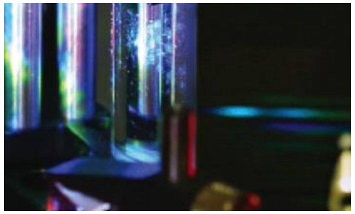
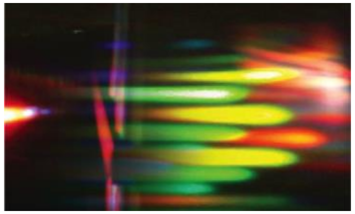




COMP70058 Computer Vision

Lecture 13 – Camera Geometry

Stamatia (Matina) Giannarou, PhD
The Hamlyn Centre for Robotic Surgery

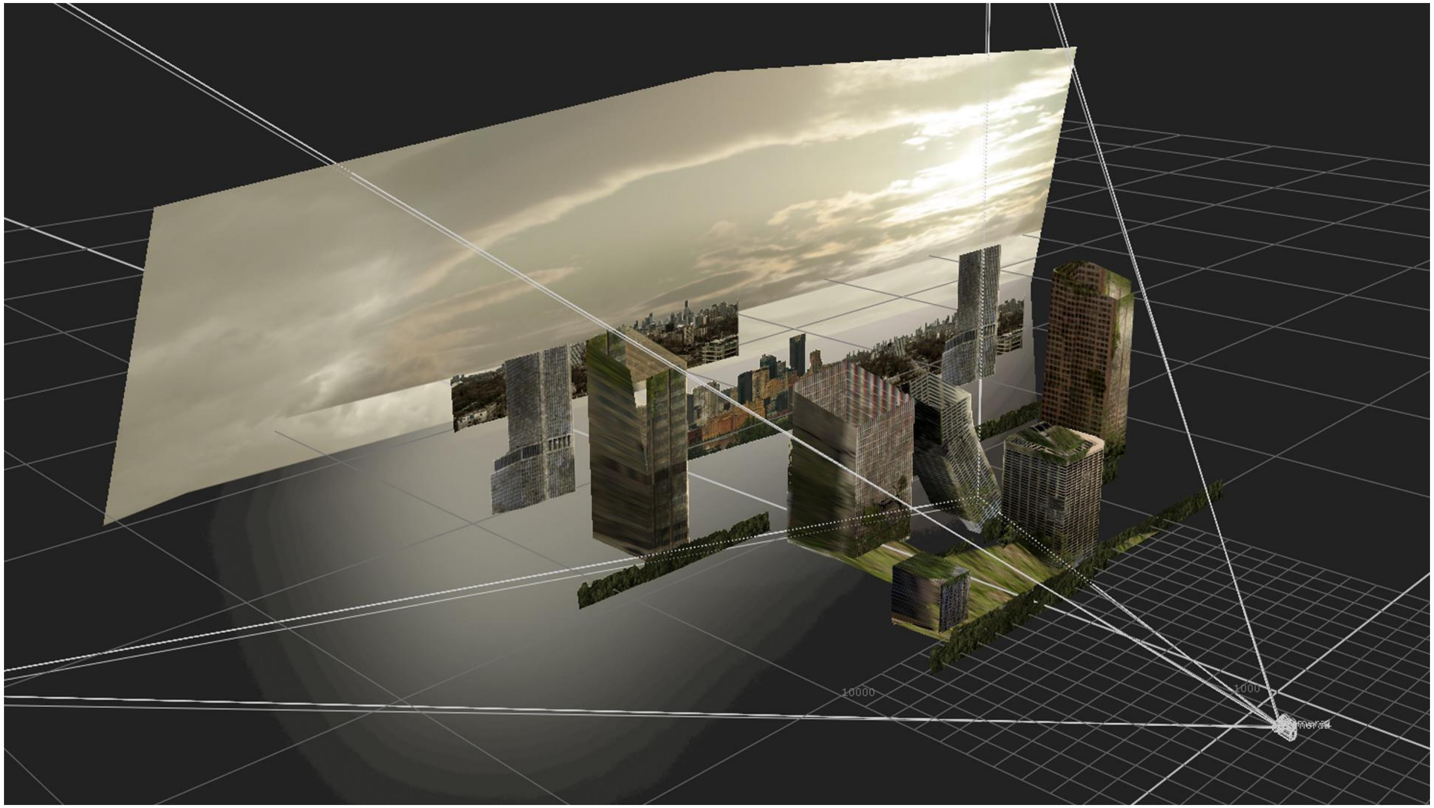
stamatia.giannarou@imperial.ac.uk

The Hamlyn Centre
for Robotic Surgery

Contents

- 2D planar transformations
- Camera geometry and transformations
- Camera projection matrix
- Camera projection models:
  - Perspective
  - Orthographic
  - Weak-perspective
- Radial distortion

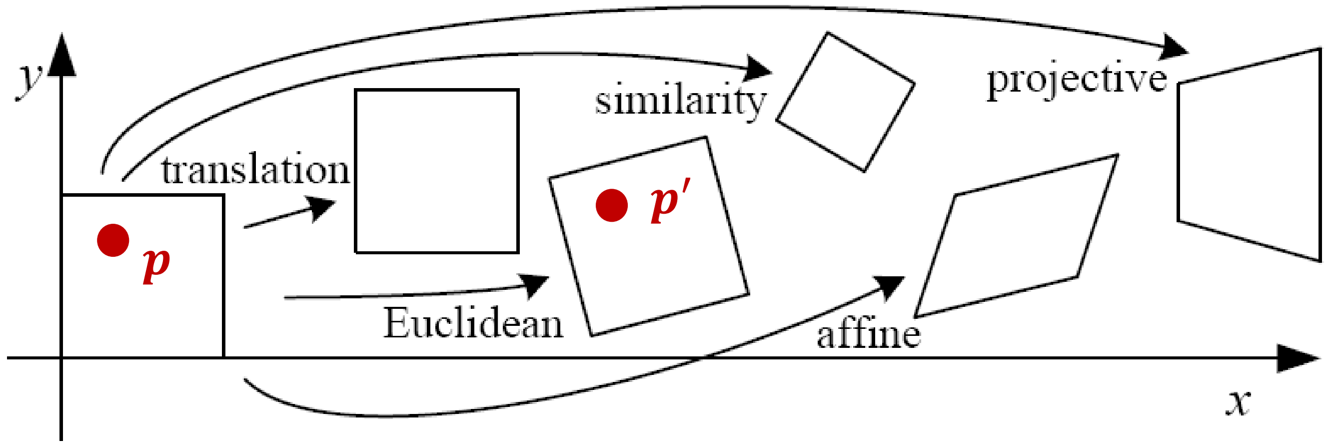
Camera models



- A camera is a mapping between the 3D world and a 2D image which is represented by a projection matrix.

