Today
- Homogeneous coordinates
- Affine transformations
- Concatenating transformations
- Change of coordinates
- Common coordinate systems

Vectors in 3D
- Orientation, length
- Describe using three basis vectors \( x, y, z \)

Points in 3D
- 3D location
- Describe using three basis vectors and origin

Vectors vs. points
- Vectors
  \[
  \mathbf{v} = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}
  \]
- Points
  \[
  \mathbf{p} = p_x \mathbf{i} + p_y \mathbf{j} + p_z \mathbf{k} + \mathbf{o}
  \]

Homogeneous coordinates
- Homogeneous point
  \[
  \mathbf{p}_h = w p_x \mathbf{i} + w p_y \mathbf{j} + w p_z \mathbf{k} + w \mathbf{o}
  \]
- Homogeneous coordinate \( w \)
- Corresponding 3D point: divide by homogeneous coordinate
  \[
  \mathbf{p} = \frac{w p_x}{w} \mathbf{i} + \frac{w p_y}{w} \mathbf{j} + \frac{w p_z}{w} \mathbf{k} + \mathbf{o}
  \]
Homogeneous coordinates

- Usually for 3D points you choose $w = 1$
- For 3D vectors $w = 0$

Benefit
- Same representation for vectors and points
- Can represent translation of points

Translation

Using homogeneous coordinates

$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$ $\begin{bmatrix} t_x \\ t_y \\ t_z \\ 0 \end{bmatrix}$ $\begin{bmatrix} p_x + t_x \\ p_y + t_y \\ p_z + t_z \\ 1 \end{bmatrix}$

Today

- 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

$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$ $\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$ $\begin{bmatrix} p_x + t_x \\ p_y + t_y \\ p_z + t_z \end{bmatrix}$

Translation

$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$ $\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$

$p' = T(t)p$
Translation

- Inverse translation

\[ T(t)^{-1} = T(-t) \]
\[
T(t) = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
T(-t) = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

- Verify that

\[ T(-t)T(t) = T(t)T(-t) = I \]

Scaling

- Origin does not change

\[
S(s_x, s_y, s_z) = \begin{bmatrix}
s_x & 0 & 0 \\
0 & s_y & 0 \\
0 & 0 & s_z
\end{bmatrix}
\]

Scaling

- Inverse scaling?

\[ S(s_x, s_y, s_z)^{-1} = S(1/s_x, 1/s_y, 1/s_z) \]

Shear

- Pure shear if only one parameter is non-zero
- Cartoon-like effects

\[
\mathbf{P} = \begin{bmatrix}
x1 & y1 & 0 \\
x2 & y2 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

Rotation around coordinate axis

- Origin does not change

\[
R_x(\theta) = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \theta & -\sin \theta \\
0 & \sin \theta & \cos \theta
\end{bmatrix}
\]

\[
R_y(\theta) = \begin{bmatrix}
\cos \theta & 0 & \sin \theta \\
0 & 1 & 0 \\
-\sin \theta & 0 & \cos \theta
\end{bmatrix}
\]

\[
R_z(\theta) = \begin{bmatrix}
\cos \theta & -\sin \theta & 0 \\
\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{bmatrix}
\]
Rotation around arbitrary axis

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

\[
R(\mathbf{a}, \theta) = \begin{bmatrix}
    a_x^2 + c(1 - a_z^2) & a_y(a_z(1 - c) - a_x s) & a_z(a_x(1 - c) + a_y s) & 0 \\
    a_y(a_z(1 - c) + a_x s) & a_y^2 + c(1 - a_x^2) & a_z(a_y(1 - c) - a_z s) & 0 \\
    a_z(a_x(1 - c) - a_y s) & a_z(a_y(1 - c) + a_x s) & a_z^2 + c(1 - a_y^2) & 0 \\
    0 & 0 & 0 & 1
\end{bmatrix}
\]

Rotation around arbitrary axis

Application
- “Virtual trackball”

Rotations

- Any rotation can be expressed as rotation about three axes
  - For example, \( x, y, z \) coordinate axes
- Any rotation can be represented (parameterized) by 3 numbers
- Rotations preserve
  - Angles
  - Lengths
  - Handedness
- Rigid transforms
  - Rotations and translations

Rotation matrices

- Orthonormal
  - Rows, columns are unit length and orthogonal
- Inverse of rotation matrix?
  - Its transpose
    
    \[
    R(\mathbf{a}, \theta)^{-1} = R(\mathbf{a}, \theta)^T
    \]

Questions?
### Rotations

- Given a rotation matrix \( \mathbf{R}(a, \theta) \)
- How do we obtain \( \mathbf{R}(a, -\theta) \)?

