Lecture 4
Two View Geometry

Liangliang Nan

Today’s Agenda

- Review of Previous Lecture
  - Camera calibration
- Epipolar geometry

Learning objectives

- Understand and explain the relationship between two cameras

Review of Camera Calibration

- Camera calibration
  - Recovering K
  - Recovering R and T

$$\mathbf{P}' = \mathcal{M}\mathbf{P}_w$$
$$= \mathcal{K}\begin{bmatrix}\mathcal{R} & \mathbf{T}\end{bmatrix}\mathbf{P}_w$$

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

$$\mathbf{P}' = \mathcal{M}\mathbf{P}_w$$
$$= \mathcal{K}\begin{bmatrix}\mathcal{R} & \mathbf{T}\end{bmatrix}\mathbf{P}_w$$

$$K = \begin{bmatrix} \alpha & -\alpha\cot\theta & u_o \\ 0 & \frac{\beta}{\sin\theta} & v_o \\ 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix} \quad T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

Review of Camera Calibration

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

$$\mathbf{p}_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = M\mathbf{P}_i = \begin{bmatrix} \frac{\mathbf{m}_1\mathbf{P}_i}{\mathbf{m}_3\mathbf{P}_i} \\ \frac{\mathbf{m}_2\mathbf{P}_i}{\mathbf{m}_3\mathbf{P}_i} \end{bmatrix}$$

$$u_i(\mathbf{m}_3\mathbf{P}_i) - \mathbf{m}_1\mathbf{P}_i = 0$$
$$v_i(\mathbf{m}_3\mathbf{P}_i) - \mathbf{m}_2\mathbf{P}_i = 0$$

Review of Camera Calibration

- 11 parameters to recover
- Corresponding 3D-2D point pairs
- How to solve it?
  - *m = 0* always a trivial solution
  - *k * m* (*k* is non-zero) 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

- 11 parameters to recover
- Corresponding 3D-2D point pairs
- How to solve it?
  - *m = 0* always a trivial solution
  - *k* * *m* (*k* is non-zero) is also a solution
  - Constrained optimization

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

Review of Camera Calibration

- Solved using SVD

$$Pm = 0$$

SVD decomposition of P

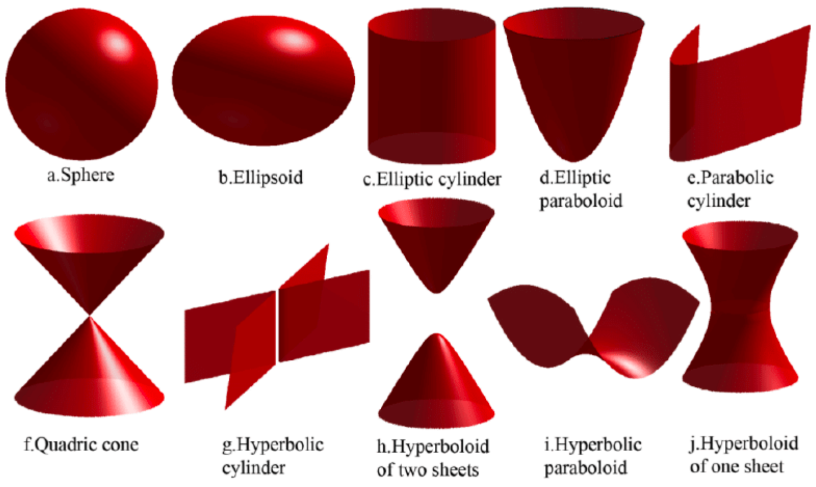
$$U_{2n\times12} \, D_{12\times12} \, V^{T}_{12\times12}$$

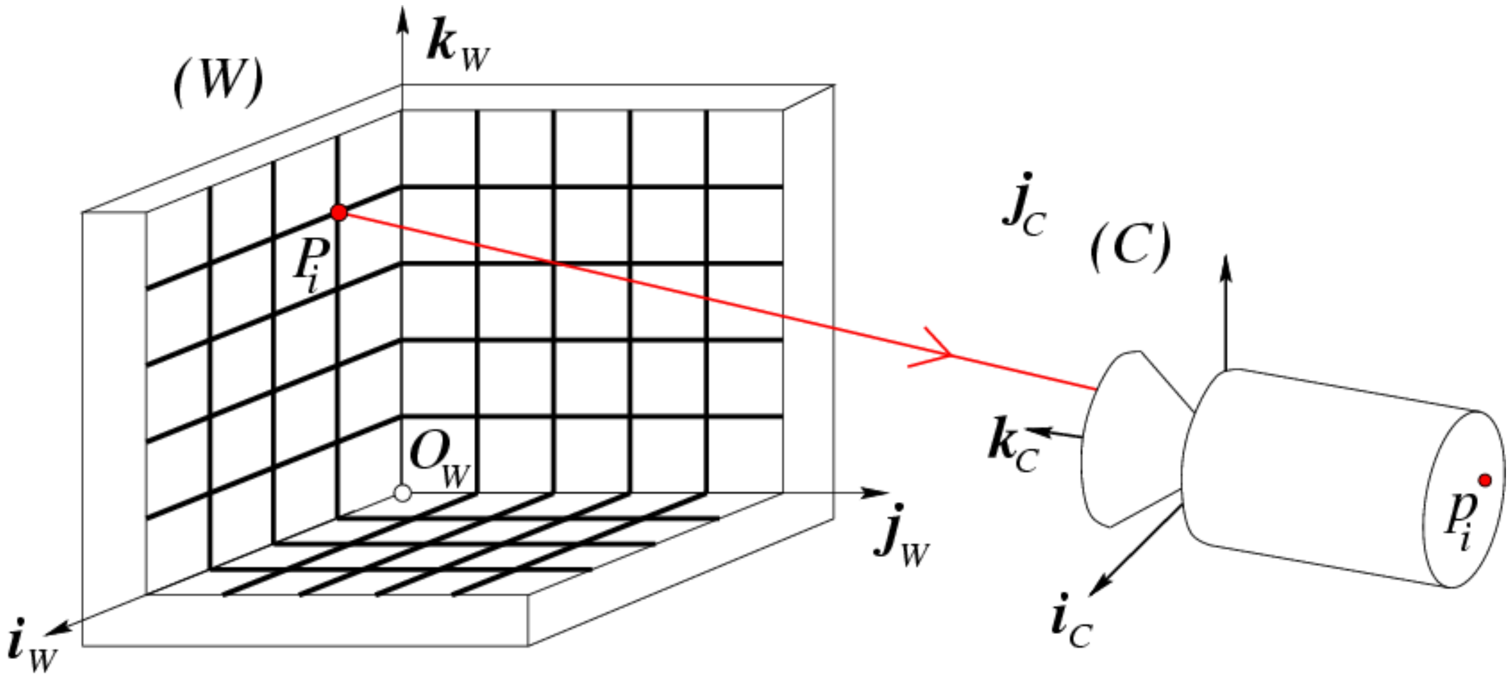
Last column of V gives ***m***

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

Review of Camera Calibration

- Not always solvable
  - P_is cannot lie on the same plane
  - P_is cannot lie on the intersection curve of two quadric surfaces





Quiz

When should we re-calibrate 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.

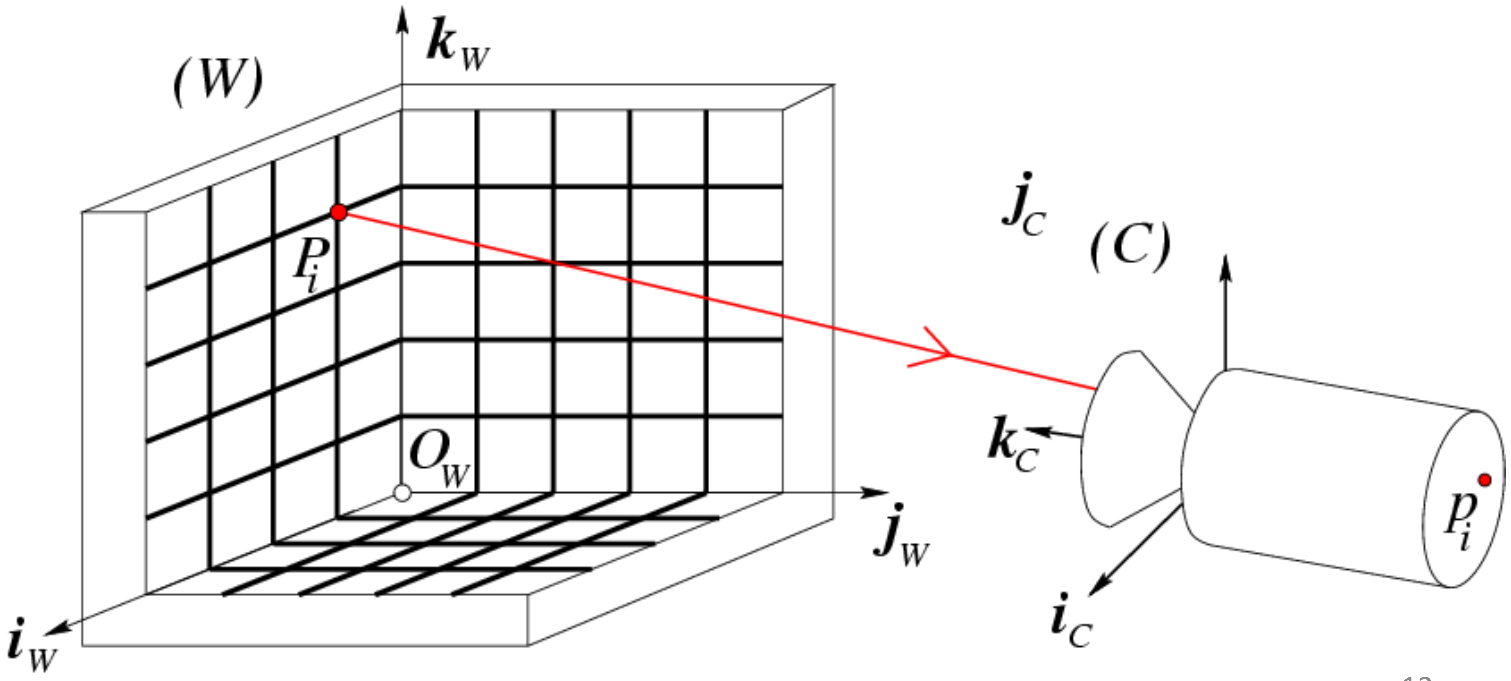
(d) When the camera is moved.

