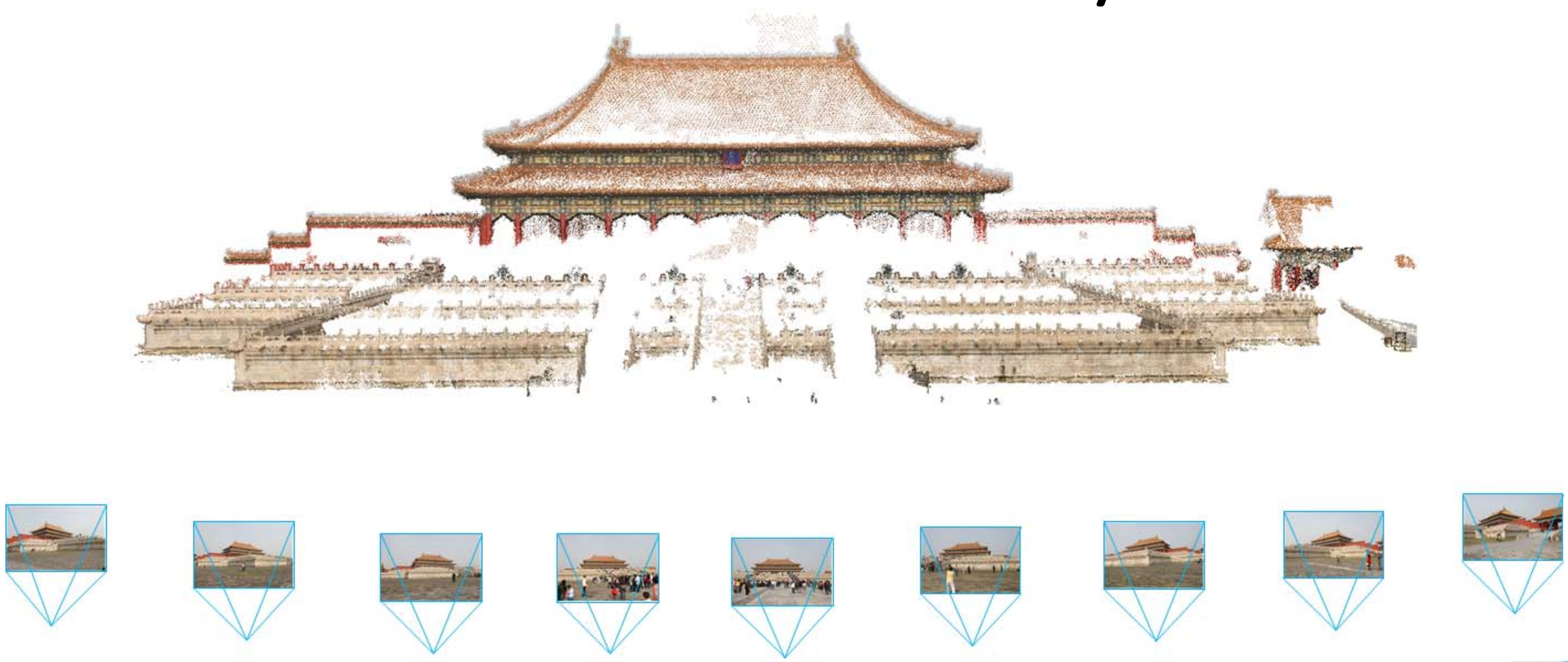
10. Camera Calibration & 2-View Geometry



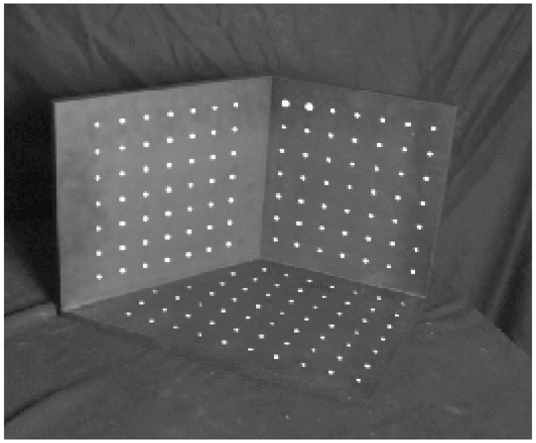
Outline

- Camera Calibration
- Two-view Geometry
- Triangulation

一种简单实现三维重建的方法：假设空间中有一个已知的对象，并且精确地知道这个对象上的点的三维坐标，对图像进行拍摄得到的图片，能清楚地知道每个三维点和图像中的点的对应关系。有了这些信息，就能进行相机标定，得到相机地内参外参。

Camera Calibration

- Place a known object in the scene
  - identify correspondence between image and scene
  - compute mapping from scene to image



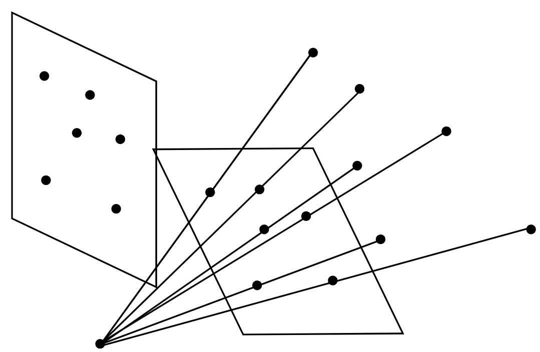
Issues

- must know geometry very accurately
- must know 3D->2D correspondence

Resectioning

Given some 3D points X_i with known coordinates and their image projections x_i , how do we compute the camera matrix P ?

$$\mathrm{X}_i \leftrightarrow \mathrm{x}_i \quad \rightarrow \quad \mathrm{P}\,?$$



warning: in practice, it is difficult to obtain 3D points of known positions

SFU

引出问题：如何获得映射关系矩阵P ?

ZHEJIANG UNIVERSITY

Basic Equations

每张图片距离目标地距离可能不相等，所以要乘以λ（缩放因子）。

The equations we have: $\mathrm{x}_i = \lambda \mathrm{P} \mathrm{X}_i$

Get rid of the λ by cross product $\mathrm{x}_i \times \mathrm{P}\mathrm{X}_i = 0$ 在homography中有相应用法

Change cross product to its matrix form: $[\mathrm{x}_i]_\times \mathrm{P}\mathrm{X}_i = 0$

两个向量叉乘，可以将前一个向量转换为一个矩阵。

Definition: $\vec{a} \times \vec{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}$

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

Rewrite the formula,

we get 3 equations (2 of them are independent) about ***P*** from each pair of $\mathrm{X}_i \leftrightarrow \mathrm{x}_i$

$$\begin{bmatrix} \mathbf{0}^\top & -w_i \mathbf{X}_i{}^\top & y_i \mathbf{X}_i{}^\top \\ w_i \mathbf{X}_i{}^\top & \mathbf{0}^\top & -x_i \mathbf{X}_i{}^\top \\ -y_i \mathbf{X}_i{}^\top & x_i \mathbf{X}_i{}^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = \mathbf{0}$$

Eventually, we obtain a linear equation:

$$\mathrm{Ap} = 0$$

SFU

ZHEJIANG UNIVERSITY

Direct Linear Transform (DLT)

$$\mathrm{Ap} = 0$$

Minimal solution

P has 11 dof, 2 independent eq./points

$\Rightarrow$ 5½ correspondences needed (say 6)

Over-determined solution

$n \geq 6$ points

minimize $\|\mathrm{Ap}\|$ subject to constraint

$$\|\mathrm{p}\| = 1$$

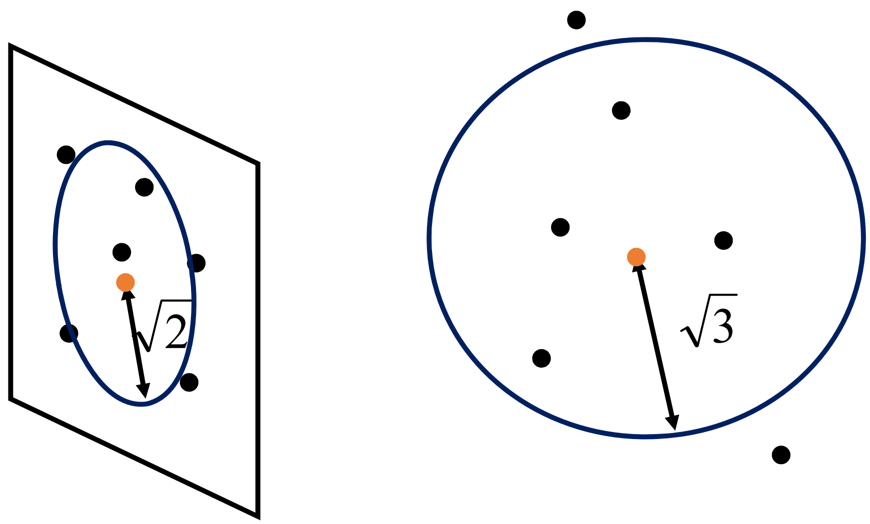
$$\|\mathrm{m}^3\| = 1 \qquad \mathrm{P} = \begin{bmatrix} \quad \\ \mathrm{m}^3 \end{bmatrix}$$

SFU

ZHEJIANG UNIVERSITY

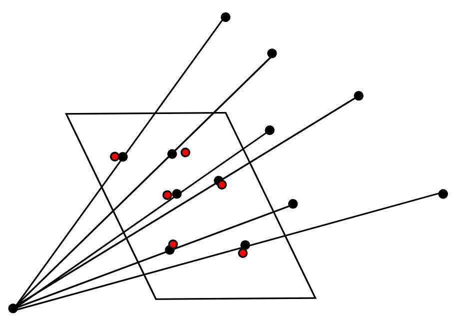
Data Normalization

Simple,
translate and scale the 2D and 3D points
make them to center at origin and with radius of √2 or √3



Geometric Error

- DLT minimizes the algebraic distance
- It is better to minimize the Euclidean distance between the projected positions and 2D points (geometric distance)
  – Assume 3D points are precisely known.
  – Noises only present in image measurements!



