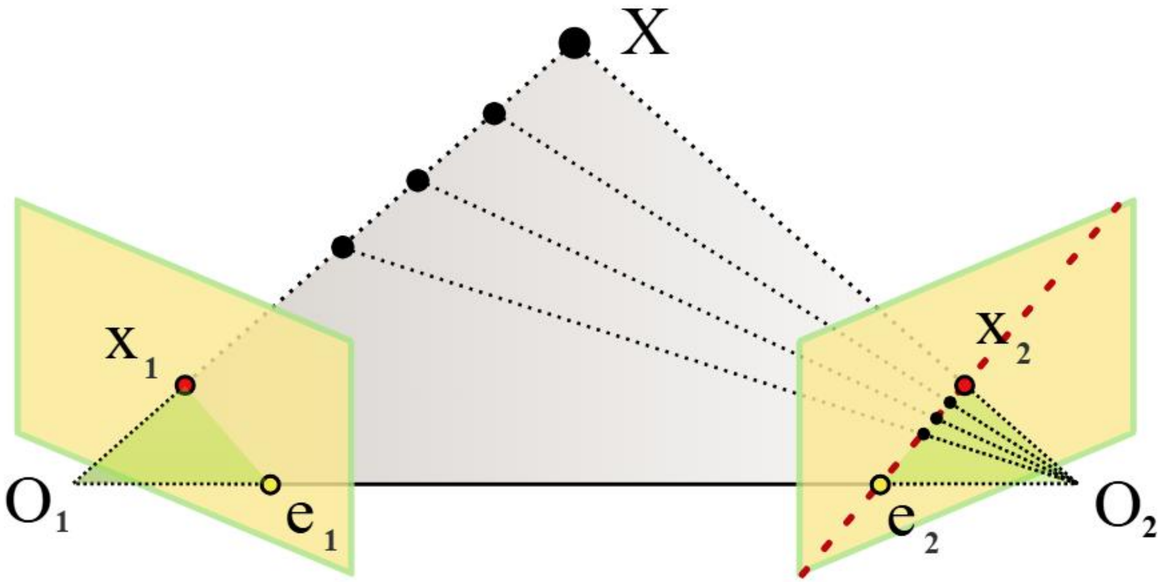
8. 2-View Geometry

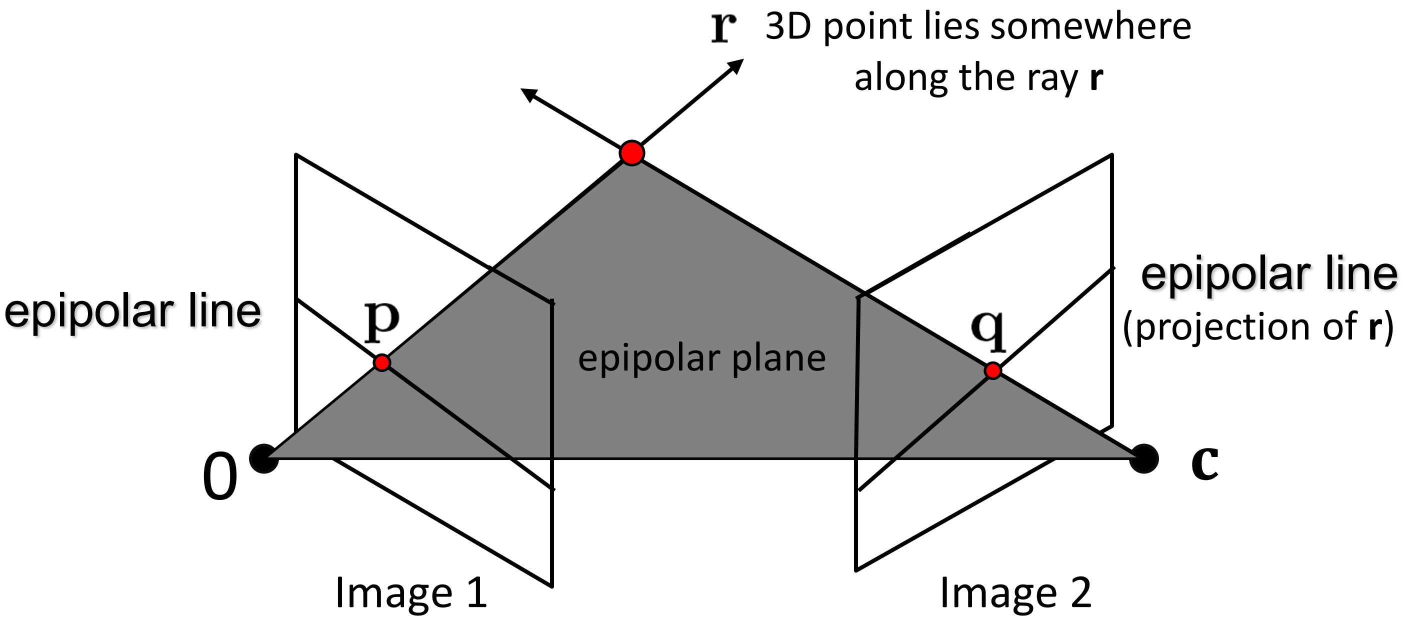


Outline

- Two-view Geometry
- Triangulation

Two-view Geometry

- What if two cameras see the same point?
  - The corresponding point in the second view must lie on a line, the epipolar line
  - The two camera centers and one 3D point defines the epipolar plane
- What can we say about these quantities?



Sophie GERMAIN (portrait de), à l'âge de 14 ans.

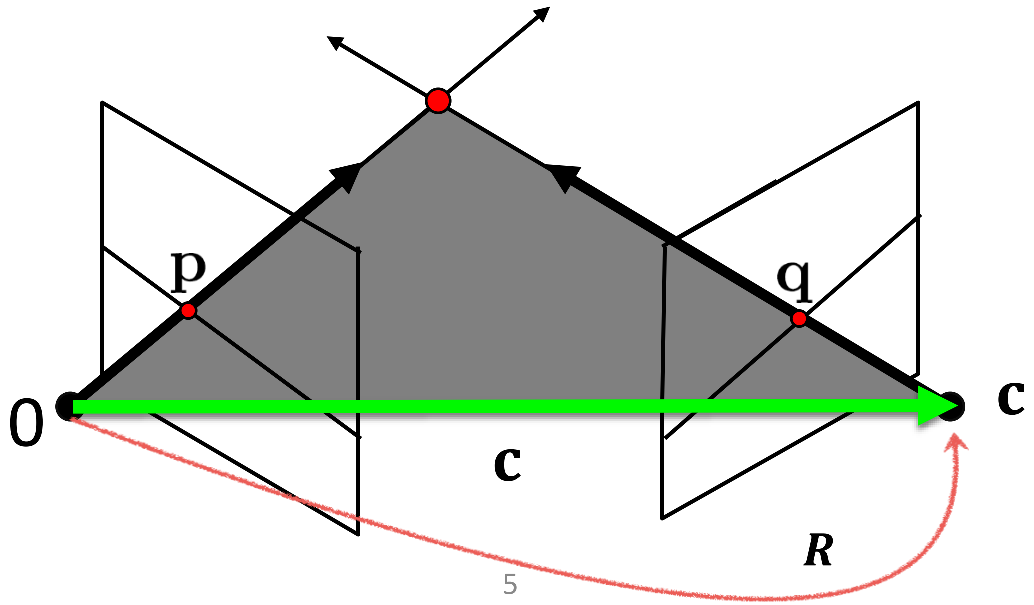
Algebra is but written geometry;
geometry is but drawn algebra.
-- Sophie Germain

Essential Matrix – calibrated case

- Assume calibrated camera
  - So that we know 3D directions in the camera coordinate system

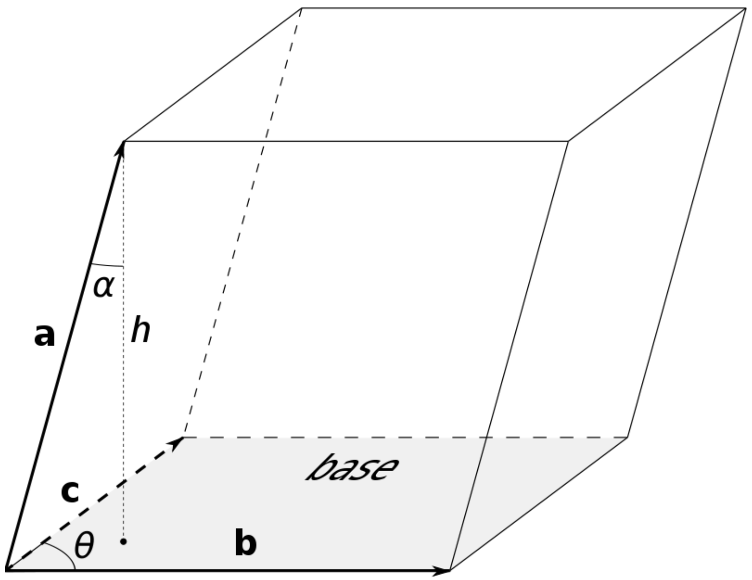
  i.e. $\mathbf{p}, \mathbf{q}$ are known directions ($\mathbf{p}$ is in camera frame 1, $\mathbf{q}$ is in camera frame 2)
  - Suppose $\mathbf{R}, \mathbf{c}$ are the rotations and translations between the two cameras

  i.e. $\mathbf{R}^{\mathrm{T}}\mathbf{q}$ is the direction of $\mathbf{q}$ in the camera frame 1
  - Constraint: $\mathbf{p}, \mathbf{c}, \mathbf{R}^{\mathrm{T}}\mathbf{q}$ are coplanar (and all in camera frame 1)



Vector Mixed Product

- Vector mixed product: $a \cdot (b \times c)$
- Geometric meaning: the volume of a parallelepiped defined by the three vectors $a, b,$ and c
- Three vectors a, b, c are coplanar iff $a \cdot (b \times c) = 0$



from wikepedia

Essential Matrix – calibrated case

- Assume calibrated camera
  - Constraint: $\mathbf{p}, \mathbf{c}, \mathbf{R}^{\mathrm{T}}\mathbf{q}$ are coplanar (and all in camera frame 1)

$$\mathbf{p} \cdot \left(\mathbf{c} \times \mathbf{R}^{\mathrm{T}}\mathbf{q}\right) = 0$$

➔

$$\mathbf{p}^{\mathrm{T}}[\mathbf{c}]_{\times}\mathbf{R}^{\mathrm{T}}\mathbf{q} = 0$$

