ELSEVIER

Computer Aided Geometric Design 18 (2001) 463–481

COMPUTER AIDED GEOMETRIC DESIGN

www.elsevier.com/locate/comaid

Trimming for subdivision surfaces

Nathan Litke [a,*], Adi Levin [b], Peter Schröder [a]

[a] *Caltech, Department of Computer Science, 1200 E California Boulevard, Pasadena, CA 91125, USA*
[b] *Tel Aviv University, Tel Aviv 69978, Israel*

Received June 2000; revised February 2001

Abstract

Trimming is an important primitive operation in geometric modeling. It is also the root of many numerical and topological problems in modern NURBS based CAGD systems. In this paper we introduce a new method for trimming subdivision surfaces. It is based on the use of *combined subdivision* schemes to *guarantee* exact interpolation of trim curves. The latter ensures, for example, that if two surfaces share a trim curve, they will meet exactly at the trim curve. In contrast to traditional approaches to trimming (e.g., for NURBS) we construct a new control mesh with each trim operation. This causes a perturbation of the surface near the trim region, which we control through the use of multiresolution details. These are computed rapidly and at low cost with the help of a novel set of quasi-interpolation operators. We demonstrate our algorithm with a number of examples.

Keywords: Subdivision surfaces; Combined subdivision; Trimming; Boundary interpolation; Approximation

1. Introduction

Subdivision surfaces are a very attractive alternative to classical NURBS (non-uniform rational B-spline) patches for free form geometric modeling. They are used increasingly in high end animation production (e.g., Pixar (DeRose et al., 1998)), game engines, and are provided as primitives in many popular modeling programs (e.g., Maya, Mirai, 3D Studio Max, LightWave, etc.). One of the chief advantages and distinguishing features of subdivision is its ability to model arbitrary topology (piecewise) smooth surfaces. In

* Corresponding author.
E-mail addresses: njlitke@cs.caltech.edu (N. Litke), adilev@math.tau.ac.il (A. Levin), ps@cs.caltech.edu (P. Schröder).

0167-8396/01/$ – see front matter
PII: S0167-8396(01)00042-5

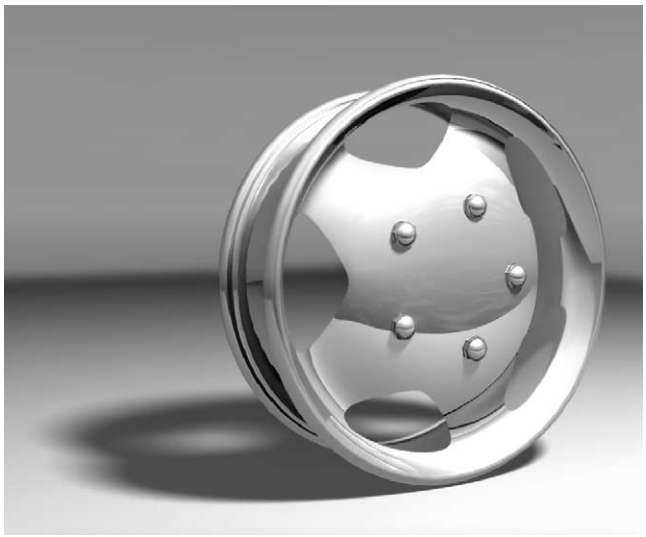


Fig. 1. Example of a subdivision surface after trimming with our procedure.

contrast, classical NURBS methods require careful patch layout and cumbersome cross boundary continuity management to build complex shapes. In practice this often leads to the appearance of kinks and cracks, especially when such patch complexes are animated.

Trimming is an important component of any free form surface modeling system. It is essential for boolean operations and arises even in as simple a context as "punching" a hole into the surface. In the case of NURBS a trimming operation does not change the domain of a patch, but rather identifies a region within the domain whose evaluation is skipped, creating the associated "hole". This requires the computation of the pre-image of the trim curve in the parametric domain. Such computations are notoriously difficult (Krishnan and Manocha, 1997; Keyser et al., 1999a, 1999b) and require potentially very expensive machinery to avoid topological inconsistencies such as cracks (see Section 1.1 for a more detailed discussion). Instead we assume a different point of view. Whenever a region of the subdivision surface is cut in a trimming operation we construct a new domain such that the resulting subdivision surface approximates the original surface away from the trim region. The approximation error away from the trim curve can be made arbitrarily small. The region over which it is non-zero can be made arbitrarily small as well. The trimming curve itself is interpolated *exactly* through the use of combined subdivision (Levin, 1999a, 1999b), avoiding any potential topological problems at the curve itself. Because combined subdivision schemes work directly with curves in world space we do not need to compute exact pre-images of trim curves.[1] This avoids the usual problems and costs associated with this operation in the traditional approach to trimming, and is one of the principal advantages of our method.

[1] *Approximate* pre-images are still required to decide how to change the control mesh.

1.1. Related work

1.1.1. Trimming NURBS patches

Trimming NURBS patches is a notoriously difficult operation and is to this day one of the weak links in even high end commercial CAGD systems. Trimming of NURBS patches is performed by identifying pre-images of desired world space trim curves in parameter space. These parameter space curves bound regions for which the patch evaluation is skipped. The original coefficients which define the patch are unchanged, which ensures that the trimmed surface is coincident with the original surface. The main difficulty in this approach is the reliable computation and representation of the trim curve pre-images. If the trim curve arises from the intersection of patches, the algebraic and combinatorial structure (loops) of such curves can be extremely complicated even for low degree polynomials. This can be addressed with hybrid methods employing symbolic and numerical techniques (Krishnan and Manocha, 1997) or methods which are mostly symbolic and use exact algebraic number representations (Keyser et al., 1999a, 1999b). The latter tend to be very slow, but they guarantee accurate results, while the former are very difficult to make completely robust in all cases. If the trim curve is defined in terms of other primitives (e.g., involving offsets) symbolic methods may not be applicable and the only recourse is to numerical methods involving approximations of the trim curves. Such scenarios almost invariably lead to topological problems such as cracks in the resulting surfaces.

The main distinction between these approaches and ours is that we define a new surface whose boundary is the *exact* trim curve, whatever form it might take. The trade-off is that the trimmed surface near the trim curve only approximates the original surface. However, as will be seen later, this approximation can be made arbitrarily close to the original. Because the trim boundary is exact, topological problems such as cracks at the trim curve are entirely avoided.

