I Move The Camera?

- If I move the camera by t, then: $X_2 = X_1 - t$. Let's try the same trick again:

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K\,\mathbf{X}_2 = K\,(\mathbf{X}_1 - \mathbf{t}) = w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} - K\mathbf{t}$$

- Now, different values of w_1 give different points in the second image!
- So, even if I have K and **t** I can't compute where a point from the first image projects to in the second image.

What If I Move The Camera?

- If I move the camera by t, then: $X_2 = X_1 - t$. Let's try the same trick again:

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K\,\mathbf{X}_2 = K\,(\mathbf{X}_1 - \mathbf{t}) = w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} - K\mathbf{t}$$

- Now, different values of w_1 give different points in the second image!
- So, even if I have K and **t** I can't compute where a point from the first image projects to in the second image.
- From

$$w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = K\mathbf{X}_1$$

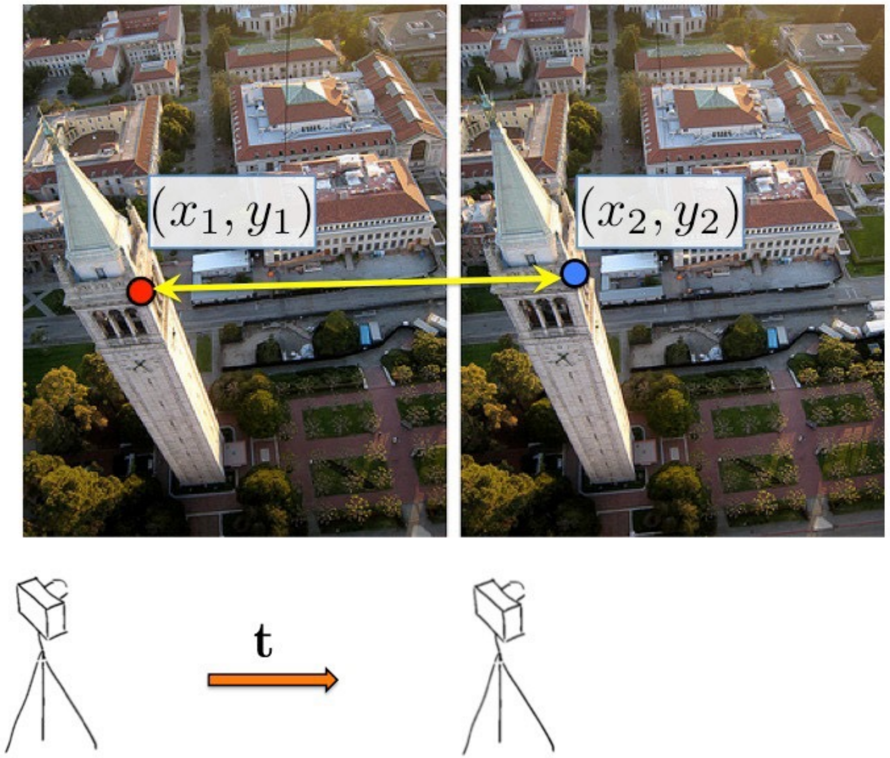
  we know that different w_1 map to different points X_1 on the projective line
- Where (x_1, y_1) maps to in the 2nd image depends on the 3D location of X_1

What If I Move The Camera?

- **Summary**: if I move the camera, I can’t easily map one image to the other. The mapping depends on the 3D scene behind the image.

What If I Move The Camera?

- **Summary**: if I move the camera, I can't easily map one image to the other. The mapping depends on the 3D scene behind the image.
- What about the opposite, what if I know that points (x_1, y_1) in the first image and (x_2, y_2) in the second belong to the same 3D point?

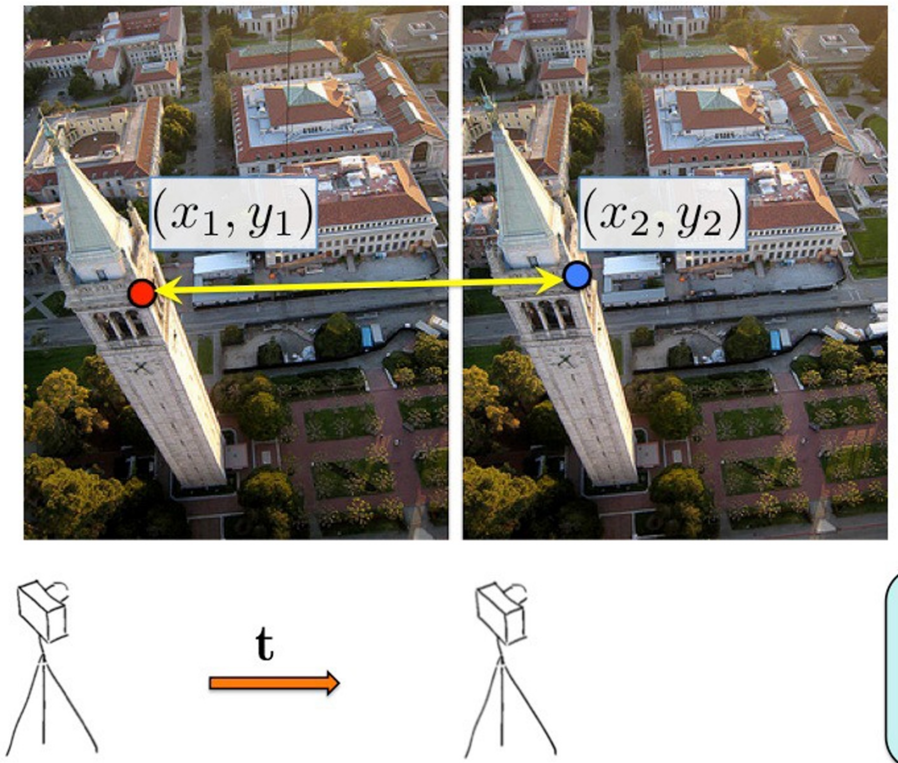


$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} - K\mathbf{t}$$

We know this

What If I Move The Camera?

- **Summary**: if I move the camera, I can't easily map one image to the other. The mapping depends on the 3D scene behind the image.
- What about the opposite, what if I know that points (x_1, y_1) in the first image and (x_2, y_2) in the second belong to the same 3D point?



$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} - K\mathbf{t}$$

We know this

➡ We can compute w_1 and w_2

➡ We can compute point in 3D!

What If I Move The Camera?

- **Summary**: if I move the camera, I can't easily map one image to the other. The mapping depends on the 3D scene behind the image.
- What about the opposite, what if I know that points (x_1, y_1) in the first image and (x_2, y_2) in the second belong to the same 3D point?
- This allows triangulating 3D points, leads to **stereo** vision and **two-view** geometry

Summary – Stuff You Need To Know

Perspective Projection

- If point Q is in camera's coordinate system:
  - $\mathbf{Q} = (X, Y, Z)^T \quad \rightarrow \quad \mathbf{q} = \left(\frac{f \cdot X}{Z} + p_x, \frac{f \cdot Y}{Z} + p_y\right)^T$

Summary – Stuff You Need To Know

Perspective Projection

- If point Q is in camera's coordinate system:
  - $\mathbf{Q} = (X, Y, Z)^T \quad \rightarrow \quad \mathbf{q} = \left(\frac{f \cdot X}{Z} + p_x, \frac{f \cdot Y}{Z} + p_y\right)^T$
  - Same as: $\mathbf{Q} = (X, Y, Z)^T \quad \rightarrow \quad \begin{bmatrix} w \cdot x \\ w \cdot y \\ w \end{bmatrix} = K \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \rightarrow \quad \mathbf{q} = \begin{bmatrix} x \\ y \end{bmatrix}$

Summary – Stuff You Need To Know

Perspective Projection

- If point Q is in camera's coordinate system:
  - $\mathbf{Q} = (X, Y, Z)^T \quad \rightarrow \quad \mathbf{q} = \left(\frac{f \cdot X}{Z} + p_x, \frac{f \cdot Y}{Z} + p_y\right)^T$
  - Same as: $\mathbf{Q} = (X, Y, Z)^T \quad \rightarrow \quad \begin{bmatrix} w \cdot x \\ w \cdot y \\ w \end{bmatrix} = K \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \rightarrow \quad \mathbf{q} = \begin{bmatrix} x \\ y \end{bmatrix}$

    where $K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}$ is camera intrinsic matrix

Summary – Stuff You Need To Know

Perspective Projection

- If point Q is in camera's coordinate system:
  - $\mathbf{Q} = (X, Y, Z)^T \quad \rightarrow \quad \mathbf{q} = \left(\frac{f \cdot X}{Z} + p_x, \frac{f \cdot Y}{Z} + p_y\right)^T$
  - Same as: $\mathbf{Q} = (X, Y, Z)^T \quad \rightarrow \quad \begin{bmatrix} w \cdot x \\ w \cdot y \\ w \end{bmatrix} = K \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \rightarrow \quad \mathbf{q} = \begin{bmatrix} x \\ y \end{bmatrix}$

  where $K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}$ is camera intrinsic matrix

- If Q is in world coordinate system, then the full projection is characterized by a 3x4 matrix P:

$$\begin{bmatrix} w \cdot x \\ w \cdot y \\ w \end{bmatrix} = \underbrace{\mathbf{K}[\mathbf{R} \mid \mathbf{t}]}_{\mathbf{P}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Summary – Stuff You Need To Know

Perspective Projection

- All parallel lines in 3D with the same direction meet in one, so-called vanishing point in the image
- All lines that lie on a plane have vanishing points that lie on a line, so-called vanishing lines.
- All parallel planes in 3D have the same vanishing line in the image

Summary – Stuff You Need To Know

Perspective Projection

- All parallel lines in 3D with the same direction meet in one, so-called vanishing point in the image
- All lines that lie on a plane have vanishing points that lie on a line, so-called vanishing lines.
- All parallel planes in 3D have the same vanishing line in the image

Orthographic Projection

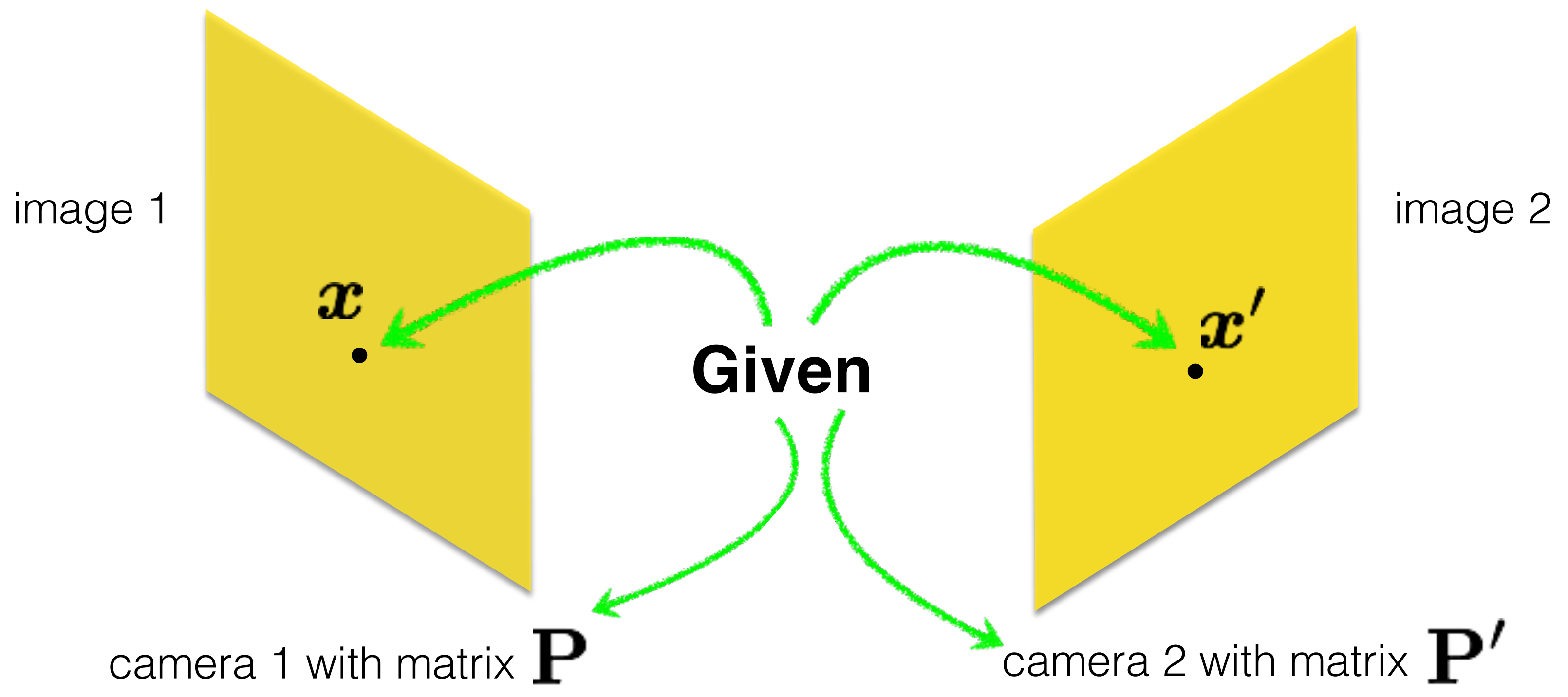
- Projections simply drops the Z coordinate:

$$\mathbf{Q} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

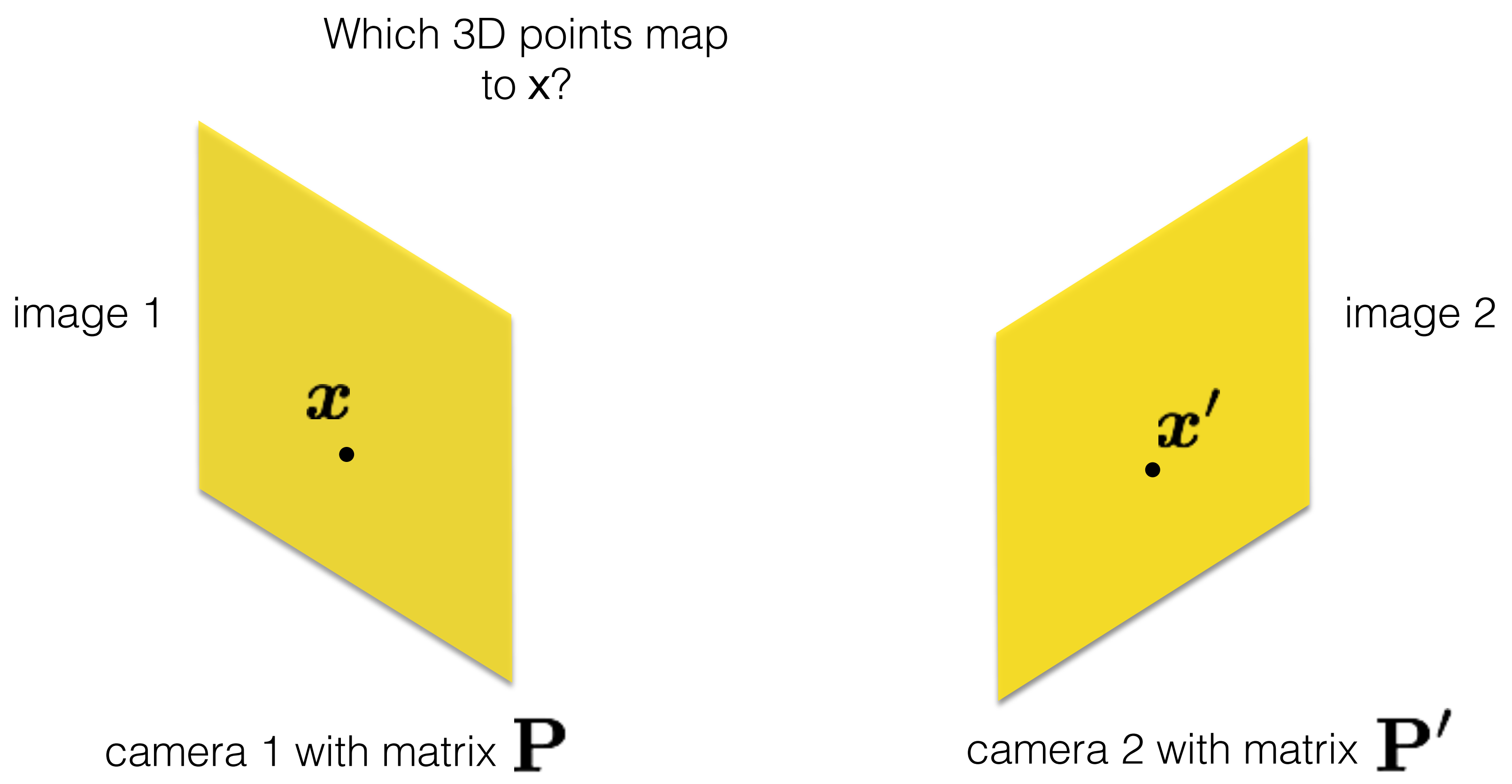
- Parallel lines in 3D are parallel in the image

Two-view Geometry

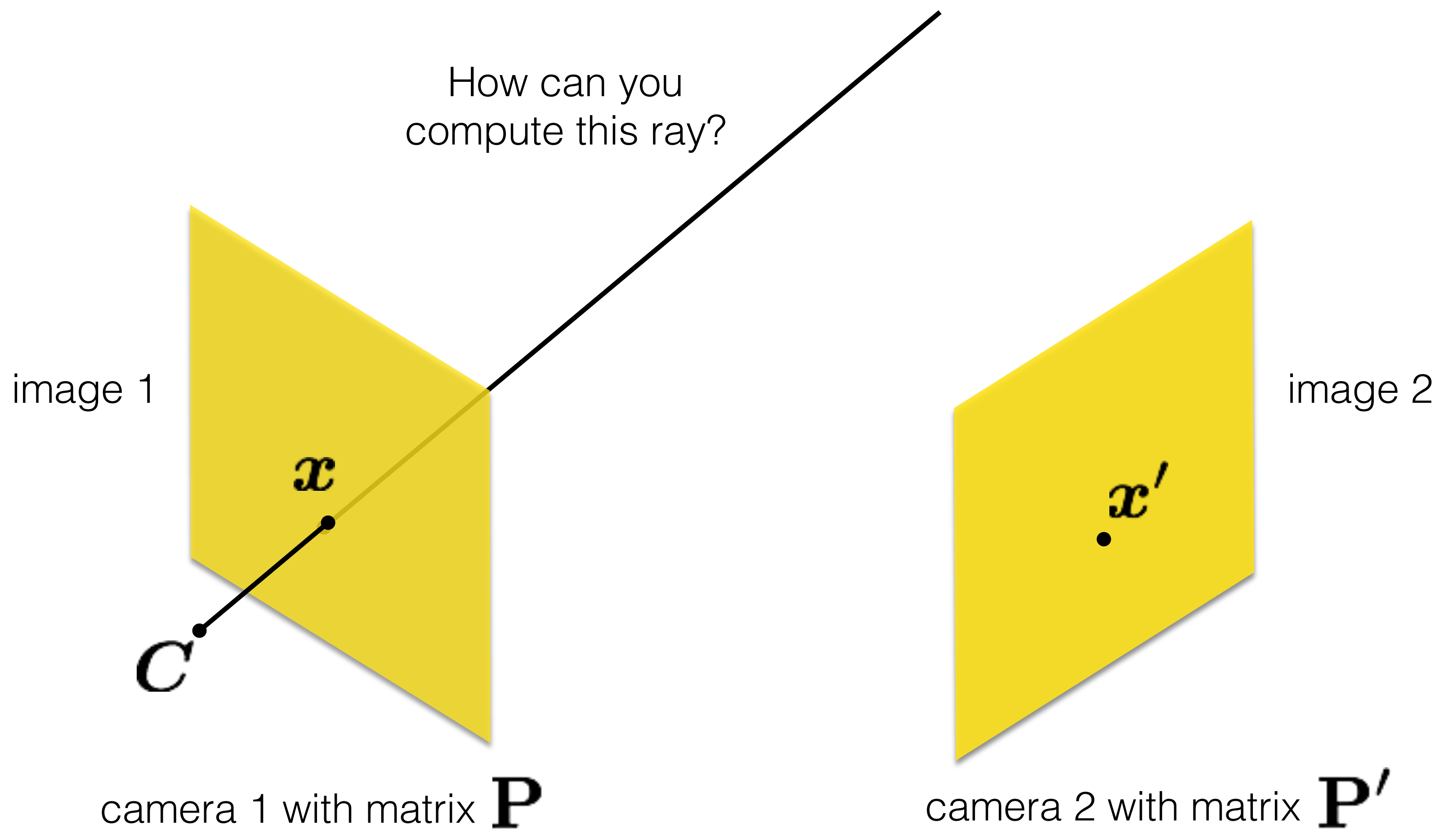
Triangulation



Triangulation



Triangulation



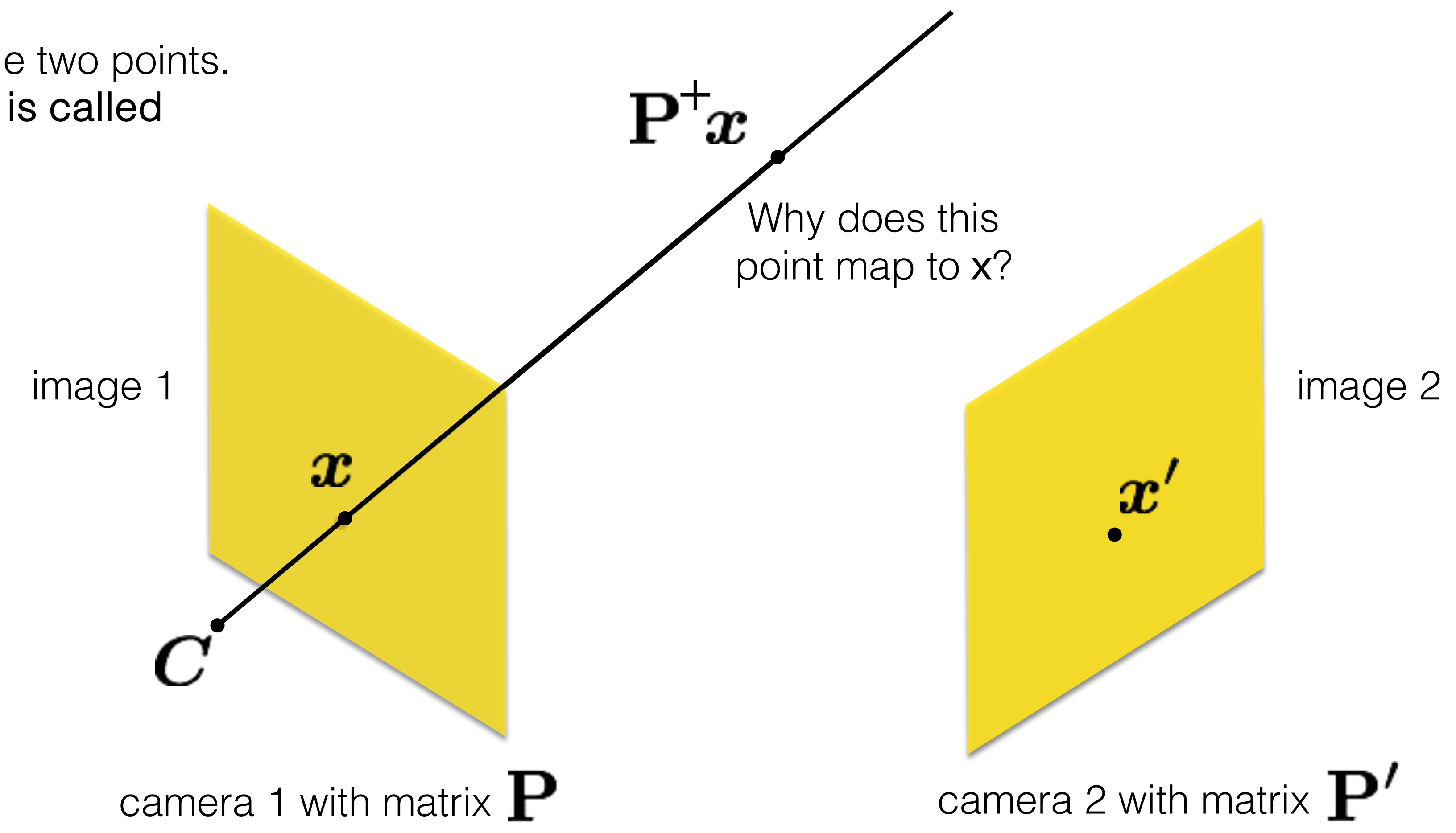
Triangulation

Create two points on the ray:

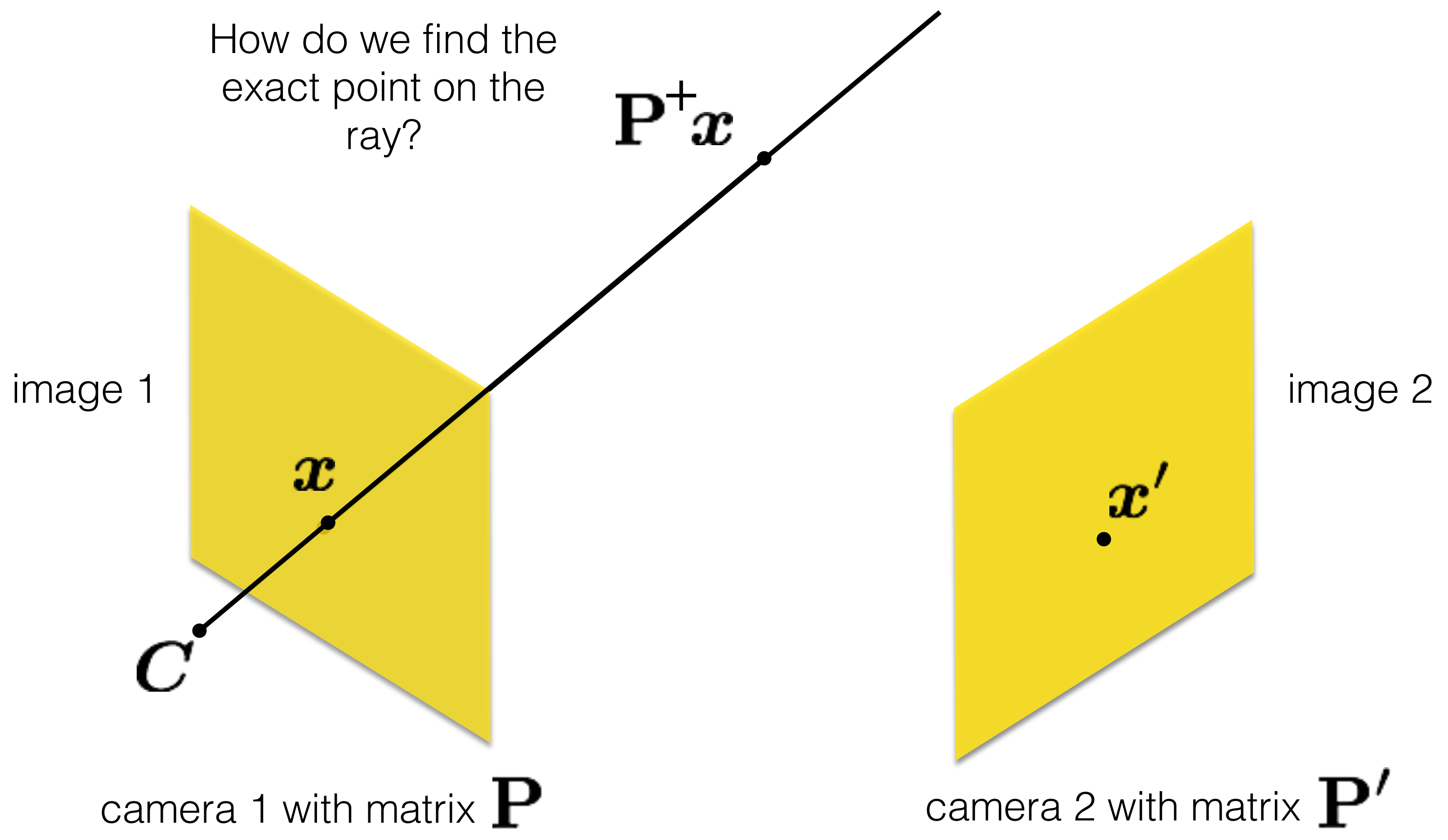
1) find the camera center; and
2) apply the pseudo-inverse of P on x.

