Algorithms for B-patches

Hans-Peter Seidel*
Computer Graphics Laboratory
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

Abstract

B-patches are the analogue to B-spline segments for triangular surfaces and are the main building block in the new multivariate B-spline surfaces recently developed in [8]. This paper discusses algorithms for B-patches and presents algorithms for computing the polar form, for evaluation, for differentiation, for computing continuous joints, and for knot insertion.

Keywords: Bézier patch, blossom, de Boor Algorithm, B-patch, de Casteljau Algorithm, control point, knot net, polar form, triangular patch.

Introduction

Tensor product B-spline surfaces provide an excellent tool for modeling free form surfaces over rectangular domains. However, as is well known, the design of more complicated real world objects often also requires the additional capability of modeling with surfaces over triangular grids. Standard examples for this situation are the so called polygonal hole problem and the construction of smooth blends. Unfortunately, most of these surfaces cannot be modeled by tensor product patches without singularities.

One way to deal with this situation is the introduction of triangular Bézier patches. These surfaces have been widely studied in the past [12, 30, 16, 17, 14, 11], and are well understood by now. However, although triangular Bézier patches allow for the modeling of surfaces over arbitrary triangular regions, they in turn lack certain other features that make interactive surface design with tensor product B-splines so attractive and easy. Therefore, not surprisingly, there has been ongoing research on the construction of more flexible surface representations for many years [22, 23].

Recently, a new B-spline like surface representation has been developed in [8]. This representation is based on a redevelopment of multivariate B-splines using B-patches. A test implementation of these surfaces is currently under way at the University of Waterloo. The central building blocks in the new surface representation are B-patches as developed in [33, 34]. B-patches share many properties with B-spline segments: They are characterized by their control points and by a 3-parameter family of knots. If the knots in each family coincide, we obtain the Bézier representation of a bivariate polynomial over a triangle. Therefore B-patches subsume Bézier patches in much the same way B-spline segments subsume Bézier curves. B-patches have a de Boor-like evaluation algorithm, and, as in the case of B-spline curves, the control points of a B-patch can be expressed by simply inserting a sequence of knots into the corresponding polar form. B-patches can be joined smoothly, and they have an algorithm for knot insertion that is completely similar to the insertion algorithm for curves. Therefore, B-patches may be considered as the analogue to B-spline segments for surfaces.

This paper focuses on computational aspects of B-patches and presents algorithms for computing the polar form, for evaluation, for differentiation, for computing continuous joints between two adjacent patches, and for knot insertion. The paper is organized as follows: Section 2 gives a brief introduction to the theory of polar forms for surfaces that is necessary for the construction of B-patches. Section 3 defines B-patches by means of their de Boor-like evaluation algorithm and discusses some of their basic features. Section 4 focuses on algorithmic aspects

*This work has been partly supported by the Natural Sciences and Engineering Research Council of Canada through Strategic Operating Grant STR0040527

and presents algorithms for evaluation, differentiation, computation of continuous joints, and knot insertion. Section 5 contains concluding remarks and points out directions for future research.

Polynomials and polar forms

This section gives a brief introduction to the theory of polar forms for surfaces. In particular we establish the principle that for every polynomial F of degree n there exists a unique symmetric n-affine map f, the *polar form* of F, satisfying $f(u,\dots,u) = F(u)$, and we show how the derivatives of F can be expressed in terms of f. Additional material on polar forms can be found in [12, 13, 26, 27, 28, 32, 33].

Recall that a map $f : \mathbb{R}^2 \to \mathbb{R}^3$ is called *affine* if it preserves affine combinations, that is, if f satisfies

$$f\Big(\sum_{\mu=1}^{m} \alpha_\mu u^\mu\Big) = \sum_{\mu=1}^{m} \alpha_\mu f(u^\mu) \tag{1}$$

for all points $u^1, ..., u^m \in \mathbb{R}^2$ and real numbers $\alpha_1, ..., \alpha_m \in \mathbb{R}$ with $\sum_{\mu=1}^{m} \alpha_\mu = 1$. A map f is affine iff f can be written as composition of a linear map $A \in L(\mathbb{R}^2, \mathbb{R}^3)$ and a translation $b \in \mathbb{R}^3$, i.e.

$$f(u) = Au + b. \tag{2}$$

Therefore the directional derivative of an affine map f w.r.t. a vector $\xi = t - s$ can be written as

$$D_\xi f(u) = f(t) - f(s). \tag{3}$$

In particular, the derivative of f is independent of u.

A map $f : (\mathbb{R}^2)^n \to \mathbb{R}^3$ is called *n-affine* (or just *multiaffine*) if it is affine in each argument. Therefore f is n-affine iff for any $\nu = 1, ..., n$ and arbitrary points $a^1, ..., a^n \in \mathbb{R}^2$ each map

$$f_{a^1,\dots,\hat{a}^\nu,\dots,a^n} : \quad \mathbb{R}^2 \to \mathbb{R}^3 : \quad u \mapsto f(a^1, ..., a^{\nu-1}, u, a^{\nu+1}, ..., a^n)$$

is affine. Finally, a map $f : (\mathbb{R}^2)^n \to \mathbb{R}^3$ is called *symmetric* if f keeps its value under any permutation of its arguments.

