Automatic Reconstruction of B-Spline Surfaces of Arbitrary Topological Type

Matthias Eck
University of Darmstadt

Hugues Hoppe
Microsoft Corporation

Submitted to SIGGRAPH 96.
(Also available as Technical Report MSR-TR-96-01, Microsoft Corporation, January 1996.)

Abstract

Even with advanced geometric modeling systems, creating freeform surfaces is a challenging task. Laser range scanners offer a promising alternative for model acquisition—the 3D scanning of existing objects or clay maquettes. The problem of converting the dense point sets produced by laser scanners into useful geometric models is referred to as surface reconstruction.

In this paper, we present a procedure for reconstructing a tensor product B-spline surface from a set of scanned 3D points. Unlike previous work which considers primarily the problem of fitting a single B-spline patch, our goal is to directly reconstruct a surface of arbitrary topological type. We must therefore define the surface as a network of B-spline patches. A key ingredient in our solution is a scheme for automatically constructing both a network of patches and a parametrization of the data points over these patches. In addition, we define the B-spline surface using a *surface spline* construction, and demonstrate that such an approach leads to an efficient procedure for fitting the surface while maintaining tangent plane continuity. We explore adaptive refinement of the patch network in order to satisfy user-specified error tolerances, and demonstrate our method on both synthetic and real data.

1 Introduction

In the fields of computer graphics and computer-aided design (CAD), advanced modeling systems such as Softimage 3D, Alias/Wavefront, CATIA, and PRO/ENGINEER have made possible the design of highly detailed models. Even so, it is still difficult with these systems to directly create organic shapes such as human faces and freeform surfaces such as car-body panels.

The advent of laser range scanners offers an alternative means of acquiring geometric models: the 3D scanning of existing objects. With 3D scanning, modeling systems can import organic or sculptured shapes that would otherwise be difficult to create. For instance, in the automobile industries, many artists prefer to initially sculpt car bodies in clay, as they find that CAD systems lack the tactile and visual advantages of the traditional medium. Similarly, many models used in computer graphics are first created in clay or wood and subsequently scanned into digital forms. In addition, 3D scanning permits *reverse engineering*, allowing existing manufactured parts to be incorporated or modified into new CAD designs.

Laser range scanners produce large collections of points on surfaces of objects. The problem of converting these data points into useful geometric models is referred to as *surface reconstruction*. There is a large body of literature on the reconstruction of surfaces of simple topological type, such as deformed planar regions and spheres (see Section 2). Methods have been developed to reconstruct meshes of arbitrary topological type [12, 31], but the resulting representations are often verbose since many planar faces are required to accurately model curved surfaces (e.g. Figure 9l). For this reason, it is desirable to use a representation with smooth surface primitives. Some recent work addresses the problem of reconstructing smooth surfaces of arbitrary topological type using subdivision surfaces [10] and algebraic surfaces [2, 21]. However, these two smooth surface representations are not commonly supported within current modeling systems. Indeed, for better or worse, the ubiquitous smooth surface primitive is the tensor product B-spline patch. The general class of non-uniform rational B-splines (NURBS) is considered by many the de facto CAD standard.

In this paper we present a procedure for automatically reconstructing a B-spline surface S of arbitrary topological type from an unorganized set of points $P = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$. To our knowledge, this reconstruction problem has not been addressed previously. The problem presents two major difficulties:

- Since a single B-spline patch can only represent surfaces of simple topological type (deformed planar regions, cylinders, and tori), a surface of arbitrary topological type must be defined as a network of B-spline patches. Automatically constructing both a network of patches and a parametrization of the data points over these patches is a difficult task.
- The reconstructed B-spline patch network is often expected to be smooth (by which we mean tangent plane continuous or G^1). Enforcing G^1 continuity between adjacent patches while at the same time fitting the patch network to the points is a challenging problem.

Our B-spline reconstruction procedure adapts the previous surface reconstruction work of Hoppe et al. [11, 12], the parametrization work of Eck et al. [5], and the B-spline construction scheme of Peters [23], as summarized in Sections 3.1, 3.2, and 3.4 respectively. The main contributions of this paper are:

- It presents a combinatorial optimization method for building a quadrilateral domain from a triangular one (Section 3.3), a crucial step in constructing the B-spline patch network. The optimization method makes use of *harmonic maps* to minimize distortion in the resulting reparametrization.
- It presents an efficient method for fitting a G^1 B-spline surface of arbitrary topological type to unorganized points. The fitting method makes use of a surface spline construction to maintain G^1 continuity between patches. As a consequence, fitting the surface to the data involves only a sparse linear least squares problem with a few linear constraints.
- It introduces a scheme for adaptive refinement of the quadrilateral patch network, and demonstrates the use of this refinement strategy in attempting to fit B-spline surfaces within user-specified error tolerances.
- It demonstrates how all these techniques can be brought together into an effective procedure addressing an important problem in computer graphics and geometric modeling: automatic reconstruction of B-spline surfaces of arbitrary topological type.

In addition to surface reconstruction, our procedure can also be applied to the problem of surface approximation. That is, it can be used to approximate an arbitrary initial surface S_0 with a B-spline surface (e.g. Figures 10j–10l) as shown in Section 4.

2 Related Work

Reconstruction of B-spline surfaces There has been considerable work on fitting B-spline surfaces to 3D points. However, most methods either assume that the surface has simple topological type, or require user intervention in setting up the patch network.

For instance, Dietz [3], Hoschek and Schneider [14], Rogers and Fog [26], and Sarkar and Menq [27] assume that the surface is a single open B-spline patch (a deformed quadrilateral region), possibly with trimmed boundaries. Forsey and Bartels [8] consider fitting a single hierarchical B-spline patch to gridded data. Schmitt et al. [28] assume that the surface is a deformed cylinder and explore adaptive refinement of the B-spline surface in fitting cylindrical range data.

Andersson et al. [1], Fang and Gossard [6], and Milroy et al. [20] fit B-spline surfaces of arbitrary topological type, but require the user to manually delineate the patch boundaries either by labeling "boundary points" or by drawing boundary curves on an approximating surface. Furthermore, the initial parametrizations of the data points is critical in the fitting process, as demonstrated by Ma and Kruth [19], and these schemes may require additional user intervention to obtain good initial parameter distributions.

In contrast, our method is able to reconstruct a B-spline surface of arbitrary topological type without user assistance. One of the key steps is the automatic construction of a low-distortion initial parametrization of the data points. To our knowledge this has not been done before. Moreover, the surface consists of a network of low-degree, tensor-product B-spline patches that meet with G^1 continuity.

Reconstruction of other surface representations Hoppe et al. [10] reconstruct piecewise smooth surfaces of arbitrary topological type using a subdivision surface representation. As part of the fitting process, they optimize both the connectivity and geometry of the control mesh defining the subdivision surface.

Both Bajaj et al. [2] and Moore and Warren [21] reconstruct G^1 piecewise algebraic surfaces of arbitrary topological type. Their surfaces are defined as algebraic patches within 3D tetrahedral triangulations of $\mathbf{R}^3$. They consider adaptive refinement of the 3D triangulation based on the quality of fit.

