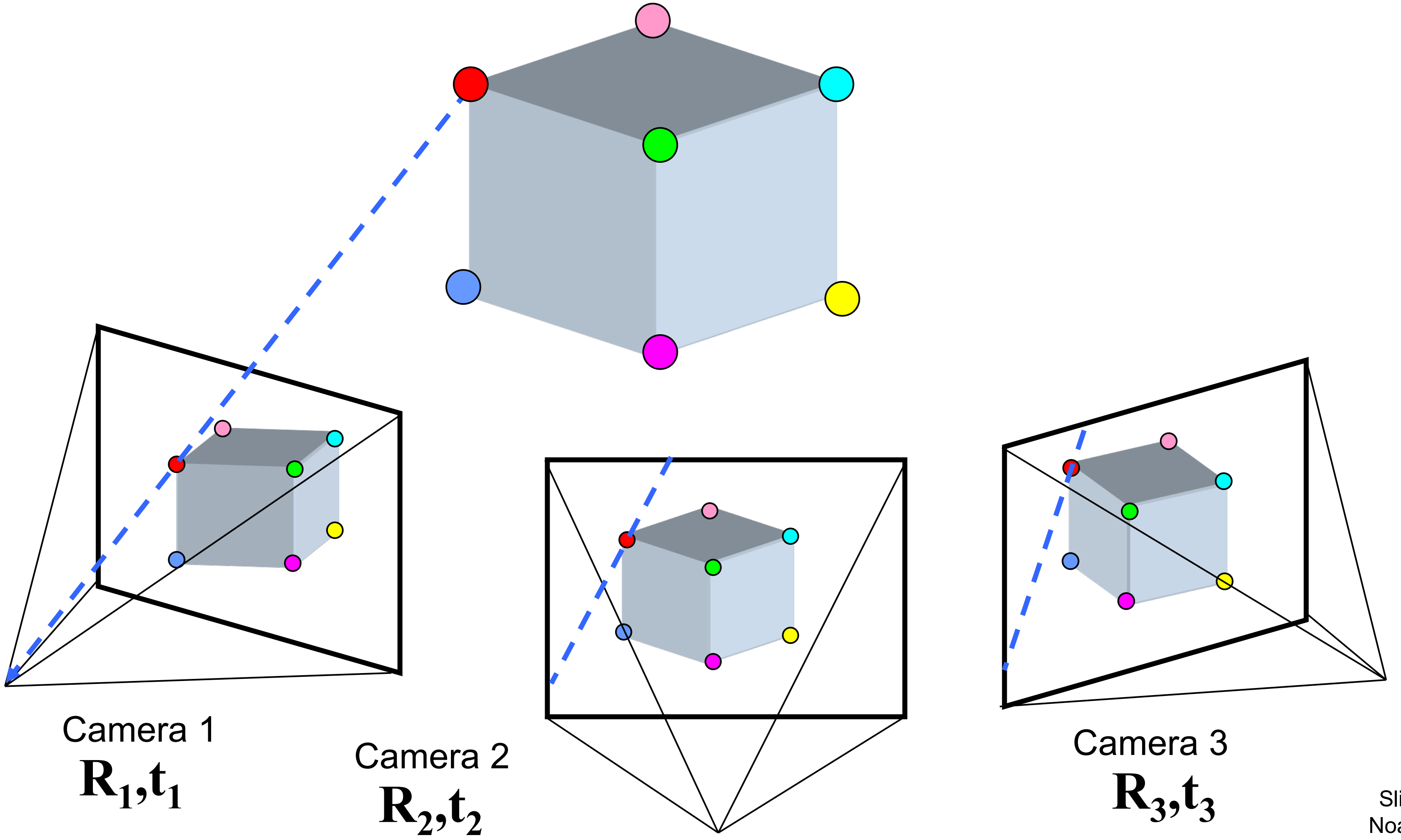
Computer Vision

A.A. 2024-2025

Lecture 13: Epipolar Geometry and Fundamental Matrix

Multi-view geometry problems

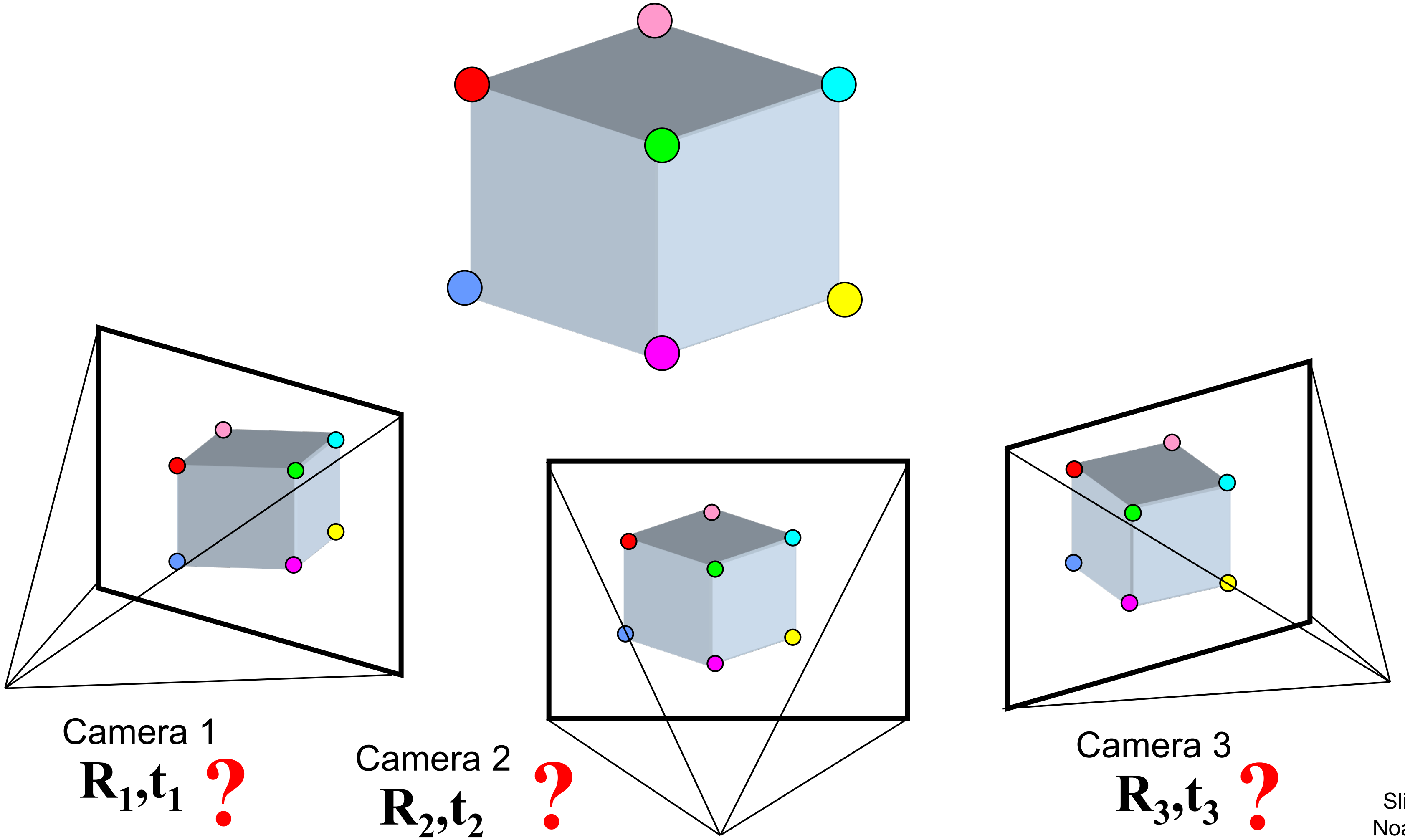
- **Stereo correspondence:** Given known camera parameters and a point in one of the images, where could its corresponding points be in the other images?



Slide credit: Noah Snavely

Multi-view geometry problems

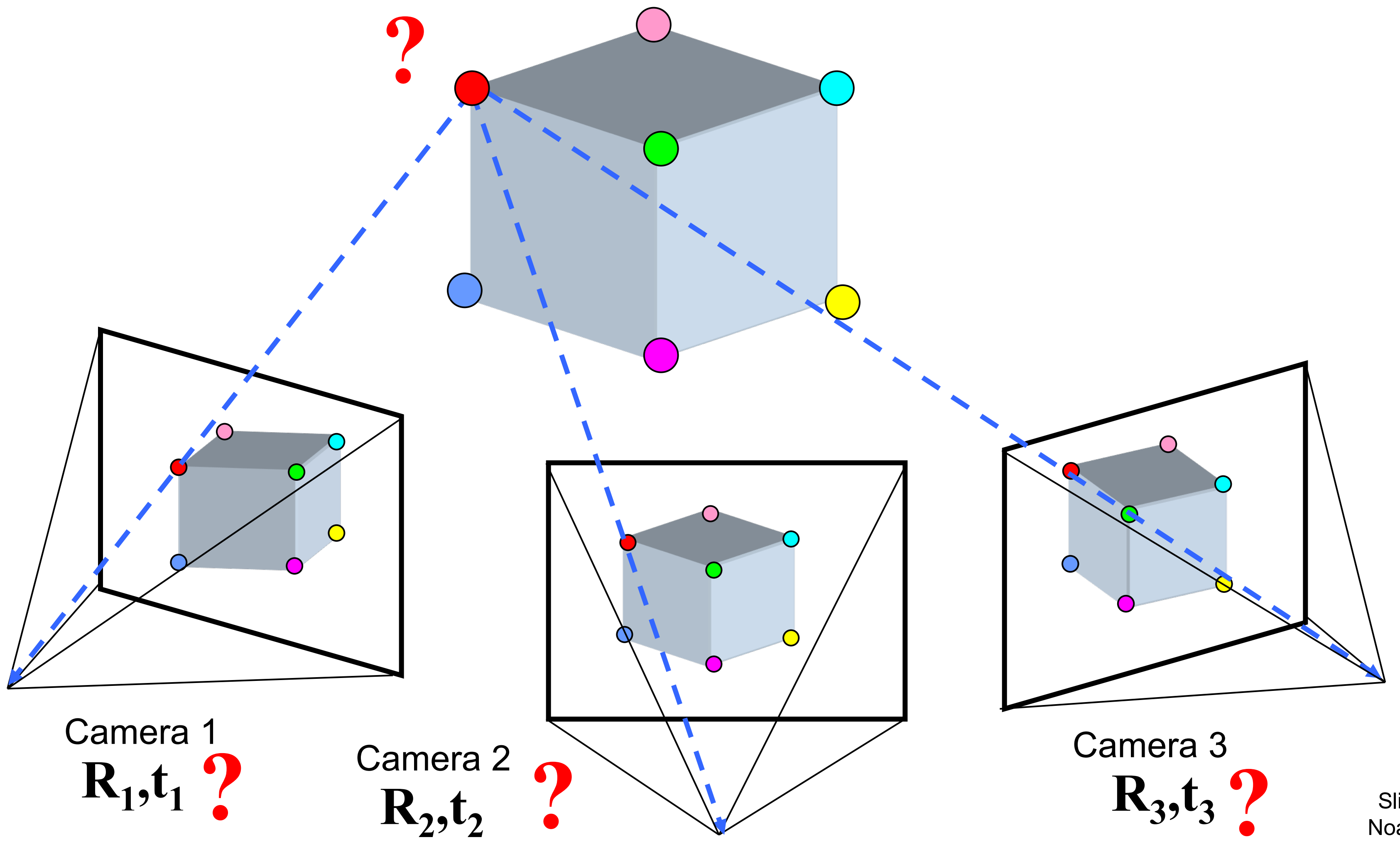
- **Camera 'Motion':** Given a set of corresponding 2D/3D points in two or more images, compute the camera parameters.



Slide credit: Noah Snavely

Multi-view geometry problems

- **Structure from Motion:** Given projections of the same 3D point in two or more images, compute the 3D coordinates of that point



Slide credit: Noah Snavely

Geometric camera calibration

Given a set of matched points

$$\{\mathbf{X}_i, \boldsymbol{x}_i\}$$

point in 3D space

point in the image

and camera model

$$\boldsymbol{x} = \boldsymbol{f}(\mathbf{X}; \boldsymbol{p}) = \mathbf{P}\mathbf{X}$$

projection model

parameters

Camera matrix

Find the (pose) estimate of

$$\mathbf{P}$$

Compute SVD of a measurement matrix to obtain P

$$\begin{aligned}\mathbf{P} &= \mathbf{K}[\mathbf{R}|\mathbf{t}] \\ &= \mathbf{K}[\mathbf{R}| - \mathbf{R}\mathbf{c}] \\ &= [\mathbf{M}| - \mathbf{M}\mathbf{c}]\end{aligned}$$

Find the camera center **C**

$$\mathbf{P}\mathbf{c} = \mathbf{0}$$

SVD of P!

***c** is the singular vector corresponding to the smallest singular value*

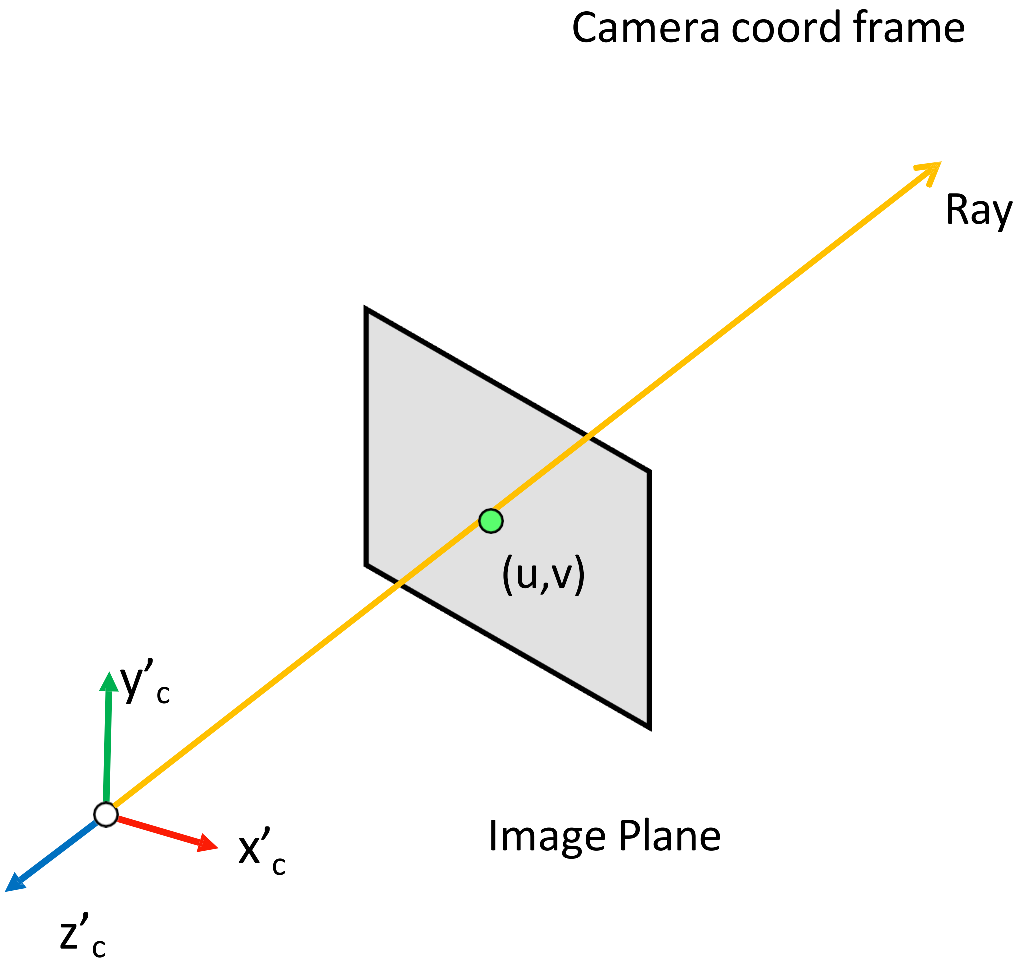
Find intrinsic **K** and rotation **R**

$$\mathbf{M} = \mathbf{K}\mathbf{R}$$

QR decomposition

Now that our cameras are calibrated, can we find the 3D scene point of a pixel?

You know we can't, but we know it'll be on the ray!



3D to 2D:
(point)

$$u = f_x \frac{x_c}{z_c} + o_x$$

$$v = f_y \frac{y_c}{z_c} + o_y$$

2D to 3D:
(ray)
Back projection

$$x = \frac{z}{f_x}(u - o_x)$$

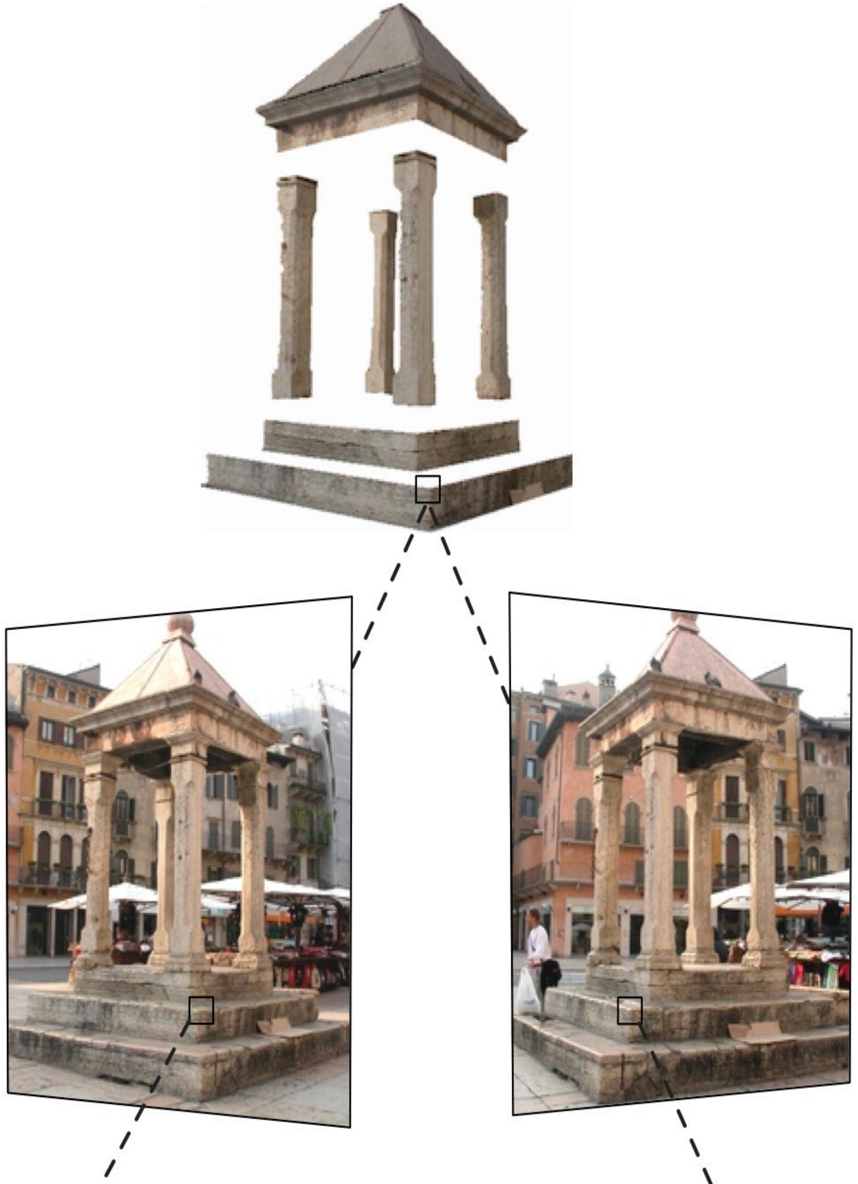
$$y = \frac{z}{f_y}(v - o_y)$$

$$z > 0$$

Slide credit: Shree Nayar

The goal

Develop theories and study how a 3D point and its projection in 2 images are related to each other!



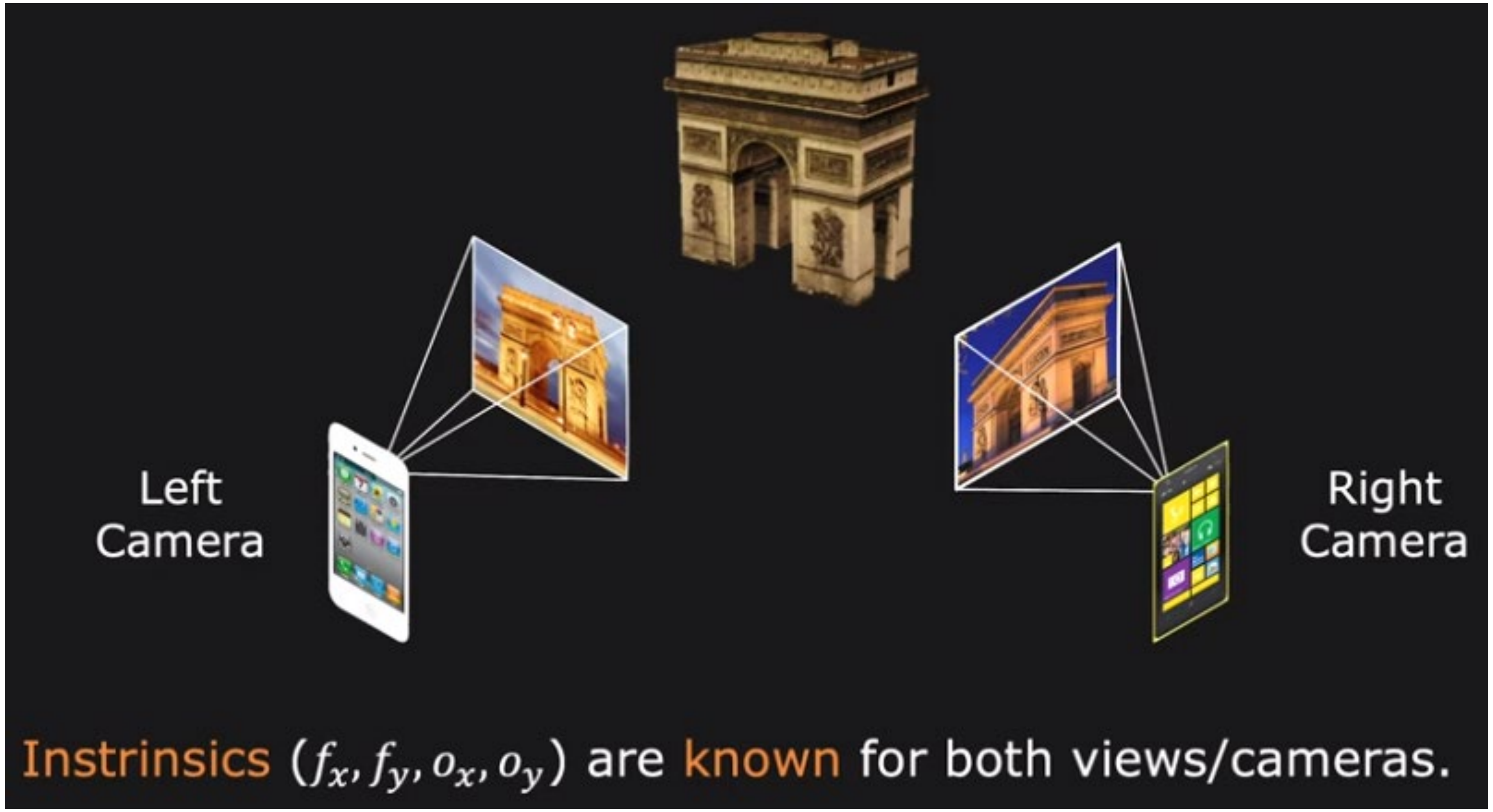
From a single image you can only back project a pixel to obtain a ray on which the actual 3D point lies

To find the actual location of the 3D point, you need:

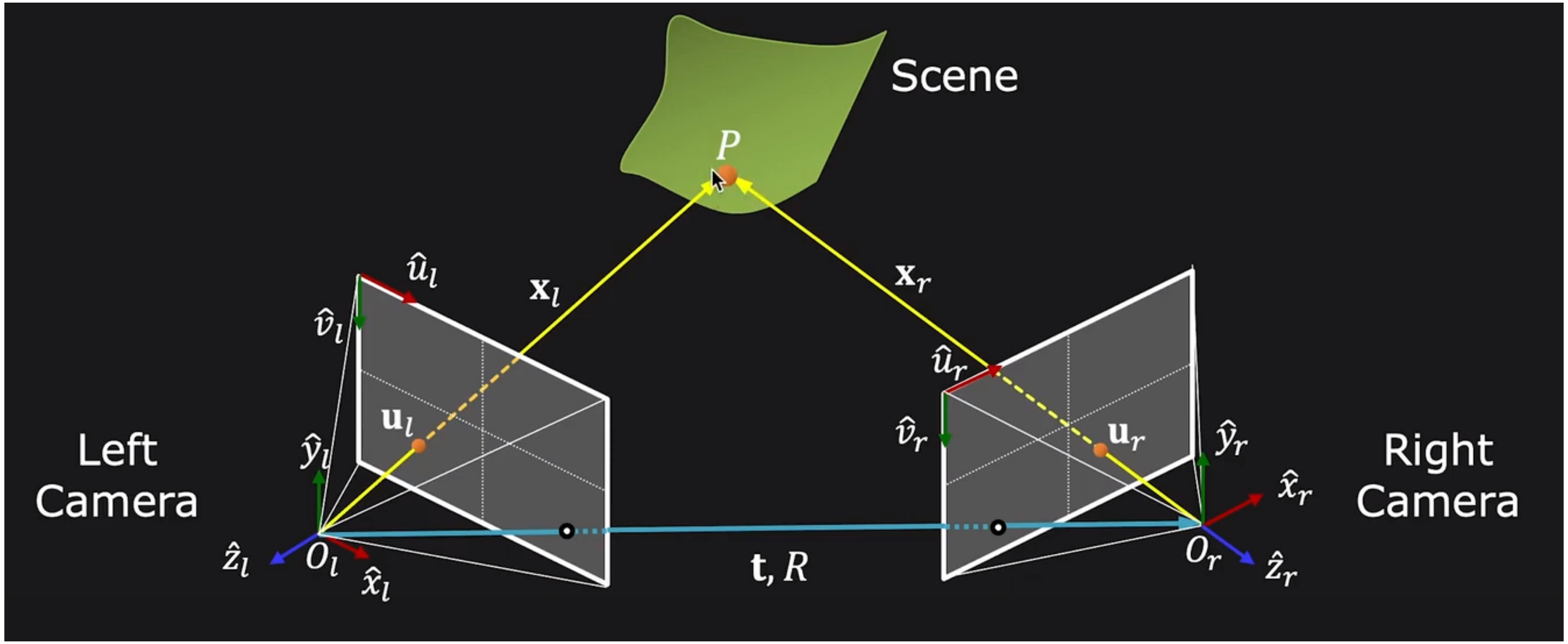
an additional image captured from another viewpoint.

