

Image-based Modeling and Rendering:

6. Coordinate and Transformation

I-Chen Lin, Assistant Professor

Dept. of CS, National Chiao Tung University

Objective

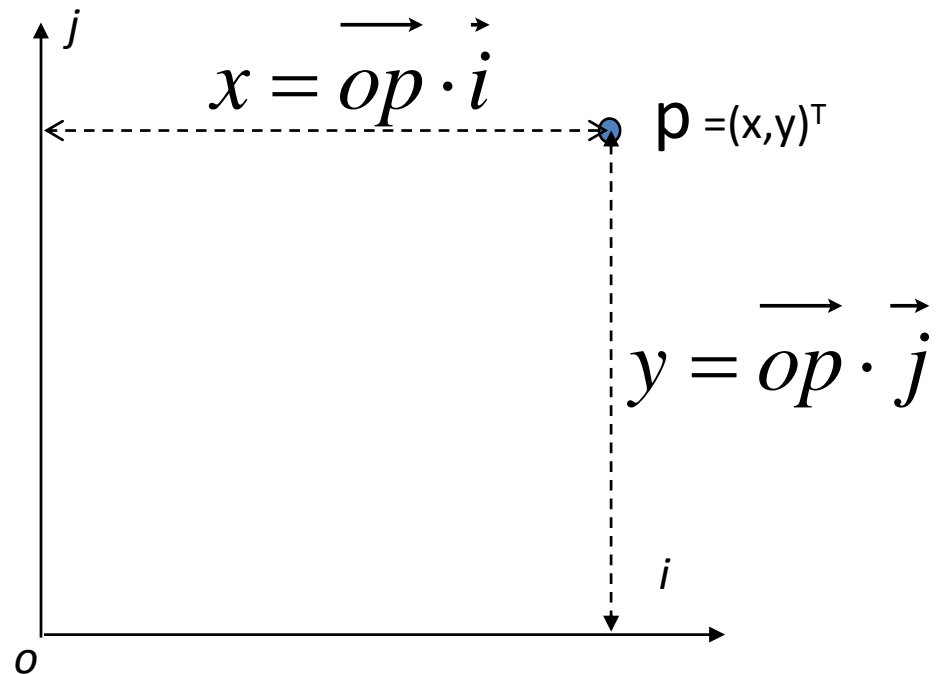
- Geometric camera models
 - Intrinsic and extrinsic parameters
 - Projection equations

Slides are modified from the reference lecture notes or project pages:

- Prof. J. Rehg, Computer Vision, Georgia Inst. of Tech.
- Prof. T. Darrell, Computer Vision and Applications, MIT.
- David A. Forsyth and Jean Ponce, Computer Vision: A Modern Approach, Prentice Hall, New Jersey, 2003.
- Prof. D.A. Forsyth, Computer Vision, UIUC.
- E. Angel, Interactive Computer Graphics, 5th Ed., Addison Wesley

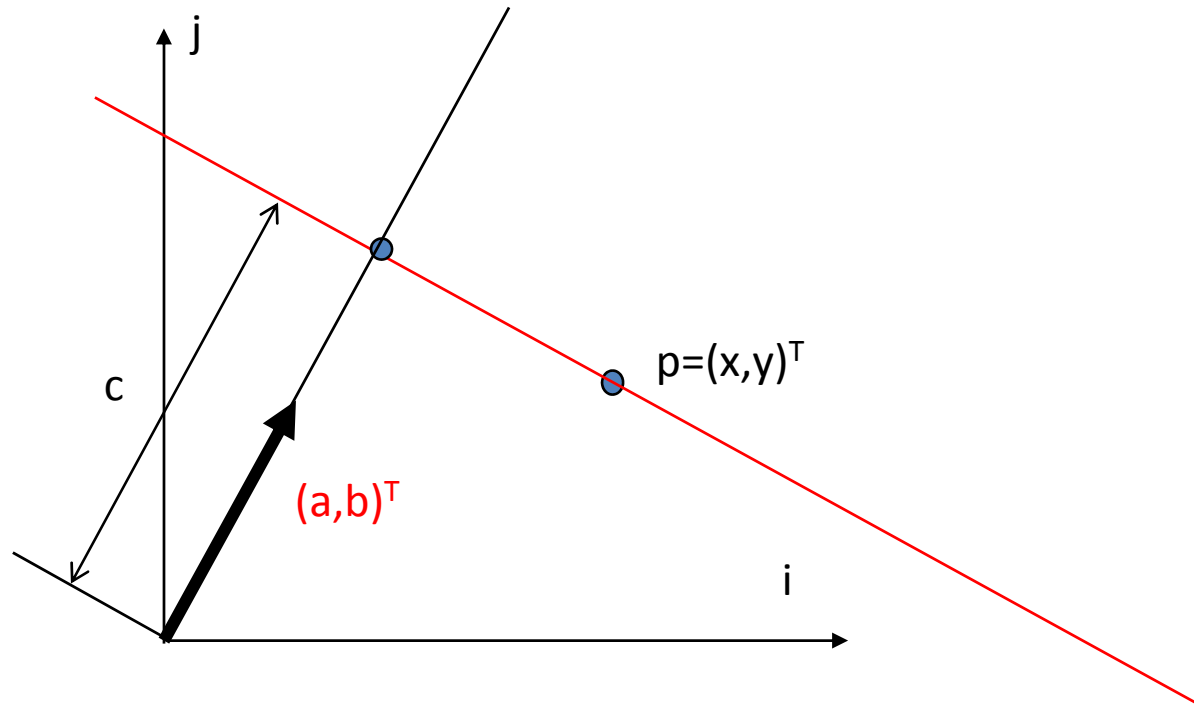
2D coordinate frames & points

- Coordinates x and y
- For a more general coordinate representation, we usually use a vector form.