- The main projection models are:
  - ➢ Perspective projection
  - ➢ Orthographic projections

2D planar transformations



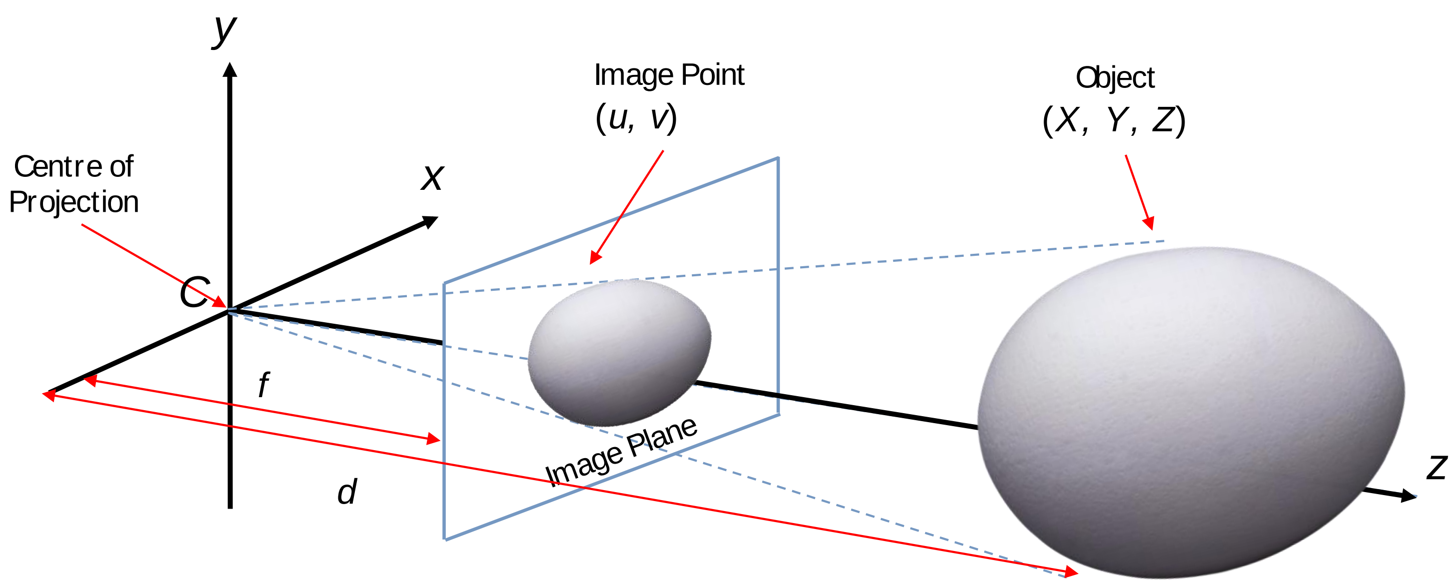
Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\begin{bmatrix} I \mid t \end{bmatrix}_{2\times 3}$	2	orientation + ⋯	
rigid (Euclidean)	$\begin{bmatrix} R \mid t \end{bmatrix}_{2\times 3}$	3	lengths + ⋯	
similarity	$\begin{bmatrix} sR \mid t \end{bmatrix}_{2\times 3}$	4	angles + ⋯	
affine	$\begin{bmatrix} A \end{bmatrix}_{2\times 3}$	6	parallelism + ⋯	
projective	$\begin{bmatrix} \tilde{H} \end{bmatrix}_{3\times 3}$	8	straight lines	

$$p' = T(p)$$

If the transformation T is the same for any point on the image, then T is a global transformation.

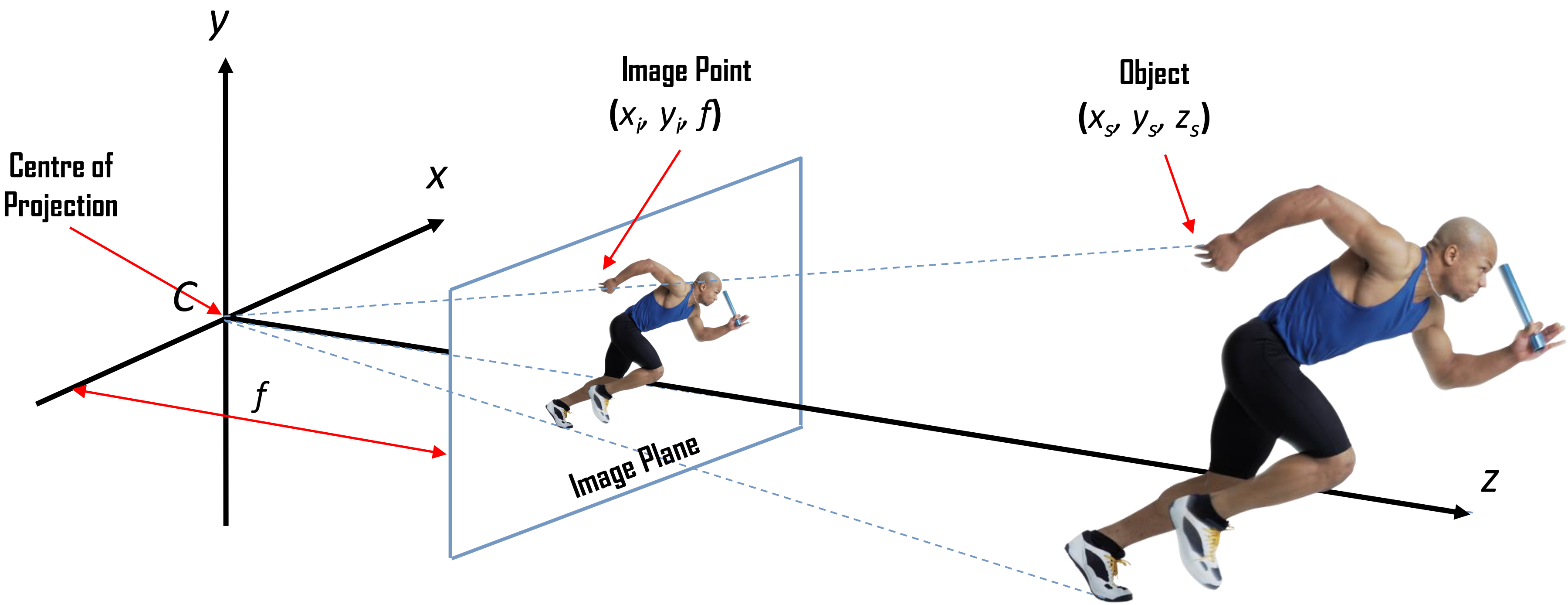
https://szeliski.org/Book/

Pinhole camera model



- The most specialized and simplest camera model is the pinhole camera.
- Under the pinhole camera model, a point in space with coordinates (X, Y, Z) is mapped to the point on the image plane where a line joining the point X to the centre of projection meets the image plane.

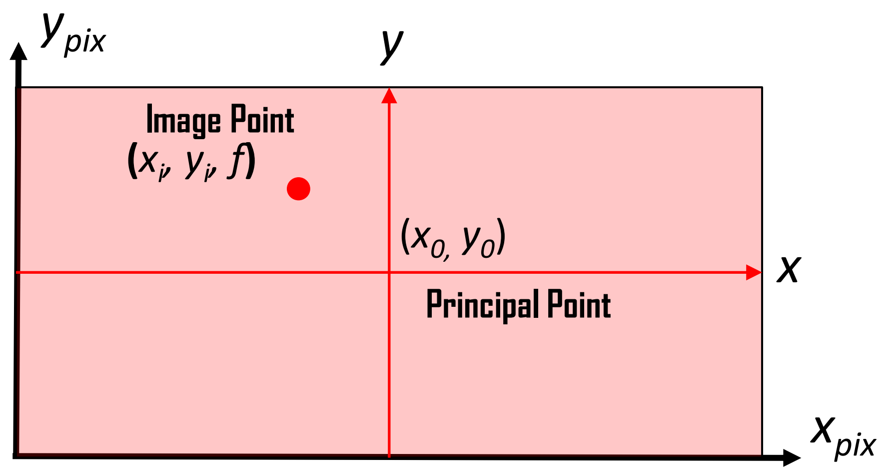
Camera Geometry



$$x_i = f\frac{x_s}{z_s}, \quad y_i = f\frac{y_s}{z_s}$$

Pixel Coordinates

- The origin of the pixel coordinates is typically located at a corner of the sensor array (e.g., top/bottom-left)
- CCD/CMOS pixels may not be square (due to unequal spacing in the horizontal/vertical directions
- There may also be skew factor and lens distortion (pin-cushion effect), all contributing to projection of the 3D object to different pixel coordinates
- The use of normal coordinates is quite awkward, let's try homogeneous coordinates instead



