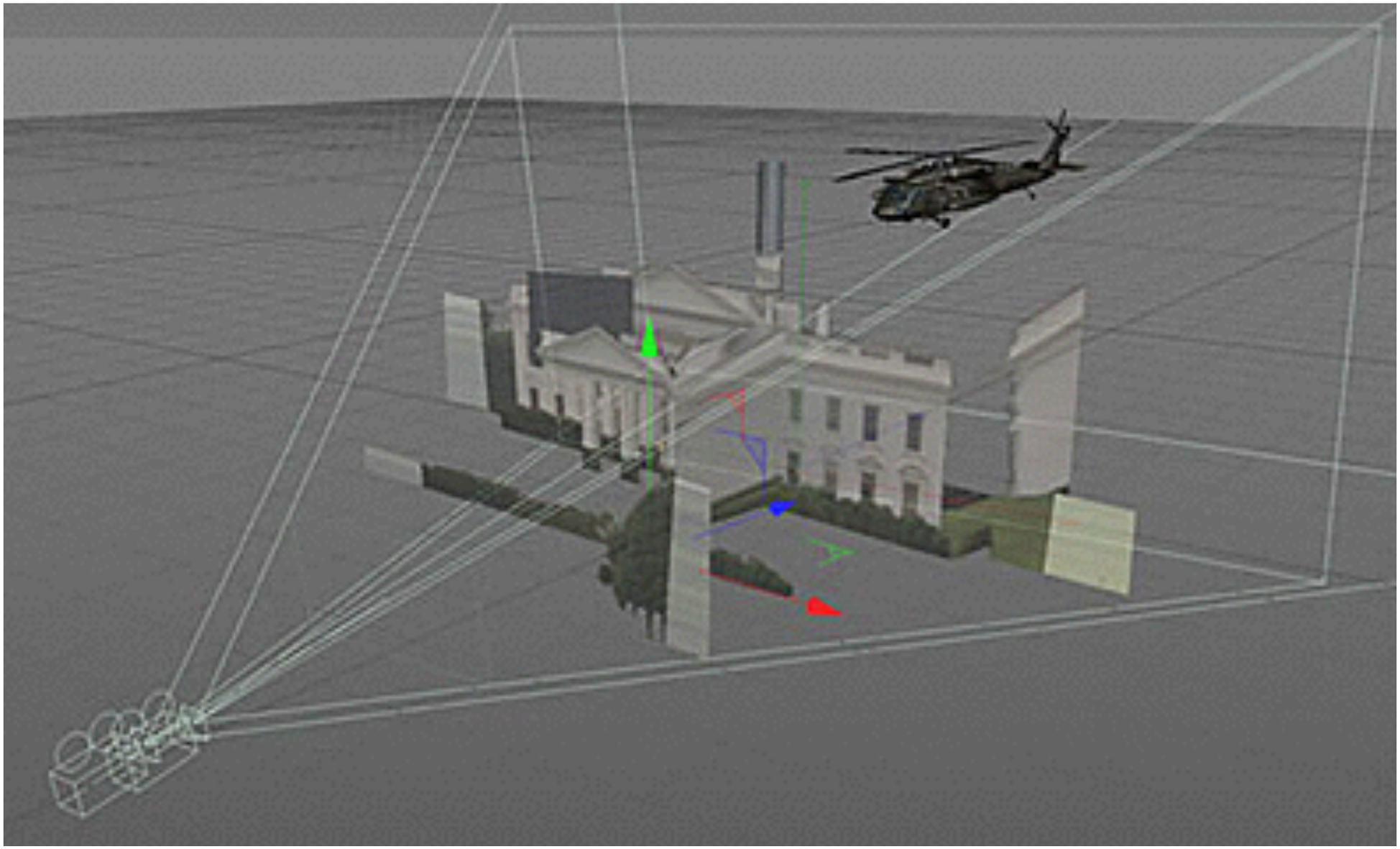


Camera Matrix

16-385 Computer Vision (Kris Kitani)

Carnegie Mellon University

A camera is a mapping between

the **3D world**

and

a **2D image**

A camera is a mapping between
the 3D world and a 2D image

$$x = PX$$

2D image point

camera matrix

3D world point

What do you think the dimensions are?

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

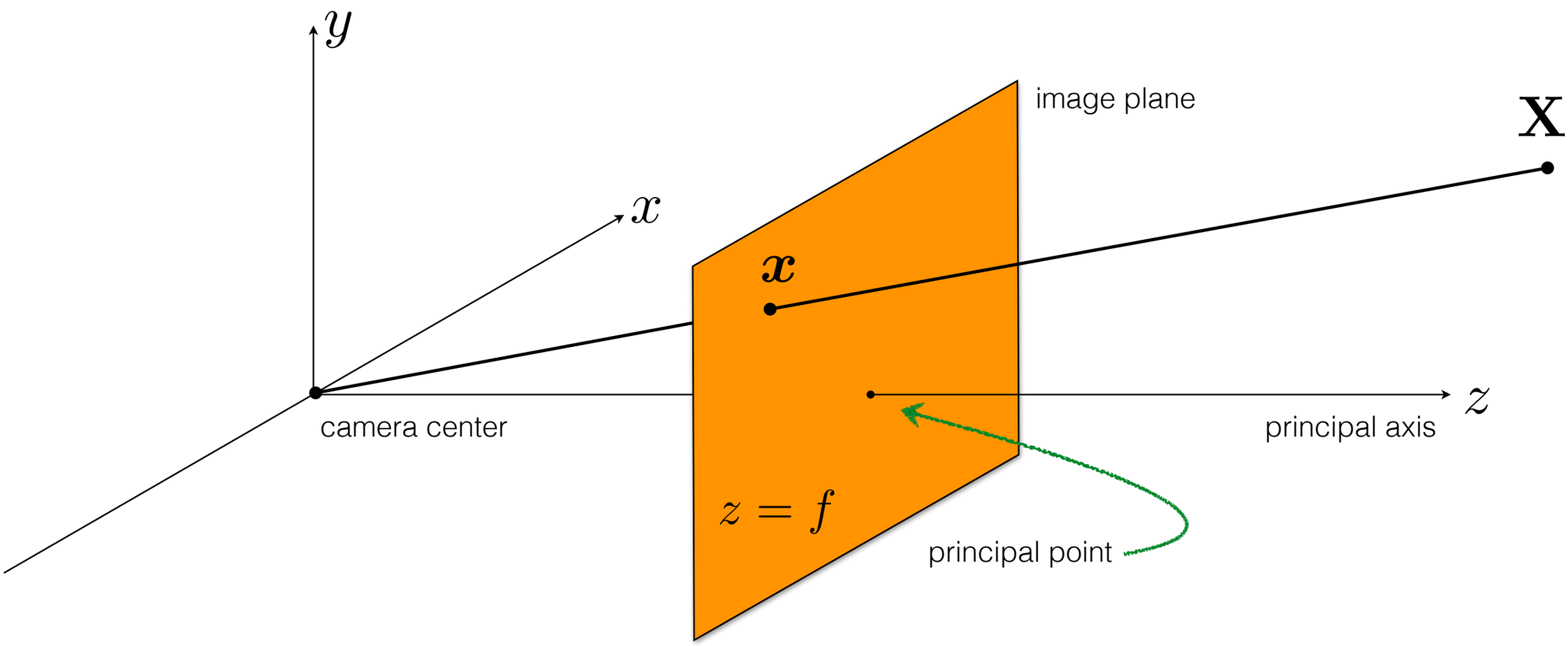
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

homogeneous image 3 x 1

Camera matrix 3 x 4

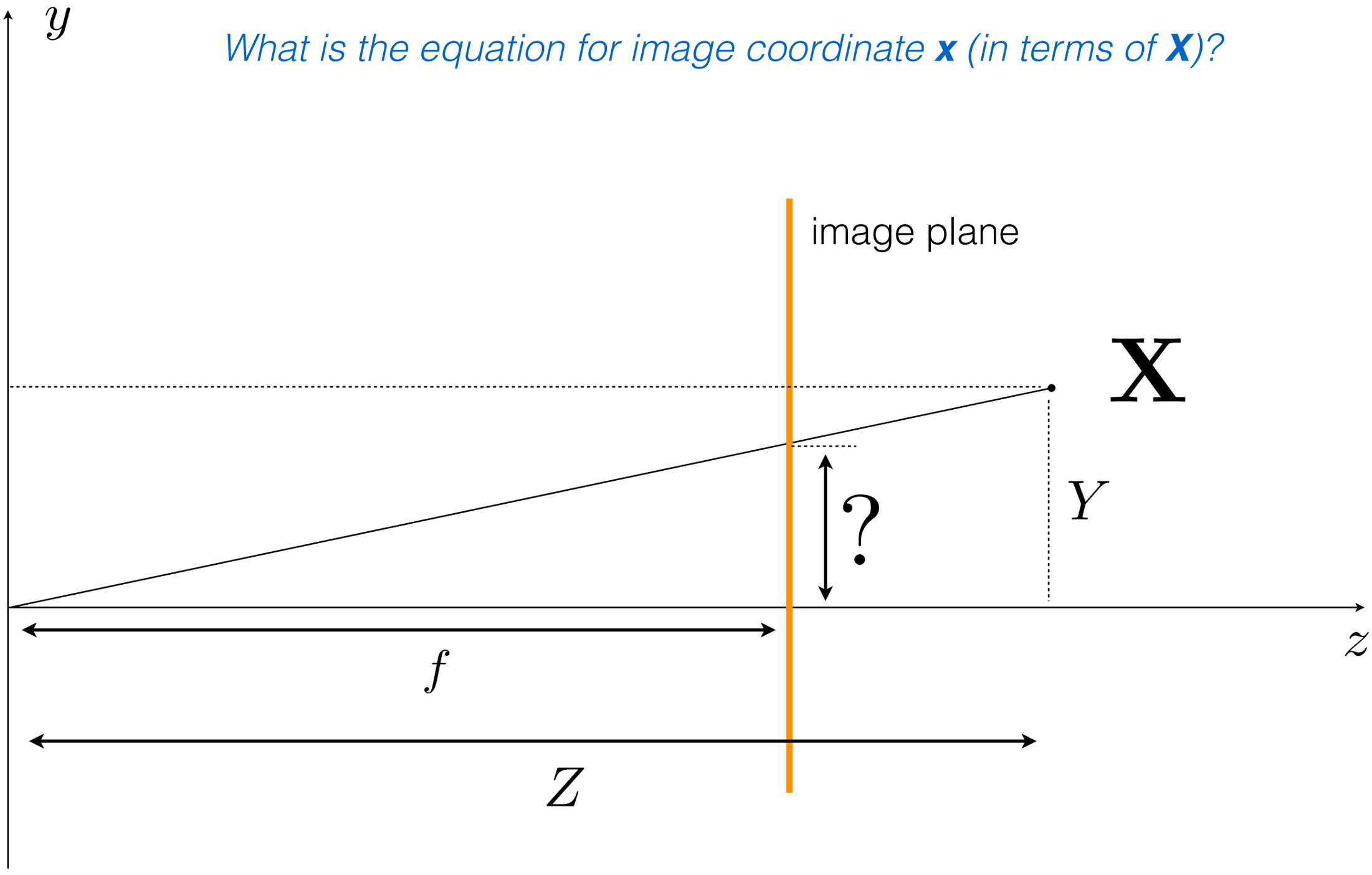
homogeneous world point 4 x 1

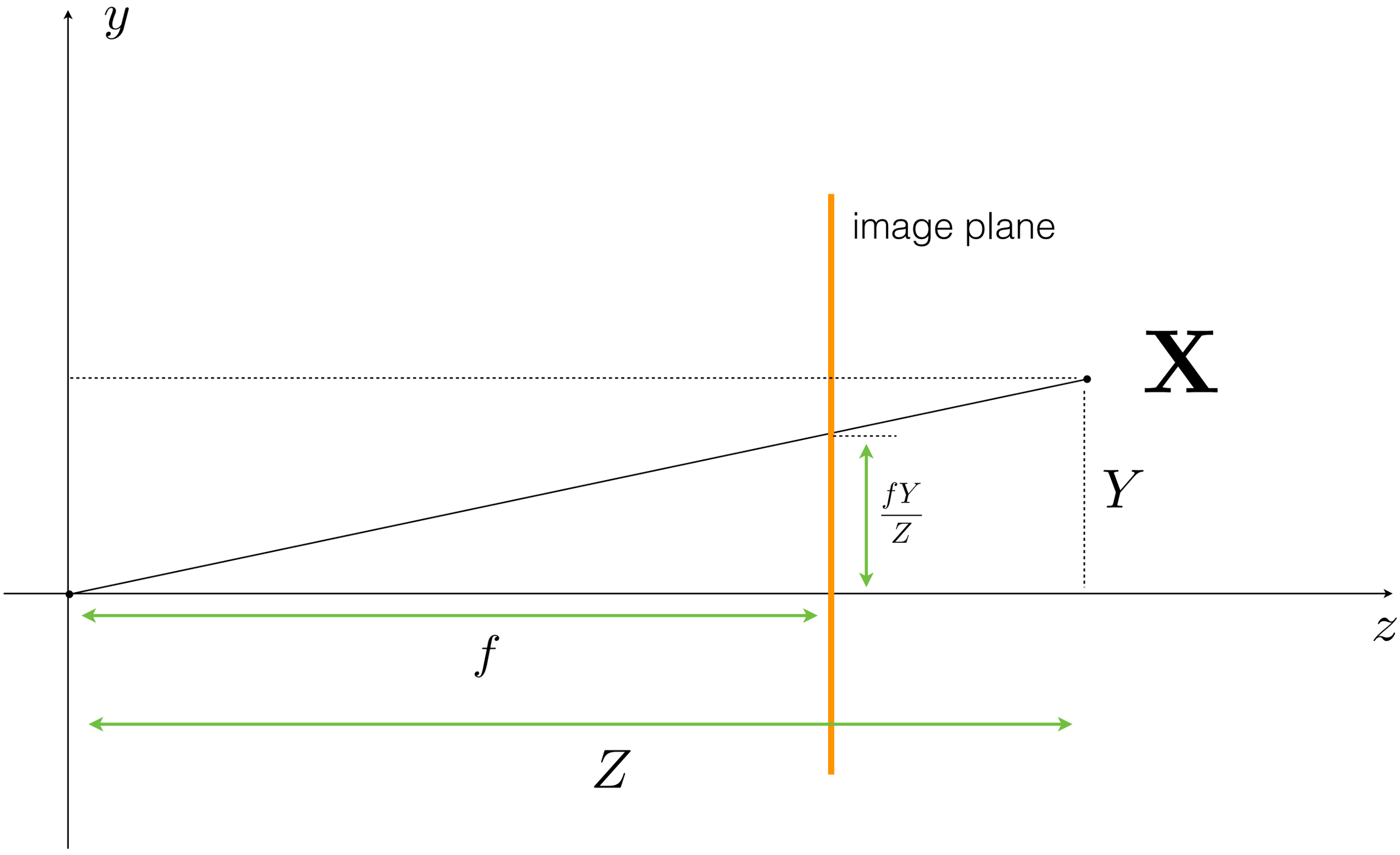
The pinhole camera



*What is the equation for image coordinate **x** (in terms of **X**)?*

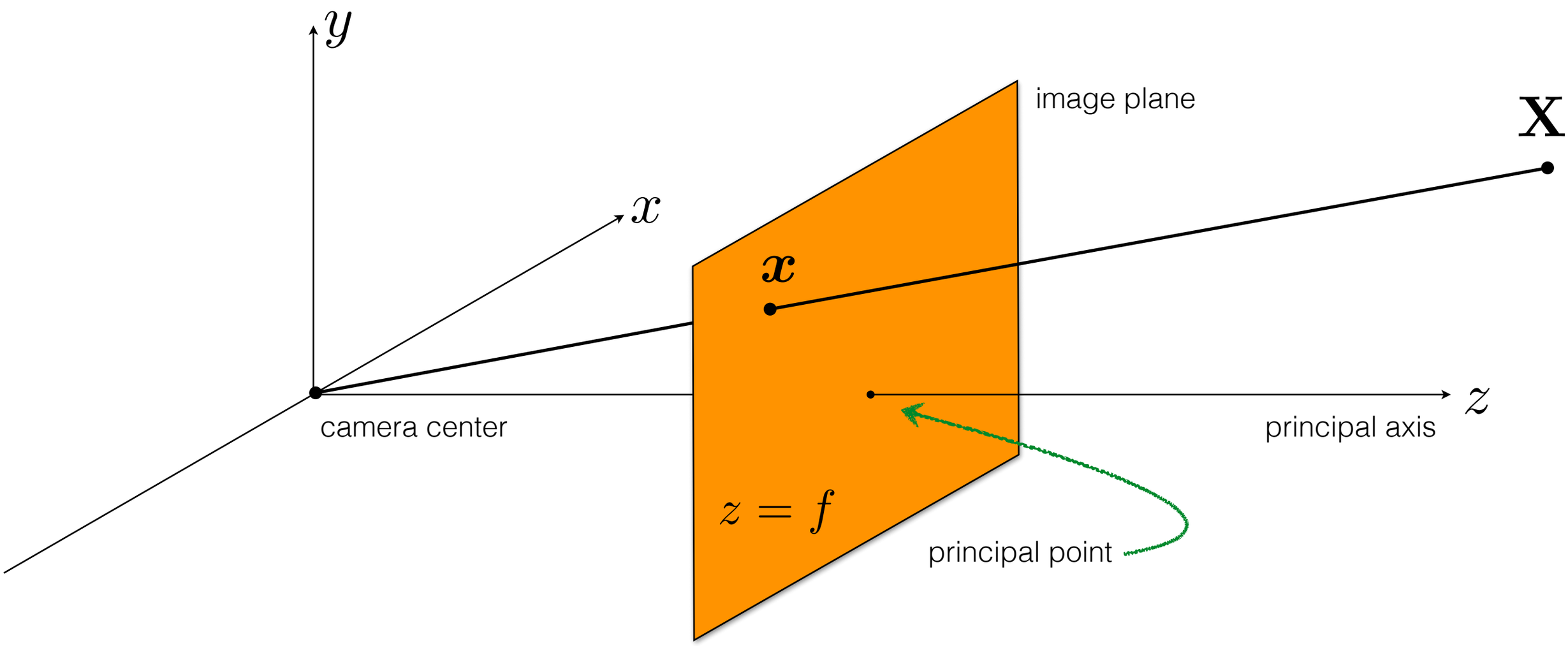
*What is the equation for image coordinate **x** (in terms of **X**)?*





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

Pinhole camera geometry



What is the camera matrix **P** *for a pinhole camera model?*

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

Relationship from similar triangles…

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

generic camera model

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

What does the pinhole camera model look like?

$$\mathbf{P} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix}$$

Relationship from similar triangles…

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

generic camera model

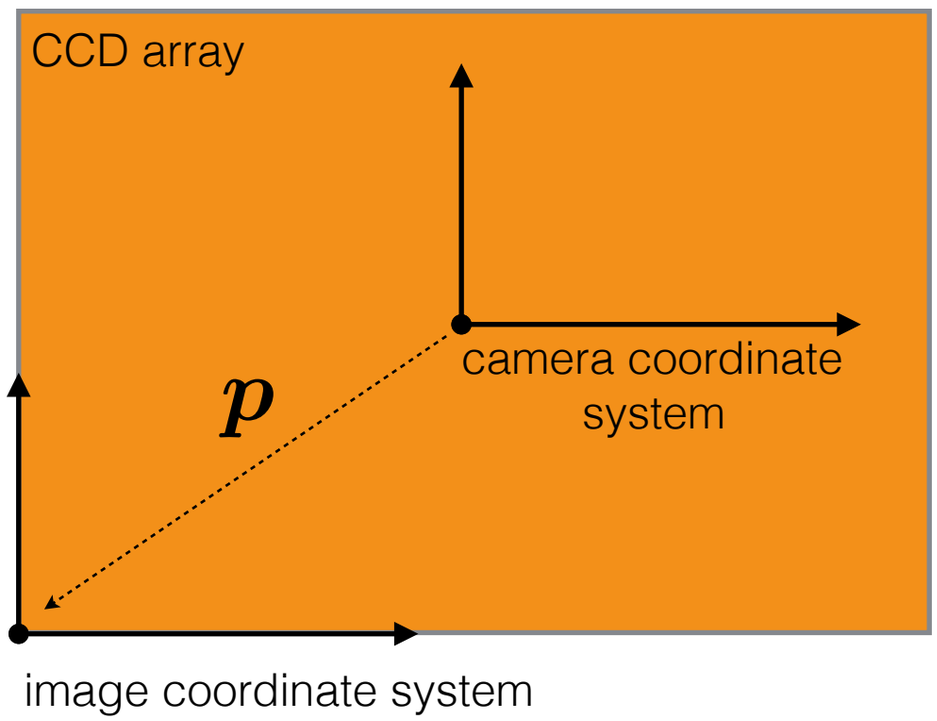
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

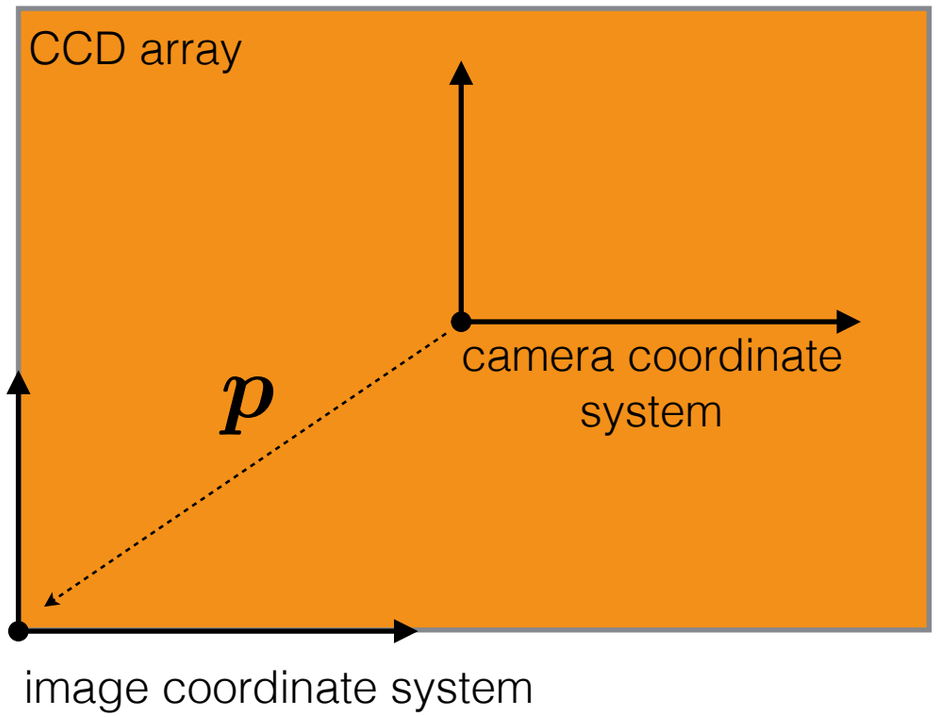
What does the pinhole camera model look like?

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

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

Camera origin and image origin might be different

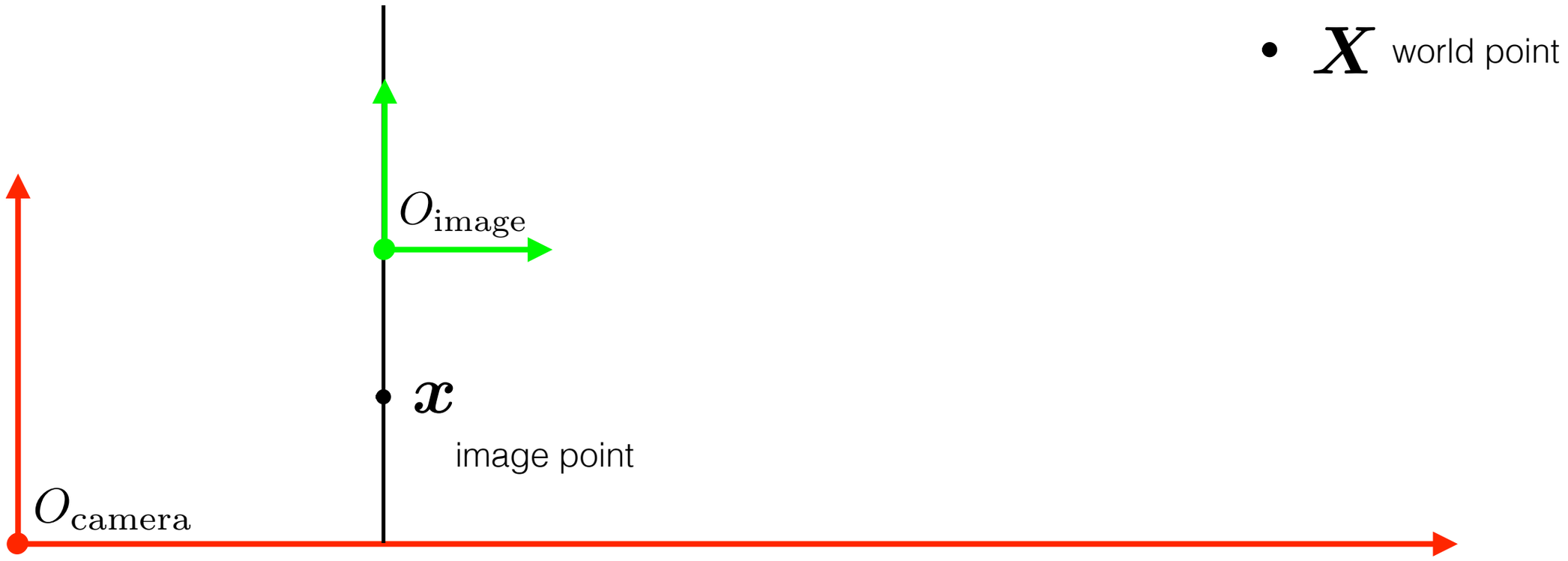




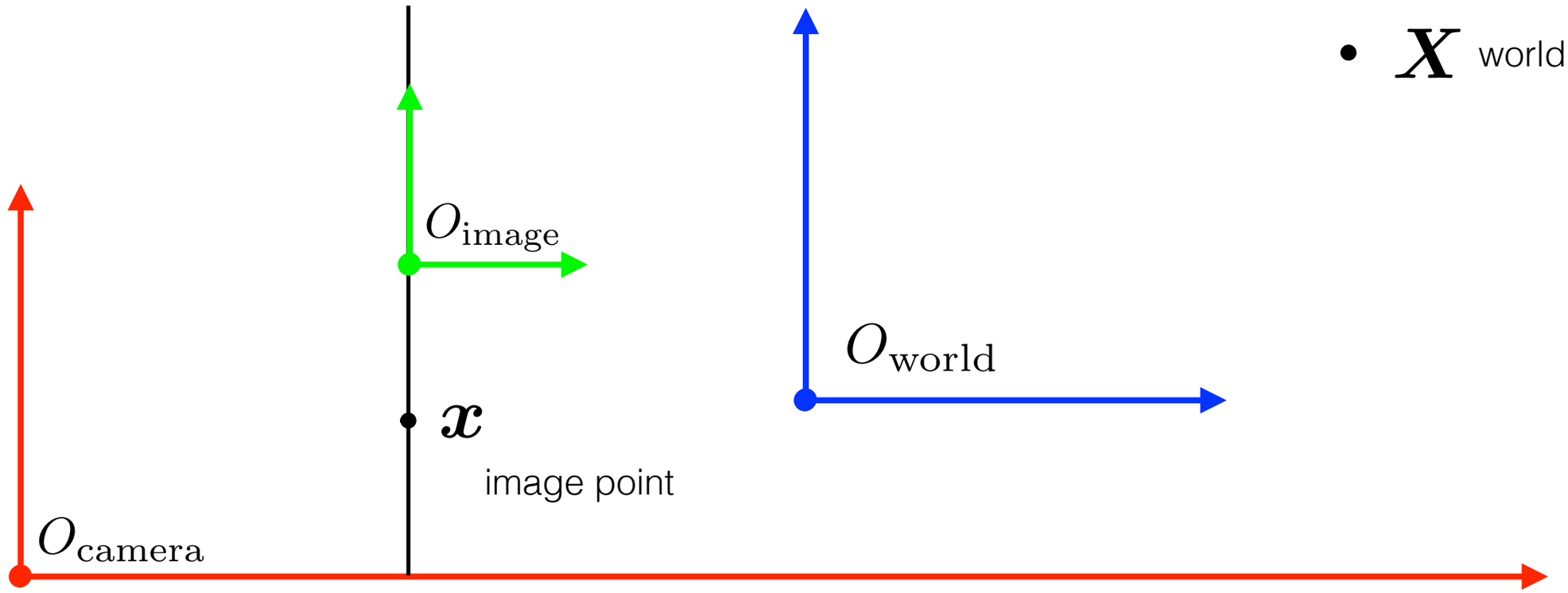
$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Accounts for different origins

In general, the camera and image sensor have **different** coordinate systems



In general, there are **three different** coordinate systems…



so you need the know the transformations between them

Can be decomposed into two matrices

$$\mathbf{P} = \underbrace{\begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}}_{(3 \times 3)} \underbrace{\left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right]}_{(3 \times 4)}$$

$$\mathbf{P} = \mathbf{K}[\mathbf{I}|\mathbf{0}]$$

$$\mathbf{K} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

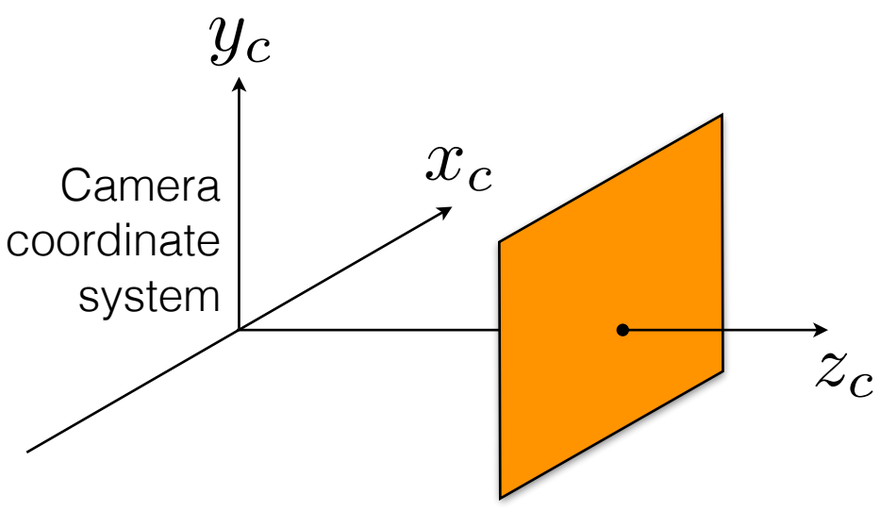
calibration matrix

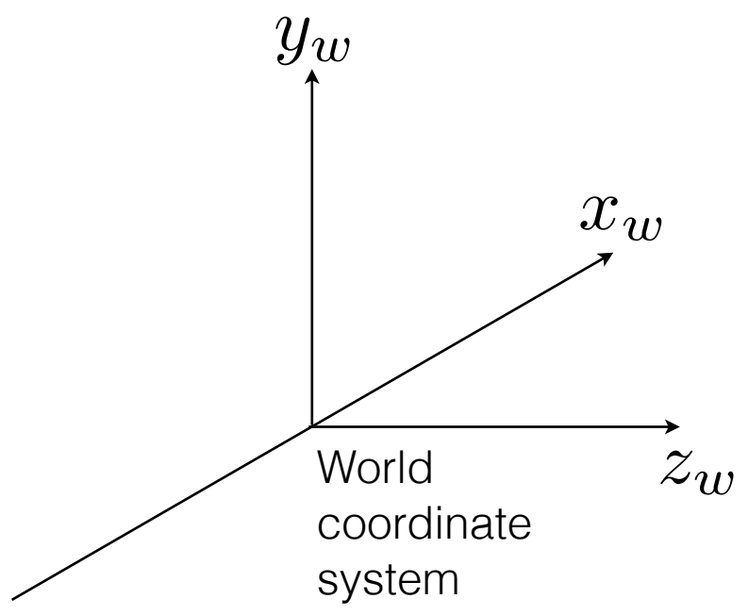
Assumes that the **camera** and **world** share the same coordinate system

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

What if they are different?

How do we align them?



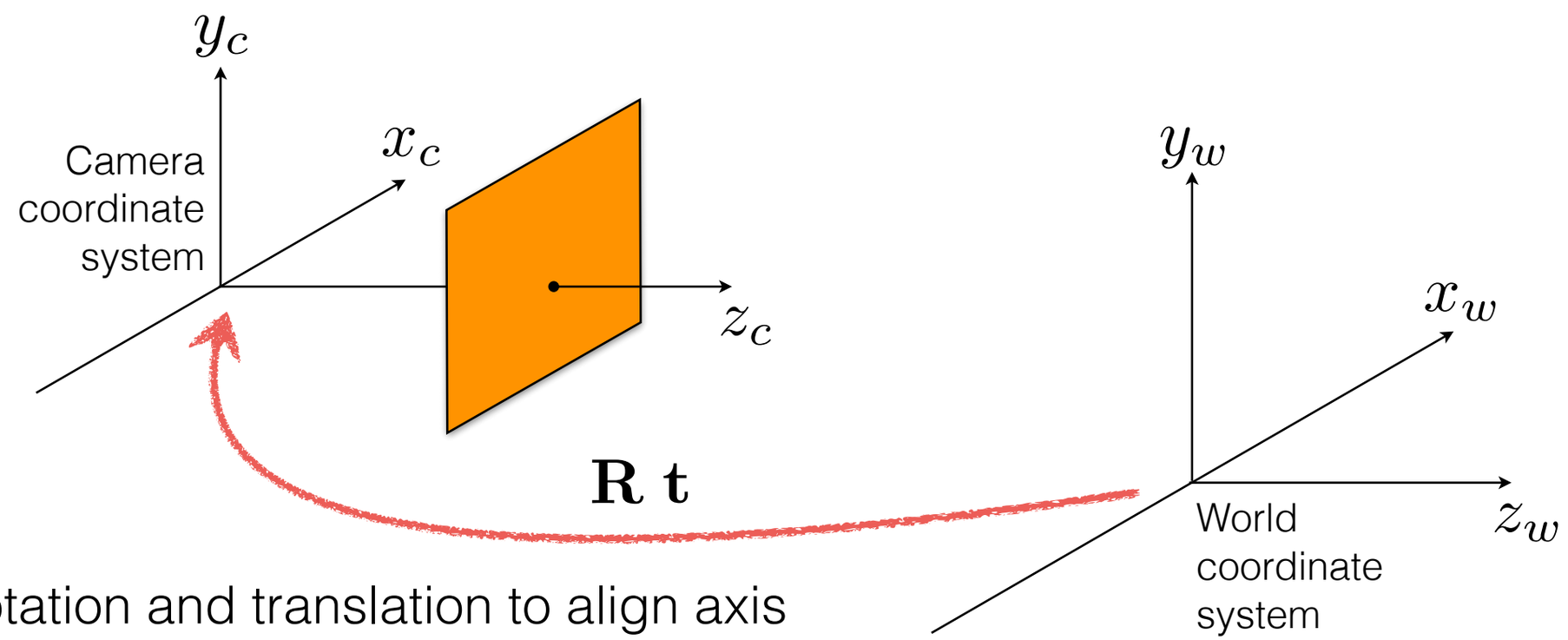


Assumes that the camera and world share the same coordinate system

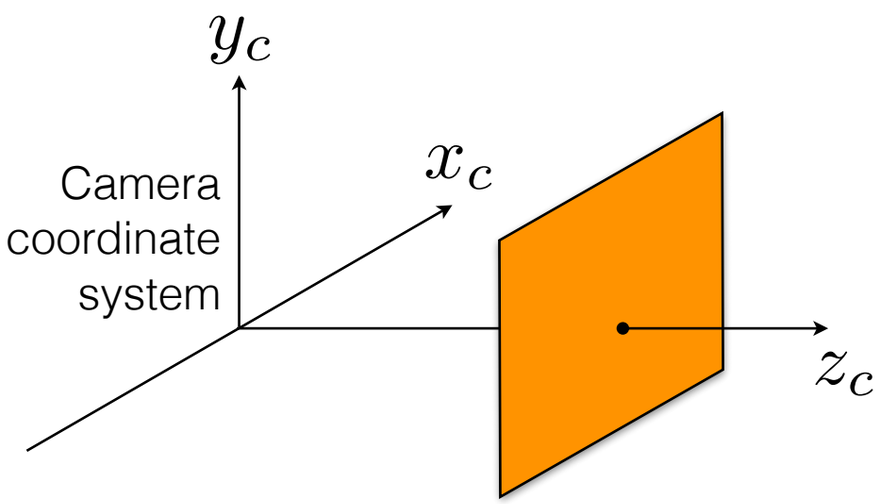
$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

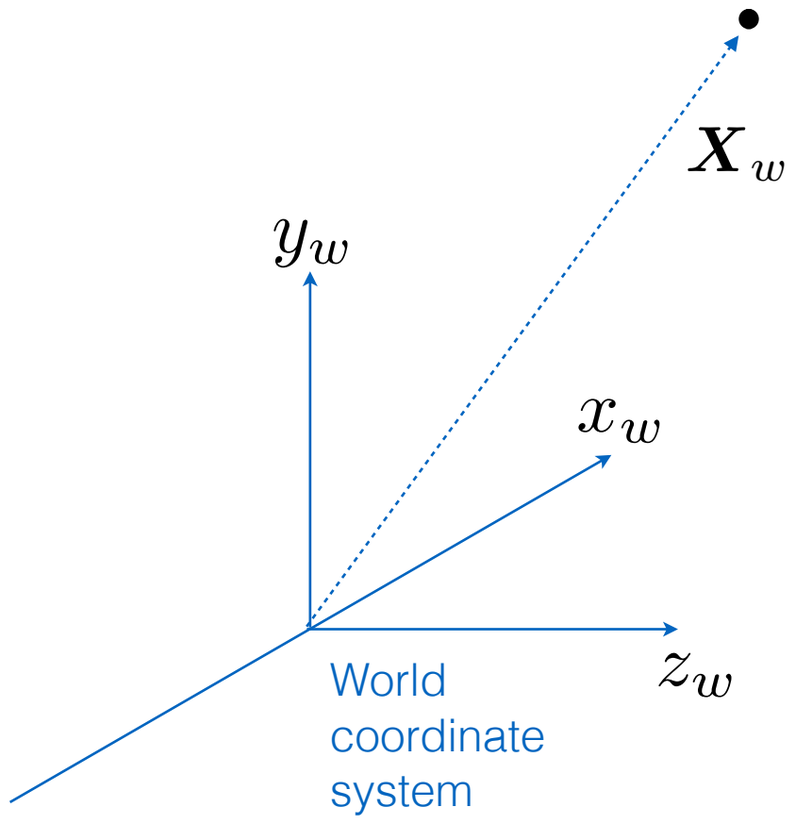
What if they are different?

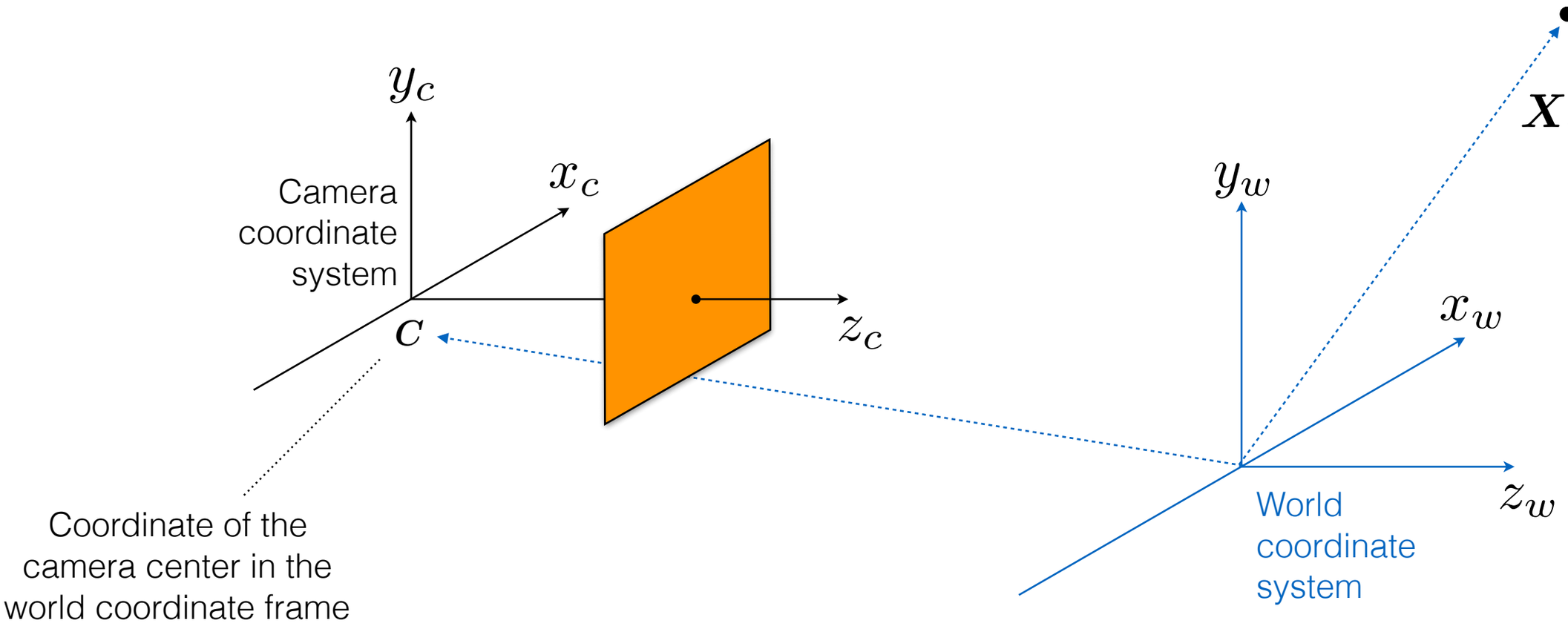
How do we align them?

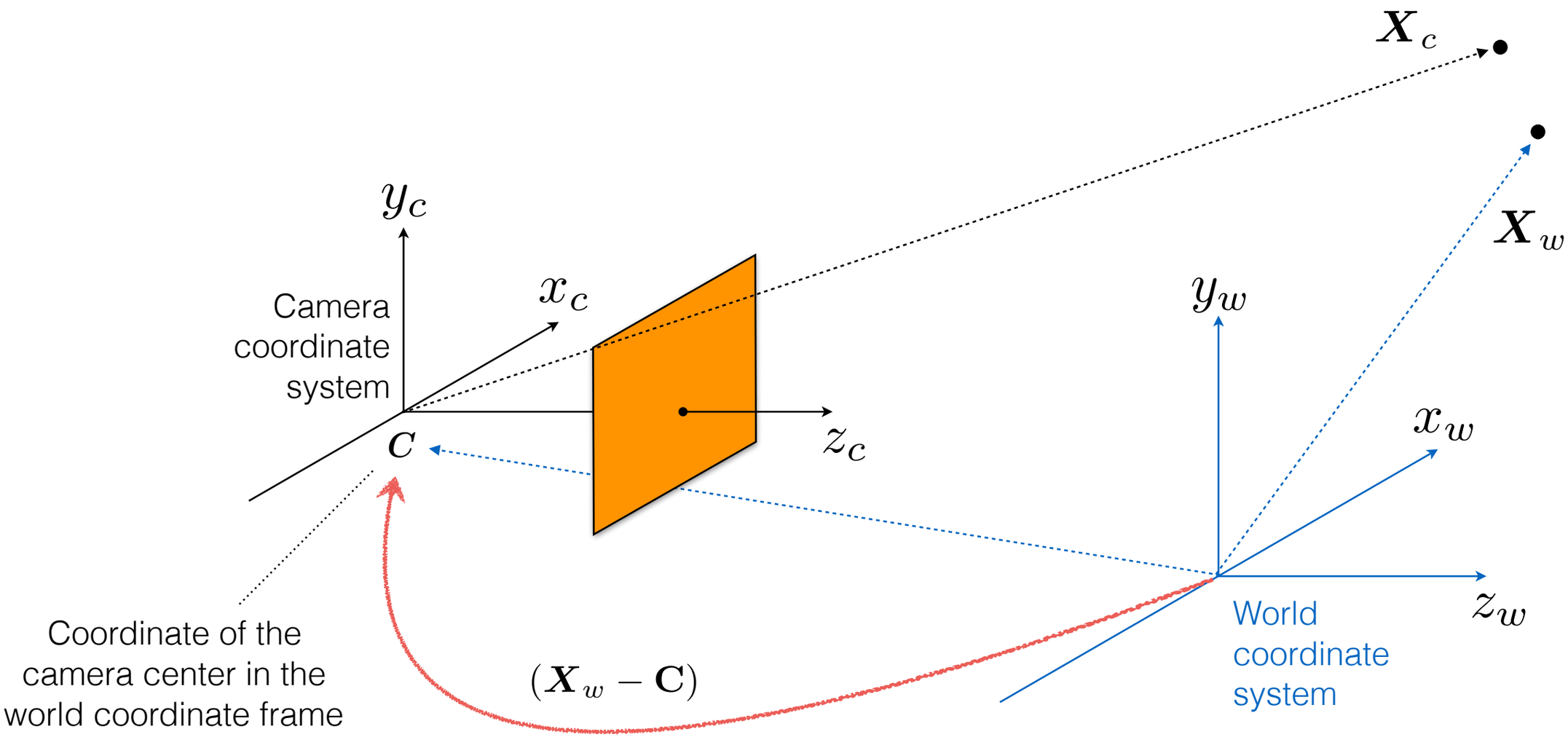


3R rotation and translation to align axis


y_c
x_c
Camera
coordinate
system
z_c

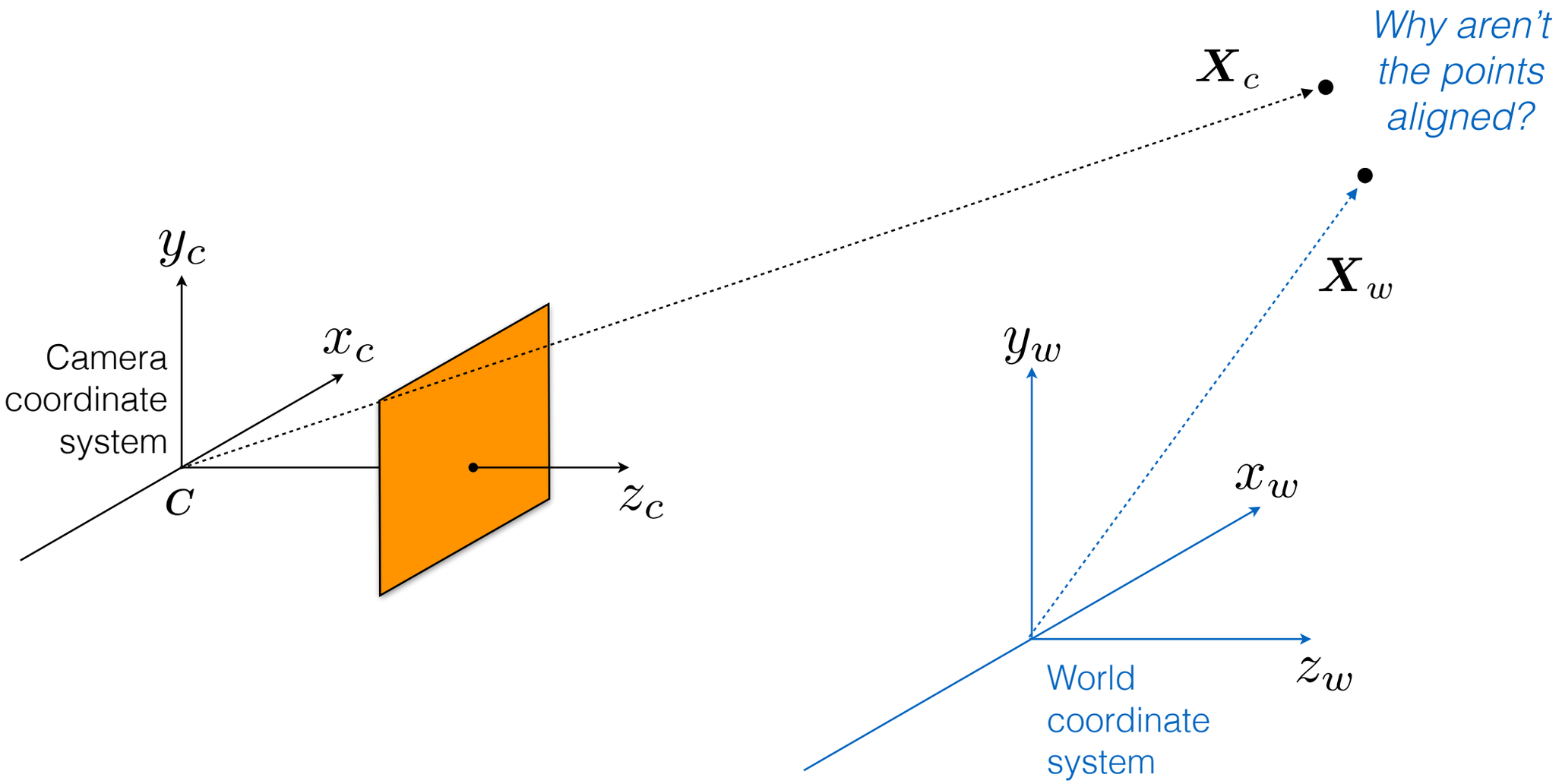

$\boldsymbol{X}_w$
y_w
x_w
z_w
World
coordinate
system


y_c
x_c
Camera
coordinate
system
C
z_c
Coordinate of the
camera center in the
world coordinate frame
X_w
y_w
x_w
z_w
World
coordinate
system



$$(\boldsymbol{X}_w - \mathbf{C})$$

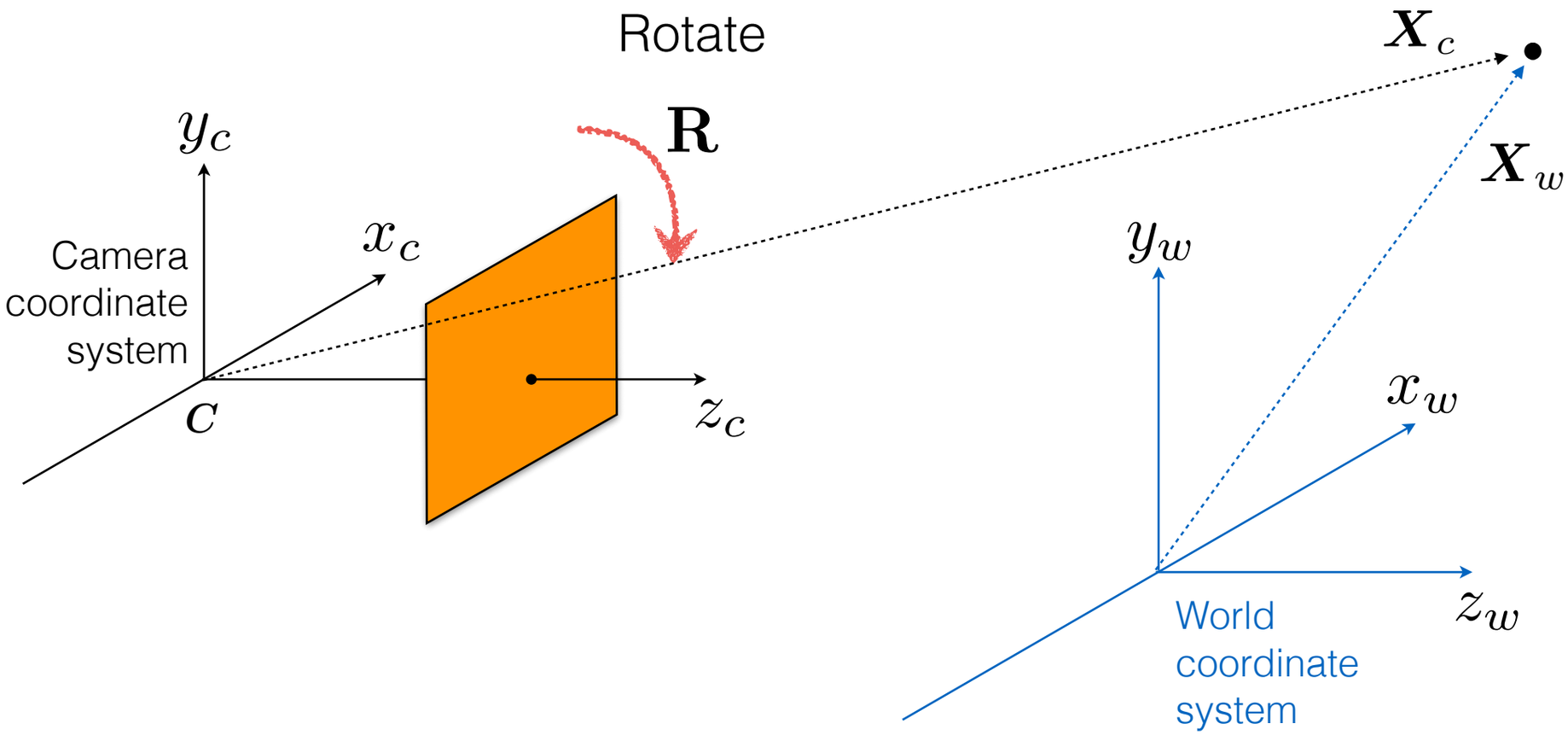
Translate



$$(\boldsymbol{X}_w - \mathbf{C})$$

Translate

What happens to points after alignment?



$$\mathbf{R}(\boldsymbol{X}_w - \mathbf{C})$$

Rotate Translate

In inhomogeneous coordinates:

$$\boldsymbol{X}_c = \mathbf{R}(\boldsymbol{X}_w - \mathbf{C})$$

Optionally in homogeneous coordinates:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{R} & -\mathbf{RC} \\ \mathbf{0} & 1 \end{bmatrix}}_{(4\times 4)} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

General mapping of a pinhole camera

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

Quiz

What is the meaning of each matrix of the camera matrix decomposition?

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

3x3
intrinsics

Quiz

What is the meaning of each matrix of the camera matrix decomposition?

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

3x3
intrinsics

3x3
3D rotation

Quiz

What is the meaning of each matrix of the camera matrix decomposition?

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

3x3 intrinsics

3x3 3D rotation

3x3 identity

Quiz

What is the meaning of each matrix of the camera matrix decomposition?

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

3x3 intrinsics

3x3 3D rotation

3x3 identity

3x1 3D translation

General mapping of a pinhole camera

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

(translate first then rotate)

Another way to write the mapping

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

where

$$\mathbf{t} = -\mathbf{R}\mathbf{C}$$

(rotate first then translate)

Quiz

The camera matrix relates what two quantities?

Quiz

The camera matrix relates what two quantities?

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

Quiz

The camera matrix relates what two quantities?

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

3D points to 2D image points

Quiz

The camera matrix relates what two quantities?

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

3D points to 2D image points

The camera matrix can be decomposed into?

Quiz

The camera matrix relates what two quantities?

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

3D points to 2D image points

The camera matrix can be decomposed into?

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

Quiz

The camera matrix relates what two quantities?

