Multi-view geometry

Slides from L. Lazebnik

Structure from motion

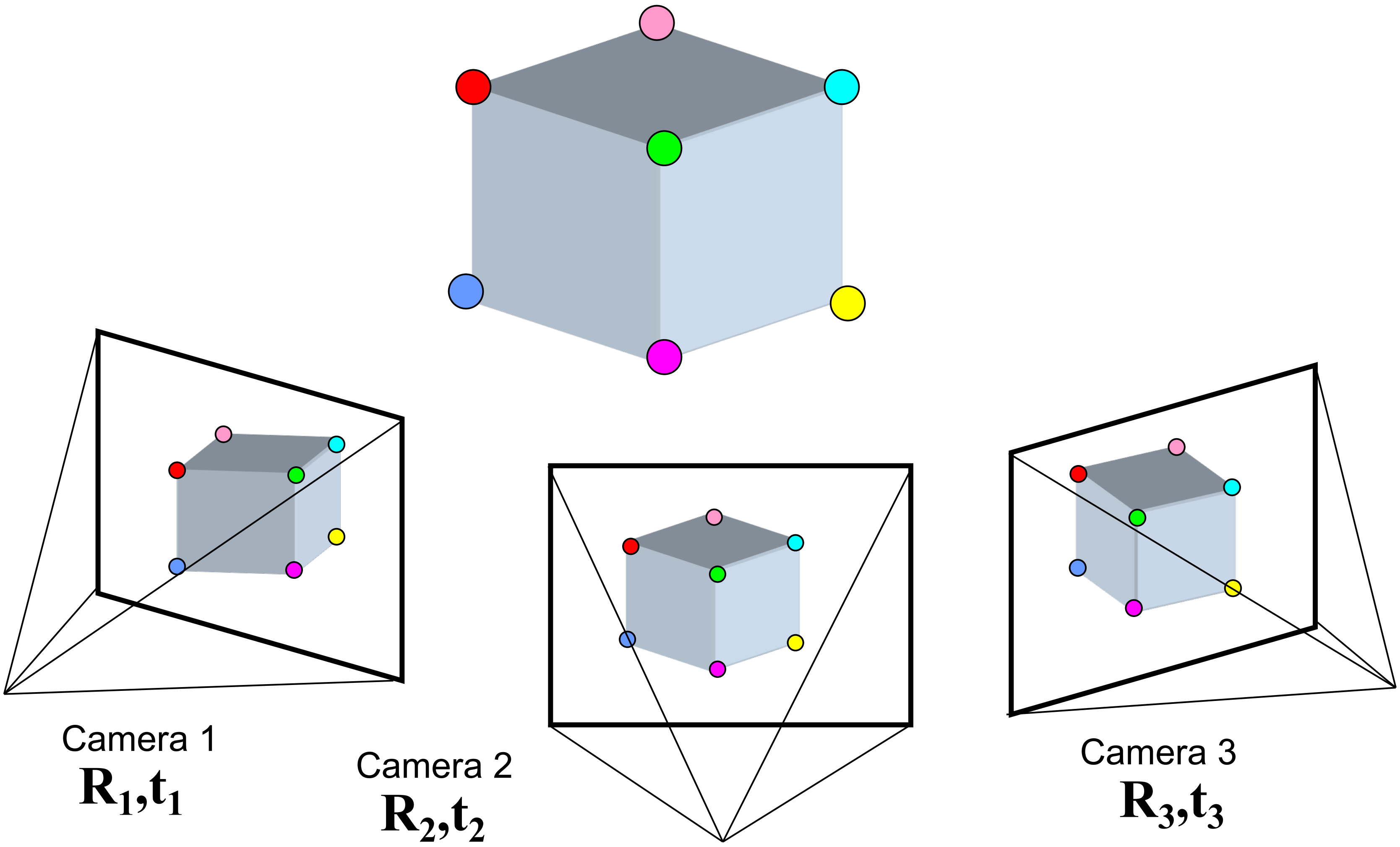
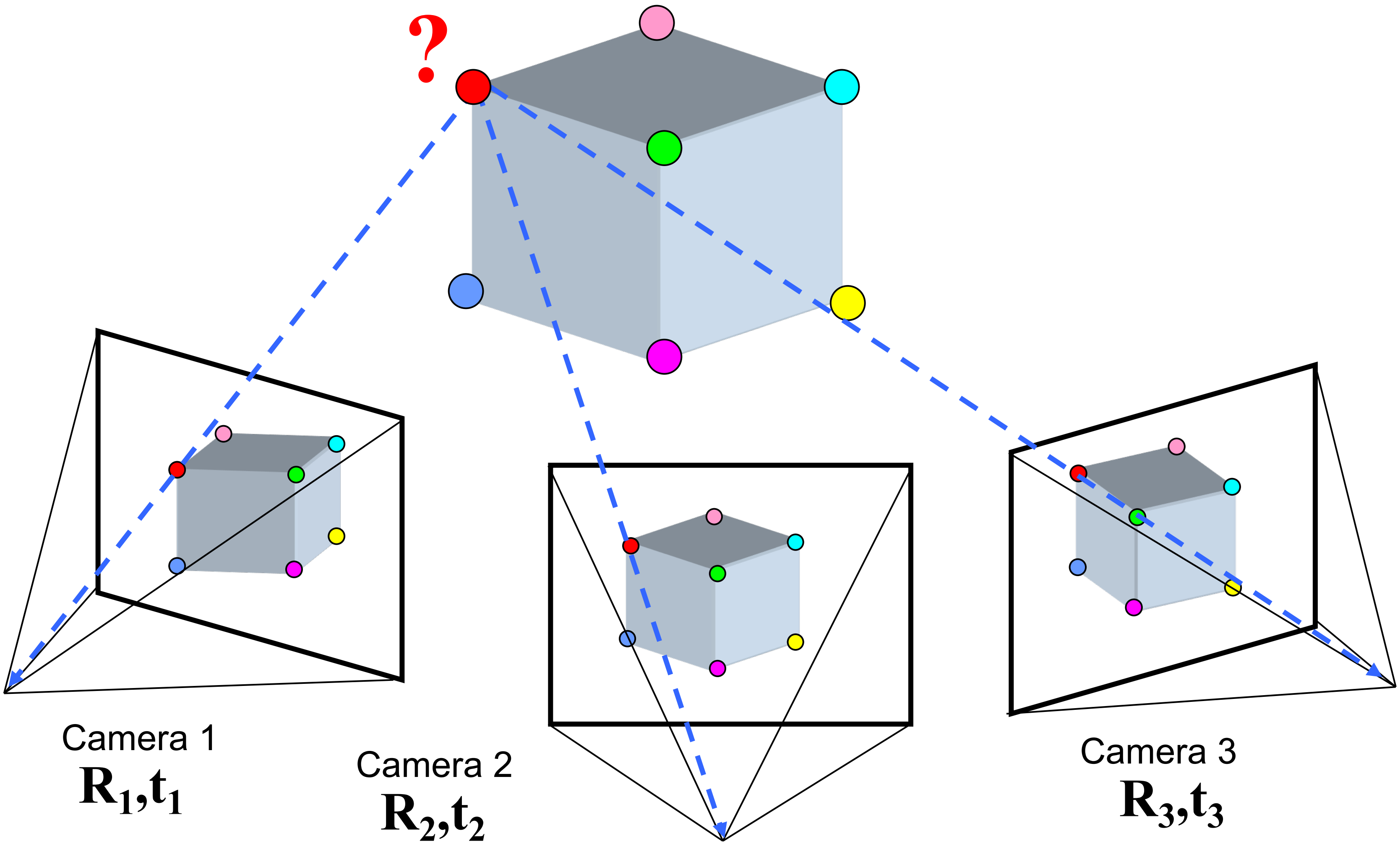