Then connect the two points.

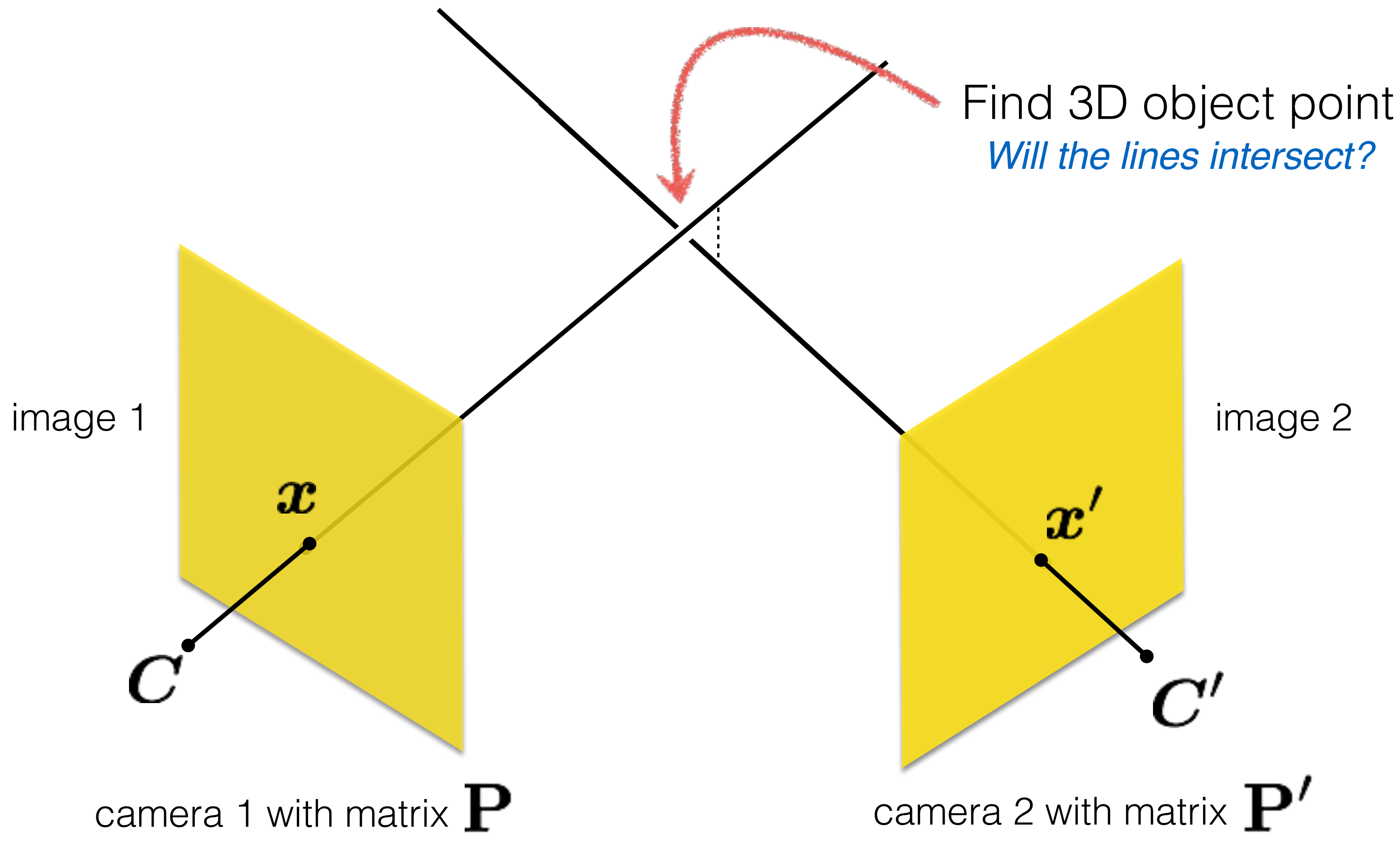
This procedure is called <u>backprojection</u>



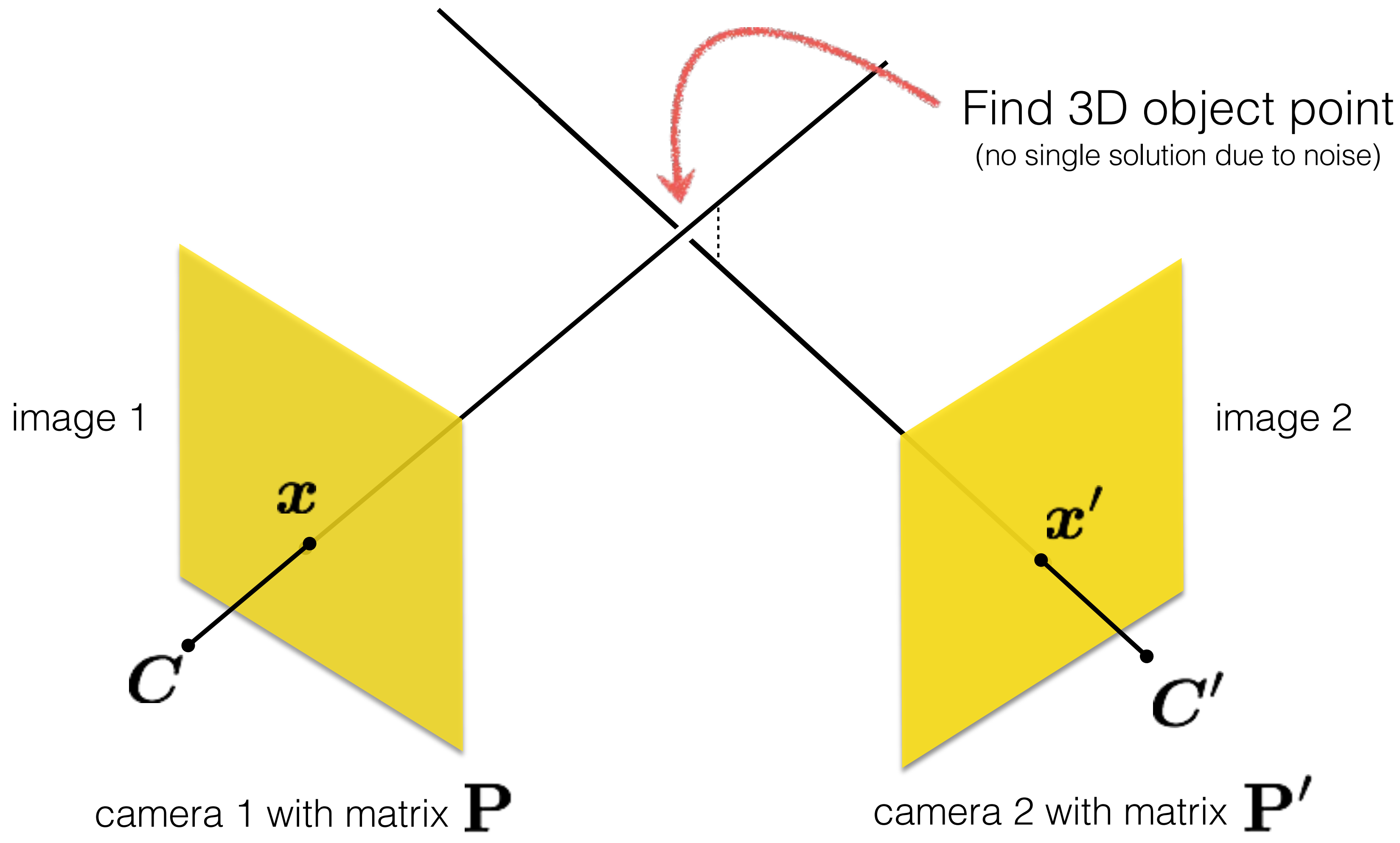
Triangulation



Triangulation



Triangulation



Triangulation

Given a set of (noisy) matched points

$$\{\boldsymbol{x}_i, \boldsymbol{x}'_i\}$$

and camera matrices

$$\mathbf{P}, \mathbf{P}'$$

Estimate the 3D point

$$\mathbf{X}$$

$$\mathbf{x} = \mathbf{P}\boldsymbol{X}$$

known known

*Can we compute **X** from a single correspondence **x**?*

$$\mathbf{x} = \mathbf{P}\boldsymbol{X}$$

(homogeneous coordinate)

This is a similarity relation because it involves homogeneous coordinates

$$\mathbf{x} = \alpha\mathbf{P}\boldsymbol{X}$$

(homorogeneous coordinate)

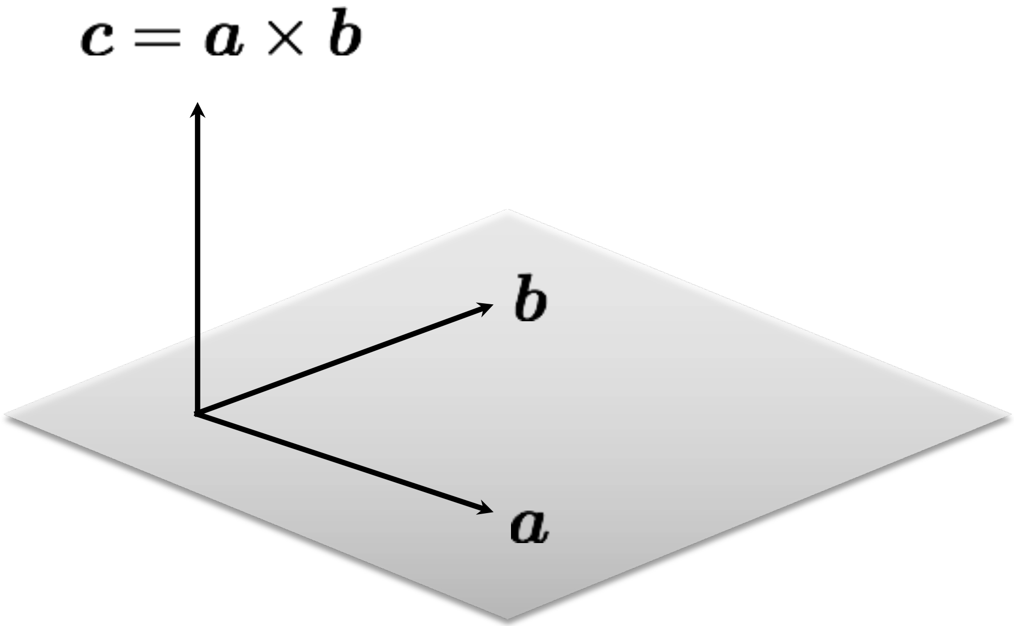
Same ray direction but differs by a scale factor

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Linear algebra reminder: cross product

Vector (cross) product

takes two vectors and returns a vector perpendicular to both



$$\boldsymbol{a} \times \boldsymbol{b} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}$$

cross product of two vectors in the same direction is zero vector

$$\boldsymbol{a} \times \boldsymbol{a} = 0$$

remember this!!!

$$\boldsymbol{c} \cdot \boldsymbol{a} = 0 \qquad \boldsymbol{c} \cdot \boldsymbol{b} = 0$$

Linear algebra reminder: cross product

Cross product

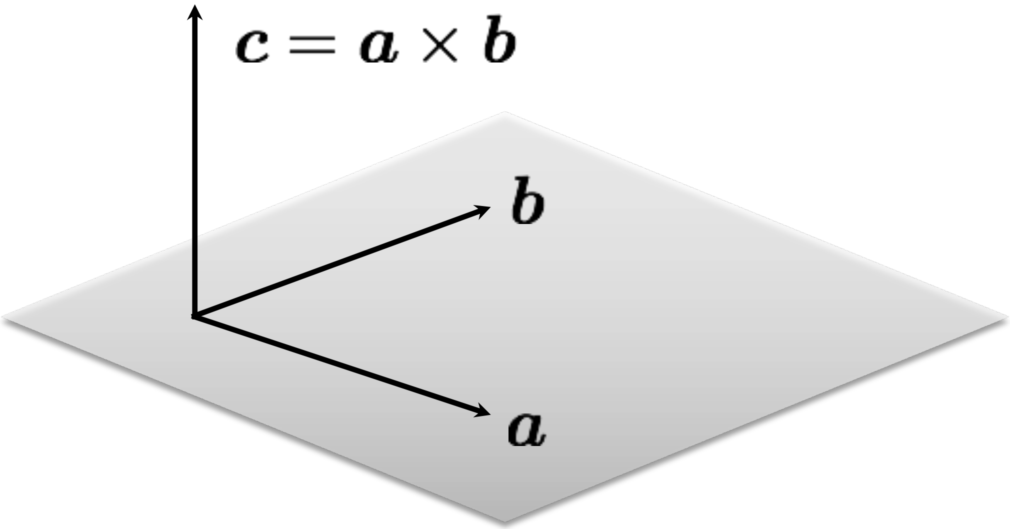
$$\boldsymbol{a} \times \boldsymbol{b} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}$$

Can also be written as a matrix multiplication

$$\boldsymbol{a} \times \boldsymbol{b} = [\boldsymbol{a}]_\times \boldsymbol{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Skew symmetric

Compare with: dot product



$c \cdot a = 0$ $\qquad$ $c \cdot b = 0$

dot product of two orthogonal vectors is (scalar) zero

Back to triangulation

$$\mathbf{x} = \alpha \mathbf{P} \boldsymbol{X}$$

Same direction but differs by a scale factor

How can we rewrite this using vector products?

$$\mathbf{x} = \alpha \mathbf{P} \boldsymbol{X}$$

Same direction but differs by a scale factor

$$\mathbf{x} \times \mathbf{P} \boldsymbol{X} = \mathbf{0}$$

Cross product of two vectors of same direction is zero
(this equality removes the scale factor)

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha \begin{bmatrix} \text{---} & \boldsymbol{p}_1^\top & \text{---} \\ \text{---} & \boldsymbol{p}_2^\top & \text{---} \\ \text{---} & \boldsymbol{p}_3^\top & \text{---} \end{bmatrix} \begin{bmatrix} | \\ \boldsymbol{X} \\ | \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha \begin{bmatrix} \text{---} & \boldsymbol{p}_1^\top & \text{---} \\ \text{---} & \boldsymbol{p}_2^\top & \text{---} \\ \text{---} & \boldsymbol{p}_3^\top & \text{---} \end{bmatrix} \begin{bmatrix} | \\ \boldsymbol{X} \\ | \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha \begin{bmatrix} \boldsymbol{p}_1^\top \boldsymbol{X} \\ \boldsymbol{p}_2^\top \boldsymbol{X} \\ \boldsymbol{p}_3^\top \boldsymbol{X} \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha \begin{bmatrix} \text{——} & \boldsymbol{p}_1^\top & \text{——} \\ \text{——} & \boldsymbol{p}_2^\top & \text{——} \\ \text{——} & \boldsymbol{p}_3^\top & \text{——} \end{bmatrix} \begin{bmatrix} | \\ \boldsymbol{X} \\ | \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha \begin{bmatrix} \boldsymbol{p}_1^\top \boldsymbol{X} \\ \boldsymbol{p}_2^\top \boldsymbol{X} \\ \boldsymbol{p}_3^\top \boldsymbol{X} \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \times \begin{bmatrix} \boldsymbol{p}_1^\top \boldsymbol{X} \\ \boldsymbol{p}_2^\top \boldsymbol{X} \\ \boldsymbol{p}_3^\top \boldsymbol{X} \end{bmatrix} = \begin{bmatrix} y\boldsymbol{p}_3^\top \boldsymbol{X} - \boldsymbol{p}_2^\top \boldsymbol{X} \\ \boldsymbol{p}_1^\top \boldsymbol{X} - x\boldsymbol{p}_3^\top \boldsymbol{X} \\ x\boldsymbol{p}_2^\top \boldsymbol{X} - y\boldsymbol{p}_1^\top \boldsymbol{X} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Using the fact that the cross product should be zero

$$\mathbf{x} \times \mathbf{P}\boldsymbol{X} = \mathbf{0}$$

$$\begin{bmatrix} y\boldsymbol{p}_3^\top \boldsymbol{X} - \boldsymbol{p}_2^\top \boldsymbol{X} \\ \boldsymbol{p}_1^\top \boldsymbol{X} - x\boldsymbol{p}_3^\top \boldsymbol{X} \\ x\boldsymbol{p}_2^\top \boldsymbol{X} - y\boldsymbol{p}_1^\top \boldsymbol{X} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Third line is a linear combination of the first and second lines.
(x times the first line plus y times the second line)

One 2D to 3D point correspondence give you [redacted] equations

Using the fact that the cross product should be zero

$$\mathbf{x} \times \mathbf{P}\boldsymbol{X} = \mathbf{0}$$

$$\begin{bmatrix} y\boldsymbol{p}_3^\top \boldsymbol{X} - \boldsymbol{p}_2^\top \boldsymbol{X} \\ \boldsymbol{p}_1^\top \boldsymbol{X} - x\boldsymbol{p}_3^\top \boldsymbol{X} \\ x\boldsymbol{p}_2^\top \boldsymbol{X} - y\boldsymbol{p}_1^\top \boldsymbol{X} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Third line is a linear combination of the first and second lines.
(x times the first line plus y times the second line)

One 2D to 3D point correspondence give you 2 equations

$$\begin{bmatrix} y\boldsymbol{p}_3^\top \boldsymbol{X} - \boldsymbol{p}_2^\top \boldsymbol{X} \\ \boldsymbol{p}_1^\top \boldsymbol{X} - x\boldsymbol{p}_3^\top \boldsymbol{X} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Remove third row, and rearrange as system of unknowns

$$\begin{bmatrix} y\boldsymbol{p}_3^\top - \boldsymbol{p}_2^\top \\ \boldsymbol{p}_1^\top - x\boldsymbol{p}_3^\top \end{bmatrix} \boldsymbol{X} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\mathbf{A}_i \boldsymbol{X} = \mathbf{0}$$

Now we can make a system of linear equations
(two lines for each 2D point correspondence)

Concatenate the 2D points from both images

Two rows from camera one

Two rows from camera two

$$\begin{bmatrix} y\boldsymbol{p}_3^\top - \boldsymbol{p}_2^\top \\ \boldsymbol{p}_1^\top - x\boldsymbol{p}_3^\top \\ y'\boldsymbol{p}_3'^\top - \boldsymbol{p}_2'^\top \\ \boldsymbol{p}_1'^\top - x'\boldsymbol{p}_3'^\top \end{bmatrix} \boldsymbol{X} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

sanity check! dimensions?

$$\mathbf{A}\boldsymbol{X} = \mathbf{0}$$

How do we solve homogeneous linear system?

Concatenate the 2D points from both images

$$\begin{bmatrix} y\boldsymbol{p}_3^\top - \boldsymbol{p}_2^\top \\ \boldsymbol{p}_1^\top - x\boldsymbol{p}_3^\top \\ y'\boldsymbol{p}_3'^\top - \boldsymbol{p}_2'^\top \\ \boldsymbol{p}_1'^\top - x'\boldsymbol{p}_3'^\top \end{bmatrix} \boldsymbol{X} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{A}\boldsymbol{X} = \mathbf{0}$$

How do we solve homogeneous linear system?

S V D !

Concatenate the 2D points from both images

$$\begin{bmatrix} y\boldsymbol{p}_3^\top - \boldsymbol{p}_2^\top \\ \boldsymbol{p}_1^\top - x\boldsymbol{p}_3^\top \\ y'\boldsymbol{p}_3'^\top - \boldsymbol{p}_2'^\top \\ \boldsymbol{p}_1'^\top - x'\boldsymbol{p}_3'^\top \end{bmatrix} \boldsymbol{X} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{A}\boldsymbol{X} = \mathbf{0}$$

How do we solve homogeneous linear system?

S V D !

This is triangulation!

