Today

Perspective Projection Geometry

Recall -- Points, Lines, and Planes

Point: represented as a vector $\mathbf{p} = [p_1, p_2, \cdots, p_N]$ **in Euclidean space** $\mathbb{R}^{\boldsymbol{N}}$

Lines in general

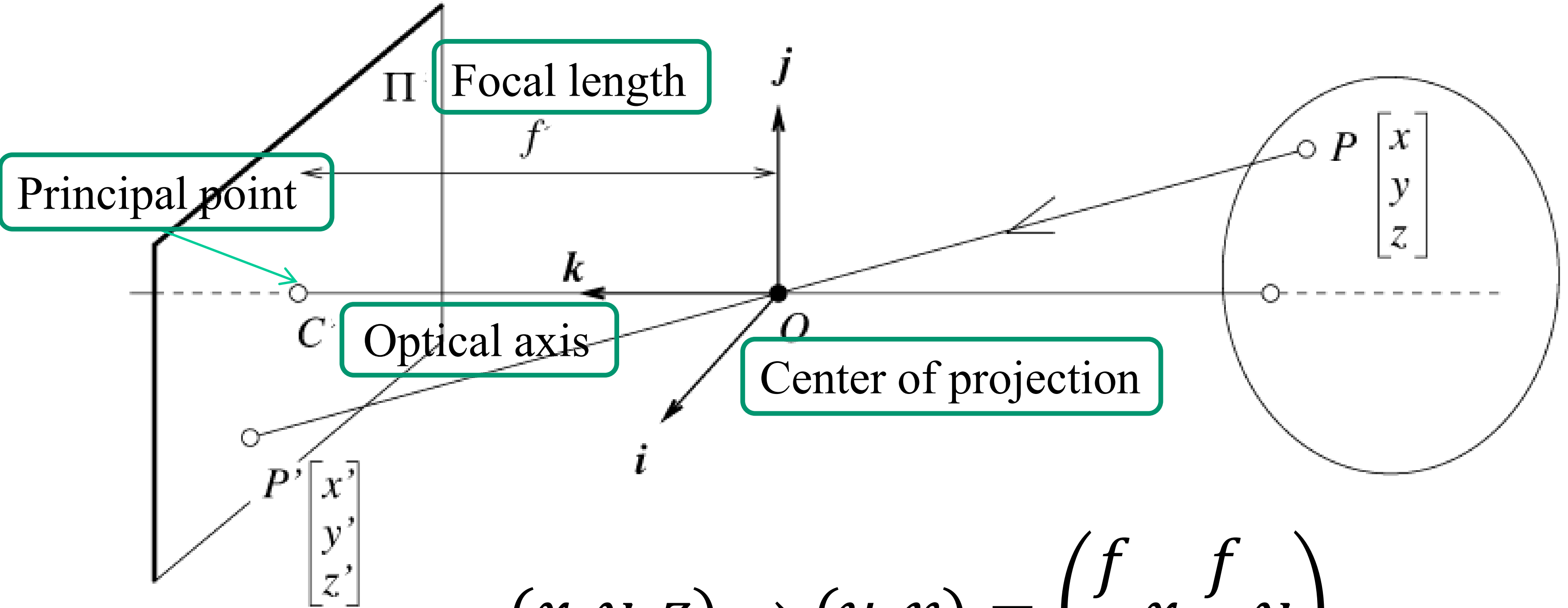
- Parametric form: $\mathbf{p}(t) = \mathbf{p}_0 + t\,\mathbf{d}$, where $\mathbf{p}_0$ is any point on the line and $\mathbf{d}$=$[d_1, d_2, \cdots, d_N]$ is a unit vector - the direction of the line
  - Parallel lines have the same $\mathbf{d}$ with different $\mathbf{p}_0$
- Point-normal: $\mathbf{m} \cdot (\mathbf{p} - \mathbf{p}_0) = 0$, where $\mathbf{p}_0$is any point on the line and $\mathbf{m}$ is normal to the line
- $\mathbf{l} \cdot \mathbf{p} = 0$, where $\mathbf{l}$=$[\mathbf{m}, -\mathbf{m} \cdot \mathbf{p}_0]$, and $\mathbf{p}$=$[p_1, p_2, \cdots, p_N, 1]$

Plane: $\mathbf{n} \cdot (\mathbf{p} - \mathbf{p_0}) = \mathbf{0}$

$\mathbf{n}$ is the normal vector – perpendicular to the plane

Recall -- Perspective Projection

Only one coordinate system – camera coordinate system

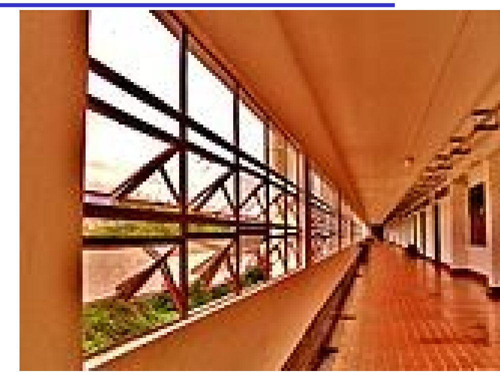


$$(x, y, z) \rightarrow (u, v) = \left(\frac{f}{z}x, \frac{f}{z}y\right)$$

3D object point → 2D image point

The perspective projection is non-linear!