$$x_i = f\frac{x_s}{z_s}, \quad x_{pix} = k_x x_i + x_0 = k_x f \frac{x_s}{z_s} + x_0$$

$$y_i = f\frac{y_s}{z_s}, \quad y_{pix} = k_y y_i + y_0 = k_y f \frac{y_s}{z_s} + y_0$$

$$\alpha_x = fk_x, \quad x_{pix} = \frac{\alpha_x x_s + z_s x_0}{z_s}$$

$$\alpha_y = fk_y, \quad y_{pix} = \frac{\alpha_y y_s + z_s y_0}{z_s}$$

Homogeneous Coordinates

- Homogeneous coordinates are typically used for projective geometry
- The advantage is that coordinates of points including those at infinity can be represented using finite coordinates
- For point (*x, y*) in the Euclidean plane, its representation in the projective plane is simply (*x, y, 1*) and for homogeneous coordinates, we have (*u, v, w*) =(*au, av, aw*) for *a≠0* and its equivalent in the Euclidean plane is (*u/w, v/w*)
- The use of homogeneous coordinates can simplify a lot of the projective representations.

$$x_i = f\frac{x_s}{z_s}$$

$$y_i = f\frac{y_s}{z_s}$$

Euclidean Coordinates

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix}$$

Homogeneous Coordinates

$$x_i = u / w$$

$$y_i = v / w$$

Back to Euclidean Coordinates

Homogeneous Coordinates

- So how about for general camera projection we mentioned earlier?

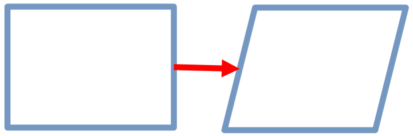
$$\alpha_x = fk_x, \quad x_{pix} = \frac{\alpha_x x_s + z_s x_0}{z_s}$$

$$\alpha_y = fk_y, \quad y_{pix} = \frac{\alpha_y y_s + z_s y_0}{z_s}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \alpha_x & 0 & x_0 & 0 \\ 0 & \alpha_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix}$$

Homogeneous Coordinates

- What if there is a skew distortion to the camera (e.g. horizontal skew)?



$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \alpha_x & s & x_0 & 0 \\ 0 & \alpha_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix}$$

How many Degrees of Freedom (DoF)?

Internal Coordinates

- Let's try to simplify the notation even further

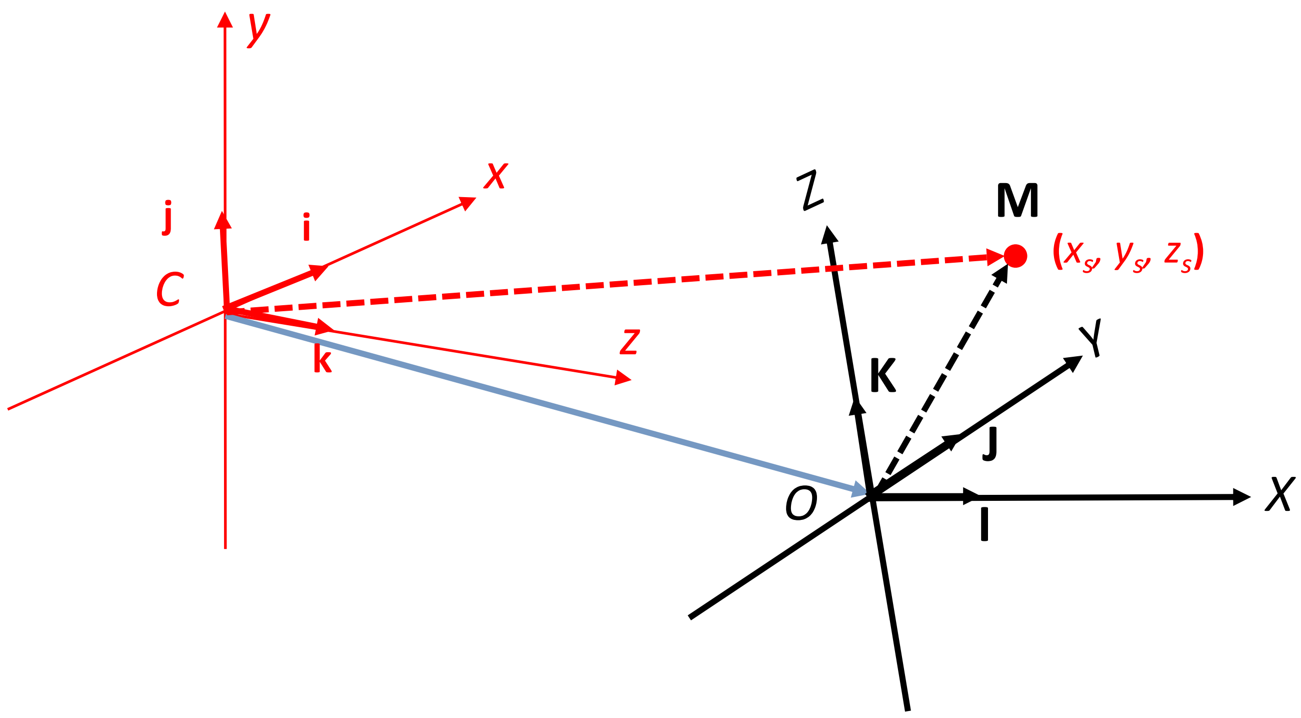
$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \alpha_x & s & x_0 & 0 \\ 0 & \alpha_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \alpha_x & s & x_0 & 0 \\ 0 & \alpha_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \underbrace{\begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \left[\underbrace{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}}_{\mathbf{I}} \; \underbrace{\begin{matrix} 0 \\ 0 \\ 0 \end{matrix}}_{\mathbf{0}} \right] = \mathbf{K}\left[\mathbf{I} \mid \mathbf{0}\right]$$

- **K** is called the calibration matrix (3x3 upper triangular matrix), representing the internal parameters of the camera and has five DoF.

