CMSC427
Transformations: Affine basics (translations, scaling)

Credit: slides 9+ from Prof. Zwicker

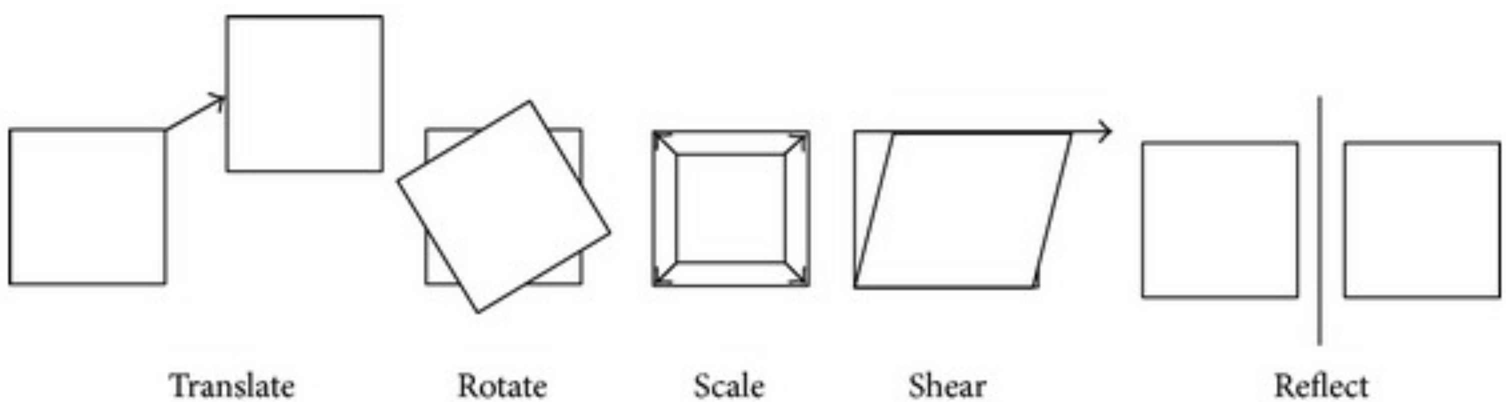
Today

Transformations & matrices

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

Classes of transformations

- Rigid
  - Translate, rotate, uniform scale
  - No distortion to object
- Affine
  - Translate, rotate, scale (non-uniform), shear, reflect
  - Limited distortions
  - Preserve parallel lines



First try: scale and rotate vertices in vector notation

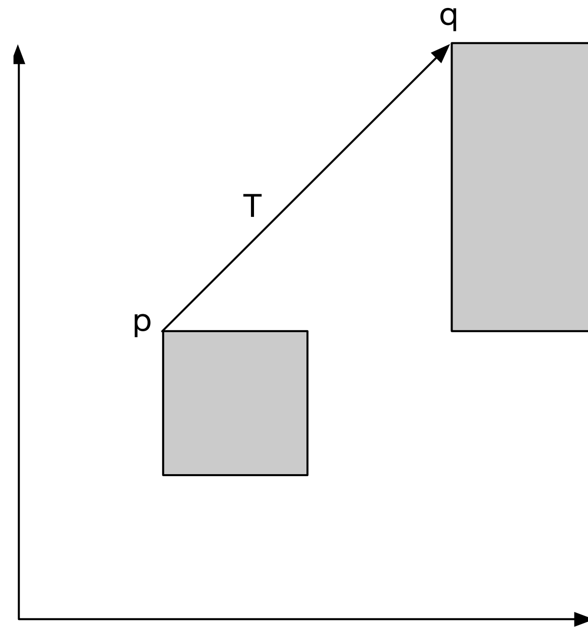
- Scale a point p by s and translate by T
- Vector multiplication and addition
- Repeat and we get

$$q = s_2(s * p + T) + T_2$$

- Gets unwieldy
- Instead – unify notation with homogeneous coordinates and matrices

