

Spring 2022

INTRODUCTION TO COMPUTER VISION

Atlas Wang

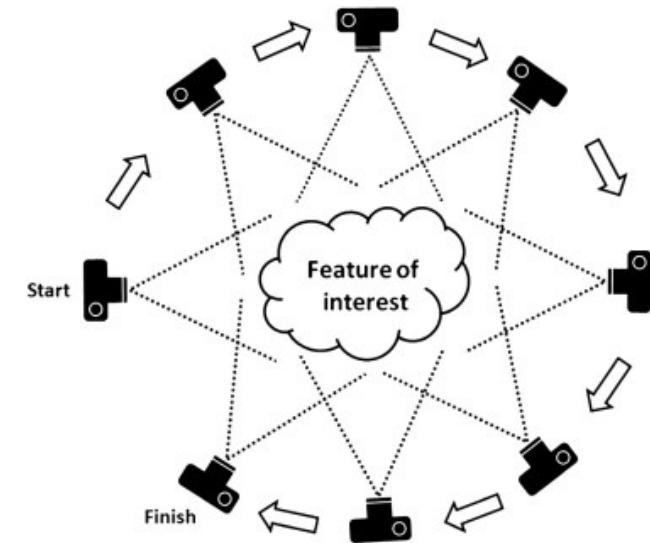
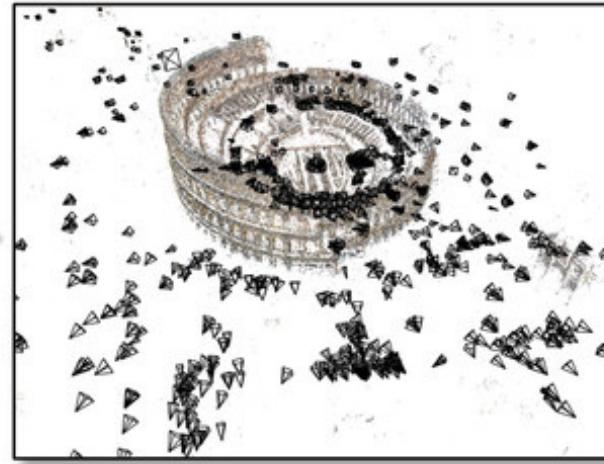
Assistant Professor, The University of Texas at Austin

Visual Informatics Group@UT Austin
<https://vita-group.github.io/>

Many slides here were adapted from CMU 16-385 & Brown CSCI 1430

“Structure from Motion”

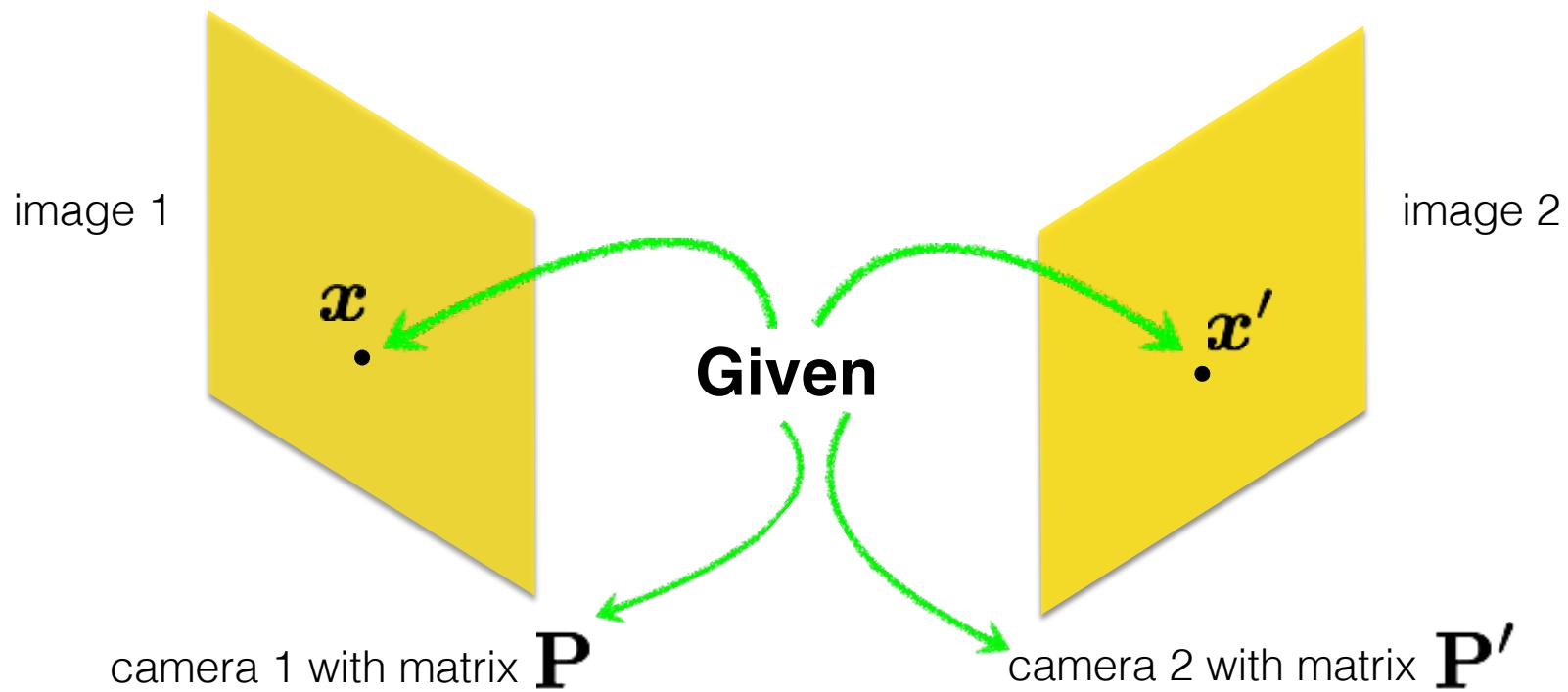
- Humans perceive the 3D structure in their environment by moving around it
 - When the observer moves, objects around them move different amounts depending on their distance from the observer.



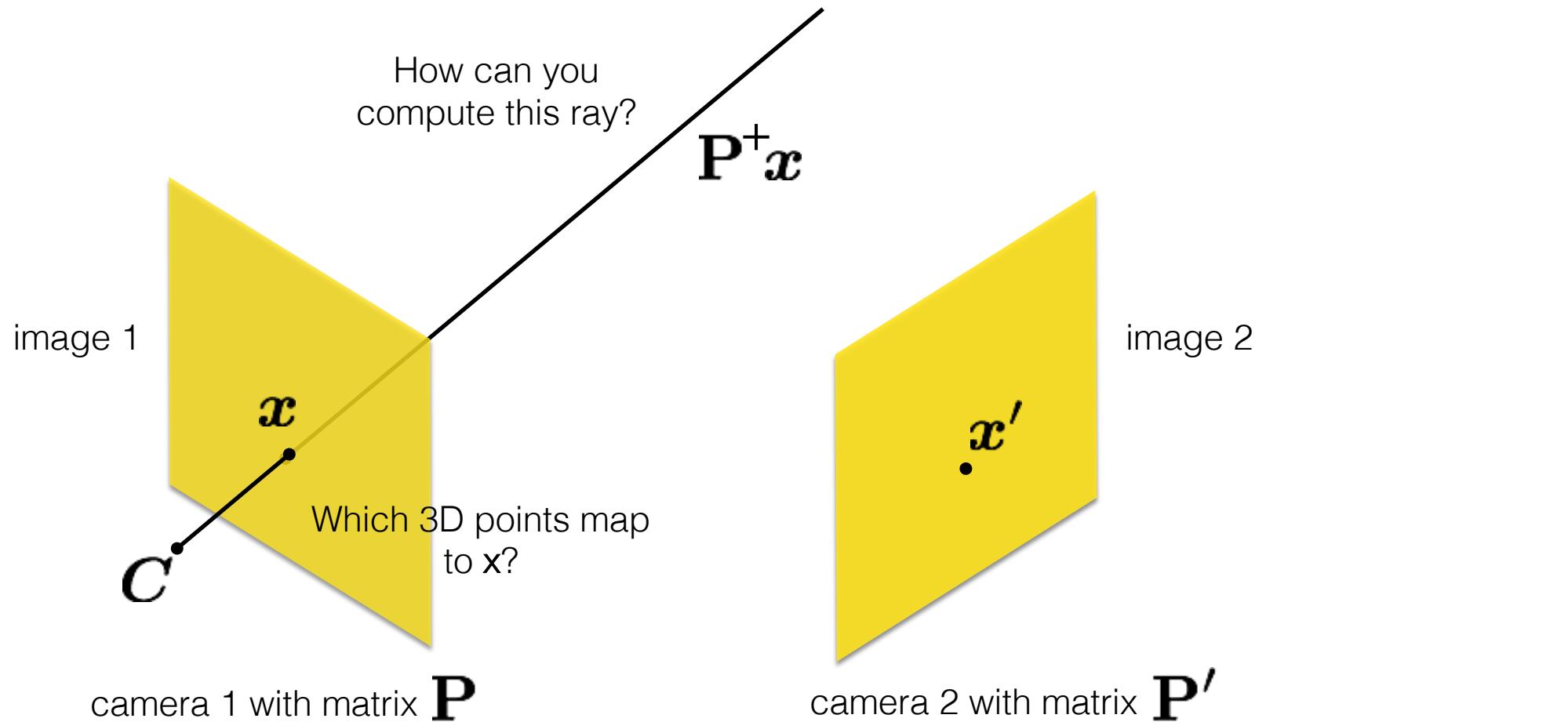
- Finding structure from motion presents a similar problem in stereo vision.
 - Estimating three-dimensional structures from two-dimensional image sequences that may be coupled with local motion signals
 - Correspondence between images and reconstruction of 3D object needs to be found

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

Triangulation (Two-view geometry)

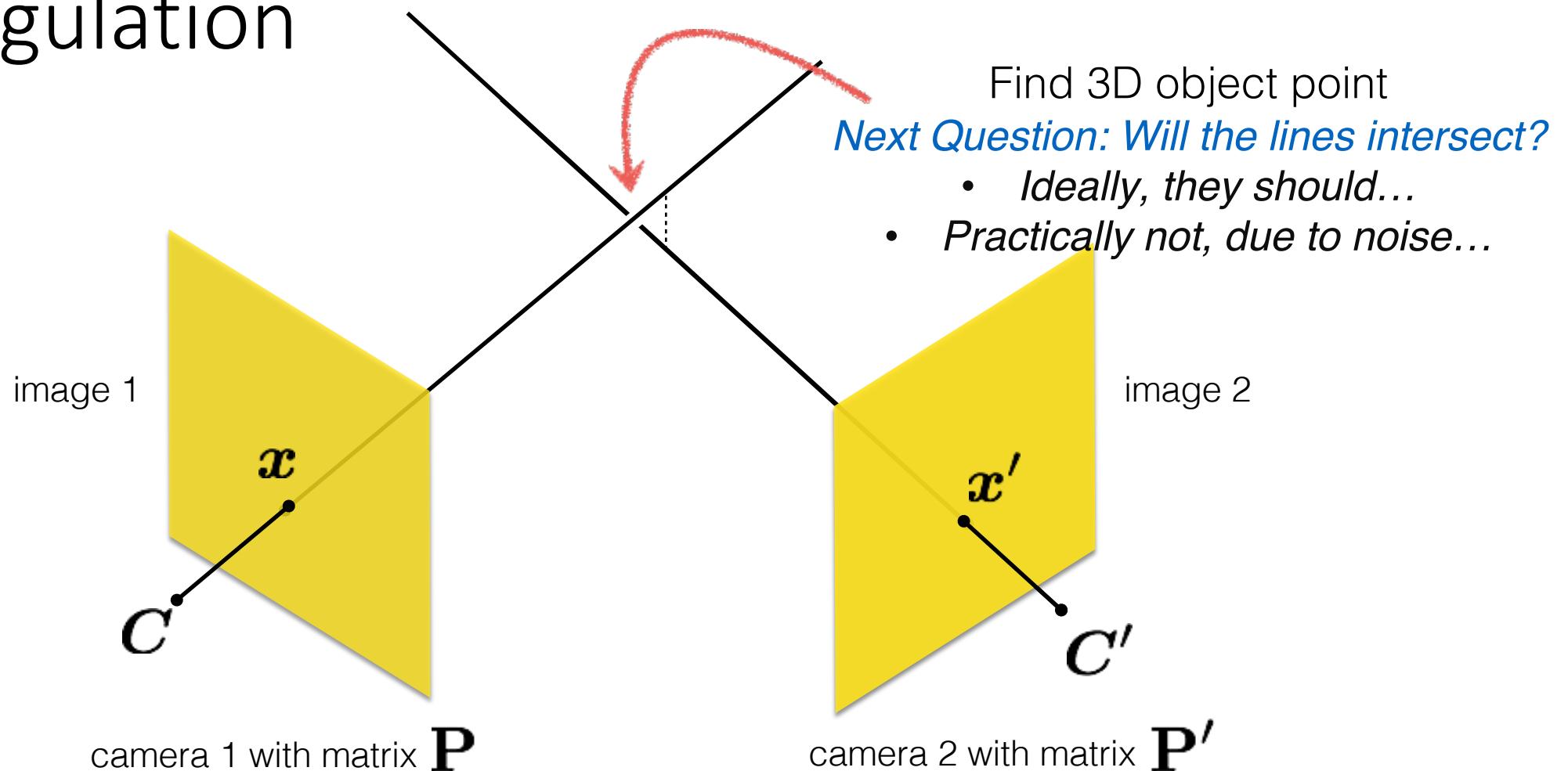


Triangulation



Create two points on the ray:
1) find the camera center; and
2) apply the pseudo-inverse of \mathbf{P} on x .
Then connect the two points.
This procedure is called backprojection

Triangulation



Triangulation

Given a set of (noisy) matched points

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

and camera matrices

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

Estimate the 3D point

$$\mathbf{X}$$

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

known known

Can we compute \mathbf{X} from a single correspondence \mathbf{x} ?

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

(homogeneous
coordinate)

This is a similarity relation because it involves homogeneous coordinates

Question: why
not directly using
homogenous
coordinate here?

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

(heterogeneous
coordinate)

Same ray direction but differs by a scale factor

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

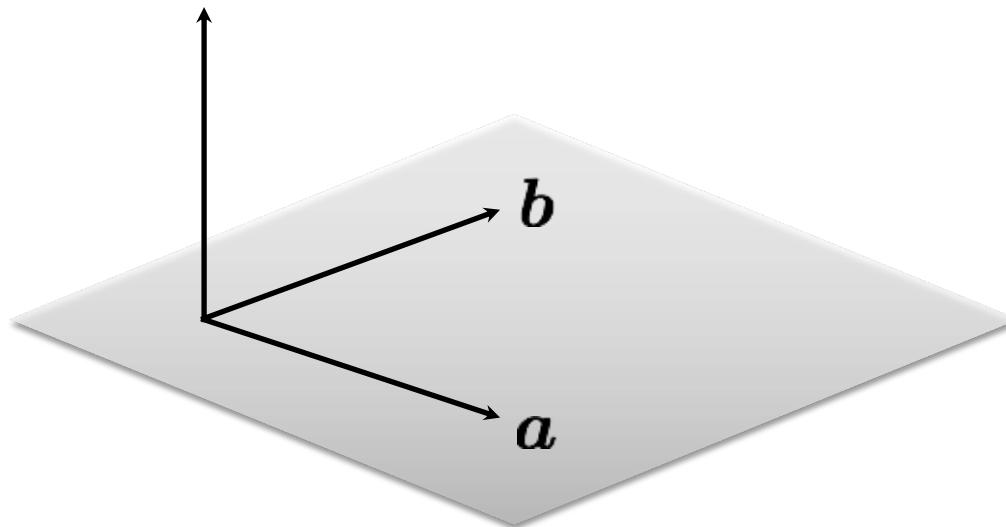
How do we solve for unknowns in a similarity relation?

Linear algebra reminder: cross product

Vector (cross) product

takes two vectors and returns a vector perpendicular to both

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



$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{bmatrix}$$

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

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

remember this!!!

$$\mathbf{c} \cdot \mathbf{a} = 0$$

$$\mathbf{c} \cdot \mathbf{b} = 0$$

Linear algebra reminder: cross product

Cross product

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

Can also be written as a matrix multiplication

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

Skew symmetric

Back to triangulation

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

Same direction but differs by a scale factor

How can we rewrite this using vector products?

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

Same direction but differs by a scale factor

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

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

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

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

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

Using the fact that the cross product should be zero

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

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

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 equations

Using the fact that the cross product should be zero

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

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

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

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

Remove third row, and
rearrange as system on
unknowns

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

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

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

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

sanity check! dimensions?

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

How do we solve homogeneous linear system?

Concatenate the 2D points from both images

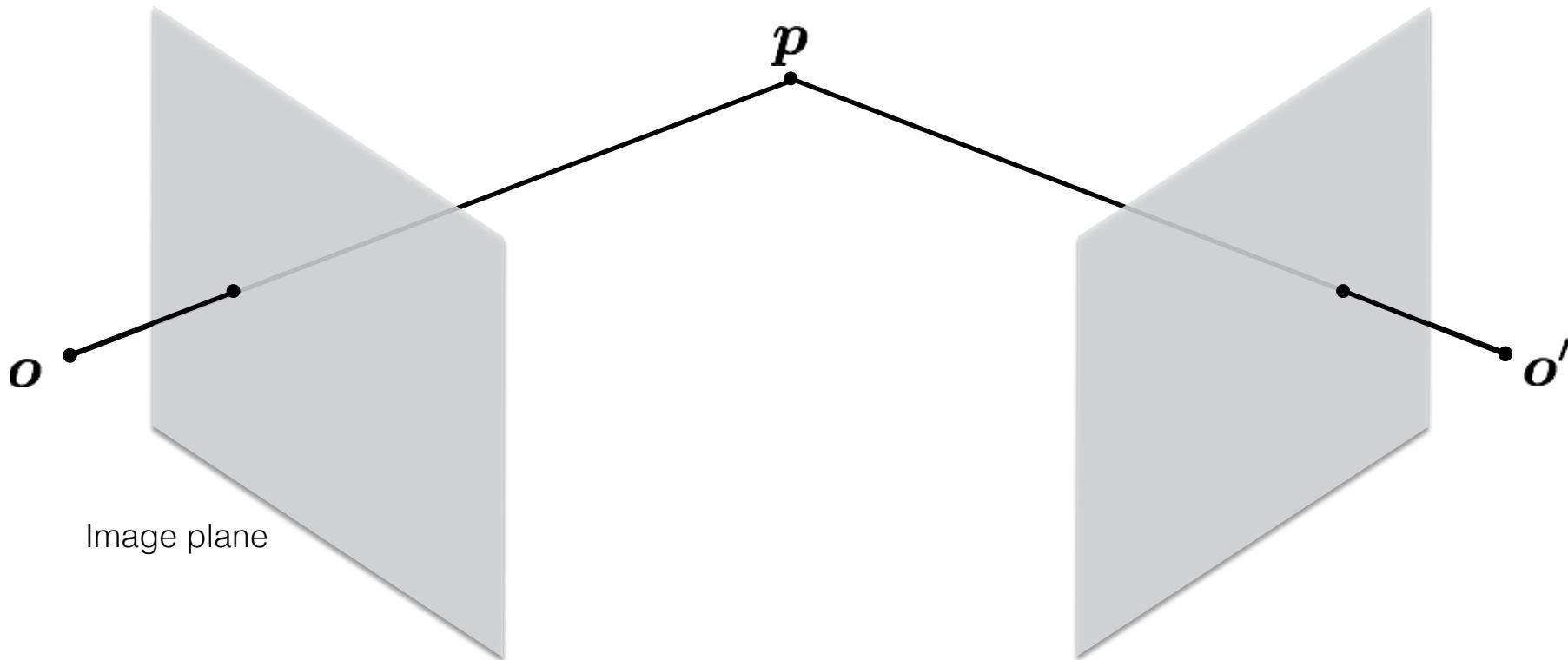
$$\begin{bmatrix} y\mathbf{p}_3^\top - \mathbf{p}_2^\top \\ \mathbf{p}_1^\top - x\mathbf{p}_3^\top \\ y'\mathbf{p}'_3^\top - \mathbf{p}'_2^\top \\ \mathbf{p}'_1^\top - x'\mathbf{p}'_3^\top \end{bmatrix} \mathbf{X} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

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

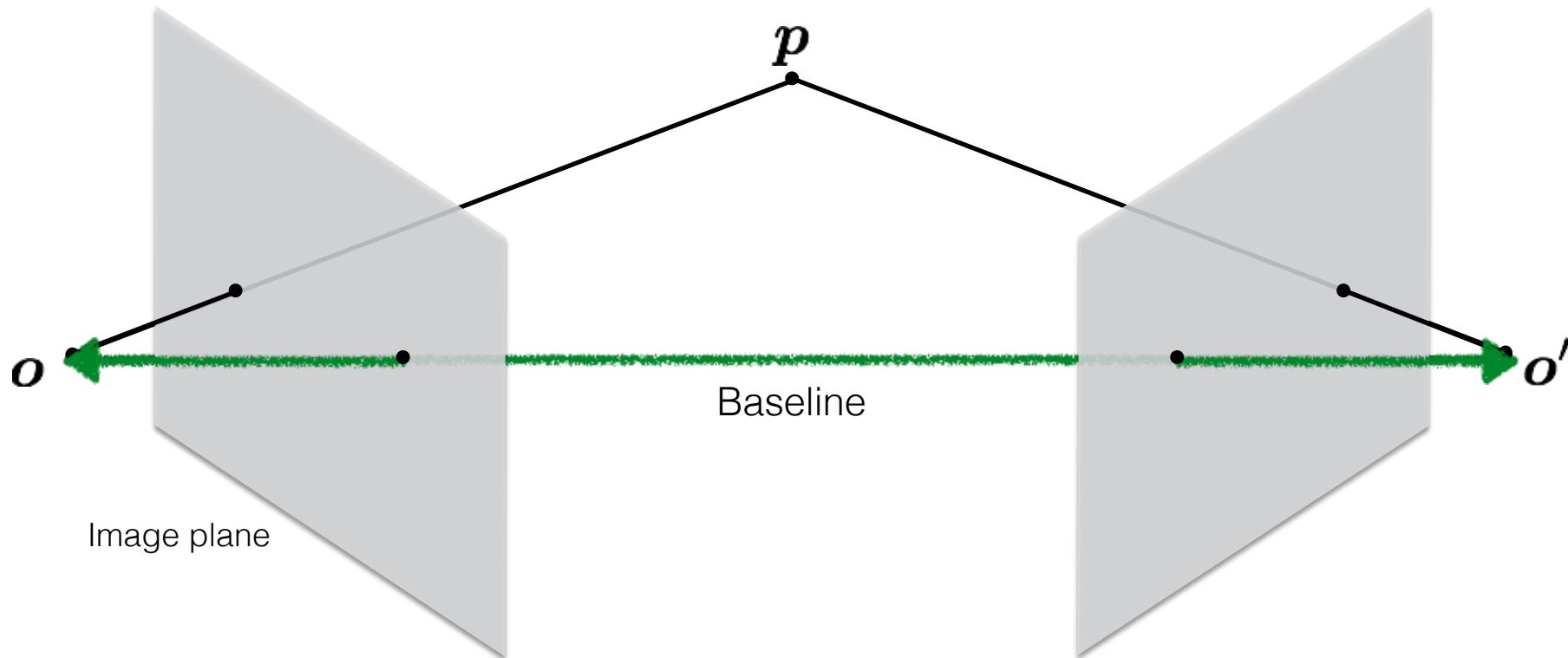
How do we solve homogeneous linear system?

