CS354 Computer Graphics
Surface Representation I

Qixing Huang
February 26th 2018

What We Have Covered So Far

- Ray Tracing
- Shading
- Aliasing
- OpenGL

Now We Move to the Second Topic - Modeling

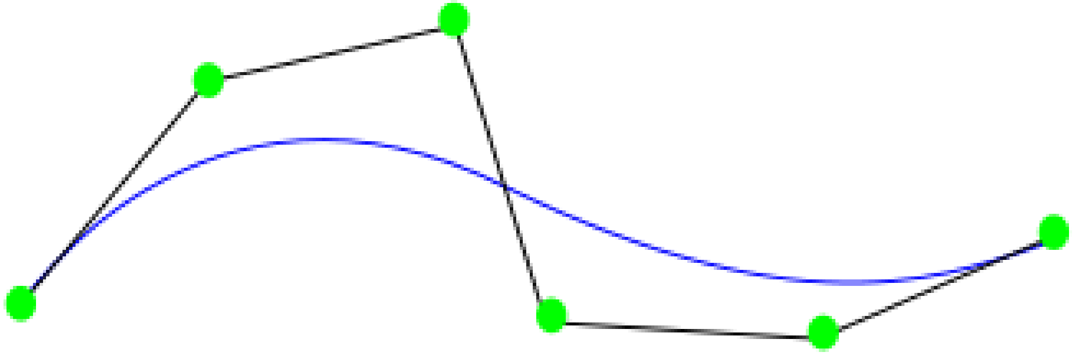
- Parametric Surfaces

- Implicit Surfaces