Today’s Agenda

- Review of Previous Lecture
  - Camera calibration
- Epipolar Geometry

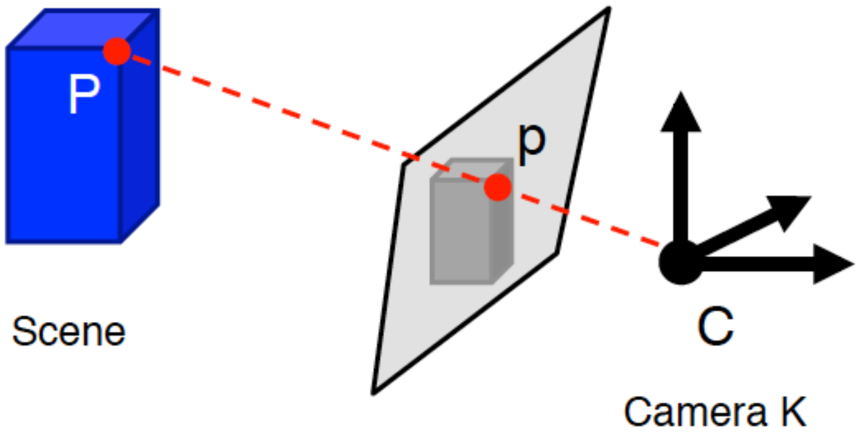
Recovering 3D Geometry

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



Recovering 3D Geometry

- Camera calibration from a single view
  - Camera intrinsic parameters
  - Camera orientation
  - Camera translation
- Can structure/geometry be recovered from a single view?



Recovering 3D Geometry

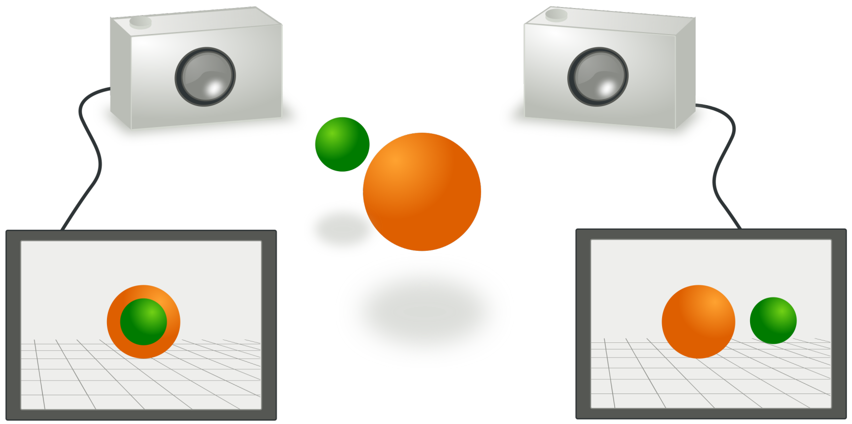
- Camera calibration from a single view
- Recover 3D geometry from a single view?
  - Intrinsic ambiguities of 3D -> 2D
  - Two eyes help

Core Problems in Recovering 3D Geometry

- **Correspondence**: Given a point in one image, how can I find the corresponding point in another one?
- **Camera geometry**: Given corresponding points in two images, establish the relations between multi-views. **Epipolar Geometry**
- **Scene geometry**: Find coordinates of 3D point from its projection into 2 or multiple images.

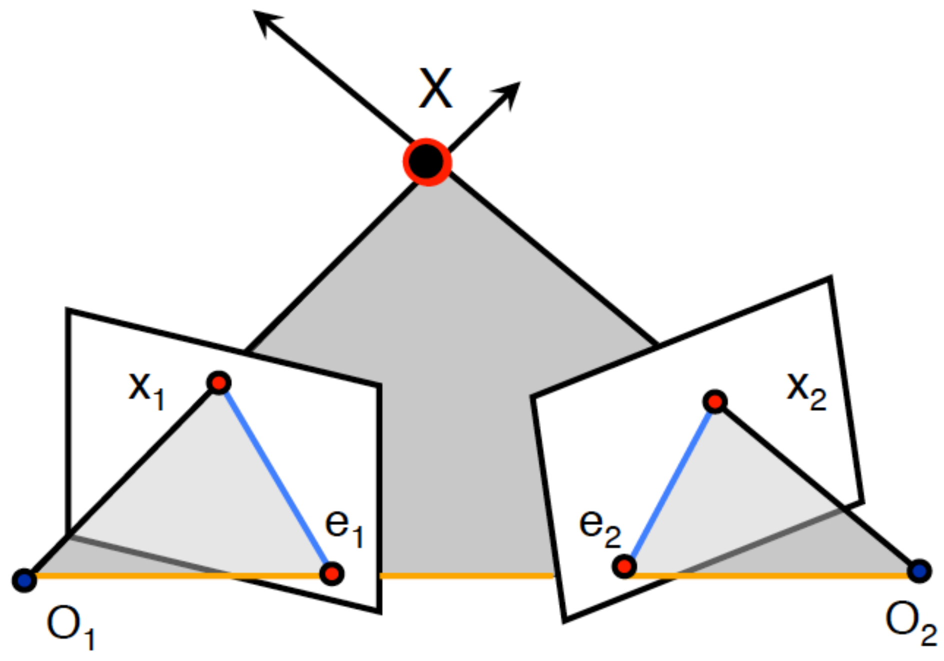
Epipolar Geometry

- The geometry of stereo vision
  - Geometric relations between 3D points and their images points
    - Define constraints between the image points
    - Derived using the pinhole camera model



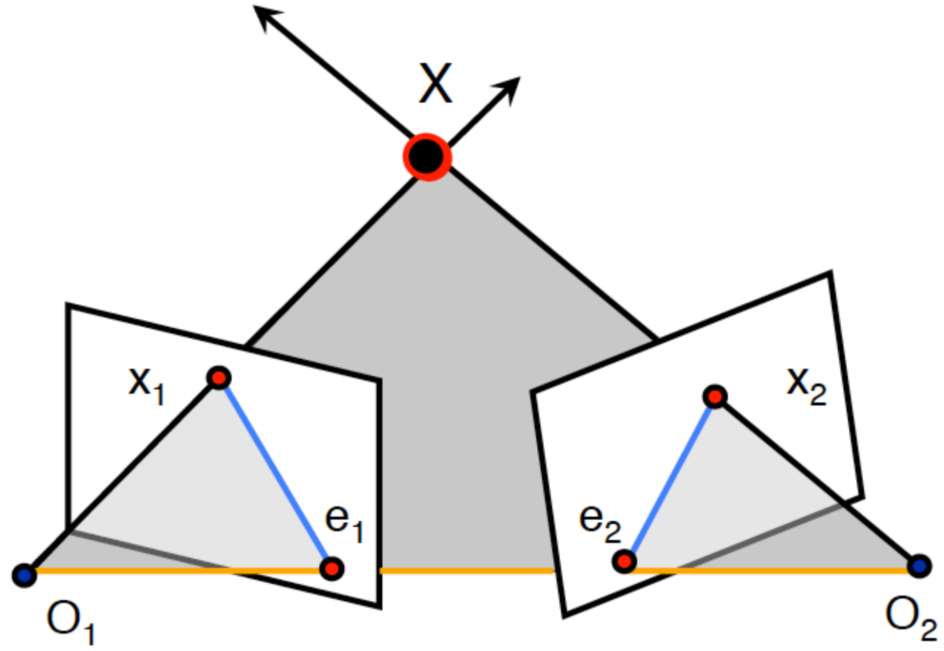
Epipolar Geometry

- Baseline
  - The line between the two camera centers



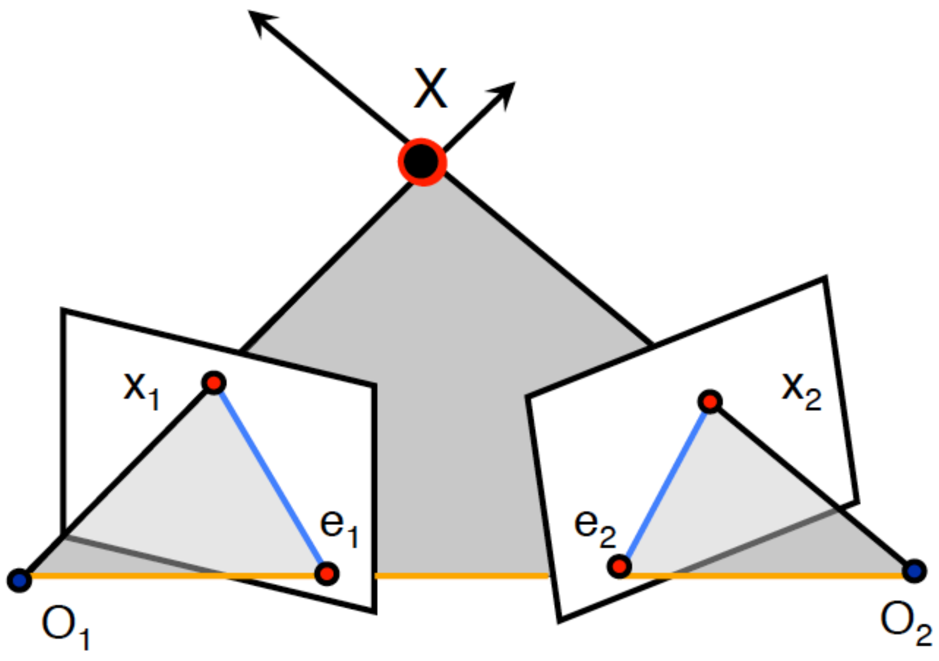
Epipolar Geometry

- Baseline
  - The line between the two camera centers
- Epipolar plane
  - The plane defined by X, O_1, and O_2



