1995

Splines and Geometric Modeling

Chandrajit L. Bajaj

Susan Evans

Report Number:
95-056

Bajaj, Chandrajit L. and Evans, Susan, "Splines and Geometric Modeling" (1995). *Department of Computer Science Technical Reports*. Paper 1230.
<https://docs.lib.purdue.edu/cstech/1230>

SPLINES AND GEOMETRIC MODELING

**Chandrajit L. Bajaj
Susan Evans**

**Department of Computer Science
Purdue University
West Lafayette, IN 47907**

**CSD-TR-95-056
September 1995**

Splines and Geometric Modeling

Chandrajit L. Bajaj

Susan Evans

Department of Computer Science,

Purdue University,

West Lafayette, IN 47907

INTRODUCTION

Piecewise polynomials of some fixed degree and continuously differentiable upto some order are known as splines or finite elements. Splines are used in applications ranging from image processing, computer aided design, to the solution of partial differential equations via finite element analysis. The spline fitting problem of constructing a mesh of splines that interpolate or approximate multivariate data is by far the primary research problem in geometric modeling. This survey shall dwell mainly on spline fitting methods in $\mathcal{R}^3$. The following criteria may be used in evaluating these methods:

- Implicit or parametric splines
- Algebraic and Geometric degree of the spline basis
- Number of spline patches required
- Computation and memory required
- Robustness of fitting algorithms
- Local or non-local interpolation
- Splitting or non-splitting of input topology

- Convexity or non-convexity of the given input
- Entities which the user has at his disposal for interactive shape manipulation

Parametric splines consist of vectors of multivariate polynomial (or rational) functions while implicit splines are the zero contours of multivariate polynomials. Comparisons between the implicit versus parametric spline representations does not yield a clear winner, even though the class of implicit splines of a fixed degree contains the class of parametric splines (i.e. bigger is not necessarily better) [4]. The parametric representations possess good properties which include easy to order, easy to generate points on, simpler patches, compact storage, and irreducibility. Implicit representations greatly facilitate the classification problem of whether a given point is on, above, or below a surface and furthermore the implicit polynomial spline class is closed under basic geometric modeling operations such as offset and intersection.

B-splines have emerged as the polynomial basis of choice for working with piecewise polynomial parametric curves and surfaces (section 1). However, it is not possible to model a general closed surface or a surface with handles (higher genus) as a single non-degenerate B-spline. To represent free-form surfaces a significant amount of recent work has been done in the areas of geometric continuity, non-tensor product patches, and in generalizing B-splines. Common schemes include splitting, convex combinations of blending functions, subdivision, and local interpolation by construction.

A-splines are a suitable polynomial basis for working with piecewise implicit polynomial curves and surfaces (section 2). While it is possible to model a general closed surface of arbitrary genus as a single A-spline the geometry of implicit surfaces has proven to be more difficult to specify, interactively control, and polygonize than parametrics. The main shortcoming held against the use of implicit representations is that the zeros of polynomials being multivalued may cause the zero contour to have multiple real sheets, self-intersections and several other undesirable singularities. On the positive end, using polynomials of the same degree, implicit polynomial splines have more degrees of freedom compared with parametric and hence potentially are more flexible to approximate a complicated surface with fewer number of pieces and to achieve

a higher order of smoothness. The potential of implicits remains largely latent and virtually all commercial and many research modeling systems are based on the parametric representation. An exception is SHAstra which allows modeling with both implicit and parametric splines [6].

The basic goal for interactive free-form spline design is to make it easy for the user to control the shape (section 3). The traditional approach is to search for the right spline representation whose local degrees of freedom allows sufficient control for direct manipulation by the user. Using a control mesh to manipulate a surface is a good way to make local modifications, but the user usually needs to modify many control points to make a simple non-local change. This sort of problem is bound to arise whenever the controls provided to the user are tied closely to the representation's degrees of freedom. Variational surfaces allow the user to specify controls that aren't related to the representation. Also recently there has been work toward using physically based models as a tool to explore the phenomenon of "natural" motion. Research includes methods for representing deformable surfaces, deformation operations upon surfaces, and how these surfaces interact with an environment acting to deform them.

1.1. Parametric Splines

1.1.1. Tensor Product Surfaces The theory of tensor product patches requires that data have a rectangular geometry and that the parametrizations of opposite boundary curves be similar. It is based on the concept of bilinear interpolation. Tensor product Bézier surfaces are defined over a rectangular domain. While linear interpolation fits the "simplest" curve between two points, bilinear interpolation fits the "simplest" surface between four points. The tensor product Bézier surfaces are obtained by repeated applications of bilinear interpolation. Properties of tensor product Bézier patches include affine invariance, convex hull property, and the variation diminishing property. The boundary curves of a patch are polynomial curves which have their Bézier polygon given by the boundary polygons of the control net of the patch. Hence the four corners of the control net lie on the patch.

Piecewise bicubic Bézier patches may be used to fit a C^1 surface through a rectangular grid of points. After the rectangular network

of curves has been created there are four coefficients left to determine the corner twists of each patch. These four corner twists can not be specified independently and must satisfy a "compatibility constraint". Common twist estimation methods include zero twists, Adini's twist, Bessel twist, and Brunet's twist. To obtain C^1 continuity between two patches the directions and lengths of the polyhedron edges must be matched across the common polyhedron boundary defining the common boundary curve. When the data points deviate from a nice rectangular grid structure however there is a problem finding a suitable parameterization and piecewise bicubic interpolation will not work. There are alternatives using a higher degree and/or replace C^1 with G^1 .

Piecewise bicubic Hermite patches are similar to the piecewise bicubic Bézier patches but take points, partials, and mixed partials as input. The mixed partials have effect only on the interior shape of the patch and are also called twist vectors. To obtain C^1 continuity between two patches it is necessary and sufficient to match the points, partials, and mixed partials at the matching corners of the two patches [81].

A B-spline surface is a deformation of a planar domain, tessellated into a regular grid of rectangles. Any B-spline surface can be written in piecewise Bézier form. It is natural for the surface to be treated as a collection of tensor product polynomial patches defined over these rectangles. This leads to notions of parametric continuity (denoted C^k continuity) where smoothness is defined in terms of matching derivatives along patch boundaries.

A bicubic B-spline surface may be used to fit a C^2 surface through a rectangular grid of points. For the bicubic Bézier and Hermite patches an initial global survey of the data is needed to determine appropriate values for the tangent and cross-derivative vectors at the patch corners. The bicubic B-spline surface obtains C^2 continuity by overlapping the control polyhedra of neighboring patches.

Instead of being described by control points, Coons patches and Gordon surfaces [34] work by generating a surface from a network of curves. Coons patches are based on a generalization of ruled, or lofted, surfaces. Ruled surfaces fit a surface through two given curves by linear interpolation. Instead of interpolating discrete points they interpolate whole curves. This is often referred to as transfinite

interpolation. Coons patches on the other hand interpolate four boundary curves. They are constructed by composing two lofted surfaces and one bilinear surface, and hence are called bilinearly blended surfaces. A Coons patch has four blending functions $f_1(u), f_2(u), g_1(v), g_2(v)$. There are only two restrictions on the f_i and g_i : each pair must sum to one, and we must have $f_1(0) = g_1(0) = 1$ and $f_2(1) = g_2(1) = 0$ in order to interpolate.

A network of curves may be filled in with a C^1 surface using bicubically blended Coons patches. For this the four twists at the data points and the four cross boundary derivatives $\mathbf{x}_v(u, 0), \mathbf{x}_v(u, 1), \mathbf{x}_u(0, v), \mathbf{x}_u(1, v)$ must be computed. Compatibility problems arise in computing the twists. If $\mathbf{x}(u, v)$ is twice differentiable, we have $\mathbf{x}_{uv} = \mathbf{x}_{vu}$. But this simplification does not apply here. One approach is to adjust the given data so that the incompatibilities disappear. Or if the data can not be changed one can use a method known as *Gregory's square* that replaces the constant twist terms by variable twists that are computed from the cross boundary derivatives. The resulting surface does not have continuous twists at the corners and is rational parametric, which may not be acceptable in certain environments.

Gordon surfaces are a generalization of Coons patches. It is used to construct a surface $\mathbf{g}$ that interpolates a rectangular network of curves, which will be isoparametric curves

$$\mathbf{g}(u_i, v), \quad i = 0, \dots, m$$

and

$$\mathbf{g}(u, v_i), \quad i = 0, \dots, n$$

The idea is to take a univariate interpolation scheme, apply it to all curves $\mathbf{g}(u, v_i)$ and $\mathbf{g}(u_i, v)$, add the resulting surfaces, and subtract the tensor product interpolant that is defined by the univariate scheme. Polynomial interpolation or spline interpolation schemes may be used. The basis functions of the univariate interpolation scheme are called *blending functions*.

Methods for Coons patches and Gordon surfaces can be formulated in terms of boolean sums and projectors. This has also been generalized to create triangular Coons patches.

1.1.2. Generalized B-Spline Surfaces using Multi-sided Patches There are various practical situations where it is desirable

to use surface patches with three or five sides or sometimes even more. B-spline surfaces have been generalized to include multi-sided patches by using convex combinations of blending functions. Multi-sided patches can be generated in basically two ways. Either the polygonal domain which is to be mapped into $\mathbb{R}^3$ is subdivided in the parametric plane, or one uniform equation is used as a combination of equations. In the first case triangular or rectangular elements are put together or recursive subdivision is applied. And in the later case either the known control point based methods are generalized or a weighted sum of interpolants is used.

Multi-sided patch schemes can be characterized by the way they represent a domain point. The situation where a domain point is defined by n dependent coordinates is called 'constrained domain mapping'. Sabin [87] uses a symmetric system of parameters where any two can be independently chosen and the others can be computed from them. Gregory and Charrot [43] use barycentric coordinates to calculate the domain points.

Individual triangles and pentagons with at least C^1 continuity with adjacent rectangles are described by Sabin [87], Hosaka and Kimura [55], and Gregory and Charrot [23, 43]. Sabin defines rational parametric biquadratic three-sided and five-sided patches suitable for embedding in a B-spline like surface among four-sided patches. For the three-sided patches B-spline like configurations are described in which occasional control points are 3-valent. This gives occasional triangular patches among the four-sided ones, like the corners on a suitcase. The five-sided patch is represented by defining a bivariate manifold in five dimensional parameter space, with convenient properties in terms of computing all five parameters from any adjacent two. Each of its five variables is zero at one edge, varies linearly along the adjacent edges, and is zero at all other edges. A generic point on the surface is expressed as a vector function of the five parameters. The n -sided patch developed by Hosaka and Kimura is limited to at most 6 sides.

Gregory and Charrot propose two C^1 interpolation schemes for a pentagonal and triangular surface patches which use blending functions and are compatible with surface patches which have a rectangular domain of definition. The first method is based on the Brown-Little construct of a convex combination or blend of one sided interpolants. The second method is the Gregory-Charrot construct

that uses interpolants which fit simultaneously to two adjacent polygon edges and are given in a boolean sum form. For both of these schemes, in order to get the final patch equation the interpolants must be multiplied by rational convex combination functions which balance the interference of the individual interpolants.