Triangulation recap

Given a set of (noisy) matched points

$$\{\boldsymbol{x}_i, \boldsymbol{x}'_i\}$$

and camera matrices

$$\mathbf{P}, \mathbf{P}'$$

Estimate the 3D point

$$\mathbf{X}$$

- use relationship $\mathbf{x} \times \mathbf{P}\boldsymbol{X} = \mathbf{0}$

Triangulation recap

Given a set of (noisy) matched points

$$\{\boldsymbol{x}_i, \boldsymbol{x}'_i\}$$

and camera matrices

$$\mathbf{P}, \mathbf{P}'$$

Estimate the 3D point

$$\mathbf{X}$$

- use relationship $\mathbf{x} \times \mathbf{P}\boldsymbol{X} = \mathbf{0}$
- formulate system of equations (2 for each correspondence)

Triangulation recap

Given a set of (noisy) matched points

$$\{\boldsymbol{x}_i, \boldsymbol{x}'_i\}$$

and camera matrices

$$\mathbf{P}, \mathbf{P}'$$

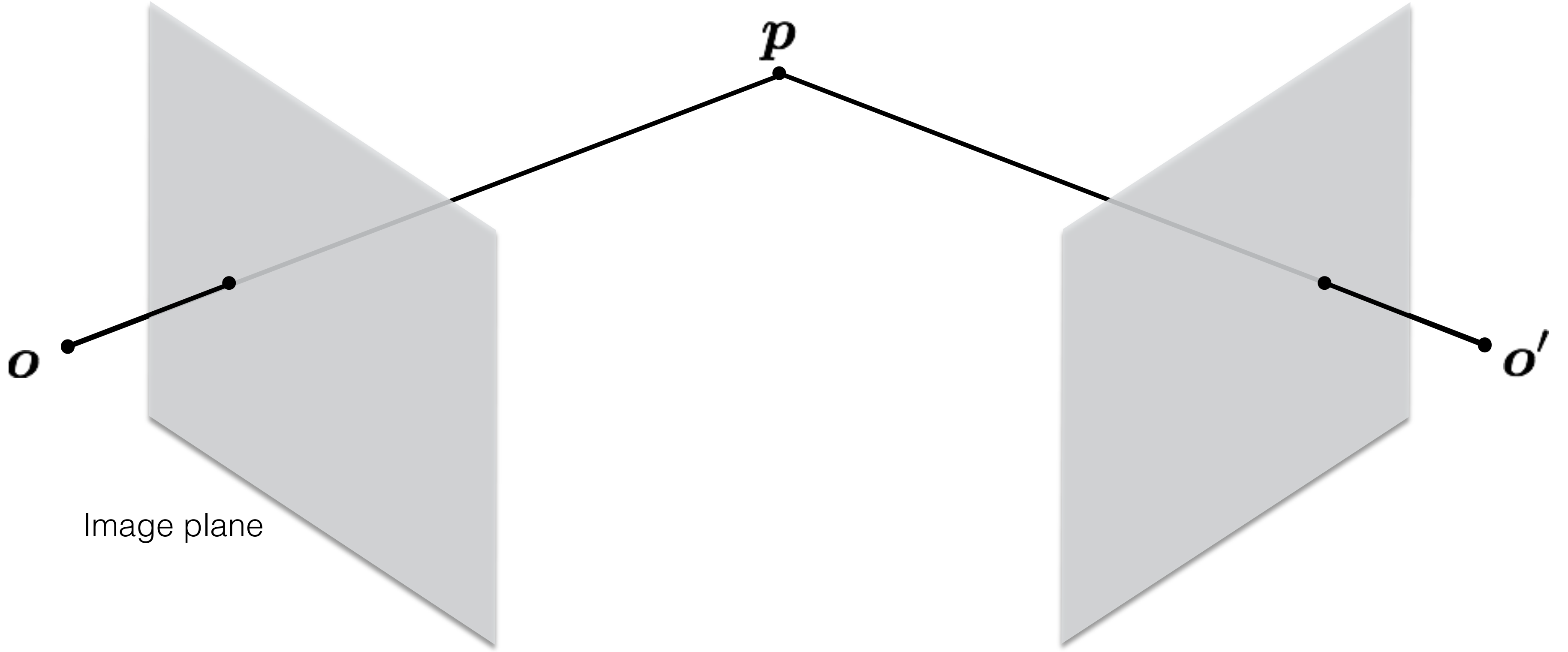
Estimate the 3D point

$$\mathbf{X}$$

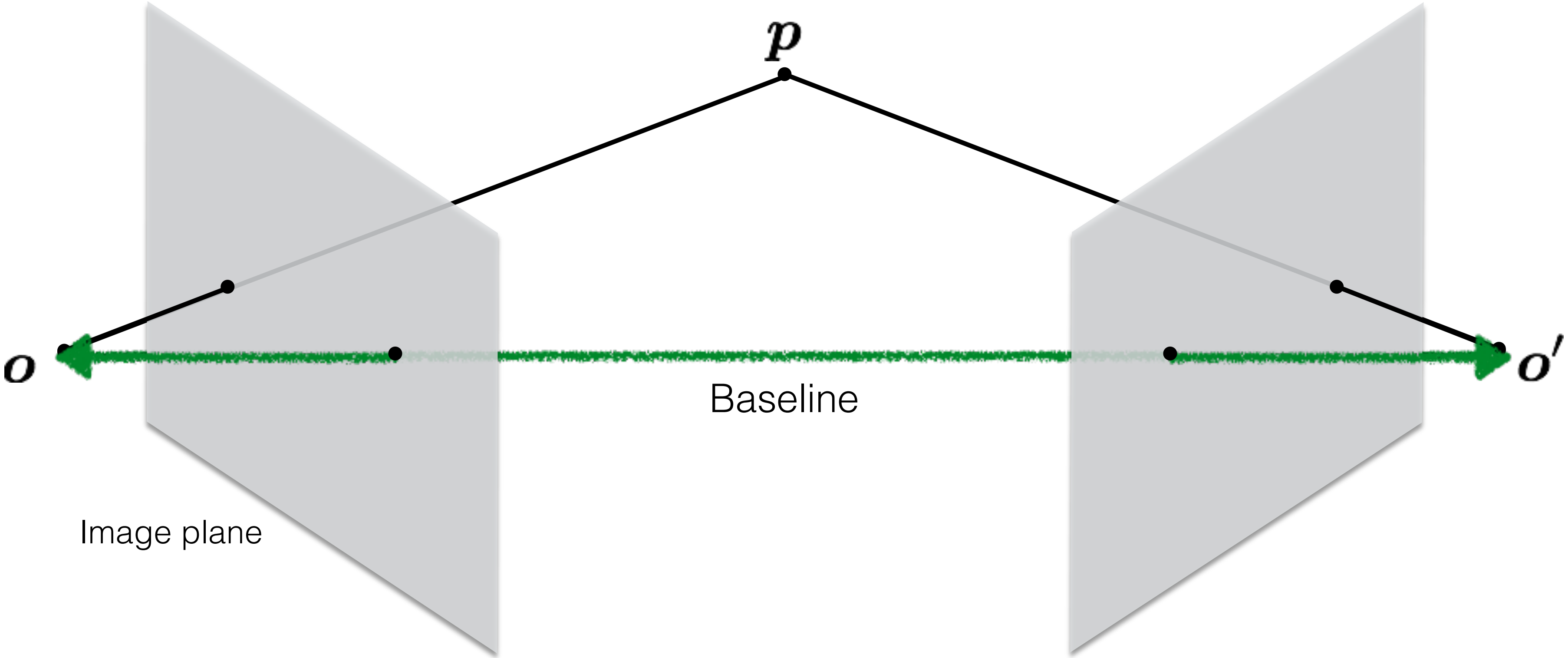
- use relationship $\mathbf{x} \times \mathbf{P}\boldsymbol{X} = \mathbf{0}$
- formulate system of equations (2 for each correspondence)
- Solve with SVD

