3D Object Representation

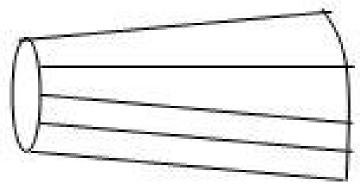
Unit 5

Nipun Thapa (computer Graphics)

3D Object Representation

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:

- ***Boundary representations (B-reps):***
  - 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*.

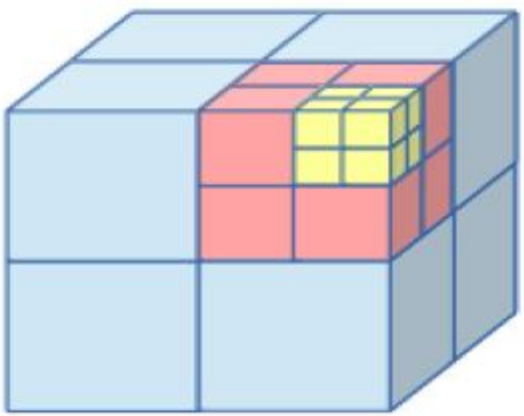


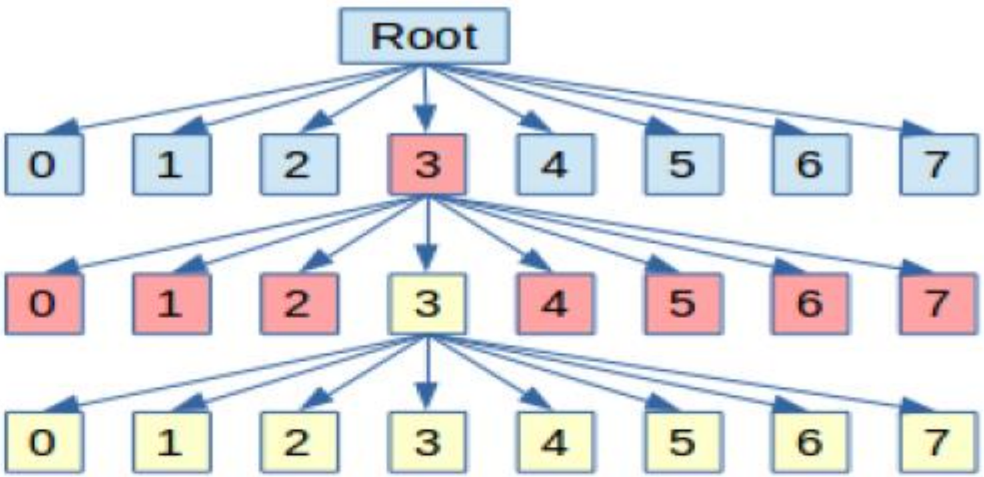
A 3D object represented by polygons

3D Object Representation

- ***Space-partitioning representations***:

are 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 Surfaces

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.

1. Polygon Tables

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 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.
2. **Attribute tables:** Provide information for an object and includes parameters specifying the degree of transparency of the object and its surface reflectivity and texture characteristics.

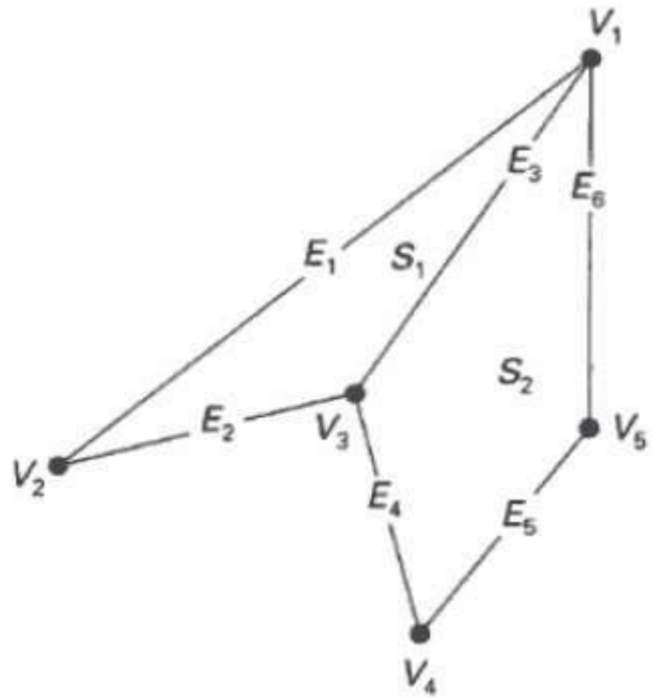
1. Polygon Tables..

Geometric data consists of three tables: a vertex table, an edge table, and a surface table.

- **Vertex table:** It stores co-ordinate values for each vertex of the object.
- **Edge Table:** The edge table contains pointers back into the vertex table to identify the vertices for each polygon edge.
- **Surface table:** And the polygon table contains pointers back into the edge table to identify the edges for each polygon surfaces.

1. Polygon Tables..

a) Geometric tables



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

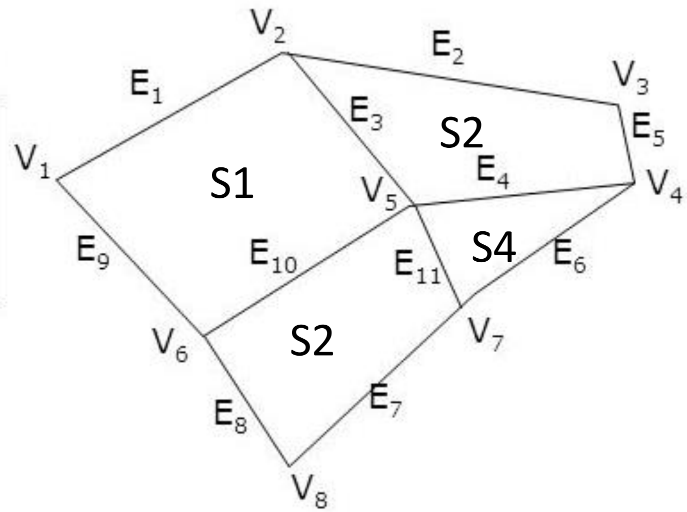
EDGE TABLE	
E_1:	V_1, V_2
E_2:	V_2, V_3
E_3:	V_3, V_1
E_4:	V_3, V_4
E_5:	V_4, V_5
E_6:	V_5, V_1

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

1. Polygon Tables..

a) Geometric tables

Vertices	Edges	Polygons
$V_1:(x_1,y_1,z_1)$	$E_1: V_1,V_2$	$S_1: E_1,E_3,E_{10},E_9$
$V_2:(x_2,y_2,z_2)$	$E_2: V_2,V_3$	$S_2: E_2,E_5,E_4,E_3$
$V_3:(x_3,y_3,z_3)$	$E_3: V_2,V_5$	$S_3: E_{10},E_{11},E_7,E_8$
$V_4:(x_4,y_4,z_4)$	$E_4: V_4,V_5$	$S_4: E_4,E_6,E_{11}$
$V_5:(x_5,y_5,z_5)$	$E_5: V_3,V_4$	
$V_6:(x_6,y_6,z_6)$	$E_6: V_4,V_7$	
$V_7:(x_7,y_7,z_7)$	$E_7: V_7,V_8$	
$V_8:(x_8,y_8,z_8)$	$E_8: V_6,V_8$	
	$E_9: V_1,V_6$	
	$E_{10}: V_5,V_6$	
	$E_{11}: V_5,V_7$	



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

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

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

1. Polygon Tables..

b) Attribute tables

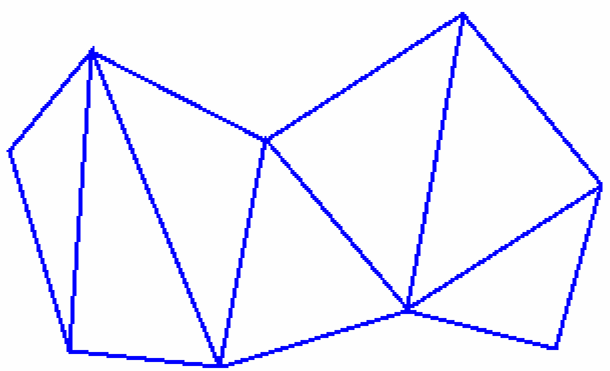
Attribute information for an object includes parameters specifying the degree of transparency of the object and its surface reflectivity and texture characteristics. The above three table also include the polygon attribute according to their pointer information.

2. 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

1. Triangular Mesh :

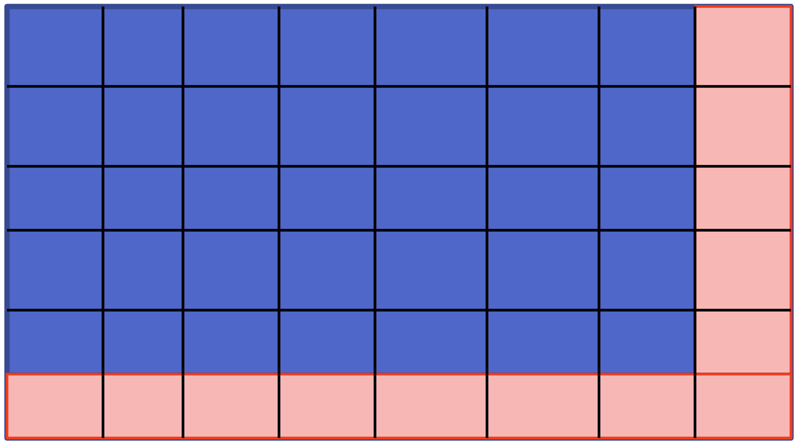
It produces n – 2 connected triangles, given the coordinates for n vertices.



