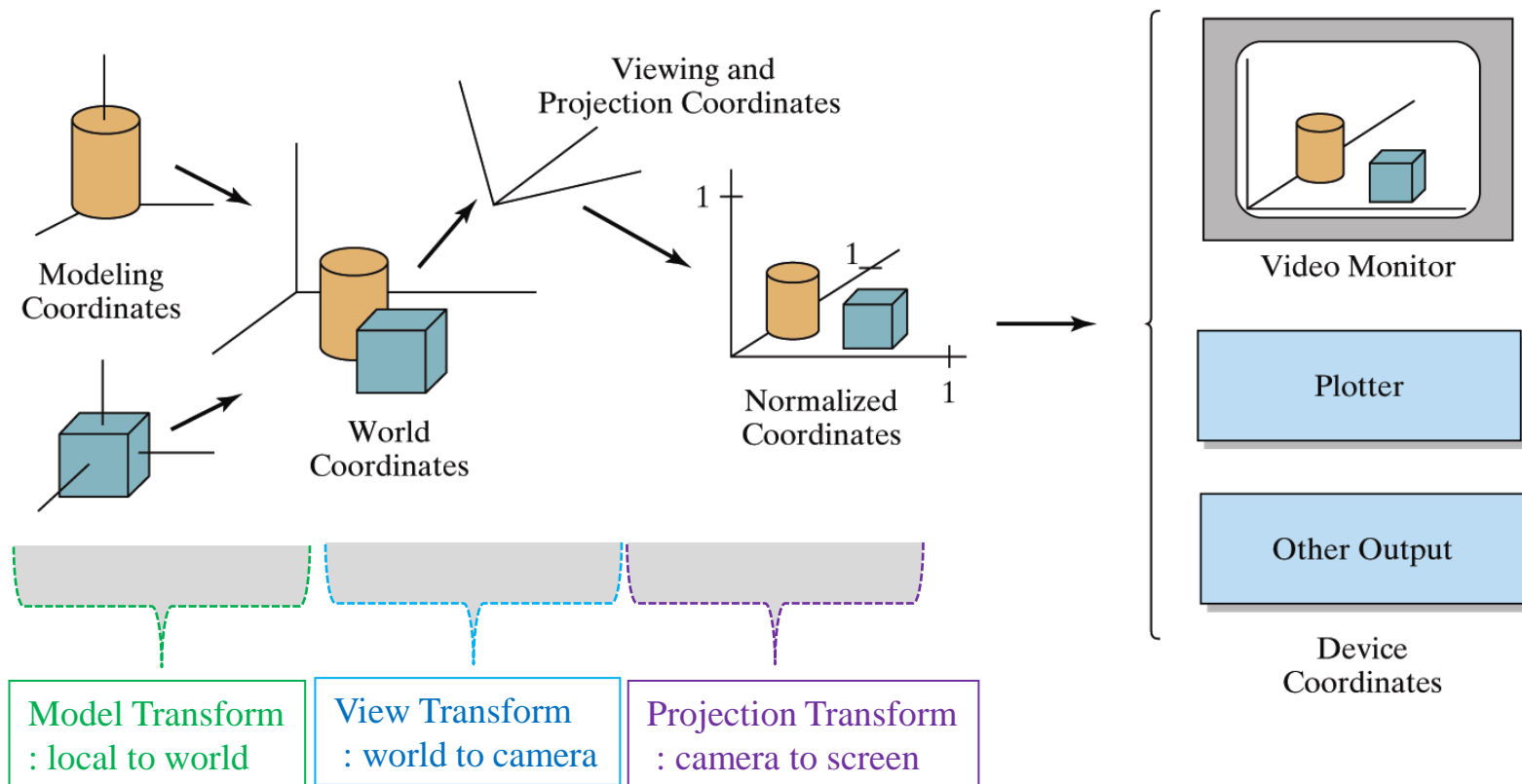

Model View Matrix and its implementation

Sang Il Park

Department of Software
Sejong University

OpenGL Geometric Transformations

- **Consecutive Transformations in OpenGL Pipeline**

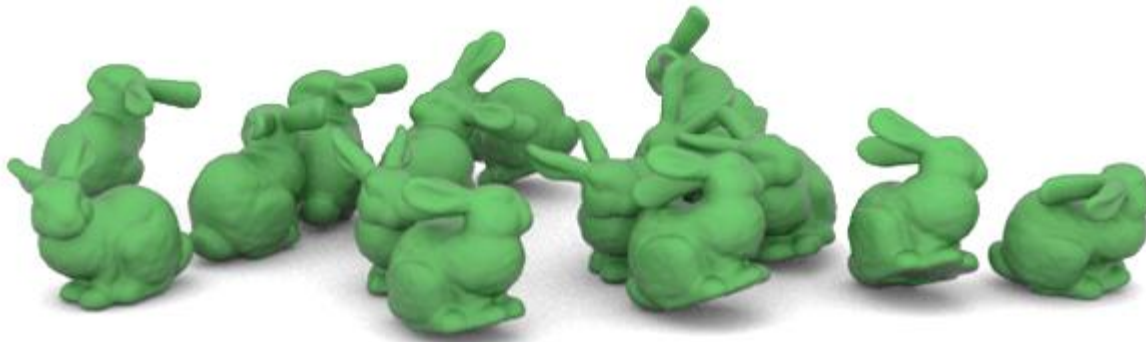