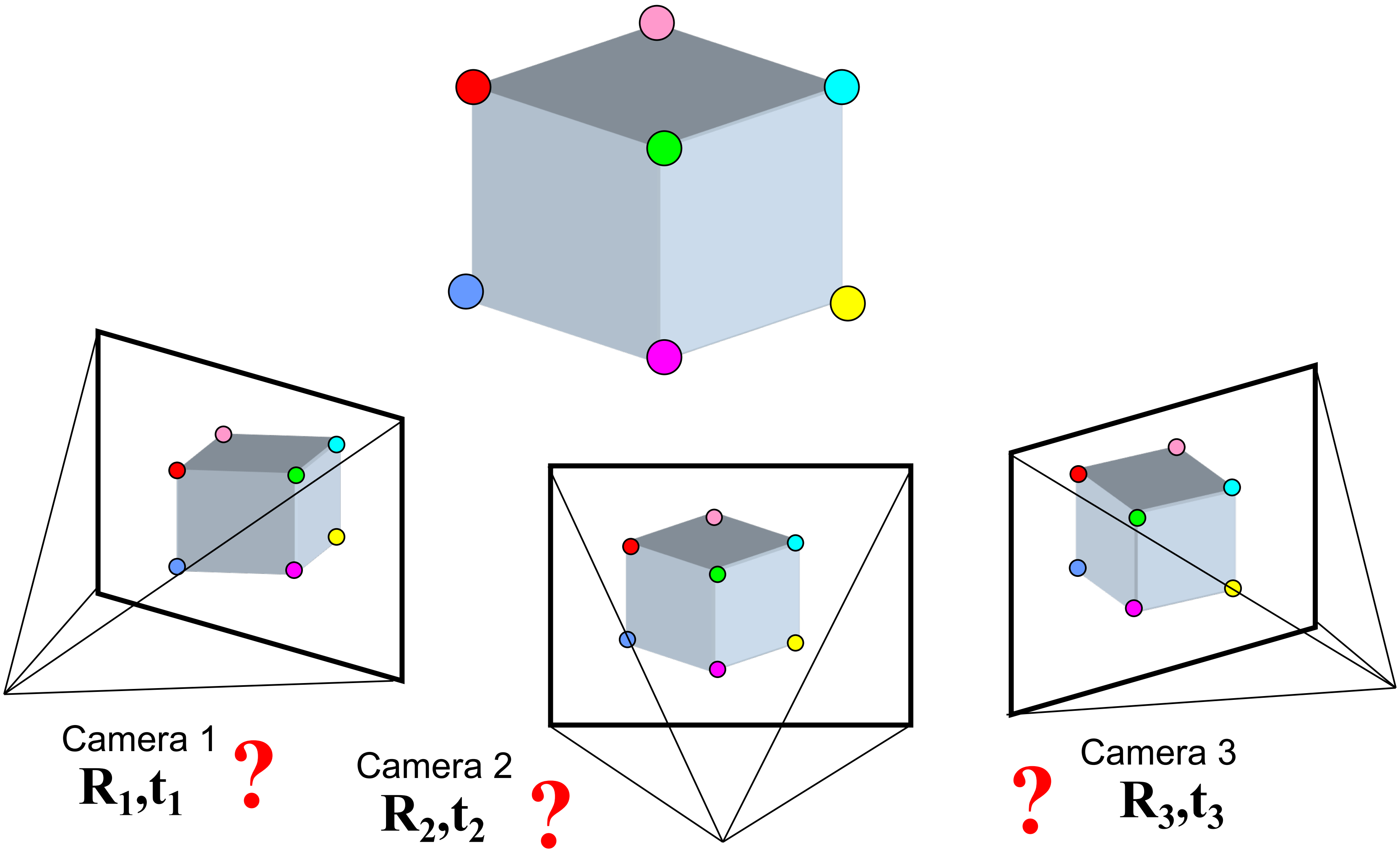
Figure credit: Noah Snavely

Structure from motion



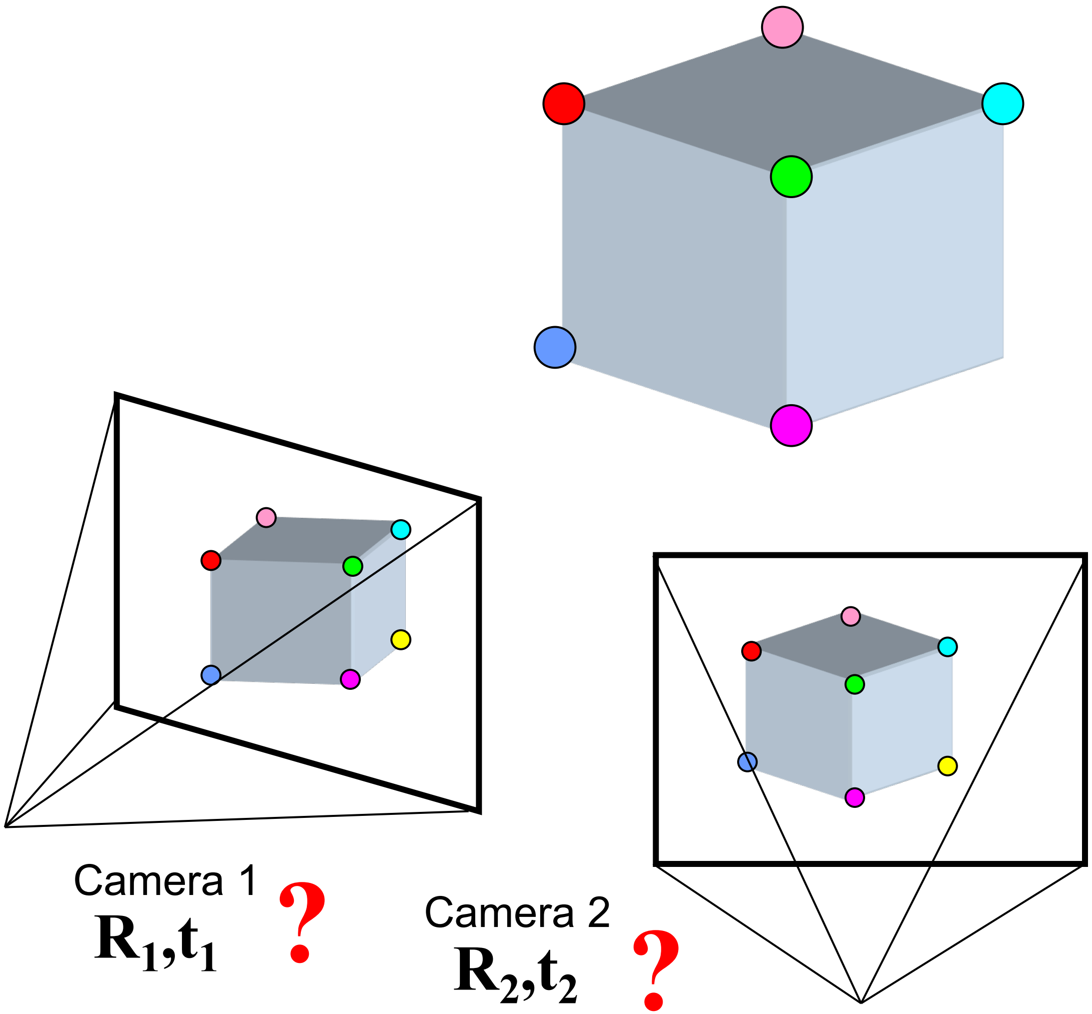
- **Structure:** Given *known cameras* and projections of the same 3D point in two or more images, compute the 3D coordinates of that point

Structure from motion



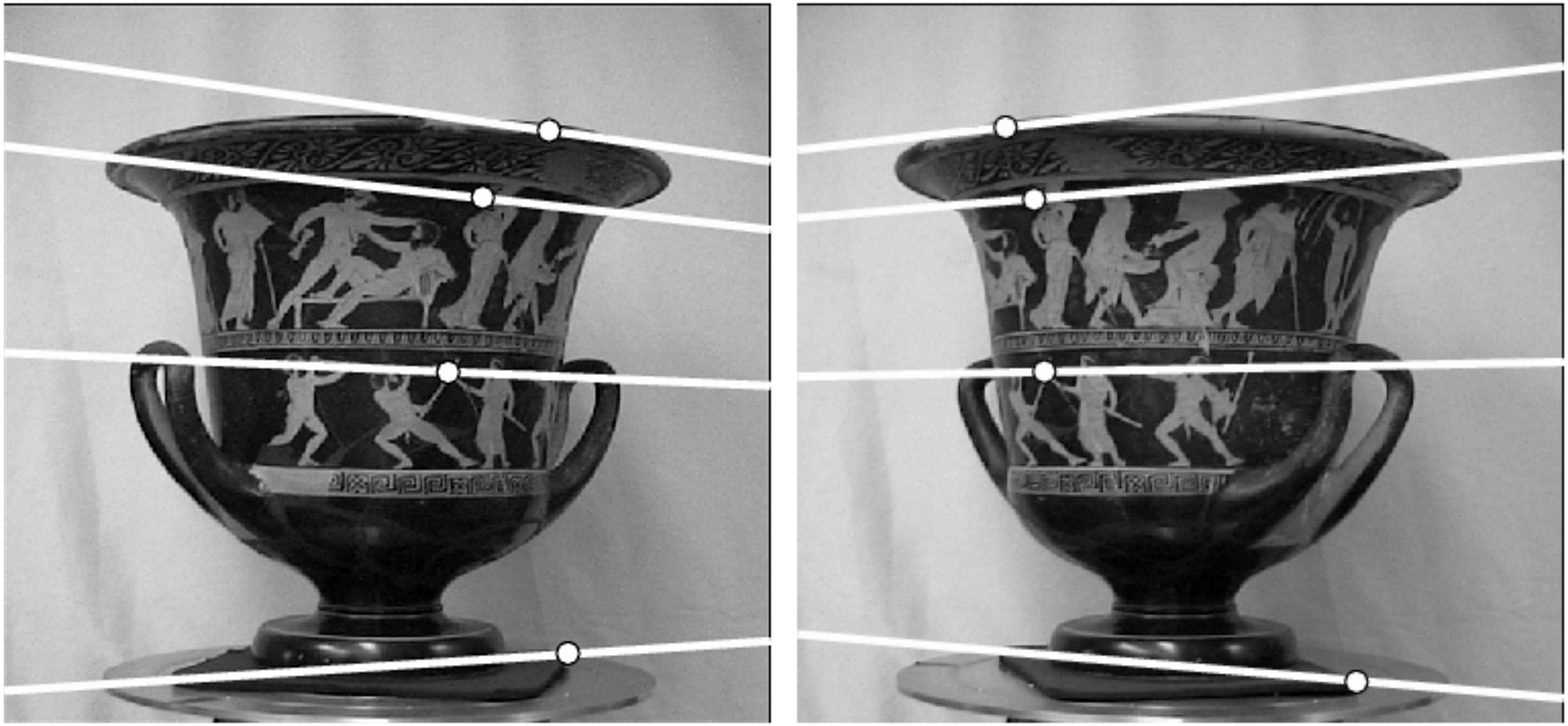
- **Motion:** Given a set of *known* 3D points seen by a camera, compute the camera parameters

Structure from motion

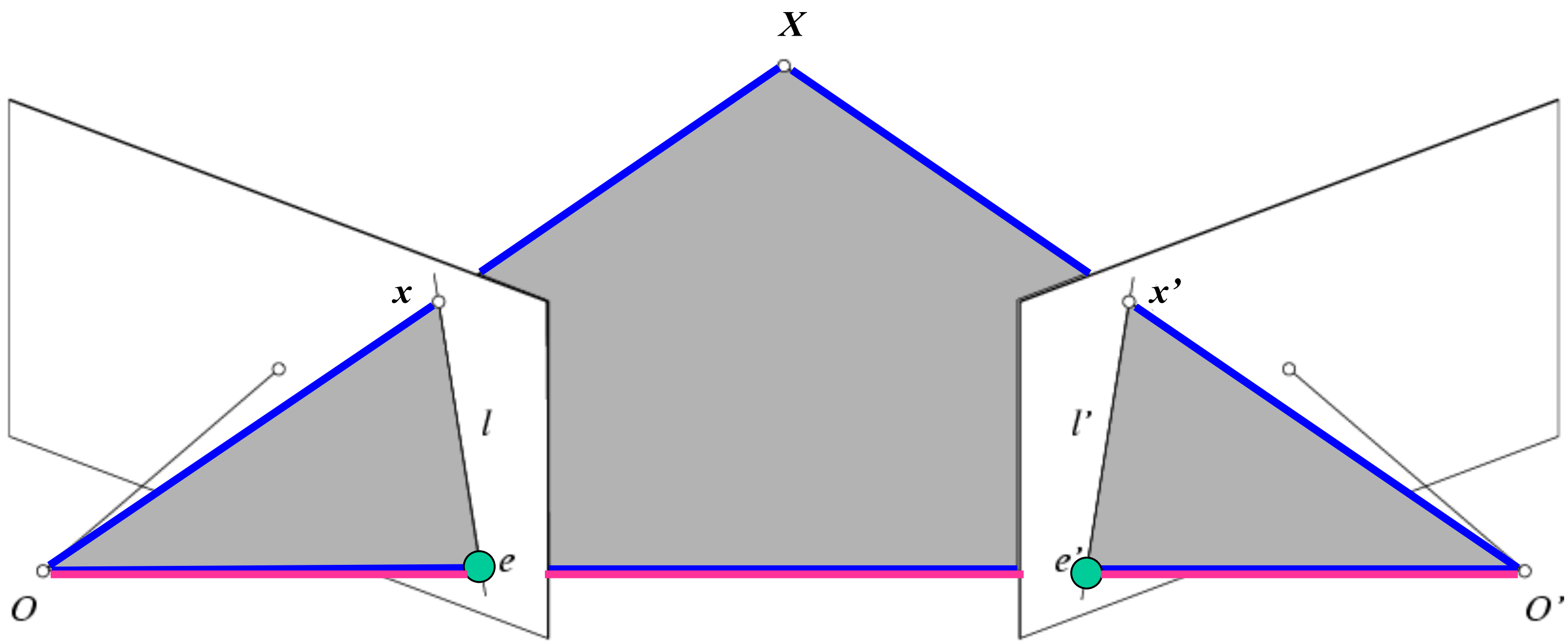


- **Bootstrapping the process:** Given a set of 2D point correspondences in *two images*, compute the camera parameters

