

# Automatic Conversion of Meshes to T-spline Surfaces

Category: Research

## Abstract

We give an *automatic* method for fitting T-spline surfaces to triangle meshes of arbitrary topology. Previous surface fitting methods required tedious human interaction to accurately capture the geometry and features. There are two main steps in our method. The first computes a curvilinear coordinate system—a *conformal net*—on the surface, induced by global conformal parameterization. This global net is then automatically partitioned to give several rectangular patches which locally have tensor product structure suitable for defining splines. In the second step, each rectangular patch is approximated to a desired positional and normal accuracy by a T-spline surface, with appropriate continuity between patch boundaries. Use of T-splines enables us to use a low number of control points while guaranteeing  $L^\infty$  error behaviour. The only user inputs required in the whole process are these two fitting tolerances.

**CR Categories:** I.3.5 [Computational Geometry and Object Modeling]: Splines—T-Splines; I.3.5 [Computational Geometry and Object Modeling]: Boundary Representations—Mesh;

**Keywords:** Polygonal meshes, Spline surfaces, Riemann surfaces, Global conformal parameterization, T-splines

## 1 Introduction

Triangle meshes and spline surfaces are the most widely used representations in computer graphics and geometric modelling. Triangle meshes are supported by graphics hardware and hence are widely used for visualization, computer games, etc. Spline surfaces are the main representation used for computer aided design and manufacturing. Using 3D laser scanners, surface data from real objects can easily be acquired [Bernardini and Rushmeier 2002]. This is usually initially in the form of unorganized points, which are then joined to form a triangular mesh. In order to use these meshes in many current CAD systems, and to facilitate interactive editing, it is necessary to convert them to spline surfaces. Furthermore, alternative representations are more suited to different applications, so automatic conversion between them is of great importance.

NURBS surfaces are of widespread use, but they have several disadvantages. Recently, Sederberg et al. generalized them, giving T-splines [Sederberg et al. 2003; Sederberg et al. 2004]. T-splines require significantly *fewer* control points to represent complex geometric information: they have better local refinement capabilities, because of the more flexible approach to joining T-splines at their common boundaries.

Traditional surface fitting methods minimize global errors in a least squares sense, and do not ensure a bounded  $L^\infty$  error norm, which may be required by users. The local refinement method of T-splines makes them well suited to mesh fitting, where an adaptive algorithm can locally achieve the desired approximation quality and control  $L^\infty$  errors. The quality of normal vectors of the fitted surface

is also often important. Our approach combines positional and normal tolerances in the objective function used for fitting, allowing us to bound both kinds of error.

To convert meshes of arbitrary topology to parametric surfaces, including T-splines, they must first be parameterized. We use *conformal parameterization*, which preserves angles between the parameterization and the surface, ensuring that the parameterization is well behaved, thus helping to guarantee high-quality reconstructed normals. Because T-splines are *globally* defined on meshes, we do not wish to partition the mesh into many separately parameterized patches: instead, we use a global parameterization: *global conformal parameterization*, introduced in [Gu and Yau 2003].

The latter works by finding a smooth pair of vector fields mutually orthogonal to each other except at a number of *zero points*; a surface of genus  $g$  has  $|2g - 2|$  zero points. The two families of integral curves of these fields are arbitrarily called *horizontal* and *vertical trajectories*, and form a *conformal net*. Locally, this has a tensor product structure, enabling knots of a T-spline to be defined directly on it. Globally, conformal nets have a particular structure which can be directly used for T-spline fitting: horizontal trajectories through the zero points segment the mesh into topological cylinders and disks. Each segment can be mapped to a planar rectangle by mapping the horizontal and vertical trajectories to iso-parametric lines in the plane. T-spline patches can be defined on these rectangles directly. By using common knots and control points along their boundaries, the desired continuity can be achieved.

**Novelty** This work presents an automatic algorithm for converting triangle meshes to T-Spline surfaces, requiring only a user-specified  $L^\infty$  position tolerance and a normal tolerances. It works surfaces with multiple handles and open boundaries. The algorithm is based on the intrinsic Riemann structure of the surface, which is independent of the triangulation: it is stable with respect to small deformations. The parameter net of the T-spline surface is close to conformal, which is valuable for purposes of texture mapping and geometric computation.

## 2 Related work

Before presenting our new method, we discuss related work in two areas: fitting methods, and global parameterization methods.

### 2.1 $L^2$ surface fitting

Surface fitting techniques can be classified as *approximating* techniques and *interpolation* techniques.

Finding a parametric surface  $S$  which approximates a target surface  $T$  is usually performed by minimizing the distance between them with respect to some metric. The distance metric usually used is a weighted sum of squared distances of samples  $t_i$  taken from  $T$  to  $S$ :

$$L^2(T, S) = \sum_{t_i \in T} w_i \|d(t_i, S)\|^2. \quad (1)$$

Here,  $d(t, S)$  denotes the distance between  $t$  and its image point  $s$  on  $S$ :  $d(t, S) = \|t - s\|$ . Weights typically reflect the sampling density, which may not be uniform. Finding the image points is a key step in fitting, and can be done in two basic ways, with or without the aid of a parameterization.

Parameterization-free methods find image points by computing foot points of the samples on the approximating surface with respect to Euclidean distance. Much work has been done on this problem. A survey is given in [Sapidis 1994]; for more recent advances, see [Dodgson et al. 2003; Marinov and Kobbelt 2004]. Parameterization-based methods parameterize the target surface in a planar domain, and then select the image points by identifying the parameter values. Such methods have also received considerable attention; an extensive survey is given in [Weiss et al. 2002].

Both of these approaches are limited. In parameterization-free methods, nearest point computation is time-consuming and error prone [Marinov and Kobbelt 2004]. Selecting a good initial surface for iterative optimization is also difficult [Cheng et al. 2004]—self-intersection of the control mesh can easily result if the curvature of the target surface changes rapidly. To avoid this problem, a surface fairness term is usually added, but choosing the best balance smoothness and a good fit is tricky [Weiss et al. 2002].

Parameterization-based methods either rely on minimizing a global error function, or projecting samples projected onto a *base surface*, which can again be hard to choose. Both approaches can introduce serious local distortions, resulting in unwanted ripples for complicated surfaces.

Irregular distribution of data points on the target surface is another serious problem for surface approximation techniques. To circumvent this issue, Pottmann et al. [Pottmann and Leopoldseder 2003] devised an approach in which the roles of  $T$  and  $S$  are swapped. One can compute the distance field for  $S$  as a prior, bypassing the problem of irregularly spaced data points. However, their approach still suffers from the other problems associated with nearest point methods and is not well suited to approximating complicated surfaces.

Interpolation schemes are an alternative to approximation. Usually, a set of interpolation constraints [Halstead et al. 1993] is generated, and fitting is performed by solving a linear system. However, this approach can suffer from poor conditioning. Litke et al. [Litke et al. 2001] introduced a quasi-interpolation scheme in which the control points are computed from the target surface directly. Interpolation schemes suffer from two problems. Again, like approximation schemes, they need a parameterization; choosing the interpolation constraints is also hard.

