CSE 167:
Introduction to Computer Graphics
Lecture #2: Coordinate Transformations

Jürgen P. Schulze, Ph.D.
University of California, San Diego
Fall Quarter 2011

Announcements

- Homework #1 due Friday Sept 30, 1:30pm; presentation in lab 260
- Don't save anything on the C: drive of the lab PCs in Windows. You will lose it when you log out!

Overview

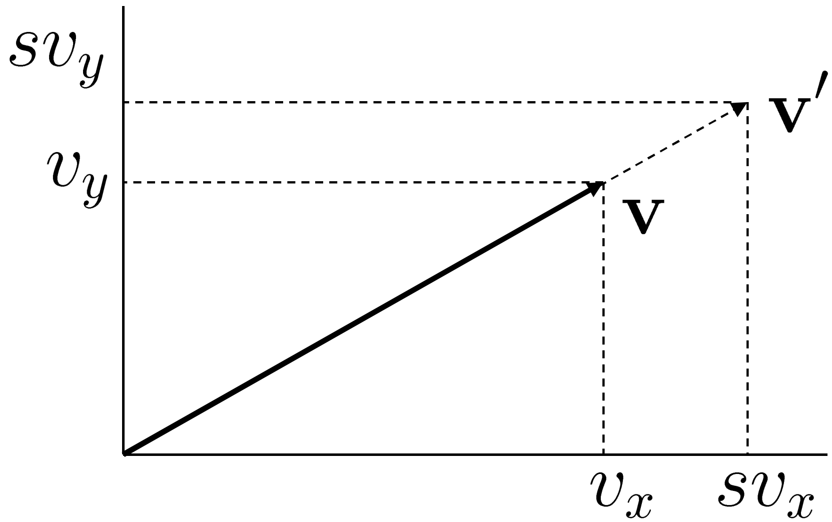
- Linear Transformations
- Homogeneous Coordinates
- Affine Transformations
- Concatenating Transformations
- Change of Coordinates
- Common Coordinate Systems

Linear Transformations

- Scaling, shearing, rotation, reflection of vectors, and combinations thereof
- Implemented using matrix multiplications

Scaling

- Uniform scaling matrix in 2D

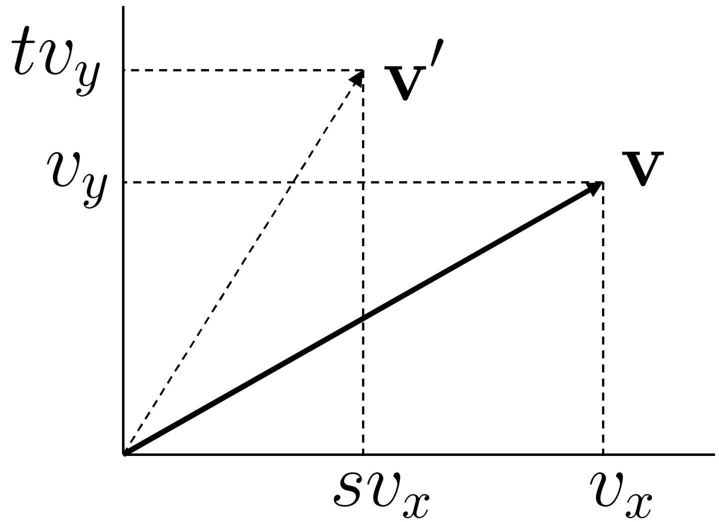


- Analogous in 3D

$$\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \mathbf{v} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} v_x' \\ v_y' \end{bmatrix} = \mathbf{v}'$$

Scaling

- Nonuniform scaling matrix in 2D

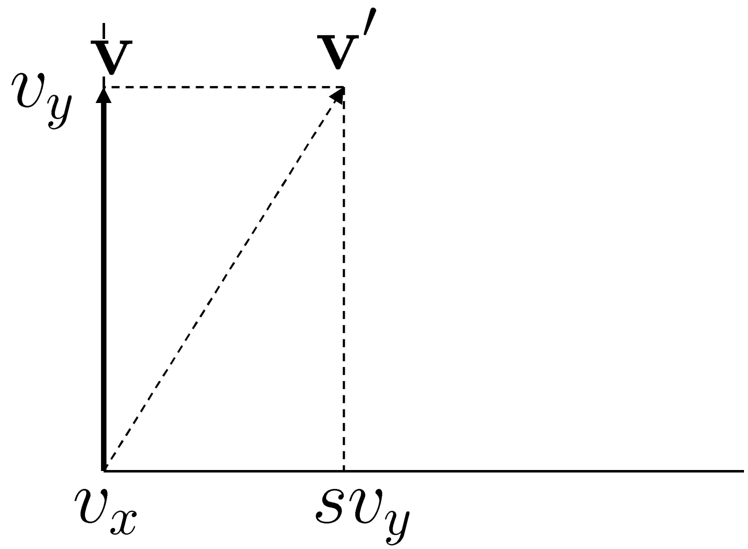


- Analogous in 3D

$$\begin{bmatrix} s & 0 \\ 0 & t \end{bmatrix} \mathbf{v} = \begin{bmatrix} s & 0 \\ 0 & t \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} v'_x \\ v'_y \end{bmatrix} = \mathbf{v}'$$

Shearing

- Shearing along x-axis in 2D



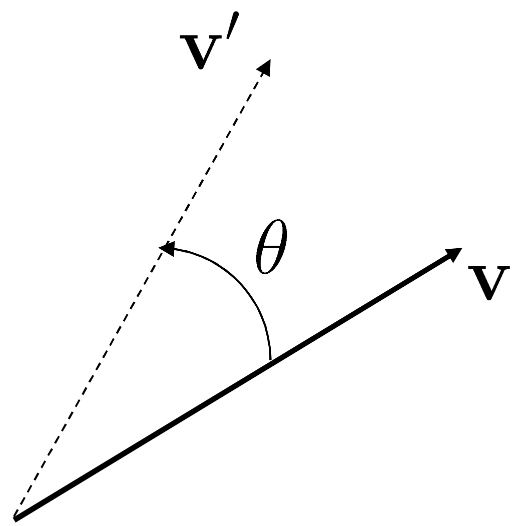
- Analogous for y-axis, in 3D

$$\mathbf{v}' = \begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix} \mathbf{v}$$

Rotation in 2D

- Convention: positive angle rotates counterclockwise
- Rotation matrix

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$



$$\mathbf{v}' = \mathbf{R}(\theta)\mathbf{v}$$

Rotation in 3D

Rotation around coordinate axes

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

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

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

Rotation in 3D

- Concatenation of rotations around x, y, z axes

$$\mathbf{R}_{x,y,z}(\theta_x, \theta_y, \theta_z) = \mathbf{R}_x(\theta_x)\mathbf{R}_y(\theta_y)\mathbf{R}_z(\theta_z)$$

- $\theta_x, \theta_y, \theta_z$ are called Euler angles
- Result depends on matrix order!

$$\mathbf{R}_x(\theta_x)\mathbf{R}_y(\theta_y)\mathbf{R}_z(\theta_z) \neq \mathbf{R}_z(\theta_z)\mathbf{R}_y(\theta_y)\mathbf{R}_x(\theta_x)$$