Two-view geometry

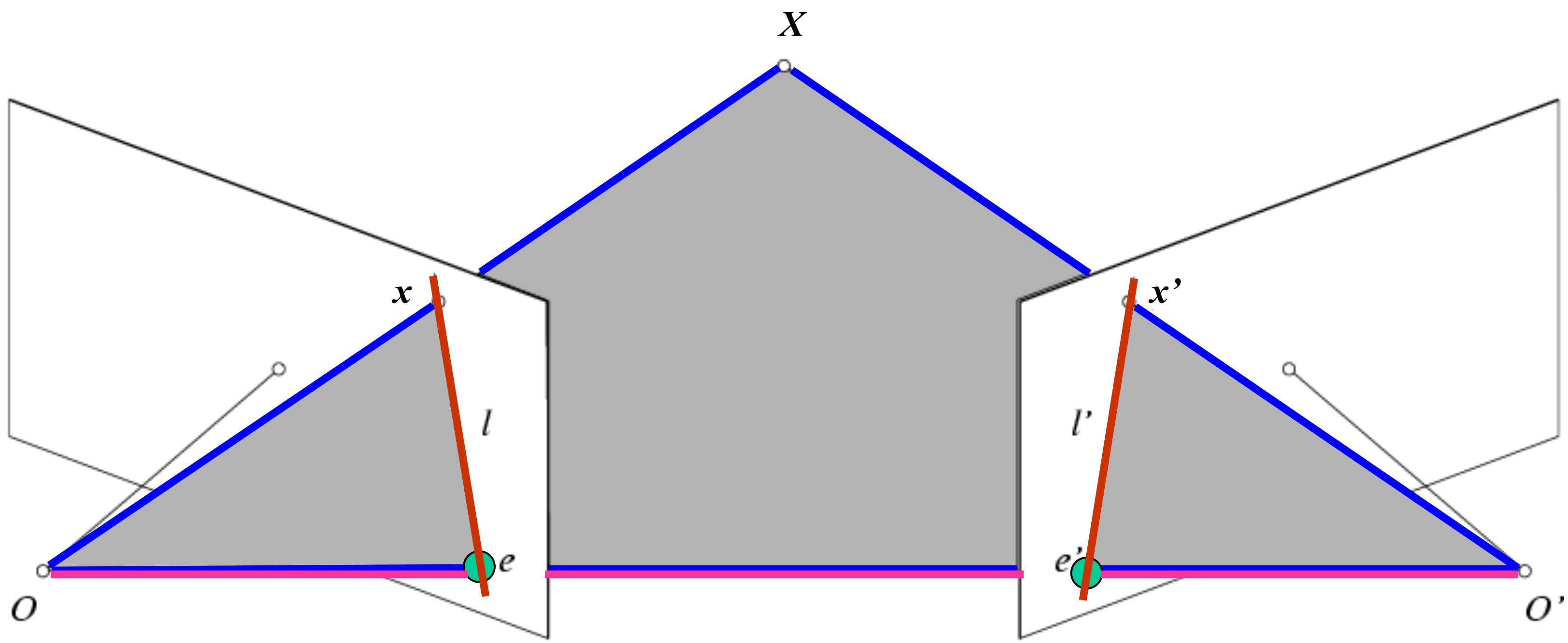


Epipolar geometry



- **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

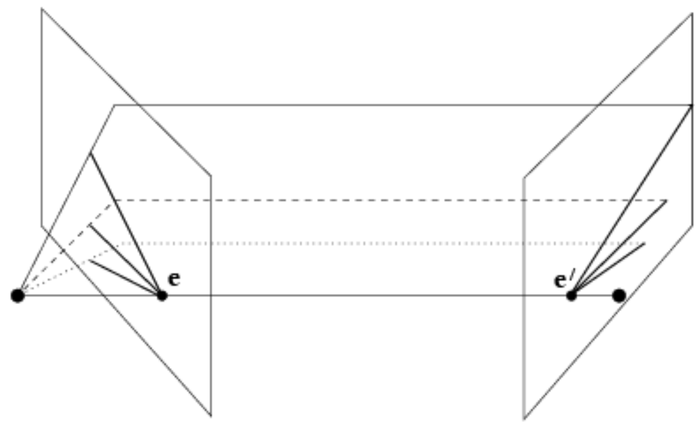
Epipolar geometry

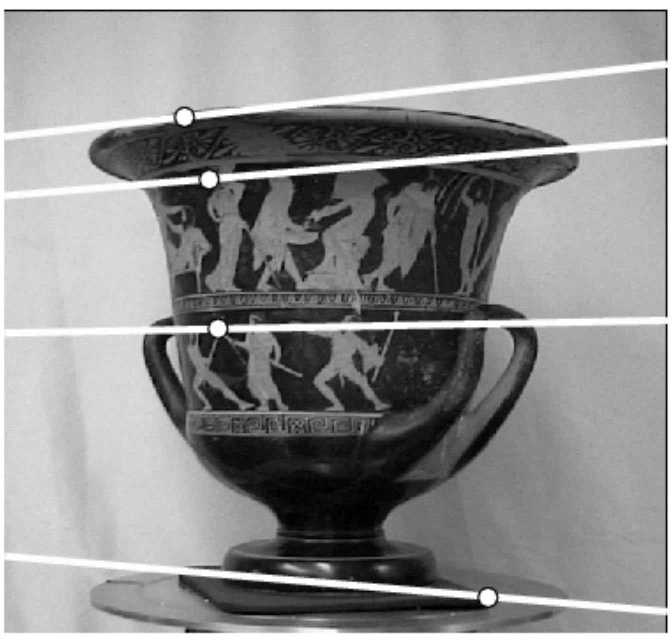


- **Baseline** – line connecting the two camera centers
- **Epipolar Plane** – plane containing baseline (1D family)
- **Epipoles**
  = intersections of baseline with image planes
  = projections of the other camera center
  = vanishing points of the motion direction
- **Epipolar Lines** - intersections of epipolar plane with image planes (always come in corresponding pairs)

Example 1

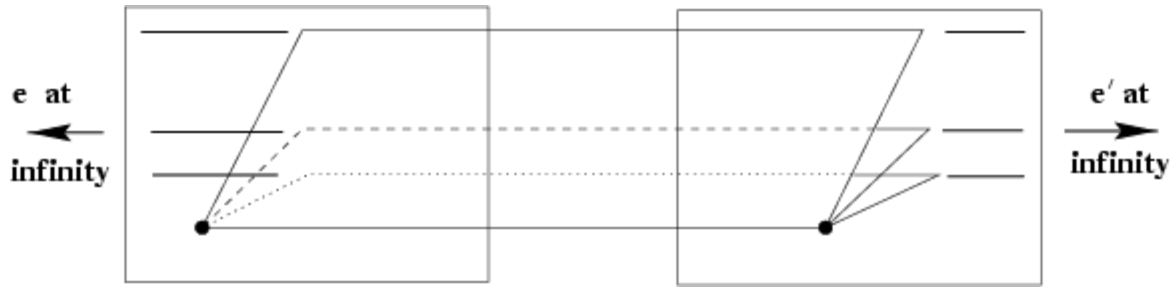
- Converging cameras





Example 2

- Motion parallel to the image plane



Example 3

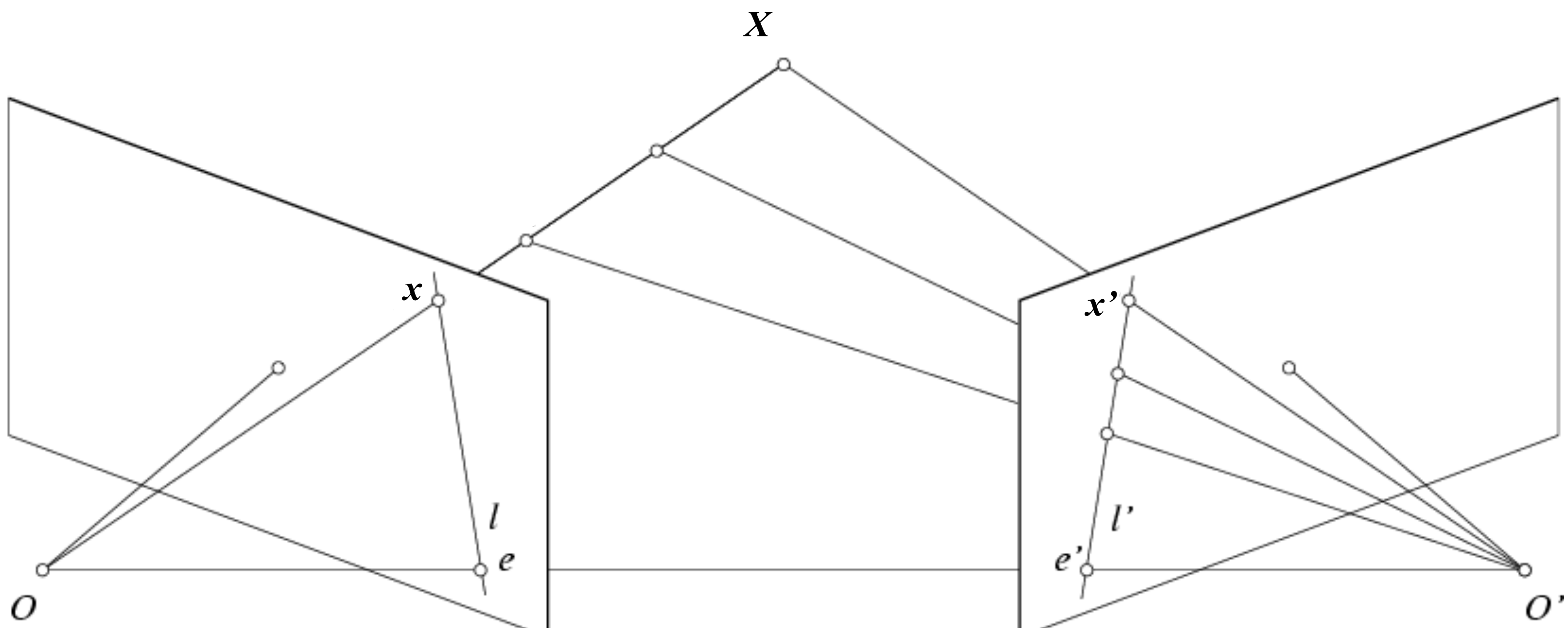
Example 3

- Motion is perpendicular to the image plane
- Epipole is the “focus of expansion” and the principal point