## 2.2 $L^\infty$ surface fitting

Surface fitting using a fixed number of control points can not guarantee high approximation quality. To obtain bounded  $L^\infty$  fitting errors, additional control points need to be inserted as fitting proceeds, to accommodate local shape details.

Multiresolution structures have been used in [Lee et al. 2000; Litke et al. 2001] as a means of reducing the  $L^\infty$  error adaptively. Although this approach can produce high approximation quality, the usual approach subdivides the surface *globally*, making many of the added control points redundant.

Instead, if we try to refine the surface locally, and insert knots where the approximation error is high, we have the difficult issue of deciding how many knots should be inserted. A common approach [Cham and Cipolla 1999; Yang et al. 2004] inserts knots into the region of maximum  $L^\infty$  error and then does global optimization. This strategy works well for curve fitting, but for surface fitting, simply subdividing the region having maximum  $L^\infty$  error does not work well in practice. Marinov et al. [Marinov and Kobbelt 2004] suggest inserting knots into *all* regions where the  $L^\infty$  error exceeds a given threshold. However, this has the disadvantage of introducing redundant control points that need to be subsequently removed.

By using T-splines, we are able locally insert knots, and only have to perform local fitting to adjust the surface, without an expensive global computation—the surface elsewhere remains unchanged.

## 2.3 Segmentation methods

To fit a parametric surface to a polygon mesh of arbitrary topology, the polygon mesh is usually segmented into a number of rectangular regions that are approximated by parametric patches. Many previous methods, e.g. [Andersson et al. 1988; Krishnamurthy and Levoy 1996], require the user to manually delineate the patch boundaries. However, for surfaces of complicated topology, this is very tedious.

Instead of using manual interaction, Eck and Hoppe [Eck and Hoppe 1996] describe a method for producing quadrilateral patches based on a parameterization phase [Eck et al. 1995] and a remeshing phase. The number of patches is adjusted to achieve desired fitting tolerances. While this method produces high quality surfaces, many extra control points are needed to achieve continuity along boundaries and at patch corners.

Katz et al. [Katz and Tal 2003] give another mesh segmentation approach based on fuzzy clustering and cuts. Their method, while useful, produces segments which do not always have rectangular boundaries. Also, further parametrization is needed to ensure continuity along boundary curves.

We use global conformal parameterization as a means of providing automatic segmentation; it also gives the parameterization needed for T-spline fitting, guaranteeing continuity along patch boundaries.

## 2.4 Conformal parameterization

Several recent advances in surface parameterization [Floater and Hormann 2004] have been based on solving a discrete Laplace system [Pinkall and Polthier 1993; Duchamp et al. 1997; Floater 1997; Floater 2003]. Lévy et al. [Lévy et al. 2002] describe a technique for finding conformal mappings by least squares minimization of *conformal energy*, and Desbrun et al. [Desbrun et al. 2002] formulate a theoretically equivalent method of discrete conformal parameterization. Sheffer et al. [Sheffer and Sturler 2001] give an angle-based flattening method for conformal parameterization.

Gu and Yau [Gu and Yau 2003] considered construction of a global conformal structure for a manifold of arbitrary topology by finding a basis for holomorphic differential forms, based on Hodge theory [Schoen and Yau 1997].

Ni et al. [Ni et al. 2004] use the idea of a harmonic Morse function to extract the topological structure of a surface. Dong et al. [Dong et al. 2005, to appear] give a method for quadrilateral remeshing of manifolds using harmonic functions. The method is theoretically equivalent to using a holomorphic differential form as described in [Gu and Yau 2003]. The differential forms in the latter have at least 4 fewer zero points than those in the former, however. For the current problem, it is desirable to have fewer zero points, as they affect the global structure of the parameterization significantly. Thus, we have adapted Gu and Yau’s method. Note, however, that we do not remesh the surface, but fit a spline on the parameter domain directly.

Figure 1 illustrates the idea of a conformal net, showing front and back of a sculpture of genus 3. The global conformal parameterization is visualized by mapping a checkerboard onto the surface. The horizontal trajectories through the zero points partition the surface into 6 cylinders. The red crosses indicate the positions of zero points of the conformal net.

## 3 Parameterization

Basic concepts of Riemann surface theory used in this section are explained in the appendix.

We now explain in detail our algorithm to construct the conformal net. The conformal net is a natural curvilinear coordinate sys-

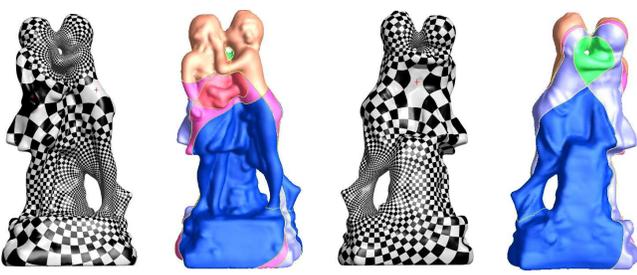


Figure 1: Global conformal parameterization on a surface. Locally, conformal nets have a tensor product structure, so are suitable for defining a T-spline surface. They also have a simple global structure: they can be treated as rectangles with regular grids, glued along their edges and at zero points in a special pattern—see Figure 2. Figure 2(a) illustrates four regular rectangles; the black dots are the zero points. Figure 2(b) shows how each rectangle is mapped to a quadrant, and merged together at the zero point. Figure 2(c) shows three at a central zero point. The red curves are the horizontal trajectories, and the blue ones

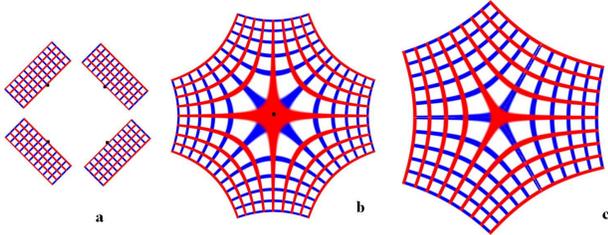


Figure 2: Global structure of conformal nets.

are the vertical trajectories. Note that in the general case, exactly four patches meet at a zero point, and each rectangle is mapped to a quadrant. Gluing opposite sides of a single rectangle gives a cylinder or a torus.

The input surface is represented by a triangular mesh, and each vector field is represented as a function defined on the edges of the mesh.

Suppose  $K$  is a simplicial complex, and a mapping  $\mathbf{r} : K \rightarrow \mathbb{R}^3$  embeds  $K$  in  $\mathbb{R}^3$ . Then  $M = (K, f)$  is called a *triangular mesh*. For dimensions  $n = 0, 1, 2$ , we denote the sets of  $n$ -simplices by  $K_n$ . Any given  $n$ -simplex is denoted by  $[v_0, v_1, \dots, v_n]$ , where  $v_i \in K_0$ , i.e. the  $v_i$  are points. A holomorphic 1-form is represented as a function defined on the edges:  $\omega : K_1 \rightarrow \mathbb{R}^2$ .

The methods for constructing a conformal net vary different types of surface. We now explain the details for each case: genus-0 closed surfaces, genus-1 closed surfaces, higher genus closed surfaces and surfaces with boundaries.

