DIMACS Series in Discrete Mathematics
and Theoretical Computer Science

A Survey of Subdivision-Based Tools for Surface Modeling

Ioana Boier-Martin, Denis Zorin, and Fausto Bernardini

This paper is dedicated to Henning Biermann, in memoriam.

Abstract. Subdivision surfaces have emerged as a powerful representation for surface modeling and design. They address important limitations of traditional spline-based methods, such as the ability to handle arbitrary topologies and to support multiscale editing operations. In this paper we survey existing subdivision-based modeling methods with emphasis on interactive tools for styling and decoration of 3D models.

1. Introduction

Subdivision surfaces offer several advantages over both irregular meshes and spline patches, two of the most commonly used surface representations today. Subdivision offers a compact way to represent geometry with minimal connectivity information. It generalizes the classical spline patch approach to arbitrary topology, it naturally accommodates multiple levels of detail, and produces meshes with well-shaped elements arranged in almost regular structures, suitable for digital processing. When combined with multiresolution analysis, subdivision offers a powerful modeling tool, allowing for complex editing operations to be applied efficiently at different resolutions.

In recent years, the set of tools available for manipulating subdivision surfaces has been growing steadily. Algorithms for direct evaluation [**Sta98, ZK02**], editing [**BKZ01, BMBZ02, BMZB02, BLZ00**], texturing [**PB00**], and conversion to other popular representations [**Pet00**] have been devised and hardware support for rendering of subdivision surfaces has been proposed [**BAD$^+$01, BKS00, PS96**].

This survey focuses on the use of subdivision-based representations for styling and conceptual design. We explore various methods for manipulating subdivision surfaces and, whenever possible, we illustrate the evolution of such methods from related representations. We pay particular attention to interactive tools which are suitable for design as they allow the designer to instantaneously evaluate results. While we are trying to provide a thorough overview of the area and include the most relevant methods, we realize that the volume of published work goes well beyond

Key words and phrases. Subdivision surfaces, geometric modeling.

that covered in this survey which is by no means exhaustive (see also [**DL02, Sab02**] for additional surveys). Many of the topics presented relate to issues we have addressed in our own work which we hope will provide some insights to those pursueing similar interests.

2. Background

The basic idea of using subdivision to produce smooth curves and later, smooth surfaces, has been around for many years (see [**ZSD$^+$00**] for a brief incursion into the history of subdivision). However, it is only recently that powerful design tools based on this representation have emerged. This is partly due to the recent advent of multiresolution techniques that facilitate capturing of non-trivial shapes and partly due to even more recent advances in subdivision theory and methods for direct and efficient evaluation of subdivision surfaces. For the purpose of this survey, we provide a brief review of the basic concepts pertaining to subdivision surfaces. For additional details we refer the reader to [**ZSD$^+$00, WW01**].

Subdivision defines a smooth surface recursively as the limit of a sequence of meshes (see Figure 1). Each finer mesh is obtained from a coarse mesh by using a set of refinement rules which define a *subdivision scheme*. Many schemes have been proposed in the literature. Examples include Doo-Sabin [**DS78**], Catmull-Clark [**CC78**], Loop [**Loo87**], Butterfly [**DLG90, ZSS96**], Kobbelt [**Kob96a**], Midedge [**PR97**]. Different schemes lead to limit surfaces with different smoothness characteristics. For design purposes, the Catmull-Clark [**CC78**], Loop [**Loo87**] schemes are most often employed as they are closely related to splines (a de-facto standard in modeling today) and generate C^2-continuous surfaces over arbitrary meshes.

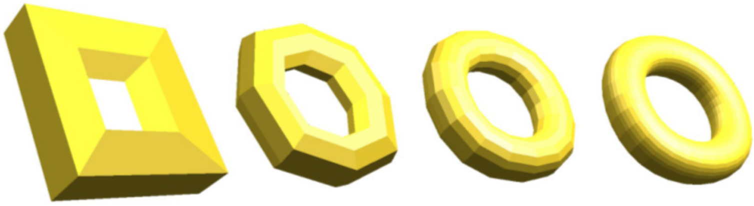


FIGURE 1. Subdivision defines a smooth surface recursively as the limit of a sequence of meshes.

Multiresolution subdivision extends the concept of subdivision by allowing *detail vectors* to be introduced at each level. Hence, a finer mesh is computed by adding detail offsets to the subdivided coarse mesh. Given a *semi-regular mesh*, i.e., a mesh with subdivision connectivity, it can be easily converted to a multiresolution surface by defining a smoothing operation to compute a coarse level from a finer level. The details are then computed as differences between levels. This representation was introduced by several authors in different forms [**LDW97, PL97, ZSS97**]. Figure 2 illustrates the power of multiresolution in capturing complex shapes.

A close connection exists between multiresolutin subdivision and wavelets [**SDS96**]. In particular, two operations known as *Synthesis* and *Analysis* can be defined to propagate data from coarse to fine and in reverse throughout the subdivision hierarchy, similar to wavelet transforms. *Analysis* computes positions of control points

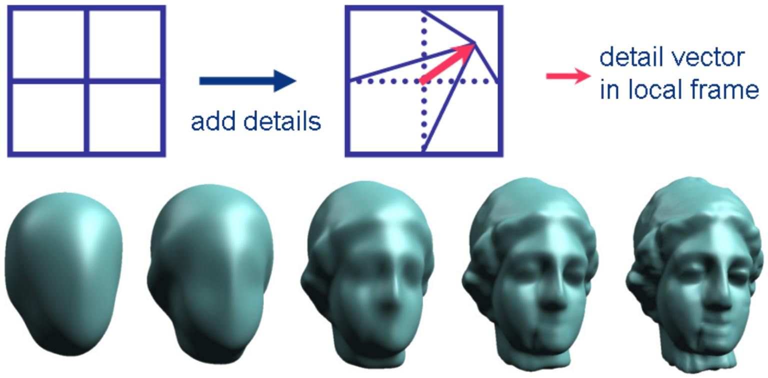


FIGURE 2. Top: multiresolution subdivision extends the concept of subdivision by introducing detail vectors at each level. Bottom: surfaces obtained by subdivision of the same coarse mesh look very different depending on the amount of detail introduced and the level at which it is introduced. From left to right: no details to progressively more details added on finer levels.

on a coarse level $i-1$ by applying a smoothing filter to points on level i. Multiresolution details on level i are computed as differences between the two levels. Conversely, *Synthesis* reconstructs the data on level i by subdividing the control mesh of level $i-1$ and adding the details [**ZSS97**].

An important property of subdivision surfaces is that they can be naturally interpreted as functions on the domain defined by the base mesh (see Figure 3). This parametric interpretation is useful in many circumstances related to design, from derivation of differential quantities to dealing with constraints along arbitrary curves as described in section 3. Figure 3 illustrates this natural parameterization.

3. Surface Modeling Tools

3.1. Free-Form Editing. Free-form manipulation of 3D models is a popular method for modifying existing shapes which attempts to mimic to a certain extent the process of modeling or sculpting a physical object by hand. The applications are numerous, from animated character creation, to virtual restorations, to industrial design.

The sculpting metaphor for geometric modeling has its roots in the parametric surface works of Sabin [**Sab71**] and Bezier [**Béz74**] which contain early mentions of surface deformations. Subsequent work has spanned more than three decades and continues to be investigated in the context of modern systems and surface representations (e.g., [**Bar84, SP86, Coq90, HKD93, CR94, MJ96, SF98, Kob96b, ZSS97, PL97, QMV98, Tak98, WW98, MQ00, TO02, GS01, BMRB04**].

The basic idea of free-form modeling is to introduce a degree of transparency between the designer and the mathematical model of the surface being shaped. Instead of controlling the shape through a set of non-intuitive surface parameters, free-form deformations allow the shape to be controlled through intuitive

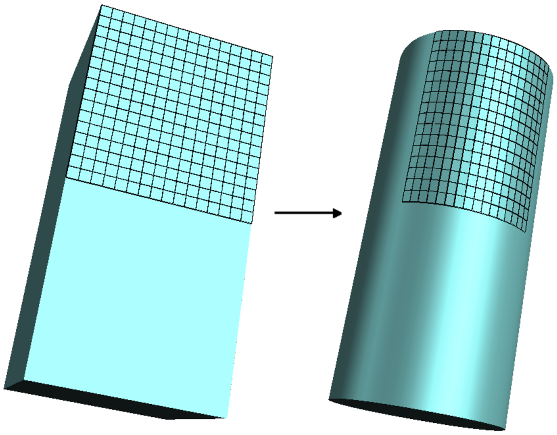


FIGURE 3. Natural parameterization of a subdivision surface. Each time we apply the subdivision rules to compute the finer control mesh we also apply midpoint subdivision to a copy of the initial control mesh. A mapping from a denser and denser subset of the control polyhedron (left) to the control points of a finer and finer control mesh (right) is obtained through repeated subdivision. In the limit, a map from the control polyhedron to the surface is obtained.

manipulation of the surface itself or the space surrounding it. The main challenge is to perform the manipulation through a limited set of controls and to define natural deformations of the surface away from the control positions. Different variations of this paradigm have been developed, including axial deformations [**Bar84, CST94, LCJ94**] which alter the axis of a shape to induce its deformation, lattice deformations [**SP86, Coq90, MJ96**] which operate on the cells of a space lattice to deform the volume inside the lattice, manipulations on scalar field embeddings [**HQ03**], control mesh editing methods which shape parameterically-defined surfaces by imposing constraints on their control meshes [**ZSS97**], and variational methods which operate by optimizing an energy functional over the surface under constraints [**Tak98, BMRB04**].

We focus our attention on methods that take advantage of subdivision representations and among these, we emphasize those that support interactive multiscale modeling. Subdivision representations are particularly suitable for free-form editing due to their hierarchical nature which easily accommodates multiscale edits, as well as their efficiency in terms of storage and access. For a survey of deformable models based on other representations see [**GM97**].

3.1.1. *Control mesh manipulations.* Manipulating control meshes offers a straightforward interface which supports interactive shape deformations. This approach has been extensively employed in spline-based modeling [**CRE01**] and can be naturally extended to subdivision surfaces. Collections of control mesh vertices, edges, and faces are re-positioned so as to induce modifications of the resulting limit surface. In addition, control points can be added and edges and faces can be split to increase the complexity of the shape as editing progresses. This type of manipulation is very common and can be found at the basis of commercial modeling packages with support for subdivision surfaces. It is routinely used for animated character design (e.g., in Discreet's 3D Studio Max [**dsm**], in Alias' Maya [**may**]) and is becoming increasingly popular for industrial modeling (e.g., in Dassault Systèmes' Catia [**cat**]). Figure 4 illustrates examples of shape modeling through control point manipulation.

