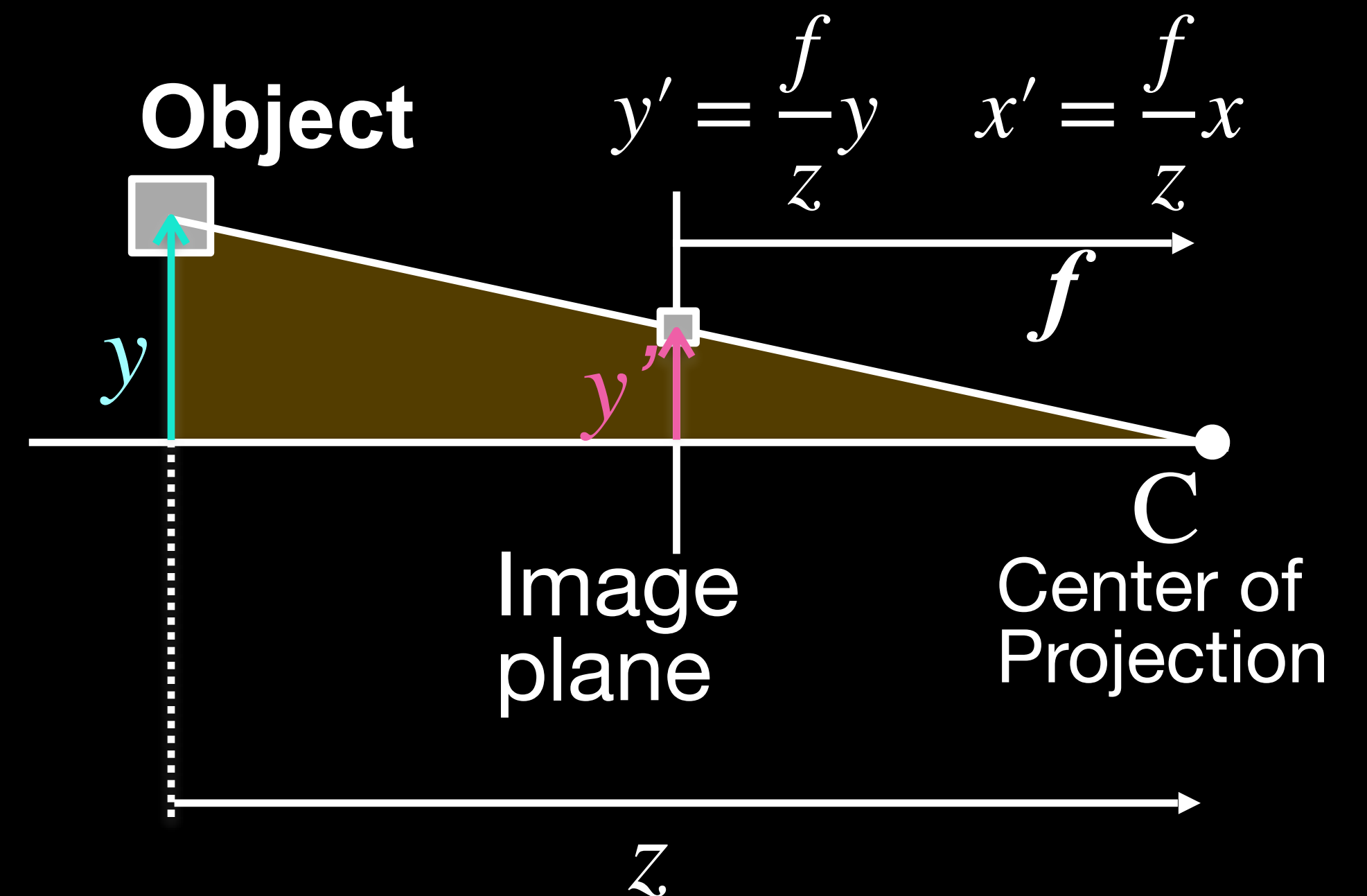
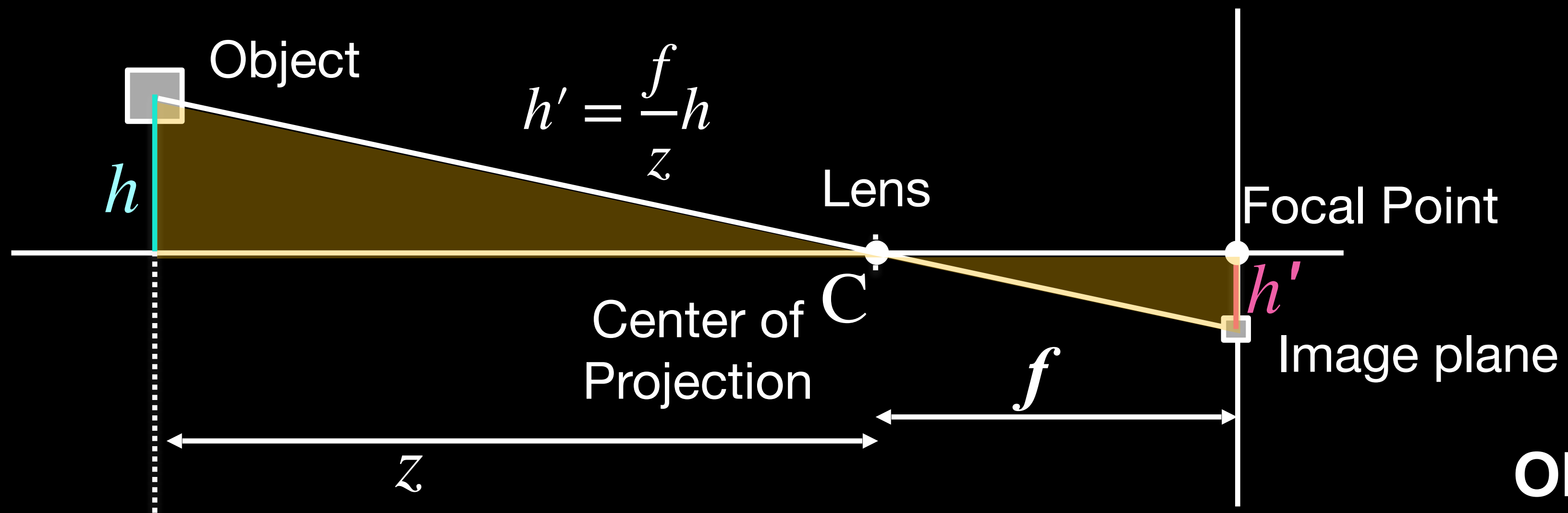