With this notation we are now able to state the following one-to-one correspondence between polynomials of degree n and symmetric n-affine maps [13, 26]:

Theorem 2.1 *For every polynomial $F : \mathbb{R}^2 \to \mathbb{R}^3$ of degree n there exists a unique symmetric n-affine map $f : (\mathbb{R}^2)^n \to \mathbb{R}^3$ satisfying*

$$f(\underbrace{u, ..., u}_{n}) = F(u). \tag{4}$$

In this situation f is called the polar form *or* blossom *of F and F is the* diagonal *of f.* ♣

In order to join a network of surface patches in a smooth fashion it is necessary to derive precise conditions on the continuity between adjacent patches. Using the equivalence between polynomials and polar forms given by the previous theorem, these conditions can be stated as follows [26]:

Proposition 2.2 *Two polynomials $F, G : \mathbb{R}^2 \to \mathbb{R}^3$ of degree n match C^q-continuously at $s \in \mathbb{R}^2$ iff*

$$f(\underbrace{s, ..., s}_{n-q}, \underbrace{u^1, ..., u^q}_{q}) = g(\underbrace{s, ..., s}_{n-q}, \underbrace{u^1, ..., u^q}_{q}) \tag{5}$$

holds for every sequence $u^1, ..., u^q \in \mathbb{R}^2$, i.e. if the polar forms f and g agree on all arguments that contain s at least $(n-q)$-times. ♣

B-patches

This section gives the definition of B-patches and discusses some of their features that make them attractive for interactive surface design. We use standard multiindex notation $\vec{\imath} = (i, j, k)$ throughout.

First, we have to establish the analogue to the knot vector:

Definition 3.3 *A sequence*

$$\mathcal{K} = (r^0, ..., r^{n-1}, s^0, ..., s^{n-1}, t^0, ..., t^{n-1})$$

of parameters in the plane $\mathbb{R}^2$ is called a knot net *iff (r^i, s^j, t^k) are affinely independent for $0 \le |\vec{\imath}| \le n-1$. In this situation, the parameters are also called* knots. ♣

With this notation we are now able to define a B-patch over a knot net $\mathcal{K}$ by means of the following algorithm:

Definition 3.4 *Let the knot net*

$$\mathcal{K} = (r^0, ..., r^{n-1}, s^0, ..., s^{n-1}, t^0, ..., t^{n-1})$$

in $\mathbb{R}^2$ be given, and let $\rho_{\vec{\imath}}(u), \sigma_{\vec{\imath}}(u)$, and $\tau_{\vec{\imath}}(u)$ be the barycentric coordinates of $u \in \mathbb{R}^2$ w.r.t. $\triangle(r^i, s^j, t^k)$, i.e.

$$u = \rho_{\vec{\imath}}(u) r^i + \sigma_{\vec{\imath}}(u) s^j + \tau_{\vec{\imath}}(u) t^k.$$

and

$$\rho_{\vec{\imath}}(u) + \sigma_{\vec{\imath}}(u) + \tau_{\vec{\imath}}(u) = 1.$$

The algorithm

$$P^0_{\vec{\imath}}(u) = P_{\vec{\imath}} \tag{6}$$

and

$$\begin{aligned} P^l_{\vec{\imath}}(u) &= \rho_{\vec{\imath}}(u)P^{l-1}_{\vec{\imath}+\vec{e}^1}(u) \\ &\quad +\sigma_{\vec{\imath}}(u)P^{l-1}_{\vec{\imath}+\vec{e}^2}(u) \\ &\quad +\tau_{\vec{\imath}}(u)P^{l-1}_{\vec{\imath}+\vec{e}^3}(u) \end{aligned} \tag{7}$$

for $1 \le l \le n$, *is called* de Boor algorithm *for polynomials over triangles. The resulting surface* $F(u) = P^n_{(0,\ldots,0)}(u)$ *is a B-patch over* $\mathcal{K}$ *with* control points $P_{\vec{\imath}} \in \mathbb{R}^3$. ♣

