RICE UNIVERSITY

Surface Approximation By Low Degree Patches With Multiple Representations

by

Suresh Kumar Lodha

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Approved, Thesis Committee:

Joe Warren, Director
Associate Professor of Computer Science

Ronald Goldman
Professor of Computer Science

Mark W. Krentel
Assistant Professor of Computer Science

Peter Varman
Associate Professor of Electrical &
Computer Engineering

Houston, Texas

August, 1992

Surface Approximation By Low Degree Patches With Multiple Representations

Suresh Kumar Lodha

Abstract

Computer Aided Geometric Design (CAGD) is concerned with the representation and approximation of curves and surfaces when these objects have to be processed by a computer. Parametric representations are very popular because they allow considerable flexibility for shaping and design. Implicit representations are convenient for determining whether a point is inside, outside or on the surface. These representations offer many complimentary advantages. Therefore, it is desirable to build geometric models with surfaces which have both parametric and implicit representations. Maintaining the degree of the surfaces low is important for practical reasons. Both the size of the surface representation, as well as the difficulties encountered in the algorithms, e.g. root finding algorithms, grow quickly with increasing degree.

This thesis introduces low degree surfaces with both parametric and implicit representations and investigates their properties. A new method is described for creating quadratic triangular Bézier surface patches which lie on implicit quadric surfaces. Another method is described for creating biquadratic tensor product Bézier surface patches which lie on implicit cubic surfaces. The resulting surface patches satisfy all of the standard properties of parametric Bézier surfaces, including interpolation of the corners of the control polyhedron and the convex hull property.

The second half of this work describes a scheme for filling n-sided holes and for approximating the resulting smooth surface consisting of high degree parametric Bézier surface patches by a continuous surface consisting of low degree patches with both parametric and implicit representations. A new technique is described for filling an n-sided hole smoothly using a single parametric surface patch with a geometrically intuitive compact representation. Next, a new degree reduction algorithm is applied to approximate high degree parametric Bézier surfaces by low degree Bézier surfaces. Finally, a variant of the least squares technique is used to approximate parametric

Bézier surfaces of low degree by low degree surfaces with both parametric and implicit representations. The resulting surfaces have boundary continuity and approximation properties.

Computer Aided Geometric Design (CAGD) is concerned with the representation and approximation of curves and surfaces when these objects have to be processed by a computer. Parametric representations are very popular because they allow considerable flexibility for shaping and design. Implicit representations are convenient for determining whether a point is inside, outside or on the surface. These representations offer many complimentary advantages. Therefore, it is desirable to build geometric models with surfaces which have both parametric and implicit representations. Maintaining the degree of the surfaces low is important for practical reasons. Both the size of the surface representation, as well as the difficulties encountered in the algorithms, e.g. root finding algorithms, grow quickly with increasing degree.

This thesis introduces low degree surfaces with both parametric and implicit representations and investigates their properties. A new method is described for creating quadratic triangular Bézier surface patches which lie on implicit quadric surfaces. Another method is described for creating biquadratic tensor product Bézier surface patches which lie on implicit cubic surfaces. The resulting surface patches satisfy all of the standard properties of parametric Bézier surfaces, including interpolation of the corners of the control polyhedron and the convex hull property.

The second half of this work describes a scheme for filling n-sided holes and for approximating the resulting smooth surface consisting of high degree parametric Bézier surface patches by a continuous surface consisting of low degree patches with both parametric and implicit representations. A new technique is described for filling an n-sided hole smoothly using a single parametric surface patch with a geometrically intuitive compact representation. Next, a new degree reduction algorithm is applied to approximate high degree parametric Bézier surfaces by low degree Bézier surfaces. Finally, a variant of the least squares technique is used to approximate parametric Bézier surfaces of low degree by low degree surfaces with both parametric and implicit representations. The resulting surfaces have boundary continuity and approximation properties.

Computer Aided Geometric Design (CAGD) is concerned with the representation and approximation of curves and surfaces when these objects have to be processed by a computer. Parametric representations are very popular because they allow considerable flexibility for shaping and design. Implicit representations are convenient

for determining whether a point is inside, outside or on the surface. These representations offer many complimentary advantages. Therefore, it is desirable to build geometric models with surfaces which have both parametric and implicit representations. Maintaining the degree of the surfaces low is important for practical reasons. Both the size of the surface representation, as well as the difficulties encountered in the algorithms, e.g. root finding algorithms, grow quickly with increasing degree.

This thesis introduces low degree surfaces with both parametric and implicit representations and investigates their properties. A new method is described for creating quadratic triangular Bézier surface patches which lie on implicit quadric surfaces. Another method is described for creating biquadratic tensor product Bézier surface patches which lie on implicit cubic surfaces. The resulting surface patches satisfy all of the standard properties of parametric Bézier surfaces, including interpolation of the corners of the control polyhedron and the convex hull property.

The second half of this work describes a scheme for filling n-sided holes and for approximating the resulting smooth surface consisting of high degree parametric Bézier surface patches by a continuous surface consisting of low degree patches with both parametric and implicit representations. A new technique is described for filling an n-sided hole smoothly using a single parametric surface patch with a geometrically intuitive compact representation. Next, a new degree reduction algorithm is applied to approximate high degree parametric Bézier surfaces by low degree Bézier surfaces. Finally, a variant of the least squares technique is used to approximate parametric Bézier surfaces of low degree by low degree surfaces with both parametric and implicit representations. The resulting surfaces have boundary continuity and approximation properties.

Acknowledgements

I would like to thank my thesis advisor, Joe warren, for his constant encourgament and support. I am very grateful to Ron Goldman for treating my research with great care and enthusiasm. Thanks go to Jarek Rossignac, Mark Krentel, Peter varman and Alex Schaffer for their interest in this work. Special thanks go to Doug Moore for helping me with all odd problems at all times. I would also like to thank Ivy Jorgensen for proofreading very enthusiastically parts of this work, which were sent for publication.

I am deeply indebted to my grandmother for the love she gave me. I am deeply indebted to my mother for the spiritual education and to my father for the ambition, he inculcated in me. I am grateful to my brothers, sister-in-laws, nephews and nieces for giving me joy and most of all to my wife, Madhu, for the perseverant faith in me, without which this thesis would not have been even started. My special gratitude is reserved for my one-and- a-half year old son, Chandan, whose birth and early childhood play has always given me the hope and the will to struggle to move on even in the most difficult times. Without the emotional support of my family, this thesis would never have been completed.

My appreciation goes to my many friends including Urmil, Jagannath, Shridhar and Subhash, who have made my life rich and work meaningful. Finally, I express my deep regards to Shing-Shen Chern, a wonderful teacher, who taught me creative research.

Contents

Illustrations

Chapter 1

Introduction

1.1 Background

A fundamental question in Computer Aided Geometric Design (CAGD) is how to represent and manipulate curves and surfaces in computers. The ease or the complexity of the algorithms for surface manipulation depends crucially upon the underlying representation. Currently, the two major surface representations, parametric and implicit, offer many complimentary advantages. Parametric representations are described by a map, usually a polynomial map, from a domain in R^2 into R^3. Implicit representations are described as a zero contour of a function, usually a polynomial function. Parametric representations allow rapid generation of a large number of points on the surface for computer graphics applications. They are very popular because they allow considerable flexibility for shaping and design. Implicit representations are very convenient in determining whether a point is inside, outside or on the surface for solid modeling applications. They are also convenient for defining blends, offsets and boolean operations for manufacturing applications.

Unfortunately, conversion between the parametric and the implicit representation is in general a complex and expensive process; thus conversion is impractical in most cases. Typically modelers have chosen one representation or the other, but not both, depending upon the domain of application. However, many application domains could use the advantages of both representations. Therefore, it is desirable to build geometric models with surfaces which have both parametric and implicit representations.

Another important choice in representing surfaces is the degree of the polynomials in parametric or implicit representations. Although higher degree polynomial representations allow approximation of any continuous surface to arbitrary precision, the difficulties encountered in the algorithms, e.g. root finding algorithms, grow quickly with increasing degree [HH86]. Low degree surface representations result in faster computations for subsequent operations such as computer graphics display, animation, and simulation [BDH+88]. Low degree surface representations also facilitate computationally complex interrogation problems such as intersection and blending. Moreover, low degree surfaces allow efficient, numerically stable and reliable implementation of several algorithms including Boolean combinations of volumes and interference preprocessing procedures [PK89]. Finally, low degree surfaces offer the possibility of implementing the algorithms for manipulating them in computer hardware, thus speeding the computation significantly. Therefore, throughout this work, surface representations of low degree are emphasized.

This thesis introduces new surface patches that have both parametric and implicit representations of low degree. The two representations of the surfaces are related to each other in a simple manner. Each representation can be computed conveniently from the other using simple arithmetic operations avoiding the general conversion process. This is achieved using a new geometric construction. Specifically, a new method is described for creating triangular Bézier surface patches with parametric and implicit degree two. This construction includes surfaces such as spheres, cones and cylinders, which are fundamental entities in CAGD [RV83]. Another method is described for creating biquadratic rectangular Bézier surface patches with implicit degree three. The resulting surface patches satisfy all of the standard properties of parametric Bézier surfaces, including interpolation of the corners of the control polyhedron and the convex hull property.

The second half of this work describes a scheme for filling n-sided holes, which are surrounded by high degree parametric Bézier patches. The resulting smooth surface is then approximated by a continuous surface consisting of low degree patches with both parametric and implicit representations. First, a new technique is described to fill an n-sided hole smoothly using a single parametric surface patch with a geometrically intuitive compact representation. Next, a new degree reduction algorithm is applied to approximate high degree parametric Bézier surfaces by low degree Bézier surfaces. Finally, a variant of the least squares technique is used to approximate parametric Bézier surfaces of low degree by low degree surfaces with both parametric and implicit representations. The resulting surfaces have boundary continuity and approximation properties.

The rest of the introduction describes the difficulties in converting between the parametric and implicit representations and reviews the previous work on the problem of filling n-sided holes and the problem of surface approximation.

1.2 Surface Representations

This section introduces the representation of surfaces used in CAGD. Currently, there are two major techniques for representing surfaces: parametric and implicit.

1.2.1 Parametric Representation

Surfaces with parametric representation are described by a map from a (u, v) domain in R^2 into R^3 as follows:

$$x_1 = f_1(u, v), \quad x_2 = f_2(u, v), \quad x_3 = f_3(u, v).$$

A surface patch, or a bounded portion of a surface, with parametric representation is defined as a surface with parametric representation on a bounded domain, usually on a triangular or a rectangular domain. Thus, for example, a rectangular surface patch with parametric representation can be defined by a map from a (u, v) domain into R^3 by imposing the rectangular bounds $0 \leq u, v \leq 1$ on the parameters u and v. Similarly, a triangular surface patch with parametric representation is defined by a map from a (u, v) domain into R^3 by imposing the triangular bounds $0 \leq u, v$ and $u + v \leq 1$ on the parameters u and v.

In geometric modeling, usually defining functions f_i are chosen to be polynomials, since they are easy to evaluate and can approximate any continuous function to arbitrary precision. Often the defining functions are chosen to be piecewise polynomials or rationals (ratios of polynomials) to allow the additional flexibility in designing the surfaces. The parametric degree of a surface patch is defined to be the maximum of the degree of these defining polynomials. A triangular surface patch with parametric degree two is referred to as a quadratic surface patch.

There are many advantages of parametric representations. Parametric surfaces exhibit considerable flexibility for shaping and design. They can be joined together smoothly with various degrees of continuity very conveniently. This is one of the primary reasons for their widespread use in the design of car bodies and ship hulls, where surfaces meeting with the second or the third order of continuity are required. Moreover, local changes in the design of the surface can be made without affecting the rest of the surface. The parametric representation allows rapid generation of a large number of points on the surface for computer graphics display. Finite segments of surfaces or surface patches can easily be defined by imposing the parameter bounds on the domain. Highly desirable geometric properties such as the convex hull property and interpolation at corners can be achieved rather easily. Furthermore, parametrically represented surfaces such as Bézier and B-spline surfaces have very robust evaluation, subdivision, differentiation, and change of basis algorithms [Far86, BFK84, BBB87]. It is for these reasons that parametric surfaces have been traditionally used in free-form surface modeling.

When the defining polynomials f_i are expressed in terms of Bernstein polynomials, the surfaces are referred to as Bézier surfaces. For example, a rectangular or tensor product Bézier surface patch is defined by a polynomial map of bidegree (m, n) from R^2 to R^3, and is represented in terms of Bernstein polynomials with respect to the domain rectangle Δ $(0 \leq u, v \leq 1)$ as follows:

$$p(u,v) = (f_1(u,v), f_2(u,v), f_3(u,v)) = \sum_{i=0}^{m}\sum_{j=0}^{n} C_{ij} B_i^m(u) B_j^n(v)$$

where the coefficients C_{ij} are points in 3-space and the univariate Bernstein polynomials of degree n are defined by

$$B_i^n(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}$$

for $0 \leq u \leq 1$, and $i = 0, \cdots, n$. The Bernstein polynomials of degree n form a basis for the space of all the polynomials of degree n, are nonnegative, and sum to unity. The coefficients C_{ij} are individually referred to as control points and collectively referred to as the control net for p relative to Δ. A rectangular Bézier surface patch with parametric bidegree $(2, 2)$ is referred to as a biquadratic Bézier surface patch.

This work emphasizes parametric Bézier representation for surfaces. The main attraction of Bézier surfaces is that they lend themselves easily to a geometric understanding. Bézier surface patches interpolate the corner control points, and lie in the convex hull of the control net. Moreover, the tangent plane to the Bézier surface patches at the corner control points is the plane determined by the corner control point and the two control points, adjacent to the corner control point.

1.2.2 Implicit Representation

Surfaces with implicit representation are described as the zero contour of a function

$$Q(x_1, x_2, x_3) = 0.$$

Throughout this work, the defining function $Q(x_1, x_2, x_3)$ is chosen to be a polynomial function for several reasons. Mathematically, the theory of implicitly defined polynomial surfaces, also known as algebraic surfaces, is very well understood. Moreover, the space of all polynomial implicit surfaces contains the space of all polynomial or rational parametric surfaces [Hof89].

An implicit surface patch or a bounded portion of a surface can be defined as that portion of an unbounded implicit surface which lies inside a tetrahedron. In other words, an implicit surface patch can also be represented as the zero contour of a trivariate polynomial $Q(x_1, x_2, x_3)$ of degree d over a defining tetrahedron. When the defining polynomial $Q(x_1, x_2, x_3)$ is expressed in terms of Bernstein polynomials, the representation is referred to as *trivariate implicit representation.* For example, if the defining tetrahedron is given by $x_1 + x_2 + x_3 \leq 1$ and $0 \leq x_1, x_2, x_3$, then the trivariate implicit representation of a surface patch is described as follows:

$$Q(x_1, x_2, x_3) = \sum_{\substack{i_1, i_2, i_3, i_4 \geq 0 \\ i_1+i_2+i_3+i_4=d}} \frac{d!}{i_1! i_2! i_3! i_4!} b_{i_1 i_2 i_3 i_4} x_1^{i_1} x_2^{i_2} x_3^{i_3} (1 - x_1 - x_2 - x_3)^{i_4}$$

where $b_{i_1 i_2 i_3 i_4}$ are real numbers.

The implicit degree of the surface is defined to be the degree of the trivariate polynomial $Q(x_1, x_2, x_3)$. A surface with implicit degree two (resp. three or four) is referred to as a quadric (resp. cubic or quartic) surface. The coefficients $b_{i_1 i_2 i_3 i_4}$ in the trivariate implicit representation are referred to as implicit weights and are associated with the domain points $(\frac{i_1}{d}, \frac{i_2}{d}, \frac{i_3}{d}, \frac{i_4}{d})$ of the tetrahedron. The shape of the surface with trivariate implicit representation can be modified interactively by changing the implicit weights. For example, the surface interpolates a vertex of a tetrahedron if and only if the implicit weight associated with the vertex is zero. Similarly, the surface interpolates an edge of the tetrahedron if and only if all the implicit weights associated with the domain points lying on that edge are zero. For these reasons, this work emphasizes trivariate implicit representation.

Recently an increasing interest in implicit surfaces can be observed in CAGD. The quadric surfaces, surfaces of implicit degree two include spheres, cones and cylinders, which are fundamental entities in many solid modeling systems [RV83]. Cubic surfaces have been used for free-form surface modeling by Sederberg [Sed90a, Sed90b]. Surfaces of implicit degree four, such as tori and cyclides, are needed in creating blends in geometric modeling applications [Pra90, Boe90, VCH88]. Implicit surfaces have been used by Blinn [Bli82] to model electron density maps of molecular structures. Wyvill and Wyvill [WMW86] have used implicit surfaces to model and animate natural phenomena such as smoke, clouds, mountains, coastlines, living forms, mud, water, and other artistically interesting shapes including fabrics and cushions.

Dahmen [Dah89] and Guo [Guo90] have described techniques to create free-form geometric models with quadric and cubic surfaces respectively. Bajaj [Baj90] has constructed implicit surfaces to solve the scattered data fitting problem. Bajaj and Ihm [BI92, Baj90] have constructed piecewise continuous quintic implicit surfaces to smooth polyhedra. Patrikalakis and Kriezis [PK89] have also studied piecewise continuous implicit surfaces in terms of B-spline surfaces.

The reason for the increasing popularity of implicit surfaces is that certain tasks in geometric modeling can be solved only with considerable effort when using parametric surface representation, while they are solved relatively easily using the implicit form. An implicit surface has the half-space property, that is, it divides space into two half-spaces, namely $Q(x_1, x_2, x_3) \geq 0$ and $Q(x_1, x_2, x_3) \leq 0$. As a consequence, testing whether a point is inside or outside an implicitly defined object, or on the surface of the object, reduces to a simple substitution of the point into the function and subsequent evaluation. This contrasts with parametric surfaces, for which inside and outside testing is difficult. This ability to classify points is one of the primary reasons for the popularity of implicit surfaces in current solid modelers such as those described by Requicha and Voelcker [RV83]. Another significant advantage of implicit surfaces is their closure under the modeling operations of intersection and offsetting, that is, the offset of an implicit surface is an implicit surface [FN89], and the intersection of two implicit surfaces is an implicit curve [GW89b]. The rational parametric surfaces are not closed under offsetting. Therefore, the construction of an offset surface to a given surface and the construction of blending surfaces, which blend two or more given surfaces, are relatively easier tasks with the implicit polynomial surfaces. Moreover, there exists a number of well-known algorithms, e.g., root-finding, symbolic manipulation, and display algorithms, for manipulating these surfaces [Sed89, Han83, Blo88, Sed83]. Finally, calculating the normal and the curvature at a particular point on a surface is usually easier for implicitly defined surfaces. These computations are important for many solid modeling operations such as finding intersection curves with other surfaces.

1.2.3 Conversion Between Representations

In view of the complimentary advantages of the two surface representations, parametric and implicit, it is desirable to able to convert from one representation to another. The process of converting from a parametric representation to an implicit representation is known as *implicitization*. The process of converting from an implicit representation to a parametric representation is known as *parametrization*.

Implicitization

Using elimination theory and resultants, any rational parametric curve or surface can be implicitized. Sederberg et al [SAG84] implicitized rational plane curves using the Sylvester resultant. Goldman et al [GSA84] succintly implemented implicitization of rational plane curves using a technique called vector elimination. Another implementation using the Bézout resultant, called Cayley's statement of Bézout's method, was discussed by Montaudouin and Tiller [dMT84]. Goldman [Gol85] implicitized cubic rational space curves by the method of resolvents. Du [Du92] generalized this to arbitrary degree rational curves in 3-space.

As with rational plane curves, the method of resultants is an important tool for implicitization in the case of surfaces. Sederberg et al [SAG84] used the Dixon formulation of the resultant to implicitize tensor-product rational parametric surfaces. However, there is a significant difference. This technique fails when there are base points[1] in the domain or excess base points at infinity. Base points of surface parametrizations, which annihilate the resultant, cannot always be removed as in the case of rational plane curves. It turns out that many rational parametrizations have base points. All quadric surfaces, e.g. spheres, cylinders, and cones are rational parametric surfaces with base points. Moreover, Sederberg's method is limited to tensor-product surfaces and cannot be used for implicitizing other parametric surfaces, e.g. triangular surfaces. Manocha and Canny [MC92] describe a technique for implicitization, which requires the symbolic expansion of determinants. For example, the implicit representation of a tensor-product surface of parametric bidegree (m, n) corresponds to the determinant of a matrix of order $2mn$, where each matrix entry is a linear polynomial in x_1, x_2 and x_3. They then use Vandermonde interpolation for computing the coefficients. Nevertheless, this implementation of the implicitization procedure does not work well with real coefficients due to an explosion in the magnitude of the coefficients.

Techniques for implicitizing parametric surfaces with base points are analyzed by Chionh [Chi90] and Manocha and Canny [MC92]. These techniques require perturbation and the introduction of an additional variable, thereby increasing the symbolic

[1] A base point is a parameter value for which the rational parametrization takes on the value $(\frac{0}{0}, \frac{0}{0}, \frac{0}{0})$.

complexity of the resulting expression. The implicitized expression can result in extraneous factors and their separation can be a difficult task [Hof90]. Furthermore, these algorithms are not able to implicitize even low degree parametric surfaces, e.g. bicubic patches, in a reasonable amount of time and space [Hof89].

Another technique of implicitization using Gröbner bases [Buc85] is also fairly expensive in practice even for low degree parametric surfaces. In general, the running time complexity of the Gröbner bases algorithm can be doubly exponential in the number of variables as compared to the singly exponential complexity of the resultants.

Even if the implicitization of a surface is achieved in some cases, there still remains the difficult problem of converting a parametric surface *patch*, a bounded portion of a parametric surface, into an equivalent implicit surface patch, because the bounded portion of the implicit representation may contain self-intersections or extraneous undesirable sheets of surface. These difficulties are described in detail later in Chapter 6.

Finally, the implicit degree is typically much higher than the parametric degree. Thus implicitization defeats the purpose of keeping the degree of the surfaces low. For example, a general tensor product rational rectangular parametric surface of bidegree (m, n) has an implicit degree of $2mn$ [Sed83]. A general triangular rational parametric surface of parametric degree n has an implicit degree of n^2. Thus, the implicit degree of the surface can be very high, even if the parametric degree is low. For example, in a general tensor product parametric biquartic surface has implicit degree 32. Therefore, it is important to find constructions to reduce the implicit degree of parametric patches by imposing certain geometric restrictions on them. Chionh [Chi90] proved that the implicit degree of a parametric surface is reduced by the number of the base points in the parametrization. For example, a triangular quadratic parametric surface without any base points has implicit degree four; but with two base points, it has implicit degree two. Similarly, a biquadratic tensor-product parametric surface without any base points has implicit degree eight, but with five base points, it has implicit degree three. Warren [War90, War91] has imposed base points on the parametrization to reduce the implicit degree of the surfaces. In this work geometric constraints are imposed on the parametric patches to introduce base points to reduce their implicit degree.

Parametrization

All implicit curves of degree two (conic curves) can be rationally parametrized. However, not all polynomial implicit curves and surfaces are rational parametric curves or surfaces. For example, the polynomial implicit curve $X^n + Y^n = 1$ is not rational parametric for $n > 2$ [Har77]. Abhyankar and Bajaj [AB87a, AB87b, AB88, AB87c] studied the parametrization of plane curves of degree two and three, plane curves of general degree, and general degree space curves.

For surfaces, every implicit quadric surface can be rationally parametrized. Every implicit cubic surface, except for cones and cylinders generated by non-rational cubic curves, can be rationally parametrized [SS87]. Even though many quartic surfaces, such as tori and cyclides, can be rationally parametrized, there exists a large class of quartic surfaces, which cannot be rationally parametrized [SR49].

Though parametrization can be achieved for all quadric, most cubic and a large class of quartic surfaces using special techniques from algebraic geometry, the problem of converting an implicit surface *patch*, a bounded portion of an implicit surface, to an equivalent parametric patch still remains. It is not clear, for example, how to convert information on the clipping planes of the implicit surface patch to information on the parameter bounds of the parametric patch.

1.2.4 Multiple Representations

Implicit and parametric representations of surfaces have their own strengths and weaknesses. The advantages of these two representations can be exploited by using surfaces with both representations. An even more significant advantage is provided by intersection problems encountered in surface and solid modeling systems. Intersection is a frequent operation in CAGD which must be performed not only accurately and efficiently, but also in a robust manner [PG85]. These problems are greatly simplified if one of the curves or surfaces can be expressed implicitly and the other parametrically [SP86, Far87]. In this case, the parametric expressions for one surface, $f_1(u,v), f_2(u,v), f_3(u,v)$, can be substituted directly into the implicit equation of the other surface, $Q(x_1, x_2, x_3) = 0$, to yield a single equation $Q(f_1(u,v), f_2(u,v), f_3(u,v)) = 0$, which expresses the curve of intersection implicitly in parameter space.

In view of the preceding discussion, it is not generally practical to represent a surface choosing one of the two representations, relying on general implicitization or parametrization algorithms to obtain the other representation. This thesis introduces surface patches with both representations, trivariate implicit representation and parametric rational Bézier representation, where each representation can be computed from the other conveniently using simple arithmetic operations without relying on the general conversion process. This multiple representation is achieved by imposing certain geometric restrictions on the control net of the parametric Bézier patches. Moreover, both the parametric and the implicit degree of the patches are low. A geometric construction is described to create triangular rational quadratic Bézier surface patches with trivariate implicit representation of degree two. Note that, in general, a triangular rational quadratic Bézier surface patch has implicit degree four. This work also describes a construction to create rectangular biquadratic Bézier surface patches with trivariate implicit representations of degree three. Again, note that, in general, a rectangular rational biquadratic Bézier surface patch has implicit degree eight. By virtue of having both representations, these surface patches inherit the advantages of both representations.

1.3 Filling n-sided Holes

The n-sided hole problem arises when n patches surround a hole. The objective is to construct a surface patch that fills the hole and meets the surrounding surface patches smoothly. We shall restrict our attention to the case where the surrounding patches have polynomial or rational parametric representations. Thus, the problem is to construct a smooth interpolant for a parametrically defined vector-valued function and its derivatives on the boundary of a polygonal domain. These interpolants on a fixed domain generate a surface patch, which then can be fitted together with adjacent patches to create a smooth surface.

The problem of filling an n-sided hole is an important problem and has been studied by various researchers over the last thirty years. In its earliest form, the n-sided hole arose as a problem of filling wire-frame meshes of curves by surface patches. Given a user-defined network of curves and cross-boundary derivatives, a solution to the n-sided hole problem creates a smooth surface which interpolates the user-specified data.

Coons [Coo64] pioneered this study by solving the problem of interpolation to position and derivative information for networks of curves with a rectilinear structure by constructing what are known today as Coons patches. His scheme involves *side-*

side interpolants, which interpolate the data and its derivatives on the two opposite sides of a rectangle. The final solution is created by taking boolean sums of these *lofting* interpolants. These patches are then fitted together over a net of rectangles to form a smooth surface. Gordon [Gor69] recognized the boolean structure of these patches, and Gordon [Gor69] and Forrest [For72b] explicated these methods in detail. Barnhill, Birkhoff, and Gordon [BBG73] extended Coons patches from rectangular domains to standard triangular domains. This construction was then generalized by Little [Bar91] to arbitrary triangular domains via barycentric coordinates.

One of the difficulties associated with this problem is the incompatibility of the cross-derivatives or the twist terms at the corners of the domain. The schemes, described above require that the cross-derivatives or the *twist* terms be compatible at the vertices of the domain. However, in general, one cannot expect the twist terms to be compatible. If the twist terms are incompatible, it is well-known that a parametric polynomial patch cannot fill the hole [Pet90]. Therefore, solutions using parametric rational patches were proposed. Gregory [Gre74] proposed alternative solutions for both rectangular and triangular domains with incompatible twist terms. These schemes involve using rational blending functions to take care of the incompatible twist terms at the corners. Gregory's solution for rectangular domains, known as the Gregory patch, was represented in Bézier form by Chiyokura and Kimura [CK83]. Here interior control points are represented no longer as constants, but as rational combinations of user-specified incompatible control points. We shall refer to this representation as a *rationally controlled Bézier representation*. This representation is compact and very suitable for computation. For example, various algorithms, such as the deCasteljau algorithm for evaluation and subdivision of Bézier surface patches, can also be applied to surface patches with a rationally controlled Bézier representation. Moreover, since this representation is expressed in terms of the user-specified control points via rational blending functions, it provides some geometric intuition for understanding the patch. Nielson [Nie79] proposed an alternative solution for triangular domains by using a *side-vertex* method. Foley [Fol91] noted that the discretized version of Nielson's solution for the cubic triangular case has a rationally controlled Bézier representation. Alternative solutions for the rectangular domain were also proposed by Brown using convex combination methods [Lit83]. Brown's square [Gre83a] is another example of a solution with a rationally controlled Bézier representation. The generalization of a family of Gregory patches proposed by Ueda and Harada [UH91] also have rationally controlled Bézier representations.

Extensions of these methods to pentagonal domains and the underlying difficulties which are encountered have been studied by Gregory, Charrot, and Hahn [CG84, GH89, GH87, Gre83b, GC80]. These schemes, however, are procedural. The resulting surface patches, which fill 5-sided holes, do not have any compact representation in terms of user-specified control points. Hosaka and Kimura [HK84] derived expressions for non-rectangular patches in terms of control points, but in general they are quite complicated. The n-sided patches developed by Sabin [Sab83] are restricted to 5 and 6 sides. Other n-sided patch representations have been proposed by Herron [Her79], Varady [Var91, Var87], and Loop and DeRose [LD90].

Attempts have also been made to generalize these results, when the hole-filling surface patch meets the surrounding surface patches, not just with C^1 continuity as in the above cases, but with higher order of continuity, such as C^2 or C^3. A generalization to smooth interpolants, which match the value of the function and C^2 cross-boundary derivatives for the case of rectangular domains is described by Barnhill [Bar83] and Worsey [Wor84]. Extensions to higher dimensions have also been studied. Interpolation to boundary data on a hypercube has been investigated by Barnhill and Worsey [BW84, Wor85, BS84]; interpolation to boundary data on a simplex is described by Gregory [Gre85].

In Chapter 4 we introduce a new solution in a compact form, very suitable for computation, which works for any number of sides, any number of derivatives, and any number of dimensions. We construct these surface patches in 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. This representation arises naturally in solving the n-sided hole filling problem, and provides good geometric intuition for the proposed solution. The S-patch representation, introduced by Loop and DeRose [LD89], is itself a natural generalization of the Bézier representation. Our construction unifies the myriad solutions, proposed for filling n-sided holes, including the Gregory patch, the Brown's square and the Nielson patch.

1.4 Surface Approximation

After filling the n-sided holes, a smooth surface consisting of high degree parametric patches is obtained. Chapters 5 and 6 investigate the approximation of high degree parametric surface patches by low degree surface patches with both parametric and implicit representations. The need to approximate curves and surfaces arises in many subdisciplines within CAGD. In computer graphics, piecewise linear approximations are used in combination with subdivision algorithms for rendering curves and surfaces. They are also used for finding intersections between curve-curve, curve-surface, and surface-surface. Approximation techniques also play a useful role in facilitating the exchange of data between different geometric modeling systems, because certain systems have limitations on the maximum polynomial degree that they can store and manipulate.

If we restrict our surface representations to parametric polynomial or rational representations or to implicit polynomial representations, then a large class of surfaces cannot be represented exactly. However, by the Weierstrass approximation theorem [Dav63], any continuous surface can be approximated with arbitrary precision by a parametric polynomial surface. This section reviews some of the previous work in CAGD on the approximation of curves and surfaces, and outlines the contribution of this thesis. There are four cases to consider depending upon the original and the final representation of the curve or the surface.

Approximation of parametric curves and surfaces by lower degree parametric curves or surfaces has been widely studied in CAGD. Watkins and Worsey [WW88] investigated approximation of Bézier curves using Tschebycheff polynomials to minimize the maximum norm between the given curve and the approximating lower degree curve. Forrest [For72a] describes a degree reduction method for Bézier curves for interactive interpolation and approximation. Farin et al [Far83, FRSW87, Far91] examine approximation of Bézier curves and cubic B-spline curves in order to improve their curvature plots.

Dannenberg and Nowacki [DN86] combine a degree reduction method with an optimum knot replacement scheme of Hoelzle [Hoe83] to approximate polynomial curves and surfaces with lower degree piecewise polynomial curves and surfaces. Hoschek [Hos87] describes an algorithm for degree reduction of integral spline curves by enforcing first or second order contacts between low degree and high degree spline segments at both ends. Patrikalakis et al [Pat89, BP89] perform degree reduction on B-spline curves and tensor product B-spline patches by enforcing continuity conditions up to certain order at end points or along boundary curves and then solving a system of linear equations whose coefficients form a banded matrix.

Lachance [Lac88] proposes an approximation scheme for tensor-product parametric surfaces using constrained Tschebycheff polynomials to minimize the maximum norm between the given surface and the approximating lower degree surface. Petersen et al [PPW87, Pet84] describe a scheme for approximating a degree d multivariate polynomial by one of degree $d-1$ for adaptive contouring of trivariate interpolants. Arner [Arn85] uses a degree reduction algorithm to solve the problem of hidden surface elimination.

There have been very few studies of the approximation of parametric curves or surfaces by implicit curves or surfaces, or vice-versa. Chuang and Hoffmann [CH89] investigate local implicit approximations of rational curves and surfaces. For surfaces, local implicit approximations of degree n results in an approximant that has roughly $n^{\frac{3}{2}}$ order of contact.

Waggenspack and Anderson [WA89] and Sederberg et al [SZZ88] investigate the problem of approximating an implicit curve by piecewise continuous parametric curves. Bajaj and Xu [BX92] study the approximation of real algebraic curves by piecewise continuous rational parametric curves.

The difficulty of approximating an implicit curve or surface by a lower degree implicit curve or surface is described by Sederberg [Sed84] and Warren [War91].