Epipolar geometry

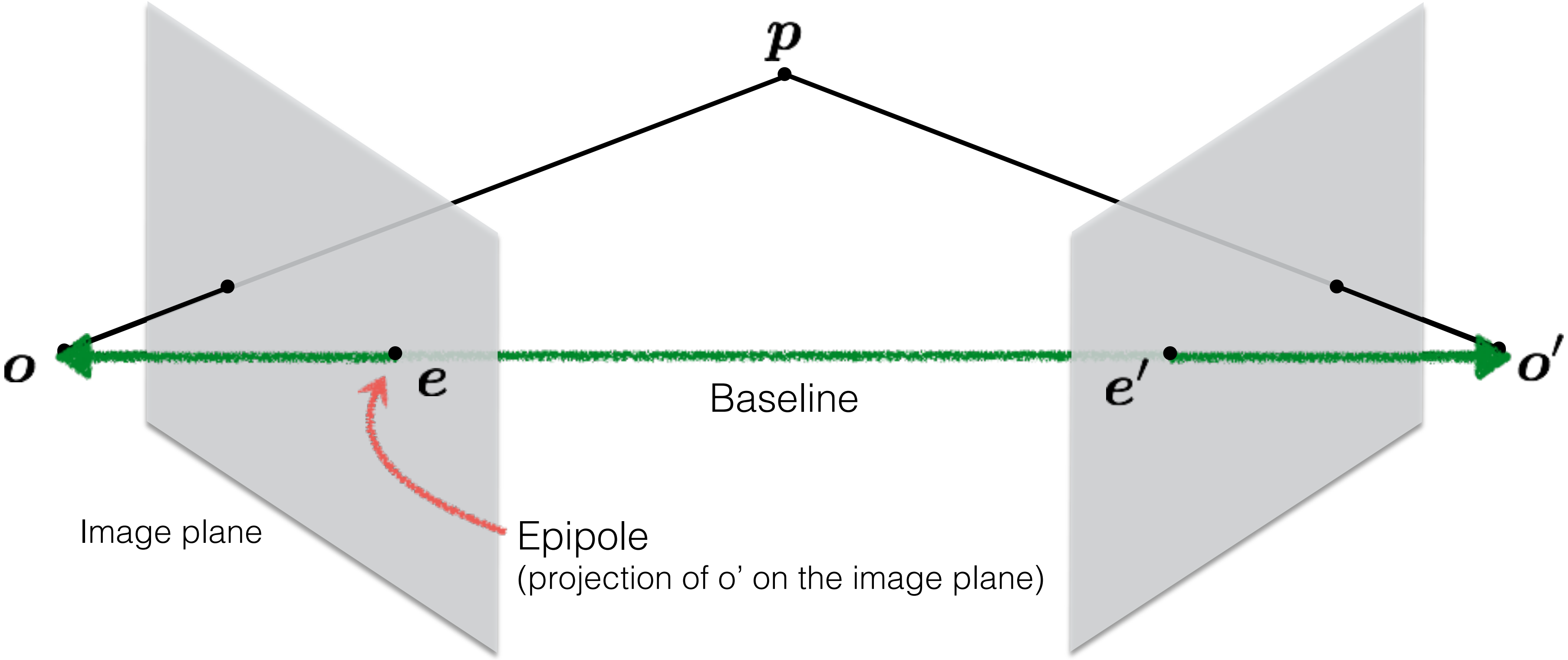
Epipolar geometry



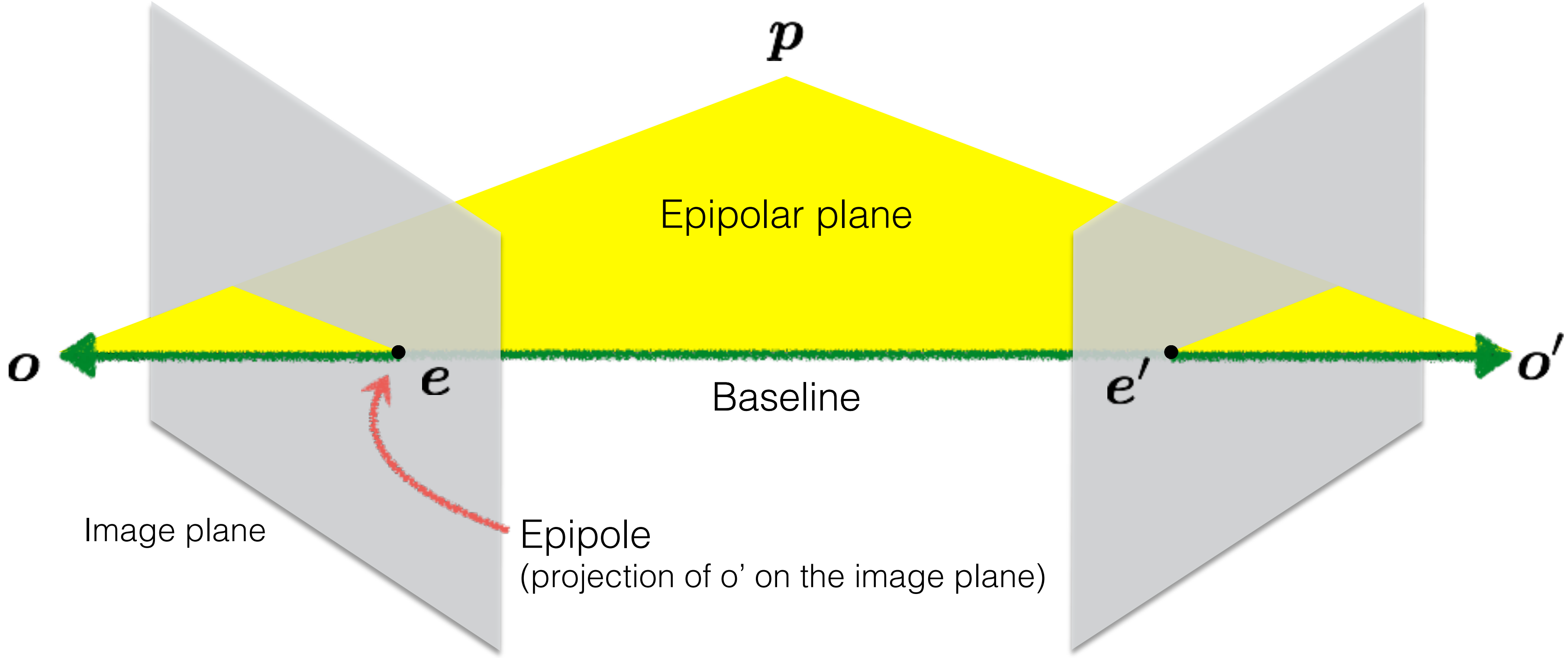
Epipolar geometry



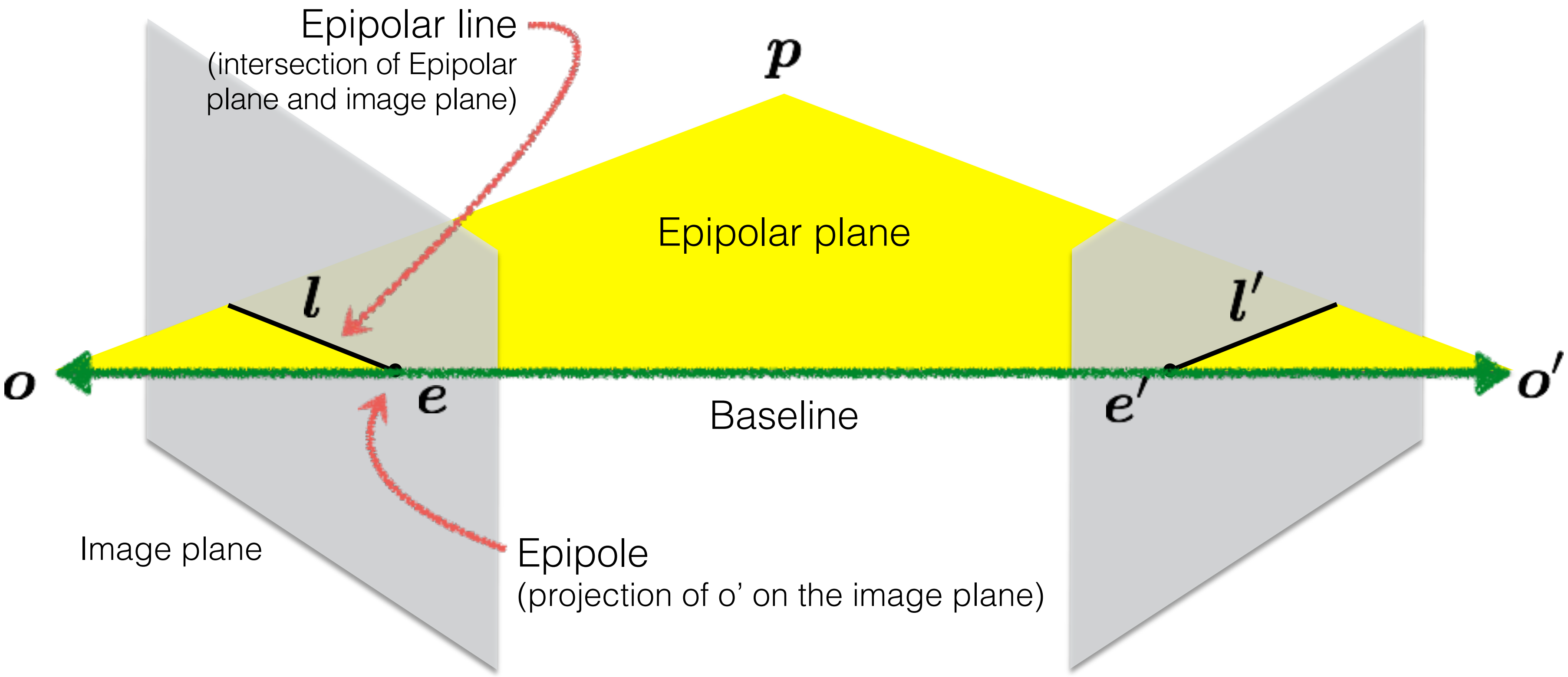
Epipolar geometry



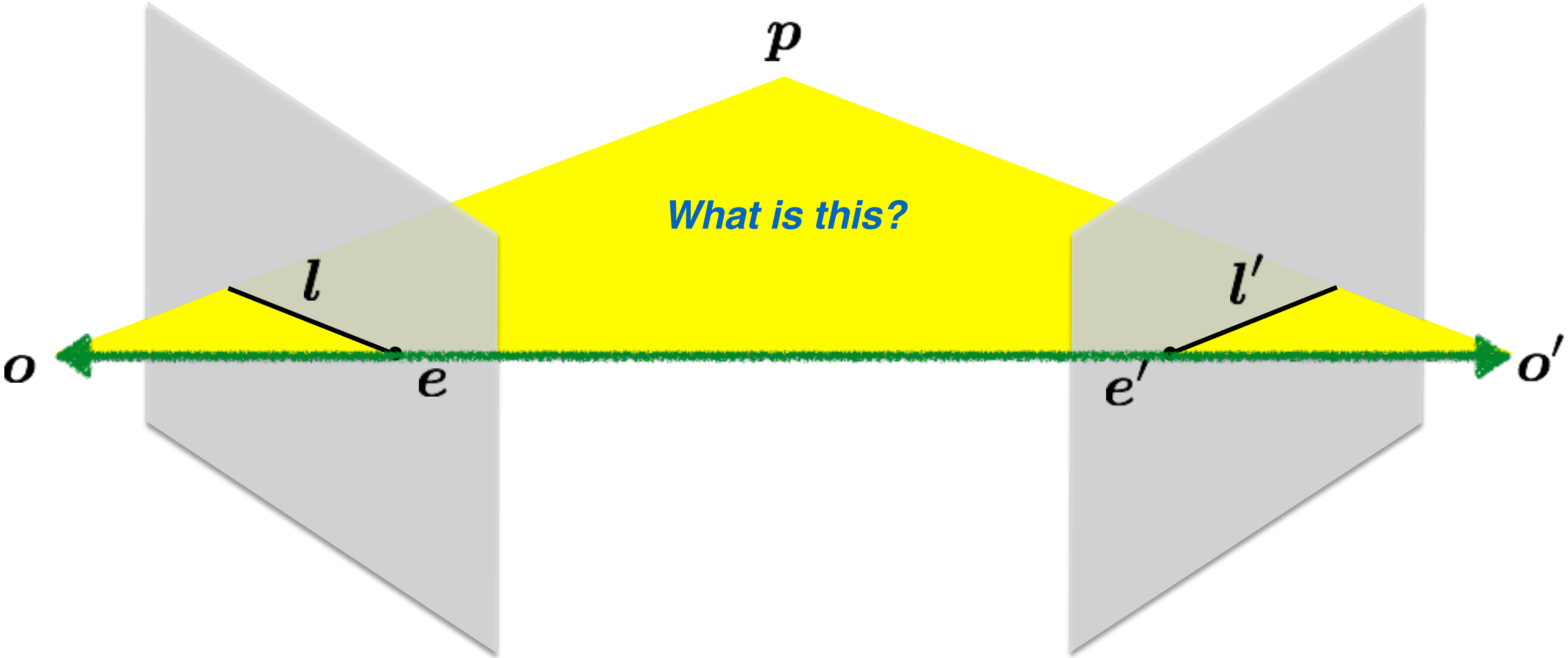
Epipolar geometry



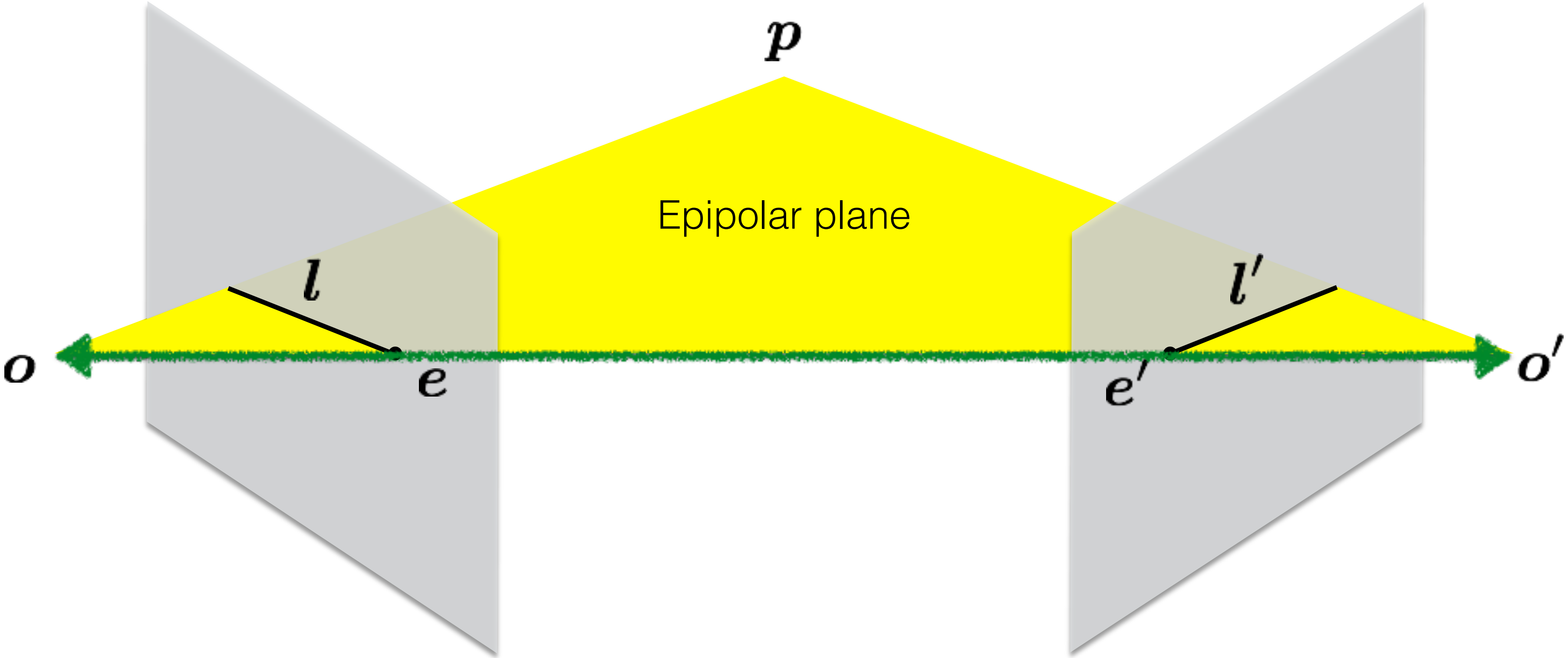
Epipolar geometry



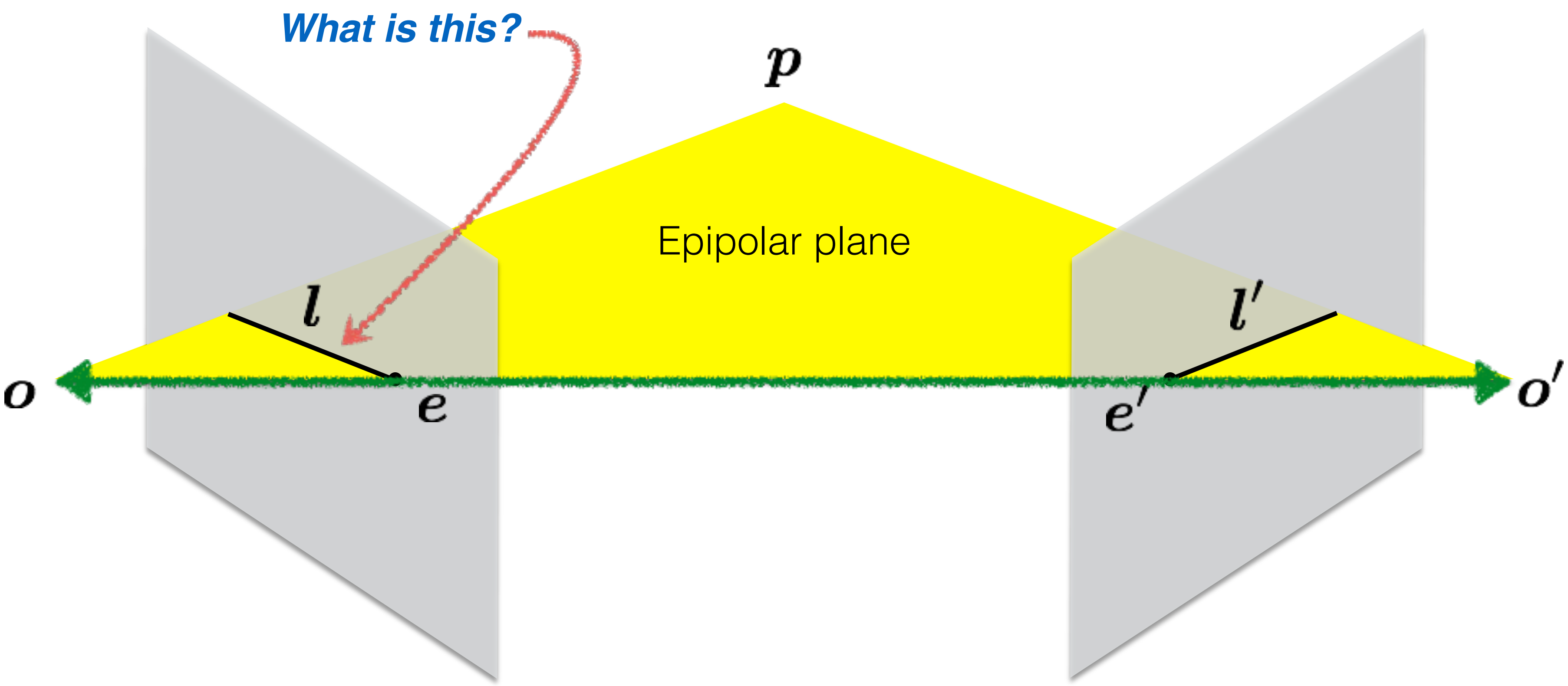
Quiz



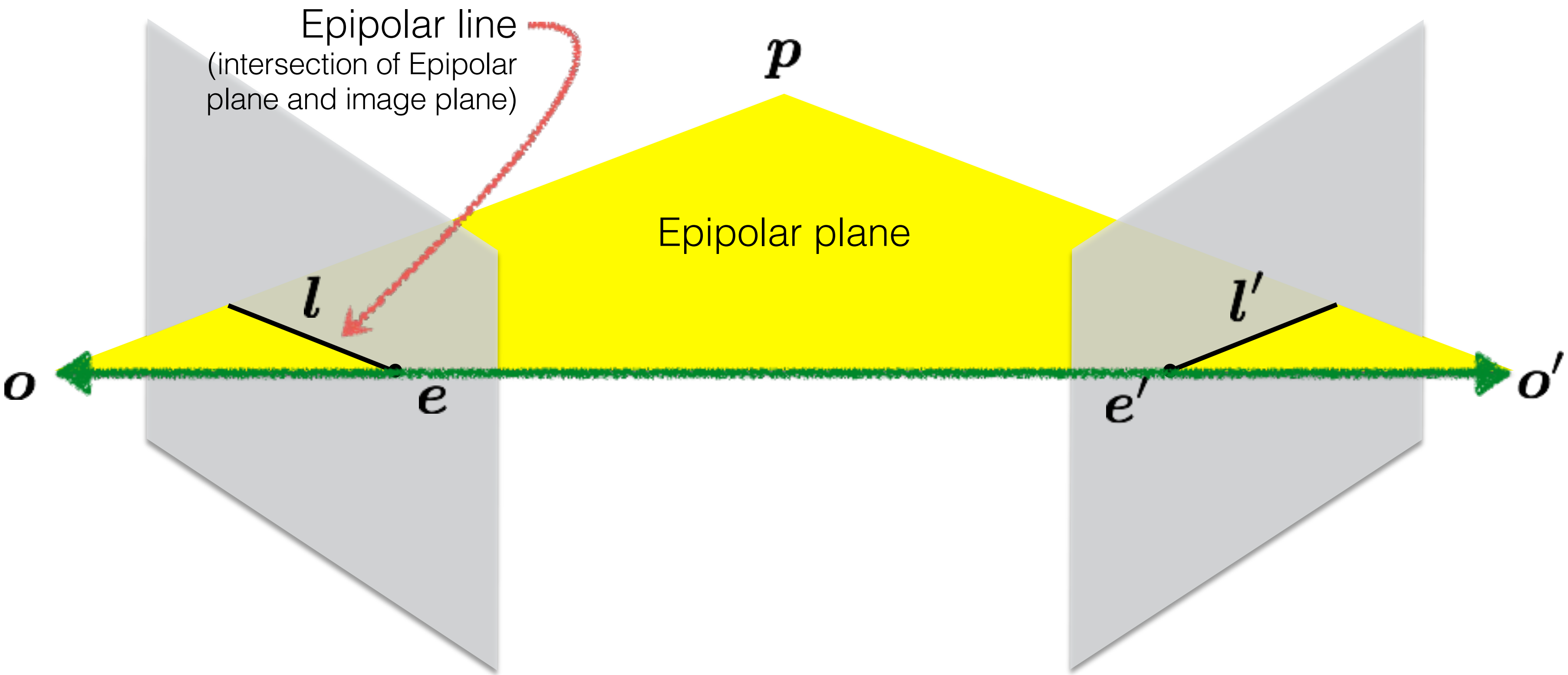
Quiz



Quiz

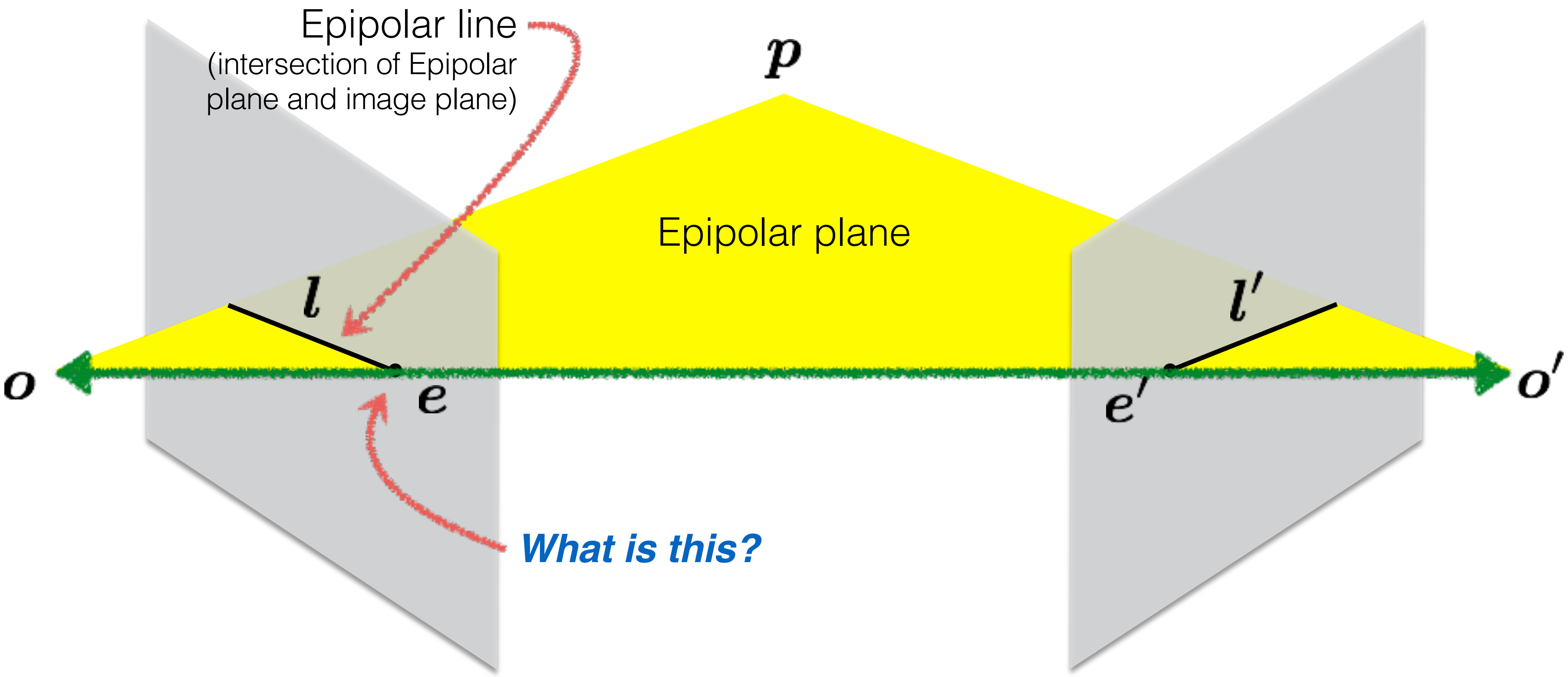

What is this?
$\boldsymbol{p}$
Epipolar plane
$\boldsymbol{l}$
$\boldsymbol{l'}$
$\boldsymbol{o}$
$\boldsymbol{e}$
$\boldsymbol{e'}$
$\boldsymbol{o'}$

Quiz

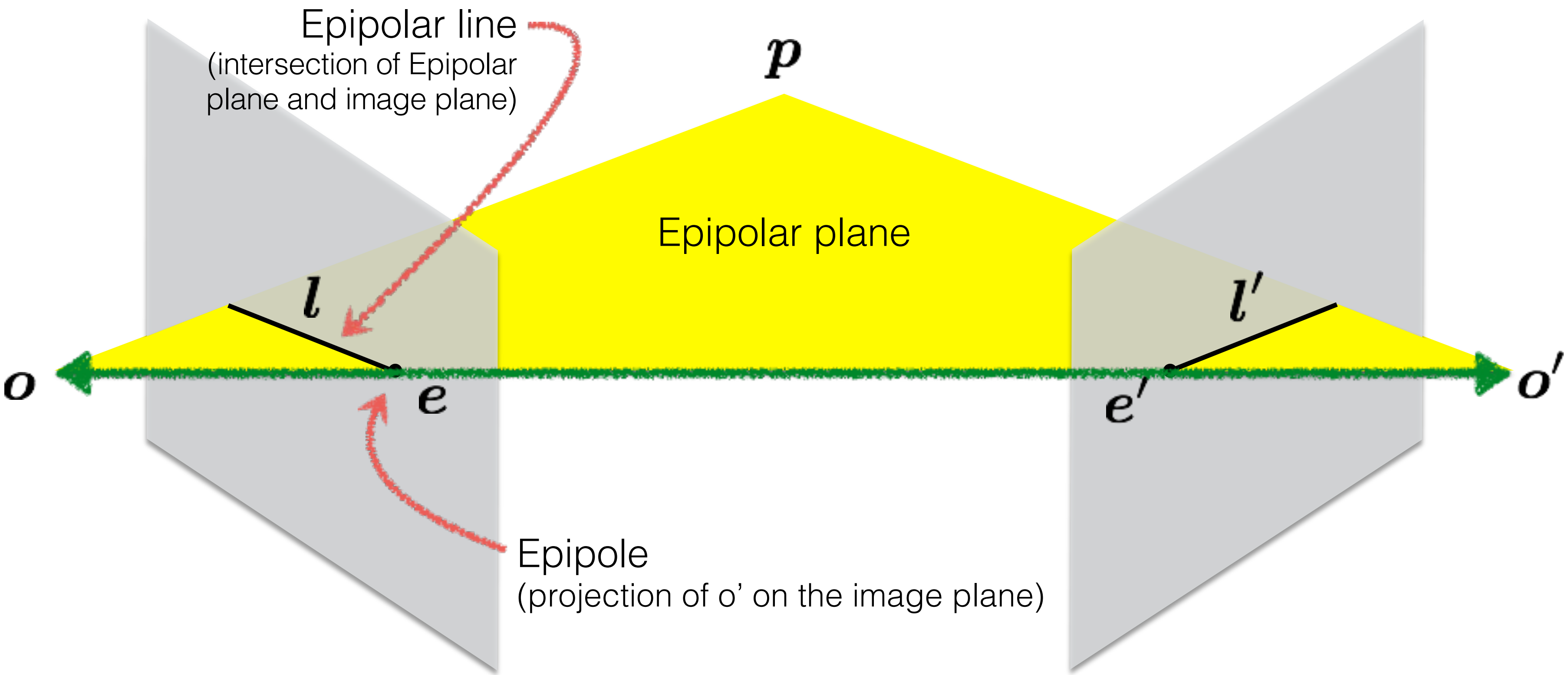


Epipolar line
(intersection of Epipolar
plane and image plane)
p
Epipolar plane
l
l'
o
e
e'
o'

Quiz

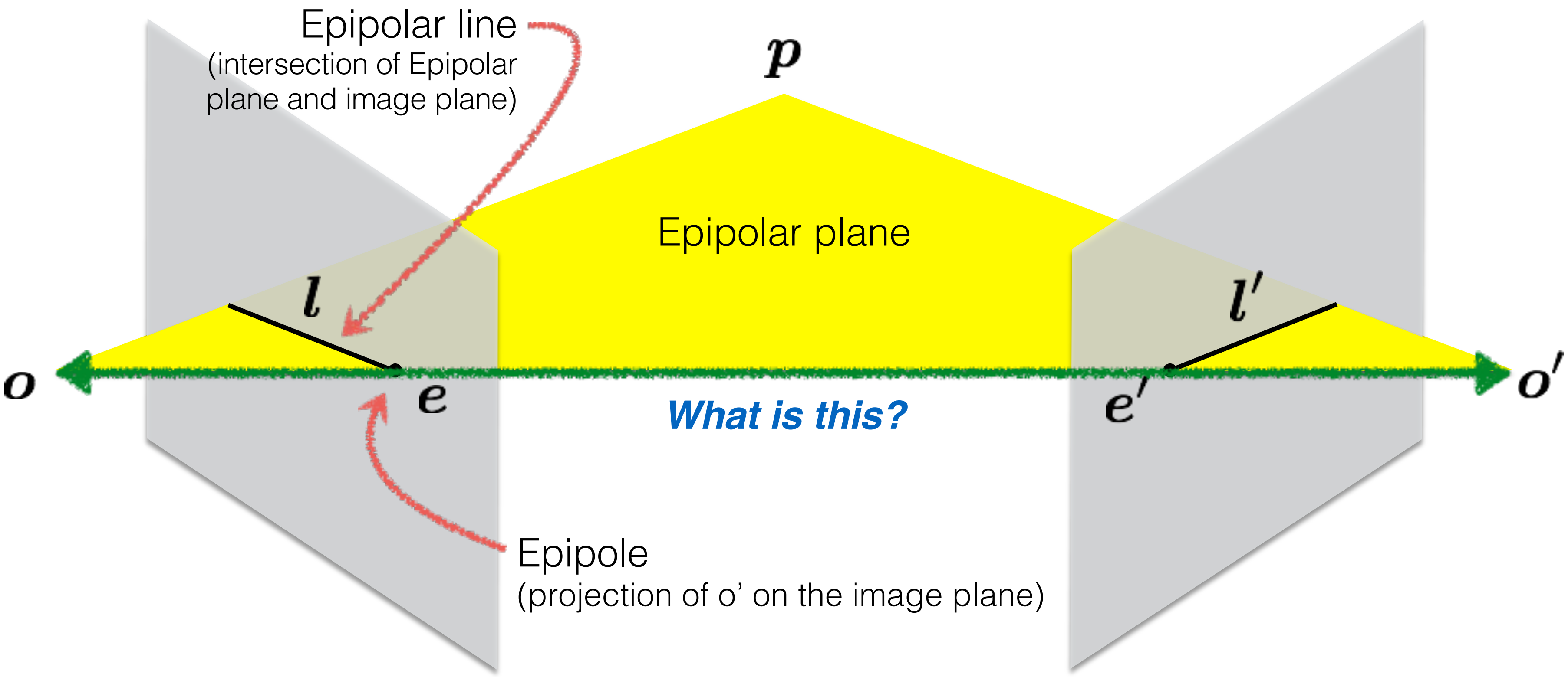

Epipolar line
(intersection of Epipolar plane and image plane)
p
Epipolar plane
l
l'
o
e
e'
o'
What is this?

Quiz

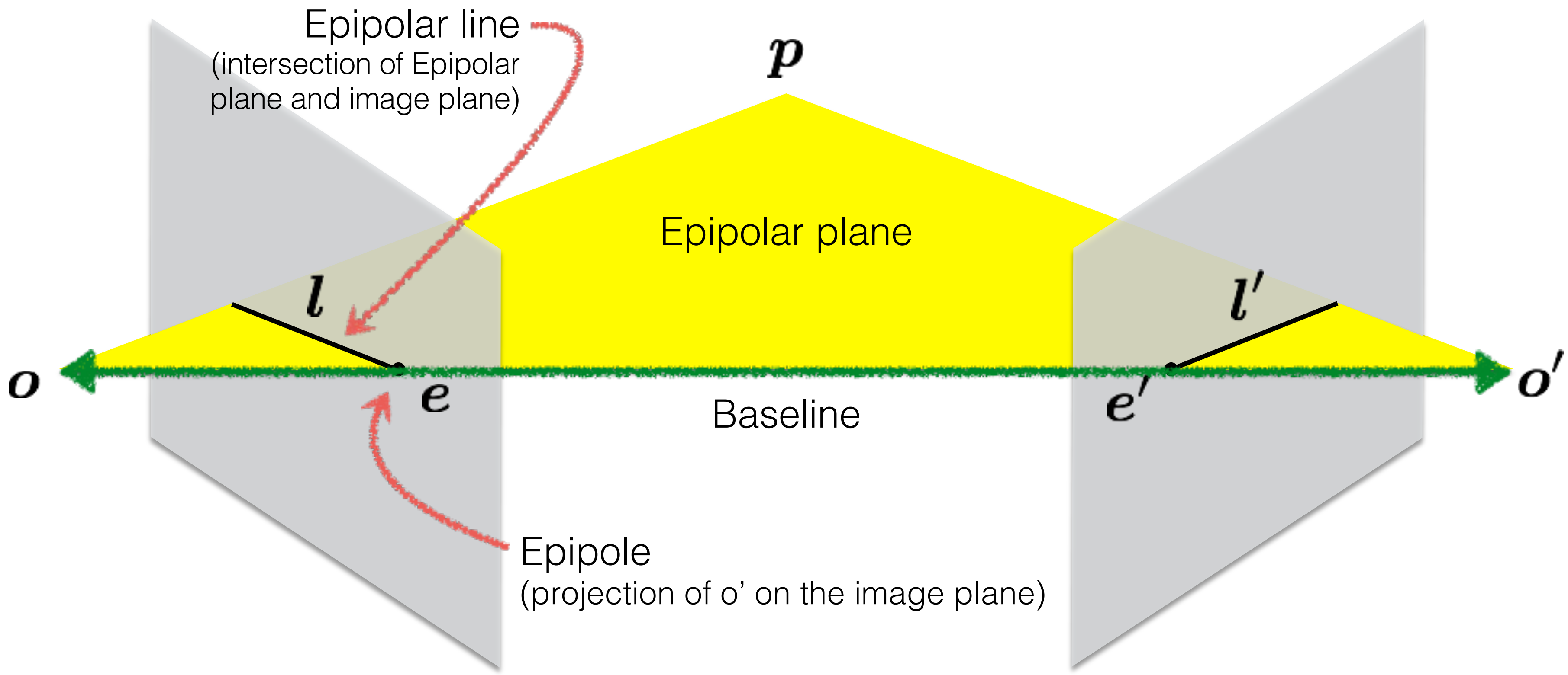


Epipolar line
(intersection of Epipolar plane and image plane)
p
Epipolar plane
l
l′
o
e
e′
o′
Epipole
(projection of o’ on the image plane)

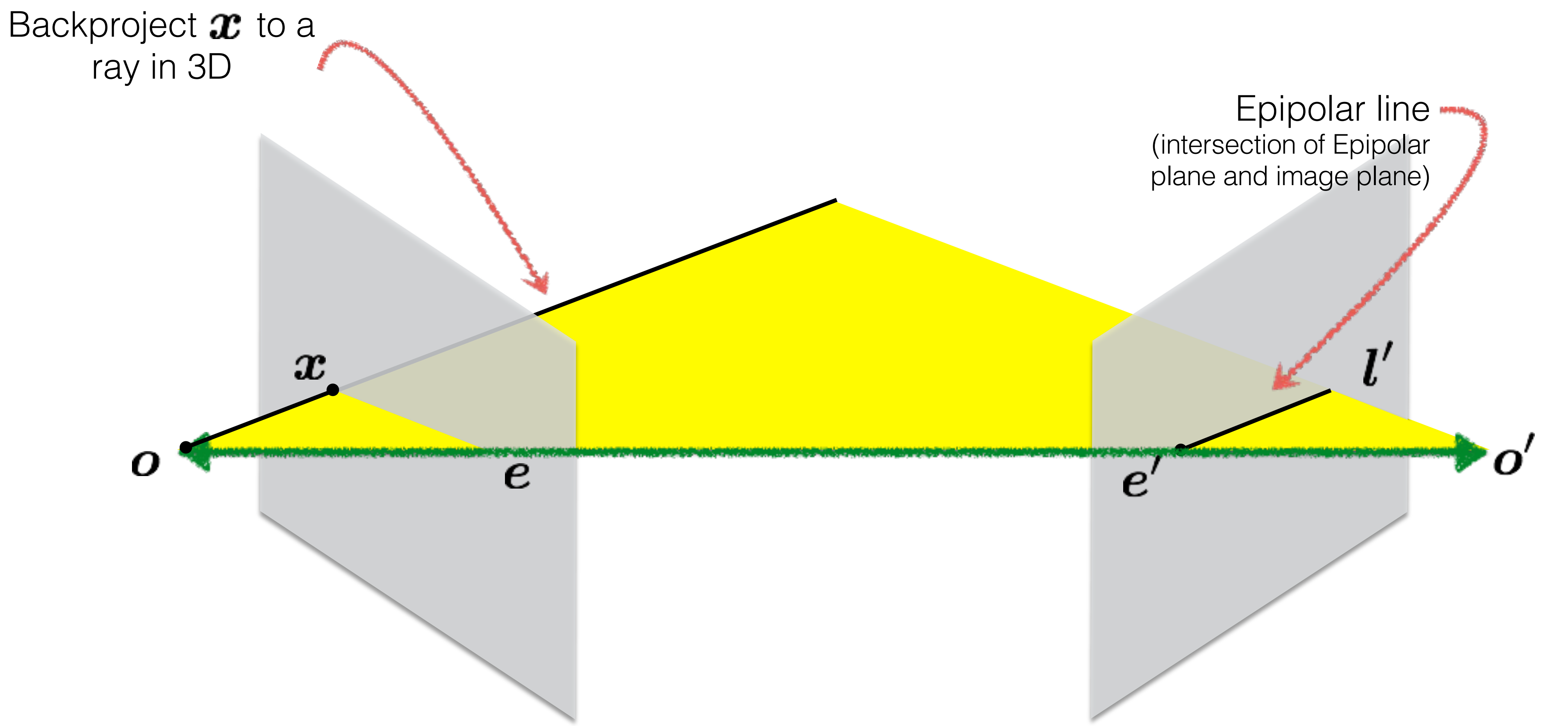
Quiz


Epipolar line
(intersection of Epipolar plane and image plane)
p
Epipolar plane
l
l′
o
e
e′
o′
What is this?
Epipole
(projection of o' on the image plane)

Quiz

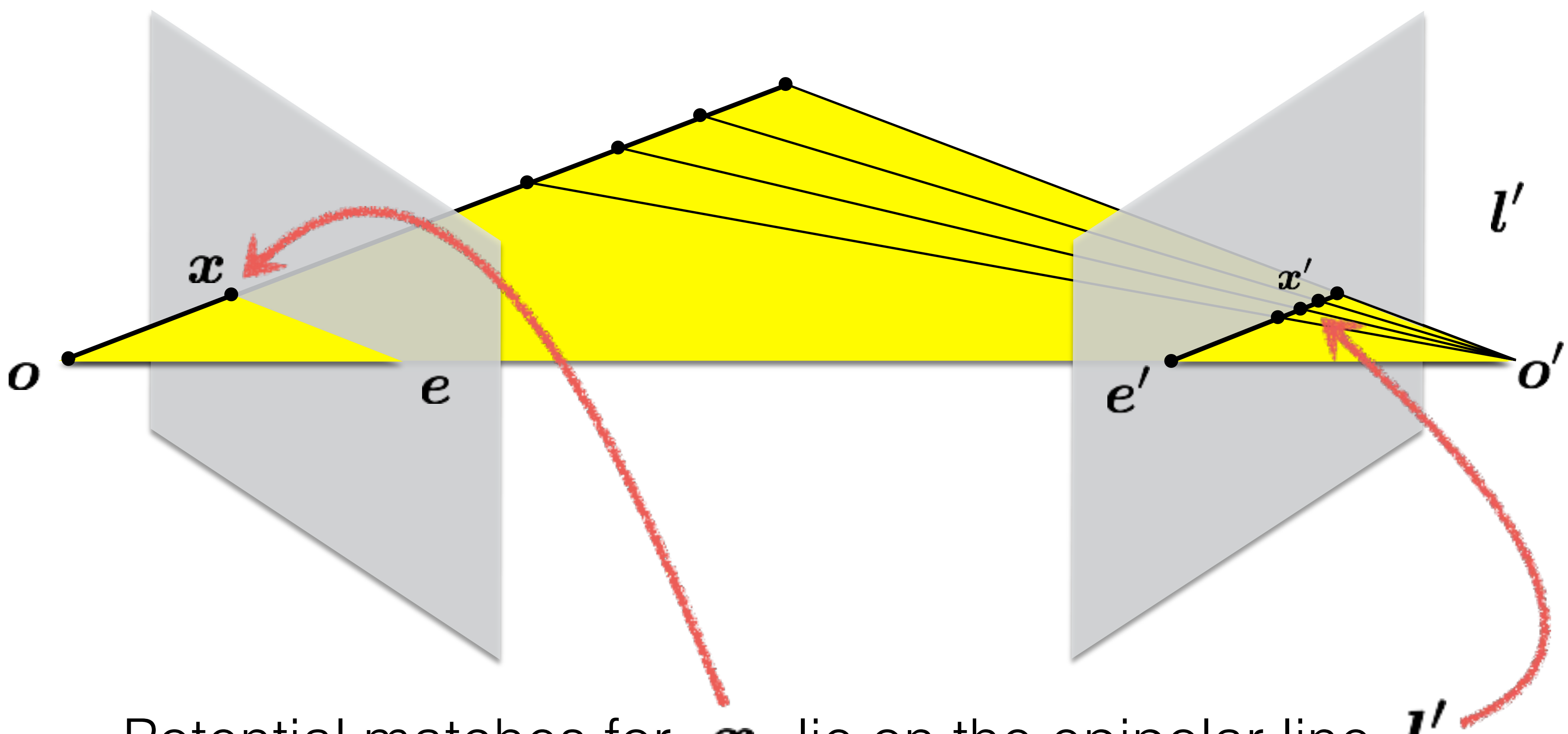

Epipolar line
(intersection of Epipolar plane and image plane)
p
Epipolar plane
l
l′
o
e
Baseline
e′
o′
Epipole
(projection of o’ on the image plane)