Turk and Levoy [31] reconstruct dense meshes by "zippering" together multiple overlapping range images. The dense meshes they produce are most appropriate for capturing models with fine geometric detail.

3 Algorithm

Our B-spline surface reconstruction algorithm consists of 5 successive steps. We first present a brief overview of these steps and illustrate them with the example in Figure 9. Sections 3.1–3.5 describe the details of the 5 steps.

1. *Constructing an initial parametrization over a dense approximating mesh M_0:*

   Using the previous surface reconstruction work of Hoppe et al. [11, 12], Step 1 constructs from an unorganized set of points $P = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$ (Figure 9a) an initial surface approximation in the form of a dense triangular mesh M_0 (Figure 9b). The points P are projected onto M_0 to obtain their initial parametrizations. Our purpose in constructing M_0 is to find a parametric domain of the correct topological type. Of course, this particular domain is unwieldy since it may consist of thousands of faces.

2. *Reparametrizing over a simple triangular base complex $K_\triangle$:*

   Using the parametrization work of Eck et al. [5], Step 2 automatically constructs from the initial mesh M_0 both a simple *base complex* $K_\triangle$ (Figure 9e) and a continuous parametrization $\rho_\triangle : K_\triangle \to M_0$. As the construction exploits the mathematical framework of harmonic maps, the parametrization $\rho_\triangle$ tends to have low metric distortion. The parametrization of P from Step 1 are mapped through $\rho_\triangle^{-1}$ to obtain new parametrizations over $K_\triangle$.

3. *Reparametrizing over a quadrilateral domain complex $K_\square$:*

   By merging faces of $K_\triangle$ pairwise, Step 3 constructs a new base complex $K_\square$ whose faces consist solely of quadrilaterals (Figure 9f). The merging process is cast as a combinatorial graph optimization problem, whose goals are both to find a maximum pairing and to minimize parametric distortion. We again make use of harmonic maps to find a good reparametrization of the points P from $K_\triangle$ to $K_\square$.

4. *B-spline fitting:*

   Step 4 defines over each face f of $K_\square$ a tensor product B-spline patch $\mathbf{s}_f$ using the surface splines scheme of Peters [23] such that the patches $\mathbf{s}_f$ collectively form a G^1 B-spline surface S. More precisely, this construction consists of two steps. First, a control mesh M_x is defined by topologically subdividing $K_\square$. Second, the control points $\mathbf{d}^f_{r,s}$ of $\mathbf{s}_f$ are defined as affine combinations of the vertices V_x of M_x. Fitting S to the points P is cast as an optimization problem over V_x, and is solved by alternating between a linear least squares fitting step and a parameter correction step. The result of this fitting process is shown in Figures 9g–9i.

5. *Adaptive refinement:*

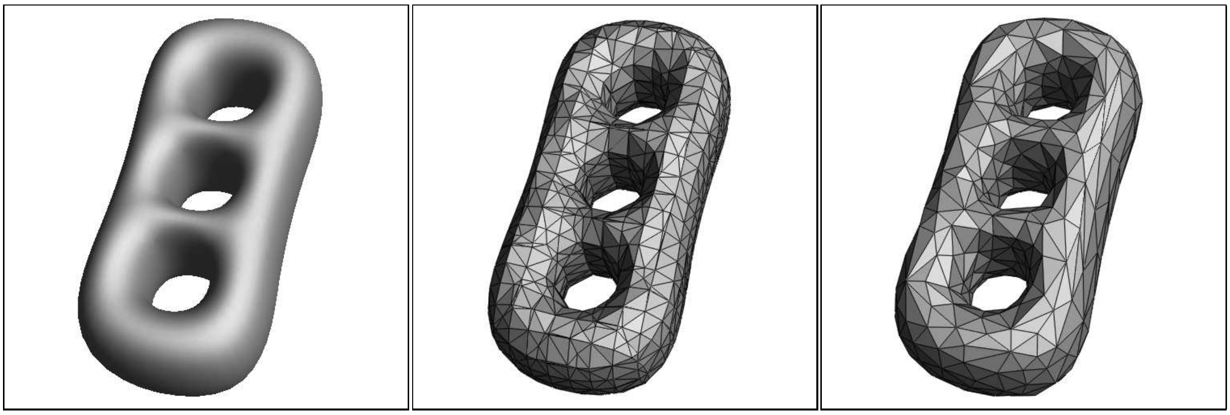
   In order for P and S to differ by no more than a user-specified error tolerance ϵ, Step 5 adaptively subdivides the faces of $K_\square$ into smaller quadrilateral subfaces based on the fit errors. After each step of subdivision, Step 4 is reinvoked to fit the refined surface. Further subdivisions are performed until the error tolerance ϵ is satisfied. The result is a refined domain complex $K_\boxplus$ (Figure 9j) and a new control mesh (Figure 9k) defining a new B-spline surface S (Figure 9l) within ϵ of P.

3.1 Constructing an initial parametrization over a dense approximating mesh M_0

From an unorganized set of points P, Step 1 constructs an initial surface approximation M_0 and parametrizes the points over this initial domain. This step is performed using the surface reconstruction method of Hoppe et al., which we briefly summarize now.

Summary of surface reconstruction method of Hoppe et al. Hoppe et al. [11, 12] develop a two-phase procedure for reconstructing a mesh (Figure 1c) approximating an unknown surface S_u (Figure 1a) from a set of unorganized points P (Figure 9a) sampled on or near S_u.

The goal of phase one [11] is to determine the topological type of S_u and to obtain a crude estimate of its geometry, in the form of a dense mesh (Figure 1b). Using P, phase one defines a function $f : \mathbf{R}^3 \to \mathbf{R}$ that estimates the signed geometric distance to S_u, and then uses a contouring algorithm to extract a mesh approximating its zero set, $Z(f) = \{\mathbf{q} \in \mathbf{R}^3 : f(\mathbf{q}) = 0\}$.



(a) Unknown surface S_u (b) Phase one mesh (c) Phase two mesh

Figure 1: Example of the two-phase surface reconstruction method of Hoppe et al.. (Refer also to Figures 9a and 9b.)

The goal of phase two [12] is to reduce the number of faces in the mesh and to improve its fit to the data (Figure 1c). Phase two optimizes over both the connectivity and geometry of the mesh in order to minimize an energy function that explicitly models the trade-offs of conciseness and accuracy.

Our use of the surface reconstruction method For our purpose, we first run phase one to obtain a crude mesh (Figure 1b). We then use the initial fitting procedure of phase two to improve the geometry of this mesh while keeping its connectivity constant, to obtain the mesh M_0 (Figure 9b). The optimization over connectivity performed later in phase two is unnecessary for our use, since Step 2 (described in the next section) provides a faster algorithm for creating a simpler domain and at the same time constructs a low-distortion parametrization of P over that domain.