Several methods have been investigated for the degree reduction of Bézier or B-spline curves and tensor-product surfaces, but very few studies have been done for the degree reduction of triangular Bézier surface patches. This thesis describes a new method for approximating high degree parametric Bézier surfaces by low degree parametric Bézier surfaces. The technique is applicable to triangular as well as rectangular Bézier surface patches, and the technique works in any dimension. Therefore, this thesis describes a general degree reduction algorithm for Bézier simplexes.

The algorithm has the following properties: (1) Symmetry: The degree reduction method is symmetric with respect to the corner points of the Bézier simplex. (2) Restriction: The degree reduction method restricted to the boundary of a Bézier simplex yields the same result as the boundary of the degree-reduced Bézier simplex. (3) Interpolation: The degree-reduced simplex of degree e interpolates the value and the first $\lfloor \frac{e-1}{2} \rfloor$ derivatives at the corner points of the original Bézier simplex. (4) Optimal order of approximation: The order of approximation of the given simplex by the degree-reduced simplex is $O(h^{e+1})$, where h is the diameter of the domain simplex, which is optimal for functional approximation.

This work concludes by describing a variant of the least squares method for approximating a low degree parametric surface by a low degree surface with both parametric and implicit representations. One of the major difficulties in approximating a parametric curve or a surface by an implicit curve or a surface is the presence of undesirable extra sheets in the implicit representation which then interfere with the subsequent modeling operations. This and other difficulties are described in detail later in Chapter 6. Previous techniques for fitting implicit surfaces to scattered data [Pra87, PK89, BIW90], as well as previous techniques of free-from modeling using implicit surface patches [Dah89, Guo90] suffer from these defects. Moore and Warren [MW90] use an auxiliary distance function to avoid these undesirable sheets. In this work, these undesirable portions of surfaces are automatically excluded because the low degree surface patches with introduced in Chapters 2 and 3 do not possess any extra sheets or self-intersections within their defining tetrahedra.

1.5 Overview

This work consists of 7 chapters. Chapter 2 describes a new method for creating triangular Bézier surface patches with parametric and implicit degree two. These surface patches are defined by imposing certain geometric restrictions on quadratic Bézier control polyhedron. Chapter 3 describes a new method for creating rectangular Bézier surface patches with biquadratic parametric degree and implicit degree three. This chapter also describes the skew-line coordinate system for a tetrahedron, which clarifies and simplifies the construction. In both these chapters, the conversion procedures between the two representations are presented. A completeness theorem is proved that states that these constructions capture all surface patches with both representations of the specified degrees. The properties of the resulting surface patches include all of the standard properties of parametric Bézier surfaces, including interpolation of the corners of the control polyhedron and the convex hull property. Chapters 2 and 3 conclude with a description of how these patches can be used for surface design.

Chapters 4 to 6 describe a scheme for filling n-sided holes and for approximating the resulting smooth surface consisting of high degree parametric surface patches by a continuous surface consisting of the low degree surface patches which were introduced in Chapters 2 and 3. Chapter 4 begins by reviewing the concepts of Bézier simplexes and S-patches, needed in the proposed solution. The key contribution of this chapter is to describe a new technique for filling an n-sided hole using a single rational parametric patch with a geometrically intuitive compact representation. Moreover, the technique helps to unify the myriad existing solutions to the problem of filling holes for any number of sides, and for any number of cross-boundary derivatives. Chapter 5 describes a new degree reduction algorithm for approximating parametric patches of high degree by parametric patches of low degree. The degree reduction procedure has the important properties of boundary continuity, affine invariance and optimal order of approximation. In Chapter 6, a variant of the least squares technique is used to approximate parametric surfaces of low degree by low degree surfaces with both parametric and implicit representations. Chapter 7 summarizes the results from the previous chapters and poses several interesting open problems.

Chapter 2

Bézier Representation for Quadric Surface Patches

The main purpose of this work is to explore surface patches with multiple representations. Section 1.2.3 of the previous chapter described the difficulty in creating such patches. This chapter introduces a method for creating triangular Bézier surface patches on an implicit quadric surface. Simple curved surfaces such as spheres, cylinders, and cones play a fundamental role in Computer Aided Geometric Design. Each is a quadric surface and can thus be defined as the zero contour of a trivariate quadratic function. Quadric surfaces and the volumes they enclose are often the primitive elements in CSG representations [RV83], but one difficulty has complicated the use of quadric surfaces in geometric modeling: the lack of a convenient method for creating bounded patches on a quadric surface.

The Bézier representation uses a control polyhedron to specify a bounded portion of a surface. Given a control polyhedron P, Bézier methods generate a surface patch S that approximates P. Several desirable properties relate S and P. For example, S interpolates the corners of P, is tangent to P at its corners, and lies in the convex hull of P. A method for creating quadric surface patches and associated control polyhedra that are similarly related would greatly simplify the process of modeling with quadrics.

Several methods have been proposed to address this problem for particularly important types of quadrics like spheres and cylinders [Boe89, Pie86, FPW88, Cob88, Hil86, PT87]. A method based on general quadratic Bézier surfaces is unlikely to succeed because, in general, a triangular parametric quadratic Bézier surface patch has implicit degree four [SA85]. Under certain restrictions, however, such quadratic parametrizations do define a portion of a quadric surface. This chapter gives a geometric method for creating control polyhedra corresponding to such parametrizations. The method relies on the fact that any quadric surface is projectively equivalent to a surface of the form $x_3 = q(x_1, x_2)$ [Bla54] for some quadratic polynomial q. The method can generate a control polyhedron to represent any triangular patch on a quadric surface that is the image of a plane triangle under a quadratic parametrization. In this sense, the method is complete.

This chapter also addresses the problem of joining with C^0 or C^1 continuity two patches represented by control polyhedra of this kind. In the case of C^1 continuity, a technique is described that blends two quadric surface patches with an auxiliary two-sided quadric surface patch. This technique, in conjunction with the Powell-Sabin interpolant, allows the creation of C^1 free-form objects using only quadric surface patches.

2.1 Bézier Representation for Quadratic Curves

Rational quadratic Bézier curves [For68, Lee87, Far88] and their associated weighted control polygons can represent arbitrary segments of implicit quadratic curves. This section describes an alternative method for constructing segments of implicit quadratic curves that does not attach weights to the control polygon. A principal advantage of the new method is that it easily generalizes to quadric surface patches.

Any quadratic function $q(x)$ can be represented as a linear combination of the Bernstein basis functions of order two.

$$q(x_1) = \sum_{\substack{i_1, i_2 \geq 0 \\ i_1 + i_2 = 2}} \frac{2!}{i_1! i_2!} c_{i_1 i_2} x_1^{i_1} (1 - x_1)^{i_2}.$$

The values $c_{i_1 i_2}$ define a *control polygon* with vertices $(1, c_{20})$, $(0.5, c_{11})$, and $(0, c_{02})$ that approximates the graph of the curve $x_2 = q(x_1)$ on the interval $0 \leq x_1 \leq 1$. The associated curve segment C is called a *functional Bézier curve.* Unfortunately, not every segment of a quadratic curve is a functional Bézier curve. However, we shall see below that any irreducible quadratic curve can be brought into functional form using a linear projective transformation.

Points are identified using *homogeneous* coordinates (x_1, x_2, w) in the projective plane. The space is two-dimensional since the triple (kx_1, kx_2, kw) denotes the same point for all $k \neq 0$. A point with homogeneous coordinates (x_1, x_2, w) has Cartesian coordinates $(x_1/w, x_2/w)$ if $w \neq 0$. If $w = 0$, then these coordinates denote a point at infinity. A linear projective transformation is simply a linear map from one set of homogeneous coordinates to another. Such a transformation is uniquely determined by specifying four points in the plane and their respective images, provided no three of either set of four points are collinear [Max63]. A linear projective transformation may be expressed in Cartesian coordinates as a rational linear transformation.

The control polygon for a functional Bézier curve may be interpreted as lying within an infinite triangle sitting on the unit interval. This triangle has vertices with homogeneous coordinates $(1,0,1)$, $(0,0,1)$, and $(0,1,0)$. The situation is illustrated in Figure 2.1. Let T be a linear projective transformation that maps the vertices p_{20}, p_{02}, and f, respectively, of a *reference triangle* to the vertices $(1,0,1)$, $(0,0,1)$, and $(0,1,0)$ and maps p_{11}, the mid-point of $p_{20}p_{02}$, to $(0.5,0,1)$. There exists a one parameter family of such transformations. If a point q in this reference triangle has the *barycentric* coordinates $(\alpha_1, \alpha_2, \alpha_3)$ where

$$q = \alpha_1 p_{20} + \alpha_2 p_{02} + \alpha_3 f$$

$\alpha_1 + \alpha_2 + \alpha_3 = 1$, and $\alpha_1, \alpha_2, \alpha_3 \geq 0$, then one such transformation T is:

$$\begin{aligned} x_1 &= \frac{\alpha_1}{1-\alpha_3} \\ x_2 &= \frac{\alpha_3}{1-\alpha_3} \end{aligned}$$

Section 2.3 shows that any transformation consistent with the above conditions above yields the same conic curve segment. The inverse transformation T^{-1} maps lines parallel to the x_2-axis to lines through f. Thus, f is called the *focal* vertex. The edge $p_{20}p_{02}$ is the *base* of the reference triangle.

Consider three points v_{20}, v_{11} and v_{02} that lie on the rays fp_{20}, fp_{11}, and fp_{02} respectively. Their images under T are $(1, c_{20})$, $(0.5, c_{11})$, and $(0, c_{02})$. These three points define the control vertices of a functional Bézier curve C. The image of C under T^{-1} is a segment of a quadratic curve. This curve segment, $\hat{C}$, and the polygon formed by v_{20}, v_{11}, and v_{02} share all of the standard properties relating a functional Bézier curve and its control polygon.

- $\hat{C}$ interpolates v_{20} and v_{02}.
- $\hat{C}$ is tangent to $v_{20}v_{11}$ at v_{20} and $v_{11}v_{02}$ at v_{02}.
- $\hat{C}$ lies in the convex hull of $v_{20}v_{11}v_{02}$.

This representation for $\hat{C}$ in terms of a reference triangle and associated control points v_{20}, v_{11} and v_{02} is called the *projective functional* representation of $\hat{C}$.

The projective functional representation can be used to design segments of quadratic curves. The designer first specifies a reference triangle and identifies a focal vertex for that triangle. The designer then selects a point on each of the three rays fp_{20}, fp_{11}, and fp_{02} to create a quadratic control polygon embedded in the reference triangle. The system computes the transformation T and applies T to the embedded control polygon to create the control polygon for a functional Bézier curve. Finally, this functional Bézier curve is mapped back to the reference triangle via T^{-1} to yield the desired quadratic curve segment. Figure 2.1 illustrates this construction.

Any segment of a quadratic curve can be represented in projective functional form. The reference triangle for the segment must be chosen so as to have its focal vertex on the curve. This is necessary because the functional curve $x_2 = q(x_1)$ always passes through the point at infinity with homogeneous coordinates $(0, 1, 0)$. This fact is obvious if one converts

$$x_2 = c_{02}(1 - x_1)^2 + 2c_{11}(1 - x_1)x_1 + c_{20}x_1^2$$

to its homogeneous form:

$$wx_2 = c_{02}(w - x_1)^2 + 2c_{11}(w - x_1)x_1 + c_{20}x_1^2.$$

This equation is satisfied at $(0, 1, 0)$. Therefore, the image of this curve under T^{-1} must always pass through the focal vertex f. Both [Pie86] and [Her89] suggest a

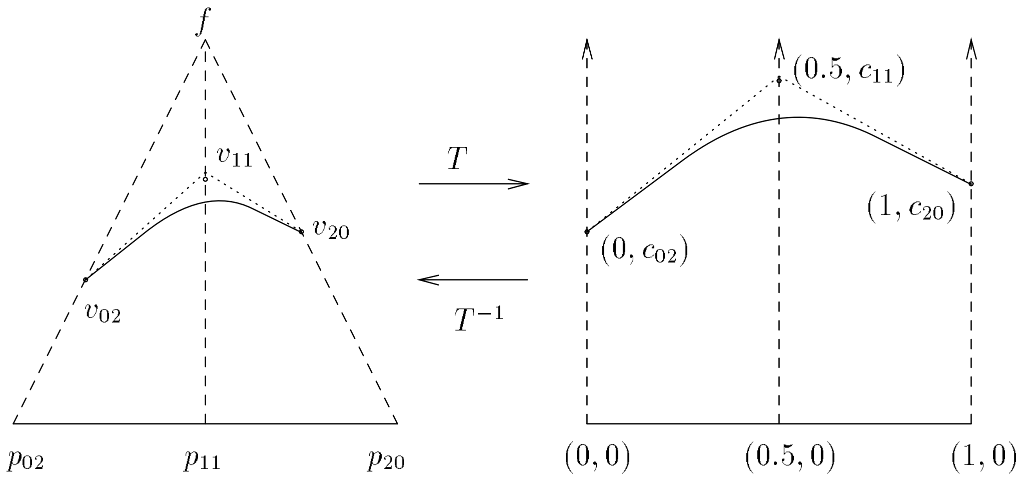

Figure 2.1 Projective functional representation for a quadratic curve segment

similar construction for quadratic curves. These constructions have their roots in classical projective geometry [Cox74, VY45, Coo45].

Figure 2.2 illustrates how the construction can be extended to create a C^1 piecewise quadratic curve. Because a curve segment and its control polygon are tangent at their common endpoint, choosing adjacent control polygons so that coincident polygon edges are collinear guarantees C^1 continuity of the curve segments.

2.2 Bézier Representation for Quadric Surfaces

The construction outlined above for quadratic curves extends easily to surfaces. A quadratic function $q(x_1, x_2)$ can be represented as a linear combination of bivariate Bernstein basis functions of order two.

$$q(x_1, x_2) = \sum_{\substack{i_1, i_2, i_3 \geq 0 \\ i_1 + i_2 + i_3 = 2}} \frac{2!}{i_1! i_2! i_3!} c_{i_1 i_2 i_3} x_1^{i_1} x_2^{i_2} (1 - x_1 - x_2)^{i_3}.$$

The Bézier ordinates $c_{i_1 i_2 i_3}$ define a *control polyhedron* with vertices $(i_1/2, i_2/2, c_{i_1 i_2 i_3})$. This control polyhedron approximates the portion of the surface $x_3 = q(x_1, x_2)$ on the triangular region where $0 \leq x_1, x_2, x_1 + x_2 \leq 1$. The associated surface patch S is called a *functional Bézier surface* [Far88]. As in the case of functional curves, functional Bézier surfaces cannot represent every quadric surface patch. However, any irreducible quadric surface is equivalent to a surface of the form $x_3 = q(x_1, x_2)$ under some linear projective transformation [Bla54].

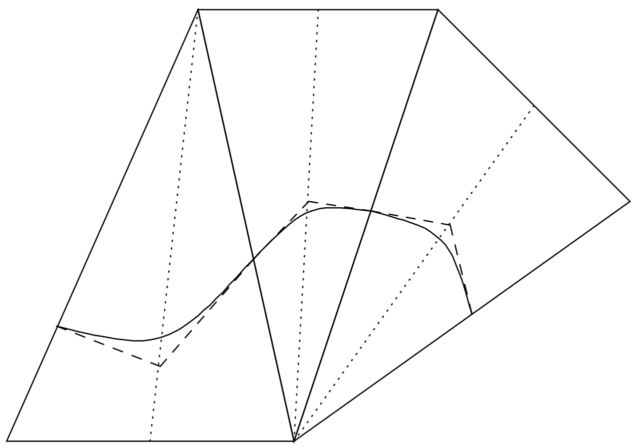

Figure 2.2 C^1 piecewise quadratic curve

The control polyhedron for a functional Bézier surface S is defined over an infinite tetrahedron whose vertices have homogeneous coordinates $(1,0,0,1)$, $(0,1,0,1)$, $(0,0,0,1)$ and $(0,0,1,0)$. Let p_{200}, p_{020}, p_{002}, and f, denote the vertices of a *reference tetrahedron* with vertex f as its focal vertex and the triangle $p_{200}p_{020}p_{002}$ as its base. Let p_{011}, p_{101}, and p_{110} be the midpoints of the edges $p_{020}p_{002}$, $p_{002}p_{200}$, and $p_{200}p_{020}$, respectively. Let T denote a linear projective transformation that maps the vertices p_{200}, p_{020}, p_{002}, and f to the vertices $(1,0,0,1)$, $(0,1,0,1)$, $(0,0,0,1)$ and $(0,0,1,0)$, respectively, of the infinite tetrahedron and maps the midpoints of the base of the reference tetrahedron to corresponding midpoints of finite edges on the infinite tetrahedron. There exists a one parameter family of such transformations. The particular choice of the transformation does not affect the final quadric surface patch, as is shown in Section 2.3.

Under the transformation T^{-1}, lines parallel to the x_3-axis are mapped to lines through f. Let $v_{i_1i_2i_3}$ denote a point on the ray from f through $p_{i_1i_2i_3}$. The image of $v_{i_1i_2i_3}$ under T is a point of the form:

$$(i_1/2, i_2/2, c_{i_1i_2i_3}) = T(v_{i_1i_2i_3})$$

for $i_1, i_2, i_3 \geq 0; i_1 + i_2 + i_3 = 2$. These new points define the control vertices of a functional Bézier surface S.

The image of S under T^{-1} is a triangular portion of a quadric surface that is embedded in the infinite triangular pyramid with vertex f and edges containing p_{200}, p_{020}, p_{002}. This triangular surface patch $\hat{S}$ and the polygon formed by v_{200}, v_{020}, v_{002}, v_{011}, v_{101}, and v_{110} share all of the properties of a standard Bézier surface.

- $\hat{S}$ interpolates v_{200}, v_{020}, and v_{002}.
- $\hat{S}$ is tangent to the triangle $v_{101}v_{200}v_{110}$ at v_{200}, tangent to $v_{110}v_{020}v_{011}$ at v_{020}, and tangent to $v_{011}v_{002}v_{101}$ at v_{002}.
- $\hat{S}$ lies in the convex hull of v_{200}, v_{020}, v_{002}, v_{011}, v_{101}, and v_{110}.

The reference tetrahedron $p_{200}p_{020}p_{002}f$ and the control points v_{200}, v_{020}, v_{002}, v_{011}, v_{101}, and v_{110} are called the *projective functional* representation for the quadric surface patch $\hat{S}$. Figure 2.3 illustrates this construction.

The projective functional form can be used to create triangular quadric surface patches. The designer first specifies a reference tetrahedron with an identified focal vertex, then creates a quadratic control polyhedron embedded in this reference

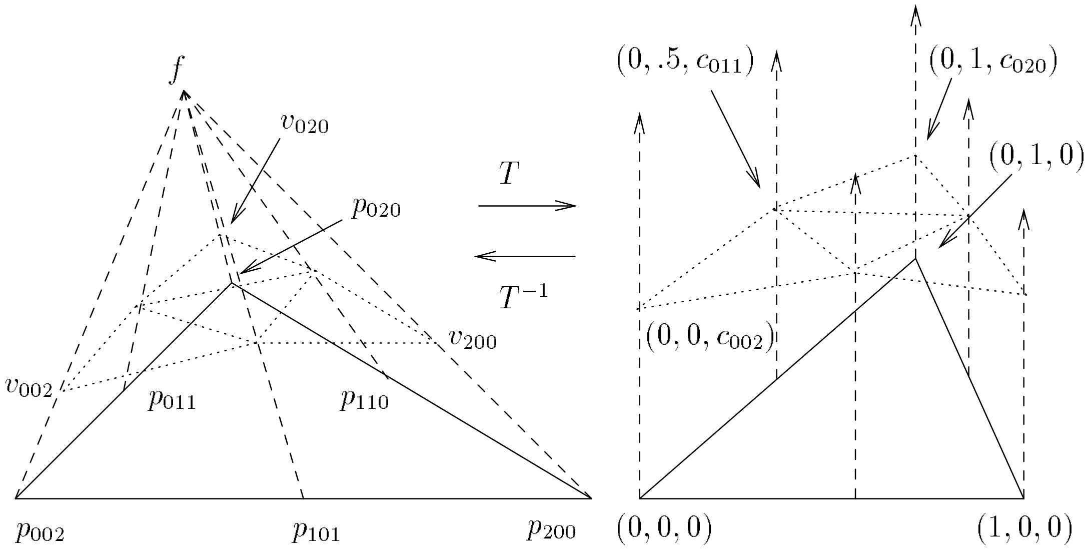

Figure 2.3 Projective functional representation for a quadric surface patch

tetrahedron, subject to the constraints described above. The system automatically computes the transformation T and applies T to the embedded control polyhedron to create the control polyhedron for a functional Bézier surface. Finally, the functional Bézier surface is mapped back to the reference tetrahedron via T^{-1} to yield the desired quadric surface patch.

2.3 Conversion Between Representations

It is now possible to establish the relationship between the projective functional representation of quadric surface patches and two other important representations, the implicit trivariate Bézier representation and the rational quadratic Bézier representation.

A portion of a quadric surface can be represented as the zero contour of a trivariate quadratic polynomial $Q(x_1, x_2, x_3)$ over a defining tetrahedron [Sed85]. Specifically, if a point (x_1, x_2, x_3) inside this tetrahedron is expressed in its barycentric coordinates

$$(x_1, x_2, x_3) = \alpha_1 p_{200} + \alpha_2 p_{020} + \alpha_3 p_{002} + \alpha_4 f,$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \geq 0$ and $\alpha_1+\alpha_2+\alpha_3+\alpha_4 = 1$, then $Q(x_1, x_2, x_3)$ can be expressed in its trivariate Bézier representation as

$$Q(x_1, x_2, x_3) = \sum_{\substack{i_1+i_2+i_3+i_4 \geq 0 \\ i_1+i_2+i_3+i_4=2}} \frac{2!}{i_1!i_2!i_3!i_4!} b_{i_1 i_2 i_3 i_4} \alpha_1^{i_1} \alpha_2^{i_2} \alpha_3^{i_3} \alpha_4^{i_4}. \tag{2.1}$$

The following theorem relates the projective functional form to the implicit trivariate form.

Theorem 2.1 Let f be the focal vertex of a reference tetrahedron with base vertices p_{200}, p_{020}, and p_{002}. Let $\hat{S}$ be a quadric surface patch in projective functional form with the control points $v_{i_1 i_2 i_3} = (1 - \lambda_{i_1 i_2 i_3}) p_{i_1 i_2 i_3} + \lambda_{i_1 i_2 i_3} f$, $\lambda_{i_1 i_2 i_3} < 1$. $\hat{S}$ is the zero contour of the trivariate Bernstein polynomial Q defined over the tetrahedron $p_{200} p_{020} p_{002} f$ where Q has the following coefficients:

$$\begin{aligned} b_{0002} &= 0 \\ b_{1001} = b_{0101} = b_{0011} &= -0.5 \\ b_{i_1 i_2 i_3 0} &= \frac{\lambda_{i_1 i_2 i_3}}{1 - \lambda_{i_1 i_2 i_3}} \; \forall i_1, i_2, i_3 \geq 0; i_1 + i_2 + i_3 = 2. \end{aligned}$$

Proof Let $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ be the barycentric coordinates of a point in the reference tetrahedron. Let a point in the functional domain associated with $\hat{S}$ have Cartesian coordinates (x_1, x_2, x_3). Any linear projective transformations T satisfying the conditions in Section 2.2 can be given by

$$\begin{aligned} x_1 &= \frac{\alpha_1}{1 - \alpha_4}, \\ x_2 &= \frac{\alpha_2}{1 - \alpha_4}, \\ x_3 &= \frac{\alpha_4}{(1 - \alpha_4)\sigma}, \end{aligned} \tag{2.2}$$

where σ is a free parameter.

If the control points $v_{i_1 i_2 i_3}$ are expressed in barycentric coordinates, then the α_4 coordinate for each point is exactly $\lambda_{i_1 i_2 i_3}$. From Equation 2.2 it follows that the image of $v_{i_1 i_2 i_3}$ under T is the point $(i_1/2, i_2/2, c_{i_1 i_2 i_3})$ where

$$c_{i_1 i_2 i_3} = \frac{\lambda_{i_1 i_2 i_3}}{(1 - \lambda_{i_1 i_2 i_3})\sigma}. \tag{2.3}$$

These points are the control points of the associated functional surface patch.

The image of the functional Bézier surface

$$x_3 = \sum_{i_1+i_2+i_3=2} \frac{2!}{i_1!i_2!i_3!i_4!} c_{i_1 i_2 i_3} x_1^{i_1} x_2^{i_2} (1 - x_1 - x_2)^{i_3}. \tag{2.4}$$

under T^{-1} is an implicit trivariate surface. Substituting Equation 2.2 for x_1,x_2,x_3 and multiplying through by $(1 - \alpha_4)^2$, we find that the resulting equation is quadratic in α_1, α_2, α_3, α_4. A common factor of σ is shared by all the terms and therefore may be canceled out. The resulting coefficients of the quadratic basis functions are exactly those stated in the theorem. $\square$

Corollary 2.1 The surface $\hat{S}$ of the previous theorem may also be expressed as a rational quadratic Bézier surface [Far88] with control points $v_{i_1 i_2 i_3}$ and associated weights $\frac{1}{1-\lambda_{i_1 i_2 i_3}}$.

This corollary follows from the fact that $\hat{S}$ is the image under the transformation T^{-1} of the functional surface given by Equation 2.4. It is straightforward to construct a parametrization for $\hat{S}$ in terms of the parameters x_1 and x_2 simply by composing Equation 2.4 and T^{-1}.

2.4 A Completeness Theorem

Not all triangular quadric surface patches that are bounded by planar curves can be represented in projective functional form. A surface patch in projective functional form is necessarily embedded in a quadric surface that passes through the focal vertex of the associated reference tetrahedron. This follows from the observation that $b_{0002} = 0$ when the surface is represented in implicit trivariate form. Alternatively, the functional surface of Equation 2.4 always passes through the vertex $(0, 0, 1, 0)$, the image of f under T. Since T^{-1} preserves incidence, this surface must pass through the focal vertex f. For example, the octant of the sphere $x_1^2 + x_2^2 + x_3^2 - 1 = 0$ where $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$ cannot be described by this construction, because the planes that contain the boundary curves intersect in the center of the sphere, not on its surface. Triangular portions of a sphere can be described if and only if the focal vertex of the reference tetrahedron lies on the sphere.

Theorem 2.2 Let $\hat{S}$ be the portion of a quadric surface Q that lies inside a triangular cone bounded by the planes P_0, P_1, and P_2. The following three statements are equivalent:

1. $\hat{S}$ can be represented in projective functional form.
2. $(P_0 \cap P_1 \cap P_2) \subseteq Q$.
3. $\hat{S}$ is the image of a domain triangle under a rational quadratic parametrization.

Proof We will show (1) $\Rightarrow$ (3) $\Rightarrow$ (2) $\Rightarrow$ (1). Statement one directly implies statement three by construction. [SA85] shows that for a rational quadratic triangular Bézier surface patch to be a quadric, the planes on which the three boundary curves lie must intersect in a point on the quadric surface. Therefore, statement three implies statement two. Finally, if statement two is satisfied, then we may choose a reference tetrahedron three of whose faces match P_0, P_1, and P_2. If the focal vertex of this tetrahedron is $(P_0 \cap P_1 \cap P_2)$, the implicit trivariate Bézier representation for Q has $b_{0002} = 0$. Finally, we may choose the base for this reference tetrahedron is such a manner that $b_{1001} = b_{0101} = b_{0011} = -0.5$. The remaining coefficients are uniquely determined by Q. Thus, statement two implies statement one. $\square$

2.5 Designing with Quadric Surface Patches

Among the most appealing features of Bézier representations is the geometric interpretation they provide for continuity conditions. For C^0 contact, quadric surfaces and their associated control polyhedra in the projective functional representation behave as expected. Consider two reference tetrahedra H_0 and H_1 that share a common focal vertex f and a common edge p_0p_1. The two quadric surfaces S_0 and S_1 associated with H_0 and H_1 meet with C^0 contact across the face p_0p_1f if they share the same embedded control polygon along this face.

For C^1 contact, the situation is little more complicated. If the bases of H_0 and H_1 are coplanar, then a single homogeneous linear transformation maps both tetrahedra to the unbounded projective tetrahedra of the functional case. A necessary and sufficient condition for S_0 and S_1 to meet with C^1 contact across the boundary is that they share the same tangent planes at both of their common corners [Far88]. In Figure 2.4, each pair of dark triangles sharing a common edge must be coplanar.

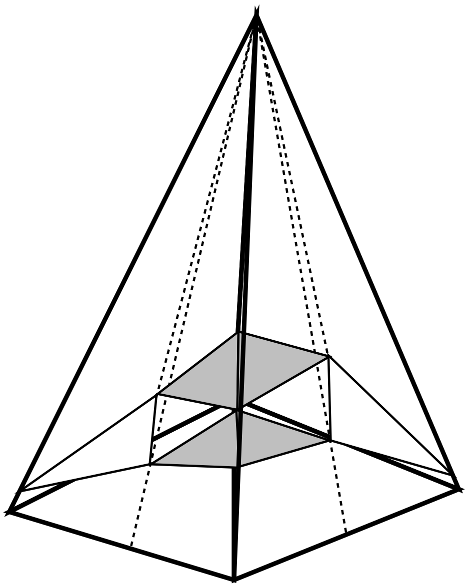

Figure 2.4 C^1 continuity condition for reference tetrahedra with coplanar bases

If the bases of H_0 and H_1 are not coplanar, agreement of the tangent planes of S_0 and S_1 at their common corners does not necessarily force tangency along the entire boundary curve. However, the following construction, essentially derived from [War86], produces a two-sided quadric surface patch that smoothly blends these two patches together. This two-sided quadric surface patch can be represented as a zero contour of a trivariate Bernstein polynomial inside a suitable tetrahedron as in Figure 2.5.

Let $Q_0 = 0$ and $Q_1 = 0$ be the implicit equations of S_0 and S_1 respectively. These patches intersect in a quadratic curve lying along the common face $p_0 p_1 f$. Let $P = 0$ be the linear equation of this face. Given these data, it is always possible to find a constant c and a linear polynomial R such that

$$Q_0 - cQ_1 = PR. \tag{2.5}$$

The geometric interpretation of this equation is that S_0 and S_1 intersect in a pair of planar curves lying in $P = 0$ and $R = 0$.

The quadric surface B with the equation

$$4aQ_0 - (P + aR)^2 = 0.$$

is tangent to S_0 along $P + aR = 0$. By Equation 2.5, B may also be expressed as

$$4acQ_1 - (P - aR)^2 = 0.$$

Thus B is also tangent to S_1 along $P - aR = 0$.

Let q_0 and q_1 denote the two common corners of S_0 and S_1. If S_0 and S_1 share the same tangent planes at q_0 and q_1, then the intersection curves lying in $P = 0$ and $R = 0$ must pass through q_0 and q_1. Thus, both $P + aR = 0$ and $P - aR = 0$ must pass through q_0 and q_1. Trimming S_0, S_1 and B by these planes forms a two-sided patch on B that interpolates q_0 and q_1, and the resulting mesh of surface patches is C^1 continuous. Note that as a approaches zero, this two-sided patch shrinks toward the intersection of S_0 and S_1.

We now describe the creation of C^1 piecewise quadric surfaces. In the functional case, Powell and Sabin describe a C^1 piecewise quadratic interpolant that splits a given triangle into six subtriangles and creates a quadric surface patch over each subtriangle [PS77]. Together the six quadric surfaces form a C^1 macro patch that has more flexibility than a single quadric surface patch. Figure 2.6 shows the details of that construction. The Powell-Sabin interpolant may be used to create even more flexible C^1 piecewise quadric macro patches in conjunction with the use of the projective functional representation. To perform the construction on a reference tetrahedron, apply the Powell-Sabin split to the base of that tetrahedron. Because the bases of the six subtetrahedra are coplanar and the same linear homogeneous transformation is applied to each tetrahedron, the C^1 continuity of the functional Powell-Sabin patch implies C^1 continuity for the transformed patch.

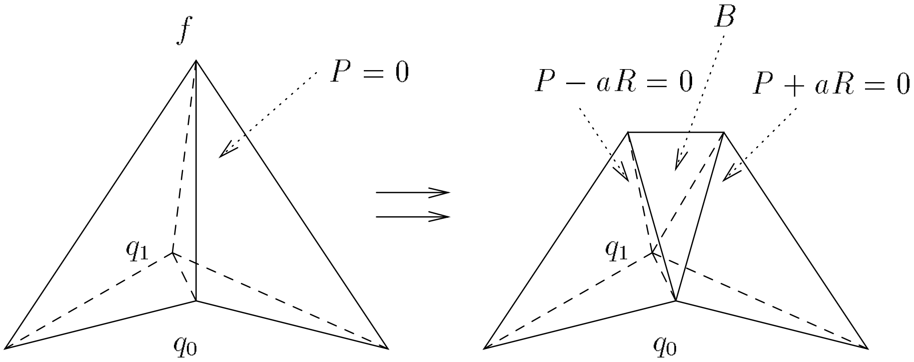

Figure 2.5 Blending two quadric surfaces with another quadric surface

Given two reference tetrahedra H_0 and H_1 sharing a common focal vertex f and a base edge p_0p_1, we may ensure C^0 continuity between adjacent macro patches by forcing the common edge p_0p_1 to have the same split point p on both tetrahedra. If h_0 and h_1 are the interior split points for the bases of H_0 and H_1, then forcing coplanarity of f, p, h_0, and h_1 enforces C^1 contact. If the bases of H_0 and H_1 are coplanar, then the macro patches necessarily meet with C^1 continuity. If the bases are not coplanar, then the quadric blending method of Section 2.5 is required to create a pair of two-sided quadric patches that smoothly join the macro patches.

