3-D Graphics

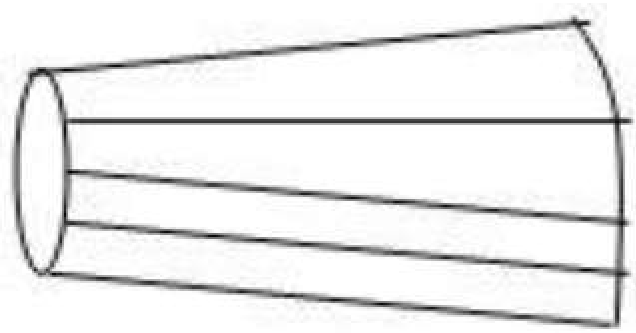
Polygon Surfaces

Objects are represented as a collection of surfaces. 3D object representation is divided into two categories.

- **Boundary Representations** B−repsB−reps − It describes a 3D object as a set of surfaces that separates the object interior from the environment.
- **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 usuallycubesusuallycubes.

The most commonly used boundary representation for a 3D graphics object is a set of surface polygons 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.

The polygon surfaces are common in design and solid-modeling applications, since their **wireframe 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.

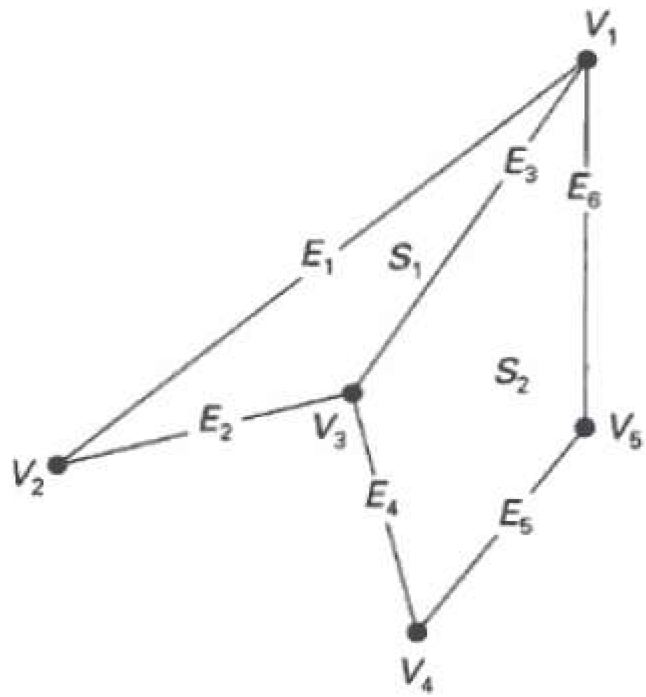


A 3D object represented by polygons

Polygon Tables

In this method, the surface is specified by the set of vertex coordinates and associated attributes. As shown in the following figure, there are five vertices, from v_1 to v_5.

- Each vertex stores x, y, and z coordinate information which is represented in the table as v_1: x_1, y_1, z_1.
- The Edge table is used to store the edge information of polygon. In the following figure, edge E_1 lies between vertex v_1 and v_2 which is represented in the table as E_1: v_1, v_2.
- Polygon surface table stores the number of surfaces present in the polygon. From the following figure, surface S_1 is covered by edges E_1, E_2 and E_3 which can be represented in the polygon surface table as S_1: E_1, E_2, and E_3.



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

Plane Equations

The equation for plane surface can be expressed as −

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

Where x,y,zx,y,z is any point on the plane, and the coefficients A, B, C, and D are constants describing the spatial properties of the plane. We can obtain the values of A, B, C, and D by solving a set of three plane equations using the coordinate values for three non collinear points in the plane. Let us assume that three vertices of the plane are (x_1, y_1, z_1), (x_2, y_2, z_2) and (x_3, y_3, z_3).

Let us solve the following simultaneous equations for ratios A/D, B/D, and C/D. You get the values of A, B, C, and D.

$$A/D\ x_1 + B/D\ y_1 + C/D\ z_1 = -1$$
$$A/D\ x_2 + B/D\ y_2 + C/D\ z_2 = -1$$
$$A/D\ x_3 + B/D\ y_3 + C/D\ z_3 = -1$$

To obtain the above equations in determinant form, apply Cramer's rule to the above equations.

A=[⌊|111y1y2y3z1z2z3]⌋|B=[⌊|x1x2x3111z1z2z3]⌋|C=[⌊|x1x2x3y1y2y3111]⌋|
D=−[⌊|x1x2x3y1y2y3z1z2z3]⌋|A=[1y1z11y2z21y3z3]B=[x11z1x21z2x31z3]
C=[x1y11x2y21x3y31]D=−[x1y1z1x2y2z2x3y3z3]
For any point x,y,zx,y,z with parameters A, B, C, and D, we can say that –

