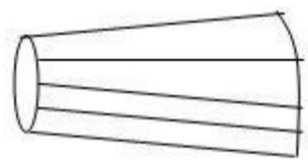
Chapter 6 Surface Modeling

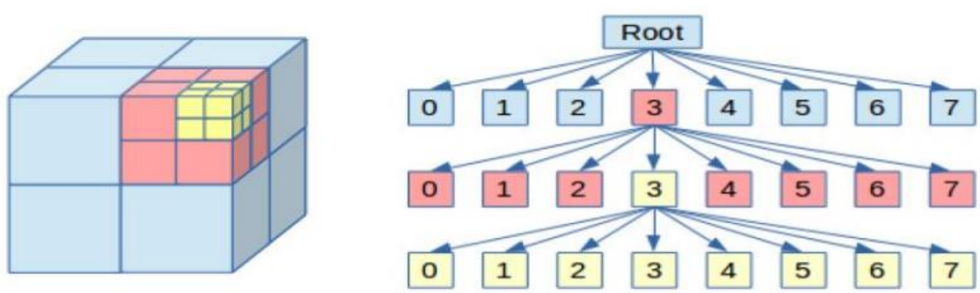
Graphics scenes can contain many different kinds of objects like trees, flowers, clouds, rocks, water etc. these cannot be describe with only one methods but obviously require large precisions such as polygon & quadratic surfaces, spline surfaces, procedural methods, volume rendering, visualization techniques etc. Representation schemes for solid objects are often divided into two broad categories:

1. **Boundary representations(B-reps):**
    - It is used to describe a three-dimensional object as a set of surfaces that separate the object interior from the environment.
    - B-reps describe the objects exterior. It describes a 3d object as a set of surfaces that encloses the objects interior. Examples: Polygon surfaces and spline patches.



A 3D object represented by polygons

2. **Space-partitioning representations:**
    - It is used to describe interior properties, by partitioning the spatial region containing an object into a set of small, non-overlapping, contiguous solids (usually cubes). For example Octree representation.



Polygon surface:

The most commonly used boundary representation for a three-dimensional graphics object is a set of surface polygons that enclose the object interior. Many graphics systems store all object descriptions as sets of surface polygons.

This simplifies and speeds up the surface rendering and display of objects, since all surfaces are described with linear equations. For this reason, polygon descriptions are often referred to as “standard graphics objects”.

Generally polygon surfaces are specified using;

1. Polygon Table
2. Plane Equations
3. Polygon Meshes.

Polygon Table:

Polygons tables can be used specified specify polygon surfaces. We specify a polygon surface with a set of vertex coordinates and associated attribute parameters. As information for each polygon is input, the data are placed into tables that are to be used in the subsequent' processing, display, and manipulation of the objects in a scene.

Polygon data tables store the coordinate description and parameters that specify the spatial orientation of polygon surface characteristics such as surface reflectivity, degree of transparency, surface texture etc.

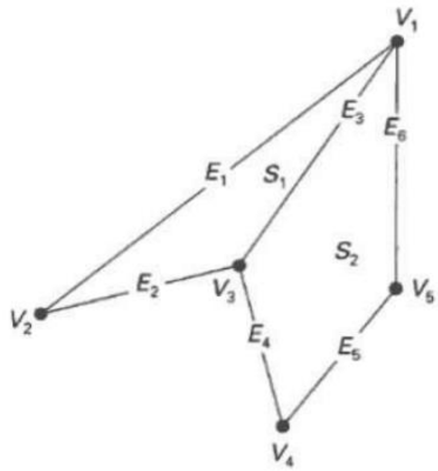
Polygon tables can be organized into two groups:

1. Geometric tables
2. Attribute tables

1. **Geometric Data tables:**
   Contain vertex coordinates and parameters to identify the spatial orientation of the polygon surfaces. Geometric data consists of three tables:
   i. **Vertex table**: It stores co-ordinate values for each vertex of the object.

   ii. **Edge table:** The edge table contains pointers back into the vertex table to identify the vertices for each polygon edge.

   iii. **Surface table**: It contains pointers back into the edge table to identify the edges for each polygon surfaces.