1.1.2. Boundary constructions for subdivision surfaces

Boundary constructions for subdivision surfaces have been described by a number of authors (Hoppe et al., 1994; Schweitzer, 1996; Biermann et al., 2000). All these constructions require that the boundary curve be a spline curve. This is not sufficient for the representation of arbitrary trim curves, which are generally not splines. Instead we make use of combined subdivision schemes (Levin, 1999a, 1999b). Combined subdivision enables the construction of subdivision surfaces with arbitrary boundary curves so long as the boundary curve is (I) piecewise smooth (meaning that the curve consists of segments with Hölder continuous second derivatives); (II) parameterized; and (III) possesses an evaluation procedure. The resulting surfaces are guaranteed to exactly interpolate the desired curve (transfinite interpolation). Such schemes only require the modification of subdivision stencils near the boundary in a straightforward way using data from the supplied boundary curve.

Our subdivision surface trimming procedure is based on the use of a novel combined subdivision scheme which extends the classic Loop scheme (Loop, 1987). This ensures that we interpolate the desired trim curve exactly. However, since the control mesh changes topologically we must also compute appropriate control point positions, and more

generally, detail offset vectors throughout the subdivision hierarchy, so that the trimmed surface fits the original surface to a desired accuracy. For this purpose we introduce a new quasi-interpolation operator which is optimal in the regular setting.

1.2. Contributions

The main contribution of this paper is a trimming procedure for subdivision surfaces which guarantees exact interpolation of the desired trim curve. In particular this implies that if two surfaces share the same trim curve in world space (for example, their intersection), the two surfaces will meet exactly at the trim curve and topological inconsistencies, such as cracks during tessellation, are easily avoided. The ultimate goal of this work is to provide tools for CAGD applications which require exactness. This may be contrasted with entertainment applications in which the exact shape of a trim curve may not be as stringent a requirement.

The development of our algorithm contains a number of additional innovations. We describe

- a novel combined subdivision scheme for Loop surfaces;
- quasi-interpolation operators for Loop surfaces;
- a procedure to establish correspondence between two subdivision surfaces given by control meshes with different connectivity;
- an adaptive control mesh remeshing procedure;
- an adaptive error control procedure based on a multiresolution surface representation with detail coefficients.

We emphasize that our algorithm does not address the full range of issues associated with boolean operations on subdivision surfaces. In particular we do not consider the problem of computing surface–surface intersections in the general setting. We assume that the desired world space trim curve is given to the algorithm. It is not required to lie exactly *on* the surface, although in practice this is generally the case. Similarly, the trim curve itself is not required to be a spline curve. Rather, the only requirement is that it be piecewise smooth, parameterized, and possess an exact evaluation procedure.

1.3. Overview

In the following sections, we develop an algorithm for trimming subdivision surfaces. The result of applying this algorithm is a new surface with the trim region removed. Since the subdivision schemes found in the literature are not suitable for accurately describing trimmed surfaces, we present a new subdivision surface representation. We use a combined subdivision scheme for boundaries to guarantee the exact interpolation of the trim curve. However, due to the change in topology, the trimmed surface does not match the original surface near the trim curve. We correct this discrepancy by introducing multiresolution detail offset vectors into the subdivision hierarchy of the new surface. These vectors modify the control points generated by subdivision and are computed using local quasi-interpolation operators in a final approximation stage of our algorithm. The fit is adaptive, producing detail vectors only where necessary to guarantee that the trimmed surface is

within a prescribed tolerance of the original surface. The resulting surface is amenable to further modeling, including successive trimming operations.

We will begin by describing the representation for the surfaces produced by our algorithm. We elected to develop our algorithm for surfaces based on Loop's scheme. However, the construction is similar for other popular subdivision schemes, such as Catmull–Clark. In the following sections, we discuss the stages of the algorithm which construct a trimmed surface of this type.

2. Surface representation

In this section we describe a novel subdivision scheme that is based on Loop's scheme and multiresolution details. It is the basic parametric primitive that we use in the construction of trimmed surfaces. In addition, we present novel quasi-interpolation operators which allow us to compute detail coefficients for approximation.

2.1. Combined Loop subdivision

Our subdivision scheme is based on Loop's scheme (Loop, 1987), which generalizes quartic box splines to the arbitrary topology surface setting. Loop's scheme is defined over a closed control polyhedron by the subdivision stencils illustrated in Fig. 2. We will refer to a control point i at level j in the subdivision hierarchy with the label p_i^j. The vertex stencil is parameterized by the number of edges incident to the control point, known as its *valence*. The limit position of a control point p_i^j is denoted by Lp_i^j and is computed with the limit stencil in Fig. 2.

We account for surfaces with boundaries by using *combined subdivision*, where the underlying surface is represented by a control polyhedron and parametric curves. Since boundary curves may be piecewise smooth, we distinguish between smooth boundary control points and corner control points. The latter may have any valence, while the former are restricted to valence four only. On the boundary, control points p_i^j are computed using a boundary curve $c : [0, 1] \to \mathbb{R}^3$ and associated parameter values u_i^j. The u_i^0 are supplied by

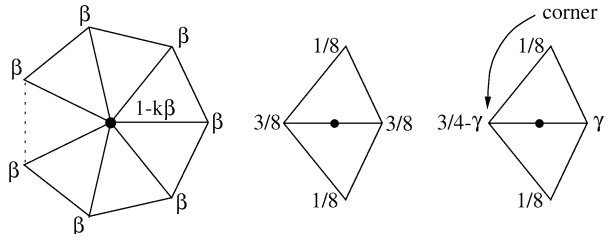


Fig. 2. Loop subdivision rules for the interior of surfaces. The rule for vertices, ordinary edges and edges adjacent to a corner (see the text for the weight γ) are shown on the left, middle and right, respectively. In the vertex subdivision stencil, $\beta = \alpha/(8k)$, with $\alpha = 5 - (3 + 2\cos(2\pi/k))^2/8$ and k the valence of the center vertex. The limit position stencil has the same form with $\beta = \alpha/(k(3+\alpha))$.