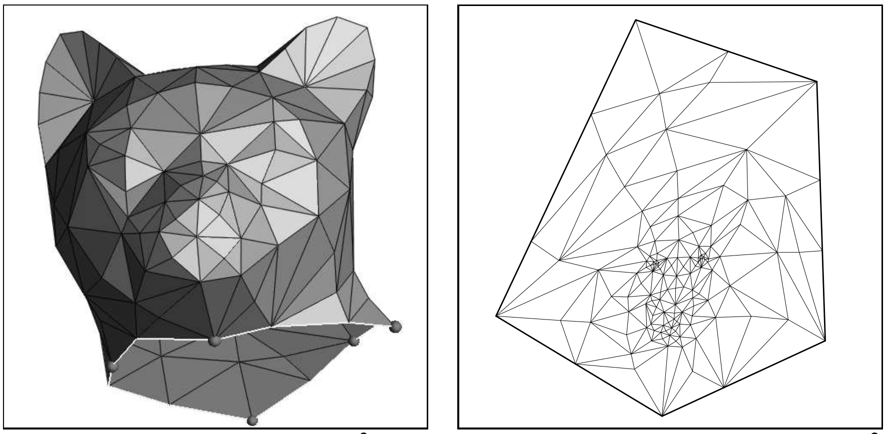
To obtain an initial parametrization of P, we project the points onto the mesh M_0. For each point $\mathbf{p}_i$, we store the closest face of M_0 and the barycentric coordinates of the projection of $\mathbf{p}_i$ onto that face.

3.2 Reparametrizing over a simple triangular base complex $K_\triangle$

From the initial mesh M_0, Step 2 constructs a simple base complex $K_\triangle$ (Figure 9e) and a map $\rho_\triangle : K_\triangle \to M_0$ allowing the points P to be reparametrized over $K_\triangle$. This step is achieved using the parametrization method of Eck et al. [5], which we briefly summarize. Next we present a minor modification to the method that facilitates the construction of $K_\square$ in Step 3.

Summary of parametrization method of Eck et al. Eck et al. describe a method for parametrizing an arbitrary mesh M_0 over a domain complex $K_\triangle$. Their method makes use of *harmonic maps*, which we describe first.

Eck et al. first introduce a method for mapping a (topological) disk $D \subset M$ of a mesh $M \subset \mathbf{R}^3$ to a convex polygonal region $R \subset \mathbf{R}^2$. As an example, the mesh region in Figure 2a is parametrized over the planar polygon in Figure 2b. Their solution, based on the theory of *harmonic maps*, has the property of minimizing metric distortion. They find that the metric distortion energy $E_{harm}[h]$ associated with a



(a) Original mesh disk $D \subset \mathbf{R}^3$ (b) Harmonic embedding in $R \subset \mathbf{R}^2$

Figure 2: An example of a harmonic map. (The vertices indicated by small balls are mapped to the vertices of the polygon.)

piecewise linear map $h : D \to R$ can be interpreted as the energy of a configuration of springs placed on the edges of D:

$$E_{harm}[h] = 1/2 \sum_{\{i,j\} \in \mathrm{Edges}(D)} \kappa_{i,j} \|h(i) - h(j)\|^2 ,$$

where each spring constant $\kappa_{i,j}$ is a simple function of the lengths of nearby edges in the original mesh D. Thus the (piecewise-linear) harmonic map h on D can be computed by solving a sparse linear least-squares problem.

Eck et al. next describe a method for partitioning a mesh M_0 into well-shaped triangular regions. This partitioning method is based on generalizing the concepts of Voronoi diagrams and Delaunay triangulations to surfaces of arbitrary topological type. The algorithm automatically selects a set of site faces in M_0 and partitions M_0 into a set of Voronoi tiles, such that each tile comprises those faces closest to a given site (Figure 9c). The Voronoi tiles are grown incrementally from their site faces using a multi-source shortest path algorithm. In order for the Voronoi-like partition to be dual to a triangulation, the sites are chosen to satisfy a set of 4 conditions (see [5]). Next, the method makes use of harmonic maps to construct a Delaunay-like triangular partition $T_1, \ldots, T_r$ (Figure 9d) that is dual to the Voronoi-like partition.

Finally, Eck et al. construct a base complex $K_\triangle$ of r faces (Figure 9e) using the connectivity of the Delaunay-like partition, and parametrize M_0 over this domain by computing the harmonic map from each Delaunay triangle T_i to the corresponding face of $K_\triangle$. The result is a global, continuous parametrization $\rho_\triangle : K_\triangle \to M_0$ of the initial mesh over a simple base complex.

Modification to the parametrization method In Step 3 described in the next section, we construct from $K_\triangle$ a new domain $K_\square$ with quadrilateral faces by matching adjacent pairs of faces in $K_\triangle$. To form a complete matching, the face merging process requires $K_\triangle$ to have an even number of faces. This requirement is met by giving the Voronoi partitioning algorithm an additional condition to satisfy: The dual to the Voronoi partition must have an even number of faces. When this condition is not satisfied, an additional site is added at the face farthest from any current site, and the Voronoi region growing algorithm restarts.

Reparametrization After constructing $K_\triangle$ and $\rho_\triangle$, we map the parametrizations of P obtained in Step 1 through $\rho_\triangle^{-1}$ to obtain parametrizations of P over $K_\triangle$. The new parametrization is illustrated in Figure 9e, where a line segment is drawn between each data point $\mathbf{p}_i$ and its parametric location on $K_\triangle$. Note that we do not define a geometric embedding of $K_\triangle$ into $\mathbf{R}^3$ but have created one in Figure 9e for illustration purposes only.

3.3 Reparametrizing over a quadrilateral domain complex $K_\square$

After Step 2, the points P are parametrized over a base complex $K_\triangle$ made up of an even number of triangular faces. Since the B-spline construction scheme in Step 4 expects a domain made up of quadrilateral faces, the goal of Step 3 is to map $K_\triangle$ onto a quadrilateral domain complex $K_\square$ (Figure 9f).

A simple strategy would be to subdivide each triangular face of $K_\triangle$ into 3 quadrilateral faces by introducing vertices at the edge midpoints and the face centroids. Instead, our method is based on merging triangle faces of $K_\triangle$ pairwise. This merging strategy is advantageous because it results in a domain $K_\square$ with one sixth the number of quadrilaterals as would be obtained from subdivision.

We cast face merging as a graph matching problem. We construct the graph $G = (V_G, E_G)$ that is the dual to $K_\triangle$: each vertex in V_G corresponds to a face of $K_\triangle$, and each edge in E_G corresponds to a pair of faces sharing an edge in $K_\triangle$. Finding a maximum pairing of faces in $K_\triangle$ then amounts to finding a maximum cardinality set $E_m \subset E_G$ of vertex-disjoint edges—an instance of the MAXIMUM MATCHING graph problem on G [15], which can be solved efficiently in $O(|V_G||E_G|)$ time [30].

We would like to obtain a *complete matching*: one in which all faces of $K_\triangle$ are paired. Since we have constructed $K_\triangle$ to have an even number of regions, a complete matching is likely to exist. Although counter-examples may be possible, we have not seen them occur in practice. (It would be interesting to prove if such counter-examples can or cannot exist.) If G was to lack a complete matching, we would resort to global subdivision as described above.