➔

$$\mathbf{q}^{\mathrm{T}}\mathbf{R}[\mathbf{c}]_{\times}\mathbf{p} = \mathbf{q}^{\mathrm{T}}\mathbf{E}\mathbf{p} = 0$$

$$[a]_{\times} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

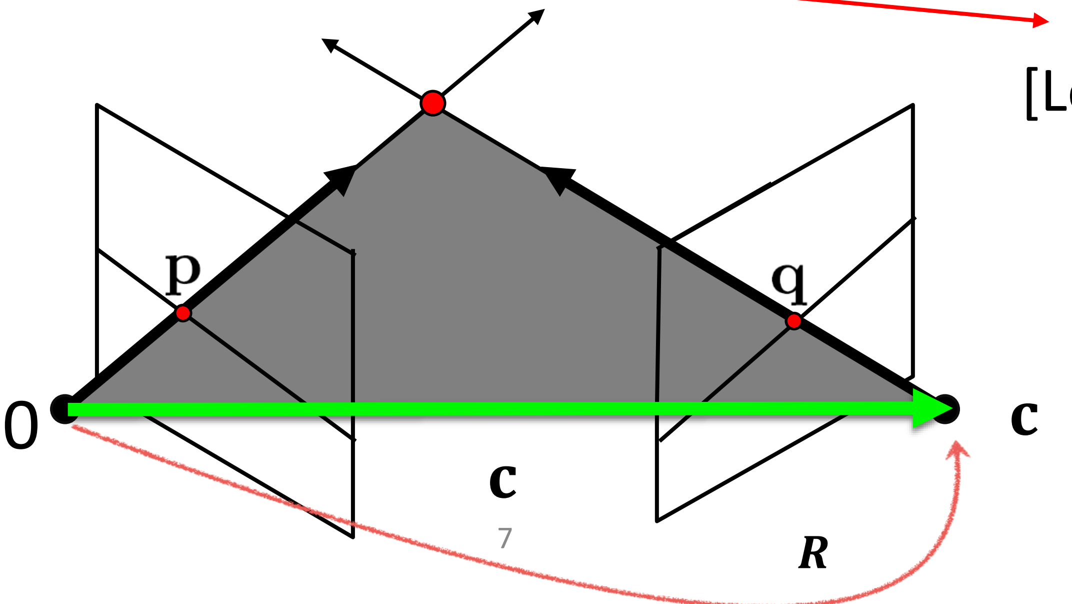
Essential Matrix

[Longuet-Higgins 1981]

$$\mathbf{E} = \mathbf{R}[\mathbf{c}]_{\times} = \mathbf{R}[-\mathbf{R}^{\mathrm{T}}\mathbf{t}]_{\times}$$
$$= \mathbf{R}[\mathbf{R}^{\mathrm{T}}\mathbf{t}]_{\times} = [\mathbf{t}]_{\times}\mathbf{R}$$

For any vector $\mathbf{t}$ *and non-singular matrix* $\mathtt{M}$ *one has*

$$[\mathbf{t}]_{\times}\mathtt{M} = \mathtt{M}^{*}[\mathtt{M}^{-1}\mathbf{t}]_{\times} = \mathtt{M}^{-\mathsf{T}}[\mathtt{M}^{-1}\mathbf{t}]_{\times} \text{ (up to scale).}$$



Fundamental Matrix – uncalibrated case

- How do we generalize the Essential matrix to uncalibrated cameras?
- The way to compute direction from pixel coordinates (see page 24):

$$y = K^{-1}x$$

- We can substitute $\mathbf{p} = \mathbf{K}_1^{-1}\widehat{\mathbf{p}}$ and $\mathbf{q} = \mathbf{K}_2^{-1}\widehat{\mathbf{q}}$ into $\mathbf{q}^{\mathbf{T}}\mathbf{E}\mathbf{p} = \mathbf{0}$
  - Where $\widehat{\mathbf{p}}, \widehat{\mathbf{q}}$ are pixel coordinates
  - Therefore,

$$\widehat{\mathbf{q}}^{\mathrm{T}}\mathbf{K}_2^{-\mathrm{T}}\mathbf{E}\mathbf{K}_1^{-1}\widehat{\mathbf{p}} = 0$$

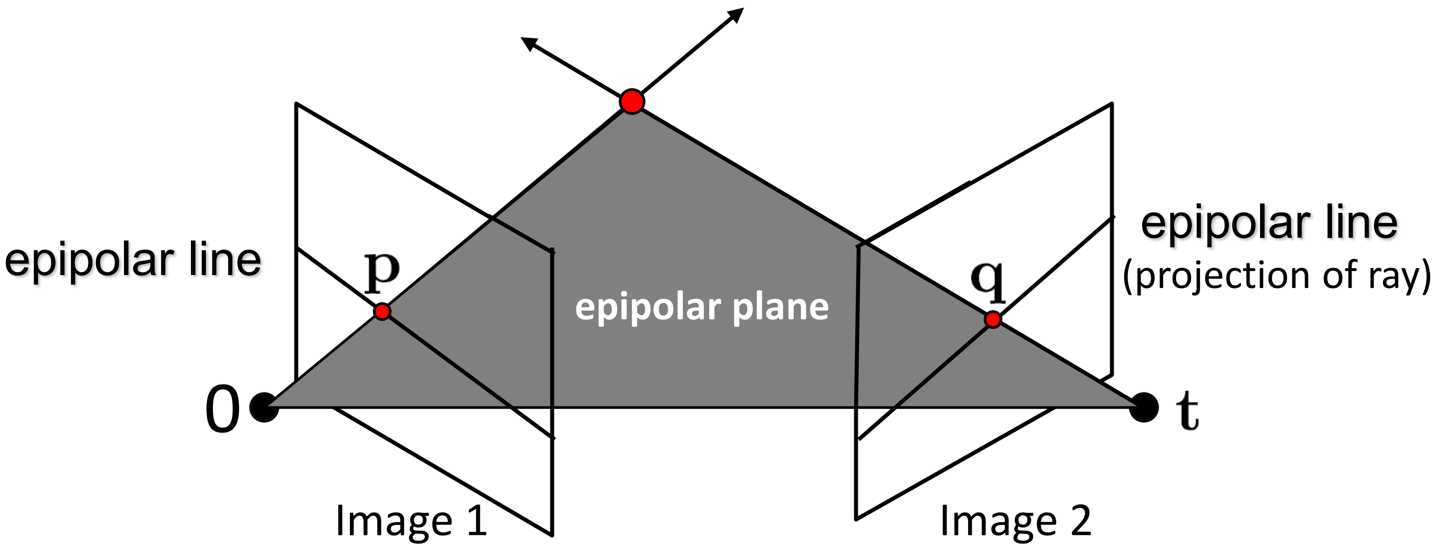
➔

$$\widehat{\mathbf{q}}^{\mathrm{T}}\left(\mathbf{K}_2^{-\mathrm{T}}\mathbf{E}\mathbf{K}_1^{-1}\right)\widehat{\mathbf{p}} = \widehat{\mathbf{q}}^{\mathrm{T}}\mathbf{F}\widehat{\mathbf{p}} = 0$$

Fundamental Matrix
[Oliver Faugeras 1992]

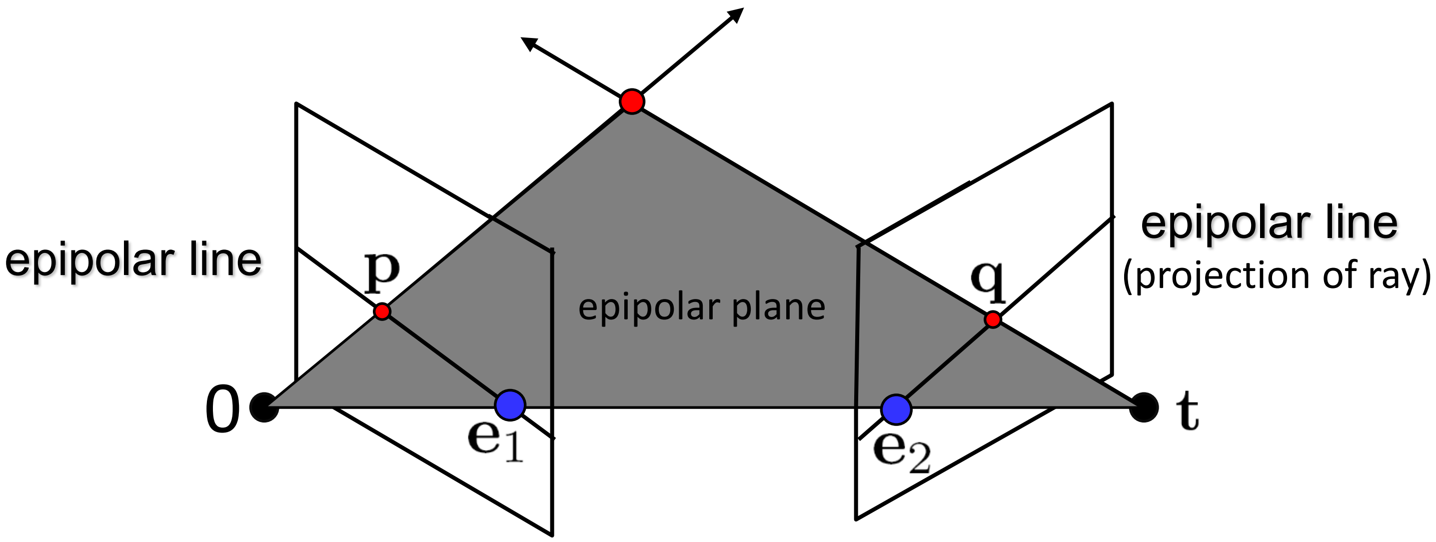
In the following, we abuse the symbols to use $\mathbf{p}, \mathbf{q}$ instead of $\widehat{\mathbf{p}}, \widehat{\mathbf{q}}$ to denote pixel coordinates

