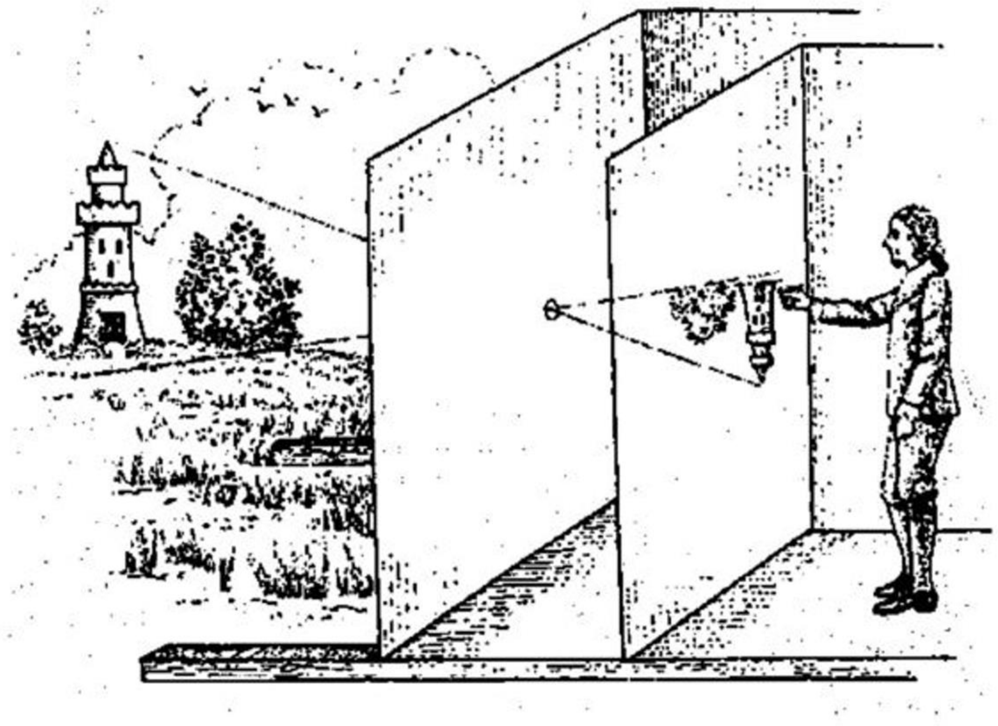
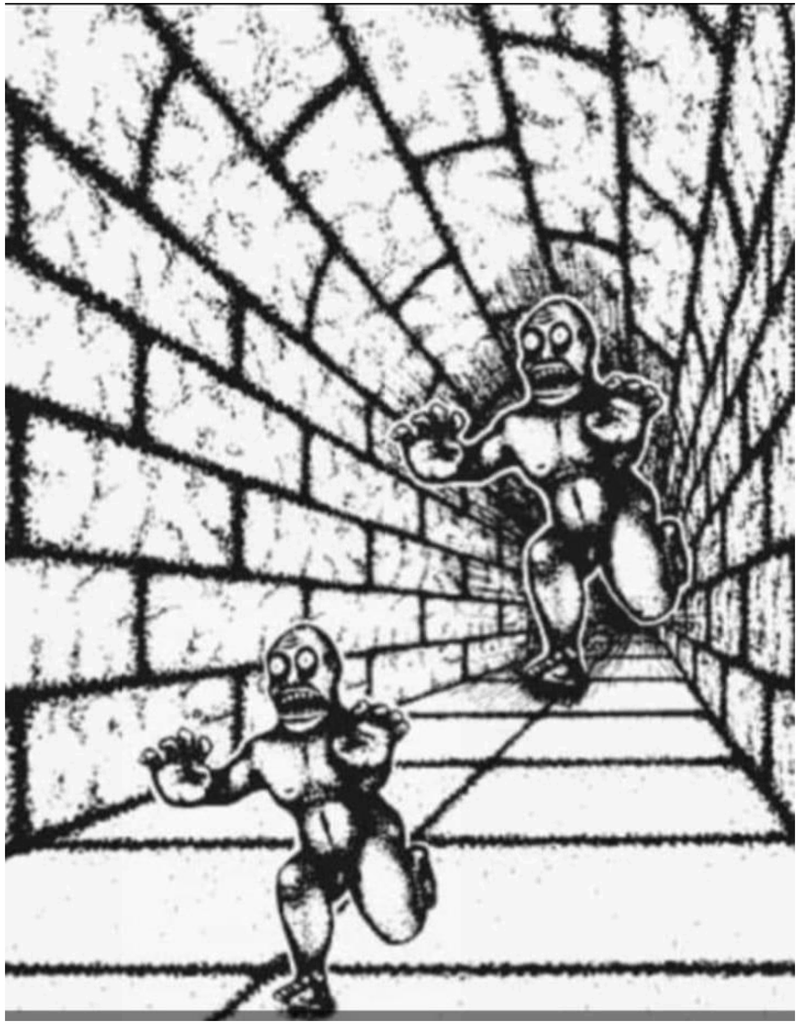
It’s a 3D World, After All



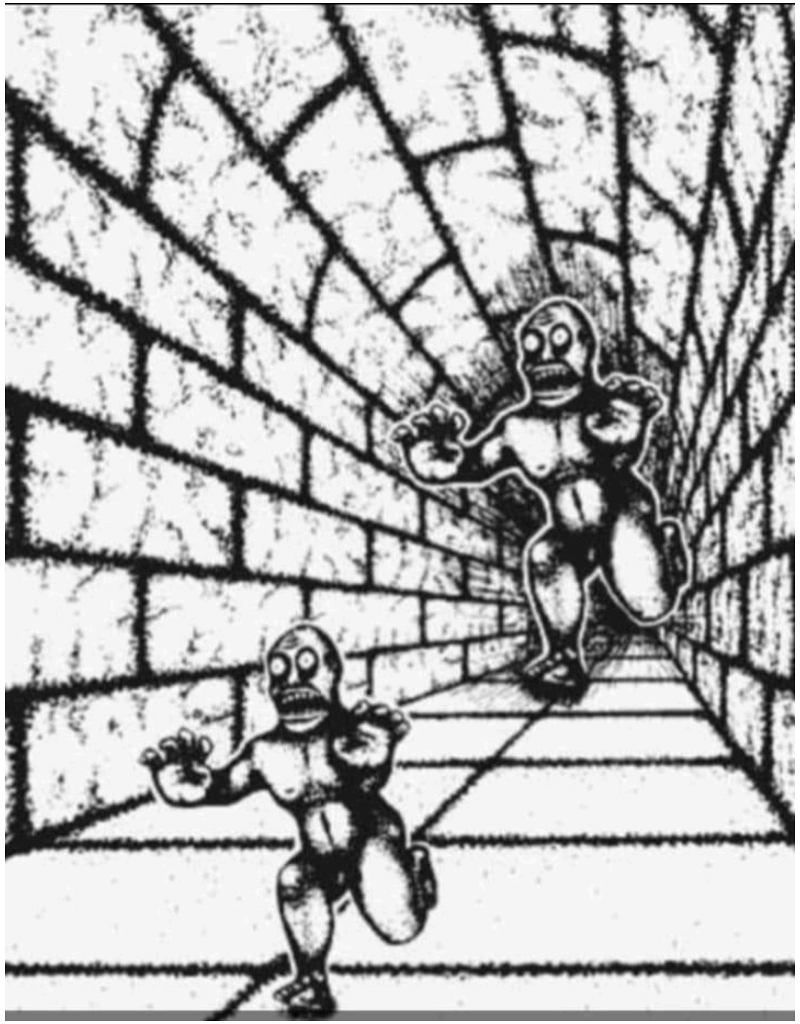
3D Computer Vision

Elliott Wu

We’re good at seeing 3D!

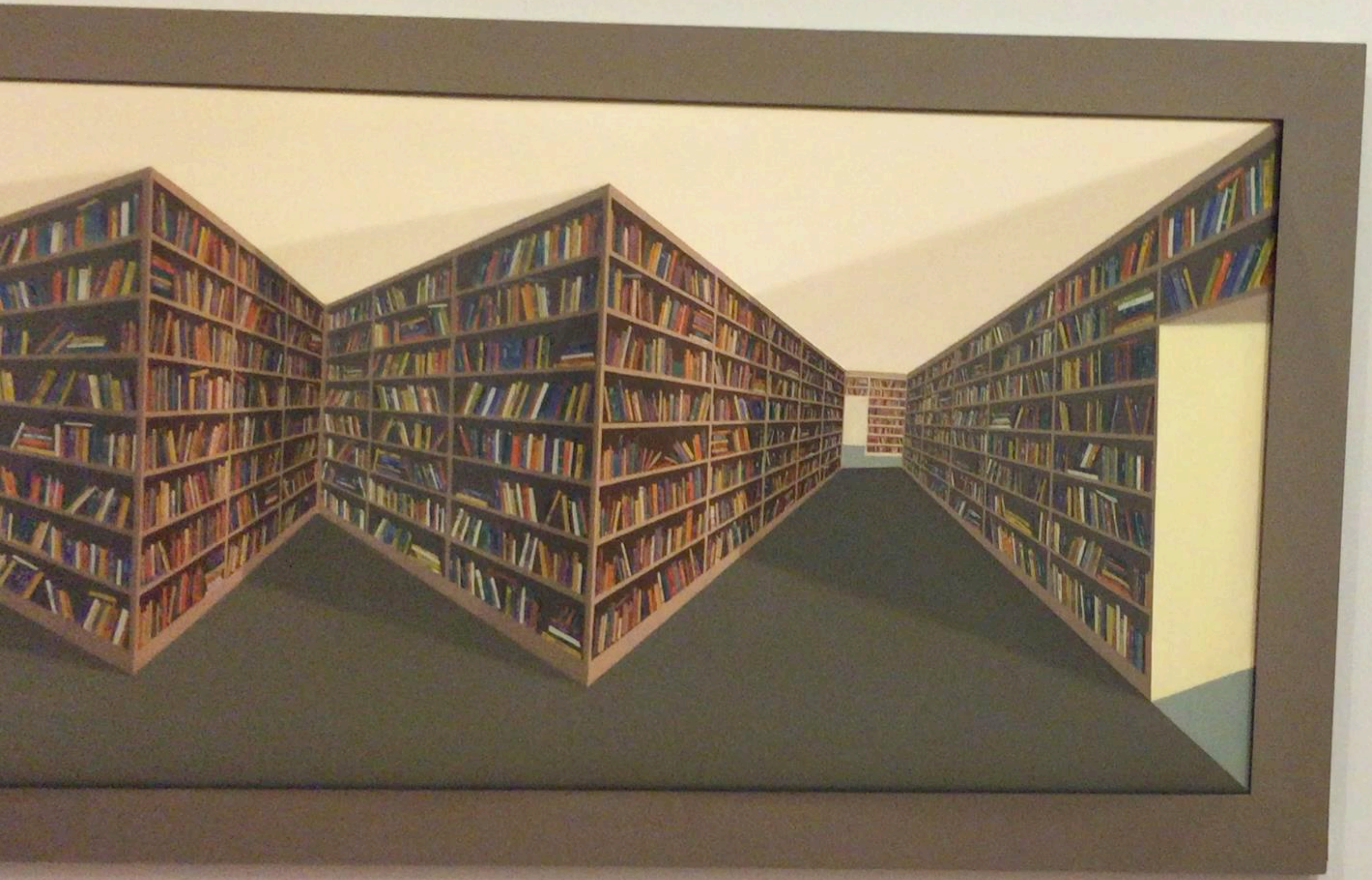


We’re good at seeing 3D!



And sometimes too good..

And sometimes too good..

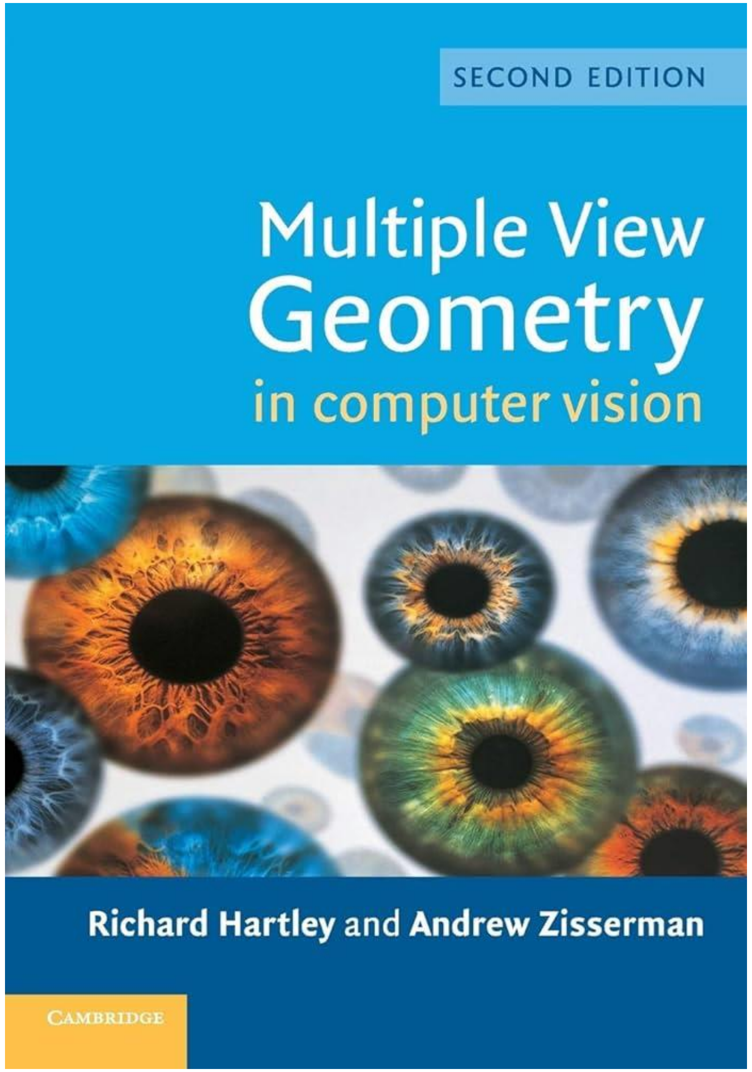



BI
UK
REVERSPECTIVE

3D Computer Vision – Outline

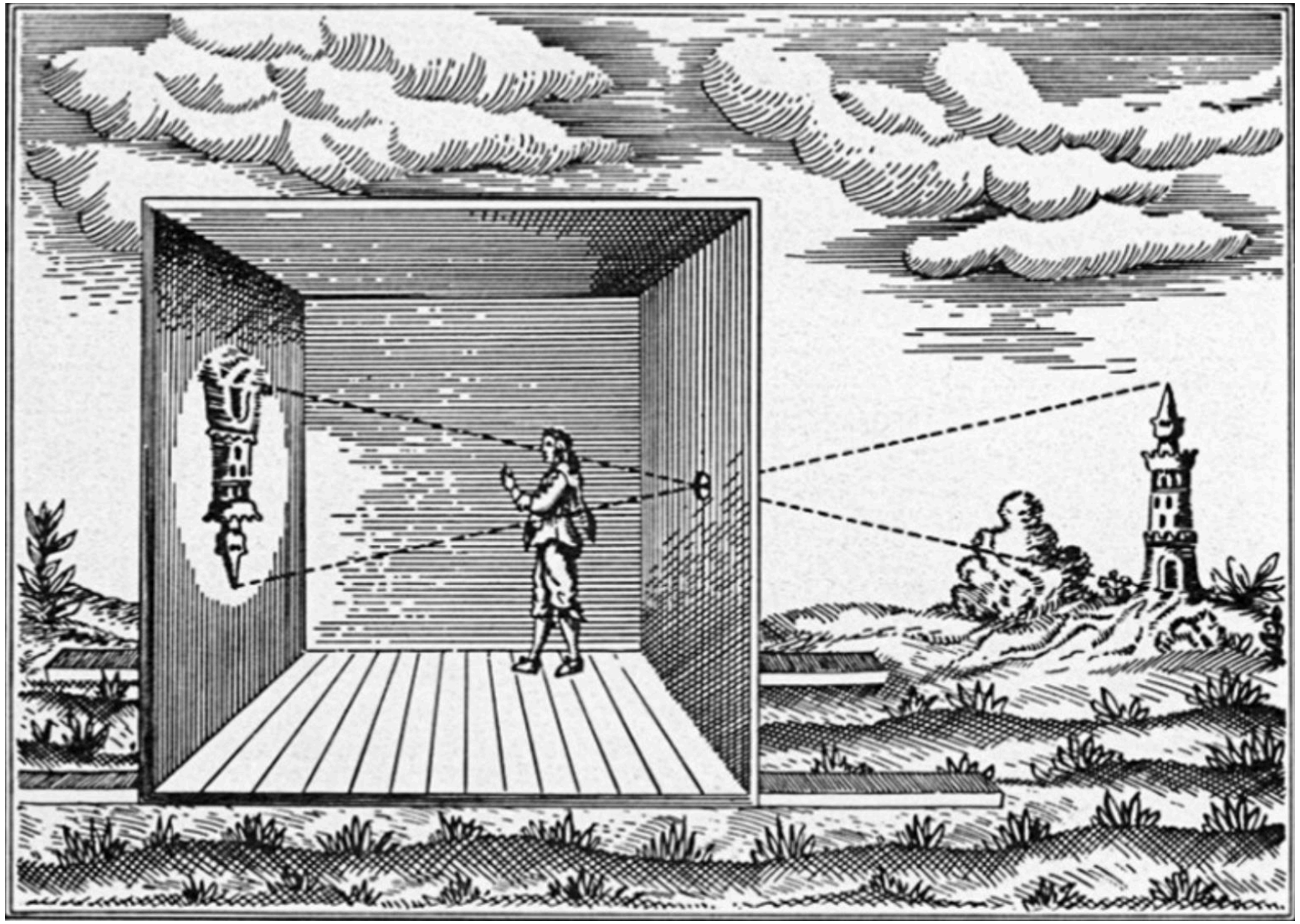
- Part 1 – Camera Model & Projection
- Part 2 – Multi-view Geometry
- Part 3 – 3D Representations & Rendering
- Part 4 – Learning-based 3D Modeling

Multi-view Geometry “Bible”



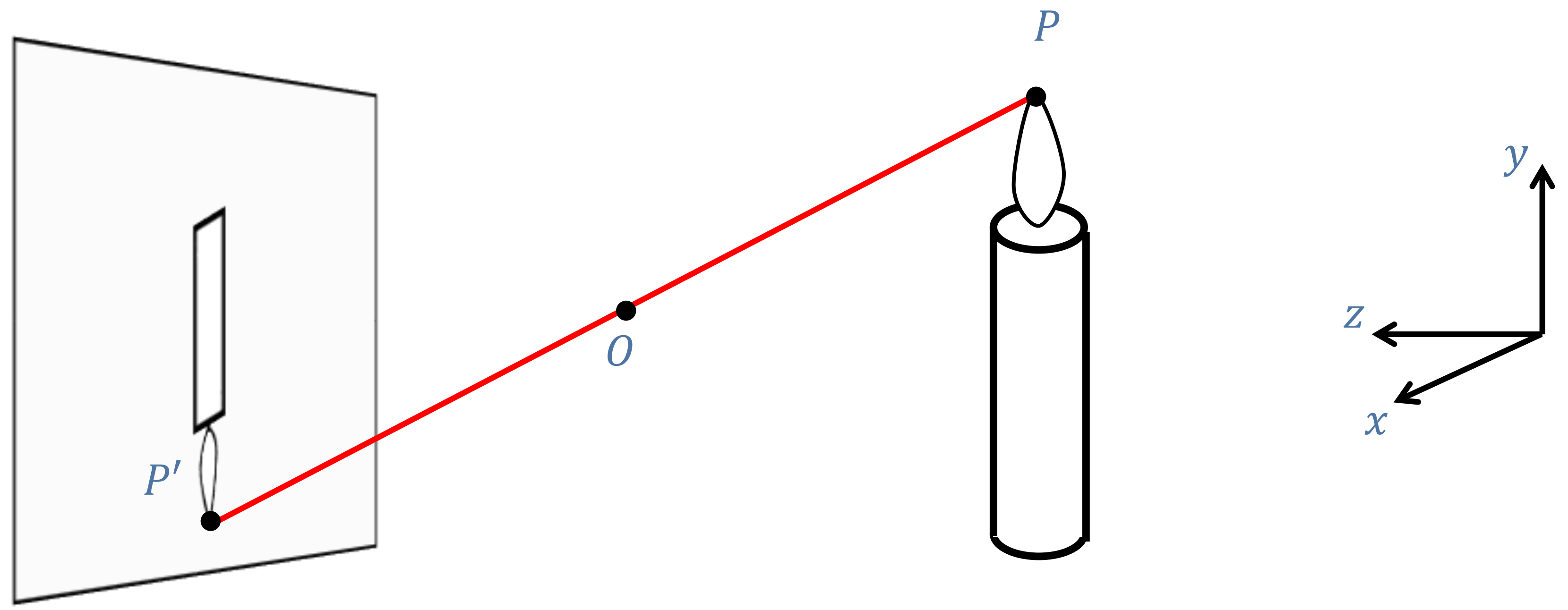
Part 1 – Camera Model & Projection

Image Formation – Camera Obscura



- Basic principle known to Mozi (470-390 BCE), Aristotle (384-322 BCE)
- Drawing aid for artists: described by Leonardo da Vinci (1452-1519)

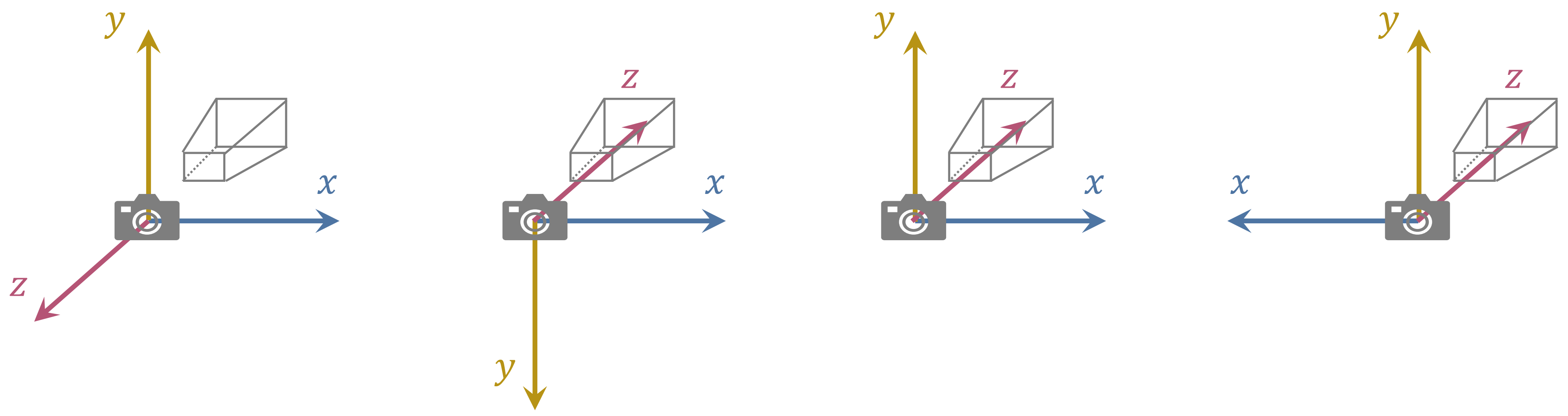
Pinhole Camera



Canonical coordinate system

- The optical center (O) is at the origin
- The z axis is the optical axis perpendicular to the image plane
- The xy plane is parallel to the image plane, x and y axes are horizontal and vertical directions of the image plane

Everybody Agrees… Except on the Axes



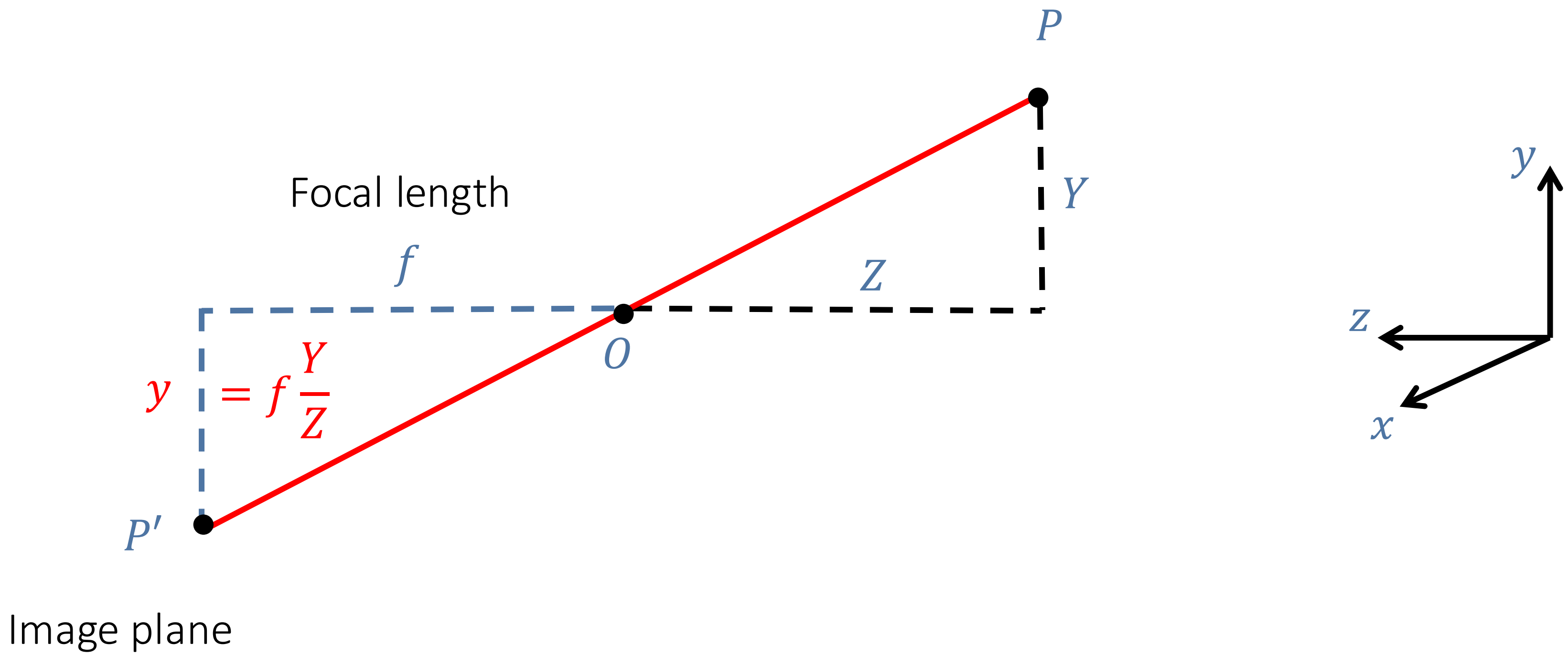
- OpenGL
- Blender
- ARKit
- Three.js
- Nerfstudio
- …

- OpenCV
- Open3D
- COLMAP
- gsplat
- …

- Unity
- …

- PyTorch3D
- ?