This chapter has described a geometric construction for creating triangular surface patches on quadric surfaces. The method is complete in that any triangular patch that has a quadratic parametrization is represented by this approach. By mapping to the functional case, all of the standard Bézier techniques, such as the deCasteljau algorithm, and many of the useful properties of Bézier representations, like the convex hull property, are preserved. Because it is easy to compute either the parametric or the implicit representation of a patch from its functional representation, the method

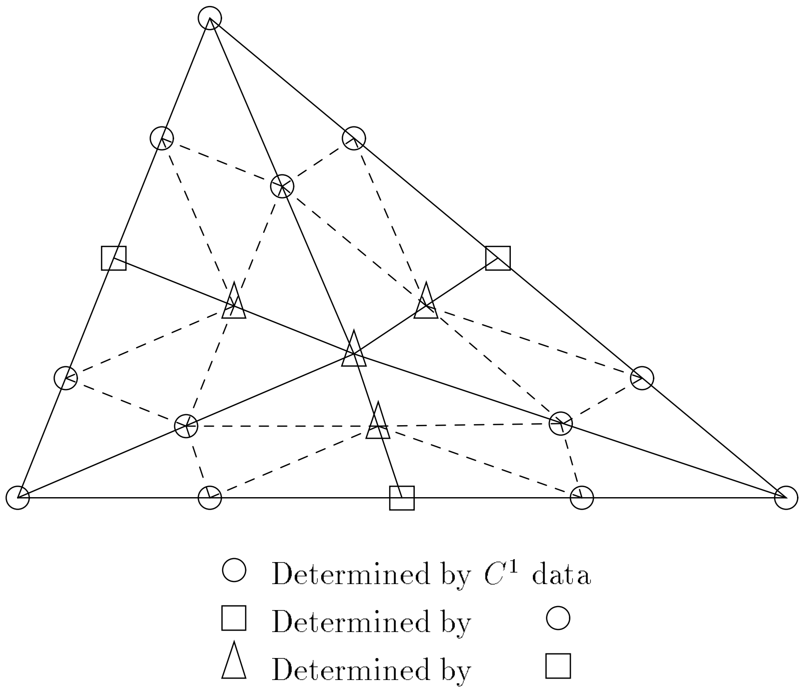

Figure 2.6 Powell-Sabin Split

allows for multiple representations of surface patches thus combining the best features of both representations.

Chapter 3

Bézier Representation for Cubic Surface Patches

The previous chapter described a new method for creating triangular Bézier surface patches on an implicit quadric surface. This chapter describes a new method for creating rectangular Bézier surface patches on an implicit cubic surface. Sederberg [Sed90a, Sed90b] introduced implicit cubics for surface design to enhance the flexibility of design, while still offering fast computation with low degree. Traditional techniques for representing surfaces have relied on rational parametric representations. However, a method based on rational quadratic Bézier surfaces is unlikely to succeed because, in general, a rectangular parametric biquadratic Bézier surface patch has an implicit degree of eight [SA85]. Under certain restrictions, however, such quadratic parametrizations do define a portion of a cubic surface by reducing the implicit degree from eight to three. This chapter describes a geometric method for creating control polyhedra corresponding to such parametrizations. The method allows modeling implicit cubics by prescribing a rectangular biquadratic Bézier control polyhedron embedded within a tetrahedron and satisfying a projective constraint. The control polyhedron and the resulting cubic surface patch satisfy all of the standard properties of parametric Bézier surfaces, including interpolation of the corners of the control polyhedron and the convex hull property.

The method relies on the fact that given two skew lines on a smooth cubic surface patch (in general there are 27 lines on a cubic surface [SS87]), the patch can be conveniently parametrized by taking the intersection point with the patch of any straight line joining two arbitrary points on the two skew lines [SS87, AB87b]. In addition, the introduction of a coordinate system based on two skew edges of a tetrahedron greatly simplifies and clarifies the construction. The method can generate a control polyhedron to represent any rectangular patch of a non-singular cubic surface that is the image of a plane rectangle under a biquadratic parametrization. In this sense, the method is complete.

3.1 Skew-line Coordinate System for a Tetrahedron

This section first describes the implicit trivariate Bézier representation of a cubic surface. A portion of a cubic surface can be represented as the zero contour of a trivariate cubic polynomial $C(x_1, x_2, x_3)$ over a defining tetrahedron [Sed85]. Specifically, if a point (x_1, x_2, x_3) inside this tetrahedron is expressed in its barycentric coordinates

$$(x_1, x_2, x_3) = \alpha_1 p_1 + \alpha_2 p_2 + \alpha_3 p_3 + \alpha_4 p_4 \tag{3.1}$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \geq 0$ and $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1$, then $C(x_1, x_2, x_3)$ can be expressed in its trivariate Bézier representation as

$$C(x_1, x_2, x_3) = \sum_{\substack{i_1, i_2, i_3, i_4 \geq 0 \\ i_1 + i_2 + i_3 + i_4 = 3}} \frac{3!}{i_1! i_2! i_3! i_4!} b_{i_1 i_2 i_3 i_4} \alpha_1^{i_1} \alpha_2^{i_2} \alpha_3^{i_3} \alpha_4^{i_4}. \tag{3.2}$$

The coefficients $b_{i_1 i_2 i_3 i_4}$ are called implicit *weights*. These weights are associated with *domain points* $(\frac{i_1}{3}, \frac{i_2}{3}, \frac{i_3}{3}, \frac{i_4}{3})$ of the tetrahedron. The weights $b_{i_1 i_2 i_3 i_4}$ with $i_3 + i_4 = n$ form the *layer* n of rectangular $(3-n) \times (n)$ tensor product net of weights. One can think of these weights as lying in separate layers from layer 0 to layer 3 as illustrated in Figure 3.1.

Next the skew-line coordinate system for a tetrahedron is introduced. This coordinate system allows an alternative method for performing the deCasteljau algorithm

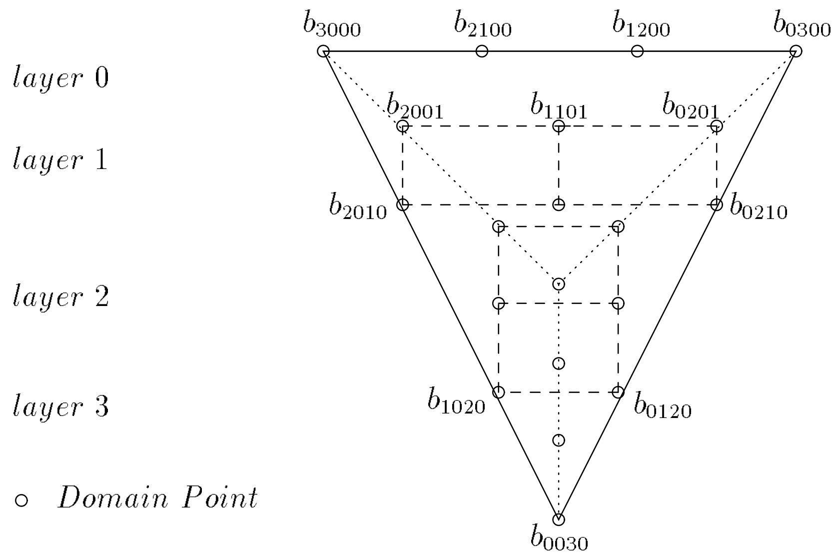

Figure 3.1 Tensor product layers of weights in the implicit trivariate representation for a cubic surface patch

to compute a point on a surface with an implicit trivariate representation. Let p_1p_2 and p_3p_4 be two skew edges of a tetrahedron. Let P and Q be arbitrary points on the straight line segments p_1p_2 and p_3p_4 respectively as illustrated in Figure 3.2. Let α and β be the barycentric coordinates of P and Q along p_1p_2 and p_3p_4 respectively. Finally, let γ be the barycentric coordinate of a point R on the straight line segment PQ so that

$$\begin{aligned} P &= \alpha p_2 + (1-\alpha)p_1 \\ Q &= \beta p_4 + (1-\beta)p_3 \\ R &= \gamma Q + (1-\gamma)P \end{aligned} \tag{3.3}$$

The skew line coordinate system for the tetrahedron is then defined as (α, β, γ).

Suppose a portion of a cubic surface is given as the zero contour of a trivariate cubic polynomial $C(x_1, x_2, x_3)$ over a tetrahedron. The skew-line coordinates (α, β, γ) of the tetrahedron can be used to compute the value of the cubic polynomial $C(x_1, x_2, x_3)$ by performing the deCasteljau algorithm as described below.

Theorem 3.1 The value of the trivariate cubic polynomial $C(x_1, x_2, x_3)$ at a point with the skew-line coordinates (α, β, γ) w.r.t. the tetrahedron $p_1p_2p_3p_4$ can be computed as follows:

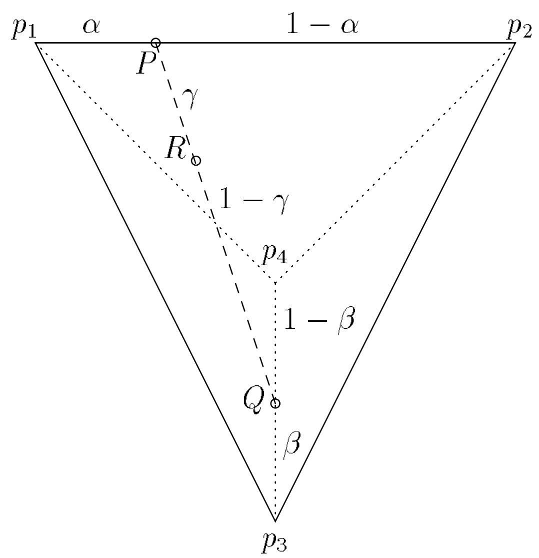

Figure 3.2 Skew-line coordinate system for a tetrahedron

1. Perform the tensor product deCasteljau algorithm in each layer w.r.t. (α, β), and let a_n be the value obtained in layer n.
2. Perform the univariate deCasteljau algorithm over $a_n, 0 \leq n \leq 3$ w.r.t. γ, and let a be the value obtained. Then $a = C(x_1, x_2, x_3)$.

Proof Let a_n be the value obtained by performing the deCasteljau algorithm in layer n w.r.t. the skew-line coordinates (α, β). In layers 0 and 3, perform the univariate deCasteljau algorithm w.r.t. the skew-line coordinates α and β respectively. In layers 1 and 2, perform the tensor product deCasteljau algorithm. The following results are obtained:

$$a_n(\alpha, \beta) = \sum_{\substack{i_3+i_4=n \\ i_1+i_2+i_3+i_4=3}} \frac{n!(3-n)!}{i_1!i_2!i_3!i_4!} b_{i_1 i_2 i_3 i_4} (1-\alpha)^{i_1} \alpha^{i_2} (1-\beta)^{i_3} \beta^{i_4} \tag{3.4}$$

Finally, perform the univariate deCasteljau algorithm over $a_n, 0 \leq n \leq 3$ w.r.t. the skew-line coordinate γ. The computed value a is then expressed as follows:

$$a = \sum_{n=0}^{3} \binom{3}{n} a_n(\alpha, \beta) \gamma^n (1-\gamma)^{3-n} \tag{3.5}$$

To conclude that a is the same as the value of the function $C(x_1, x_2, x_3)$, derive the transformation between the barycentric coordinates and the skew-line coordinates of the tetrahedron. By substituting for P and Q in Equation 3.3, and comparing with Equation 3.1, the following transformation of coordinates is obtained:

$$\begin{aligned} \alpha_1 &= (1-\alpha)(1-\gamma) \\ \alpha_2 &= \alpha(1-\gamma) \\ \alpha_3 &= (1-\beta)\gamma \\ \alpha_4 &= \beta\gamma \end{aligned} \tag{3.6}$$

Now, substituting for α_1, α_2, α_3, and α_4 from Equations 3.6 into Equation 3.2, observe that $a = C(x_1, x_2, x_3)$. □

Note that the above algorithm works for surfaces of any degree.

3.2 Bézier Representation for Cubic Surfaces

Using the skew-line coordinate system for a tetrahedron, this section describes the construction of a rectangular cubic surface patch embedded inside a tetrahedron by making two skew edges of the tetrahedron lie on the cubic surface patch. This is possible since in general there exists 27 lines on a cubic surface [SS87]. A natural parametrization for the patch in terms of the skew-line coordinates for the tetrahedron is then derived. Moreover, it is proved that the cubic patch has a rational biquadratic Bézier representation.

In the construction for the cubic surface patch, the weights on the layers 0 and 3 are chosen to be identically 0, i.e. $b_{i_1 i_2 i_3 i_4} = 0$ for $i_3 + i_4 = 0$ and $i_3 + i_4 = 3$. These choices are equivalent to the assertion that the edges $p_1 p_2$ and $p_3 p_4$ of the tetrahedron lie on the cubic surface patch. Because of these choices, $a_0 = a_3 = 0$ in the deCasteljau algorithm, and therefore, after the substitution $a = C(x_1, x_2, x_3) = 0$ Equation 3.5 for the zero contour of the surface yields the simple relationship $\gamma = \frac{a_1}{a_1 - a_2}$. By substituting for α_1, α_2, α_3, and α_4 from Equations 3.6 and substituting for γ from the above relationship into Equation 3.1, the following parametrization for the cubic surface patch in terms of the skew-line coordinates for the tetrahedron is obtained:

$$q(\alpha, \beta) = \frac{a_1((1-\beta)p_3 + \beta p_4) - a_2((1-\alpha)p_1 + \alpha p_2)}{a_1 - a_2} \tag{3.7}$$

One can also visualize this parametrization by observing that, in general, a straight line intersects an implicit cubic surface at three points [SR49]. In Figure 3.2, since P and Q lie on the cubic surface, the straight line PQ must intersect the cubic surface at a third point R, whose barycentric coordinate γ can be represented in terms of the barycentric coordinates α and β of P and Q, yielding the desired parametrization.

Now the theorem that this cubic patch with implicit trivariate representation also has a rational biquadratic Bézier representation, is to be established. For this purpose, the following notation for the control points of a rational biquadratic Bézier patch is introduced. The four corner control points are labeled as follows: $v_{13} = c_{00}, v_{14} = c_{02}, v_{23} = c_{20}$, and $v_{24} = c_{22}$; the control point v_{ij} lies on the edge $p_i p_j$. Let d_{ij} be the weights associated with the control points v_{ij}. The four face control points are labeled as follows: $v_{134} = c_{01}, v_{312} = c_{10}, v_{234} = c_{21}$, and $v_{412} = c_{12}$; the control point v_{ijk} lies on the face p_{ijk}. Let d_{ijk} be the weights associated with the control points v_{ijk}. The remaining ninth interior control point is labeled as $v_{1234} = c_{11}$. The weight associated with the control point v_{1234} is d_{1234}. Moreover, the weights associated with the control points c_{ij} are also labeled as w_{ij}. The above labeling scheme is illustrated in Figure 3.3.

The double labeling scheme and the following functions enable us to write the expressions for the control points and the associated weights in a compact form and

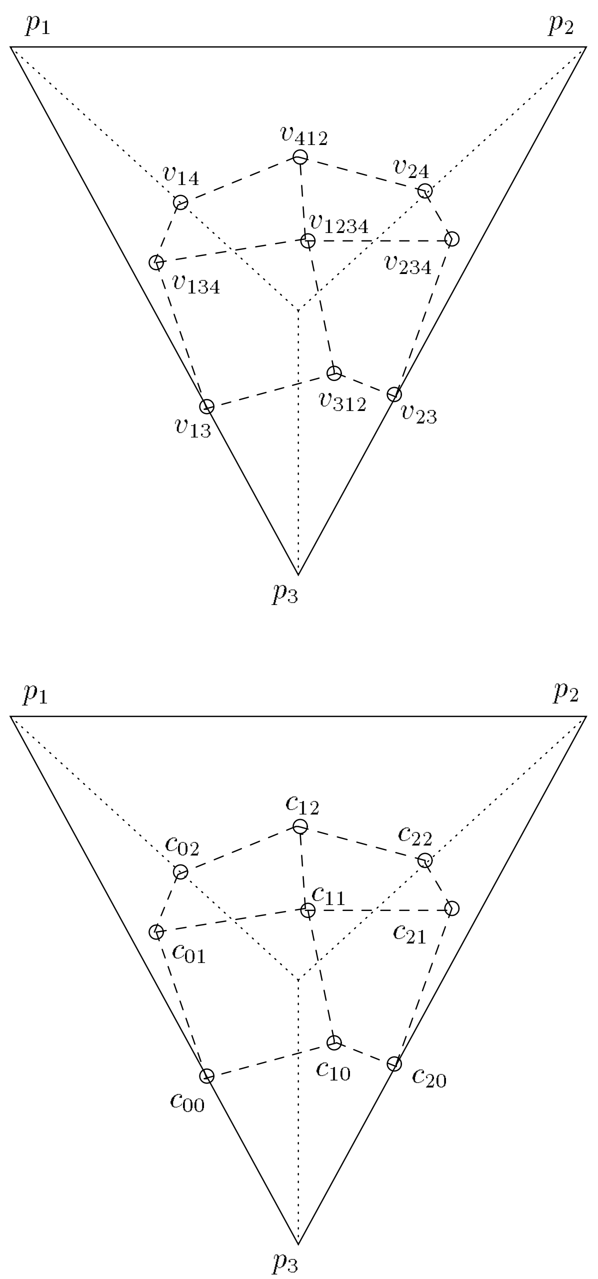

Figure 3.3 Control net for an implicit cubic patch

brings out the structural similarity of formulas for the four corner control points and for the four face control points. Let $e_i, 1 \le i \le 4$, be the 4-tuple with 1 in the i-th position and 0 everywhere else. For example, $e_2 = (0, 1, 0, 0)$. Let $\sigma_1(ij) = e_i + 2e_j$, and $\sigma_2(ij) = 2e_i + e_j$. Let $\delta_1(ijk) = e_i + e_j + e_k$, $\delta_2(ijk) = 2e_i + e_k$, and $\delta_3(ijk) = 2e_i + e_j$.

Theorem 3.2 Let S be a rectangular portion of a cubic surface patch in the implicit trivariate form, embedded within a tetrahedron, with weights $b_{i_1i_2i_3i_4} = 0$ for $i_3+i_4 = 0$ and $i_3+i_4 = 3$. Then S is a rational biquadratic Bézier surface. Moreover, if all the weights of layer one are positive and all the weights of layer two are negative, or vice-versa, then the surface patch lies inside the tetrahedron. The control points and the associated weights are given by the following formulas:

- Formulas for the corner control points and weights:
  For $< i, j >= (1,3), (1,4), (2,3)$, and $(2,4)$

$$\begin{aligned} v_{ij} &= \frac{-b_{\sigma_1(ij)}p_i + b_{\sigma_2(ij)}p_j}{d_{ij}}; \\ d_{ij} &= -b_{\sigma_1(ij)} + b_{\sigma_2(ij)}. \end{aligned}$$

- Formulas for the face control points and weights:
  For $< i, j, k >= (1,3,4), (2,3,4), (3,1,2)$, and $(4,1,2)$

$$\begin{aligned} v_{ijk} &= \frac{(-1)^{\frac{k}{2}}(-2b_{\delta_1(ijk)}p_i + b_{\delta_2(ijk)}p_j + b_{\delta_3(ijk)}p_k)}{2d_{ijk}}; \\ d_{ijk} &= \frac{(-1)^{\frac{k}{2}}}{2}(-2b_{\delta_1(ijk)} + b_{\delta_2(ijk)} + b_{\delta_3(ijk)}). \end{aligned}$$

- Formula for the interior control point and weight:

$$\begin{aligned} v_{1234} &= \frac{-b_{0111}p_1 - b_{1011}p_2 + b_{1101}p_3 + b_{1110}p_4}{2d_{1234}}; \\ d_{1234} &= \frac{1}{2}(-b_{0111} - b_{1011} + b_{1101} + b_{1110}). \end{aligned}$$

Proof Earlier, it was proved that any point on the rectangular portion of an implicit cubic surface patch satisfying the equations $b_{i_1i_2i_3i_4} = 0$ for $i_3 + i_4 = 0$ and $i_3 + i_4 = 3$ can be represented by Equation 3.7, in which each term is biquadratic in the numerator, but not in the denominator. To make the denominator biquadratic, the degree of a_1 is now raised by multiplying with $\beta + (1 - \beta)$, and the degree of a_2 is raised by multiplying with $\alpha + (1 - \alpha)$. A simple calculation yields the expressions for the control points and the weights mentioned in the statement of the theorem.

Furthermore, if all the weights of layer one are positive, and all the weights of layer two are negative, or vice-versa, it is clear from the expression for the control points v_{ij} and v_{ij} that they are obtained from convex combination of corresponding vertices of the tetrahedron. In particular, the corner control points v_{ij} on the edges p_{ij} are obtained from a convex combination of the vertices p_i and p_j. Similarly, the control points v_{ijk} are obtained from a convex combination of the vertices p_i, p_j, and p_k. Finally, the control point v_{1234} is a convex combination of the vertices of the tetrahedron and hence lies inside the tetrahedron. Because of the convex hull property of the rational biquadratic Bézier representation, the rectangular portion of the cubic patch lies inside the convex hull of its control points and hence inside the tetrahedron. $\square$

Corollary 3.1 The rectangular cubic surface patch S and the control net formed by v_{13}, v_{14}, v_{23}, v_{24}, v_{134}, v_{234}, v_{312}, v_{412}, and v_{1234}, share all of the properties of a standard Bézier surface.

- S interpolates v_{13}, v_{14}, v_{23}, and v_{24}.
- S is tangent to the triangle $v_{13}v_{134}v_{312}$ at v_{13}, $v_{23}v_{234}v_{312}$ at v_{23}, $v_{14}v_{134}v_{412}$ at v_{14}, and $v_{24}v_{234}v_{412}$ at v_{24}.
- S lies in the convex hull of the control net.

This corollary follows from the standard properties of a rational Bézier surface.

3.3 A Completeness Theorem

Theorem 3.3 Let S be a rectangular portion of a non-singular cubic surface Q in the rational biquadratic Bézier form such that the planes of the boundary curves form a tetrahedron. Then S can be represented in the implicit trivariate form using the construction outlined earlier in Section 3.1.

Proof Let $q(\alpha, \beta)$ be the rational biquadratic parametrization of the rectangular portion S of a non-singular cubic surface Q. The domain rectangle is defined by $0 \leq \alpha \leq 1$, $0 \leq \beta \leq 1$. Consider the one parameter family $R(t)$ of the image of the isoparameter lines $\alpha = t, 0 \leq t \leq 1$ on S. Since the patch has a rational biquadratic parametrization, $R(t)$ is a curve of degree at most two for all t. However, except for finitely many $t's$, $R(t)$ is a curve of exactly degree two, because a non-singular cubic surface has exactly 27 lines and no more [Har77]. Therefore, except for finitely many $t's$, $R(t)$ being a quadratic curve, lies in a plane $P(t)$. By taking the limit of $P(t)$ as t approaches one of the finitely many missing $t's$, a one parameter family of planes $P(t)$ for *all* $t, 0 \le t \le 1$, is obtained. $P(0)$ and $P(1)$ are two boundary planes of the tetrahedron.

By Bezout's theorem, $P(t)$ intersects the cubic surface Q in a curve of degree 3. However, for almost all $t's$, $P(t)$ intersects S in a quadratic curve, and therefore intersects Q in a remaining straight line $L(t)$. This one parameter family of straight lines $L(t)$ must be identical since there are no more than 27 lines on the smooth surface Q. Again, by the limit argument given before, the one parameter family of planes P(t) intersects the cubic surface Q in the same straight line $L = L(t)$ for *all* $t, 0 \le t \le 1$. Therefore, the line L must be the straight line $P(0) \cap P(1)$, which is one of the edges of the tetrahedron. Similarly, the opposite skew edge of the tetrahedron lies on the cubic surface Q. This implies that the weights $b_{i_1 i_2 i_3 i_4} = 0$ for $i_3 + i_4 = 0$ and $i_3 + i_4 = 3$ in the implicit trivariate representation of S. It now follows that S can be represented in the implicit trivariate form via the construction, outlined earlier in Section 3.1. □

3.4 Designing with Cubic Surface Patches

This section describes a method of designing rectangular cubic surface patches by specifying a rectangular biquadratic Bézier control net embedded inside a tetrahedron. Let p_1, p_2, p_3, and p_4 be the vertices of a tetrahedron. The control net is designed as follows:

1. Specify the four corner control points v_{ij} along the edges p_{ij} for $(i, j) = (1, 2)$, $(1, 3)$, $(2, 3)$ and $(2, 4)$. Let λ_{ij} be the barycentric coordinates of the control point v_{ij} on the straight line segment $p_i p_j$ so that the following equations hold for these pairs of (i, j):
$$v_{ij} = \lambda_{ij} p_i + (1 - \lambda_{ij}) p_j$$

2. Specify the four face control points v_{ijk} on the faces p_{ijk} for $(i, j, k) = (1, 3, 4)$, $(2, 3, 4)$, $(3, 1, 2)$, and $(4, 1, 2)$. Let the straight line $p_i v_{ijk}$ intersect the edge $p_j p_k$ at p_{jk}. Let λ_{ijk} be the barycentric coordinates of v_{ijk} on the straight line segment $p_i p_{jk}$, and let β_{ijk} be the barycentric coordinates of p_{jk} on the

straight line segment $p_j p_k$ so that the following equations hold for these triplets of (i, j, k):

$$v_{ijk} = \lambda_{ijk} p_i + (1 - \lambda_{ijk})(\beta_{ijk} p_j + (1 - \beta_{ijk}) p_k)$$

In addition, these control points need to satisfy a projective constraint specified in Equation 3.8 later.

This design for the face $p_1 p_3 p_4$ is illustrated in Figure 3.4A. Given this design of the rectangular biquadratic control net embedded within a tetrahedron, a cubic surface patch with an implicit trivariate representation is now constructed. The construction completely determines the remaining ninth interior control point and all the weights in the rational biquadratic Bézier representation of the patch.

Let us define the following ratios:

$$\begin{aligned} r_{ij} &= \frac{\lambda_{ij}}{1 - \lambda_{ij}} \\ r_{ijk} &= \frac{\lambda_{ijk}}{1 - \lambda_{ijk}} \\ s_{ijk} &= \frac{\beta_{ijk}}{1 - \beta_{ijk}} \end{aligned}$$

Theorem 3.4 Given the design of the rectangular biquadratic Bézier control net, as specified in steps 1 and 2 above, there exists a cubic surface patch, which admits a rational biquadratic Bézier representation with the given net as the control net. The patch is defined as the zero contour of a trivariate cubic polynomial defined over the tetrahedron $p_1 p_2 p_3 p_4$ in the implicit trivariate representation with the following weights: $b_{i_1 i_2 i_3 i_4} = 0$ for $i_3 + i_4 = 0$ and $i_3 + i_4 = 3$;

$$\begin{array}{lll} b_{2010} = -2(1 - \beta_{134}) & b_{2001} = -2\beta_{134} & b_{1011} = r_{134} \\ b_{1020} = 2(1 - \beta_{134}) r_{13} & b_{1002} = 2\beta_{134} r_{14} & \\ b_{0 i_1 i_2 k} = \kappa b_{i 0 j k} & \kappa = \frac{1-\beta_{134}}{1-\beta_{234}} \frac{r_{13}}{r_{23}} s_{312} & \\ b_{1110} = -\frac{1-\beta_{134}}{1-\beta_{312}} r_{13} r_{312} & b_{1101} = -\frac{\beta_{134}}{1-\beta_{412}} r_{14} r_{412} & \end{array}$$

Proof

The design of the Bézier control net on the face $p_1 p_3 p_4$ of the tetrahedron is shown in Figure 3.4A. First, an implicit *quadratic* curve, which also has a rational quadratic Bézier representation with the given net as the control net, is constructed on the face

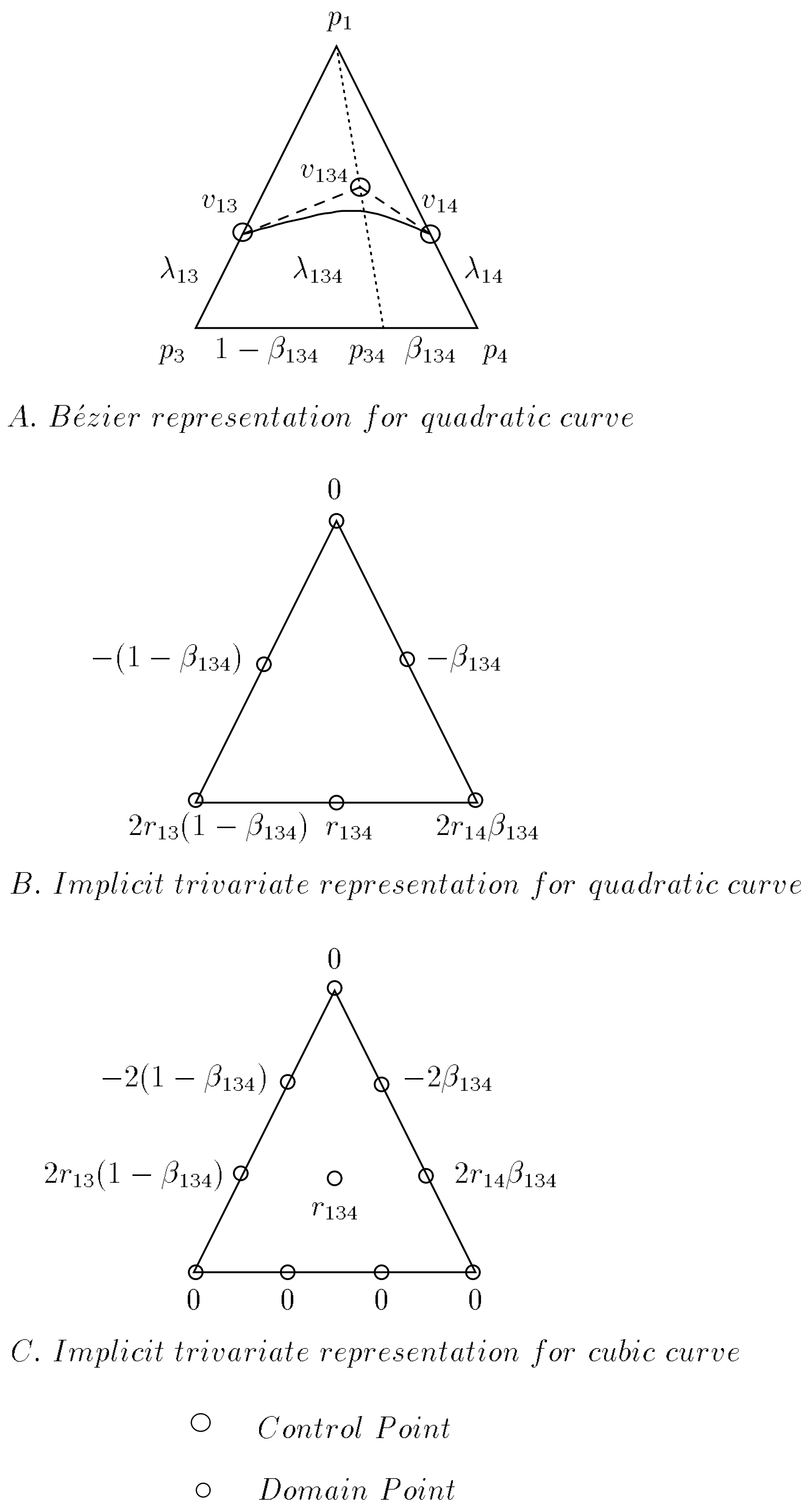

Figure 3.4 Construction of implicit cubic curve

$p_1p_3p_4$ of the tetrahedron. An outline of the construction is given below. The details of this construction are provided in Chapter 2. The designed curve is an implicit quadratic and has the following unique (up to a multiplicative constant) implicit trivariate representation :

$$\begin{array}{lll} b_{002} = 0 & b_{101} = -(1-\beta_{134}) & b_{011} = -\beta_{134} \\ b_{110} = r_{134} & b_{200} = 2(1-\beta_{134})r_{13} & b_{020} = 2\beta_{134}r_{14} \end{array}$$

This representation is shown in Figure 3.4B. This implicit quadratic curve, when multiplied by the straight line containing the edge p_3p_4, yields a cubic curve with an implicit trivariate representation, which is shown in Figure 3.4C.

This construction is repeated on each of the four faces of the tetrahedron yielding four cubic curves lying on the respective faces. In order for these cubic curves to form a part of a single cubic surface patch, the implicit trivariate representation must agree on the common edges. The compatibility conditions are shown in Figure 3.5. There still exists the choice of multiplying the weights in the implicit trivariate representation of the cubic curves on each face by a constant. Let ρ_{ijk} be the constant with which the coefficients of the cubic curve lying on the face $p_ip_jp_k$ are multiplied.