### 3.1 Genus-0 closed surfaces

Every genus zero closed surface  $S$  can be conformally mapped to a sphere. Practical algorithms for computing such maps are given in [Gu et al. 2004; C. Gotsman 2003]. The idea used in [Gu et al. 2004] is that, for genus-0 closed surfaces, conformal maps are equivalent to harmonic maps, which can be computed using the heat flow method.

Having found the conformal map  $f : S \rightarrow S^2$  to the sphere, we use spherical coordinates  $(\theta, \phi)$  as parameters. The horizontal trajectories on  $S$  are the curves  $f^{-1}(\phi = \text{const.})$ , and the vertical trajectories are  $f^{-1}(\theta = \text{const.})$ . The preimages of the North and South poles are the zero points. The trajectories are orthogonal everywhere except at the zero points and form the conformal net.

### 3.2 Genus-1 closed surfaces

In this case, to find the conformal net, we need to first compute a *holomorphic 1-form*  $\omega$  which can be treated as a pair of vector fields  $\omega = (\omega_u, \omega_v)$  satisfying  $\omega_v = \mathbf{n} \times \omega_u$ , where  $\mathbf{n}$  is the surface normal. Furthermore,  $\omega_u$  and  $\omega_v$  should be harmonic (informally, they should be as smooth as possible). The horizontal trajectories are the integral curves of  $\omega_u$ , the vertical trajectories are the integral curves of  $\omega_v$ .

For a surface of genus  $g$ , all holomorphic 1-forms form a real  $2g$  dimensional linear space. Algorithms for finding a holomorphic 1-form basis are given in [Gu and Yau 2003], which can be summarized as computing in turn a homology basis, a cohomology basis, a harmonic 1-form basis and a holomorphic 1-form basis. It is beyond the scope of the current paper details this further. See [Gu and Yau 2003] for thoroughly explanations.

In our implementation, the holomorphic 1-form basis  $\{\omega_1, \omega_2, \dots, \omega_{2g}\}$  is represented by vector-valued functions defined on the edges of the mesh,  $\omega_i : K_1 \rightarrow \mathbb{R}^2, i = 1, 2, \dots, 2g$ . Any discrete holomorphic one-form  $\omega$  is a linear combination of them. There are an infinite number of holomorphic 1-forms on  $S$ ; we want to choose the best one for the purposes of spline fitting.

Each holomorphic 1-form  $\omega$  induces a parameterization of  $S$ . We can choose a topological disk  $U$  on  $S$ , an arbitrary point  $p \in U$ , and define the parameterization  $z : U \rightarrow \mathbb{R}^2$ , for any point  $q \in U$ , using

$$z(q) = \int_p^q \omega = \left( \int_p^q \omega_u, \int_p^q \omega_v \right), \quad (2)$$

where the path from  $p$  to  $q$  is arbitrary as long as it lies inside  $U$ .

For the mesh,  $U$  is a set of neighboring triangles, and  $p$  and  $q$  are vertices. The path connecting  $p$  to  $q$  is a series of consecutive edges, and denoted by  $\{e_1, e_2, \dots, e_n\}$ . The parameter of vertex  $q$  is computed by the discrete sum

$$z(q) = \sum_{i=1}^n \omega(e_i). \quad (3)$$

Under a conformal parameterization  $\{U, z\}$ ,  $z = (u, v)$ , the metric on  $S$  can be represented in the simple form

$$ds^2 = \lambda^2(u, v)(du^2 + dv^2), \quad (4)$$

where  $\lambda(u, v)$ —the *conformal factor*—measures area stretching of the parameterization. The uniformity of the parameterization can be measured by computing

$$E(\omega) = \int_U |\nabla \lambda|^2 dudv. \quad (5)$$

We select the holomorphic 1-form which minimizes this uniformity functional, using an automatic algorithm in [Jin et al. 2004].

A holomorphic 1-form  $\omega$  on a genus-1 closed surface  $S$  is nonzero everywhere, so there are no zero points, and the conformal net is simple. By integrating  $\omega$  on  $S$ , the whole surface can be conformally mapped to a parallelogram on the plane, called the *fundamental period* of  $S$ . In general, this is not a rectangle, but a skewed parallelogram whose shape is determined by the conformal structure of  $S$ . If the fundamental period is a rectangle, then all the horizontal and vertical trajectories forming the conformal net on the surface are closed circles. Otherwise, we two families of curves parallel to the sides of the parallelogram are used as the trajectories.

### 3.3 Higher genus closed surfaces

The global structure of conformal nets on higher genus closed surfaces is more complicated due to the existence of zero points. By the Poincaré-Hopf theorem, every vector field on a surface of genus

$g > 1$  must have zero points. A holomorphic 1-form has the unique property that it has the minimal number of zero points, which for a surface of genus  $g$  is  $|2g - 2|$ .

A horizontal trajectory starting from a zero point also ends at a zero point. In general there are four horizontal trajectories starting at each zero point. These trajectories partition the surface into several patches, each of which is either a topological cylinder or disk. By integrating  $\omega$  on each patch using Eqns. 2,3, each patch can be conformally mapped to a rectangle. The resulting conformal net on each patch has a regular tensor product structure.

Using this, the conformal net of a higher genus closed surface can be constructed using the following steps: (i) Define a regular tensor product grid structure on each rectangle. (ii) Glue pairs of opposite sides of each rectangle to form a cylinder. (iii) Glue the various rectangles and cylinders along their boundaries as determined by the conformal structure and the selected holomorphic 1-form.

In order to partition the conformal net into rectangles, we first need to locate the  $|2g - 2|$  zero points, then trace the horizontal trajectories through them.

**Locating zero points** Suppose we have computed a holomorphic 1-form  $\omega$ . We first estimate the inverse of the conformal factor (see Eqn. 4) for each vertex  $v$  of the input surface mesh, using the following formula:

$$\tau(v) = \lambda^{-1}(v) = \frac{1}{n} \sum_{[u,v] \in K_1} \frac{|\omega([u,v])|^2}{|\mathbf{r}(u) - \mathbf{r}(v)|^2}, \quad u, v \in K_0, \quad (6)$$

where  $u$  runs over all vertices connected by an edge to  $v$ ,  $\mathbf{r}(u)$  is the position of vertex  $u$ ,  $[u, v]$  represents an edge from  $u$  to  $v$ ,  $\omega([u, v])$  is the value of  $\omega$  on edge  $[u, v]$ , and  $n$  is the valence of vertex  $v$ . At zero points,  $\lambda$  approaches zero.

We select clusters of vertices whose conformal factors have the lowest 5% of values as candidate locations for zero points. We measure the geometric size of each cluster by computing its diameter, sort them by decreasing size and keep the first  $|2g - 2|$  clusters. For each cluster, we select the vertex closest to the center of gravity of the cluster as the zero point associated with that cluster.

**Tracing horizontal trajectories through zero points** Next, each of the four horizontal trajectories through each zero point needs to be found.

Suppose  $p$  is a zero point. We select an open set  $U$  on the mesh around  $p$ : a collection of neighboring faces forming a topological disk. Let the boundary of  $U$  be  $\partial U$ . We parameterize  $U$  by integrating  $\omega$  as described in Eqns. 2,3. We denote the parameterization by  $z: U \rightarrow \mathbb{R}^2$ ,  $z(q) = (z_1(q), z_2(q)) = \int_p^q \omega$ . Let the horizontal trajectory through  $p$  be  $\gamma$ ;  $\gamma$  is mapped to the  $x$ -axis of the plane.

