Lecture
Epipolar Geometry

Liangliang Nan

Today’s Agenda

- Review Camera calibration
- Epipolar geometry

Review of Camera Calibration

- Camera calibration
  - Recovering K
  - Recovering R and $\mathbf{t}$

$$\mathbf{p} = M\mathbf{P}$$
$$= \boxed{K}\ \boxed{\begin{bmatrix} R & \mathbf{t} \end{bmatrix}}\ \mathbf{P}$$

Internal (intrinsic) parameters

External (extrinsic) parameters

Review of Camera Calibration

- How many parameters to recover?
  - 5 intrinsic parameters
    - 2 for focal length
    - 2 for offset
    - 1 for skewness
  - 6 extrinsic parameters
    - 3 for rotation
    - 3 for translation

$$\mathbf{p} = M\mathbf{P}$$
$$= \boxed{K} \boxed{\begin{bmatrix} R & \mathbf{t} \end{bmatrix}} \mathbf{P}$$

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

Review of Camera Calibration

- Parameters to recover: 11
- Corresponding 3D-2D point pairs
  - Each 3D-2D point pair -> 2 constraints
  - 11 unknown -> 6 point correspondence
  - Use more to handle noisy data

$$\mathbf{p}_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = M\mathbf{P}_i = \begin{bmatrix} \frac{\mathbf{P}_i^T \mathbf{m}_1}{\mathbf{P}_i^T \mathbf{m}_3} \\ \\ \frac{\mathbf{P}_i^T \mathbf{m}_2}{\mathbf{P}_i^T \mathbf{m}_3} \end{bmatrix} \Rightarrow \begin{aligned} \mathbf{P}_i^T \mathbf{m}_1 - u_i(\mathbf{P}_i^T \mathbf{m}_3) &= 0 \\ \mathbf{P}_i^T \mathbf{m}_2 - v_i(\mathbf{P}_i^T \mathbf{m}_3) &= 0 \end{aligned}$$

Review of Camera Calibration

- Parameters to recover: 11
- Corresponding 3D-2D point pairs: >= 6
- How to solve it?
  - $\mathbf{m} = 0$ is always a trivial solution
  - If $\mathbf{m} \neq 0$ is a solution, then any $k * \mathbf{m}$ is also a solution

$$\begin{bmatrix} \mathbf{P}_1^T & 0^T & -u_1\mathbf{P}_1^T \\ 0^T & \mathbf{P}_1^T & -v_1\mathbf{P}_1^T \\ & \vdots & \\ \mathbf{P}_n^T & 0^T & -u_n\mathbf{P}_n^T \\ 0^T & \mathbf{P}_n^T & -v_n\mathbf{P}_n^T \end{bmatrix} \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \mathbf{m}_3^T \end{bmatrix} = P\mathbf{m} = 0$$

Review of Camera Calibration

- Parameters to recover: 11
- Corresponding 3D-2D point pairs: >= 6
- How to solve it?
  - $\mathbf{m} = 0$ is always a trivial solution
  - If $\mathbf{m} \neq 0$ is a solution, then any $k * \mathbf{m}$ is also a solution
  - Constrained optimization

$$\begin{bmatrix} \mathbf{P}_1^T & 0^T & -u_1\mathbf{P}_1^T \\ 0^T & \mathbf{P}_1^T & -v_1\mathbf{P}_1^T \\ & \vdots & \\ \mathbf{P}_n^T & 0^T & -u_n\mathbf{P}_n^T \\ 0^T & \mathbf{P}_n^T & -v_n\mathbf{P}_n^T \end{bmatrix} \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \mathbf{m}_3^T \end{bmatrix} = P\mathbf{m} = 0 \Rightarrow \begin{array}{ll} \underset{\mathbf{m}}{\text{minimize}} & \|P\mathbf{m}\|^2 \\ \text{subject to} & \|\mathbf{m}\|^2 = 1 \end{array}$$

Review of Camera Calibration

- Solved using SVD

$$\boxed{P}\mathbf{m} = 0$$

SVD decomposition of P

$$\boxed{U_{2n\times12}\ D_{12\times12} V^{T}_{12\times12}}$$

Last column of V gives $\mathbf{m}$

(Why? See page 593 of Hartley & Zisserman. Multiple view geometry in computer vision)

Review of Camera Calibration

Intrinsic parameters:

$$\rho = \pm \frac{1}{\|\mathbf{a}_3\|}$$

$$c_x = \rho^2 (\mathbf{a}_1 \cdot \mathbf{a}_3)$$

$$c_y = \rho^2 (\mathbf{a}_2 \cdot \mathbf{a}_3)$$

$$\cos\theta = -\frac{(\mathbf{a}_1 \times \mathbf{a}_3) \cdot (\mathbf{a}_2 \times \mathbf{a}_3)}{\|\mathbf{a}_1 \times \mathbf{a}_3\| \cdot \|\mathbf{a}_2 \times \mathbf{a}_3\|}$$

$$\alpha = \rho^2 \|\mathbf{a}_1 \times \mathbf{a}_3\| \sin\theta$$

$$\beta = \rho^2 \|\mathbf{a}_2 \times \mathbf{a}_3\| \sin\theta$$

Extrinsic parameters:

$$\mathbf{r}_1 = \frac{\mathbf{a}_2 \times \mathbf{a}_3}{\|\mathbf{a}_2 \times \mathbf{a}_3\|}$$

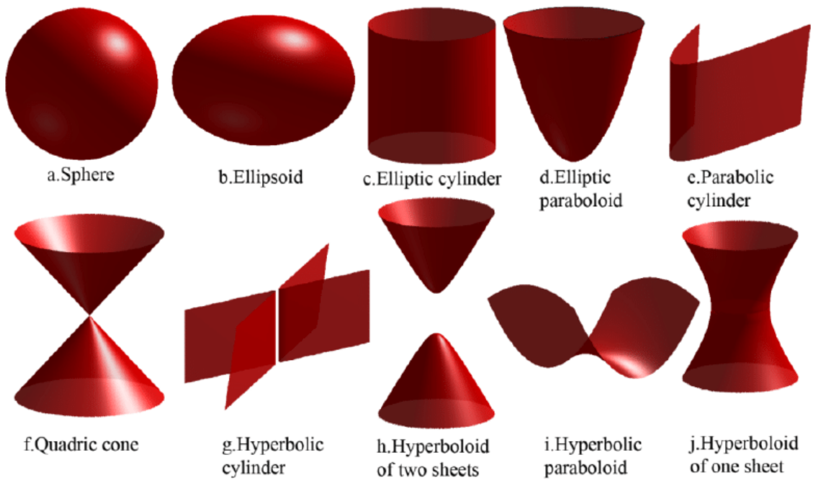
$$\mathbf{r}_3 = \rho \mathbf{a}_3$$

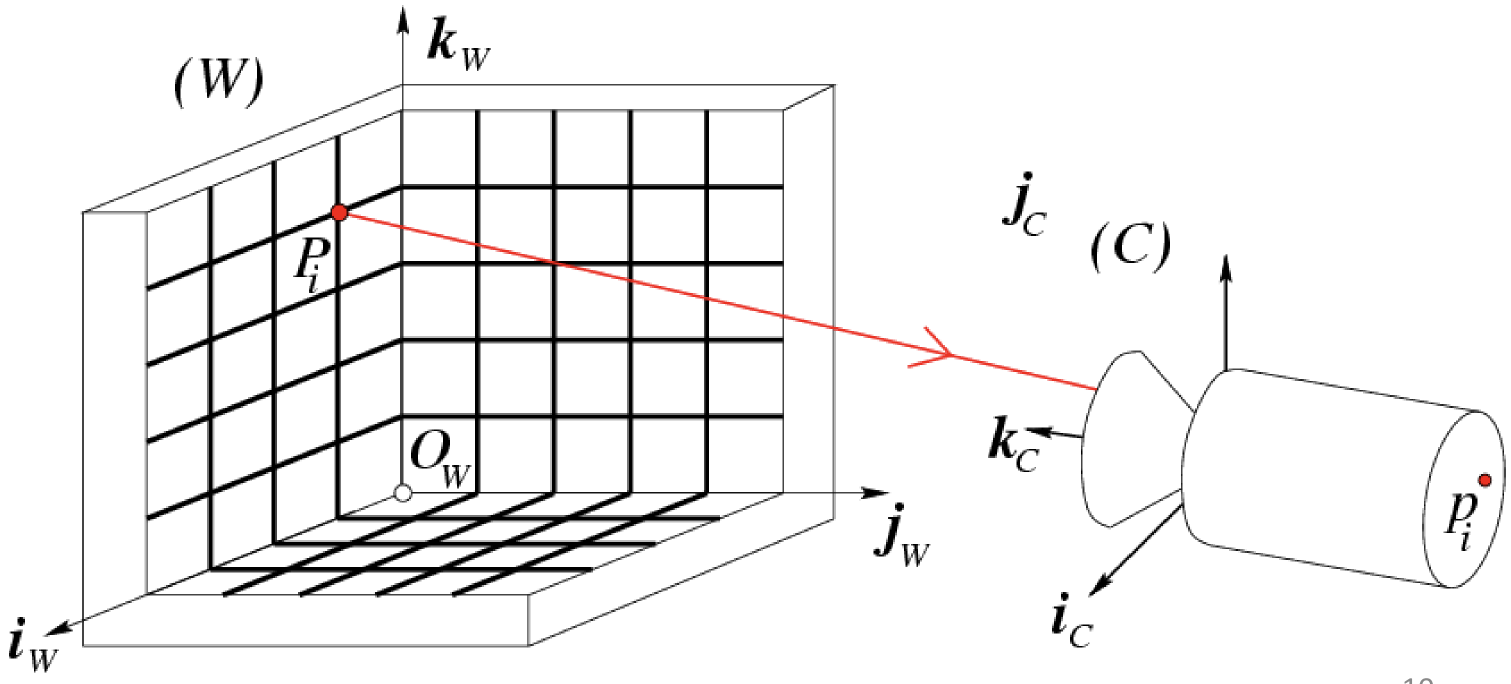
$$\mathbf{r}_2 = \mathbf{r}_3 \times \mathbf{r}_1$$

$$\mathbf{t} = \rho K^{-1} \mathbf{b}$$

Review of Camera Calibration

- Not always solvable
  - $\{P_i\}$ cannot lie on the same plane
  - $\{P_i\}$ cannot lie on the intersection curve of two quadric surfaces





Quiz

Which of the following will change the camera intrinsic matrix?

(a) When zooming in.

(b) When rotating the camera around its local origin.

(c) When changing the resolution of the image.

(d) When the camera is moved.

Quiz

Which of the following will change the camera intrinsic matrix?

