



Lecture 2: Parametric Curve and Surface

Lecturer: Dr. NGUYEN Hoang Ha



Reference: JungHyun Han. 2011. 3D Graphics for Game Programming (1st ed.), chapter 6

Line Segment



A ray defined by the start point and the direction vector is represented in a parametric equation. direction Vector

 (d_x, d_y, d_z)

start point (s_x, s_y, s_z)

• Consider a line segment between two end points, p_0 and p_1 . Vector $p_1 - p_0$ corresponds to the direction vector \rightarrow line segment can be represented as:

 $x(t) = s_x + td_x$ $y(t) = s_y + td_y$

 $z(t) = s_z + td_z$



Line Segment (cont')



- A line segment connecting two end points is represented as a linear interpolation of the points.
- The line segment may be considered as being divided into two parts by p(t), and the weight for an end point is proportional to the length of the part "on the opposite side," i.e., the weights for p₀ and p₁ are (1-t) and t, respectively.



Quadratic Bézier Curve



De Casteljau algorithm = recursive linear interpolations for defining a curve. The quadratic Bézier curve interpolates the end points, p₀ and p₂, and is pulled toward p₁, but does not interpolate it.



$$p_{0} \xrightarrow{t} p_{0}^{1-t} p_{0}^{1} = (1-t)p_{0} + tp_{1}$$

$$p_{2} \xrightarrow{t} p_{1}^{1} = (1-t)p_{1} + tp_{2} \xrightarrow{t} p_{0}^{2} = (1-t)p_{0}^{1} + tp_{1}^{1}$$

$$= (1-t)^{2}p_{0} + 2t(1-t)p_{1} + t^{2}p_{2}$$

Cubic Bézier Curve



The cubic Bézier curve interpolates the end points, p₀ and p₃, and is pulled toward p₁ and p₃.



 $p(t) = (1-t)^3 p_0 + 3t(1-t)^2 p_1 + 3t^2(1-t)p_2 + t^3 p_3$

Quartic Bézier Curve



The de Casteljau algorithm can be applied for a higher-degree Bézier curve. For example, a quartic (degree-4) Bézier curve can be constructed using five points.



Such higher-degree curves often reveal undesired wiggles. Worse still, they do not bring significant advantages. In contrast, quadratic curves have little flexibility. Therefore, cubic Bézier curves are most popularly used in the graphics field.





Different orders of the control points produce different curves.



Bézier Curve – Tessellation for Rendering

The typical method to display a Bézier curve is to approximate it using a series of line segments. This process is often called *tessellation*. It evaluates the curve at a fixed set of parameter values, and joins the evaluated points with straight lines.





Bézier Curve – Affine Invariance





Hermite Curve

$$\dot{p}(t) = \frac{d}{dt} [(1-t)^3 p_0 + 3t(1-t)^2 p_1 + 3t^2(1-t) p_2 + t^3 p_3] = -3(1-t)^2 p_0 + [3(1-t)^2 - 6t(1-t)] p_1 + [6t(1-t) - 3t^2] p_2 + 3t^2 p_3$$



$$p(t) = (1-t)^{3} p_{0} + 3t(1-t)^{2} p_{1} + 3t^{2}(1-t)p_{2} + t^{3} p_{3}$$

= $(1-t)^{3} p_{0} + 3t(1-t)^{2} (p_{0} + \frac{1}{3}v_{0}) + 3t^{2}(1-t)(p_{3} - \frac{1}{3}v_{3}) + t^{3} p_{3}$
= $(1-3t^{2}+2t^{3})p_{0} + t(1-t)^{2} v_{0} + (3t^{2}-2t^{3})p_{3} - t^{2}(1-t)v_{3}$

Catmull-Rom Spline



A spline (piecewise curve) composed of cubic Hermite curves passes through the given points q_is. Two adjacent Hermite curves should share the tangent vector at their junction.





Application



Application (cont')







15

Bilinear Patch

- Combination of linear interpolations using four control points
- p_{00} p_{00} $p_0^u = (1-u)p_{00} + up_{01}$ p_{01} 001 p(u,v)v $=(1-v)p_{a}^{u}+$ p_{11} $p_1^u = (1-u)p_{10} + up_{11}$ p_{10} u p_{10} 2×2 control points Bilinear interpolation (u first, and then v)
- Matrix representation where the center matrix corresponds to the control point net

$$p(u,v) = (1-v \quad v) \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix} \begin{pmatrix} 1-u \\ u \end{pmatrix}$$

= $(1-v \quad v) \begin{pmatrix} (1-u)p_{00} + up_{01} \\ (1-u)p_{10} + up_{11} \end{pmatrix}$
= $(1-u)(1-v)p_{00} + u(1-v)p_{01} + (1-u)vp_{10} + uvp_{11}$
weights for the control points

A collection of line segments





Bilinear Patch (cont')



Tessellation with nested for loops





 $p(u,v) = (1-u)(1-v)p_{00} + u(1-v)p_{01} + (1-u)vp_{10} + uvp_{11}$

```
foreach u in the range [0,1]
  foreach v in the range [0,1]
    Evaluate the patch using (u,v) to obtain (x,y,z)
  endforeach
endforeach
```

Bilinear Patch (cont')



Let's reverse the order of u and v in linear interpolations.