We next find all edges  $[u, v]$ , such that  $z([u, v])$  intersects the  $x$ -axis, i.e. for which  $z_2(u) \cdot z_2(v) < 0$ , and we split each edge  $[u, v]$  by adding a new vertex at  $\mathbf{r}(w)$  with parameter  $z(w)$ :

$$\mathbf{r}(w) = \alpha \mathbf{r}(v) + (1 - \alpha) \mathbf{r}(u), \quad z(w) = \alpha z(v) + (1 - \alpha) z(u), \quad (7)$$

where  $\alpha = z_2(u) / (z_2(u) - z_2(v))$ . We now connect all newly inserted vertices to form the horizontal line in the plane and the horizontal trajectory on the mesh accordingly. We will find four curves which are part of the horizontal trajectory  $\gamma$ .

Next, we select another open set  $U'$  on the mesh, overlapping with  $U$ , to be the next chart. We parameterize  $U'$  in a similar way by integrating  $\omega$  on it, giving  $z': U' \rightarrow \mathbb{R}^2$ . The image of  $\gamma$ ,  $z'(\gamma)$ , is still a horizontal line. We select one of the inserted vertices  $w \in U \cap U'$ ; the line  $y = z'_2(w)$  is  $z'(r)$ . Using the same approach as for finding  $\gamma$  in  $U$ , we can extend  $\gamma$  on  $U'$  accordingly. In this way we extend  $\gamma$  chart by chart, until reaching another zero point.

Ultimately, all the horizontal trajectories through zero points partition the surface into several patches, each of which is either a topological cylinder or a disk. If it is a cylinder, we pick an arbitrary point on it, and trace the corresponding vertical trajectory through the point to slice it into a disk. Then all disks are conformally parameterized by a rectangle using  $\omega$ .

### 3.4 Surfaces with boundaries

For surfaces with boundaries, computing a conformal net needs a further step: *double covering*.

Given a surface  $S$  with boundaries  $\partial S$ , we replicate it, and reverse the orientation of the copy,  $-S$ , by reversing the vertex order around each face. We glue  $S$  to  $-S$  along corresponding boundaries: if  $\partial S = \sigma_1 \cup \sigma_2 \cdots \cup \sigma_k$ , then  $\partial -S = -\sigma_1 \cup -\sigma_2 \cdots \cup -\sigma_k$ , and  $\sigma_i$  is glued to  $-\sigma_i$ . The double covered surface is a closed surface  $\bar{S}$ . Each face of  $S$  has two copies on  $\bar{S}$ .

If  $\bar{S}$  is a topological sphere, we can conformally map it to a sphere using a special conformal map which maps the boundary of  $S$  to the equator. The preimages of the lines of longitude and latitude form the conformal net.

More generally, if  $S$  has genus  $g$  and  $b$  boundaries,  $\bar{S}$  is a closed surface with genus  $2g + b - 1$ . We compute the holomorphic 1-form basis of  $\bar{S}$ , and then find a special holomorphic 1-form  $\omega = (\omega_u, \omega_v)$  on it such that  $\omega_u$  is orthogonal to  $\partial S$  everywhere, using the approach in [Gu and Yau 2003]. This  $\omega$  induces a conformal net on  $S$  itself for which all curves in  $\partial S$  are vertical trajectories. The algorithm for locating zero points is similar to that for closed surfaces, except that in the tracing algorithm, horizontal trajectories starting from zero points may end at the boundary.

### 3.5 Mesh thinning

Computing the global conformal parametrization for a triangle mesh needs the solution of a linear system, which is expensive for a large number of vertices. We use this parameterisation to give the segmentation, and the topology of the initial T-spline. For this, we do not need a precise conformal parameter for every vertex of the input mesh. Thus, for speed, we simply the initial mesh using Garland's method [Garland and Heckbert 1997] and before computing the global conformal parametrization.

## 4 T-splines

This section briefly reviews T-splines and their properties [Sederberg et al. 2003; Sederberg et al. 2004]. T-spline surfaces are a generalization of B-spline surfaces.

The control mesh for a T-spline surface is called a T-mesh. Figure 3 shows a pre-image of a T-mesh: it is a rectangular grid

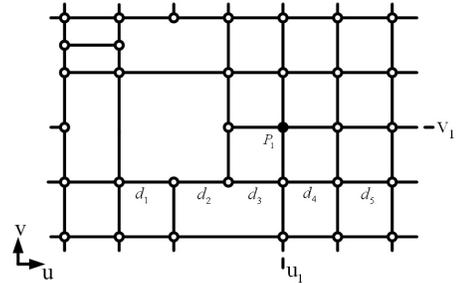


Figure 3: Pre-image of a T-mesh

allowing T-junctions. Each edge is a line segment along the  $u$  or  $v$  direction. A T-junction is a vertex shared by one  $u$ -edge and two

$v$ -edges, or vice versa. Each edge in a T-mesh is associated with a knot interval, constrained by the following rules:

**Rule 1:** The sum of the knot intervals on opposing edges of any face must be equal.  $\square$

**Rule 2:** If two T-junctions on opposing edges of a face can be connected without violating the previous rule, that edge must be included in the T-mesh.  $\square$

If a T-mesh does not contain any T-junctions, the corresponding T-spline degenerates to a B-spline surface.

The knot information is used to represent a T-spline surface:

$$P(u, v) = \sum_{i=1}^N w_i P_i B_i(u, v) / \sum_{i=1}^N w_i B_i(u, v), \quad (8)$$

where  $P_i = (x_i, y_i, z_i)$  are the  $N$  control points in  $R^3$  with associated weights  $w_i$ , and  $B_i(u, v)$  are blending functions defined in terms of cubic B-spline basis functions:

$$B_i(u, v) = N[u_{i0}, u_{i1}, u_{i2}, u_{i3}, u_{i4}](u) N[v_{i0}, v_{i1}, v_{i2}, v_{i3}, v_{i4}](v). \quad (9)$$

The knot vectors  $u_i = [u_{i0}, u_{i1}, u_{i2}, u_{i3}, u_{i4}]$  and  $v_i = [v_{i0}, v_{i1}, v_{i2}, v_{i3}, v_{i4}]$  are determined as follows:

**Rule 3:** Let  $(u_{i2}, v_{i2})$  be the knot coordinates of  $P_i$ . Consider a ray in parameter space  $R(\alpha) = (u_{i2} + \alpha, v_{i2})$ . Starting at  $P_i$ ,  $u_{i3}$  and  $u_{i4}$  are the  $u$  coordinates of the first two  $u$ -edges intersected by the ray going in the  $+\alpha$  direction, and  $u_{i1}$  and  $u_{i0}$  in the opposite direction. The  $v$  knots are found likewise.  $\square$

For example, in Figure 3, consider  $P_1$  with knot value  $u_1$ . Following a horizontal ray, we hit edges to the left giving knot values of  $u_1 - d_1 - d_2 - d_3$  and  $u_1 - d_3$ , and to the right, knot values of  $u_1 + d_4$  and  $u_1 + d_4 + d_5$ .