$$\sum_i d(\mathbf{x}_i, \hat{\mathbf{x}}_i)^2$$

$$\min_{\mathrm{P}} \sum_i d(\mathbf{x}_i, \mathrm{P}\mathbf{X}_i)^2$$

Geometric Error

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \cong \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

$$u_i = \frac{m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}}$$

$$v_i = \frac{m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}}$$

$$\sum_i \left| u_i - \frac{m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}} \right|^2 + \left| v_i - \frac{m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}} \right|^2$$

$$= \sum_i d^2(x_i, \hat{x}_i)$$

Geometric Error vs Direct Linear Transform

The DLT method can be derived from a different way:

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \cong \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

$$u_i = \frac{m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}}$$

$$v_i = \frac{m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}}$$

$$u_i(m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}) = m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}$$

$$v_i(m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}) = m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}$$

$$\begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_iX_i & -u_iY_i & -u_iZ_i & -u_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_iX_i & -v_iY_i & -v_iZ_i & -v_i \end{bmatrix} \begin{bmatrix} m_{00} \\ m_{01} \\ m_{02} \\ m_{03} \\ m_{10} \\ m_{11} \\ m_{12} \\ m_{13} \\ m_{20} \\ m_{21} \\ m_{22} \\ m_{23} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

SFU

ZHEJIANG UNIVERSITY

Geometric Error vs Direct Linear Transform

Notice the DLT method minimizes a different error

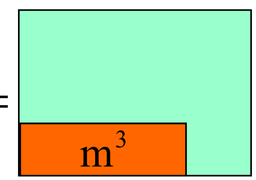
$$\sum_i |u_i(m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}) - m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}|^2$$

$$+ \sum_i |v_i(m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}) - m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}|^2$$

$$= \sum_i \hat{w}_i^2 d^2(x_i, \hat{x}_i)$$

$$\hat{w}_i = ||m^3|| depth(X, P)$$

P =



The DLT minimizes a weighted geometric error, where faraway points have larger weights

Gold Standard Algorithm

<u>Objective</u>

Given n≥6 2D to 2D point correspondences $\{X_i \leftrightarrow x_i'\}$, determine the Maximum Likelyhood Estimation of P (assuming 3D points are precisely known)

<u>Algorithm</u>

(i) Linear solution:

(a) Normalization: $\widetilde{X}_i = UX_i \qquad \widetilde{x}_i = Tx_i$

(b) DLT: $A\widetilde{p} = 0$

(ii) Minimization of geometric error: using the linear estimate as a starting point minimize the geometric error:

$$\min_{\mathrm{P}} \sum_i d(\widetilde{\mathbf{x}}_i, \widetilde{\mathrm{P}}\widetilde{\mathbf{X}}_i)^2$$

(iii) Denormalization: $P = T^{-1}\widetilde{P}U$

Restricted Camera Estimation

Sometimes, we have partial knowledge of the camera matrix

- skew s is zero
- pixels are square $\alpha_x = \alpha_y$
- principal point is known, $x_0 = y_0 = 0$
- complete camera matrix K is known (e.g. focal length known from EXIF)

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

SFU

浙江大學 ZHEJIANG UNIVERSITY

Restricted Camera Estimation

Initialization
- Use general DLT
- Clamp values to desired values, e.g. s=0, $\alpha_x = \alpha_y$

Note: can sometimes cause big jump in error

Alternative initialization
- Use general DLT
- Impose soft constraints
- gradually increase weights

$$\sum_i d(\mathbf{x}_i, \mathrm{P}\mathbf{X}_i)^2 + ws^2 + w(\alpha_x - \alpha_y)^2$$

Minimize the geometric error to refine

SFU

浙江大學 ZHEJIANG UNIVERSITY

Questions?

PnP: Perspective-n-Points

- PnP: determine camera extrinsics when intrinsics are known
- How many pairs of $\mathrm{X}_i \leftrightarrow \mathrm{x}_i$ do we need?
  - Each pair provides 2 constraints
  - There are in total 6 dof (3 for R and 3 for t)
- The PnP algorithm ($n \geq 3$)
  - From n pairs of $\mathrm{X}_i \leftrightarrow \mathrm{x}_i$, first solve 3D positions of x_i in camera frame
  - Then solve R, t from 3D $\leftrightarrow$ 3D correspondences
    - Between camera frame and world frame
    - Solution easy to derive (try it yourself)

P3P Algorithm

- Assume $||c - X_i|| = d_i, ||c - X_j|| = d_j, ||X_i - X_j|| = d_{ij}$

$$d_{ij}^2 = d_i^2 + d_j^2 - 2d_i d_j cos\theta_{ij}$$

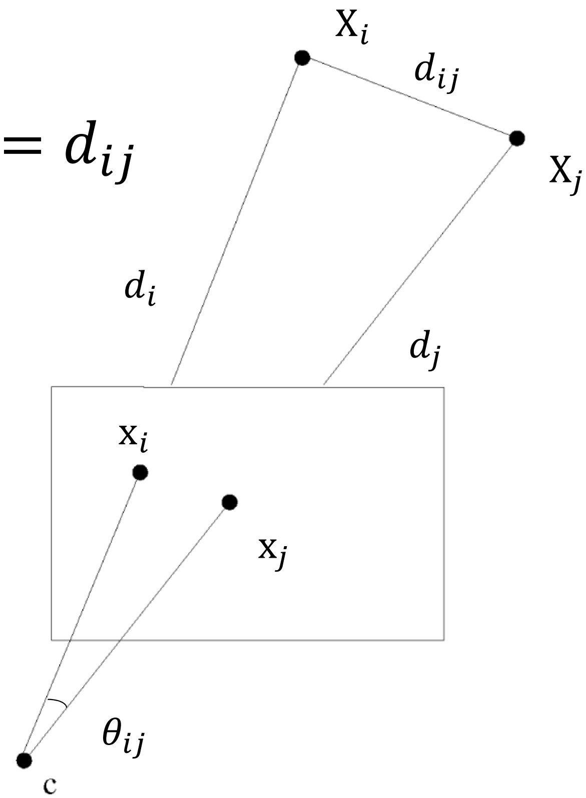
  - The angle θ_{ij} is known from the intrinsics
- Denote $f_{ij}(d_i, d_j) = d_i^2 + d_j^2 - 2d_i d_j cos\theta_{ij} - d_{ij}^2$
- For 3 points, we can solve d_1, d_2, d_3 from

$$\begin{cases} f_{12}(d_1, d_2) = 0 \\ f_{23}(d_2, d_3) = 0 \\ f_{13}(d_1, d_3) = 0 \end{cases}$$

  - After eliminating d_2, d_3, obtain a 4-th order equation of $d_1^2 \triangleq x$

$$g(x) = a_5 x^4 + a_4 x^3 + a_3 x^2 + a_2 x + a_1 = 0$$

  - 4 solutions exist (more points needed for uniqueness)



SFU

浙江大學 ZHEJIANG UNIVERSITY

Linear PnP Algorithm

- For $n \geq 5$, pick any point $\mathrm{X_i}$, and another two points
  - Obtain a 4-th order polynomial of d_i ($d_i^2 \triangleq x$)

$$g(x) = a_5^i x^4 + a_4^i x^3 + a_3^i x^2 + a_2^i x + a_1^i = 0$$

  - There are $(n-1)(n-2)/2$ such polynomials in total
- Stacking all such equations

$$\begin{pmatrix} & & \vdots & & \\ a_5^i & a_4^i & a_3^i & a_2^i & a_1^i \\ & & \vdots & & \end{pmatrix} \begin{pmatrix} x^4 \\ x^3 \\ x^2 \\ x \\ 1 \end{pmatrix} = 0$$

- Solving d_i linearly, and repeat for all $\mathrm{X_i}$

EPnP Algorithm

- The linear algorithm in previous slice is $O(n^5)$
- EPnP is $O(n)$
  - Also first computes 3D points in the camera frame

EPnP: An Accurate $O(n)$ Solution to the PnP Problem

Vincent Lepetit · Francesc Moreno-Noguer · Pascal Fua

IJCV 2008

SFU

ZHEJIANG UNIVERSITY

Questions?