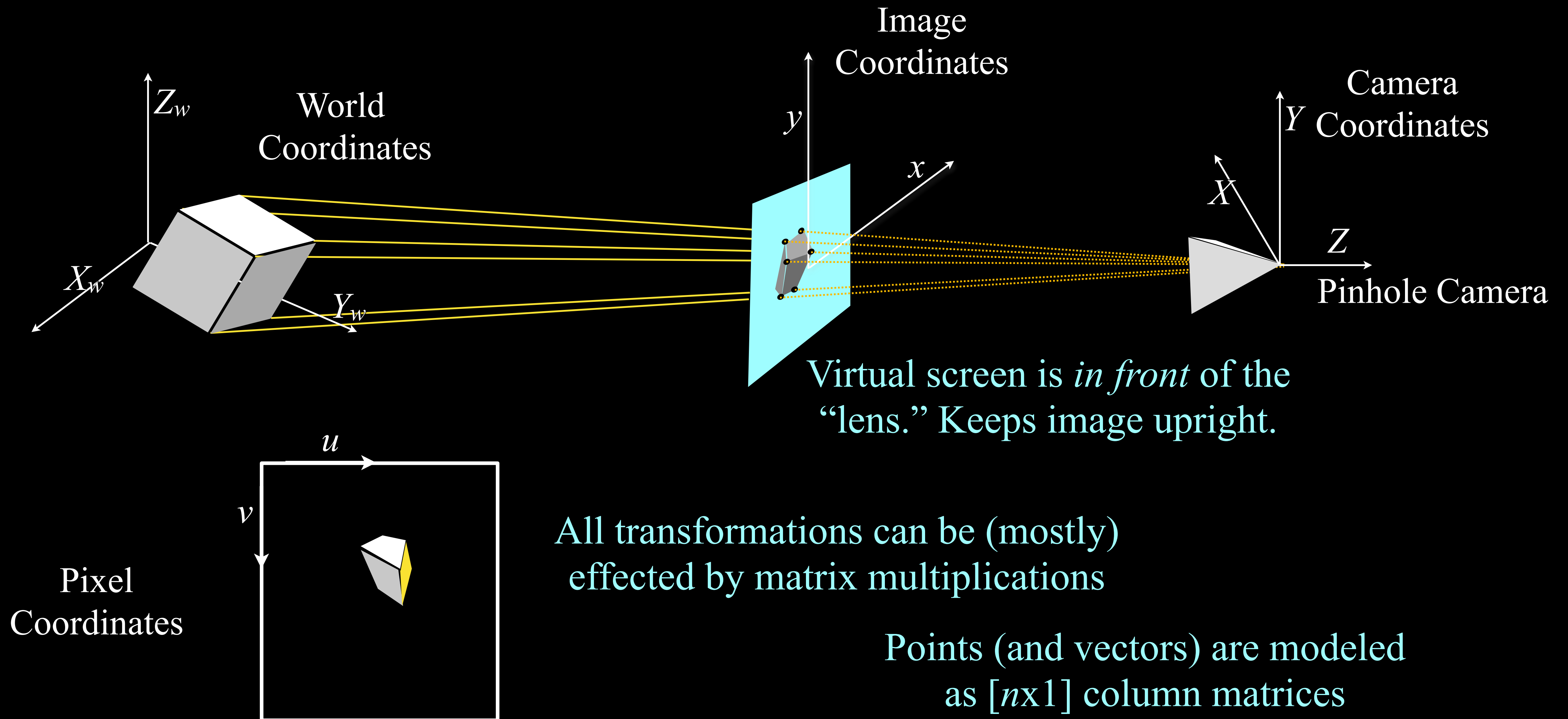