The Brown-Little scheme for triangular patches matches a scalar valued function and its first derivatives on the sides of a triangular domain. The scheme is a convex combination of three constituent interpolants, each of which match data on just one of the sides of the triangle. The original scheme uses interpolants defined along directions normal to the sides of the triangular domain. Modified schemes use interpolants defined along radial directions from a side to its opposite vertex.

The Gregory-Charrot scheme for triangular patches is a vectored valued scheme that uses a convex combination of three constituent interpolants, each of which match data on two sides of the triangle. The two-sided interpolants are constructed using the boolean sum ideas formalized by Gordon [40]. The Gregory-Charrot scheme for pentagonal patches is a convex combination or blend of five component interpolants, each of which match given boundary curves and cross boundary slope conditions on two sides of a regular polygon. The resulting triangular or pentagonal surface patch can be joined with position and slope continuity to adjacent rectangular patches. The Gregory-Charrot scheme generally gives good surfaces for compatible data and avoids the use of singular functions. The Brown-Little scheme is easier to implement, but for the general polygon has only linear precision.

Varady [105] describes a general scheme for filling n -sided surface patches using overlap patches. Overlap patches are made up of n bicubic vertex patches, each interpolating position and tangent data relating to one corner. Instead of using constrained domain mapping where n variables determine a domain point, $2n$ variables are used to determine a domain point and they are constrained only along the polygon sides in both position and a differential sense. These patches are based on the Brown-Little scheme and use linear Taylor interpolation for positional and tangential functions of one polygon edge. A radial direction is also defined by means of the intersection point of the neighbouring polygon sides. Each vertex patch is guaranteed to lie fully within the polygon and disappear

where needed so there is no need to use further weighting functions with the vertex patches are added. On vertex patch interpolates only the vector quantities of one corner and the overlapping of two adjacent vertex patches together provide proper positional and tangential functions for the whole edge of the polygon. These overlap patches give VC^1 continuity and have an extra degree of freedom to adjust the interior of the patch.

Loop and DeRose [64, 63] present generalizations of biquadratic and bicubic B-spline surfaces that are capable representing surfaces of arbitrary topology by placing restrictions on the connectivity of the control mesh, relaxing C^1 continuity to G^1 continuity, and allowing n -sided S-patch elements. This generalized view considers the spline surface to be a collection of possibly rational polynomial maps from independent n -sided polygonal domains, whose union possesses continuity of some number of geometric invariants, such as tangent planes. In this view patches are required to meet with geometric continuity (denoted G^k continuity). This more general view allows patches to be sewn together to describe free form surfaces in more complex ways. We will first discuss the main ideas behind S-patches and then show how they may be incorporated into generalized B-spline surfaces.

The problem that originally motivated the development of S-patches is that of constructing smooth surfaces that interpolate the vertices of an arbitrary polyhedron. S-patches are based on the idea of restricting Bézier simplexes to embedded surfaces. They unify and generalize triangular and tensor product Bézier surfaces by allowing patches to be defined over any convex polygonal domain. S-patches may have any number of boundary curves, have geometrically meaningful control points, are confined to the convex hull of their control points, have separate control over positions and derivatives along boundary curves, and have a geometric construction algorithm based on de Casteljau's algorithm. Also regular S-patches, that is, S-patches defined on regular polygonal domains, possess additional special properties. Regular S-patches can be joined to Bézier triangles with either C^k continuity for arbitrary k , or G^1 continuity.

An n -sided S-patch S is constructed by embedding its n -sided domain polygon P into a simplex Δ whose dimension is one less than the number of sides of the polygon. The edges of the polygon map to edges of the simplex. A Bézier simplex B is then constructed using

Δ as a domain. The patch representation S is obtained by restricting the Bézier simplex to the embedded domain polygon. If E denotes the embedding, the patch representation S can be expressed as

$$S(p) = B \circ E(p), \quad p \in P$$

More precisely, an n -sided S-Patch of depth d is a map $S : P \rightarrow \mathcal{M}$ of the form $S = B \circ E$ where $E : P \rightarrow \Delta$ is an edge-preserving embedding of P into Δ and $B : \Delta \rightarrow \mathcal{M}$ is a rational Bézier simplex of degree d , expressed relative to the domain simplex Δ .

The control net of S is then defined to be the control net of B . Repeated depth elevation produces control nets that converge to the image of Δ , but it does not in general converge to the surface patch except when $n = 3$. Each of the n boundary curves of an S-patch of depth d are rational Bézier curves of degree d defined by the control points associated with the boundary. Just as the domain triangle of a Bézier triangle does not affect the shape of the resulting patch, the simplex Δ does not affect the shape of an S-patch. This can be seen by writing S as

$$S(p) = \sum_{\vec{i}} V_{\vec{i}} W_{\vec{i}}^d, \quad p \in P$$

$$W_{\vec{i}}^d(p) = \frac{w_{\vec{i}} B_{\vec{i}}^d(e_1(p), \dots, e_n(p))}{\sum_{\vec{j}} w_{\vec{j}} B_{\vec{j}}^d(e_1(p), \dots, e_n(p))}$$

where $V_{\vec{i}}$ are the S-patch control points for S relative to P , $w_{\vec{i}}$ is the rational weight associated with the control point $V_{\vec{i}}$, $(e_1(p), \dots, e_n(p))$ are the barycentric coordinates of $E(p)$ relative to Δ , and $W_{\vec{i}}^d$ is the $\vec{i}$ -th S-patch blending function of depth d .

One possible edge-preserving embedding is denoted L . Let $\alpha_i(p)$ denote the ratio of the signed area of the triangle p, p_i, p_{i+1} to the area of the triangle p_i, p_{i+1}, p_{i+2} , where the sign is chosen to be positive if p is inside. Let

$$\pi_i(p) = \alpha_1(p) \cdots \alpha_{i-2}(p) \alpha_{i+1}(p) \cdots \alpha_n(p), \quad i = 1, \dots, n$$

And let

$$l_i(p) = \frac{\pi_i(p)}{\pi_1(p) + \cdots + \pi_n(p)}, \quad i = 1, \dots, n$$

The functions $l_1, \dots, l_n$ are a generalization of barycentric coordinates. They form a partition of unity and are rational polynomial

functions of degree $n - 2$. Moreover they are guaranteed to be non-negative whenever $p \in P$ since each of the functions α_i are nonnegative in this case. The embedding $L : P \rightarrow \Delta$ given by

$$L(p) = l_1(p)v_1 + \cdots + l_n(p)v_n, \quad p \in P$$

is edge-preserving and maps the interior of the polygon P into the interior of the simplex Δ . However the embedding L is pseudoaffine only when the domain polygon is regular. When L is used as the embedding, an n -sided S-patches of depth d is of rational degree $d(n - 2)$.

In [63] Loop and DeRose present generalized B-spline surfaces that use S-patches. Just as for B-spline surfaces, the general surfaces are created as smooth approximations to control meshes (a collection of control vertices together with connectivity information used to define edges and faces). Loop and DeRose present two methods for transforming control meshes into G^1 spline surfaces which differ in the restrictions placed on the control mesh. The first method is a generalization of biquadratic B-splines and requires the control mesh to be constructed entirely from four sided faces, although any number of faces may meet at a vertex. The second method is a generalization of bicubic B-splines and requires that exactly four faces meet at a vertex, although faces may contain any number of edges. These restrictions are sufficiently relaxed to describe surfaces of arbitrary topology, and guarantee that all surfaces have exactly four patches meeting at each interior corner so a simple solution to the "twist compatibility" problem can be utilized.

Lodha [60] constructed surface patches using a "rationally controlled" S-patch representation, where the interior control points are expressed as convex combinations of user-specified incompatible control points using rational blending functions. The patches have a compact form, work for any number of sides, any number of derivatives, and any number of dimensions. This approach can represent S-patches of high rational degrees in terms of a compact, low degree, rationally controlled Bézier representation.

1.1.3. Triangular Surfaces Triangular pieces are a fairly standard tool of approximation theory. For complicated surfaces, no rectangular geometry is apparent and the use of triangular patches is advantageous since every surface can be covered with a triangular

network. Barnhill [14] was one of the first to apply these triangular methods to vector-valued surface patches. Bézier curves admit a symmetric formulation in terms of affine invariant barycentric coordinates. This concept is generalized to surfaces to yield triangular patches.

Consider a triangle with vertices $p_1, p_2, p_3 \in R^2$, and a point $p \in R^2$. Then we may write

$$p = up_1 + vp_2 + wp_3$$

where u, v, w are the barycentric coordinates of p with respect to $\Delta(p_1, p_2, p_3)$ and $u + v + w = 1$. The barycentric coordinates are give by:

$$u = \frac{\text{area}(p, p_2, p_3)}{\text{area}(p_1, p_2, p_3)}$$

$$v = \frac{\text{area}(p_1, p, p_3)}{\text{area}(p_1, p_2, p_3)}$$

$$w = \frac{\text{area}(p_1, p_2, p)}{\text{area}(p_1, p_2, p_3)}$$

$$\text{area}(a, b, c) = \frac{1}{2} \begin{vmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \\ 1 & 1 & 1 \end{vmatrix}$$

Much of the history of triangular Bernstein-Bézier patches is summarized by Farin[33, 34]. The de Casteljau algorithm for triangular patches is a generalization of the corresponding algorithm for curves and uses repeated linear interpolation. The control net is a triangular structure, and a degree n patch has a control net of $\frac{1}{2}(n+1)(n-1)$ vertices. These triangular patches are affine invariant, invariant under affine parameter transformations, have the convex hull property, and have boundary curves that are Bézier polynomials determined by the boundary control vertices. Algorithms exists for degree elevation, subdivision, and differentiation.

Farin [33] presents several schemes to interpolate position and derivative information at the vertices and across edges. C^0 nine parameter interpolants give quadratic precision and interpolate position and gradient at each triangle vertex. C^1 interpolants that are at least quintic need input that is up to second derivative data at the vertices and a cross-boundary derivative at each edge midpoint.

In general a local piecewise polynomial interpolant that interpolates to all derivatives up to order r of some primitive function f at the vertices of a triangulation and is globally r times differentiable must be of degree $n \geq 4r + 1$. For these Hermite interpolants you have to provide derivative data of higher order than the desired order of continuity.

Schmidt [90], and Bohem, Farin, and Hahmann [17] have considered global interpolants to minimize a certain functional (such as strain energy) over the whole triangulated domain. This leads to the solution of a large linear system.

1.1.4. Generalized Triangular Surfaces using Splitting

With splitting schemes every triangle in the triangulation of the data points (also called a “macro-triangle”) is split in to several “mini-triangles”. Split-triangle interpolants do not require derivative information of higher order than the continuity of the desired interpolant. The simplest of the split-triangle interpolants in the C^1 Clough-Tocher interpolant. Each vertex is joined to the centroid, and the macro-triangle is split into three mini-triangles. The first order data that this interpolant requires are position and gradient value at the vertices macro-triangle plus some cross-boundary derivative at the midpoint of each edge. There are twelve data per macro-triangle and cubic polynomials are used over each mini-triangle. The C^1 Powell-Sabin interpolants produce C^1 piecewise quadratic interpolants to C^1 data at the vertices of a triangulated data set. Each macro-triangle is split into several mini triangles. If all angles of the macro-triangle are smaller than 75 degrees, a split into six mini-triangles is performed. Otherwise twelve mini-triangles are generated. Piper [77] and Shirman and Séquin [99] use splitting schemes to locally interpolate mesh vertices with collections of triangular Bézier patches that meet smoothly.