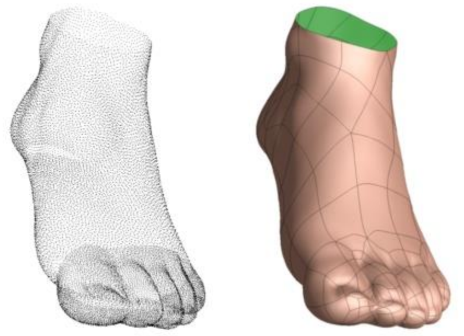
- Triangular Meshes

- Part-Based Models
  - Hierarchical Modeling

Parametric Surfaces



Bspline curve



Eck and Hoppe' 96

Implicit Surfaces

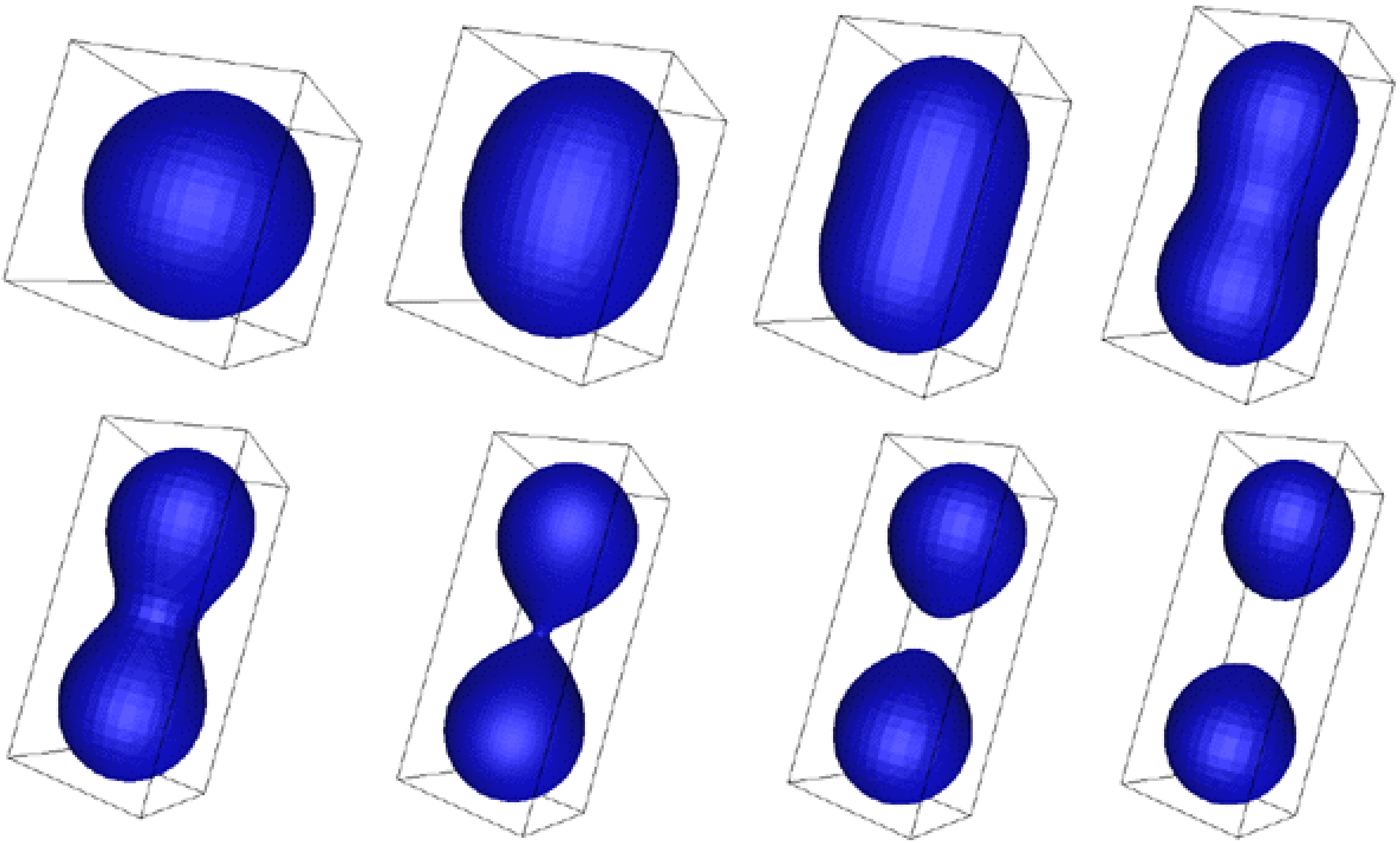
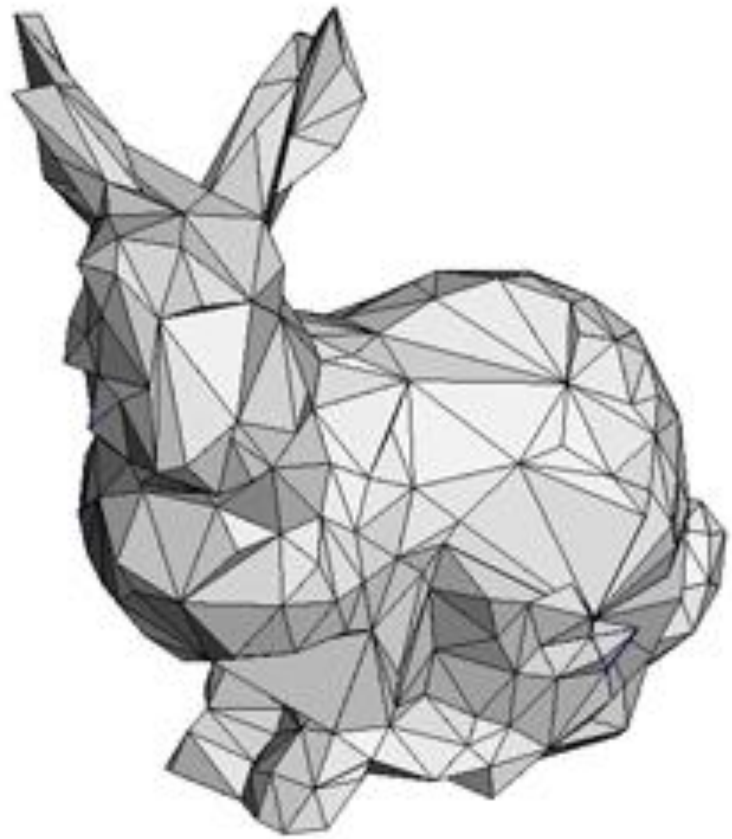


