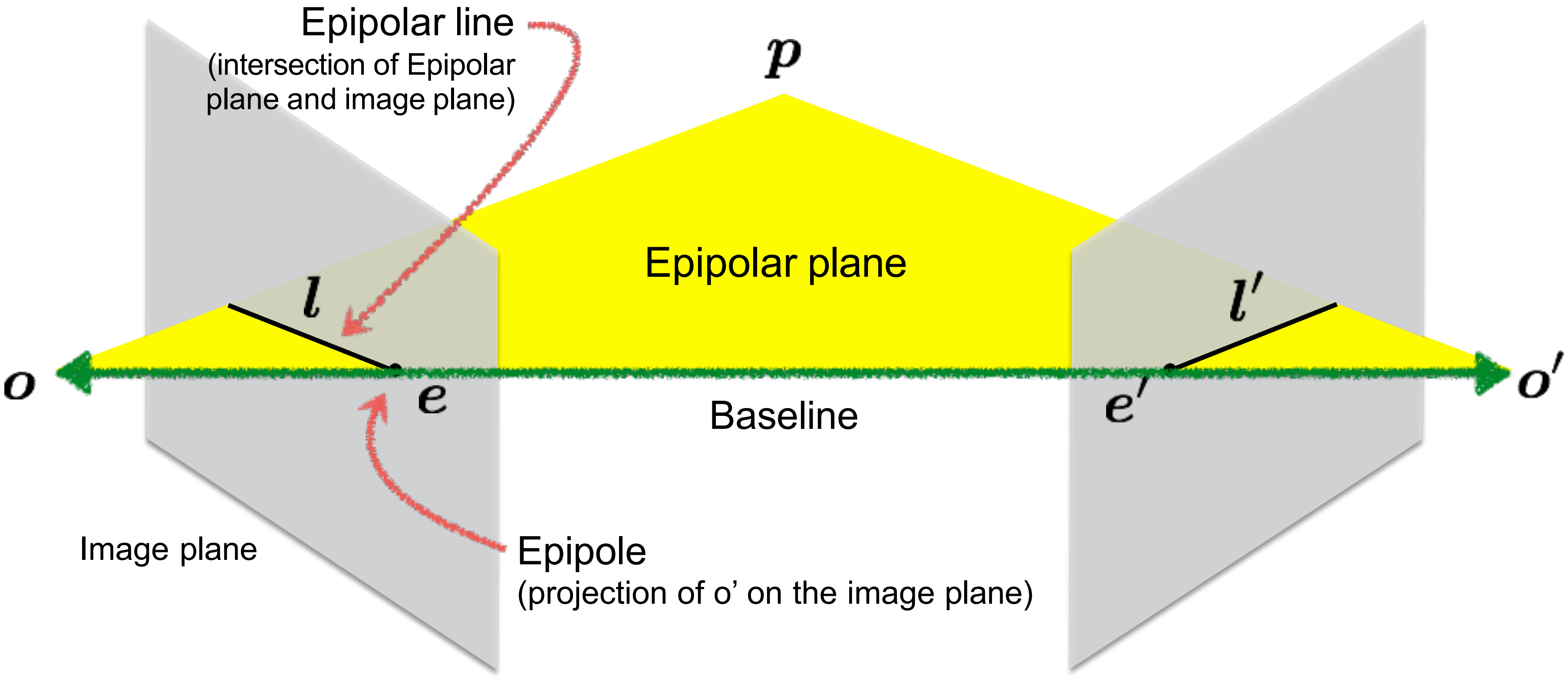
Lecture 18: Two-view Geometry

Instructor: Roni Sengupta

ULA: Andrea Dunn Beltran, William Li,
Liujie Zheng

Course Website:
Scan Me!

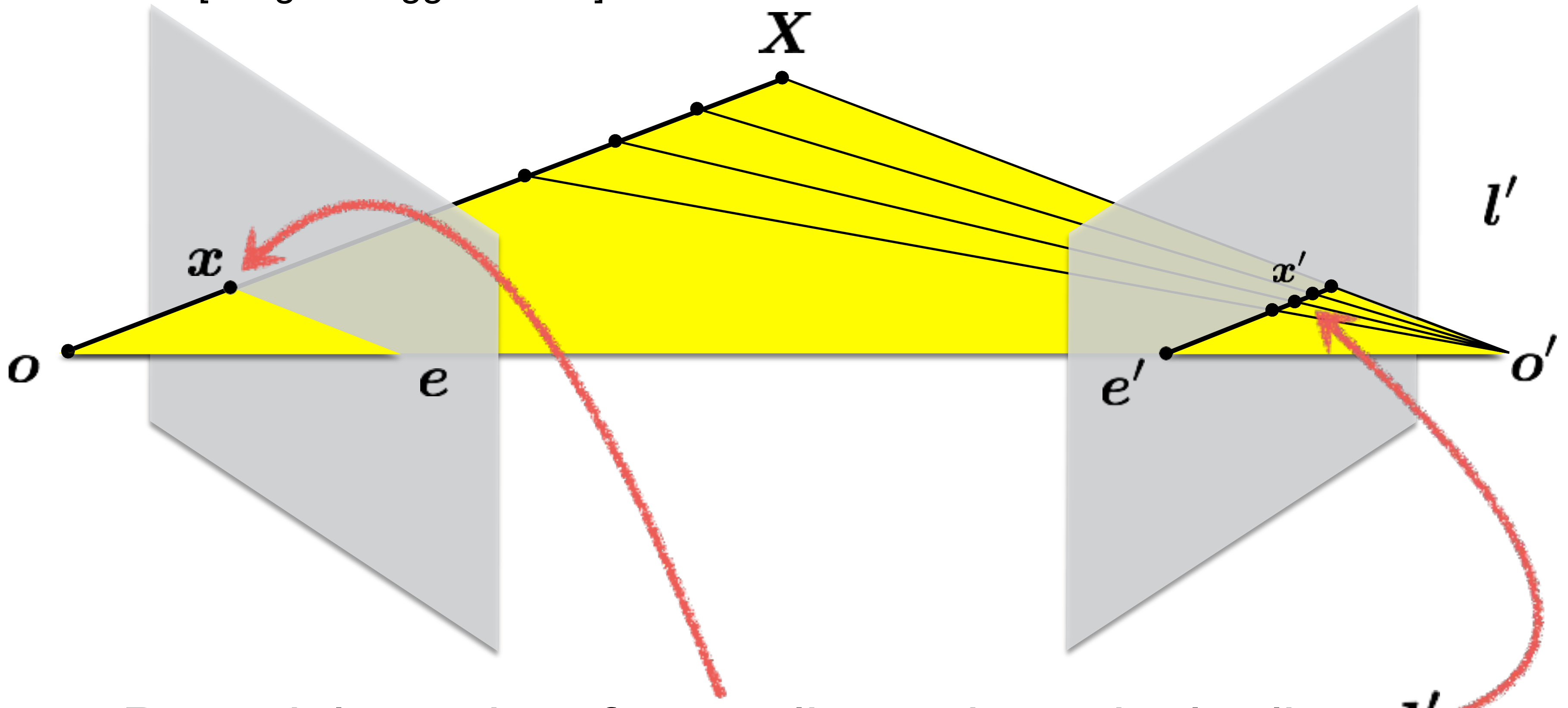
Epipolar geometry



Epipolar constraint

$$\mathbf{E}\boldsymbol{x} = \boldsymbol{l}' \rightarrow \boldsymbol{x}'^{\top}\mathbf{E}\boldsymbol{x} = 0 \rightarrow \mathbf{E} = \mathbf{R}\,[\mathbf{t}]_{\times}$$

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):