1.1.5. Generalized Triangular Surfaces using Blossoming

The B-patches developed by Seidel [96, 98, 97, 29] 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.

A polynomial surface that has a symmetric recursive evaluation algorithm is called a *B-Patch*. B-patches generalize Bézier patches

over triangles and are characterized by control points and a three parameter family of knots. A sequence

$$\mathcal{K} = (R_0, \dots, R^{n-1}, S_0, \dots, S^{n-1}, T_0, \dots, T^{n-1})$$

of parameters in R^2 is called a *knot net* if (R_i, S_j, T_k) are affinely independent for $0 \leq |\vec{i}| \leq n-1$. The parameters are also called *knots*.

Every bivariate polynomial $F : R^2 \rightarrow R^d$ of degree n has a unique representation

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

as a B-patch with parameters $R_0, \dots, R^{n-1}, S_0, \dots, S^{n-1}, T_0, \dots, T^{n-1}$ in R^2 if the parameters (R_i, S_j, T_k) are affinely independent for $0 \leq |\vec{i}| \leq n-1$. The real-valued polynomials $N_{\vec{i}}^n(U)$ are called the *normalized B-weights* of degree n over $\mathcal{K}$. Let

$$(R_{\vec{i}}^l, S_{\vec{i}}^l, T_{\vec{i}}^l), \quad 1 \leq l \leq n, \quad |\vec{i}| = n-l$$

be affinely independent parameters in R^2 where $\vec{i} = (i, j, k)$ and $|\vec{i}| = i + j + k$. Let

$$r_{\vec{i}}^l(U), \quad s_{\vec{i}}^l(U), \quad t_{\vec{i}}^l(U)$$

be the barycentric coordinates of $U \in R^2$ with respect to $\Delta(R_{\vec{i}}^l, S_{\vec{i}}^l, T_{\vec{i}}^l)$. And let the control points be:

$$P_{\vec{i}} \in R^d, \quad |\vec{i}| = n$$

Every polynomial surface $F : R^2 \rightarrow R^d$ can be evaluated by a recurrence relation. For every polynomial $F : R^2 \rightarrow R^d$ of degree n there exists a unique symmetric n -affine map $f : (R^2)^n \rightarrow R^d$ satisfying

$$f(\underbrace{U, \dots, U}_n) = F(U)$$

called the polar form of F . The *recursive* associated multiaffine version is *symmetric* and only if the parameters $R_{\vec{i}}^l$ depend only on i , the parameters $S_{\vec{i}}^l$ depend only on j , and the parameters $T_{\vec{i}}^l$ depend

only on k . In this case we get triangles (R_i, S_j, T_k) for $0 \leq |\vec{i}| \leq n-1$. A symmetric recursive multiaffine algorithm is given by

$$p_i^0() = P_i$$

$$p_i^l(U_1, \dots, U_l) = \tau_i^l(U_l) p_{i+e_1}^{l-1}(U_1, \dots, U_{l-1}) +$$

$$s_i^l(U_l) p_{i+e_2}^{l-1}(U_1, \dots, U_{l-1}) +$$

$$t_i^l(U_l) p_{i+e_3}^{l-1}(U_1, \dots, U_{l-1})$$

where $1 \leq l \leq n, |\vec{i}| = n-l$, $\tau_i^l(U)$, $s_i^l(U)$, and $t_i^l(U)$ are the barycentric coordinates of $U \in R^2$ with respect to $\Delta(R_i, S_j, T_k)$.

Consider a symmetric generalized de Casteljau algorithm with parameters $R_0, \dots, R_{n-1}, S_0, \dots, S_{n-1}, T_0, \dots, T_{n-1}$. Let $F(U) = P_{0,0,0}^n(U)$ be the resulting polynomial. Then the polar form f of F is given by

$$f(U_1, \dots, U_n) = p_{(0,0,0)}^n(U_1, \dots, U_n)$$

And symmetric 1-affine maps $p_i^l(U_1, \dots, U_l)$ in the associated multi-affine version satisfy

$$p_i^l(U_1, \dots, U_l) =$$

$$f(\underbrace{R_0, \dots, R_{i-1}}_i, \underbrace{S_0, \dots, S_{j-1}}_j, \underbrace{T_0, \dots, T_{k-1}}_k, \underbrace{U_0, \dots, U_l}_l)$$

In particular the control points satisfy

$$P_i = f(\underbrace{R_0, \dots, R_{i-1}}_i, \underbrace{S_0, \dots, S_{j-1}}_j, \underbrace{T_0, \dots, T_{k-1}}_k)$$

Consider the B-patch $F(U)$ with control points P_i of degree n over the knot net

$$\mathcal{K} = (R_0, \dots, R^{n-1}, S_0, \dots, S^{n-1}, T_0, \dots, T^{n-1})$$

and the B-Patch $G(U)$ with control points $\tilde{P}_i$ over the knot net

$$\mathcal{K} = (\tilde{R}_0, \dots, \tilde{R}^{n-1}, S_0, \dots, S^{n-1}, T_0, \dots, T^{n-1})$$

where the knots $S_0, \dots, S_{n-1}, T_0, \dots, T_{n-1}$ all lie on a line L . Then F and G are C^q continuous along L iff the B-patch control points $\tilde{P}_i$ of G satisfy

$$\tilde{P}_i = p_{0,j,k}^i(\underbrace{\tilde{R}_0, \dots, \tilde{R}_{i-1}}_i), \quad 0 \leq i \leq q$$

where the points $p_{0,j,k}^i(\underbrace{\tilde{R}_0, \dots, \tilde{R}_{i-1}}_i)$ are generated by the associated multiaffine version of the symmetric recursive algorithm for F .

The shape of a B-patch is related to its control net in the following ways. The relationship between the control points and the B-Patch surface is affine invariant. If each triangle $\Delta(R_i, S_j, T_k)$ contains the domain triangle $\Delta(R_0, S_0, T_0)$ then $F(U)$ is contained in the convex hull of the control points for any point U in the domain triangle. If all knots $R_0 = \dots = R_{n-1}$ coincide then $F(R_0) = P_{n,0,0}$ is a control point and the surface F is tangent to the control net at this point. This also holds for S and T .

1.1.6. Generalized Triangular Surfaces using Multi-sided Patches To address the problem of irregular patch networks many non-tensor product patches have developed. One approach to create multi-sided patches has been by introducing base points into rational parametric functions. Base points are parameter values for which the homogeneous coordinates (x,y,z,w) are mapped to $(0,0,0,0)$ by the rational parameterization.

Gregory's patch [42] is defined using a special collection of rational basis functions that evaluate to 0/0 at vertices of the parametric domain and thus introduce base points in the resulting parameterization. It is possible to describe Gregory's patch solely in terms of control points and weights.

Warren [107] uses base points to create parameterizations of four-, five-, and six-sided surface patches using rational Bézier surfaces defined over triangular domains. Setting a triangle of weights to zero at one corner of the domain triangle produces a four-sided patch that is the image of the domain triangle. This technique can be generalized to create five- and six-sided patches by treating each vertex of the triangular domain independently. The approach differs from previous multi-sided patches in that the patches are created using the properties of triangular Bézier surfaces. Extra sides are added to the patches by setting the weights of an appropriate collection of control points to zero.

1.1.7. Spline Surfaces Over Meshes With Arbitrary Topology The representation of free-form surfaces is one of the major issues in geometric modeling. The problem of constructing a surface

from an irregular mesh in space has been considered by many. These surfaces are generally defined in a piecewise manner by smoothing joining several mostly four-sided patches. The patches are given in vector valued parametric form, mapping a rectangular parametric domain into $\mathbb{R}^3$.

Local construction, blending polynomial pieces, and splitting are common approaches to constructing surfaces over irregular meshes. Because of its nonlinear nature and the advantages of a local construction, different approaches have been centered around selecting geometrically meaningful variables that can be fixed (as input or derived from data) so as to arrive at a sufficient and consistent set of linear constraints on the remaining variables. Blending polynomial pieces means constructing k pieces for a k -sided mesh facet such that each piece matches a part of the facet data and a convex combination of the pieces matches the whole. Blending approaches prescribe a mesh of boundary curves and their normal derivatives. However for this approach the existence of a well-defined tangent plane at the data points is not sufficient to guarantee the existence of a C^1 mesh interpolant since the mixed derivatives p_{uv} and p_{vu} are given independently at any point p . Splitting approaches on the other hand expect at least tangent vectors at the data points and sometimes the complete boundary to be given.

Chiokura and Kimura [24] construct a solid with free-form surfaces. They start with a polyhedral solid which can be edited using local modifications and generate a curve model for the solid consisting of cubic Bézier curves and straight lines. *Gregory Patches* are used to interpolate the multi-sided faces of the curved model. One step of subdivision is applied to make all faces have four edges. The *Gregory Patches* are generalizations of bicubic Bézier patches that contain removable singularities at patch corners. The normals along the four boundary curves can be specified with no compatibility constraint, and are computed as a quadratic function. A point on the patch is represented as an addition of sixteen points to which various weights are given.

Van Wijk [104] gives a generalization of B-splines that uses subdivision and Coons patches, but this scheme uses bicubic tensor product patches exclusively and imposes relatively strict requirements on the form of the control mesh. The faces of the mesh must have four edges, and at all vertices either three or four edges are al-

lowed to meet or at all vertices an odd number of edges must meet.

Peters [73] describes an algorithm for the interpolation of a mesh of points in space by a piecewise parametric Bézier C^1 surface. At each point the normal can be specified. The surface is constructed locally and consists of cubic and bicubic patches to match the underlying mesh facets. Instead of restricting the boundaries themselves, it constrains the normal direction along the patch boundaries such that the normal along patch boundaries varies linearly. The method encounters a version of the 'compatibility problem' since it uses only one patch per facet, and one additional constraint is added for each point that has an even number of neighbors.

Peters [76] gives an algorithm for the local interpolation of a mesh of cubic curves with 3- and 4-sided facets by a piecewise cubic C^1 surface. Conditions for admissible data for cubic mesh interpolation with splitting are given and a scheme for generating admissible curve meshes from data points and normals is presented. The construction of the surface is local and consists of splitting and averaging. The algorithm guarantees interpolating surfaces without cusps and has a simple extension to n -sided facets with a piecewise quartic surface.