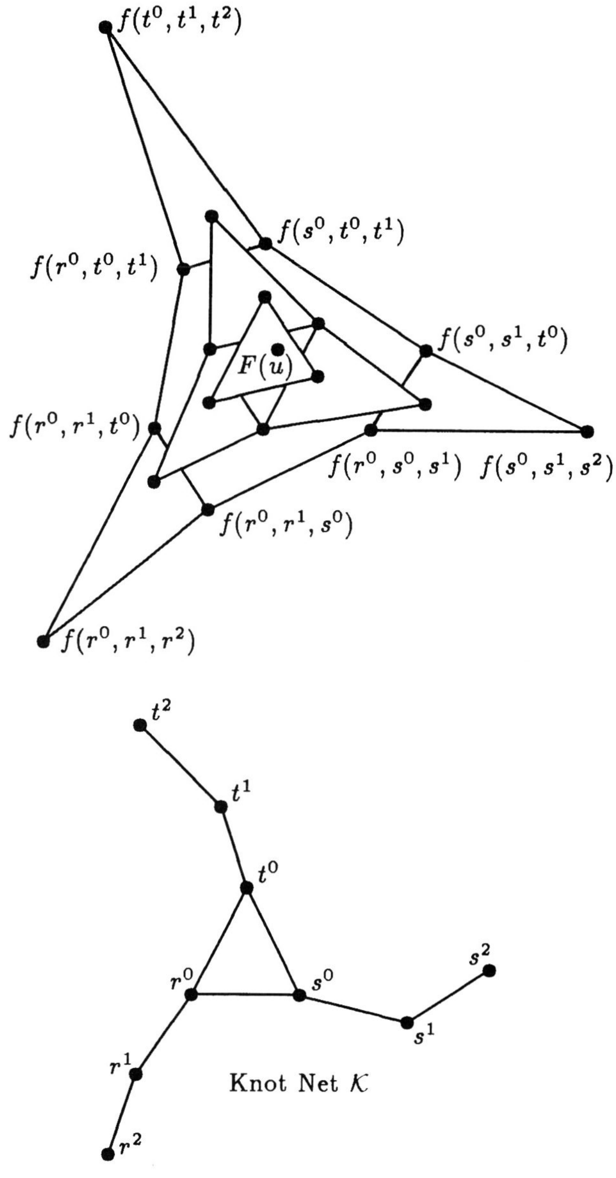

Figure 1: The de Boor algorithm for the evaluation of a cubic B-patch over the given knot net $\mathcal{K}$.

It is shown in [33] that every polynomial surface F of degree n can be represented as a B-patch over an arbitrary knot net $\mathcal{K} = (r^0, \ldots, r^{n-1}, s^0, \ldots, s^{n-1}, t^0, \ldots, t^{n-1})$ in the parameter domain $\mathbb{R}^2$. More specifically, the control points $P_{\vec{\imath}}$ in this representation are obtained by evaluating the corresponding polar form f on sequences of consecutive knots in $\mathcal{K}$, i.e.

$$P_{\vec{\imath}} = f(r^0, \ldots, r^{i-1}, s^0, \ldots, s^{j-1}, t^0, \ldots, t^{k-1}). \tag{8}$$

A special situation in Definition 5.2 arises if the control points in the de Boor algorithm are given by

$$P_{\vec{\imath}'} = \begin{cases} 1 & \text{if } \vec{\imath}' = \vec{\imath} \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

The resulting real-valued functions $N^n_{\vec{\imath}}(u)$ are called *normalized B-weights.* Using the normalized B-weights, every polynomial surface of degree n can then be represented in the form

$$F(u) = \sum_{|\vec{\imath}|=n} N^n_{\vec{\imath}}(u) P_{\vec{\imath}}, \tag{10}$$

as a weighted average of its B-patch control points.

The following example may be helpful to clarify these concepts:

Example 3.5 *Consider three affinely independent points* $r, s, t \in \mathbb{R}^2$. *Setting* $r^i = r$, $s^j = s$, $t^k = t$ *we obtain the familiar de Casteljau algorithm for the evaluation of a triangular Bézier surface*

$$F(u) = \sum_{|\vec{\imath}|=n} B^n_{\vec{\imath}}(u) P_{\vec{\imath}} \tag{11}$$

over the reference triangle $\triangle(r, s, t)$. *The control points*

$$P_{\vec{\imath}} = f(\underbrace{r, \ldots, r}_{i}, \underbrace{s, \ldots, s}_{j}, \underbrace{t, \ldots, t}_{k}) \tag{12}$$

are the Bézier points, and the polynomials $B^n_{\vec{\imath}}(u)$ *are the Bernstein polynomials w.r.t.* $\triangle(r, s, t)$. *In particular, B-patches subsume triangular Bézier patches in much the same way as B-splines subsume Bézier curves.* ♣

We conclude this section with a brief discussion on how the shape of a B-patch is related to the shape of its control net [33]:

Proposition 3.6 *Consider a B-patch*

$$F(u) = \sum_{|\vec{i}|=n} N_{\vec{i}}^n(u) P_{\vec{i}}, \quad P_{\vec{i}} \in \mathbb{R}^3$$

of degree n over

$$\mathcal{K} = (r^0, \ldots, r^{n-1}, s^0, \ldots, s^{n-1}, t^0, \ldots, t^{n-1}).$$

The shape of F is related to the shape of its control net $(P_{\vec{i}})_{|\vec{i}|=n}$ in the following way:

1. *Suppose that for $|\vec{i}| \leq n-1$ each triangle $\triangle(r^i, s^j, t^k)$ contains the domain triangle $\triangle(r^0, s^0, t^0)$. Then for $u \in \triangle(r^0, s^0, t^0)$ the function value $F(u)$ is contained in the convex hull of the control points $P_{\vec{i}}$.*

2. *If n knots $r^0 = \ldots = r^{n-1} =: r$ coincide, then $F(r) = P_{n,0,0}$ is a control point and the surface F is tangent to the control net at this point, i.e. the tangent plane at $F(r)$ is spanned by the points $P_{n,0,0}, P_{n-1,1,0}$ and $P_{n-1,0,1}$. The analogous assertion holds for the knots $s^0, \ldots, s^{n-1}$ and for the knots $t^0, \ldots, t^{n-1}$, respectively.*

3. *The relationship between F and its control net is affinely invariant: If $\phi : \mathbb{R}^3 \to \mathbb{R}^3$ is an affine map, the control points of the image surface $\phi \circ F$ are given as images $\phi(P_{\vec{i}})$ of the control points $P_{\vec{i}}$, i.e. the diagram*

$$\begin{array}{ccc} (P_{\vec{i}})_{|\vec{i}|=n} & \xrightarrow{\phi} & (\phi(P_{\vec{i}}))_{|\vec{i}|=n} \\ \downarrow & & \downarrow \\ F & \xrightarrow{\phi} & \phi \circ F \end{array}$$

commutes.