Recall: Properties of Perspective Projection



Points project to points

Lines project to lines

Cross-ratio is preserved after projection

Planes project to the whole or half image

- A plane may only has half of its area in the projection side

Scaling and foreshortening

Angles are not preserved

- Parallel lines may be not projected to parallel lines unless they are parallel to the image plane

Degenerate cases

- Line through center of projection projects to a point.
- Plane through center of projection projects to line

Homogenous Coordinates

The line function: $\mathrm{l} \cdot \mathbf{p} = 0$, where $\mathrm{l}=[a, b, c]$ and $\mathbf{p}=[x, y, 1]$

⇒ $$\mathrm{l} \cdot (\boldsymbol{k}\mathrm{p}) = 0$$

for 2D point (x,y)

- (x,y,1) is the same as (kx,ky,k)

for 3D point (x,y,z)

- (x,y,z,1) is the same as (kx,ky,kz,k)

⇒ **Add an extra coordinate and has an equivalence relation**

Perspective Projection using Matrix-Vector Form

Turn previous expression into homogeneous coordinates

- 3D point in homogeneous coordinates --- $(X, Y, Z, 1)$
- Projected 2D point in image in homogeneous coordinates --- $(\lambda x, \lambda y, \lambda)$

$$\begin{pmatrix} \lambda x \\ \lambda y \\ \lambda \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & {}^{1}/_{f} & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} X \\ Y \\ {}^{Z}/_{f} \end{bmatrix}$$

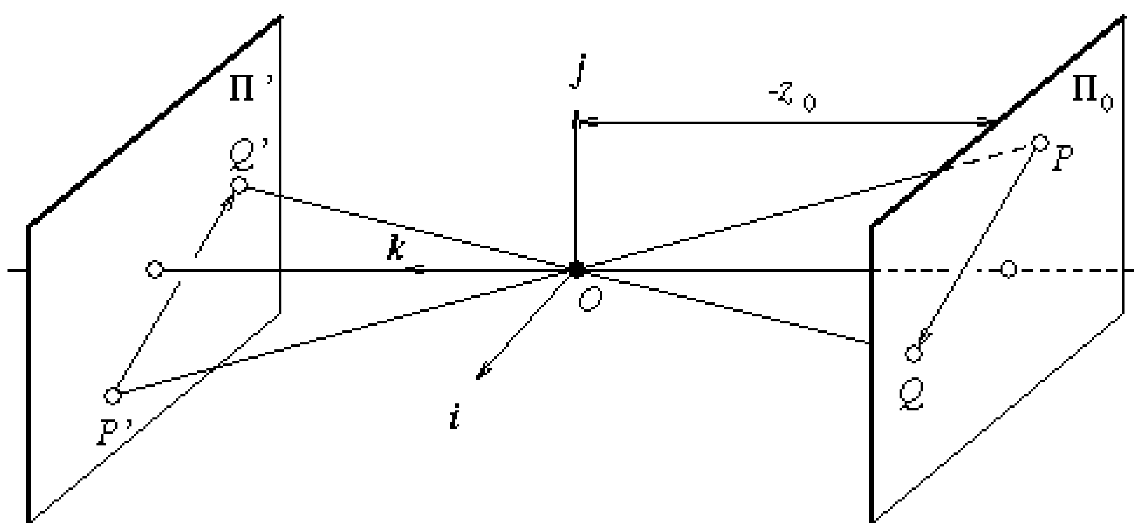
$$\lambda = Z/f$$

Weak Perspective Projection

If the relative depth Δz between two 3D points P and Q is **relative**
small (e.g., $\Delta z < \frac{\bar{z}}{20}$), $z \approx \bar{z}$

Usage:

- Perfect for planar object parallel to the image plane
- collect points into a group at about the same depth, then divide each point by the depth of its group
- Advantage: easy and is acceptable in many conditions