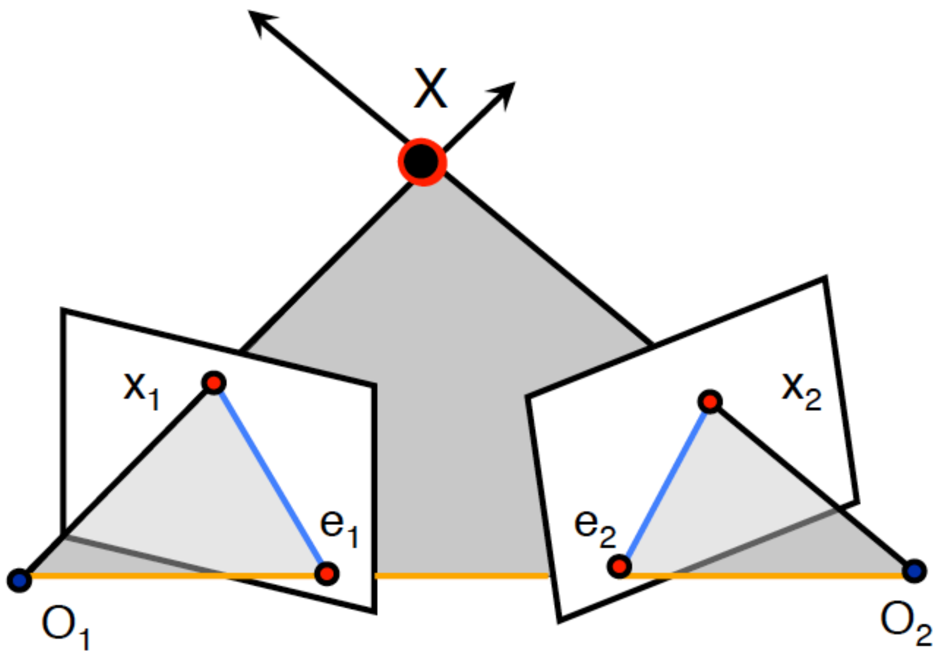
Epipolar Geometry

- Baseline
  – The line between the two camera centers
- Epipolar plane
  – The plane defined by X, O_1, and O_2
- Epipoles
  – ∩ of baseline and image plane
  – Projection of the other camera center



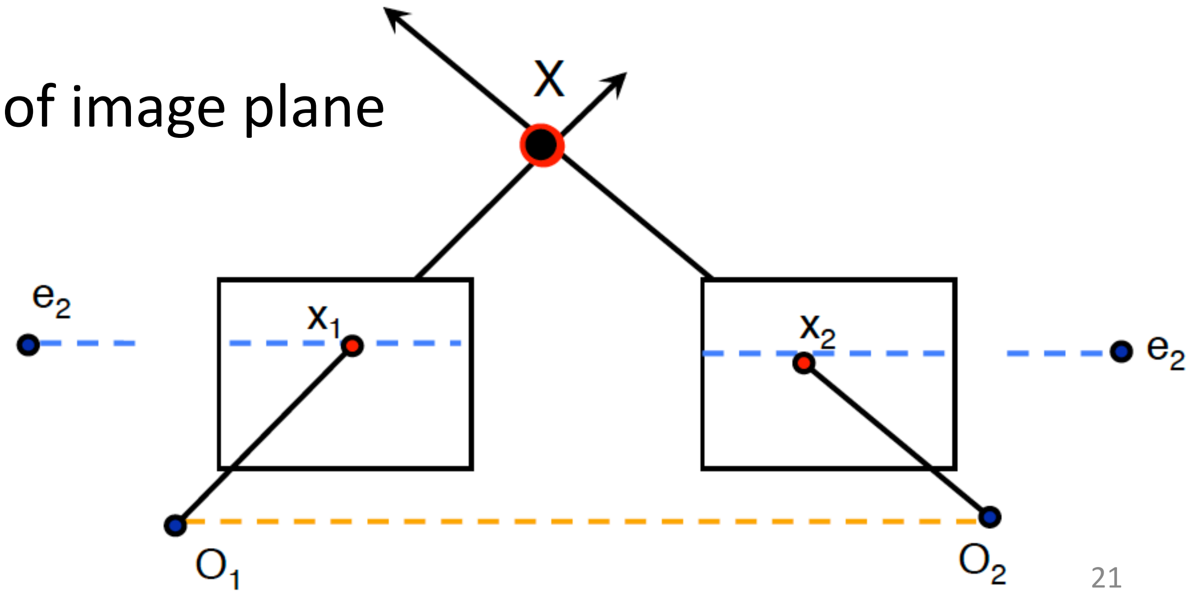
Epipolar Geometry

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



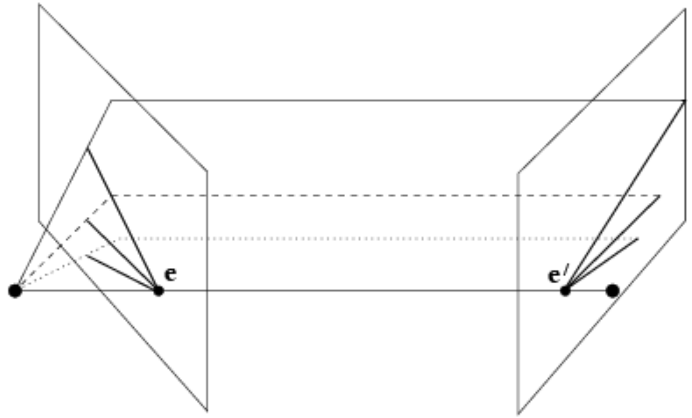
Epipolar Geometry

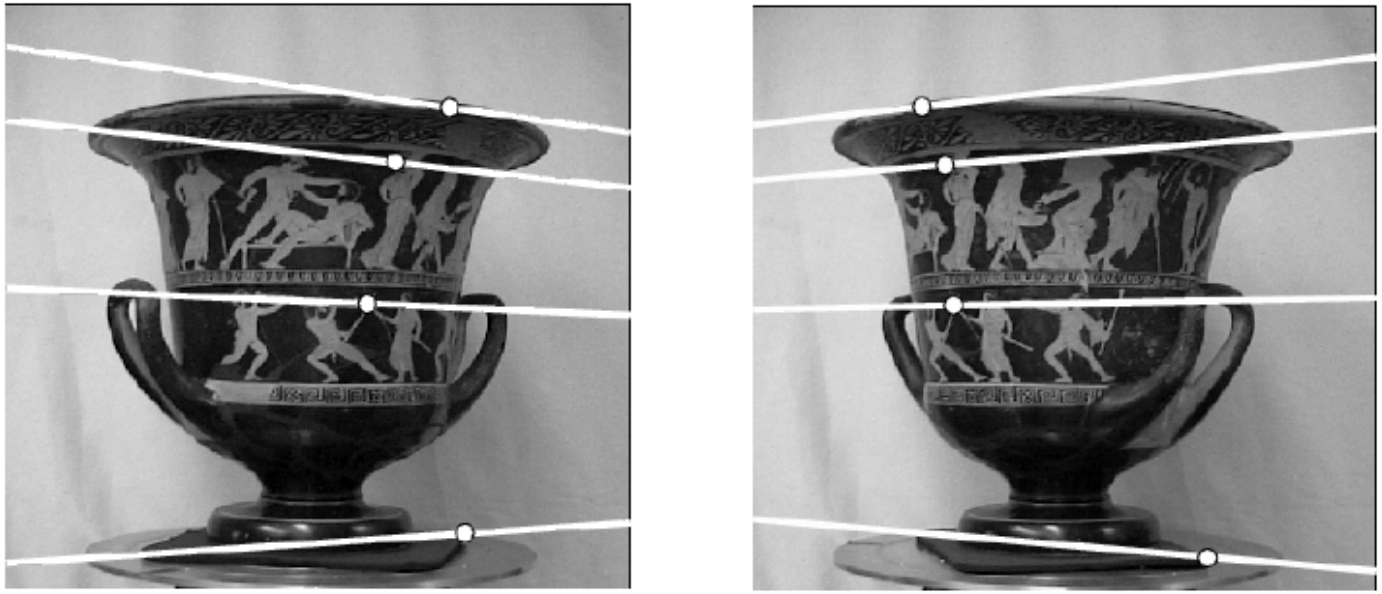
- Example
  - Converging image planes
  - Parallel Image Planes
    - Baseline intersects the image plane at infinity!
    - Epipoles are at infinity!
    - Epipolar lines are parallel to U axis of image plane