(a) When zooming in. $[f_x, f_y]$

(b) When rotating the camera around its local origin. R

(c) When changing the resolution of the image. $[c_x, c_y]$

(d) When the camera is moved. $\mathbf{t}$

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

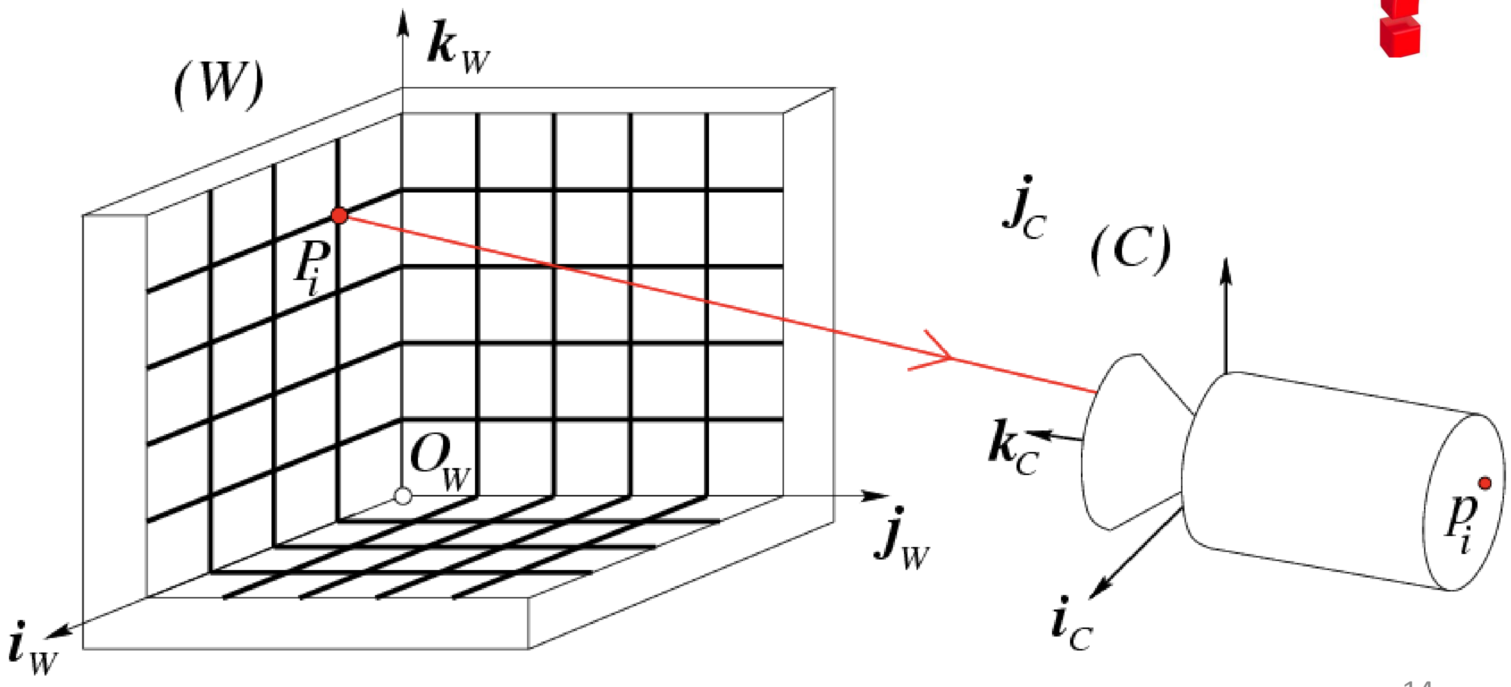
Today's Agenda

- Review Camera calibration
- Epipolar geometry

Recovering 3D Geometry

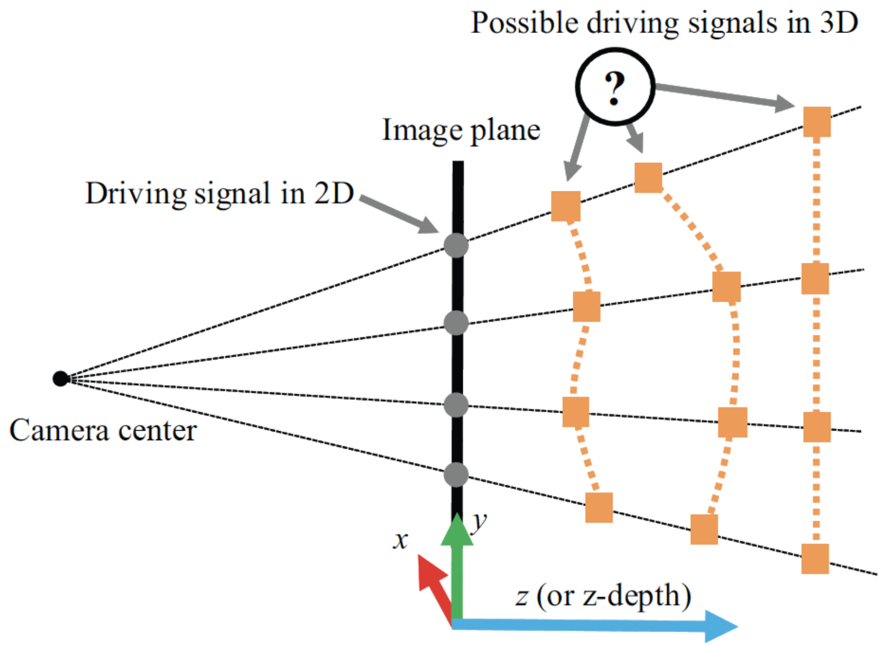
- Camera calibration from a single view
  – Camera intrinsic parameters
  – Camera orientation
  – Camera translation

Sufficient to recover some 3D geometry from a single image?



Recovering 3D Geometry

- Camera calibration from a single view
- Recover 3D geometry from a single view?
  - No: due to ambiguity of 3D -> 2D mapping

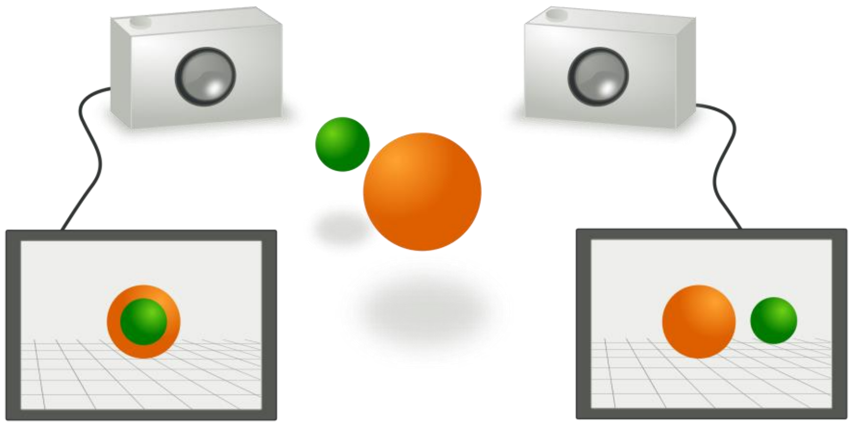


Core Problems in Recovering 3D Geometry

- **Image correspondences**: find the corresponding points in two or more images – **code, lab exercise**
- **Calibration**: given corresponding points in images, recover the relation of the cameras. **Epipolar Geometry**
- **Recover scene geometry**: find coordinates of 3D point from its projections onto 2 or multiple images – **next lecture**

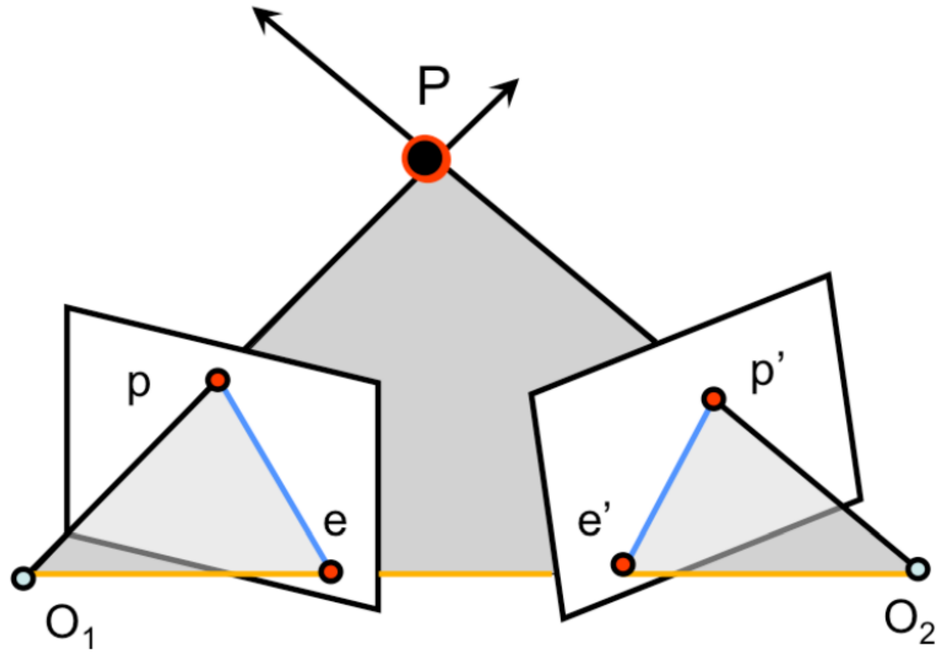
Epipolar Geometry

- The geometry of stereo vision
  - Camera model relates 3D points and corresponding images points
    - Geometric relations between the corresponding image points
      - Define constraints between the image points



Epipolar Geometry

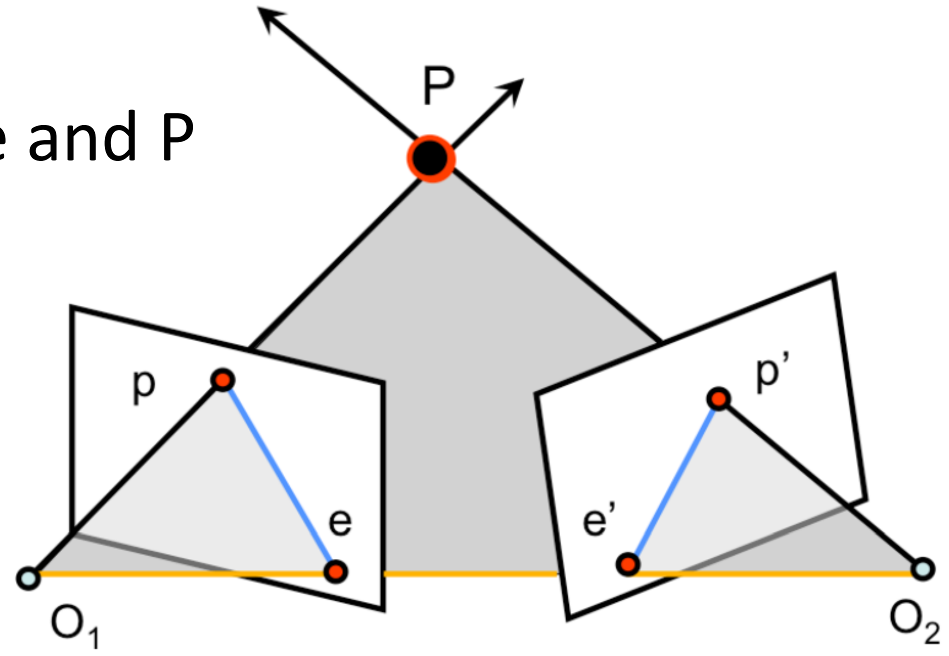
- Baseline
  - The line between the two camera centers O_1 and O_2