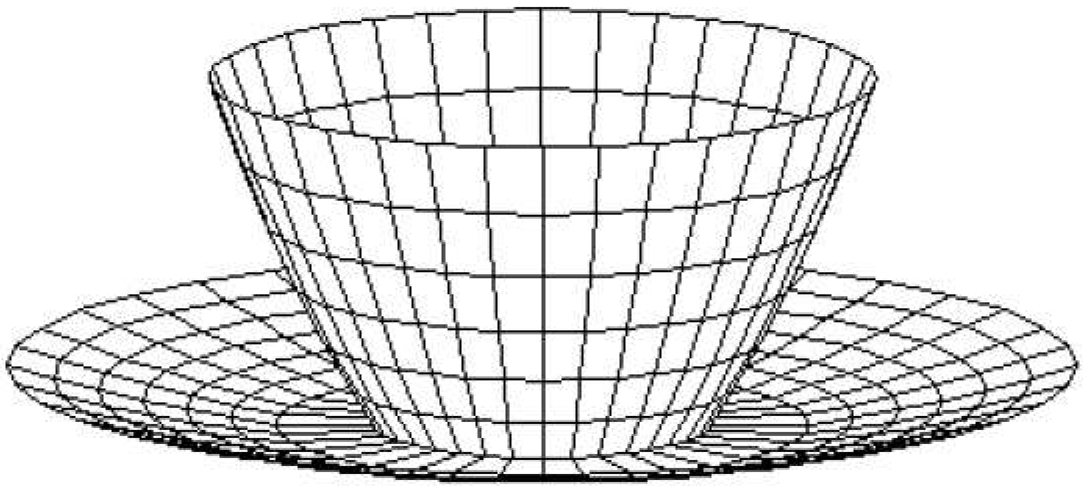
- $Ax + By + Cz + D \neq 0$ means the point is not on the plane.
- $Ax + By + Cz + D < 0$ means the point is inside the surface.
- $Ax + By + Cz + D > 0$ means the point is outside the surface.

Polygon Meshes

3D surfaces and solids can be approximated by a set of polygonal and line elements. Such surfaces are called **polygonal meshes**. In polygon mesh, each edge is shared by at most two polygons. The set of polygons or faces, together form the "skin" of the object.

This method can be used to represent a broad class of solids/surfaces in graphics. A polygonal mesh can be rendered using hidden surface removal algorithms. The polygon mesh can be represented by three ways –

- Explicit representation
- Pointers to a vertex list
- Pointers to an edge list



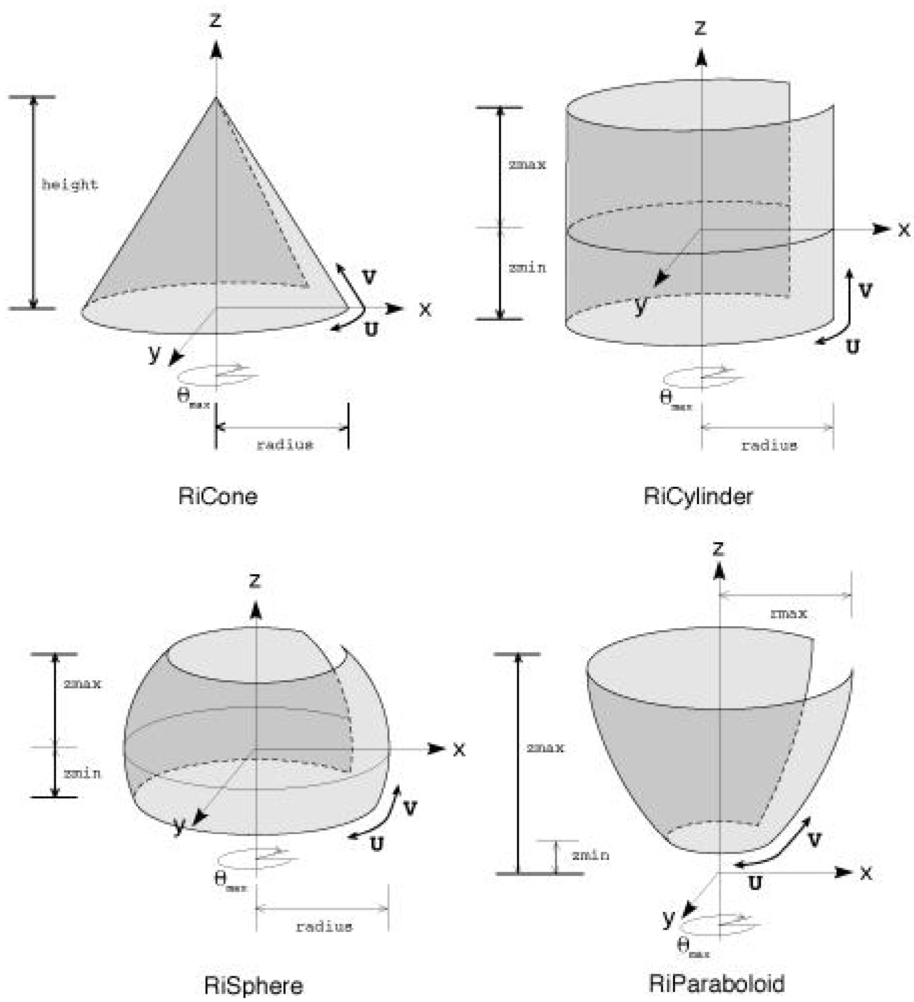
Advantages

- It can be used to model almost any object.
- They are easy to represent as a collection of vertices.
- They are easy to transform.
- They are easy to draw on computer screen.