Proof: (1) Under the given assumptions the barycentric coordinates $\rho_{\vec{i}}(u)$, $\sigma_{\vec{i}}(u)$, and $\tau_{\vec{i}}(u)$ of a point $u \in \triangle(r^0, s^0, t^0)$ w.r.t. $\triangle(r^i, s^j, t^k)$ are all positive, and the point $P_{\vec{i}}^r(u)$ lies in the closed convex hull of $P_{\vec{i}+\vec{e}_1}^{l-1}(u)$, $P_{\vec{i}+\vec{e}_2}^{l-1}(u)$, and $P_{\vec{i}+\vec{e}_2}^{l-1}(u)$. Induction over l shows that $P_{\vec{i}}^l(u)$ is contained in the convex hull of the control points $P_{\vec{i}}$, and the assertion follows from $F(u) = P_{(0,0,0)}^n(u)$.

(2) It follows from Algorithm 4.9 below that the directional derivative of F at r w.r.t. $\xi = s^0 - r$ is given as

$$\begin{aligned} D_\xi F(r) &= nf(r, \ldots, r, \xi) \\ &= n(f(r, \ldots, r, s^0) - f(r, \ldots, r)) \\ &= n(P_{(n-1,1,0)} - P_{(n,0,0)}), \end{aligned}$$

so that the tangent plane at $F(r)$ is spanned by $P_{(n,0,0)}$, $P_{(n-1,1,0)}$, and $P_{(n-1,0,1)}$.

(3) Let f be the polar form of F, and let $\phi : \mathbb{R}^3 \to \mathbb{R}^3$ be an affine map. Then $\phi \circ f$ is the polar form of the image surface $\phi \circ F$, and the control points $P_{\vec{i}}^*$ of $\phi \circ F$ are given as

$$\begin{aligned} P_{\vec{i}}^* &= \phi \circ f(r^0, \ldots, r^{i-1}, s^0, \ldots, s^{j-1}, t^0, \ldots, t^{k-1}) \\ &= \phi(P_{\vec{i}}). \end{aligned}$$

This completes the proof of Proposition 3.6. ♣

Algorithms for B-patches

In this section we present some of the basic algorithms for modeling with B-patches. In particular, we present algorithms for evaluation, differentiation, computation of continuous joints, and knot insertion. These algorithms form the basis of the B-patch module within the interactive surface modeler TRIMO [36], and are also partly used in the implementation of the new multivariate B-spline surfaces developed in [8] that is currently under way at the University of Waterloo.

We start with an algorithm for computing the polar form f of a B-patch F out of the given control points. This algorithm is central for all other algorithms that follow.

Algorithm 4.7 (Polar Form) *Let a B-patch F over a knot net*

$$\mathcal{K} = (r^0, \ldots, r^{n-1}, s^0, \ldots, s^{n-1}, t^0, \ldots, t^{n-1})$$

with control points $P_{\vec{i}} \in \mathbb{R}^3$ be given. Let $\rho_{\vec{i}}(u^\nu), \sigma_{\vec{i}}(u^\nu)$, and $\tau_{\vec{i}}(u^\nu)$ be the barycentric coordinates of $u^\nu \in \mathbb{R}^2$ w.r.t. $\triangle(r^i, s^j, t^k)$, i.e.

$$u^\nu = \rho_{\vec{i}}(u^\nu) r^i + \sigma_{\vec{i}}(u^\nu) s^j + \tau_{\vec{i}}(u^\nu) t^k.$$

and

$$\rho_{\vec{i}}(u^\nu) + \sigma_{\vec{i}}(u^\nu) + \tau_{\vec{i}}(u^\nu) = 1.$$

Define

$$p_{\vec{i}}^0() := P_{\vec{i}}, \tag{13}$$

and

$$\begin{aligned} p_{\vec{i}}^l(u^1, \ldots, u^l) = \\ \rho_{\vec{i}}(u^l) p_{\vec{i}+\vec{e}_1}^{l-1}(u^1, \ldots, u^{l-1}) \\ +\sigma_{\vec{i}}(u^l) p_{\vec{i}+\vec{e}_2}^{l-1}(u^1, \ldots, u^{l-1}) \\ +\tau_{\vec{i}}(u^l) p_{\vec{i}+\vec{e}_3}^{l-1}(u^1, \ldots, u^{l-1}) \end{aligned} \tag{14}$$

for $1 \le l \le n$. Then the polar form f of F is given by

$$f(u^1, \ldots, u^n) = p^n_{(0,0,0)}(u^1, \ldots, u^n).$$

In particular, the maps $p^l_{\vec{\imath}}(u^1, \ldots, u^l)$ do not depend on the ordering of the arguments $u^1, \ldots, u^l$. ♣

A full proof of Algorithm 4.7 is given in [33]. The proof is based on the fact that the de Boor algorithm of the previous section has been carefully constructed in such a way that the maps $p^l_{\vec{\imath}}(u^1, \ldots, u^l)$ that appear in Algorithm 4.7 are symmetric throughout.

