# Detail-Preserving Variational Surface Design
# with Multiresolution Constraints

Ioana Boier-Martin
IBM T. J. Watson Research Center
Hawthorne, New York, USA
ioana@us.ibm.com

Remi Ronfard
INRIA Rhône-Alpes
Montbonnot, France
remi.ronfard@inrialpes.fr

Fausto Bernardini
IBM T. J. Watson Research Center
Hawthorne, New York, USA
fausto@us.ibm.com

## Abstract

*We present a variational framework for rapid shape prototyping. The modeled shape is represented as a Catmull-Clark multiresolution subdivision surface which is interactively deformed by direct user input. Free-form design goals are formulated as constraints on the shape and the modeling problem is cast into a constrained optimization one. The focus of this paper is on handling multiresolution constraints of different kinds and on preserving surface details throughout the deformation process. Our approach eliminates the need for an explicit decomposition of the input model into frequency bands and the overhead associated with saving and restoring high-frequency detail after global shape fairing. Instead, we define a deformation vector field over the model and we optimize its energy. Surface details are considered as part of the rest shape and are preserved during free-form model editing. We explore approximating the solution of the optimization problem to various degrees to balance trade-offs between interactivity and accuracy of the results.*

## 1. Introduction

A common design paradigm is to allow designers to interactively deform an initial geometric shape to obtain a new one that satisfies certain requirements. The requirements are typically formulated as a set of constraints and the underlying geometric representation is modified to meet these constraints (see Figure 1). In general, the desired result is the one that has the most pleasing or fairest overall shape among all solutions that satisfy the constraints.

A commonly used procedure to attain this result is to optimize a fairness measure representing physical parameters of a real object bearing the shape. A standard such measure [26] is the linear combination of the so-called *stretch* and *bend* energies:

$$E = \alpha \int ||I||^2 dS + \beta \int ||II||^2 dS \qquad (1)$$

where $I$ and $II$ are the first and second fundamental forms of the surface $S$ and $||.||$ is a suitably chosen matrix norm.

It is often the case, however, that the input model has high-frequency geometric detail across multiple resolutions which needs to be preserved during global deformations of its shape. Fairing techniques like the one just described tend to smooth out not only the global shape of the object, but the high-frequency detail as well (see Figure 2).

To avoid this problem, multi-band decomposition schemes have been proposed [13], in which a multiresolution modeling operation and the associated fairing step are applied within one frequency band. Subsequently, the higher frequency detail is reconstructed using a displacement map. This implies computing and storing the displacement map prior to editing and restoring it afterwards. We opt for an alternative approach that avoids the computation of the displacement map and the overhead associated with saving and restoring high-frequency information by considering the deformations applied to the initial shape as a vector field defined over the input model. Instead of optimizing the energy of the deformed shape, we optimize the energy of the deformations and we apply the resulting smooth vector field to the original shape to obtain the deformed one. Using this approach, the input model becomes the rest shape to which the optimization converges in the absence of constraints. In the language of elastic body deformations, this is equiva-
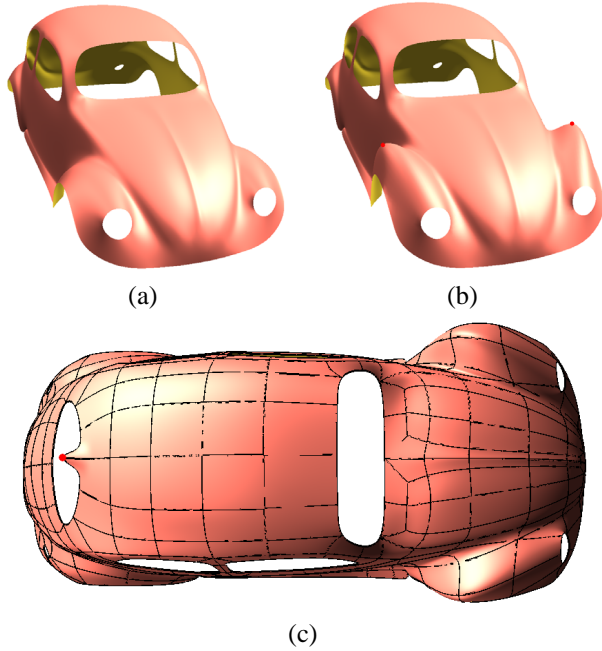
**Figure 1. Variational design with multiresolution constraints. (a) Input model. (b) Coarse-scale edits affect the global shape. (c) Fine scale edit with local effect (patch structure of underlying surface representation is shown). Red dots indicate constraints.**
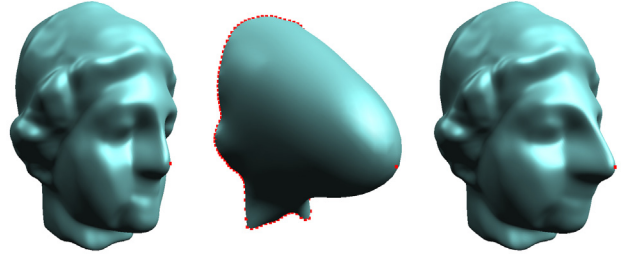


**Figure 2. Free-form modeling with constraints. (a) Original model with multiresolution details. A point constraint at the tip of the nose is used to deform the model. (b) The model is deformed using thin-plate energy minimization. Note the smoothing of the original details (boundary constraints are necessary to prevent the collapse of the model). (c) Deformation with detail preservation (no boundary constraints needed in this case).**

lent to considering the input shape as the *natural state* of the model [26].