Disadvantages

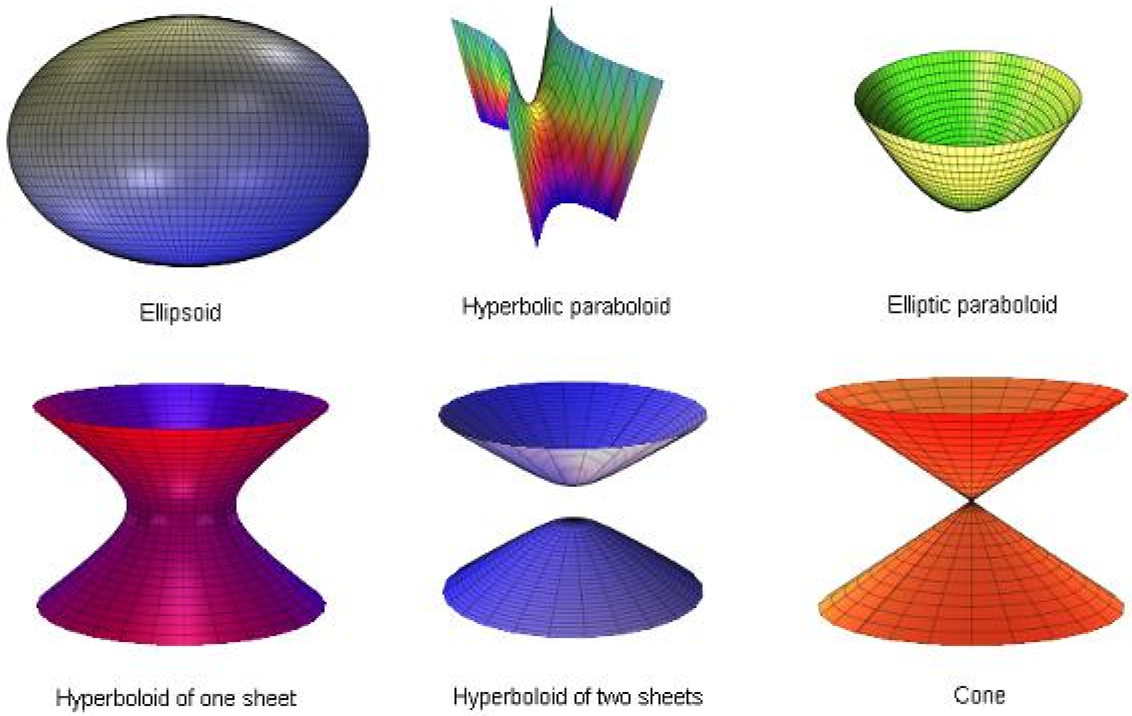
- Curved surfaces can only be approximately described.
- It is difficult to simulate some type of objects like hair or liquid.

Quadric surfaces



1. A frequently used class of objects is the quadric surfaces.
2. These are described with second degree equations – quadratics.
3. They include spheres, ellipsoids, torus, paraboloids and hyperboloids.
4. These quadric surfaces are often available in graphic packages as

primitives from which more complex objects can be constructed.



Spline representations

- Spline representations are used in drafting terminology.
- Drafting terminology is a tool in CAD used to create drawings in computer instead of hand.
- Spline is a flexible strip used to produce a smooth curve through a designated set of points.
- Several small weights are distributed along the length of the strip to hold it in position on the drafting table as the curve is drawn.
- Mathematically, we describe spline curve as a curve with a piecewise

cubic polynomial function.

Splines are used in graphics applications

1. To design curve and surface shapes
2. To digitize drawings for computer storage
3. To specify animation paths for the objects in a scene

Typical CAD applications for splines include

1. The design of automobile bodies
2. Aircrafts surfaces
3. Spacecrafts surfaces
4. Ship hulls

Hermite curve

A Hermite curve is a spline where every piece is a third degree polynomial defined in Hermite form: that is, by its values and initial derivatives at the end points of the equivalent domain interval. Cubic Hermite splines are normally used for interpolation of numeric values defined at certain dispute values x1,x2,x3, …..xn,to achieve a smooth continuous function. The data should have the preferred function value and derivative at each Xk. The Hermite formula is used to every interval (Xk, Xk+1) individually. The resulting spline become continuous and will have first derivative.

Cubic polynomial splines are specially used in computer geometric modeling to attain curves that pass via defined points of the plane in 3D space. In these purposes, each coordinate of the plane is individually interpolated by a cubic spline function of a divided parameter‘t’.

Cubic splines can be completed to functions of different parameters, in several ways. Bicubic splines are frequently used to interpolate data on a common rectangular grid, such as pixel values in a digital picture. Bicubic surface patches, described by three bicubic splines, are an necessary tool in computer graphics. Hermite curves are simple to calculate but also more powerful. They are used to well interpolate between key points.

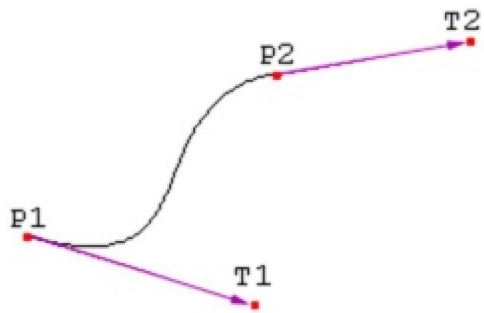


Fig.2.2. Hermite curve

The following vectors needs to compute a Hermite curve:

- P1: the start point of the Hermite curve
- T1: the tangent to the start point
- P2: the endpoint of the Hermite curve
- T2: the tangent to the endpoint

These four vectors are basically multiplied with four Hermite basis functions h1(s), h2(s), h3(s) and,h4(s) and added together.

h1(s) = 2s3 - 3s2 + 1 h2(s) = -2s3 + 3s2

h3(s) = s3 - 2s2 + s h4(s) = s3 - s2

Figure 2.3 shows the functions of Hermite Curve of the 4 functions (from left to right: h1, h2, h3, h4).

