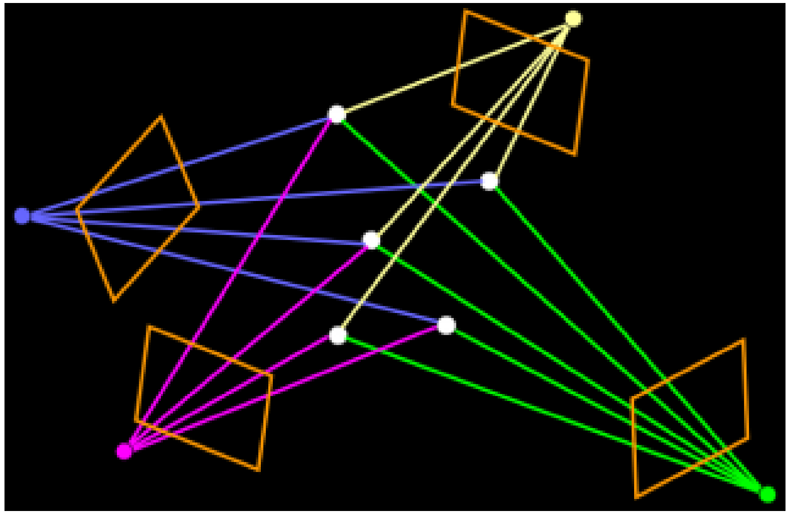
Camera Calibration
+
Multiview Geometry

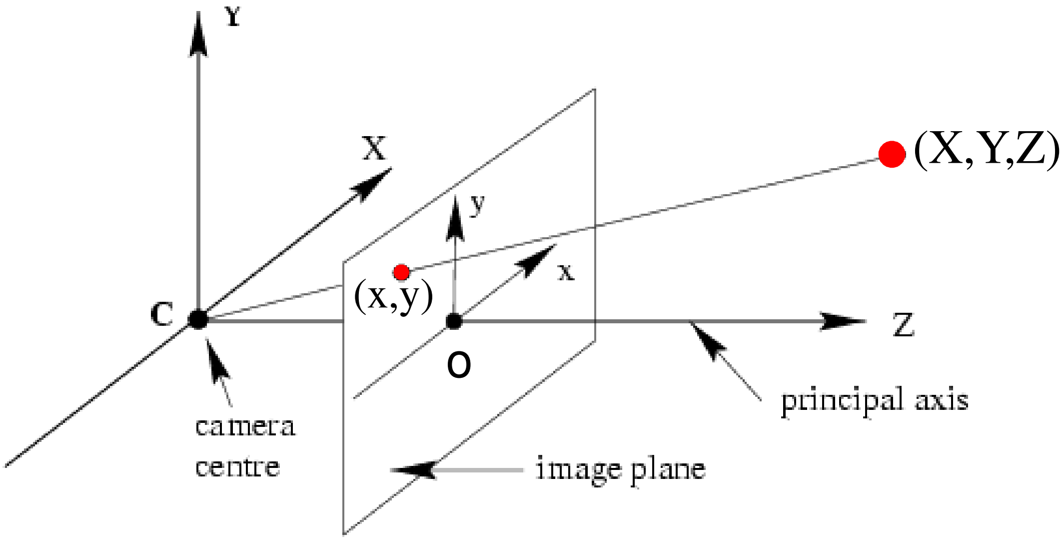


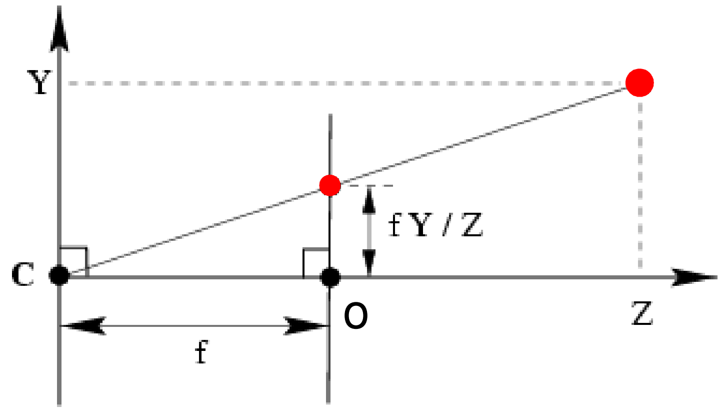
slides are courtesy of Svetlana Lazebnik

Projective transformation:

from world coordinates to image coordinates

Modeling projection

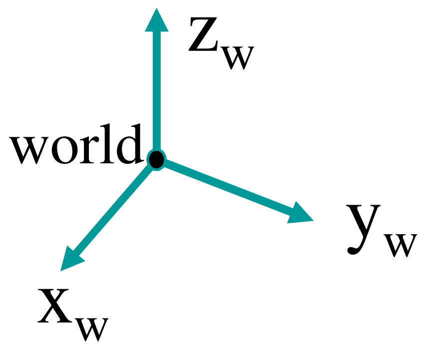




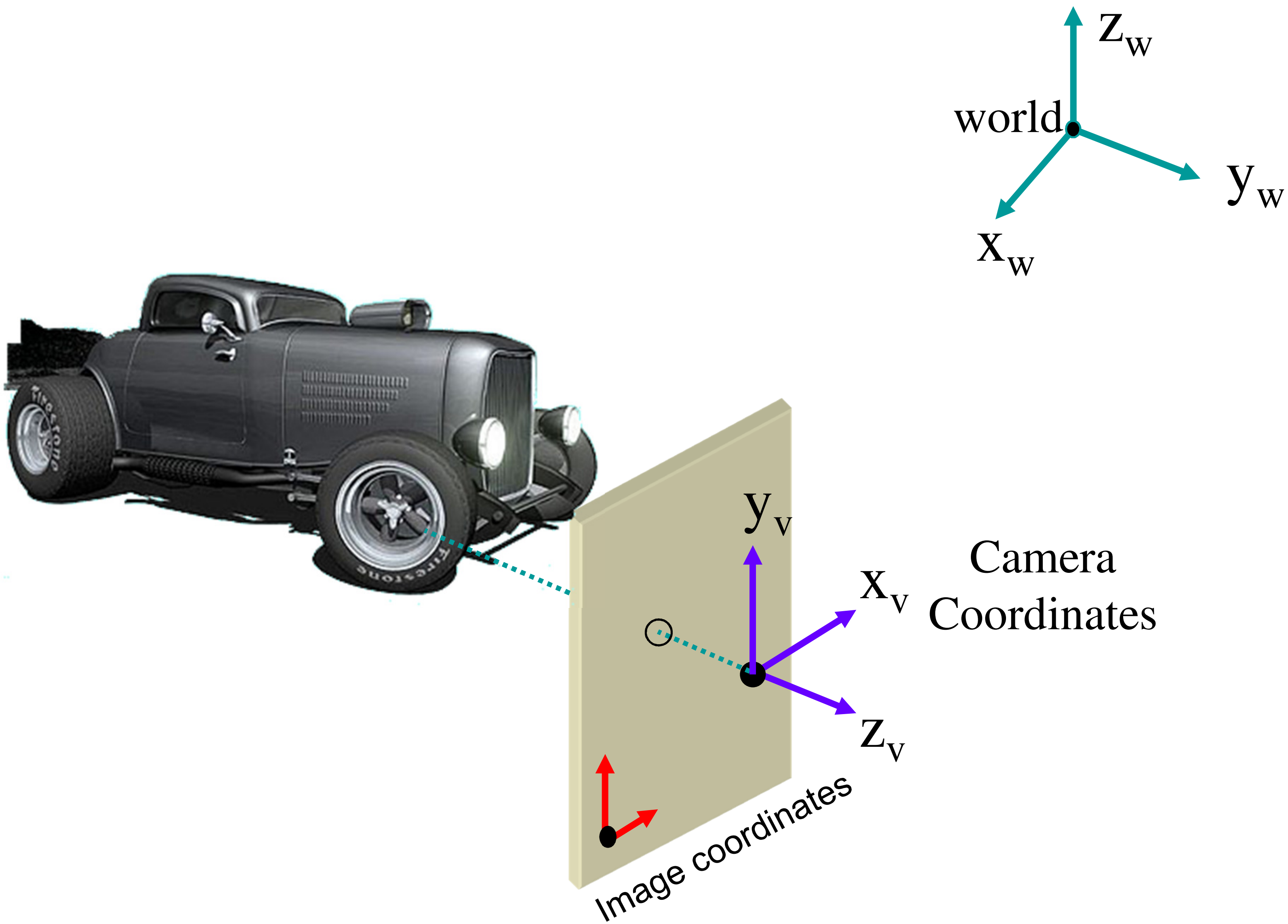
$$(X,Y,Z)^T \mapsto (fX/Z, fY/Z)^T$$

This is the Camera Model viewed from the camera point of view, ie Camera Coordinates.

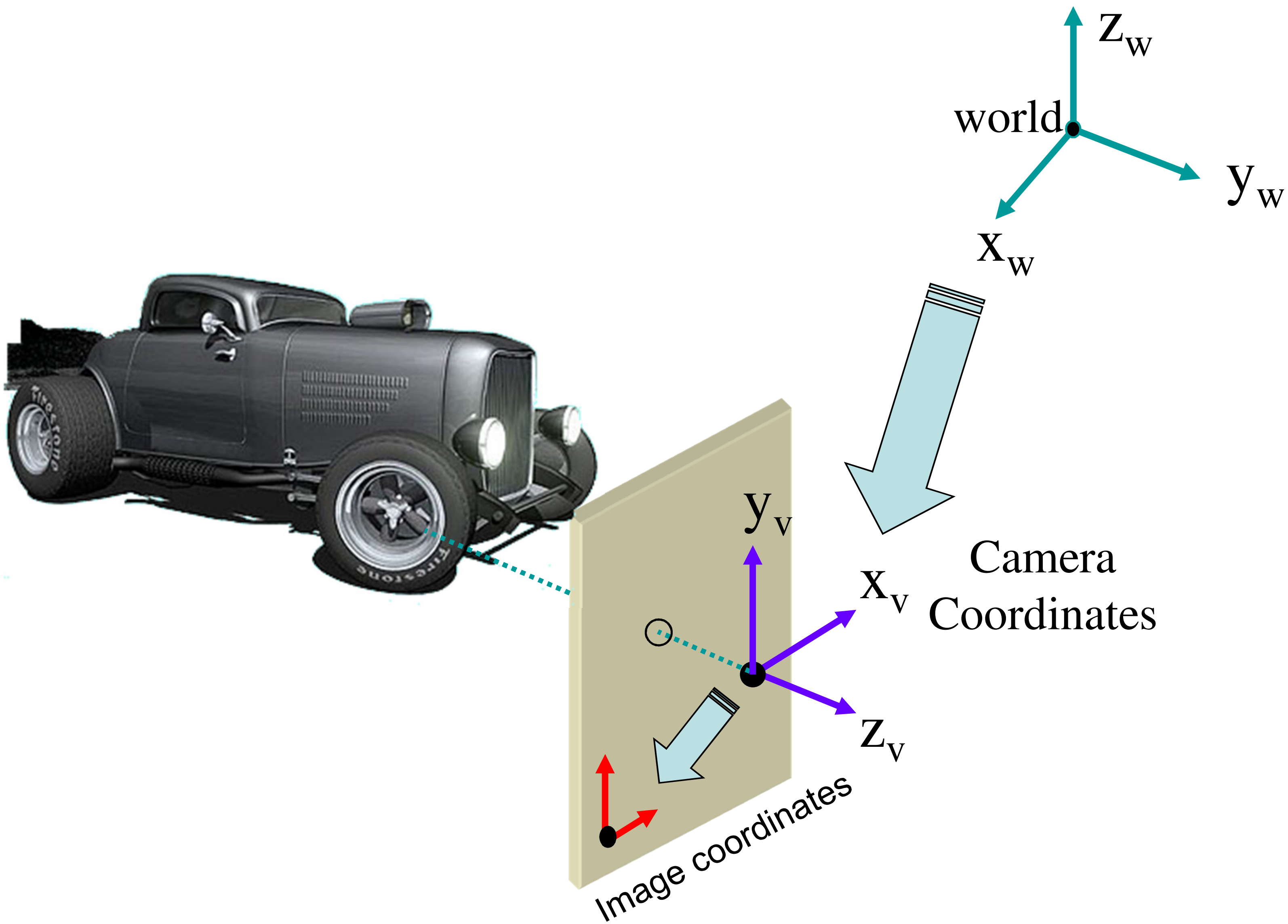
Modeling Projection



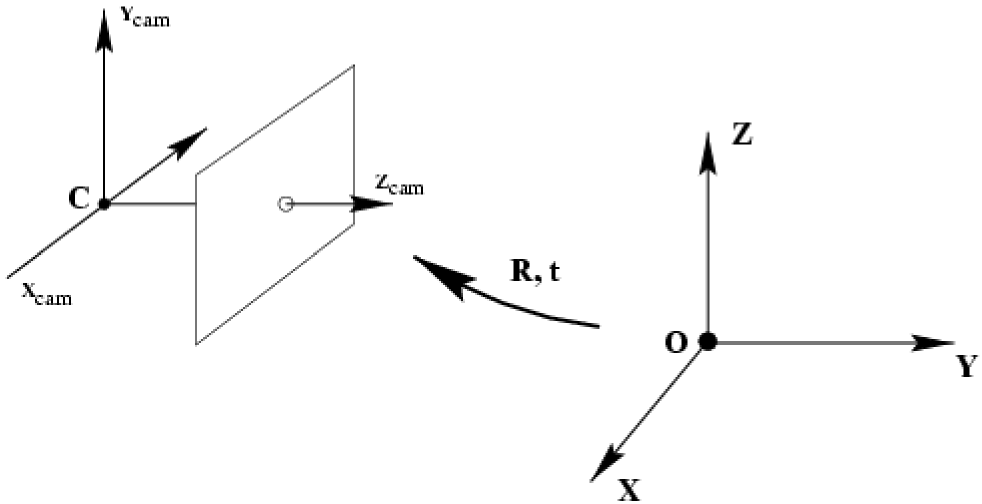
Modeling Projection



Modeling Projection



Camera Parameters



$$\lambda \mathbf{x} = \mathbf{P}\mathbf{X} \qquad \mathbf{P} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]$$

camera calibration (K) — intrinsic

rotation (R), translation (t) from world to camera coordinate frame — extrinsic

Recap: Homogeneous Coordinates

- Homogeneous Coordinates is a mapping from R^n to R^{n+1}:

$$(x, y) \rightarrow (X, Y, W) \equiv (tx, ty, t)$$

- Note: (tx, ty, t) all correspond to the same non-homogeneous point (x, y). E.g. $(2,3,1) \equiv (6,9,3) \equiv (4,6,2)$.

- Inverse mapping:

$$(X, Y, W) \rightarrow \left(\frac{X}{W}, \frac{Y}{W}\right) = (\mathrm{x}, \mathrm{y})$$

Recap: Homogeneous Coordinates

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image coordinates

$$(X, Y, Z) \Rightarrow \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

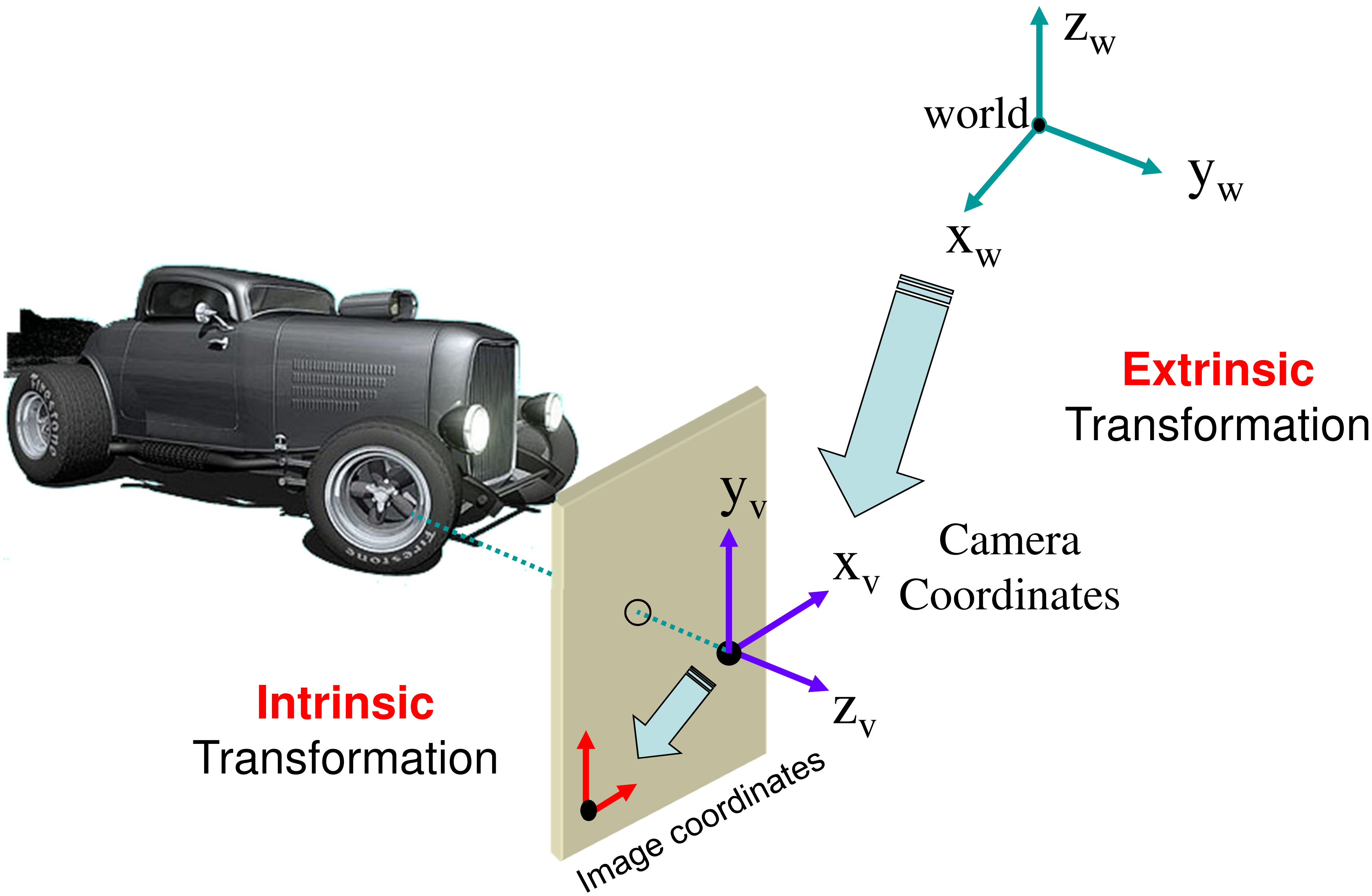
homogeneous scene coordinates

Converting *from* homogeneous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

$$\begin{bmatrix} X \\ Y \\ Z \\ w \end{bmatrix} \Rightarrow (X/w, Y/w, Z/w)$$

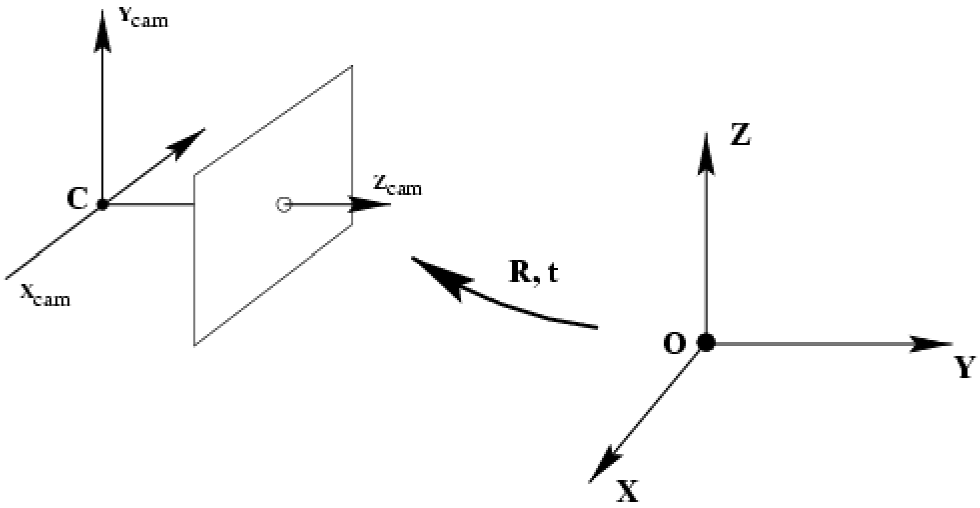
Modeling Projection



Extrinsic Transformation:

from world-coordinates
to
camera-coordinates

Camera Rotation and Translation



In general, the *camera* coordinate frame will be related to the *world* coordinate frame by a rotation and a translation