Note that the de Boor algorithm of the previous section is actually a special case of Algorithm 4.7:

Algorithm 4.8 (Evaluation) *If the parameters $u^1 = \ldots = u^n = u$ in Algorithm 4.7 all coincide, then Algorithm 4.7 reduces to the de Boor algorithm of Definition 3.4. Therefore in this situation*

$$p^n_{(0,0,0)}(u, \ldots, u) = P^n_{(0,0,0)}(u) = F(u)$$

is a point on the surface. ♣

Algorithm 4.7 may also be used for differentiation:

Algorithm 4.9 (Differentiation) *Let the direction vectors $\xi_1, ..., \xi_q$ in $\mathbb{R}^2$ be given. Then the q-th directional derivative*

$$D^{(q)}_{\xi_1, \ldots, \xi_q} F(u)$$

of F at u can be computed as

$$D^{(q)}_{\xi_1, \ldots, \xi_q} F(u) = \frac{n!}{(n-q)!} p^n_{(0,0,0)}(u, \ldots, u, \xi_1, ..., \xi_q)$$

with $p^n_{0,0,0}(u, \ldots, u, \xi_1, ..., \xi_q)$ given by Algorithm 4.7. Note, however, that the coordinates $\rho_{\vec{\imath}}(\xi_\nu)$, $\sigma_{\vec{\imath}}(\xi_\nu)$, $\tau_{\vec{\imath}}(\xi_\nu)$ in the expression

$$\xi_\nu = \rho_{\vec{\imath}}(\xi_\nu) r^i + \sigma_{\vec{\imath}}(\xi_\nu) s^j + \tau_{\vec{\imath}}(\xi_\nu) t^k$$

add up to 0 instead of 1, since ξ_ν is a vector, and not a point in $\mathbb{R}^2$. ♣

A proof of Algorithm 4.9 follows e.g from [26, 8.4] or from [33, 2.4]. A direct proof is given in [35].

Algorithm 4.9 can be used to convert from B-patch to monomial form:

Algorithm 4.10 (Monomial Form) *Consider a B-patch*

$$F(u) = \sum_{|\vec{\imath}|=n} N^n_{\vec{\imath}}(u) P_{\vec{\imath}}$$

over the knot net

$$\mathcal{K} = (r^0, \ldots, r^{n-1}, s^0, \ldots, s^{n-1}, t^0, \ldots, t^{n-1}).$$

The coefficients $a_{j,k}$ of the monomial representation

$$F(u) = \sum_{j+k \le n} a_{j,k}\ u_1^j u_2^k \tag{15}$$

can be computed as

$$a_{j,k} = \binom{n}{ijk} p^n_{(0,0,0)}(\underbrace{o, .., o}_{i}, \underbrace{\delta_x, .., \delta_x}_{j}, \underbrace{\delta_y, .., \delta_y}_{k}) \tag{16}$$

with $o = (0,0)$, $\xi_x = (1,0)$, $\xi_y = (0,1) \in \mathbb{R}^2$, and $p^n_{(0,0,0)}(o, \ldots, o, \delta_x, \ldots, \delta_x, \delta_y, \ldots, \delta_y)$ given by Algorithm 4.9. Again, the coefficients $\rho_{\vec{\imath}}(\xi_{x(y)})$, $\sigma_{\vec{\imath}}(\xi_{x(y)})$, $\tau_{\vec{\imath}}(\xi_{x(y)})$ in the expression

$$\xi_{x(y)} = \rho_{\vec{\imath}}(\xi_{x(y)}) r^i + \sigma_{\vec{\imath}}(\xi_{x(y)}) s^j + \tau_{\vec{\imath}}(\xi_{x(y)}) t^k$$

add up to 0 instead of 1, since $\xi_{x(y)}$ is a vector, and not a point in $\mathbb{R}^2$.

Proof: We briefly sketch a proof that Algorithm 4.10 is correct: Taylor expansion of F at $o = (0,0) \in \mathbb{R}^2$ gives

$$F(u) = \sum_{j+k \le n} \frac{\partial^{j+k} F(o)}{\partial_x^j \partial_y^k} \frac{u_1^j u_2^k}{j!k!},$$

and Algorithm 4.9 yields

$$\begin{aligned} F(u) &= \sum_{|\vec{\imath}|=n} \frac{n!}{i!}\ p^n_{(0,0,0)}(o, \ldots, \delta_y) \frac{u_1^j u_2^k}{j!k!} \\ &= \sum_{j+k \le n} a_{j,k} \frac{u_1^j u_2^k}{j!k!}. \end{aligned}$$

From this the assertion follows. ♣

Algorithm 4.9 also allows us to join two B-patches F and G along a line L with arbitrary continuity. The theorem is a generalization of the analogous construction for Bézier patches, due to Farin [16].