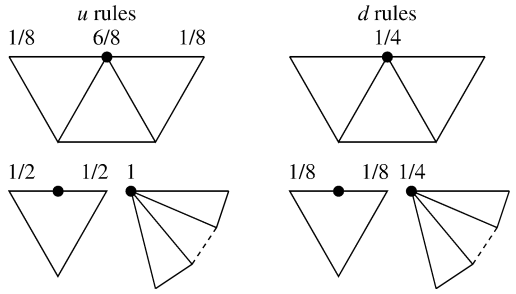


Fig. 3. At the boundary/corner (u, d) values are propagated to finer levels as indicated by these stencils.

the user, or assigned by our algorithm when a surface is trimmed (see Section 3.1). To allow for unevenly spaced parameter values, we use uniform cubic B-spline subdivision rules to obtain a sufficiently smooth parameterization. Thus for consecutively indexed control points on the boundary,

$$p_i^j = c\big(1/6\big(u_{i-1}^j + 4u_i^j + u_{i+1}^j\big)\big).$$

Note that this simplifies to $p_i^j = c(u_i^j)$ for $u_{i+1}^j - u_i^j = \text{const}$. Corner control points are defined as $p_i^j = c(u_i^j)$. The update rules for u_i^j are given in Fig. 3. It follows that control points on the boundary are at their limit position, i.e., $Lp_i^j = p_i^j$.

To handle convex as well as concave corners correctly, we use a modification of our subdivision rules that is similar to (Biermann et al., 2000). An interior edge adjacent to a corner control point p_c^j uses a modified stencil as shown in Fig. 2 (right), with parameter $\gamma = 1/2 - 1/4\cos(\theta_k)$. For convex corners we define $\theta_k = \phi/(k-1)$ and for concave corners $\theta_k = (2\pi - \phi)/(k-1)$, where k is the valence and ϕ is the angle made by the one sided tangent vectors at the corner $c(u_c^0)$. Let $\tilde{p}_i^{j+1}$ be the control point computed with this stencil, and let $\tilde{p}^\perp$ be its projection in the tangent space at $c(u_c^0)$. We control the rate of convergence near the corner by introducing a *flatness parameter* s:

$$p_i^{j+1} = (1-s)\,\tilde{p}_i^{j+1} + s\big((1-t)\,p_c^{j+1} + t\,\tilde{p}^\perp\big)$$

with $s = 1 - 1/(4\gamma + \cos(\pi/(k-1)))$ and $t = \|\tilde{p}_i^{j+1} - p_c^{j+1}\|/\|\tilde{p}^\perp - p_c^{j+1}\|$. This value of s is necessary (but not sufficient) for C^2 continuity.

2.2. Multiresolution detail

The trimmed surfaces that we produce must contain a sufficient number of degrees of freedom for approximation. We can capitalize on the multiresolution structure of our surfaces by storing a finite set of detail vectors in the subdivision hierarchy. Each detail vector d_i^j produces a perturbation of the surface near its corresponding control point p_i^j, which gives us the necessary degrees of freedom. A zero detail vector has no effect on the surface, so we only store the non-zero details in a sparse data structure. Typically, the

Fig. 4. Influence of the vector valued detail d at an exemplary boundary control point. The detail is zero on the left, and non-zero for the middle and right, as illustrated. Note the influence on the surface shape near the boundary.

surface for trimming provided by the user contains no details, but in subsequent trimming operations the surface includes details computed for approximation.

In this new regime, letting S denote the entirety of all subdivision rules, we have $\boldsymbol{p}^{j+1} = S(\boldsymbol{p}^j + \boldsymbol{d}^j)$, $j \geqslant 0$. The detail coefficients d_i^j on the boundary are related to the (one sided) second derivatives of the surface at the boundary point Lp_i^j,

$$d_i^j = 1/6\left(\frac{\partial^2}{\partial u^2} + \frac{\partial^2}{\partial u \partial v} + \frac{\partial^2}{\partial v^2}\right) S(u,v)\Big|_{Lp_i^j},$$

where $S(u,v)$ denotes the surface parameterized by (u,v), with u along the boundary curve and v transverse to the curve. Unlike the detail at interior control points, boundary detail vectors are subtracted from the control point prior to subdivision (see Fig. 4 for examples showing the effect of boundary detail coefficients on the surface near the boundary). The update rules for boundary detail coefficients in Fig. 3 show that the details decay at each subdivision step. Any additional details computed for finer levels are additive. From this definition, it is clear that the boundary control points converge to samples of the curve, thus in the limit exactly interpolating the boundary curve everywhere independent of the exact form of the curve. If two different surfaces share boundary data they will both sample the curve in exactly the same locations ensuring a seamless tessellation.

2.3. Quasi-interpolation

To approximate a surface by subdivision, we must construct a control mesh with a set of control point positions and detail vectors chosen so that the associated limit surface approximates the given surface to within some user selected bound $\varepsilon > 0$. The approximation algorithm chooses the detail vectors at the control points for a prescribed connectivity. There are many possible ways to do this, for example through least squares fitting (Hoppe et al., 1994) or through interpolation constraints (Halstead et al., 1993). Either one of those approaches requires the solution of linear systems. Instead we opt for a purely local approach based on quasi-interpolation.

To develop this idea we begin by considering the regular setting, i.e., all control points have valence six and the surface consists of quartic box splines. Given a regular mesh, which samples the desired surface A in points $a_i \in A \subset \mathbb{R}^3$, the usual interpolation problem is: find $\boldsymbol{p}$ such that $\boldsymbol{a} = L\boldsymbol{p}$. $\boldsymbol{a}$ and $\boldsymbol{p}$ are the vectors of samples and control points

respectively, while L contains the limit evaluation masks for each of the control points (Fig. 2, left). Ignoring rank questions for the moment the interpolation problem may be solved as $\boldsymbol{p} = L^{-1}\boldsymbol{a}$. In general L^{-1} is dense making this approach expensive.

Quasi-interpolation circumvents this problem by requiring the inversion of L only on the space of polynomials up to some order. The natural choice in our case is the space of all cubic polynomials, π_3, since this is the highest order polynomial space entirely contained in the span of quartic box splines. In this case the quasi-interpolation operator $Q = (L|_{\pi_3})^{-1}$ is sparse with the same structure as the vertex subdivision stencil (Fig. 5(a)). Q is not unique, but can be chosen to have local support and a low norm. For the regular setting we have the following properties of our quasi-interpolation operator Q:

- if the a_i sample a cubic surface A, then $\boldsymbol{p} = Q\boldsymbol{a}$ yields control points which *reproduce* the cubic surface;
- for an arbitrary surface A, $LQ \neq I$; i.e., quasi-interpolation applied to $\boldsymbol{a}$ does not reproduce A.