Fundamental Matrix – summary



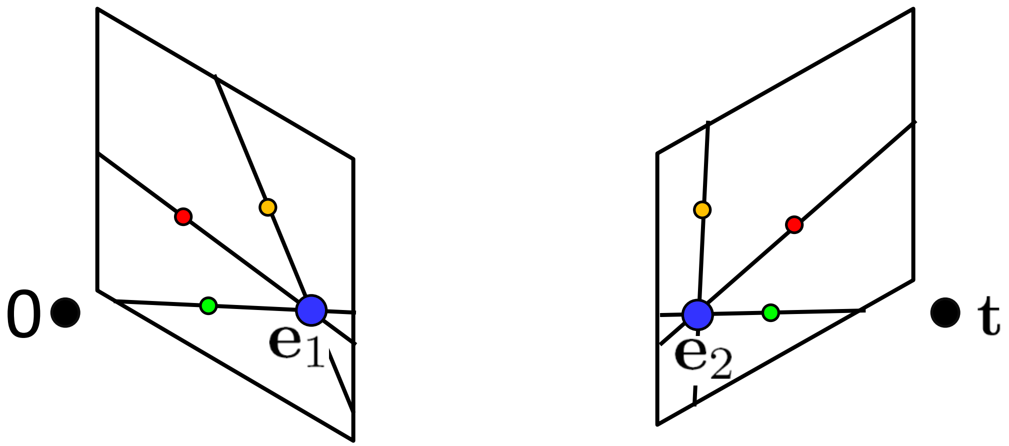
- This *epipolar geometry* of two views is described by a Special 3x3 matrix $\mathbf{F}$, called the *fundamental matrix*
- $\mathbf{F}$ maps (homogeneous) *points* in image 1 to *lines* in image 2!
- The epipolar line (in image 2) of point $\mathbf{p}$ is: $\mathbf{Fp}$
- *Epipolar constraint* on corresponding points: $\mathbf{q}^{\mathrm{T}}\mathbf{Fp} = 0$

Fundamental Matrix – summary



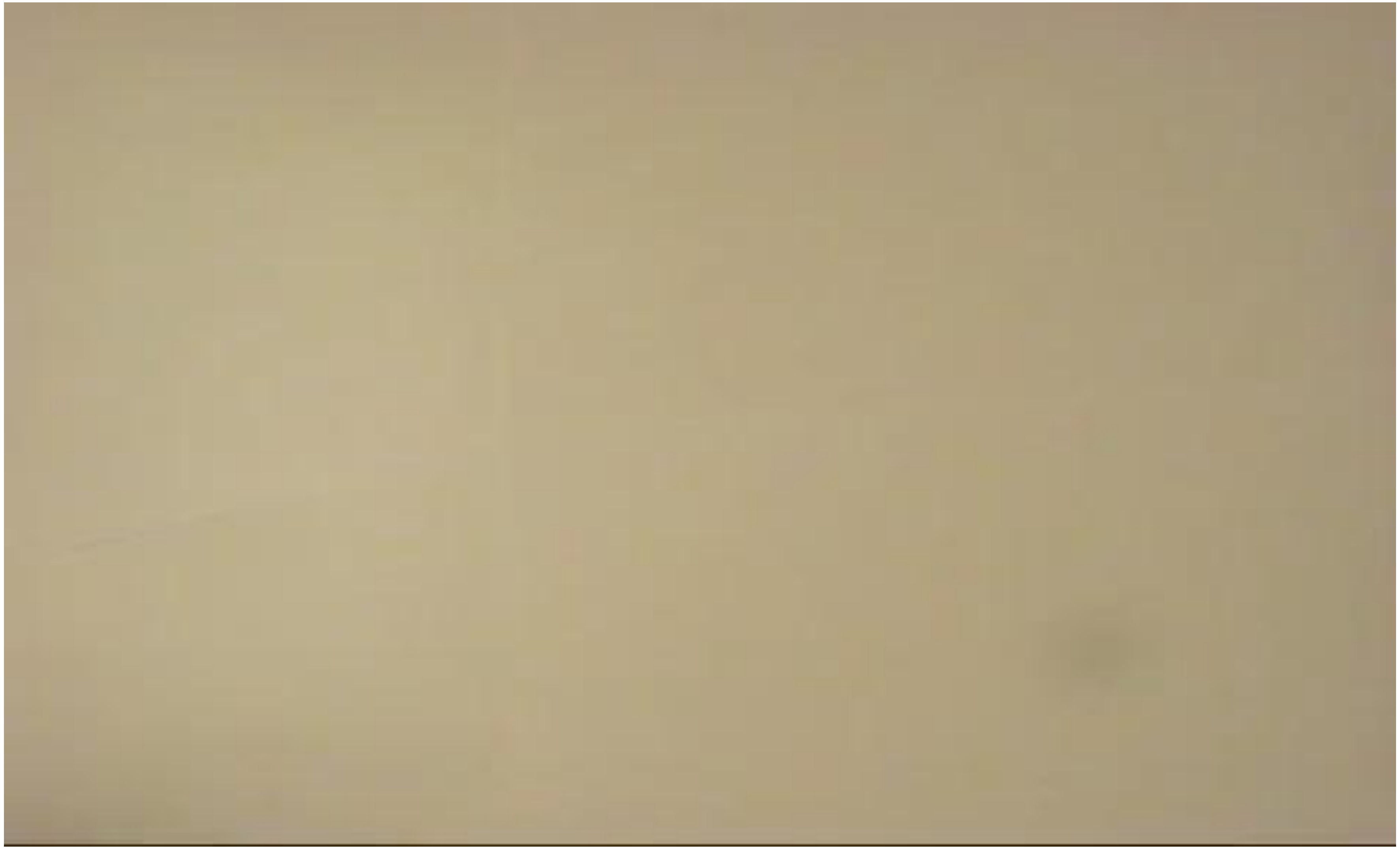
- Two Special points: $\mathbf{e}_1$ and $\mathbf{e}_2$ (the *epipoles*): projection of one camera into the other
- All of the epipolar lines in an image pass through the epipole
- Epipoles can be computed from $\mathbf{F}$ as well: $\mathbf{e}_2^{\mathrm{T}}\mathbf{F} = 0$ and $\mathbf{F}\mathbf{e}_1 = 0$
  - For any pixel $\mathbf{p}$, $\mathbf{Fp}$ is its epipolar line, which must pass through $\mathbf{e}_2$
  - Therefore, $\mathbf{e}_2^{\mathrm{T}}\mathbf{F}\mathbf{p} = 0$ for any $\mathbf{p}$ ➔ $\mathbf{e}_2^{\mathrm{T}}\mathbf{F} = 0$
  - So, $\mathbf{F}$ is rank 2

Fundamental Matrix – summary



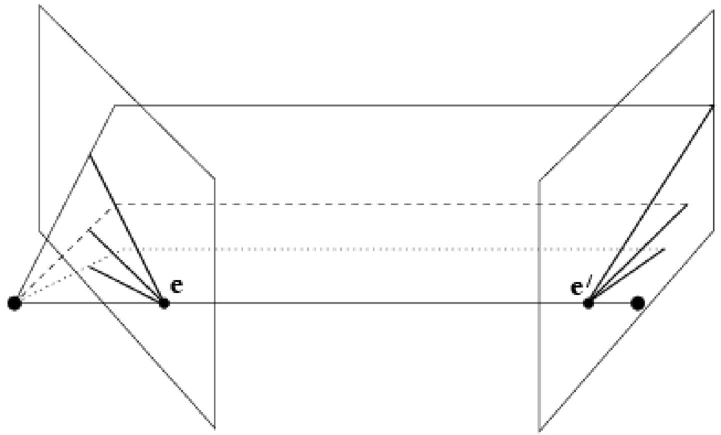
- Two Special points: $\mathbf{e}_1$ and $\mathbf{e}_2$ (the *epipoles*): projection of one camera into the other
- All of the epipolar lines in an image pass through the epipole
- Epipoles can be computed from $\mathbf{F}$ as well: $\mathbf{e}_2^{\mathbf{T}}\mathbf{F} = 0$ and $\mathbf{F}\mathbf{e}_1 = 0$
  - For any pixel $\mathbf{p}$, $\mathbf{F}\mathbf{p}$ is its epipolar line, which must pass through $\mathbf{e}_2$
  - Therefore, $\mathbf{e}_2^{\mathbf{T}}\mathbf{F}\mathbf{p} = 0$ for any $\boldsymbol{p}$ ➔ $\mathbf{e}_2^{\mathbf{T}}\mathbf{F} = 0$
  - So, $\mathbf{F}$ is rank 2

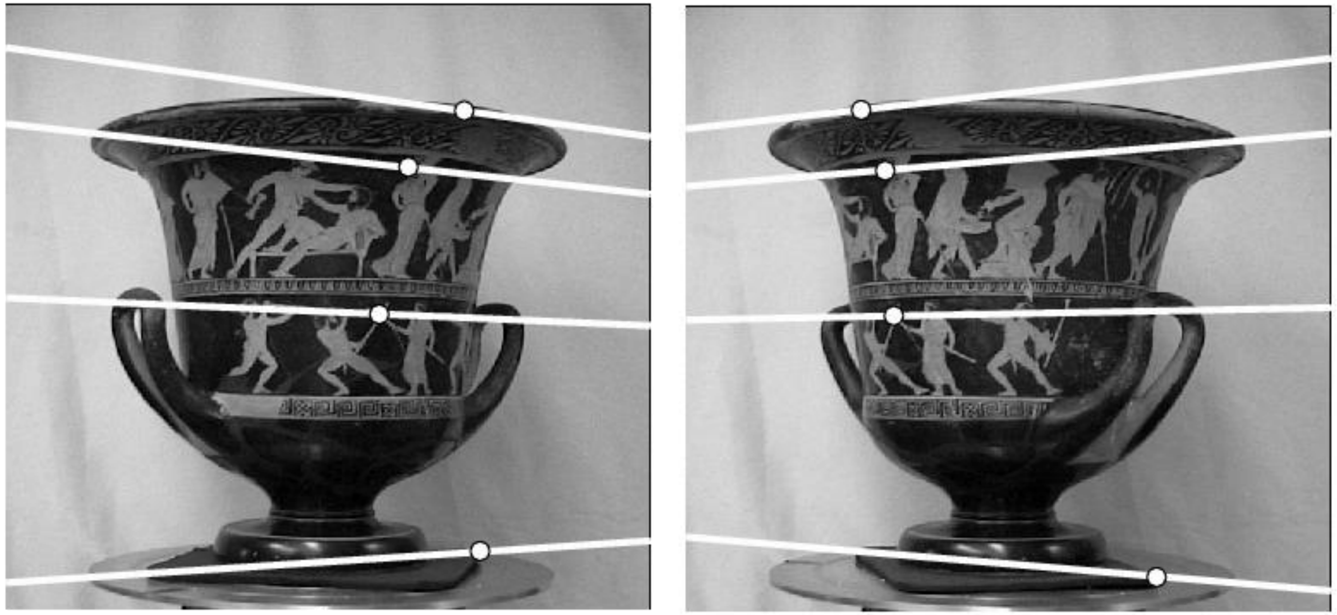
The Fundamental Matrix Song



Questions?

