6. Three Dimensional object Representation

Contents

Introduction

- Graphics scenes can contain many different kinds of objects and material surfaces such asTrees, flowers, clouds, rocks, water, bricks, wood paneling, rubber, paper, steel, glass, plastic and cloth
- Not possible to have a single representation for all
  - Polygon surfaces
  - Spline surfaces
  - Procedural methods
  - Physical models
  - Solid object models
  - Fractals

3D object representation

- 3D solid object representations can be generally classified into two broad categories

1. Boundary Representation (B-reps)

   It describes a three dimensional object as a set of surfaces that separate the object interior from the environment. Examples are polygon facets and spline patches.

2. Space Partitioning representation

   It describes the interior properties, by partitioning the spatial region containing an object into a set of small, non overlapping, contiguous solids (usually cubes).

   Eg: Octree Representation

6.1 Polygon Surfaces

- Polygon surfaces are boundary representations for a 3D graphics object is a set of polygons that enclose the object interior.
- Set of adjacent polygons representing the object exteriors.
- All operations linear, so fast.

- Non-polyhedron shapes can be approximated by polygon meshes.
- Smoothness is provided either by increasing the number of polygons or interpolated shading methods.

6.2 Polygon Tables

- The polygon surface is specified with a set of vertex coordinates and associated attribute parameters.
- For each polygon input, the data are placed into tables that are to be used in the subsequent processing.
- Polygon data tables can be organized into two groups: Geometric tables and attribute tables.
  - **Geometric Tables**

  Contain vertex coordinates and parameters to identify the spatial orientation of the polygon surfaces.
  - **Attribute tables**

  Contain attribute information for an object such as parameters specifying the degree of transparency of the object and its surface reflectivity and texture characteristics.

A convenient organization for storing geometric data is to create three lists:

1. The Vertex Table

Coordinate values for each vertex in the object are stored inthis table.

2. The Edge Table

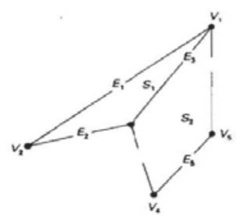
It contains pointers back into the vertex table to identify the vertices for each polygon edge.

3. The Polygon Table

It contains pointers back into the edge table to identify the edges for each polygon.

This is shown in fig

Vertex table	Edge Table	Polygon surface table
V_1 : X_1, Y_1, Z_1	E_1 : V_1, V_2	S_1 : E_1, E_2, E_3
V_2 : X_2, Y_2, Z_2	E_2 : V_2, V_3	S_2 : E_3, E_4, E_5, E_6
V_3 : X_3, Y_3, Z_3	E_3 : V_3, V_1	
V_4 : X_4, Y_4, Z_4	E_4 : V_3, V_4	
V_5 : X_5, Y_5, Z_5	E_5 : V_4, V_5	
	E_6 : V_5, V_1	



- Listing the geometric data in three tables provides a convenient to the individual component (vertices, edges and polygons) of each object.
- The object can be displayed efficiently by using data from the edge table to draw the component lines.
- Extra information can be added to the data tables for faster information extraction. For instance, edge table can be expanded to include forward points into the polygon table so that common edges between polygons can be identified more rapidly.

$E_1 : V_1, V_2, S_1$

$E_2 : V_2, V_3, S_1$

$E_3 : V_3, V_1, S_1, S_2$

$E_4 : V_3, V_4, S_2$

$E_5 : V_4, V_5, S_2$

$E_6 : V_5, V_1, S_2$

- This is useful for the rendering procedure that must vary surface shading smoothly across the edges from one polygon to the next. Similarly, the vertex table can be expanded so that vertices are cross-referenced to corresponding edges.
- Additional geometric information that is stored in the data tables includes the slope for each edge and the coordinate extends for each polygon. As vertices are input, we can calculate edge slopes and we can scan the coordinate values to identify the minimum and maximum x, y and z values for individual polygons.
- The more information included in the data tables will be easier to check for errors.
- Some of the tests that could be performed by a graphics package are:

1. That every vertex is listed as an endpoint for at least two edges.
2. That every edge is part of at least one polygon.
3. That every polygon is closed.
4. That each polygon has at least one shared edge.
5. That if the edge table contains pointers to polygons, every edge referenced by a polygon pointer has a reciprocal pointer back to the polygon.