2. Polygon Meshes ...

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

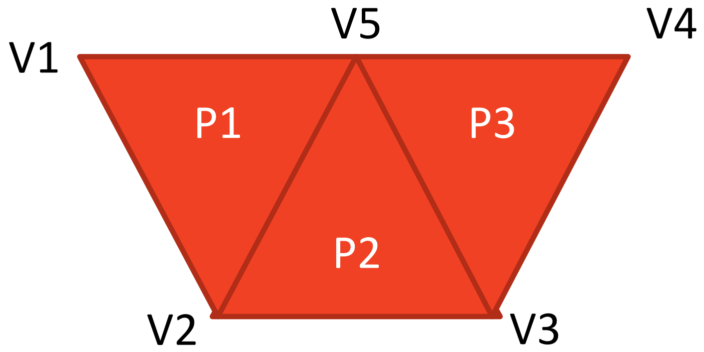
If the surface of 3D object is planner. It is confortable to represent surface with meshes

2. Polygon Meshes …

- Representation Polygon meshes

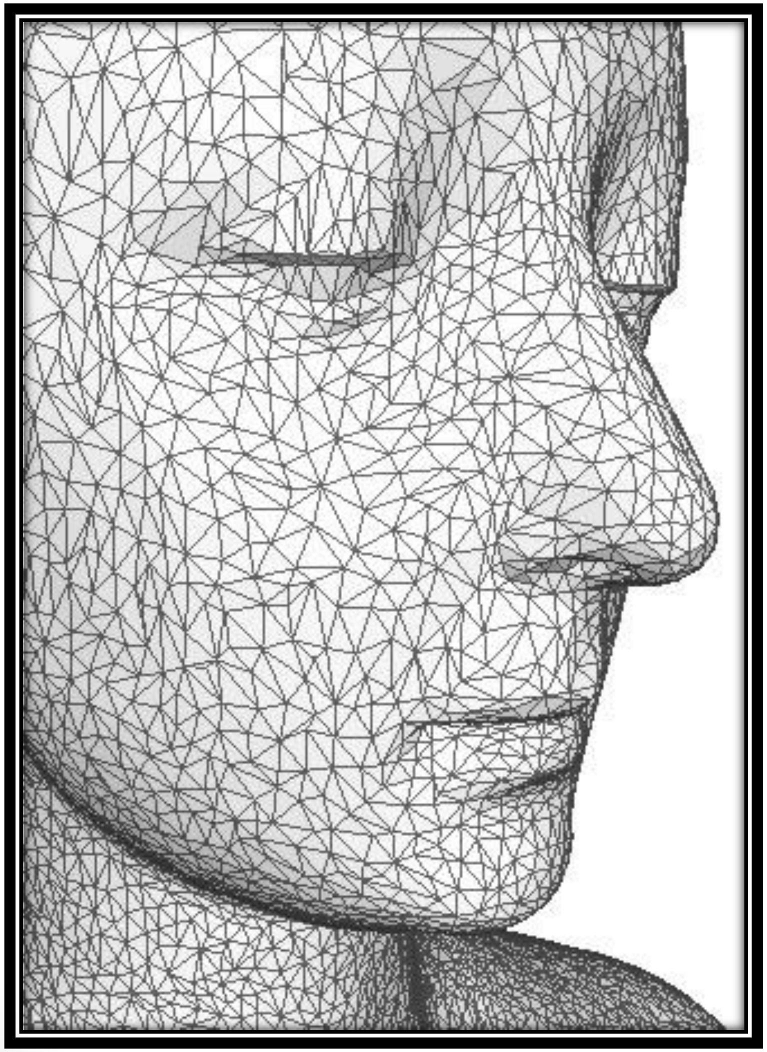
$P = \{ (X_1,Y_1,Z_1), (X_2,Y_2,Z_2),\ldots\ldots\ldots,(X_n,Y_n,Z_n)\}$

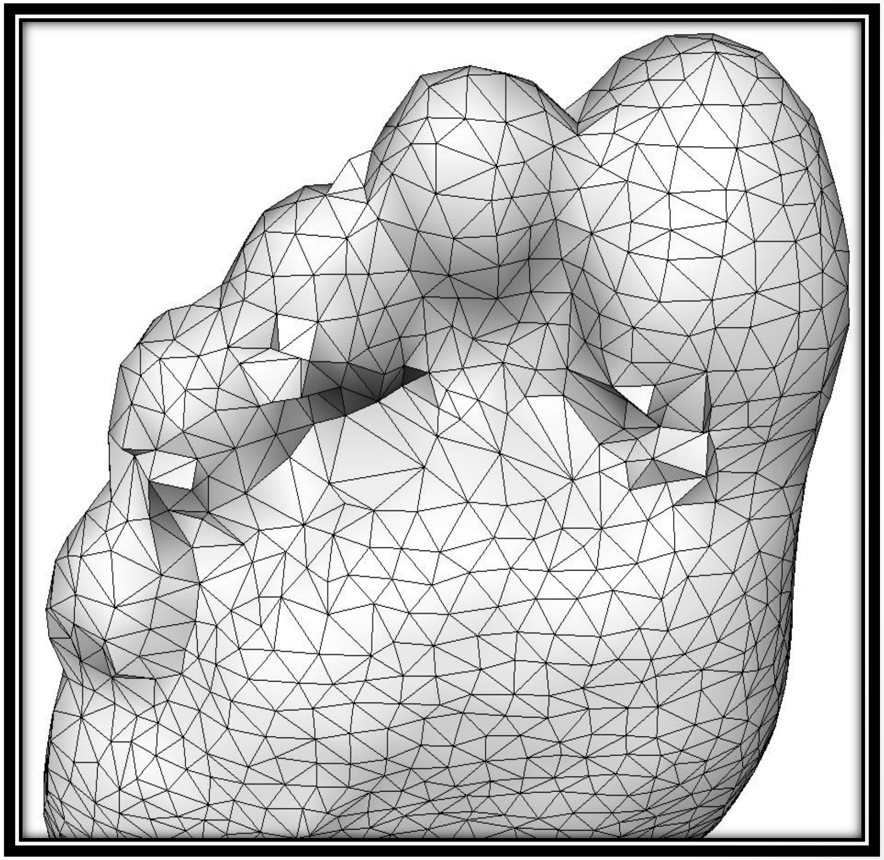
Each polygon represent by a list of vertex of coordinate



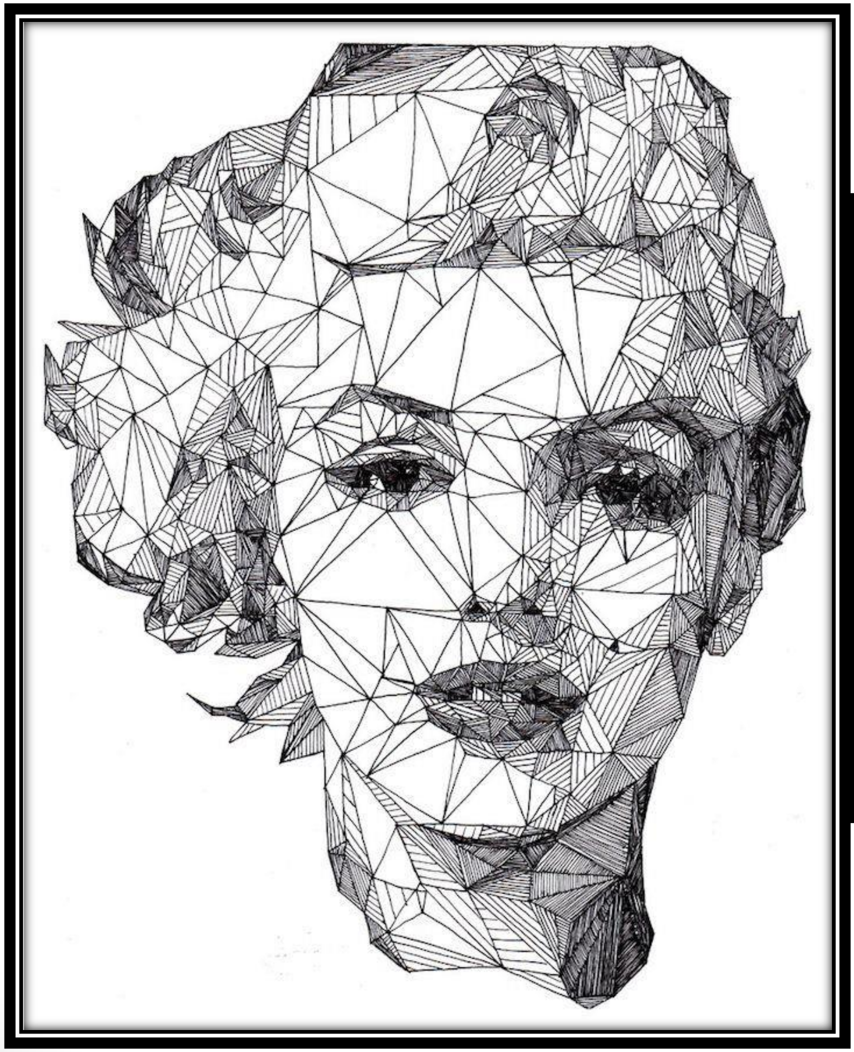
P1= { V1,V2,V3}
P2={ V2,V3,V5}
P3={V3,V4,V5}