Perspective Projection

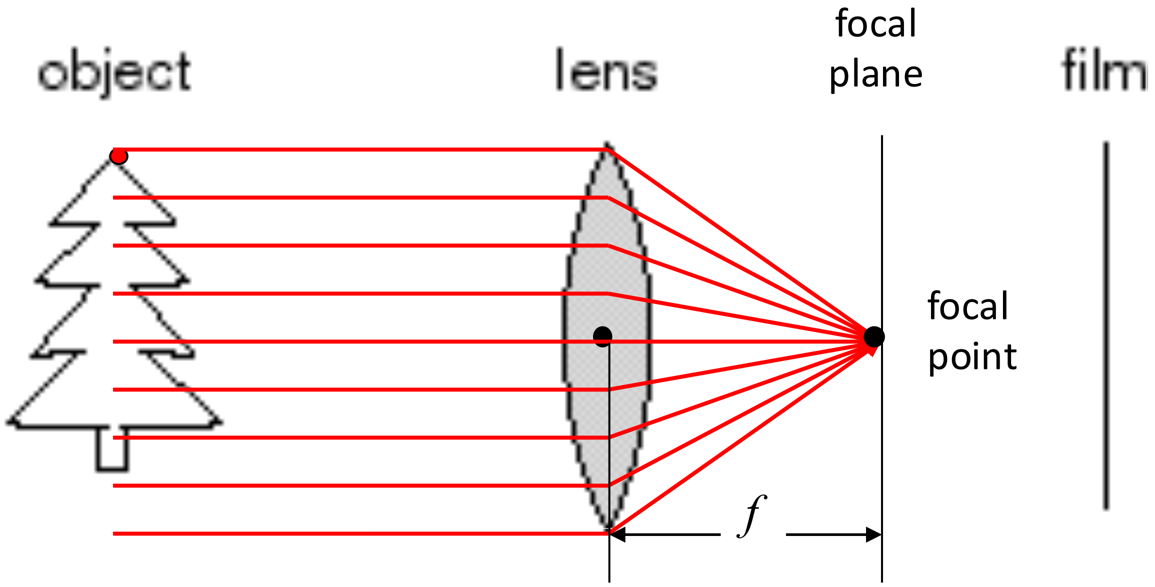


$$(X, Y, Z) \to \left(f\frac{X}{Z}, f\frac{Y}{Z}\right)$$

3D point 2D point

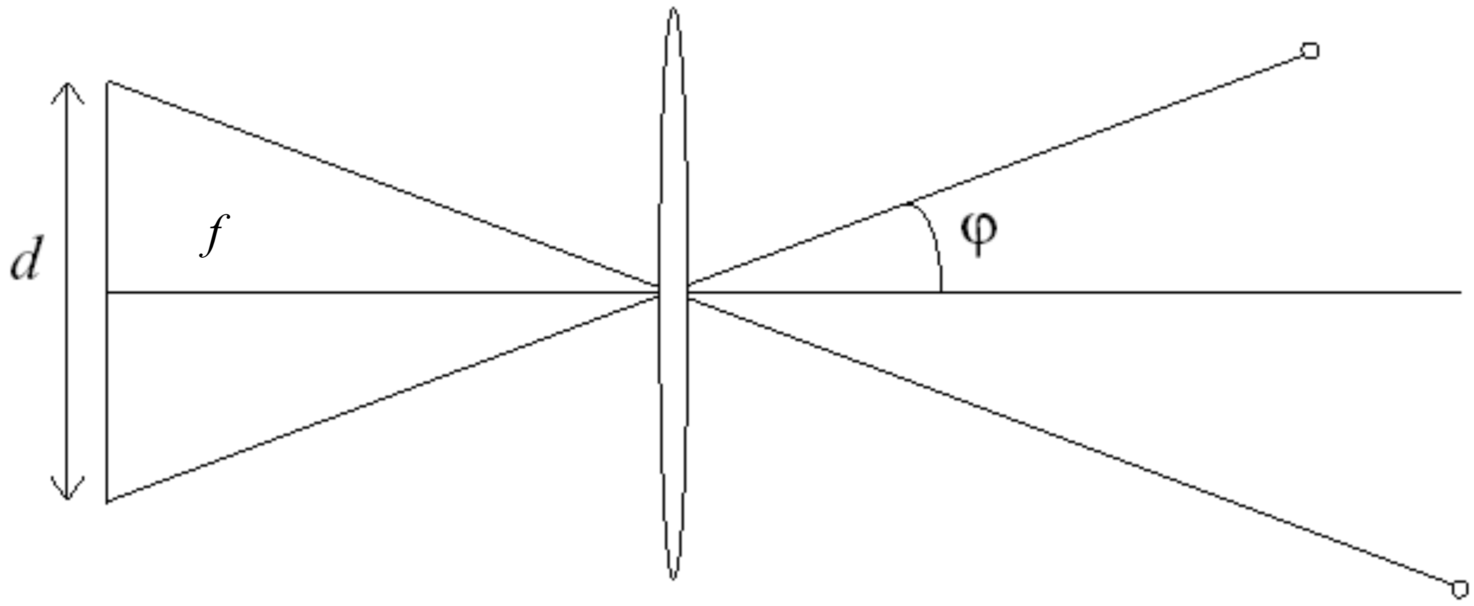
Focal Length of Lenses

- In practice, most cameras use lenses, and the focal length is determined by the physical properties of the lens, such as refraction index, thickness, curvature etc.



Field of View (FOV)

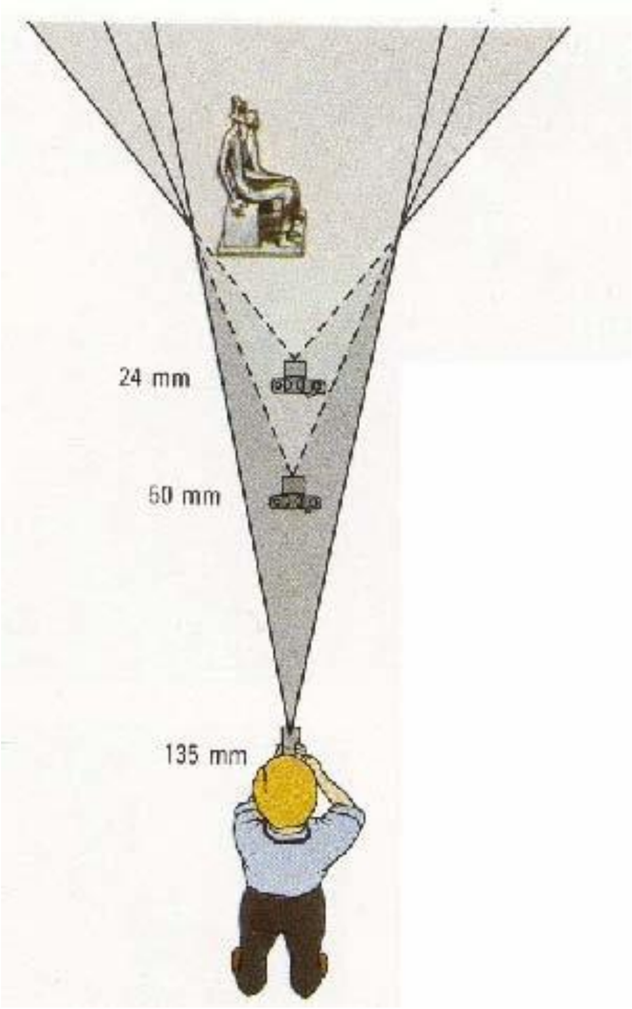
- The field of view is the angular extent of the world observed by the camera.
- Focal length (f), length of the sensor (d):



$$\varphi = \tan^{-1}\frac{d}{2f}$$

- Larger focal length = smaller FOV (assuming a constant sensor size)

Field of View / Focal Length Ambiguity



Large FOV, small f
Camera close to car

Small FOV, large f
Camera far from the car

Field of View / Focal Length Ambiguity

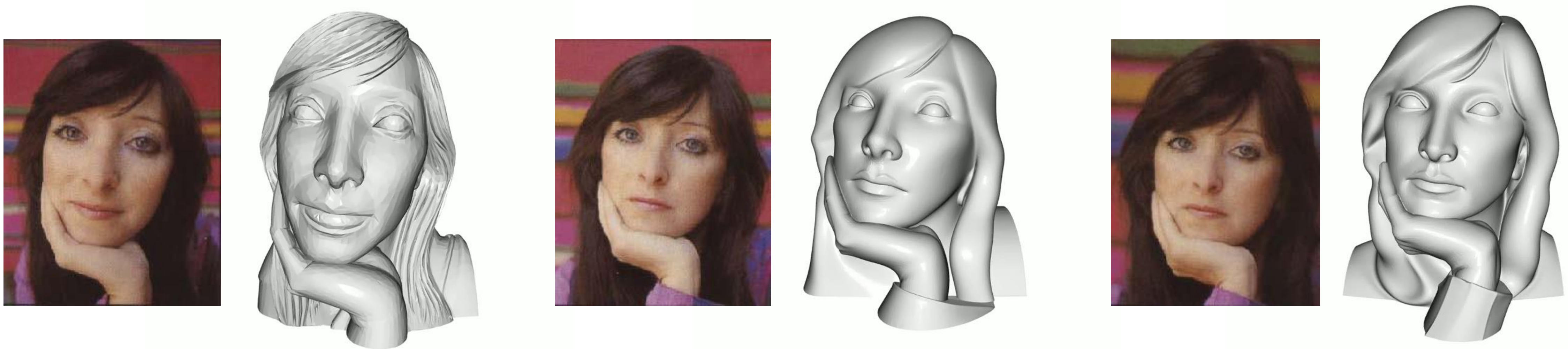
wide-angle

standard

telephoto

- What would happen when reconstructing 3D shapes from a single image with *unknown* focal length/FOV?

Field of View / Focal Length Ambiguity



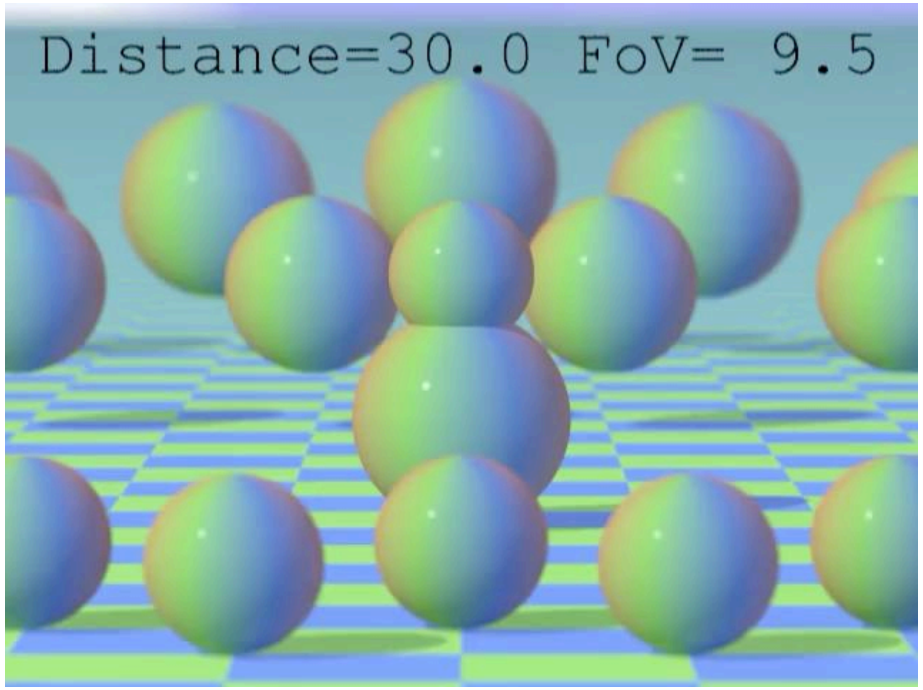
wide-angle standard telephoto

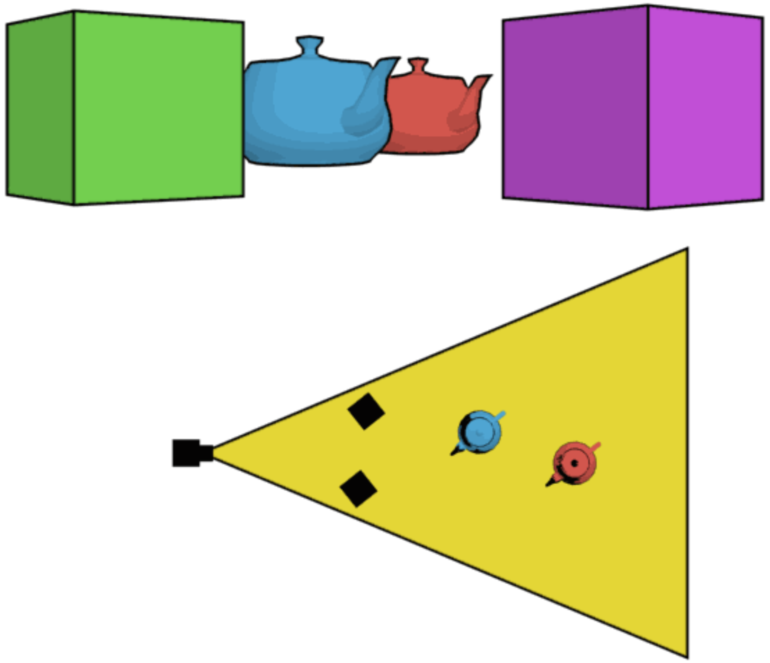
(reconstructed by Hunyuan3D V3.1)

- What would happen when reconstructing 3D shapes from a single image with *unknown* focal length/FOV?

Dolly Zoom Effect

- Continuously adjusting the focal length, while dollying toward or away from the subject.

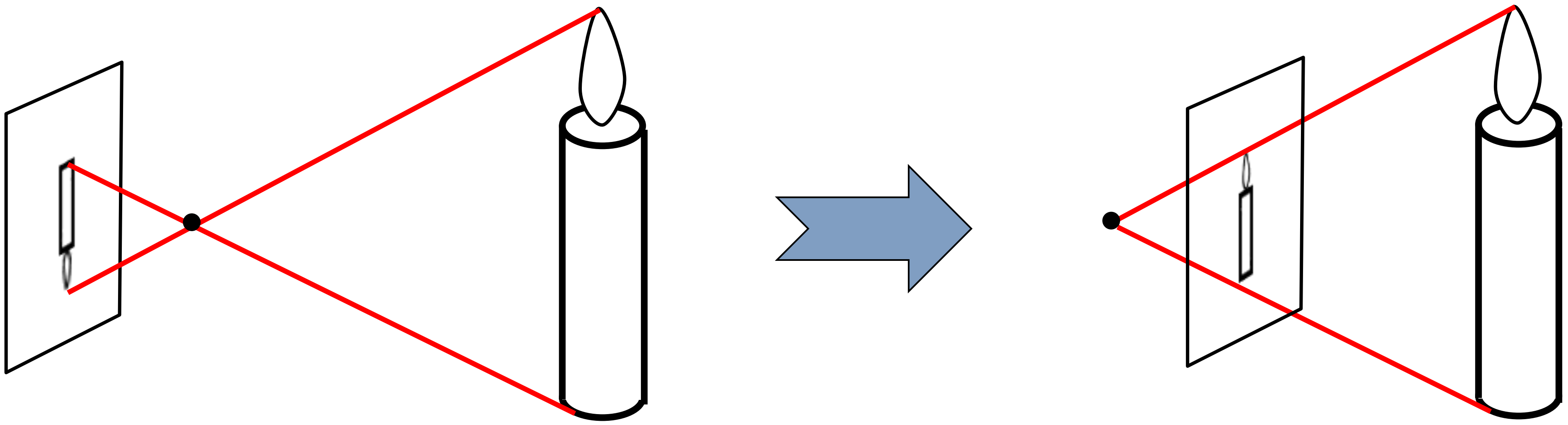




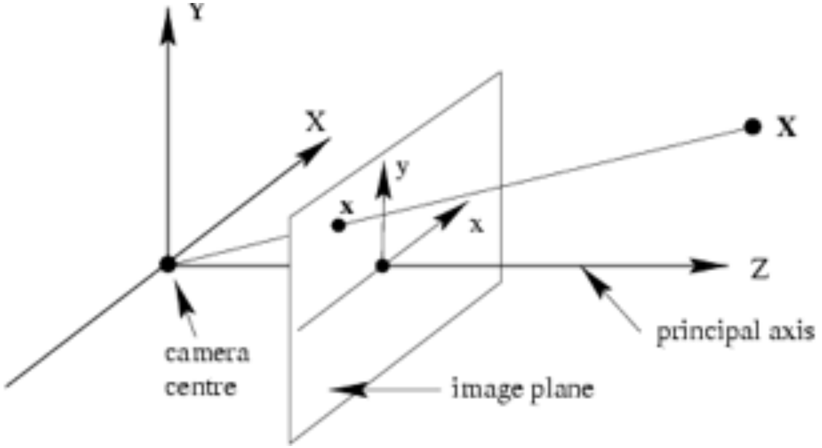
https://en.wikipedia.org/wiki/Dolly_zoom

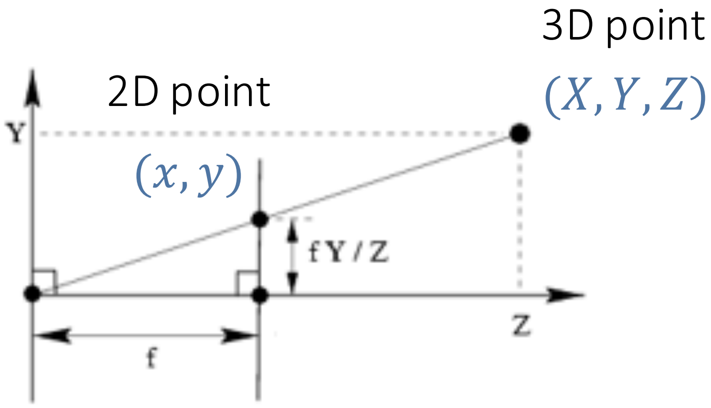
Perspective Projection

- Instead of dealing with an image that is upside down, most of the time we will pretend that the image plane is in front of the camera center.



Perspective Projection





$$(X, Y, Z) \rightarrow (x, y)$$

$$x = f\frac{X}{Z}, y = f\frac{Y}{Z}$$

- Perspective projection is not a linear transformation!
- But most other transformations we will be dealing with are.
- Projective geometry provides a handy mathematical tool to unify them.

Homogeneous Coordinates

- To form homogeneous coordinates from Euclidean coordinates, append 1 as the last entry:

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous coordinates of a 2D point (x, y)

$$(X, Y, Z) \Rightarrow \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

homogeneous coordinates of a 3D point (X, Y, Z)

- To convert homogeneous coordinates to Euclidean coordinates, divide by the last entry:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \Rightarrow (X/W, Y/W, Z/W)$$

- All scalar multiples represent the same point! $\begin{bmatrix} x \\ y \\ w \end{bmatrix} \sim \lambda \begin{bmatrix} x \\ y \\ w \end{bmatrix}, \quad \lambda \neq 0$

Homogeneous Coordinates

2D		3D	
2D point (x, y)	$\mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$	3D point (x, y, z)	$\mathbf{X} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$
2D line $\mathbf{l}$	$\mathbf{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ $\mathbf{l}^\top \mathbf{x} = ax + by + c = 0$	3D plane $\boldsymbol{\pi}$	$\boldsymbol{\pi} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$ $\boldsymbol{\pi}^\top \mathbf{X} = aX + bY + cZ + d = 0$
2 points $\mathbf{x}_1, \mathbf{x}_2$ form a line $\mathbf{l}$	$\mathbf{l} = \mathbf{x}_1 \times \mathbf{x}_2$	3 points $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ form a plane $\boldsymbol{\pi}$	$\boldsymbol{\pi} = \mathbf{x}_1 \times \mathbf{x}_2 \times \mathbf{x}_3$
2 lines $\mathbf{l}_1, \mathbf{l}_2$ intersect at point $\mathbf{x}$	$\mathbf{x} = \mathbf{l}_1 \times \mathbf{l}_2$	3 planes $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3$ intersect at point $\mathbf{X}$	$\mathbf{X} = \boldsymbol{\pi}_1 \times \boldsymbol{\pi}_2 \times \boldsymbol{\pi}_3$

Perspective Projection Matrix

- Projection is a matrix multiplication using homogeneous coordinates:

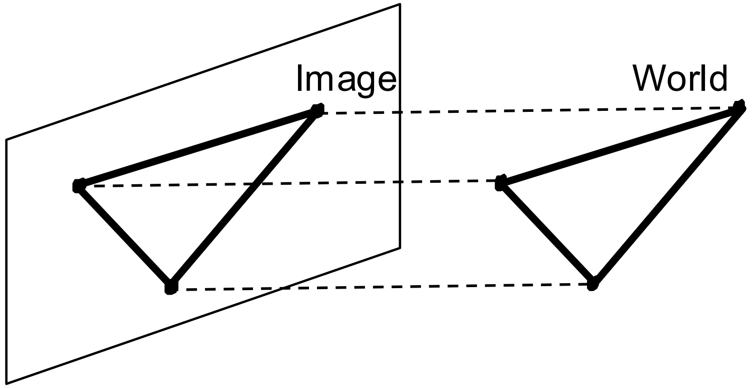
$$\begin{bmatrix} \quad & \quad & \quad & \quad \\ & & & \\ & & & \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} \quad \\ \\ \\ \end{bmatrix} \Rightarrow \left(f\frac{X}{Z}, f\frac{Y}{Z}\right)$$

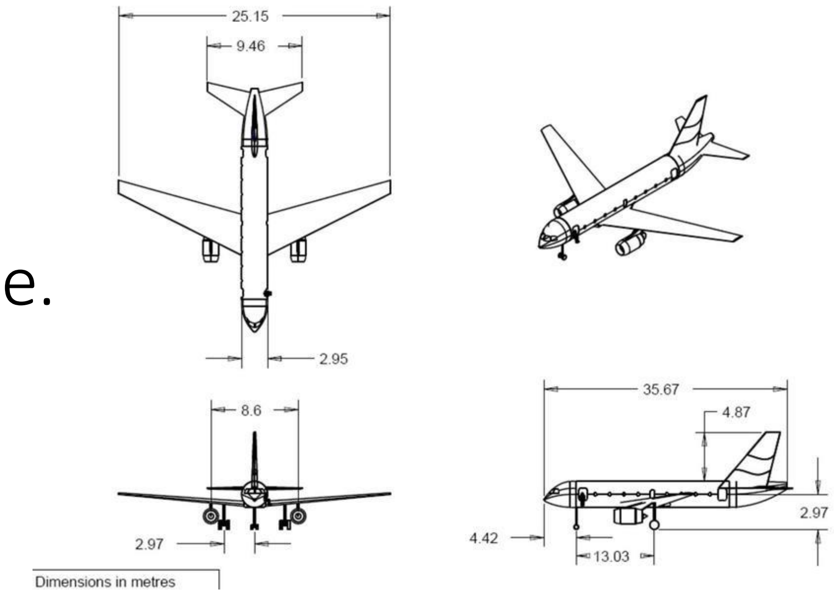
divide by the third coordinate

- What will happen if f is very large?

Orthographic Projection

- A form of *parallel projection* where the projection axis is orthogonal to image plane.