VERTEX TABLE
V_1: x_1, y_1, z_1
V_2: x_2, y_2, z_2
V_3: x_3, y_3, z_3
V_4: x_4, y_4, z_4
V_5: x_5, y_5, z_5

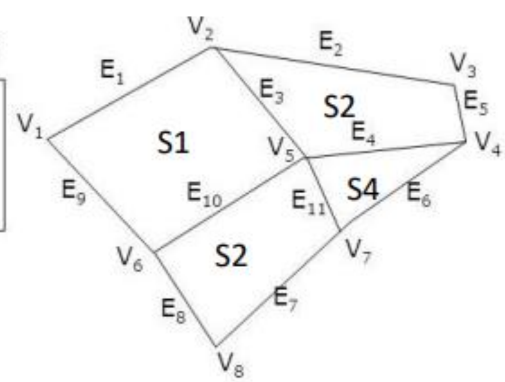
EDGE TABLE
E_1: V_1, V_2
E_2: V_2, V_3
E_3: V_3, V_1
E_4: V_3, V_4
E_5: V_4, V_5
E_6: V_5, V_1

POLYGON-SURFACE TABLE
S_1: E_1, E_2, E_3
S_2: E_3, E_4, E_5, E_6

Vertices
V_1:(x_1,y_1,z_1)
V_2:(x_2,y_2,z_2)
V_3:(x_3,y_3,z_3)
V_4:(x_4,y_4,z_4)
V_5:(x_5,y_5,z_5)
V_6:(x_6,y_6,z_6)
V_7:(x_7,y_7,z_7)
V_8:(x_8,y_8,z_8)

Edges
E_1: V_1,V_2
E_2: V_2,V_3
E_3: V_2,V_5
E_4: V_4,V_5
E_5: V_3,V_4
E_6: V_4,V_7
E_7: V_7,V_8
E_8: V_6,V_8
E_9: V_1,V_6
E_{10}: V_5,V_6
E_{11}: V_5,V_7

Polygons
S_1: E_1,E_3,E_{10},E_9
S_2: E_2,E_5,E_4,E_3
S_3: E_{10},E_{11},E_7,E_8
S_4: E_4,E_6,E_{11}



Forward pointers: i.e. to access adjacent surfaces edges

V_1: E_1,E_9
V_2: E_1,E_2,E_3
V_3: E_2,E_5
V_4: E_4,E_5,E_6
V_5: E_3,E_4E_{10},E_{11}
V_6: E_8,E_9,E_{10}
V_7: E_6,E_7,E_{11}
V_8: E_7,E_8

E_1: S_1	
E_2: S_2	
E_3: S_1,S_2	
E_4: S_2,S_4	
E_5: S_2	E_6: S_4
E_7: S_3	E_8: S_3
E_9: S_1	E_{10}: S_1,S_3
E_{11}: S_3,S_4	

2. **Attribute tables:**

It provides information for an object and includes parameters specifying the degree of transparency of the object and its surface reflectivity and texture characteristics. The above three table also include the polygon attribute according to their pointer information.

Guidelines to Generate Error Free Table:

- Every vertices listed as an end point for at least two edges.
- Every edge is part of at least one polygon
- Every polygon is closed.
- Every polygon has at least one shared edge.
- If edge table contains pointers to polygons, every edge referenced by a polygon pointer has a reciprocal pointer back to polygon

Plane Equations:

To produce a display of 3D object, we must process the input representation for the object through several procedures. These processing steps include:

- Transformation of MC to WC to VC to DC.
- Identification of visible surfaces.
- The application of surface rendering procedures.

It is used to determine the spatial orientation of the individual surface component of the object. The equation for a plane surface can be expressed in the form

$$Ax+By+Cz+D=0$$

where (x, y, z) is any point on the plane, and the coefficients A , B , C , and D are constants. Let $(x_1\ y_1,\ z_1,)$, $(x_2,\ y_2,\ z_2)$, and $(x_3,\ y_3,\ z_3)$ be three successive polygon vertices of the polygon.

$$Ax_1 + By_1+Cz_1 + D= 0,$$

$$Ax_2 + By_2+Cz_2 + D= 0,$$

$$Ax_3 + By_3+Cz_3 + D= 0$$

Using Cramer's rule, we get

$$A = \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{vmatrix} \quad B = \begin{vmatrix} x_1 & 1 & z_1 \\ x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \end{vmatrix} \quad C = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \quad D = - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}$$

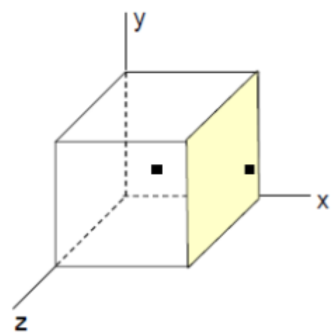
Expanding the determinant we can write that,

$$A = y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2)$$
$$B = z_1(x_2 - x_3) + z_2(x_3 - x_1) + z_3(x_1 - x_2)$$
$$C = x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)$$
$$D = -x_1(y_2 z_3 - y_3 z_2) - x_2(y_3 z_1 - y_1 z_3) - x_3(y_1 z_2 - y_2 z_1)$$

Inside outside tests of the surface:

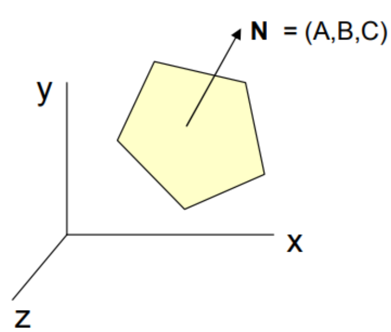
$A\,x + B\,y + C\,z + D < 0$, point (X,Y,Z) is inside the surface

$A\,x + B\,y + C\,z + D > 0$, point (X,Y,Z) is outside the surface



Orientation of a plane surface in space can be described with the normal vector to the plane.