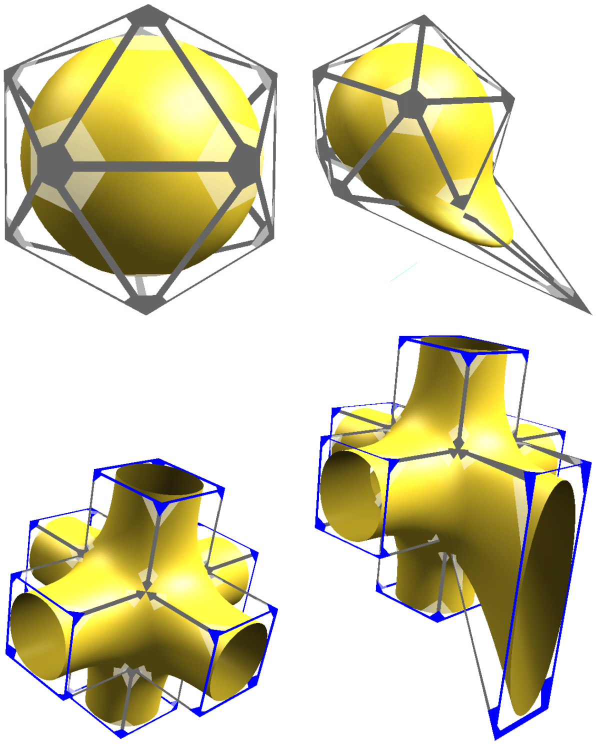


FIGURE 4. Shape modeling through control point manipulation: Loop subdivision surface (top), Catmull-Clark subdivision surface (bottom).

Single resolution control mesh manipulations offer only limited flexibility in designing shapes: only coarse shape deformations can be accommodated. Multiresolution subdivision surfaces are a much more powerful representation which lends itself very naturally to multiscale editing. Depending on the level at which the editing occurs, either a global deformation (coarse level) or a local deformation (fine level) is induced. This idea was exploited, for instance, in [**ZSS97, PL97**] for interactive multiresolution editing of Loop surfaces and in [**DKT98**] for Catmull-Clark ones. Using a combination of subdivision (i.e., transforming a coarse mesh into a finer one) and smoothing (i.e., transforming a fine mesh into a coarser one), edits performed at different levels of subdivision can be propagated through the hierarchy while keeping the magnitude of multiresolution details under control. Figure 5 illustrates edits at various scales performed on the Armadillo model.

Variations of this approach include modeling with *displaced subdivision surfaces* [**LMH00**] and subdivision surface *fitting* [**STKK99, LLS01a, MZ00**]. The

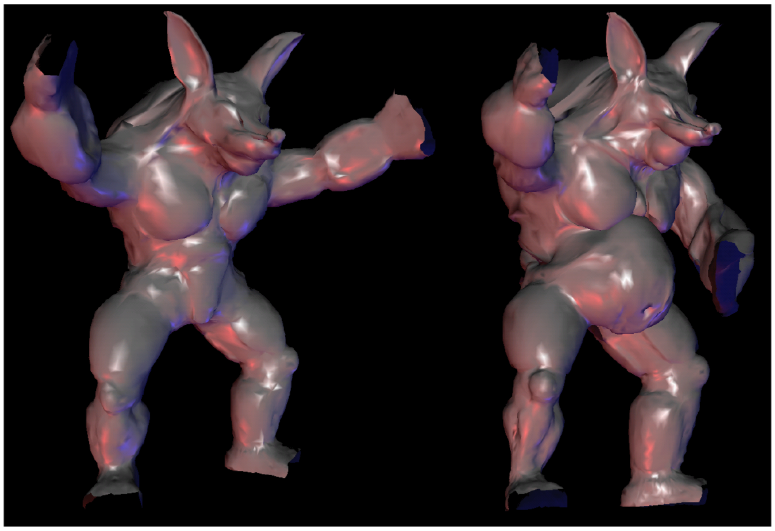


FIGURE 5. Multiresolution editing according to [**ZSS97**]: left – input model; right – editing result. Note the large-scale edit of the belly and the fine-scale edit around the chin.

displaced representation can be viewed as a restricted form of multiresolution subdivision consisting of a control mesh and a single level of scalar details. A domain surface is generated from the control mesh using Loop subdivision [**Loo87**]. A displacement map computed from the scalar displacement is then applied over the domain to generate the final surface. The displacements can be edited to create fine-level features on the surface, while control mesh edits lead to global shape alterations. In surface fitting a surface is deformed to conform to the shape of another given data set (e.g., points, curves, another surface). This approach is somewhat different than those discussed so far in that it is less suitable for interactive manipulation. Typically some optimization of the surface being fitted is performed in order to determine optimal control point positions which lead to a best fit between the surface and the target. The accuracy of the fit is controlled through a threshold parameter that bounds the error between the target and the fitted surface.

3.1.2. *Variational design.* Variational surface design operates on the principle of modifying a shape so that its *fairness* is optimized. Surface fairness is typically measured in terms of its energy and the idea is to find a minimum-energy state which, in turn, corresponds to the fairest possible shape. In Computer Graphics, energy-minimizing surfaces became popular in the context of simulating physical properties of materials [**Bar84, TF88, WW92**]. Celniker and Gossard [**CG99**] and later Welch and Witkin [**WW92**] pointed out the relationship between fair surface design and energy minimization.

Most commonly, fairness is expressed as an integral of a physical parameter associated with a real object bearing the shape of the surface [**Hal96**]. A widely used measure of fairness is the combination of *stretching* and *bending* energies:

$$Energy(S) = \alpha \int ||I||^2 dS + \beta \int ||II||^2 dS \tag{3.1}$$

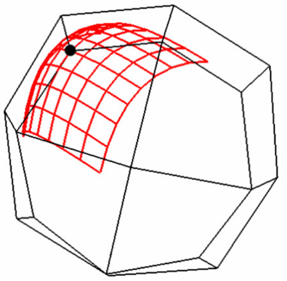


FIGURE 6. Local quadratic interpolant used to approximate first and second order derivatives [**BMRB04**].

where I and II denote the first and second fundamental forms of the surface and $||\cdot||$ is a suitably chosen matrix norm [**TPBF87**].

For practical purposes, discretized linear forms of equation (3.1) using parametric derivatives are typically employed:

$$E_{stretch} \approx \int_{\Omega} \left(\frac{\partial S}{\partial u}\right)^2 + \left(\frac{\partial S}{\partial v}\right)^2 dudv \tag{3.2}$$

$$E_{bend} \approx \int_{\Omega} \left(\frac{\partial^2 S}{\partial u^2}\right)^2 + 2\left(\frac{\partial^2 S}{\partial u \partial v}\right)^2 + \left(\frac{\partial^2 S}{\partial v^2}\right)^2 dudv \tag{3.3}$$

where Ω denotes the parametric domain of the surface S. Most variational approaches take advantage of these expressions, although alternative approaches have been proposed (e.g., [**CSA$^+$02**]). The main differences are in the types of parameterizations used to derive the differential quantities. For example, Greiner [**Gre94**] and later Kobbelt [**Kob96a**] suggested a discrete exponential map for local parameterizations (see Figure 6) such that each vertex P_0 has coordinates $(0, 0)$ and its 1-ring neighbors $P_i \in R(P_0)$ are assigned coordinates:

$$(u_i, v_i) = e_i \left(\cos\left(\sum_{j \in R(P_0)} \alpha_j\right), \sin\left(\sum_{j \in R(P_0)} \alpha_j\right)\right) \tag{3.4}$$

where

$$\alpha_j = \frac{2\pi \angle(P_j^l P_0^l P_{j+1}^l)}{\sum_{j \in R(0)} \angle(P_j^l P_0^l P_{j+1}^l)}. \tag{3.5}$$

In the context of subdivision surfaces, Halstead et al. [**HKD93**] were among the first to describe a method for interpolating a given shape with a Catmull-Clark surface while minimizing surface fairness. Given the lack of a "natural" parameterization near extraordinary points, they re-formulated the stretch and thin-plate energy definitions in terms of the control meshes at different subdivision levels (rather than the limit surface). In their method subdivision is used to isolate extraordinary vertices and bi-cubic spline evaluation is used to evaluate the fairness norm away from such vertices.

Kobbelt [**Kob96b**] introduced the concept of *variational subdivision* to create interpolatory subdivision rules that place newly inserted vertices so as to minimize a global energy functional. Using a similar idea, Weimer and Warren [**WW98**] propose two schemes for variational subdivision of thin-plate splines. One scheme provides an exact solution to the variational problem, but the subdivision matrix has to be recomputed at every subdivision level. The other scheme is only approximate, but has the advantage that rules can be precomputed. Both schemes are restricted to rectilinear grids. Another method which connects subdivision with fairing and cascading multigrid methods was proposed in [**DMR02**]. The basic idea in this case is to interpret the evolution of the surface under curvature motion as a filtering proces.

Later on, Friedel et al. [**FMS03**] proposed using the characteristic map parametrization to construct first order data-dependent energies. This leads to a nonlinear minimization problem which is solved by re-writing the surface energy as a linear combination of precomputed stiffness matrices.

Constraints play an important role in variational design methods. In their absence, the optimization problem has a trivial solution, which usually leads to the collapse of the surface to a single point (an exception is the method of Boier-Martin et al. [**BMRB04**] in which the trivial solution corresponds to the input surface). We distinguish between two classes of constraints [**WW92**]:

- *Finite-dimensional*: involve point and normal constraints at discrete locations on the surface. These are the most commonly used. Point constraints are used to enforce spatial interpolation conditions. For subdivision surfaces such constraints typically correspond to control points and are easy to implement by solving linear systems. Normal constraints are used to enforce surface normals at certain points on a surface. Different approaches can be used to constrain normals: expressing the fact that two tangent vectors must be perpendicular to the prescribed normal, enforcing the positions of the vertices of a given face so that the face normal coincides with the prescribed one, or constraining tangent vectors rather than normals (the last two tend to over-constrain the problem).
- *Transfinite*: involve one or two-dimensional surface entities such as embedded curves and patches. Curve constraints are among the most common in this category. Enforcing such constraints involves solving an integral over the entity. For example, to constrain a surface curve $C(t) = S(u(t), v(t))$ along a given space curve $C_0(t)$, the following must be satisfied:

$$\int (C - C_0)^2 = 0 \tag{3.6}$$

Such constraints are usually discretized and enforced either by using a least-squares approach [**WW92**] or by reparameterizing the surface to align control points or edges with constraints [**BMRB04**] (see also Algorithm 1 in section 3.2). An alternative approach is to evaluate the curves and to incorporate the result of the evaluation into the subdivision rules to produce a limit surface that interpolates the curves. This is the object of *combined subdivision schemes* [**Lev99**] (see also [**Nas00, NA02, SWZ04**]).