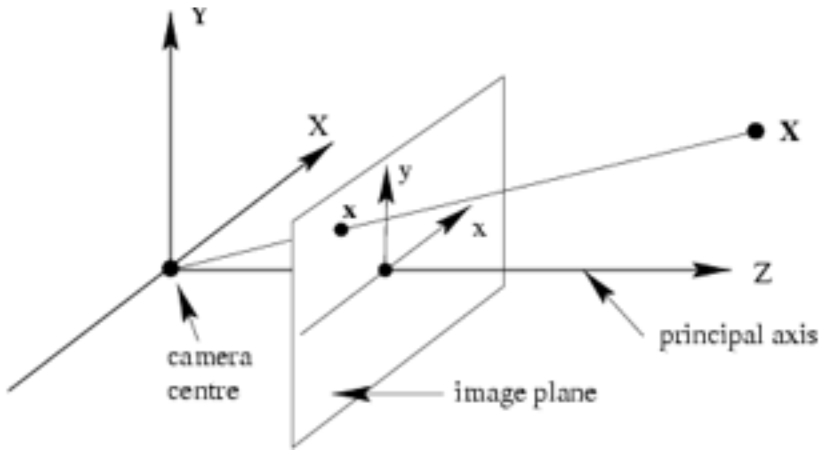
Three-view drawing

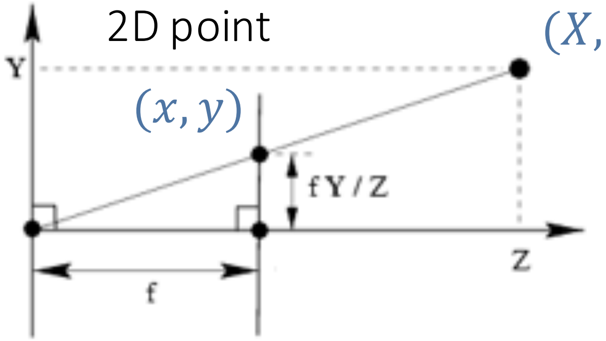
- Assuming projection along the z axis, what's the matrix?

$$\begin{bmatrix} \quad & \quad & \quad & \quad \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} \quad \end{bmatrix}$$

Perspective Projection *in Normalized Coordinates*

(Assuming camera is looking toward z axis)





$$(X, Y, Z) \rightarrow (x, y)$$

$$x = f\frac{X}{Z}, y = f\frac{Y}{Z}$$

$$\underbrace{\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}}_{\boldsymbol{x}} \cong \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \underbrace{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\boldsymbol{P}} \underbrace{\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}}_{\boldsymbol{X}}$$

Homogeneous coordinates of image point

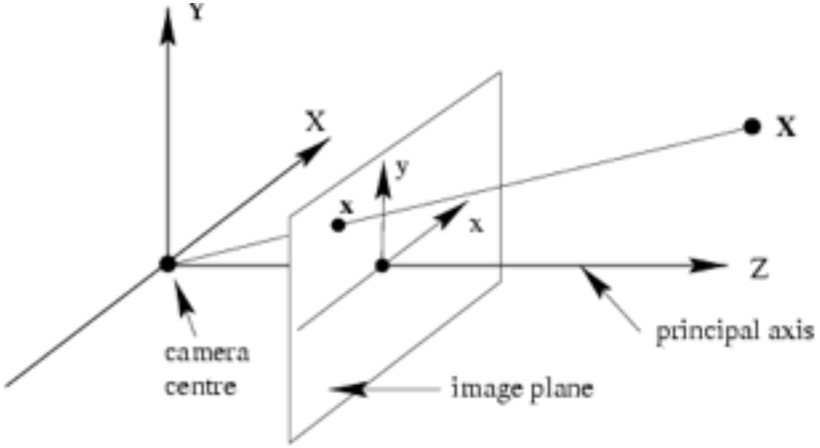
Camera projection matrix

Homogeneous coordinates of 3D point

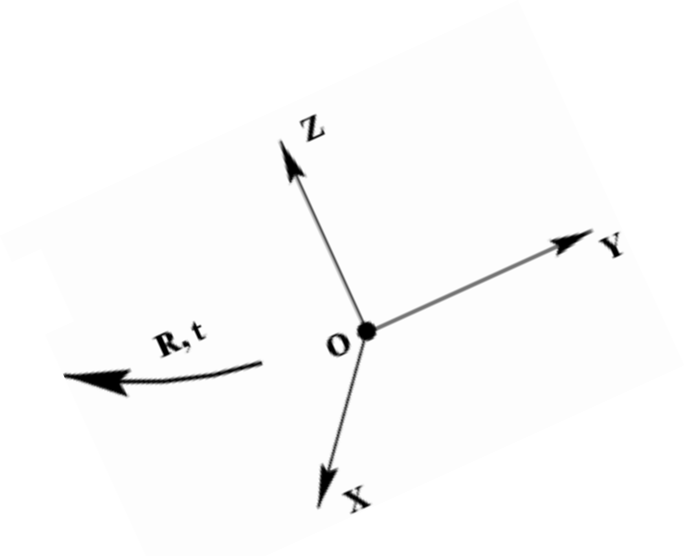
$$\boldsymbol{x} \cong \boldsymbol{PX}$$

Equality up to scale

Camera Calibration



camera coordinate system



world coordinate system

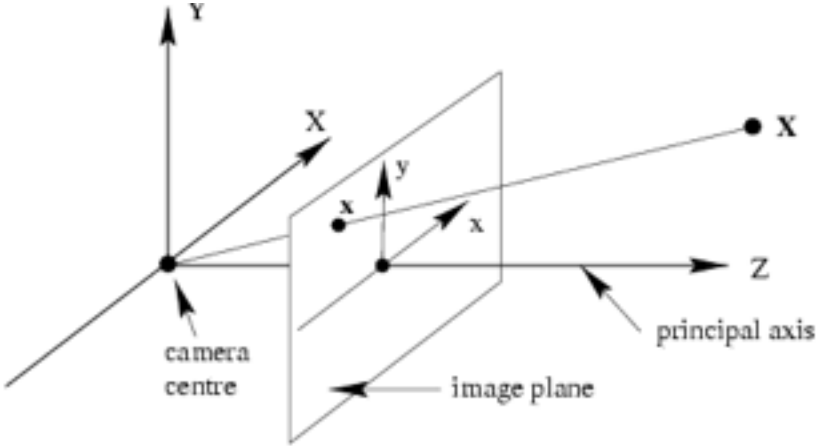
- **Camera calibration**: figuring out transformation from *world* coordinate system to *image* coordinate system

(2D point $\boldsymbol{x}$ (3x1)) $\cong$ (Camera to pixel coord. trans. matrix $\boldsymbol{K}$ (3x3)) (*Canonical* projection matrix $[\boldsymbol{I} \mid \boldsymbol{0}]$ (3x4)) (World to camera coord. trans. matrix $\begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix}$ (4x4)) (3D point $\boldsymbol{X}$ (4x1))

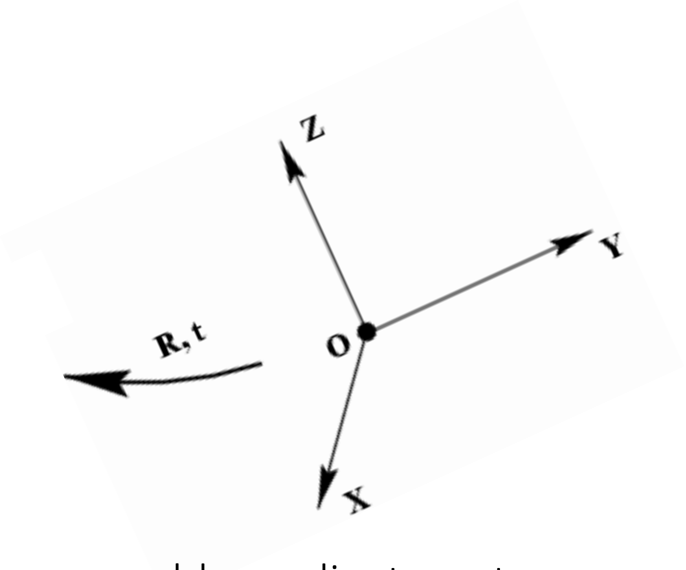
***Intrinsic* camera parameters**: principal point, scaling factors

***Extrinsic* camera parameters**: rotation, translation

Camera Calibration



camera coordinate system



world coordinate system

- **Camera calibration**: figuring out transformation from *world* coordinate system to *image* coordinate system

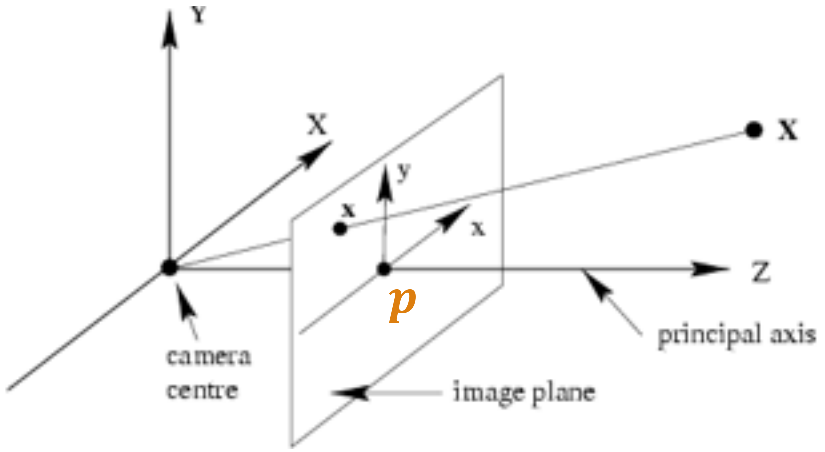
(2D point $\boldsymbol{x}$ (3x1)) $\cong$ (Camera to pixel coord. trans. matrix $\boldsymbol{K}$ (3x3)) (*Canonical* projection matrix $[\boldsymbol{I} \mid \boldsymbol{0}]$ (3x4)) (World to camera coord. trans. matrix $\begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix}$ (4x4)) (3D point $\boldsymbol{X}$ (4x1))

$$\underbrace{}_{\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]}$$

$\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]$

General camera projection matrix

Intrinsic Parameters – Principal Point



camera coordinate system

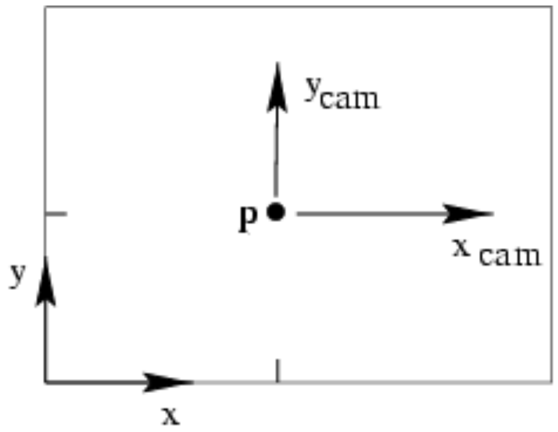
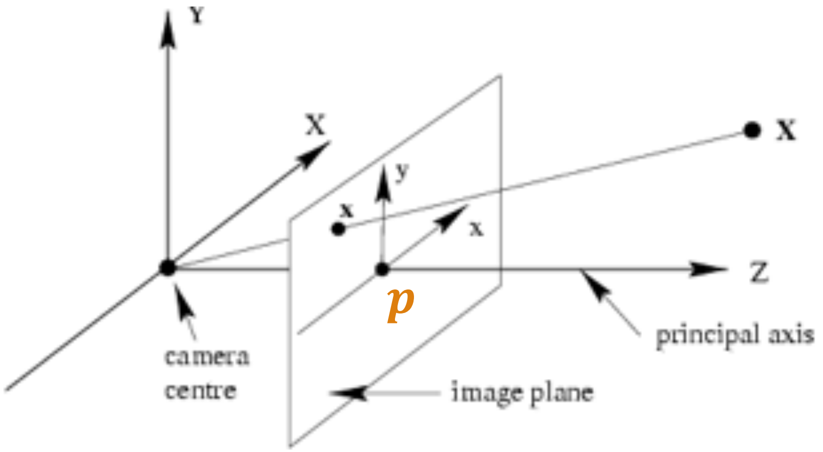


image coordinate system

Principal point (p): point where principal axis intersects the image plane

- In the *normalized* coordinate system, the origin is at the principal point.
- In the *image* coordinate system, the origin is in the corner.

Intrinsic Parameters – Principal Point



camera coordinate system

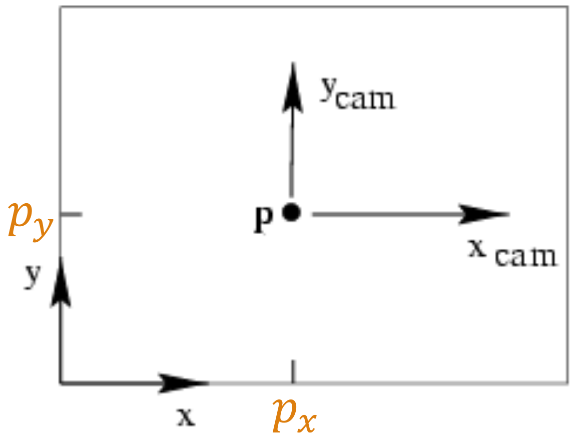


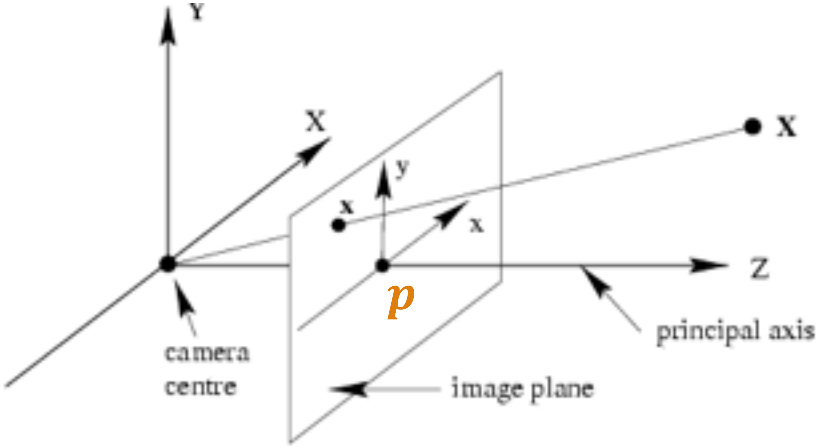
image coordinate system

$$x = f\frac{X}{Z} + p_x, \qquad y = f\frac{Y}{Z} + p_y$$

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{bmatrix} & & & \\ & & & \\ & & & \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

calibration matrix ***K***

Intrinsic Parameters – Scaling



camera coordinate system

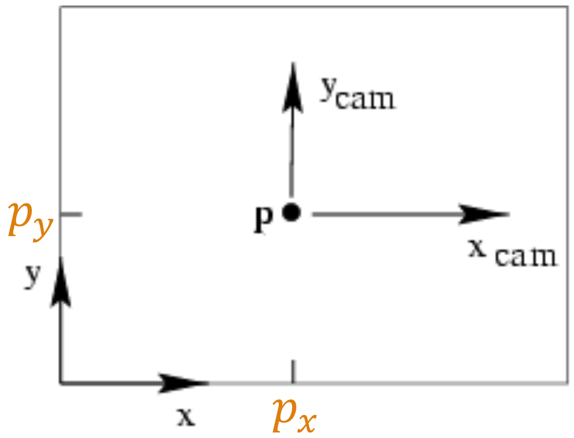


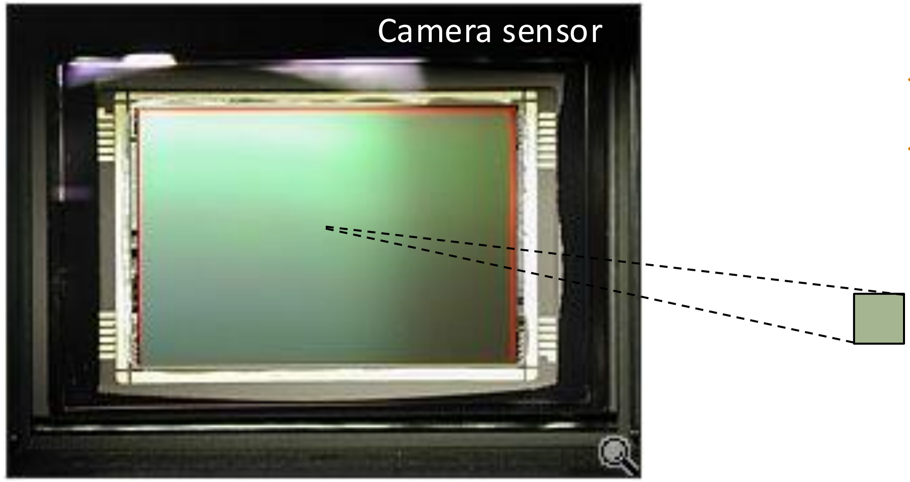
image coordinate system

$$x = f\frac{X}{Z} + p_x, \qquad y = f\frac{Y}{Z} + p_y$$

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \xleftarrow[\text{pixels/m}]{\alpha} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

pixels m calibration matrix ***K***

Intrinsic Parameters – Scaling



m_x pixels/m in horizontal direction
m_y pixels/m in vertical direction

Pixel size (m): $\frac{1}{m_x} \times \frac{1}{m_y}$

Scaling factors

$$\begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

pixels/m

Calibration matrix **K** in metric units

$$\begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

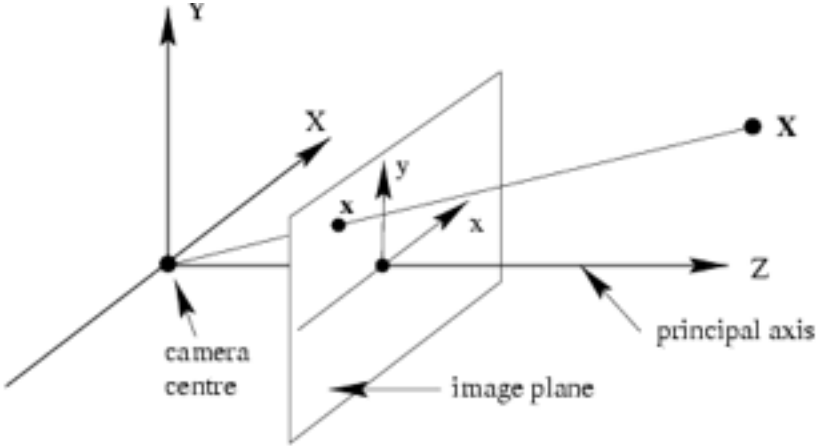
m

Calibration matrix **K** in pixel units

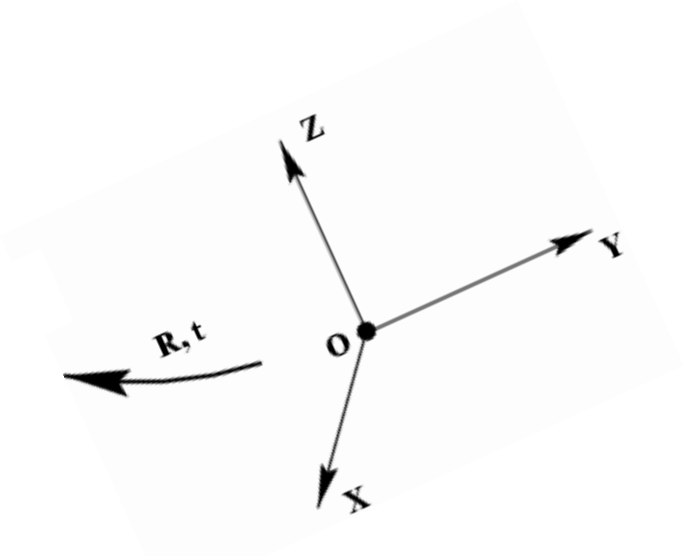
$$= \begin{bmatrix} \alpha_x & 0 & \beta_x \\ 0 & \alpha_y & \beta_y \\ 0 & 0 & 1 \end{bmatrix}$$

pixels

Camera Calibration



camera coordinate system



world coordinate system

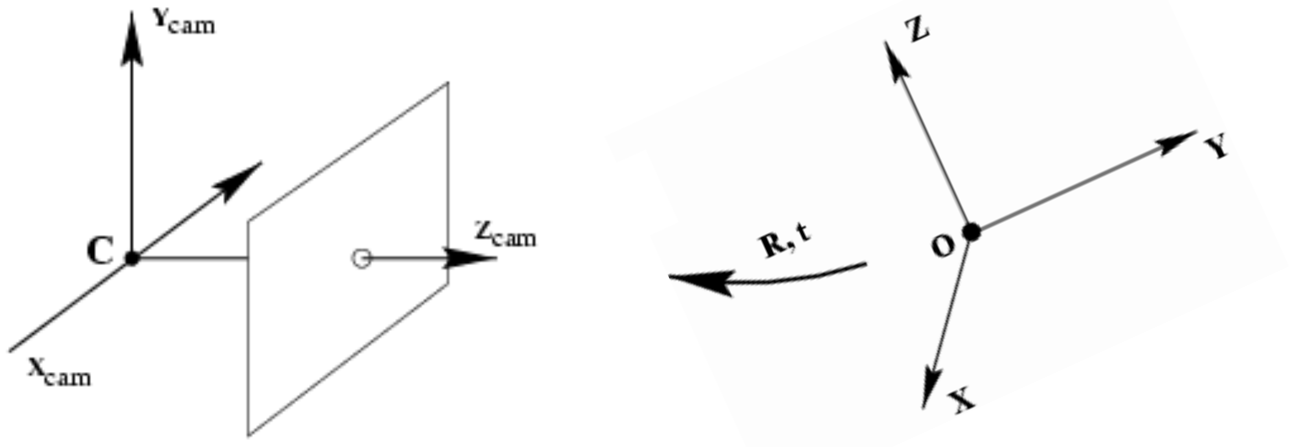
- **Camera calibration**: figuring out transformation from *world* coordinate system to *image* coordinate system

2D point $\boldsymbol{x}$ (3x1)	$\cong$	Camera to pixel coord. trans. matrix $\boldsymbol{K}$ (3x3)	*Canonical* projection matrix $[\boldsymbol{I} \mid \boldsymbol{0}]$ (3x4)	World to camera coord. trans. matrix $\begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix}$ (4x4)	3D point $\boldsymbol{X}$ (4x1)

$$\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]$$

General camera projection matrix

Extrinsic Parameters



camera coordinate system

world coordinate system

- In *non-homogeneous* coordinates, the transformation from world to normalized camera coordinate system is given by:

$$\tilde{X}_{\text{cam}} = R(\tilde{X} - \tilde{C}) = R\tilde{X} + t$$

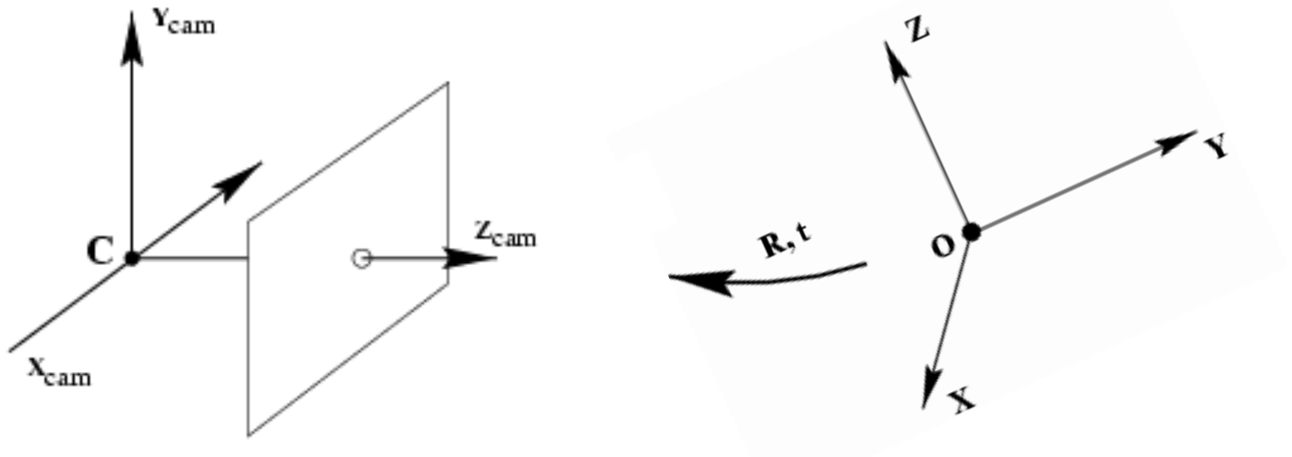
coords. of point in normalized camera frame

3x3 rotation matrix

coords. of a point in world frame

coords. of camera center in world frame

Extrinsic Parameters



camera coordinate system

world coordinate system

In non-homogeneous coordinates:

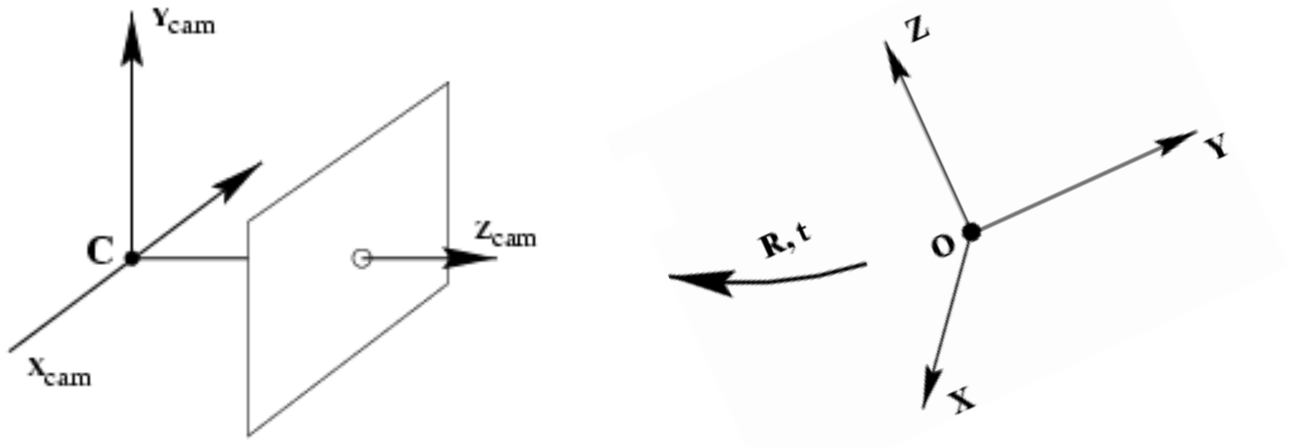
$$\tilde{\boldsymbol{X}}_{\text{cam}} = \boldsymbol{R}\tilde{\boldsymbol{X}} + \boldsymbol{t}$$

In homogeneous coordinates:

$$\begin{pmatrix} \tilde{\boldsymbol{X}}_{\text{cam}} \\ 1 \end{pmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{pmatrix} \tilde{\boldsymbol{X}} \\ 1 \end{pmatrix}$$

3D transformation matrix (4 x 4)

Extrinsic Parameters



camera coordinate system

world coordinate system

In non-homogeneous coordinates:

$$\widetilde{\boldsymbol{X}}_{\text{cam}} = \boldsymbol{R}\widetilde{\boldsymbol{X}} + \boldsymbol{t}$$

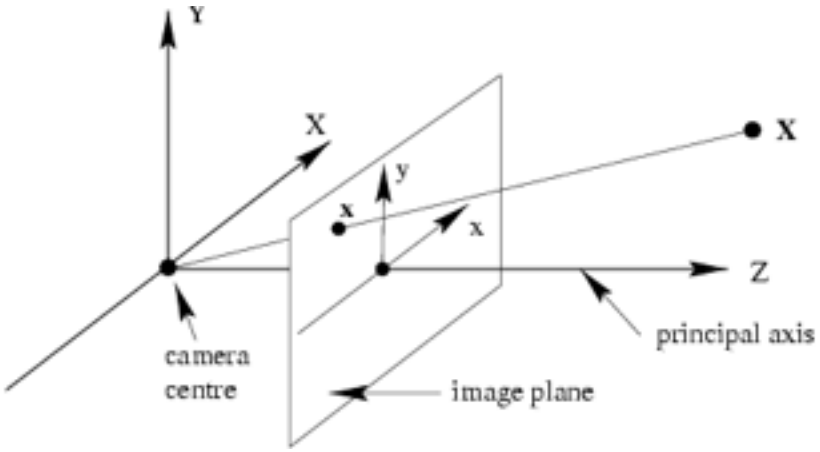
In homogeneous coordinates:

$$\boldsymbol{X}_{\text{cam}} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \boldsymbol{X}$$

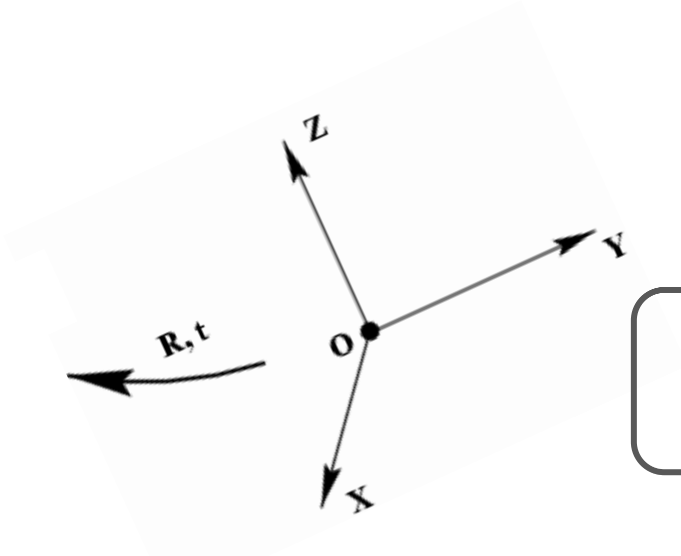
3D transformation matrix (4 x 4)

And then to image coordinates: $\boldsymbol{x} \cong \boldsymbol{K}[\boldsymbol{I}|\boldsymbol{0}]\boldsymbol{X}_{\text{cam}}$

Camera Calibration



camera coordinate system



world coordinate system

$$\boxed{x \cong K[R|t]X}$$

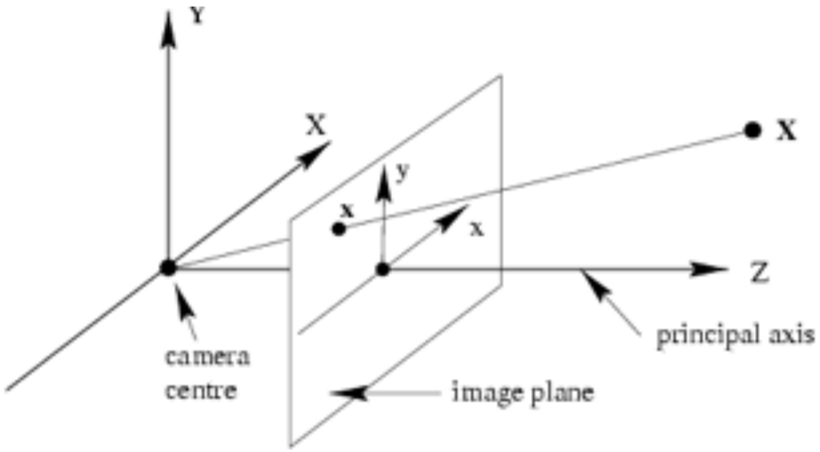
- **Camera calibration**: figuring out transformation from *world* coordinate system to *image* coordinate system

$$\begin{pmatrix}\text{2D}\\\text{point}\\x\\(3\text{x}1)\end{pmatrix} \cong \underbrace{\begin{pmatrix}\text{Camera to}\\\text{pixel coord.}\\\text{trans. matrix}\\K\ (3\text{x}3)\end{pmatrix}\begin{pmatrix}\textit{Canonical}\\\text{projection matrix}\\[I\,|\,0]\ (3\text{x}4)\end{pmatrix}\begin{pmatrix}\text{World to}\\\text{camera coord.}\\\text{trans. matrix}\\\begin{bmatrix}R & t\\0^T & 1\end{bmatrix}\ (4\text{x}4)\end{pmatrix}}_{P = K[R|t]}\begin{pmatrix}\text{3D}\\\text{point}\\X\\(4\text{x}1)\end{pmatrix}$$

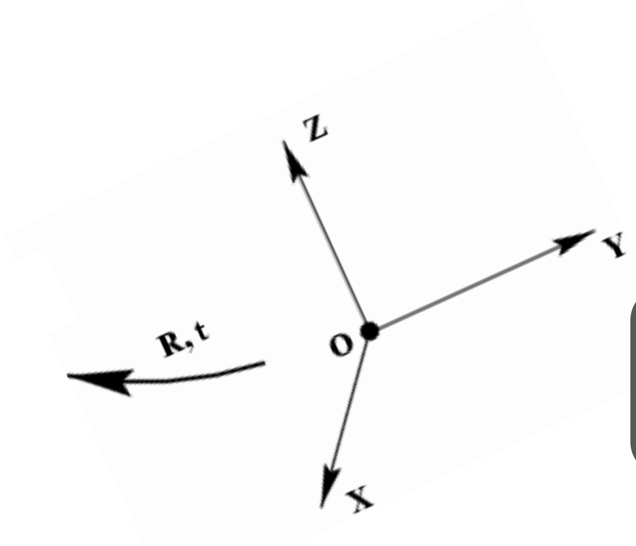
General camera projection matrix

Slide Source: S. Lazebnik

Camera Calibration



camera coordinate system



world coordinate system

$$\boxed{\boldsymbol{x} \cong \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]\boldsymbol{X}}$$

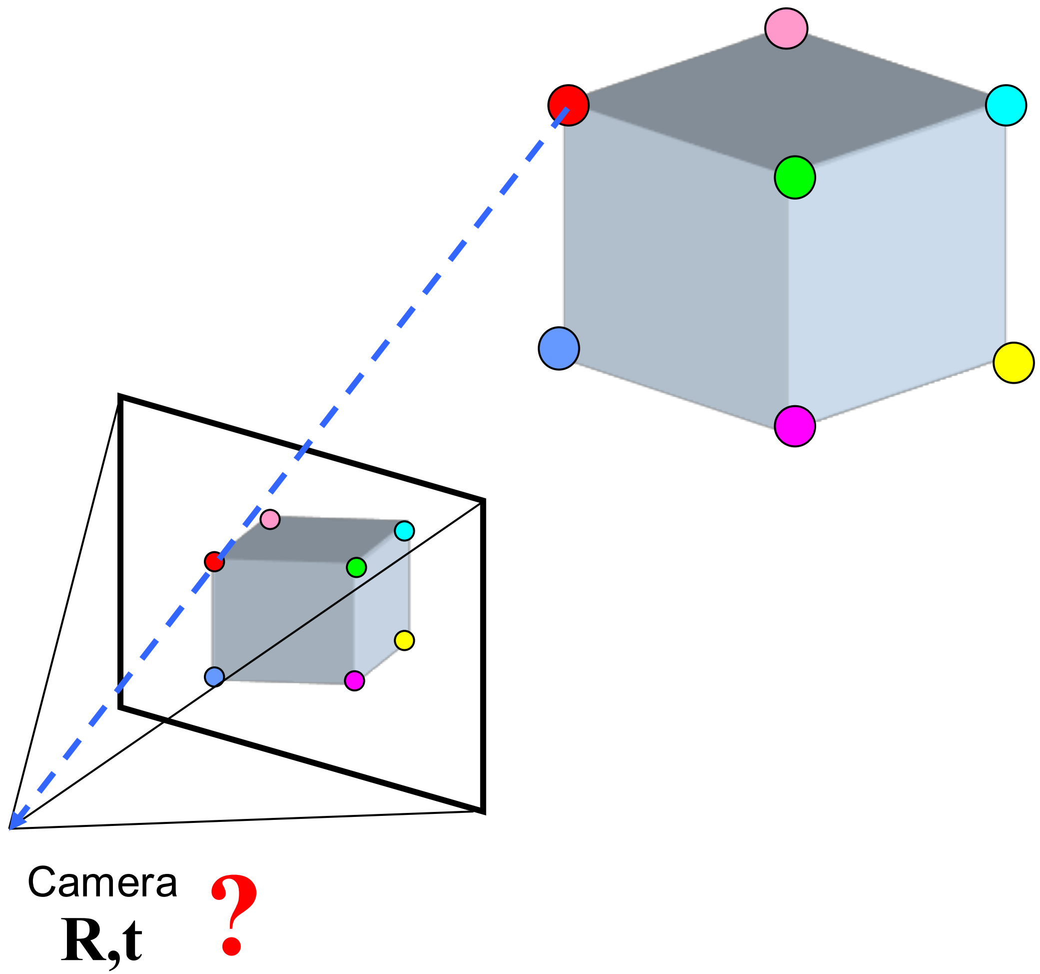
- **Camera calibration**: figuring out transformation from *world* coordinate system to *image* coordinate system

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \leftarrow$$

We could solve this as a linear system, then perform QR decomposition to get $\boldsymbol{K}$, $\boldsymbol{R}$, $\boldsymbol{t}$. (not robust in practice; sensitive to noise)

Perspective-n-Point (PnP)

$$\boldsymbol{x} \cong \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]\boldsymbol{X}$$



Known:

- n 3D points $\boldsymbol{X}$
- Corresponding 2D pixel coordinates $\boldsymbol{x}$
- Camera intrinsics $\boldsymbol{K}$

Solve for:

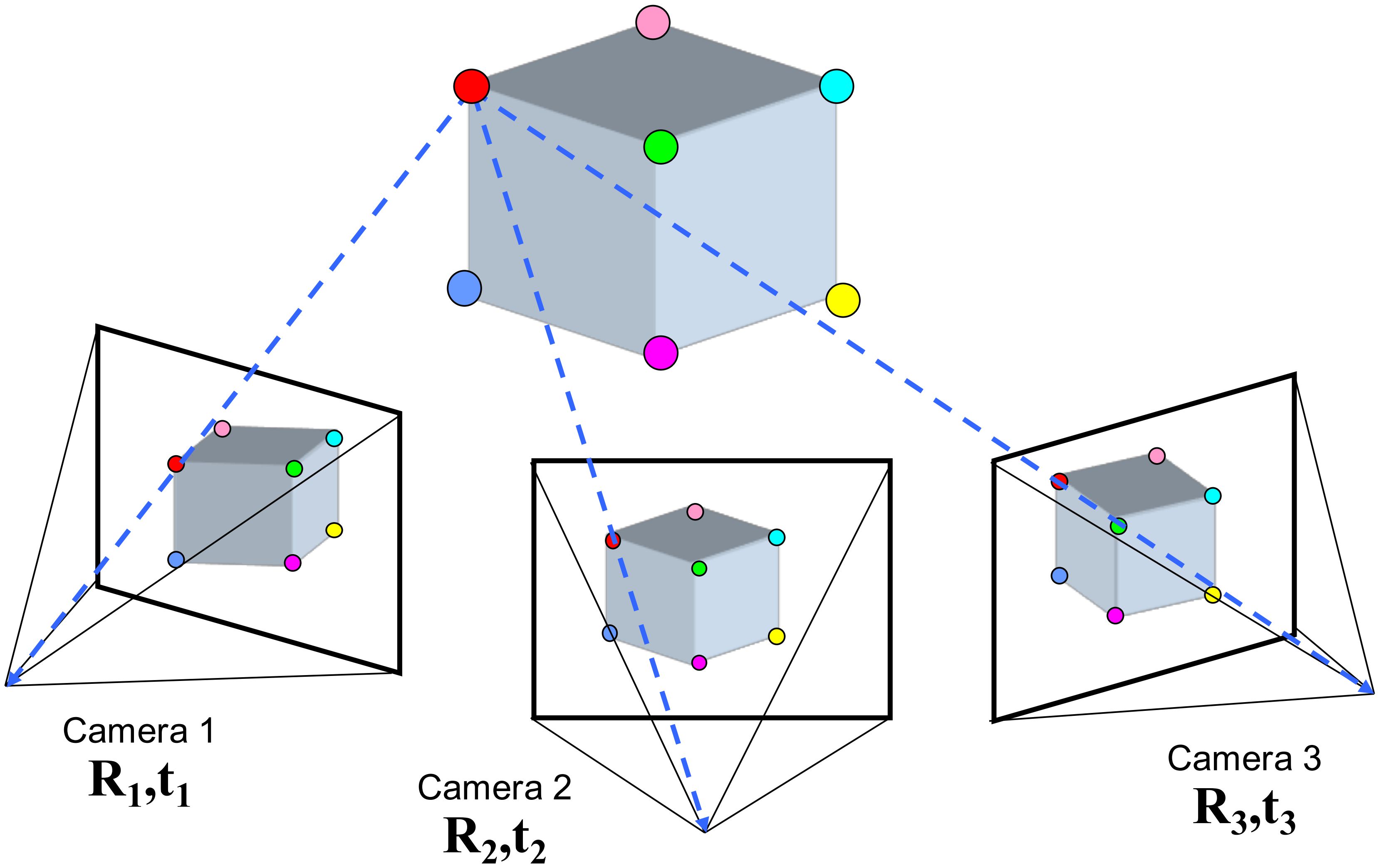
- Camera pose (extrinsics) $\boldsymbol{R}$, $\boldsymbol{t}$

Q: How many points at least do we need?

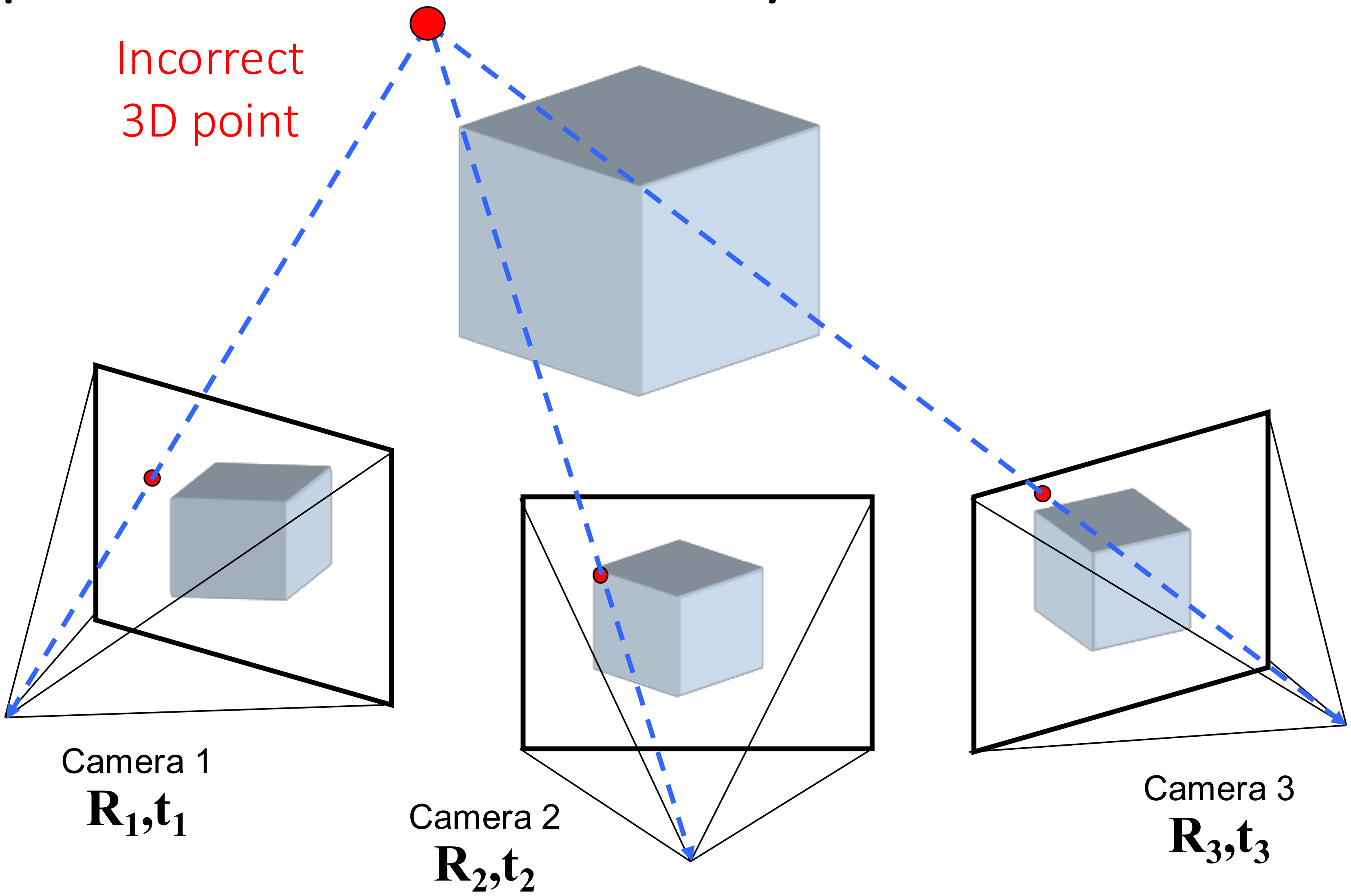
A: n ≥ 3 (in general). We have 6 DoF unknowns $\boldsymbol{R}$, $\boldsymbol{t}$. Each point provides 2 constraints (2 equations for x and y).