Rotation in 3D

Around arbitrary axis

$$\mathbf{R}(\mathbf{a},\theta)=\begin{bmatrix} 1+(1-\cos(\theta))(a_x^2-1) & -a_z\sin(\theta)+(1-\cos(\theta))a_x a_y & a_y\sin(\theta)+(1-\cos(\theta))a_x a_z \\ a_z\sin(\theta)+(1-\cos(\theta))a_y a_x & 1+(1-\cos(\theta))(a_y^2-1) & -a_x\sin(\theta)+(1-\cos(\theta))a_y a_z \\ -a_y\sin(\theta)+(1-\cos(\theta))a_z a_x & a_x\sin(\theta)+(1-\cos(\theta))a_z a_y & 1+(1-\cos(\theta))(a_z^2-1) \end{bmatrix}$$

- Rotation axis **a**
  - **a** must be a unit vector: $|\mathbf{a}|=1$
- Right-hand rule applies for direction of rotation
  - Counterclockwise rotation

Overview

- Linear Transformations
- Homogeneous Coordinates
- Affine Transformations
- Concatenating Transformations
- Change of Coordinates
- Common Coordinate Systems

Homogeneous Coordinates

- Generalization: homogeneous point

$$\mathbf{p}_h = wp_x\mathbf{x} + wp_y\mathbf{y} + wp_z\mathbf{z} + w\mathbf{o} \qquad \begin{bmatrix} wp_x \\ wp_y \\ wp_z \\ w \end{bmatrix}$$

- Homogeneous coordinate
- Corresponding 3D point: divide by homogeneous coordinate w

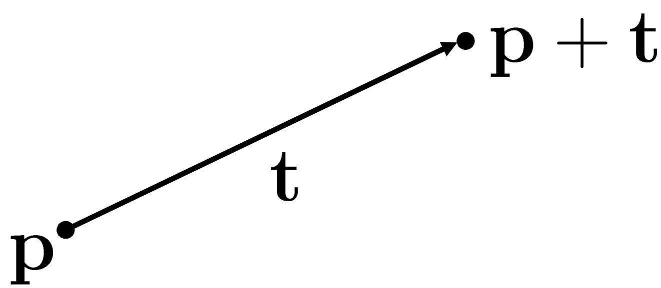
$$\mathbf{p} = p_x\mathbf{x} + p_y\mathbf{y} + p_z\mathbf{z} + \mathbf{o} \qquad \begin{bmatrix} wp_x/w \\ wp_y/w \\ wp_z/w \\ w/w \end{bmatrix}$$

Homogeneous coordinates

- Usually for 3D points you choose $w = 1$
- For 3D vectors $w = 0$
- Benefit: same representation for vectors and points

Translation

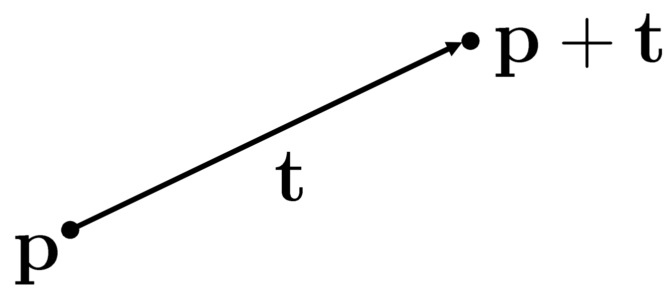
Using homogeneous coordinates



$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \qquad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \\ 0 \end{bmatrix} \qquad \mathbf{p} + \mathbf{t} = \begin{bmatrix} p_x + t_x \\ p_y + t_y \\ p_z + t_z \\ 1 \end{bmatrix}$$

Translation

Using homogeneous coordinates



Matrix notation

$$\mathbf{p}+\mathbf{t} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} p_x + t_x \\ p_y + t_y \\ p_z + t_z \\ 1 \end{bmatrix}$$

Translation matrix

Transformations

- Add 4th row/column to 3 x 3 transformation matrices
- Example: rotation

$$\mathbf{R}(\mathbf{a}, \theta) \in \mathbf{R}^{3\times 3}$$

$$\begin{bmatrix} & \mathbf{R}(\mathbf{a}, \theta) & & 0 \\ & & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformations

Concatenation of transformations:

- Arbitrary transformations (scale, shear, rotation, translation) $\mathbf{M}_3, \mathbf{M}_2, \mathbf{M}_1 \in \mathbf{R}^{4\times 4}$
- Build "chains" of transformations $\mathbf{p}'_h = \mathbf{M}_3\mathbf{M}_2\mathbf{M}_1\mathbf{p}_h$
- Result depends on order

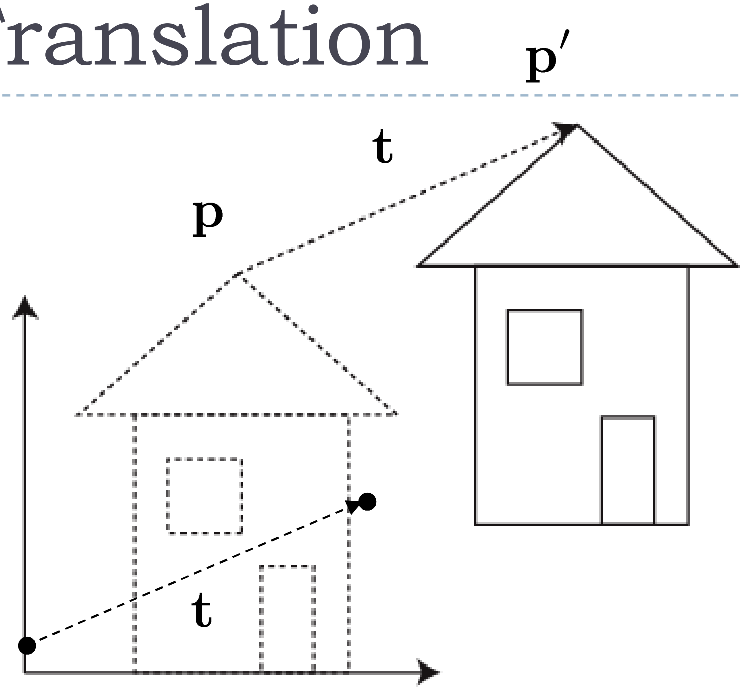
Overview

- Linear Transformations
- Homogeneous Coordinates
- Affine Transformations
- Concatenating Transformations
- Change of Coordinates
- Common Coordinate Systems

Affine transformations

- Generalization of linear transformations
  - Scale, shear, rotation, reflection (linear)
  - *Translation*
- Preserve straight lines, parallel lines
- Implementation using 4x4 matrices and homogeneous coordinates