Conversion from world to camera coordinate system (in non-homogeneous coordinates):

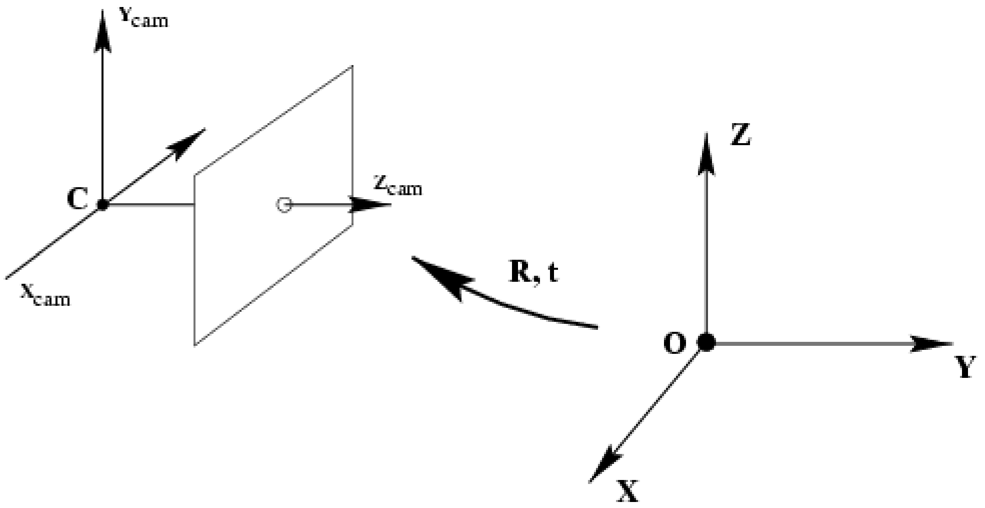
$$\tilde{X}_{cam} = R\left(\tilde{X} - \tilde{C}\right)$$

coords. of point in camera frame

coords. of a point in world frame

coords. of camera center in world frame

Camera Rotation and Translation



In non-homogeneous coordinates:

$$\tilde{X}_{cam} = R\left(\tilde{X} - \tilde{C}\right)$$

$$X_{cam} = \begin{bmatrix} R & -R\tilde{C} \\ \mathbf{0} & 1 \end{bmatrix} \begin{pmatrix} \tilde{X} \\ 1 \end{pmatrix} = \begin{bmatrix} R & t \\ \mathbf{0} & 1 \end{bmatrix} X$$

Extrinsic Transformation

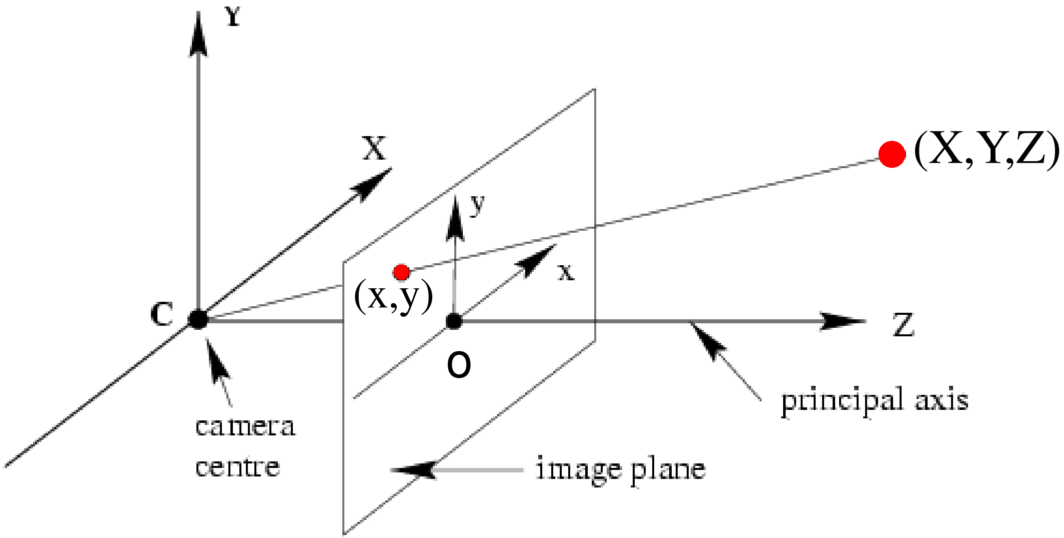
where $t = -R\tilde{C}$

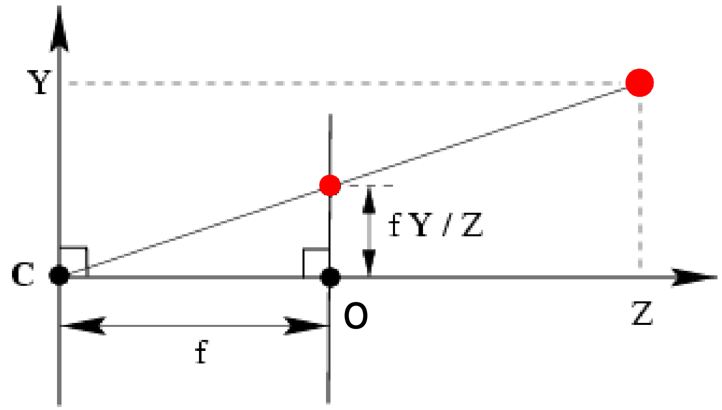
X is $\tilde{X}$ in homogeneous coord.

Intrinsic Transformation:

from camera coordinates
to
image coordinates

Modeling projection

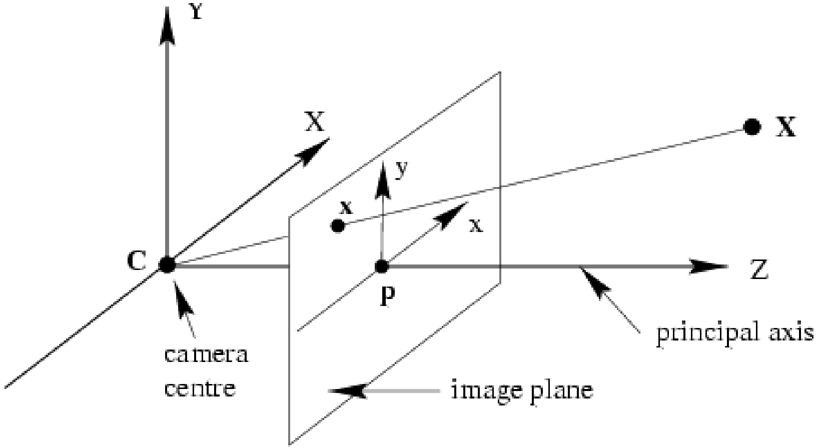


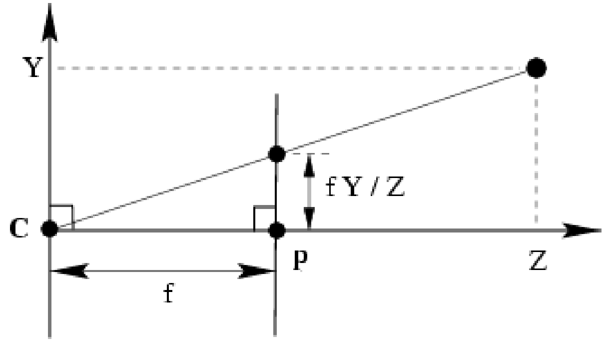


$$(X, Y, Z)^T \mapsto (fX / Z, fY / Z)^T$$

This is the Camera Model viewed from the camera point of view, ie Camera Coordinates.

The Intrinsic Transformation



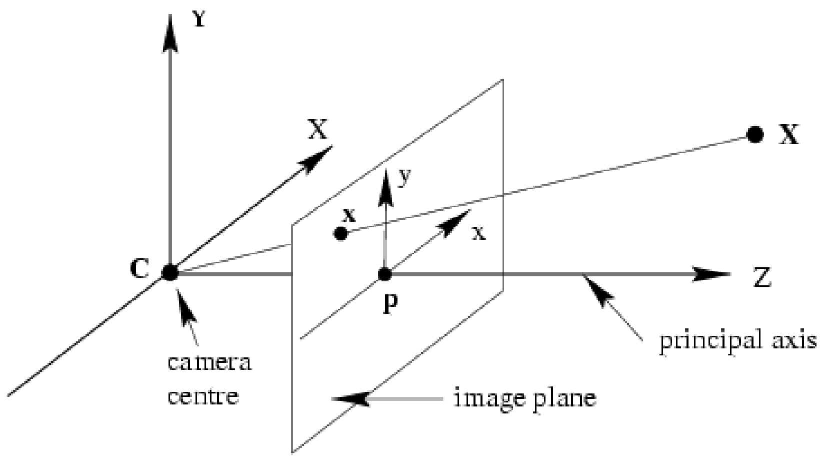


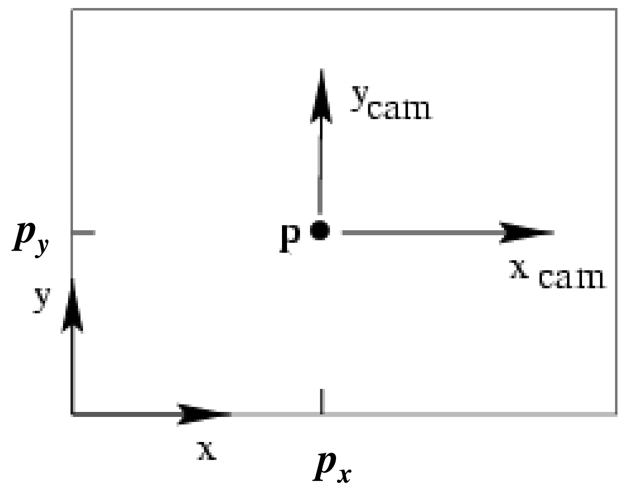
$$(X, Y, Z) \mapsto (f\,X / Z,\, f\,Y / Z)$$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$\mathrm{x} = \mathrm{PX}$$

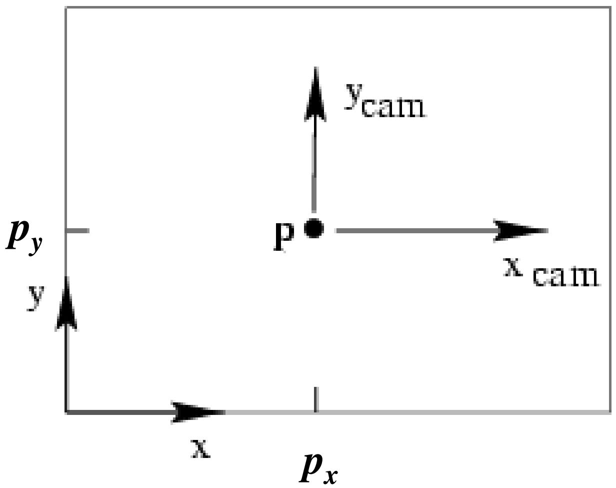
Principal Point





- **Principal point (p):** point where principal axis intersects the image plane
- Normalized coordinate system: origin of the image is at the principal point
- Image coordinate system: origin is in the corner
- How to go from normalized coordinate system to image coordinate system?

Principal Point Offset

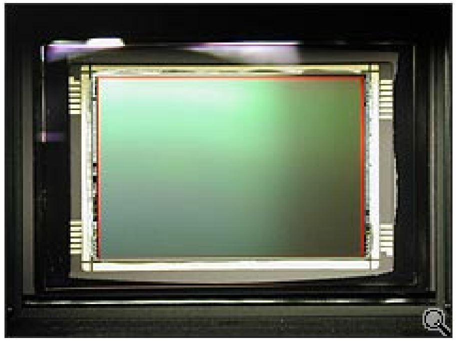


principal point: (p_x, p_y)

$$(X, Y, Z) \mapsto (fX/Z + p_x, fY/Z + p_y)$$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Pixel Coordinates



Pixel size: $\frac{1}{m_x} \times \frac{1}{m_y}$

- m_x pixels per meter in horizontal direction, m_y pixels per meter in vertical direction

$$\begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} = \begin{bmatrix} \alpha_x & & \beta_x & 0 \\ & \alpha_y & \beta_y & 0 \\ & & 1 & 0 \end{bmatrix} = \begin{bmatrix} K & \mathbf{0} \end{bmatrix}$$

pixels/m | m | pixels

Intrinsic Transformation

The projective transformation:

intrinsic + extrinsic

From world to Image transformation

In practice: lots of coordinate transformations…

(2D point (3x1)) = (Camera to pixel coord. trans. matrix (3x3)) (Perspective projection matrix (3x4)) (World to camera coord. trans. matrix (4x4)) (3D point (4x1))

Intrinsic (3x3):
From camera to image

$$\begin{bmatrix} \alpha_x & & \beta_x \\ & \alpha_y & \beta_y \\ & & 1 \end{bmatrix}$$

Extrinsic (3x4):
from world to camera

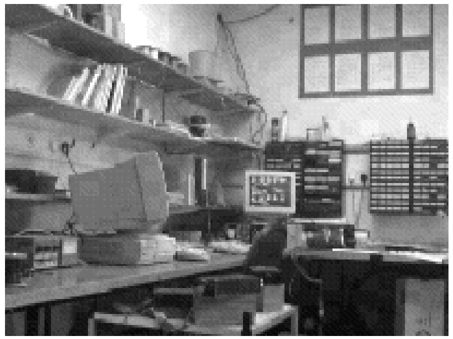
$$\mathrm{x} = \mathrm{P}\mathrm{X}$$

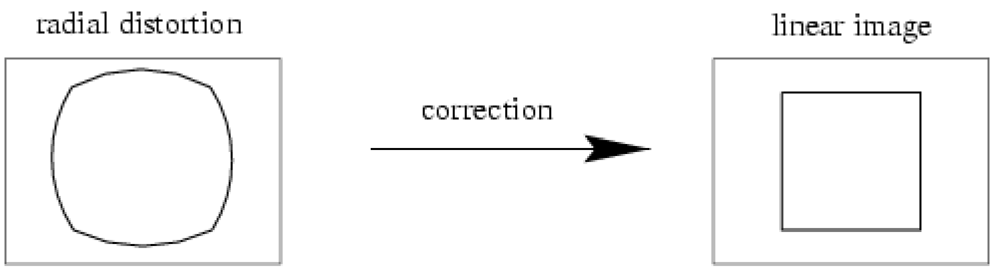
$$\text{where} \quad \mathrm{P} = \begin{bmatrix} K & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathrm{R} & \mathrm{t} \\ \mathbf{0} & 1 \end{bmatrix} = \mathrm{K}\begin{bmatrix} \mathrm{R} & \mathrm{t} \end{bmatrix}$$

Camera parameters $\mathbf{P} = \mathbf{K}\left[\mathbf{R} \quad \mathbf{t}\right]$

- Intrinsic parameters
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew (non-rectangular pixels)*
  - *Radial distortion*

$$\mathbf{K} = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & & \beta_x \\ & \alpha_y & \beta_y \\ & & 1 \end{bmatrix}$$





Camera parameters $\mathbf{P} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]$

- Intrinsic parameters
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew (non-rectangular pixels)*
  - *Radial distortion*
- Extrinsic parameters
  - Rotation and translation relative to world coordinate system

Camera calibration

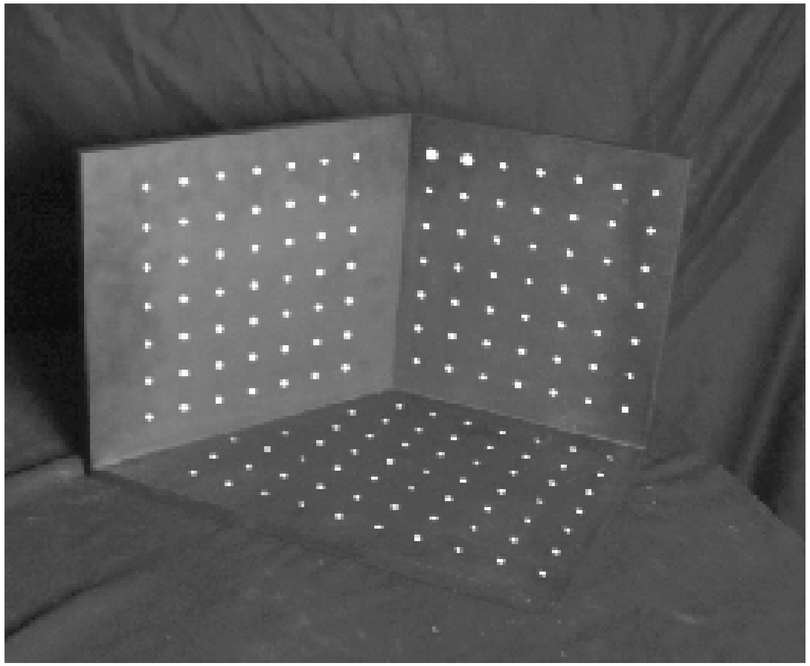
$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

