A Short Survey on Curves and Surfaces in CAGD

Katja Bühler
Institut für Computergrafik und Algorithmen
TU Wien

Topics

Parametric curves and surfaces

Polynomial curves

Rational curves

Tensor product surfaces

Triangular surfaces

Parametric Curves

$$c: \quad \mathbf{c}(u) = \begin{pmatrix} x(u) \\ y(u) \\ z(u) \end{pmatrix}; \quad u \in [a,b] =: I \subset \mathrm{IR}$$

- $x(u), y(u), z(u)$ are differentiable functions in u
- Tangent vector : $\mathbf{t}(u) = \frac{d}{du}\mathbf{c}(u)$
- c regular in $\mathbf{c}(u_0) \Leftrightarrow \mathbf{t}(u_0) \neq \mathbf{o}$
- c regular $\Leftrightarrow$ c can be parametrized in a way that all curve points are regular.

Parametric Surfaces

$$s: \quad \mathbf{s}(u,v) = \begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix}; \quad (u,v) \in [a,b] \times [c,d] =: I \times J \subset \mathrm{IR}^2$$

- $x(u,v), y(u,v), z(u,v)$ are differentiable functions in u and v
- Tangent plane : $\mathbf{t}_0(l,m) = \mathbf{s}(u_0,v_0) + l\,\mathbf{s}_u(u_0,v_0) + m\,\mathbf{s}_v(u_0,v_0)$
- Normal vector : $\mathbf{n}(u_0,v_0) = \mathbf{s}_u(u_0,v_0) \times \mathbf{s}_v(u_0,v_0)$
- s regular in $\mathbf{s}(u_0,v_0) \Leftrightarrow \mathbf{n}(u_0,v_0) \neq \mathbf{o}$
- s regular $\Leftrightarrow$ s can be parametrized in a way that all surface points are regular.

Bézier Curves: The de Casteljau Algorithm

Given n + 1 points $\mathbf{b}_0,...,\mathbf{b}_n \in IE^3$ and an arbitrary $t \in IR$.

Set

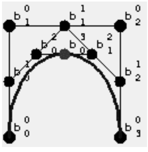
$$\mathbf{b}_i^r := (1\text{-}t)\,\mathbf{b}_i^{r-1} + t\,\mathbf{b}_{i+1}^{r-1} \quad \text{with } \mathbf{b}_i^0 := \mathbf{b}_i.$$

Then $\mathbf{b}_0^n$ is a curve point on the corresponding Bézier curve.

The points $\mathbf{b}_0^{n-1}$ and $\mathbf{b}_1^{n-1}$ determine the tangent line of the curve at point $\mathbf{b}_n^n$.

$\mathbf{b}_0^0$
$\mathbf{b}_0^1$
$\mathbf{b}_1^0$ $\mathbf{b}_0^1$
$\mathbf{b}_1^1$ $\mathbf{b}_0^3$
$\mathbf{b}_2^0$ $\mathbf{b}_1^2$
$\mathbf{b}_2^1$
$\mathbf{b}_3^0$

The de Casteljau scheme



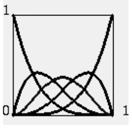
Bézier Curves and Bernstein Polynomials

Bézier curve with respect to Bézier points $\mathbf{b}_i$, i = 0,..., n :

$$\mathbf{b}(t) = \sum_{i=0}^{n} \mathbf{b}_i B_i^n(t)$$

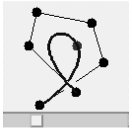
Bernstein polynomials :

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



Properties of Bézier Curves

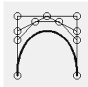
Affine invariance
Convex hull property
Endpoint interpolation
Linear precision
Variation diminishing property



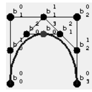
Disadvantages
Only pseudo local control
High degree

Important Algorithms for Bézier Curves

Degree elevation
- to increase flexibility



Subdivision
- to increase flexibility
- to approximate the curve



The subdivision is a byproduct of the de Casteljau algorithm

Evaluation/Approximation of Bézier Curves

The **de Casteljau** algorithm is numerical stabil, but inefficient for evaluation

Horner scheme like evaluation is more efficient

$$\mathbf{b}(t) = (\ldots((\binom{n}{0} s\mathbf{b}_0 + \binom{n}{1} t\mathbf{b}_1)s + \binom{n}{2} t^2\mathbf{b}_2)s + \ldots)s + \binom{n}{n} t^n \mathbf{b}_n \quad \text{with } s = 1\text{-}t$$

Repeated subdivision gives in a fast way a good approximation of the curve

B-Spline Curves

B-spline curves

are piecewise polynomial curves of degree k-1

have a degree (almost) independent of the number of control points

allow local control over the shape of a curve

B-Spline Curves: Definition

Given :

- n+1 control points $\mathbf{d}_i \in IE^3,\ i = 0,\ldots,n$
- knot vector $U = (u_0,\ldots,u_{n+k})$

B-spline curve :

$$\mathbf{s}(u) = \sum_{i=0}^{n} \mathbf{d}_i N_i^k(u), \quad u \in [u_0, u_{n+k})$$

with the B-spline basis functions $N_i^k(u)$ of order k.