Translation



$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \qquad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \\ 0 \end{bmatrix}$$

$$\mathbf{p}' = \mathbf{p} + \mathbf{t} = \begin{bmatrix} p_x + t_x \\ p_y + t_y \\ p_z + t_z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} p'_x \\ p'_y \\ p'_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

$$\mathbf{p}' = \mathbf{T}(\mathbf{t})\mathbf{p}$$

Translation

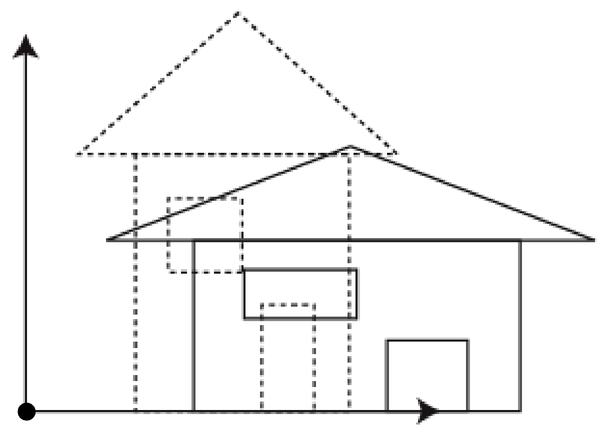
- Inverse translation

$$\mathbf{T}(\mathbf{t})^{-1} = \mathbf{T}(-\mathbf{t})$$

$$\mathbf{T}(\mathbf{t}) = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{T}(-\mathbf{t}) = \begin{bmatrix} 1 & 0 & 0 & -t_x \\ 0 & 1 & 0 & -t_y \\ 0 & 0 & 1 & -t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scaling

- Origin does not change



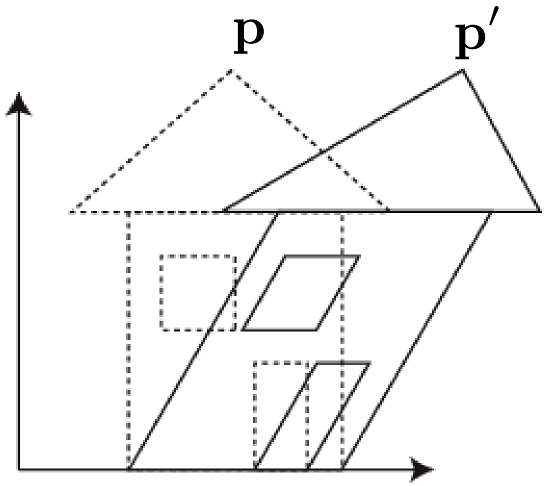
$$\mathbf{S}(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scaling

- Inverse of scale:

$$\mathbf{S}(s_x, s_y, s_z)^{-1} = \mathbf{S}(1/s_x, 1/s_y, 1/s_z)$$

Shear



$$\mathbf{p}' = \begin{bmatrix} 1 & z \\ 0 & 1 \end{bmatrix} \mathbf{p}$$

- Pure shear if only one parameter is non-zero

$$\mathbf{Z}(z_1 \ldots z_6) = \begin{bmatrix} 1 & z_1 & z_2 & 0 \\ z_3 & 1 & z_4 & 0 \\ z_5 & z_6 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

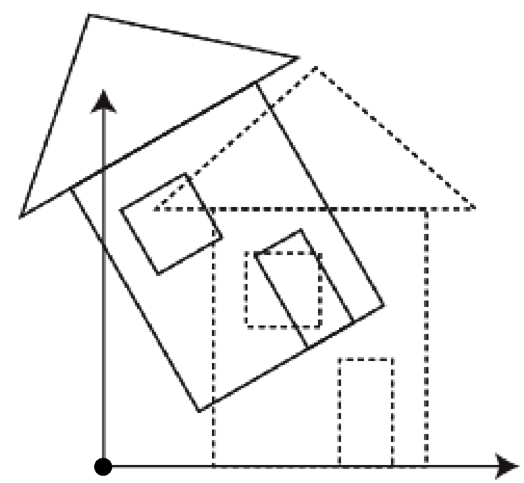
Rotation around coordinate axis

- Origin does not change

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

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

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



Rotation around arbitrary axis

- Origin does not change
- Angle θ, unit axis $\mathbf{a}$
- $c_\theta = \cos\theta, s_\theta = \sin\theta$

$$\mathbf{R}(\mathbf{a},\theta) = \begin{bmatrix} a_x^2 + c_\theta(1-a_x^2) & a_x a_y(1-c_\theta) - a_z s_\theta & a_x a_z(1-c_\theta) + a_y s_\theta & 0 \\ a_x a_y(1-c_\theta) + a_z s_\theta & a_y^2 + c_\theta(1-a_y^2) & a_y a_z(1-c_\theta) - a_x s_\theta & 0 \\ a_x a_z(1-c_\theta) - a_y s_\theta & a_y a_z(1-c_\theta) + a_x s_\theta & a_z^2 + c_\theta(1-a_z^2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation matrices

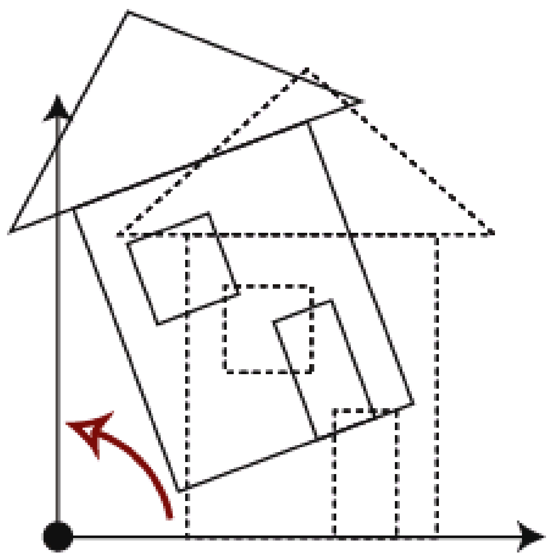
- Orthonormal
  - Rows, columns are unit length and orthogonal
- Inverse of rotation matrix:
  - Its transpose

$$\mathbf{R}(\mathbf{a}, \theta)^{-1} = \mathbf{R}(\mathbf{a}, \theta)^T$$

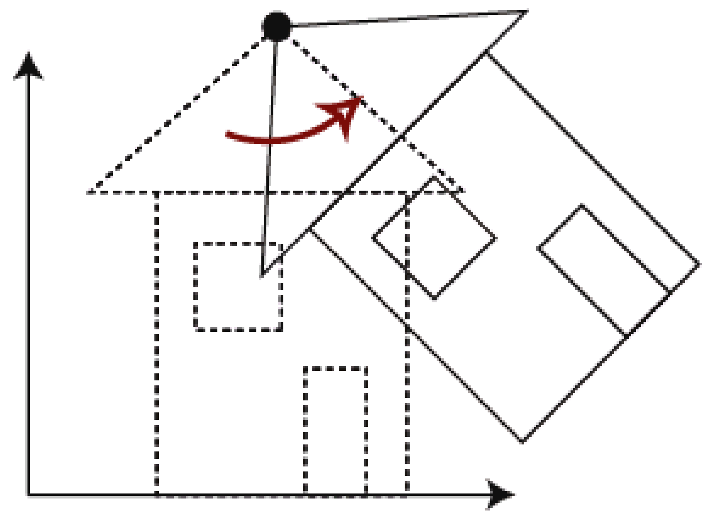
Overview

- Linear Transformations
- Homogeneous Coordinates
- Affine Transformations
- Concatenating Transformations
- Change of Coordinates
- Common Coordinate Systems