The graph G typically has many possible complete matchings. Of those, we would prefer one that minimizes the distortion of the resulting reparametrization. In order to achieve this, we define a heuristic for the distortion associated with the pairing of two adjacent faces F_i and F_j of $K_\triangle$ as follows. We construct the harmonic map $h_{i,j}$ of the region $T_i \cup T_j$ of M_0 onto a unit square, and use the resulting harmonic energy term $E_{harm}[h_{i,j}]$ as our heuristic measure of distortion. We encode these distortion measures into G by assigning to each edge $e = \{i, j\} \in E_G$ the weight $w(e) = -E_{harm}[h_{i,j}]$. The face merging problem is now cast as an instance of the MAX-MIN MATCHING problem—finding a maximum cardinality matching for which the minimum weight of the edges is maximum [15]. A solution to this combinatorial problem corresponds to a complete pairing of faces of $K_\triangle$ for which the maximal distortion of the face pairs is minimized. The MAX-MIN MATCHING problem can be solved in $O(|V_G|^3)$ time [15]. Since our graphs G typically have on the order of a hundred vertices, computing the matching requires only a few seconds.

Once the matching is computed, the parametrizations of the points P are mapped from $K_\triangle$ to $K_\square$ using the same harmonic maps constructed for the graph optimization problem. Specifically, for each edge $\{i, j\} \in E_m$ of the matching solution, we map the points whose parametrizations lie on faces F_i and F_j of $K_\triangle$ through $h_{i,j}$ onto the unit square, and use the resulting coordinates as (bilinear) parametrizations on the new face $F_{i,j}$ in $K_\square$. The parametrizations are illustrated by the line segments in Figure 9f. Again, we have created an embedding for $K_\square$ in $\mathbf{R}^3$ in the figure for illustration purposes only.

There is one final complication. The resulting $K_\square$ may have interior vertices of degree 2, and such vertices are best avoided for Step 4. When such vertices are present, we merge the two quadrilateral faces adjacent to them into larger quadrilateral faces.

3.4 B-spline fitting

3.4.1 General framework

In the most general setting, a B-spline surface $S(K_\square, \mathbf{d})$ is defined as a network of tensor product B-spline surface patches

$$\mathbf{s}_f(u, v) = \sum_{r=0}^{n_f} \sum_{s=0}^{m_f} \mathbf{d}^f_{r,s} N_{r,k_f}(u) N_{s,l_f}(v)$$

over a domain complex $K_\square$, with local coordinates $(u, v) \in [0, 1]^2$ on each face $f \in K_\square$. Here $\mathbf{d}^f_{r,s} \in \mathbf{R}^3$ denote the control points, $N_{r,k_f}(u)$ are the univariate B-spline basis functions of order k_f in the u-direction, defined over the knot sequences $U_f = (u_0, u_1, \ldots, u_{n_f+k_f})$, and $N_{s,l_f}(v)$ are defined analogously over the knot vectors V_f in the v-direction. Definitions of the B-spline basis functions and related evaluation algorithms can be found in textbooks on geometric modeling (e.g. [7, 13]).

Surface reconstruction In surface reconstruction we seek to find the control points $\mathbf{d}^f_{r,s}$ of all patches $\mathbf{s}_f$ such that the distance of the data points P to the surface $S(K_\square, \mathbf{d})$ is minimized. More precisely, we minimize the distance functional

$$E_{dist}(S) = \sum_{i=1}^{N} d^2(\mathbf{p}_i, S) \ .$$

Note that the distance of each point $\mathbf{p}_i$ to the surface S is itself the solution of a minimization problem:

$$d(\mathbf{p}_i, S) = \min_{t_i} \|\mathbf{p}_i - \mathbf{s}(t_i)\|^2 = \min_{f_i \in K_\square, (u_i, v_i) \in [0,1]^2} \|\mathbf{p}_i - \mathbf{s}_{f_i}(u_i, v_i)\|^2$$

in which $t_i = (f_i, u_i, v_i)$ is the parametrization of the projection of $\mathbf{p}_i$ onto S.

Iterative methods have been developed to solve this type of nested minimization problem in the context of B-spline surface fitting [14, 26]. In these methods, each iteration consists of two steps:

1. *Fitting step:* For fixed parametrizations t_i, the optimal control points $\mathbf{d}$ are found by solving a linear least-squares problem.

2. *Parameter correction step:* For fixed control points $\mathbf{d}$, optimal parametrizations t_i are found by projecting the points onto S.

Usually the fit accuracy is improved considerably after only a few iterations (we typically use 4). (An alternative solution method to this nonlinear problem is the Levenberg–Marquardt optimization method, which has faster convergence rate [27]; however, our simple iterative scheme is sufficient for obtaining reasonable fits.)

Fairness functional One problem with surface fitting is that the resulting surface may have unwanted "wiggles". It is therefore common to augment the energy functional with an additional fairness term [3, 6]:

$$E(\mathbf{d}) = E_{dist}(\mathbf{d}) + \lambda \cdot E_{fair}(\mathbf{d}) \ , \quad \lambda \in \mathbf{R}^+_0 \ . \tag{1}$$

The fairness term is often defined to be the *thin plate energy* functional

$$E_{fair}(\mathbf{d}) = \sum_{f \in K_\square} \int_0^1 \int_0^1 \left(\frac{\partial^2}{\partial u^2} \mathbf{s}_f(u,v) + 2 \cdot \frac{\partial^2}{\partial u\, \partial v} \mathbf{s}_f(u,v) + \frac{\partial^2}{\partial v^2} \mathbf{s}_f(u,v) \right) du\, dv \ .$$

(Greiner [9] also discusses alternative functionals involving higher-order derivatives.) Note that $E(\mathbf{d})$ can still be minimized with the iterative scheme described previously since $E_{fair}(\mathbf{d})$ is independent of the parameter values t_i and its minimization still gives rise to a linear system.

There remains the problem of finding a reasonable choice for the fairness weight λ. Dietz [3] suggests starting with a relatively large initial weight λ and reducing λ by a factor of 2 after each iteration of parameter correction. In our case, the initial parametrizations obtained in Step 3 are quite good, and we have obtained satisfactory results using simply a small, constant λ.

Continuity Obviously, constraints must be established between adjacent B-spline patches so that they join up seamlessly. To simplify these constraints, most schemes (e.g. [20]) set all patches to have the same knot vectors (i.e. $n = n_f = m_f$ and $U = U_f = V_f$) and the same order $k = k_f = l_f$. Then, simple (G^0) continuity is achieved trivially by sharing control points along the boundaries of adjacent patches. In contrast, tangent plane (G^1) continuity is more difficult since it involves nonlinear constraints on the control points of adjacent patches. There are two main approaches to satisfying these G^1 continuity constraints.