Part 2 – Multi-view Geometry

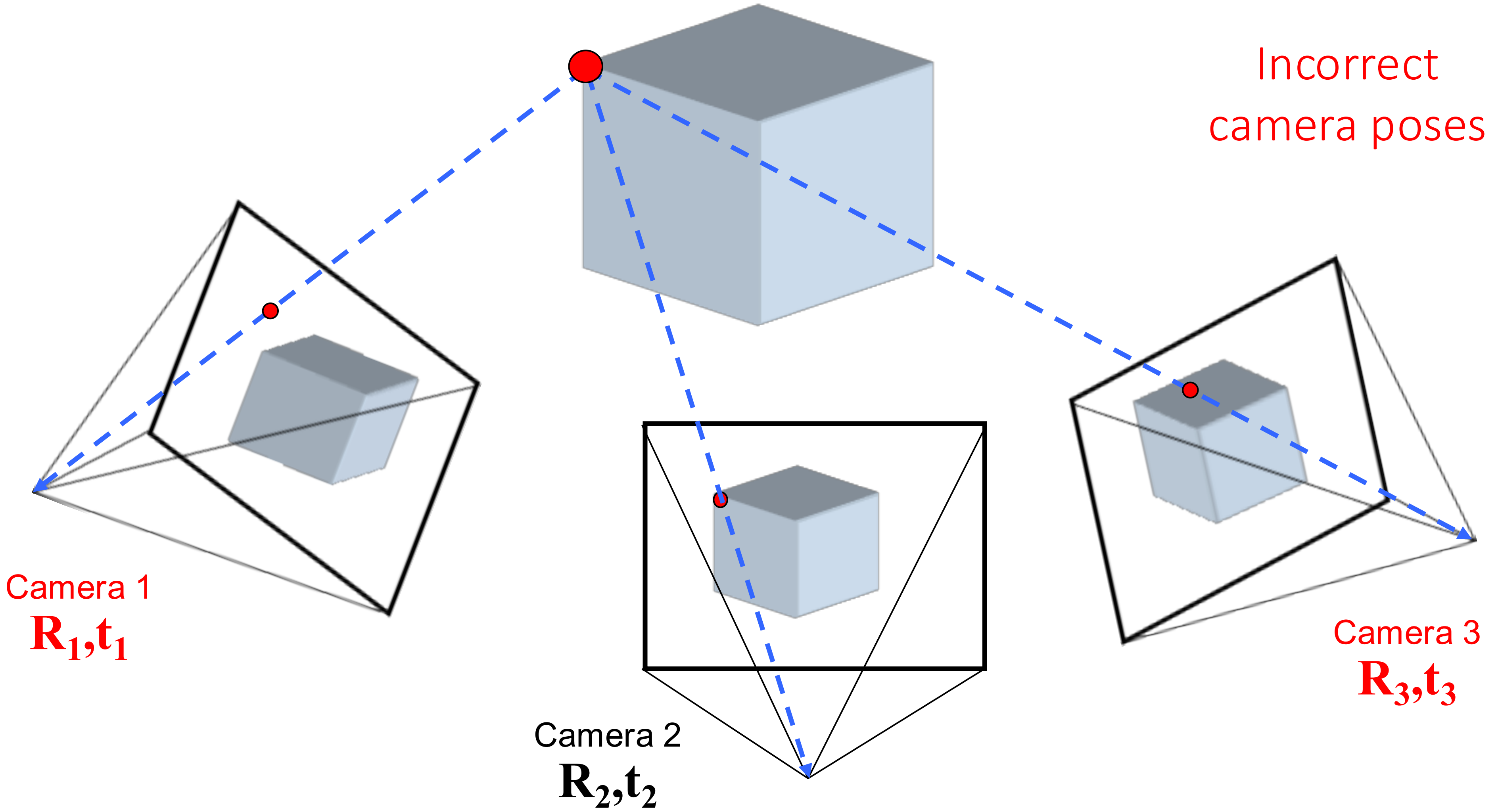
Multiple View Geometry



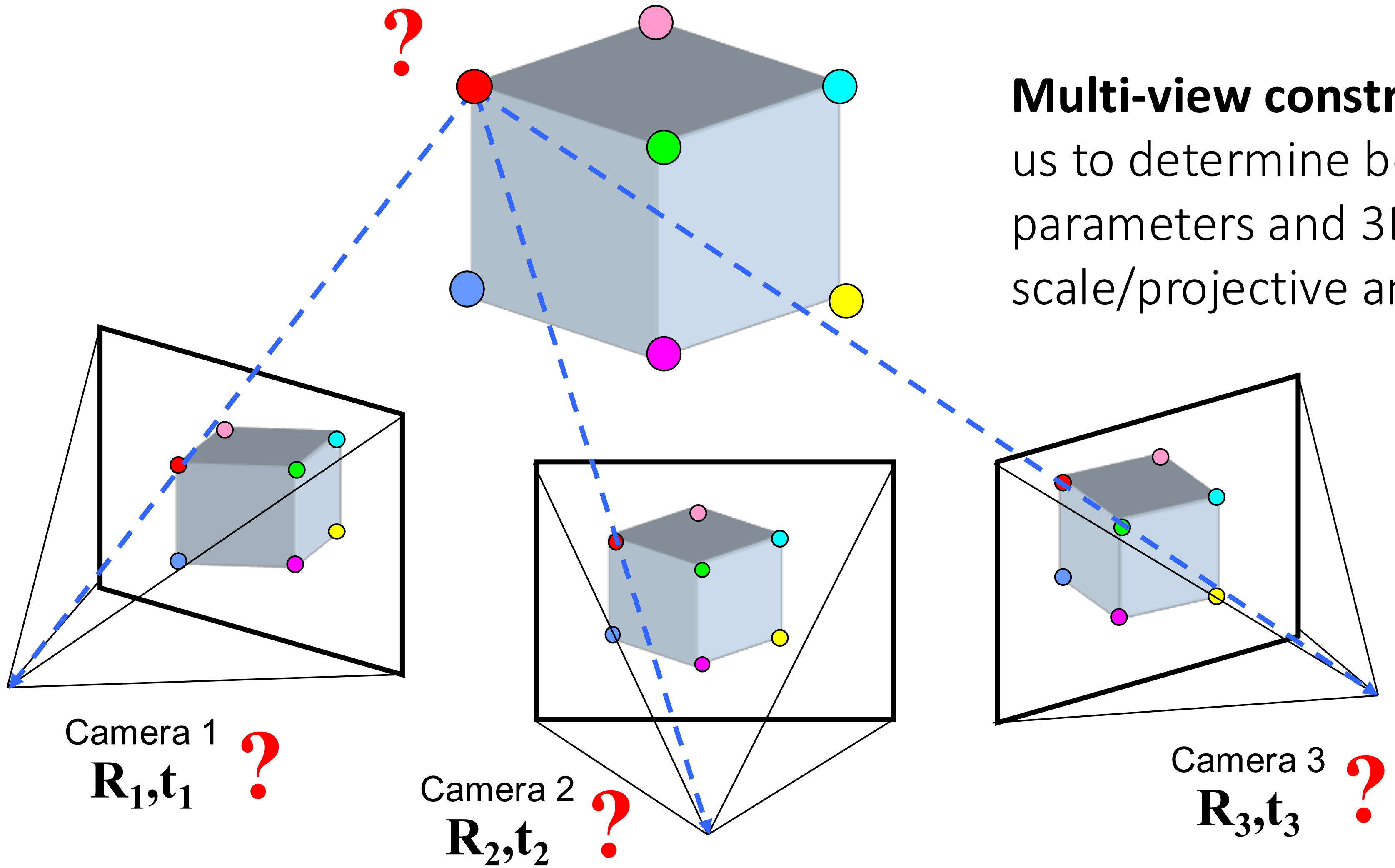
Multiple View Geometry



Multiple View Geometry

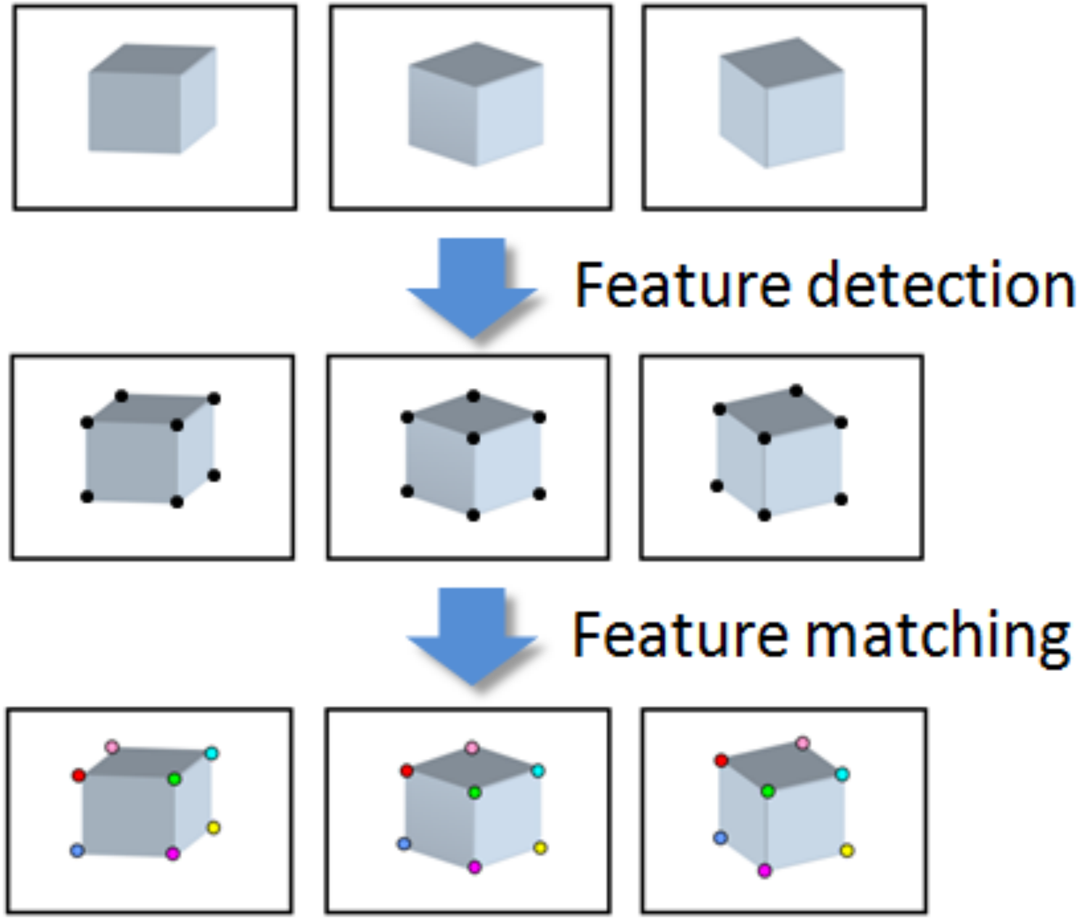


Multiple View Geometry

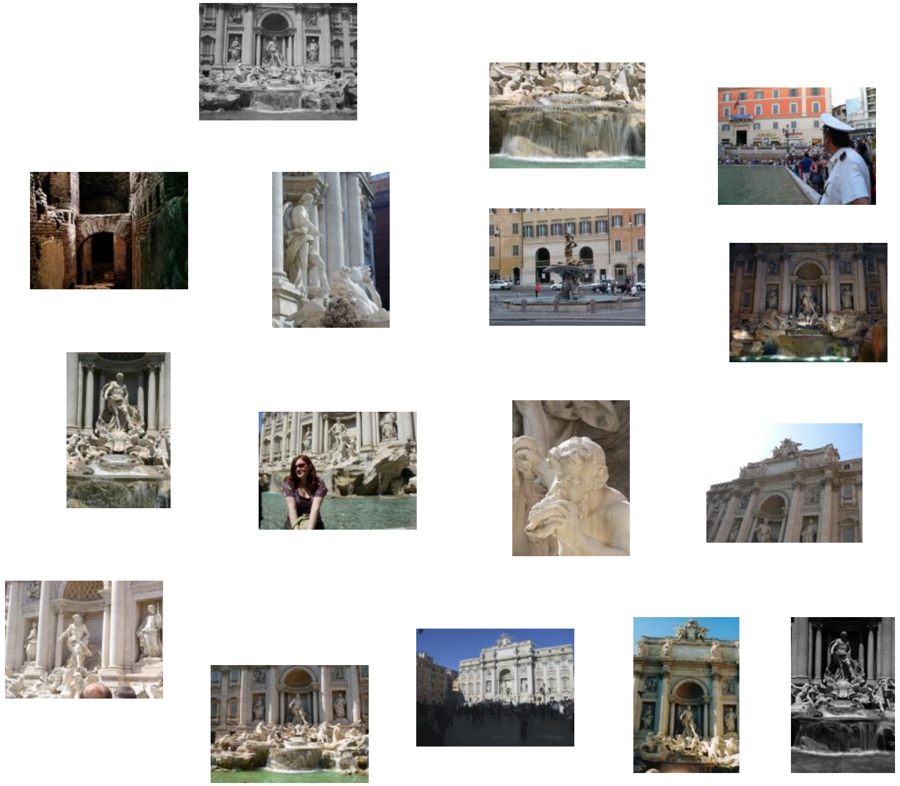


Multi-view constraints allow us to determine both camera parameters and 3D (up to scale/projective ambiguities)

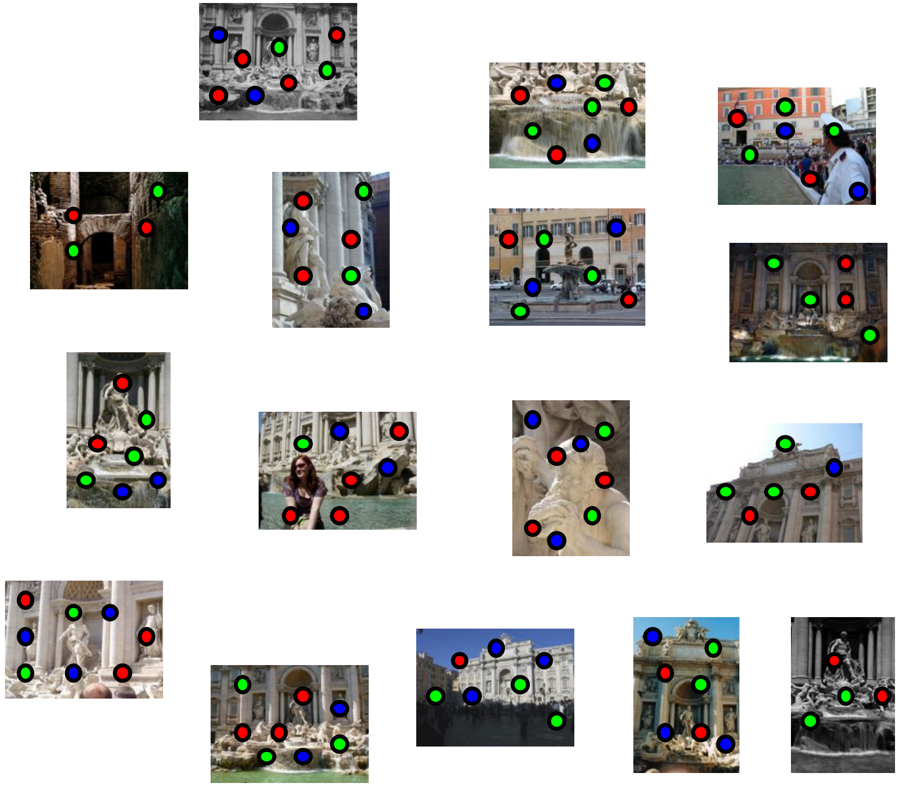
Estimating Correspondences



Estimating Correspondences

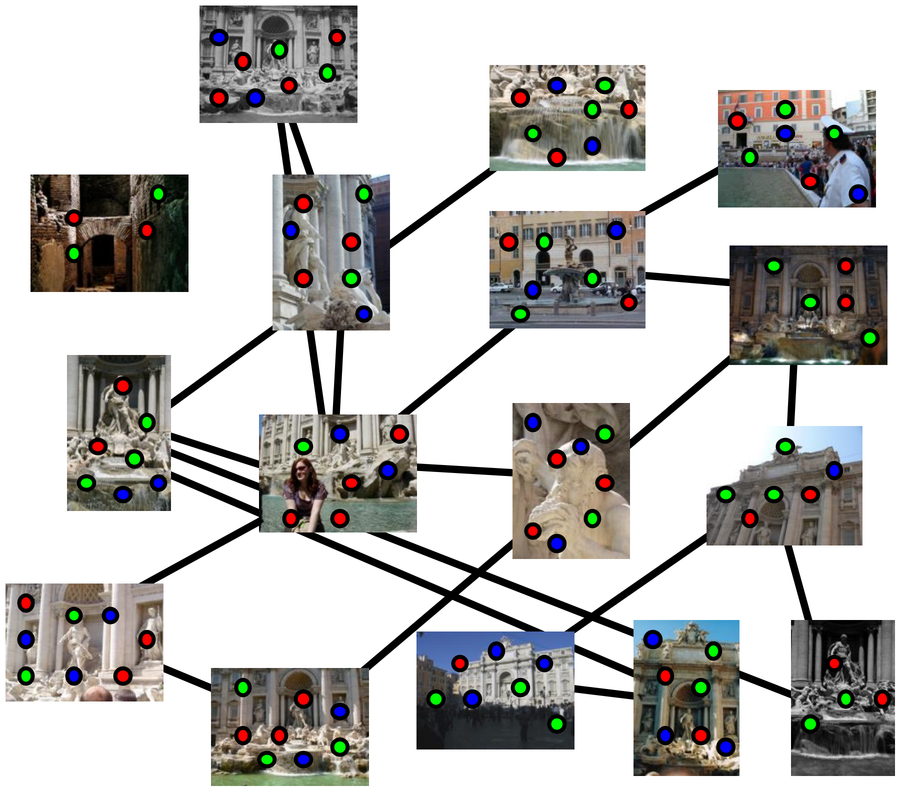


Estimating Correspondences



- **Detect feature points**
  - SIFT descriptors [Lowe, 2004]
  - Learned features – SuperPoint [DeTone, 2018])

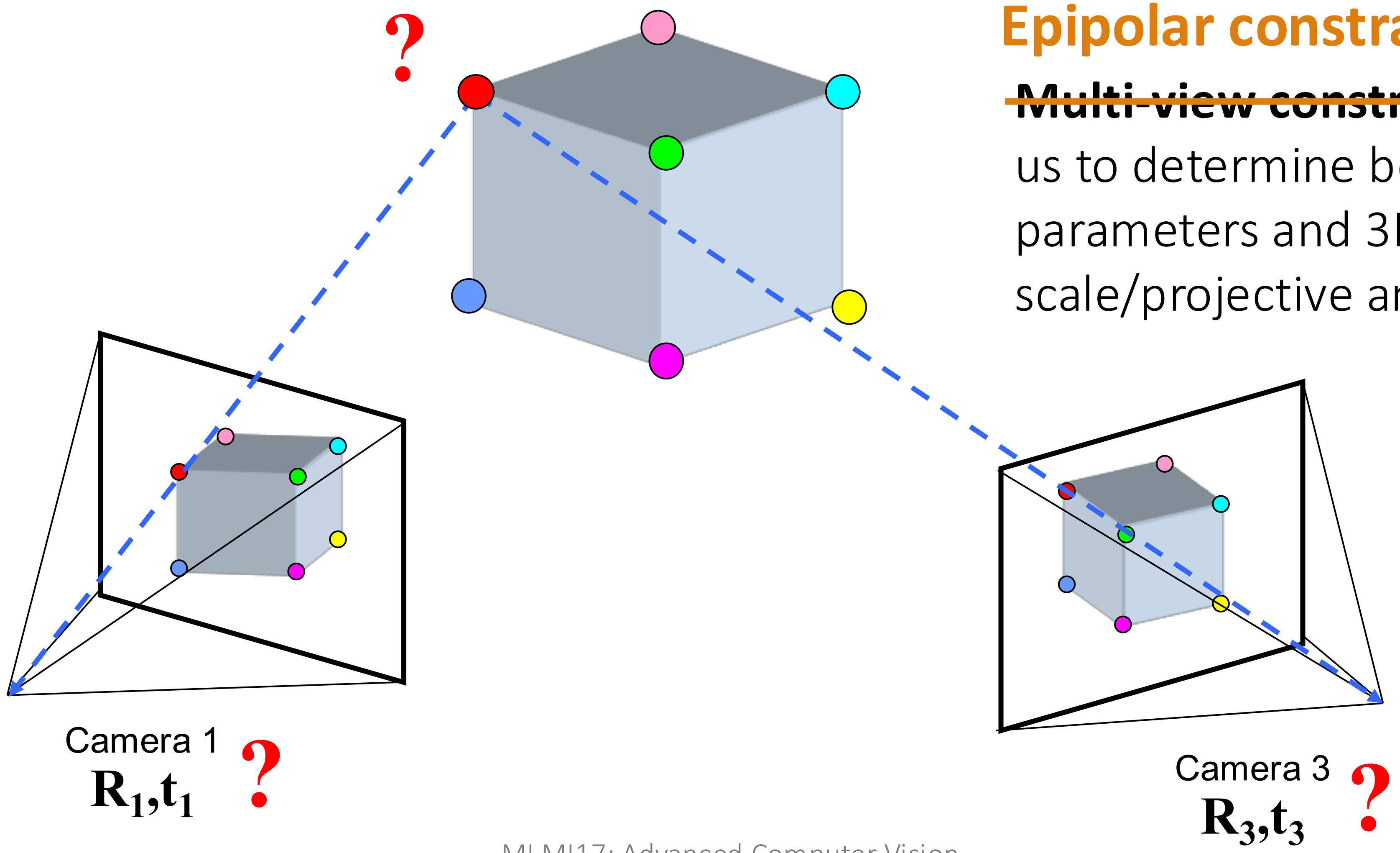
Estimating Correspondences



- **Detect feature points**
  - SIFT descriptors [Lowe, 2004]
  - Learned features – SuperPoint [DeTone, 2018])
- **Match features** between pair of images
  - Use RANSAC to filter outliers
  - Learned matching – SuperGlue [Sarlin, 2020]

Let's Start with Two Views – Epipolar Geometry

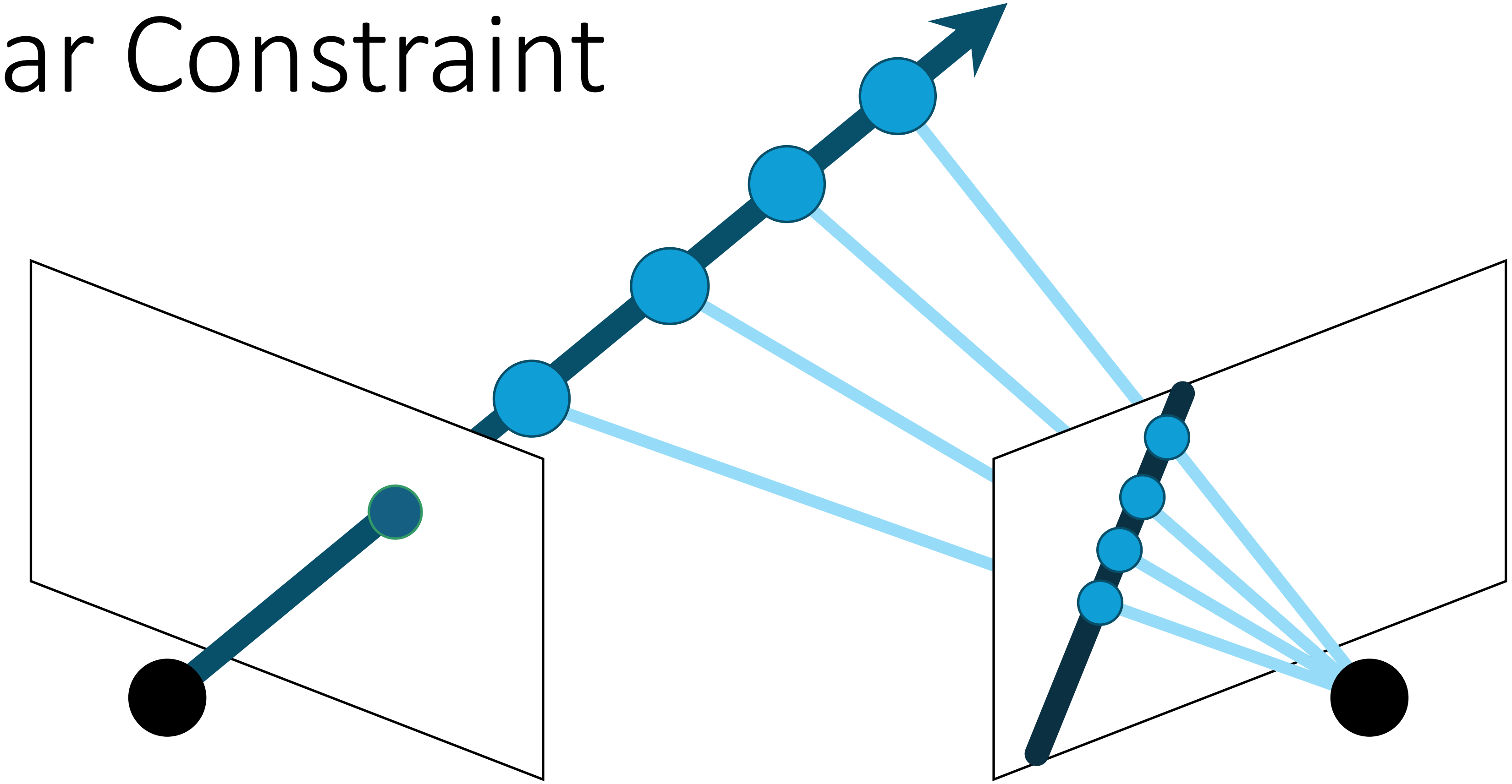
"epi-pole" ≈ upon the pole/pivot



Epipolar constraints

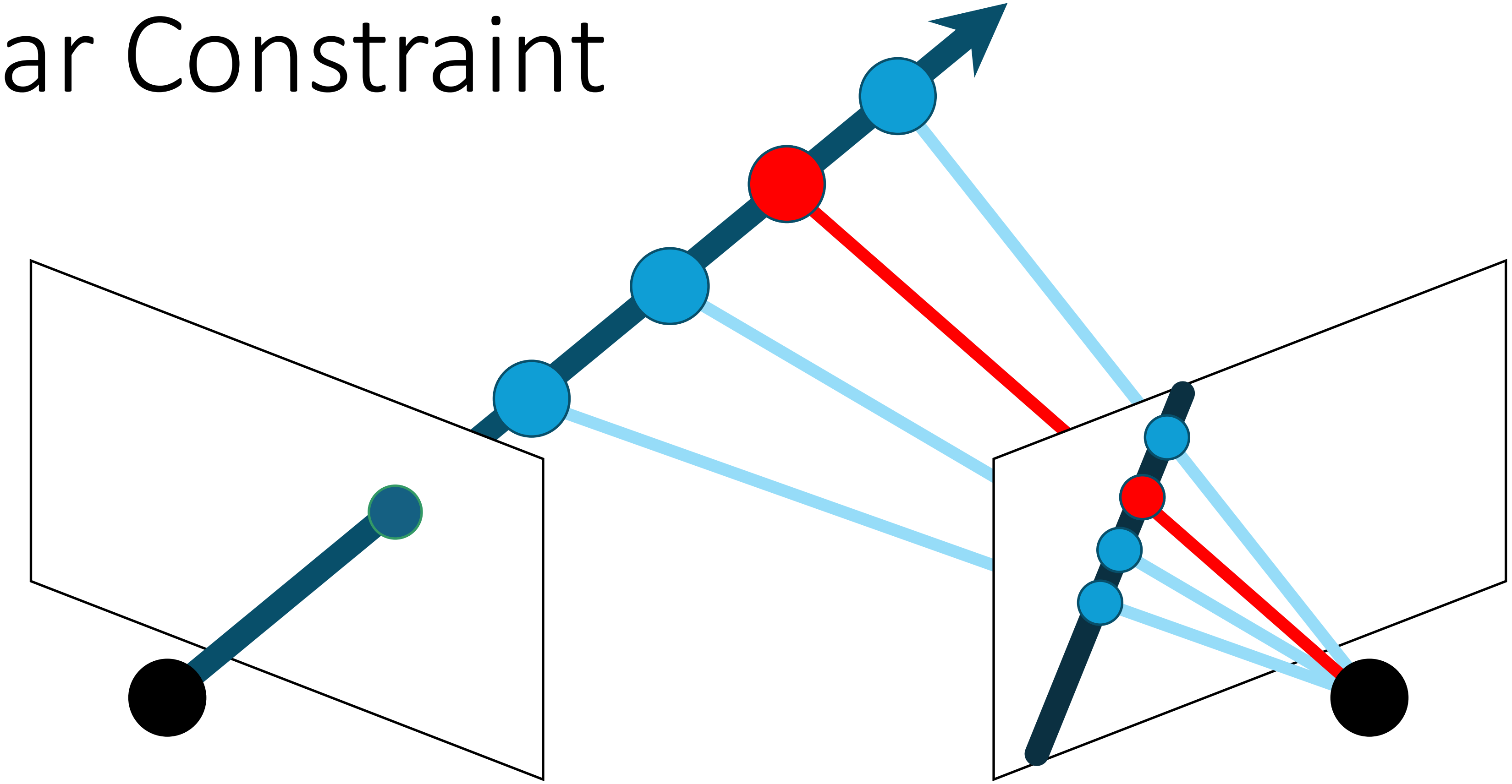
~~**Multi-view constraints**~~ allow us to determine both camera parameters and 3D (up to scale/projective ambiguities)