Rotating with pivot

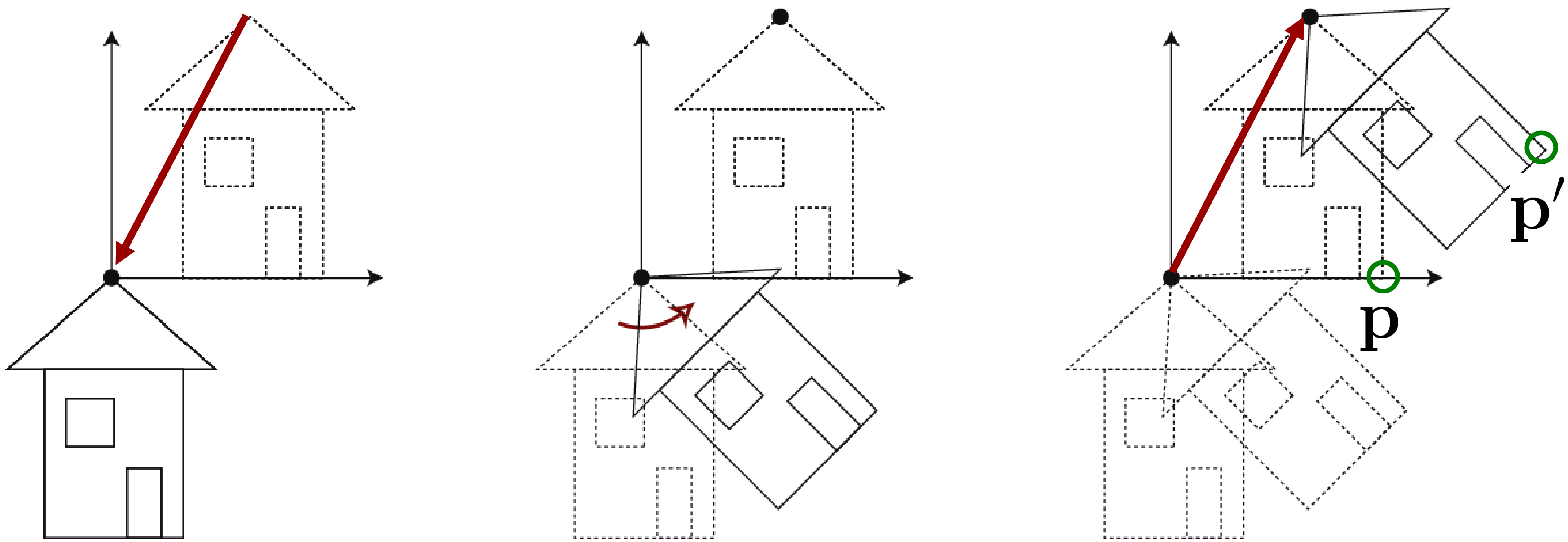


Rotation around origin



Rotation with pivot

Rotating with pivot



1. Translation $\mathbf{T}$ 2. Rotation $\mathbf{R}$ 3. Translation $\mathbf{T}^{-1}$

$$\mathbf{p}' = \mathbf{T}^{-1}\mathbf{R}\mathbf{T}\mathbf{p}$$

Concatenating transformations

- Arbitrary sequence of transformations

$$\mathbf{p}' = \mathbf{M}_3\mathbf{M}_2\mathbf{M}_1\mathbf{p}$$

$$\mathbf{M}_{total} = \mathbf{M}_3\mathbf{M}_2\mathbf{M}_1$$

$$\mathbf{p}' = \mathbf{M}_{total}\mathbf{p}$$

- Note: associativity

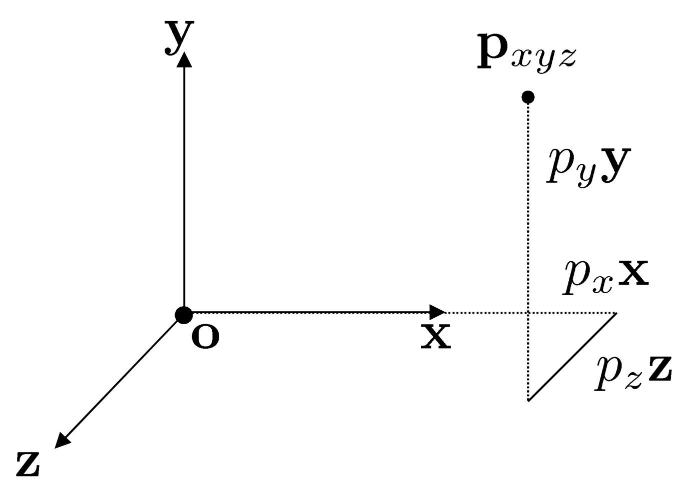
$$\mathbf{M}_{total} = (\mathbf{M}_3\mathbf{M}_2)\mathbf{M}_1 = \mathbf{M}_3(\mathbf{M}_2\mathbf{M}_1)$$

Overview

- Linear Transformations
- Homogeneous Coordinates
- Affine Transformations
- Concatenating Transformations
- Change of Coordinates
- Common Coordinate Systems

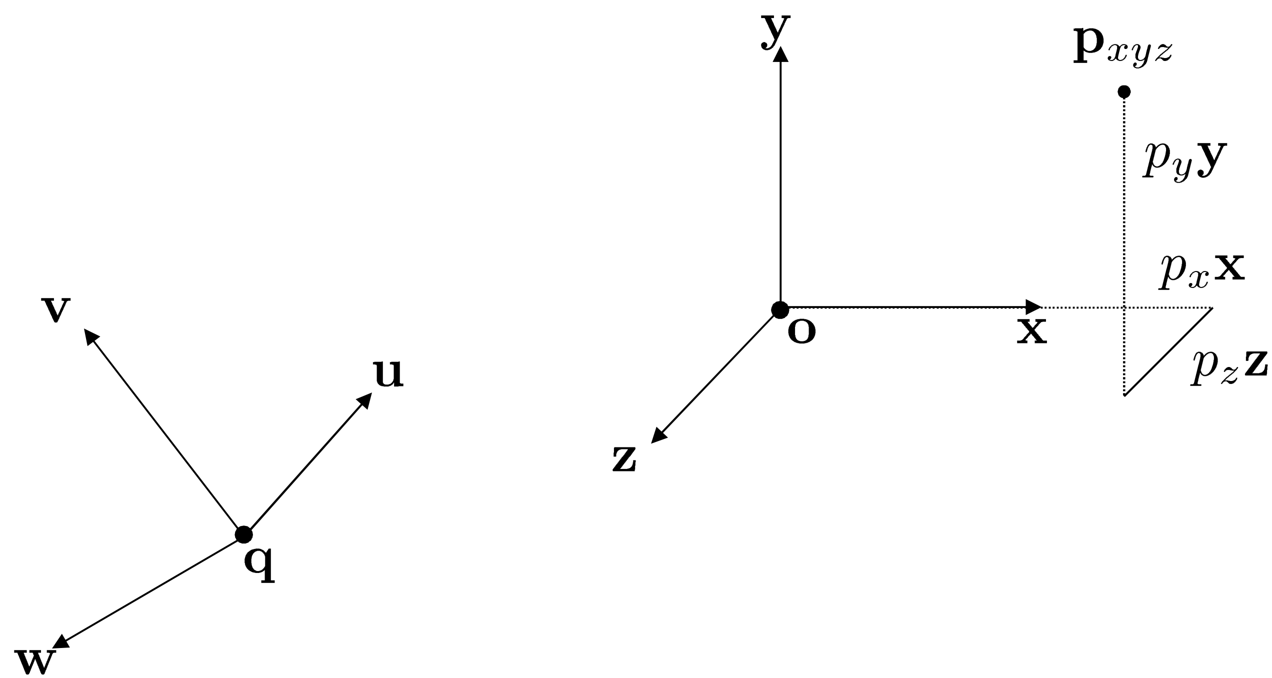
Change of coordinates

- Point with homogeneous coordinates $\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$
- Position in 3D given with respect to a coordinate system



$$\mathbf{p}_{xyz} = p_x\mathbf{x} + p_y\mathbf{y} + p_z\mathbf{z} + \mathbf{o}$$