Peters [74] considers the interpolation of a mesh of curves by a smooth regularly parameterized surface with one polynomial piece per facet. Not every mesh with a well-defined tangent plane at the mesh points has such an interpolant. Necessary and sufficient *vertex enclosure* constraints on a mesh of polynomial curves that guarantees the existence of a regular smooth interpolant are given. The curvature of mesh curves emanating from mesh points with an even number of neighbors must satisfy an additional "vertex enclosure constraint". The vertex enclosure constraint is automatically satisfied by the splitting construction and can be satisfied by singularly parameterizing one of the boundary curves. An algorithm for the local interpolation of a cubic curve mesh by a piecewise [bi]quartic C^1 surface is described. The scheme is based on a sufficient constraint that forces the mesh curves to interpolate second-order data at the mesh points. Rational patches, singular parameterizations, and the splitting of patches are interpreted as techniques to enforce the vertex enclosure constraint. Farin, Piper, Shirman, Séquin, and Peters [33, 77, 99, 74, ?] locally interpolate mesh vertices with collections of triangular or rectangular Bézier

patches that meet smoothly. In Mann et al. [65] these results are surveyed and it is concluded that local polynomial interpolants generally produce unsatisfactory shapes.

Reif [84] constructs a piecewise G^1 spline surface based on biquadratic rectangular Bézier patches from a set of control points on meshes with arbitrary topology. Geometrical smoothness conditions are used only near the singular vertices of a mesh. He constructs an additional ring of "G-edges" around singular vertices and expresses the smoothness conditions in a system of linear equations.

Peters [75] gives an algorithm for refining an irregular mesh of points into a bivariate C^1 surface. The algorithm generalizes the construction of quadratic splines from a mesh of control points, and an explicit parameterization of the surface with quadratic and cubic pieces is given. When the mesh is regular then a quadratic spline surface is generated. Irregular input meshes with nonquadrilateral mesh cells more or fewer than four cells meeting at a point are allowed and generate spline spaces that generalize the space of quadratic splines. The main idea is to refine the irregular input mesh by the averaging process of Doo-Sabin and generate strips of regular mesh points that isolate regions of irregular points. After the mesh has been refined each mesh point is surrounded by four cells. Control points of symmetric piecewise quadratic box splines over the four-direction mesh are generated and the holes are filled with cubic triangular patches. The cubics are determined by averaging the box spline control points. An optional conversion of the box splines into Bernstein-Bézier form is an evaluation step that creates a large number of patches whose control mesh is closer to the surface than the box spline control mesh. The algorithm can model bivariate open or closed surfaces of general topological structure. However the algorithm generates a large number of patches relative to the number of faces in the control mesh.

Loop [62] constructs a piecewise G^1 spline surface composed of sextic triangular Bézier patches in one-to-one correspondence with the faces of a triangular control mesh. Surfaces of arbitrary topological type are created by approximating any mesh that represents a triangulated 2-manifold. The surface has local support and is affine invariant since the Bézier points of the patches are affine combinations of the vertices of the mesh. A pair of shape parameters is available at each vertex of the mesh for additional local control over

the shape of the surface. If some local regularities are present in the structure of the control mesh then the corresponding patches may be parameterized as quintic or even quartic Bézier triangles. In the case of a regular parameterization the surface generated by this method is equivalent to a quartic C^2 triangular B-spline. The question of whether there exists values in the space of the shape parameters so that the affine combinations of mesh vertices are strictly convex has not been resolved.

1.1.8. Subdivision Surfaces Subdivision techniques can be used to produce generally pleasing surfaces from arbitrary control meshes. Subdivision consists of splitting and averaging. Each edge or face is split and each new vertex introduced by the splitting is positioned at a fixed affine combination of its neighbor's weights.

The algorithms start with a polyhedral configuration of points, edges, and faces. The control mesh will in general consist of large regular regions and isolated singular regions. Subdivision enlarges the regular regions of the control net and shrinks the singular regions. Each application of the subdivision algorithm constructs a refined polyhedron, consisting of more points and smaller faces, tending in the limit to a smooth surface. In general the new control points are computed as a linear combination of old control points. The associated matrix is called the *subdivision matrix*.

The earliest of these approaches are the recursive subdivision schemes of Doo and Sabin [32] and Catmull and Clark [21]. These algorithms generate C^1 surfaces that interpolate the centroids of all faces at every step of subdivision. Nasri [69] describes a recursive subdivision surface scheme that is capable of interpolating points on irregular networks as well as normal vectors given at these points. The subdivision scheme developed by Loop [61] splits each triangle of a triangular mesh into four triangles. Each new vertex is positioned using a fixed convex combination of the vertices of the original mesh. The final limit surface has a continuous tangent plane. Hoppe et al. [46] extends Loop's method to incorporate sharp edges into the final limit surface. The vertices of the initial polyhedron are tagged as belonging on a face, edge, or vertex of the final limit surface. Based on this tag different averaging masks are used to produce new polyhedra.

Storry and Ball [100] demonstrate that a B-spline subdivision

patch can be fitted into a general n -sided area of a bicubic surface with at least tangent plane continuity on the boundary. A Hermite formulation is used for the surface patches, and after the control points have been determined the subdivision algorithm is applied to produce an n -sided patch of optimal continuity properties. One degree of freedom is identified and related to shape control. However tangent plane continuity and the existence of a limit for the normal vectors is a rather weak smoothness category since local self-intersections of the surface are still possible.

Reif [83] presents a unified approach to subdivision algorithms for meshes with arbitrary topology and gives a sufficient condition for the regularity of the surface. The existence of a smooth regular parametrization for the generated surface near the point is determined from the leading eigenvalues of the subdivision matrix and an associated characteristic map.

1.1.9. Multivariate Box Splines Multivariate splines are a generalization of univariate B-splines to a multivariate setting. Multivariate splines have applications in data fitting, computer-aided design, the finite element method, and image analysis. Alfeld [2], Dahmen and Micchelli [31], Höllig [51] and Schumaker [91] summarize much of the history of scattered data fitting and multivariate splines. Work on splines has traditionally been for a given planar triangulation using a polynomial function basis [2, 96]. Regular arrays of triangles were first described by Frederickson [39], and later independently by Sabin [86] and Sablonniere [88]. These results are now regarded as being the special "box spline" case of multivariate B-splines.

Box-splines are multivariate generalizations of B-splines with uniform knots. Many of the basis functions used in finite element calculations on uniform triangles occur as special instances of box splines. In general a box spline is a locally supported piecewise polynomial. One can define translates of box splines which form a negative partition of unity. Algorithms for box splines can be found in [19, 26, 28].

In the bivariate case box splines correspond to surfaces defined over a regular tessellation of the plane. If the tessellation is composed of triangles, it is possible to represent the surface as a collection of Bernstein-Bézier patches [28, 20]. The two most commonly used

special tessellations arise from a rectangular grid by drawing in lines in north-easterly diagonals in each subrectangle or by drawing in both diagonals for each subrectangle. For these special triangulations there is an elegant way to construct locally supported splines. The classical unnormalized univariate B-spline can be defined recursively by

$$M_k(x) = \int_0^1 M_{k-1}(x-t)dt$$

where

$$M_1(x) = \begin{cases} 1 & \text{if } x \in [0, 1) \\ 0 & \text{otherwise} \end{cases}$$

To generalize this to a bivariate case suppose we are given a $2 \times m$ matrix $A_m = [a^1 \dots a^m]$ where we may assume a^1 and a^2 are linearly independent. Let

$$M(x, y|A_m) = \int_0^1 M(x - ta_x^m, y - ta_y^m|A_{m-1})dt$$

where $a^i = (a_x^i, a_y^i)$ and

$$M_1(x) = \begin{cases} 1/|\det(A_2)| & \text{if } (x, y) \in [A_2) \\ 0 & \text{otherwise.} \end{cases}$$

with

$$[A_j) = t_1 a^1 + \dots + t_j a^j : 0 \leq t_i < 1, 1 \leq i \leq j$$

for $j = 2, \dots, m$. Each of the functions $M(x, y|A_j)$ is called a *box spline* with the follow properties:

1. $M(x, y|A_m)$ is positive on the interior of the set $[A_m)$, and vanishes outside of its closure $[A_m]$
2. $\int_{[A_m]} M(x, y|A_m) dx dy = 1$
3. $M(x, y|A_m)$ is a piecewise polynomial of degree at most $m-2$ defined on the partition of $[A_m]$ obtained by connecting each of its vertices with all other vertices
4. $M(x, y|A_m) \in C^{r-1}$, where r is the smallest integer k such that all the subsets of $m-k$ vectors chosen from A_m still span $\mathbb{R}^2$.
5. The set of integer translates $M(x-i, y-j|A_m)$ are linearly independent on $\mathbb{R}^2$ if and only if all sets of pairs of vectors b_1, b_2 from A_m are such that $\det[b_1, b_2]$ has values -1, 0, or 1.

In [31] discrete multivariate truncated powers were defined and it was shown that they provide a relation between continuous truncated powers and box splines. Cohen et al. [26] introduced discrete box splines. Given integers $n \geq s \geq 1$ and vectors $X = (x^1, \dots, x^n)$ with $x^j \in \mathbb{R}^s, j = 1, \dots, n$, the box spline $M_X = M(\cdot|X) : \mathbb{R}^2 \rightarrow \mathbb{R}$ is defined as a distribution give by

$$(1.1) \quad \int_{\mathbb{R}^s} M_X(x) f(x) dx = \int_{I^n} f(Xv) dv_1 \cdots dv_n$$

for all $f \in C_0^\infty(\mathbb{R}^2)$ where $I^n = [0, 1]^n$ and $Xv = v_1 x^1 + \cdots + v_n x^n$. The discrete box splines are define by discretizing (1.1) and converge in a weak sense to the continuous box spline.

1.1.10. Multivariate Simplex Splines Splines over arbitrary triangulations of the parameter plane were first considered by Dahmen and Michelli [30] and Höllig [51]. These multivariate splines are defined as projections of simplices and are therefore called simplex splines. Suppose B is a convex set in $\mathbb{R}^m$ and that P is an affine map of $\mathbb{R}^m$ into $\mathbb{R}^2$. Then the corresponding analog of the univariate B-spline M_B is defined to be the distribution which represents the linear functional

$$f \rightarrow \int_B f \circ P$$

If B is a polytope (the convex hull of a finite set of points) we get *polyhedral* splines with the following properties: [91]

1. M_B is positive on the interior of the set Ω_B which is the convex hull of $Px : x \in B$, and vanishes outside Ω_B
2. $\int_{\Omega_B} M_B dx dy = 1$
3. If B is in general position and P is onto $\mathbb{R}^2$ then M_B is a piecewise polynomial of degree at most $m - 2$ defined on the partition of Ω_B obtained by connecting each of its vertices with all other vertices
4. $M_B \in C^{\tau-1}$, where τ is the smallest integer k such that P maps all k -dimensional faces of B into a set in $\mathbb{R}^2$ with interior

If B is chosen to be a simplex in $\mathbb{R}^m$ then we get simplex splines.