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

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

Generalized pinhole camera model

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

$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \end{bmatrix}$$

intrinsic parameters

extrinsic parameters

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

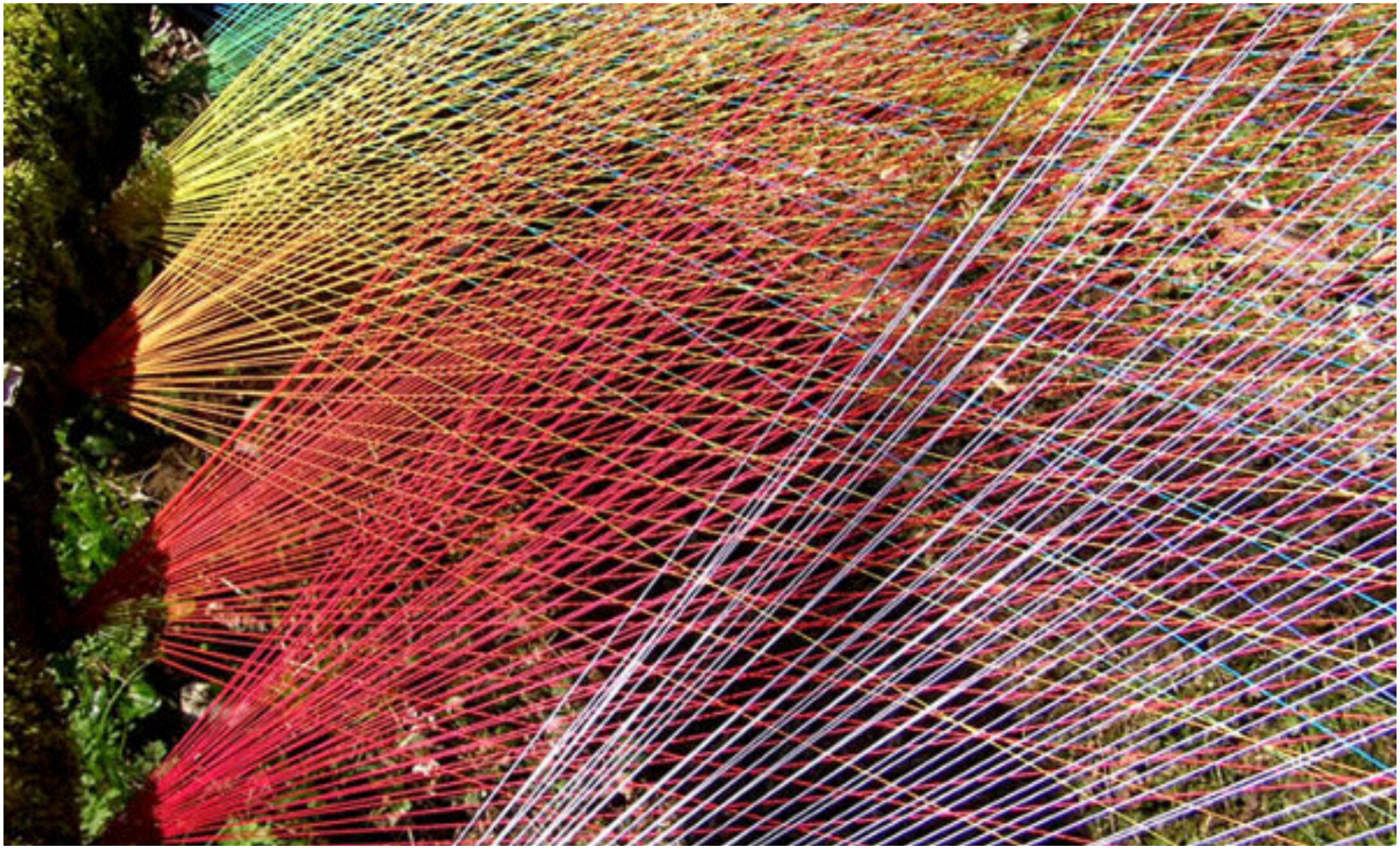
3D rotation

$$\mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

3D translation

*Why do we need **P**?*

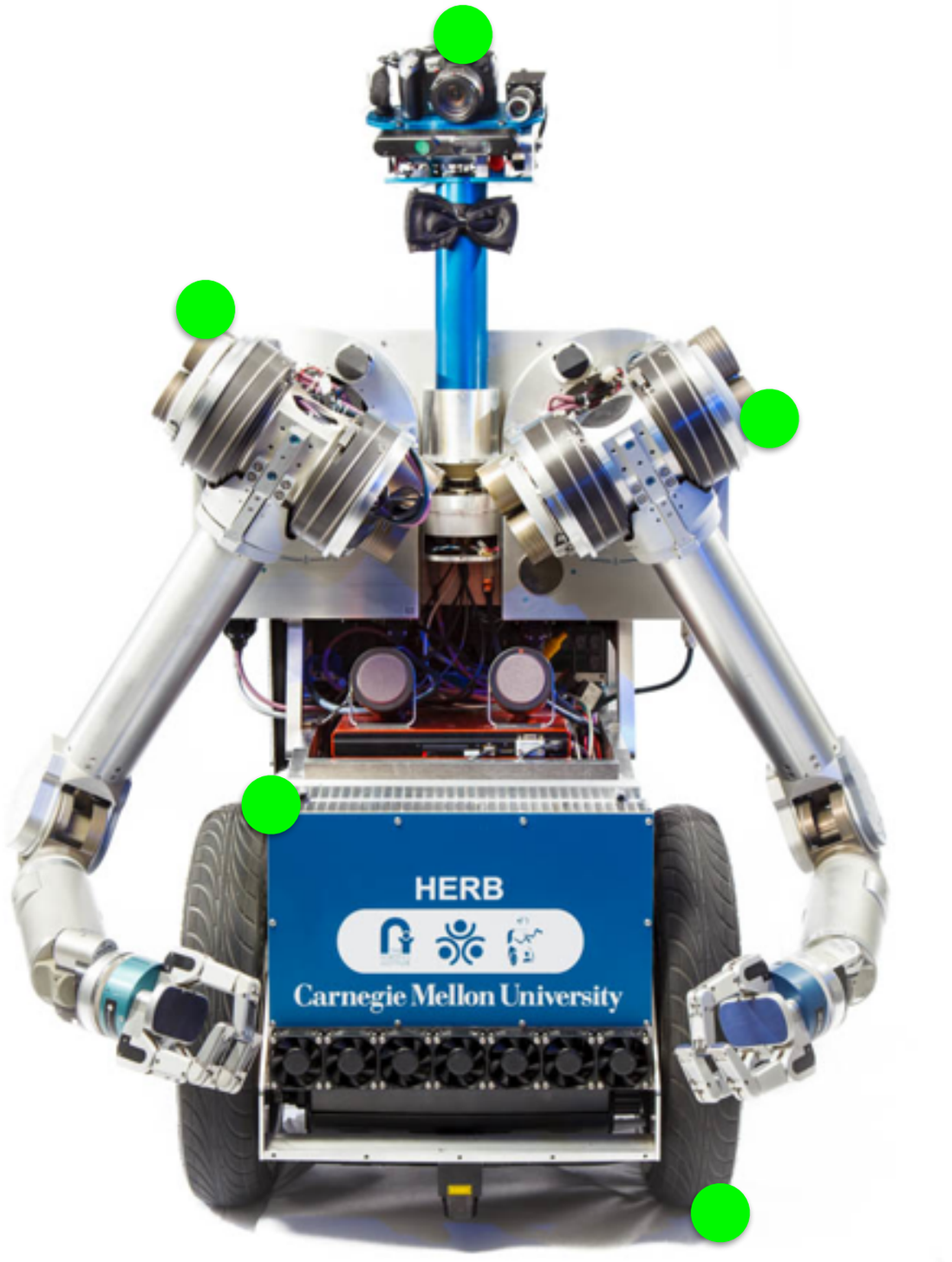
to properly relate **world points** to **image points**
(by taking into account different coordinate systems)



Epipolar Geometry

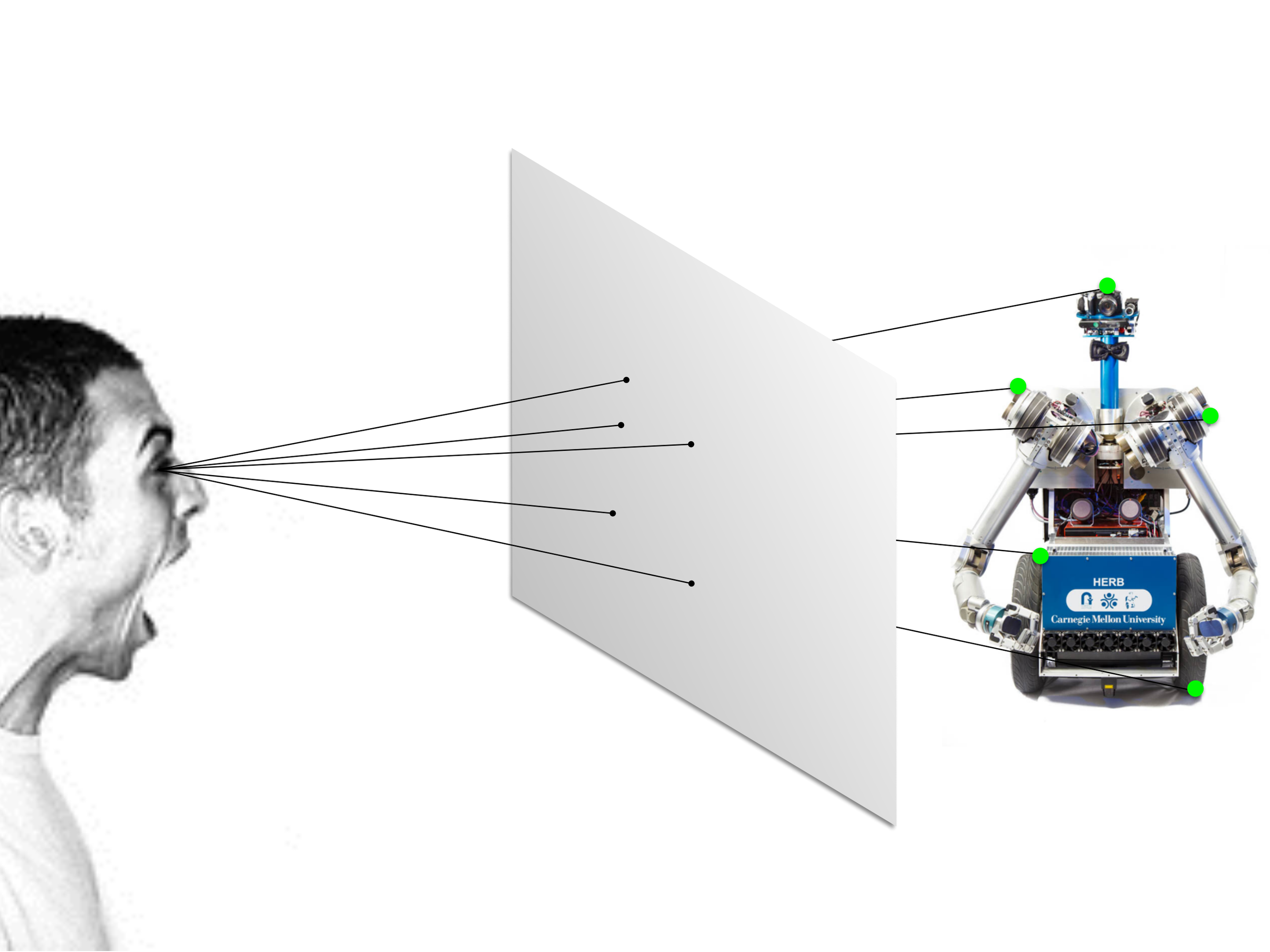
16-385 Computer Vision (Kris Kitani)

Carnegie Mellon University

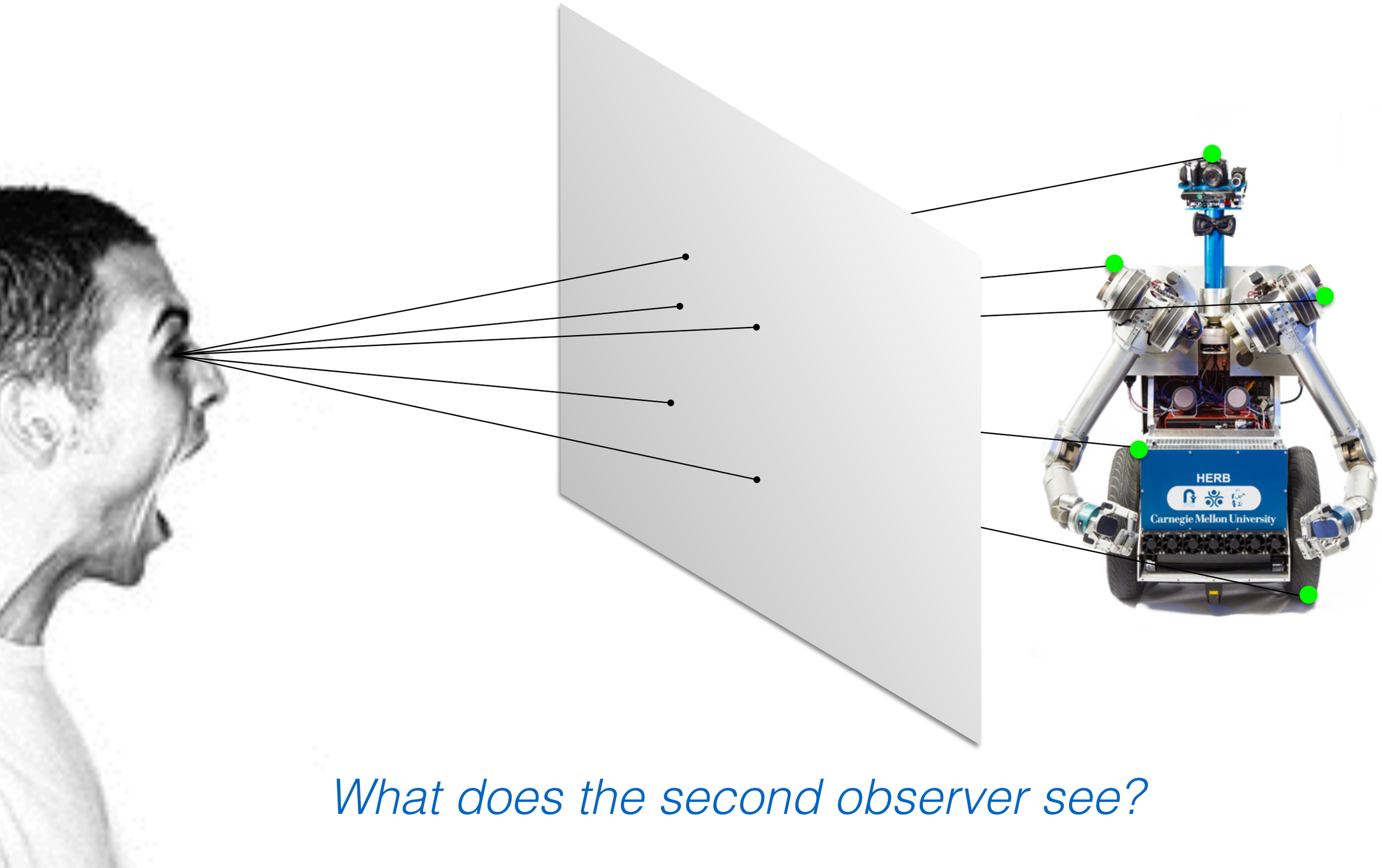


Tie tiny threads on HERB and pin them to your eyeball

What would it look like?

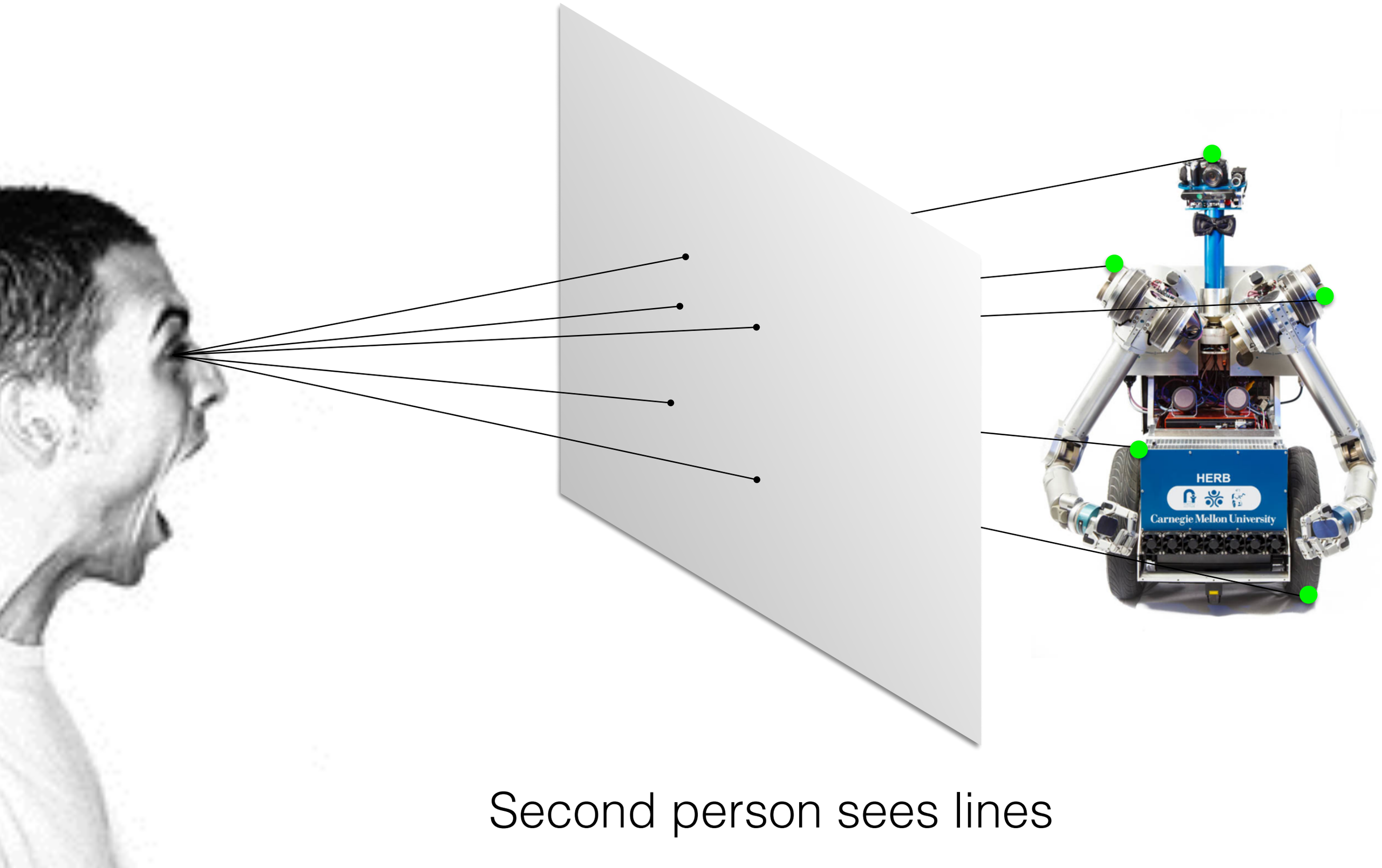

HERB
Carnegie Mellon University

You see points on HERB



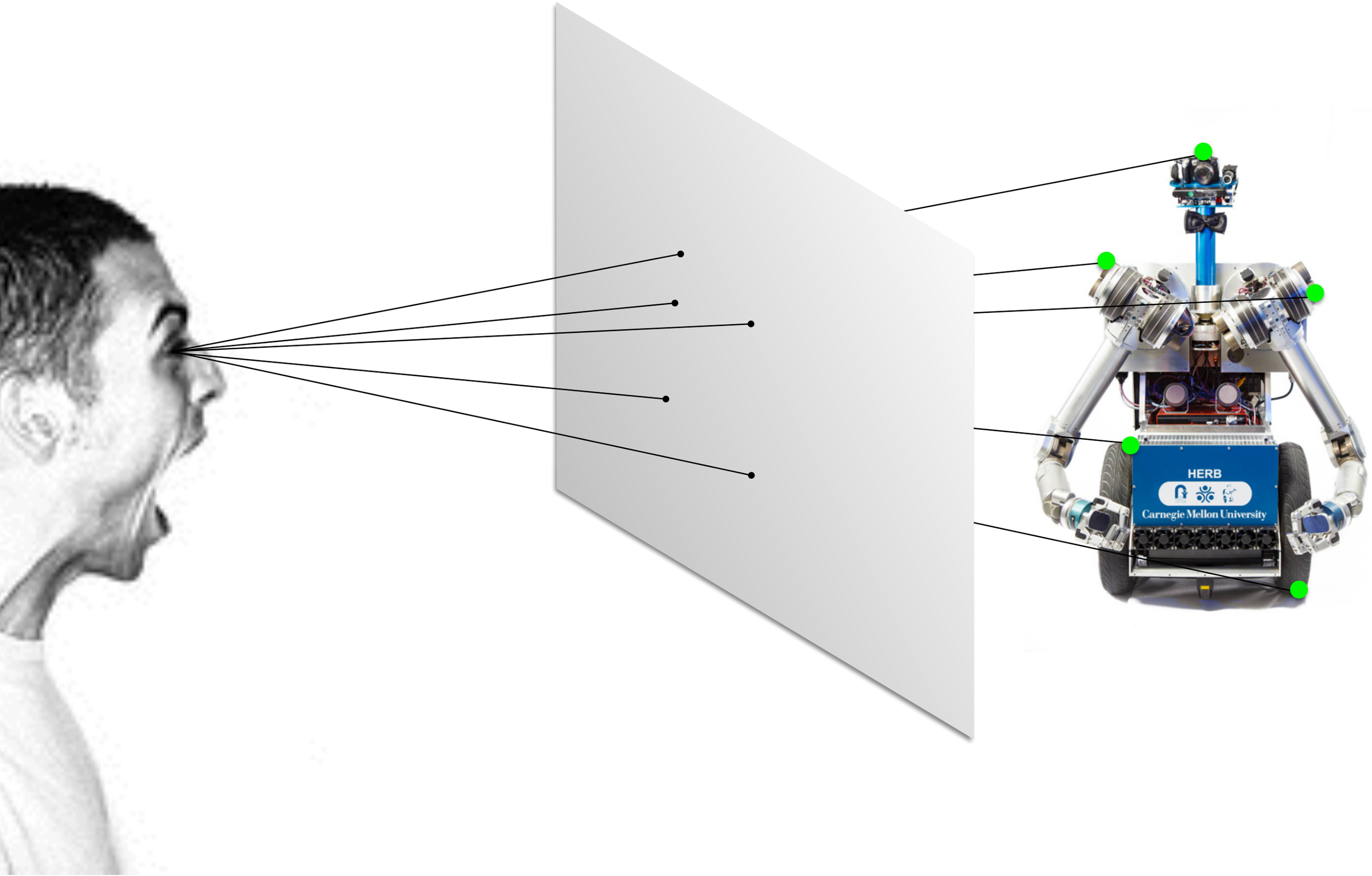
What does the second observer see?

You see points on HERB

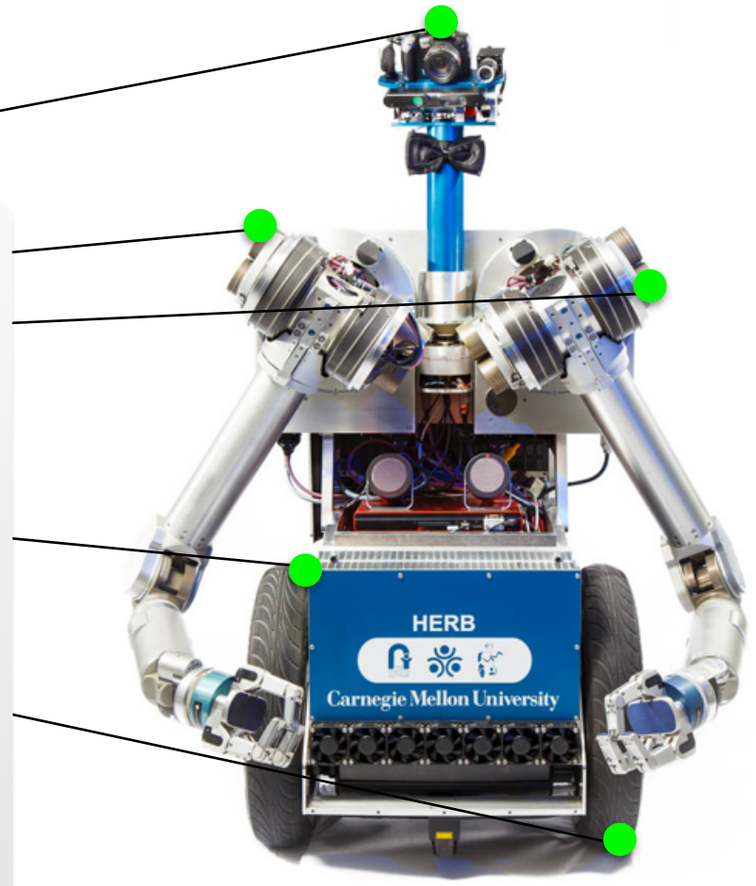


Second person sees lines

This is Epipolar Geometry

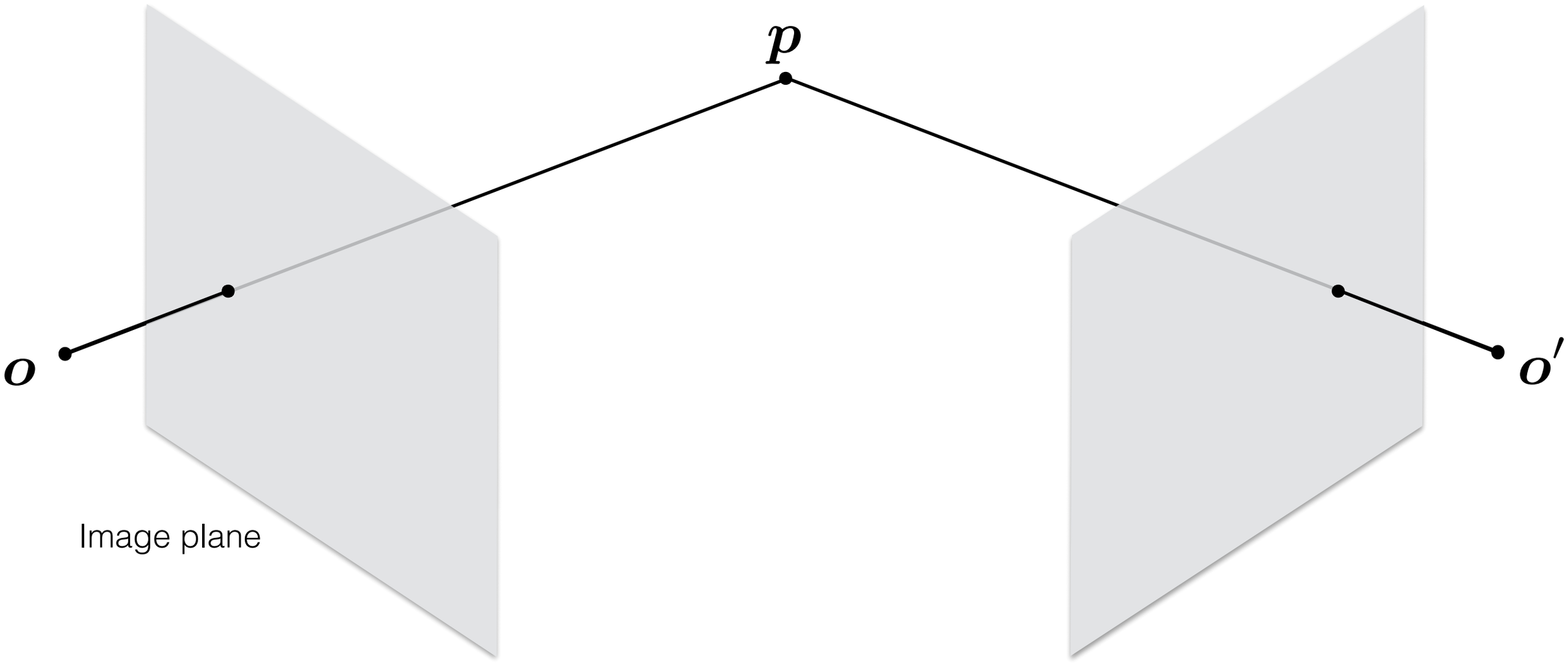


Epipole

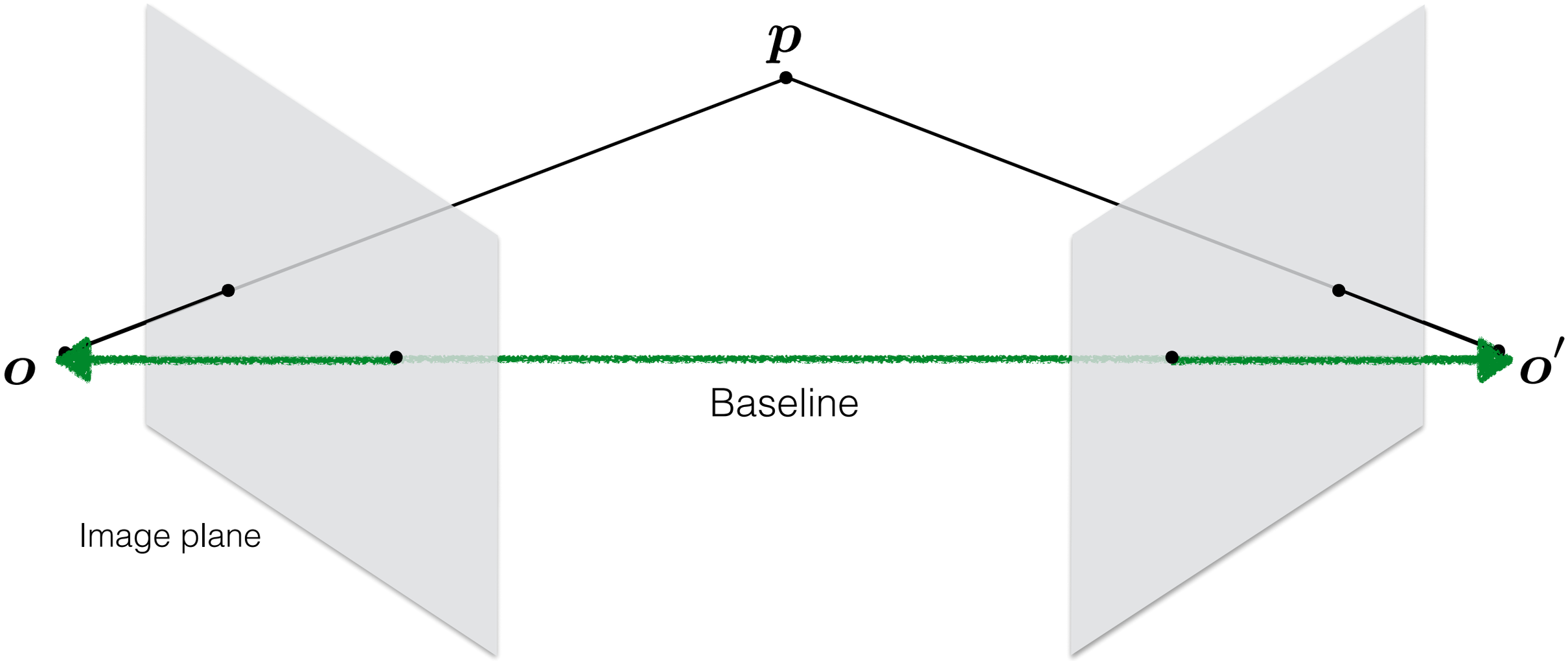


Epipolar lines

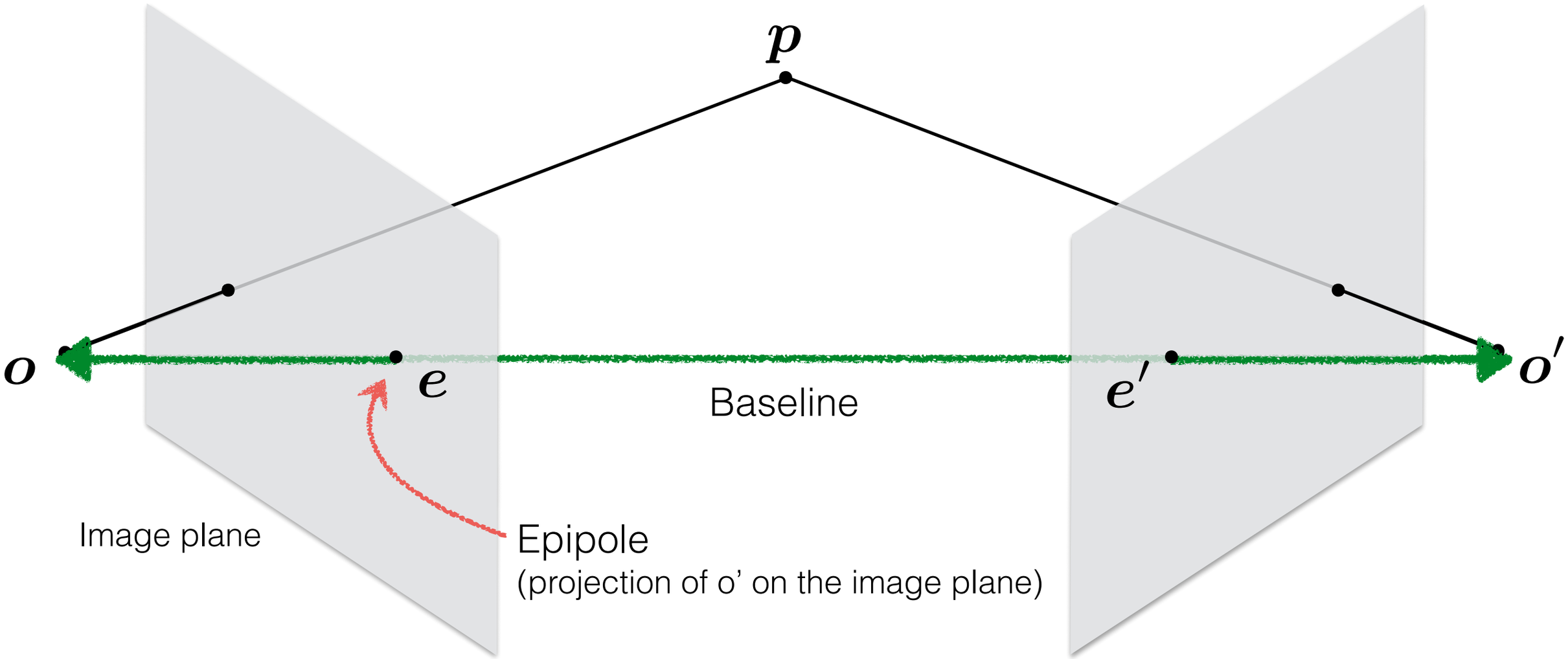
Epipolar geometry



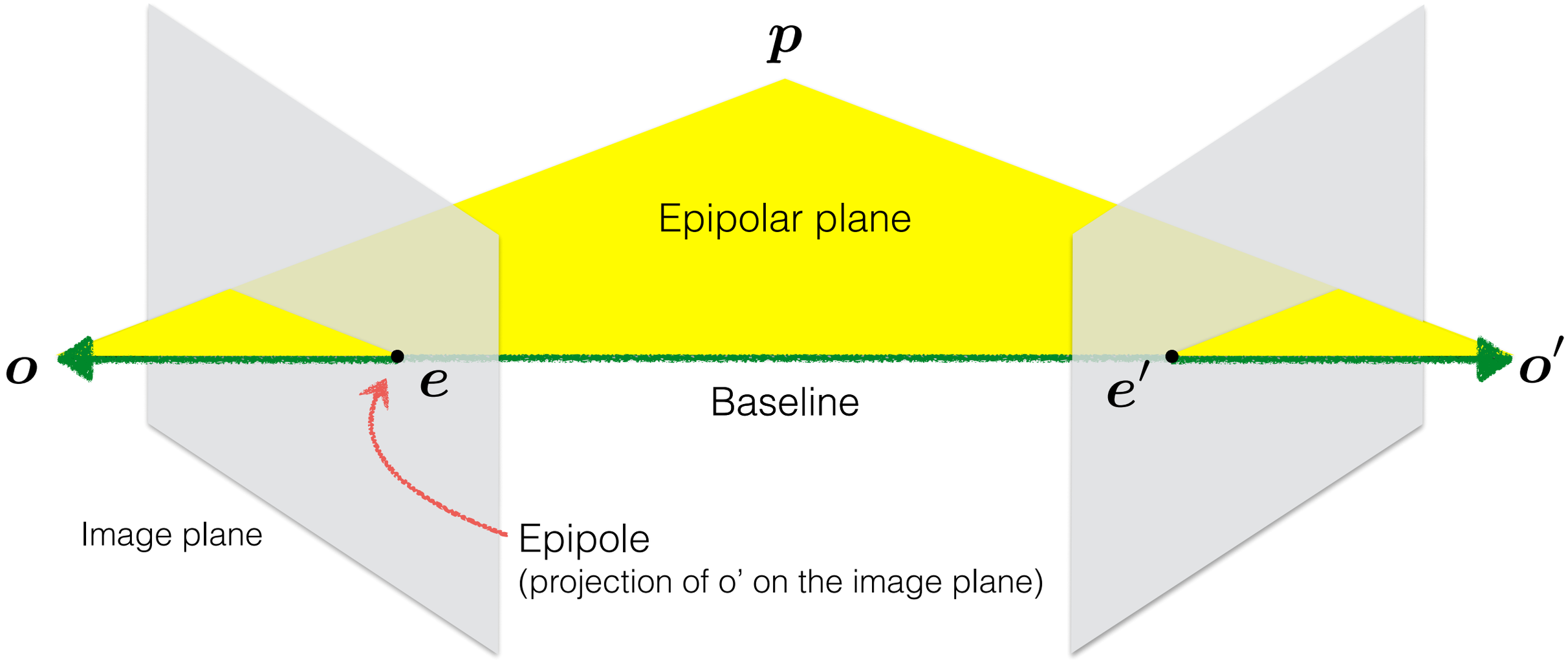
Epipolar geometry



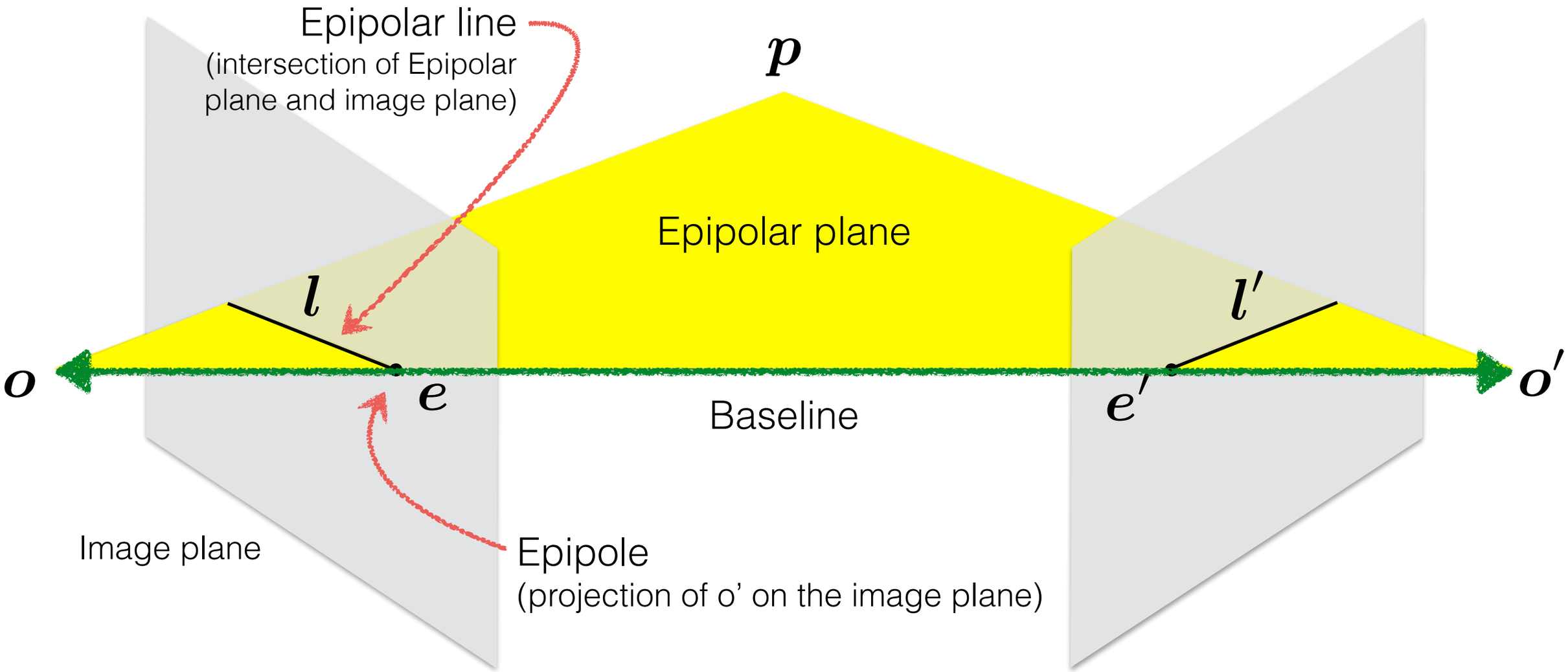
Epipolar geometry



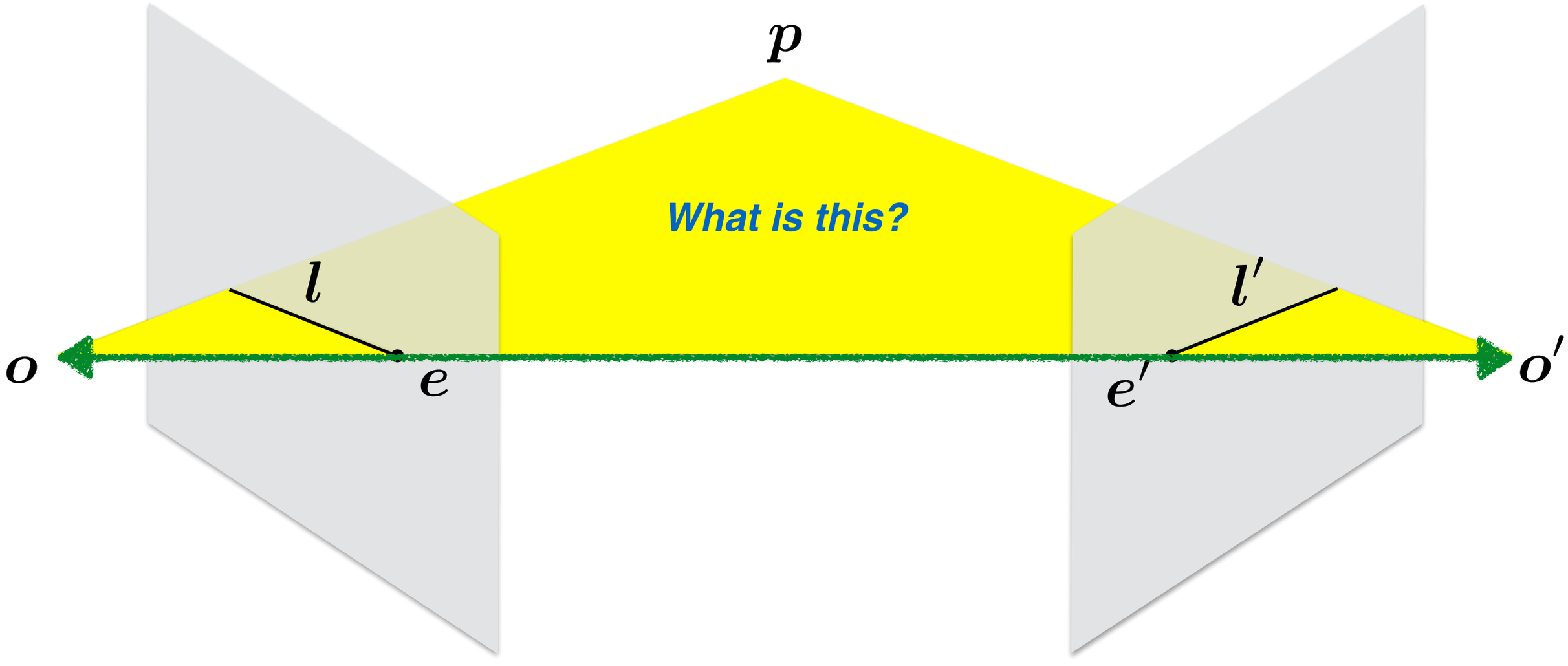
Epipolar geometry



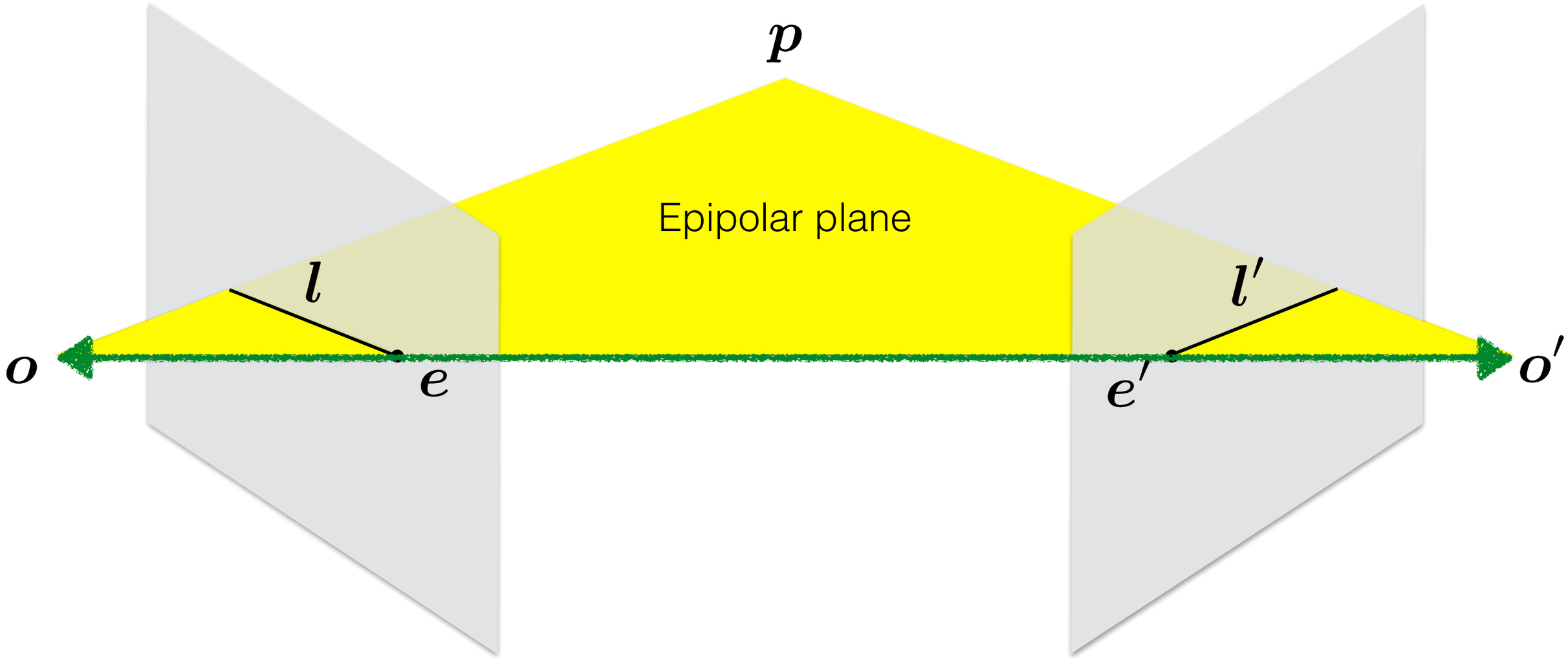
Epipolar geometry



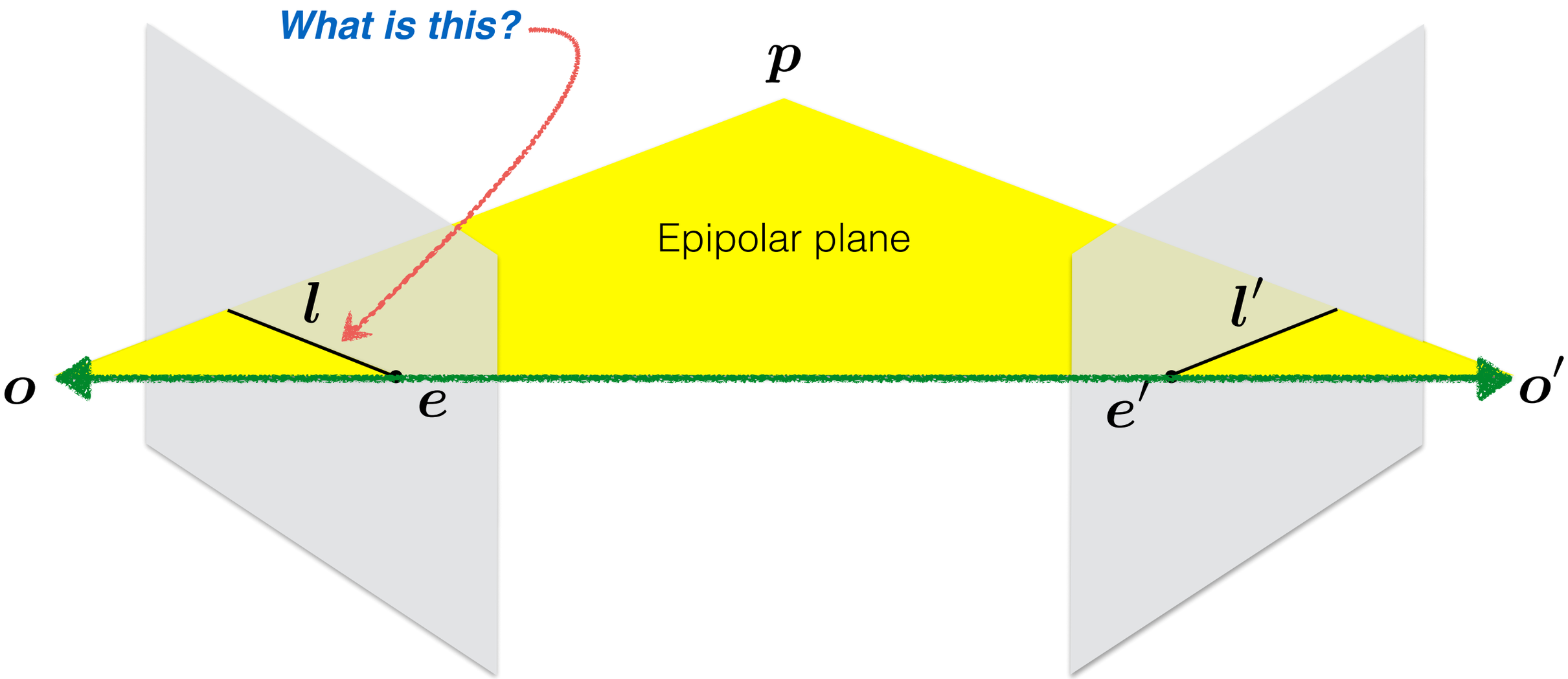
Quiz



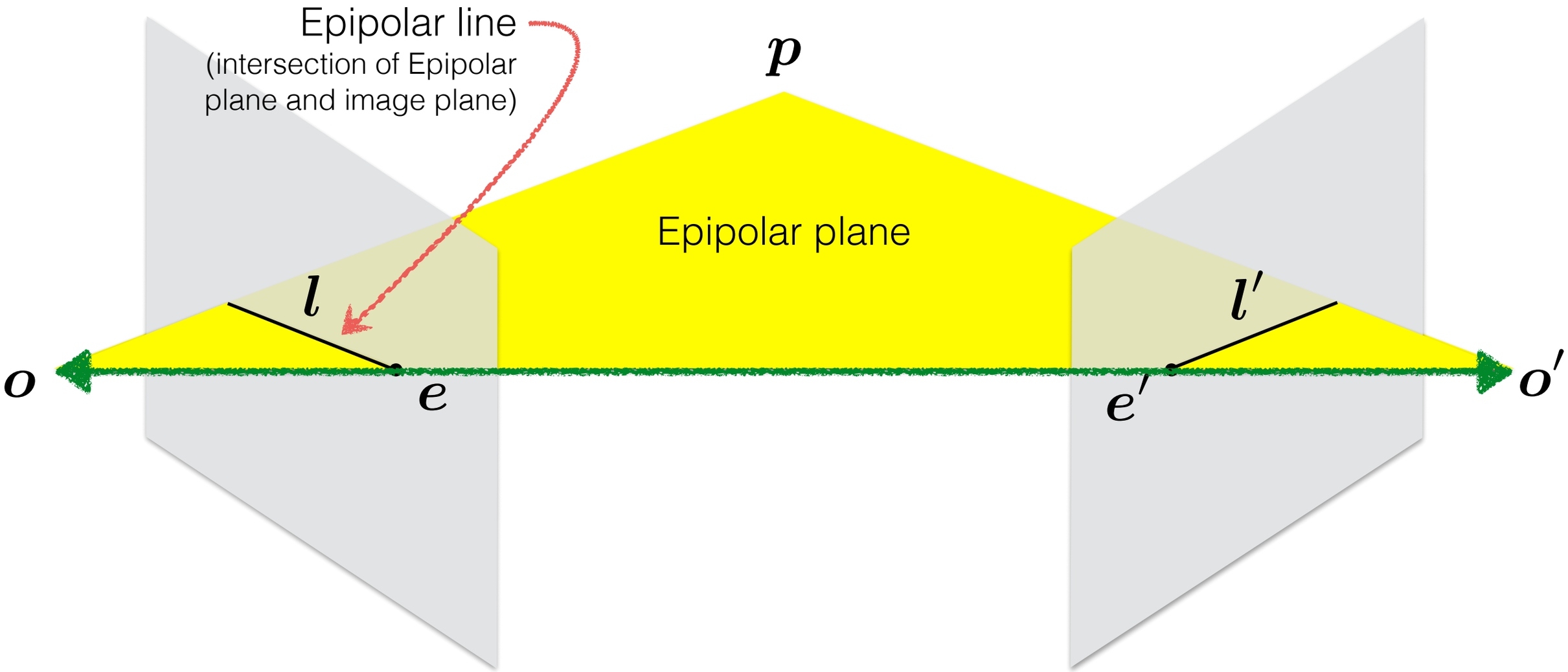
Quiz



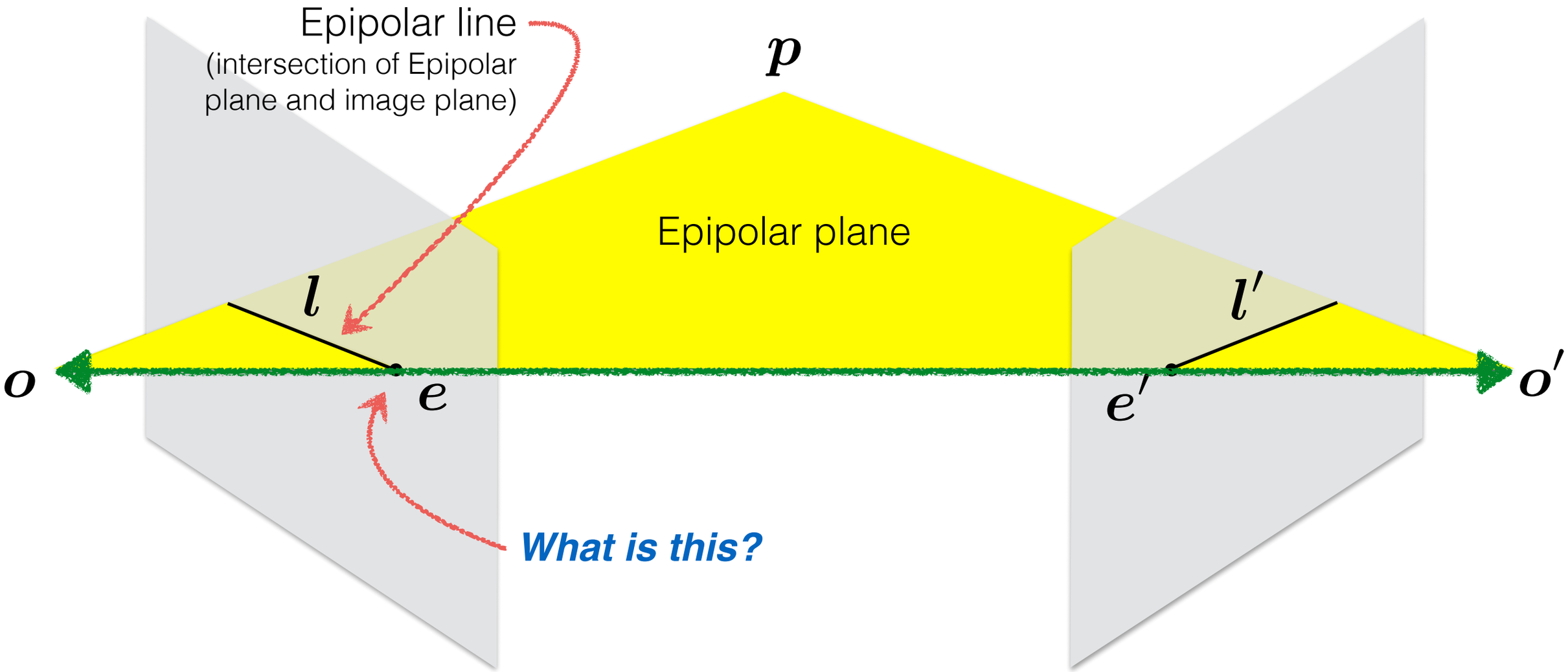
Quiz



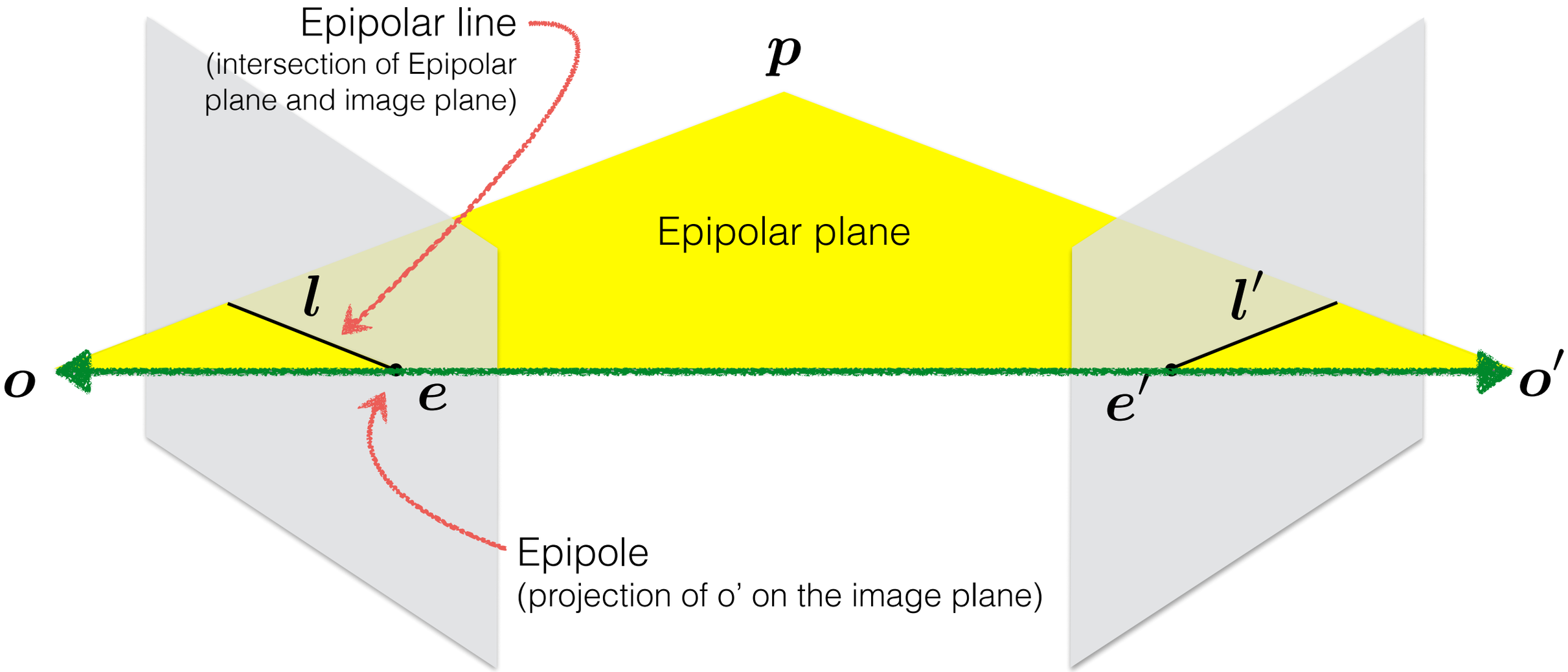
Quiz


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

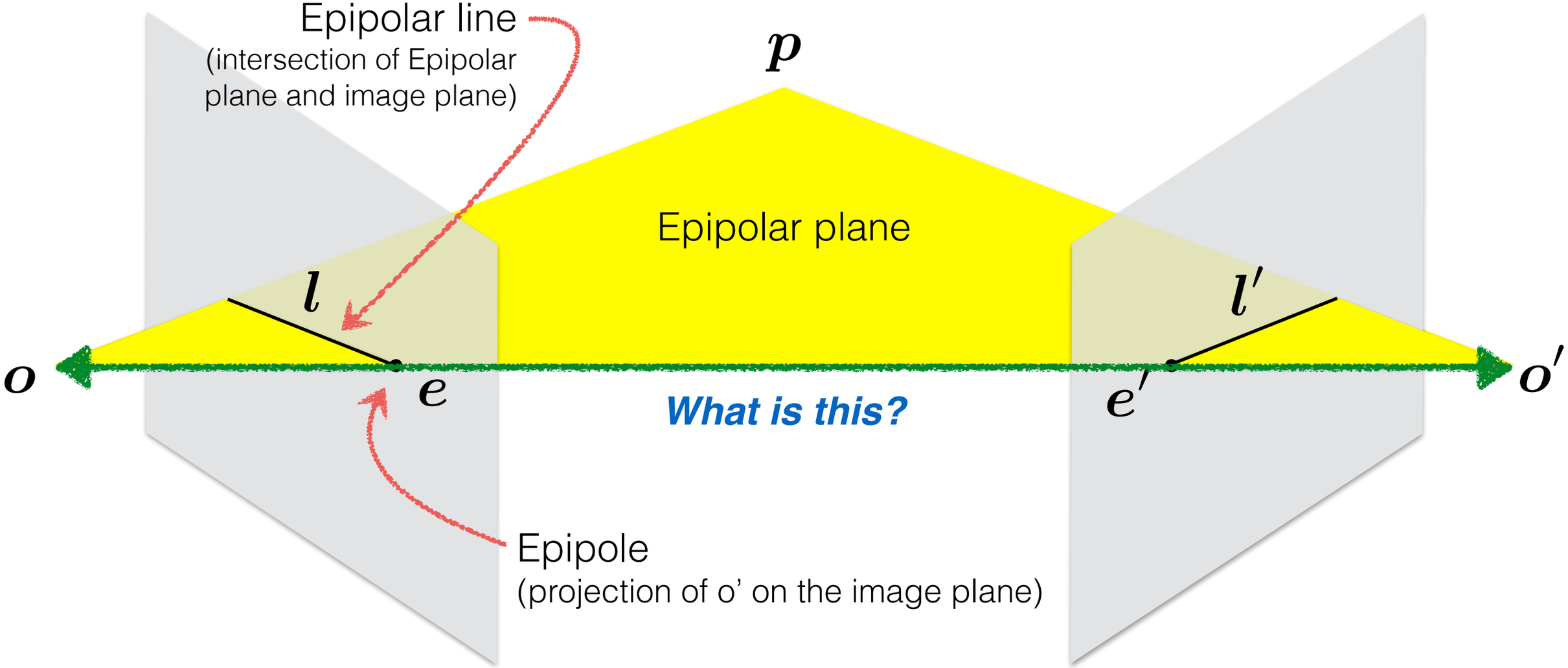
Quiz


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

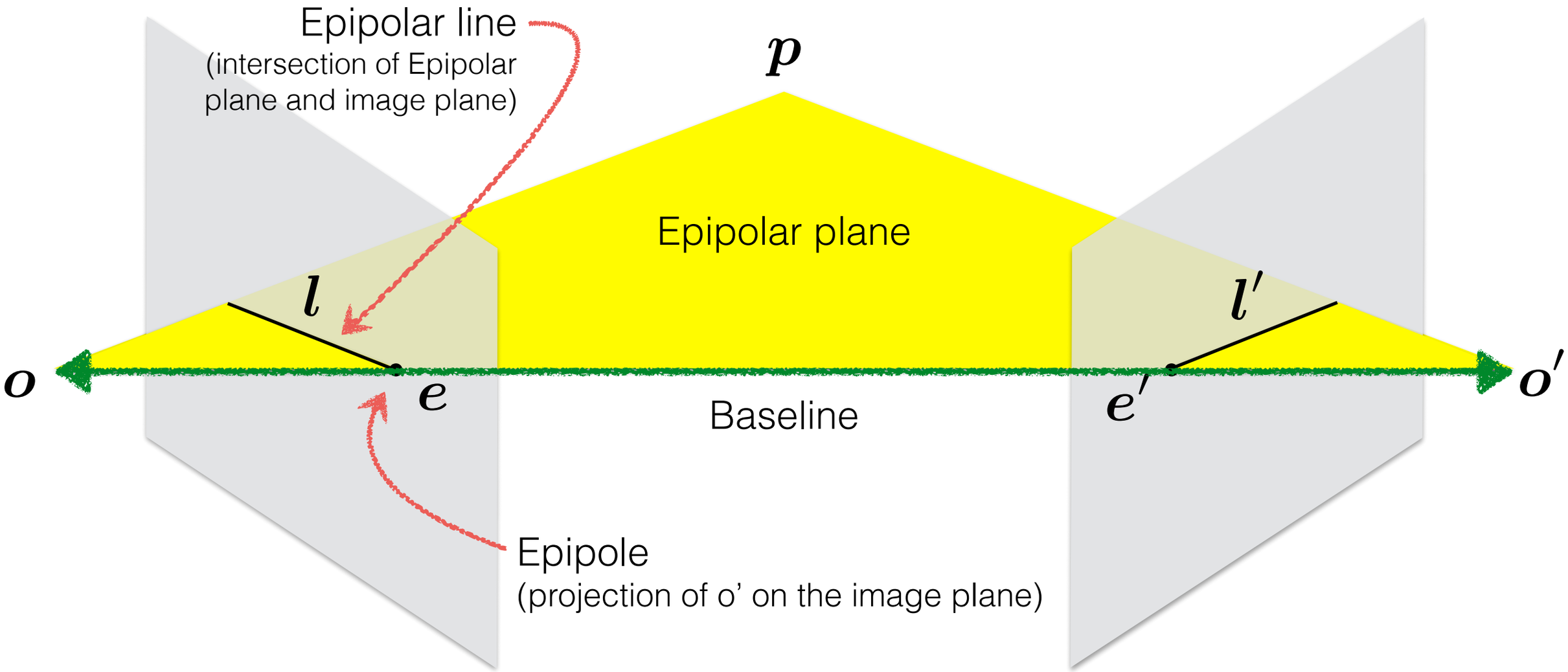
Quiz


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)