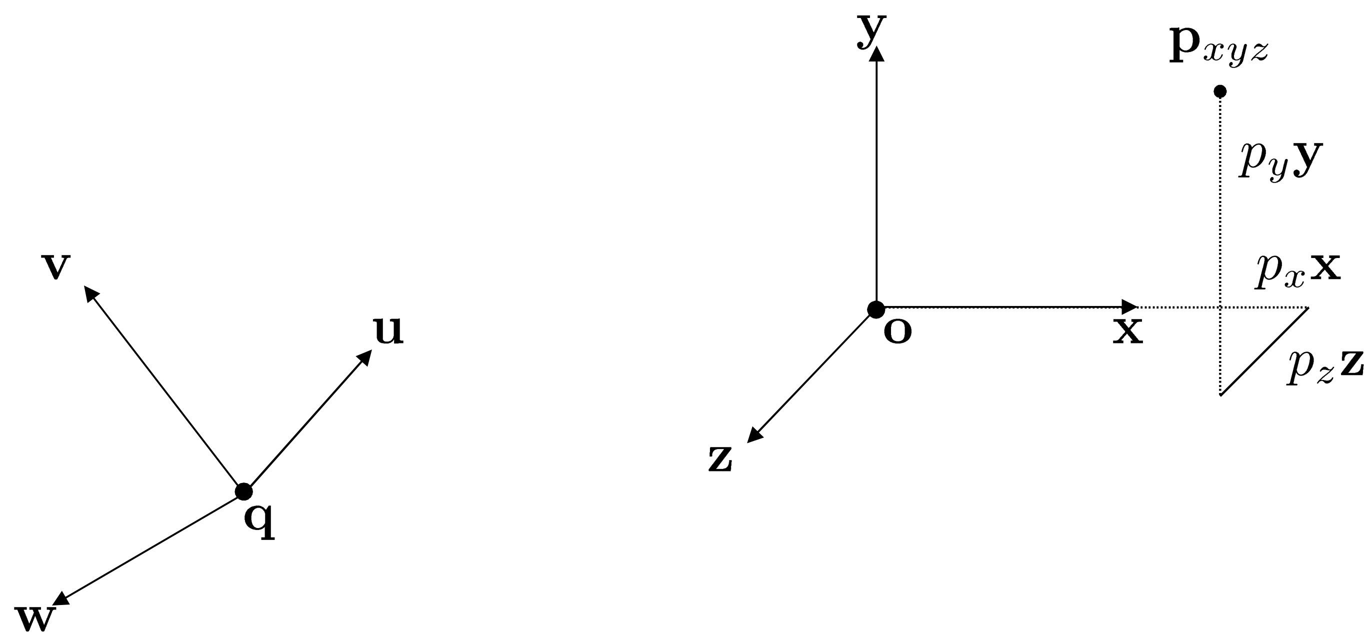
Change of coordinates



New $\mathbf{uvwq}$
coordinate system

Goal: Find coordinates of $\mathbf{p}_{xyz}$ with respect to new $\mathbf{uvwq}$ coordinate system

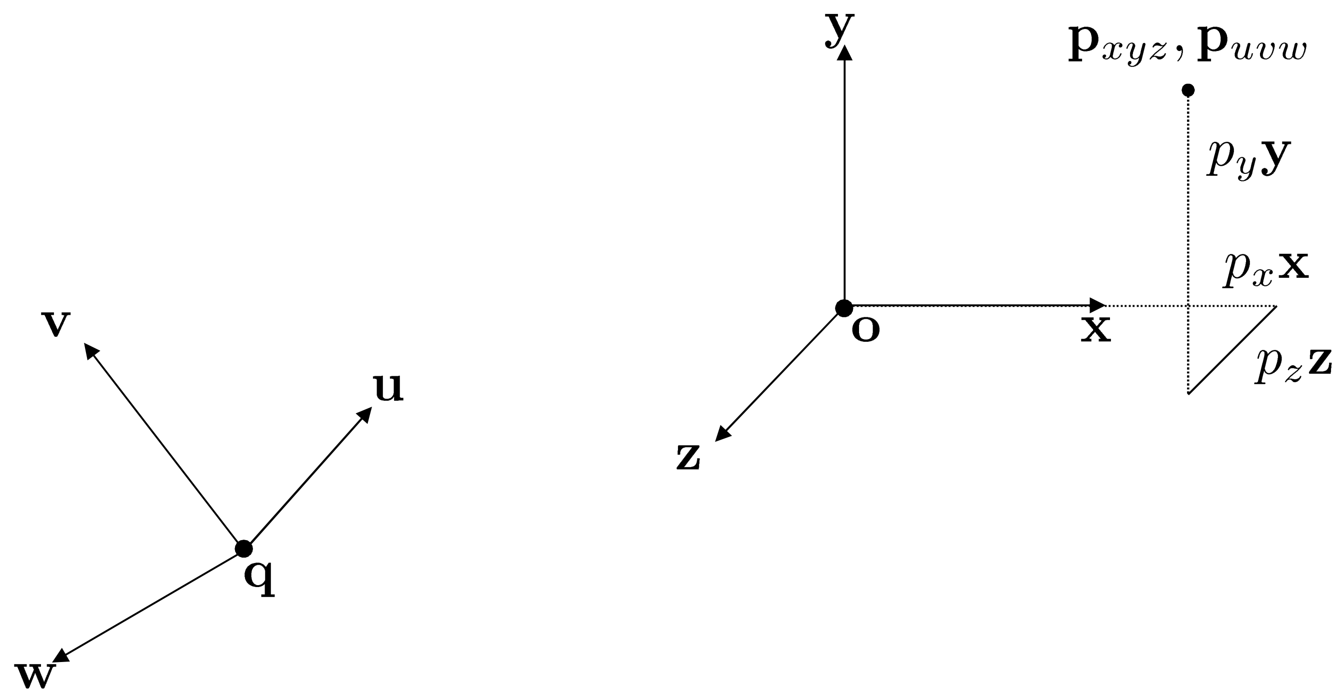
Change of coordinates



Coordinates of **xyzo** frame w.r.t. **uvwq** frame

$$\mathbf{x} = \begin{bmatrix} x_u \\ x_v \\ x_w \\ 0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_u \\ y_v \\ y_w \\ 0 \end{bmatrix} \quad \mathbf{z} = \begin{bmatrix} z_u \\ z_v \\ z_w \\ 0 \end{bmatrix} \quad \mathbf{o} = \begin{bmatrix} o_u \\ o_v \\ o_w \\ 1 \end{bmatrix}$$