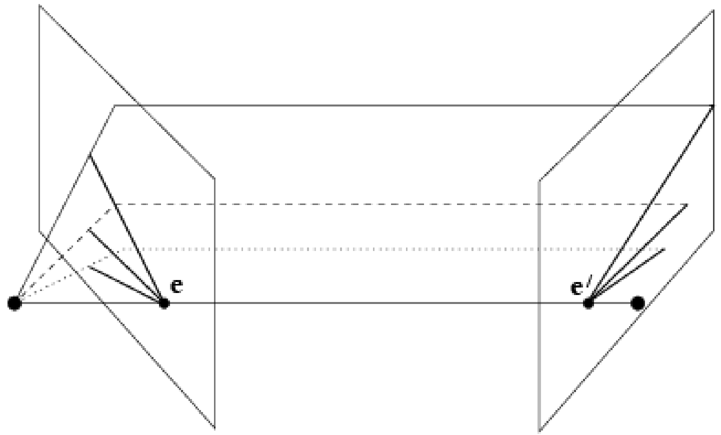
Example: converging cameras

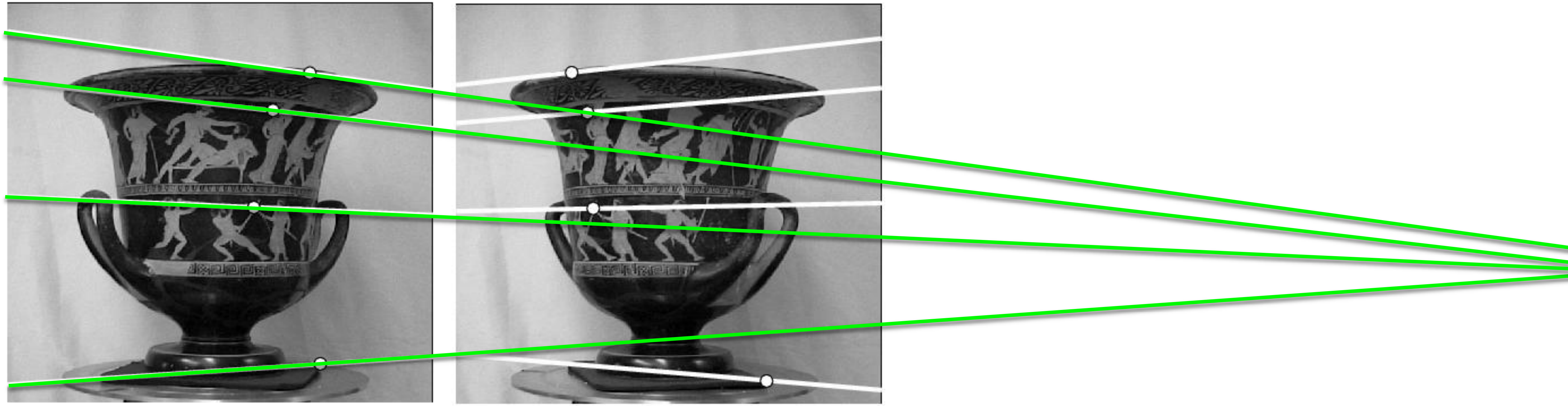




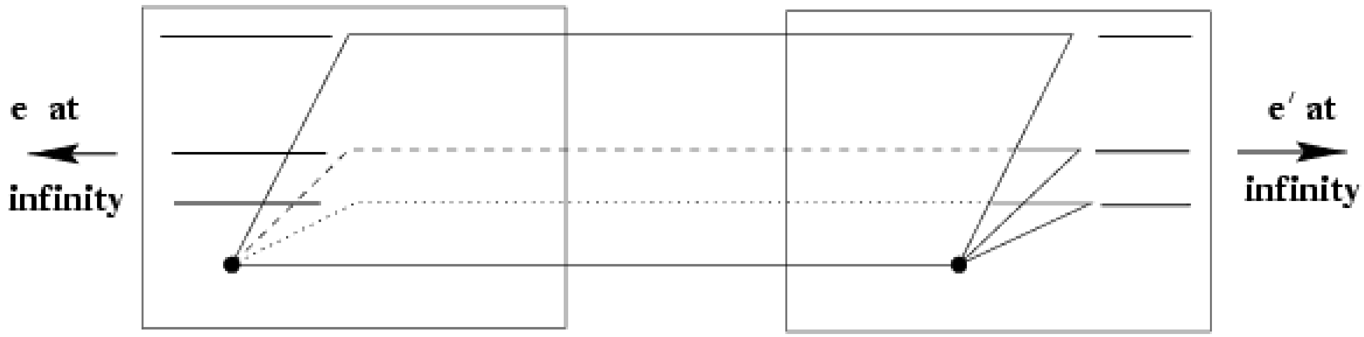
Where is the epipole in this image?

Example: converging cameras

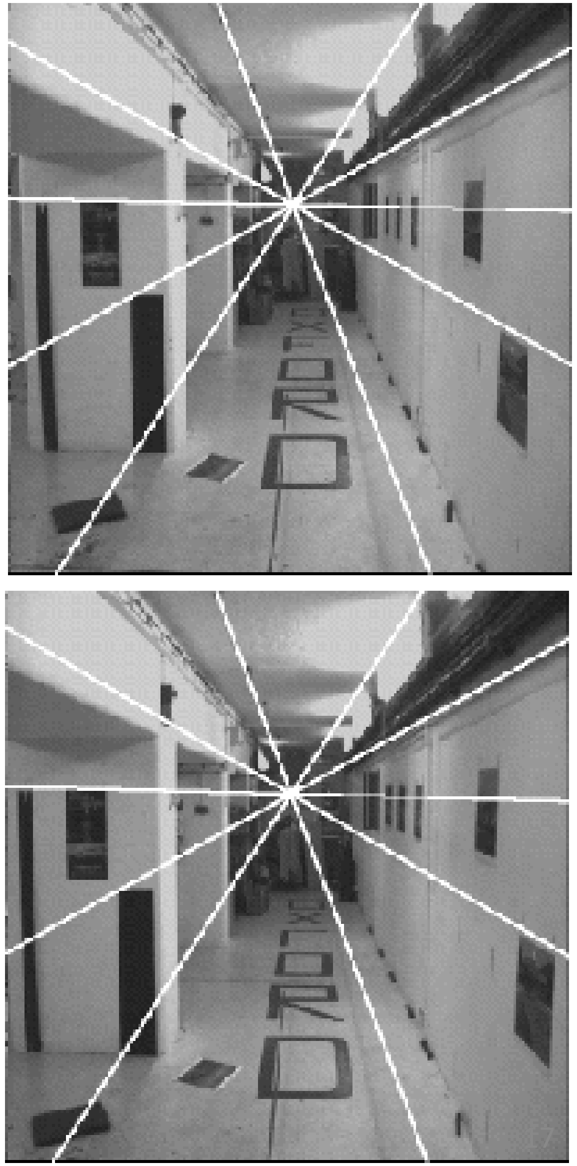


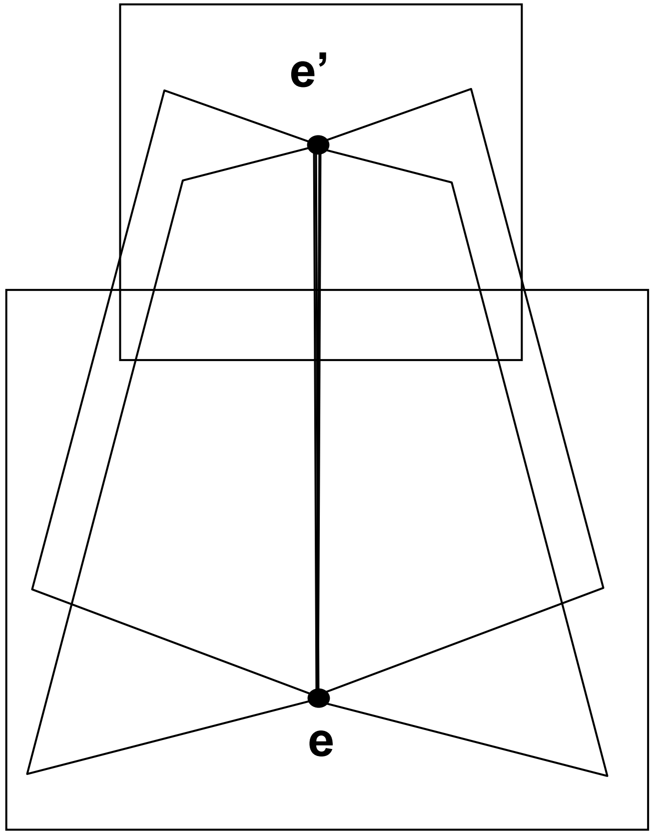


Example: motion parallel with image plane



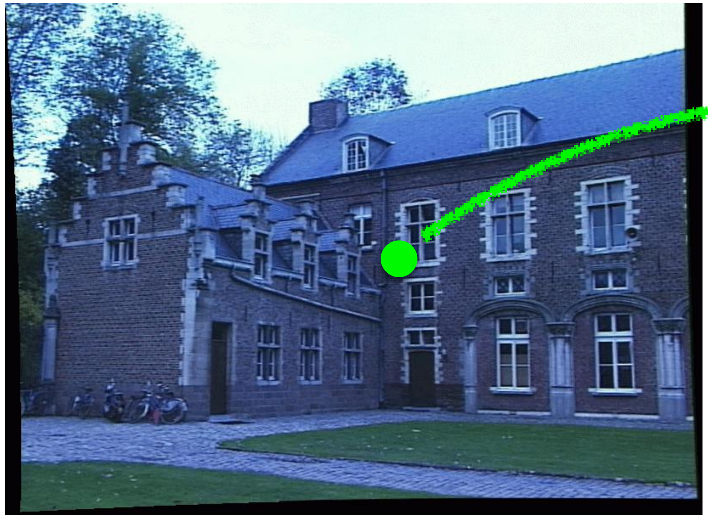
Example: forward motion



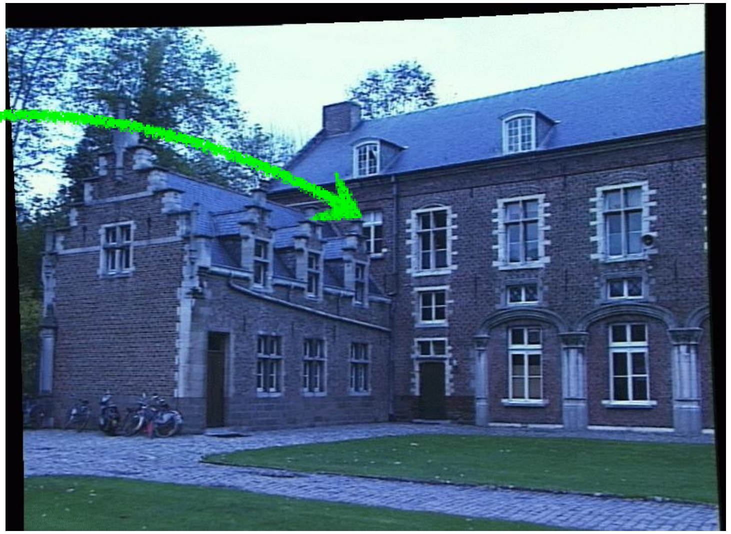


the epipolar constraint for stereo vision

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



Left image



Right image

Epipolar constraint reduces search to a **single** line

Questions?