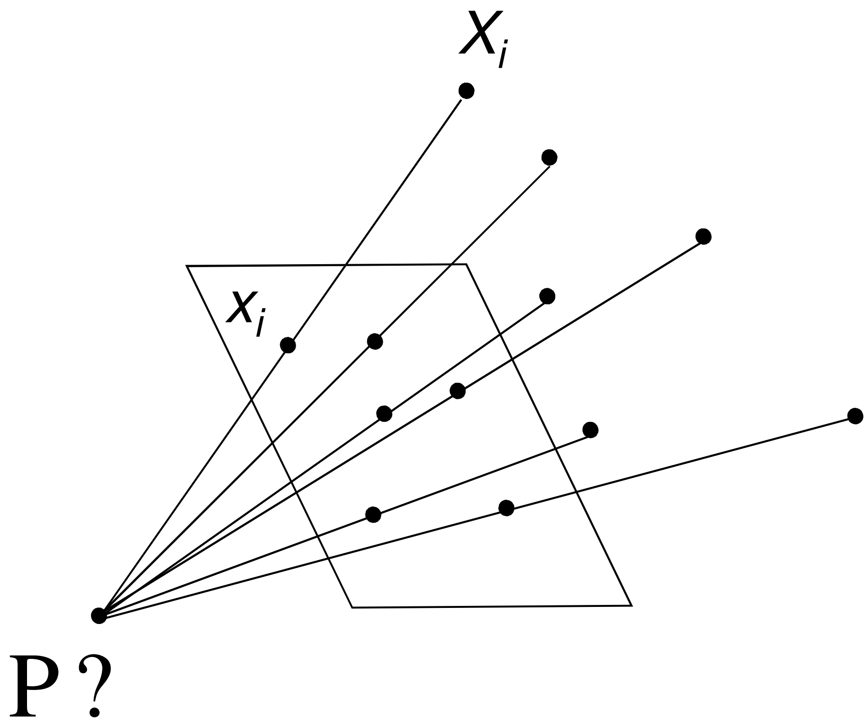
$$\begin{bmatrix}\lambda x \\ \lambda y \\ \lambda\end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Source: D. Hoiem

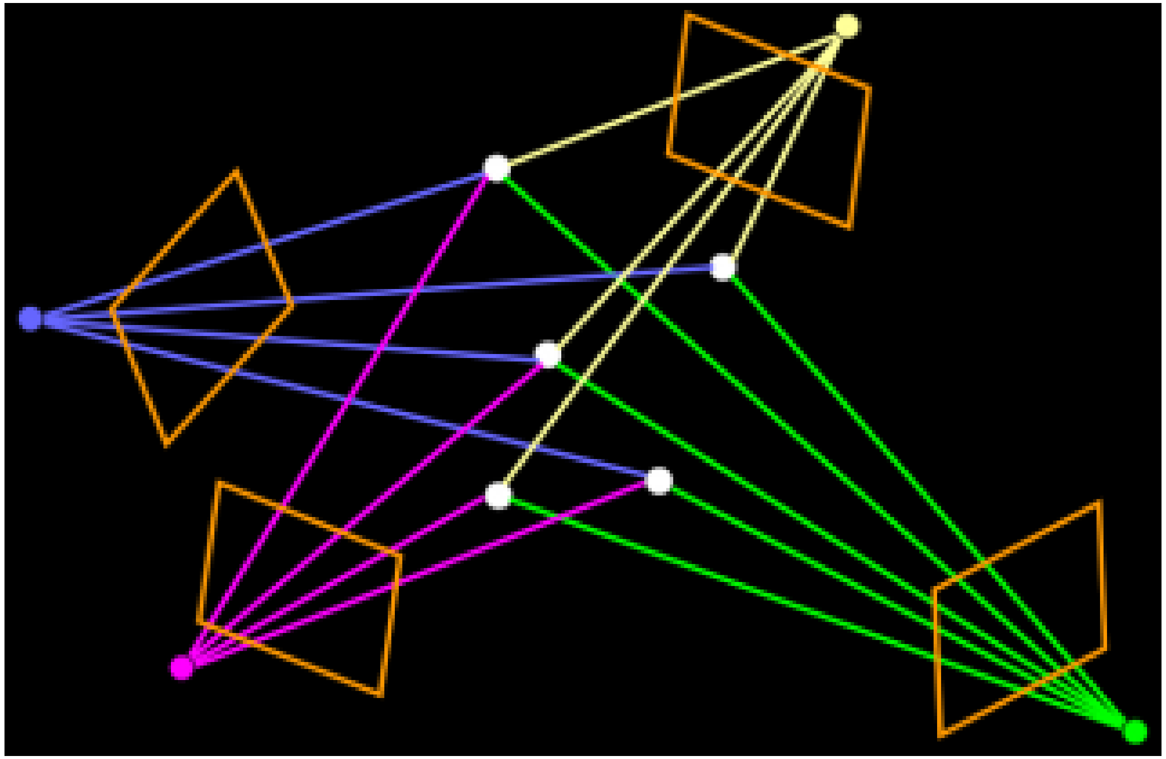
Camera calibration

- Given *n* points with known 3D coordinates $\boldsymbol{X}_i$ and known image projections $\boldsymbol{x}_i$, estimate the camera parameters



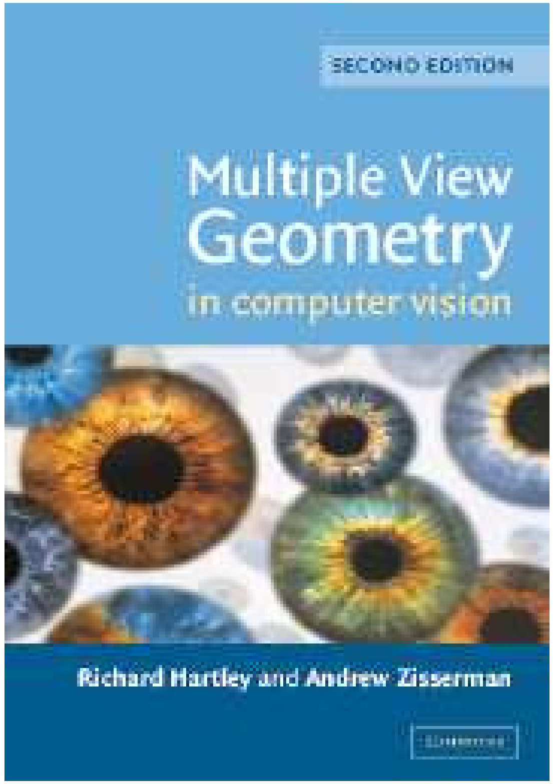


Multiview Geometry



slides are courtesy of Svetlana Lazebnik

Multiview Geometry

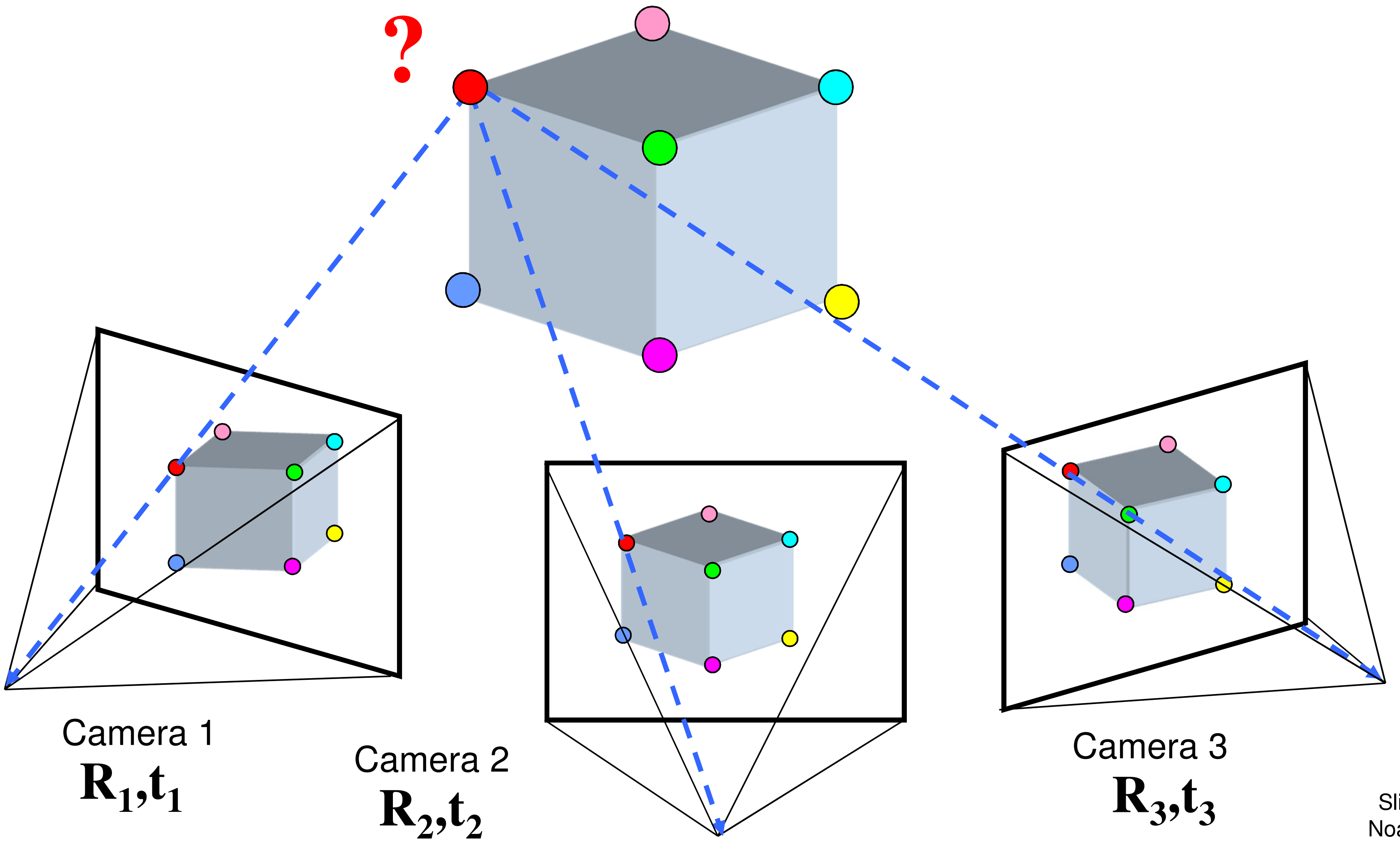


Richard Hartley and Andrew Zisserman,
Cambridge University Press,
March 2004.

Multi-view geometry

Multi-view geometry problems

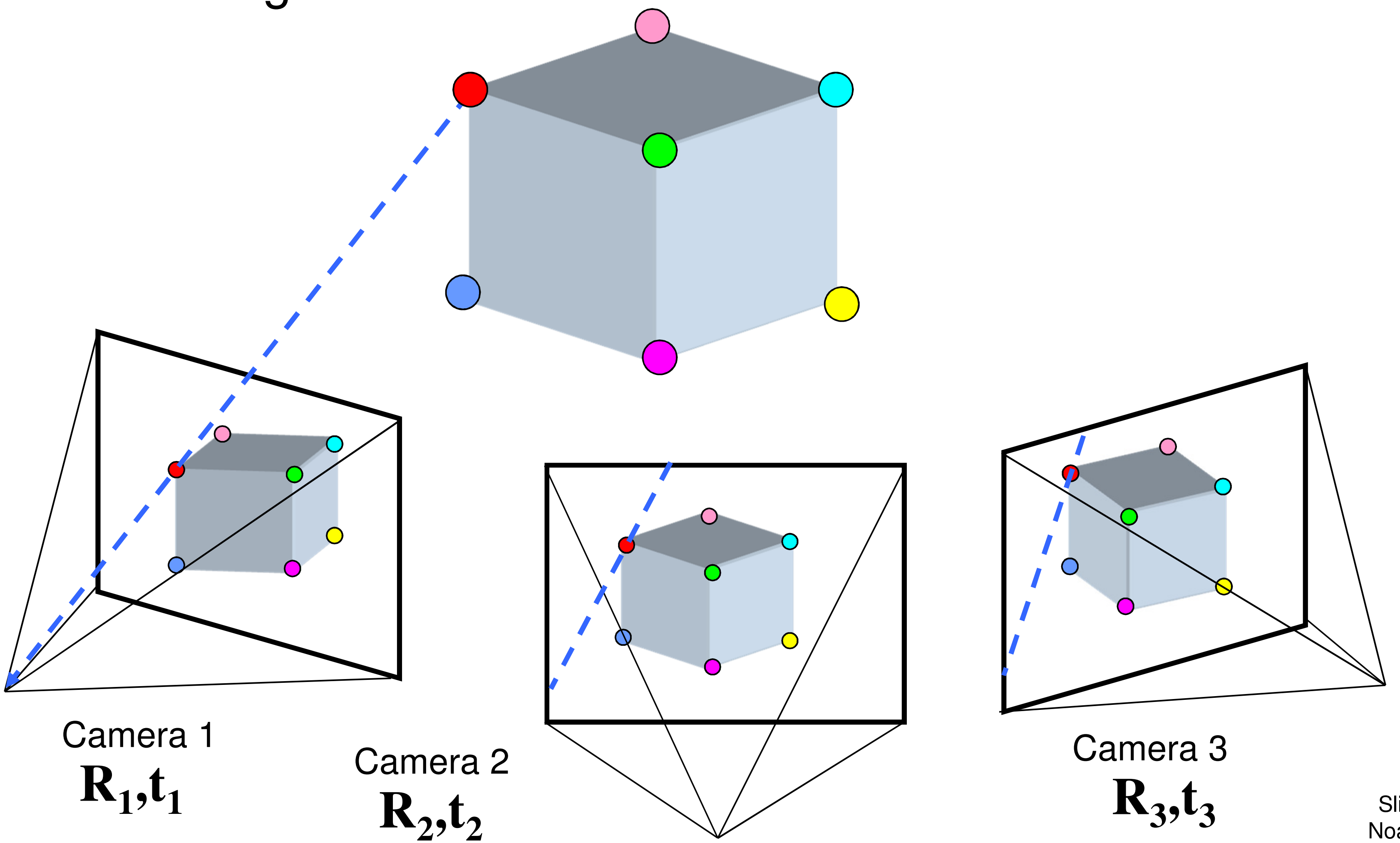
- **Structure:** Given projections of the same 3D point in two or more images, compute the 3D coordinates of that point



Slide credit: Noah Snavely

Multi-view geometry problems

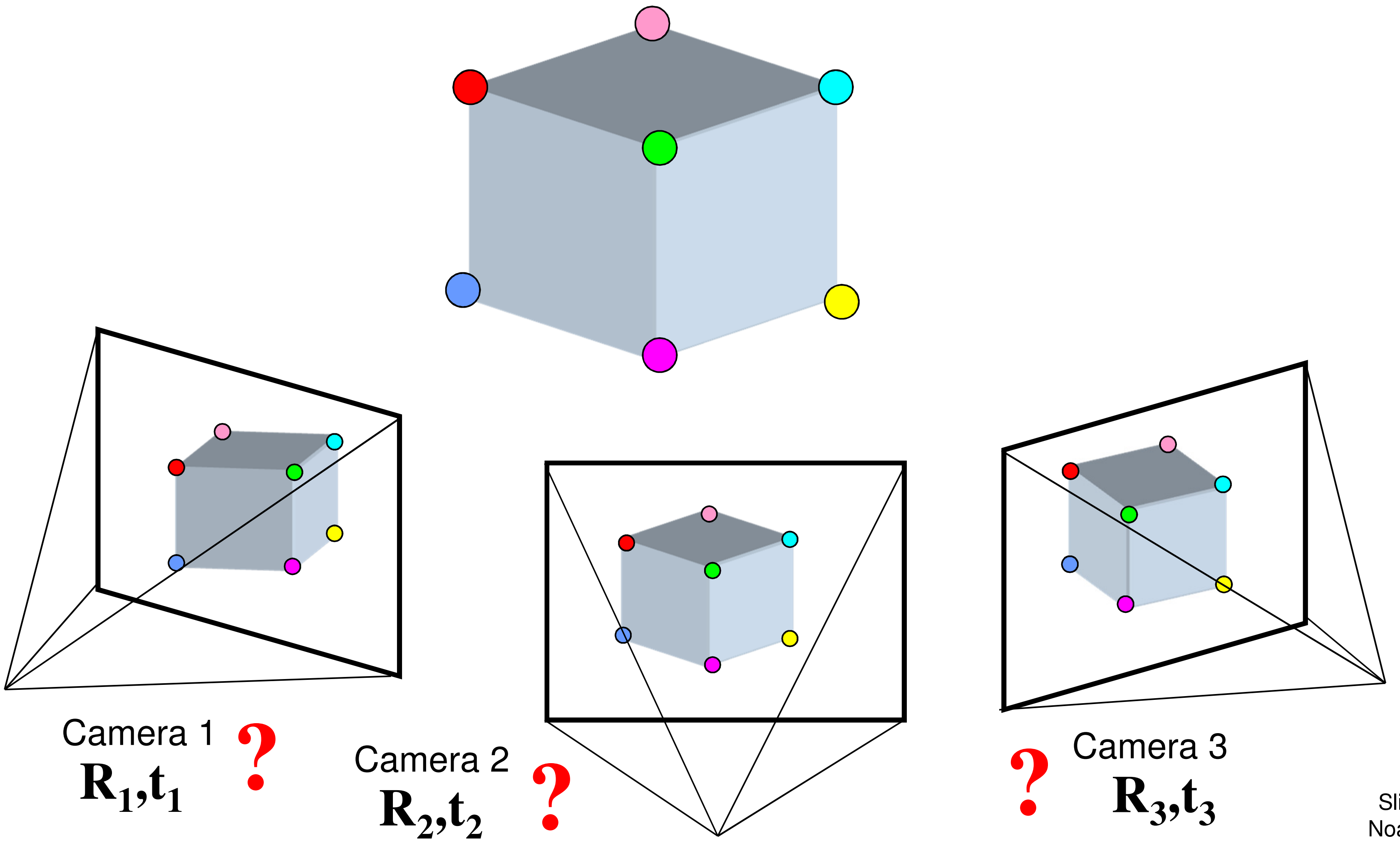
- **Stereo correspondence:** Given a point in one of the images, where could its corresponding points be in the other images?