The general setup of epipolar geometry

Epipolar Geometry

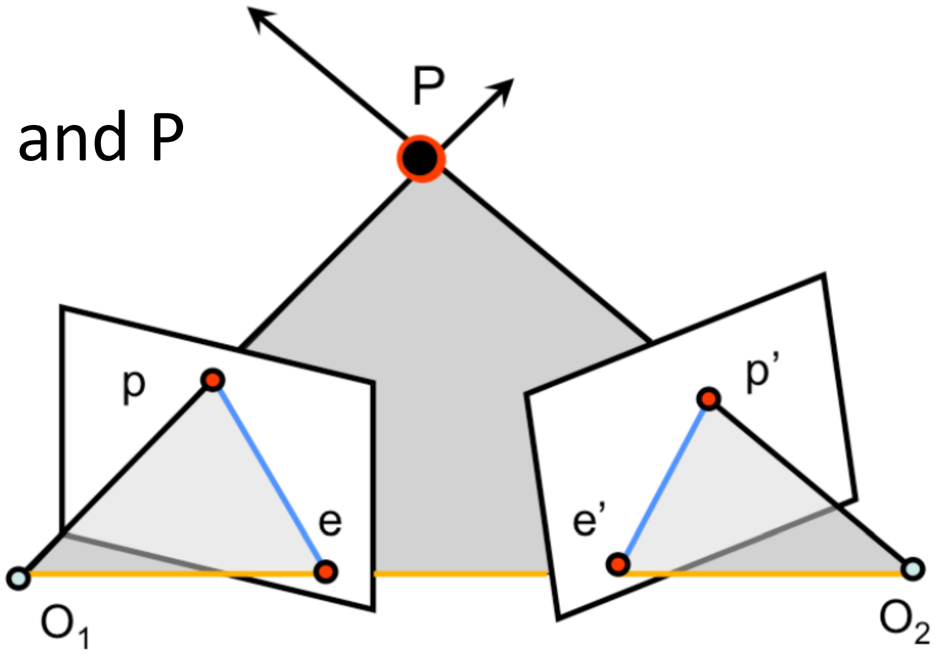
- Baseline
  - The line between the two camera centers O_1 and O_2
- Epipolar plane
  - Defined by P, O_1, and O_2; contains baseline and P



The general setup of epipolar geometry

Epipolar Geometry

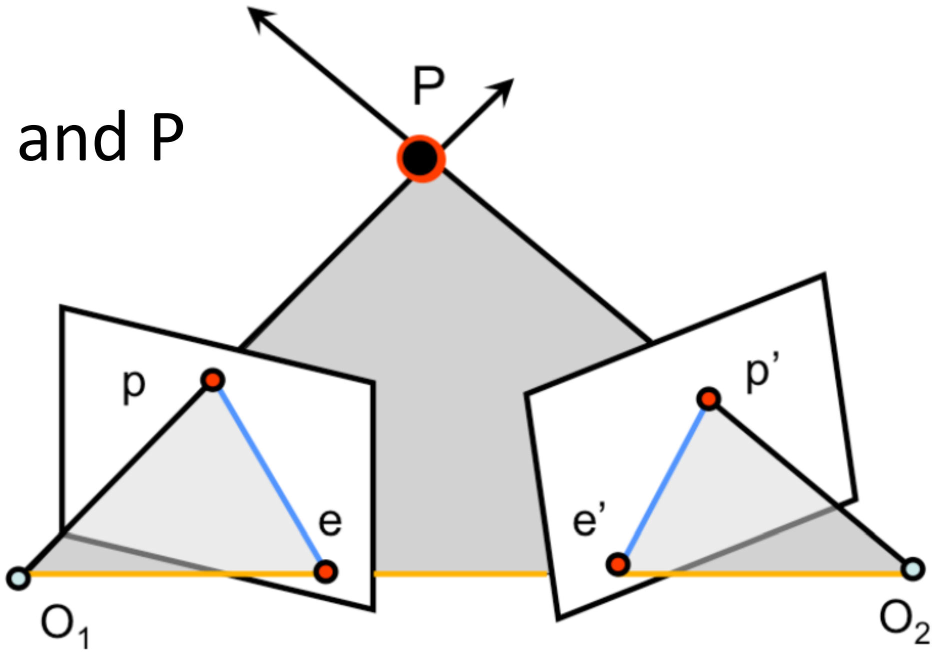
- Baseline
  - The line between the two camera centers O_1 and O_2
- Epipolar plane
  - Defined by P, O_1, and O_2; contains baseline and P
- Epipoles
  - ∩ of baseline and image plane
  - Projection of the other camera center



The general setup of epipolar geometry

Epipolar Geometry

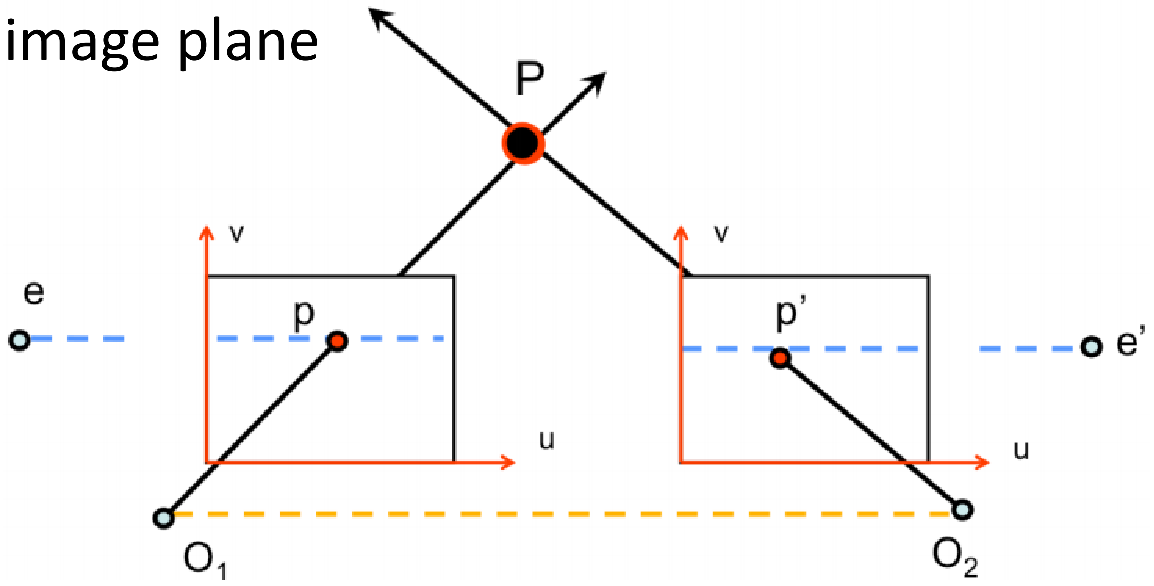
- Baseline
  - The line between the two camera centers O_1 and O_2
- Epipolar plane
  - Defined by P, O_1, and O_2; contains baseline and P
- Epipoles
  - ∩ of baseline and image plane
  - Projection of the other camera center
- Epipolar lines
  - ∩ of epipolar plane with the image plane



The general setup of epipolar geometry

Epipolar Geometry

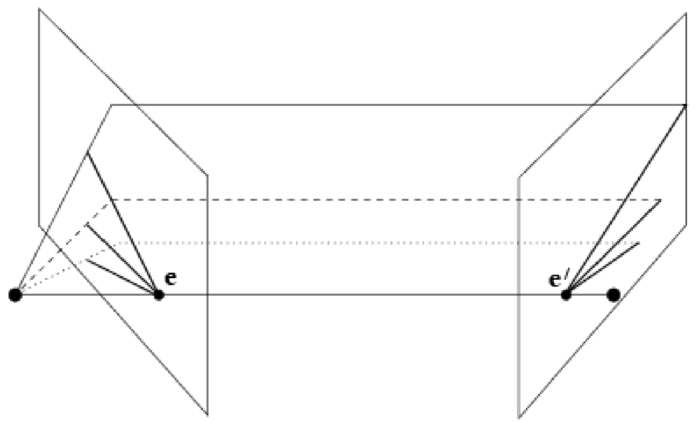
- Examples
  - Parallel image planes (a special case)
    - Baseline is parallel to the image plane
    - Baseline intersects the image plane at infinity → epipoles are at infinity
    - Epipolar lines are parallel to U-axis of image plane

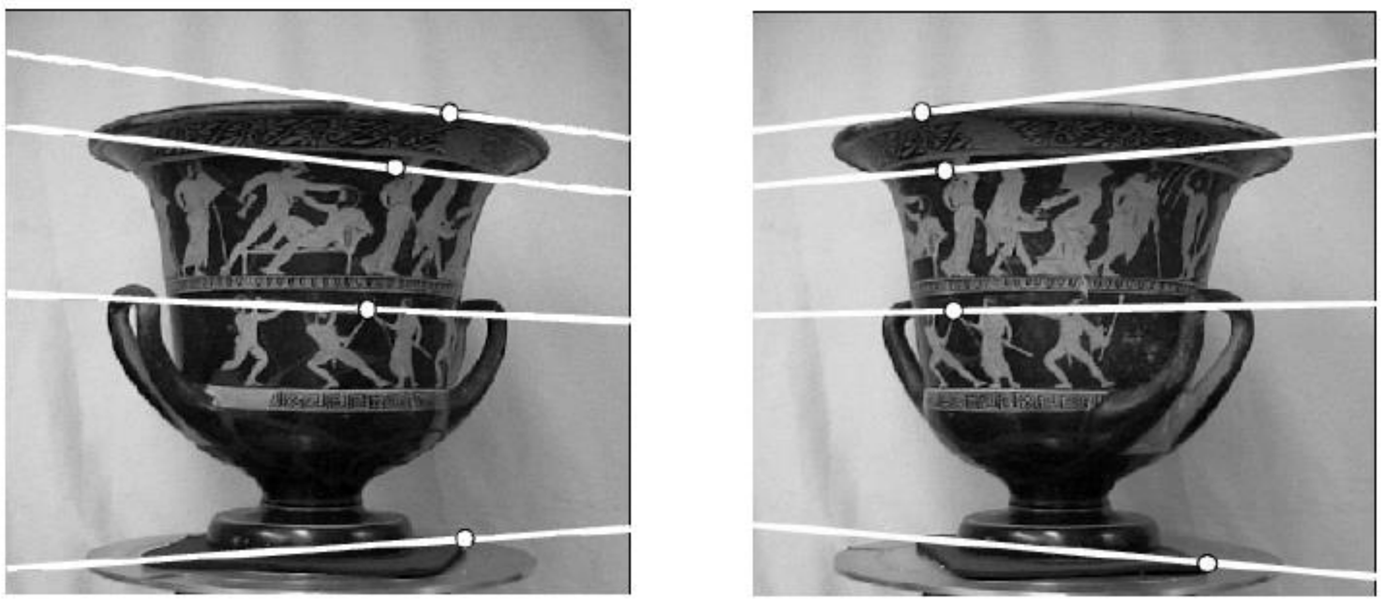


Epipolar Geometry

- Examples
  - Converging image planes (most common case)
    - All epipolar lines intersect at the epipole