6.3 Plane Equations

To produce a display of a 3D object, we must process the input data representation for the object through several procedures such as,

- o Transformation of the modelling and world coordinate descriptions to viewing coordinates.
  - Then to device coordinates:

  - Identification of visible surfaces
  - The application of surface-rendering procedures.

- For these processes, we need information about the spatial orientation of the individual surface components of the object.
- This information is obtained from the vertex coordinate value and the equations that describe the polygon planes.
- The equation for a plane surface is

Ax + By+ Cz + D = 0 ----(1)

Where (x, 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.
- For that, we can select three successive polygon vertices (x1, y1, z1), (x2, y2, z2) and (x3, y3, z3) and solve the following set of simultaneous linear plane equations for the ratios $A/D\ ,B/D\ \ \ and\ C/D$.

$$(A/B)\ x_k + (B/D)\ y_k + (C/D)\ z_k = -1\ ,\ Where\ k = 1{,}2{,}3 \qquad (2)$$

The solution for this set of equations can be obtained in determinant form, using Cramer's rule as

$$A = \begin{bmatrix} 1 & y1 & z1 \\ 1 & y2 & z2 \\ 1 & y3 & z3 \end{bmatrix}\ B = \begin{bmatrix} x1 & 1 & z1 \\ x2 & 1 & z2 \\ x3 & 1 & z3 \end{bmatrix}\ C = \begin{bmatrix} x1 & y1 & 1 \\ x2 & y2 & 1 \\ x3 & y3 & 1 \end{bmatrix}\ D = -\begin{bmatrix} x1 & y1 & z1 \\ x2 & y2 & z2 \\ x3 & y3 & z3 \end{bmatrix} \qquad (3)$$

Expanding the determinants, we can write the calculations for the plane coefficients in the form:

$$A = y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_{3)}$$

$$B = z_1\ (x_2 - x_3\ ) + z_2\ (x_3 - x_1\ ) + z_3(x_1 - x_3\ )$$

$$C = x_1(\ y_2 - y_3\ ) + x_2\ (y_3 - y_1\ ) + x_3\ (y_1 - y_3\ )$$

$$D = -x_1(y_2\ z_3 - y_3\ z_2) - x_2(y_3\ z_1 - y_1 z_3) - x_3(\ y_1 z_2 - y_2 z_1) \qquad (4)$$

- As vertex values and other information are entered into the polygon data structure, values for A, B, C and D are computed for each polygon and stored with the other polygon data.
- Plane equations are used also to identify the position of spatial points relative to the plane surfaces of an object. For any point (x, y, z) hot on a plane with parameters A,B,C,D, we have $Ax + By + Cz\ \neq 0$

We can identify the point as either inside or outside the plane surface according o the sigh (negative or positive) of $Ax + By + Cz\ + D$:

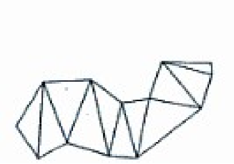
$if\ Ax + By + Cz\ + D < 0$, the point (x, y, z) is inside the surface.

$if\ Ax + By + Cz + D > 0$, the point (x, y, z) is outside the surface.

- These inequality tests are valid in a right handed Cartesian system, provided the plane parameters A,B,C and D were calculated using vertices selected in a counter clockwise order when viewing the surface in an outside-to-inside direction.

6.4 Polygon Meshes

- A single plane surface can be specified with a function such as fill Area. But when object surfaces are to be tiled, it is more convenient to specify the surface facets with a mesh function.
- One type of polygon mesh is the triangle strip A triangle strip formed with 11 triangles connecting 13 vertices.



- This function produces n-2 connected triangles given the coordinates for n vertices.
- Another similar function in the quadrilateral mesh, which generates a mesh of (n-1) by (m-1) quadrilaterals, given the coordinates for an n by m array of vertices. Figure shows 20 vertices forming a mesh of 12 quadrilaterals.