Motion perpendicular to image plane

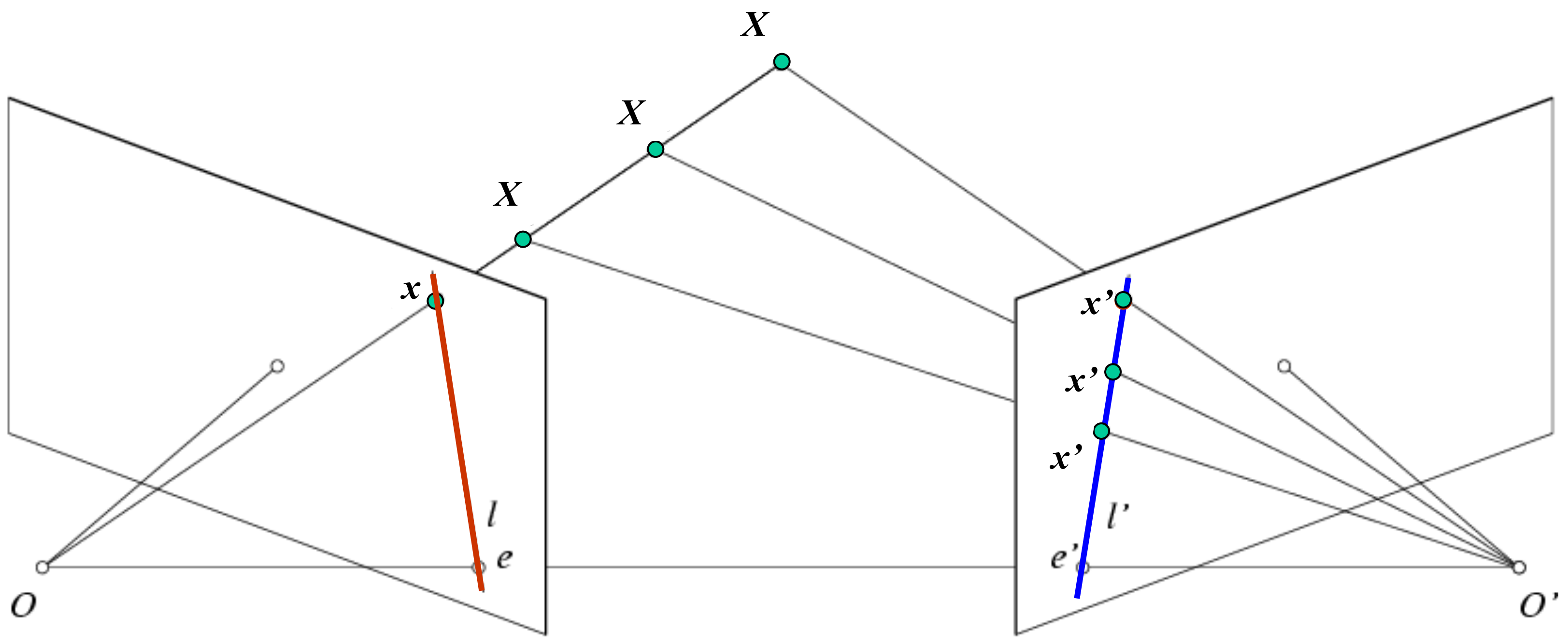
http://vimeo.com/48425421

Epipolar constraint



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

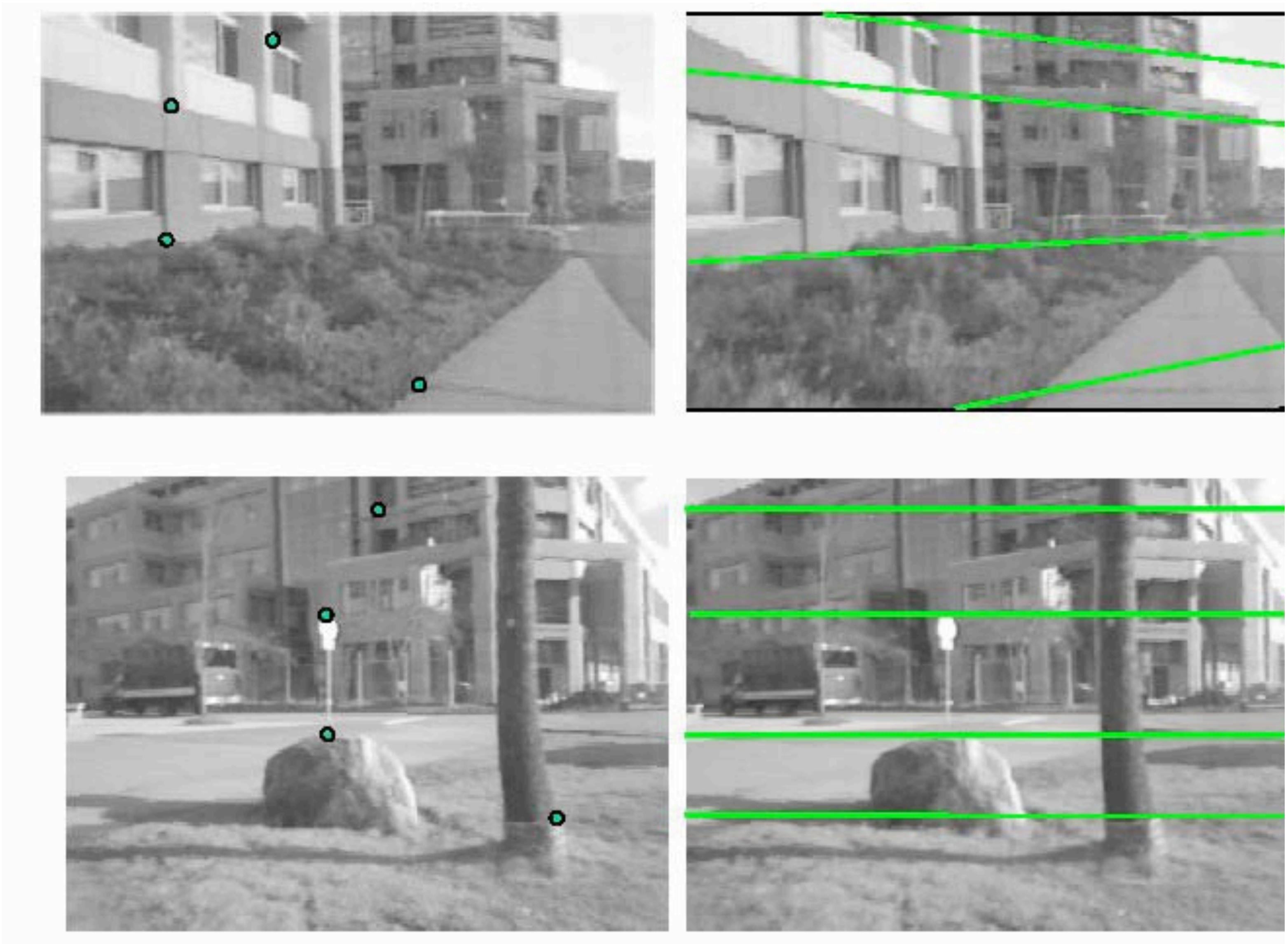
Epipolar constraint



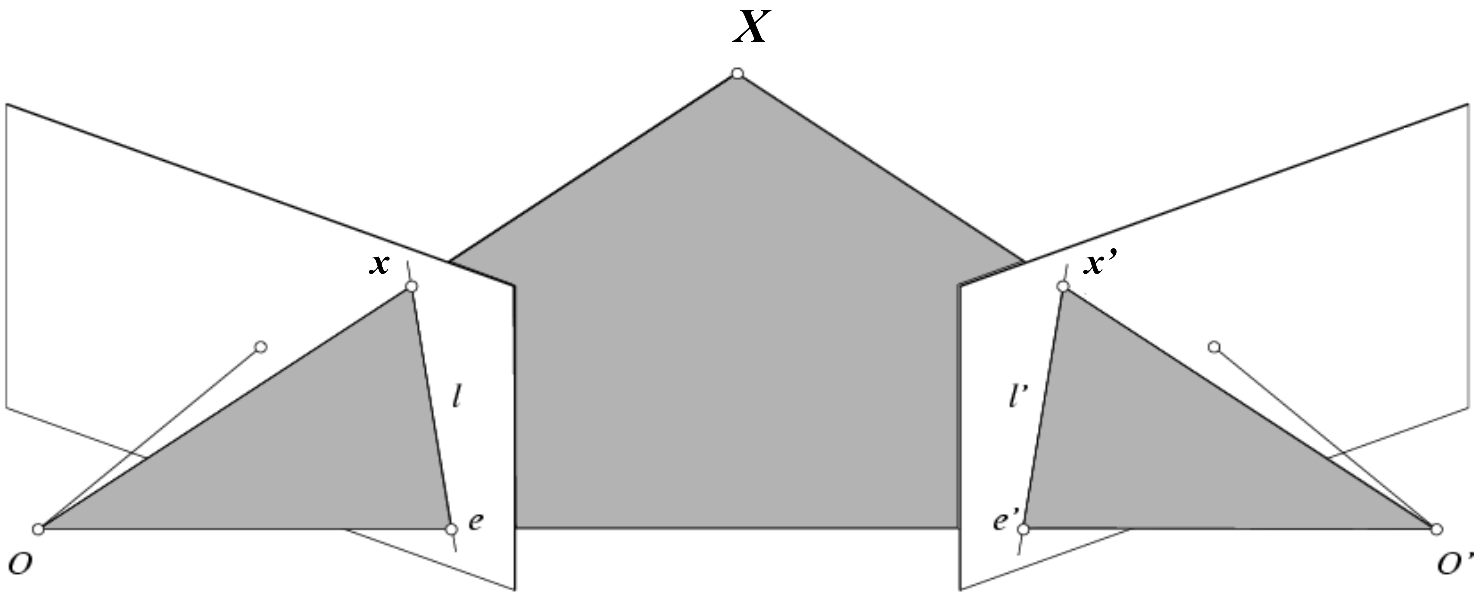
- Potential matches for ***x*** have to lie on the corresponding epipolar line ***l'***.

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

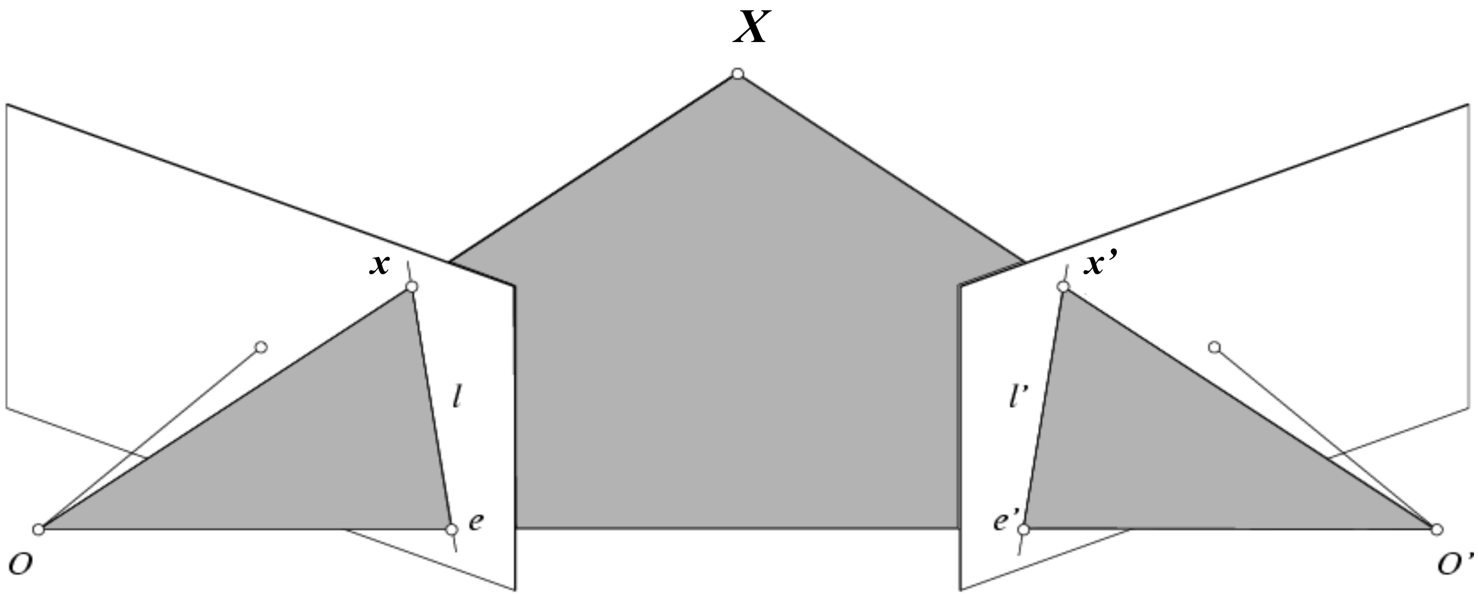
Epipolar constraint example



Epipolar constraint: Calibrated case



Epipolar constraint: Calibrated case

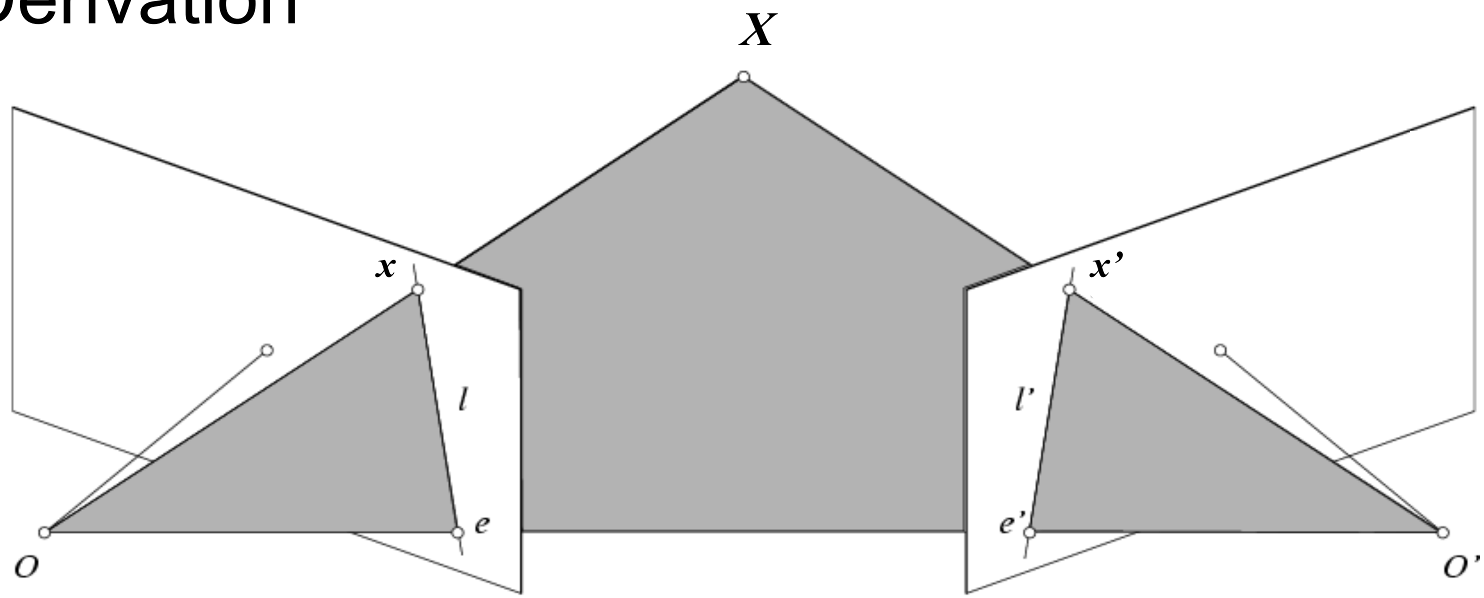


- Intrinsic and extrinsic parameters of the cameras are known, world coordinate system is set to that of the first camera
- Then the projection matrices are given by $K[I \mid 0]$ and $K'[R \mid t]$
- We can multiply the projection matrices (and the image points) by the inverse of the calibration matrices to get *normalized* image coordinates:

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

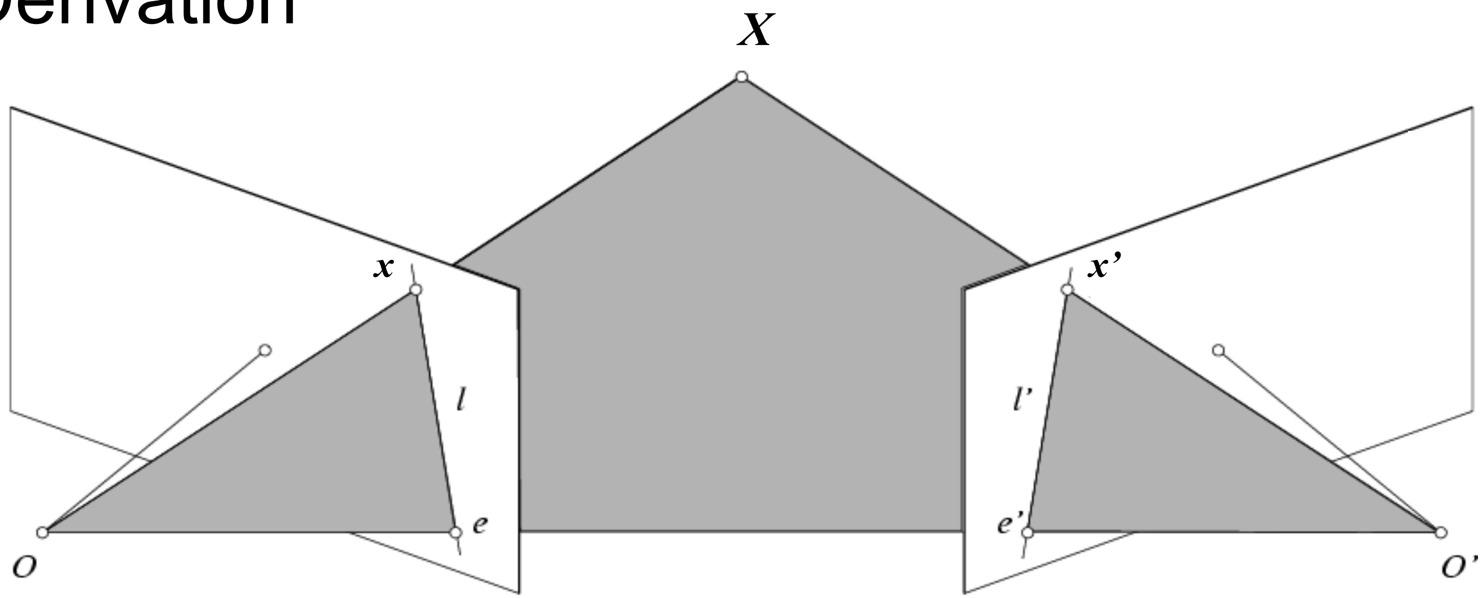
Epipolar constraint: Calibrated case

Derivation



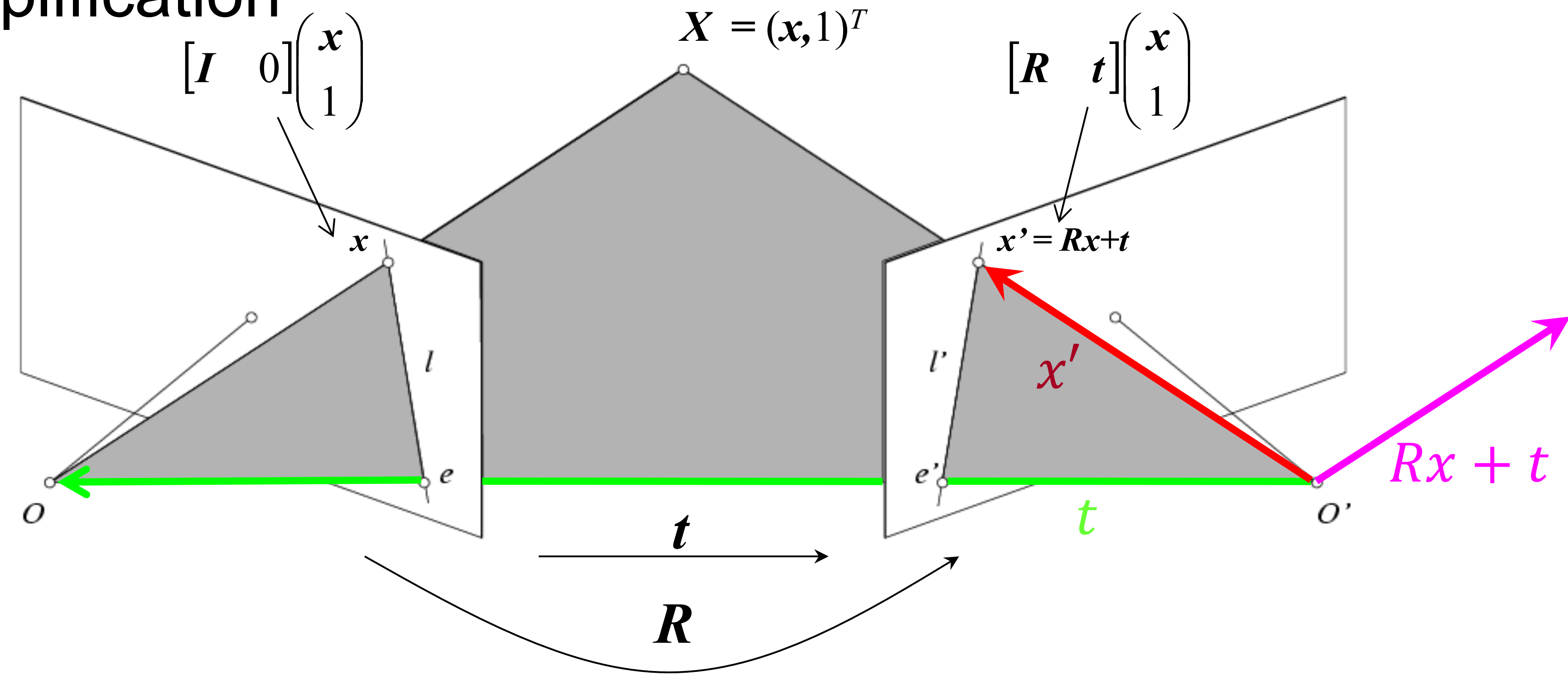
Epipolar constraint: Calibrated case

Derivation



Epipolar constraint: Calibrated case

Simplification

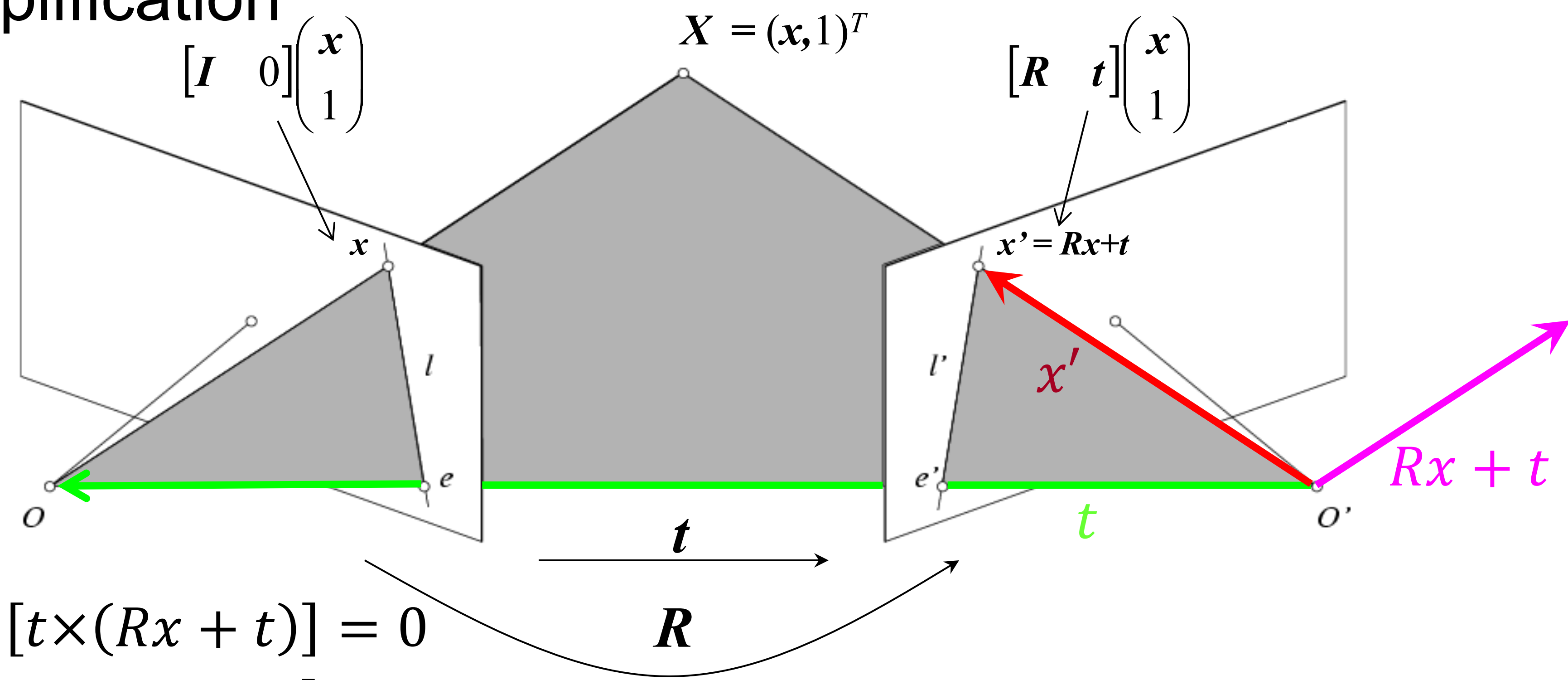


Recall:
$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}\begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = [\mathbf{a}_\times]\mathbf{b}$$