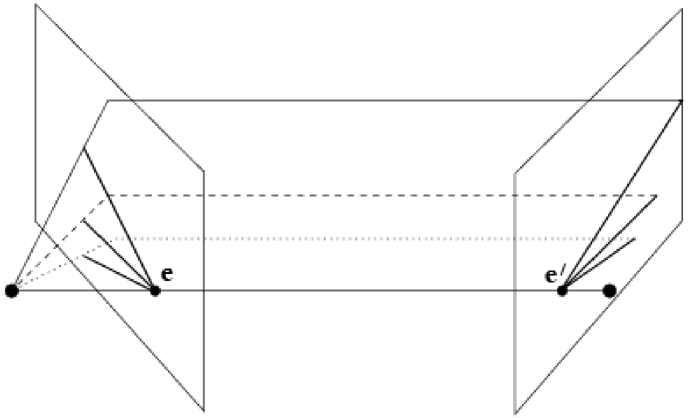
- all epipolar lines lie on epipolar planes
- all epipolar planes intersect at baseline
- base line intersect the image plane at epipole

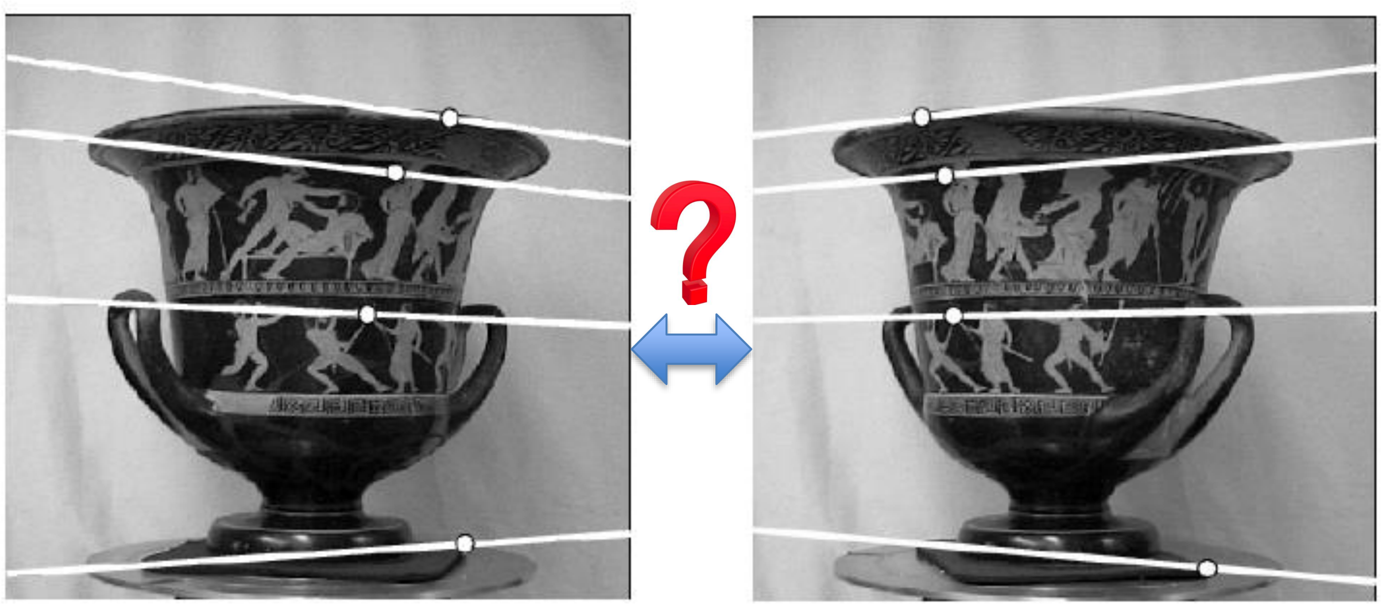




Epipolar Geometry

- The relations between different views?
  - How to use for recovering 3D geometry?
    - Unknown: 3D points
    - Known: image points; camera parameters (from camera calibration)





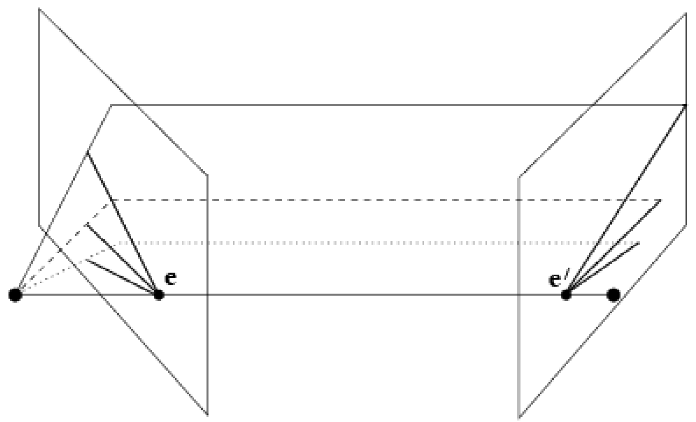
Epipolar Geometry

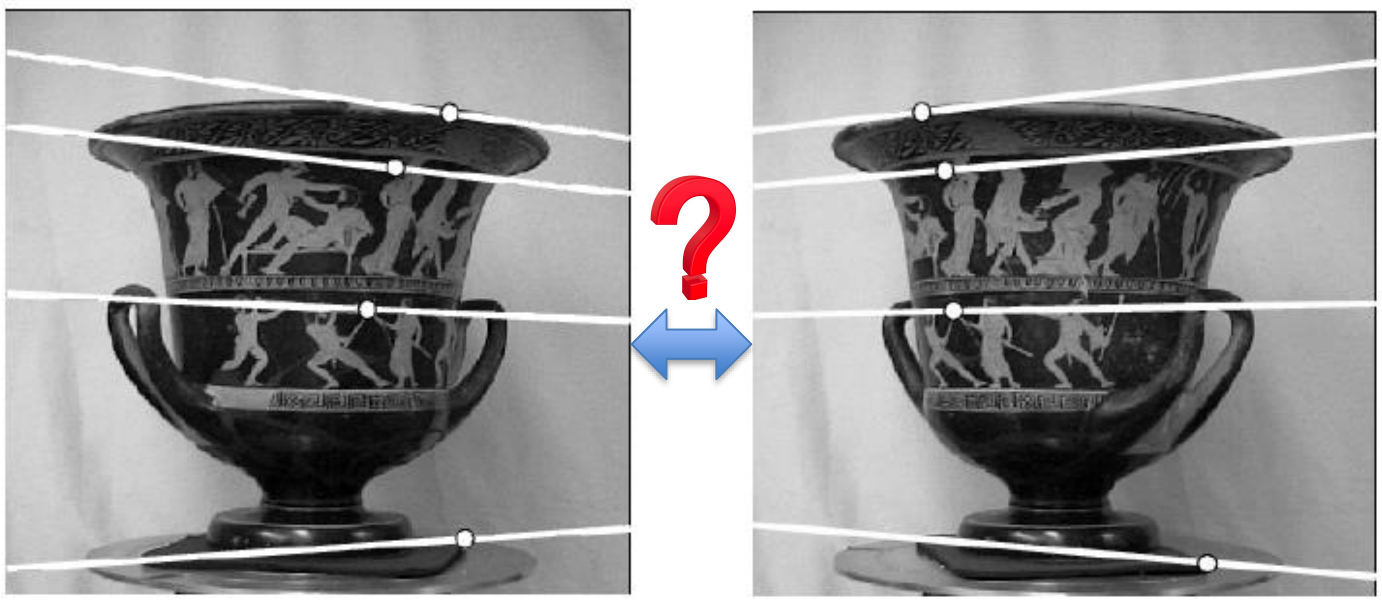
- Constraints between images (without knowing 3D geometry)

  O_1, O_2, image point → epipolar plane → epipolar line (no known 3D)

  – Epipolar line determined by just camera centers and point in one image

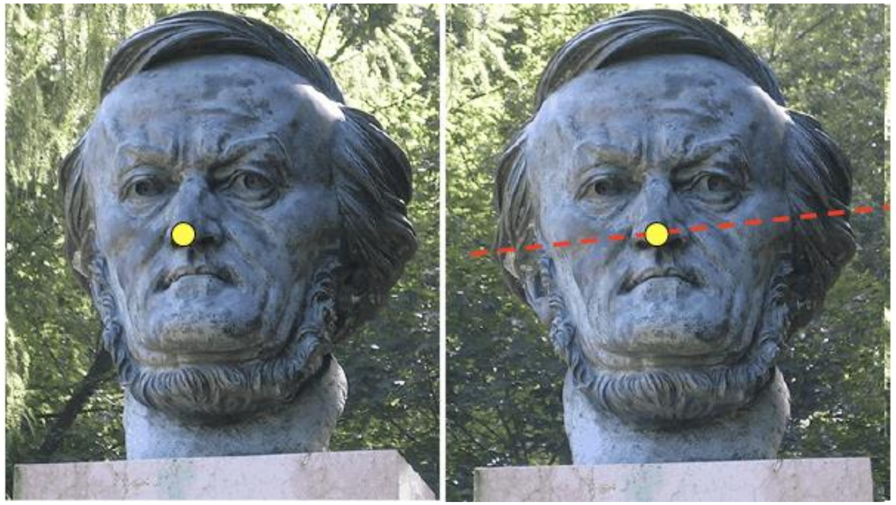
  – The image point on the second image must be on its epipolar line