Uncalibrated Stereo

Compute 3D structure of a static scene from two arbitrary views



Uncalibrated Stereo



- Assume Camera Matrix K is known for each camera
- Find a few reliable corresponding points
- Find relative camera position t and orientation R
- Find dense correspondance
- Compute depth using Triangulation

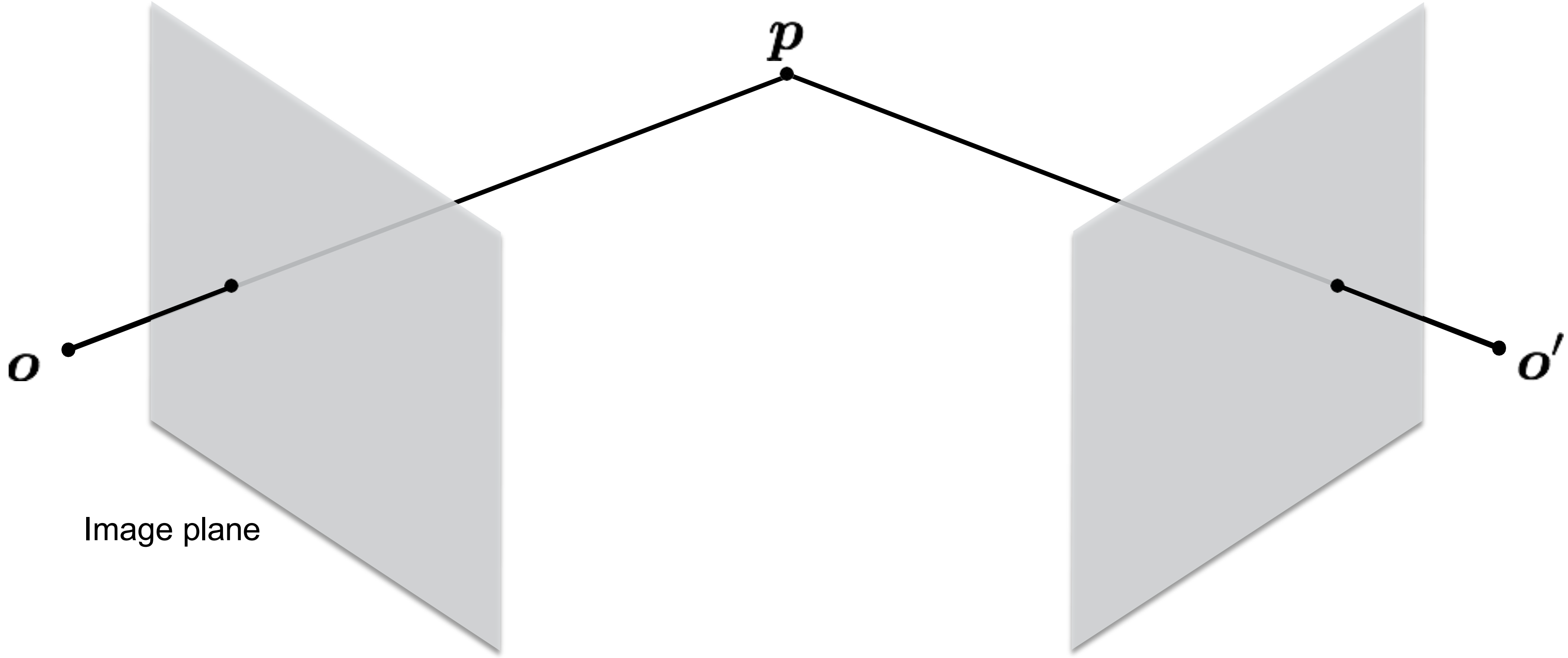
Today’s class

- Epipolar Geometry
- Essential Matrix
- Fundamental Matrix
- 8-point Algorithm
- Triangulation

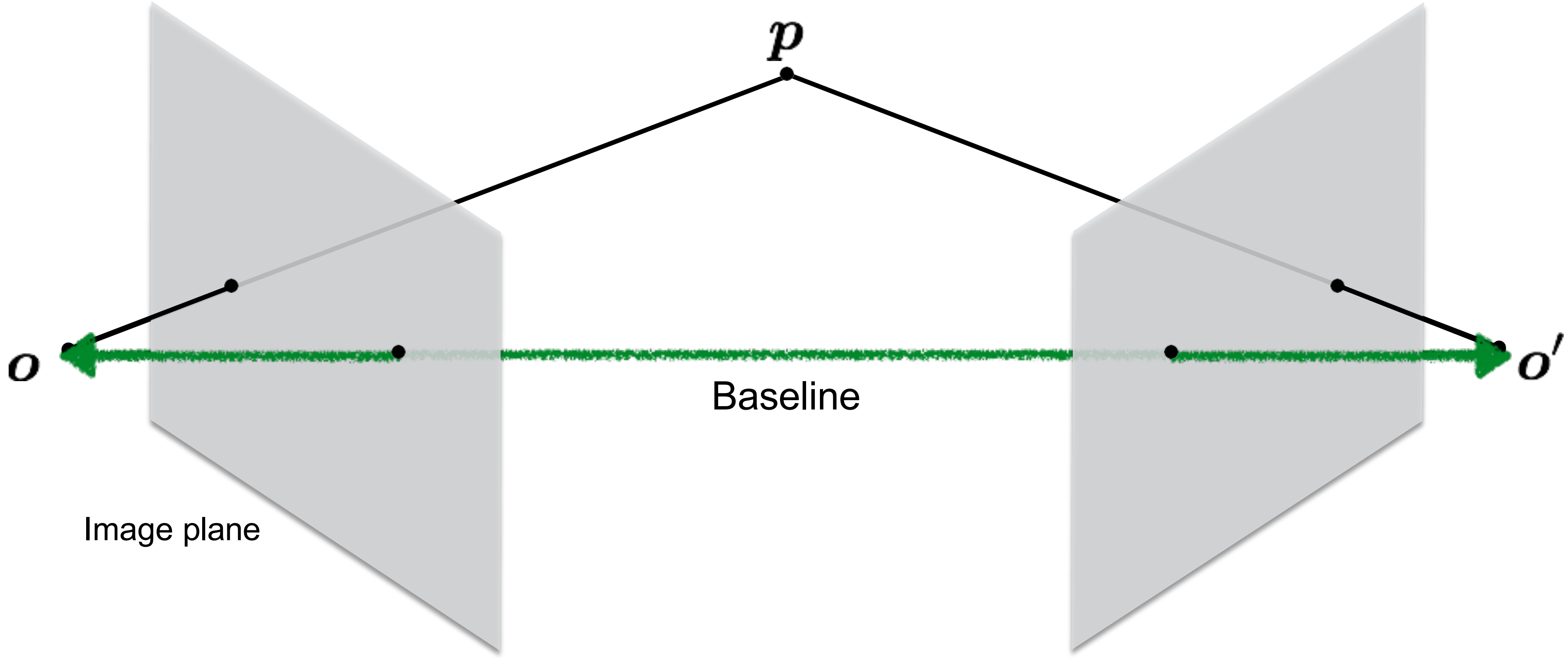
Today’s class

- Epipolar Geometry (few definitions)
- Essential Matrix
- Fundamental Matrix
- 8-point Algorithm
- Triangulation

Epipolar geometry

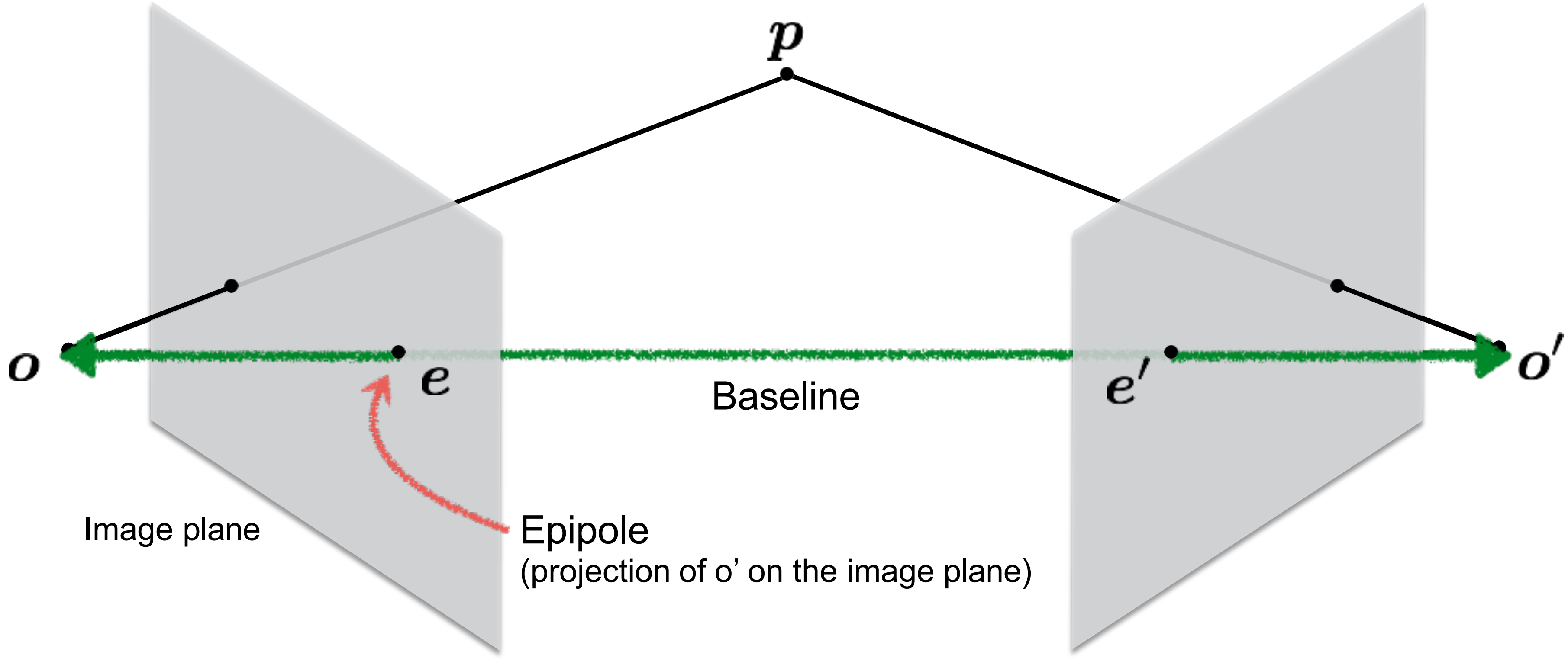


Epipolar geometry

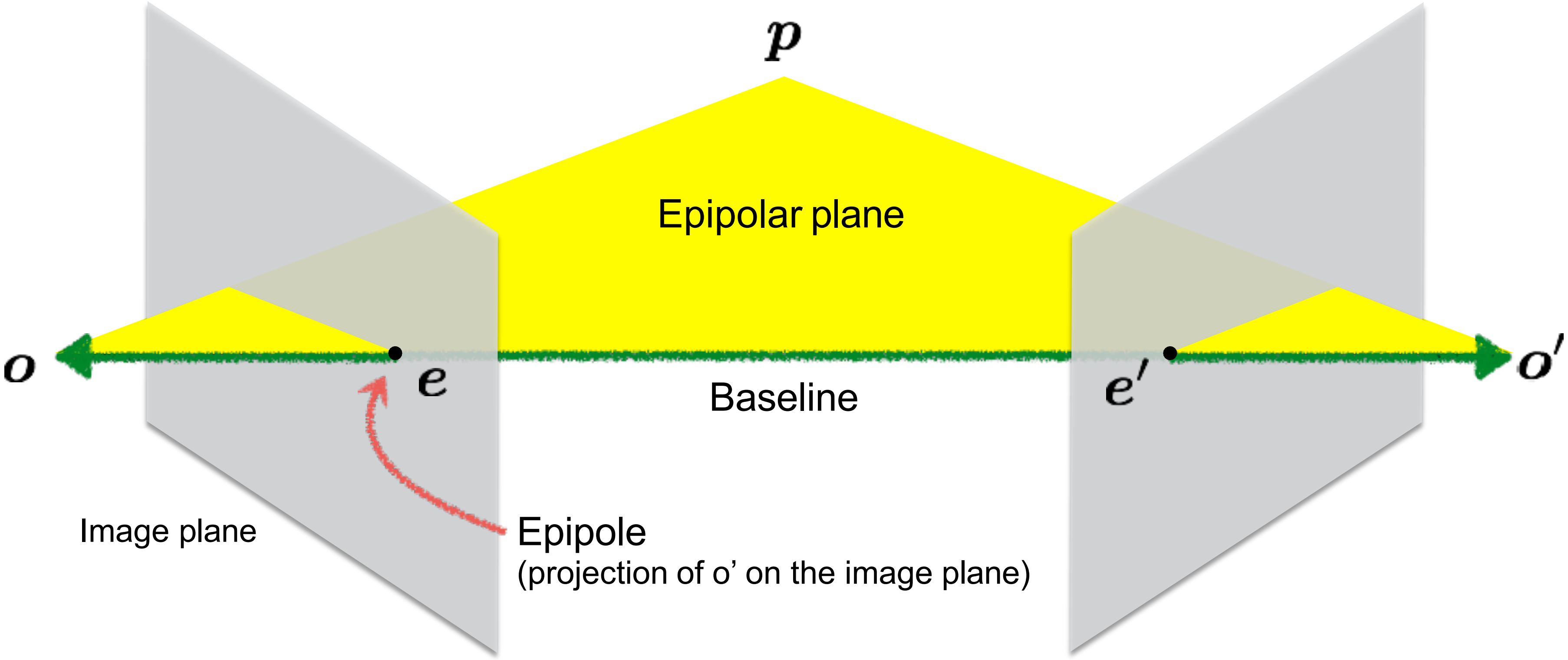


Baseline refers to the **distance between the centers of projection (optical centers)** of the **two cameras** in a stereo setup — usually the left and right cameras.

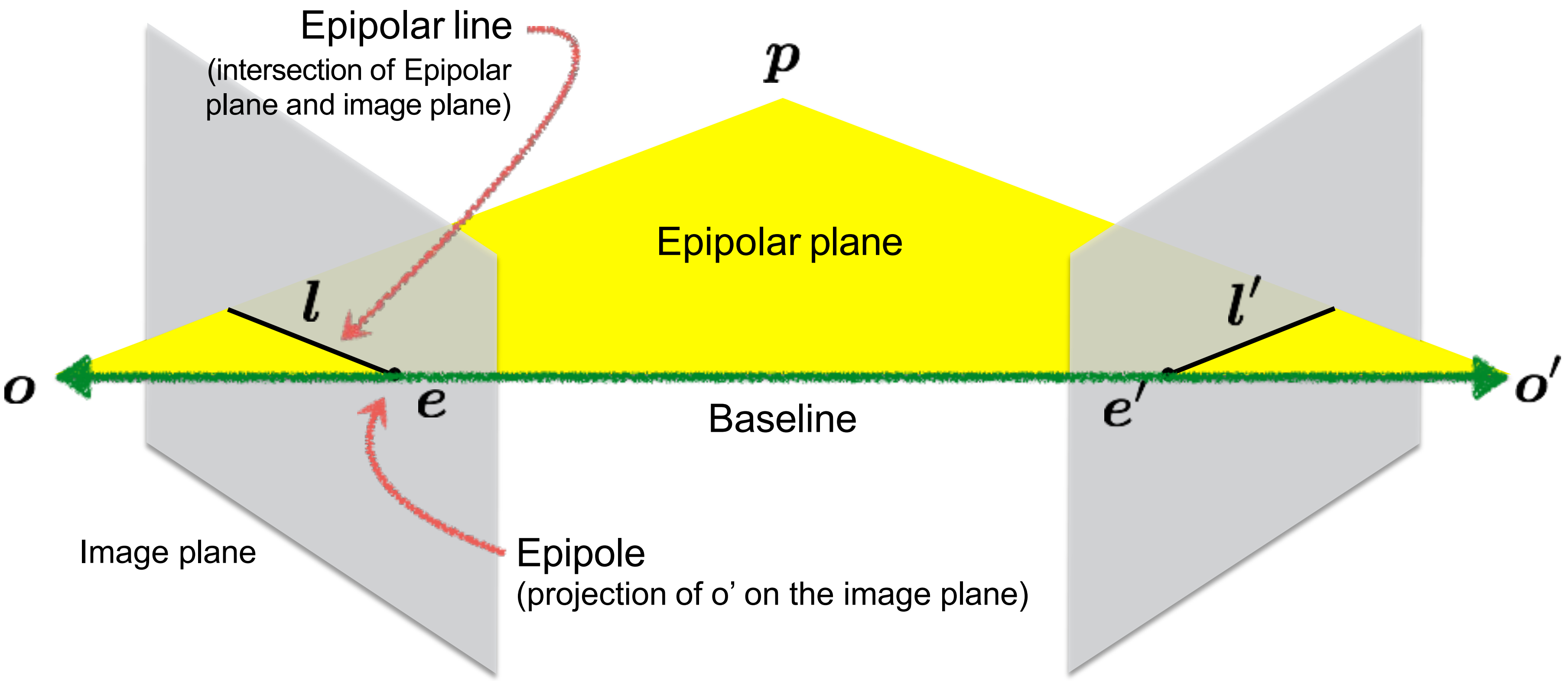
Epipolar geometry



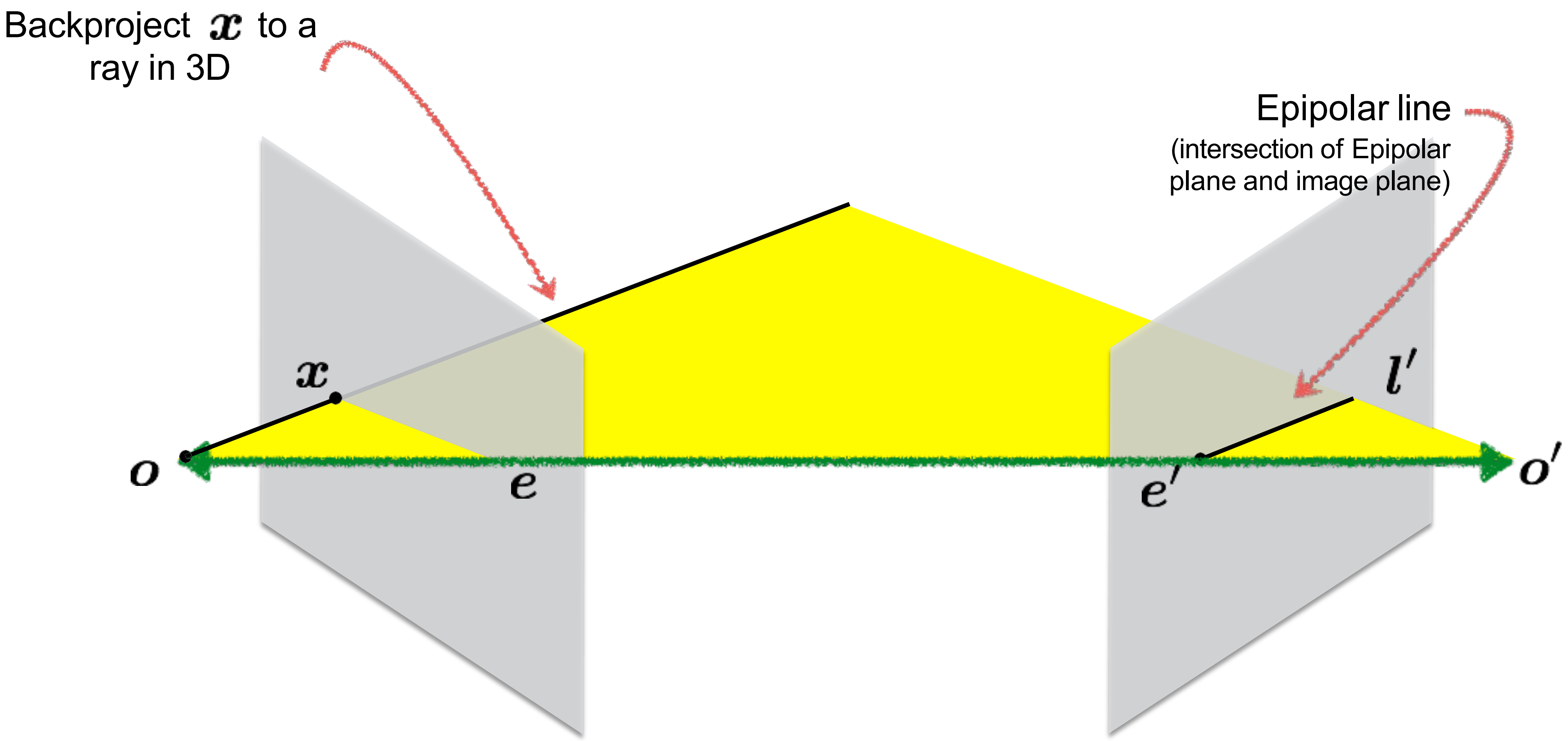
Epipolar geometry



Epipolar geometry

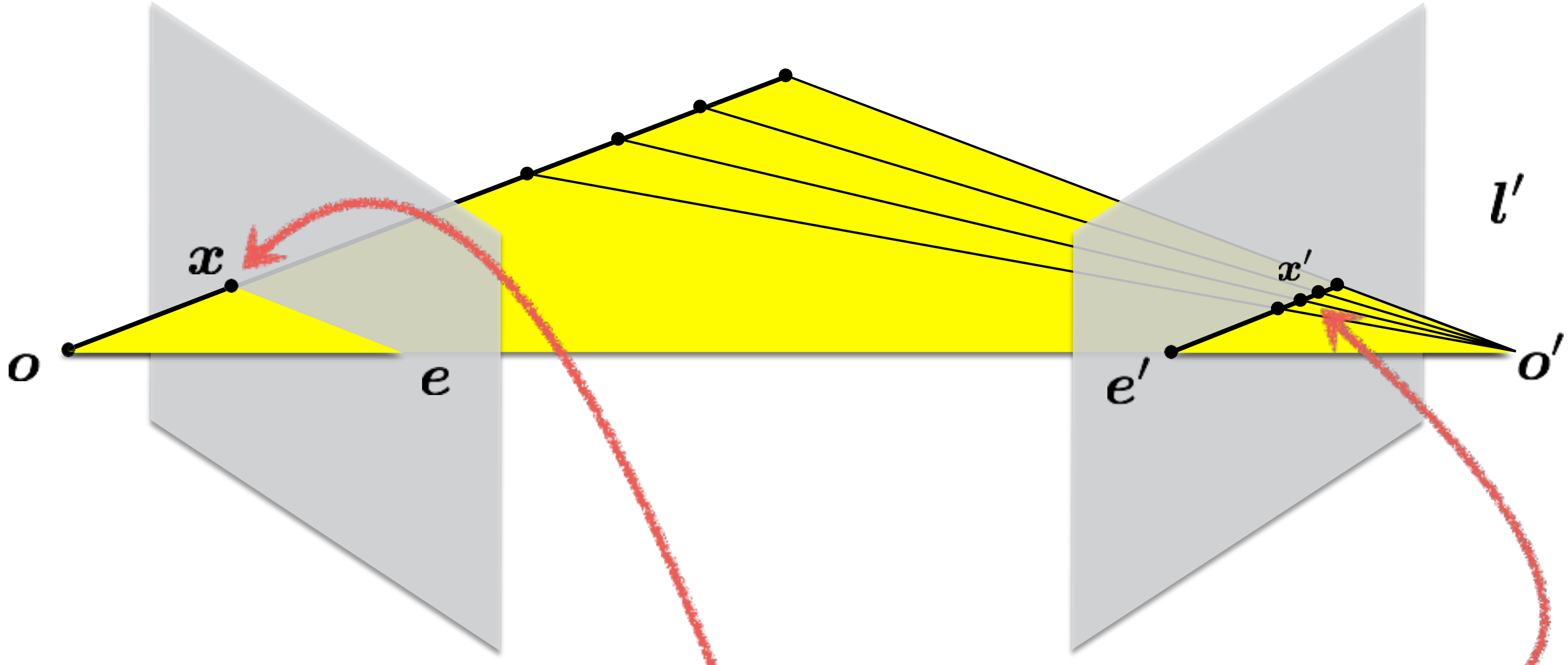


Epipolar constraint



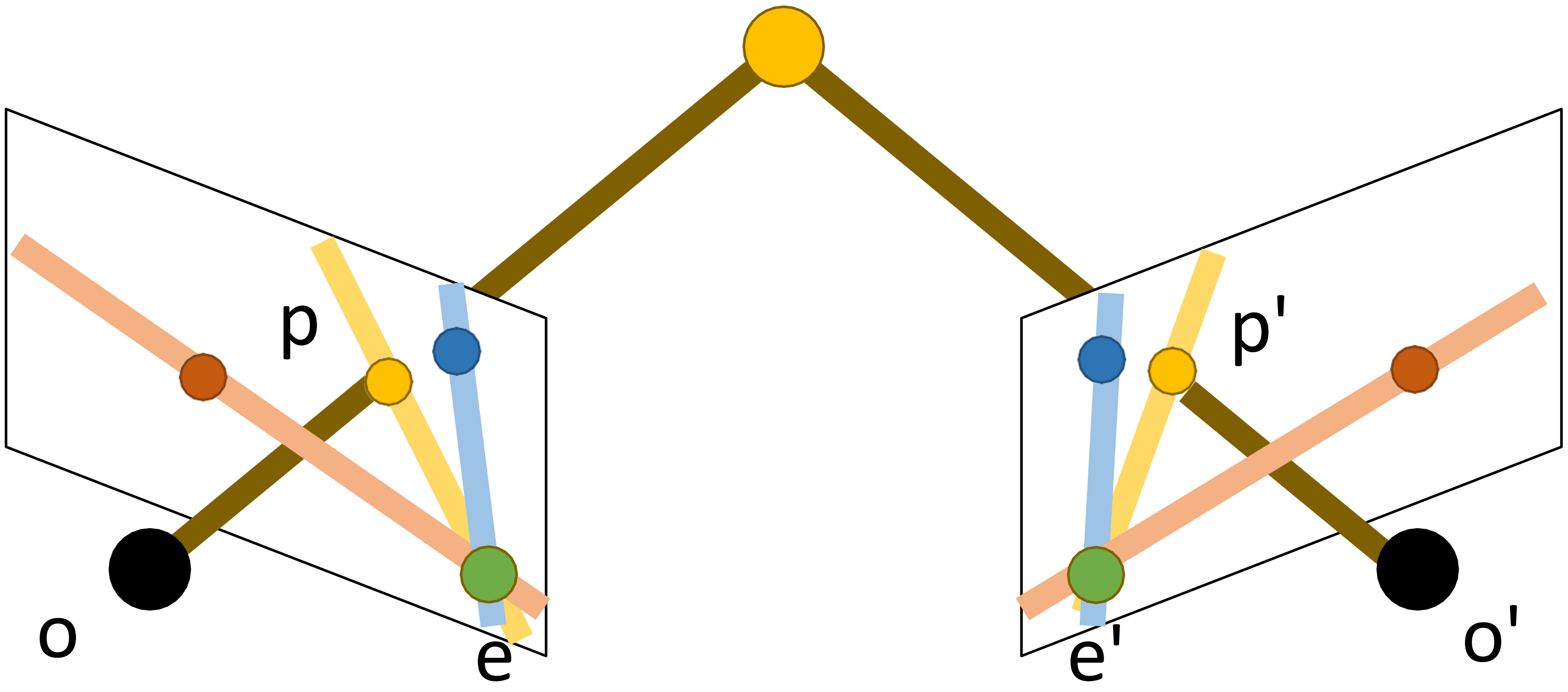
Another way to construct the epipolar plane, this time given $\boldsymbol{x}$