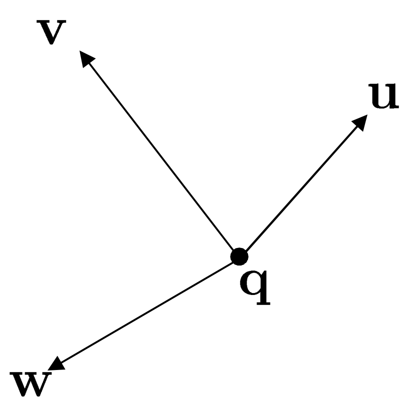
Change of coordinates

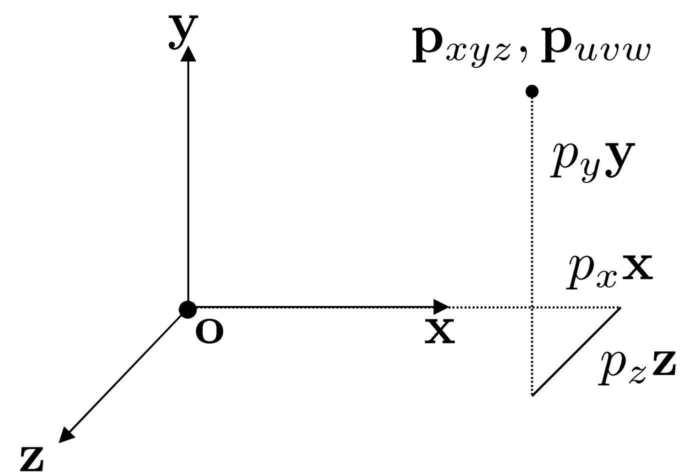


Same point $\mathbf{p}$ in 3D, expressed in new $\mathbf{uvwq}$ frame

$$\mathbf{p}_{uvw} = p_x \begin{bmatrix} x_u \\ x_v \\ x_w \\ 0 \end{bmatrix} + p_y \begin{bmatrix} y_u \\ y_v \\ y_w \\ 0 \end{bmatrix} + p_z \begin{bmatrix} z_u \\ z_v \\ z_w \\ 0 \end{bmatrix} + \begin{bmatrix} o_u \\ o_v \\ o_w \\ 1 \end{bmatrix}$$

Change of coordinates





$$\mathbf{p}_{uvw} = \begin{bmatrix} x_u & y_u & z_u & o_u \\ x_v & y_v & z_v & o_v \\ x_w & y_w & z_w & o_w \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{x} & \mathbf{y} & \mathbf{z} & \mathbf{o} \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

Change of coordinates

Inverse transformation

- Given point $\mathbf{p}_{uvw}$ w.r.t. frame $\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{q}$
- Coordinates $\mathbf{p}_{xyz}$ w.r.t. frame $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{o}$

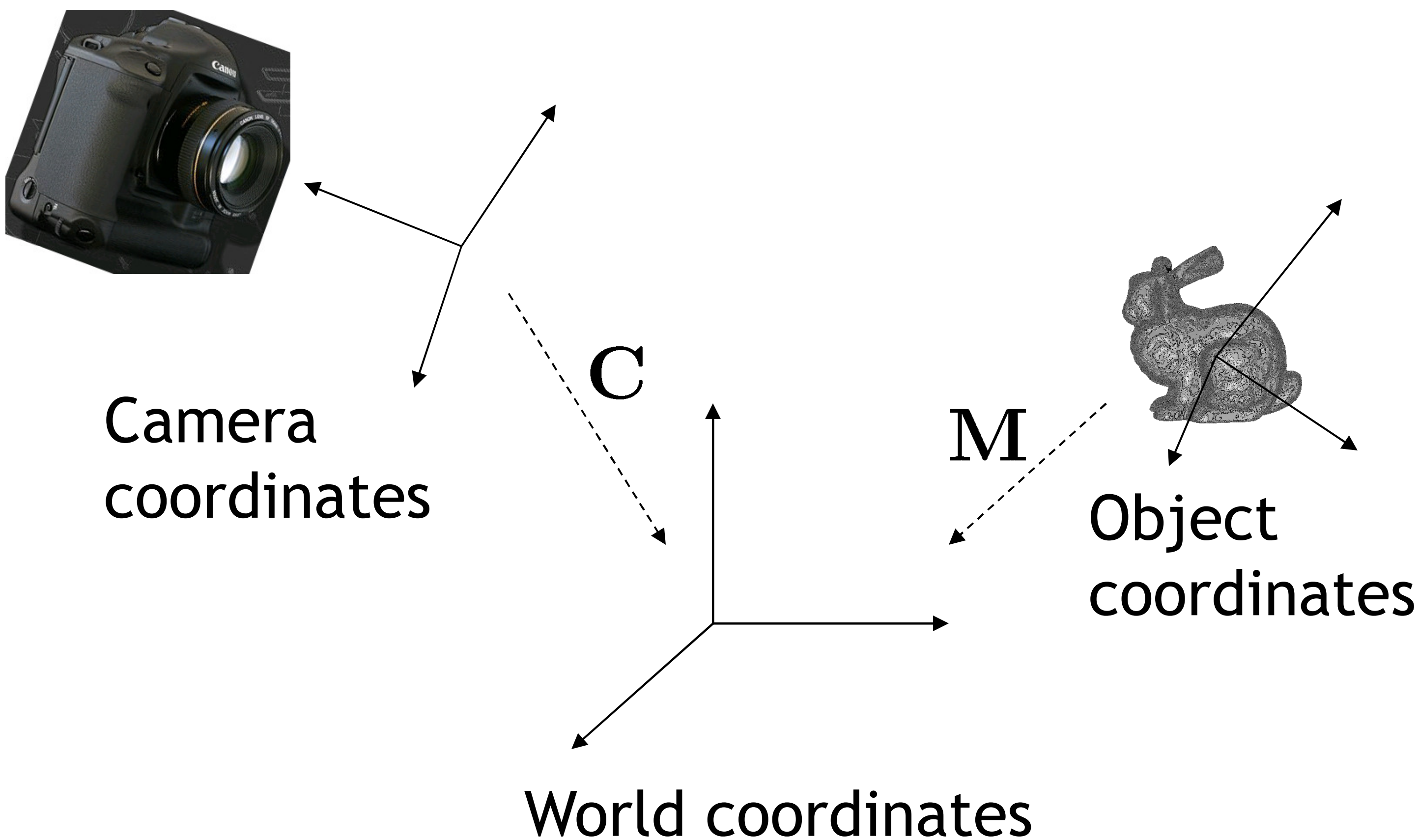
$$\mathbf{p}_{xyz} = \begin{bmatrix} x_u & y_u & z_u & o_u \\ x_v & y_v & z_v & o_v \\ x_w & y_w & z_w & o_w \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} p_u \\ p_v \\ p_w \\ 1 \end{bmatrix}$$

Overview

- Linear Transformations
- Homogeneous Coordinates
- Affine Transformations
- Concatenating Transformations
- Change of Coordinates
- Typical Coordinate Systems