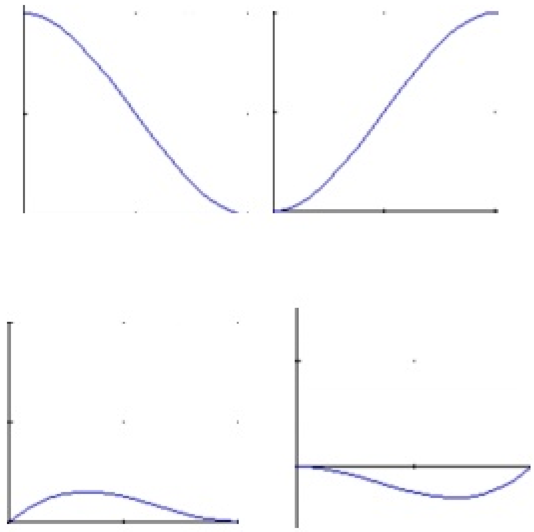


Fig.2.3. Functions of Hermite curve

Fig.2.3. Functions of Hermite curve

A closer view at functions ‘h1’and ‘h2’,the result shows that function ‘h1’starts at one and goes slowly to zero and function ‘h2’starts at zero and goes slowly to one.

At the moment, multiply the start point with function ‘h1’and the endpoint with function ‘h2’. Let s varies from zero to one to interpolate between start and endpoint of Hermite Curve. Function

‘h3’and function ‘h4’are used to the tangents in the similar way. They make confident that the Hermite curve bends in the desired direction at the start and endpoint.

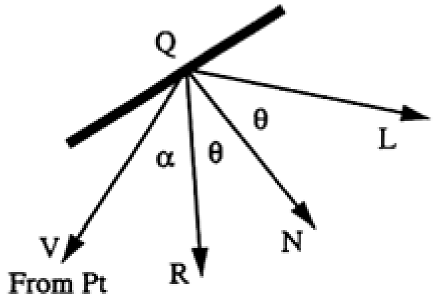
POLYGON RENDERING METHODS

- **RENDERING** means giving proper intensity at each point in a graphical object to make it look like real world object.
- Different Rendering Methods are
  - Constant Intensity Shading
  - Gouraud Shading
  - Phong Shading
  - Fast Phong Shading

Illumination Models

Three Components to Illumination of Objects

- Ambient - Total of all the light bouncing.
- Diffuse reflection - Matted surfaces.
- Specular reflection - Mirror effect.



Ambient

$$I_a = \begin{bmatrix} I_{ar} \\ I_{ag} \\ I_{ab} \end{bmatrix}$$ *- intensity of the ambient light.*

$$k_a = \begin{bmatrix} k_{ar} \\ k_{ag} \\ k_{ab} \end{bmatrix}$$ *- ambient reflection coefficient.*

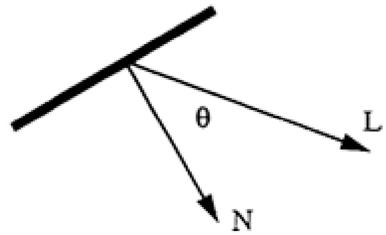
I_a is a property of the scene and

k_a is a material property (varies with object) Illumination (due to ambient).

$$I = I_a k_a = \begin{bmatrix} I_{ar}\, k_{ar} \\ I_{ag}\, k_{ag} \\ I_{ab}\, k_{ab} \end{bmatrix}$$

Diffuse Reflection

The light that gets reflected from the object's surface in all directions when a light source illuminates its surface. Based on Lamberts Law for reflected light off a matte surface.



I_p point light source's intensity k_d material's diffuse-reflection coefficient

Illumination: $I = I_p k_d \cos(\theta)$

Assumptions: $0 \leq \theta \leq p$ otherwise diffuse contribution is

i zero.

$\overline{2}$

Normalized vectors $||N|| = ||L|| = 1$

$\cos(\theta) = (N \cdot L)$

Ambient and Diffuse

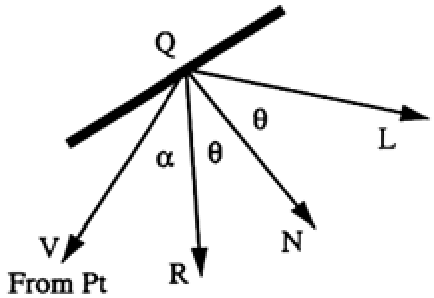
Illumination:

$$I = I_a k_a + I_p k_d (N \cdot L)$$

$$(N \cdot L) = \begin{cases} (N \cdot L), & \text{if } (N \cdot L) \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

Specular reflection

The glare seen when a light source is mirrored on the surface of a shiny object Phong Model



k_s specular reflection coefficien

n Phong constant

Illumination:

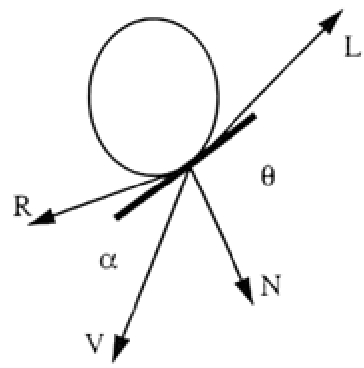
$$I = I_a k_a + I_p \left[k_d \cdot \cos(\theta) + k_s \cdot \cos^n(\alpha) \right]$$

Calculate once for red, once for green and once for blue.