$$x = \frac{f}{\bar{Z}} X$$

$$y = \frac{f}{\bar{Z}} Y$$

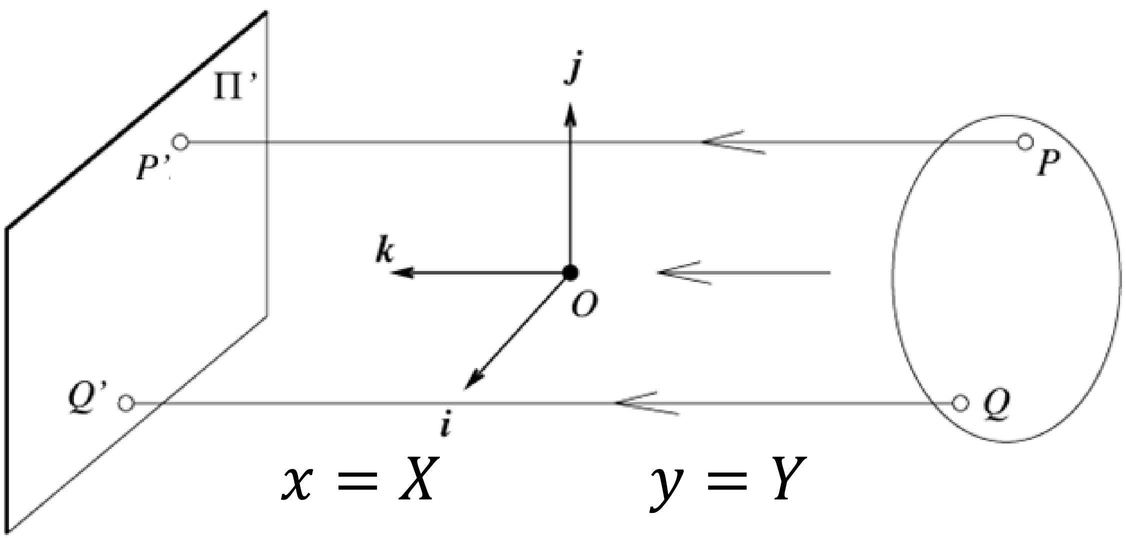
Orthographic Projection

A special case of projection, where all light rays (projection lines) are parallel and orthogonal to the image plane

- $f = z = \infty$
- Parallel lines project to parallel lines
- Size of object does not change on the image plane

Assuming pure orthographic projection is usually unrealistic for a camera

A good model for drawings and computer graphics



$x = X \qquad y = Y$

From a 3D Object Point to a 2D Pixel

In the perspective projection model, both 3D object point and 2D image point are measured in the same coordinate system – the camera coordinate system.

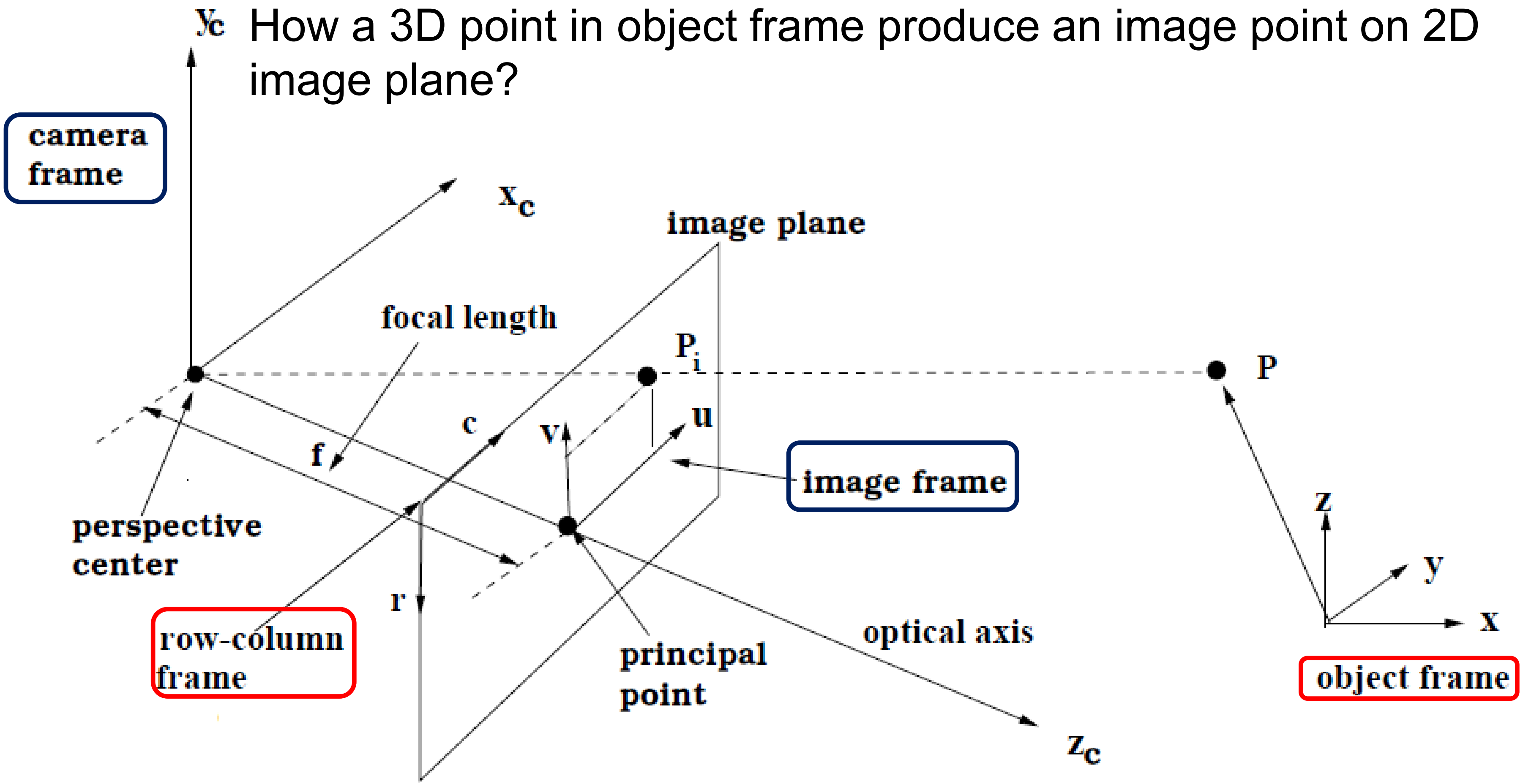
Not true in practice!