Image from http://paulbourke.net/geometry/implicitsurf/implicitsurf4.gif

Triangular Mesh



Part-based Models

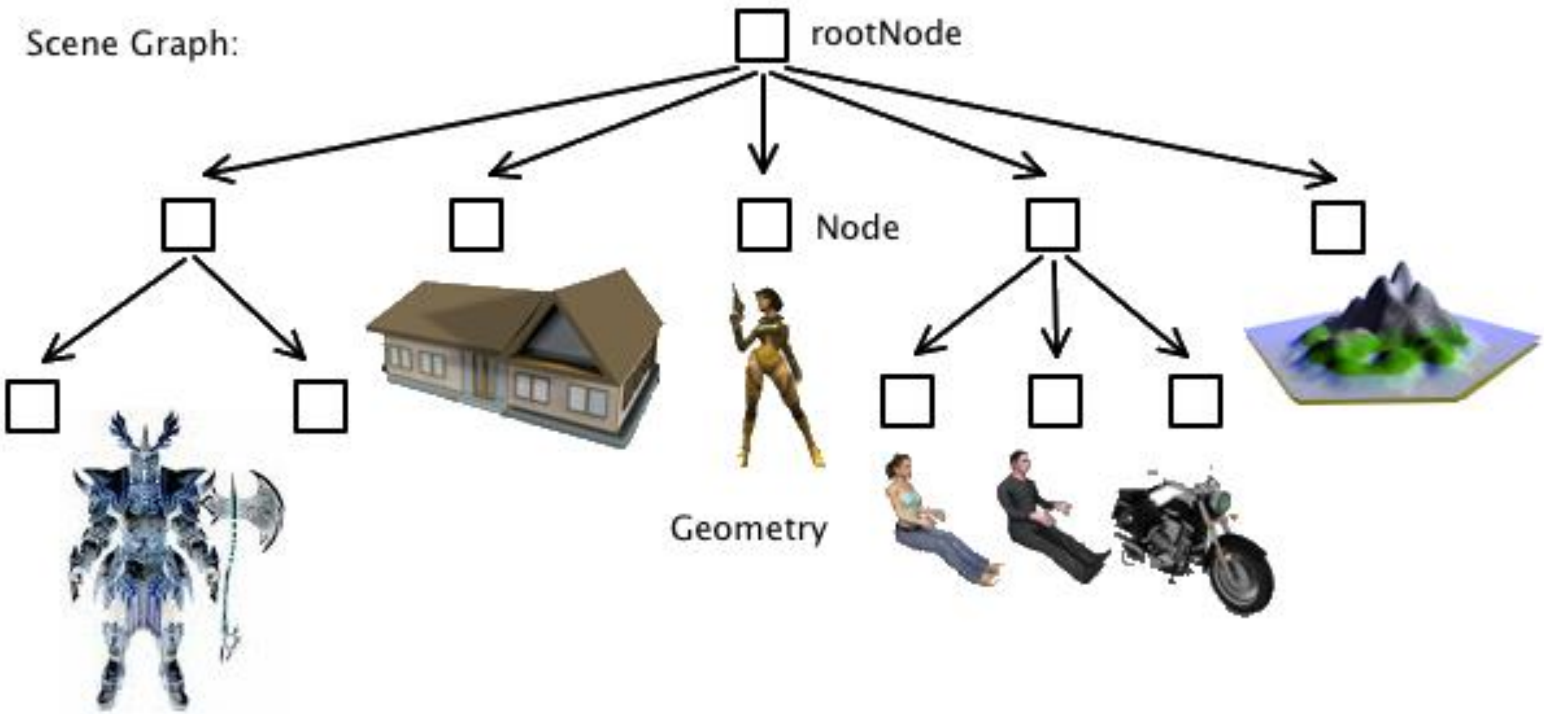


Image from https://gamedev.stackexchange.com/tags/scene-graph/info

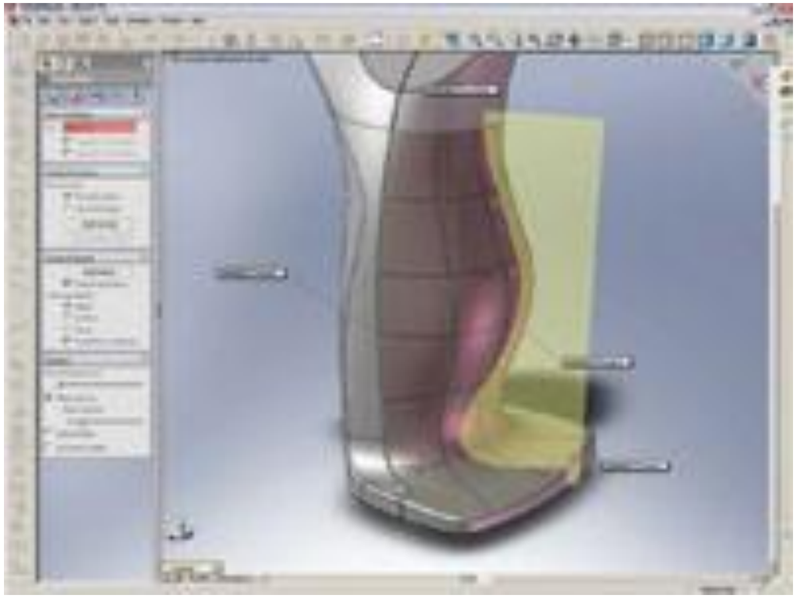
What we need to know

- Pros and Cons of each representation
- The surface “space”
- Normal computation
- Ray-surface intersection

Parametric Representation

Parametric Representation

- Widely used in Graphics/CAD/Industrial design



- What we will learn
  - Hermite
  - Bézier
  - Bspline
  - Many of their variants