Pinhole Camera Approximation



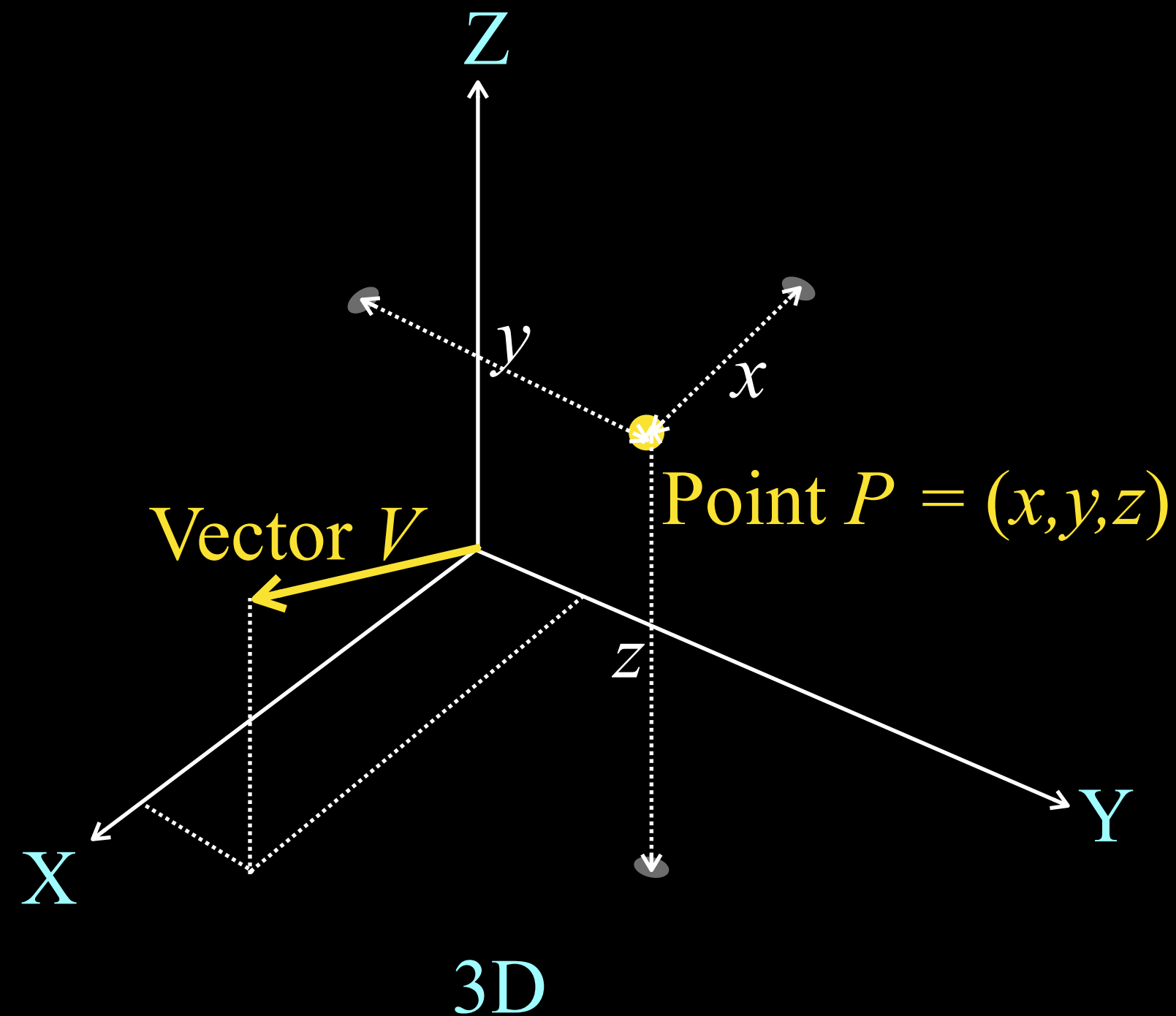
Multiple Coordinate Spaces



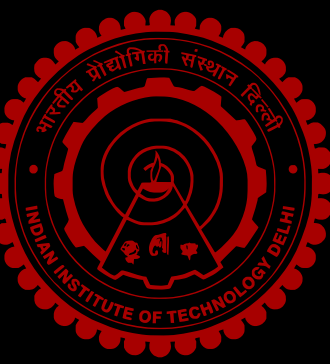
Points and Vectors

Coordinate Geometry and Affine Space

- Coordinate axes are directions: n axes for n dimensional space
 - Orthonormal mutually perpendicular, unit length axes



- Vectors/points are represented by n coordinate values
 - Also as $n \times 1$ (column) matrix
- A vector V has no position, it indicates a pure direction
 - Thus, starting at any point P , one can move in direction V to reach point $P+V$ (coordinate-wise addition)
 - Similarly, given points P and Q , $(P-Q)$ is a vector
- Define Euclidean metrics:
 - Distance, Magnitude, Angle, Inner and Outer products



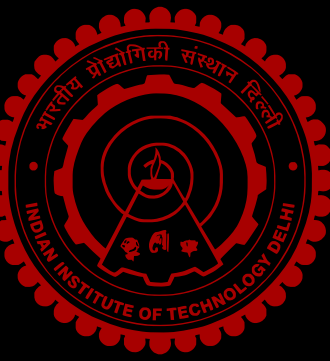
Essential Transformations

In n -dimensional Euclidean space

We can represent transformations
as matrix (pre)multiplication

$$\begin{bmatrix} m & a & t \\ r & i & x \end{bmatrix} \begin{bmatrix} p \\ n \\ t \end{bmatrix}$$

