An Implementation of Multivariate B-Spline Surfaces over Arbitrary Triangulations

Philip Fong and Hans-Peter Seidel *

Computer Graphics Laboratory,
Department of Computer Science,
University of Waterloo,
Waterloo, Ontario, Canada N2L 3G1

Abstract

Recently in [7], a new multivariate B-spline scheme based on blending functions and control vertices was developed. This surface scheme allows C^{k-1}-continuous piecewise polynomial surfaces of degree k over arbitrary triangulations to be modelled. Actually, piecewise polynomial surfaces over a *refined* triangulation are produced given an arbitrary triangulation. The scheme exhibits both affine invariance and the convex hull property, and the control points can be used to manipulate the shape of the surface locally. This paper describes a test implementation of the scheme for quadratic and cubic surfaces. Issues such as evaluating points on the surface, evaluating derivatives on the surface and representing piecewise polynomial surfaces as linear combinations of B-splines will be discussed. Several examples illustrate the implementation. The work is incorporated into a surface editor which is currently being developed at the University of Waterloo.

Keywords: Blossoming, B-patch, B-spline surface, blending functions, control points, simplex splines, polar forms.

1 Introduction

Tensor-product B-spline surfaces [1, 2, 8, 9, 14, 24] have proven themselves an excellent tool for the modelling of free form surfaces. However, tensor-product surfaces also have their well-known draw-backs if the modelling of largely irregular objects is required. Therefore, not surprisingly, the need for B-splines over non-rectangular regions has been expressed quite early [25].

Splines over arbitrary triangulations of the parameter plane have first been considered in [5, 17]. These multivariate splines are defined as projections of simplices and are therefore called simplex splines. The main drawback of simplex splines in the past has been the difficulty to form linear combinations and the absence of control points.

A different approach has been taken in [30]. The B-patches developed there are based on the study of symmetric recursive evaluation algorithms and are defined by generalizing the de Boor algorithm for the evaluation of a B-spline segment from curves to surfaces. B-patches have control points but the construction of smooth surfaces still requires considerable computation.

Other approaches to the construction of B-splines over irregular domains have been based on subdivision [3, 11], interpolation [21], and on the use of multisided patches [18, 19, 26]. However, each of these schemes has its own difficulties.

One really needs a scheme which constructs automatically smooth complex surfaces and which contains control vertices for shape manipulation. A new multivariate B-spline scheme based on a combination of B-patches and simplex splines which meets these criteria was developed in [7]. This paper discusses details of an implementation of it which is being used in a surface editor being developed at the University of Waterloo. A process of converting piecewise Bézier polynomials to this new scheme and vice-versa will be explained. This leads to a method for surface refinement.

The paper is divided up in the following way. Section 2 introduces some notation which will be used in the remainder of the paper. Section 3 describes the new B-spline scheme. Section 4 discusses the implementation while Section 5 illustrates the new B-spline scheme through examples. We finally finish off with some concluding remarks.

2 Notation and Definitions

This section introduces some notation which is used in the rest of the paper.

Let $W = \{\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2\} \subset \mathbb{R}^2$ be a set of affinely independent points and let $\mathbf{u} \in \mathbb{R}^2$. If the determinant

*This work has been partly supported by the Natural Sciences and Engineering Research Council of Canada through Personal Operating Grant OGP0105573 and Strategic Operating Grant STR0040527

$d(W)$ is defined as

$$d(W) = \det \begin{pmatrix} 1 & 1 & 1 \\ \mathbf{w}_0 & \mathbf{w}_1 & \mathbf{w}_2 \end{pmatrix}$$

and the determinant $d_j(\mathbf{u}|W)$ as $d(W)$ with the point $\mathbf{w}_j$ being replaced by $\mathbf{u}$, then the *barycentric coordinates* of $\mathbf{u}$ with respect to the ordered set W are given as

$$\lambda_j(\mathbf{u}) = \frac{d_j(\mathbf{u}|W)}{d(W)}, \quad j = 0 \ldots 2 \tag{1}$$

Note that

$$\mathbf{u} = \sum_{j=0}^{2} \lambda_j(\mathbf{u})\mathbf{w}_j \quad \text{and} \quad \sum_{j=0}^{2} \lambda_j(\mathbf{u}) = 1.$$

For a set $V = \{\mathbf{v}_0, \ldots, \mathbf{v}_n\} \subset \mathbf{R}^2$, we let $[V]$ denote the convex hull of V and we let $[V)$ denote the half-open convex hull of V. The definition of the half-open convex hull is given in [29] and is repeated here.

Definition 2.1 (Half-Open Convex Hull) *Given $\mathbf{v}_0, \ldots, \mathbf{v}_n \in \mathbb{R}^2$, the half-open convex hull is then defined as follows: Let ξ be the unit horizontal vector in $\mathbb{R}^2$. A point $\mathbf{u} \in \mathbb{R}^2$ belongs to the half-open convex hull $[\mathbf{v}_0, \ldots, \mathbf{v}_n)$ if and only if there exists a vector η with positive slope and a positive scalar ϵ such that the set $\{s\xi + t\eta \mid 0 < s, t, \ 0 < s + t < \epsilon\}$ is completely contained in the interior of $[\mathbf{v}_0, \ldots, \mathbf{v}_n]$.*

3 Bivariate B-Splines

The new B-Spline scheme is obtained by *matching* B-patches [28, 30] with simplex splines [5, 17]. By matching we mean that the recurrence relation which describes simplex splines is made to agree with the recurrence relation for B-patches under some conditions. Before we present this, we need some background information on simplex splines and B-patches.

3.1 Simplex Splines

Definition 3.1 (Simplex Splines)
Let $V = \{\mathbf{t}_0, \ldots, \mathbf{t}_m\}$ be a finite set of points in $\mathbb{R}^2$ and let $\mathbf{u}$ be a point in $\mathbb{R}^2$. The simplex spline $M(\mathbf{u}|V) = M(\mathbf{u}|\mathbf{t}_0, \ldots, \mathbf{t}_m)$ is defined recursively as follows. For $V = \{\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2\}$ we let

$$M(\mathbf{u}|\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2) = \frac{\chi_{[\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2)}(\mathbf{u})}{|d(\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2)|}, \tag{2}$$

where

$$\chi_{[\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2)}(\mathbf{u}) = \begin{cases} 1 & \text{if } \mathbf{u} \in [\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

is the characteristic function on $[\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2)$. For $V = \{\mathbf{t}_0, \ldots, \mathbf{t}_m\}$, $m > 2$, we set

$$M(\mathbf{u}|V) = \sum_{j=0}^{2} \frac{d_j(\mathbf{u}|W)}{d(W)} M(\mathbf{u}|V \backslash \{\mathbf{t}_{i_j}\}) \tag{4}$$

where $W = \{\mathbf{t}_{i_0}, \mathbf{t}_{i_1}, \mathbf{t}_{i_2}\}$ is any subset of affinely independent points in V [20]. The points $\mathbf{t}_0, \ldots, \mathbf{t}_m$ are referred to as knots.