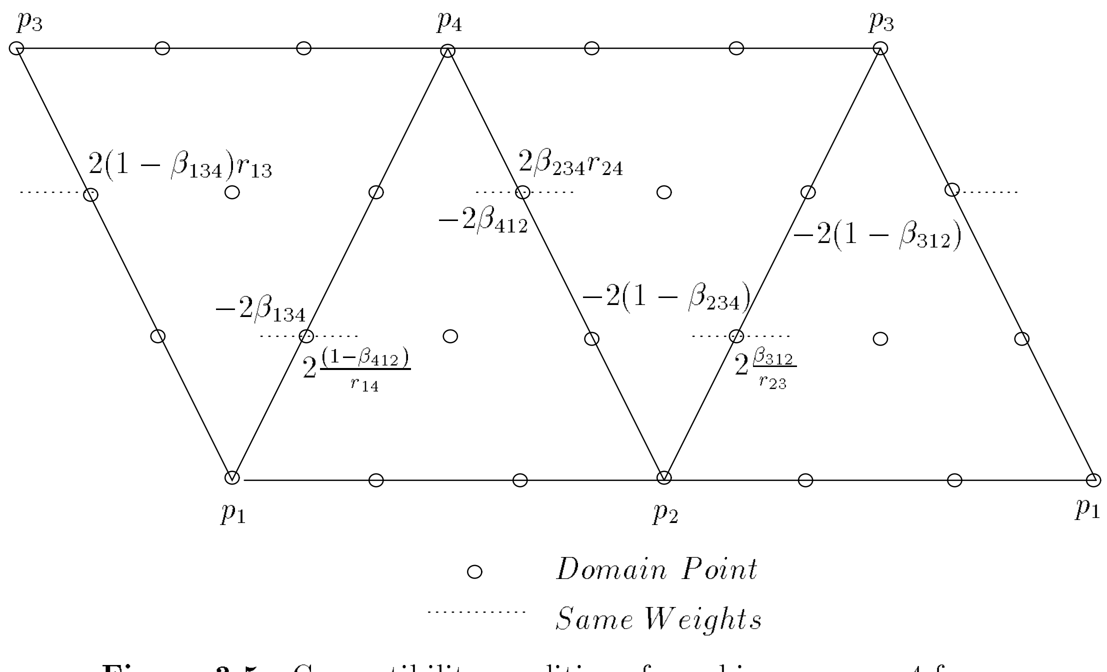

Figure 3.5 Compatibility conditions for cubic curves on 4 faces

The following compatibility conditions are obtained along the common edges:

$$\begin{aligned}
\rho_{134}(1-\beta_{134})r_{13} &= -\rho_{312}(1-\beta_{312}) \\
\rho_{312}\beta_{312}\frac{1}{r_{23}} &= -\rho_{234}(1-\beta_{234}) \\
\rho_{234}\beta_{234}r_{24} &= -\rho_{412}\beta_{412} \\
\rho_{412}(1-\beta_{412})\frac{1}{r_{14}} &= -\rho_{134}\beta_{134}.
\end{aligned}$$

By multiplying the above equations, the one and the only projective constraint is obtained as:

$$[r_{13}, r_{14}, r_{23}, r_{24}] = [s_{134}, s_{234}, s_{312}, s_{412}] \tag{3.8}$$

where the projective ratio of four real numbers a, b, c, and d, denoted by $[a, b, c, d]$ is defined as follows:

$$[a, b, c, d] = \frac{\frac{a}{b}}{\frac{c}{d}}.$$

The weights $b_{i_1i_2i_3i_4}$ in the implicit trivariate representation of the cubic surface patch are now uniquely determined up to a multiplicative constant. These weights are easily computed by multiplying the weights in the implicit trivariate representation of the cubic curve lying on face $p_ip_jp_k$ with the constants ρ_{ijk}, and computing the constants ρ_{312}, ρ_{234}, and ρ_{412} in terms of the constant ρ_{134} from the compatibility conditions. Thus, for example, the coefficients $b_{1110} = \rho_{312}r_{312}$, and $b_{1101} = \rho_{412}r_{412}$. The expressions for the weights $b_{i_1i_2i_3i_4}$ mentioned in the statement of the theorem correspond to the choice $\rho_{134} = 1$.

Now, since $b_{i_1i_2i_3i_4} = 0$ for $i_3 + i_4 = 0$, and $i_3 + i_4 = 3$, it follows from Theorem 3.2 that this cubic surface patch admits a rational biquadratic Bézier representation. It also follows that the patch admits the given control net in each of the faces of the tetrahedron as the control net for the designed patch. □

If the designer specifies all the control points to lie inside the tetrahedron, which is equivalent to the assertion that all the ratios r_{ij}, r_{ijk} and s_{ijk} lie between 0 and 1, then all the weights $b_{i_1i_2i_3i_4}$ of layer one are positive and all the weights of layer two are negative. Therefore, by Theorem 3.2, the patch lies inside the tetrahedron. Furthermore, this design method has *linear precision*, meaning that if the designer specifies all the control points to lie in one plane, then the designed cubic surface is the same plane.

3.5 Conversion Between Representations

Given the design specifications of the control net of a biquadratic Bézier control polyhedron, subject to the constraint 3.8, the formulas for the implicit weights in the trivariate implicit representation of the cubic surface patch were derived in the previous section. To compute the rational biquadratic Bézier representation, the formulas, given in the statement of Theorem 3.2, can be used. Thus, for example, the formula for the only yet unspecified ninth interior control point v_{1234} is:

$$v_{1234} = \frac{\rho_{234}r_{234}p_1 + \rho_{134}r_{134}p_2 - \rho_{412}r_{412}p_3 - \rho_{312}r_{312}p_4}{\rho_{234}r_{234} + \rho_{134}r_{134} - \rho_{412}r_{412} - \rho_{312}r_{312}}$$

Given the trivariate implicit representation of the cubic surface patch, the formulas in the statement of Theorem 3.2 can be used to compute the parametric Bézier representation of the patch.

This chapter has described a method of modeling with implicit cubics by prescribing a Bézier biquadratic control net, within a tetrahedron, satisfying a projective constraint. The method is complete in the sense that any rectangular implicit cubic patch that has a biquadratic parametrization can be modeled in this way. All of the standard Bézier techniques, such as the deCasteljau algorithm, and many of the useful properties of Bézier representations, like the convex hull property, are preserved. Because it is easy to compute either the parametric or the implicit representation of a patch from a given control net, the method allows for multiple representations of surface patches thus combining the best features of both representations.

Chapter 4

Filling N-sided Holes

The n-sided hole problem arises when n patches surround a hole. The objective is to construct a surface patch that fills the hole and meets the surrounding surface patches smoothly. We shall restrict our attention to the case where the surrounding patches have polynomial or rational parametric representations. Thus, the problem is to construct a smooth interpolant for a parametrically defined vector-valued function and its derivatives on the boundary of a polygonal domain. These interpolants on a fixed domain generate a surface patch, which then can be fitted together with adjacent patches to create a smooth surface.

After a review of Bézier simplexes in Section 4.1, of barycentric coordinates for a convex polygon in Section 4.2, and of the basic properties of S-patches in Section 4.3, the n-sided hole problem is described in Section 4.4. Sections 4.5 and 4.7 introduce the rationally controlled Bézier representation and rational blending functions, which are used to describe the general solution presented in Section 4.8. Section 4.6 motivates the proposed solution by giving an example of a 2-sided hole problem from the curve case. Finally, Section 4.9 presents many examples of filling n-sided holes using the techniques described in this paper including the Gregory patch, Nielson-Foley patch, Barnhill-Worsey patch and Brown's square.

4.1 Bézier Simplexes

This section describes the multi-index notation, and reviews the definition of a Bézier simplex. A detailed treatment of these topics is given in [dB87, LD89]. A multi-index will be denoted by an italic character with a diacritical arrow at the top, as in $\vec{i}$. A multi-index is a tuple of non-negative integers, for instance, $\vec{i} = (i_0, \ldots, i_n)$. The norm of a multi-index $\vec{i}$, denoted by $\mid \vec{i} \mid$ is defined to be the sum of the components of $\vec{i}$. By setting $\vec{i} = (i_0, \ldots, i_n)$ and requiring that $\mid \vec{i} \mid = d$, the n-variate Bernstein polynomials of degree d can be defined by

$$B^d_{\vec{i}}(\alpha_0, \ldots, \alpha_n) = \begin{pmatrix} d \\ \vec{i} \end{pmatrix} \alpha_0^{i_0} \cdots \alpha_n^{i_n}$$

where $\begin{pmatrix} d \\ \vec{i} \end{pmatrix}$ is the multinomial coefficient defined by

$$\begin{pmatrix} d \\ \vec{i} \end{pmatrix} = \frac{d!}{i_0! \cdots i_n!}$$

and $\alpha_0, \ldots, \alpha_n$ are non-negative real numbers that sum to one. A Bézier simplex is a polynomial map B of degree d from an affine space A_1 of dimension n to an affine space A_2 of arbitrary dimension s, and is represented in the Bernstein-Bézier basis with respect to a domain simplex Δ in A_1 as follows:

$$B(v) = \sum_{\vec{i}} C_{\vec{i}} B^d_{\vec{i}}(\alpha_0(v), \ldots, \alpha_n(v)) \tag{4.1}$$

where $\alpha_0, \ldots, \alpha_n$ are the barycentric coordinates of a point v relative to the domain simplex Δ. The summation in equation (4.1) is taken over all multi-indexes where the norm is equal to the degree of the Bernstein polynomials, viz. $\mid \vec{i} \mid = d$. It is well-known that such a representation for a polynomial map is unique. The points $C_{\vec{i}}$ are individually referred to as Bézier control points and collectively referred to as the Bézier control net for B relative to Δ. Note that for $n = 1$, $s = 3$, Bézier simplexes are Bézier curves in 3-space and for $n = 2$, $s = 3$, they are triangular Bézier patches.

4.2 Barycentric Coordinates

This section reviews the definition of barycentric coordinates for a regular convex polygon. Let P be a convex n-gon, defined by the intersection of n halfspaces, so that

$$P = \bigcap_{k=1}^{n} (u_k \geq 0)$$

Let $p_1, \cdots, p_n$ denote the vertices and $E_1, \cdots, E_n$ denote the edges of the polygon P. Let $u_k = 0$ be the equation of the edge E_k, which goes from the vertex p_k to the vertex p_{k+1}. In the previous statement and in what follows, all indexes are to be interpreted in a cyclic fashion, so that every index k is mapped into the range $1, \cdots, n$ by the formula $[(k-1) \;\; mod \;\; n] + 1$.

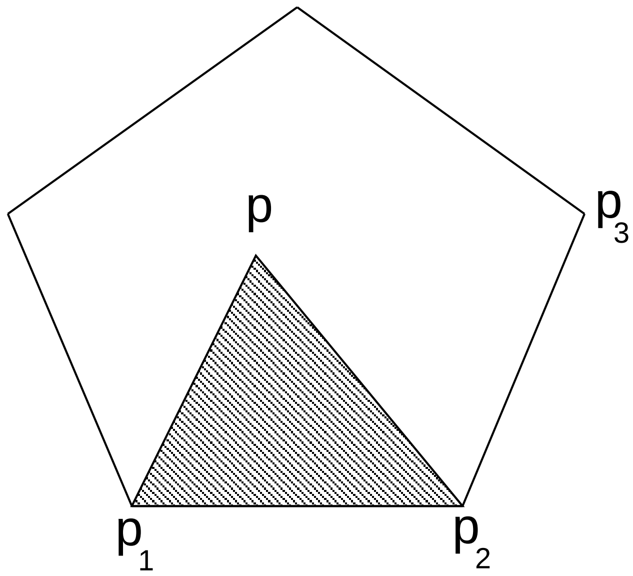

Figure 4.1 Geometry of the functions σ_i used to construct barycentric coordinates

Let $\sigma_k(p)$ denote the signed area of the triangle pp_kp_{k+1} as shown in Figure 4.1, where the sign is chosen to be positive for points inside the polygon P. Since $\sigma_k(p)$ and $u_k(p)$ are both linear functions, both vanish at p_k and p_{k+1}, and both are positive inside the polygon P, they must differ only by a positive multiplicative constant. Therefore, $\sigma_k(p) = \xi_k u_k(p)$. Let $\pi_k(p)$ denote the product of all σ's except for $\sigma_{k-1}(p)$ and $\sigma_k(p)$; that is,

$$\pi_k(p) = \sigma_1(p) \cdots \sigma_{k-2}(p)\sigma_{k+1}(p) \cdots \sigma_n(p), k = 1, \cdots, n$$

and therefore,

$$\pi_k(p) = \rho_k u_1(p) \cdots u_{k-2}(p)u_{k+1}(p) \cdots u_n(p), k = 1, \cdots, n \tag{4.2}$$

where $\rho_k = \xi_1 \cdots \xi_{k-2}\xi_{k+1} \cdots \xi_n$. Let

$$\gamma_k(p) = \frac{\pi_k(p)}{\sum_{k=1}^{n} \pi_k(p)} \tag{4.3}$$

Theorem[LD89]: The functions $\gamma_k(p)$ for a regular convex n-gon satisfy the following properties:

1. The functions $\gamma_k(p)$ are nonnegative whenever p is inside the polygon P, that is, $\gamma_k(p) \geq 0 \ \forall \ p \epsilon P, \ k = 1, \cdots, n$.
2. The functions $\gamma_k(p)$ form a partition of unity, that is, $\sum_{k=1}^{n} \gamma_k(p) = 1 \ \forall \ p \epsilon P$.
3. $p = \sum_{i=1}^{n} \gamma_i(p) p_i$.

Remarks: The first two properties follow easily from the definition of the $\gamma_k(p)$. The third property generalizes a property satisfied by the barycentric coordinates for the triangles, and is referred to as the pseudo-affine property by Loop and DeRose [LD89]. The functions $\gamma_k(p)$ will be referred to as barycentric coordinates for the convex polygon P. More generally, any set of functions satisfying properties 1 to 3 above for a fixed domain will be referred to as barycentric coordinates for that domain.

4.3 S-patches

S-patches build on the theory of Bézier simplexes and barycentric coordinates. A regular n-sided S-patch is a mapping from a regular convex n-gon P and is constructed conceptually in two phases: first, the polygon P is embedded into an intermediate domain simplex $\Delta = v_1, \cdots, v_n$ in an affine space of dimension $n-1$; next, a Bézier simplex is created using Δ as its domain; finally, S is defined as the composition of the embedding and the Bézier simplex. That is, if $L : P \rightarrow \Delta$ represents the embedding, and if $B : \Delta \rightarrow R^3$ is the Bézier simplex, then

$$S(p) = B \odot L(p), \quad p \epsilon P,$$

as indicated in Figure 4.2.

Given the barycentric coordinates $\gamma_k(p)$ of a point p inside the regular convex n-gon P, the embedding L is defined as

$$L(p) = \sum_{k=1}^{n} \gamma_k(p) v_k.$$

If $C_{\vec{i}}$ denotes the control net of a Bézier simplex B, then an S-patch is defined as

$$S(p) = B \odot L(p) = \sum_{\vec{i}} C_{\vec{i}} B_{\vec{i}}^{d}(\gamma_1(p), \cdots, \gamma_n(p)).$$

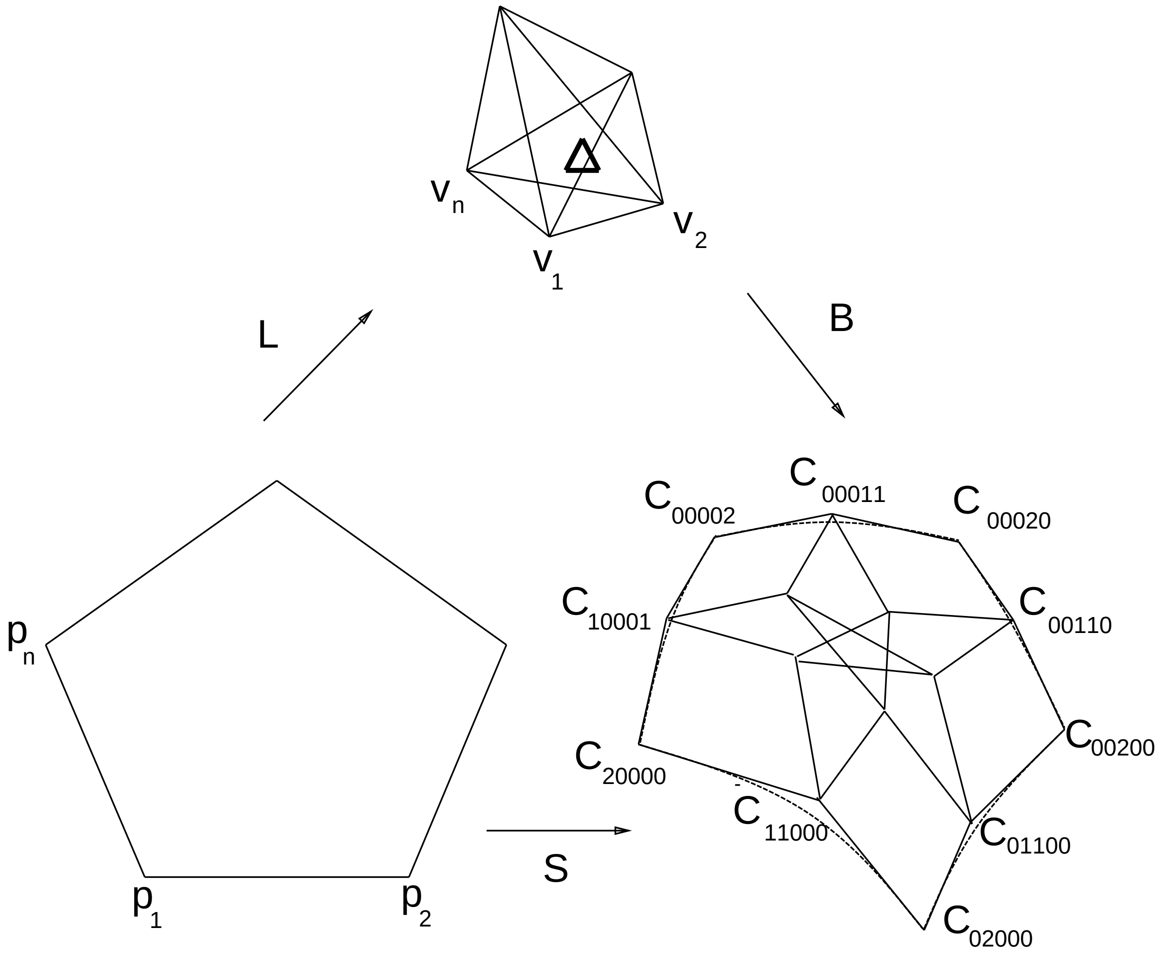

Figure 4.2 Schematic representation of S-patches

A rational S-patch is defined as

$$S(p) = \frac{\sum_{\vec{i}} w_{\vec{i}} C_{\vec{i}} B^d_{\vec{i}}(\gamma_1(p), \cdots, \gamma_n(p))}{\sum_{\vec{i}} w_{\vec{i}} B^d_{\vec{i}}(\gamma_1(p), \cdots, \gamma_n(p))}$$

where the weights $w_{\vec{i}}$ associated with the control points $C_{\vec{i}}$ are generally chosen to be positive.

The integer d in the above equation is known as the depth of the S-patch. Since the barycentric coordinates $\gamma_k(p)$ are rational functions of degree $n-2$, the S-patch is a rational parametric patch of degree $d(n-2)$. The control net $C_{\vec{i}}$ is referred to as the control net of S, an example of which is shown in Figure 4.2. The S-patch representation has the following useful properties:

- S-patches can be evaluated by first evaluating L, and then evaluating B using the deCasteljau algorithm.
- S-patches lie within the convex hull of their control nets.
- An S-patch of depth d can be described as an S-patch of depth $d+1$ using a depth elevation algorithm.
- Boundary curves are in Bézier form. For example, the boundary curve corresponding to the bottom boundary of the control net is a quadratic Bézier curve whose control points are C_{20000}, C_{11000}, and C_{02000}. Control points such as these are called *boundary control points*.
- The tangent plane variation along a boundary curve is determined entirely by the control points, which are at unit distance or less from that boundary. The distance $g_k(\vec{i})$ of a control point $C_{\vec{i}}$ from the k-th boundary is defined to be $d - (i_k + i_{k+1})$. Referring to Figure 4.2, the tangent plane along the bottom boundary ($k = 1$) of the control net is entirely determined by the boundary control points, which are at distance 0 from the boundary, and by the control points C_{10001}, C_{10010}, C_{10100}, C_{01001}, C_{01010}, and C_{01100}, which are at unit distance from the boundary.
- The h-th cross-boundary derivative along the k-th boundary curve is determined entirely by the control points, which are at a distance h units or less from that boundary, that is, $g_k(\vec{i}) \leq h$. These control points will be referred to as the first h *layers* of control points across the k-th boundary. Therefore, in what follows, the user will be required to prescribe the first h cross-boundary derivatives along a boundary curve in terms of the S-patch control points, which are at a distance h units or less from that boundary. If the first h cross-boundary derivatives across a boundary arise from an adjacent Bézier triangle, an algorithm to compute the regular S-patch control points at a distance h units or less from the boundary is described in Loop and DeRose[LD89]. It is only in this algorithm that the properties of the barycentric coordinates of the regular convex polygon are used. Since barycentric coordinates have now been developed for any convex polytope in arbitrary dimension [War92], the construction described later in this chapter is generalizable to higher dimensions.

- When $n = 3$, that is, for triangular domains, an S-patch is simply a triangular Bézier surface patch. In other words, regular S-patches generalize triangular Bézier surface patches. When $n = 4$, that is, for rectangular domains, an S-patch is simply a bi-d-ic tensor product Bézier surface patch. In other words, S-patches also generalize rectangular tensor product Bézier surface patches.

- More significantly, there exists an algorithm for representing an n-sided regular S-patch as m-sided regular S-patch using multiprojective blossoming [LD89, DeR88, DGHM92]. In particular, an n-sided regular S-patch can be represented as a collection of triangular Bézier surface patches.

4.4 The N-sided Hole Problem

The n-sided hole problem arises in situations as the one shown in Figure 4.3, where polynomial patches surround an n-sided hole. The objective is to construct a surface patch that fills the hole and meets the surrounding patches with C^h continuity. We shall consider the more general case where the surface patch meets the surrounding k-th patch with C^{a_k} continuity. In other words, the surface patch meets different surrounding patches with different orders of continuity.

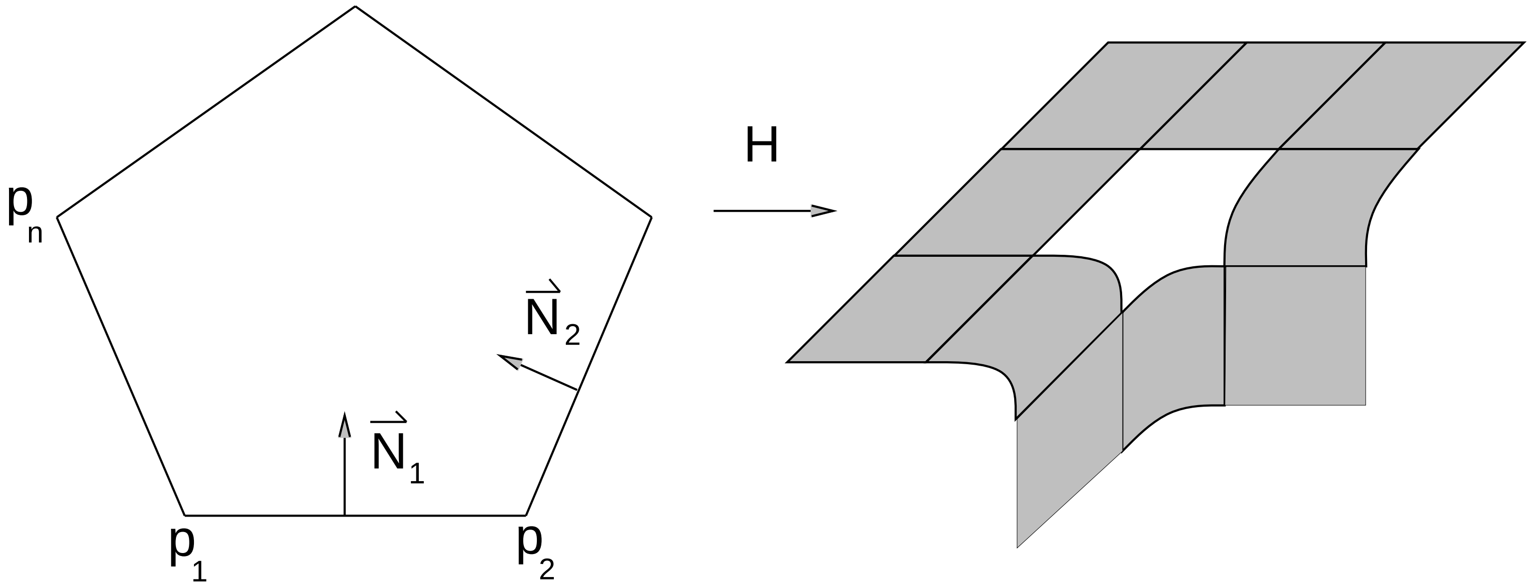

Figure 4.3 The n-sided hole problem

Referring to Figure 4.3, the hole to be filled is assumed to be surrounded by n patches $F_1, \cdots, F_n$. The domain polygon P of H is a regular convex n-gon. Let $\vec{N}_k$ be a unit vector normal to the edge E_k. The following information is prescribed in the n-sided hole problem:

- The C^{a_k} cross-boundary derivatives of the hole across the kth boundary are given by the S-patch control points $C^k_{\vec{i}}$, which are at a distance a_k units or less from the k-th boundary, that is, $g_k(\vec{i}) \leq a_k$.

4.5 Rationally Controlled Bézier Representation

This chapter describes a solution which fills the hole with a single rationally controlled S-patch. The rationally controlled representation is compact and is particularly suitable for efficient computation. This section introduces the rationally controlled Bézier representation. Any parametric polynomial curve $S(p)$ of degree d can be represented in Bézier form as

$$S(p) = \sum_{\vec{i}} C_{\vec{i}} B^d_{\vec{i}}(\gamma_1(p), \gamma_2(p))$$

where $\vec{i} = (i_1, i_2)$ with $|\vec{i}| = d$, and $\gamma_1(p)$ and $\gamma_2(p)$ are the barycentric coordinates of a point p belonging to the domain interval Δ, so that $\gamma_1(p), \gamma_2(p) \geq 0$, and $\gamma_1(p) + \gamma_2(p) = 1$. If the domain interval is the standard unit interval $[0, 1]$ and the point p on the interval has the coordinates t, then $\gamma_1(p) = t$, and $\gamma_2(p) = 1 - t$.

Any rational parametric curve $S(p)$ of degree d can be represented in rational Bézier form as

$$S(p) = \frac{\sum_{\vec{i}} C_{\vec{i}} w_{\vec{i}} B^d_{\vec{i}}(\gamma_1(p), \gamma_2(p))}{\sum_{\vec{i}} w_{\vec{i}} B^d_{\vec{i}}(\gamma_1(p), \gamma_2(p))}$$

where w_i are the weights associated with the control points.

A rational parametric curve $S(p)$ admits a *rationally controlled Bézier representation of degree d*, if it can be represented in the following form:

$$S(p) = \sum_{\vec{i}} C_{\vec{i}}(p) B^d_{\vec{i}}(\gamma_1(p), \gamma_2(p))$$

where the control points $C_{\vec{i}}(p)$ are rational functions of p. Similarly, an S-patch admits a rationally controlled S-patch representation of degree d, if it can be represented in the following form:

$$S(p) = \sum_{\vec{i}} C_{\vec{i}}(p) B^d_{\vec{i}}(\gamma_1(p), \cdots, \gamma_n(p))$$

where $C_{\vec{i}}(p)$ are rational functions of p. In this chapter, we will be interested in a very specific form for $C_{\vec{i}}(p)$, where $C_{\vec{i}}(p) = \sum_k f_{\vec{i}}^k(p) C_{\vec{i}}^k$. The control points $C_{\vec{i}}^k$ are the user-specified control points across the k-th boundary and the functions $f_{\vec{i}}^k(p)$ are rational blending functions.

Remark: It is clear that any rational parametric curve $S(p)$ (resp. S-patch) of degree d admits a rationally controlled Bézier representation of degree d. In general, a rationally controlled Bézier curve (resp. S-patch) of degree d is a rational curve (S-patch) of higher degree, which depends upon the degree of $C_{\vec{i}}(p)$. The advantage of this representation is precisely to represent curves (S-patches) of high rational degrees in terms of a compact, low degree, rationally controlled Bézier representation. Moreover, this representation arises naturally in the n-sided hole filling problem, and it provides some good intuition for the proposed solution.

4.6 Motivation

This section describes the motivation for introducing the rationally controlled Bézier representation by presenting an example from the curve case. The objective is to construct a curve that fills an interval, also referred to as a 2-sided hole, and meets the surrounding curves F_1 and F_2 with C^h continuity. We demonstrate how a rationally controlled Bézier curve can provide a naturally elegant solution to this problem. Section 4.8 generalizes these results to n-sided holes.

First, we consider the case $h = 1$, which is well-known as the Hermite interpolation problem. As a first solution, we seek a parametric curve of degree 3, which solves this problem. We assume that the user specifies the position and the derivative data at both ends of the curve in terms of the control points of the interval filling curve $H(t)$ of degree 3, as shown in Figure 4.4. In other words,

$$\begin{aligned}
F_1(1) &= C_{30}^1 \\
F_1'(1) &= 3(C_{21}^1 - C_{30}^1) \\
F_2(0) &= C_{03}^2 \\
F_2'(0) &= -3(C_{12}^2 - C_{03}^2)
\end{aligned}$$

The superscript k in $C_{\vec{i}}^k$ denotes the control point specified from the k-th end point. In this case, a well-known solution [Far88] to the 2-sided hole problem is given

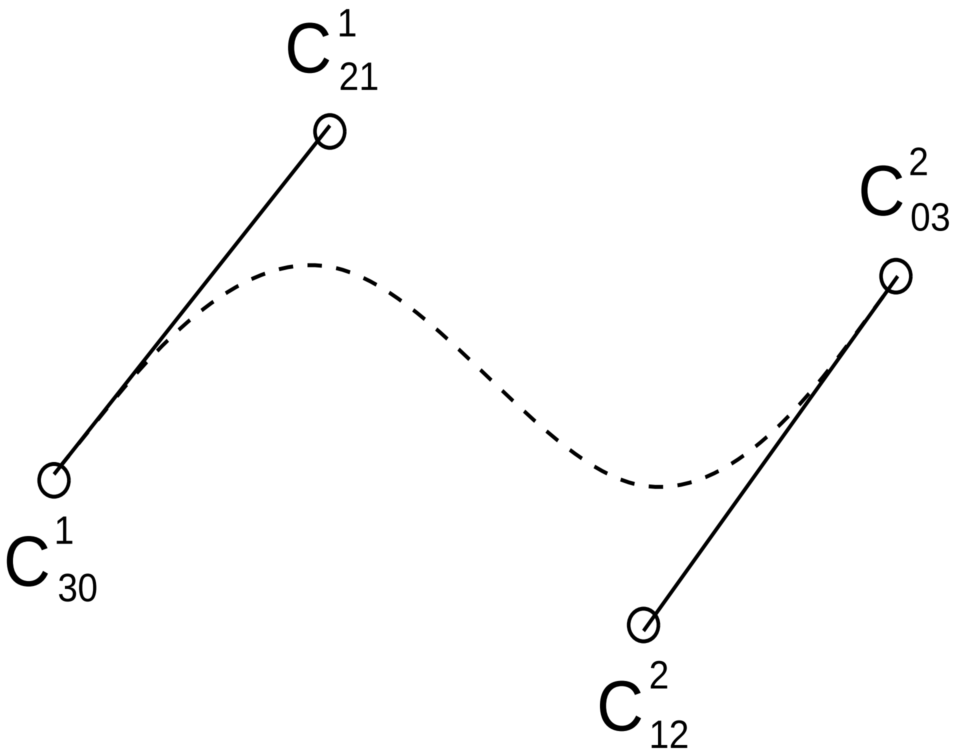

Figure 4.4 The 2-sided hole problem: C^1 cubic case

by the following parametric polynomial Bézier curve of degree 3:

$$H(t) = \sum_{|\vec{i}|=3} C_{\vec{i}} B^3_{\vec{i}}$$

where $C_{30} = C^1_{30}, C_{21} = C^1_{21}, C_{12} = C^2_{12}$, and $C_{03} = C^2_{03}$.

Now let us consider the case when the desired interval filling curve $H(t)$ of degree 2 is sought. In general, there does not exist a parametric polynomial curve of degree 2, which solves the Hermite interpolation problem. Therefore, what we seek is a rationally controlled Bézier curve of degree 2. Again, let us assume that the user specifies the position and the derivative data at the end points in terms of the control points of a curve of degree 2, as shown in Figure 4.5, so that the following equations hold:

$$\begin{aligned} F_1(1) &= C^1_{20} \\ F_1'(1) &= 2(C^1_{11} - C^1_{20}) \end{aligned}$$

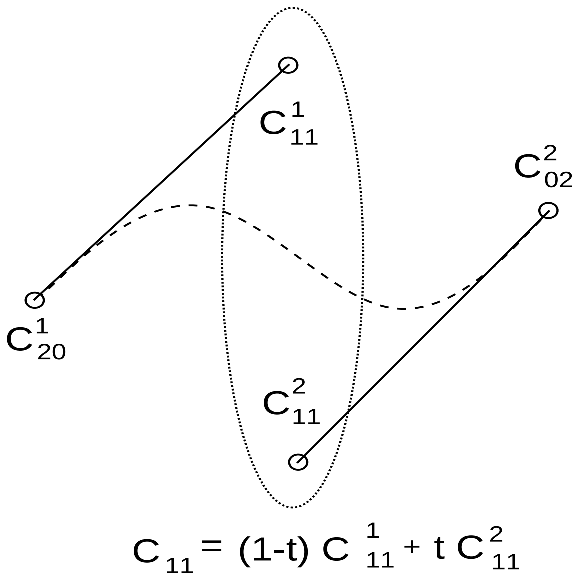

Figure 4.5 The 2-sided hole problem: C^1 quadratic case

$$\begin{aligned} F_2(0) &= C_{02}^2 \\ F_2'(0) &= -2(C_{11}^2 - C_{02}^2) \end{aligned}$$

The following rationally controlled Bézier curve of degree 2 solves the interval filling problem:

$$H(t) = \sum_{|\vec{i}|=2|} C_{\vec{i}} B_{\vec{i}}^2$$

where $C_{20} = C_{20}^1$, $C_{11} = \frac{(1-t)C_{11}^1 + tC_{11}^2}{(1-t)+t}$, and $C_{02}^2 = C_{02}$. In the preceding expression for C_{11}, even though the denominator sums to 1, we have written it this way to emphasize the structural similarity of this solution to the solution in the general case, presented in Section 4.8. A simple calculation and comparison with the first solution shows that the second solution is in fact identical to the first solution. Thus, the second solution, even though it is a rationally controlled Bézier curve of degree 2, it is in fact a parametric polynomial Bézier curve of degree 3. The control points in the rationally controlled representation are convex combinations of the user-specified control points from the two ends. Moreover, the rational blending functions, in this case simply $1-t$ and t, provide an intuitive way to blend the incompatible control points specified by the user.