Curved Lines and Surfaces

- Displays of three dimensional curved lines and surface can be generated from an input set of mathematical functions defining the objects or from a set of user specified data points.
- When function are specified ,a package can project the defining equations for a curve to the display plane and plot pixel positions along the path of the projected function.
- For surfaces, a functional description in decorated to produce a polygon-mesh approximation to the surface.

6.5 Quadric Surfaces

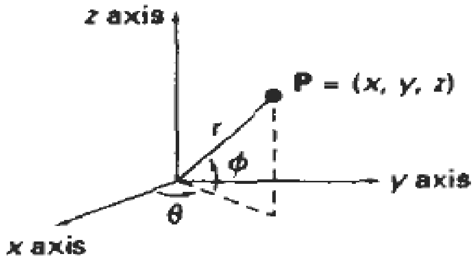
The quadric surfaces are described with second degree equations (quadratics). They include spheres, ellipsoids, tori, parabolids, and hyperboloids.

6.6 Sphere

In Cartesian coordinates, a spherical surface with radius r cantered on the coordinates origin is defined as the set of points (x, y, z) that satisfy the equation.

$$x^2 + y^2 + z^2 = r^2 \qquad (1)$$

The spherical surface can be represented in parametric form by using latitude and longitude angles, the parameter representation in eqn (2) provides a symmetric range for the angular parameter θ and φ.



x = r cosφ cosθ, -π/2 <= φ<= π/2

y = r cosφ sinθ, -π <= φ <= π (2)

z = rsinφ

Ellipsoid

- Ellipsoid surface is an extension of a spherical surface where the radius in three mutually perpendicular directions can have different values the Cartesian representation for points over the surface of an ellipsoid centered on the origin is

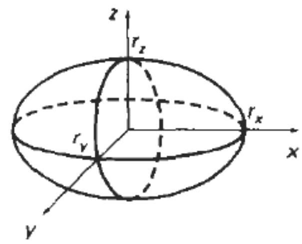
$$\frac{x^2}{r_x} + \frac{y^2}{r_y} + \frac{z^2}{r_z} = 1$$

- The parametric representation for the ellipsoid in terms of the latitude angle φ and the longitude angle θ is

$x = r_x \cos\varphi \cos\theta$, $-\pi/2 <= \varphi <= \pi/2$

$y = r_y \cos\varphi \sin\theta$, $-\pi <= \varphi <= \pi$

$z = r_z \sin\varphi$



6.7 <u>Spline Representations</u>

- A 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.
- The Spline curve refers to any sections curve formed with polynomial sections satisfying specified continuity conditions at the boundary of the pieces.
- A Spline surface can be described with two sets of orthogonal spline curves.
- Splines are used in graphics applications to design curve and surface shapes, to digitize drawings for computer storage, and to specify animation paths for the objects or the camera in the scene.
- CAD applications for splines include the design of automobiles bodies, aircraft and spacecraft surfaces, and ship hulls.

Interpolation and Approximation Splines

- Spline curve can be specified by a set of coordinate positions called control points which indicates the general shape of the curve.
- These control points are fitted with piecewise continuous parametric polynomial functions in one of the two ways.

Interpolation

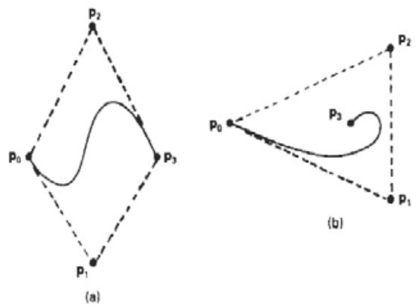
- When polynomial sections are fitted so that the curve passes through each control point the resulting curve is said to interpolate the set of control points.
- A set of six control points interpolated with piecewise continuous polynomial sections

Approximation

When the polynomials are fitted to the general control point path without necessarily passing through any control points, the resulting curve is said to approximate the set of control points.

- A set of six control points approximated with piecewise continuous polynomial sections

- Interpolation curves are used to digitize drawings or to specify animation paths.
- Approximation curves are used as design tools to structure object surfaces.
- A spline curve is designed , modified and manipulated with operations on the control points. The curve can be translated, rotated or scaled with transformation applied to the control points.
- The convex polygon boundary that encloses a set of control points is called the convex hull.
- The shape of the convex hull is to imagine a rubber band stretched around the position of the control points so that each control point is either on the perimeter of the hull or inside it.
- Convex hull shapes (dashed lines) for two sets of control points



Parametric Continuity Conditions

- For a smooth transition from one section of a piecewise parametric curve to the next various continuity conditions are needed at the connection points.