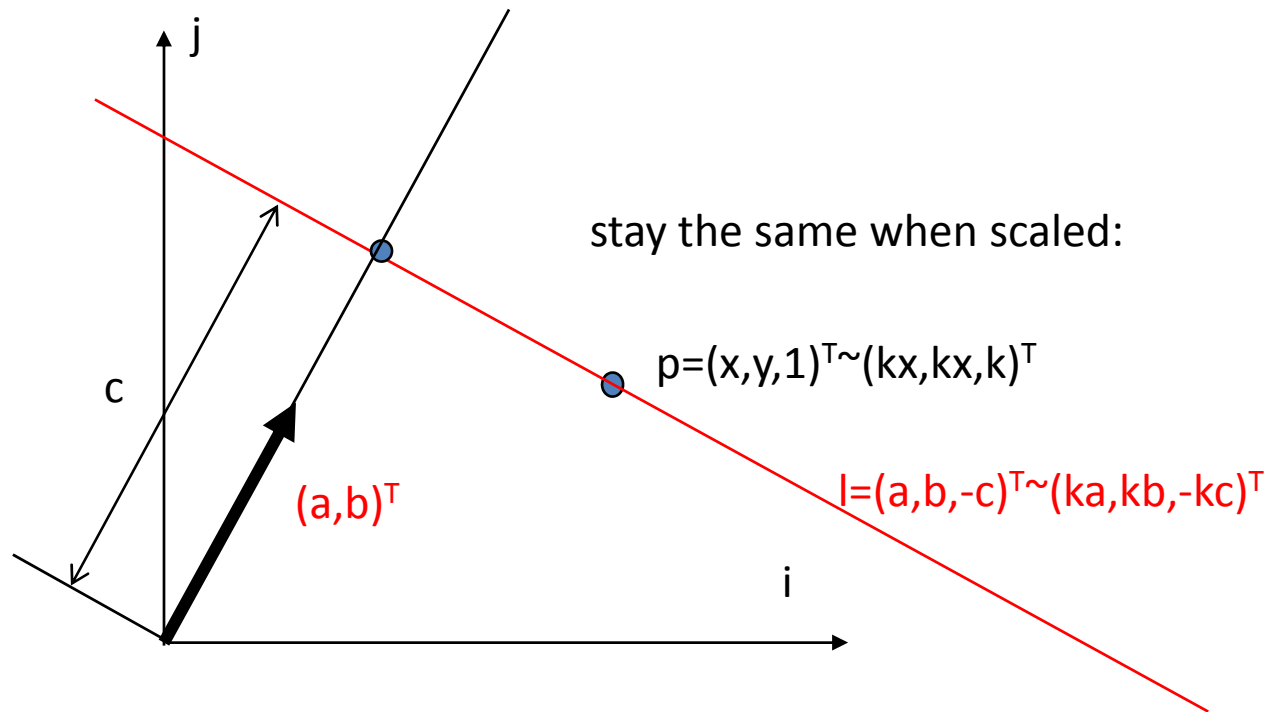
2D lines

- Line l : $ax+by=c \leftrightarrow (a,b)^T(x,y)=c$



Homogeneous coordinates

- Uniform treatment of points and lines
- Line-point incidence: $l^T p = 0$

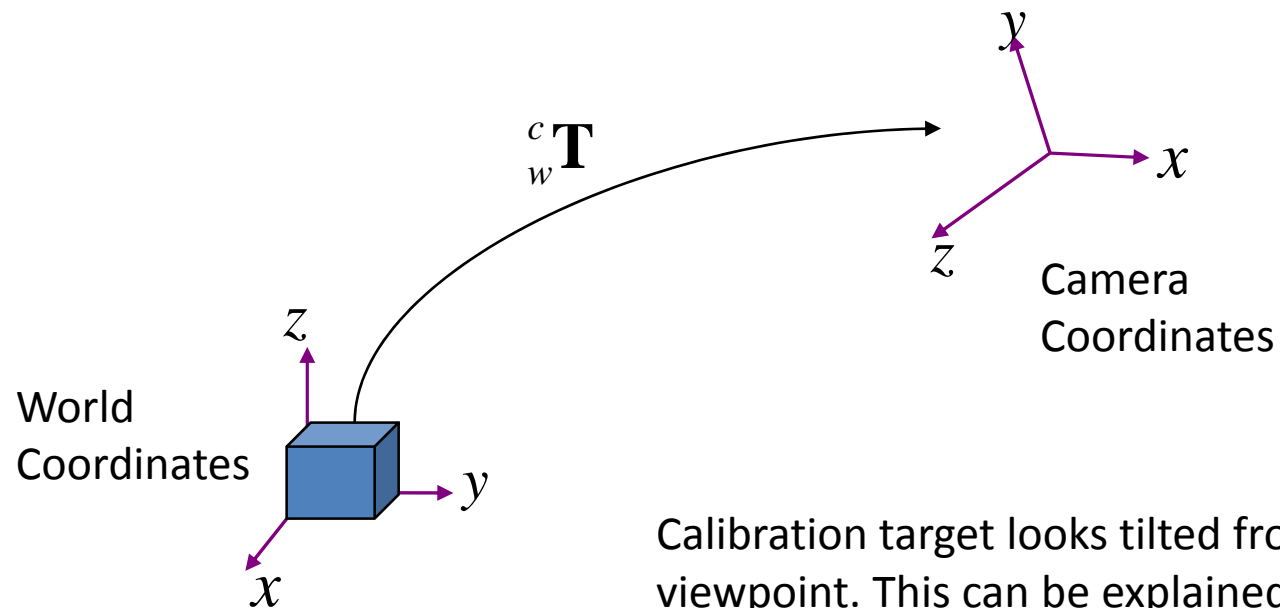


Homogeneous coordinates (cont.)

- Furthermore, ...
 - We use homogenous coordinates to combine rotation and translation into same framework: matrix transformation.
 - It allows easy transformation between “frames” – common between computer graphics and vision.

Camera pose

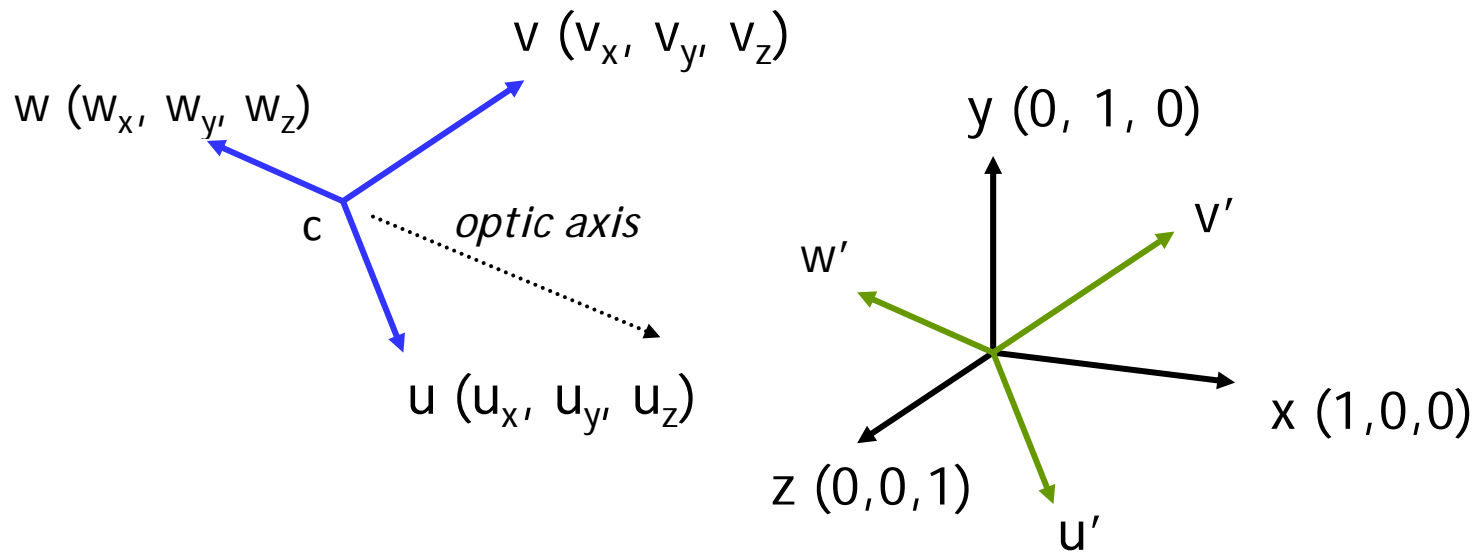
- To apply the camera model, objects in the scene must be expressed in *camera coordinates*.