In this paper we describe the design and implementation of a variational modeler that allows interactive editing of complex objects of arbitrary topology. We use Catmull-Clark multiresolution subdivision surfaces as our underlying representation and we take advantage of their hierarchical organization to allow editing with constraints at different resolution levels.

The main contributions of our research are:

1. A variational approach that leads to a smooth shape while preserving multiresolution details.

2. A framework for free-form design with constraints which can be imposed at different levels of resolution.

3. An implementation setup in which a fast approximate solution is computed at interactive rates during the deformation process. Solutions of increasing accuracy may follow upon user request.

The work presented in this paper adds a new technique to a growing set of surface modeling tools based on multiresolution subdivision surfaces that have been proposed in re-

cent years (e.g., boolean operations [1], engraving, embossing, trimming [3], cut-and-paste editing [2]).

The remaining sections of the paper are organized as follows: in section 2 we discuss related work, in section 3 we briefly review the underlying data representation and we outline our approach; modeling with constraints is presented in section 4; results are discussed and illustrated in section 5; section 6 concludes our paper.

## 2. Related Work

Variational design of surfaces has emerged as a powerful modeling paradigm. It entails finding a surface that satisfies a number of constraints and minimizes a given continuous functional that represents the energy of the surface. Expression (1) gives an example of a commonly used such functional. In practice, discrete approximations of each of its two terms are used.

Following the pioneering work of Duchon [6] and Meinguet [19], many authors (e.g., [5, 12, 32, 11, 27]) approximate surface energy by one or a combination of the following parameterization-dependent expressions:

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

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

where $\Omega$ denotes the parametric domain of the surface $S$. In our approach we also make use of these linearized forms of the stretch and bend energies. We note that other approximations, generally more expensive to compute, have also been proposed [20, 21, 8], as well as alternative energy functionals [16, 28]. Various types of constraints can be imagined. We address here some of the most common categories.

**Constrained optimization** Variational surfaces are often modeled by specifying a set of constraints and solving for the solution that minimizes the objective functional (1) or an approximation thereof and satisfies the constraints.

*Interpolating point constraints* prescribe desired positions for points on the surface. This type of constraints are considered by virtually all methods. In our approach we translate point constraints into linear combinations of mesh vertices and we perform energy minimization with linear constraints.

*Interpolating normal constraints* prescribe desired normal vectors at given points on the surface. A common approach (see, for example, [11]) is to formulate the fact that desired normal $N$ at a point $S(u_0, v_0)$ must be perpendicular to the vectors defined by the partial derivatives, i.e.:

$$N \frac{\partial S}{\partial u}(u_0, v_0) = 0 \text{ and } N \frac{\partial S}{\partial v}(u_0, v_0) = 0$$

These expressions can also be translated into linear constraints on the mesh vertices. Alternatively, some authors [15] enforce normal constraints by freezing all the vertices of a planar mesh face. While more stable, this approach tends to overly constrain the optimization problem.

*Interpolating curve constraints* prescribe desired positions at points along one or more curves on the surface. Such constraints typically require the constraint curves to be aligned with mesh edges or iso-parameter lines [27]. Using the re-parameterization idea of [3] we are able to avoid this requirement and to allow for constraints to be imposed along arbitrary curves.

**Multiresolution** We take advantage of our underlying hierarchical surface representation to handle constraints in a multiresolution fashion. An important issue to be addressed is identifying the *region of influence* of a given constraint. As observed in [22], the same constraint can be satisfied at a coarse scale by the global rigid motion of the entire surface or at a fine scale by the motion of an isolated surface point. The designer's intent usually lies somewhere in between. Except for the work of Takahashi [24, 25], this problem has received little attention in existing literature. Takahashi has developed a wavelet-based framework that accommodates linear constraints at multiple resolutions. His framework, however, is limited to topological patches. Surfaces of arbitrary topological type are handled by "gluing" patches and only simple examples are documented. In our work, we use

a similar approach in which we propagate constraints from fine to coarse scales. However, we give the user explicit control over the region of influence of each constraint and we rely on subdivision rules to perform constraint restriction in a more straightforward fashion. Arbitrary topology is handled automatically due to the nature of the surface representation we use.

Another relevant question is how to reconcile different energy measures computed at different resolutions. In [24, 25] they are combined into a weighted sum of energy functions at multiple levels. This approach causes undesirable side-effects of constraints at coarse resolutions over the shape at finer levels. A recursive scheme of solving the shape level-by-level is used to avoid the interactions between constraints at different levels. Instead, we have chosen to fix a target resolution level at which the energy is minimized and to use the solutions at coarser resolutions as approximations in a multigrid fashion.

**Subdivision and variational design** Halstead et al. [11, 10] were among the first to propose a variational modeling approach in the context of subdivision surfaces. An alternative re-formulation using wavelets was introduced by Gortler and Cohen [7]. Recent work by Warren, Weimer, Kobbelt, and Schröder [29, 30, 14, 15] emphasizes the relationship between variational methods and subdivision. *Variational subdivision* seeks to define subdivision rules that produce a sequence of shapes that not only converges to a limit shape that follows the initial control shape, but also minimizes the energy functional associated with the limit surface.

An appealing aspect of subdivision hierarchies which we exploit for efficiency is that they naturally accommodate multigrid-type solvers [9]. The basic operations, i.e., *restriction* and *prolongation* can be easily formulated as local masks. The restriction operator maps the data from a fine level to a coarser level. When applying a deformation at the finer level, the main question to be addressed is how project it onto coarser levels. The prolongation operator achieves the inverse mapping, from coarse to fine. We apply the Catmull-Clark subdivision masks for this purpose, in contrast with the approach suggested in [31] in which special-purpose prolongation operators are devised.