- If each section of a spline in described with a set of parametric coordinate functions or the form x = x(u), y = y(u), z = z(u), u1<= u <= u2 (a)
  we set parametric continuity by matching the parametric derivatives of adjoining curve section at their common boundary.
- Zero order parametric continuity referred to as C^0 continuity, means that the curves meet. (i.e) the values of x,y, and z evaluated at u_2 for the first curve section are equal. Respectively, to the value of x,y, and z evaluated at u_1 for the next curve section.
- First order parametric continuity referred to as C^1 continuity means that the first parametric derivatives of the coordinate functions in equation (a) for two successive curve sections are equal at their joining point.
- Second order parametric continuity, or C2 continuity means that both the first and second parametric derivatives of the two curve sections are equal at their intersection equation (a) for two successive curve sections are equal at their joining point.
- Second order parametric continuity, or C2 continuity means that both the first and second parametric derivatives of the two curve sections are equal at their intersection.
- Piecewise construction of a curve by joining two curve segments using different orders of continuity

  a)Zero order continuity only 

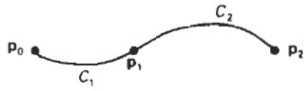
  b)First order continuity only 

  c) Second order continuity only 

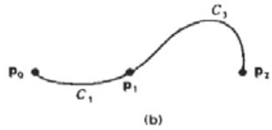
Geometric Continuity Conditions

- To specify conditions for geometric continuity is an alternate method for joining two successive curve sections.
- The parametric derivatives of the two sections should be proportional to each other at their common boundary instead of equal to each other.
- Zero order Geometric continuity referred as G^0 continuity means that the two curves sections must have the same coordinate position at the boundary point.
- First order Geometric Continuity referred as G^1 continuity means that the parametric first derivatives are proportional at the interaction of two successive sections.

- Second order Geometric continuity referred as G^2 continuity means that both the first and second parametric derivatives of the two curve sections are proportional at their boundary. Here the curvatures of two sections will match at the joining position.
  a) Three control points fitted with two curve sections joined with a parametric continuity



  b) geometric continuity where the tangent vector of curve C3 at point p1 has a greater magnitude than the tangent vector of curve C1 at p1.



Spline specifications

There are three methods to specify a spline representation:

1. We can state the set of boundary conditions that are imposed on the spline;
2. We can state the matrix that characterizes the spline; (or)
3. We can state the set of blending functions that determine how specified geometric constraints on the curve are combined to calculate positions along the curve path.

To illustrate these three equivalent specifications, suppose we have the following parametric cubic polynomial representation for the x coordinate along the path of a spline section.

$$x(u) = a_x u^3 + b_x u^2 + c_x u + d_x \qquad Where\ 0 \le u \le 1 \qquad (1)$$

- Boundary condition for this curve might be set ,for example on the end point coordinates x(0) and x(1) and on parametric derivatives $x'(0)$ and $x'(1)$.These four boundary conditions are sufficient to determine the values of four coefficients a_x, b_x, c_x and d_x.
- From the boundary conditions, we can obtain the matrix that characterizes the spline curve by first rewriting the eq-1 as the matrix product

$$x(u) = [u^3\ u^2\ u\ 1].\begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \end{bmatrix} \qquad (2)$$

$= U.C$

Where the U is the row matrix of [powers of parameter u and C is the coefficient column matrix. From the above equation the boundary conditions in the matrix form and solve for the coefficient matrix C.

$$C = M_{spline} . M_{geom} \qquad (3)$$

Where M_{geom} is a four-element column matrix containing geometric constraint values on the spline and M_{spline} is 4-by-4 matrix that performs the geometric constraint values to the polynomial coefficients and provides a characterization for spline curve. Matrix M_{geom} contains control point coordinate values and other geometric constraints that have been specified. We can substitute the matrix representation for C into equ-2 then

x (u) = U. M_{spline} . M_{geom} (4)

The matrix M_{spline} ,characterizing a spline representation, sometimes called the basis matrix ,is particularly useful for transforming from one spline representation to another.

