Constructing Curve Segments

Linear blend:

- Line segment from an affine combination of points

\[ P^1_0(t) = (1 - t)P_0 + tP_1 \]
Quadratic blend:

- Quadratic segment from an affine combination of line segments

\[
P_0(t) = (1 - t)P_0 + tP_1 \\
P_1(t) = (1 - t)P_1 + tP_2 \\
P_0^2(t) = (1 - t)P_0^1(t) + tP_1^1(t)
\]
Cubic blend:

- Cubic segment from an affine combination of quadratic segments

\[
\begin{align*}
P_0^1(t) & = (1 - t)P_0 + tP_1 \\
P_1^1(t) & = (1 - t)P_1 + tP_2 \\
P_0^2(t) & = (1 - t)P_0^1(t) + tP_1^1(t) \\
P_1^1(t) & = (1 - t)P_1 + tP_2 \\
P_2^1(t) & = (1 - t)P_2 + tP_3 \\
P_1^2(t) & = (1 - t)P_1^1(t) + tP_2^1(t) \\
P_0^3(t) & = (1 - t)P_0^2(t) + tP_1^2(t)
\end{align*}
\]
- The pattern should be evident for higher degrees
Geometric view (de Casteljau Algorithm):

- Join the points $P_i$ by line segments
- Join the $t : (1 - t)$ points of those line segments by line segments
- Repeat as necessary
- The $t : (1 - t)$ point on the final line segment is a point on the curve
- The final line segment is tangent to the curve at $t$
Expanding Terms (Basis Polynomials):

- The original points appear as coefficients of Bernstein polynomials

\[
P_0^0(t) = P_0 1
\]
\[
P_0^1(t) = (1 - t)P_0 + tP_1
\]
\[
P_0^2(t) = (1 - t)^2 P_0 + 2(1 - t)tP_1 + t^2 P_2
\]
\[
P_0^3(t) = (1 - t)^3 P_0 + 3(1 - t)^2 tP_1 + 3(1 - t)t^2 P_2 + t^3 P_3
\]
\[
P_0^n(t) = \sum_{i=0}^{n} P_i B_i^n(t)
\]

where
\[
B_i^n(t) = \frac{n!}{(n-i)!i!} (1 - t)^{n-i} t^i = \left( \begin{array}{c} n \\ i \end{array} \right) (1 - t)^{n-i} t^i
\]

- The Bernstein polynomials of degree \( n \) form a basis for the space of all degree-\( n \) polynomials
Recursive evaluation schemes:

- To obtain curve points:
  - Start with given points and form successive, pairwise, affine combinations
    \[
    P_i^0 = P_i \\
    P_i^j = (1 - t)P_i^{j-1} + tP_{i+1}^{j-1}
    \]
    - The generated points \(P_i^j\) are the deCasteljau points
- To obtain basis polynomials:
  - Start with 1 and form successive, pairwise, affine combinations
    \[
    B_0^0 = 1 \\
    B_i^j = (1 - t)B_i^{j-1} + tB_{i+1}^{j-1}
    \]
    where \(B_i^s = 0\) when \(r < 0\) or \(r > s\)
Recursive triangle diagrams (upward):

Computing deCasteljau points

- Each node gets the affine combination of the two nodes entering from below
  - Leaf nodes have the value of their respective points
    \[ P_1^2 = (1 - t)P_1^1 + tP_2^1 \]

- Each node gets the sum of the path products entering from below

\[
\begin{align*}
P_1^2 &= P_0^1(1 - t)(1 - t) + P_0^2 t(1 - t) + P_0^2(1 - t)t + P_0^3 tt \\
\Rightarrow P_1^2 &= (1 - t)^2 P_0^1 + 2(1 - t)t P_0^2 + t^2 P_3^0
\end{align*}
\]
Recursive triangle diagrams (downward):

Computing Bernstein (basis) polynomials

- Each node gets the affine combination of the two nodes entering from above
  - Root node has value 1
  - For other nodes, missing entries above count as zero
- Each node gets the sum of the path products entering from above

\[
B_1^3 = t(1-t)(1-t) + (1-t)t(1-t) + (1-t)t(1-t)t
\]

\[
\Rightarrow P_1^3 = 3(1-t)^2t
\]
Recurrence, Subdivision:

\[ B_i^n(t) = (1 - t)B_i^{n-1} + tB_{i-1}^{n-1}(t) \]

\[ \implies \text{deCasteljau's algorithm:} \]

\[ P(t) = P_o^n(t) \]

\[ P_i^k(t) = (1 - t)P_i^{k-1}(t) + tP_{i+1}^{k-1} \]

\[ P_i^0 = P_i \]

Use to evaluate point at \( t \), or subdivide into two new curves:

- \( P_0^0, P_0^1, \ldots P_0^n \) are the control points for the left half
- \( P_n^0, P_{n-1}^1, \ldots P_0^n \) are the control points for the right half
Discontinuities in Splines

\textit{Bézier Discontinuities}:

- Two Bézier segments can be completely disjoint
- Two segments join if they share last/first control point
Common Parameterization and Blending Functions

- Joined curves can be given common parameterization
  - Parameterize first segment with $0 \leq t < 1$
  - Parameterize next segment with $1 \leq t \leq 2$, etc.