Let $V = \{v_0, \dots, v_m\}$ be a finite set of points in $\mathbb{R}^2$, and let $[V] = \{v_0, \dots, v_n\}$ denote the convex hull of an arbitrary set

$\mathbf{v}_0, \dots, \mathbf{v}_n \in \mathbb{R}^2$. For any ordered set of affinely independent points $W = \mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^2$ define

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

$$d_0(W) = \det \begin{pmatrix} 1 & 1 & 1 \\ \mathbf{u} & \mathbf{w}_1 & \mathbf{w}_2 \end{pmatrix}$$

$$d_1(W) = \det \begin{pmatrix} 1 & 1 & 1 \\ \mathbf{w}_0 & \mathbf{u} & \mathbf{w}_2 \end{pmatrix}$$

$$d_2(W) = \det \begin{pmatrix} 1 & 1 & 1 \\ \mathbf{w}_0 & \mathbf{w}_1 & \mathbf{u} \end{pmatrix}$$

The simplex spline $M(\mathbf{u} \mid V) = M(\mathbf{u} \mid \mathbf{v}_0, \dots, \mathbf{v}_m)$ is defined recursively as follows. For $V = \mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$ set

$$M(\mathbf{u} \mid \mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2) = \chi[\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2](\mathbf{u}) / \left| \det \begin{pmatrix} 1 & 1 & 1 \\ \mathbf{v}_0 & \mathbf{v}_1 & \mathbf{v}_2 \end{pmatrix} \right|,$$

where $\chi[\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2](\mathbf{u})$ is the characteristic function on $[\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2]$. For $V = \mathbf{v}_0, \dots, \mathbf{v}_m$, $m > 2$, set

$$M(\mathbf{u} \mid V) = \sum_{j=0}^2 \frac{d_j(W \mid \mathbf{u})}{d(W)} M(\mathbf{u} \mid V \setminus \mathbf{v}_{i_j})$$

where $W = \mathbf{v}_{i_0}, \mathbf{v}_{i_1}, \mathbf{v}_{i_2}$ is any subset of affinely independent points in V . The simplex spline $M(\mathbf{u} \mid \mathbf{v}_0, \dots, \mathbf{v}_m)$ is positive and is known to be a piecewise polynomial of degree $k = m - 2$, supported on the convex hull $[\mathbf{v}_0, \dots, \mathbf{v}_m]$, that is C^{k-1} continuous everywhere. Grandine [41] gives a general method for the stable evaluation of multivariate simplex splines.

Auerbach [3] constructs approximations with simplex splines over irregular triangles. Bivariate quadratic simplicial B-splines defined by their corresponding sets of knots derived from a (suboptimal) constrained Delaunay triangulation of the domain are employed to obtain a C^1 surface. This approach is well suited for scattered data. Each vertex of a given triangle is associated with two additional points which give rise to six configurations of five knots defining six linearly independent bivariate quadratic B-splines supported over

the convex hull of the corresponding five knots. The coefficients of the linear combinations of normalized simplicial B-splines are visualized as geometric control points satisfying the convex hull property.

Fong and Seidel [36, 35] construct multivariate B-splines for quadratics and cubics by matching B-patches with simplex splines. The surface scheme is an approximation scheme based on blending functions and control points and allows the modeling of C^{k-1} continuous piecewise polynomial surfaces of degree k over arbitrary triangulations of the parameter plane. The resulting surfaces are defined as linear combinations of the blending functions and are parametric piecewise polynomials over a triangulation of the parameter plane whose shape is determined by their control points. In order to construct simplex splines of degree k over the triangulation T of the parameter plane, they first assign a sequence of knots $t_{i0}, \dots, t_{ik}$ to every vertex t_i of the triangulation in such a way that $t_{i0} = t_i$ and that any set of three knots is affinely independent. They then consider the simplex splines

$$M_{\beta}^I(\mathbf{u}) = M(\mathbf{u} \mid V_{\beta}^I)$$

where

$$V_{\beta}^I = \mathbf{v}_{i_0 0}, \dots, \mathbf{v}_{i_0 \beta_0}, \mathbf{v}_{i_1 0}, \dots, \mathbf{v}_{i_1 \beta_1}, \mathbf{v}_{i_2 0}, \dots, \mathbf{v}_{i_2 \beta_2},$$

$I = (i_0, i_1, i_2)$, $\beta = (\beta_0, \beta_1, \beta_2)$, $\beta_0 + \beta_1 + \beta_2 = k$, and $[\mathbf{v}_{i_0}, \mathbf{v}_{i_1}, \mathbf{v}_{i_2}]$ is one of the triangles in the triangulation of the parameter plane. Each simplex spline $M(\mathbf{u} \mid V_{\beta}^I)$ can then be associated with a B-patch blending function B_{β}^I and a normalized B-spline N_{β}^I , where $V_{\beta}^I = t_{i,j,l} \mid j = 0, 1, 2, l = 0, \dots, \beta_j$. These normalized B-splines are used as blending functions in their scheme.

Their surface 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. Smoothness, locality, and modeling of discontinuities are inherited from simplex splines, while control points, affine invariance, and the representation of piecewise polynomials are inherited from B-patches. Additional degrees of freedom in the underlying knot net allow for the modeling of discontinuities. In [35] they give an implementation to demonstrate the practical feasibility of the fundamental algorithms in their surface scheme.

Quadratic and cubic surfaces for modeling complex and irregular objects over arbitrary triangulations can be edited and rendered in real time. Applications such as the filling of polygonal holes demonstrate the potential of the new scheme when dealing with concrete design problems.

1.2. Implicit Polynomial Splines

In this section we review some approaches to implicit surface fitting and scattered data interpolation methods. There are two main advantages of using implicit surfaces instead of parametrics. First, the set of algebraic surfaces are closed under basic modeling operations such as offset and intersection that are often required in a solid modeling system. Second, for the same polynomial of degree n , implicit algebraic surfaces have more degrees of freedom ($= \frac{(n+3)(n+2)(n+1)}{6} - 1$) compared with $4n + 2$ degrees of freedom for rational parametric surfaces.

The quadric surfaces include spheres, cones, and cylinders, which are fundamental entities in many solid modeling systems. Sederberg [94, 95] used cubic surfaces for free-form surface modeling. Surfaces of implicit degree four, such as tori and cyclides, are needed in creating blends in geometric modeling applications. Blinn [18] used implicit surfaces to model electron density maps of molecular structures. Wyvill [111] used implicit surfaces to model animate natural phenomena such as smoke, clouds, mountains, coastlines, living forms, mud, water, and fabrics.

1.2.1. Surfaces Over Meshes of Arbitrary Topology The geometry of implicit surfaces has proven to be more difficult to specify, interactively control, and polygonize than parametrics. Sederberg [93] showed how various smooth implicit algebraic surfaces in trivariate Bernstein basis can be manipulated as functions in Bézier control tetrahedra with finite weights. He showed that if the coefficients of the Bernstein-Bézier form of the trivariate polynomial on the lines that parallel one edge, say L , of the tetrahedron all increase (or decrease) monotonically in the same direction, then any line parallel to L will intersect the zero contour algebraic surface patch at most once. Patrikalakis and Kriezis [72] extended this by considering implicit algebraic surfaces in a tensor product B-spline basis. However the problem of selecting weights or specifying knot

sequences for C^1 meshes of implicit algebraic surface patches which fit given spatial data was left open.

The problem of constructing a C^1 mesh of implicit algebraic patches based on an input polyhedron $\mathcal{P}$ has been considered by many. Dahmen [27] presented a scheme for constructing C^1 continuous piecewise quadric surface patches over a data triangulation in space. In his construction each triangular face is split and replaced by six micro quadric triangular patches, similar to the splitting scheme of Powell–Sabin [80]. Dahmen’s technique however works only if the original triangulation of the data set allows a transversal system of planes, and hence is quite restricted. Moore and Warren [67] extended the marching cubes scheme to compute a C^1 piecewise quadratic approximation to scattered data using a Powell–Sabin like split over subcubes. Guo [44] used cubics to create free-form geometric models and enforced monotonicity conditions on a cubic polynomial along the direction from one vertex to a point of the opposite face of the vertex. He derived a condition $a_{\lambda-e_1+e_4} - a_\lambda \geq 0$ for all λ with $\lambda_1 \geq 1$, where a_λ are the coefficients of the cubic in Bernstein–Bézier form.

Lodha [60] constructed low degree surfaces with both parametric and implicit representations and investigated their properties. A method is described for creating quadratic triangular Bézier surface patches which lie on implicit quadric surfaces. And another method is described for creating biquadratic tensor product Bézier surface patches which lie on implicit cubic surfaces. The resulting patches satisfy all the standard properties of parametric Bézier surfaces, including interpolation of the corners of the control polyhedron and the convex hull property.

Bajaj and Ihm [10] construct low-degree algebraic surfaces that approximate or contain with C^1 continuity any collection of points and algebraic space curves with derivative information. Their Hermite interpolation algorithm solves a homogeneous linear system of equations to compute the coefficients of the polynomial defining the algebraic surface. Bajaj, Ihm and Warren [12] extend this idea to C^k (rescaling continuity) interpolate or least squares approximate implicit or parametric curves in space. They show this problem can be formulated as a constrained quadratic minimization problem, where the algebraic distance is minimized instead of the geometric distance.

Bajaj, Ihm, Guo, and Dahmen [27, 44, 45, 10] provide heuristics based on monotonicity and least square approximation to circumvent the multiple sheeted and singularity problems of implicit patches. Bajaj [5, 11, 12] constructed implicit surfaces to solve the scattered data fitting problem. Bajaj and Ihm [10] considered an arbitrary spatial triangulation $\mathcal{T}$ consisting of vertices in $\mathcal{R}^3$ (or more generally a simplicial polyhedron $\mathcal{P}$ when the triangulation is closed) with possibly normal vectors at the vertex points. Their algorithm constructs a C^1 continuous mesh of real implicit algebraic surface patches over $\mathcal{T}$ or $\mathcal{P}$. The scheme is local (each patch has independent free parameters) and there is no local splitting. The algorithm first converts the given triangulation or polyhedron into a curvilinear wireframe with at most cubic parametric curves which C^1 interpolate all the vertices. The curvilinear wireframe is then fleshed to produce a single implicit surface patch of degree at most 7 for each triangular face $\mathcal{T}$ of $\mathcal{P}$. If the triangulation is convex then the degree is at most 5. Similar techniques exist for parametrics [74, 33, 89] however the geometric degree of the solution surfaces tend to be prohibitively high.

Bajaj, Chen and Xu [9] construct 3- and 4-sided A-patches that are implicit surfaces in Bernstein-Bézier(BB) form that are smooth and single-sheeted. They give sufficiency conditions for the BB form of a trivariate polynomials within a tetrahedron such that the zero contour of the polynomial is a single sheeted non-singular surface within the tetrahedron and its cubic-mesh complex for the polyhedron $\mathcal{P}$ is guaranteed to be both nonsingular and single sheeted. They distinguish between convex and non-convex facets and edges of the triangulation. For non-convex facets and edges a double-sided tetrahedra is built and for convex facets and edges single-sided tetrahedra are built. A generalization of Sederberg's condition is given for a three-sided j-patch where any line segment passing through the j-th vertex of the tetrahedron and its opposite face intersects the patch only once. Instead of having coefficients be monotonically increasing or decreasing there is a single sign change condition. There are also free parameters for both local and global shape control.