## 5 Surface fitting

We now consider how to find the optimal collection of T-splines minimizing the distance between the fitted surface and the triangular mesh, while capturing the geometric shape.

After creating an initial approximate T-spline surface, we carry out global and local optimization steps. The local steps are aimed at minimizing the  $L^\infty$  errors for positions and normals by local adjustment, while global optimization is used to ensure appropriate positioning of the control points overall and to remove unwanted ripples generated during the local approximation process.

Finding a good balance between global and local optimization is essential for achieving high quality surface; we use an automatic strategy to do so. Between two successive global approximation phases, several phases of local optimization are performed, during which any one face of the T-mesh may only be subdivided once. Faces generated in the local approximation phases are flagged, and if any face is to be subdivided again, we switch from local approximation back to global approximation. These two processes are iterated until the desired tolerance is reached after a global approximation step.

We now consider these steps in detail

### 5.1 Initial T-spline surface

Segmentation splits the simplified mesh into several rectangular patches. We define a T-spline over each patch such that continuity is preserved along their common boundary curves.

The initial T-splines are constructed in two steps. A topology step determines the structure of the T-mesh for each T-spline. Once the required knots for each T-mesh have been specified, a geometry step computes the Cartesian coordinates of each knot. Weights of control points in the initial T-splines are set to 1.

In the topology step, for each T-mesh, interior knots and boundary knots are distributed separately. Using the conformal parameterization, interior knots are distributed on a uniform rectangular grid. Boundary knots are distributed to preserve continuity along boundary curves and points. At ordinary boundary points and T-junctions, we follow the usual T-spline approach.  $C^2$  continuity is obtained at such points as they locally form a T-mesh. Boundary points arising near double covering fold points and zero points are treated specially. See 4, left (before and after folding) and right, respectively. Certain control points around these points are constrained to lie in the same plane. The red point is the point in question, and the black points show the surrounding knots which are constrained. This enforces  $G^1$  continuity at these locations.

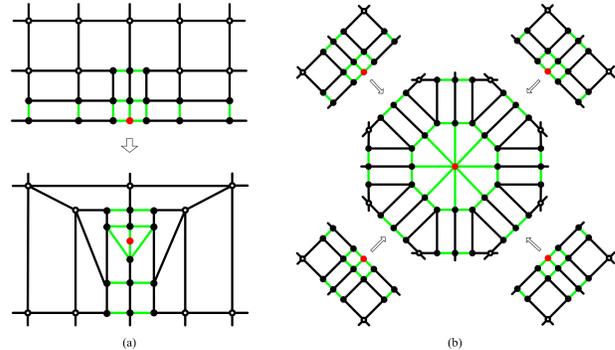


Figure 4: Knots near double covering fold points and zero points.

The geometry step is then posed as a constrained least-squares minimization problem. Under the continuity constraints mentioned above, we minimize the sum of squared distances between vertices in the simplified mesh and their images in the initial T-splines at the corresponding parameter values. The objective function is quadratic and the constraints linear, leading to a linear system.

### 5.2 Notation

Before discussing global and local approximation, we give the notation used. Let  $V = \{v_1, \dots, v_m\}$  and  $F = \{f_1, \dots, f_n\}$  denote the sets of vertices and faces of the input mesh  $M$ . We use  $(u_{v_i}, v_{v_i})$  to denote the conformal parameters,  $n_{v_i}$ , the normal direction, and  $v_i^u, v_i^v$ , the first-order derivatives of the discrete mesh in the  $u$  and  $v$  directions.  $S = \{S_i\}$  is the piecewise approximating T-spline surface and  $T = \{T_i\}$  is its T-mesh. The T-spline cells are collected in  $C = \{C_j\}$ . Certain samples are selected from each cell  $C_j$ , as explained later; let them be denoted by  $p_{jk}$ . The conformal parameters and their image points on the input mesh  $M$  are denoted by  $(u_{jk}, v_{jk})$  and  $q_{jk}$  respectively.

### 5.3 Global approximation

In this phase, we attempt to find the optimal approximation of  $M$  with a fixed T-mesh structure and knot values, allowing all control points in the T-mesh to move. First, we select samples from the approximating T-splines,  $S_i$ . We then find their images on the input mesh  $M$  using the global conformal parametrization. Finally, a weighted sum of position and first order derivative errors between the samples and their images is minimized.

#### 5.3.1 Sampling and weighting

We now discuss how we perform the sampling and weighting.

In each cell  $C_j$ , we select some samples  $p_{j1}, \dots, p_{jN_j}$ . There are many methods for sampling smooth surfaces based on curvature

criteria or budgets [Chhugani and Kumar 2003]. We simply uniformly sample the parameter domain of each cell  $C_j$ . Since we will adaptively subdivide any cell with complicated geometry and hence large approximation errors, we will thereby automatically use more samples in such cells.

The weight for each sample is determined as follows. Sample points in each cell are either *interior*, *boundary*, or *corner* samples. We assign a neighboring region to each sample  $p_{jk}$  by connecting it to the adjacent sample points in the same cell. The neighboring region thus comprises 4, 2 and 1 triangles for an interior sample, a boundary sample, and a corner sample, respectively. The area of this neighboring region is used as the weight for this sample.

### 5.3.2 Computing image points

The image point  $q_{jk}$  for each sample  $p_{jk}$  is computed by identifying its conformal parameters. In the parameter domain, we choose the face  $f_l$  that contains  $(u_{jk}, v_{jk})$ . Let the barycentric coordinates of  $(u_{jk}, v_{jk})$  with respect to the vertices of  $v_l^1, v_l^2, v_l^3$  of  $f_l$  be  $\alpha_{jk}^1, \alpha_{jk}^2, \alpha_{jk}^3$ .

The position and derivative information at  $q_{jk}$  are interpolated from those at the  $v_l^i$  using the  $\alpha_{jk}^i$ :

$$\mathbf{X}_{q_{jk}} = \alpha_{jk}^1 \cdot \mathbf{X}_{v_{jk}^1} + \alpha_{jk}^2 \cdot \mathbf{X}_{v_{jk}^2} + \alpha_{jk}^3 \cdot \mathbf{X}_{v_{jk}^3}, \quad (10)$$

where  $\mathbf{X}$  represents the collection of position and derivative information.

### 5.3.3 Optimization

We wish both to minimize the  $L^\infty$  norm for positional errors, and to achieve good approximation of normal information. There are two problems with incorporating normal approximation with position approximation. Firstly, normal vectors for parametric surfaces are not a linear function of their control points, making the optimization problem non-linear. Secondly, a tradeoff must be made between the error terms for positions and normals.

To avoid the first problem, instead of approximating the normal information, we use as a substitute approximating the first order derivatives. Thus, the normal error term we use for each sample is

$$E_{p_{jk}}^{nor} = \frac{\|S_u(u_{jk}, v_{jk}) - q_{jk}^u\|^2}{\|q_{jk}^u\|^2} + \frac{\|S_v(u_{jk}, v_{jk}) - q_{jk}^v\|^2}{\|q_{jk}^v\|^2}. \quad (11)$$

Having bounded  $E_{p_{jk}}^{nor}$  ensures bounded errors in normal.