Figure 7 illustrates the result of modeling with various types of constraints.

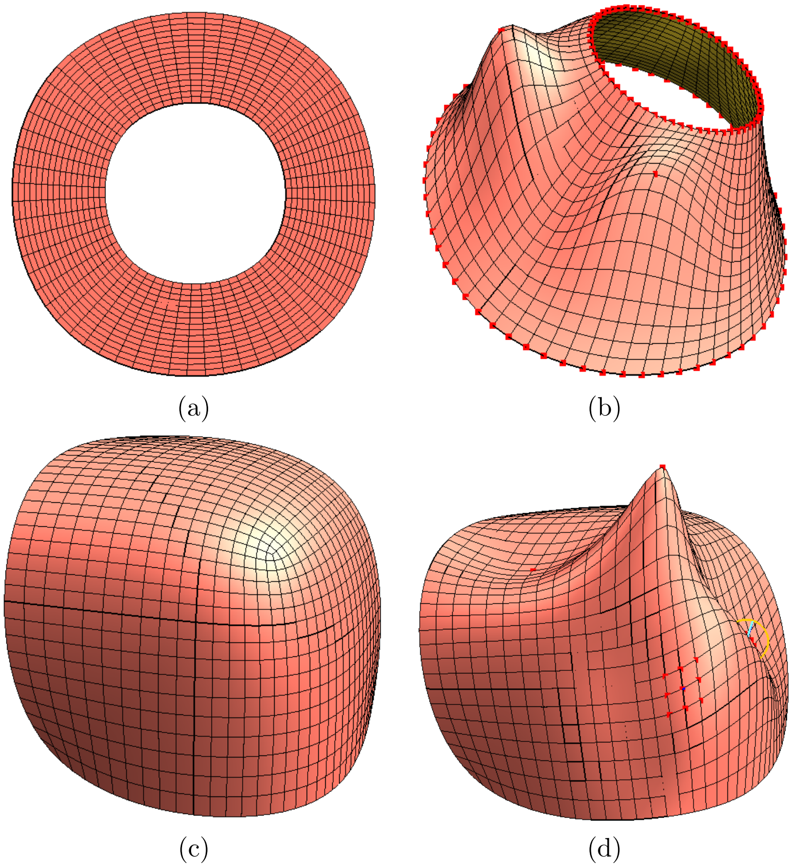


Figure 7. Constraint types: (a)-(b) point and discretized curve constraints; (c)-(d) normal constraints.

Another important consideration in dealing with constraints is the *region of influence* of a constraint. It is defined as the portion of the surface affected by the constraint. The region of influence can be explicitly enforced [**Kob00**] by letting the designer encircle an area on the surface. This generates boundary constraints between the surface inside the area of influence and the rest of the surface. Alternatively, in the case of hierarchical representations such as subdivision hierarchies, the region of influence can be controlled indirectly through the levels at which constraints are defined. For example, Takahashi et al. [**Tak98**] impose constraints at various scales using a wavelet framework. Constraints are being propagated from finer to coarser scales, however, the region of influence of each constraint is not controlled in any way. In [**BMRB04**] the influence of a constraint is explicitly enforced by the coarse level at which the constraint is propagated. Thus, more global or local edits can be performed depending on the level to which the constraint is restricted: a coarser level will induce a more global deformation, whereas a finer level will produce a more local edit (see Figure 8).

A related issue is that of detail preservation. When a global shape change occurs, it is often expected that the high frequency details are preserved over the modified surface. The face of Venus in Figure 9 is represented as a multiresolution subdivision surface in which non-trivial detail vectors capture the organic shape of the model. If a shape deformation is performed by pulling on a single point at the tip of the nose, a naive energy optimization approach leads to a fair shape that satisfies the constraints, but all the details of the face are lost (note that boundary constraints must also be imposed in this case to avoid the collapse of the surface to a single point). One solution is to separate the high-frequency information before optimization and to ”re-apply” it to the new shape [**Kob00**]. This introduces an overhead related to saving and restoring surface details. To avoid this overhead, Boier-Martin et al. [**BMRB04**] propose to define a vector field of deformations over the surface and to optimize the energy of this vector field rather than the energy of the surface itself. Initially all deformation vectors are null. When an edit occurs, the corresponding deformation vector (i.e., at the tip of the nose) becomes non-null. The optimization procedure tries to smooth the deformation field under the constraints defined by the non-null vectors. Since the deformations are defined with respect to the detailed shape, the details are preserved during deformation. Note that, in this case, boundary constraints are not necessary as the rest shape in the absence of constraints is the input shape.

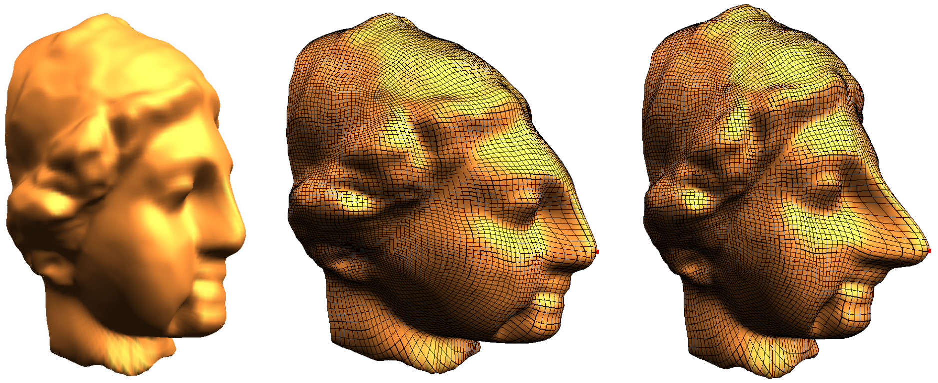


FIGURE 8. Region of influence of a multiresolution constraint: left – input model; middle – constraint is propagated to the coarsest subdivision level, inducing a global deformation of the head by pulling a single point on the nose; right – constraint is propagated only two levels coarser inducing a more localized edit.

An added advantage of subdivision hierarchies is that they facilitate the use of multigrid methods [**Bri87**] to solve the constrained minimization problem. In the presence of many constraints, however, even multigrid solvers may be too slow to yield results at interactive rates. A possible solution [**BMRB04**] is to aim for an approximate solution during interaction and a more accurate (non-interactive) result after the interaction has stopped. Figure 10 illustrates the differences between

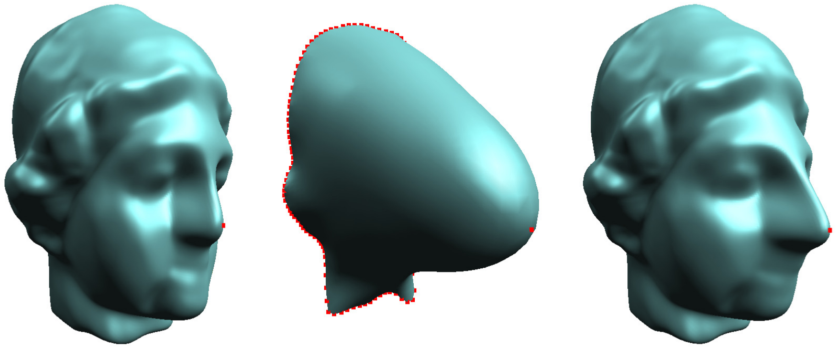


Figure 9. Energy optimization with constraints: left – input multiresolution subdivision surface with details; middle – optimization without detail preservation; right – optimization with detail preservation.

a Catmull-Clark approximation obtained at interactive rates and a more accurate multigrid minimization.

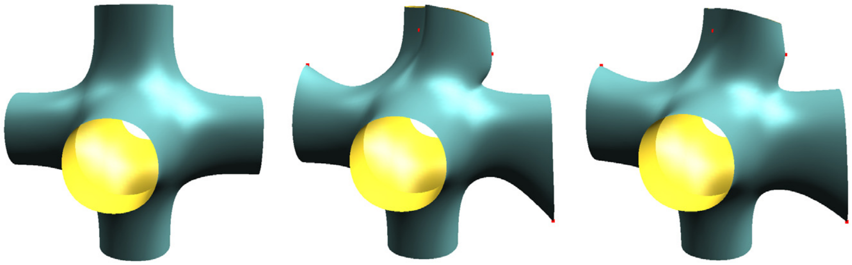


Figure 10. Computing a solution to the energy minimization problem with different accuracies: left - input model; middle - Catmull-Clark solution obtained interactively; right - multigrid solution.

For completeness, we mention the fact that the evolution of energy over time has also been considered to derive *dynamic surface models* [**TQ94, QT96**]. Dynamic models based on subdivision surfaces have been proposed by Qin et al. [**QMV98**]. Such models are typically too complex to support interactive design operations.

Topology modifications. The free-form modeling methods discussed so far operate by deforming the input surface without changing its topology. Some applications, however, may require topological modifications, such as creating handles and tunnels. An interactive sculpting environment which supports this type of edits was proposed in [**GOP99**]. The *Localized hierarchy Surface Splines* allow adding handles and punching holes, while maintaining C^1 continuity across the surface which is represented explicitly in piecewise polynomial or spline form. The main idea behind localized hierarchies is to allow local edits on locally refined mesh fragments based solely on coarser level data. Direct manipulation is performed by interacting directly with the surface rather than with control mesh. The types of operations supported include fillets, blends, semi-sharp features, extrusions, holes, and bridges.

Using meshes as an underlying representation, Guskov et al. [**GKSS02**] propose a user-driven procedure for inducing topological modifications in a semi-regular setting. The so-called *hybrid meshes* are multiresolution surface representations which enhance subdivision-based refinement operations with irregular operations that support changes in topology and approximate detailed features at multiple scales. In [**GKSS02**], hybrid meshes are defined as quadrilateral meshes on which regular $1-4$ face splits are combined with irregular operations through which groups of quads are removed and/or replaced.

3.2. Boolean Operations. Boolean operations provide a straightforward approach to creating complex models from simpler ones using intuitive combinations. Addition, subtraction, and intersection can be packaged into editing tools for modeling solids bounded by subdivision surfaces.