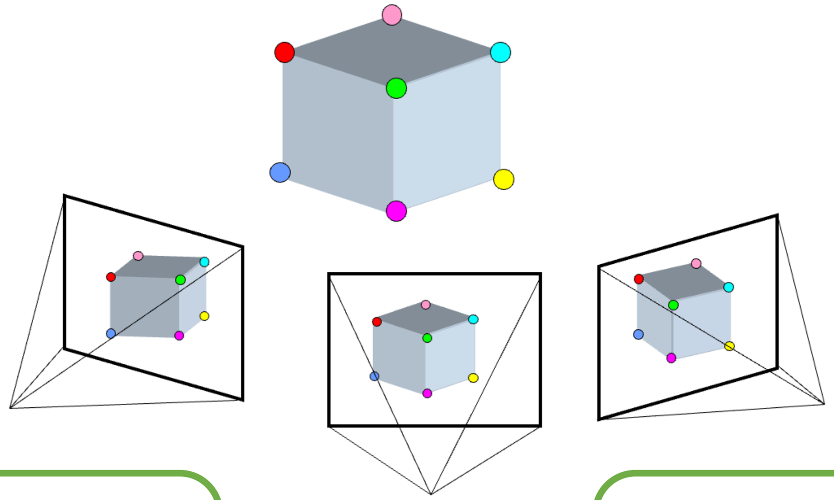
C1 at (0,0,0) with image plane at z=2

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

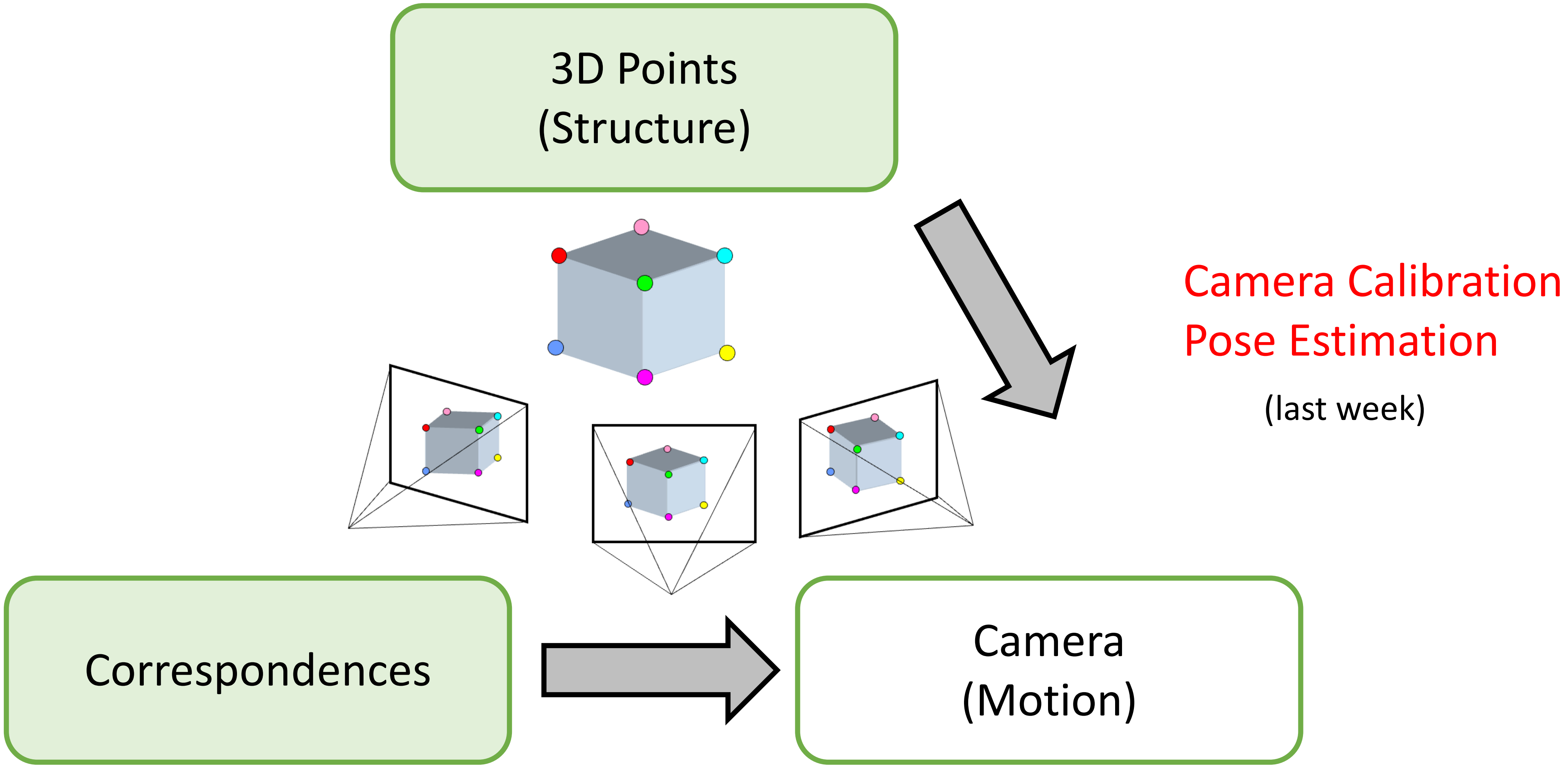
3D Points
(Structure)



Correspondences

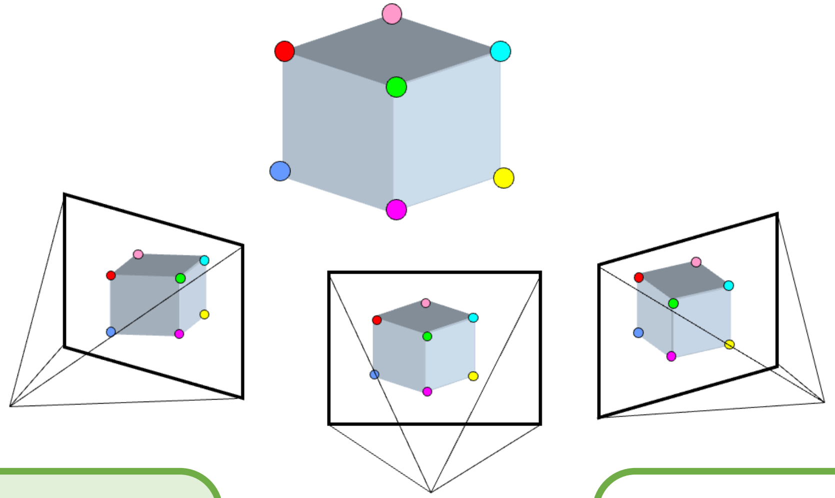
Camera
(Motion)

Big picture: 3 key components in 3D



Big picture: 3 key components in 3D

3D Points
(Structure)

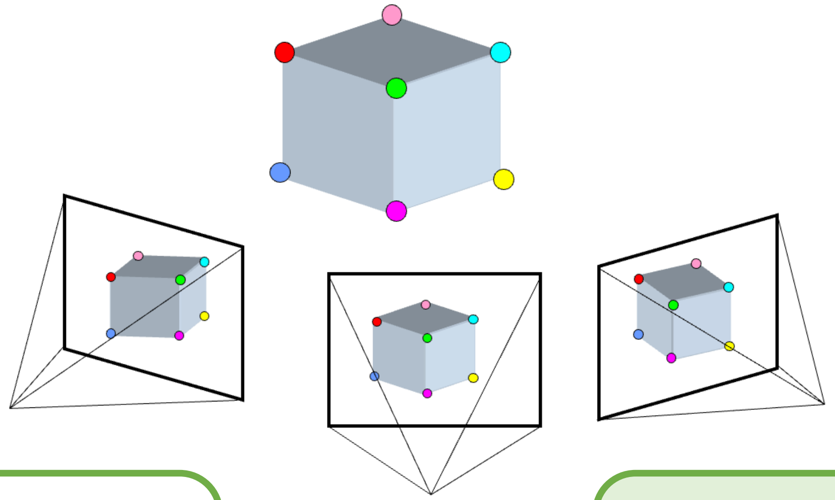


Epipolar /
Two-view
Geometry

Correspondences → Camera
(Motion)

Big picture: 3 key components in 3D

3D Points
(Structure)



Correspondences

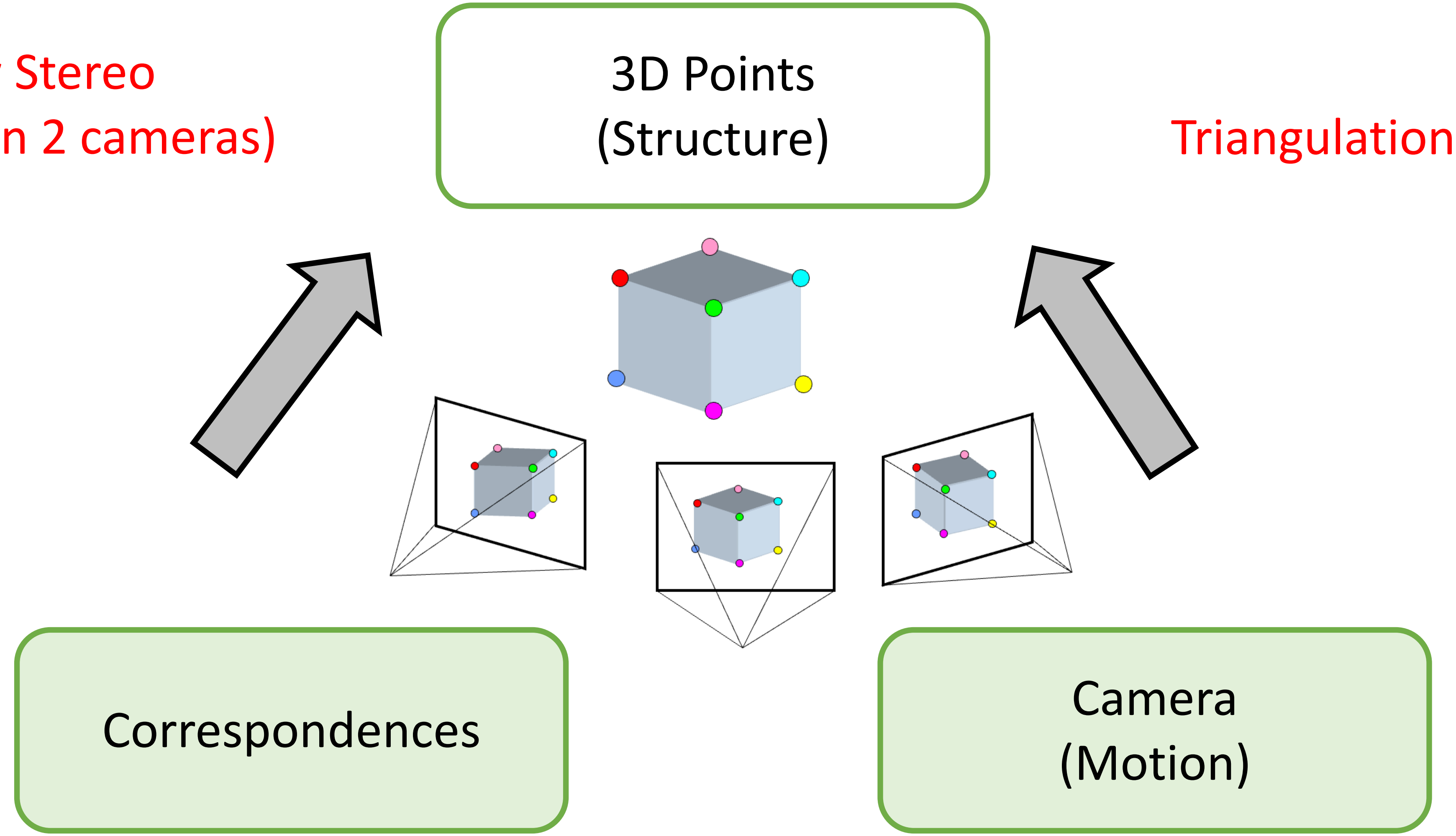
Camera
(Motion)