Epipolar Constraint

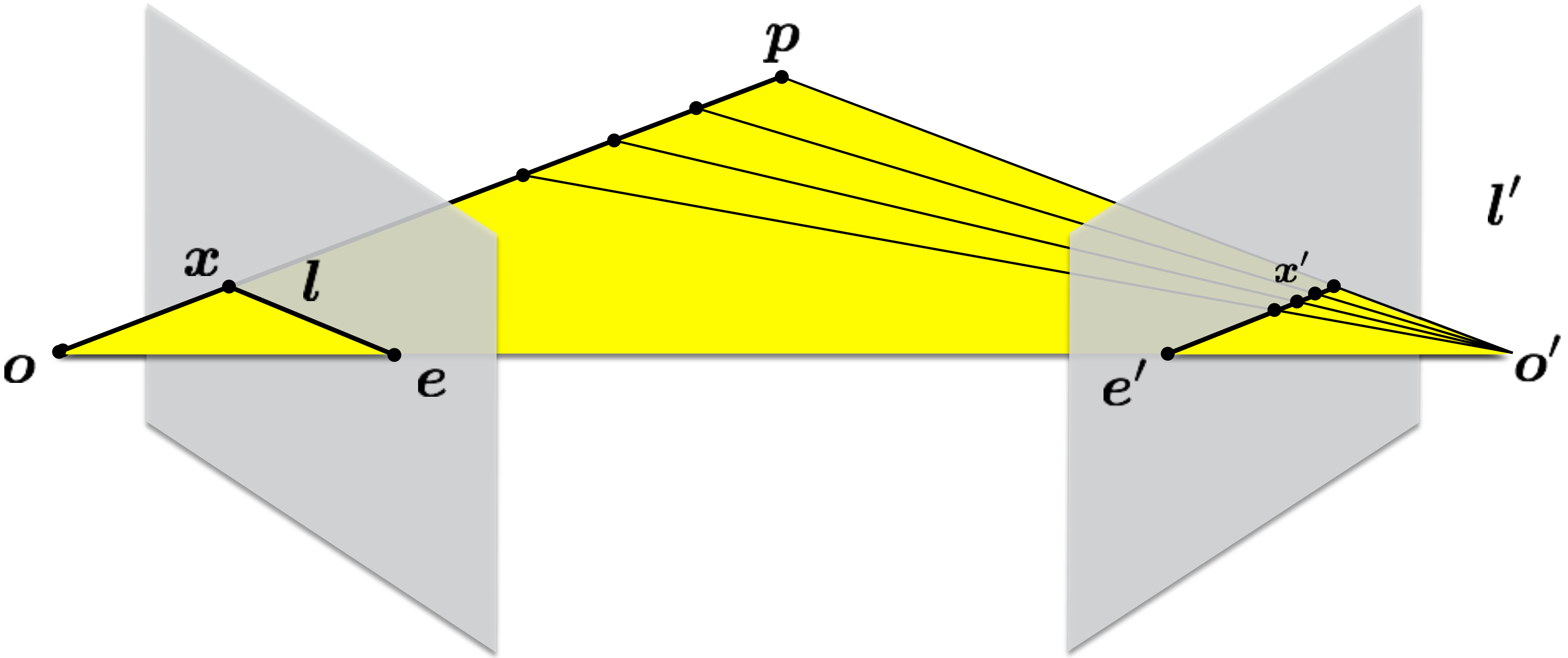


Another way to construct the epipolar plane, this time given $\boldsymbol{x}$

Epipolar Constraint



Potential matches for x lie on the epipolar line l'



The point **x** (left image) maps to a ___________ in the right image

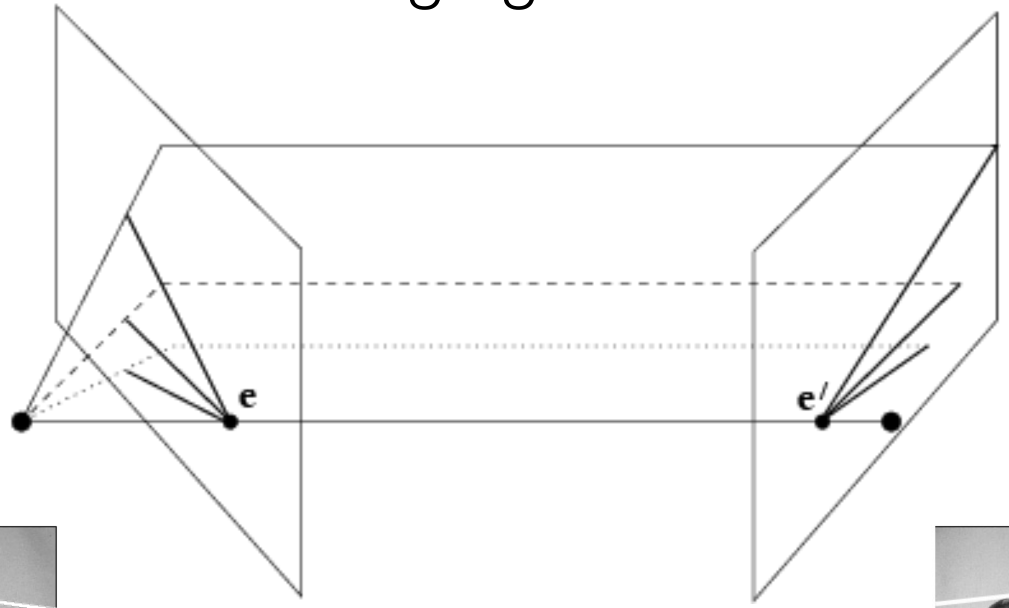
The baseline connects the ___________ and ____________

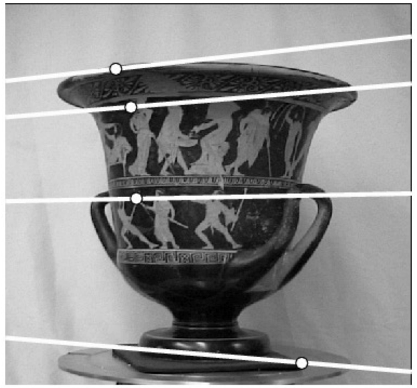
An epipolar line (left image) maps to a __________ in the right image

An epipole **e** is a projection of the ______________ on the image plane

All epipolar lines in an image intersect at the ______________

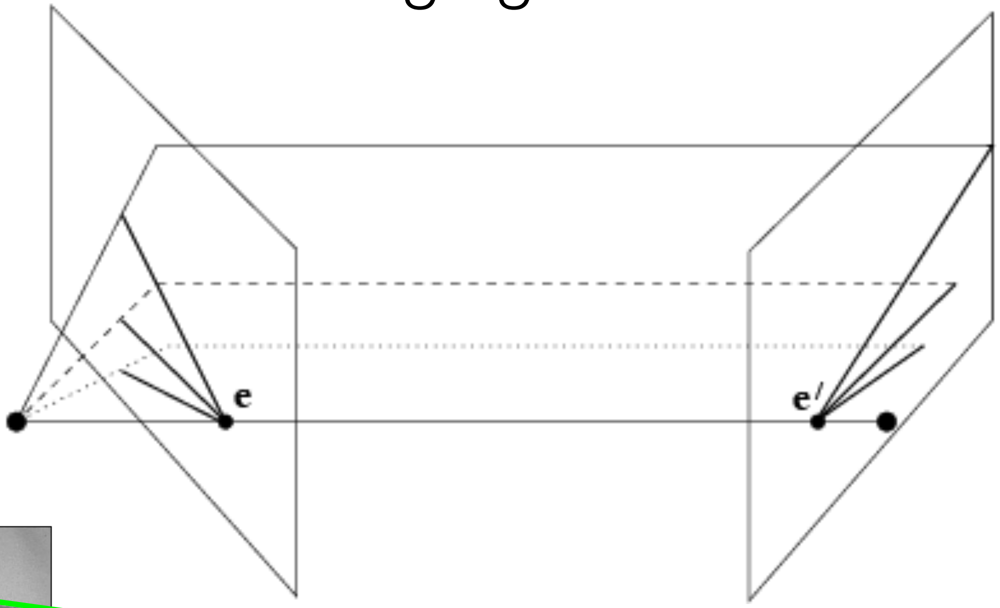
Converging cameras

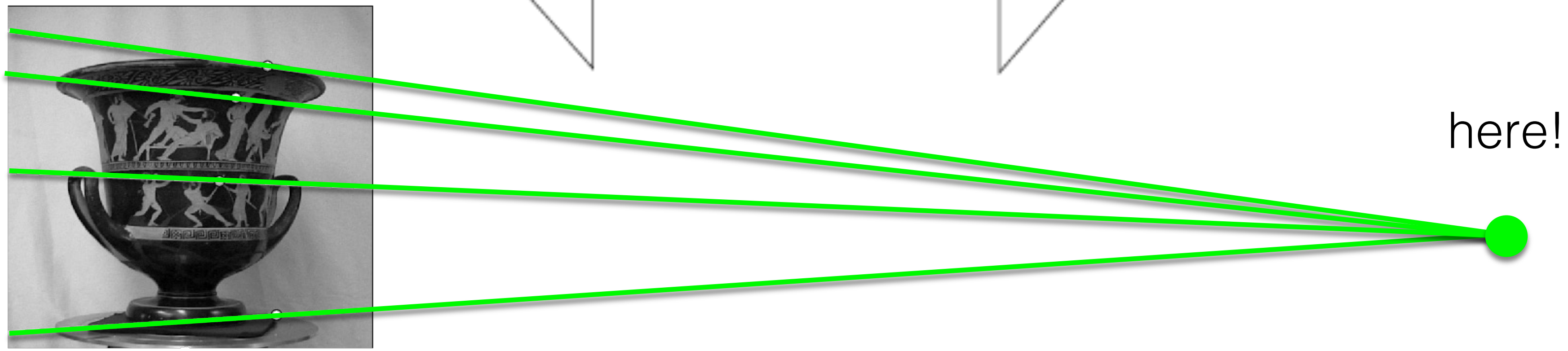




Where is the epipole in this image?

Converging cameras

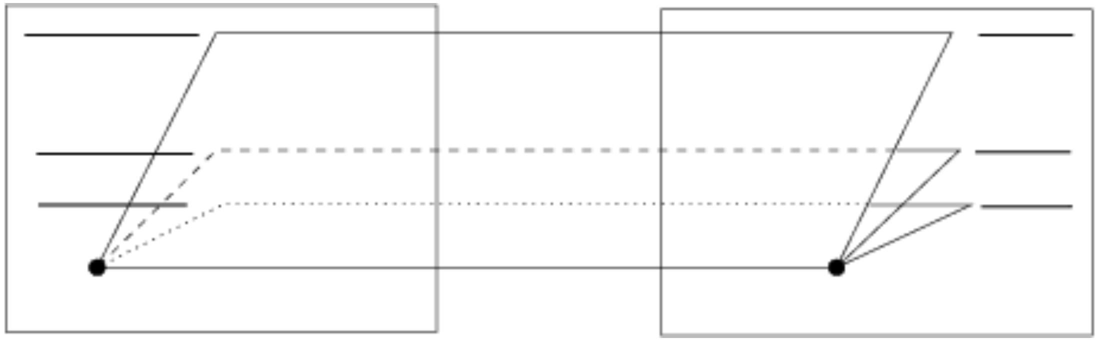




Where is the epipole in this image?

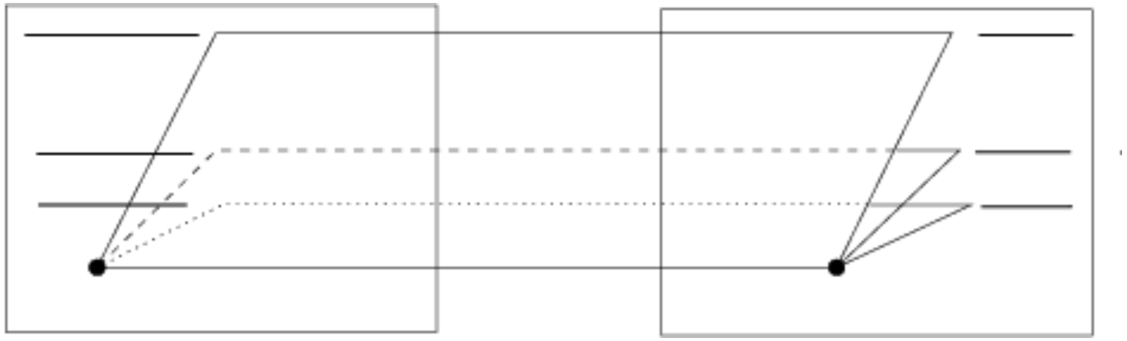
It's not always in the image

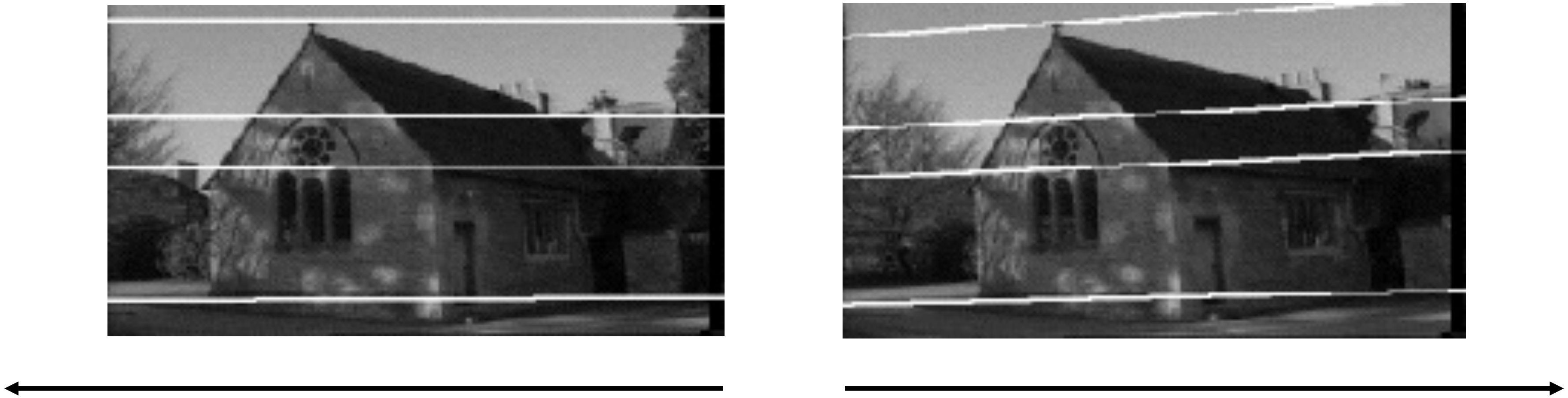
Parallel cameras



Where is the epipole?

Parallel cameras

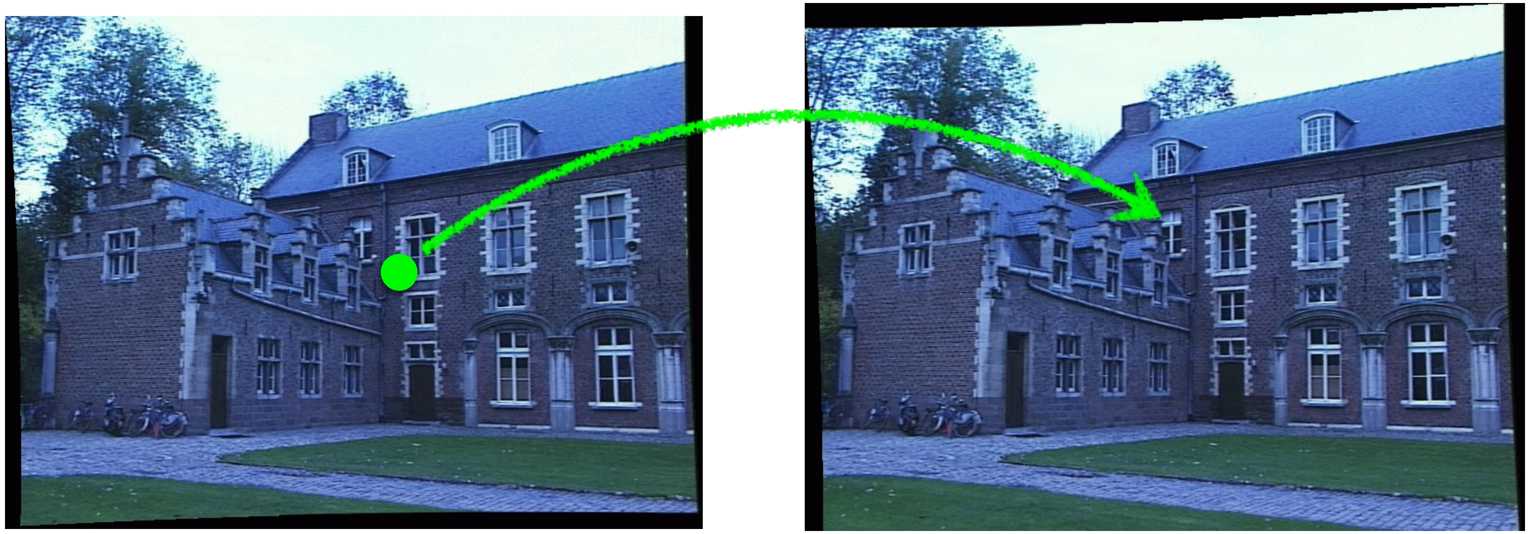




epipole at infinity

The epipolar constraint is an important concept for stereo vision

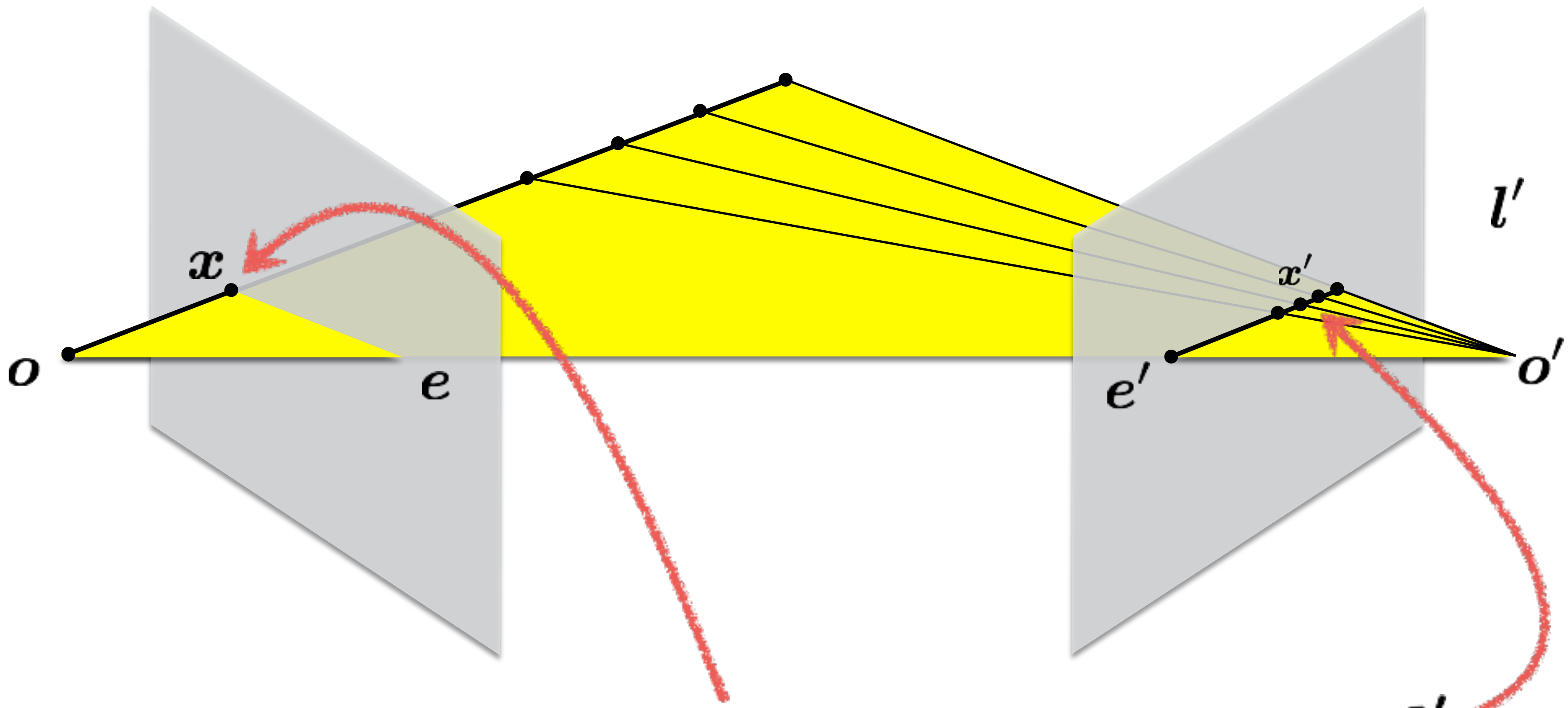
Task: Match point in left image to point in right image



Left image Right image

How would you do it?