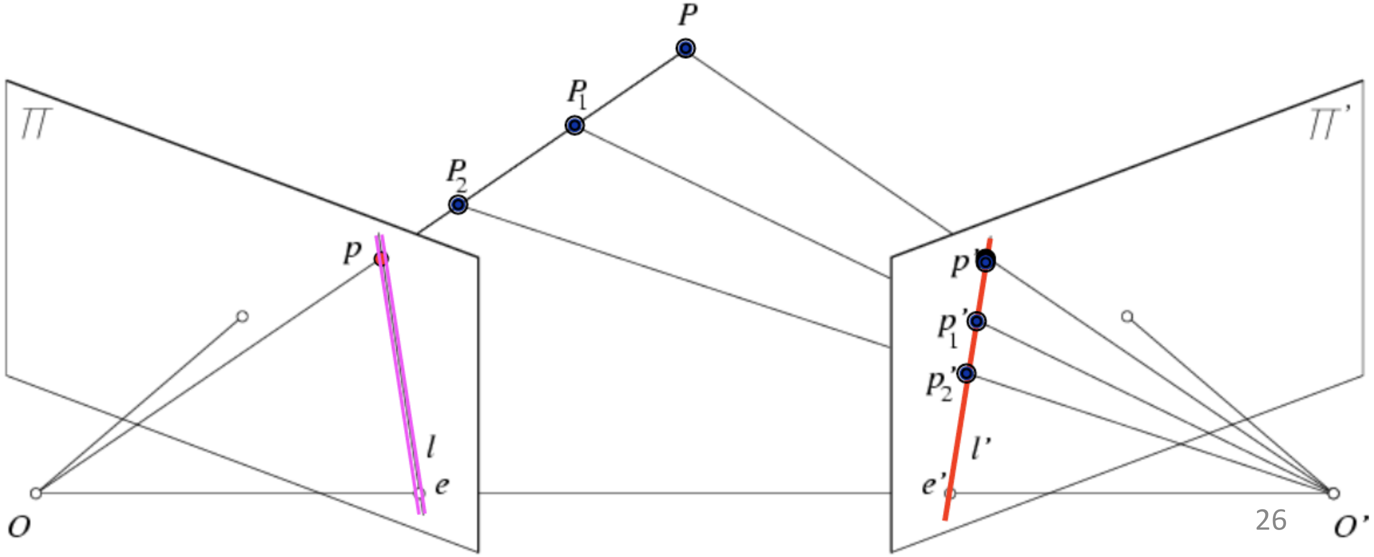




Epipolar Constraint

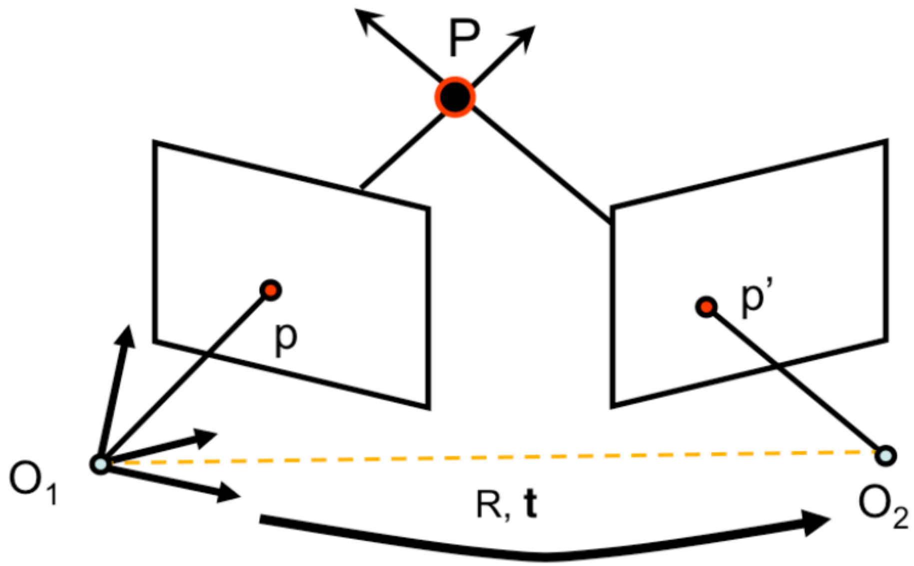
- Given a point on left image, what are the potential locations of the corresponding point on right image?
  – have to lie on the corresponding epipolar line of the other image
- Model of the relation between corresponding image points





Epipolar Constraint

- The relationship between corresponding image points
  - The world reference system aligned with the left camera
  - The right camera has orientation R and offset $\mathbf{t}$



Camera projection matrices

Left camera

$$M = K[I \;\; 0]$$

$$\mathbf{p} = M\mathbf{P} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Right camera

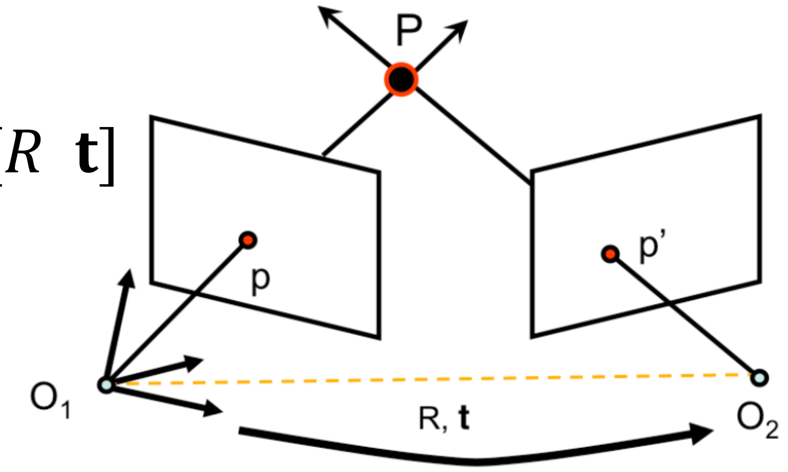
$$M' = K'[R \;\; \mathbf{t}]$$

$$\mathbf{p}' = M'\mathbf{P} = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$$

Epipolar Constraint

- The relationship between corresponding image points
  - Canonical cameras ($K = K' = I$)

$$M = K[I \;\; 0] \rightarrow M = [I \;\; 0] \qquad M' = K'[R \;\; \mathbf{t}] \rightarrow M' = [R \;\; \mathbf{t}]$$

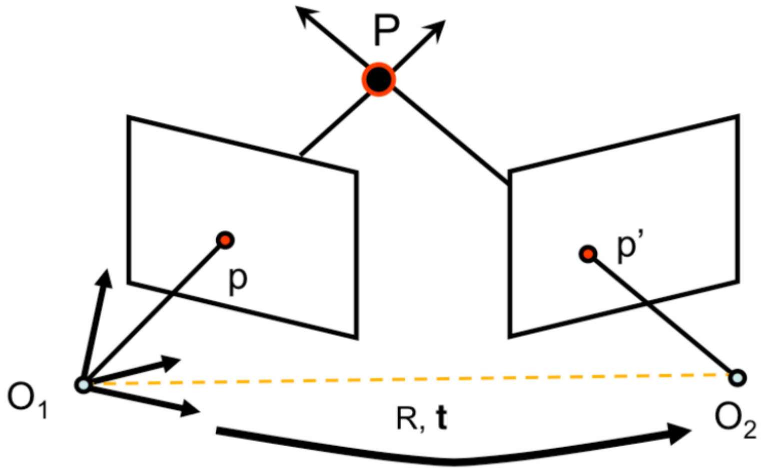


p' in camera 1's coordinate system

Epipolar Constraint

- The relationship between corresponding image points
  - Canonical cameras ($K = K' = I$)

    $M = [I \;\; 0] \qquad M' = [R \;\; \mathbf{t}]$



p' in camera 1's coordinate system $\qquad R^T(\mathbf{p}' - \mathbf{t})$

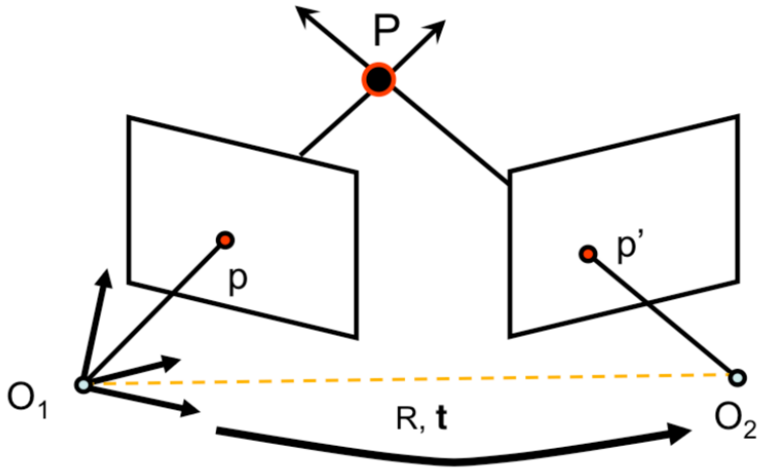
O_2 in camera 1's coordinate system

Epipolar Constraint

- The relationship between corresponding image points
  – Canonical cameras ($K = K' = I$)

$$M = [I \;\; 0] \qquad M' = [R \;\; \mathbf{t}]$$

Type equation here.



p' in camera 1's coordinate system $\quad R^T(\mathbf{p}' - \mathbf{t})$

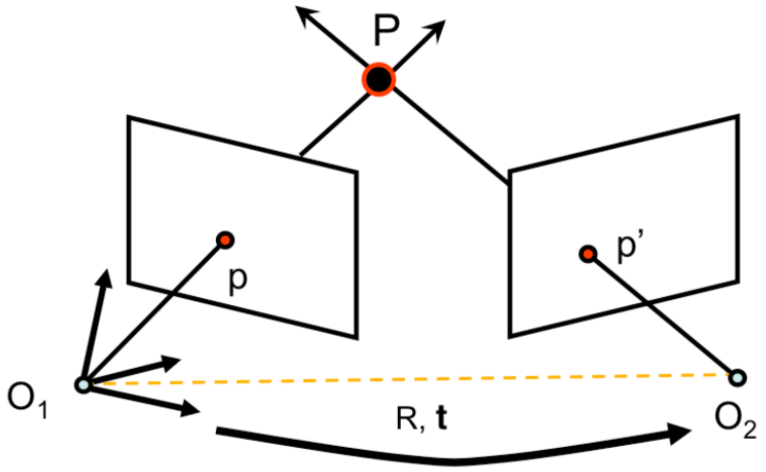
O_2 in camera 1's coordinate system $\quad R^T(\mathbf{O_2} - \mathbf{t}) = -R^T\mathbf{t}$

Normal of the epipolar plane

Epipolar Constraint

- The relationship between corresponding image points
  - Canonical cameras ($K = K' = I$)

$$M = [I \;\; 0] \qquad M' = [R \;\; \mathbf{t}]$$



p' in camera 1's coordinate system $\quad R^T(\mathbf{p}' - \mathbf{t})$

O$_2$ in camera 1's coordinate system $\quad R^T(\mathbf{O_2} - \mathbf{t}) = -R^T\mathbf{t}$