Camera Transformation Matrix

- The use of homogeneous coordinates gives a convenient way of representing points in two coordinate systems
- For vision, it is necessary to consider projection of points to a camera with arbitrary position and orientation (pose) in 3D space
- We use unit vectors **I, J, K** to represent the world coordinate system and unit vectors **i, j, k** to represent the camera coordinate system



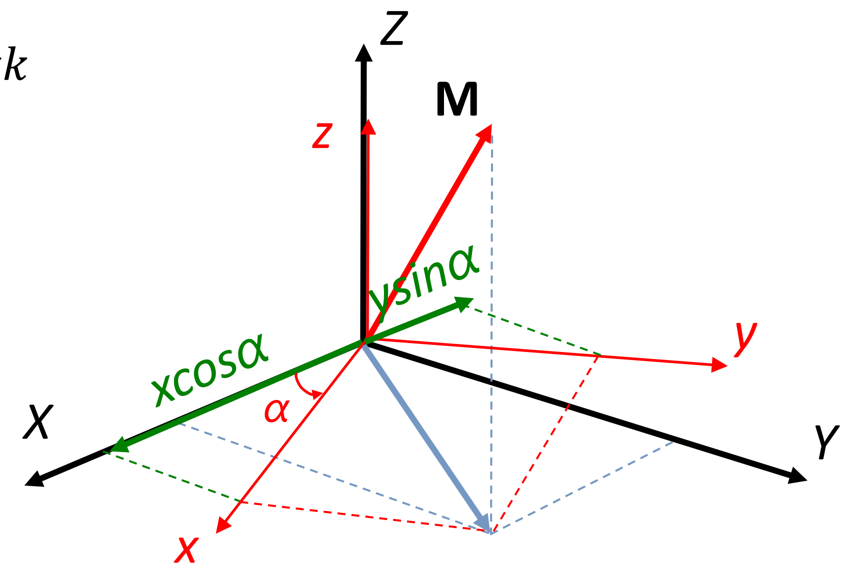
Use Inner Product for Rotation

- Our aim is to find the relationship that expresses how the components of a vector in one coordinate system relate to the components of the same vector in a different coordinate system.
- We can find the components of the vector M in the transformed system in term of the components of M in the original system by simply taking the dot product of the equation below with the desired unit vector in the transformed system.

$$M = XI + YJ + ZK = xi + yj + zk$$

$$I\cdot\ XI + I\cdot YJ + I\cdot ZK = I\cdot xi + I\cdot yj + I\cdot zk$$

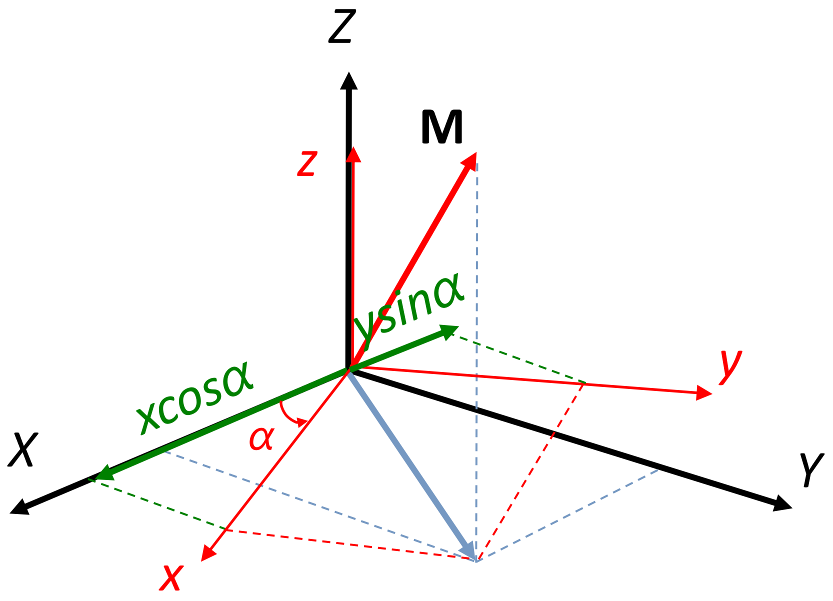
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} I\cdot xi + I\cdot yj + I\cdot zk \\ J\cdot xi + J\cdot yj + J\cdot zk \\ K\cdot xi + K\cdot yj + K\cdot zk \end{bmatrix}$$



Use Inner Product for Rotation

- We often use rotation matrices to represent coordinate transforms, this in fact can be more conveniently represented as the inner product of coordinate unit vectors, for example, when you rotate along the Z axis, we have

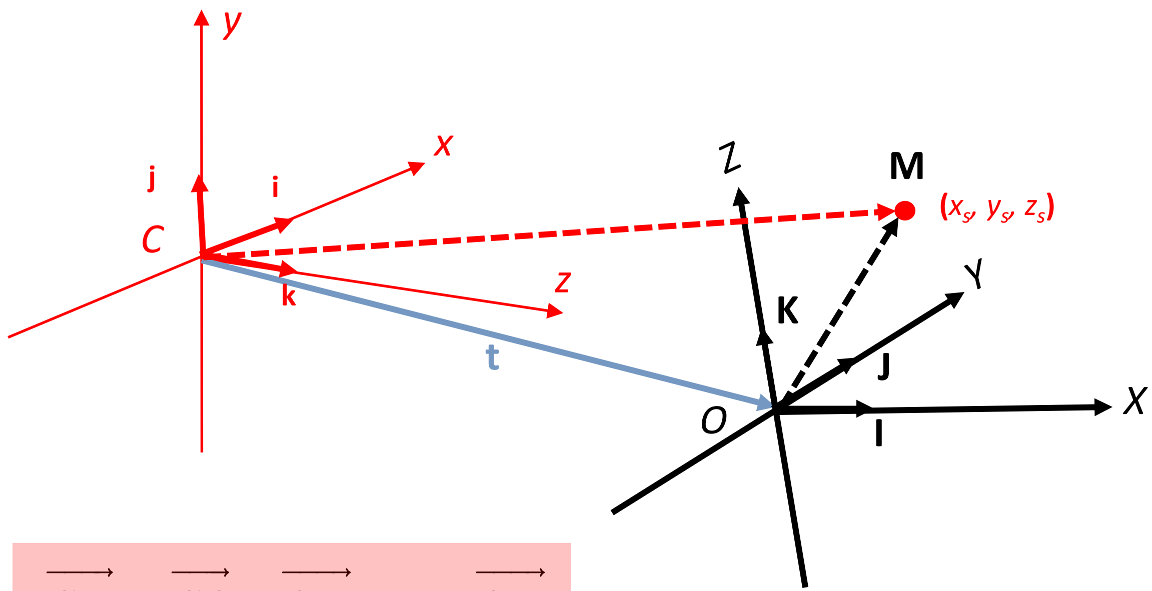
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \mathbf{I} \cdot x\mathbf{i} + \mathbf{I} \cdot y\mathbf{j} + \mathbf{I} \cdot z\mathbf{k} \\ \mathbf{J} \cdot x\mathbf{i} + \mathbf{J} \cdot y\mathbf{j} + \mathbf{J} \cdot z\mathbf{k} \\ \mathbf{K} \cdot x\mathbf{i} + \mathbf{K} \cdot y\mathbf{j} + \mathbf{K} \cdot z\mathbf{k} \end{bmatrix} = \begin{bmatrix} xc\alpha - ys\alpha + 0 \\ xs\alpha + yc\alpha + 0 \\ 0 + 0 + z \end{bmatrix}$$



This is the rotation matrix we are familiar with which shows how to turn a point represented in local coordinate to the world coordinate system.

In projective geometry, we are interested in the inverse, i.e., how a point in the world coordinates is projected to the local (camera) coordinates.