The positional error term for each sample is simply defined as

$$E_{p_{jk}}^{pos} = (S(u_{jk}, v_{jk}) - q_{jk})^2. \quad (12)$$

The overall error for any sample is defined as

$$E_{p_{jk}} = E_{p_{jk}}^{pos} + \lambda_{jk} E_{p_{jk}}^{nor}, \quad (13)$$

Here  $\lambda_{jk}$  determines the relative importance of positional and normal errors during optimization. Initially,  $E_{p_{jk}}^{pos}$  is large, as each  $p_{jk}$  is far from its image point  $q_{jk}$ . Consequently,  $\lambda_{jk}$  should be made small, in order to ensure that we first meet the positional requirements. As we proceed,  $E_{p_{jk}}^{pos}$  becomes smaller, and the geometry approaches the required position, so we can pay more attention to the normal errors by increasing  $\lambda_{jk}$ . Ultimately, we need to make the normal terms compatible with the positional terms. Our implementation sets  $\lambda_{jk}$  to:

$$\lambda_{jk} = 10E_{jk} \exp(-E_{jk}^{pos}/h^2), \quad (14)$$

where  $h$  is a length relative to the length scale  $L$  of the bounding box of the whole model; we set it to  $h = 0.01L$ . Although these constants are empirically chosen, the method is stable to changes in these tuning parameters.

Figure 5 compares fitting using only positional error control, and fitting using a combination of positional and normal error control.

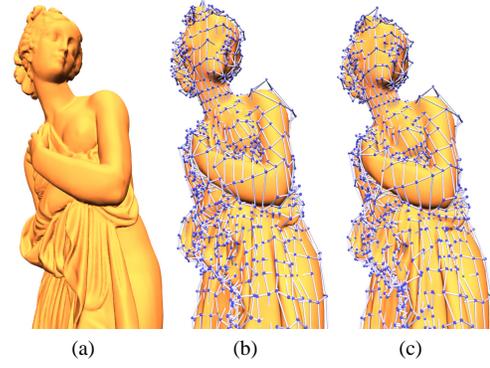


Figure 5: Iphigenia, 4000 control points: (a) mesh surface, (b) spline with position tolerance only, (c) with position and normal tolerance

Optimization is performed by minimizing the weighted sum of the error for each sample

$$E = \sum_{j,k} w_{jk} E_{p_{jk}}, \quad (15)$$

where  $w_{jk}$  are the weights explained earlier.

Because  $S, S_u$  and  $S_v$  are linear in the control points, the objective function is quadratic, and hence minimization requires solution of a linear system. As the T-spline basis has local support, this system is sparse. We efficiently solve it using the conjugate descent method.

## 6 Local approximation

Global approximation does not necessarily ensure bounded  $L^\infty$  error. Thus, after each global approximation step, we need to insert appropriate knots in regions where the approximation error remains large. Several local approximation steps are carried out between pairs of successive global approximation steps.

We now consider how to perform one local approximation step, by first finding those regions where the approximation quality is low and refinement is necessary, and then performing local optimization.

### 6.1 Refinement

We start by computing the approximation error of all cells at the current optimization stage. The approximation error can be defined using an  $L^\infty$  norm or an  $L^1$  norm. In practice, we find that the latter gives more stable results, especially during the initial steps. For each cell  $C_j$ , the mean  $L^1$  approximation error  $M_{C_j}$  is defined as:

$$M_{C_j} = \sum_k w_{jk} E_{p_{jk}} / \sum_k w_{jk}. \quad (16)$$

We refine the cell, denoted  $C_{max}$ , with the largest value of  $M_{C_j}$ . Sederberg et al [Sederberg et al. 2004] presented one approach for carrying out local refinement. Extra knots are added so that the initial surface retains the same geometry but is represented using more control points. Iterative fitting then starts, initialized with this new surface.

We prefer an alternative approach, which avoids iteration and keeps the number of additional control points low. We simply subdivide  $C_{max}$  into four subcells. Where this results in new points on the old cell boundary, a check must be made if additional connections are needed across neighboring cells. Figure 6 illustrates such a case. A cell in the upper left diagram is subdivided as shown on the lower left; the additional edge colored red is also required, and added. We then compute the basis functions for the knots near

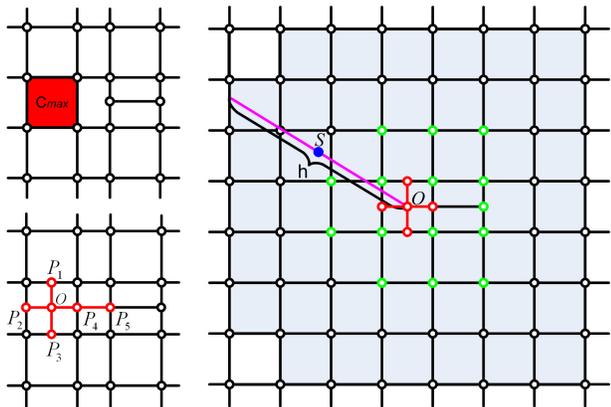


Figure 6: Local refinement

the local refinement performed. Let  $P = \{P_i\}$  denote all control points whose basis function have changed, and let  $C = \{C_j\}$  denote all rectangles that have control points in  $P$ . The right hand side of Figure 6 illustrates  $P$  and  $C$  for this example. The red points are inserted points, additional points in  $P$  are colored green, and the affected cells  $C$  are colored gray.

## 6.2 Optimization

We now perform local optimization by allowing the positions of the control points in  $P$  to move.

We use samples taken uniformly from the cells of  $C$  to form the objective function. Samples in those cells of  $C$  that have only one or two control points in  $P$  must be given lower weights: as such cells have few degrees of freedom, minimization without such weighting would tend to move these control points too far, have a negative impact on the quality of fit of cells surrounding  $C_{max}$ . Thus, we weight the samples with respect to their relative distance from the center of the local refinement. Let  $o$  be the center of this refinement, and  $h$  measure the radius of  $C$  in the direction from  $o$  to the current sample  $S$  at  $p_{jk}$  (see the right hand side of Figure 6). The weight for this sample is set to:

$$W(p'_{jk}) = W(\|p'_{jk} - o\|/h), \quad (17)$$

where  $W(\cdot)$  is a function defined over  $[0, \infty)$  with support  $[0, 1]$ . We use the cubic B-spline basis function:

$$w(t) := \begin{cases} (1-t)^3 & 0 \leq t \leq 1 \\ 0 & t > 1 \end{cases}. \quad (18)$$

The objective function for local approximation is the weighted sum of error terms of the samples  $p'_{jk}$ :

$$E_{local} = \sum_{p_{jk} \in C'_j \in C} w_{jk} W(p'_{jk}) E_{p'_{jk}}. \quad (19)$$

This objective function is also quadratic in the positions of the control points, and can be optimized by solving a linear system. This linear system is much smaller than the one used in global approximation.