## 3. Multiresolution Variational Design

We reformulate the optimization problem previously introduced in a discrete multiresolution setting. To justify our choices, we begin with an overview of the underlying representation used. Basic concepts related to variational calculus can be found in any standard textbook (e.g., [17]).
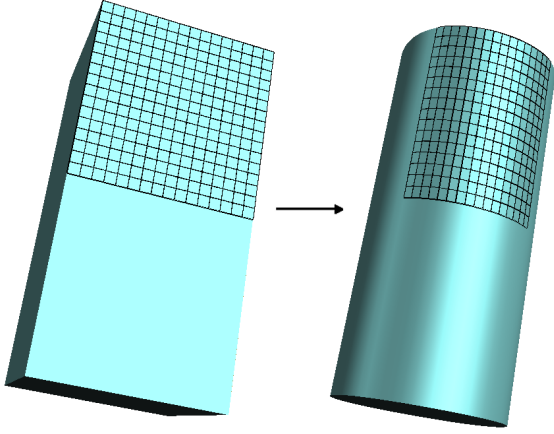
**Figure 3. Natural parameterization of a subdivision surface. Each time we apply the subdivision rules to compute the finer control mesh we also apply midpoint subdivision to a copy of the initial control mesh. As we repeatedly subdivide, we get a mapping from a denser and denser subset of the control polyhedron (left) to the control points of a finer and finer control mesh (right). In the limit we get a map from the control polyhedron to the surface.**

## 3.1. Multiresolution Subdivision Surfaces

The representation we use was introduced by several authors in different forms [18, 23, 33]. Subdivision defines a smooth surface recursively as the limit of a sequence of meshes. Each finer mesh is obtained from a coarse mesh by using a set of fixed refinement rules. In the work described in this paper we use the Catmull-Clark rules [4]. 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.

For our purposes, it is important to recognize that a multiresolution subdivision surface can be naturally interpreted as a function on the domain defined by the base mesh (see Figure 3). This interpretation is useful in many circumstances, including dealing with constraints along arbitrary curves as described in section 4.

It is, however, a known fact that the first and second order partial derivatives of the surface with respect to the nat-

ural parameterization diverge around extraordinary points. Therefore, for the purpose of evaluating expressions (2) we define a different parameterization as described in section 3.3.

## 3.2. Problem Formulation

Let $\mathcal{H} = (M^0, M^1, \cdots, M^{L-1})$ denote a multiresolution subdivision hierarchy with $L$ levels such that the control mesh $M^l$ at each level $l$ is obtained from the coarser mesh $M^{l-1}$ by subdividing it and adding detail offsets. Let $P^l = \{P_i^l\}, i = 0, \cdots, N-1$ denote the vertices of $M^l$. By applying quadrature formulas to discretize the integrals in (2), we obtain a discrete formulation of the energy associated with $M^l$:

$$E(M^l) = E(P^l) =$$
$$\alpha \sum_i E_{stretch}(P_i^l) + \beta \sum_i E_{bend}(P_i^l)) \qquad (3)$$

where

$$E_{stretch}(P_i^l) = ||\frac{\partial M_i^l(u,v)}{\partial u}||^2 + ||\frac{\partial M_i^l(u,v)}{\partial v}||^2$$
$$E_{bend}(P_i^l) =$$
$$||\frac{\partial^2 M_i^l(u,v)}{\partial u^2}||^2 + 2||\frac{\partial^2 M_i^l(u,v)}{\partial u \partial v}||^2 + ||\frac{\partial^2 M_i^l(u,v)}{\partial v^2}||^2$$

and $M_i^l(u,v)$ denotes the parameterization of $M^l$ around vertex $P_i^l$.

In the presence of constraints on subsets of the vertices $\{P_i^l\}$, the constrained energy optimization problem for level $l$ becomes:

$$E(P^l) \rightarrow min$$
$$f_k^l(P_{i_1}^l, \cdots, P_{i_k}^l) = C_k^l, k = 1, \cdots, m_l$$

with $C_k^l$ the target value of the $k^{th}$ constraint on level $l$. In this paper we restrict our attention to cases in which the constraints $f_k^l$ are linear combinations of the vertices $P_{i_j}^l$.

As mentioned in the previous section, instead of combining energies defined at different resolutions, we set as our goal the minimization of the energy of the finest level mesh $E(P^{L-1})$. To efficiently handle the optimization problem at this level, we use solutions from coarser levels as part of a multigrid approach.

## 3.3. Local Parameterization

In section 3.1 we pointed out that the natural parameterization induced by subdivision over the base control mesh is not suitable for evaluating partial derivatives everywhere on the surface. While feasible, finding a global parameterization that satisfies certain smoothness requirements is not a simple task. Fortunately, for the purpose of the work presented here, local parameterizations that allow us to approximate first and second order derivatives with divided differences are sufficient. We have opted for using local quadratic

polynomial interpolants as done in [15]. We briefly review this approach next.

