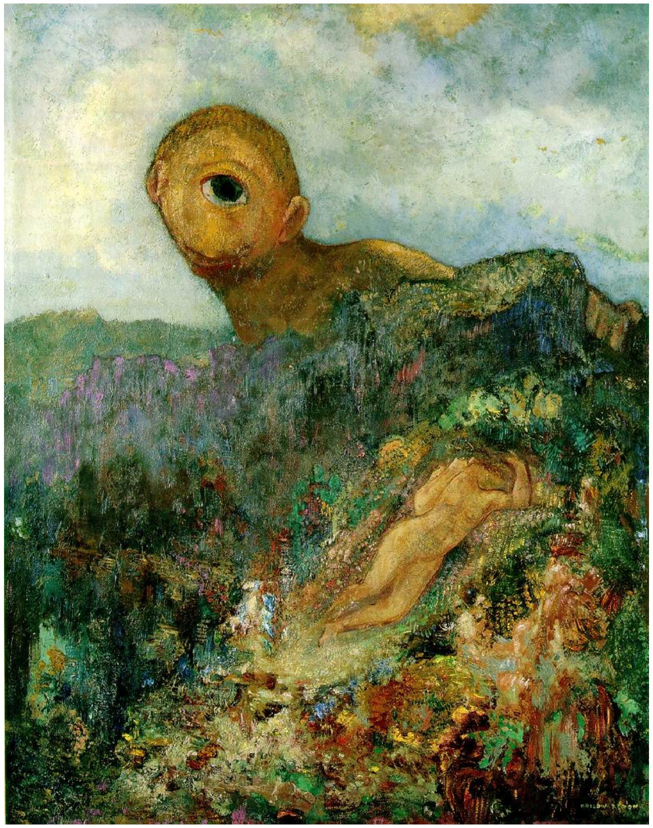
Camera calibration

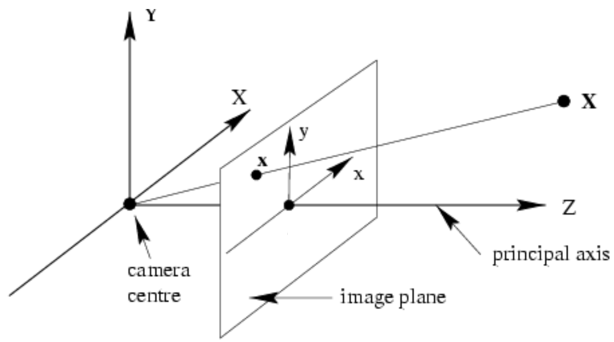


Odilon Redon, *Cyclops*, 1914

Overview

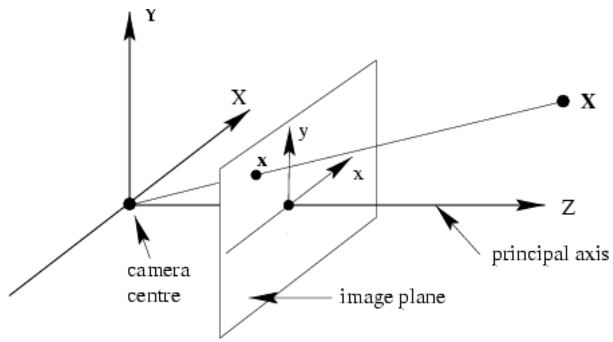
- Camera calibration
  - Intrinsic camera parameters
  - Extrinsic camera parameters
  - Estimation
- First taste of 3D reconstruction: triangulation

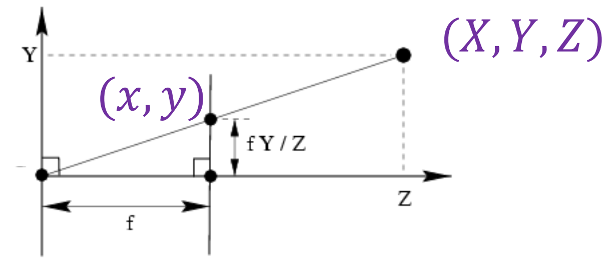
Perspective projection in normalized coordinates



- **Normalized (camera) coordinate system:** camera center is at the origin, the *principal axis* is the z-axis, x and y axes of the image plane are parallel to x and y axes of the world

Perspective projection in normalized coordinates





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

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

$$\boldsymbol{x} \cong \boldsymbol{P}\boldsymbol{X}$$

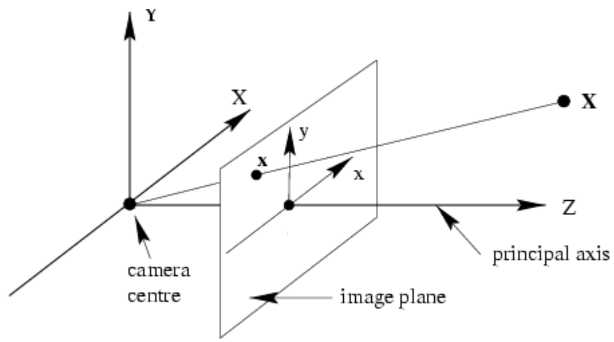
Equality up to scale

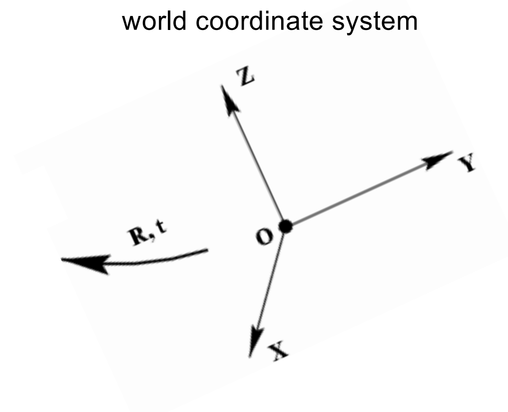
$\boldsymbol{x}$: Homogeneous coord. vec. of image point

$\boldsymbol{P}$: Camera projection matrix

$\boldsymbol{X}$: Homogeneous coord. vec. of 3D point

Camera calibration: Overview





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

[2D point (3x1)] ≅ (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 (4x1)]