S V D !

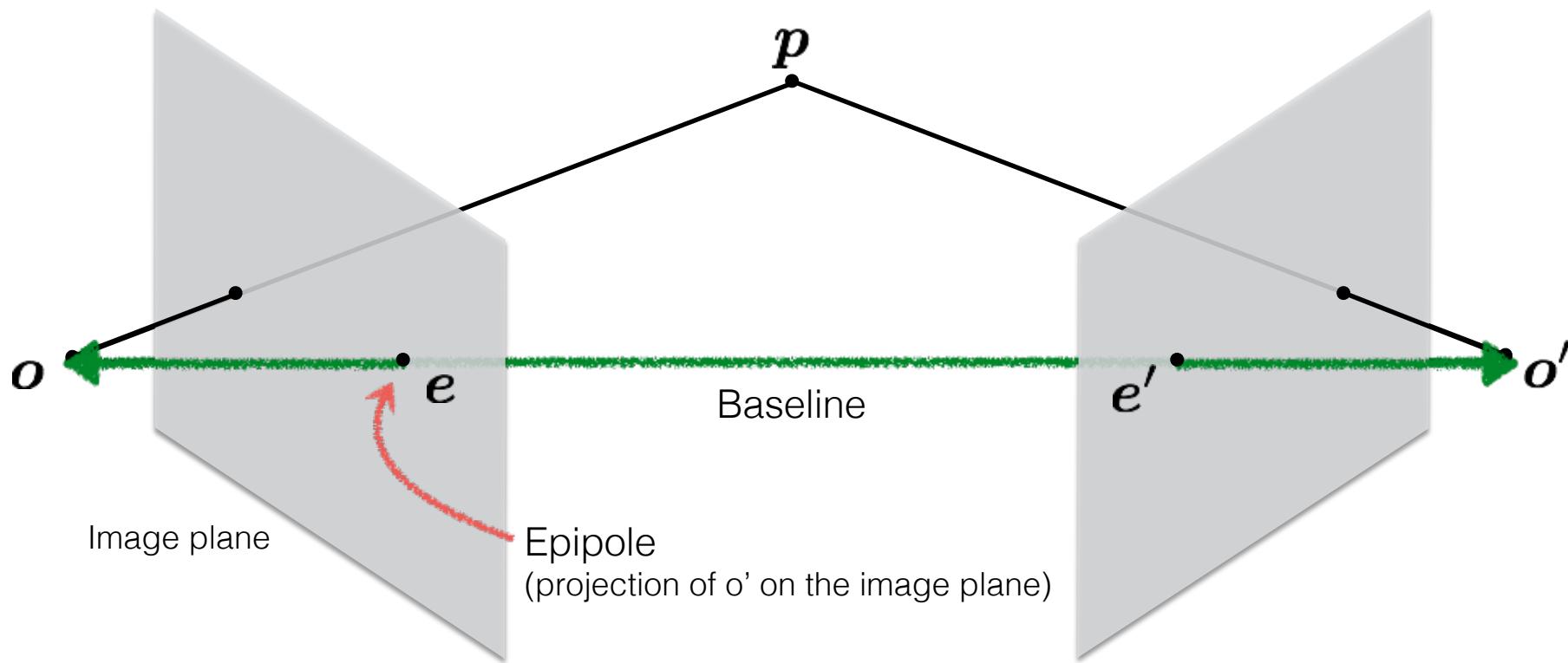
Epipolar geometry



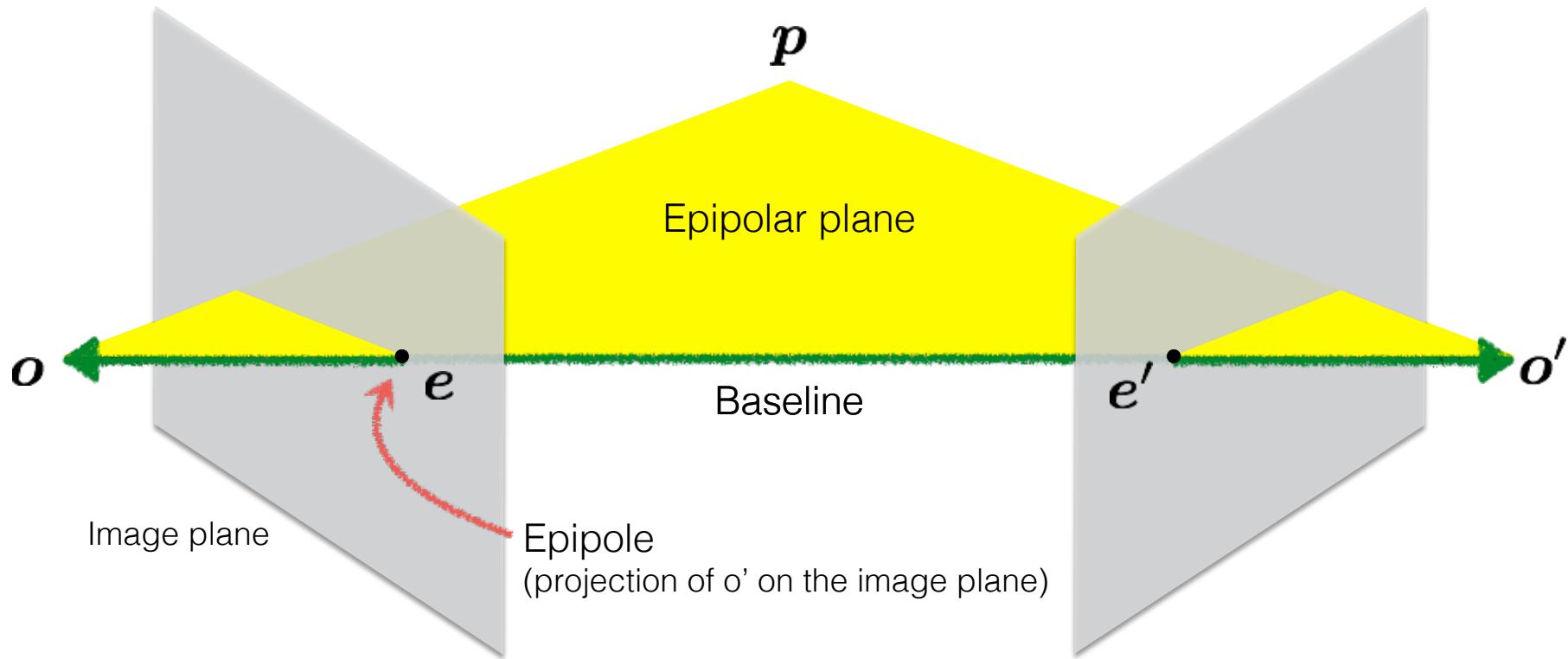
Epipolar geometry



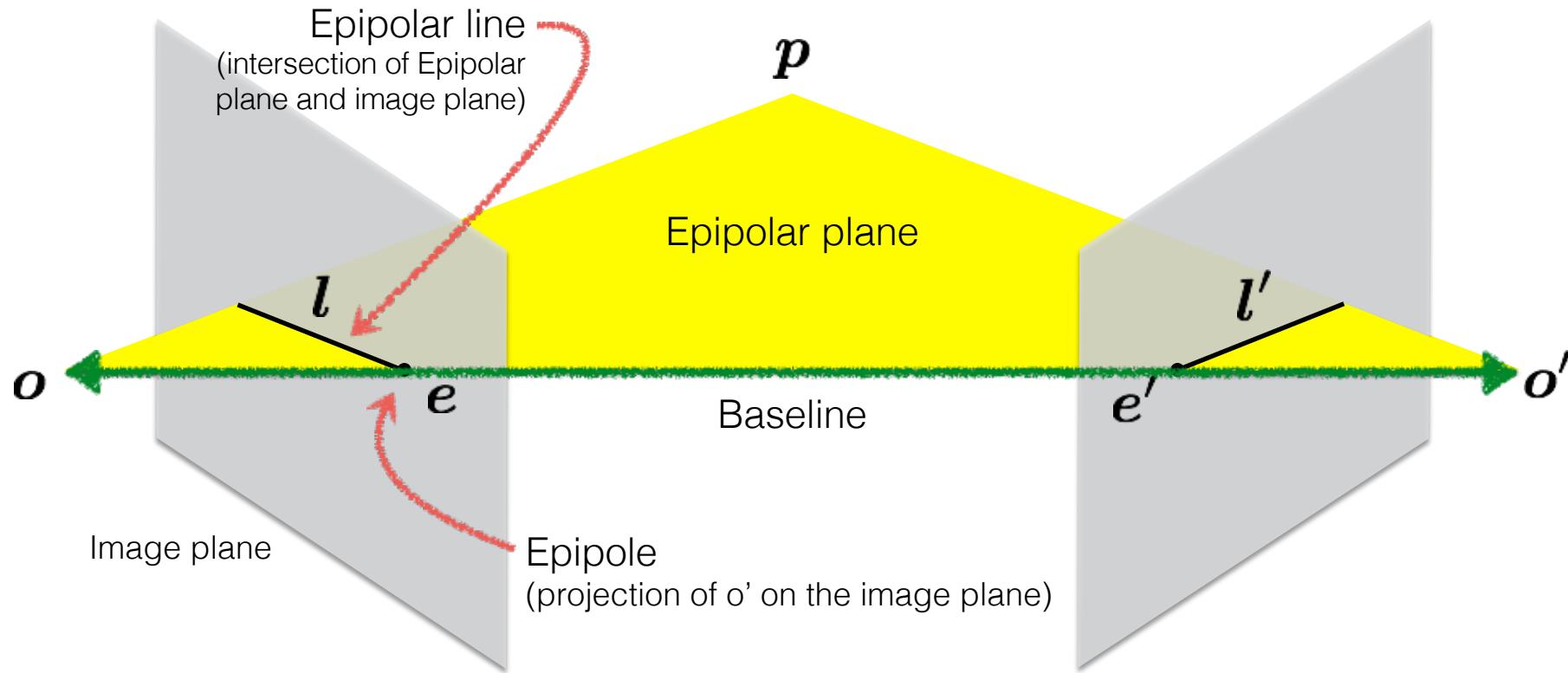
Epipolar geometry



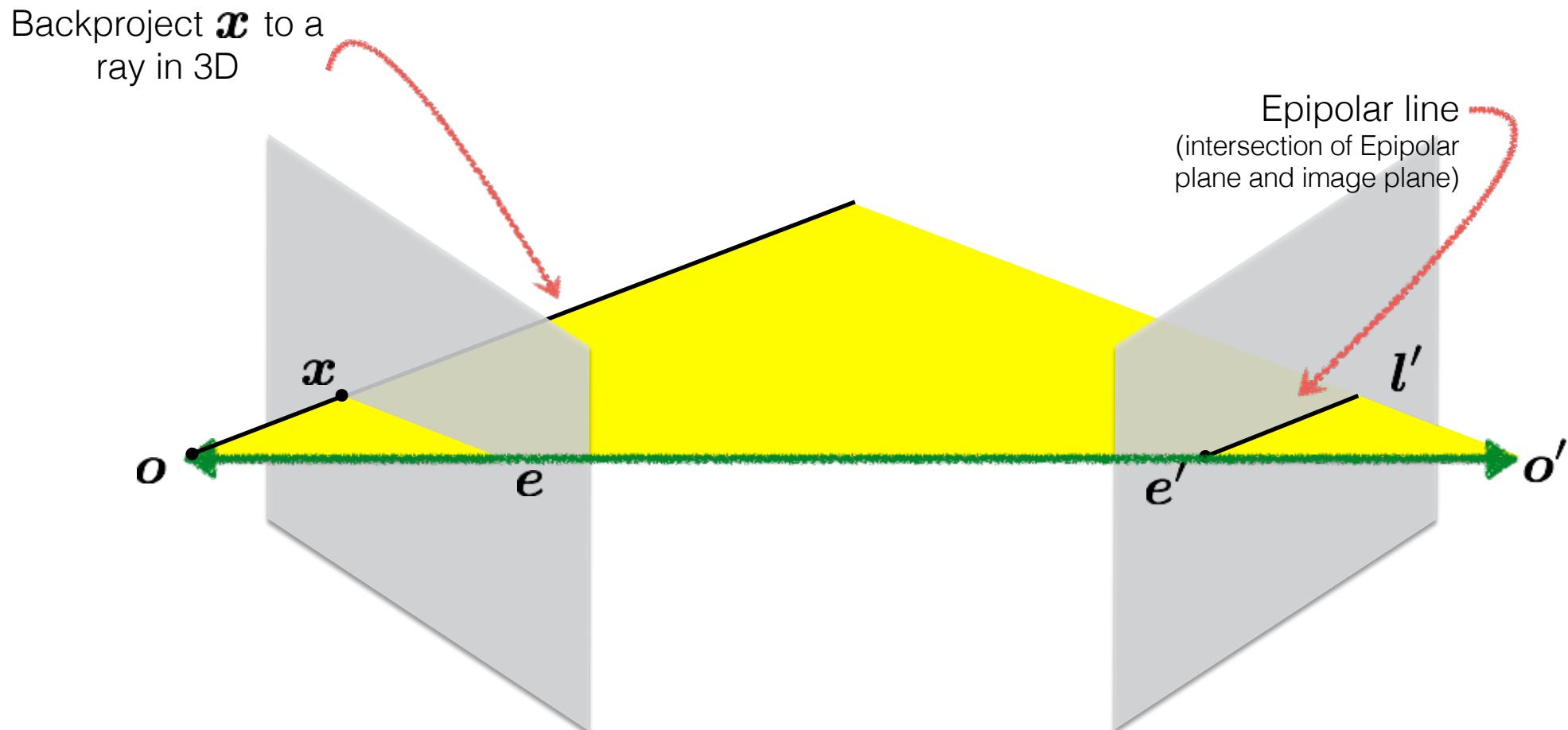
Epipolar geometry



Epipolar geometry

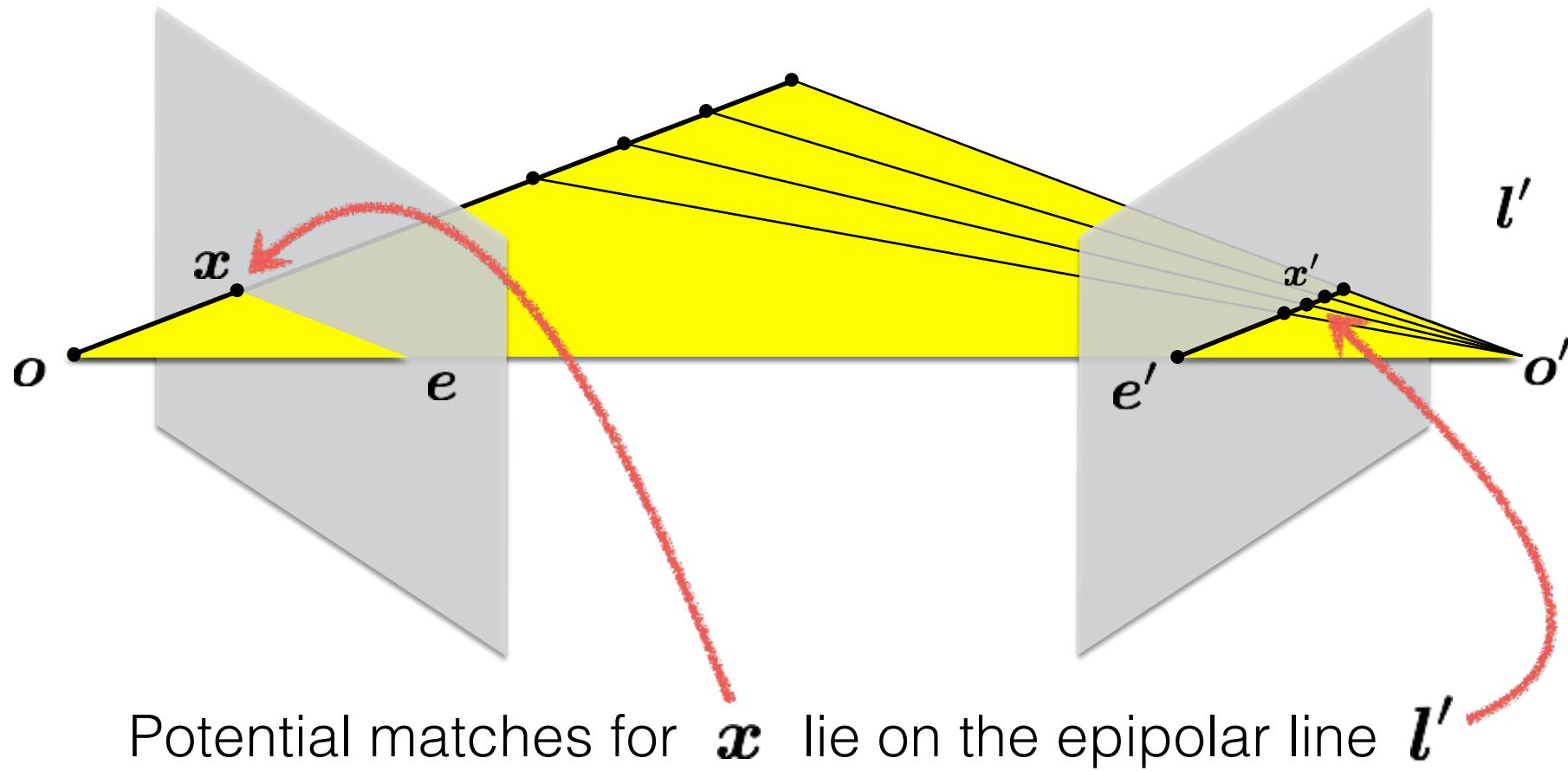


Epipolar constraint



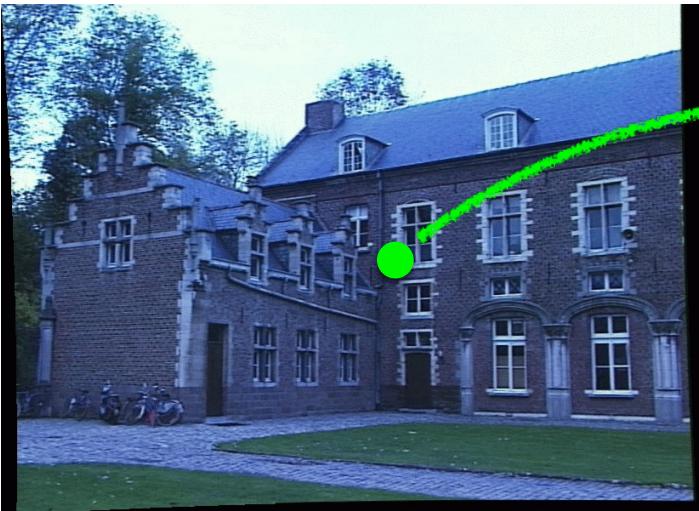
Another way to construct the epipolar plane, this time given \mathbf{x}