$$q = s * p + T$$

$$q = 2 * (2,3) +< 2,2 >$$
$$q = (6,8)$$



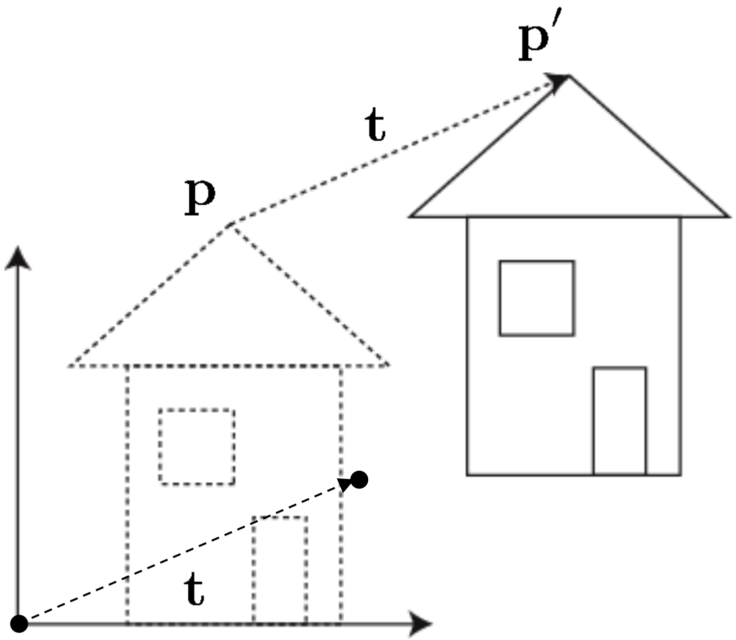
Affine transformations

- Transformation, or mapping: function that maps each 3D point to a new 3D point „f: $\mathbf{R}^3$ -> $\mathbf{R}^3$"
- Affine transformations: class of transformations to position 3D objects in space
- Affine transformations include
  - Rigid transformations
    - Rotation
    - Translation
  - Non-rigid transformations
    - Scaling
    - Shearing

Affine transformations

- Definition: mappings that preserve colinearity and ratios of distances
  http://en.wikipedia.org/wiki/Affine_transformation
  - Straight lines are preserved
  - Parallel lines are preserved
- Linear transformations + translation
- Nice: All desired transformations (translation, rotation) implemented using homogeneous coordinates and matrix-vector multiplication

Translation



Point

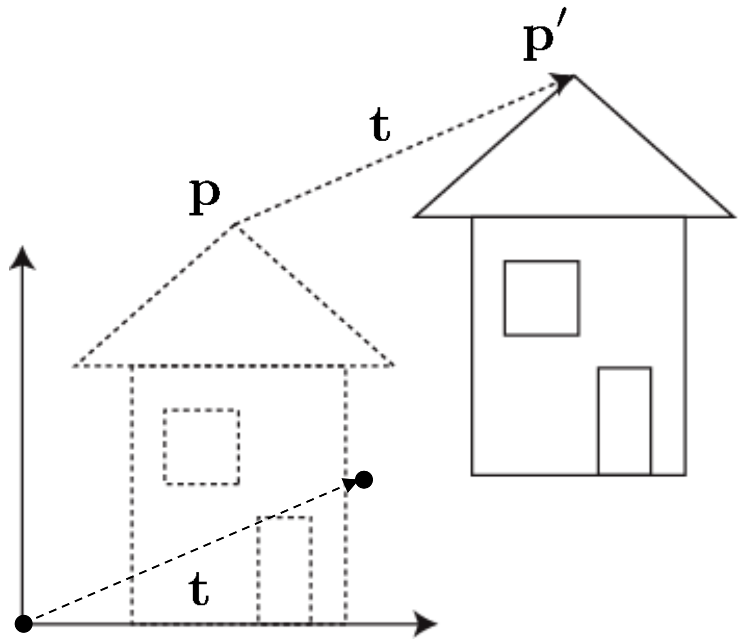
$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

Vector

$$\mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \\ 0 \end{bmatrix}$$

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

Matrix formulation



Point

$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

Vector

$$\mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \\ 0 \end{bmatrix}$$

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

$$\underbrace{\begin{bmatrix} p'_x \\ p'_y \\ p'_z \\ 1 \end{bmatrix}}_{\mathbf{p}'} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T}(\mathbf{t})} \underbrace{\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}}_{\mathbf{p}}$$

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

- Inverse translation

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

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

- Verify that

$$\mathbf{T}(-\mathbf{t})\mathbf{T}(\mathbf{t}) = \mathbf{T}(\mathbf{t})\mathbf{T}(-\mathbf{t}) = \mathbf{I}$$

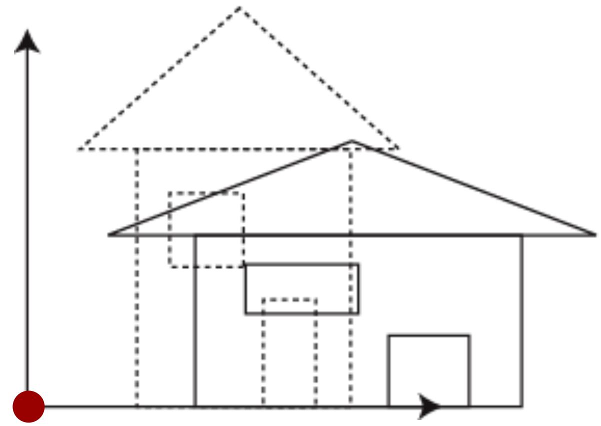
Note

- What happens when you translate a vector?