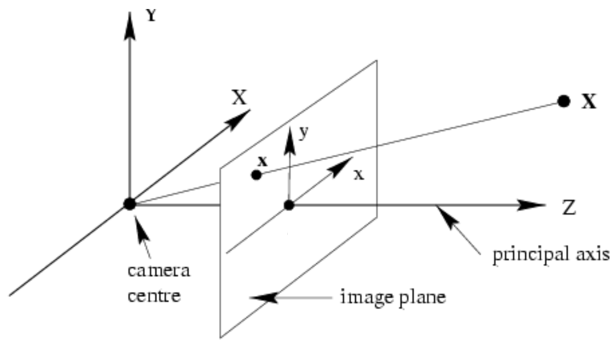
$\boldsymbol{x}$

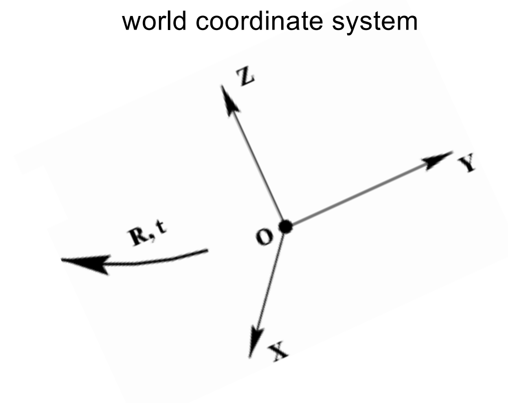
***Intrinsic* camera parameters**: principal point, scaling factors

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

$\boldsymbol{X}$

Camera calibration: Overview





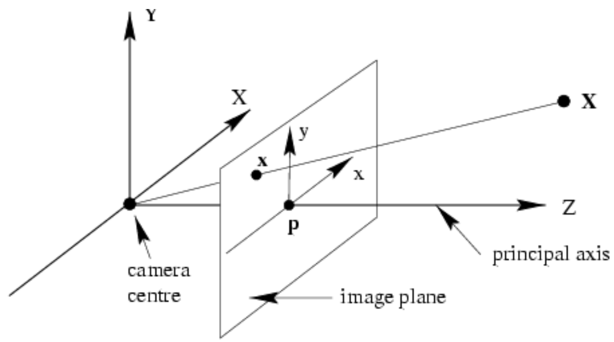
- **Camera calibration:** figuring out transformation from *world* coordinate system to *image* coordinate system

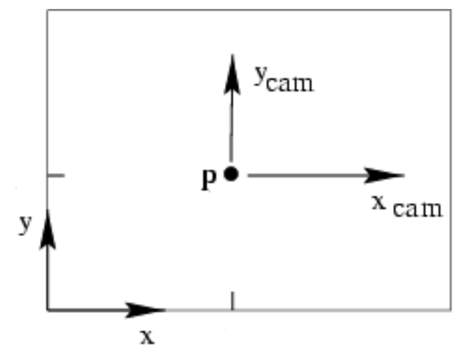
2D point (3x1)		Camera to pixel coord. trans. matrix	*Canonical* projection matrix	World to camera coord. trans. matrix	3D point (4x1)
	$\cong$	K (3x3)	$[I \mid 0]$ (3x4)	$\begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$ (4x4)	
x			$P = K[R \mid t]$		X

$$P = K[R|t]$$

General camera projection matrix

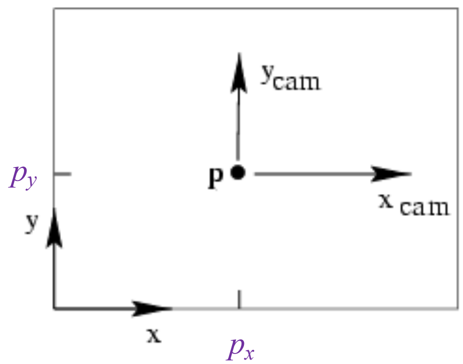
Intrinsic parameters: Principal point





- **Principal point (p):** point where principal axis intersects the image plane
- In the *normalized* coordinate system, the origin of the image is at the principal point
- In the *image* coordinate system: the origin is in the corner

Intrinsic parameters: Principal point

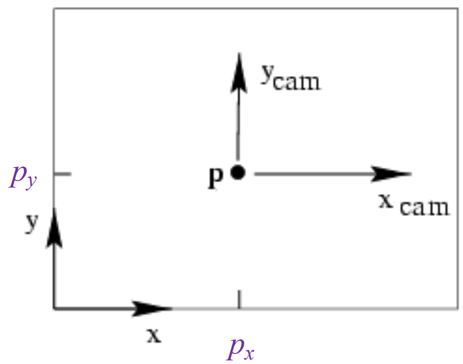


We want the principal point to map to (p_x, p_y) instead of $(0,0)$

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

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

Intrinsic parameters: Principal point



Principal point: (p_x, p_y)

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

calibration matrix $\boldsymbol{K}$

Canonical projection matrix $[\boldsymbol{I} \mid \mathbf{0}]$

$$\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{I}|\mathbf{0}]$$

Intrinsic parameters: Principal point

- What are the units of the focal length f and principal point coordinates (p_x, p_y)?
  - Same as world units – presumably metric units
- What units do we want for measuring image coordinates?
  - Pixel units
- Thus, we need to introduce *scaling factors* for mapping from world to pixel units

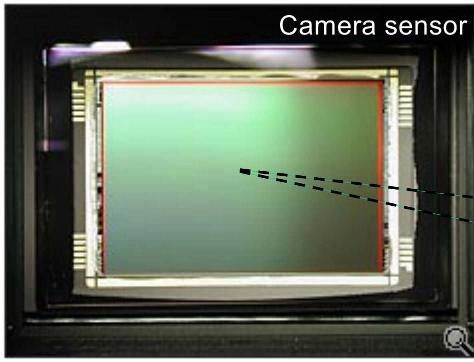
$$\begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

calibration matrix $\boldsymbol{K}$

Canonical projection matrix $[\boldsymbol{I} \mid \boldsymbol{0}]$

$\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{I}|\boldsymbol{0}]$

Intrinsic parameters: Scaling factors



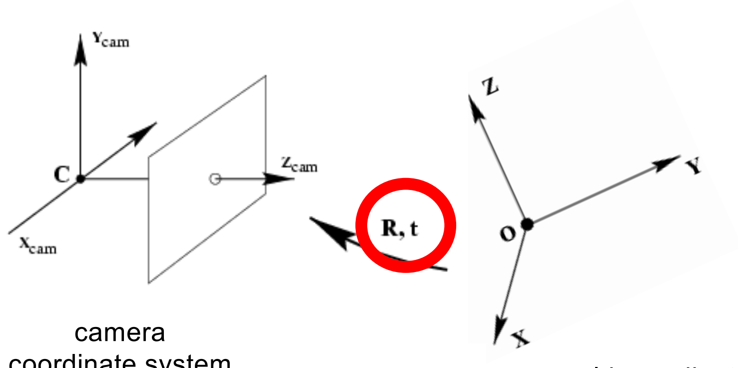
m_x pixels/m in horizontal direction,
m_y pixels/m in vertical direction

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

Scaling factors (pixels/m) · Calibration matrix K in metric units (m) = Calibration matrix K in pixel units (pixels)

$$\begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & 0 & \beta_x \\ 0 & \alpha_y & \beta_y \\ 0 & 0 & 1 \end{bmatrix}$$

