
Introduction to Polynomial Curves Part II

Peter Schröder

CS175 2000

1

Affine Space

Affine Combination

$$u_k \in \mathbb{R}^n \quad \sum \alpha_k u_k \quad \sum \alpha_k = 1$$

Affine Map

$$f: \mathbb{R}^n \rightarrow \mathbb{R}^m \quad f(\sum \alpha_k u_k) = \sum \alpha_k f(u_k)$$

- exactly the linear maps and translation

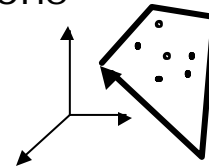
CS175 2000

2

Affine Space

Definition

- subset of vector space closed under affine combinations



- exactly the linear subspaces plus translation

CS175 2000

3

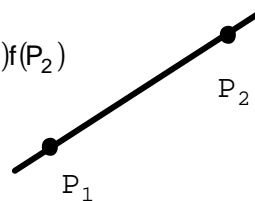
Affine Forms

Evaluation

$$f(P) = f(tP_1 + (1-t)P_2) = tf(P_1) + (1-t)f(P_2)$$

Differentiation

$$\begin{aligned} Dg|_P &= \lim_{\tau \rightarrow 0} \frac{g(P + \tau) - g(P)}{\tau} = \\ &= g(P+1) - g(P) \end{aligned}$$



$$\begin{aligned} g(t) &= at + b \\ Dg(t) &= g(1) - g(0) = a \end{aligned}$$

CS175 2000

4

Polar Forms

Blossom

- For every polynomial $F(u)$ of degree n there exists a unique symmetric multiaffine map $f(u_1, \dots, u_n)$ for which $F(u) = f(u, \dots, u)$. This map is called the polar form. $F(u)$ is called its diagonal and f is called the blossom of F

CS175 2000

5

Example

$$F(u) = a_0 + a_1u + a_2u^2 + a_3u^3$$

$$\begin{aligned} f(u_1, u_2, u_3) = & a_0 \\ & + \frac{a_1}{3}(u_1 + u_2 + u_3) \\ & + \frac{a_2}{3}(u_1u_2 + u_2u_3 + u_3u_1) \\ & + a_3u_1u_2u_3 \end{aligned}$$

CS175 2000

6

Polar Form

Correspondence

$$F(u) = \sum_{i=0}^n a_i u^i \quad f(u_1, \dots, u_n) = \sum_{i=0}^n a_i \binom{n}{i}^{-1} \sum_{\substack{S \subseteq \{1, \dots, n\} \\ |S|=i}} \prod_{j \in S} u_j$$

Differentiation

- differencing

$$\Delta[\tau]f(u_1, \dots, u_{n-1}) := f(u_1, \dots, u_n + \tau) - f(u_1, \dots, u_n)$$

- well defined

- linear in tau

Polar Form

Differentiation

$$F^{(q)}(u) = \frac{n!}{(n-q)!} \Delta[1^q]f(u^{n-q})$$

$$a_q = \frac{F^{(q)}(0)}{q!} = \binom{n}{q} \Delta \left[\underbrace{1, \dots, 1}_q \right] f \left(\underbrace{0, \dots, 0}_{n-q} \right) = \binom{n}{q} \Delta[1^q]f(0^{n-q})$$

Continuity

Two Polynomials

- Let F and G be two polynomials of degree n and let u be a point on the real line. Then F and G are C^q continuous at u iff

$$f(u^{n-q}, u_1, \dots, u_q) = g(u^{n-q}, u_1, \dots, u_q) \\ u_1, \dots, u_q \in \mathbb{R}$$

CS175 2000

9

Bezier Points

Finding Control Points

- The Bezier control points of F w.r.t $[s, t]$ are

$$b_i = f(s^{n-i}t^i) \quad i = 0, \dots, n \quad u = \frac{t-u}{t-s}s + \frac{u-s}{t-s}t$$

Proof

$$F(u) = f(u^n) = \left(\frac{t-u}{t-s}\right)^n f(s^{n-1}) + \left(\frac{u-s}{t-s}\right)^n f(u^{n-1}t) \\ = \dots = \sum_{i=0}^n B_i^{\Delta, n}(u) f(s^{n-i}t^i)$$

CS175 2000

10

de Casteljau Alg.

Evaluate Bezier form

$$b_i^0(u) = b_i \quad i = 0, \dots, n$$

$$b_i^l(u) = \left(\frac{t-u}{t-s}\right) b_i^{l-1}(u) + \left(\frac{u-s}{t-s}\right) b_{i+1}^{l-1}(u)$$

$$l = 1, \dots, n \quad i = 0, \dots, n-l$$

- very stable, uses convex combinations only

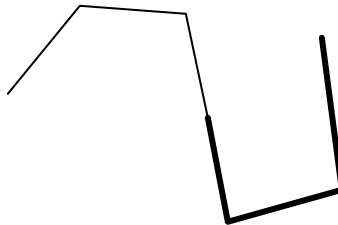
CS175 2000

11

Multiple Segments

How to stick them together?

- can match first derivative easily
- second derivative?