- Look at blending/basis polynomials under this parameterization
  - Combine those for common $P_j$ into a single piecewise polynomial
Combined Curve Segments

- Curve is $P(t) = P_0B_0(t) + P_1B_1(t) + P_2B_2(t) + P_3B_3(t) + P_4B_4(t)$, where

\[
B_0(t) = \begin{cases} 
(1 - t)^2 & 0 \leq t < 1 \\
0 & 1 \leq t \leq 2 
\end{cases}
\]

\[
B_1(t) = \begin{cases} 
2(1 - t)t & 0 \leq t < 1 \\
0 & 1 \leq t \leq 2 
\end{cases}
\]

\[
B_2(t) = \begin{cases} 
t^2 & 0 \leq t < 1 \\
(2 - t)^2 & 1 \leq t \leq 2 
\end{cases}
\]

\[
B_3(t) = \begin{cases} 
0 & 0 \leq t < 1 \\
2(2 - t)(t - 1) & 1 \leq t \leq 2 
\end{cases}
\]

\[
B_4(t) = \begin{cases} 
0 & 0 \leq t < 1 \\
(t - 1)^2 & 1 \leq t \leq 2 
\end{cases}
\]
Curve Discontinuities from Basis Discontinuities

- $P_2$ is scaled by $B_2(t)$, which has a discontinuous derivative
- The corner in the curve results from this discontinuity
Spline Continuity

Smother Blending Functions:

- Can $B_0(t), \ldots, B_4(t)$ be replaced by smoother functions?
  - Piecewise polynomials on $0 \leq t \leq 2$
  - Continuous derivatives
- Yes, but we lose one degree of freedom
  - Curve has no corner if segments share a common tangent
  - Tangent is given by the chords $P_1P_2, P_2P_3$
  - An equation constrains $P_1, P_2, P_3$
    \[ P_3 - P_2 = P_2 - P_1 \implies P_2 = \frac{P_1 + P_3}{2} \]
- This equation leads to combinations:

\[
P_0B_0(t) + P_1 \left( B_1(t) + \frac{1}{2}B_2(t) \right) + P_3 \left( \frac{1}{2}B_2(t) + B_3(t) \right) + P_4B_4(t)
\]
Spline Basis:

• Combined functions form a smoother spline basis

\[ \overline{B}_0(t) = B_0(t) \]
\[ \overline{B}_1(t) = \left( B_1(t) + \frac{1}{2}B_2(t) \right) \]
\[ \overline{B}_2(t) = \left( \frac{1}{2}B_2(t) + B_3(t) \right) \]
\[ \overline{B}_3(t) = B_4(t) \]
Smother Curves:

• Control points used with this basis produce smoother curves.
B-Splines

General B-Splines:

- Nonuniform B-splines (NUBS) generalize this construction
- A B-spline, $B^d_i(t)$, is a piecewise polynomial:
  - each of its segments is of degree $\leq d$
  - it is defined for all $t$
  - its segmentation is given by knots $t = t_0 \leq t_1 \leq \cdots \leq t_N$
  - it is zero for $T < T_i$ and $T > T_{i+d+1}$
  - it may have a discontinuity in its $d - k + 1$ derivative at $t_j \in \{t_i, \ldots, t_{i+d+1}\}$, if $t_j$ has multiplicity $k$
  - it is nonnegative for $t_i < t < t_{i+d+1}$
  - $B^d_i(t) + \cdots + B^d_{i+d}(t) = 1$ for $t_{i+d} \leq t < t_{i+d+1}$, and all other $B^d_j(t)$ are zero on this interval
- Bézier blending functions are the special case where all knots have multiplicity $d + 1$
Example (Quadratic):
Evaluation:

- There is an efficient, recursive evaluation scheme for any curve point
- It generalizes the triangle scheme (deCasteljau) for Bézier curves
- Example (for cubics and $t_{i+3} \leq t < t_{i+4}$):