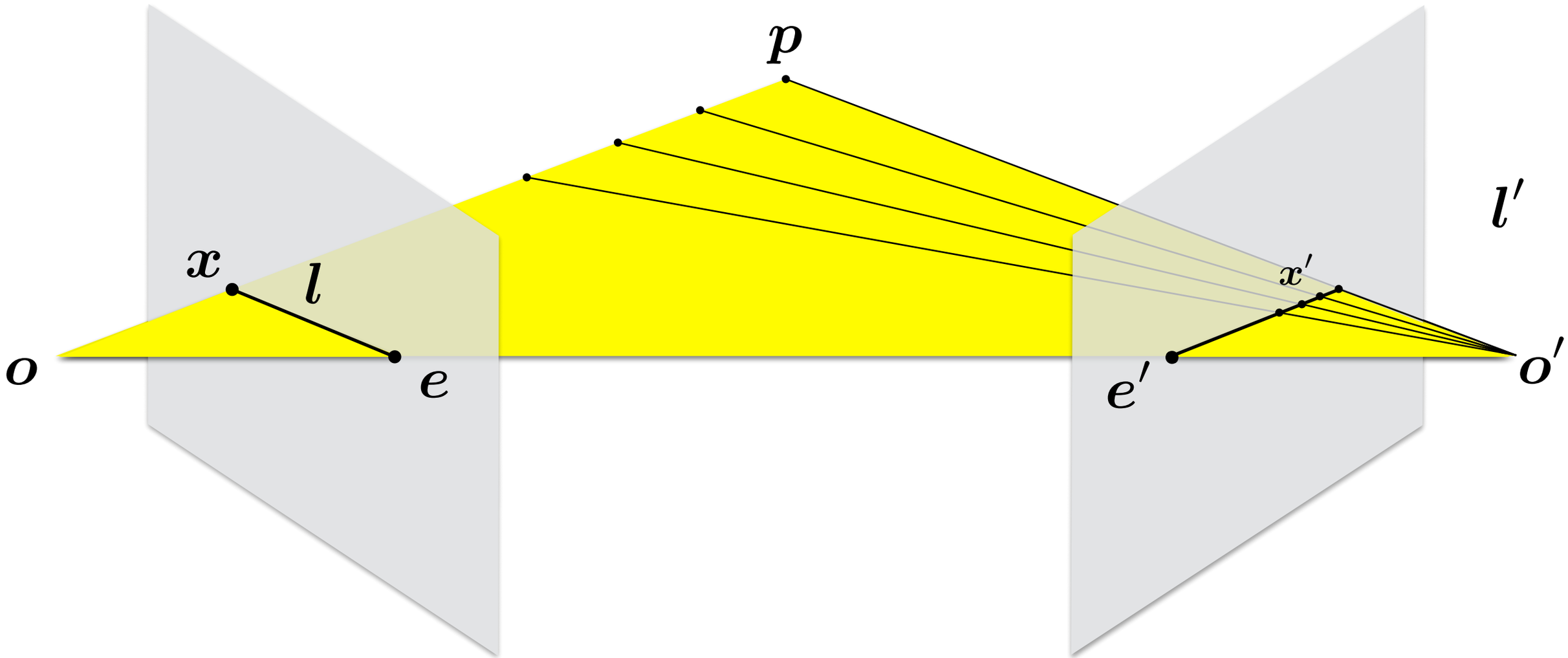
Quiz


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

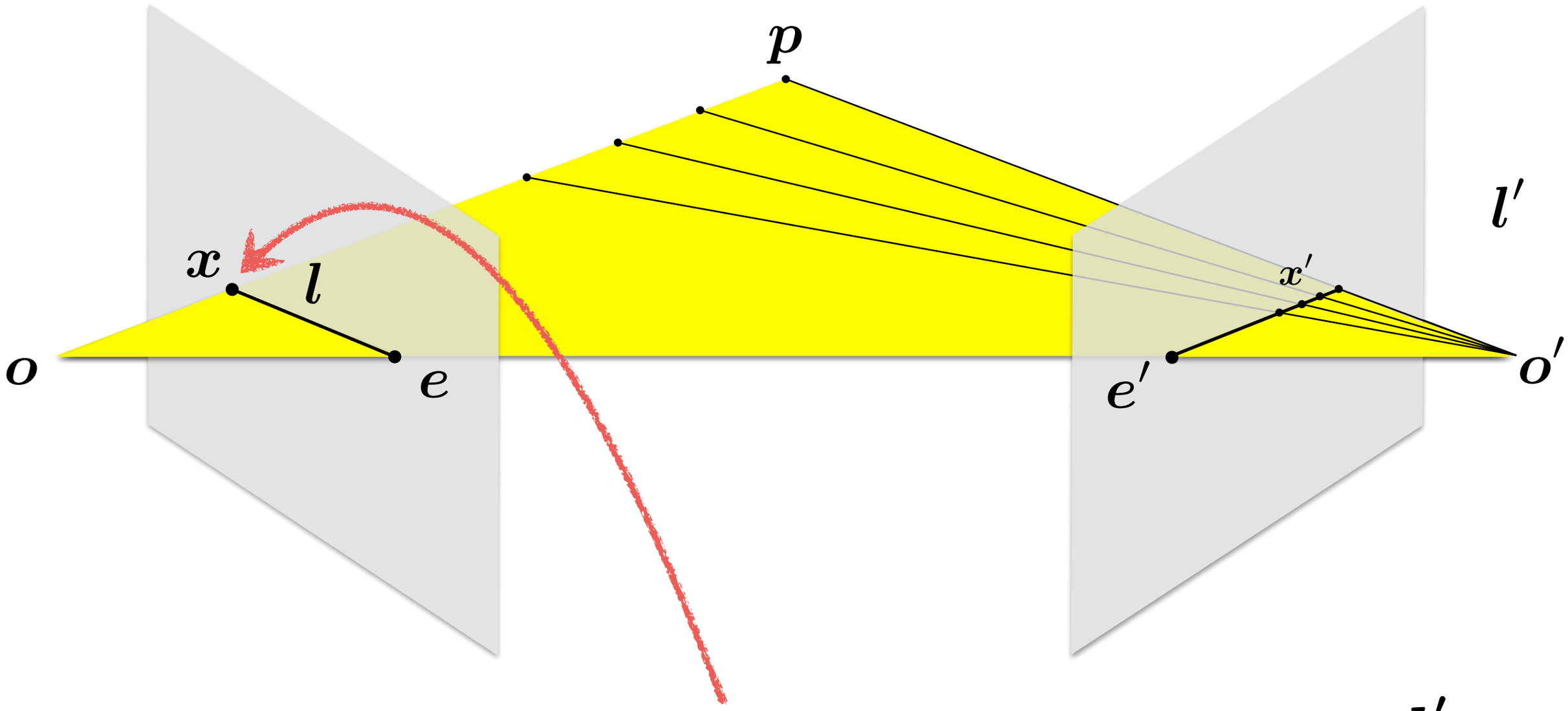

Epipolar line
(intersection of Epipolar plane and image plane)
p
Epipolar plane
l
l'
o
e
e'
o'
Baseline
Epipole
(projection of o' on the image plane)

Epipolar constraint



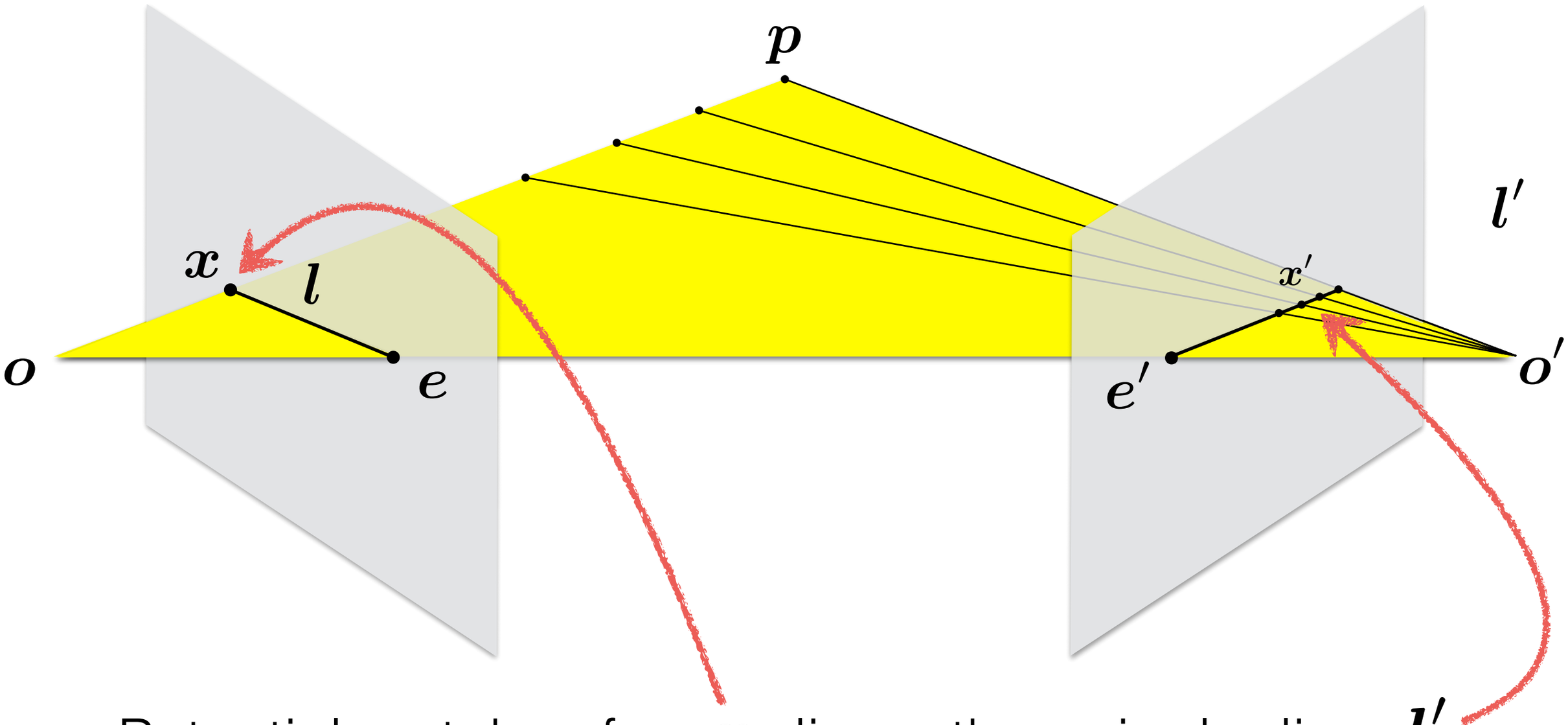
Potential matches for x lie on the epipolar line l'

Epipolar constraint

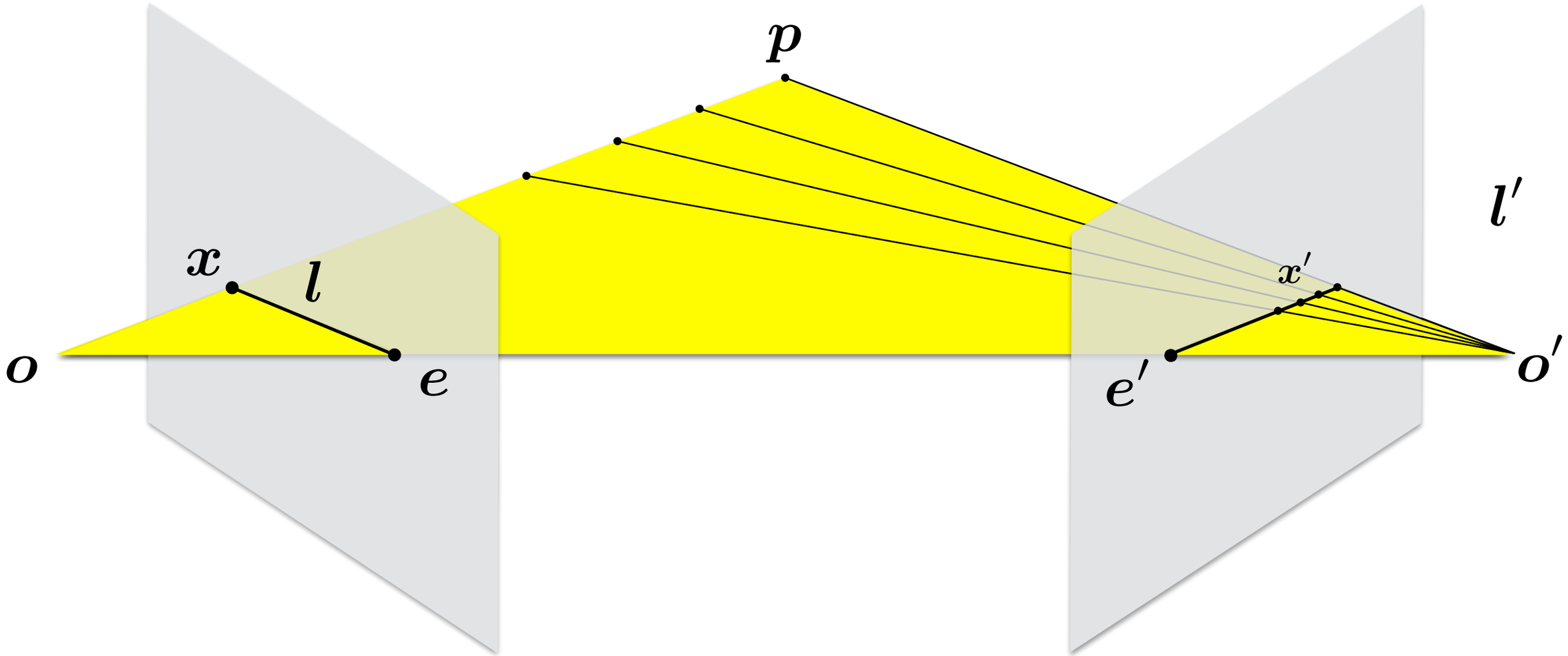


Potential matches for x lie on the epipolar line l'

Epipolar constraint



Potential matches for x lie on the epipolar line l'



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

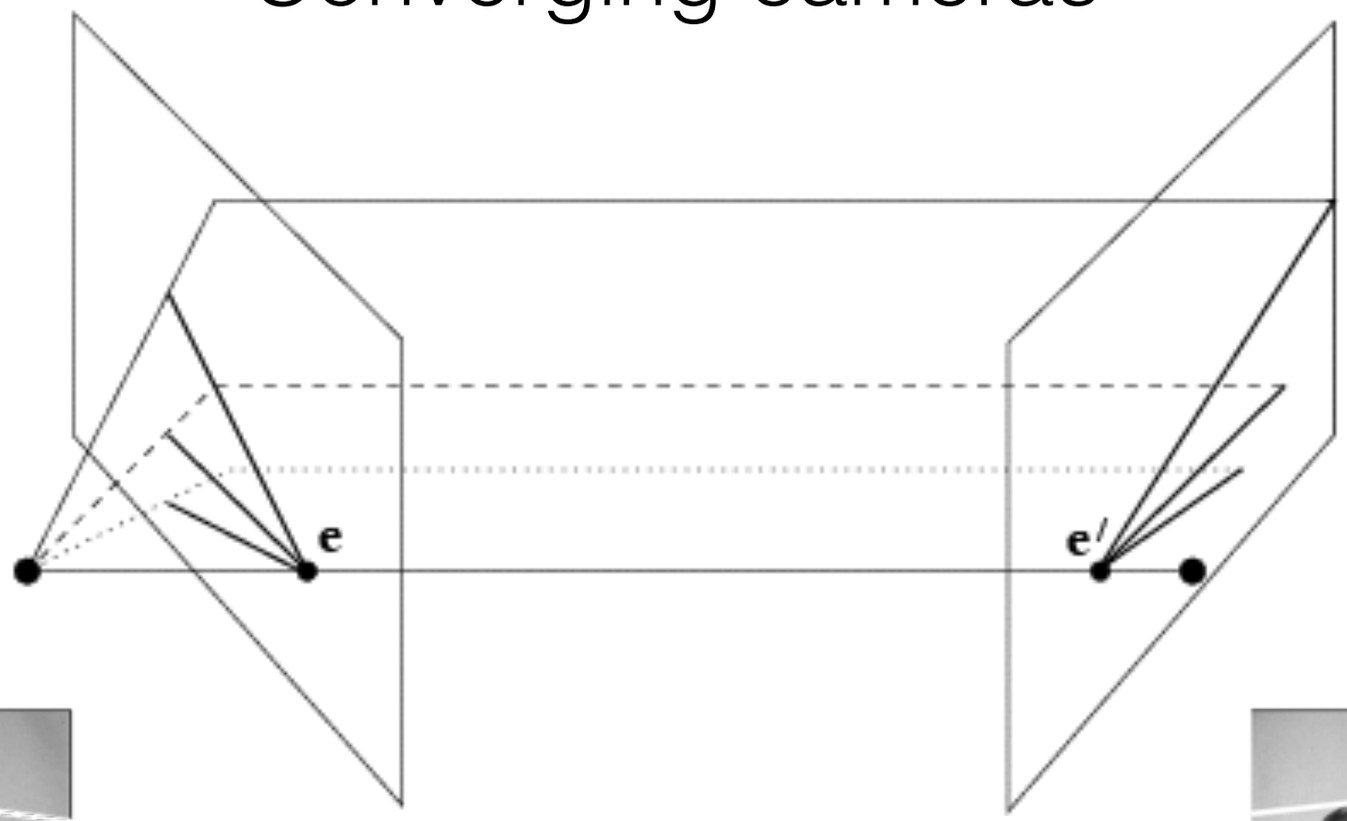
The baseline connects the _____ and _____

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

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

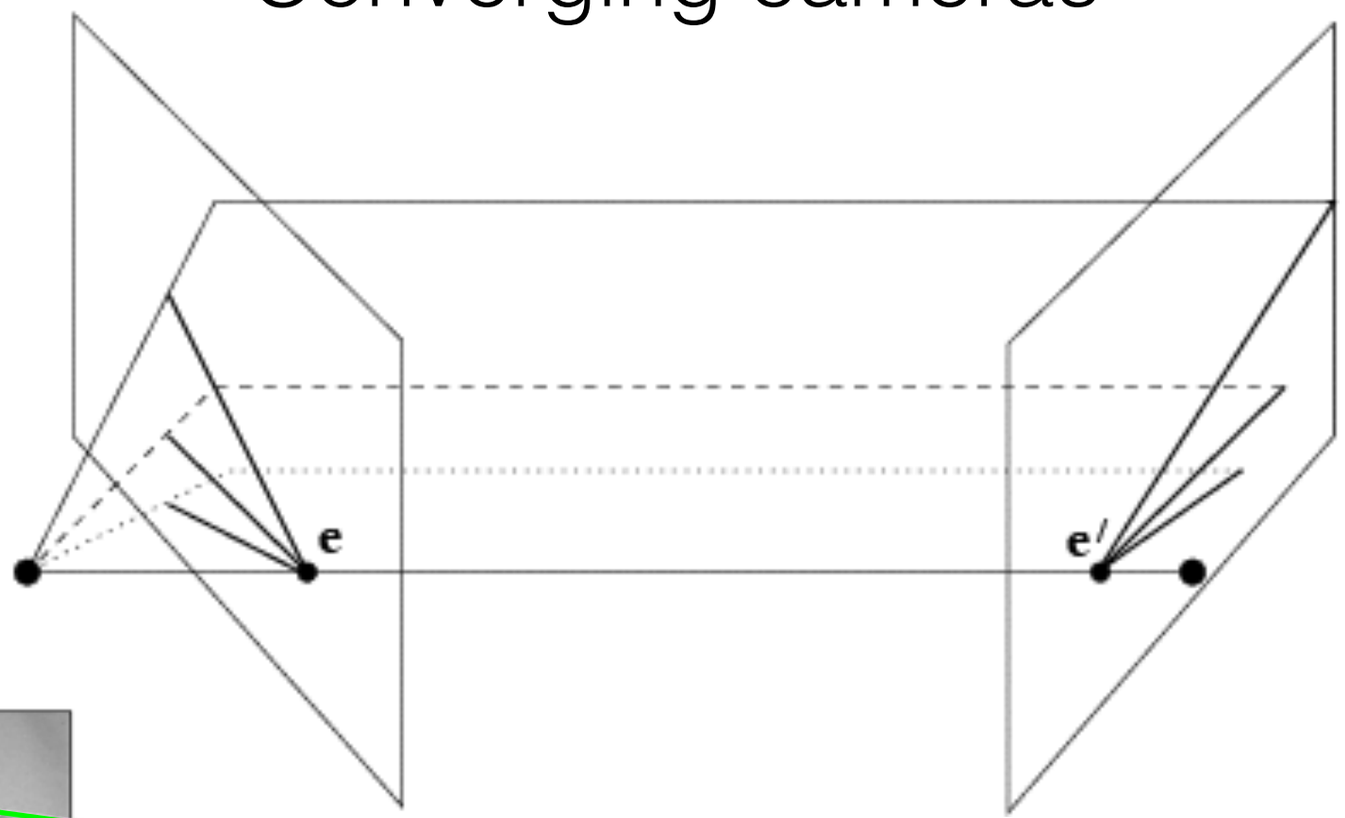
All epipolar lines in an image intersect at the _____

Converging cameras



Where is the epipole in this image?

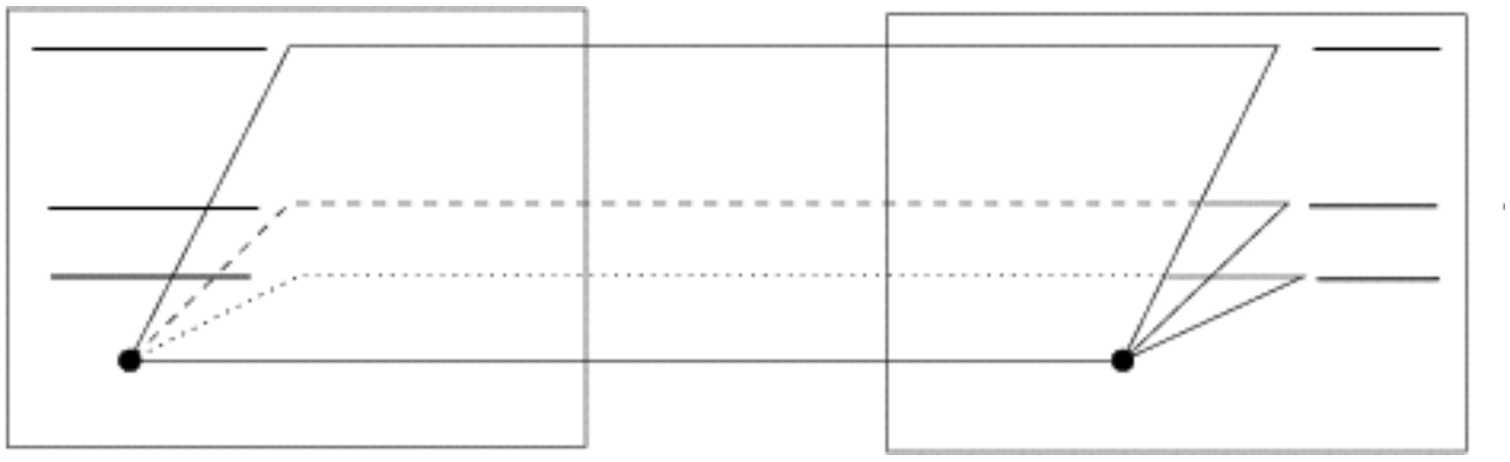
Converging cameras



Where is the epipole in this image?

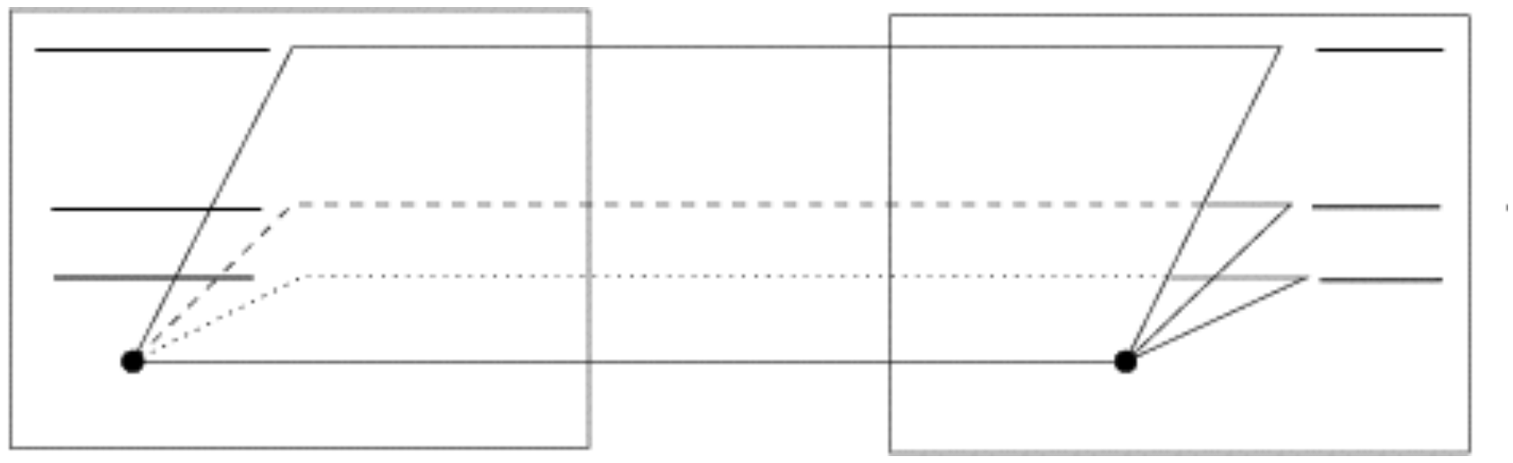
It's not always in the image

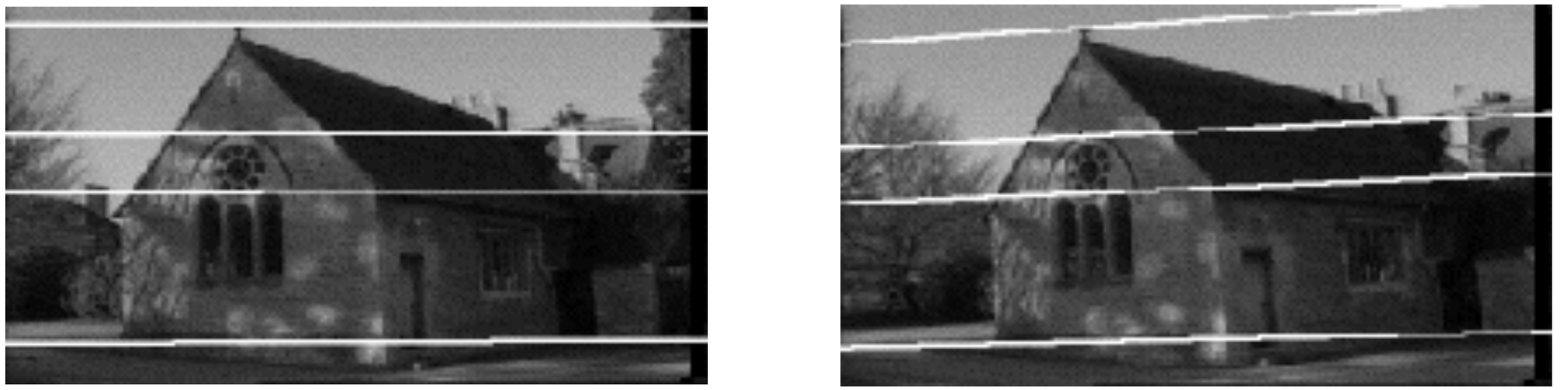
Parallel cameras



Where is the epipole?

Parallel cameras





epipole at infinity

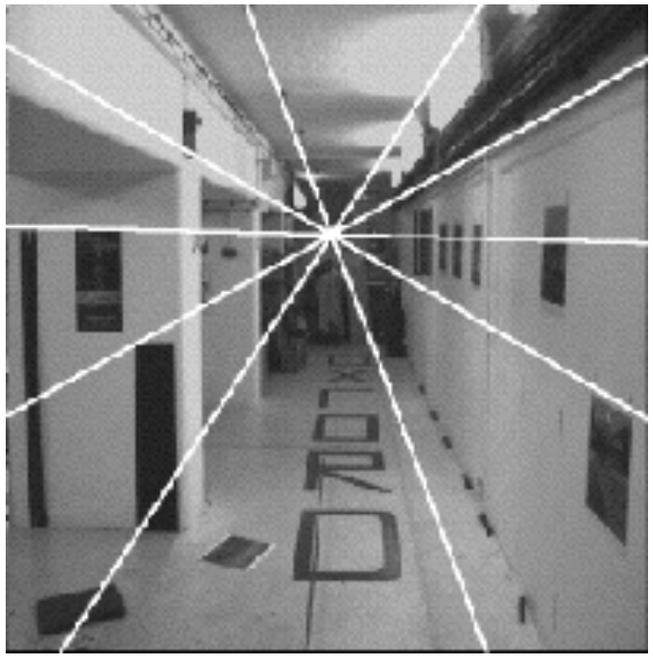
Forward moving camera

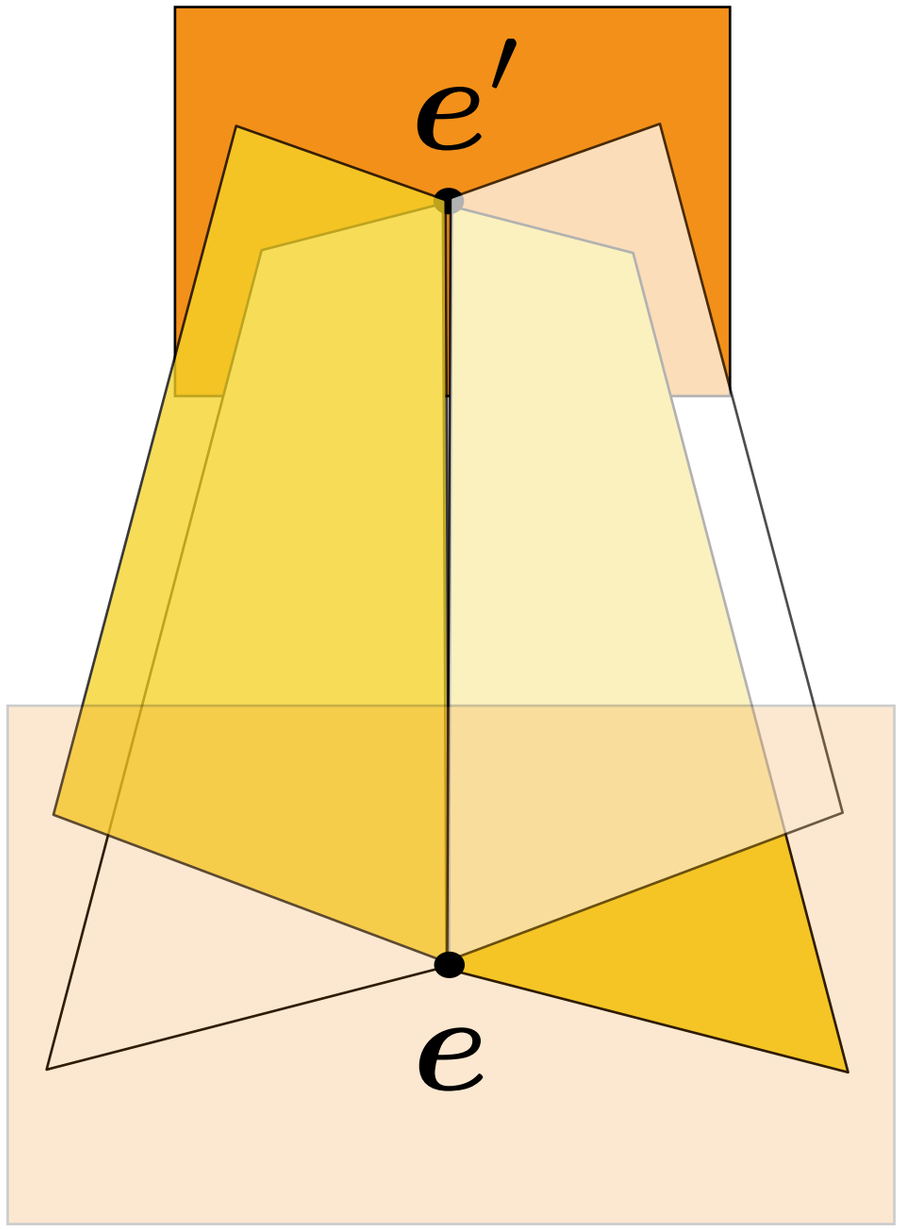
Forward moving camera

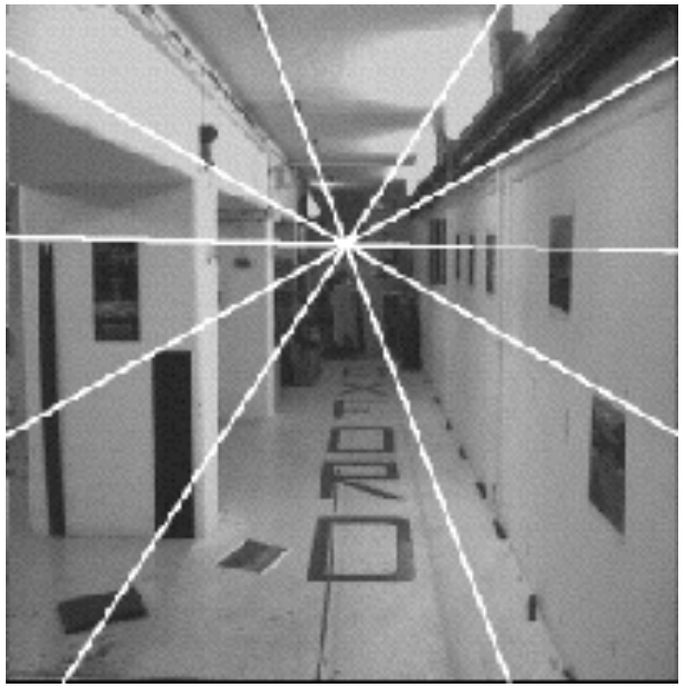
Where is the epipole?

What do the epipolar lines look like?

Epipole has same coordinates in both images.
Points move along lines radiating from “Focus of expansion”

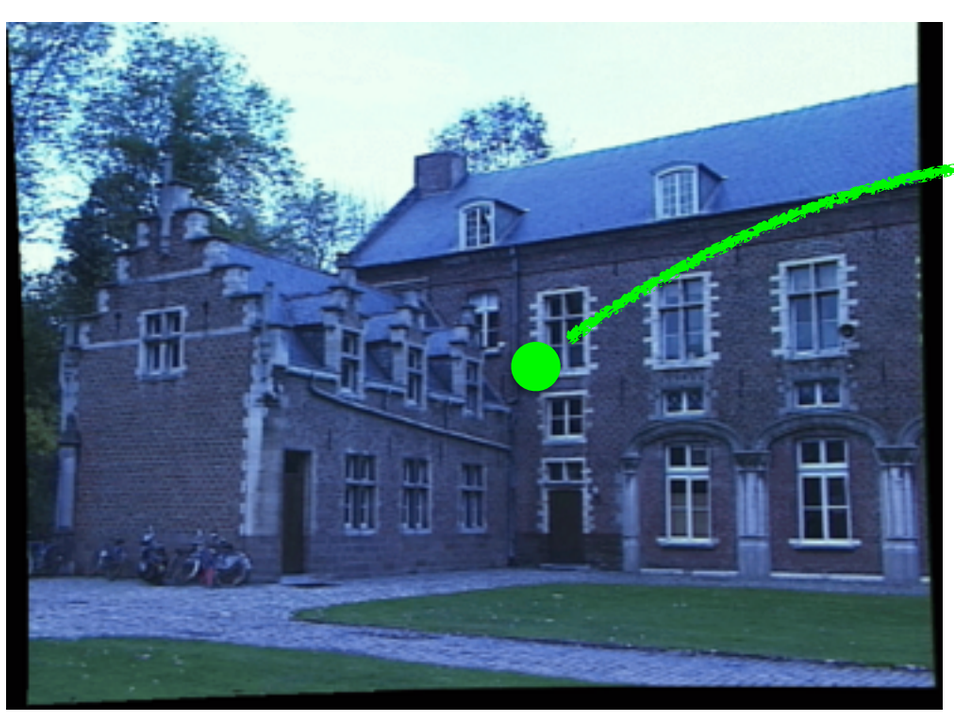




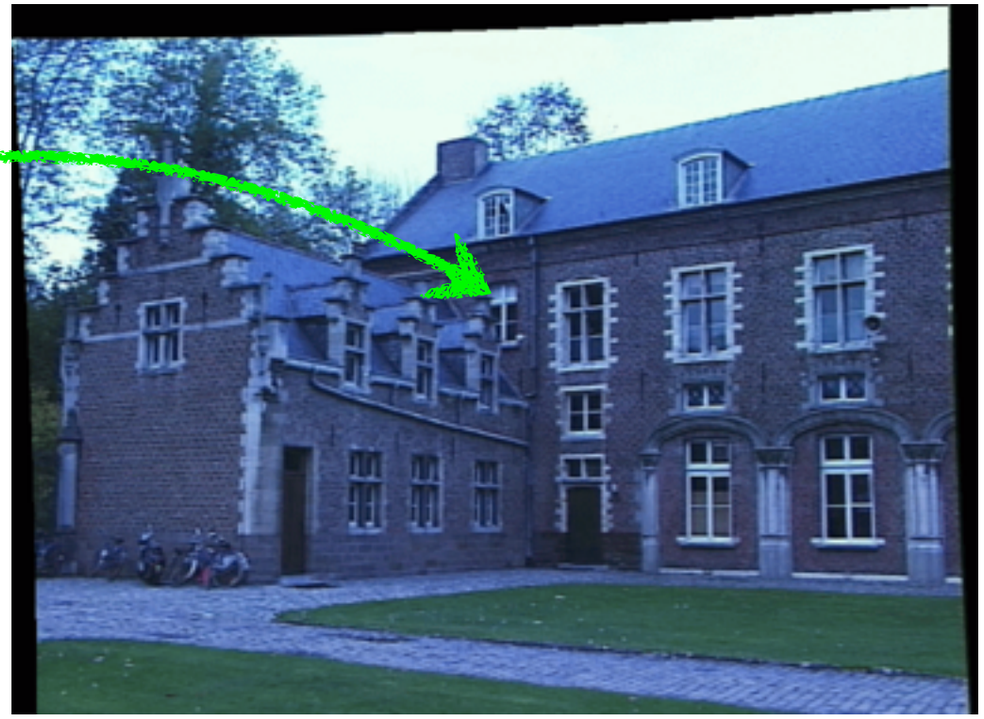


The epipolar constraint is an important concept for stereo vision

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



Left image

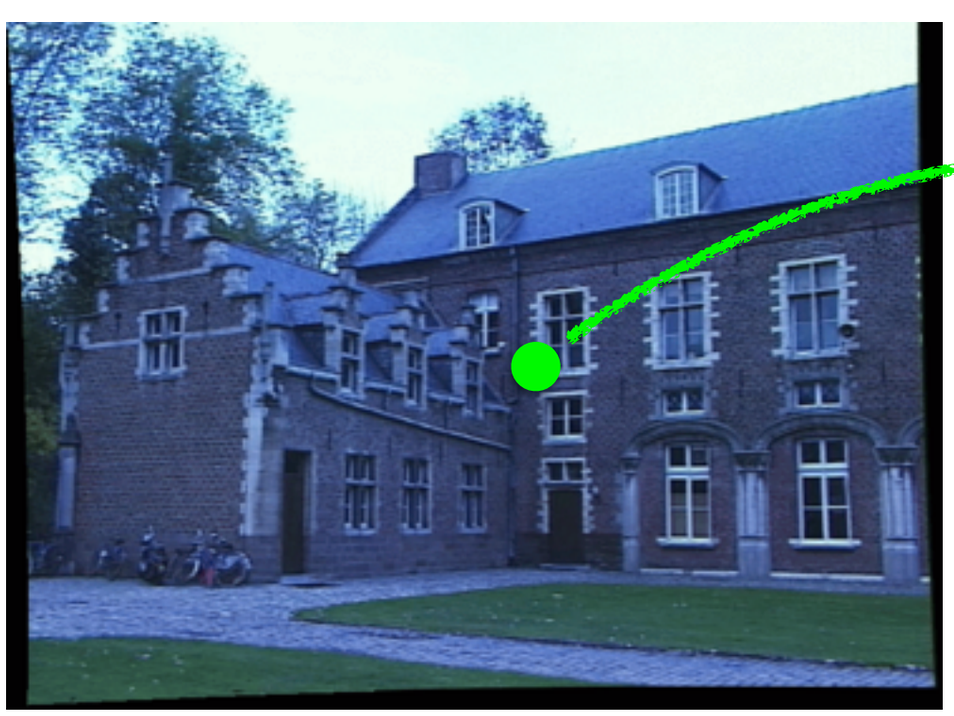


Right image

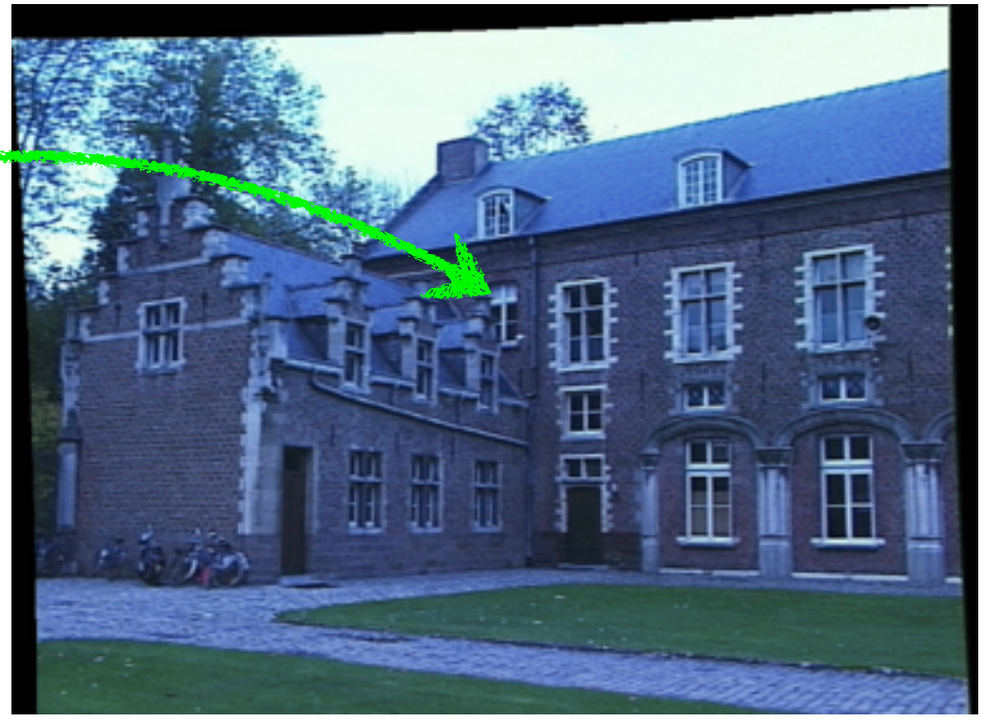
How would you do it?

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

(if the images have been rectified)
Epipolar constrain reduces search to a single line

iv-tec

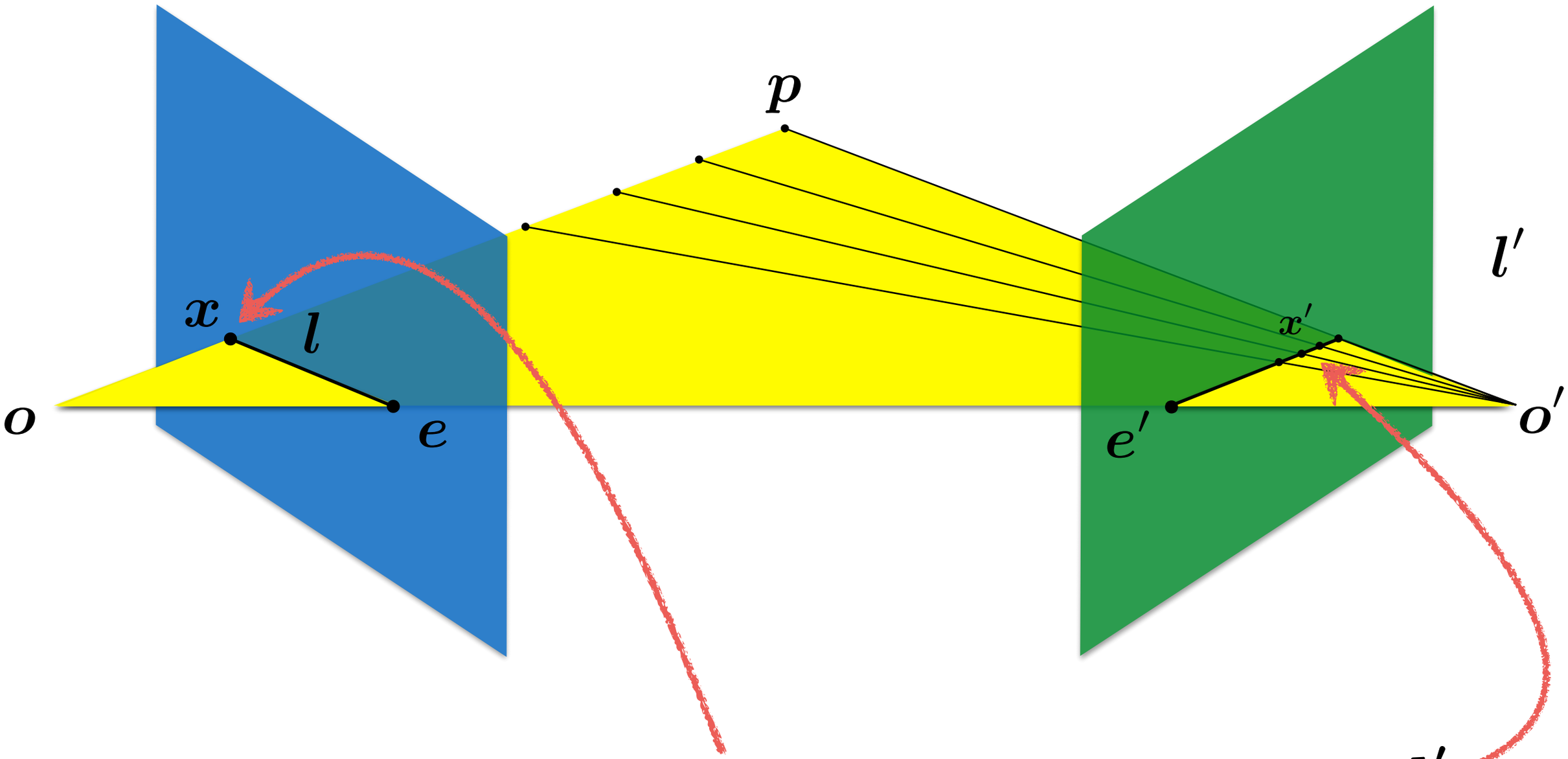
imagination and vision

Essential Matrix

16-385 Computer Vision (Kris Kitani)

Carnegie Mellon University

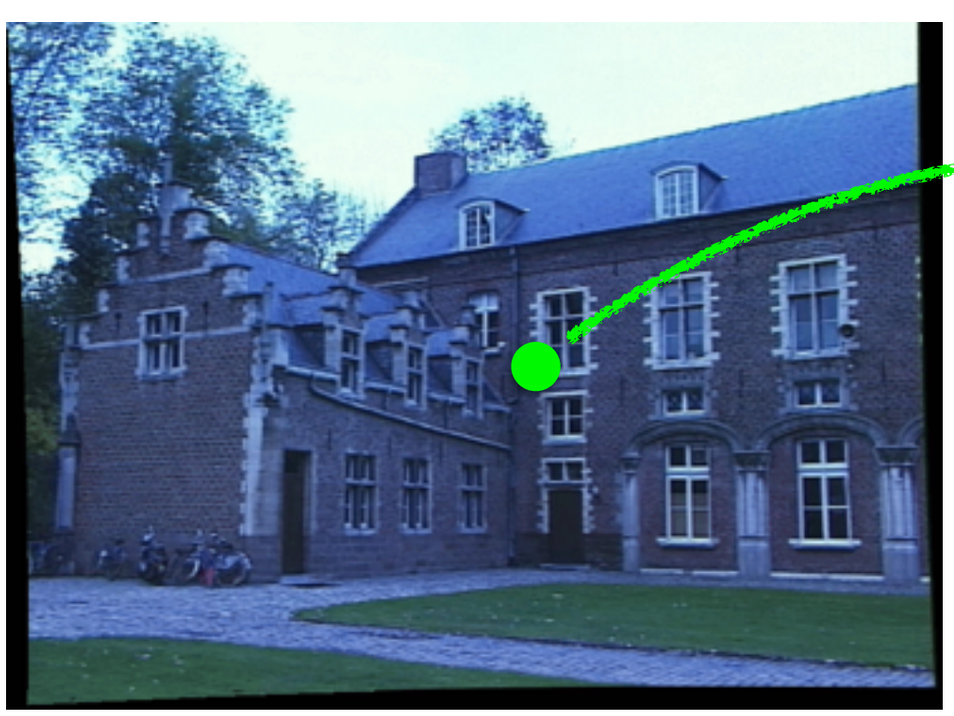
Recall:Epipolar constraint



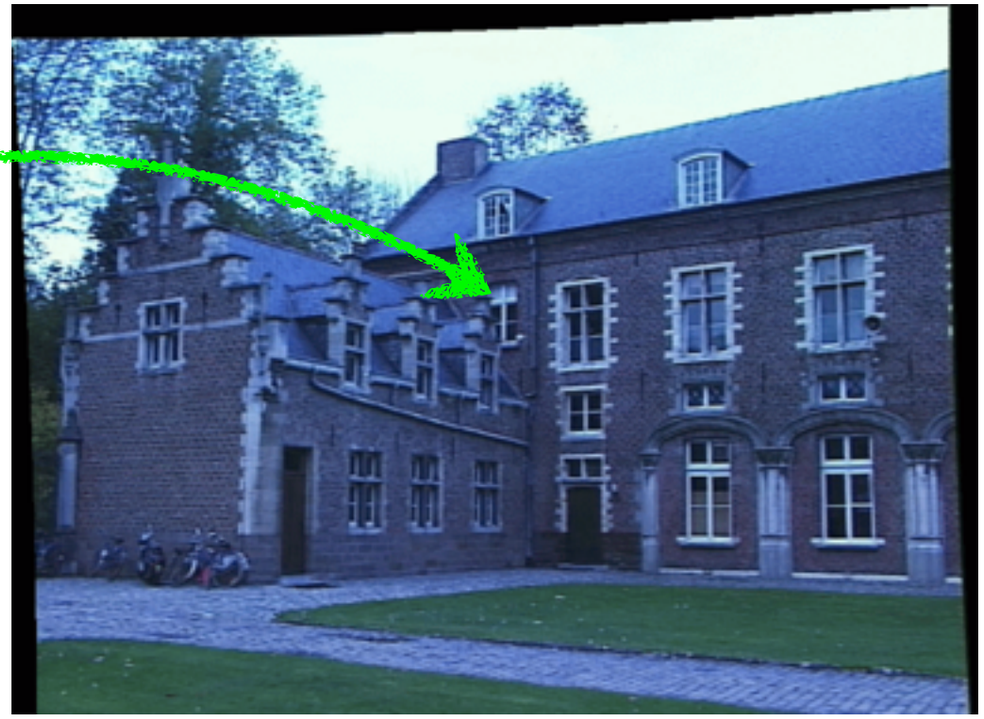
Potential matches for x lie on the epipolar line l'

The epipolar geometry is an important concept for stereo vision

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



Left image

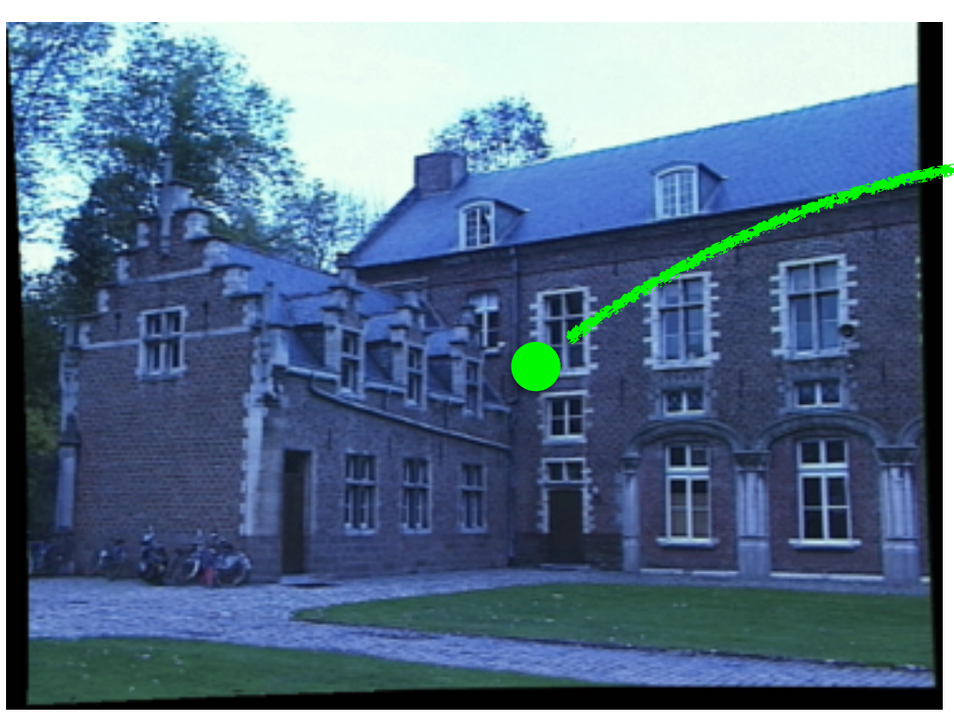


Right image

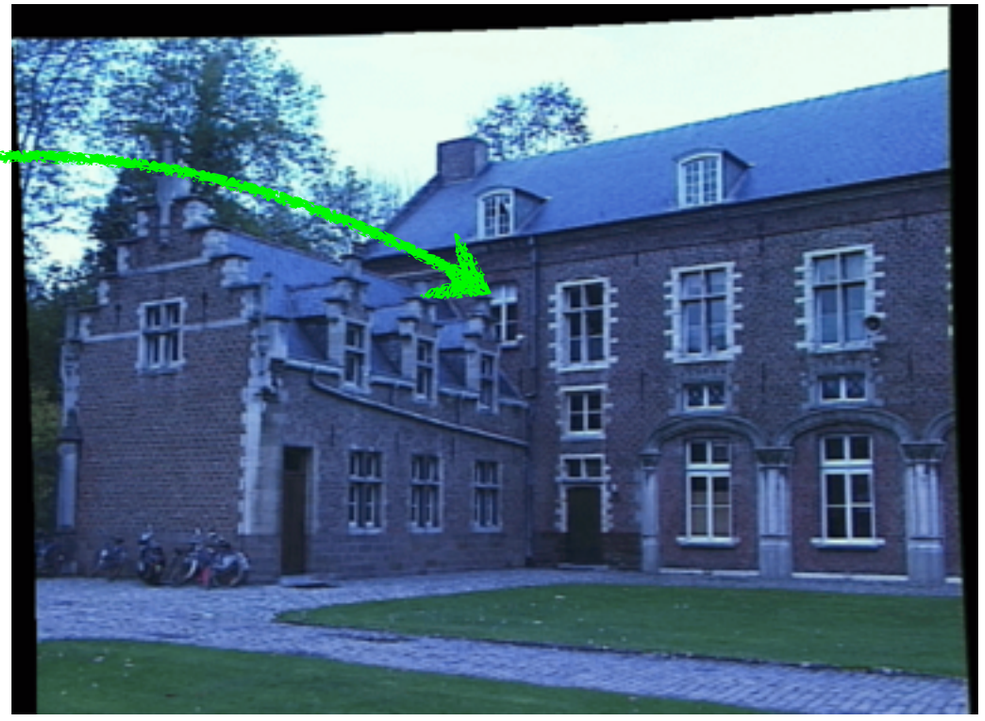
How would you do it?

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

Epipolar constrain reduces search to a single line

How do you compute the epipolar line?

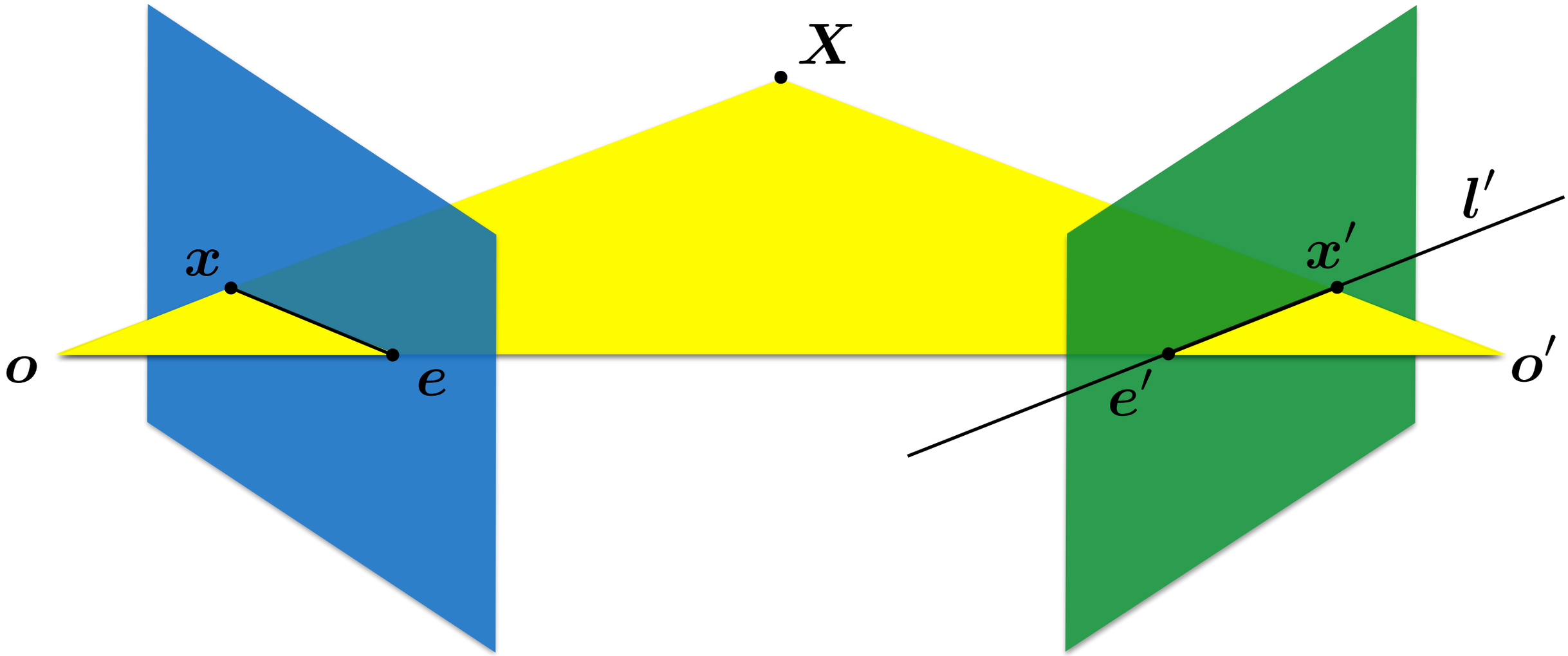
Essential Matrix

$$E$$

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 view.