It is worth mentioning that the above definition is completely independent of the choice of W [20].

Equations (2) and (4) differ slightly from the ones given in [17, 20] in that we have based the characteristic function on the half-open convex hull $[\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2)$ (Definition 2.1) instead of the convex hull $[\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2]$. Otherwise, problems arise when the recurrence relation is used for points $\mathbf{u}$ which lie along knot lines (lines connecting any two knots) [20]. The above definition alleviates the problem by modifying the area of support for the B-splines. This is analogous to the case for univariate B-splines [27] where they are non-zero on the half-open interval $[\mathbf{t}_0, \mathbf{t}_1)$ instead of the closed interval $[\mathbf{t}_0, \mathbf{t}_1]$.

The simplex splines $M(\mathbf{u}|V)$ then exhibit the following properties:

- Piecewise polynomial of degree $k = m - 2$
- Local support on the closed convex hull $[V]$
- Non-negative - $M(\mathbf{u}|V) \geq 0$ for all $\mathbf{u} \in \mathbf{R}^2$
- C^{k-1}-continuous everywhere

Further information can be obtained from [5, 6, 15, 16, 17, 20, 32].

Definition 3.1 shows us that plenty of simplex splines exist. The question which remains is how to form linear combinations from them such that piecewise polynomial surfaces over arbitrary triangulations can be constructed. This involves choosing the right simplex splines and the right normalization. These problems may be solved by studying B-patches.

3.2 B-patches

B-patches [30] are a patch representation for polynomial surfaces that arises from generalizing the de Boor algorithm from curves to surfaces [28, 30]. One definition of B-patches is by means of their blending functions $B^I_\beta(\mathbf{u})$.

Definition 3.2 (B-patch Blending Functions)
Let $\triangle(I) = [\mathbf{t}_{i_0}, \mathbf{t}_{i_1}, \mathbf{t}_{i_2}] \in \mathbb{R}^2$, $I = (i_0, i_1, i_2)$ be given along with the additional set of knots $\mathbf{t}_{i_0,0}, \ldots, \mathbf{t}_{i_0,k-1}, \mathbf{t}_{i_1,0}, \ldots, \mathbf{t}_{i_1,k-1}, \mathbf{t}_{i_2,0}, \ldots, \mathbf{t}_{i_2,k-1}$ in $\mathbb{R}^2$ such that $\mathbf{t}_{i_0,0} = \mathbf{t}_{i_0}$, $\mathbf{t}_{i_1,0} = \mathbf{t}_{i_1}$, and $\mathbf{t}_{i_2,0} = \mathbf{t}_{i_2}$. Also, assume that every triple set of knots $(\mathbf{t}_{i_0,\beta_0}, \mathbf{t}_{i_1,\beta_1}, \mathbf{t}_{i_2,\beta_2})$, $0 \leq \beta_0 + \beta_1 + \beta_2 \leq k - 1$ is affinely independent, i.e. $[\mathbf{t}_{i_0,\beta_1}, \mathbf{t}_{i_1,\beta_1}, \mathbf{t}_{i_2,\beta_2}]$ forms a proper triangle. Then, for $\mathbf{u} \in \mathbb{R}^2$, the B-patch blending functions $B^I_\beta(\mathbf{u})$, $|\beta| = k$, of degree k over $\triangle(I)$ are given by the recurrence

$$B^I_{(0,0,0)}(\mathbf{u}) = 1, \tag{5}$$

and

$$B^I_\beta(\mathbf{u}) = \sum_{j=0}^{2} \lambda^I_{\beta-e^j,j}(\mathbf{u}) B^I_{\beta-e^j}(\mathbf{u}), \quad |\beta| > 0. \qquad (6)$$

Terms with negative indices are set to zero and $\lambda^I_{\beta,j}(\mathbf{u}) = d_j(\mathbf{u}|W^I_\beta)/d(W^I_\beta)$ *are the barycentric coordinates of* $\mathbf{u}$ *with respect to* $W^I_\beta = \{\mathbf{t}_{i_0,\beta_0}, \mathbf{t}_{i_1,\beta_1}, \mathbf{t}_{i_2,\beta_2}\}$. *Here* $e^0 = (1,0,0)$, $e^1 = (0,1,0)$, *and* $e^2 = (0,0,1)$.

The B-patch blending functions form a partition of unity [30], i.e. $\sum_{|\beta|=k} B^I_\beta(\mathbf{u}) = 1$. Every polynomial surface F can be represented as a linear combination of them as follows:

$$F(\mathbf{u}) = \sum_{|\beta|=k} \mathbf{c}^I_\beta B^I_\beta(\mathbf{u}), \quad \mathbf{c}^I_\beta \in \mathbf{R}^3 \qquad (7)$$

where

$$\mathbf{c}^I_\beta = f(\mathbf{t}_{i_0,0}, \ldots, \mathbf{t}_{i_0,\beta_0}, \mathbf{t}_{i_1,0}, \ldots, \mathbf{t}_{i_1,\beta_1}, \mathbf{t}_{i_2,0}, \ldots \mathbf{t}_{i_2,\beta_2}) \qquad (8)$$

are the B-patch control points which form the B-patch control net. Here f represents the *blossom* or *polar form* of F [10, 22, 23]. The representation given by (7) is called the *B-patch.*

The shape of a B-patch is strongly influenced by the shape of its control net. We can form larger surfaces by piecing together individual B-patches. However, the construction of overall smooth surfaces still requires quite a bit of computation. What is needed are blending functions which produce smooth piecewise polynomial surfaces automatically. Simplex splines, which were introduced in Section 3.1, give us just that and by combining them with B-patches, we are led to the new B-spline scheme.