Slide credit: Noah Snavely

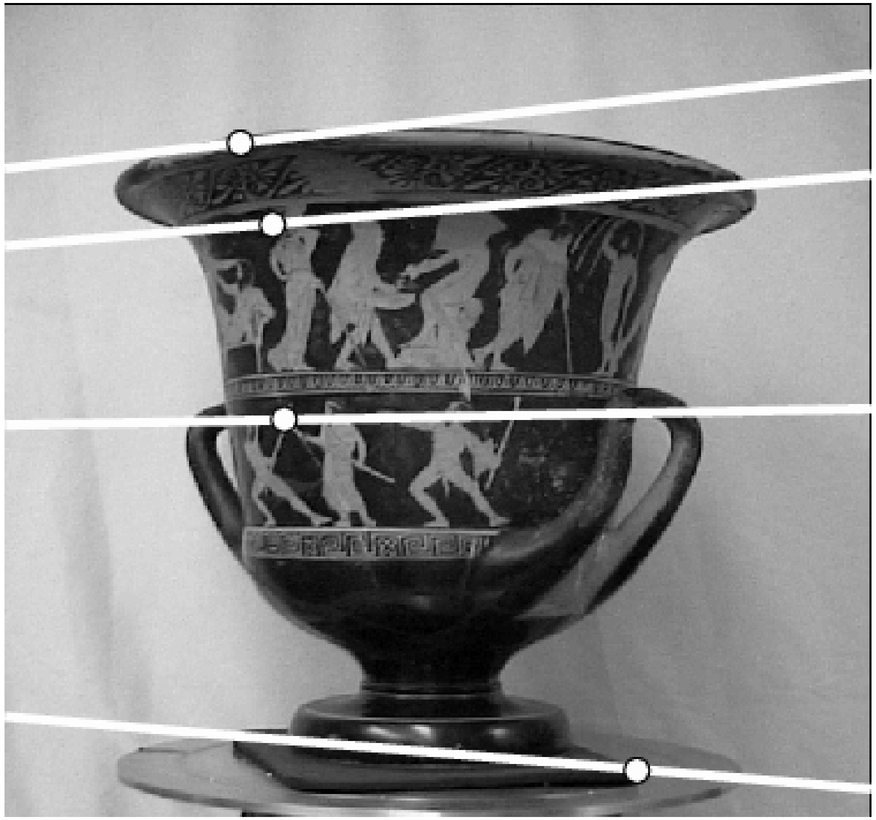
Multi-view geometry problems

- **Motion:** Given a set of corresponding points in two or more images, compute the camera parameters



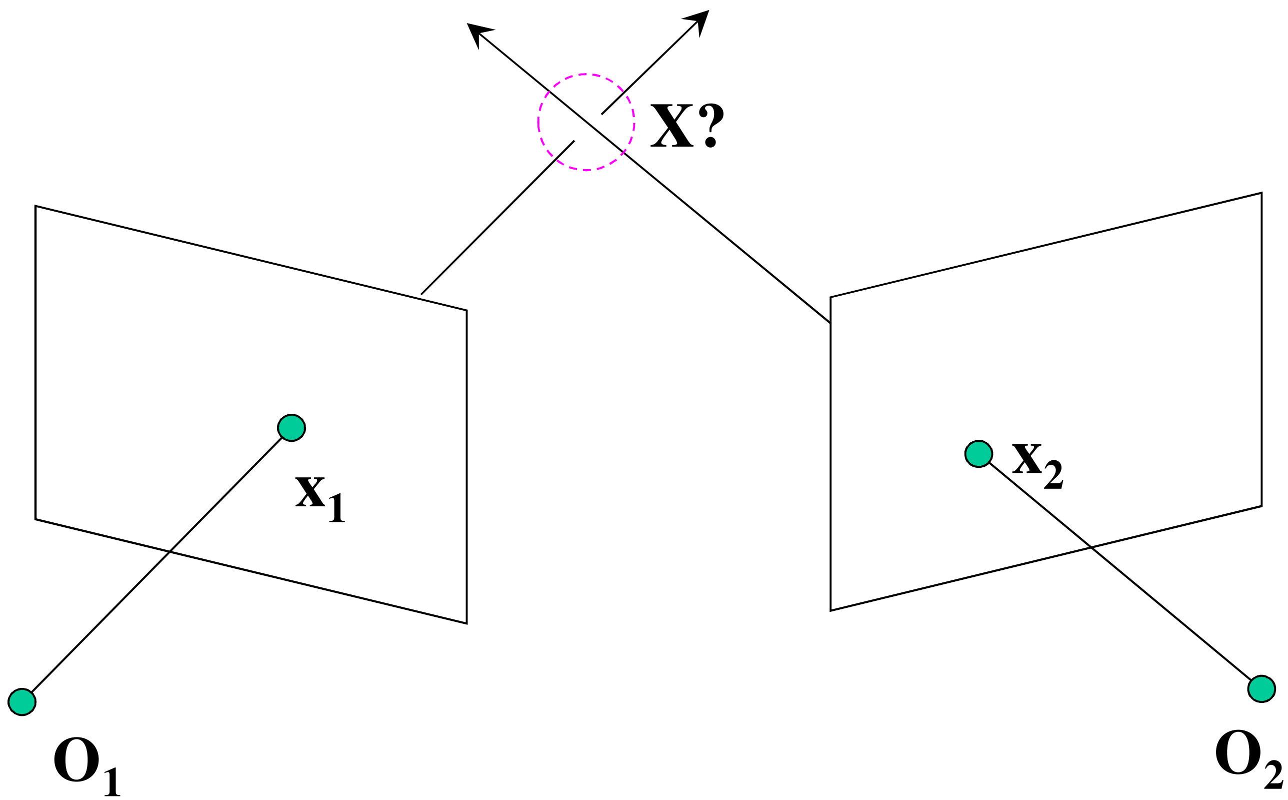
Slide credit: Noah Snavely

Two-view geometry

Triangulation

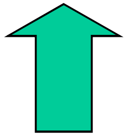
- Given projections of a 3D point in two or more images (with known camera matrices), find the coordinates of the point



Triangulation: Linear approach

Camera Matrices P are known:

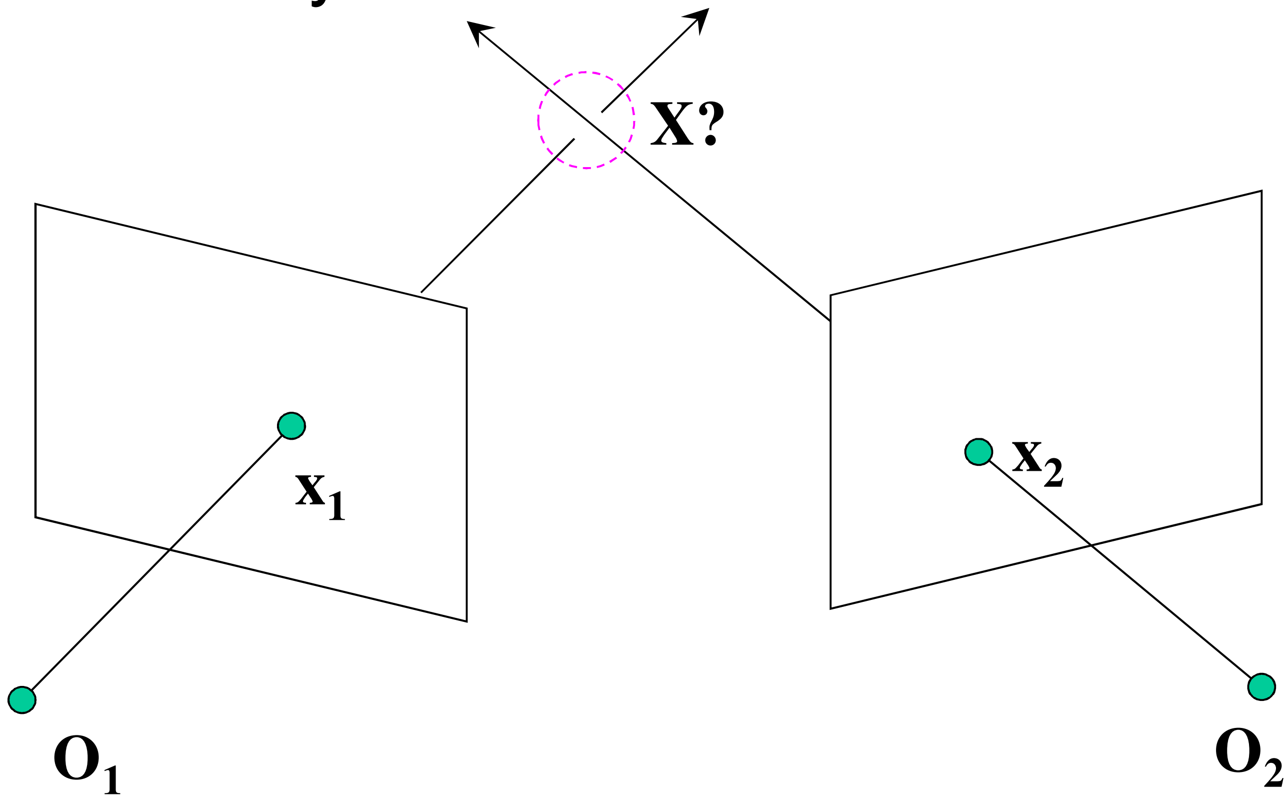
$$\lambda_1 \mathbf{x_1} = \mathbf{P_1 X}$$
$$\lambda_2 \mathbf{x_2} = \mathbf{P_2 X}$$



Two independent equations each in terms of three unknown entries of **X**

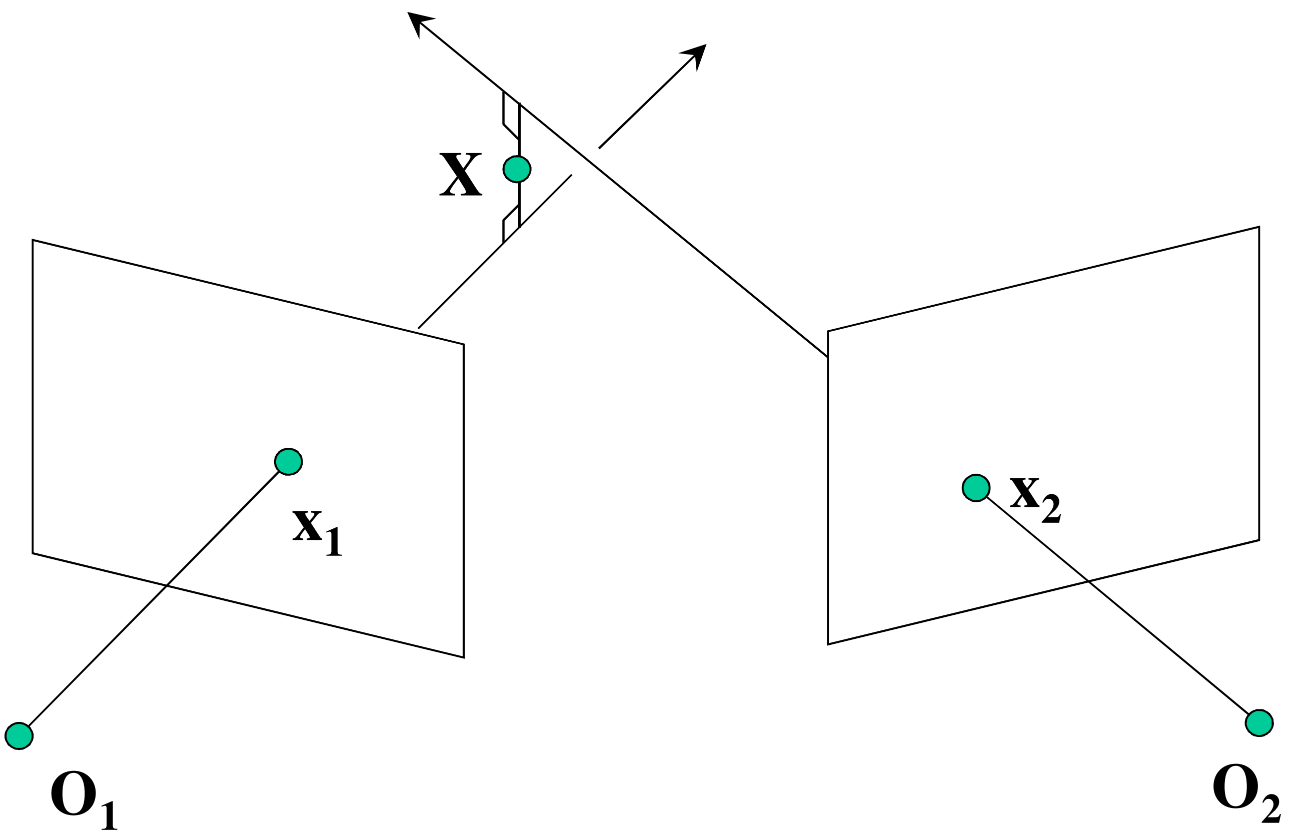
Triangulation

- We want to intersect the two visual rays corresponding to $\mathbf{x_1}$ and $\mathbf{x_2}$, but because of noise and numerical errors, they don't always meet exactly



Triangulation: Geometric approach

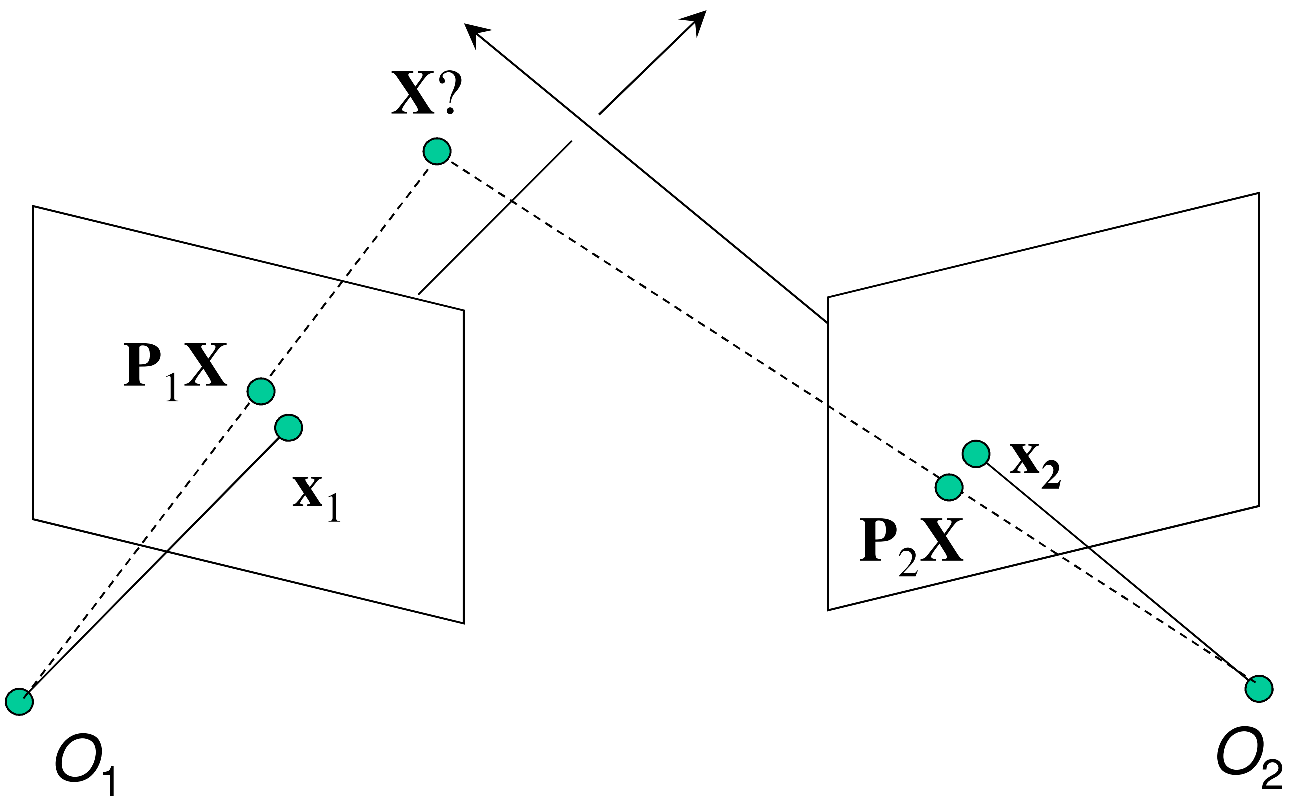
- Find shortest segment connecting the two viewing rays and let **X** be the midpoint of that segment