1.2.2. Implicit Surfaces and Surfaces-on-Surfaces from Scattered Data

Here we consider the problem of reconstructing

surfaces and scalar fields defined over it (surfaces-on-surfaces), from scattered trivariate data. The points are assumed sampled from the surface of a 3D object, and the sampling is assumed to be *dense* for unambiguous reconstruction. Laser range scanners are able to produce such a dense sampling, usually organized in a rectangular grid, of an object surface. Some 3D scanners are also able to measure the RGB components of the object color (i.e. three scalar fields) at each sampled point. When the object has a simple shape, this grid of points can be a sufficient representation. However, multiple scans are needed for objects with more complicated geometry, e.g. objects with holes, handles, pockets cannot be scanned in a single pass. Other applications, for example recovering the shape of a bone from contour data extracted from a CT scan, require reconstruction of a surface from data points organized in slices. The approach of considering the input points as unorganized has the advantage of generating cross-derivatives by a uniform treatment of all spatial directions.

Bajaj, Bernardini and Xu [8] reconstruct the sampled surface using implicit Bernstein-Bézier patches, which are guaranteed to be single-sheeted within each tetrahedron (barycentric basis). Their scheme effectively utilizes an incremental Delaunay 3D triangulation for a more adaptive fit; the dual 3D Voronoi diagram for efficient point location in signed distance computations and degree three implicit surface patches. Furthermore, in the same time they also compute a C^1 smooth approximation of the sampled surface-on-surface. Bajaj, Bernardini and Xu [] have also developed a method similar to the one described here, but based on tensor-product Bernstein-Bézier patches [7].

A different, three-step solution is described in papers by Hoppe et. al [54, 52, 53]. In the first phase, a triangular mesh that approximates the data points is created. In a second phase, the mesh is optimized with respect to the number of triangles and the distance from the data points. A third step constructs a smooth surface from the mesh.

The problem of modeling and visualizing just surfaces-on-surfaces arises in several physical analysis application areas: characterizing the rain fall on the earth, the pressure on the wing of an airplane and the temperature on the surface of a human body. A number of methods have been developed for dealing with this problem.

Currently known approaches for approximating surface-on-surface data however possess restrictions either on the domain surfaces or the surface-on-surface. The domain surfaces are usually assumed to be spherical, convex or genus zero. The surface-on-surface are not always polynomial [15], [70] or rather higher order polynomial [85] or a large number of pieces [1] compared to the approach of this paper. The method of [1] is a C^1 Clough-Tocher scheme that splits a tetrahedron into 4 subtetrahedra, uses degree 5 polynomials and requires C^2 data on the vertices of each subtetrahedron. Another Clough-Tocher scheme [110] requires only C^1 data at the vertices, for again constructing a C^1 function which is a cubic polynomial over each subtetrahedron, however splits the original tetrahedron into 12 pieces. A C^1 scheme [85] that does not split each tetrahedron uses degree 9 polynomials and requires C^4 data at the vertices. In extending the method of [85] to a C^2 scheme, requires degree 17 polynomials and C^8 data at the vertices of each tetrahedron. Compared to these approaches, the C^1/C^2 construction of [13] has no splitting and uses much lower degree polynomials (cubic/quintic) requiring only C^1/C^2 data respectively, at the vertices of each tetrahedron.

1.2.3. Implicit Surface Blending In constructing objects a designer often adds surfaces whose sole function is to provide a smooth transition between functional features of the object. Such surfaces are referred to as blending surfaces. Most approaches to blending are generalizations of Liming's [59] method of constructing conics tangent to two given lines. Middleditch and Sears [66] generalized this idea to blend the edge between two implicit surfaces $S_1(x, y, z) = 0$ and $S_2(x, y, z) = 0$. They find a conic $Q : \mathbb{R}^2 \rightarrow \mathbb{R}$ such that $Q^{-1}(0)$ is tangent to the positive coordinate axes and then form the blending surface $(Q \circ S)^{-1}(0)$ where $S = (S_1, S_2) : \mathbb{R}^3 \rightarrow \mathbb{R}^2$. When the points of tangency are fixed there is a one parameter family of such conics and the shape of the blending surface can be varied by changing the shape of Q . Rockwood and Owen [71] use a similar approach except superellipsoids are used instead of quadrics and the shape of the blend is controlled by changing the degree of the ellipsoids.

Holmström [58] blends n surfaces $S_i(x, y, z) = 0, i = 1, \dots, n$ by generalizing this approach. He uses a piecewise quadric function $Q : \mathbb{R}^n \rightarrow \mathbb{R}$ and a blending surface of the form $(Q \circ S)^{-1}(0)$ where

$S = (S_i) : \mathbb{R}^n \rightarrow \mathbb{R}^2$. This method assumes the blend is a convex combination of surfaces, that is near the intersection of the surfaces the solid is defined by $S_i(x, y, z) \geq 0$.

Hoffmann and Hopcroft [48, 50, 49] use similar affine and projective potential methods to blend two or three algebraic surfaces with quadrics. The blending surfaces can be derived by substitution from a parametric base curve. The affine potential method views the base curve in affine space and the projective potential method views the base curve in projective space. Given quadrics $G = 0$ and $F = 0$, a quadric $K = 0$ is chosen as a clipping surface that intersects the other two surfaces in the desired curves of tangency. The blending surfaces is then given by $K^2 - \mu GH$ where μ controls the curvature of the resulting blending surface. For the projective method $K = aG_1 + bG_2 - abW$ where a and b are design parameters and W is a free polynomial. The affine potential method is not general enough to derive all quartic blends of the form $K^2 - \mu GH$, but all may be derived by using a projective base polynomial f and substituting for the homogenizing variable.

Warren [106] gives a definition for geometric continuity of algebraic surfaces and shows that algebraic surfaces that meet smoothly under this condition have a very specific form that can be described in terms of ideals. Any polynomial whose zero set blends the zero sets of several other polynomials is always expressible as a simple combination of these polynomials. More specifically any polynomial F defining a surface that is tangent to the surfaces ($G_i = 0$) along the clipping surface ($K = 0$) can be expressed as

$$F = \prod G_i - aK^2$$

where a is greater than zero.

Kosters [56, 57] generalizes the potential method by considering the probleming of C^m blendings of d hyperplanes in d -dimensional space meeting at a single point or corner using implicitly defined blending surfaces given by homogeneous polynomials. For convex corners a C^m blending can be construct with patches of degree $m+1$. In [56] Kosters shows that for any corner it is possible to construct a one-parameter family of C^m blendings using patches of degree $2^{d-1}m$. The equation of the central patch of the blending surface is found by recursively computing cylinders, where the depth of the recursion is the dimension of the corner. In [57] Kosters extends

the potential method to find C^k blends for arbitrary simple corners by connecting cylinders to a surface using a composition property that constructs higher dimensional blends from lower dimensional blends. Degree $k + 1$ blends are used for simple convex corners and degree $2k$ blends are used for simple non-convex corners. Non-simple corners can be reduced to simple corners in a higher dimensional space, giving a higher degree.

1.3. Interactive Editing of Surfaces

1.3.1. Physically Based Modeling Recently there has been work toward using physically based models as a tool to explore the phenomenon of "natural" motion. Research includes methods for representing deformable surfaces, deformation operations upon surfaces, and how these surfaces interact with an environment acting to deform them. A deformation operation is a geometric operation that takes a set of points and moves them in some manner, such as a warp or bend operation, regardless of any physical relationship.

A typical representation of a deformable surface uses a grid of points where the points are allowed to move in relation to one another. The manner in which the points are allowed to move determines the properties of the deformable surface. One approach is a point mass-spring-hinge model. The points represent the often used abstraction called the *point mass*. The point mass has no size but does have a finite mass. Other characteristics of points include position and velocity. The spring model defines a distance relationship between two points. This relationship is the force that keeps the model together. The hinge model defines an orientation relationship between four points. Hinges maintain angular relationships and thus torques are involved.

Terzopoulos, Platt, Barr, Barzel, Fleischer, and Witkin [101, 78, 79, 109] have presented discrete models which are based extensively on the theory of elasticity and plasticity and use energy fields to define and enforce constraints. Haumann [47] used the same approach but used a triangularized model and a simpler physical model based on points, springs, and hinges. Work on deformable models not based on physical modeling has been done by Barr, Cobb, and Sederberg. Barr [16] presented a model for use with solid primitives using a functional map from one space into another to define the deformation. Sederberg and Parry [92] extended this

model to use Bernstein polynomials. Cobb [25] defined a set of deformation modeling operations for use with the B-spline surface representation which were designed to modify groups of points in a correct and intuitive way.

Thingvold and Cohen [103] defined a model of elastic and plastic B-spline surfaces which supports both animation and design operations. The basis for the physical model is a generalized point mass-spring-hinge model which has been adapted into simultaneous refinement of the geometric/physical model. Always having a sculptured surface representation as well as the physical hinge/spring/mesh model allows the user to intertwine physical based operations, such as force application, with geometrical modeling. Refinement operations for spring and hinge B-spline models are compatible with the physics and mathematics of B-spline models. The models of elasticity and plasticity are written in terms of springs and hinges, and can be implemented with standard integration techniques to model realistic motions of elastic and plastic surfaces. These motions are controlled by the physical properties assigned and by kinematic constraints on various portions of the surface.

Terzopoulos and Qin [102] develop a dynamic generalization of the nonuniform rational B-spline (NURBS) model. They present a physics-based model that incorporates mass distributions, internal deformation energies, and other physical quantities into the NURBS geometric substrate. A modeler can interactively sculpt curves and surfaces and design complex shapes by adjusting control points and weights or through direct physical manipulation by applying simulated forces and local and global shape constraints. The dynamic behavior results from the numerical integration of a set of nonlinear differential equations that automatically evolve the control points and weights in response to the applied forces and constraints. The equations are derived from Lagrangian mechanics and a finite-element-line discretization. The interactive response of the dynamic NURBS may be modified by varying its mass and damping distributions. Global design requirements may also be achieved by varying physical parameters such as elastic energies. These dynamic NURBS can be used in applications such as rounding of solids, optimal surface fitting to unstructured data, surface design from cross-sections, and free-form deformations.

Qin and Terzopoulos [82] develop a dynamic freeform surface

model based on swung NURBS surfaces which is useful for representing objects with symmetries and topological variability. NURBS swung surfaces are formed by swinging one planar NURBS profile curve along a second NURBS trajectory curve. Applications for these dynamic NURBS swung surfaces include interactive sculpting through the imposition of forces, the adjustment of physical parameters such as mass, damping, and elasticity, and surface design with geometric and physical constraints. The equations for motion are derived using the Lagrangian mechanics of an elastic surface and the finite element method. These swung NURBS surfaces are a special case of the dynamic NURBS surfaces in [102].

1.3.2. Hierarchical Splines Hierarchical splines are a multi-resolution approach to the representation and manipulation of free-form surfaces. A hierarchal B-spline is constructed from a base surface (level 0) and a series of overlays are derived from the immediate parent in the hierarchy. Forsey [37] presents a refinement scheme that uses a hierarchy of rectangular B-spline overlays to produce C^2 surfaces. Overlays can be added manually to add detail to the surface, and local or global changes to the surface can be made by manipulating control points at different levels.

