TIEA311
Tietokonegrafiikan perusteet
kevät 2018

("Principles of Computer Graphics" – Spring 2018)

TIEA311 Tietokonegrafiikan perusteet – kevät 2018
(“Principles of Computer Graphics” – Spring 2018)

Frontpage of the local course version, held during Spring 2018 at the Faculty of Information technology, University of Jyväskylä:
`http://users.jyu.fi/~nieminen/tgp18/`

Recap

- Cubic polynomials form a 4D vector space.
- Bernstein basis is canonical for Bézier.
  - Can be seen as influence function of data points
  - Or data points are coordinates of the curve in the Bernstein basis
- We can change between basis with matrices.

General Spline Formulation

$$Q(t) = \mathbf{GBT(t)} = \text{Geometry } \mathbf{G} \cdot \text{Spline Basis } \mathbf{B} \cdot \text{Power Basis } \mathbf{T(t)}$$

- Geometry: control points coordinates assembled into a matrix (P1, P2, …, Pn+1)
- Spline matrix: defines the type of spline
  - Bernstein for Bézier
- Power basis: the monomials (1, t, ..., *t*n)
- Advantage of general formulation
  - Compact expression
  - Easy to convert between types of splines
  - Dimensionality (plane or space) does not really matter

A Cubic Only Gets You So Far

- What if you want more control?

Higher-Order Bézier Curves

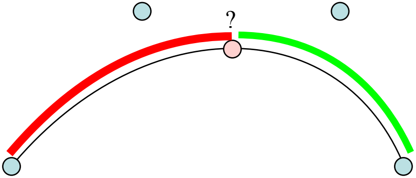
- > 4 control points
- Bernstein Polynomials as the basis functions
  - For polynomial of order *n*, the *ith* basis function is

$$B_i^n(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}$$

- Every control point affects the entire curve
  - Not simply a local effect
  - More difficult to control for modeling
- You will not need this in this class

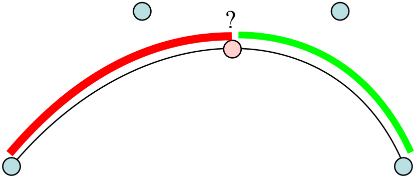
Subdivision of a Bezier Curve

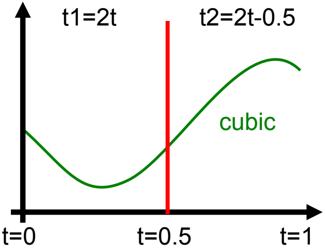
- Can we split a Bezier curve in the middle into two Bézier curves?
  - This is useful for adding detail
  - It avoids using nasty higher-order curves



Subdivision of a Bezier Curve

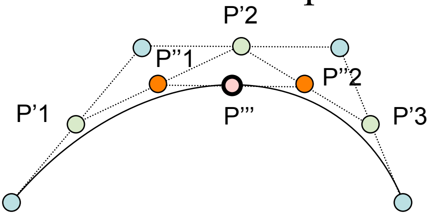
- Can we split a Bezier curve in the middle into two Bézier curves?
  - The resulting curves are again a cubic (Why? A cubic in *t* is also a cubic in 2*t*)
  - Hence it must be representable using the Bernstein basis. So yes, we can!





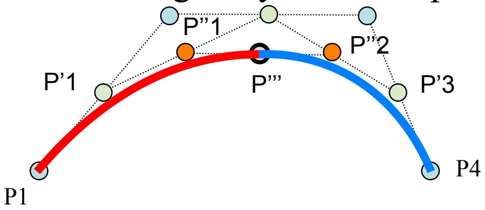
De Casteljau Construction

- Take the middle point of each of the 3 segments
- Construct the two segments joining them
- Take the middle of those two new segments
- Join them
- Take the middle point P’’’



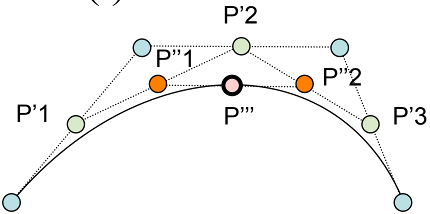
Result of Split in Middle

- The two new curves are defined by
  - P1, P’1, P”1, and P’”
  - P’”, P”2, P’3, and P4
- Together they exactly replicate the original curve!
  - Originally 4 control points, now 7 (more control)