Epipolar constraint



Potential matches for $\boldsymbol{x}$ lie on the epipolar line $\boldsymbol{l}'$

Example: Converging Cameras



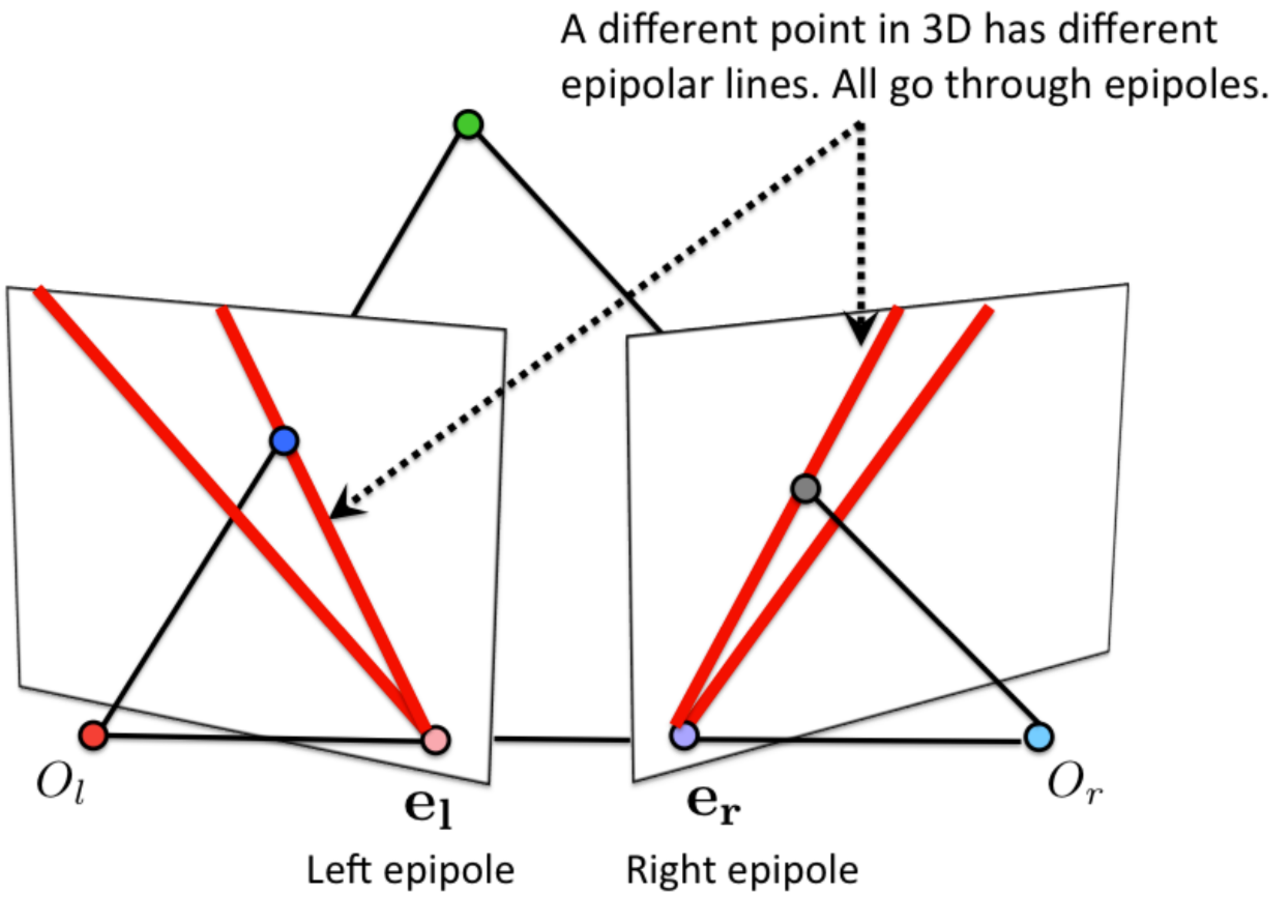
Epipoles finite, maybe in image; epipolar lines converge

Epipolar lines come in pairs:

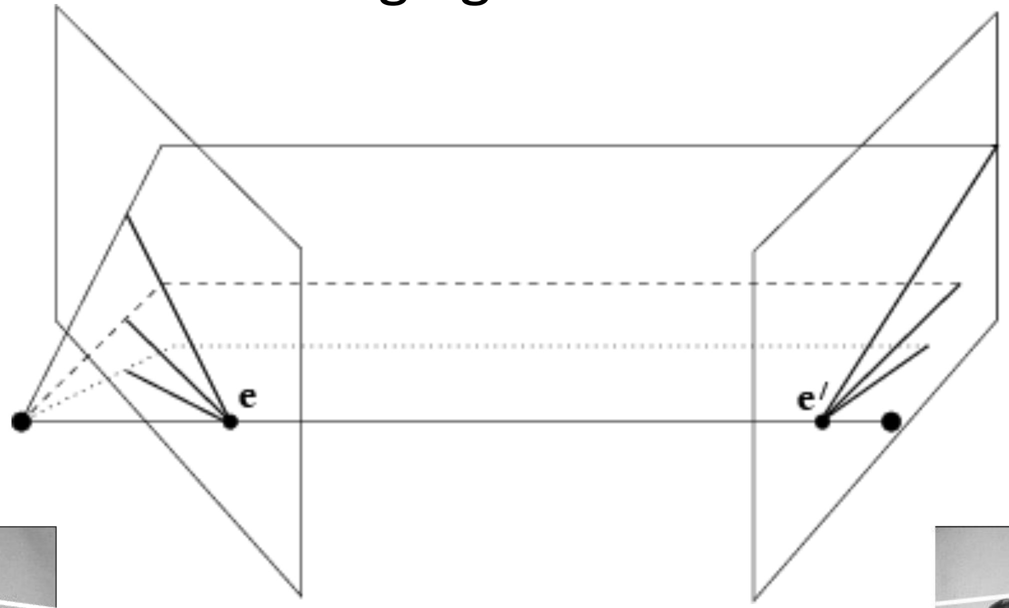
given a point p, we can construct the epipolar line for p’.

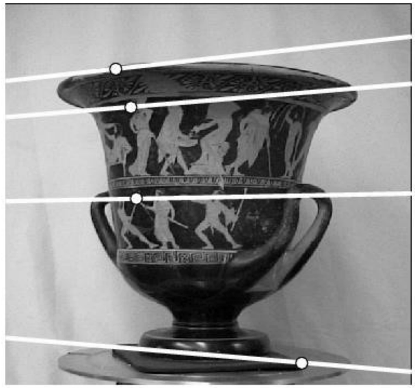
Slide credit: David Fouhey

Obviously a different point in 3D will form a different epipolar plane and therefore different epipolar lines. But these epipolar lines go through epipoles as well.



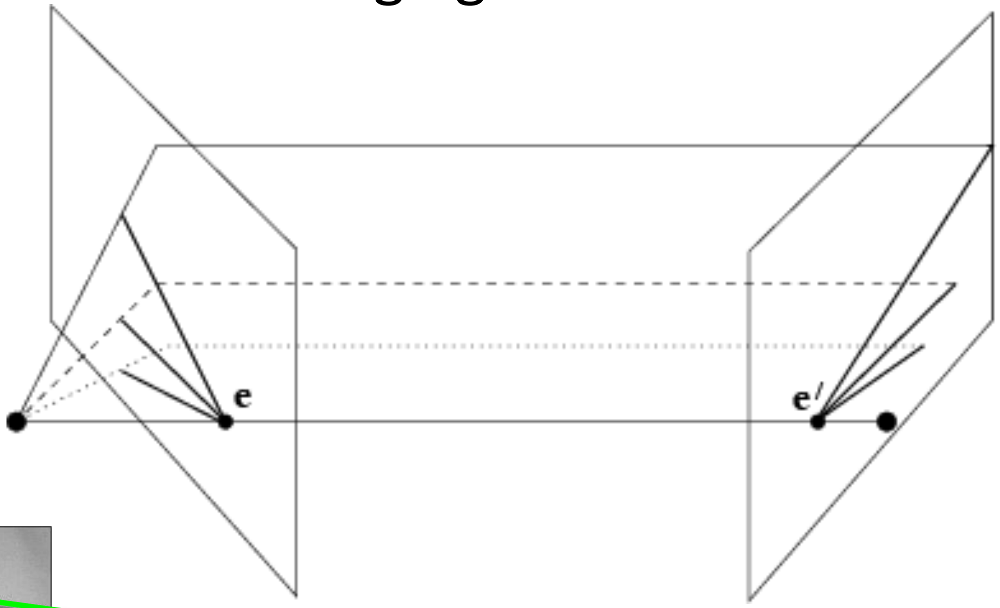
Converging cameras

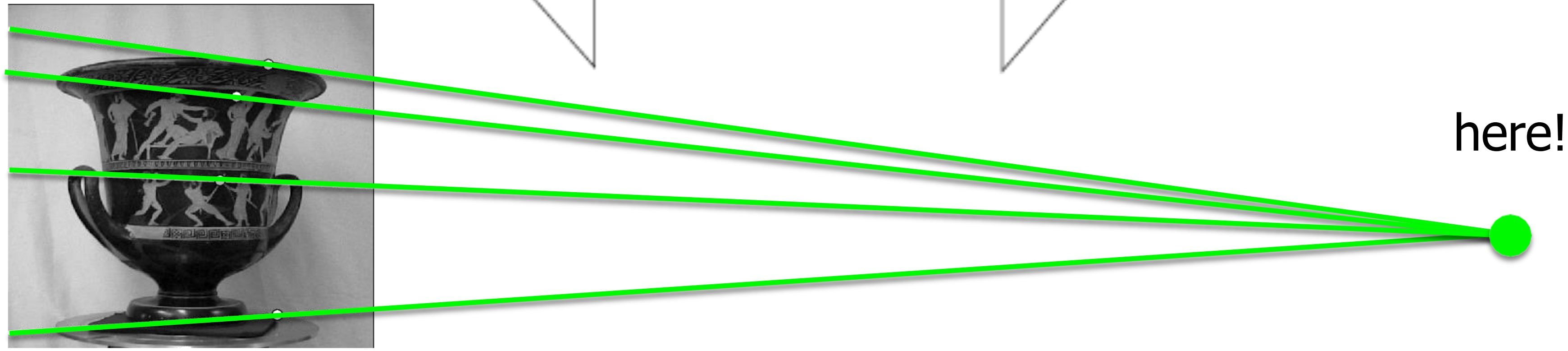




Where is the epipole in this image?

Converging cameras

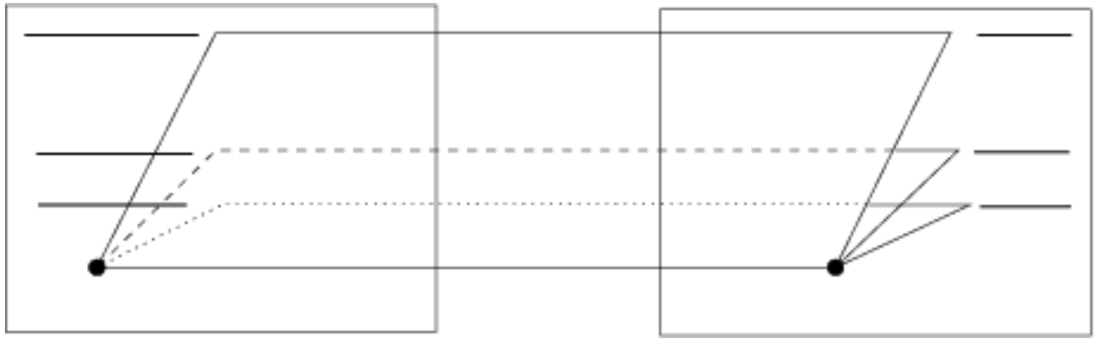


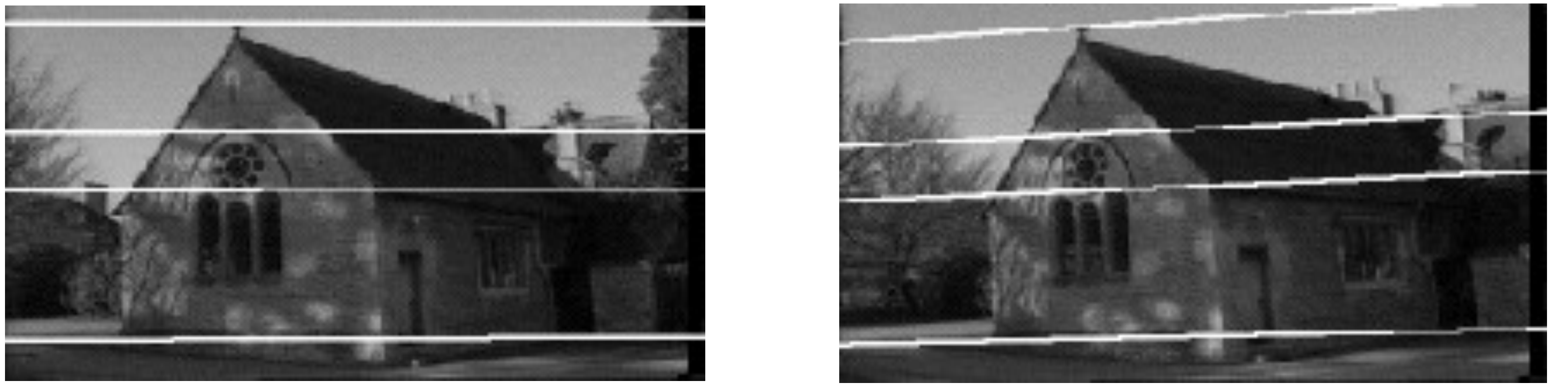


Where is the epipole in this image?

It's not always in the image

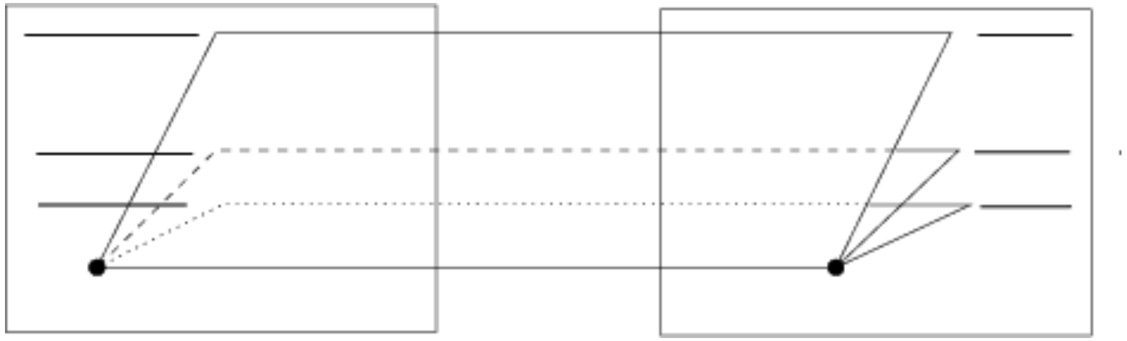
Parallel cameras

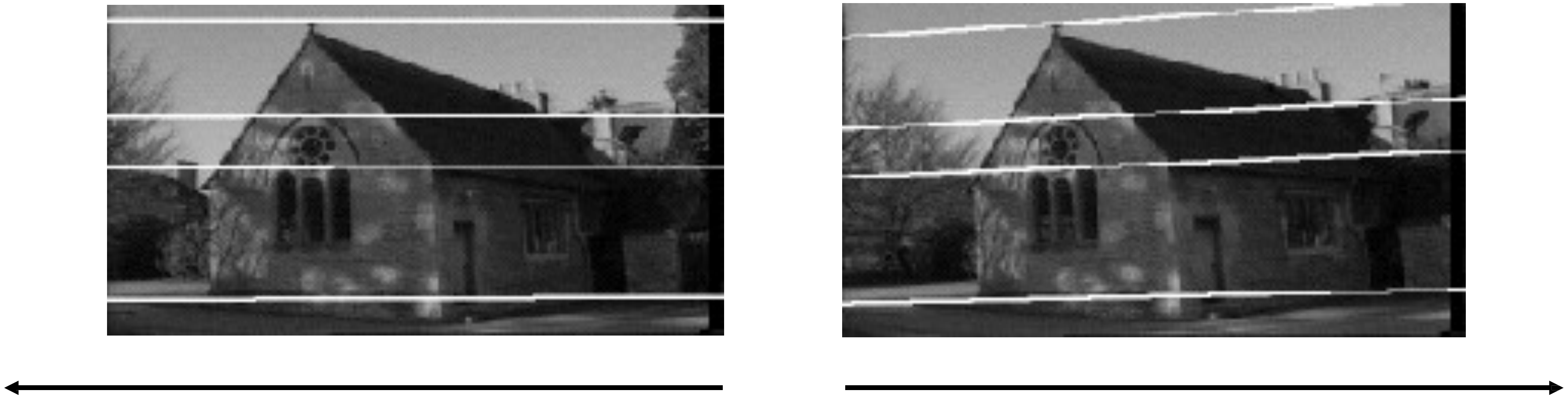




Where is the epipole?

Parallel cameras

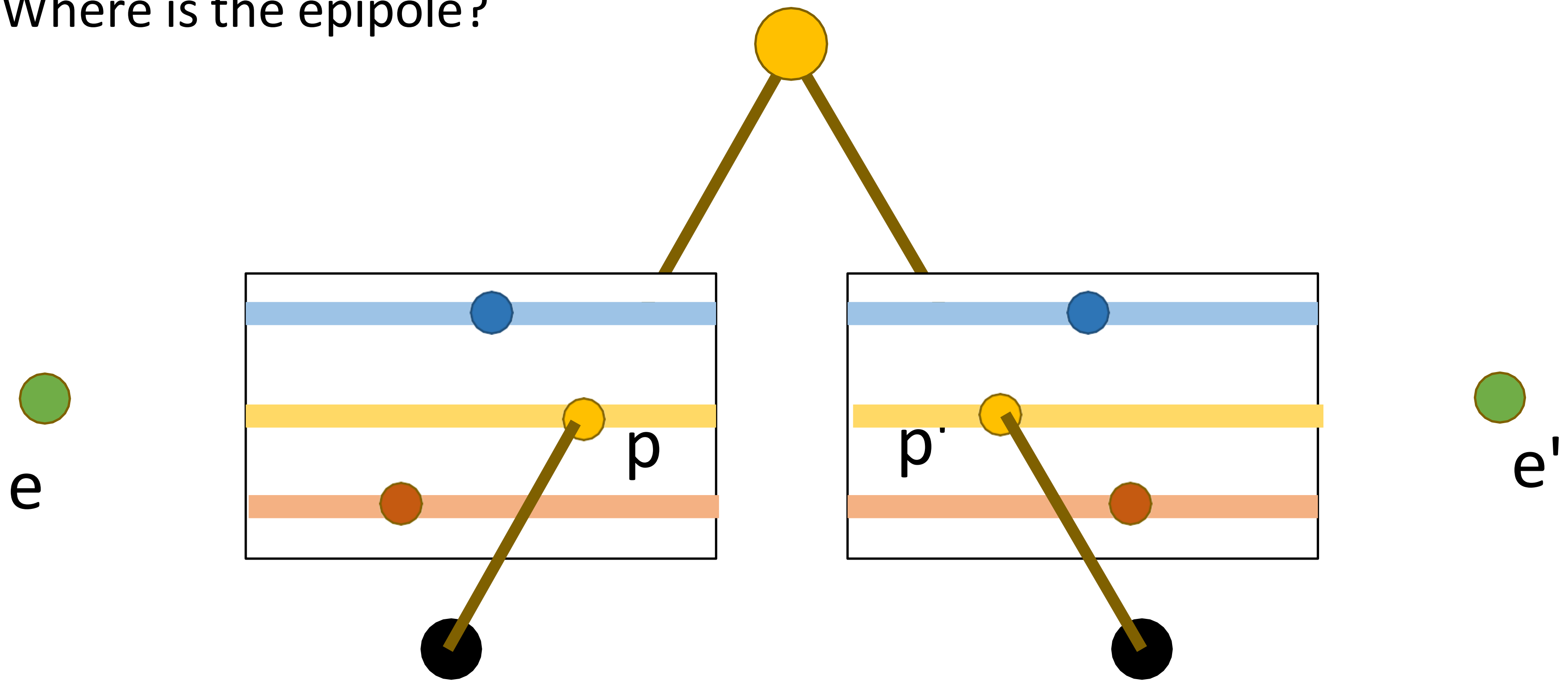




epipole at infinity

Example: Parallel to Image Plane

Where is the epipole?



Epipoles *infinitely* far away, epipolar lines parallel

Slide credit: David Fouhey

Example: Forward Motion

Image Credit: Hartley & Zisserman

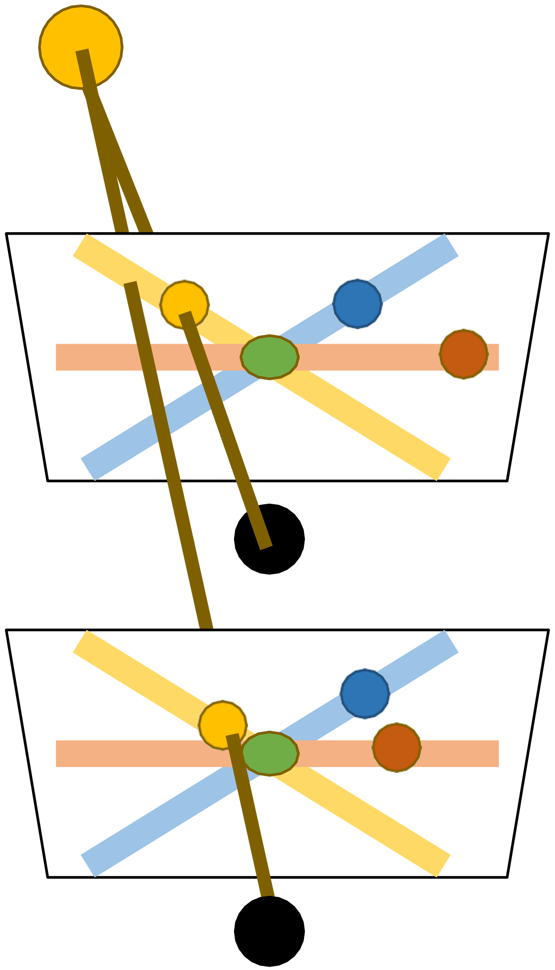
Example: Forward Motion

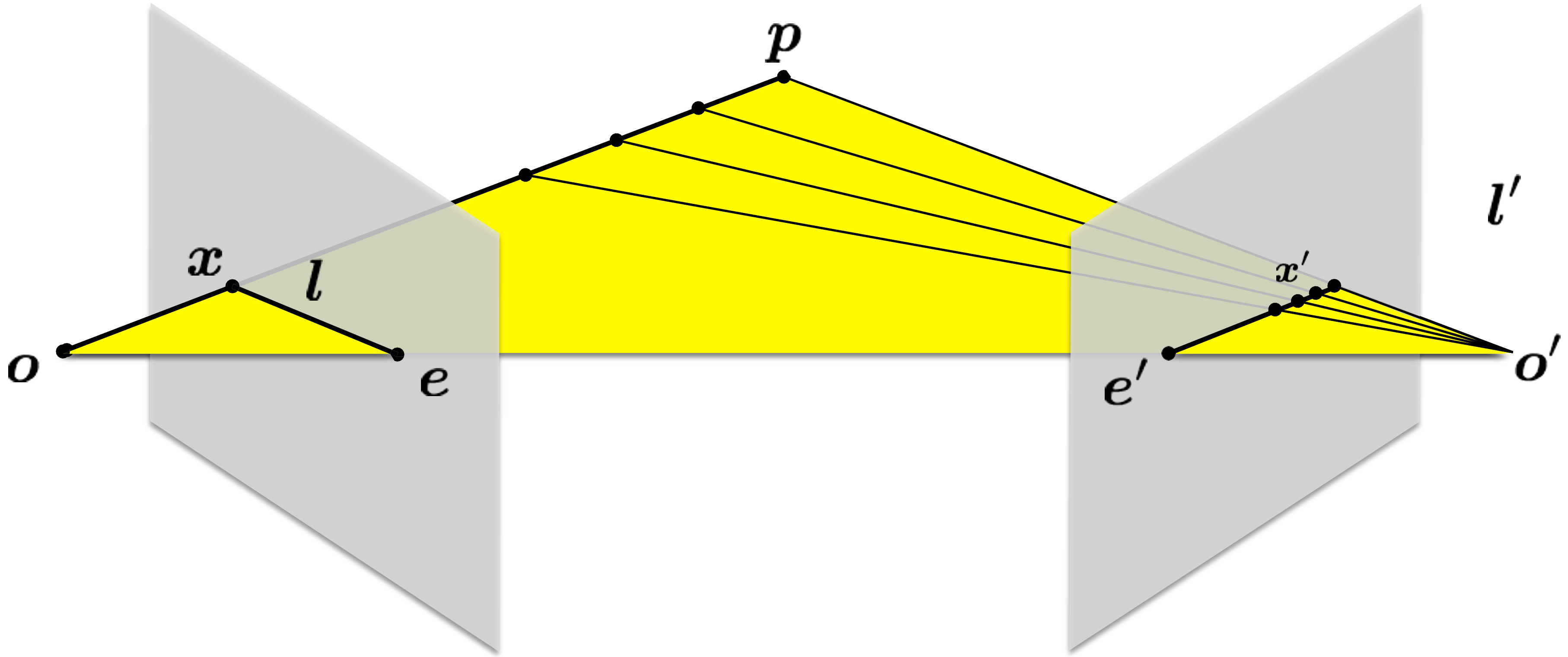
Image Credit: Hartley & Zisserman

Example: Forward Motion

Epipole is focus of expansion / principal point of the camera.