To compute divided differences in the vicinity of a vertex $P_0^l$, a quadratic interpolating polynomial is fitted in least-squares sense to the local geometry defined by $P_0^l$ and its immediate neighbors (see Figure 4):

$$Q(u,v) =$$
$$Q + uQ_u + vQ_v + \frac{1}{2}u^2 Q_{uu} + uvQ_{uv} + \frac{1}{2}v^2 FQ_{vv}$$

To solve this problem we need at least six interpolation conditions which we formulate by assigning local parameter values $(u_i, v_i)$ to $P_0^l$ and its one-ring neighbors (we consider both edge and face neighbors). Since our underlying representation is a quadrilateral mesh, each vertex (interior or on the boundary) has at least five neighbors, with the exception of boundary vertices of valence one when only three direct neighbors exist. In such cases, we compute a least norm solution. Following [15], we assign coordinates $(0,0)$ to $P_0^l$ and

$$(u_i, v_i) =$$
$$||P_i^l - P_0^l|| \left( \cos\left( \sum_{j \in R(0)} \alpha_j \right), \sin\left( \sum_{j \in R(0)} \alpha_j \right) \right)$$

to its neighbors, where $R(0)$ denotes the set of indices of vertices in the one-ring of $P_0^l$ and

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

The least-squares solution obtained by solving the interpolation problem yields values for the partial derivatives:

$$[Q_u, Q_v, Q_{uu}, Q_{uv}, Q_{vv}]^T =$$
$$(\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T [\cdots, P_i^l - P_0^l, \cdots]^T$$

where

$$\mathbf{\Phi} = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ u_i & v_i & \frac{1}{2}u_i^2 & u_i v_i & \frac{1}{2}v_i^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

In the remainder of the text, we denote by $\mathbf{D^{i,l}} = \left[ D_{m,k}^{i,l} \right]_{m=\overline{1,5}, k \in R(i)} = (\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T$ the coefficients of the divided difference operators corresponding to vertex $P_i^l$.

### 3.4. Discrete Energy Formulation

By replacing the partial derivatives in the energy expression (3) with divided differences computed from the local parameterization, we obtain our discrete energy formulation:
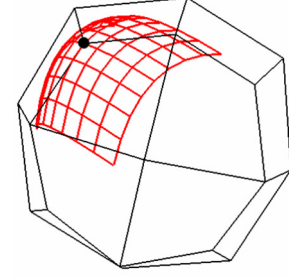


**Figure 4. Local quadratic interpolant used to approximate first and second order derivatives.**

$$E(P) = \sum_i \sum_{j,k \in R(i)} E_{ijk} (P_j - P_i)^T (P_k - P_i)$$

where the level index $l$ is omitted to simplify the notation, and the coefficients $E_{ijk}$ are defined as follows:

$$E_{ijk} = \alpha E_{ijk}^{stretch} + \beta E_{ijk}^{bend}$$
$$E_{ijk}^{stretch} = D_{1,j}^i D_{1,k}^i + D_{2,j}^i D_{2,k}^i,$$
$$E_{ijk}^{bend} = D_{3,j}^i D_{3,k}^i + 2D_{4,j}^i D_{4,k}^i + D_{5,j}^i D_{5,k}^i$$

In matrix form, this is equivalent to

$$E(P) = \frac{1}{2} P^T \mathbf{H} P,$$

where $\mathbf{H}$ is an $N \times N$ matrix. The minimum of $E(P)$ is found by setting all partial derivatives with respect to $P_i$ to zero and solving the corresponding system:

$$\frac{\partial E(P)}{\partial P_i} = 0, i = 0, \cdots, N-1$$

or equivalently:

$$\nabla E(P) = \mathbf{H}P = 0 \qquad (4)$$

Since the functional $E(P)$ is quadratic in every vertex, the system (4) is linear. In the next section we describe a strategy for solving it taking into account constraints.

## 4. Detail-Preserving Modeling with Constraints

### 4.1. Shape Deformation

We regard the deformations applied to a given mesh $M$ as vector offsets with respect to the original vertex positions:

$$\Delta P_i = P_i^{deformed} - P_i^{original}$$

Instead of minimizing the energy of the deformed mesh $E(P^{deformed})$, we would like to minimize solely its change in stretching and bending with respect to the initial shape.
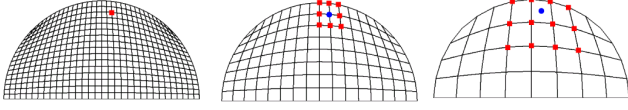
**Figure 5. Constraint propagation from a fine level (left) to coarser ones. Constrained vertices are shown as (red) squares. The target value of the constraint is marked with a (blue) circle.**

Thus, our constrained optimization problem at resolution level $l$ becomes:

$$E(\Delta P^l) \to min$$
$$f_k^l(\Delta P_{i_1}^l, \cdots, \Delta P_{i_k}^l) = C_k^l, k = 1, \cdots, m_l$$

Before we present our solution to this problem, we describe the types of constraints we consider and how they are propagated and enforced at different resolutions.

### 4.2. Constraints

We model shape by prescribing points, normals, and curves that should be interpolated by the surface. The energy optimization model discussed in the previous section defines the behavior of the shape in the regions without constraints so that the designer does not need to directly specify the surface in these regions.

**Region of influence** In our multiresolution setting, constraints can be imposed at any level. However, since it is typically the actual surface that needs to interpolate the constraints and not the mesh at some intermediate level, we first project all constraints to the finest level (or, alternatively, the limit surface), and then we propagate them through the multiresolution hierarchy to coarser levels.

We define the *region of influence* of a constraint as the portion of the surface affected by the constraint. We control the region of influence by the coarse level to which a constraint is propagated. Hence, a constraint propagated to a relatively fine level in the hierarchy will have only *local* impact (see Figure 1 (c)), whereas a constraint propagated to a very coarse level will have an effect on the *global* shape of the model (see Figure 1 (b)).