Computational considerations

$$\cos(\theta) = \frac{(L \cdot N)}{||L|| \cdot ||N||}$$ *(dot product)*

$$\cos(\alpha) = \frac{(R \cdot V)}{||R|| \cdot ||V||}$$ *(dot product)*

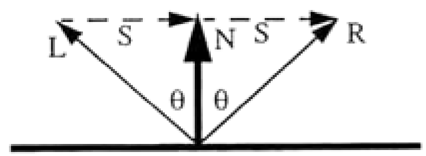


Diffuse requires: requires *$\cos(\theta) \geq 0$*

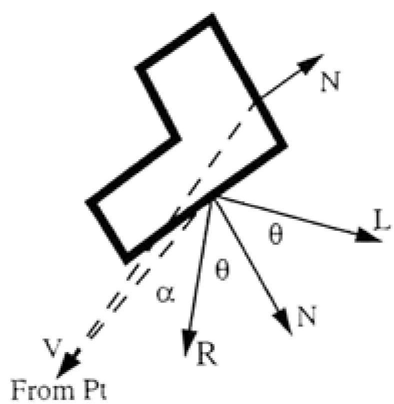
Specular requires: recuires *$\cos(\alpha) \geq 0$* and *$\cos(\theta) \geq 0$*

Calculation Reflection Vector

$$R = 2N(N \cdot L) - L$$



Backface culling (and inside/outside coloring)



Backface culling:
If *$(V \cdot N) < 0$* *then the polygon cannot be seen.*

In/Out Coloring:
If *$(V \cdot N) \geq 0$* then object is color.out

else object is color.in with (outward) normal *-N*

Bezier Curves

Bezier curve is defined in terms of the locations of n+1 points. These points are called data or control points. They form the vertices of what is called the control or Bezier characteristic polygon.

In a Bezier curve, only the first and the last control points or vertices of the polygon actually lie on the curve. The curve is also always tangent to the first and last polygon segments. In addition, the curve shape tends to follow the polygon shape. These three observations should enable the user to sketch or predict the curve shape once its control points are given.

Parametric equations of the Bezier curve

In general, a Bezier curve section can be fitted to any number of control points. The number of control points to be approximated determines the degree of the Bezier curve. For n+1 control points, the Bezier curve is defined by the following polynomial of degree n is also called blending function.

$$\mathbf{P}(u) = \sum_{i=0}^{n} \mathbf{P}_i B_{i,n}(u), \quad 0 \leq u \leq 1 \tag{1}$$

where $\mathbf{P}(u)$ is any point on the curve and $\mathbf{P}_i$ is a control point, $B_{i,n}$ are the Bernstein polynomials. The Bernstein polynomial serves as the blending or basis function for the Bezier curve and is given by

$$B_{i,n}(u) = C(n,i)u^i(1-u)^{n-i} \tag{2}$$

where $C(n,i)$ is the binomial coefficient

$$C(n,i) = \frac{n!}{i!(n-i)!} \tag{3}$$

Equation (1) can be expanded to give

$$\begin{aligned}\mathbf{P}(u) = \mathbf{P}_0(1-u)^n + \mathbf{P}_1C(n,1)u(1-u)^{n-1} + \mathbf{P}_2C(n,2)u^2(1-u)^{n-2} \\ + \cdots + \mathbf{P}_{n-1}C(n,n-1)u^{n-1}(1-u) + \mathbf{P}_nu^n, \quad 0 \le u \le 1\end{aligned} \tag{4}$$

From Equation (2) and (3), we can get the followings:

$n = 0$, $B_{0,0}(u) = 1$

$n = 1$, $B_{0,1}(u) = (1-u)$

$B_{1,1}(u) = u$

$n = 2$, $B_{0,2}(u) = (1-u)^2$

$B_{1,2}(u) = 2u(1-u)$

$B_{2,2}(u) = u^2$

$n = 3$, $B_{0,3}(u) = (1-u)^3$

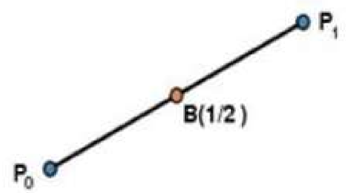
$B_{1,3}(u) = 3u(1-u)^2$

$B_{2,3}(u) = 3u^2(1-u)$

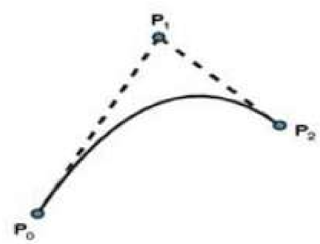
$B_{3,3}(u) = u^3$

Therefore for $n = 3$,

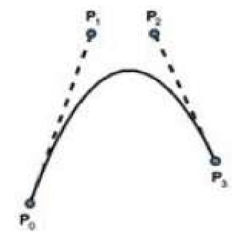
$$\mathbf{P}(u) = (1-u)^3\mathbf{P}_0 + 3u(1-u)^2\mathbf{P}_1 + 3u^2(1-u)\mathbf{P}_2 + u^3\mathbf{P}_3 . \quad (5)$$