Epipolar lines go out from principal point





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

The baseline connects the ____________and _____________

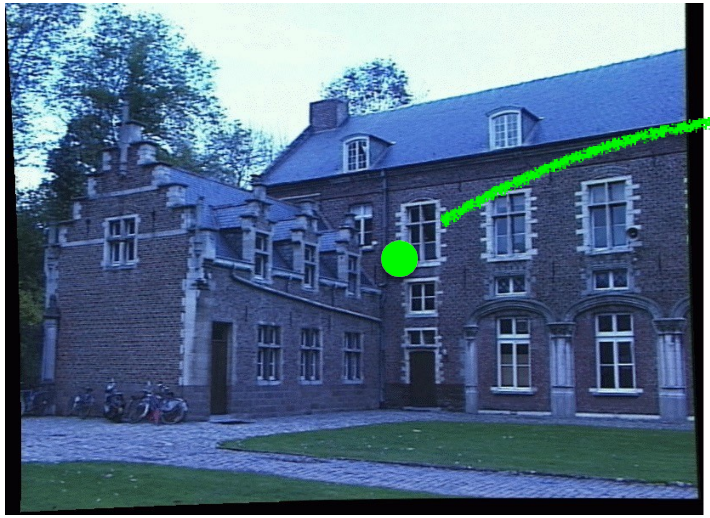
An epipolar line (left image) maps to a ___________in the right image

An epipole **e** is a projection of the ________________on the image plane

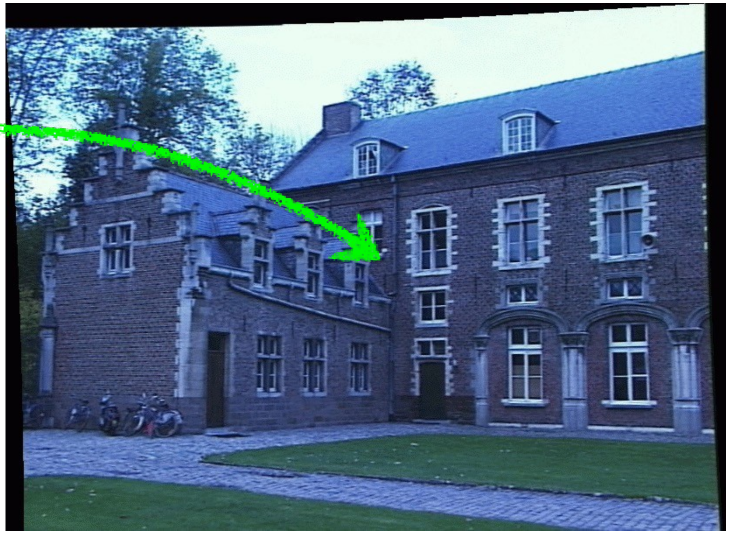
All epipolar lines in an image intersect at the ________________

The epipolar constraint is an important concept for stereo vision

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



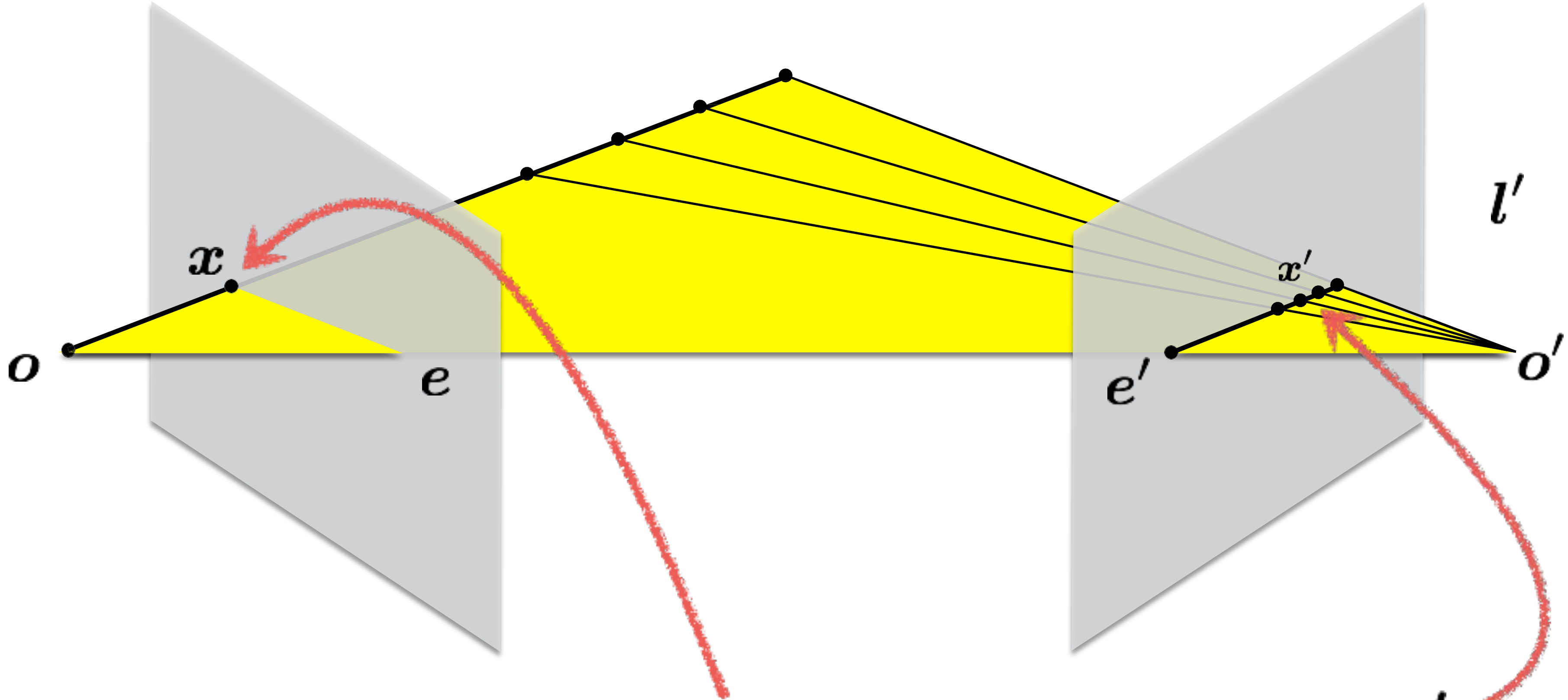
Left image



Right image

How would you do it?

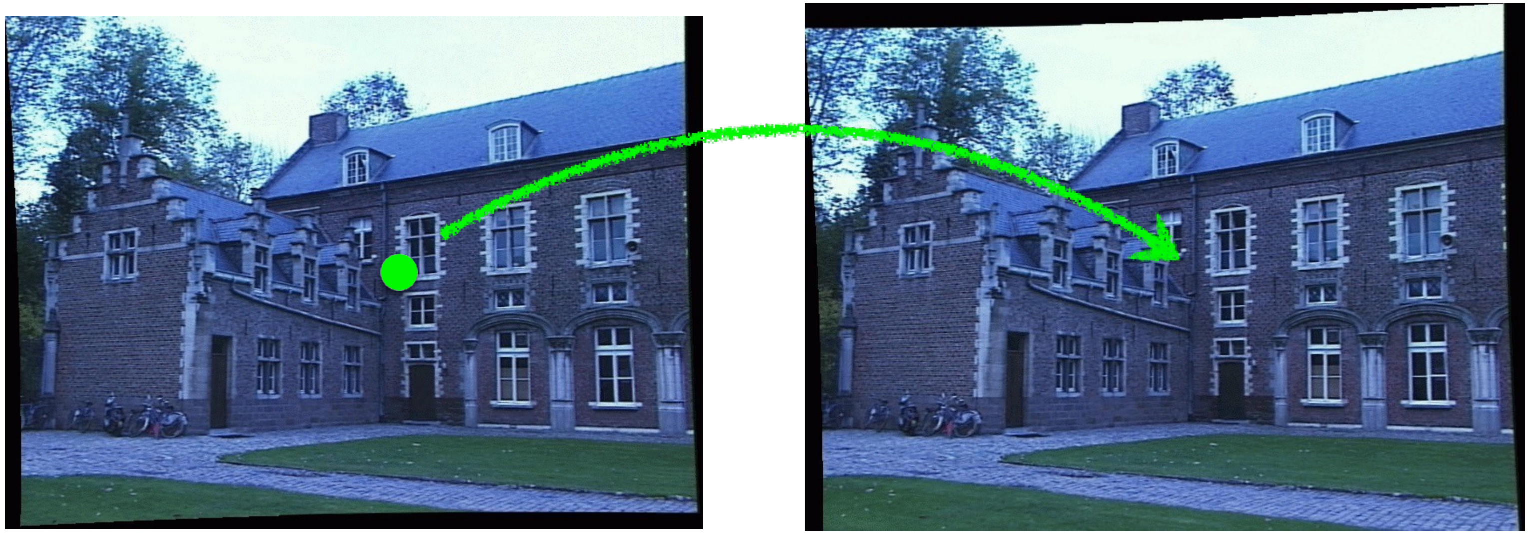
Epipolar constraint



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

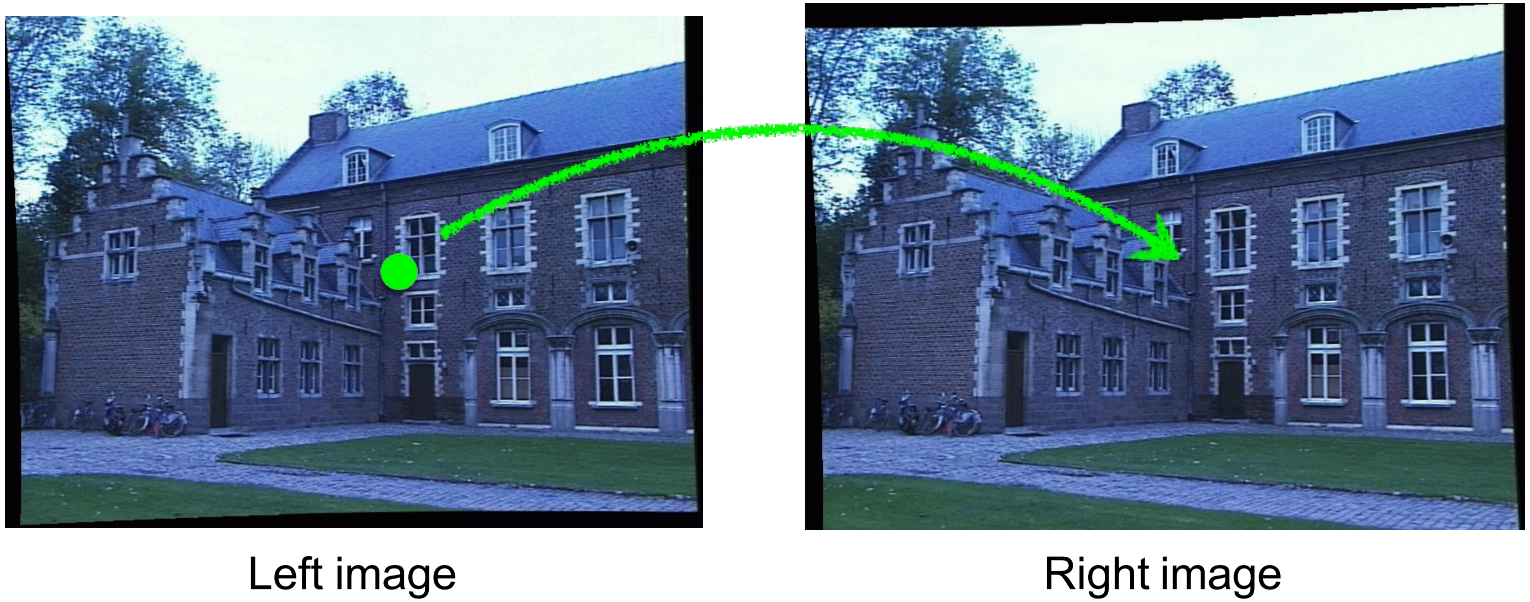
Right image

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



Want to avoid search over entire image

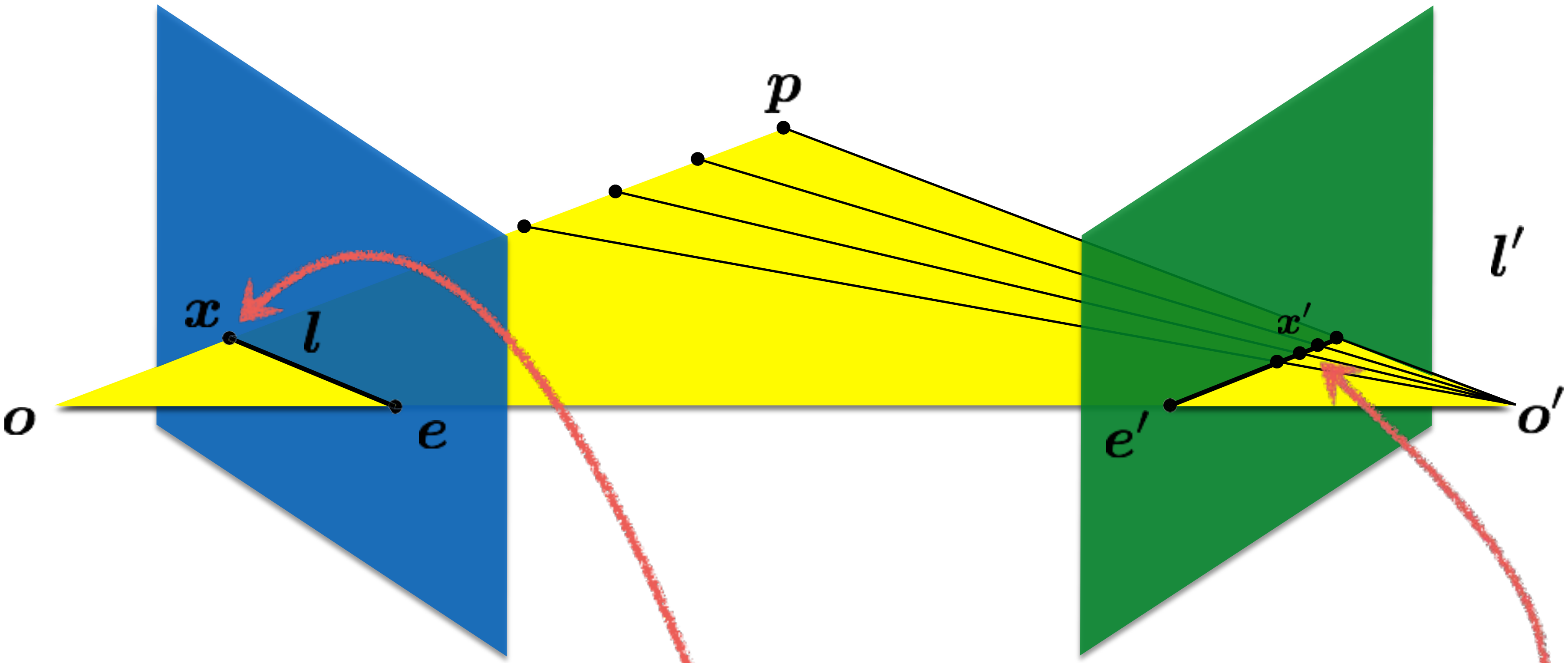
Epipolar constraint reduces search to a single line

How do you compute the epipolar line?

Today’s class

- Epipolar Geometry
- Essential Matrix
- Fundamental Matrix
- 8-point Algorithm
- Triangulation

Recall:Epipolar constraint

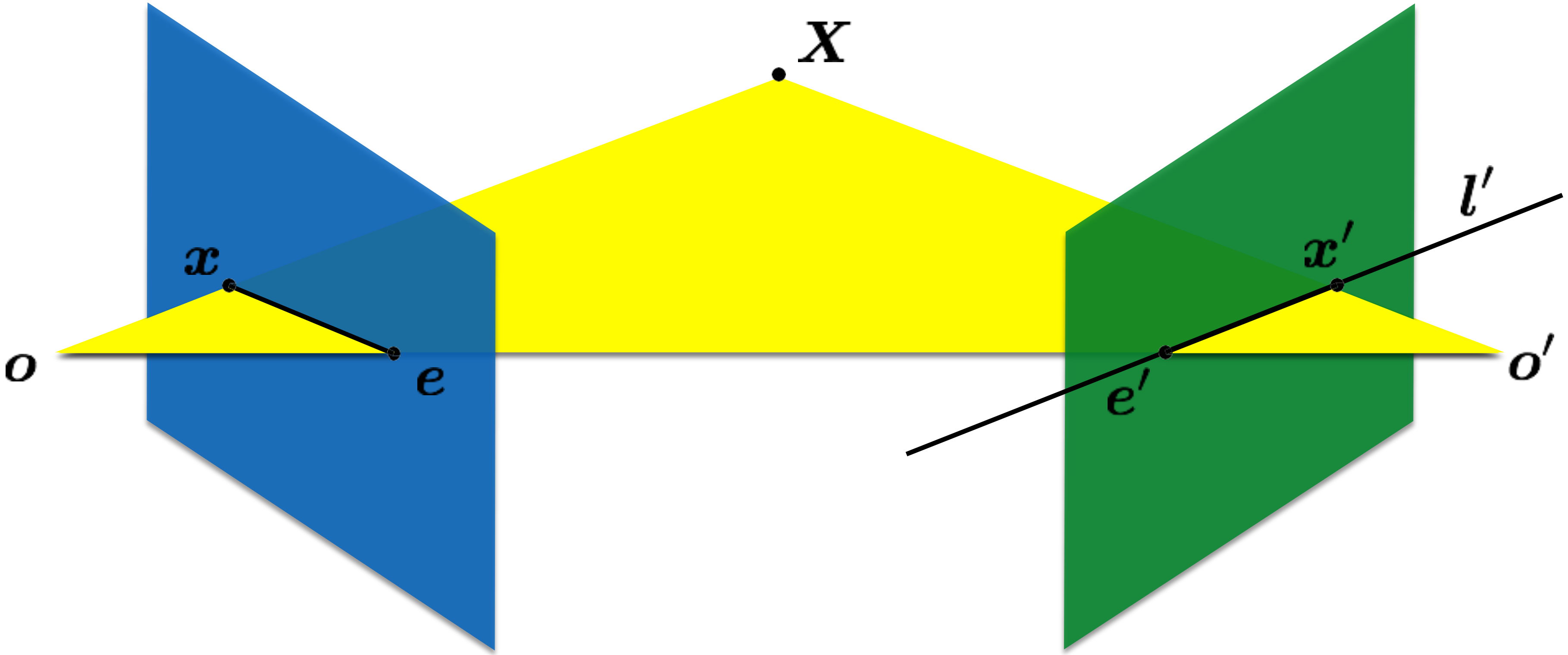


Potential matches for x lie on the epipolar line l'

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

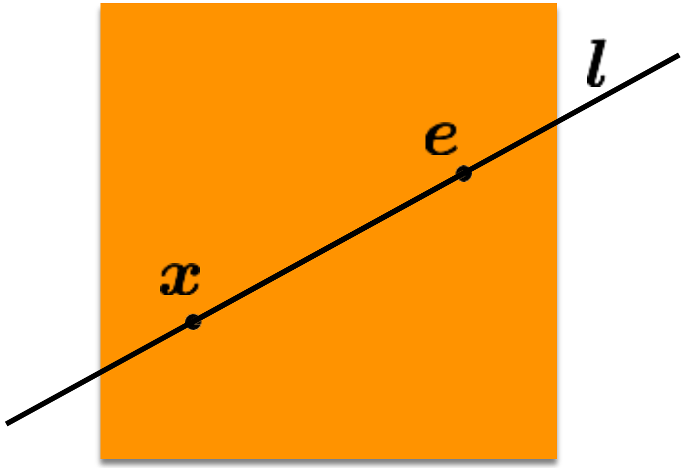
$$\boxed{\mathbf{E}\boldsymbol{x} = \boldsymbol{l}'}$$

Essential matrix is 3x3 and encodes epipolar geometry.



Epipolar Line

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

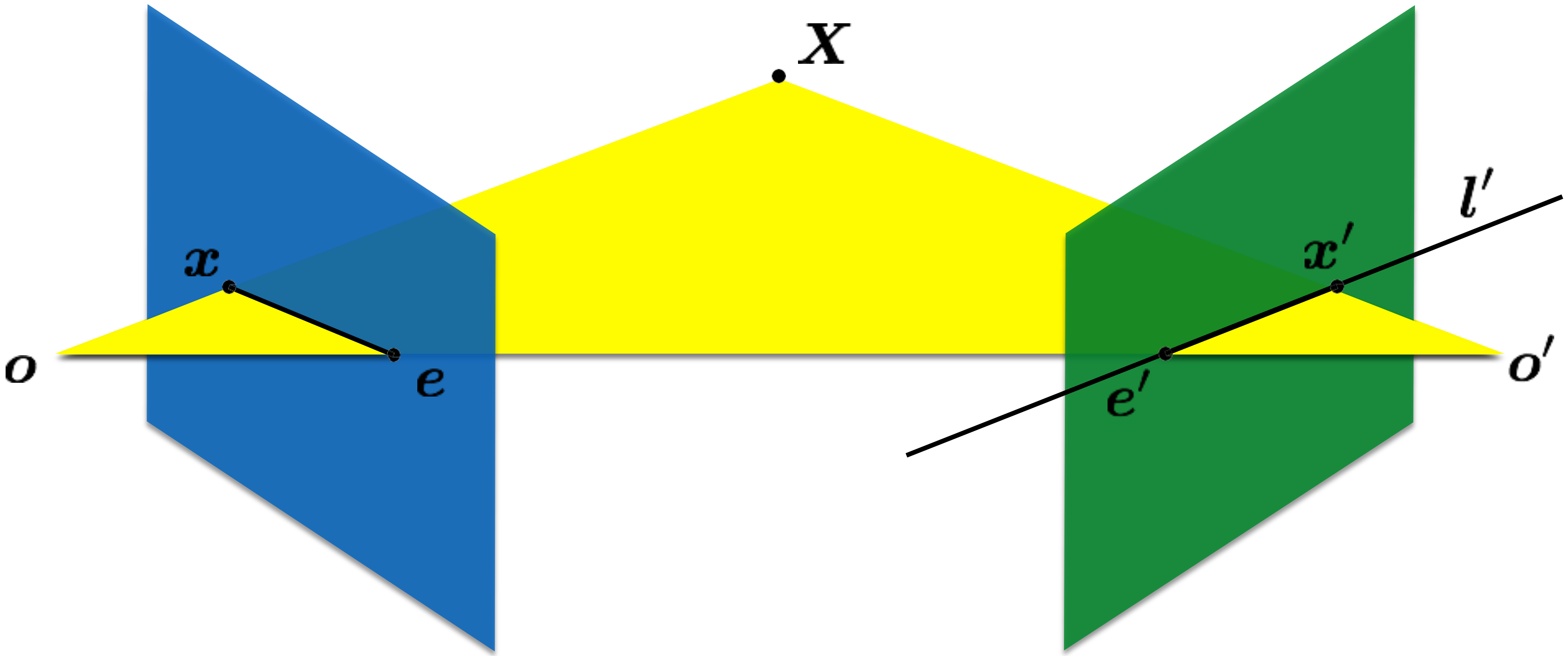


If the point $\boldsymbol{x}$ is on the epipolar line $\boldsymbol{l}$ then

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

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

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



Where does the essential matrix come from?