Epipolar Constraint



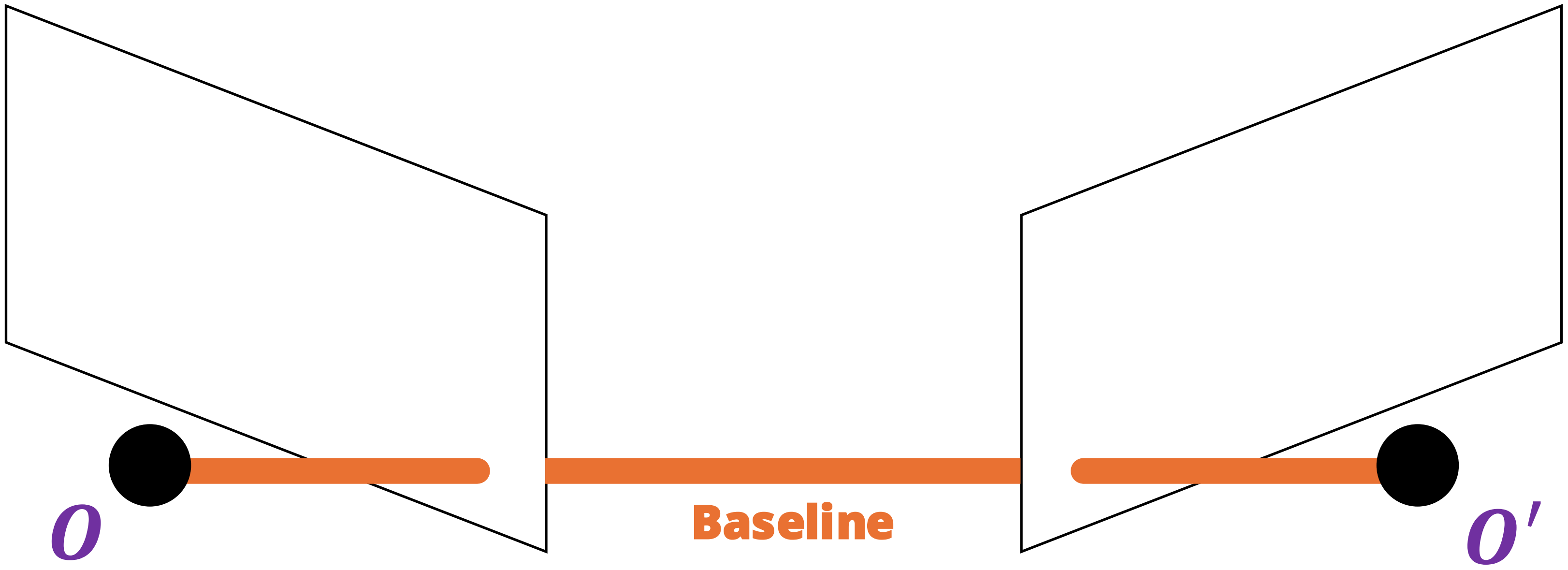
- Projecting points on a ray to another camera forms a **line**, i.e., the corresponding point on the second image *must* lie on this line

Epipolar Constraint



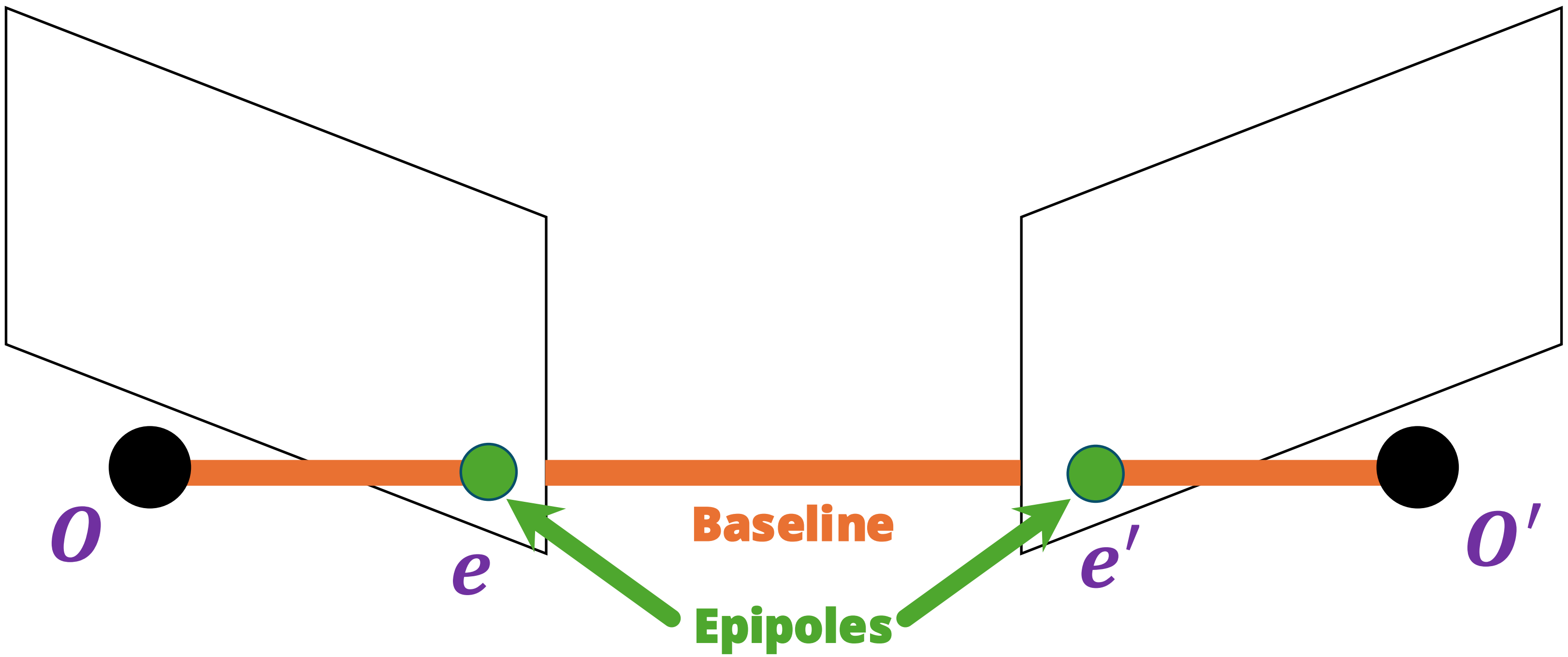
- Projecting points on a ray to another camera forms a **line**, i.e., the corresponding point on the second image *must* lie on this line
- Hence, if the *matched* 2D point is found along this line, we can determine its 3D location

Epipolar Geometry



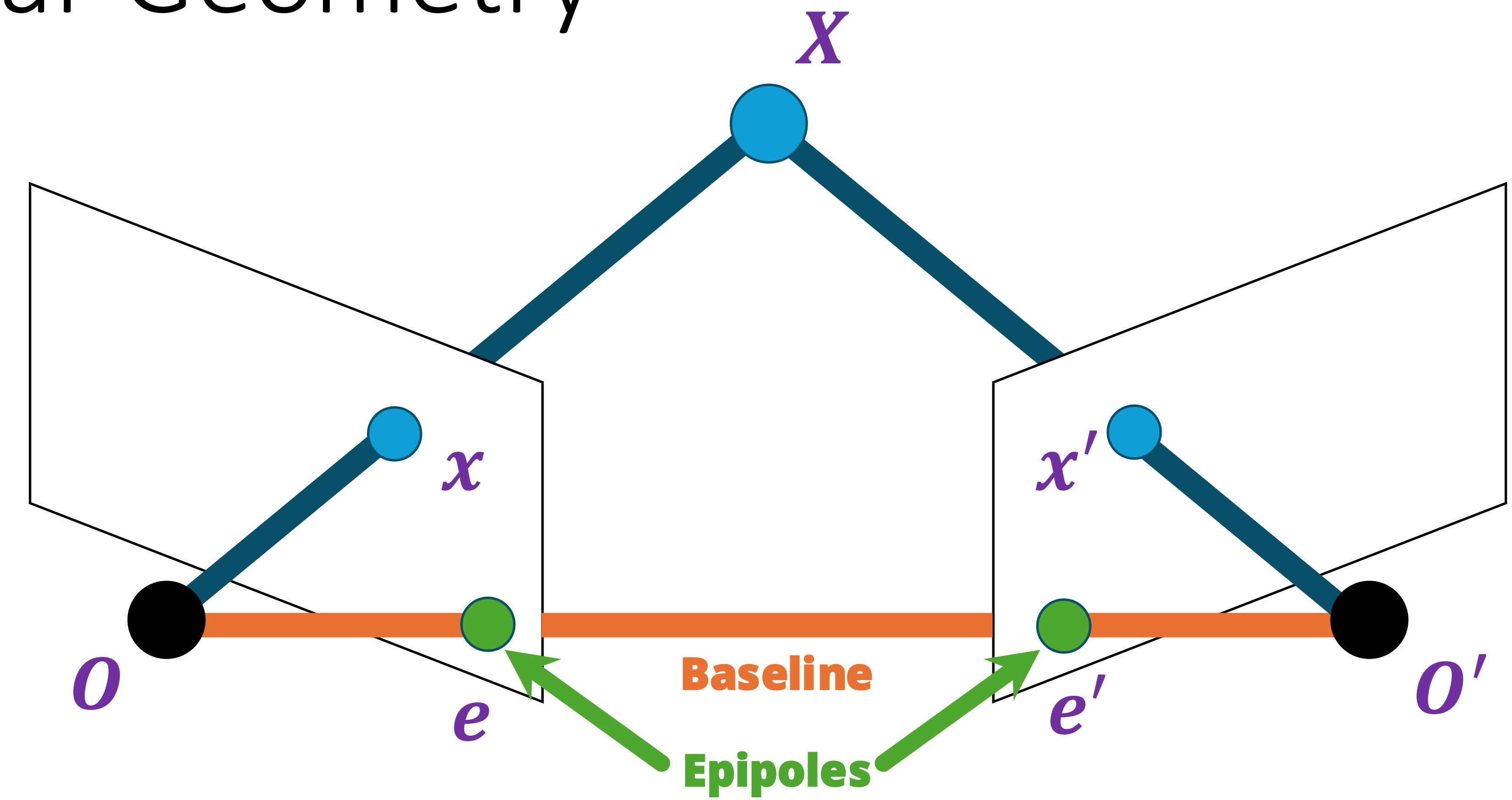
- Suppose we have two cameras with centers O, O'
- The **baseline** is the line connecting the origins

Epipolar Geometry



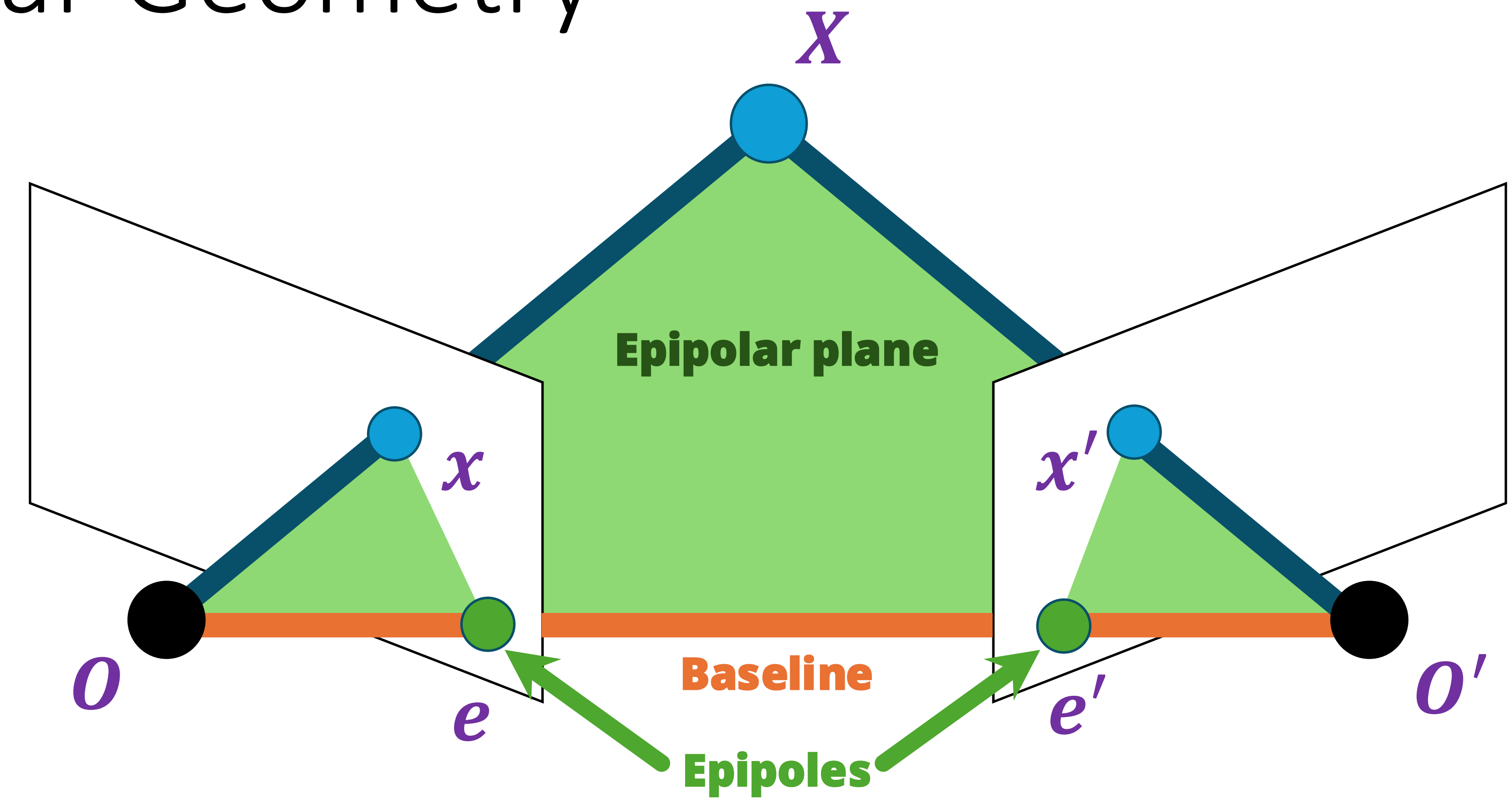
- **Epipoles** e, e' are where the baseline intersects the image planes, or projections of the other camera in each view

Epipolar Geometry



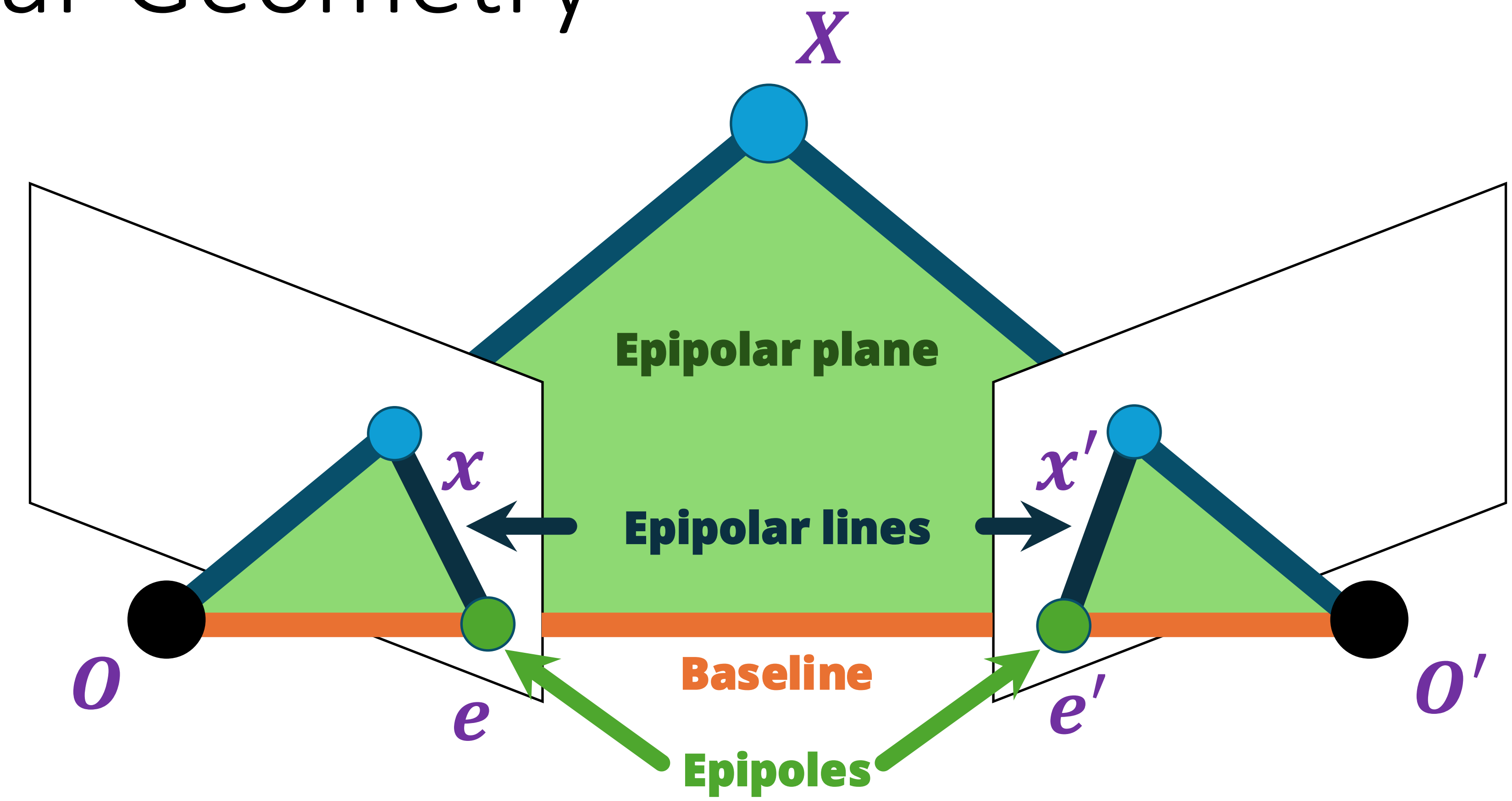
- Consider a **point** X, which projects to x and x'

Epipolar Geometry



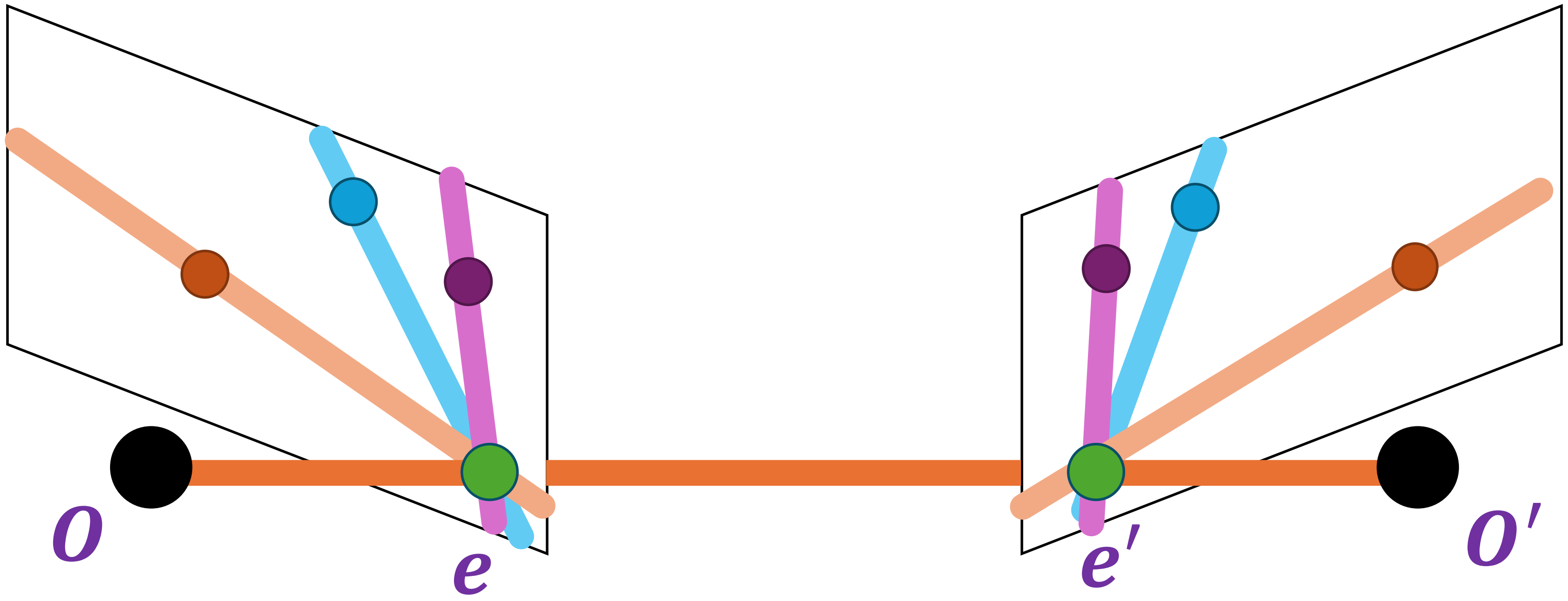
- The plane formed by X, O, and O' is called an **epipolar plane**
- There is a family of planes passing through O and O'

Epipolar Geometry



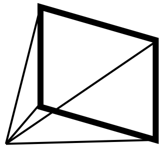
- **Epipolar lines** connect the epipoles to the projections of X
- Equivalently, they are intersections of the epipolar plane with the image planes – thus, they come in matching pairs

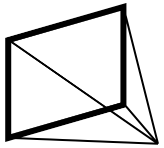
Epipolar Geometry

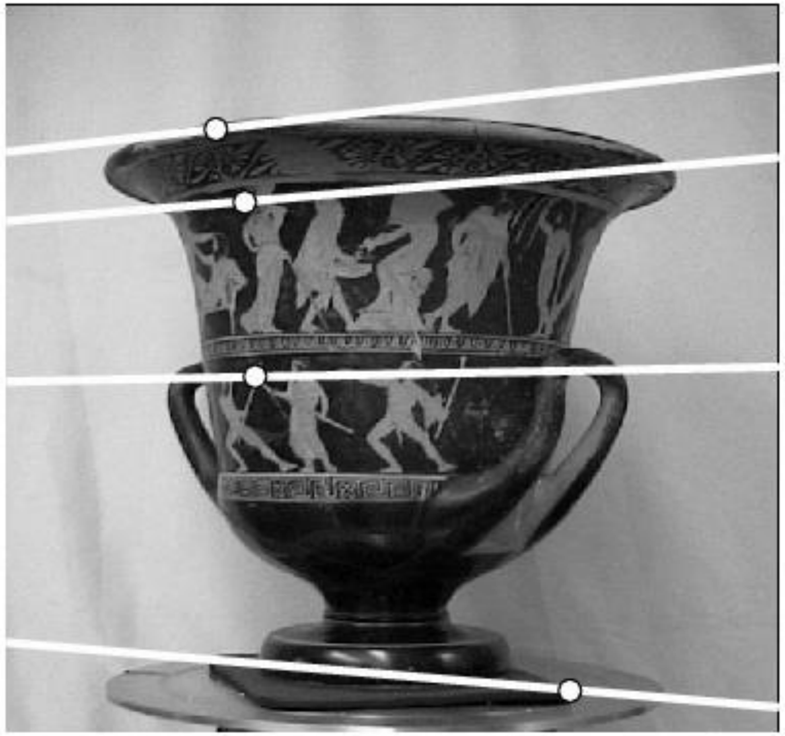


- All **epipolar lines** pass through the **epipoles**
- **Epipoles** can lie outside of the image

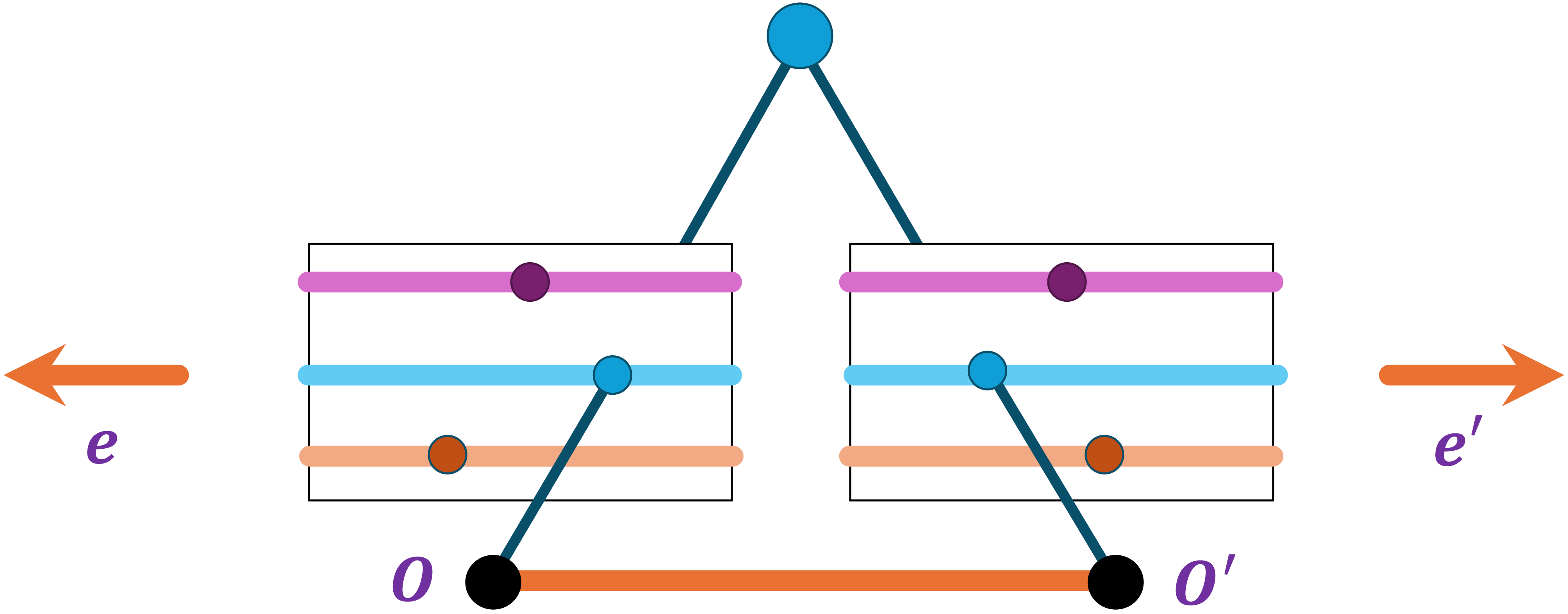
Example of Epipolar Lines





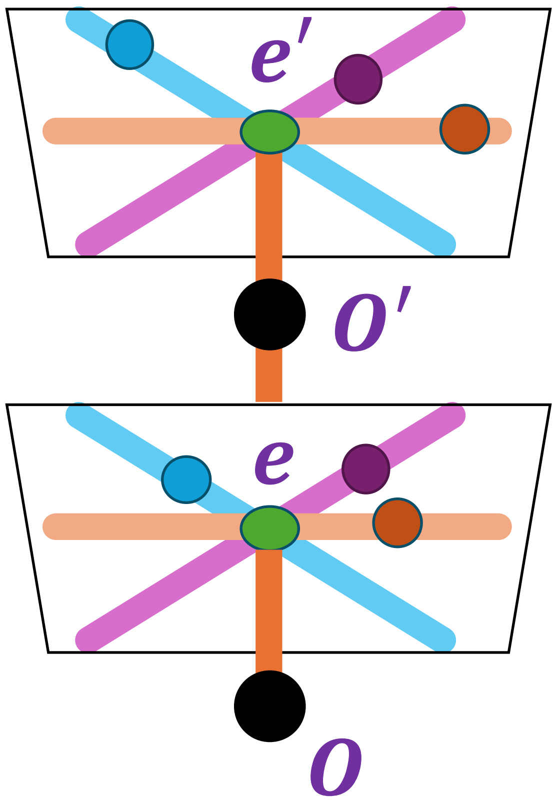


Parallel to Image Plane



- Where are the epipoles and what do the epipolar lines look like?
- Epipoles ***infinitely*** far away, epipolar lines parallel