Simple Bezier Curve



Quadratic Bazier Curve



Cubic Bazier Curve

Properties of Bezier Curves

Bezier curves have the following properties –

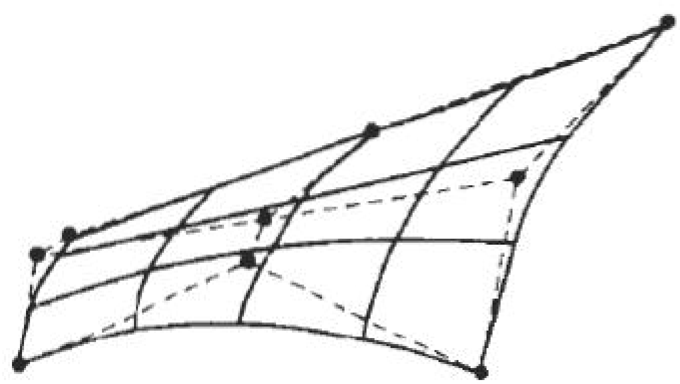
- A very useful property of a Bezier is that it always passes through the first and last control points
- They generally follow the shape of the control polygon, which consists of the segments joining the control points.
- The degree of the polynomial defining the curve segment is one less that the number of defining polygon point. Therefore, for 4 control points, the degree of the polynomial is 3, i.e. cubic polynomial.
- They are contained in the convex hull of their defining control points.
- A Bezier curve generally follows the shape of the defining polygon.
- The direction of the tangent vector at the end points is same as that of the vector determined by first and last segments.
- The convex hull property for a Bezier curve ensures that the polynomial smoothly follows the control points.
- No straight line intersects a Bezier curve more times than it intersects its control polygon.

Bezier surfaces

- The 3-D bezier curves can be used to design an object surface by specifying by an input mesh of control points.
- The bezier blending function can be written as,

$$\mathbf{P}(u, v) = \sum_{j=0}^{m} \sum_{k=0}^{n} \mathbf{p}_{j,k} BEZ_{j,m}(v) BEZ_{k,n}(u)$$

- Bezier surfaces have the same properties as bezier curves.
- They provide a convenient method for interactive design application.
- Bezier surfaces constructed for m=3, n=3



B-Spline Curves

B-spline curves are powerful generalization of Bezier curves. They provide local control of the curve shape as opposed to global control by using a special set of blending functions that provide local influence. They also provide the ability to add control points without increasing the degree of the curve.

B-spline curves have the ability to interpolate or approximate a set of given data points.

The theory of B-spline curves separates the degree of the resulting curve from the number of the given control points. The B-spline curve defined by n+1 control points P_i is given by

$$\mathbf{P}(u)=\sum_{i=0}^{n}\mathbf{P}_i N_{i,k}(u),\quad 0\le u\le u_{max} \tag{1}$$

where $N_{i,k}(u)$ are the B-spline functions. First, the parameter k controls the degree (k-1) of the resulting B-spline curve and is usually independent of the number of control points. The B-spline functions have the following properties:

Partition of unity: $\sum_{i=0}^{n} N_{i,k}(u)=1$

Positivity: $N_{i,k}(u)\ge 0$

Local support: $N_{i,k}(u)=0$ if $u\notin[u_i, u_{i+k+1}]$

Continuity: $N_{i,k}(u)$ is (k-2) times continuously differentiable

The B-spline function also has the property of recursion, which is defined as

$$N_{i,k}(u)=(u-u_i)\frac{N_{i,k-1}(u)}{u_{i+k-1}-u_i}+(u_{i+k}-u)\frac{N_{i+1,k-1}(u)}{u_{i+k}-u_{i+1}} \tag{2}$$

where

$$N_{i,1}=\begin{cases}1, & u_i\le u\le u_{i+1}\\ 0, & \text{otherwise}\end{cases} \tag{3}$$

The u_i are called parametric knots or knot values. For an open curve,

$$u_j = \begin{cases} 0, & j < k \\ j-k+1, & k \le j \le n \\ n-k+2, & j > n \end{cases} \tag{4}$$

where

$$0 \le j \le n+k \tag{5}$$

and the range of *u* is

$$0 \le u \le n-k+2 \tag{6}$$

Relation (5) shows that (n+k+1) knots are needed to create a (k-1) degree curve defined by (n+1) control points.

$$n-k+2 > 0 \tag{7}$$

This relation shows that a minimum of two, three, and four control points are required to define a linear, quadratic, and cubic B-spline curve respectively. A cubic B-spline is sufficient for a large number of applications. Figure 2 shows the shapes of the B-spline functions.

Properties of B-spline Curve

B-spline curves have the following properties −

- B-spline allows the local control over the curve surface because each vertex affects the shape of a curve only over a range of parameter values where its associated basis function is nonzero.
- The maximum order of the curve is equal to the number of vertices of defining polygon.
- The degree of B-spline polynomial is independent on the number of vertices of defining polygon.
- The sum of the B-spline basis functions for any parameter value is 1.