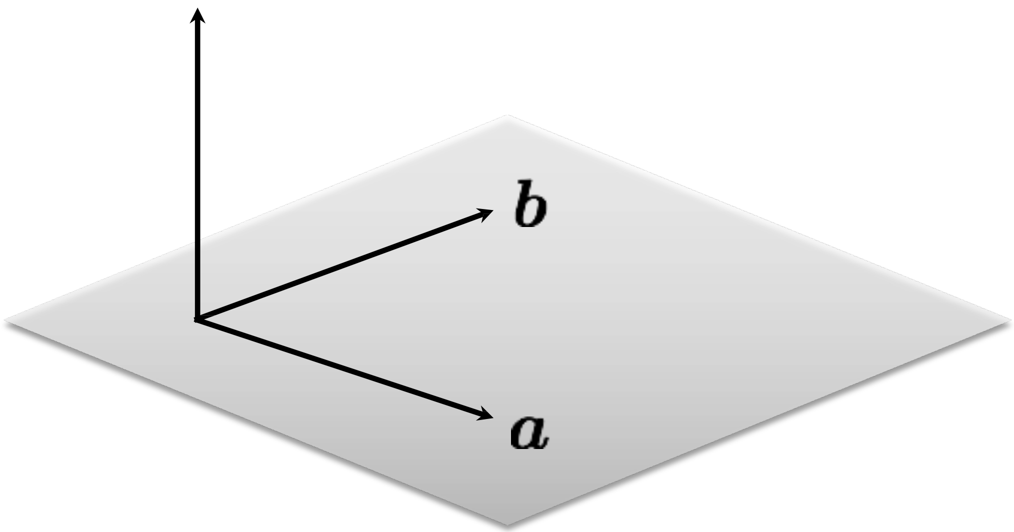
Can we express essential matrix as function of camera parameters?

Linear algebra reminder: cross product

Vector (cross) product

takes two vectors and returns a vector <u>perpendicular</u> to both

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



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

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

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

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

remember this!!!

Linear algebra reminder: cross product

Cross product

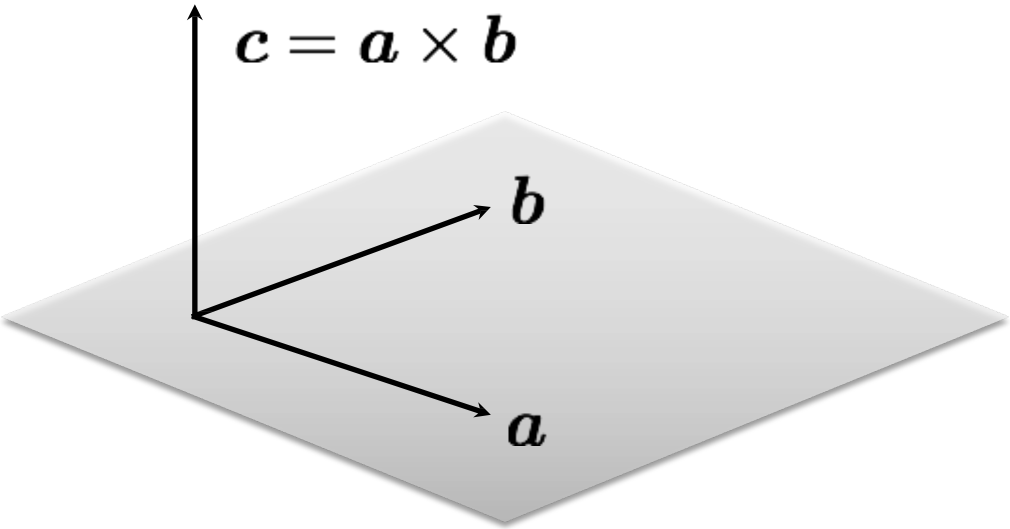
$$\boldsymbol{a} \times \boldsymbol{b} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}$$

Can also be written as a matrix multiplication

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

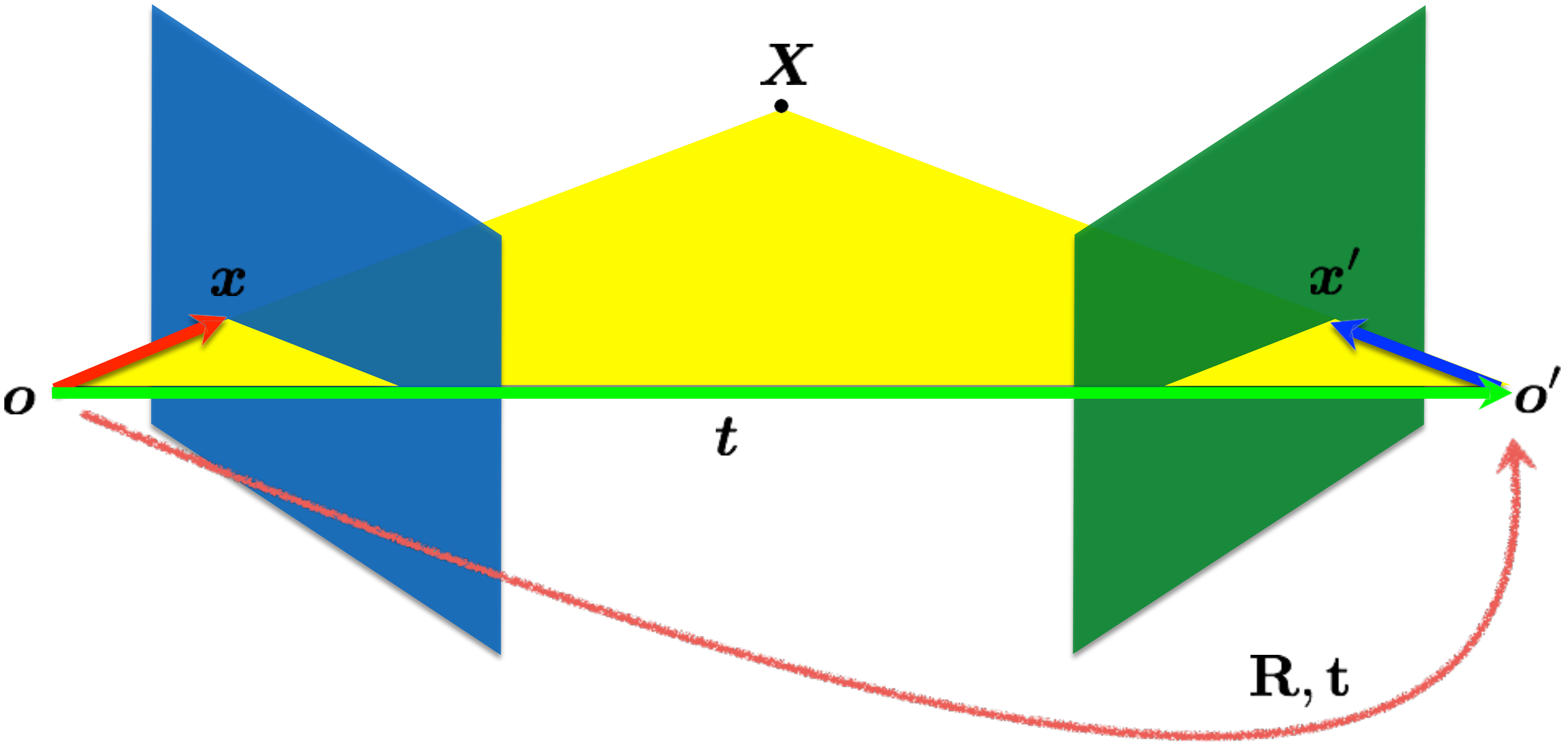
Skew symmetric

Compare with: dot product



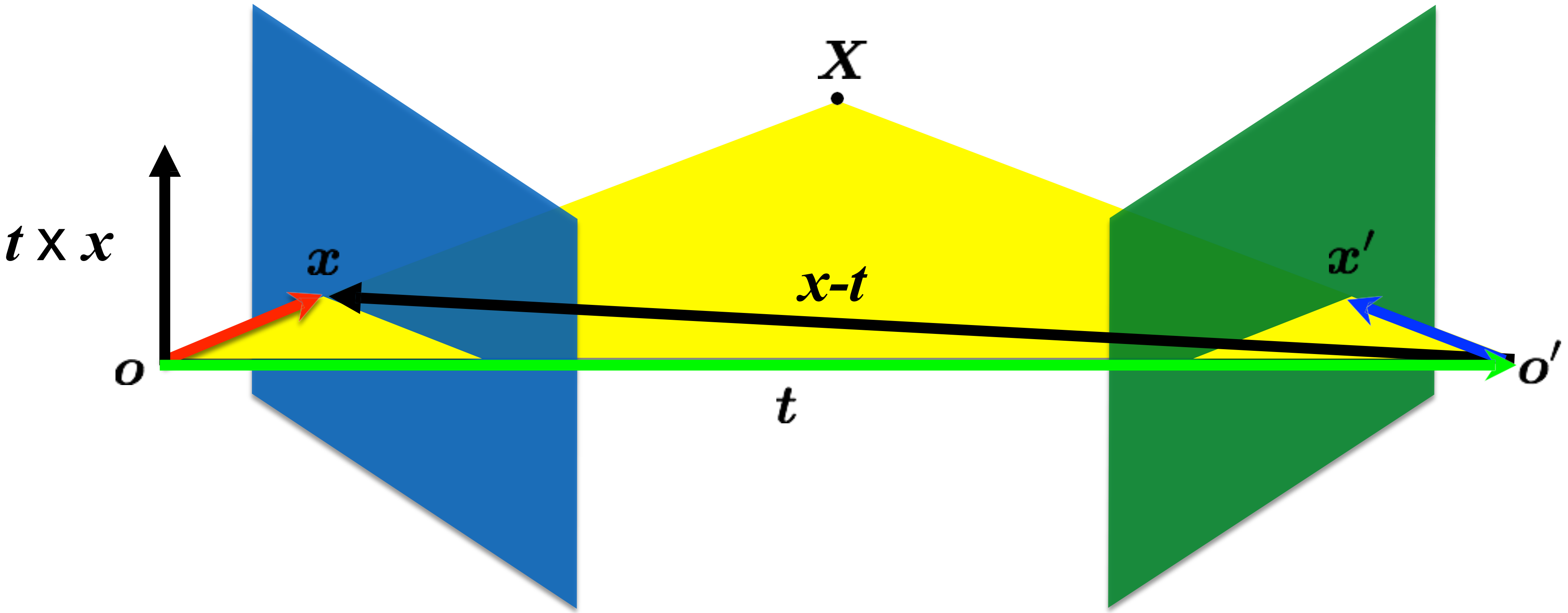
$\boldsymbol{c} \cdot \boldsymbol{a} = 0$ $\qquad$ $\boldsymbol{c} \cdot \boldsymbol{b} = 0$

dot product of two orthogonal vectors is (scalar) zero



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

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



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

dot product of orthogonal vectors

cross-product: vector orthogonal to plane

Putting it together

rigid motion

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

coplanarity

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

use skew-symmetric matrix to represent cross product

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

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

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

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

Essential Matrix
[Longuet-Higgins 1981]

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

Skew symmetric

$$\boxed{\mathbf{E} = \mathbf{R}\,[\mathbf{t}]_\times}$$

Properties of the E matrix

$$\boxed{\mathbf{E} = \mathbf{R}\,[\mathbf{t}]_\times}$$

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

Epipolar lines

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

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

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

(2D points expressed in <u>camera</u> coordinate system)

Properties of the E matrix

$$\boxed{\mathbf{E} = \mathbf{R}\,[\mathbf{t}]_{\times}}$$

- E has 5 degrees of freedom, why?
  - R has 3 degree of freedom
  - T has 3 degree of freedom
  - However since this is a projective transformation one can apply an arbitrary scale to E. Thus 1 degree of freedom less.

- E is rank 2, why?
  - $[t_x]$ is skew symmetric, hence rank 2.
  - Thus Det(E) = 0.

- E has 2 singular value both of which are equal.
  - $[t_x]$ a skew symmetric matrix has 2 equal singular values

2 possible notation

x' = R (x – t)

$$\boxed{\mathbf{E} = \mathbf{R}\,[\mathbf{t}]_\times}$$

x' = Rx – Rt
= Rx + t'

$$\mathbf{E} = [\tilde{\mathbf{t}}]_\times\, \mathbf{R}$$

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**

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

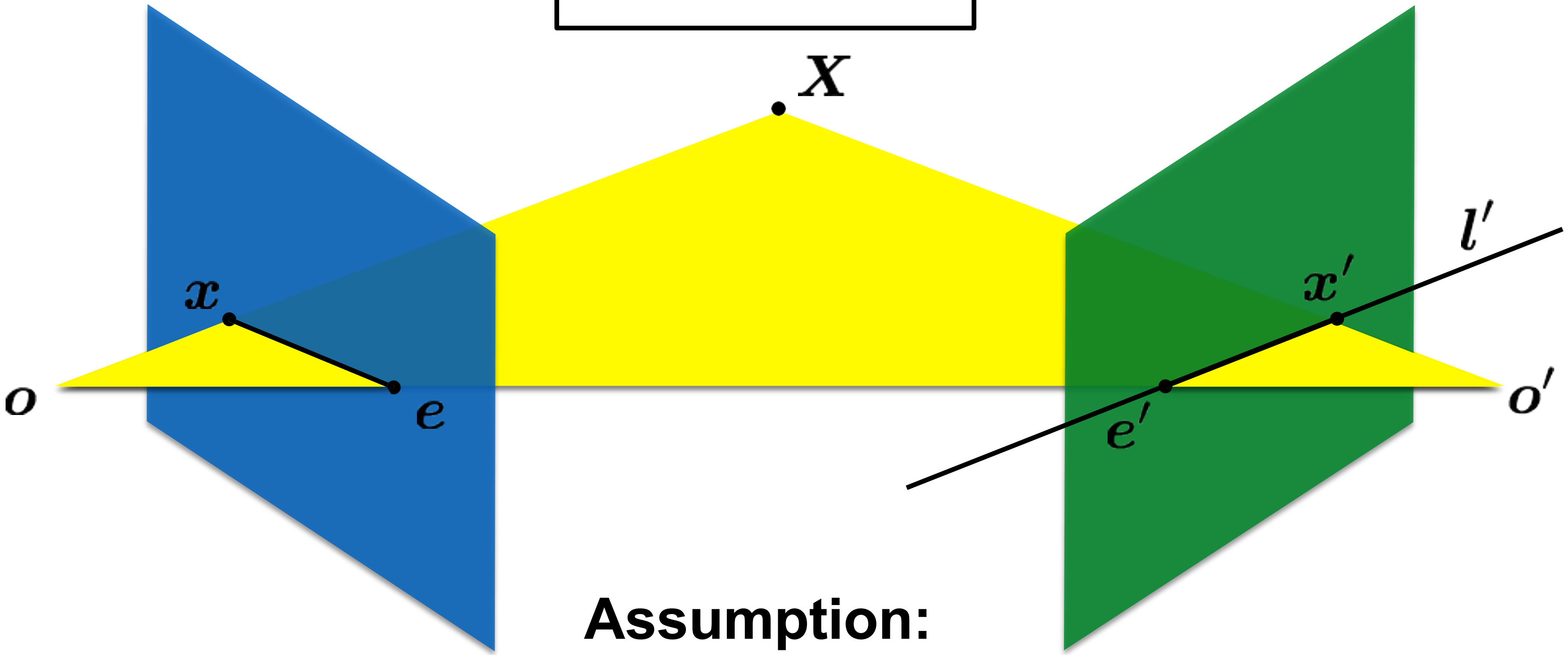
Homography maps a
point to a **point**

Today’s class

- Epipolar Geometry
- Essential Matrix
- Fundamental Matrix
- 8-point Algorithm
- Triangulation

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

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



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

How do you generalize to non-identity intrinsic matrices?

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

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

$$\underbrace{\hat{\boldsymbol{x}}}_{\text{camera point}} = \mathbf{K}^{-1}\underbrace{\boldsymbol{x}}_{\text{image point}}$$

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

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

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

$$\underbrace{\hat{\boldsymbol{x}}}_{\text{camera point}} = \mathbf{K}^{-1}\underbrace{\boldsymbol{x}}_{\text{image point}}$$

Writing out the epipolar constraint in terms of image coordinates

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

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

Fundamental Matrix

In practice we have points in image coordinate, i.e. pixel values.

Properties of the E matrix

$$\boxed{\mathbf{E} = \mathbf{R}\,[\mathbf{t}]_\times}$$

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

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

(2D points expressed in image coordinate system)

Same equation works in image coordinates!

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

it maps pixels to epipolar lines

Properties of the E matrix

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

- ~~E~~ F has ~~5~~ 7 degrees of freedom, why?
  - F is 3x3, has 8 degrees of freedom, since it is a projective transformation.
  - F is rank 2. So 1 less degree of freedom.

- ~~E~~ F is rank 2, why?
  - Same reason as E
  - $[t_x]$ is skew symmetric, hence rank 2.

- ~~E~~ F has 2 singular value both of which are ~~equal.~~

Essential/Fundamental Matrix vs Homography

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

They are both 3 x 3 matrices but …

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

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

- Rank 2
- 5 DoF

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

Fundamental matrix maps a **point** to a **line**

- Rank 2
- 7 DoF

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

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

- Rank 3
- 8 DoF

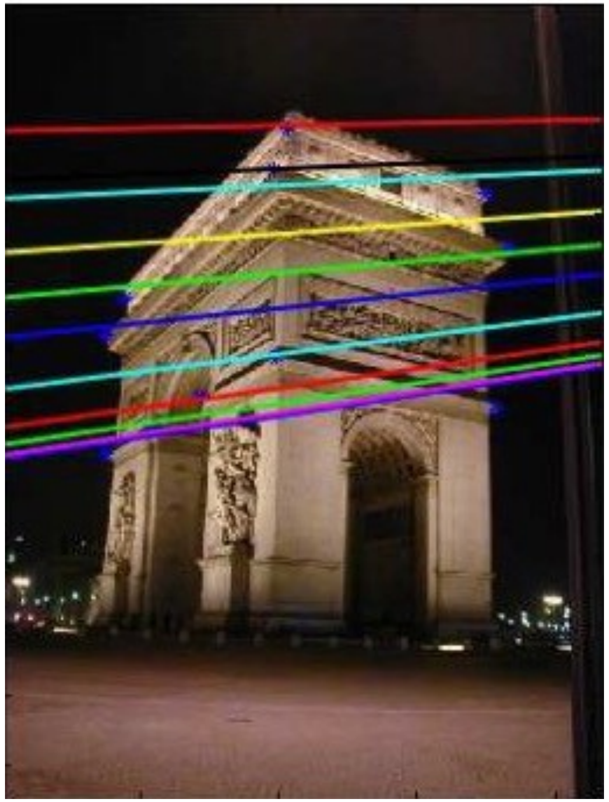
Homography is a special case of the Essential/Fundamental matrix, for planar scenes

The Fundamental Matrix Song

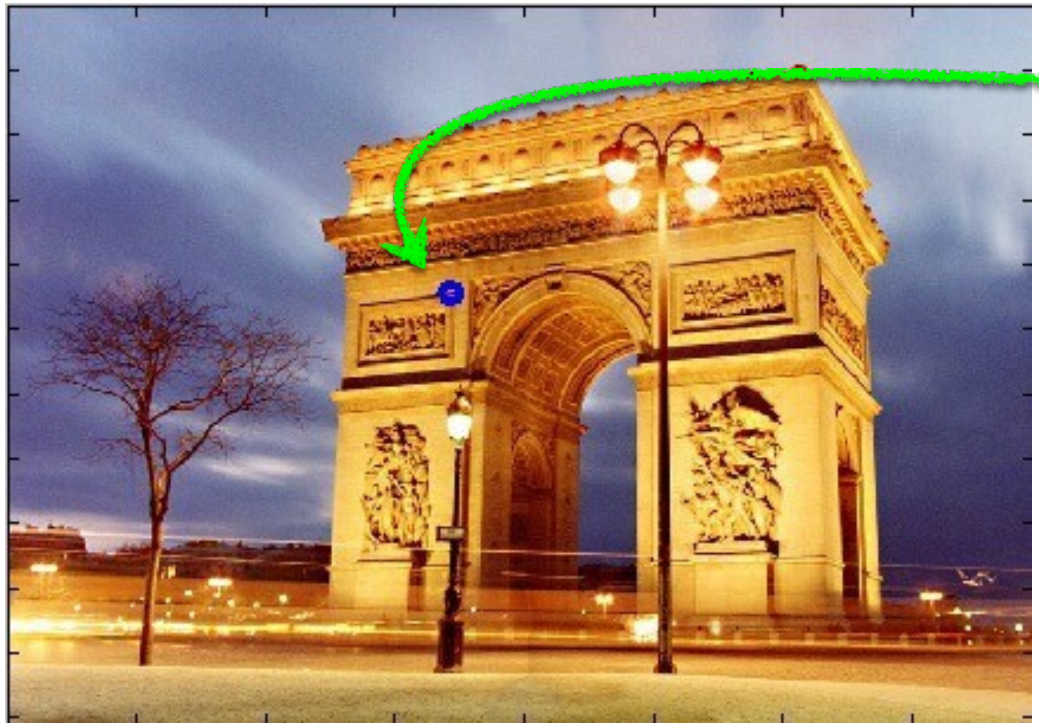
Daniel Wedge
https://www.youtube.com/watch?v=DgGV3l82NTk

Example

Epipolar lines



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



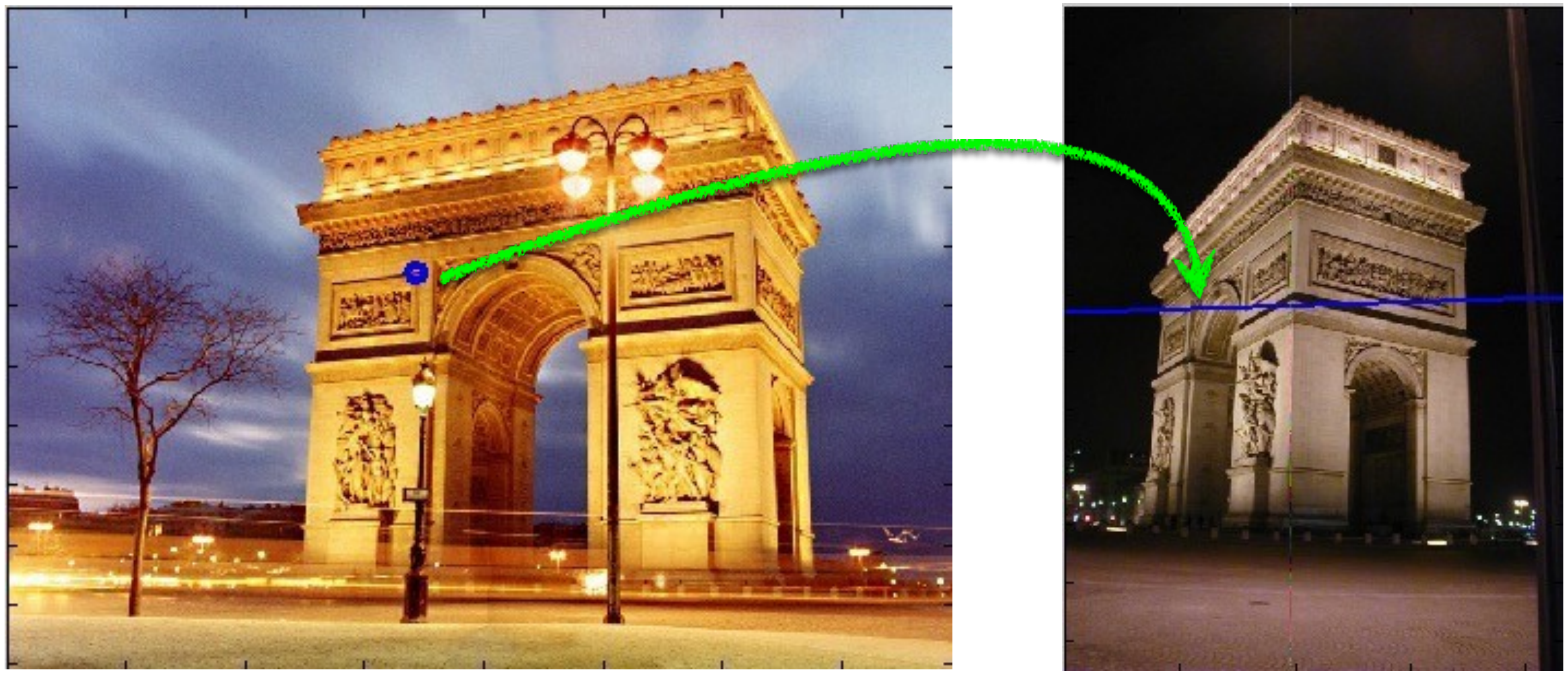
$$\boldsymbol{x} = \begin{bmatrix} 343.53 \\ 221.70 \\ 1.0 \end{bmatrix}$$

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

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

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

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



Where is the epipole?

How would you compute it?

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

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?