3.3 The New B-Spline Scheme

The development of the B-spline scheme in [7] is based upon the fact that the recurrence relations (2),(4) and (5),(6) agree under the proper renormalization and the proper selection of knots. We now briefly describe its construction.

Let $T = \{\triangle(I) = [\mathbf{t}_{i_0}, \mathbf{t}_{i_1}, \mathbf{t}_{i_2}] \mid I = (i_0, i_1, i_2) \in \mathcal{I} \subseteq \mathbf{Z}^3_+\}$ define a triangulation of $\mathbf{R}^2$ or some bounded domain $D \subset \mathbf{R}^2$. Then, for any two $I, J \in \mathcal{I}$, $\triangle(I) \cap \triangle(J)$ is empty or is a common vertex or edge of $\triangle(I)$ and $\triangle(J)$ (see Fig. 1).

Next, a sequence of knots $\mathbf{t}_{i,0}, \ldots, \mathbf{t}_{i,k}$ is assigned to each vertex $\mathbf{t}_i$ in the triangulation such that $\mathbf{t}_{i,0} = \mathbf{t}_i$ and that any set of three knots is affinely independent. The sequence of knots $\mathbf{t}_{i,0}, \ldots, \mathbf{t}_{i,k}$ is referred to as the cloud of knots associated with the vertex $\mathbf{t}_i$. We are now in a situation to construct simplex splines of degree k over the triangulation T. We consider the following simplex splines:

$$M(\mathbf{u}|V) = M(\mathbf{u}|V^I_\beta) \qquad (9)$$

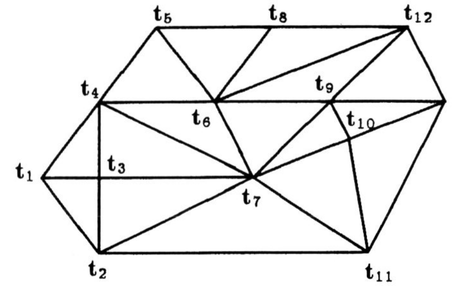


Figure 1: Triangulation of a bounded domain $D \subset \mathbb{R}^2$

where $I \in \mathcal{I}$, $|\beta| = k$, and

$$V^I_\beta = \{\mathbf{t}_{i_0,0}, \ldots, \mathbf{t}_{i_0,\beta_0}, \mathbf{t}_{i_1,0}, \ldots, \mathbf{t}_{i_1,\beta_1}, \mathbf{t}_{i_2,0}, \ldots, \mathbf{t}_{i_2,\beta_2}\}. \qquad (10)$$

We define the regions Ω^I_k as follows:

$$\Omega^I_\beta := \cap_{\gamma \le \beta}[W^I_\gamma], \quad \Omega^I_k := \mathrm{int}(\cap_{|\beta|=k}\Omega^I_\beta). \qquad (11)$$

We also assume that $\Omega^I_k \neq 0$ which can be obtained if each of the clouds of knots associated with the three vertices of a triangle is kept separate from one another. In other words, for each vertex $\mathbf{t}_i$ in the triangulation T, its cloud of knots is contained within a circle C_i centred at $\mathbf{t}_i$ such that $C_i \cap C_j = 0$, for all $i \neq j$ (i.e., none of the circles intersect one another). Figure 2 below illustrates an example of this setup.

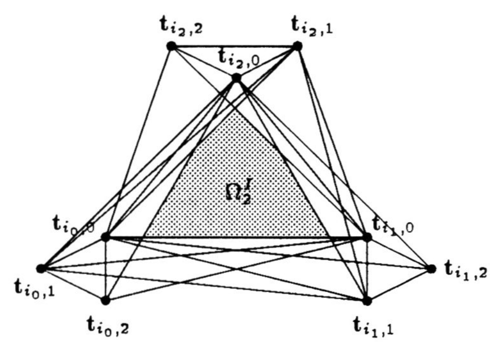


Figure 2: The region Ω^I_2

Then, under these conditions, it is shown in [7] that

$$B^I_\beta(\mathbf{u}) = |d(W^I_\beta)| M(\mathbf{u}|V^I_\beta), \text{ for all } \mathbf{u} \in \Omega^I_k \qquad (12)$$

where $|\beta| = k$, V^I_β is defined in (10) and $W^I_\beta = \{\mathbf{t}_{i_0,\beta_0}, \mathbf{t}_{i_1,\beta_1}, \mathbf{t}_{i_2,\beta_2}\}$. From (12), we let the *normalized B-splines* be defined as

$$N^I_\beta(\mathbf{u}) := |d(W^I_\beta)| M(\mathbf{u}|V^I_\beta). \qquad (13)$$

These will be the blending functions used in the new B-spline scheme. A B-spline surface F of degree k over a given triangulation T with a knot net $\mathcal{K} = \{\mathbf{t}_{i,l} \mid i \in \mathbf{Z}, l = 0, \ldots, k\}$ can then be defined as

$$F(\mathbf{u}) = \sum_{I \in \mathcal{I}} \sum_{|\beta|=k} \mathbf{c}_{I,\beta} N_\beta^I(\mathbf{u}). \tag{14}$$

The $\mathbf{c}_{I,\beta} \in \mathbf{R}^3$ are the control points which make up the control net for the surface F.

Since both simplex splines and B-patches are used to develop the new scheme, their individual properties are inherited by the new scheme. It is these properties which make it possible (relatively easily) to model C^{k-1}-continuous piecewise polynomial surfaces of degree k over arbitrary triangulations.

Affine Invariance: The relationship between the control points and the B-spline surface is invariant under affine coordinate transformations. That is, if $\Phi : \mathbf{R}^3 \to \mathbf{R}^3$ is an affine map (rotation, translation, scaling), then

$$\Phi\Big(\sum_{I \in \mathcal{I}} \sum_{|\beta|=k} \mathbf{c}_{I,\beta} N_\beta^I(\mathbf{u})\Big) = \sum_{I \in \mathcal{I}} \sum_{|\beta|=k} \Phi(\mathbf{c}_{I,\beta}) N_\beta^I(\mathbf{u}). \tag{15}$$

Convex Hull Property: A B-spline surface lies in the convex hull of its control points.