CS175 2000

12

Modeling

Bezier curves

- requires to keep track of continuity conditions explicitly... Possible, but very messy
- better: use basis which has continuity conditions built in!

B-Splines

CS175 2000

13

B-splines

Definition

- knotvector

$$T = (t_i)_{i \in \mathbb{Z}} \quad t_i < t_{i+n+1}$$

- non-decreasing sequence, can have multiple entries (multiplicity)

- curve:

$$C(u) = \sum N_i^n(u) d_i$$

CS175 2000

14

B-Splines

Definition

- recursive

$$N_i^0(u) = \begin{cases} 1 & t_i \leq u < t_{i+1} \\ 0 & \text{else} \end{cases}$$
$$N_i^l(u) = \frac{u - t_i}{t_{i+1} - t_i} N_i^{l-1}(u) + \frac{t_{i+2} - u}{t_{i+2} - t_{i+1}} N_{i+1}^{l-1}(u)$$

- support

$$N_i^l(u) = 0 \quad u \notin [t_i, t_{i+2}]$$

CS175 2000

15

B-Splines

de Boor Algorithm $t_n \leq \dots \leq r_1 < s_1 \leq \dots \leq s_n$

$$\begin{aligned} d_j^l(u) &= f(r_1, \dots, r_{n-l-j}, u, \dots, u, s_1, \dots, s_j) \\ &= \left(\frac{s_{j+1} - u}{s_{j+1} - r_{n-l-j+1}} \right) f(r_1, \dots, r_{n-l-j+1}, u, \dots, u, s_1, \dots, s_j) \\ &\quad + \left(\frac{u - r_{n-l-j+1}}{s_{j+1} - r_{n-l-j+1}} \right) f(r_1, \dots, r_{n-l-j}, u, \dots, u, s_j, \dots, s_{j+1}) \\ &= \left(\frac{s_{j+1} - u}{s_{j+1} - r_{n-l-j+1}} \right) d_j^{l-1}(u) + \left(\frac{u - r_{n-l-j+1}}{s_{j+1} - r_{n-l-j+1}} \right) d_{j+1}^{l-1}(u) \end{aligned}$$

CS175 2000

16

de Boor Algorithm

Recursion

$$N_i^0(u) = \begin{cases} 1 & t_i \leq u < t_{i+1} \\ 0 & \text{else} \end{cases}$$

$$N_i^l(u) = \frac{u - t_i}{t_{i+1} - t_i} N_i^{l-1}(u) + \frac{t_{i+2} - u}{t_{i+2} - t_{i+1}} N_{i+1}^{l-1}(u)$$

$$d_i^0(u) = d_i \quad i = j - n \dots j$$

$$d_i^l(u) = \frac{t_{i+n+1} - u}{t_{i+n+1} - t_{i+l}} d_i^{l-1}(u) + \frac{u - t_{i+l}}{t_{i+n+1} - t_{i+l}} d_{i+1}^{l-1}(u)$$

$$l = 1 \dots n \quad i = j - n, \dots, j - l$$

CS175 2000

17

B-Splines

Terms

- degree: highest exponent in polynomial
- order: 1+degree
- multiplicity of a knot: how many times repeated
- continuity: C^{n-m} with multiplicity m

CS175 2000

18

B-splines

Knot insertion

- create new degree of freedom along curve

$$t_j \leq t \leq t_{j+1}$$

$$d_i^* = \alpha_i d_i + (1 - \alpha_i) d_{i-1}$$

$$\alpha_i = \begin{cases} 1 & i \leq j - n \\ \frac{t - t_i}{t_{i+n} - t_i} & j - n + 1 \leq i \leq j \\ 0 & j + 1 \leq i \end{cases}$$

Surfaces

Tensor product case

- nothing special happens

$$F(u,v) = f(u^n; v^n)$$

- treat each parameter direction separately

Surfaces

Non-tensor product

$$F(u,v) = \sum_{i+j+k=n} p_{ijk} \binom{n}{i!j!k!} u^i v^j (1-u-v)^k$$

- example: quadratic in two variables

$$F(u,v) = a_{00} + a_{10}u + a_{01}v + a_{20}u^2 + a_{11}uv + a_{02}v^2$$

- Bezier functions for triangles
- B-patches? very hard.

Surfaces

Tensor product

- 2 parameter directions which decouple $F(u,v) = F_u(u)F_v(v)$

- example: bi-cubic patches

$$\begin{aligned} F(u,v) &= \sum_{i,j=0}^3 p_{ij} B_i^3(u) B_j^3(v) \\ &= (a_0 + a_1u + a_2u^2 + a_3u^3)(b_0 + b_1v + b_2v^2 + b_3v^3) \end{aligned}$$

Tensor Product Surf.

Polarform

- reduces to 1D setting

$$F(u,v) = F_u(u)F_v(v) = f_u(uuu)f_v(vvv) = f(uuu;vvv)$$

$$F(u,v) = \sum_{i=0}^n \sum_{j=0}^m f(u_1^{n-i}u_2^i; v_1^{m-j}v_2^j) B_i^n(u) B_j^m(v)$$

- control points of tensor product surface are outerproduct of control points for 1D components