The vectors $Rx + t$, t, and x' are coplanar

Epipolar constraint: Calibrated case

Simplification



$$x'.[t\times(Rx + t)] = 0$$

$$x'.[t\times Rx + t\times t\,] = 0$$

$$x'.[t\times Rx] = 0 \quad x'^T E x = 0$$

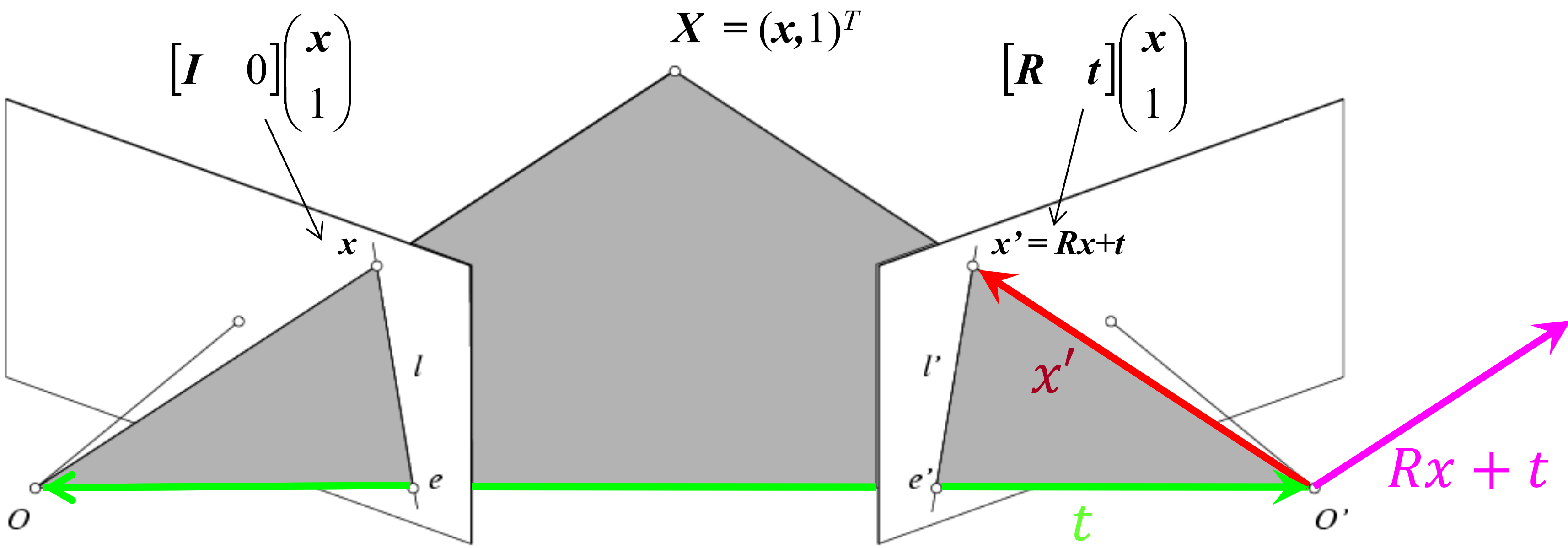
$$x'.[t_\times]Rx = 0$$

Recall:

$$\mathbf{a}\times\mathbf{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}\begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = [\mathbf{a}_\times]\mathbf{b}$$

The vectors $Rx + t$, t, and x' are coplanar

Epipolar constraint: Calibrated case

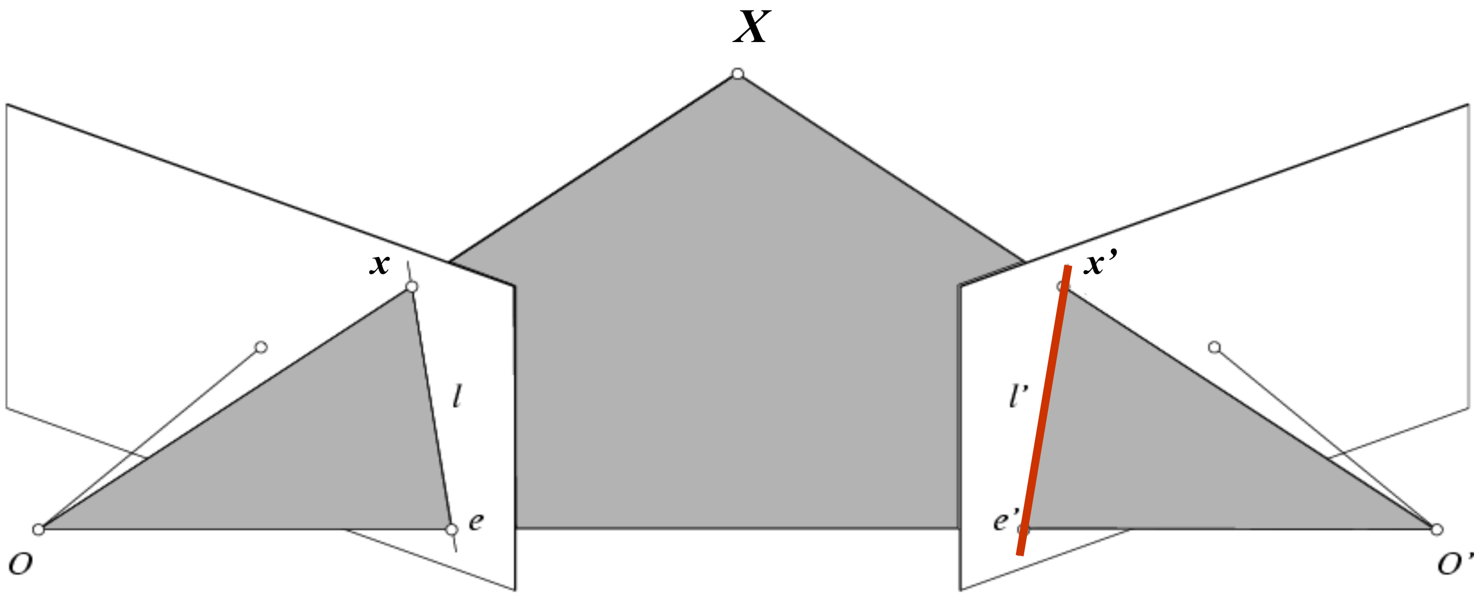


$$x' \cdot [t \times (Rx)] = 0 \quad \Rightarrow \quad x'^T [t_\times] R x = 0 \quad \Rightarrow \quad x'^T E x = 0$$

Essential Matrix
(Longuet-Higgins, 1981)

The vectors $Rx + t$, t, and x' are coplanar

Epipolar constraint: Calibrated case

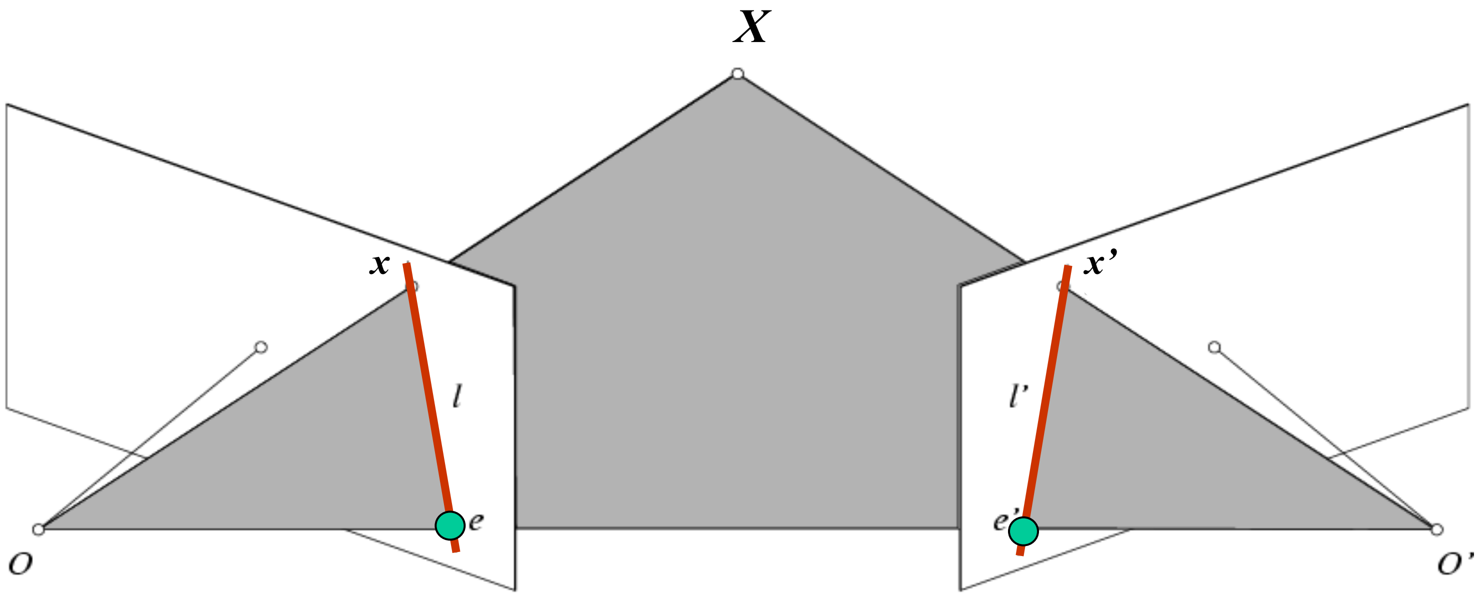


$$\boldsymbol{x}'^T E \boldsymbol{x} = 0$$

- ***E x*** is the epipolar line associated with ***x*** (***l'*** = ***E x***)
  - Recall: a line is given by $ax + by + c = 0$ or

$$\mathbf{l}^T \mathbf{x} = 0 \quad \text{where} \quad \mathbf{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