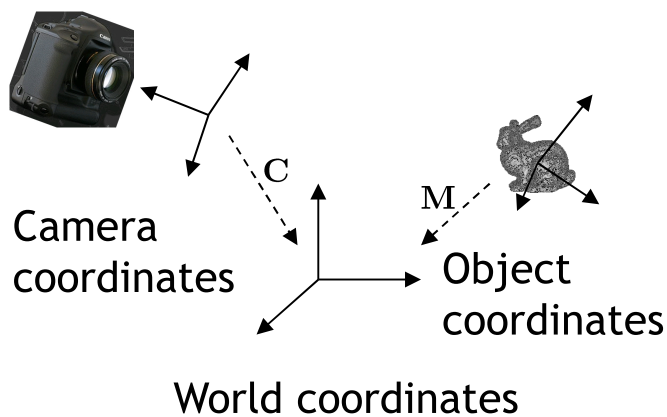
Typical Coordinate Systems

- Camera, world, object coordinates:



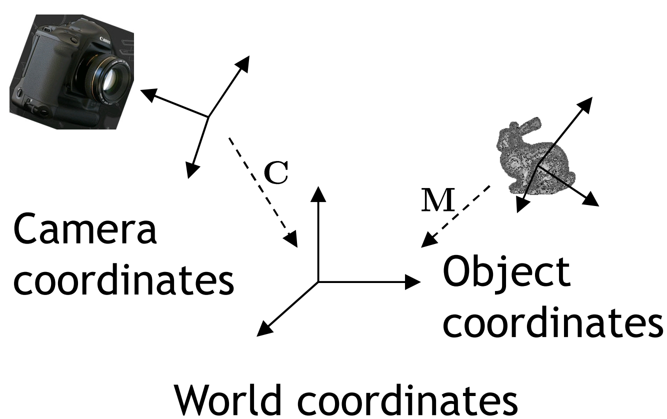
Object Coordinates

- Coordinates the object is defined with
- Often origin is in middle, base, or corner of object
- No right answer, whatever was convenient for the creator of the object



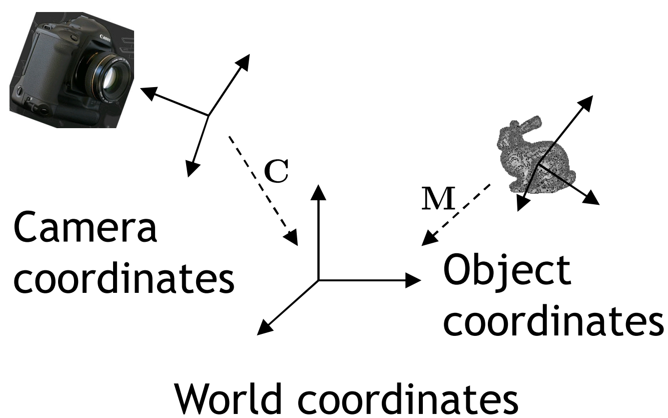
World Coordinates

- "World space"
- Common reference frame for all objects in the scene
- Chosen for convenience, no right answer
  - If there is a ground plane, usually x/y is horizontal and z points up (height)
  - In OpenGL x/y is screen plane, z comes out



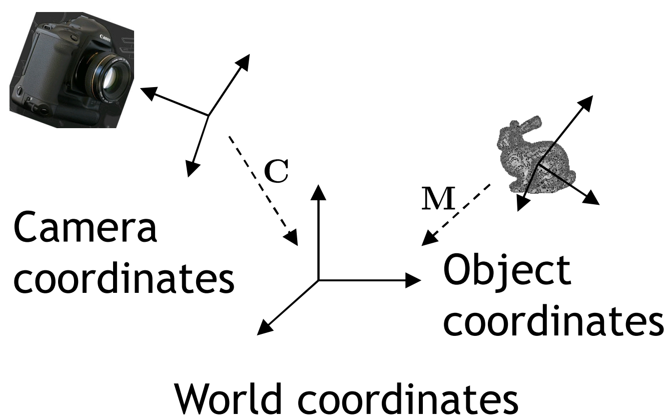
World Coordinates

- Transformation from object to world space is different for each object
- Defines placement of object in scene
- Given by "model matrix" (model-to-world transform) **M**



Camera Coordinate System

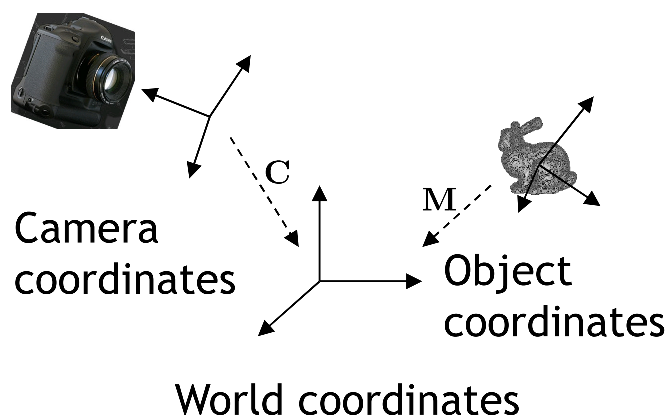
- "Camera space"
- Origin defines center of projection of camera
- x-y plane is parallel to image plane
- z-axis is perpendicular to image plane



Camera Coordinate System

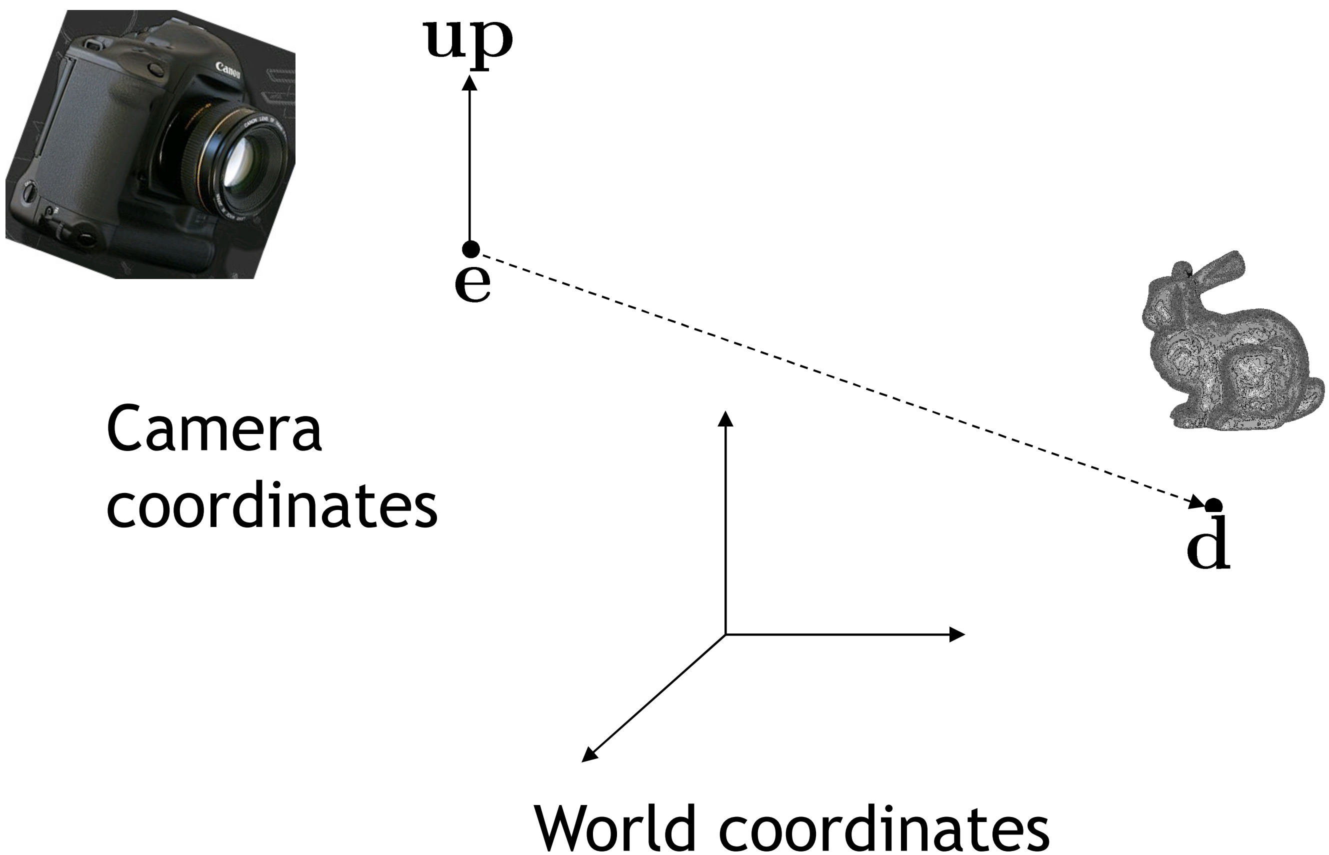
- The Camera Matrix defines the transformation from camera to world coordinates
  - Placement of camera in world
- Transformation from object to camera coordinates