Camera Transformation Matrix

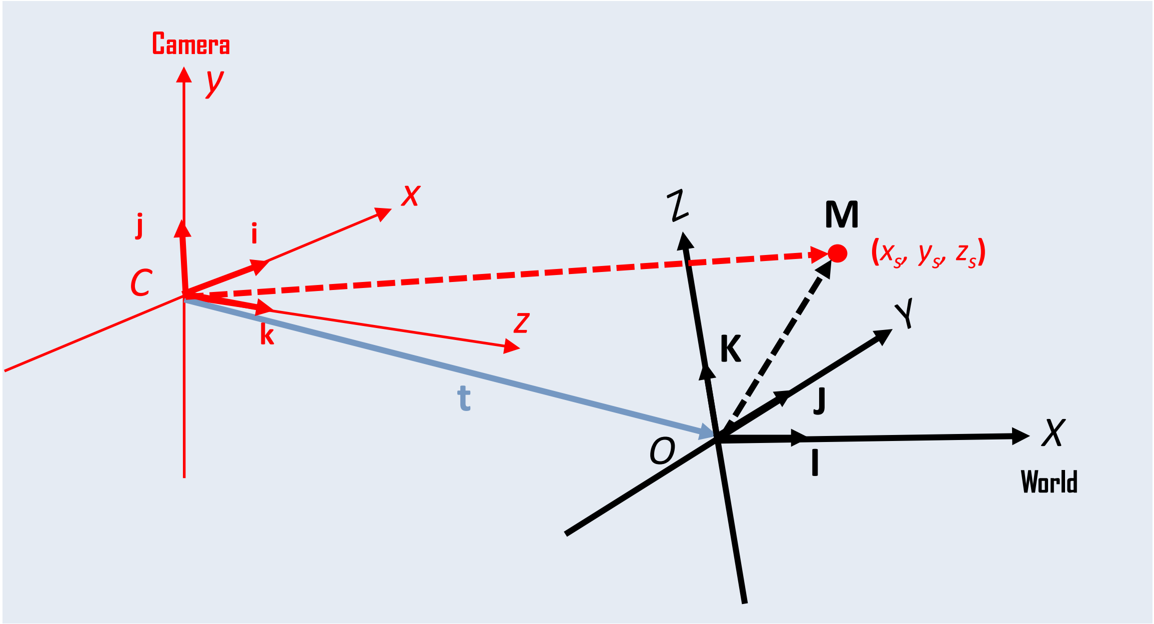


$$\overrightarrow{CM} = \overrightarrow{CO} + \overrightarrow{OM} = \mathbf{t} + \overrightarrow{OM}$$

$$\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} \mathbf{i}\cdot X\mathbf{I} + \mathbf{i}\cdot Y\mathbf{J} + \mathbf{i}\cdot Z\mathbf{K} \\ \mathbf{j}\cdot X\mathbf{I} + \mathbf{j}\cdot Y\mathbf{J} + \mathbf{j}\cdot Z\mathbf{K} \\ \mathbf{k}\cdot X\mathbf{I} + \mathbf{k}\cdot Y\mathbf{J} + \mathbf{k}\cdot Z\mathbf{K} \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} \mathbf{i}\cdot\mathbf{I} & \mathbf{i}\cdot\mathbf{J} & \mathbf{i}\cdot\mathbf{K} \\ \mathbf{j}\cdot\mathbf{I} & \mathbf{j}\cdot\mathbf{J} & \mathbf{j}\cdot\mathbf{K} \\ \mathbf{k}\cdot\mathbf{I} & \mathbf{k}\cdot\mathbf{J} & \mathbf{k}\cdot\mathbf{K} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

Rotation matrix R (to align global to local camera coordinates)

Camera Transformation – in homogeneous coordinates

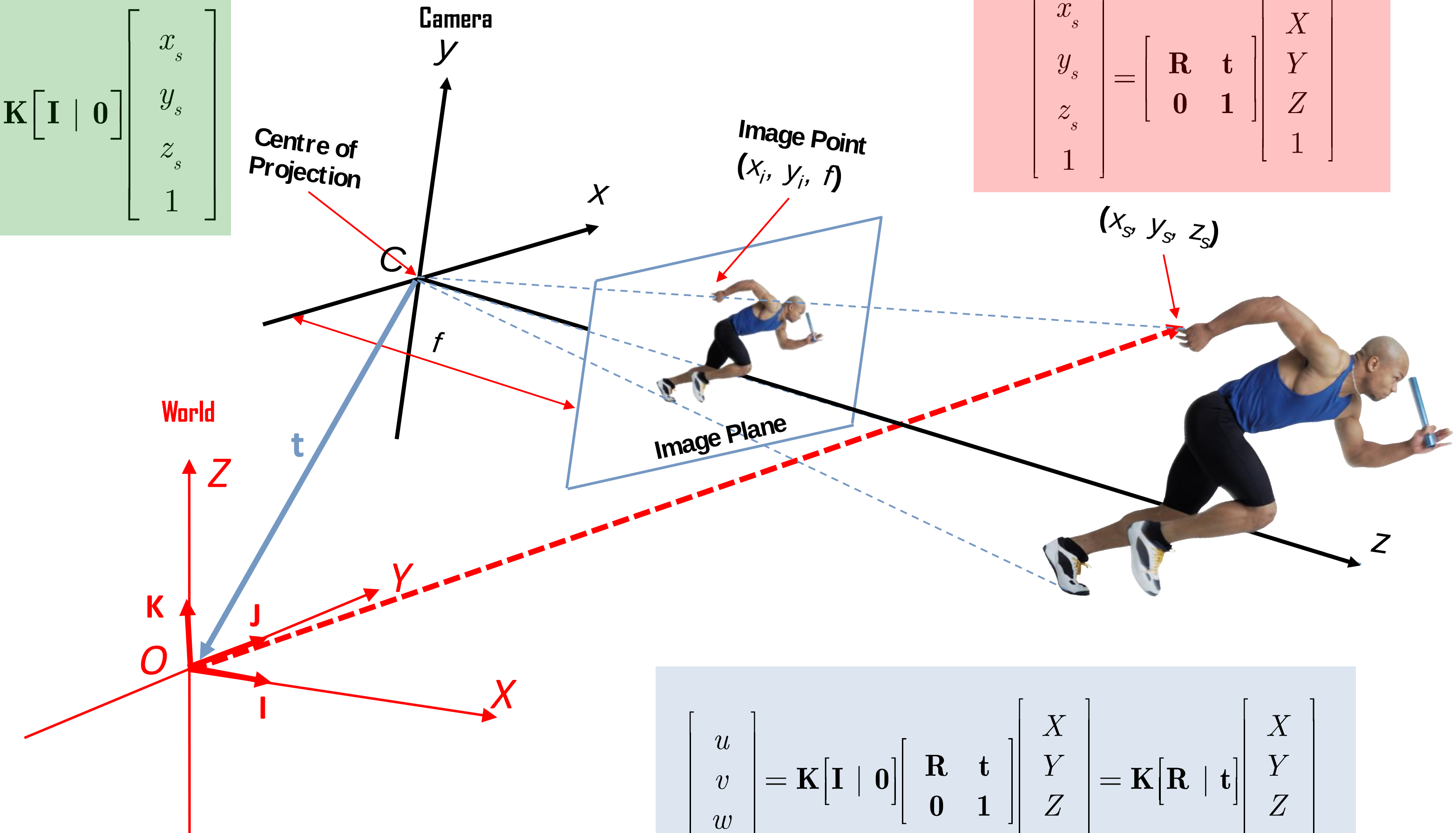


$$\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \mathbf{R} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \mathbf{t} \qquad \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Camera Projection Matrix

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{K}\left[\mathbf{I} \mid \mathbf{0}\right]\begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix}$$