3.2.1. *Mesh-Based Approximations.* Traditionally, Boolean operations on boundary representations (B-reps) of solids have required intersecting parametric surfaces, removing the unwanted parts, and building new surfaces from the remaining ones. This approach presents a number of challenges, as intersections are difficult to perform for high-order B-reps and often lead to increasingly complex intersection curves. Exact matching of surfaces bordering such curves is also problematic, as it is not easy to ensure that curves in different parametric domains coincide in 3D. Consequently, subsequent editing of the resulting models may lead to unwanted artifacts in the surface (e.g., cracks) which require special handling.

A substantially simpler approach, proposed by Linsen [**Lin00**] is to use the control meshes corresponding to the parametric parts being combined, rather than the surfaces themselves. This implies that the intersections between solids are only approximately computed. At the same time, the problem of intersecting arbitrary surfaces translates into the much simpler one of intersecting arbitrary meshes. The meshes are first triangulated to avoid difficulties posed by handling of non-planar faces. Two approaches to building a combined control mesh are discussed: clipping triangles along the intersection boundaries and connecting intersection points and removing faces along the intersection curves and remeshing the resulting gaps. The latter has the advantage that it produces a more visually pleasing result. The main drawbacks in both approaches lie in the inefficiency of computing triangle-mesh intersections and robustness issues associated with such computations as well as gap filling for arbitrary gap topologies (see also [**LFKN03**] for variations on the topic of computing intersection curves for subdivision surfaces).

Using a similar control-mesh based approach, Biermann et al. [**BKZ01**] propose an approximate scheme for computing Boolean operations which deals with several important issues: matching the topology and the geometry of the intersection curve, fitting the resulting surface to the original data, and accurately capturing and representing sharp features in the result. The method uses piecewise-smooth multiresolution Loop [**Loo87**] subdivision surfaces to represent surfaces being combined. The algorithm assumes that each part being used in a Boolean operation is bounded by a closed orientable surface. It follows several steps:

(1) Compute intersection curves.
(2) Build resulting control mesh and compute an initial parameterization of the resulting surface over this mesh.
(3) Optimize the parameterization from the previous step.
(4) Use multiresolution fitting to approximate the input data as closely as possible.

For the first step, the authors improve on both the efficiency and the robustness of the naive mesh-mesh intersection approach by using bounding box hierarchies to accelerate computations and a perturbation scheme [**Sei98**] to increase robustness.

After determining the topology of the intersection, control meshes are merged with special consideration for several issues: preserving the topology of the cut, inserting a minimal number of new vertices, and keeping their valence small. The input control meshes are cut along intersection curves and a new control mesh is combined from the remaining pieces. The cutting process takes advantage of the natural parameterization of subdivision surfaces over their control meshes (see section 2) to approximate the intersection curve by alternating so-called *Snapping* and *Refinement* steps:
Algorithm 1 (snapping and refinement):

```
Given a domain mesh $M$ and an intersection curve $c(t)$ in $M$
Repeat
    For each vertex $v$ of a triangle intersected by $c$ do
        1. Find $\alpha \in c$ closest to $v$
        2. Snap $v$ to $\alpha$ if possible
    Adaptively refine parameterization
until (curve adequately approximated)
```

Snapping is performed between points of the curve and parametric mesh vertices, if they are sufficiently close. While optional, this step considerably reduces the complexity of the resulting domain (fewer faces). The role of the *refinement* is to increase the accuracy with which intersection curves are approximated. It is typically performed by midpoint subdivision of triangles which are intersected by curves multiple times or which fully contain curves. Figure 11 illustrates this process. The output of this step consists of piecewise linear approximations of the intersection curves, either along input edges or along newly introduced edges obtained by splitting triangles.

After cutting, the portions of the control meshes not required in the Boolean operation are removed and the meshes are joined along their boundaries. This is also done in two steps: vertices along one boundary are paired to corresponding vertices along the other boundary. When correspondences do not exist, triangles along the boundary are refined so as to introduce new vertices. Paired vertices close to one-another are merged together. During merging, intersection curves are also tagged with sharp feature tags (see also section 3.3).

By construction, the resulting merged control mesh constitutes a parameterization domain with the property that every one of its vertices belongs to one of

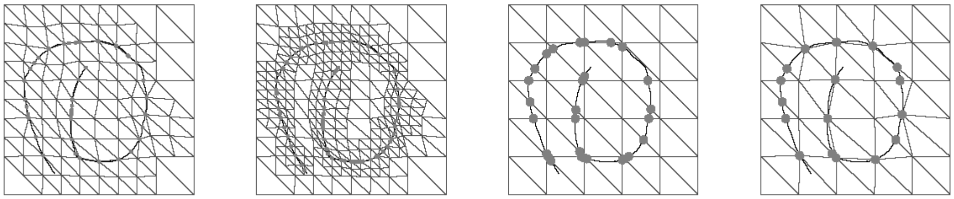


Figure 11. Refinement and snapping: two steps of refinement are shown on the left. The image of the curve in parameter space and vertex snapping are shown on the right (see [**BKZ01**]).

the original domains. However, the initial parameterizations of the parts of the input models corresponding to the Boolean operation may not be optimally parameterized over the new domain. An optimization procedure is used to reduce the distortion of the resulting surface over the new domain.

The last step of this method computes optimal positions of control points given the previously computed parameterization. The merged control mesh is subdivided a number of times and the resulting mesh is fitted to the original data in least-squares sense. Results of Boolean operations obtained with this method are shown in Figure 12.

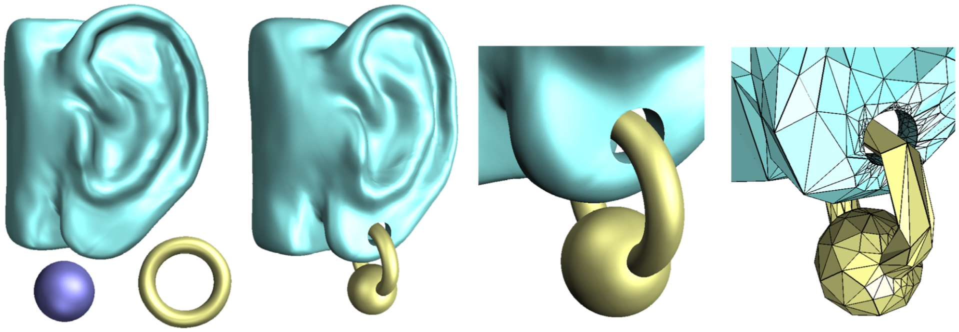


Figure 12. Boolean operations on multireolution subdivision surfaces [**BKZ01**].

3.2.2. *Surface Cut-and-Paste.* Surface pasting can be viewed as an instance of a Boolean operation. The basic paradigm implies creating new models by combining pieces of existing models. In its most basic form, a cut-and-paste operation involves selecting and transferring a *feature* of interest from a *source surface* to a *target surface*. There are several fundamental steps involved such an operation:

(1) Feature selection
(2) Separation of surfaces into base and detail parts
(3) Transferring the feature onto the target surface

The idea of pasting surfaces was first introduced in the context of hierarchical splines [**BBF94, CMB97**]. In this case a tensor-product B-spline surface is designated as the feature to be attached to another surface. Steps (1), (2) are assumed to have been performed in a pre-processing stage and (3) is achieved by representing tensor-product B-splines as Greville displacement B-splines [**BBF94**] and applying a mapping that takes into account the topology of the target surface and the Greville displacement representation of the feature [**BBF94**]. The main restriction is that there are no smoothness guarantees at the boundary between the feature and the target surface (not even C^0 continuity). One solution is to refine the feature surface so that its boundary better approximates the target. However, this amounts to introducing unnecessary control points over the entire feature (rather than only along boundaries), making subsequent processing of the feature very inefficient. An alternative solution was proposed by [**CM00**] and makes use of quasi-interpolation [**dF73**] to improve the result of pasting. In this case, interior feature control points are pasted using Greville displacements, while boundary points are pasted using quasi-interpolation. This leads to a composite surface which still exhibits discontinuities along the pasting boundary, however, less severe than in the original approach. In addition to the lack of continuity, the types of features that can be pasted are also limited by the underlying surface representation. Performance is also an issue due to expensive evaluations. An interactive spline-based interface was developed in [**Ma00**]. Due to performance limitations, the feature is not positioned directly onto the target surface, but rather is floating in its vicinity and the user is presented with a rough outline of the contour of the feature on the target. Once a position is decided upon, the actual pasting occurs.

Biermann et al. [**BMBZ02**] describe a more general procedure for cutting and pasting portions of existing surfaces using an intuitive approach, similar to those commonly used for 2D image cut-and-paste. The user initiates a cutting operation by selecting a feature of interest on an existing surface (termed the *source* surface (see Figure 13 (a)). She also specifies a position on a *target* surface where the source feature is to be pasted (see Figure 13 (b)). The actual pasting is performed in a sequence of steps (Figures 13(c)-(g)) which take advantage of the underlying semi-regular representation to achieve interactive rates. A discussion of the main steps follows.

Feature selection is performed interactively by the user who selects a region of interest on the source surface. A free-form closed space curve is used to outline the selection. The portion of the surface inside the curve constitutes the feature(s) of interest. The curve can have an arbitrary shape and does not have to be aligned with underlying mesh edges. The portion of the surface inside the curve must have disk topology, but it does not have to be a height field (see Figure 14).

Base / detail separation must be performed on both the source and target surfaces to define what constitutes feature detail as opposed to the larger-scale surface shape that should be ignored. Since this is largely dependent on the semantics of the operation, it is best left to the user. In [**BMBZ02**], the authors propose a continuum of base surface choices controlled by a *flatness* parameter. The base surfaces are obtained by smoothing the original surface to various degrees or by simple energy minimization within the feature boundary. Figure 15 illustrates the different effects obtained using different base surfaces for the source and target models.