Image from http://www.cadalyst.com/cad/solidworks/solidworks-2007-office-premium-cadalyst-labs-review-10923

Parametric surfaces

Hermite curves

- A cubic polynomial
- Polynomial can be specified by the position of, and gradient at, each endpoint of curve
- Determine: $x = X(t)$ in terms of x_0, x_0', x_1, x_1'

Now:

$$X(t) = a_3t^3 + a_2t^2 + a_1t + a_0$$

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

Finding Hermite coefficients

Substituting for t at each endpoint:

$x_0 = X(0) = a_0$ $\quad$ $x_0' = X'(0) = a_1$

$x_1 = X(1) = a_3 + a_2 + a_1 + a_0$ $\quad$ $x_1' = X'(1) = 3a_3 + 2a_2 + a_1$

And the solution is:

$a_0 = x_0$ $\quad$ $a_1 = x_0'$

$a_2 = -3x_0 - 2x_0' + 3x_1 - x_1'$ $\quad$ $a_3 = 2x_0 + x_0' - 2x_1 + x_1'$

The Hermite matrix: M_H

- The resultant polynomial can be expressed in matrix form:

$X(t) = t^T M_H q$ (q is the control vector)

$$X(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -2 & 1 \\ -3 & -2 & 3 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_0' \\ x_1 \\ x_1' \end{bmatrix}$$

We can now define a parametric polynomial for each coordinate required independently, ie. X(t), Y(t) and Z(t)

Hermite Basis (Blending) Functions

$$X(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -2 & 1 \\ -3 & -2 & 3 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_0' \\ x_1 \\ x_1' \end{bmatrix}$$

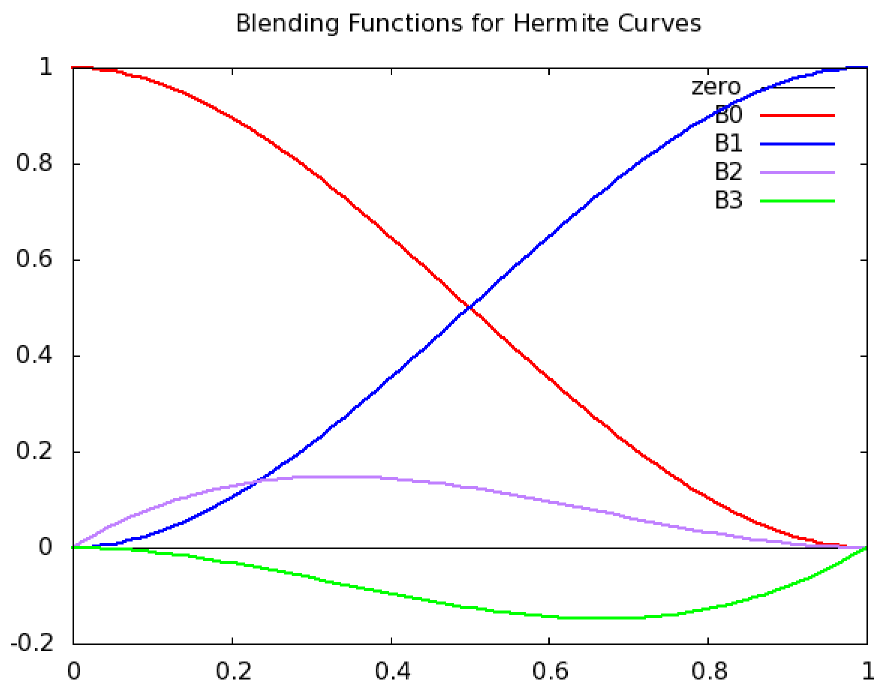
$$= \underline{\underline{(2t^3 - 3t^2 + 1)}}x_0 + \underline{\underline{(t^3 - 2t^2 + t)}}x_0' + \underline{\underline{(-2t^3 + 3t^2)}}x_1 + \underline{\underline{(t^3 - t^2)}}x_1'$$