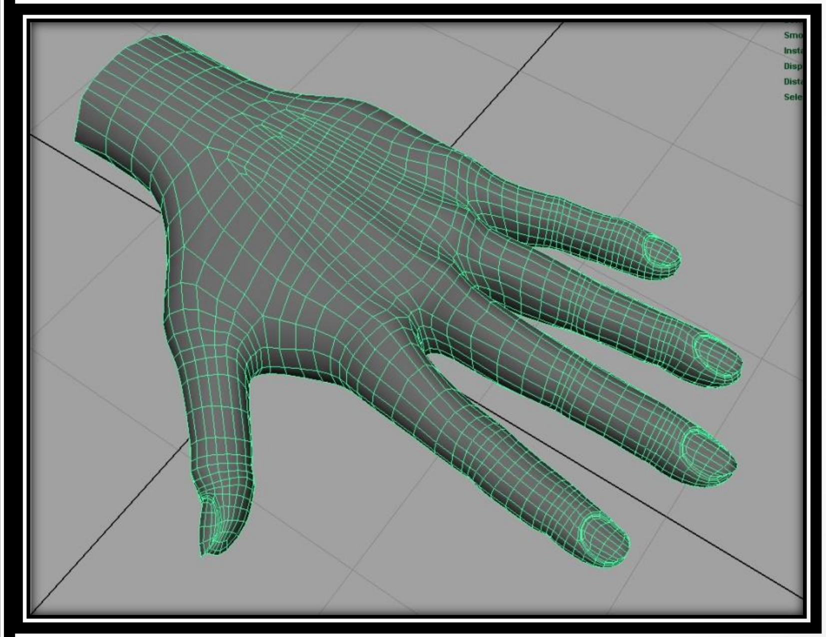
2. Polygon Meshes …



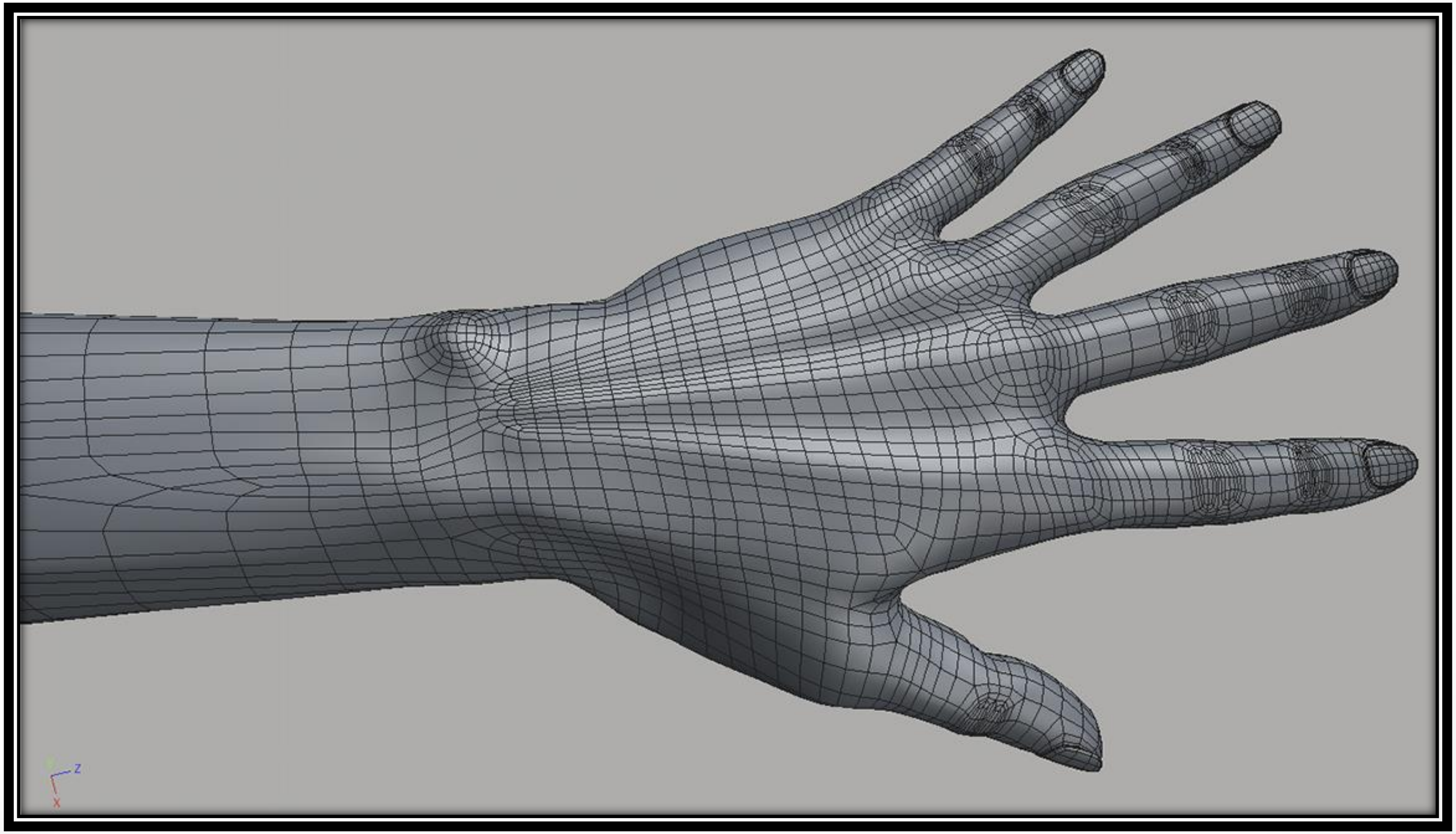


2. Polygon Meshes ...



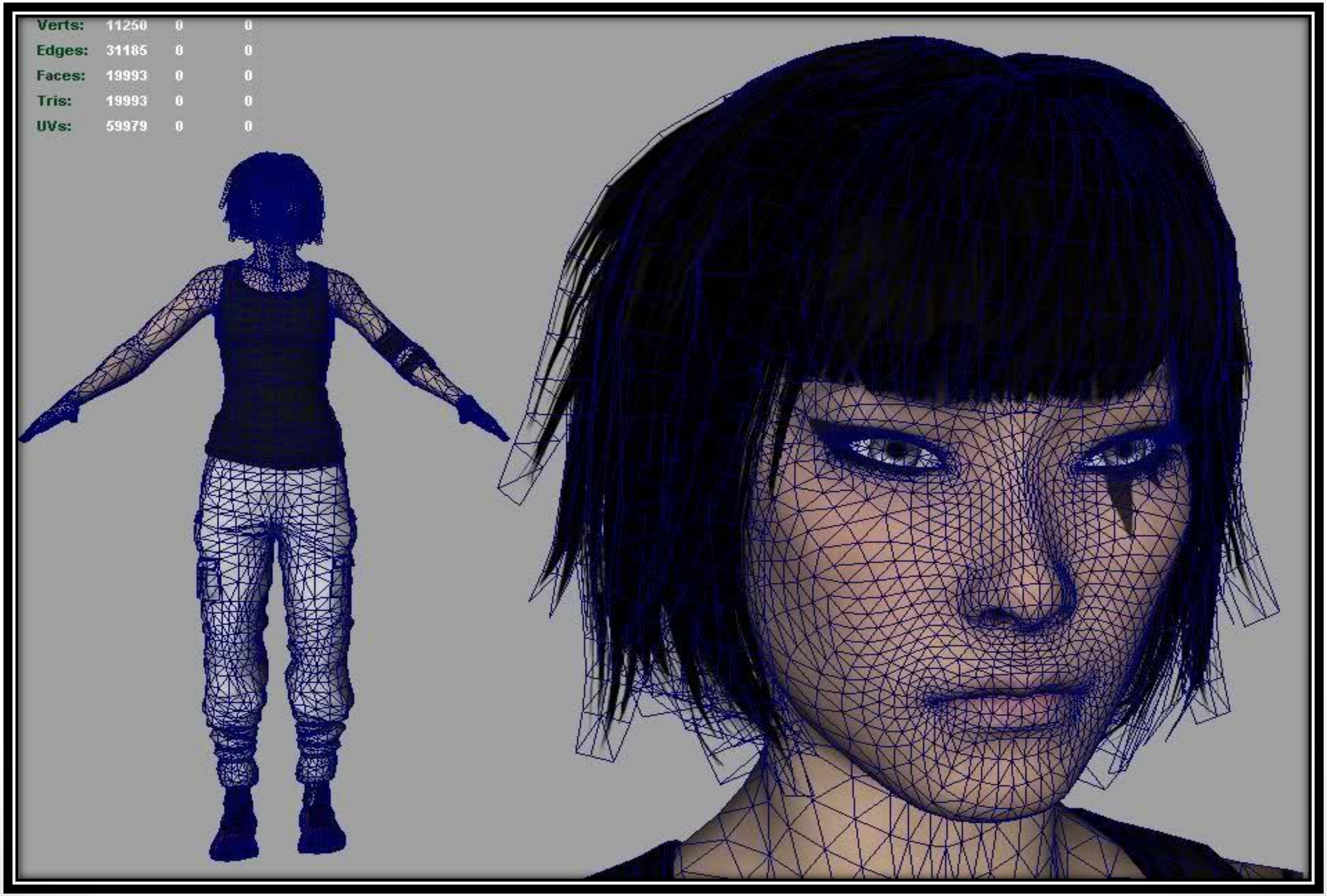


2. Polygon Meshes …

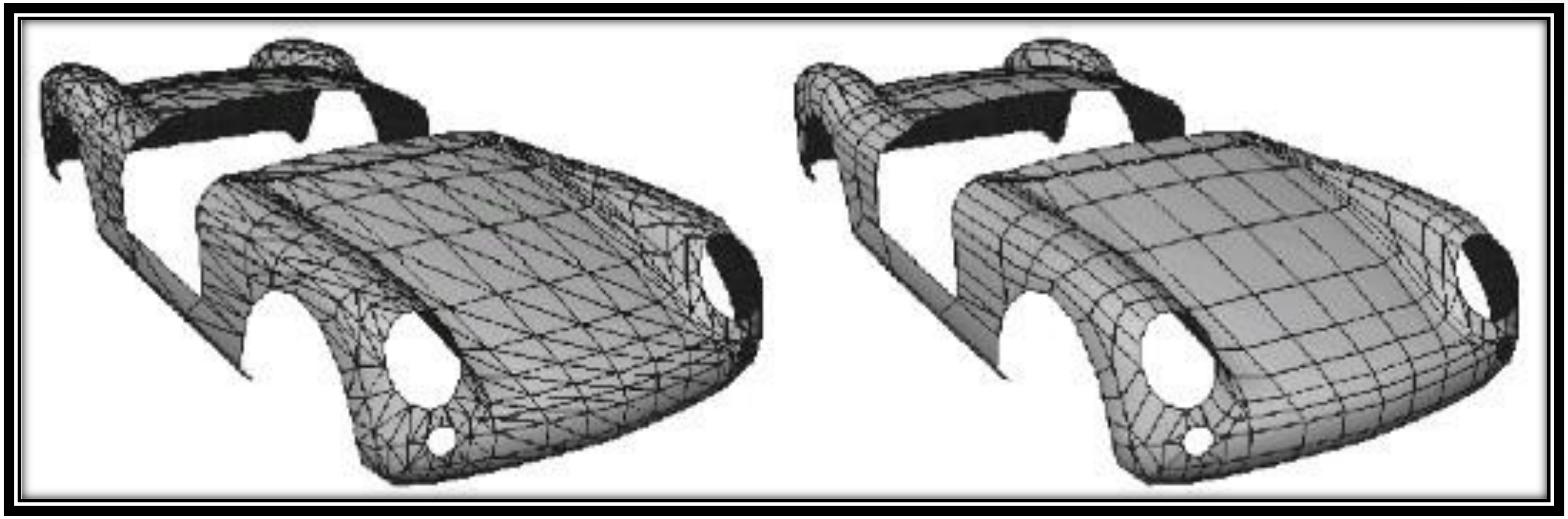


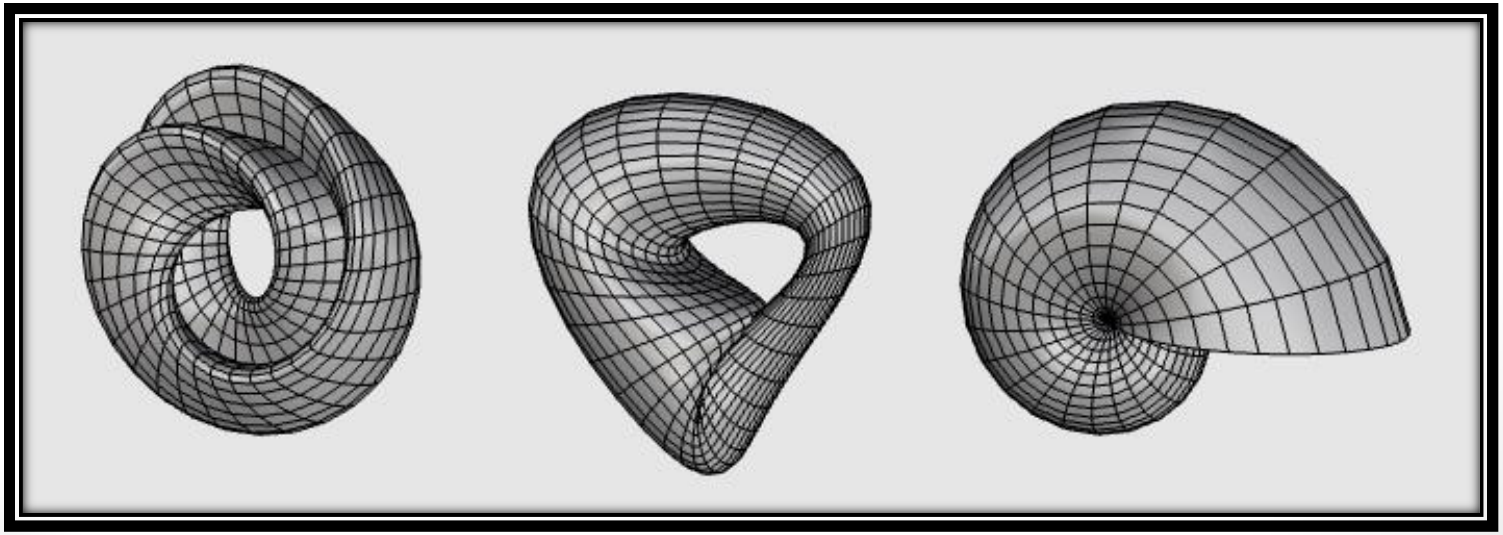
2. Polygon Meshes ...

2. Polygon Meshes ...

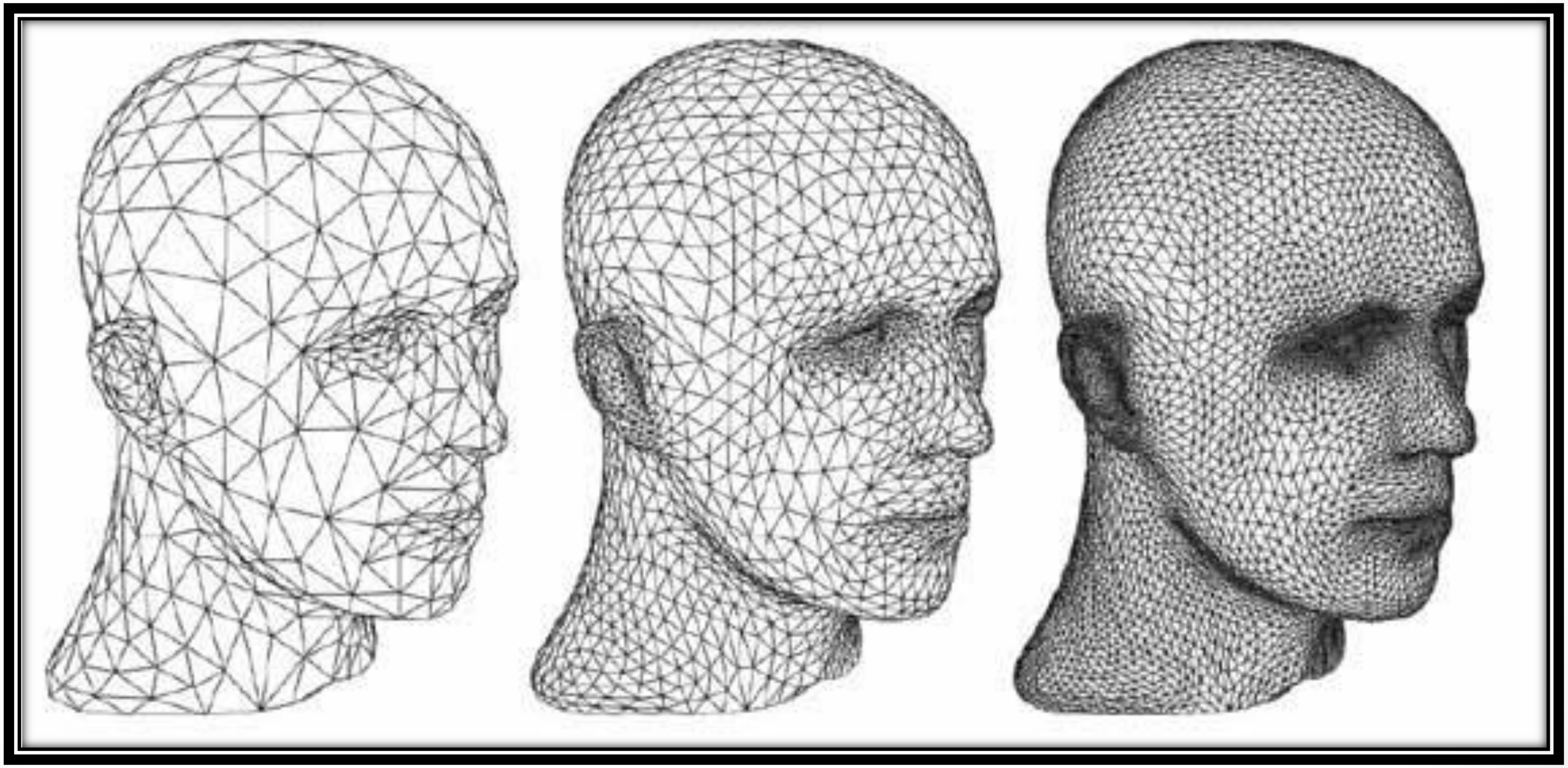


2. Polygon Meshes ...





2. Polygon Meshes …



3. Plane Equation

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.*

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