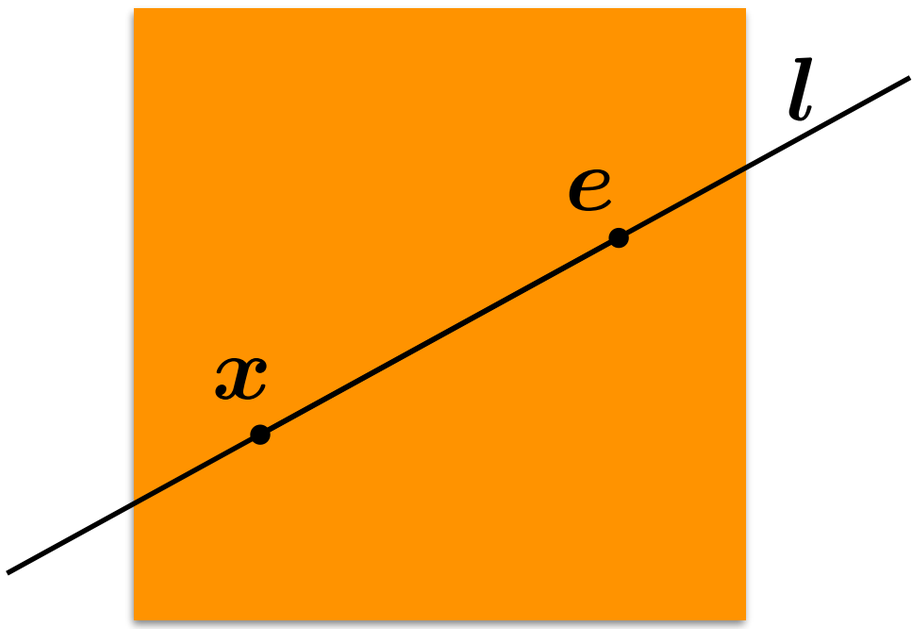
$$\mathbf{E}x = l'$$



Representing the …

Epipolar Line

$$ax + by + c = 0 \qquad \text{in vector form} \qquad \boldsymbol{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

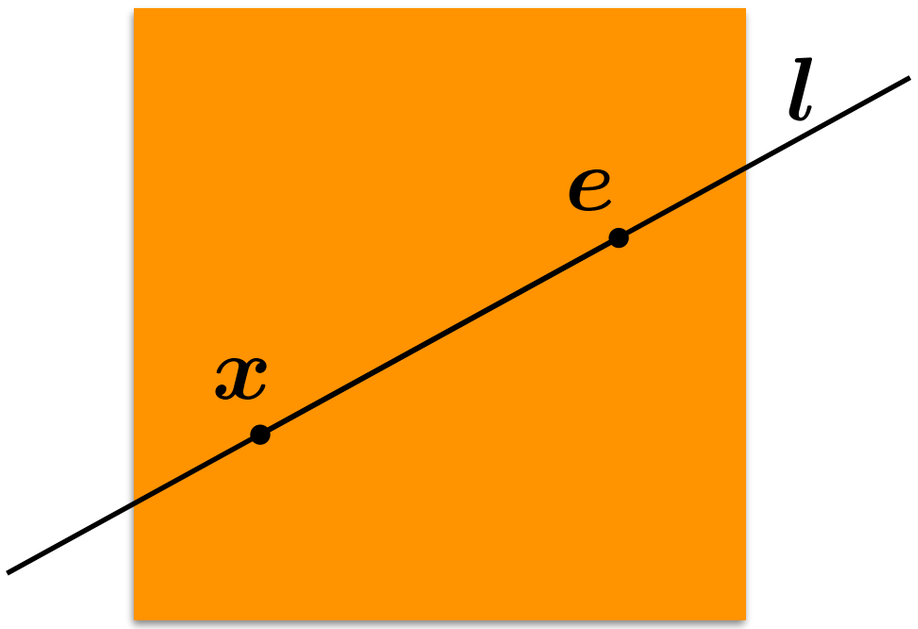


If the point $\boldsymbol{x}$ is on the epipolar line $\boldsymbol{l}$ then

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

Epipolar Line

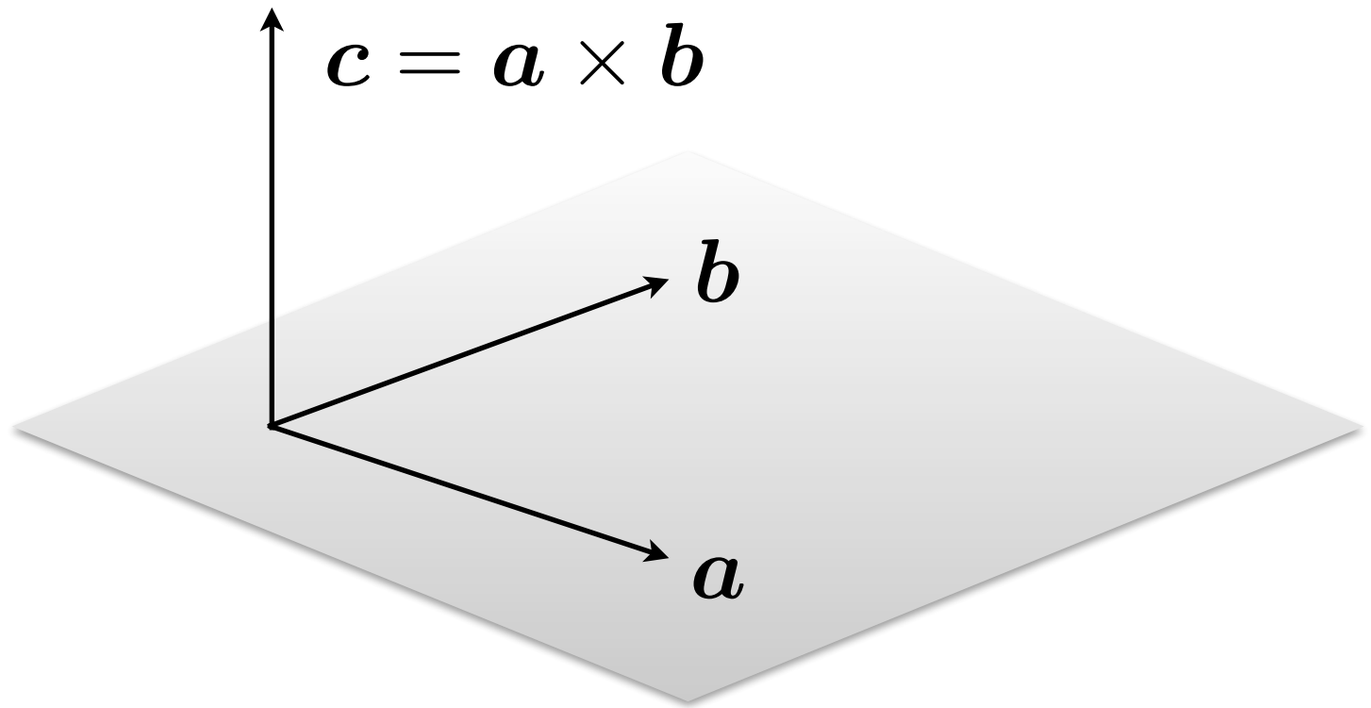
$$ax + by + c = 0 \quad \text{in vector form} \quad \boldsymbol{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$



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

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

Recall: Dot Product

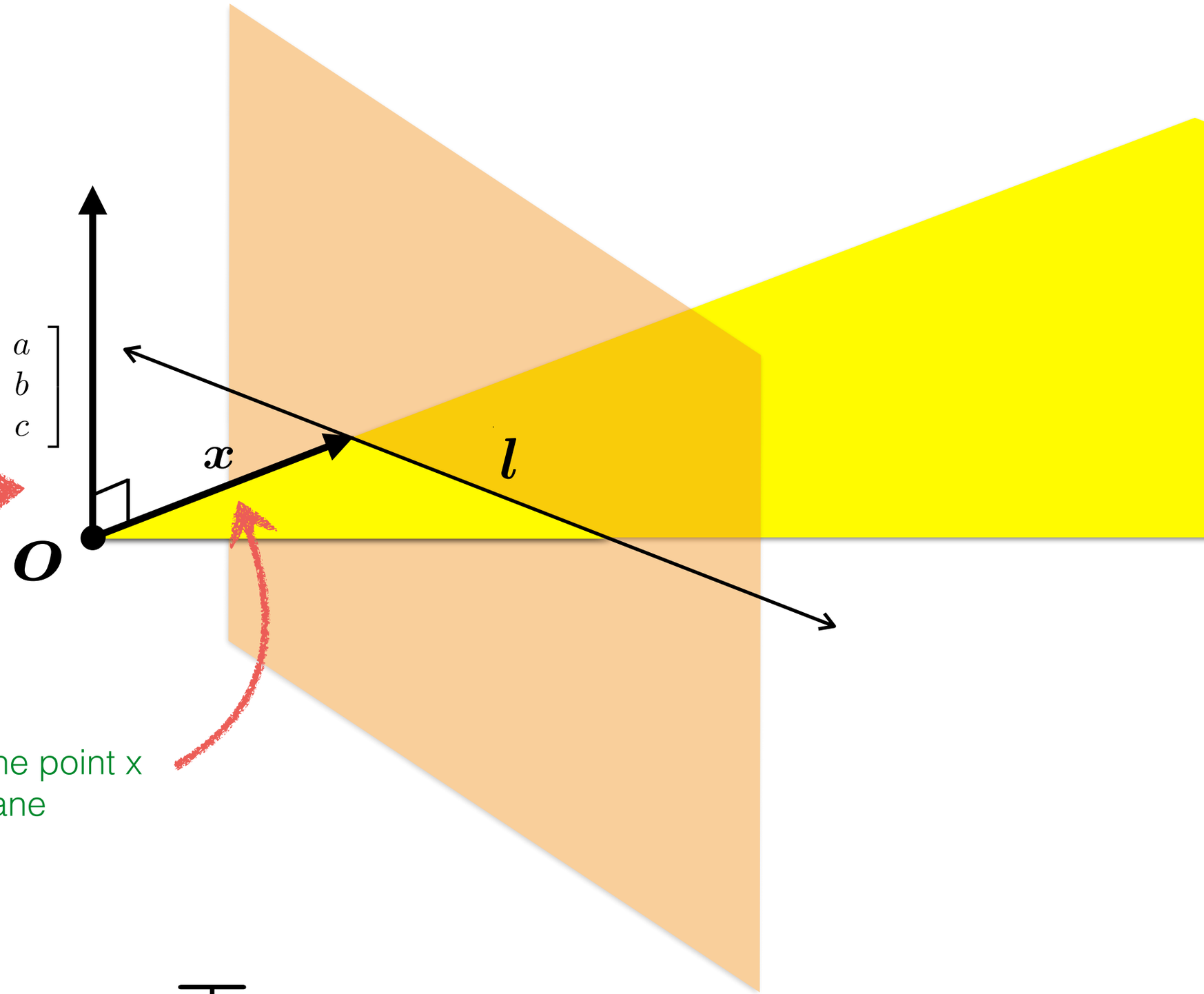


$$c \cdot a = 0 \qquad\qquad c \cdot b = 0$$

dot product of two orthogonal vectors is zero

vector representing the line is
normal (orthogonal) to the plane

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

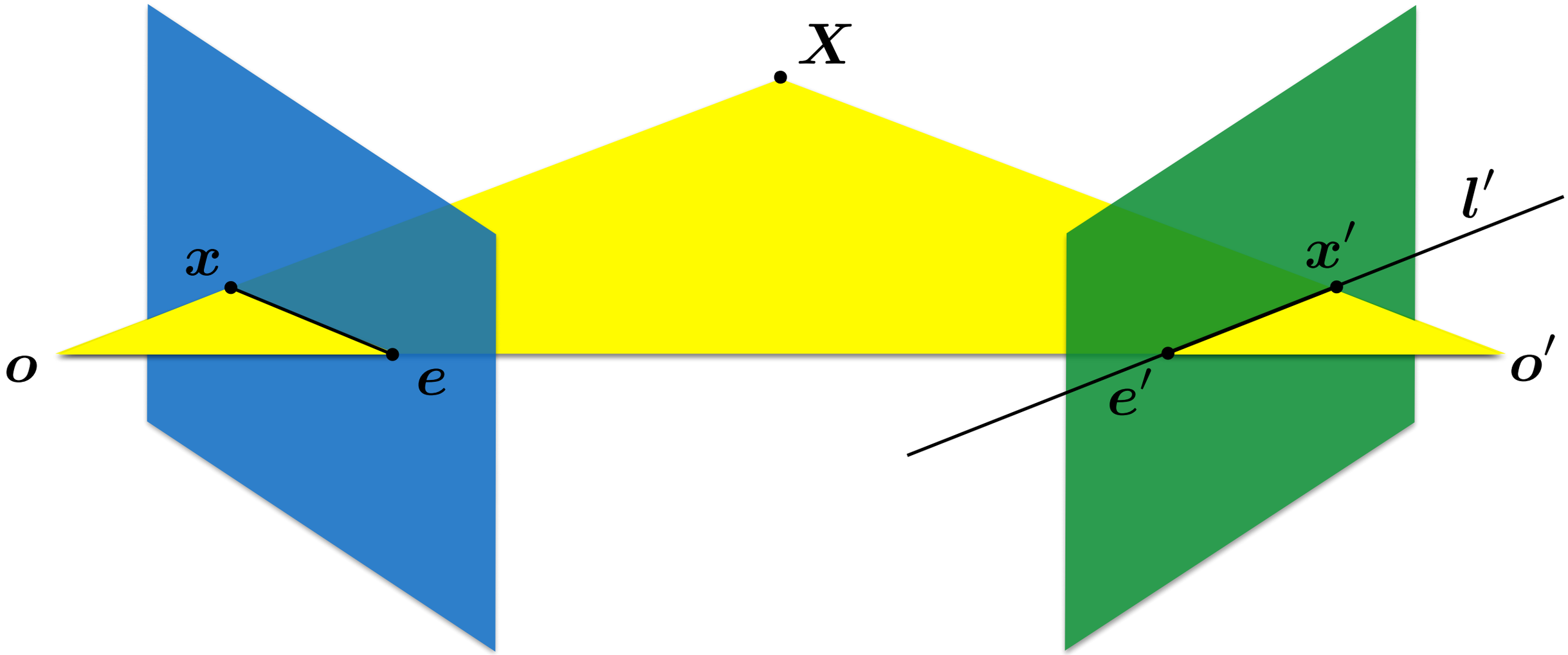


vector representing the point x
is inside the plane

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

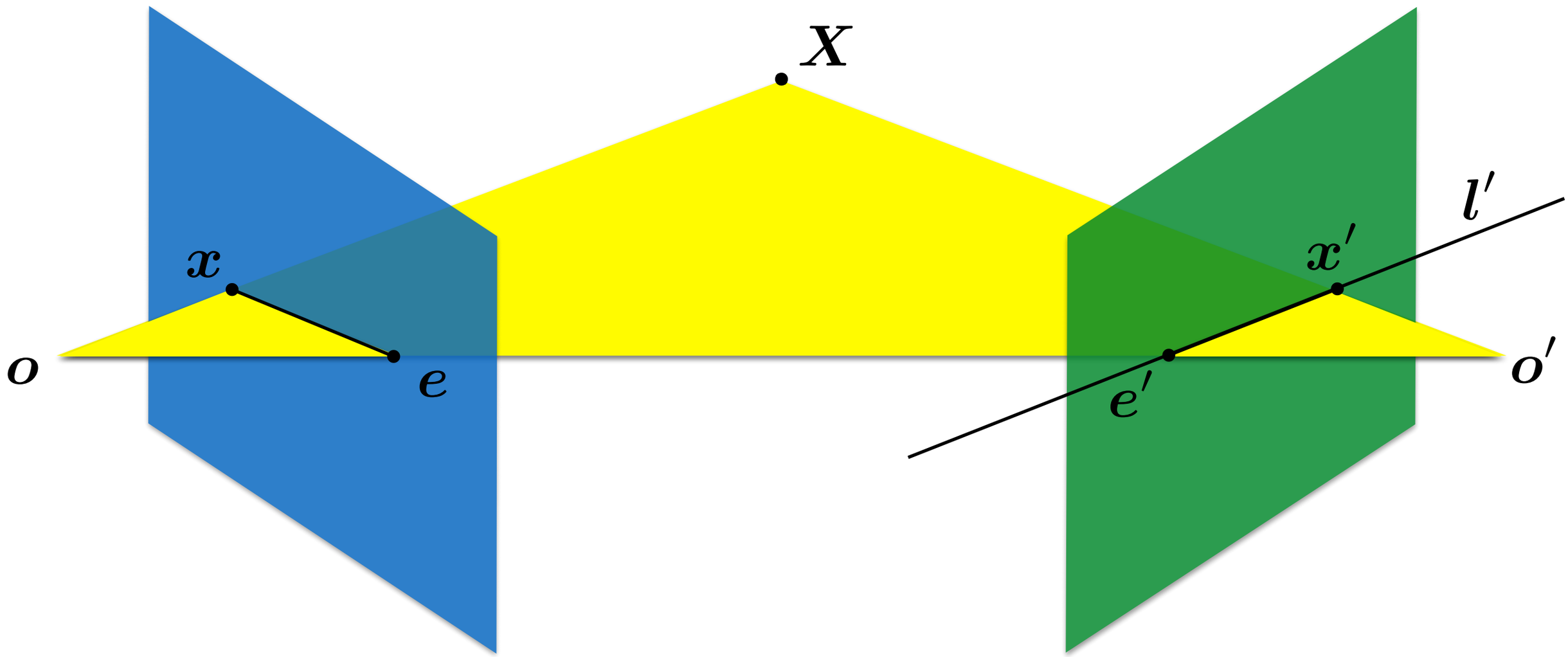
So if $\boldsymbol{x}^\top \boldsymbol{l} = 0$ and $\mathbf{E}\boldsymbol{x} = \boldsymbol{l}'$ then

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



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

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



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 view.

Essential Matrix vs Homography

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

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

They are both 3 x 3 matrices but …

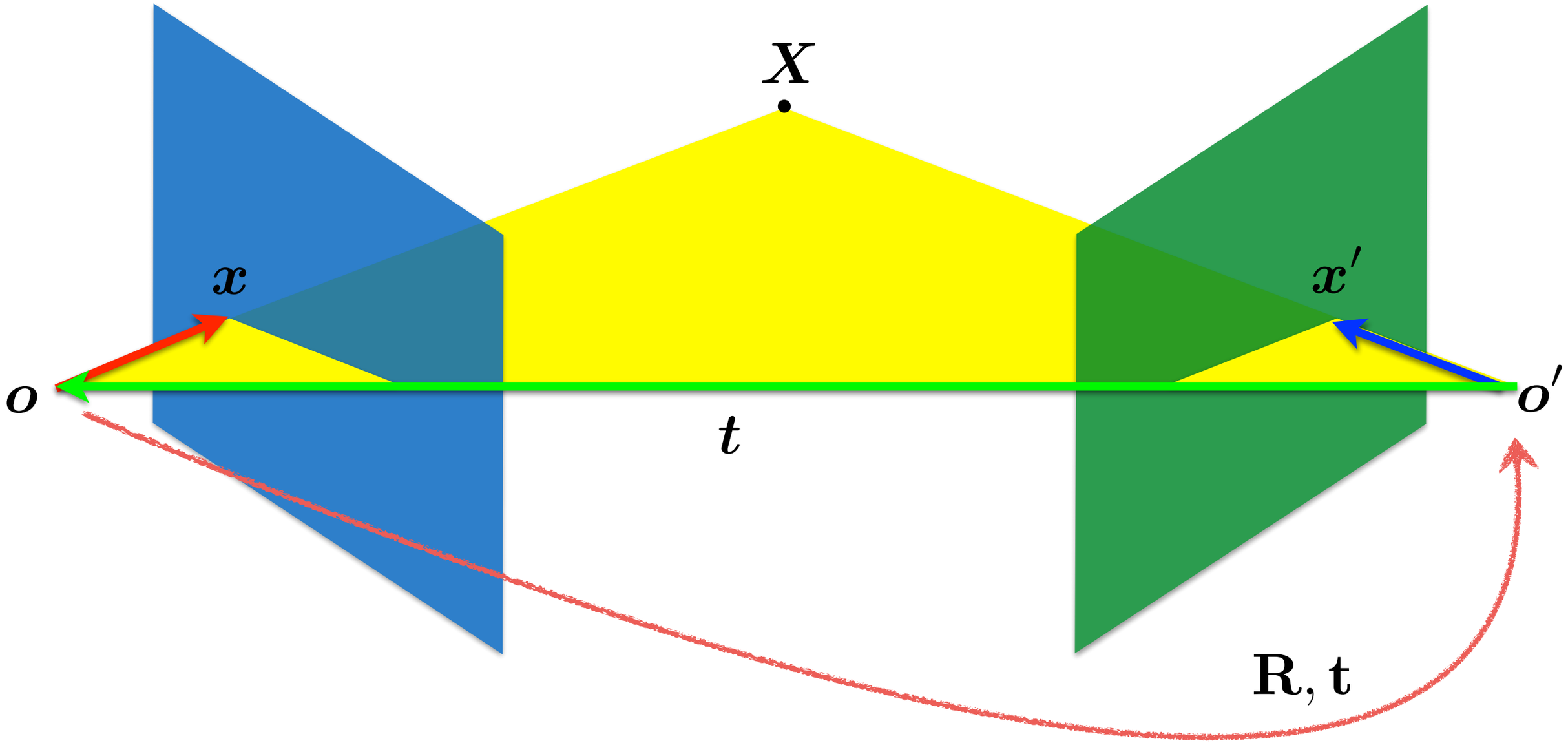
$$\boldsymbol{l}' = \mathbf{E}\boldsymbol{x}$$

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

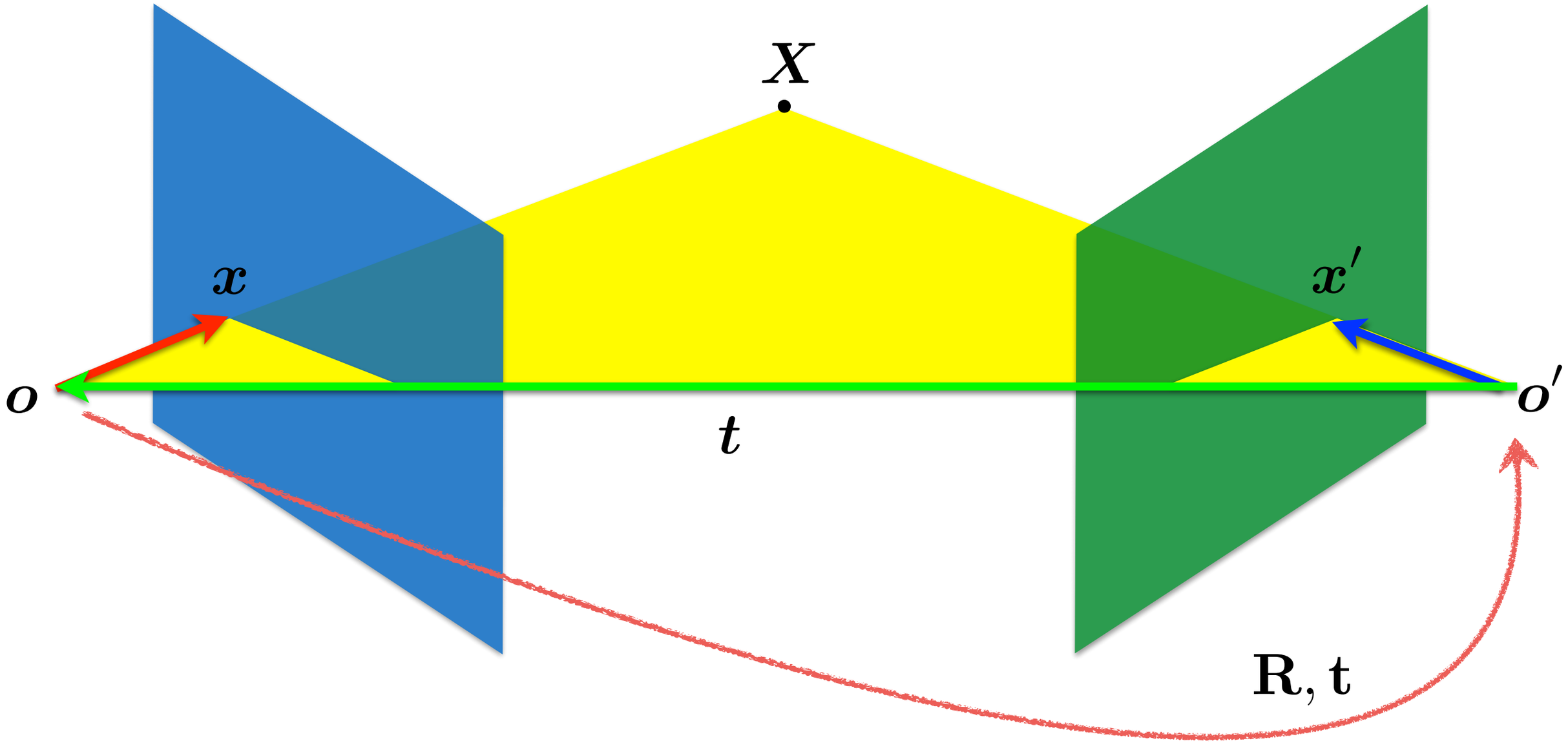
$$\boldsymbol{x}' = \mathbf{H}\boldsymbol{x}$$

Homography maps a
point to a **point**

Where does the Essential matrix come from?

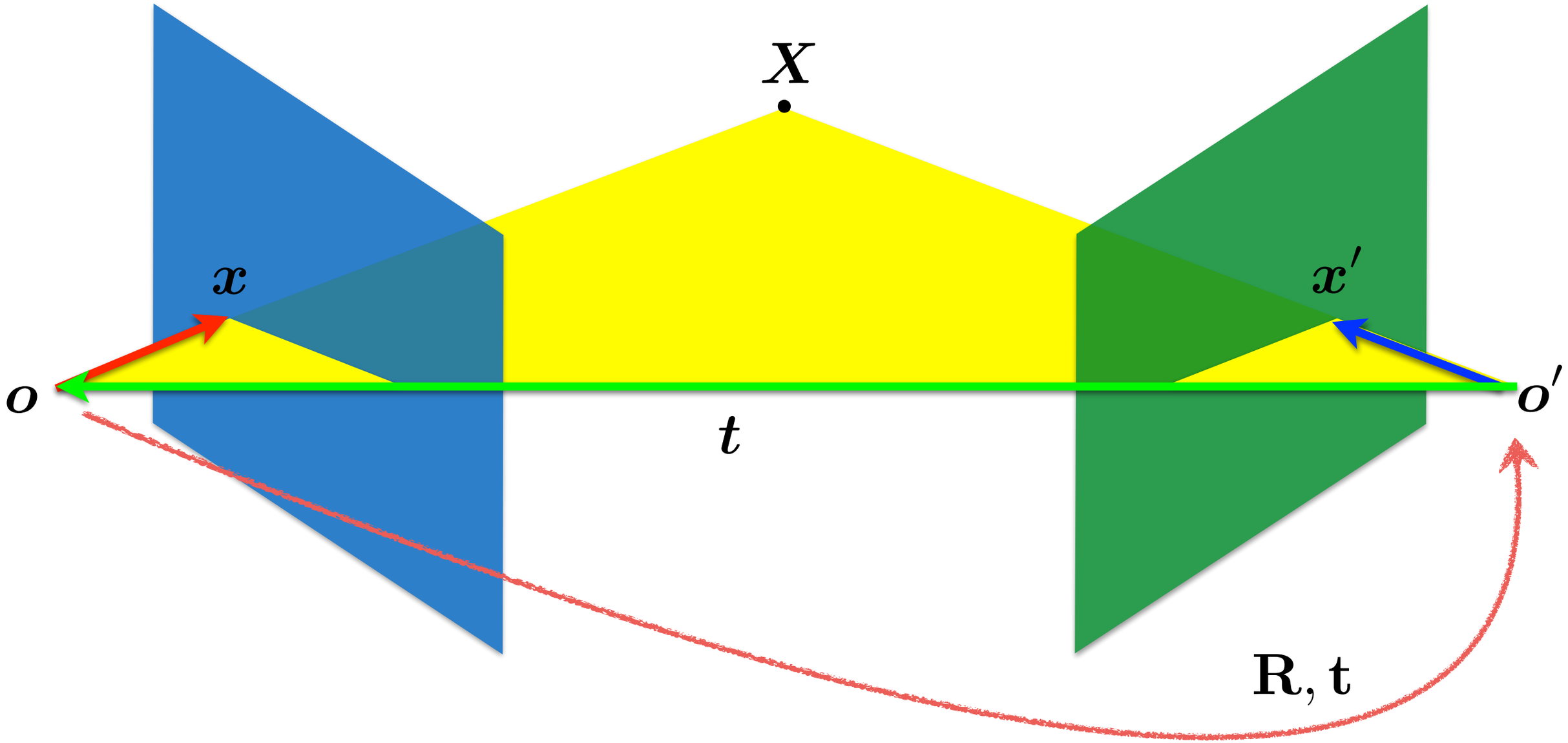


$$\boldsymbol{x}' = \mathbf{R}(\boldsymbol{x} - \boldsymbol{t})$$



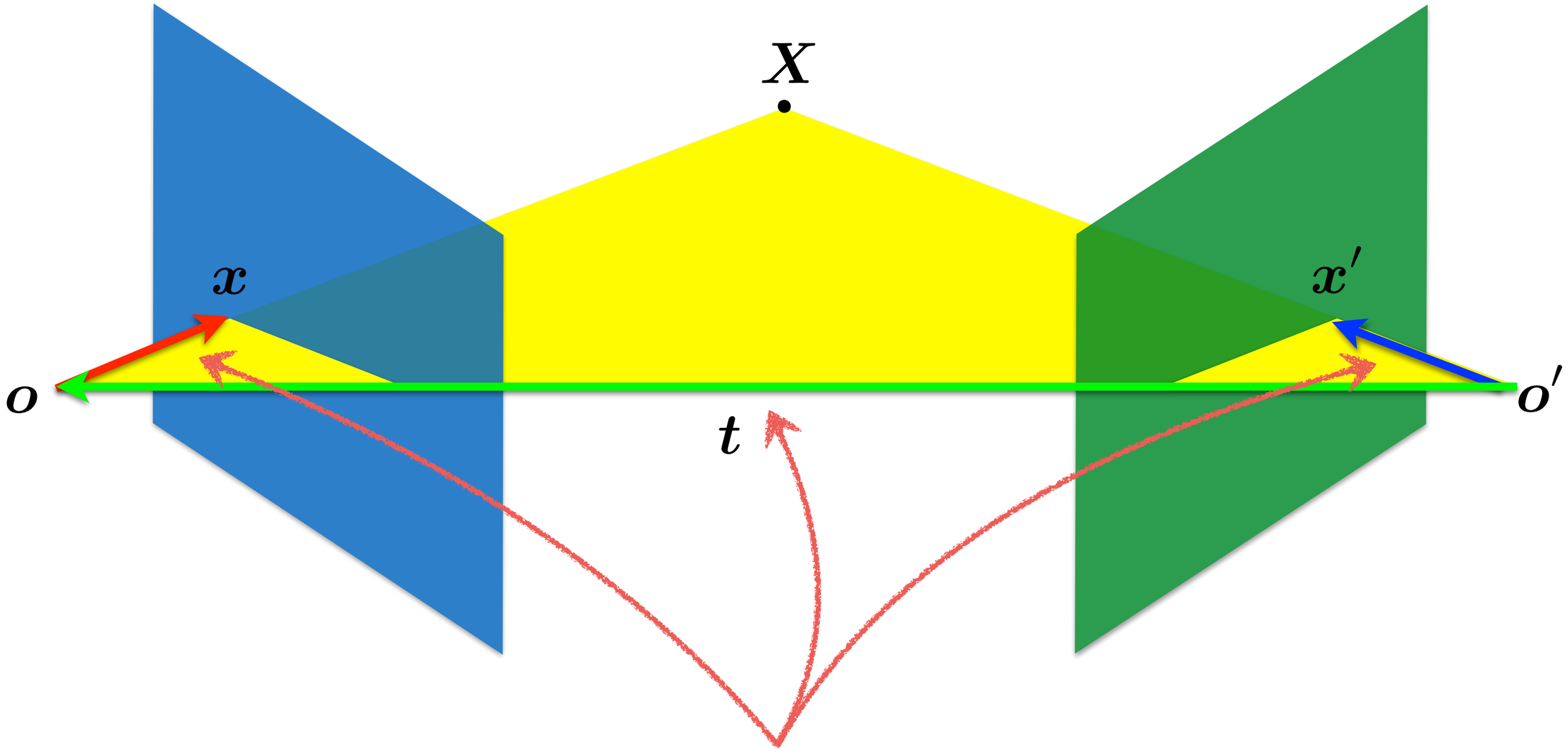
$$\boldsymbol{x}' = \mathbf{R}(\boldsymbol{x} - \boldsymbol{t})$$

Does this look familiar?



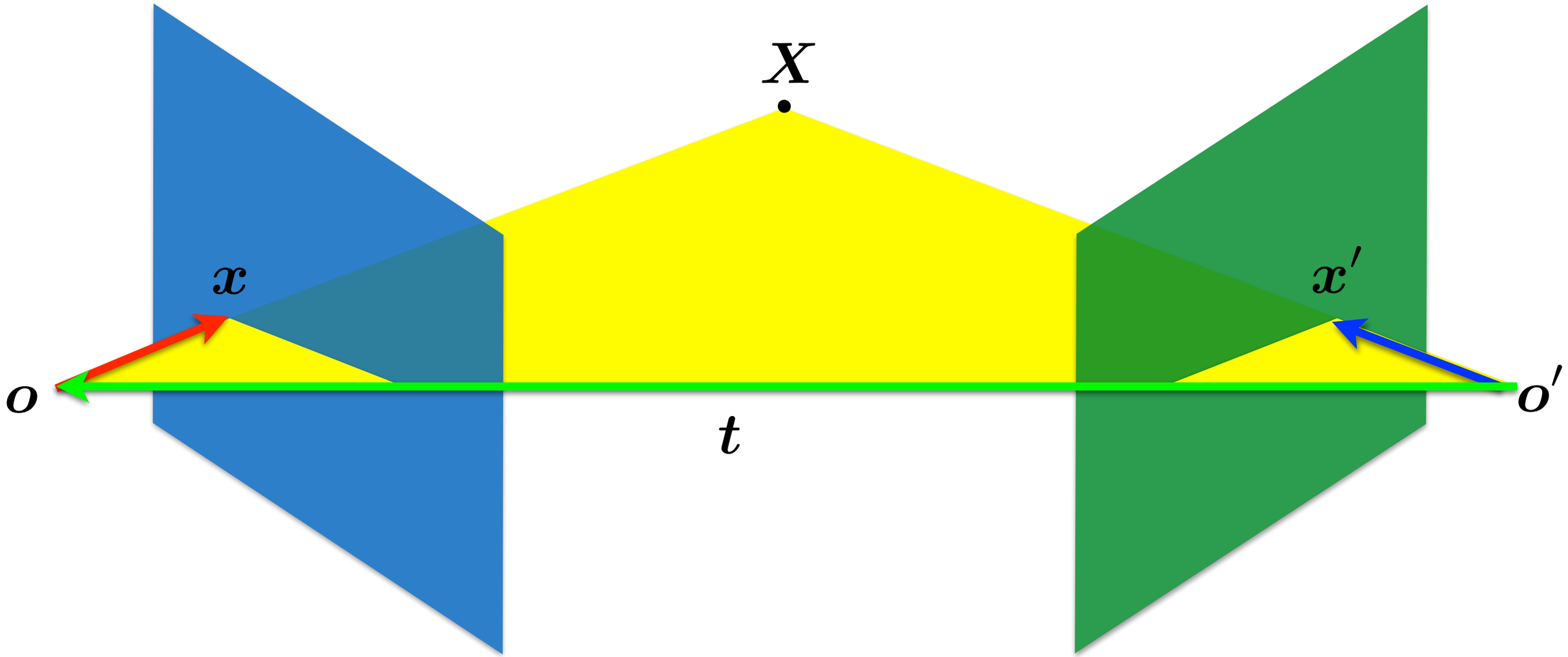
$$x' = \mathbf{R}(x - t)$$

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



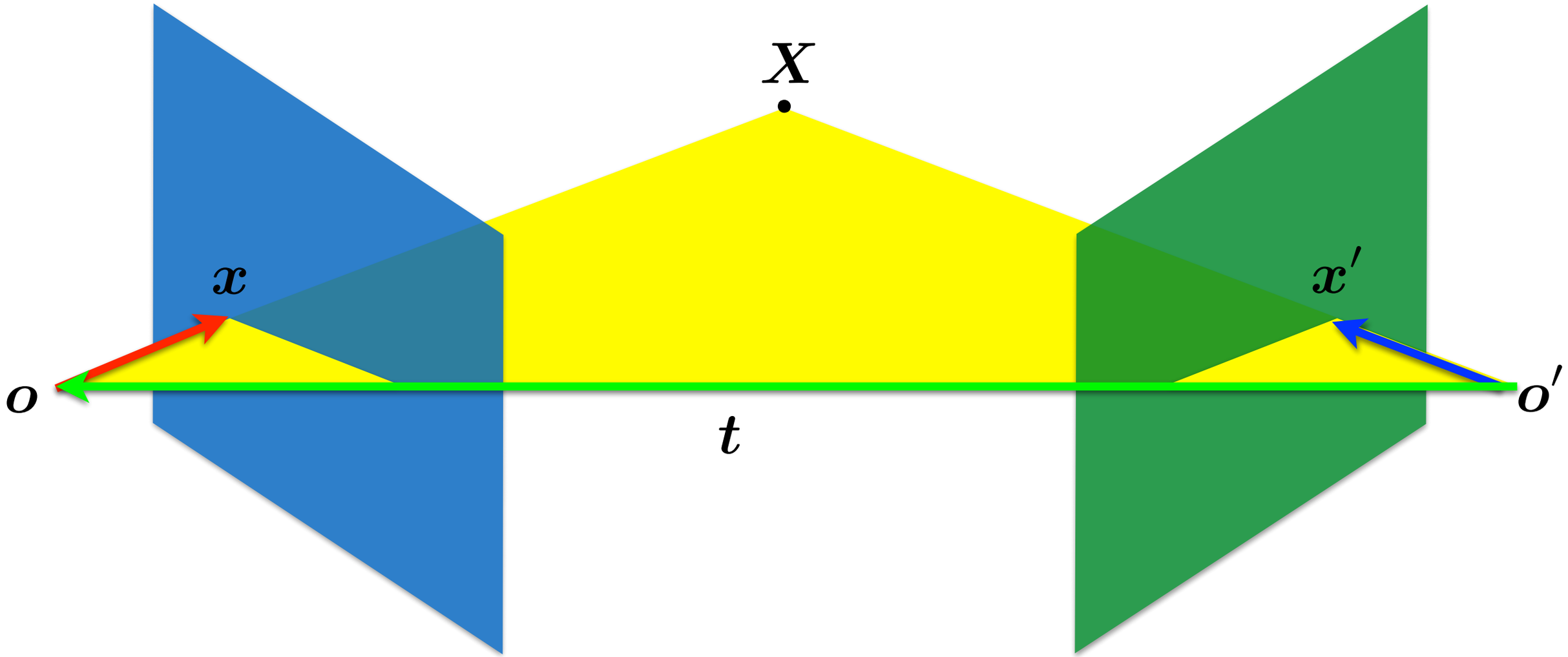
These three vectors are coplanar

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



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

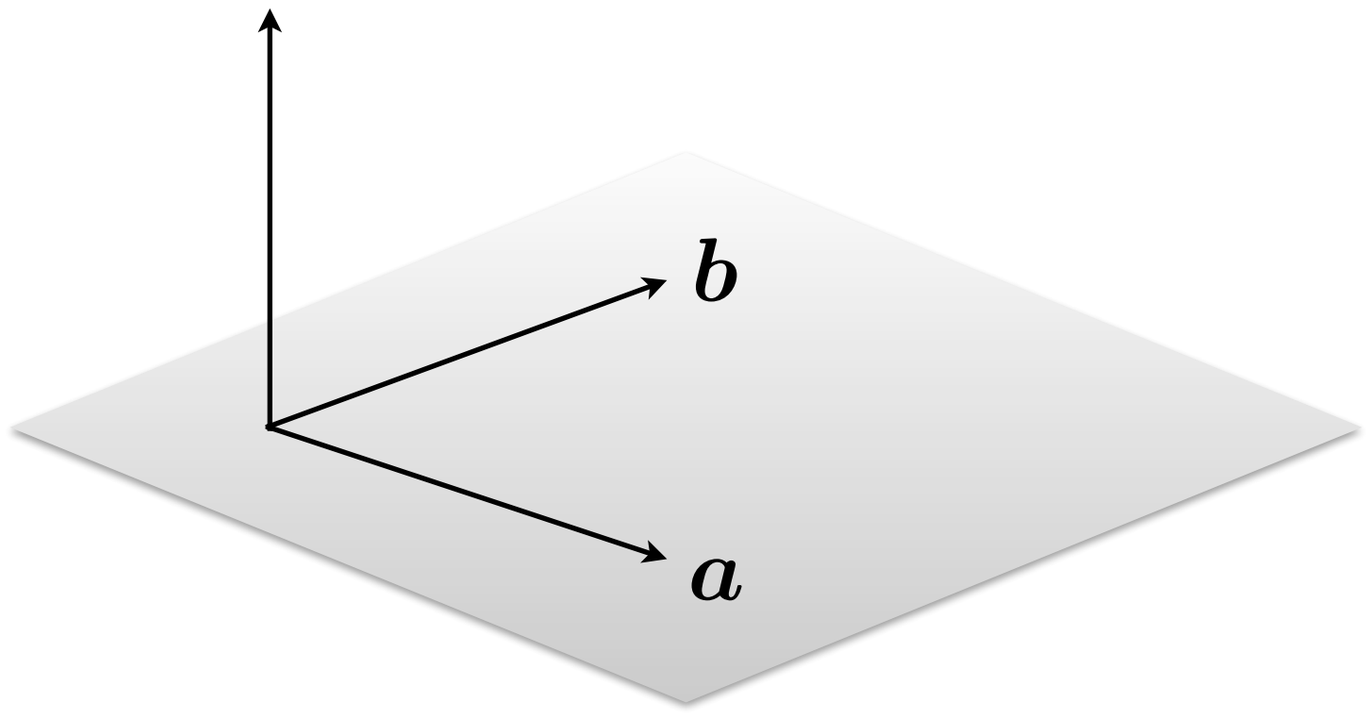
$$\boldsymbol{x}^\top (\boldsymbol{t} \times \boldsymbol{x}) = 0$$

Recall: Cross Product

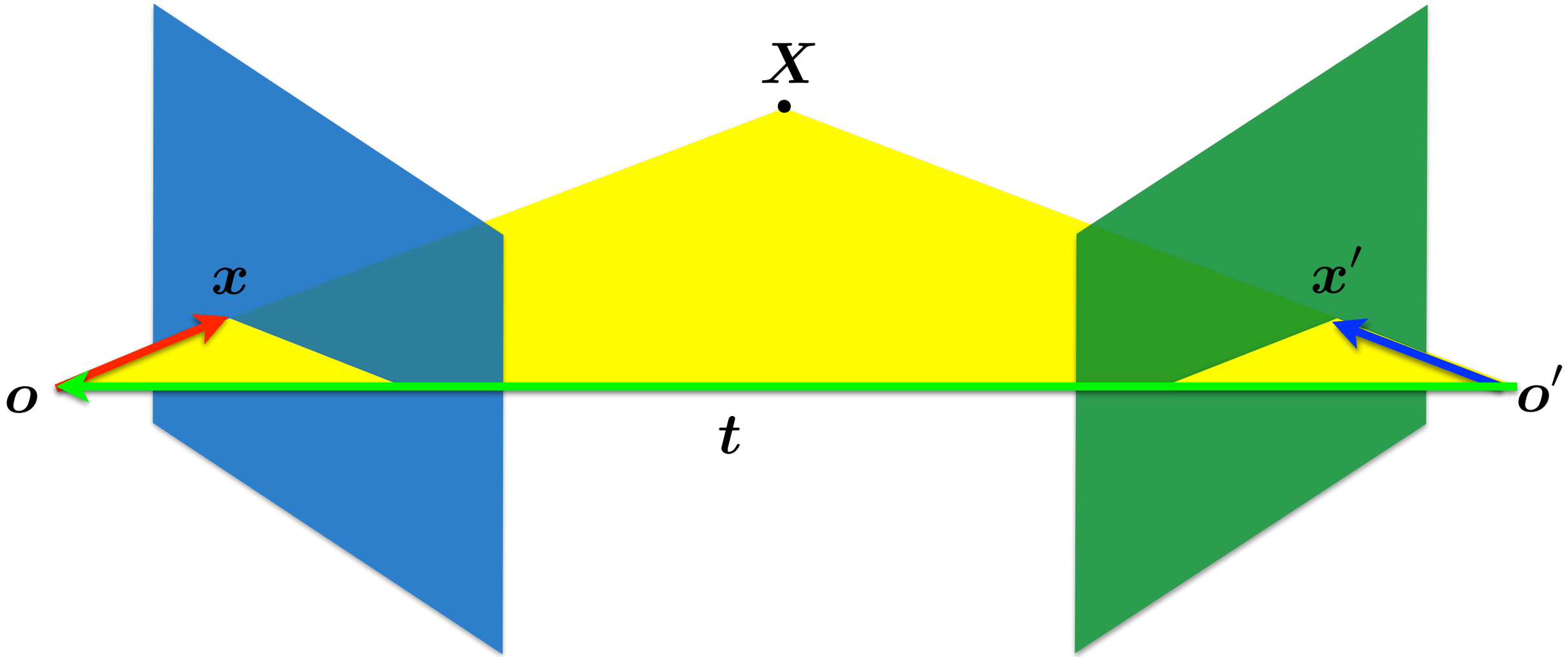
Vector (cross) product

takes two vectors and returns a vector perpendicular to both

$$\boldsymbol{c} = \boldsymbol{a} \times \boldsymbol{b}$$

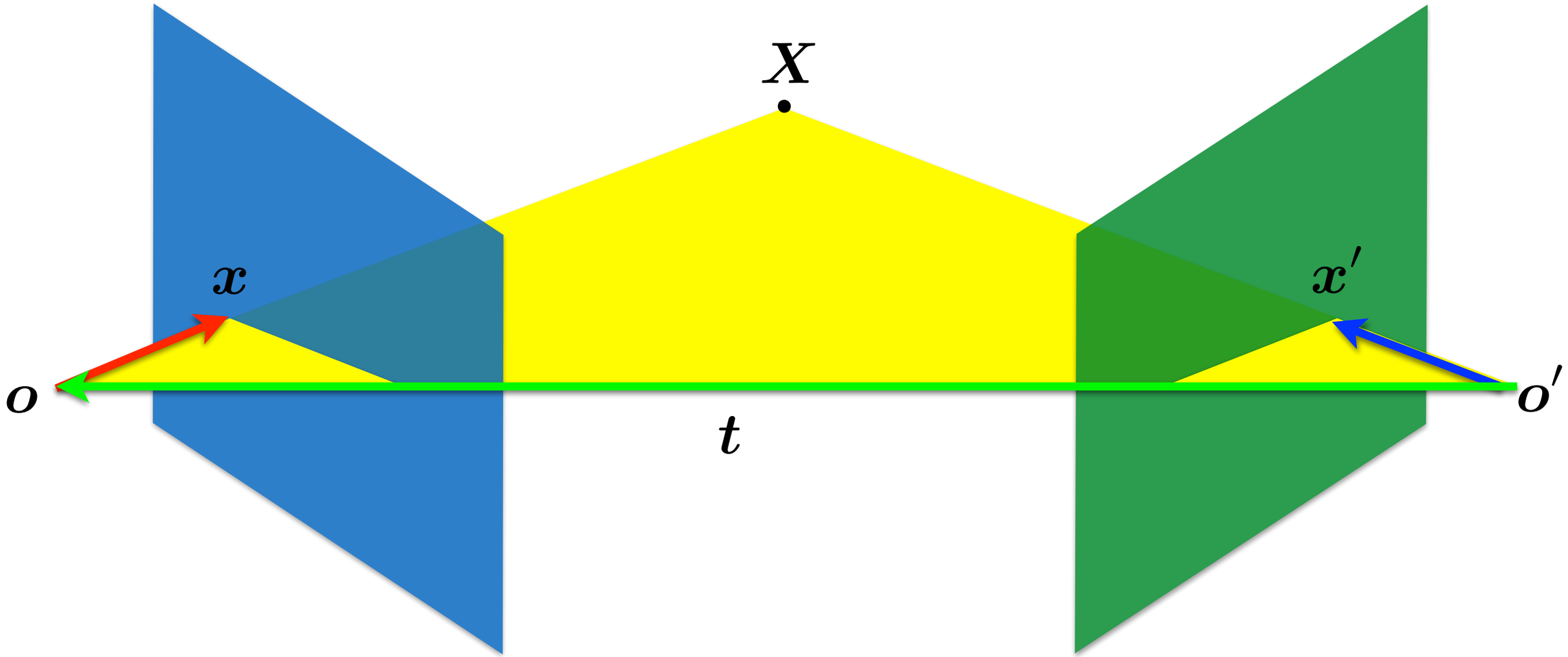


$$\boldsymbol{c} \cdot \boldsymbol{a} = 0 \qquad \boldsymbol{c} \cdot \boldsymbol{b} = 0$$



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

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

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

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

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

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

(points in normalized coordinates)

properties of the E matrix

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

Epipolar lines

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

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

(points in normalized coordinates)

properties of the E matrix

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

Epipolar lines

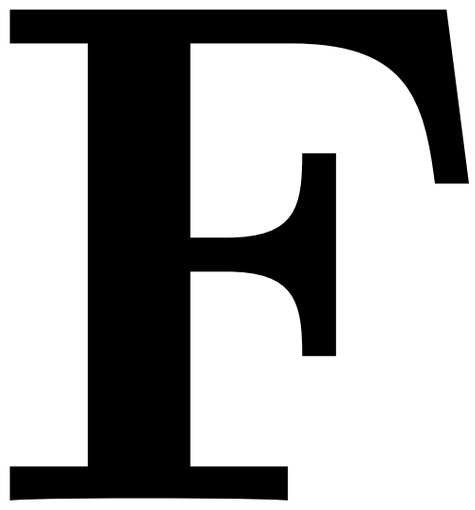
$\boldsymbol{x}^{\top}\boldsymbol{l} = 0$ $\quad$ $\boldsymbol{x}'^{\top}\boldsymbol{l}' = 0$

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

Epipoles $\boldsymbol{e}'^{\top}\mathbf{E} = \mathbf{0}$ $\quad$ $\mathbf{E}\boldsymbol{e} = \mathbf{0}$

(points in normalized camera coordinates)

How do you generalize to uncalibrated cameras?

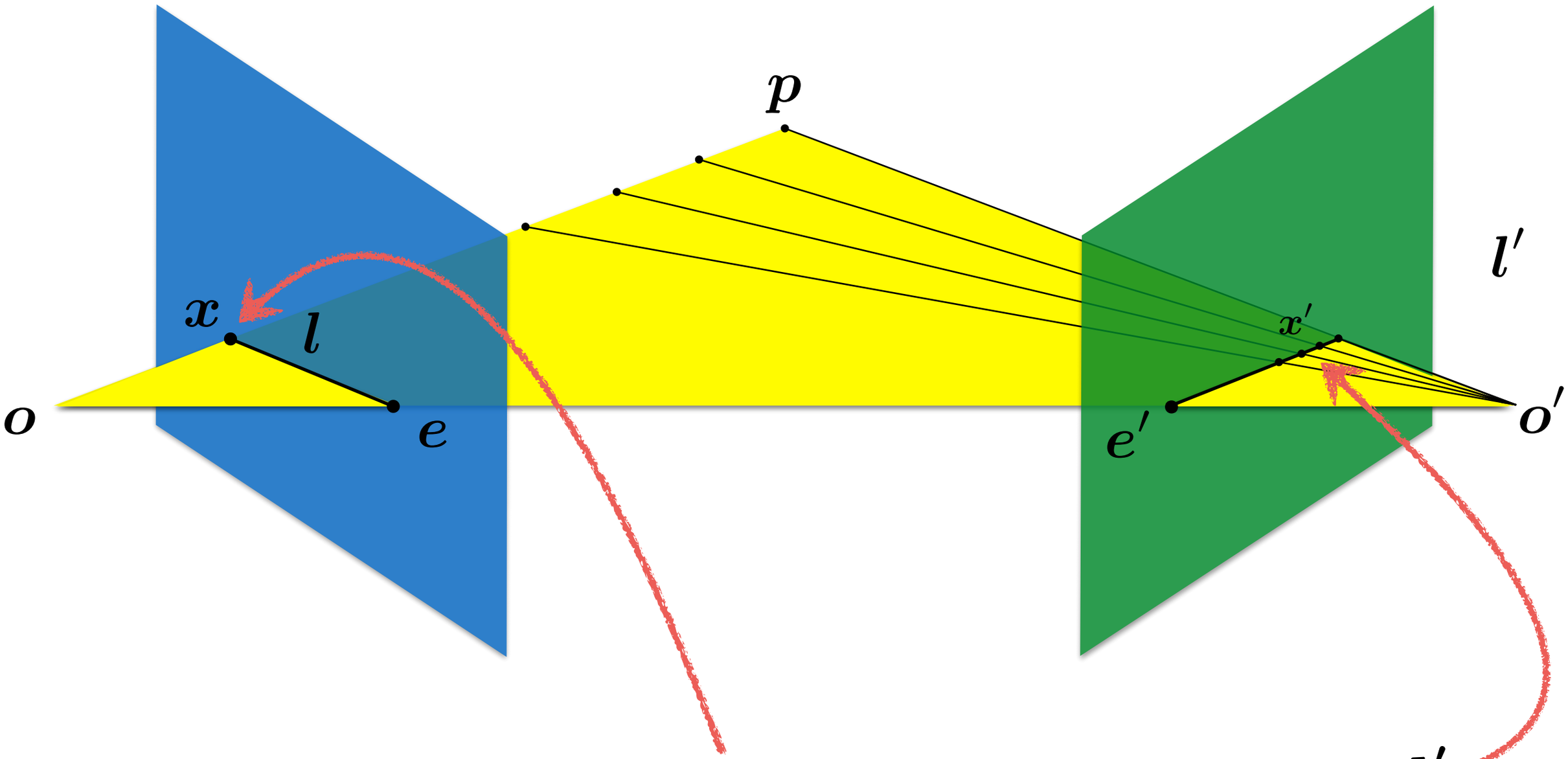


Fundamental Matrix

16-385 Computer Vision (Kris Kitani)

Carnegie Mellon University

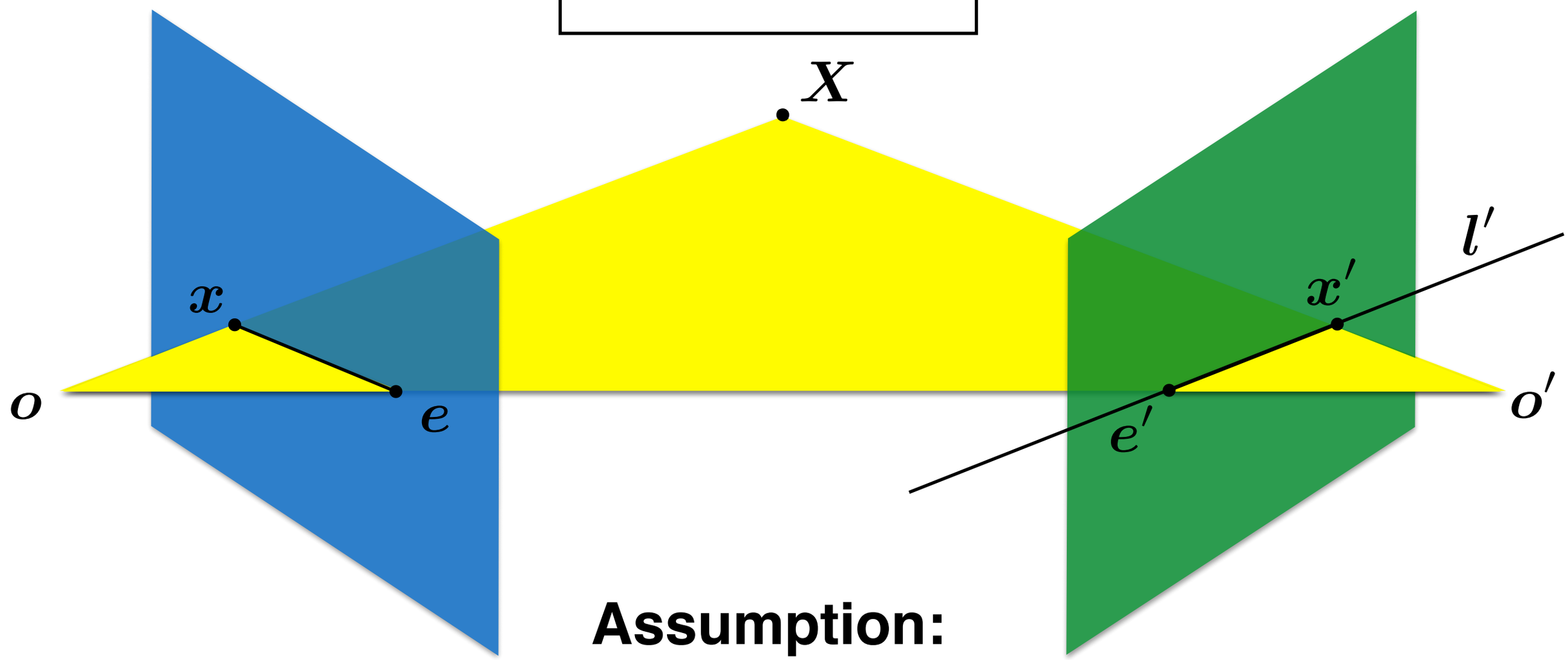
Recall:Epipolar constraint



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.

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



Assumption:
points aligned to camera coordinate axis (calibrated camera)

How do you generalize to uncalibrated cameras?

The

Fundamental matrix

is a

generalization

of the

Essential matrix,

where the assumption of

calibrated cameras

is removed

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

The Essential matrix operates on image points expressed in
normalized coordinates
(points have been aligned (normalized) to camera coordinates)

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

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

camera point

image point

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

The Essential matrix operates on image points expressed in
normalized coordinates
(points have been aligned (normalized) to camera coordinates)

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

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

camera point

image point

Writing out the epipolar constraint in terms of image coordinates

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

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

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

Same equation works in image coordinates!

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

it maps pixels to epipolar lines

properties of the F matrix

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

Epipolar lines

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

$\boldsymbol{l}' = F\boldsymbol{x}$ $\quad$ $\boldsymbol{l} = F^{T}\boldsymbol{x}'$

Epipoles $\quad \boldsymbol{e}'^{\top} F = \mathbf{0}$ $\quad$ $F\boldsymbol{e} = \mathbf{0}$

(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

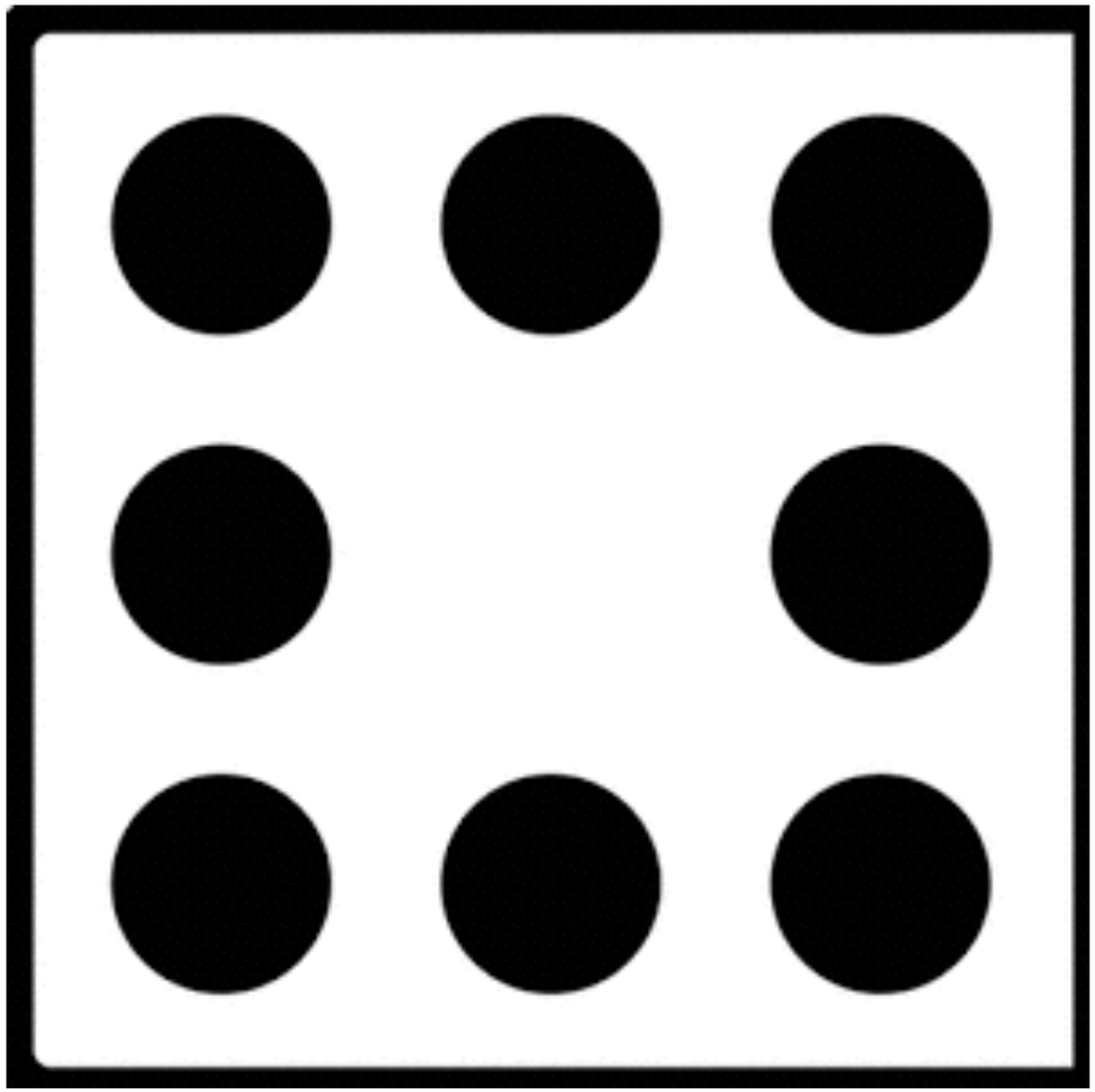
$$\mathbf{F} = \mathbf{K}'^{-\top}\mathbf{E}\mathbf{K}^{-1}$$

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

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

16-385 Computer Vision (Kris Kitani)

Carnegie Mellon University

Fundamental Matrix Estimation

Given a set of matched *image* points

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

Estimate the Fundamental Matrix

$$\mathbf{F}$$

What's the relationship between F and x?