**Point constraints** Figure 5 shows how positional constraints are generated. In general, if $C_i^{l+1}$ is the prescribed deformation at a mesh vertex $P_i^{l+1}$ on level $l + 1$, then a linear constraint is generated at level $l$ using the Catmull-Clark subdivision masks. This choice allows us to apply the Catmull-Clark subdivision rules to the deformation vectors to obtain a fast approximate solution during interactive manipulation. This solution is sub-

| |
|---|
| $(1 - \beta - \gamma)\Delta P_i^l + \frac{\beta}{k} \sum\limits_{2j \in R(i)} \Delta P_{2j}^l + \frac{\gamma}{k} \sum\limits_{2j+1 \in R(i)} \Delta P_{2j+1}^l$ $= C_i^{l+1}$ |
| $\frac{3}{4}\Delta P_i^l + \frac{1}{8}(\Delta P_0^l + \Delta P_1^l) = C_i^{l+1}$ |
| $\frac{1}{4}(\Delta P_0^l + \Delta P_1^l + \Delta P_2^l + \Delta P_3^l) = C_i^{l+1}$ |
| $\frac{3}{16}(\Delta P_0^l + \Delta P_3^l) + \frac{1}{8}(\Delta P_1^l + \Delta P_2^l + \Delta P_4^l + \Delta P_5^l)$ $= C_i^{l+1}$ |
| $\frac{1}{2}(\Delta P_0^l + \Delta P_1^l) = C_i^{l+1}$ |

**Table 1. Linear constraints are created at coarse levels using the Catmull-Clark rules. From top to bottom, the rules are for interior even vertex, boundary even vertex, inteior odd face vertex, interior odd edge vertex, and boundary odd vertex, respectively. Vertex indexing corresponds to Figure 6 ($k$ denotes vertex valence, $\beta = \frac{3}{2k}$, and $\gamma = \frac{1}{4k}$).**
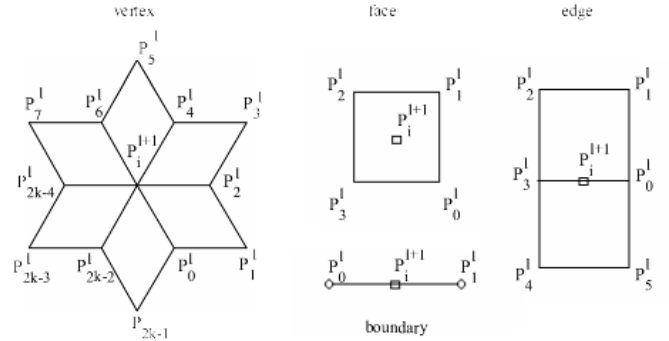


**Figure 6. Linear constraints are generated on coarse levels using the Catmull-Clark masks accoring to the expression listed in Table 1.**

sequently improved by using a solver as discussed later in this section. With the notations in Figure 6, the expressions for the linear constraints generated depending on the position of the constrained vertex in the mesh are given in Table 1.

**Normal constraints** Normal constraints are imposed by constraining two tangent vectors at a point to be perpendicular to the normal at that point. We use the vectors $Q_u$ and $Q_v$ estimated from the quadratic interpolant in place of the two tangents:

$$Q_u(P_i^l) \cdot N(P_i^l) = 0 \text{ and } Q_v(P_i^l) \cdot N(P_i^l) = 0$$

Using the notations from section 3.3, we see that these two conditions translate into linear equations involving the point $P_i^l$ and its immediate neighbors:

$$\sum_{j \in R(i)} D_{k,j}^{i,l}(P_j^l - P_i^l)^T N(P_i^l) = 0, \ k = 1, 2$$

The equivalent conditions on the deformations become:

$$\sum_{j \in R(i)} D_{k,j}^{i,l}(\Delta P_j^l - \Delta P_i^l)^T N(P_i^l) =$$

$$- \sum_{j \in R(i)} D_{k,j}^{i,l}(P_j^{l,original} - P_i^{l,original})^T N(P_i^l), \ k = 1, 2$$

**Curve constraints** Besides interpolation of prescribed points and normals, it may be useful to manipulate the surface by modifying the shape of embedded lower-dimensional entities. We restrict our attention to arbitrary curves lying on the surface. The most common example is modifying an open surface by rigidly manipulating its boundary curves (see Figures 9 (a) and (b)). In this case, we can sample the curves with boundary vertices and create collections of point constraints at these vertices. However, in the general case, a curve lying on the surface is not aligned with the underlying mesh edges, so a direct sampling with vertices is not possible. To allow for constraints along such curves, we re-parameterize the surface as in [3] to align it with the curve. The details of the re-parameterization are appended for convenience in the Appendix at the end of the paper. After re-parameterization, the original curve is approximated by a piecewise linear one that passes through mesh vertices. Point constraints are placed at vertices along the approximating curve and propagated through the multiresolution hierarchy as previously described.