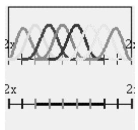
B-Spline Basis Functions

$$k = 1: \quad N_i^1(u) = \begin{cases} 1 & u \in [u_i, u_{i+1}) \\ 0 & sonst \end{cases}$$

$$k > 1: \quad N_i^k(u) = \frac{u - u_i}{u_{i+k-1} - u_i} N_i^{k-1}(u) + \frac{u_{i+k} - u}{u_{i+k} - u_{i+1}} N_{i}^{k-1}(u)$$

Properties of the basis functions :

- partition of unity : $\sum_{i=0}^{n} N_i^k(u) \equiv 1$
- positivity : $N_i^k(u) \geq 0$
- local support : $N_i^k(u) = 0 \quad \text{if } u \notin [u_i, u_{i+k})$



Properties of B-Spline Curves

Affine invariance

Strong convex hull property

Variation diminishing property

Local support

Knot points of multiplicity k are coincident with one of the control points.

A B-spline curve of order k which has only knots of multiplicity k is a Bézier curve

Evaluating B-Spline Curves: The de Boor Algorithm

Given

n+1 control points $\mathbf{d}_0,\dots,\mathbf{d}_n \in IE^3$, a knot vector $U = (u_0,\dots,u_{n+k})$ and an arbitrary $t \in [u_0, u_{n+k})$.

Set

$$\mathbf{d}_i^r := (1-\alpha_i^r)\,\mathbf{d}_{i-1}^{r-1} + \alpha_i^r\,\mathbf{d}_i^{r-1}$$

with $\mathbf{d}_i^0 := \mathbf{d}_i$ and $\alpha_i^r := \dfrac{t-u_i}{u_{i+k-r}-u_i}$

Then $\mathbf{d}_m^{k-1}$ is the point for $\mathbf{x}(t_0)$, $t_0 \in [t_m, t_{m+1}]$ on the corresponding B-spline curve.

The points $\mathbf{d}_{m-1}^{k-2}$ and $\mathbf{d}_m^{k-2}$ determine the tangent line of the curve at point $\mathbf{b}_m^{k-1}$.

$\mathbf{d}_0^0$

$\mathbf{d}_1^1$

$\mathbf{d}_1^0$ $\mathbf{d}_2^2$

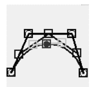
$\mathbf{d}_2^1$ $\mathbf{d}_3^3$

$\mathbf{d}_2^0$ $\mathbf{d}_3^2$

$\mathbf{d}_3^1$

$\mathbf{d}_3^0$

The de Boor scheme



Direct Evaluation of B-Spline Curves

Find the knot span $[u_i,u_{i+1})$ in which the parameter value t lies

Compute all non zero basis functions

Multiply the values of the nonzero basis functions with the corresponding control points

Special B-Spline Curves

open:

$$u_0 = \dots = u_k < u_{k+1} < \dots < u_n = \dots = u_{n+k}$$

closed:

$$\mathbf{d}_{n+1} := \mathbf{d}_0, \dots, \mathbf{d}_{n+k} := \mathbf{d}_{k-1} \text{ und}$$
$$U = (u_0, \dots, u_{n+k}, \dots, u_{n+2k-2})$$

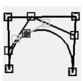
uniform:

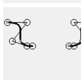
$$U = (u_0, u_0 + d, \dots, u_0 + (n+k)d)$$

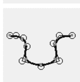
Important Algorithms for B-Spline Curves

Knot insertion
- to increase flexibility
- to compute derivatives
- to split curves (subdivision algorithms)
- to evaluate the curve (see de Boor algorithm)
- to approximate the curve

Degree elevation
- to adapt curve degrees







Rational Curves

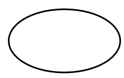
Rational curve:

$$\mathbf{c}(u) = \frac{1}{w(u)}\begin{pmatrix} x(u) \\ y(u) \\ z(u) \end{pmatrix}, \quad u \in I \subset IR$$

Homogeneous representation:

$$\mathbf{c}_H(u) = \begin{pmatrix} w(u) \\ x(u) \\ y(u) \\ z(u) \end{pmatrix}, \quad u \in I \subset IR$$

Example: conic section



$$\mathbf{c}(u) = \frac{1}{1+u^2}\begin{pmatrix} a\,(1-u^2) \\ b\,2u \end{pmatrix}$$

$$\mathbf{c}_H(u) = \begin{pmatrix} 1+u^2 \\ a\,(1-u^2) \\ b\,2u \end{pmatrix}$$

Rational Bézier Curves

A rational Bézier curve is defined as

$$\mathbf{b}(u) = \frac{\sum_{i=0}^{n} w_i \mathbf{b}_i B_i^n(u)}{\sum_{i=0}^{n} w_i B_i^n(u)}, \quad u \in I \subset IR$$

The $w_i > 0,\ i = 0,\dots,n$ are called weights.