More Issues:

- Objects and images are not measured in the camera coordinate system
- Principal point is not the center of image
- How to map mm to pixel?
- The center of 2D image is usually not the origin of the 2D row-column image

Perspective Projection Geometry

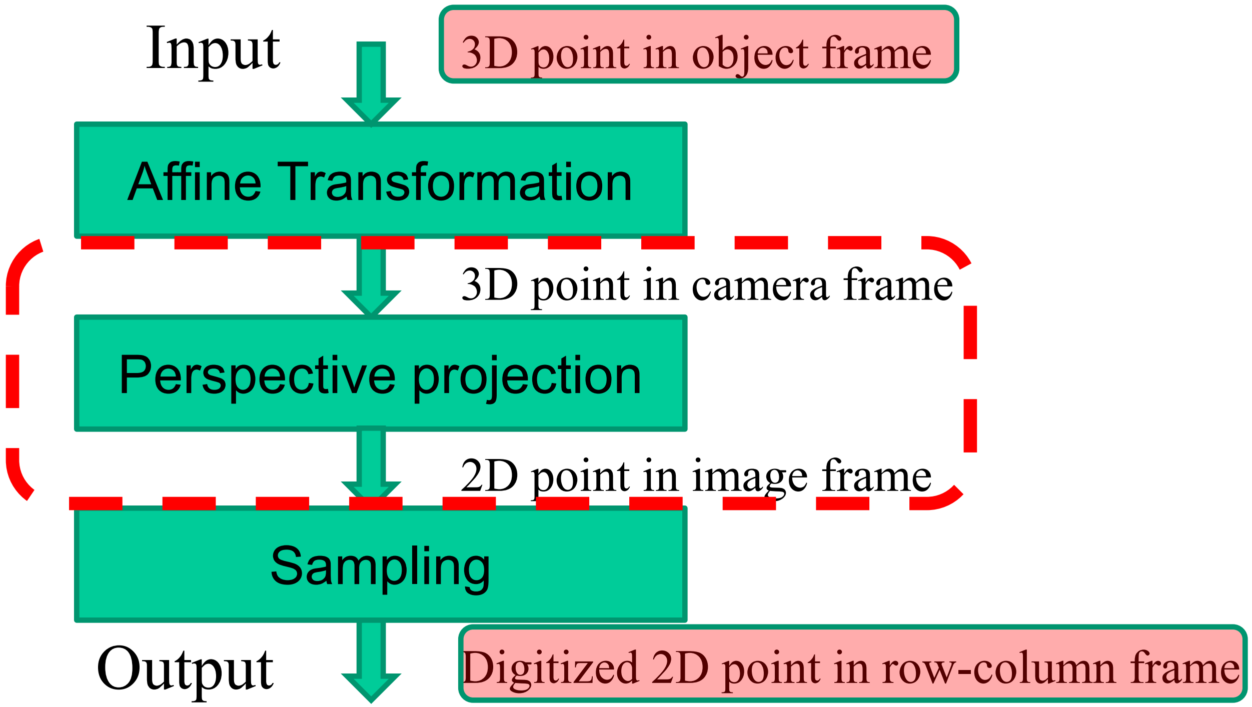
How a 3D point in object frame produce an image point on 2D image plane?