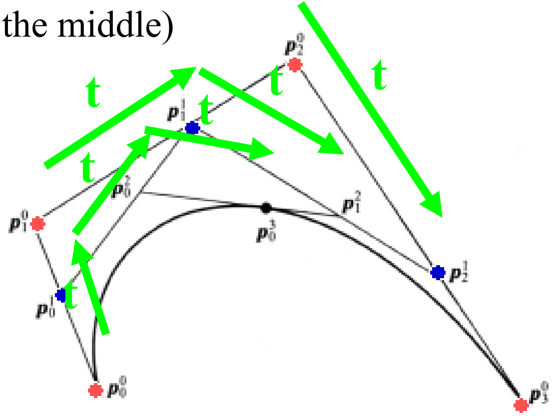
Sanity Check

- Do we actually get the middle point?
- $B1(t)=(1-t)^3$
- $B2(t)=3t(1-t)^2$
- $B3(t)=3t^2(1-t)$
- $B4(t)=t^3$



De Casteljau Construction

- Actually works to construct a point at any *t,* not just 0.5
- Just subdivide the segments with ratio (1-*t*), *t* (not in the middle)



Recap

- Bezier curves: piecewise polynomials
- Bernstein polynomials
- Linear combination of basis functions
  - Basis: control points   weights: polynomials
  - Basis: polynomials   weights: control points
- Subdivision by de Casteljau algorithm
- All linear, matrix algebra

6.837 Computer Graphics

Curve Properties & Conversion, Surface Representations

General Spline Formulation

$$Q(t) = \mathbf{GBT(t)} \ = \text{Geometry } \mathbf{G} \cdot \text{Spline Basis } \mathbf{B} \cdot \text{Power Basis } \mathbf{T(t)}$$

- Geometry: control points coordinates assembled into a matrix (P1, P2, …, Pn+1)
- Power basis: the monomials 1, *t*, *t*2, ...
- Cubic Bézier:

$$\mathrm{P(t)} = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} =$$

$$\begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \end{pmatrix} \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$

Linear Transformations & Cubics

- What if we want to transform each point on the curve with a linear transformation **M**?

$$P'(t) = \mathbf{M}\left[\begin{pmatrix} P_{1,x} & P_{2,x} & P_{3,x} & P_{4,x} \\ P_{1,y} & P_{2,y} & P_{3,y} & P_{4,y} \end{pmatrix}\begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}\right]$$

Linear Transformations & Cubics

- What if we want to transform each point on the curve with a linear transformation **M**?
  - Because everything is linear, it is the same as transforming only the control points

$$P'(t) = \mathbf{M}\left[\begin{pmatrix} P_{1,x} & P_{2,x} & P_{3,x} & P_{4,x} \\ P_{1,y} & P_{2,y} & P_{3,y} & P_{4,y} \end{pmatrix}\begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}\right]$$

$$= \left[\mathbf{M}\begin{pmatrix} P_{1,x} & P_{2,x} & P_{3,x} & P_{4,x} \\ P_{1,y} & P_{2,y} & P_{3,y} & P_{4,y} \end{pmatrix}\right]\begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$

Affine Transformations

- Homogeneous coordinates also work
  - Means you can translate, rotate, shear, etc.
  - Note though that you need to normalize P' by 1/*w'*

$$P'(t) = \mathbf{M}\left[\begin{pmatrix} P_{1,x} & P_{2,x} & P_{3,x} & P_{4,x} \\ P_{1,y} & P_{2,y} & P_{3,y} & P_{4,y} \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}\right]$$

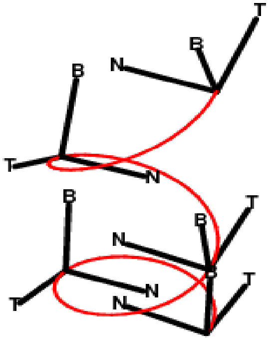
$$= \left[\mathbf{M} \begin{pmatrix} P_{1,x} & P_{2,x} & P_{3,x} & P_{4,x} \\ P_{1,y} & P_{2,y} & P_{3,y} & P_{4,y} \\ 1 & 1 & 1 & 1 \end{pmatrix}\right] \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$

The Plan for Today

- Differential Properties of Curves & Continuity
- B-Splines
- Surfaces
  - Tensor Product Splines
  - Subdivision Surfaces
  - Procedural Surfaces
  - Other