Topics

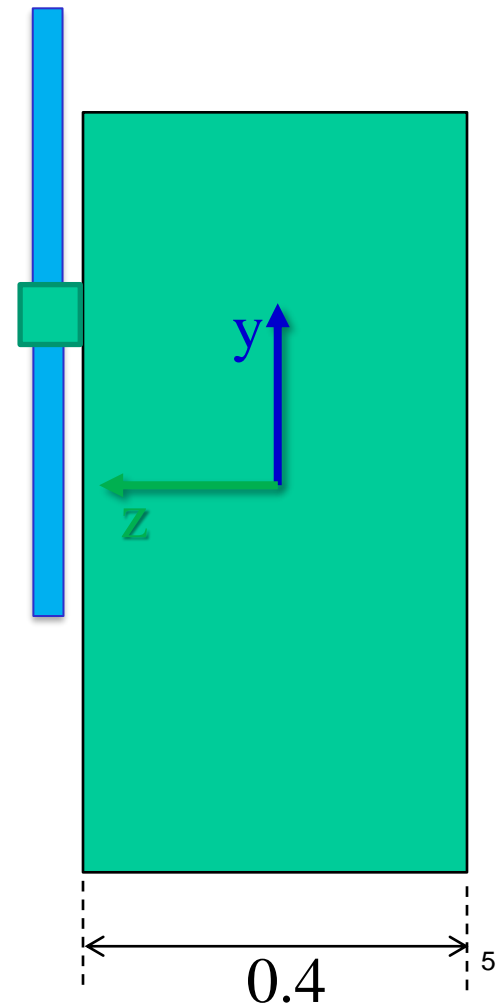
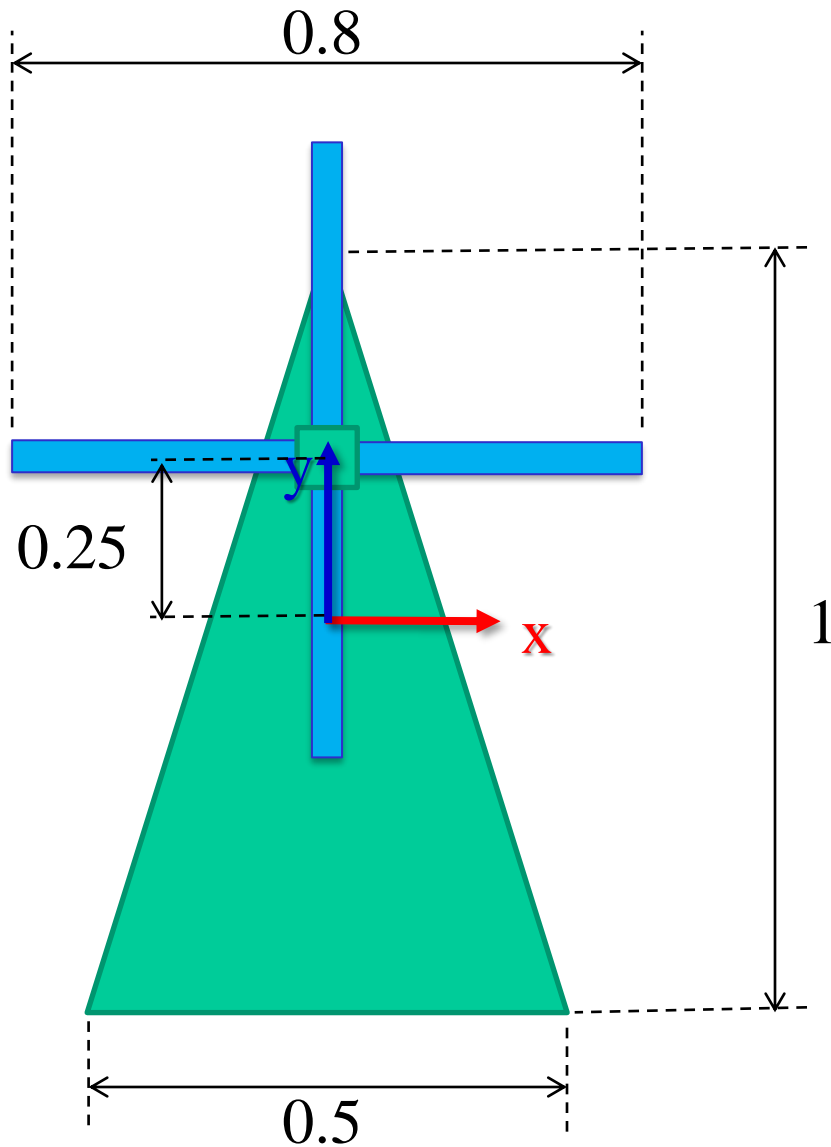
- Local to World Coordinate Transform
- Camera Positioning

Instance Transformation

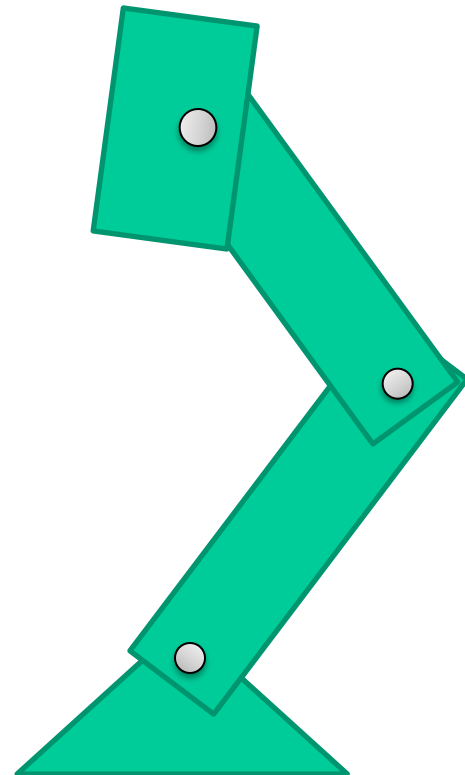
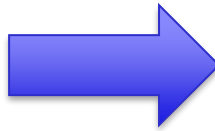
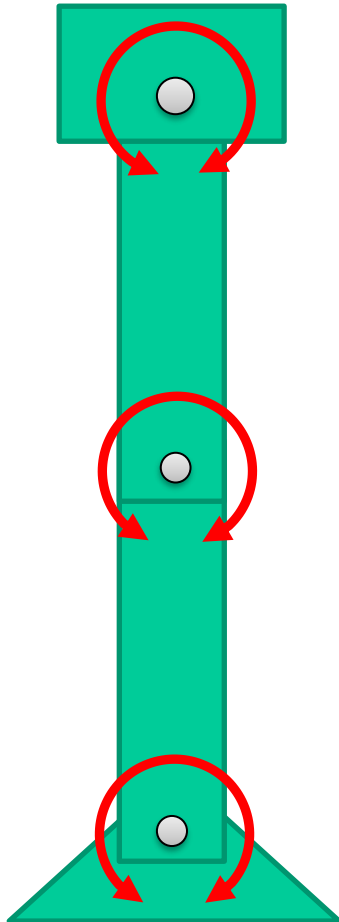
- Instances can be shared across space or time
- Write a function that renders the object in “standard” configuration
- Apply transformations to different instances
- Typical order: *scaling* → *rotation* → *translation*



