Document Summary

Generated on: 2025-07-07 18:50:39

Section 1:

Wojciech Matusik 6.837 Computer Graphics Bézier Curves and Splines . Curves in 2D are useful in their own right – Provides basis for surface editing . Polylines – Sequence of vertices connected by straight line segments – Useful, but not for smooth curves . Smooth curves – How do we specify them? – A little harder (but not too much)

Section 2:

This curve lies on the 2D plane, but is itself 1D . You can just as well define 1D curves in 3D space . The second definition can describe trajectories, the speed at which we move on the curve . Animation: trajectories • In general: interpolation and approximation .

Section 3:

The curve is completely determined by the control points . We will interpolate the curve by a smooth curve . The curve will be determined by a control point . Good for user interface/modeling is good for storage and good for data storage . The result is a smooth parametric curve P(t)

Section 4:

In practice: low-order polynomials, chained together – Convenient for animation, where t is time . Convenient to tessellation because we can discretize t and approximate the curve with a polyline . It is easy to rasterize mathematical line segments into pixels – OpenGL and the graphics hardware can do it for you .

Section 5:

Interpolation – Goes through all specified points – Sounds more logical – But can be more unstable . Approximation – Does not go through all points – Turns out to be convenient • We will do something in between . Cubic polynomial 23 Cubic Bézier Curve that is, • $P(t) = (1-t)^3 P1 + 3t(1)^2 P2 + tt^2(1-T)^2 P4$.

Section 6:

A Bézier curve is bounded by the convex hull of its control points . P(t) is a weighted combination of the 4 control points with weights . P1 is the most influential point, then P2, P3, and P4 . P2 and P3 never have full influence .

Section 7:

$P(t) = (1-t)^3$ P1 + $3t(1-1)^2$ P2 + $(3t^2)$ P3 + t^3 P4 is a linear combination of basis polynomials . In 3D, each vector has three components x, y, z; geometrically each vector is actually the sum . i, j, k, k are basis vectors; i, k is basis vectors .

Section 8:

Polynomials are closed under addition & scalar multiplication . Cubic polynomials also compose a vector space . The x and y coordinates of cubic Bézier curves belong to this subspace as functions of t. The result is still a cubic polynomial (verify!)

Section 9:

i, j, i+j is not a basis (missing k direction!) i j k In 3D . Any cubic polynomial is a linear combination of these: a0+a1t+a2t2+a3t3 = a0*1+a 1*t +a1*t, $1+t^2$, t^3+t^3, t^2+t $t¼t^2$. For example: 1 t t2 t3 = 1, t3 + t2 + t3: 1, 1 + t + t, 1+ t +

Section 10:

Change-of-basis matrix “Canonical” monomial basis These relationships hold for each value of t . For Bézier curves, the basis polynomials/vectors are Bernstein polynomsials . Sum to 1 for every t – called partition of unity .

Section 11:

P(t) is a linear combination of the control points with weights equal to Bernstein polynomials at t . The control points (P1, P2, P3, P4) are the “coordinates” of the curve in the Bernstein basis . In this sense, specifying a Bézier curve with control points is like specifying a 2D point with its x and y coordinates .

Section 12:

How do we go from Bernstein basis to the canonical monomial basis 1, t, t^2, t^3 and back? – With a matrix! The coefficients are the entries in the matrix B! Cubic polynomials form a 4D vector space .

Section 13:

Matrix- Vector Notation matrix of control points (2 x 4) Bernstein polynomials (4x1 vector) point on curve (2x 1 vector) 57 Flashback 58 Cubic Bézier in Matrix Notation . “Geometry matrix” of control point P1..P4 “Spline Matrix” (Bernstein) Canonical monomial basis .

Section 14:

Can we split a Bezier curve in the middle into two Bézier curves? This is useful for adding detail – It avoids using nasty higher-order curves . The resulting curves are again a cubic (Why? A cubic in t is also a cubic in 2t) – Hence it must be representable using the Bernstein basis .

Section 15:

The two new curves are defined by – P1, P’1, . P‘1 P■'1 P'2 P'3 P'4 P'1 – Together they exactly replicate the original curve! – Originally 4 control points, now 7 (more control points) 66 Result of Split in Middle: P■.'1(t) = t, not just 0.5, but actually works to construct a point at any t .

Section 16:

Inkscape is an open source vector drawing program for Mac/Windows . Bezier curves: piecewise polynomials, Bernstein polynomsials . Subdivision by de Casteljau algorithm . All linear, matrix algebra: linear combination of basis functions .