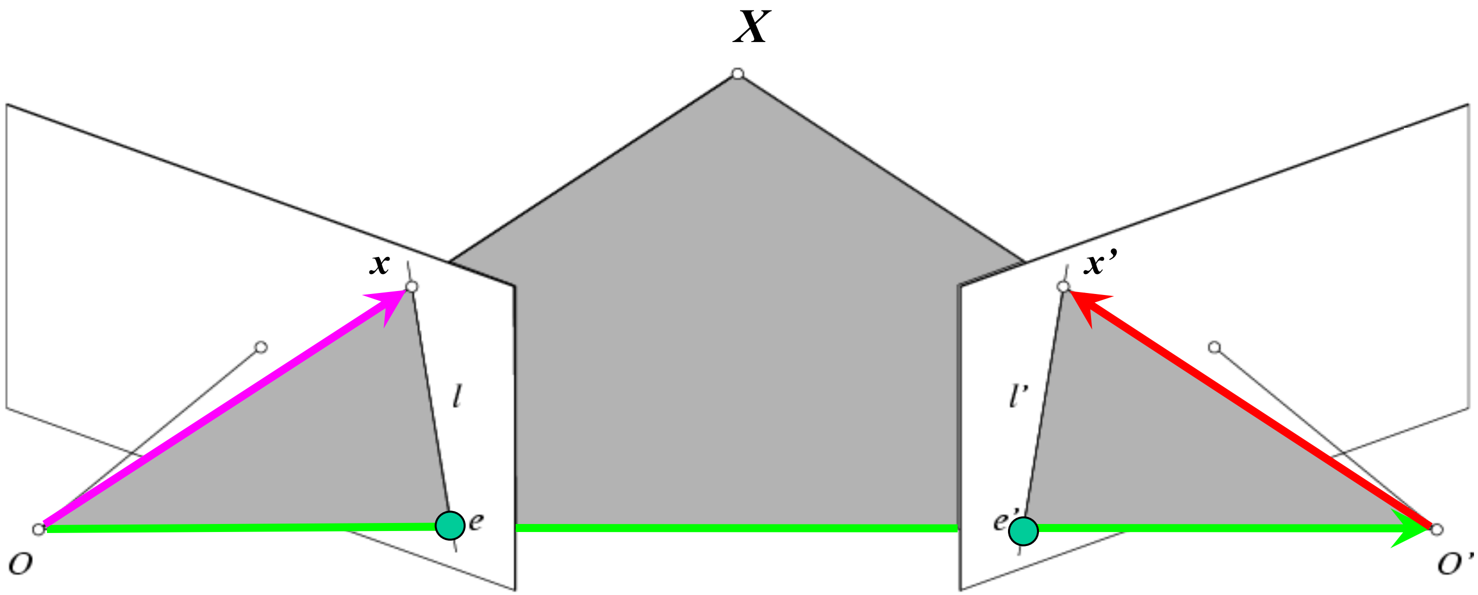
Epipolar constraint: Calibrated case



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

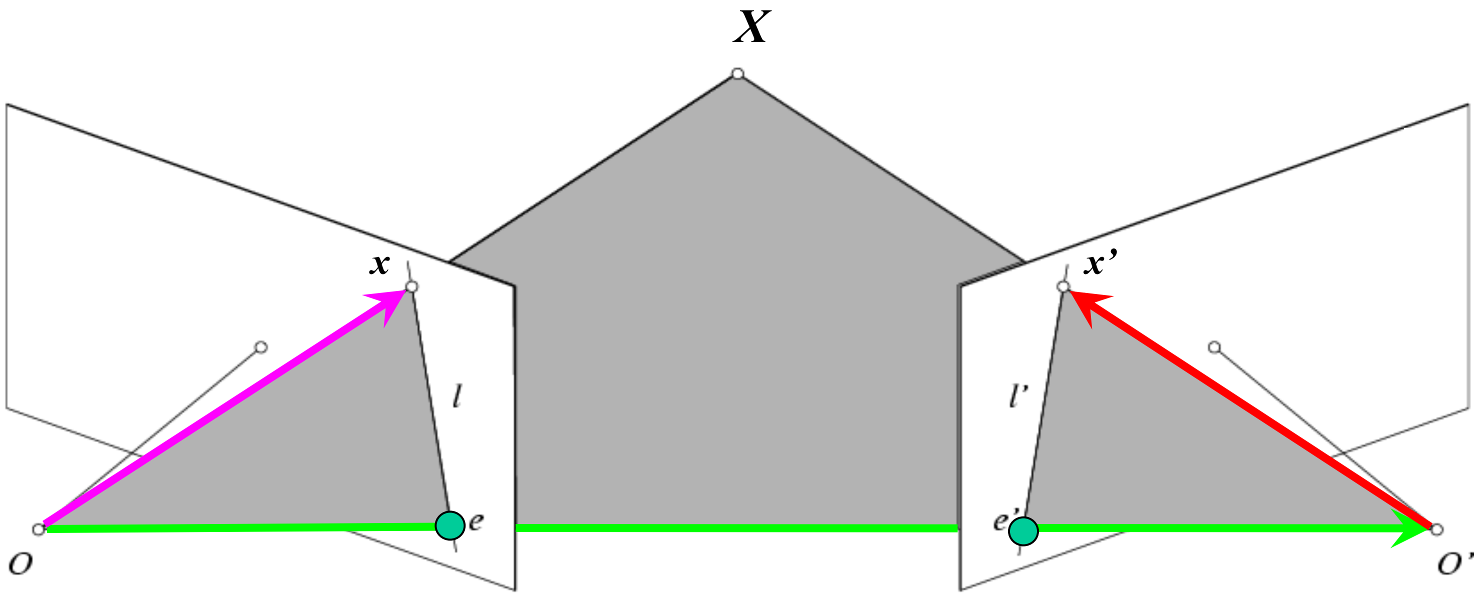
- ***E x*** is the epipolar line associated with ***x*** (***l'*** = ***E x***)
- ***E^{T}x'*** is the epipolar line associated with ***x'*** (***l*** = ***E^{T}x'***)
- ***E e*** = 0 and ***E^{T}e'*** = *0*
- ***E*** is singular (rank two)
- ***E*** has five degrees of freedom

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:

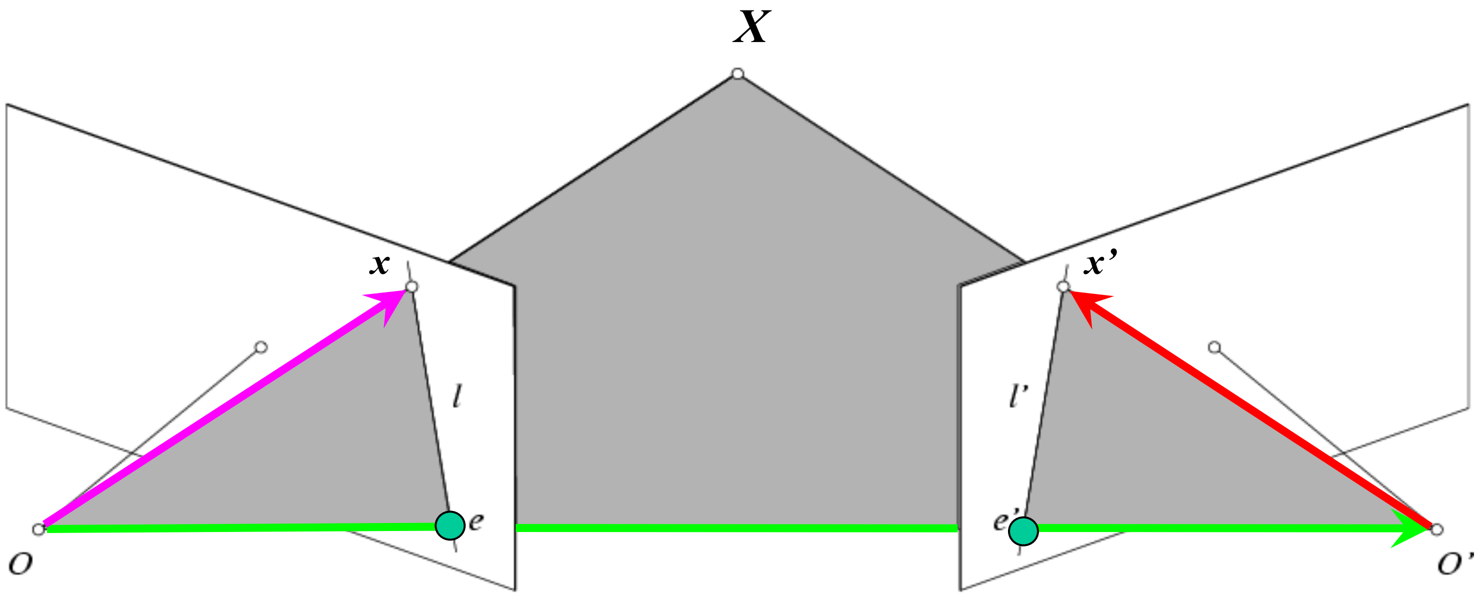
Epipolar constraint: Uncalibrated case



- 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



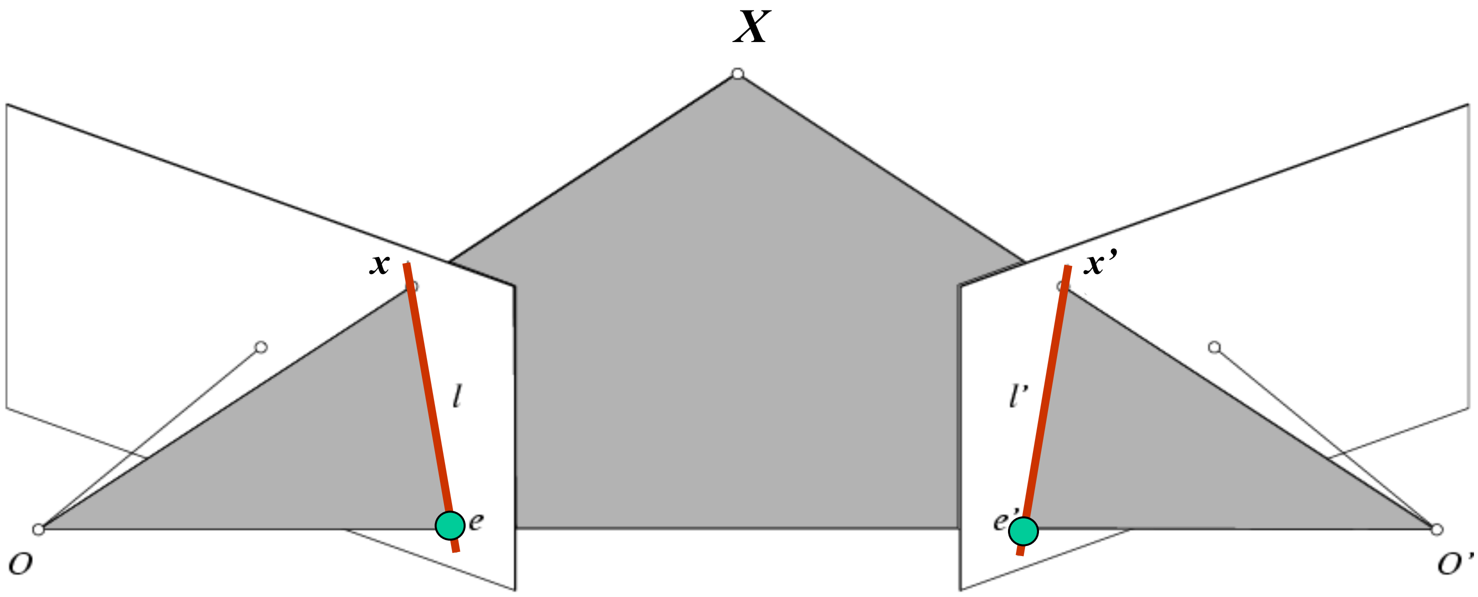
$$\hat{x}'^T E \hat{x} = 0 \quad \Rightarrow \quad x'^T F x = 0 \quad \text{with} \quad F = K'^{-T} E K^{-1}$$

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

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

Fundamental Matrix
(Faugeras and Luong, 1992)

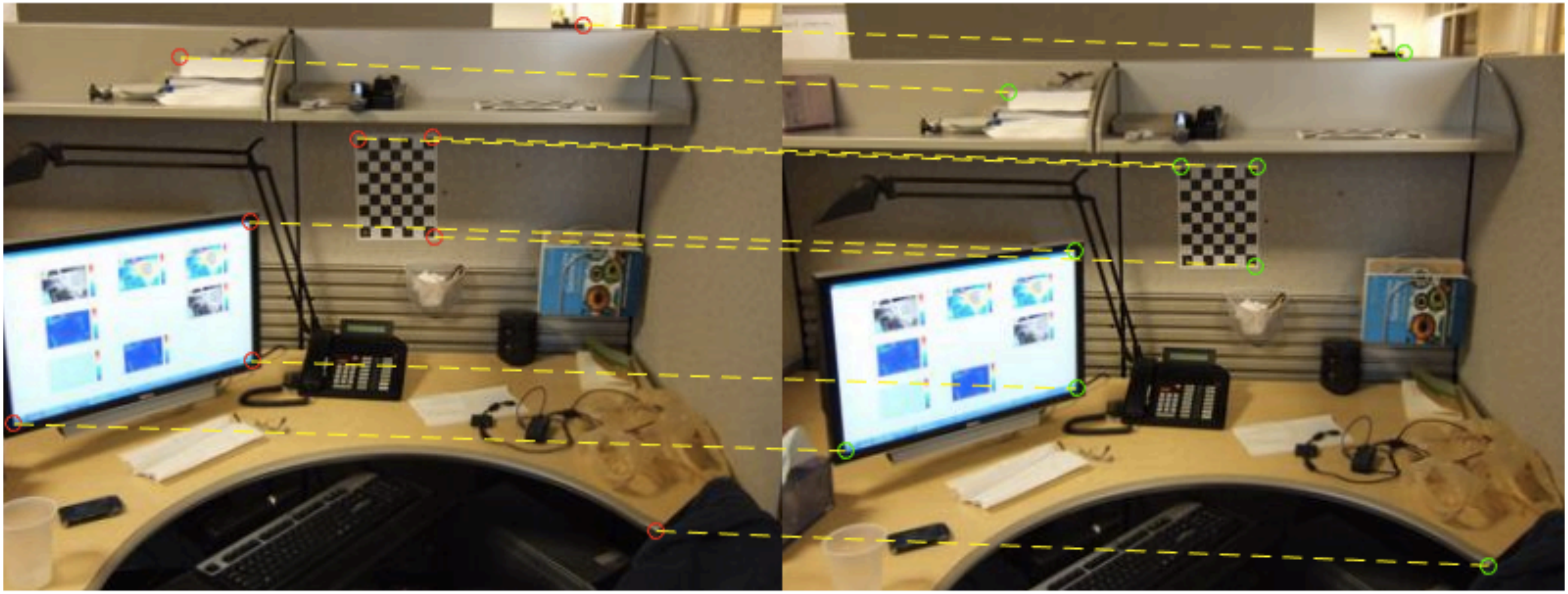
Epipolar constraint: Uncalibrated case



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

- $\boldsymbol{F}\,\boldsymbol{x}$ 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