Stereo Matching

Big picture: 3 key components in 3D

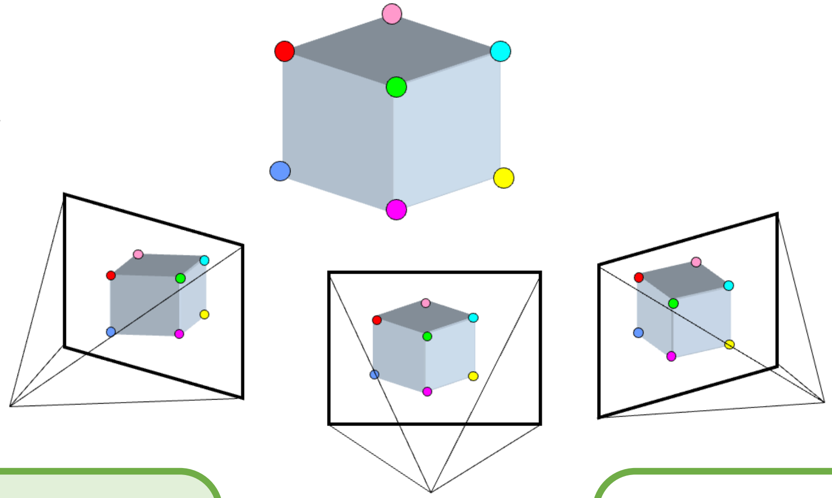
Multiview Stereo
(more than 2 cameras)

Triangulation



Big picture: 3 key components in 3D

3D Points
(Structure)



Correspondences

Camera
(Motion)

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

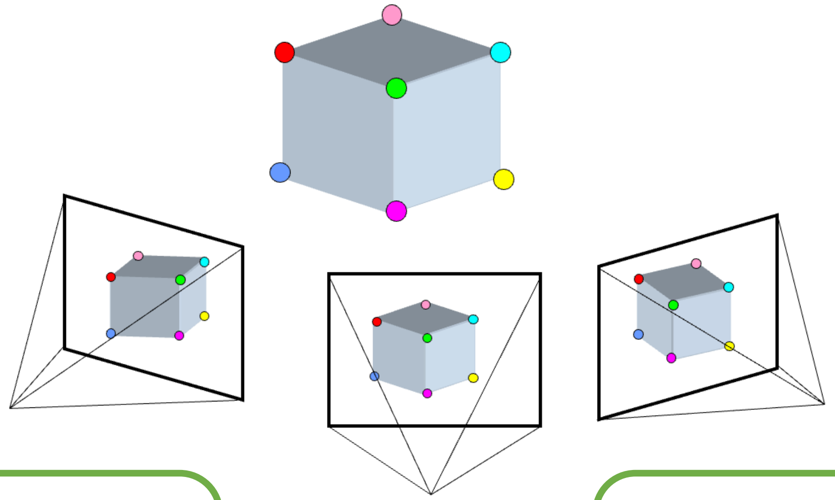
	Structure (scene geometry)	Motion (camera parameters)	**Measurements** (camera parameters)
Camera Calibration (Pose Estimation)	known	estimate	3D to 2D correspondences
Triangulation (Stereo, Multi-view Stereo)	estimate	known	2D to 2D coorespondences
Reconstruction (Structure from Motion, SLAM)	estimate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

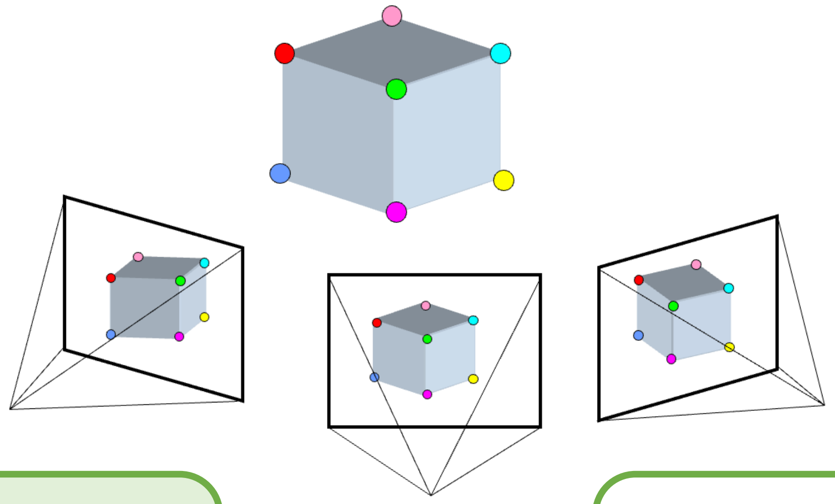


Correspondences

Camera
(Motion)

Big picture: 3 key components in 3D

3D Points
(Structure)



Epipolar /
Two-view
Geometry

Correspondences → Camera
(Motion)

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

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

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

Set up a homogeneous linear system with 9 unknowns

Hence, the 8 point algorithm!

$$\begin{bmatrix} x_1 x'_1 & x_1 y'_1 & x_1 & y_1 x'_1 & y_1 y'_1 & y_1 & x'_1 & y'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_M x'_M & x_M y'_M & x_M & y_M x'_M & y_M y'_M & y_M & x'_M & y'_M & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix} = \mathbf{0}$$

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 eight-point algorithm

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

Problem with eight-point algorithm

250906.36	183269.57	921.81	200931.10	146766.13	738.21	272.19	198.81
2692.28	131633.03	176.27	6196.73	302975.59	405.71	15.27	746.79
416374.23	871684.30	935.47	408110.89	854384.92	916.90	445.10	931.81
191183.60	171759.40	410.27	416435.62	374125.90	893.65	465.99	418.65
48988.86	30401.76	57.89	298604.57	185309.58	352.87	846.22	525.15
164786.04	546559.67	813.17	1998.37	6628.15	9.86	202.65	672.14
116407.01	2727.75	138.89	169941.27	3982.21	202.77	838.12	19.64
135384.58	75411.13	198.72	411350.03	229127.78	603.79	681.28	379.48

$$\begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \end{bmatrix} = -1$$

- Poor numerical conditioning
- Can be fixed by rescaling the data

Problem with 8-point algorithm

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

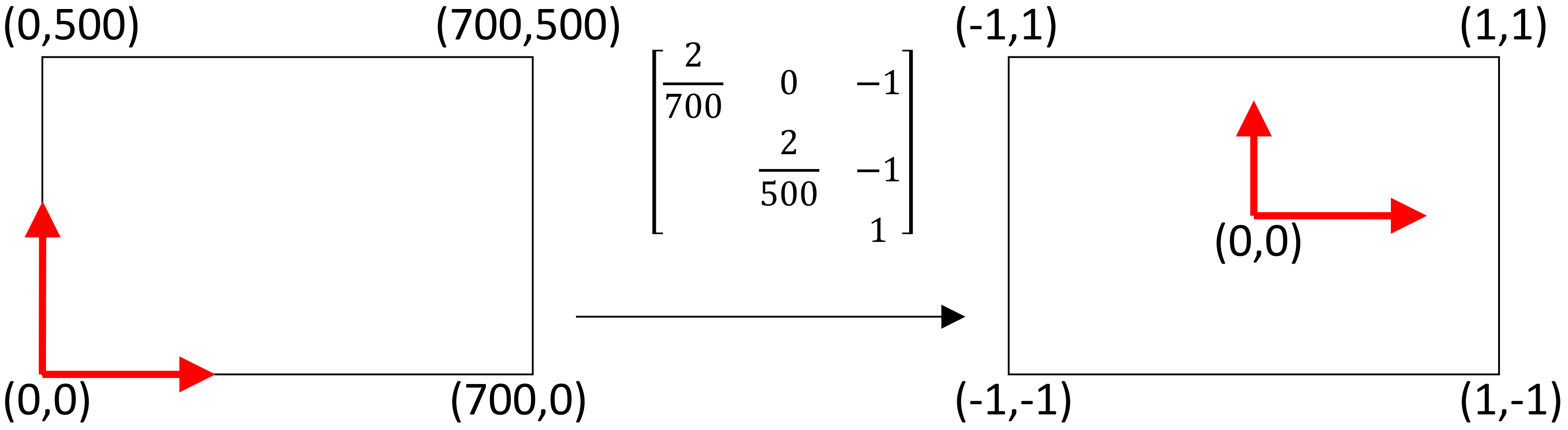
~10000 ~10000 ~100 ~10000 ~10000 ~100 ~100 ~100 1

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]



$$\begin{bmatrix} \frac{2}{700} & 0 & -1 \\ & \frac{2}{500} & -1 \\ & & 1 \end{bmatrix}$$