Algorithm 4.11 (Continuous Joint) *Consider the B-patch*

$$F(u) = \sum_{|\vec{\imath}|=n} N^n_{\vec{\imath}}(u) P_{\vec{\imath}}$$

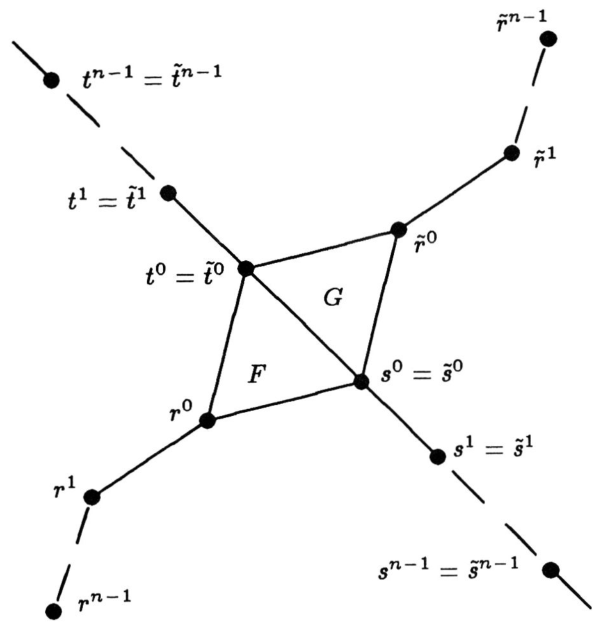

Figure 2: Continuous joint of two B-patches F and G along a line L according to Algorithm 4.11

over

$$\mathcal{K} = (r^0, \ldots, r^{n-1}, s^0, \ldots, s^{n-1}, t^0, \ldots, t^{n-1})$$

and the B-patch

$$G(u) = \sum_{|\vec{\imath}|=n} N_{\vec{\imath}}^n(u) \tilde{P}_{\vec{\imath}}$$

over

$$\tilde{\mathcal{K}} = (\tilde{r}^0, \ldots, \tilde{r}^{n-1}, \tilde{s}^0, \ldots, \tilde{s}^{n-1}, \tilde{t}^0, \ldots, \tilde{t}^{n-1}),$$

where the knots $s^0, \ldots, s^{n-1}, t^0, \ldots, t^{n-1}$ all lie on a line L (cf. Fig. 2). Then F and G are C^q-continuous along L iff for $0 \le i \le q$ the B-patch control points $P_{\vec{\imath}}'$ of G satisfy

$$\tilde{P}_{\vec{\imath}} = p^i_{(0,j,k)}(\underbrace{\tilde{r}^0, \ldots, \tilde{r}^{i-1}}_{i}), \quad 0 \le i \le q,$$

where the points $p^i_{0,j,k}(\underbrace{\tilde{r}^0, \ldots, \tilde{r}^{i-1}}_{i})$ are generated from the control points $P_{\vec{\imath}}$ of F by means of Algorithm 4.7.

Proof: Again we briefly sketch the proof. Suppose that F and G are in fact C^q-continuous along L. According to Proposition 2.2 this implies

$$f(u \ldots, u, \tilde{r}^0, \ldots, \tilde{r}^{i-1}) = g(u \ldots, u, \tilde{r}^0, \ldots, \tilde{r}^{i-1})$$

for $0 \le i \le q$ and arbitrary $u \in L$, and the polynomials

$$F_L(u) = f(u \ldots, u, \tilde{r}^0, \ldots, \tilde{r}^{i-1})$$

and

$$G_L(u) = g(u \ldots, u, \tilde{r}^0, \ldots, \tilde{r}^{i-1})$$

of degree $n-i$ defined on L agree. Hence their polar forms f_L and g_L agree, too, and we get

$$f(s^0, \ldots, s^{j-1}, t^0, \ldots, t^{k-1}, \tilde{r}^0, \ldots, \tilde{r}^{i-1}) = \\ g(s^0, \ldots, s^{j-1}, t^0, \ldots, t^{k-1}, \tilde{r}^0, \ldots, \tilde{r}^{i-1})$$

for $|\vec{\imath}| = n$. Therefore

$$\begin{aligned} \tilde{P}_{\vec{\imath}} &= g(s^0, \ldots, s^{j-1}, t^0, \ldots, t^{k-1}, \tilde{r}^0, \ldots, \tilde{r}^{i-1}) \\ &= f(s^0, \ldots, s^{j-1}, t^0, \ldots, t^{k-1}, \tilde{r}^0, \ldots, \tilde{r}^{i-1}) \\ &= p^i_{0,j,k}(\tilde{r}^0, \ldots, \tilde{r}^{i-1}) \end{aligned}$$

and the assertion follows.