$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ 0 \end{bmatrix} = ?$$

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 scaling?

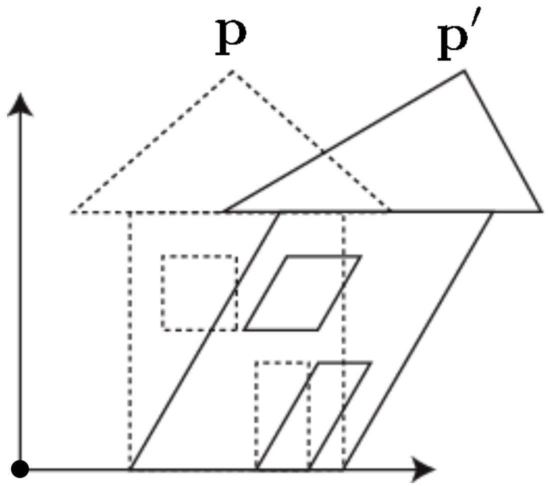
$$\mathbf{S}(s_x, s_y, s_z)^{-1} =$$

Scaling

- Inverse scaling?

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

Shear



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

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

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

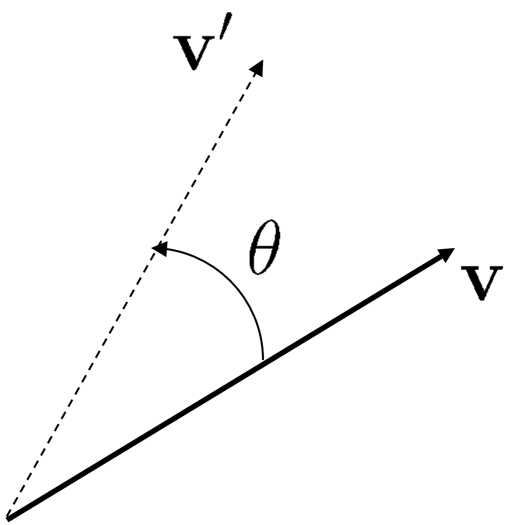
CMSC427
Transformations:
Affine rotations

Credit: slides 9+ from Prof. Zwicker

First: rotating a vector in 2D

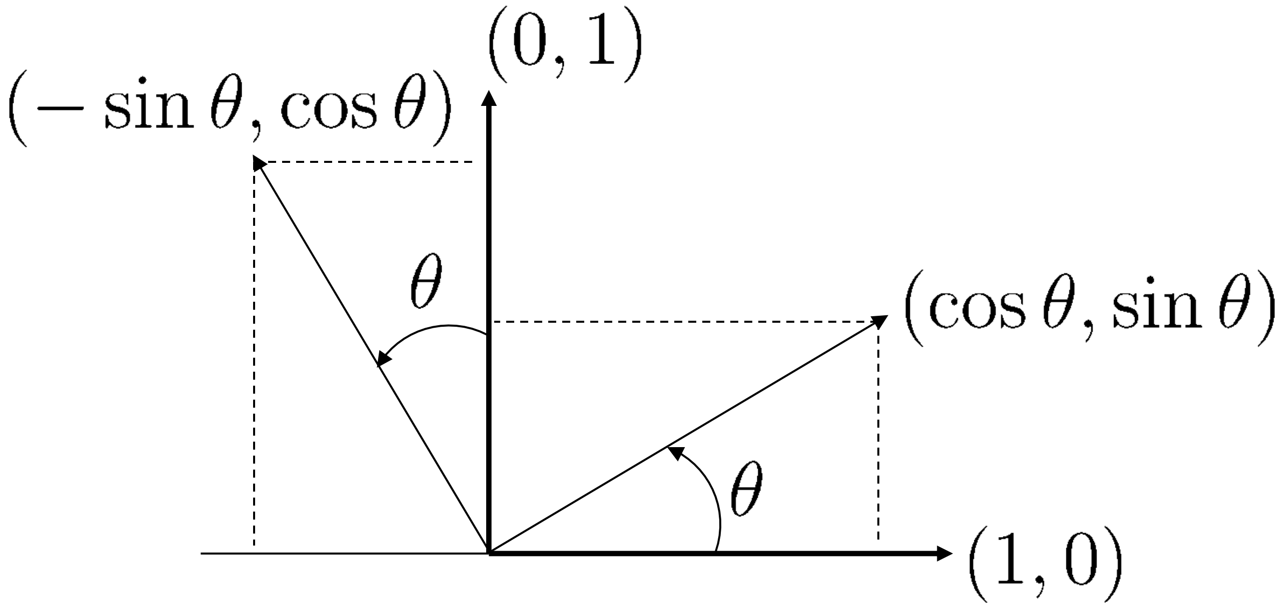
- Convention: positive angle rotates counterclockwise
- Express using rotation matrix