$$\mathbf{p}_{camera} = \mathbf{C}^{-1}\mathbf{M}\mathbf{p}_{object}$$



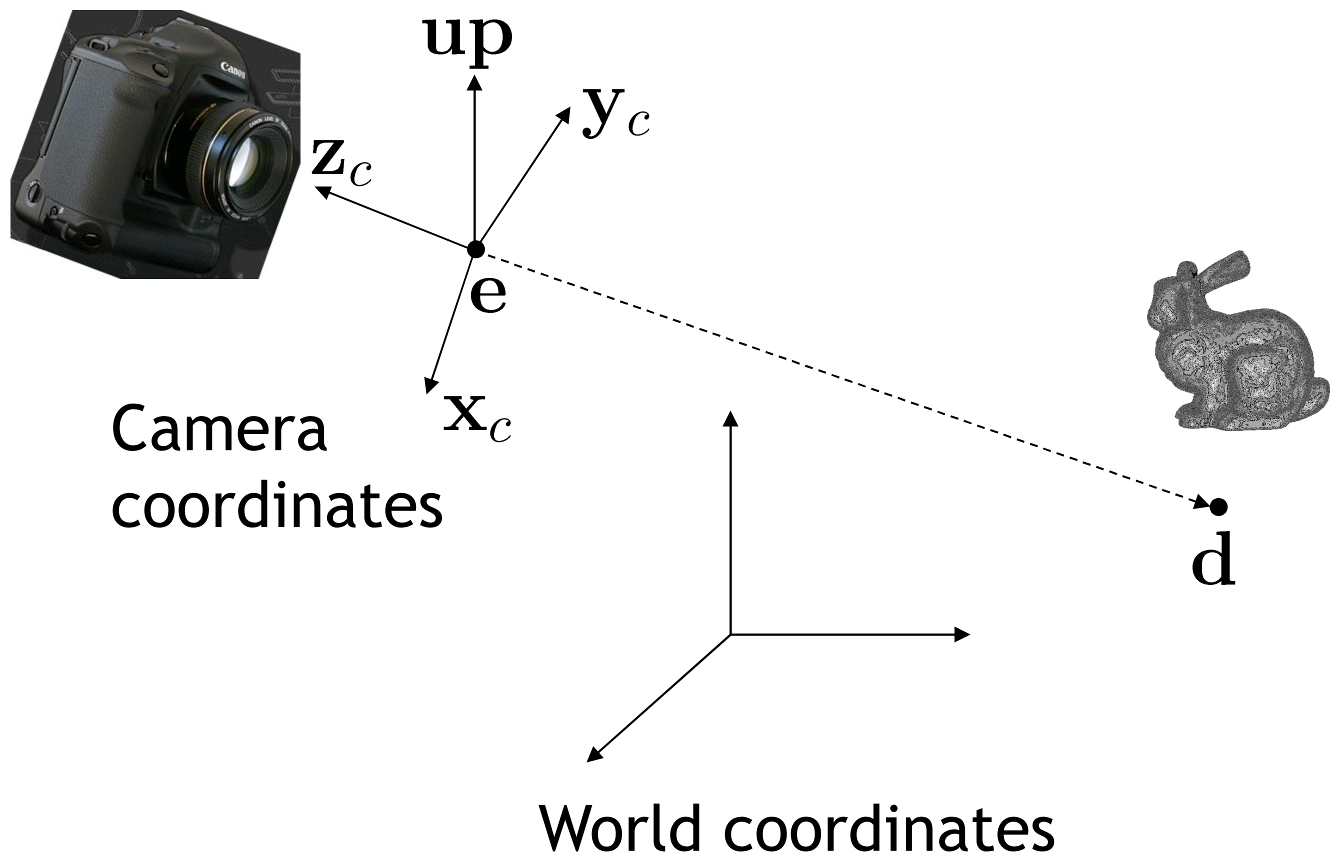
Camera Matrix

- Construct from center of projection **e**, look at **d**, up-vector **up:**



Camera Matrix

- Construct from center of projection **e**, look at **d**, up-vector **up:**



Camera Matrix

- z-axis

$$\mathbf{z}_c = \frac{\mathbf{e} - \mathbf{d}}{\|\mathbf{e} - \mathbf{d}\|}$$

- x-axis

$$\mathbf{x}_c = \frac{\mathbf{up} \times \mathbf{z}_c}{\|\mathbf{up} \times \mathbf{z}_c\|}$$

- y-axis

$$\mathbf{y}_c = \mathbf{z_c} \times \mathbf{x}_c$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{x_c} & \mathbf{y_c} & \mathbf{z_c} & \mathbf{e} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Inverse of Camera Matrix

- How to calculate the inverse of the camera matrix $\mathbf{C}^{-1}$?
- Generic matrix inversion is complex and compute-intensive
- Observation:
  - camera matrix consists of rotation and translation: $\mathbf{R} \times \mathbf{T}$
- Inverse of rotation: $\mathbf{R}^{-1} = \mathbf{R}^{T}$
- Inverse of translation: $\mathbf{T}(t)^{-1} = \mathbf{T}(-t)$
- Inverse of camera matrix: $\mathbf{C}^{-1} = \mathbf{T}^{-1} \times \mathbf{R}^{-1}$

Objects in Camera Coordinates

- We have things lined up the way we like them on screen
  - x to the right
  - y up
  - $-z$ going into the screen
  - Objects to look at are in front of us, i.e. have negative z values
- But objects are still in 3D
- Next step: project scene into 2D

Next Lecture

- Rendering Pipeline
- Perspective Projection