<u>**3D Object representations**</u>

Graphical scenes can contain many different kinds of objects like trees, flowers, rocks, waters...etc. There is no one method that we can use to describe objects that will include all features of those different materials. To produce realistic display of scenes, we need to use representations that accurately model object characteristics.

- Simple Euclidean objects like polyhedrons and ellipsoids can be represented by polygon and quadric surfaces.
- Spline surface are useful for designing aircraft wings, gears and other engineering objects.
- Procedural methods and particle systems allow us to give accurate representation of clouds, clumps of grass, and other natural objects.
- Octree encodings are used to represent internal features of objects. Such as medical CT images.

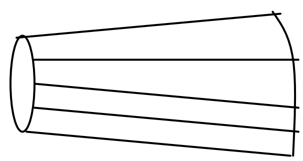
Representation schemes for solid objects are often divided into two broad categories:

1. <u>**Boundary representations**</u>: describes a 3D object as a set of polygonal surfaces, separate the object interior from environment.

2. <u>**Space-partitioning representation**</u>: used to describe interior properties, by partitioning the spatial region, containing an object into a set of small, non overlapping, contiguous solids. e.g. 3D object as Octree representation.

<u>**Boundary Representation**</u>: Each 3D object is supposed to be formed its surface by collection of polygon facets and spline patches. Some of the boundary representation methods for 3D surface are:

1. Polygon Surfaces: It is he most common representation for 3D graphics object. In this representations, a 3D object is represented by a set of surfaces that enclose the object interior. Many graphics system use this method. Set of polygons are stored for object description. This simplifies and speeds up the surface rendering and display of object since all surfaces can be described with linear equations.



A 3D object represented by polygons

The polygon surface are common in design and solid-modeling applications, since wire frame display can be done quickly to give general indication of surface structure. Then realistic scenes are produced by interpolating shading patterns across polygon surface to illuminate.

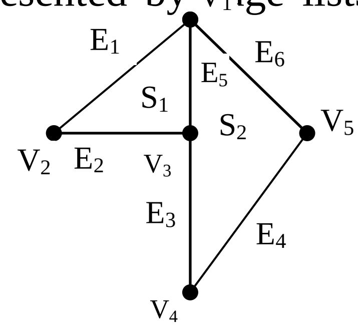
Polygon Table: A polygon surface is specified with a set of vertex co-ordinates and associated attribute parameters. A convenient organization for storing geometric data is to create 3 lists:

- A vertex table
- An edge table
- A polygon surface table.

− Vertex table stores co-ordinates of each vertex in the object.
− The edge table stores the Edge information of each edge of polygon facets.

− The polygon surface table stores the surface information for each surface i.e. each surface is represented by V_1lge lists of polygon.

Consider the surface contains polygonal facets as shown in figure (only two polygon are taken here)



→ S_1 and S_2 are two polygon surface that represent the boundary of some 3D object.

Foe storing geometric data, we can use following three ta

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

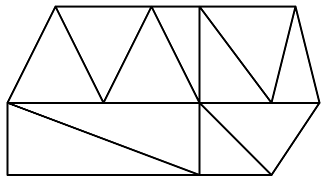
EDGE TABLE
E_1: V_1,V_2
E_2: V_2,V_3
E_3: V_3,V_4
E_4: V_4,V_5
E_5: V_1,V_3
E_6: V_5, V_1

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

The object can be displayed efficiently by using data from tables and processing them for surface rendering and visible surface determination.

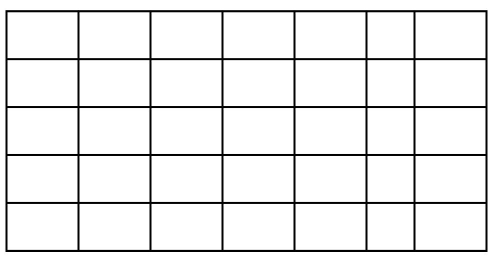
<u>Polygon Meshes</u>: A polygon mesh is collection of edges, vertices and polygons connected such that each edge is shared by at most two polygons. An edge connects two vertices and a polygon is a closed sequence of edges. An edge can be shared by two polygons and a vertex is shared by at least two edges.

When object surface is to be tiled, it is more convenient to specify the surface facets with a mesh function. One type of polygon mesh is triangle strip. This function produce n-2 connected triangles.



Triangular Mesh

Another similar function is the quadrilateral mesh, which generates a mesh of (n-1) by (m-1) quadrilaterals, given the co-ordinates for an $n \times m$ array of vertices.



6 by 8 vertices array , 35 element quadrilateral mesh

- If the surface of 3D object is planer, it is comfortable to represent surface with meshes.

<u>Representing polygon meshes</u>

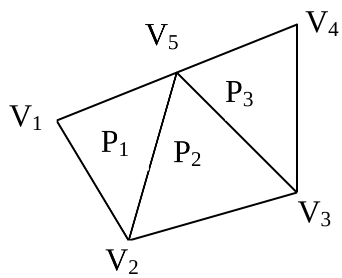
In explicit representation, each polygon is represented by a list of vertex co-ordinates.

$$P = ((x_1, y_1, z_1), (x_2, y_2, y_2), \ldots\ldots. (x_n, y_n, z_n))$$

The vertices are stored in order traveling around the polygon. There are edge between successive vertices in the list and between the last and first vertices.