In the first approach [20, 22], the nonlinear G^1 constraints are approximated by introducing an additional penalty term $E_{G^1}(\mathbf{d})$ to minimize. Unfortunately, minimizing $E_{dist}(\mathbf{d}) + E_{fair}(\mathbf{d}) + E_{G^1}(\mathbf{d})$ requires more expensive nonlinear optimization. Moreover, the resulting surface is only approximately tangent-plane smooth, or ϵ-G^1, and the lack of smoothness is often visible in the resulting surfaces (e.g. [20]).

In the second approach, often referred to as *surface splines* or *G-splines* [17, 18, 23, 24, 25], the idea is to construct a network of triangular and/or tensor product Bézier patches from a global control mesh M_x. The control points of these Bézier patches are computed using local combinations of vertices in M_x, and are defined in such a way that the Bézier patches automatically meet with G^1 continuity. Using this approach, the surface is exactly G^1, and the fitting process again involves solving a simple linear system, in which the unknowns are the vertex positions of M_x.

We have opted for the second approach, and have adapted a surface spline scheme of Peters [23]. As described in the next section, we construct over each face of $K_\square$ a single tensor product B-spline patch $\mathbf{s}_f$ with $k = 4$ and $n = 11$. To overcome the problem of fixed n and k, we present in Section 3.5 a refinement scheme that adaptively subdivides $K_\square$ to locally introduce additional degrees of freedom.

3.4.2 B-spline fitting using construction of Peters

Review of Peters' scheme From a closed mesh M_c of arbitrary topological type, the construction scheme of Peters [23] creates a G^1 B-spline surface S. This construction proceeds in two steps, as illustrated in Figure 3. First, a refined control mesh M_x is created by subdividing M_c using two Doo-Sabin subdivisions [4]. In our application M_c has only quadrilateral faces, and therefore a 4×4 grid of vertices in M_x is associated with each face of M_c as shown in Figure 3b. Note that all vertices of M_x have valence 4 (i.e. 4 adjacent edges) and that M_x consists mainly of 4-sided faces, except for a small number of extraordinary m-sided faces ($m \neq 4$). Also note that the extraordinary faces are isolated, in the sense that each vertex of M_x is adjacent to at most one extraordinary face.

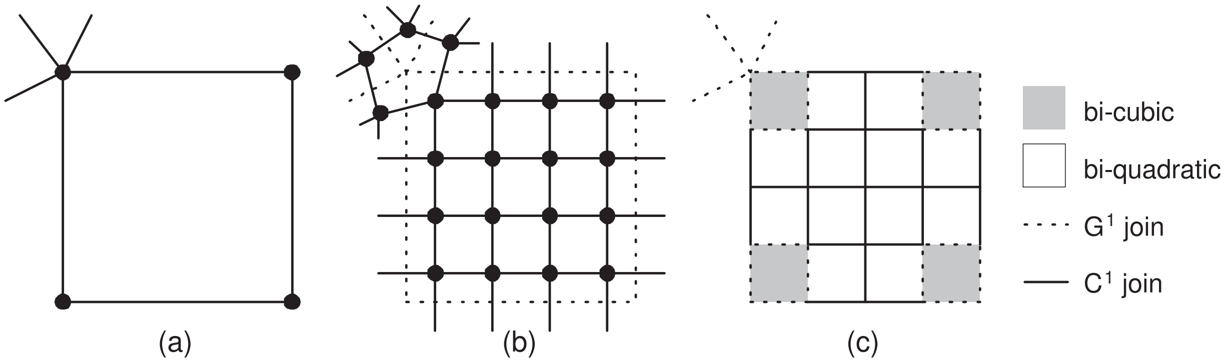


Figure 3: Schematic of the B-spline construction scheme of Peters. (a) one quadrilateral face f of the input mesh M_c; (b) the 4×4 vertices of the refined control mesh M_x associated with f; (c) the 4×4 Bézier patches created from M_x associated with f.

In the second step, a tensor product Bézier patch is constructed centered on each vertex of M_x as shown in Figure 3c. The Bézier patch is defined to be bicubic if the vertex is adjacent to an extraordinary face, otherwise it is defined to be biquadratic. The affine combinations for setting the Bézier control points of these patches as functions of the vertices V_x of M_x are given in the Appendix. Peters [23] proves that the resulting collection of Bézier patches form a G^1 surface, subject to a few linear constraints on V_x near those extraordinary faces for which m is even and greater than 4 (see Appendix). We denote this G^1 surface as $S(V_x)$.

Over each quadrilateral face of M_c, the collection of 4×4 Bézier patches (in general 12 biquadratic and 4 bicubic) can be combined into a single tensor product bicubic B-spline patch (with $k = 4$ and $n = 11$). To satisfy the G^1 and C^1 joins indicated in Figure 3c, the knot sequences in both parameter directions are set to $U_f = V_f = (0, 0, 0, 0, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{2}, \frac{1}{2}, \frac{3}{4}, \frac{3}{4}, \frac{3}{4}, 1, 1, 1, 1)$. The B-spline representation requires 15% less storage than storing each Bézier patch separately.

Modified fitting step To apply Peters' scheme to the problem of B-spline fitting, we modify the *fitting step* in the iterative procedure described in Section 3.4.1.

We use the quadrilateral domain complex $K_\square$ as the input mesh M_c to Peters' scheme. Since $K_\square$ does not possess a geometric embedding, only the topological structure K_x of the control mesh M_x can be constructed initially. The vertices V_x of M_x are computed by fitting the B-spline surface $S(V_x)$ to the data points. Specifically, we compute V_x by minimizing the energy functional $E(V_x) = E_{dist}(V_x) + \lambda E_{fair}(V_x)$ for fixed parametrizations $t_i = (f_i, u_i, v_i)$ of the data points $\mathbf{p}_i$.

Since Peters' construction is affine, every point $\mathbf{s}(t)$ on the surface S can be written as an affine combination of V_x. Treating V_x as a matrix whose rows are (x, y, z) coordinates, we can express this affine combination as $\mathbf{s}(t) = \mathbf{y} V_x$ where the entries of the row vector $\mathbf{y}$ are obtained by appropriately composing Bernstein polynomials and the formulas given in the Appendix. We can therefore rewrite E_{dist} as

$$E_{dist}(V_x) = \sum_{i=1}^{N} \|\mathbf{p}_i - \mathbf{y}_i V_x\|^2$$

which is quadratic on V_x. The term E_{fair} can similarly be expressed as a quadratic function over V_x by summing up the thin-plate energies of all Bézier patches and using the formulas given in the Appendix.

Thus, $E(V_x)$ is a quadratic functional on V_x, and therefore its minimization is a linear least squares problem. Moreover, the linear system is sparse because of the locality of the surface construction. As mentioned earlier, some linear constraints on V_x must be satisfied near extraordinary faces for the surface to be G^1. These constraints are introduced into the optimization through the use of *Lagrange multipliers*, making the problem only slightly more difficult (see [16] for details).