Extrinsic parameters: Rotation and translation

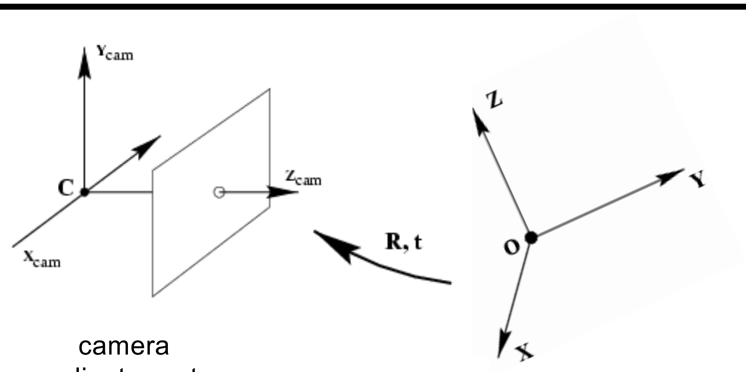


camera
coordinate system

world coordinate
system

- In general, the camera coordinate frame will be related to the world coordinate frame by a rotation and a translation

Extrinsic parameters: Rotation and translation



camera coordinate system

world coordinate system

- In general, the camera coordinate frame will be related to the world coordinate frame by a rotation and a translation

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

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

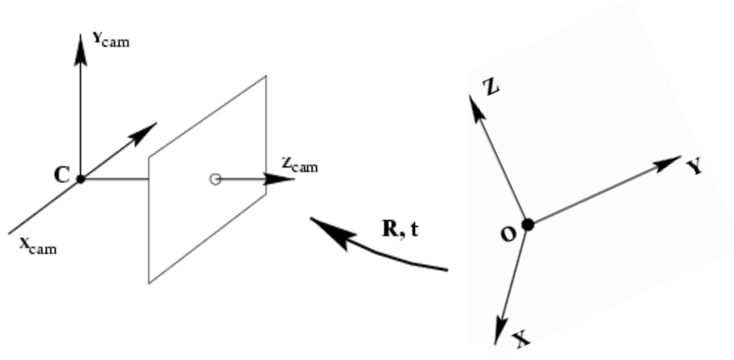
coords. of point in normalized camera frame

3x3 rotation matrix

coords. of a point in world frame

coords. of camera center in world frame

Extrinsic parameters: Rotation and translation



In *non-homogeneous* coordinates:

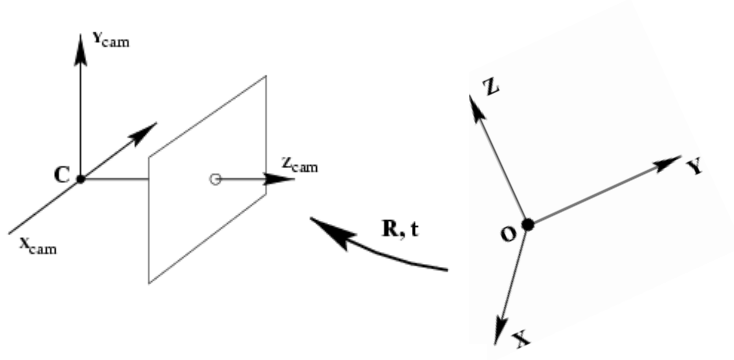
$$\widetilde{\boldsymbol{X}}_{\text{cam}} = \boldsymbol{R}\widetilde{\boldsymbol{X}} + \boldsymbol{t}$$

In *homogeneous* coordinates:

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

3D transformation matrix (4 x 4)

Extrinsic parameters: Rotation and translation



In *non-homogeneous* coordinates:

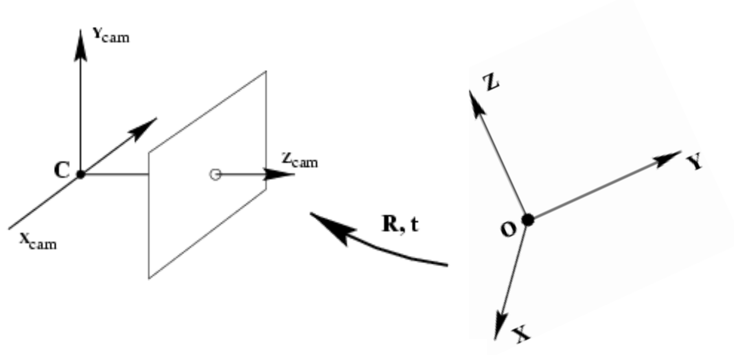
$$\widetilde{\boldsymbol{X}}_{\text{cam}} = \boldsymbol{R}\widetilde{\boldsymbol{X}} + \boldsymbol{t}$$

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)

Extrinsic parameters: Rotation and translation



In *non-homogeneous* coordinates:

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

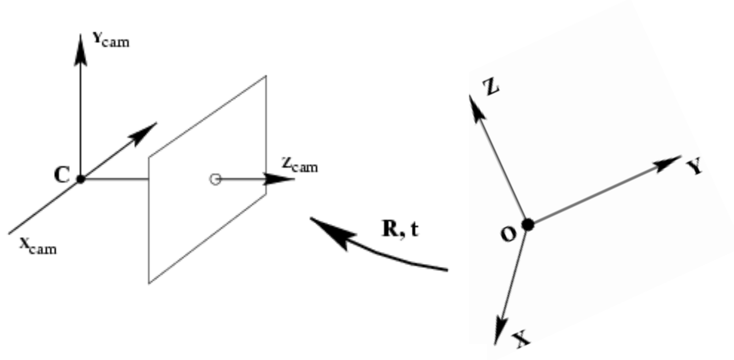
In *homogeneous* coordinates:

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

Transformation from normalized 3D coordinates to pixel image coordinates:

$$\boldsymbol{x} \cong \boldsymbol{K}[\boldsymbol{I}|\boldsymbol{0}]\boldsymbol{X}_{\text{cam}}$$

Extrinsic parameters: Rotation and translation



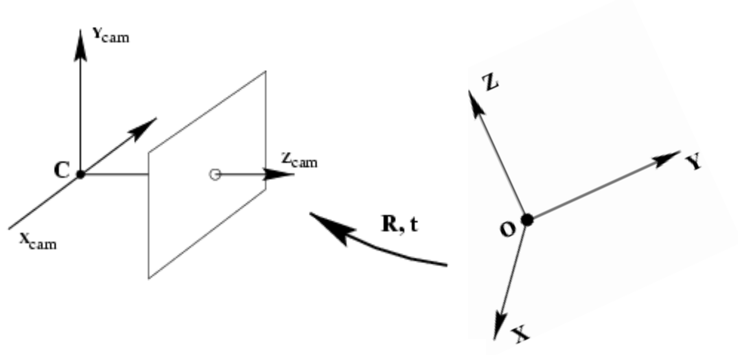
In *homogeneous* coordinates:

$$x \cong K[I|0]\begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} X$$

Finally:

$$x \cong K[R|t]X \qquad t = -R\tilde{C}$$

Extrinsic parameters: Rotation and translation



$$x \cong K[R|t]X \qquad t = -R\widetilde{C}$$

coords. of camera center in world frame

- What is the projection of the camera center in world coordinates?