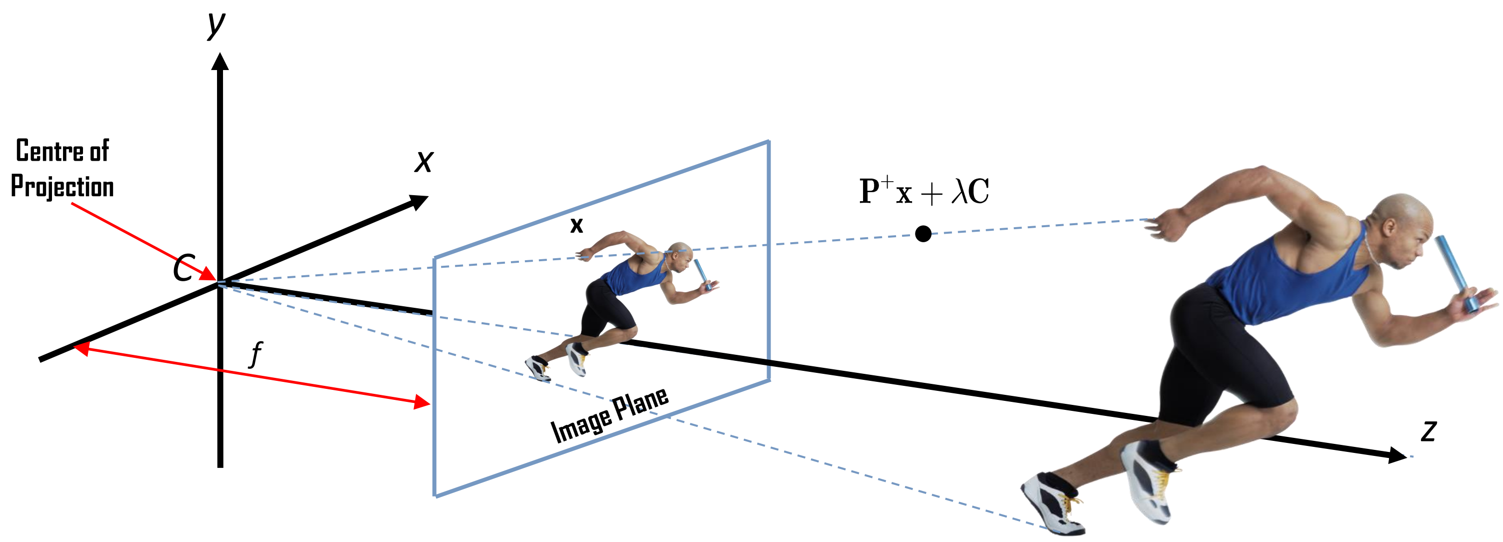
$$\begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$



$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{K}\left[\mathbf{I} \mid \mathbf{0}\right]\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{K}\left[\mathbf{R} \mid \mathbf{t}\right]\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

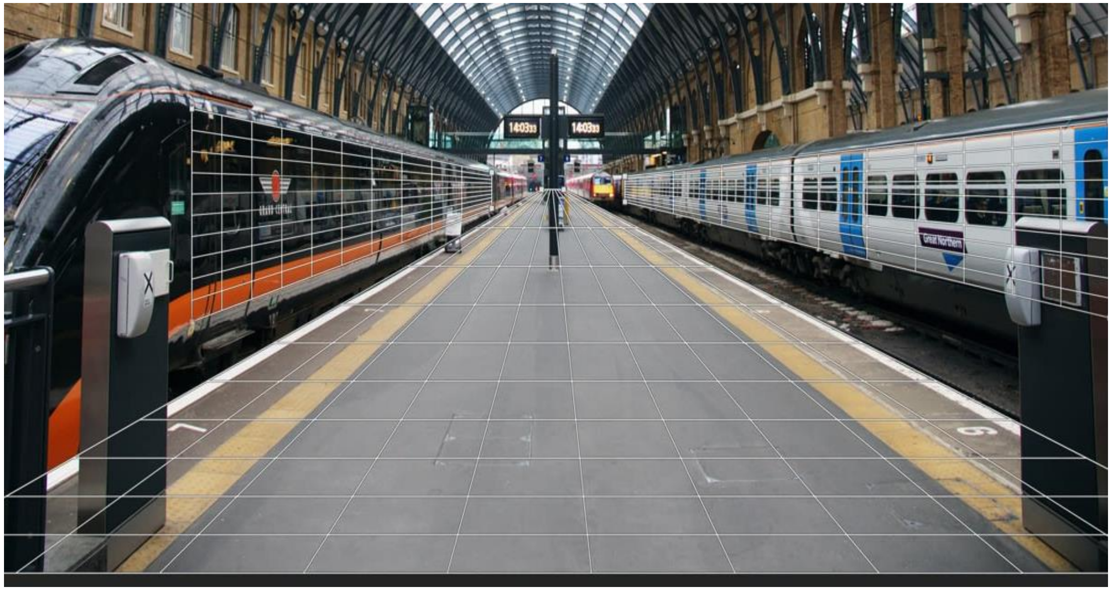
Camera Projection Matrix P=K[R | t]

Some Simple Properties of P



- The camera projection matrix defines how a point in the world is projected to the image plane (in homogeneous coordinates)
- P has 11 degrees-of-freedom (5 from calibration matrix **K**, 3 from **R**, and 3 from **t**)
- P is a 3x4 matrix with relatively general elements and its inverse can be represented as $\mathbf{P}^{+} = \mathbf{P}^{T}(\mathbf{P}\mathbf{P}^{T})^{-1}$ (the pseudo-inverse as it is not square)
- If **x** is the projection of point **X** to the camera coordinates, then **x=PX** defines the ray that links **C** with **X**, any point on that ray $\mathbf{P}^{+}\mathbf{x}+\lambda\mathbf{C}$ will project to the same point on the image plane.

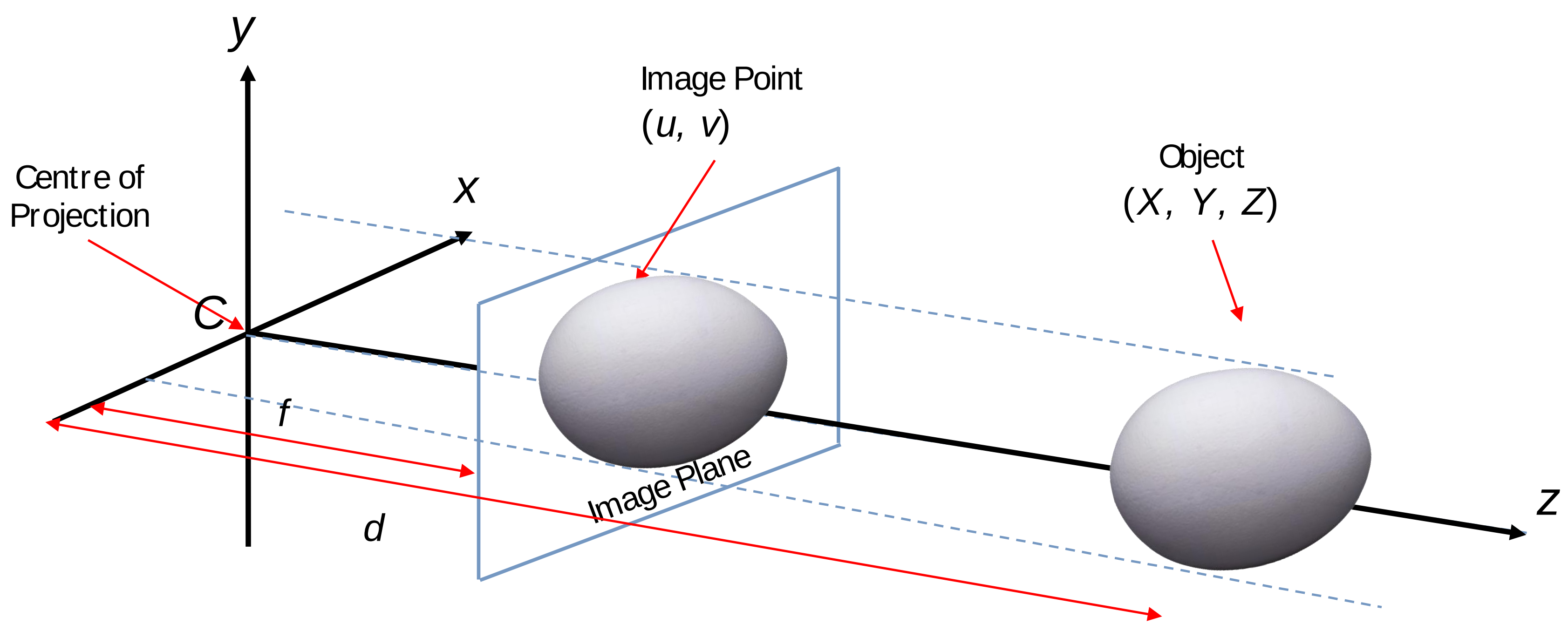
Geometric properties of perspective projection



- What happens to parallel lines?
- What happens to angles?
- What happens to distances?