Forsey and Wang [38] create hierarchical bicubic B-spline approximations to scanned cylindrical data. The resulting hierarchical spline surface is interactively modifiable using editing capabilities of the hierarchical surface representation allowing either local or global changes to surface shape while retaining the details of the scanned data. However oscillations occur when the data has high-amplitude or high-frequency regions.

1.3.3. Variational Surfaces The most basic goal for interactive free-form surface design is to make it easy for the user to control the shape of the surface. The traditional approach is to search for the right surface representation whose degrees of freedom are sufficient controls for direct manipulation by the user. Using a control mesh to manipulate a surface is a good way to make local modifications, but the user usually needs to modify many control points to make a simple global change. This sort of problem is bound to arise whenever the controls provided to the user are tied closely to the representation's degrees of freedom. Variational surfaces allow the

user to specify controls that aren't related to the representation.

Variational constrained optimization plays a central role in the formulation of natural splines. Natural splines minimize the integral of the second derivative squared subject to the interpolation constraints. Surface models based on variational principles have been widely used in computer vision to solve surface reconstruction problems. Similar formulations have been employed in computer graphics for physically based modeling of deformable surfaces. These methods are based on regular finite difference grids of fixed resolution. Constrained optimization based on second-derivative norms has been used in fairing B-spline surfaces. Moreton [68] minimizes variation of curvature to generate surfaces which skin networks of curves while seeking circular or straight line cross-sections. The surfaces are specified through interpolated geometric constraints consisting of positions and optionally surface normals and curvatures. Nonlinear optimization techniques are then used to minimize a fairness functional based on the variation of curvature. The curves of the network are represented by quintic Hermite polynomial segments and biquintic patches are used for the interpolatory surface. Such schemes can give rise to very fair surfaces but the nonlinearity of their fairness metrics prevents them from being used for interactive surface design.

Celniker and Gossard [22] proposed a physically based model for interactive free-form surface design in which the surface is modeled using a C^1 mesh of triangular patches, and positions and normals may be controlled along patch boundaries. Interactivity is possible because the surface fairing problem is formulated as a minimization of a quadratic functional subject to linear constraints.

Welch and Witkin [108] use an approach that is closely related to that of Celniker and Gossard, however they consider more general formulations for both surface objective functionals and shape control constraints. Their goal is to present the user with a infinitely malleable piecewise smooth surface with no fixed control or structure of its own. The user interactively manipulates variational curves and surfaces, controlling and combining them through a variety of constraints and objective functions. The user may attach a variety of features to the surface such as points and flexible curves, which then serve as handles for direct interactive manipulation of the surface. Within the constraints imposed by these controls surface behavior is governed by one or more simply expressed criteria such as the

surface should be as smooth as possible or conform as closely as possible to a prototype. Formally the surfaces are specified as the solutions to constrained variational optimization problems. They seek shapes that extremize a variety of surface integrals subject to linear geometric constraints. Constraints that are linear in the surface control vector are used since this leads to a constrained optimization problem which can be solved at interactive speeds. The surface approximation formulated as a quadratic objective function with linear constraints can be solved using Lagrange multipliers to enforce a least-squares fit to the constraint matrix or with a penalty-based approach that associates a penalty term with each constraint. Such shapes are not intrinsically linked to any particular surface representation scheme, and the surface representation they use is piecewise polynomial nonuniform B-spline. The nonuniform surface is a simple sum of sparse uniform surface layers which may overlap in arbitrary ways. The resulting surface shape remains a linear function of the control points. The modeler must construct acceptable approximations to such infinite-dimensional variational surfaces using a finite number of control parameters. Automatic subdivision is used to ensure that constraints are met and to enforce error bounds.

References

- [1] P. Alfeld. A trivariate clough-tocher scheme for tetrahedral data. *Computer Aided Geometric Design*, 1:169-181, 1984.
- [2] P. Alfeld. Scattered Data Interpolation in Three or More Variables. In T. Lyche and L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design*, pages 1-34. Academic Press, 1989.
- [3] S. Auerbach, R. H. J. Gmelig Meyling, M. Neamtu, and H. Schachben. Approximation and geometric modeling with simplex B-splines associated with irregular triangles. *Computer Aided Geometric Design*, 8:67-87, 1991.
- [4] C. Bajaj. Geometric modeling with algebraic surfaces. In D. Handscomb, editor, *The Mathematics of Surfaces III*, pages 3-48. Oxford Univ. Press, 1988.
- [5] C. Bajaj. Surface fitting with implicit algebraic surface patches. In H. Hagen, editor, *Topics in Surface Modeling*, pages 23 - 52. SIAM Publications, 1992.
- [6] C. Bajaj. The Emergence of Algebraic Curves and Surfaces in

- Geometric Design. In R. Martin, editor, *Directions in Geometric Computing*, pages 1 – 29. Information Geometers Press, 1993.
- [7] C. Bajaj, F. Bernardini, and G. Xu. *Adaptive Reconstruction of Surfaces and Surfaces-on-Surfaces from Dense Scattered Trivariate Data*. Computer Science Technical Report, CS-95-028, Purdue University, 1994.
 - [8] C. Bajaj, F. Bernardini, and G. Xu. Reconstruction of Surfaces and Surfaces-on-Surfaces from Unorganized Weighted Points. *Computer Graphics*, 29:x-y, 1995.
 - [9] C. Bajaj, J. Chen, and G. Xu. Modeling with cubic a-patches. *ACM Transactions on Graphics*, April 1995, To appear.
 - [10] C. Bajaj and I. Ihm. *C^1 Smoothing of Polyhedra with Implicit Surface Patches*. Technical report, Purdue University, 1991. Computer Science Technical Report, CAPO-91-27.
 - [11] C. Bajaj and I. Ihm. Algebraic Surface Design with Hermite Interpolation. *ACM Transactions on Graphics*, 11(1):61-91, 1992.
 - [12] C. Bajaj, I. Ihm, and J. Warren. Higher-order interpolation and least-squares approximation using implicit algebraic surfaces. *ACM Transactions on Graphics*, 12(4):327-347, 1993.
 - [13] C. Bajaj and G. Xu. Modeling Scattered Function Data on Curved Surface. In J. Chen, N. Thalmann, Z. Tang, and D. Thalmann, editor, *Fundamentals of Computer Graphics*, pages 19 – 29, Beijing, China, 1994.
 - [14] R. E. Barnhill. Smooth interpolation over triangles. In Barnhill and Riesenfeld, editors, *Computer Aided Geometric Design*, pages 45-70. Academic Press, 1974.
 - [15] R. E. Barnhill, K. Opitz, and H. Pottmann. Fat surfaces: a trivariate approach to triangle-based interpolation on surfaces. *Computer Aided Geometric Design*, 9:365-378, 1992.
 - [16] A. Barr. Dynamic constraints for modeling, 1986. Siggraph Course Notes #23.
 - [17] W. Böhm, G. Farin, and J. Kahmann. A survey of curve and surface methods in CAGD. *Computer Aided Geometric Design*, 1:1-60, 1984.
 - [18] J. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3):235-256, 1983.
 - [19] W. Boehm. Calculating with box splines. *Computer Aided Geometric Design*, 1:149-162, 1984.
 - [20] W. Boehm. Multivariate spline algorithms. In J. Gregory, editor, *The Mathematics of Surfaces*. Oxford University Press, 1986.
 - [21] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, 10(6):350-355, 1978.
 - [22] G. Celniker and D. Gossard. Deformable Curve and Surface Finite-Elements for Free-Form Shape Design. *Computer Graphics*,

- 25(4):257-266, July 1991.
- [23] Peter Charrot and John A. Gregory. A pentagonal surface patch for computer aided geometric design. *Computer Aided Geometric Design*, 1(00):87-94, 1984.
 - [24] H. Chiyokura and K. Fumihiko. Design of solids with free form surfaces. *Computer Graphics*, 17(3):289-298, 1983.
 - [25] E. S. Cobb. *Design of Sculptered Surfaces using the B-spline Representation*. PhD thesis, University of Utah, June 1984.
 - [26] E. Cohen, T. Lyche, and R. Riesenfeld. Discrete box splines and refinement algorithms. *Computer Aided Geometric Design*, 1:131-148, 1984.
 - [27] W. Dahmen. Smooth piecewise quadratic surfaces. In T. Lyche and L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design*, pages 181-193. Academic Press, Boston, 1989.
 - [28] W. Dahmen and C. Micchelli. Subdivision algorithms for the generation of box spline surfaces. *Computer Aided Geometric Design*, 1:115-129, 1984.
 - [29] W. Dahmen, C. Micchelli, and H. Seidel. Blossoming Begets B-Spline Bases Built Better By B-Patches. *Mathematics of Computation*, 59(199):97-115, July 1992.
 - [30] W. Dahmen and C. Michelli. On the linear independence of multivariate b-splines i. triangulations of simplexes. *SIAM Journal on Numerical Analysis*, 19:993-1012, 1982.
 - [31] W. Dahmen and C. Michelli. Recent progress in multivariate splines. In L. Shumaker, C. Chui, and J. Word, editors, *Approximation Theory IV*, pages 27-121. Academic Press, 1983.
 - [32] D. Doo and M. Sabin. Behaviour of recursive division surfaces near extraordinary points. *Computer Aided Design*, 10(6):356-360, 1978.
 - [33] G. Farin. Triangular Bernstein-Bézier Patches. *Computer Aided Geometric Design*, 3(00):83-127, 1986.
 - [34] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Academic Press Inc., 1993.
 - [35] P. Fong and H-P. Seidel. An implementation of multivariate b-spline surfaces over arbitrary triangulations. In *Proceedings of Graphics Interface'92*, pages 1-10, Vancouver, B.C., 1992.
 - [36] Peter Fong and Hans-Peter Seidel. Control Points for Multivariate B-Spline Surfaces over Arbitrary Triangulations. *Computer Graphics Forum*, 10(00):309-317, 1986.
 - [37] D. Forsey and R. Bartels. Hierarchical B-spline refinement. *Computer Graphics*, 22(4):205-212, August 1988.
 - [38] D. Forsey and Lifeng Wang. Multi-resolution Surface Approximation for Animation. In *Proc. Graphics Interface*, pages 192-199, Toronto, May 1993.
 - [39] P. O. Frederickson. Triangular spline interpolation. Technical report,