Finally we can expand the eq-4 to obtain a polynomial representation for coordinate x in terms of geometric constraint parameters

$$x\,(u) = \sum_{0}^{3} g_k \;.BFk(u)$$

Where g_k are the constraint parameters, such as the control-point coordinates and slope of the curve at the control points. BFk(u) are the polynomial blending function.

6.8 Bézier Curves and surfaces

- This spline approximation method was developed by the French engineer Pierre Bézier for the use of Renault automobile bodies .
- Bézier splines have a number of properties that make them highly useful and convenient for curves surface design.
- They are also easy to implement.
- It is also widely available in various CAD systems, in general graphics packages and in assorted drawing and painting packages.

Bézier Curves

- Bézier Curve section can be fitted to any number of control points.
- The number of control points to be approximated and their relative position determined the degree of the Bézier polynomial
- Suppose we are given n+1 control points position: p_k = (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 Bézier polynomial function between p_0 and p_n .

$$P(u) = \sum_{k=0}^{n} p_k \;\; BEZ_{k,n}\,(u)\; where\; 0 \le u \le 1 \qquad (1)$$

The Bézier blending function $BEZ_{k,n}(u)$ are the Bernstein Polynomials:

$$BEZ_{k,n}(u) = c(n,k)u^k\;(1-u)^{n-k} \qquad (2)$$

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

$$c(n,k) = \frac{n!}{k!(n-k)!} \qquad (3)$$

Equivalently, we can define Bézier blending functions with the recursive calculation

$$BEZ_{k,n}(u) = (1-u)BEZ_{k,n-1}(u) + uBEZ_{k-1,n-1}(u), n > \; k \ge 1 \qquad (4)$$

With $BEZ_{k,k}(u)$ = u^k and $BEZ_{0,k}(u)$ = $(1-u)^k$ .Vector eq-1 represents a set of three parametric equations for the individual curve coordinates:

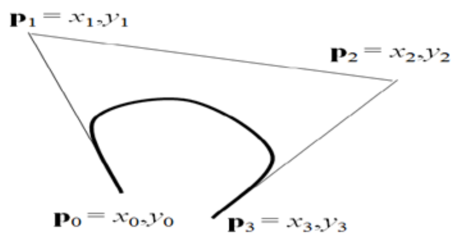
$$x(u) = \sum_{k=0}^{n} x_k \ BEZ_{k,n}\ (u)$$

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

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

As a rule Bézier curve is a polynomial of degree one less than the control points used: Three points generate a parabola, Four points generate a cubic curve and so forth. Example of four control point in the following diagram

<u>Proerties Of Bézier Curve</u>



- A very useful property of a Bézier Curves is that it always passes through the first and last control points
- That is boundary conditions at the two ends of the curve are

  P(0) = p_0

  P(1) = p_n

Value of the parametric first derivatives of a Bézier Curve at the end points can be calculated from control-point coordinates as

P'(0) = $-np_0 + np_1$

P'(1) = $-np_{n-1} + np_n$

Thus the slope at the beginning of the curve is along the line joining the first two control points , and the slope at the end of the curve is along the line joining the last two end points.

- Another important property of any Bézier curve is that it is lies within the convex hull of control points .This following form the properties of Bézier blending functions:

  They are all positive and their sum is always 1,

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

So that any curve simply the weighted the sum of the control-point positions.

<u>Cubic Bézier curve</u>

- Many graphics package provide only cubic spline functions.
- Cubic Bézier curves are generated with four control points
- Four blending functions for cubic Bézier curves obtained by substituting n=4. Then the equations are

  $BEZ_{0,3}\ (u) = (1-u)^3$

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

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

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

At the end position of the cubic Bézier curve, parametric first derivatives are

$P'(0) = 3(p_1 - 2p_0)$, $p'(1)=3(p_3 - p_2)$

And parametric second derivatives are

$P''(0) = 6(p_0 - 2p_1 + p_2)$, $P''(0) = 6(p_1 - 2p_2 + p_3)$

We can use these expressions for the parametric derivatives to construct piece wise curves with c^1 or c^2 continuity sections.