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

$Ax_3 + By_3+Cz_3 + D= 0$

3. Plane Equation ..

- Using Cramer's Rule

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

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

Expanding the determinant we can write that,

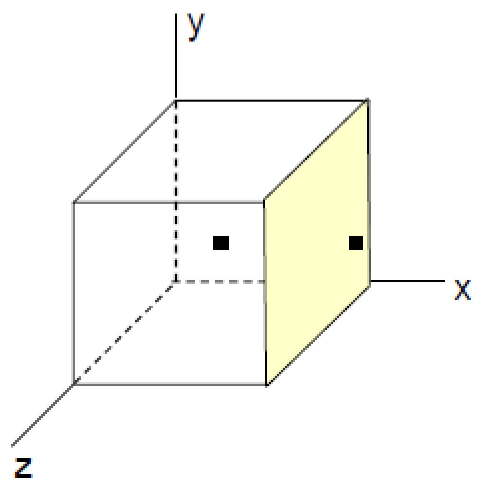
$$A=y_1(z_2-z_3)+y_2(z_3-z_1)+y_3(z_1-z_2)$$
$$B=z_1(x_2-x_3)+z_2(x_3-x_1)+z_3(x_1-x_2)$$
$$C=x_1(y_2-y_3)+x_2(y_3-y_1)+x_3(y_1-y_2)$$
$$D=-x_1(y_2 z_3-y_3 z_2)-x_2(y_3 z_1-y_1 z_3)-x_3(y_1 z_2-y_2 z_1)$$