Assume you have *M* point correspondences

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

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* point correspondences

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

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

Assume you have *M* point correspondences

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

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

Assume you have *M* point correspondences

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

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* point correspondences

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

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

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

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

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

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

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

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

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

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?

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

Total Least Squares

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

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

How do you solve a homogeneous linear system?

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

Total Least Squares

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

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

S

How do you solve a homogeneous linear system?

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

Total Least Squares

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

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

SV

How do you solve a homogeneous linear system?

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

Total Least Squares

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

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

SVD!

Eight-Point Algorithm

0. (Normalize points)

1. Construct the M x 9 matrix **A**

2. Find the SVD of $\mathbf{A}^T\mathbf{A}$

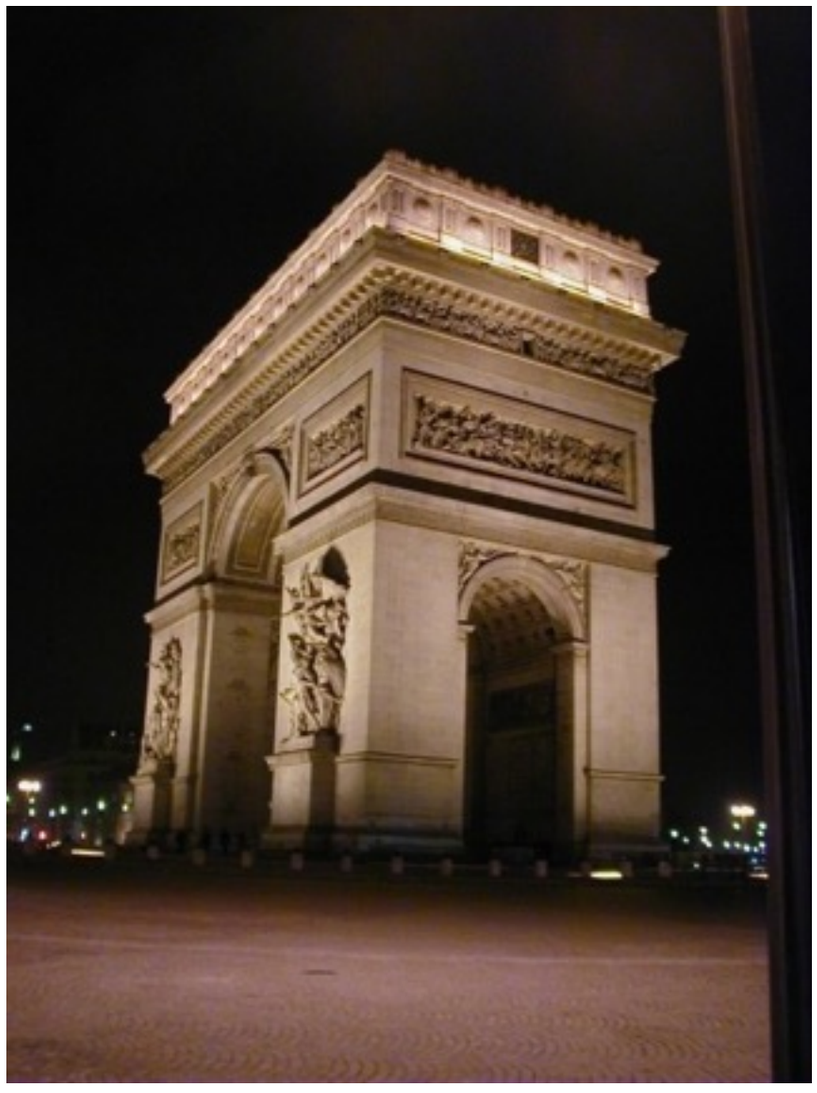
3. Entries of **F** are the elements of column of

   V corresponding to the least singular value

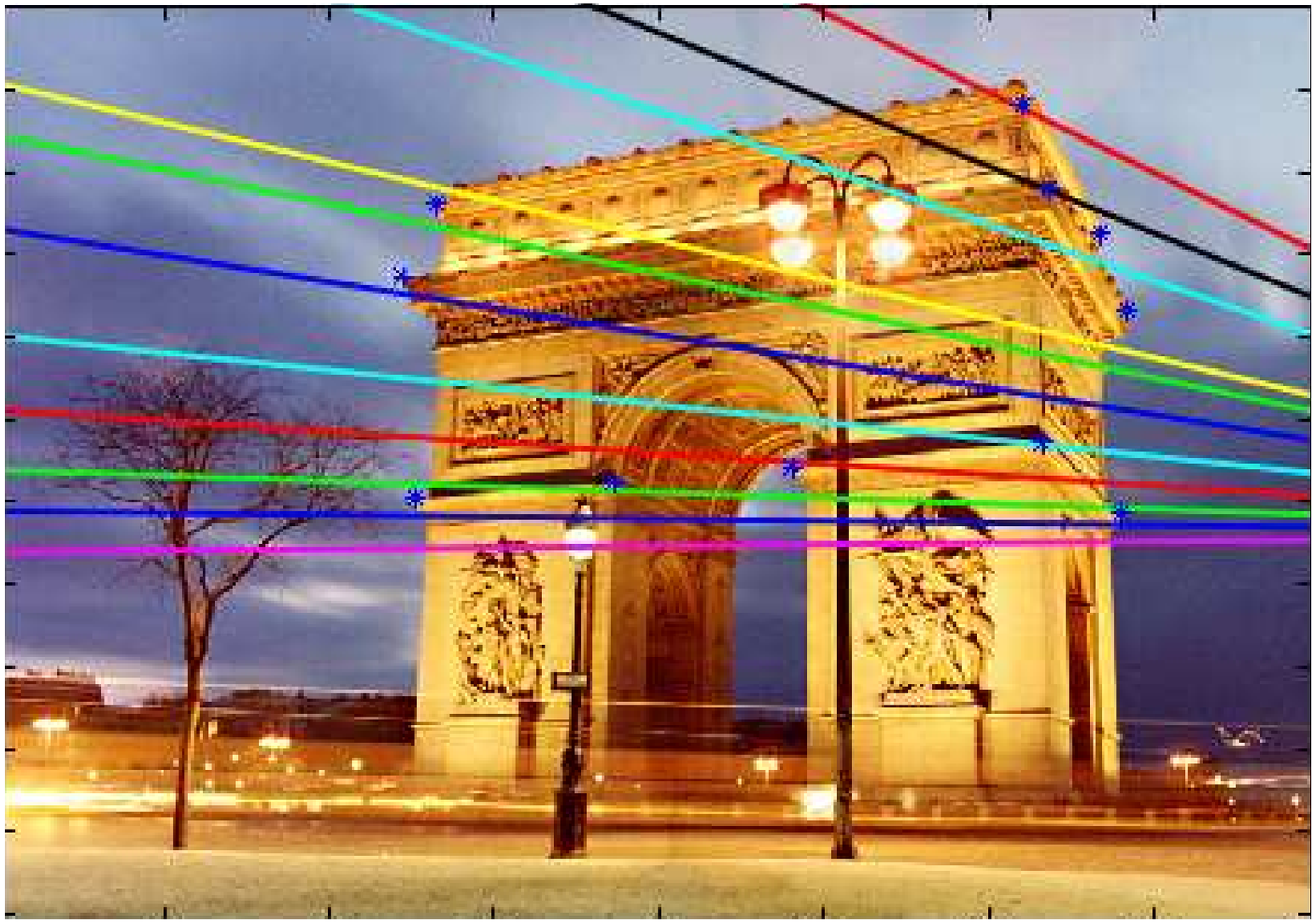
4. (Enforce rank 2 constraint on F)

5. (Un-normalize F)

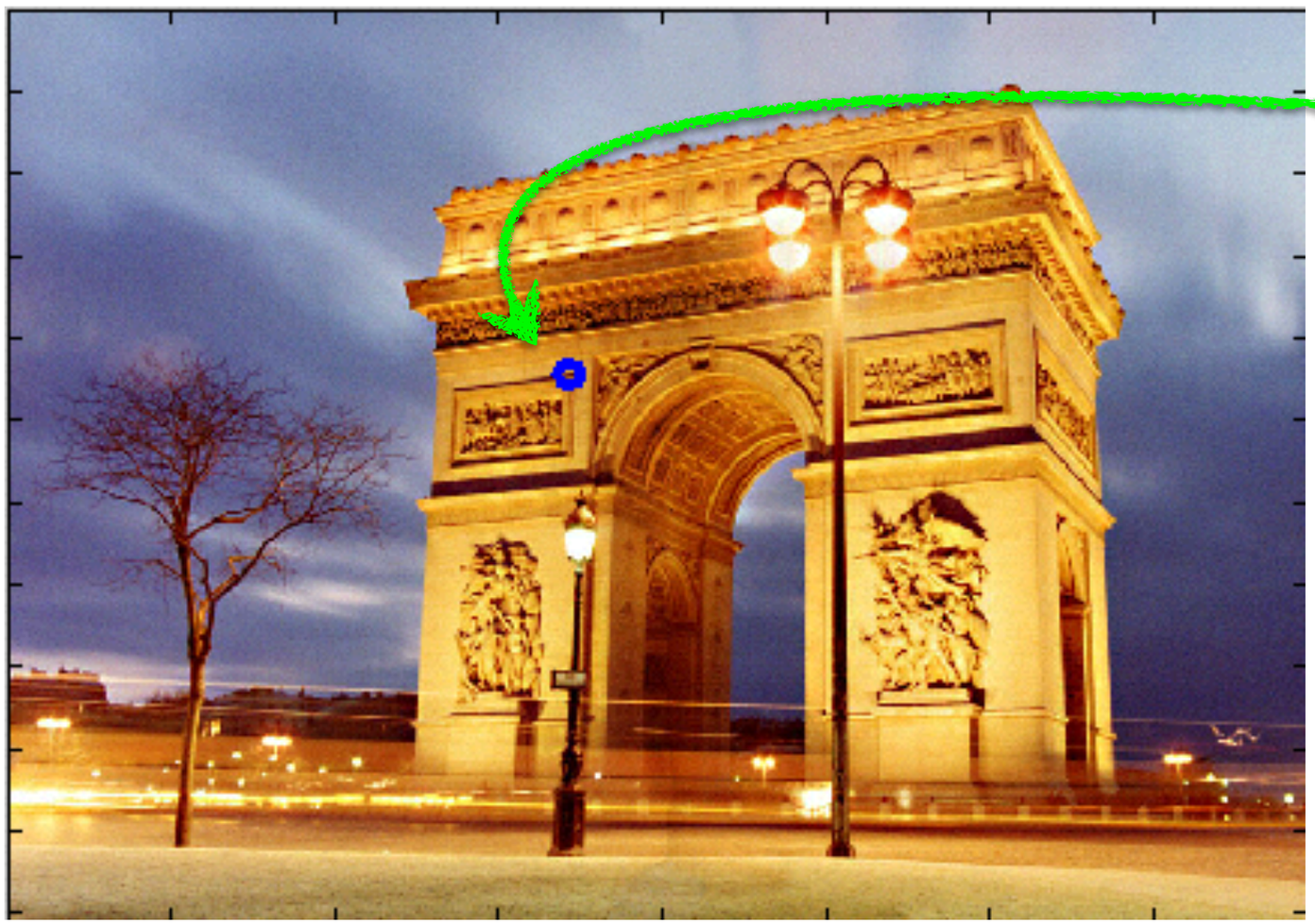
Example



epipolar lines



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



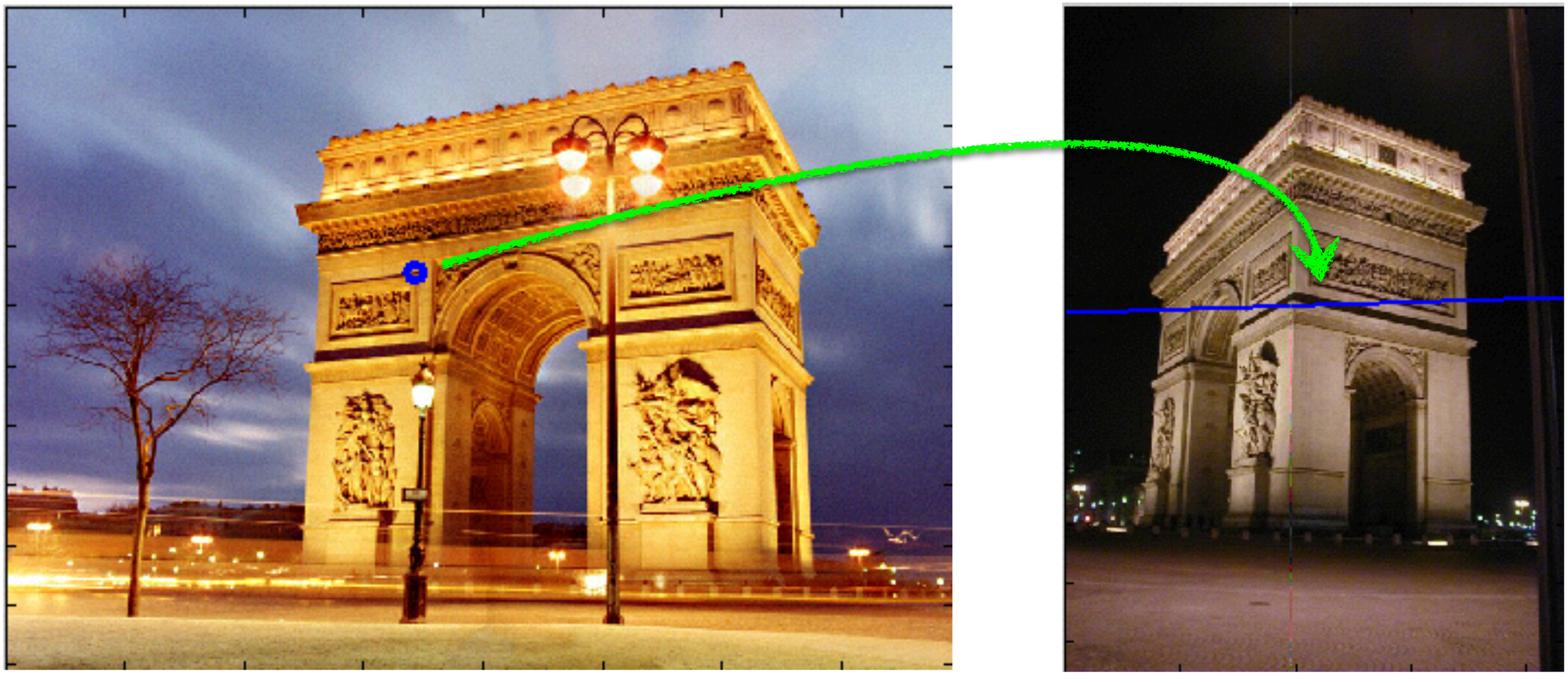
$$\boldsymbol{x} = \begin{bmatrix} 343.53 \\ 221.70 \\ 1.0 \end{bmatrix}$$

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

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

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

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



Where is the epipole?

How would you compute it?

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

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

How would you solve for the epipole?

(hint: this is a homogeneous linear system)

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

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

How would you solve for the epipole?

(hint: this is a homogeneous linear system)

S

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

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

How would you solve for the epipole?

(hint: this is a homogeneous linear system)

SV

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

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

How would you solve for the epipole?

(hint: this is a homogeneous linear system)

SVD!

```
>> [u,d] = eigs(F' * F)
```

eigenvectors

```
u =
    -0.0013    0.2586   -0.9660
     0.0029   -0.9660   -0.2586
     1.0000    0.0032   -0.0005
```

eigenvalue

```
d = 1.0e8*
   -1.0000         0         0
         0   -0.0000         0
         0         0   -0.0000
```

```
>> [u,d] = eigs(F' * F)
```

eigenvectors

```
u =
    -0.0013    0.2586   -0.9660
     0.0029   -0.9660   -0.2586
     1.0000    0.0032   -0.0005
```

eigenvalue

```
d = 1.0e8*
   -1.0000         0         0
         0   -0.0000         0
         0         0   -0.0000
```

```
>> [u,d] = eigs(F' * F)
```

eigenvectors

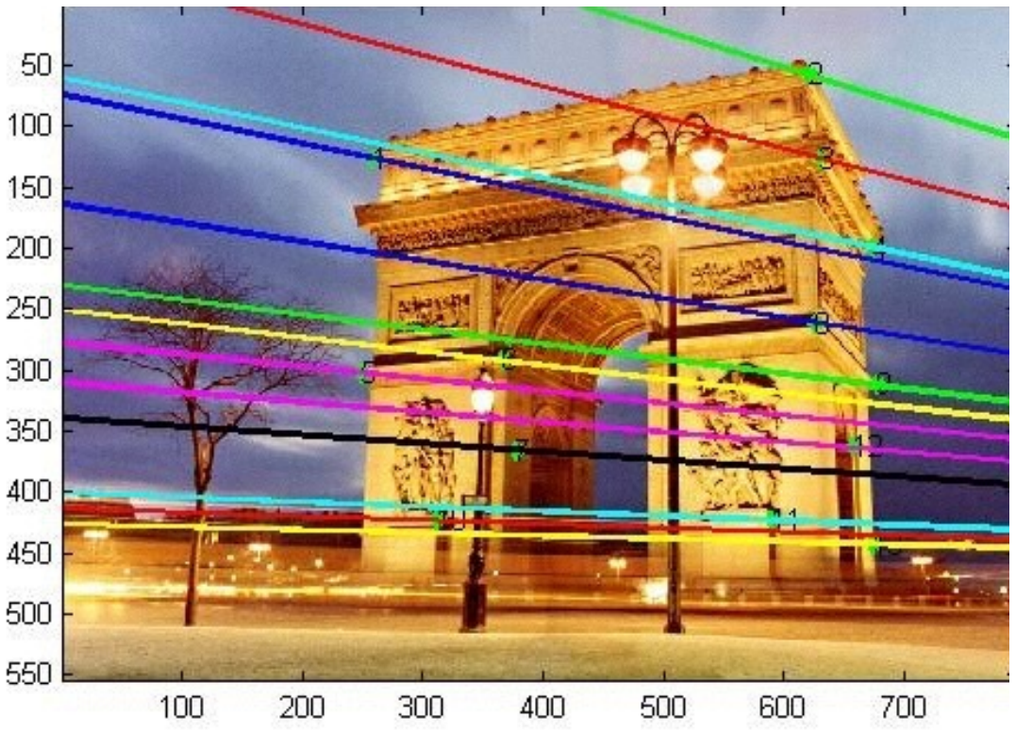
```
u =
    -0.0013    0.2586   -0.9660
     0.0029   -0.9660   -0.2586
     1.0000    0.0032   -0.0005
```

eigenvalue

```
d = 1.0e8*
   -1.0000         0         0
         0   -0.0000         0
         0         0   -0.0000
```

Eigenvector associated with smallest eigenvalue

```
>> uu = u(:,3)
(  -0.9660    -0.2586    -0.0005)
```




```
>> [u,d] = eigs(F' * F)
```

eigenvectors

```
u =
    -0.0013    0.2586   -0.9660
     0.0029   -0.9660   -0.2586
     1.0000    0.0032   -0.0005
```

eigenvalue

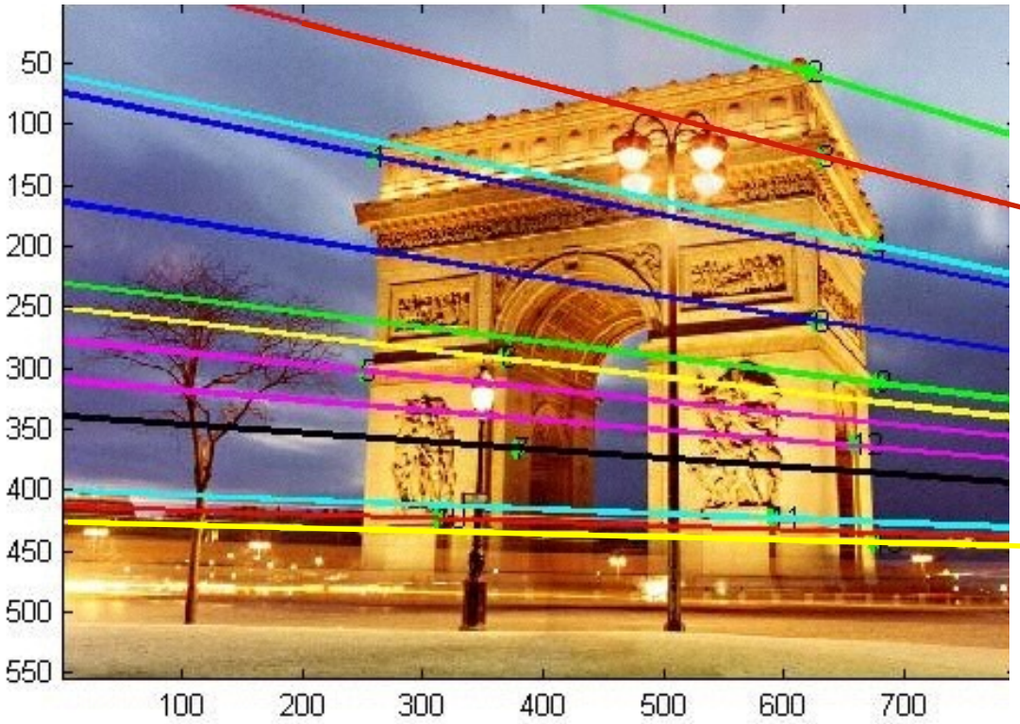
```
d = 1.0e8*
   -1.0000         0         0
         0   -0.0000         0
         0         0   -0.0000
```

Eigenvector associated with smallest eigenvalue

```
>> uu = u(:,3)
(  -0.9660    -0.2586    -0.0005)
```

Epipole projected to image coordinates

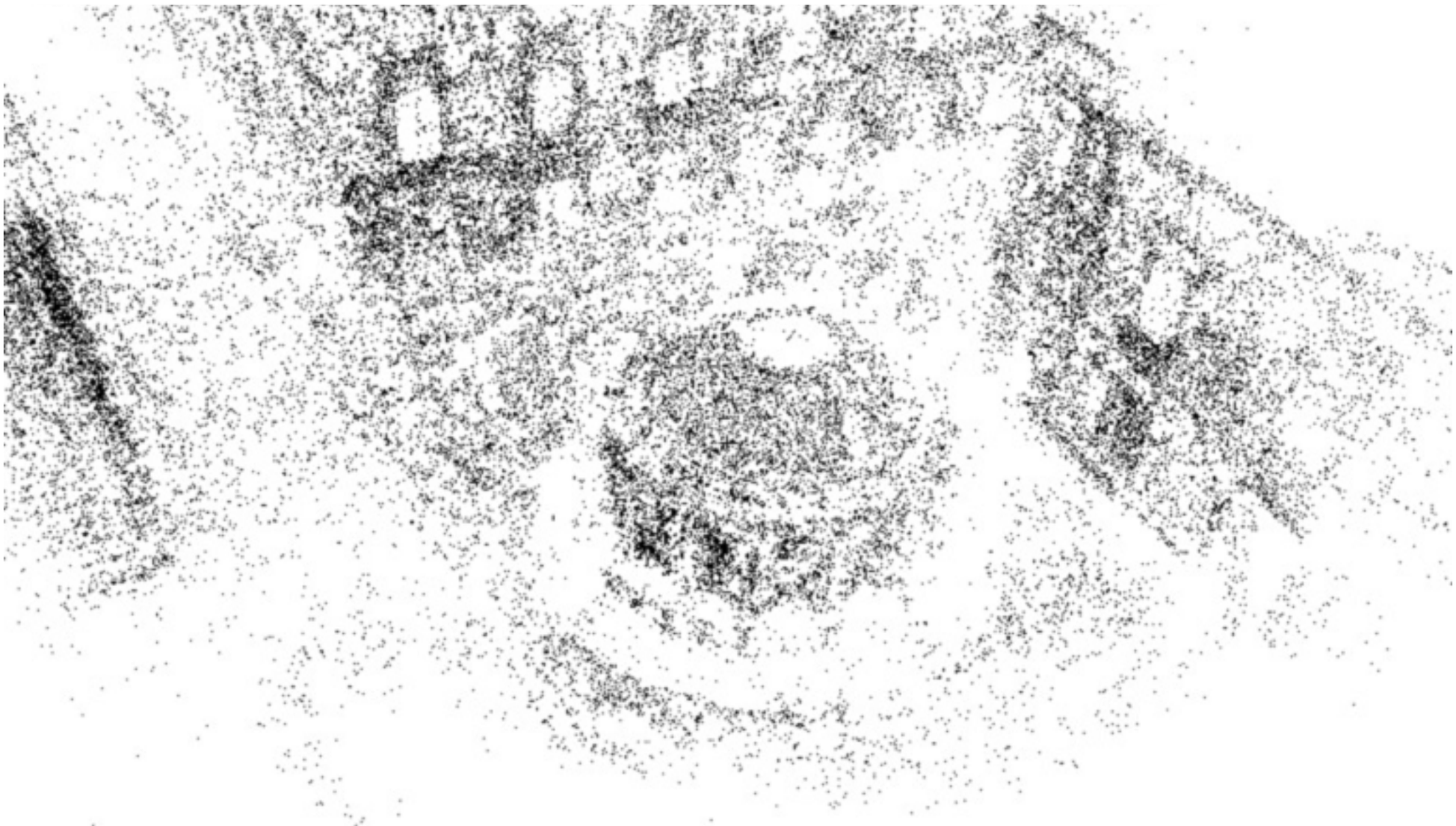
```
>> uu / uu(3)
(1861.02     498.21     1.0)
```



this is where the other picture is being taken

Epipole projected to image coordinates

```
>> uu / uu(3)
(1861.02      498.21      1.0)
```



Reconstruction

16-385 Computer Vision (Kris Kitani)

Carnegie Mellon University

	Structure (scene geometry)	Motion (camera geometry)	Measurements
Pose Estimation	known	estimate	3D to 2D correspondences
Triangulation	estimate	known	2D to 2D coorespondences
Reconstruction	estimate	estimate	2D to 2D coorespondences

Reconstruction

(2 view structure from motion)

Given a set of matched points

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

Estimate the camera matrices

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

Estimate the 3D point

$$\mathbf{X}$$

Reconstruction

(2 view structure from motion)

Given a set of matched points

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

Estimate the camera matrices

$$\mathbf{P}, \mathbf{P}'$$ ← ‘motion’ (of the cameras)

Estimate the 3D point

$$\mathbf{X}$$ ← ‘structure’

Procedure for Reconstruction

1. Compute the Fundamental Matrix **F** from points correspondences
   8-point algorithm

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

Procedure for Reconstruction

1. Compute the Fundamental Matrix **F** from points correspondences
   8-point algorithm

2. Compute the camera matrices **P** from the Fundamental matrix
   $\mathbf{P} = [\, \mathbf{I} \mid \mathbf{0} \,]$ and $\mathbf{P'} = [\, [\mathbf{e'}_x]\mathbf{F} \mid \mathbf{e'} \,]$

Camera matrices corresponding to the fundamental matrix **F** may be chosen as

$$\mathbf{P} = [\mathbf{I}|\mathbf{0}] \qquad \mathbf{P}' = [[\boldsymbol{e}_\times]\mathbf{F}|\boldsymbol{e}']$$

(See Hartley and Zisserman C.9 for proof)

Decomposing **F** into **R** and **T**

If we have calibrated cameras we have $\mathbf{K}$ and $\mathbf{K}'$

Essential matrix: $\mathbf{E} = \mathbf{K}'^{\top}\mathbf{F}\mathbf{K}$

Decomposing **F** into **R** and **T**

If we have calibrated cameras we have $\mathbf{K}$ and $\mathbf{K}'$

Essential matrix: $\mathbf{E} = \mathbf{K}'^{\top}\mathbf{F}\mathbf{K}$

SVD: $\mathbf{E} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top}$ Let $\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Decomposing **F** into **R** and **T**

If we have calibrated cameras we have $\mathbf{K}$ and $\mathbf{K}'$

Essential matrix: $\mathbf{E} = \mathbf{K}'^{\top}\mathbf{F}\mathbf{K}$

SVD: $\mathbf{E} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top}$ Let $\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

We get FOUR solutions:

$$\mathbf{E} = [\mathbf{R}|\mathbf{T}]$$

$\mathbf{R}_1 = \mathbf{U}\mathbf{W}\mathbf{V}^{\top}$ $\mathbf{R}_2 = \mathbf{U}\mathbf{W}^{\top}\mathbf{V}^{\top}$ $\mathbf{T}_1 = U_3$ $\mathbf{T}_2 = -U_3$

two possible rotations

two possible translations

We get FOUR solutions:

$$\mathbf{R}_1 = \mathbf{U}\mathbf{W}\mathbf{V}^\top \qquad \mathbf{R}_1 = \mathbf{U}\mathbf{W}\mathbf{V}^\top$$

$$\mathbf{T}_1 = U_3 \qquad \mathbf{T}_2 = -U_3$$

$$\mathbf{R}_2 = \mathbf{U}\mathbf{W}^\top\mathbf{V}^\top \qquad \mathbf{R}_2 = \mathbf{U}\mathbf{W}^\top\mathbf{V}^\top$$

$$\mathbf{T}_2 = -U_3 \qquad \mathbf{T}_1 = U_3$$

Which one do we choose?

Compute determinant of R, valid solution must be equal to 1

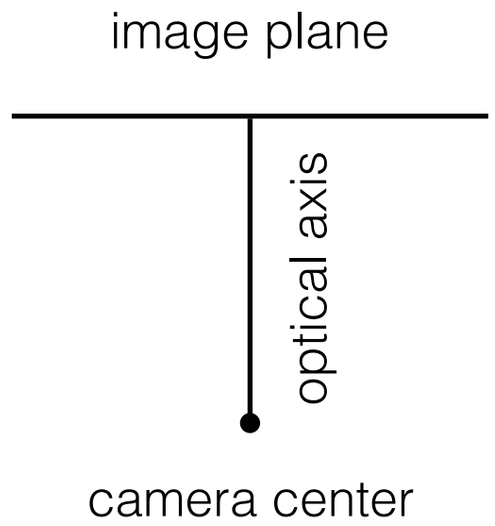
(note: det(R) = -1 means rotation and reflection)

Compute 3D point using triangulation, valid solution has positive Z value

(Note: negative Z means point is behind the camera)

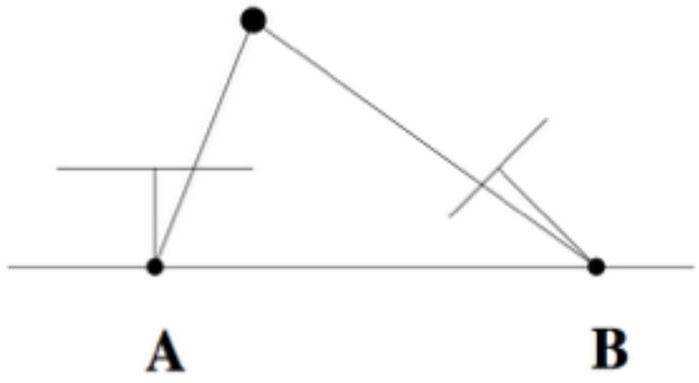
Let’s visualize the four configurations…

Camera Icon

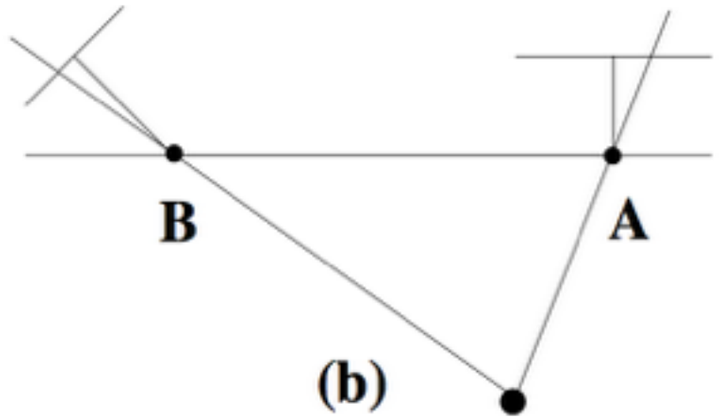


Find the configuration where the points is in front of both cameras

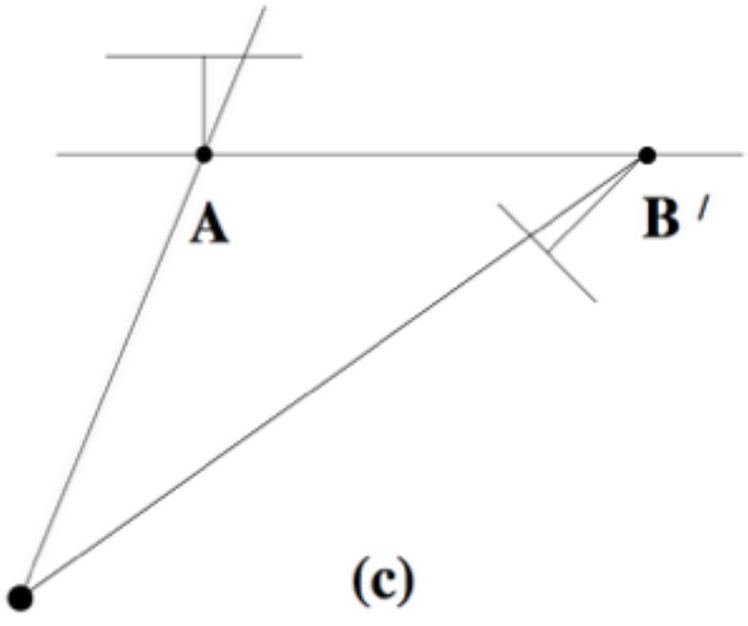
Find the configuration where the points is in front of both cameras



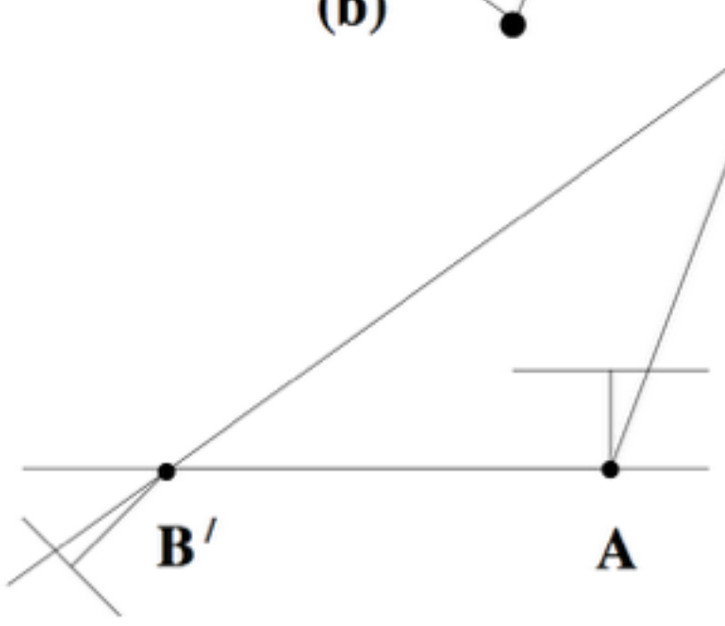
(a)



(b)

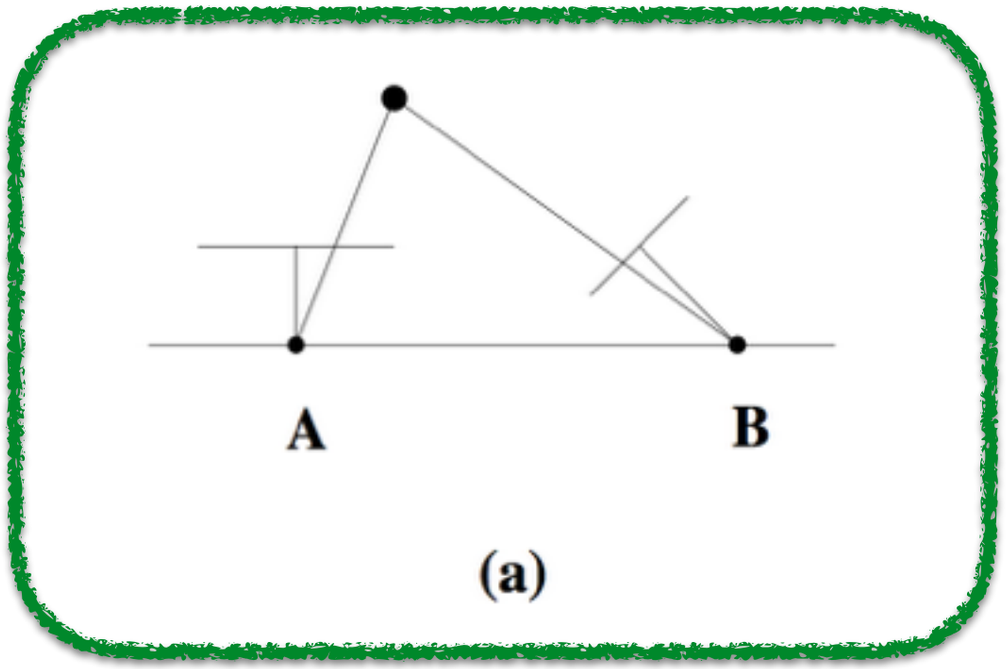


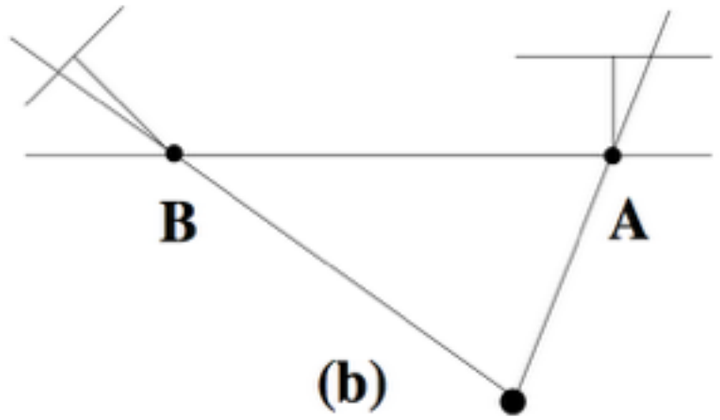
(c)

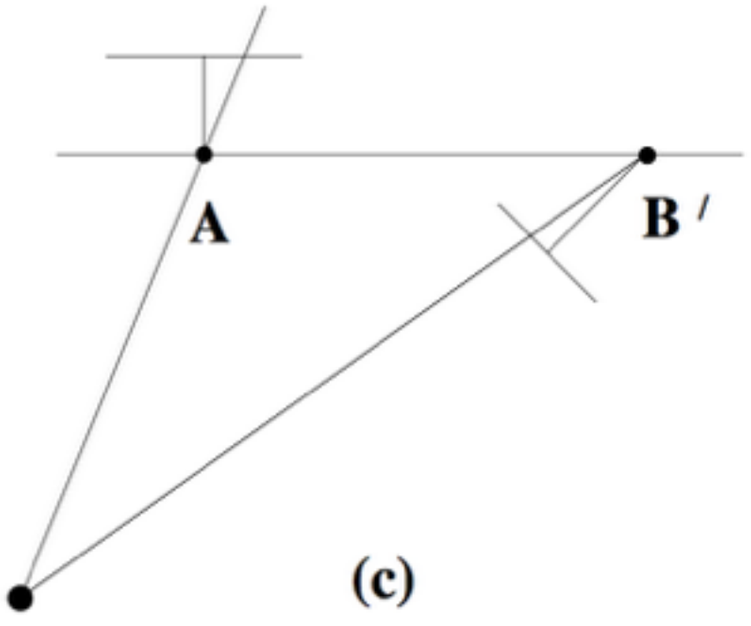


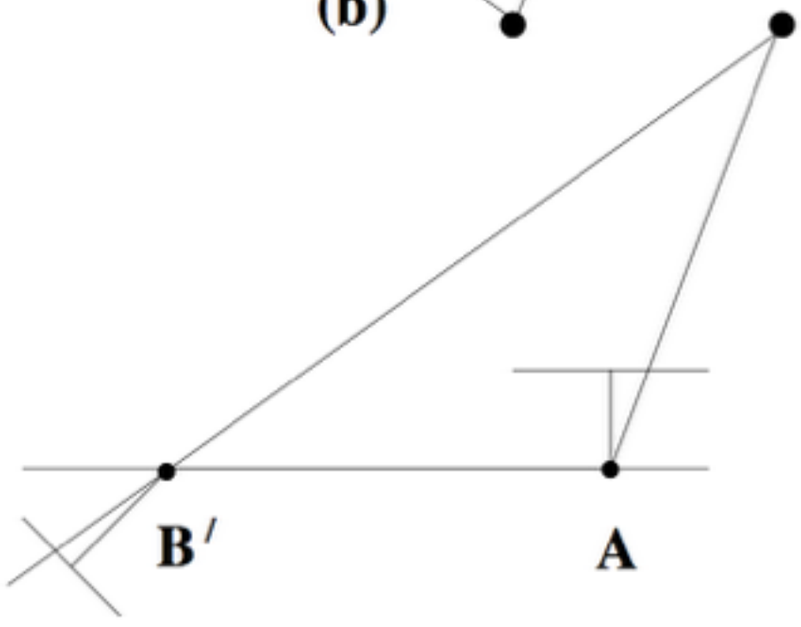
(d)

Find the configuration where the points is in front of both cameras









From points correspondences to camera displacement

1. Normalize the image points **x,x′** using **K,K′**

2. Use the 8-point algorithm to find an approximation of **E** (SVD!)

3. Project **E** to essential space (SVD!!)
   (set smallest SV to zero)

4. Recover possible solutions for **R** and **T** (SVD!!!)

5. Use point correspondence to find the correct **R,T** pair (don’t use SVD…)

Procedure for Reconstruction

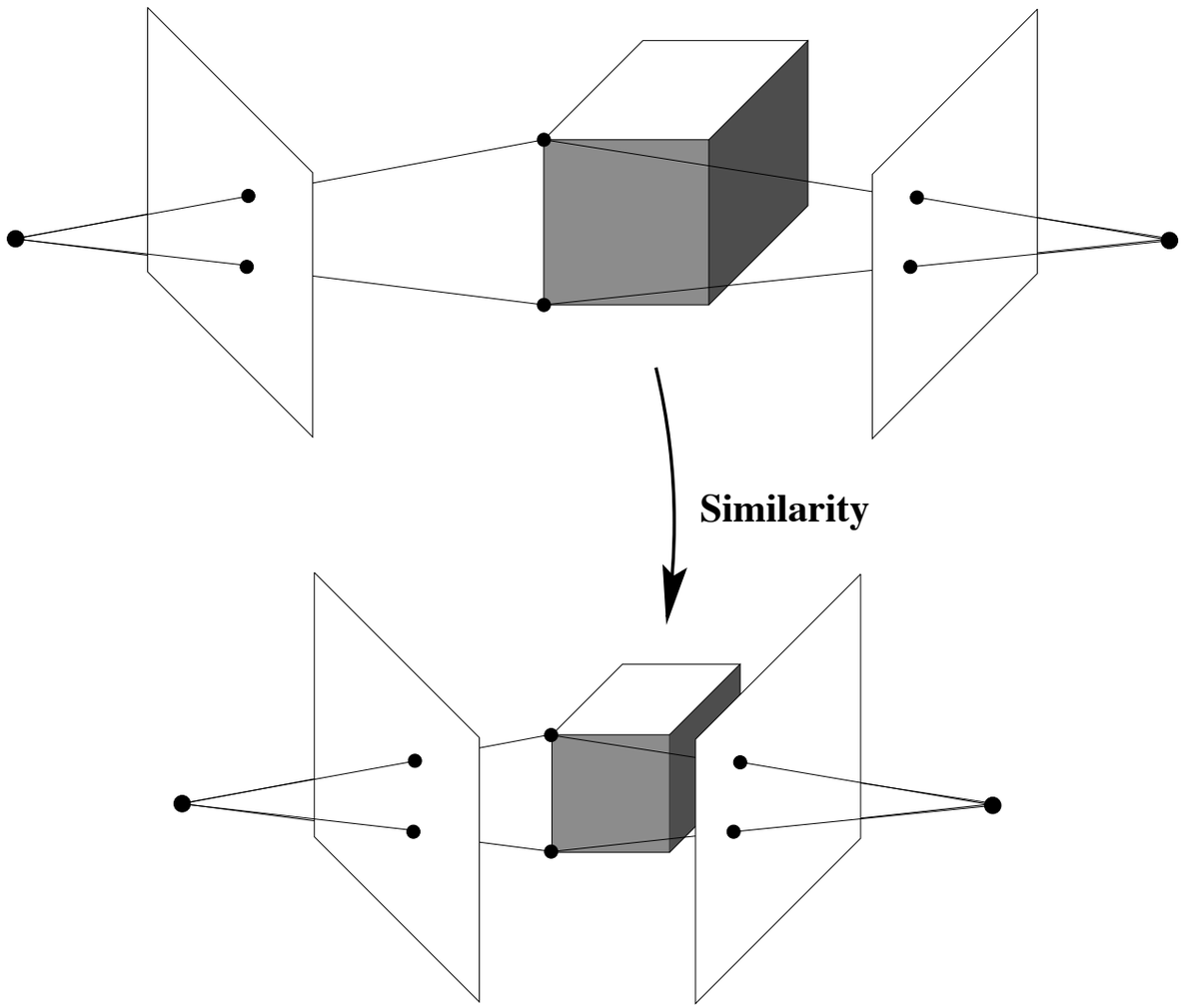
1. Compute the Fundamental Matrix **F** from points correspondences
   8-point algorithm

2. Compute the camera matrices **P** from the Fundamental matrix
   $\mathbf{P} = [\, \mathbf{I} \mid \mathbf{0} \,]$ and $\mathbf{P'} = [\, [\mathbf{e'}_x]\mathbf{F} \mid \mathbf{e'} \,]$

3. For each point correspondence, compute the point **X** in 3D space (triangularization)
   DLT with $x = \mathbf{P}\,X$ and $x' = \mathbf{P'}\,X$

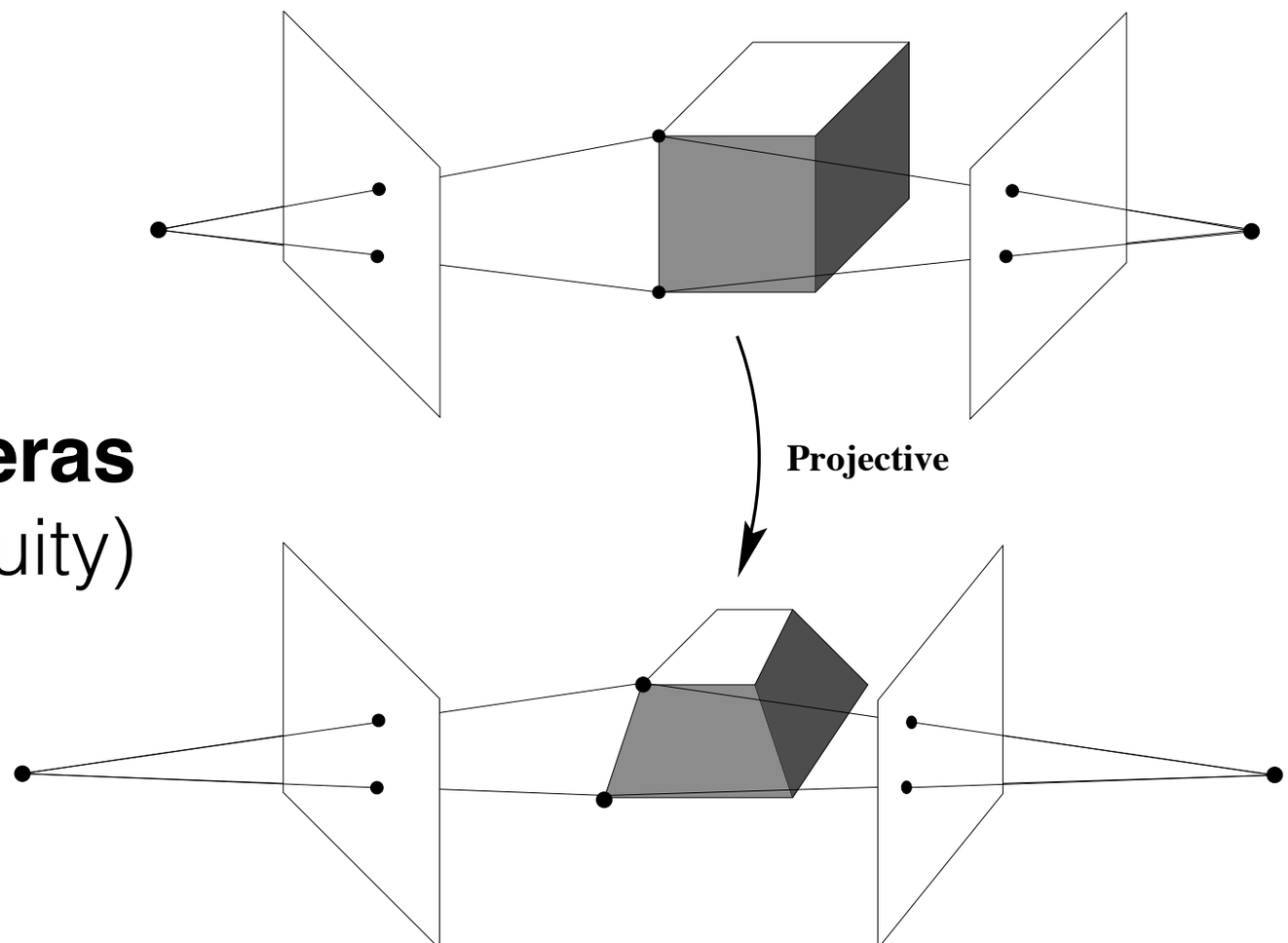
Projective Ambiguity

- Reconstruction is ambiguous by an arbitrary 3D projective transformation without prior knowledge of camera parameters

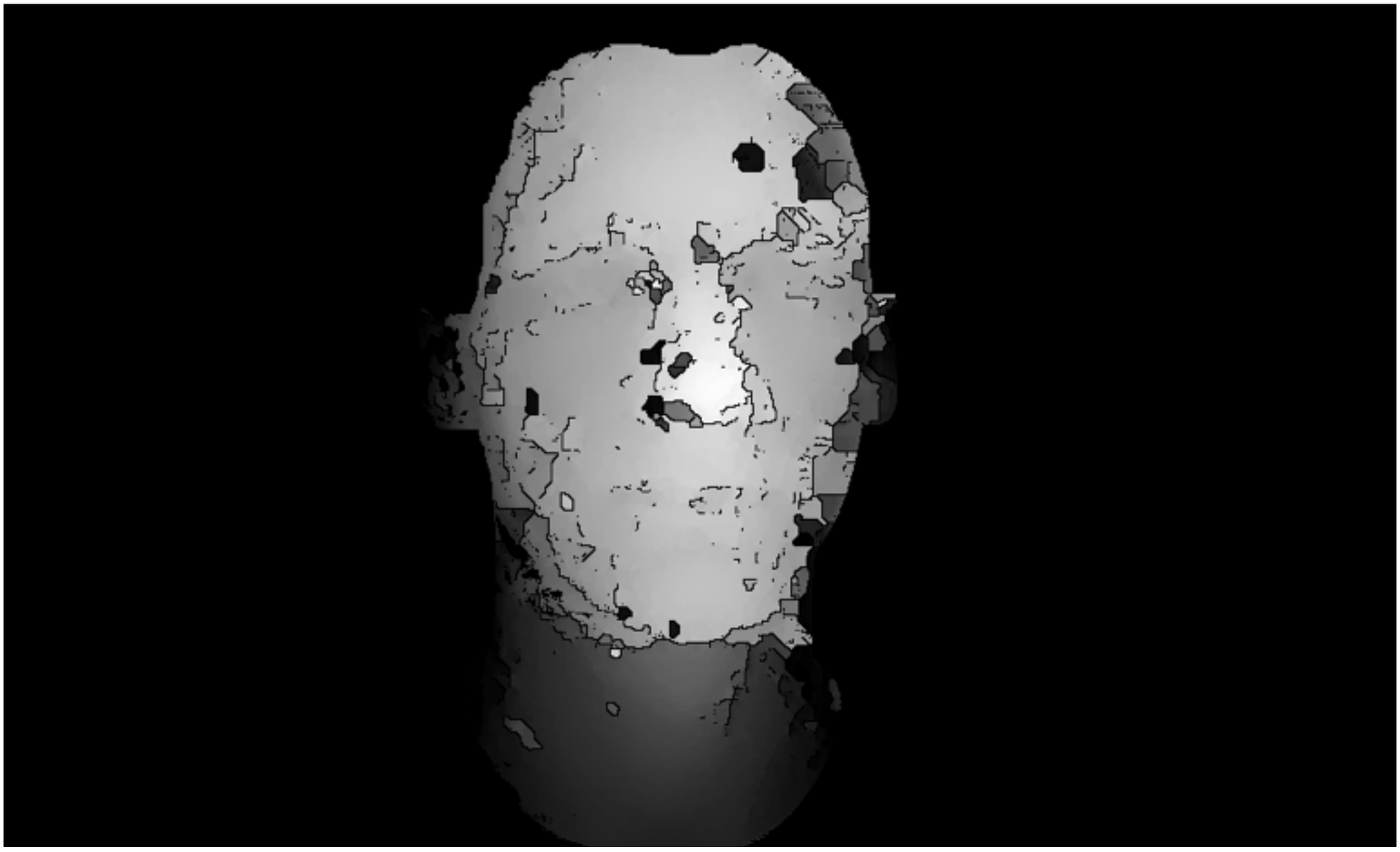


Calibrated cameras
(similarity projection ambiguity)

Uncalibrated cameras
(projective projection ambiguity)



	Structure (scene geometry)	Motion (camera geometry)	Measurements
Pose Estimation	known	estimate	3D to 2D correspondences
Triangulation	estimate	known	2D to 2D coorespondences
Reconstruction	estimate	estimate	2D to 2D coorespondences



Stereo Vision

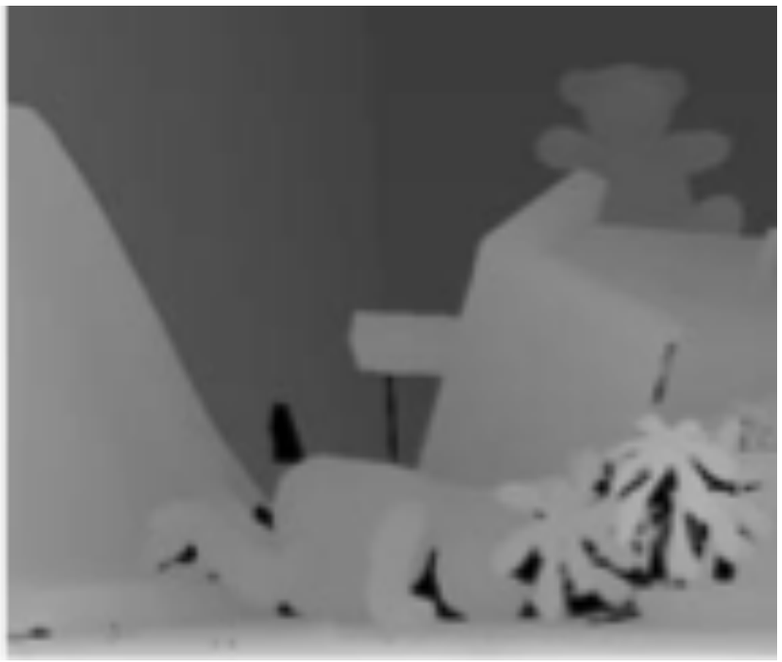
16-385 Computer Vision (Kris Kitani)

Carnegie Mellon University

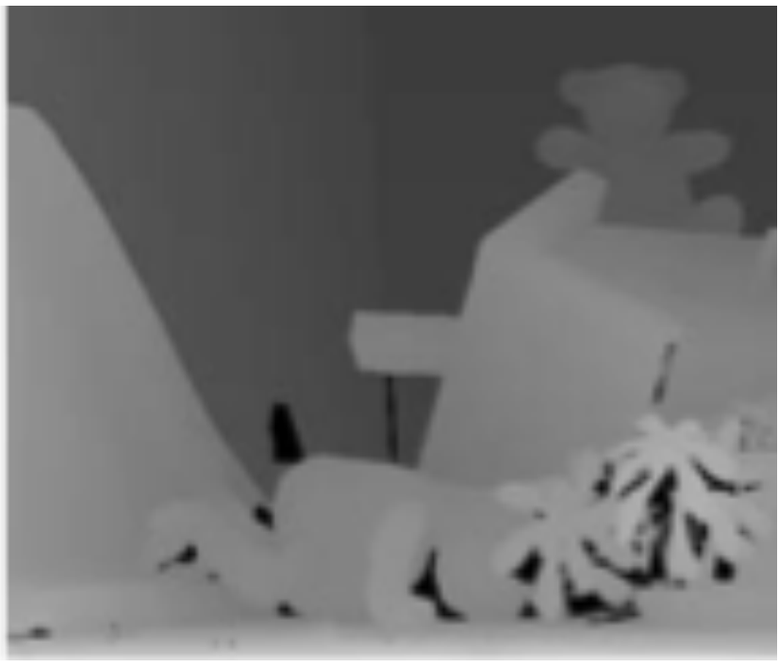
What's different between these two images?

Objects that are close move more or less?