Bézier surfaces

- Two sets of orthogonal Bézier curves can be used to design an object surface by specifying by an input mesh of control points.
- The parametric vector function for the Bézier blending functions:

$$\sum_{j=0}^{m}\sum_{k=0}^{n} p_{j,k}\, BEZ_{j,m}(v) BEZ_{k,n}(u)$$

With $p_{j,k}$ specifying the location of the (m+1) by(n+1) control points.

Bézier surfaces have the same properties as Bézier curves and they provide a convenient method for interactive design applications .For each surface patch, we can select a mesh of control points in the xy "ground" plane, then we choose elevations above the ground plane for z-coordinate values of the control points. Patches then pieced together using the boundary constraint.

6.9 B-Splines (for basis splines)

- B-splines have two advantage over Bézier splines: 1.the degree of a B-spline polynomial can be set independently of the no of control points and 2.B-spline allow local control over the shape of a spline curve and surface.

B-spline curves

- We can write a general expression for the calculation of coordinate positions along a B-spline curve in a blending-function formulation as

$$P(u) = \sum_{k=0}^{n} p_k\ B_{k,d}(u), \qquad u_{min} \le u \le u_{\max} \quad 2 \le d \le n+1 - - - - - - - - (1)$$

Where p_k are an input set of n+1 control points. The B-spline blending functions

$B_{k,d}$ are polynomial of degree d-1 ,where parameter d can be chosen to be any integer value in the range from 2 up to the number of control points,n+1.Local control for B-spline is achieved by defining the blending functions over subintervals of the total range of u.

Blending functions for B-spline curves are defined by the Cox-deBoor recursion formulas

$$B_{k,1}(u) = \begin{cases} 1, & if\ u_k \le u \le u_{k+1}\ otherwise \\ 0, & \end{cases} - - - - - - - - - - - - - - (2)$$

$$B_{k,d}(u) = \frac{u - u_k}{u_{k+d-1} - u_k} B_{k,d-1}(u) + \frac{u_{k+d} - u}{u_{k+d} - u_{k+1}} B_{k+1,d-1}(u)$$

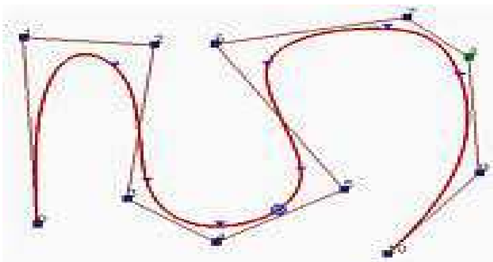
B-spline curves have the following properties

- The polynomial curve has degree $d-1$ and C^{d-2} continuity over the range of u.
- For $n+1$ control points , the curve is described with n+1 blending functions.
- Each blending function $B_{k,d}$ is defined over d subintervals of the total range of u, starting at knot value u_k.
- The range of parameter u is divided into $n+d$ subintervals by the $n+d+1$ values specified in the knot vectors.
- With knot values labeled as $\{u_0\ ,u_1, \ldots\ldots\ldots\ldots\ldots\ldots.., u_{n+d}\}$,the resulting B-spline curve is defined only the interval from knot value u_{d-1} up to knot value u_{n+1}.
- Each section of the spline curve is influenced by d control points.
- Any one control point can affect the shape of at most d curve section

  A B-spline curve lies within the convex hull at most $d+1$ control points ,so that B-spline tightly bound to the input position. Any value u in the interval from knot value u_{d-1} to u_{n+1} ,the sum over all basis function is:

$$\sum_{k=0}^{n} B_{k,d}\,(u) = 1$$

- Consists of curve segments whose polynomial coefficients only depend on just a few control points
- Local control are Segments joined at knots.
- The curve does not necessarily pass through the control points
- The shape is constrained to the convex hull made by the control points
- Uniform cubic B-splines has C^2 continuity
  - Higher than Hermite or Bezier curves.



Basis Functions

- We can create a long curve using many knots and B-splines
- The unweighted cubic B-Splines have been shown for clarity.
- These are weighted and summed to produce a curve of thedesired shape