Normalized 8-point algorithm

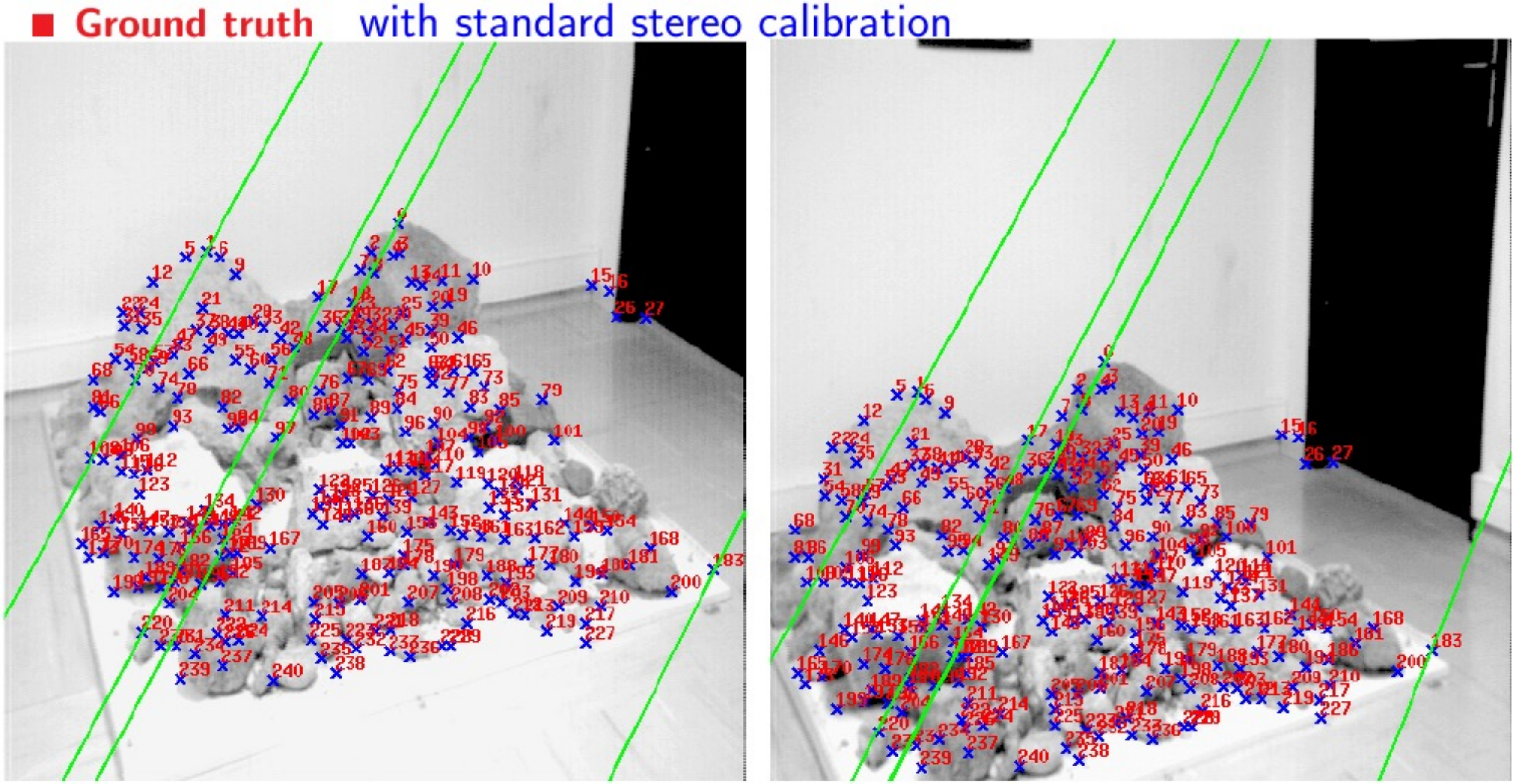
- Transform input by $\hat{\mathbf{x}}_\mathbf{i} = \mathbf{T}\mathbf{x}_\mathbf{i}$, $\hat{\mathbf{x}}'_\mathbf{i} = \mathbf{T}\mathbf{x}'_\mathbf{i}$
- Call 8-point on $\hat{\mathbf{x}}_\mathbf{i}, \hat{\mathbf{x}}'_\mathbf{i}$ to obtain $\hat{\mathbf{F}}$
- $\mathbf{F} = \mathbf{T}'^{\mathrm{T}}\hat{\mathbf{F}}\mathbf{T}$

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

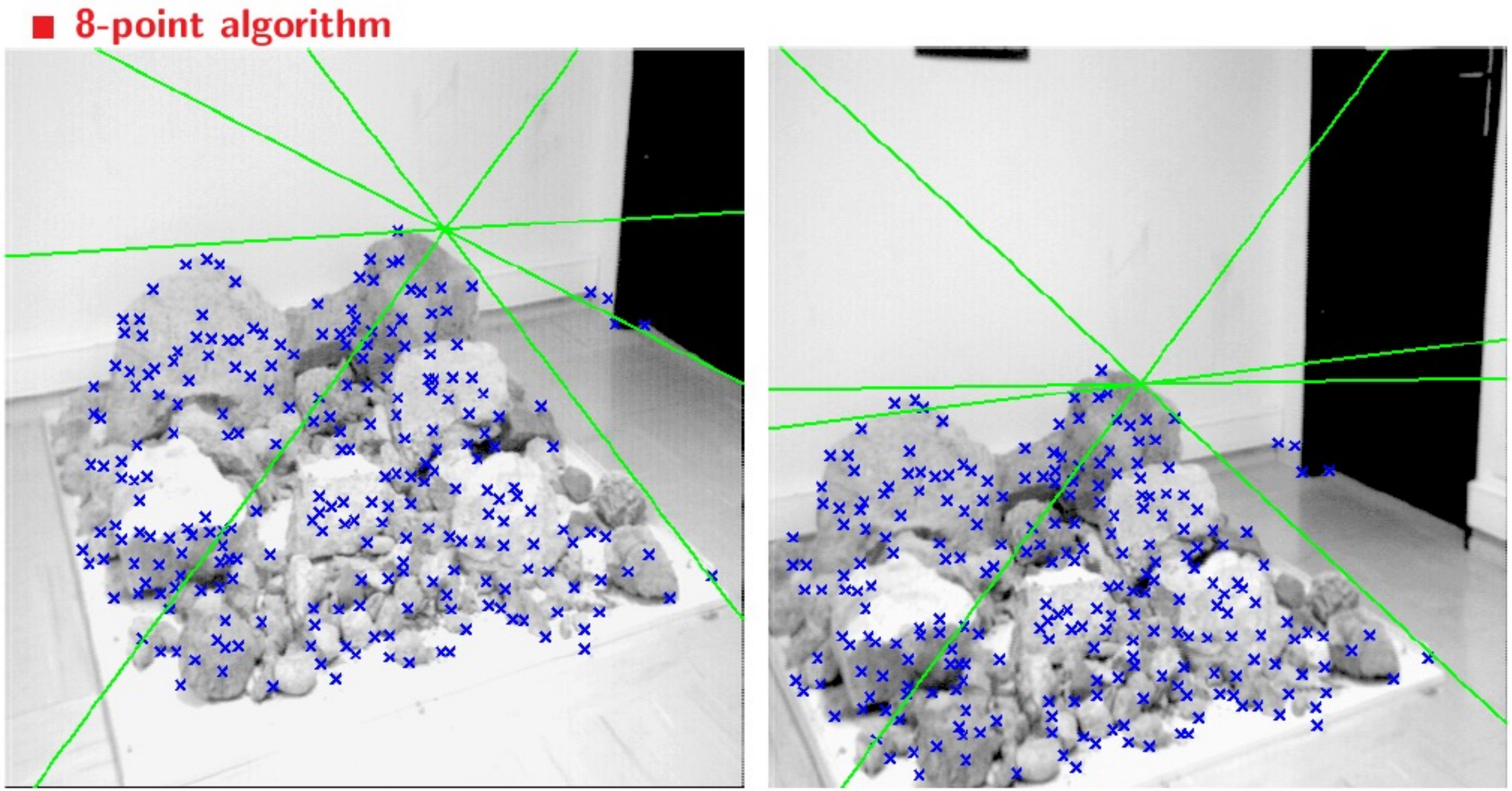
$$\hat{\mathbf{x}}'^{\mathrm{T}}\mathbf{T}'^{-\mathrm{T}}\mathbf{F}\mathbf{T}^{-1}\hat{\mathbf{x}} = 0$$

$\hat{\mathbf{F}}$ Fundamental matrix of normalized camera coordinate

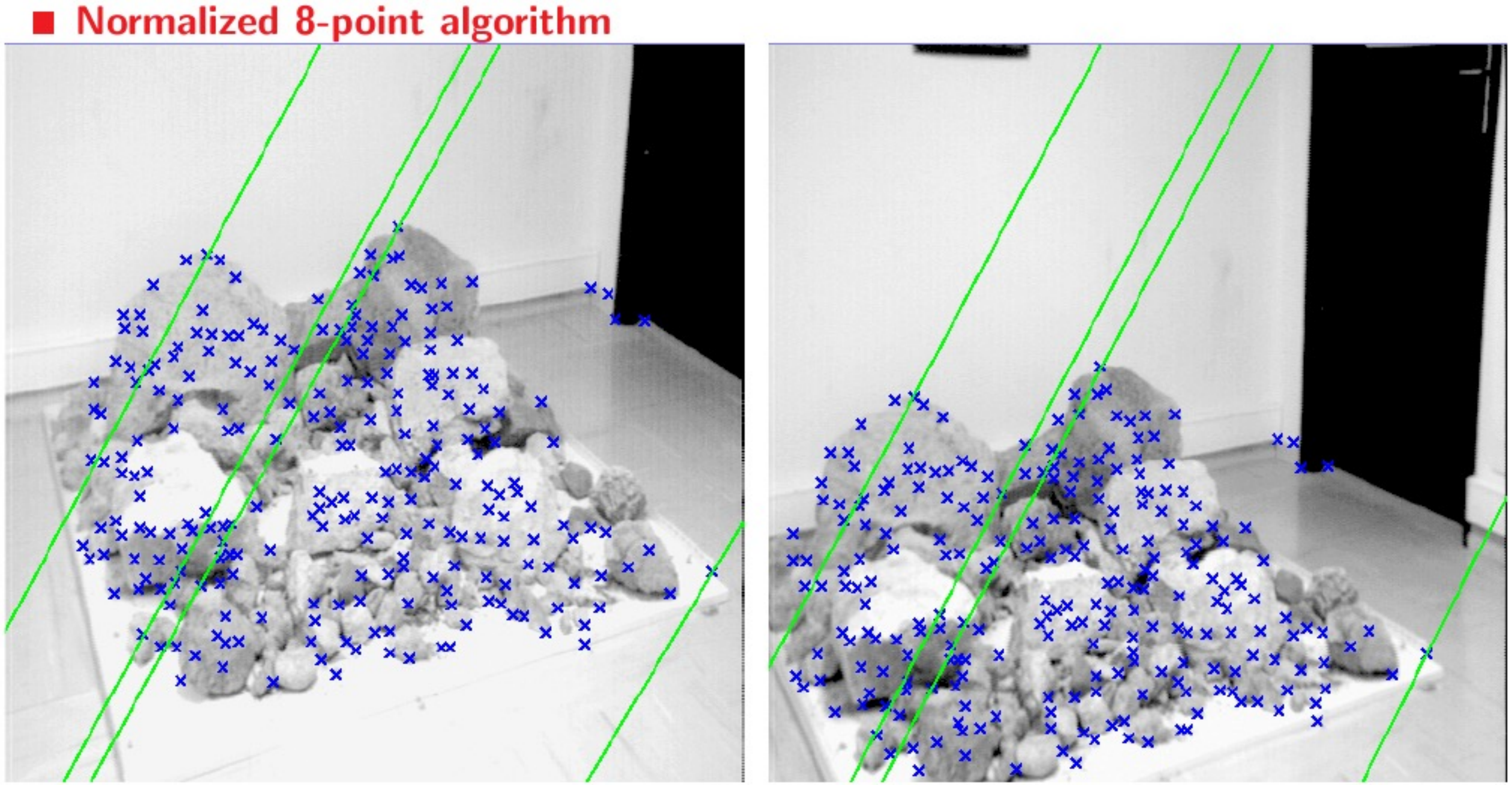
Results (ground truth)



Results (8 point algorithm)



Results (normalized 8-point algorithm)



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!