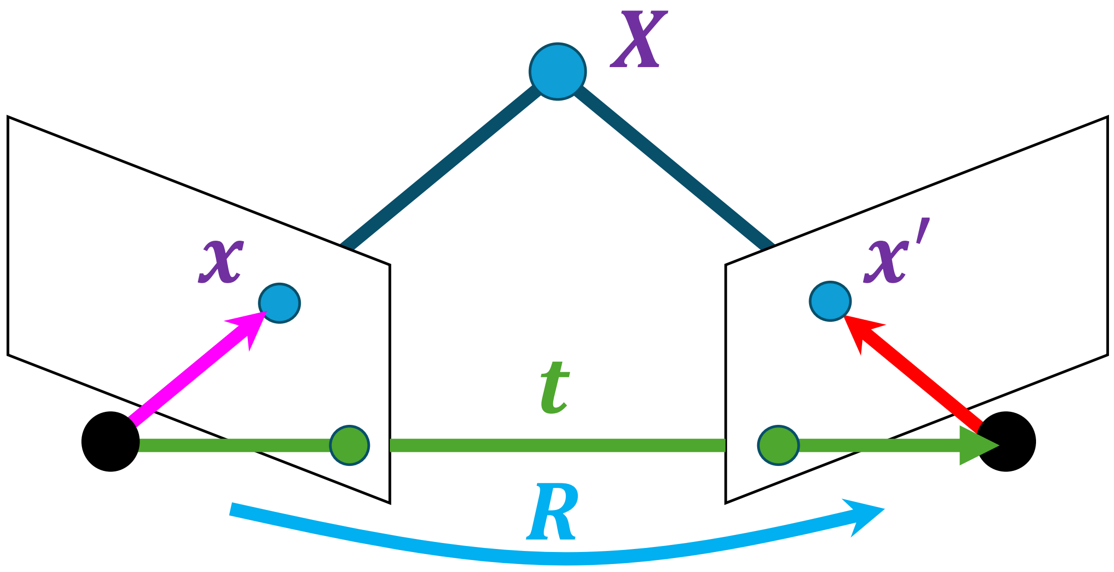
Perpendicular to Image Plane



- Where are the epipoles and what do the epipolar lines look like?
- Epipole is "focus of expansion" and coincides with the principal point of the camera
- Epipolar lines go out from principal point

Epipolar Constraint: Calibrated Case

$\boxed{x \cong K[R|t]X}$

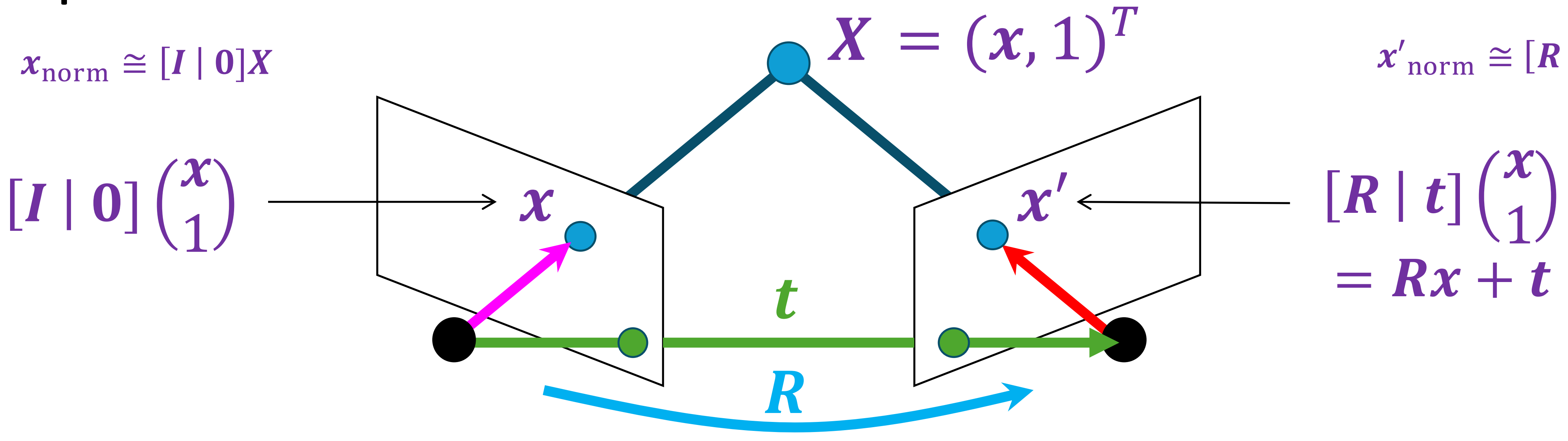


- Assume the intrinsic and extrinsic parameters of the cameras are known (*calibrated*), and world coordinate system is set to that of the first camera
- Then the projection matrices are given by $K[I \mid 0]$ and $K'[R \mid t]$
- We can pre-multiply the projection matrices (and the image points) by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

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

Epipolar Constraint: Calibrated Case

$\boldsymbol{x} \cong \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]\boldsymbol{X}$



$$\boldsymbol{x}' \cong \boldsymbol{R}\boldsymbol{x} + \boldsymbol{t} \quad \Rightarrow \quad \boldsymbol{x}' \cdot [\boldsymbol{t} \times (\boldsymbol{R}\boldsymbol{x})] = 0$$

- This means the three vectors $\boldsymbol{x}'$, $\boldsymbol{R}\boldsymbol{x}$, and $\boldsymbol{t}$ are linearly dependent, i.e., lying on the same plane
- This constraint can be written using the *triple product*

Epipolar Constraint: Calibrated Case

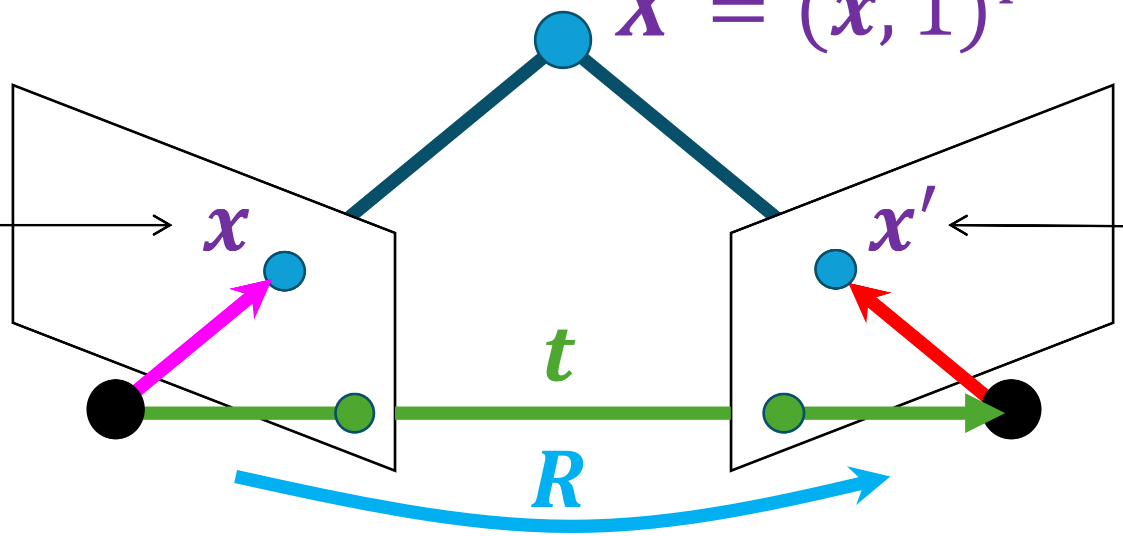
$\boldsymbol{x} \cong \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]\boldsymbol{X}$

$\boldsymbol{x}_{\text{norm}} \cong [\boldsymbol{I} \mid \boldsymbol{0}]\boldsymbol{X}$

$\boldsymbol{X} = (\boldsymbol{x}, 1)^T$

$\boldsymbol{x}'_{\text{norm}} \cong [\boldsymbol{R} \mid \boldsymbol{t}]\boldsymbol{X}$

$[\boldsymbol{I} \mid \boldsymbol{0}]\begin{pmatrix}\boldsymbol{x}\\1\end{pmatrix}$



$[\boldsymbol{R} \mid \boldsymbol{t}]\begin{pmatrix}\boldsymbol{x}\\1\end{pmatrix} = \boldsymbol{Rx} + \boldsymbol{t}$

$\boldsymbol{x}' \cong \boldsymbol{Rx} + \boldsymbol{t} \Rightarrow \boldsymbol{x}' \cdot [\boldsymbol{t} \times (\boldsymbol{Rx})] = 0 \Rightarrow \boldsymbol{x}'^T[\boldsymbol{t}_\times]\boldsymbol{Rx} = 0$

Recall: $\boldsymbol{a} \times \boldsymbol{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = [\boldsymbol{a}_\times]\boldsymbol{b}$ $\Rightarrow$ skew-symmetric matrix: $\boldsymbol{A}^T = -\boldsymbol{A}$

Epipolar Constraint: Calibrated Case

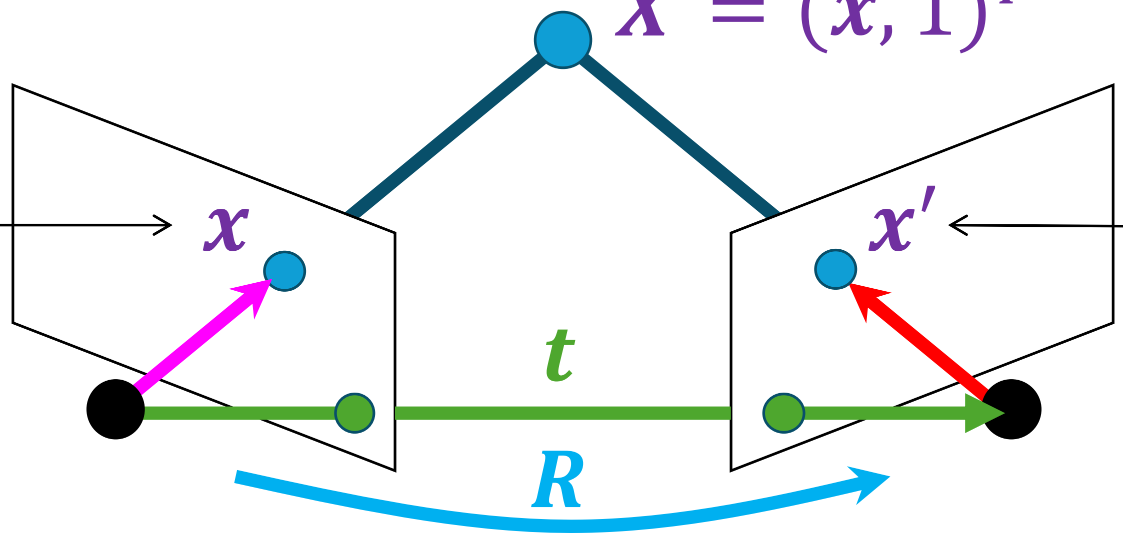
$\boldsymbol{x} \cong \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]\boldsymbol{X}$

$\boldsymbol{x}_{\text{norm}} \cong [\boldsymbol{I} \mid \boldsymbol{0}]\boldsymbol{X}$

$\boldsymbol{X} = (\boldsymbol{x}, 1)^T$

$\boldsymbol{x}'_{\text{norm}} \cong [\boldsymbol{R} \mid \boldsymbol{t}]\boldsymbol{X}$

$[\boldsymbol{I} \mid \boldsymbol{0}]\begin{pmatrix}\boldsymbol{x}\\1\end{pmatrix}$

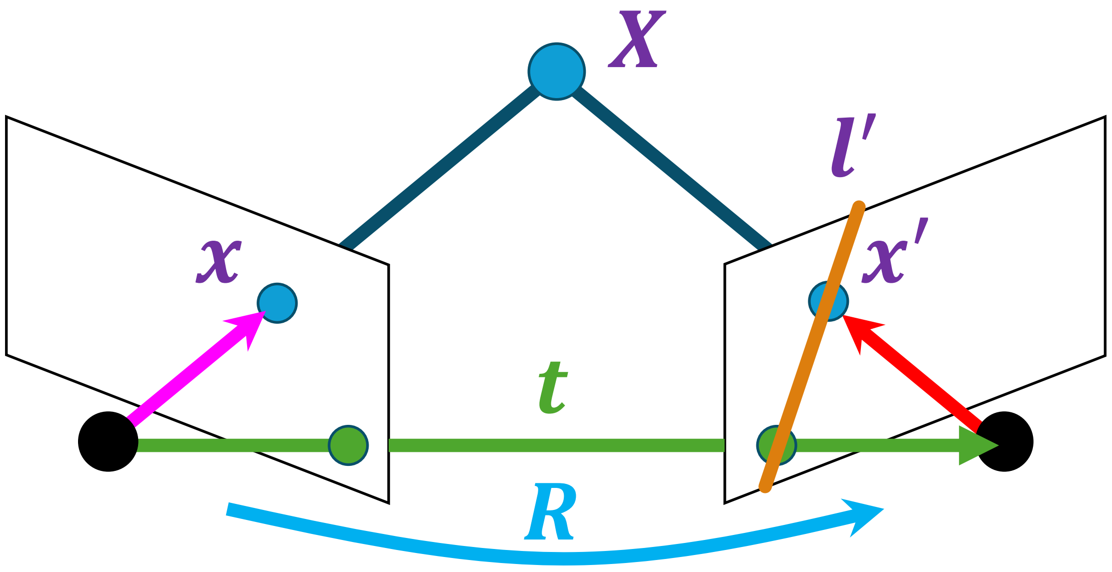


$$[\boldsymbol{R} \mid \boldsymbol{t}]\begin{pmatrix}\boldsymbol{x}\\1\end{pmatrix} = \boldsymbol{R}\boldsymbol{x} + \boldsymbol{t}$$

$$\boldsymbol{x}' \cong \boldsymbol{R}\boldsymbol{x} + \boldsymbol{t} \Rightarrow \boldsymbol{x}' \cdot [\boldsymbol{t} \times (\boldsymbol{R}\boldsymbol{x})] = 0 \Rightarrow \boldsymbol{x}'^T[\boldsymbol{t}_\times]\boldsymbol{R}\boldsymbol{x} = 0 \Rightarrow \boldsymbol{x}'^T\boldsymbol{E}\boldsymbol{x} = 0$$

Essential Matrix

The Essential Matrix

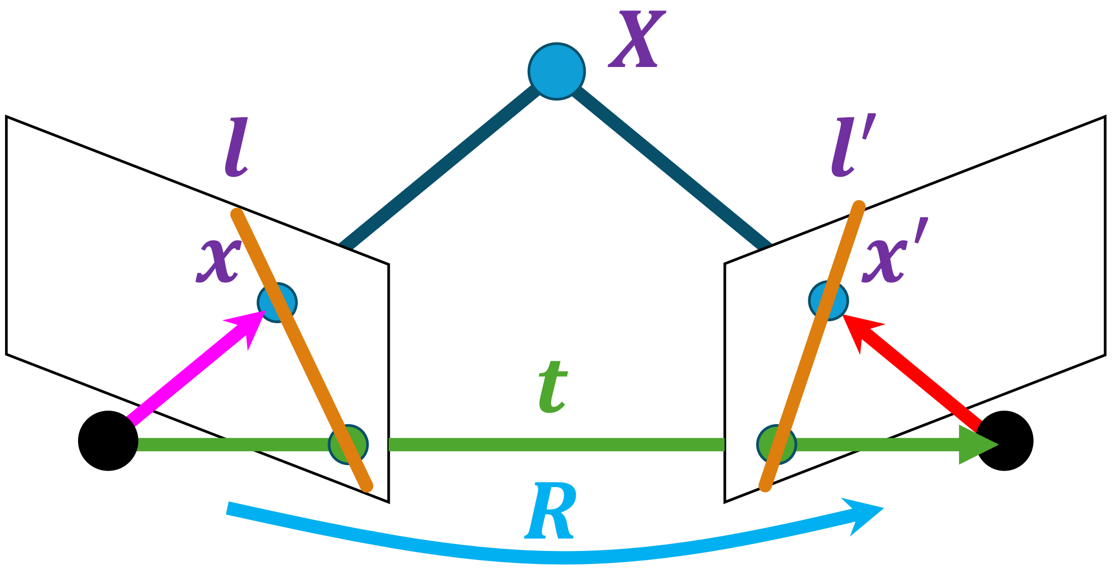


$$\boldsymbol{x}'^T \overbrace{\boxed{\boldsymbol{E}\boldsymbol{x}}}^{\boldsymbol{l}' = \boldsymbol{E}^T\boldsymbol{x}} = 0$$

$$(x', y', 1)\begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$

- $\boldsymbol{E}\boldsymbol{x}$ is the **epipolar line** associated with $\boldsymbol{x}$ ($\boldsymbol{l}' = \boldsymbol{E}\boldsymbol{x}$)
- Recall: a line is given by $ax + by + c = 0$ or $\boldsymbol{l}^T\boldsymbol{x} = 0$ in homogeneous coordinates, where $\boldsymbol{l} = (a, b, c)^T$ and $\boldsymbol{x} = (x, y, 1)^T$
- $\boldsymbol{x}'^T\boldsymbol{E}\boldsymbol{x} = \boldsymbol{x}'^T\boldsymbol{l}' = 0$ means $\boldsymbol{x}'$ lies on the epipolar line $\boldsymbol{l}'$

The Essential Matrix

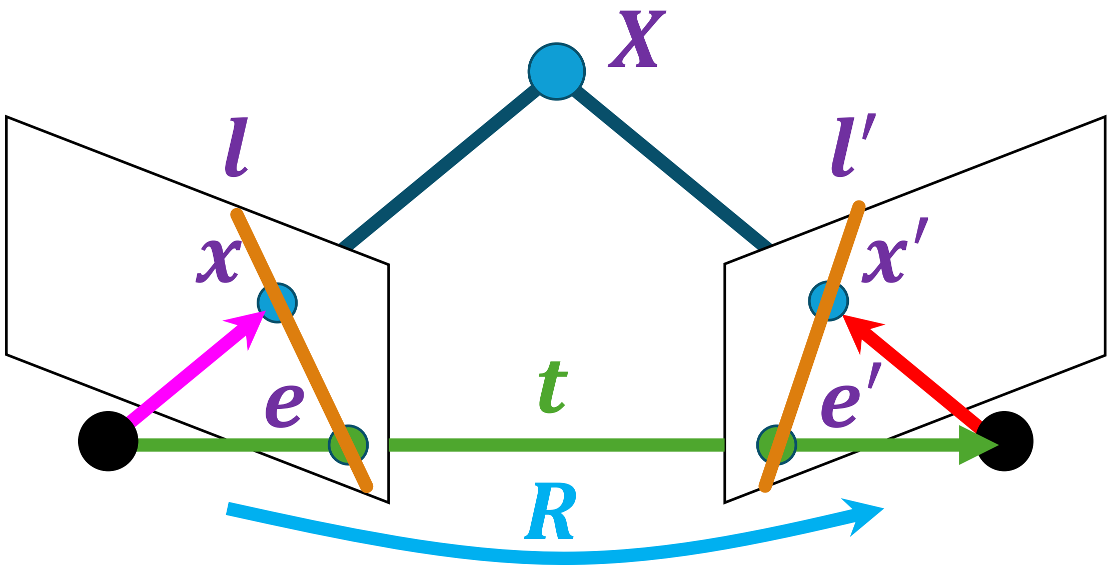


$$\underbrace{\boxed{\boldsymbol{x}'^T\boldsymbol{E}}}_{\boldsymbol{l} = \boldsymbol{E}^T\boldsymbol{x}'}\boldsymbol{x} = 0$$

$$(x', y', 1)\begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$

- $\boldsymbol{Ex}$ is the **epipolar line** associated with $\boldsymbol{x}$ ($\boldsymbol{l}' = \boldsymbol{Ex}$)
- $\boldsymbol{x}'^T\boldsymbol{Ex} = \boldsymbol{x}'^T\boldsymbol{l}' = 0$ means $\boldsymbol{x}'$ lies on the epipolar line $\boldsymbol{l}'$
- Equivalently, $\boldsymbol{E}^T\boldsymbol{x}'$ is the **epipolar line** associated with $\boldsymbol{x}'$ ($\boldsymbol{l} = \boldsymbol{E}^T\boldsymbol{x}'$)
- $\boldsymbol{x}'^T\boldsymbol{Ex} = \boldsymbol{lx} = 0$ means $\boldsymbol{x}$ lies on the epipolar line $\boldsymbol{l}$

The Essential Matrix



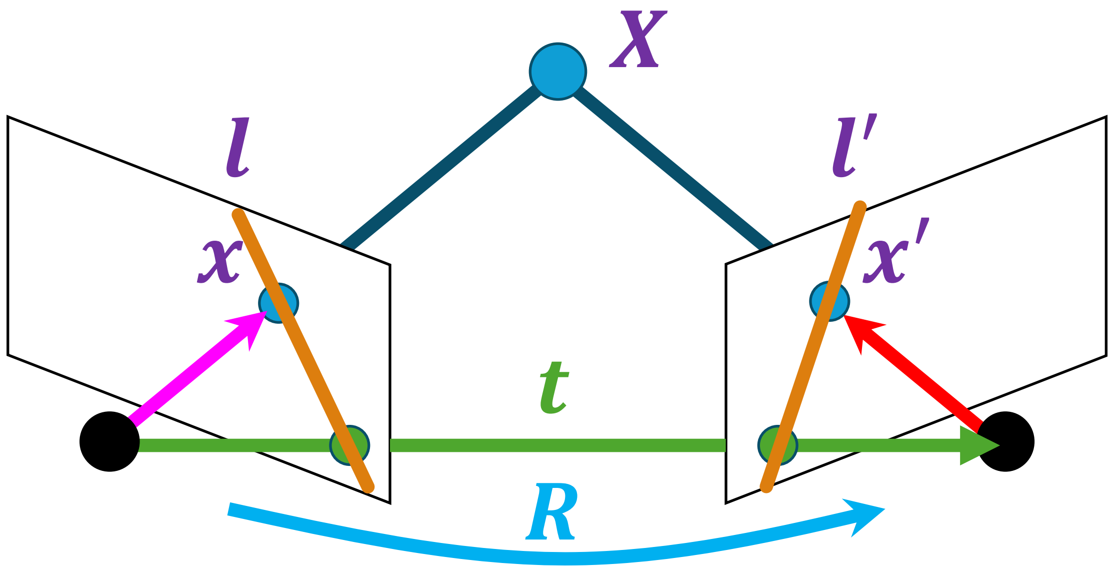
$$\boldsymbol{x}'^{T}\boldsymbol{E}\boldsymbol{x} = 0$$

$$(x', y', 1)\begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$

- $\boldsymbol{E}\boldsymbol{x}$ is the **epipolar line** associated with $\boldsymbol{x}$ ($\boldsymbol{l}' = \boldsymbol{E}\boldsymbol{x}$)
- $\boldsymbol{E}^T\boldsymbol{x}'$ is the **epipolar line** associated with $\boldsymbol{x}'$ ($\boldsymbol{l} = \boldsymbol{E}^T\boldsymbol{x}'$)
- $\boldsymbol{E}\boldsymbol{e} = \boldsymbol{0}$ and $\boldsymbol{E}^T\boldsymbol{e}' = \boldsymbol{0}$, where $\boldsymbol{e}$, $\boldsymbol{e}'$ are the epipoles
- $\boldsymbol{E}$ is singular (rank two) and has five degrees of freedom => why?
  - $\boldsymbol{E} = [\boldsymbol{t}_{\times}]\boldsymbol{R}$! $[\boldsymbol{t}_{\times}]$ is skew-symmetric; has rank 2
  - $\boldsymbol{R}$: 3 DoF, $\boldsymbol{t}$: 3 DoF, but we lost 1 DoF due to scale; move along $\boldsymbol{t}$ doesn't change $\boldsymbol{l}$