Epipolar constraint

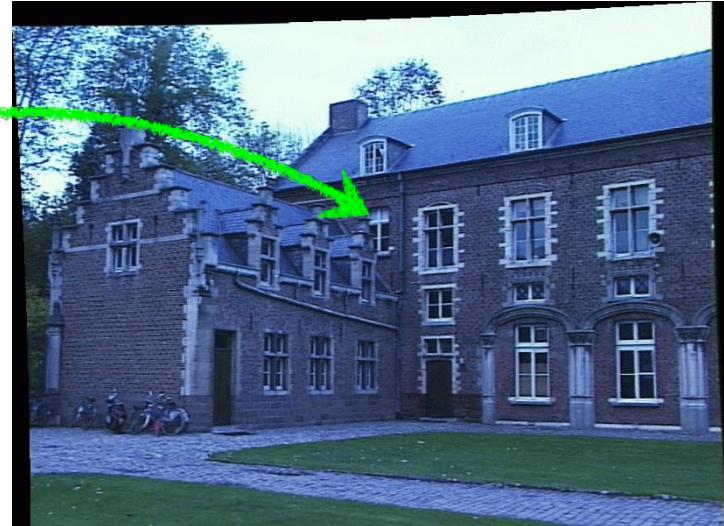


The epipolar constraint is an important concept for stereo vision

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



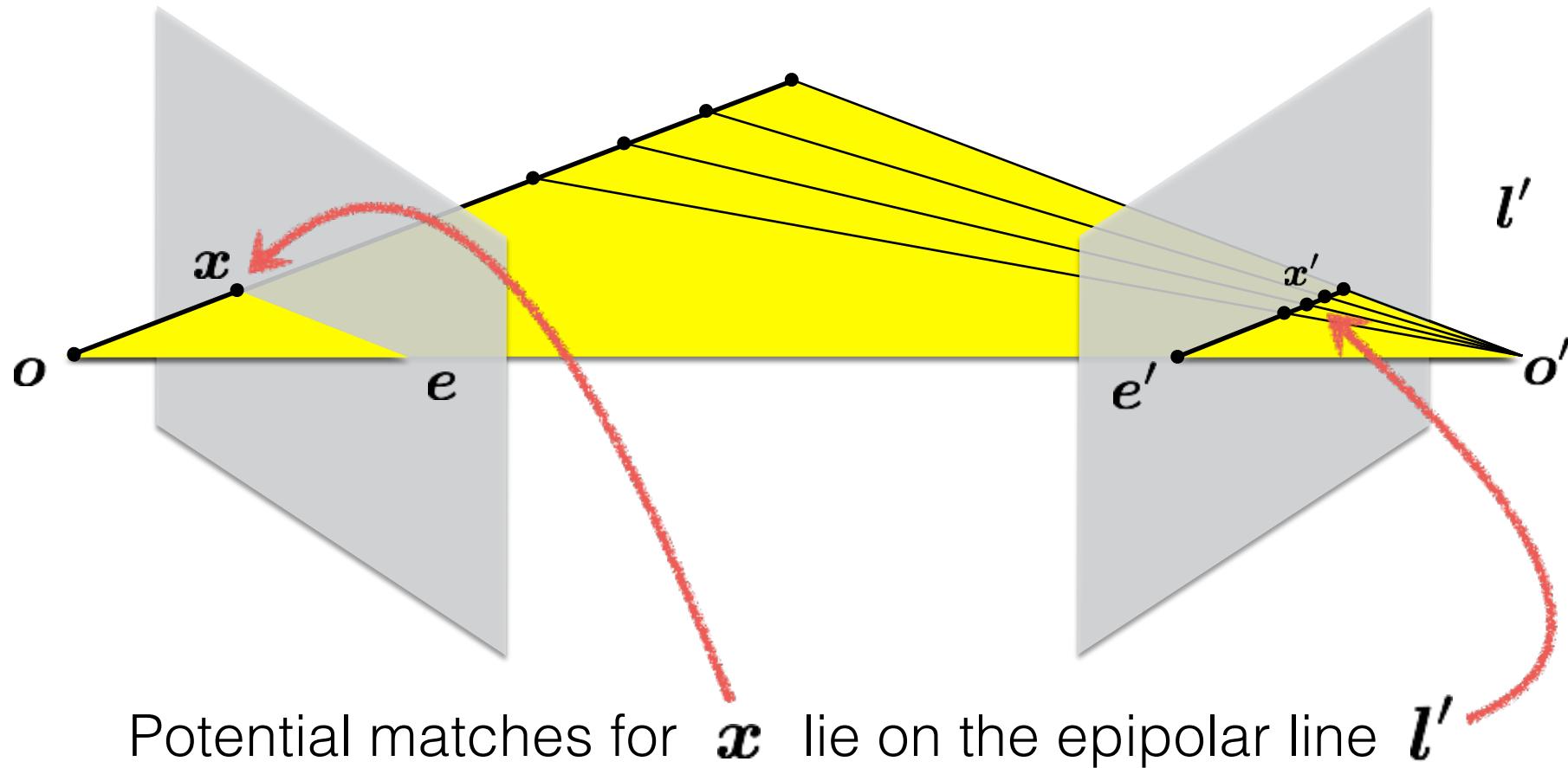
Left image



Right image

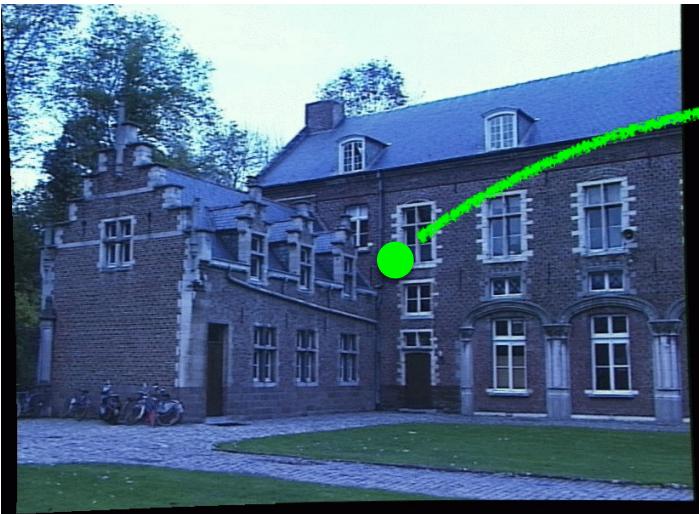
How would you do it?

Epipolar constraint



The epipolar constraint is an important concept for stereo vision

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



Left image



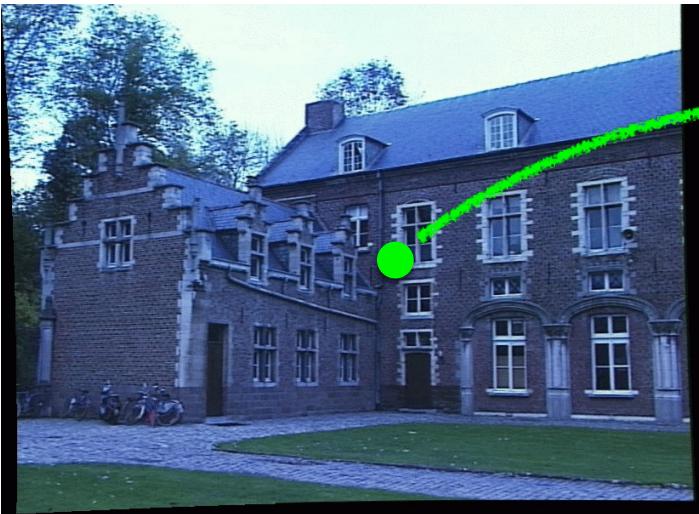
Right image

Want to avoid search over entire image

Epipolar constraint reduces search to a single line

The epipolar constraint is an important concept for stereo vision

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



Left image



Right image

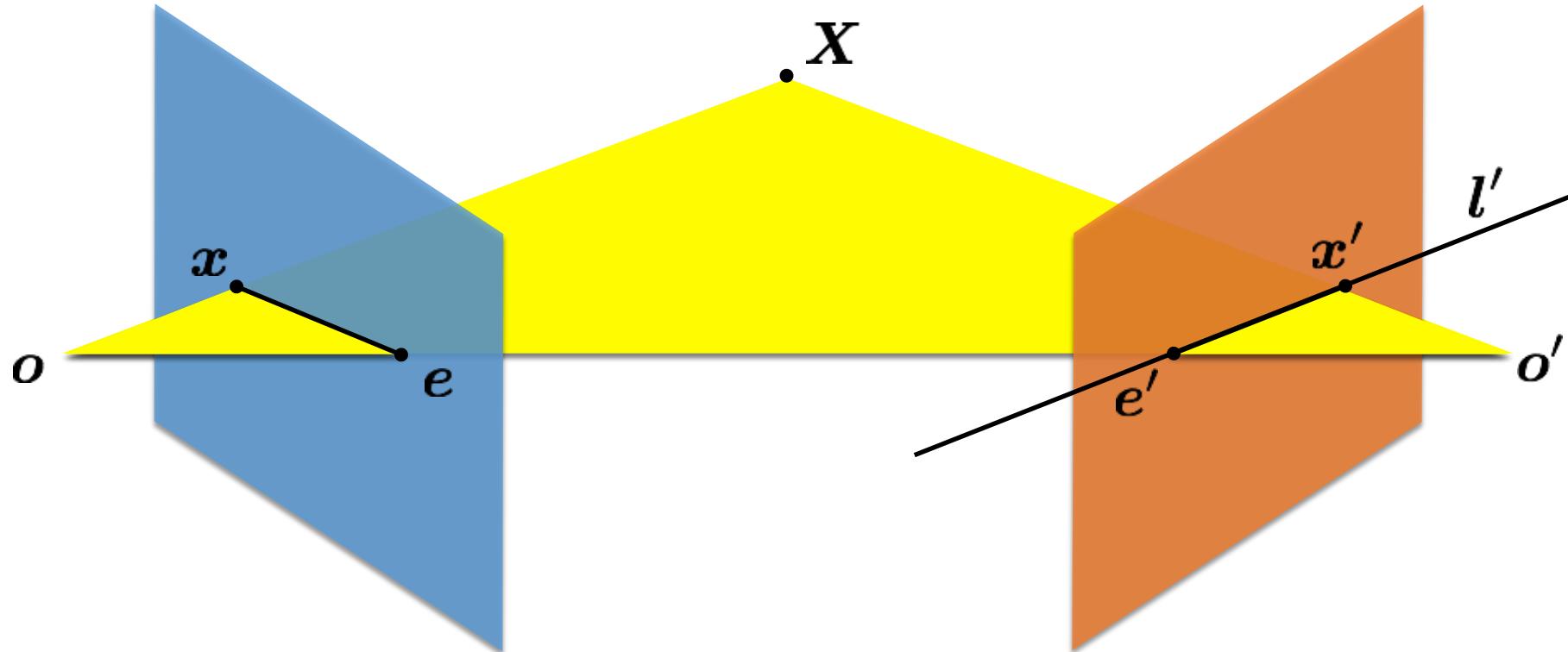
Want to avoid search over entire image

Epipolar constraint reduces search to a single line

How do you compute the epipolar line?

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

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



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

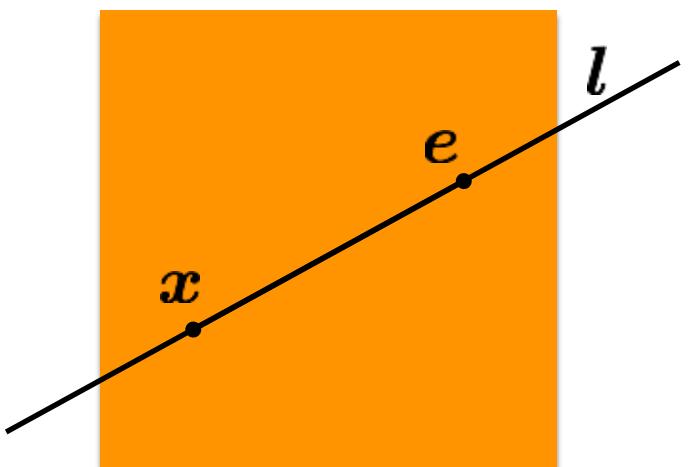
Representing the ...

Epipolar Line

$$ax + by + c = 0$$

in vector form

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

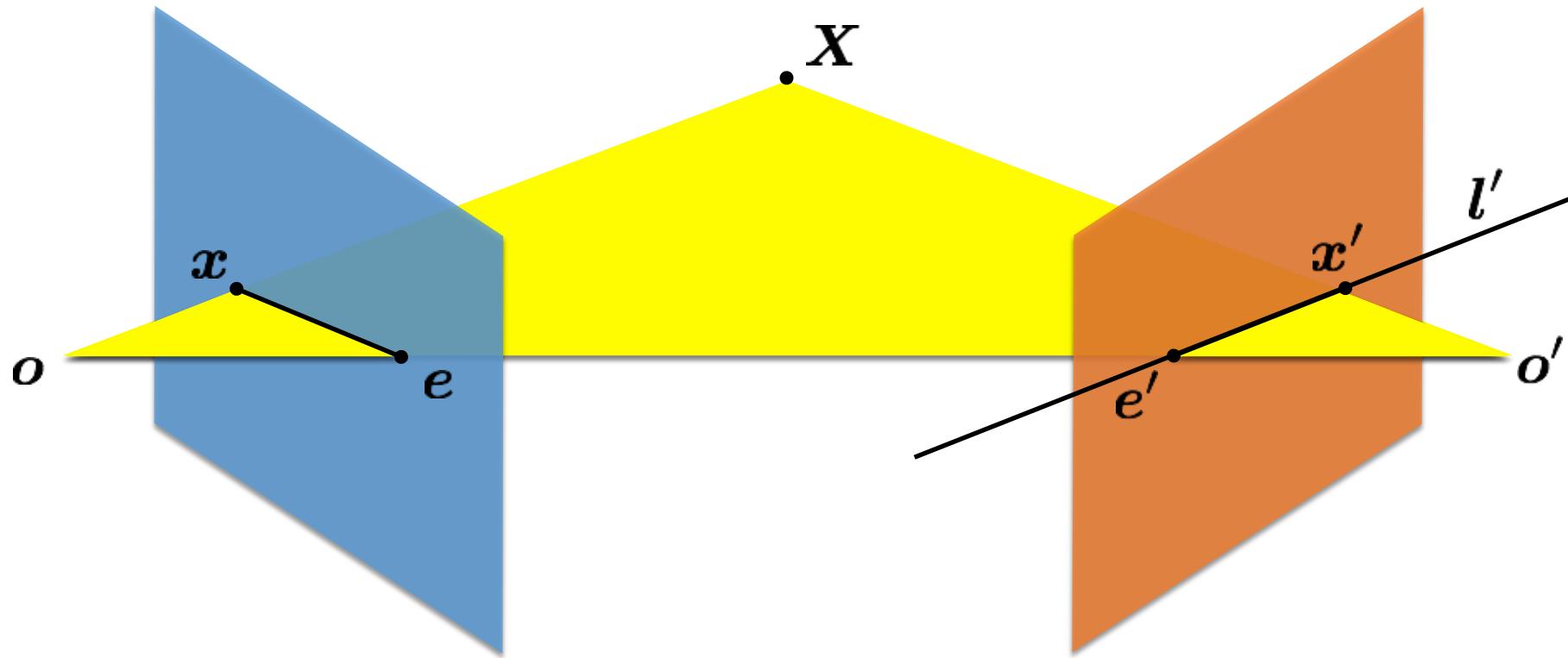


If the point \mathbf{x} is on the epipolar line \mathbf{l} then

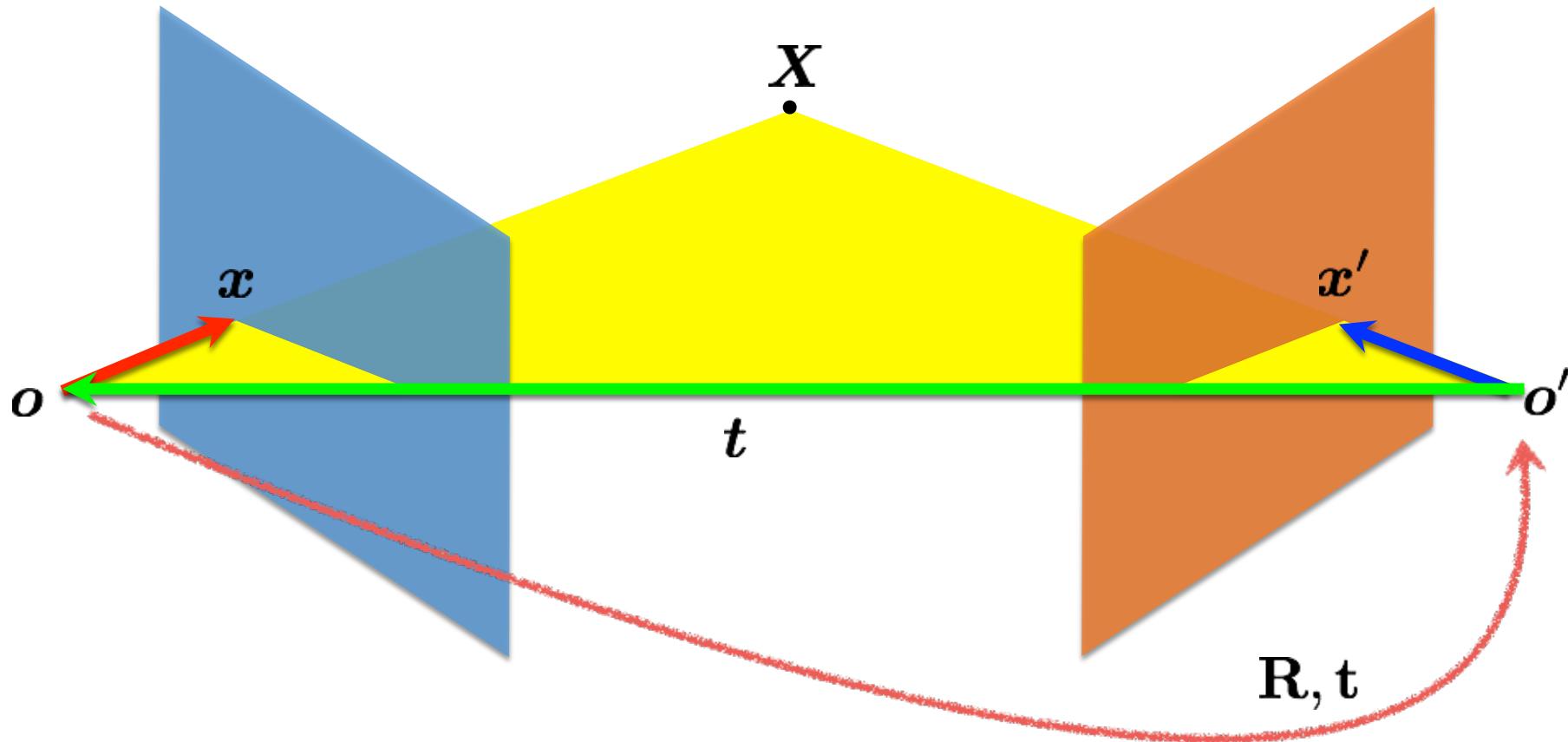
$$\mathbf{x}^\top \mathbf{l} = ?$$

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

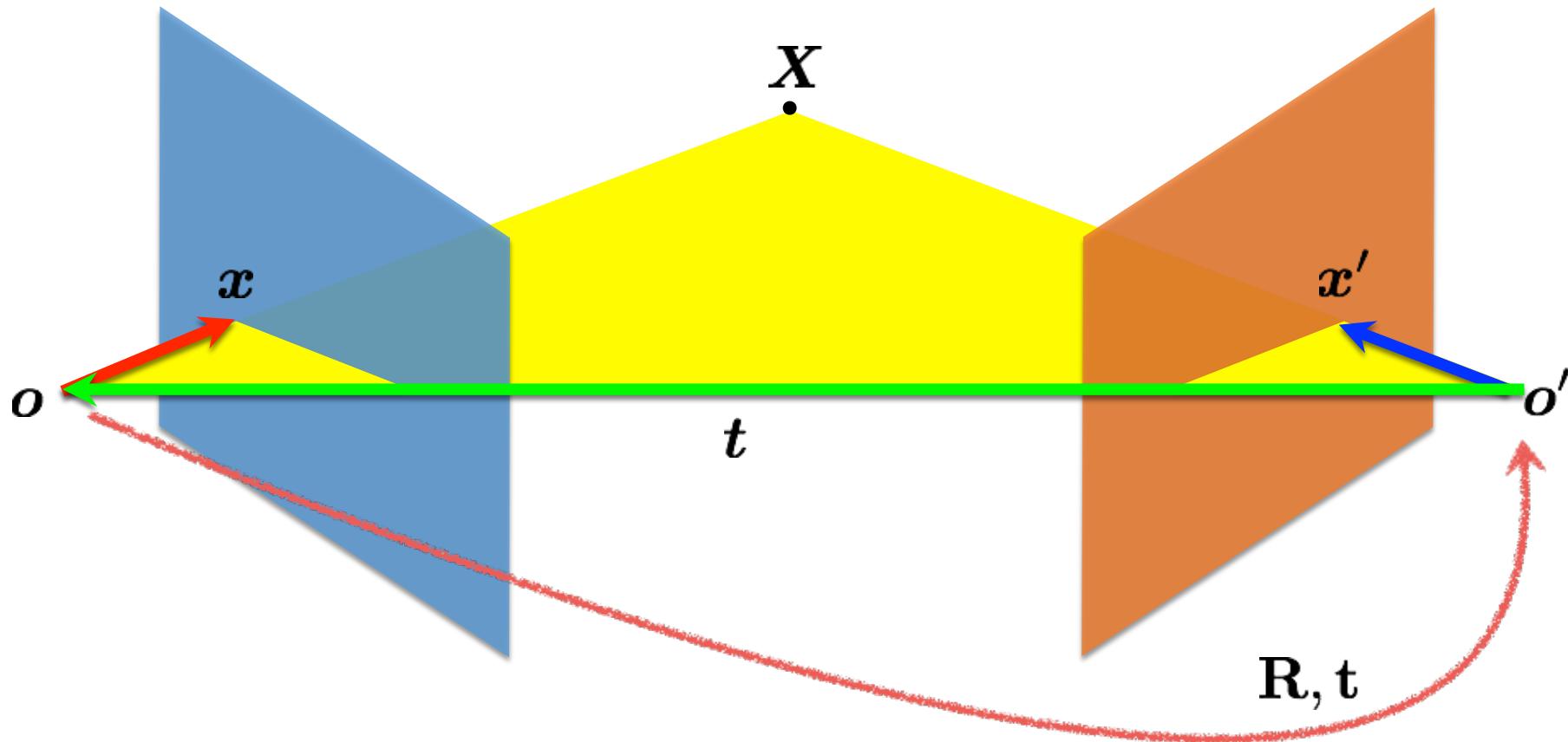
$$\mathbf{x}'^\top \mathbf{E}\mathbf{x} = ?$$



Where does the essential matrix come from?

