Composing 2D Affine Transformations

Matrix multiplication - Just Works

Example: Rotate around a non-origin point \( P \)

\[
T_1 = \text{translate}(-P)
\]

\[
T_2 = \text{rot}(45^\circ)
\]

\[
T_3 = \text{translate}(P)
\]

\[
T = T_3 T_2 T_1
\]

right-to-left!

Points vs Directions

Both \( n \)-vectors, but one is a place, the other is a direction.

The real \( \vec{n} \)

Plane through pt \( a \) w/ normal vector \( \vec{n} \)

Plants get moved by translations

Directions don't

\[
\vec{V} = \vec{P} - \vec{Q}
\]

\[
\begin{bmatrix}
1 & 0 & \alpha_x + 1 \\
0 & 1 & \alpha_y + 1 \\
0 & 0 & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
0 & 1 & \beta_x \\
1 & 0 & \beta_y \\
0 & 0 & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
0 & 0 & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
N_x \\
N_y \\
0
\end{bmatrix}
\]

\[
\begin{bmatrix}
\vec{N_x} \\
\vec{N_y} \\
\vec{0}
\end{bmatrix}
\]

Turn off translation for vectors that indicate directions
Affine Composition - Similarity Transforms:

1. Transform to a frame where it's easy
2. Do the easy thing
3. Transform back.

Let $F = \begin{bmatrix} \tilde{u} & \tilde{v} & \tilde{p} \\ 0 & 0 & 1 \end{bmatrix}$, frame where desired transformation $T$ is easy to write.

This is frame-to-canonical!

Canonical-to-frame is $F^{-1}$, so whole transform is $F^{-1}T^{-1}$.

Inverses?

Can compute algebraically, but sometimes it's easy:

- $\text{translate}(\tilde{z})^{-1} = \text{translate}(-\tilde{z})$
- $\text{rotate}(\theta)^{-1} = \text{rotate}(-\theta)$
- $\text{scale}(s_x, s_y)^{-1} = \text{scale}(1/s_x, 1/s_y)$

Inverse of frame w/ orthonormal basis (rigid + $t$)

$$\begin{bmatrix} \tilde{q} & \tilde{u} \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} q^T & -q^T\tilde{u} \\ 0 & 1 \end{bmatrix}$$

(Math Fact)

Inverse of an orthogonal matrix is its transpose.
Rotations in 3D

Euler angles - composition of 3 axis-aligned rotations.

Example \(\mathbf{R}_z \mathbf{R}_x \mathbf{R}_z\)

- 12 possible sequences of 3 axis rotations!
- Also: can choose intrinsic vs extrinsic

Axes of rotation stick with object as it rotates
Axes of rotation are fixed to the global coordinate system

Yikes! See Wikipedia. Pick a convention and stick with it.

Axis-angle

\[ \mathbf{a} \rightarrow \text{Rotate } \theta \text{ about axis } \mathbf{a} \]

How? Change coordinate frame so \(\mathbf{a}\) is the "z" axis. Similarity transform!

Rotate around \(\mathbf{z}\), change coordinate frame back.

Basis for frame: \(\mathbf{w} \perp \mathbf{a}\)

Pick a \(\mathbf{z}\) not collinear with \(\mathbf{w}\)

\[ \mathbf{z} = \frac{\mathbf{w} \times \mathbf{u}}{||\mathbf{w} \times \mathbf{u}||} \]

\[ \mathbf{u} = \frac{\mathbf{z} \times \mathbf{w}}{||\mathbf{z} \times \mathbf{w}||} \]

\[ \mathbf{v} = \mathbf{w} \times \mathbf{u} \]