Calibration target looks tilted from camera viewpoint. This can be explained as a difference in coordinate systems.

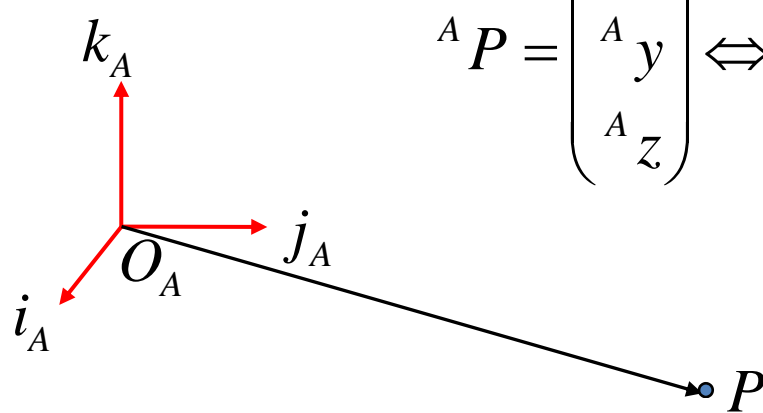
Rigid body transformations

- Need a way to specify the six degrees-of-freedom of a rigid body.
- 3 rotation and 3 translation DOFs.
- R, t : the extrinsic parameters.



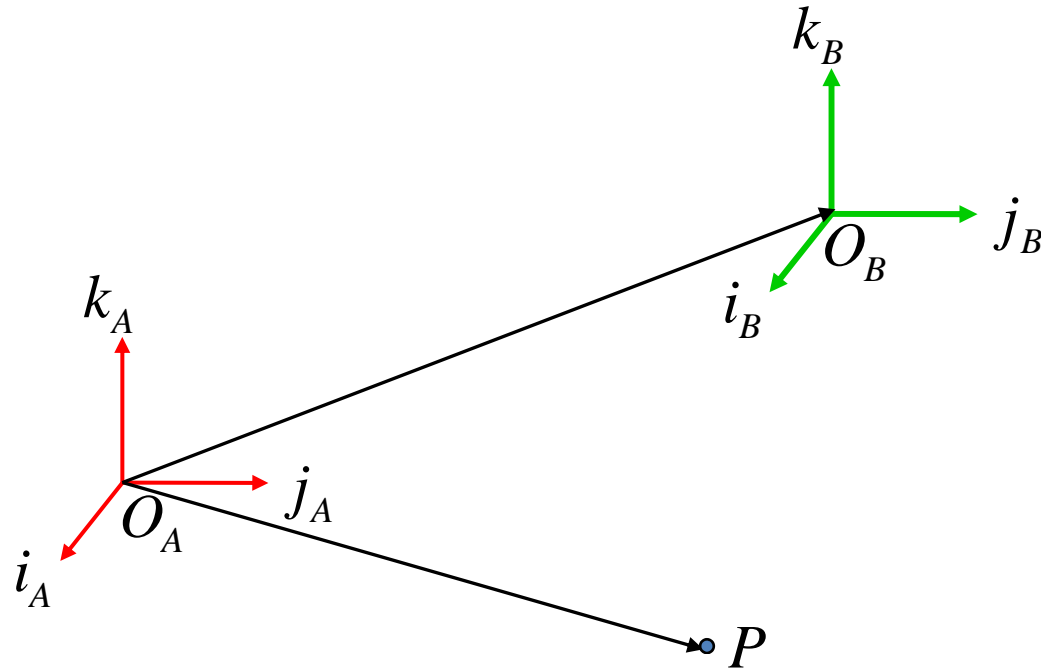
Notations

- Superscript references coordinate frame
- ${}^A P$ is coordinates of P in frame A
- ${}^B P$ is coordinates of P in frame B


$${}^A P = \begin{pmatrix} {}^A x \\ {}^A y \\ {}^A z \end{pmatrix} \Leftrightarrow \overline{OP} = ({}^A x \cdot \overline{i_A}) + ({}^A y \cdot \overline{j_A}) + ({}^A z \cdot \overline{k_A})$$

Translation

$${}^B P = {}^A P + {}^B(O_A)$$



Translation

- Using homogeneous coordinates, translation can be expressed as a matrix multiplication.

$${}^B P = {}^A P + {}^B O_A$$

$$\begin{bmatrix} {}^B P \\ 1 \end{bmatrix} = \begin{bmatrix} I & {}^B O_A \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^A P \\ 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