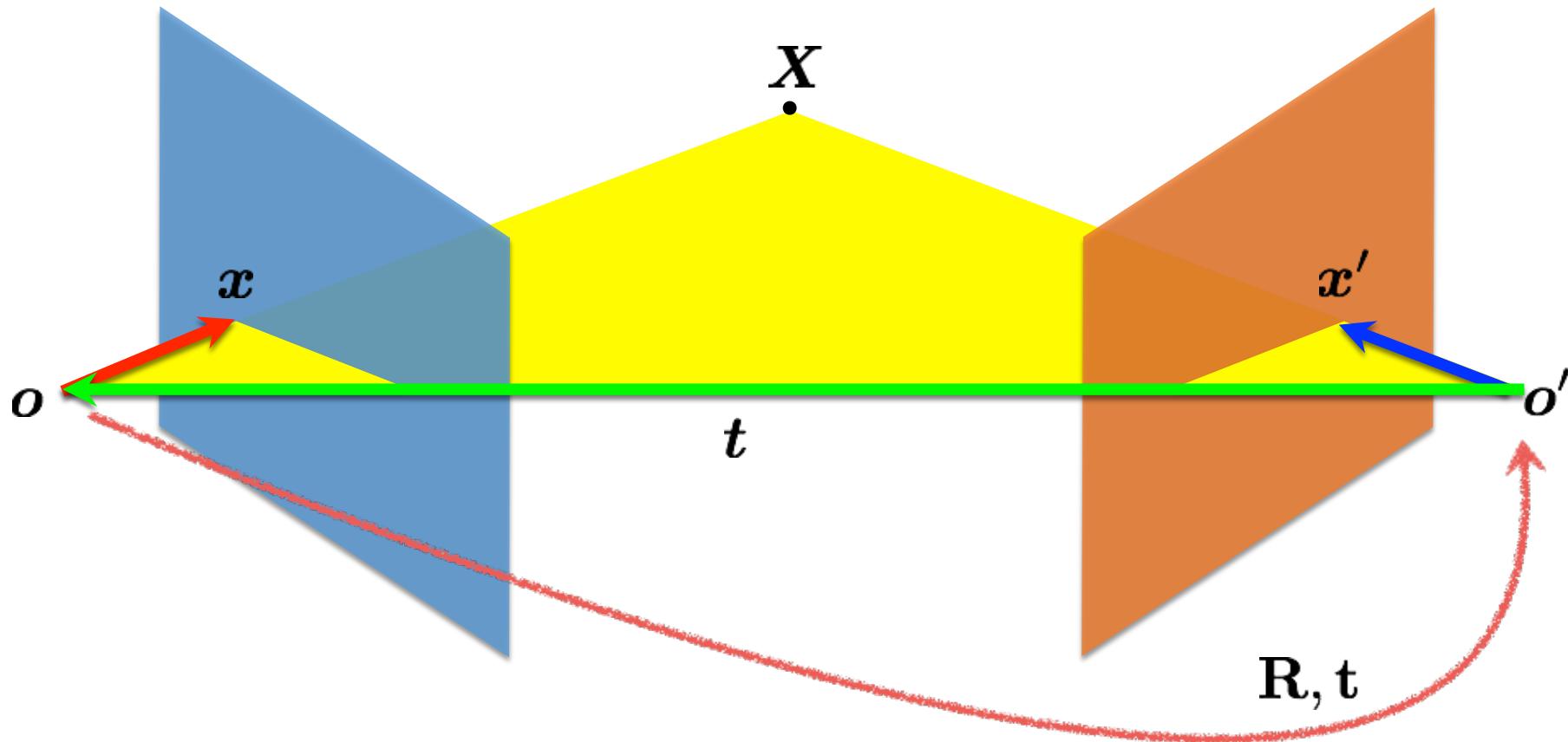


$$x' = R(x - t)$$



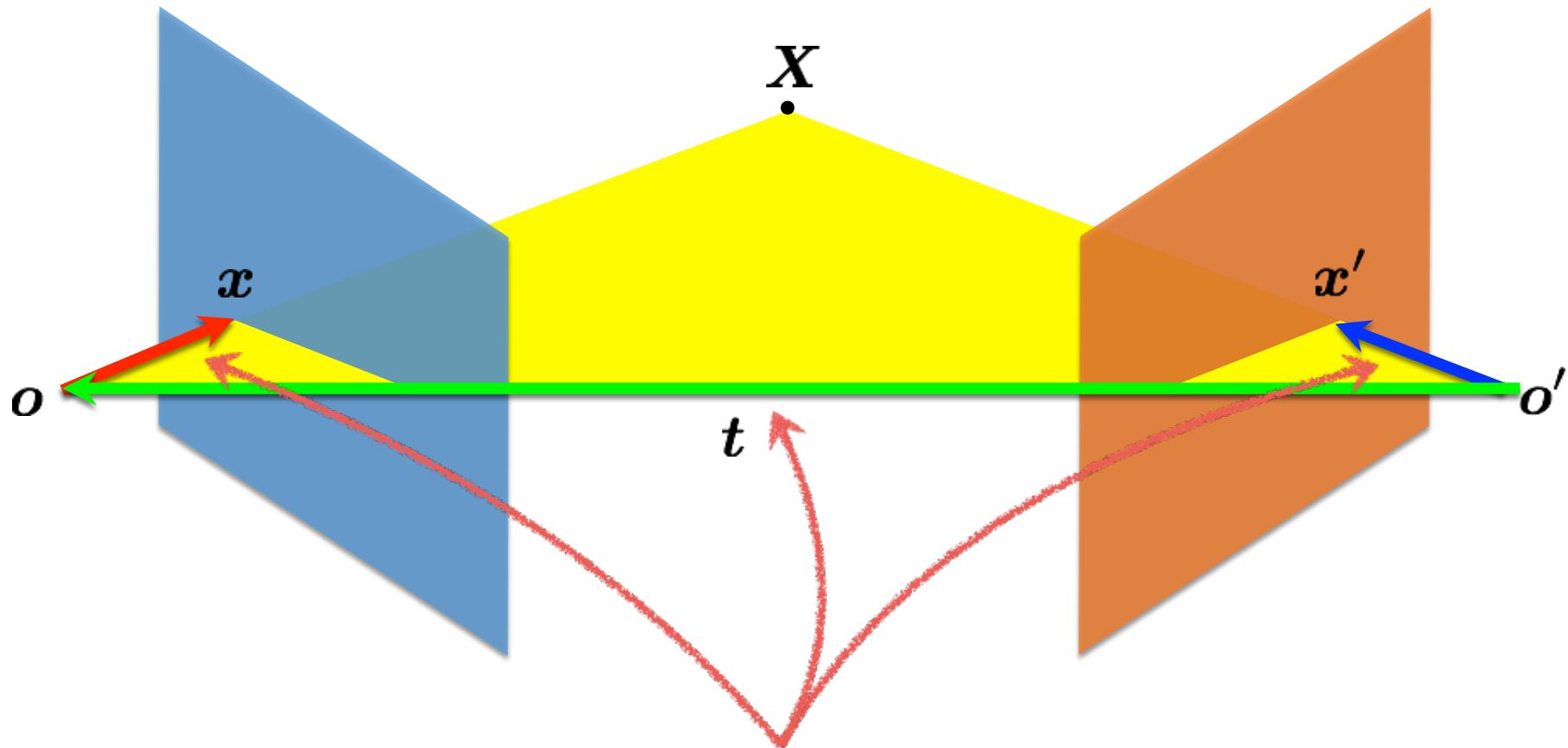
$$\mathbf{x}' = \mathbf{R}(\mathbf{x} - \mathbf{t})$$

Does this look familiar?



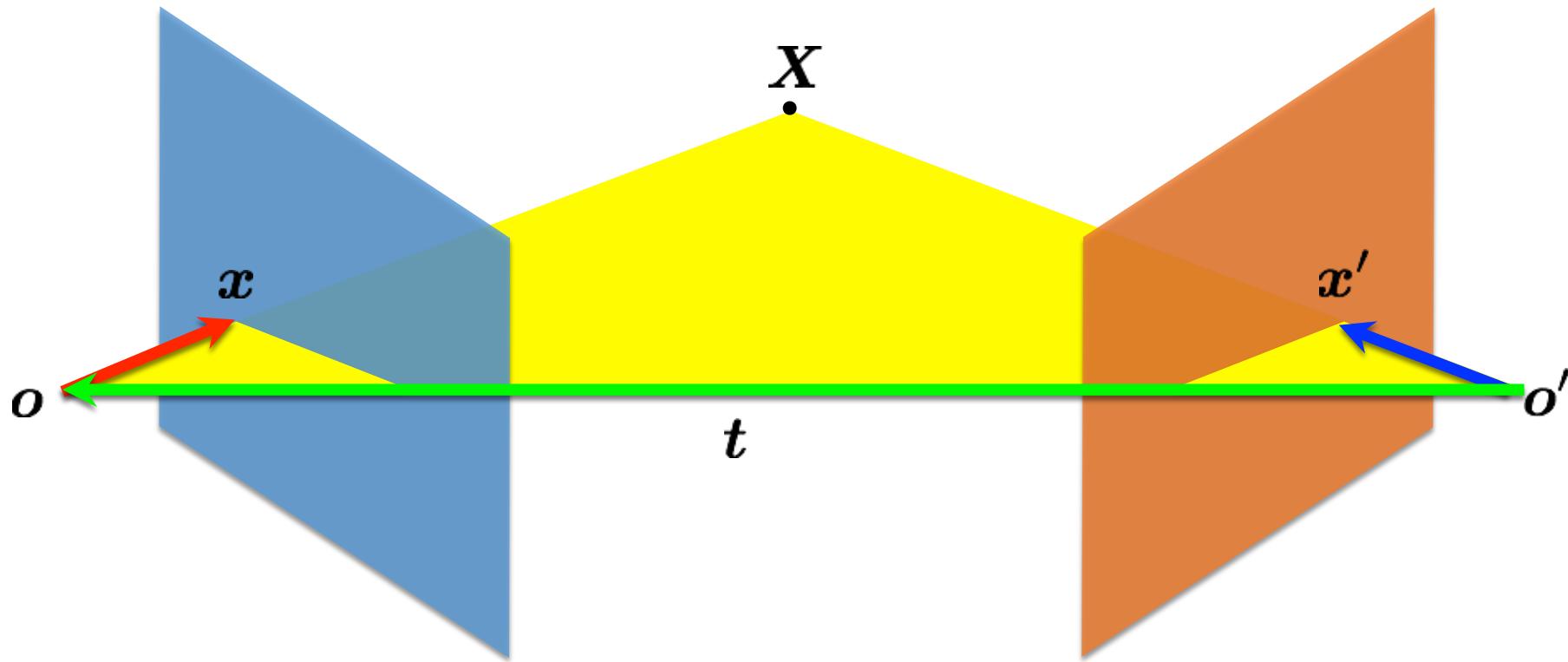
$$\mathbf{x}' = \mathbf{R}(\mathbf{x} - \mathbf{t})$$

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



These three vectors are coplanar

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

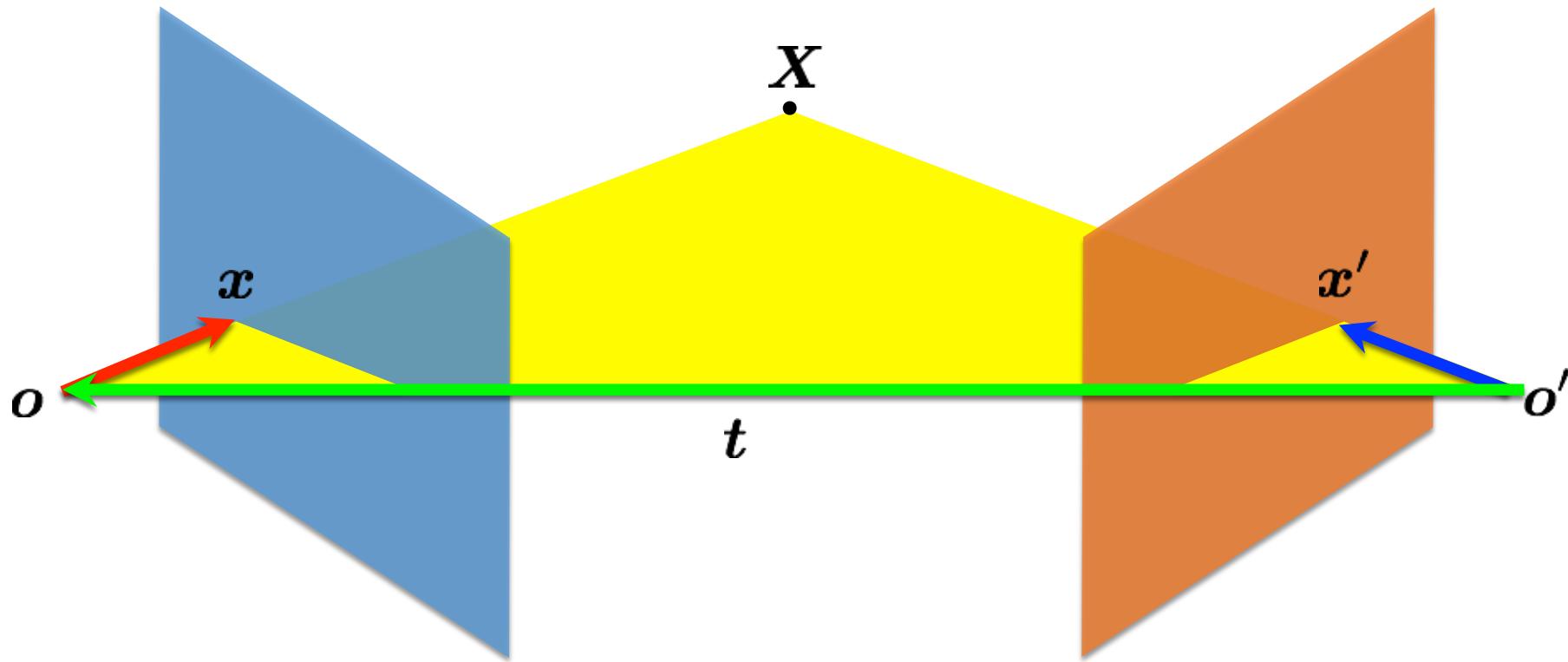


If these three vectors are coplanar $\mathbf{x}, \mathbf{t}, \mathbf{x}'$ then

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

dot product of orthogonal vectors

cross-product: vector orthogonal to plane



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

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

Putting it Together

rigid motion

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

The outer product
(w/ vector) could be
re-written as inner
product (w/ matrix)!

coplanarity

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

Rotation matrix is
orthonormal
(transpose = inverse!)

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

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

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

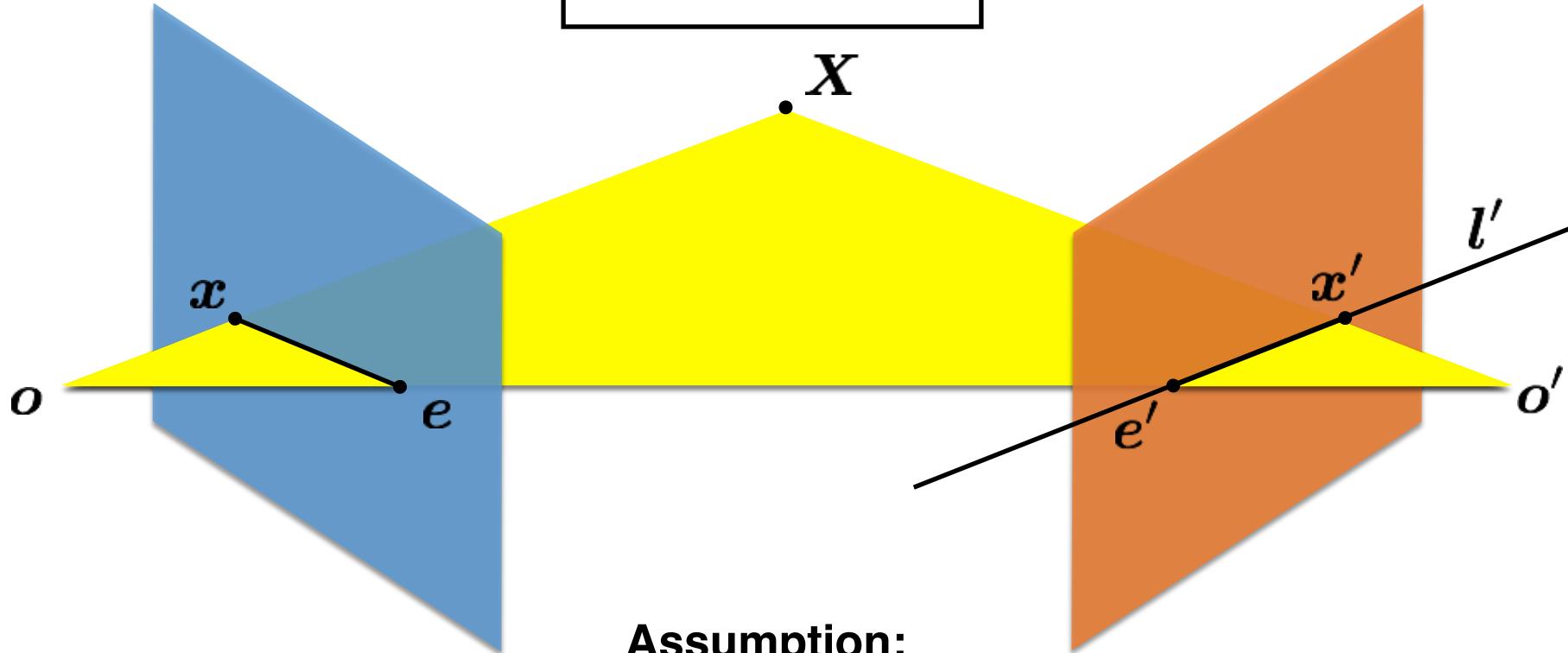
Sanity check:
dimension?

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

Essential Matrix
[Longuet-Higgins 1981]

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:

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

More Properties of E

Longuet-Higgins equation

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

Epipolar lines

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

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

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

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

Epipoles

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

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

(2D points expressed in camera coordinate system)

How do you generalize
to non-identity intrinsic
matrices?

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

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

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

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

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

camera
point

image
point

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

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

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

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

Writing out the epipolar constraint in terms of image coordinates

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

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

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

Same equation works in image coordinates!

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

it maps pixels to epipolar lines

More Properties of \mathbf{E}

Longuet-Higgins equation

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

Epipolar lines

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

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

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

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

Epipoles

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

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

(points in **image** coordinates)

Breaking down the fundamental matrix

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

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

Depends on both intrinsic and extrinsic parameters

Breaking down the fundamental matrix

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

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

Depends on both intrinsic and extrinsic parameters

How would you solve for F?

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

Assume you have M matched *image* points

$$\{\mathbf{x}_m, \mathbf{x}'_m\} \quad m = 1, \dots, M$$

Each correspondence should satisfy

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

How would you solve for the $3 \times 3 \mathbf{F}$ matrix?

Assume you have M matched *image* points (via Harris, SIFT...)

$$\{\mathbf{x}_m, \mathbf{x}'_m\} \quad m = 1, \dots, M$$

Each correspondence should satisfy

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

How would you solve for the $3 \times 3 \mathbf{F}$ matrix?

S V D

Assume you have M matched *image* points

$$\{\mathbf{x}_m, \mathbf{x}'_m\} \quad m = 1, \dots, M$$

Each correspondence should satisfy

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

How would you solve for the $3 \times 3 \mathbf{F}$ matrix?