Conversely, suppose that $\tilde{P}_{\vec{\imath}} = p^i_{0,j,k}(\tilde{r}^0, \ldots, \tilde{r}^{i-1})$ for $0 \le i \le q$. Then the $(n-i)$-polar forms

$$f_L(u^1, \ldots, u^{n-i}) := f(u^1, \ldots, u^{n-i}, \tilde{r}^0, \ldots, \tilde{r}^{i-1})$$

and

$$g_L(u^1, \ldots, u^{n-i}) := g(u^1, \ldots, u^{n-i}, \tilde{r}^0, \ldots, \tilde{r}^{i-1})$$

defined on L have the same poles $P^*_{j,k} = p^i_{0,j,k}(\tilde{r}^0, \ldots, \tilde{r}^{i-1})$, $|\vec{\imath}| = n$, and are therefore identical. A similar argument as above then completes the proof. ♣

Note that in the case of n-fold knots $r^0 = \ldots = r^{n-1} = r$, $s^0 = \ldots = s^{n-1} = s$, and $t^0 = \ldots = t^{n-1} = t$, the above algorithm specializes to Farin's construction for joining two Bézier patches using the de Casteljau algorithm.

We conclude this section with an algorithm for exchanging the knots in the knot net $\mathcal{K}$ of a B-patch. The algorithm is similar to the insertion algorithm for B-splines [3, 32]:

Algorithm 4.12 (Knot Insertion) *Consider a B-patch*

$$F(u) = \sum_{|\vec{\imath}|=n} N_{\vec{\imath}}^n(u) P_{\vec{\imath}} \tag{17}$$

of degree n over

$$\mathcal{K} = (r^0, \ldots, r^{n-1}, s^0, \ldots, s^{n-1}, t^0, \ldots, t^{n-1}) \tag{18}$$

and suppose that the knot net

$$\begin{aligned} \mathcal{K}^* = \; &(r^0, \ldots, r^l, r, r^{l+1}, \ldots, r^{n-2}, \\ &s^0, \ldots, s^{n-1}, t^0, \ldots, t^{n-1}) \end{aligned} \tag{19}$$

is obtained from $\mathcal{K}$ by inserting a new knot r between r^l and r^{l+1} for some $-1 \leq l \leq n-2$, and dropping r^{n-1}. Then F has a unique representation

$$F(u) = \sum_{|\vec{i}|=n} N_{\vec{i}}^n(u) P_{\vec{i}}^* \tag{20}$$

as B-patch of degree n over $\mathcal{K}^*$, and the new control points $P_{\vec{i}}^*$ are given as

$$P_{\vec{i}}^* = P_{\vec{i}}, \tag{21}$$

if $0 \leq i \leq l+1$, and

$$\begin{aligned} P_{\vec{i}} &= \rho_{i-1}(r) P_{\vec{i}} + \sigma_j(r) P_{\vec{i}-\vec{e}_1+\vec{e}_2} \\ &\quad + \tau_k(r) P_{\vec{i}-\vec{e_1}+\vec{e_3}} \end{aligned} \tag{22}$$

if $l+2 \leq i \leq n$. Here $\rho_{i-1}(r), \sigma_j(r)$ and $\tau_k(r)$ denote the barycentric coordinates of

$$r = \rho_{i-1}(r) r^{i-1} + \sigma_j(r) s^j + \tau_k(r) t^k$$

w.r.t. $\triangle(r^{i-1}, s^j, t^k)$. ♣

Multiple application of Algorithm 4.12 allows to subdivide a B-patch into several pieces. The refined control net converges to the surface and can be used as a piecewise linear approximation to the surface. The subdivision process can be carried out adaptively, and the level of subdivision that is necessary to approximate the surface within a given tolerance ϵ can be precomputed from estimates on the second derivatives, based on Algorithm 4.9. By precomputing the necessary level of subdivision, the time-consuming flatness testing at every level of the algorithm can be completely avoided.

Conclusion

We have presented a new representation for bivariate polynomials, the B-patch, and discussed some of its main properties of interest in the construction of smooth surfaces in CAGD. It has been shown that B-patches subsume triangular Bézier patches and that many important properties of B-splines carry over to B-patches almost word by word. B-patches are the main building block in the new multivariate B-spline surfaces recently developed in [8]. These new spline surfaces allow to construct automatically smooth surfaces over arbitrary triangulations of the parameter plane. A test-implementation for these new B-spline surfaces that partly uses some of the algorithms given in this paper is currently under way at the University of Waterloo.

References

[1] P.J. Barry. *Urn Models, Recursive Curve Schemes, and Computer Aided Geometric Design.* PhD thesis, University of Utah, Salt Lake City, 1987.

[2] R.H. Bartels, J.C. Beatty, and B.A. Barsky. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling.* Morgan Kaufmann Publishers, 1987.

[3] W. Boehm. Inserting new knots into a B-spline curve. *Comput. Aided Design*, 12:50–62, 1980.

[4] W. Boehm. The de boor algorithm for triangular splines. In R.E. Barnhill and W. Boehm, editors, *Surfaces in CAGD*, pages 109–120. North-Holland, 1983.

[5] W. Boehm, G. Farin, and J. Kahmann. A survey of curve and surface methods in CAGD. *Computer-Aided Geom. Design*, 1:1–60, 1984.

[6] E. Cohen, T. Lyche, and R.F. Riesenfeld. Discrete B-splines and subdivision techniques in computer aided geometric design and computer graphics. *Computer Graphics and Image Processing*, 14:87–111, 1980.