Frame – coordinate system

Figure adapted from Qiang Ji, Computer Vision slides

Projection Process



Camera Parameters

Intrinsic parameters: ⟵ Camera calibration

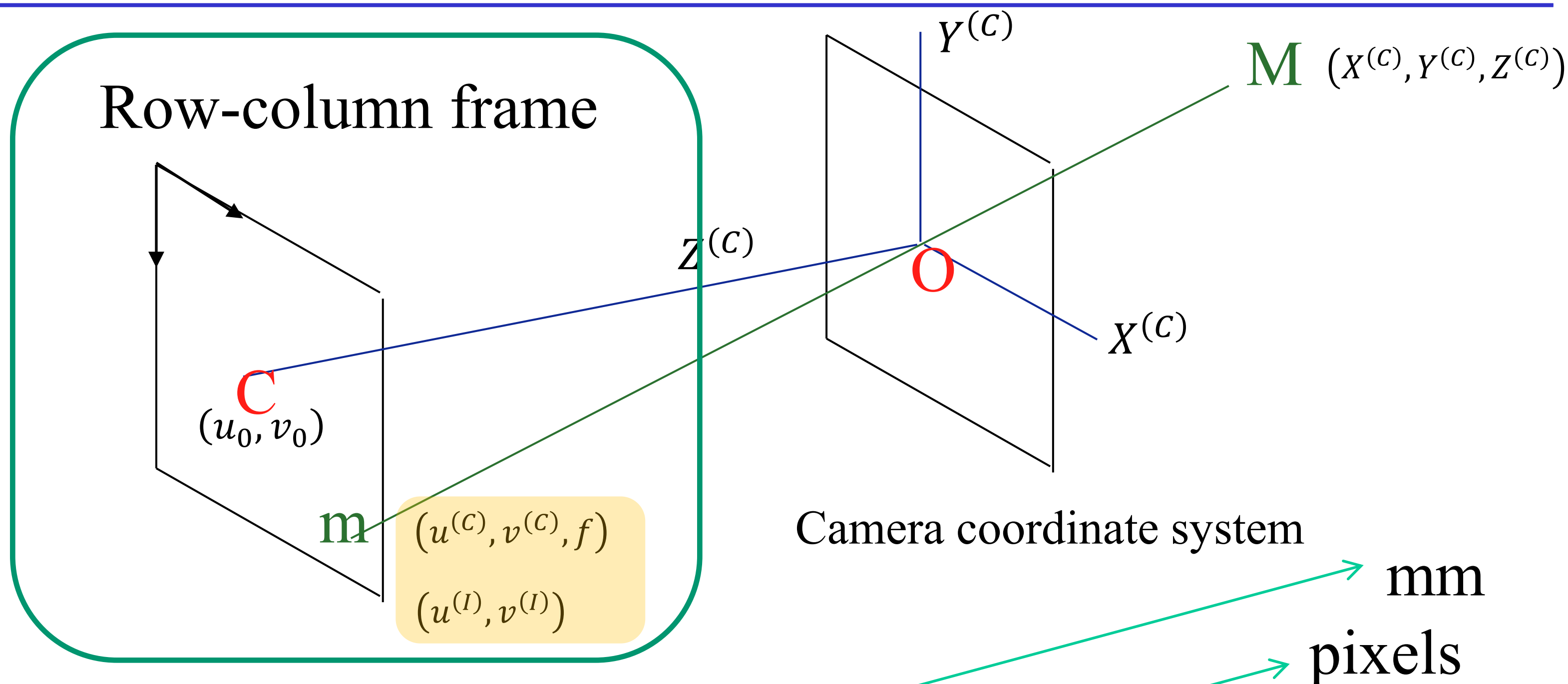
- Independent of the position and orientation of the camera
- Unique for a camera
- focal length, principal point, aspect ratio, angle between axes, etc.
- E.g., one unit in camera coordinates may not be the same as one unit in world coordinates

Extrinsic parameters: ⟵ Pose estimation

- the relative pose between the object frame and the camera frame
- E.g., camera may not be at the origin of object frame, looking down the z-axis

Intrinsic+ extrinsic parameters ⟶ Full projection matrix

From Image Plane to Row-Column Frame: Spatial Sampling & Affine Transformation



k_u and k_v are spatial sampling rate (pixels/mm)

$$\begin{bmatrix} u^{(I)} \\ v^{(I)} \\ 1 \end{bmatrix} = \begin{bmatrix} k_u & 0 & 0 \\ 0 & k_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u^{(c)} \\ v^{(c)} \\ 1 \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \\ 0 \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u^{(c)} \\ v^{(c)} \\ 1 \end{bmatrix}$$

Affine transformation

Project a 3D Point in Camera Coordinate System to a 2D Point in Row-Column Frame

Since $$\lambda \begin{bmatrix} u^{(C)} \\ v^{(C)} \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X^{(C)} \\ Y^{(C)} \\ Z^{(C)} \\ 1 \end{bmatrix}$$

$$\lambda \begin{bmatrix} u^{(I)} \\ v^{(I)} \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u^{(C)} \\ v^{(C)} \\ 1 \end{bmatrix} =$$

$$\begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X^{(C)} \\ Y^{(C)} \\ Z^{(C)} \\ 1 \end{bmatrix} = \begin{bmatrix} fk_u & 0 & u_0 & 0 \\ 0 & fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X^{(C)} \\ Y^{(C)} \\ Z^{(C)} \\ 1 \end{bmatrix}$$

Project a 3D Point in Camera Coordinate System to a 2D Point in Row-Column Frame

$$\lambda \begin{bmatrix} u^{(I)} \\ v^{(I)} \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X^{(C)} \\ Y^{(C)} \\ Z^{(C)} \\ 1 \end{bmatrix}$$

$$f_u = f k_u$$
$$f_v = f k_v$$

Intrinsic Camera Parameters

Intrinsic Parameters (Do not depend on camera position):

1. $f_u = f k_u$
2. $f_v = f k_v$
3. u_0
4. v_0

Intrinsic Parameters

$$\lambda \begin{bmatrix} u^{(I)} \\ v^{(I)} \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X^{(C)} \\ Y^{(C)} \\ Z^{(C)} \\ 1 \end{bmatrix}$$