Local Support: Movement of the control point $\mathbf{c}_\beta^I$ only influences the region of the surface on $\triangle(I)$ and those surrounding it.

Continuity A degree k B-spline surface is a piecewise polynomial of degree k over the sub-triangulation induced by its knot net that is C^{k-1}-continuous everywhere if its knots are in general position. But, from the theory of simplex splines, knot multiplicities along a line reduce the order of continuity along this line [20]. For example, a degree 2 surface with knots in general position is C^1-continuous everywhere. Placing three knots on a line reduces the continuity to C^0 and placing four knots produces a discontinuity along the line. Thus, the underlying knot net provides additional degrees of freedom to control the shape of the surface. Figure 3 shows the quadratic normalized B-splines over different knot configurations.

4 Implementation

The theory presented in the previous sections is used in a surface editor which is being developed at the University of Waterloo. A surface editor allows one to manipulate the shape of a surface through the movement of the control vertices which make up the control net. The new B-spline scheme also allows surface changes to be made through movement of knots. This section describes some of the algorithms used in the editor.

4.1 Evaluation

The most important algorithm required is one which evaluates points on the surface. That is, given a parameter value $\mathbf{u} \in \mathbf{R}^2$ in the triangulation T, we want the value of the point on the surface corresponding to $\mathbf{u}$. We use equation (14) as the basic formula in our algorithm. Since the normalized B-splines $N_\beta^I(\mathbf{u})$ are the most complex terms in the equation, we will only concentrate on them.

In order to evaluate the normalized B-splines $N_\beta^I(\mathbf{u})$ defined in (13), we first need to compute the simplex splines $M(\mathbf{u}|V_\beta^I)$ defined by the recurrence (2), (4). We start off by describing the evaluation of linear simplex splines because all higher order splines are composed of these.

Let $V_\beta^I = \{\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3\} \subset \mathbf{R}^2$ and without loss of generality, let the set $W = \{\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2\}$. Then, after expansion of the recurrence and substitution of the base case (2), the linear simplex spline becomes

$$\begin{aligned} M(\mathbf{u}|V_\beta^I) \;=\; & \frac{d_0(\mathbf{u}|W)}{d(W)} \frac{\chi_{[\mathbf{t}_1,\mathbf{t}_2,\mathbf{t}_3)}(\mathbf{u})}{|d(\mathbf{t}_1,\mathbf{t}_2,\mathbf{t}_3)|} + \\ & \frac{d_1(\mathbf{u}|W)}{d(W)} \frac{\chi_{[\mathbf{t}_0,\mathbf{t}_2,\mathbf{t}_3)}(\mathbf{u})}{|d(\mathbf{t}_0,\mathbf{t}_2,\mathbf{t}_3)|} + \\ & \frac{d_2(\mathbf{u}|W)}{d(W)} \frac{\chi_{[\mathbf{t}_0,\mathbf{t}_1,\mathbf{t}_3)}(\mathbf{u})}{|d(\mathbf{t}_0,\mathbf{t}_1,\mathbf{t}_3)|}. \end{aligned} \tag{16}$$

One can evaluate (16) by blindly computing and plugging in values for each of the terms. However, depending on the value of the characteristic function χ (3), some of the terms may be zero. This can lead to a very inefficient evaluation technique. A better method is to compute only those terms for which the characteristic function is non-zero. This is, in itself, governed by the choice of W. To do this, we need to know where in the knot configuration the point $\mathbf{u}$ lies. This involves looking at the various knot configurations for a linear spline (see Fig. 4). We first point out that

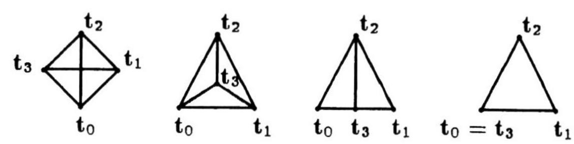


Figure 4: The four essentially different knot configurations for a linear B-spline $M(\mathbf{u}|\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3)$.

each of the configurations is composed of the four triangles (although some may be degenerate): $\triangle[\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2]$, $\triangle[\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_3]$, $\triangle[\mathbf{t}_0, \mathbf{t}_2, \mathbf{t}_3]$, $\triangle[\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3]$. Each of these triangles appears in at most one of the characteristic functions in (16) or in the set W. For every single one of the configurations above, if a point $\mathbf{u}$ is inside its half-open convex hull, then $\mathbf{u}$ belongs to exactly two of its triangles. For instance, in the leftmost configuration above, if $\mathbf{u} \in [\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2)$, then either $\mathbf{u} \in [\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_3)$ or $\mathbf{u} \in [\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3)$ but not both. If $\mathbf{u}$ lies on the boundary