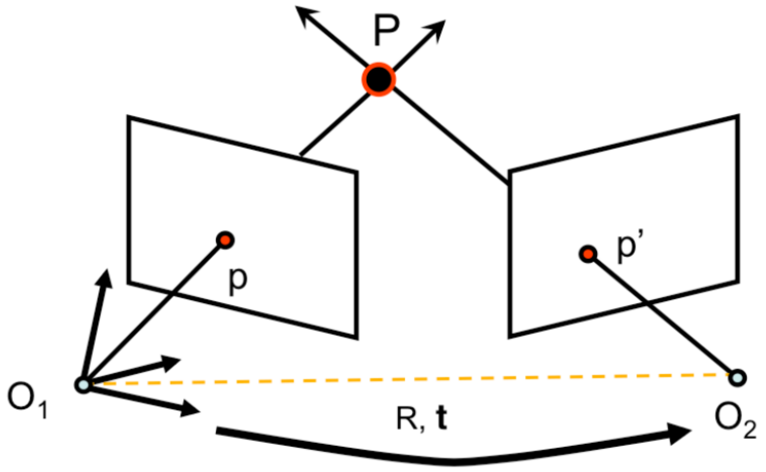
Normal of the epipolar plane $\quad R^T\mathbf{t} \times [R^T(\mathbf{p}' - \mathbf{t})] = R^T(\mathbf{t} \times \mathbf{p}')$

O$_1$p lies in the epipolar plane

Epipolar Constraint

- The relationship between corresponding image points
  - Canonical cameras ($K = K' = I$)

    $M = [I \;\; 0] \qquad M' = [R \;\; \mathbf{t}]$



p' in camera 1's coordinate system $\quad R^T(\mathbf{p}' - \mathbf{t})$

$\mathbf{O_2}$ in camera 1's coordinate system $\quad R^T(\mathbf{O_2} - \mathbf{t}) = -R^T\mathbf{t}$

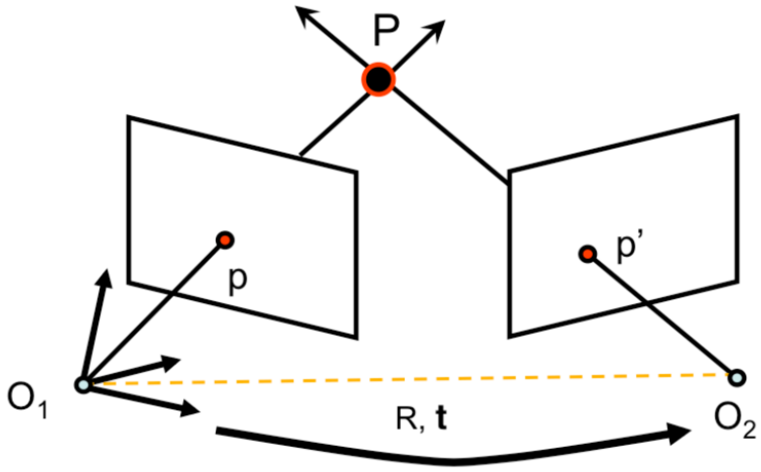
Normal of the epipolar plane $\quad R^T\mathbf{t} \times [R^T(\mathbf{p}' - \mathbf{t})] = R^T(\mathbf{t} \times \mathbf{p}')$

$\mathbf{O_1 p}$ lies in the epipolar plane $\quad [R^T(\mathbf{t} \times \mathbf{p}')]^T\mathbf{p} = 0$

Epipolar Constraint

- The relationship between corresponding image points
  - Canonical cameras ($K = K' = I$)

$$M = [I \;\; 0] \qquad M' = [R \;\; \mathbf{t}]$$



p' in camera 1's coordinate system $\quad R^T(\mathbf{p}' - \mathbf{t})$

$\mathbf{O_2}$ in camera 1's coordinate system $\quad R^T(\mathbf{O_2} - \mathbf{t}) = -R^T\mathbf{t}$

Normal of the epipolar plane $\quad R^T\mathbf{t} \times [R^T(\mathbf{p}' - \mathbf{t})] = R^T(\mathbf{t} \times \mathbf{p}')$

$\mathbf{O_1p}$ lies in the epipolar plane $\quad [R^T(\mathbf{t} \times \mathbf{p}')]^T\mathbf{p} = 0 \Rightarrow (\mathbf{t} \times \mathbf{p}')^T R\mathbf{p} = 0$

Epipolar Constraint

- The relationship between corresponding image points
  - Canonical cameras ($K = K' = I$)

$$M = [I \;\; 0] \qquad M' = [R \;\; \mathbf{t}]$$

Cross product as matrix-vector multiplication

$$\boxed{\mathbf{a} \times \mathbf{b} = \begin{bmatrix} 0 & -\mathbf{a}_z & \mathbf{a}_y \\ \mathbf{a}_z & 0 & -\mathbf{a}_x \\ -\mathbf{a}_y & \mathbf{a}_x & 0 \end{bmatrix} \begin{bmatrix} \mathbf{b}_x \\ \mathbf{b}_y \\ \mathbf{b}_z \end{bmatrix} = [\mathbf{a}_\times]\mathbf{b}}$$

$[\mathbf{a}_\times]^T = -[\mathbf{a}_\times]$

p' in camera 1's coordinate system $\quad R^T(\mathbf{p}' - \mathbf{t})$

O$_2$ in camera 1's coordinate system $\quad R^T(\mathbf{O}_2 - \mathbf{t}) = -R^T\mathbf{t}$

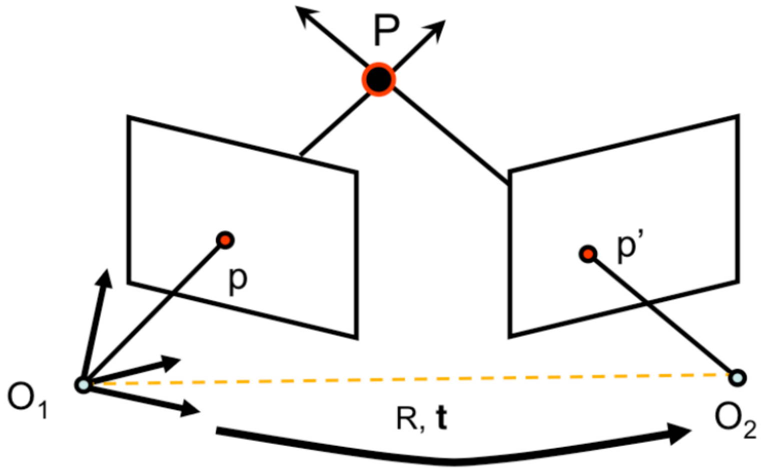
Normal of the epipolar plane $\quad R^T\mathbf{t} \times [R^T(\mathbf{p}' - \mathbf{t})] = R^T(\mathbf{t} \times \mathbf{p}')$

O$_1$p lies in the epipolar plane $\quad [R^T(\mathbf{t} \times \mathbf{p}')]^T\mathbf{p} = 0 \Rightarrow (\mathbf{t} \times \mathbf{p}')^T R\mathbf{p} = 0 \Rightarrow ([\mathbf{t}_\times]\,\mathbf{p}')^T R\mathbf{p} = 0$

Epipolar Constraint

- The relationship between corresponding image points
  - Canonical cameras ($K = K' = I$)

$$M = [I \;\; 0] \qquad M' = [R \;\; \mathbf{t}]$$



p' in camera 1's coordinate system $\quad R^T(\mathbf{p}' - \mathbf{t})$

$\mathbf{O_2}$ in camera 1's coordinate system $\quad R^T(\mathbf{O_2} - \mathbf{t}) = -R^T\mathbf{t}$

Normal of the epipolar plane $\quad R^T\mathbf{t} \times [R^T(\mathbf{p}' - \mathbf{t})] = R^T(\mathbf{t} \times \mathbf{p}')$

$\mathbf{O_1 p}$ lies in the epipolar plane $\quad [R^T(\mathbf{t} \times \mathbf{p}')]^T\mathbf{p} = 0 \Rightarrow (\mathbf{t} \times \mathbf{p}')^T R\mathbf{p} = 0 \Rightarrow ([\mathbf{t}_\times]\,\mathbf{p}')^T R\mathbf{p} = 0$

$$\Rightarrow \mathbf{p}'^T[\mathbf{t}_\times]R\mathbf{p} = 0$$

Epipolar Constraint

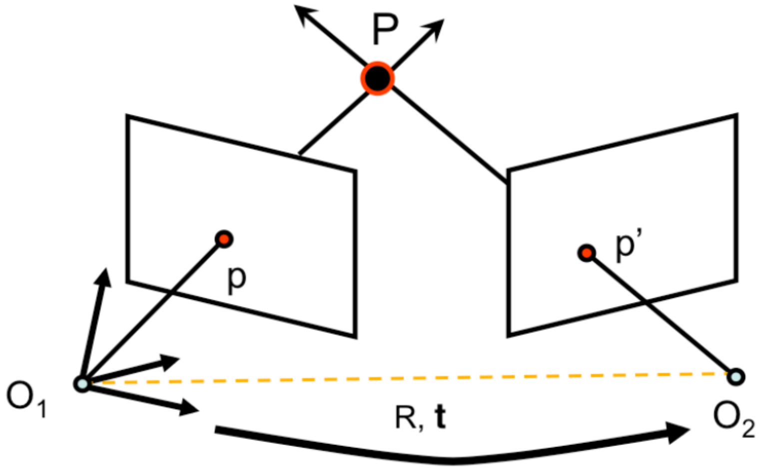
- Essential matrix
  - Establish constraints between matching image points
  - Determine relative position and orientation of two cameras
  - 5 degrees of freedom (R: 3, $\mathbf{t}$: 3, but scale is not known)

$$\mathbf{p}'^T \boxed{[\mathbf{t}_\times]R}\mathbf{p} = 0$$

$$E = [\mathbf{t}_\times]R$$

$$\mathbf{p}'^T \boxed{E}\mathbf{p} = 0$$

Essential matrix



Epipolar Constraint

- How to generalize Essential matrix?
  - Canonical cameras → general cameras

$K = K' = I$

$M = K[I \;\; 0]$

$M' = K'[R \;\; \mathbf{t}]$

$K \neq I, K' \neq I$

$M = [I \;\; 0]$

$M' = [R \;\; \mathbf{t}]$

$\mathbf{p} = M\mathbf{P} = [I \;\; 0]\mathbf{P}$

$\mathbf{p}' = M'\mathbf{P} = [R \;\; \mathbf{t}]\mathbf{P}$

$\mathbf{p}'^T E \mathbf{p} = 0$

$E = [\mathbf{t}_{\times}]R$

?