Feature transfer is a complex process as it generally involves finding a mapping between two arbitrary surfaces. The solution proposed in [**BMBZ02**] is to build the mapping as a composition of maps to an auxiliary plane. The advantage of this approach is that it no longer requires parameterizing an arbitrary (source) surface onto another arbitrary (target) surface. Instead, flattening methods which

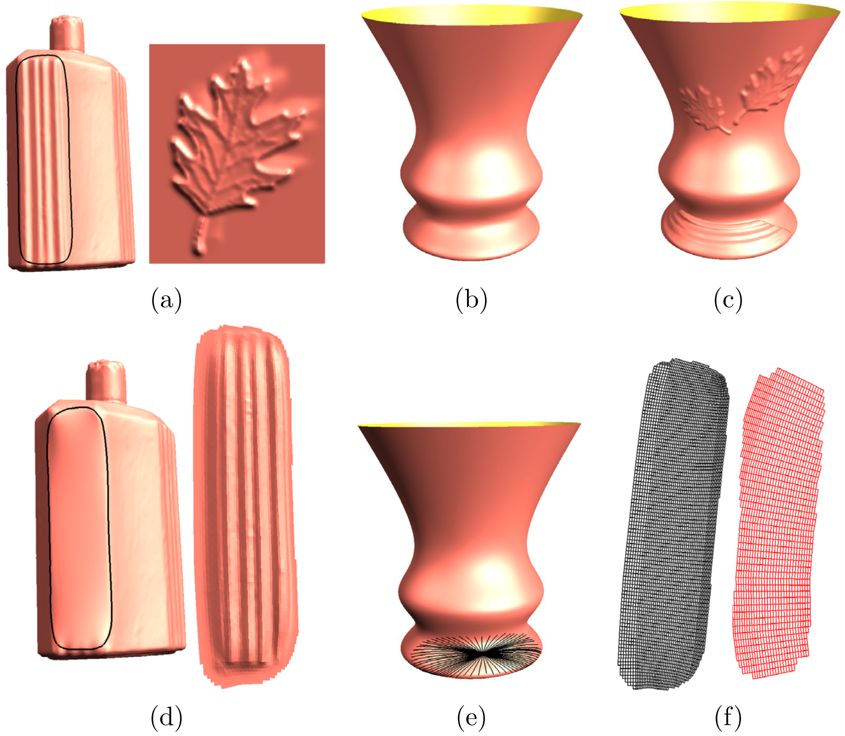


Figure 13. Feature-based design of an ornate vase: (a) input (source) surfaces; (b) target surface; (c) result after multiple pasting operations; Steps of a cut-and-paste sequence according to [**BMBZ02**] shown for the bottle pattern: (d) source feature selection; (e) finding a target region around a user-selected position on the target surface; (f) parameterization of source (left) and target (right) regions onto a common plane (shown displaced for illustration).

have been much more extensively researched [**SdS01, DMA02**] are needed. For the source surface the flattening is relatively easy to perform as the feature is already selected and homeomorphic to a disk. For the target surface the problem is more complicated as the surface can have any shape and can be quite large. In order to avoid flattening the entire target surface, the authors propose a method for approximating the portion of the target surface that is actually involved in pasting. As the user specifies a target location where the feature is to be pasted, the goal is to find a region that resembles the source feature in shape and size. To identify such a region, a generalized radial parameterization of the feature boundary is used [**BMBZ02**] (see also Figure 16). Once such a region is found, the transfer of the source feature onto the target model is done by aligning the planar parameterizations of the source and target regions followed by resampling the source feature onto the target connectivity.

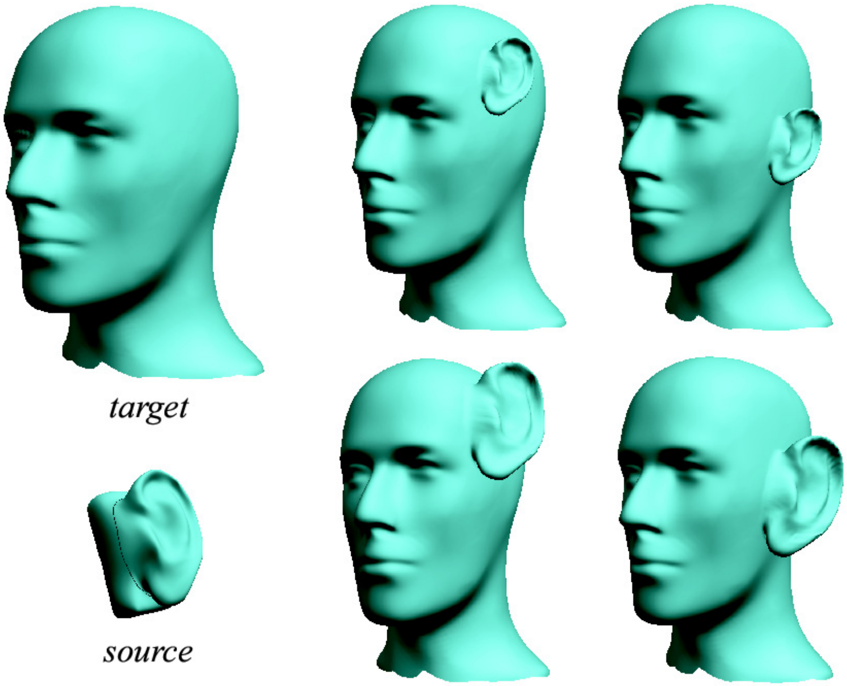


FIGURE 14. Pasting and interactively placing a complex feature: a digitized model of a clay ear constitutes the feature to be pasted onto the mannequin head. The ear can be interactively scaled, rotated, and translated on the surface of the head.

The pasting method of Biermann et al. [**BMBZ02**] provides a robust and efficient pipeline for interactive surface pasting. Its main constraints are related to self-intersections that may appear when features are pasted onto highly curved surfaces and to topological constraints on the features that can be pasted (i.e., only features with disk topology are handled). The former can be solved using a hierarchical pasting approach, in which the feature to be pasted is decomposed into frequency bands and the pasting is performed progressively, by pasting low-frequency details first, and high-frequency ones on top. The latter problem is more complex and requires more careful handling. A possible solution, albeit outside the subdivision framework, has been recently proposed in [**FMMY03**]. In this case, a volumetric approach is used to parameterize the feature and B-spline fitting is used to separate base from details. The advantage lies in the generality of features that are handled, including higher genus ones and the ability to paste them on highly curved areas. The main drawbacks are related to B-spline fitting and the need to introduce a large number of points in order to obtain a good fit. The result is not a seamless representation, but rather a composite one consisting of the original and the pasted part. In addition, the feature cannot be interactively dragged on the target surface.

3.2.3. *Surface Trimming.* Trimming, i.e., cutting holes in the surface of an object along specified curves, can also be considered as an instance of a Boolean

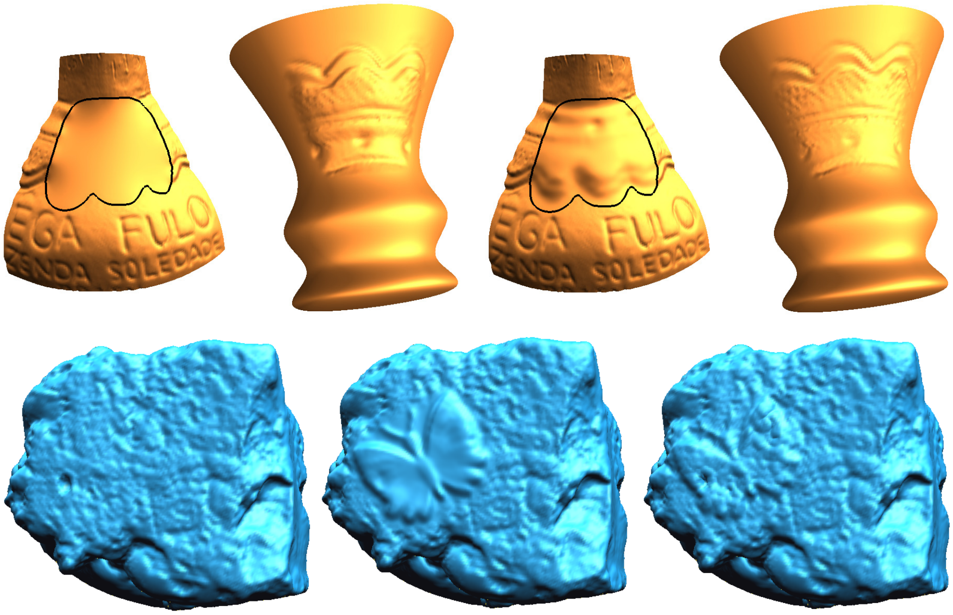


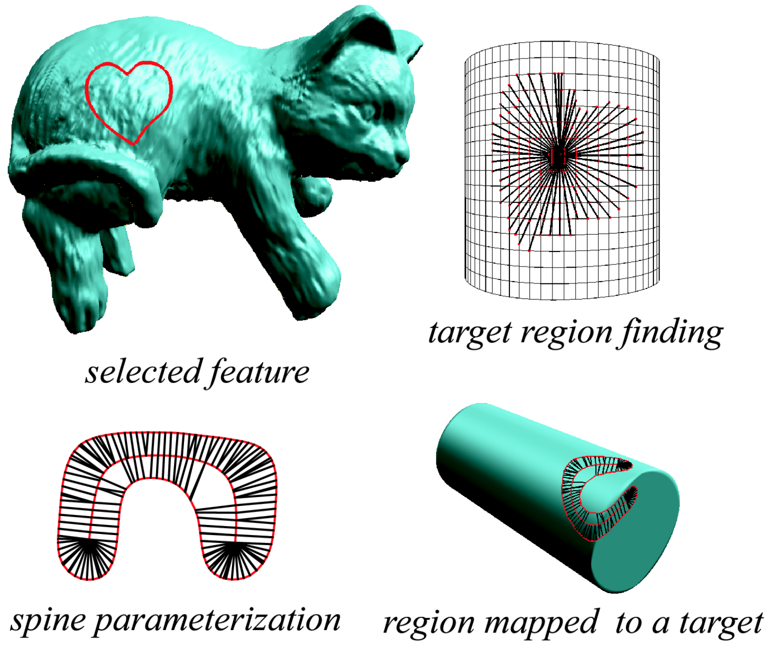
FIGURE 15. The effects of changing the base surface on the result of pasting: (top) digitized bottle detail appears on the vase differently, depending on the choice of base surface; (bottom) the butterfly feature is pasted on a rock model with and without preservation of target detail.

operation. Since this type of operation requires special subdivision rules along the trim boundary, we classify it as a special case of a non-smooth feature and we discuss it in section 3.3.