Hermite Basis (Blending) Functions

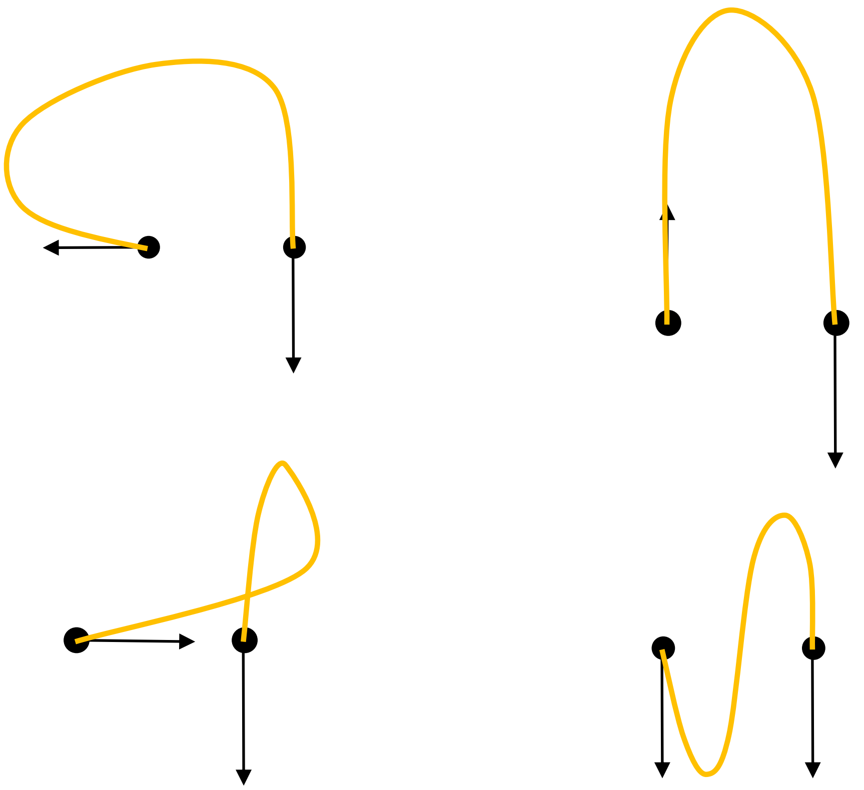
$$X(t) = \underline{\underline{(2t^3 - 3t^2 + 1)}}x_0 + \underline{\underline{(t^3 - 2t^2 + t)}}x_0' + \underline{\underline{(-2t^3 + 3t^2)}}x_1 + \underline{\underline{(t^3 - t^2)}}x_1'$$

The plot shows the shape of the so-called *blending functions.*

Note that at each end only position is non-zero, so the curve must touch the endpoints



Hermite curves can be hard to model



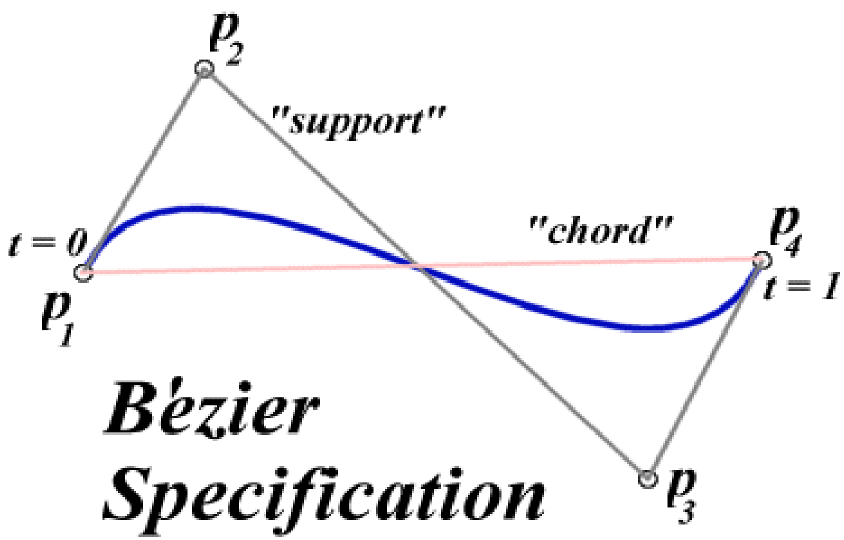
Note that the shape of the curve may not be intuitive from the boundary constraints

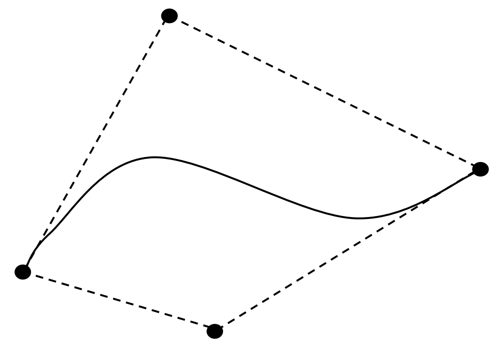
Bézier Curves

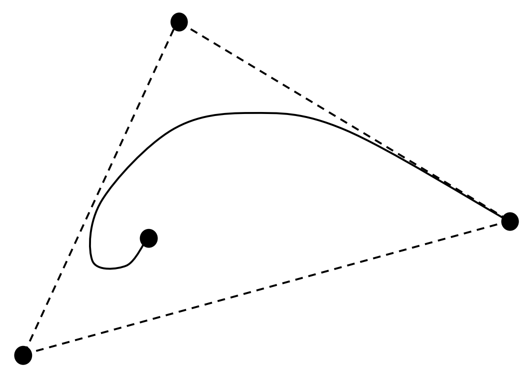
- Hermite cubic curves are mainly designed to be stitched into long curves
  - Yet the shapes are hard to control
- More intuitive to only specify points.
- Pierre Bézier (an engineer at Renault) specified 2 endpoints and 2 additional control points to specify the gradient at the endpoints.
- Strongly related to the Hermite curve