In the general case of the interval filling problem, where the interval filling curve meets the surrounding curves with C^{a_k} continuity from the two ends for $k = 1, 2$, a rationally controlled Bézier curve of degree $d \geq max(a_1, a_2)$ can be constructed.

> **Theorem 4.1** Let the control points $C^k_{\vec{i}}$ with the distance $g_k(\vec{i}) \leq a_k$ for $k = 1, 2$ of a Bézier curve of degree $d \geq max(a_1, a_2)$ be prescribed at both ends. Then there exists a rationally controlled Bézier curve of degree d, which solves the interval filling problem; that is, it meets the surrounding curves with C^{a_k} continuity.

Proof This is an immediate consequence of the more general Theorem 4.2, proved in Section 4.8. An explicit solution is given in Section 4.9. □

4.7 Rational Blending Functions

This section describes certain rational blending functions, which play a crucial role in constructing a solution to the problem of filling n-sided holes. We shall consider the general case where the n-sided hole, surrounded by polynomial patches as shown in Figure 4.3, meets the surrounding k-th patch with C^{a_k} order of continuity. Thus, the user will be required to prescribe a_k layers of control points across the k-th boundary. Given an edge k and a multi-index $\vec{i}$, we associate rational blending functions $f^k_{\vec{i}}(p)$, which are then used to blend the user-specified control points $C^k_{\vec{i}}$ as follows:

$$C_{\vec{i}}(p) = \sum_{k=1}^{n} f^k_{\vec{i}}(p) C^k_{\vec{i}} \tag{4.4}$$

Let us introduce the set $S(\vec{i}) = \{k | g_k(\vec{i}) \leq a_k\}$. In other words, $S(\vec{i})$ is the collection of indices of those edges across which the user has prescribed control points associated with the multi-index $\vec{i}$. Also, let $h^k_{\vec{i}}(p) = u_k^{a_k+1-g_k(\vec{i})}(p)$. Recall that $u_k = 0$ is the equation of the edge E_k. Therefore, the functions $h^k_{\vec{i}}$ are chosen to be suitable powers of the equations of the edges, where the power depends upon a_k, the prescribed number of cross-boundary derivatives across the k-th boundary and $g_k(\vec{i})$, the distance of the $\vec{i}$-th control point from the k-th boundary. We define the rational blending functions for $k \epsilon S(\vec{i})$ as follows:

$$f^k_{\vec{i}}(p) = \frac{\hat{f}^k_{\vec{i}}(p)}{\sum_{j \epsilon S(\vec{i})} \hat{f}^j_{\vec{i}}(p)} \tag{4.5}$$

$$\hat{f}_{\vec{i}}^{k}(p) = z_{\vec{i}}^{k}(p) \prod_{j \epsilon S(\vec{i})-k} h_{\vec{i}}^{j}(p) \tag{4.6}$$

where the weights $z_{\vec{i}}^{k}(p)$ are chosen to be user-specified positive functions.

Let us consider the example of a 3-sided hole, which is required to meet the surrounding patches with C^2 continuity on all the three sides, as shown in Figure 4.9. The objective is to construct a rationally controlled quartic triangular Bézier surface patch. Since $g_1(211) = 1$, $g_2(211) = 2$, $g_3(211) = 1$, and $a_k = 2$ for $k = 1, \cdots, 3$, $h_{111}^1 = u_1^2$, $h_{111}^2 = u_2$, and $h_{111}^3 = u_3^2$. Moreover, $S(211) = \{1, 2, 3\}$. If the weight functions $z_{\vec{i}}^{k}(p)$ are chosen to be constant unit functions, then $\hat{f}_{211}^1 = u_2 u_3^2$, $\hat{f}_{211}^2 = u_3^2 u_1^2$, and $\hat{f}_{211}^3 = u_1^2 u_2$. Therefore, the rational blending functions are:

$$f_{211}^1 = \frac{u_2 u_3^2}{u_2 u_3^2 + u_3^2 u_1^2 + u_1^2 u_2} \tag{4.7}$$

$$f_{211}^2 = \frac{u_3^2 u_1^2}{u_2 u_3^2 + u_3^2 u_1^2 + u_1^2 u_2} \tag{4.8}$$

$$f_{211}^3 = \frac{u_1^2 u_2}{u_2 u_3^2 + u_3^2 u_1^2 + u_1^2 u_2} \tag{4.9}$$

Remark: Note that when $g_k(\vec{i}) > a_k$ for all $k = 1, \cdots, n$, then the control point $C_{\vec{i}}^{k}$ associated with the multi-index $\vec{i}$ has not been specified for any k between 1 to n. In this case, the set $S(\vec{i})$ is empty, and we do not define any rational blending functions. In the next section, we will see that the choice of the control points associated with these multi-indices does not affect the solution of the n-sided hole problem.

We introduce the following notation. Given any function $f(p)$, let $D_k^j(f(p))$ denote the jth derivative of the function $f(p)$ in the direction $\vec{N}_k$, which is normal to the edge E_k. In other words, $D_k^j(f(p))$ denotes the j-th cross-boundary derivative of the function $f(p)$ across the k-th boundary. Also, let $f(p)|_{E_k}$ denote the value of the function $f(p)$ evaluated along the edge E_k.

Lemma 4.1 The rational blending functions defined above satisfy the following properties:

1. $f_{\vec{i}}^{k}(p) \geq 0 \;\; \forall \;\; p \epsilon P$.
2. $\sum_{k \epsilon S(\vec{i})} f_{\vec{i}}^{k}(p) = 1$.
3. $f_{\vec{i}}^{k}(p)|_{E_k} = 1$.
4. $D_k^j(f_{\vec{i}}^{l}(p))|_{E_k} = 0$ for $1 \leq j \leq a_k - g_k(\vec{i})$, and $1 \leq l \leq n$.

Proof (1) and (2) follow immediately from the definition of the blending functions above.

(3) follows from the observation that $\hat{f}_{\vec{i}}^{j}|E_k = 0$ for $j \epsilon S(\vec{i}) - k$.

(4) follows from the observation that for $k \neq l$, $f_{\vec{i}}^{l}(p)$ contains the factor $h_{\vec{i}}^{k}$, that is, $u_k^{a_k+1-g_k(\vec{i})}$. Since, by assumption, $a_k - g_k(\vec{i}) \geq j$, $f_{\vec{i}}^{l}(p)$ contains at least the factor u_k^{j+1}. Therefore, by differentiating $f_{\vec{i}}^{l}(p)$ only j times, there still remains at least a linear factor of u_k, which vanishes along E_k.

For $k = l$, observe that $f_{\vec{i}}^{k}(p)$ is of the form $\frac{A(p)}{A(p)+u_k^{a_k+1-g_k(\vec{i})}B(p)}$, where $A(p)$ and $B(p)$ are functions of p. A simple calculation shows that the first cross-boundary derivative of $f_{\vec{i}}^{k}(p)$ contains a factor of $u_k^{a_k-g_k(\vec{i})}$. Therefore, by differentiating the blending function $f_{\vec{i}}^{k}(p)$ at most j times, where $j \leq a_k - g_k(\vec{i})$, there still remains at least a linear factor of u_k, which vanishes along E_k. $\square$

The first property states that the rational blending functions are positive inside the polygon. The second property states that they sum to unity. These two properties together guarantee that the rationally blended control points in equation 4.4 are formed by taking convex combinations of the user-specified control points. The third property states that the k-th blending function is 1 on the k-th boundary. The fourth property states that the j-th cross-boundary derivatives of the rational blending function $f_{\vec{i}}^{k}$ vanish along the k-th boundary, whenever $g_k(\vec{i}) \leq a_k - j$. The third and the fourth property together ensure that the hole-filling patch meets the surrounding patches smoothly across the boundary, as proved in the next section.

Lemma 4.2 The rationally blended control points $C_{\vec{i}}(p)$ in equation 4.4 satisfy the following properties:

1. $C_{\vec{i}}(p)|_{E_k} = C_{\vec{i}}^{k}$ for $g_k(\vec{i}) \leq a_k$.
2. $D_k^j(C_{\vec{i}}(p))|_{E_k} = 0$ for $1 \leq j \leq a_k - g_k(\vec{i})$.

Proof This is an immediate consequence of the lemma above and the definition of the rationally blended control points as given in equation 4.4. $\square$

Referring to the example, discussed in this section earlier, the rationally blended control point C_{211} is given by $C_{211} = \frac{u_2 u_3^2 C_{211}^1 + u_3^2 u_1^2 C_{211}^2 + u_1^2 u_2 C_{211}^3}{u_2 u_3^2 + u_3^2 u_1^2 + u_1^2 u_2}$, where the rational blending functions f_{211}^1, f_{211}^2, and f_{211}^3 are described in Equations 4.7 to 4.9. It is clear that these blending functions satisfy the properties (1) and (2) of Lemma 4.1.

Moreover, $f^1_{211}|_{E_1} = 1$, $f^2_{211}|_{E_2} = 1$, and $f^3_{211}|_{E_3} = 1$, so that they satisfy the third property of Lemma 4.1. This in turn implies the first statement of Lemma 4.2. For $k = 1$ and $k = 3$, the fourth property of Lemma 4.1 states that $D^1_1(f^l_{211})|_{E_1} = 0$ and $D^3_1(f^l_{211})|_{E_3} = 0$ for $l = 1, \cdots, 3$. This is easily verified. For $k = 2$, since $a_2 - g_2(211) = 0$, the fourth property of Lemma 4.1 is vacuously true. This in turn implies the second statement of Lemma 4.2.

4.8 General Solution

This section presents a solution to the problem of filling n-sided holes by a single rationally controlled S-patch using the rationally blended control points, defined in the previous section. Let a_k layers of control points, that is the control points $C^k_{\vec{i}}$ with the distance $g_k(\vec{i}) \leq a_k$, of an S-patch of depth d be prescribed across its kth boundary for $k = 1, \cdots, n$, where $a_k \geq 0$, and $d \geq max^n_{k=1} a_k$. We prove the theorem that there exists a rationally controlled S-patch H of depth d which solves the n-sided hole problem; that is, the first a_k cross-boundary derivatives of the patch H across the k-th boundary are the same as those given by the prescribed a_k layers of control points across that boundary.

Theorem 4.2 Let the hole-filling patch $H(p)$ be defined as follows:

$$H(p) = \sum_{\vec{i}} C_{\vec{i}}(p) B^d_{\vec{i}}(\gamma_1(p), \cdots, \gamma_n(p)) \tag{4.10}$$

$$C_{\vec{i}}(p) = \sum_{k \epsilon S(\vec{i})} f^k_{\vec{i}}(p) C^k_{\vec{i}} \tag{4.11}$$

In equation 4.10, the summation is to be taken over only those multi-indices $\vec{i}$ for which there exists at least one k such that $g_k(\vec{i}) \leq a_k$. In equation 4.11, the rational blending functions satisfy properties 1 to 4 stated in Lemma 4.1. Then, for $k = 1, \cdots, n$ and for $j = 0, \cdots, a_k$,

$$D^j_k(H(p))|_{E_k} = \sum_{g_k(\vec{i}) \leq j} C^k_{\vec{i}} D^j_k(B^d_{\vec{i}}(\gamma_1(p), \cdots, \gamma_n(p)))|_{E_k} \tag{4.12}$$

Proof The proof depends upon the following lemma:

Lemma 4.3 The j-th cross-boundary derivative of the $\vec{i}$-th Bernstein polynomial evaluated at $(\gamma_1(p), \cdots, \gamma_n(p))$ is 0 if the distance of the $\vec{i}$-th

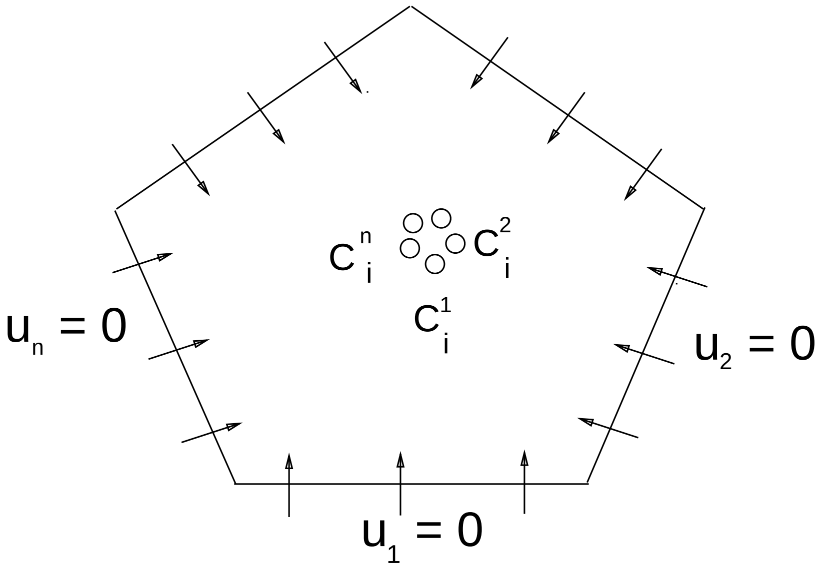

Figure 4.6 The n-sided hole problem: general case

control point from that boundary is greater than j. In other words,

$$D_k^j B_{\vec{i}}^d(\gamma_1(p), \cdots, \gamma_n(p))|_{E_k} = 0 \quad for \quad g_k(\vec{i}) > j. \tag{4.13}$$

Proof By the definition of $\gamma_i(p)$ as given in equations 4.2 and 4.3, every $\gamma_i(p)$ except $\gamma_k(p)$ and $\gamma_{k+1}(p)$ contains a factor of $u_k(p)$. Therefore, it follows from the definition of $B_{\vec{i}}^d(\gamma_1(p), \cdots, \gamma_n(p))$ that this Bernstein polynomial contains a factor of $u_k^{g_k(\vec{i})}$. After differentiating $B_{\vec{i}}^d$ j times, there still remains at least one factor of u_k common to every term, because $g_k(\vec{i}) > j$. However, $u_k = 0$ along E_k; hence the result follows. □

The proof of Theorem 4.2 is as follows:

$$\begin{aligned}
D_k^j(H(p))|_{E_k} &= D_k^j(\sum_{\vec{i}} C_{\vec{i}}(p) B_{\vec{i}}^d(\gamma_1(p), \cdots, \gamma_n(p)))|_{E_k} \\
&= D_k^j(\sum_{g_k(\vec{i}) \le j} C_{\vec{i}}(p) B_{\vec{i}}^d(\gamma_1(p), \cdots, \gamma_n(p)))|_{E_k} \quad by \ \ Lemma \ \ 4.3 \\
&= \sum_{g_k(\vec{i}) \le j} \sum_{l=0}^{j} \begin{pmatrix} j \\ l \end{pmatrix} D_k^l(C_{\vec{i}}(p))|_{E_k} D_k^{j-l}(B_{\vec{i}}^d(p))|_{E_k}
\end{aligned}$$

For $1 \leq l \leq a_k - g_k(\vec{i})$, it follows from Lemma 4.2 that the first term $D_k^l(C_{\vec{i}}(p))|_{E_k}$ on the right hand side vanishes. On the other hand, for $a_k - g_k(\vec{i}) < l \leq j$, which implies that, $g_k(\vec{i}) > a_k - l \geq j - l$, it follows from Lemma 4.3 that the second term $D_k^{j-l}(B_{\vec{i}}^d(p))|_{E_k}$ on the right hand side vanishes. Therefore, the only contribution on the right hand side comes from the terms when $l = 0$. However, for $l = 0$, the first term on the right reduces to $C_{\vec{i}}^k$ by Lemma 4.2. Hence the following desired result is obtained:

$$D_k^j(H(p))|_{E_k} = \sum_{g_k(\vec{i}) \leq j} C_{\vec{i}}^k D_k^j(B_{\vec{i}}^d(\gamma_1(p), \cdots, \gamma_n(p)))|_{E_k} \quad for \quad j = 0, \cdots, a_k. \tag{4.14}$$

This concludes the proof of the theorem. □

Remarks:

- We have not assumed any compatibility conditions whatsoever on the position or derivative information of the surrounding patches. However, it is inherent in the n-sided hole filling problem that the surrounding patches meet at corners. This ensures that there are no cracks between the surrounding patches. These corner compatibility condition can be expressed as

$$C_{n\vec{e_k}}^{k-1} = C_{n\vec{e_k}}^{k} \quad for \quad k = 1, \cdots, n$$

  Also, in most practical situations, one assumes the derivative compatibility conditions along the boundary of the curve. In other words, the boundary control points are the same irrespective of the side from which they have been prescribed, that is,

$$C_{\vec{i}} = C_{\vec{i}}^k \ \ if \ \ g_k(\vec{i}) = 0 \ \ for \ \ any \ \ k = 1, \cdots, n$$

  However, we certainly do *not* assume any twist compatibility conditions, such as $C_{\vec{i}}^{k-1} = C_{\vec{i}}^k$ if $g_{k-1}(\vec{i}) * g_k(\vec{i}) \neq 0$.

- By Lemma 4.3, the control points $C_{\vec{i}}^k$, where $g_k(\vec{i}) > a_k$ for every k, do not affect the desired C^{a_k} continuity with the surrounding patches. Therefore, these control points can be chosen completely arbitrarily. Moreover, the weights $z_{\vec{i}}^k$ appearing in expression 4.6 provide additional flexibility in constructing these patches.

- The proposed solution solves the n-sided hole filling problem, for any number of prescribed cross-boundary derivatives when the domain is a regular convex n-gon in R^2. Using generalized barycentric coordinates [War92], the solution is easily generalizable for any number of prescribed cross-boundary derivatives when the domain is a convex polytope in arbitrary dimension.

4.9 Examples

This section presents several applications of Theorem 4.2 for the case of 2-, 3-, 4- , and 5-sided hole filling problems.

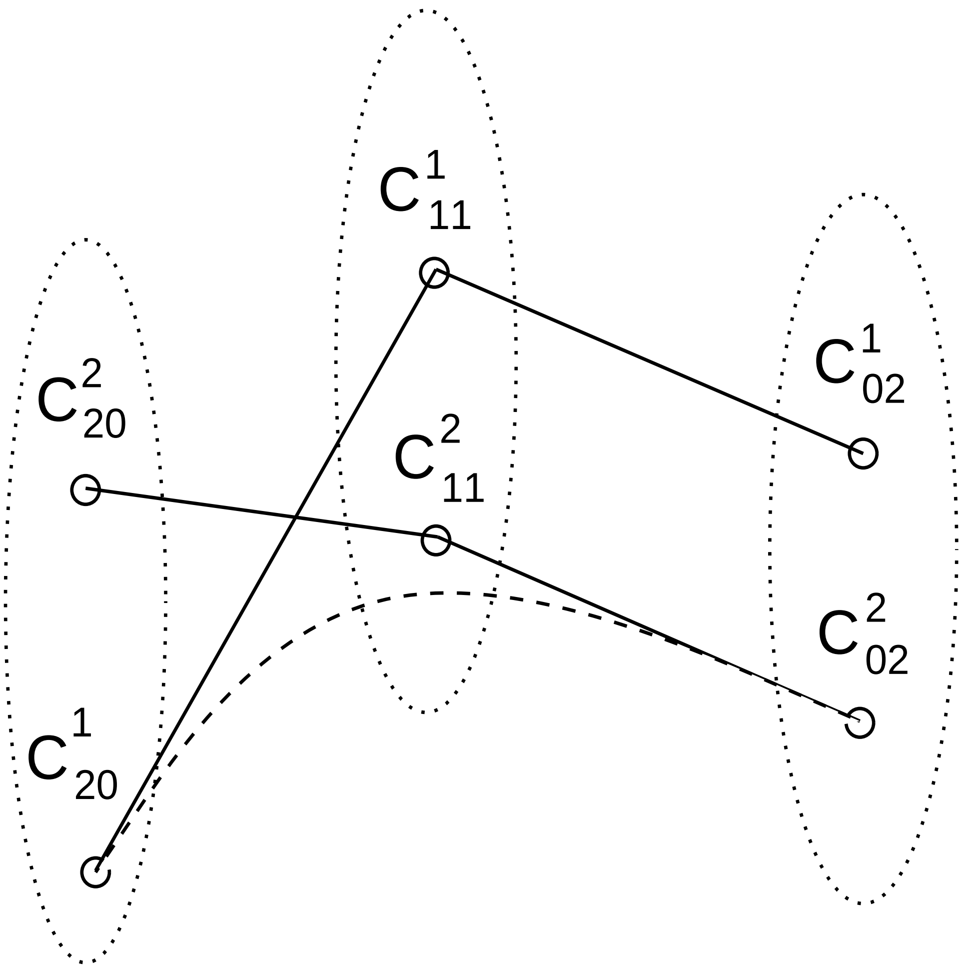

Figure 4.7 The 2-sided hole problem: C^2 quadratic case

1. 2-sided Hole Problem:

   (a) General Case:
   An explicit solution for the 2-sided hole problem, described in Theorem 4.1, can be given as follows:

   $$\begin{aligned} H(t) &= \sum_{\vec{i}} C_{\vec{i}}(t) B^d_{\vec{i}}(t, 1-t) \\ C_{\vec{i}}(t) &= \frac{(1-t)^{a_2+1-g_2(\vec{i})} C^1_{\vec{i}} + t^{a_1+1-g_1(\vec{i})} C^2_{\vec{i}}}{(1-t)^{a_2+1-g_2(\vec{i})} + t^{a_1+1-g_1(\vec{i})}} \end{aligned}$$

   The particular case of a solution with C^2 continuity from both ends by a rationally controlled quadratic Bézier curve is illustrated in Figure 4.7. In this case $a_1 = a_2 = 2$ and $g_1(20) = 0$, $g_1(11) = 1$, $g_1(02) = 2$, $g_2(20) = 2$, $g_2(11) = 1$, and $g_2(02) = 0$. Therefore, an explicit solution is given by

   $$H(t) = \sum_{|\vec{i}|=2} C_{\vec{i}} B^2_{\vec{i}}$$

   where $C_{20} = \frac{(1-t)C^1_{20} + t^3 C^2_{20}}{(1-t)+t^3}$, $C_{11} = \frac{(1-t)^2 C^1_{11} + t^2 C^2_{11}}{(1-t)^2+t^2}$, and $C_{02} = \frac{(1-t)^3 C^1_{02} + t C^2_{02}}{(1-t)+t^3}$.

   (b) Particular Case:
   The objective is to construct a rationally controlled Bézier curve of degree 3, which meets the surrounding curves with C^2 continuity. An explicit solution is given by

   $$H(t) = \sum_{|\vec{i}|=3} C_{\vec{i}} B^3_{\vec{i}}$$

   where $C_{30} = C^1_{30}$, $C_{21} = \frac{(1-t)C^1_{21} + t^2 C^2_{21}}{(1-t)+t^2}$, $C_{12} = \frac{(1-t)^2 C^1_{12} + t C^2_{12}}{(1-t)^2+t}$, and $C_{03} = C^2_{03}$. Note that in the above solution, since the denominators of the expressions for C_{21} and C_{12} are different, the solution after expansion is a rational curve given by a ratio of polynomial of degree 7 to a polynomial of degree 4.

   An alternative representation can be obtained by choosing the weights $z^1_{21} = 1-t$, $z^2_{21} = 1$, $z^1_{12} = 1$, and $z^2_{12} = t$. The solution is now described as follows: $C_{30} = C^1_{30}$, $C_{21} = \frac{(1-t)^2 C^1_{21} + t^2 C^2_{21}}{(1-t)^2+t^2}$, $C_{12} = \frac{(1-t)^2 C^1_{12} + t^2 C^2_{12}}{(1-t)^2+t^2}$, and $C_{03} = C^2_{03}$. Since the denominators for the expressions C_{21} and C_{12} are the same, this solution has the advantage that it represents a rational curve given by a ratio of polynomial of degree 5 to a polynomial of degree 2. We shall refer to this technique of choosing weights to make the denominator common as the common denominator technique.

2. 3-sided Hole Problem:

   (a) C^1 Cubic Case (Nielson-Foley Patch):
   The objective is to construct a rationally controlled cubic triangular Bézier surface patch which meets with the surrounding patches with C^1 continuity. The nine boundary control points are prescribed to be compatible. The only remaining interior control point is prescribed from each of the three sides, as shown in Figure 4.8. Since $a_k = 1$ and $g_k(111) = 1$ for $k = 1, \cdots, 3$, an explicit solution by Theorem 4.2 is given as follows: $C_{111} = \frac{u_2 u_3 C^1_{111} + u_3 u_1 C^2_{111} + u_1 u_2 C^3_{111}}{u_2 u_3 + u_3 u_1 + u_1 u_2}$, which agrees with the solution proposed by Nielson [Nie87] and Foley [Fol91].

   (b) C^2 Quartic Case:
   The objective is to construct a rationally controlled quartic triangular Bézier surface patch, which meets with the surrounding patches with C^2 continuity. The twelve boundary control points are prescribed to be compatible. The remaining three interior control points are prescribed from

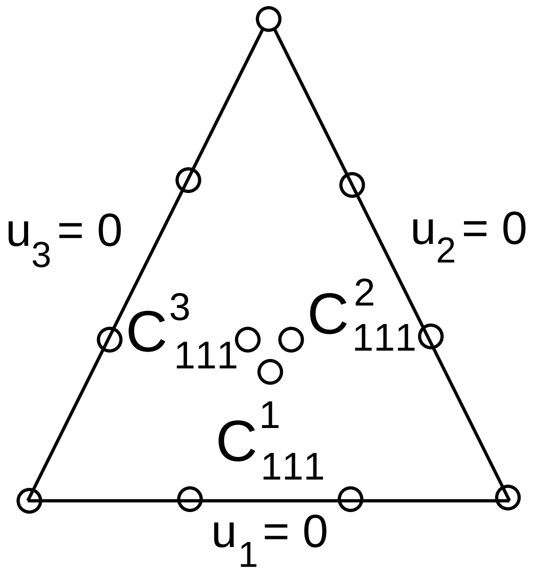

Figure 4.8 The 3-sided hole problem: Nielson-Foley patch

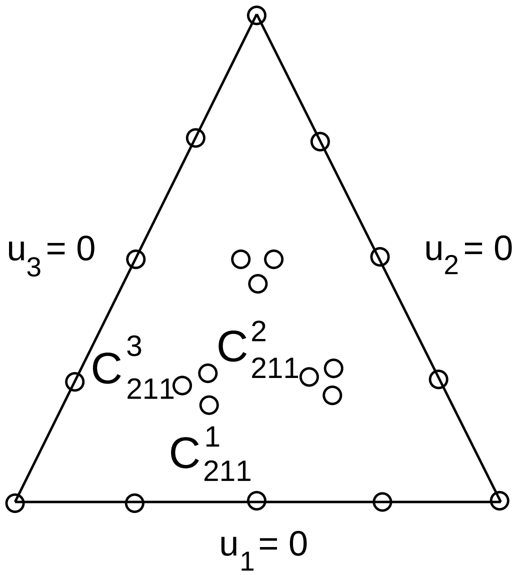

Figure 4.9 The 3-sided hole problem: C^2 quartic Case

each of the three sides, as shown in Figure 4.9. Following Theorem 4.2, an explicit solution is given by setting: $C_{211} = \frac{u_2 u_3^2 C_{211}^1 + u_3^2 u_1^2 C_{211}^2 + u_1^2 u_2 C_{211}^3}{u_2 u_3^2 + u_3^2 u_1^2 + u_1^2 u_2}$. The expressions for the other two interior control points can be derived by symmetry. An alternative new solution using the common denominator technique is given by setting $C_{211} = \frac{u_2^2 u_3^2 C_{211}^1 + u_3^2 u_1^2 C_{211}^2 + u_1^2 u_2^2 C_{211}^3}{u_2^2 u_3^2 + u_3^2 u_1^2 + u_1^2 u_2^2}$. In this case, the weight functions have been chosen as follows: $z_{211}^1 = u_2$, $z_{211}^2 = 1$, and $z_{211}^3 = u_2$.

3. 4-sided Hole Problem: A 4-sided hole is filled with a rationally controlled 4-sided S-patch. Since 4-sided S-patches are generalizations of tensor product Bézier surface patches, we will use the notation for rationally controlled tensor product Bézier surface patches to describe the solution to the 4-sided hole problem. A rational tensor product Bézier surface patch of bidegree (s, t) is represented as follows:
$$S(p) = \frac{\sum_{i=0}^{i=s} \sum_{j=0}^{j=t} w_{ij} C_{ij} B_i^s(u_1) B_j^t(u_4)}{\sum_{i=0}^{i=s} \sum_{j=0}^{j=t} w_{ij} B_i^s(u_1) B_j^t(u_4)}$$
where C_{ij} are control points, w_{ij} are weights, $B_i^s(u_1) = \binom{s}{i} u_1^i (1-u_1)^{s-i}$ are the univariate Bernstein polynomials, and (u_1, u_4) are the coordinates of the point p inside the domain rectangle $0 \leq u_1, u_4 \leq 1$. A tensor product Bézier

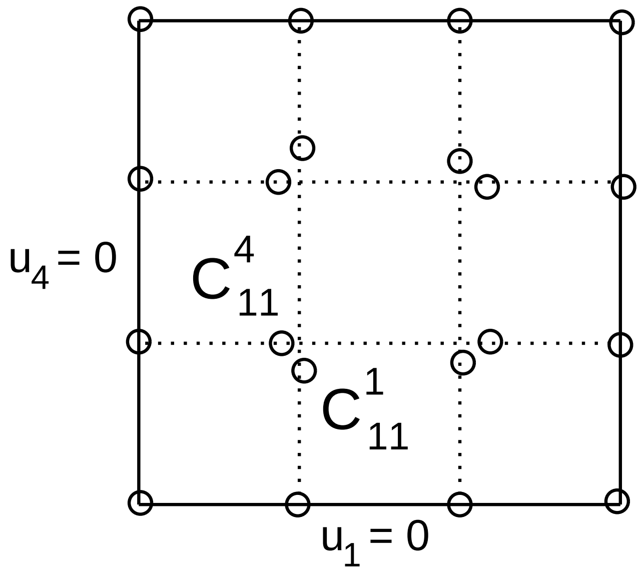

Figure 4.10 The 4-sided hole problem: Gregory patch

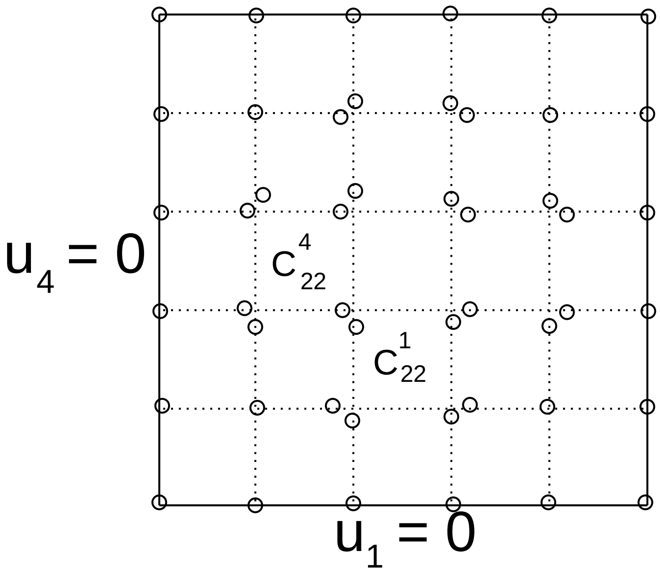

Figure 4.11 The 4-sided hole problem: Barnhill-Worsey patch

surface patch has a rationally controlled representation of bidegree (s, t), when the coefficients C_{ij} are rational functions of p instead of constants.

(a) C^1 Cubic Case (Gregory Patch:) The objective is to construct a rationally controlled bicubic tensor product Bézier surface patch which meets the surrounding patches with C^1 continuity. The twelve boundary control points are prescribed to be compatible. Each of the remaining four interior control points are prescribed from only two sides as shown in Figure 4.9. Following Theorem 4.2, an explicit solution is given by setting: $C_{11} = \frac{u_4 C_{11}^1 + u_1 C_{11}^4}{u_4 + u_1}$, which agrees with the solution described by Chiyokura and Kimura [CK83] for the Gregory patch [Gre83a]. The expressions for the remaining three interior control points can be derived by symmetry.

(b) C^1 Cubic Case (Brown's Square:) An alternative solution for the same problem can be obtained using the common denominator technique. This gives the solution: $C_{11} = \frac{u_4^2 u_2^2 C_{11}^1 + u_3^2 u_1^2 C_{11}^4}{u_4^2 u_2^2 + u_3^2 u_1^2}$, which is known as Brown's square [Gre83a]. Note that in this case, all the interior control points have the same common denominator.

(c) C^2 Quintic Case (Barnhill-Worsey Patch:) The objective is to construct a rationally controlled biquintic tensor product Bézier surface patch, which meets with the surrounding patches with C^2 continuity. The twenty boundary control points are prescribed to be compatible. Moreover the first order twists are prescribed to be compatible, so that the four control points diagonal to the corner control points are also also uniquely determined by the user-specified control points. The remaining eight control points on the first layer can be derived using Theorem 4.2, which agrees with the solution proposed by Barnhill [Bar83]. Each of the remaining four interior control points are prescribed from only two sides as shown in Figure 4.11. Following Theorem 4.2, an explicit solution is given by setting: $C_{22} = \frac{u_4 C_{22}^1 + u_1 C_{22}^4}{u_4 + u_1}$, which agrees with the solution described by Worsey [Wor84] and is an improvement over the original solution proposed by Barnhill [Bar83]. The expressions for the remaining three interior control points can be derived by symmetry.