SFU

浙江大學
ZHEJIANG UNIVERSITY

Calibration from Vanishing Points

- Scenes contain parallel lines, which intersect at vanishing points

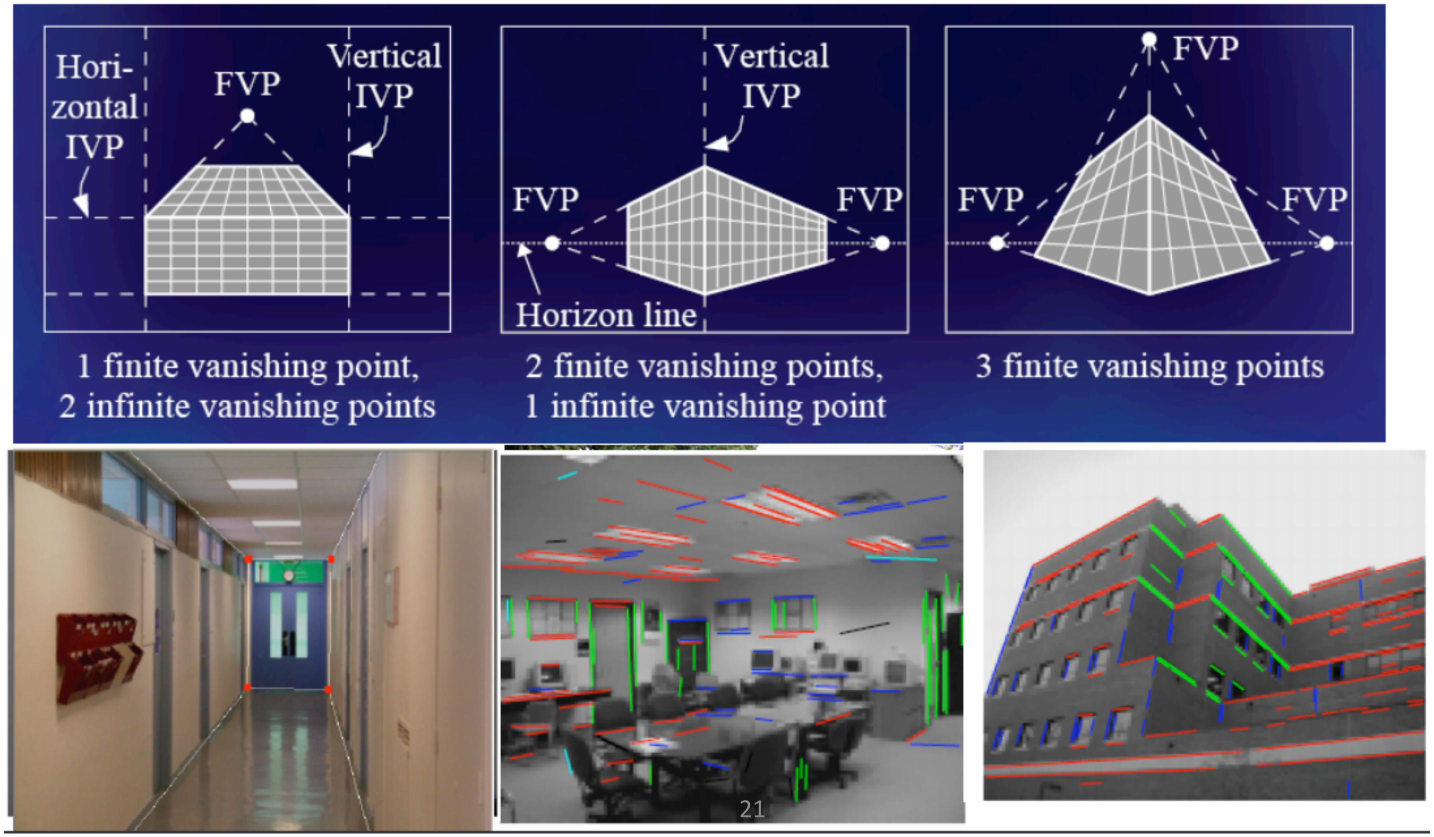


Image of Absolute Conic (IAC)

- A 3D direction e, e.g. $e = [0,0,1]^T$
- Its vanishing point is projected at

$$v = K[R\ t]\begin{bmatrix} e \\ 0 \end{bmatrix} = KRe$$

$$\Rightarrow \qquad e = R^T K^{-1} v$$

- For two perpendicular 3D directions e_i, e_j

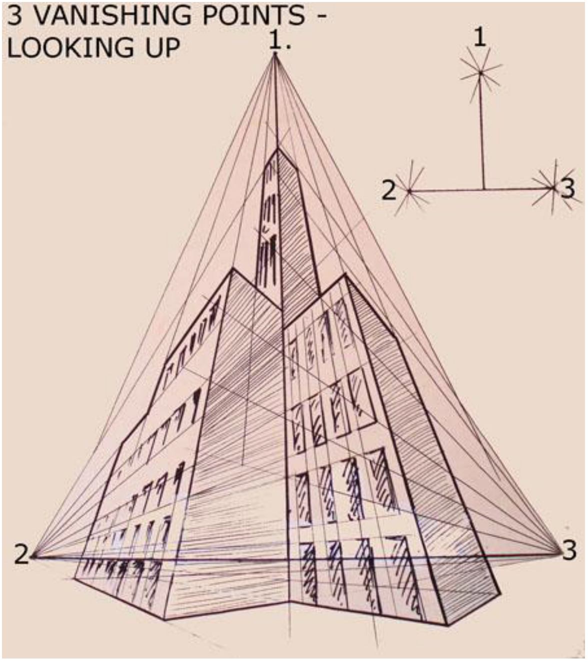
$$0 = e_i^T e_j = (R^T K^{-1} v_i)^T \left(R^T K^{-1} v_j\right)$$
$$= v_i^T K^{-T} R R^T K^{-1} v_j$$
$$= v_i^T \underline{K^{-T} K^{-1}} v_j$$

image of the absolute conic (IAC), denoted by ω

SFU

浙江大學 ZHEJIANG UNIVERSITY

Calibration from Vanishing Points



- From 3 known vanishing points v_i, v_j, v_k
  - Assume their directions are mutual orthogonal
- Obtain three equations:

$$v_i^T \omega v_j = 0$$
$$v_j^T \omega v_k = 0$$
$$v_i^T \omega v_k = 0$$

- Assume simple intrinsics:

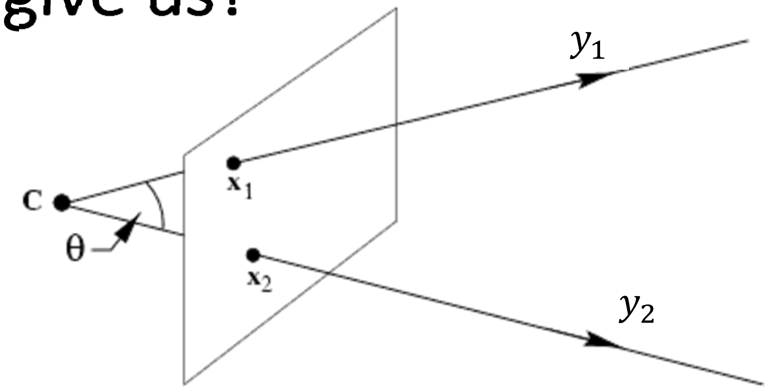
$$K = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow K^{-1} = \begin{bmatrix} 1/f & 0 & -u_0/f \\ 0 & 1/f & -v_0/f \\ 0 & 0 & 1 \end{bmatrix}$$

- Solve for f, u_0, v_0 from the three equations

what does the intrinsics give us?

$$x = K[I|0]\begin{bmatrix} y \\ 0 \end{bmatrix}$$

$$\Rightarrow \qquad y = K^{-1}x$$



We can recover the 3D ray defined by the camera center and a pixel, if the camera is calibrated.

We can further measure the angle between two such rays.

$$\cos\theta = \frac{y_1^T y_2}{\sqrt{(y_1{}^T y_1)(y_2{}^T y_2)}} = \frac{x_1{}^T (K^{-T}K^{-1}) x_2}{\sqrt{\left(x_1{}^T \underline{(K^{-T}K^{-1})} x_1\right)\left(x_2{}^T (K^{-T}K^{-1}) x_2\right)}}$$

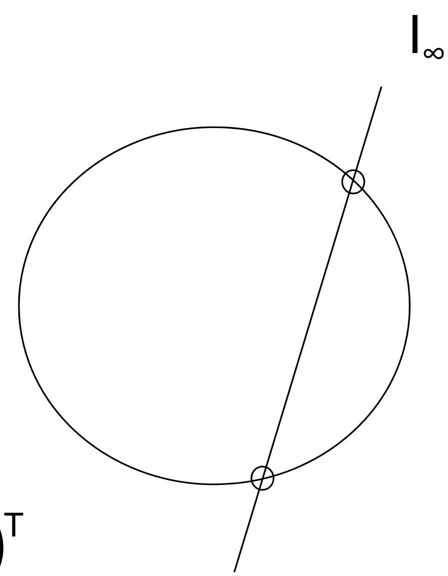
image of the absolute conic (IAC)

Questions?

The Circular Points on a 2D Plane

The circular points I, J are two special points at infinity

$$\mathrm{I} = \begin{pmatrix} 1 \\ i \\ 0 \end{pmatrix} \quad \mathrm{J} = \begin{pmatrix} 1 \\ -i \\ 0 \end{pmatrix} \quad i = \sqrt{-1}$$



"circular points": any circle must pass through these two points

Intersect a circle with the line at infinity

$$x_1^2 + x_2^2 + dx_1x_3 + ex_2x_3 + fx_3^2 = 0$$
$$x_3 = 0$$

It leads to an equation with two solutions

$$x_1^2 + x_2^2 = 0 \quad \Rightarrow \quad \mathrm{I} = (1, i, 0)^{\mathrm{T}} \quad \mathrm{J} = (1, -i, 0)^{\mathrm{T}}$$

Fit a general conic needs five points

Fit a general circle only needs three points (because we implicitly used these two circle points)

SFU

浙江大学 ZHEJIANG UNIVERSITY

The Absolute Conic in 3D Space

- Consider the 3D space consists of many 2D planes (layer by layer).
- Each plane has two circular points.
- All these points form a conic, called the absolute conic Ω_∞!
  - All points satisfying a conic equation.

$$X_4 = 0;$$

$$(X_1, X_2, X_3)\begin{pmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{pmatrix}\begin{pmatrix}X_1\\X_2\\X_3\end{pmatrix} = 0 \qquad \Omega_\infty = \begin{pmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{pmatrix}$$

- Any sphere must pass through Ω_∞!
- Ω_∞ lies on the plane at infinity π_∞!

SFU

ZHEJIANG UNIVERSITY

Image of Absolute Conic (IAC)

- Why Ω_∞ matters?
- It allows measurement of angles (even after projective transformations)!