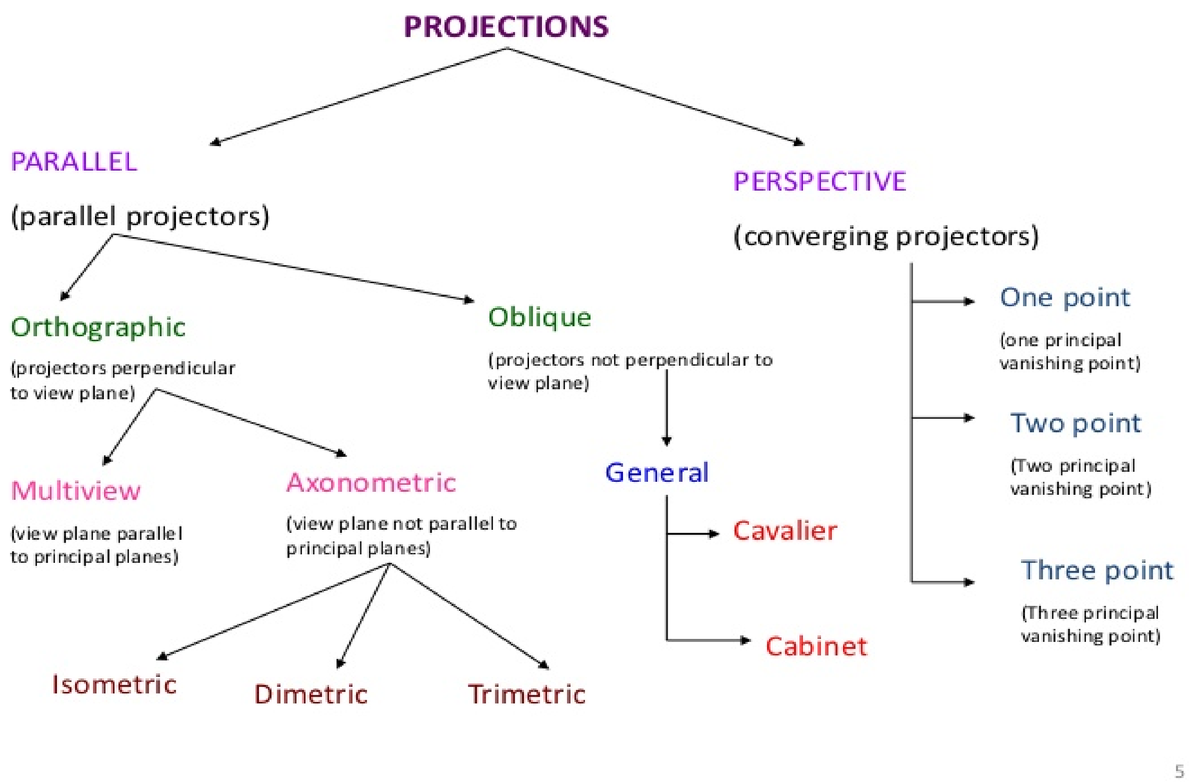
- Each basis function is positive or zero for all parameter values.
- Each basis function has precisely one maximum value, except for k=1.
- The curve line within the convex hull of its defining polygon.

B-spline curves and surfaces

1. These are the most widely used class of approximation splines.
2. The advantage over bezier is that, the degree of polynomial can be set independently of the number of control points.
3. The disadvantage is that b-splines are more complex than bezier splines.

Projections

It is the process of converting a 3D object into a 2D object. It is also defined as mapping or transformation of the object in projection plane or view plane. The view plane is displayed surface.



Perspective Projection

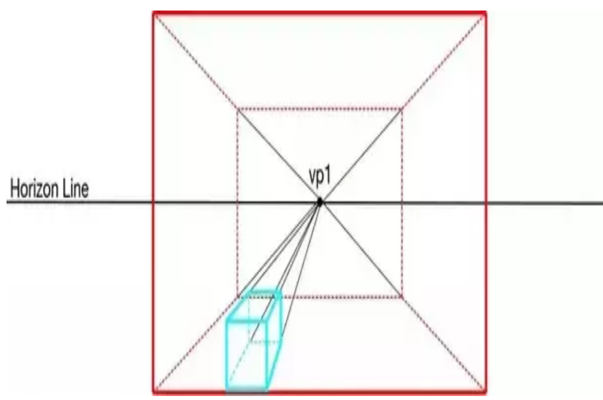
In perspective projection farther away object from the viewer, small it appears. **This property of projection gives an idea about depth**. The artist use perspective projection from drawing three-dimensional scenes.

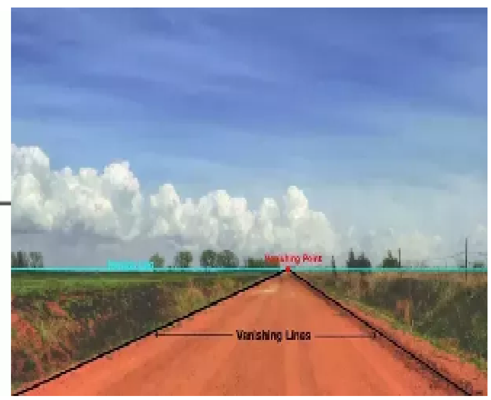
Two main characteristics of perspective are vanishing points and perspective foreshortening. Due to foreshortening object and lengths appear smaller from the center of projection. More we increase the distance from the center of projection, smaller will be the object appear.