SVD (Singular Value Decomposition)

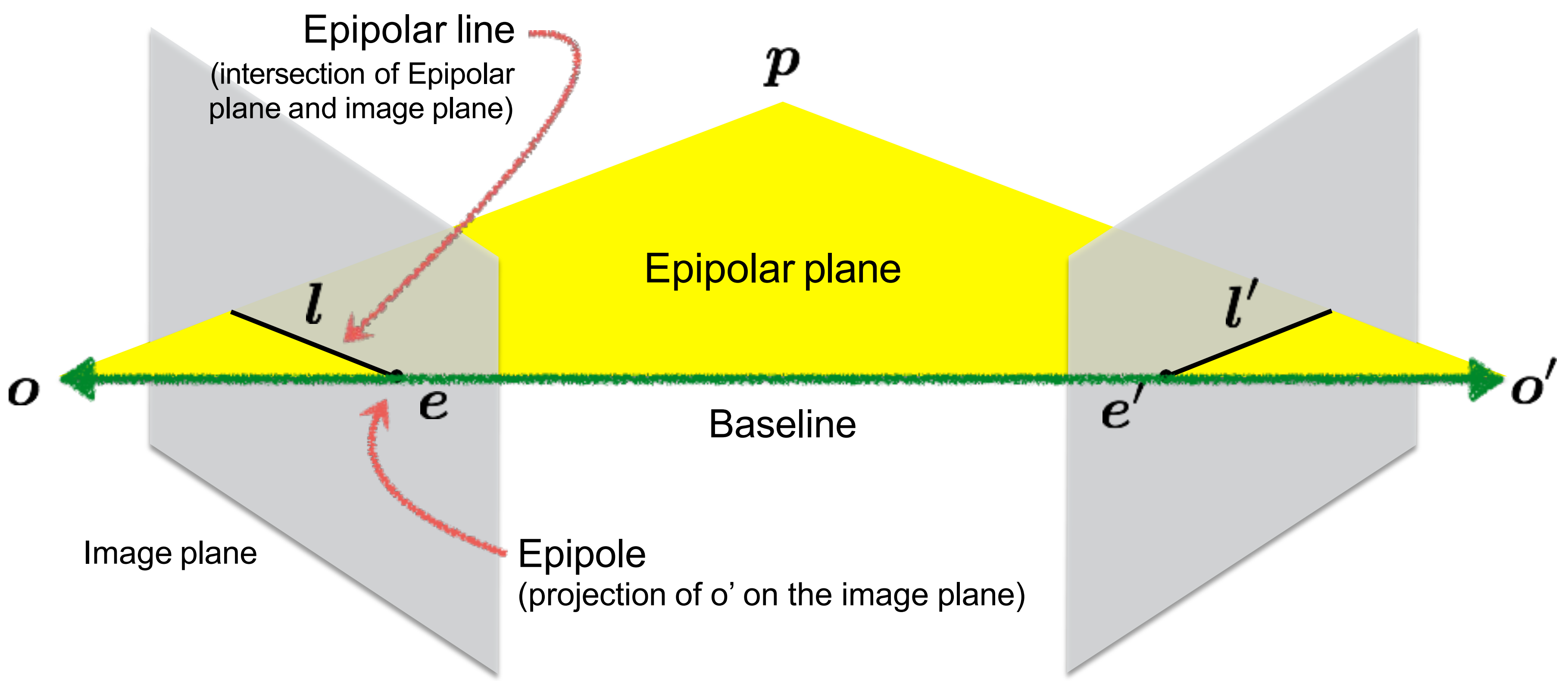
Slide Credits

- CS5670, Introduction to Computer Vision, Cornell Tech, by Noah Snavely.
- CS 194-26/294-26: Intro to Computer Vision and Computational Photography, UC Berkeley, by Angjoo Kanazawa.
- CS 16-385: Computer Vision, CMU, by Matthew O'Toole

Additional Reading

- Multiview Geometry, Hartley & Zisserman,
  - Chapter 9 (focus on topics discussed or mentioned in the slides).
  - Chapter 10.1, 10.2 (not discussed in class but important to understand, practical importance.)
  - Chapter 11.1, 11.2
  - Chapter 12.1, 12.2, 12.3, 12.4 (imp to understand)

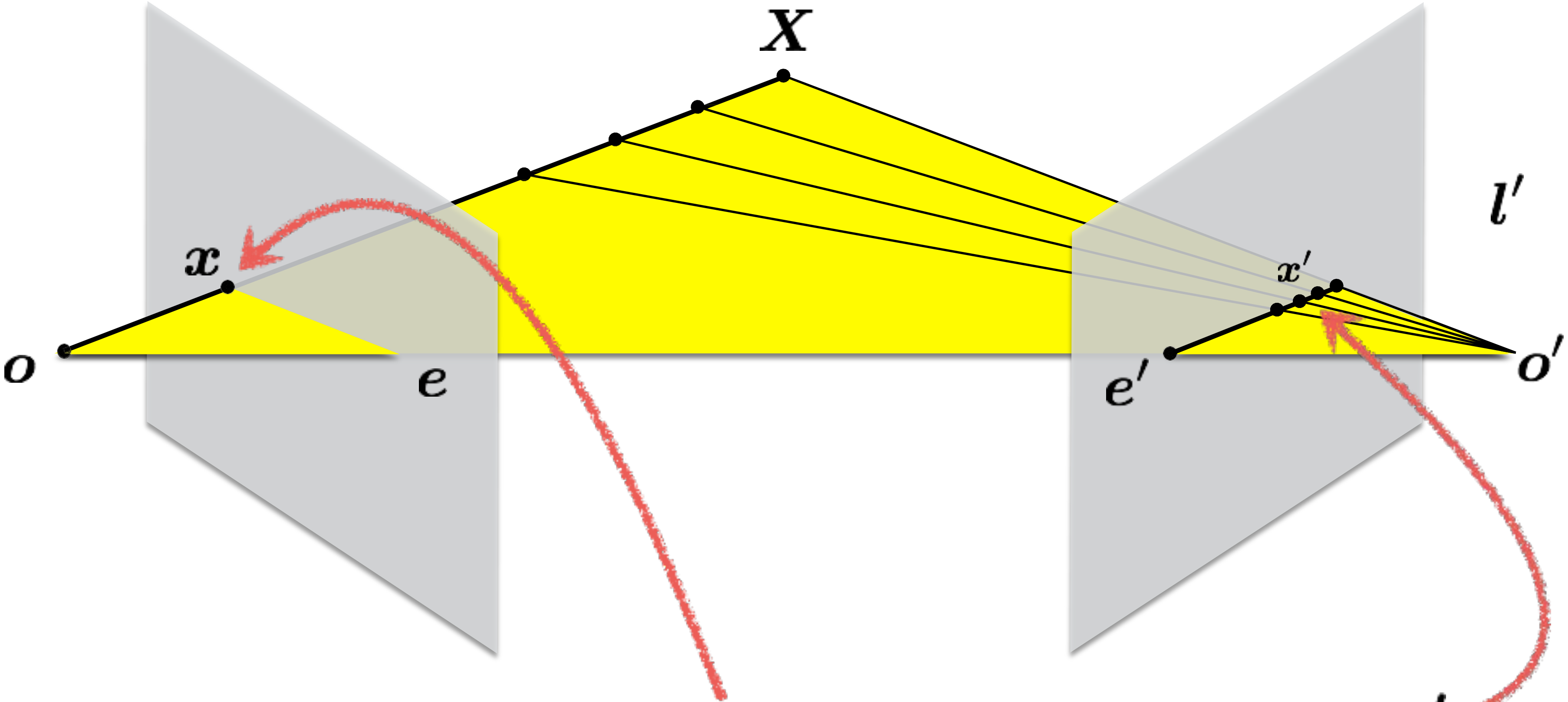
Epipolar geometry



Epipolar constraint

$$\mathbf{E}\boldsymbol{x} = \boldsymbol{l}' \Rightarrow \boldsymbol{x}'^{\top}\mathbf{E}\boldsymbol{x} = 0 \Rightarrow \boxed{\mathbf{E} = \mathbf{R}\left[\mathbf{t}\right]_{\times}}$$

Essential Matrix
[Longuet-Higgins 1981]



Potential matches for $\boldsymbol{x}$ lie on the epipolar line $\boldsymbol{l}'$

Fundamental Matrix

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

- Essential Matrix operates on points in camera coordinate system (after projection from 3D to 2D)
- Fundamental Matrix operates on points in pixel coordinate system
- E and F are both rank(2), but E has 2 singular values that are equal, but not F.
- E has 5 DoF and F has 7 DoF.

Sample problem

Suppose we have 2 cameras (all expressed in world-coordinate):

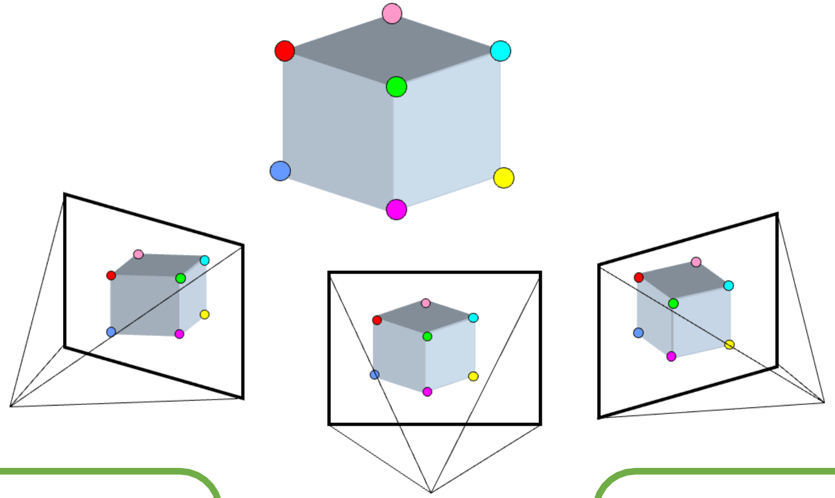
C1 at (0,0,0) with image plane at z=2

C2 at (1,1,1) with image plane at z=3

a) What is the equation of the baseline?
b) What are the two epipoles?
c) What is the Essential Matrix?
d) Show that epipoles lie in the null space of Essential matrix.
e) Consider a point (0,0,2) that lies on the image plane of C1. Where will that point lie on image plane of C2?

Big picture: 3 key components in 3D

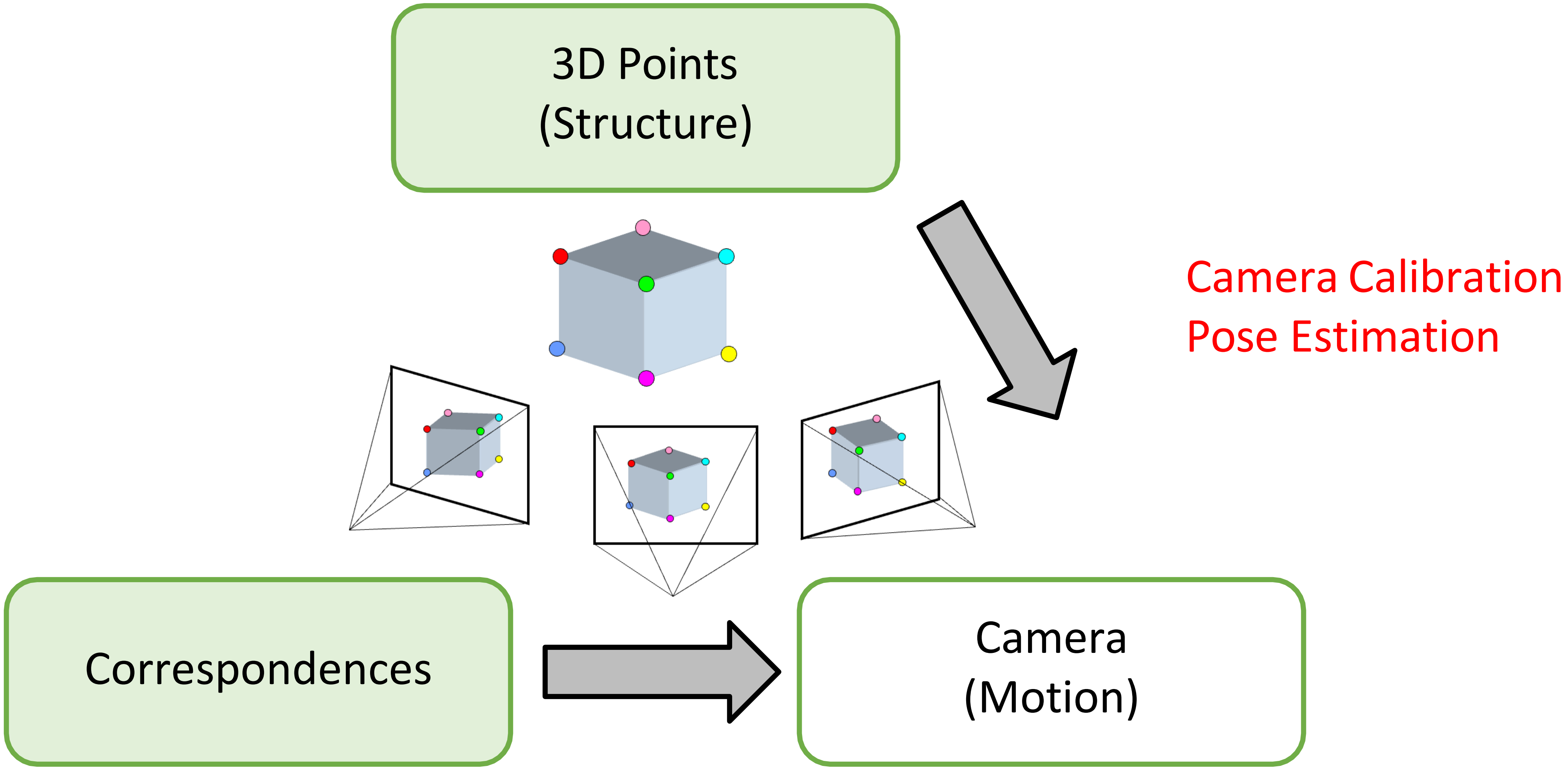
3D Points
(Structure)



Correspondences

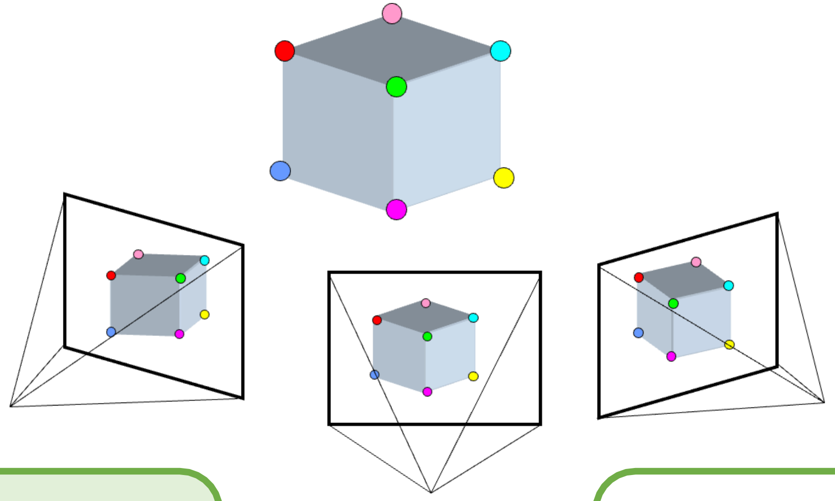
Camera
(Motion)

Big picture: 3 key components in 3D



Big picture: 3 key components in 3D

3D Points
(Structure)



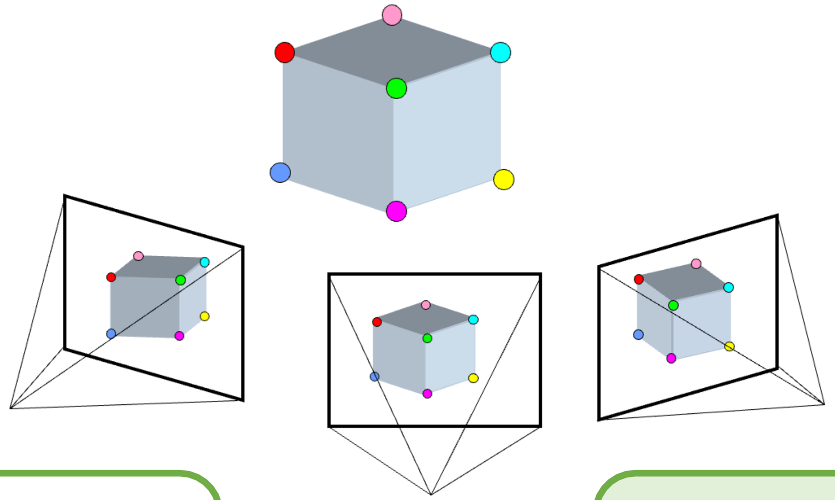
Epipolar /
Two-view
Geometry

Correspondences

Camera
(Motion)

Big picture: 3 key components in 3D

3D Points
(Structure)



Correspondences

Camera
(Motion)

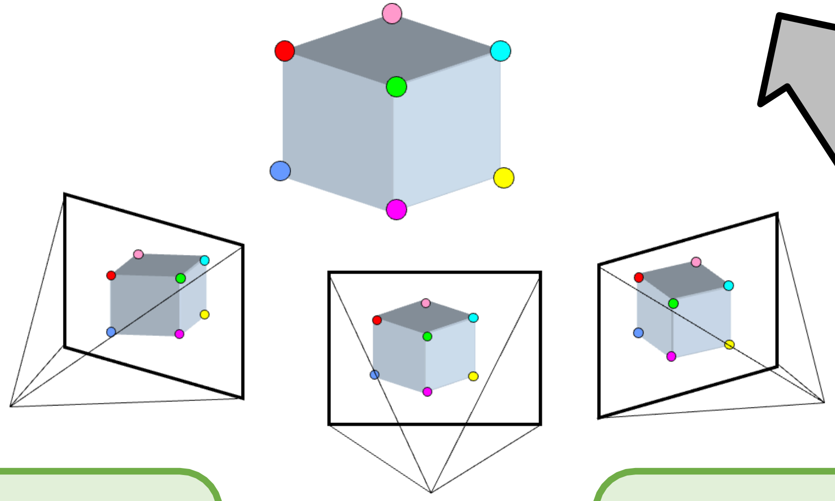
Stereo Matching

Big picture: 3 key components in 3D

Multiview Stereo
(more than 2 cameras)

3D Points
(Structure)

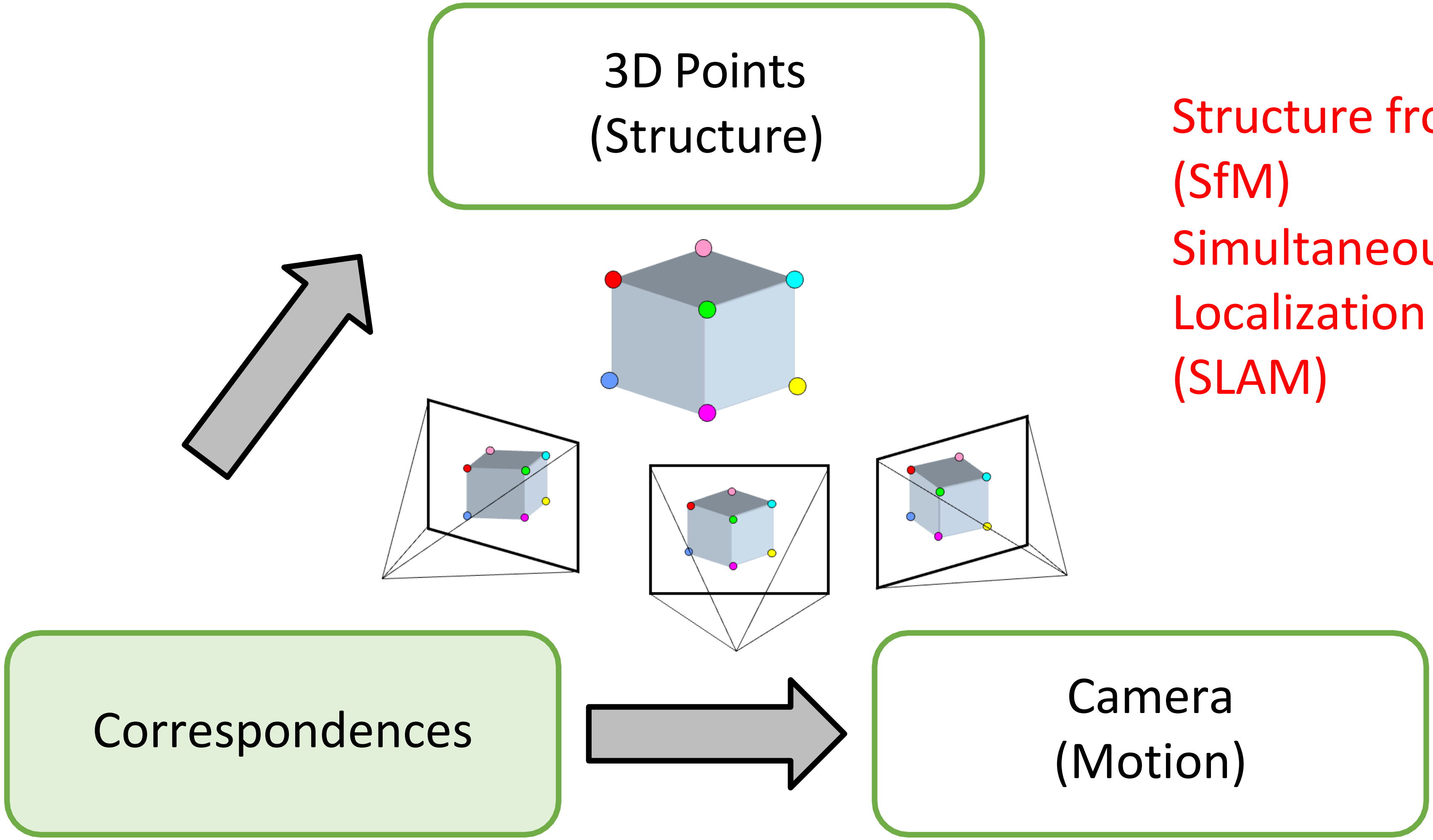
Triangulation



Correspondences

Camera
(Motion)

Big picture: 3 key components in 3D



Structure from Motion
(SfM)
Simultaneous
Localization & Mapping
(SLAM)

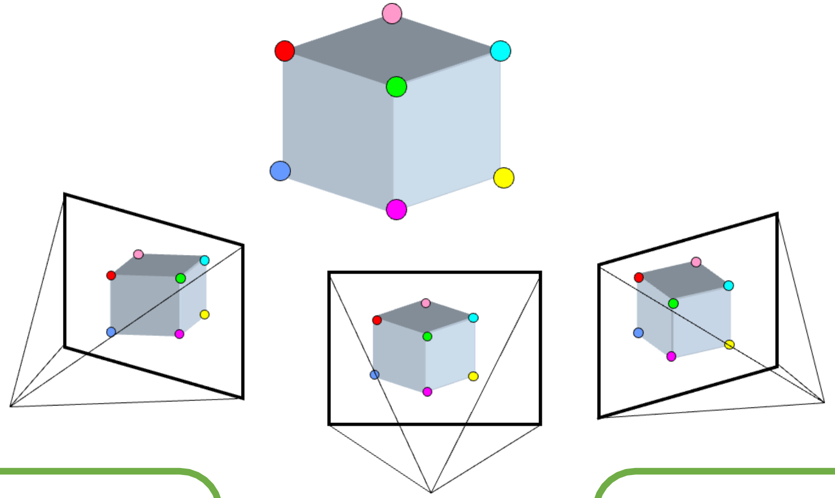
	Structure (scene geometry)	Motion (camera parameters)	Measurements (camera parameters)
Camera Calibration (Pose Estimation)	known	estimate	3D to 2D correspondences
Triangulation (Stereo, Multi-view Stereo)	estimate	known	2D to 2D coorespondences
Reconstruction (Structure from Motion, SLAM)	estimate	estimate	2D to 2D coorespondences

Today’s class

- Epipolar Geometry
- Essential Matrix
- Fundamental Matrix
- 8-point Algorithm
- Triangulation

Big picture: 3 key components in 3D

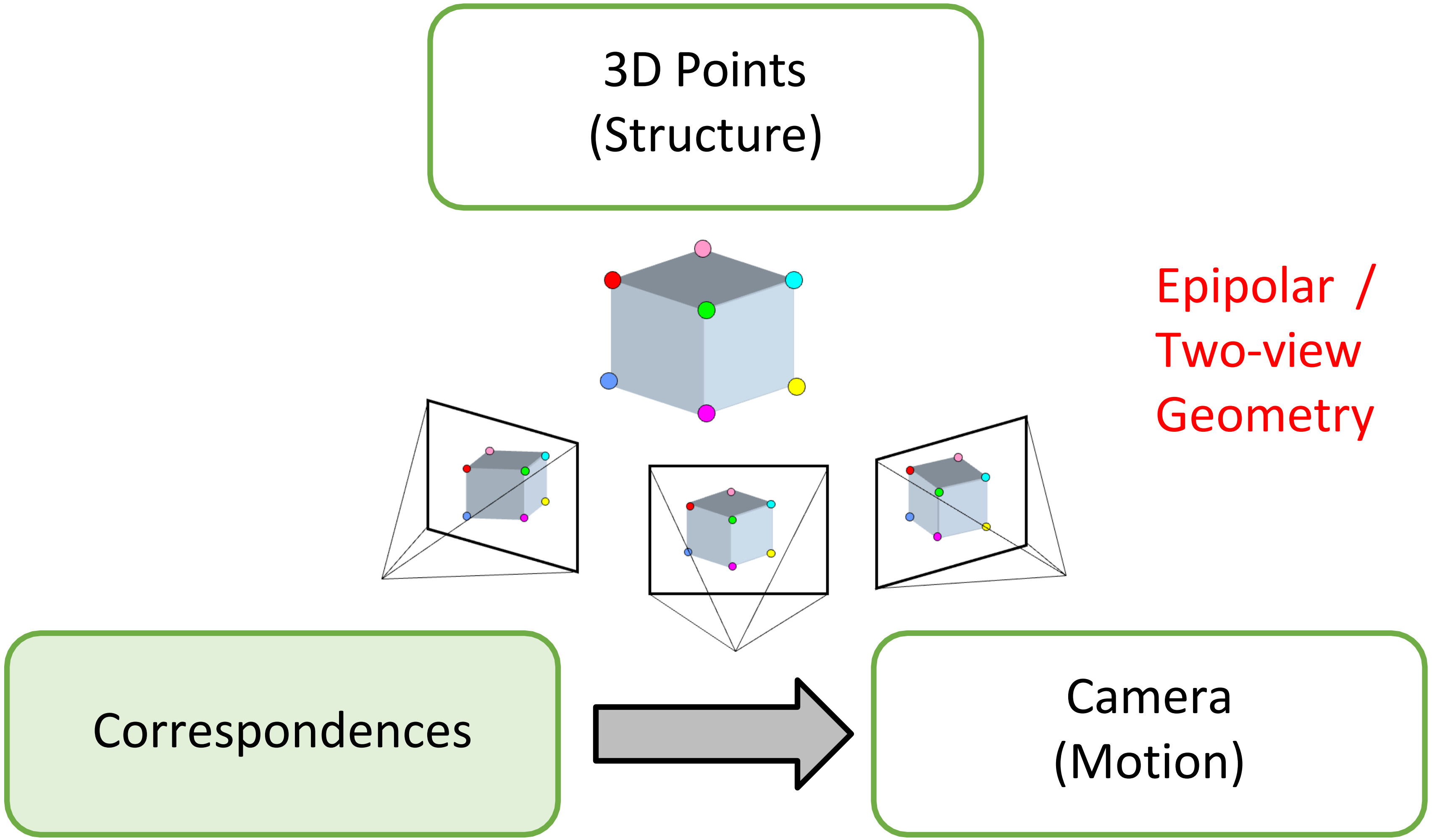
3D Points
(Structure)



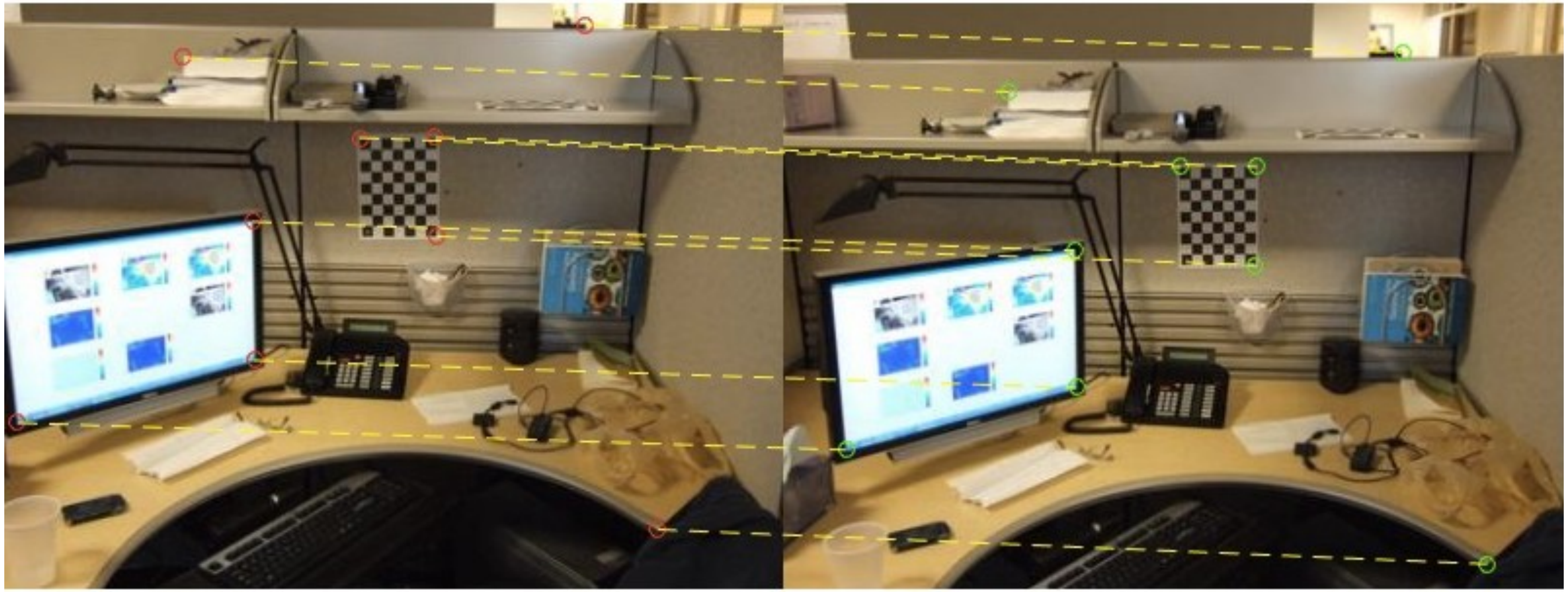
Correspondences

Camera
(Motion)

Big picture: 3 key components in 3D



Estimating the fundamental matrix



How do we get these?
- Run SIFT detector on both images
- Match detected points in both images to establish correspondence.