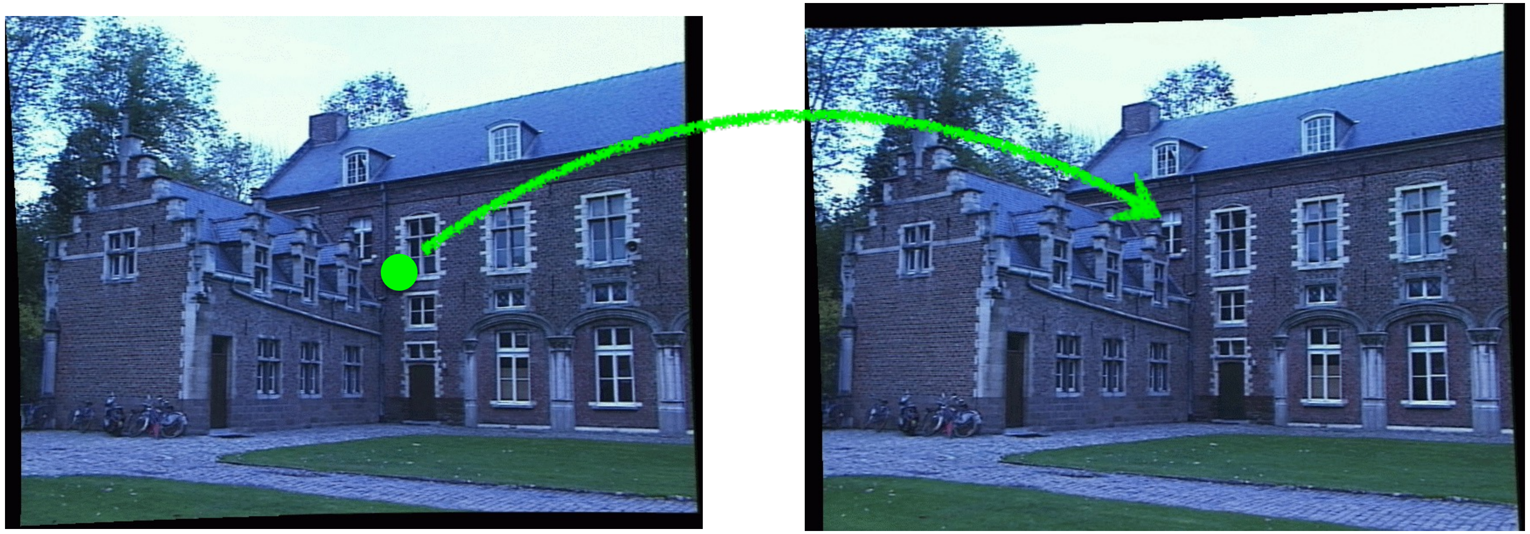
Epipolar Constraint



Potential matches for x lie on the epipolar line l'

The epipolar constraint is an important concept for stereo vision

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



Left image

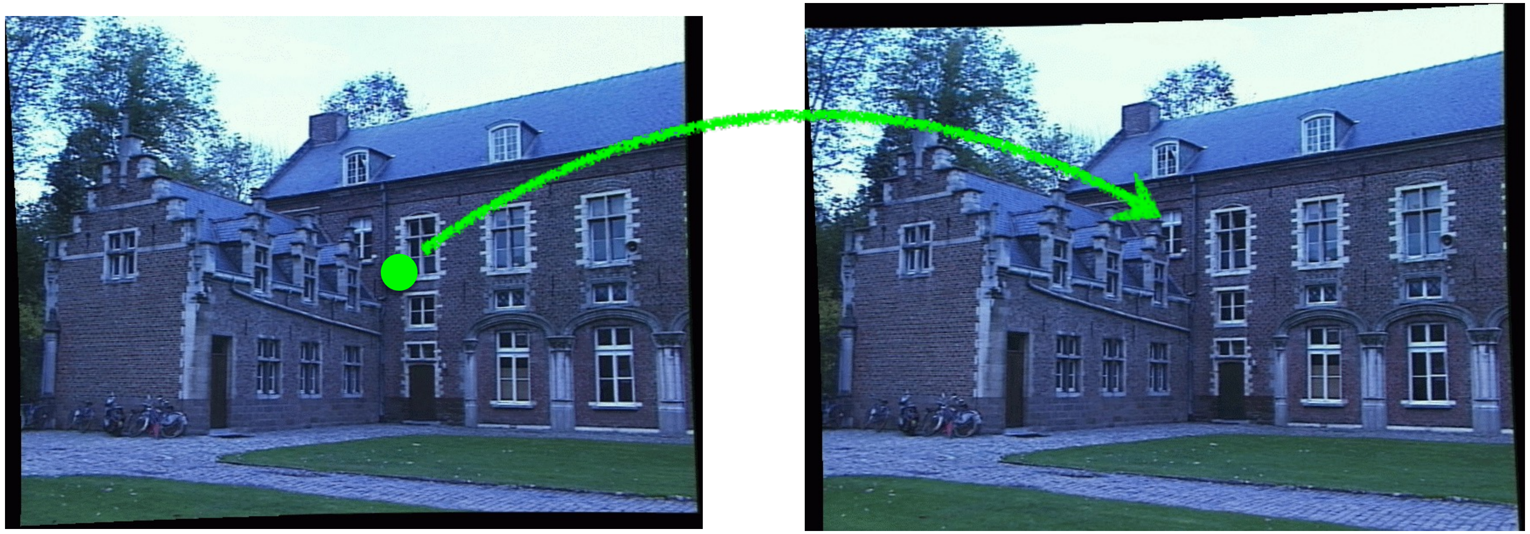
Right image

Want to avoid search over entire image

Epipolar constraint reduces search to a single line

The epipolar constraint is an important concept for stereo vision

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



Left image

Right image

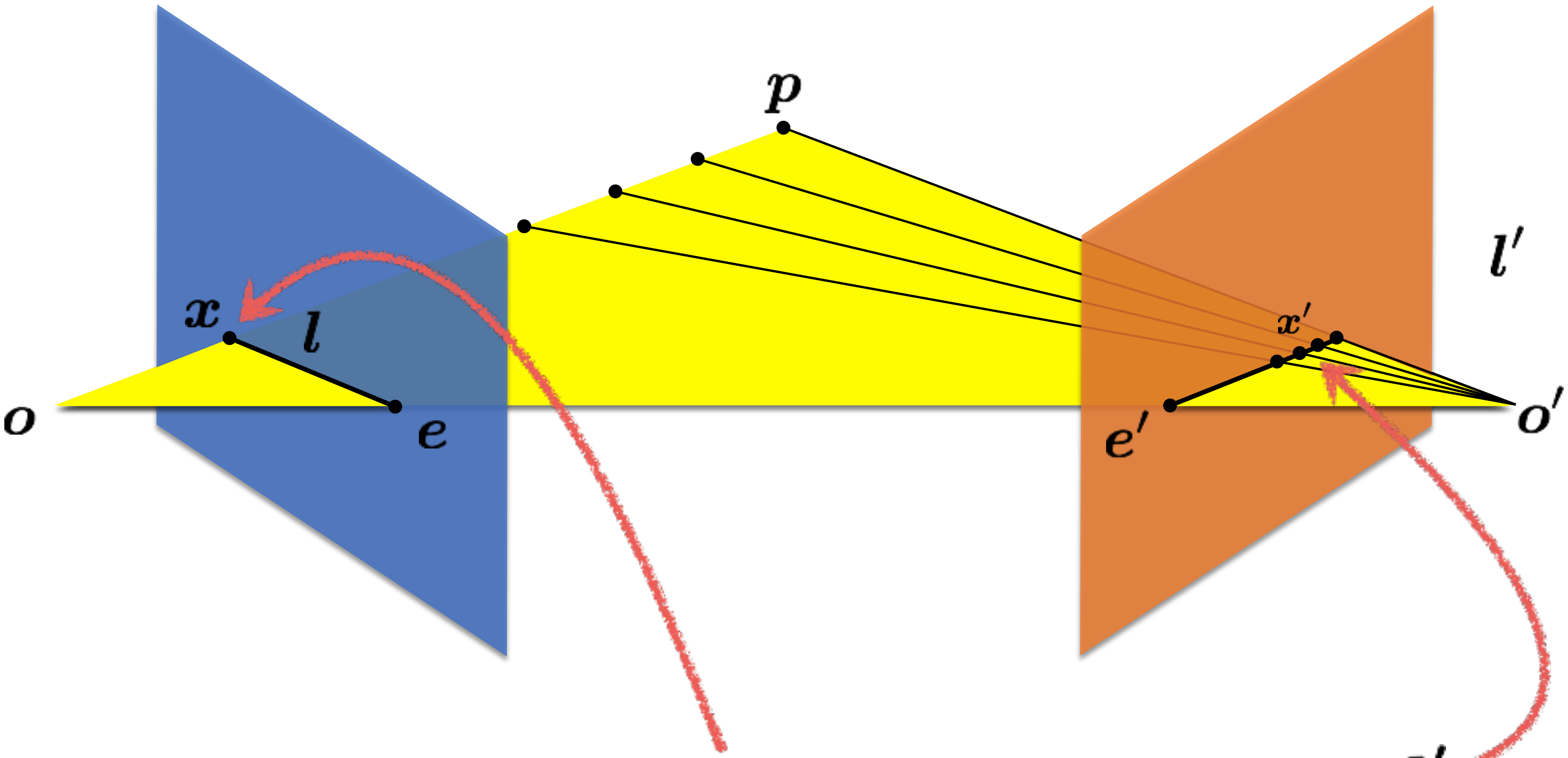
Want to avoid search over entire image

Epipolar constraint reduces search to a single line

How do you compute the epipolar line?

The essential matrix

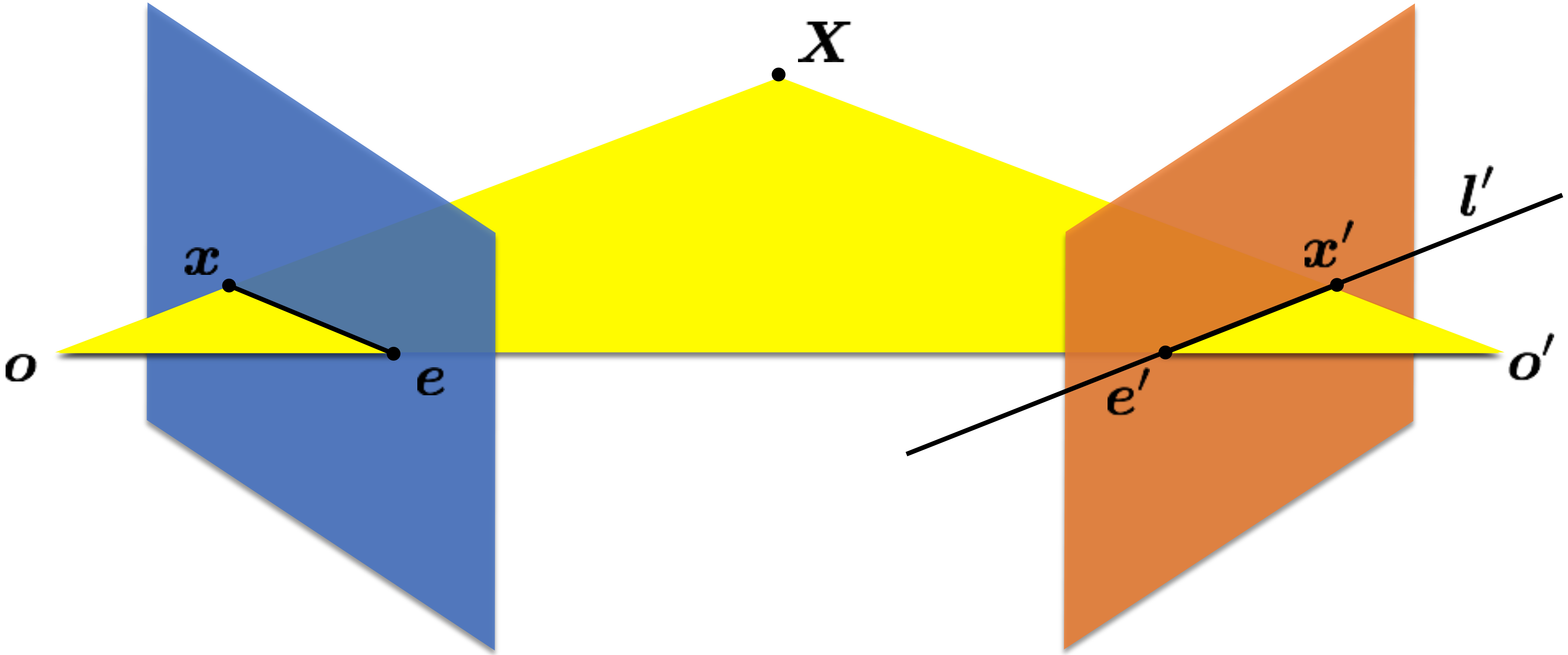
Recall: Epipolar Constraint



Potential matches for x lie on the epipolar line l'

Given a point in one image,
multiplying by the **essential matrix** will tell us
the **epipolar line** in the second view.

$$\mathbf{E}\boldsymbol{x} = \boldsymbol{l}'$$



Motivation

The Essential Matrix is a 3 x 3 matrix that
encodes **epipolar geometry**

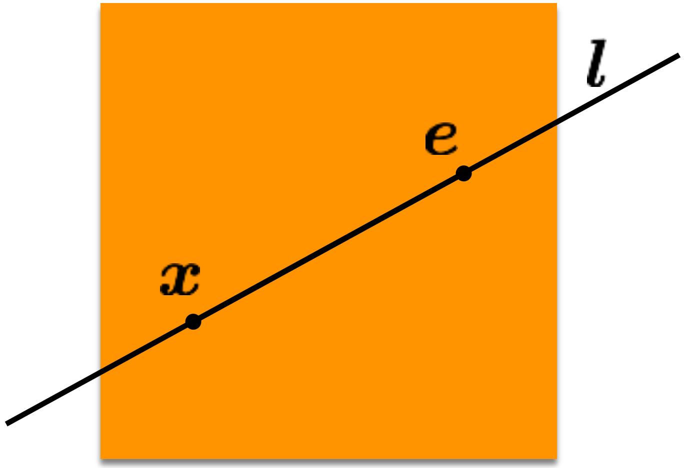
Given a point in one image,
multiplying by the **essential matrix** will tell us
the **epipolar line** in the second image.

Representing the epipolar line

$$ax + by + c = 0$$

in vector form

$$\boldsymbol{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$



If the point $\boldsymbol{x}$ is on the epipolar line $\boldsymbol{l}$ then

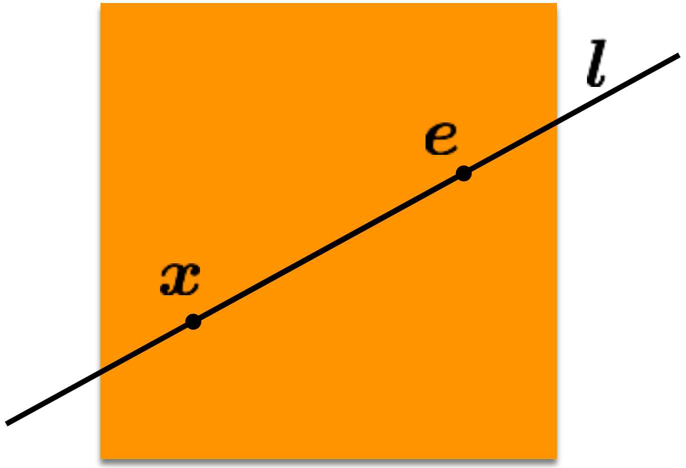
$$\boldsymbol{x}^\top \boldsymbol{l} = \ ?$$

Representing the epipolar line

$$ax + by + c = 0$$

in vector form

$$\boldsymbol{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

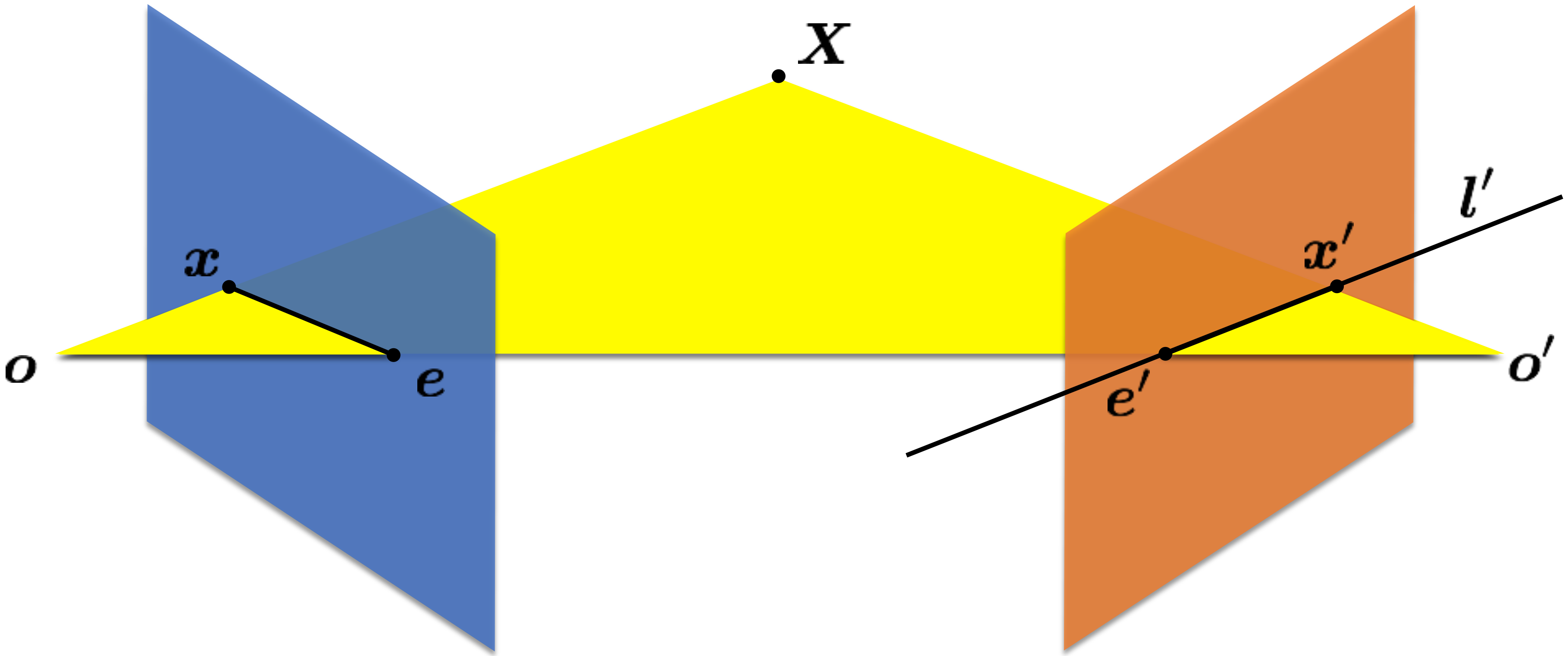


If the point $\boldsymbol{x}$ is on the epipolar line $\boldsymbol{l}$ then

$$\boldsymbol{x}^\top \boldsymbol{l} = 0$$

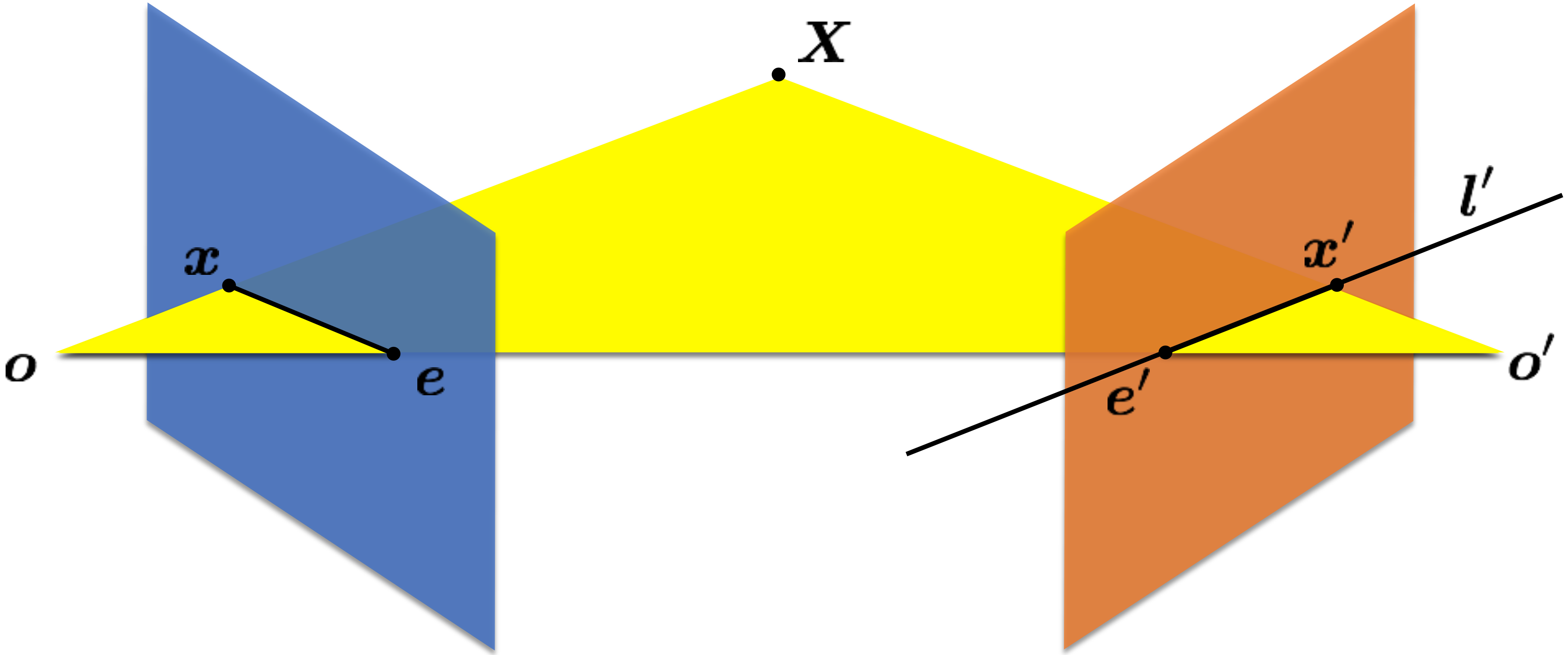
So if $\boldsymbol{x}'^\top \boldsymbol{l}' = 0$ and $\mathbf{E}\boldsymbol{x} = \boldsymbol{l}'$ then

$$\boldsymbol{x}'^\top \mathbf{E}\boldsymbol{x} = \ ?$$



So if $\boldsymbol{x}'^{\top}\boldsymbol{l}' = 0$ and $\mathbf{E}\boldsymbol{x} = \boldsymbol{l}'$ then

$$\boldsymbol{x}'^{\top}\mathbf{E}\boldsymbol{x} = 0$$



What’s the difference between the essential matrix and a homography?

Essential Matrix vs Homography

What's the difference between the essential matrix and a homography?

They are both 3 x 3 matrices but …

Essential Matrix vs Homography

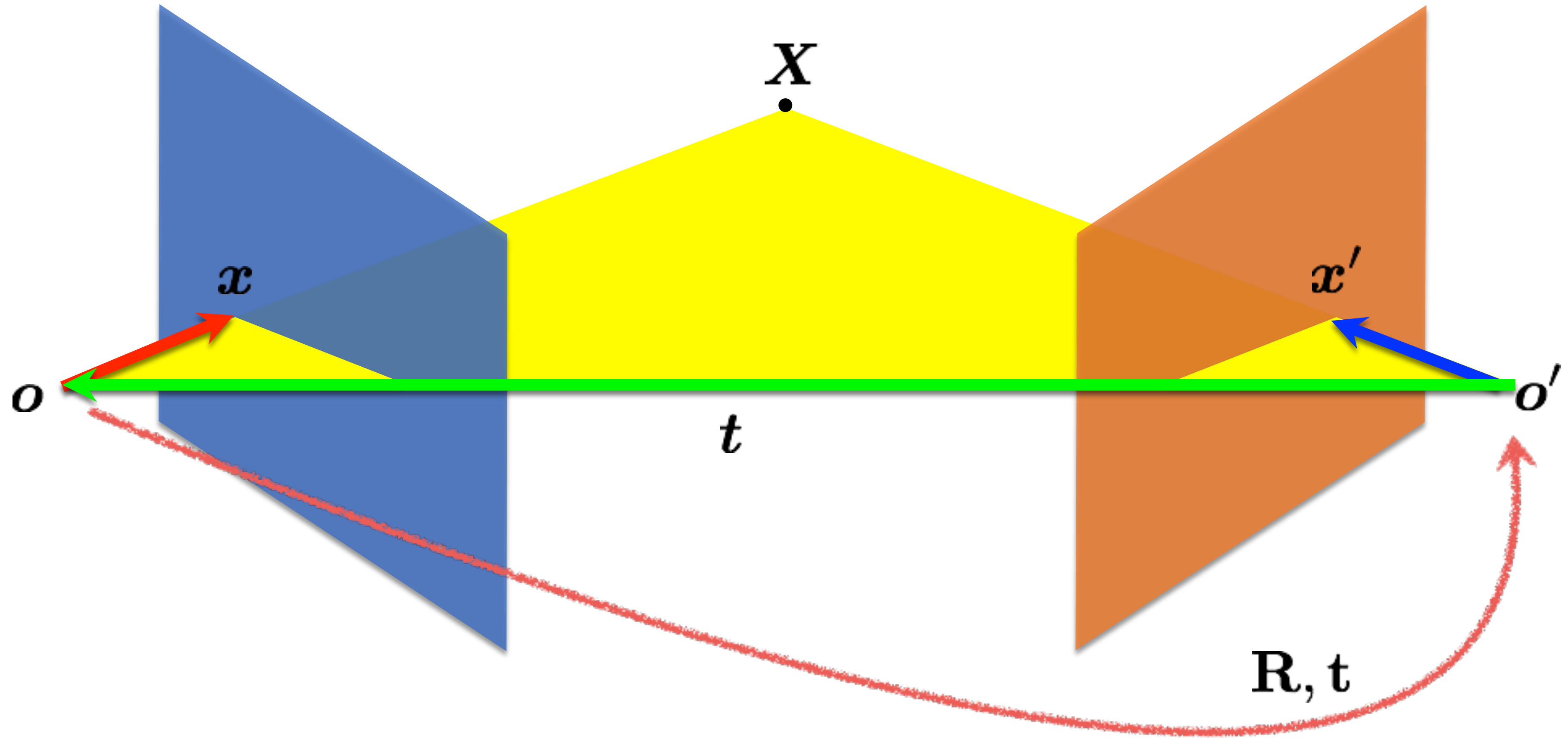
$$\boldsymbol{l}' = \mathbf{E}\boldsymbol{x}$$

Essential matrix maps a **point** to a **line**

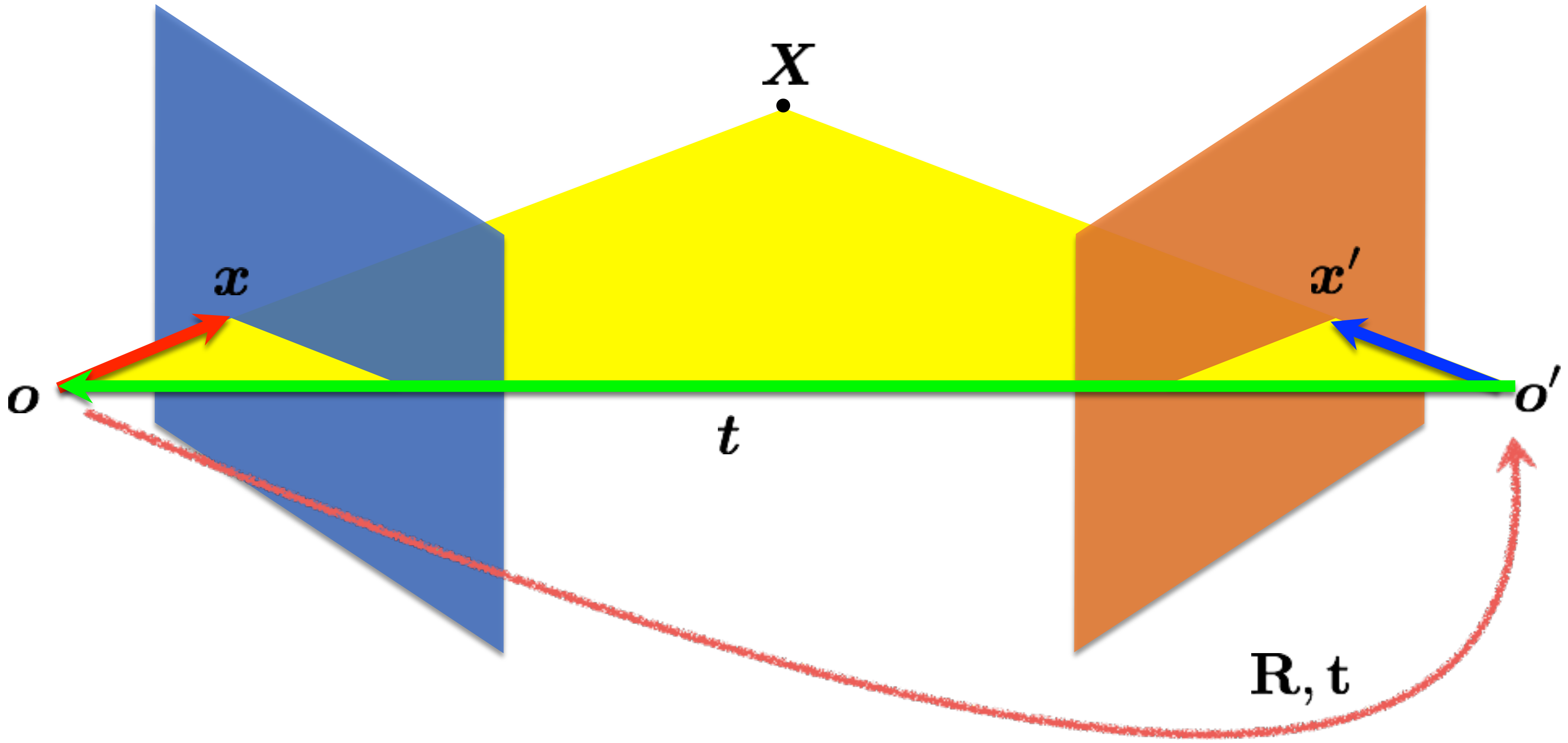
$$\boldsymbol{x}' = \mathbf{H}\boldsymbol{x}$$

Homography maps a **point** to a **point**

Where does the essential matrix come from?