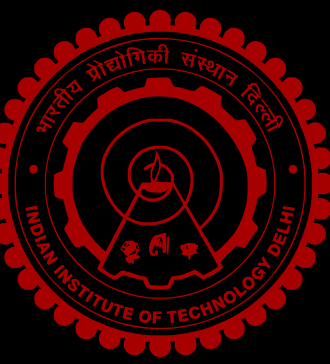
- **Rigid-body:** move around (no change to shape or size, no length changes)
 - Translate, Rotate
- **Similarity:** No change to shape, but allows change of size (No change to ratios of lengths)
 - Translate, Rotate, Reflect, Uniformly scale
- **Linear:** $a \mathcal{T}(P) + b \mathcal{T}(Q) = \mathcal{T}(aP+bQ)$, scalar a, b
 - Rotate, Scale, Reflect, Shear (parallel lines remain parallel, no change to ratio of lengths on parallel lines)
 - Can be formulated as $n \times n$ matrix-multiplication, n = number of coordinate dimensions
- **Affine:** Linear + add: $a\mathcal{T}(P) + b\mathcal{T}(Q) = \mathcal{T}(aP+bQ) + K$, K is a vector with same dimension as P, Q
 - Preserves parallel lines and ratios of lengths along parallel lines
 - Can be formulated as matrix-multiplication in a higher dimension: $n+1$



Projective Transformation

Projective transformations involve matrix multiplication and homogenizing/dehomogenizing

- An n -dimensional projective space is given by $\mathbb{R}^{n+1}$, where
 - 0_{n+1} is undefined and $P = wP$
 - The additional coordinate is called **homogeneous** coordinate
- Not a vector space (no addition, no origin)
- Convert Affine $\rightarrow$ Projective: add a new coordinate value = 1
- Convert Projective $\rightarrow$ Affine: divide by homogeneous coordinate, and remove 1
- **Projective transformation** can represent projection
 - Lines transform to lines, parallelism and ratios can be lost (preserves cross-ratio)



Camera Projection Pipeline

$$\begin{bmatrix} u \\ v \end{bmatrix} = [M_{Image \rightarrow Pixel}] [M_{Cam \rightarrow Image}] [M_{World \rightarrow Cam}] \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$$

A sequence of transformations =
A sequence of matrix multiplications
← (Right to left)

Adding Homogeneous
Coordinates:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$= [M_{Image \rightarrow Pixel}] [M_{Cam \rightarrow Image}] [M_{World \rightarrow Cam}]$$

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Homogenize:
add 1

(Supports Affine and
Projective Transformations)

In Projective Space:

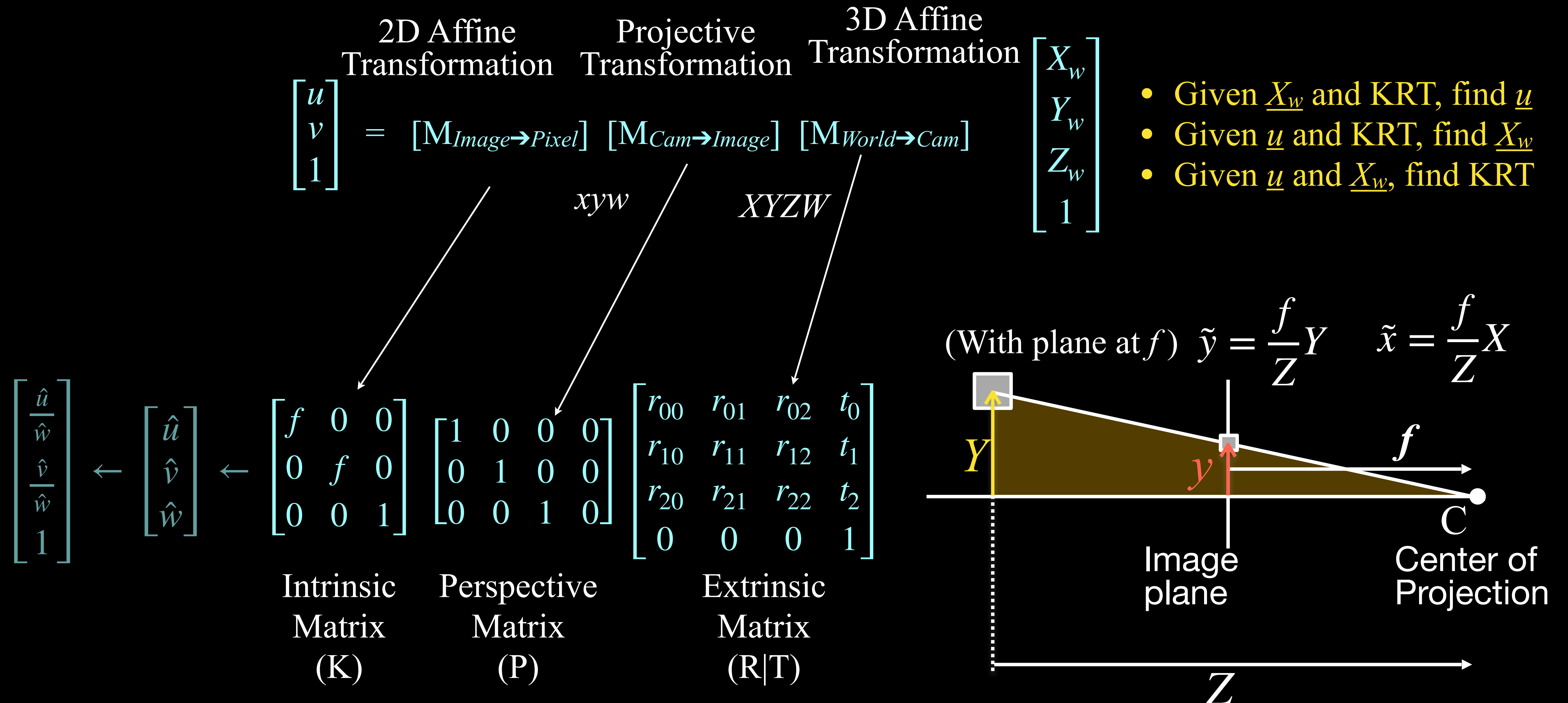
$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} \frac{x}{w} \\ \frac{y}{w} \\ \frac{z}{w} \\ 1 \end{bmatrix}$$