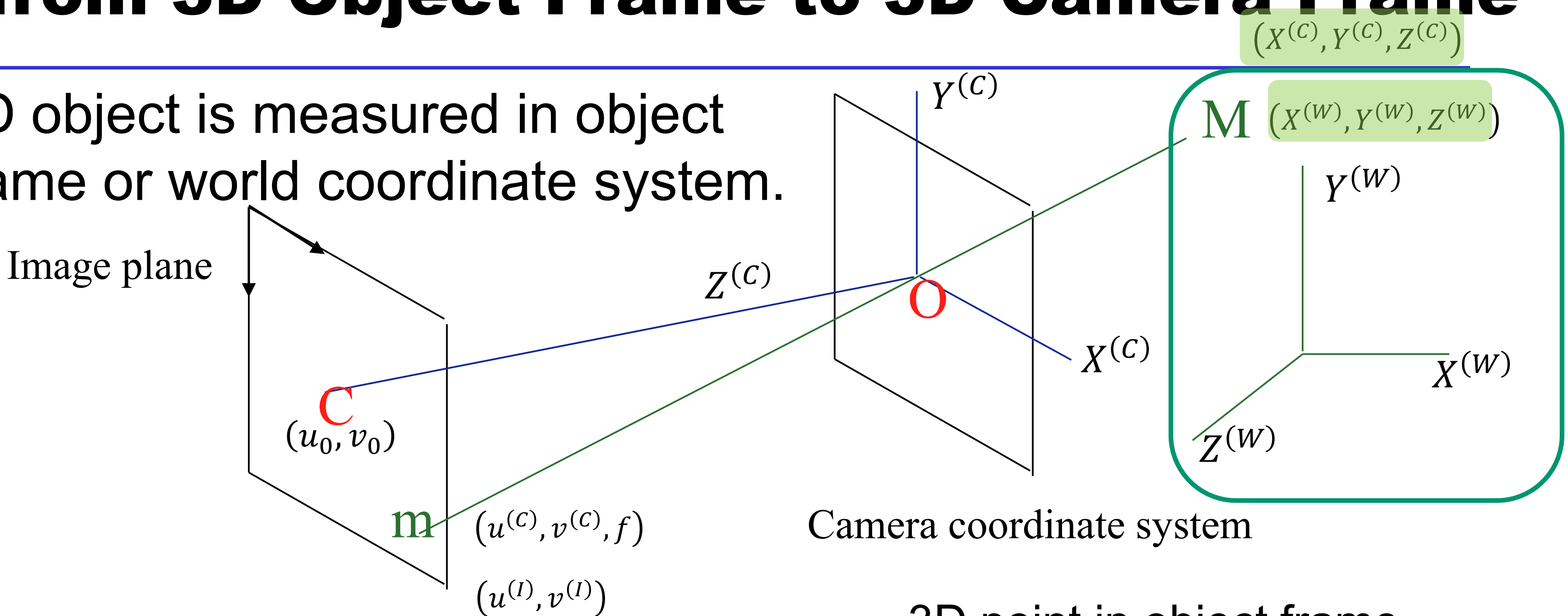
$$\lambda \mathbf{m}^{(I)} = \mathbf{W}\mathbf{M}^{(C)} = \begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \mathbf{w}_3^T \end{bmatrix} \mathbf{M}^{(C)} = \begin{bmatrix} \mathbf{w}_1^T \mathbf{M}^{(C)} \\ \mathbf{w}_2^T \mathbf{M}^{(C)} \\ \mathbf{w}_3^T \mathbf{M}^{(C)} \end{bmatrix}$$

2D pixel location

$\mathbf{w}_i$ is a row vector of $\mathbf{W}$

Extrinsic Parameters: Rigid Transformation from 3D Object Frame to 3D Camera Frame

3D object is measured in object frame or world coordinate system.



3D point in object frame

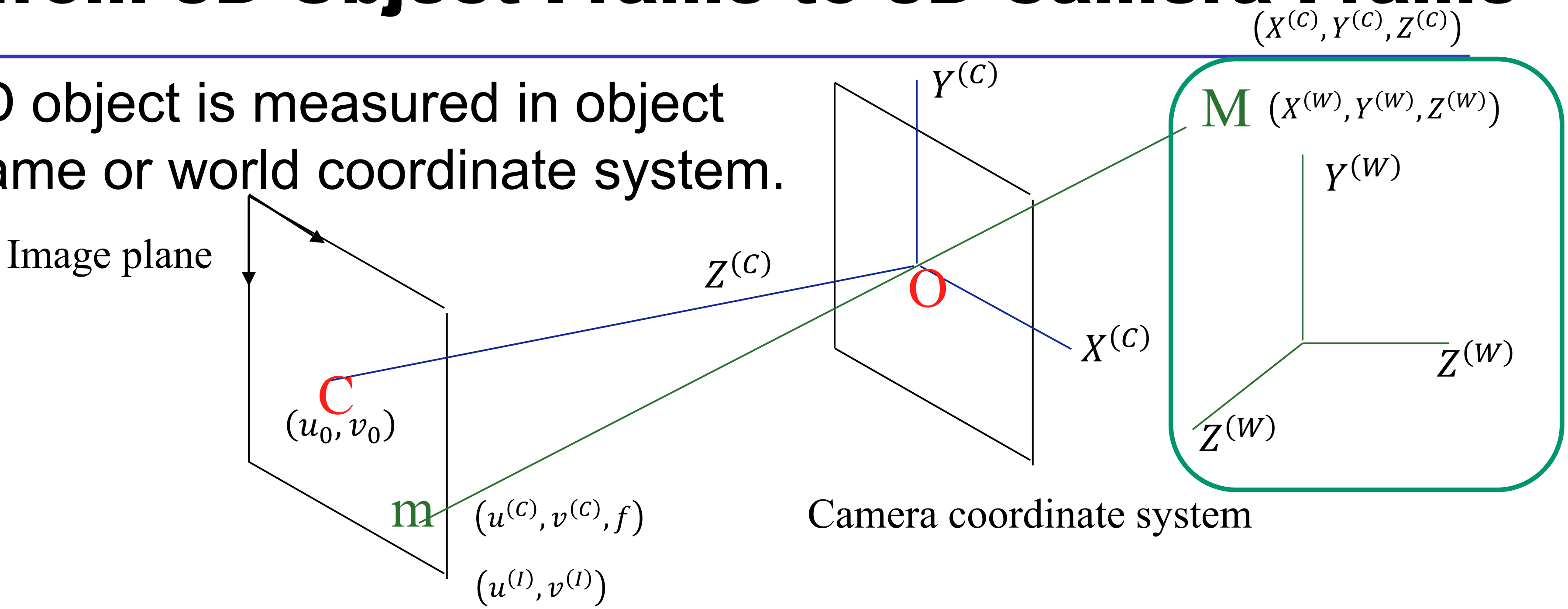
By Rigid Body Transformation:

$$\begin{bmatrix} X^{(C)} \\ Y^{(C)} \\ Z^{(C)} \end{bmatrix} = \mathbf{R}_{3\times3} \begin{bmatrix} X^{(W)} \\ Y^{(W)} \\ Z^{(W)} \end{bmatrix} + \mathbf{T}_{3\times1}$$

3D point in camera frame

Extrinsic Parameters: Rigid Transformation from 3D Object Frame to 3D Camera Frame

3D object is measured in object frame or world coordinate system.



By Rigid Body (Affine) Transformation:

$$\begin{bmatrix} X^{(C)} \\ Y^{(C)} \\ Z^{(C)} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{T}_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} X^{(W)} \\ Y^{(W)} \\ Z^{(W)} \\ 1 \end{bmatrix}$$

3D point in camera frame

$$\Rightarrow \mathbf{M}^{(C)} = \mathbf{D}\mathbf{M}^{(W)}$$

3D point in object frame

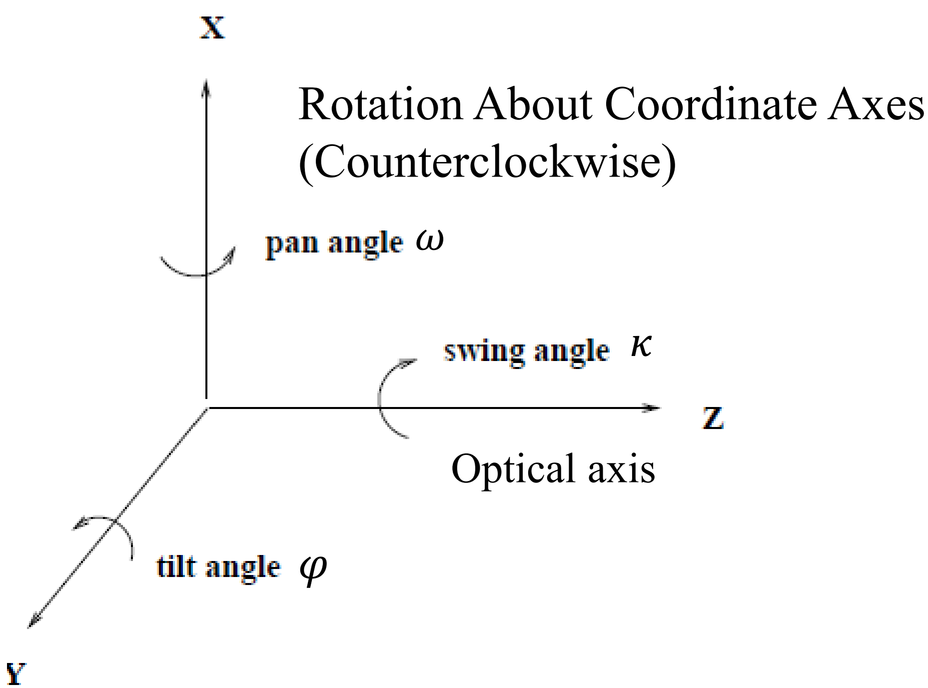
Rotation Matrix Representation

The rotation matrix can be decomposed as a sequential combination of these three special rotations.

$$\mathbf{R} = \mathbf{R}_z \mathbf{R}_y \mathbf{R}_x$$

Order matters!