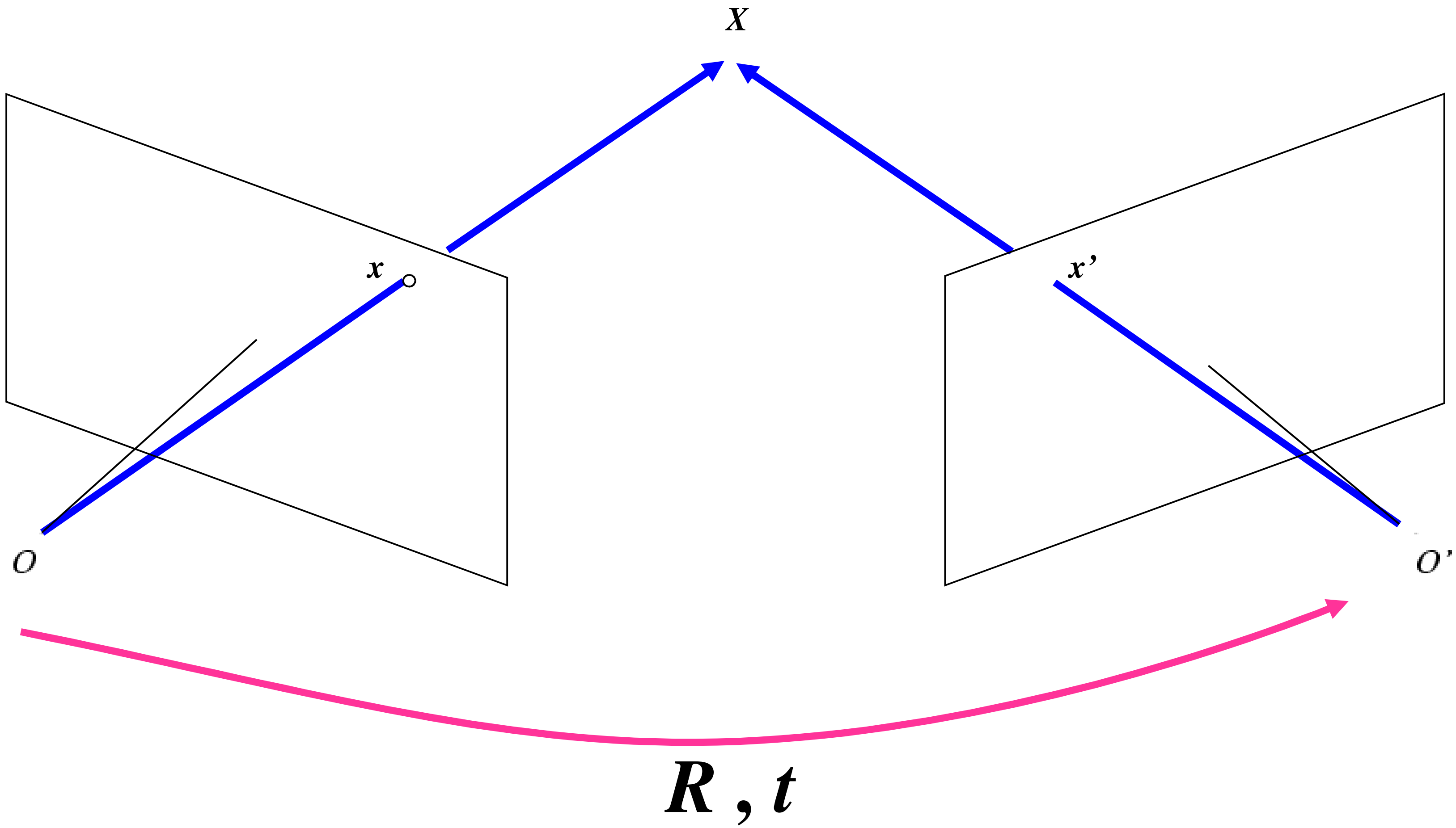
Triangulation: Nonlinear approach

Find X that minimizes

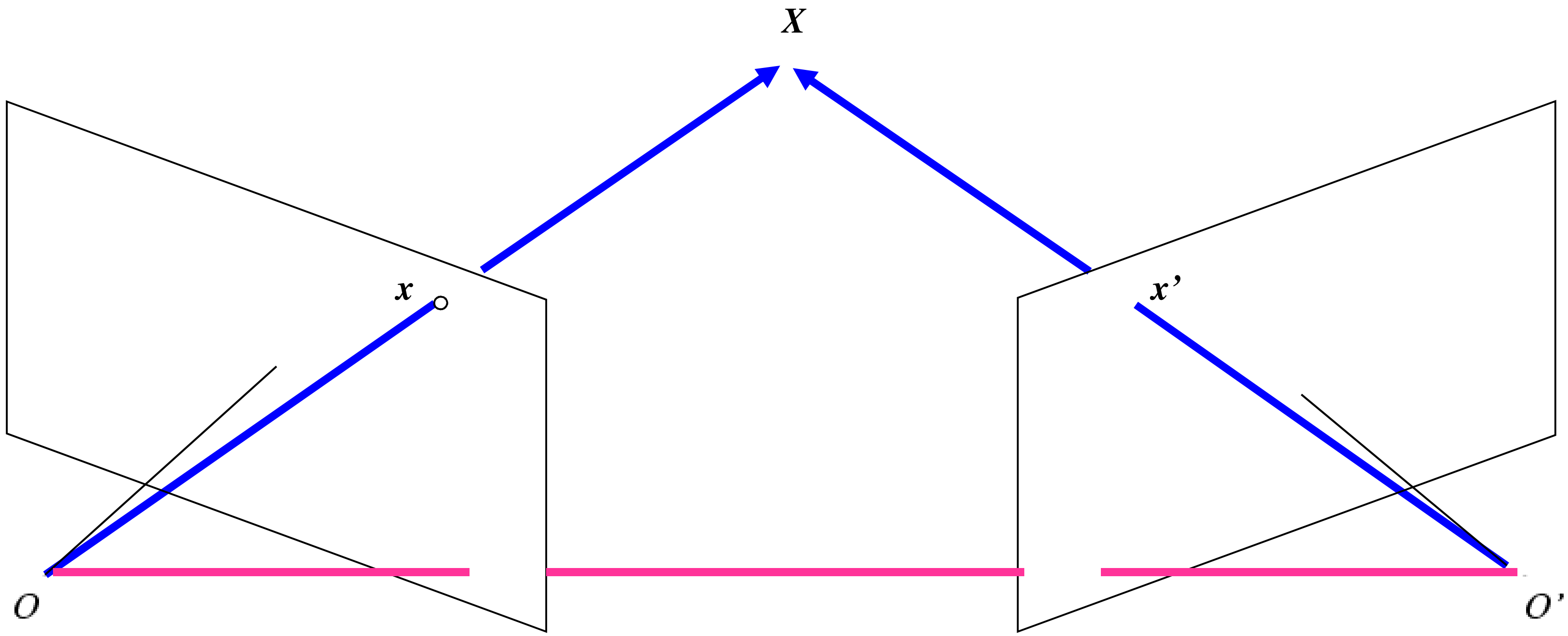
$$d^2(\mathbf{x_1}, \mathbf{P_1X}) + d^2(\mathbf{x_2}, \mathbf{P_2X})$$



Epipolar geometry

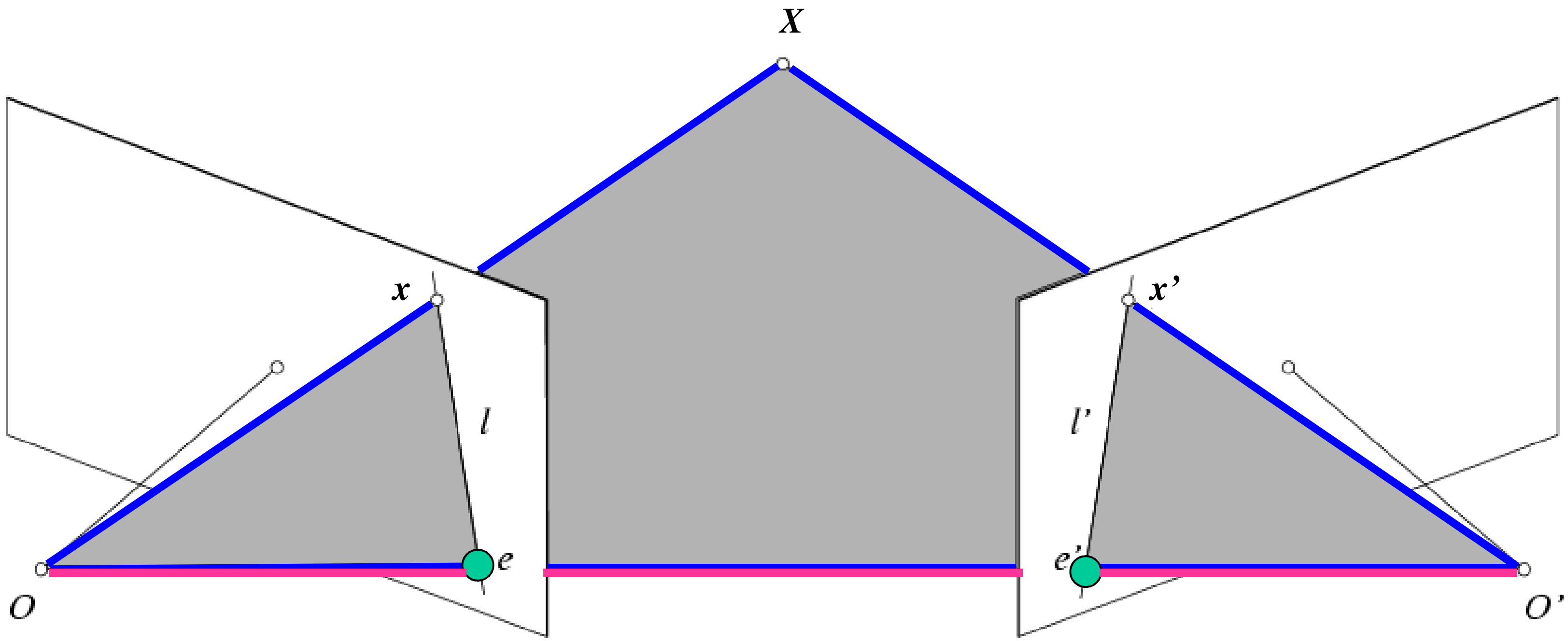


Epipolar geometry



- **Baseline** – line connecting the two camera centers

Epipolar geometry



- **Baseline** – line connecting the two camera centers
- **Epipolar Plane** – plane containing baseline (1D family)
- **Epipoles**

= intersections of baseline with image planes

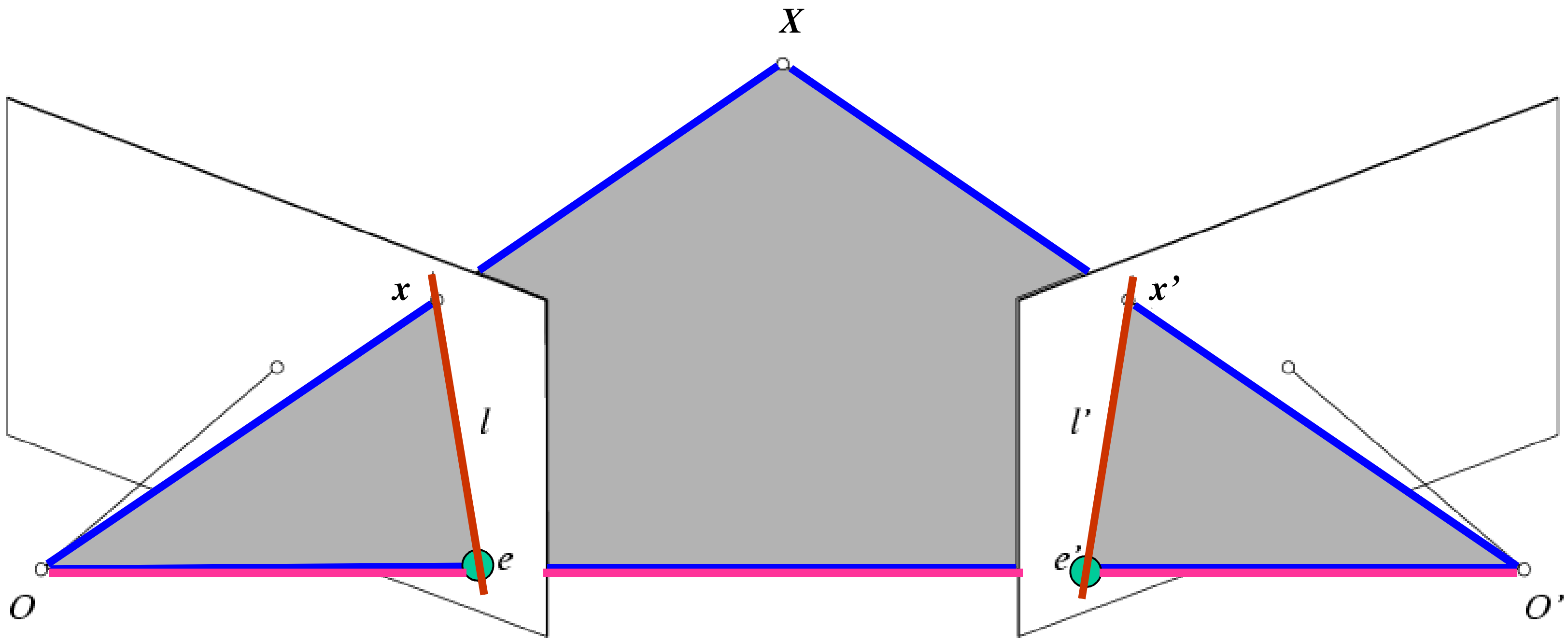
= projections of the other camera center

= vanishing points of the motion direction

The Epipole

Photo by Frank Dellaert

Epipolar geometry



- **Baseline** – line connecting the two camera centers
- **Epipolar Plane** – plane containing baseline (1D family)
- **Epipoles**

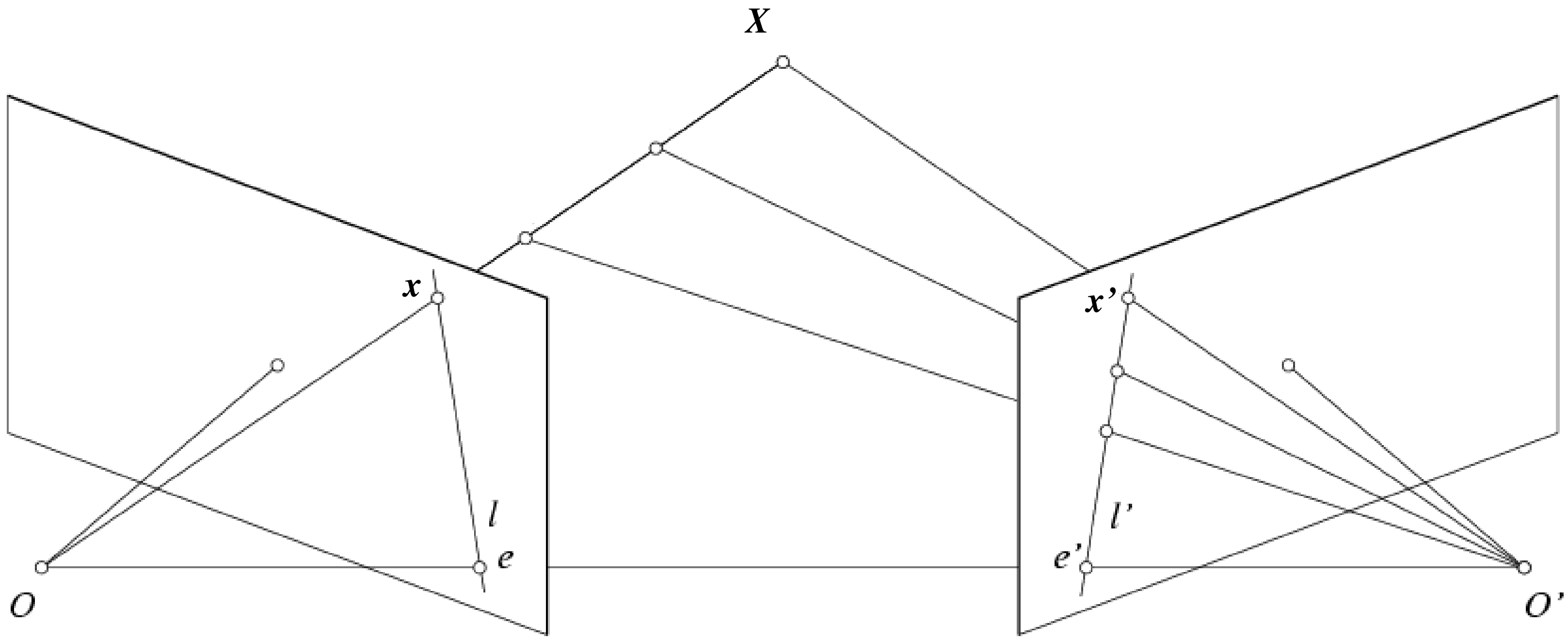
= intersections of baseline with image planes

= projections of the other camera center

= vanishing points of the motion direction

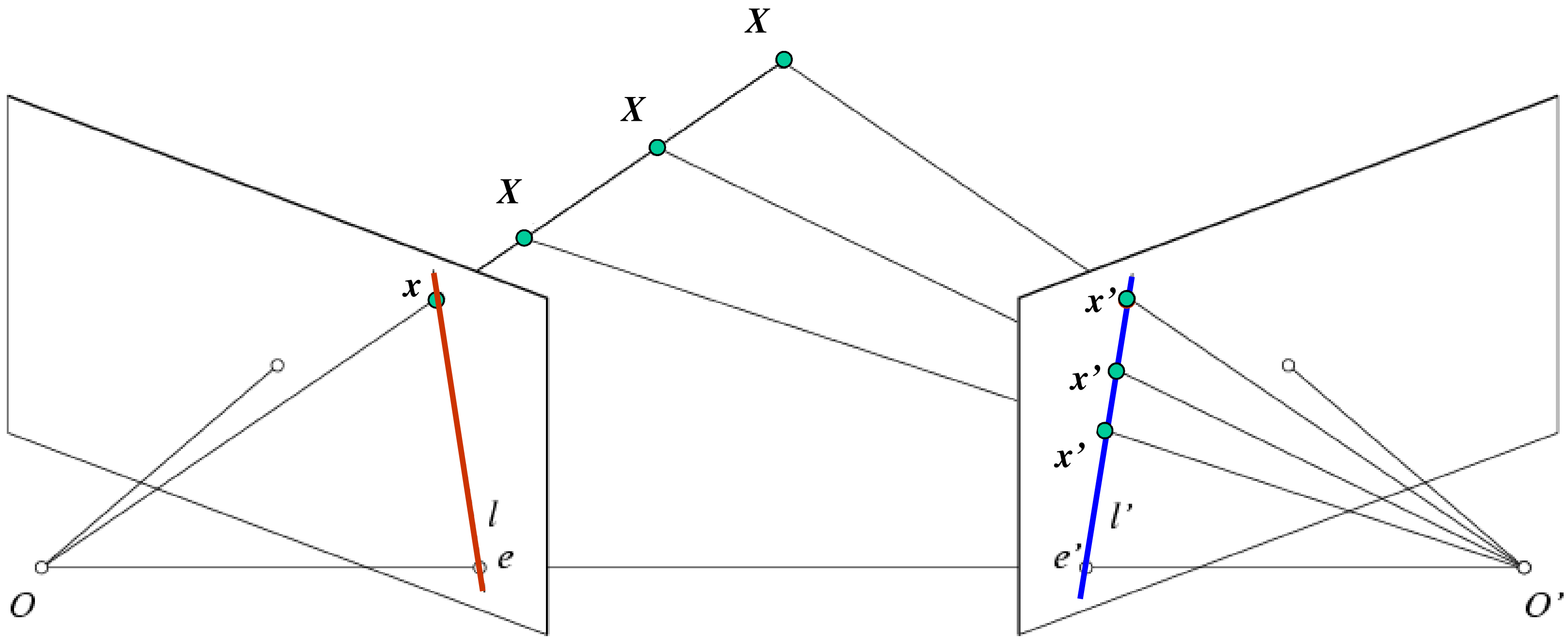
- **Epipolar Lines** - intersections of epipolar plane with image planes (always come in corresponding pairs)

Epipolar constraint



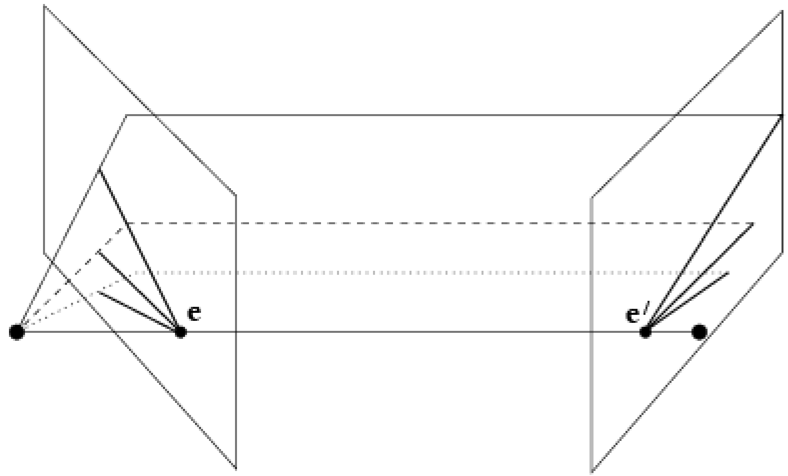
- If we observe a point ***x*** in one image, where can the corresponding point ***x'*** be in the other image?