Differential Properties of Curves

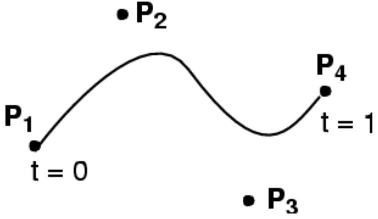
- Motivation
  - Compute normal for surfaces
  - Compute velocity for animation
  - Analyze smoothness



Velocity

- First derivative w.r.t. *t*
- Can you compute this for Bezier curves?

$$P(t) = (1-t)^3\, P1 + 3t(1-t)^2\, P2 + 3t^2(1-t)\, P3 + t^3\, P4$$

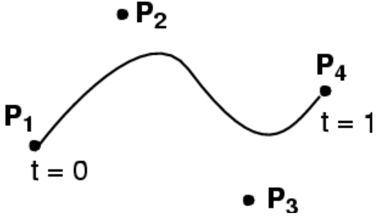


- You know how to differentiate polynomials...

Velocity

- First derivative w.r.t. *t*
- Can you compute this for Bezier curves?

$$P(t) = (1-t)^3 \; P1 + 3t(1-t)^2 \; P2 + 3t^2(1-t) \; P3 + t^3 \; P4$$



- **P'(t) = -3(1-t)2 P1**
  + [3(1-t) 2 -6t(1-t)] P2
  + [6t(1-t)-3t 2] P3
  + 3t 2 P4

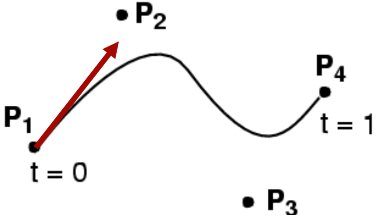
Sanity check: t=0; t=1

Linearity?

- Differentiation is a linear operation
  - (f+g)'=f'+g'
  - (af)'=a f'
- This means that the derivative of the basis is enough to know the derivative of any spline.
- Can be done with matrices
  - Trivial in monomial basis
  - But get lower-order polynomials

Tangent Vector

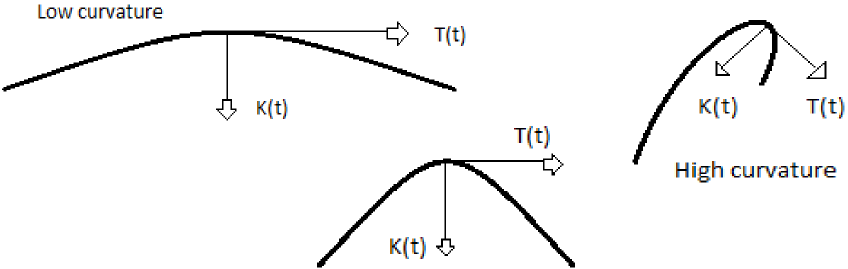
- The tangent to the curve P(t) can be defined as T(t)=P'(t)/||P'(t)||
  - normalized velocity, ||T(t)|| = 1
- This provides us with one orientation for swept surfaces later



Courtesy of Seth Teller.

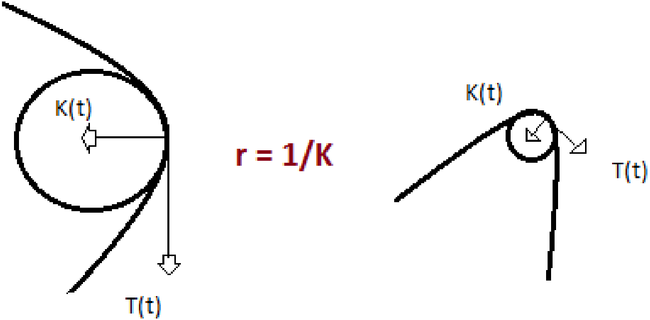
Curvature Vector

- Derivative of unit tangent
  - K(t)=T'(t)
  - Magnitude ||K(t)|| is constant for a circle
  - Zero for a straight line
- Always orthogonal to tangent, ie. $K \cdot T = 0$