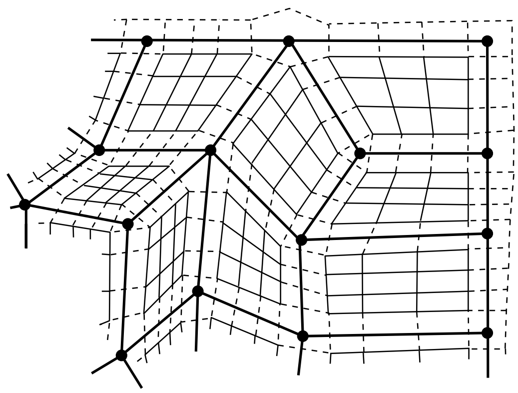


Figure 4: Example of construction of M_x from a domain $K_\square$ containing a boundary.

Extensions to the basic fitting method We generalize the construction of M_x to allow surface boundaries in $K_\square$. For each boundary edge of $K_\square$, we add an additional layer of vertices to M_x. To each valence m boundary vertex of $K_\square$ we associate in M_x a $(2m-2)$-sided face if $m \neq 2$ and a 4-sided face if $m = 2$. This process is illustrated in Figure 4. As a result, the boundaries of S are smooth everywhere except at valence 2 boundary vertices of $K_\square$ where surface corners are introduced.

The two Doo-Sabin subdivisions in the first step of Peters' construction serve to isolate the extraordinary faces. With two subdivisions, a 4×4 grid of vertices is introduced on each face of $K_\square$ as shown in Figure 3b. More generally, a construction with $s \times s$ vertices on each face of $K_\square$ still results in a G^1 surface for any $s \geq 2$. We have experimented with different values of s in the fitting procedure, but have obtained best visual results with the original setting of $s = 4$.

3.5 Adaptive refinement

The surface fitting algorithm described in Section 3.4 minimizes the total squared distances $\sum_i d(\mathbf{p}_i, S)^2$ of the data points $\mathbf{p}_i$ to the B-spline surface S. It is often desirable to specify a maximum error tolerance for the fit. Step 5 attempts to find a surface S such that $\max_i d(\mathbf{p}_i, S) < \epsilon$ for a user-specified error tolerance ϵ.

To achieve a given tolerance within our least squares optimization framework, it may be necessary to introduce new degrees of freedom into the surface representation. One could achieve this by globally subdividing the domain $K_\square$ (e.g. using template 4 in Figure 5). However, this would introduce degrees of freedom uniformly over the whole surface, even if data points exceed the error tolerance only in isolated neighborhoods.

We instead develop an adaptive refinement scheme. The goal of this refinement scheme is to subdivide any face of $K_\square$ onto which any point $\mathbf{p}_i$ projects with $d(\mathbf{p}_i, S) > \epsilon$, while at the same time ensuring that the resulting subdivided faces still form a valid patch network $K_\boxplus$.

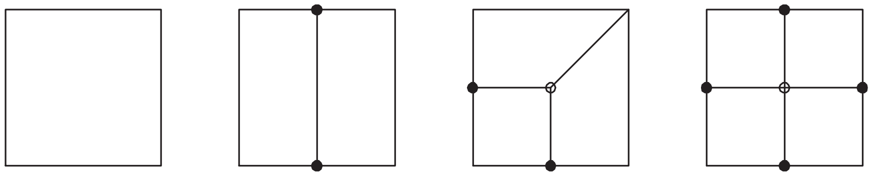


Figure 5: The four face refinement templates.

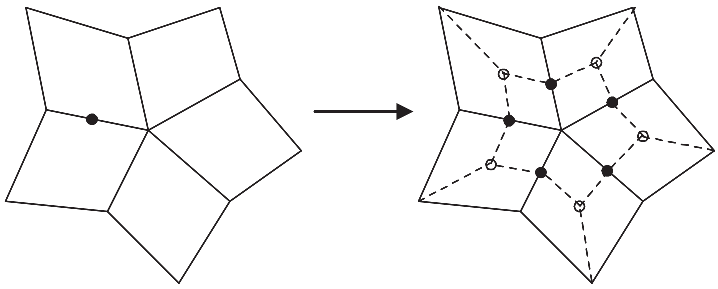


Figure 6: Example of closure of E': on the left E' initially contains only one edge; on the right its computed closure contains 5 edges, resulting in the face refinement indicated by the dashed edges.

We specify the refinement of $K_\square$ by selecting a subset $E' \subset E$ of edges in $K_\square$. For each edge in E', a new vertex is introduced at its midpoint. (The selection of E' will be discussed shortly.) We then subdivide each face of $K_\square$ using one of the 4 face refinement templates shown in Figure 5, depending on which of its edges are in E'.

Note that constraints exist on valid choices of E', since the face refinement templates can only be applied to faces with 0, 2, or 4 refined edges. To satisfy these constraints, any chosen set E' is augmented with additional edges so that all faces have an even number of refined edges. Our algorithm for achieving this closure is as follows. We place all faces of $K_\square$ onto a stack. In each iteration, we remove the face at the front of the stack. If it has three refined edges, we add the fourth edge to E' and push the neighboring face on the stack. If instead it has one refined edge, we add to E' the next clockwise edge on the face and push the neighboring face on the stack. The algorithm is guaranteed to terminate, since, in the worst case, E' will contain all edges of $K_\square$ (which leads to global refinement). Figure 6 demonstrates a refinement obtained when a single edge is initially placed in E'.

We now address the problem of selecting the set E' that determines the refinement. Our algorithm considers all data points with $d(\mathbf{p}_i, S) > \epsilon$ in order of decreasing $d(\mathbf{p}_i, S)$. For each of these data points, if the face onto which it projects is not set to be subdivided (i.e. none of its edges are in E'), then all its edges are added to E', and the closure of the resulting E' is computed.

Having constructed the locally refined domain $K_\boxplus$, we update the parametrizations of the points P. The new vertices introduced in $K_\boxplus$ lie either at the midpoints of edges (coordinates $(0, \frac{1}{2})$, $(1, \frac{1}{2})$, $(\frac{1}{2}, 0)$, $(\frac{1}{2}, 1)$), or at the centroid of faces (coordinates $(\frac{1}{2}, \frac{1}{2})$). Reparametrization on faces created by face refinement templates 1, 2, and 4 proceeds in the obvious way, since there exists a unique piecewise bilinear map between the original face and the quadrilateral subfaces. For a face subdivided by template 3, however, such a bilinear map does not exist on the two trapezoid pieces, so we approximate it by assuming that the original quadrilateral has the geometry of a square.

After adaptive refinement, the fitting method of Step 4 is reinvoked. The resulting surface may still not be within ϵ of all the points, indicating that further refinement is necessary. We repeat the process of refinement and refitting until the error tolerance ϵ is satisfied. Figures 9j– 9l show the resulting surfaces.

4 Results

Figure 9 shows the reconstruction of a B-spline surface from a set of 4000 points; this synthetic data set was obtained by randomly sampling an existing surface. Figures 10a–10c, 10d–10f, and 10g–10i show reconstructions using real data obtained from a laser range scanner (courtesy of Technical Arts Co.).