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,

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

Solution: Compute the singular value decomposition of F,

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

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

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

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

(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)

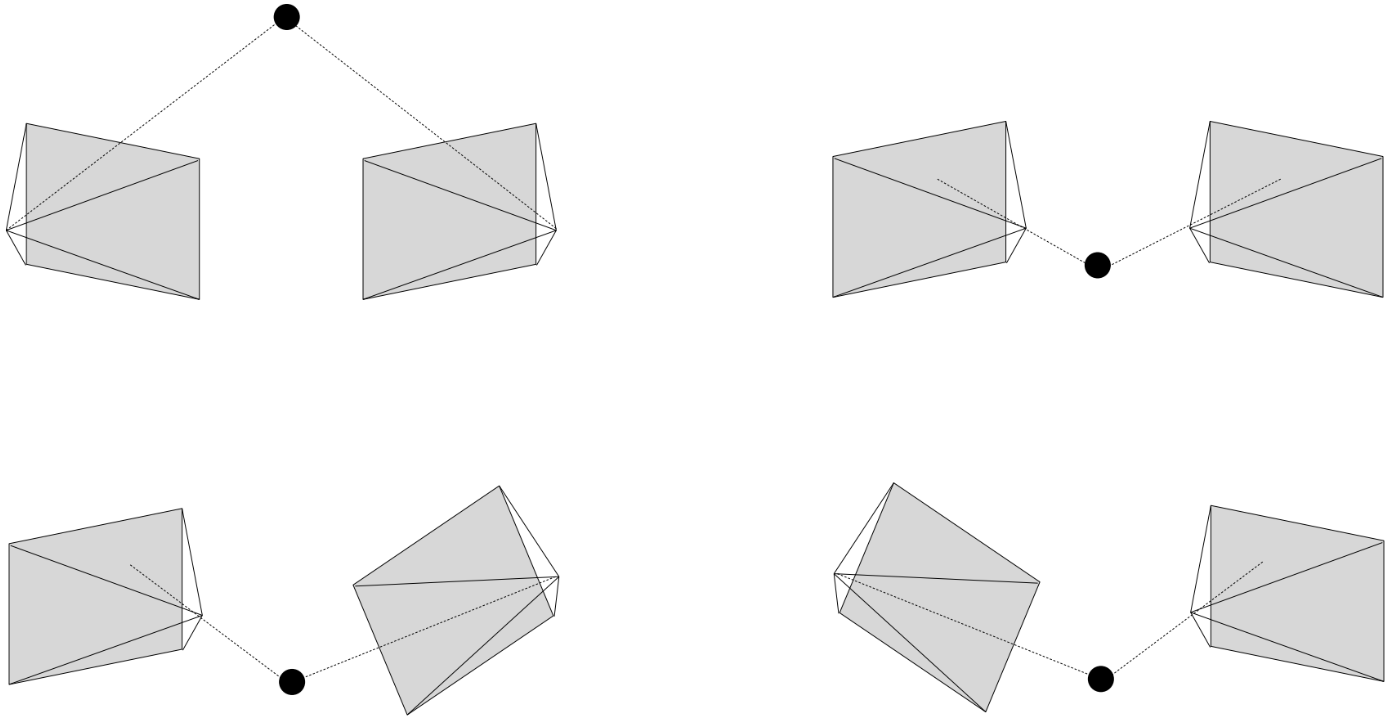
5. (Un-normalize F)

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 -> only 1 case where reconstructed 3D pt is in front of both cameras.
  - 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.

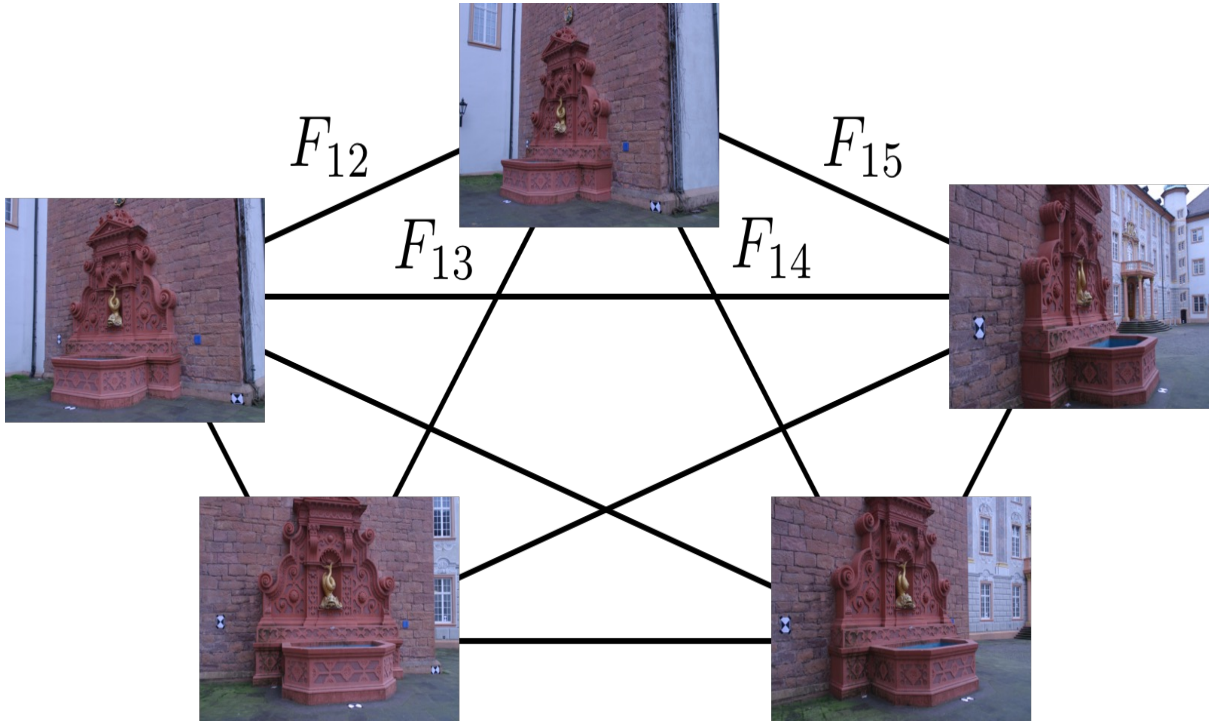


What about more than two views?

- 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.

Necessary but not sufficient



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

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

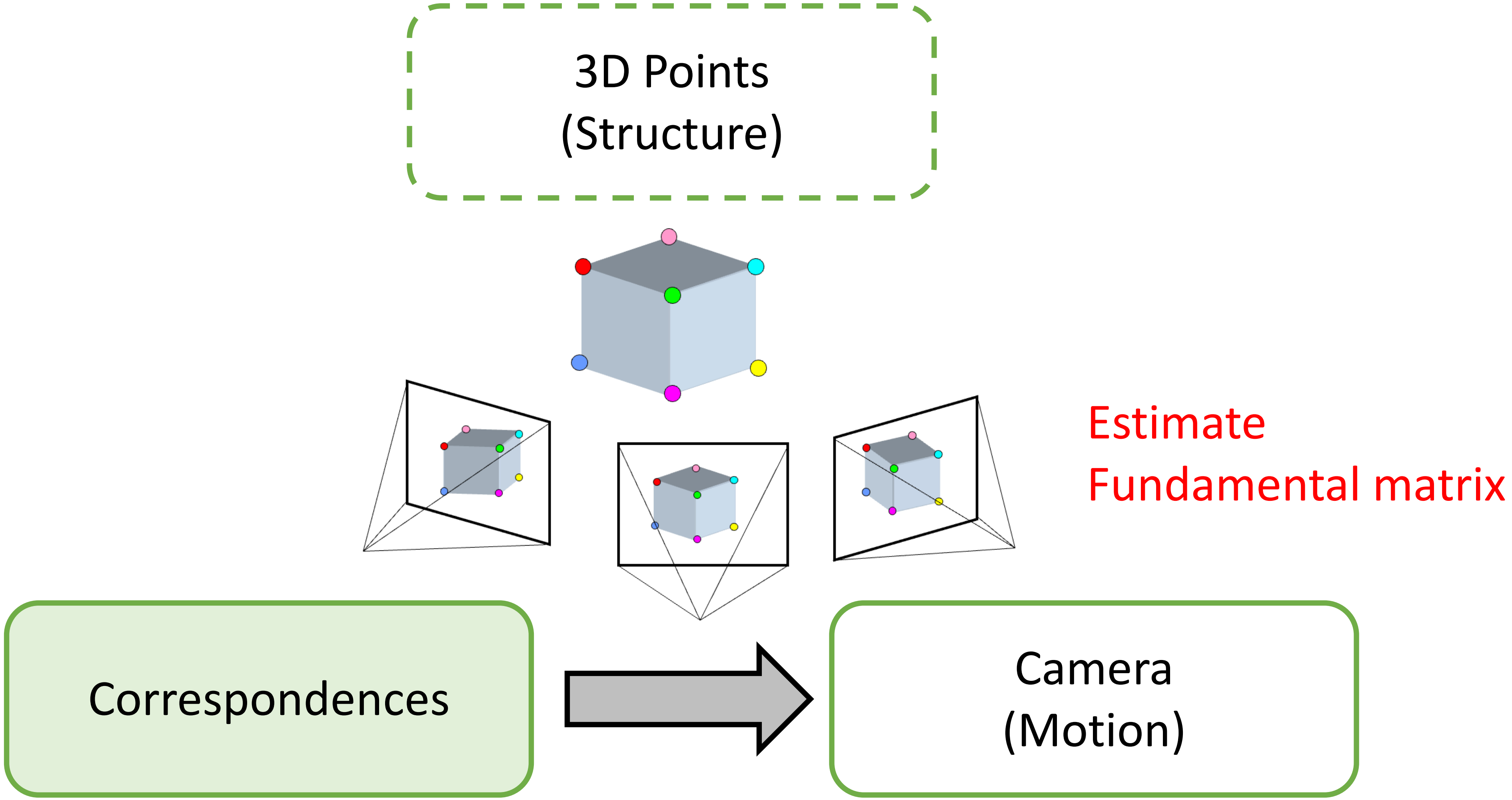
In case of all collinear cameras : $rank(A) \leq 2$ and $rank(F) \leq 4$