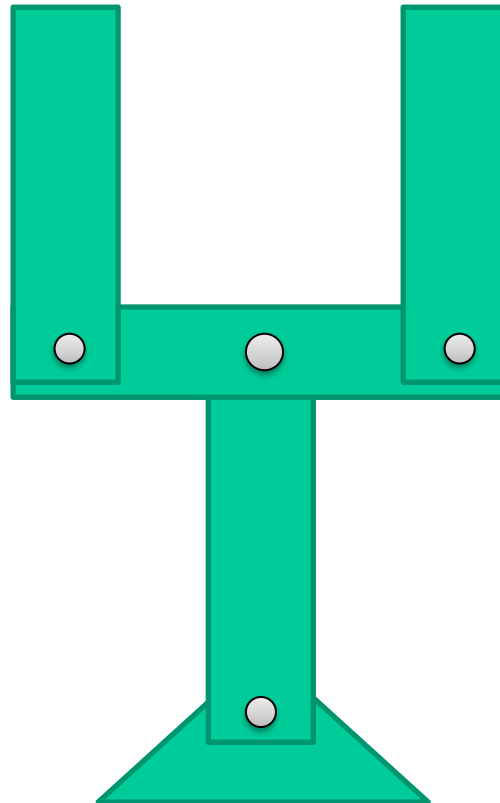
풍차 만들기



로봇 팔 만들기



로봇 팔 만들기2



Hierarchical Modeling

- A hierarchical model is created by nesting the descriptions of subparts into one another to form a tree organization