- Lines in 3D project to lines in 2D
- Parallel lines do not in general project to parallel lines (unless they are parallel to the image plane). They intersect at a single point in the image plane called vanishing point or point at infinity.
- Distances and angles are not preserved
- Image size is inversely proportional to the object’s distance from the camera

Orthographic Projection



- When imaging objects far away from a camera, small differences in depth become less apparent. In this case the perspective effect can be ignored in a camera model.
- In the orthographic camera model, the image of a world point is found by simply translating the world point parallel to the optical axis until it lands in the image plane.

Orthographic Projection

Perspective Projection

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Orthographic Projection

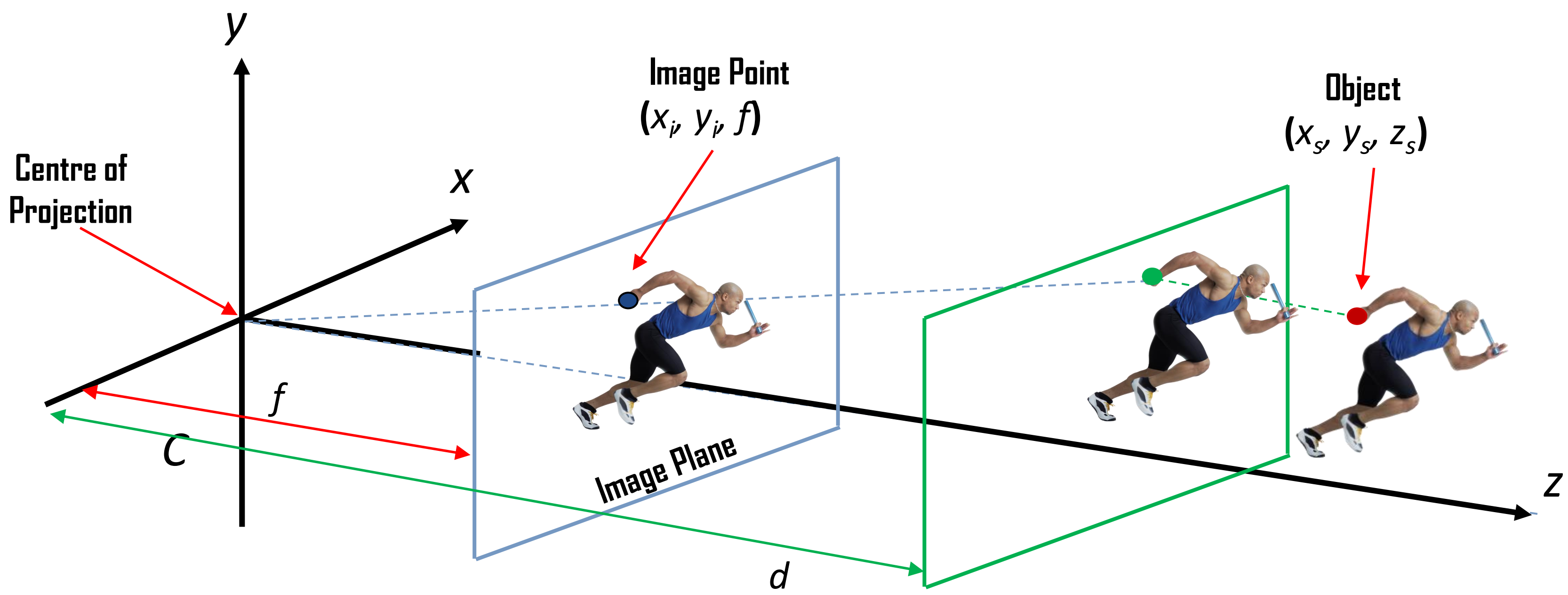
$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \alpha_x & s & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Geometric properties of orthographic projection

- What happens to parallel lines?
- What happens to angles?
- What happens to distances?

- Parallel lines project to parallel lines
- Image size does not change with the object's distance from the camera

Weak Perspective Projection

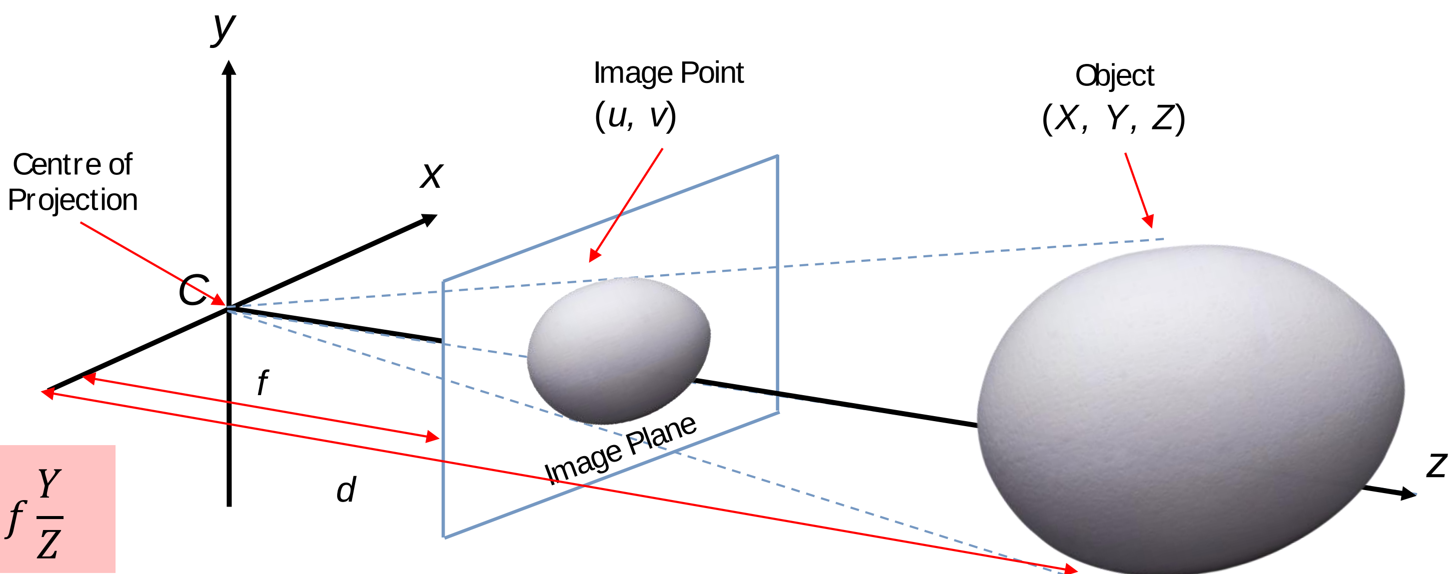


- The action of the weak perspective camera is equivalent to orthographic projection onto a plane (Z = d), followed by perspective projection from that plane.