- For a single polygon it is efficient but for polygon mesh it is not space efficient since no of vertices may duplicate.
- So another method is to define polygon with pointers to a vertex list. So each vertex is stored just once, in vertex list

$V = \{v_1, v_2 .....v_n\}$ A polygon is defined by list of indices (pointers) into the vertex list e.g. A polygon made up of vertices $3,5,7,10$ in vertex list be represented as $P_1 = \{3,5,7,10\}$



- Representing polygon mesh with each polygon as vertex list.
  - $P_1 = \{v_1, v_2, v_5\}$
  - $P_2 = \{v_2, v_3, v_5\}$
  - $P_3 = \{v_3, v_4, v_5\}$

  Here most of the vertices are duplicated so it is not efficient.
- Representation with indexes into a vertex list

  $V = \{v_1, v_2, v_3, v_4, v_5\} = \{(x_1, y_1, z_1),......(x_5, y_5, z_5)\}$

  $P_1 = \{1,2,3\}$

  $P_2 = \{2,3,5\}$

  $P_3 = \{3,4,5\}$

- **Defining polygons by pointers to an edge list**

In this method, we have vertex list V, represent the polygon as a list of pointers not to the vertex list but to an edge list. Each edge in edge list points to the two vertices in the vertex list. Also to one or two polygon, the edge belongs.

Hence we describe polygon as

$P = (E_1, E_2, ...E_n)$

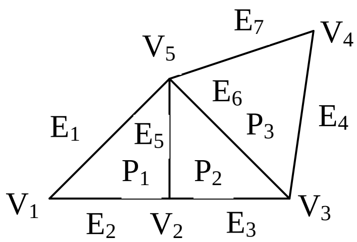
and an edge as

$E = (V_1, V_2, P_1, P_2)$

2Here if edge belongs to only one polygon, either

Then P_1 or P_2 is null.

For the mesh given below,



$$V = \{v_1, v_2, v_3, v_4, v_5\} = \{(x_1, y_1, z_1), \ldots\ldots (x_5, y_5, z_5)\}$$

$E_1 = (V_1, V_5, P_1, N)$

$E_2 = (V_1, V_2, P_1, N)$

$E_3 = (V_2, V_3, P_2, N)$

$E_4 = (V_3, V_4, P_3, N)$

$E_5 = (V_2, V_5, P_1, P_2)$

$E_6 = (V_3, V_5, P_1, P_3)$ Here N represents Null.

$E_7 = (V_4, V_5, P_3, N)$

$P_1 = (E_1, E_2, E_3)$

$P_2 = (E_3, E_6, E_5)$

$P_3 = (E_4, E_7, E_6)$

Polygon mesh defined with edge lists for each polygon.

3D-object representation of Curve and Surfaces

1. Polygon Surface: Plane Equation Method

Plane equation method is another method for representation the polygon surface for 3D object. The information about the spatial orientation of object is described by its individual surface, which is obtained by the vertex co-ordinates and the equation of each surface. 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 A,B,C,D are constants describing the spatial properties of the plane. The values of A,B,C,D can be obtained by solving a set of three plane equations using co-ordinate values of 3 non collinear points on the plane.

Let (x_1,y_1,z_1), (x_2,y_3,z_2) and (x_3,y_3,z_3) are three such points on the plane, then-

$$Ax_1 + By_1 + Cz_1 + D = 0$$
$$Ax_2 + By_2 + Cz_2 + D = 0$$
$$Ax_3 + By_3 + Cz_3 + D = 0$$

The solution of these equations can be obtained in determinant from using Cramer's rule as:-

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

For any points (x,y,z)

If $Ax + By + Cz + D \neq 0$, then (x,y,z) is not on the plane.

If $Ax + By + Cz + D < 0$, then (x,y,z) is inside the plane i. e. invisible side

If $Ax + By + Cz + D > 0$, then (x,y,z) is lies out side the surface.

2. Wireframe Representation:

In this method 3D objects is represented as a list of straight lines, each of which is represented by its two end points (x1,y1,z1) and (x2,y2,z2). This method only shows the skeletal structure of the objects.

It is simple and can see through the object and fast method. But independent line data structure is very inefficient i.e don't know what is connected to what. In this method the scenes represented are not realistic.

3. Spline Representation

A Spline is a flexible strips used to produce smooth curve through a designated set of points. A curve drawn with these set of points is spline curve. Spline curves are used to model 3D object surface shape smoothly.

Mathematically, spline are described as pice-wise cubic polynomial functions. In computer graphics, a spline surface can be described with two set of orthogonal spline curves. Spline is used in graphics application to design and digitalize drawings for storage in computer and to specify animation path. Typical CAD application for spline includes the design of automobile bodies, aircraft and spacecraft surface etc.

