Readings for Transformations 1-5
- Shirley/Marschner
  - Ch 6: Transformation Matrices
  - 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

Correction: Composing Transformations
\[ p' = T \cdot R \cdot p \]
- \( p' \) is the new position
- \( T \) is the translation
- \( R \) is the rotation
- \( p \) is the original position

Correction: Composing Transformations
- which direction to read?
  - left to right
- interpret operations wrt local coordinates
- changing coordinate system
  - translate coordinate system (2, 3) over
  - rotate coordinate system 45 degrees wrt LOCAL origin
  - draw object in current coordinate system

Rotation About an Arbitrary Axis
- axis defined by two points
- translate point to the origin
- rotate to align axis with z-axis (or x or y)
- perform rotation
- undo aligning rotations
- undo translation

Transformation Hierarchies
- scene may have a hierarchy of coordinate systems
- stores matrix at each level with incremental transformation
- change coordinate system
  - translate coordinate system to origin
  - rotate coordinate system
  - draw object in current coordinate system

General Transform Composition
- transformation of geometry into coordinate system
  - typically translate to origin
  - perform operation
  - transform geometry back to original coordinate system

Transformation Hierarchy Example 1
- world
  - torso
    - head
    - arms
      - legs
        - feet
Transformation Hierarchy Example 2

- draw same 3D data with different transformations: instancing

Modularization

- drawing a scaled square
- push/pop ensures no coord system change

```cpp
void drawBlock(float k) {
  pushMatrix();
  scale(k, k, k);
  drawBox();
  popMatrix();
}
```

Hierarchical Modelling

- advantages
  - no need to compute inverse matrices all the time
  - modularize changes to pipeline state
  - avoids incremental changes to coordinate systems
  - accumulation of numerical errors

Arbitrary Rotation

- arbitrary rotation: change of basis
- given two orthonormal coordinate systems $XYZ$ and $ABC$
- $A$'s location in the $XYZ$ coordinate system is $(ax, ay, az, 1)$, ...
- $B$'s location in the $XYZ$ coordinate system is $(bx, by, bz, 1)$, ...
- $C$'s location in the $XYZ$ coordinate system is $(cx, cy, cz, 1)$, ...

Transformation Hierarchy Example 3

- challenge of avoiding unnecessary computation
  - using inverse to return to origin
  - computing incremental $T_1 \rightarrow T_2$

Matrix Stacks

- advantages
  - no need to compute inverse matrices all the time
  - modularize changes to pipeline state
  - avoids incremental changes to coordinate systems
  - accumulation of numerical errors

Translation Hierarchy Example 4

- arbitrary rotation: change of basis
- given two orthonormal coordinate systems $XYZ$ and $ABC$
- $A$'s location in the $XYZ$ coordinate system is $(a_x, a_y, a_z, 1)$, ...
- $B$'s location in the $XYZ$ coordinate system is $(b_x, b_y, b_z, 1)$, ...
- $C$'s location in the $XYZ$ coordinate system is $(c_x, c_y, c_z, 1)$, ...

Matrix Stacks

- advantages
  - no need to compute inverse matrices all the time
  - modularize changes to pipeline state
  - avoids incremental changes to coordinate systems
  - accumulation of numerical errors

Hierarchical Modelling

- advantages
  - define object once, instantiate multiple copies
  - transformation parameters often good control knobs
  - maintain structural constraints if well-designed
- limitations
  - expressivity: not always the best controls
  - can't do closed kinematic chains
    - keep hand on hip
    - can't do other constraints
    - collision detection
    - walk through walls

Arbitrary Rotation

- arbitrary rotation: change of basis
  - given two orthonormal coordinate systems $XYZ$ and $ABC$
  - $A$'s location in the $XYZ$ coordinate system is $(a_x, a_y, a_z, 1)$, ...
  - $B$'s location in the $XYZ$ coordinate system is $(b_x, b_y, b_z, 1)$, ...
  - $C$'s location in the $XYZ$ coordinate system is $(c_x, c_y, c_z, 1)$, ...

Arbitrary Rotation

- arbitrary rotation: change of basis
  - given two orthonormal coordinate systems $XYZ$ and $ABC$
  - $A$'s location in the $XYZ$ coordinate system is $(a_x, a_y, a_z, 1)$, ...
  - $B$'s location in the $XYZ$ coordinate system is $(b_x, b_y, b_z, 1)$, ...
  - $C$'s location in the $XYZ$ coordinate system is $(c_x, c_y, c_z, 1)$, ...

Arbitrary Rotation

- arbitrary rotation: change of basis
  - given two orthonormal coordinate systems $XYZ$ and $ABC$
  - $A$'s location in the $XYZ$ coordinate system is $(a_x, a_y, a_z, 1)$, ...
  - $B$'s location in the $XYZ$ coordinate system is $(b_x, b_y, b_z, 1)$, ...
  - $C$'s location in the $XYZ$ coordinate system is $(c_x, c_y, c_z, 1)$, ...

Matrix Stacks

- advantages
  - no need to compute inverse matrices all the time
  - modularize changes to pipeline state
  - avoids incremental changes to coordinate systems
  - accumulation of numerical errors