3. Plane Equation ..

- Inside outside tests of the surface:

  $Ax + By + Cz + D < 0$, point (X,Y,Z) is inside the surface

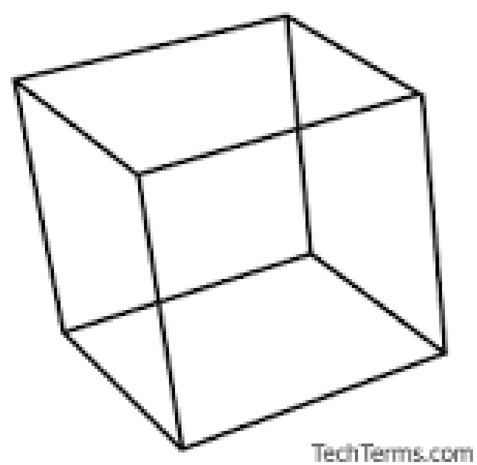
  $Ax + By + Cz + D > 0$, point (X,Y,Z) is outside the surface



Wireframe

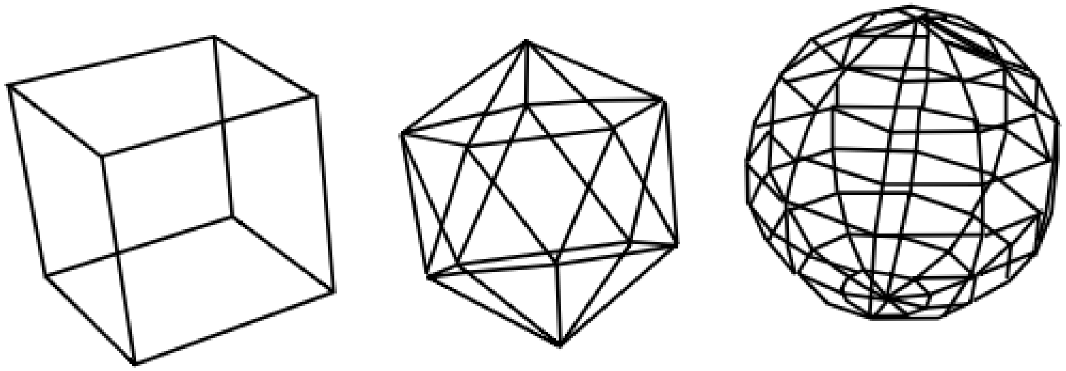
Wireframe

- 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.



Wireframe

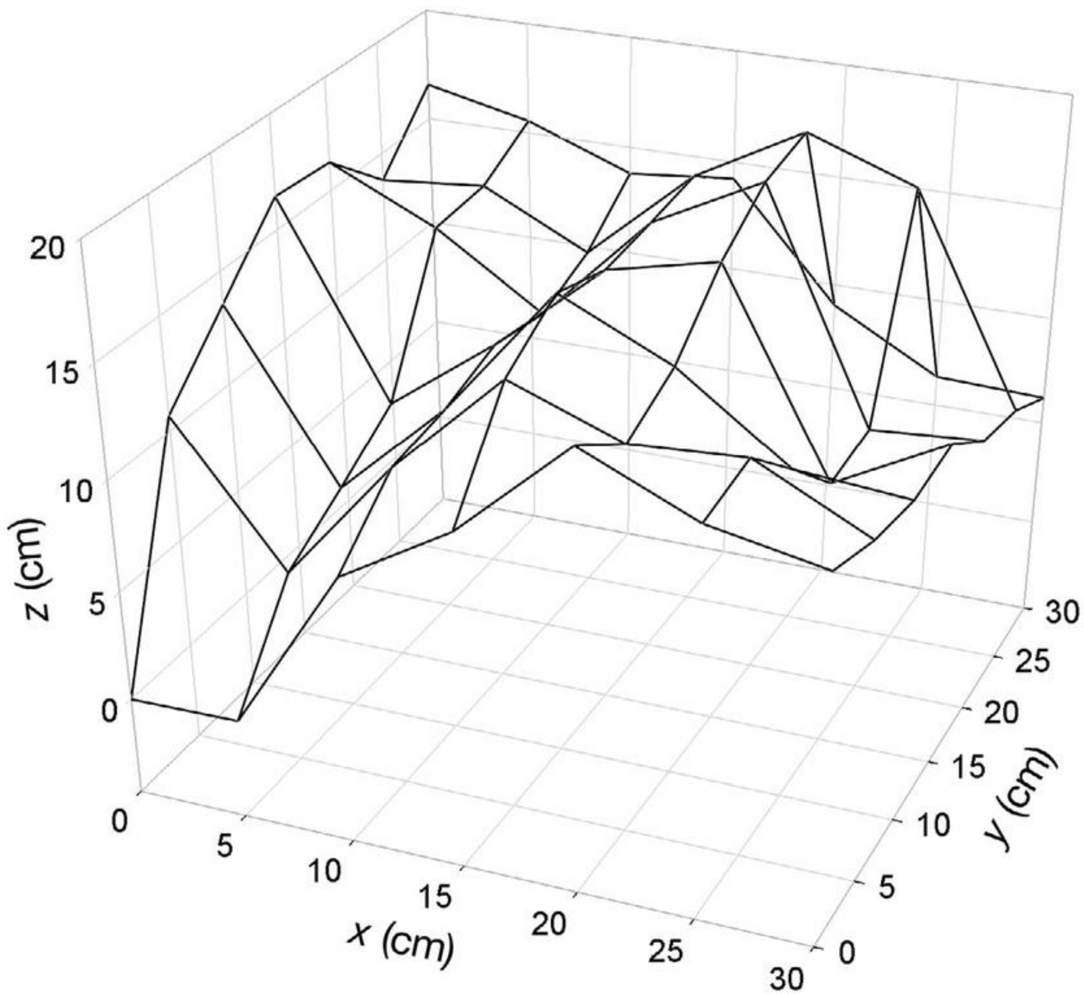
- Wireframes provide the most basic representation of a three-dimensional scene or object.
- They are often used as the starting point in 3D modeling since they create a "frame" for 3D structures. For example, a 3D graphic designer can create a model from scratch by simply defining points (vertices) and connecting them with lines (paths).
- Once the shape is created, surfaces or textures can be added to make the model appear more realistic.



Wireframe

- 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.

Wireframe



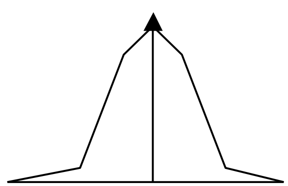
Blobby Objects

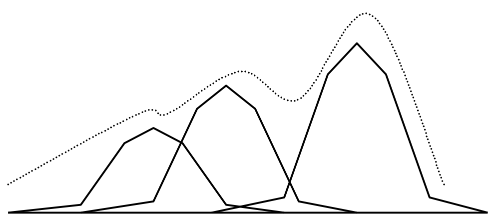
Blobby Objects

- By a blobby object we mean a nonrigid object. That is things, like cloth, rubber, liquids, water droplets, etc.
- These objects tend to exhibit a degree of fluidity.
- For example, in a chemical compound electron density clouds tend to be distorted by the presence of other atoms/molecules

More Blobby Objects

- Several models have been developed to handle these kind of objects.
- One technique is to use a combination of Gaussian density functions (Gaussian bumps).