Because of the latter property we will apply Q only to samples of the *difference* between the desired surface and the surface induced by our control polyhedron. Specifically, let a_i^j be samples of the desired surface, one associated with each control point p_i^j of the control mesh and all its refinement levels. Defining differences as $\Delta^j = \boldsymbol{a}^j - L\boldsymbol{p}^j$, the details are given by $\boldsymbol{d}^j = Q\Delta^j$.

Q as given in Fig. 5(a) is only optimal in the above sense for $k = 6$. At the coarsest level of the control mesh, where possibly all interior vertices are irregular, we ignore this distinction. This is justified by the fact that *asymptotic* approximation properties are not influenced by whatever steps we take at the coarsest level. In practice we have found $\beta = -1/(2k)$ to work well even when $k \neq 6$. At finer levels almost all vertices will be regular and Q as given above will be optimal. For the few irregular vertices at finer levels we modify Q to be interpolating rather than quasi-interpolating (Fig. 5(d)), as experimental evidence suggests this gives a better approximation near extraordinary vertices. Note that for Q to be interpolating, it is applied *after* the surface is modified by the quasi-interpolating stencils.

The Q stencils for boundary/corner control points are shown in Fig. 5. The support of the boundary stencil (Fig. 5(b)) can be reduced by removing the coefficients 8 and -8.

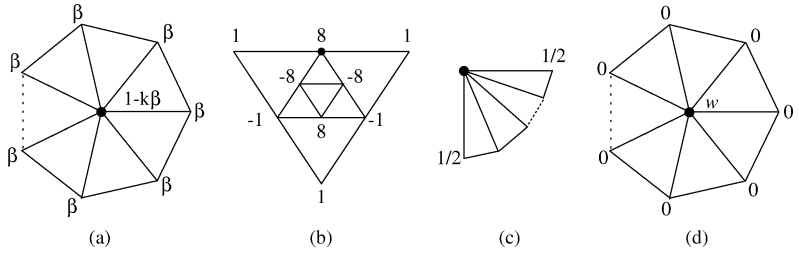


Fig. 5. Detail stencils for (a) interior, (b) boundary, (c) corner and (d) irregular vertices at levels $j > 0$. For interior vertices, stencil (a) with $\beta = -1/(2k)$ is quasi-interpolating and stencil (d) with $w = (3+\alpha)/3$ (see Fig. 2) is interpolating. Note that the corner stencil (c) is applied to detail vectors rather than differences.

The resulting stencil only has quadratic precision, but is more efficient and works well in practice. We do not have a rule to compute proper corner details. Instead the corner stencil (Fig. 5(c)) is applied not to differences, but rather to the detail vectors calculated at the adjacent boundary control points (or zero for an adjacent corner).

Fig. 6 shows an example of these operators used to approximate one octant of a sphere. The top left shows the control mesh for a combined subdivision surface whose boundary is given by three spherical arcs which bound an octant of a sphere 1m in diameter. The associated limit surface exactly interpolates the boundary curves, but is otherwise

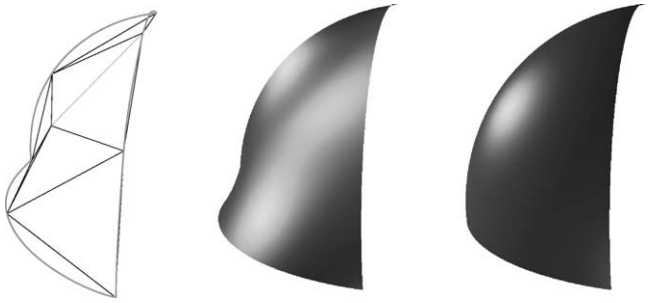


Fig. 6. Control mesh (left) and associated limit surface (middle). On the right the result of adding detail coefficients to ensure that the resulting surface matches an octant of a sphere.

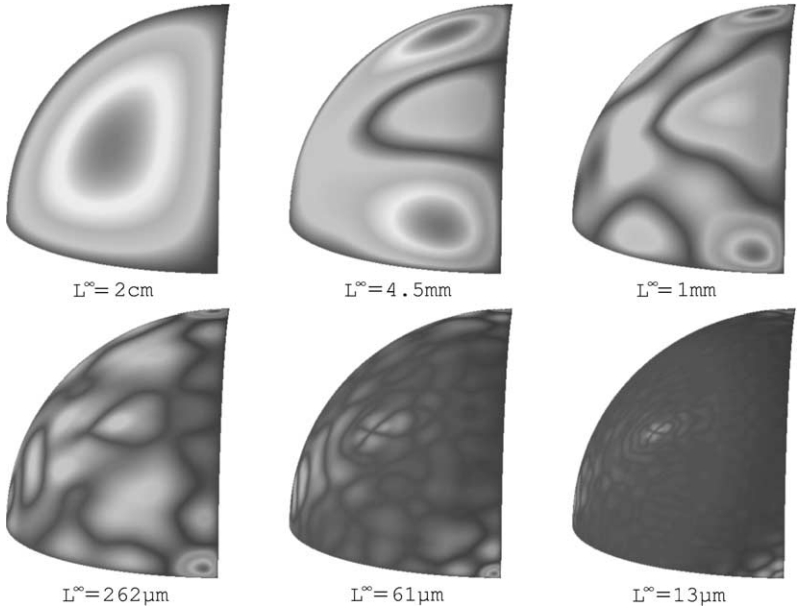


Fig. 7. Pseudo color images showing the successive approximation of the sphere octant by the combined subdivision surface with details computed through quasi-interpolation (Section 2.3).

not close to the sphere (maximum deviation: 18 cm). We apply our quasi-interpolation operators adaptively over five levels. The resulting combined subdivision surface with details approximates the sphere with an accuracy of 0.3 μm (L^1 norm), respectively 13 μm (L^∞ norm). The corresponding pseudo color error plots are shown in Fig. 7. Note that the colors were rescaled for each pseudo color plot to visualize the rapidly decreasing error. The maximum error was 2 cm, 4.5 mm, 1 mm, 262 μm, 61 μm, and 13 μm for levels 0 through 5. The maximum error is quickly isolated near the valence 5 vertex and the corners.

3. Trimming algorithm

In this section we describe the trimming procedure for subdivision surfaces of the type given above. The emphasis of this procedure is on producing a trimmed surface which is *exact* to the degree required in CAGD applications. We achieve this by limiting the deviation between the original surface and the trimmed surface to within a prescribed bound $\varepsilon > 0$. Furthermore, the trimmed surface is identical to the original surface away from the trim curve and exactly interpolates the trim curve at its boundary.