$$PC = K[R \mid -R\widetilde{C}]\begin{pmatrix}\widetilde{C}\\1\end{pmatrix} = K(R\widetilde{C} - R\widetilde{C}) = \mathbf{0}$$

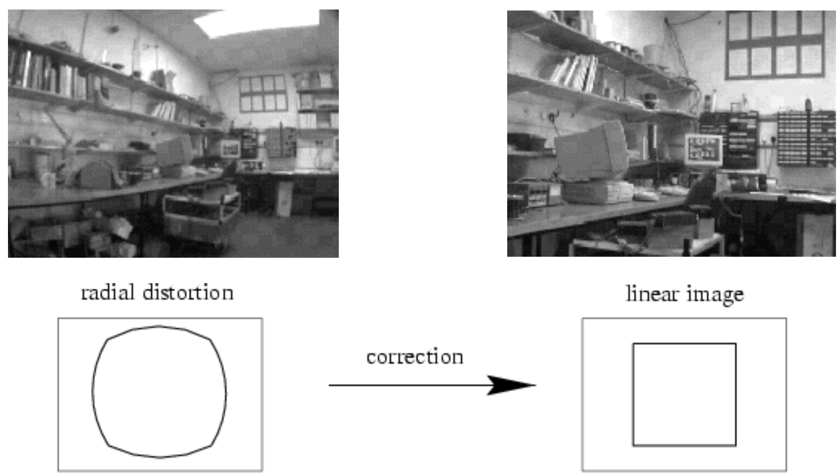
- The camera center is the null space of the projection matrix!

Camera parameters: Summary

$$P = K[R|t]$$

- Intrinsic parameters
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - *Skew (non-rectangular pixels) – not important in practice*
  - *Radial distortion – important in practice!*

$$K = \begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & 0 & \beta_x \\ 0 & \alpha_y & \beta_y \\ 0 & 0 & 1 \end{bmatrix}$$



Overview

- Camera calibration
  - Intrinsic camera parameters
  - Extrinsic camera parameters
  - Estimation

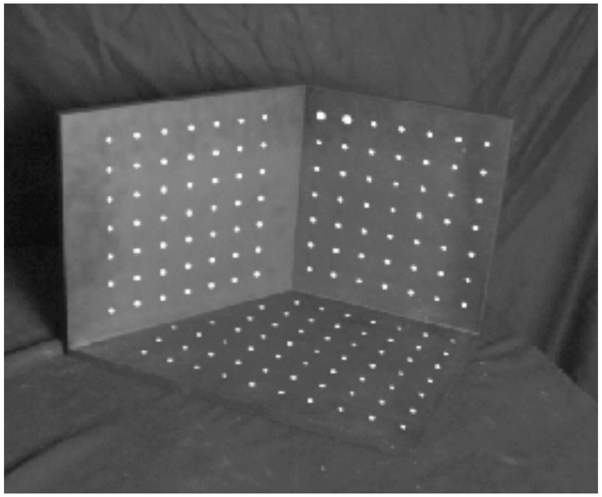
Camera calibration

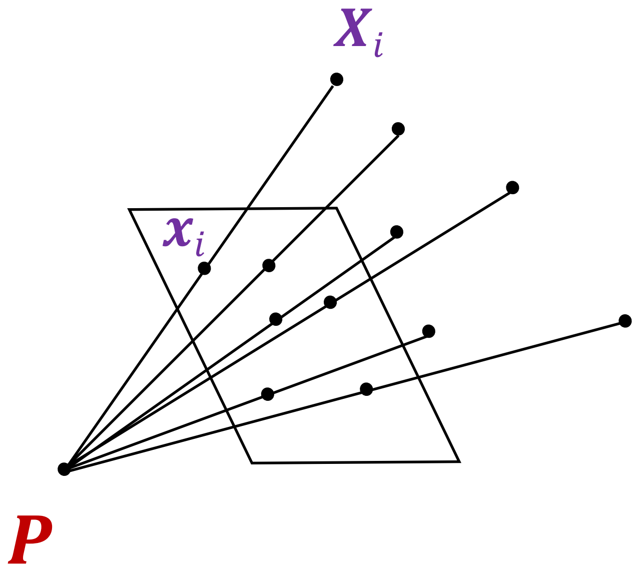
$$\boldsymbol{x} \cong \boldsymbol{K}[\boldsymbol{R} \;\; \boldsymbol{t}]\boldsymbol{X}$$

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

Camera calibration

- Given n points with known 3D coordinates $\boldsymbol{X}_i$ and known image projections $\boldsymbol{x}_i$, estimate the camera parameters





Camera calibration

- Given n points with known 3D coordinates $\boldsymbol{X}_i$ and known image projections $\boldsymbol{x}_i$, estimate the camera parameters

Known 2D image coords		Known 3D locations		
880	214	312.747	309.140	30.086
43	203	305.796	311.649	30.356
270	197	307.694	312.358	30.418
886	347	310.149	307.186	29.298
745	302	311.937	310.105	29.216
943	128	311.202	307.572	30.682
476	590	307.106	306.876	28.660
419	214	309.317	312.490	30.230
317	335	307.435	310.151	29.318
783	521	308.253	306.300	28.881
235	427	306.650	309.301	28.905
665	429	308.069	306.831	29.189
655	362	309.671	308.834	29.029
427	333	308.255	309.955	29.267
412	415	307.546	308.613	28.963
746	351	311.036	309.206	28.913
434	415	307.518	308.175	29.069
525	234	309.950	311.262	29.990
716	308	312.160	310.772	29.080
602	187	311.988	312.709	30.514

Image credit: J. Hays

Camera calibration: Linear method

- Recall homography fitting:

$$\boldsymbol{x}_i \cong \boldsymbol{P}\boldsymbol{X}_i \qquad \boldsymbol{x}_i \times \boldsymbol{P}\boldsymbol{X}_i = \boldsymbol{0} \qquad \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \times \begin{pmatrix} \boldsymbol{P}_1^T \boldsymbol{X}_i \\ \boldsymbol{P}_2^T \boldsymbol{X}_i \\ \boldsymbol{P}_3^T \boldsymbol{X}_i \end{pmatrix} = \boldsymbol{0}$$

$$\begin{bmatrix} \boldsymbol{0}^T & -\boldsymbol{X}_i^T & y_i \boldsymbol{X}_i^T \\ \boldsymbol{X}_i^T & \boldsymbol{0}^T & -x_i \boldsymbol{X}_i^T \\ -y_i \boldsymbol{X}_i^T & x_i \boldsymbol{X}_i^T & \boldsymbol{0}^T \end{bmatrix} \begin{pmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \\ \boldsymbol{P}_3 \end{pmatrix} = 0$$