3.3. Non-Smooth Features. Subdivision surfaces can be naturally used to model smooth surfaces of arbitrary topological type. Many real objects, however, exhibit non-smooth features, such as sharp edges and boundaries, corners, and darts. While multiresolution detail vectors may be used to approximate sharp features (see Figure 17), a different setting is required to represent such features exactly. It entails altering the subdivision rules to produce limit surfaces that are only piecewise smooth, i.e., consist of smooth patches joined together along possibly sharp boundaries. To represent piecewise smooth surfaces, control mesh edges and vertices are typically *tagged* for special handling (see Figure 18). Special subdivision rules are employed in the vicinity of tagged mesh elements so as to avoid smoothing them. An edge can be tagged as a *crease edge* and vertices incident to crease edges may be tagged as one of the following (see Figure 18):

- *crease vertex*: exactly two crease edges join smoothly at this vertex
- *corner vertex*: two or more crease edges join non-smoothly at this vertex
- *dart vertex*: exactly one crease edge is adjacent to this vertex

Early mention of special rules for surfaces with boundaries appeared in the work of Doo [**Doo78**] and Nasri [**Nas91**], however accompanied only by partial analyses of the resulting surfaces. The first rules leading to provably C^1-continuous



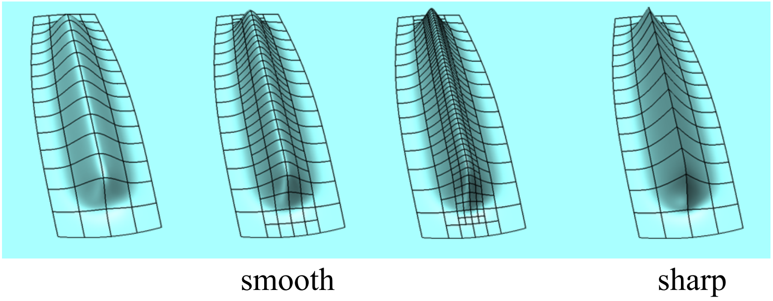
selected feature

target region finding

spine parameterization

region mapped to a target

FIGURE 16. Finding a target region through radial parameterization of the feature outline.



smooth

sharp

FIGURE 17. Multiresolution details are required to approximate sharp features using smooth surface representations (left three images). A piecewise smooth surface representation (right) allows sharp features to be modeled with detail vectors.

surfaces were defined in [**HDD$^+$94**] as a generalization of the Loop subdivision rules [**Loo87**]. The analysis of the resulting surfaces can be found in [**Sch96**]. As pointed out in [**BLZ00**], the rules introduced in [**HDD$^+$94**] have two main drawbacks: they are not suitable for modeling concave corners and the shape of the generated surface boundaries depends on the number of interior control points adjacent to each boundary point. The latter leads to undesirable gaps between surfaces joined along such a boundary. Both problems were handled by Biermann et al. [**BLZ00**] for the Loop [**Loo87**] and Catmull-Clark [**CC78**] subdivision schemes.

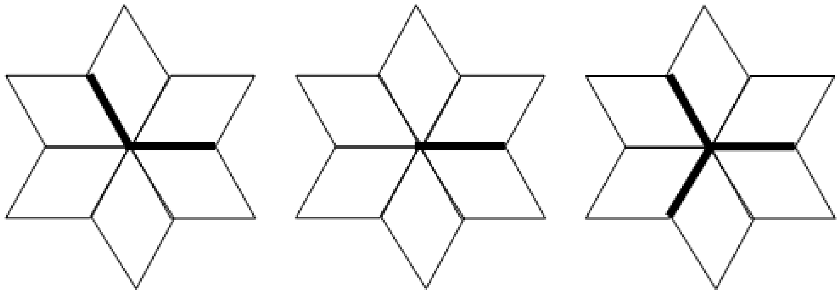


FIGURE 18. Mesh tags corresponding to (from left to right): crease, dart, and corner sharp features.

Modifications are sometimes applied to subdivision rules to achieve different effects. For example, deRose et al. [**DKT98**] propose an *edge sharpness* parameter s to vary sharpness along an edge and to allow for different degrees of sharpness. The parameter is used to blend between the positions p^{smooth} of a control point obtained with the smooth subdivision rules and a point p^{sharp} obtained with sharp subdivision rules:

$$p^{new} = (1-s)p^{smooth} + sp^{sharp},\ s \in [0,1]$$

Biermann et al. [**BLZ00**] propose a *flatness* parameter f and a *normal modification*. The flatness parameter control the speed at which control points in a neighborhood converge to the tangent plane. The subdivision rules are modified to blend between control point positions obtained without flatness modification and points in the tangent plane (p denotes the vector of control points in the neighborhood of a point,a_0, a_1, a_2 are the limit position and tangents at that point, and x^i denote the right eigenvectors of the subdivision matrix):

$$p^{new} = (1-f)p + f(a_0x^0 + a_1x^1 + a_2x^2),\ f \in [0,1]$$

The normal modification is somewhat similar, in that it interpolates between the control point position obtained without the modification and positions in a prescribed tangent plane (a normal n is prescribed through a pair of tangent vectors computed as $a_i' = (a_in)n$):

$$p^{new} = p + t((a_1' - a_1)x^1 + (a_2' - a_2)x^2),\ t \in [0,1]$$

Examples of sharp features modeled as proposed by Biermann et al. [**BLZ00**] are illustrated in Figure 19. A software library for piecewise smooth subdivision based on these rules is freely available from [**BZ99**].

A generalization of the subdivision concept that accommodates sharp features was developed by Sederberg et al. [**SZSS98**]. By drawing an analogy between recursive subdivision schemes and knot insertion for B-splines, the authors propose non-uniform versions of the Doo-Sabin [**Doo78**] and Catmull-Clark [**CC78**] subdivision schemes (under the general denomination of *non-uniform recursive subdivision surfaces* or NURSS). Each edge in a non-uniform Catmull-Clark control mesh (each control point in a Doo-Sabin mesh) is assigned a *knot spacing*. When all knot spacings are equal, the standard schemes are obtained. Two types of subdivision rules have to be considered for NURSS: the usual refinement scheme for the geometric positions of control points and an additional refinement scheme for knot spacings. Sharp features can be generated by setting certain knot spacings to zero.

The methods described so far require sharp features to be aligned with control mesh edges. Moreover, they provide little control over the profile of the resulting features. To address these limitations, Khodakovsky et al. [**KS99**] propose a curve-based feature editing approach. Feature curves are defined directly on the model surface through user interaction and can follow arbitrary paths, unconstrained by the connectivity of the underlying mesh. Features are obtained by perturbing the surface in the vicinity of feature curves. The curves can exist on multiple levels of a subdivision hierarchy. At each level, perturbations are computed with respect to local frames, so any coarse level modifications of the surface are carried through to finer levels. Several parameters are used to control the profile of a feature. In particular, sharp features can be obtained by specifying different normal directions for the profile on either side of the curve. This method brings forth a number of significant contributions with respect to previous approaches: it takes advantage of the multiresolution setting to define features through detail vectors at different levels, it does not impose any restrictions on the location of the feature curves on the surface or on their topology (curves can intersect or self intersect), and varying profiles allow both smooth and sharp features to be represented. The main drawback is that it does not preserve the input representation: after editing, the result is no longer a pure multiresolution subdivision surface, but rather a combined representation, consisting of a surface and a curve. This means that other subdivision-based tools that require as input a pure multireolution representation cannot be directly applied to the result of an editing operation performed with this method.

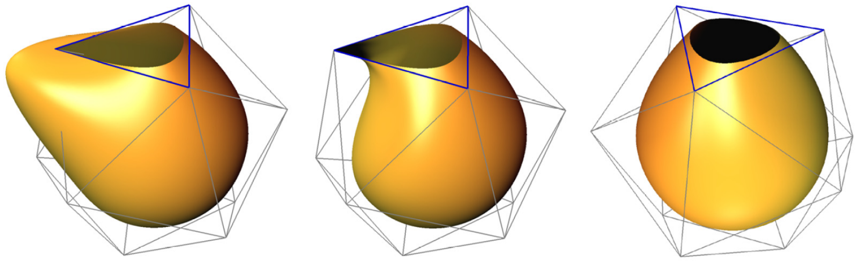


FIGURE 19. Sharp features generated with the rules proposed in [**BLZ00**]. From left to right: concave corner, convex corner, and smooth crease.

This problem is solved in [**BMZB02**] which uses the reparameterization idea described in section 3.2.1 to align the parameterization of the surface with the feature curves. Subsequently, sharp subdivision rules can be used along such curves. Figure 20 illustrates this process. An arbitrary feature curve is first projected onto the control mesh at some subdivision level (typically a coarse level which is subsequently refined). A piecewise linear approximation of the curve image in the parametric domain is computed by alternating *Snapping* and *Refinement* steps similar to those of Algorithm 1. The *Snapping* step moves mesh vertices onto the curve if they are sufficiently close, while the *Refinement* step subdivides the parameterization linearly. If $c : [0,1] \to X$ denotes the image of a feature curve in the parameter domain X of the goal is to reparameterize the domain X such that c passes through the vertices of X. This reduces to finding a one-to-one mapping $\Pi : X \to X$ which maps vertices of X to curve points: $\Pi(v_i) = c(t_i)$, for some vertices $\{v_0, v_1, \dots\}$ and curve parameters $\{t_0, t_1, \dots\}$. After a finite number of iterations of snapping and refinement, the resulting curve $[v_0, v_1, \dots]$ is guaranteed to have the same topology as c and to follow along mesh edges (and / or diagonals in the case of the Catmull-Clark scheme). After reparameterization, the input surface is resampled according to the new parameterization. Intuitively, this moves the control mesh on the surface and places mesh vertices on the feature curve. Subsequently, the actual feature can be created by tagging the appropriate mesh edges and applying sharp subdivision rules.

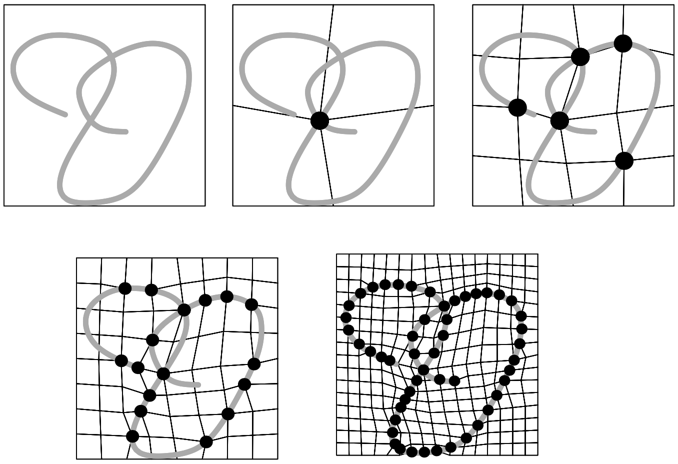


FIGURE 20. Reparameterization for approximating a feature curve: quads in parameter domain are recursively split and vertices are snapped to the curve. After several subdivision steps the curve is approximated by a sequence of vertices and follows along quad edges or diagonals.