Model	$L^\infty(\mathbf{p})$	$L^\infty(\mathbf{n})$	Mesh	T-spline	Time
Fig. 5	0.2%	0.1	300k	4000	456s
Fig. 7	0.15%	0.1	80k	5000	106s
Fig. 8	0.1%	0.1	120k	6800	870s
Fig. 9	0.1%	0.1	60k	2400	183s

## 7 Results

First, we compared our algorithm with the subdivision surface approximation method in [Marinov and Kobbelt 2004]. At a comparable approximation quality to the one they report for the rocker arm model, our method takes about 1/2 of the time. See Figure 7. Moreover, our method requires no choice of initial positions.

We now show further results using of our method. The goal was to achieve high-quality approximation of position and normals.

Tests using head of Max Planck's (Figure 9) and David's (Figure 8) show the ability of our approach to capture intricate geometry.

Table 7 gives the user selected position and normal tolerances, the number of vertices in the triangulation, the number of control points in the final mesh, and the time taken to compute these results. Position errors are as a percentage of the size of the diagonal of the bounding box of the model. Normal errors are measured as  $\text{mean}(\|\mathbf{n}_T - \mathbf{n}_S\|)$ , where  $\mathbf{n}_T$  and  $\mathbf{n}_S$  are normals of the triangulation and the fitted surface; averaging is done over each cell.

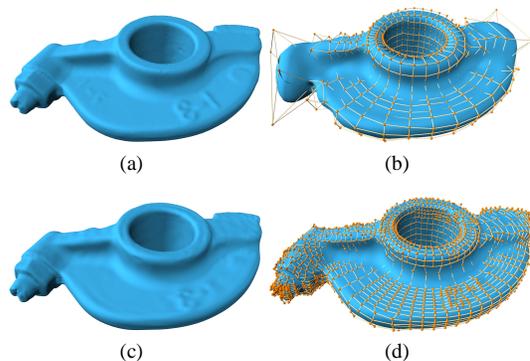


Figure 7: Rocker arm: (a) mesh, 80k triangles, (b) initial T-spline, (c) final T-spline, (d) overlaid T-mesh.

Our results shows the ability of our method to rapidly and automatically convert complex mesh geometries to splines with few control points.

## 8 Conclusions

We have given an easy-to-use and efficient framework for automatically converting surface meshes of arbitrary topology into T-spline surfaces. Our approach depends only on user-specified position and normal tolerances, and no other user input. Our method provides high quality approximation in a short time.

## References

- ANDERSSON, E., ANDERSSON, R., BOMAN, M., ELMROTH, T., DAHLBERG, B., AND JOHANSSON, B. 1988. Automatic construction of surfaces with prescribed shape. *CAD* 20, 6, 317–324.
- BERNARDINI, F., AND RUSHMEIER, H. 2002. State of the art reviews: The 3d model acquisition pipeline. *Computer Graphics Forum* 21 (June), 240–251.

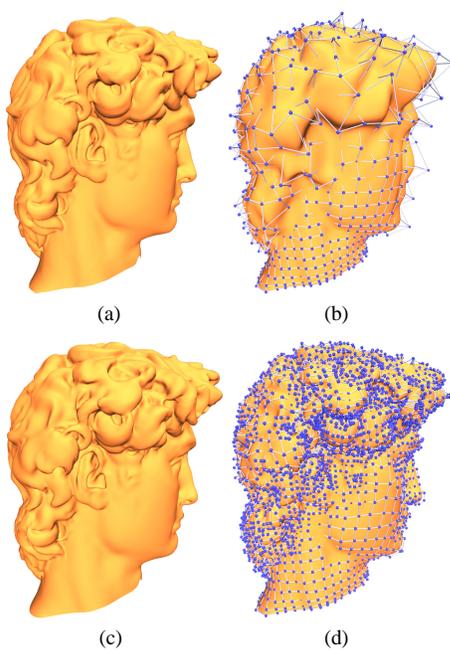


Figure 8: Head of David: (a) mesh, 100k triangles, (b) initial T-spline, (c) final T-spline, (d) overlaid T-mesh.

C. GOTSMAN, X. GU, A. S. 2003. Fundamentals of spherical parameterization for 3d meshes. In *SIGGRAPH 2003*, vol. 22, 358–363.

CHAM, T.-J., AND CIPOLLA, R. 1999. Automated b-spline curve representation incorporating mdl and error-minimizing control point insertion strategies. *IEEE PAMI 21*, 1, 49–53.

CHENG, D., WANG, W., QIN, H., WONG, K., YANG, H., AND LIU, Y. 2004. Fitting subdivision surfaces to unorganized point data using sdm. In *12th Pacific Graphics Conf.*, 16–24.

CHHUGANI, J., AND KUMAR, S. 2003. Budget sampling of parametric surface patches. In *SI3D'03: Proc. 2003 Symp. Interactive 3D Graphics*, 131–138.

DESBRUN, M., MEYER, M., AND ALLIEZ, P. 2002. Intrinsic parameterizations of surface meshes. In *Eurographics 2002*, vol. 12, 209–218.

DODGSON, N., IVRISSIMTZIS, I., AND SABIN, M. 2003. In *Curve and Surface Fitting: Saint-Malo 2002*, Nashboro Press, St Malo, France, A. Cohen et al., Eds., vol. 2, 119–128.

DONG, S., KIRCHER, S., AND GARLAND, M. 2005, to appear. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. In *CAGD*.

DUCHAMP, T., CERTAIN, A., DEROSE, A., AND STUETZLE, W. 1997. Hierarchical computation of PL harmonic embeddings. In *Tech. Report, Univ. Washington*.

ECK, M., AND HOPPE, H. 1996. Automatic reconstruction of b-spline surfaces of arbitrary topological type. In *SIGGRAPH 1996*, 325–334.

ECK, M., DEROSE, T., DUCHAMP, T., HOPPE, H., LOUNSBERY, M., AND STUETZLE, W. 1995. Multiresolution analysis of arbitrary meshes. In *SIGGRAPH 1995*, 173–182.

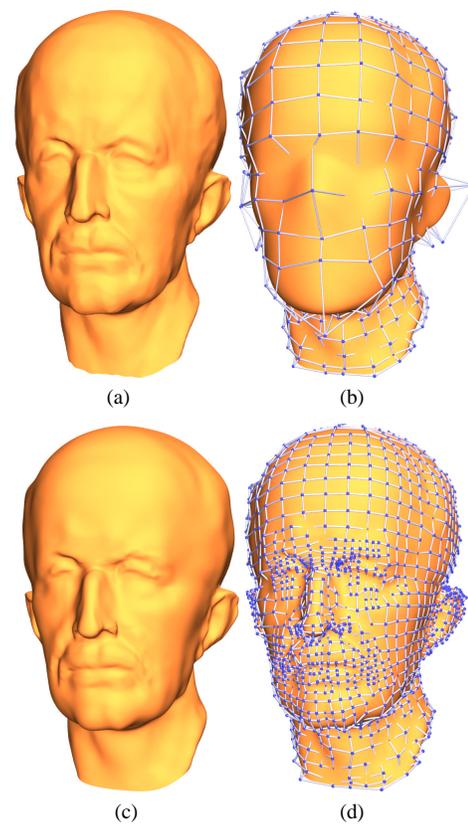


Figure 9: Max Planck's head: (a) mesh, 50k triangles, (b) initial T-spline, (c) final T-spline, (d) overlaid T-mesh.