A typical trimming operation begins with the user specifying a region to remove from a surface (Fig. 8). The trim region is bounded by a trim curve that lies on the surface. The representation for the trim curve is a black box which allows evaluation of the curve for any parameter value $u \in [0, 1]$. Additionally, we require knowledge of the relationship between the trim curve and the control polyhedron in the form of an *approximate* pre-image. This pre-image is only used to adapt the control polyhedron to the features of the curve and should not be confused with the calculation of *exact* pre-images for NURBS trimming.

There are three main steps to a trimming operation. In the first step, the control polyhedron is locally remeshed to accommodate the trim curve. Next, an initial sampling of the input surface is chosen for the control points of the new trimmed surface through a correspondence. Finally, an approximation stage optimizes the surface shape near the trim curve.

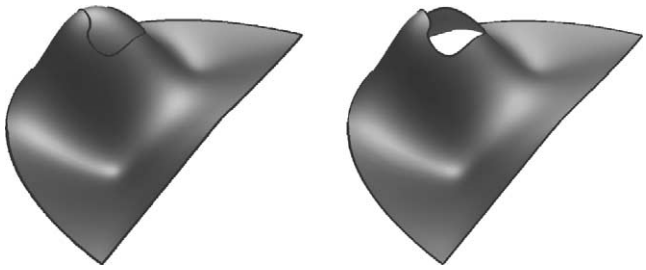


Fig. 8. The input to the trimming algorithm is a combined subdivision surface and a trim region on the surface (left). The trim region is removed and the trim curve is interpolated at the boundary of the surface (right).

3.1. Remeshing

Adaptive remeshing of the original control mesh is an important feature of our trimming procedure which distinguishes it from classical methods. The remeshing algorithm localizes the trimming operation and reduces the size of the control polyhedron to the support of the trimmed surface, as follows:

- triangles on the interior of the trim region are removed;
- near the trim curve, the control polyhedron is locally adapted to the features of the curve;
- the control polyhedron is unchanged away from the trim curve.

This principle is illustrated in Fig. 9. The remeshing algorithm proceeds through three stages:

(1) *The control polyhedron is locally adapted near the trim curve*. The triangles which map to the curve under the approximate pre-image (Fig. 9(a)) are tested against a refinement criterion. A triangle that is identified for refinement is subdivided and

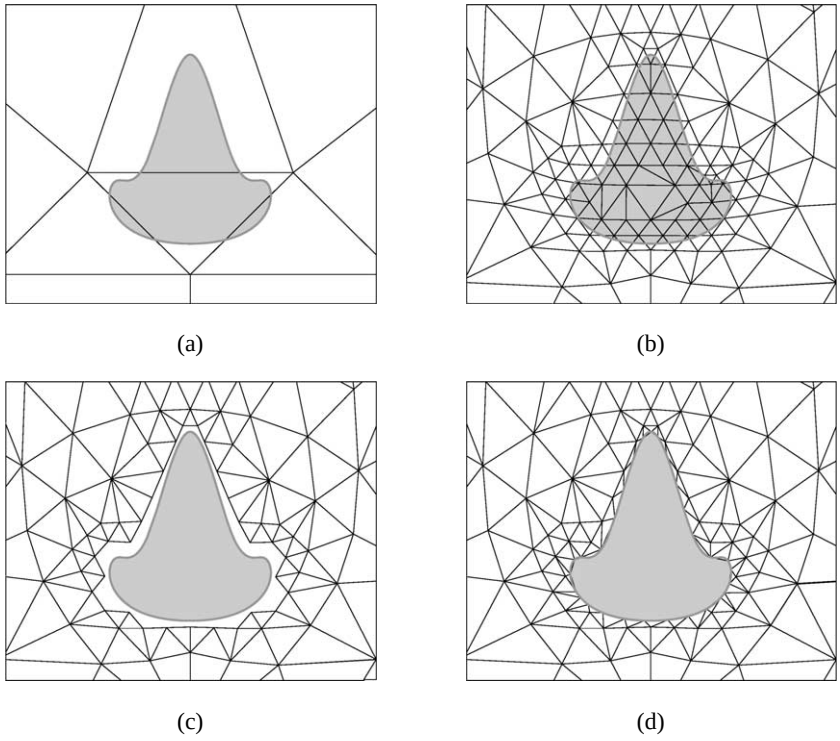


Fig. 9. (a) An initial non-symmetric combined subdivision surface with an approximate pre-image of the trim curve. (b) The triangles in the pre-image are refined as needed to adapt to the curve complexity. (c) Extraneous triangles are removed. (d) The curve is attached to the mesh. Subsequent relaxation optimizes the control point position and triangle quality. The result is the trimmed combined subdivision surface.

the pre-image is updated. This continues until all of the triangles in the trim curve pre-image require no further refinement (fine triangles in Fig. 9(b)). In our example, we refine a triangle if the length of the curve segment exceeds its chord length by a prescribed threshold value.

(2) *The triangles in the trim region are removed* (Fig. 9(c)). Using the updated trim curve pre-image from stage 1, the triangles in the trim region are identified and discarded from the control polyhedron.

(3) *The control polyhedron is attached to the trim curve* (Fig. 9(d)). A triangle strip is added along the control polyhedron boundary created in stage (2). The new vertices that attach to the trim curve are assigned initial parameter values u_i^0 taken locally from the trim curve. These vertices are then iteratively relaxed to improve the aspect ratios of the new triangles. In the relaxation step, we select the triangle $p_i^0 q^0 r^0$ for each boundary vertex p_i^0 and estimate the parameter value $\tilde{u}_i^0$ for an isosceles triangle, i.e.,

$$\left(\left(q^0 + r^0\right)/2 - c\left(\tilde{u}_i^0\right)\right) \cdot \left(q^0 - r^0\right) = 0.$$

We then modify u_i^0 to be $(1-\delta)u_i^0 + \delta\tilde{u}_i^0$ for step size δ. The parameter values typically converge after a few iterations.

The initial control polyhedron for the trimmed mesh is generated by tessellating the nested subdivision hierarchy produced in the first stage. This introduces irregular vertices into the trimmed mesh and causes the limit surface to deviate from the original surface, which we control in subsequent stages of the algorithm with detail coefficients. Only in stage (3) are new vertices inserted which do not correspond to subdividing the original control mesh. Consequently, the new surface is a reasonable initial approximation of the original surface near the trim curve, and an exact representation away from the trim curve where no refinement occurred.