Another collection of line segments

$$p(u,v) = (1 - v \quad v) \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix} \begin{pmatrix} 1 - u \\ u \end{pmatrix}$$
$$= ((1 - v)p_{00} + vp_{10} (1 - v)p_{01} + vp_{11}) \begin{pmatrix} 1 - u \\ u \end{pmatrix}$$
$$= (1 - u)(1 - v)p_{00} + u(1 - v)p_{01} + (1 - u)vp_{10} + uvp_{11}$$

Bilinear Patch (cont')



The control points are not necessarily in a plane, and consequently the bilinear patch is not necessarily a plane.







Biquadratic Bézier Patch





as a collection of Bézier curves



Biquadratic Bézier Patch (cont')

$$\begin{split} p(u,v) &= (1-u)^2(1-v)^2 p_{00} + 2u(1-u)(1-v)^2 p_{01} + u^2(1-v)^2 p_{02} + \\ &\quad 2(1-u)^2 v(1-v) p_{10} + 4uv(1-u)(1-v) p_{11} + 2u^2 v(1-v) p_{12} + \\ &\quad (1-u)^2 v^2 p_{20} + 2u(1-u)v^2 p_{21} + u^2 v^2 p_{22} \end{split}$$





(f) Tessellation



(g) Three quadratic Bézier curves (in v)



(h) A biquadratic Bézier patch as a collection of Bézier curves (each defined in u)



Biquadratic Bézier Patch (cont')









(d)

Biquadratic Bézier Patch (cont')



- So far, we have seen *two-stage explicit evaluation*.
- Now, let's see repeated bilinear interpolations, which produce the same result.



 $\begin{aligned} p_{00}^1 &= (1-u)(1-v)p_{00} + u(1-v)p_{01} + (1-u)vp_{10} + uvp_{11} \\ p_{01}^1 &= (1-u)(1-v)p_{01} + u(1-v)p_{02} + (1-u)vp_{11} + uvp_{12} \\ p_{10}^1 &= (1-u)(1-v)p_{10} + u(1-v)p_{11} + (1-u)vp_{20} + uvp_{21} \\ p_{11}^1 &= (1-u)(1-v)p_{11} + u(1-v)p_{12} + (1-u)vp_{21} + uvp_{22} \end{aligned}$

$$p_{00}^2 = (1-u)(1-v)p_{00}^1 + u(1-v)p_{01}^1 + (1-u)vp_{10}^1 + uvp_{11}^1$$

Bicubic Bézier Patch



A simple extension of biquadratic Bézier patch

$$p(u,v) = \left((1-v)^3 \ 3v(1-v)^2 \ 3v^2(1-v) \ v^3 \right)$$

$$\begin{pmatrix} p_{00} \ p_{01} \ p_{02} \ p_{03} \\ p_{10} \ p_{11} \ p_{12} \ p_{13} \\ p_{20} \ p_{21} \ p_{22} \ p_{23} \\ p_{30} \ p_{31} \ p_{32} \ p_{33} \end{pmatrix} \begin{pmatrix} (1-u)^3 \\ 3u(1-u)^2 \\ 3u^2(1-u) \\ u^3 \end{pmatrix}$$



We can of course reverse the order of *u* and *v*.

Bicubic Bézier Patch (cont')



Let's apply repeated bilinear interpolations.



Tessellation and rendering result





Shape Control





Bézier Triangle



p(u, v, w) = ua + vb + wc

$$u = \frac{area(p, b, c)}{area(a, b, c)}, v = \frac{area(p, c, a)}{area(a, b, c)}, w = \frac{area(p, a, b)}{area(a, b, c)}$$

- The weights (u,v,w) are called the barycentric coordinates of p.
- The triangle is divided into three sub-triangles, and the weight given for a control point is proportional to the area of the sub-triangle "on the opposite side."
- Obviously, u+v+w=1, and therefore w can be replaced by (1-u-v).



Bézier Triangle (cont')



See what s represents when u=0 and u=1.

Bézier Triangle (cont')



Degree-3 Bézier triangle