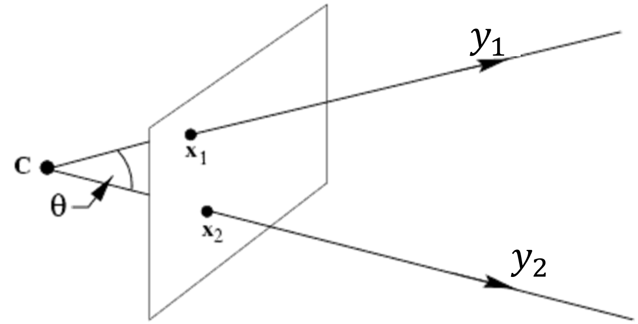
$$\cos\theta = \frac{\left(\mathrm{y}_1^{\mathrm{T}}\Omega_\infty \mathrm{y}_2\right)}{\sqrt{\left(\mathrm{y}_1^{\mathrm{T}}\Omega_\infty \mathrm{y}_1\right)\left(\mathrm{y}_2^{\mathrm{T}}\Omega_\infty \mathrm{y}_2\right)}}$$

$(y_1, 0)$ is the vanishing point of a 3D direction

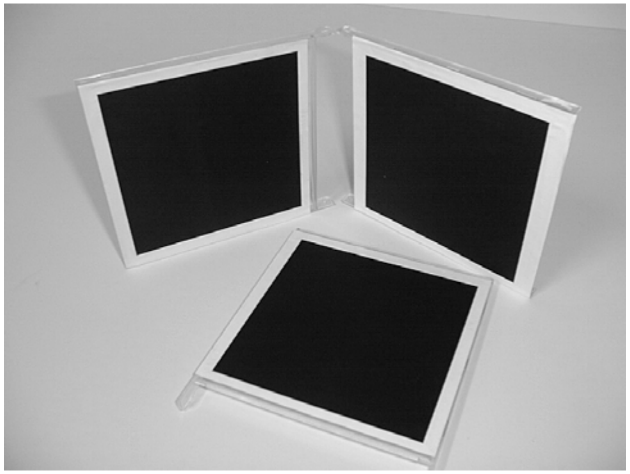
- The angle between two camera rays:

$$\cos\theta = \frac{\mathrm{y}_1^T y_2}{\sqrt{(\mathrm{y}_1{}^{\mathrm{T}}\mathrm{y}_1)(\mathrm{y}_2{}^{\mathrm{T}}\mathrm{y}_2)}} = \frac{\mathrm{x}_1{}^{\mathrm{T}}(\mathrm{K}^{-\mathrm{T}}\mathrm{K}^{-1})\mathrm{x}_2}{\sqrt{\left(\mathrm{x}_1{}^{\mathrm{T}}(\mathrm{K}^{-\mathrm{T}}\mathrm{K}^{-1})\mathrm{x}_1\right)\left(\mathrm{x}_2{}^{\mathrm{T}}(\mathrm{K}^{-\mathrm{T}}\mathrm{K}^{-1})\mathrm{x}_2\right)}}$$

Consider the mapping from π_∞ to the image plane.
$K^{-T}K^{-1}$ is the image of Ω_∞ after mapping!
It allows us to measure angles between rays.



Zhang's Calibration Method



(i) compute H for each square; mapping the checkboard to the image plane (assuming corners are (0,0),(1,0),(0,1),(1,1) in the checkboard plane)

(ii) compute the imaged circular points $H(1,\pm i,0)^T$ (3D translation of the checkboard does not move its circular points)

(iii) fit a conic to 6 circular points

(iv) compute K from ω through Cholesky factorization

SFU

浙江大学 ZHEJIANG UNIVERSITY

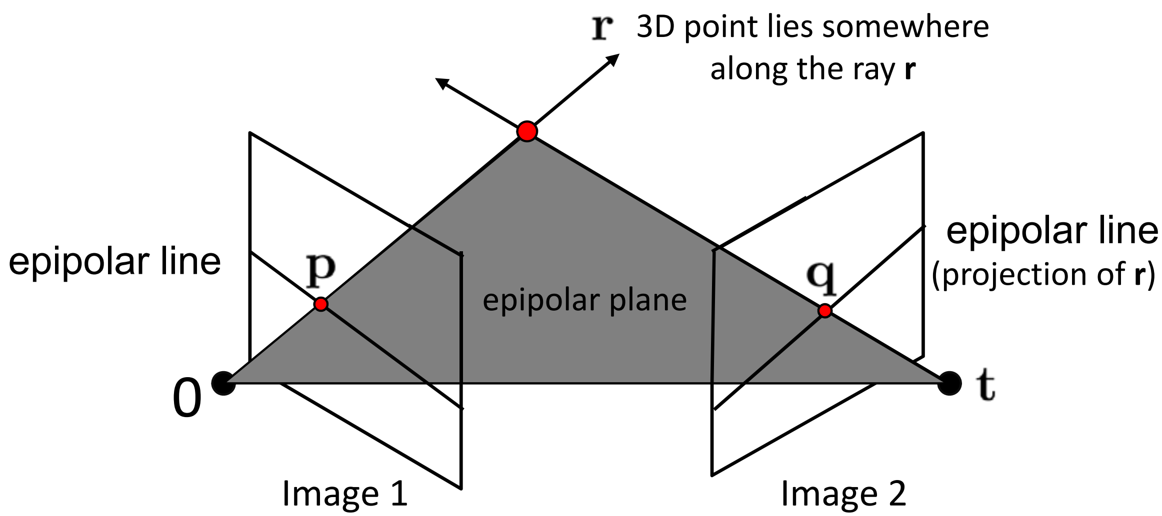
Questions?

Outline

- Camera Calibration
- 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.

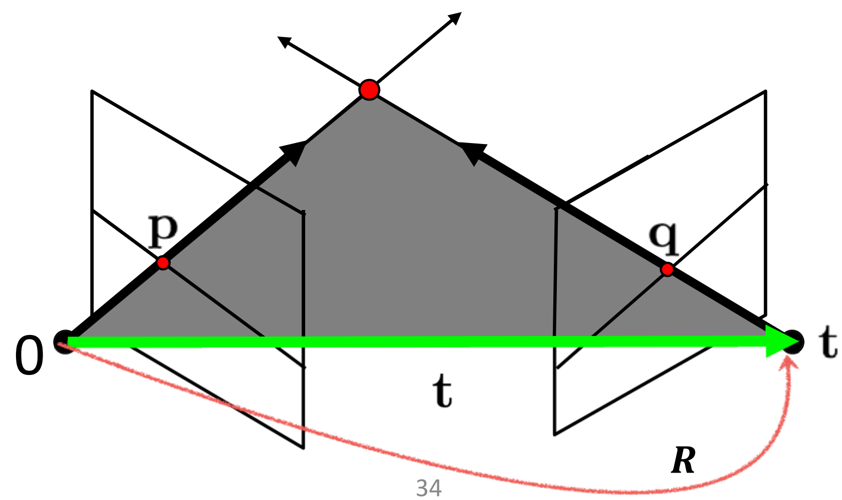
Algebra is but written geometry;
geometry is but drawn algebra.
-- Sophie Germain

Essential Matrix – calibrated case

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

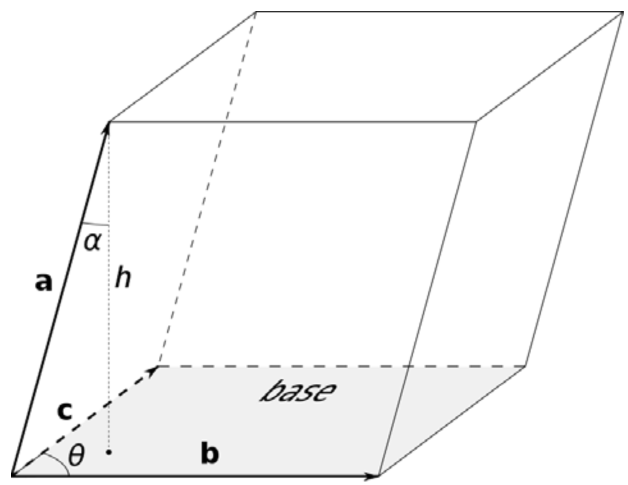
  i.e. $\boldsymbol{p}, \boldsymbol{q}$ are known directions ($\boldsymbol{p}$ is in camera frame 1, $\boldsymbol{q}$ is in camera frame 2)
  - Suppose $\boldsymbol{R}, \mathbf{t}$ are the rotations and translations between the two cameras

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

$$\boldsymbol{p} \cdot (\boldsymbol{t} \times \boldsymbol{R}^T\boldsymbol{q}) = 0$$

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

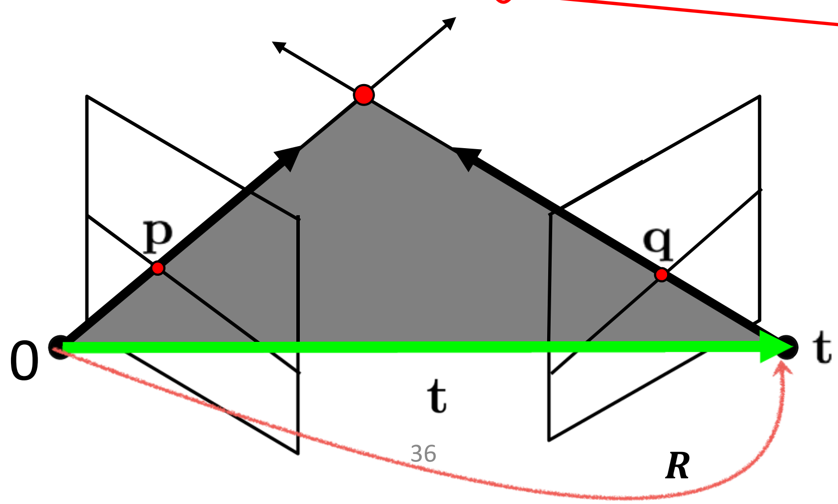
➔

$$\boldsymbol{p}^T[\boldsymbol{t}]_\times \boldsymbol{R}^T\boldsymbol{q} = 0$$

➔

$$\boldsymbol{q}^T\boldsymbol{R}[\boldsymbol{t}]_\times\boldsymbol{p} = \boldsymbol{q}^T\boldsymbol{E}\boldsymbol{p} = 0$$

Essential Matrix
[Longuet-Higgins 1981]



SFU

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):

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

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

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

➔

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

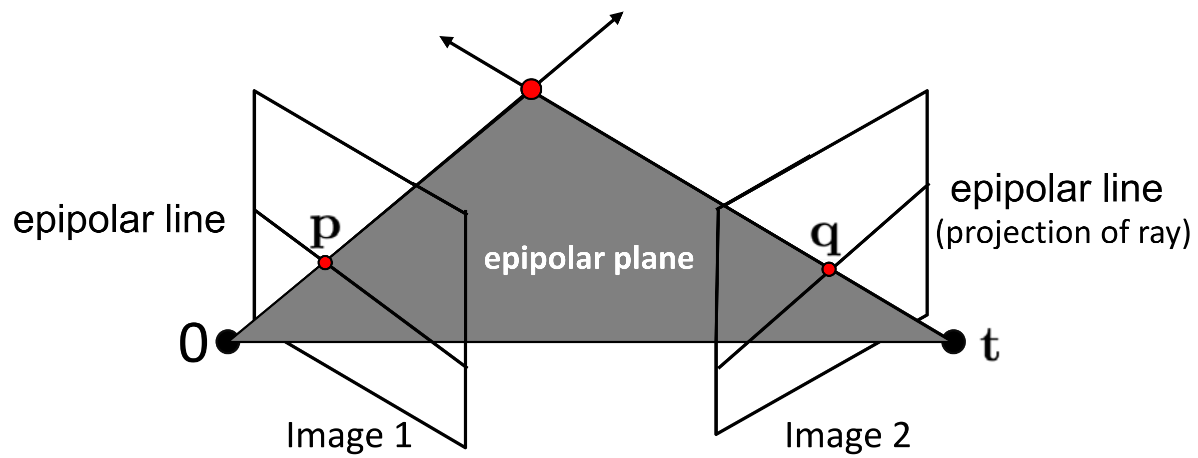
Fundamental Matrix
[Oliver Faugeras 1992]