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?*

Solve with SVD!

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 but F is defined up to scale

Hence, the 8 point algorithm!

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

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

We need at least 8 points

How many equations do you need?

How do you solve a homogeneous linear system?

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

8 x9 9x1

Total Least Squares

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

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

SVD!

Problem with 8-point algorithm

$$\begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu_n' & v_nu_n' & u_n' & u_nv_n' & v_nv_n' & v_n' & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

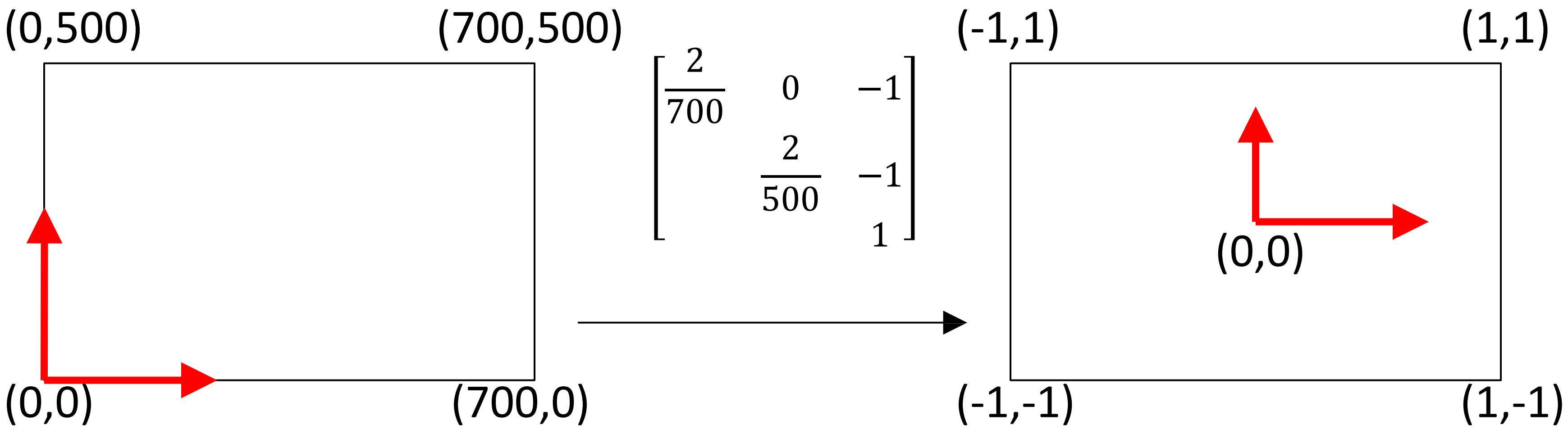
$$\begin{array}{ccccccccc} \sim 10000 & \sim 10000 & \sim 100 & \sim 10000 & \sim 10000 & \sim 100 & \sim 100 & \sim 100 & 1 \end{array}$$

Orders of magnitude difference
between column of data matrix
–→ least-squares yields poor results

Normalized 8-point algorithm

Normalized least squares yields good results

Transform image to ~[-1,1]



Normalized 8-point algorithm

- Transform input by $\hat{\mathbf{x}}_\mathrm{i} = \mathbf{T}\mathbf{x}_\mathrm{i}$, $\hat{\mathbf{x}}'_\mathrm{i} = \mathbf{T}\mathbf{x}'_\mathrm{i}$
- Call 8-point on $\hat{\mathbf{x}}_\mathrm{i}, \hat{\mathbf{x}}'_\mathrm{i}$ to obtain $\hat{\mathbf{F}}$
- $\mathbf{F} = \mathbf{T}'^{\mathrm{T}}\hat{\mathbf{F}}\mathbf{T}$

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

$$\underbrace{\boxed{\hat{\mathbf{x}}'^{\mathrm{T}}\mathbf{T}'^{-\mathrm{T}}}\mathbf{F}\boxed{\mathbf{T}^{-1}\hat{\mathbf{x}}}}_{\hat{\mathbf{F}}} = 0$$

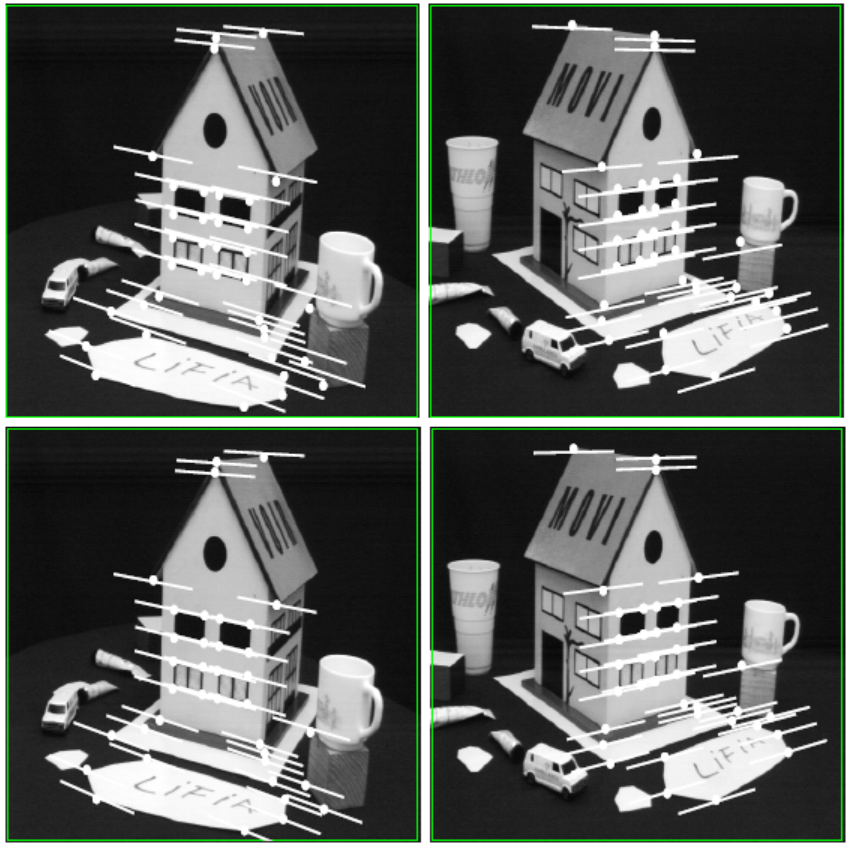
Fundamental matrix of normalized camera coordinate

The Normalized 8-Point Algorithm

- Center the image data at the origin, and scale it so the mean squared distance between the origin and the data points is 2 pixels
- Use the eight-point algorithm to compute ***F*** from the normalized points
- Enforce the rank-2 constraint (for example, take SVD of ***F*** and throw out the smallest singular value)
- Transform fundamental matrix back to original units: if ***T*** and ***T'*** are the normalizing transformations in the two images, then the fundamental matrix in original coordinates is $\boldsymbol{T'}^{T} \boldsymbol{F} \boldsymbol{T}$

(Hartley, 1995)

Comparison



	8-point	Normalized 8-point	Nonlinear least squares
Av. Dist. 1	2.33 pixels	0.92 pixel	0.86 pixel
Av. Dist. 2	2.18 pixels	0.85 pixel	0.80 pixel

Notes

- If we know the calibration matrices of the two cameras, we can estimate the essential matrix: $E = K^T F K'$
- The essential matrix gives us the relative rotation and translation between the cameras, or their extrinsic parameters.
- Fundamental matrix lets us compute relationship up to scale for cameras with unknown intrinsic calibrations.
- **Estimating the fundamental matrix is a kind of “weak calibration”**

How do you solve a homogeneous linear system?

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

8 x9 9x1

Total Least Squares

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

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

SVD!

How do we guarantee that rank(F)=2?

Enforcing rank constraints

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

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

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,

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

(Normalized) Eight-Point Algorithm

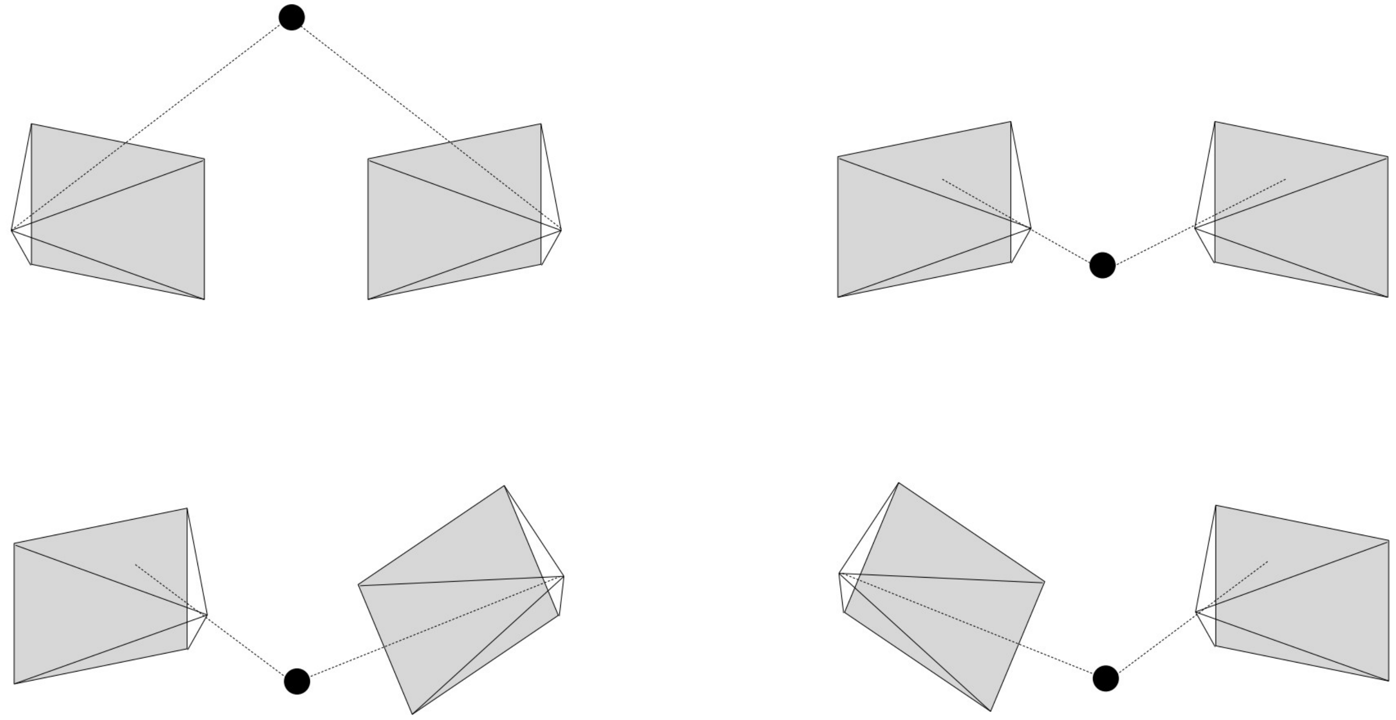
1. (Normalize points)
2. Construct the M x 9 matrix **A** (M=8 atleast)
3. Find the SVD of **A**
4. Entries of **F** are the elements of column of **V** corresponding to the least singular value
4. (Enforce rank 2 constraint on F)
5. (Un-normalize F)

Fundamental -> Essential -> Rotation + Translation

- From normalized 8-pt algorithm we have F, s.t. rank(F)=2.
- Recover intrinsic camera matrix K and K' (find focal length of 2 cameras, often comes as a part of meta data).
- Recover Essential matrix E from $\mathbf{F} = \mathbf{K}'^{-\top}\mathbf{E}\mathbf{K}^{-1}$
- An ideal E is rank(2) and has 2 singular values that are equal, and is upto a scale.
  - An ideal E will have SVD E=U diag(1,1,0) V^T.
  - Project estimated E such that 2 singular values are 1.
- Decompose Essential matrix to obtain Rotation and Translation $\mathbf{E} = [\tilde{\mathbf{t}}]_\times \mathbf{R}$
  - 4 possible solutions
  - See Results 9.18 & 9.19, pg 258-259 for the proof.

4 possible solutions of $\mathbf{E} = [\tilde{\mathbf{t}}]_{\times} \mathbf{R}$ decomposition

Four configurations: can be resolved by point triangulation.



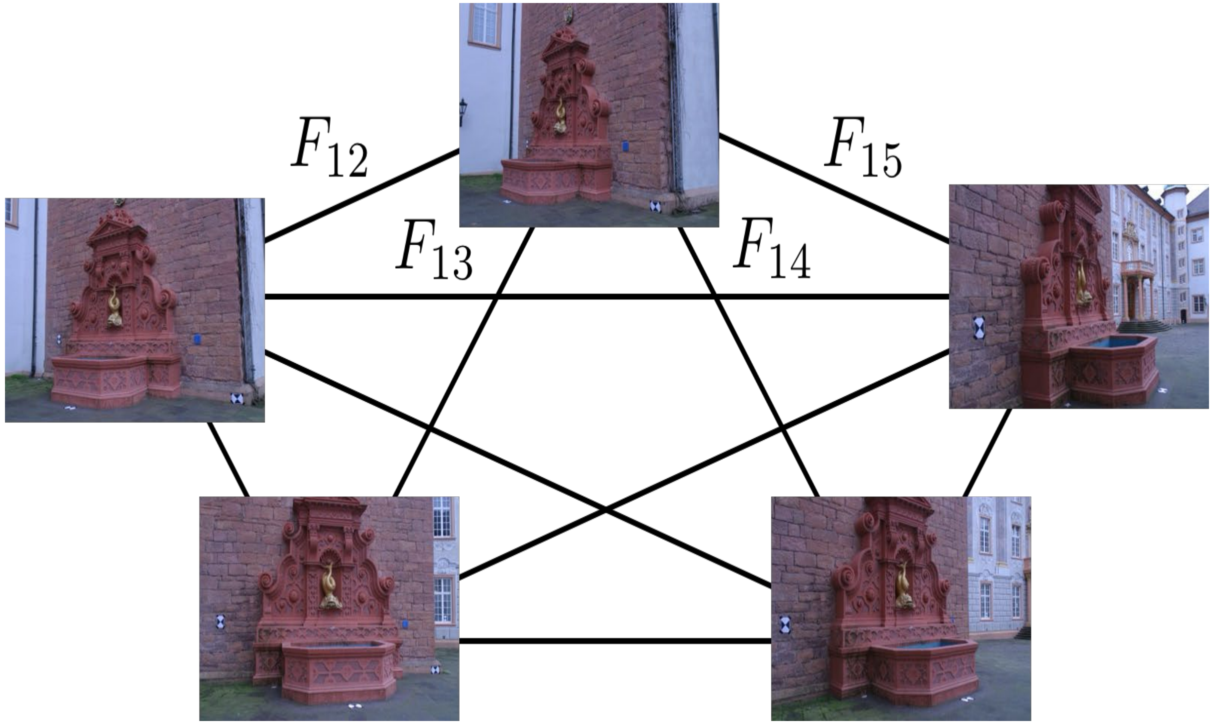
What about more than two views?

- The geometry of three views is described by a 3 x 3 x 3 tensor called the *trifocal tensor*

- The geometry of four views is described by a 3 x 3 x 3 x 3 tensor called the *quadrifocal tensor*

- After this it starts to get complicated…

"A New Rank Constraint on Multi-view Fundamental Matrices, and its Application to Camera Location Recovery", Sengupta et. al. CVPR 2017.



$$F = \begin{bmatrix} \mathbf{0} & F_{12} & F_{13} & F_{14} & F_{15} \\ F_{21} & \mathbf{0} & F_{23} & F_{24} & F_{25} \\ F_{31} & F_{32} & \mathbf{0} & F_{34} & F_{35} \\ F_{41} & F_{42} & F_{43} & \mathbf{0} & F_{45} \\ F_{51} & F_{52} & F_{53} & F_{54} & \mathbf{0} \end{bmatrix}$$

$$\text{with } F = A + A^T,$$
$$rank(A) = 3 \text{ and } rank(F) = 6.$$

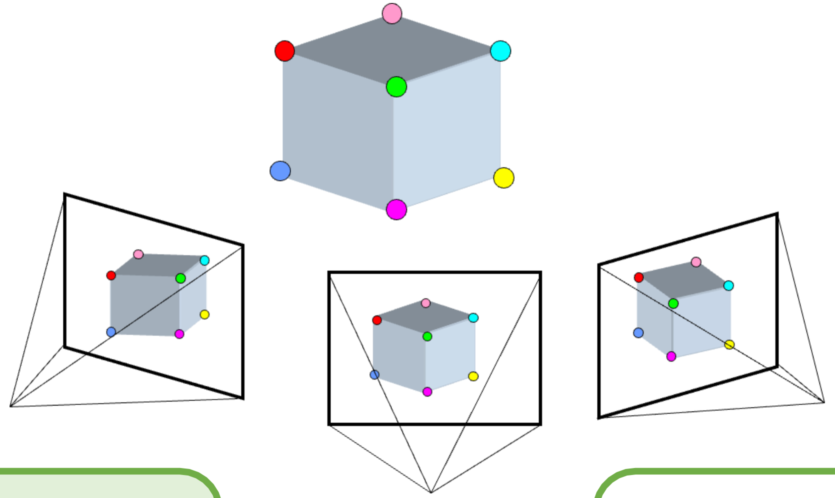
In case of all collinear cameras : $rank(A) \leq 2 \text{ and } rank(F) \leq 4$

Today’s class

- Epipolar Geometry
- Essential Matrix
- Fundamental Matrix
- 8-point Algorithm
- Triangulation

Big picture: 3 key components in 3D

3D Points
(Structure)

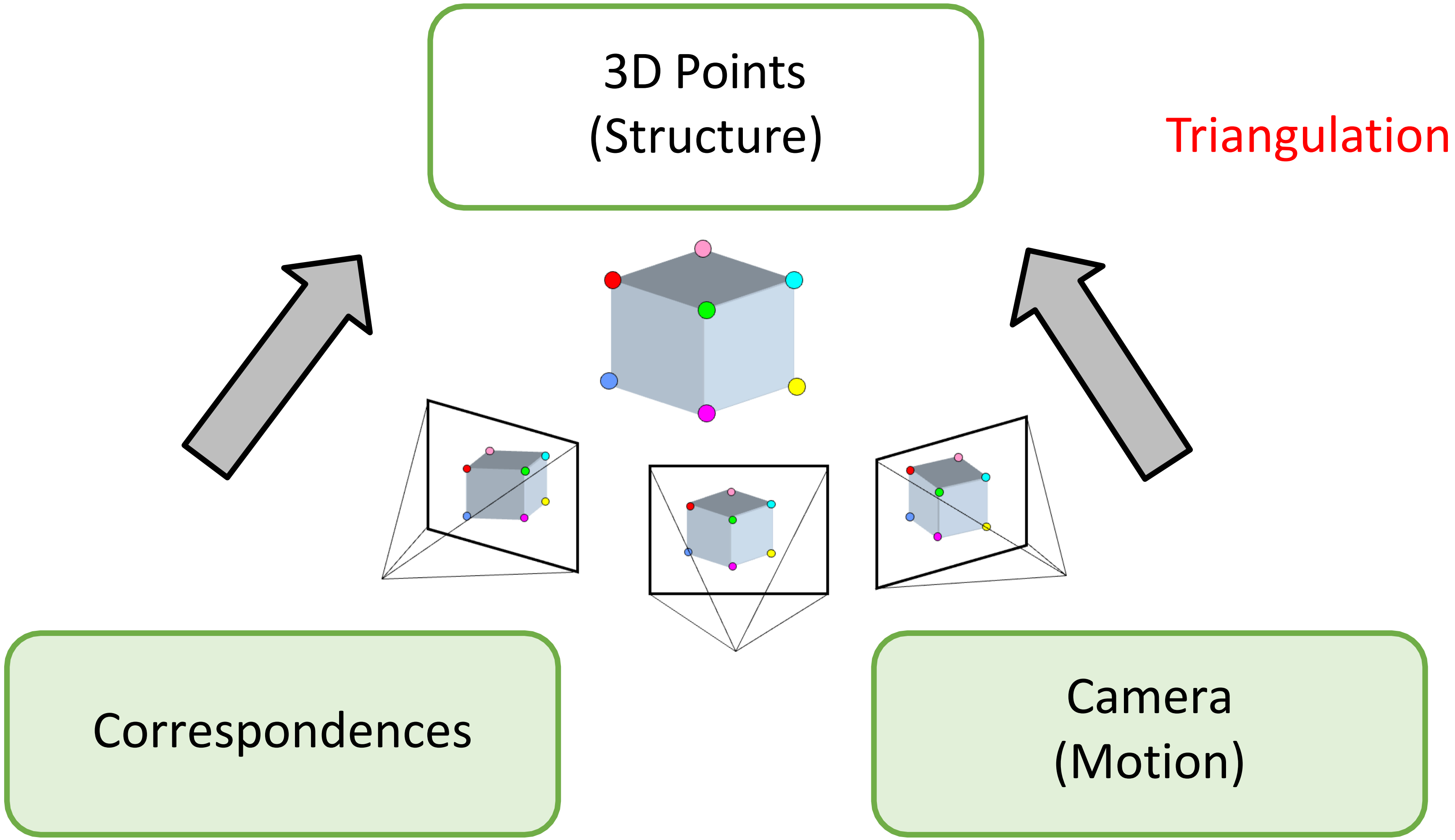


Estimate
Fundamental matrix

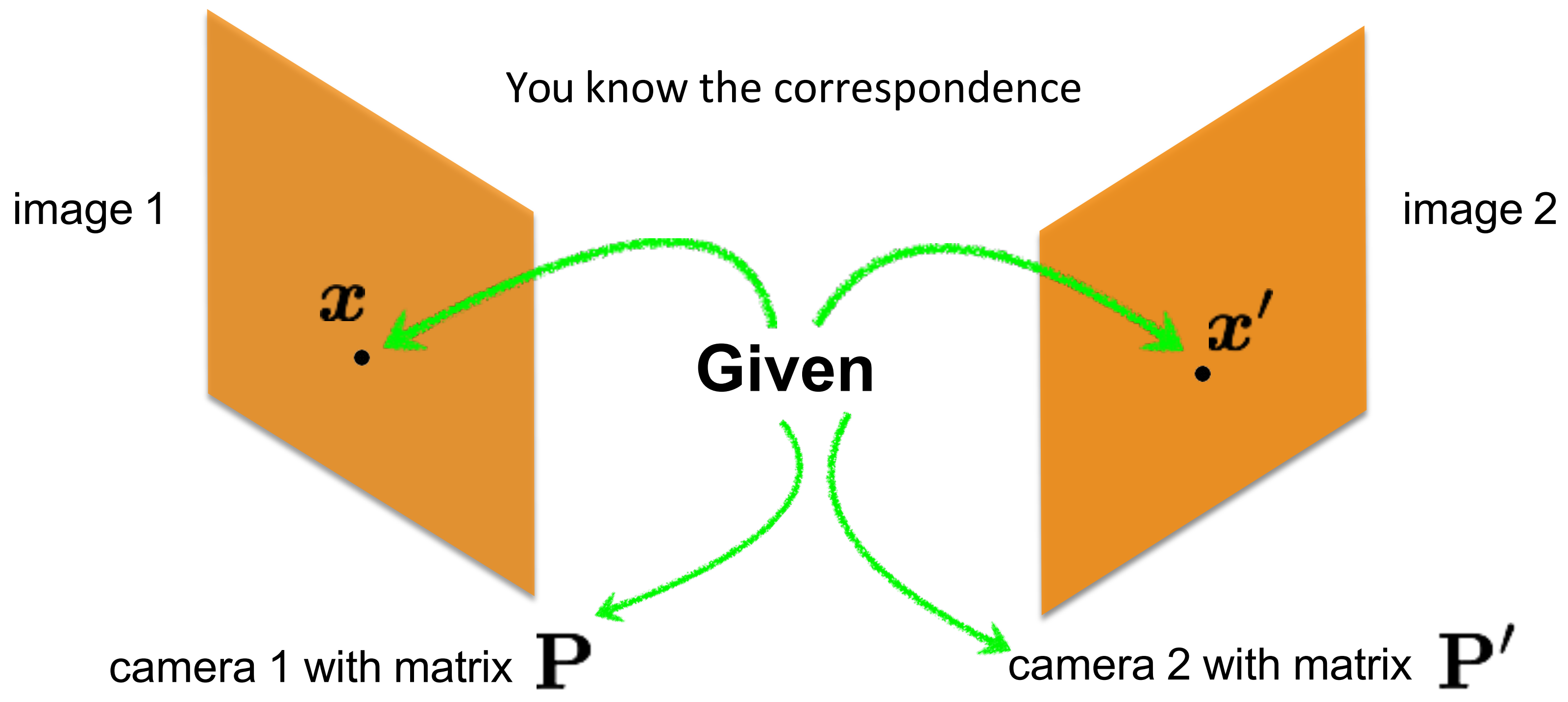
Correspondences

Camera
(Motion)