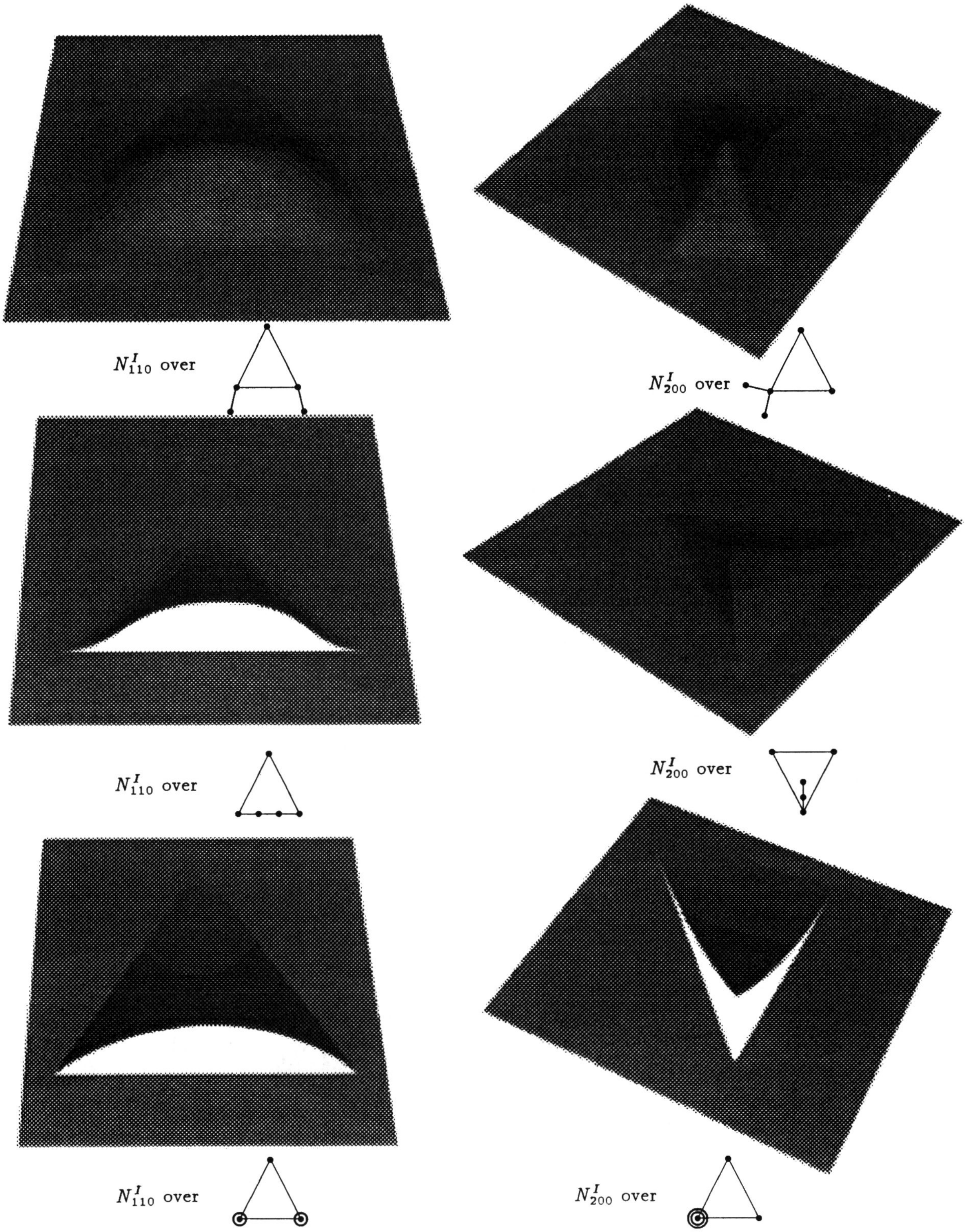


Figure 3: The quadratic normalized B-splines $N^I_{110}(\mathbf{u})$ and $N^I_{200}(\mathbf{u})$ over the six different knot configurations. ⊚ is a double knot and ◎ is a triple knot.

between two triangles or on the vertex of two or more triangles, then Definition 2.1 for the half-open convex hull ensures that $\mathbf{u}$ only belongs to one of them. It is this, which allow points on knot lines to be evaluated correctly [16, 20].

Using these facts then, we let W be one of the triangles in which $\mathbf{u}$ belongs. In light of the above discussion, this will force two of the three characteristic functions to be zero, and hence leaving only one term in (16) to be evaluated.

The only real work involved then is to figure out which triangles the point $\mathbf{u}$ belong. One way is to calculate the barycentric coordinates of $\mathbf{u}$ with respect to the vertices of the triangle. If the coordinates are all greater or equal to zero, then $\mathbf{u}$ is inside, otherwise it's outside. Of course, a slight modification has to be made since we are dealing with the half-open convex hull instead of the closed convex hull.

In the implementation of the surface editor, the computation is organized such that intermediate results obtained from determining the regions containing $\mathbf{u}$ are later re-used in the evaluation of the linear B-spline. In this way, only a small number of determinants need to be explicitly computed while others can be derived from some linear combination of these.

Higher order simplex splines ($m \geq 4$) are simply computed using the recurrence relations (2),(4). We do not try to *optimize* the computation like we did in the linear case above because it is not worthwhile due to the increase in the number of knot configurations. At each level of the recurrence, any choice is suitable for the set $W \subset V_\beta^I$ as long as it forms a proper triangle. However, a good choice is to pick W such that $\mathbf{u} \in [W]$ which gives positive barycentric coordinates. This eliminates any negative terms and hence, increases the numerical stability of the evaluation [16].

Having computed the value for the simplex spline, we can get the value for the normalized B-spline from (13) and finally evaluate the point on the surface from (14).

4.2 Derivatives

A directional derivative along a given direction $\mathbf{v} \in \mathbf{R}^2$ for a parameter value $\mathbf{u} \in \mathbf{R}^2$ may be computed in the same manner as in its evaluation. The only difference lies in the fact that the barycentric coordinates of a vector $\mathbf{v}$ add up to zero instead of one, i.e.

$$\sum_{j=0}^{2} \mu_j(\mathbf{v}) = 0 \text{ and } \mathbf{v} = \sum_{j=0}^{2} \mu_j(\mathbf{v})\mathbf{t}_j. \tag{17}$$

The directional derivatives for degree k simplex splines is then given as

$$\mathcal{D}_\mathbf{v} M(\mathbf{u}|V) = k \sum_{j=0}^{2} \mu_j(\mathbf{v}) M(\mathbf{u}|V \backslash \{t_{i_j}\}) \tag{18}$$

with V as defined in Definition 3.1. Then the directional derivative along the direction $\mathbf{v}$ at a parameter value $\mathbf{u}$ for a surface F is given by

$$\mathcal{D}_\mathbf{v} F(\mathbf{u}) = \sum_{I \in \mathcal{I}} \sum_{|\beta|=k} \mathbf{c}_\beta^I \mathcal{D}_\mathbf{v} N_\beta^I(\mathbf{u}) \tag{19}$$

with

$$\mathcal{D}_\mathbf{v} N_\beta^I(\mathbf{u}) = |d(W_\beta^I)| \mathcal{D}_\mathbf{v} M(\mathbf{u}|V). \tag{20}$$

We can then use (19) to calculate the tangent normal for a point on the surface in the following way. Any two directional derivatives $\mathbf{v}_1, \mathbf{v}_2 \in \mathbf{R}^2$ ($\mathbf{v}_1 \neq a\mathbf{v}_2, a \in \mathbf{R}$) are computed at the point and the resulting cross-product $\mathcal{D}_{\mathbf{v}_1} F(\mathbf{u}) \times \mathcal{D}_{\mathbf{v}_2} F(\mathbf{u})$ will yield the desired normal vector. The surface editor makes use of these directional derivatives in its surface shading (Gouraud) routines.