One match gives two linearly independent constraints

Camera calibration: Linear method

- Final linear system:

$$\begin{bmatrix} \mathbf{0}^T & \boldsymbol{X}_1^T & -y_1\boldsymbol{X}_1^T \\ \boldsymbol{X}_1^T & \mathbf{0}^T & -x_1\boldsymbol{X}_1^T \\ \cdots & \cdots & \cdots \\ \mathbf{0}^T & \boldsymbol{X}_n^T & -y_n\boldsymbol{X}_n^T \\ \boldsymbol{X}_n^T & \mathbf{0}^T & -x_n\boldsymbol{X}_n^T \end{bmatrix} \begin{pmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \\ \boldsymbol{P}_3 \end{pmatrix} = 0 \qquad \boldsymbol{Ap} = 0$$

- One 2D/3D correspondence gives two linearly independent equations
  - The projection matrix has 11 degrees of freedom
  - 6 correspondences needed for a minimal solution
- Homogeneous least squares: find $\boldsymbol{p}$ minimizing $\|\boldsymbol{Ap}\|^2$
  - Solution is eigenvector of $\boldsymbol{A}^T\boldsymbol{A}$ corresponding to smallest eigenvalue

Camera calibration: Linear method

- Final linear system:

$$\begin{bmatrix} \mathbf{0}^T & \boldsymbol{X}_1^T & -y_1\boldsymbol{X}_1^T \\ \boldsymbol{X}_1^T & \mathbf{0}^T & -x_1\boldsymbol{X}_1^T \\ \dots & \dots & \dots \\ \mathbf{0}^T & \boldsymbol{X}_n^T & -y_n\boldsymbol{X}_n^T \\ \boldsymbol{X}_n^T & \mathbf{0}^T & -x_n\boldsymbol{X}_n^T \end{bmatrix} \begin{pmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \\ \boldsymbol{P}_3 \end{pmatrix} = 0 \qquad \boldsymbol{A}\boldsymbol{p} = 0$$

- What if all the n 3D points are *coplanar*, i.e., there exists a set of line parameters $\mathbf{\Pi}^T = (a, b, c, d)^T$ such that $\mathbf{\Pi}^T\boldsymbol{X}_i = 0$ for all i?
  - Then we will get *degenerate solutions* $(\mathbf{\Pi}, \mathbf{0}, \mathbf{0})$, $(\mathbf{0}, \mathbf{\Pi}, \mathbf{0})$, or $(\mathbf{0}, \mathbf{0}, \mathbf{\Pi})$

Camera calibration: Linear vs. nonlinear

- Linear calibration is easy to formulate and solve, but it doesn't directly tell us the camera parameters

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad \text{vs.} \quad \boldsymbol{x} \cong \boldsymbol{K}[\boldsymbol{R} \mid \boldsymbol{t}]\boldsymbol{X}$$

Camera calibration: Linear vs. nonlinear

- Linear calibration is easy to formulate and solve, but it doesn't directly tell us the camera parameters
- In practice, *non-linear* methods are preferred
  - Write down objective function in terms of intrinsic and extrinsic parameters, as sum of squared distances between measured 2D points and estimated projections of 3D points:

$$\sum_i \|\mathrm{proj}(\boldsymbol{K}[\boldsymbol{R} \mid \boldsymbol{t}]\boldsymbol{X}_i) - \boldsymbol{x}_i\|_2^2$$

  - Can include radial distortion and constraints such as known focal length, orthogonality, visibility of points
  - Minimize error using non-linear optimization package
  - Can initialize solution with output of linear method (perform QR decomposition to get $\boldsymbol{K}$ and $\boldsymbol{R}$ from $\boldsymbol{P}$)

Overview

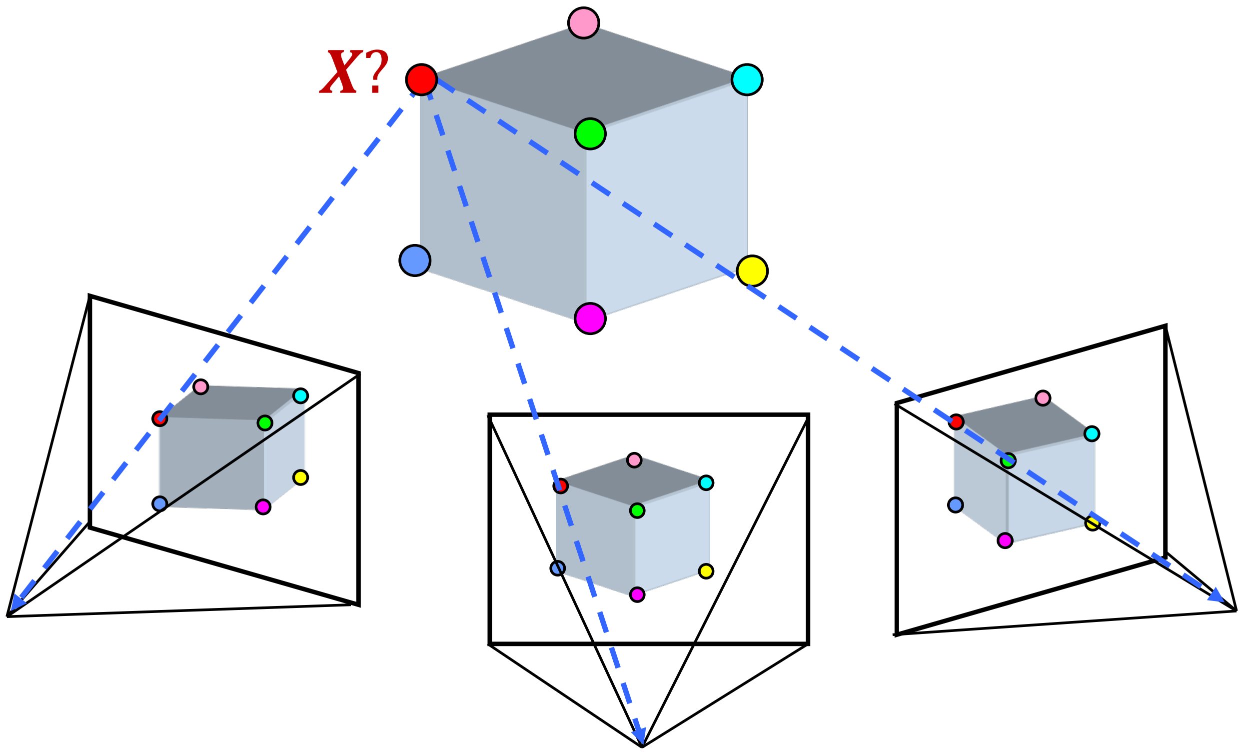
- Camera calibration
  - Intrinsic camera parameters
  - Extrinsic camera parameters
  - Estimation
- First taste of 3D reconstruction: triangulation