[7] W. Dahmen and C.A. Micchelli. Recent progress in multivariate splines. In C.K. Chui, L.L. Schumaker, and J.D. Ward, editors, *Approximation Theory IV*, pages 27–121. Academic Press, 1983.

[8] W. Dahmen, C.A. Micchelli, and H.-P. Seidel. Blossoming begets B-Splines built better by B-Patches. Research Report RC 16261(#72182), IBM Research Division, Yorktown Heights, 1990.

[9] C. de Boor. On calculating with B-splines. *Journal of Approximation Theory*, 6:50–62, 1972.

[10] C. de Boor. *A Practical Guide to Splines.* Springer, New York, 1978.

[11] C. de Boor. B-form basics. In G. Farin, editor, *Geometric Modeling, Algorithms and New Trends*, pages 131–148. SIAM, 1987.

[12] P. de Casteljau. Courbes et surfaces à pôles. Technical report, Andre Citroen, Paris, 1963.

[13] P. de Casteljau. *Formes à Pôles.* Hermes, Paris, 1985.

[14] T. DeRose. *Geometric Continuity: A Parametrization Independent Measure of Continuity for Computer Aided Geometric Design.* PhD thesis, UC Berkeley, 1985.

[15] T. DeRose and T. Hollmann. The triangle: a multiprocessor architecture for fast curve and surface generation. Technical Report 87-08-07, Computer Science Department, Univ. of Washington, Seattle, 1987.

[16] G.E. Farin. *Subsplines über Dreiecken.* PhD thesis, TU Braunschweig, Braunschweig, F.R.G., 1979.

[17] G.E. Farin. Triangular Bernstein-Bézier patches. *Computer-Aided Geom. Design*, 3:83–127, 1986.

[18] G.E. Farin. *Curves and Surfaces for Computer Aided Geometric Design.* Academic Press, 1988.

[19] R.N. Goldman. Blossoming and knot insertion algorithms for B-spline curves. *Computer-Aided Geom. Design*, 7:69–81, 1990.

[20] W.J. Gordon and R.F. Riesenfeld. B-spline curves. In R.E. Barnhill and R.F. Riesenfeld, editors, *Computer Aided Geometric Design.* Academic Press, 1974.

[21] J. Hoschek and D. Lasser. *Grundlagen der geometrischen Datenverarbeitung.* Teubner, 1989.

[22] C. Loop and T.D. DeRose. A multisided generalization of Bézier surfaces. *ACM Trans. Graph.*, 8(3):204–234, 1989.

[23] C. Loop and T.D. DeRose. Generalized b-spline surfaces of arbitrary topology. In *Proc. SIGGRAPH'90*, pages 347–356. ACM SIGGRAPH, 1990.

[24] J. Peters. *Fitting smooth parametric surfaces to 3D data.* PhD thesis, University of Wisconsin, Madison, 1990.

[25] H. Prautzsch. *Unterteilungsalgorithmen für multivariate Splines.* PhD thesis, TU Braunschweig, Braunschweig, F.R.G., 1984.

[26] L. Ramshaw. Blossoming: A connect-the-dots approach to splines. Technical report, Digital Systems Research Center, Palo Alto, 1987.

[27] L. Ramshaw. Béziers and B-splines as multiaffine maps. In *Theoretical Foundations of Computer Graphics and CAD*, pages 757–776. Springer, 1988.

[28] L. Ramshaw. Blossoms are polar forms. *Computer-Aided Geom. Design*, 6:323–358, 1989.

[29] R.F. Riesenfeld. *Applications of B-spline Approximation to Geometric Problems of Computer Aided Design.* PhD thesis, Syracuse University, 1973.

[30] M.A. Sabin. *The Use of Piecewise Forms for the Numerical Representation of Shape.* PhD thesis, Hungarian Academy of Sciences, Budapest, Hungary, 1976.

[31] L.L. Schumaker. *Spline Functions: Basic Theory.* John Wiley & Sons, New York, 1981.

[32] H.-P. Seidel. A new multiaffine approach to B-splines. *Computer-Aided Geom. Design*, 6:23–32, 1989.

[33] H.-P. Seidel. *Polynome, Splines und symmetrische rekursive Algorithmen im Computer Aided Geometric Design.* Habilitationsschrift, Universität Tübingen, 1989.

[34] H.-P. Seidel. Symmetric recursive algorithms for surfaces: B-patches and the de boor algorithm for polynomials over triangles. *Constructive Approximation*, 1991. Accepted for publication.

[35] P.B. Slusallek. Flächenmodellierung mit splines über dreiecken. Master's thesis, University of Tübingen, Tübingen, F.R.G., 1989.

[36] P.B. Slusallek and H.-P. Seidel. TRIMO - a workstation-based interactive system for the generation, manipulation, and display of surfaces over arbitrary topological meshes. In *Proc. Eurographics'90*, pages 147–158. North Holland, 1990.

[37] E. Stärk. *Mehrfach differenzierbare Bézier - Kurven und Bézier -Flächen.* PhD thesis, TU Braunschweig, Braunschweig, F.R.G., 1976.

[38] W. Straßer and H.-P. Seidel, editors. *Theory and Practice of Geometric Modeling.* Springer, 1989.