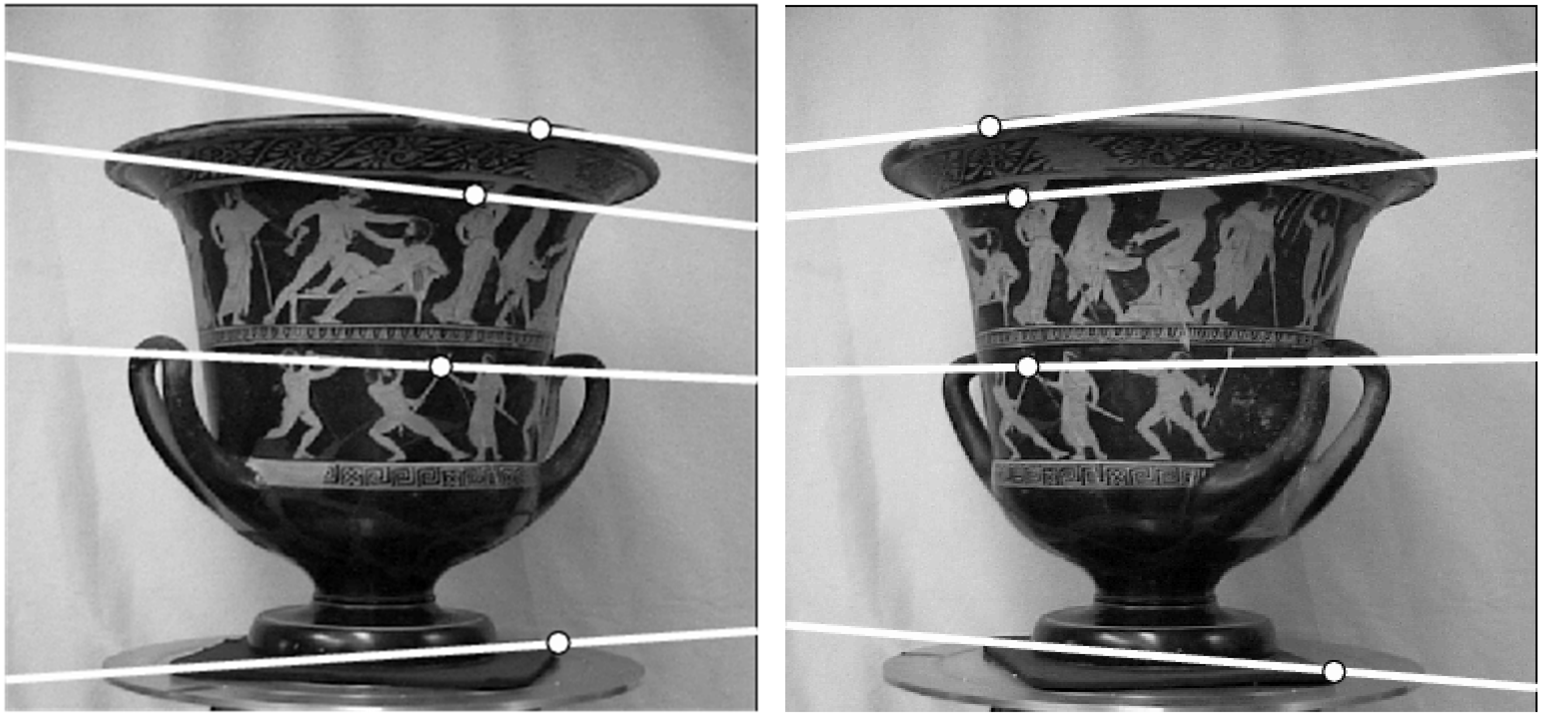
Epipolar constraint



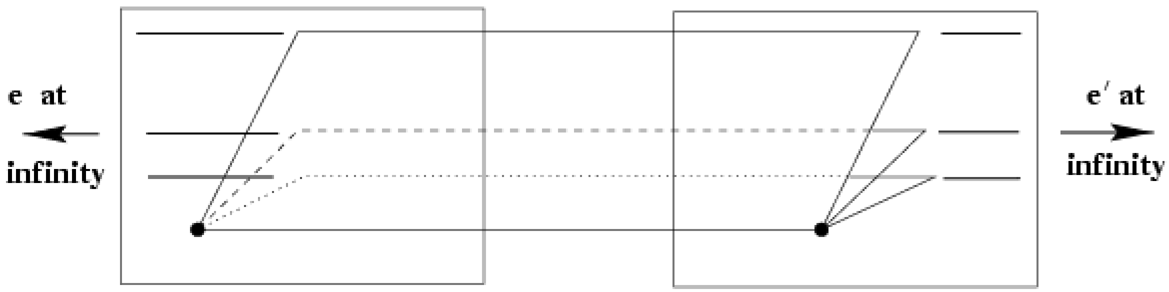
- Potential matches for ***x*** have to lie on the corresponding epipolar line ***l'***.
- Potential matches for ***x'*** have to lie on the corresponding epipolar line ***l***.

Example: Converging cameras





Example: Motion parallel to image plane



Example: Motion perpendicular to image plane

Example: Motion perpendicular to image plane

- Points move along lines radiating from the epipole: “focus of expansion”
- Epipole is the principal point

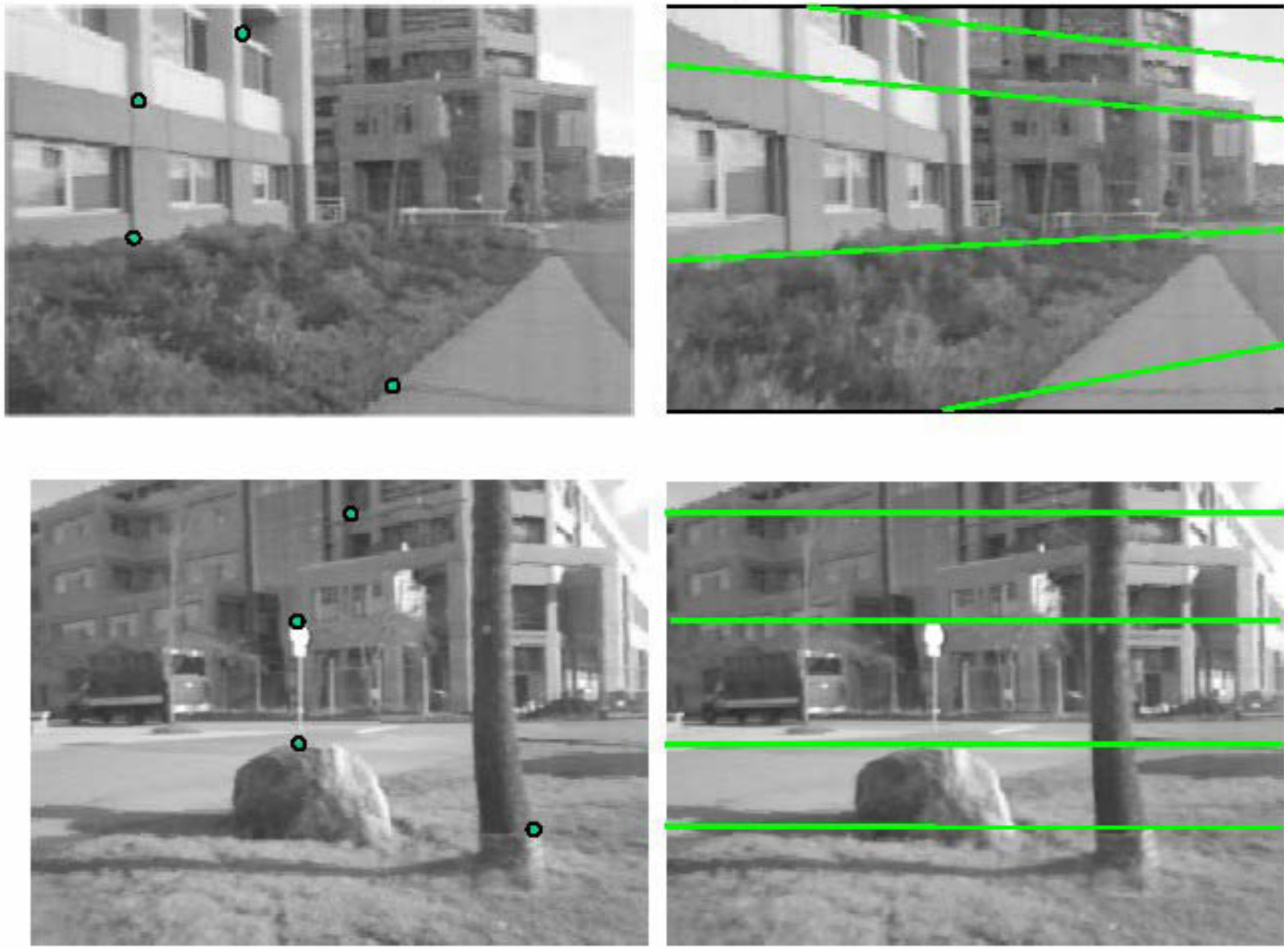
Example: Motion perpendicular to image plane

http://vimeo.com/48425421

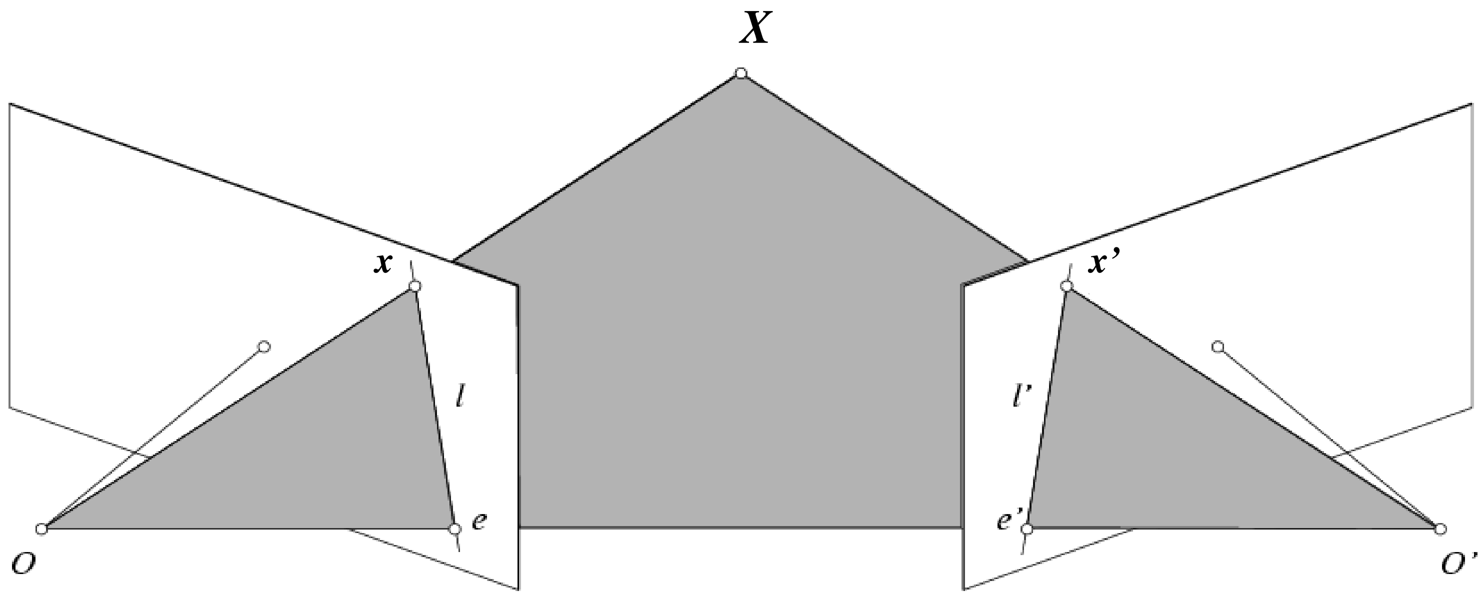
Stanley Kubrick – Film creator1928-1999

(Clockwork Orange, 2001: A Space Odyssey, The Shining)

Another Epipolar constraint example



Epipolar constraint: Calibrated case



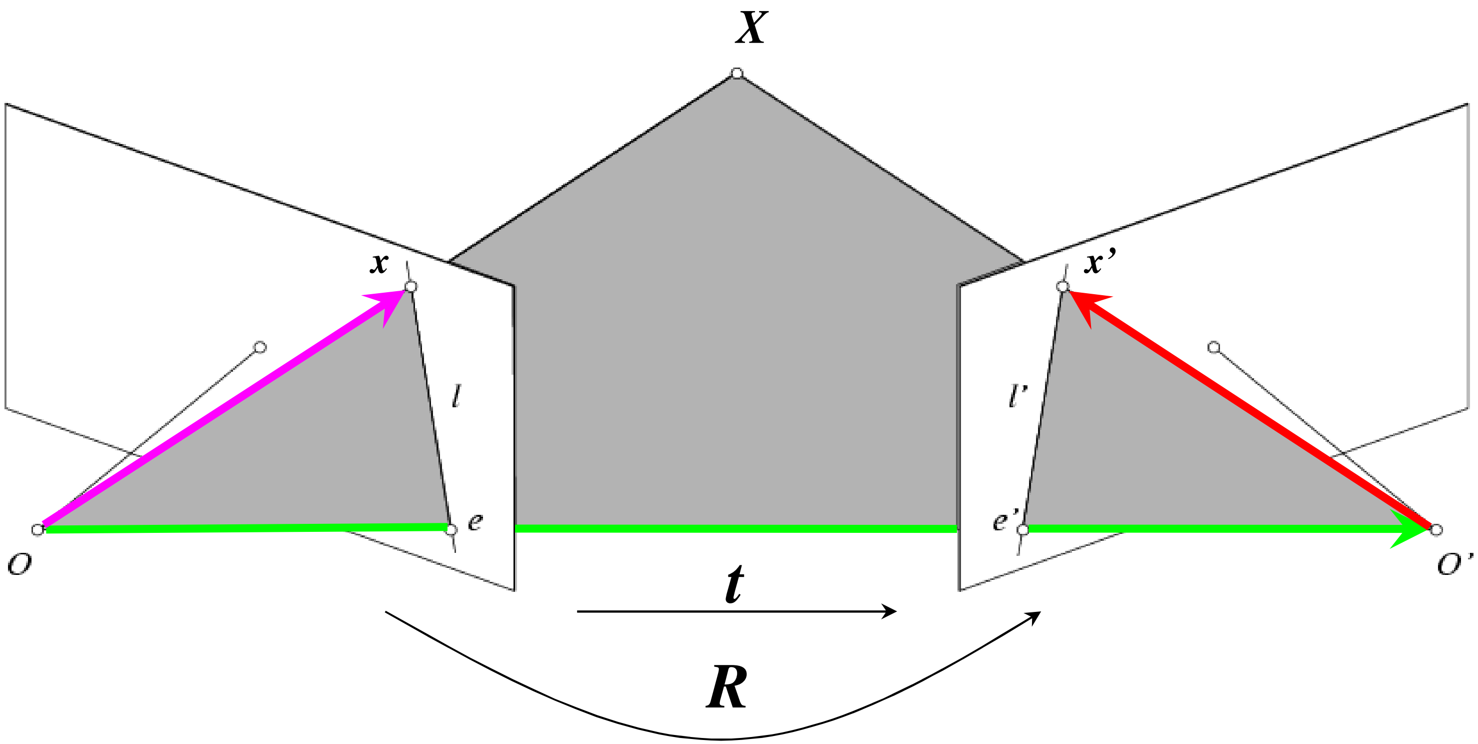
- Intrinsic and extrinsic parameters of the cameras are known, world coordinate system is set to that of the first camera
- Then the projection matrices are given by $K[I \mid 0]$ and $K'[R \mid t]$

$$x = K[I \mid 0]X \qquad x' = K'[R \mid t]X$$

- We can multiply image points by the inverse of the calibration matrices to get *normalized* image coordinates:

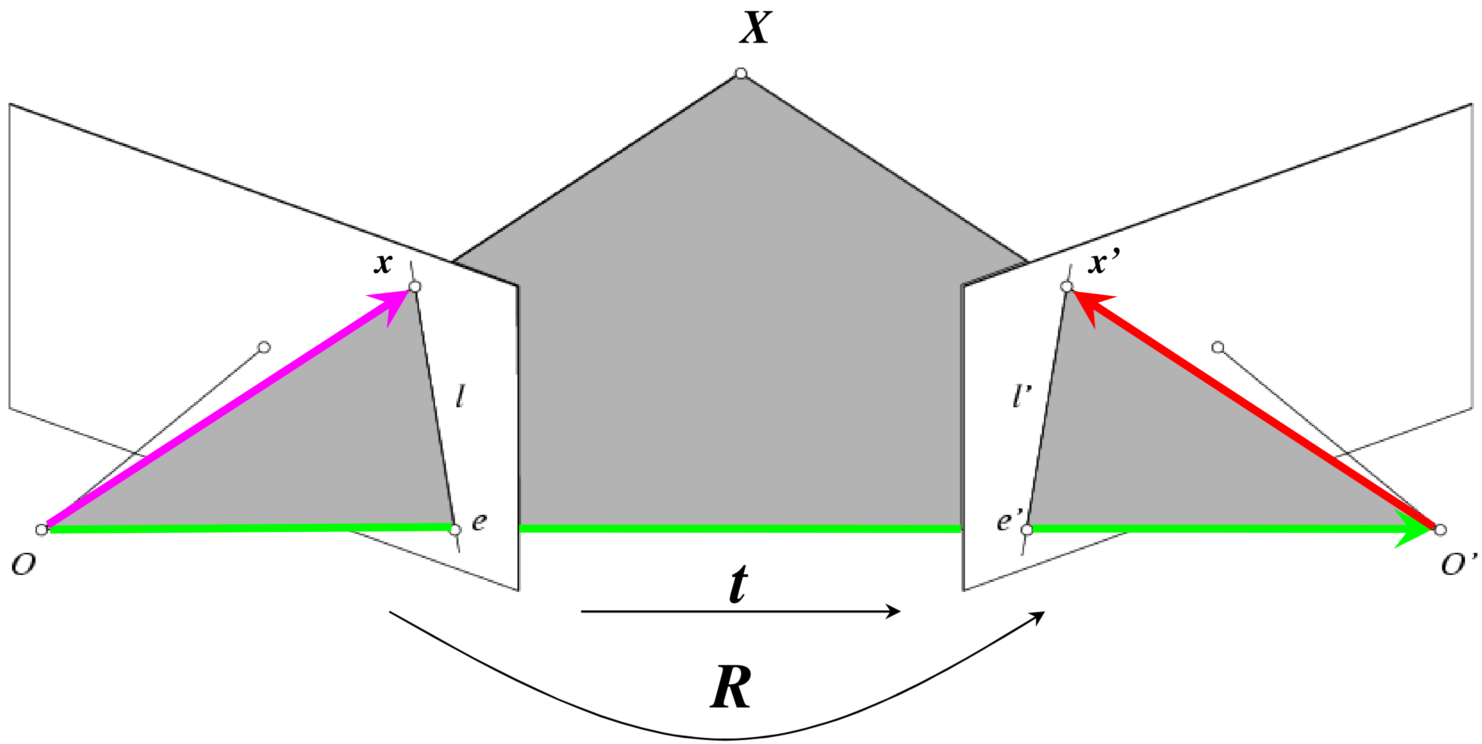
$$x_{\text{norm}} = K^{-1} x_{\text{pixel}} = [I \;\; 0]\, X, \qquad x'_{\text{norm}} = K'^{-1} x'_{\text{pixel}} = [R \;\; t]\, X$$