Epipolar Geometry

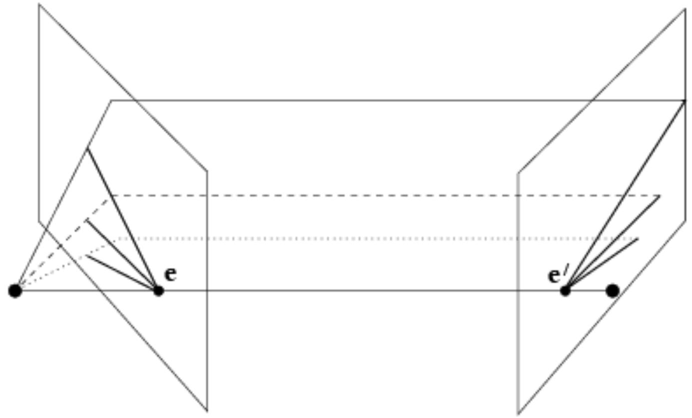
- Example
  - Converging image planes
    - All epipolar lines intersect at the epipole

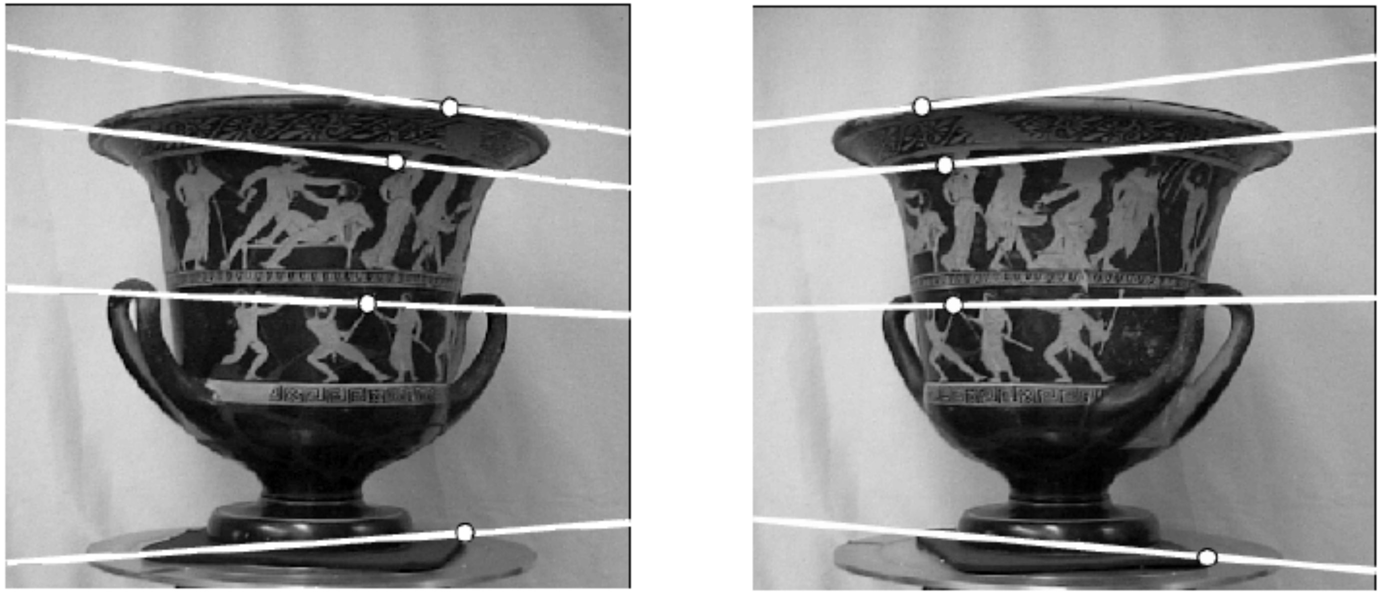




Epipolar Geometry

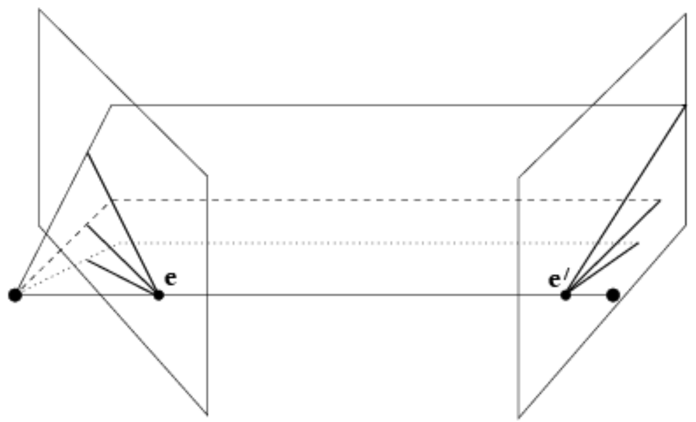
- The relations between different views?
  - How to use for recovering 3D geometry?
    - Unknown: 3D points
    - Known: image points; camera parameters

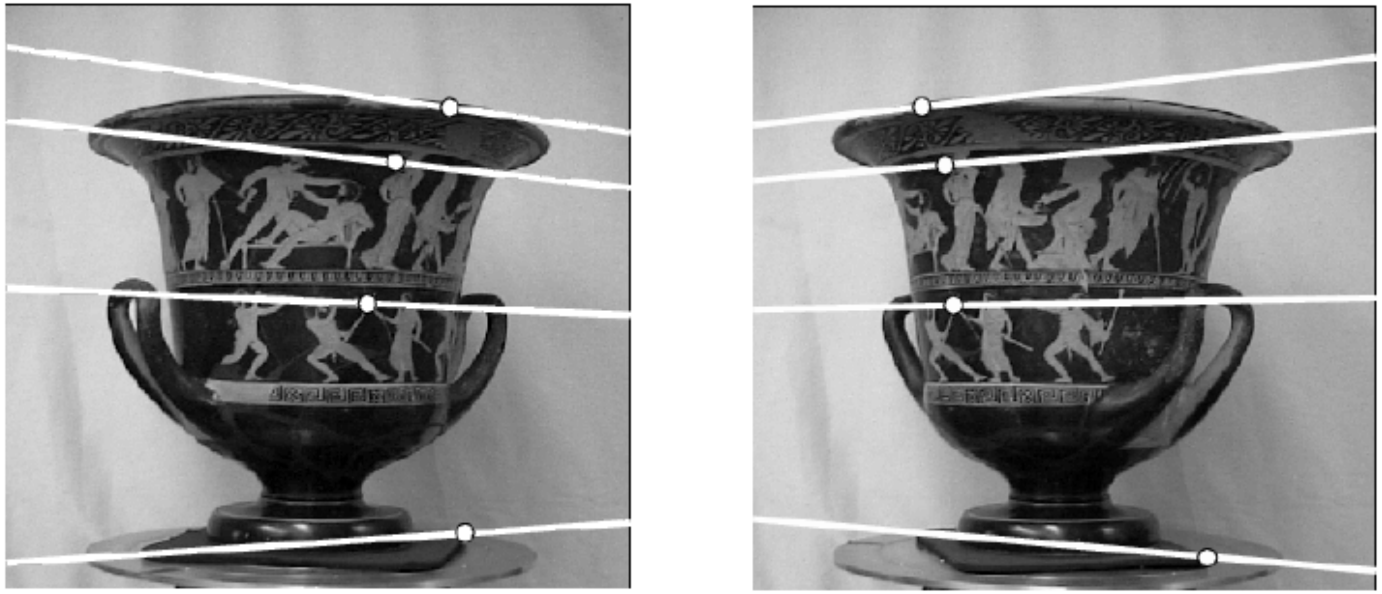




Epipolar Geometry

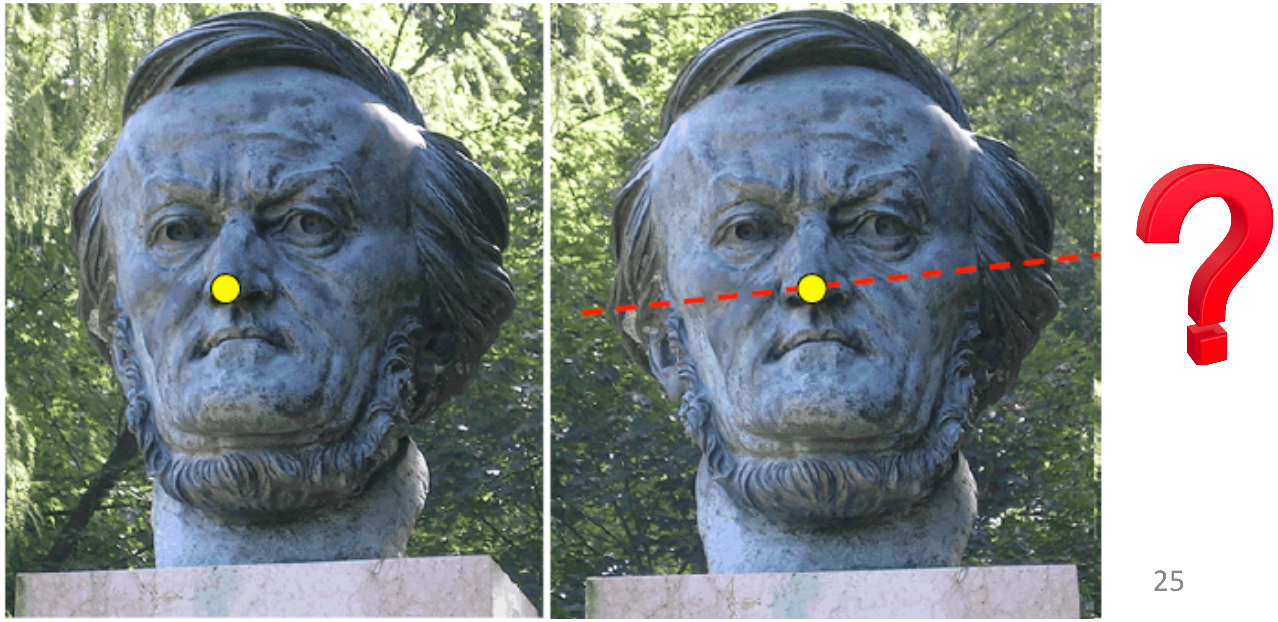
- Constraints between images (without knowing 3D structure)
  - Epipolar lines determined by just camera centers and an image point
  - The image point on the second image must be on its Epipolar line