### 4.3. Constrained Optimization

The simplest approach to solving our constrained minimization problem is to fix the deformation vectors at the constrained vertices to certain values and to solve an unconstrained energy minimization problem for the remaining ones. To determine the fixed deformations for the constrained vertices, we solve the linear system of constraints in the least-squares sense. Using matrix notation, we can write the system as:

$$\mathbf{A} \Delta P_c^l = C^l \qquad (5)$$

where $\Delta P_c^l$ denotes the vector of deformations corresponding to the constrained vertices at level $l$. The least-squares solution is given by:

$$\Delta P_c^l = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T C^l$$

We update the deformations associated with the unconstrained vertices iteratively, using a Gauss-Seidel approach ($\Delta P_i^{l,n}$ denotes the deformation at iteration $n$ and $\{H_{ij}\}$ are entries of the matrix $\mathbf{H}$):

$$\Delta P_i^{l,n+1} = \frac{1}{H_{ii}}(-\sum_{j<i} H_{ij} \Delta P_j^{l,n+1} - \sum_{j>i} H_{ij} \Delta P_j^{l,n})$$

We have also experimented with successive over-relaxation to accelerate convergence:

$$\Delta P_i^{l,n+1} =$$
$$(1-\omega)\Delta P_i^{l,n} + \frac{\omega}{H_{ii}}(-\sum_{j<i} H_{ij} \Delta P_j^{l,n+1} - \sum_{j>i} H_{ij} \Delta P_j^{l,n})$$

where $\omega$ is a relaxation parameter between $0$ and $2$. It has been our experience that choosing a suitable value for $\omega$ results in considerable speedup. The choice, however, is typically problem dependent.

## 5. Results and Implementation

Figures 7, 8, and 9 illustrate sample results obtained with our system. Figure 9 top shows the smooth deformation of a planar surface using constraints along the boundary. In particular, the vertices on the inner boundary are rigidly rotated as indicated by the yellow arcball and the surface follows naturally. Figures 9 bottom and 7 illustrate deformations of organic shapes rich in high-frequency detail. Note the preservation of this detail on the modified shapes, on both the cow and the vase models. Figure 8 illustrates different accuracy approximations of the solution to the optimization problem. Combining constraint propagation through the multiresolution hierarchy and applying Catmull-Clark rules as a smoothing operation yields a good initial approximation of the thin plate energy minimization (see Figure 8 (b)) and allows the user to assess the modified shape at interactive rates. An improved solution later computed using a multigrid approach is shown in (c). The shape at intermediate levels and the multiresolution constraints are shown in Figure 8 bottom.

We make several observations pertaining to our implementation.

**Point vs. normal constraints** In the case of point constraints (including those generated from curve constraints), the system (5) represents, in fact, a collection of three linear systems, one for each of the spatial dimensions of the deformation vectors. Unfortunately, our formulation of normal constraints violates this independence of spatial dimensions which is attractive from a computational point of view. If normal constraints are present in the optimization problem, we set up a single coupled system for all dimensions, and we solve the larger system. Each point constraint is represented by three equations in this system (one for each of the $x, y$, and $z$ components of a deformation). Each normal constraint translates into two equations as described in section 4.2. The resulting system is larger, and hence, more expensive to compute, however, it allows for more flexible design options.

**Multigrid optimization** Since the emphasis in our prototype is on interactivity, we explore approximating the solution of the optimization problem to various degrees to

balance interactivity and accuracy. During constraint manipulation, we use Catmull-Clark smoothing which yields a good approximation of the solution very fast (see Figure 8). At the end of a design step (e.g., upon mouse release), a more accurate solution is obtained through energy minimization. Finally, per user request, a fully converged solution will also be provided (typically at non-interactive rates).

To accelerate convergence of the iterative energy minimization process, we use a multigrid approach to compute the solution in a hierarchical fashion. We use a simplified coarse grid correction method consisting of two main steps. *Relaxation* iteratively minimizes the energy at level $l$ using Gauss-Seidel iterations as previously described. *Prolongation* is used to propagate the solution computed at level $l$ to the next finer level. We use Catmull-Clark subdivision as our prolongation operator. Note that we are not currently using restriction from fine to coarse as full multigrid implementations typically do.



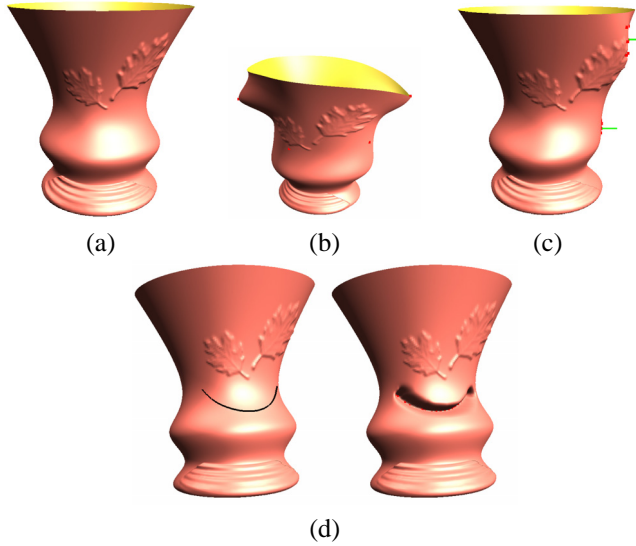(a)          (b)          (c)

(d)

**Figure 7. Modeling with detail preservation (a) Original model. (b) Deformation under point constraints (marked with red points). (c) Normal constraints (marked with green lines; red points indicate constrained region). (d) Constraints imposed along an arbitrary curve on the surface (left) are used to rigidly deform the surface in the vicinity of the curve (right).**