Epipolar Constraint

- How to generalize Essential matrix?
  - Canonical cameras → general cameras

$$M = K[I \;\; 0]$$
$$M' = K'[R \;\; \mathbf{t}]$$

$K = K' = I$

$$M = [I \;\; 0] \qquad \mathbf{p} = M\mathbf{P} = [I \;\; 0]\mathbf{P} \qquad \mathbf{p}'^T E \mathbf{p} = 0$$
$$M' = [R \;\; \mathbf{t}] \qquad \mathbf{p}' = M'\mathbf{P} = [R \;\; \mathbf{t}]\mathbf{P} \qquad E = [\mathbf{t}_{\times}]R$$

$K \neq I, K' \neq I$

$$\mathbf{p} = M\mathbf{P} = K[I \;\; 0]\mathbf{P}$$
$$\mathbf{p}' = M'\mathbf{P} = K'[R \;\; \mathbf{t}]\mathbf{P}$$

$\mathbf{p} \rightarrow K^{-1}\mathbf{p}$
$\mathbf{p}' \rightarrow K'^{-1}\mathbf{p}'$

$$\mathbf{p}'^T F \mathbf{p} = 0$$
$$F = K'^{-T}[\mathbf{t}_{\times}]RK^{-1}$$

Epipolar Constraint

- Essential matrix vs. Fundamental matrix
  - Similarity
    - Both relate the matching image points
      - Encode epipolar geometry of two views & camera parameters
  - Differences
    - E encodes only the camera extrinsic parameter
    - F also encodes the intrinsic parameters

$$\mathbf{p}'^T E \mathbf{p} = 0$$

$$E = [\mathbf{t}_{\times}]R$$

Essential matrix

$$\mathbf{p}'^T F \mathbf{p} = 0$$

$$F = K'^{-T}[\mathbf{t}_{\times}]RK^{-1}$$

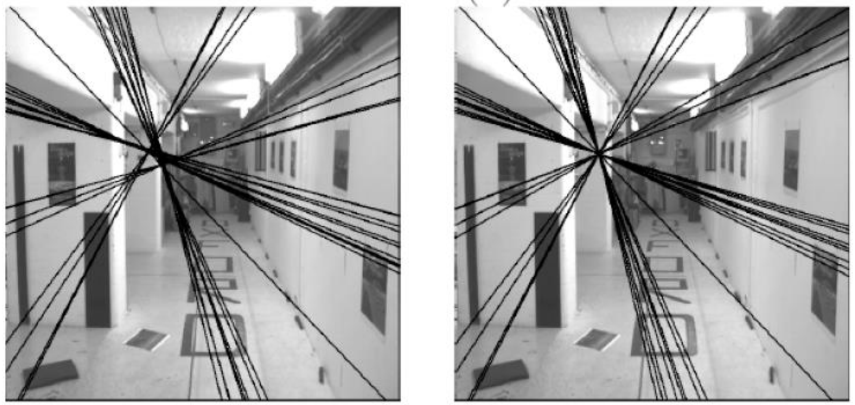
Fundamental matrix

Epipolar Constraint

- Properties of the Fundamental matrix
  - 3 x 3
  - homogeneous (has scale ambiguity)
  - rank(F) = 2
    - The potential matching point is located on a line
  - F has 7 degrees of freedom

$$\mathbf{p}'^{T} F \mathbf{p} = 0 \qquad F = K'^{-T}[\mathbf{t}_{\times}]RK^{-1}$$

Fundamental matrix has rank 2 : $\det(\mathtt{F}) = 0$.

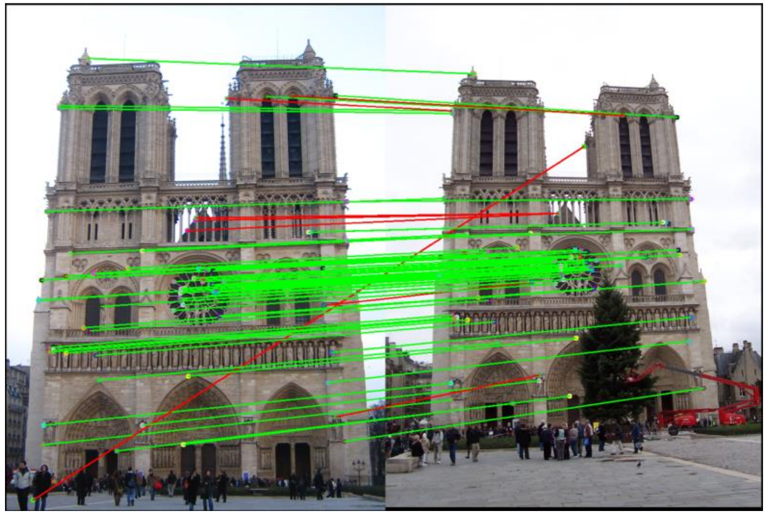


Left : Uncorrected $\mathtt{F}$ – epipolar lines are not coincident.

Right : Epipolar lines from corrected $\mathtt{F}$.

Epipolar Constraint

- Properties of the Fundamental matrix
- How is the fundamental matrix useful?
  - A 3D point's image in one image -> the epipolar line in the other image
    - Without knowing 3D location, camera intrinsic and extrinsic parameters
  - Powerful tool
    - Establishing reliable correspondences
    - Multi-view object/scene matching
    - Multi-view camera calibration

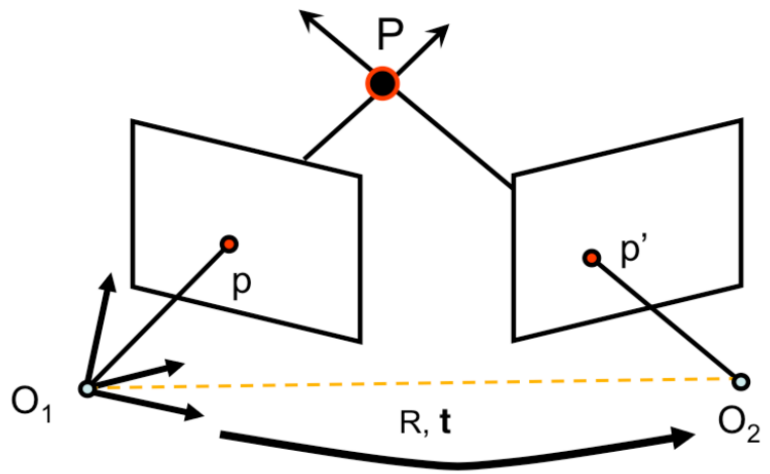


$$\mathbf{p}'^T F \mathbf{p} = 0 \qquad F = K'^{-T}[\mathbf{t}_\times]RK^{-1}$$

Recovering Fundamental Matrix

- How to recover F?
  - From image correspondences

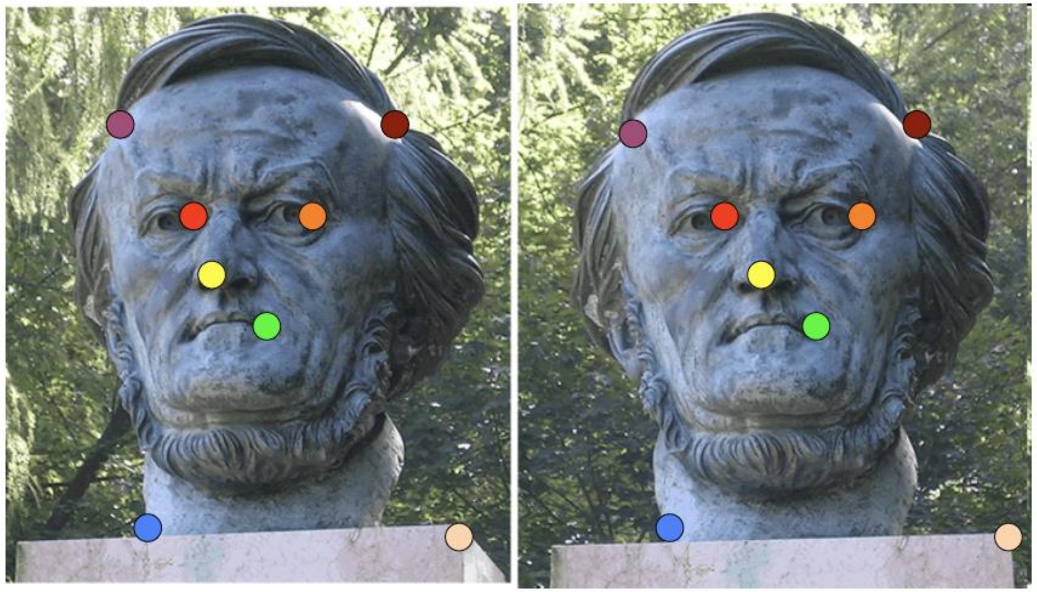
$\mathbf{p}'^{T}F\mathbf{p} = 0$ $\quad F = K'^{-T}[\mathbf{t}_{\times}]RK^{-1}$



Recovering Fundamental Matrix

- How to recover F?
  - From image correspondences
  - How many point pairs needed?

$$\mathbf{p}'^T F \mathbf{p} = 0 \qquad F = K'^{-T}[\mathbf{t}_\times] R K^{-1}$$



Recovering Fundamental Matrix

- How to recover F?
  - From image correspondences
  - 8-point pairs required
    - Each point pair gives one equation
    - F is known up to scale
      - The linear system is homogeneous

$$\begin{cases} \mathbf{p}_i = (u_i, v_i, 1) \\ \boldsymbol{p}'_i = (u'_i, v'_i, 1) \end{cases} \qquad \mathbf{p}'^T F\, \mathbf{p} = 0$$

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

Recovering Fundamental Matrix

- How to recover F?
  - From image correspondences
  - 8-point pairs required
    - Each point pair gives one equation
    - F is known up to scale
      - The linear system is homogeneous

$$\begin{cases} \mathbf{p}_i = (u_i, v_i, 1) \\ p'_i = (u'_i, v'_i, 1) \end{cases} \qquad \mathbf{p}'^T F\, \mathbf{p} = 0$$

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

F has 7 degrees of freedom

Are 7-point pairs sufficient?