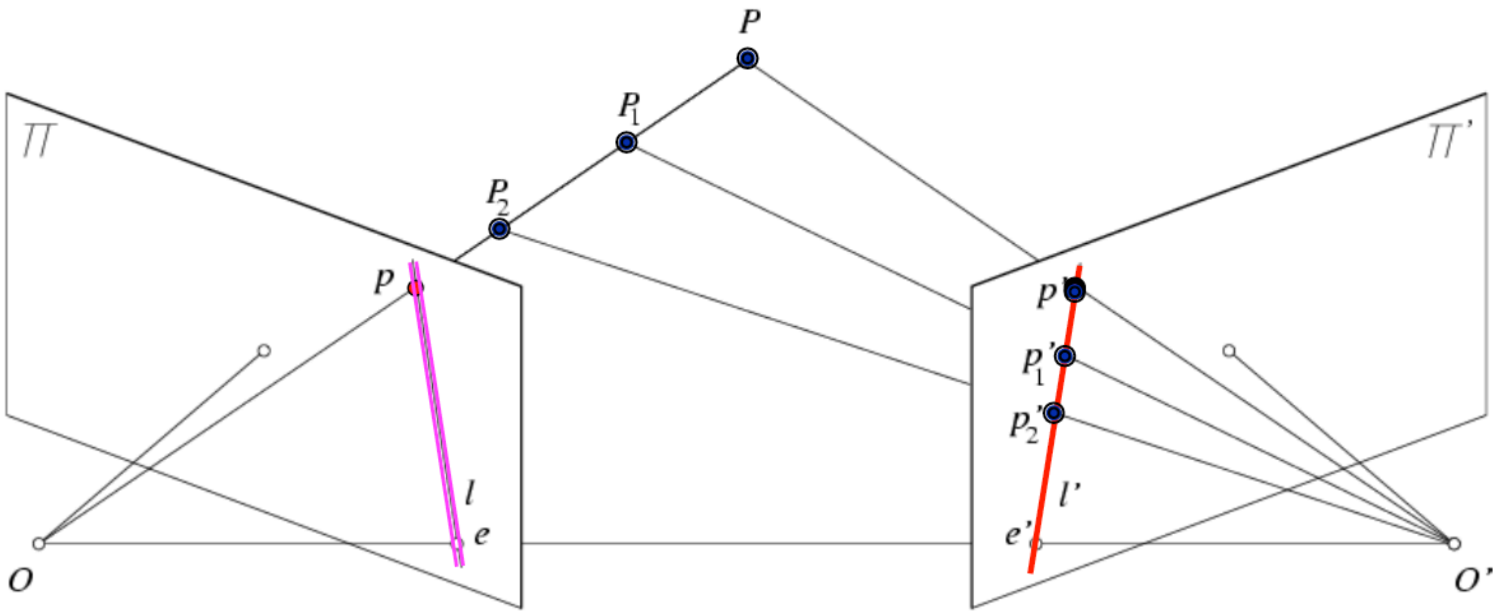
Epipolar Constraint

- Given a point on left image, find the corresponding point on right image?
  - Two views of the same object
  - Known camera positions and camera matrices



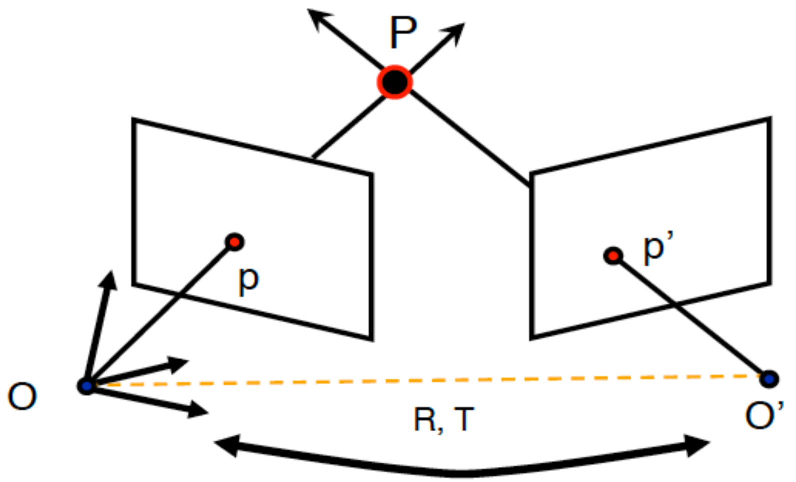
Epipolar Constraint

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



Epipolar Constraint

- The relationship between the two image points
  - Assume the world reference system aligned with the left camera
  - The right camera has offset *T* and orientation *R*



Camera projection matrices

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

$$P \rightarrow M\,P = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

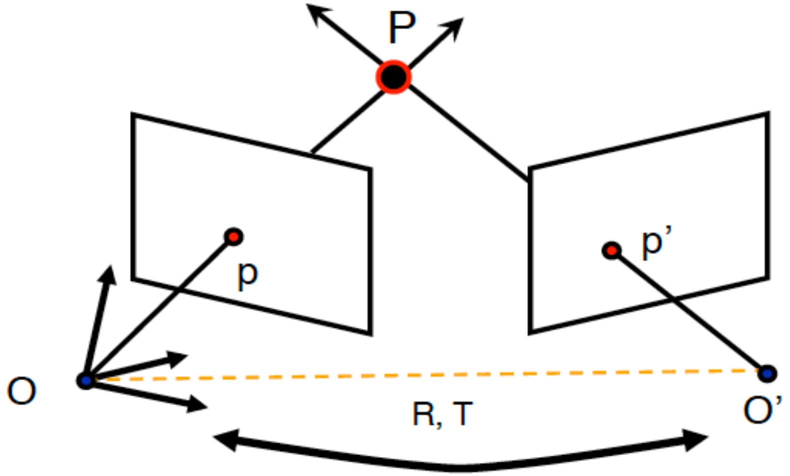
$$M' = K[R \quad T]$$

$$P \rightarrow M'\,P = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$$

Epipolar Constraint

- The relationship between the two image points
  - Canonical cameras
    - *K* is identity

p' in camera 1's coordinate system

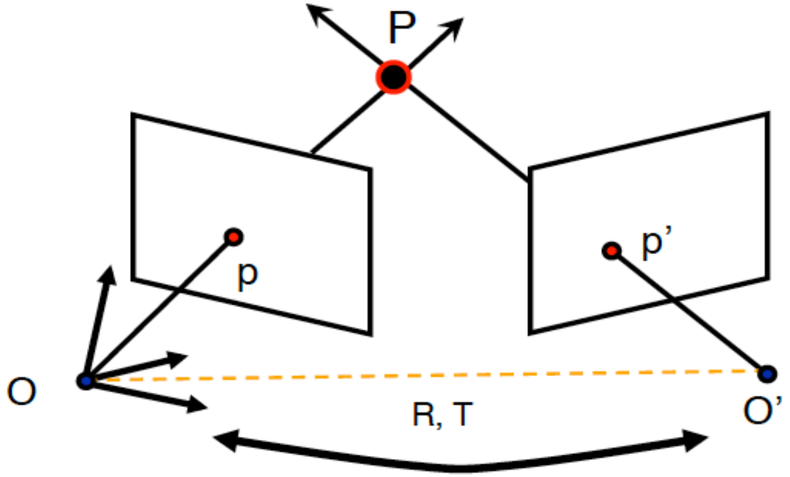


Epipolar Constraint

- The relationship between the two image points
  - Canonical cameras
    - *K* is identity

p' in camera 1's coordinate system $R^T(p' - T)$

O' in camera 1's coordinate system



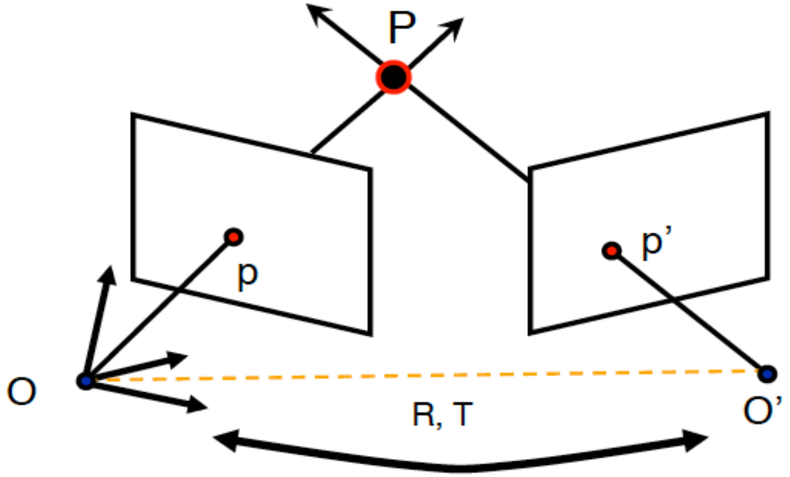
Epipolar Constraint

- The relationship between the two image points
  - Canonical cameras
    - *K* is identity