Figures 10j–10l show the B-spline approximation of a mesh S_0 of 69,473 faces (courtesy of Turk and Levoy). To approximate S_0, a set P of 30,000 points is sampled randomly from its surface. Step 1 of the procedure is skipped, and S_0 is used directly as the initial mesh M_0.

Table 1: Parameter settings and execution times.

Object	#points N	Tolerance ϵ	Fairness λ	Execution times (minutes) Step 1	Step 2	Step 3	Step 4	Step 5
holes3	4,000	0.6%	0.1	1	1	1	12	134
club	16,585	0.7%	0.1	6	1	1	11	599
foot	20,021	0.3%	0.05	7	13	1	12	228
skidoo	37,974	0.7%	0.1	12	2	1	14	132
bunny	30,000	1.5%	0.1	—	16	1	45	200

Table 2: Surface complexities and B-spline fit errors.

Object	M_0 #faces	$K_\triangle$ #faces	Initial S #patches	Initial S fit error rms	Initial S fit error max	Refined S #patches	Refined S fit error rms	Refined S fit error max
holes3	2,080	98	49	0.14%	0.75%	178	0.07%	0.59%
club	5,152	72	35	0.22%	1.36%	285	0.06%	0.41%
foot	10,972	62	29	0.20%	1.20%	156	0.03%	0.27%
skidoo	3,661	30	15	0.23%	3.00%	94	0.03%	0.69%
bunny	69,473	162	72	0.43%	4.64%	153	0.19%	1.44%

As Table 1 indicates, the user-specified parameters are the maximum error tolerance ϵ and the fairness weight λ. (To make these values unitless, we uniformly scale the data points P to fit within a unit cube.) The table also compares the execution times of the 5 successive steps, as obtained on a 105 MHz HP 735 workstation. Table 2 lists for each example the complexities of the initial mesh M_0 and the base complex $K_\triangle$. It also shows the fit errors of both the initial B-spline surface (Step 4) and the adaptively refined B-spline surface (Step 5), giving both rms and maximum errors as percentages of the object diameter.

5 Summary and Future Work

We have developed a procedure for constructing a G^1 tensor product B-spline surface of arbitrary topological type from a set of 3D points without user assistance. The procedure makes use of a surface spline construction to obtain G^1 continuity; we show that such an approach leads to an efficient B-spline fitting method. We have introduced an adaptive refinement algorithm. Finally, we have applied our procedure to reconstruct B-spline surfaces within user-specified maximum error tolerances on a number of real data sets.

There exist a number of areas for future research. The procedure should be extended to allow reconstruction of piecewise smooth surfaces that contain discontinuities such as creases and corners [10]. To this end, some user intervention may be desirable in identifying the discontinuities during the construction of the patch network.

In the context of surface approximation, the current procedure provides error bounds $d(\mathbf{p}_i, S)$ between a set of sampled points and the approximating surface; instead a stronger error bound would be the distance $d(S_0, S)$ between the original surface and its approximation.

Some surfaces such as the mesh in Figure 10j contain fine geometric detail that is difficult to approximate with a smooth surface representation. This detail could easily be stored in the form of a displacement map from the underlying smooth surface as is done in [29]. The resulting hybrid representation (smooth surface plus displacement map) could then be edited with conventional modeling systems.

References

[1] E. Andersson, R. Andersson, M. Boman, T. Elmroth, B. Dahlberg, and B. Johansson. Automatic construction of surfaces with prescribed shape. *CAD*, 20:317–324, 1988.

[2] Chandrajit Bajaj, Fausto Bernardini, and Guoliang Xu. Automatic reconstruction of surfaces and scalar fields from 3D scans. *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 109–118, August 1995.

[3] Ulrich Dietz. Erzeugung glatter Flächen aus Meßpunkten. Technical Report 1717, Department of Mathematics, University of Darmstadt, Germany, February 1995.

[4] D. Doo and M. Sabin. Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design*, 10(6):356–360, September 1978.

[5] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis on arbitrary meshes. *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 173–182, August 1995.

[6] Lian Fang and David Gossard. Reconstruction of smooth parametric surfaces from unorganized data points. In J. Warren, editor, *Curves and Surfaces in Computer Vision and Graphics 3*, volume 1830, pages 226–236. SPIE, 1992.

[7] Gerald Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, 3rd edition, 1992.

[8] David Forsey and Richard Bartels. Surface fitting with hierarchical splines. *ACM Transactions on Graphics*, 14(2):134–161, 1995.

[9] Günther Greiner. Variational design and fairing of spline surfaces. *Computer Graphics Forum*, 13(3):143–154, 1994.

[10] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. *Computer Graphics (SIGGRAPH '94 Proceedings)*, pages 295–302, 1994.

[11] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics (SIGGRAPH '92 Proceedings)*, 26(2):71–78, 1992.

[12] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. *Computer Graphics (SIGGRAPH '93 Proceedings)*, pages 19–26, 1993.

[13] J. Hoschek and D. Lasser. *Fundamentals of Computer Aided Geometric Design*. AK Peters, Wellesley, 1993.

[14] J. Hoschek, F.-J. Schneider, and P. Wassum. Optimal approximate conversion of spline surfaces. *CAGD*, 6:293–306, 1989.

[15] Eugene L. Lawler. *Combinatorial optimization: networks and matroids*. Holt, Rinehart, and Winston, 1976.

[16] C.L. Lawson and R.J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs, NJ, 1974.

[17] S.L. Lee, H.H. Tan, and A.A. Majid. Smooth piecewise biquartic surfaces from quadrilateral control polyhedra with isolated n-sided faces. *CAD*, 27:741–758, 1995.

[18] Charles Loop. Smooth spline surfaces over irregular meshes. *Computer Graphics (SIGGRAPH '94 Proceedings)*, pages 303–310, July 1994.

[19] W. Ma and J.P. Kruth. Parametrization of randomly measured points for least squares fitting of B-spline curves and surfaces. *CAD*, 27:663–675, 1995.

[20] M.J. Milroy, C. Bradley, G.W. Vickers, and D.J. Weir. G^1 continuity of B-spline surface patches in reverse engineering. *CAD*, 27:471–478, 1995.

[21] Doug Moore and Joe Warren. Approximation of dense scattered data using algebraic surfaces. In V. Milutinovic and D.B. Shriver, editors, *Proccedings of the 24th Annual Hawaii International Conference on System Sciences, Kauai, HI, USA*, pages 681–690, Los Alamitos, CA, USA, 1991. IEEE Comp. Soc. Press.

[22] Henry Moreton and Carlo Séquin. Functional optimization for fair surface design. *Computer Graphics*, 26(3):167–176, July 1992.

[23] Jörg Peters. Constructing C^1 surfaces of arbitrary topology using biquadratic and bicubic splines. In N. Sapidis, editor, *Designing Fair Curves and Surfaces*, pages 277–293. SIAM, 1994.

[24] Jörg Peters. Biquartic C^1-surface splines over irregular meshes. *CAD*, 1995. submitted.

[25] Ulrich Reif. Biquadratic G-spline surfaces. *CAGD*, 12:193–205, 1995.

[26] D.F. Rogers and N.G. Fog. Constrained B-spline curve and surface fitting. *CAD*, 21:641–648, 1989.