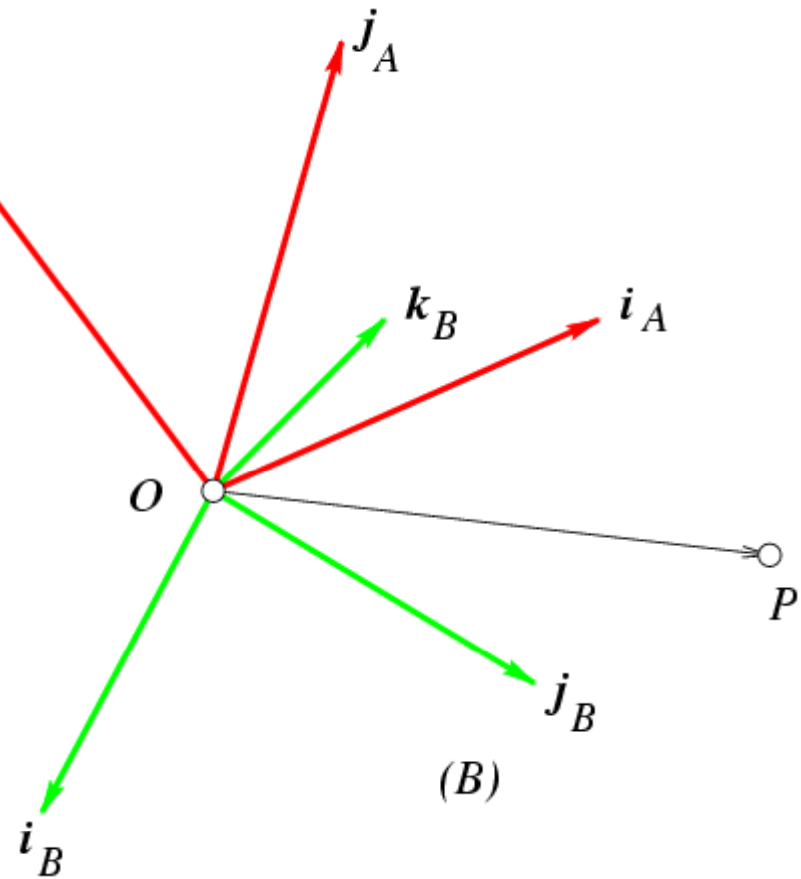
- Translation is commutative

Rotation

$$\overline{OP} = (i_A \quad j_A \quad k_A) \begin{pmatrix} {}^A x \\ {}^A y \\ {}^A z \end{pmatrix} = (i_B \quad j_B \quad k_B) \begin{pmatrix} {}^B x \\ {}^B y \\ {}^B z \end{pmatrix}$$

$${}^B P = {}^B R^A P$$

${}^B R^A$ means describing frame A in
The coordinate system of
frame B



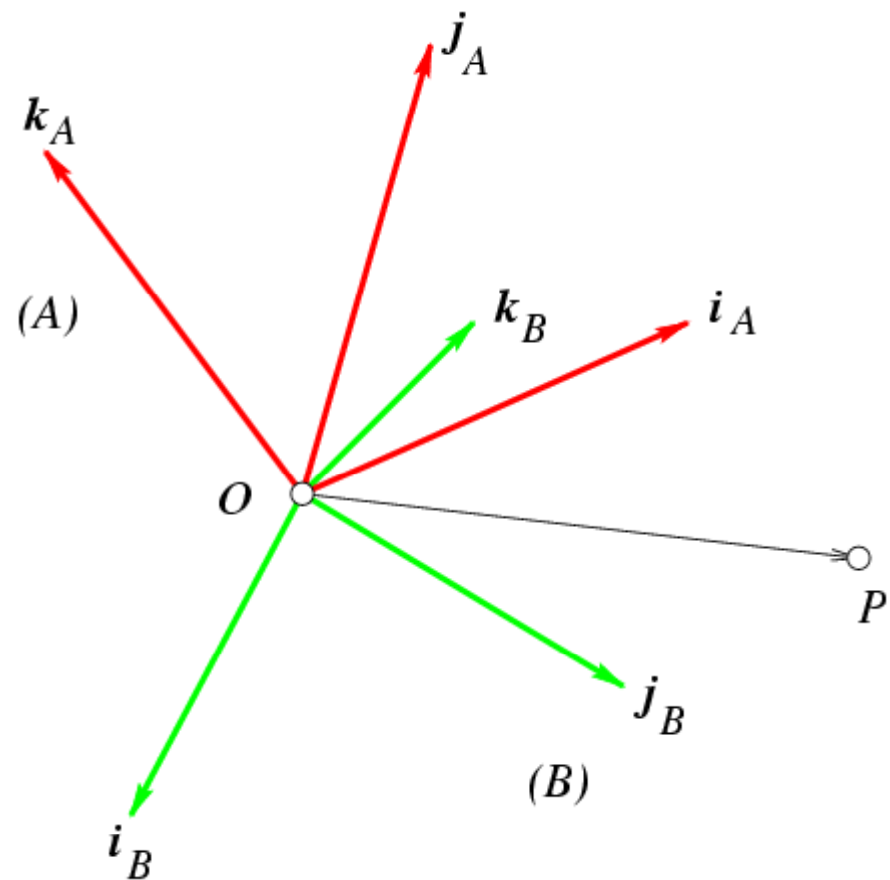
Rotation (from frame A to B)

$${}^B_A R = \begin{bmatrix} \mathbf{i}_A \cdot \mathbf{i}_B & \mathbf{j}_A \cdot \mathbf{i}_B & \mathbf{k}_A \cdot \mathbf{i}_B \\ \mathbf{i}_A \cdot \mathbf{j}_B & \mathbf{j}_A \cdot \mathbf{j}_B & \mathbf{k}_A \cdot \mathbf{j}_B \\ \mathbf{i}_A \cdot \mathbf{k}_B & \mathbf{j}_A \cdot \mathbf{k}_B & \mathbf{k}_A \cdot \mathbf{k}_B \end{bmatrix}$$

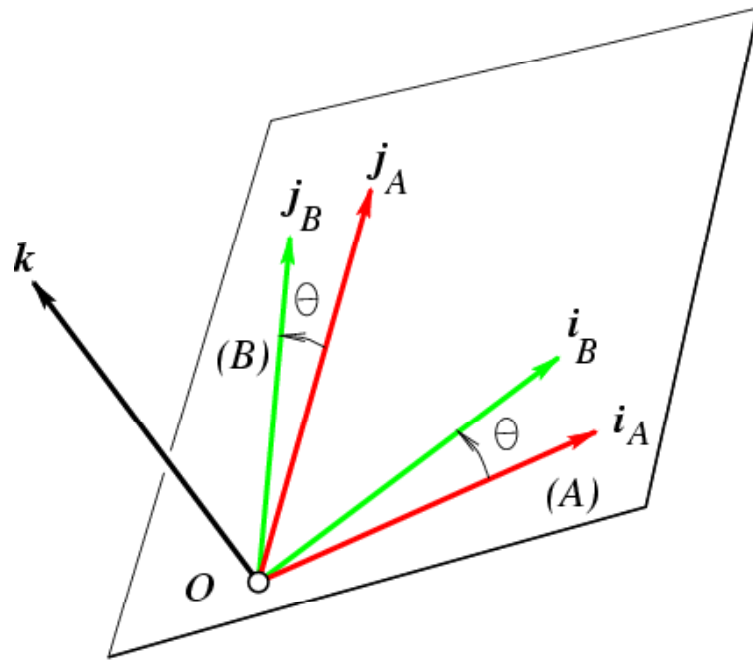
$$= \begin{bmatrix} {}^B \mathbf{i}_A & {}^B \mathbf{j}_A & {}^B \mathbf{k}_A \end{bmatrix}$$

$$= \begin{bmatrix} {}^A \mathbf{i}_B^T \\ {}^A \mathbf{j}_B^T \\ {}^A \mathbf{k}_B^T \end{bmatrix}$$