Bézier Curves

- Note the Convex Hull has been shown as a dashed line – used as a bounding extent for intersection purposes







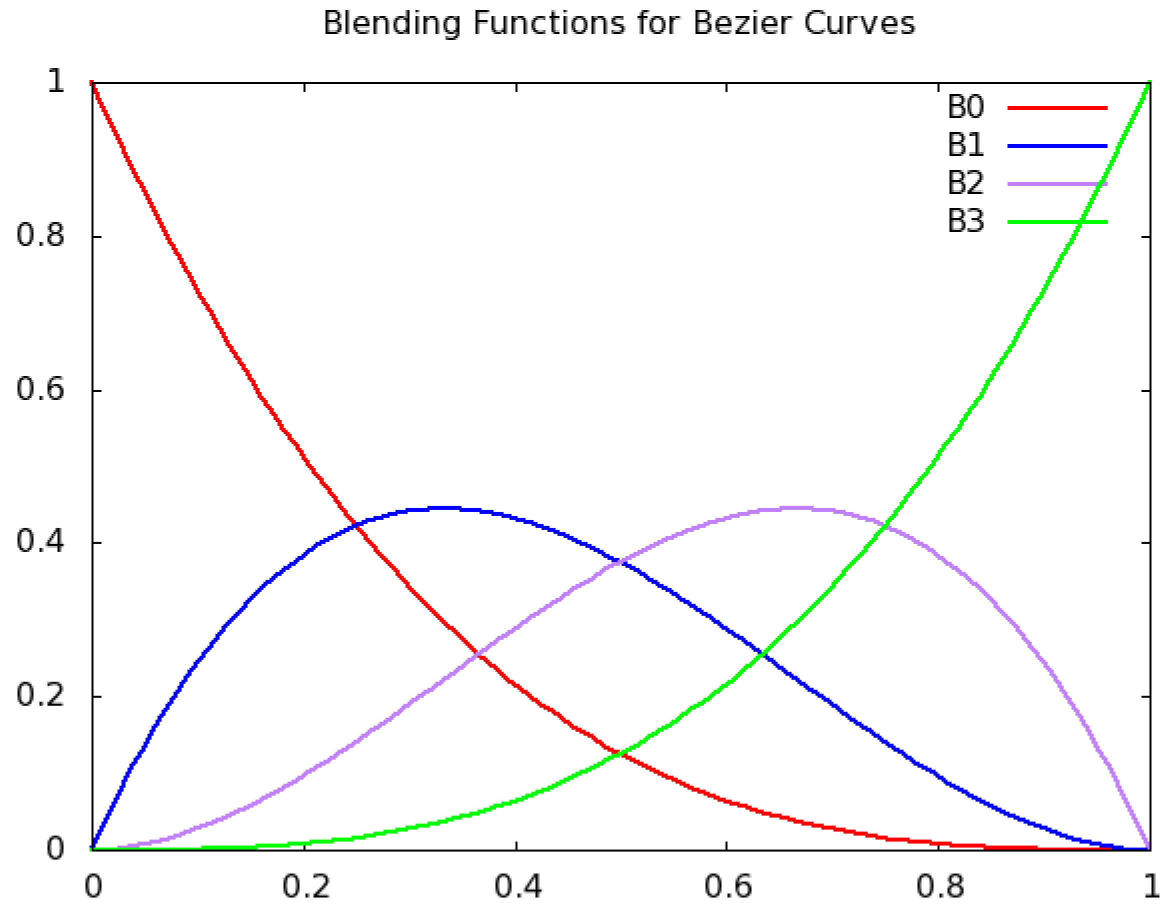
Bézier Matrix

- The cubic form is the most popular
  $X(t) = t^T M_B q$ (M_B is the Bézier matrix)
- With *n=4* and *r=0,1,2,3* *we get:*