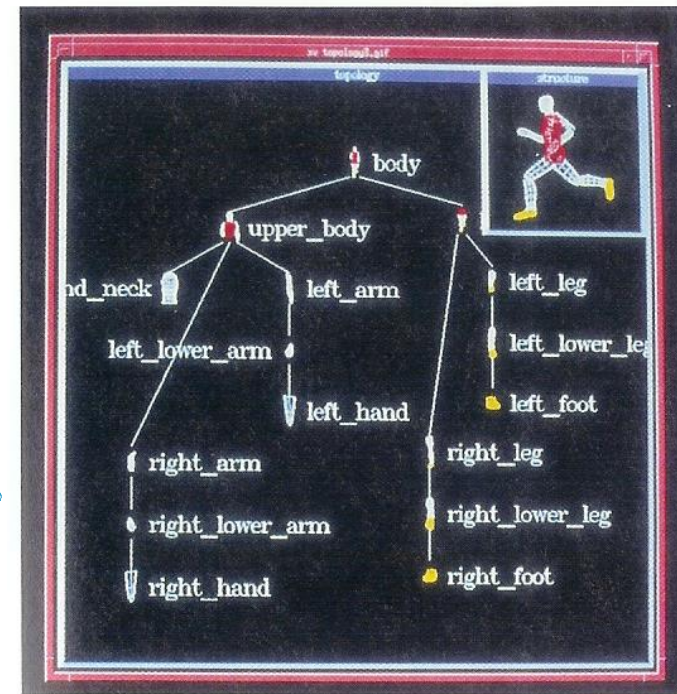
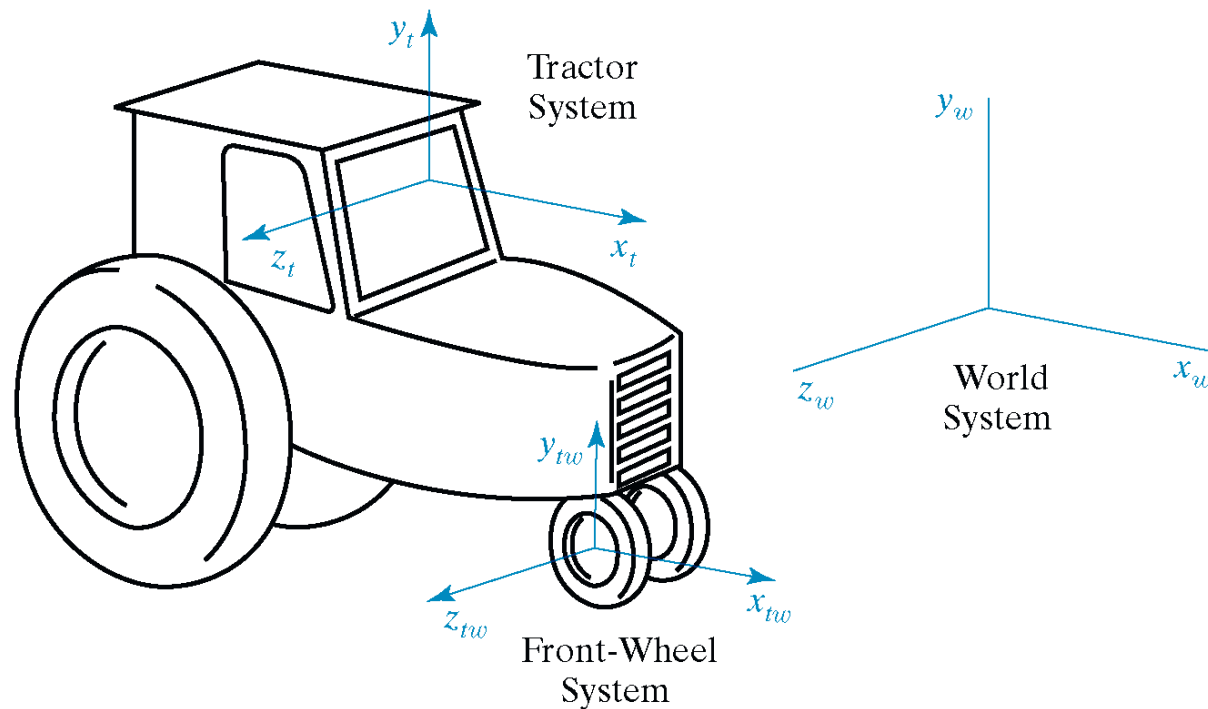


FIGURE 14-4 An object hierarchy generated using the PHIGS Toolkit package developed at the University of Manchester. The displayed object tree is itself a PHIGS structure. (Courtesy of T. L. J. Howard, J. G. Williams, and W. T. Hewitt, Department of Computer Science, University of Manchester, United Kingdom.)

OpenGL Matrix Stacks (OLD)

- Stack processing

- The top of the stack is the “current” matrix

- **glPushMatrix () ;** // Duplicate the current matrix at the top

- **glPopMatrix () ;** // Remove the matrix at the top

Matrix Stacks by your own

- We emulate Matrix Stacks by using:
 - Linked List such as *std::list* or *std::deque*
 - Or a tree structure for more generality.