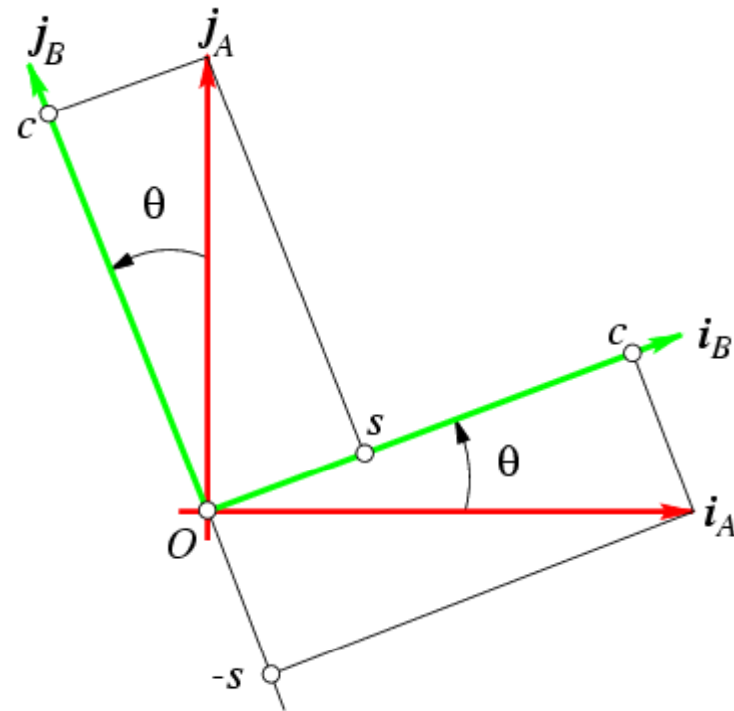
Orthogonal matrix: $R^{-1} = R^T$



Example: Rotation about z axis



What is the rotation matrix?



Combine 3 to get arbitrary rotation

- Euler angles: Z, X', Y''
- Heading, pitch roll: world Z, new X, new Y
- Three basic matrices: order matters, but we'll probably not focus on that

$$R_Z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R_X(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}$$

$$R_Y(\kappa) = \begin{bmatrix} \cos(\kappa) & 0 & \sin(\kappa) \\ 0 & 1 & 0 \\ -\sin(\kappa) & 0 & \cos(\kappa) \end{bmatrix}$$

Remind: applying coordinate rotation φ is equal to applying rotation $-\varphi$ to objects.

Rotation in homogeneous coordinates

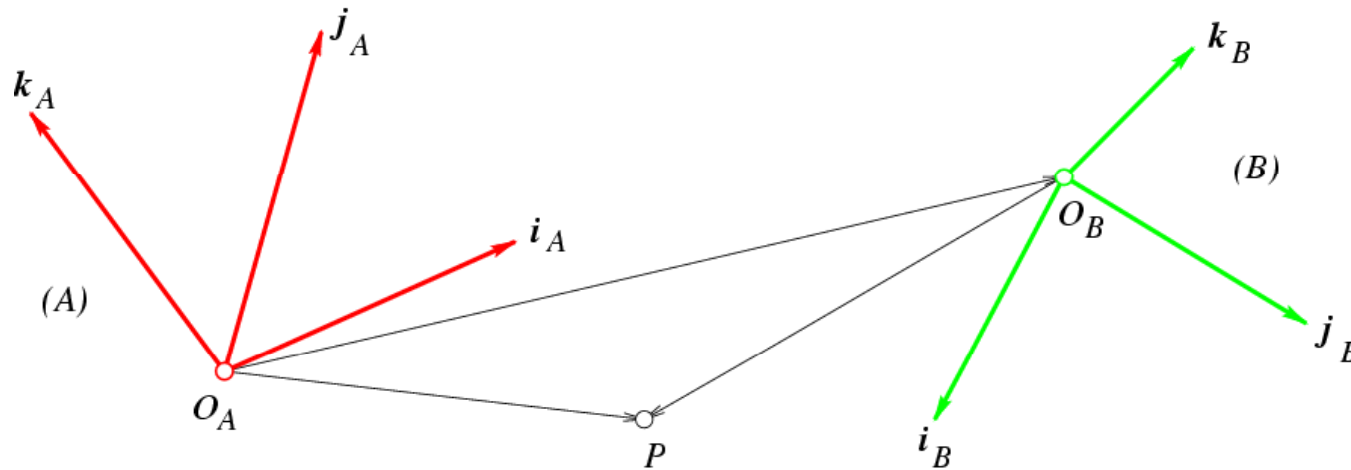
- Using homogeneous coordinates, rotation can be expressed as a matrix multiplication.

$${}^B P = {}^B_A R {}^A P$$

$$\begin{bmatrix} {}^B P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^B_A R & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^A P \\ 1 \end{bmatrix}$$

- Rotation is not commutative

Rigid transformations



$${}^B P = {}^B R^A P + {}^B O_A$$

Rigid transformations (cont.)

- Unified treatment using homogeneous coordinates.

$$\begin{aligned} \begin{bmatrix} {}^B P \\ 1 \end{bmatrix} &= \begin{bmatrix} 1 & {}^B O_A \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^B R & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^A P \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} {}^B R & {}^B O_A \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} {}^A P \\ 1 \end{bmatrix} \end{aligned}$$



$$\begin{bmatrix} {}^B P \\ 1 \end{bmatrix} = {}^B T_A \begin{bmatrix} {}^A P \\ 1 \end{bmatrix}$$

Invertible!