Estimating the Fundamental Matrix

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

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0$$

separate known from unknown

$$[x'x, x'y, x', y'x, y'y, y', x, y, 1][f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33}]^{\mathrm{T}} = 0$$

(data) (unknowns)

linear equation

$$\begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_n x_n & x'_n y_n & x'_n & y'_n x_n & y'_n y_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = 0$$

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

The Singularity Constraint

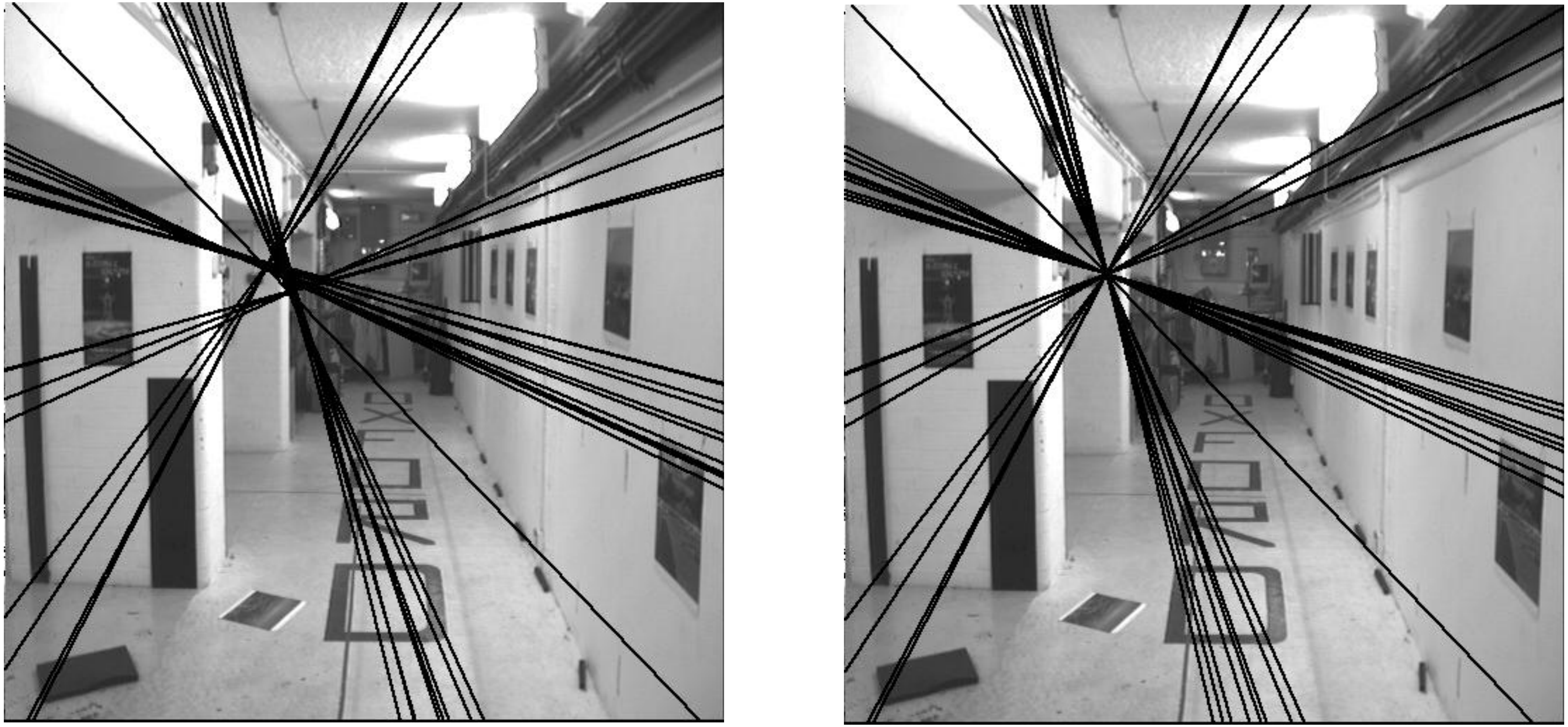
$$e'^{T} F = 0 \qquad Fe = 0 \quad \det(F) = 0 \quad \text{rank } F = 2$$

SVD from linearly computed F matrix (rank 3)

$$F = U\begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{bmatrix}V^T = U_1\sigma_1V_1^T + U_2\sigma_2V_2^T + U_3\sigma_3V_3^T$$

Compute closest rank-2 approximation $\min \|F - F'\|_F$

$$F' = U\begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & 0 \end{bmatrix}V^T = U_1\sigma_1V_1^T + U_2\sigma_2V_2^T$$



when F is non-singular, epipolar lines won‘t intersect at the same point

the NOT normalized 8-point algorithm

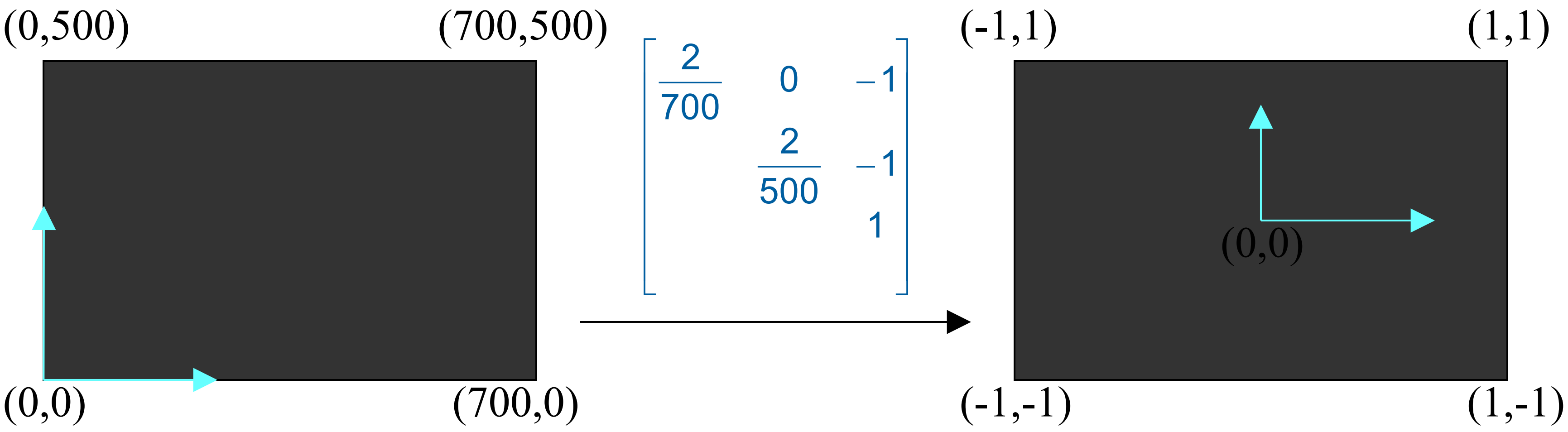
$$\begin{bmatrix} x_1x_1' & y_1x_1' & x_1' & x_1y_1' & y_1y_1' & y_1' & x_1 & y_1 & 1 \\ x_2x_2' & y_2x_2' & x_2' & x_2y_2' & y_2y_2' & y_2' & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_nx_n' & y_nx_n' & x_n' & x_ny_n' & y_ny_n' & y_n' & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

~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

Transform image to ~ [-1,1]x[-1,1]



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

Least squares yields good results (Hartley, PAMI´97)

In Defence of the 8-point Algorithm

Richard I. Hartley,

7-point Algorithm – the minimum case

$$\begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_7 x_7 & x'_7 y_7 & x'_7 & y'_7 x_7 & y'_7 y_7 & y'_7 & x_7 & y_7 & 1 \end{bmatrix} \mathrm{f} = 0$$

7 equations, 9 unknowns

$$\mathrm{A} = \mathrm{U}_{7x7} \mathrm{diag}(\sigma_1, \ldots, \sigma_7, 0, 0) \mathrm{V}_{9x9}{}^{\mathrm{T}}$$

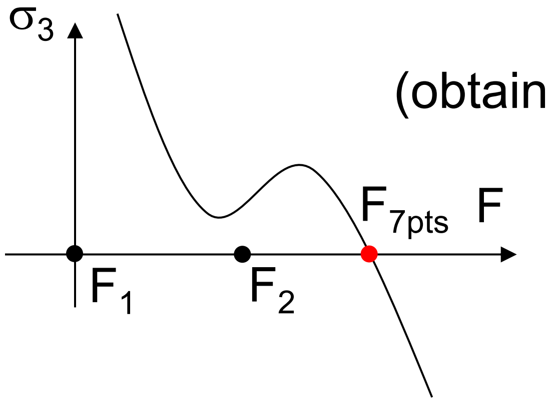
$$\Rightarrow \mathrm{A}[\mathrm{V}_8 \mathrm{V}_9] = 0_{9x2} \qquad \Rightarrow \mathrm{A}(\mathrm{V}_8 + \lambda \mathrm{V}_9) = 0_{9x2}$$

$$\mathrm{x}_i{}^{\mathrm{T}} (\mathrm{F}_1 + \lambda \mathrm{F}_2) \mathrm{x}_i = 0, \forall i = 1 \ldots 7$$

one parameter family of solutions

but $F_1 + \lambda F_2$ not automatically rank 2
so we can solve λ by letting the rank equal to 2

7-point Algorithm – the minimum case



(obtain 1 or 3 solutions)

$$|A| = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a\begin{vmatrix} e & f \\ h & i \end{vmatrix} - b\begin{vmatrix} d & f \\ g & i \end{vmatrix} + c\begin{vmatrix} d & e \\ g & h \end{vmatrix}$$
$$= aei + bfg + cdh - ceg - bdi - afh$$

$$\det(\mathrm{F}_1 + \lambda \mathrm{F}_2) = a_3\lambda^3 + a_2\lambda^2 + a_1\lambda + a_0 = 0$$ (cubic equation)

$$\det(\mathrm{F}_1 + \lambda \mathrm{F}_2) = \det \mathrm{F}_2 \det(\mathrm{F}_2^{-1}\mathrm{F}_1 + \lambda \mathrm{I}) = 0$$

Compute possible λ as eigenvalues of $\mathrm{F}_2^{-1}\mathrm{F}_1$ (only real solutions are potential solutions)

Three solutions when the points and camera center are on a 'critical surface'.