Interpolation and approximation spline

- Given the set of control points, the curve is said to interpolate the control point if it passes through each points.
- If the curve is fitted from the given control points such that it follows the path of control point without necessarily passing through the set of point, then it is said to approximate the set of control point.

4.Quadric Surface

Quadric Surface is one of the frequently used 3D objects surface representation. The quadric surface can be represented by a second degree polynomial. This includes:

1. Sphere: For the set of surface points (x,y,z) the spherical surface is represented as:

   $x^2+y^2+z^2 = r^2$, with radius r and centered at co-ordinate origin.

2. Ellipsoid: $\frac{x^2}{a^2}+\frac{y^2}{b^2}+\frac{z^2}{c^2}=1$, where (x,y,z) is the surface points and a,b,c are the radii on X,Y and Z directions respectively.
3. Elliptic parboiled: $\frac{x^2}{a^2}+\frac{y^2}{b^2}=z$

4. Hyperbolic parboiled : $\frac{x^2}{a^2}-\frac{y^2}{b^2}=z$

5. Elliptic cone : $\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=0$

6. Hyperboloid of one sheet: $\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=1$

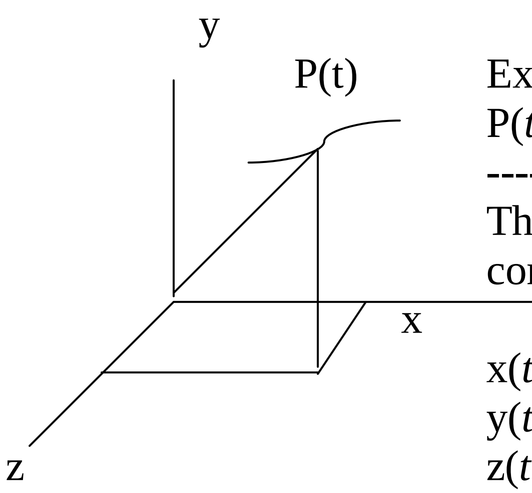
7. Hyperboloid of two sheet: $\frac{x^2}{a^2}-\frac{y^2}{b^2}-\frac{z^2}{c^2}=1$

5. Parametric Cubic Curve

A parametric cubic curve is defined as $P(t) = \sum a_i t^i \quad 0 <= t <= 1$ ------ (i)

$i = 1$ to n

Where, $P(t)$ is a point on the curve



Expanding equation (i) yields

$P(t) = a_3t^3 + a_2t^2 + a_1t + a_0$ -----------------------(ii)

This equation is separated into three components of P(t)

$x(t) = a_{3x}t^3 + a_{2x}t^2 + a_{1x}t + a_{0x}$

$y(t) = a_{3y}t^3 + a_{2y}t^2 + a_{1y}t + a_{0y}$

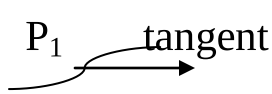
$z(t) = a_{3z}t^3 + a_{2z}t^2 + a_{1z}t + a_{0z}$----------------------(iii)

To be able to solve (iii) the twelve unknown coefficients a_{ij} (algebraic coefficients) must be specified

From the known end point coordinates of each segment, six of the twelve needed equations are obtained.

The other six are found by using tangent vectors at the two ends of each segment

The direction of the tangent vectors establishes the slopes(direction cosines) of the curve at the end points



This procedure for defining a cubic curve using end points and tangent vector is one form of *hermite* interpolation

Each cubic curve segment is parameterized from 0 to 1 so that known end points correspond to the limit values of the parametric variable *t*, that is P(0) and P(1)

Substituting $t = 0$ and $t = 1$ the relation ship between two end point vectors and the algebraic coefficients are found

$P(0) = a_0$ $\quad$ $P(1) = a_3 + a_2 + a_1 + a_0$

To find the tangent vectors equation ii must be differentiated with respect to t

$P'(t) = 3a_3t^2 + 2a_2t + a_1$

The tangent vectors at the two end points are found by substituting t = 0 and t = 1 in this equation

$P'(0) = a_1$ $\quad$ $P'(1) = 3a_3 + 2a_2 + a_1$

The algebraic coefficients 'a_i' in equation (ii) can now be written explicitly in terms of boundary conditions – endpoints and tangent vectors are

$a_0 = P(0)$ $\quad$ $a_1 = P'(0)$

$a_2 = -3\,P(0) - 3\,P(1) - 2\,P'(0) - P'(1)$ $\quad$ $a3 = 2\,P(0) - 2\,P(1) + P'(0) + P'(1)$

substituting these values of 'a_i' in equation (ii) and rearranging the terms yields

$P(t) = (2t^3 - 3t^2 + 1)\,P(0) + (-2t^3 + 3t^2)\,P(1) + (t^3 - 2t^2 + t)\,P'(0) + (t^3 - t^2)\,P'(1)$

The values of P(0), P(1), P'(0), P'(1) are called *geometric coefficients* and represent the known vector quantities in the above equation

The polynomial coefficients of these vector quantities are commonly known as *blending functions*

By varying parameter t in these blending function from 0 to 1 several points on curve segments can be found