Homogeneous representation :

$$\mathbf{b}_H(u) = \sum_{i=0}^{n} \mathbf{b}_{H\,i}\ B_i^n(u), \quad u \in I \subset IR$$

with the homogeneous Bézier points

$$\mathbf{b}_{H\,i} = \begin{pmatrix} w_i \\ w_i \mathbf{b}_i \end{pmatrix}$$

Rational Bézier Curves: Properties and Algorithms

Properties:
- the same properties like polynomial curve, and
- projective invariance
- the weights are an additional design parameter

Algorithms
- All algorithms of polynomial Bézier curves can be applied without any change to the homogeneous representation of rational Bézier curves.

Non Uniform Rational B-Splines

A NURBS curve with respect to the control points $d_i,\ i = 0,\dots,n$ and the knot vector $U = (u_0,\dots,u_{n+k})$ is defined as

$$\mathbf{n}(u) = \frac{\sum_{i=0}^{n} w_i \mathbf{d}_i N_i^k(u)}{\sum_{i=0}^{n} w_i N_i^k(u)}, \quad u \in [u_0, u_{n+k}) \subset IR$$

The $w_i > 0,\ i = 0,\dots,n$ are called weights.

Homogeneous representation :

$$\mathbf{n}_H(u) = \sum_{i=0}^{n} \mathbf{n}_{H\,i}\ N_i^k(u), \quad u \in [u_0, u_{n+k}) \subset IR$$

with the homogeneous Bézier points

$$\mathbf{n}_{H\,i} = \begin{pmatrix} w_i \\ w_i \mathbf{n}_i \end{pmatrix}$$

NURBS: Properties and Algorithms

Properties:
- the same properties like polynomial curve, and
- projective invariance
- changing the weight w_i affects only the interval $[u_i, u_{i+k})$

Algorithms
- All algorithms of B-spline curves can be applied without any change to the homogeneous representation of NURBS curves.

Tensorproduct surfaces

"A surface is the locus of a curve that is moving through space and thereby changing the shape"

Given a curve

$$\mathbf{f}(u) = \sum_{i=0}^{n} \mathbf{c}_i F_i(u), \quad u \in I \subset IR$$

moving the control points yields

$$\mathbf{c}_i(v) = \sum_{j=0}^{m} \mathbf{a}_{ij} G_j(v), \quad v \in J \subset IR$$

Combining both yields a tensor product surface

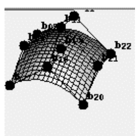
$$s(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} \mathbf{a}_{ij} F_i(u) G_j(v), \quad (u,v) \in I \times J \subset IR^2$$

Tensorproduct Bézier Surfaces

A tensorproduct Bézier surface is given by

$$\mathbf{b}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} \mathbf{b}_{ij} B_i^n(u) B_j^m(v), \quad (u,v) \in I \times J \subset IR^2$$

The Bézier points $\mathbf{b}_{ij}$ form the control net of the surface.



Tensor Product Bézier Surfaces Properties and Algorithms

Properties:

analogue to that of Bézier curves

Algorithms:

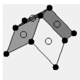
Apply algorithms for curves in two steps:

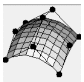
- Apply to

$$\mathbf{b}_i(v) = \sum_{j=0}^{m} \mathbf{b}_{ij} B_j^m(v), \quad i = 0,\ldots,n$$

- Apply to

$$\mathbf{b}(u,v) = \sum_{i=0}^{n} \mathbf{b}_i(v) B_i^n(u)$$





Tensorproduct B-Spline Surfaces

A tensorproduct B-spline surface with respect to the knot vectors

$U = (u_0,\ldots,u_{n+k}), V = (v_0,\ldots,v_{m+l})$

is given by

$$\mathbf{d}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} \mathbf{d}_{ij} N_i^k(u) N_j^l(v), \quad (u,v) \in [u_0, u_{n+k}] \times [v_0, v_{m+l}] \subset IR^2.$$

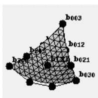
The control points $\mathbf{d}_{ij}$ form the control net of the surface.

Properties and Algorithms are analogue to the description for Bézier tensor product surfaces.

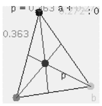
Bézier Triangles

A triangular Bézier patch is defined by

$$\mathbf{x}(u,v) = \sum_{\substack{i+j+k=1 \\ i,j,k>0}} \mathbf{b}_{ijk} B_{ijk}^n(u,v,w)$$

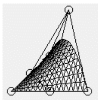


The u,v,w are baryzentric coordinates of the triangular parameter domain.



Generalized Bernstein polynomials:

$$B_{ijk}^n(u,v,w) = \frac{n!}{i!\,j!\,k!} u^i v^j w^k$$



Bézier Triangles: Properties and Algorithms

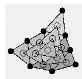
Properties:

- the same as in the univariate case

Algorithms:

- De Casteljau:

$$\mathbf{b}_{ijk}^l = u\mathbf{b}_{i+1\,jk}^{l-1} + v\mathbf{b}_{ij+1k}^{l-1} + w\mathbf{b}_{ijk+1}^{l-1}$$



- Subdivision
- Degree elevation