Undoing Transformations: Inverses

- \( T^{-1}(x, y, z) = T(-x, -y, -z) \)
- \( R(z, \theta)^{-1} \) (\( R \) is orthogonal)
- \( S(s_x, s_y, s_z)^{-1} = S(1/s_x, 1/s_y, 1/s_z) \)

Readings for Transformations 1-5

- Shirley/Marschner
  - Ch 6: Transformation Matrices (avoids 6.1.6, 6.3.1)
  - Sect 12.2 Scene Graphs
- Gortler
  - Ch 2: Linear, Sec 2.5-2.6
  - Ch 3: Affine
  - Ch 4: Respect
  - Ch 5: Frames in Graphics, 5.3-5.4

Composing Transformations

- translation
  - \( T_1 = T(x_1, y_1, z_1) \)
  - \( T_2 = T(x_2, y_2, z_2) \)
  - \( T_1 \cdot T_2 = T(x_1 + x_2, y_1 + y_2, z_1 + z_2) \)

Composing Transformations

- scaling
  - \( T_1 \cdot T_2 = T(ax_1, ay_1, az_1) \) so scales multiply

Composing Transformations

- rotation
  - \( T_1 \cdot T_2 = T(ax_1, ay_1, az_1) \) so rotations add

Homogeneous Coordinates Review

- point in 2D cartesian + weight \( w = \) point \( P \) in 3D homog. coords
- multiples of \( (x,y,w) \)
- form a line \( L \) in 3D
- all homogeneous points on \( L \) represent same 2D cartesian point
- example: \((2,2,1) = (4,4,2) = (11,10,5)\)
Composing Transformations

\[ T_a T_b = T_b T_a \], but \( R_a R_b \neq R_b R_a \) and \( T_a R_b \neq R_b T_a \)

- translations commute
- rotations around same axis commute
- rotations around different axes do not commute
- rotations and translations do not commute

Correction: Composing Transformations

- which direction to read?
- left to right
  - interpret operations wrt fixed coordinates
  - moving object
- left to right
  - interpret operations wrt local coordinates
  - changing coordinate system
- in GL, cannot move object once it is drawn!!
  - object specified as set of coordinates wrt specific coord sys

Interpreting Transformations

translate by (-1.0)

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

same relative position between object and basis vectors

Matrix Composition

- matrices are convenient, efficient way to represent series of transformations
- general purpose representation
- hardware matrix multiply
- matrix multiplication is associative
- \( p'' = (T(R(S(p)))) \)
- \( p'' = (T(R(S(p)))) \)
- procedure
  - correctly order your matrices!
  - multiply matrices together
  - result is one matrix, multiply vertices by this matrix
  - all vertices easily transformed with one matrix multiply