$$\mathbf{R}(\theta)$$

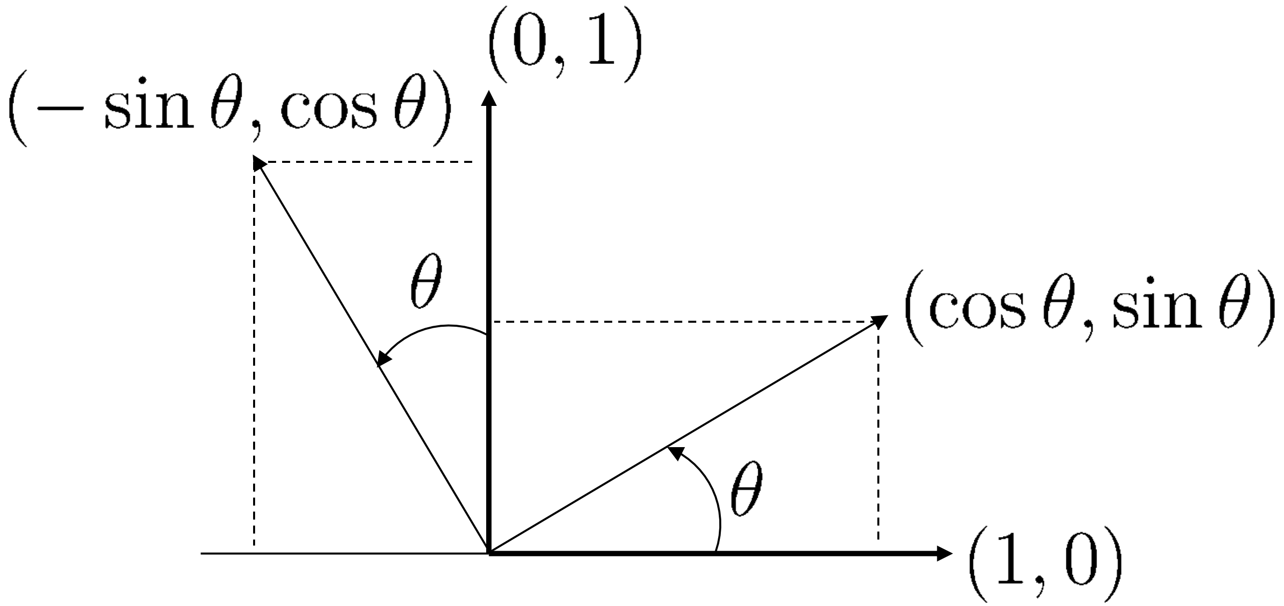


$$\mathbf{v}' = \mathbf{R}(\theta)\mathbf{v}$$

Rotating a vector in 2D



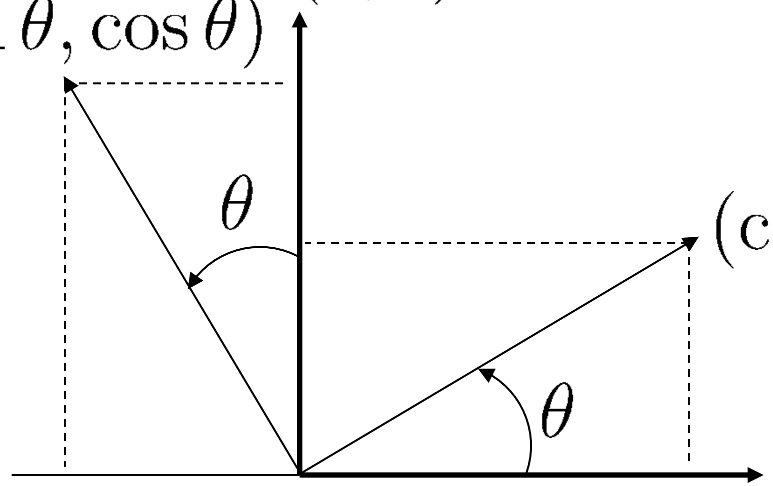
Rotating a vector in 2D



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

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

Rotating a vector in 2D



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

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

Rotation around z-axis

- z-coordinate does not change

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

$$v^0 = R_z(\mu)v$$

- What is the matrix for $\theta = 0, \theta = 90, \theta = 180$

$$\mathbf{R}_z(\theta)\mathbf{v} = \begin{bmatrix} \cos(\theta)v_x - \sin(\theta)v_y \\ \sin(\theta)v_x + \cos(\theta)v_y \\ v_z \\ 1 \end{bmatrix}$$