- A surface function could then be defined by:

$$f(x,y,z) = \sum_k b_k * \exp(-a_k * r_k^2) - T = 0$$

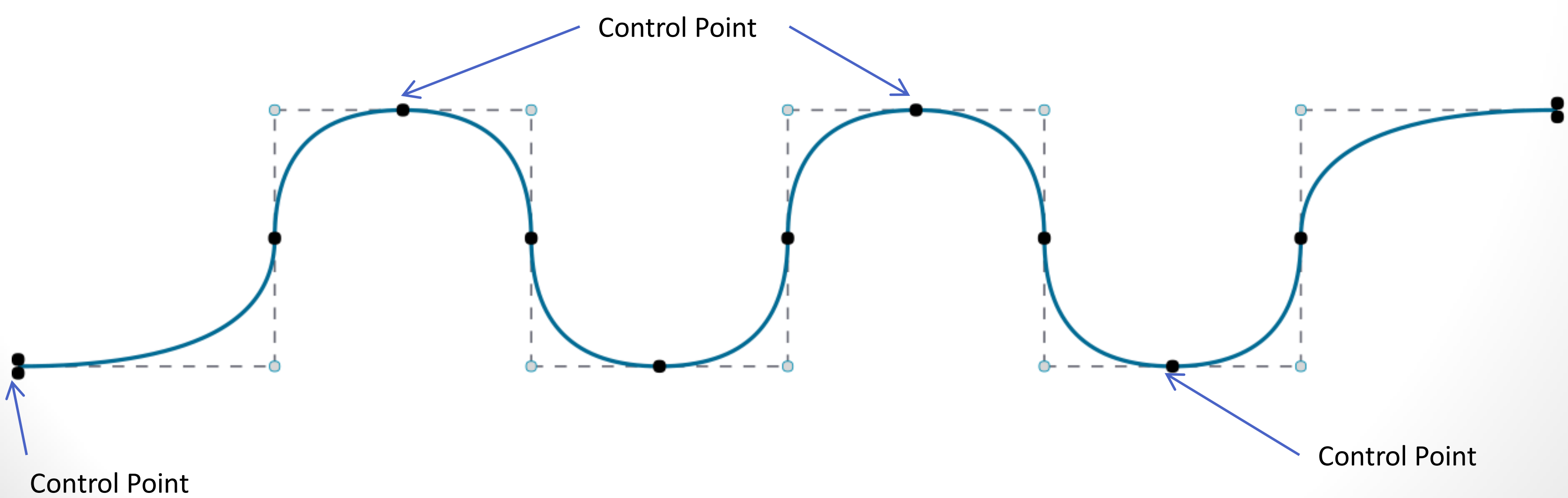
$$\text{where } r_k^2 = x_k^2 + y_k^2 + z_k^2$$

Still More Blobby Objects

- Another technique called the **meta-ball** technique is to describe the object as being made of density functions much like balls.
- The advantage here is that the density function falls of in a finite interval.

Spline Representation

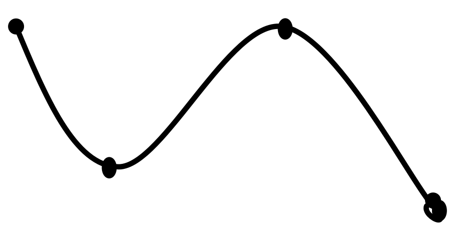
- A spline curve is a mathematical representation for which it is easy to build an interface that will allow a user to design and control the shape of complex curves and shapes
- The general approach is that the user enters a sequence of points and a curve is constructed whose shape closely follows this sequence . The point are called **control point**.



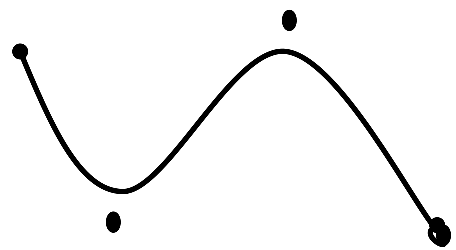
Nipun Thapa (computer Graphics)

Spline Representation

- A curve is actually passes through each control point is called ***interpolating curve***
- A curve that passes near to the control point but not necessarily through them is called an ***approximating curve***.



Interpolating Curves



Approximation Curves

Spline Representation

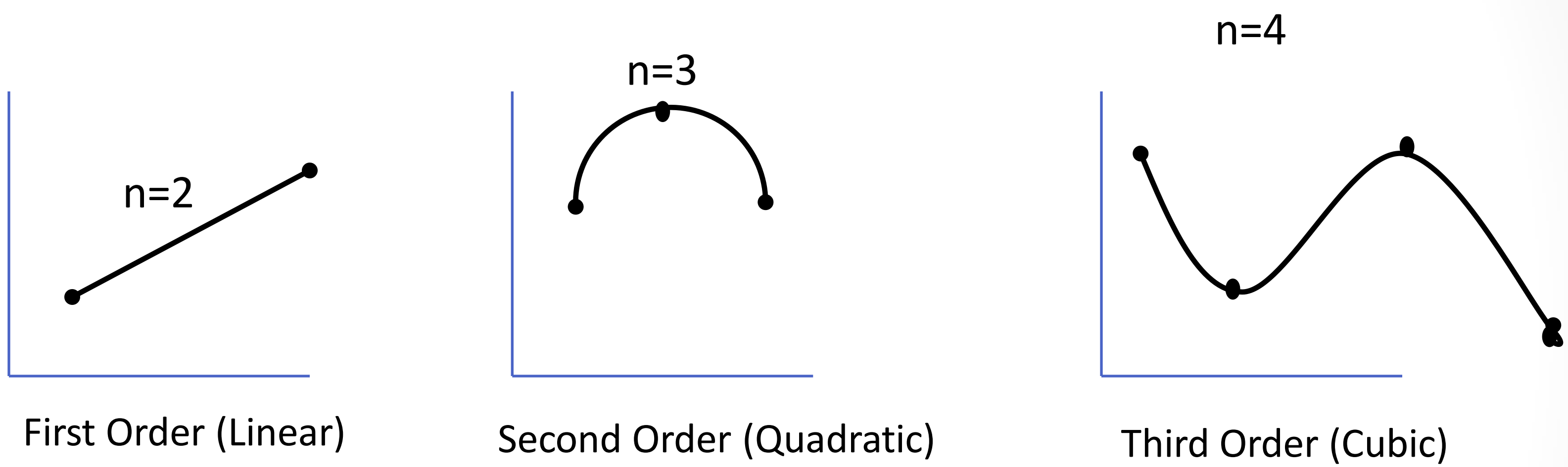
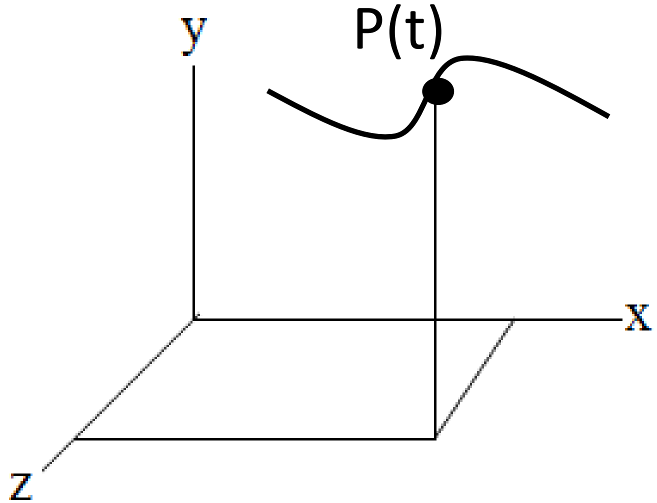


Fig : Interpolating

Parametric Cubic Curve

- A parametric cubic curve is defined as

$$P(t) = \sum_{i=0}^{3} a_i t^i \qquad 0 <= t <= 1 \quad \text{------} \ (i)$$



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

a= algebraic coefficients

t= tangent Vector

Parametric Cubic Curve

Expanding equation (i) yield

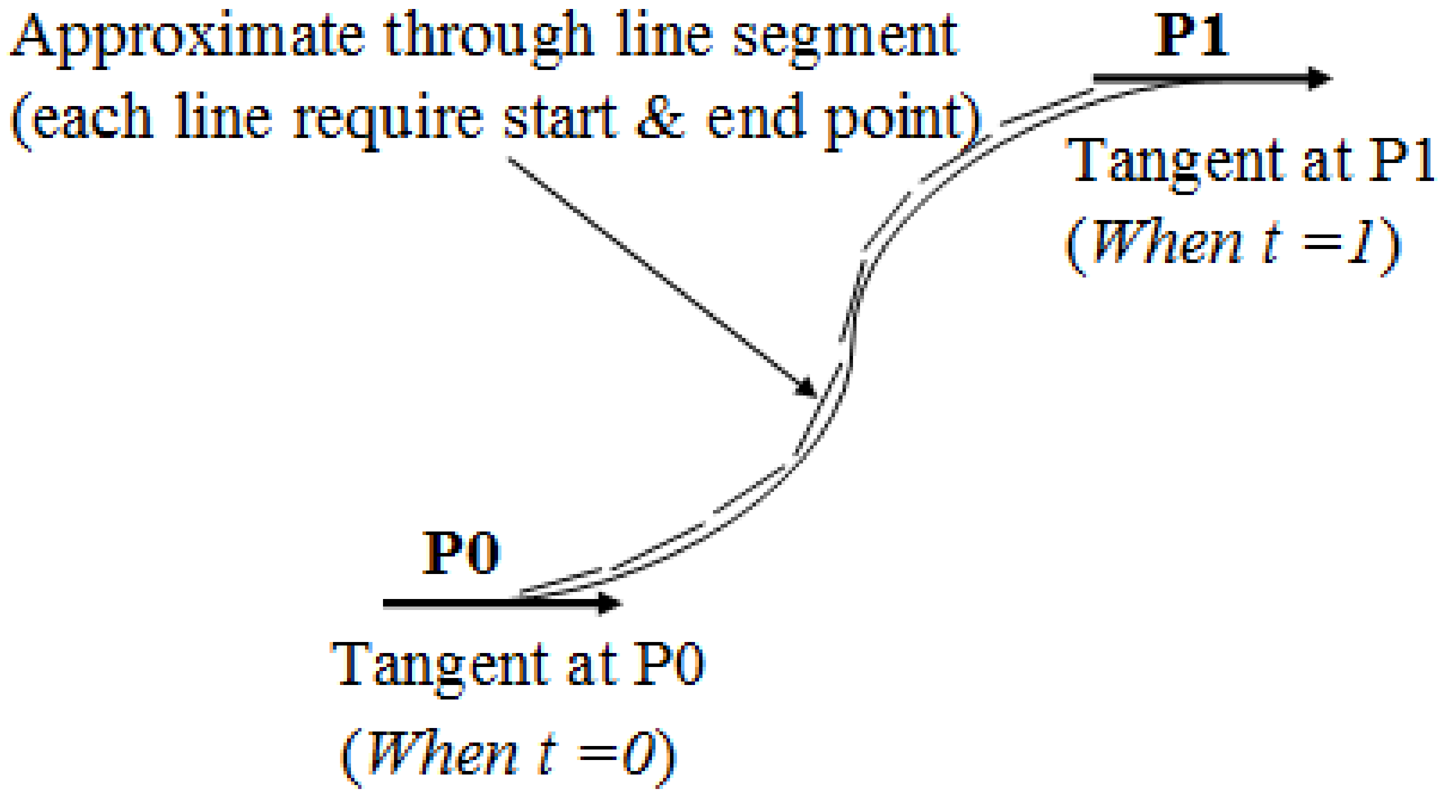
$$P\ (t) = a_3\, t^3 + a_2\, t^2 + a_1\, t + a_0 \text{ ------------------------------ (ii)}$$