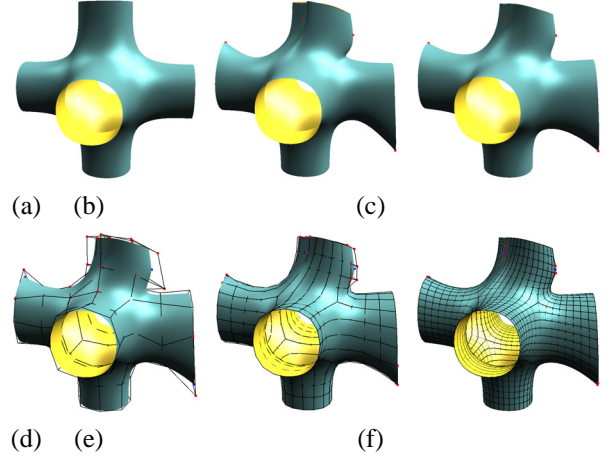(a)    (b)                    (c)

(d)    (e)                    (f)

**Figure 8. Top: approximating energy minimization. (a) Input model. (b) Fast solution obtained using Catmull-Clark smoothing. (c) Improved solution obtained via multigrid energy minimization. Bottom: (d)-(f) the optimized control mesh and the constraints at levels 2, 3, and 4 during multigrid.**

## 6. Conclusions and Future Work

In this paper we described a variational modeling approach as implemented in a system we have developed. Models are represented by multiresolution subdivision surfaces. During interactive deformations, their shapes are recomputed via energy optimization with constraints. Point, normal, and curve constraints are considered at multiple resolution levels. To preserve multiresolution details, we optimize only the energy of the deformations, instead of the total surface energy.

In our work we have adapted existing variational methods to multiresolution subdivision surfaces and in doing so, we have shown the advantages of using this representation for interactive free-form design. Some of the remaining issues to be solved include the derivation of better stopping criteria for the iterative solver (we are currently using a fixed number of iterations), a full-fledged multigrid implementation including restriction of deformations to avoid recomputing surface regions already optimized, as well as alternative solvers.

## Appendix: Re-Parameterization for Placing Constraints Along Arbitrary Curves

We briefly review the re-parameterization algorithm of [3] which we use for allowing constraints to be imposed along arbitrary curves on a surface.
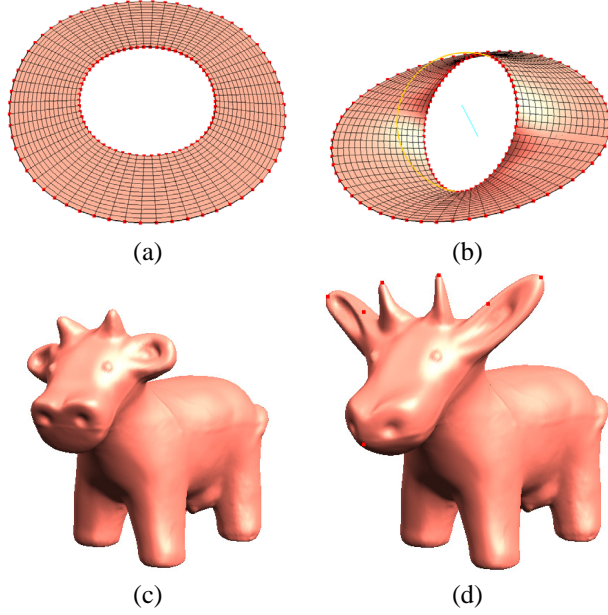
**Figure 9. Top: curve constraints along boundary. (a) Input model. (b) Surface after rotating the inner boundary curve and energy optimization (rotation arc and axis are superimposed). Bottom: interactive editing. (c) Input model with details at multiple resolutions. (d) The model in (c) after variational editing. Red points indicate constraints.**

As mentioned in section 2, we view the input surface as a parametric surface over the domain defined by the base control polyhedron. The main idea is to re-parameterize the surface to align the parameterization with a given curve or set of curves.

Let $X$ denote the parameter domain of the surface defined by its base mesh and let $c$ denote an input curve defined on $X$, $c : [0, 1] \rightarrow X$. In general, $c$ traverses the domain $X$ at arbitrary positions. We want to re-parameterize the domain $X$ such that $c$ passes through the vertices of $X$. Therefore, we compute a one-to-one mapping $\Pi : X \rightarrow X$ which maps vertices of $X$ to curve points: $\Pi(v_i) = c(t_i)$, for some vertices $\{v_0, v_1, \ldots\}$ and curve parameters $\{t_0, t_1, \ldots\}$. The mapping $\Pi$ is built to satisfy the following *approximation property* (AP):

(AP): *the piecewise linear curve $[v_0, v_1, \ldots]$ has the same topology as $c$ and either follows along mesh edges or crosses mesh faces diagonally.*

The re-parameterization algorithm alternates between *snapping* and *refinement* steps. The snapping step moves mesh vertices onto the curve if they are sufficiently close. In the refinement step we simply subdivide the parameter-

ization linearly. The algorithm terminates if the sequence of vertices $\{v_0, v_1, \cdots\}$ along $c$ satisfies the approximation property (AP). Property (AP) is guaranteed to be satisfied after a finite number of steps for piecewise-linear curves $c$.
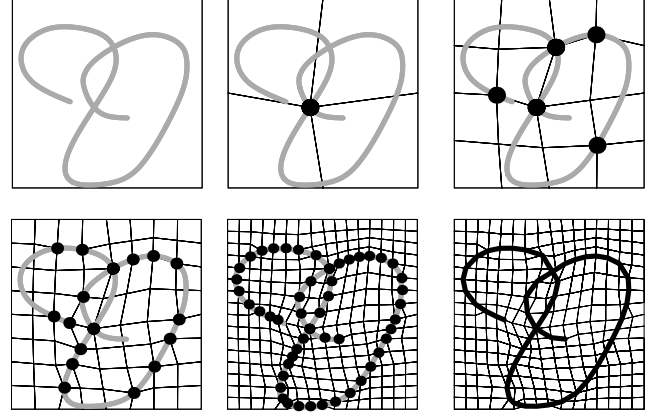