Set up a homogeneous linear system with 9 unknowns

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

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

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

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

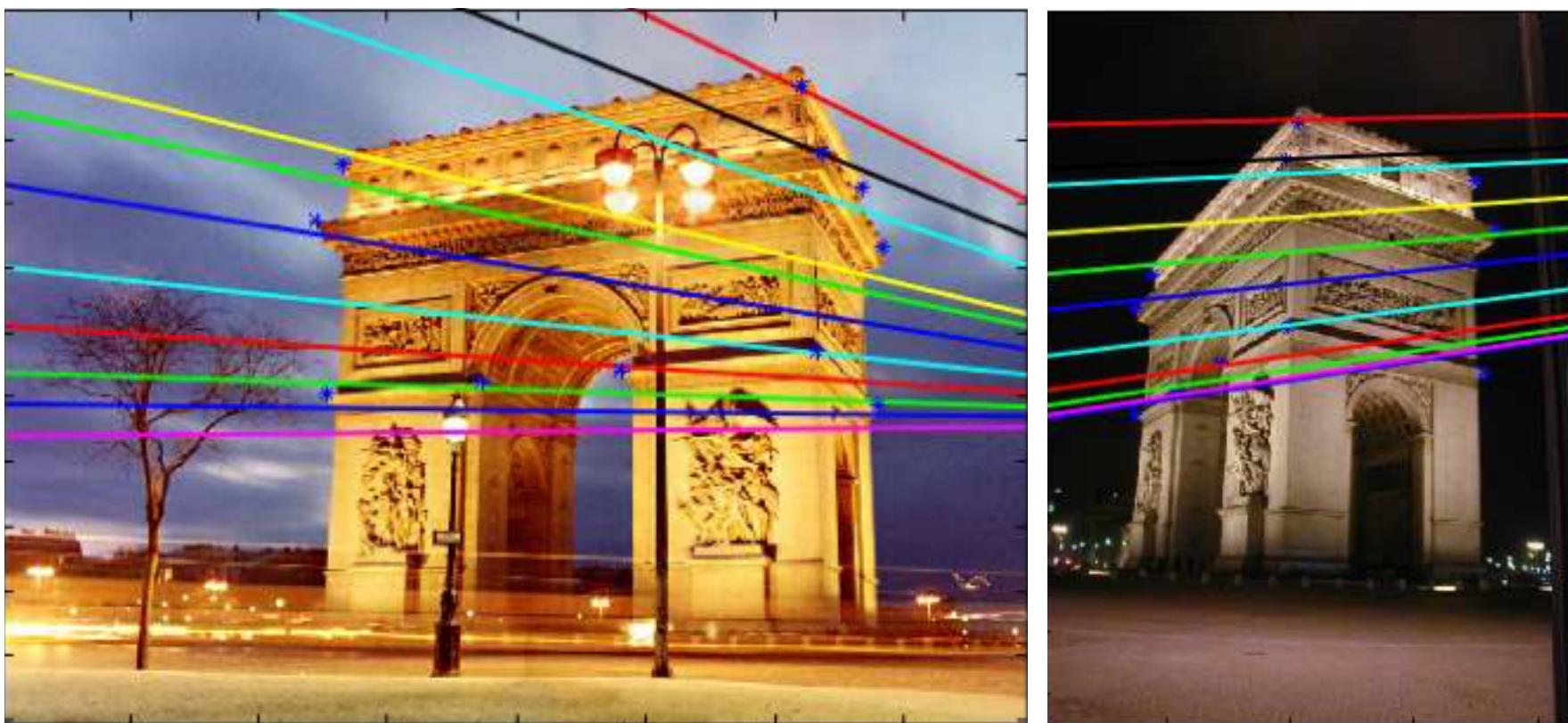
$$\begin{bmatrix} x'_m & y'_m & 1 \end{bmatrix} \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \begin{bmatrix} x_m \\ y_m \\ 1 \end{bmatrix} = 0$$

Set up a homogeneous linear system with 9 unknowns

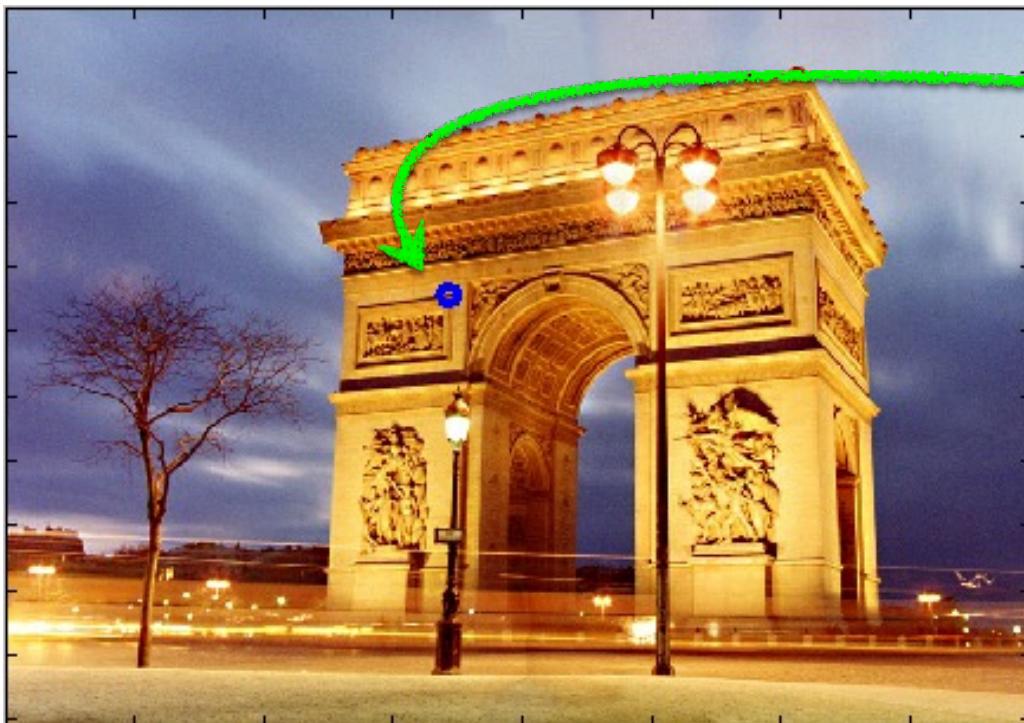
$$\begin{bmatrix} x_1x'_1 & x_1y'_1 & x_1 & y_1x'_1 & y_1y'_1 & y_1 & x'_1 & y'_1 & 1 \\ \vdots & \vdots \\ x_Mx'_M & x_My'_M & x_M & y_Mx'_M & y_My'_M & y_M & x'_M & y'_M & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix} = \mathbf{0}$$

SVD !

Example: epipolar lines



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



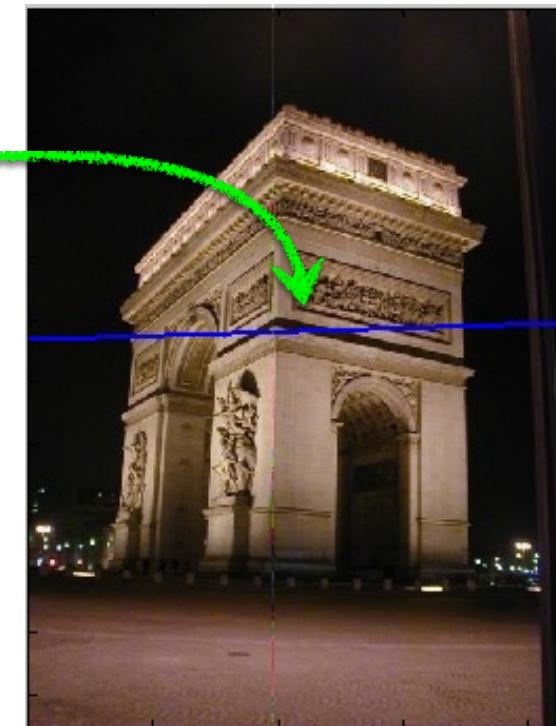
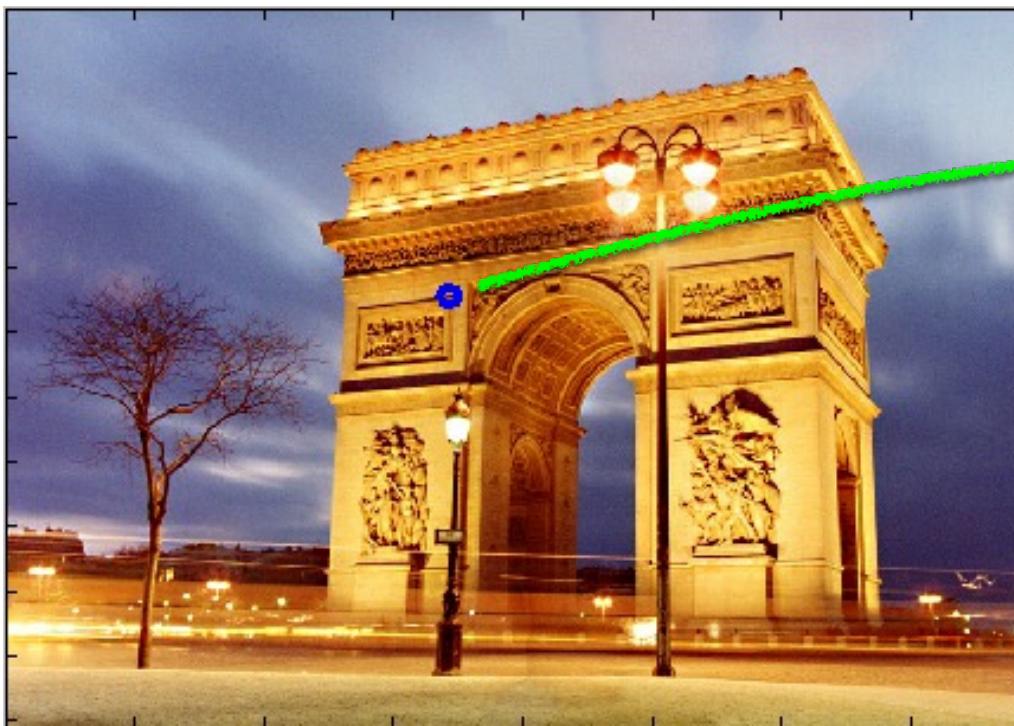
$$\mathbf{x} = \begin{bmatrix} 343.53 \\ 221.70 \\ 1.0 \end{bmatrix}$$

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

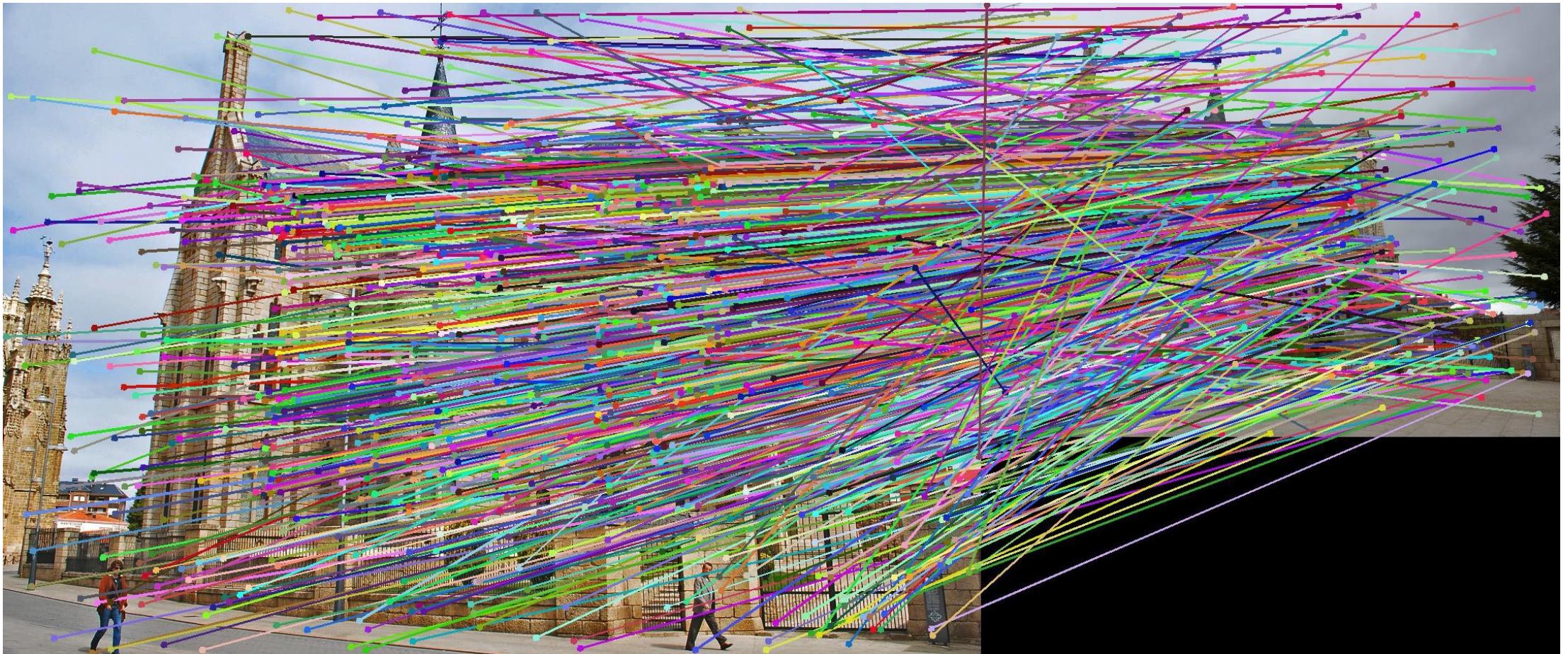
$$= \begin{bmatrix} 0.0295 \\ 0.9996 \\ -265.1531 \end{bmatrix}$$

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

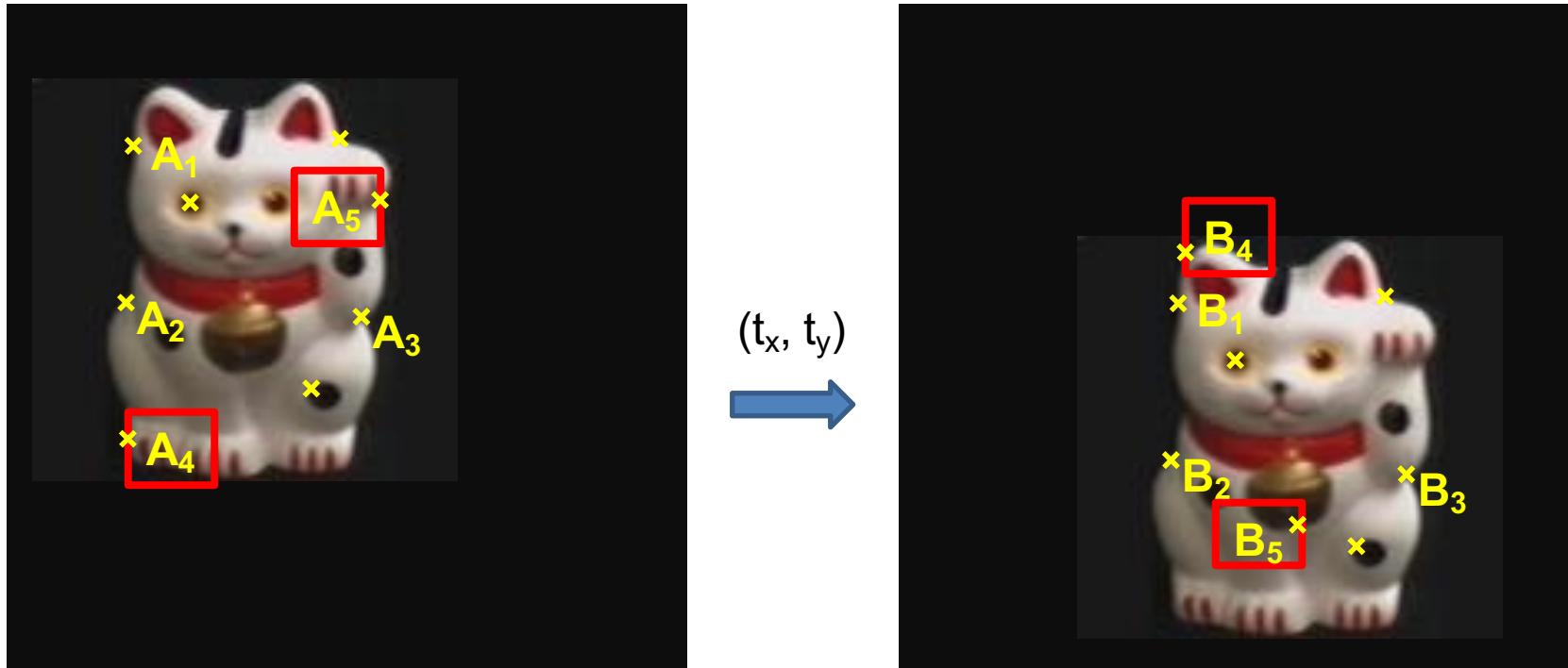
$$= \begin{bmatrix} 0.0295 \\ 0.9996 \\ -265.1531 \end{bmatrix}$$



8-point is sufficient in theory...
but least square often not robust enough



Example: solving for translation?



Problem: outliers A₄-B₄ and A₅-B₅ which *incorrectly* correspond

RANSAC solution (RANdom SAmples Consensus) :
Fischler & Bolles in '81.

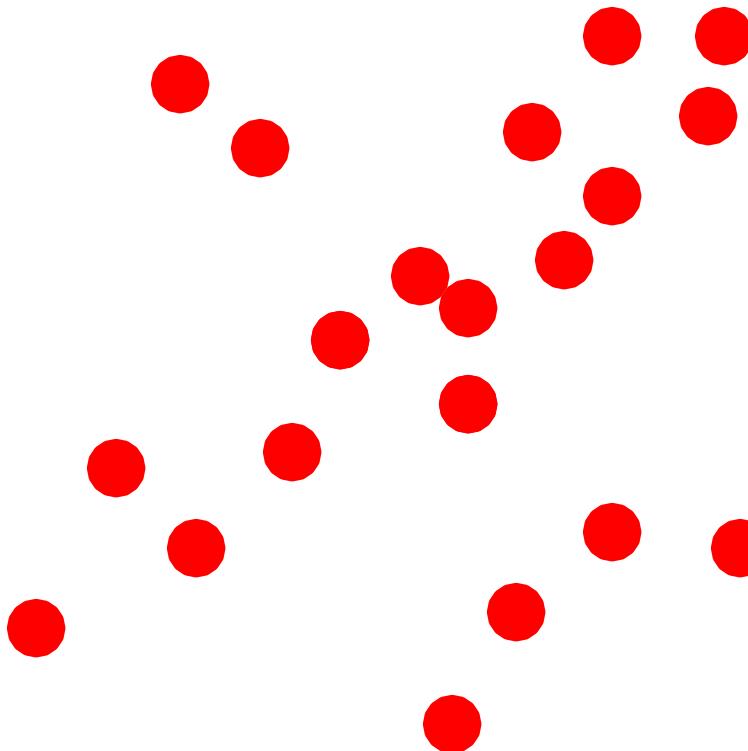
1. Sample a set of matching points (1 pair)
2. Solve for transformation parameters
3. Score parameters with number of inliers
4. Repeat steps 1-3 N times

$$\begin{bmatrix} x_i^B \\ y_i^B \end{bmatrix} = \begin{bmatrix} x_i^A \\ y_i^A \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

RANSAC

(RANdom SAmple Consensus) :

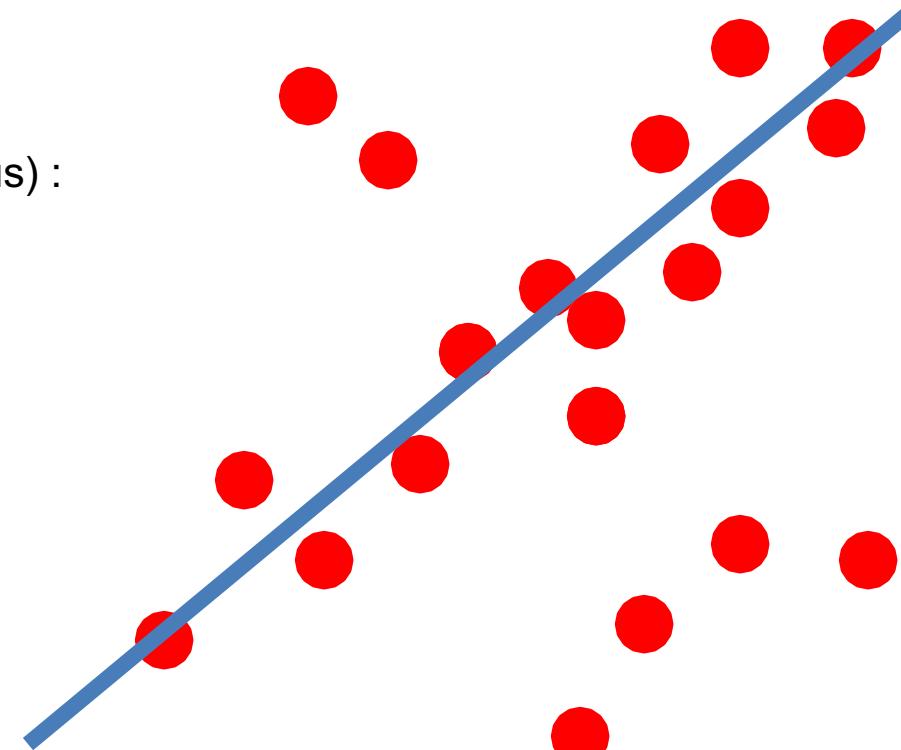
Fischler & Bolles in '81.



RANSAC

(RANdom SAmple Consensus) :

Fischler & Bolles in '81.

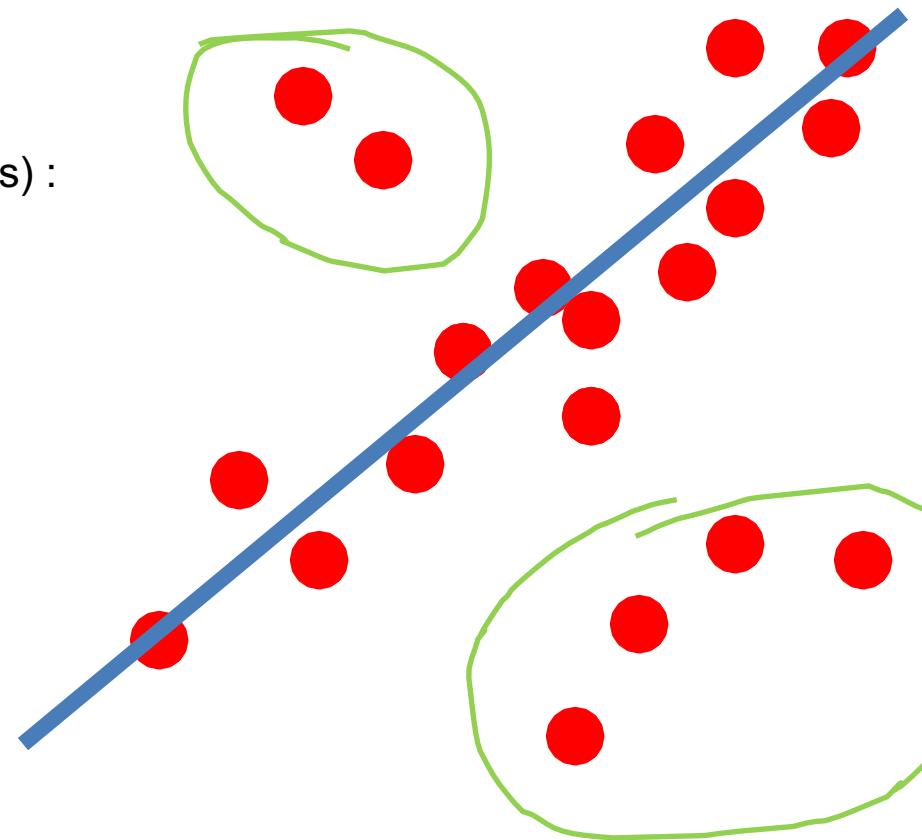


This data is noisy, but we expect a good fit
to a known model.