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

- How do we obtain \( \mathbf{R}(a, 2\theta), \mathbf{R}(a, 3\theta) \) ...?

### Rotating with pivot

- Given a rotation matrix \( \mathbf{R}(a, \theta) \)
- How do we obtain \( \mathbf{R}(a, -\theta) \)?

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

- How do we obtain \( \mathbf{R}(a, 2\theta), \mathbf{R}(a, 3\theta) \) ...?

\[
\begin{align*}
\mathbf{R}(a, 2\theta) &= \mathbf{R}(a, \theta)^2 = \mathbf{R}(a, \theta)\mathbf{R}(a, \theta) \\
\mathbf{R}(a, 3\theta) &= \mathbf{R}(a, \theta)^3 = \mathbf{R}(a, \theta)\mathbf{R}(a, \theta)\mathbf{R}(a, \theta)
\end{align*}
\]

### Today

- Homogeneous coordinates
- Affine transformations
- Concatenating transformations
- Change of coordinates
- Common coordinate systems

### Rotation around origin

Rotation with pivot

```
```
Rotating with pivot

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

$p' = T^{-1}RTp$

Concatenating transformations

- Arbitrary sequence of transformations
  \[ p' = M_3M_2M_1p \]
  \[ M_{total} = M_3M_2M_1 \]
  \[ p' = M_{total}p \]

- Note: associativity
  \[ M_{total} = (M_3M_2)M_1 = M_3(M_2M_1) \]

\[ T = M3.multiply(M2); M_{total} = T.multiply(M1) \]
\[ T = M2.multiply(M1); M_{total} = M3.multiply(T) \]

Summary affine transformations

- Scale, shear, rotation, reflection, translation
- Implemented using 4x4 matrices, homogeneous coordinates
  - Last row is always $[0 \ 0 \ 0 \ 1]$
- Any such matrix represents an affine transformation in 3D
- Factorization into scale, shear, rotation, etc. is non-trivial

Today

- 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

\[ p_{xyz} = p_x + p_y + p_z + o \]
**Change of coordinates**

**New uvwq coordinate system**

Goal: Find coordinates of \( p_{xyz} \) with respect to new \( uvwq \) coordinate system.

**Change of coordinates**

Same point \( p \) in 3D, expressed in new \( uvwq \) frame:

\[
p_{uvwq} = p_x + p_y + p_z + o_x + o_y + o_z + 1
\]

**Change of coordinates**

- Given coordinates of basis \( xyz \) with respect to new frame \( uvwq \):
- Coordinates of any point \( p_{xyz} \) with respect to new frame are:
- Matrix contains old basis vectors in new coordinates.

**Change of coordinates**

Coordinates of \( xyz \) frame w.r.t. \( uvwq \) frame:

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

**Change of coordinates**

Inverse transformation:

- Given point \( p_{uvw} \) w.r.t. frame \( u,v,w,q \)
- Coordinates \( p_{xyz} \) w.r.t. frame \( x,y,z,o \)

\[
p_{xyz} = \begin{bmatrix} x_u & y_u & z_u & o_u \\ x_v & y_v & z_v & o_v \\ x_w & y_w & z_w & o_w \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} p_u \\ p_v \\ p_w \\ 1 \end{bmatrix}
\]
Today
- Homogeneous coordinates
- Affine transformations
- Concatenating transformations
- Change of coordinates
- Common coordinate systems

Common coordinate systems
- Camera, world, and 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

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 matrix**

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

  \[
  z_c = \frac{e - d}{\|e - d\|}
  \]

  \[
  x_c = \frac{up \times z_c}{\|up \times z_c\|}
  \]

  \[
  y_c = z_c \times x_c
  \]
Camera matrix

\[
C = \begin{bmatrix}
x_c & y_c & z_c & e \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

- Think about: What does it mean to compute

\[
p' = Cp
\]

\[
q' = C^{-1}q
\]

Next time
- Given camera coordinates, how to compute image coordinates (perspective projection)
- Rendering