4. 5-sided Hole Problem: The objective is to construct a rationally controlled 5-sided S-patch of depth 3, which meets the surrounding patches with C^1 continuity. All the boundary control points are prescribed to be compatible. Moreover, the tangent plane compatibility from the two sides at each corner forces the remaining control points to be uniquely determined except for

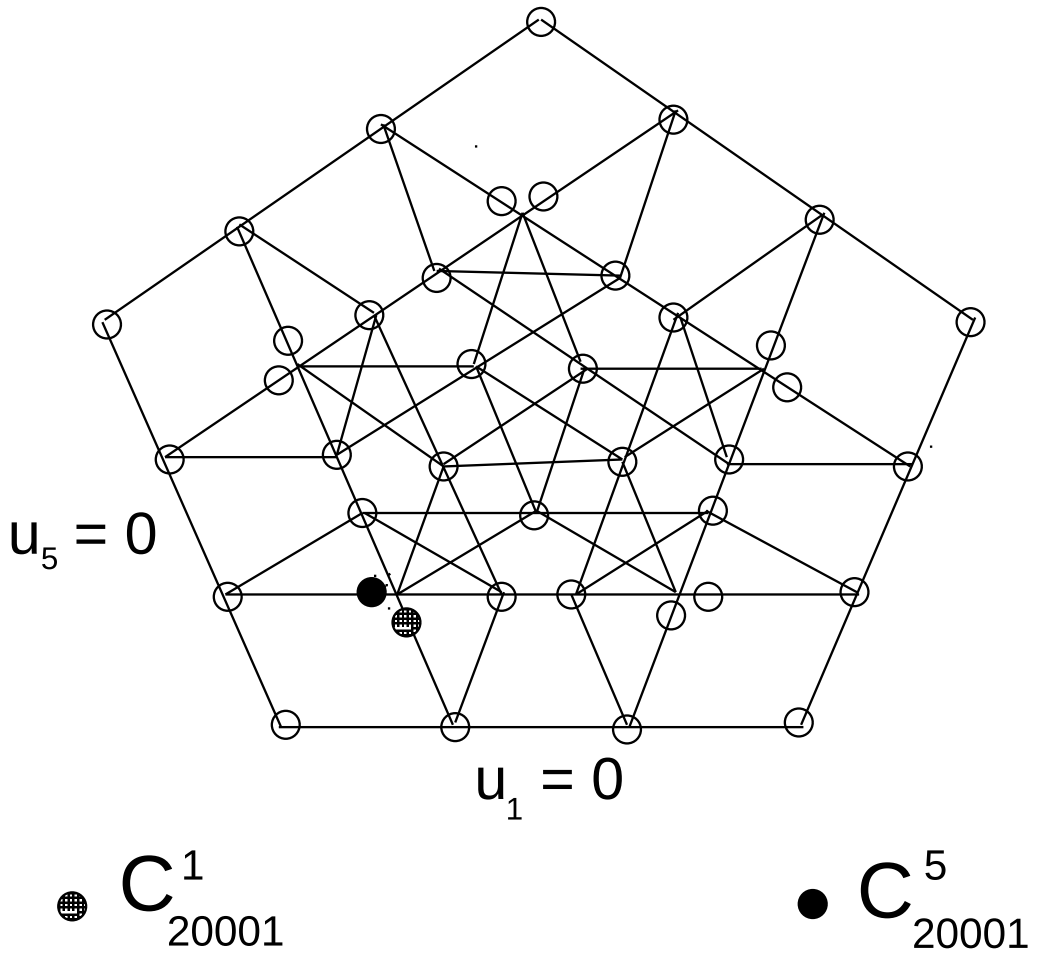

Figure 4.12 The 5-sided hole problem

the five control points C_{20001}, C_{12000}, C_{01200}, C_{00120}, and C_{00012} as shown in Figure 4.12. Following Theorem 4.2, an explicit solution is given by setting: $C_{20001} = \frac{u_5 C^1_{20001} + u_1 C^5_{20001}}{u_5 + u_1}$. The expressions for the remaining four interior control points can be derived by symmetry.

This paper described a novel mathematical technique for filling an n-sided hole by a single rationally controlled S-patch. This compact representation is constructed by taking the convex combination of user-specified control points using a judicious choice of rational blending functions.

Chapter 5

Degree Reduction of Bézier Simplexes

The previous chapter described a technique to fill an n-sided hole smoothly by a single rational S-patch. This S-patch can be represented as a collection of triangular Bézier surface patches using multiprojective blossoming algorithm [LD89, DeR88, DGHM92]. After filling the holes, we obtain a smooth surface consisting of high degree parametric Bézier surfaces, which does not contain any holes. This chapter describes a method of approximating high degree parametric Bézier patches by low degree Bézier patches. The technique is applicable in any dimension. Therefore, this chapter describes a general degree reduction algorithm for Bézier simplexes. The algorithm has the following properties: (1) Symmetry: The degree reduction method is symmetric with respect to the corner points of the Bézier simplex. (2) Restriction: The degree reduction method restricted to the boundary of a Bézier simplex yields the same result as the boundary of the degree-reduced Bézier simplex. (3) Interpolation: The degree-reduced simplex of degree e interpolates the value and the first $\lfloor \frac{e-1}{2} \rfloor$ derivatives at the corner points of the original Bézier simplex. (4) Optimal order of approximation: The order of approximation of the given simplex by the degree-reduced simplex is $O(h^{e+1})$, where h is the diameter of the domain simplex, which is optimal for functional approximation. The method, restricted to Bézier surfaces, describes a new technique for degree reduction which is easy to implement.

5.1 Conversion Between Bézier and Multinomial Basis

The degree reduction algorithm depends crucially upon an efficient algorithm for conversion between Bézier and multinomial basis. To this purpose, we first introduce the multinomial basis and then describe the basis conversion algorithm.

In the previous chapter we reviewed the definition of a Bézier simplex, and described the multi-index notation. Recall that a multi-index is a tuple of non-negative integers, for instance, $\vec{i} = (i_0, \cdots, i_n)$. This multi-index has *length* $n+1$. A multi-index $\vec{j}$ is defined to be less than or equal to another multi-index $\vec{i}$, denoted by $\vec{j} \leq \vec{i}$, if they are of the same length, and $j_k \leq i_k$ for $k = 0, \cdots, n$. A multi-index $\frac{\vec{i}}{d}$ is obtained by dividing each component of $\vec{i}$ by d. Recall that a Bézier simplex is a polynomial map B of degree d from an affine space A_1 of dimension n to an affine space A_2 of arbitrary dimension s, and is represented in the Bernstein-Bézier basis with respect to a domain simplex Δ in A_1 as follows:

$$B(v) = \sum_{\vec{i}} C_{\vec{i}} B^d_{\vec{i}}(\alpha_0(v), \cdots, \alpha_n(v)) \tag{5.1}$$

where $\alpha_0, \cdots, \alpha_n$ are the barycentric coordinates of a point v relative to the domain simplex Δ. Also, recall that

$$B^d_{\vec{i}}(\alpha_0, \ldots, \alpha_n) = \frac{d!}{i_0! \cdots i_n!} \alpha_0^{i_0} \cdots \alpha_n^{i_n}. \tag{5.2}$$

Let $v_0, \cdots, v_n$ be the vertices of the domain simplex Δ. We now define the n-variate *multinomial* polynomials of degree up to d around the vertex v_k as follows:

$$M^d_{\vec{i},k}(\alpha_0, \ldots, \alpha_{k-1}, \alpha_{k+1}, \ldots, \alpha_n) = \frac{\alpha_0^{i_0} \cdots \alpha_{k-1}^{i_{k-1}} \alpha_{k+1}^{i_{k+1}} \cdots \alpha_n^{i_n}}{i_0! \cdots i_{k-1}! i_{k+1}! \cdots i_n!}$$

where $|\vec{i}| = d$. To simplify the notation, we shall denote the n-variate multinomial polynomials $M^d_{\vec{i},0}$ of degree d around the vertex v_0 as $M^d_{\vec{i}}$ by omitting the subscript k. Note that if the domain simplex is chosen to be the standard simplex with vertices located at a unit distance from the origin in the positive direction along the coordinate axes, then these multinomial polynomials around the origin are none other than the polynomials occurring in the Taylor series expansion of the polynomial map. For example, for $n = 2$ and $d = 3$, the multinomial polynomials around the origin for the standard simplex are simply $1, x, y, \frac{x^2}{2}, xy, \frac{y^2}{2}, \frac{x^3}{6}, \frac{x^2y}{2}, \frac{xy^2}{2}$, and $\frac{y^3}{6}$. Any polynomial map, in particular any Bézier simplex, can also be written in multinomial basis as follows:

$$B(v) = \sum_{\vec{i}} \frac{d!}{i_k!} T_{\vec{i},k} M^d_{\vec{i},k}(\alpha_0, \ldots, \alpha_{k-1}, \alpha_{k+1}, \ldots, \alpha_n) \tag{5.3}$$

This representation is unique as well, and the coefficients $T_{\vec{i},k}$ will be referred to as *multinomial* coefficients with respect to the vertex v_k. To simplify the notation, we shall denote the multinomial coefficients $T_{\vec{i},0}$ with respect to the vertex v_0 also as $T_{\vec{i}}$.

It is easy to derive a relationship between the multinomial coefficients and the Bézier control points by substituting $\alpha_k = 1 - \alpha_0 - \cdots - \alpha_{k-1} - \alpha_{k+1} - \cdots - \alpha_n$ in Equation 5.1 and comparing with Equation 5.3. The computation, although notationally complex, is straightforward. To describe the relationship, let $\vec{i'} = (i_1, \cdots, i_n)$

be the n-tuple obtained from a $(n+1)$-tuple $\vec{i} = (i_0, i_1, \cdots, i_n)$ by deleting the 0-th element. Moreover, given an n-tuple $\vec{j'} = (j_1, \cdots, j_n)$, let $\vec{j} = (d - |\vec{j'}|, j_1, \cdots, j_n)$ be the $(n+1)$-tuple. Then we have the following relationship:

$$T_{\vec{i}} = \sum_{\vec{j'} \leq \vec{i'}} (-1)^{|\vec{i'}| - |\vec{j'}|} C_{\vec{j}} \binom{i_1}{j_1} \cdots \binom{i_n}{j_n} \tag{5.4}$$

For example, for $d = 2$ and $n = 1$, Equation 5.4 translates into the following statements: $T_{20} = C_{20}$, $T_{11} = -C_{20} + C_{11}$, and $T_{02} = C_{20} - 2C_{11} + C_{02}$.

We now describe an efficient algorithm to convert a polynomial map given in Bézier basis to multinomial basis and vice-versa. To simplify the notation, we consider the multinomial basis with respect to the vertex v_0; the algorithm is similar for a multinomial basis with respect to any other vertex v_k. This algorithm will be used later in the degree reduction procedure. Although one can use Equation 5.4 to compute the multinomial coefficients from the Bézier control points explicitly, this would be an inefficient approach. The key to an efficient algorithm is to observe that the summands on the right hand side of Equation 5.4 can be rearranged to organize the computation as follows:

$$T_{\vec{i}} = \sum_{j_n=0}^{i_n} (-1)^{i_n - j_n} (\cdots (\sum_{j_1=0}^{i_1} (-1)^{i_1 - j_1} C_{\vec{j}} \binom{i_1}{j_1}) \cdots) \binom{i_n}{j_n} \tag{5.5}$$

where $\vec{j} = (d - |\vec{j'}|, j_1, \cdots, j_n)$ and $\vec{j'} = (j_1, \cdots, j_n)$. Referring to the example of $d = 2$ and $n = 1$, Equation 5.5 organizes the computation as follows: $T_{20} = C_{20}$, $T_{11} = -C_{20} + C_{11}$, and $T_{02} = (C_{20} - C_{11}) - (C_{11} - C_{02})$.

The algorithm for converting from a Bézier basis to a multinomial basis is presented in Figure 5.1. We now explain how Equation 5.5 can be utilized to construct this algorithm for converting between a Bézier basis and a multinomial basis. The idea is to carry out the total computation in n steps. In each of these n steps, there are d layers of computation, which is simplified by a recurrence relation. To describe the recurrence relation, let $C^k_{\vec{i}}$ denote the control points obtained after k layers of computation. Thus the superscript k denotes the layer. The algorithm starts by setting the control points of 0-th layer $C^0_{\vec{i}}$ as the given control points $C_{\vec{i}}$. In the first step, the recurrence relation is given by

$$C^k_{\vec{i}} = C^{k-1}_{\vec{i}+\vec{e_0}} - C^{k-1}_{\vec{i}+\vec{e_1}} \quad for \quad k = 1, \cdots, d$$

where $\vec{e_l}$ denotes the unit vector in the l-direction, that is, $\vec{e_l} = (0, \cdots, 1, \cdots, 0)$, where 1 appears in the l-th position. We will refer to this computation as a computation

Input: Bézier control points $C_{\vec{i}}$.
Output: Multinomial coefficients $T_{\vec{i}}$ with respect to the vertex v_0.
Algorithm:
$C_{\vec{i}}^{n,0} = C_{\vec{i}}^{n}$ /* Label the given control points as control points of layer 0 */
for l = 1 to n do /* For each vertex v_1 to v_n do */
{
/* Recurrence Relation */
  for k = 1 to d do /* For layers 1 to d do */
    for $\vec{i} = (i_0, \ldots, i_n)$ with $|\vec{i}| = d - k$ do
      $C_{\vec{i}}^{k} = C_{\vec{i}+\vec{e_0}}^{k-1} - C_{\vec{i}+\vec{e_l}}^{k-1}$
/* Relabel the coefficients on the simplicial face opposite the vertex v_l */
  for k = 1 to d do
    for $\vec{i} = (i_0, \ldots, i_n)$ with $i_l = 0$ do
      $C_{\vec{i}+k\vec{e_l}}^{0} = C_{\vec{i}}^{k}$
} /* n steps of computation in directions v_0v_1 to v_0v_n */
$T_{\vec{i}} = C_{\vec{i}}^{0}$.

Figure 5.1 Algorithm for converting from Bézier basis to multinomial basis

in the direction of the edge v_0v_1. This recurrence relation is an efficient way to execute the computation $\sum_{j_1=0}^{i_1}(-1)^{i_1-j_1} C_{\vec{j}} \binom{i_1}{j_1}$, described in the innermost bracket in Equation 5.5. Figure 5.2 illustrates the first step of the algorithm for converting from a Bézier basis to a multinomial basis for the case of a cubic Bézier surface. The new coefficients $C_{\vec{i}}^{k}$ for layers 1 to 3 are obtained by taking the difference of the appropriate coefficients of the previous layer in the direction of the edge v_0v_1. The process is illustrated schematically in Figure 5.2 by building a higher dimensional simplex, in this case a tetrahedron, with the original control points at the base of the tetrahedron. The coefficients on the face of this tetrahedron opposite to the vertex v_1 have the property that $i_1 = 0$. These coefficients are now relabeled by the equation

$$C_{\vec{i}+k\vec{e_1}}^{0} = C_{\vec{i}}^{k}.$$

These control points now form the starting point for the next iteration. In Figure 5.3, the coefficients of Figure 5.2 on the face opposite to the vertex v_1 are shown with the new labels. The process is now repeated in the direction v_0v_2. Specifically, new coefficients are computed from these coefficients for layers 1 to 3 by taking the

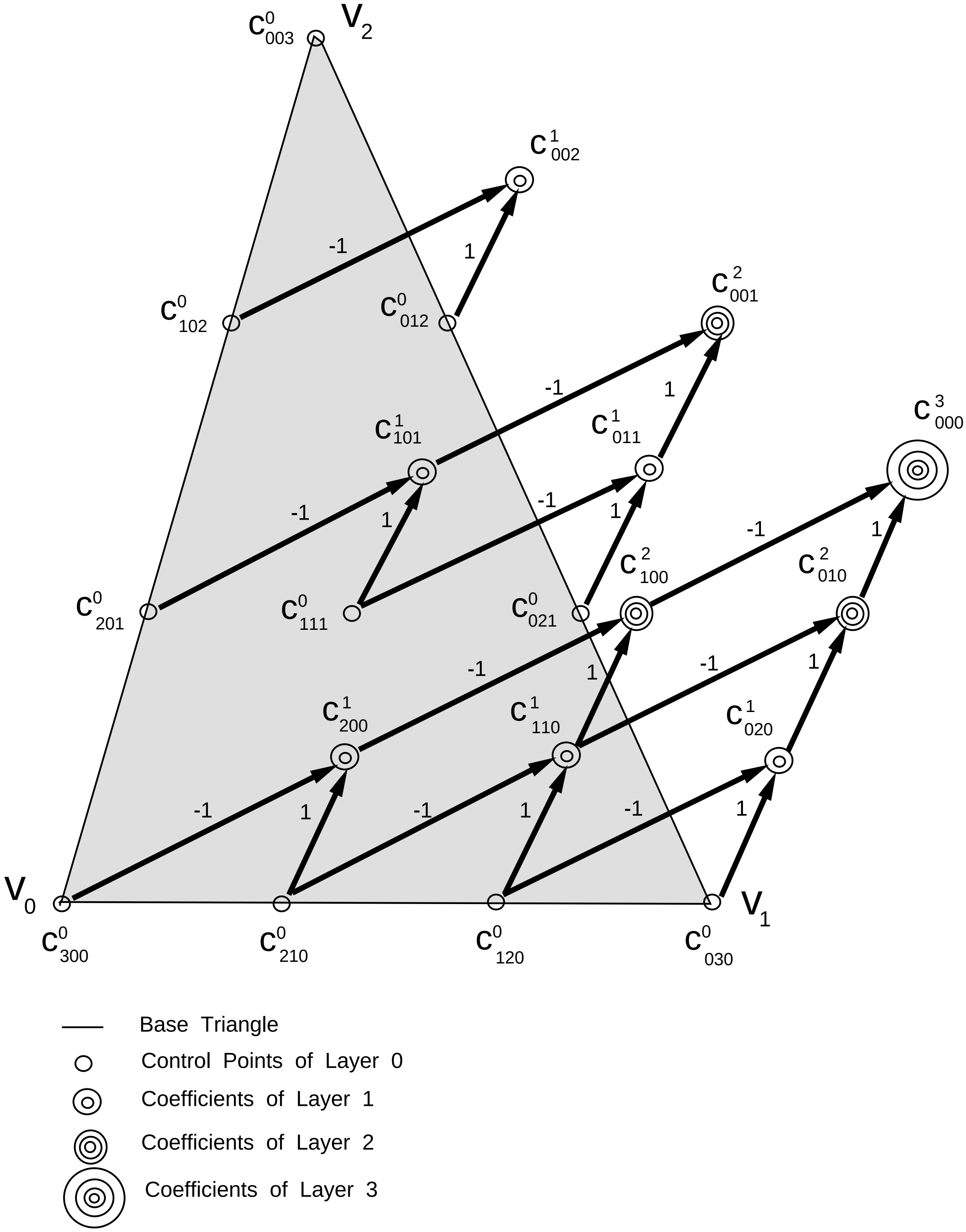

Figure 5.2 Conversion from Bézier basis to multinomial Basis

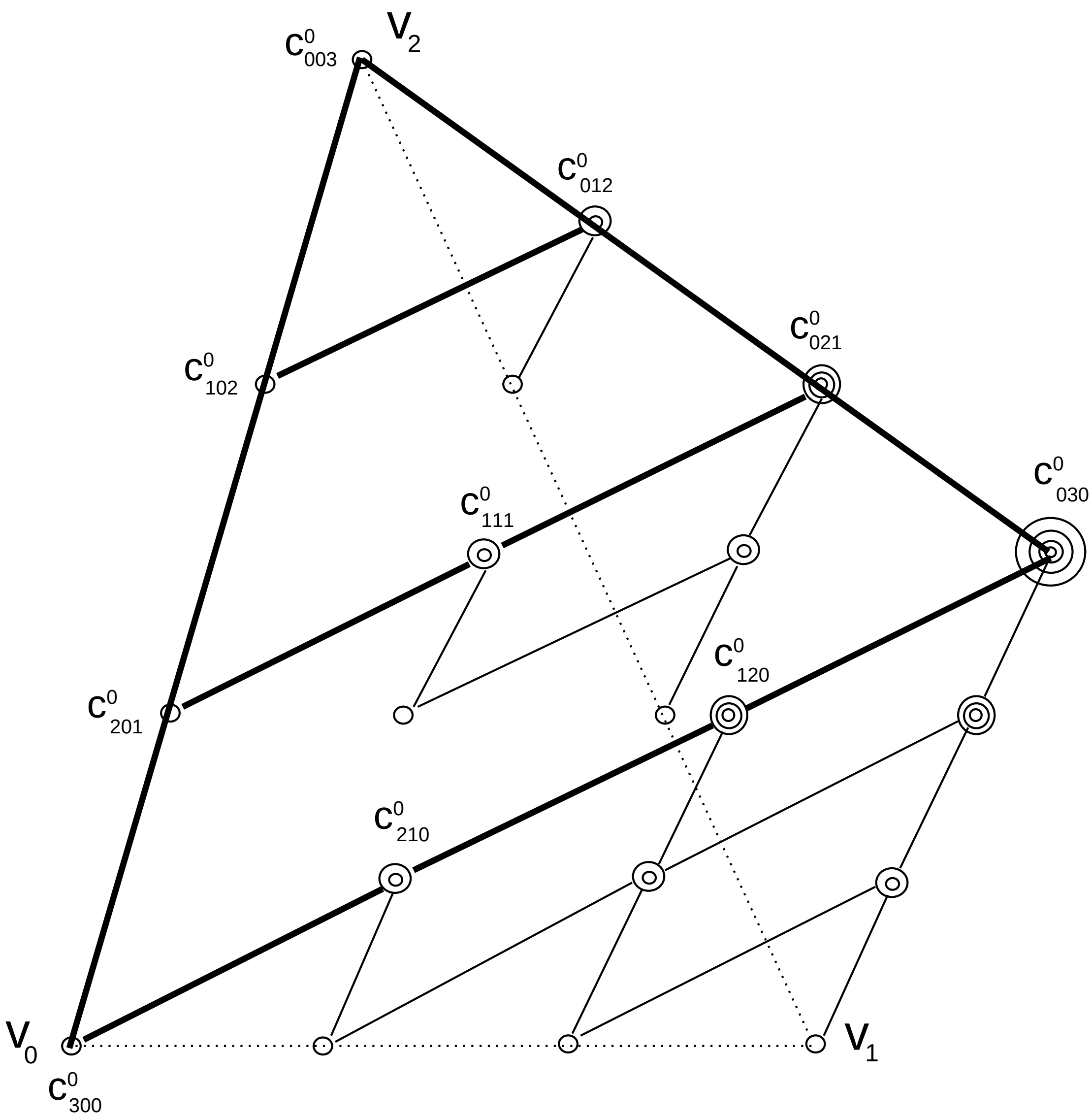

Figure 5.3 Relabeling of coefficients on the face opposite to the vertex v_1

difference of the coefficients of the previous layer in the direction v_0v_2. The coefficients on the face of the new tetrahedron opposite to the vertex v_2 now yield the desired multinomial coefficients. For curves the analogous algorithm using the recurrence relation to convert from the Bézier basis to the monomial basis has been derived using blossoming techniques [BG91].

The conversion from the multinomial basis to the Bézier basis is obtained by a similar algorithm, where in each step new coefficients are computed by taking the *sum* of the coefficients of the previous layer instead of the difference.

5.2 Weight Function

The other major component in the degree reduction algorithm is the judicious choice of a weight function. We now describe the notation for a weight function, which will be used later in our discussion. Any function $Y_{\vec{i},k}$ of k and $\vec{i}$ satisfying the following two properties will be referred to as a weight function:

1. Positivity: $Y_{\vec{i},k} \geq 0 \quad for \mid \vec{i} \mid = d \quad and \quad for \quad k = 1, 2, \ldots, n$
2. Partition of unity: $\sum_{k=1}^{n} Y_{\vec{i},k} = 1 \quad for \quad \mid \vec{i} \mid = d$

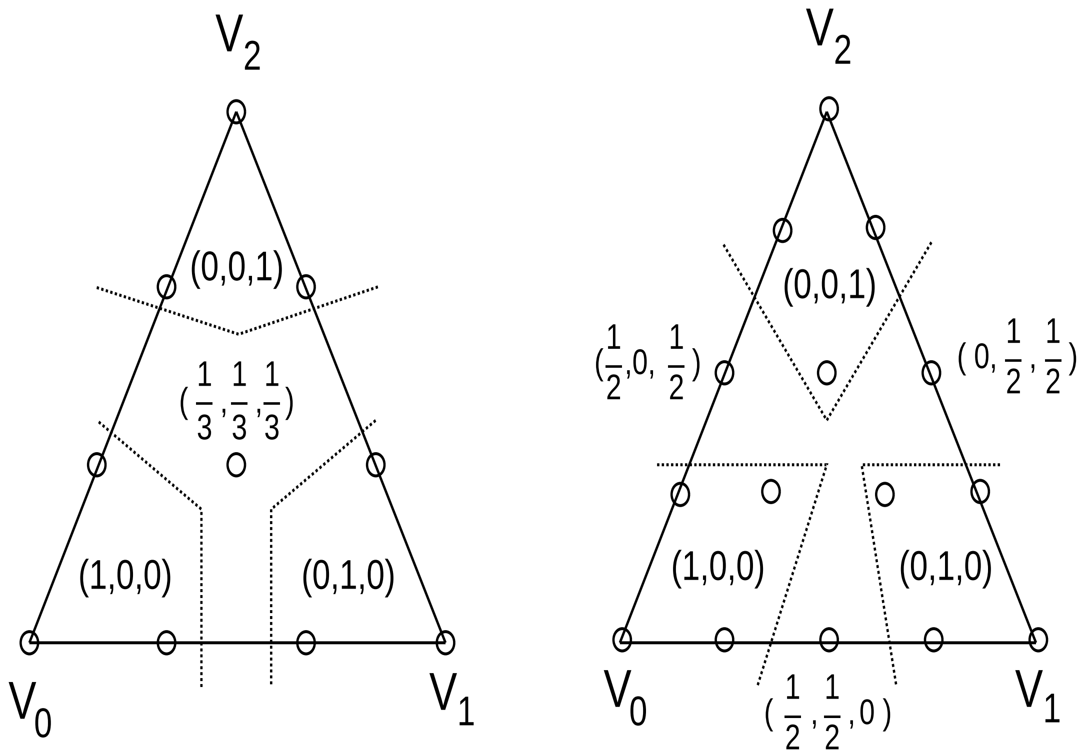

Figure 5.4 Weight function for cubic and quartic Bézier surfaces

We now define a particular weight function $W_{\vec{i},k}$. Informally, $W_{\vec{i},k}$ is equal to 1 if the vertex v_k is closest to the point with barycentric coordinates $\frac{\vec{i}}{d}$, equal to $\frac{1}{l}$ if v_k is one of l vertices closest to the point with barycentric coordinates $\frac{\vec{i}}{d}$, and equal to 0 otherwise. Formally, given $\vec{i} = (i_0, i_1, \ldots, i_n)$, let

$$S(\vec{i}) = \{k | i_k = max\{i_0, i_1, \cdots, i_n\}\}$$

Let $\mid S(\vec{i}) \mid$ denote the cardinality of $S(\vec{i})$. Let $\xi_k(S(\vec{i}))$ be the following characteristic function:

$$\begin{aligned} \xi_k(S(\vec{i})) &= 1 \quad if \quad k \in S(\vec{i}) \\ &= 0 \quad otherwise \end{aligned}$$

Define

$$W_{\vec{i},k} = \frac{\xi_k(S(\vec{i}))}{\mid S(\vec{i}) \mid}$$

Examples of these weight functions $(W_{\vec{i},0}, W_{\vec{i},1}, W_{\vec{i},2})$ for the case of Bézier surfaces ($n = 2$, $s = 3$) of degree 3 and 4, i.e. $d = 3$ and $d = 4$ are shown in Figures 5.4A and 5.4B respectively. Points falling in the region surrounded by the dotted lines have the same weight function associated with them.

It is immediate that the weight functions $W_{\vec{i},k}$ satisfy the positivity and the partition of unity properties. In addition, they satisfy the following properties:

1. Closest vertex property:

$$W_{\vec{i},k} = \delta_{kl} \quad for \quad i_l > max(i_0, \cdots, i_{l-1}, i_{l+1}, \cdots, i_n)$$

   where

$$\begin{aligned} \delta_{kl} &= 1 \quad if \quad k = l, \\ &= 0 \quad otherwise. \end{aligned}$$

2. Symmetry: The weight function is symmetric with respect to the vertices of the simplex. More precisely, let π be a permutation of indices 0 to n. Let $\hat{\pi}(\vec{i}) = \hat{\pi}(i_0, \ldots, i_n) = (i_{\pi(0)}, \ldots, i_{\pi(n)})$. Then $W_{\hat{\pi}^{-1}(\vec{i}),\pi(k)} = W_{\vec{i},k}$.

5.3 Degree Reduction Algorithm

Let $B(v)$ be a n-variate Bézier simplex of degree d. We describe a method of degree reduction for the given simplex to $A(v)$, which is a n-variate Bézier simplex of degree e with $e \leq d$. The idea is as follows:

- Find a Bézier simplex $A_k(v)$ of degree e which interpolates the value and the first e derivatives at the k-th corner point of the original Bézier simplex for each k from 0 to n.

- Use the weight function described in Section 5.2 to form a convex combination of the Bézier control points of the Bézier simplexes $A_k(v)$ to obtain the degree-reduced Bézier simplex $A(v)$.

The major computational task in this algorithm is to compute the Bézier control points of the simplexes $A_k(v)$ from the Bézier control points of the original simplex $B(v)$. For this purpose, we derive the following relationship between their multinomial coefficients.

Theorem 5.1 Let $T^d_{\vec{i},k}$, $|\vec{i}| = d$, be the multinomial coefficients of the Bézier simplex $B(v)$ of degree d with respect to the vertex v_k. Let $T^e_{\vec{j},k}$, $|\vec{j}| = e$, be the multinomial coefficients of the Bézier simplex $A_k(v)$ of degree e, which interpolates the value and the first e derivatives at the k-th corner point of the original Bézier simplex. Let $\vec{i} = \vec{j} + (d-e)\vec{e_k}$. Then

$$T^e_{\vec{j},k} = \frac{d!j_k!}{e!i_k!} T^d_{\vec{i},k} \tag{5.6}$$

Proof Let $T_k(v)$ be the polynomial map of degree e with respect to the vertex v_k, obtained by discarding terms of degree greater than e in the multinomial basis representation of the Bézier simplex $B(v)$ of degree d with respect to the vertex v_k. The expression for $T_k(v)$ is obtained by restricting i_k to be greater than $d-e-1$ in Equation 5.3:

$$T_k(v) = \sum_{i_k \geq d-e} \frac{d!}{i_k!} T^d_{\vec{i},k} M^d_{\vec{i},k}(\alpha_1(v), \ldots, \alpha_{k-1}(v), \alpha_{k+1}(v), \ldots, \alpha_n(v)) \tag{5.7}$$

Now observe that two polynomial maps $T_k(v)$ and $A_k(v)$ of the *same* degree e have the same value and the same first e derivatives at a corner point iff they have the same multinomial coefficients up to degree e in their multinomial basis representation at that corner. The expression for the multinomial basis representation for the Bézier simplex $A_k(v)$ is as follows:

$$A_k(v) = \sum_{\vec{j}} \frac{e!}{j_k!} T^e_{\vec{j},k} M^e_{\vec{j},k}(\alpha_1(v), \ldots, \alpha_{k-1}(v), \alpha_{k+1}(v), \ldots, \alpha_n(v)) \tag{5.8}$$

Now by comparing Equation 5.7 with Equation 5.8 and noting that in Equation (5.7), the summation is taken over all multi-indexes $\vec{i}$, whose norm sum up to d, while in Equation 5.8, the summation is taken over all multi-indexes $\vec{j}$, whose norm sum up to e, we obtain the desired relationship. □

The computational task in the degree reduction algorithm can be described as follows:

- *Step 1: Convert to the truncated multinomial basis of $B(v)$:*
  Compute the multinomial coefficients $T^d_{\vec{i},k}$ for $i_k \geq d - e$ and for $k = 0$ to n from the Bézier control points $C_{\vec{i}}$ of the Bézier simplex $B(v)$, using the basis conversion algorithm described in Section 5.1.

- *Step 2: Convert to the multinomial basis of $A_k(v)$:*
  Compute the multinomial coefficients $T^e_{\vec{j},k}$ of the Bézier simplexes $A_k(v)$ for $k = 0$ to n using Equation 5.6.

- *Step 3: Convert back to the Bézier basis:*
  Compute the Bézier control points of the Bézier simplexes $A_k(v)$ from their multinomial coefficients derived in step 2 for $k = 0$ to n by using the basis conversion algorithm described in Section 5.1. Let $A^e_{\vec{j},k}$ represent the $\vec{j}$-th Bézier control point of the degree-reduced Bézier simplex $A_k(v)$.

- *Step 4: Form a Convex Combination:*
  Form the following convex combination of the Bézier control points computed above:

  $$A_{\vec{j}} = \sum_{k=0}^{n} W_{\vec{j},k} A^e_{\vec{j},k}$$

  where $W_{\vec{j},k}$ is the weight function with $|\ \vec{j}\ | = e$ described in Section 5.2. The degree-reduced n-variate Bézier simplex $A(v)$ of degree e is then given by

  $$A(v) = \sum_{\vec{j}} A_{\vec{j}} B^e_{\vec{j}}(\alpha_0(v), \ldots, \alpha_n(v))$$

5.4 Properties of Degree Reduction Algorithm

The method of degree reduction from a n-variate Bézier simplex of degree d to a n-variate Bézier simplex of degree e, as described above, has the following properties:

- *Symmetry:* The degree reduction method is symmetric with respect to the vertices of the domain simplex. Let $\Pi(C_{\vec{i}}) = C_{\hat{\pi}^{-1}(\vec{i})}$, and let $\mathcal{D}$ be the degree reduction operator. Then Π commutes with $\mathcal{D}$, i.e. $\Pi \circ \mathcal{D}\ (C^n_{\vec{i}}) = \mathcal{D} \circ \Pi\ (C^n_{\vec{i}})$. This follows from the symmetry property of the weight function, described in Section 5.2. In particular, if the given Bézier simplex is symmetric with respect to the vertices of the domain simplex, so is the degree-reduced Bézier simplex.