Error Functions

The 8-point and 7-point algorithm minimizes an algebraic error

We can define a symmetric geometric error as:

$$\sum_i d(\mathrm{x}'_i, \mathrm{F}\mathrm{x}_i)^2 + d(\mathrm{x}_i, \mathrm{F}^\mathrm{T}\mathrm{x}'_i)^2$$

Minimizing the distance between the corresponding point and epipolar line

The best objective function is the re-projection error (= Maximum Likelihood Estimation for Gaussian noise)

$$\sum_i d(\mathrm{x}_i, \hat{\mathrm{x}}_i)^2 + d(\mathrm{x}'_i, \hat{\mathrm{x}}'_i)^2 \quad \text{subject to} \quad \hat{\mathrm{x}}'^\mathrm{T}\,\mathrm{F}\hat{\mathrm{x}} = 0$$

Recommendations:

1. Do not use unnormalized algorithms
2. Quick and easy to implement: 8-point normalized
3. Better: enforce rank-2 constraint during minimization
4. Best: Maximum Likelihood Estimation by minimizing re-projection error

Degenerate cases:

- Degenerate cases (only a homography can be estimated)
  - Planar scene
  - Pure rotation

Questions?

Computation of the Essential matrix

$$x'^{T} F x = 0$$

Compute $\mathbf{F}$ first, then simply take: $\mathbf{E} = \mathbf{K}_1^{T}\mathbf{F}\mathbf{K}_2$

$$y'^{T} E y = 0 \qquad y_i = \mathbf{K}^{-1} x_i$$

Or, take 8-point algorithm to solve $\mathbf{E}$, then enforce:

$$E = U \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{bmatrix} V^{T} \Rightarrow E = U \begin{bmatrix} \frac{\sigma_1+\sigma_2}{2} & & \\ & \frac{\sigma_1+\sigma_2}{2} & \\ & & 0 \end{bmatrix} V^{T}$$

Computation of the Essential matrix

- $\mathbf{E}$ has less degrees of freedom than $\mathbf{F}$
- In principal, 5 pair of corresponding points are sufficient to decide $\mathbf{E}$.
  - The 5-point algorithm by David Nister

An Efficient Solution to the Five-Point Relative Pose Problem

David Nistér
Sarnoff Corporation
CN5300, Princeton, NJ 08530
dnister@sarnoff.com

PAMI 2004

Getting Camera Matrices from E

For a given $\mathbf{E} = \mathbf{U}\, diag\,(1,1,0)\,\mathbf{V}^T$(by SVD decomposition),
and the first camera matrix $\mathbf{P} = [\mathbf{I}\,|0]$,
there are 4 choices for the second camera matrix $\mathbf{P}'$, namely

$$P' = [UWV^T \mid \mathbf{u}_3] \quad \text{or} \quad P' = [UWV^T \mid -\mathbf{u}_3]$$

$$\text{or} \qquad P' = [UW^TV^T \mid \mathbf{u}_3] \quad \text{or} \quad P' = [UW^TV^T \mid -\mathbf{u}_3]$$

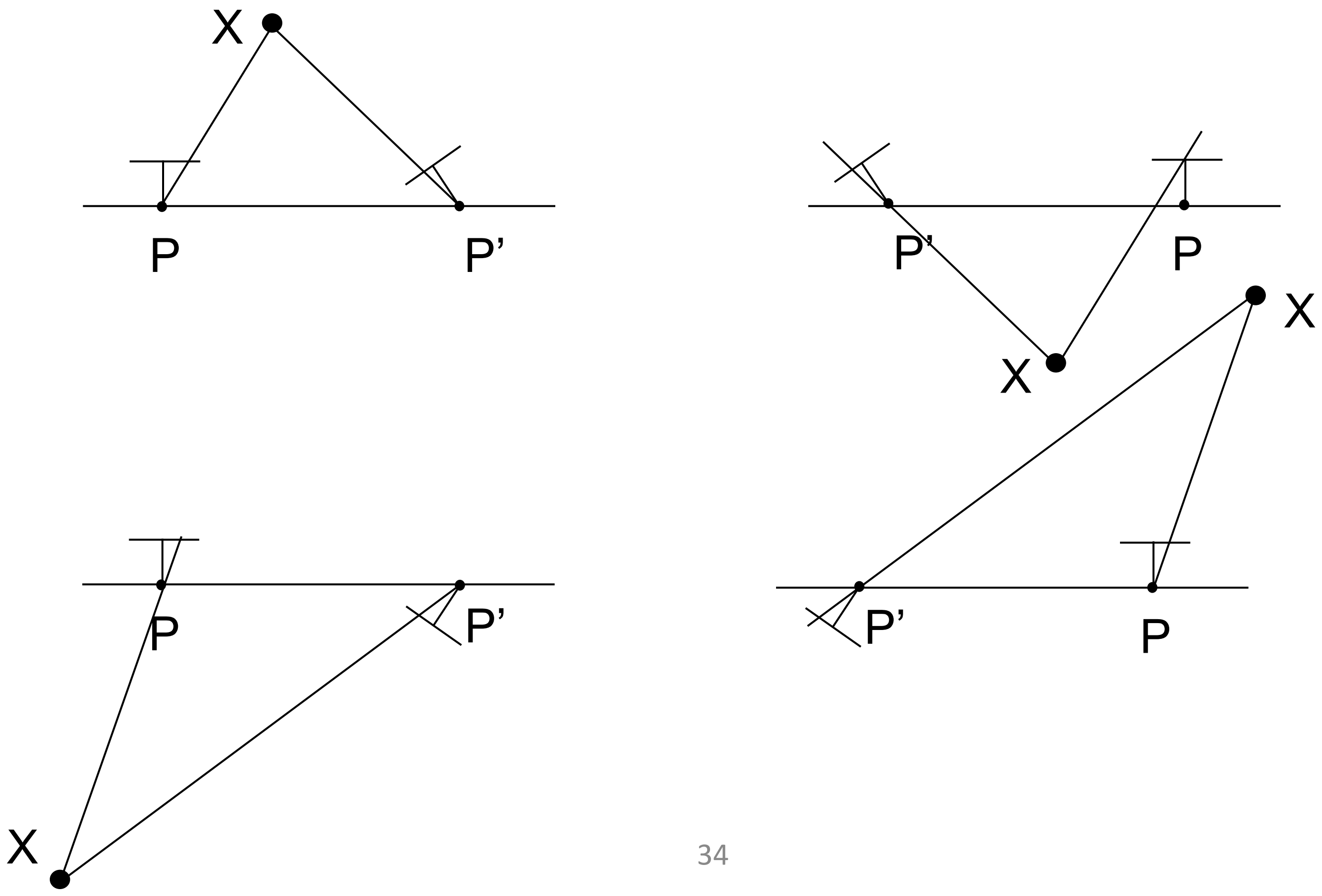
$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$\mathbf{u}_3$ is the last column of $\boldsymbol{U}$

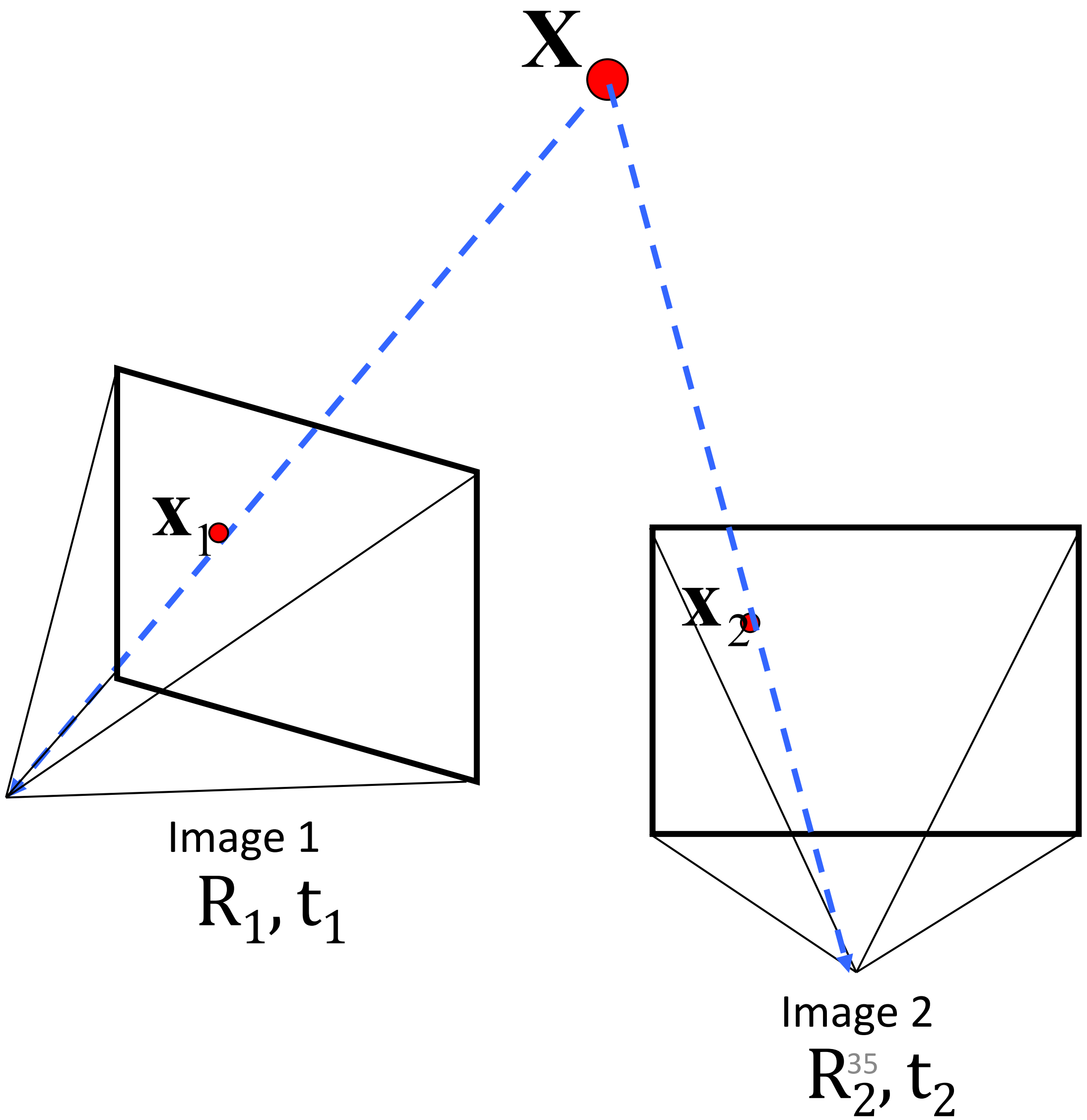
For the proof, please refer to section 9.6 of the ‘multiview geometry’ book