p' in camera 1's coordinate system $R^T(p' - T)$

O' in camera 1's coordinate system $R^T(O' - T) = -R^T T$

Normal of the Epipolar plane



Epipolar Constraint

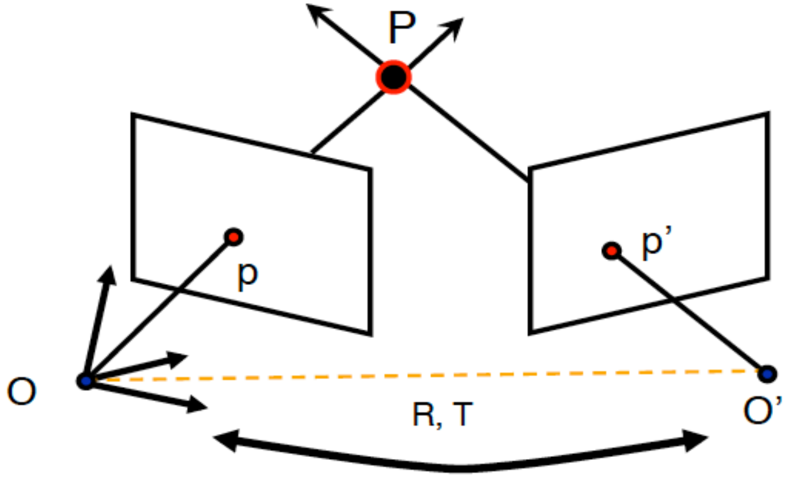
- The relationship between the two image points
  - Canonical cameras
    - *K* is identity

p' in camera 1's coordinate system $R^T(p' - T)$

O' in camera 1's coordinate system $R^T(O' - T) = -R^T T$

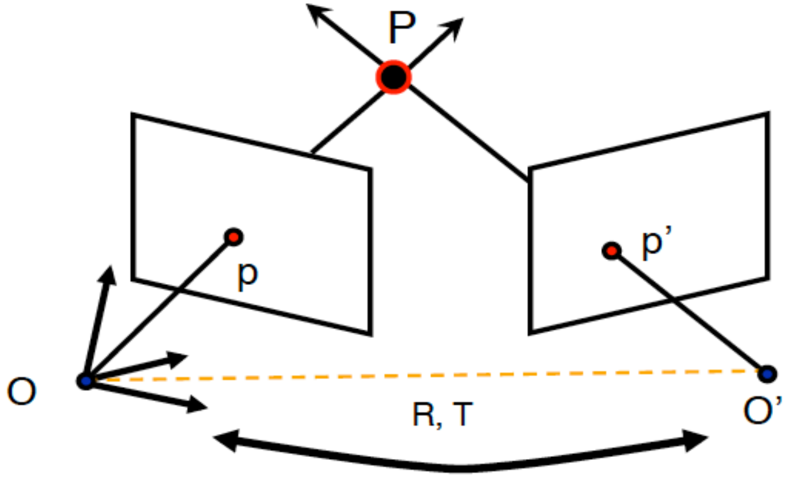
Normal of the Epipolar plane $R^T T \times [R^T(p' - T)] = R^T(T \times p')$

Op lies in the Epipolar plane



Epipolar Constraint

- The relationship between the two image points
  - Canonical cameras
    - *K* is identity



p' in camera 1's coordinate system $R^T(p' - T)$

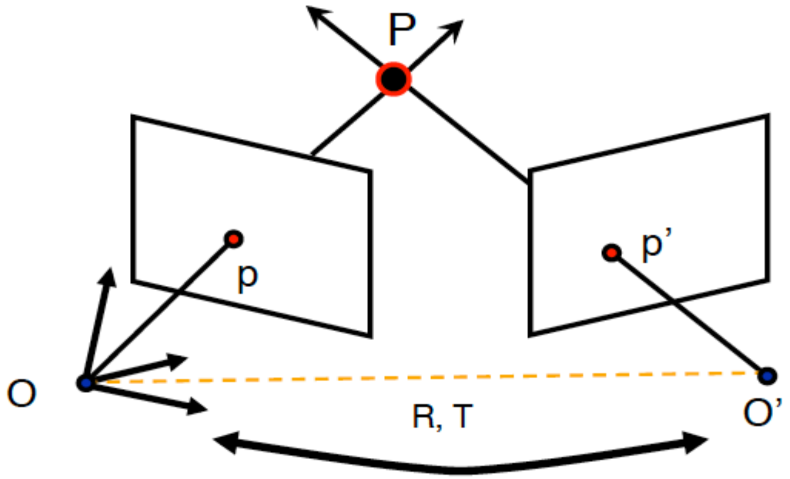
O' in camera 1's coordinate system $R^T(O' - T) = -R^T T$

Normal of the Epipolar plane $R^T T \times [R^T(p' - T)] = R^T(T \times p')$

Op lies in the Epipolar plane $[R^T(T \times p')]^T p = 0$

Epipolar Constraint

- The relationship between the two image points
  - Canonical cameras
    - *K* is identity



p' in camera 1's coordinate system $R^T(p' - T)$

O' in camera 1's coordinate system $R^T(O' - T) = -R^T T$

Normal of the Epipolar plane $R^T T \times [R^T(p' - T)] = R^T(T \times p')$

Op lies in the Epipolar plane $[R^T(T \times p')]^T p = 0$ ⇒ $(T \times p')^T R p = 0$

Epipolar Constraint

- The relationship between the two image points
  - Canonical cameras
    - *K* is identity

Cross product as matrix-vector multiplication

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

p′ in camera 1's coordinate system $R^T(p' - T)$

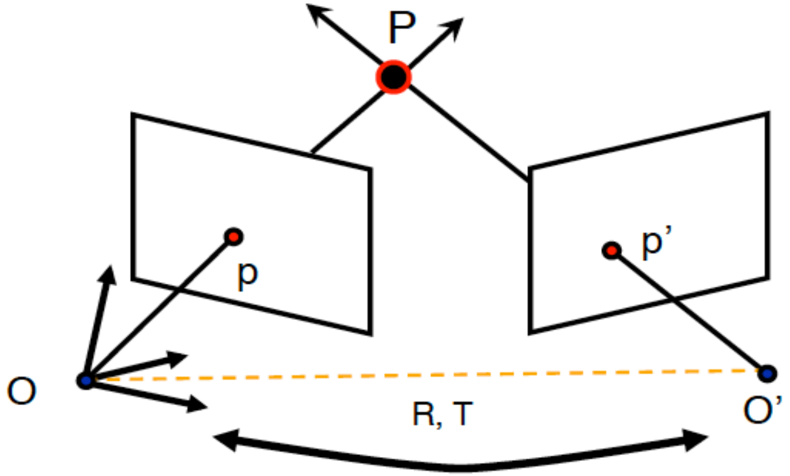
O′ in camera 1's coordinate system $R^T(O' - T) = -R^T T$

Normal of the Epipolar plane $R^T T \times [R^T(p' - T)] = R^T(T \times p')$

Op lies in the Epipolar plane $[R^T(T \times p')]^T p = 0 \Rightarrow (T \times p')^T R p = 0$

Epipolar Constraint

- The relationship between the two image points
  - Canonical cameras
    - *K* is identity



p' in camera 1's coordinate system $R^T(p' - T)$

O' in camera 1's coordinate system $R^T(O' - T) = -R^T T$

Normal of the Epipolar plane $R^T T \times [R^T(p' - T)] = R^T(T \times p')$

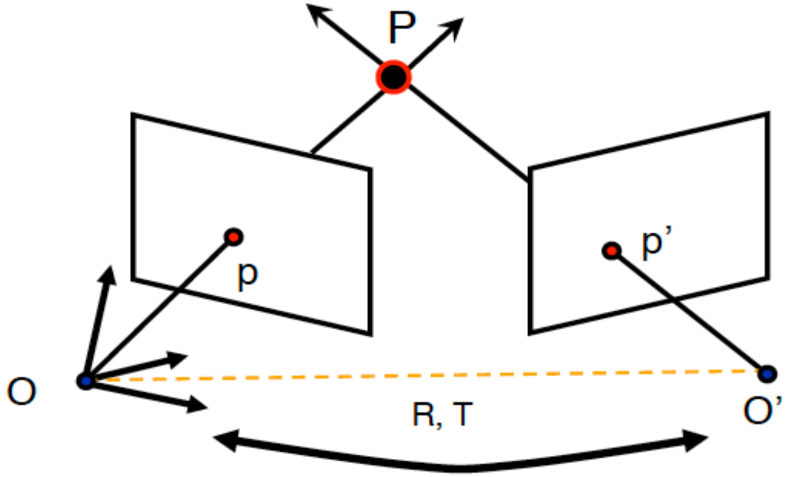
Op lies in the Epipolar plane $[R^T(T \times p')]^T p = 0 \Rightarrow (T \times p')^T R p = 0$

$T \times p' = [T_\times] p'$

$\Rightarrow ([T_\times] p')^T R p = 0$

Epipolar Constraint

- The relationship between the two image points
  - Canonical cameras
    - *K* is identity



p' in camera 1's coordinate system	$R^T(p' - T)$
O' in camera 1's coordinate system	$R^T(O' - T) = -R^T T$
Normal of the Epipolar plane	$R^T T \times [R^T(p' - T)] = R^T(T \times p')$
Op lies in the Epipolar plane	$[R^T(T \times p')]^T p = 0 \Rightarrow (T \times p')^T R p = 0$