3.2. Correspondence

Before we can apply the quasi-interpolation stencils given in Section 2.3, we need to identify samples on the original limit surface that correspond to the control points of the trimmed mesh. Thus in an intermediate stage between remeshing and approximation, we choose an initial sampling of the original surface through a correspondence relationship.

There is a natural correspondence for the control points not on the trim curve itself, since they were generated from the subdivision hierarchy of the original control mesh. However, the connectivity of the control points near the trim curve changes through remeshing. Therefore the natural correspondence for the control points near the trim curve is poor, since their limit points are different from the corresponding points on the original surface. For these control points, we improve upon the natural correspondence by choosing new samples on the original surface (see Fig. 10). This is done by performing a limited search on the original surface for a sample that is closer to the new limit point. By starting at the natural correspondence, this search is very effective in establishing the initial sampling of the original surface.

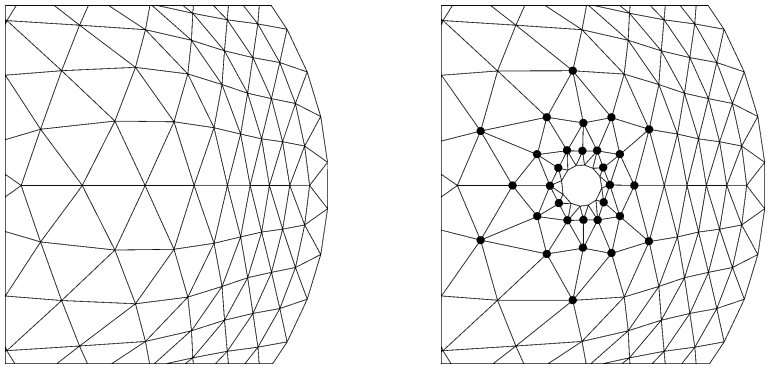


Fig. 10. The control mesh for the original surface (left) and the trimmed control mesh (right). The marked control points have their natural correspondence improved.

3.3. Approximation algorithm

The final stage in the trimming algorithm is to fit the trimmed surface to the original surface. This operation is only required near the trim curve, where the control mesh was generated by the remeshing algorithm in Section 3.1. Away from this region, the trimmed surface is not affected by the fitting operation and thus remains identical to the original surface. Within this region the approximation of the original surface is generated as a hierarchy of detail coefficients. A local refinement criterion limits the depth of the hierarchy in areas where the surface satisfies a convergence threshold $\varepsilon > 0$.

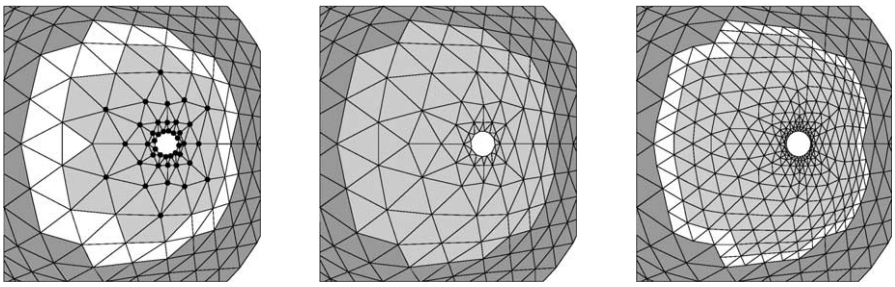
The approximation algorithm is applied to a set of triangles T^0 whose corresponding limit patches overlap the region influenced by the trim curve. Letting $N(p)$ denote the set of triangles incident to a control point p, we initialize T^0 with the union of $N(p_i^0)$ for each p_i^0 whose connectivity changed through remeshing (see Fig. 11(a)). The algorithm proceeds as follows:

```
Approximate(T^j)
    while T^j ≠ ∅:
        Modify(T^j)
        T^{j+1} := Refine(T^j)
        j := j + 1
```

First the limit surface is modified to approximate the original surface at the current level of the hierarchy. Then the triangle set T^j is locally refined to produce a new set of triangles T^{j+1} at the next level in the hierarchy. If $T^j = \emptyset$, the limit surface requires no further improvement, and the algorithm terminates.

The limit surface is modified locally by only computing detail coefficients for the control points P^j on the interior of T^j (see Fig. 11(a)). Since the interpolating stencils are applied after adding the quasi-interpolation detail to the surface, we partition P^j into two sets P^j_{QI} and P^j_I which are modified separately. P^j_{QI} contains:

(1) all coarsest level control points;



(a) (b) (c)

Fig. 11. (a) The input set T^0 is illustrated by the region shaded light gray, with the modified vertices marked. The limit surface corresponding to the dark shaded region matches the original surface exactly and must be preserved. (b) The limit patches that are compared to the original surface are given by $N(T)$. (c) The set T^1 is produced in the refinement step.

(2) all interior control points at finer levels with valence six;
(3) all boundary vertices that are not at a corner.

We identify these control points with the predicate $\mathsf{QI}(p_i^j)$. The control points in P_{QI}^j are modified first: the appropriate quasi-interpolation stencil is applied to each control point to obtain a detail coefficient (computed in $\mathsf{detail}(p_i^j)$), which is added to the control point after being computed for all of P_{QI}^j. This procedure is repeated for the control points in P_I, which become increasingly isolated as the triangle set is refined. T^j is refined in subsequent iterations of the approximation algorithm. The modification algorithm follows:

$\mathsf{Modify}(T^j)$

$$
\begin{aligned}
&P^j := \left\{ p_i^j \mid N(p_i^j) \in T^j \right\} \\
&P_{QI}^j := \left\{ p_i^j \mid p_i^j \in P^j \wedge \mathsf{QI}(p_i^j) \right\} \\
&\forall p_i^j \in P_{QI}^j : d_i^j := \mathsf{detail}(p_i^j) \\
&\forall p_i^j \in P_{QI}^j : p_i^j := p_i^j + d_i^j \\
&P_I^j := \left\{ p_i^j \mid p_i^j \in P^j \wedge \neg\mathsf{QI}(p_i^j) \right\} \\
&\forall p_i^j \in P_I^j : d_i^j := \mathsf{detail}(p_i^j) \\
&\forall p_i^j \in P_I^j : p_i^j := p_i^j + d_i^j
\end{aligned}
$$

