University of Birmingham

Computer Science Year 2

Faisal IMH Alrajhi

Year 2 Study Guide

Contents

Chapter 1

Graphics

1.1 Surface Geometry

This section covers the basics introduced in how to represent shapes in a computer.

1.1.1 Definitions

- Vertex: A point with three numbers representing its XYZ position in a plane
- Edge: An edge is the difference between two vertices; the segment connecting them
- Surface: A closed set of edges representing a face of a 3D object
- Polygon: A shape in space usually representing by a set of surfaces (other methods listed below)
- Polygon Table: A table containing a set of either vertices, edges and/or surfaces that is used to define the boundaries of a polygon. This is one method to define Polygons.
- Delaunay Triangulation: Given a set P of points in a plane, creates a triangular mesh DT(P) such that no point in P is inside the circumcircle of any triangle in DT(P).

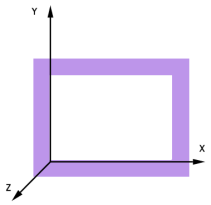


Figure 1.1: Coordinate system assumed throughout module

The default coordinate system assumed is right-handed: the positive x and y axes point right and up, and the negative z axis points forward. Positive rotation is counterclockwise about the axis of rotation.
Polygon Table consistency checks:

1. Every vertex is listed as an endpoint of at least two edges
2. Every surface is closed
3. Each surface has at least one shared edge

The order the vertices/edges are listed in a Geometric Polygon table do matter. Vertices written in clockwise order represent a surface pointing outwards. Whereas listing them counterclockwise represents an inwards pointing surface.
Meshes are a wireframe representation in which all vertices form a single set of continuous triangles, and all edges are a part of at least two triangles. Meshes can be generated by triangulation; but we covered just Delaunay Triangulation, defined above. Meshes can also be

progressive. Detail in meshes is unnecessary at farther distances, so vertices can be removed and added to create less detailed or more detailed meshes, respectively. Progress meshes do this dynamically based on viewer distance.
There are a few ways to represent polygons in a space, with boundary representations being only one method.

1. Boundary Representation: Using vertices and drawing edges and surfaces from them
2. Volumetric Models: Using simple shapes and various operations to create more complex shapes
3. Implicit Models: Using implicit equations, such as that of a sphere, to generate shapes
4. Parametric Models: Uses parametric equations to plot the multiple axes of a shape

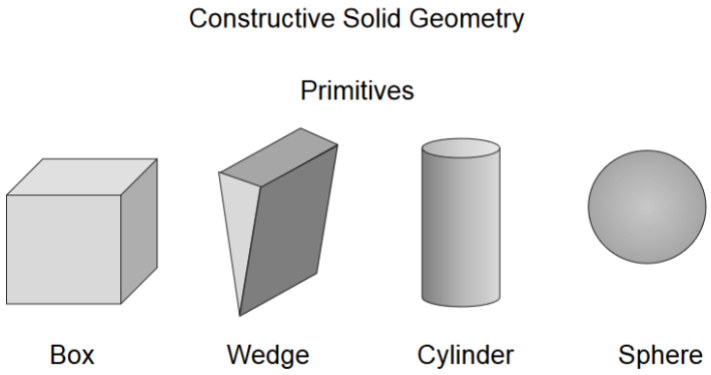


Figure 1.2: Constructive Solid Geometry (CSG) Primitives

We covered a few volumetric models in the module.

1. CSG: Uses primitve shapes and combines them uses set operations (union, difference, exclude, etc.) to generate new, more complex shapes.
2. Voxels: 3D Pixels, unit cubes
3. Octrees: Quad trees that divide in 3D space. Individual partitions are voxels
4. Sweep: Using a 2D shape, moves that shape across a path, generating a volume in position the 2D shape occupies during its path

One can also use implicit or parametric equations to generate shapes. Below is a list of equations that are common.

2D Circle:

$$\left(\frac{x}{r}\right)^2 + \left(\frac{y}{r}\right)^2 = 1 \tag{1.1}$$

2D Circle - Parametric:

$$\begin{aligned} x &= r\cos\theta \\ y &= r\sin\theta \\ -\pi &\leq \theta \leq \pi \end{aligned} \tag{1.2}$$

2D Ellipse - Parametric:

$$\begin{aligned} x &= r_x\cos\theta \\ y &= r_y\sin\theta \\ -\pi &\leq \theta \leq \pi \end{aligned} \tag{1.3}$$

3D Sphere:

$$\left(\frac{x}{r}\right)^2 + \left(\frac{y}{r}\right)^2 + \left(\frac{z}{r}\right)^2 = 1 \tag{1.4}$$

3D Ellipsoid:

$$\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2 + \left(\frac{z}{r_z}\right)^2 = 1 \tag{1.5}$$

3D Sphere - Parametric:

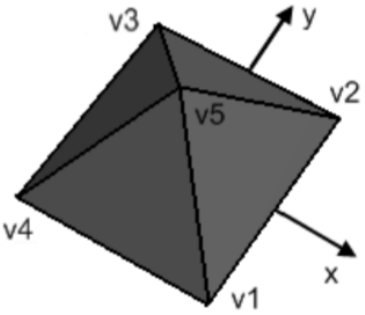
$$\begin{aligned} x &= r\cos\phi\cos\theta \\ y &= r\cos\phi\sin\theta \\ z &= r\sin\phi \\ -\pi &\leq \theta \leq \pi \\ -\pi/2 &\leq \phi \leq \pi/2 \end{aligned} \tag{1.6}$$

3D Ellipsoid - Parametric:

$$\begin{aligned} x &= r_x\cos\phi\cos\theta \\ y &= r_y\cos\phi\sin\theta \\ z &= r_z\sin\phi \\ -\pi &\leq \theta \leq \pi \\ -\pi/2 &\leq \phi \leq \pi/2 \end{aligned} \tag{1.7}$$

1.1.2 Examples

Define a vertex table and a surface table for the pyramid, depicted on the right. The base of the pyramid is a square with the side a=2 centered in the origin. The height of the pyramid is equal to 2. Work in the right handed coordinate system.



v1 [1; -1; 0]
v2 [1; 1; 0]
v3 [-1; 1; 0]
v4 [-1; -1; 0]
v5 [0; 0; 2]

f1 : v1 - v2 - v5
f2 : v2 - v3 - v5
f3 : v3 - v4 - v5
f4 : v4 - v1 - v5
f5 : v1 - v4 - v3 - v2

Figure 1.3: Example from Lecture

Further Examples are taken from quizzes and assignments

1.1.3 Further Sources

Surface Representations

1.2 Transforms

1.3 Lighting

1.4 Projection

1.5 Texture Mapping

1.6 Past Exam Practice

Chapter 2

Computational Vision

Chapter 3

Models of Computation

Chapter 4

Introductory Databases

Chapter 5

Computer Systems & Architecture

Chapter 6

C/C++

Chapter 7

Mathematical Techniques for Computer Science

Chapter 8

Introduction to Computer Security

Chapter 9

Professional Computing

Chapter 10

Functional Programming