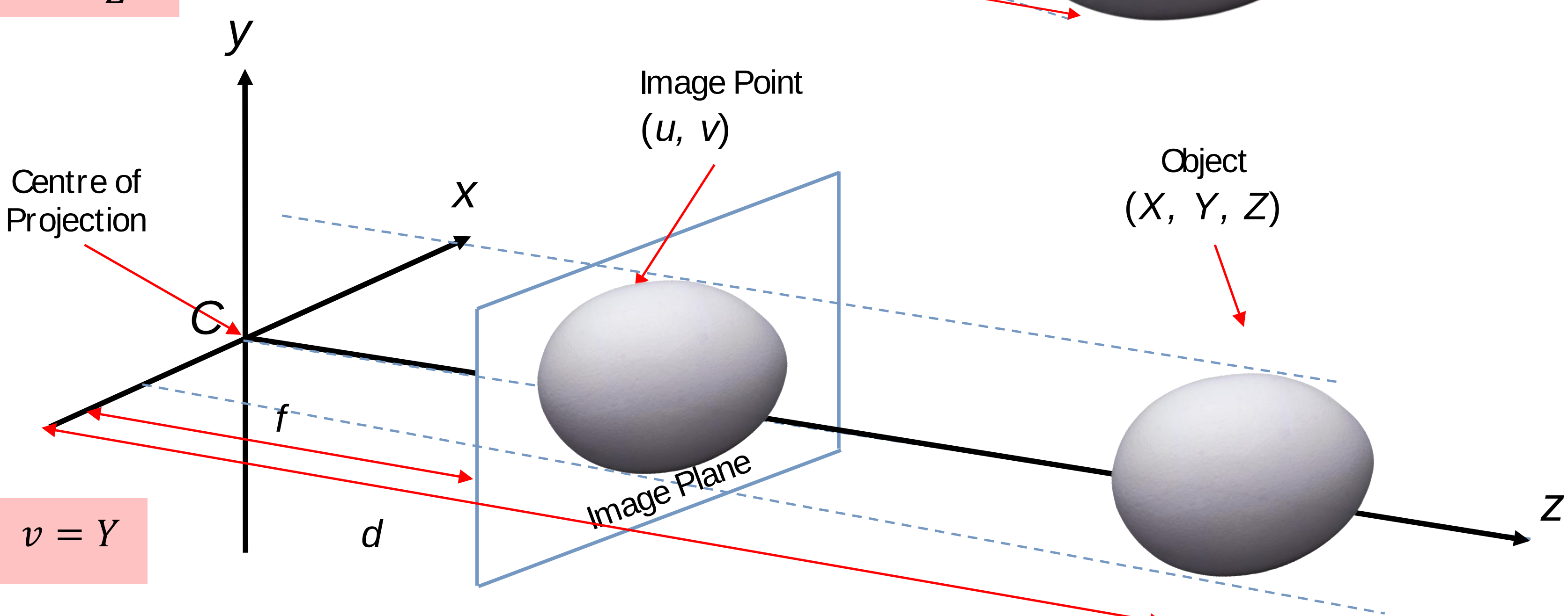
$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \alpha_x & s & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} k_x & 0 & 0 & 0 \\ 0 & k_y & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- Weak perspective projections are accurate when the object is small and distant from the camera.

Perspective vs Orthogonal Projection

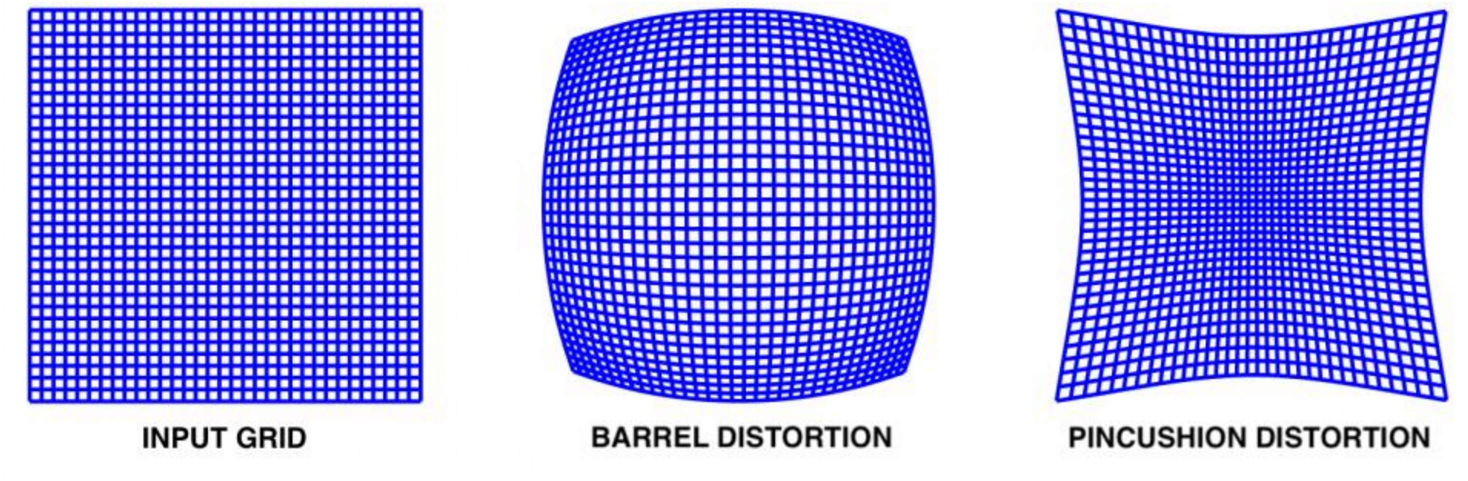


$$u = f\frac{X}{Z} \quad v = f\frac{Y}{Z}$$



$$u = X \quad v = Y$$

Radial Distortion



- Many wide-angle lenses have radial distortion, which creates a visible curvature in the projection of straight lines
- The coordinates in the distorted images are displaced away (**barrel distortion**) or towards (**pincushion distortion**) the image centre by an amount proportional to their radial distance
- Deviations are most noticeable for rays that pass through the edge of the lens

Radial Distortion

- The simplest radial distortion models use low-order polynomials
- From the pinhole camera model, the ideal normalised image coordinates are $[x_n, y_n] = \left[\frac{X}{Z}, \frac{Y}{Z}\right]$. The distorted point after radial distortion is given as:

$$x_d = x_n(1 + k_1 r^2 + k_2 r^4)$$
$$y_d = y_n(1 + k_1 r^2 + k_2 r^4)$$

where,

$$r = \sqrt{{x_n}^2 + {y_n}^2}$$

- After the radial distortion step, the final pixel coordinates can be computed as:

$$u_d = f x_d + u_0$$
$$v_d = f y_d + v_0$$

where, f is the focal length and (u_0, v_0) is the principal point.

Radial Distortion

- After having lens distortion corrected, the pinhole-camera model can be used

http://blog.photoshopcreative.co.uk/general/fix-barrel-distortion/

Contents

- 2D planar transformations
- Camera geometry and transformations
- Camera projection matrix
- Camera projection models:
  - Perspective
  - Orthographic
  - Weak-perspective
- Radial distortion