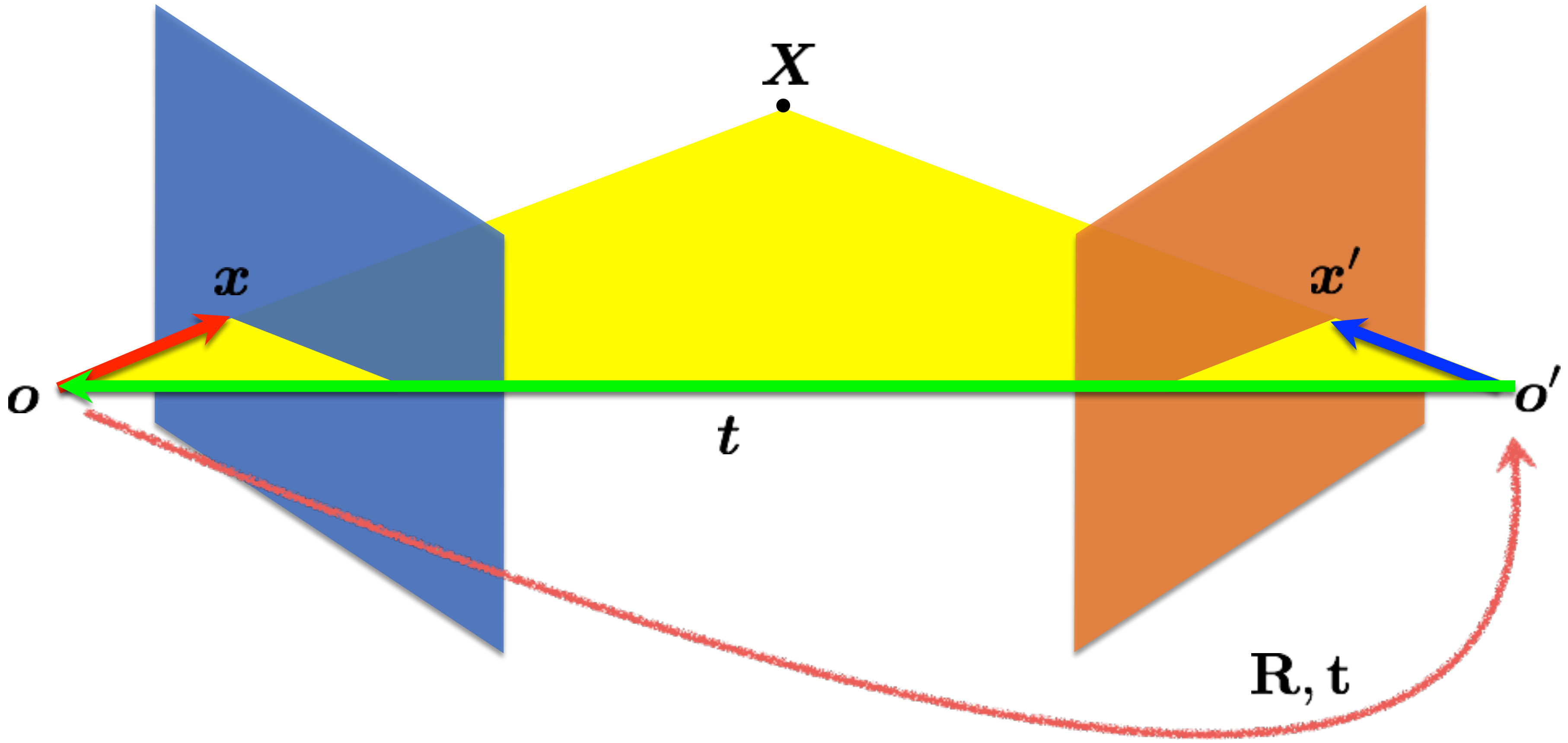


$$\boldsymbol{x}' = \mathbf{R}(\boldsymbol{x} - \boldsymbol{t})$$



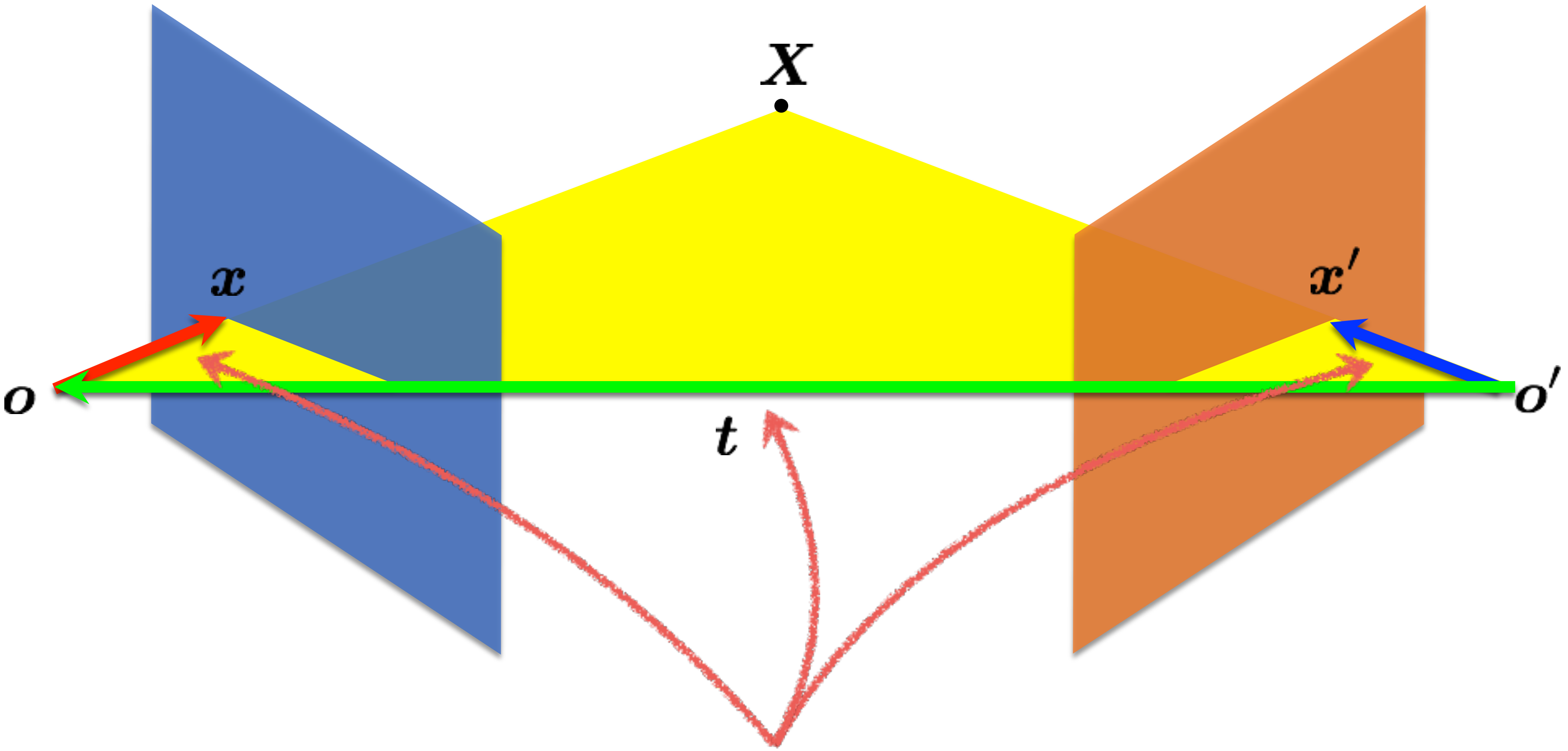
$$\boldsymbol{x}' = \mathbf{R}(\boldsymbol{x} - \boldsymbol{t})$$

Does this look familiar?



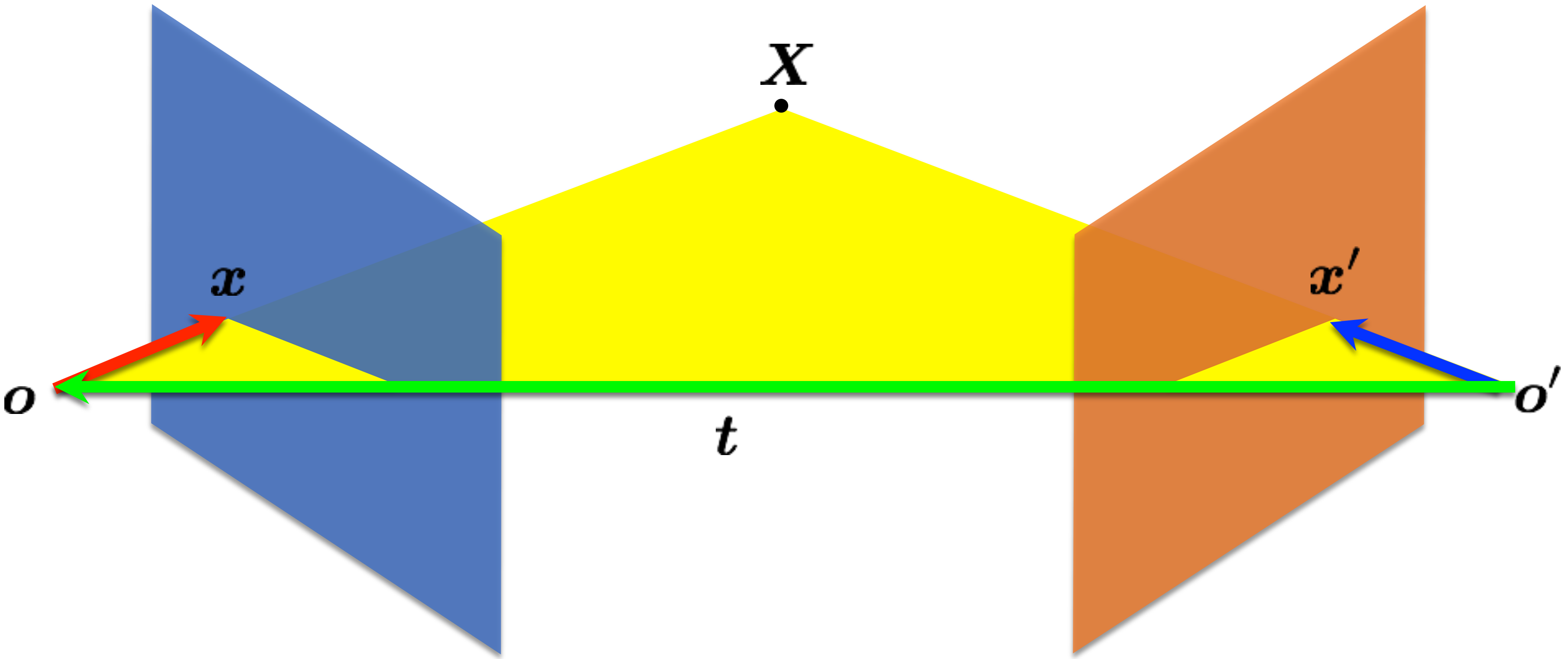
$$x' = \mathbf{R}(x - t)$$

Camera-camera transform just like **world-camera** transform



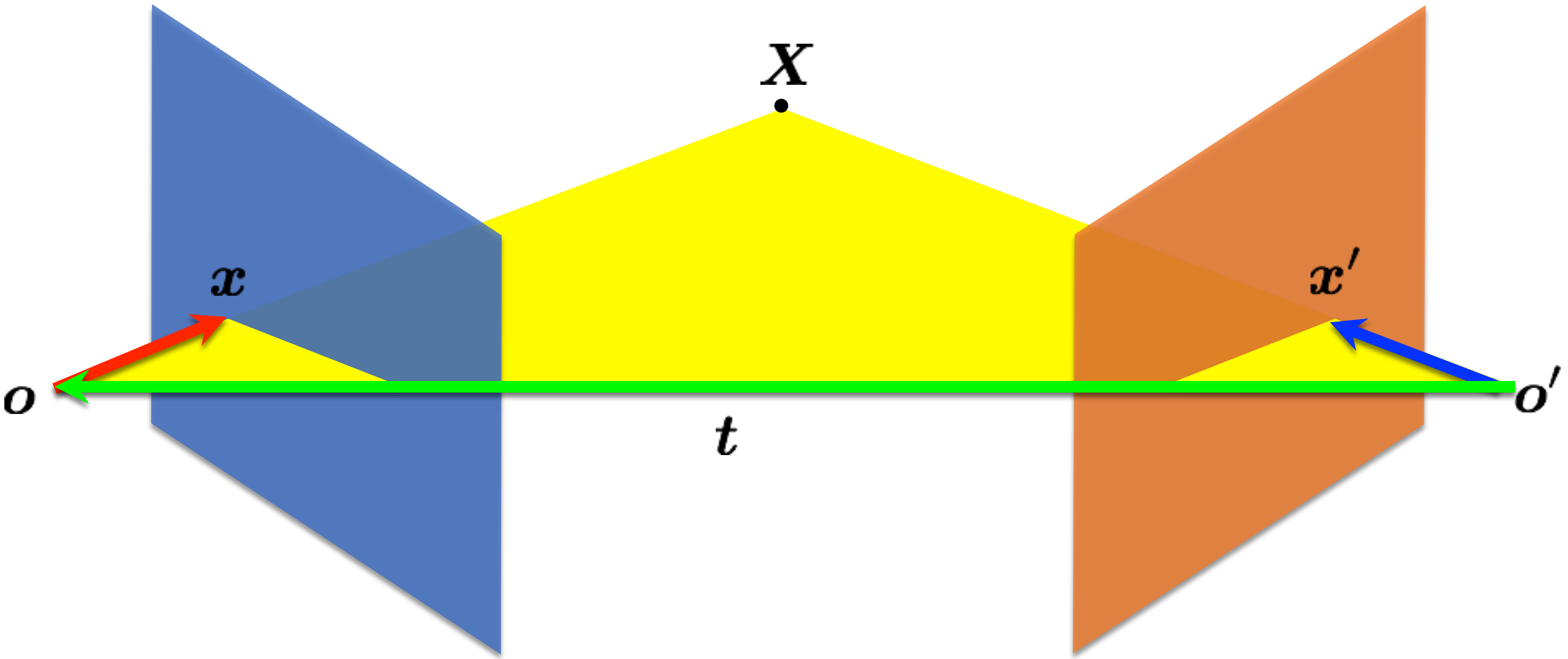
These three vectors are coplanar

$$\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{x}'$$



If these three vectors are coplanar $\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{x}'$ then

$$\boldsymbol{x}^\top(\boldsymbol{t} \times \boldsymbol{x}) = \ ?$$

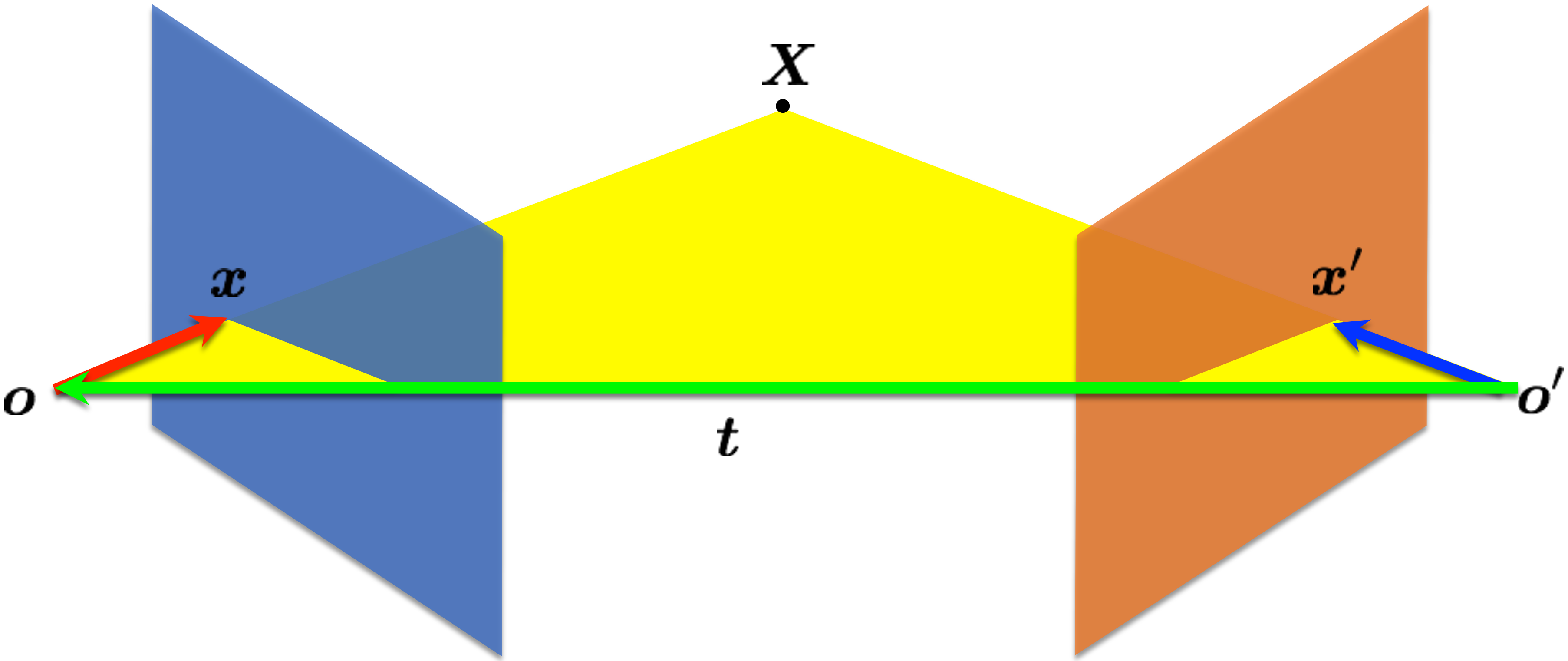


If these three vectors are coplanar $\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{x}'$ then

$$\boldsymbol{x}^\top(\boldsymbol{t} \times \boldsymbol{x}) = 0$$

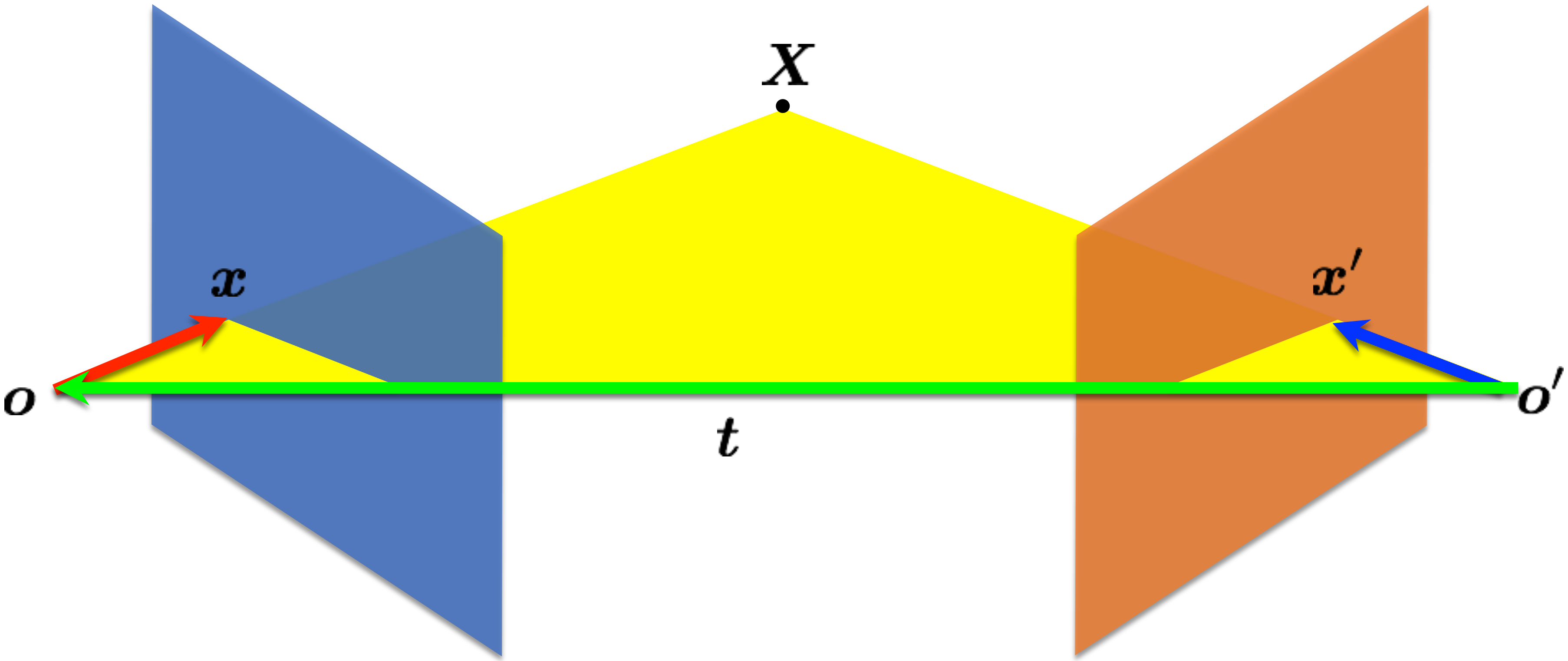
dot product of orthogonal vectors

cross-product: vector orthogonal to plane



If these three vectors are coplanar $\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{x}'$ then

$$(\boldsymbol{x} - \boldsymbol{t})^\top (\boldsymbol{t} \times \boldsymbol{x}) = \ ?$$



If these three vectors are coplanar $\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{x}'$ then

$$(\boldsymbol{x} - \boldsymbol{t})^\top (\boldsymbol{t} \times \boldsymbol{x}) = 0$$