Selecting from the Four Solutions

- Among these four configurations, only one is valid where the reconstructed points are in front of both cameras



Triangulation (to be studied soon)

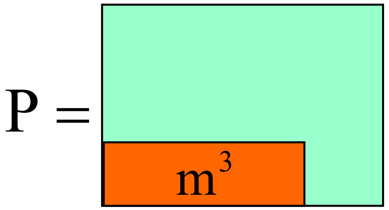


In front of the camera?

- A point X
- Direction from camera center to point $X - C$
- The direction of principal axis m^3
- Compute the angle between $(X - C)$ and m^3

$$C = \begin{pmatrix} -\mathrm{M}^{-1}\mathrm{p}_4 \\ 1 \end{pmatrix}$$

- Just need to test $(X - C) \cdot m^3 > 0$



Pick the Solution

- With maximal number of points in front of both cameras.

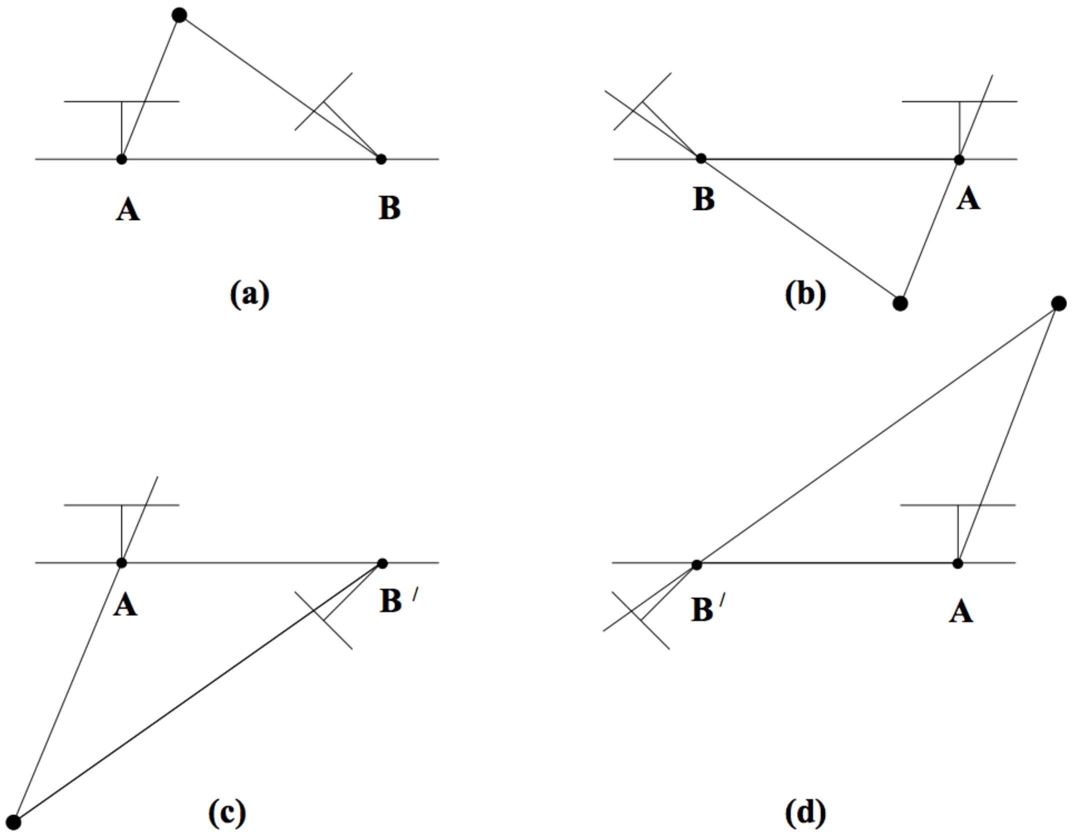


Fig. 9.12. **The four possible solutions for calibrated reconstruction from E.** *Between the left and right sides there is a baseline reversal. Between the top and bottom rows camera B rotates* 180° *about the baseline. Note, only in (a) is the reconstructed point in front of both cameras.*

Page 260 of the Multiple View Geometry in Computer Vision, 2^{nd} Ed

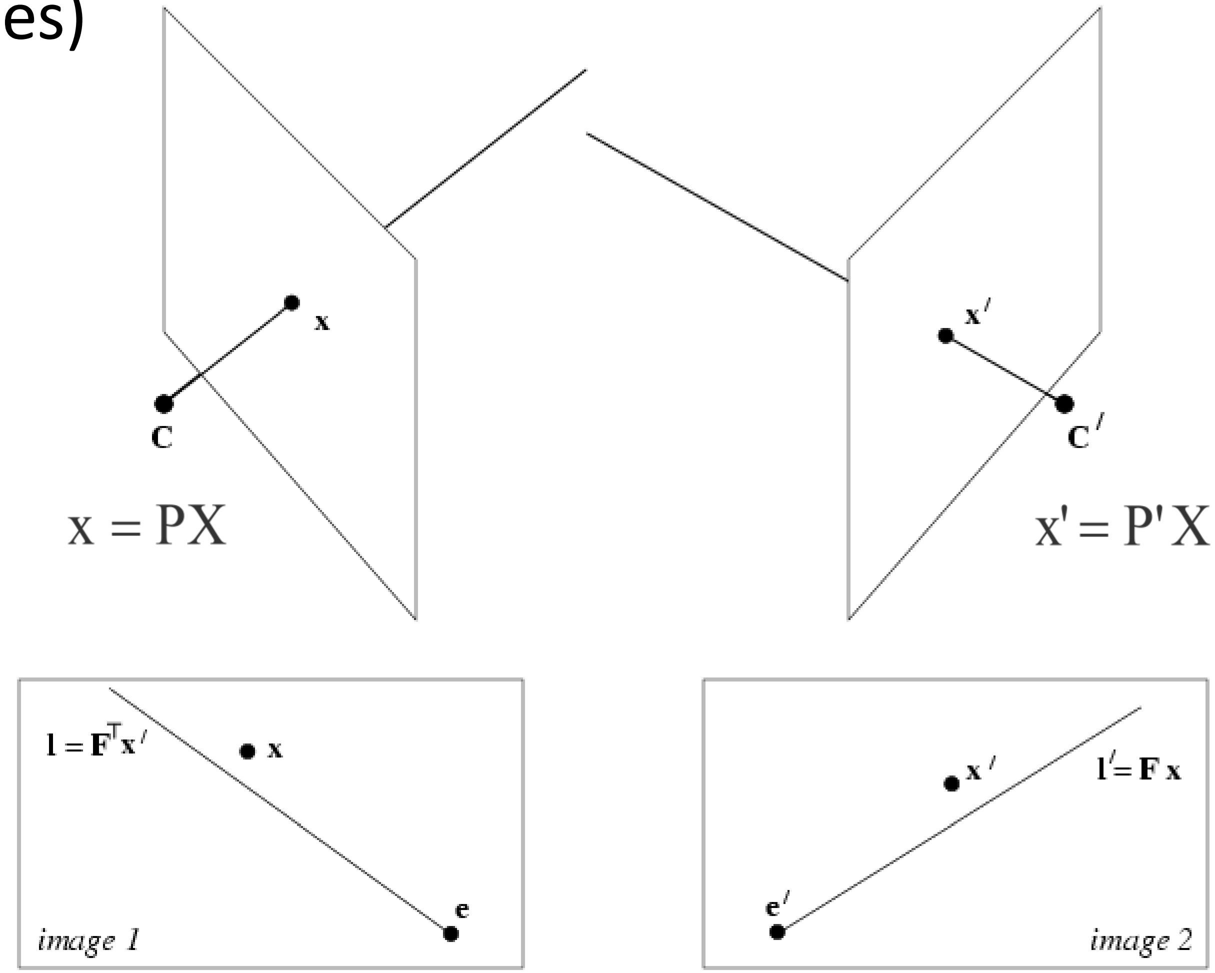
Questions?