FLOATER, M. S., AND HORMAN, K. 2004. Surface parameterization: a tutorial and survey. In *Multiresolution in Geometric Modelling*, Springer, N. A. Dodgson, M. S. Floater, and M. A. Sabin, Eds.

FLOATER, M. S. 1997. Parametrization and smooth approximation of surface triangulations. *CAGD 14*, 3, 231–250.

FLOATER, M. S. 2003. Mean value coordinates. *CAGD 20*, 1, 19–27.

GARLAND, M., AND HECKBERT, P. S. 1997. Surface simplification using quadric error metrics. In *SIGGRAPH 1997*, 209–216.

GU, X., AND YAU, S.-T. 2003. Global conformal surface parameterization. In *Proc. Eurographics/SIGGRAPH Symp. Geometry Processing*, Eurographics Association, 127–137.

GU, X., WANG, Y., CHAN, T. F., THOMPSON, P. M., AND YAU, S.-T. 2004. Genus zero surface conformal mapping and its application to brain surface mapping. In *IEEE Trans. Medical Imaging*, vol. 23, 949–958.

HALSTEAD, M., KASS, M., AND DEROSE, T. 1993. Efficient, fair interpolation using catmull-clark surfaces. In *SIGGRAPH 1993*, 35–44.

JIN, M., WANG, Y., YAU, S.-T., AND GU, X. 2004. Optimal global conformal surface parameterization. In *IEEE Visualization 2004*, IEEE Computer Society, 267–274.

JOST, J. 2000. *Compact Riemann Surfaces*. Springer.

- KATZ, S., AND TAL, A. 2003. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM TOG* 22, 3, 954–961.
- KRISHNAMURTHY, V., AND LEVOY, M. 1996. Fitting smooth surfaces to dense polygon meshes. In *SIGGRAPH 1996*, 313–324.
- LEE, A., MORETON, H., AND HOPPE, H. 2000. Displaced subdivision surfaces. In *SIGGRAPH 2000*, 85–94.
- LÉVY, B., PETITJEAN, S., RAY, N., AND MAILLOT, J. 2002. Least squares conformal maps for automatic texture atlas generation. In *SIGGRAPH 2002*, vol. 21, 362–371.
- LITKE, N., LEVIN, A., AND SCHRÖDER, P. 2001. Fitting subdivision surfaces. In *VIS '01*, IEEE Computer Society, 319–324.
- MARINOV, M., AND KOBELT, L. 2004. Optimization techniques for approximation with subdivision surfaces. In *Symp. Solid Modeling and Applications*, 113–122.
- NI, X., JOHN, M., AND HART, J. 2004. Fair morse functions for extracting the topological structure of a surface mesh. In *SIGGRAPH 2004*, vol. 23, 613–622.
- PINKALL, U., AND POLTHIER, K. 1993. Computing discrete minimal surfaces and their conjugate. In *Experimental Mathematics* 2 (1), 15–36.
- POTTMANN, H., AND LEOPOLDSEDER, S. 2003. A concept for parametric surface fitting which avoids the parametrization problem. *CAGD* 20, 6, 343–362.
- SAPIDIS, N. S. 1994. *Designing Fair Curves and Surfaces: Shape Quality in Geometric Modeling and Computer-Aided Design*. SIAM Press.
- SCHOEN, R., AND YAU, S.-T. 1997. *Lectures on Harmonic Maps*. International Press.
- SEDERBERG, T. W., ZHENG, J., BAKENOV, A., AND NASRI, A. 2003. T-splines and t-nurccs. *ACM TOG* 22, 3, 477–484.
- SEDERBERG, T. W., CARDON, D. L., FINNIGAN, G. T., NORTH, N. S., ZHENG, J., AND LYCHE, T. 2004. T-spline simplification and local refinement. *ACM TOG* 23, 3, 276–283.
- SHEFFER, A., AND STURLER, E. 2001. Parameterization of faceted surfaces for meshing using angle-based flattening. In *Engineering with Computers*, vol. 17, 326–337.
- SIEGEL, C. L. 1957. *Lectures on quadratic forms*. 7. Tata Institute of Fundamental Research, Bombay.
- WEISS, V., ANDOR, L., RENNER, G., AND VÁRADY, T. 2002. Advanced surface fitting techniques. *CAGD* 19, 1, 19–42.
- YANG, H., WANG, W., AND SUN, J. 2004. Control point adjustment for b-spline curve approximation. *CAD* 36, 7 (June), 639–652.

## Appendix

Details of Riemann surface theory used in this paper can be found in [Jost 2000], other results are in [Siegel 1957; Schoen and Yau 1997]. We summarize the main ideas used.

First we informally explain charts and atlases. A chart maps part of a surface, topologically equivalent to an open disk, to the plane. An atlas is a collection of overlapping charts which cover a surface, and the transition map says how the planar coordinates are related between the two charts.

**Definition 1** A complex function  $\phi : C \rightarrow C$ , where  $C$  indicates the complex plane,  $\phi : (x, y) \rightarrow (u, v)$  is holomorphic if it satisfies the Cauchy-Riemann equations:  $\partial u / \partial x = \partial v / \partial y$ ,  $\partial u / \partial y = -\partial v / \partial x$ .

**Definition 2** An atlas on a surface  $S$  with charts  $z_\alpha : U_\alpha \rightarrow C$  is called conformal if the transition maps  $z_\beta \circ z_\alpha^{-1} : z_\alpha(U_\alpha \cap U_\beta) \rightarrow z_\beta(U_\alpha \cap U_\beta)$  are holomorphic.

**Definition 3** Two conformal atlases are equivalent if their union is still a conformal atlas. Each equivalence class of conformal atlases is called a conformal structure. A Riemann surface is a surface together with a conformal structure.

**Theorem 1** All oriented metric surfaces are Riemann surfaces, and the metric on each conformal chart can be represented in the form  $ds^2 = \lambda^2(u, v)(du^2 + dv^2)$ ,  $(u, v)$  are the local coordinates.

**Definition 4** A holomorphic differential form [Jost 2000]  $\omega$  is a complex differential form, such that for each local coordinate  $z_\alpha$ ,  $\omega$  can be represented as  $\omega = f(z_\alpha)dz_\alpha$ . Point  $p$  is called a zero if  $f(z_\alpha)$  is zero.

**Definition 5** Let  $S$  be a Riemann surface, and  $\omega$  be a holomorphic 1-form on  $S$ . A horizontal trajectory is a curve on  $S$  along which  $\omega$  is real, and a vertical trajectory is a curve on  $S$  along which  $\omega$  is imaginary.

The definitions of zero points and horizontal and vertical trajectories are independent of the choice of local coordinates. If a trajectory starts from a zero point, it will end at a zero point or intersect the boundary. Zero points are also called *zero points* in this paper.

The intersecting horizontal and vertical trajectories form the *conformal net*, which locally has a tensor product structure. Its global structure is described by the following theorem:

**Theorem 2** Let  $S$  be a closed Riemann surface with genus  $g > 1$ , and let  $\omega$  be a holomorphic one-form. The horizontal trajectories through the zero points of  $\omega$  partition  $S$  into cylinders, each of which can be conformally mapped to a rectangle by integrating  $\omega$ .