The Fundamental Matrix Song

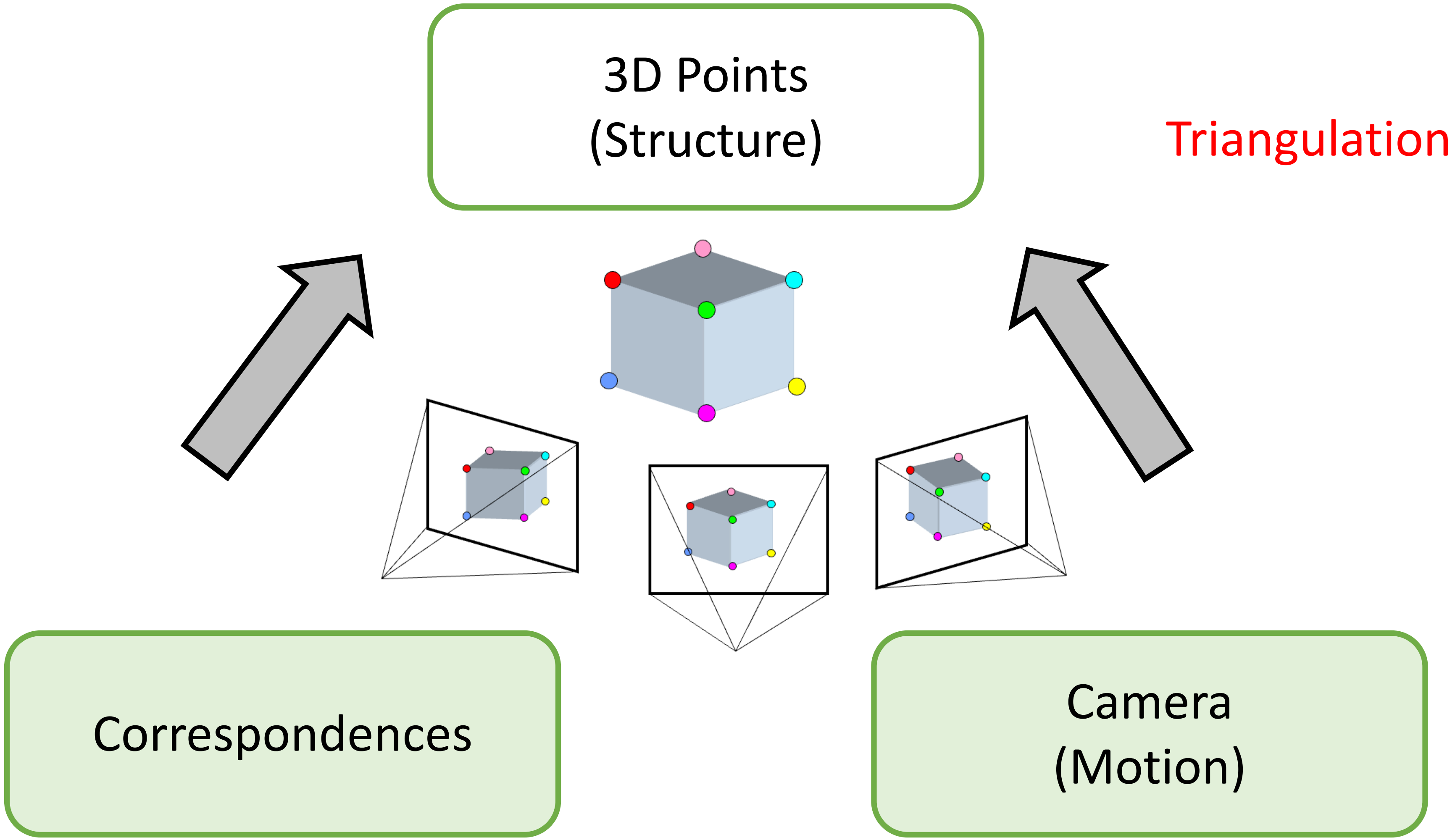
Today’s class

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

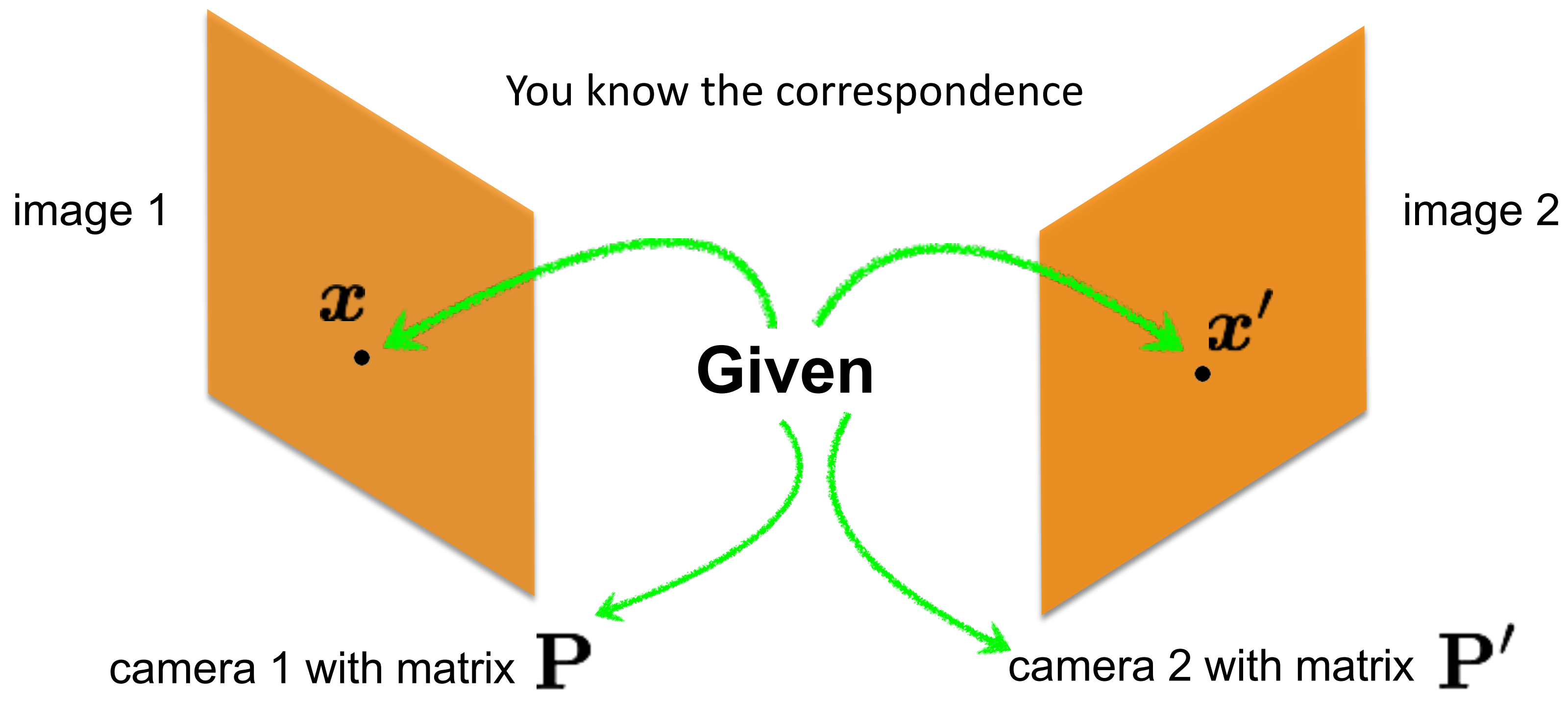
Big picture: 3 key components in 3D



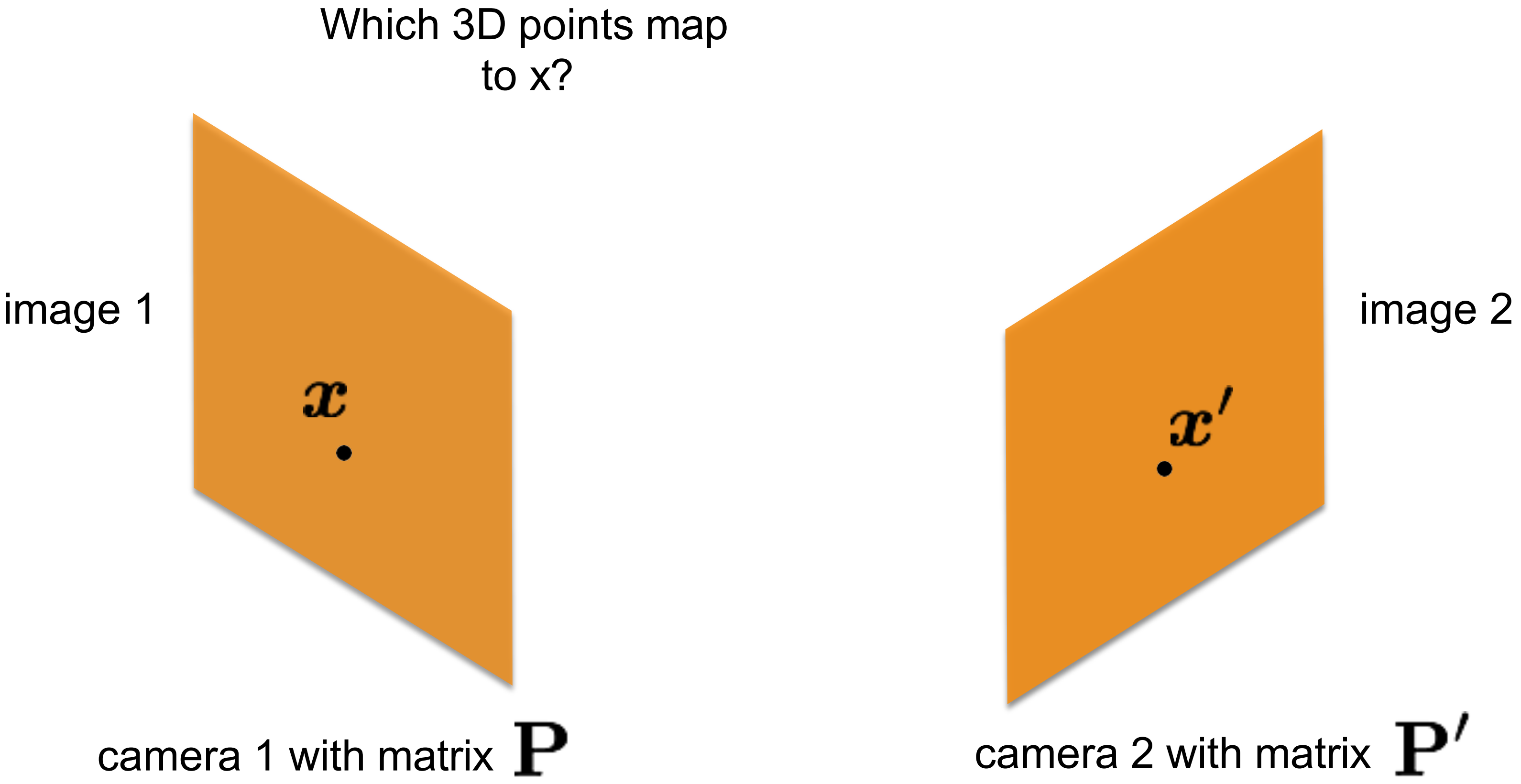
Big picture: 3 key components in 3D



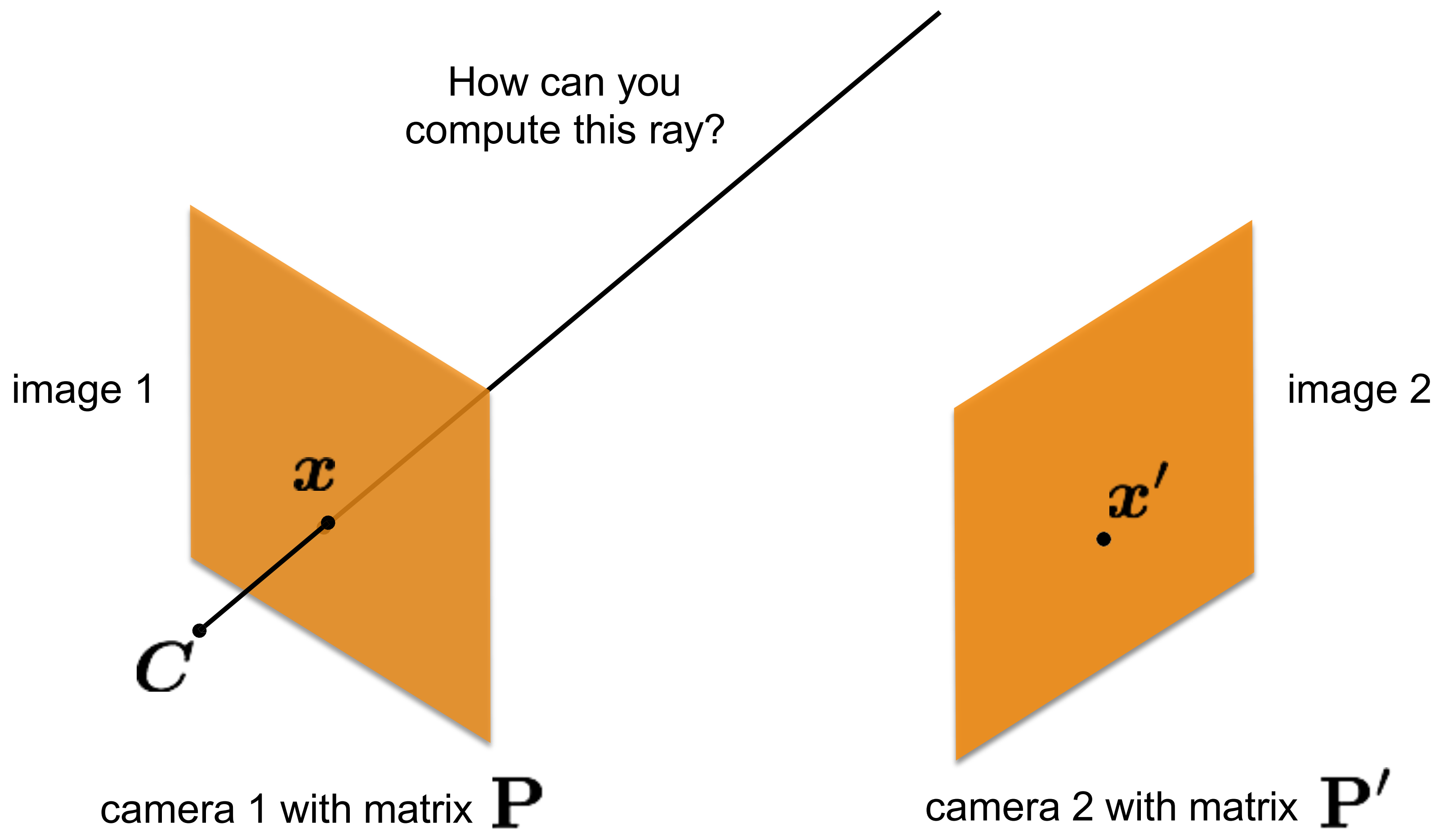
Triangulation



Triangulation



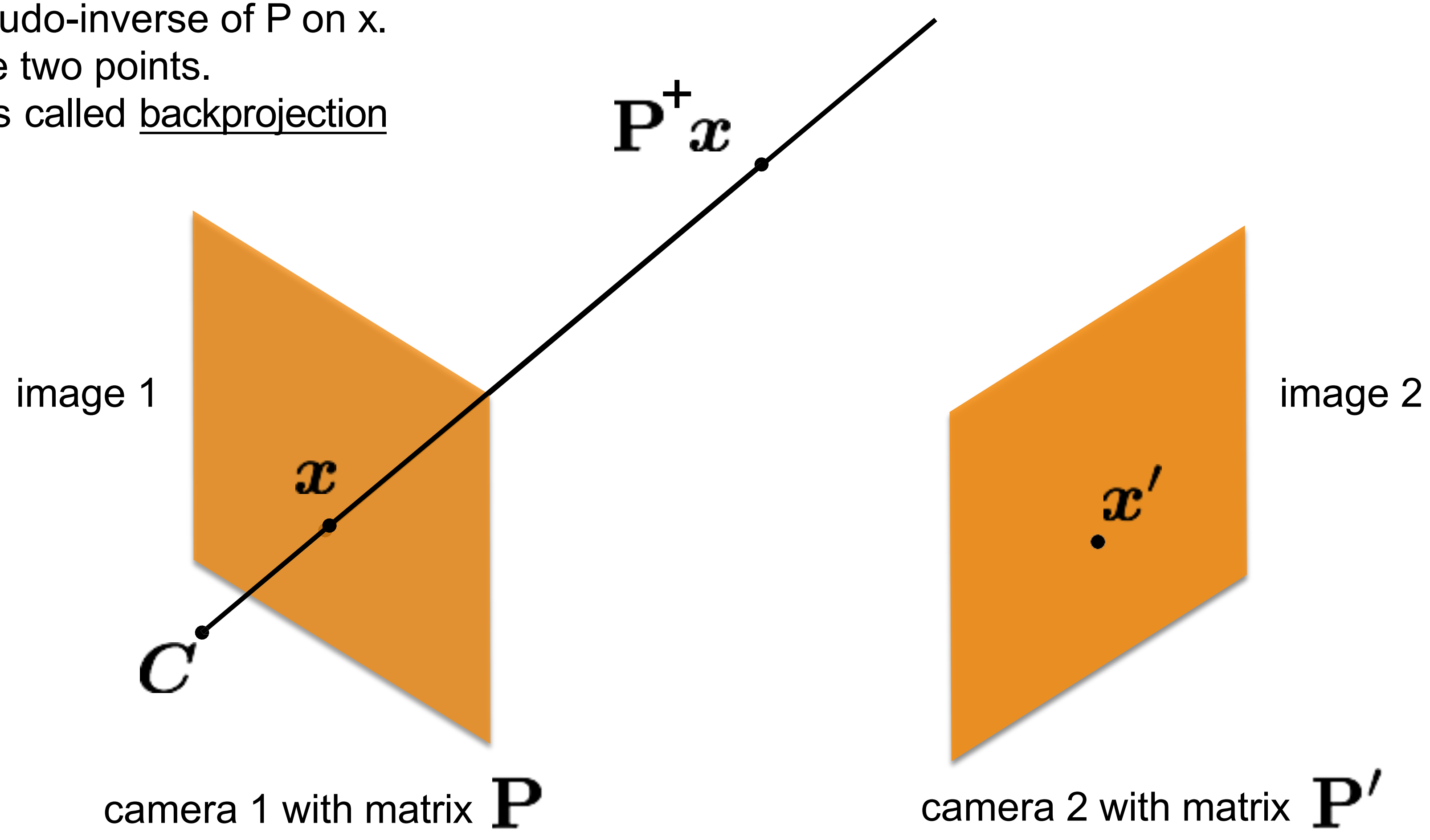
Triangulation



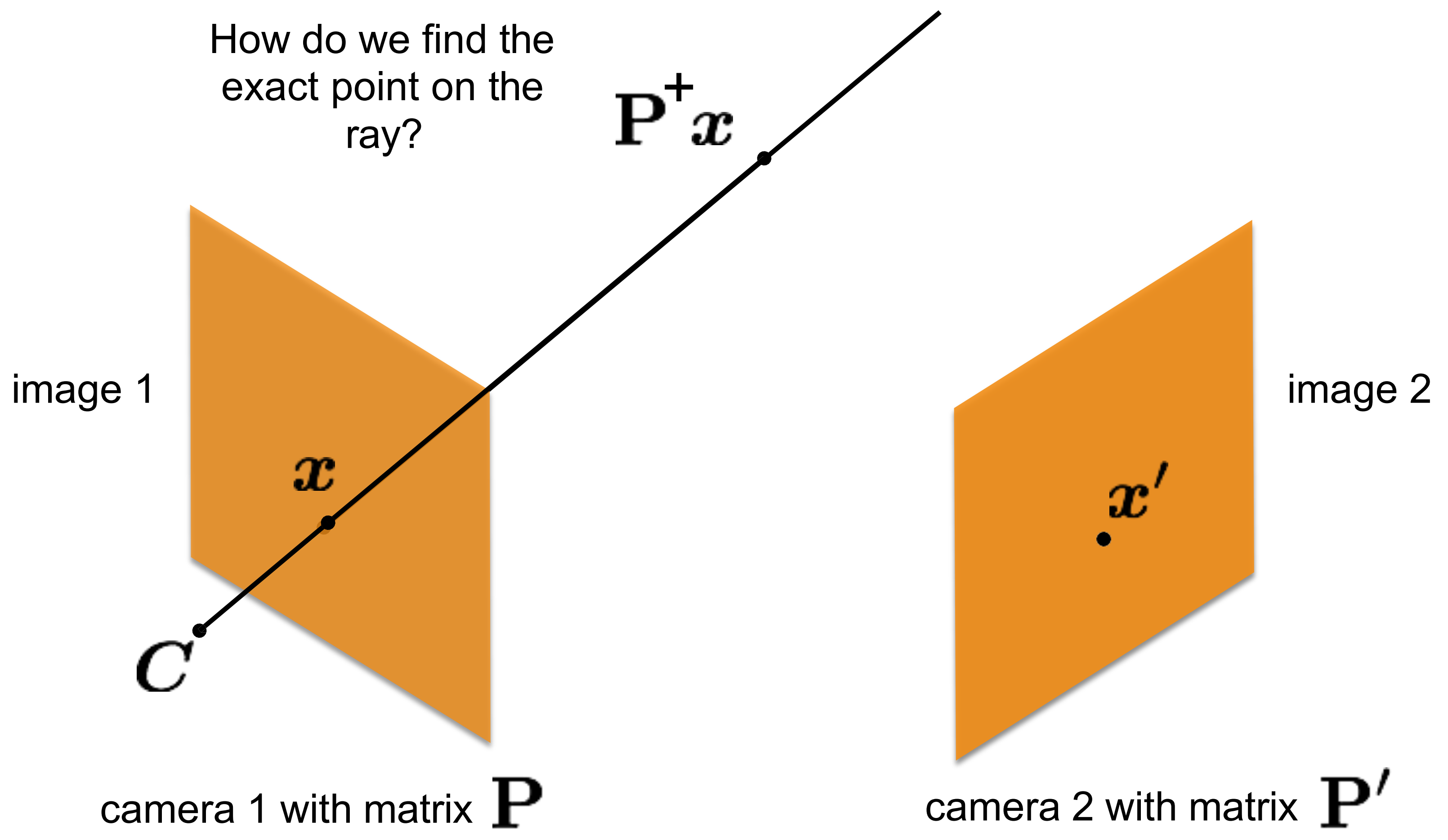
Triangulation

Create two points on the ray:
1) find the camera center; and