RANSAC

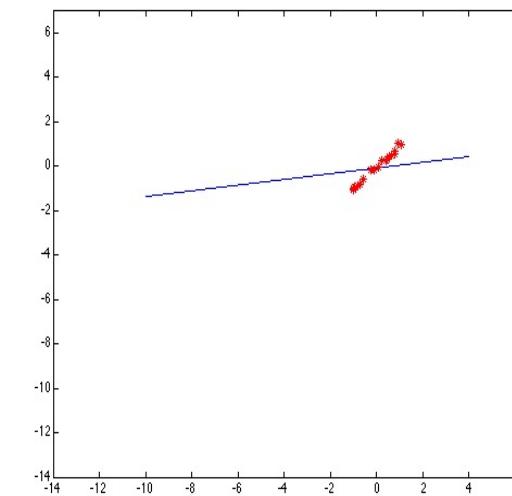
(RANdom SAmple Consensus) :

Fischler & Bolles in '81.



This data is noisy, but we expect a good fit to a known model.

Here, we expect to see a line, but least-squares fitting will produce the wrong result due to strong outlier presence.

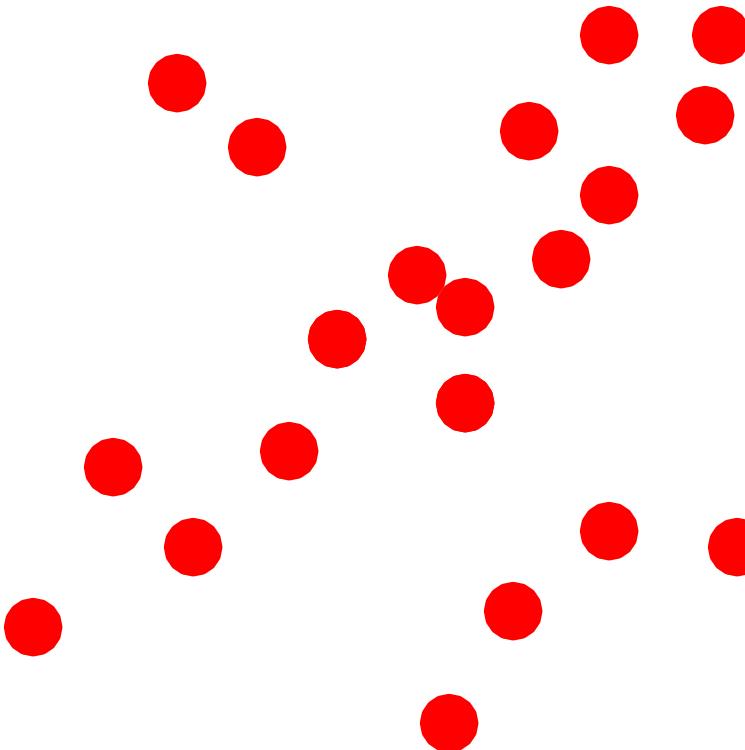


Problem: squared error heavily penalizes outliers

RANSAC

(RANdom SAmple Consensus) :

Fischler & Bolles in '81.



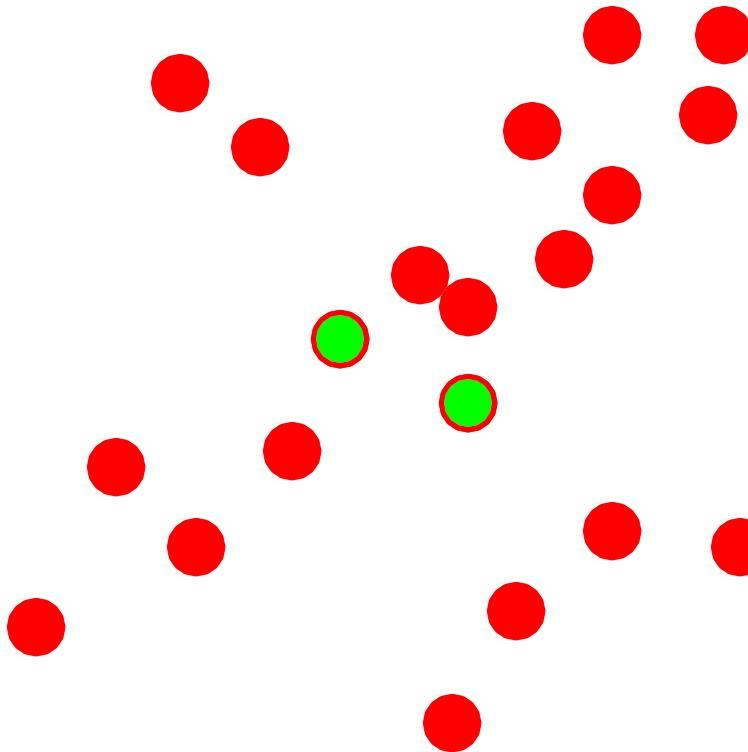
Algorithm:

1. **Sample** (randomly) the number of points s required to fit the model
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

RANSAC

Line fitting example



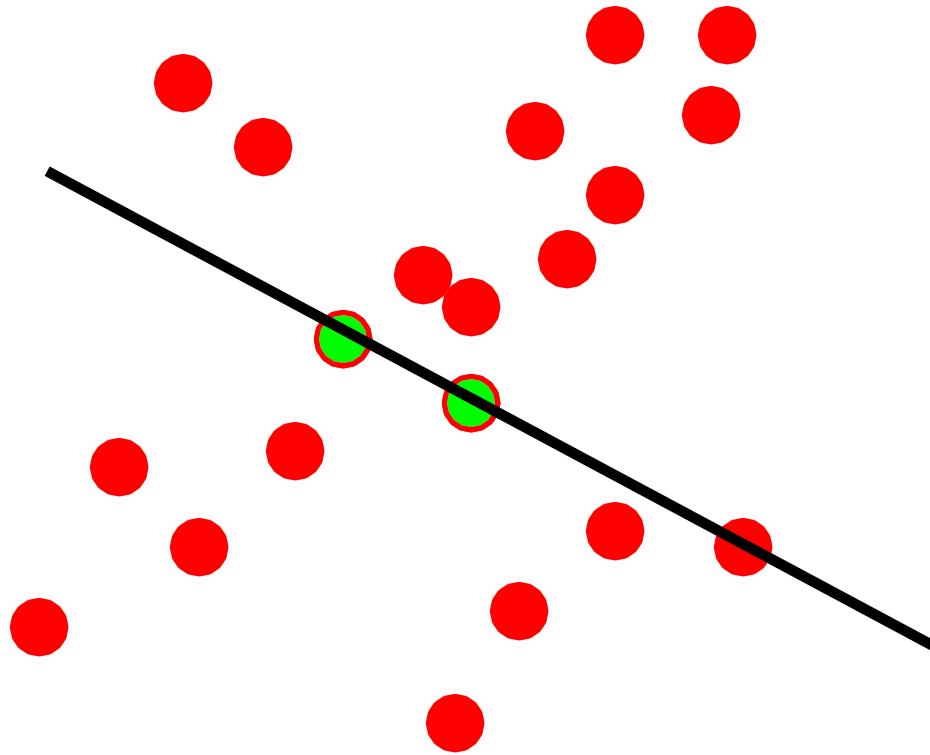
Algorithm:

1. **Sample** (randomly) the number of points required to fit the model ($s=2$)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

RANSAC

Line fitting example



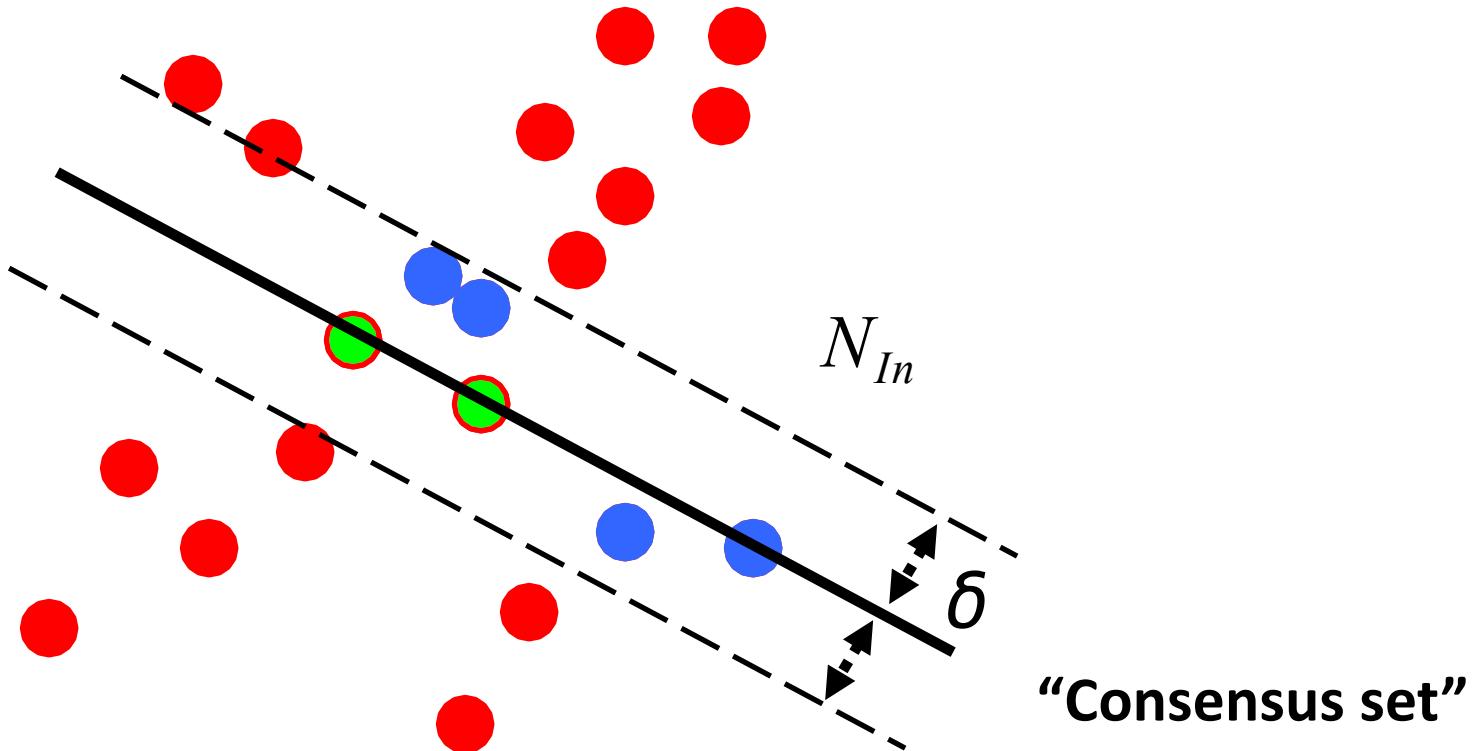
Algorithm:

1. **Sample** (randomly) the number of points required to fit the model ($s=2$)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

RANSAC

Line fitting example

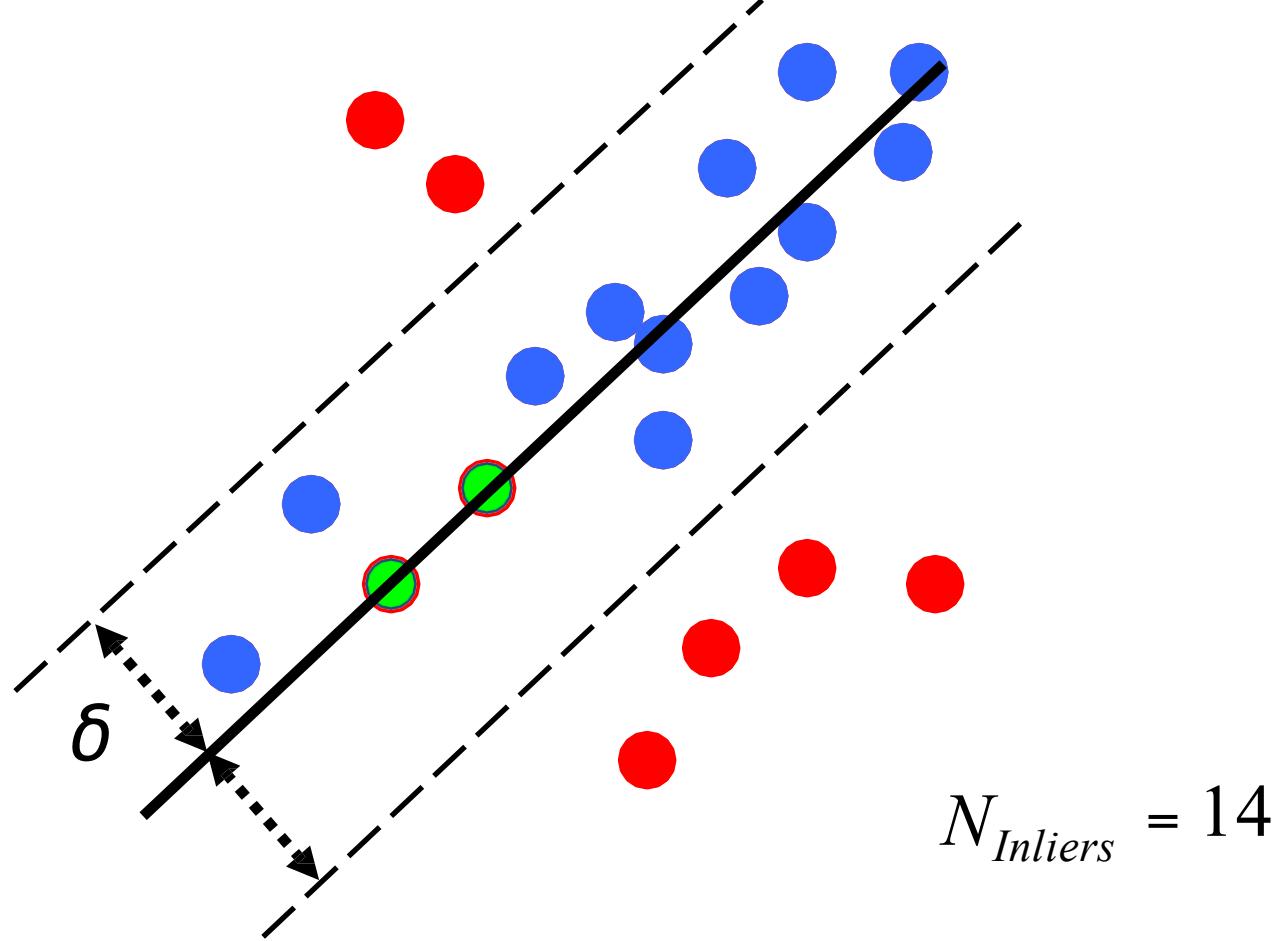


Algorithm:

1. **Sample** (randomly) the number of points required to fit the model ($s=2$)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

RANSAC

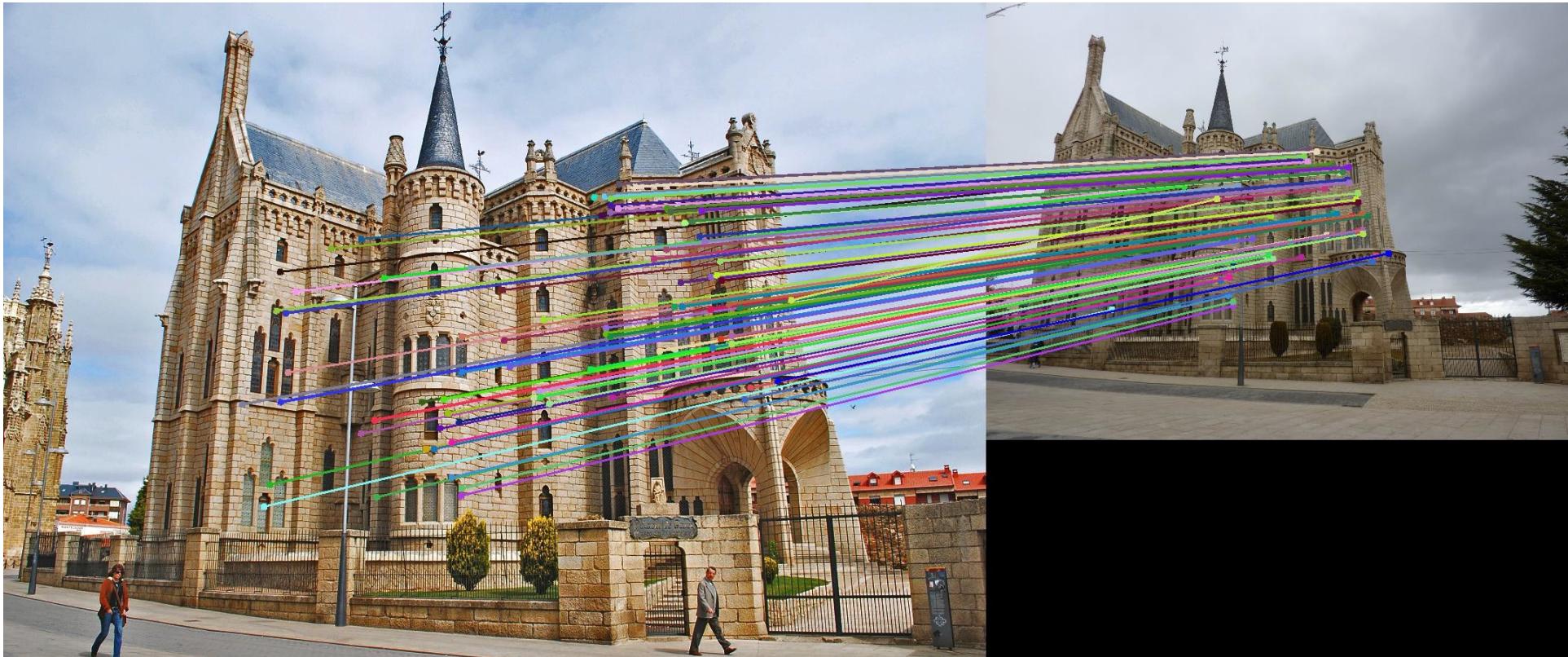


Algorithm:

1. **Sample** (randomly) the number of points required to fit the model ($s=2$)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

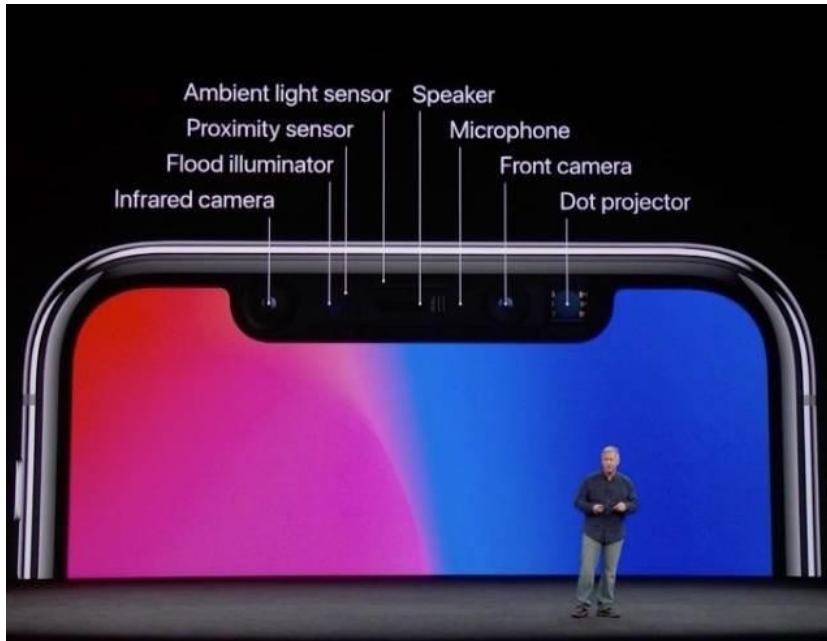
Keep only the matches that are “inliers” with respect to the “best” fundamental matrix



Recap: epipolar geometry & camera calibration

- 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”

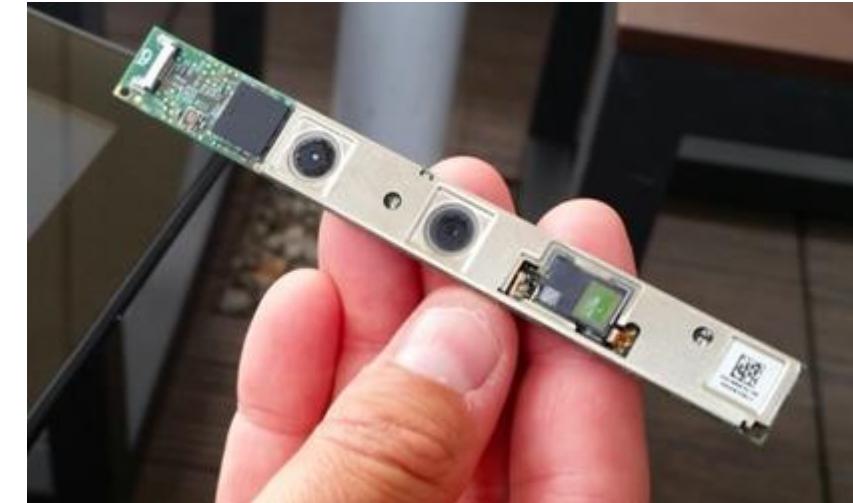
Depth and Camera



iPhone X



Microsoft Kinect v1



Intel laptop depth camera



What's different between these two images?