Perspective camera model

- Linear transformation of perspective projection coordinate.

$$p = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = [I \quad 0]P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

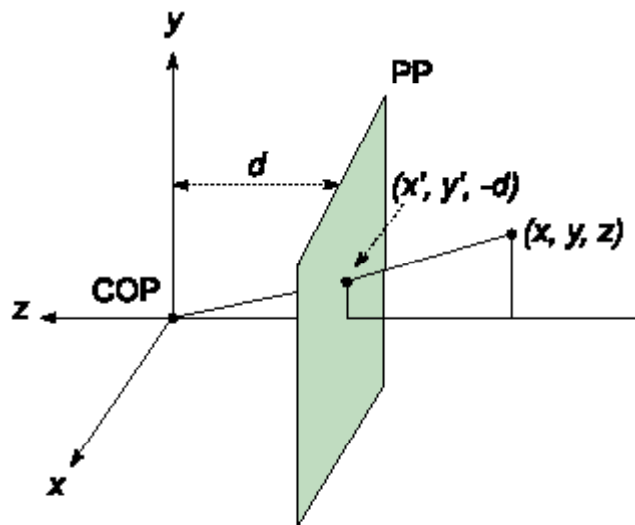
- Recover image (normalized) coordinate by projection.

$$\hat{u} = \frac{u}{w} = \frac{X}{Z}$$

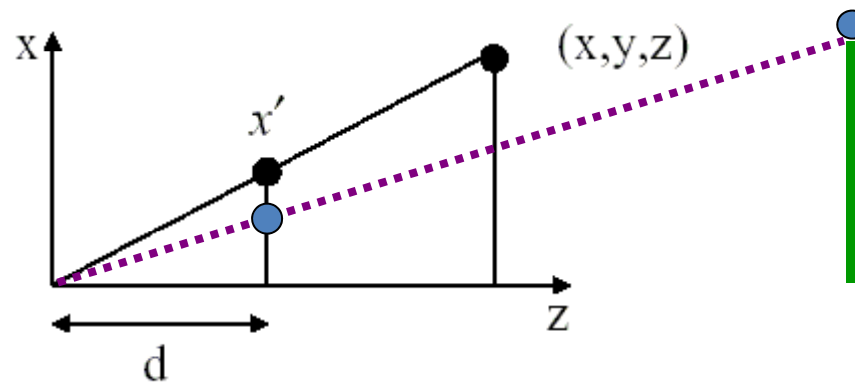
$$\hat{v} = \frac{v}{w} = \frac{Y}{Z}$$

Perspective projection

- Recall perspective projection



Using similar triangles gives:



http://commons.wikimedia.org/wiki/File:Taiwan_HighSpeedRail_Train_Business_Class_Car.JPG

Affine camera model

Pretend depth is constant (often OK !)

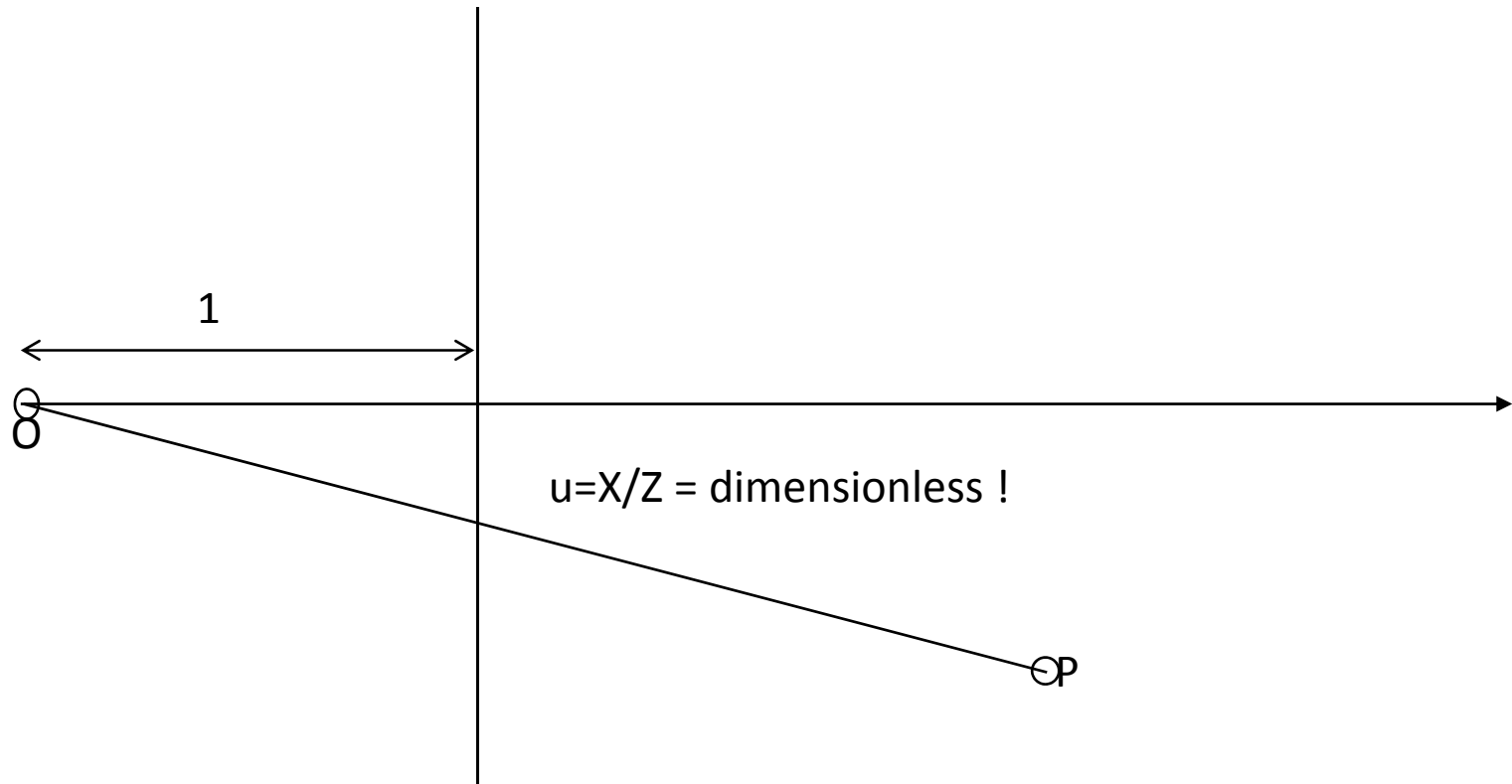
$$\hat{u} = \frac{X}{Z_r}$$

$$\hat{v} = \frac{Y}{Z_r}$$

Can also be written as a linear transformation :

