Curves

Adam Finkelstein
Princeton University
COS 426, Fall 2001

Introduction

• Splines: mathematical way to express curves
• Motivated by “loftsman’s spline”
  o Long, narrow strip of wood/plastic
  o Used to fit curves through specified data points
  o Shaped by lead weights called “ducks”
  o Gives curves that are “smooth” or “fair”
• Have been used to design:
  o Automobiles
  o Ship hulls
  o Aircraft fuselages and wings

Many applications in graphics

• Fonts
• Animation paths
• Shape modeling
• etc…

Goals

• Some attributes we might like to have:
  o Predictable control
  o Multiple values
  o Local control
  o Versatility
  o Continuity
• We’ll satisfy these goals using:
  o Piecewise
  o Parametric
  o Polynomials

Parametric curves

A parametric curve in the plane is expressed as:
\[
\begin{align*}
  x &= x(u) \\
  y &= y(u)
\end{align*}
\]

Example: a circle with radius \( r \) centered at origin:
\[
\begin{align*}
  x &= r \cos u \\
  y &= r \sin u
\end{align*}
\]

In contrast, an implicit representation is:
\[
x^2 + y^2 = r^2
\]

Parametric polynomial curves

A parametric polynomial curve is described:
\[
\begin{align*}
  x(u) &= \sum_{i=0}^{n} a_i u^i \\
  y(u) &= \sum_{i=0}^{n} b_i u^i
\end{align*}
\]

• Advantages of polynomial curves
  o Easy to compute
  o Infinitely differentiable
**Piecewise parametric polynomials**

- Use different polynomial functions on different parts of the curve
  - Provides flexibility
  - How do you guarantee smoothness at "joints"? (continuity)
- In the rest of this lecture, we’ll look at:
  - Bézier curves: general class of polynomial curves
  - Splines: ways of putting these curves together

**Bézier curves**

- Developed simultaneously in 1960 by
  - Bézier (at Renault)
  - deCasteljau (at Citroen)
- Curve \( Q(u) \) is defined by nested interpolation:

\[
V_j \text{'s are control points } \{V_0, V_1, ..., V_n\} \text{ is control polygon}
\]

**Basic properties of Bézier curves**

- Endpoint interpolation:
  \[
  Q(0) = V_0 \\
  Q(1) = V_n
  \]
- Convex hull:
  - Curve is contained within convex hull of control polygon
- Symmetry
  \[
  Q(u) \text{ defined by } \{V_0, ..., V_n\} = Q(1-u) \text{ defined by } \{V_n, ..., V_0\}
  \]

**Explicit formulation**

- Let’s indicate level of nesting with superscript \( j \):
- An explicit formulation of \( Q(u) \) is given by:
  \[
  V_j' = (1-u)V_{j+1}' + uV_{j+1}'
  \]
- Case \( n=2 \):
  \[
  Q(u) = V_0' = (1-u)V_1' + uV_1'
  \]
  \[
  = (1-u)[(1-u)V_2' + uV_2'] + u[(1-u)V_2' + uV_2']
  \]
  \[
  = (1-u)^2V_0'' + 2u(1-u)V_1'' + u^2V_2''
  \]

**More properties**

- General case: Bernstein polynomials
  \[
  Q(u) = \sum_{i=0}^{n} \binom{n}{i} u^i (1-u)^{n-i}
  \]
- Degree: is a polynomial of degree \( n \)
- Tangents:
  \[
  Q'(0) = n(V_1' - V_0') \\
  Q'(1) = n(V_n' - V_{n-1}')
  \]

**Cubic curves**

- From now on, let’s talk about cubic curves (\( n=3 \))
- In CAGD, higher-order curves are often used
- In graphics, piecewise cubic curves will do it
  - Specify points and tangents
  - Will describe curve in space
- All these ideas generalize to higher-order curves
Matrix form

- Bézier curves can also be described in matrix form:

\[
Q(u) = \sum_{i=0}^{3} \binom{n}{i} u^i (1-u)^{n-i}
\]

\[
= (1-u)^3 V_0 + 3u(1-u)^2 V_1 + 3u^2(1-u)V_2 + u^3 V_3
\]

\[
= \begin{bmatrix} u^3 & u^2 & u \end{bmatrix} M_{Bézier}
\]

Display

- Here is pseudocode for displaying Bézier curves:

```plaintext
procedure Display({Vi}): 
  if {Vi} flat within ε then
    output line segment V0Vn
  else
    subdivide to produce {Li} and {Ri}
    Display({Li})
    Display({Ri})
  end if
end procedure
```

Flatness

- Q: How do you test for flatness?
- A: Compare the length of the control polygon to the length of the segment between endpoints

\[
\frac{||V_0 - V_n||}{||V_n - V_0||} < 1 + \varepsilon
\]

Splines

- For more complex curves, piece together Béziers
- We want continuity across joints:
  - Positional (C0) continuity
  - Derivative (C1) continuity
- Q: How would you satisfy continuity constraints?
- Q: Why not just use higher-order Bézier curves?
- A: Splines have several of advantages:
  - Numerically more stable
  - Easier to compute
  - Fewer bumps and wiggles

Catmull-Rom splines

- Properties
  - Interpolate control points
  - Have C0 and C1 continuity
- Derivation
  - Start with joints to interpolate
  - Build cubic Bézier between each joint
  - Endpoints of Bézier curves are obvious
- What should we do for the other Bézier control points?
Catmull-Rom Splines

- Catmull & Rom use:
  - half the magnitude of the vector between adjacent CP's
- Many other formulations work, for example:
  - Use an arbitrary constant \( \tau \) times this vector
  - Gives a "tension" control
  - Could be adjusted for each joint

Properties

- Catmull-Rom splines have these attributes:
  - C1 continuity
  - Interpolation
  - Locality of control
  - No convex hull property
    (Proof left as an exercise.)

Matrix formulation

- Express conversion from Catmull-Rom CP's to Bezier CP's with a matrix:

\[
\begin{bmatrix}
    B_0 & B_1 & B_2 & B_3 \\
\end{bmatrix}
\begin{bmatrix}
    0 & 6 & 0 & 0 & 0 \\
    1 & -1 & 6 & 1 & 0 \\
    0 & 6 & 0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
    V_0 \\
    V_1 \\
    V_2 \\
    V_3 \\
\end{bmatrix}
\]

- Exercise: Derive this matrix.

B-splines

- We still want local control
- Now we want C2 continuity
- Give up interpolation
- It turns out we get convex hull property
- Constraints:
  - Three continuity conditions at each joint \( j \):
    - Position of two curves same
    - Derivative of two curves same
    - Second derivatives same
  - Local control
    - Each joint affected by small set of (4) CP

Matrix formulation for B-splines

- Grind through some messy math to get:

\[
Q(u) = [u^3 \ u^2 \ u \ 1] \begin{bmatrix}
    -1 & 3 & -3 & 1 \\
    3 & -6 & 3 & 0 \\
    -3 & 0 & 3 & 0 \\
    1 & 4 & 1 & 0 \\
\end{bmatrix}
\begin{bmatrix}
    V_0 \\
    V_1 \\
    V_2 \\
    V_3 \\
\end{bmatrix}
\]

What's next?

- Use curves to create parameterized surfaces
- Surface of revolution
- Swept surfaces
- Surface patches

Przemysław Prusinkiewicz
Demetri Terzopoulos