Outline

- Two-view Geometry
- Triangulation

Point Reconstruction

- Estimate 3D point X from known cameras P, P' (and feature correspondences)



Direct Linear Transform

$$x = PX \qquad \Rightarrow x \times PX = 0 \qquad\qquad x' = P'X \qquad \Rightarrow x' \times P'X = 0$$

$$AX = 0 \qquad A = \begin{bmatrix} x p^{3T} - p^{1T} \\ y p^{3T} - p^{2T} \\ x' p'^{3T} - p'^{1T} \\ y' p'^{3T} - p'^{2T} \end{bmatrix} \qquad P = \begin{bmatrix} {p^1}^T \\ {p^2}^T \\ {p^3}^T \end{bmatrix}$$

Homogeneous coordinate, add constraint $\|X\| = 1$

Convert to inhomogeneous coordinate $(X, Y, Z, 1)$

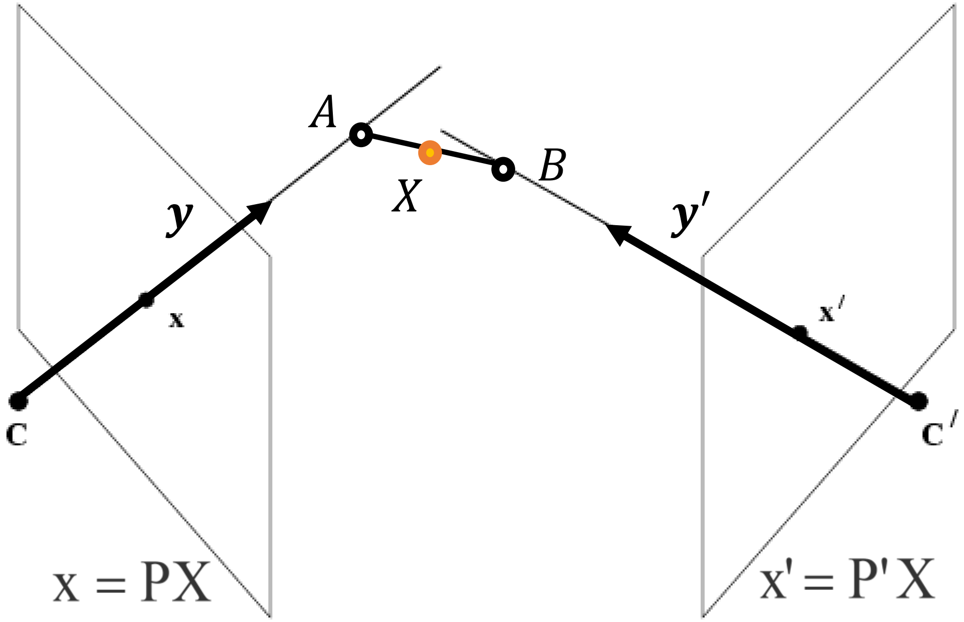
This method minimizes an algebraic error

Mid-point Algorithm

- Find the middle point of the mutual perpendicular line segment $\boldsymbol{AB}$

$$\boldsymbol{A} = \mathbf{c} + d_1\boldsymbol{y} \qquad \boldsymbol{B} = \mathbf{c}' + d_2\boldsymbol{y}'$$

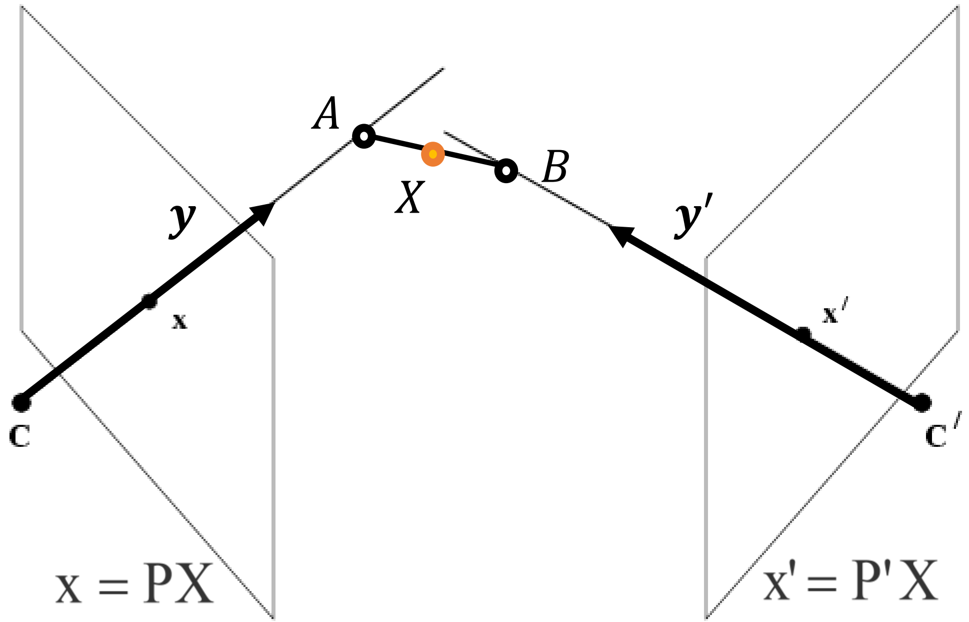
- $\mathbf{c}, \mathbf{c}', \mathbf{y}, \mathbf{y}'$ are all in the same coordinate system (e.g. camera frame 1)



$$\boldsymbol{y} = \boldsymbol{K}_1^{-1}\boldsymbol{x}$$
$$\boldsymbol{y}' = \boldsymbol{R}^T\boldsymbol{K}_2^{-1}\boldsymbol{x}'$$

Mid-point Algorithm

- Choose the first camera's coordinate as a reference
  - $\mathbf{c} = 0, \mathbf{P} = \mathbf{K_1}[\mathbf{I} \mid 0]$
- Put the second camera in that coordinate system
  - Assume known relative rotation $\mathbf{R}$, translation $\mathbf{t}$
  - $\mathbf{P}' = \mathbf{K}_2[\mathbf{R}|\mathbf{t}], \mathbf{c}' = -\mathbf{R}^{\mathrm{T}}\mathbf{t}$
  - $\mathbf{y}' = \mathbf{R}^{\mathrm{T}}\mathbf{K}_2^{-1}\mathbf{x}'$



Mid-point Algorithm

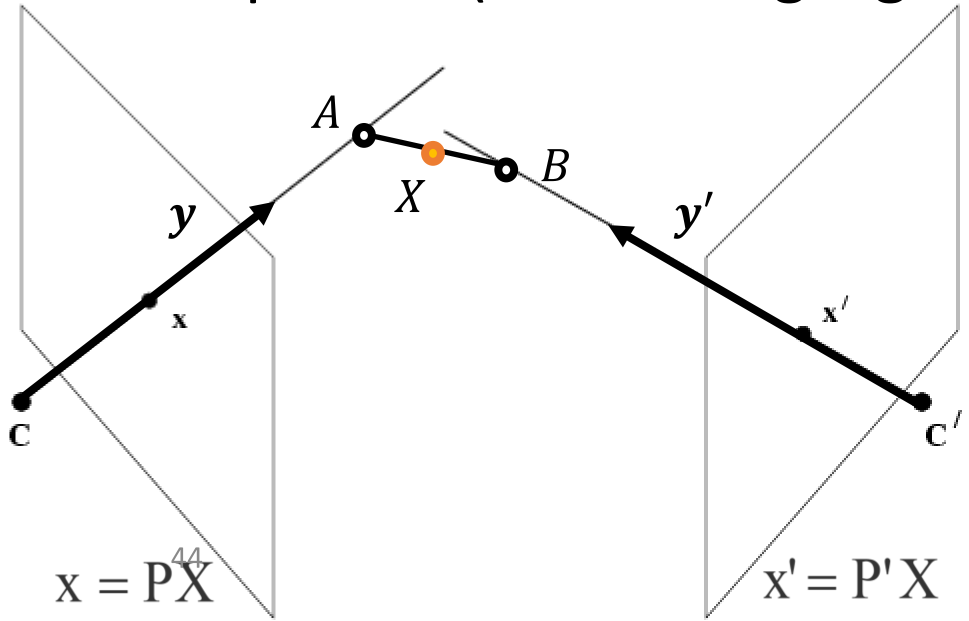
- Since $\boldsymbol{AB}$ is the mutual perpendicular line segment
  - $\boldsymbol{AB} \perp \boldsymbol{y},\ \boldsymbol{AB} \perp \boldsymbol{y}'$
- This means:

$$(\boldsymbol{A} - \boldsymbol{B}) \times (\boldsymbol{y} \times \boldsymbol{y}') = \boldsymbol{0}$$

- This generates three equations of d_1, d_2

$$\boldsymbol{A} = \mathbf{c} + d_1\boldsymbol{y}$$
$$\boldsymbol{B} = \mathbf{c}' + d_2\boldsymbol{y}'$$

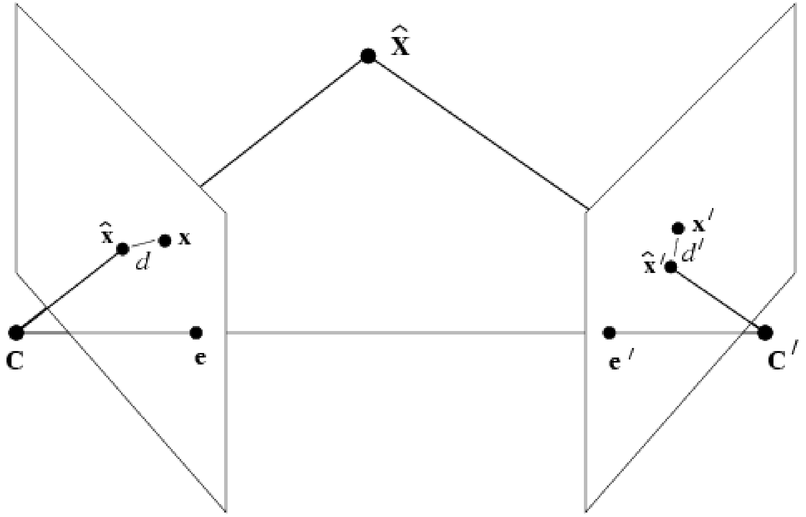
- Solve d_1, d_2 from the above linear equation (minimizing a geometric error)



Reprojection Error

$$d(\mathbf{x},\hat{\mathbf{x}})^2 + d(\mathbf{x}',\hat{\mathbf{x}}')^2 \ \text{ subject to } \ \hat{\mathbf{x}}'^{\mathrm{T}}\mathbf{E}\hat{\mathbf{x}} = 0 \ \ (\text{or } \hat{\mathbf{x}}'^{\mathrm{T}}\mathbf{F}\hat{\mathbf{x}} = 0)$$

or equivalently subject to $\hat{\mathbf{x}} = \mathbf{P}\hat{\mathbf{X}}$ and $\hat{\mathbf{x}}' = \mathbf{P}'\hat{\mathbf{X}}$



compute using the Levenberg-Marquardt algorithm

This triangulation works for uncalibrated cameras

- The algebraic error and mid-point algorithm needs $\mathbf{K}, \mathbf{K}'$ (pre-calibrated cameras)

Questions?

Algorithms Studied Today

- Epipolar geometry (relative motion estimation)
  - 2D <-> 2D correspondences, compute relative camera motion (up to a scale)
- Triangulation
  - 2D <-> 2D correspondences (and known camera poses), compute 3D point