$T \times p' = [T_\times] p'$ $\Rightarrow$ $([T_\times] p')^T R p = 0 \Rightarrow p'^T [T_\times] R\, p = 0$

Epipolar Constraint

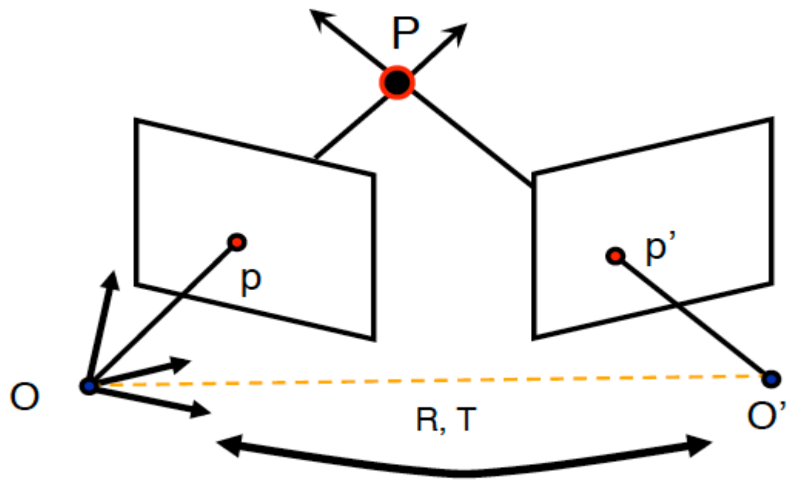
- Essential matrix
  - Establish constraints between matching image points
  - Determine relative position and orientation of two cameras

$$p'^T [T_\times] R \, p = 0$$

$$E = [T_\times] R$$

$$p'^T E \, p = 0$$

Essential matrix



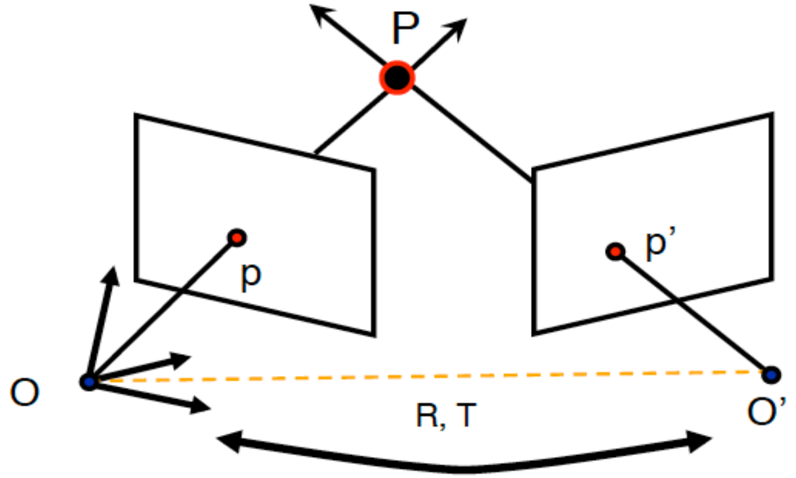
Epipolar Constraint

- How to generalize Essential matrix?
  - Canonical cameras
    - *K* is identity

$$E = [T_X]R \qquad p'^T E\, p = 0$$

$$M = [I \quad 0] \qquad K = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad M' = [R \quad T]$$

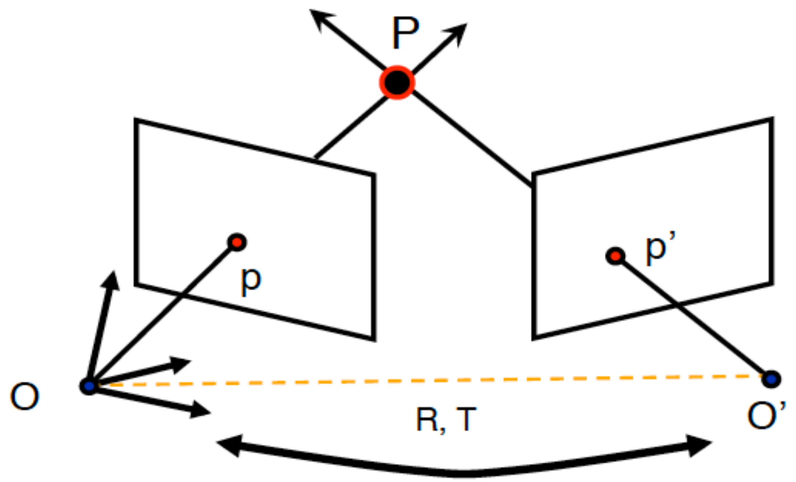
$$M = K[I \quad 0]$$ **K is unknown** $$M' = K[R \quad T]$$



Epipolar Constraint

$M = [I \quad 0]$ $\quad K = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $\quad M' = [R \quad T]$ $\Rightarrow$ $p'^T E\, p = 0, \quad E = [T_X]R$

$M = K[I \quad 0]$ **K is unknown** $M' = K[R \quad T]$?



Epipolar Constraint

- Fundamental matrix

$M = [I \quad 0]$ $\quad K = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $\quad M' = [R \quad T]$ ⇨ $p'^T E\, p = 0, \quad E = [T_\times]R$

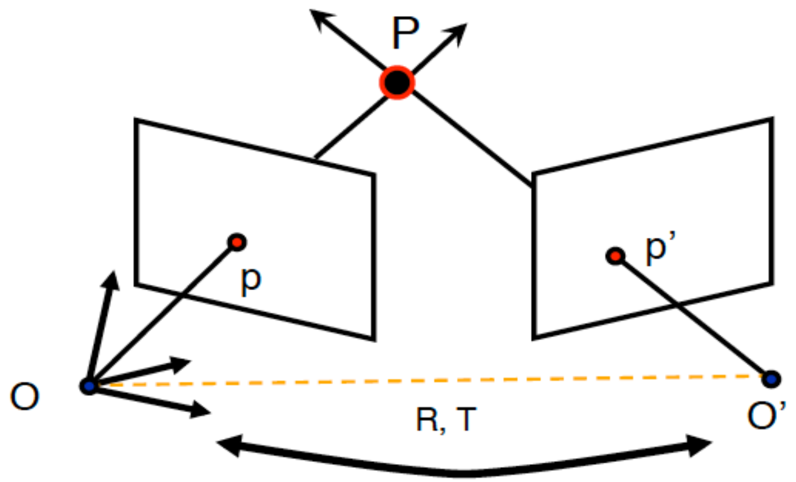
$\mathrm{M} = \mathrm{K}[\mathrm{I} \quad 0]$ **K is unknown** $\mathrm{M}' = \mathrm{K}[\mathrm{R} \quad \mathrm{T}]$ ⇨ $p'^T F\, p = 0, \quad F = K'^{-T}[T_\times]RK^{-1}$

Hint for derivation

$$\begin{cases} p \rightarrow K^{-1}\, p \\ p' \rightarrow K'^{-1}\, p' \end{cases}$$

$$p'^T K'^{-T}[T_\times]RK^{-1} p = 0$$

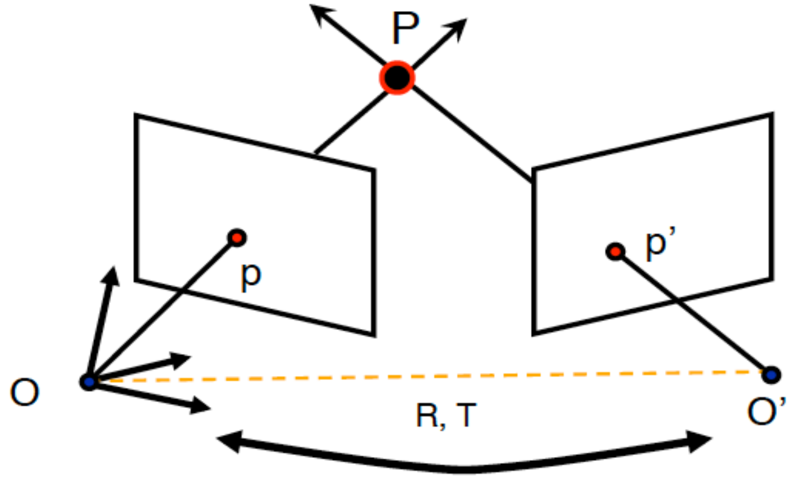
F



Epipolar Constraint

- Fundamental matrix *F* is a matrix product of camera parameters
- Encode Epipolar geometry of 2 views & camera parameters

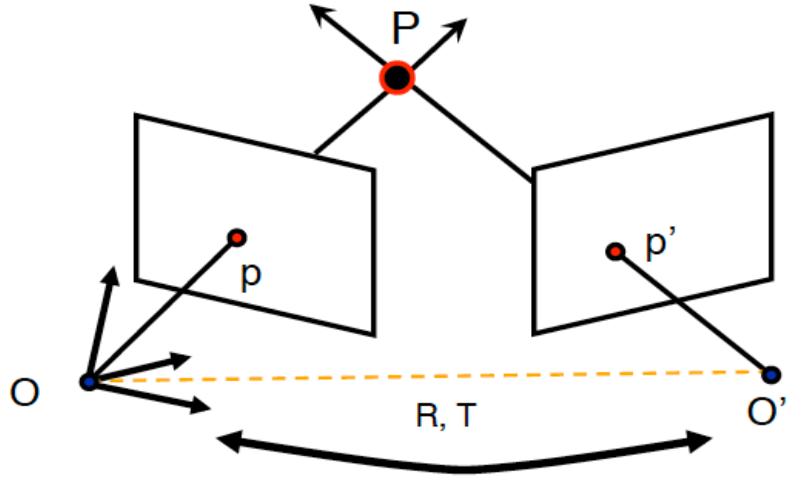
$$p'^T F\, p = 0, \quad F = K'^{-T}[T_\times]RK^{-1}$$