Triangulation

- Given projections of a 3D point in two or more images (with known camera matrices), find the coordinates of the point

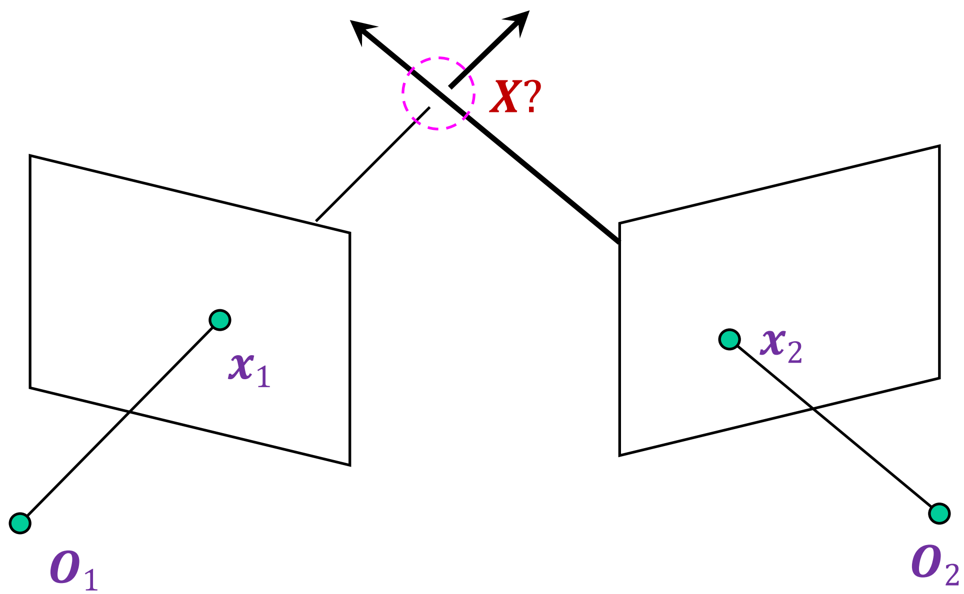
Triangulation

- Given projections of a 3D point in two or more images (with known camera matrices), find the coordinates of the point



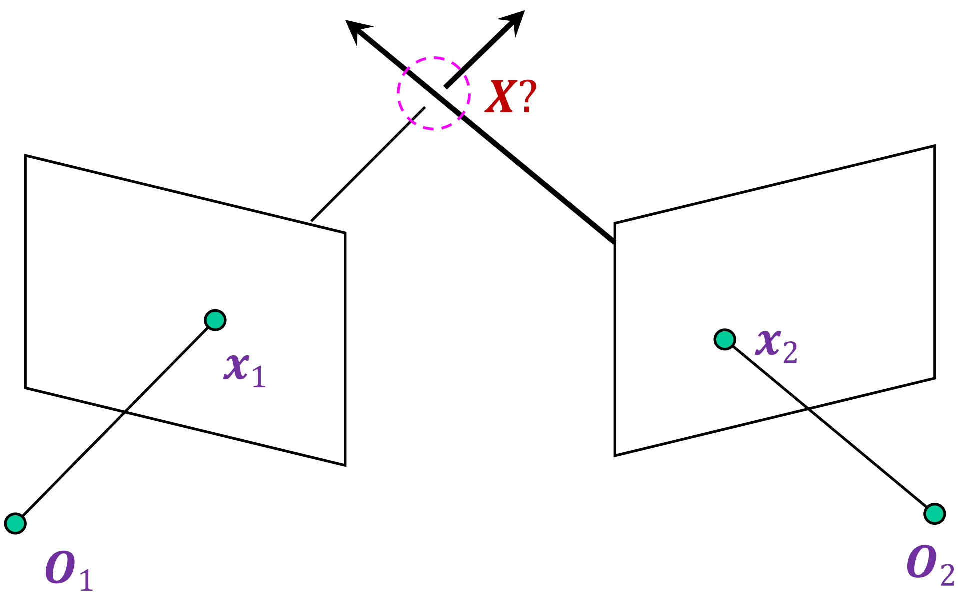
Triangulation

- Given projections of a 3D point in two or more images (with known camera matrices), find the coordinates of the point



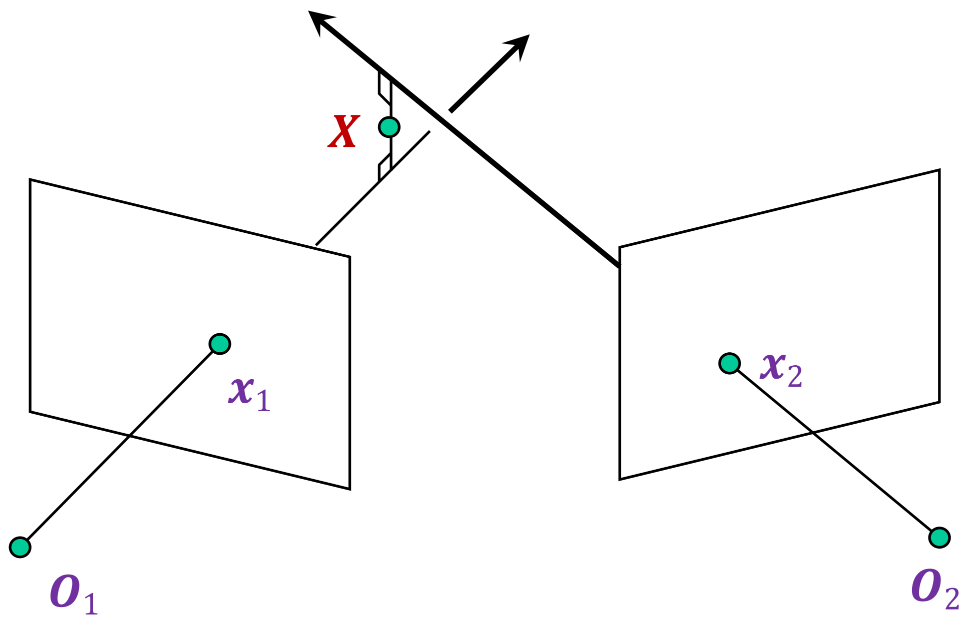
Triangulation

- We want to intersect the two visual rays corresponding to $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, but because of noise and numerical errors, they don't meet exactly