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

SFU

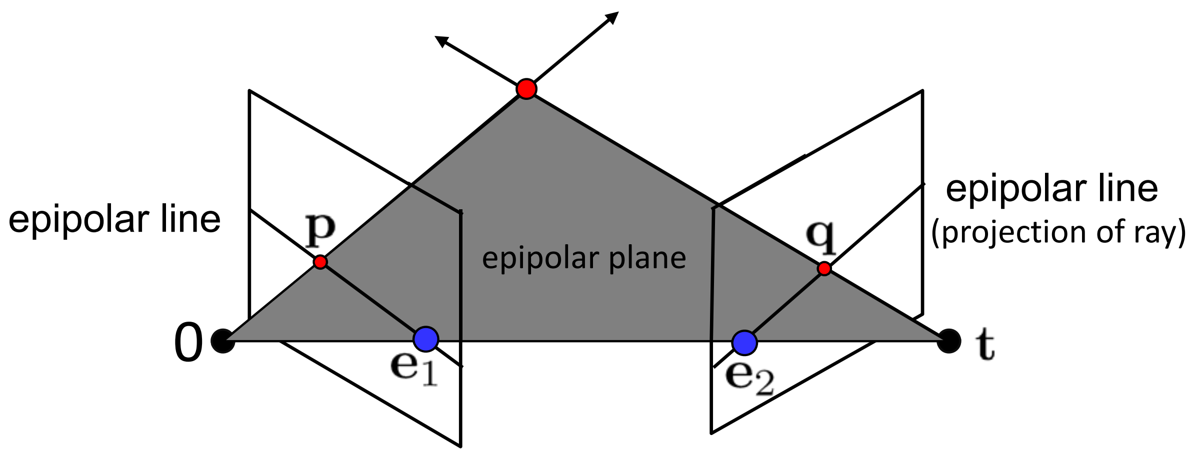
浙江大學 ZHEJIANG UNIVERSITY

Fundamental Matrix – summary



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

Fundamental Matrix – summary

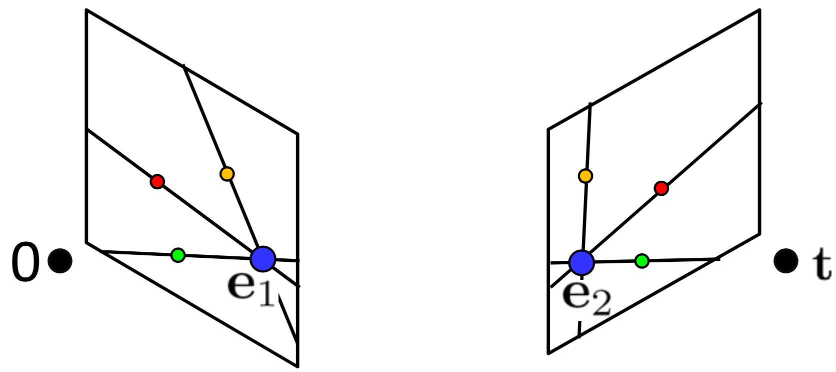


- Two Special points: $\mathbf{e}_1$ and $\mathbf{e}_2$ (the *epipoles*): projection of one camera into the other
- All of the epipolar lines in an image pass through the epipole
- Epipoles can be computed from $\boldsymbol{F}$ as well: $\boldsymbol{e}_2^T\boldsymbol{F} = 0$ and $\boldsymbol{F}\boldsymbol{e}_1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

SFU

浙江大学 ZHEJIANG UNIVERSITY

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 $\boldsymbol{F}$ as well: $\boldsymbol{e}_2^T\boldsymbol{F} = 0$ and $\boldsymbol{F}\boldsymbol{e}_1 = 0$
  - For any pixel $\boldsymbol{p}$, $\boldsymbol{F}\boldsymbol{p}$ is its epipolar line, which must pass through $\boldsymbol{e}_2$
  - Therefore, $\boldsymbol{e}_2^T\boldsymbol{F}\boldsymbol{p} = 0$ for any $\boldsymbol{p}$ ➔ $\boldsymbol{e}_2^T\boldsymbol{F} = 0$
  - So, $\boldsymbol{F}$ is rank 2

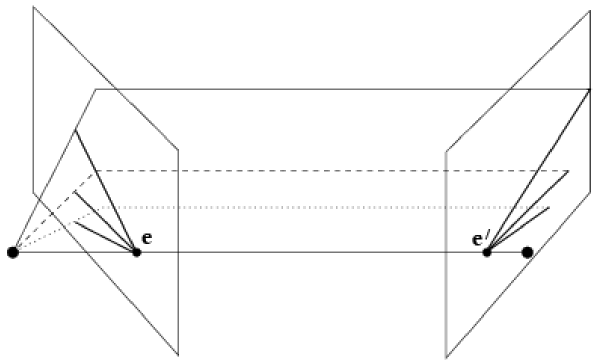
SFU

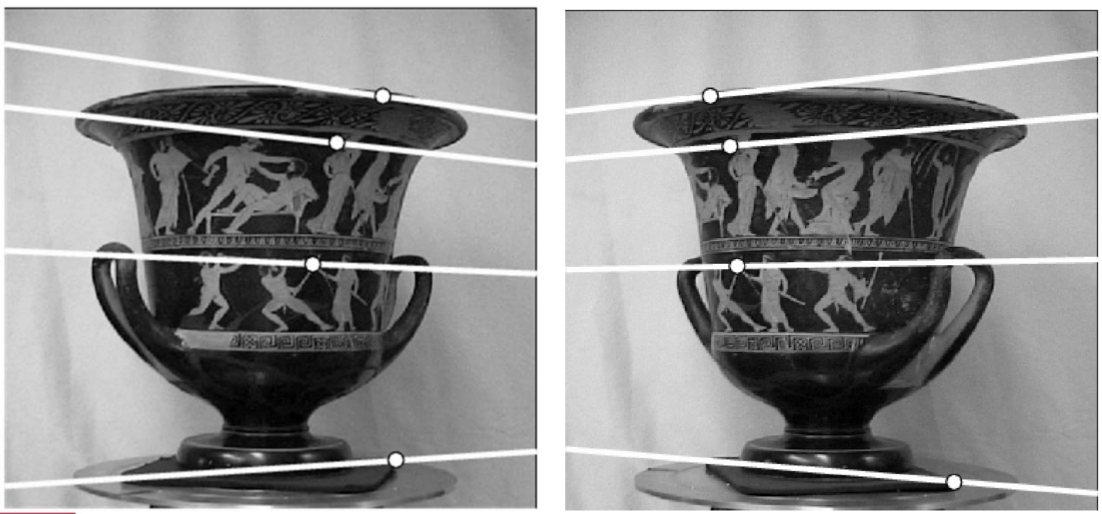
浙江大學 ZHEJIANG UNIVERSITY

The Fundamental Matrix Song

Questions?

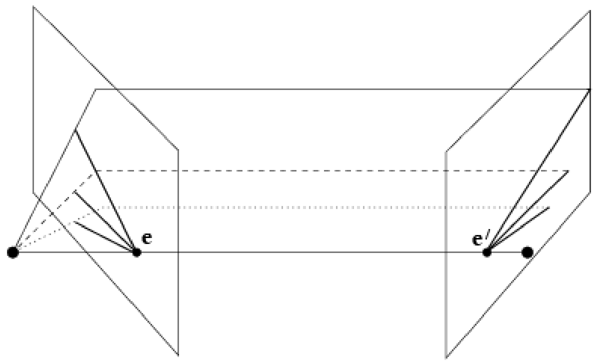
Example: converging cameras

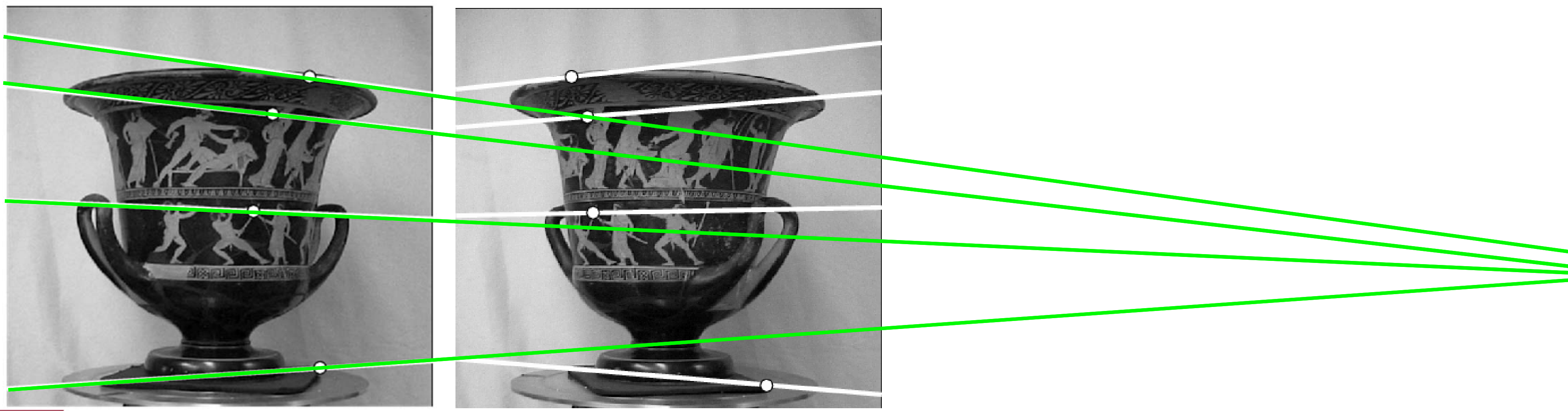




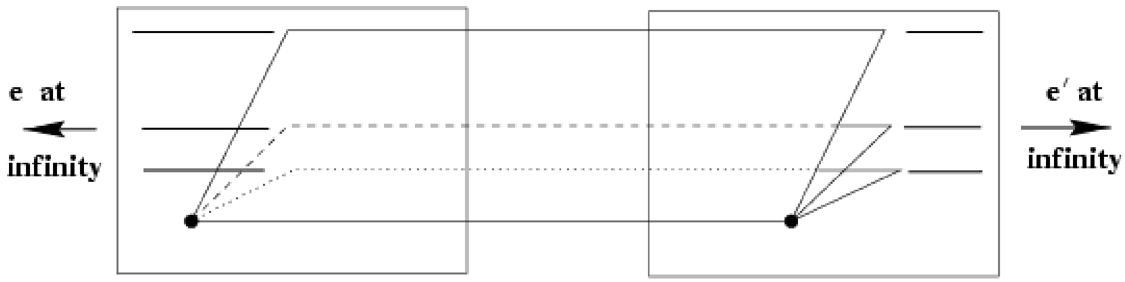
Where is the epipole in this image?

