Introduction to Parametric interpolation for computer animation

François Faure

EVASION-LJK
Main idea

- Series of pairs (time, parameter values)
- Interpolate inbetween

Key T₀,q₀  
T, interpolated q  
Key T₁,q₁
General form: \( P(u) = (x(u), y(u), z(u)) \quad 0 \leq u \leq 1 \)
Linear curves

- General form: \( P(u) = (1 - u)P_0 + uP_1 \)
- Use fractional value of global parameter \( U \) for piecewise curves
- Sharp discontinuities
Cubic curves

- Patches are defined using endpoints and tangents (Hermite splines)
- $x(u)$ and $y(u)$ are cubic functions
- Smooth continuity
Hermite spline equation

\[ y = b_0 + b_1 u + b_2 u^2 + b_3 u^3 \]
\[ y' = b_1 + 2b_2 u + 3b_3 u^2 \]

- Match values and slopes

\[ b_0 = y_0 \]
\[ b_1 = y'0 \]
\[ b_2 = 3(y_1 - y_0) - 2y'_0 - y'_1 \]
\[ b_3 = 2(y_0 - y_1) + y'_0 + y'_1 \]
Hermite blending functions

\[ y(t) = (1-3u^2+2u^3)y_0+(3u^2-2u^3)y_1+(u-2u^2+u^3)y_0'+(u^3-u^2)y_1' \]
Beziers curves

- 4 points
- tangents along the first and last line segments
Catmull-Rom splines

- Derived from Hermite splines
- Approximate tangents using control points
  \[ D_i = \frac{1}{2}(P_{i+1} - P_{i-1}) \]
- Arbitrary first and last points
Add intuitive control parameters to Catmull-Rom splines

- Tension $t$
- Bias $b$
- Continuity $c$

\[
D^- = \frac{(1 - t)(1 - b)(1 + c)}{2}(P_{i+1} - P_i)
\]
\[
+ \frac{(1 - t)(1 + b)(1 - c)}{2}(P_i - P_{i-1})
\]
\[
D^+ = \frac{(1 - t)(1 - b)(1 - c)}{2}(P_{i+1} - P_i)
\]
\[
+ \frac{(1 - t)(1 + b)(1 + c)}{2}(P_i - P_{i-1})
\]
Tension $t$ is responsible for sharpness

$$D = \frac{1 - t}{2}(P_{i+1} - P_{i-1})$$
Bias $b$ modifies the slope

$$D = \frac{1 - b}{2}(P_{i+1} - P_i) + \frac{1 + b}{2}(P_i - P_{i-1})$$
Discontinuity $c$ splits the tangent in two pieces

\[
D^- = \frac{1 + c}{2} (P_{i+1} - P_i) + \frac{1 - c}{2} (P_i - P_{i-1})
\]

\[
D^+ = \frac{1 - c}{2} (P_{i+1} - P_i) + \frac{1 + c}{2} (P_i - P_{i-1})
\]
It is difficult to control $P$ and $\frac{dP}{du}$ independently.

Example using two Kochaneck-Bartels with same shape.
Arclength parameterization

- We want to control the arclength $s(t)$

- Problem: $s(t)$ is far from trivial
Approximate arclength parameterization

- Approximate $s(t)$ using distances between regularly sampled points
- Model $s(t)$ as a tabulated function
- For a given $s$, find the corresponding interval and apply linear approximation
We want to control the velocity of a moving object along a given path (spline)

- Use arclength parameterization
- Apply velocity control as $s(t)$ with $s = 0$ at starting point and $s = 1$ at end point