Vanishing Point

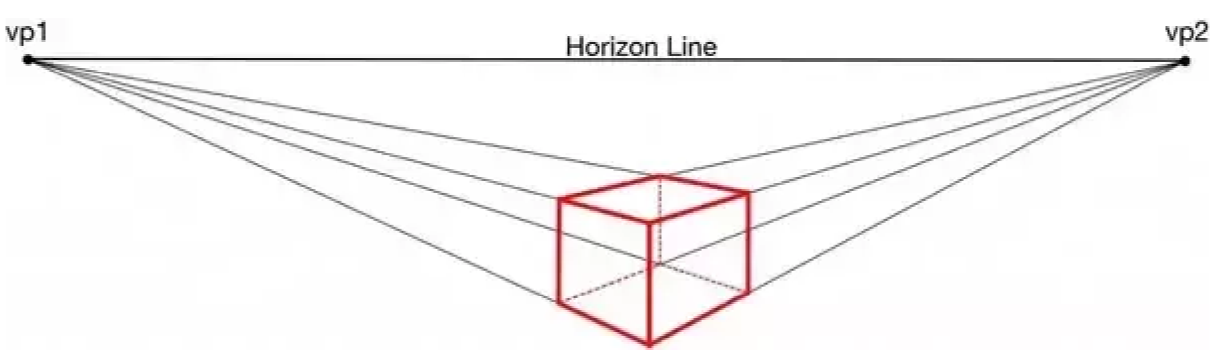
It is the point where all lines will appear to meet. There can be one point, two point, and three point perspectives.

One Point: There is only one vanishing point

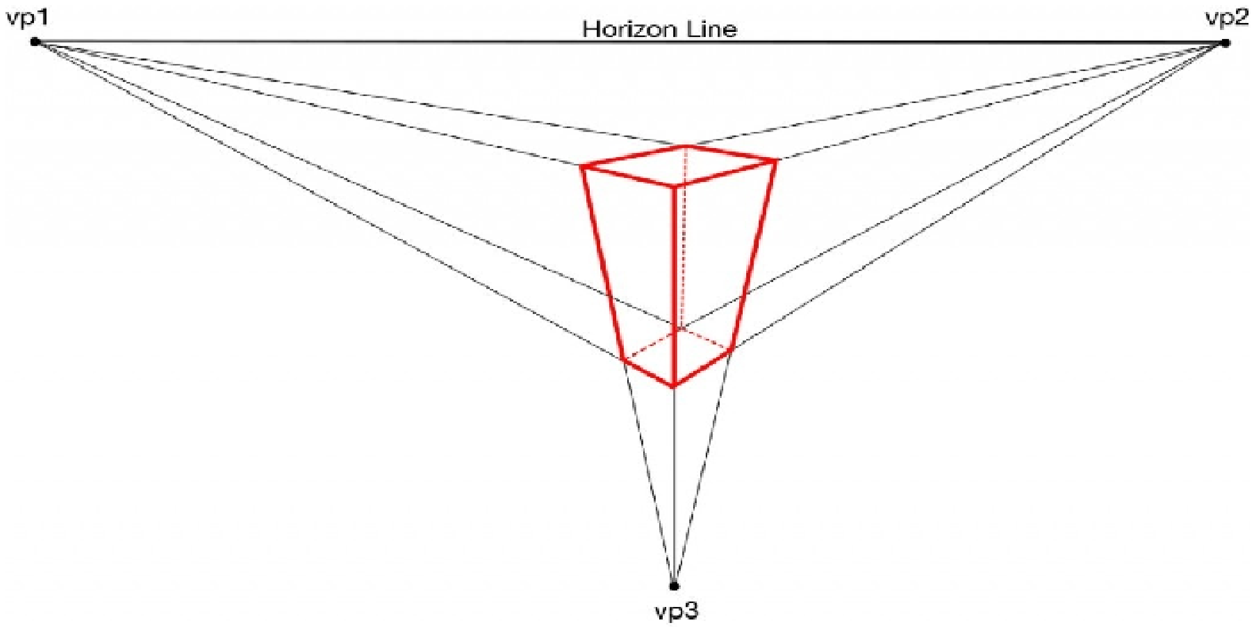




Two Points: There are two vanishing points. One is the x-direction and other in the y -direction



Three Points: There are three vanishing points. One is x second in y and third in two directions.



In Perspective projection lines of projection do not remain parallel. The lines converge at a single point called a center of projection. The projected image on the screen is obtained by points of intersection of converging lines with the plane of the screen. The image on the screen is seen as of viewer's eye were located at the Centre of projection, lines of projection would correspond to path travel by light beam originating from object.

Parallel Projection

Parallel Projection use to display picture in its true shape and size. When projectors are perpendicular to view plane then is called orthographic projection. The parallel projection is formed by extending parallel lines from each vertex on the object until they intersect the plane of the screen. The point of intersection is the projection of vertex.

In oblique projection, the direction of projection is not normal to the projection of plane. In oblique projection, we can view the object better than orthographic projection

Parallel projections are used by architects and engineers for creating working drawing of the object, for complete representations require two or more views of an object using different planes.

1. Isometric Projection: All projectors make equal angles generally

angle is of 30°.

2. Dimetric: In these two projectors have equal angles. With respect to two principle axis.
3. Trimetric: The direction of projection makes unequal angle with their principle axis.
4. Cavalier: All lines perpendicular to the projection plane are projected with no change in length. The Cavalier projection makes 45° angle with the projection plane.
5. Cabinet: All lines perpendicular to the projection plane are projected to one half of their length. These give a realistic appearance of object. The Cabinet projection makes 63.4° angle with the projection plane

References

1. *Computer Graphics: A Programming Approach Steven Harrington*
2. *chaum's Outline of Computer Graphics*
3. *Computer Graphics by Donald Hearn and M. Pauline Baker*
4. *Principles of Interactive Computer Graphics by William Newman and Robert Sproull*
5. *http://www.eenadupratibha.net/Pratibha/Engineering-Colleges/Engineering-Jobs/cse_sem_3-dim_obj_rep_un5.html*