Triangulation: Geometric approach

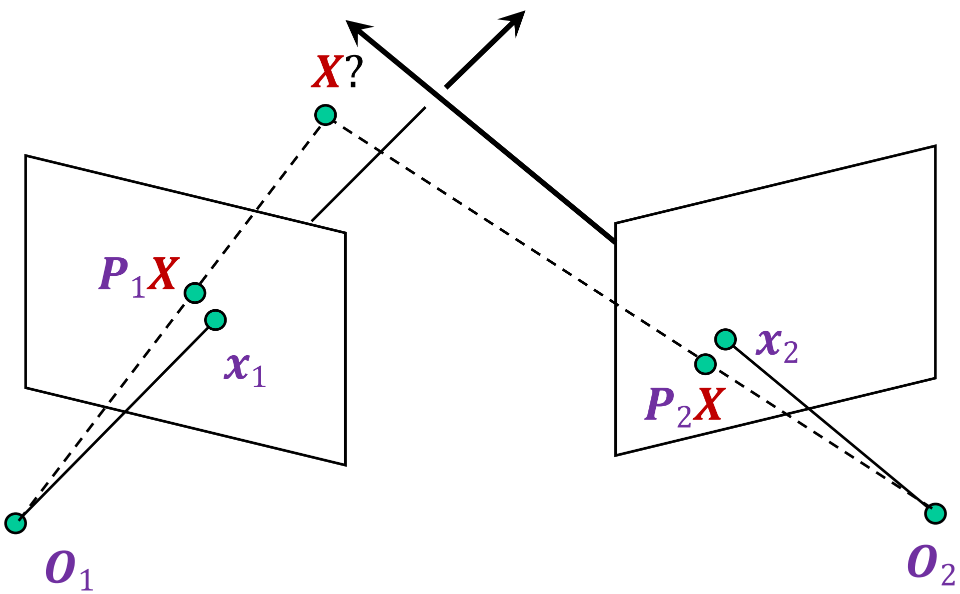
- Find shortest segment connecting the two viewing rays and let $\boldsymbol{X}$ be the midpoint of that segment



Triangulation: Nonlinear approach

- Find $\boldsymbol{X}$ that minimizes

$$\|\text{proj}(\boldsymbol{P}_1\boldsymbol{X}) - \boldsymbol{x}_1\|_2^2 + \|\text{proj}(\boldsymbol{P}_2\boldsymbol{X}) - \boldsymbol{x}_2\|_2^2$$



Triangulation: Linear approach

$$\boldsymbol{x}_1 \cong \boldsymbol{P}_1\boldsymbol{X} \qquad \boldsymbol{x}_1 \times \boldsymbol{P}_1\boldsymbol{X} = \boldsymbol{0} \qquad [\boldsymbol{x}_{1_\times}]\boldsymbol{P}_1\boldsymbol{X} = \boldsymbol{0}$$
$$\boldsymbol{x}_2 \cong \boldsymbol{P}_2\boldsymbol{X} \qquad \boldsymbol{x}_2 \times \boldsymbol{P}_2\boldsymbol{X} = \boldsymbol{0} \qquad [\boldsymbol{x}_{2_\times}]\boldsymbol{P}_2\boldsymbol{X} = \boldsymbol{0}$$

- Rewrite cross product as matrix multiplication:

$$\boldsymbol{a} \times \boldsymbol{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = [\boldsymbol{a}_\times]\boldsymbol{b}$$

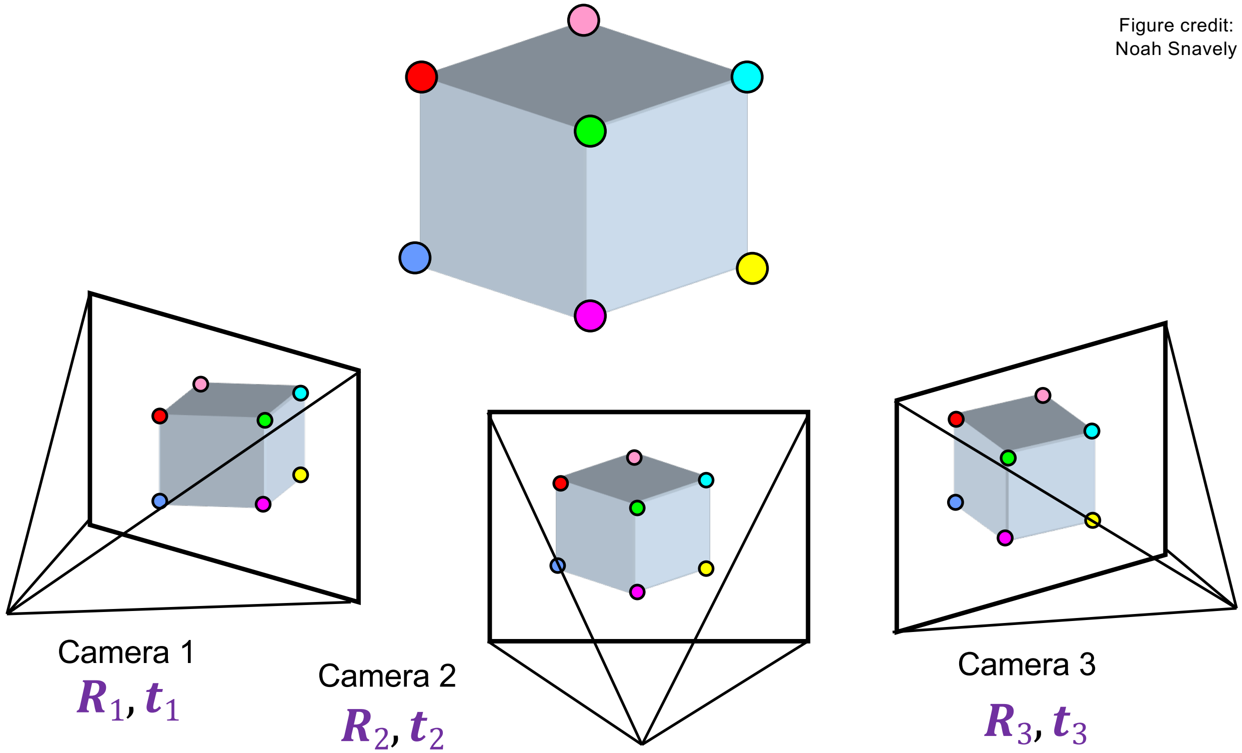
Triangulation: Linear approach

$$\boldsymbol{x}_1 \cong \boldsymbol{P}_1\boldsymbol{X} \qquad \boldsymbol{x}_1 \times \boldsymbol{P}_1\boldsymbol{X} = \boldsymbol{0} \qquad [\boldsymbol{x}_{1_\times}]\boldsymbol{P}_1\boldsymbol{X} = \boldsymbol{0}$$

$$\boldsymbol{x}_2 \cong \boldsymbol{P}_2\boldsymbol{X} \qquad \boldsymbol{x}_2 \times \boldsymbol{P}_2\boldsymbol{X} = \boldsymbol{0} \qquad [\boldsymbol{x}_{2_\times}]\boldsymbol{P}_2\boldsymbol{X} = \boldsymbol{0}$$