2) apply the pseudo-inverse of P on x.

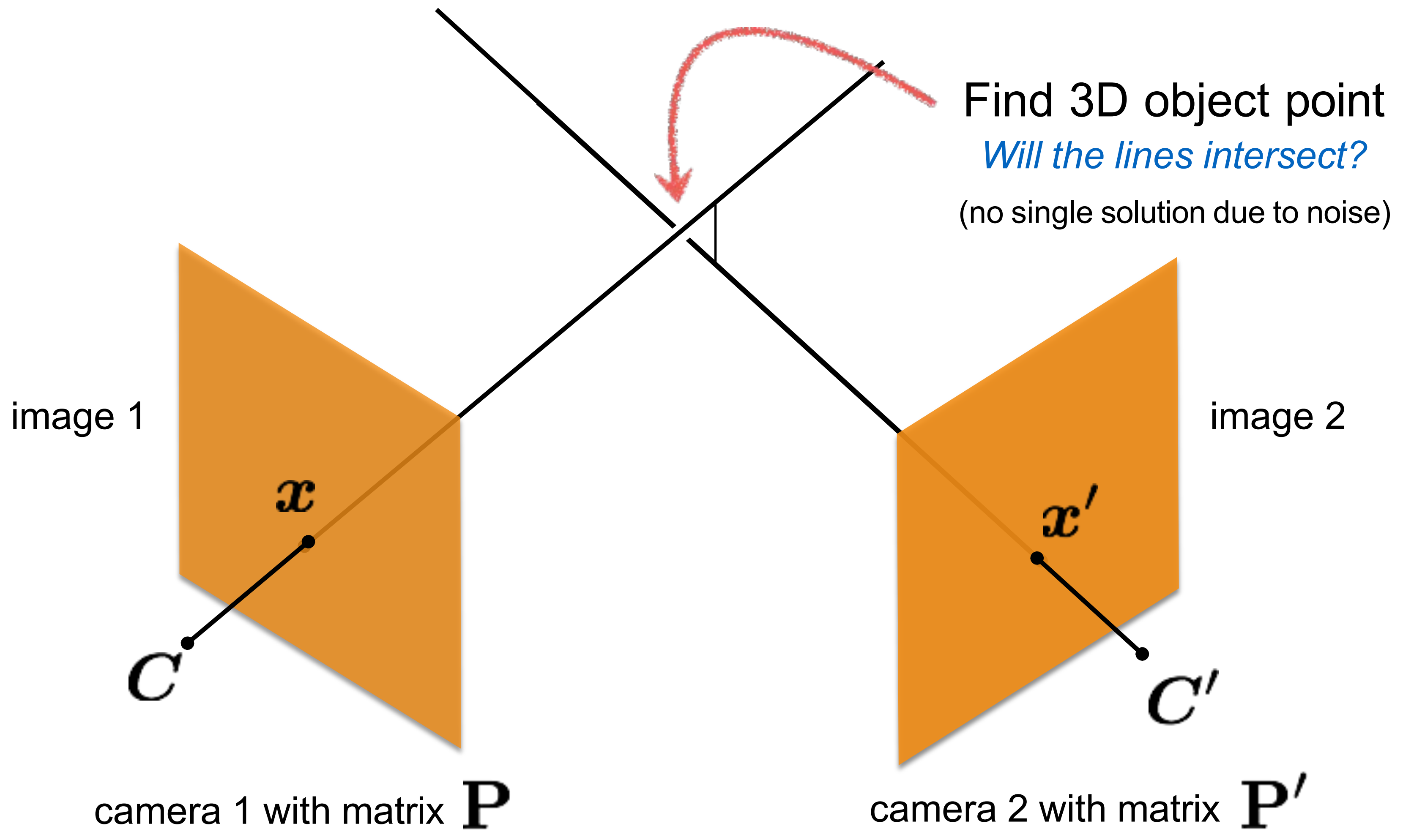
Then connect the two points.
This procedure is called <u>backprojection</u>



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)

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

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

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

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

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

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