$$X(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

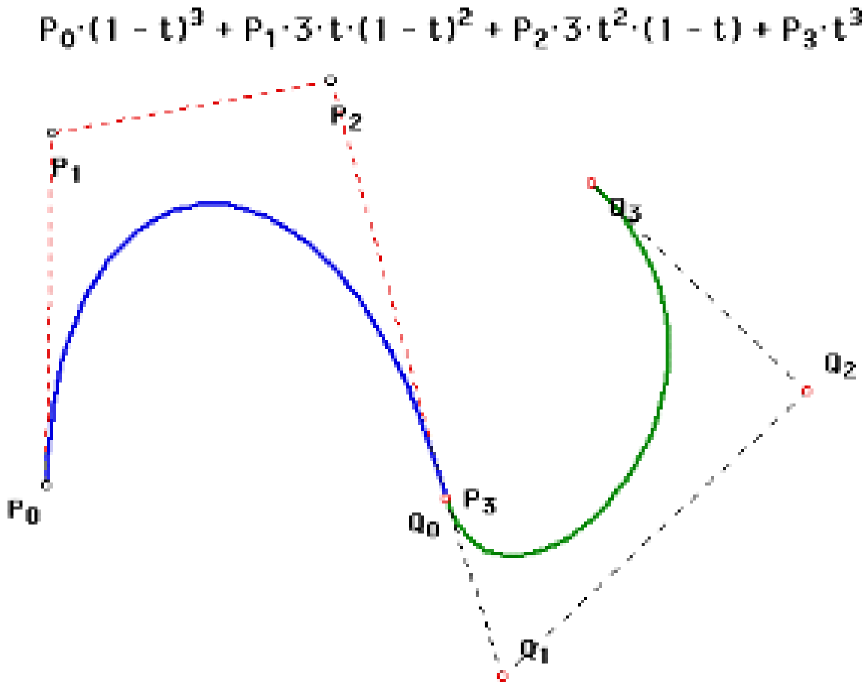
- Similarly for Y(t) and Z(t)

Bézier blending functions



Joining Bezier Curves

- *G* continuity is provided at the endpoint when $P_2 - P_3 = k\,(Q_1 - Q_0)$
- if $k=1$, *C* continuity is obtained



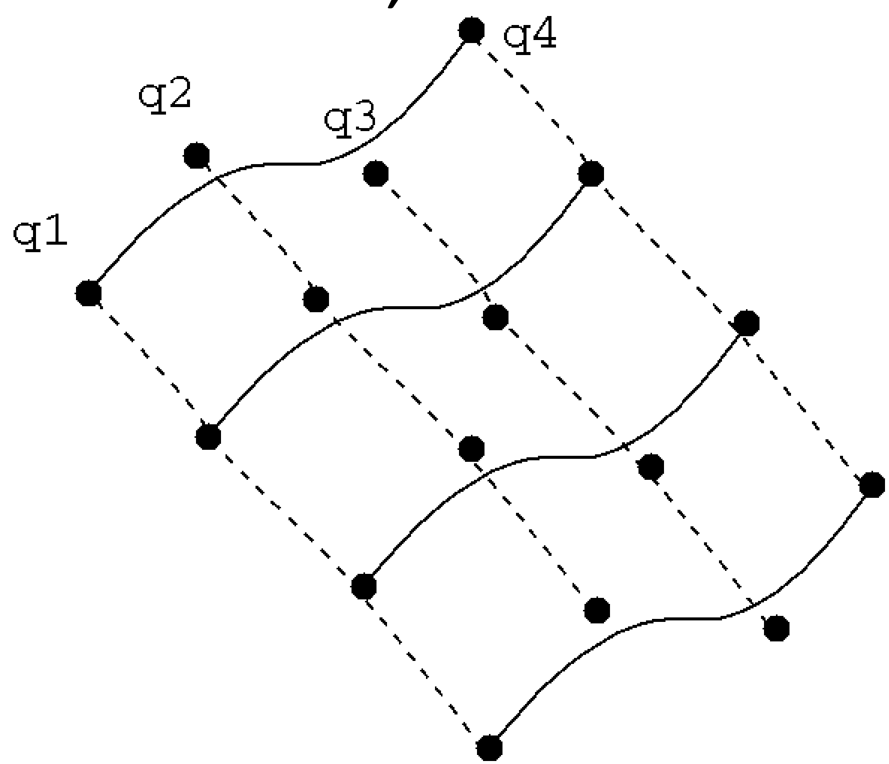
Bicubic patches

- The concept of parametric curves can be extended to surfaces

- The cubic parametric curve is in the form of $Q(t)=\boldsymbol{s}^T\boldsymbol{M}\ \boldsymbol{q}$ where $\boldsymbol{q}=(q_1,q_2,q_3,q_4)$: q_i control points, $\boldsymbol{M}$ is the basis matrix (Hermite or Bezier,…), $\boldsymbol{s}^T=(s^3, s^2, s, 1)$