In the case of Catmull-Clark meshes, there is an additional complication: the feature curve may pass through mesh diagonals after reparameterization (see Figure 21) and the standard crease rules do not support this situation. Biermann et al. [**BMZB02**] introduce new subdivision rules to deal with creases along quad diagonals. Sample results obtained with this method are shown in Figures 22. Note that the output surface is a multiresolution subdivision surface which can be manipulated with other tools designed to operate on such a representation. In addition, the framework is suitable not only for creating interior sharp features with various profiles (e.g., engravings, embossings), but also to create boundaries, i.e., to trim the input surface along the feature curves. An example of a trimmed surface is shown in Figure 22.

For completeness, we also mention the trimming method proposed by Litke et al. [**LLS01b**] which is complementary to that of Biermann et al. [**BMZB02**]. In this case quasi-interpolation is used to approximate a trimmed surface with a combined subdivision surface [**Lev99**].

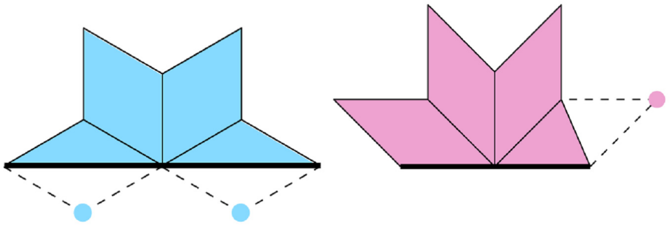


FIGURE 21. Standard sharp rules do not cover cases when the sharp edge (thick line) passes through a quad diagonal. New rules are necessary for such cases. Dotted lines and circles indicate vertices obtained by reflection used to define subdivision rules for such cases (see [**BMZB02**] for details).

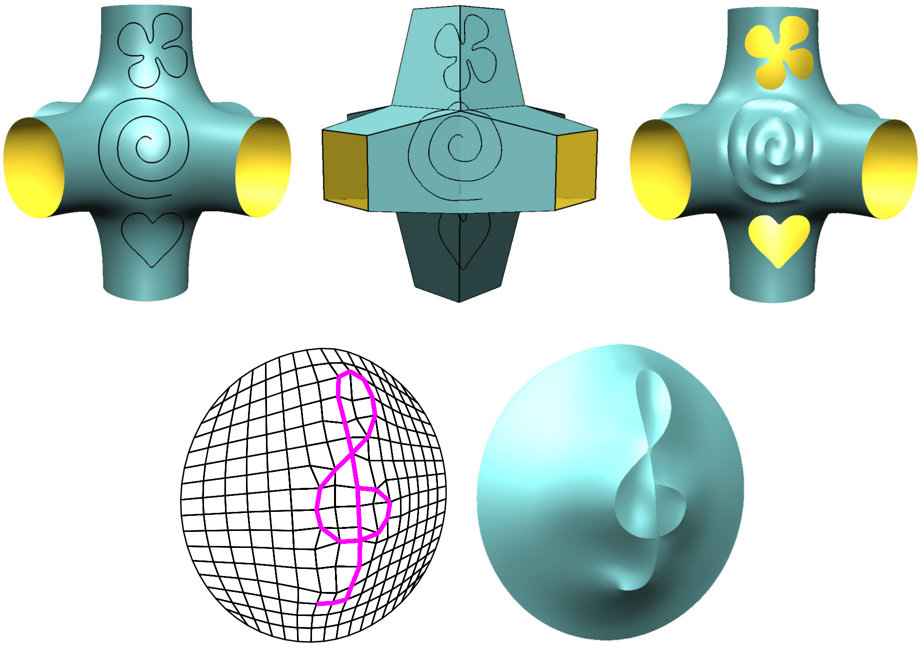


FIGURE 22. Surfaces obtained after trimming and embossing with sharp features using the method described in [**BMZB02**]. Top: input curves are shown on the surface (left) and projected into parameter space (middle). The surface obtained is shown on the right. Bottom: a self-intersecting feature.

4. Conclusions

In this paper we presented an overview of subdivision-based tools for surface design. We considered three major design categories: free-form deformations, Boolean operations, and surface decorations with sharp features. Such tools have multiple applications to animated character creation, virtual reconstructions and restorations, industrial design, shape modeling for medical and scientific simulations, etc. While modeling with subdivision surfaces has received a significant boost in recent years, there are a number of open issues still to be resolved. Among them are physical validation and interactive collision detection, guaranteed error bounds for surface fitting, and better integration between subdivision-based tools and methods operating on other representations.

References

[BAD$^+$01] M. Boo, M. Amor, M. Doggett, J. Hirche, and Wolfgang Strasser, *Hardware support for adaptive subdivision surface rendering*, Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware, 2001.

[Bar84] A. H. Barr, *Global and local deformations of solid primitives*, Proc. of SIGGRAPH 84 (1984), 21–30.

[BBF94] C. Barghiel, R. Bartels, and D. Forsey, *Pasting spline surfaces*, Mathematical Methods for Curves and Surfaces: Ulvik, Norway, Vanderbilt University Press, 1994, Available at ftp://cgl.uwaterloo.ca/pub/users/rhbartel/Paste.ps.gz, pp. 31–40.

[Béz74] P. Bézier, *Mathematical and practical possibilities of UNISURF*, CAD (1974), 127–152.

[BKS00] S. Bischoff, L. P. Kobbelt, and H.-P. Seidel, *Towards hardware implementation of loop subdivision*, Proc. of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics Hardware, 2000, pp. 41–50.

[BKZ01] H. Biermann, D. Kristjansson, and D. Zorin, *Approximate boolean operations on free-form solids*, Proceedings of SIGGRAPH 01, August 2001, pp. 185–194.

[BLZ00] H. Biermann, A. Levin, and D. Zorin, *Piecewise smooth subdivision surfaces with normal control*, Proceedings of SIGGRAPH 00, 2000, pp. 113–120.

[BMBZ02] H. Biermann, I. Martin, F. Bernardini, and D. Zorin, *Cut-and-paste editing of multiresolution surfaces*, ACM TOG. Special issue for SIGGRAPH conference **21** (2002), no. 3, 312–321.

[BMRB04] I. Boier-Martin, R. Ronfard, and F. Bernardini, *Detail-preserving variational surface design with multiresolution constraints*, Proc. Shape Modeling International, SMI'04, 2004.

[BMZB02] H. Biermann, I. Martin, D. Zorin, and F. Bernardini, *Sharp features on multiresolution subdivision surfaces*, Graphical Models **64** (2002), no. 2, 61–77.

[Bri87] W. L. Briggs, *Multigrid tutorial*, SIAM, 1987.

[BZ99] H. Biermann and D. Zorin, *Subdivide 2.0 software*, 1999.

[cat] http://www.catia.ibm.com.

[CC78] E. Catmull and J. Clark, *Recursively generated B-spline surfaces on arbitrary topological meshes*, CAD **10** (1978), no. 6, 350–355.

[CG99] G. Celniker and D. Gossard, *Energy-based models for free-form surface shape design*, Proc. ASME Design Automation Conference, 1999.

[CM00] B. Conrad and S. Mann, *Better pasting via quasi-interpolation*, Curve and Surface Design: Saint-Malo, 1999 (Nashville, TN) (P.-J. Laurent, P. Sablonnière, and L. L. Schumaker, eds.), Vanderbilt University Press, 2000, pp. 27–36.

[CMB97] L. K. Y. Chan, S. Mann, and R. Bartels, *World space surface pasting*, Proceedings of Graphics Interface (W. Davis, M. Mantei, and V. Klassen, eds.), May 1997, pp. 146–154.

[Coq90] S. Coquillart, *Extended free-form deformation: a sculpturing tool for 3d geometric modeling*, Proc. of SIGGRAPH 90, 1990, pp. 187–196.

[CR94] Y. Chang and A. P. Rockwood, *A generalized de casteljau approach to 3D Free-Form deformation*, Proc. of SIGGRAPH 94 (1994), 257–260.

[CRE01] E. Cohen, R. F. Riesenfeld, and G. Elber, *Geometric modeling with splines*, A K Peters Ltd, 2001.

[CSA$^+$02] F. Cirak, M.J. Scott, E.K. Antonson, M. Ortiz, and P. Schröder, *Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision*, Computer Aided Design **43** (2002), 137–148.

[CST94] M. M. Chang, M. I. Sezan, and A. M. Tekalp, *Adaptive bayesian segmentation of color images*, Journal of Electronic Imaging **3** (1994), 404–414.

[dF73] C. deBoor and G. J. Fix, *Spline approximtion by quasiinterpolants*, Journal of Approximation Theory **8** (1973), 19–45.

[DKT98] T. DeRose, M. Kass, and T. Truong, *Subdivision surfaces in character animation*, Proceedings of SIGGRAPH 98, 1998, pp. 85–94.

[DL02] N. Dyn and D. Levin, *Subdivision schemes in geometric modelling*, Acta Numerica **11** (2002).

[DLG90] N. Dyn, D. Levin, and J. A. Gregory, *A butterfly subdivision scheme for surface interpolation with tension control*, ACM TOG **9** (1990), no. 2, 160–169.

[DMA02] Mathieu Desbrun, Mark Meyer, and Pierre Alliez, *Intrinsic parameterizations of surface meshes*, Eurographics conference proceedings, 2002, pp. 209–218.

[DMR02] U. Diewald, S. Morigi, and M. Rumpf, *A cascadic geometric filtering approach to subdivision*, CAGD **19** (2002), no. 9, 675 – 694.

[Doo78] D. Doo, *A subdivision algorithm for smoothing down irregularly shaped polyhedrons*, Proceedings on Interactive Techniques in Computer Aided Design (Bologna), 1978, pp. 157–165.

[DS78] D. Doo and M. Sabin, *Analysis of the behaviour of recursive division surfaces near extraordinary points*, CAD **10** (1978), no. 6, 356–360.

[dsm] http://www.discreet.com.

[FMMY03] Y. Furukawa, H. Masuda, K. T. Miura, and H. Yamato, *Cut-and-paste editing based on constrained b-spline volume fitting*, Proceedings Computer Graphics International, 2003.

[FMS03] I. Friedel, P. Mullen, and P. Schroder, *Data-dependent fairing of subdivision surfacesn*, Proc. of SM 03, 2003.

[GKSS02] I. Guskov, A. Khodakovsky, P. Schröder, and W. Sweldens, *Hybrid meshes: Multiresolution using regular and irregular refinement*, Proc. Symp. Comp. Geom., 2002, pp. 264–272.

[GM97] S. F. F. Gibson and B. Mirtich, *A survey of deformable modeling in computer graphics*, Tech. Report TR-97-19, MERL, Cambridge, MA, 1997.