Example: converging cameras

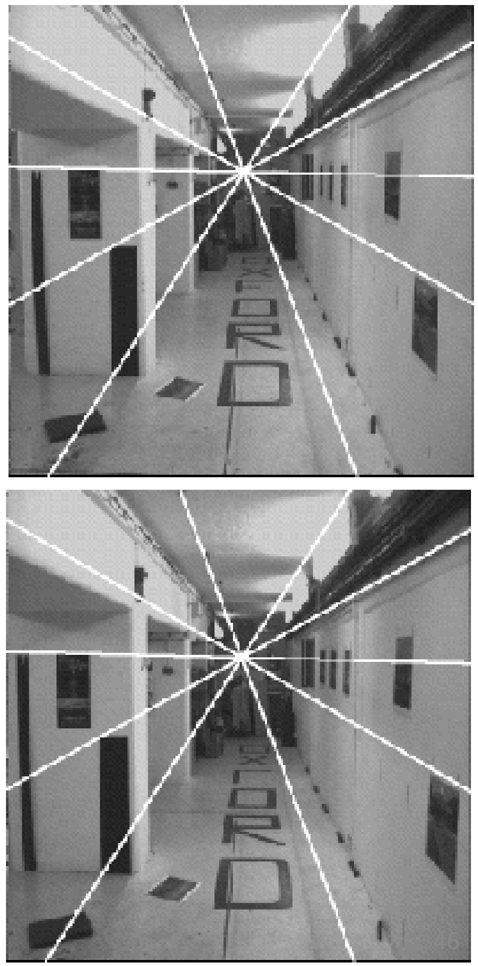


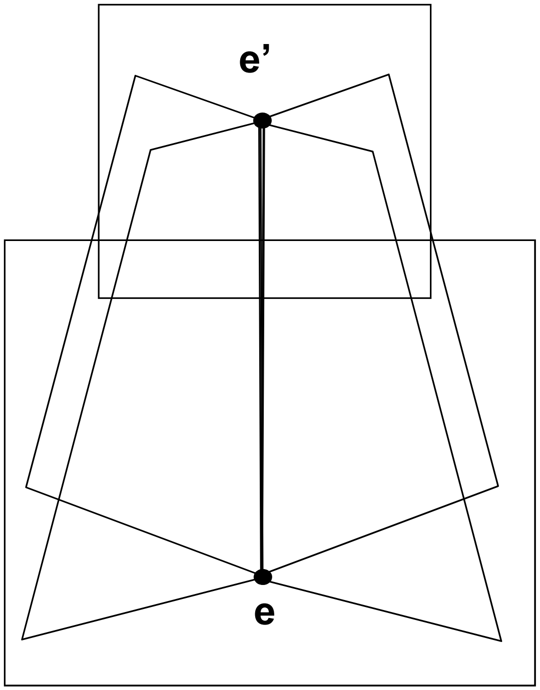


Example: motion parallel with image plane



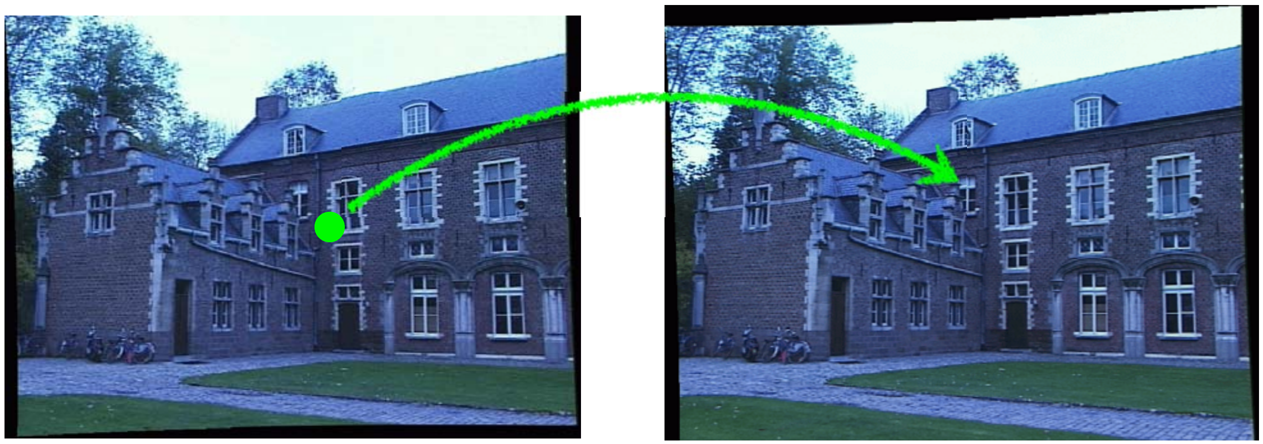
Example: forward motion





the epipolar constraint for stereo vision

Task: P dwfk#srlqw#lq#ohiw#lp djh#wr#srlqw#lq#uljkw#lp djh



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

$$\underbrace{[x'x, x'y, x', y'x, y'y, y', x, y, 1]}_{\text{(data)}}\underbrace{[f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33}]^{\mathrm{T}}}_{\text{(unknowns)}} = 0$$

linear equation

$$\begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix}\mathbf{f} = 0$$

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

The Singularity Constraint

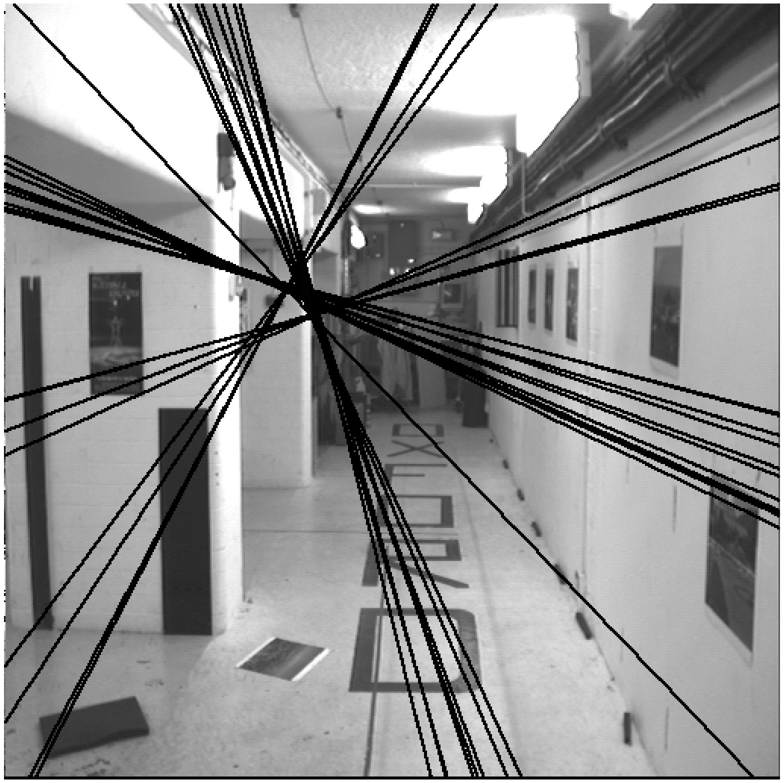
$$e'^{T} F = 0 \qquad Fe = 0 \qquad \det F = 0 \qquad \text{rank } F = 2$$

SVD from linearly computed F matrix (rank 3)

$$F = U \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{bmatrix} V^T = U_1 \sigma_1 V_1^T + U_2 \sigma_2 V_2^T + U_3 \sigma_3 V_3^T$$

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

$$F' = U \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & 0 \end{bmatrix} V^T = U_1 \sigma_1 V_1^T + U_2 \sigma_2 V_2^T$$



when F is non-singular, epipolar lines won't intersect at the same point

the NOT normalized 8-point algorithm

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

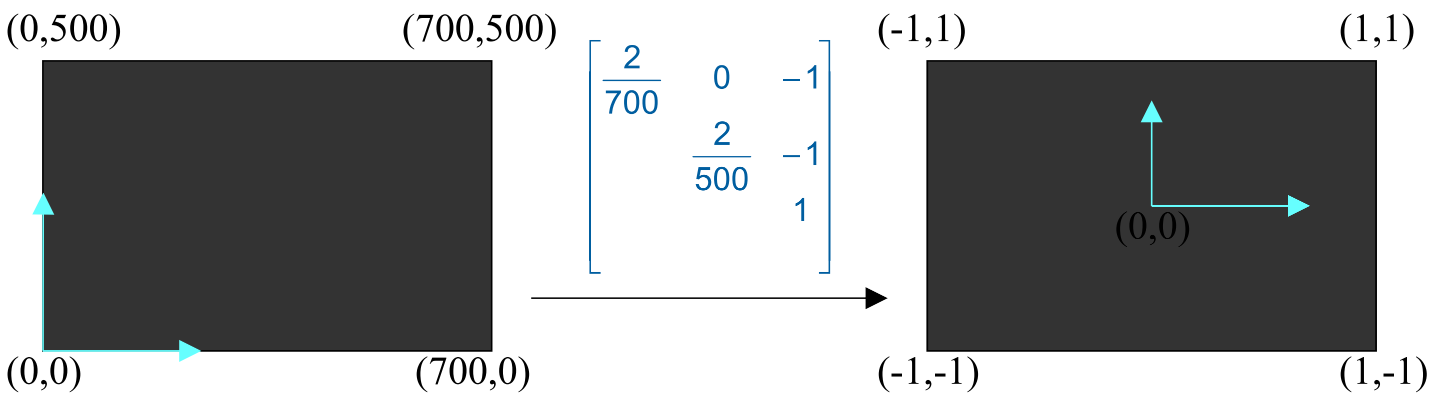
$$
\begin{array}{ccccccccc}
\sim 10000 & \sim 10000 & \sim 100 & \sim 10000 & \sim 10000 & \sim 100 & \sim 100 & \sim 100 & 1
\end{array}
$$

!