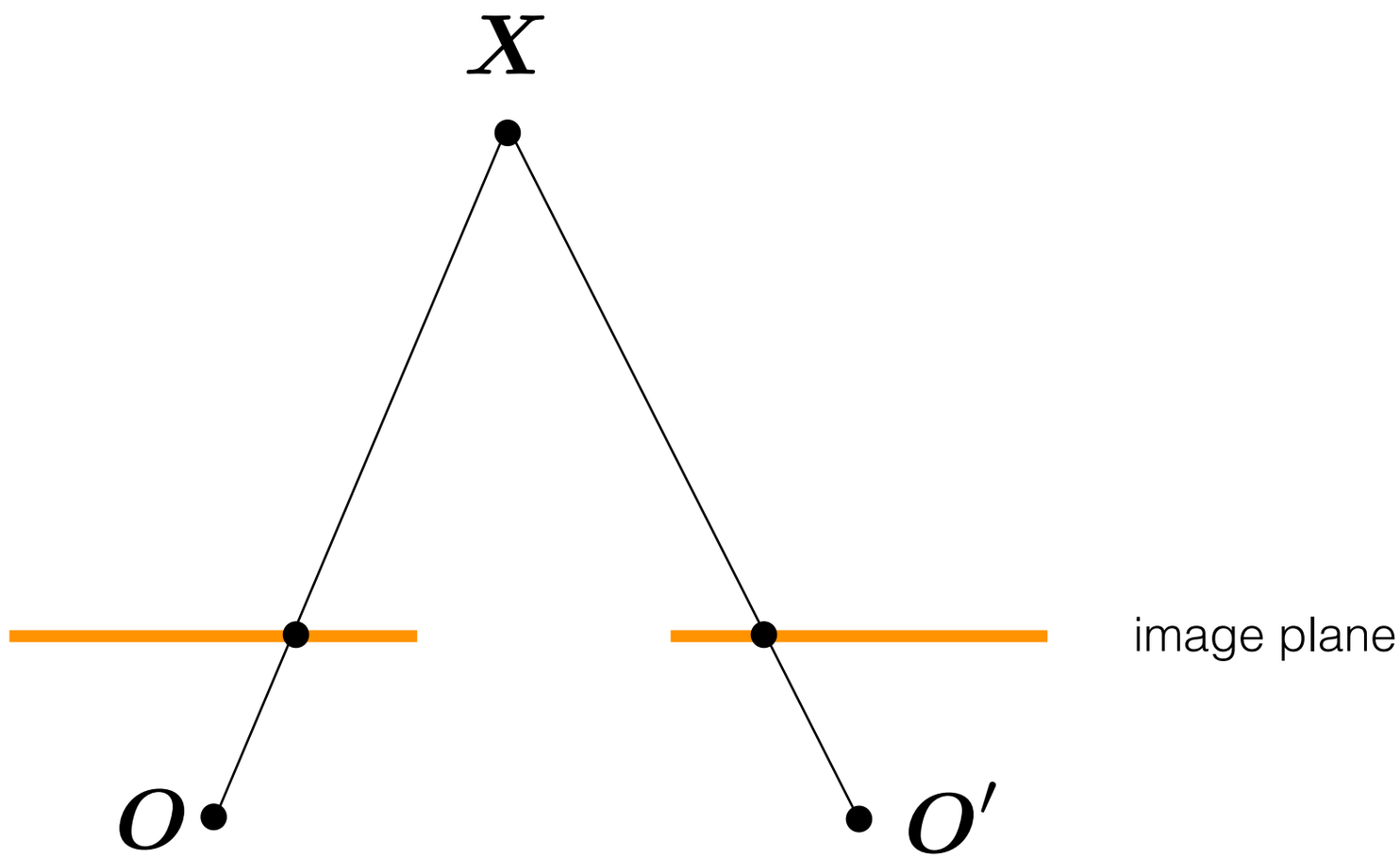
The amount of horizontal movement is inversely proportional to …

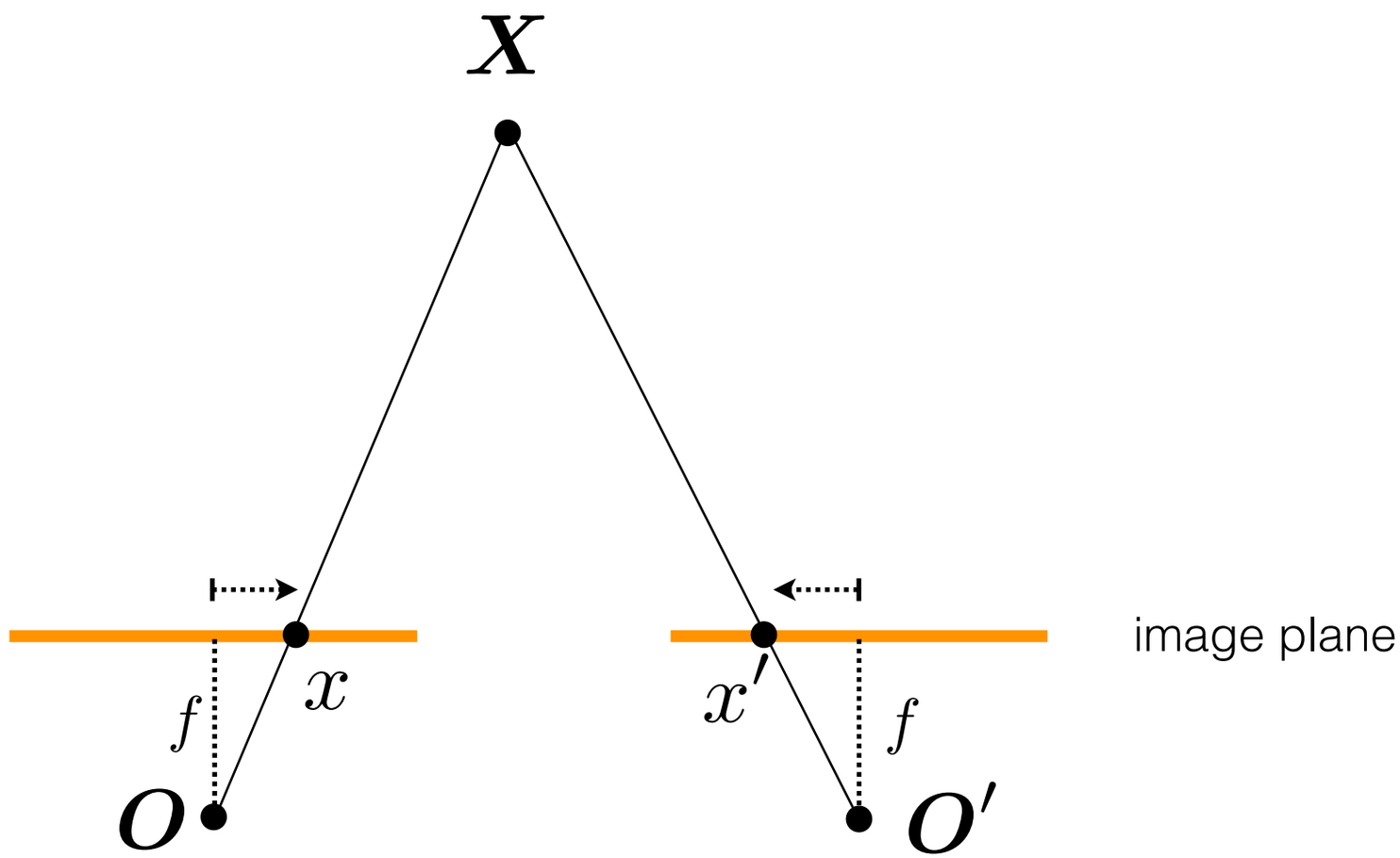


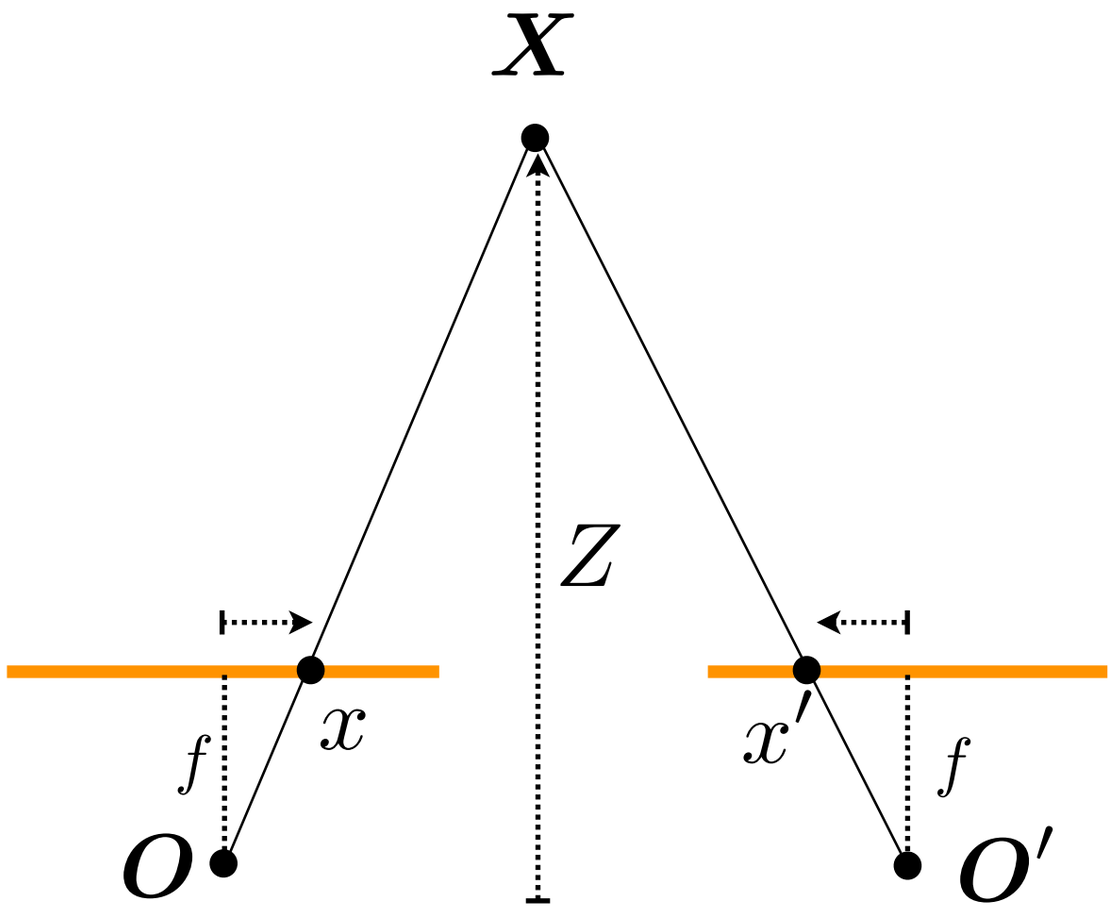
The amount of horizontal movement is inversely proportional to …

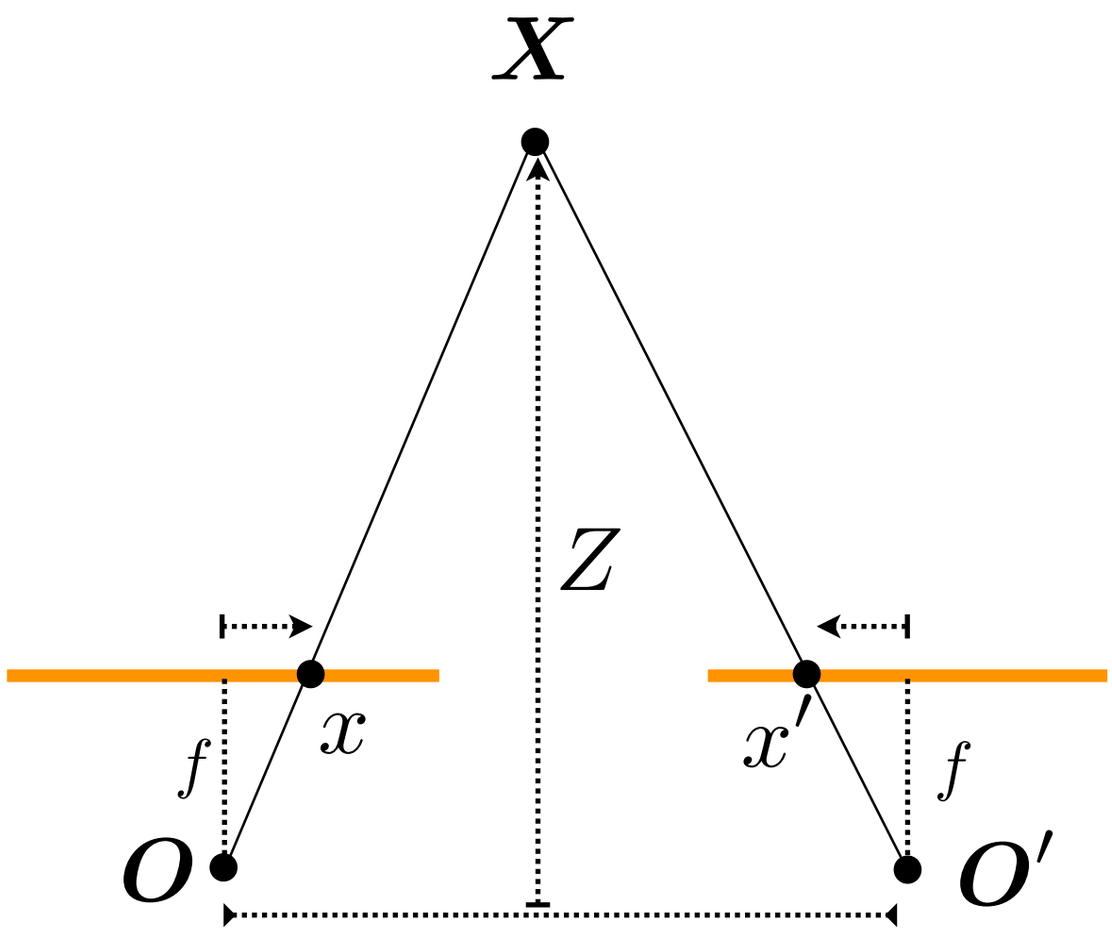


… the distance from the camera.

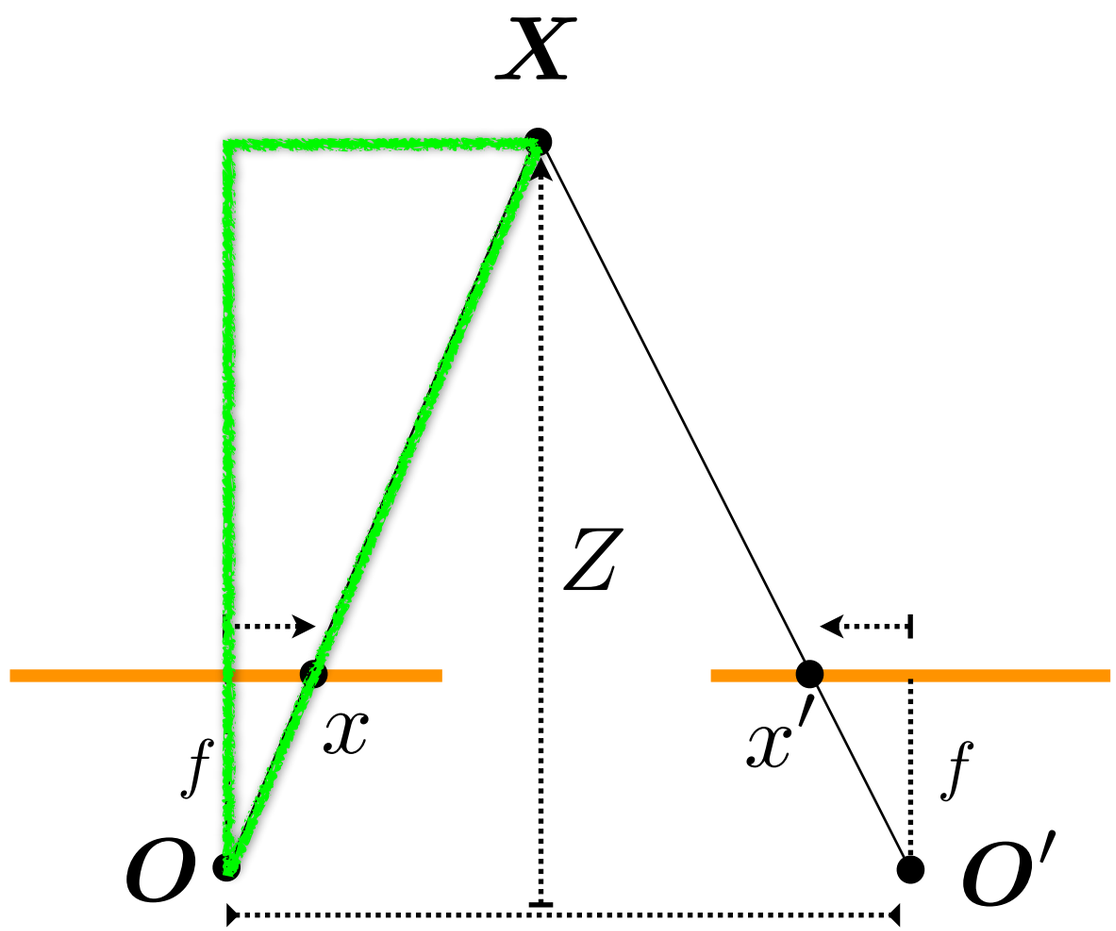

X
O
O′
image plane


X
x
x'
f
f
O
O'
image plane


X
Z
x
x′
f
f
O
O′

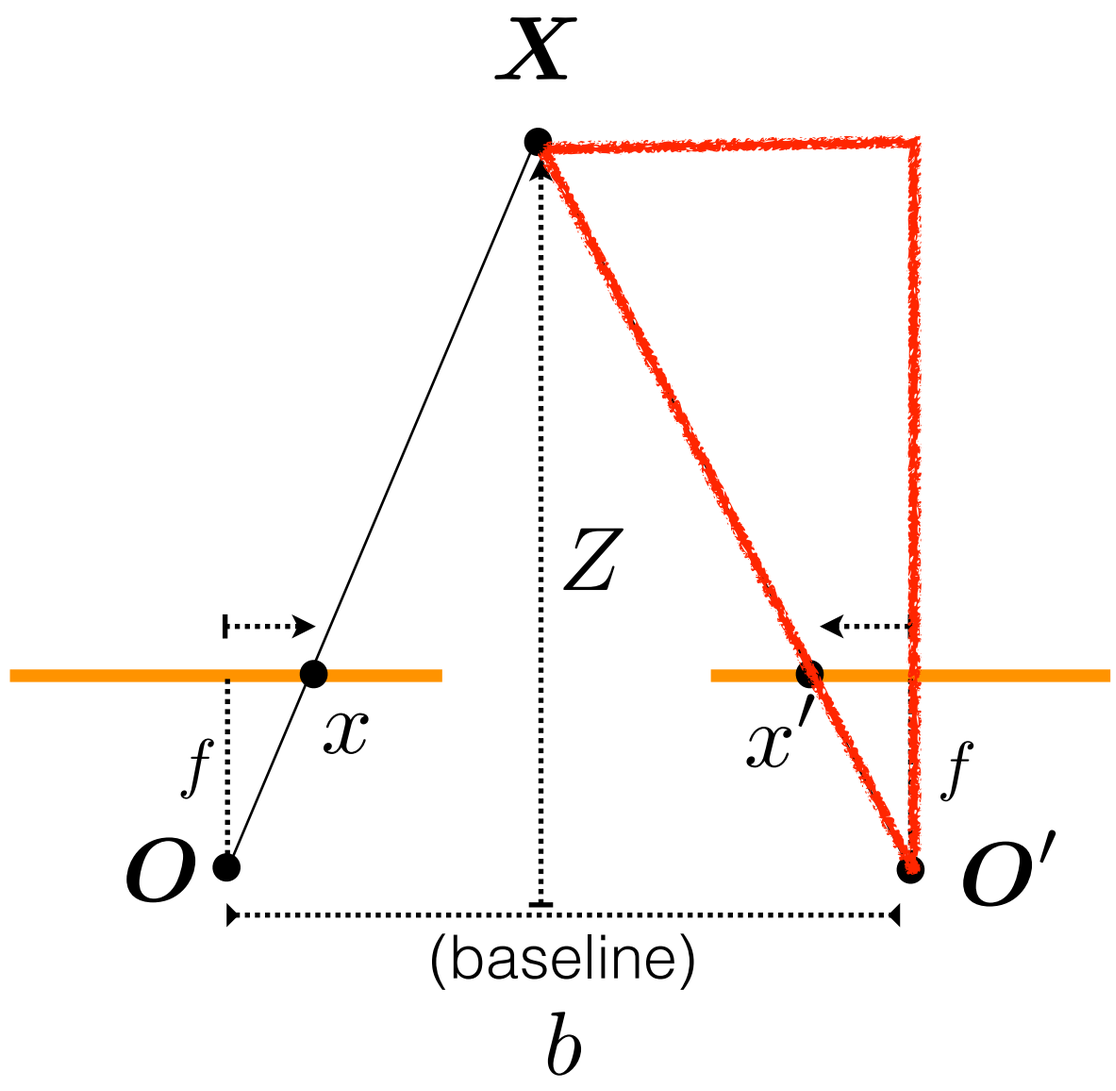

X
Z
x
x′
f
f
O
O′
(baseline)
b

$$\frac{X}{Z} = \frac{x}{f}$$



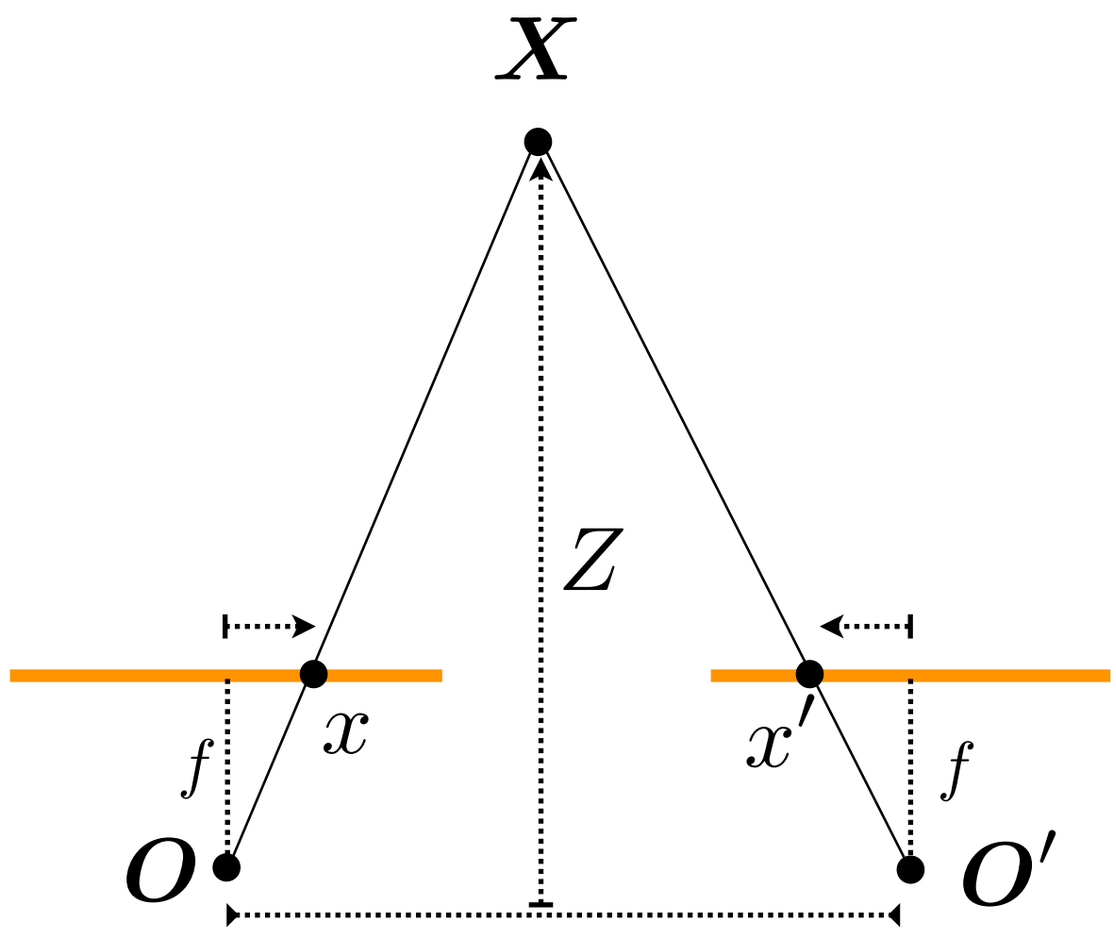
(baseline)
b

$$\frac{X}{Z} = \frac{x}{f}$$



$$\frac{b - X}{Z} = \frac{x'}{f}$$

$$\frac{X}{Z} = \frac{x}{f}$$



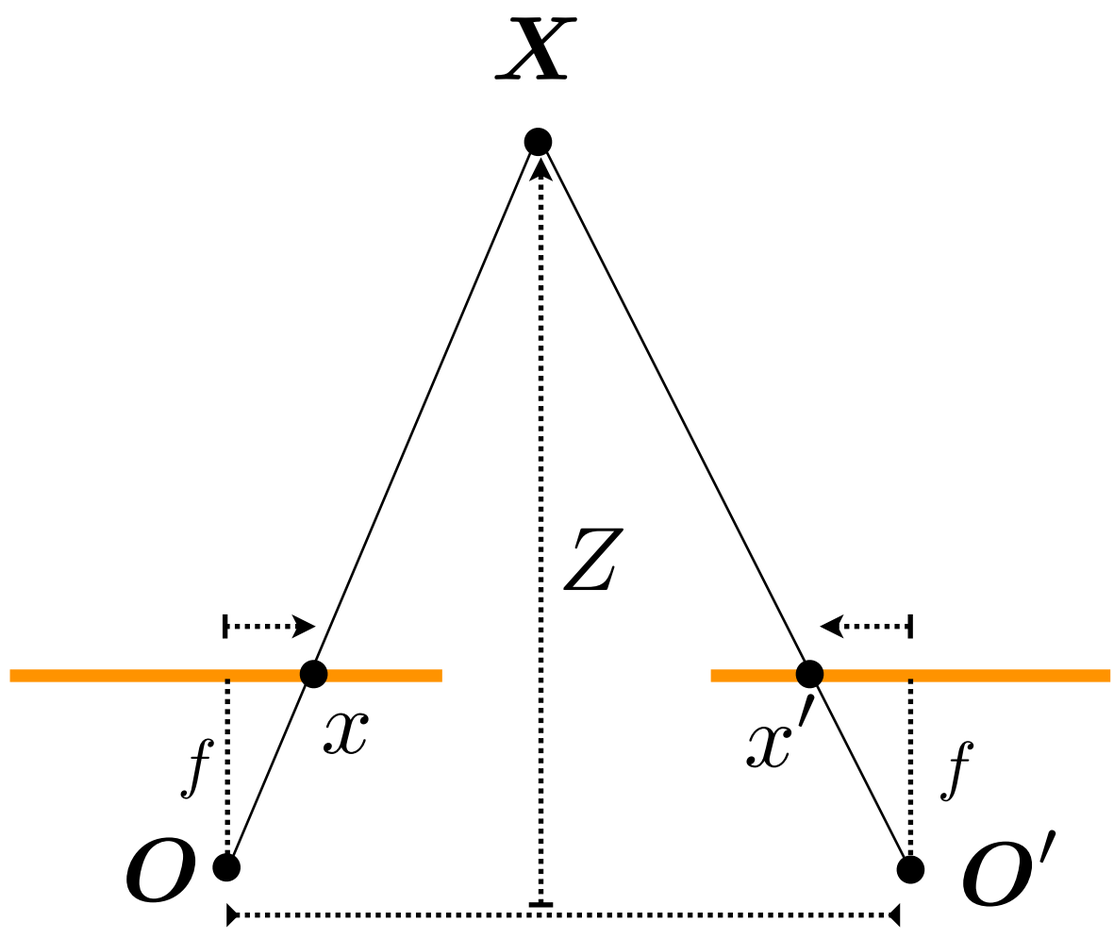
$$\frac{b - X}{Z} = \frac{x'}{f}$$

Disparity

$$d = x - x'$$

$$= \frac{bf}{Z}$$

$$\frac{X}{Z} = \frac{x}{f}$$



$$\frac{b - X}{Z} = \frac{x'}{f}$$

Disparity

$$d = x - x'$$

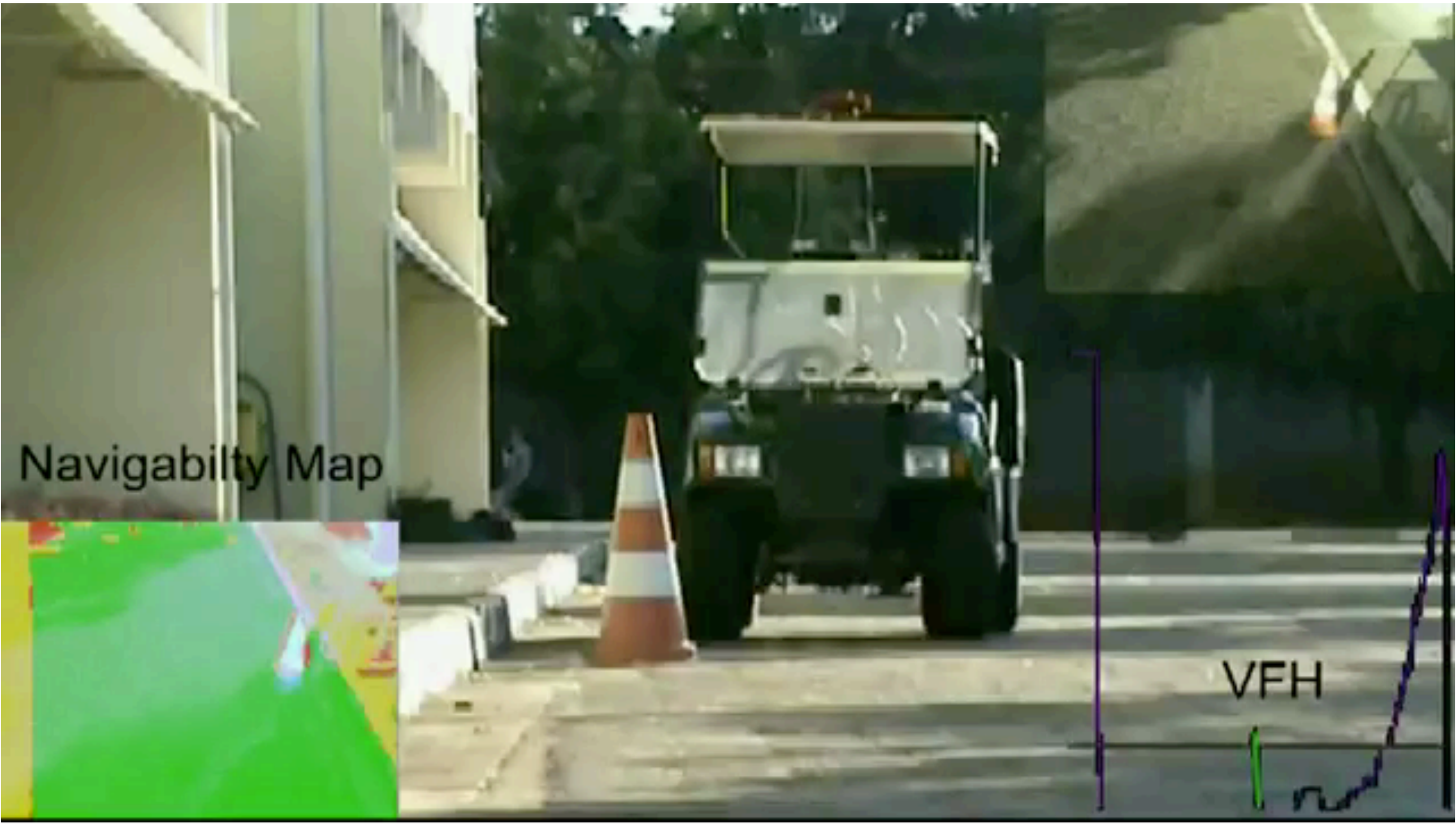
$$= \frac{bf}{Z}$$

inversely proportional to depth

Real-time stereo sensing

Nomad robot searches for meteorites in Antartica

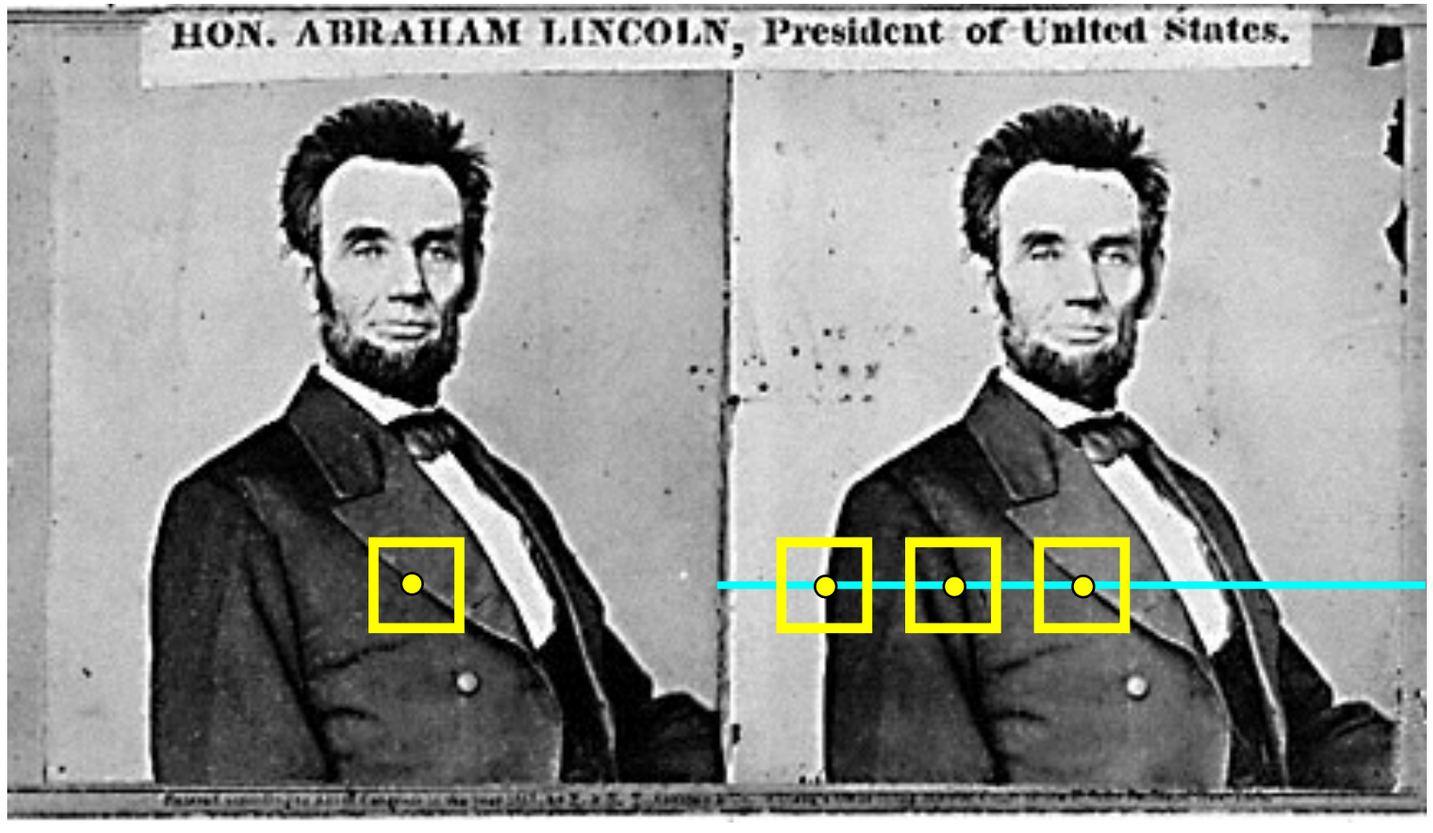
http://www.frc.ri.cmu.edu/projects/meteorobot/index.html


Navigabilty Map
VFH

Subaru
Eyesight system

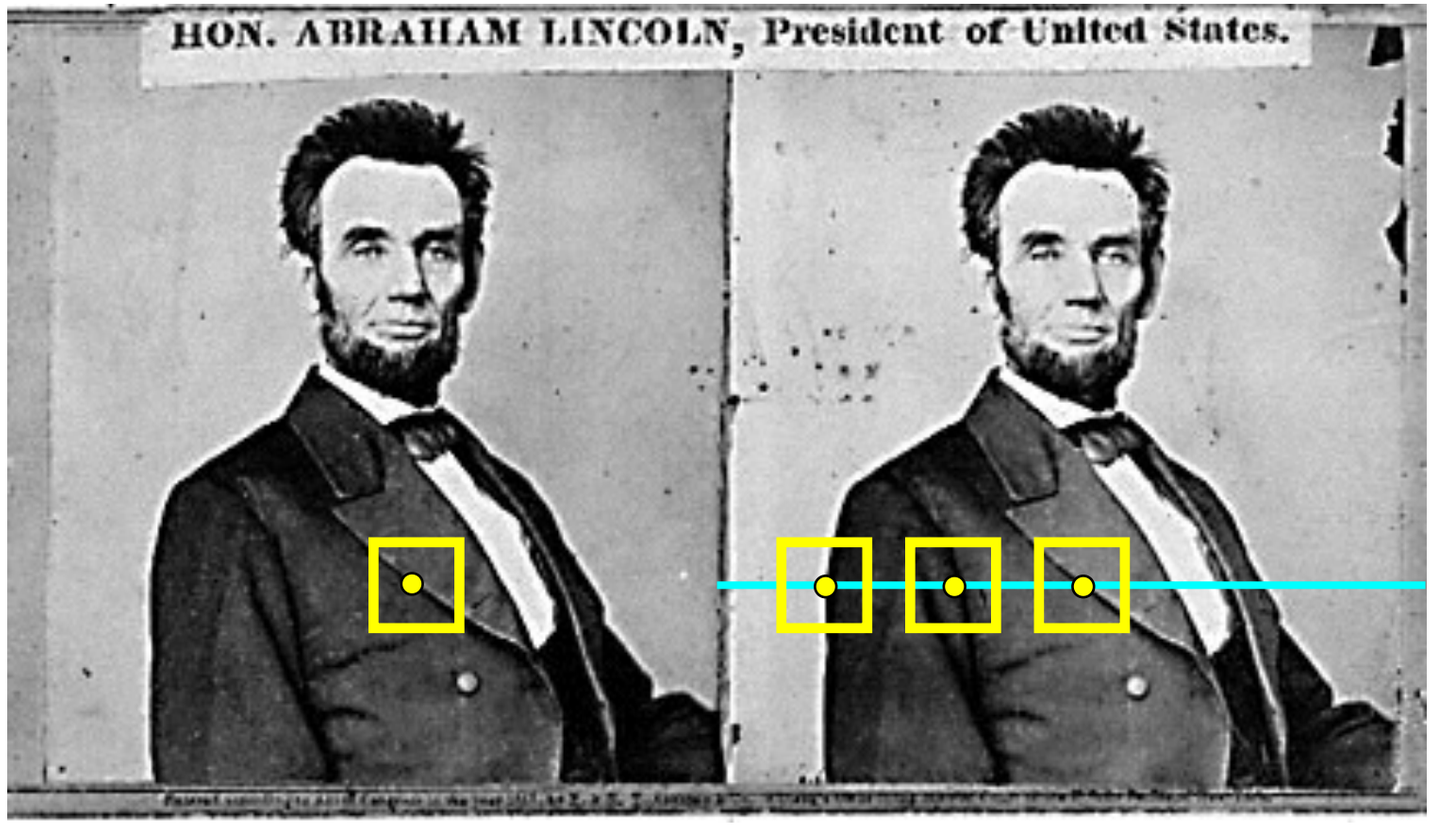
Pre-collision
braking

How so you compute depth from a stereo pair?

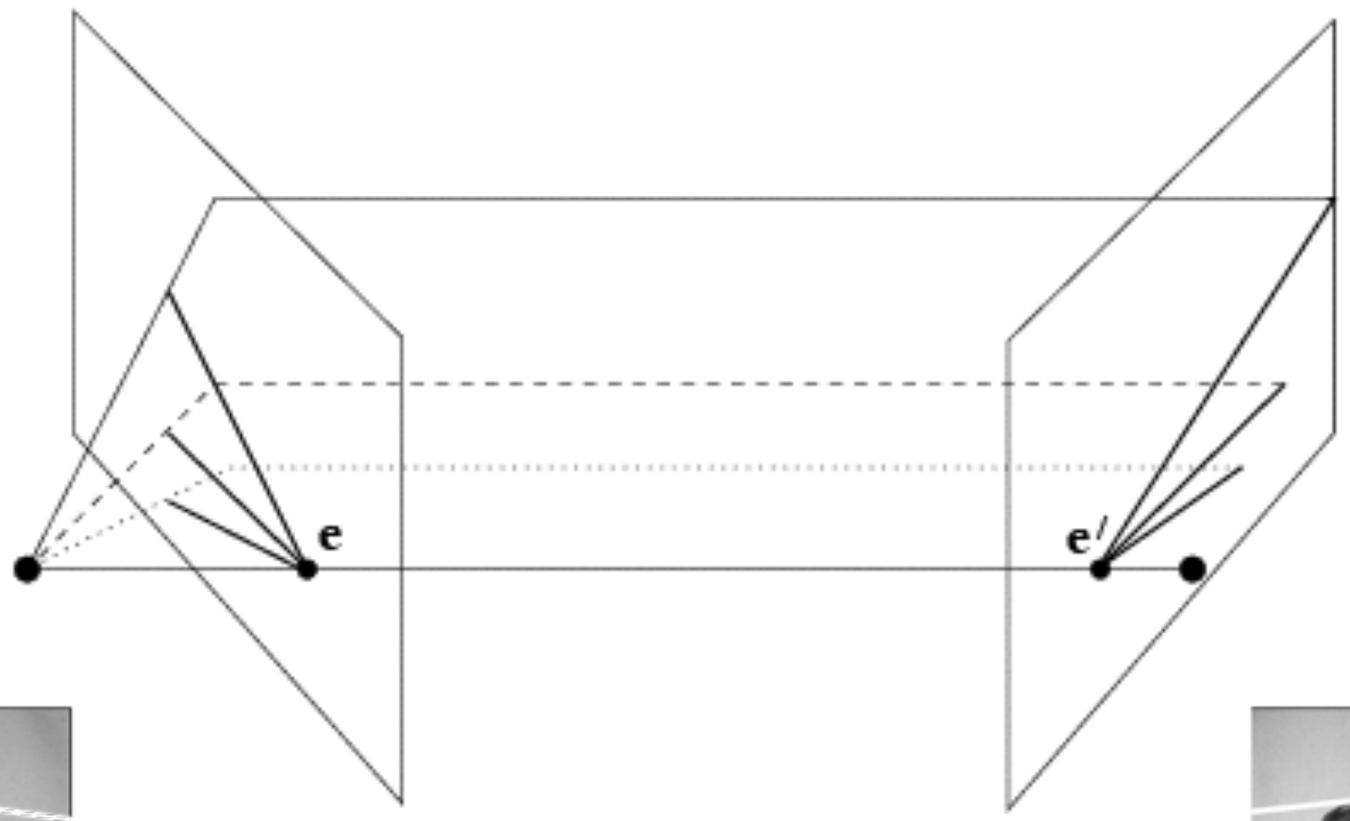


1. Rectify images
   (make epipolar lines horizontal)
2. For each pixel
   a. Find epipolar line
   b. Scan line for best match
   c. Compute depth from disparity

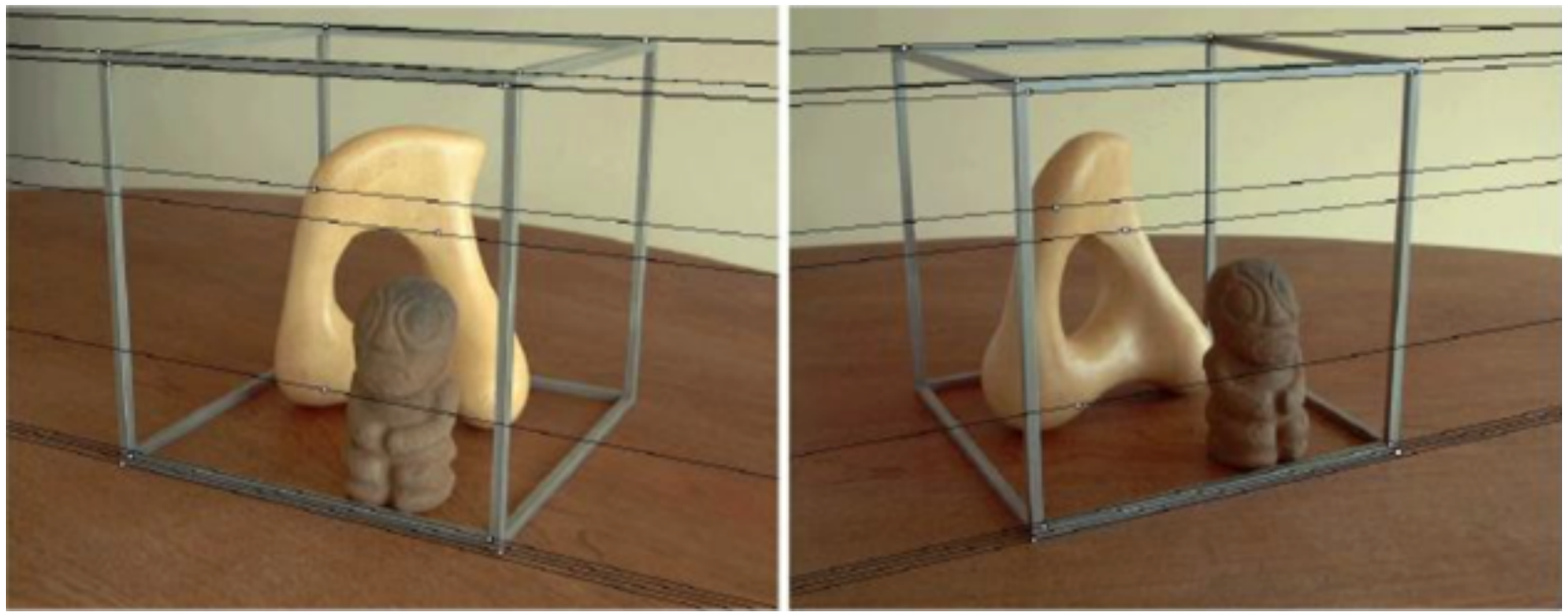
$$Z = \frac{bf}{d}$$



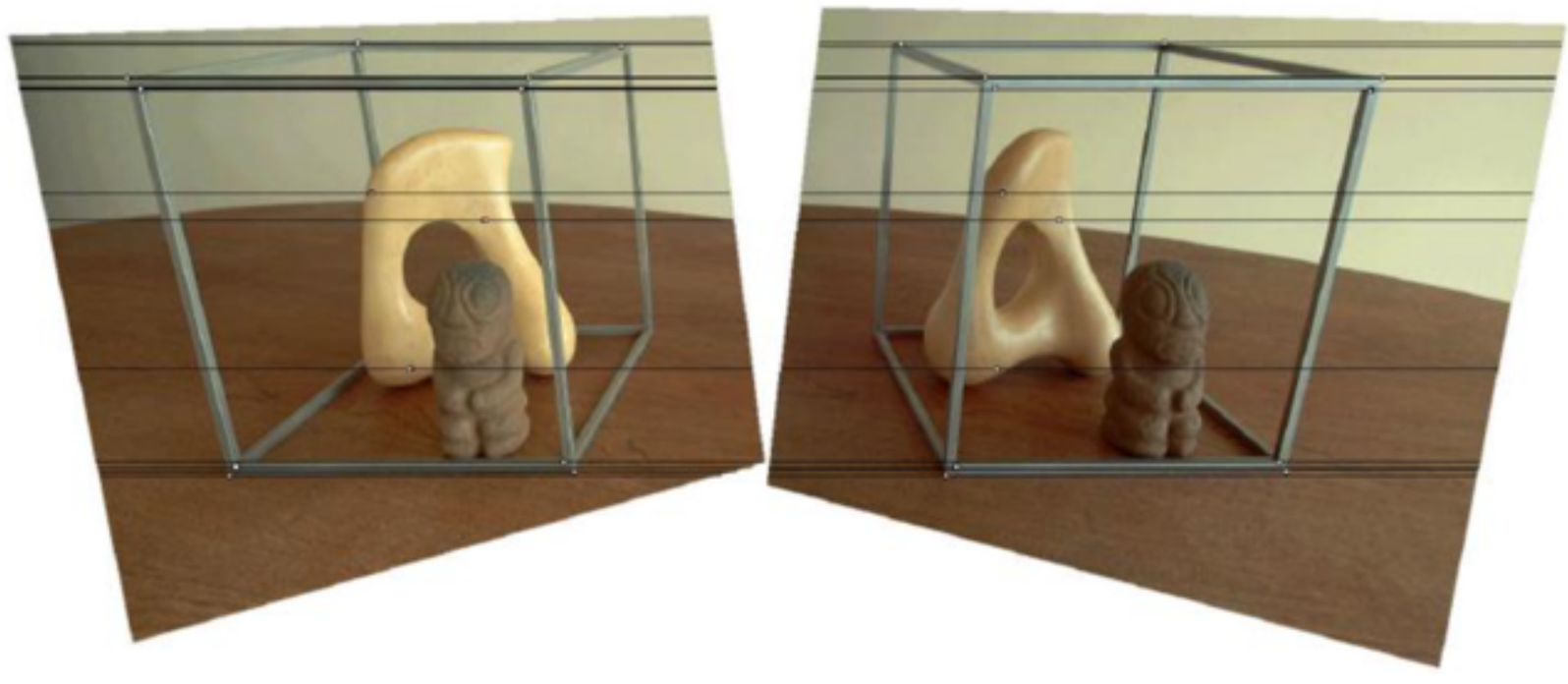
How can you make the epipolar lines horizontal?



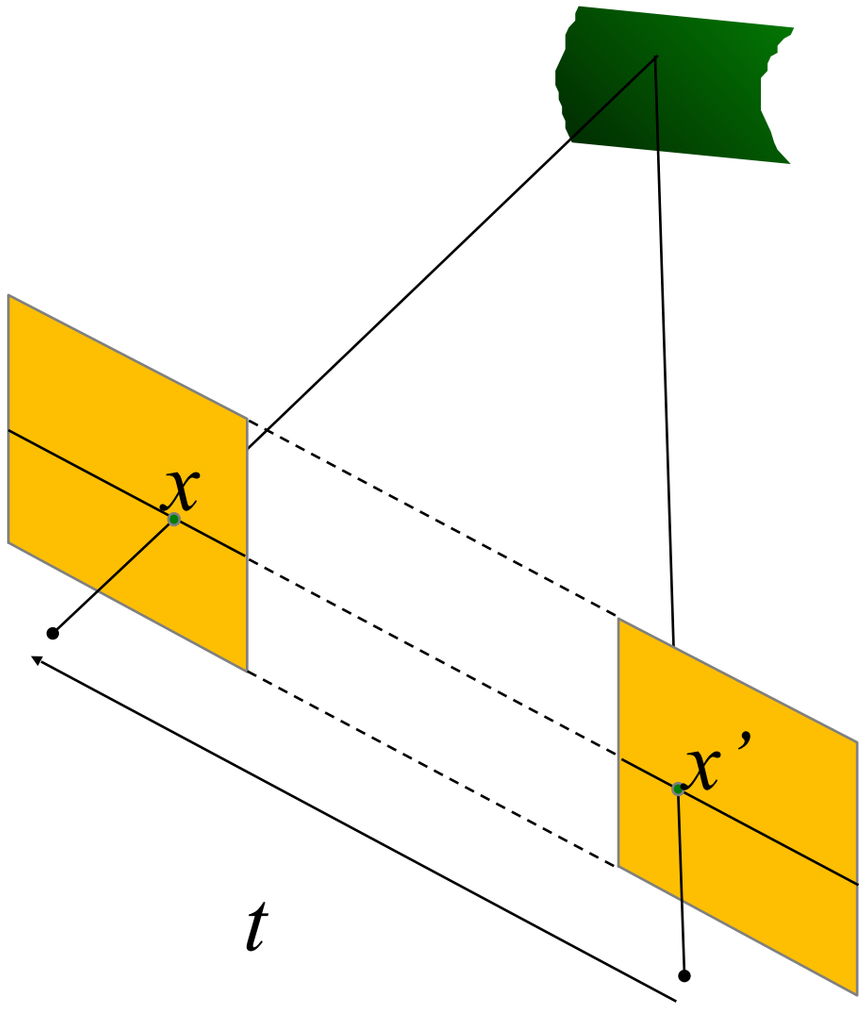
It’s hard to make the image planes exactly parallel



How can you make the epipolar lines horizontal?



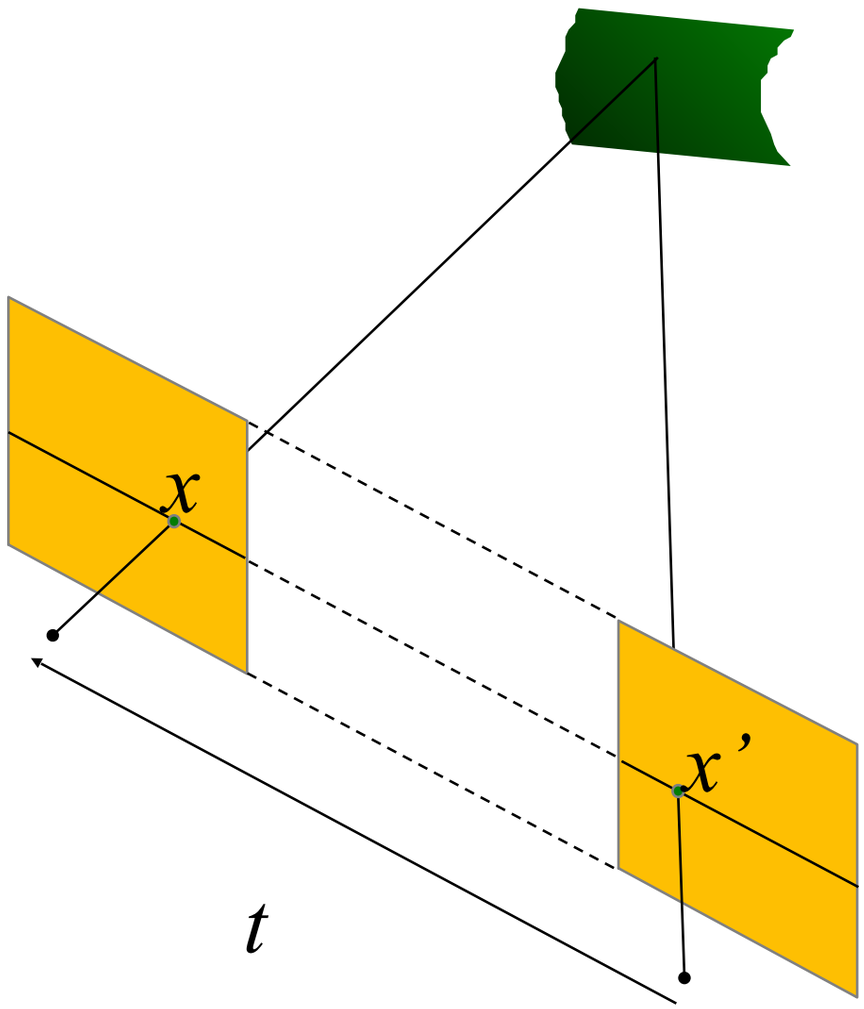
How can you make the epipolar lines horizontal?



When this relationship holds:

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

How can you make the epipolar lines horizontal?



When this relationship holds:

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

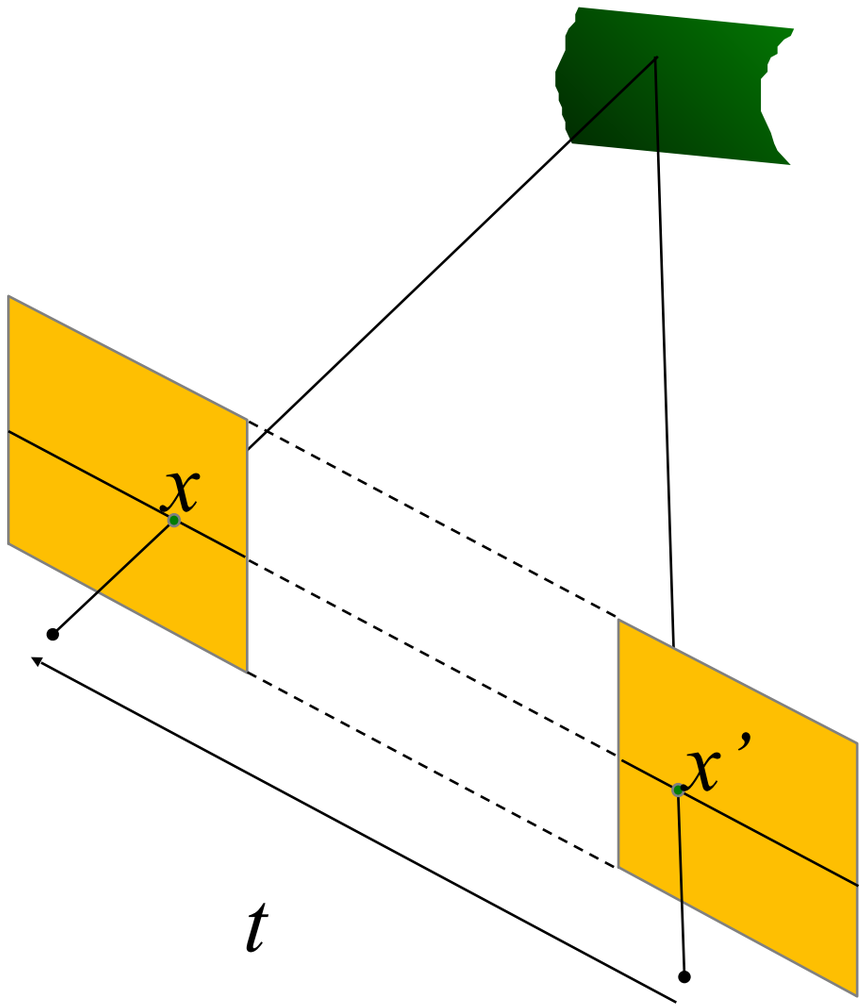
Let’s try this out…

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

This always has to hold

$$x^T E x' = 0$$

How can you make the epipolar lines horizontal?



When this relationship holds:

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

Let’s try this out…

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

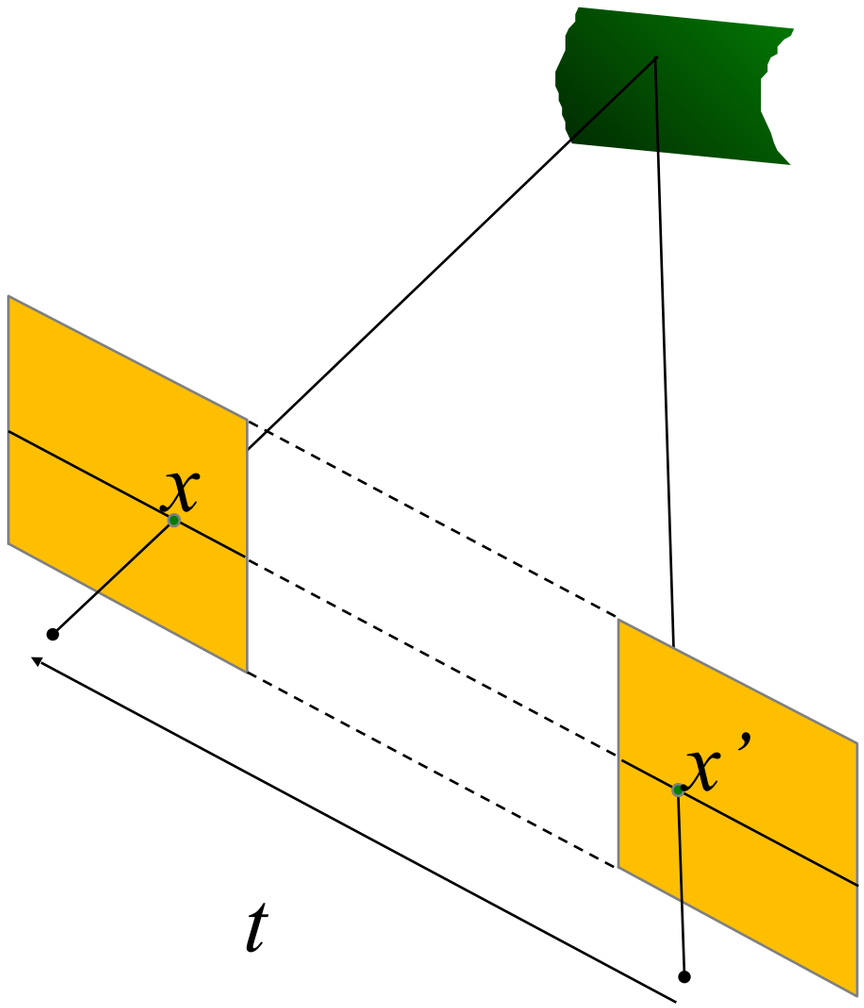
This always has to hold

$$x^T E x' = 0$$

Write out the constraint

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

How can you make the epipolar lines horizontal?