Estimating the fundamental matrix



The eight-point algorithm

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

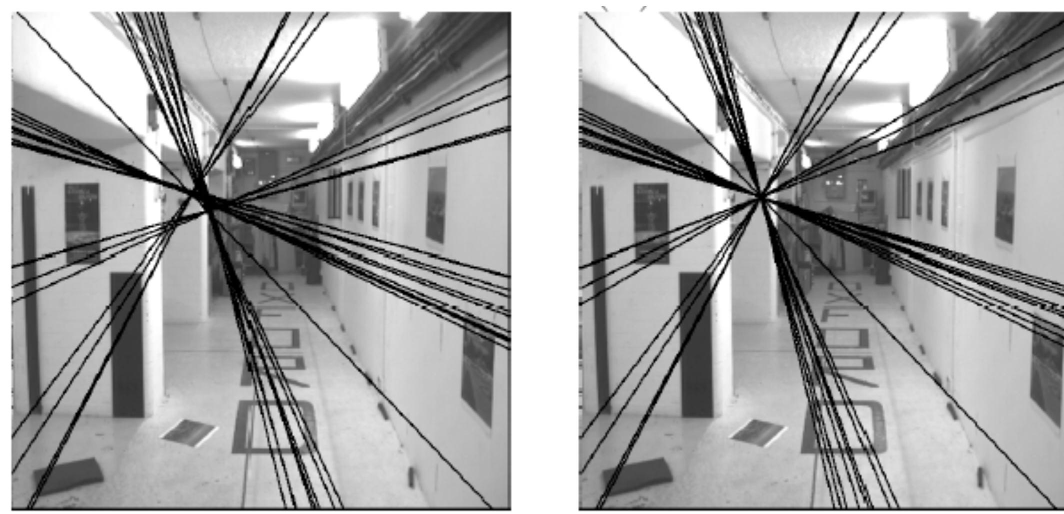
The eight-point algorithm

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

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

Solve homogeneous linear system using eight or more matches

Enforce rank-2 constraint (take SVD of ***F*** and throw out the smallest singular value)



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 $T'^{T} F\, T$

Seven-point algorithm

- Set up least squares system with seven pairs of correspondences and solve for null space (two vectors) using SVD
- Solve for linear combination of null space vectors that satisfies det(F)=0

Source: D. Hoiem

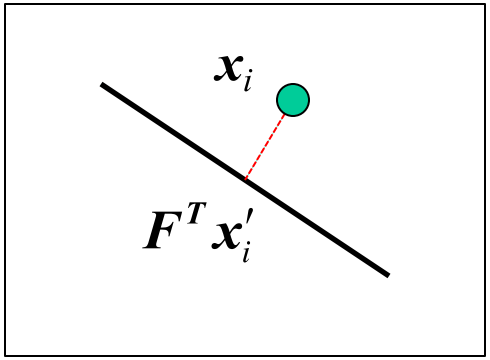
Nonlinear estimation

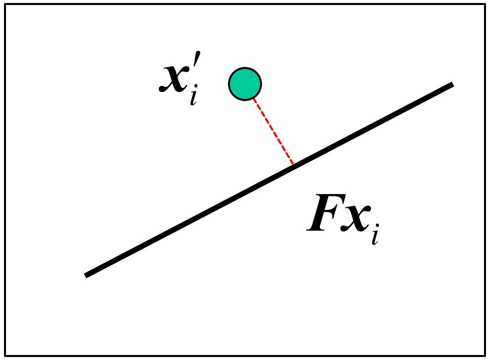
- Linear estimation minimizes the sum of squared *algebraic* distances between points $\boldsymbol{x}'_i$ and epipolar lines $\boldsymbol{F}\,\boldsymbol{x}_i$ (or points $\boldsymbol{x}_i$ and epipolar lines $\boldsymbol{F}^T\boldsymbol{x}'_i$):

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

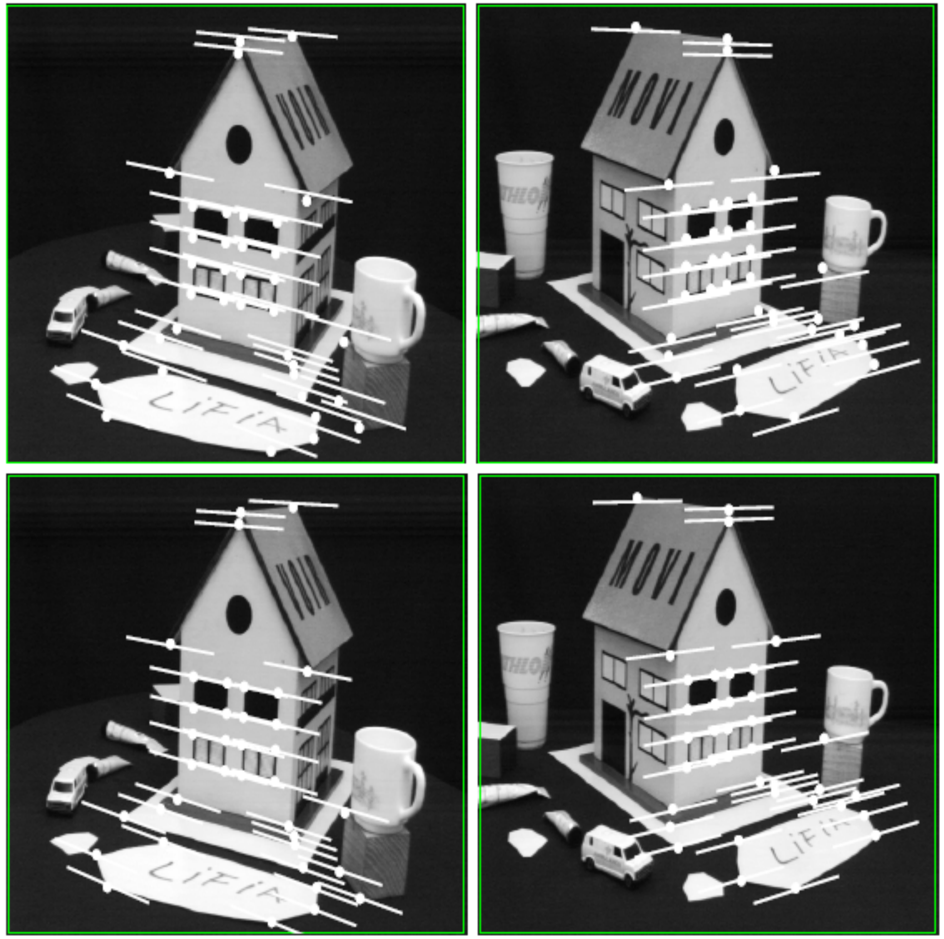
- Nonlinear approach: minimize sum of squared *geometric* distances

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



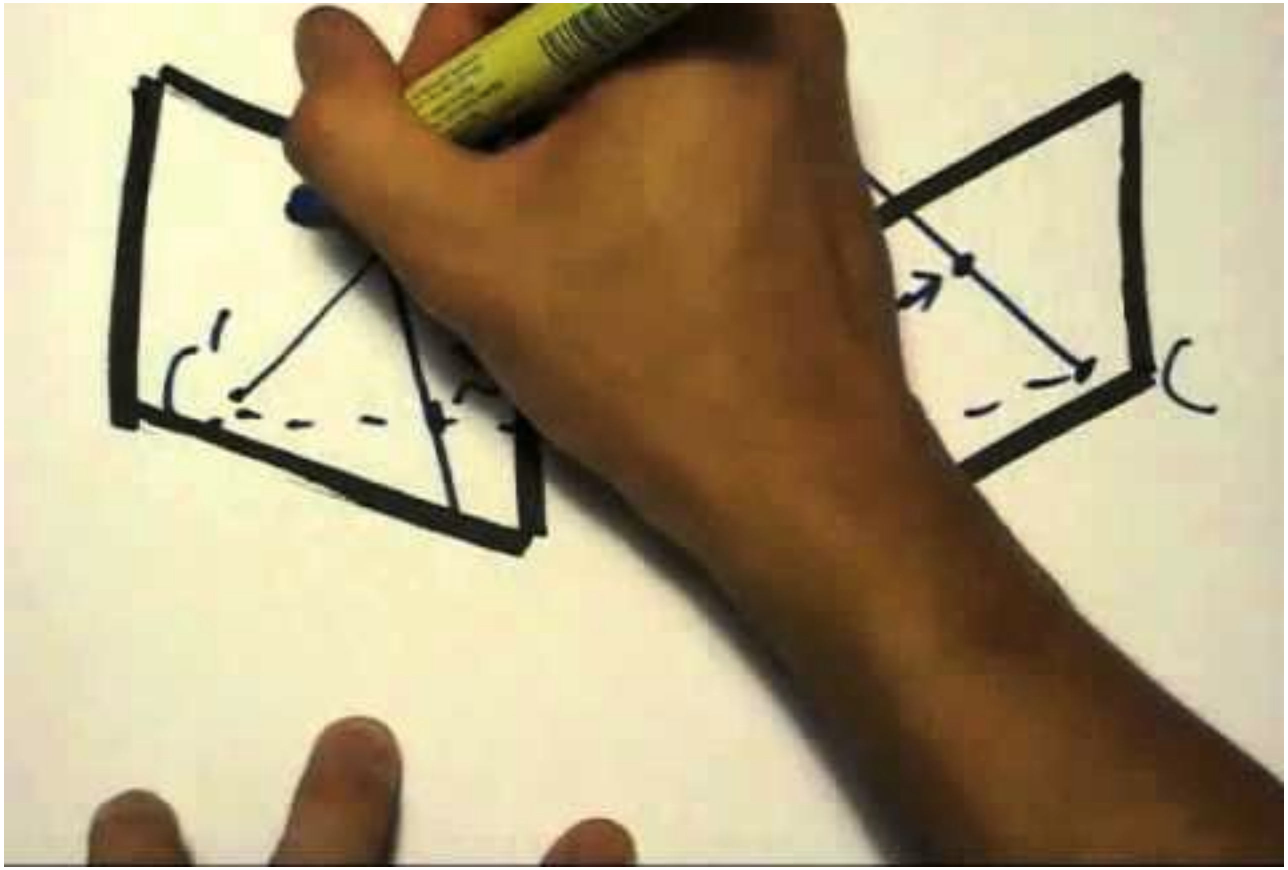


Comparison of estimation algorithms



	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

The Fundamental Matrix Song



http://danielwedge.com/fmatrix/

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
- Alternatively, if the calibration matrices are known, the five-point algorithm can be used to estimate relative camera pose