Orders of magnitude difference
Between column of data matrix
$\rightarrow$ 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)

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

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

$$\Rightarrow \mathrm{A}[\mathrm{V}_8 \mathrm{V}_9] = 0_{9\mathrm{x}2} \qquad \Rightarrow \mathrm{A}\left(\mathrm{V}_8 + \lambda \mathrm{V}_9\right) = 0_{9\mathrm{x}2}$$

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

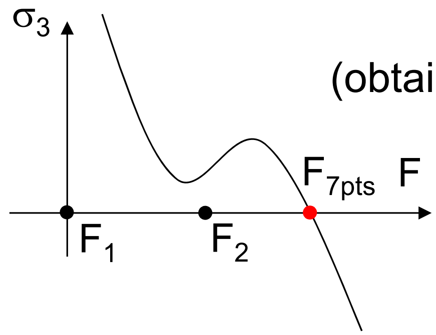
one parameter family of solutions

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

SFU

浙江大學 ZHEJIANG UNIVERSITY

7-point Algorithm – the minimum case



(obtain 1 or 3 solutions)

$$\det(F_1 + \lambda F_2) = a_3\lambda^3 + a_2\lambda^2 + a_1\lambda + a_0 = 0$$ (cubic equation)

$$\det(F_1 + \lambda F_2) = \det F_2 \det(F_2^{-1}F_1 + \lambda I) = 0$$

Compute possible λ as eigenvalues of $F_2^{-1}F_1$ (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:

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

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)

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

SFU

浙江大學 ZHEJIANG UNIVERSITY

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

SFU

浙江大學 ZHEJIANG UNIVERSITY

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

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

Or, take 8-point algorithm to solve $\boldsymbol{E}$, 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

- $\boldsymbol{E}$ has less degrees of freedom than $\boldsymbol{F}$
- In principal, 5 pair of corresponding points are sufficient to decide $\boldsymbol{E}$.
  - 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

SFU

ZHEJIANG UNIVERSITY

Getting Camera Matrices from E

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

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

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

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

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

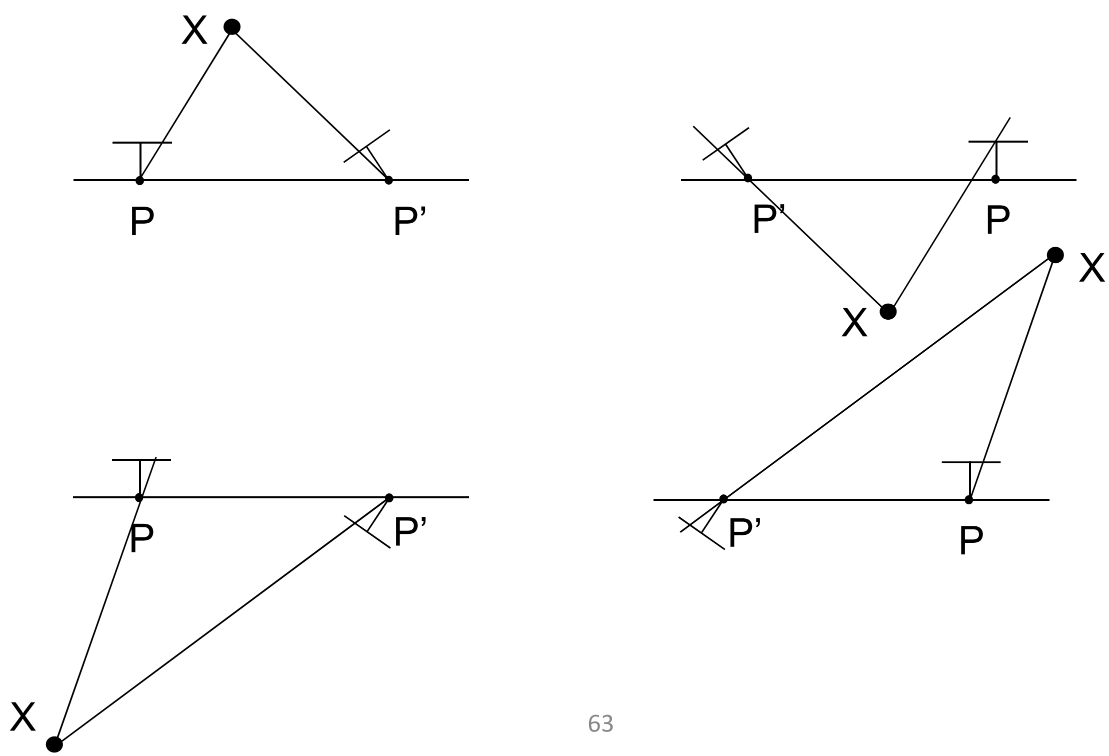
For the proof, please refer to section 9.6 of the 'multiview geometry' book

SFU

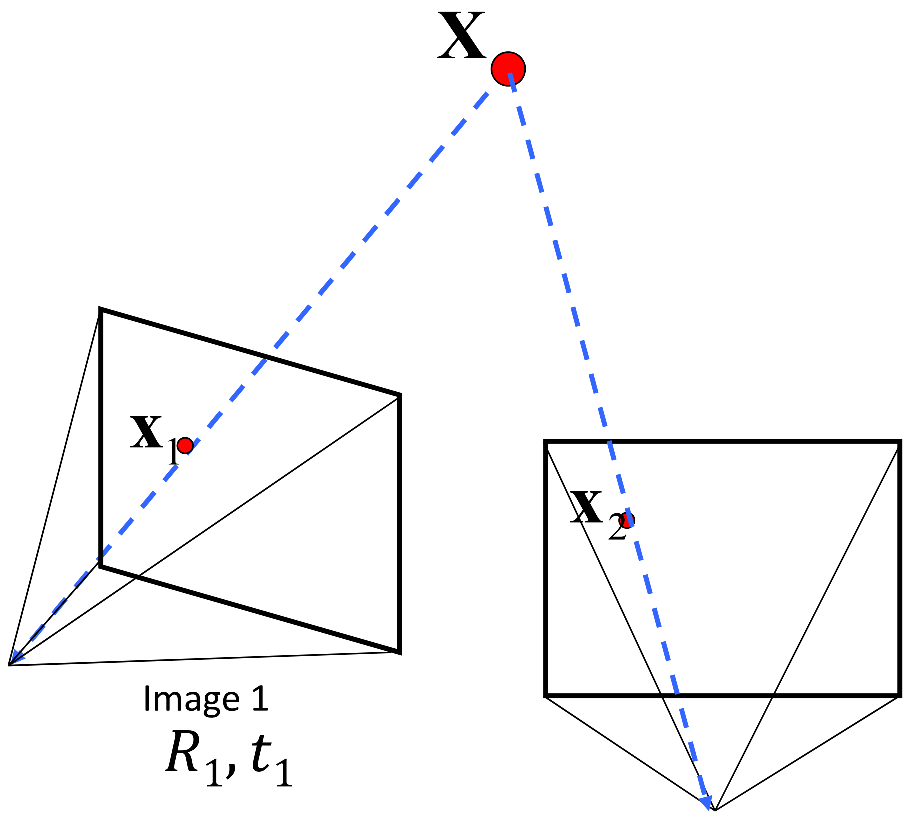
ZHEJIANG UNIVERSITY

Selecting from the Four Solutions

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



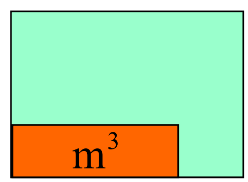
Triangulation (to be studied soon)



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

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

P =


Pick the Solution

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

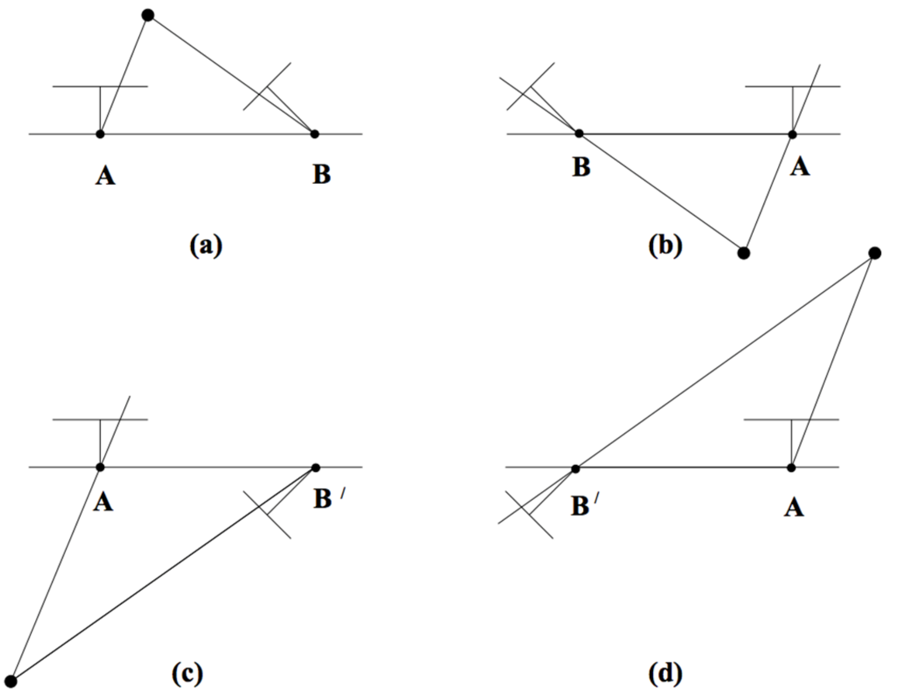


Fig. 9.12. **The four possible solutions for calibrated reconstruction from E.** *Between the left and right sides there is a baseline reversal. Between the top and bottom rows camera B rotates* 180° *about the baseline. Note, only in (a) is the reconstructed point in front of both cameras.*