- *Restriction:* The degree reduction method applied to the boundary of a n-variate Bézier simplex of degree d, which itself is a union of $(n-1)$-variate Bézier simplexes of degree d, yields the boundary of the degree-reduced Bézier simplex. More precisely, the degree reduction operator commutes with the operation of taking the boundary of a Bézier simplex. This property is an easy consequence of the fact that the Bézier representation, and hence the multinomial representation, of a boundary of a Bézier simplex depends only on the boundary Bézier control points of the given simplex. In particular, this property implies that the degree reduction procedure applied to each piece separately of a given C^k piecewise continuous Bézier simplex will produce a C^0 piecewise continuous Bézier simplex. In other words, there will be no cracks between the adjacent Bézier patches after degree reduction. This situation is illustrated in Figure 5.5, where the adjacent Bézier patches have the same boundary control points after degree reduction.

- *Affine invariance:* The degree reduction method is affinely invariant. This follows from the fact that the Bézier simplexes $A_k(v)$ are affine invariant. This in turn follows from the fact that $T^d_{\vec{i},k}$ is affine invariant, which is a consequence of the basis conversion algorithm described in Section 5.1. Therefore, the degree-reduced Bézier simplex $A(v)$, which is obtained by taking a convex combination of the Bézier control points of the Bézier simplexes $A_k(v)$, is also affinely invariant.

- *Hermite Interpolation:* The degree-reduced Bézier simplex of degree e interpolates the given Bézier simplex and its $\lfloor \frac{e-1}{2} \rfloor$ derivatives at the corner point.

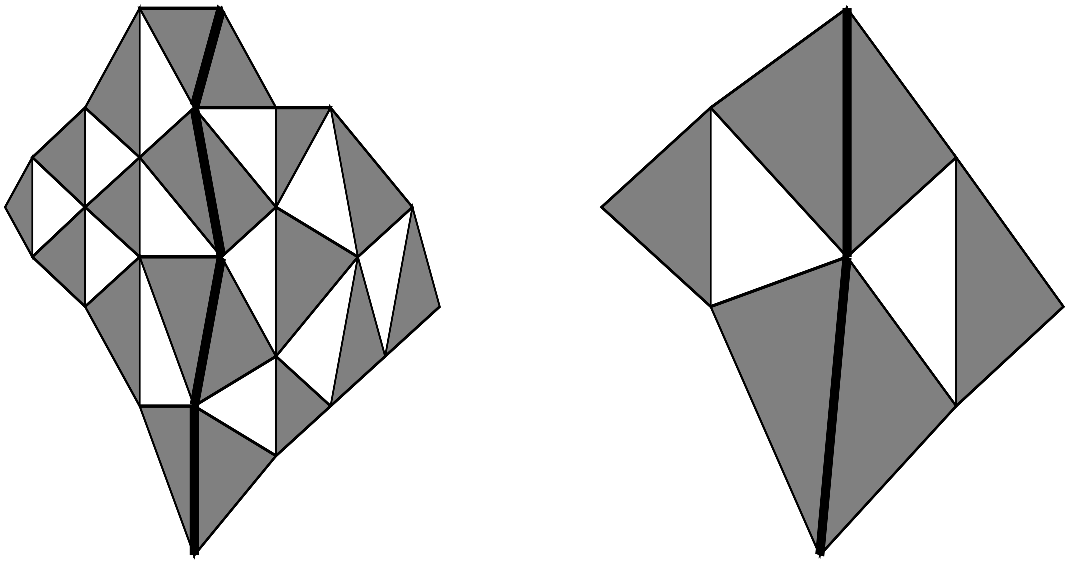

Figure 5.5 Adjacent Bézier patches share the same boundary after degree reduction

This follows from the closest vertex property of the weight function described in Section 5.2 which implies that

$$A_{\vec{j}} = A^{e}_{\vec{j},k} \quad for \quad j_k \geq \lceil \frac{e+1}{2} \rceil$$

In other words, the Bézier control points of the degree-reduced Bézier simplex up to $\lfloor \frac{e-1}{2} \rfloor$ layers at any corner vertex v_k are totally decided by the value of the given Bézier simplex and its $\lfloor \frac{e-1}{2} \rfloor$ derivatives at that vertex.

Remark 1: The above observation yields an alternative view-point for visualizing the degree reduction method. Suppose that the degree-reduced Bézier simplex of degree e is degree-elevated to degree d. In this degree-elevated representation, $\lfloor \frac{e-1}{2} \rfloor$ layers of control points nearest to each vertex are identical to the control points of the original Bézier simplex. The remaining control points are computed by averaging the derivative information from the nearest vertices.

Remark 2: In the case of Bézier curves, for odd e our degree reduction method yields the unique Hermite curve which interpolates the value and the first $\lfloor \frac{e-1}{2} \rfloor$ derivatives at the two end-points of the given Bézier curve. For even e, let H_1 (respectively H_2) be the unique Hermite curve which interpolates the values at the two end-points, $\frac{e}{2}$ derivatives at the end-point Q_1 (resp. Q_2), and the $\frac{e}{2}+1$ derivatives at the end-point Q_2 (resp. Q_1). Then our degree reduction method yields a curve, which is a convex combination of these two curves H_1 and H_2. Forrest [For72a] first suggested this degree reduction method for curves.

- *Precision:* If the given Bézier simplex of degree d happens to be a degree-elevated Bézier simplex of degree e, the degree reduction method reproduces the original Bézier simplex of degree e. This follows from the observation that truncating the monomial representation has no effect.

- *Linearity:* The linearity property states that the degree reduced polynomial map of a sum or difference of two polynomial maps B_1 and B_2 is the sum or difference respectively of the degree-reduced polynomial maps A_1 and A_2. This follows from the observation that each step of the degree reduction procedure is linear.

5.5 Optimal Order of Approximation

In this section, we state and prove the property of optimal order of approximation of the given Bézier simplex $B(v)$ by the degree-reduced Bézier simplex $A(v)$ for functional approximation. Here, we will view these Bézier simplexes as polynomial maps.

The distance between any two points in a Euclidean space of dimension n is defined as $dist(P,Q) = ||P-Q|| = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$, where $(x_1, x_2, \ldots, x_n)$ and $(y_1, y_2, \ldots, y_n)$ are the Cartesian coordinates of the points P and Q. Let h be the diameter of the domain simplex Δ, i.e. $h = \max_{P,Q\in\Delta} dist(P,Q)$. To measure the distance between the original Bézier simplex and the degree-reduced Bézier simplex, we consider the distance between points P and Q on the two simplexes corresponding to the *same* barycentric coordinates. We shall prove that this distance between the original Bézier simplex and the degree-reduced Bézier simplex is bounded by a constant multiplied by h^{e+1}, which is optimal for approximation of real-valued multivariate functions by polynomials of degree e [Fle77]. However, note that there may be points on the degree-reduced simplex with different barycentric coordinates whose actual Euclidean distance to P may be smaller than the distance PQ. Thus it may be possible to achieve better order of approximation than $O(h^{e+1})$ by using the notion of actual Euclidean distance between the two simplexes without regards to their respective parametrizations.

Theorem 5.2 Let $B(v)$ be a n-variate polynomial map of degree d from a Euclidean space E_1 of dimension n to a Euclidean space E_2 of dimension s. Let $A(v)$ be the degree-reduced n-variate polynomial of degree $e \leq d$, obtained by the degree reduction algorithm given in Section 5.3. Let h be the diameter of a domain simplex Δ in E_1. Then

$$||B(v) - A(v)|| \leq C(B, e, n)h^{e+1} \quad \forall v \epsilon \Delta$$

where $C(B, e, n)$ is a constant depending only upon the polynomial B, and the constants e and n.

Proof Let $B(v) = (B_1(v), \ldots, B_s(v))$, and let $A(v) = (A_1(v), \ldots, A_s(v))$. We will prove that for each component $i = 1, 2, \ldots, s$, $|B_i(v) - A_i(v)| \leq C(B, e, n)h^{e+1}$. Therefore, in the following discussion, we will drop the subscript i and assume that the polynomial map B is real-valued.

Now let $f(v)$ be the difference between the given polynomial function $B(v)$ of degree d and its truncated multinomial basis representation $T_k(v)$ of degree e computed in step 1 of the algorithm described in Section 5.3. In other words, $f(v) = B(v) - T_k(v)$ for any arbitrary, but fixed k. To simplify notation, we make the choice $k = 0$. We first establish that the degree reduction procedure applied to $f(v)$ yields a function $g(v)$ such that $g(v) \leq C(B, e, n)h^{e+1}$ $\forall v \epsilon \Delta$. We achieve this in four lemmas corresponding to the four steps in the computation of the degree reduction algorithm described in Section 5.3.

Let $T^d_{\vec{i}}(f) =$ Multinomial coefficients of the polynomial $f(v)$ of degree d with respect to the vertex v_0

$T^e_{\vec{j}}(g) =$ Multinomial coefficients of the degree-reduced polynomial $g(v)$ of degree e with respect to the vertex v_0

$A^e_{\vec{j}}(g) =$ *Bézier* control points of $g(v)$

In the following discussion, $C_0(B, e, n), \cdots, C_4(B, e, n)$ will be used to denote constants depending only upon the polynomial B and the constants e and n. Given a polynomial function f from R^n to R, we denote the partial derivative of f with respect to the coordinates $x_{j_1}, \ldots, x_{j_m}$ by $f_{j_1, j_2, \ldots, j_m}$. Moreover, if $\vec{j} = (j_1, \ldots, j_m)$, we write $D^{\vec{j}} f = f_{j_1, \ldots, j_m}$. Let λ_i denote the vector from vertex v_0 to v_i for $i = 1$ to n, and let $h_i = ||v_i - v_0|| = ||\lambda_i||$. Let $\lambda_{i,l}$ means the l-th component of the vector λ_i. Moreover, let $\lambda_{i,.}$ denote any component of the vector λ_i. Also let $\vec{k} = (k_1, \ldots, k_n)$ with $|\vec{k}| = m$. We denote $\nabla^{\vec{k}} f$ to represent the directional derivatives of f with k_i

directional derivatives taken in the direction λ_i. Thus, for example, when $\vec{k} = (2,1)$, so that $n = 2$ and $m = 3$, $\nabla^{(2,1)} f$ represents the directional derivative of f taken twice in the direction λ_1 and once in the direction λ_2. By chain rule, it follows that

$$\nabla^{\vec{k}} f = \sum_{<\vec{j}>} (D^{\vec{j}} f) \lambda_{1,j_1} \ldots \lambda_{n,j_m} \tag{5.9}$$

where $\lambda_{i,.}$ is repeated k_i times and the sum in Equation (5.9), denoted by $< \vec{j} >$, is to be taken over all n^m different sequences $\vec{j} = (j_1, \ldots, j_m)$ of integers in $1, 2, \ldots, n$. As an example, $\nabla^{(2,1)} f = (D^{111} f)\lambda_{1,1}\lambda_{1,1}\lambda_{2,1} + (D^{112} f)\lambda_{1,1}\lambda_{1,1}\lambda_{2,2} + (D^{121} f)\lambda_{1,1}\lambda_{1,2}\lambda_{2,1} + (D^{122} f)\lambda_{1,1}\lambda_{1,2}\lambda_{2,2} + (D^{211} f)\lambda_{1,2}\lambda_{1,1}\lambda_{2,1} + (D^{212} f)\lambda_{1,2}\lambda_{1,1}\lambda_{2,2} + (D^{221} f)\lambda_{1,2}\lambda_{1,2}\lambda_{2,1} + (D^{222} f)\lambda_{1,2}\lambda_{1,2}\lambda_{2,2}$.

Lemma 5.1 $|T^d_{\vec{i}}(f)| \leq C_1(B,e,n) h^{e+1} \quad \forall v \epsilon \Delta.$

Proof We first note that by the well-known property [Fle77] of Taylor series approximation,

$$|f(v)| \leq C_0(B,e,n) h^{e+1} \;\; \forall v \epsilon \Delta. \tag{5.10}$$

In fact, $C_0(B,e,n) = \frac{K(n-1)^{\frac{e+1}{2}}}{(e+1)!}$, where $K = max|B_{i_1 i_2 \ldots i_{e+1}}|$ over all possible sequences $(i_1, \ldots, i_{e+1})$, where each i_k is one of the coordinates from x_1 to x_n. Moreover,

$$|D^{\vec{j}} f| \leq C_4(B,e,n) h^{e+1-|\vec{j}|} \;\; \forall v \epsilon \Delta. \tag{5.11}$$

A simple calculation yields the following relationship between the multinomial coefficients $T^d_{\vec{i},k}$ and the directional derivatives of f, where $\vec{i} = (i_0, \cdots, i_n)$, and $\vec{i'} = (i_1, \cdots, i_n)$:

$$T^d_{\vec{i},0} = \frac{i_0!}{d!} \nabla^{\vec{i'}} f \tag{5.12}$$

The desired inequality now follows from Equations 5.12, 5.11, and 5.9:
$|T^d_{\vec{i},0}| = |\frac{i_0!}{d!} \nabla^{\vec{i'}} f|$
$\leq \frac{i_0!}{d!} |\sum_{<\vec{j}>} (D^{\vec{j}} f) \lambda_{1,j_1} \ldots \lambda_{n,j_m}| \quad$ where $\quad m = d - i_0$
$\leq \frac{i_0!}{d!} C_4(B,e,n) h^{e+1-m} h^m \quad$ because $\quad |\lambda_{i,l}| \leq h$
$\leq C_1(B,e,n) h^{e+1}$. □

Lemma 5.2 $|T^e_{\vec{j},0}(g)| \leq C_2(B,e,n) h^{e+1} \quad \forall v \epsilon \Delta.$

Proof The proof follows immediately from Lemma 5.1 and Equation 5.6 used in step 2 of the degree reduction algorithm. $\square$

Lemma 5.3 $|A^e_{\vec{j}}(g)| \leq C_3(B,e,n)h^{e+1} \quad \forall v \epsilon \Delta.$

Proof The proof follows from Lemma 5.2 and the observation that the Bézier control points $A^e_{\vec{j}}(g)$ are obtained by taking finite sums of multinomial coefficients $T^e_{\vec{j},0}(g)$, as described in step 3 of the degree reduction algorithm. $\square$

Lemma 5.4 $|g(v)| \leq C_3(B,e,n)h^{e+1} \quad \forall v \epsilon \Delta.$

Proof By definition, $g(v) = \sum_{\vec{j}} \left(\sum_{k=0}^{n} W_{\vec{j},k} A^e_{\vec{j},k}(f) \right) B^e_{\vec{j}}(v)$
Therefore,
$|g(v)|$
$\leq \sum_{\vec{j}} \sum_{k=0}^{n} |W_{\vec{j},k}| \ \ |A^e_{\vec{j},k}(f)| \ \ |B^e_{\vec{j}}(v)|$
$\leq C_3(B,e,n)h^{e+1} \left(\sum_{\vec{j}} \sum_{k=0}^{n} |W_{\vec{j},k}||B^e_{\vec{j}}| \right)$ by Lemma 5.3
$\leq C_3(B,e,n)h^{e+1} \left(\sum_{\vec{j}} \sum_{k=0}^{n} W_{\vec{j},k}|B^e_{\vec{j}}| \right)$ by the positivity of the weight functions
$\leq C_3(B,e,n)h^{e+1} \left(\sum_{\vec{j}} |B^e_{\vec{j}}| \right)$ by the partition of unity property of weight functions
$\leq C_3(B,e,n)h^{e+1}$ by the positivity and partition of unity properties of the Bernstein polynomials. $\square$

We now complete the proof of Theorem 5.2. In the following, given any function $f(v)$, let $A(f(v))$ denote the polynomial of degree e obtained by applying the degree reduction algorithm to the polynomial $f(v)$.

Proof of the theorem (continued):
$|B(v) - A(v)|$
$= |B(v) - A(B(v))|$ by notation
$= |B(v) - T_k(v) + T_k(v) - A(B(v))|$
$= |f(v) + A(T_k(v)) - A(B(v)))|$ by the precision property
because the polynomial $T_k(v)$ has degree e
$= |f(v) + A(T_k(v) - B(v))|$ by the linearity property of the degree reduction procedure
$\leq |f(v)| + |g(v)|$ by triangular inequality
$\leq C_0(B,e,n)h^{e+1} + C_3(B,e,n)h^{e+1}$ by Equation 5.10 and Lemma 5.4
$\leq C(B,e,n)h^{e+1}$. $\square$

This chapter described a method of degree reduction for Bézier simplexes which works in any dimension, has several useful properties, and is easy to implement. In particular, the method requires taking only sums and differences with very few multiplications, followed by convex combinations of given control points. The method restricted to triangular Bézier patches yields a new technique of degree reduction for surfaces. The degree reduction method described here is potentially useful in rendering and intersection calculations, in improving curvature plots and in exchanging data between different geometric modeling systems.

Chapter 6

Perturbation to Patches with Multiple Representations

The previous chapter described a degree reduction algorithm to approximate a high degree parametric Bézier surface patch by a low degree parametric Bézier surface patch. This chapter describes a perturbation algorithm to approximate a low degree parametric Bézier surface patch by a low degree surface patch with multiple representations. More specifically, this chapter describes a variant of the least squares technique to approximate a triangular quadratic rational Bézier surface patch by a surface patch with both implicit and parametric representation, which was introduced in Chapter 2. The same technique is also used to approximate a rectangular biquadratic rational Bézier surface patch by a surface patch with both implicit and parametric representation, which was introduced in Chapter 3. Section 6.1 discusses the difficulties associated with least squares approximation using implicit surfaces. Section 6.2 describes a perturbation algorithm that minimizes over the domain of the parametric patch the integral of the square of the deviation of the parametric patch from a patch with multiple representations. Section 6.3 discusses the properties of the perturbation algorithm. Section 6.4 describes the piecewise continuous approximation of a surface consisting of parametric patches by a surface consisting of low degree patches with both implicit and parametric representations.

6.1 Least Squares Approximation

Least squares approximation has been studied mostly in the context of the scattered data fitting problem, where given a finite number of data points in space, a curve (or surface) is to be constructed such that the distance between the data points and the curve (or surface) is small for all the data points. The technique of least squares approximation as applied to the approximation of parametric surfaces by implicit surfaces is similar. Least squares fitting by parametric polynomial curves and surfaces is treated in a number of papers and textbooks [LH74]. In contrast, a thorough search made by Pratt [Pra87] revealed only a few treatments of least squares fitting by implicit curves. In addition, Pratt [Pra87], Patrikalakis [PK89], and Moore and Warren [MW90] have dealt with the least squares fitting by non-planar implicit surfaces.

One of the difficulties associated with this problem is the measurement of distance between a point and a surface. Ideally, this distance should be taken as the geometric distance. The *geometric distance* of a point p from a surface S is the distance from p to the nearest point of S, that is, the minimum over all points p' of S of the Euclidean distance from p' to p. However, the geometric distance is neither computationally nor algebraically convenient [Pra87].

Therefore, it is customary to use the algebraic distance of a point p to an implicit surface S, where S is defined as the zero contour of a polynomial function $f(x_1, x_2, x_3)$. The algebraic distance is defined to be the value of the function $f(x_1, x_2, x_3)$ at the point p. The error measure for this approximation is $\sum f(p)^2$, where the sum is taken over all the data points p. If f minimizes this error, f is called the least squares approximation to the given data. However, since the same implicit surface S can also be represented by the zero contour of the function $c \times f(x_1, x_2, x_3)$, where $c \neq 0$, the algebraic distance $f(p)$ is normalized by scaling the function f so as to set some function of the coefficients of the polynomial f to a constant, usually unity. For conics, for example, a quadratic function of the coefficients is usually set to unity. Distance computed in this manner is called algebraic distance and is evaluated using a fixed representative polynomial $c \times f$, chosen independently of the point p.

Pratt [Pra87] discusses some of the difficulties and pitfalls associated with the least squares approximation using algebraic distance. Bias may be introduced into the fitting surface by choosing a normalization for the function f. For example, if a fitting quadric surface is assumed to be of the form $\sum_{\vec{i}} A_{\vec{i}} x_1^{i_1} x_2^{i_2} x_3^{i_3} = 0$, where the summation is taken over all the multi-indexes $\vec{i} = (i_1, i_2, i_3, i_4)$ with $|\vec{i}| = 2$ and the constant coefficient A_{0002} is chosen to be -1, then under the least squares error measure, the sum $\sum_k (\sum_{\vec{i}} A_{\vec{i}} x_{1,k}^{i_1} x_{2,k}^{i_2} x_{3,k}^{i_3})^2$, is minimized, where $(x_{1,k}, x_{2,k}, x_{3,k})$ are the coordinates of the data points p_k. This restriction introduces a bias against fitting surfaces that pass near the origin. It is easily seen that no quadric surface passing through the origin can be of this form, and surfaces that pass near the origin must have very large coefficients $A_{\vec{i}}$. Unfortunately, some constraint upon the coefficients of the polynomials is necessary since the trivial solution, that all the coefficients are zero, always minimizes the least squares error. In this work, we restrict the coefficients of the polynomial $f(x_1, x_2, x_3)$ so that the sum of the squares of the coefficients is unity. This normalization treats all the coefficients in the same manner.

Another major difficulty of approximating parametric surfaces by implicit surfaces has been the presence of self-intersections or extra sheets in the implicit representation of the surfaces. These problems can be illustrated with curves. Figure 6.1 depicts the dark portion of a branch of hyperbola, a parametric curve. The implicit representation of the curve contains the other undesirable branch of the hyperbola, which may subsequently interfere with the modeling operations. Figure 6.2 depicts the parametric curve, $x_1(t) = t^3 - t$, $x_2(t) = t^2 - 1$, where the parameters are restricted to represent only the dark portion of the curve. The implicit representation of the curve, $x_2^2 = x_1^3 + x_1^2$, contains the undesirable self-intersection, which again may interfere with subsequent modeling operations. The geometric models produced by Dahmen [Dah89], Guo [Guo90], and Patrikalakis and Kriezis [PK89] may contain extra undesirable sheets or self-intersections of implicit surfaces. Moore and Warren [MW90] use auxiliary distance data to avoid these extraneous sheets. In this work, the extraneous sheets are avoided by using the surface patches introduced in Chapters 2 and 3 as will be described later in Section 6.3.

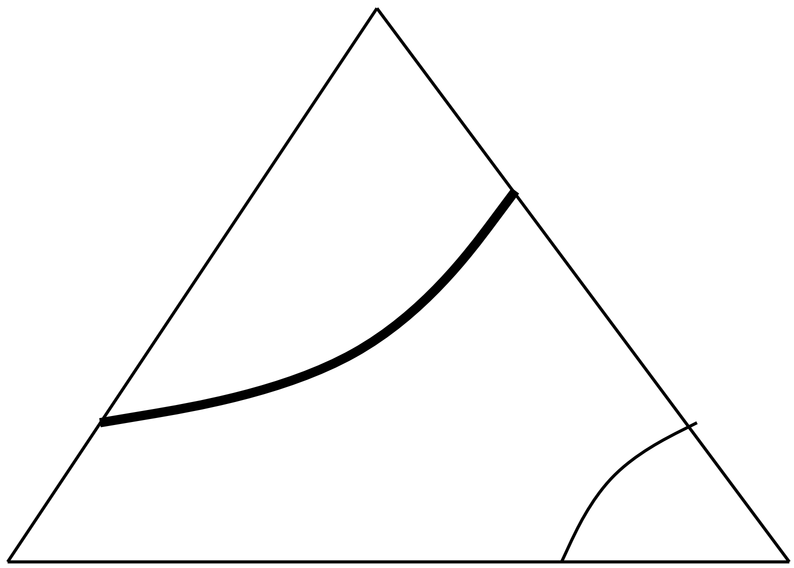

Figure 6.1 The approximating curve with an extra branch

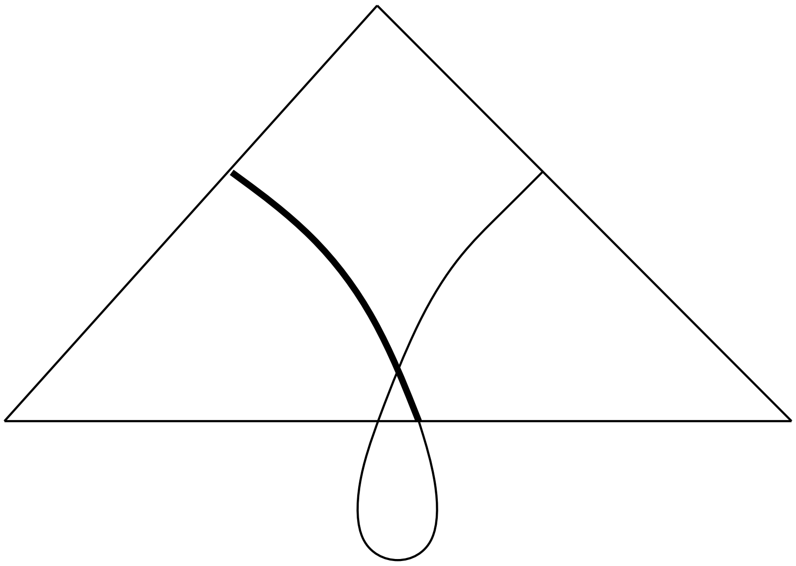

Figure 6.2 The approximating curve with self-intersection

6.2 Perturbation Algorithm

This section describes an algorithm to approximate a triangular quadratic Bézier surface patch by a surface patch with the additional property of having implicit degree *two*, which was introduced in Chapter 2. A similar algorithm can be used for approximating a rectangular biquadratic Bézier surface patch by a surface patch with the additional property of having implicit degree *three*, which was introduced in Chapter 3. Note that the triangular quadratic and rectangular biquadratic Bézier patches could be represented exactly as implicit surfaces of degree four and eight respectively [SA85], with the possible complications of having extra sheets or self-intersections in the implicit representation. Moreover, the general implicitization procedure using the elimination theory and resultants would be computationally complex and expensive [Hof90, Hof89]. The approximation algorithm achieves both low degree implicit representations and avoids the undesirable features of extra sheets and self-intersections.

Suppose now that a triangular rational quadratic Bézier surface patch $B(v)$ is given with control points $C_{\vec{j}}$ and weights $w_{\vec{j}}$ for multi-index $\vec{j} = (j_1, j_2, j_3)$ so that

$$B(v) = \frac{\sum_{\vec{j}} w_{\vec{j}} C_{\vec{j}} B^2_{\vec{j}}(\gamma_1(v), \gamma_2(v), \gamma_3(v))}{\sum_{\vec{j}} w_{\vec{j}} B^2_{\vec{j}}(\gamma_1(v), \gamma_2(v), \gamma_3(v))} \tag{6.1}$$

where $(\gamma_1, \gamma_2, \gamma_3)$ are the barycentric coordinates of a point v with respect to a domain triangle Δ and the summation is taken over all multi-indexes $\vec{j}$ with $|\vec{j}| = 2$. This patch will be approximated by a triangular surface patch that has a rational triangular quadratic Bézier surface representation with control points $\hat{C}_{\vec{j}}$ and weights $\hat{w}_{\vec{j}}$ with respect to the same domain triangle Δ, and also has a quadric implicit representation.

The approximation algorithm is as follows:

1. *Compute the enclosing tetrahedron* $\hat{\Delta}$: Let P be the plane defined by the three corner control points of the given patch. Since the given patch is a quadratic Bézier surface patch, each boundary curve lies in the plane determined by the three control points on the boundary. For the degenerate case of collinear control points, the boundary plane is taken to be the plane passing through the boundary line and perpendicular to P. If the three boundary planes intersect in a unique point f in R^3, then the three corner control points, together with the intersection point f, also referred to as the focal vertex, define the enclosing tetrahedron $\hat{\Delta}$. If the three boundary planes are parallel, heuristics described by Dahmen [Dah89] can be used to find a suitable point f, which together with the three corner control points define the enclosing tetrahedron $\hat{\Delta}$. One of the heuristics is to choose the point f at a sufficiently large distance away on the line perpendicular to the plane P and passing through the centroid of the triangle, defined by the three corner control points. The distance is large in comparison to the size of the prescribed control net, and usually also takes into account the information regarding normals at the three corner control points so that the chances of enclosing the desired implicit surface within the chosen tetrahedron are improved.

2. *Compute the deviation of the given patch from a quadric patch:* Let the trivariate implicit representation of the approximating quadric patch with respect to the tetrahedron $\hat{\Delta}$ with undetermined coefficients $b_{\vec{i}}$ for the multi-index $\vec{i} = (i_1, i_2, i_3, i_4)$ be given as

$$Q(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = \sum_{\vec{i}} \binom{2}{\vec{i}} b_{\vec{i}} \alpha_1^{i_1} \alpha_2^{i_2} \alpha_3^{i_3} \alpha_4^{i_4} \tag{6.2}$$

where α_1, α_2, α_3 and α_4 are the barycentric coordinates of the tetrahedron $\hat{\Delta}$. Now the deviation of the given patch from a quadric patch can be obtained by substituting the parametric equation 6.1 of the given patch into the implicit quadric equation 6.2. To obtain an explicit expression for this deviation, the parametric equations of the given patch are rewritten in terms of homogeneous coordinates (x_1, x_2, x_3, x_4):

$$\begin{aligned} x_k &= \sum_{\vec{j}} w_{\vec{j}} C_{\vec{j},k} B^2_{\vec{j}}(\gamma_1, \gamma_2, \gamma_3) \;\; for \;\; k = 1, \cdots, 3 \\ x_4 &= \sum_{\vec{j}} w_{\vec{j}} B^2_{\vec{j}}(\gamma_1, \gamma_2, \gamma_3) \end{aligned}$$

where $C_{\vec{j},k}$ denotes the k-th component of the control point $C_{\vec{j}}$. Let the barycentric coordinates α_i of the tetrahedron $\hat{\Delta}$ be expressed in terms of the homogeneous coordinates x_l's so that $\alpha_k = \sum_{l=1}^{4} s_{kl} x_l$ for $k = 1, \cdots, 4$. Using Equation 6.2, the above relationship of α's in terms of x's, and the parametric equation of the patch expressing x's in terms of γ's, the deviation is defined as:

$$D = \sum_{\vec{i}} b_{\vec{i}} A_{\vec{i}}(\gamma_1, \gamma_2, \gamma_3)$$

where

$$A_{\vec{i}}(\gamma_1, \gamma_2, \gamma_3) = \binom{2}{\vec{i}} \prod_{k=1}^{4} \left(\sum_{l=1}^{4} s_{kl} x_l(\gamma_1, \gamma_2, \gamma_3)\right)^{i_k}.$$

Furthermore, by introducing the column matrix $\mathbf{b}$ for the undetermined coefficients $b_{\vec{i}}$, and the row matrix $\mathbf{A}(\gamma_1, \gamma_2, \gamma_3)$ for the entries $A_{\vec{i}}(\gamma_1, \gamma_2, \gamma_3)$, the deviation can be simply expressed in matrix notation as

$$D = \mathbf{A}(\gamma_1, \gamma_2, \gamma_3)\mathbf{b}.$$

Observe that the deviation is a polynomial expression of degree four in terms of $(\gamma_1, \gamma_2, \gamma_3)$, but is a linear expression in terms of the undetermined coefficients of the matrix $\mathbf{b}$. Also observe that if the given patch were a quadric patch given by the implicit quadric equation 6.2, then the above deviation would be zero.

3. *Compute the least squares deviation patch:* Since the deviation at any point of the patch can be positive or negative, the square of the deviation integrated over the parametric domain Δ is minimized to compute the least squares deviation patch. However, there are two important constraints. First, the implicit weight b_{0002} associated with the focal vertex f of the tetrahedron is taken to be zero. This is a necessary and sufficient condition to ensure a triangular surface patch with multiple representations as described in Chapter 2. Next,

to ensure that the remaining coefficients $b_{\vec{i}}$ are not all zero, the constraint $\hat{\mathbf{b}}^t\hat{\mathbf{b}} = 1$ is imposed, where the superscript t denotes the transpose of a matrix. Thus, denoting the matrices $\hat{\mathbf{b}}$ and $\hat{\mathbf{A}}(\gamma_1, \gamma_2, \gamma_3)$ without the entries b_{0002} and A_{0002} by $\mathbf{b}$ and $\mathbf{A}(\gamma_1, \gamma_2, \gamma_3)$, the problem reduces to minimizing the integral $\int_\Delta(\hat{\mathbf{A}}(\gamma_1, \gamma_2, \gamma_3)\hat{\mathbf{b}})^2$, subject to the constraint that $\hat{\mathbf{b}}^t\hat{\mathbf{b}} = 1$. Observe that the integrand is a polynomial expression of degree eight in $(\gamma_1, \gamma_2, \gamma_3)$, but is only a quadratic expression in the undetermined coefficients. In fact, by denoting $M = \int_\Delta \hat{\mathbf{A}}^t\hat{\mathbf{A}}$, the problem is reduced to minimizing the quadratic expression $\hat{\mathbf{b}}^t M\hat{\mathbf{b}}$ subject to the constraint that $\hat{\mathbf{b}}^t\hat{\mathbf{b}} = 1$. This is equivalent to solving the eigenvalue problem $M\hat{\mathbf{b}} = \lambda\hat{\mathbf{b}}$ [LH74], where M is a symmetric matrix. This problem is easily solved using a method like Rayleigh quotient iteration [Ste73]. Since the deviation of the patch is given by $\lambda(\hat{\mathbf{b}}^t\hat{\mathbf{b}})$, the eigenvector corresponding to the smallest eigenvalue is selected as the undetermined coefficients.