The normal vector has Cartesian components (A, B, C) where A, B, C are the plane coefficients calculated above.



Chapter 6 Surface Modeling

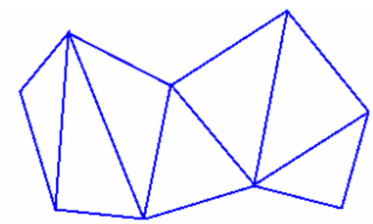
Polygon Meshes:

A polygon mesh is a collection of vertices, edges and faces that defines the shape of a polyhedral object in 3D computer graphics

Common Types:

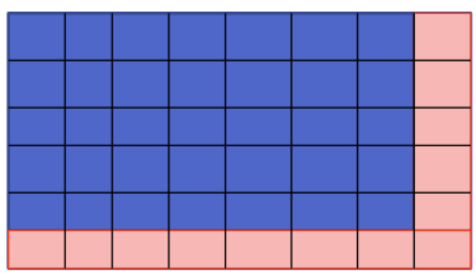
1. **Triangular Mesh:**

   It produces minimum n – 2 connected triangles, given the coordinates for n vertices.



2. **Quadrilateral Mesh**

   Another similar functions the quadrilateral mesh that generates a mesh of (n-1)(m-1) quadrilaterals, given the coordinates for an n by m array of vertices.



6 by 8 vertices array , 35 element quadrilateral mesh

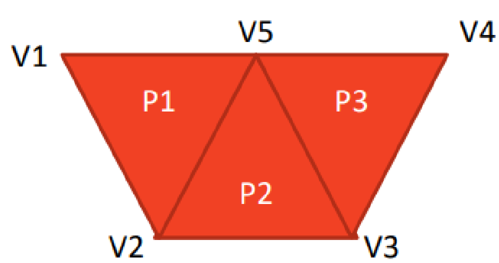
If the surface of 3D object is planner. It is comfortable to represent surface with meshes.

Representation Polygon meshes:

$$P = \{ (X_1,Y_1,Z_1), (X_2,Y_2,Z_2),\ldots\ldots\ldots,(X_n,Y_n,Z_n)\}$$

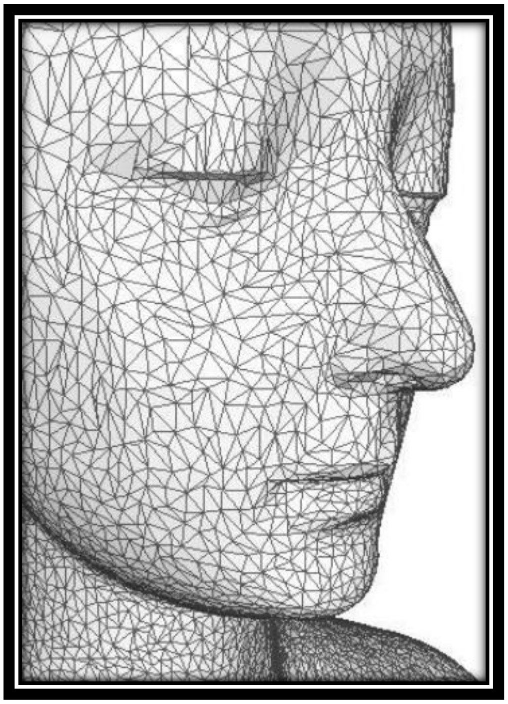
Each polygon represent by a list of vertex of coordinate