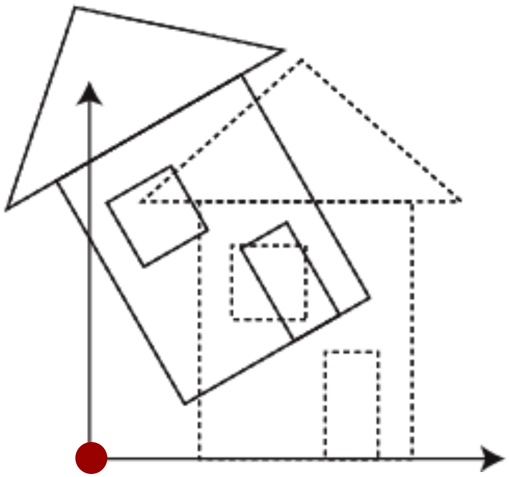
Other coordinate axes

- Same matrix to rotate points and vectors
- Points are rotated around origin

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

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

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

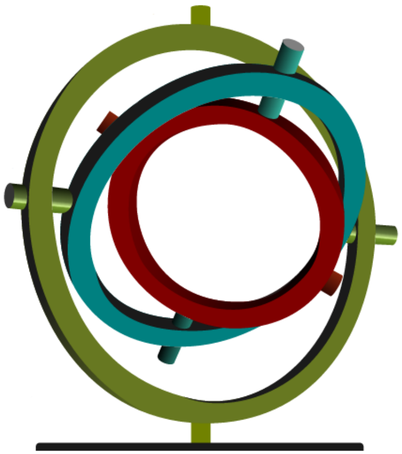


Rotation in 3D

- Concatenate rotations around *x,y,z* axes to obtain rotation around arbitrary axes through origin

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

- $\theta_x, \theta_y, \theta_z$ are called Euler angles
  http://en.wikipedia.org/wiki/Euler_angles
- Disadvantage: result depends on order!



Gimbal

https://en.wikipedia.org/wiki/Gimbal

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

Rotation around arbitrary axis

- Still: origin does not change
- Counterclockwise rotation
- Angle θ, unit axis $\mathbf{a}$
-

$$c_\theta = \cos\theta, s_\theta = \sin\theta$$

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

Summary

- Different ways to describe rotations mathematically
  - Sequence of rotations around three axes (Euler angles)
  - Rotation around arbitrary angles (axis-angle representation)
  - Better internal representation (quaternions)
- Rotations preserve
  - Angles
  - Lengths
  - Handedness of coordinate system
- Rigid transforms
  - Rotations and translations

Rotation matrices

- Orthonormal
  - Rows, columns are unit length and orthogonal
- Inverse of rotation matrix?

Rotation matrices

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

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

Rotations

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

Rotations

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

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

Rotations

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

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

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

Rotations

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

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

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

$$\mathbf{R}(\mathbf{a}, 2\theta) = \mathbf{R}(\mathbf{a}, \theta)^{2} = \mathbf{R}(\mathbf{a}, \theta)\mathbf{R}(\mathbf{a}, \theta)$$

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

Summary affine transformations

- Linear transformations (rotation, scale, shear, reflection) + translation

Vector space,
http://en.wikipedia.org/wiki/Vector_space

- vectors as [xyz] coordinates
- represents vectors
- linear transformations

Affine space
http://en.wikipedia.org/wiki/Affine_space

- points and vectors as [xyz1], [xyz0] homogeneous coordinates
- distinguishes points and vectors
- linear tranforms and translation

Summary affine transformations

- Implemented using 4x4 matrices, homogeneous coordinates
  - Last row of 4x4 matrix is always $[0\ 0\ 0\ 1]$
- Any such matrix represents an affine transformation in 3D
- Factorization into scale, shear, rotation, etc. is always possible, but non-trivial
  - Polar decomposition
    http://en.wikipedia.org/wiki/Polar_decomposition