Johns Hopkins
Engineering for Professionals

605.767 Applied Computer Graphics

Brian Russin

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Module 6E
NURBS and Other Splines

Non-uniform B-splines

- Non-uniform B-splines can have multiple internal knot values and unequal spacing between knot values
- Increased flexibility in controlling curve shape
  - Different intervals produce different shapes for blending functions in the intervals
  - Knot **multiplicity** (number of identical values) can produce variations in the curve shape and introduce discontinuities
    - Curve continuity is reduced by 1 for each repeat of a knot value
  - Watt pg. 88-89 has many illustrations of effects
- Generally use the Cox-deBoor algorithm to recursively generate the blending functions

Rational Splines

- **Rational splines** are ratios between two spline functions
  - Just as rational functions are ratios between two polynomials
- Non-uniform rational B-splines (NURBS) extend control of a curve or surface beyond just moving control points
  - An extra parameter called a **weight** is added to control points
  - Rational B-spline curve is defined by a set of homogeneous control points:
    - $P_i^w = (w_i x_i, w_i y_i, w_i z_i, w_i)$

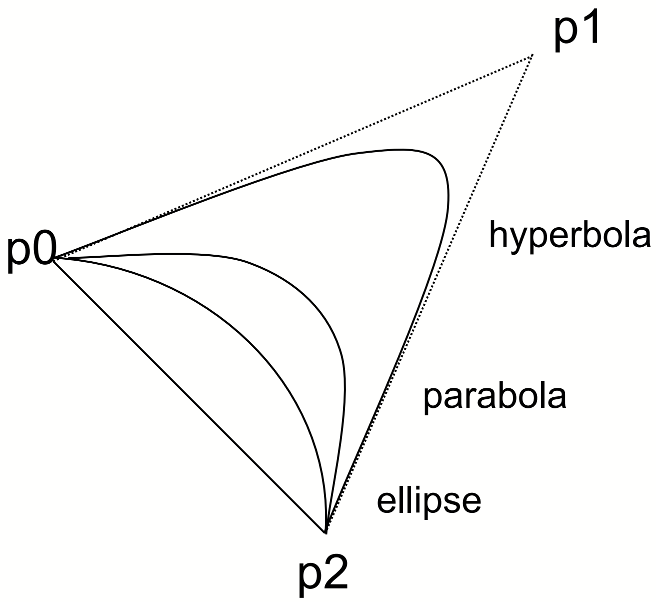
$$p(t) = \frac{\sum_{i=0}^{n} \omega_i B_{i,k}(t) p_i}{\sum_{i=0}^{n} \omega_i B_{i,k}(t)}$$

NURBS

- The w_i parameters are called weights
  - Extra shape parameters
- Weights affect the curve locally
  - Can be thought of as a coupling factor
    - Increase w_i will pull curve towards control point p_i
    - Decrease w_i will move curve away from control point p_i
- Rational splines have 2 major advantages over nonrational splines
  - Provide an exact representation for quadric curves (circles, ellipses)
    - Graphics package could model all curve shapes with rational spline representations
  - Invariant with respect to perspective transformation
    - Can be evaluated in screen space
  - Graphics design packages typically use nonuniform rational B-splines (NURBS) for constructing rational B-splines
    - Nonuniform knot representations

Plotting Conic Sections with NURBS

- Use quadratic spline functions
  - d = 3, 3 control points
- Use the knot vector (0,0,0,1,1,1)
  - Same as quadratic Bezier curve
- Set weighting functions:
  - $w_0 = w_2 = 1$
  - $w_1 = r / (1-r)$, $0 <= r < 1$
  - Yields the following conics:
    - Hyperbola section: $r > 1/2$, $w_1 > 1$
    - Parabola section: $r = 1/2$, $w_1 = 1$
    - Ellipse section: $r < 1/2$, $w_1 < 1$
    - Straight line segment: $r = 0$, $w_1 = 0$



Other Spline Representations

- Natural cubic splines
  - Mathematical representation of original draftsman spline
- Cardinal splines
  - Sequence of individual curves joined to form a larger curve
  - An array of points and a tension parameter
- Kochanek-Bartels splines
  - Cubic Hermite spline with tension, bias and continuity parameters
    - Changes behavior at tangents
- Catmull-Rom
  - Curve will pass through all control points
    - Kochanek-Bartels spline with tension, bias, and continuity set to 0
- β-*spline*
  - Introduce bias and tension to control the whole curve
    - Bias controls the skewness of the curve
    - Tension controls how tightly or loosely the spline fits the control polygon

Conversion Between Spline Representations

- Often convenient to convert one spline representation into another
  - Model and design using B-splines – local control
    - Local control
  - Render using Bezier splines - efficient rendering
- Hearn and Baker derive the transformations
  - Product of the inverse basis matrix of one with the basis matrix of the other
  - $M_{s1,s2} = M^{-1}_{spline2}\, M_{spline1}$
    - Transform control points from spline 1 to the control points for spline 2
- Non-uniform B-splines cannot be characterized with a single basis matrix
  - Can rearrange the knot sequence to change the non-uniform B-spline to a Bezier representation
    - Non-uniform B-spline over 4 control points with knot sequence (0,0,0,0,1,1,1,1) is a cubic Bezier curve

Cubic B-spline to cubic Bezier:

$$M_{B\text{-}spline \to Bezier} = \frac{1}{6}\begin{bmatrix} 1 & 4 & 1 & 0 \\ 0 & 4 & 2 & 0 \\ 0 & 2 & 4 & 0 \\ 0 & 1 & 4 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} p_{1,Bezier} \\ p_{2,Bezier} \\ p_{3,Bezier} \\ p_{4,Bezier} \end{bmatrix} = \frac{1}{6}\begin{bmatrix} 1 & 4 & 1 & 0 \\ 0 & 4 & 2 & 0 \\ 0 & 2 & 4 & 0 \\ 0 & 1 & 4 & 1 \end{bmatrix}\begin{bmatrix} p_{1,B\text{-}spline} \\ p_{2,B\text{-}spline} \\ p_{3,B\text{-}spline} \\ p_{4,B\text{-}spline} \end{bmatrix}$$