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

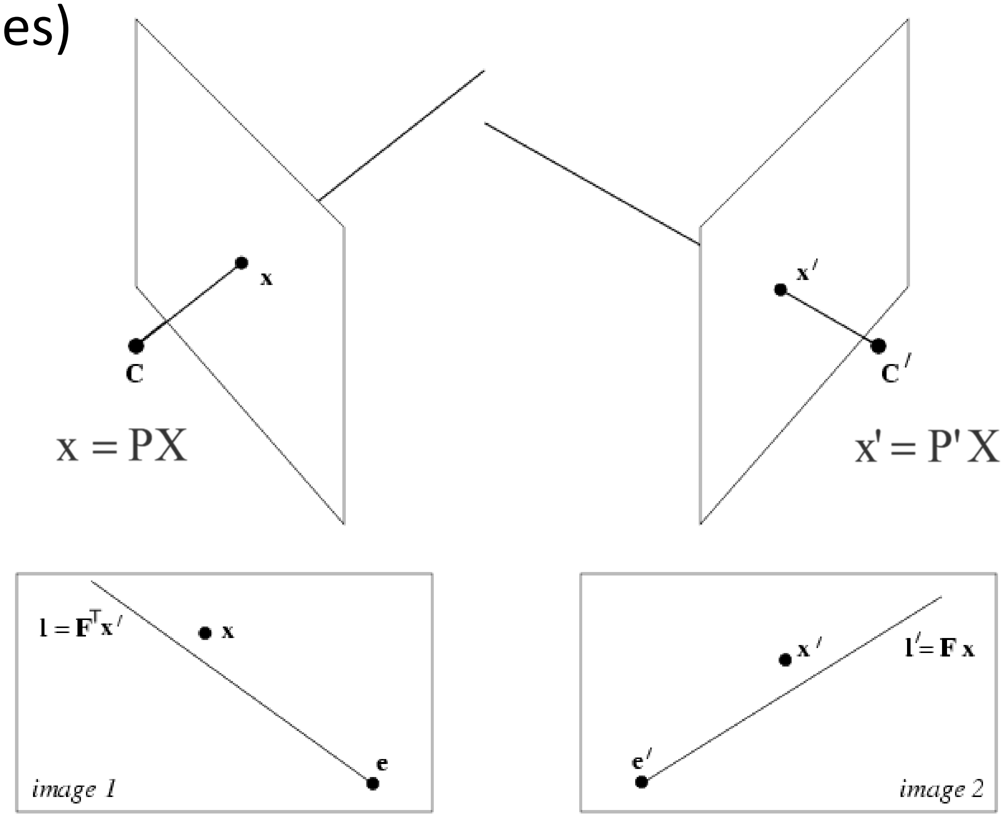
Questions?

Outline

- Camera Calibration
- Two-view Geometry
- Triangulation

Point Reconstruction

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



Direct Linear Transform

$$\mathrm{x} = \mathrm{PX} \quad \Rightarrow \mathrm{x} \times \mathrm{PX} = 0 \qquad \mathrm{x}' = \mathrm{P}'\mathrm{X} \quad \Rightarrow \mathrm{x}' \times \mathrm{P}'\mathrm{X} = 0$$

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

Homogeneous coordinate, add constraint $\|\mathrm{X}\| = 1$

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

This method minimizes an algebraic error

SFU

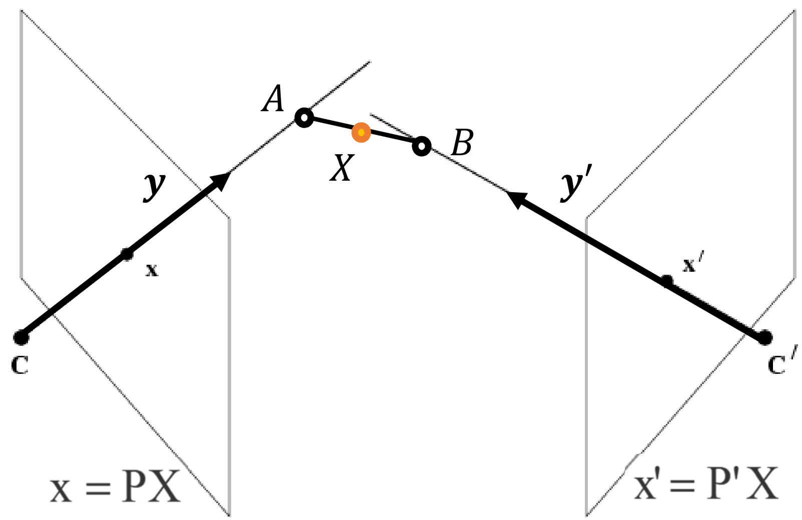
ZHEJIANG UNIVERSITY

Mid-point Algorithm

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

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

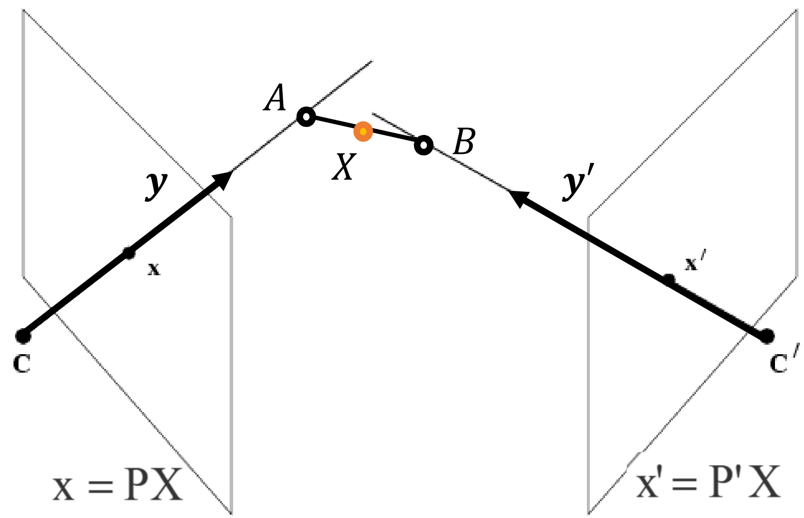
- $\boldsymbol{c}, \boldsymbol{c}', \mathbf{y}, \mathbf{y}'$ are all in the same coordinate system (e.g. camera frame 1)



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

Mid-point Algorithm

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



SFU

浙江大學 ZHEJIANG UNIVERSITY

Mid-point Algorithm

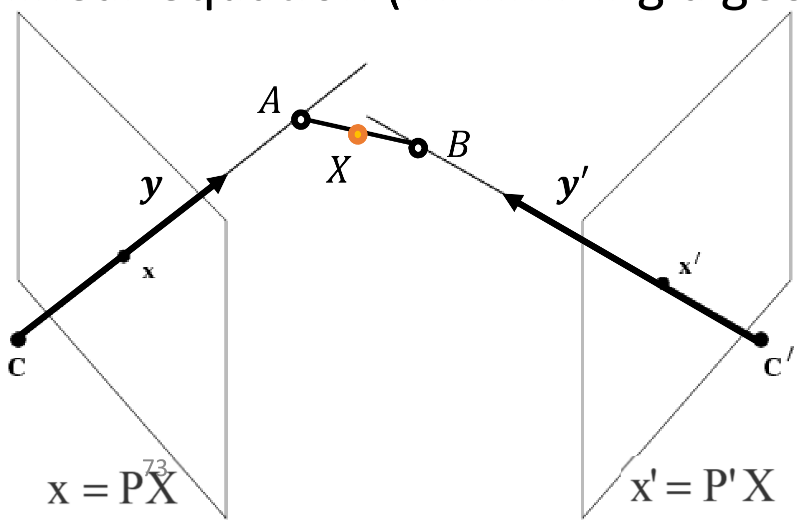
- Since $\boldsymbol{AB}$ is the mutual perpendicular line segment
  - $\boldsymbol{AB} \perp \boldsymbol{y},\ \boldsymbol{AB} \perp \boldsymbol{y}'$
- This means:

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

- This generates three equations of d_1, d_2

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

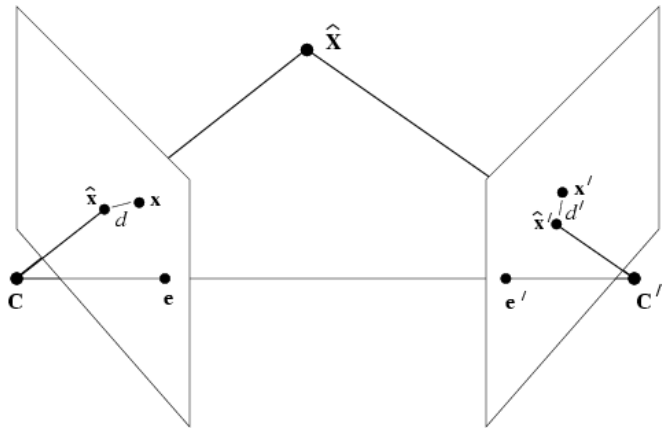
- Solve d_1, d_2 from the above linear equation (minimizing a geometric error)



Reprojection Error

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

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



compute using the Levenberg-Marquardt algorithm

This triangulation works for uncalibrated cameras

- The algebraic error and mid-point algorithm needs K, K' (pre-calibrated cameras)

Questions?

Algorithms Studied Today

- Camera calibration (resection / PnP)
  - 3D <-> 2D correspondences, compute the camera pose
- 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

SFU

浙江大学 ZHEJIANG UNIVERSITY