[27] Biplab Sarkar and Chia-Hsiang Menq. Parameter optimization in approximating curves and surfaces to measurement data. *CAGD*, 8:267–290, 1991.

[28] F.J.M. Schmitt, B.A. Barsky, and W. Hui Du. An adaptive subdivision method for surface fitting from sampled data. *Computer Graphics (SIGGRAPH '86 Proceedings)*, 20:179–188, August 1986.

[29] Stan Sclaroff and Alex Pentland. Generalized implicit functions for computer graphics. *Computer Graphics (SIGGRAPH '91 Proceedings)*, 25:247–250, 1991.

[30] R.E. Tarjan. Data structures and network algorithms. *CBMS-NSF Regional Conference Series in Applied Mathematics*, 44, 1983.

[31] Greg Turk and Marc Levoy. Zippered polygon meshes from range images. *Computer Graphics (SIGGRAPH '94 Proceedings)*, 28(3):311–318, July 1994.

A Appendix

The purpose of this appendix is to present the formulas expressing the control points of the Bézier patches of S (Figure 3c) as affine combinations of the control mesh vertices V_x (Figure 3b) in Peters' surface spline construction [23]. Recall that a Bézier patch is associated with each vertex of M_x.

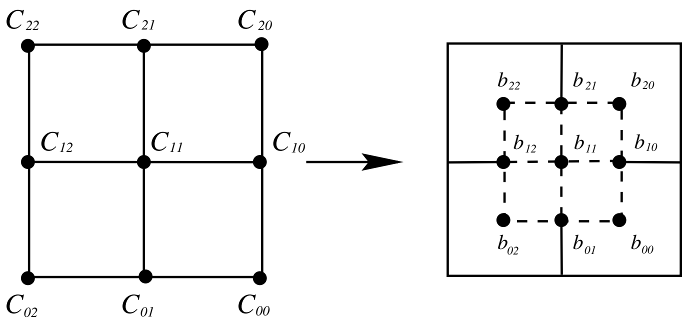


Figure 7: Regular case: neighborhood of vertex $C_{11} \in V_x$ giving rise to a biquadratic patch.

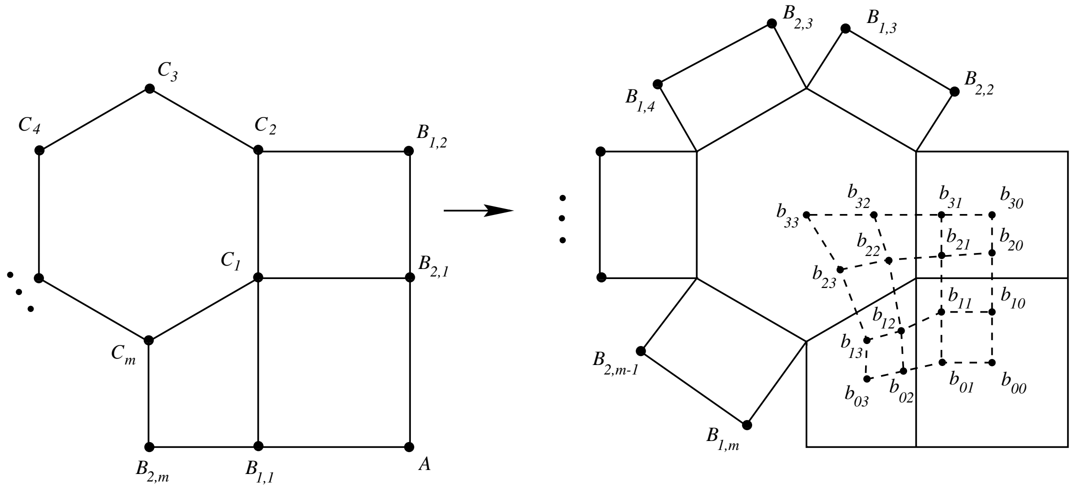


Figure 8: Extraordinary case: neighborhood of vertex $C_1 \in V_x$ giving rise to a bicubic patch

1. Since Peters' surface scheme generalizes biquadratic B-splines, in the regular case of a vertex C_{11} adjacent to four 4-sided faces (Figure 7), a biquadratic Bézier patch is created. The formulas for its Bézier points are obtained trivially:

$$\begin{aligned} b_{00} &= (C_{00} + C_{10} + C_{01} + C_{11})/4 \\ b_{10} &= (C_{11} + C_{10})/2 \\ b_{01} &= (C_{11} + C_{01})/2 \\ b_{11} &= C_{11} \end{aligned}$$

(The remaining Bézier points follow by symmetry.)

2. At a vertex C_1 near an m-sided extraordinary face (Figure 8), a bicubic Bézier patch is created. The formulas for its Bézier points are quite difficult and are derived in [23]:

$$\begin{aligned} b_{00} &= (B_{2,1} + B_{1,1} + C_1 + A)/4 \\ b_{10} &= (5B_{2,1} + B_{1,1} + 5C_1 + A)/12 \\ b_{20} &= (5B_{2,1} + B_{1,2} + 5C_1 + C_2)/12 \\ b_{30} &= (B_{2,1} + B_{1,2} + C_1 + C_2)/4 \\ b_{11} &= (5B_{2,1} + 5B_{1,1} + (25 + 4a)C_1 + (1 - 4a)A)/36 \end{aligned}$$

$$\begin{aligned}
b_{21} &= ((5-10a)B_{2,1} + (1+2a)B_{1,2} + (25+6a)C_1 + (5+2a)C_2)/36 \\
b_{31} &= h_{1,1} \\
b_{22} &= \begin{cases} -\sum_{i=1}^{m}(-1)^i\, h_{3,i} & \text{if } m \text{ is odd,} \\ -\frac{2}{m}\sum_{i=1}^{m}(-1)^i\,(m-i)\,h_{3,i} & \text{if } m \text{ is even,} \end{cases} \\
b_{32} &= h_{2,1} \\
b_{33} &= \frac{1}{m}\sum_{i=1}^{m} C_i
\end{aligned}$$

where the following abbreviations are used:

$$\begin{aligned}
c &= \cos(2\pi/m) \\
a &= c/(1-c) \\
h_{1,i} &= ((1-2a)B_{2,1} + (1-2a)B_{1,2} + (5+2a)C_1 + (5+2a)C_2)/12 \\
h_{2,i} &= \frac{1}{m}\sum_{l=1}^{m} C_l + \frac{2a}{3c}\cos(2\pi l/m)\,(C_{i+l} + C_{i+l+1}) \\
h_{3,i} &= (1-\frac{2}{3}c)\,h_{2,i} + \frac{2}{3}\,c\,h_{1,i}
\end{aligned}$$

(The remaining Bézier points again follow by symmetry.)

Finally, in the case that the number of sides m of the extraordinary face is even and greater than 4 the following linear condition must hold for G^1 continuity:

$$\sum_{i=1}^{m}\sum_{j=1}^{2}(-1)^{i+j}B_{i,j} = 0$$