A similar perturbation algorithm can be used to approximate a rectangular rational biquadratic Bézier surface patch by a rectangular surface patch with the additional property of having a trivariate cubic implicit representation as described in Chapter 3. The only change in the above algorithm is in step 3, where the following constraints are imposed: $b_{i_1 i_2 i_3 i_4} = 0$ for $i_3 + i_4 = 0$ and for $i_3 + i_4 = 3$.

6.3 Properties of Perturbation Algorithm

The perturbation algorithm has the following properties:

- *No self-intersections or extra sheets:*

  One of the key advantages of the perturbation scheme presented in the previous section is that the approximating patches do not contain extraneous sheets or self-intersections within the enclosing tetrahedron $\hat{\Delta}$, which is the region of interest. In the case of triangular surface patches, this property follows from the fact that the implicit weight $b_{0002} = 0$, which ensures that the focal vertex f of the tetrahedron lies on the quadric surface. By Bezout's theorem, any straight line joining the focal vertex f and any point on the base plane P of the tetrahedron intersects the quadric surface patch in at most two points. Since one of the points is f, there is at most one point (and in fact exactly one point due to the minimization of the least squares deviation of the approximating patch) of the quadric surface inside the tetrahedron $\hat{\Delta}$. Similarly, in the case of rectangular surface patch, the constraints $b_{1 i_2 i_3 i_4} = 0$ for $i_3 + i_4 = 0$ and $i_3 + i_4 = 3$ ensure that the two skew edges p_1p_2 and p_3p_4 of the tetrahedron lie on the cubic surface patch, where $p_1, \cdots, p_4$ are the vertices of the tetrahedron $\hat{\Delta}$. Again by Bezout's theorem, any straight line joining a point P on p_1p_2 and a point Q on p_3p_4 intersects the cubic surface patch in at most three points, two of which are P and Q. Therefore, it intersects the remaining patch in at most one more point, and this by the same argument as before, in exactly one point inside the enclosing tetrahedron $\hat{\Delta}$.

- *Quadric / Cubic Precision:* For the case of triangular surface patches, the perturbation algorithm has quadric precision, that is, if the given Bézier surface patch is a quadric patch (implicit degree two), the approximation algorithm yields the same patch. This follows easily from the fact that the deviation of the quadric representation of the given patch from itself is zero. By a similar argument, it follows that in the case of rectangular surface patches, the approximation algorithm has cubic precision, that is, if the given Bézier surface patch is a cubic patch (implicit degree three), the approximating algorithm yields the same patch.

6.4 Piecewise Continuous Approximation

This section combines the techniques of Chapters 4 and 5 to approximate a piecewise continuous mesh of high degree parametric triangular or rectangular Bézier patches with n-sided holes by a piecewise continuous mesh of triangular or rectangular low degree surface patches with both parametric and implicit representations.

The algorithm is as follows:

1. An n-sided hole is filled with a rationally controlled S-patch using techniques described in Chapter 4. These S-patches are then converted by a multiprojective polarization algorithm into triangular rational Bézier surface patches. This algorithm converts an n-sided S-patch of depth d into a collection of n triangular Bézier surface patches of degree $d(n-2)$ [LD89]. Since the holes are filled smoothly, we obtain a piecewise continuous mesh of high degree parametric patches with a triangular or rectangular Bézier representation.

2. Apply the degree reduction algorithm of Chapter 5 to approximate the triangular rational Bézier surface patches by quadratic triangular rational Bézier sur-

face patches and the rectangular rational Bézier surface patches by biquadratic rectangular Bézier surface patches. Since the degree reduction algorithm preserves the boundary continuity, we obtain a piecewise continuous mesh of triangular quadratic and rectangular biquadratic Bézier surface patches.

3. Perturb the above piecewise continuous mesh of N triangular quadratic and rectangular biquadratic Bézier surface patches to a mesh of N triangular and rectangular surface patches with both parametric and implicit representations by using *global* minimization of the integral of the square of the deviation of the given mesh from the approximating mesh. This amounts to the minimization of the integral $\sum_{I=1}^{N} \int_{\Delta_I} (\hat{\mathbf{A}}^I \hat{\mathbf{b}}^I)^2$, where the undetermined coefficients in $\hat{\mathbf{b}}$ are taken to be identical on the faces which are shared by the adjacent tetrahedra, and are subject to the additional constraint that not all the coefficients are zero. The perturbation algorithm applied separately to two adjacent boundary continuous surface patches will not produce boundary continuous surface patches, but this problem is overcome by imposing the global minimization criterion where the coefficients are shared by the common faces of the tetrahedra. This ensures boundary continuity between adjacent surface patches and produces a piecewise continuous mesh of approximating surface patches. The geometry of the piecewise continuous mesh should be such that the constraints $b_{0002} = 0$ for the triangular patches and the constraints $b_{i_1 i_2 i_3 i_4} = 0$ for $i_3 + i_4 = 0$ and $i_3 + i_4 = 3$ for the rectangular patches allow for a non-trivial solution of the global minimization problem. This is possible, for example, when the reference tetrahedra for the surface patches share the same focal vertex. This condition restricts the class of possible objects to those that are star-shaped, in which each point on the object is visible from the focal vertex. The center of the star-shaped object serves as the focal vertex of the tetrahedra. Creation of surfaces whose reference tetrahedra are not adjacent or which have different focal vertices is an interesting problem worth examination.

This chapter has described a perturbation algorithm for approximating triangular quadratic Bézier surface patches and rectangular biquadratic Bézier surface patches by surface patches with the additional property of having trivariate implicit representation of degree two and three respectively. The approximating patches do not have self-intersections or extra sheets within their defining tetrahedra. The degree reduction and perturbation method together allow the approximation of smooth surfaces consisting of high degree Bézier surface patches by boundary continuous surfaces consisting of low degree patches with both parametric Bézier and trivariate implicit representations.

Chapter 7

Conclusions

This work introduced low degree surface patches with both parametric Bézier representation and implicit trivariate representation. Specific geometric constructions were presented to design triangular quadratic parametric Bézier surface patches with implicit degree two and rectangular biquadratic parametric Bézier surface patches with implicit degree three. Each representation can be conveniently computed from the other using simple arithmetic operations. The resulting surface patches inherit all the complimentary advantages of both representations. In particular, they possess the highly desirable geometric properties of interpolation at the corner control points and the convex hull property.

However, this is only a first step towards the goal of using surface patches with multiple representations in a practical CAGD system. There are a number of interesting problems concerning surface patches with both representations that still need to be investigated.

- *Continuity:* A practical CAGD system requires flexible and geometrically continuous techniques for joining surface patches smoothly with tangent or curvature continuity. Local editing properties enabling a designer to change the shape of a surface locally without altering the shape of the surface elsewhere are also very desirable. These properties, for example, are exhibited by B-spline surfaces, which are generalization of Bézier surfaces. Therefore an important contribution would be to generalize the surface patches with multiple representations introduced in this work so that they can be joined smoothly and conveniently with tangent or curvature continuity. Creation of C^1 piecewise smooth surfaces using quadric and cubic patches with Bézier representations have proved difficult so far due to severe constraints on the geometry of adjacent patches. However, given the extra freedom of higher degrees, it should be possible to build models with very complex smooth geometry using a C^1 mesh of multiple representation surface patches. Therefore, there is a need to extend the class of surface patches with multiple representations.

- *Further Extending the Class of Surface Patches with Multiple Representations:* Implicit quartic surfaces such as tori and cyclides are increasingly being used for modeling and blending in CAGD [Pra90, Boe90, VCH88]. It is also known that tori and cyclides admit rational biquadratic parametrizations. It would be useful to find necessary and sufficient geometric restrictions on the Bézier control to create cyclides or a more general class of quartic implicit surfaces which includes cyclides.

  Hopcroft and Hoffmann [HH87] have used implicit quartic surfaces to blend quadrics. Warren [War89, War87] has used low degree implicit surfaces to blend implicit surfaces. Several interesting and useful blending surfaces, such as blends of two axially intersecting circular cylinders, have been represented implicitly. It would be very useful to find rational parametrizations for these blending surfaces whenever possible.

  In certain special cases of high degree implicit surfaces, it is easy to construct surface patches with parametric representations using the techniques presented in this work. For example, a monoidal surface, an implicit surface of degree d with a point of multiplicity $d-1$ can be easily parametrized [Hof89]. It would be useful to find other geometric conditions that extend the class of surfaces with both representations.

  Another possible approach to cutting down the implicit degree of a Bézier surface patch is to impose base points [Chi90, War90]. The relationship between geometric constraints on the control net and the base points is worth investigating.

  Another fruitful extension would be to introduce surface patches with different kinds of multiple representations. For example, the implicit representation need not remain confined within a tetrahedron, but may instead be bounded by a convex polytope in three dimensions. Similarly, the parametric representation need not be a Bézier representation, but may instead be a B-spline or Lagrange representation.

- *Properties of Surfaces with Multiple Representations:* This work has introduced surface patches with multiple representations. However the full potential advantages of these patches to create or approximate useful geometric models in a practical setting still needs to be demonstrated conclusively. This task requires a detailed investigation of surface interrogation primitives such as offsets, blends, display, symbolic manipulation, intersection algorithms etc., for surfaces with multiple representations.

The second half of this thesis focused on approximating a piecewise continuous mesh of high degree parametric Bézier surface patches with n-sided holes by a piecewise continuous mesh of low degree surface patches with multiple representations which were introduced in the first part of the work. Chapter 4 described a new technique for filling an n-sided hole using a single parametric patch with a geometrically intuitive compact representation. The technique unified many known solutions for filling a hole with different number of sides, and different degrees of cross-boundary derivative continuity. Chapter 5 presented a new algorithm for approximating high degree parametric Bézier surfaces by low degree parametric Bézier surfaces. The algorithm works equally well in higher dimensions for Bézier simplexes. This degree reduction algorithm maintains boundary continuity between adjacent patches. Moreover, it has the desirable affine invariance and optimal order of approximation properties. Chapter 6 described an algorithm to approximate low degree parametric surface patches by low degree surface patches with multiple representations. The algorithm minimizes the integral of the square of the deviation of the given patch from a surface patch with multiple representation. The problem reduces to the problem of finding the eigenvalue of a matrix and can be solved easily with standard techniques. The solution has the highly desirable property of avoiding extra sheets or self-intersections.

However, there still remains many interesting problems that are worth investigating.

- *N-sided Hole Filling Patch with Additional Properties:* It seems possible to reduce the degree of the n-sided hole filling patch, if the patches are to be joined smoothly not with parametric C^1 or C^2 continuity, but with only G^1 or G^2 geometric continuity [GW89a, LD90]. A general theory for constructing a geometrically continuous n-sided hole filling patch would be extremely useful. Moreover, since the proposed solution for filling an n-sided hole is not unique (for example, the weights can be chosen arbitrarily), it would be a useful exercise to show how to choose these weights to enforce certain constraints such as the minimization of the total curvature.

- *Approximation Problems:* An important problem in CAGD is to find a piecewise continuous smooth surface that fits scattered data with good approximation properties. Sometimes the scattered data points have prescribed normals which also need to be interpolated or approximated. A popular technique to solve this problem is to use the least squares method. It would be worthwhile to solve the scattered data fitting problem using surface patches with multiple representations.

  Moreover, in view of the difficulty of approximating implicit surfaces by lower degree implicit surfaces [Hof89], the approximation of implicit surfaces by surfaces with both implicit and parametric representations looks more promising.

- *Data-Dependent Triangulation:* Given a Bézier surface patch, the approximation scheme in chapter 6 creates a tetrahedron within which the implicit surface is bounded. This choice of tetrahedron is not unique. Results in two dimensions for data-dependent triangulations indicate that the quality of the approximation can vary significantly depending upon the choice of triangulation [DLR90, Bro91]. Similarly, in the free-form modeling schemes of Dahmen [Dah89], Guo [Guo90], and Bajaj and Ihm [BI92] using quadric, cubic, and quintic implicit surfaces, the choice of the tetrahedra affects the shape and geometric properties of the model. However, this dependence of the quality of the approximation or the geometry of the model on the subdivision of space by the tetrahedra is poorly understood. A related problem is the adaptive, data-dependent subdivision of space to improve the quality of the approximation in regions of high curvature or to control the shape of the geometric model more effectively in desired regions. Future research providing a better understanding of these problems would have significant impact in the field.

It has been interesting and fruitful to explore surface patches with multiple representations. Approximation of surfaces by these patches is useful. Nevertheless it is important to generalize these patches in order to gain the additional flexibility of joining them with tangent or curvature continuity. This investigation is fundamental to realizing their full potential in Computer Aided Geometric Design.

Bibliography

[AB87a] S. Abhyankar and C. Bajaj. Automatic parameterization of rational curves and surfaces 1: Conics and conicoids. *Computer-Aided Design*, 19(1):11–14, 1987.

[AB87b] S. Abhyankar and C. Bajaj. Automatic parameterization of rational curves and surfaces II: Cubics and cubicoids. *Computer-Aided Design*, 19(9):499–502, 1987.

[AB87c] S. Abhyankar and C. Bajaj. Automatic rational parameterization of rational curves and surfaces IV: Algebraic space curves. Technical Report CSD-TR-703, Purdue University, Department of Computer Science, 1987.

[AB88] S. Abhyankar and C. Bajaj. Automatic parameterization of rational curves and surfaces III: Algebraic plane curves. *Computer Aided Geometric Design*, 5:309–321, 1988.

[Arn85] P. R. Arner. Hidden surface elimination. Master's thesis, University of Utah, Salt Lake City, Department of Mathematics, 1985.

[Baj90] C. L. Bajaj. Surface fitting using implicit algebraic surface patches. Technical Report CSD-TR-1001, Purdue University, Department of Computer Science, August 1990.

[Bar83] R. E. Barnhill. Computer aided surface representation and design. In R. E. Barnhill and W. Boehm, editors, *Surfaces in CAGD*. North-Holland, 1983.

[Bar91] R. E. Barnhill. Coons patches and convex combinations. Manuscript, 1991.

[BBB87] R. Bartels, J. Beatty, and B. Barsky. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann Publishers, Inc., 1987.

[BBG73] R. E. Barnhill, G. Birkhoff, and W. J. Gordon. Smooth interpolation in triangles. *Journal of Approximation Theory*, 8(2):114–128, 1973.

[BDH+88] C. Bajaj, W. Dyksen, C. Hoffmann, E. Houstis, J. Korb, and J. Rive. Computing about physical objects. Proceedings of the 12th IMACS WORLD Congress, 1988.

[BFK84] W. Boehm, G. Farin, and J. Kahmann. A survey of curve and surface methods in CAGD. *Computer Aided Geometric Design*, 1(1):1–60, 1984.

[BG91] P. Barry and R. Goldman. Algorithms for progressive curves: Extending B-spline and blossoming techniques to the monomial, power, and Newton bases. Manuscript, 1991.

[BI92] C. L. Bajaj and I. Ihm. Smoothing polyhedra using implicit algebraic splines. *Computer Graphics*, 26(2):79–88, July 1992.

[BIW90] C. L. Bajaj, I. Ihm, and J. Warren. Higher order fitting of algebraic surfaces. Technical Report CSD-TR-944, Purdue University Department of Computer Sciences, West Lafayette, Indiana, January 1990.

[Bla54] W. Blaschke. *Projektive Geometrie*. Birkhaesuer, 1954.

[Bli82] J. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3):235–256, 1982.

[Blo88] J. Bloomenthal. Polygonalization of implicit surfaces. *Computer Aided Geometric Design*, 5:341–355, 1988.

[Boe89] W. Boehm. Bézier patches on quadrics. Presented in *SIAM Conference on Geometric Design*, November 1989.

[Boe90] W. Boehm. On cyclides in geometric modeling. *Computer Aided Geometric Design*, 7:243–255, 1990.

[BP89] L. Bardis and N. Patrikalakis. Approximate conversion of rational B-spline patches. *Computer Aided Geometric Design*, 6:189–204, 1989.

[Bro91] J. L. Brown. Vertex based data dependent triangulations. *Computer Aided Geometric Design*, 8:239–251, 1991.

[BS84] R. E. Barnhill and S. E. Stead. Multistage trivariate surfaces. *Rocky Mountain Journal of Mathematics*, 14:103–118, 1984.

[Buc85] B. Buchberger. Gröbner bases: an algorithmic method in polynomial ideal theory. In N. Bose, editor, *Recent Trends in Multidimensional Systems Theory*, chapter 6. D. Reidel Publishing Company, 1985.

[BW84] R. E. Barnhill and A. J. Worsey. Smooth interpolation over hypercubes. *Computer Aided Geometric Design*, 1:101–113, 1984.

[BX92] C. L. Bajaj and G. Xu. Piecewise rational approximations of real algebraic curves. Technical report, Purdue University, Department of Computer Science, 1992.

[CG84] P. Charrot and J. A. Gregory. A pentagonal surface patch for computer aided geomteric design. *Computer Aided Geometric Design*, 1:87–94, 1984.

[CH89] J. H. Chuang and C. M. Hoffmann. On local implicit approximation and its applications. *ACM Transactions on Graphics*, 8(4):298–324, October 1989.

[Chi90] Eng-Wee Chionh. *Base points, resultants, and the implicit representation of rational surfaces*. PhD thesis, University of Waterloo, Canada, 1990.

[CK83] H. Chiyokura and F. Kimura. Design of solids with free-form surfaces. *Computer Graphics*, 17(3):289–298, 1983.

[Cob88] J. Cobb. Tiling the sphere with rational Bézier patches. Technical Report UUCS–88–009, University of Utah, July 1988.

[Coo45] J. Coolidge. *A History of the Conic Sections and Quadric Surfaces*. Clarendon Press, Oxford, 1945.

[Coo64] S. A. Coons. Surfaces for computer aided design. Technical report, MIT, Department of Mechanical Engineering, 1964. Revised 1967, available as AD 663 504 from the National Technical Information Service, Springfield, VA 22161.

[Cox74] H. S. M. Coxeter. *Projective Geometry.* University of Toronto Press, 1974.

[Dah89] W. A. Dahmen. Smooth piecewise quadric surfaces. In T. Lyche and Larry Schumaker, editors, *Mathematical Methods in CAGD*, pages 181–193. 1989.

[Dav63] P. J. Davis. *Interpolation and Approximation.* Blaisdell Publishing Company, 1963.

[dB87] C. de Boor. B-form basics. In G. Farin, editor, *Geometric Modeling: Algorithms and New Trends*, pages 131–148. SIAM, 1987.

[DeR88] T. DeRose. Composing Bézier simplexes. *ACM Transactions on Graphics*, 7(3):198–221, July 1988.

[DGHM92] T. DeRose, R.N. Goldman, H. Hagen, and S. Mann. Composition algorithms via blossoming: Theory, applications and implementation. *ACM Transactions on Graphics*, 1992. to appear.

[DLR90] N. Dyn, D. Levin, and S. Rappa. Data dependent triangulations for piecewise linear interpolant. *IMA Journal of Numerical Analysis*, 10:137–154, 1990.

[dMT84] Y. de Montaudouin and W. Tiller. The cayley method in CAGD. *Computer Aided Geometric Design*, 1:309–326, 1984.

[DN86] Dannenberg and Nowacki. Approximate conversion of surface representations with polynomial bases. In R. E. Barnhill and W. Boehm, editors, *Surfaces in CAGD.* North-Holland, 1986.

[Du92] H. Du. *New resolvent methods with applications to curves and surfaces in geometric modeling.* PhD thesis, University of Waterloo, Canada, Department of Computer Science, 1992. private communication.

[Far83] G. Farin. Algorithms for rational Bézier curves. *Computer-Aided Design*, 15(2):73–77, 1983.

[Far86] G. Farin. Triangular Bernstein-Bézier patches. *Computer Aided Geometric Design*, 3(2):83–128, 1986.

[Far87] R. Farouki. Direct surface section evaluation. In G. E. Farin, editor, *Geometric Modeling:Algorithms and New Trends*, pages 319–334. siam, 1987.

[Far88] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Academic Press Inc., New York, 1988.

[Far91] G. Farin. Degree reduction fairing of cubic B-spline curves. Manuscript, 1991.

[Fle77] W. Fleming. *Functions of Several Variables*. Springer-Verlag, 1977.

[FN89] R. Farouki and C. A. Neff. Some analytic and algebraic properties of plane offset curves. Technical Report RC 14364, IBM, 1989.

[Fol91] T. A. Foley. The cubic side-vertex method as a modified cubic triangular Bézier patch. abstract, 1991.

[For68] A. R. Forrest. *Curves and Surfaces for Computer-aided Design*. PhD thesis, University of Cambridge, 1968.

[For72a] A. R. Forrest. Interactive interpolation and approximation by Bézier curves. *Computer Journal*, 15(1):71–79, 1972.

[For72b] A. R. Forrest. On Coons and other methods for the representation of curved surfaces. *Computer Graphics and Image Processing*, 1:341–359, 1972.

[FPW88] G. Farin, B. Piper, and A. J. Worsey. The octant of a sphere as a non-degenerate triangular Bézier patch. *Computer Aided Geometric Design*, 4(4):329–332, 1988.

[FRSW87] G. Farin, G. Rein, N. Sapidis, and A. J. Worsey. Fairing cubic B-spline curves. *Computer Aided Geometric Design*, 4:91–103, 1987.

[GC80] J. A. Gregory and P. Charrot. A C^1 triangular interpolation patch for computer aided geomteric design. *Computer Graphics and Image Processing*, 13:80–87, 1980.

[GH87] J. A. Gregory and J. M. Hahn. Geometric continuity and convex combination patches. *Computer Aided Geometric Design*, 4(1–2):79–89, 1987.

[GH89] J. A. Gregory and J. M. Hahn. A C^2 polygonal surface patch. *Computer Aided Geometric Design*, 6(1):69–75, 1989.

[Gol85] R. N. Goldman. The method of resolvents: A technique for the implicitization, inversion and intersection of non-planar, parametric, rational cubic curves. *Computer Aided Geometric Design*, 2(4):237–255, December 1985.

[Gor69] W. J. Gordon. Distributive lattices and the approximation of multivariate functions. In I. J. Schoenberg, editor, *Approximations with Special Emphasis on Splines.* University of Wisconsin Press, Madison, 1969.

[Gre74] J. A. Gregory. Smooth interpolation without twist constraints. In R. E. Barnhill and R. F .Riesenfeld, editors, *Computer Aided Geometric Design*, pages 71–88. Academic Press, New York, 1974.

[Gre83a] J. A. Gregory. C^1 rectangular and non-rectangular surface patches. In R. E. Barnhill and W. Boehm, editors, *Surfaces in CAGD*, pages 25–33. North-Holland, 1983.

[Gre83b] J. A. Gregory. n-sided surface patches. In J. A. Gregory, editor, *Mathematics of Surfaces*, pages 217–232. Clarendon Press, Oxford, 1983.

[Gre85] J. A. Gregory. Interpolation to boundary data on the simplex. *Computer Aided Geometric Design*, 2:43–52, 1985.

[GSA84] R. N. Goldman, T. W. Sederberg, and D. C. Anderson. A technique for the implicitization, inversion and intersection of planar parametric rational polynomial curves. *Computer Aided Geometric Design*, 1(4):327–356, December 1984.

[Guo90] B. Guo. Automatic surface generation using implicit cubics. Manuscript, 1990.

[GW89a] T. Garrity and J. Warren. Geometric continuity. Technical Report TR-89-89, Rice University, Department of Computer Science, 1989.

[GW89b] T. Garrity and J. Warren. On computing the intersection of a pair of algebraic surfaces. *Computer Aided Geometric Design*, 6(2), 1989.

[Han83] P. Hanrahan. Ray tracing algebraic surfaces. *Computer Graphics*, 17(3):83–90, 1983.

[Har77] R. Hartshorne. *Algebraic Geometry*. Springer-Verlag, 1977.

[Her79] G. Herron. *Triangular and Multisided Patch Schemes*. PhD thesis, University of Utah, Salt Lake City, Department of Mathematics, 1979.

[Her89] G. Herron. Surface patches with low algebraic degree. Presented in *SIAM Conference on Geometric Design*, November 1989.

[HH86] C. Hoffmann and J. Hopcroft. Quadratic blending surfaces. *Computer-Aided Design*, 18:301–307, 1986.

[HH87] C. Hoffmann and J. Hopcroft. The potential method for blending surfaces and corners. In G. Farin, editor, *Geometric Modeling: Algorithms and New Trends*. SIAM, 1987.

[Hil86] D. Hildebrand. Rationale polynomialkurven und-flächen 2. grades für CAD. Master's thesis, Technische Hochschule Darmstadt, 1986.

[HK84] M. Hosaka and F. Kimura. Non-four-sided patch expressions with control points. *Computer Aided Geometric Design*, 1(1):75–86, 1984.

[Hoe83] J. E. Hoelzle. Knot placement for piecewise polynomial approximation of curves. *Computer-Aided Design*, 15(5):295–296, 1983.

[Hof89] C. M. Hoffmann. *Geometric and Solid Modeling: An Introduction*. Morgan Kaufmann Publishers, 1989.

[Hof90] C. M. Hoffmann. Algebraic and numeric techniques for offsets and blends. In M. Gasca W. Dahmen and C. Micchelli, editors, *Computation of Curves and Surfaces*, pages 499–529. Kluwer academic Publishers, 1990.

[Hos87] J. Hoschek. Approximate conversion of spline curves. *Computer Aided Geometric Design*, 4:59–66, 1987.

[Lac88] M. A. Lachance. Chebyshev economization for parametric surfaces. *Computer Aided Geometric Design*, 5:195–208, 1988.

[LD89] C. Loop and T. DeRose. A multisided generalization of Bézier surfaces. *ACM Transactions on Graphics*, 8(3), 1989.

[LD90] C. Loop and T. DeRose. Generalized B-spline surfaces of arbitrary topology. *Computer Graphics*, 24(4):347–356, 1990.

[Lee87] E. Lee. The rational Bézier representation for conics. In G. Farin, editor, *Geometric Modeling: Algorithms and New Trends*, pages 3–19. SIAM, 1987.

[LH74] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems.* Prentice-Hall Series in Automatic Computation. Prentice-Hall, 1974.

[Lit83] F. F. Little. Convex combination surfaces. In R. E. Barnhill and W. Boehm, editors, *Surfaces in CAGD.* North-Holland, 1983.

[Max63] E. Maxwell. *The Methods of Plane Projective Geometry Based on the Use of General Homogeneous Coordinates.* Cambridge University Press, 1963.

[MC92] D. Manocha and J. F. Canny. Algorithm for implicitizing rational parametric surfaces. *Computer Aided Geometric Design*, 9(1):25–50, May 1992.

[MW90] D. Moore and J. Warren. Adaptive approximation of scattered contour data using piecewise implicit surfaces. Proceedings of the Hawaii International Conference on System Sciences, 1990.

[Nie79] G. M. Nielson. The side-vertex method for interpolation in triangles. *Journal of Approximation Theory*, 25:318–336, 1979.

[Nie87] G. M. Nielson. A transinite visually continuous triangular interpolant. In G. Farin, editor, *Geometric Modeling: Algorithms and New Trends*, pages 235–246. Siam, 1987.

[Pat89] N. M. Patrikalakis. Approximate conversion of rational splines. *Computer Aided Geometric Design*, 6:155–165, 1989.

[Pet84] C. S. Petersen. Adaptive contouring of three-dimensional surfaces. *Computer Aided Geometric Design*, 1:61–74, 1984.

[Pet90] J. Peters. *Fitting smooth parametric surfaces to 3D data.* PhD thesis, University of Wisconsin, Madison, Center for the Mathematical Sciences, 1990.

[PG85] M. J. Pratt and A. Geisow. Surface/surface intersection problems. In J. A. Gregory, editor, *The Mathematics of Surfaces*, pages 117–142. Oxford University Press, 1985.

[Pie86] L. Piegl. The sphere as a rational Bézier surface. *Computer Aided Geometric Design*, 3(1):45–52, 1986.

[PK89] 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.

[PPW87] C. Petersen, B. Piper, and A. J. Worsey. Adaptive contouring of a trivariate interpolant. In G. Farin, editor, *Geometric Modeling: Algorithms and New Trends*. SIAM, 1987.

[Pra87] V. Pratt. Direct least-squares fitting of algebraic surfaces. *Computer Graphics*, 21(4):145–152, 1987.

[Pra90] M. J. Pratt. Cyclides in CAGD. *Computer Aided Geometric Design*, 7:221–242, 1990.

[PS77] M. J. D. Powell and M. A. Sabin. Piecewise quadratic approximations on triangles. *ACM Transactions on Mathematical Software*, 3(4):316–325, 1977.

[PT87] L. Piegl and W. Tiller. Curve and surface constructions using rational B-splines. *Computer-Aided Design*, 19(9):485–498, 1987.

[RV83] A. A. G. Requicha and H. B. Voelcker. Solid modeling: Current status and research directions. *IEEE Computer Graphics and Applications*, pages 25–37, October 1983.

[SA85] T. W. Sederberg and D. C. Anderson. Steiner surface patches. *IEEE Computer Graphics and Applications*, 5(5):23–36, May 1985.

[Sab83] M. Sabin. Non-rectangular surface patches suitable for inclusion in a B-spline surface. In P. ten Hagen, editor, *Proceedings of Eurographics 1983*, pages 57–69, Amsterdam, 1983. North-Holland.

[SAG84] T. W. Sederberg, D. C. Anderson, and R. Goldman. Implicit representation of parametric curves and surfaces. *Computer Vision Graphics and Image Processing*, 28:72–74, 1984.

[Sed83] T. W. Sederberg. *Implicit and parametric curves and surfaces for computer aided geometric design*. PhD thesis, Purdue University, Department of Computer Science, 1983.

[Sed84] T. W. Sederberg. Planar piecewise algebraic curves. *Computer Aided Geometric Design*, 1(3):241–255, December 1984.

[Sed85] T. W. Sederberg. Piecewise algebraic surface patches. *Computer Aided Geometric Design*, 2:53–59, 1985.

[Sed89] T. W. Sederberg. Scan line display of algebraic surfaces. *Computer Graphics*, 23(3):147–156, 1989.

[Sed90a] T. W. Sederberg. Techniques for cubic algebraic surfaces. *IEEE Computer Graphics and Applications*, 10(3):14–25, July 1990.

[Sed90b] T. W. Sederberg. Techniques for cubic algebraic surfaces. *IEEE Computer Graphics and Applications*, 10(5):12–21, September 1990.

[SP86] T. W. Sederberg and S. Parry. Comparison of three curve intersection algorithms. *Computer-Aided Design*, 18(1):58–63, 1986.

[SR49] J. G. Semple and L. Roth. *Introduction to Algebraic Geometry*. Clarendon Press, Oxford, 1949.

[SS87] T. W. Sederberg and J. Snively. Parametrization of cubic algebraic surfaces. In R. R. Martin, editor, *Mathematics of Surfaces II*, pages 299–319. Clarendon Press, Oxford, 1987.

[Ste73] G. W. Stewart. *Introduction to Matrix Computations*. Academic Press, 1973.

[SZZ88] T. W. Sederberg, J. Zhao, and A. K. Zundel. Approximate parametrization of algebraic curves. Technical Report ECGL-88104, Brigham Young University, 1988.

[UH91] K. Ueda and T. Harada. Generalization of a family of gregory surfaces. In N. M. Patrikalakis, editor, *Volume Visualization*, pages 417–434. 1991.

[Var87] T. Varady. Survey and new results in n-sided patch generation. In R. Martin, editor, *The Mathematics of Surfaces II*, pages 203–236. Oxford University Press, 1987.

[Var91] T. Varady. Overlap patches: A new scheme for interpolating curve networks with n-sided regions. *Computer Aided Geometric Design*, 8:7–27, 1991.

[VCH88] D. Dutta V. Chandru and C. M. Hoffmann. On the geometry of dupin cyclides. Technical Report CSD-TR-818, Purdue University, November 1988.

[VY45] O. Veblen and J. Young. *Projective Geometry*. Blaisdell Publishing, New York, 1945.

[WA89] W. N. Waggenspack and D. C. Anderson. Piecewise parametric approximations for algebraic curves. *Computer Aided Geometric Design*, 6:33–53, 1989.

[War86] J. Warren. *On Algebraic Surfaces Meeting with Geometric Continuity*. PhD thesis, Cornell University, Department of Computer Science, 1986.

[War87] J. Warren. Blending quadric surfaces with quadric and cubic surfaces. In *Third ACM Symposium on Computational Geometry*, pages 341–347. ACM, 1987.

[War89] J. Warren. Blending algebraic surfaces. *ACM Transactions on Graphics*, 8(4):263–278, October 1989.

[War90] J. Warren. The effect of base points on rational Bézier surfaces. Technical Report TR 90-122, Rice University, Department of Computer Science, May 1990.

[War91] J. Warren. A bound on the degree of polygonal Bézier surfaces. In C. Bajaj, editor, *Algebraic Geometry and Applications*. Springer Verlag, 1991.

[War92] J. Warren. Barycentric coordinates for convex polytopes. private communication, 1992.

[War91] J. Warren. A subdivision method for algebraic surfaces. presented in *Siam Conference on Geometric Design*, November 91.

[WMW86] G. Wyvill, C. McPheeters, and B. Wyvill. Data structure for *soft* objects. *The Visual Computer*, 2:227–234, 1986.

[Wor84] A. J. Worsey. A modified C^2 Coons' patch. *Computer Aided Geometric Design*, 1:357–360, 1984.

[Wor85] A. J. Worsey. C^2 interpolation over hypercubes. In R. E. Barnhill and W. Boehm, editors, *Surfaces in CAGD '84*. North-Holland, 1985.

[WW88] M. A. Watkins and A. J. Worsey. Degree reduction of Bézier curves. *Computer-Aided Design*, 22(7):398–405, September 1988.