Epipolar constraint: Calibrated case



The vectors → → → are coplanar

Epipolar constraint: Calibrated case

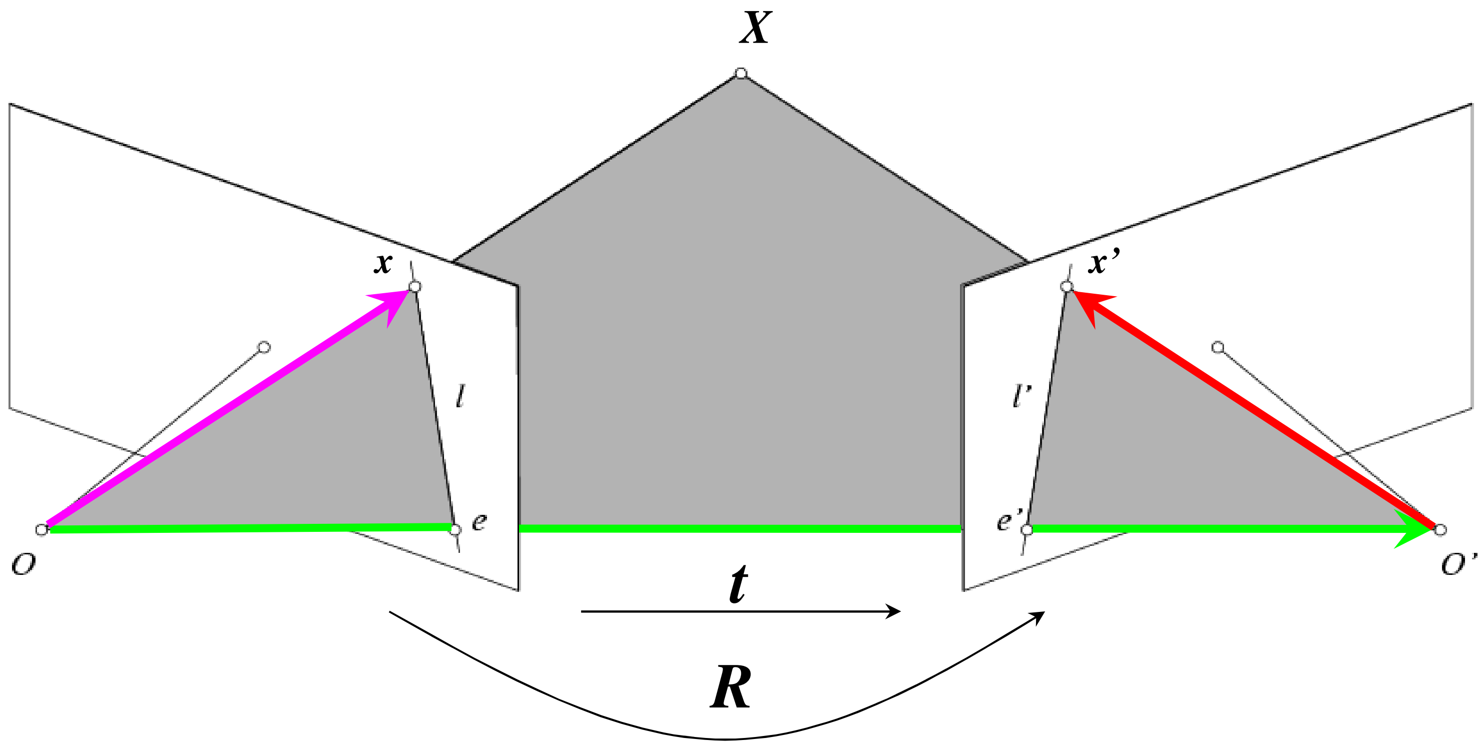


Transfer all vectors to 1st camera coordinate system.

Transfer direction x' to 1st camera coordinate system.

$$\mathbf{x}' \longrightarrow \mathbf{R}^T\mathbf{x}'$$

Epipolar constraint: Calibrated case

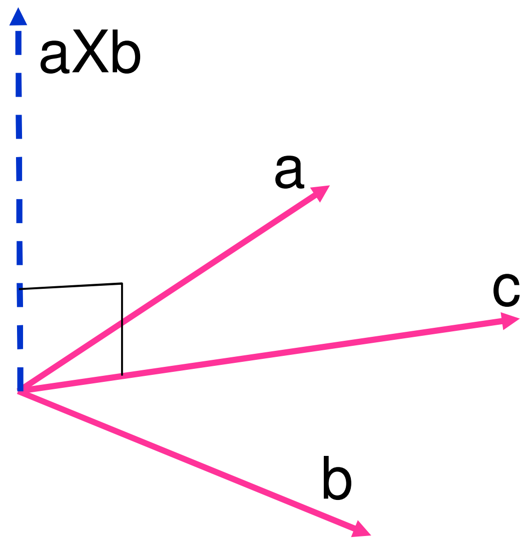


Now the vectors $\boldsymbol{x}$, $\boldsymbol{t}$, and $\boldsymbol{R}^T\boldsymbol{x}'$ are coplanar

$$(\boldsymbol{R}^T \boldsymbol{x}') \cdot [\boldsymbol{t} \times \boldsymbol{x}] = 0$$

Matrix form of cross product

a,b,c are coplanar



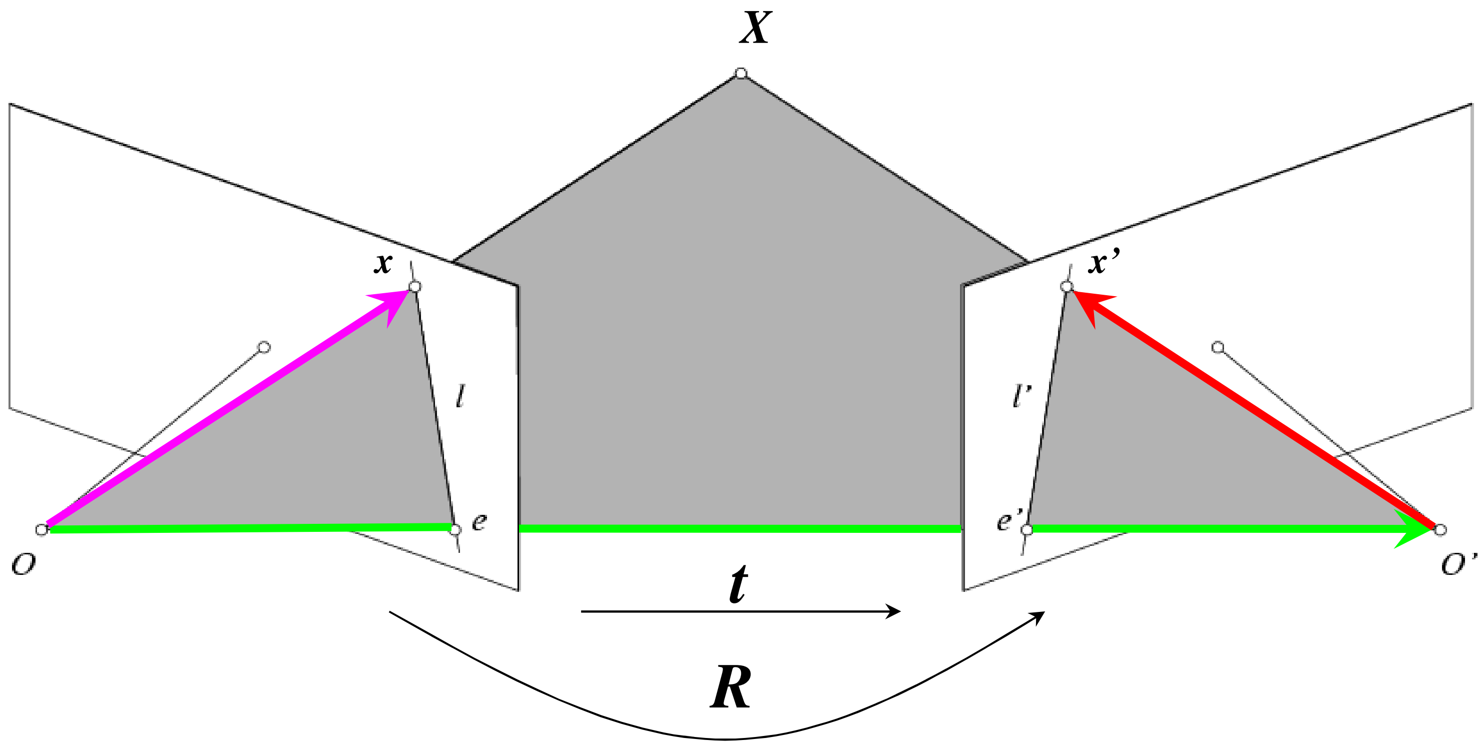
$\Rightarrow$ c · (aXb) = 0

In Matrix form:

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

$c^T [a_x] b = 0$

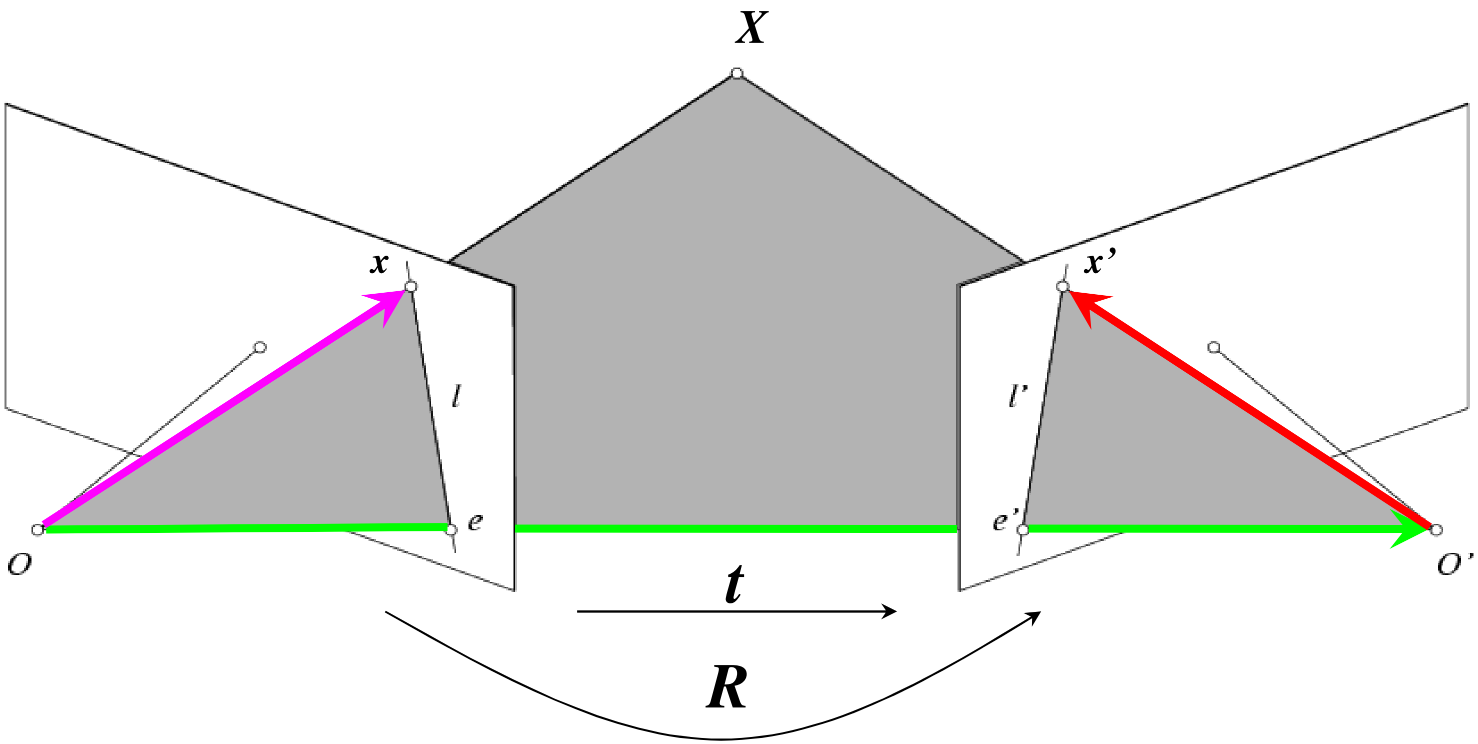
Epipolar constraint: Calibrated case



Now the vectors $\boldsymbol{x}$, $\boldsymbol{t}$, and $\boldsymbol{R}^T\boldsymbol{x}'$ are coplanar

$$(\boldsymbol{R}^T\boldsymbol{x}') \cdot [\boldsymbol{t} \times \boldsymbol{x}] = 0$$

Epipolar constraint: Calibrated case

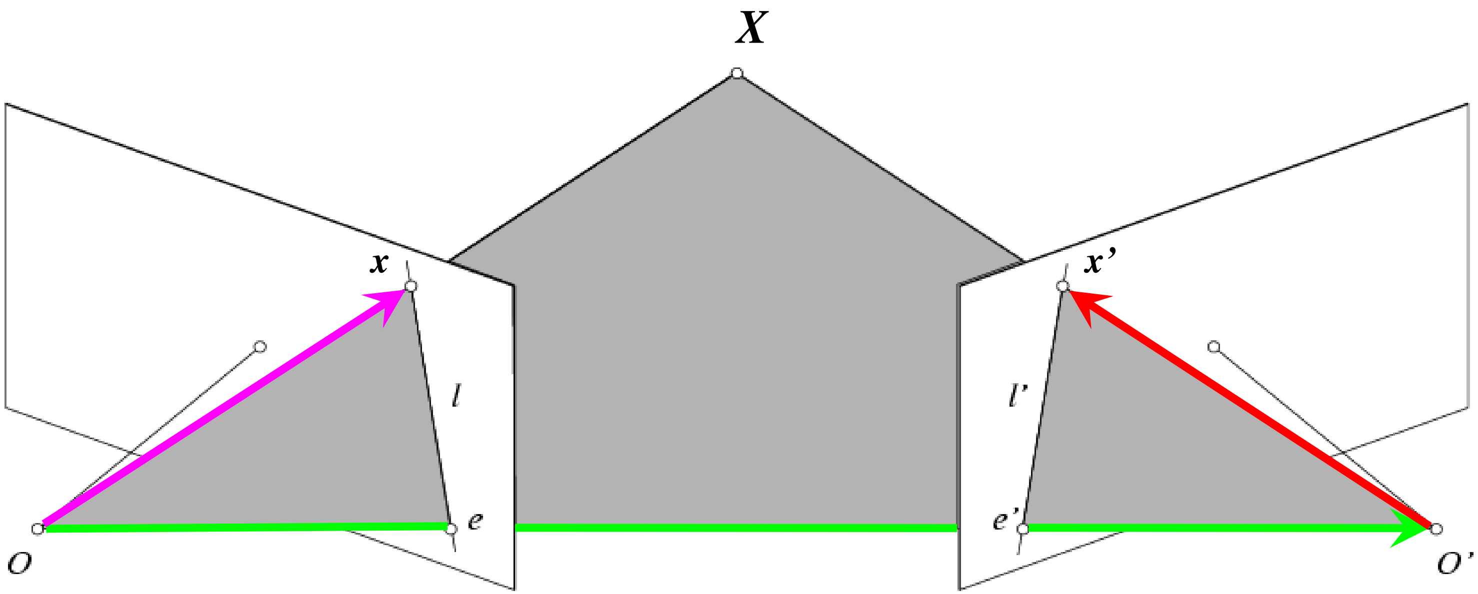


Now the vectors $\boldsymbol{x}$, $\boldsymbol{t}$, and $\boldsymbol{R^T x'}$ are coplanar

$$(\boldsymbol{R}^T \boldsymbol{x}')^{\mathrm{T}} [\boldsymbol{t}_x]\, \boldsymbol{x} = 0$$

$$\boldsymbol{x}'^{\mathrm{T}} \boldsymbol{R}\, [\boldsymbol{t}_x]\, \boldsymbol{x} = 0$$