Recovering Fundamental Matrix

- 8-point algorithm

$$\begin{bmatrix} u_i u_i' & v_i u_i' & u_i' & u_i v_i' & v_i v_i' & v_i' & u_i & v_i & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

$$\boxed{\begin{bmatrix} u_1 u_1' & v_1 u_1' & u_1' & u_1 v_1' & v_1 v_1' & v_1' & u_1 & v_1 & 1 \\ u_2 u_2' & v_2 u_2' & u_2' & u_2 v_2' & v_2 v_2' & v_2' & u_2 & v_2 & 1 \\ u_3 u_3' & v_3 u_3' & u_3' & u_3 v_3' & v_3 v_3' & v_3' & u_3 & v_3 & 1 \\ u_4 u_4' & v_4 u_4' & u_4' & u_4 v_4' & v_4 v_4' & v_4' & u_4 & v_4 & 1 \\ u_5 u_5' & v_5 u_5' & u_5' & u_5 v_5' & v_5 v_5' & v_5' & u_5 & v_5 & 1 \\ u_6 u_6' & v_6 u_6' & u_6' & u_6 v_6' & v_6 v_6' & v_6' & u_6 & v_6 & 1 \\ u_7 u_7' & v_7 u_7' & u_7' & u_7 v_7' & v_7 v_7' & v_7' & u_7 & v_7 & 1 \\ u_8 u_8' & v_8 u_8' & u_8' & u_8 v_8' & v_8 v_8' & v_8' & u_8 & v_8 & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0}$$

Recovering Fundamental Matrix

- 8-point algorithm
  - Construct linear system using corresponding image points

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

How to solve it?

$$\begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\ u_3u_3' & v_3u_3' & u_3' & u_3v_3' & v_3v_3' & v_3' & u_3 & v_3 & 1 \\ u_4u_4' & v_4u_4' & u_4' & u_4v_4' & v_4v_4' & v_4' & u_4 & v_4 & 1 \\ u_5u_5' & v_5u_5' & u_5' & u_5v_5' & v_5v_5' & v_5' & u_5 & v_5 & 1 \\ u_6u_6' & v_6u_6' & u_6' & u_6v_6' & v_6v_6' & v_6' & u_6 & v_6 & 1 \\ u_7u_7' & v_7u_7' & u_7' & u_7v_7' & v_7v_7' & v_7' & u_7 & v_7 & 1 \\ u_8u_8' & v_8u_8' & u_8' & u_8v_8' & v_8v_8' & v_8' & u_8 & v_8 & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

Recovering Fundamental Matrix

- 8-point algorithm
  - Construct linear system using corresponding image points
  - Solve for **f** using SVD

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

$$W = USV^T$$

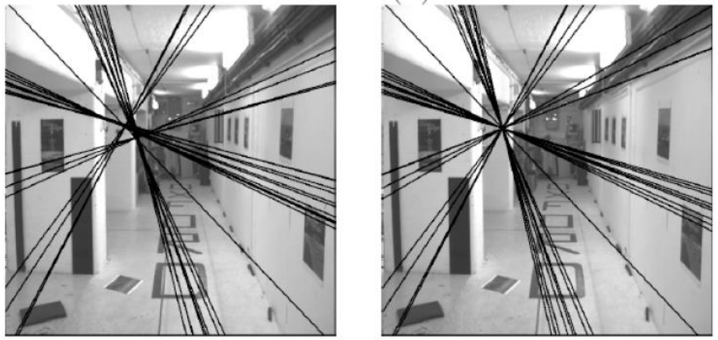
Last column of V gives **f**

$$\begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\ u_3u_3' & v_3u_3' & u_3' & u_3v_3' & v_3v_3' & v_3' & u_3 & v_3 & 1 \\ u_4u_4' & v_4u_4' & u_4' & u_4v_4' & v_4v_4' & v_4' & u_4 & v_4 & 1 \\ u_5u_5' & v_5u_5' & u_5' & u_5v_5' & v_5v_5' & v_5' & u_5 & v_5 & 1 \\ u_6u_6' & v_6u_6' & u_6' & u_6v_6' & v_6v_6' & v_6' & u_6 & v_6 & 1 \\ u_7u_7' & v_7u_7' & u_7' & u_7v_7' & v_7v_7' & v_7' & u_7 & v_7 & 1 \\ u_8u_8' & v_8u_8' & u_8' & u_8v_8' & v_8v_8' & v_8' & u_8 & v_8 & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

Recovering Fundamental Matrix

- 8-point algorithm
  - Construct linear system using corresponding image points
  - Solve for $\mathbf{f}$ using SVD
  - Constraint enforcement (essential step)
    - rank(F) = 2

Fundamental matrix has rank 2 : $\det(\mathtt{F}) = 0$.



Left : Uncorrected $\mathtt{F}$ – epipolar lines are not coincident.

Right : Epipolar lines from corrected $\mathtt{F}$.

Recovering Fundamental Matrix

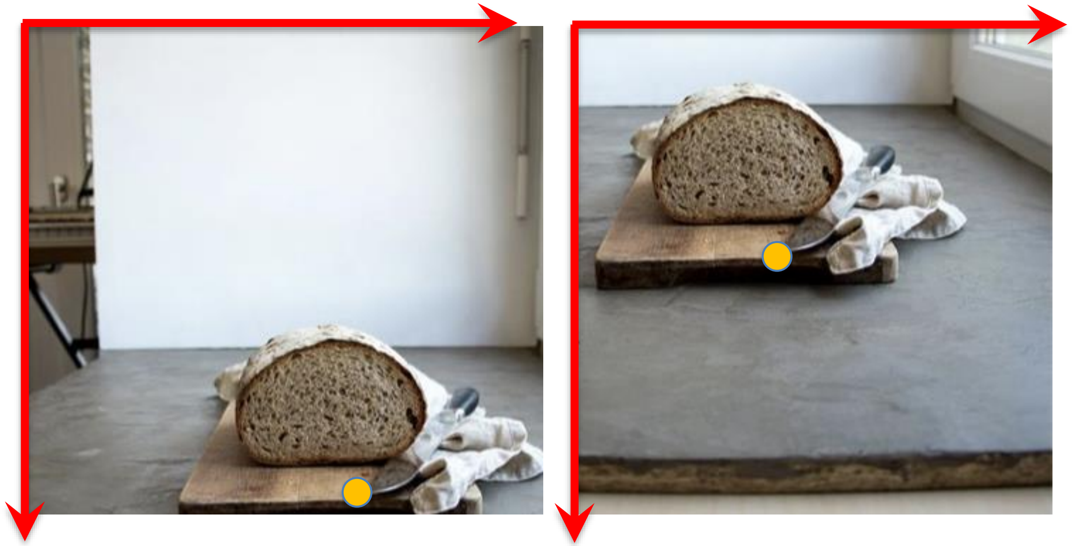
- 8-point algorithm
  - – Construct linear system using corresponding image points
  - – Solve for $\mathbf{f}$ using SVD
  - – Constraint enforcement (essential step)
    - rank(F) = 2

$$\hat{F} = UDV^T \qquad D = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} \Rightarrow F = U \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T$$

Recovering Fundamental Matrix

- Problems of 8-point algorithm
  - Sensitive to the origin of coordinates
  - Sensitive to scales

(568, 723) (284, 366)



Same scale, different origins (i.e., camera translation)

Image taken using different focal lengths

Recovering Fundamental Matrix

- Problems of 8-point algorithm
  - Sensitive to the origin of coordinates
  - Sensitive to scales

$$\begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\ u_3u_3' & v_3u_3' & u_3' & u_3v_3' & v_3v_3' & v_3' & u_3 & v_3 & 1 \\ u_4u_4' & v_4u_4' & u_4' & u_4v_4' & v_4v_4' & v_4' & u_4 & v_4 & 1 \\ u_5u_5' & v_5u_5' & u_5' & u_5v_5' & v_5v_5' & v_5' & u_5 & v_5 & 1 \\ u_6u_6' & v_6u_6' & u_6' & u_6v_6' & v_6v_6' & v_6' & u_6 & v_6 & 1 \\ u_7u_7' & v_7u_7' & u_7' & u_7v_7' & v_7v_7' & v_7' & u_7 & v_7 & 1 \\ u_8u_8' & v_8u_8' & u_8' & u_8v_8' & v_8v_8' & v_8' & u_8 & v_8 & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

$$\begin{bmatrix} 250906.36 & 183269.57 & 921.81 & 200931.10 & 146766.13 & 738.21 & 272.19 & 198.81 & 1 \\ \hline 2692.28 & 131633.03 & 176.27 & 6196.73 & 302975.59 & 405.71 & 15.27 & 746.79 & 1 \\ \hline 416374.23 & 871684.30 & 935.47 & 408110.89 & 854384.92 & 916.90 & 445.10 & 931.81 & 1 \\ \hline 191183.60 & 171759.40 & 410.27 & 416435.62 & 374125.90 & 893.65 & 465.99 & 418.65 & 1 \\ \hline 48988.86 & 30401.76 & 57.89 & 298604.57 & 185309.58 & 352.87 & 846.22 & 525.15 & 1 \\ \hline 164786.04 & 546559.67 & 813.17 & 1998.37 & 6628.15 & 9.86 & 202.65 & 672.14 & 1 \\ \hline 116407.01 & 2727.75 & 138.89 & 169941.27 & 3982.21 & 202.77 & 838.12 & 19.64 & 1 \\ \hline 135384.58 & 75411.13 & 198.72 & 411350.03 & 229127.78 & 603.79 & 681.28 & 379.48 & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

Poor numerical conditioning → fix by scaling the data

Recovering Fundamental Matrix

- ~~8-point algorithm~~
- Normalized 8-point algorithm
  - Idea: normalize before constructing the equations
    - Translation: make centroid of image points at origin ← reduce translation effect
    - Scaling: make average distance of points from origin $\sqrt{2}$ ← reduce scaling effect

$$\mathbf{q}_i = T\mathbf{p}_i$$
$$\mathbf{q}'_i = T'\mathbf{q}'_i$$

Recovering Fundamental Matrix

- ~~8-point algorithm~~
- Normalized 8-point algorithm
  - Normalization of image points (essential step)
  - Solve for F_q using the original 8-point algorithm
    - F_q is the fundamental matrix computed form the normalized image points
    - Same procedure as in original 8-point algorithm

Recovering Fundamental Matrix

- ~~8-point algorithm~~
- Normalized 8-point algorithm
  - Normalization of image points (essential step)
  - Solve for F_q using the original 8-point algorithm
  - De-normalization (essential step)

$$\boxed{\mathbf{q}'}^T F_q \boxed{\mathbf{q}} = 0$$

Normalized image points

$$\mathbf{q} = T\boxed{\mathbf{p}} \qquad \mathbf{q}' = T'\boxed{\mathbf{p}'}$$

Original image points

$$(T'\mathbf{p}')^T F_q (T\mathbf{p}) = 0 \Rightarrow \mathbf{p}'^T \underbrace{\boxed{(T'^T F_q T)}}_{F} \mathbf{p} = 0$$

F_q: fundamental matrix computed form normalized image points

Next lecture

- 2-view 3D reconstruction
  - Camera calibration
  - Triangulation
- Structure from Motion
  - Go beyond two views
  - Simultaneously
    - recover 3D structure
    - Refine camera parameters