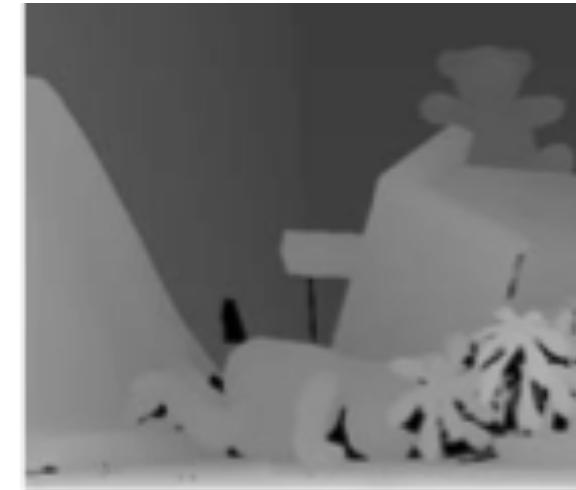




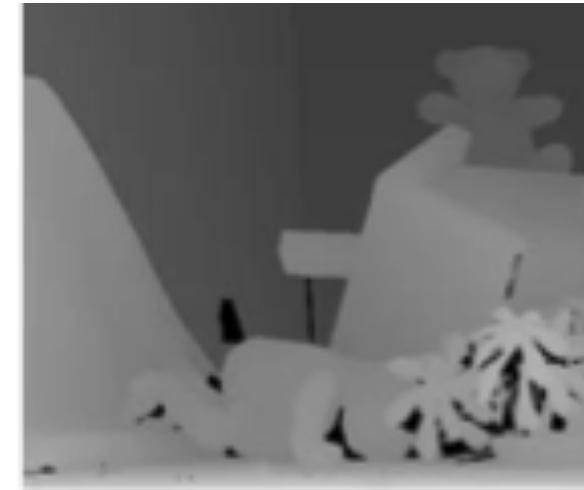


Objects that are close move more or less?

The amount of horizontal movement is
inversely proportional to ...

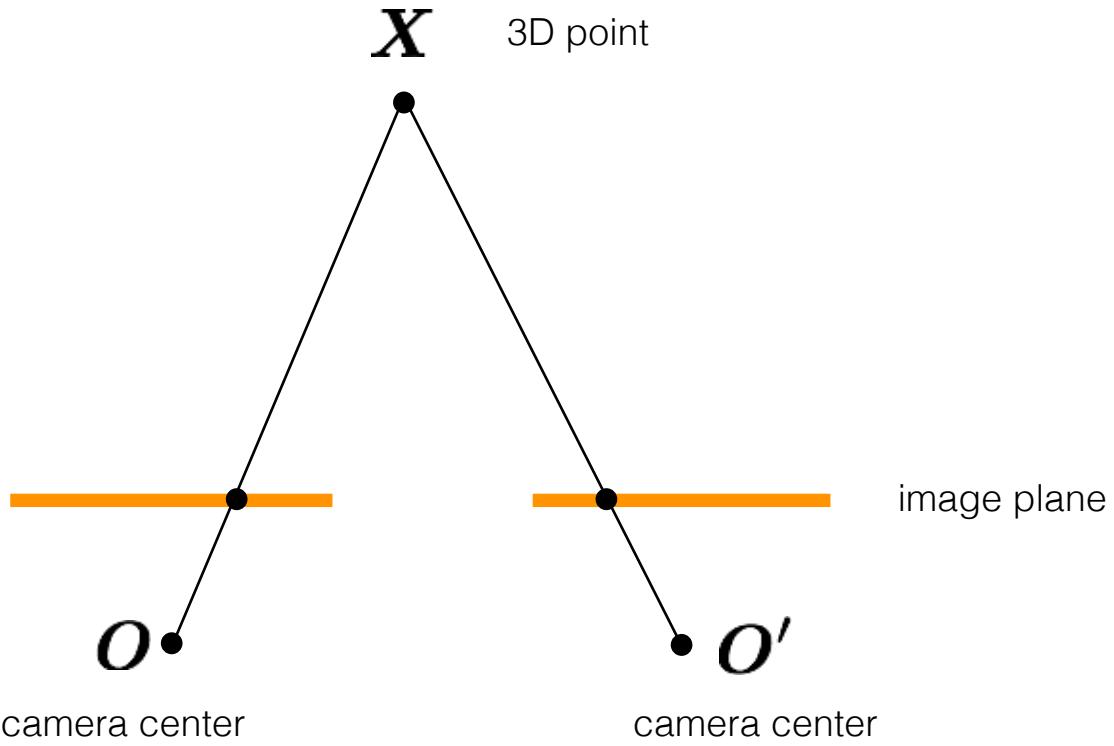


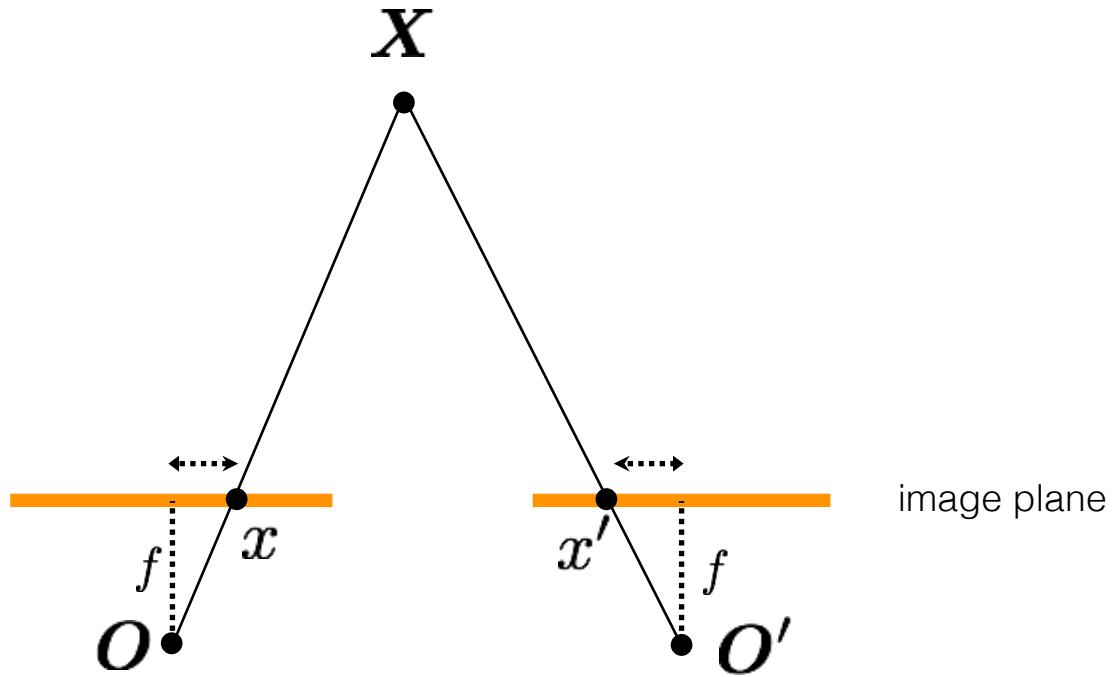
The amount of horizontal movement is
inversely proportional to ...

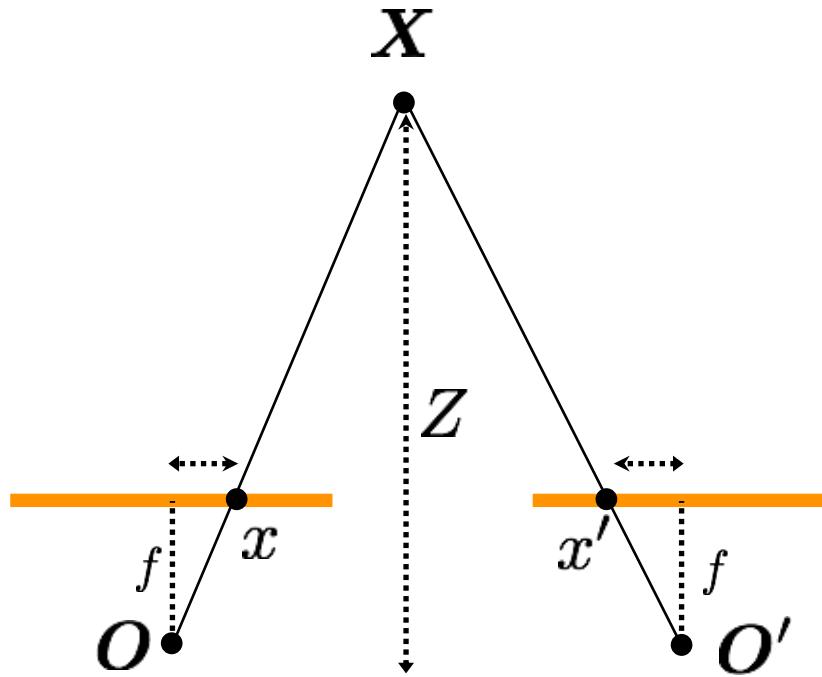


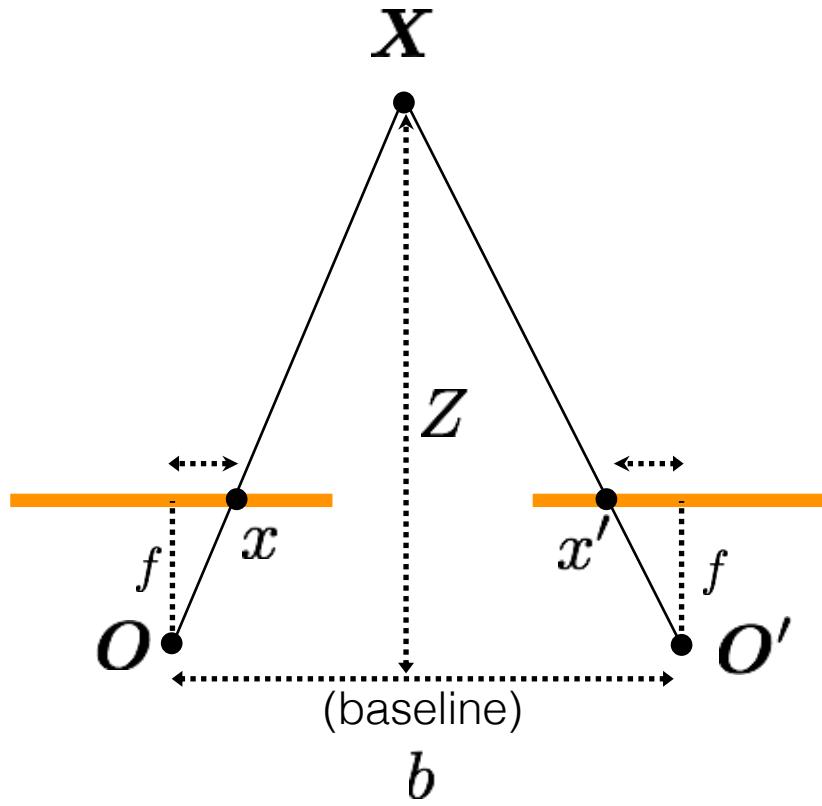
... the distance from the camera.

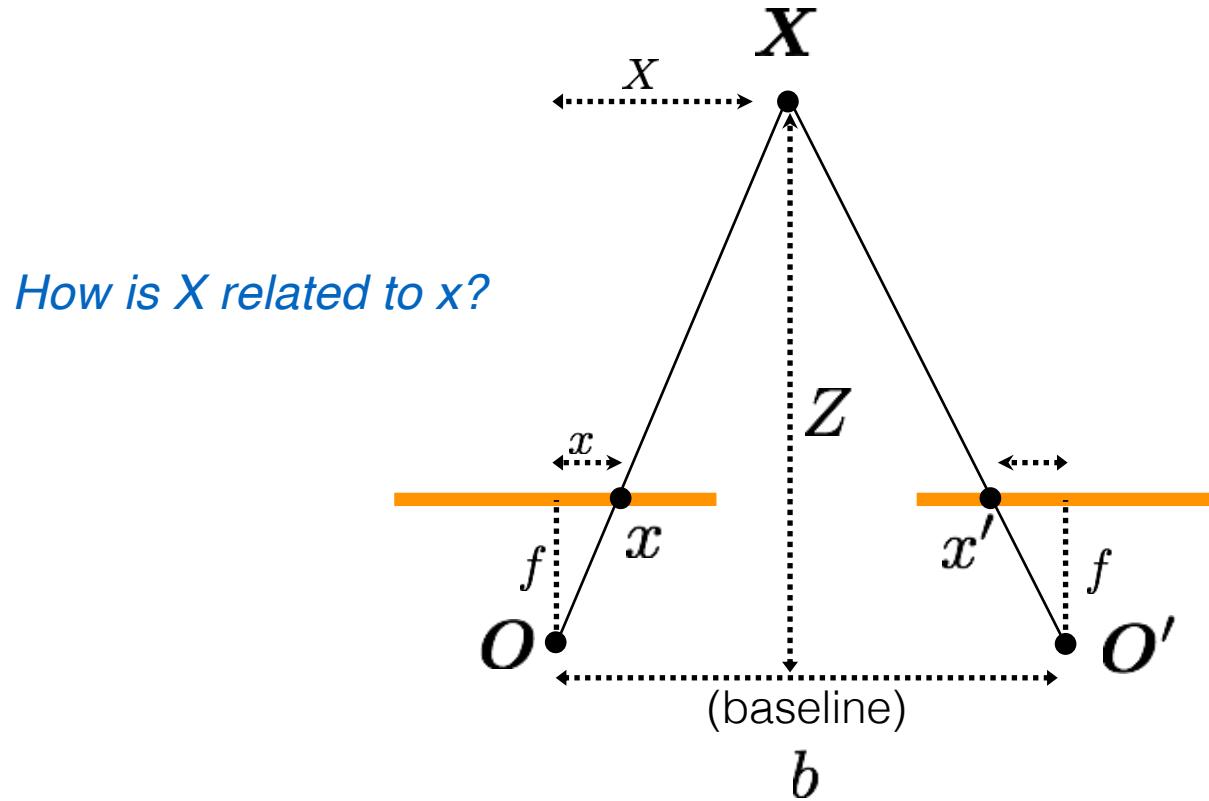
More formally...