Epipolar Constraint

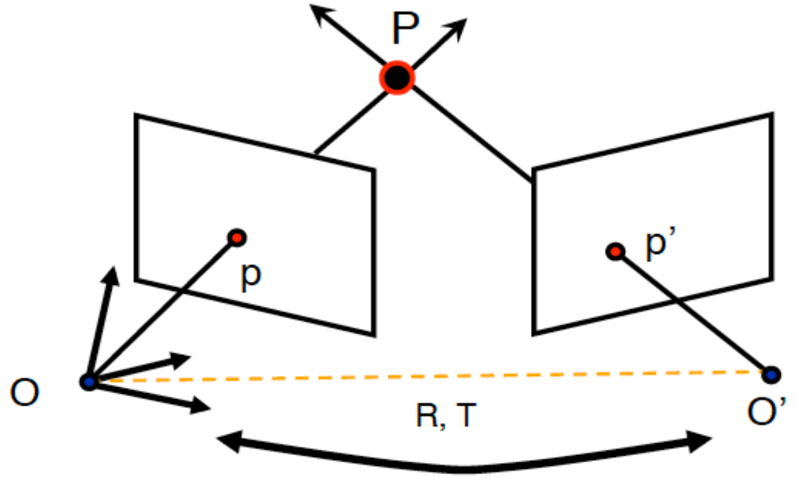
- Fundamental matrix *F* is a matrix product of camera parameters
- Encode Epipolar geometry of 2 views & camera parameters
- How is the fundamental matrix useful?
  - – A 3D point's image in one image -> the Epipolar line in the other image
    - No need 3D point location
    - No need camera extrinsic parameters
    - No need camera intrinsic parameters

$$p'^T F\, p = 0, \quad F = K'^{-T}[T_\times]RK^{-1}$$



Epipolar Constraint

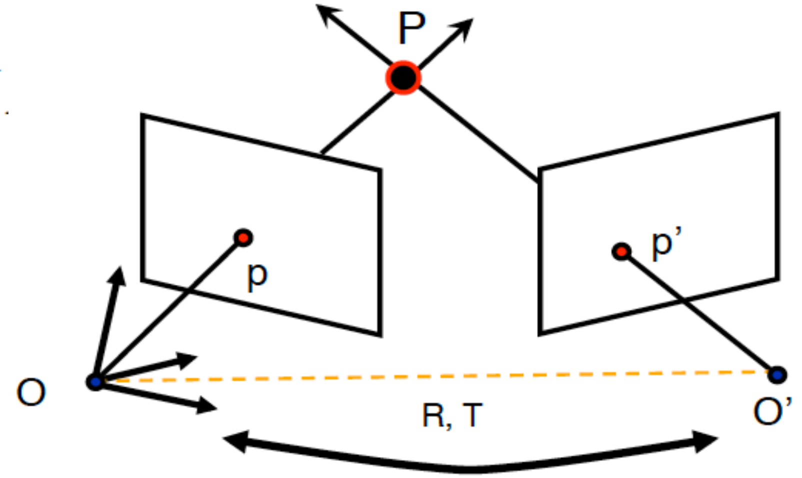
- Fundamental matrix *F* is a matrix product of camera parameters
- Encode Epipolar geometry of 2 views & camera parameters
- How is the fundamental matrix useful?
  - A 3D point's image in one image -> the image point in the other image
    - No need 3D point location
    - No need camera extrinsic parameters
    - No need camera intrinsic parameters
  - Powerful tool in:
    - 3D reconstruction
    - Multi-view object/scene matching



Epipolar Constraint

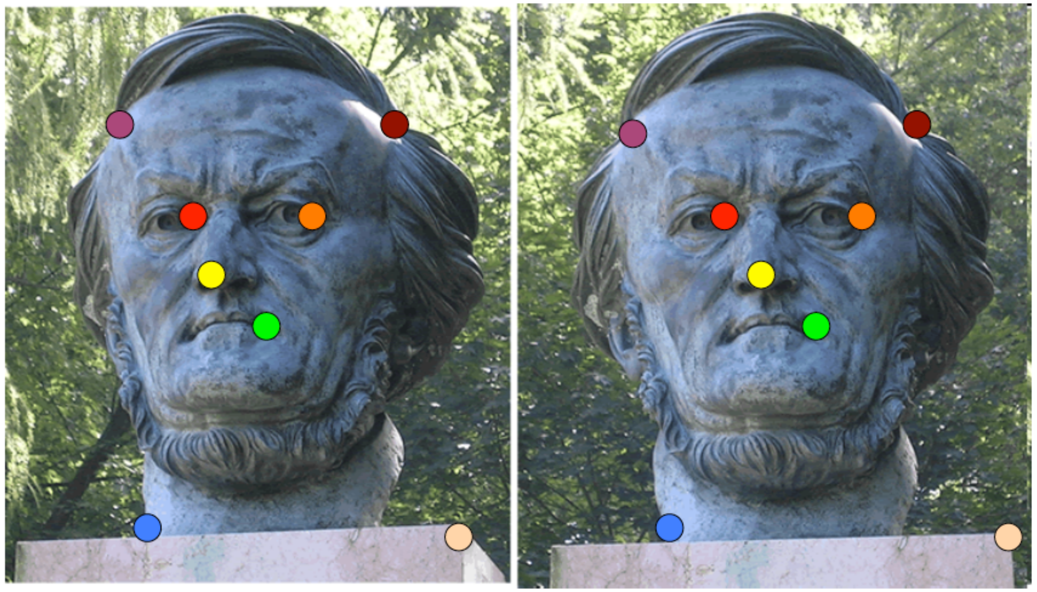
- Fundamental matrix *F* is a matrix product of camera parameters
- Encode Epipolar geometry of 2 views & camera parameters
- How is the fundamental matrix useful?
- How to recover *F*?
  - From image correspondences

$$p'^T F\, p = 0, \quad F = K'^{-T}[T_\times]RK^{-1}$$



Epipolar Constraint

- Fundamental matrix *F* is a matrix product of camera parameters
- Encode Epipolar geometry of 2 views & camera parameters
- How is the fundamental matrix useful?
- How to recover *F*?
  – From image correspondences
  – *F* has 7 degrees of freedom
    - How many point pairs needed?



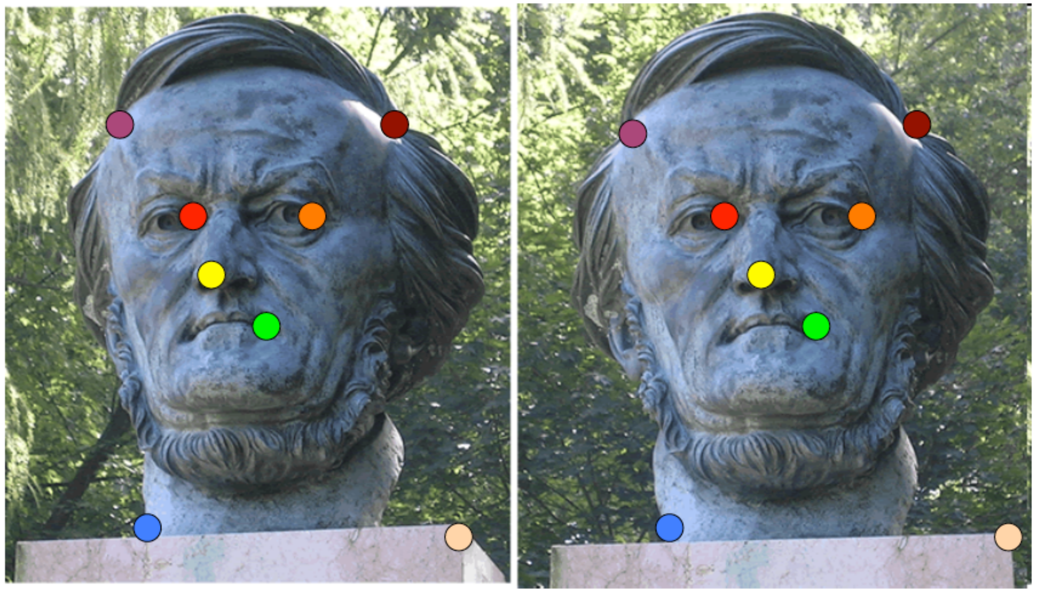
Recovering Fundamental Matrix

Assume a correspondence

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

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

8+ point pairs



Recovering Fundamental Matrix

- 8-point algorithm

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

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

Recovering Fundamental Matrix

- 8-point algorithm
- Solved using SVD

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

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

Recovering Fundamental Matrix

- 8-point algorithm
- Solved using SVD

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

Just the idea on how to recover F.
Advanced techniques exist to improve robustness.
Details in the lecture note 3.

Normalized 8-point algorithm

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

Next Lecture

- Image Matching
  - Find corresponding image points
- Triangulation
- Structure from Motion
  - Go beyond two views
  - Simultaneously determine 3D structure & camera parameters