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

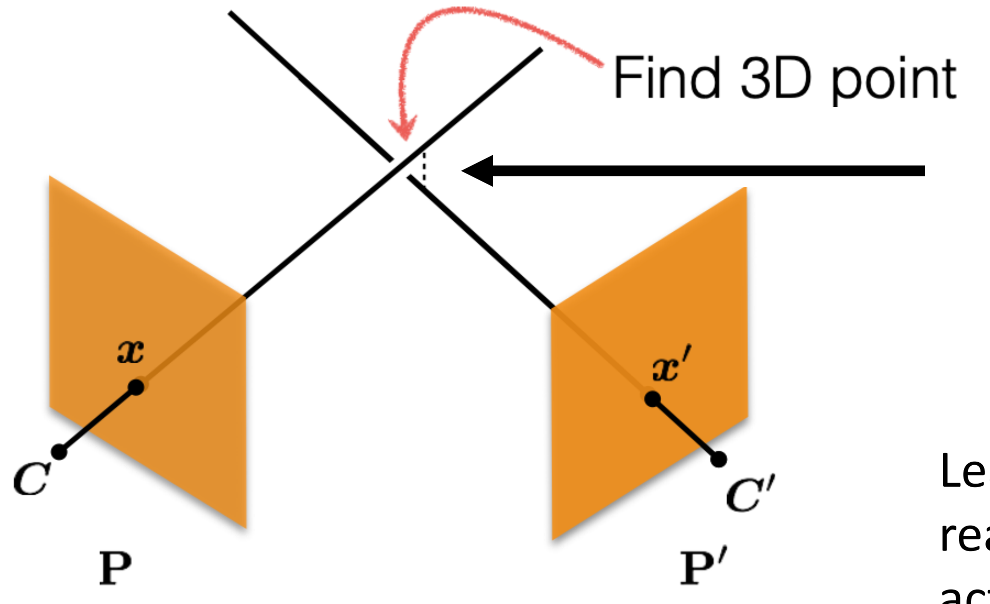
sanity check! dimensions?

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

How do we solve homogeneous linear system?

SVD!

Triangulation Disclaimer: Noise



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

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

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

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" (more than 2 image case, next lecture..)

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)