Similarly,

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \frac{u}{w} \\ \frac{v}{w} \\ 1 \end{bmatrix}$$

Dehomogenize:
Divide, then remove 1

Camera Projection Pipeline



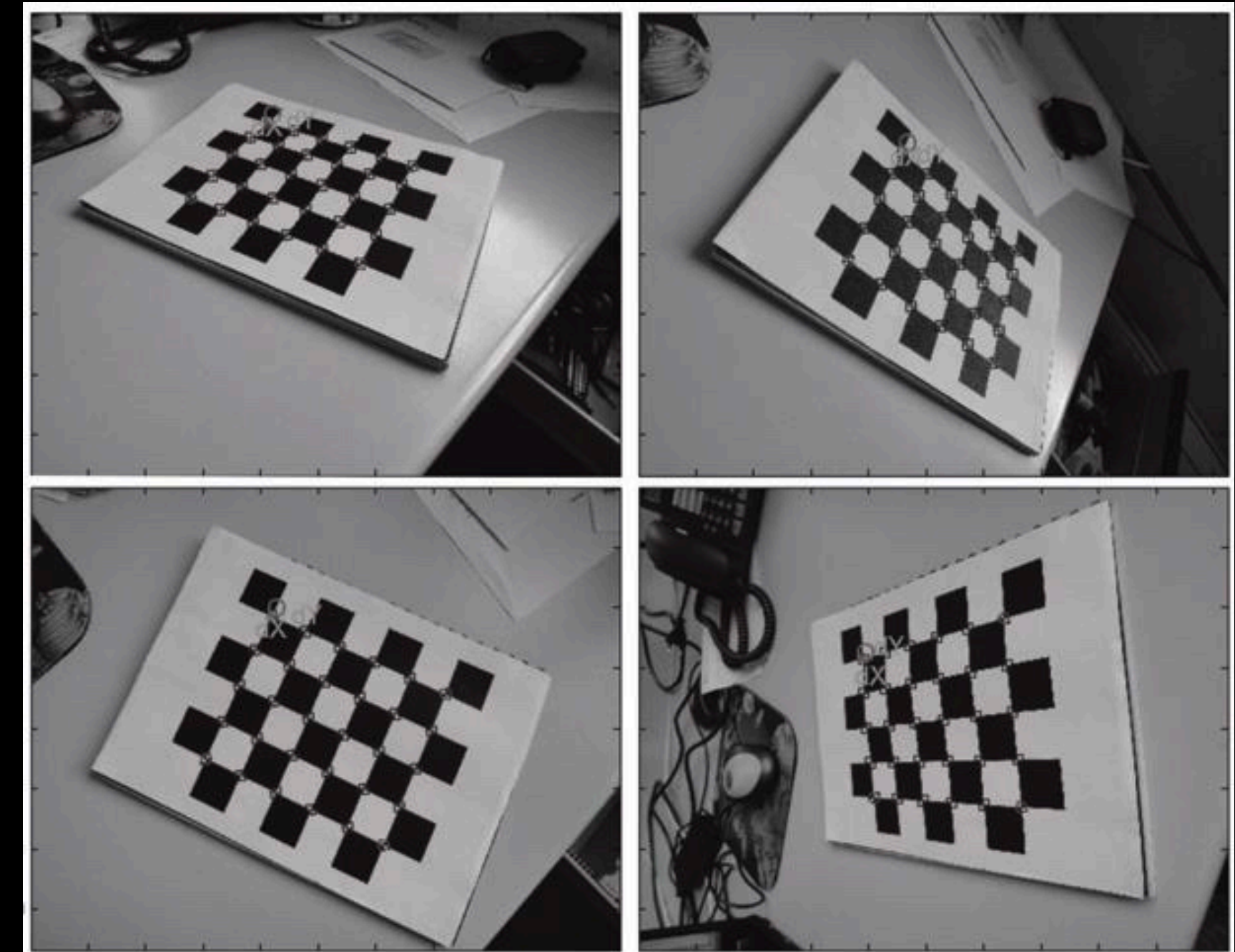
Zhang's Camera Calibration

1. Precisely measure the 3D location of points on a single flat image
2. Image the grid from ~30 camera distances, directions, and orientations
3. Detect corners in images with sub-pixel accuracy
4. Compute *composite* matrix for each camera position
5. Starting with an initial guess for K matrix, refine it using Levenberg-Marquardt bundle adjustment