When this relationship holds:

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

Let’s try this out…

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

This always has to hold

$$x^T E x' = 0$$

Write out the constraint

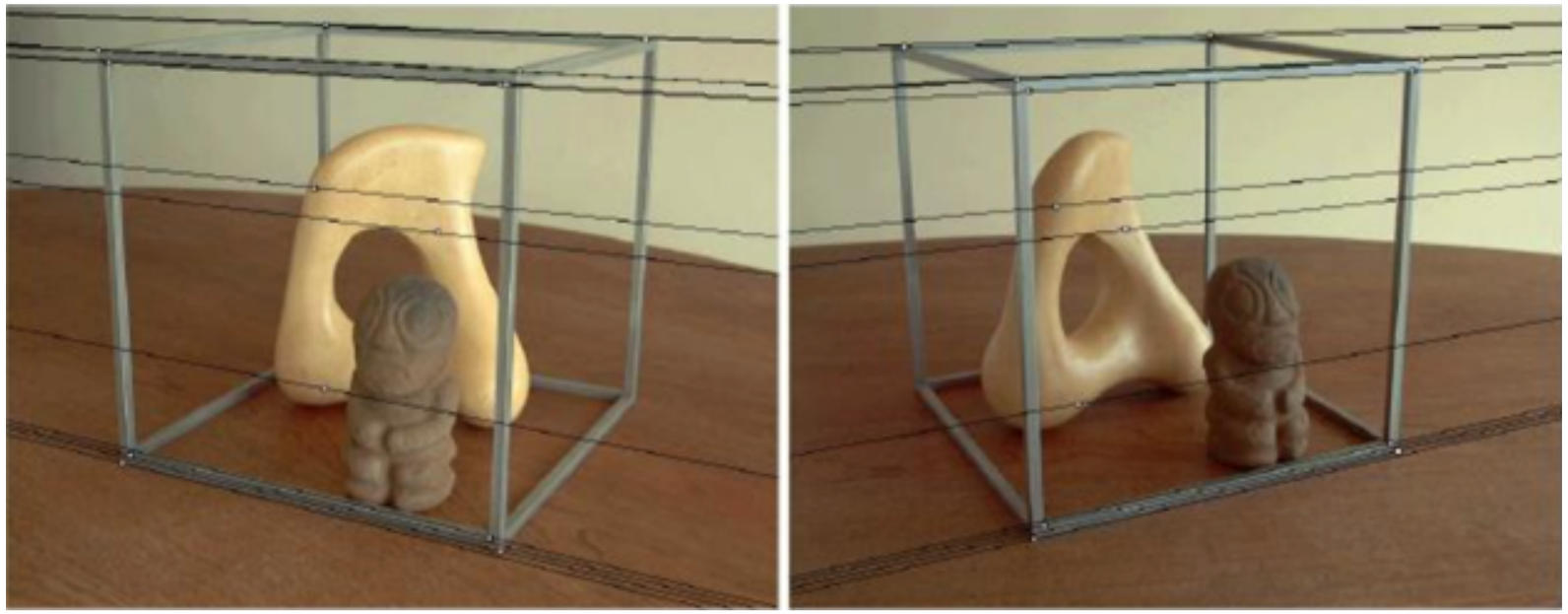
$$\begin{pmatrix} u & v & 1 \end{pmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0$$

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

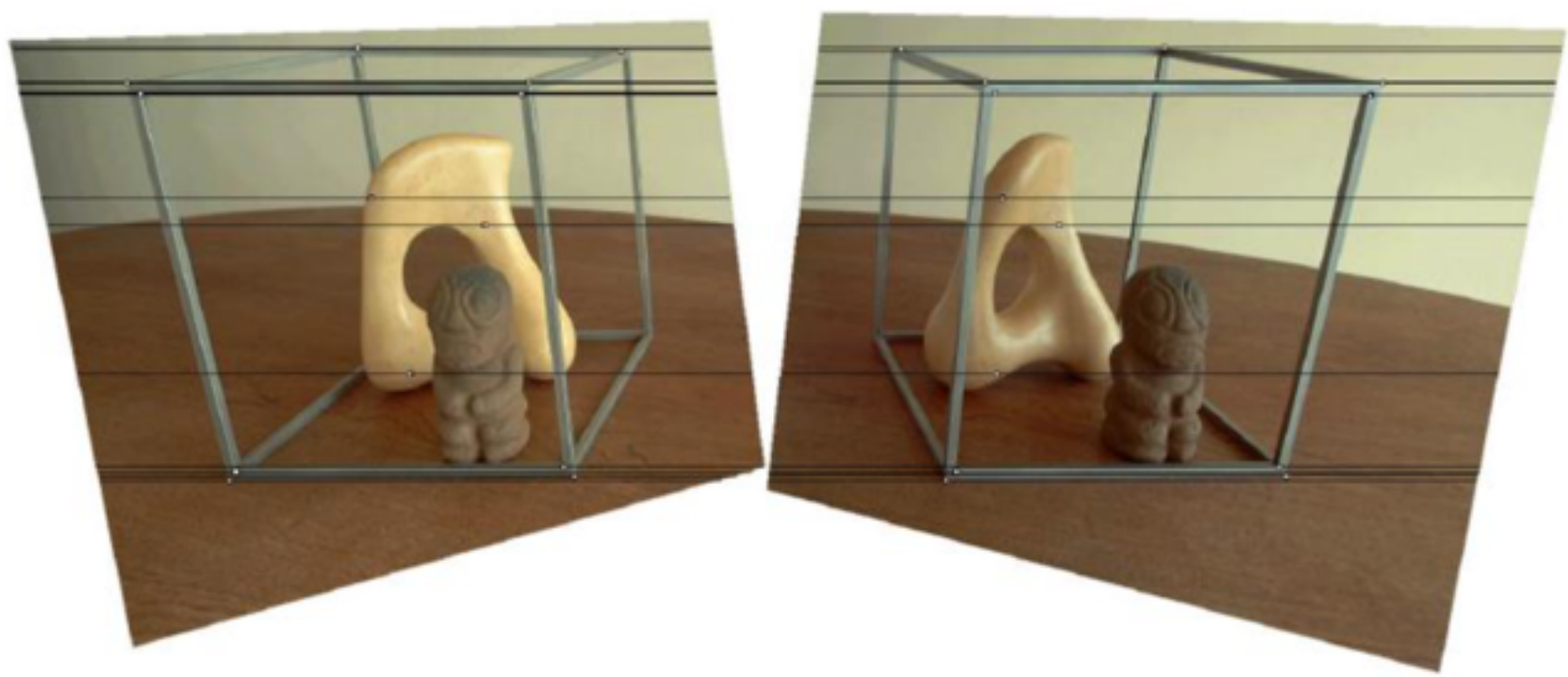
The image of a 3D point will always be on the same horizontal line

$$Tv = Tv'$$

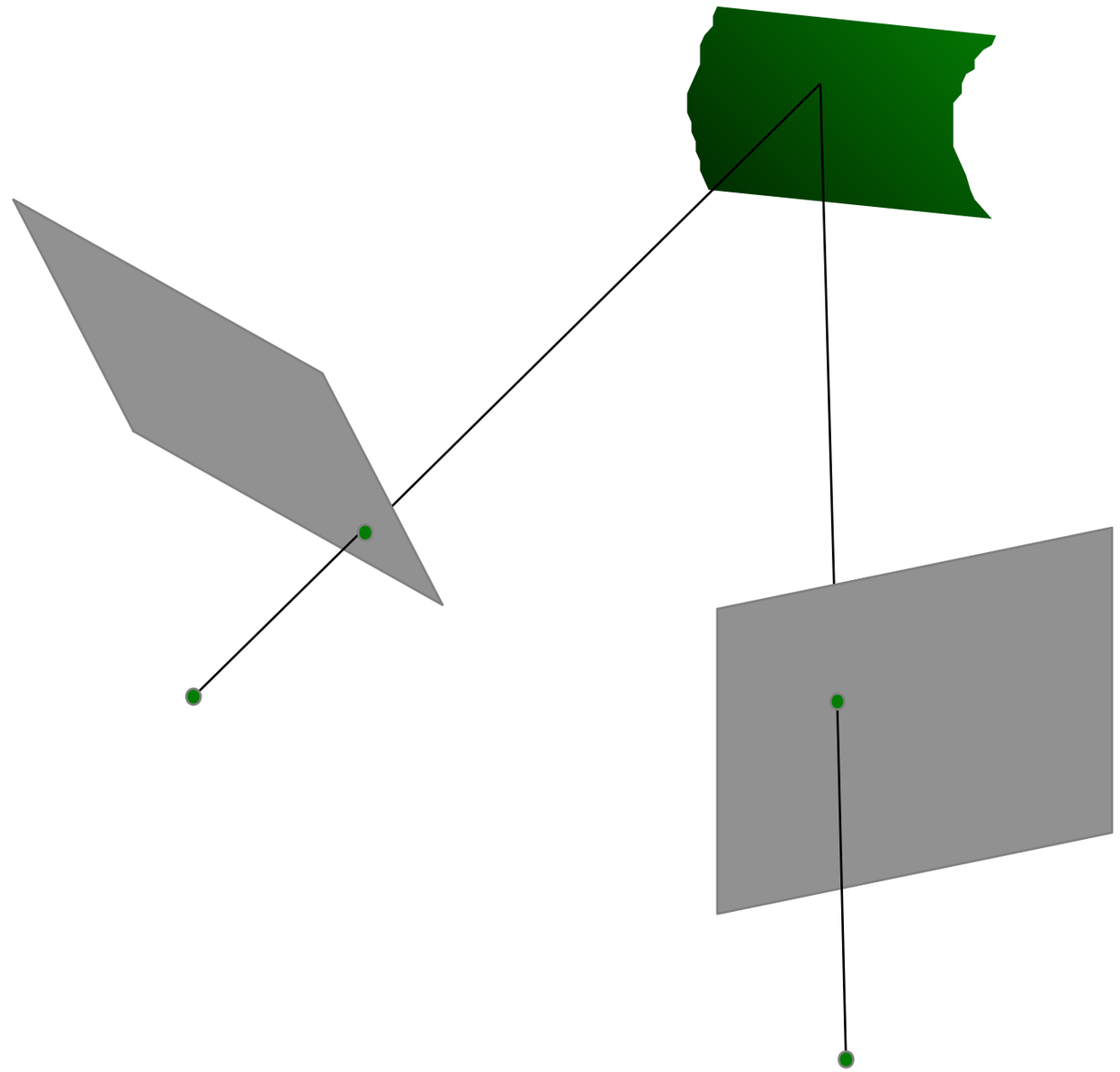
y coordinate is always the same!



What is stereo rectification?

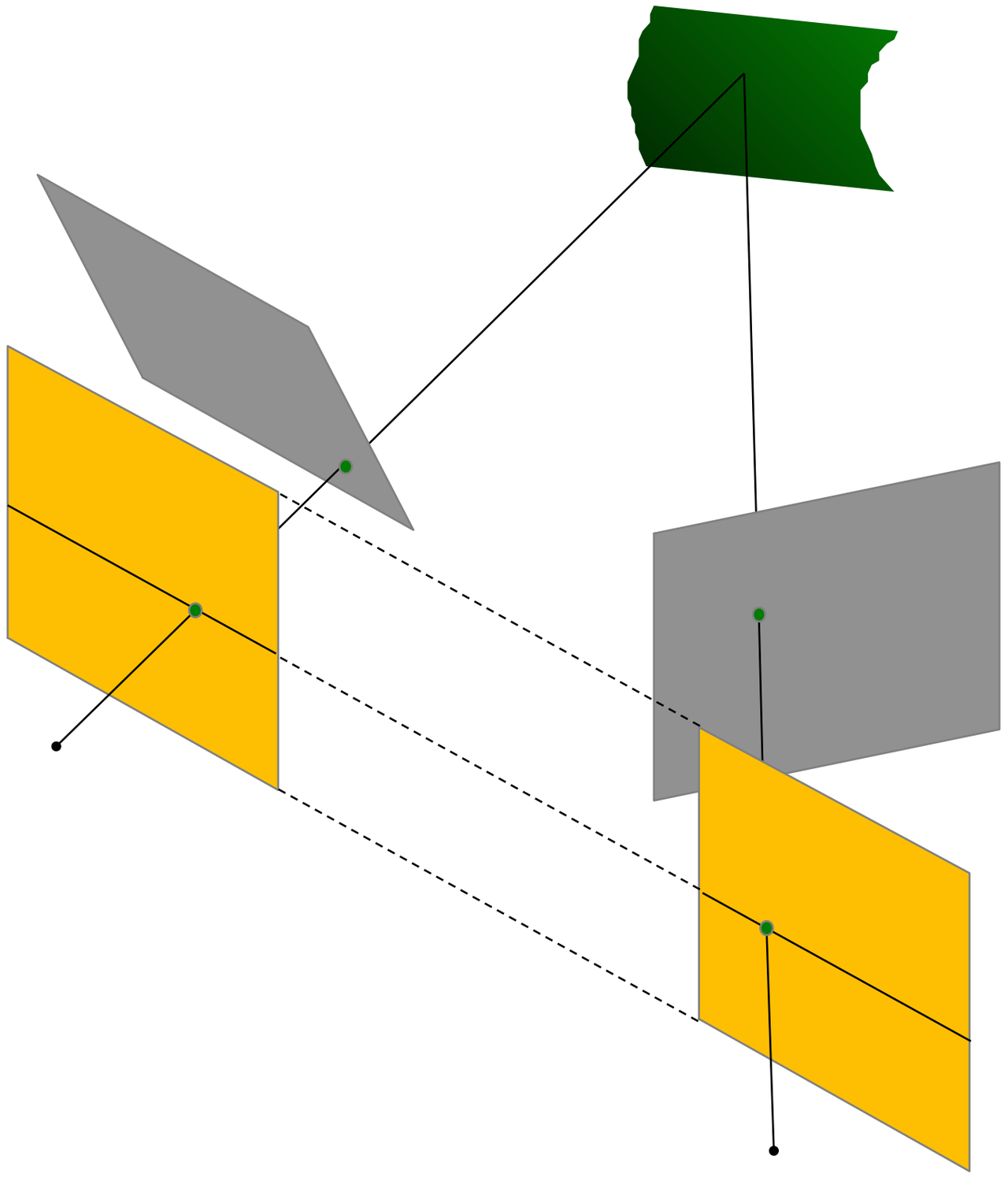


What is stereo rectification?



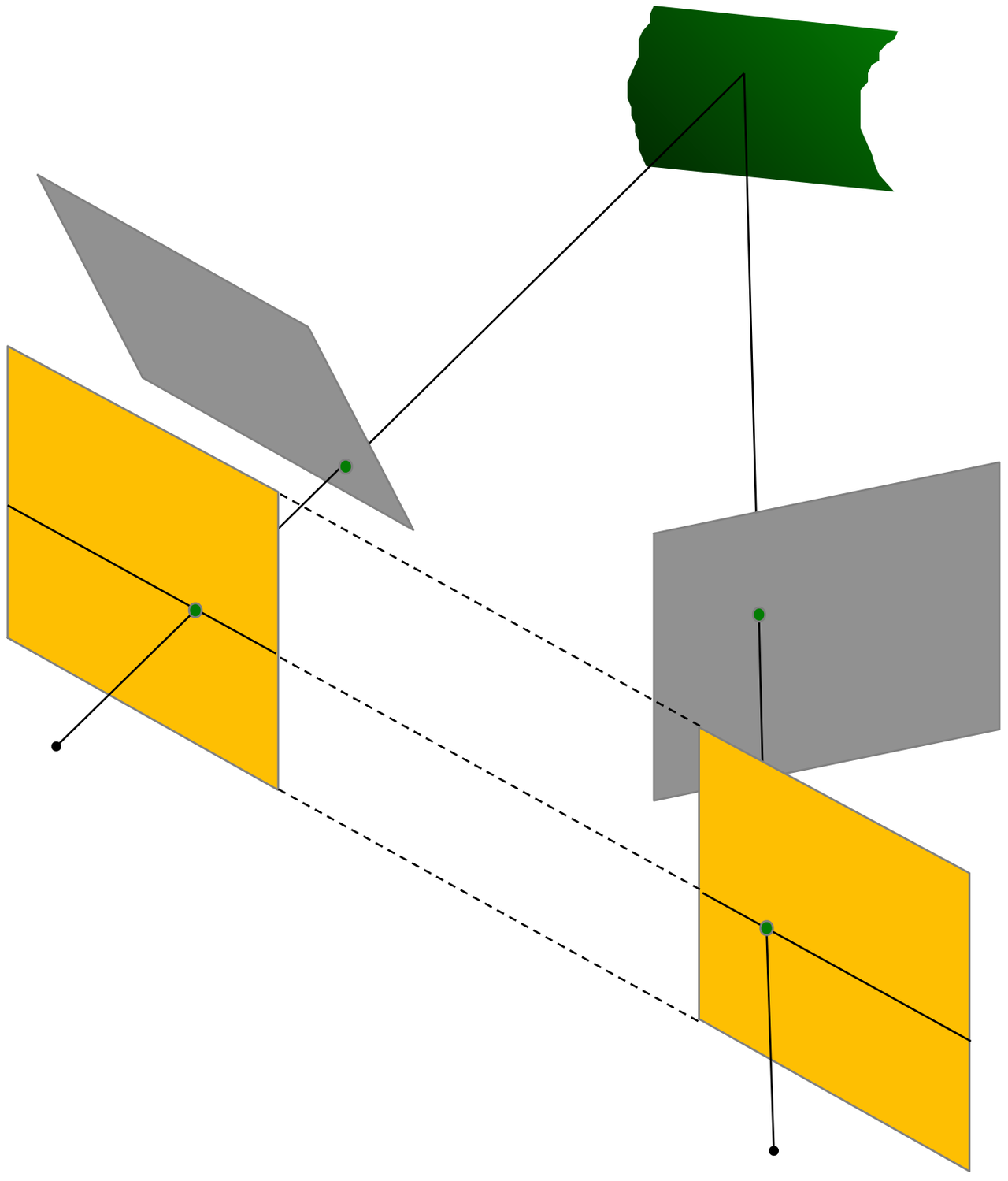
What is stereo rectification?

Reproject image planes onto a common plane parallel to the line between camera centers



What is stereo rectification?

Reproject image planes onto a common plane parallel to the line between camera centers



Need two homographies (3x3 transform), one for each input image reprojection

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

Stereo Rectification

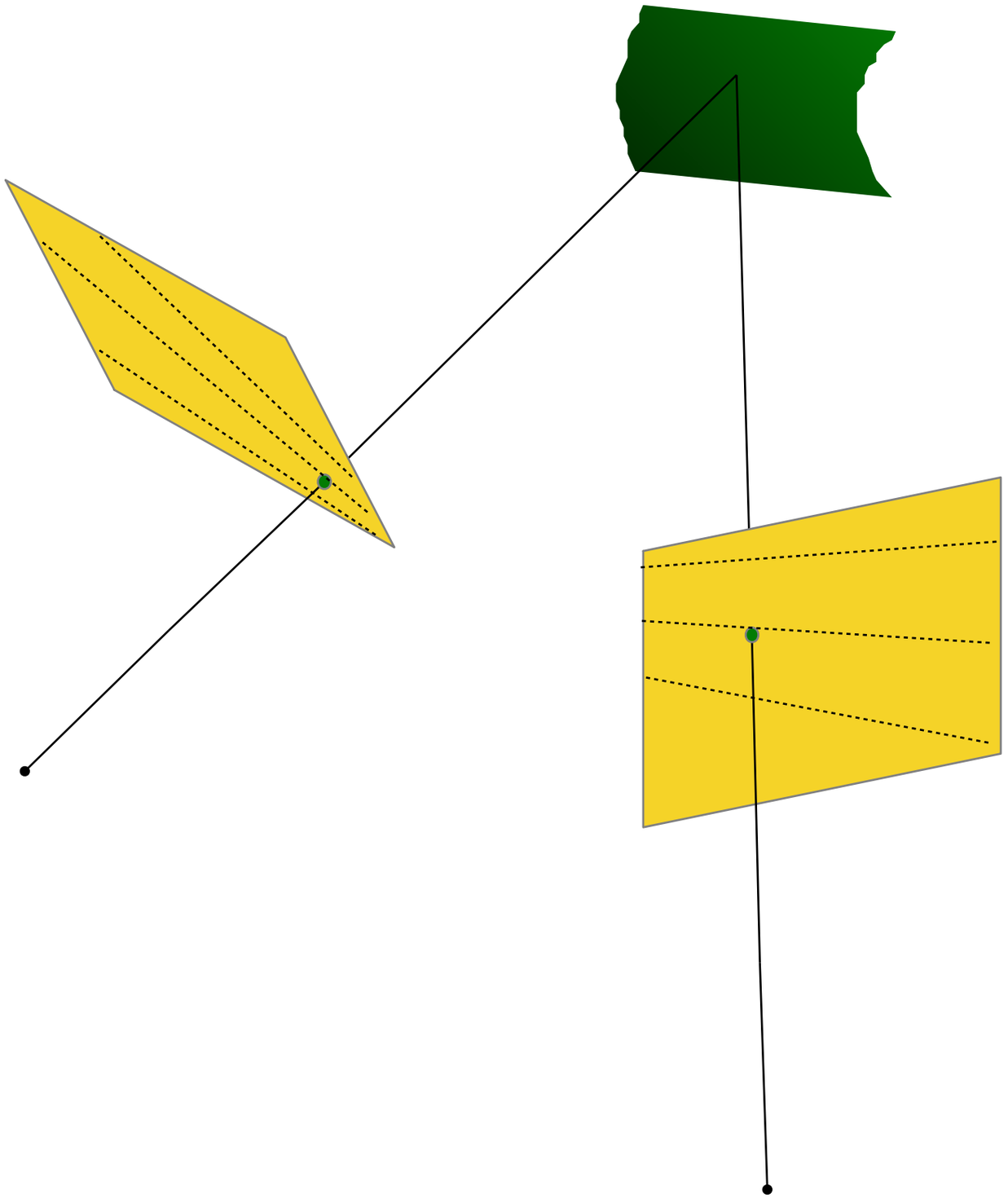
1. **Rotate** the right camera by **R**
   (aligns camera coordinate system orientation only)

2. Rotate (**rectify**) the left camera so that the epipole is at infinity

3. Rotate (**rectify**) the right camera so that the epipole is at infinity

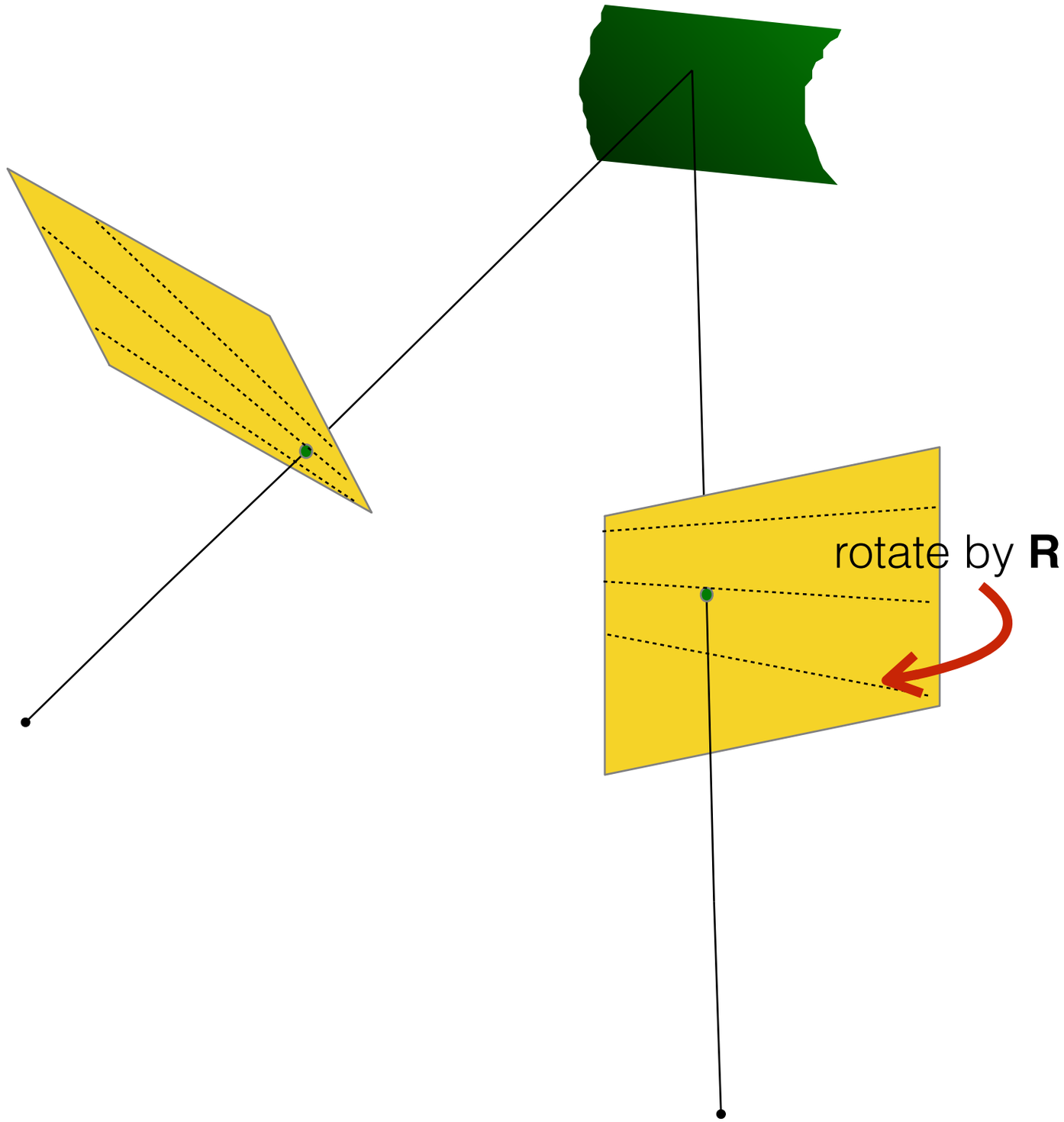
4. Adjust the **scale**

Stereo Rectification:



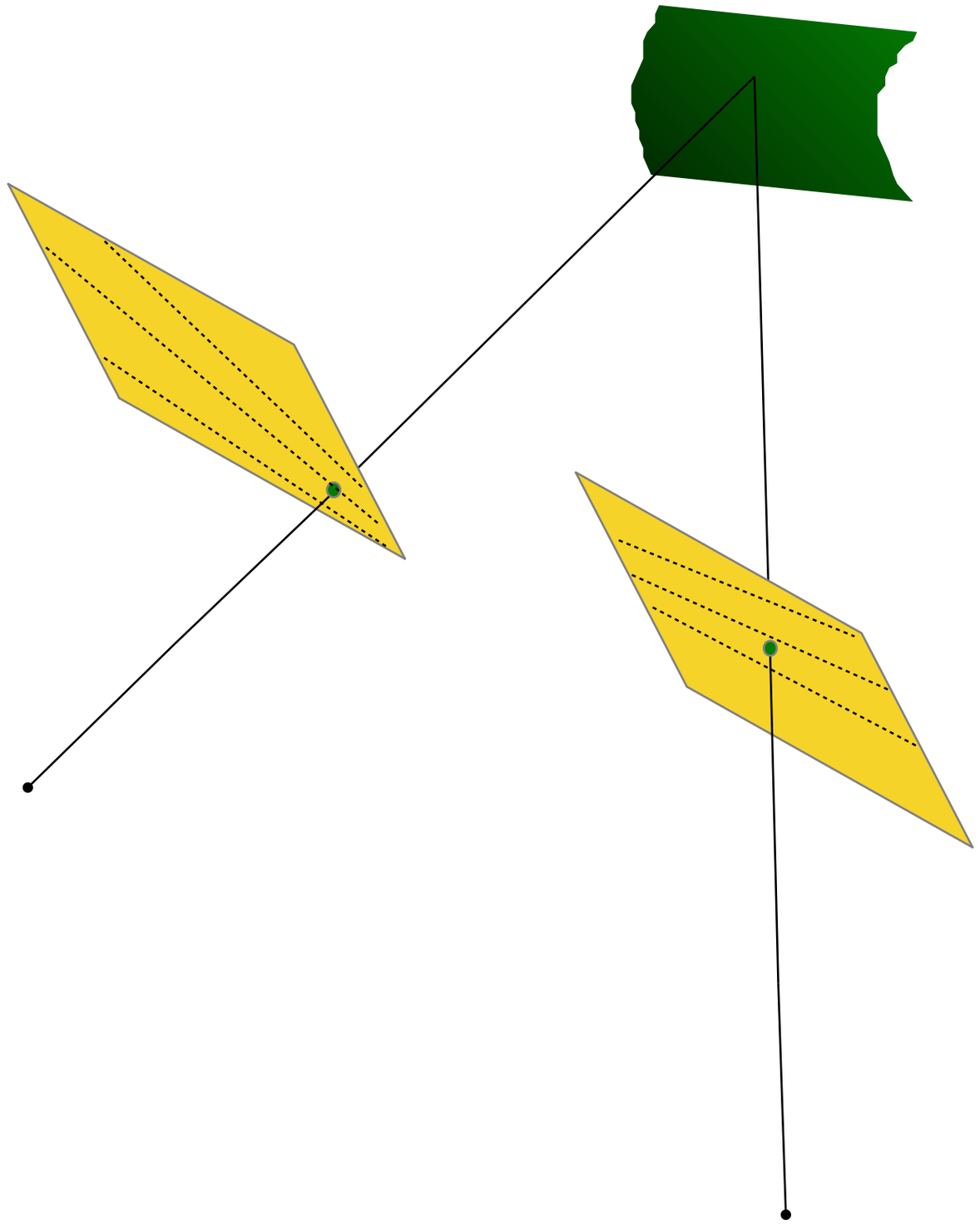
1. Compute **E** to get **R**
2. Rotate right image by **R**
3. Rotate both images by $\mathbf{R_{rect}}$
4. Scale both images by **H**

Stereo Rectification:



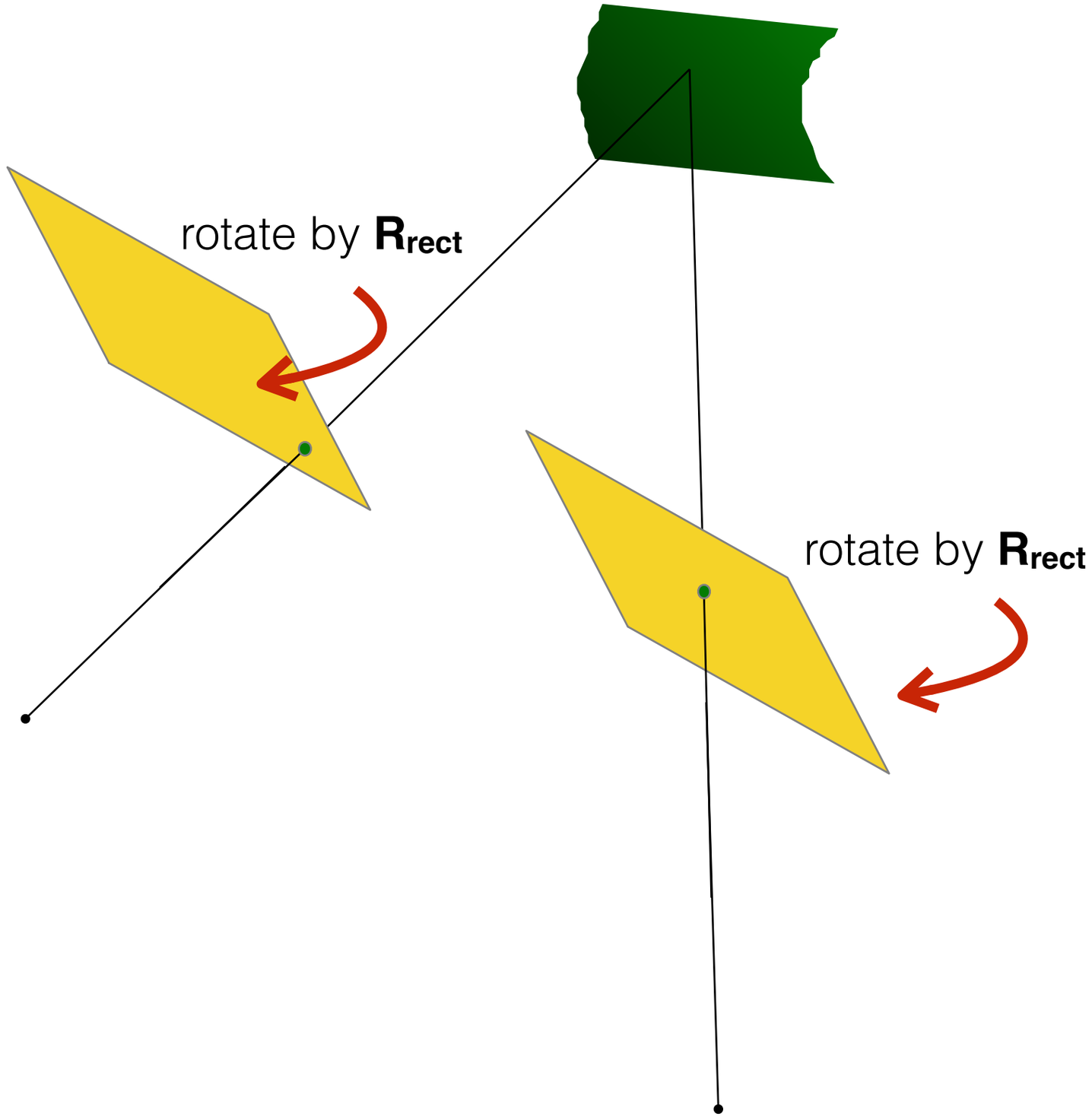
1. Compute **E** to get **R**
2. Rotate right image by **R**
3. Rotate both images by $\mathbf{R_{rect}}$
4. Scale both images by **H**

Stereo Rectification:



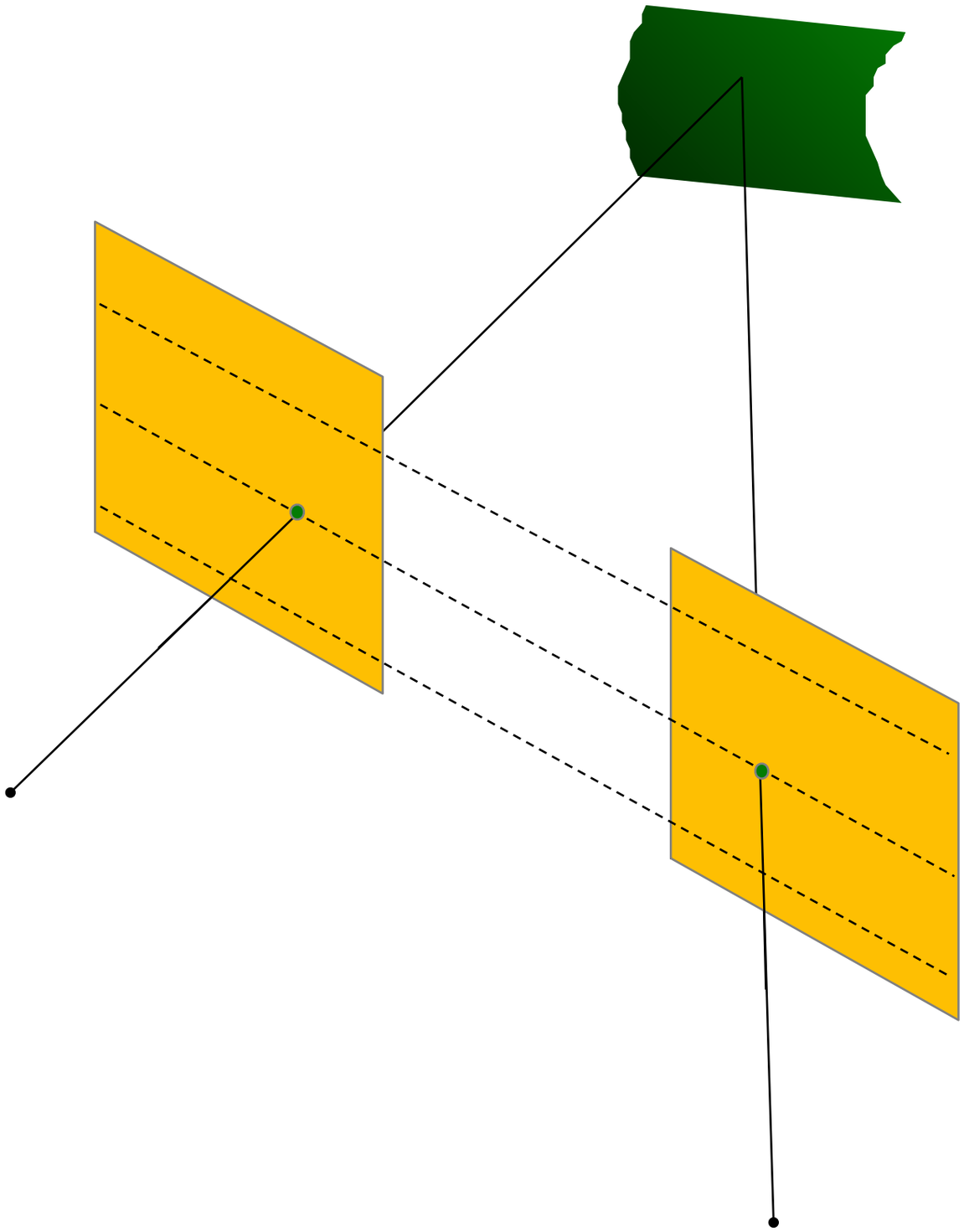
1. Compute **E** to get **R**
2. Rotate right image by **R**
3. Rotate both images by $\mathbf{R_{rect}}$
4. Scale both images by **H**

Stereo Rectification:



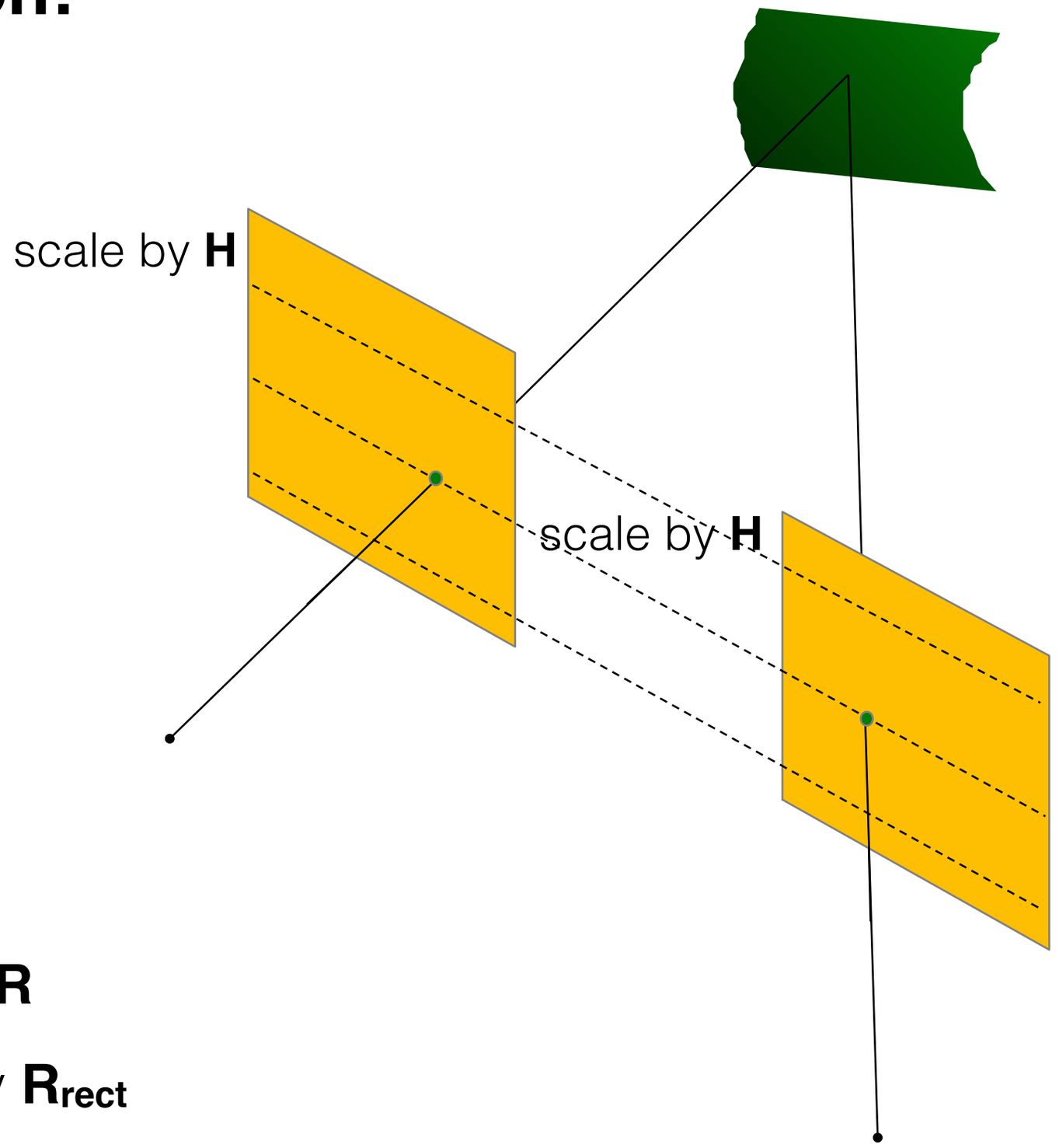
1. Compute **E** to get **R**
2. Rotate right image by **R**
3. Rotate both images by $\mathbf{R_{rect}}$
4. Scale both images by **H**

Stereo Rectification:



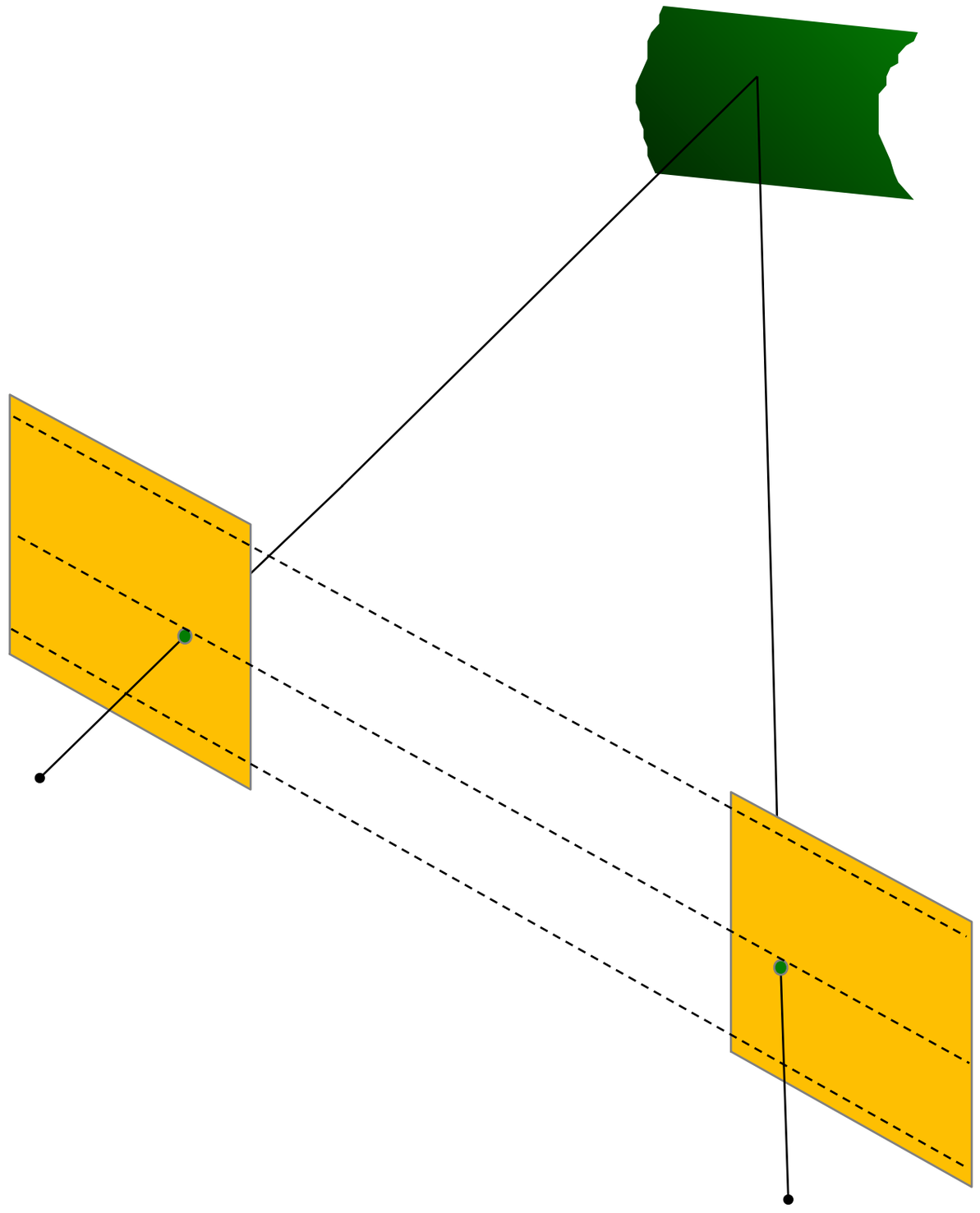
1. Compute **E** to get **R**
2. Rotate right image by **R**
3. Rotate both images by $\mathbf{R_{rect}}$
4. Scale both images by **H**

Stereo Rectification:

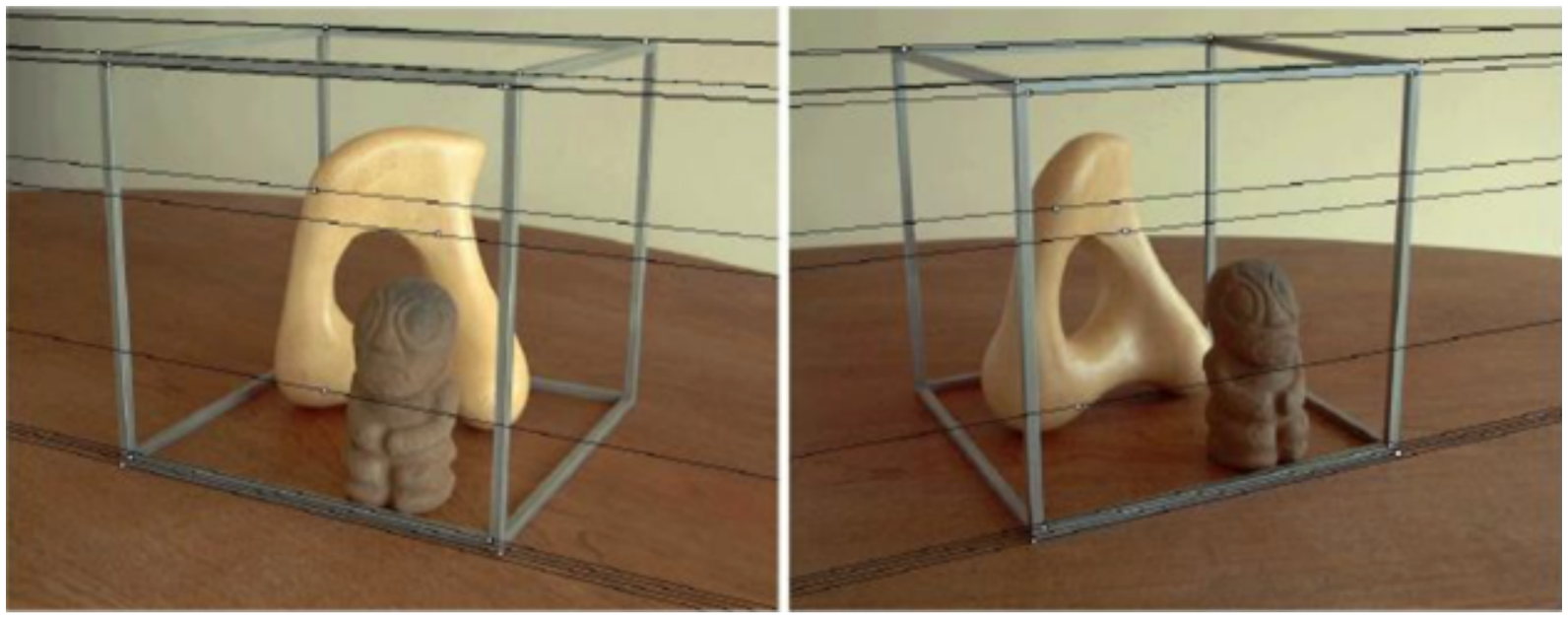


1. Compute **E** to get **R**
2. Rotate right image by **R**
3. Rotate both images by $\mathbf{R_{rect}}$
4. Scale both images by **H**

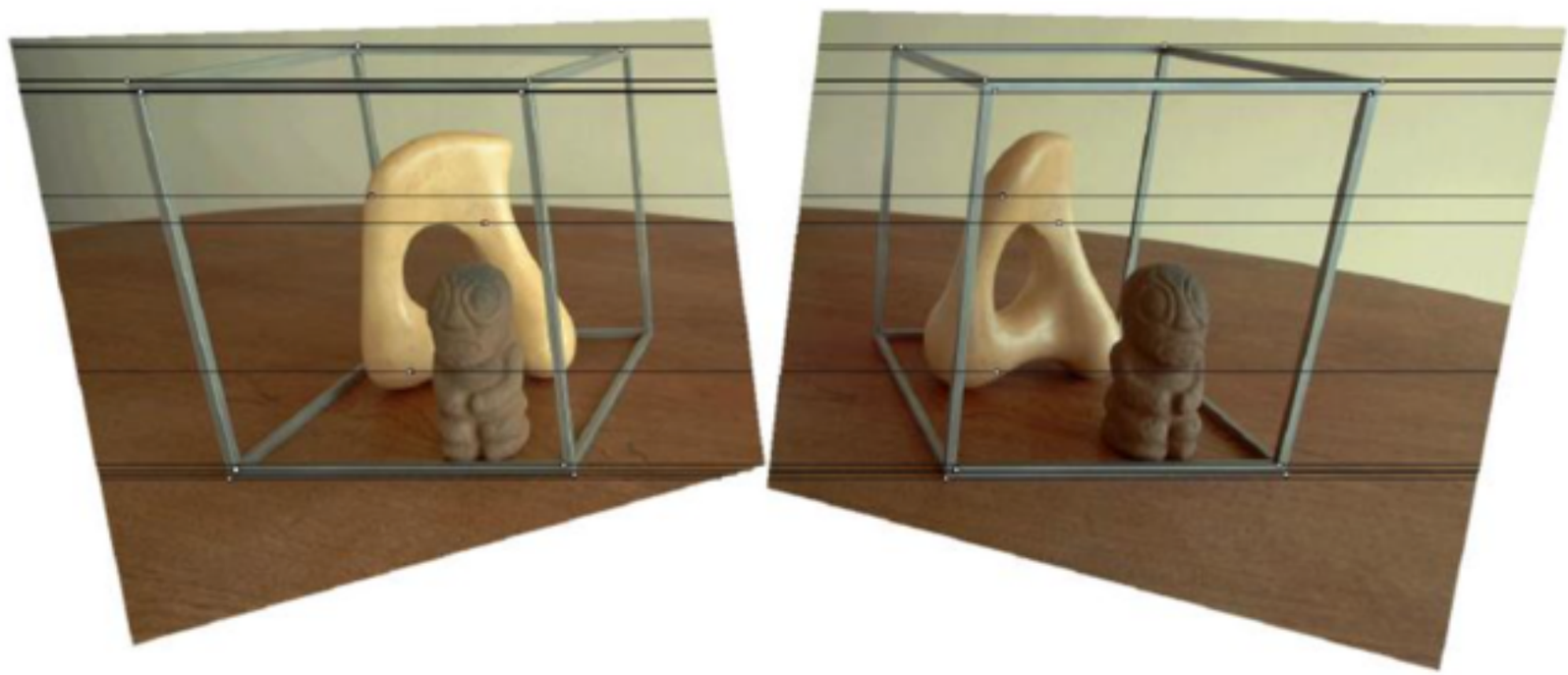
Stereo Rectification:



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



What can we do after rectification?



Setting the epipole to infinity

(Building $\mathbf{R}_{\text{rect}}$ from **e**)

Let $R_{\text{rect}} = \begin{bmatrix} \boldsymbol{r}_1^\top \\ \boldsymbol{r}_2^\top \\ \boldsymbol{r}_3^\top \end{bmatrix}$ Given: epipole **e** (using SVD on E) (translation from **E)**

$$\boldsymbol{r}_1 = \boldsymbol{e}_1 = \frac{T}{||T||}$$

epipole coincides with translation vector

$$\boldsymbol{r}_2 = \frac{1}{\sqrt{T_x^2 + T_y^2}} \begin{bmatrix} -T_y & T_x & 0 \end{bmatrix}$$

cross product of e and the direction vector of the optical axis

$$\boldsymbol{r}_3 = \boldsymbol{r}_1 \times \boldsymbol{r}_2$$

orthogonal vector

If $\boldsymbol{r}_1 = \boldsymbol{e}_1 = \dfrac{T}{||T||}$ and $\boldsymbol{r}_2$ $\boldsymbol{r}_3$ orthogonal

then $R_{\text{rect}}\boldsymbol{e}_1 = \begin{bmatrix} \boldsymbol{r}_1^\top \boldsymbol{e}_1 \\ \boldsymbol{r}_2^\top \boldsymbol{e}_1 \\ \boldsymbol{r}_3^\top \boldsymbol{e}_1 \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ ? \end{bmatrix}$

If $\boldsymbol{r}_1 = \boldsymbol{e}_1 = \dfrac{T}{||T||}$ and $\boldsymbol{r}_2$ $\boldsymbol{r}_3$ orthogonal

then $R_{\text{rect}}\boldsymbol{e}_1 = \begin{bmatrix} \boldsymbol{r}_1^\top \boldsymbol{e}_1 \\ \boldsymbol{r}_2^\top \boldsymbol{e}_1 \\ \boldsymbol{r}_3^\top \boldsymbol{e}_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

Where is this point located on the image plane?

If $\boldsymbol{r}_1 = \boldsymbol{e}_1 = \dfrac{T}{||T||}$ and $\boldsymbol{r}_2$ $\boldsymbol{r}_3$ orthogonal

then $R_{\text{rect}}\boldsymbol{e}_1 = \begin{bmatrix} \boldsymbol{r}_1^\top \boldsymbol{e}_1 \\ \boldsymbol{r}_2^\top \boldsymbol{e}_1 \\ \boldsymbol{r}_3^\top \boldsymbol{e}_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

Where is this point located on the image plane?

At x-infinity

Stereo Rectification Algorithm

1. Estimate **E** using the 8 point algorithm (SVD)
2. Estimate the epipole **e** (SVD of **E**)
3. Build $\mathbf{R}_{rect}$ from **e**
4. Decompose **E** into **R** and **T**
5. Set $\mathbf{R}_1=\mathbf{R}_{rect}$ and $\mathbf{R}_2 = \mathbf{R}\mathbf{R}_{rect}$
6. Rotate each left camera point (warp image)
   $[x' \; y' \; z'] = \mathbf{R}_1 [x \; y \; z]$
7. Rectified points as $\mathbf{p} = f/z'[x' \; y' \; z']$
8. Repeat 6 and 7 for right camera points using $\mathbf{R}_2$

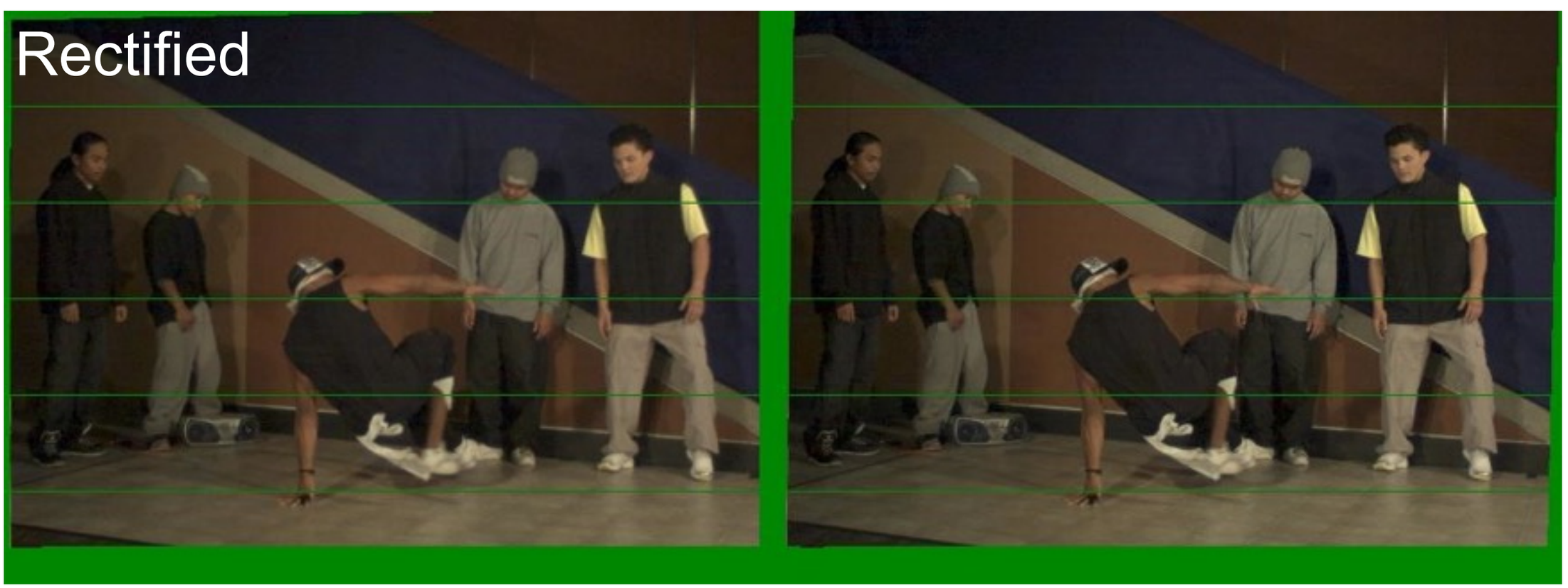
Stereo Rectification Algorithm

1. Estimate **E** using the 8 point algorithm
2. Estimate the epipole **e** (solve **Ee**=0)
3. Build $\mathbf{R}_{rect}$ from **e**
4. Decompose **E** into **R** and **T**
5. Set $\mathbf{R}_1=\mathbf{R}_{rect}$ and $\mathbf{R}_2 = \mathbf{R}\mathbf{R}_{rect}$
6. Rotate each left camera point $\mathbf{x}' \sim \mathbf{H}\mathbf{x}$ where $\mathbf{H} = \mathbf{K}\mathbf{R}_1$

   *You may need to alter the focal length (inside **K**) to keep points within the original image size
7. Repeat 6 and 7 for right camera points using $\mathbf{R}_2$

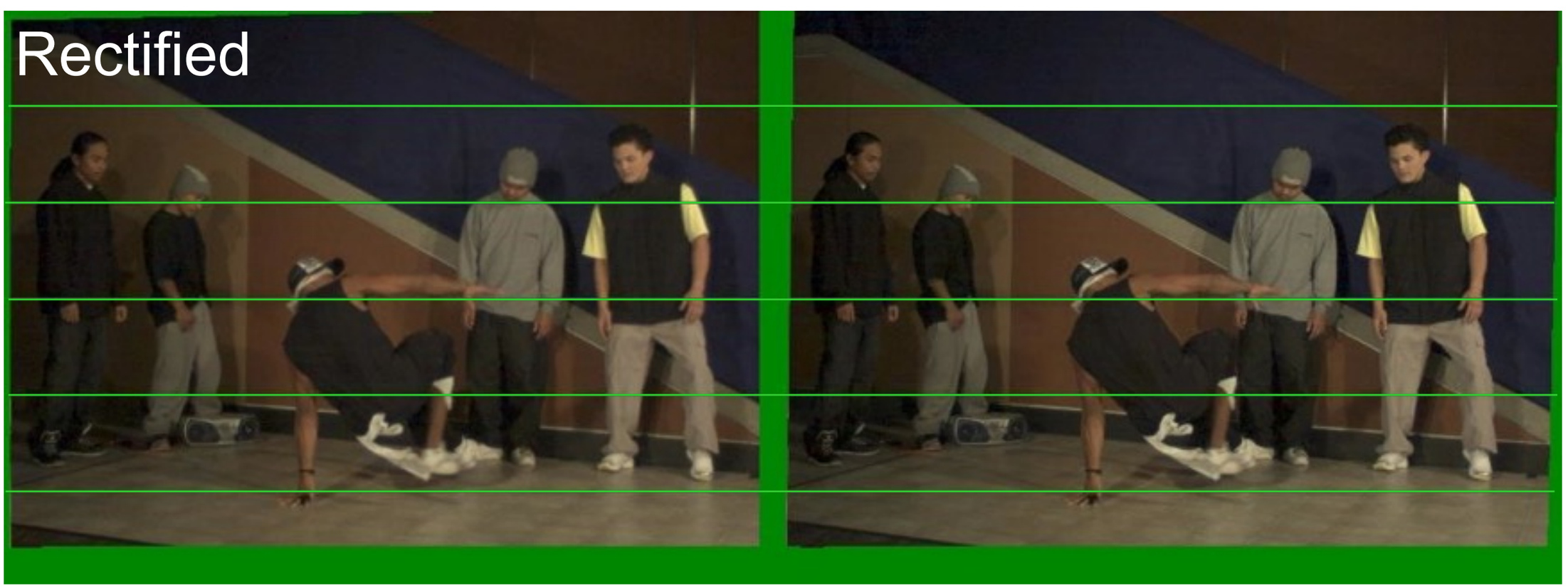

Unrectified


Unrectified


Rectified

Unrectified

Rectified



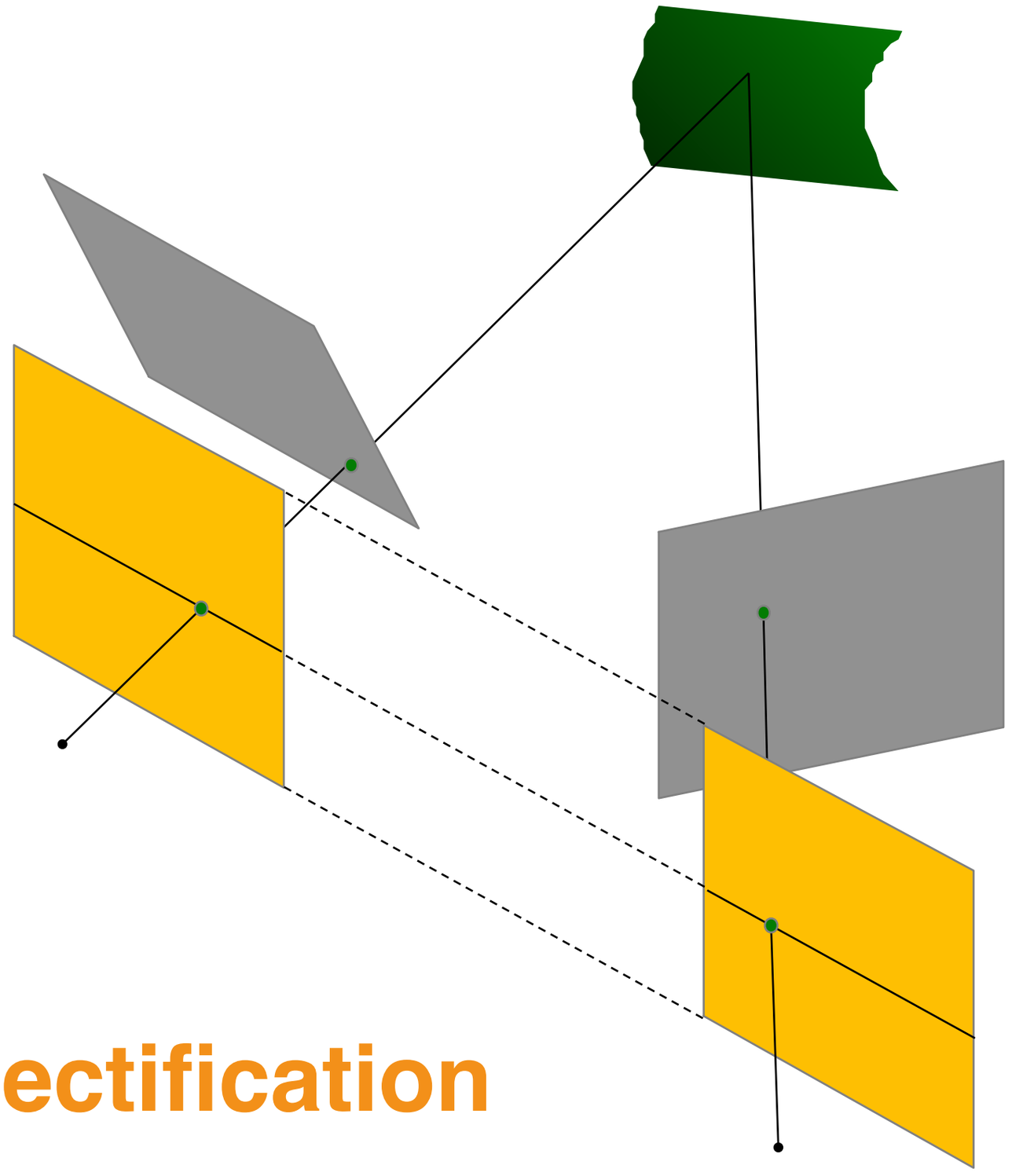
Stereo Matching

16-385 Computer Vision (Kris Kitani)

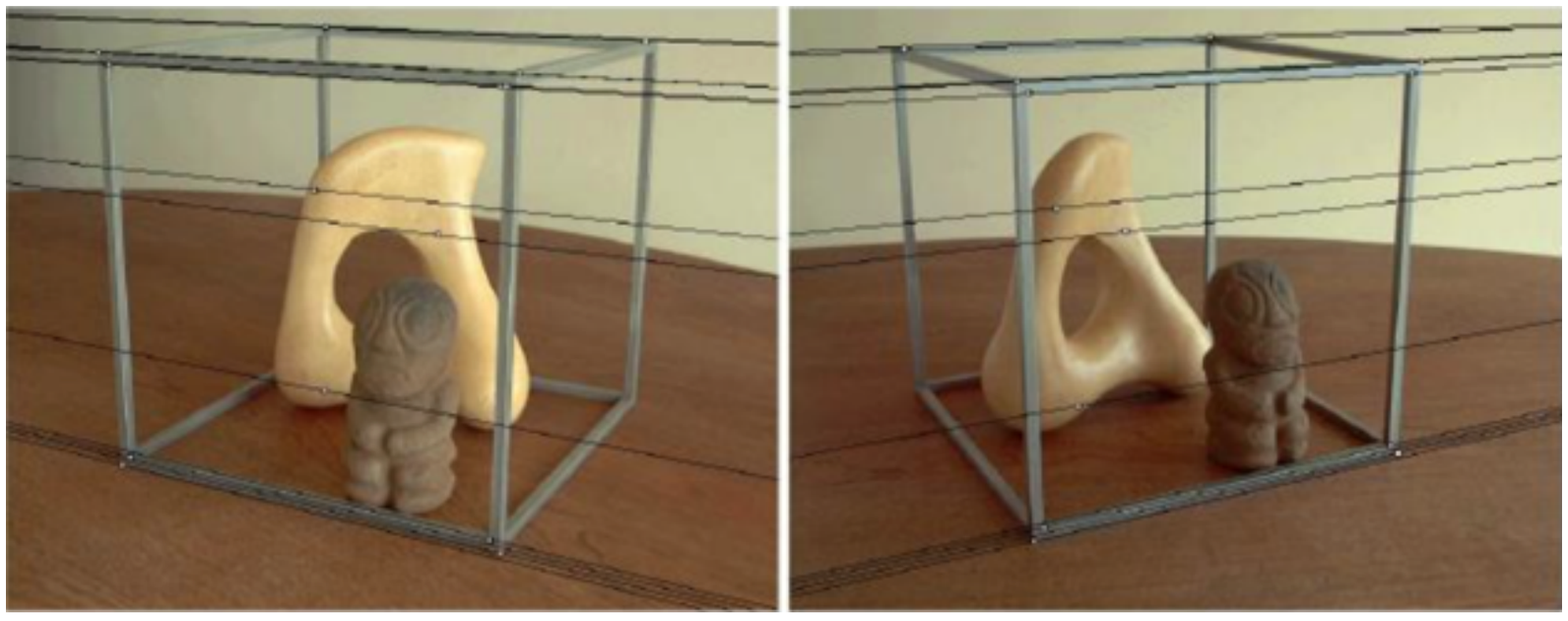
Carnegie Mellon University

What is stereo rectification?