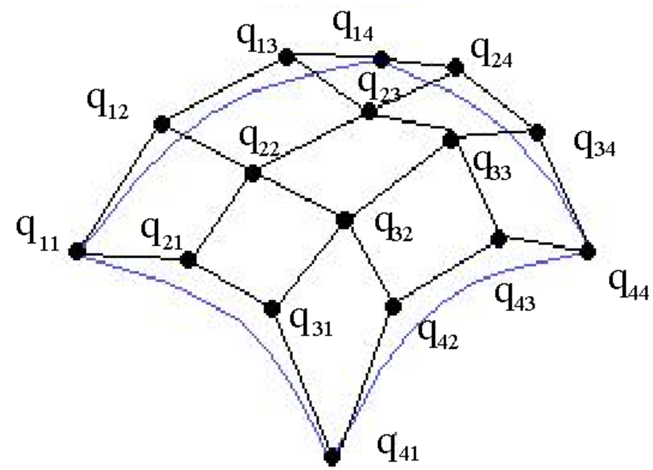
Bicubic patches

- Now we assume q_i to vary along a parameter *s,*
- $Q_i(s,t)=\mathbf{s}^T\mathbf{M}\ [q_1(t),q_2(t),q_3(t),q_4(t)]$
- $q_i(t)$ are themselves cubic curves, we can write them in the form …



Bézier example

- We compute (x,y,z) by



$$P(s,t) = \sum_{i=1}^{4}\sum_{j=1}^{4} P_{i,j} B_i(s) B_j(t)$$

$$x(s,t) = \begin{pmatrix} B_1(s) & B_2(s) & B_3(s) & B_4(s) \end{pmatrix} \begin{pmatrix} P^x_{11} & P^x_{12} & P^x_{13} & P^x_{14} \\ P^x_{21} & P^x_{22} & P^x_{23} & P^x_{24} \\ P^x_{31} & P^x_{32} & P^x_{33} & P^x_{34} \\ P^x_{41} & P^x_{42} & P^x_{43} & P^x_{44} \end{pmatrix} \begin{pmatrix} B_1(t) \\ B_2(t) \\ B_3(t) \\ B_4(t) \end{pmatrix}$$

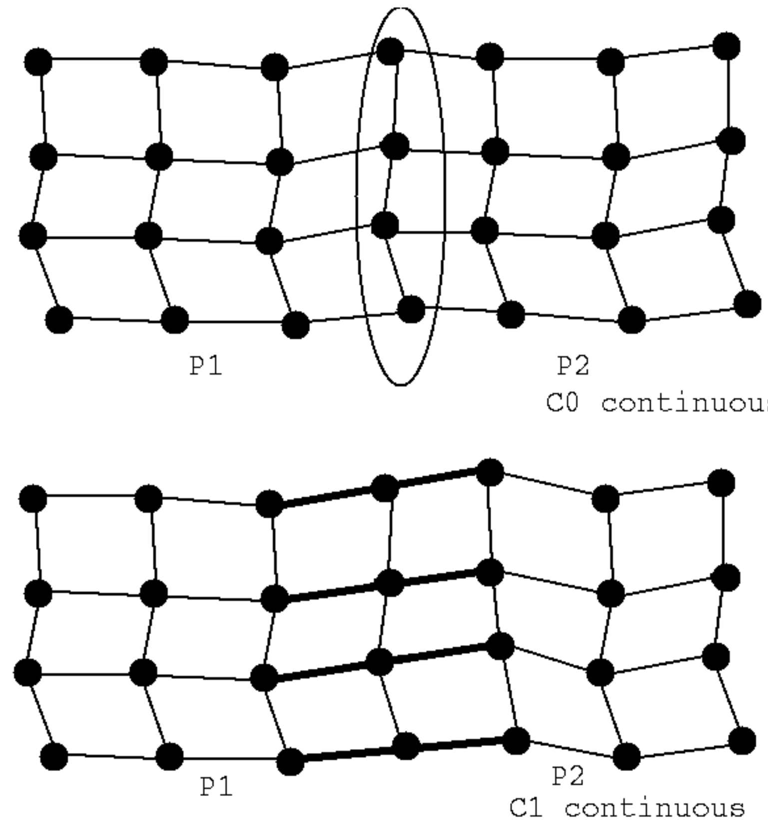
$\mathbf{s}^T M$

$M\mathbf{t}$

Replace *x* by *y* and *z*

Continuity of Bicubic Patches

- Hermite and Bézier patches
  - C^0 continuity when sharing boundary control points
  - C^1 continuity when sharing boundary control points and boundary edge vectors



Next Lecture

- Bspline
- Normal calculation
- Ray-surface intersection

Questions?