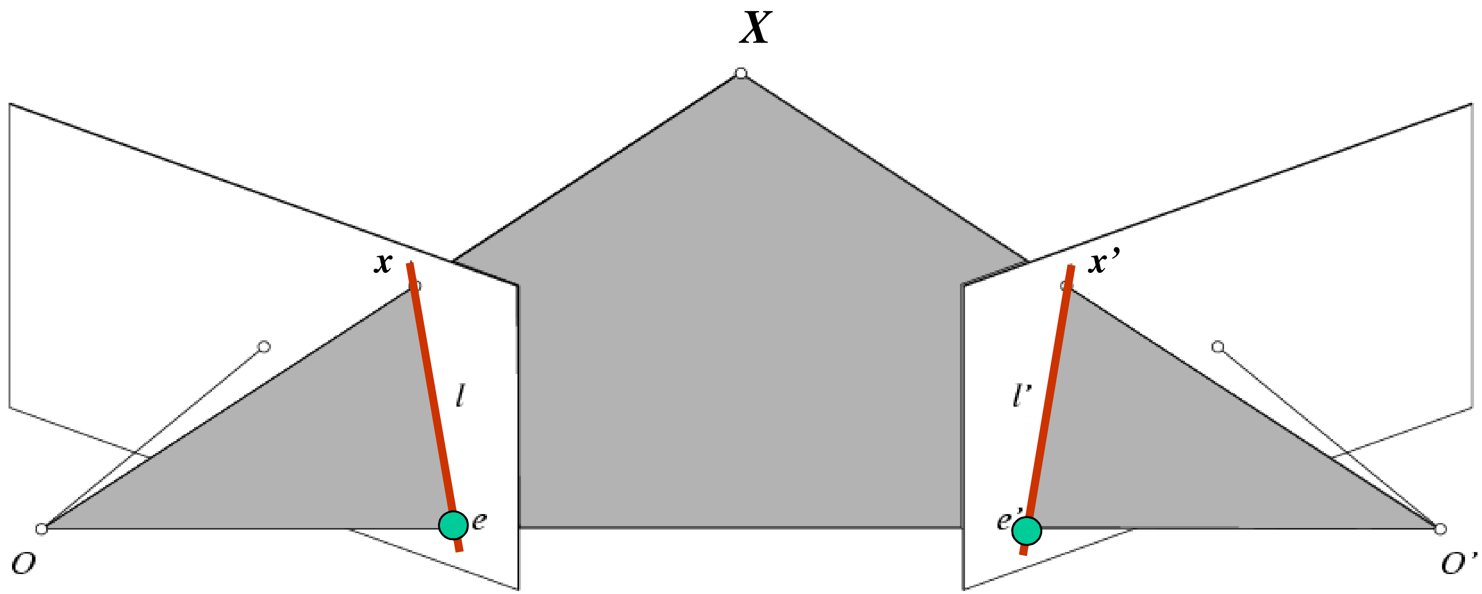
Epipolar constraint: Calibrated case



$$x'^{T} R\,[t_x]\,x = 0 \;\Rightarrow\; x'^{T} E x = 0 \quad \text{with} \quad E = R[t_\times]$$

Essential Matrix
(Longuet-Higgins, 1981)

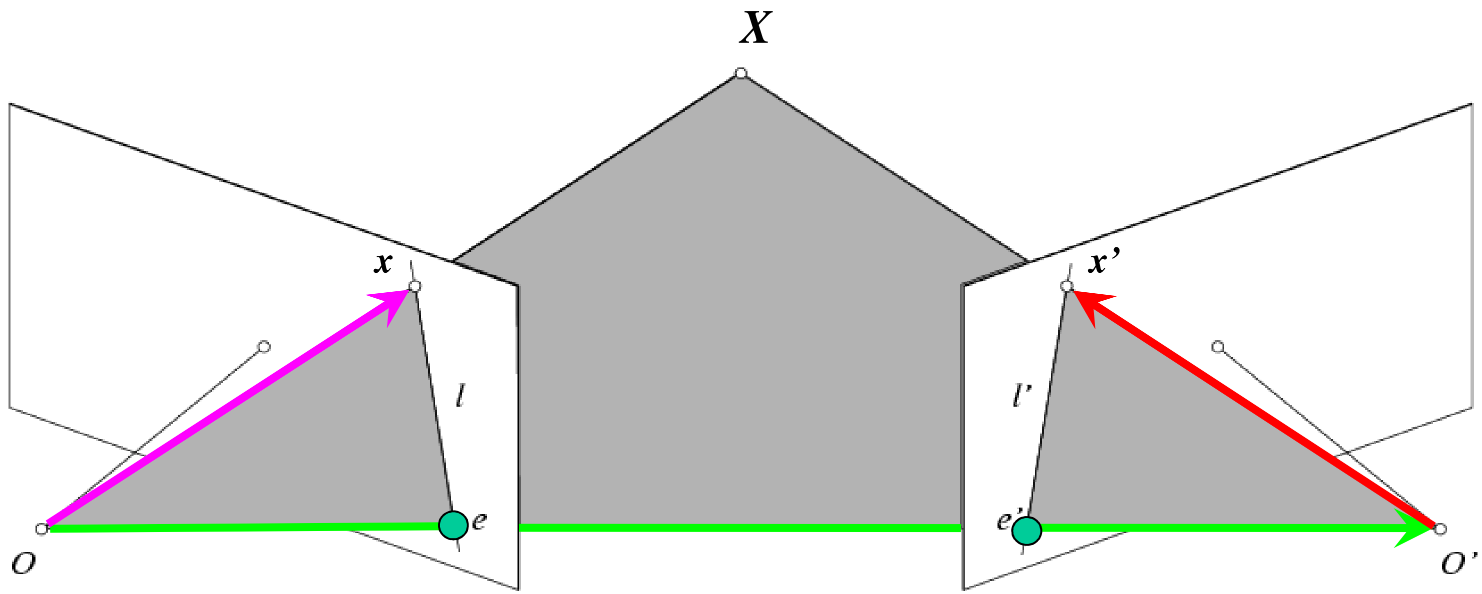
Epipolar constraint: Calibrated case



$$\boldsymbol{x}'^{\mathrm{T}} \boldsymbol{R}\,[\boldsymbol{t}_x]\,\boldsymbol{x} = 0 \quad \Rightarrow \quad \boldsymbol{x}'^{T} \boldsymbol{E}\,\boldsymbol{x} = 0 \quad \textbf{\textit{with}} \quad \boldsymbol{E} = \boldsymbol{R}[\boldsymbol{t}_\times]$$

- $\boldsymbol{E}\,\boldsymbol{x}$ is the epipolar line associated with $\boldsymbol{x}$ ($\boldsymbol{l}' = \boldsymbol{E}\,\boldsymbol{x}$)
- $\boldsymbol{E}^T\boldsymbol{x}'$ is the epipolar line associated with $\boldsymbol{x}'$ ($\boldsymbol{l} = \boldsymbol{E}^T\boldsymbol{x}'$)
- $\boldsymbol{E}\,\boldsymbol{e} = 0$ and $\boldsymbol{E}^T\boldsymbol{e}' = 0$
- $\boldsymbol{E}$ is singular (rank two)
- $\boldsymbol{E}$ has five degrees of freedom

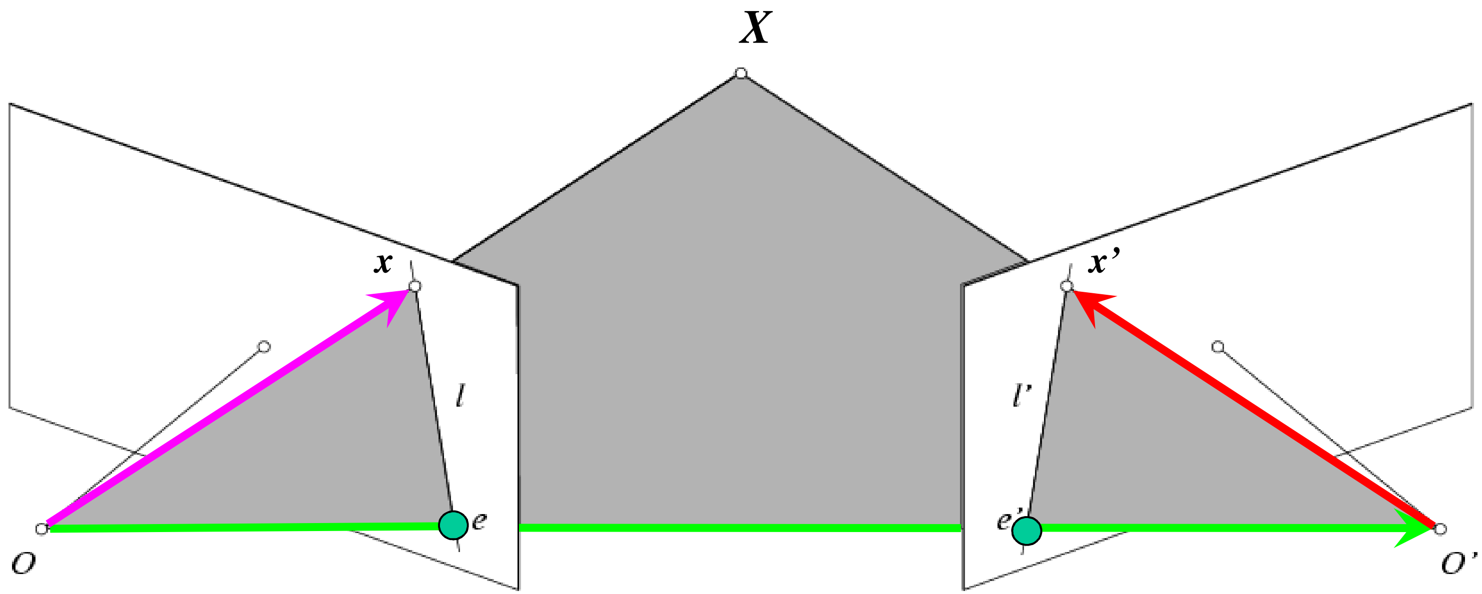
Epipolar constraint: Uncalibrated case



- The calibration matrices ***K*** and ***K'*** of the two cameras are unknown
- We can write the epipolar constraint in terms of *unknown* normalized coordinates:

$$\hat{\boldsymbol{x}}'^T \boldsymbol{E} \hat{\boldsymbol{x}} = 0 \qquad \hat{\boldsymbol{x}} = \boldsymbol{K}^{-1} \boldsymbol{x}, \quad \hat{\boldsymbol{x}}' = \boldsymbol{K}'^{-1} \boldsymbol{x}'$$

Epipolar constraint: Uncalibrated case



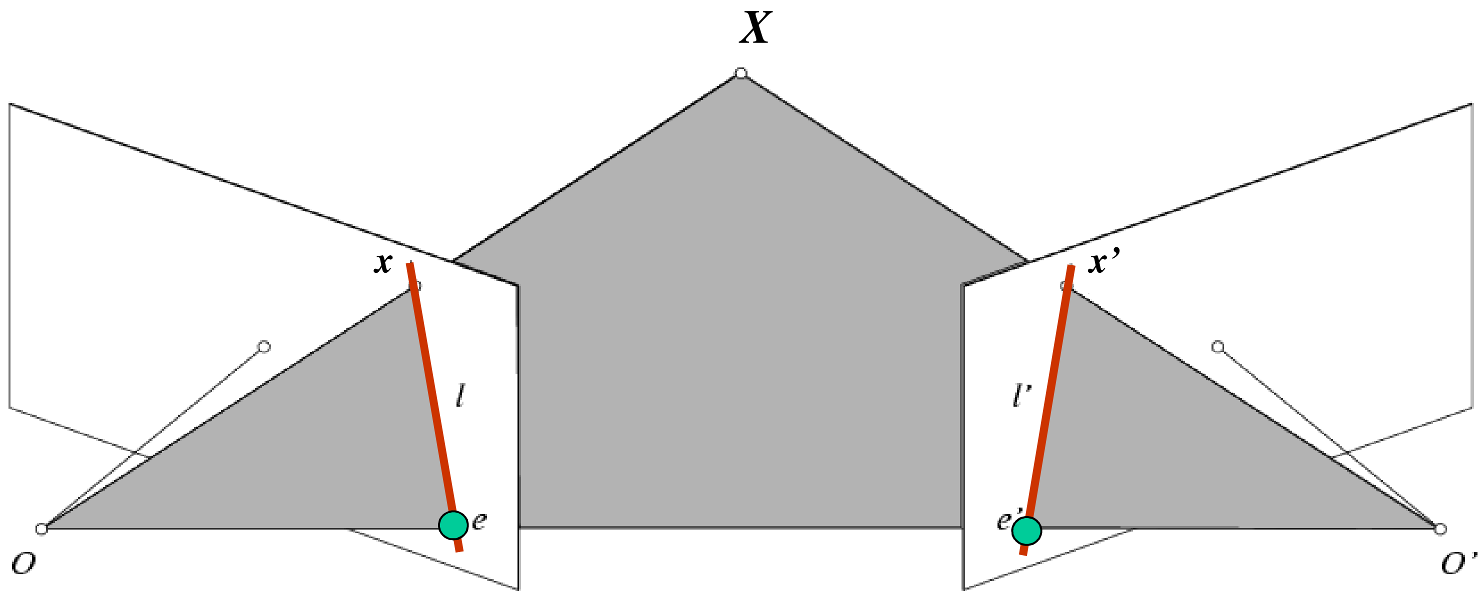
$$\hat{\boldsymbol{x}}'^T \boldsymbol{E}\, \hat{\boldsymbol{x}} = 0 \quad \Rightarrow \quad \boldsymbol{x}'^T \boldsymbol{F}\, \boldsymbol{x} = 0 \quad \text{with} \quad \boldsymbol{F} = \boldsymbol{K}'^{-T} \boldsymbol{E}\, \boldsymbol{K}^{-1}$$

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

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

Fundamental Matrix
(Faugeras and Luong, 1992)

Epipolar constraint: Uncalibrated case



$$\hat{\boldsymbol{x}}'^T \boldsymbol{E} \hat{\boldsymbol{x}} = 0 \quad \Rightarrow \quad \boldsymbol{x}'^T \boldsymbol{F} \boldsymbol{x} = 0 \quad \text{with} \quad \boldsymbol{F} = \boldsymbol{K}'^{-T} \boldsymbol{E} \boldsymbol{K}^{-1}$$

- $\boldsymbol{F}\,\boldsymbol{x}$ is the epipolar line associated with $\boldsymbol{x}$ ($\boldsymbol{l}' = \boldsymbol{F}\,\boldsymbol{x}$)
- $\boldsymbol{F}^T\boldsymbol{x}'$ is the epipolar line associated with $\boldsymbol{x}'$ ($\boldsymbol{l}' = \boldsymbol{F}^T\boldsymbol{x}'$)
- $\boldsymbol{F}\,\boldsymbol{e} = 0$ and $\boldsymbol{F}^T\boldsymbol{e}' = 0$
- $\boldsymbol{F}$ is singular (rank two)
- $\boldsymbol{F}$ has *seven* degrees of freedom

The eight-point algorithm

$$\boldsymbol{x} = (u, v, 1)^T, \quad \boldsymbol{x}' = (u', v', 1)$$

$$\begin{bmatrix} u' & v' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$$

$$\Rightarrow \begin{bmatrix} u'u & u'v & u' & v'u & v'v & v' & u & v & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

Minimize:

$$\sum_{i=1}^{N} (\boldsymbol{x}_i'^T \boldsymbol{F} \boldsymbol{x}_i)^2$$

under the constraint

$||\boldsymbol{F}||^2 = 1$

The eight-point algorithm

- Meaning of error $\sum_{i=1}^{N} (\boldsymbol{x}_i'^T \boldsymbol{F} \boldsymbol{x}_i)^2$:

  sum of squared *algebraic* distances between points $\boldsymbol{x}'_i$ and epipolar lines $\boldsymbol{Fx}_i$ (or points $\boldsymbol{x}_i$ and epipolar lines $\boldsymbol{F}^T\boldsymbol{x}'_i$)
- Nonlinear approach: minimize sum of squared *geometric* distances

$$\sum_{i=1}^{N} \left[\mathrm{d}^2(\boldsymbol{x}_i', \boldsymbol{F}\,\boldsymbol{x}_i) + \mathrm{d}^2(\boldsymbol{x}_i, \boldsymbol{F}^T \boldsymbol{x}_i') \right]$$

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

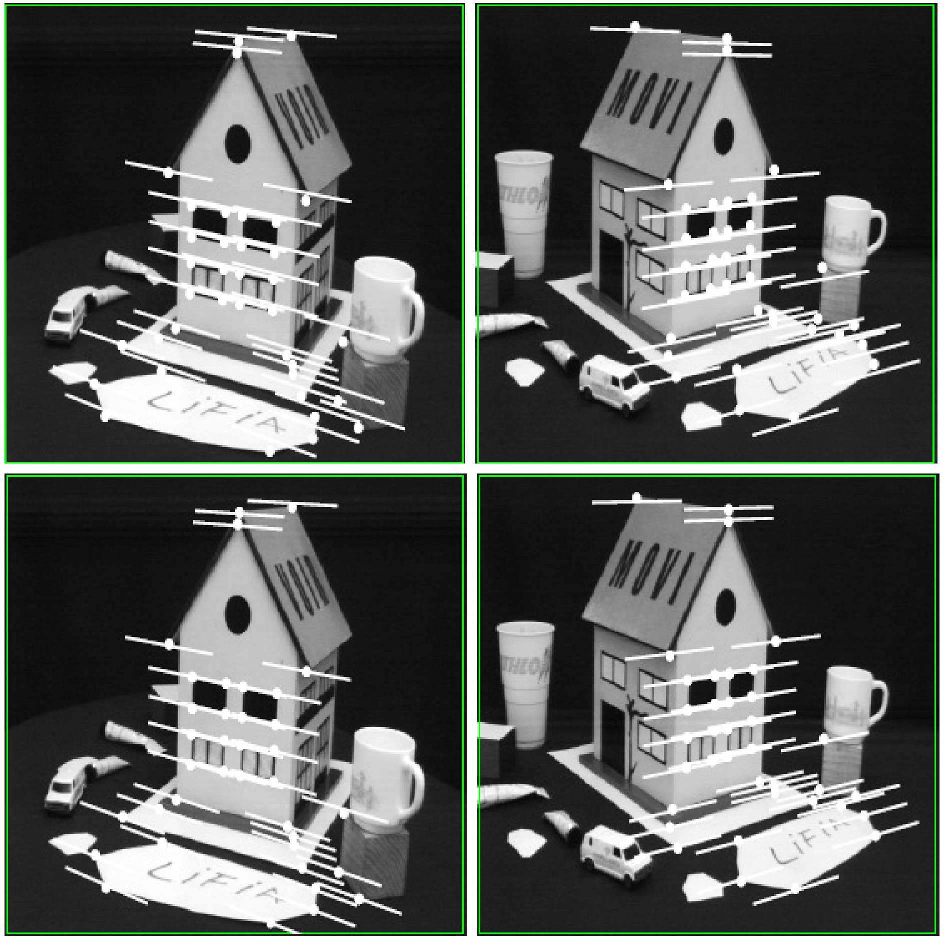
Can be fixed by rescaling the data

The normalized eight-point algorithm

(Hartley, 1995)

- 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, than the fundamental matrix in original coordinates is $\boldsymbol{T'}^T \boldsymbol{F} \boldsymbol{T}$

Comparison of estimation algorithms



	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

From epipolar geometry to camera calibration

- Estimating the fundamental matrix is known as “weak calibration”
- If we know the calibration matrices of the two cameras, we can estimate the essential matrix: $\boldsymbol{E} = \boldsymbol{K}'^{T}\boldsymbol{F}\boldsymbol{K}$
- The essential matrix gives us the relative rotation and translation between the cameras, or their extrinsic parameters