$$\sum_{i=0}^{n-1} \sum_{c=0}^{m-1} \|p_{ic} - f(K, R_i, T_i, P_c)\|^2$$

Allows a general projection function including non-linear lens distortion

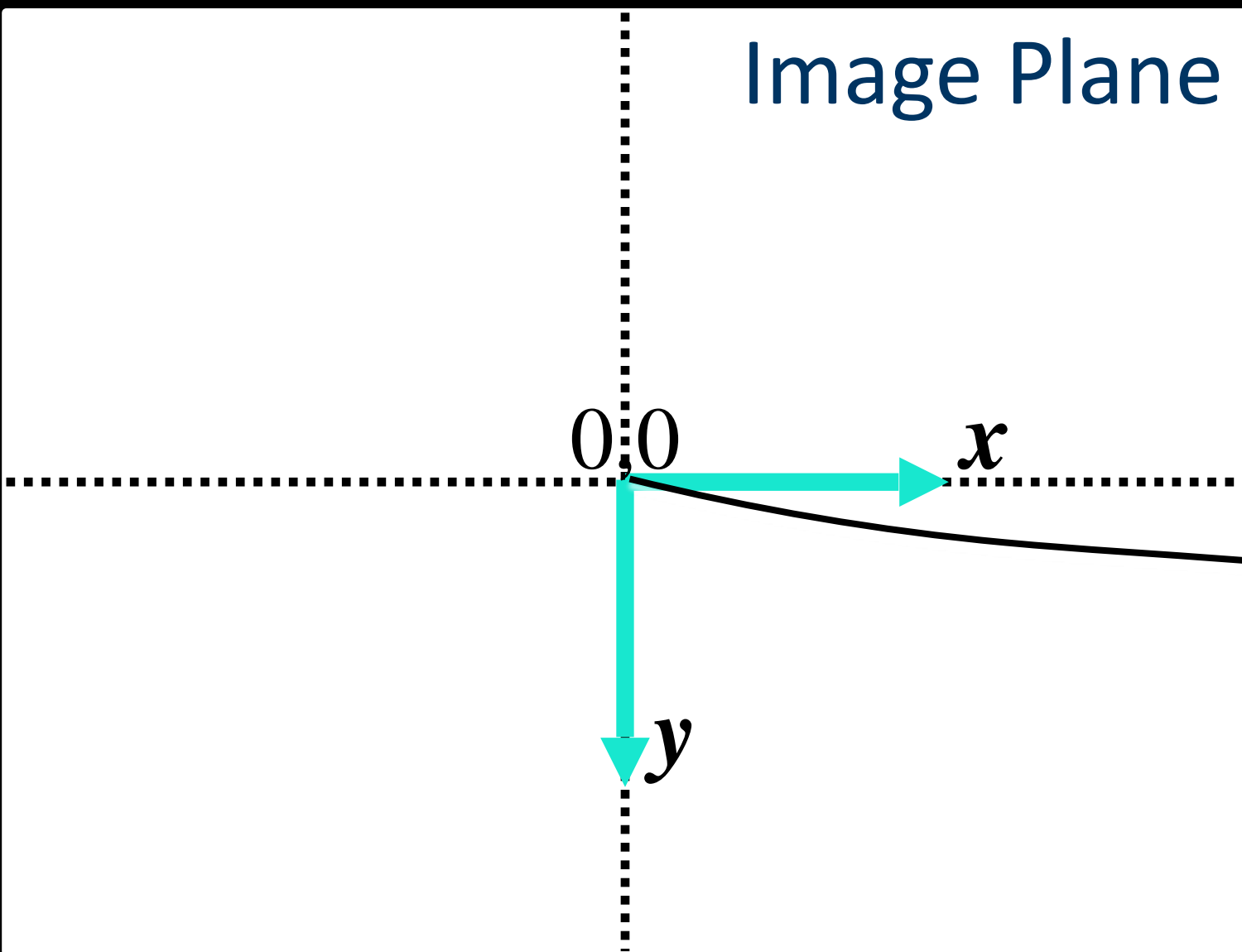
For each image i and corner point c



Intrinsic Matrix

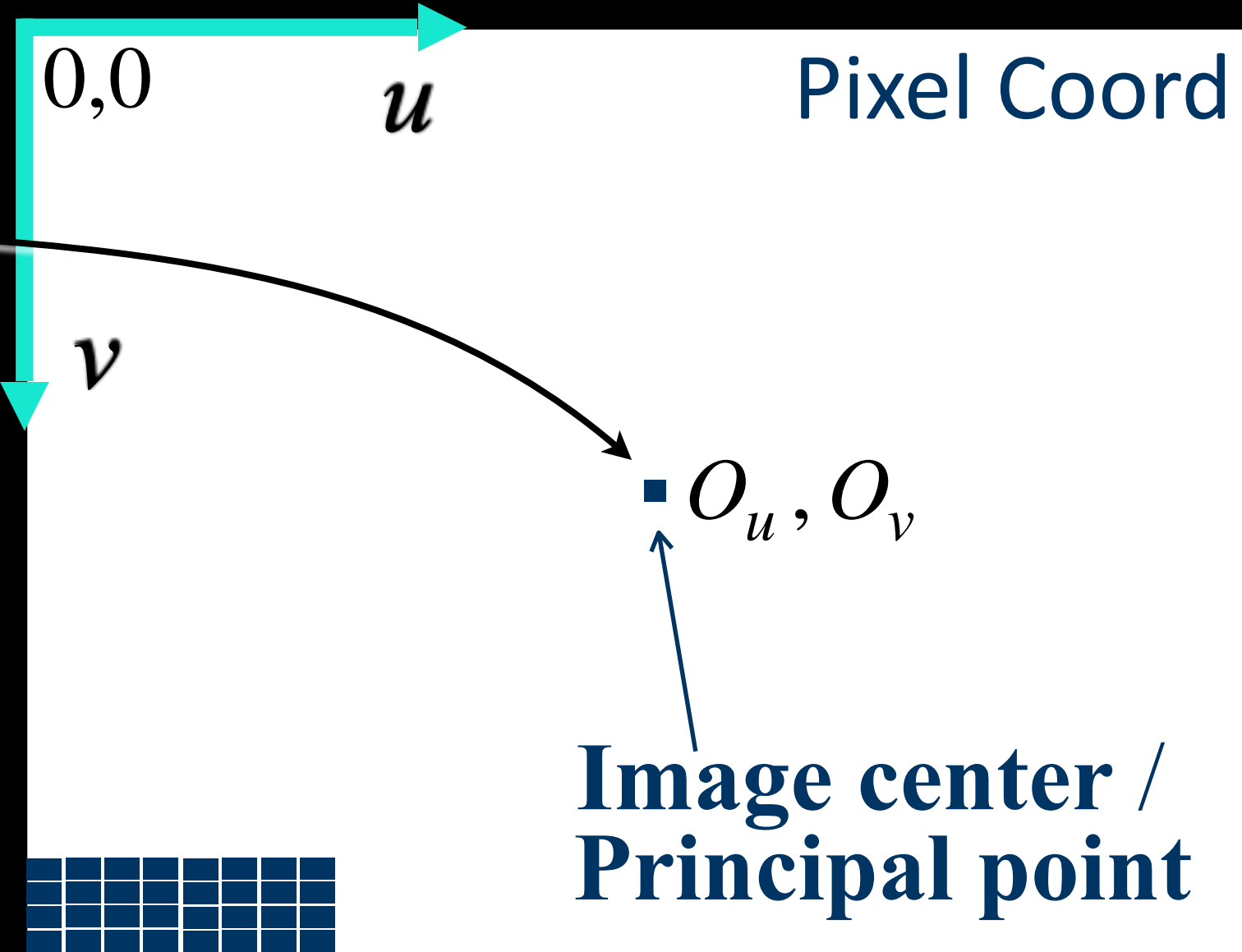
Internal camera parameters