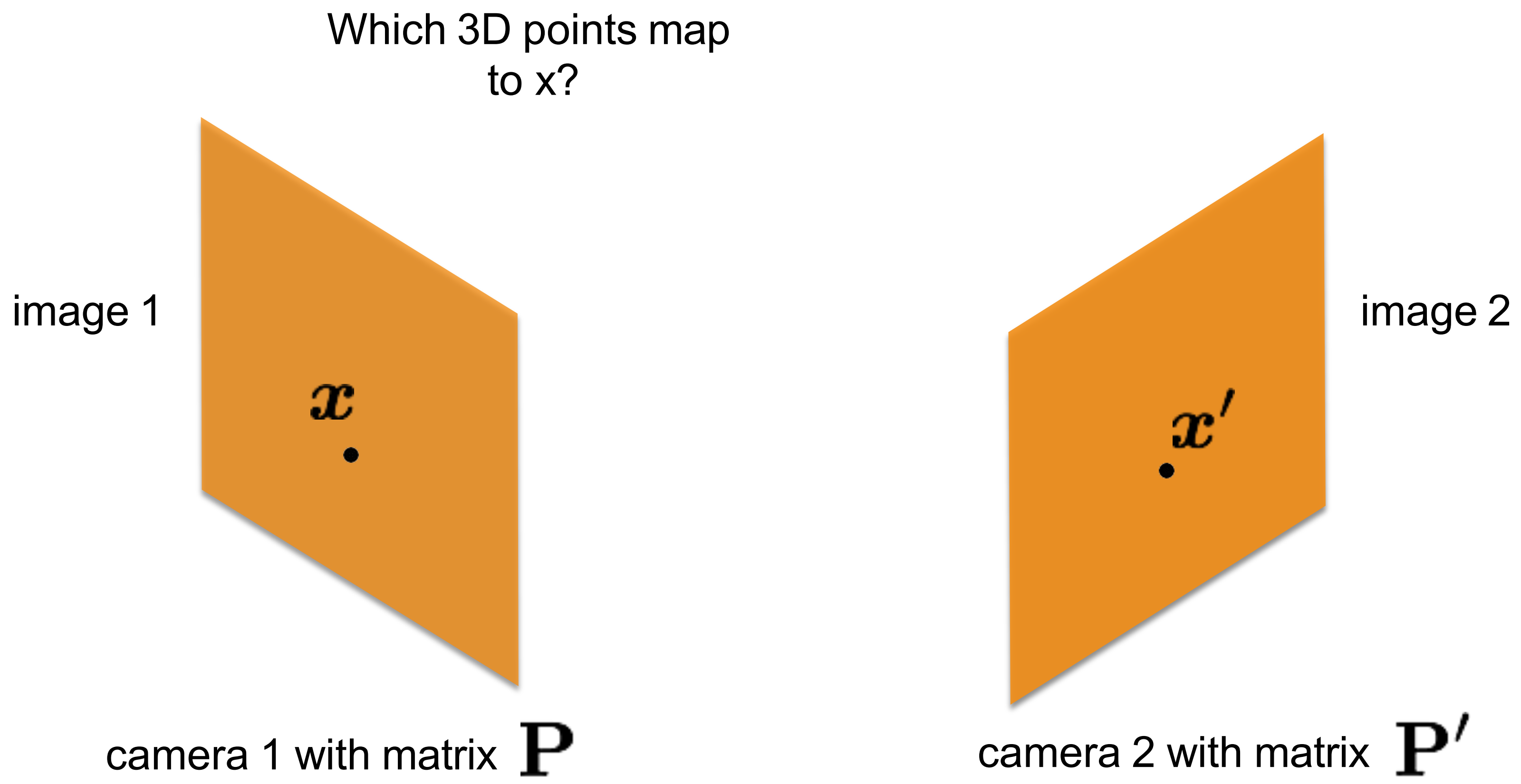
Big picture: 3 key components in 3D



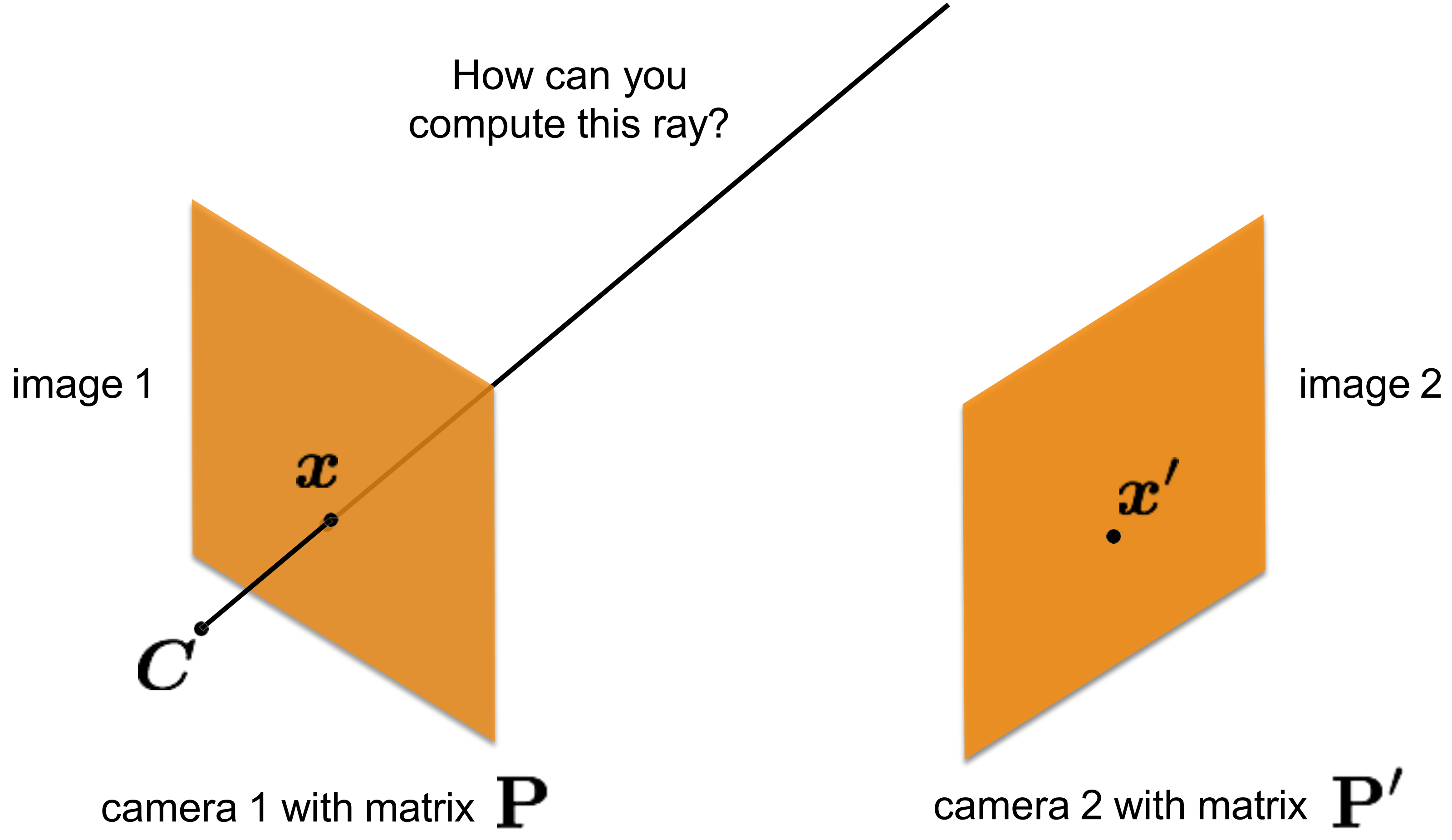
Triangulation



Triangulation



Triangulation

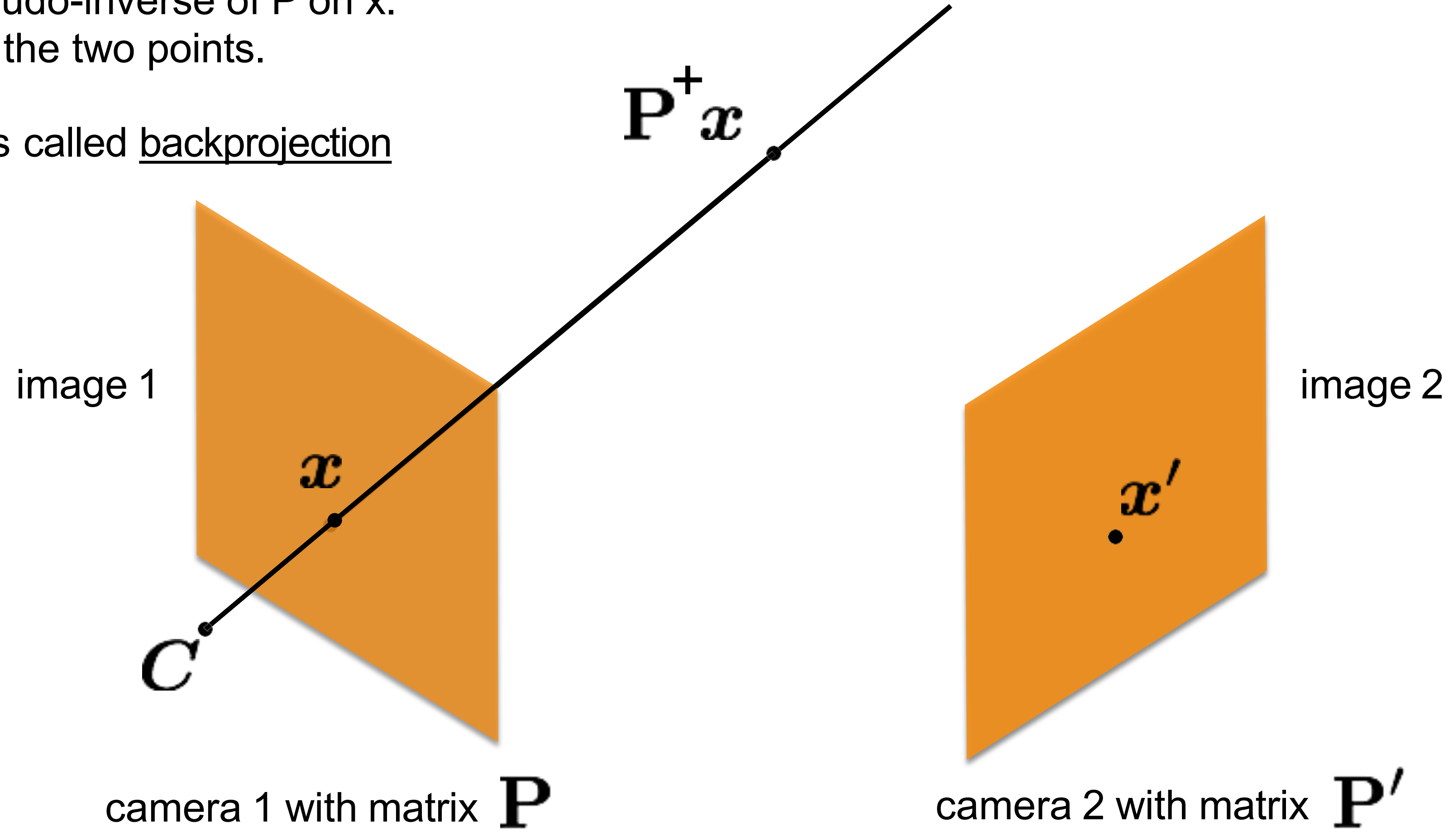


Triangulation

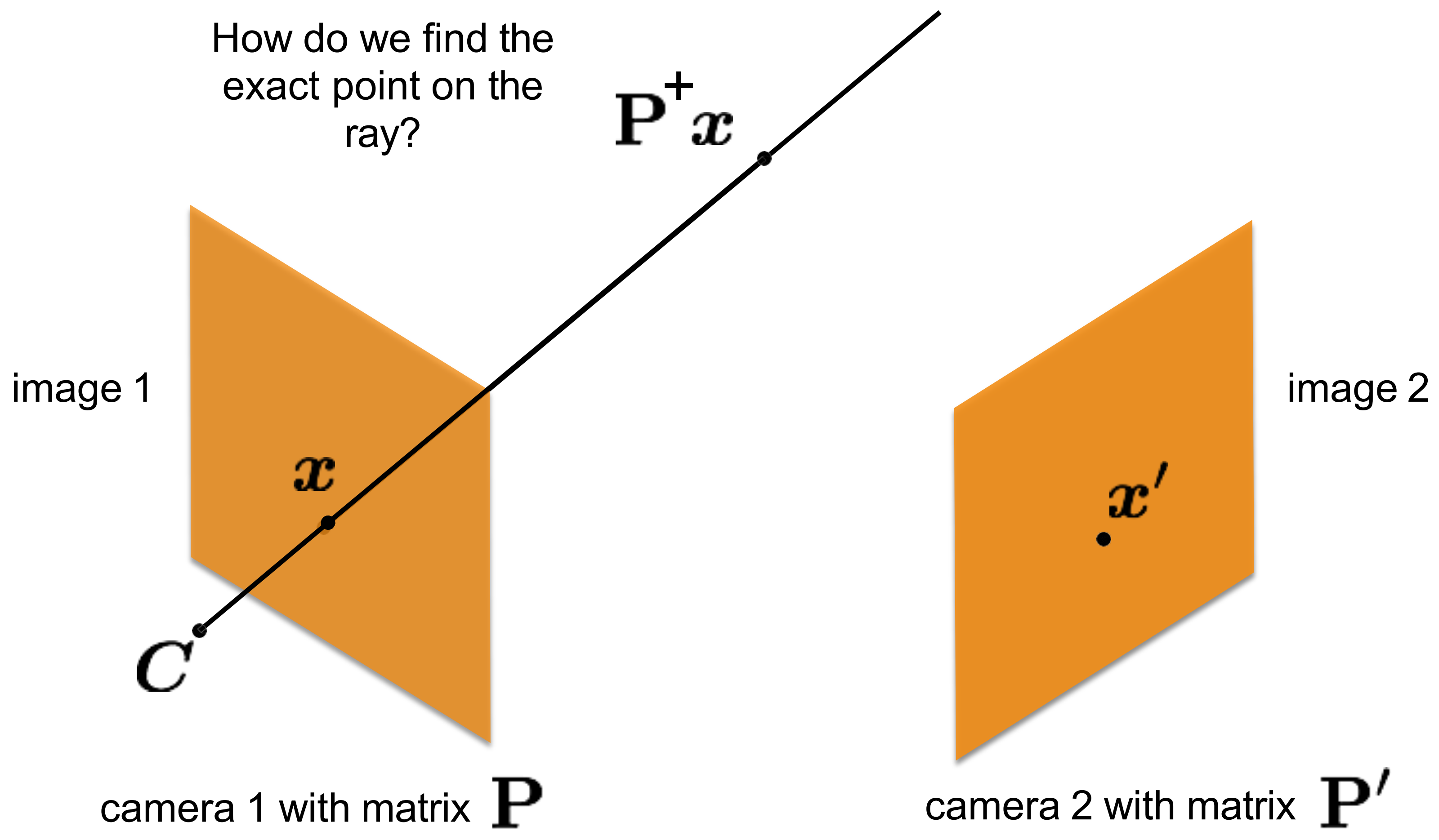
Create two points on the ray:

1. find the camera center; and
2. apply the pseudo-inverse of P on x.
3. then connect the two points.

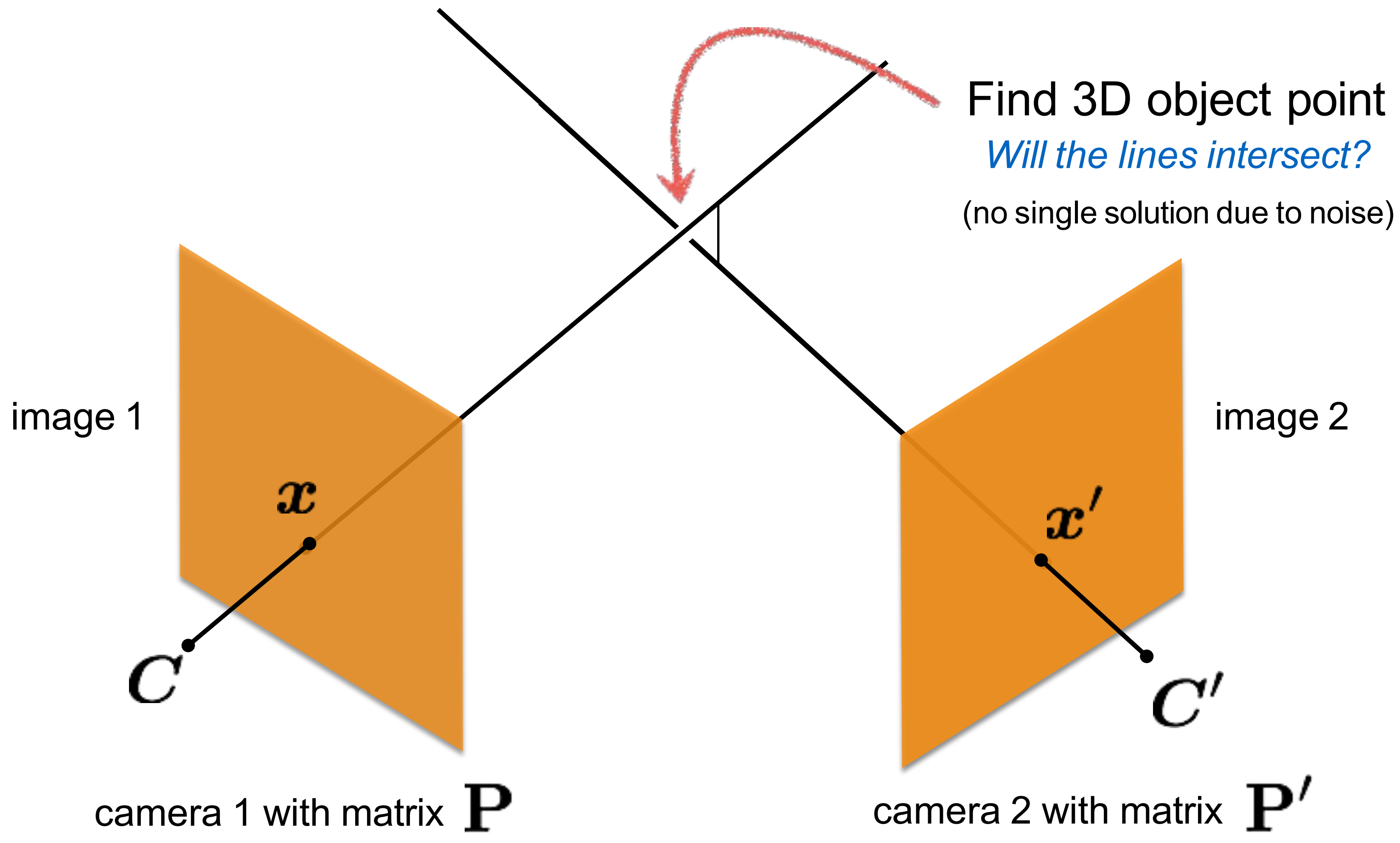
This procedure is called <u>backprojection</u>



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

$$\mathbf{X}$$

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

(homogeneous coordinate)

This is a similarity relation because it involves homogeneous coordinates

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

(heterogeneous coordinate)

Same ray direction but differs by a scale factor

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

How do we solve for unknowns in a similarity relation?

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

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)

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

Do the same after first expanding out the camera matrix and points

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

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

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

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 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 on 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

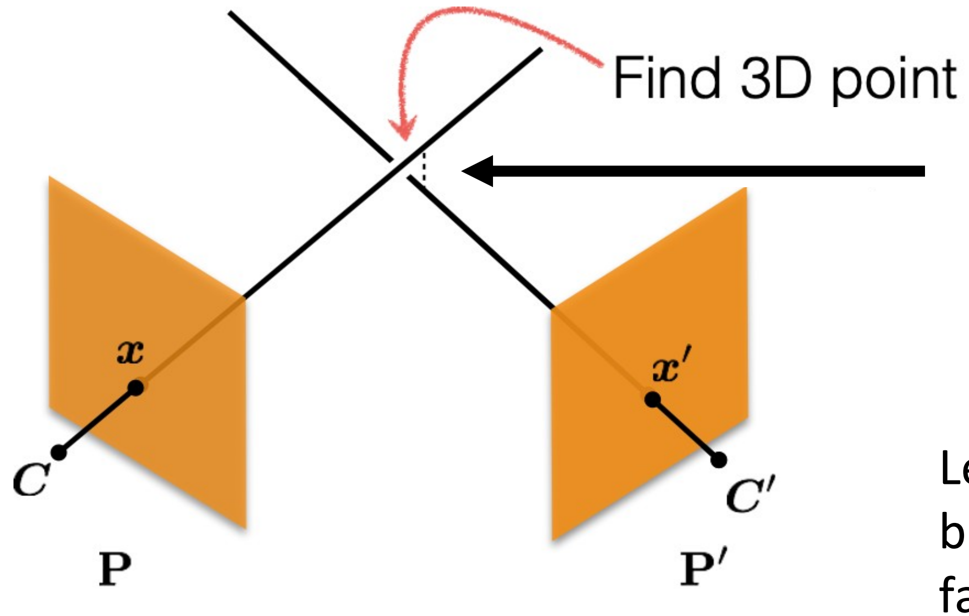
$$\begin{bmatrix} y\boldsymbol{p}_3^\top - \boldsymbol{p}_2^\top \\ \boldsymbol{p}_1^\top - x\boldsymbol{p}_3^\top \\ y'\boldsymbol{p}_3'^\top - \boldsymbol{p}_2'^\top \\ \boldsymbol{p}_1'^\top - x'\boldsymbol{p}_3'^\top \end{bmatrix} \boldsymbol{X} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

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

How do we solve homogeneous linear system?

SVD!

Triangulation Disclaimer: Noise



Ray's don't always intersect
because of noise!!!

$$\mathbf{X} \text{ s.t.}$$

$$\mathbf{x} = \mathbf{P}\mathbf{X}, \;\; \mathbf{x}' = \mathbf{P}'\mathbf{X}$$

Least squares get you to a reasonable solution but it's not the actual geometric error (it's how far away the solution is from Ax = 0)

In practice with noise, you do non- linear least squares, or "bundle adjustment"

Slide credit: Shubham Tulsiani

Slide Credits

- CS5670, Introduction to Computer Vision, Cornell Tech, by Noah Snavely.
- CS 194-26/294-26: Intro to Computer Vision and Computational Photography, UC Berkeley, by Angjoo Kanazawa.
- CS 16-385: Computer Vision, CMU, by Matthew O'Toole

Additional Reading

- Multiview Geometry, Hartley & Zisserman,
  - Chapter 9 (focus on topics discussed or mentioned in the slides).
  - Chapter 10.1, 10.2 (not discussed in class, no midterm ques, but imp to understand, practical importance.)
  - Chapter 11.1, 11.2
  - Chapter 12.1, 12.2, 12.3, 12.4 (no midterm ques, but imp to understand)

Acknowledgement: some slides and material from Bernt Schiele, Mario Fritz, Michael Black, Bill Freeman, Fei-Fei Li, Justin Johnson, Serena Yeung, R. Szelisky, Ioannis Gkioulekas, Noah Snavely, Abe Davis, Kris Kitani, Xavier Giró-i-Nieto, Shree Nayar, Andreas Geiger