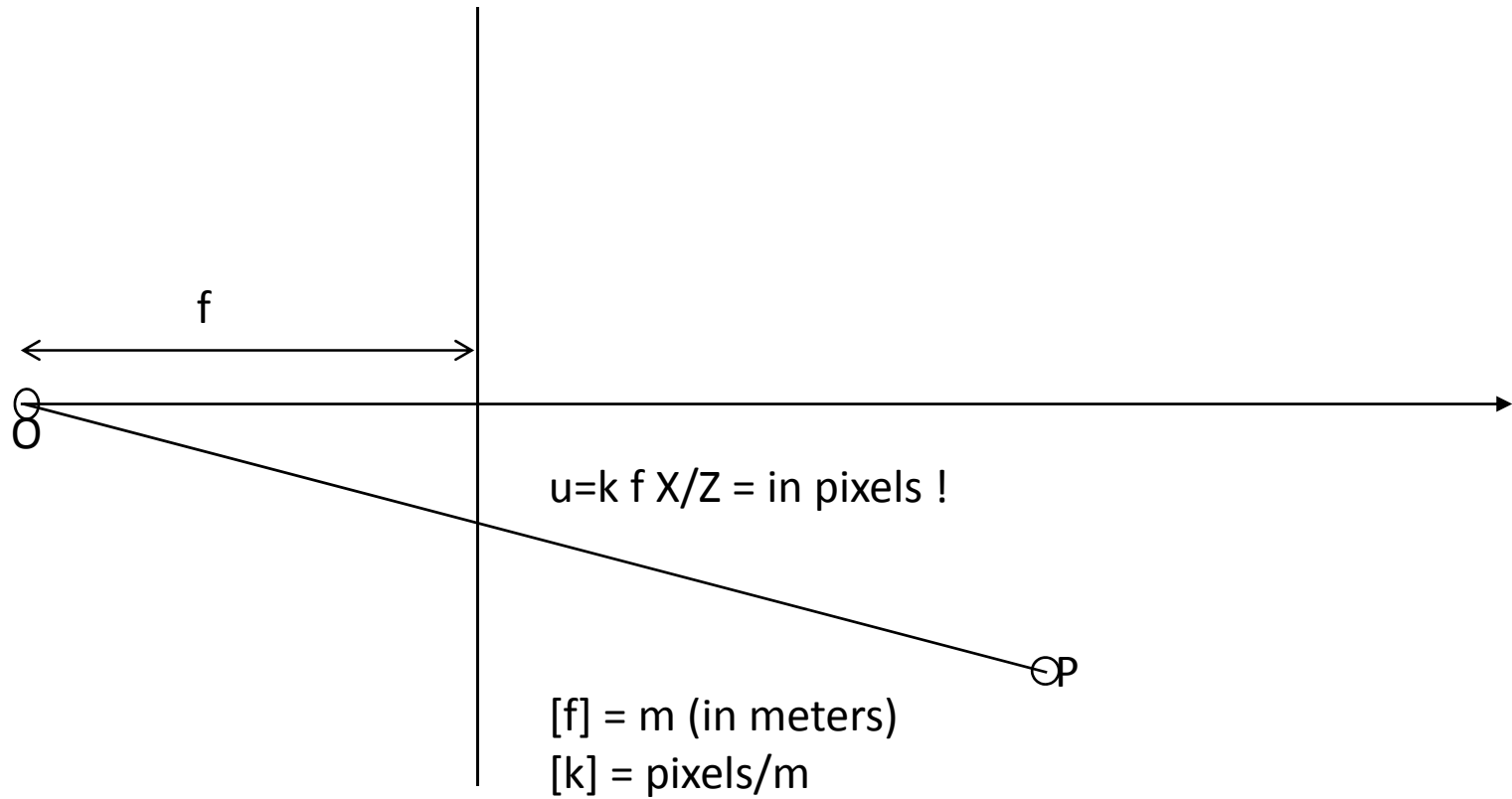
$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \frac{1}{Z_r} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & Z_r \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Normalized Image coordinates