4.3 Piecewise Polynomial Surfaces

For a surface scheme to be as flexible as possible, it must be able to represent as many surfaces as possible. This section shows that any piecewise polynomial surface F over a triangulation T can be represented as a linear combination of normalized B-splines $N_\beta^I(\mathbf{u})$. It also shows that B-splines can be represented as piecewise Bézier surfaces.

The precise statement for the representation of piecewise polynomials as linear combinations of B-splines is as follows [31]

Theorem 4.1 *Let F be any piecewise polynomial surface of degree k over a given triangulation T that is C^{k-1}-continuous everywhere and let f_I be the polar form of the restriction of F to the triangle $\triangle(I)$, $I \in \mathcal{I}$. Then*

$$F(\mathbf{u}) = \sum_{I \in \mathcal{I}} \sum_{|\beta|=k} \mathbf{c}_{I,\beta} N_\beta^I(\mathbf{u}) \tag{21}$$

with

$$\mathbf{c}_{I,\beta} = f_I(\mathbf{t}_{i_0,0}, \ldots, \mathbf{t}_{i_0,\beta_0-1}, \ldots, \mathbf{t}_{i_2,0}, \ldots, \mathbf{t}_{i_2,\beta_2-1}). \tag{22}$$

If we let $F \equiv 1$, then its polar form $f \equiv 1$ and from Theorem 4.1, we get $\sum_{I,\beta} N_\beta^I(\mathbf{u}) = 1$ which shows that the normalized B-splines $N_\beta^I(\mathbf{u})$ form a global partition of unity.

Piecewise Bézier surfaces F of degree k over irregular triangulations T can be converted to B-spline surfaces of the same degree by using Theorem 4.1. Briefly, the algorithm is as follows. For each vertex $\mathbf{t}_i$ of a given triangulation T of a bounded region $D \subseteq \mathbf{R}^2$, knots $\mathbf{t}_{i,0}, \ldots, \mathbf{t}_{i,k}$ are assigned (in general position) to it such that $\mathbf{t}_{i,0} = \mathbf{t}_i$. The assignment must follow the conditions in Section 3.3 and in addition, vertices on the boundary of D must have their knots outside of D. Then, polar forms f_I of the restriction of F to every $\triangle(I) \in T$, $I \in \mathcal{I}$, are computed using the multiaffine version of the de Casteljau algorithm [10, 23]. The B-spline control points $\mathbf{c}_\beta^I$ are then obtained by evaluating f_I using (22).

Converting degree k B-spline surfaces F over arbitrary triangulations T to piecewise Bézier surfaces $\tilde{F}$ requires a bit more work. We need to come up with two things: a triangulation and the Bézier control points. We cannot use T as our triangulation, as in previous case, because we now need a finer triangulation due to the additional lines introduced by the knots. A finer triangulation can be derived from the knot net associated with each $\triangle(I) \in T, I \in \mathcal{I}$ (see Fig. 2). Let's consider only the $\triangle[\mathbf{t}_{i_0,0}, \mathbf{t}_{i_1,0}, \mathbf{t}_{i_2,0}]$ in Figure 2 and its interior including all line segments passing through the interior We have, in effect, divided up the triangle into regions using the knot lines. Note that not all of the regions in $\triangle[\mathbf{t}_{i_0,0}, \mathbf{t}_{i_1,0}, \mathbf{t}_{i_2,0}]$ are triangular; so, we must further divide (arbitrarily) these regions up. The resulting construction yields the refined triangulation $\tilde{T}_I$ restricted to $\triangle(I)$. Then, the refined triangulation for the piecewise Bézier surface $\tilde{F}$ is $\tilde{T} = \cup_I \tilde{T}_I, \ \triangle(I) \in T, \ I \in \mathcal{I}$.

Having constructed the triangulation, we next deal with the Bézier control points. For each domain triangle $\triangle(\tilde{I}) \in \tilde{T}$, we associate with it, a Bézier triangle with control vertices $\mathbf{b}_\beta^{\tilde{I}}$. From the theory of Bézier triangles [12], a Bézier surface interpolates the corner vertices (those which lie at the corners of the domain triangles) of its control net. Thus, these *corner* control vertices will precisely lie on the surface F and can be computed by evaluating $F(\tilde{\mathbf{t}}_i)$ for all vertices $\tilde{\mathbf{t}}_i$ in the triangulation $\tilde{T}$. The other control vertices are given by

$$\mathbf{b}_\beta^{\tilde{I}} = f(\underbrace{\tilde{\mathbf{t}}_{i_0,0}, \ldots, \tilde{\mathbf{t}}_{i_0,0}}_{\beta_0 \text{ times}}, \ldots, \underbrace{\tilde{\mathbf{t}}_{i_2,0}, \ldots, \tilde{\mathbf{t}}_{i_2,0}}_{\beta_2 \text{ times}}) \qquad (23)$$

where f is the multiaffine definition of F.

4.4 Refinement

For practical purposes, surface schemes must also allow for refinement or subdivision [4, 13]. The idea is that fine detail may be required for parts of the surface but the existing control points do not allow for the modelling of such detail. Thus, we need to be able to add *extra* control points only to those regions.

Then, for the new B-spline scheme, we need to have a finer triangulation over the areas that need to be refined. A finer triangulation will, in effect, provide us with more control vertices. We use a combination of the conversion algorithms from Section 4.3 to solve the problem.

Suppose we are given a B-spline surface F over an arbitrary triangulation T and we want to refine the surface region F_I that's restricted to $\triangle(I) \in T$. First, F_I is converted into a piecewise Bézier surface $\tilde{F}_I$ using the latter algorithm in Section 4.3. Then, the resulting Bézier surface $\tilde{F}_I$ can be converted back into B-spline representation using Theorem 4.1. Hence, a finer triangulation with control vertices over $\triangle(I)$ has been produced. This technique also lends itself to recursive refinement (i.e. refined areas may be further refined, etc).

5 Examples

This section illustrates examples of quadratic surfaces produced from our test implementation. The creation of the triangulations and the positioning of the knots for the surfaces were all done manually — no automatic procedure was involved. However, the B-spline editor was used to position the control vertices. The examples show advantages and applications of the new B-spline scheme.

