3D Transformations

CS 465 Lecture 9
Translation

\[
\begin{bmatrix}
x' \\
y' \\
z'
\end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}
\]
Scaling

\[
\begin{bmatrix}
  x' \\
  y' \\
  z' \\
  1
\end{bmatrix} =
\begin{bmatrix}
  s_x & 0 & 0 & 0 \\
  0 & s_y & 0 & 0 \\
  0 & 0 & s_z & 0 \\
  0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z \\
  1
\end{bmatrix}
\]

P' = (x', y', z')
Rotation about $z$ axis

\[
\begin{bmatrix}
  x' \\
  y' \\
  z' \\
  1
\end{bmatrix} =
\begin{bmatrix}
  \cos \theta & -\sin \theta & 0 & 0 \\
  \sin \theta & \cos \theta & 0 & 0 \\
  0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z \\
  1
\end{bmatrix}
\]
Rotation about $x$ axis

\[
\begin{bmatrix}
  x' \\
  y' \\
  z' \\
  1
\end{bmatrix}
= \begin{bmatrix}
  1 & 0 & 0 & 0 \\
  0 & \cos \theta & -\sin \theta & 0 \\
  0 & \sin \theta & \cos \theta & 0 \\
  0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z \\
  1
\end{bmatrix}
\]
Rotation about y axis

\[
\begin{bmatrix}
  x' \\
  y' \\
  z' \\
  1
\end{bmatrix}
= \begin{bmatrix}
  \cos \theta & 0 & \sin \theta & 0 \\
  0 & 1 & 0 & 0 \\
  -\sin \theta & 0 & \cos \theta & 0 \\
  0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z \\
  1
\end{bmatrix}
\]
General rotations

- A rotation in 2D is around a point
- A rotation in 3D is around an axis
  - so 3D rotation is w.r.t an orientation as well as a position
- Compute by composing elementary transforms
  - transform rotation axis to align with x axis
  - apply rotation
  - inverse transform back into position
- Just as in 2D this can be interpreted as a similarity transform
Building general rotations

• Using elementary transforms you need three
  – translate axis to pass through origin
  – rotate about $y$ to get into $x-y$ plane
  – rotate about $z$ to align with $x$ axis

• Alternative: construct frame and change coordinates
  – choose $p, u, v, w$ to be orthonormal frame with $p$ and $u$
    matching the rotation axis
  – apply similarity transform $T = F R_x(\theta) F^{-1}$
Orthonormal frames in 3D

• Useful tools for constructing transformations
• Recall rigid motions
  – affine transforms with pure rotation
  – columns (and rows) form right handed ONB
    • that is, an orthonormal basis

\[
F = \begin{bmatrix}
  u & v & w & p \\
  0 & 0 & 0 & 1 
\end{bmatrix}
\]
Building 3D frames

• Given a vector \( \mathbf{a} \) and a secondary vector \( \mathbf{b} \)
  
  – The \( \mathbf{u} \) axis should be parallel to \( \mathbf{a} \); the \( \mathbf{u}-\mathbf{v} \) plane should contain \( \mathbf{b} \)
    
    • \( \mathbf{u} = \mathbf{u} / \|\mathbf{u}\| \)
    • \( \mathbf{w} = \mathbf{u} \times \mathbf{b}; \mathbf{w} = \mathbf{w} / \|\mathbf{w}\| \)
    • \( \mathbf{v} = \mathbf{w} \times \mathbf{u} \)

• Given just a vector \( \mathbf{a} \)
  
  – The \( \mathbf{u} \) axis should be parallel to \( \mathbf{a} \); don’t care about orientation about that axis
    
    • Same process but choose arbitrary \( \mathbf{b} \) first
    • Good choice is not near \( \mathbf{a} \): e.g. set smallest entry to 1
Building general rotations

• Alternative: construct frame and change coordinates
  – choose \( p \), \( u \), \( v \), \( w \) to be orthonormal frame with \( p \) and \( u \) matching the rotation axis
  – apply similarity transform \( T = FR_x(\theta) F^{-1} \)
  – interpretation: move to \( x \) axis, rotate, move back
  – interpretation: rewrite \( u \)-axis rotation in new coordinates
  – (each is equally valid)
Transforming normal vectors

- Transforming surface normals
  - differences of points (and therefore tangents) transform OK
  - normals do not

\[ \mathbf{t} \cdot \mathbf{n} = \mathbf{t}^T \mathbf{n} = 0 \]
want: \[ M\mathbf{t} \cdot X \mathbf{n} = \mathbf{t}^T M^T X \mathbf{n} = 0 \]
so set \[ X = (M^T)^{-1} \]
them: \[ M\mathbf{t} \cdot X \mathbf{n} = \mathbf{t}^T M^T (M^T)^{-1} \mathbf{n} = \mathbf{t}^T \mathbf{n} = 0 \]