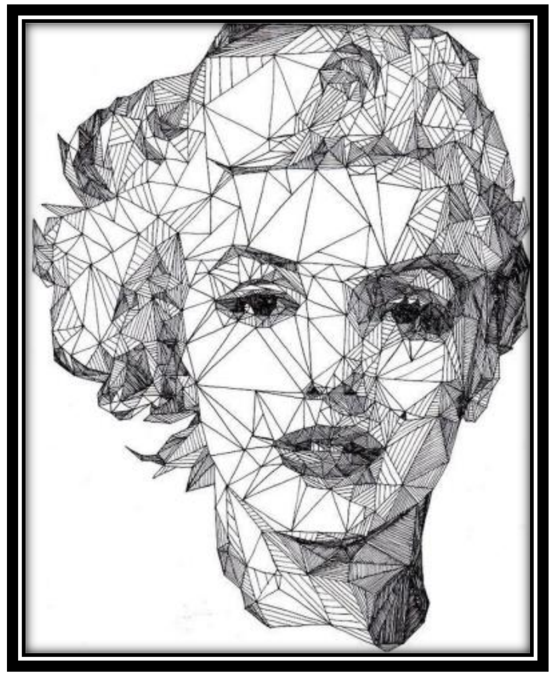
P1= { V1,V2,V5}
P2={ V2,V3,V5}
P3={V3,V4,V5}

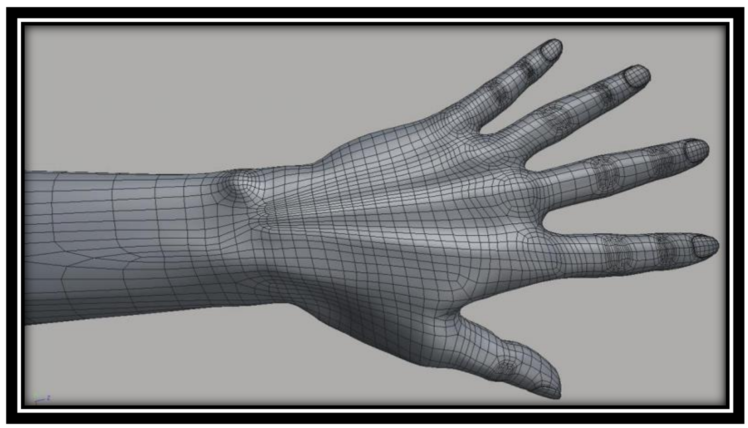


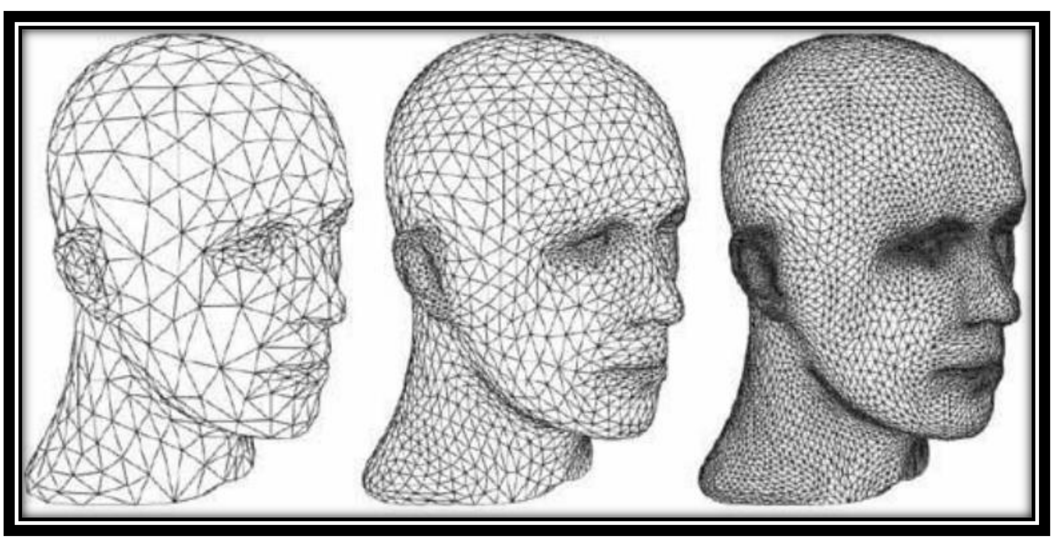
Chapter 6 Surface Modeling

Polygon Meshes:









Chapter 6 Surface Modeling

Wireframe:

- A wireframe is a three-dimensional model that only includes vertices and lines. It does not contain surfaces, textures, or lighting like a 3D mesh.
- Instead, a wireframe model is a 3D image comprised of only "wires" that represent three-dimensional shapes.
- A wire-frame model is a visual presentation of a 3- dimensional (3D) or physical object used in 3D computer graphics.
- They are often used as the starting point in 3D modeling since they create a "frame" for 3D structures. For example, a 3D graphic designer can create a model from scratch by simply defining points (vertices) and connecting them with lines (paths).
- Once the shape is created, surfaces or textures can be added to make the model appear more realistic.
- The lines within a wireframe connect to create polygons, such as triangles and rectangles that together represent three dimensional shapes.
- The result may be as simple as a cube or as complex as a three-dimensional scene with people and objects. The number of polygons within a wireframe is typically a good indicator of how detailed the 3D model is.