Figure 5 shows the advantage of converting a C^1-continuous piecewise polynomial quadratic Bézier surface into a quadratic B-spline surface. The movement of a Bézier control point will generally destroy the continuity of the surface (Fig. 5(b)), but the movement of a B-spline control point will preserve the smoothness and C^1-continuity throughout the entire surface (Fig. 5(d)). Thus, when designing an object, one does not need to worry about preserving its smoothness, but, can concentrate solely on designing its shape.

Figure 6 shows two examples of an application to the *polygonal hole problem*. This problem involves a degree k piecewise polynomial surface containing an interior hole. We wish to *patch* up the hole such that the overall smoothness or continuity of the surface is preserved (especially around the boundary of the hole). The idea of the solution is the following. We first represent the piecewise polynomial surface around the hole as a linear combination of B-splines (Theorem 4.1). Then, this B-spline surface is extended into the hole to produce an overall C^{k-1}-continuous fill of the hole.

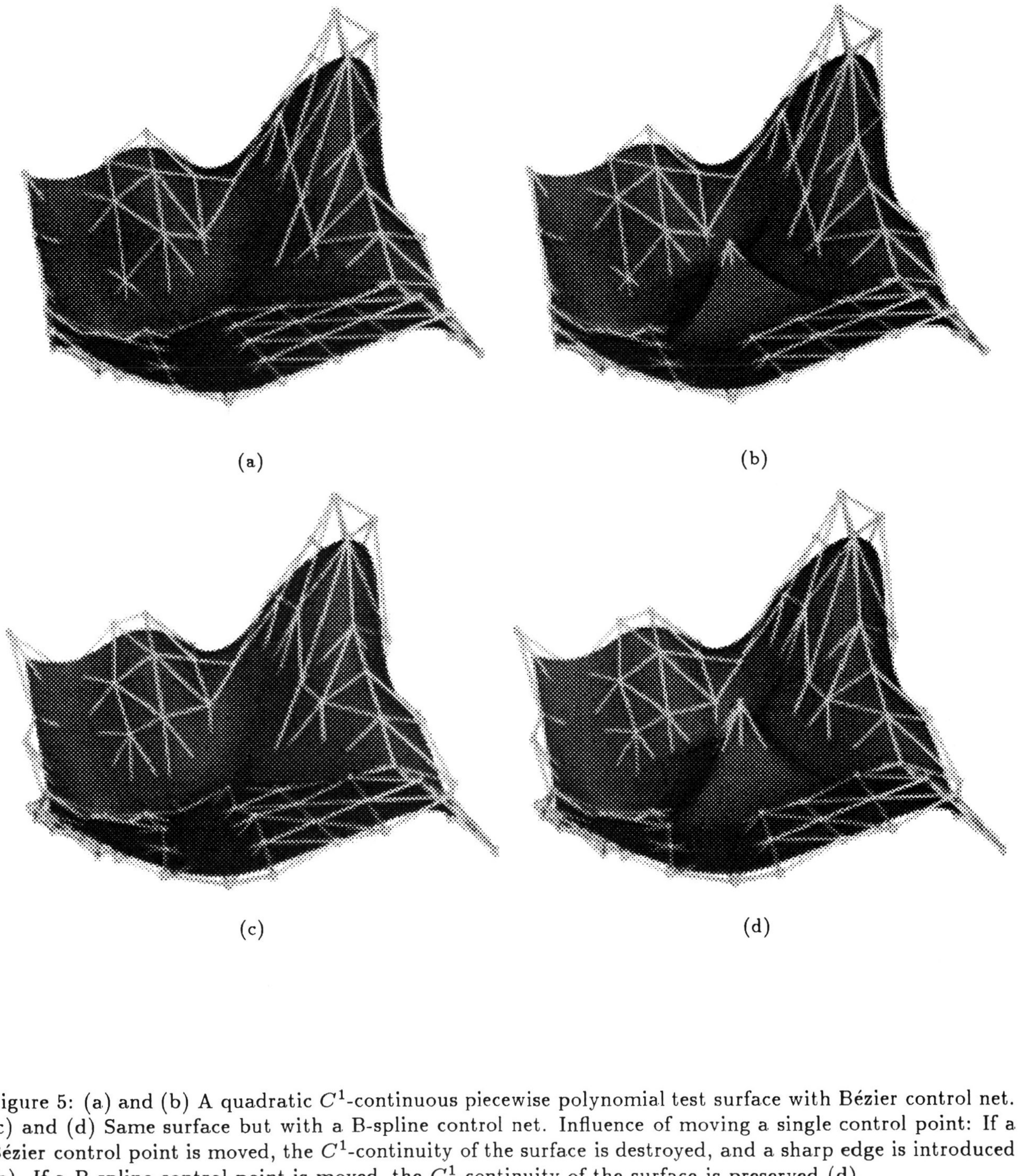
6 Conclusion

The new B-spline scheme offers a method of modelling complex and irregular objects over arbitrary triangulations. Smoothness, locality and the modelling of discontinuities are inherited from simplex splines while control points, affine invariance, and the representation of piecewise polynomials are obtained from B-patches.

The implementation that is presented in this paper has succeeded in demonstrating the practical feasibility of the fundamental algorithms underlying the new surface scheme. Quadratic and cubic surfaces over arbitrary triangulations can be edited and rendered in real-time. Applications like the filling of polygonal holes demonstrate the potential of the new scheme when dealing with concrete design problems. Further improvements to our editor that simplify user input and additional applications are currently under way.

References

[1] 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.



(a) (b) (c) (d)

Figure 5: (a) and (b) A quadratic C^1-continuous piecewise polynomial test surface with Bézier control net. (c) and (d) Same surface but with a B-spline control net. Influence of moving a single control point: If a Bézier control point is moved, the C^1-continuity of the surface is destroyed, and a sharp edge is introduced (b). If a B-spline control point is moved, the C^1-continuity of the surface is preserved (d).

