Basic Objects for CG: Platonic Solids

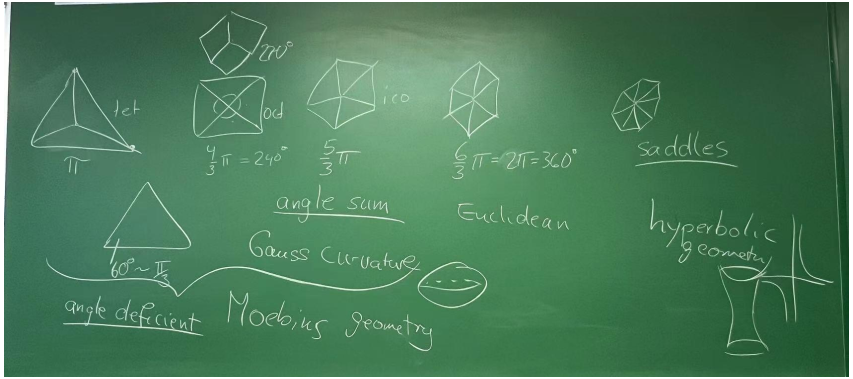
Computer Graphics Jorg Peters

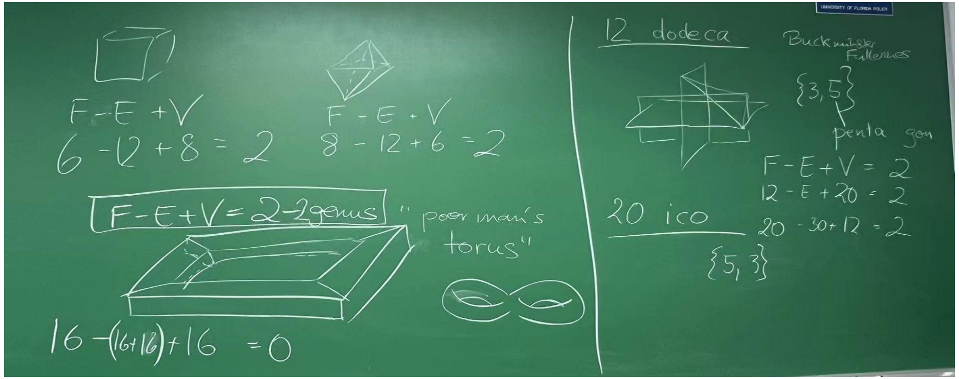
- 5 Platonic solids:

  Tetrahedron, hexahedron, octahedron, dodecahedron, icosahedron,

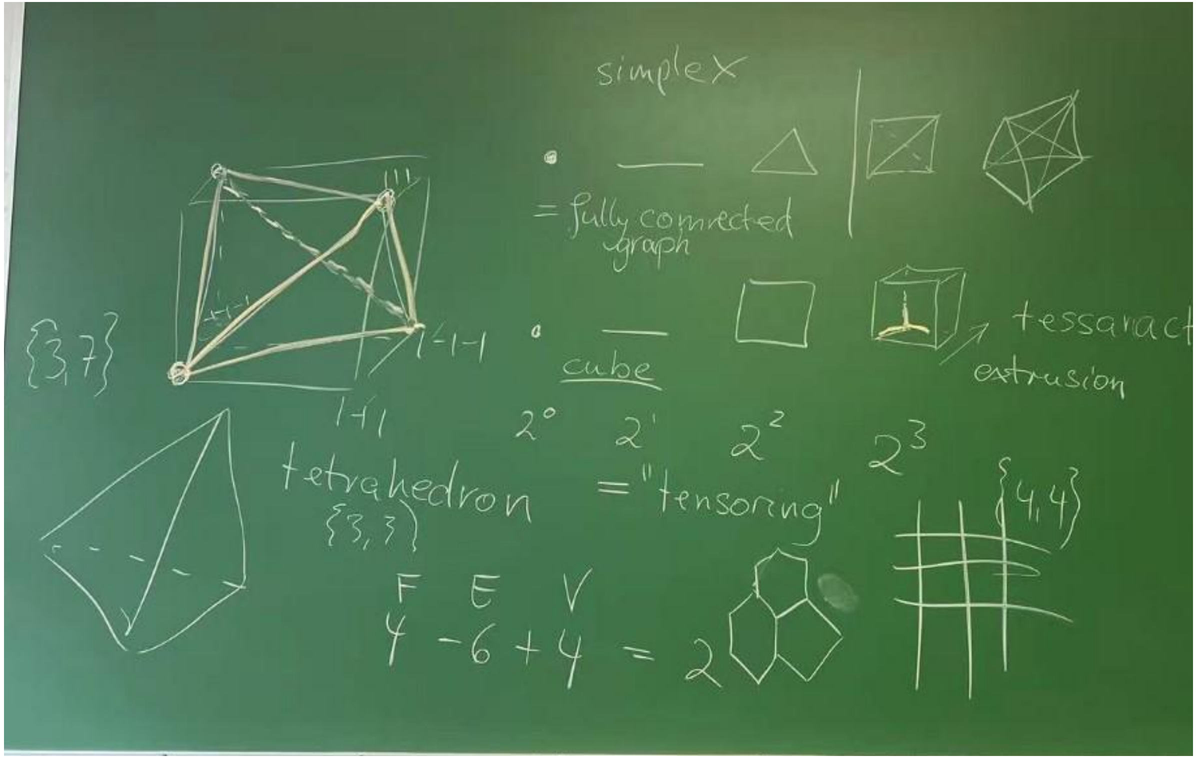
  Good choice for vertex coordinates: start with cube with vertices (±1, ±1, ±1)