- Lakehead University, 1970. Mathematics Report 6-70.
- [40] W. J. Gordon. Blending function methods of bivariate and multivariate interpolation and approximation. *SIAM J. Numerical Analysis*, 8:158-177, 1977.
 - [41] T. Grandine. The Stable Evaluation of Multivariate Simplex Splines. *Mathematics of Computation*, 50(181):197-205, January 1988.
 - [42] J. A. Gregory. C^1 rectangular and non-rectangular surface patches. In R. E. Barnhill and W. Boehm, editors, *Computer Aided Geometric Design*, pages 25-33. North-Holland, 1983.
 - [43] J. A. Gregory. N-Sided Surface Patches. In J. Gregory, editor, *The Mathematics of Surfaces*, pages 217-232. Oxford University Press, 1986.
 - [44] B. Guo. Surface generation using implicit cubics. In N.M. Patrikalakis, editor, *Scientific Visualization of Physical Phenomena*, pages 485-530. Springer-Verlag, Tokyo, 1991.
 - [45] B. Guo. Non-splitting Macro Patches for Implicit Cubic Spline Surfaces. *Computer Graphics Forum*, 12(3):434 - 445, 1993.
 - [46] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweizer and W. Stuetzle. Piecewise smooth surface reconstruction. *Computer Graphics*, 28:295-302, 1994.
 - [47] D. Haumann. Modeling the physical behavior of flexible objects, 1987. Siggraph Course Notes #17.
 - [48] C. Hoffmann and J. Hopcroft. Quadratic Blending Surfaces. *Computer Aided Design*, 18(00):301-306, 1986.
 - [49] C. Hoffmann and J. Hopcroft. The geometry of projective blending surfaces. *Artificial Intelligence*, 37:357-376, 1987.
 - [50] C. Hoffmann and J. Hopcroft. The Potential Method for Blending Surfaces and Corners. In SIAM G. Farin, editor, *Geometric Modeling: Algorithms and New Trends*, pages 347-366. 1987.
 - [51] K. Hollig. Multivariate splines. *SIAM Journal on Numerical Analysis*, 19:1013-1031, 1982.
 - [52] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. *Computer Graphics*, 27(2):19-26, 1992.
 - [53] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Piecewise smooth surface reconstruction. *Computer Graphics*, 28(2):295-302, 1992.
 - [54] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 26(2):71-78, 1992.
 - [55] M. Hosaka and K. Fumihiko. Non-four-sided patch expressions with control points. *Computer Aided Geometric Design*, 1:75-86, 1984.
 - [56] M.T. Kisters. An extension of the potential method to higher-order blendings. In *Proc. ACM Symposium on Solid Modeling and CAD/CAM Applications*, pages 329-337, Austin, TX, June 1991.

- [57] M.T. Kisters. High-order implicit blending surfaces of low degree. *Computer Aided Geometric Design*, 8:495-512, 1991.
- [58] L. Holmström. Piecewise quadric blending of implicitly defined surfaces. *Computer Aided Geometric Design*, 4:171-189, 1987.
- [59] R. A. Liming. *Practical Analytic Geometry with Applications to Aircraft*. Macmillan, New York, 1944.
- [60] S. Lodha. *Surface approximation by low degree patches with multiple representations*. PhD thesis, Purdue University, West Lafayette, IN, 1992.
- [61] C. Loop. Smooth Subdivision based on triangles. Master's thesis, University of Utah, 1987.
- [62] C. Loop. A G^1 triangular spline of arbitrary topological type. *Computer Aided Geometric Design*, 11:303-330, 1994.
- [63] C. Loop and T. DeRose. Generalized B-spline Surfaces of Arbitrary Topology. *Computer Graphics*, 24(4):347-356, August 1990.
- [64] C. Loop and T. DeRose. A Multisided Generalization of Bézier Surfaces. *ACM Transactions on Graphics*, 8(3):205-234, July 1989.
- [65] S. Mann, C. Loop, M. Lounsbery, D. Meyers, J. Painter, T. DeRose, and K. Sloan. A survey of parametric scattered data fitting using triangular interpolants. In H. Hagen, editor, *Curve and Surface Modeling*, pages 145-172. SIAM, Philadelphia, PA, 1992.
- [66] A. E. Middleditch and K. H. Sears. Blend surfaces for set theoretic volume modelling systems. *Computer Graphics*, 19:161-170, 1985.
- [67] D. Moore and J. Warren. Approximation of dense scattered data using algebraic surfaces. In *Proc. of the 24th Hawaii Intl. Conference on System Sciences*, pages 681-690, Kauai, Hawaii, 1991.
- [68] H. Moreton and C. Séquin. Functional Optimization for Fair Surface Design. *Computer Graphics*, 26(2):167-176, 1992.
- [69] A. H. Nasri. Surface interpolation on irregular networks with normal conditions. *Computer Aided Geometric Design*, 8:89-96, 1991.
- [70] G. Nielson, T. Foley, B. Hamann, and D. Lane. Visualizing and Modeling Scattered Multivariate Data. *IEEE Computer Graphics And Applications*, 11:47-55, 1991.
- [71] J.C. Owen and A.P. Rockwood. Blending surfaces in solid modeling. In G. Farin, editor, *Geometric Modeling : Algorithms and New Trends*, pages 367-383. SIAM, Philadelphia, 1987.
- [72] N. M. Patrikalakis and G.A. Kriezis. Representation of piecewise continuous algebraic surfaces in terms of b-splines. *The Visual Computer*, 5:360-374, 1989.
- [73] Jörg Peters. Local cubic and bicubic C^1 surface interpolation with linearly varying boundary normal. *Computer Aided Geometric Design*, 7:499-516, 1990.
- [74] Jörg Peters. Smooth Interpolation of a Mesh of Curves. "Constructive Approximation", 7:221-246, 1991.

- [75] Jörg Peters. Smooth free-form surfaces over irregular meshes generalizing quadratic splines. *Computer Aided Geometric Design*, 10:347–361, 1993.
- [76] Jörg Peters. Smooth Mesh Interpolation with Cubic Patches. *Computer Aided Design*, 22(2):109–120, March 1990.
- [77] B. Piper. Visually smooth interpolation with triangular bezier patches. In G. Farin, editor, *Geometric Modeling: Algorithms and New Trends*, pages 221–234. SIAM, Philadelphia, PA, 1987.
- [78] J. Platt and A. Barr. Constraint methods for flexible models. In *Siggraph Proceedings*, pages 279–288, August 1988.
- [79] J. Platt, D. Terzopoulos, and A. Barr. Elastically deformable models. In *Siggraph Proceedings*, pages 205–214, August 1987.
- [80] M. Powell and M. Sabin. Piecewise quadratic approximations on triangles. *ACM Trans. on Math. Software*, 3:316–325, 1977.
- [81] M. Pratt. Parametric curves and surfaces as used in computer aided design. In J. Gregory, editor, *The Mathematics of Surfaces*, pages 1–18. Oxford University Press, 1986.
- [82] H. Qin and D. Terzopoulos. Dynamic nurbs swung surfaces for physics-based shape design. *Computer Aided Design*, 27(2):111–127, 1995.
- [83] U. Reif. A unified approach to subdivision algorithms. Technical report, Mathematisches Institut A, Universität Stuttgart, 1992. Preprint 92-16.
- [84] U. Reif. Biquadratic g-spline surfaces. Technical report, Mathematisches Institut A, Universität Stuttgart, 1993. Preprint 93-4.
- [85] K. L. Rescorla. C^1 Trivariate Polynomial Interpolation. *Computer Aided Geometric Design*, 4:237–244, 1987.
- [86] M. A. Sabin. *The use of piecewise forms for the numerical representation of shape*. PhD thesis, Hung. Acad. Sciences Tanulmányok 60, 1977.
- [87] M. A. Sabin. Non-Rectangular Surface Patches Suitable for Inclusion in a B-Spline Surface. In P.J.W. ten Hagen, editor, *Eurographics '83*, pages 57–69. Eurographics Association, 1983.
- [88] P. Sablonniere. *Bases de Bernstein et approximations splines*. PhD thesis, Université de Lille, 1982.
- [89] R. Sarraga. G^1 interpolation of generally unrestricted cubic Bézier curves. *Computer Aided Geometric Design*, 4:23–39, 1987.
- [90] R. Schmidt. Eine Methode zur Konstruktion von C^1 -Flächen zur Interpolation unregelmässig verteilter Daten. In Schempp and Zeller, editors, *Multivariate Approximation II*, pages 343–361. Birkhäuser, Basel, 1982.
- [91] L. Schumaker. Applications of multivariate splines. In *Mathematics of computation, 1949–1999: a half-century of computations mathematics*. American Mathematical Society, Providence, RI, 1994. *Proceedings of Symposia in Applied Mathematics, Volume 48*.

- [92] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. *Computer Graphics*, 20(4):151–160, 1986.
- [93] T.W. Sederberg. Piecewise algebraic patches. *Computer Aided Geometric Design*, 2:53–59, 1985.
- [94] T.W. Sederberg. Techniques for cubic algebraic surfaces, tutorial part i. *IEEE Computer Graphics and Applications*, 10(4):14–25, July 1990.
- [95] T.W. Sederberg. Techniques for cubic algebraic surfaces, tutorial part ii. *IEEE Computer Graphics and Applications*, 10(5):12–21, Sept. 1990.
- [96] H. Seidel. A new multiaffine approach to B-splines. *Computer Aided Geometric Design*, 6:23–32, 1989.
- [97] H. Seidel. Algorithms for B-patches. In *Proc. Graphics Interface*, 1991.
- [98] H. Seidel. Symmetric recursive algorithms for surfaces: B-Patches and the de Boor algorithm for polynomials over triangles. *Constructive Approximation*, 7(2):257–279, 1991.
- [99] L. Shirman and C. Séquin. Local surface interpolation with bezier patches. *Computer Aided Geometric Design*, 4:279–296, 1988.
- [100] D.J.T. Story and A.A. Ball. Design of an n -sided surface patch from Hermite boundary data. *Computer Aided Geometric Design*, 6:111–120, 1989.
- [101] D. Terzopolos and K. Fleischer. Modeling inelastic deformation: viscoelasticity, plasticity, fracture. In *Siggraph Proceedings*, pages 269–278, August 1988.
- [102] D. Terzopolos and H. Qin. Dynamic nurbs with geometric constraints for interactive sculpting. *ACM Transactions on Graphics*, 13(2):103–136, April 1994.
- [103] J. Thingvold and E. Cohen. Physical modeling with b-spline surfaces for interactive design and animation. *Computer Graphics*, 24(2):129–137, 1990.
- [104] J. van Wijk. Bicubic patches for approximation nonrectangular control-point meshes. *Computer Aided Geometric Design*, 3:1–13, 1986.
- [105] T. Varady. Overlap patches: a new scheme for interpolation curve networks with n -sided regions. *Computer Aided Geometric Design*, 8:7–27, 1991.
- [106] J. Warren. Blending algebraic surfaces. *ACM Transactions on Graphics*, 8(4):263–278, 1989.
- [107] J. Warren. Creating Multisided Rational Bezier Surfaces Using Base Points. *ACM Transactions on Graphics*, 11(2):127–139, 1992.
- [108] W. Welch and A. Witkin. Variational Surface Modeling. *Computer Graphics*, 26(2):157–166, 1992.
- [109] A. Witkin, K. Fleischer, and A. Barr. Energy constraints on

- parameterized models. In *Siggraph Proceedings*, pages 225–232, July 1987.
- [110] A. J. Worsey and G. Farin. An n-dimensional clough-tocher interpolant. *Constructive Approximation*, 3:99–110, 1987.
 - [111] G. Wyvill, C. McPheeters, and B. Wyvill. Data structure for *soft* objects. *The Visual Computer*, 2:227–234, 1986.