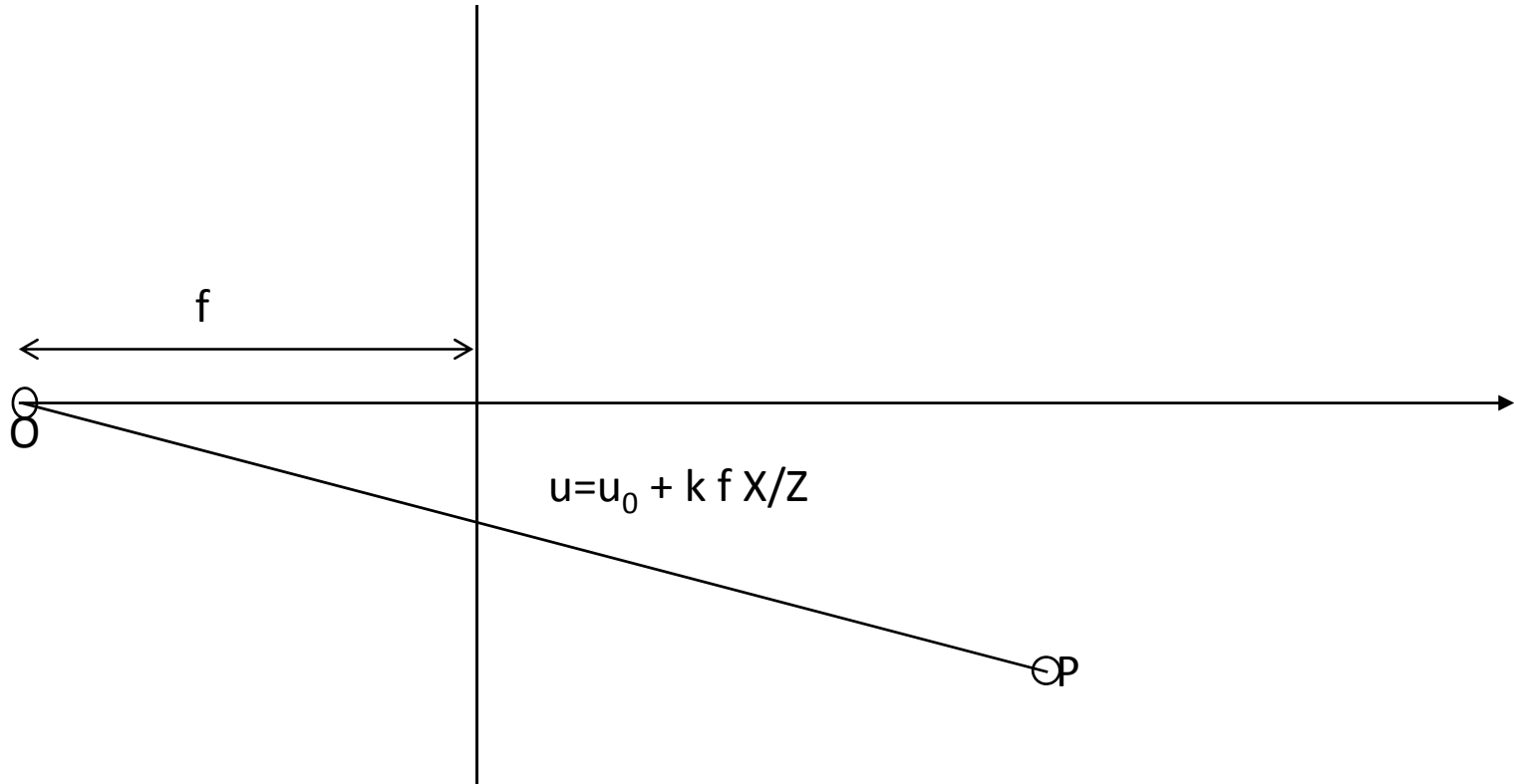
Pixel units

Pixels are on a grid of a certain dimension

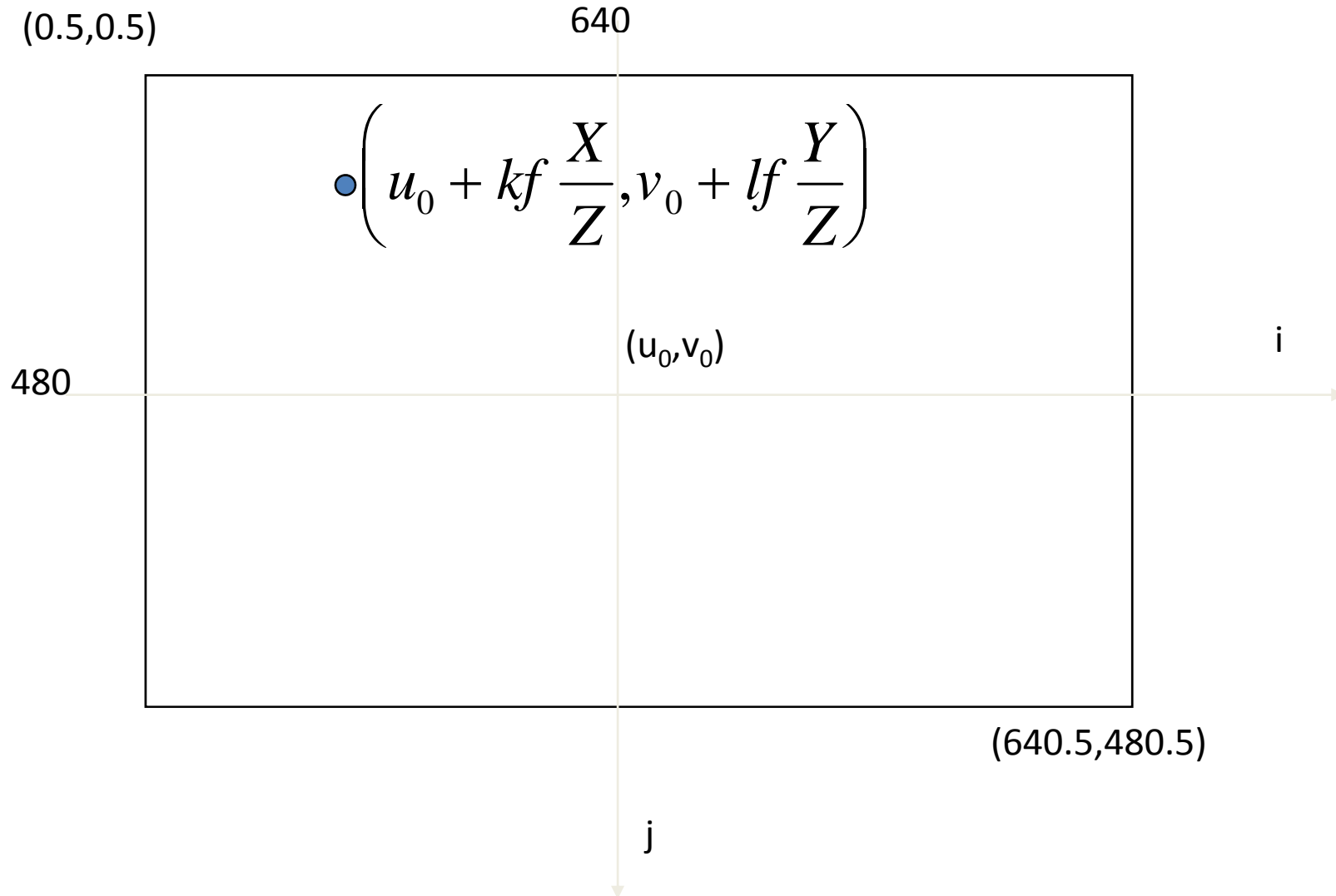


Pixel coordinates

We put the pixel coordinate origin on topleft



Pixel coordinates in 2D



Intrinsic parameters (in references)

3×3 Calibration Matrix K

$$p = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = K \begin{bmatrix} I & 0 \end{bmatrix} P = \begin{bmatrix} \alpha & s & u_0 \\ & \beta & v_0 \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Recover image (Euclidean) coordinates by normalizing :

$$\hat{u} = \frac{u}{w} = \frac{\alpha X + sY}{Z} + u_0$$

$$\hat{v} = \frac{v}{w} = \frac{\beta Y}{Z} + v_0$$

skew

5 Degrees of Freedom !

Intrinsic parameters (in the textbook)

3×3 Calibration Matrix K

$$p = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z} K [I \quad 0] P = \begin{bmatrix} \alpha & -\alpha \cot \theta & u_0 \\ & \frac{\beta}{\sin \theta} & v_0 \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

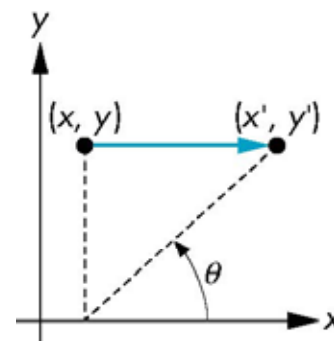
Recover image (Euclidean) coordinates by normalizing :

$$u = \frac{\alpha x - \alpha \cot \theta y}{z} + u_0$$

$$v = \frac{\beta y}{z \sin \theta} + v_0$$

$$\mathbf{H}(\theta) = \begin{bmatrix} 1 & \cot \theta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Shear matrix



Combining intrinsic and extrinsic param.

- Perspective projection mapping (including intrinsic and extrinsic parameters).

$$p = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z} K \begin{bmatrix} I & 0 \end{bmatrix} T P = \begin{bmatrix} \alpha & -\alpha \cot \theta & u_0 \\ & \frac{\beta}{\sin \theta} & v_0 \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^c_w R \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$= \frac{1}{z} K \begin{bmatrix} {}^c_w R & {}^c_w O \end{bmatrix} P = \frac{1}{z} M P$$

5+6 DOF = 11 !