The final step refines elements in T^j to produce the input set for the next iteration of the approximation algorithm. Triangles in T^j are selected for refinement by comparing the approximation to the original surface. We must consider all of the limit patches that were modified above (see Fig. 11(b)). A refinement criterion $\mathsf{test}(t)$ assesses the approximation over a single limit patch, and the limit patches which do not adequately approximate the original surface are subdivided. In our implementation, $\mathsf{test}(t)$ estimates the L^∞ error over t by measuring the differences at its vertices after subdividing once, and chooses to refine t if this estimate exceeds a prescribed threshold $\varepsilon > 0$. The accumulation of these child

triangles and their neighbors produces the triangle set T^{j+1}, where the neighborhood $N(T)$ of a triangle set T is the union of $N(p)$ over all control points p in T. We remove from T^{j+1} those triangles that are not adjacent to a child of a triangle in T^j (see Fig. 11(c)). This guarantees that the surface beyond the influence of the trim curve will not be modified. The remaining triangles become the input to the next iteration of the approximation algorithm. The refinement procedure is given below.

```
Refine(T^j)
    T^{j+1} := ∅
    ∀t ∈ N(T^j) :
        if test(t) = false
            T^{j+1} := T^{j+1} ∪ N(subdivide(t))
    ∀t ∈ T^{j+1} :
        if parent(N(t)) ∩ T^j = ∅
            T^{j+1} := T^{j+1} \ {t}
    return T^{j+1}
```

4. Results

We demonstrate the trimming algorithm with a number of examples based on the hubcap design study featured in Fig. 1. All computations were performed in single precision

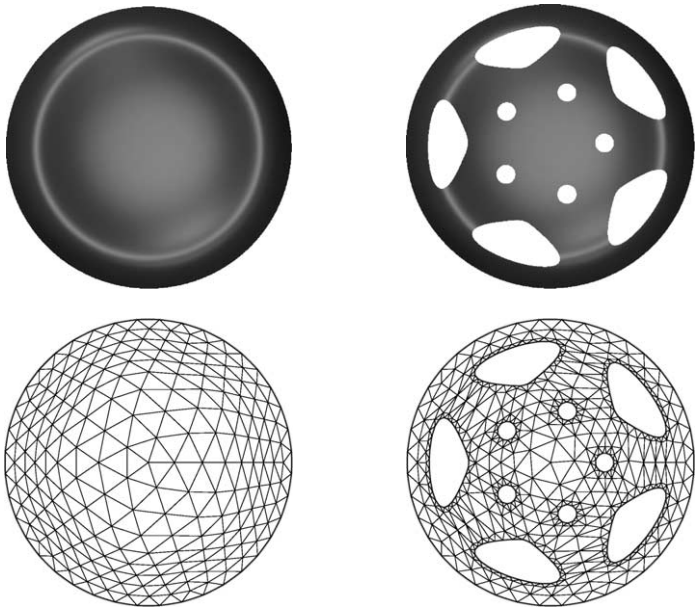


Fig. 12. The initial plate model (left) and the trimmed hubcap model (right). The corresponding control meshes are shown on the bottom.

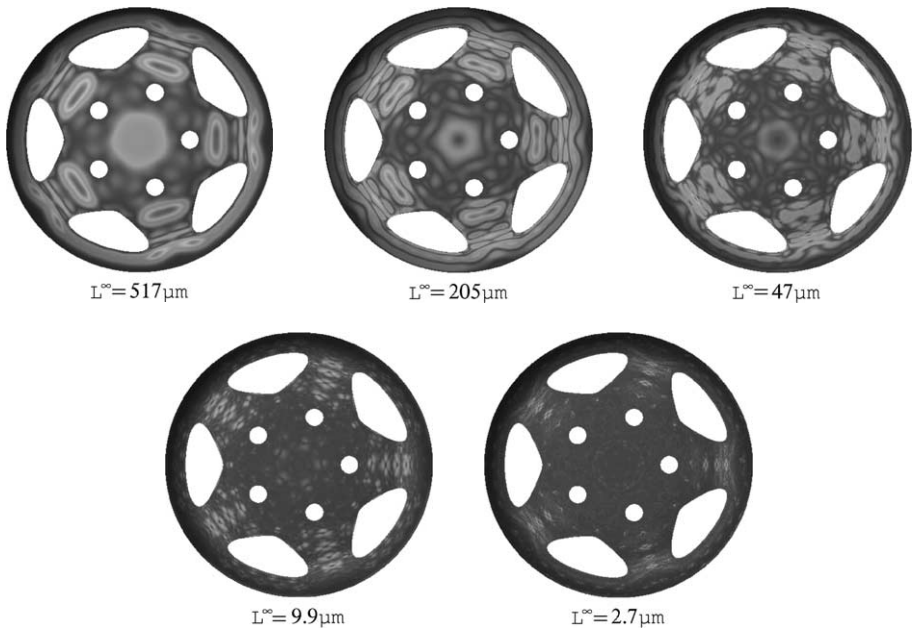


Fig. 13. Pseudo color images showing the error at successive levels of approximation. Shown top left is the surface generated by the remeshing algorithm, before detail coefficients were computed. The colors were rescaled for each pseudo color plot.

arithmetic. The hubcap was produced by trimming circular bores and styling detail from a contoured plate 40 cm in diameter (see Fig. 12). A pseudo color plot of the error in the trimmed surface is depicted in Fig. 13, displayed at successive levels of refinement during the approximation algorithm. The maximum error was 517 μm in the surface produced by the remeshing algorithm (top left) and 205 μm, 47 μm, 9.9 μm and 2.7 μm for levels 0 through 3 respectively. The maximum error is isolated near the high valence vertices.

Fig. 14 illustrates the approximation of the original surface for a circular bore. This example demonstrates the locality and adaptivity of the approximation algorithm. The region influenced by the remeshing algorithm is shown on the control mesh on the top left, and superimposed on the limit surface below. Away from this region, the control mesh and limit surface are identical to the plate model. The adjacent diagrams illustrate the steps in the approximation algorithm through successive levels of refinement. By restricting the deviation of trimmed surface from the original surface to $\varepsilon = 13$ μm, the region for the approximation vanishes after three iterations. The L^∞ error for a limit patch was estimated by measuring the differences at the control points for the patch after subdividing once. The maximum error was 415 μm in the surface produced by the remeshing algorithm (far left) and 218 μm, 29 μm and 10 μm for levels 0 through 2 respectively.