[GOP99] C. Gonzalez-Ochoa and J. Peters, *Localized-hierarchy surface splines (less)*, Proc. Symp. on Interactive 3D Graphics, 1999, pp. 7–15.

[Gre94] G. Greiner, *Surface construction based on variational principles*, Wavelets, Images and Surface Fitting (P. J. Laurent, A. LeMéhauté, and L. Schumaker, eds.), AK Peters, 1994, pp. 277–286.

[GS01] E. Grinspun and P. Schröder, *Normal bounds for subdivision-surface interference detection*, Proc. of IEEE Visualization 01, 2001.

[Hal96] M. A. Halstead, *Efficient techniques for surface design using constrained optimization*, Ph.D. thesis, Univ. of California at Berkeley, 1996.

[HDD$^+$94] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle, *Piecewise smooth surface reconstruction*, Computer Graphics Proceedings, Annual Conference Series, ACM Siggraph, 1994, pp. 295–302.

[HKD93] M. Halstead, M. Kass, and T. DeRose, *Efficient, fair interpolation using Catmull-Clark surfaces*, Proc. SIGGRAPH 1993, 1993, pp. 35–44.

[HQ03] J. Hua and H. Qin, *Free-Form deformations via sketching and manipulating scalar fields*, Proc. ACM Symp. on Solid Modeling and Applications, 2003, pp. 328 – 333.

[Kob96a] L. Kobbelt, *Interpolatory subdivision on openquadrilateral nets with arbitrary topology*, Proc. of Eurographics 96, 1996, pp. 409–420.

[Kob96b] L. Kobbelt, *A variational approach to subdivision*, Comput. Aided Geom. Design **13** (1996), no. 8, 743–761. MR **97j:**65029

[Kob00] L. P. Kobbelt, *Discrete fairing and variational subdivision for freeform surface design*, The Visual Computer **16** (2000), no. 3-4, 142–150.

[KS99] A. Khodakovsky and P. Schröder, *Fine level feature editing for subdivision surfaces*, Proceedings of ACM Solid Modeling, 1999.

[LCJ94] F. Lazarus, S. Coquillart, and P. Jancene, *Axial deformations: an intuitive deformation technique*, CAD **26** (1994), no. 8, 607–61.

[LDW97] M. Lounsbery, T. DeRose, and J. Warren, *Multiresolution analysis for surfaces of arbitrary topological type*, ACM TOG **16** (1997), no. 1, 34–73.

[Lev99] A. Levin, *Interpolating nets of curves by smooth subdivision surfaces*, Proc. of SIGGRAPH 99 (1999), 57–64.

[LFKN03] S. Lanquetin, S. Foufou, H. Kheddouci, and M. Neveu, *Computing subdivision surface intersection*, Proc. WSCG '03, 2003.

[Lin00] L. Linsen, *Netbased modelling*, Proc. SCCG '00, 2000, pp. 259–266.

[LLS01a] N. Litke, A. Levin, and P. Schröder, *Fitting subdivision surfaces*, Proc. of IEEE Visualization 2001, October 2001, pp. 319–324.

[LLS01b] N. Litke, A. Levin, and P. Schroder, *Trimming for subdivision surfaces*, Computer Aided Geometric Design **18** (2001), no. 5, 463–481.

[LMH00] A. Lee, H. Moreton, and H. Hoppe, *Displaced subdivision surfaces*, Proc. of SIGGRAPH 00, 2000, pp. 85–94.

[Loo87] C. Loop, *Smooth subdivision surfaces based on triangles*, Master's thesis, University of Utah, Department of Mathematics, 1987.

[Ma00] M. Ma, *The direct manipulation of pasted surfaces*, Master's thesis, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1, 2000, Available on WWW as ftp://cs-archive.uwaterloo.ca/cs-archive/CS-2000-15/.

[may] http://www.alias.com.

[MJ96] R. MacCracken and K. I. Joy, *Free-Form deformations with lattices of arbitrary topology*, Proc. of SIGGRAPH 96, 1996, pp. 181–188.

[MQ00] K.T. McDonnell and H. Qin, *Dynamic sculpting and animation of Free-Form subdivision solids*, Proc. of IEEE Computer Animation, 2000.

[MZ00] W. Ma and N. Zhao, *Catmull-clark surface fitting for reverse engineering applications*, Proceedings of Geometric Modeling and Processing, 2000, pp. 274–282.

[NA02] A. H. Nasri and A. Abbas, *Designing Catmull-Clark subdivision surfaces with curve interpolation constraints*, Computers and Graphics (2002).

[Nas91] A. H. Nasri, *Surface interpolation on irregular networks with normal conditions*, CAGD **8** (1991), 89–96.

[Nas00] A. Nasri, *Interpolating meshes of boundary intersecting curves by subdivision surfaces*, The Visual Computer **16** (2000).

[PB00] D. Piponi and G. Borshukov, *Seamless texture mapping of subdivision surfaces by model pelting and texture blending*, Proceedings of SIGGRAPH 00, 2000, pp. 471–478.

[Pet00] Jörg Peters, *Patching Catmull-Clark meshes*, Proceedings of SIGGRAPH 00, 2000, pp. 255–258.

[PL97] K. Pulli and M. Lounsbery, *Hierarchical editing and rendering of subdivision surfaces*, Tech. Report UW-CSE-97-04-07, Dept. of CS&E, University of Washington, Seattle, WA, 1997.

[PR97] J. Peters and U. Reif, *The simplest subdivision scheme for smoothing polyhedra*, ACM TOG **16** (1997), no. 4.

[PS96] K. Pulli and M. Segal, *Fast rendering of subdivision surfaces*, Proc. Eurographics Workshop on Rendering, 1996, pp. 61–70.

[QMV98] H. Qin, C. Mandal, and B. Vemuri, *Dynamic Catmull-Clark subdivision surfaces*, IEEE TVCG **4** (1998), no. 3, 215–229.

[QT96] H. Qin and D. Terzopoulos, *D-NURBS: A physics based framework for geometric design*, IEEE TVCG **2** (1996), no. 1, 85–96.

[Sab71] M. Sabin, *Interrogation techniques for parametric surfaces*, Advanced computer graphics - economics, techniques and applications (R. D. Parslow and R. E. Green, eds.), Plenum Press, 1971, pp. 1095–1118.

[Sab02] M. A. Sabin, *Subdivision surfaces tutorial*, SMI 02 (2002).

[Sch96] J. E. Schweitzer, *Analysis and application of subdivision surfaces*, Ph.D. thesis, University of Washington, Seattle, 1996.

[SDS96] E. J. Stollnitz, T. DeRose, and D. H. Salesin, *Wavelets for computer graphics: theory and applications*, Morgan Kaufmann, 1996.

[SdS01] A. Sheffer and E. de Sturler, *Parameterization of faceted surfaces for meshing using angle based flattening*, Engineering with Computers **17(3)** (2001), 326–337.

[Sei98] R. Seidel, *The nature and meaning of perturbations in geometric computing*, Discrete Computational Geometry **19** (1998), no. 1, 1–17.

[SF98] K. Singh and E. Fiume, *Wires: A geometric deformation technique*, Proc. of SIGGRAPH 98, 1998, pp. 405–414.

[SP86] T. W. Sederberg and S. R. Parry, *Free-form deformation of solid geometric models*, Proc. of SIGGRAPH 86, 1986, pp. 151–160.

[Sta98] J. Stam, *Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values*, Proceedings of SIGGRAPH 98, July 1998, pp. 395–404.

[STKK99] H. Suzuki, S. Takeuchi, F. Kimura, and T. Kanai, *Subdivision surface fitting to a range of points*, Proc. Pacific Graphics, 1999.

[SWZ04] S. Schaeffer, J. Warren, and D. Zorin, *Lofting curve networks using subdivision surfaces*, submitted (2004).

[SZSS98] T. W. Sederberg, J. Zheng, D. Sewell, and M. Sabin, *Non-uniform recursive subdivision surfaces*, Proceedings of SIGGRAPH 98, 1998, pp. 387–394.

[Tak98] S. Takahashi, *Variational design of curves and surfaces using multiresolution constraints*, The Visual Computer **14(5/6)** (1998), 208–227.

[TF88] D. Terzopoulos and K. Fleischer, *Modeling inelastic deformation: Viscoelasticity, plasticity, fracture*, Proc. of SIGGRAPH '88, 1988, pp. 269–278.

[TO02] G. Turk and J. O'Brien, *Modelling with implicit surfaces that interpolate*, ACM TOG **21** (2002), no. 4, 855 – 873.

[TPBF87] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, *Elastically deformable models*, Proc. of SIGGRAPH 87, 1987, pp. 205–214.

[TQ94] D. Terzopoulos and H. Qin, *Dynamic NURBS with geometric constraints for interactive sculpting*, ACM TOG **13** (1994), no. 2, 103–136.

[WW92] W. Welch and A. Witkin, *Variational surface modeling*, Proceedings of SIGGRAPH '92, vol. 26, 1992, pp. 157–166.

[WW98] H. Weimer and J. Warren, *Subdivision schemes for thin-plate splines*, Proc. EUROGRAPHICS '98 **17** (1998), no. 3.

[WW01] J. Warren and H. Weimer, *Subdivision methods for geometric design: A constructive approach*, Morgan Kaufmann, 2001.

[ZK02] D. Zorin and D. Kristjansson, *Evaluation of piecewise smooth subdivision surfaces*, The Visual Computer **18** (2002), no. 5–6, 299 – 315.

[ZSD$^+$00] D. Zorin, P. Schröder, T. DeRose, L. Kobbelt, A. Levin, and W. Sweldens, *Subdivision for modeling and animation*, SIGGRAPH'00 Course Notes, 2000.

[ZSS96] D. Zorin, P. Schröder, and W. Sweldens, *Interpolating subdivision for meshes with arbitrary topology*, Proc. of SIGGRAPH 96, August 1996, pp. 189–192.

[ZSS97] D. Zorin, P. Schröder, and W. Sweldens, *Interactive multiresolution mesh editing*, Proc. of SIGGRAPH 97, 1997, pp. 259–268.

Ioana Boier-Martin, IBM T. J. Watson Research Center, Hawthorne, NY 10532
E-mail address: ioana@us.ibm.com

Denis Zorin, Media Research Lab, New York University, New York, NY 10003
E-mail address: dzorin@mrl.nyu.edu

Fausto Bernardini, IBM T. J. Watson Research Center, Hawthorne, NY 10532
E-mail address: fausto@us.ibm.com