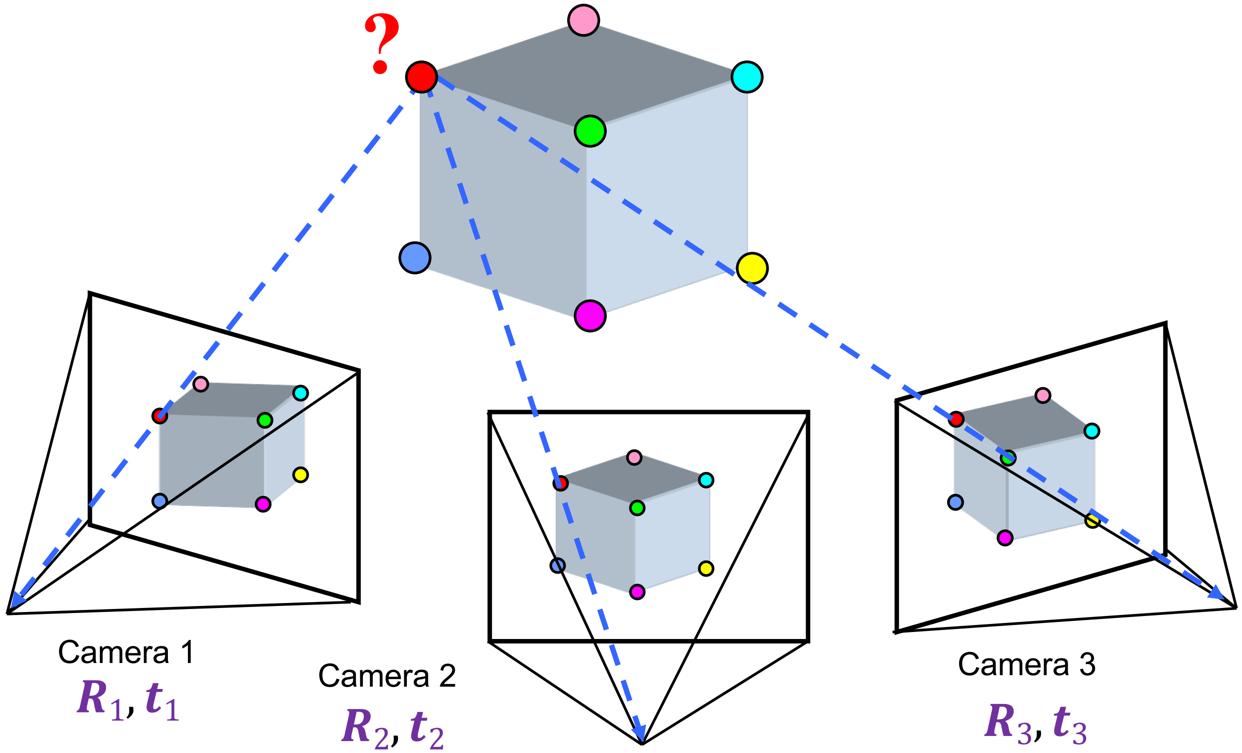
Two independent equations each in terms of three unknown entries of $\boldsymbol{X}$

Preview: Structure from motion



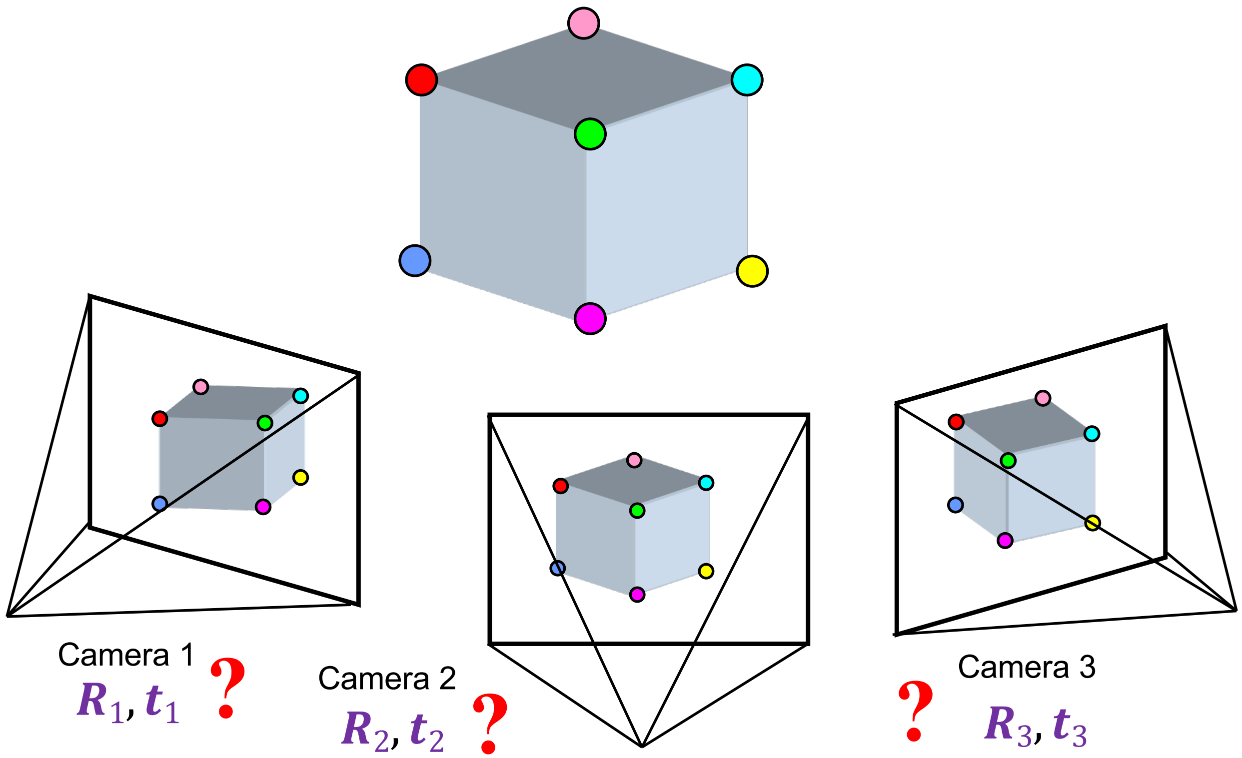
- Given 2D point correspondences between multiple images, compute the camera parameters and the 3D points

Preview: Structure from motion



- **Structure:** Given *known cameras* and projections of the same 3D point in two or more images, compute the 3D coordinates of that point
  - Triangulation!

Preview: Structure from motion



- **Motion:** Given a set of *known* 3D points seen by a camera, compute the camera parameters
  - Calibration!

Multi-view geometry “Bible”