  Euler's formula: v-e+f = 2 - genus





Basic Objects for CG: simplex and tensor-product

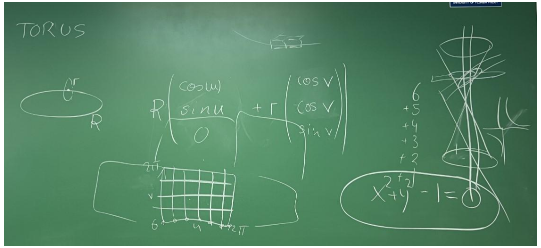


Basic Objects for CG

- Torus and sphere

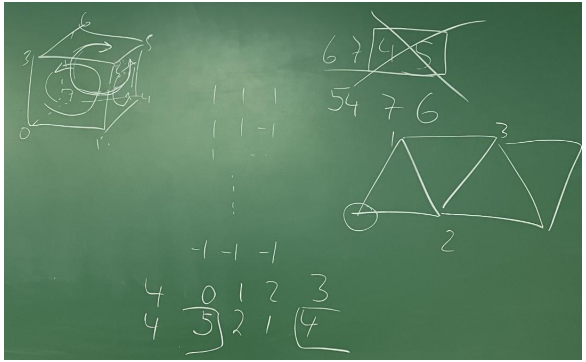
$$
\begin{aligned}
x(\theta,\varphi) &= (R + r\cos\theta)\cos\varphi \\
y(\theta,\varphi) &= (R + r\cos\theta)\sin\varphi \\
z(\theta,\varphi) &= r\sin\theta
\end{aligned}
$$

$$\theta,\varphi \in [0, 2\pi)$$



- Conics and quadrics

Data Structures and File Formats



- Connectivity(topology)
- Attributes (position, normal,color)

File Formats: e.g. .off (Object File Format)

Data Structures: Half-Edge

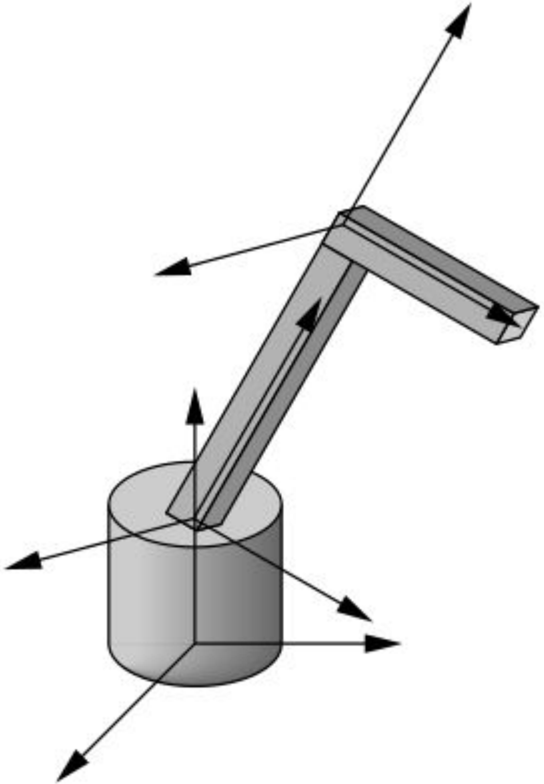
Hierarchical & Scene Graphs

Robot Arm transformations:

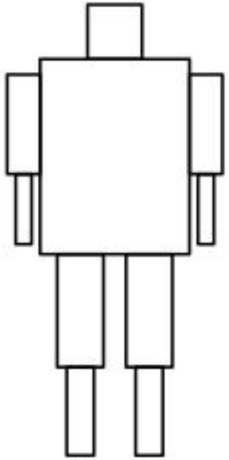
R (Base) {

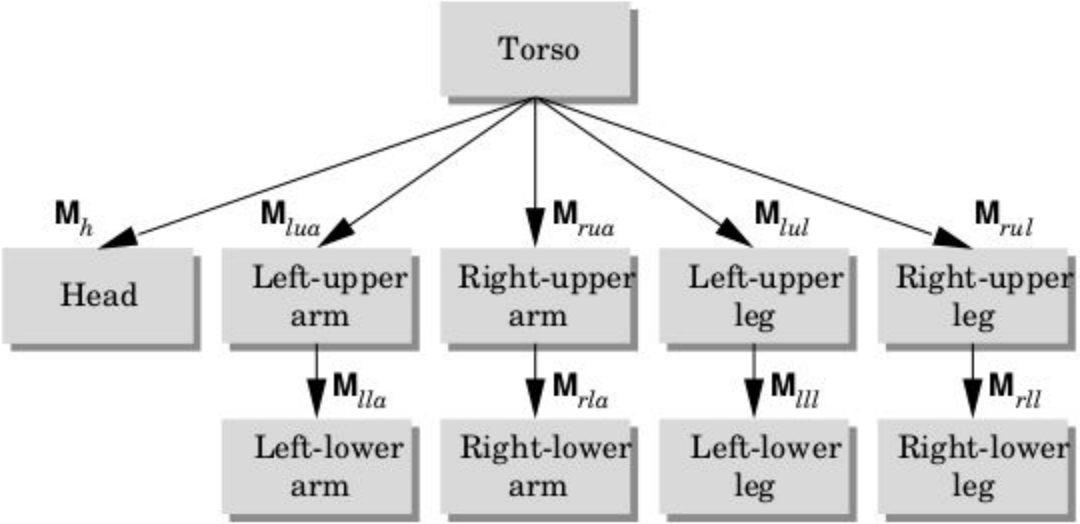
{ T R (lower arm)

{ T R (upper arm) } }

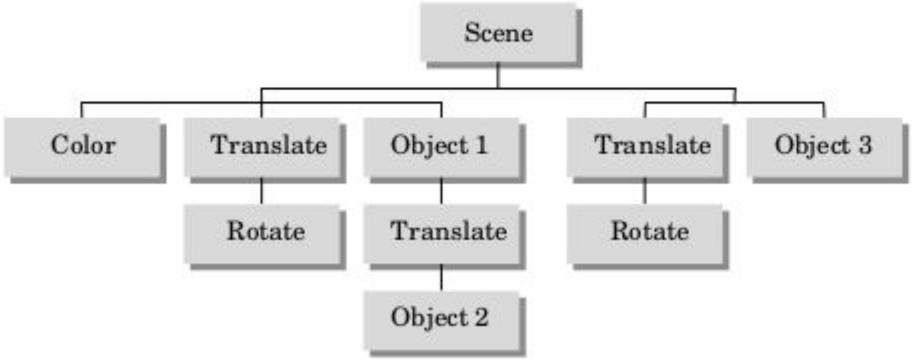


Hierarchical & Scene Graphs





Hierarchical & Scene Graphs



DAG = directed acyclic graph

depth first traversal: left child, right sibling

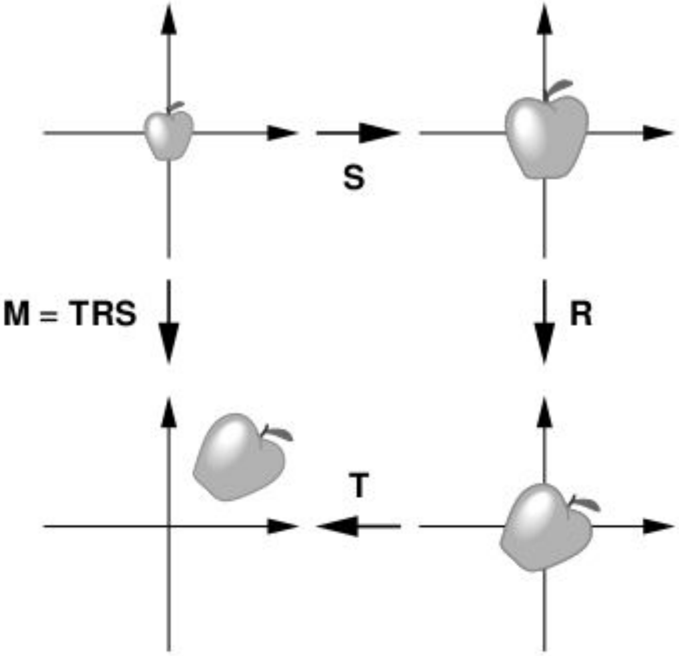
Data structure:

```
typedef struct treenode {
        GLfloat m[16]; // transformation
        void (*f)();  // figure
        struct treenode *sibling;
        struct treenode *child;
} treenode;
```

traverse:

```
glMultMatrixf(root->m);
root->f();
if(root->child!=NULL) traverse(root->child);
if(root->sibling!=NULL) traverse(root->sibling);
```

Hierarchical & Scene Graphs



$Translate * Rotate * Scale * object$

Two ways to view transformations.

- **object-centric**:
Define the object in its own model coordinate system, apply S, R, T in order.
That is, read T RS(v) from right to left and the code from bottom (=glVertex) up.

- **finite-state machine**:
Modify the ModelView matrix, i.e. form T RS then apply to the object v.
That is, read T RS(v) from left to right and the code from top (=glLoadIdentity) down.

The ordering of operations is important!