- Internal camera parameters: normalized image plane to pixel array



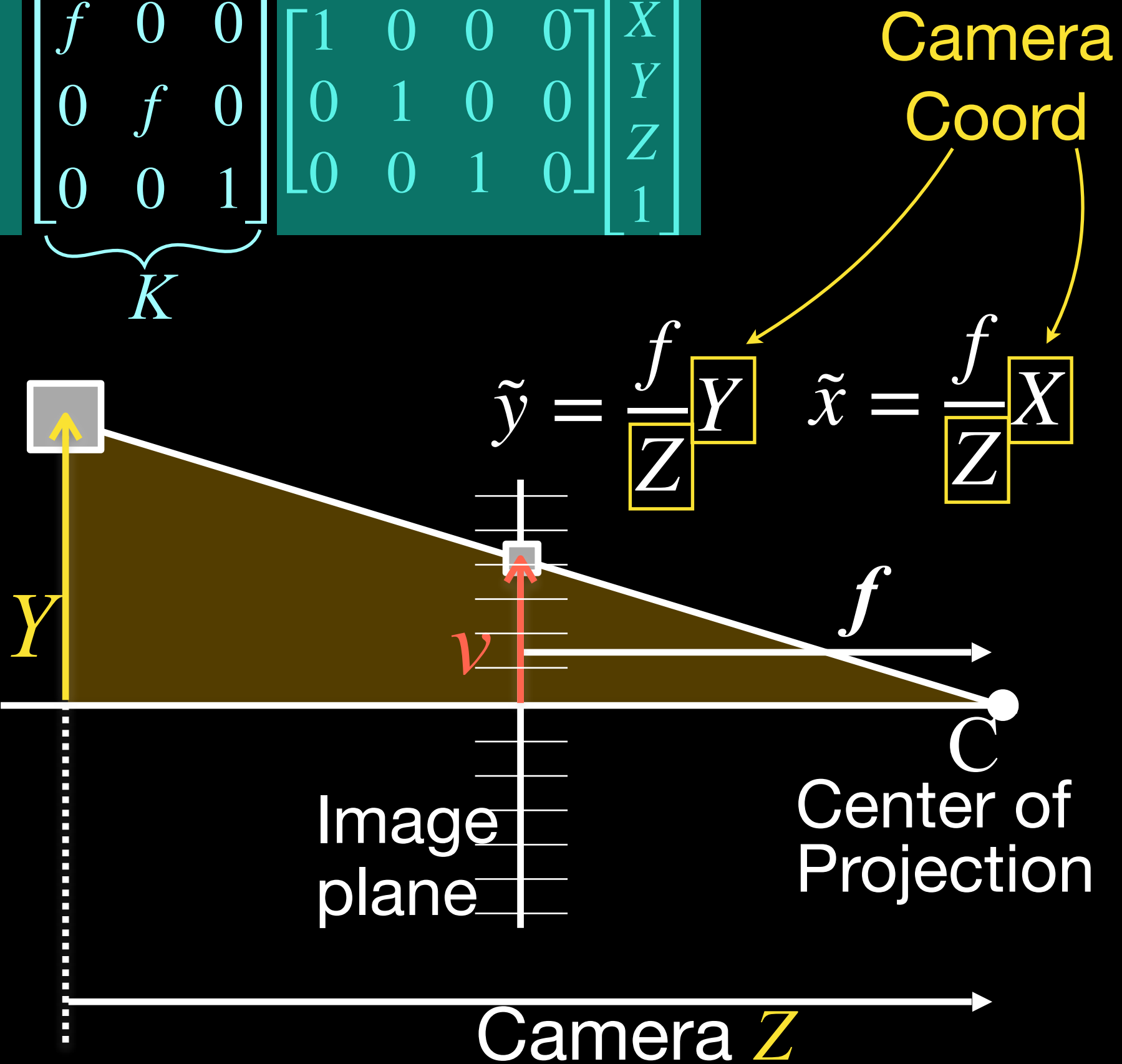
$$K = \begin{bmatrix} fs_u & 0 & O_u \\ 0 & fs_v & O_v \\ 0 & 0 & 1 \end{bmatrix}$$

s_u, s_v are pixel densities, f is focal length



Pixel density in u and v may differ

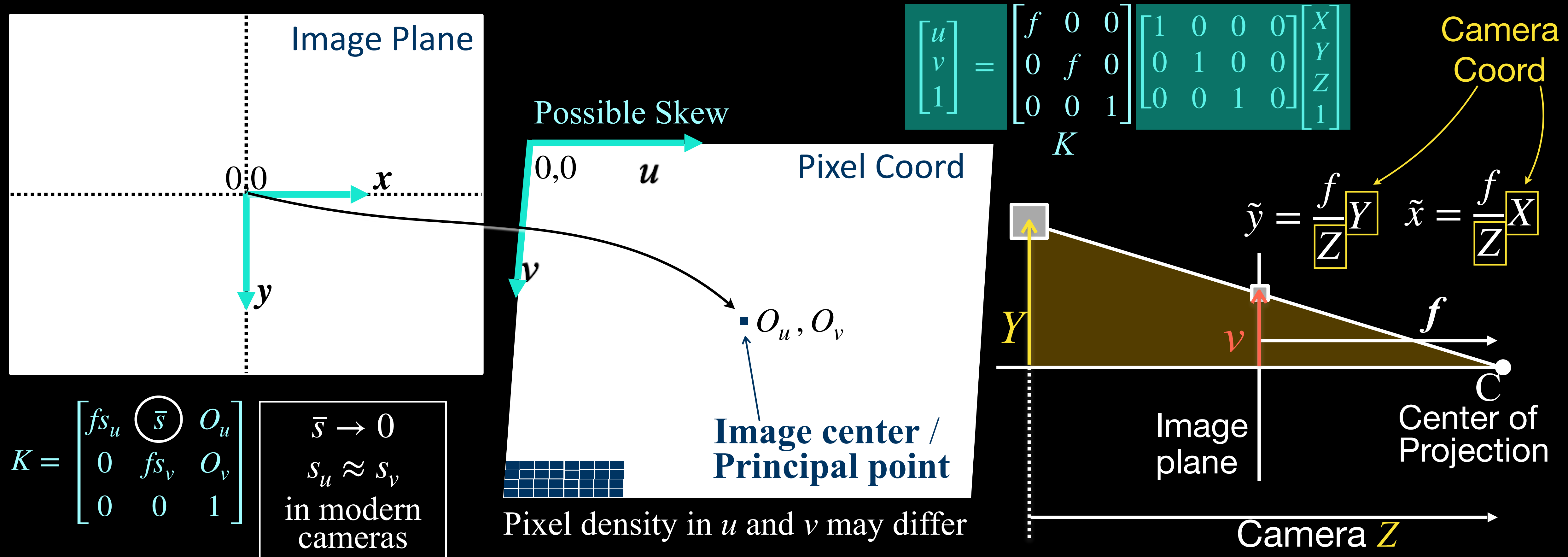
$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}}_K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$



Intrinsic Matrix

Internal camera parameters

- Internal camera parameters: normalized image plane to pixel array



Intrinsic Matrix

Internal camera parameters

- Internal camera parameters: normalized image plane to pixel array