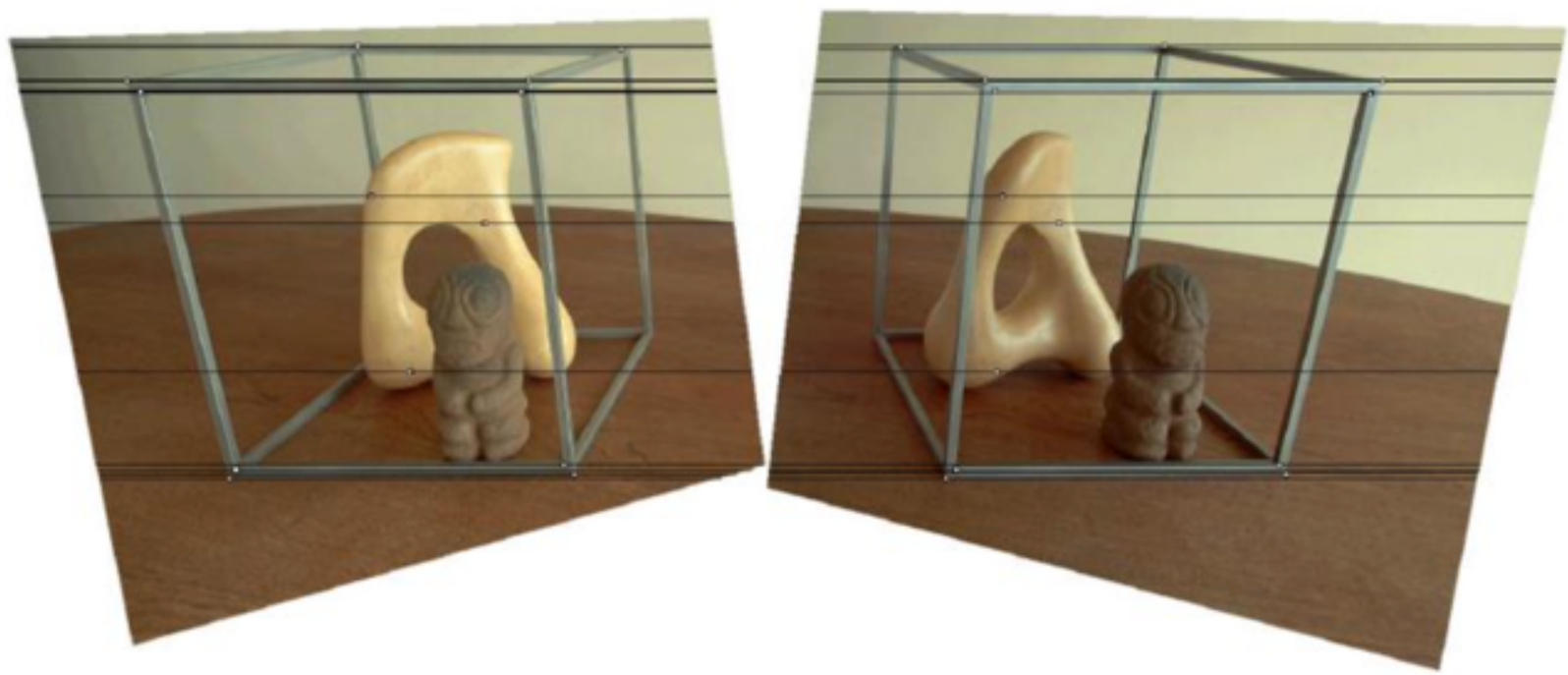
Reproject image planes onto a common plane parallel to the line between camera centers



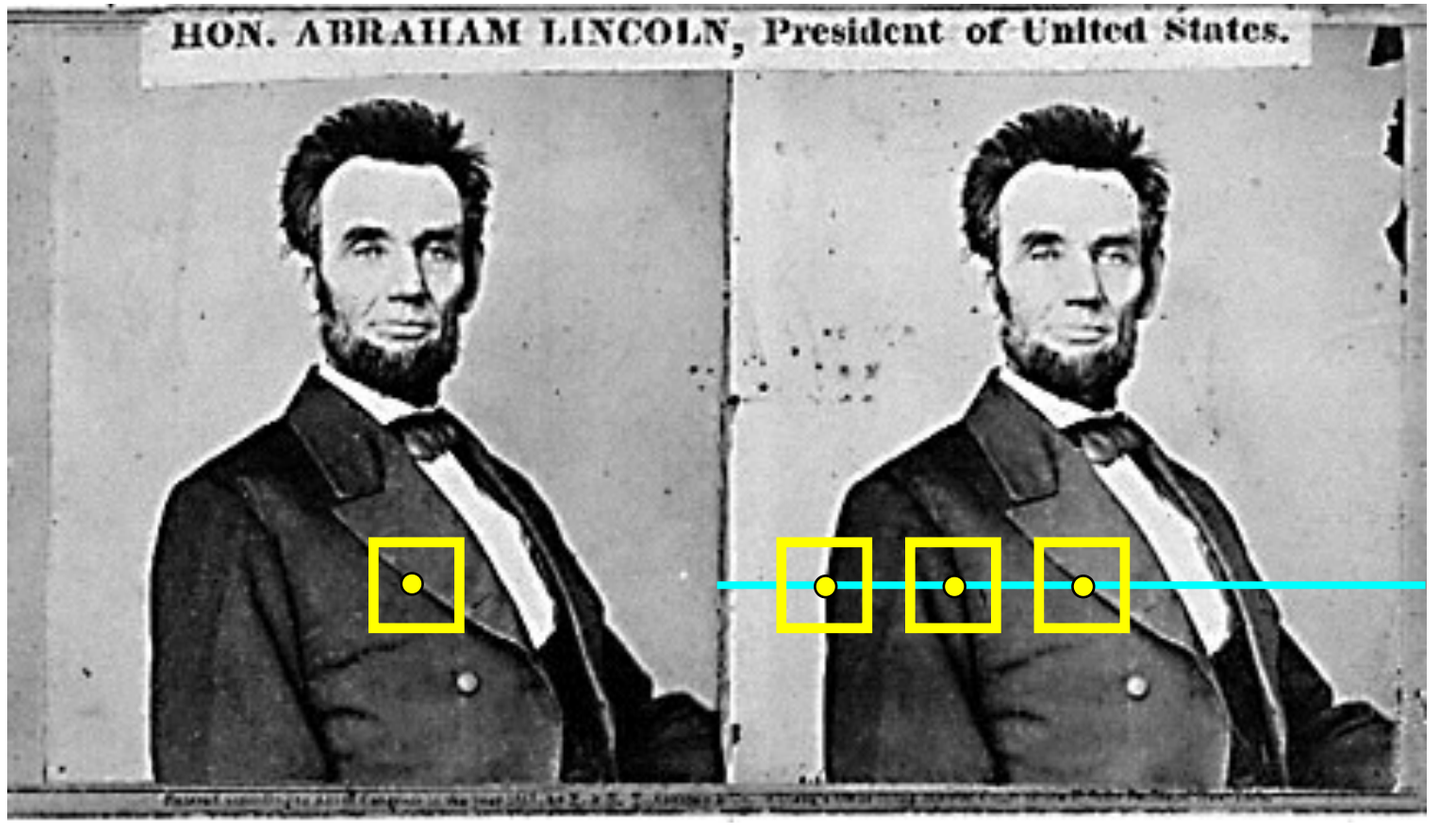
Recall: Stereo Rectification



What can we do after rectification?



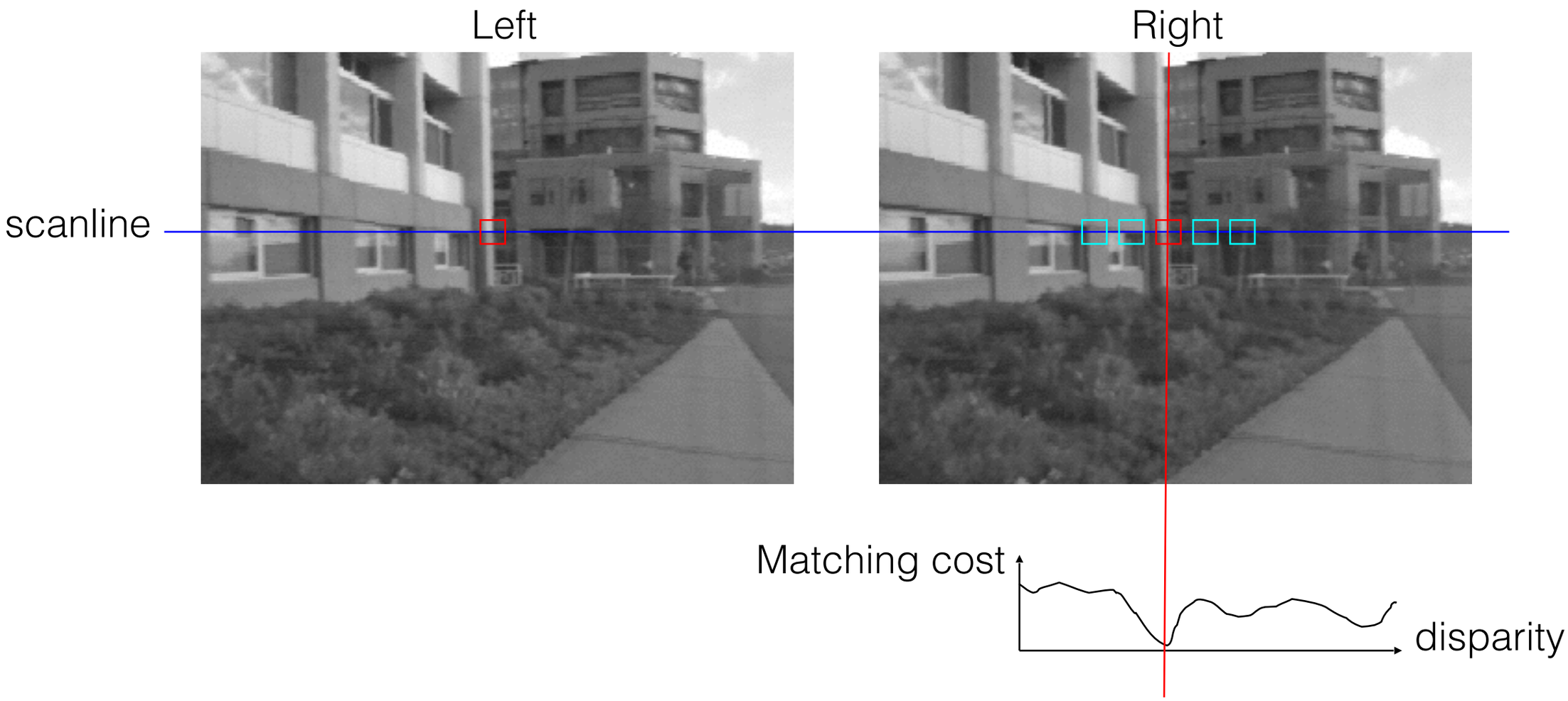
Depth Estimation via Stereo Matching



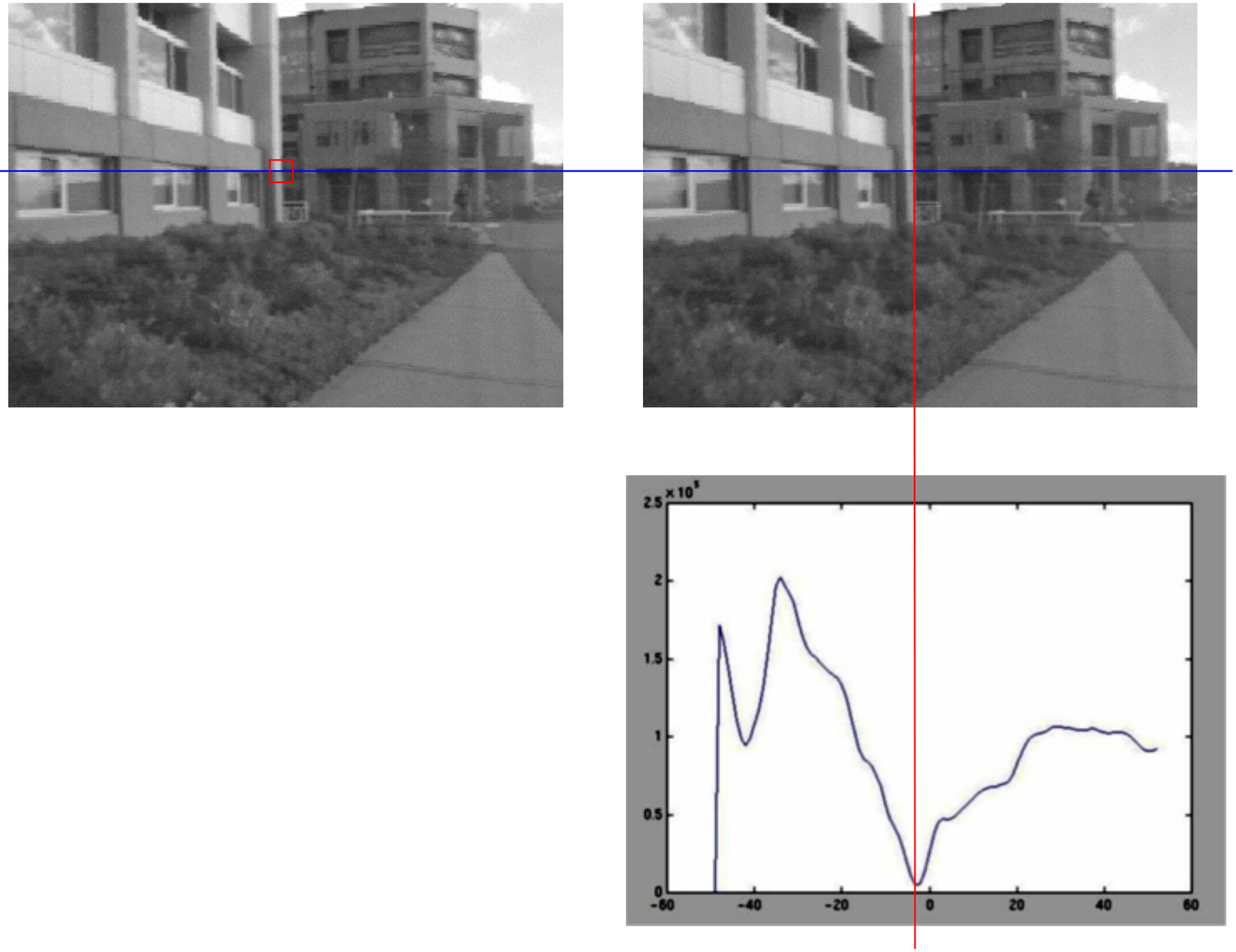
1. Rectify images
   (make epipolar lines horizontal)
2. For each pixel
   a. Find epipolar line
   b. Scan line for best match
   c. Compute depth from disparity

$$Z = \frac{bf}{d}$$

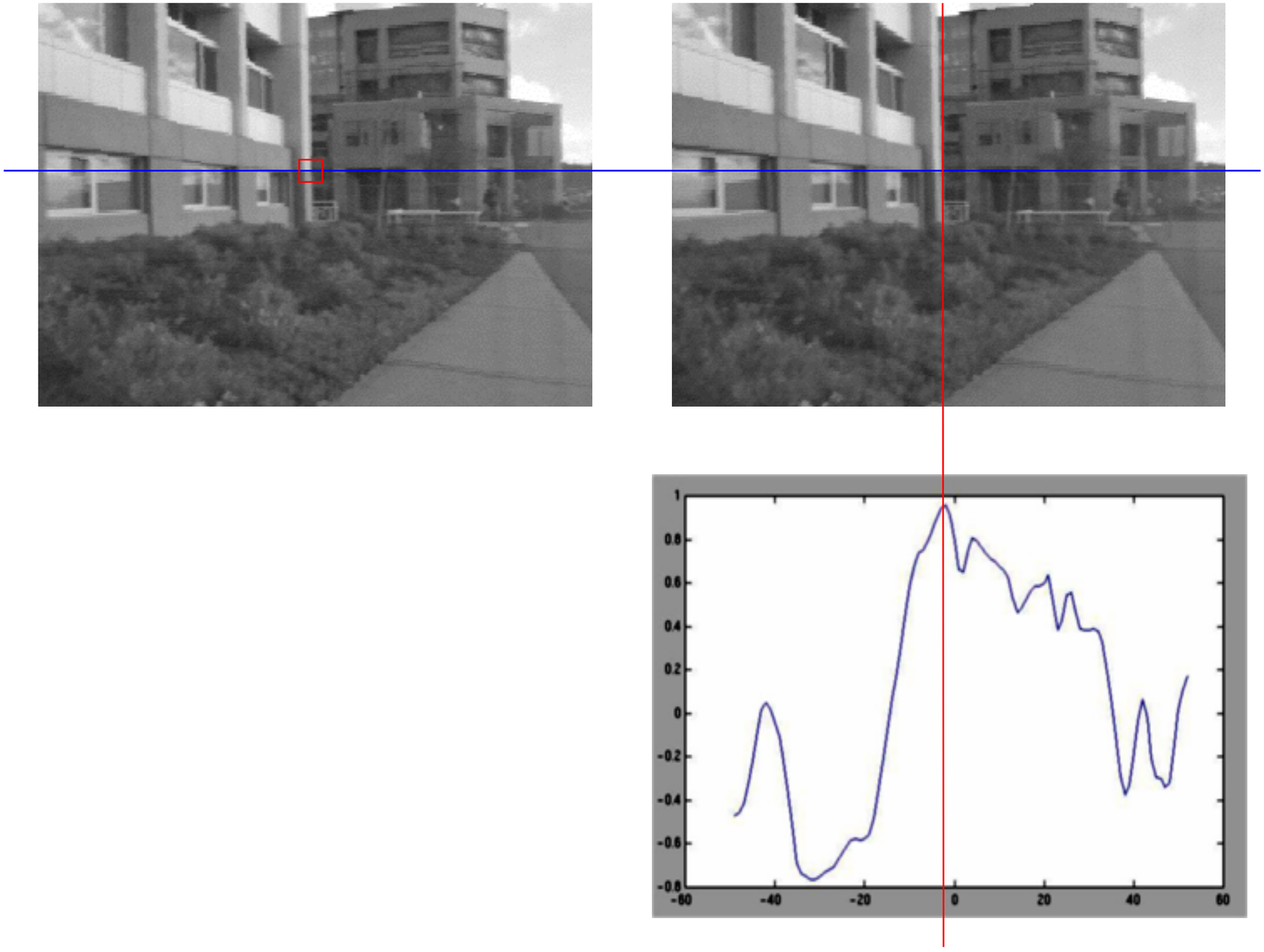
Stereo Block Matching



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

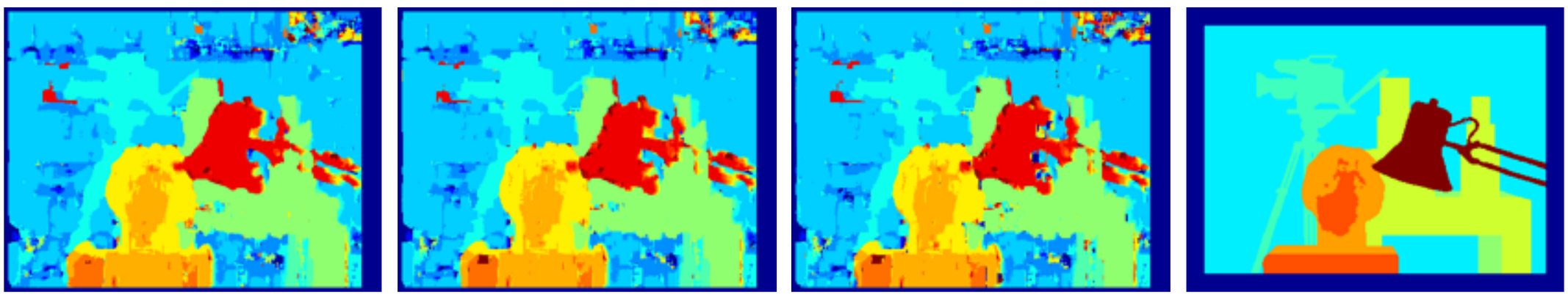


SSD



Normalized cross-correlation

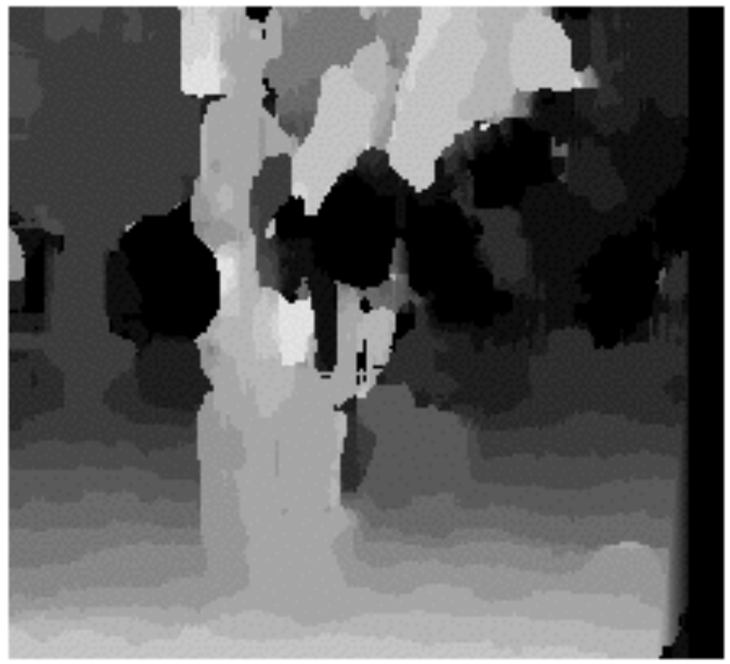
Similarity Measure	Formula
Sum of Absolute Differences (SAD)	$\sum_{(i,j)\in W} \lvert I_1(i,j) - I_2(x+i,y+j)\rvert$
Sum of Squared Differences (SSD)	$\sum_{(i,j)\in W} \left(I_1(i,j) - I_2(x+i,y+j)\right)^2$
Zero-mean SAD	$\sum_{(i,j)\in W} \lvert I_1(i,j) - \bar{I}_1(i,j) - I_2(x+i,y+j) + \bar{I}_2(x+i,y+j)\rvert$
Locally scaled SAD	$\sum_{(i,j)\in W} \lvert I_1(i,j) - \frac{\bar{I}_1(i,j)}{\bar{I}_2(x+i,y+j)} I_2(x+i,y+j)\rvert$
Normalized Cross Correlation (NCC)	$\frac{\sum_{(i,j)\in W} I_1(i,j).I_2(x+i,y+j)}{\sqrt[2]{\sum_{(i,j)\in W} I_1^2(i,j).\sum_{(i,j)\in W} I_2^2(x+i,y+j)}}$



SAD SSD NCC Ground truth

Effect of window size

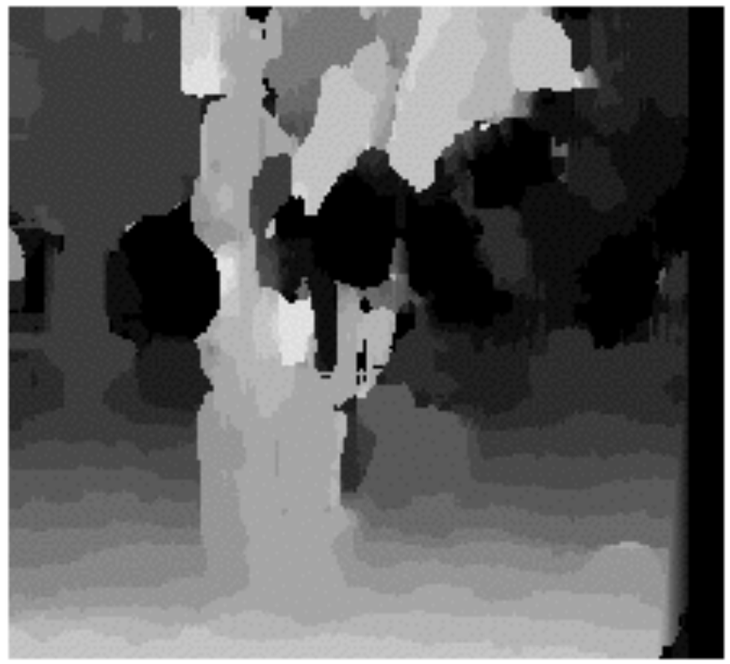
W = 3



W = 20

Effect of window size

W = 3



W = 20

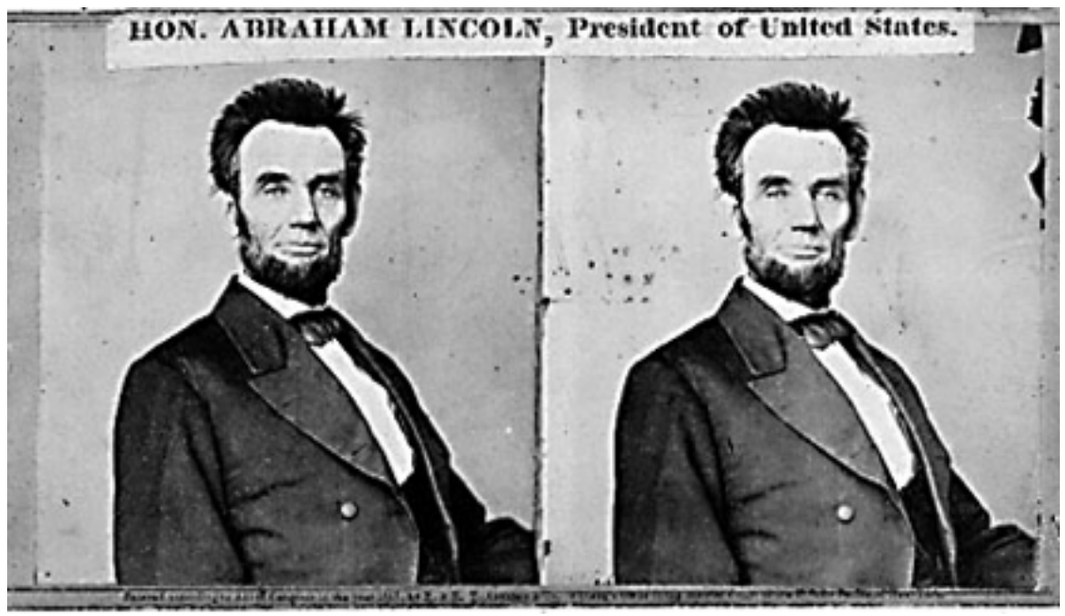
Smaller window

+ More detail
- More noise

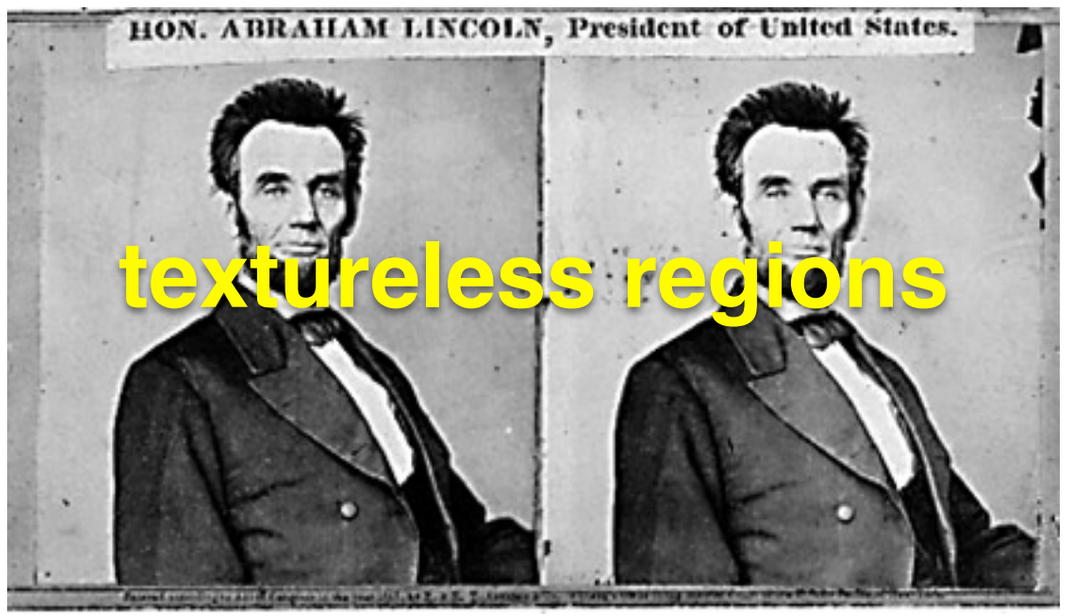
Larger window

+ Smoother disparity maps
- Less detail
- Fails near boundaries

When will stereo block matching fail?

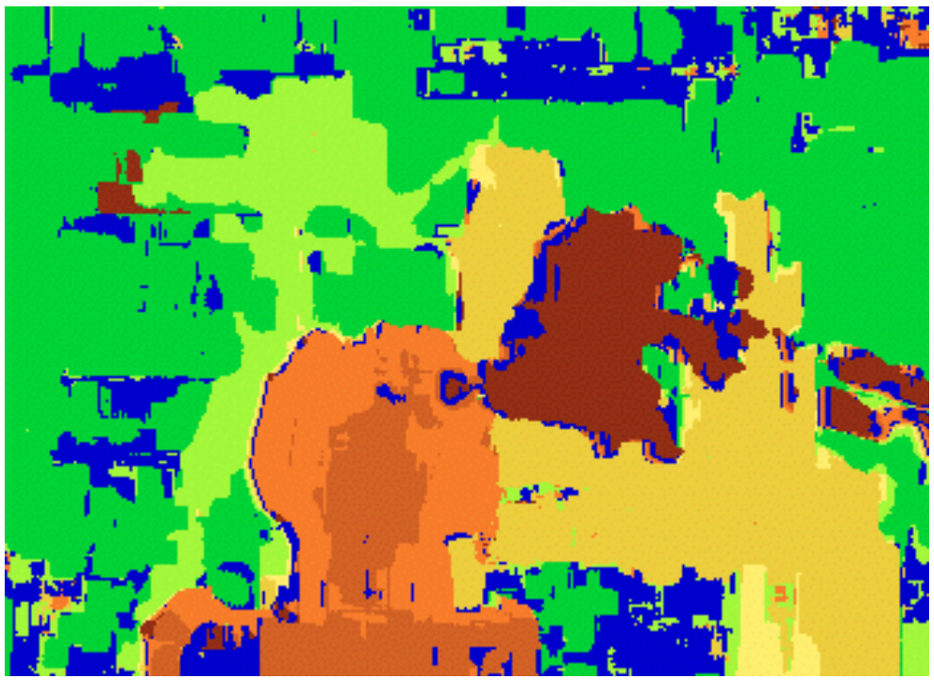


When will stereo block matching fail?



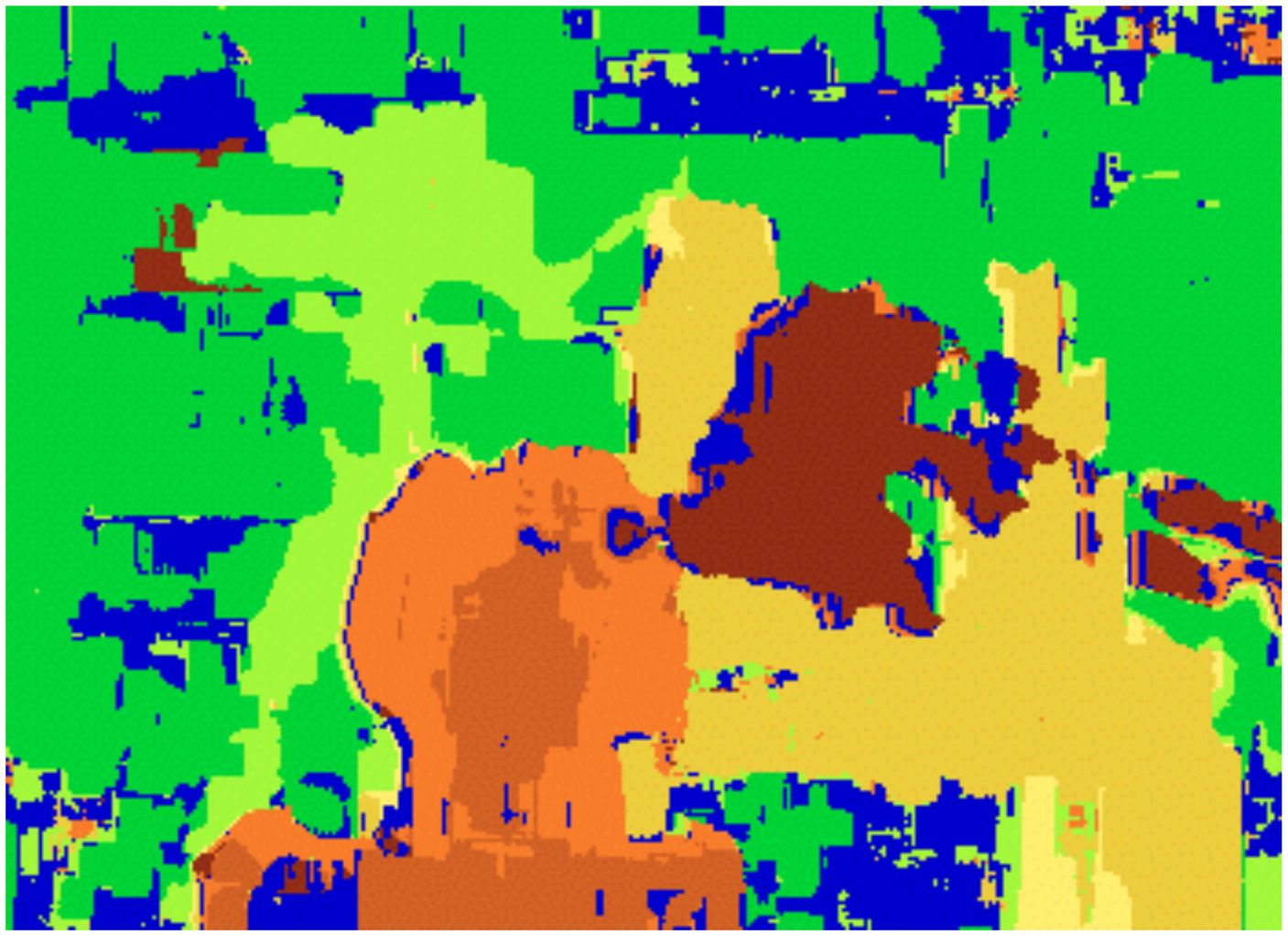
Improving Stereo Block Matching

Block matching

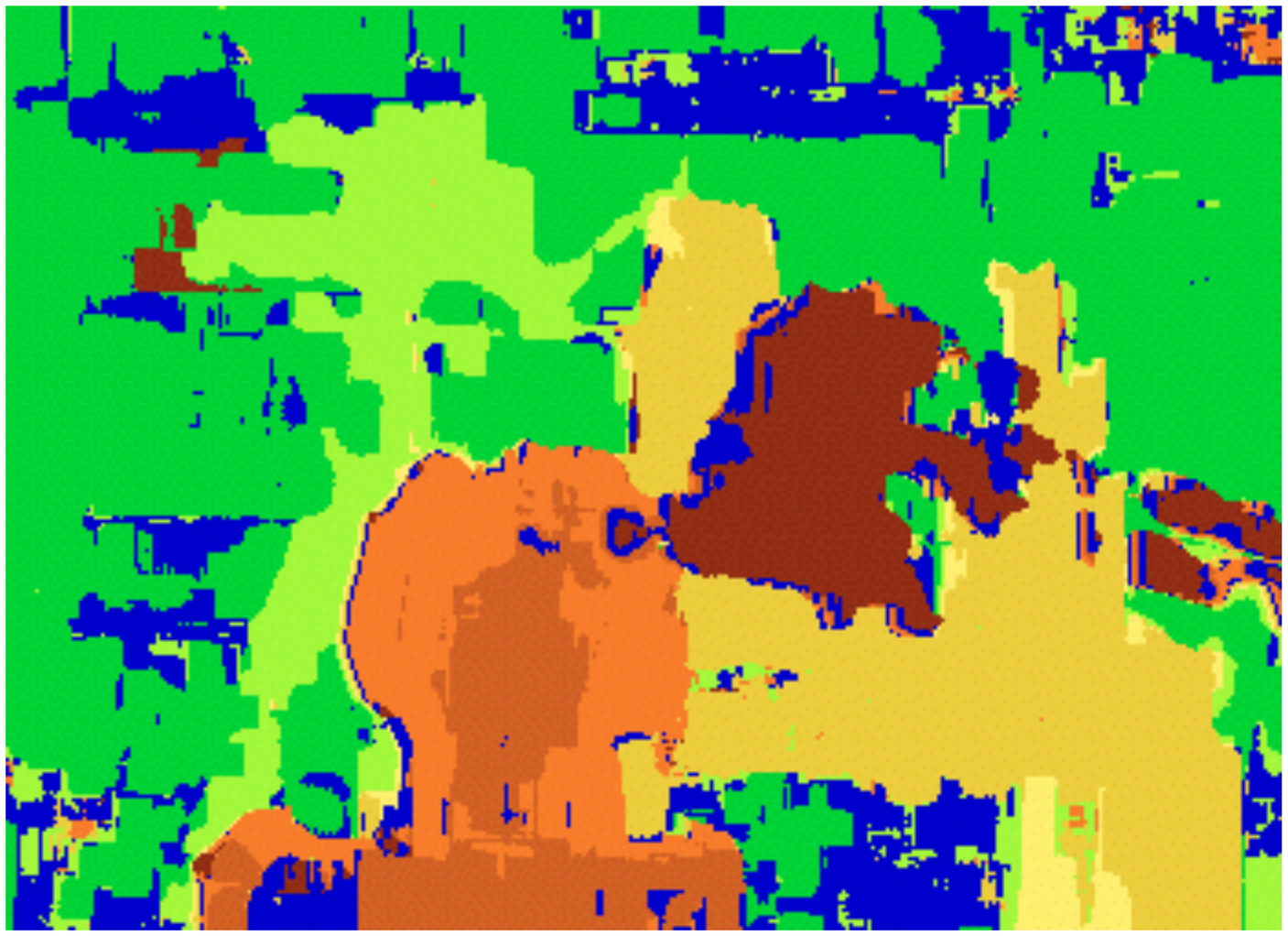


Ground truth

What are some problems with the result?



How can we improve depth estimation?



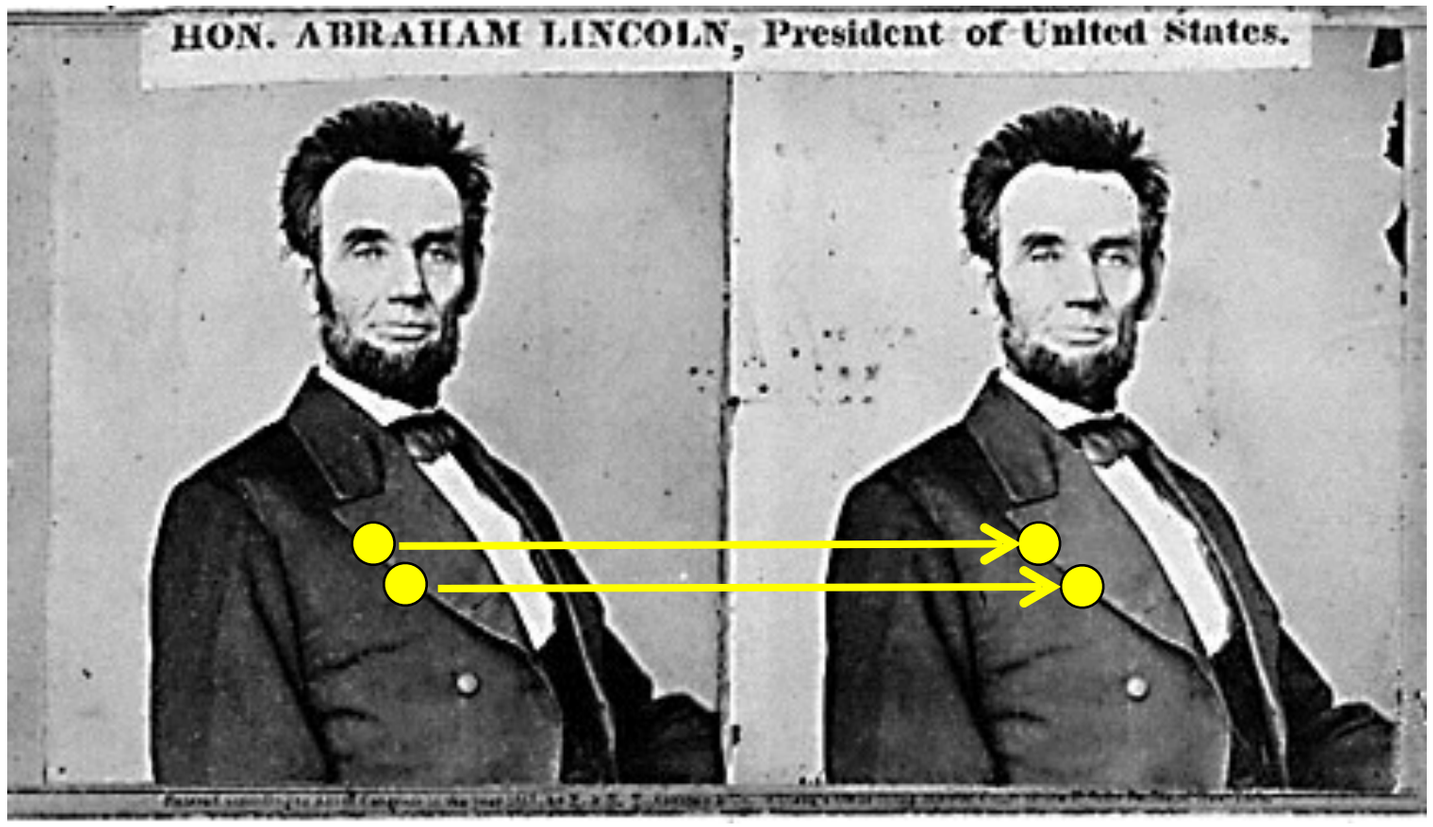
How can we improve depth estimation?

Too many discontinuities.
We expect disparity values to change slowly.

Let's make an assumption:
depth should change smoothly

Stereo matching as …

Energy Minimization



What defines a good stereo correspondence?

1. **Match quality**
   - Want each pixel to find a good match in the other image
2. **Smoothness**
   - If two pixels are adjacent, they should (usually) move about the same amount

energy function
(for one pixel)

$$E(d) = \underbrace{E_d(d)}_{\text{data term}} + \underbrace{\lambda E_s(d)}_{\text{smoothness term}}$$

energy function
(for one pixel)

$$E(d) = \underbrace{E_d(d)}_{\text{data term}} + \lambda \underbrace{E_s(d)}_{\text{smoothness term}}$$

Want each pixel to find a good match in the other image

(block matching result)

Adjacent pixels should (usually) move about the same amount

(smoothness function)

$$E(d) = E_d(d) + \lambda E_s(d)$$

$$E_d(d) = \sum_{(x,y) \in I} C(x, y, d(x, y))$$

data term

SSD distance between windows
centered at I(x, y) and J(x+ d(x,y), y)

$$E(d) = E_d(d) + \lambda E_s(d)$$

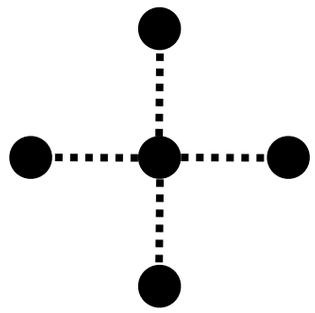
$$E_d(d) = \sum_{(x,y) \in I} C(x, y, d(x, y))$$

SSD distance between windows centered at I(x, y) and J(x+ d(x,y), y)

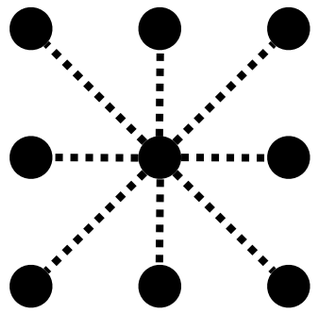
$$E_s(d) = \sum_{(p,q) \in \mathcal{E}} V(d_p, d_q)$$

smoothness term

$\mathcal{E}$: set of neighboring pixels



4-connected neighborhood



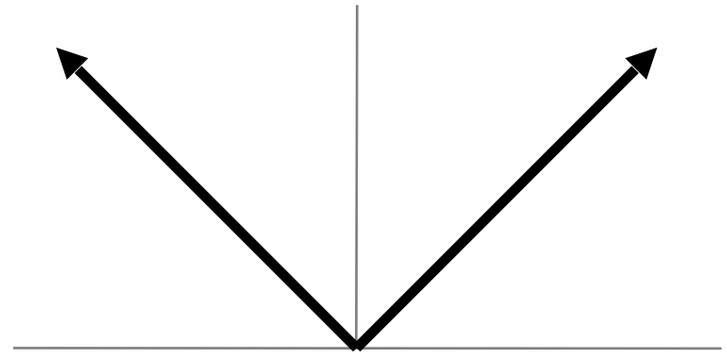
8-connected neighborhood

$$E_s(d) = \sum_{(p,q)\in\mathcal{E}} V(d_p, d_q)$$

smoothness term

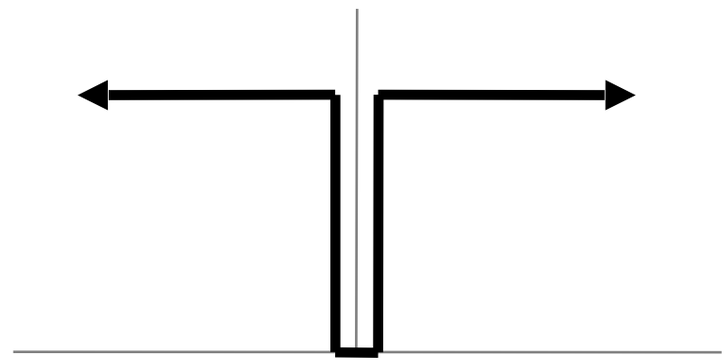
$$V(d_p, d_q) = |d_p - d_q|$$

L_1 distance



$$V(d_p, d_q) = \begin{cases} 0 & \text{if } d_p = d_q \\ 1 & \text{if } d_p \neq d_q \end{cases}$$

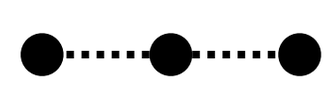
“Potts model”



Dynamic Programming

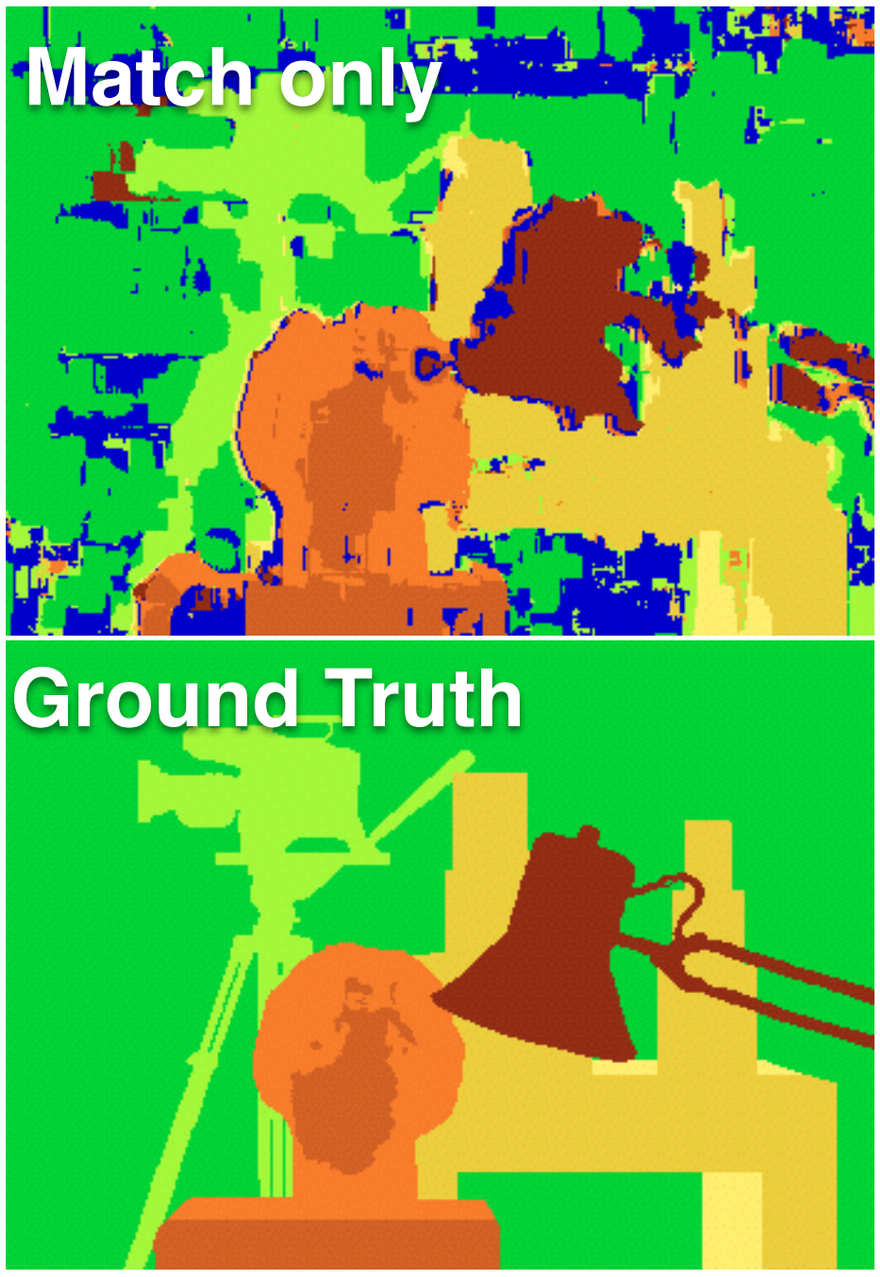
$$E(d) = E_d(d) + \lambda E_s(d)$$

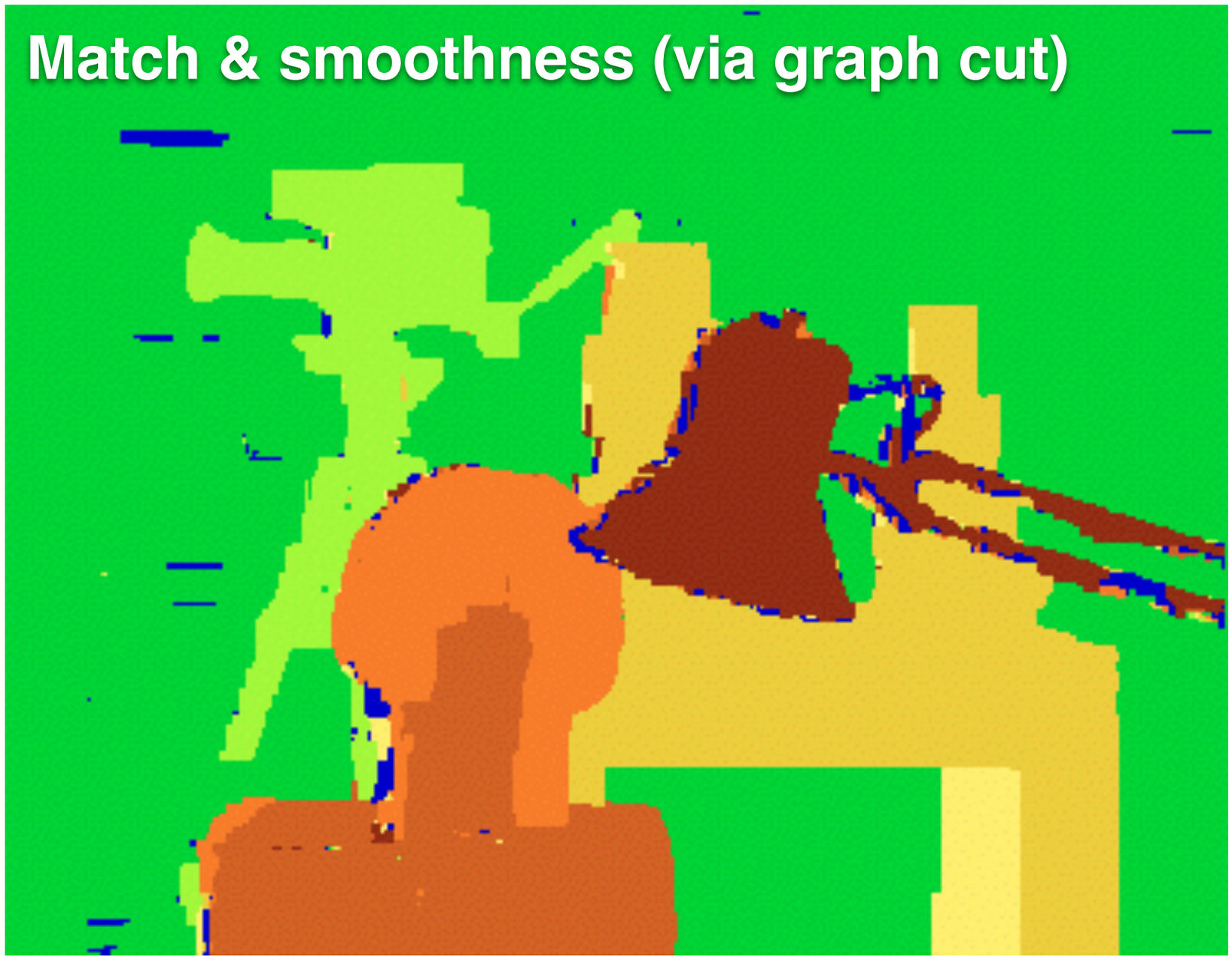
Can minimize this independently per scanline using dynamic programming (DP)



$D(x, y, d)$: minimum cost of solution such that d(x,y) = d

$$D(x, y, d) = C(x, y, d) + \min_{d'} \{D(x-1, y, d') + \lambda \, |d - d'|\}$$





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