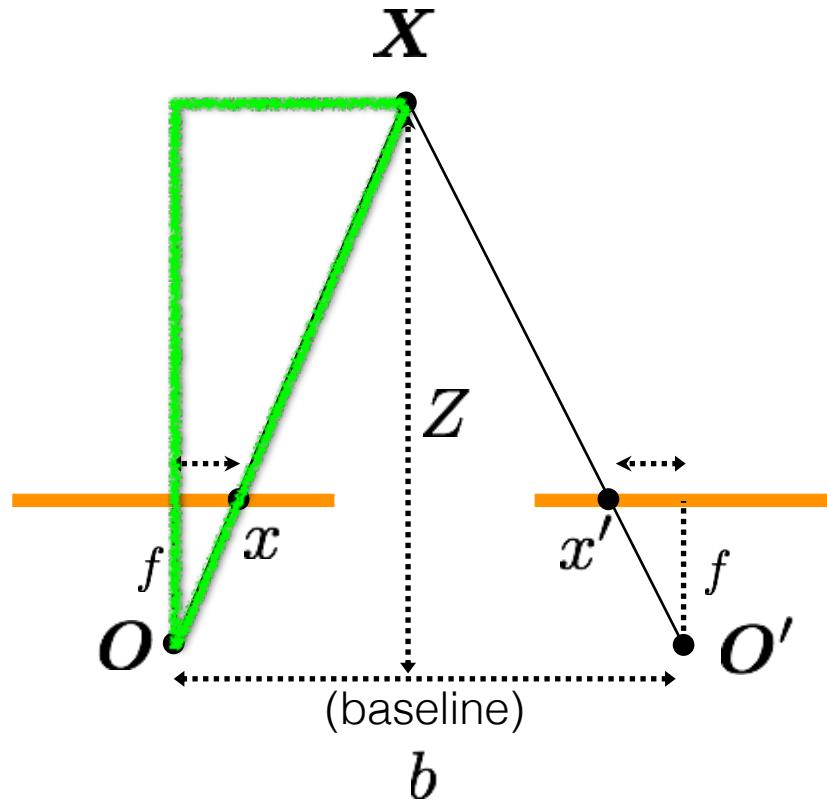




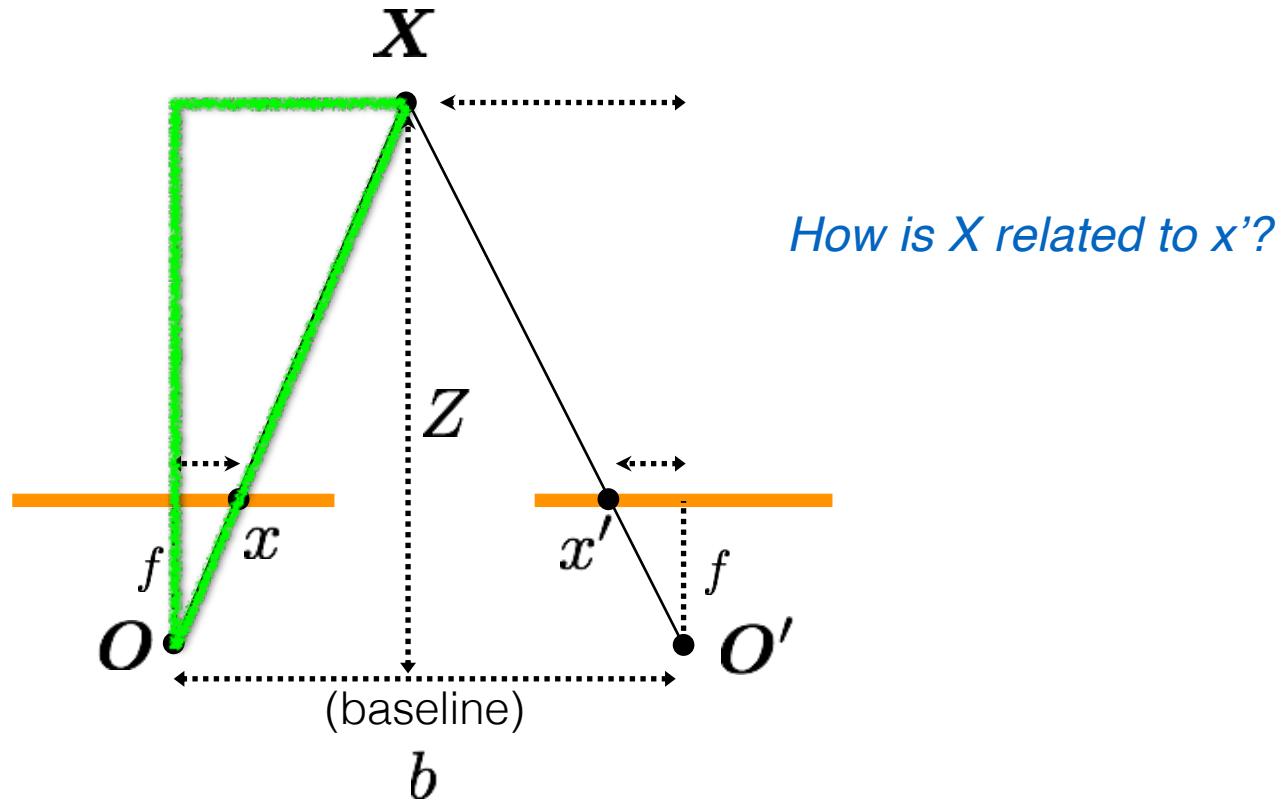




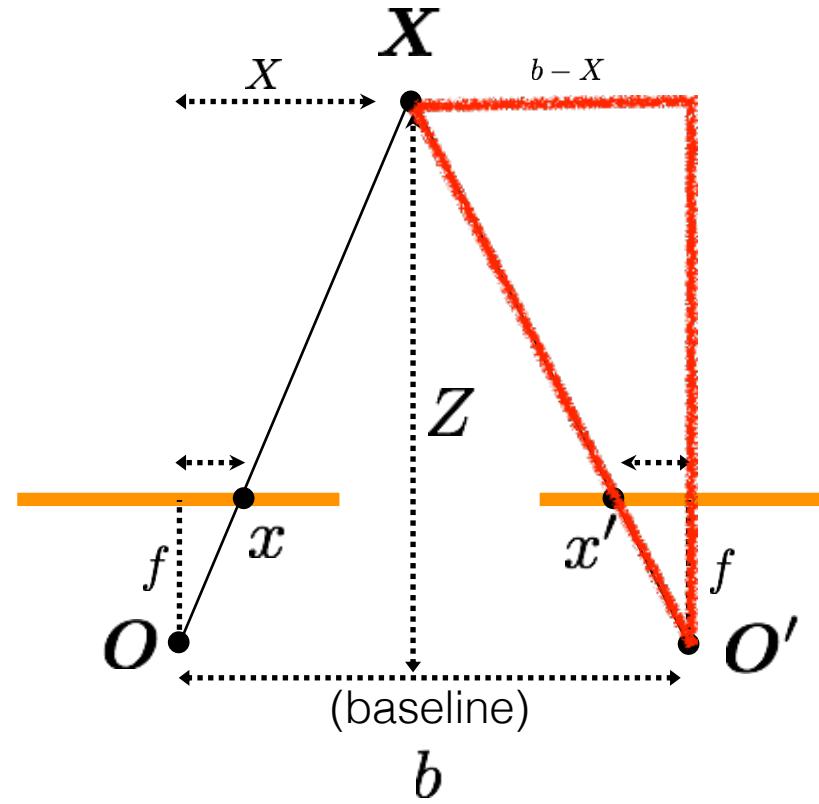
$$\frac{X}{Z} = \frac{x}{f}$$



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



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

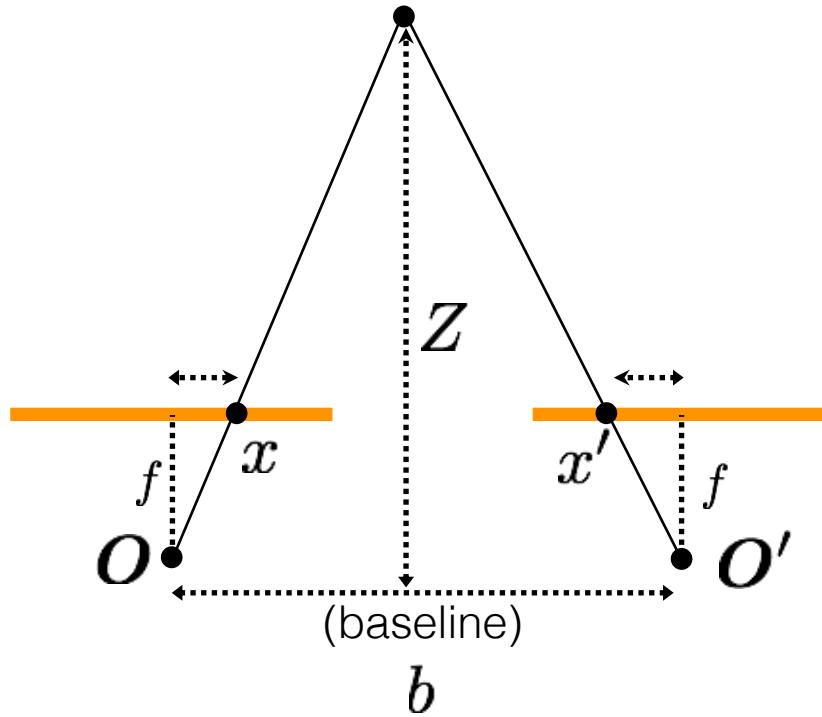


$$\frac{b - X}{Z} = \frac{x'}{f}$$

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

X

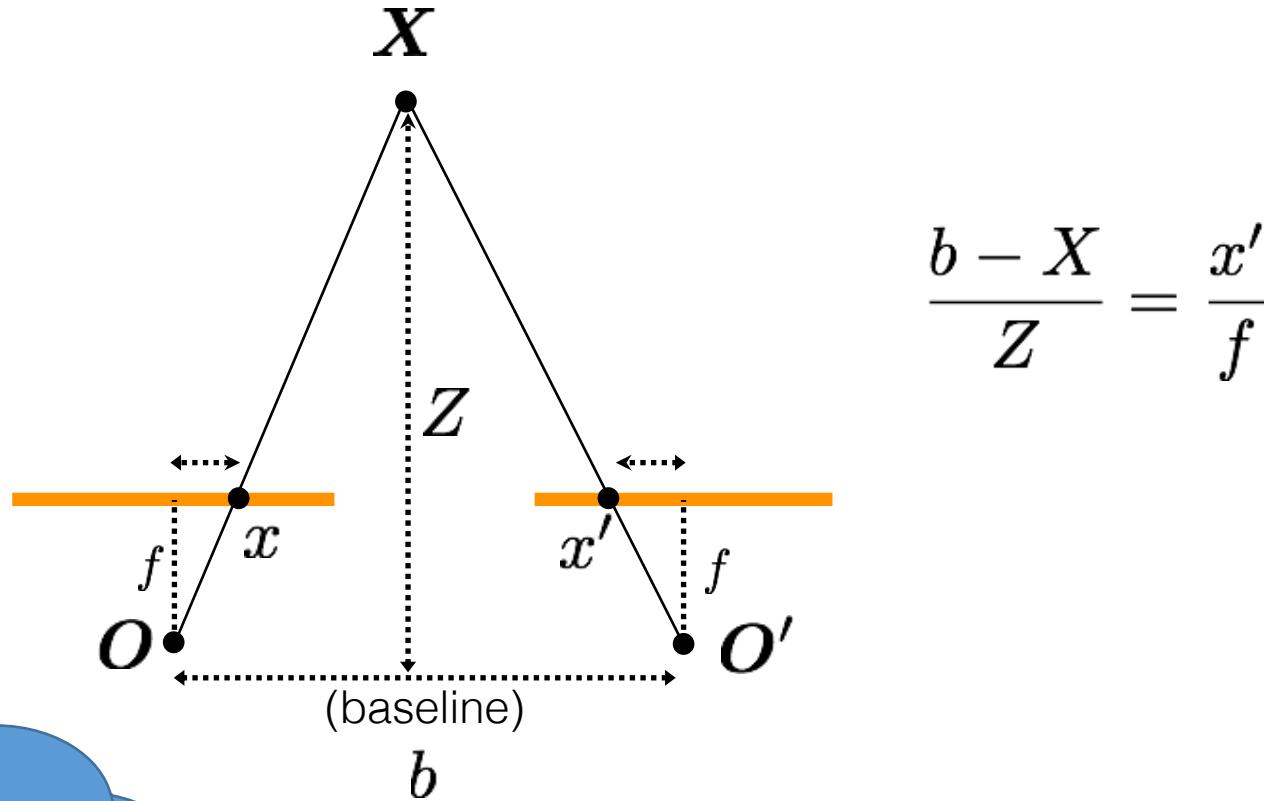
$$\frac{b - X}{Z} = \frac{x'}{f}$$



Disparity

$$d = x - x' \quad (\text{wrt to camera origin of image plane})$$
$$= \frac{bf}{Z}$$

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



So, if I know x and x' , I can compute depth!!

Disparity

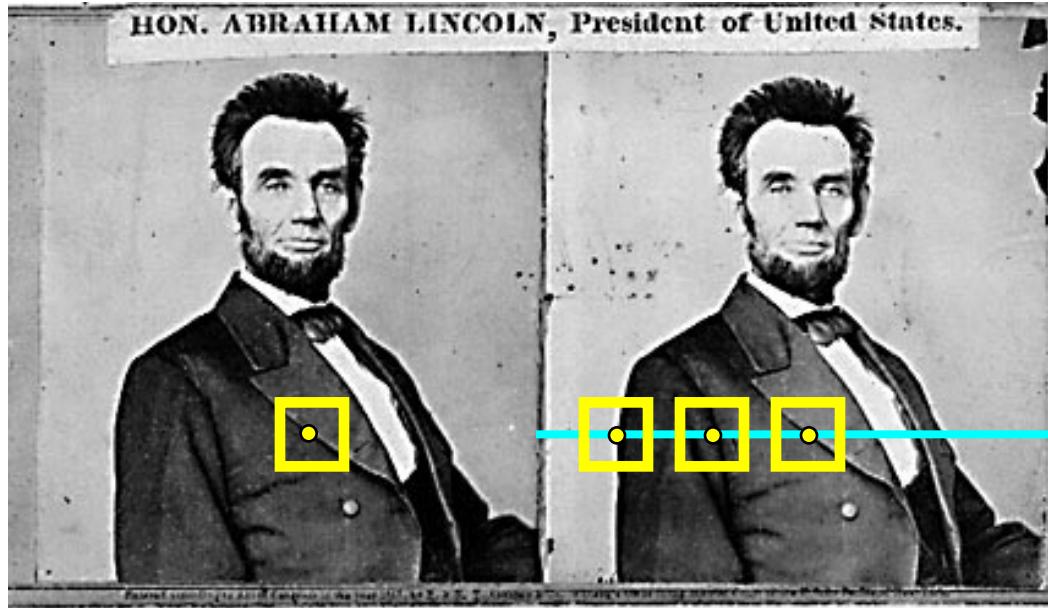
$$d = x - x'$$
$$= \frac{bf}{Z}$$

inversely proportional
to depth



Depth Estimation via Stereo Matching





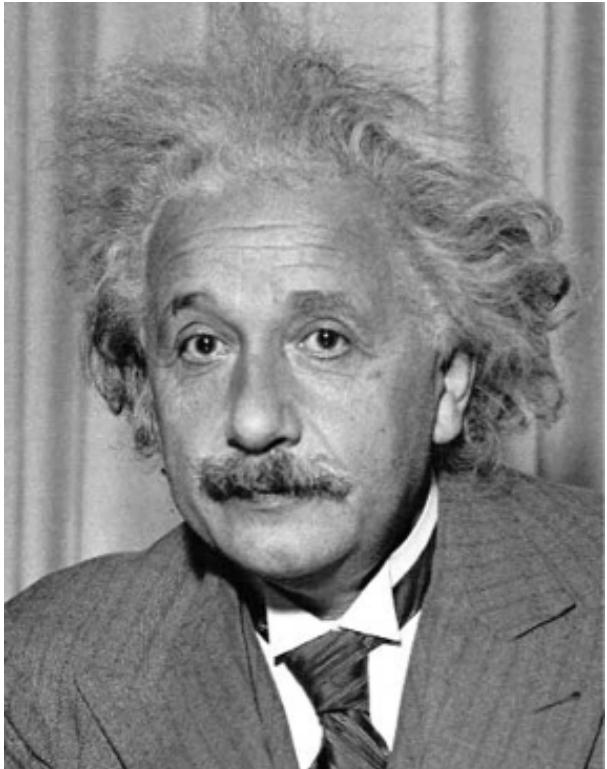
1. Rectify images
(make epipolar lines horizontal)
2. For each pixel
 - a. Find epipolar line
 - b. Scan line for best match
 - c. Compute depth from disparity

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

How would
you do this?
**Template
Matching**

Find this template

How do we detect the template  in the following image?



What will
the output
look like?

output

$$h[m, n] = \frac{\sum_{k,l} (g[k, l] - \bar{g})(f[m + k, n + l] - \bar{f}_{m,n})}{\sqrt{(\sum_{k,l} (g[k, l] - \bar{g})^2 \sum_{k,l} (f[m + k, n + l] - \bar{f}_{m,n})^2)}}$$

filter 

template mean

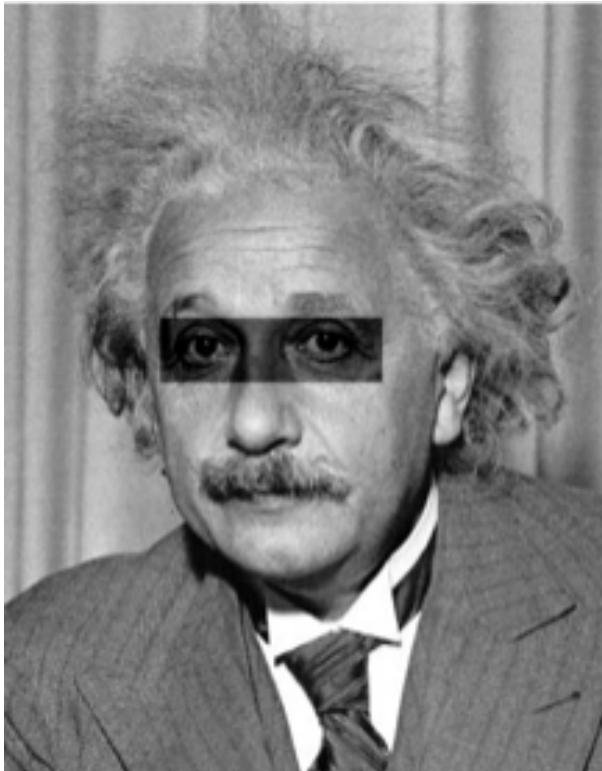
image

local patch mean

Normalized cross-correlation (NCC).

Find this template

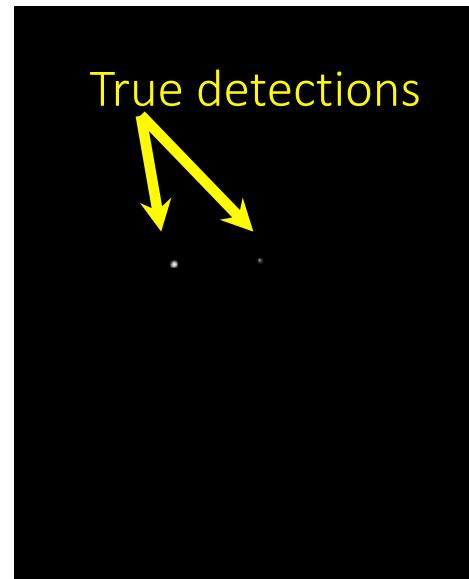
How do we detect the template  in the following image?



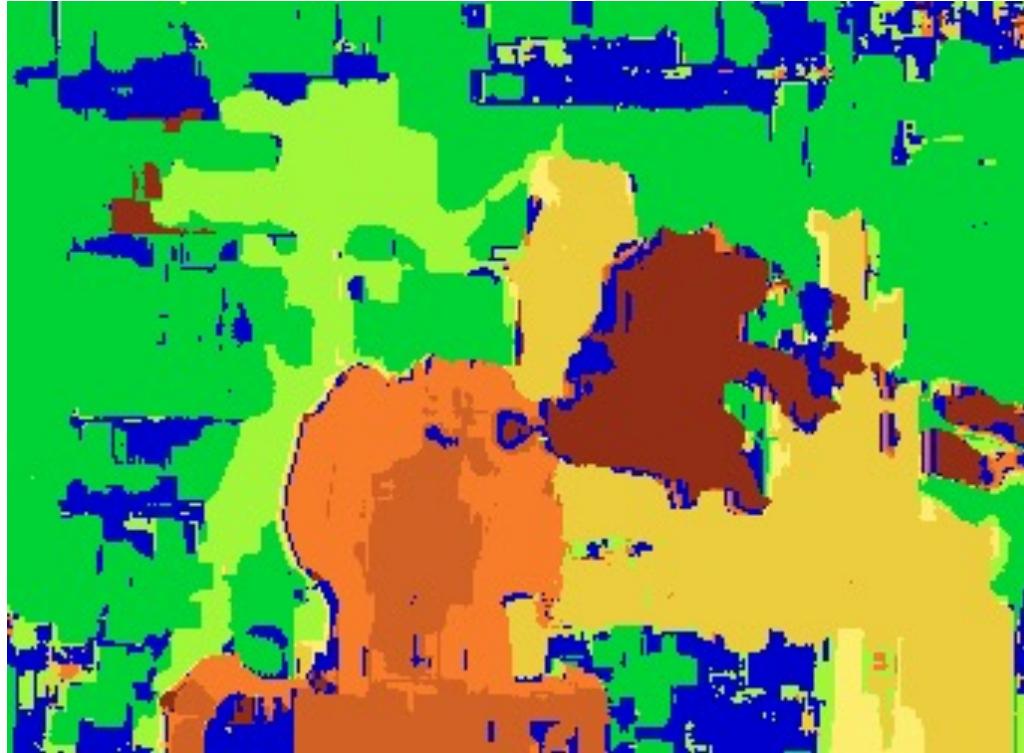
1-output



thresholding



Normalized cross-correlation (NCC).



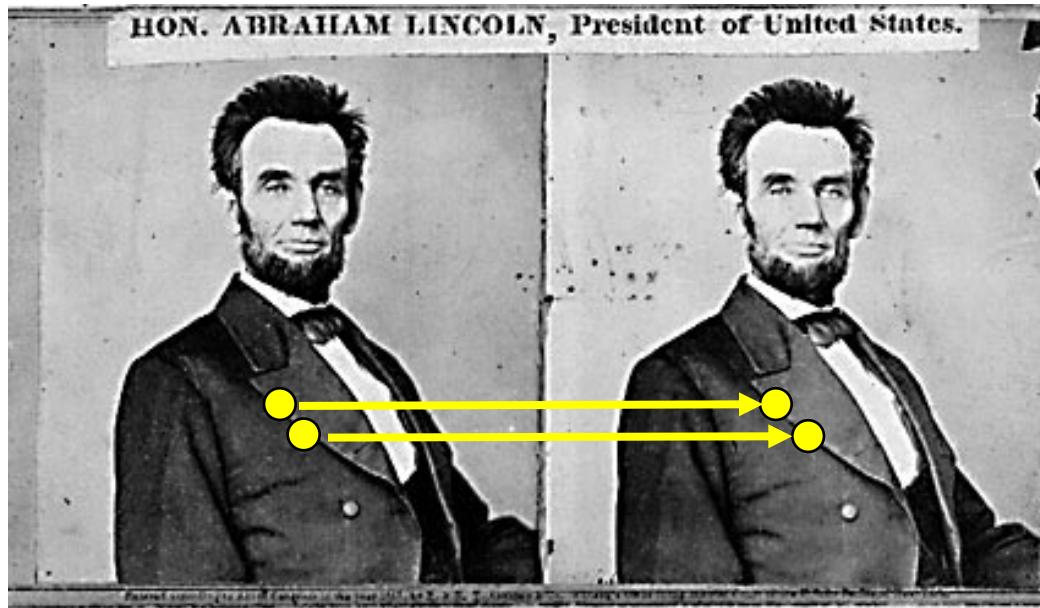
How can we improve depth estimation?

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)

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

data term

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 $J(x + d(x, y), y)$

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

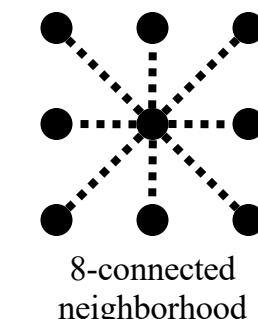
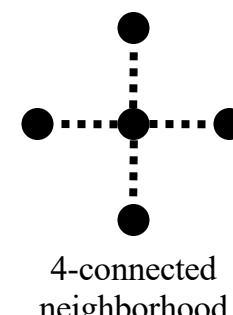
$$E_d(d) = \sum_{(x,y) \in I} C(x, y, d(x, y))$$

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

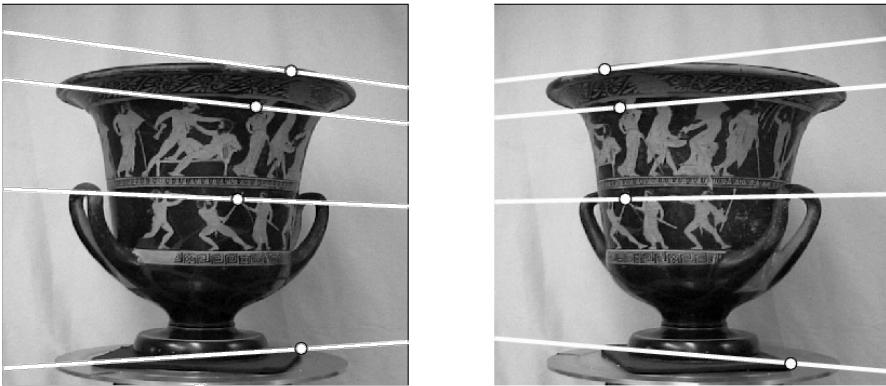
$$E_s(d) = \sum_{(p,q) \in \mathcal{E}} V(d_p, d_q)$$

smoothness term

\mathcal{E} : set of neighboring pixels



Can we compute depth from any two images of the same object?



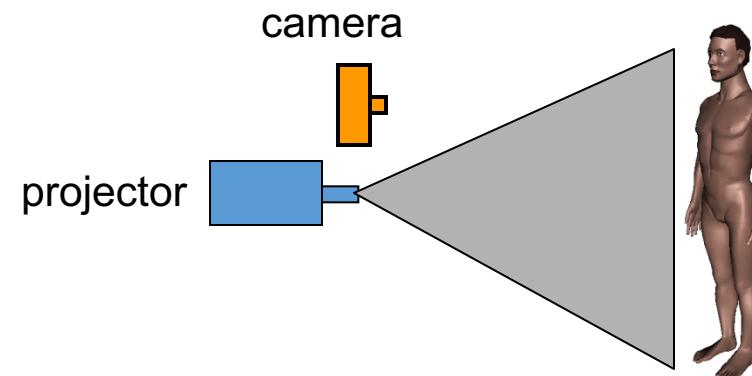
Images need to be ‘rectified’ first

Stereo Rectification: make epipolar lines horizontal

Active stereo with structured light



- Project “structured” light patterns onto the object
 - Simplifies the correspondence problem
 - Allows us to use only one camera



L. Zhang, B. Curless, and S. M. Seitz. [Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming](#). 3DPVT 2002

Kinect: Structured infrared light



<http://bbzippo.wordpress.com/2010/11/28/kinect-in-infrared/>

iPhone X



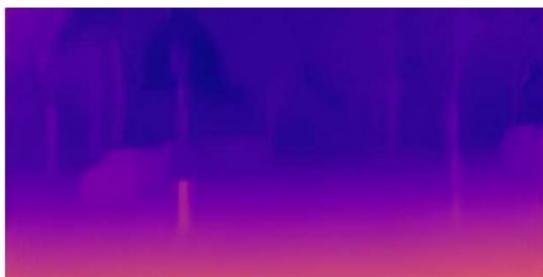
iPhone 12 has lidar



“Semantic” Depth Estimation



(a) Input Image



(b) Baseline Disparity Map



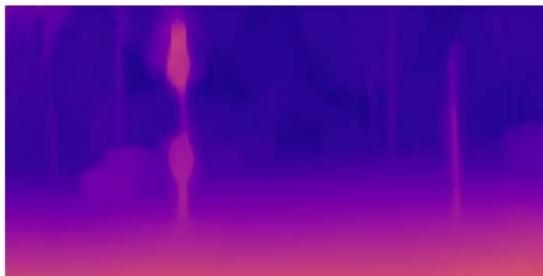
(a) Input Image



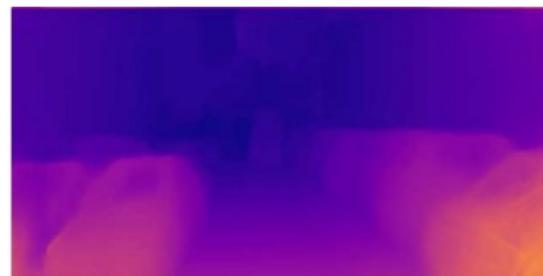
(b) Baseline Semantic Map



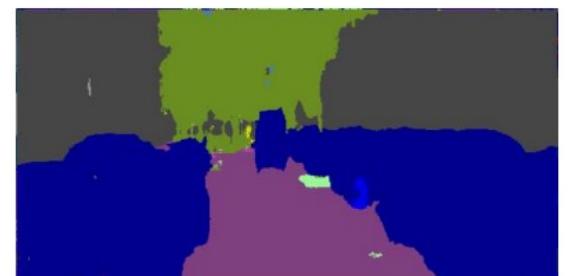
(c) SceneNet Semantic Map



(d) SceneNet Disparity Map



(c) SceneNet Disparity Map



(d) SceneNet Semantic Map



The University of Texas at Austin
**Electrical and Computer
Engineering**
Cockrell School of Engineering