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

$$x\ (t) = a_{3x}\, t^3 + a_{2x}\, t^2 + a_{1x}\, t + a_{0x}$$
$$y\ (t) = a_{3y}\, t^3 + a_{2y}\, t^2 + a_{1y}\, t + a_{0y}$$
$$z\ (t) = a_{3z}\, t^3 + a_{2z}\, t^2 + a_{1z}\, t + a_{0z} \qquad \text{---------------------(iii)}$$

- To be able to solve (iii) the **twelve unknown** coefficients **aij** (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 point

Parametric Cubic Curve



Parametric Cubic Curve

- 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 relationship between two end point vectors and the algebraic coefficients are found

$$P(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

$$\mathbf{P(0) = a_0} \qquad \mathbf{P(1) = a_3 + a_2 + a_1 + a_0} \text{ ----- ---- ---- (IV)}$$

Parametric Cubic Curve

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

$$P(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

$$\mathbf{P'(t) = 3 a_3 t^2 + 2 a_2 t + a_1}$$

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

$\mathbf{P'(0) = a_1}$ $\mathbf{P'(1) = 3 a_3 + 2 a_2 + a_1}$------------------ (V)

Parametric Cubic Curve

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

  a0= P (*0*) **a1= P' (0)**

  a2= -3 P (0) - 3 P (1) -2 P' (0) - P' (1)

  a3 = 2 P (0) - 2 P (1) + P' (0) + P' (1)

  (Note: - The value of a2 & a3 can be determined by solving the equation IV & V)

- 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

Bezier Curve and Surfaces

Bezier splines are highly useful, easy to implement and convenient for curve and surface design so are widely available in various CAD systems, graphics packages, drawing and painting packages.

Bezier curve

In general, a Bezier curve can be fitted to any number of control points. The number of control points to be ***approximated*** and their relative position determine the degree of the Bezier polynomial. As with the interpolation splines, a Bezier curve can be specified with boundary conditions, with a characterizing matrix, or with ***blending functions***.

Bezier Curve

The Bezier curve has two important properties:

1. It always passes through the first and last control points.
2. It lies within the convex hull (convex polynomial boundary) of the control points. This follows from the properties of Bezier blending function: they are positive and their sum is always 1, i.e.

$$\sum_{k=0}^{n} z_k \, BEZ_{k,n}(u) = 1$$

Bezier Curve

Suppose we are given n + 1 control-point positions:

pk = (x_k, y_k, z_k), with k varying from 0 to n. These coordinate points can be blended to produce the following position vector P(u), which describes the path of an approximating Bezier polynomial function between p_0 and p_n.

$$\mathbf{P}(u) = \sum_{k=0}^{n} \mathbf{p}_k \, BEZ_{k,n}(u), \qquad 0 \le u \le 1$$

Bezier Curve

This, vector equation represents a set of three parametric equations for the individual curve coordinates:

$$\mathbf{x}(u) = \sum_{k=0}^{n} \mathbf{x}_k \, BEZ_{k,n}(u)$$

$$\mathbf{y}(u) = \sum_{k=0}^{n} \mathbf{y}_k \, BEZ_{k,n}(u)$$

$$\mathbf{z}(u) = \sum_{k=0}^{n} \mathbf{z}_k \, BEZ_{k,n}(u)$$

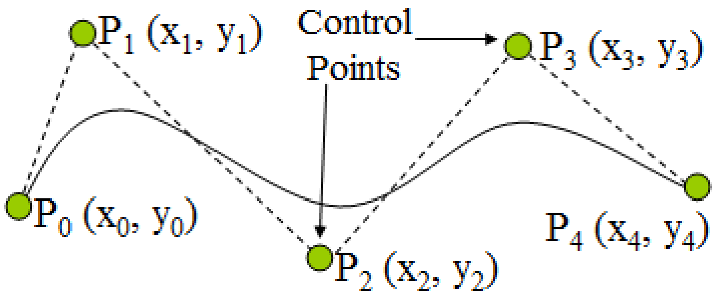
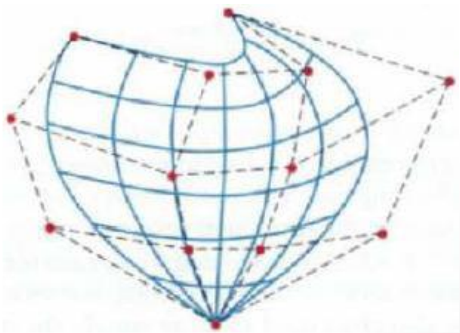


Fig. Bezier curve with 4-control points in xy-plane (z=0)



The Bezier blending functions $BEZ_{k,n}(u)$ are the Bernstein polynomials:

$$BEZ_{k,n}(u) = C(n, k)\, u^k (1-u)^{n-k}$$

Where the ***C (n, k)*** are the binomial coefficients:

$$C(n, k) = \frac{n!}{k!\,(n-k)!}$$

Nipun Thapa (computer Graphics)

Bezier Curve

Now

$$\mathbf{P}(u) = \sum_{k=0}^{n} \mathbf{p}_k \, BEZ_{k,n}(u), \qquad 0 \le u \le 1$$

(n = 3)

Then ,

$P(u) = P_0 \, BEZ_{0,3}(u) + P_1 \, BEZ_{1,3}(u) + P_2 \, BEZ_{2,3}(u) + P_3 \, BEZ_{3,3}(u)$

Four blending functions must be found based on Bernstein Polynomials

$$BEZ_{0,3}(u) = \frac{3!}{0!\,3!} u^0 (1-u)^3 = (1-u)^3 \qquad BEZ_{1,3}(u) = \frac{3!}{1!\,2!} u^1 (1-u)^2 = 3u(1-u)^2$$

$$BEZ_{2,3}(u) = \frac{3!}{2!\,1!} u^2 (1-u) = 3u^2(1-u) \qquad BEZ_{3,3}(u) = \frac{3!}{3!\,0!} u^3 (1-u)^0 = u^3$$

Bezier Curve

Normalizing properties apply to blending function s that means thy all add up to one Substituting these functions in above equation

$P(u) = (1-u)^3 P_0 + 3u (1-u)^2 P_1 + 3u^2 (1-u) P_2 + u^3 P_3$

When $u = 0$ then $P(u) = P_0$ and

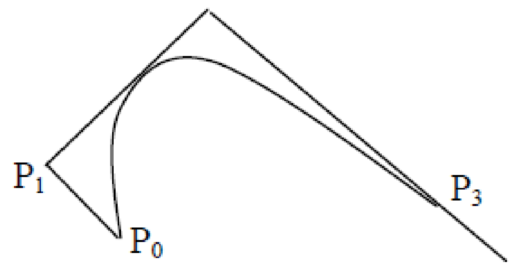
when $u = 1$ then $P(u) = P_3$

in Matrix Form

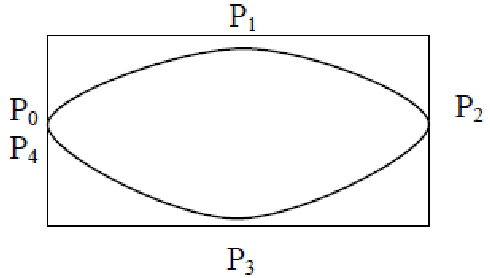
$$P(u) = \begin{bmatrix} (1-u)^3 & 3u(1-u)^2 & 3u^2(1-u) & u^3 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

Properties Of Bezier Curve

1. Bezier curve lies in the convex hull of the control points which ensure that the curve smoothly follows the control Points



2. Four Bezier polynomials are used in the construction of curve to fit four control points
3. It always passes thru the end points
4. Closed curves can be generated by specifying the first and last control points at the same position



Properties Of Bezier Curve

5. Specifying multiple control points at a single position gives more weight to that position
6. Complicated curves are formed by piecing several sections of lower degrees together
7. The tangent to the curve at an end point is along the line joining the end point to the adjacent control point

Example

Q.N. > Calculate (x,y) coordinates of Bézier curve described by the following 4 control points: (0,0), (1,2), (3,3), (4,0).