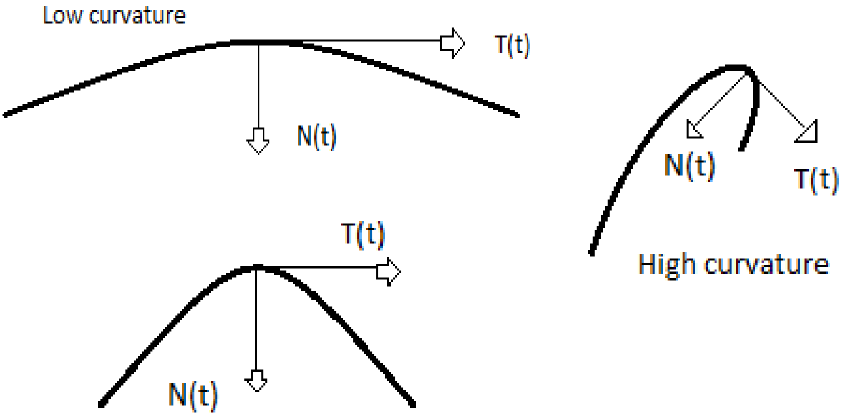
Geometric Interpretation

- K is zero for a line, constant for circle
  - What constant? 1/r
- 1/||K(t)|| is the radius of the circle that touches P(t) at *t* and has the same curvature as the curve



Curve Normal

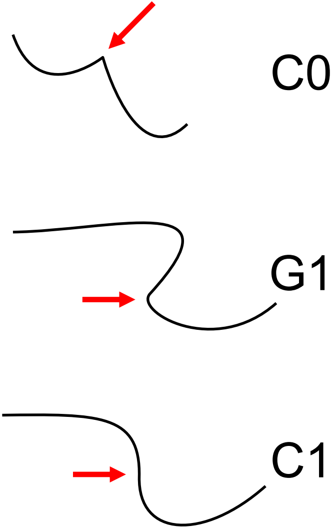
- Normalized curvature: T’(t)/||T’(t)||



Questions?

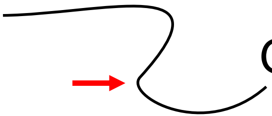
Orders of Continuity

- C0 = continuous
  - The seam can be a sharp kink
- G1 = geometric continuity
  - Tangents **point to the same direction** at the seam
- C1 = parametric continuity
  - Tangents **are the same** at the seam, implies G1
- C2 = curvature continuity
  - Tangents and their derivatives are the same

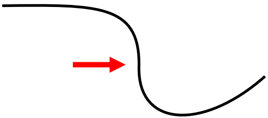


Orders of Continuity

- G1 = geometric continuity
  - Tangents **point to the same direction** at the seam
  - good enough for modeling
- C1 = parametric continuity
  - Tangents **are the same** at the seam, implies G1
  - often necessary for animation

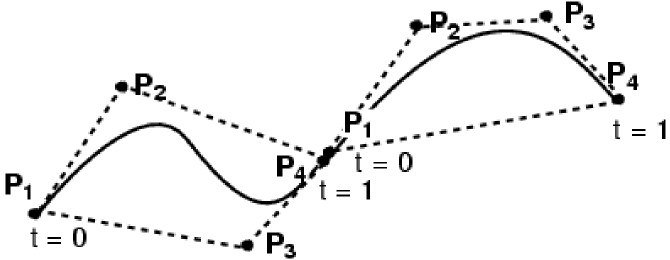


G1



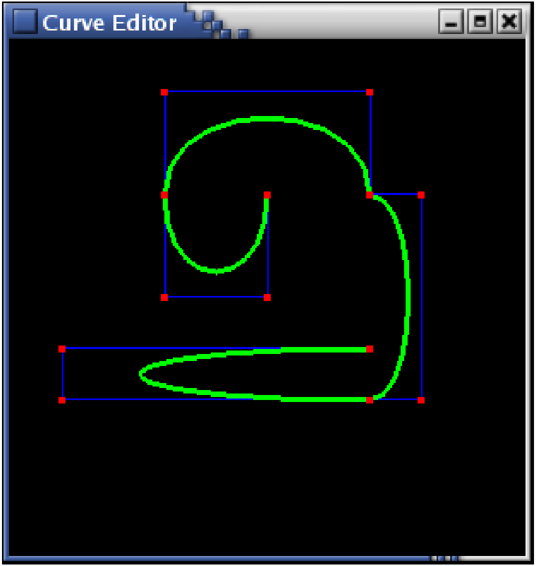
C1

Connecting Cubic Bézier Curves



- How can we guarantee C0 continuity?
- How can we guarantee G1 continuity?
- How can we guarantee C1 continuity?
- C2 and above gets difficult

Connecting Cubic Bézier Curves



- Where is this curve
  - C0 continuous?
  - G1 continuous?
  - C1 continuous?
- What's the relationship between:
  - the # of control points, and the # of cubic Bézier subcurves?

Questions?