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & \sin\omega \\ 0 & -\sin\omega & \cos\omega \end{bmatrix}$$



$$\mathbf{R}_y = \begin{bmatrix} \cos\varphi & 0 & -\sin\varphi \\ 0 & 1 & 0 \\ \sin\varphi & 0 & \cos\varphi \end{bmatrix}$$

$$\mathbf{R}_z = \begin{bmatrix} \cos\kappa & \sin\kappa & 0 \\ -\sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Property of the Rotation Matrix

The rotation matrix is an orthogonal matrix

$$\mathbf{R} = \mathbf{R}_z\mathbf{R}_y\mathbf{R}_x \quad \mathbf{R}^T = {\mathbf{R}_x}^T{\mathbf{R}_y}^T{\mathbf{R}_z}^T$$

$$\mathbf{R}^T = \mathbf{R}^{-1} \Leftrightarrow \mathbf{R}\mathbf{R}^T = \mathbf{I}$$

$$\text{Let } \mathbf{R} = \begin{bmatrix} \mathbf{r}_1^t \\ \mathbf{r}_2^t \\ \mathbf{r}_3^t \end{bmatrix} \qquad \text{Then } \mathbf{r}_i \cdot \mathbf{r}_j = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$$

$$\mathbf{R} = \begin{bmatrix} \cos\varphi\cos\kappa & \cos\omega\sin\kappa + \sin\omega\sin\varphi\cos\kappa & \sin\omega\sin\kappa - \cos\omega\sin\varphi\cos\kappa \\ -\cos\varphi\sin\kappa & \cos\omega\cos\kappa - \sin\omega\sin\varphi\sin\kappa & \sin\omega\cos\kappa + \cos\omega\sin\varphi\sin\kappa \\ \sin\varphi & -\sin\omega\cos\varphi & \cos\omega\cos\varphi \end{bmatrix}$$

Extrinsic Camera Parameters

Three rotation angles ω φ κ

Three translation parameters t_X t_Y t_Z

Camera Parameters

Intrinsic parameters: (Do not depend on camera position):

1. $f_u = f k_u$
2. $f_v = f k_v$
3. u_0
4. v_0

Extrinsic parameters:

- Three rotation angles $\omega \quad \varphi \quad \kappa$
- Three translation parameters $t_X \; t_Y \; t_Z$

Full Perspective Camera Model

$$\lambda \mathbf{m}^{(I)} = \mathbf{W}\mathbf{M}^{(C)}, \text{and } \mathbf{M}^{(C)} = \mathbf{D}\mathbf{M}^{(W)} \Rightarrow \lambda \mathbf{m}^{(I)} = \mathbf{W}\mathbf{D}\mathbf{M}^{(W)}$$

$$\lambda \begin{bmatrix} u^{(I)} \\ v^{(I)} \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{T}_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} X^{(W)} \\ Y^{(W)} \\ Z^{(W)} \\ 1 \end{bmatrix}$$

Full Perspective Camera Model

Let $\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1^t \\ \mathbf{r}_2^t \\ \mathbf{r}_3^t \end{bmatrix}$ and $\mathbf{T} = \begin{bmatrix} t_X \\ t_Y \\ t_Z \end{bmatrix}$

$$\begin{bmatrix} f_u & 0 & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{3\times 3} & \mathbf{T}_{3\times 1} \\ 0_{1\times 3} & 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1^t & t_X \\ \mathbf{r}_2^t & t_Y \\ \mathbf{r}_3^t & t_Z \\ 0_{3\times 1} & 1 \end{bmatrix}$$

$$= \begin{bmatrix} f_u \mathbf{r}_1^t + u_0 \mathbf{r}_3^t & f_u t_X + u_0 t_z \\ f_v \mathbf{r}_2^t + v_0 \mathbf{r}_3^t & f_v t_Y + v_0 t_z \\ \mathbf{r}_3^t & t_z \end{bmatrix}$$

Full Perspective Camera Model

$$\begin{bmatrix} f_u & 0 & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{3\times 3} & \mathbf{T}_{3\times 1} \\ 0_{1\times 3} & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_u \mathbf{r}_1^t + u_0 \mathbf{r}_3^t & f_u t_X + u_0 t_z \\ f_v \mathbf{r}_2^t + v_0 \mathbf{r}_3^t & f_v t_Y + v_0 t_z \\ \mathbf{r}_3^t & t_z \end{bmatrix}}$$

Projection matrix P

$$\lambda \begin{bmatrix} u^{(I)} \\ v^{(I)} \\ 1 \end{bmatrix} = \begin{bmatrix} f_u \mathbf{r}_1^t + u_0 \mathbf{r}_3^t & f_u t_X + u_0 t_z \\ f_v \mathbf{r}_2^t + v_0 \mathbf{r}_3^t & f_v t_Y + v_0 t_z \\ \mathbf{r}_3^t & t_z \end{bmatrix} \begin{bmatrix} X^{(W)} \\ Y^{(W)} \\ Z^{(W)} \\ 1 \end{bmatrix}$$

$$\lambda = Z^{(C)} = \mathbf{r}_3{}^T \mathbf{M}^{(W)} + t_z$$

Full Perspective Camera Model

$$\lambda \mathbf{m}^{(I)} = \mathbf{P}\mathbf{M}^{(W)}$$

Collinearity equation

$$u^{(I)} - u_0 = f_u \frac{r_{11}X^{(W)} + r_{12}Y^{(W)} + r_{13}Z^{(W)} + t_X}{r_{31}X^{(W)} + r_{32}Y^{(W)} + r_{33}Z^{(W)} + t_Z}$$

$$v^{(I)} - v_0 = f_v \frac{r_{21}X^{(W)} + r_{22}Y^{(W)} + r_{23}Z^{(W)} + t_Y}{r_{31}X^{(W)} + r_{32}Y^{(W)} + r_{33}Z^{(W)} + t_Z}$$