Epipolar Constraint: Uncalibrated

$\boldsymbol{x} \cong \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]\boldsymbol{X}$



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

$$(x', y', 1) \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$

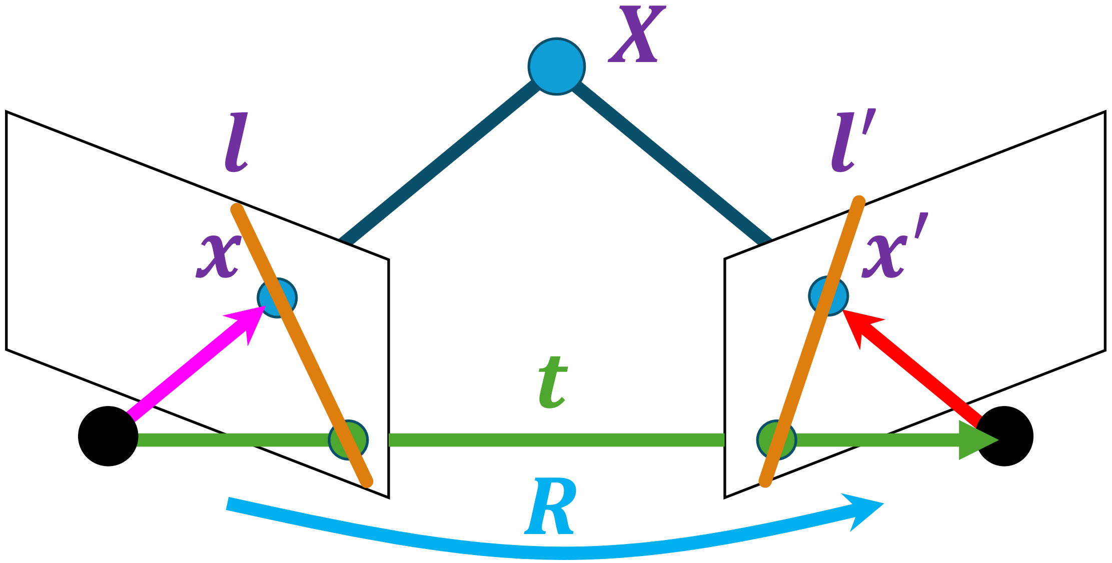
- What if camera intrinsics $\boldsymbol{K}$, $\boldsymbol{K}'$ are unknown?
- We can write the epipolar constraint:

$$\boldsymbol{x}'^T_{\text{norm}} \boldsymbol{E} \boldsymbol{x}_{\text{norm}} = 0$$

where $\boldsymbol{x}_{\text{norm}} = \boldsymbol{K}^{-1}\boldsymbol{x}$, $\boldsymbol{x}'_{\text{norm}} = \boldsymbol{K}'^{-1}\boldsymbol{x}'$

Epipolar Constraint: Uncalibrated

$\boldsymbol{x} \cong \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]\boldsymbol{X}$



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

$$(x', y', 1) \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$

- What if camera intrinsics $\boldsymbol{K}$, $\boldsymbol{K}'$ are unknown?
- We can write the epipolar constraint:

$$\boldsymbol{x}'^T_{\text{norm}} \boldsymbol{E} \boldsymbol{x}_{\text{norm}} = \boldsymbol{x}'^T \boxed{\boldsymbol{K}'^{-T} \boldsymbol{E} \boldsymbol{K}^{-1}} \boldsymbol{x} = 0$$

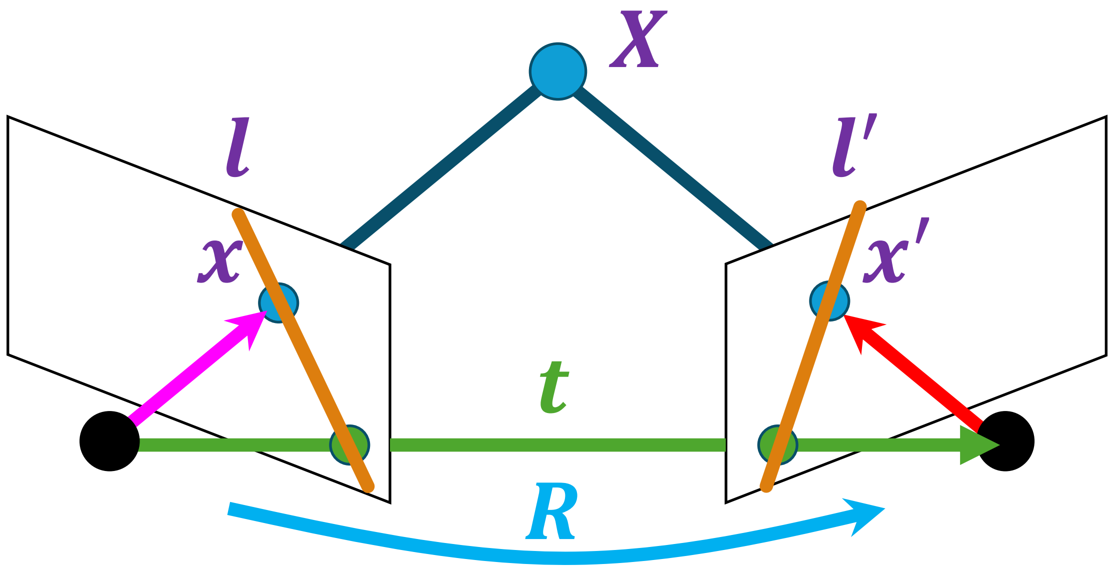
where $\boldsymbol{x}_{\text{norm}} = \boldsymbol{K}^{-1}\boldsymbol{x}$, $\boldsymbol{x}'_{\text{norm}} = \boldsymbol{K}'^{-1}\boldsymbol{x}'$

Fundamental Matrix

$$\boldsymbol{F} = \boldsymbol{K}'^{-T} \boldsymbol{E} \boldsymbol{K}^{-1}$$

The Fundamental Matrix

$\boldsymbol{x} \cong \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]\boldsymbol{X}$



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

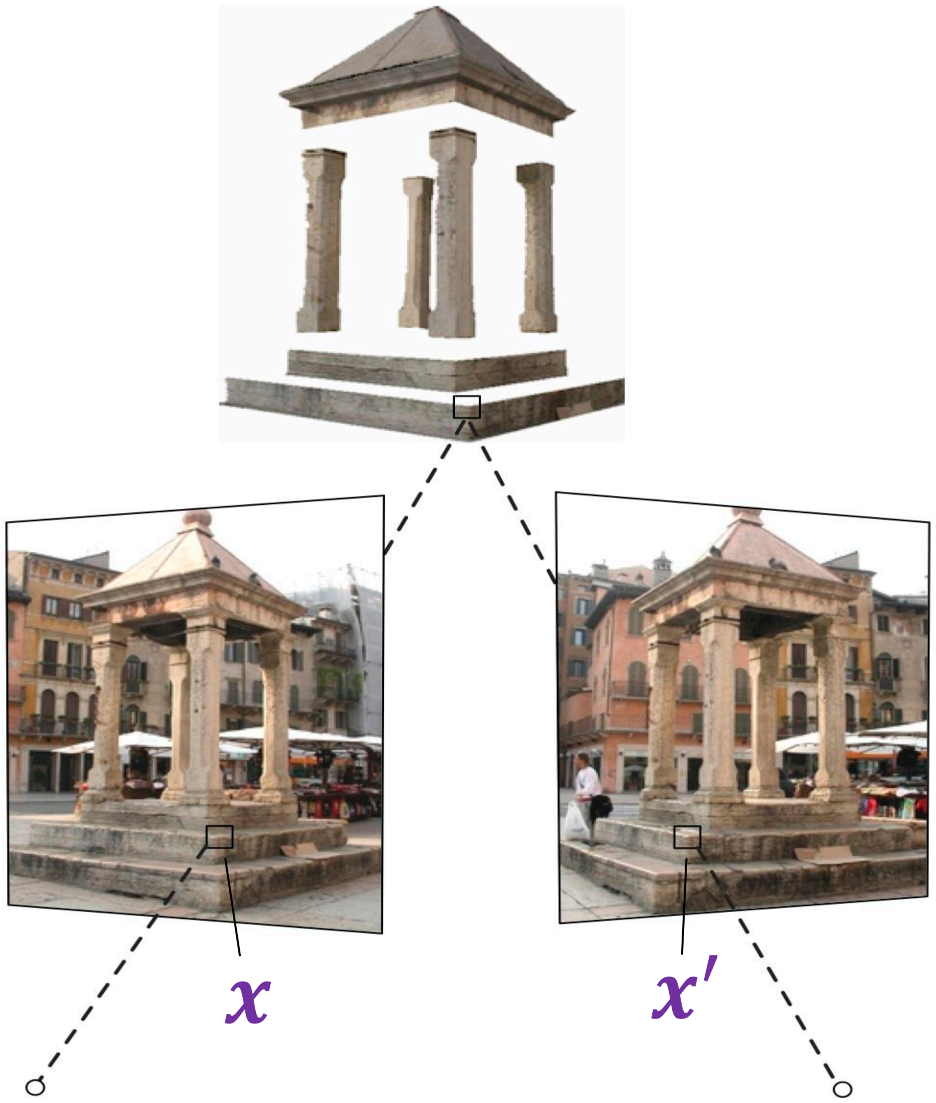
$$(x', y', 1)\begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$

- $\boldsymbol{F}\boldsymbol{x}$ is the **epipolar line** associated with $\boldsymbol{x}$ ($\boldsymbol{l}' = \boldsymbol{F}\boldsymbol{x}$)
- $\boldsymbol{F}^T\boldsymbol{x}'$ is the **epipolar line** associated with $\boldsymbol{x}'$ ($\boldsymbol{l} = \boldsymbol{F}^T\boldsymbol{x}'$)
- $\boldsymbol{F}\boldsymbol{e} = \boldsymbol{0}$ and $\boldsymbol{F}^T\boldsymbol{e}' = \boldsymbol{0}$, where $\boldsymbol{e}$, $\boldsymbol{e}'$ are the epipoles
- $\boldsymbol{F}$ is singular (rank two) and has seven degrees of freedom => why?
  - $\boldsymbol{F} = \boldsymbol{K}'^{-T}[\boldsymbol{t}_\times]\boldsymbol{R}\boldsymbol{K}^{-1}$! $[\boldsymbol{t}_\times]$ is skew-symmetric; has rank 2
  - 9 entries, but we lost 1 DoF to scale, and 1 DoF to rank constraint

How Can We Use the Epipolar Constraint?

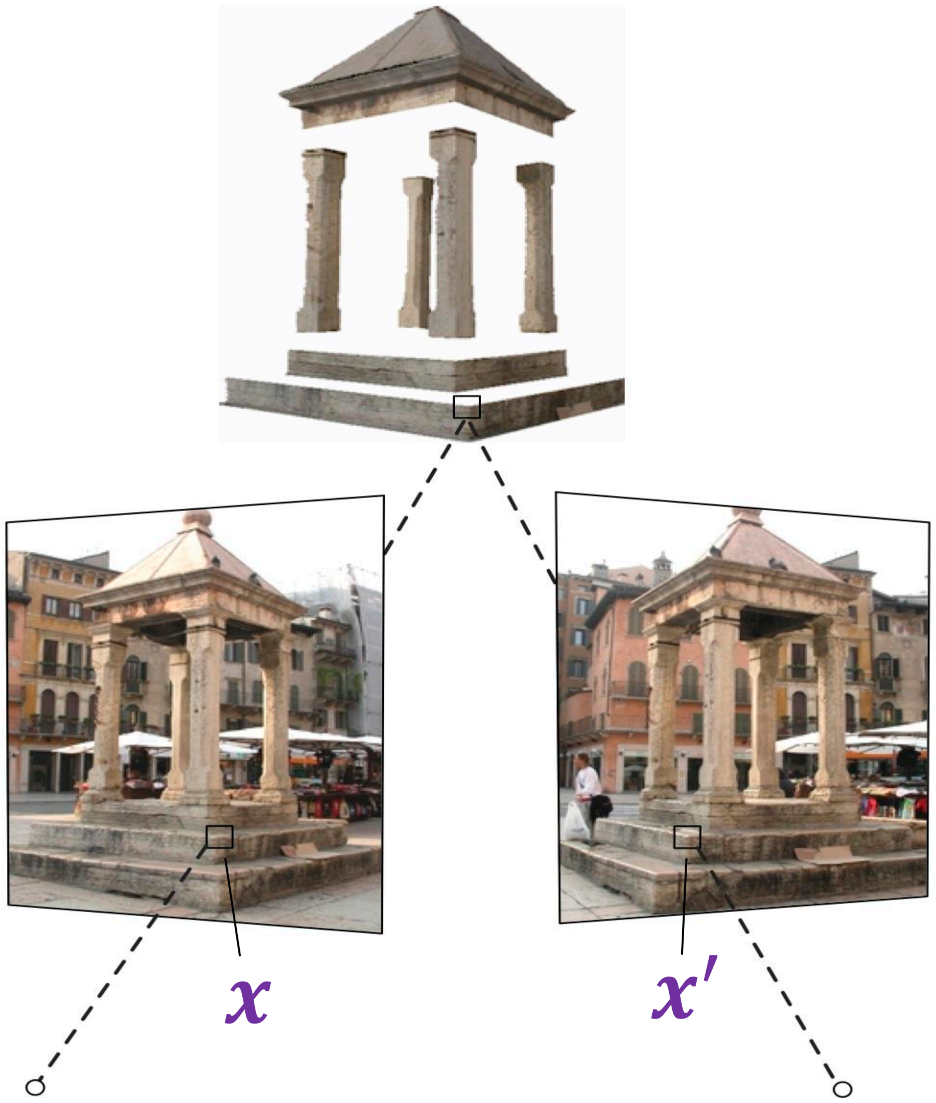
- **Given:** $\boldsymbol{F}$, $\boldsymbol{x}$, $\boldsymbol{x}'$
- **Q:** does there exist a 3D point that projects to $\boldsymbol{x}$ and $\boldsymbol{x}'$?
- **A:** Yes, if $\mathrm{residual}(\boldsymbol{x}'^T\boldsymbol{F}\boldsymbol{x})$ is sufficiently low

- Note: the interpretation of $\mathrm{residual}(\boldsymbol{x}'^T\boldsymbol{F}\boldsymbol{x})$ is the distance (geometric or algebraic) between $\boldsymbol{x}$ and $\boldsymbol{l} = \boldsymbol{F}^T\boldsymbol{x}'$, or $\boldsymbol{x}'$ and $\boldsymbol{l}' = \boldsymbol{F}\boldsymbol{x}$



How Can We Use the Epipolar Constraint?

- **Given:** $\boldsymbol{F}$
- **Q:** how do we find $\boldsymbol{R}$, $\boldsymbol{t}$?
- **A:**
  - **Step 0:** estimate $\boldsymbol{K}$, $\boldsymbol{K}'$ if not known (self-calibration)
  - **Step 1:** compute $\boldsymbol{E} = \boldsymbol{K}'^T \boldsymbol{F} \boldsymbol{K}$
  - **Step 2:** since $\boldsymbol{E} = [\boldsymbol{t}_\times]\boldsymbol{R}$, perform SVD on $\boldsymbol{E} = \boldsymbol{U\Sigma V}^T$. We have 4 solutions $(\boldsymbol{R}_1, \pm\boldsymbol{t})$, $(\boldsymbol{R}_2, \pm\boldsymbol{t})$, where $\boldsymbol{t} = \boldsymbol{U}[:,3]$, $\boldsymbol{R}_1 = \boldsymbol{UWV}^T$, $\boldsymbol{R}_2 = \boldsymbol{UW}^T\boldsymbol{V}^T$, and $\boldsymbol{W}$ is the matrix that rotates 90° about z-axis.
  - **Step 3:** pick the one that gives 3D points in front of both cameras (cheirality check).



How to Estimate the Fundamental Matrix?

- **Given:** correspondences $\boldsymbol{x_i} = (x_i, y_i, 1)^T$ and $\boldsymbol{x_i'} = (x_i', y_i', 1)^T$
- **Constraint:** $\boldsymbol{x}'^T \boldsymbol{F} \boldsymbol{x} = 0$

$$(x', y', 1) \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0 \Rightarrow (x'x, x'y, x', y'x, y'y, y', x, y, 1) \begin{pmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{pmatrix} = 0$$

How to Estimate the Fundamental Matrix?

- **Given:** correspondences $\boldsymbol{x_i} = (x_i, y_i, 1)^T$ and $\boldsymbol{x_i'} = (x_i{}', y_i{}', 1)^T$
- **Constraint:** $\boldsymbol{x}'^T \boldsymbol{F} \boldsymbol{x} = 0$

$$\overbrace{\begin{bmatrix} x_1'x_1 & x_1'y_1 & x_1' & y_1'x_1 & y_1'y_1 & y_1' & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n'x_n & x_n'y_n & x_n' & y_n'x_n & y_n'y_n & y_n' & x_n & y_n & 1 \end{bmatrix}}^{\boldsymbol{A}} \begin{pmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{pmatrix} = \boldsymbol{0}$$

Homogeneous least squares to find $\boldsymbol{f}$:

$$\arg\min_{\|\boldsymbol{f}\|=1} \|\boldsymbol{A}\boldsymbol{f}\|_2^2 \quad \Rightarrow \quad \text{Least eigenvector of } \boldsymbol{A}^T\boldsymbol{A}$$

Small Trick to Enforce Rank-2 Constraint

- We know $\boldsymbol{F}$ must be singular/rank 2. How do we force that?
- **Solution:** take SVD of the initial estimate and throw out the smallest singular value

$$\boldsymbol{F}_{\text{init}} = \boldsymbol{U\Sigma V}^T$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \cancel{\sigma_3} \end{bmatrix} \longrightarrow \boldsymbol{\Sigma}' = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{F} = \boldsymbol{U\Sigma' V}^T$$

The Fundamental Matrix Song

By Daniel Wedge: https://danielwedge.com/fmatrix/

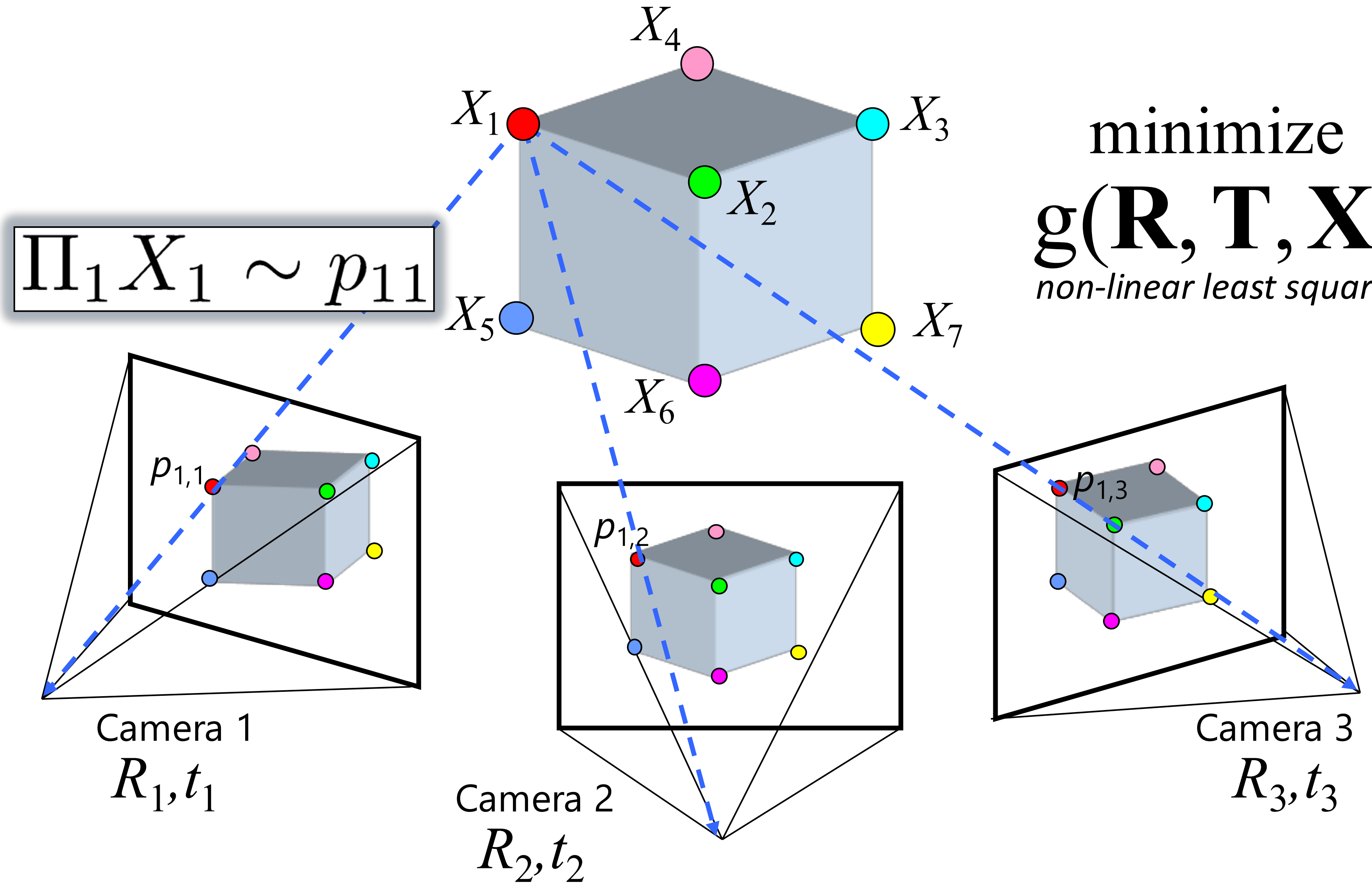
The Fundamental Matrix Song – Live!

Daniel Wedge and the CVPR house band at CVPR 2023 in Vancouver

More Than Two Views?

- The geometry of three views is described by a 3 x 3 x 3 tensor called the *trifocal tensor*
- The geometry of four views is described by a 3 x 3 x 3 x 3 tensor called the *quadrifocal tensor*
- After this it starts to get complicated…
- Or we can pose it as an optimization problem

Structure from Motion (SfM)



minimize

$$g(\mathbf{R}, \mathbf{T}, \mathbf{X})$$

non-linear least squares

Bundle Adjustment

- Minimize reprojection errors:

$$g(\mathbf{X},\mathbf{R},\mathbf{T}) = \sum_{i=1}^{m}\sum_{j=1}^{n} \underbrace{w_{ij}}_{\substack{\textit{indicator variable:}\\ \text{is point } i \text{ visible in image } j\text{ ?}}} \cdot \left\| \underbrace{\mathbf{P}(\mathbf{x}_i,\mathbf{R}_j,\mathbf{t}_j)}_{\substack{\textit{predicted}\\ \text{image location}}} - \underbrace{\begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix}}_{\substack{\textit{observed}\\ \text{image location}}} \right\|^2$$

- Optimized using non-linear least squares, e.g., Levenberg-Marquardt algorithm
- Susceptible to local minima; requires careful initialization

Incremental Structure from Motion

- Photo Tourism (Snavely et al., SIGGAPH'06)
- Trevis Fountain, Rome
  - 466 Internet photos
  - > 100,000 3D points
  - Very large optimization problem

Practical SfM Tools

- COLMAP (by Schönberger et al.): https://colmap.github.io/
- nerfstudio COLMAP Python wrapper: https://docs.nerf.studio/quickstart/custom_dataset.html

- Still, with thousands of images, it can take many hours or even days!
- There's a family of methods optimized for efficiency and continuous streams, often referred to as *Simultaneous Localization and Mapping* (**SLAM**), particularly useful for robot navigation for instance
  - One classic, widely-used system is *ORB-SLAM* (Mur-Artal et al., 2015)

- We will discuss learning-based approaches later

Part 1&2 Summary – Multi-view Geometry

- Camera model and projection
  - Homogeneous coordinates
  - Pinhole camera, perspective projection $\boldsymbol{x} \cong \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]\boldsymbol{X}$
  - Intrinsics $\boldsymbol{K}$, extrinsics $[\boldsymbol{R}|\boldsymbol{t}]$
  - Camera calibration
- Epipolar geometry
  - Epipolar plane, epipolar lines
  - Essential matrix $\boldsymbol{E}$ => $\boldsymbol{x}'^{T}_{\mathrm{norm}}\boldsymbol{E}\boldsymbol{x}_{\mathrm{norm}} = 0$ (calibrated)
  - Fundamental matrix $\boldsymbol{F}$ => $\boldsymbol{x}'^{T}\boldsymbol{E}\boldsymbol{x} = 0$ (uncalibrated)
- Structure from Motion (SfM)
  - Bundle adjustment