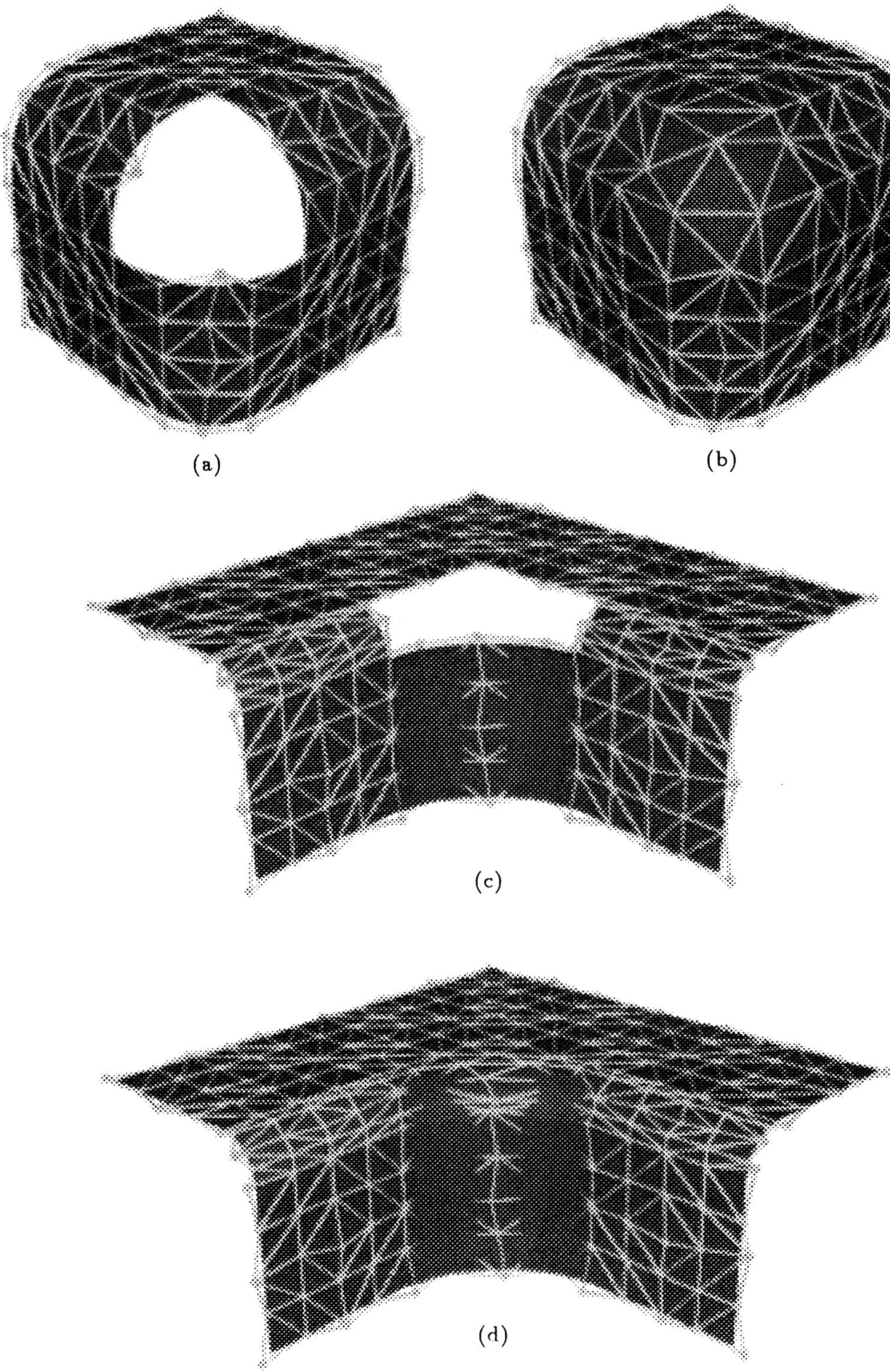


Figure 6: Solving the polygonal hole problem using triangular B-splines: (a) a C^1-continuous quadratic surface containing a 3-sided interior hole. (b) the hole in (a) is filled and the resulting surface is still overall C^1-continuous. (c) and (d) same problem but with a quadratic surface containing a 5-sided hole.

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

[3] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Comput. Aided Design*, 10:350–355, 1978.

[4] 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.

[5] W. Dahmen and C.A. Micchelli. On the linear independence of multivariate B-splines I. Triangulations of simploids. *SIAM J. Numer. Anal.*, 19:993–1012, 1982.

[6] 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.

[7] 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. Accepted for publication in Math. Comp..

[8] C. de Boor. On calculating with B-splines. *J. Approx. Th.*, 6:50–62, 1972.

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

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

[11] D. Doo and M.A. Sabin. Behaviour of recursive division surfaces near extraordinary points. *Comput. Aided Design*, 10:356–360, 1978.

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

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

[14] 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.

[15] T.A. Grandine. The computational costs of simplex spline functions. *SIAM J. Numer. Anal.*, 24:887–890, 1987.

[16] T.A. Grandine. The stable evaluation of multivariate simplex splines. *Math. Comp.*, 50:197–205, 1988.

[17] K. Höllig. Multivariate splines. *SIAM J. Numer. Anal.*, 19:1013–1031, 1982.

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

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

[20] C.A. Micchelli. On a numerically efficient method for computing with multivariate B-splines. In W. Schempp and K. Zeller, editors, *Multivariate Approximation Theory*, pages 211–248, Basel, 1979. Birkhäuser.

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

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

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

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

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

[26] M.A. Sabin. Non-rectangular surface patches suitable for inclusion in a B-spline surface. In *Proc. EUROGRAPHICS'83*, pages 57–69. North-Holland, 1983.

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

[28] H.-P. Seidel. Algorithms for B-patches. In *Proc. Graphics Interface '91*, pages 8–15. Morgan Kaufmann Publishers, 1991.

[29] H.-P. Seidel. Polar forms and triangular B-Spline surfaces. In *Blossoming: The New Polar-Form Approach to Spline Curves and Surfaces, SIGGRAPH '91 Course Notes #26*, pages 8.1–8.52. ACM SIGGRAPH, 1991.

[30] H.-P. Seidel. Symmetric recursive algorithms for surfaces: B-patches and the de Boor algorithm for polynomials over triangles. *Constr. Approx.*, 7:257–279, 1991.

[31] H.-P. Seidel. Representing piecewise polynomials as linear combinations of multivariate B-splines. In T. Lych and L.L. Schumaker, editors, *Curves and Surfaces.* Academic Press, to appear 1992.

[32] C.R. Traas. Practice of bivariate quadratic simplicial splines. In W. Dahmen, M. Gasca, and C.A. Micchelli, editors, *Computation of Curves and Surfaces*, pages 383–422, Dordrecht, 1990. NATO ASI Series, Kluwer Academic Publishers.