**Figure 10. Re-parameterization matching a feature curve. The quad is recursively split and vertices are snapped to the curve. After four subdivision steps, the curve is approximated by a sequence of mesh vertices. Constraints may be imposed at these vertices.**

## References

[1] H. Biermann, D. Kristjansson, and D. Zorin. Approximate boolean operations on free-form solids. In *Proceedings of SIGGRAPH 01*, pages 185–194. ACM SIGGRAPH, 2001.

[2] H. Biermann, I. Martin, F. Bernardini, and D. Zorin. Cut-and-paste editing of multiresolution surfaces. In *Proceedings of SIGGRAPH 02*, page to appear. ACM SIGGRAPH, 2002.

[3] H. Biermann, I. Martin, D. Zorin, and F. Bernardini. Sharp features on multiresolution subdivision surfaces. *Graphical Models*, 64(2):61–77, 2002.

[4] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. 10(6):350–355, 1978.

[5] G. Celniker and D. Gossard. Deformable curve and surface finite-elements for freeform shape design. In *Proceedings of SIGGRAPH 91*, pages 257–265. ACM SIGGRAPH / Addison Wesley, 1991.

[6] J. Duchon. *Splines minimizing rotation-invariant semi-norms in Sobolev spaces*. Springer–Verlag, 1977.

[7] S. J. Gortler and M. F. Cohen. Hierarchical and variational geometric modeling with wavelets. In *Proceedings Symposium on Interactive 3D Graphics*. Siggraph, May 1995.

[8] G. Greiner. Variational design and fairing of spline surfaces. *Computer Graphics Forum*, 13:143–154, 1994.

[9] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer–Verlag, 1985.

[10] M. Halstead. *Efficient Techniques for Surface Design Using Constrained Optimization*. PhD thesis, UC Berkeley, 1996.

[11] M. Halstead, M. Kass, and T. DeRose. Efficient, fair interpolation using Catmull-Clark surfaces. In *Computer Graphics Proceedings*, Annual Conference Series, pages 35–44. ACM Siggraph, 1993.

[12] M. Kallay. *Constrained optimization in surface design*. Springer–Verlag, 1993.

[13] L. Kobbelt, T. Bareuther, and H.-P. Seidel. Multiresolution shape deformations for meshes with dynamic vertex connectivity. In *Proceedings of Eurographics 00*, pages 249–260, 2000.

[14] L. Kobbelt and P. Schröder. A multiresolution framework for variational subdivision. *ACM Transactions on Graphics*, 17(4):209–237, October 1998. ISSN 0730-0301.

[15] L. P. Kobbelt. Discrete fairing and variational subdivision for freeform surface design. *The Visual Computer*, 16(3-4):142–150, 2000. ISSN 0178-2789.

[16] S. Kuriyama and K. Tachibana. Polyhedra surface modeling with a diffusion system. In *Proceedings of Eurographics 97*, pages 39–46, 1997.

[17] C. Lanczos. *The Variational Principles of Mechanics*. Dover Publications, 1986.

[18] M. Lounsbery, T. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *Transactions on Graphics*, 16(1):34–73, January 1997.

[19] J. Meinguet. Multivariate interpolation at arbitrary points made simple. *Journal of Applied Mathematics and Physics*, 30:292–304, 1979.

[20] H. Moreton. *Minimum Curvature Variation Curves, Networks, and Surfaces for Fair Free-Form Shape Design*. PhD thesis, UC Berkeley, 1993.

[21] H. Moreton and C. Sequin. Functional optimization for fair surface design. In *Proceedings of SIGGRAPH 92*, pages 167–176. ACM SIGGRAPH, 1992.

[22] L. F. Palazzi and D. L. Forsey. A multilevel approach to surface response in dynamically deformable models. In *Computer Animation Proceedings*, pages 21–30. Springer-Verlag, 1994.

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

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

[25] S. Takahashi. Multiresolution constraints for designing subdivision surfaces via local smoothing. In *Proceedings of Pacific Graphics 99*, pages 168–178. IEEE, 1999.

[26] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. In *Proceedings of SIGGRAPH 87*, pages 205–214. ACM SIGGRAPH / Addison Wesley, 1987.

[27] R. C. Veltkamp and W. Wesselink. Variational modeling of triangular bezier surfaces.

[28] O. Volpin, M. Bercovier, and T. Matskewich. A comparison of invarian energies for free-form surface construction. *The Visual Computer*, 15(4):199–210, 1999.

[29] J. Warren. Draft: Subdivision schemes for variational splines.

[30] H. Weimer. *Subdivision Schemes for Physical Problems*. PhD thesis, Rice University, 2000.

[31] H. Weimer and J. Warren. Subdivision schemes for thin plate splines. In *Proceedings of Eurographics 98*, pages 303–313, 1998.

[32] W. Welch and A. Witkin. Variational surface modeling. In E. E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 157–166, July 1992.

[33] D. Zorin, P. Schröder, and W. Sweldens. Interactive multiresolution mesh editing. *Proceedings of SIGGRAPH 97*, pages 259–268, August 1997. ISBN 0-89791-896-7. Held in Los Angeles, California.