

Polyhedra

CS116B

Chris Pollett

Jan 26, 2005.

Outline

- Polyhedra
- OpenGL Functions
- Curved Surfaces

Introduction

- We want to be able to draw complicated 3D objects
- Typically ,will use a **boundary representation** (B-rep).
 - This will be made of polygons or splines
- Alternatively, can use **space-partitioning representations** (octrees)
- Today, we are going to start with the simplest kind of 3D object and discuss how we can represent them

Polyhedra

- We will most often use polygon representations.
- Have a lists of point vertices. Then have a list of polygon facets (usually triangles or quadrilaterals) based on these vertices.
- Most graphics packages have commands for drawing simple polyhedra (3D objects whose faces are made of polygons) as these are easy to represent using polygon facets

OpenGL Polygon Fill-Area Functions

- We could draw polygons in polyhedra ourselves:

```
glBegin(GL_POLYGON);
```

```
//list points with glVertex
```

```
glEnd();
```

- Can also use `GL_TRIANGLES`,
`GL_TRIANGLE_STRIP`, `GL_TRIANGLE_FAN`,
`GL_QUADS`, etc

GLUT Regular Polyhedron Functions

- It is often easier to draw polyhedra using GLUT commands:
 - `glutWireTetrahedron();`
 - `glutSolidTetrahedron();`
 - `glutWireCube(edgeLength); //edgeLength`
`//is a double`
 - `glutSolidCube(edgeLength);`
 - `glutWireOctahedron(); //unit size`
 - `glutSolidOctahedron();`
 - `glutWire/SolidDodecahedron();`
 - `glutWire/SolidIcosahedron();`

Curved Surfaces

- For curved surfaces we can either use a parametric or nonparametric form for the surface.
- Next day we will consider such representations for Sphere/Ellipsoids, etc.