Step by step solution

For four control points, n = 3.

First calculate all the blending functions, Bkn for k=0,..,n using the formula:

$$B_{kn}(u) = C(n,k)\,u^k(1-u)^{n-k} = \frac{n!}{k!\cdot(n-k)!}\,u^k(1-u)^{n-k}$$

$$B_{03}(u) = \frac{3!}{0!\cdot 3!}\,u^0(1-u)^3 = 1\cdot u^0(1-u)^3 = \mathbf{(1-u)^3}$$

$$B_{13}(u) = \frac{3!}{1!\cdot 2!}\,u^1(1-u)^2 = 3\cdot u^1(1-u)^2 = \mathbf{3u\cdot(1-u)^2}$$

$$B_{23}(u) = \frac{3!}{2!\cdot 1!}\,u^2(1-u)^1 = 3\cdot u^2(1-u)^1 = \mathbf{3u^2(1-u)}$$

$$B_{33}(u) = \frac{3!}{3!\cdot 0!}\,u^3(1-u)^0 = 1\cdot u^3(1-u)^0 = \mathbf{u^3}$$

Numerical calculations are shown below:

$$\underline{u = 0.0}\ x(0) = \sum_{k=0}^{n} x_k B_{k\,n}(0) = x_0 B_{0\,3}(0) + x_1 B_{13}(0) + x_2 B_{23}(0) + x_3 B_{33}(0) =$$

$$= 0\cdot(1 - u)^3 + 1\cdot 3u\cdot(1 - u)^2 + 3\cdot 3u^2(1 - u) + 4\cdot u^3 =$$

$$= 0\cdot 1 + 1\cdot 0 + 3\cdot 0 + 4\cdot 0 =$$

$$= 0$$

$$y(0) = \sum_{k=0}^{n} y_k B_{k\,n}(0) = y_0 B_{0\,3}(0) + y_1 B_{1\,3}(0) + y_2 B_{2\,3}(0) + y_3 B_{3\,3}(0) =$$

$$= 0\cdot(1 - u)^3 + 2\cdot 3u\cdot(1 - u)^2 + 3\cdot 3u^2(1 - u) + 0\cdot u^3 =$$

$$= 0$$

<u>u = 0.2</u> $x(0.2) = \sum_{k=0}^{n} x_k B_{kn}(0.2) = x_0 B_{03}(0.2) + x_1 B_{13}(0.2) + x_2 B_{23}(0.2) + x_3 B_{33}(0.2) =$

$= 0\cdot(1 - u)^3 + 1\cdot 3u\cdot(1 - u)^2 + 3\cdot 3u^2 (1 - u) + 4\cdot u^3 =$

$= 0\cdot 0.512 + 1\cdot 0.384 + 3\cdot 0.096 + 4\cdot 0.008 =$

$= 0.7$

$y(0.2) = \sum_{k=0}^{n} y_k B_{kn}(0.2) = y_0 B_{03}(0.2) + y_1 B_{13}(0.2) + y_2 B_{23}(0.2) + y_3 B_{33}(0.2) =$

$= 0\cdot(1 - u)^3 + 2\cdot 3u\cdot(1 - u)^2 + 3\cdot 3u^2 (1 - u) + 0\cdot u^3 =$

$= 0\cdot 0.512 + 2\cdot 0.384 + 3\cdot 0.096 + 0\cdot 0.008 =$

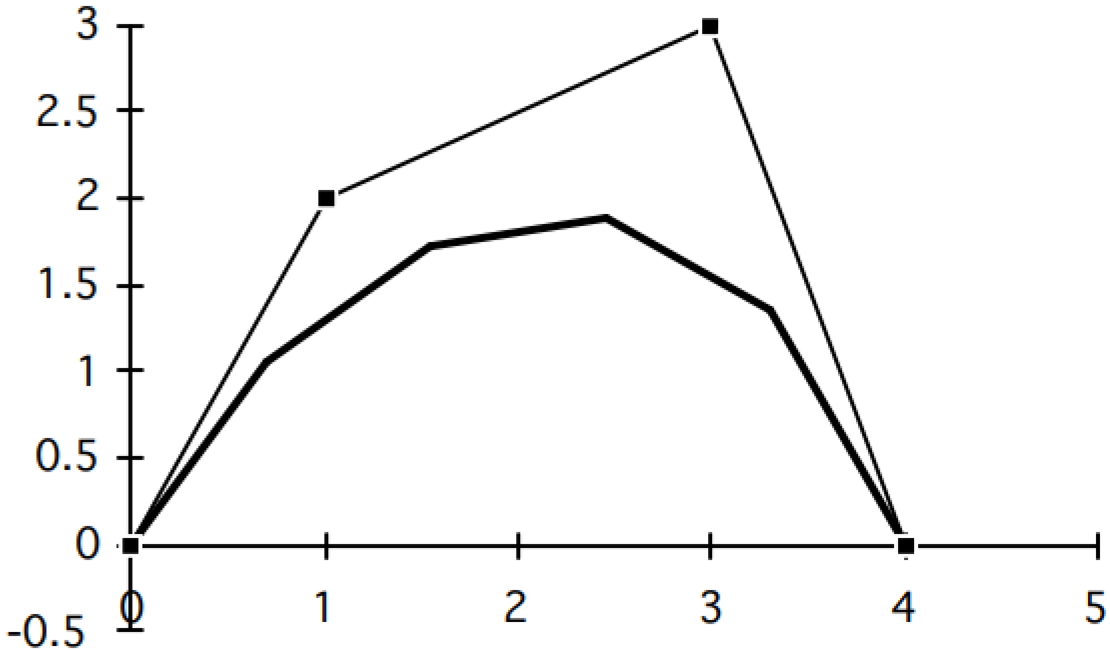
$= 1.1$

etc, giving:

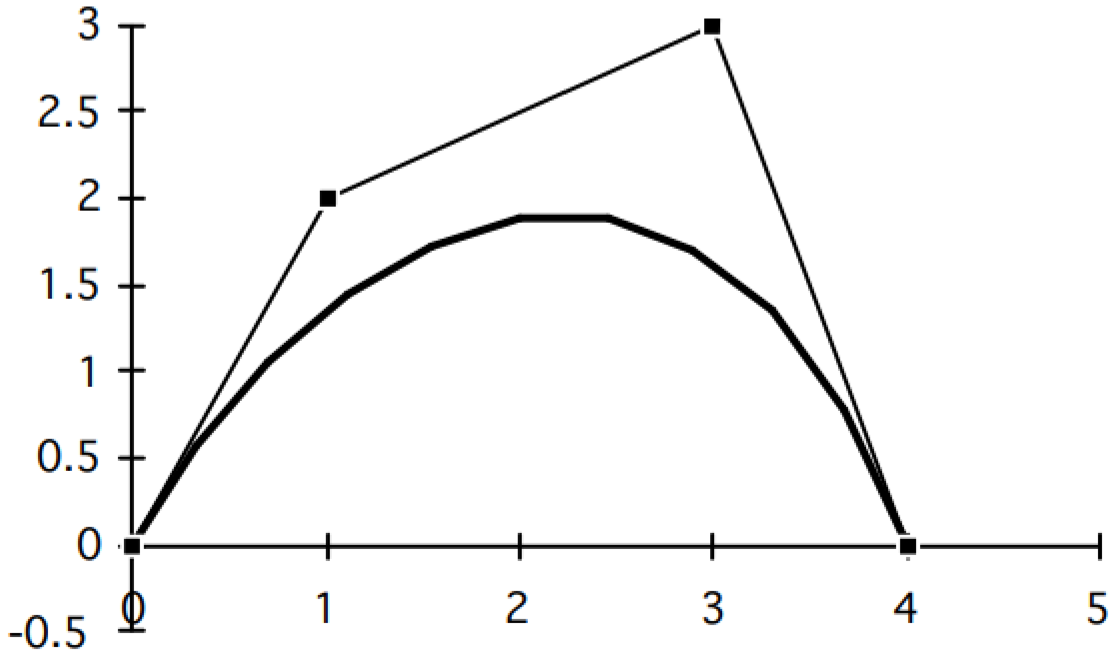
$u = 0.0$ $x(u) = 0.0$	$y(u) = 0.0$
$u = 0.2$ $x(u) = 0.7$	$y(u) = 1.1$
$u = 0.4$ $x(u) = 1.55$	$y(u) = 1.7$
$u = 0.6$ $x(u) = 2.45$	$y(u) = 1.9$
$u = 0.8$ $x(u) = 3.3$	$y(u) = 1.3$
$u = 1.0$ $x(u) = 4.0$	$y(u) = 0.0$

$(\, x(u), y(u) \,)_{u = 0,1}$ are coordinates of the curve points.

The plot below shows control points (joined with a thin line) and a Bézier curve with 6 steps



The plot below shows control points (joined with a thin line) and a Bézier curve with 11 steps; note smoother appearance of this curve in comparison to the previous one.



Q.N.>Construct the Bezier curve of order 3 and 4 polygon vertices A(1,1) B(2,3) C(4,3) and D(6,4). ***(TU 2072)***

Quadric surfaces

- If a surface is the graph in three-space of an equation of ***second degree***, it is called a quadric surface. Cross section of quadric surface are conics.
- 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:

Quadric surfaces

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 : $\frac{x^2}{a^2}-\frac{y^2}{b^2}=z$

Quadric surfaces

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: $\frac{x^2}{a^2}-\frac{y^2}{b^2}-\frac{z^2}{c^2}=1$

Chapter 5 Finished