Putting it together

rigid motion

$$\boldsymbol{x}' = \mathbf{R}(\boldsymbol{x} - \boldsymbol{t})$$

coplanarity

$$(\boldsymbol{x} - \boldsymbol{t})^\top (\boldsymbol{t} \times \boldsymbol{x}) = 0$$

$$(\boldsymbol{x}'^\top \mathbf{R})(\boldsymbol{t} \times \boldsymbol{x}) = 0$$

Putting it together

rigid motion

$$\boldsymbol{x}' = \mathbf{R}(\boldsymbol{x} - \boldsymbol{t})$$

coplanarity

$$(\boldsymbol{x} - \boldsymbol{t})^\top (\boldsymbol{t} \times \boldsymbol{x}) = 0$$

$$(\boldsymbol{x}'^\top \mathbf{R})(\boldsymbol{t} \times \boldsymbol{x}) = 0$$

use skew-symmetric matrix to represent cross product

$$(\boldsymbol{x}'^\top \mathbf{R})([\mathbf{t}_\times]\boldsymbol{x}) = 0$$

Putting it together

rigid motion

$$\boldsymbol{x}' = \mathbf{R}(\boldsymbol{x} - \boldsymbol{t})$$

coplanarity

$$(\boldsymbol{x} - \boldsymbol{t})^\top (\boldsymbol{t} \times \boldsymbol{x}) = 0$$

$$(\boldsymbol{x}'^\top \mathbf{R})(\boldsymbol{t} \times \boldsymbol{x}) = 0$$

$$(\boldsymbol{x}'^\top \mathbf{R})([\mathbf{t}_\times]\boldsymbol{x}) = 0$$

$$\boldsymbol{x}'^\top (\mathbf{R}[\mathbf{t}_\times])\boldsymbol{x} = 0$$

Putting it together

rigid motion

$$\boldsymbol{x}' = \mathbf{R}(\boldsymbol{x} - \boldsymbol{t})$$

coplanarity

$$(\boldsymbol{x} - \boldsymbol{t})^\top (\boldsymbol{t} \times \boldsymbol{x}) = 0$$

$$(\boldsymbol{x}'^\top \mathbf{R})(\boldsymbol{t} \times \boldsymbol{x}) = 0$$

$$(\boldsymbol{x}'^\top \mathbf{R})([\mathbf{t}_\times]\boldsymbol{x}) = 0$$

$$\boldsymbol{x}'^\top (\mathbf{R}[\mathbf{t}_\times])\boldsymbol{x} = 0$$

$$\boldsymbol{x}'^\top \mathbf{E}\boldsymbol{x} = 0$$

Putting it together

rigid motion

$$\boldsymbol{x}' = \mathbf{R}(\boldsymbol{x} - \boldsymbol{t})$$

coplanarity

$$(\boldsymbol{x} - \boldsymbol{t})^\top (\boldsymbol{t} \times \boldsymbol{x}) = 0$$

$$(\boldsymbol{x}'^\top \mathbf{R})(\boldsymbol{t} \times \boldsymbol{x}) = 0$$

$$(\boldsymbol{x}'^\top \mathbf{R})([\mathbf{t}_\times]\boldsymbol{x}) = 0$$

$$\boldsymbol{x}'^\top (\mathbf{R}[\mathbf{t}_\times])\boldsymbol{x} = 0$$

$$\boxed{\boldsymbol{x}'^\top \mathbf{E}\boldsymbol{x} = 0}$$

Essential Matrix
[Longuet-Higgins 1981]

properties of the E matrix

Longuet-Higgins equation $\boldsymbol{x}'^\top \mathbf{E} \boldsymbol{x} = 0$

(2D points expressed in camera coordinate system)

properties of the E matrix

Longuet-Higgins equation $\boldsymbol{x}'^\top \mathbf{E} \boldsymbol{x} = 0$

Epipolar lines

$\boldsymbol{x}^\top \boldsymbol{l} = 0$ $\boldsymbol{x}'^\top \boldsymbol{l}' = 0$

$\boldsymbol{l}' = \mathbf{E} \boldsymbol{x}$ $\boldsymbol{l} = \mathbf{E}^T \boldsymbol{x}'$

(2D points expressed in camera coordinate system)

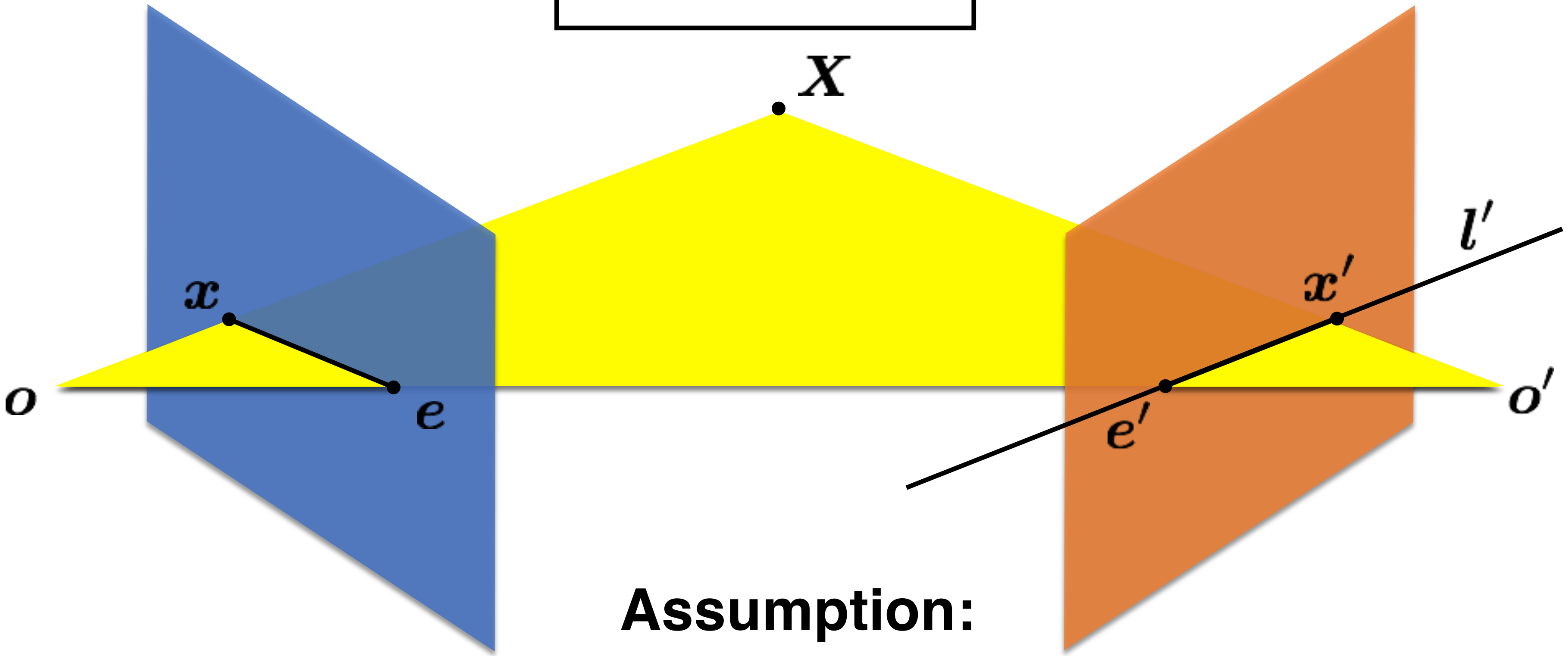
properties of the E matrix

Longuet-Higgins equation	$\boldsymbol{x}'^{\top}\mathbf{E}\boldsymbol{x} = 0$	
Epipolar lines	$\boldsymbol{x}^{\top}\boldsymbol{l} = 0$ $\boldsymbol{l}' = \mathbf{E}\boldsymbol{x}$	$\boldsymbol{x}'^{\top}\boldsymbol{l}' = 0$ $\boldsymbol{l} = \mathbf{E}^{T}\boldsymbol{x}'$
Epipoles	$\boldsymbol{e}'^{\top}\mathbf{E} = \mathbf{0}$	$\mathbf{E}\boldsymbol{e} = \mathbf{0}$

(2D points expressed in camera coordinate system)

Given a point in one image,
multiplying by the **essential matrix** will tell us
the **epipolar line** in the second view.

$$\mathbf{E}\boldsymbol{x} = \boldsymbol{l}'$$



Assumption:
2D points expressed in camera coordinate system
(i.e., intrinsic matrices are identities)

How do you generalize to
non-identity intrinsic matrices?

The fundamental matrix

The
fundamental matrix
is a
generalization
of the
essential matrix,
where the assumption of
Identity matrices
is removed

$$\hat{\boldsymbol{x}}'^{\top} \mathbf{E} \hat{\boldsymbol{x}} = 0$$

The essential matrix operates on image points expressed in
2D coordinates expressed in the camera coordinate system

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

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

camera point

image point

$$\hat{\boldsymbol{x}}'^{\top}\mathbf{E}\hat{\boldsymbol{x}} = 0$$

The essential matrix operates on image points expressed in
2D coordinates expressed in the camera coordinate system

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

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

camera point

image point

Writing out the epipolar constraint in terms of image coordinates

$$\mathbf{K}'^{-\top}\mathbf{E}\mathbf{K}^{-1}\boldsymbol{x} = 0$$

$$\boldsymbol{x}'^{\top}(\mathbf{K}'^{-\top}\mathbf{E}\mathbf{K}^{-1})\boldsymbol{x} = 0$$

$$\boldsymbol{x}'^{\top}\mathbf{F}\boldsymbol{x} = 0$$

Same equation works in image coordinates!

$$\boldsymbol{x}'^{\top}\mathbf{F}\boldsymbol{x} = 0$$

it maps pixels to epipolar lines

properties of the ~~E~~ F matrix

Longuet-Higgins equation	$\boldsymbol{x}'^{\top}\mathbf{F}\boldsymbol{x} = 0$	
Epipolar lines	$\boldsymbol{x}^{\top}\boldsymbol{l} = 0$ $\boldsymbol{l}' = \mathbf{F}\boldsymbol{x}$	$\boldsymbol{x}'^{\top}\boldsymbol{l}' = 0$ $\boldsymbol{l} = \mathbf{F}^{T}\boldsymbol{x}'$
Epipoles	$\boldsymbol{e}'^{\top}\mathbf{F} = \mathbf{0}$	$\mathbf{F}\boldsymbol{e} = \mathbf{0}$

(points in **image** coordinates)

Breaking down the fundamental matrix

$$\mathbf{F} = \mathbf{K}'^{-\top}\mathbf{E}\mathbf{K}^{-1}$$

$$\mathbf{F} = \mathbf{K}'^{-\top}[\mathbf{t}_{\times}]\mathbf{R}\mathbf{K}^{-1}$$

Depends on both intrinsic and extrinsic parameters

Breaking down the fundamental matrix

$$\mathbf{F} = \mathbf{K}'^{-\top}\mathbf{E}\mathbf{K}^{-1}$$

$$\mathbf{F} = \mathbf{K}'^{-\top}[\mathbf{t}_{\times}]\mathbf{R}\mathbf{K}^{-1}$$

Depends on both intrinsic and extrinsic parameters

How would you solve for F?

$$\boldsymbol{x}'^{\top}_m \mathbf{F} \boldsymbol{x}_m = 0$$

The 8-point algorithm

Assume you have ***M*** matched *image* points

$$\{\boldsymbol{x}_m, \boldsymbol{x}'_m\} \qquad m = 1, \ldots, M$$

Each correspondence should satisfy

$$\boldsymbol{x}'^{\top}_m \mathbf{F} \boldsymbol{x}_m = 0$$

How would you solve for the 3 x 3 ***F*** *matrix?*

Assume you have ***M*** matched *image* points

$$\{\boldsymbol{x}_m, \boldsymbol{x}'_m\} \qquad m = 1, \ldots, M$$

Each correspondence should satisfy

$$\boldsymbol{x}'^{\top}_m \mathbf{F} \boldsymbol{x}_m = 0$$

How would you solve for the 3 x 3 ***F*** *matrix?*

S V D

Assume you have ***M*** matched *image* points

$$\{\boldsymbol{x}_m, \boldsymbol{x}'_m\} \qquad m = 1, \ldots, M$$

Each correspondence should satisfy

$$\boldsymbol{x}'^{\top}_m \mathbf{F} \boldsymbol{x}_m = 0$$

How would you solve for the 3 x 3 ***F*** *matrix?*

Set up a homogeneous linear system with 9 unknowns

$$\boldsymbol{x}_m'^\top \mathbf{F} \boldsymbol{x}_m = 0$$

$$\begin{bmatrix} x_m' & y_m' & 1 \end{bmatrix} \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \begin{bmatrix} x_m \\ y_m \\ 1 \end{bmatrix} = 0$$

How many equation do you get from one correspondence?

$$\begin{bmatrix} x'_m & y'_m & 1 \end{bmatrix} \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \begin{bmatrix} x_m \\ y_m \\ 1 \end{bmatrix} = 0$$

ONE correspondence gives you ONE equation

$$x_m x'_m f_1 + x_m y'_m f_2 + x_m f_3 +$$
$$y_m x'_m f_4 + y_m y'_m f_5 + y_m f_6 +$$
$$x'_m f_7 + y'_m f_8 + f_9 = 0$$

$$\begin{bmatrix} x'_m & y'_m & 1 \end{bmatrix} \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \begin{bmatrix} x_m \\ y_m \\ 1 \end{bmatrix} = 0$$

Set up a homogeneous linear system with 9 unknowns

$$\begin{bmatrix} x_1x'_1 & x_1y'_1 & x_1 & y_1x'_1 & y_1y'_1 & y_1 & x'_1 & y'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_Mx'_M & x_My'_M & x_M & y_Mx'_M & y_My'_M & y_M & x'_M & y'_M & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix} = \mathbf{0}$$

How many equations do you need?

Each point pair (according to epipolar constraint)
contributes only one scalar equation

$$\boldsymbol{x}'^{\top}_m \mathbf{F} \boldsymbol{x}_m = 0$$

Note: This is different from the Homography estimation
where each point pair contributes 2 equations.

We need at least 8 points

Hence, the 8 point algorithm!

How do you solve a homogeneous linear system?

$$\mathbf{A}\boldsymbol{X} = \mathbf{0}$$

How do you solve a homogeneous linear system?

$$\mathbf{AX} = \mathbf{0}$$

Total Least Squares

minimize $\|\mathbf{A}\boldsymbol{x}\|^2$

subject to $\|\boldsymbol{x}\|^2 = 1$

How do you solve a homogeneous linear system?

$$\mathbf{A}\mathbf{X} = \mathbf{0}$$

Total Least Squares

minimize $\|\mathbf{A}\boldsymbol{x}\|^2$

subject to $\|\boldsymbol{x}\|^2 = 1$

S V D !

Eight-Point Algorithm

0. (Normalize points)
1. Construct the M x 9 matrix **A**
2. Find the SVD of **A**
3. Entries of **F** are the elements of column of **V** corresponding to the least singular value
4. (Enforce rank 2 constraint on F)
5. (Un-normalize F)

Eight-Point Algorithm

0. (Normalize points)
1. Construct the M x 9 matrix **A**
2. Find the SVD of **A**
3. Entries of **F** are the elements of column of **V** corresponding to the least singular value
4. (Enforce rank 2 constraint on F)
5. (Un-normalize F)

See Hartley-Zisserman for why we do this

Eight-Point Algorithm

0. (Normalize points)
1. Construct the M x 9 matrix **A**
2. Find the SVD of **A**
3. Entries of **F** are the elements of column of **V** corresponding to the least singular value
4. (Enforce rank 2 constraint on F)
5. (Un-normalize F)

How do we do this?

Eight-Point Algorithm

0. (Normalize points)
1. Construct the M x 9 matrix **A**
2. Find the SVD of **A**
3. Entries of **F** are the elements of column of **V** corresponding to the least singular value
4. (Enforce rank 2 constraint on F)
5. (Un-normalize F)

How do we do this?

S V D !

Enforcing rank constraints

Problem: Given a matrix F, find the matrix F' of rank k that is closest to F,

$$\min_{\substack{F' \\ \mathrm{rank}(F')=k}} \quad ||F - F'||^2$$

Solution: Compute the singular value decomposition of F,

$$F = U\Sigma V^T$$

Form a matrix Σ' by replacing all but the k largest singular values in Σ with 0.

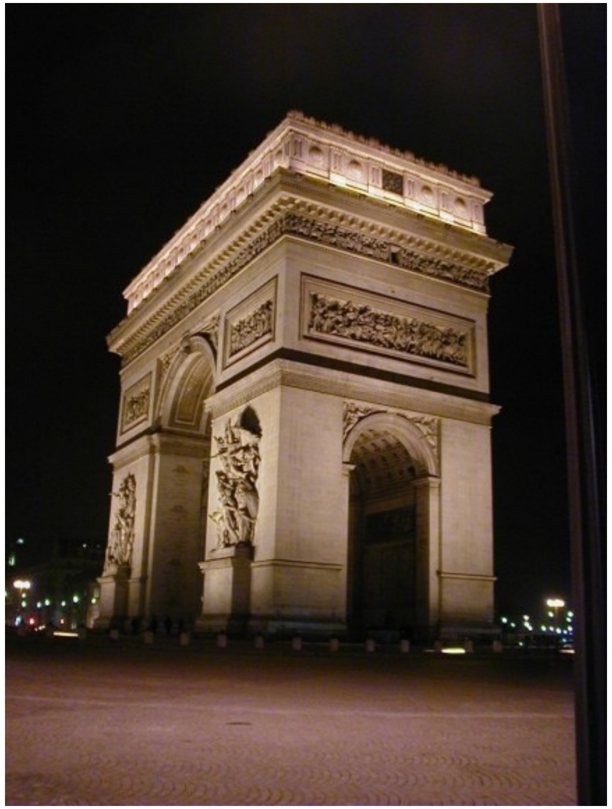
Then the problem solution is the matrix F' formed as,

$$F' = U\Sigma' V^T$$

Eight-Point Algorithm

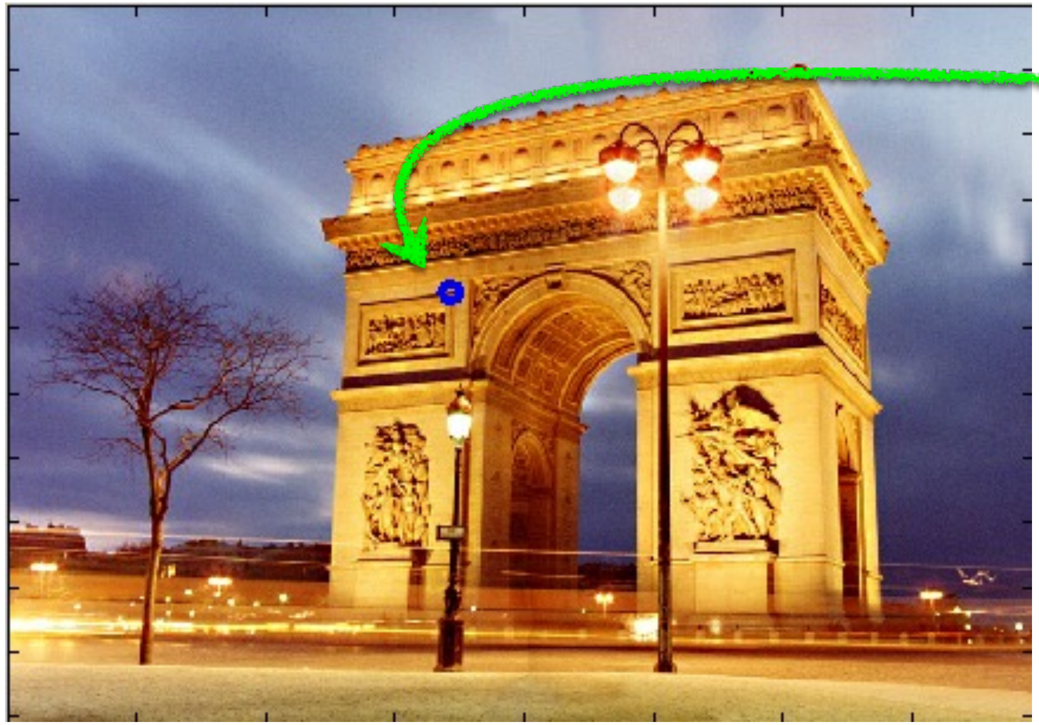
0. (Normalize points)
1. Construct the M x 9 matrix **A**
2. Find the SVD of **A**
3. Entries of **F** are the elements of column of **V** corresponding to the least singular value
4. (Enforce rank 2 constraint on F)
5. (Un-normalize F)

Example



epipolar lines

$$\mathbf{F} = \begin{bmatrix} -0.00310695 & -0.0025646 & 2.96584 \\ -0.028094 & -0.00771621 & 56.3813 \\ 13.1905 & -29.2007 & -9999.79 \end{bmatrix}$$



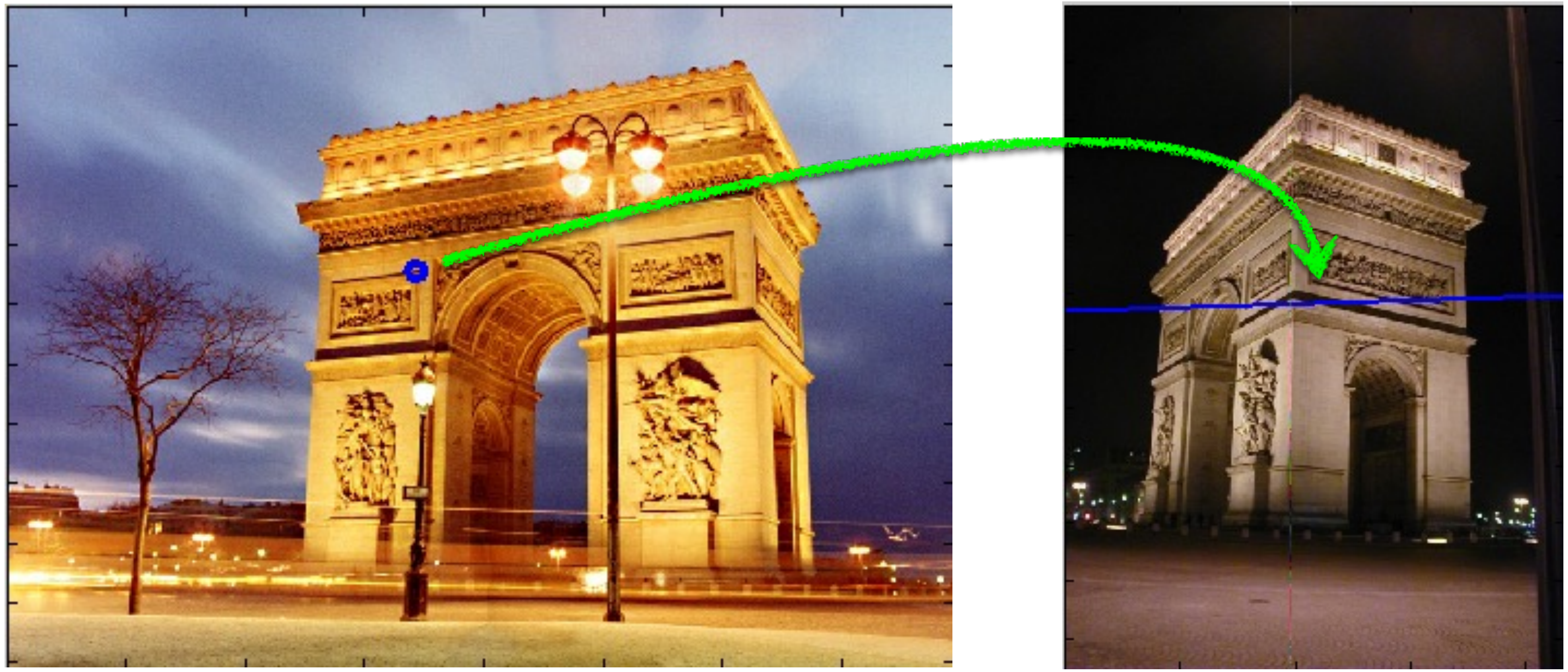
$$\boldsymbol{x} = \begin{bmatrix} 343.53 \\ 221.70 \\ 1.0 \end{bmatrix}$$

$$\boldsymbol{l}' = \mathbf{F}\boldsymbol{x}$$

$$= \begin{bmatrix} 0.0295 \\ 0.9996 \\ -265.1531 \end{bmatrix}$$

$$\boldsymbol{l}' = \mathbf{F}\boldsymbol{x}$$

$$= \begin{bmatrix} 0.0295 \\ 0.9996 \\ -265.1531 \end{bmatrix}$$



Where is the epipole?

How would you compute it?

$$\mathbf{F}e = \mathbf{0}$$

The epipole is in the right null space of **F**

How would you solve for the epipole?

$$\mathbf{F}\boldsymbol{e} = \mathbf{0}$$

The epipole is in the right null space of **F**

How would you solve for the epipole?

S V D !

Next Time:
Stereo depth estimation