Finally, we demonstrate the remeshing algorithm by moving the circular bore towards the circumference of the plate. As the bore approaches the edge of the plate, the tessellation is made sufficiently fine to separate the boundaries of plate and bore (see Fig. 16(a)). This

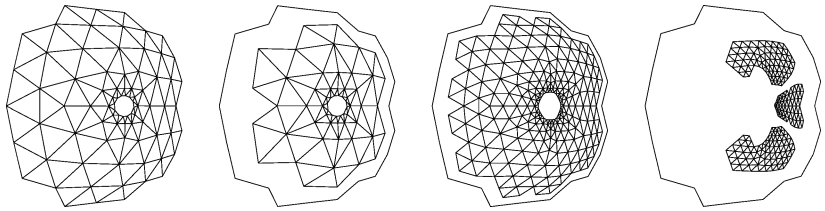


Fig. 14. Stages in the approximation of the original surface near a circular bore. The region of the control mesh affected by the remeshing algorithm is shown on the far left. To the right, the triangles being approximated at successive levels of refinement are shown superimposed on the region of influence.

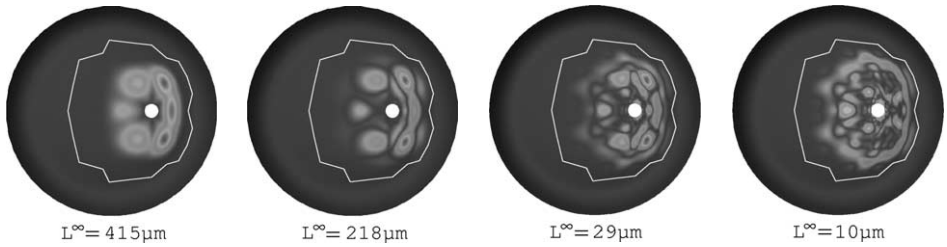


Fig. 15. Corresponding pseudo color plots with the region of influence superimposed. The colors were rescaled for each pseudo plot. For the final surface on the far right, the colors are scaled relative to the prescribed tolerance ($\varepsilon = 13\ \mu$m).

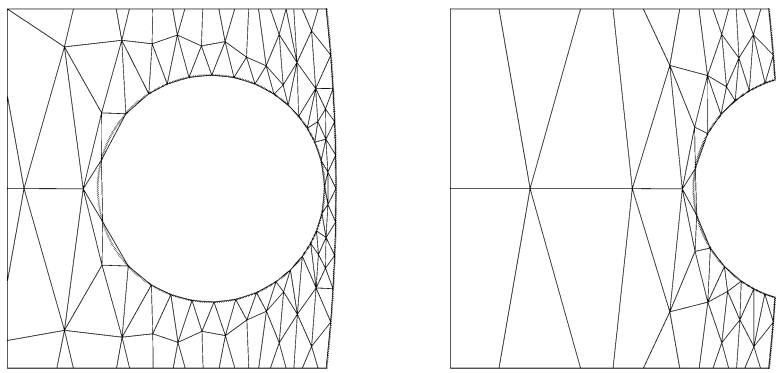


Fig. 16. The control mesh produced for a circular bore. As the bore approaches the edge of the plate, the tessellation between the boundaries becomes finer (left). When the bore cuts the edge of the plate, corners form at the intersection (right).

example demonstrates the robustness of our remeshing procedure near increasingly thin surface sections. In Fig. 16(b), the bore cuts the edge of the plate producing corners where they intersect.

5. Conclusions

We presented a novel trimming algorithm for subdivision surfaces based on the use of a combined subdivision scheme. The latter allows us to guarantee exact (transfinite) interpolation of the desired piecewise smooth trim curve, easily avoiding topological inconsistencies, such as cracks at trim boundaries. In contrast to traditional patch trimming approaches we construct a new control mesh whenever the surface is trimmed. This implies that the original surface is perturbed in the vicinity of the trim curve. However, the approximation error can be controlled and we are able to achieve L^∞ accuracies on the order of 1 part in 10^5 in single precision arithmetic (10 μm for a 1 m model). This is achieved through the use of quasi-interpolation operators together with multiresolution detail coefficients.

Possible future work directions include:

- *Quasi-interpolation*: The approximation properties of our quasi-interpolation operators are as yet poorly understood and more work is needed to fill this gap as well as design better operators.
- *Surface–surface intersection*: Since our trimming procedure only requires an evaluation procedure but no explicit representation of the trim curve it would be interesting to consider it as the basis for CSG operations on subdivision surfaces.

Acknowledgements

This work has been supported in part by NSF (DMS-9874082, DMS-9872890, ACI-9982273), Alias|Wavefront, Microsoft, Intel, Lucent, and the Packard Foundation. We would like to thank the anonymous reviewers for their helpful comments. Special thanks to Khrysaundt Koenig for modeling, lighting, and texturing.

References

Biermann, H., Levin, A., Zorin, D., 2000. Piecewise smooth subdivision surfaces with normal control. Proceedings of SIGGRAPH 2000, 113–120.

DeRose, T., Kass, M., Truong, T., 1998. Subdivision surfaces in character animation. Proceedings of SIGGRAPH 98, 85–94.

Halstead, M., Kass, M., DeRose, T., 1993. Efficient, fair interpolation using Catmull–Clark surfaces. Proceedings of SIGGRAPH 93, 35–44.

Hoppe, H., DeRose, T., Duchamp, T., Halstead, M., Jin, H., McDonald, J., Schweitzer, J., Stuetzle, W., 1994. Piecewise smooth surface reconstruction. Proceedings of SIGGRAPH 94, 295–302.

Keyser, J., Krishnan, S., Manocha, D., 1999a. Efficient and accurate b-rep generation of low degree sculptured solids using exact arithmetic: I—representations. Computer Aided Geometric Design 16 (9), 841–859.

Keyser, J., Krishnan, S., Manocha, D., 1999b. Efficient and accurate b-rep generation of low degree sculptured solids using exact arithmetic: II—computation. Computer Aided Geometric Design 16 (9), 861–882.

Krishnan, S., Manocha, D., 1997. An efficient surface intersection algorithm based on lower-dimensional formulation. ACM Transactions on Graphics 16 (1), 74–106.

Levin, A., 1999a. Combined subdivision schemes for the design of surfaces satisfying boundary conditions. Computer Aided Geometric Design 16 (5), 345–354.

Levin, A., 1999b. Interpolating nets of curves by smooth subdivision surfaces. Proceedings of SIGGRAPH 99, 57–64.

Loop, C., 1987. Smooth subdivision surfaces based on triangles, Master's Thesis. University of Utah, Department of Mathematics.

Schweitzer, J.E., 1996. Analysis and application of subdivision surfaces, Ph.D. Thesis. University of Washington.