Variational Modeling of Triangular Bezier Surfaces

Remco C. Veltkamp* Wieger Wesselink‡

* Utrecht University, Department of Computing Science
Padualaan 14, 3584 CH Utrecht, The Netherlands
Remco.Veltkamp@cs.ruu.nl

‡ Technical University of Eindhoven, Department of Computing Science
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
e-mail: wieger@win.tue.nl

Abstract

A constrained variational surface is a surface that minimizes some energy functional under certain interpolation constraints. Modeling surfaces using constrained variational principles is attractive, because the designer is not bothered with the precise representation of the surface (e.g. control points). Until now, the modeling of variational surfaces is mainly done by specifying a number of constraints. If such a surface of least energy is deformed locally (e.g. by moving its control points) the concept of energy minimization is lost. In this paper we introduce deform operators with built-in energy terms. Their effect is to move groups of control points to modify the shape of the surface, while the surface is still computed by minimizing a certain energy functional. We have tested our ideas in a prototype system for modeling surfaces of triangular Bézier patches. Experiments show that these operators offer an effective way of modeling.

1. Introduction

Variational modeling is a powerful way of modeling. It allows the design of very smooth surfaces satisfying a number of interpolation constraints in a very easy way. A variational surface minimizes a given functional that represents the energy of the surface. This functional typically depends on local properties of the surface such as the normal vector and curvature. It is called the internal energy of the surface. A well known example of such a functional is the bend energy. Because of its physical background, the bend energy has been used extensively (see e.g. [Witkin, 94]). Other energy functionals have been studied as well (see e.g. [Hagen and Schulze, 91], [Rando and Roulier, 91]). Through the choice of an internal energy the designer has a certain kind of global control over the shape of the surface.

Another basic way of modeling the shape of a variational surface is by means of specifying constraints (see e.g. [Celniker and Gossard, 91], [Kallay, 93], [Moreton and Séquin, 92], and [Welch and Witkin, 92]). The most common constraints are point and normal interpolation. Sometimes also curve interpolation constraints are used (see e.g. [Kallay, 93]). The interpolation constraints can be used to sketch a rough outline of a surface.

These two ways of modeling are useful, but sometimes they are not flexible enough. Flexibility can be provided by operators that deform the surface locally. In the case of control

point based surfaces (e.g. Bézier or B-spline surfaces) this can easily be done by repositioning a few control points. However, this does not fit in the concept of variational modeling, because the deformed surface does in general not minimize an energy functional.

This paper introduces some design operators that do not suffer from this problem. The effect of such an operator is defined by an energy functional. Since this functional depends on properties from outside the surface, it is called external. This has been proved to be successful for 3D curves [Veltkamp and Wesselink, 95], and is extended to surfaces in this paper. We have tested these concepts in a prototype system for modeling surfaces of cubic triangular Bézier patches. Our external energy operators are more general for design than only spring forces (see e.g. [Terzopoulos and Qin, 94]). Other surface representations used are tensor product surfaces, e.g. [Kallay, 93], [Moreton and Séquin, 92], [Welch and Witkin, 92], [Terzopoulos and Qin, 94], or triangular patches with rational weighting functions [Celniker and Gossard, 91]. A drawback of tensor product surfaces is that they cannot be subdivided locally.

The internal and external energies E_{int} and E_{ext} are combined into one functional, the total energy E_{tot} of the surface, which has to be minimized:

$$E_{tot} = E_{int} + E_{ext}. \tag{1}$$

In order to improve the speed of the computation the energy functionals are approximated with quadratic expressions. This results in a quadratic minimization problem with linear constraints, that is solved using a standard method.

2. Surface representation

We have used triangular Bézier patches. Spline surfaces composed of triangular patches have some advantages over rectangular patches (see [Farin, 93]). For instance, triangular patches are better suited to describe complex geometries, and they can be subdivided locally.

A triangular Bézier patch T is defined by Eq. (2) as the weighted sum of a number of control points $P_{i,j,k}$. Analogous to the univariate Bernstein polynomials over an interval, the Bernstein polynomials of degree n over a non-degenerate triangle $V_1V_2V_3$ are defined by

$$B^n_{i,j,k}(u,v,w) = \frac{n!}{i!j!k!}u^i v^j w^k, \qquad i+j+k=n, \quad i,j,k \in \mathbb{N},$$

where u, v, and w, with $u+v+w=1$, are barycentric coordinates with respect to triangle $V_1V_2V_3$. A parametric Bézier triangle is defined as:

$$T(u,v,w) = \sum_{i+j+k=n} P_{i,j,k} B^n_{i,j,k}(u,v,w), \tag{2}$$

where $P_{i,j,k}$ are the control points. If two patches join along a boundary curve, then the control points at the corresponding edges of the two control polyhedra should coincide. A continuous piecewise Bézier surface consists of a collection of abutting Bézier triangles.

The boundary of a patch consists of three univariate Bézier curves. For example for $w=0$ we get $v=1-u$, and thus

$$T(u,1-u,0) = C(u) = \sum_{i=0}^{n} P_{i,n-i,0} B^n_i(u), \tag{3}$$

with

$$B_i^n(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}, \quad i \in \mathbb{N}.$$

So, a boundary curve of a patch is defined by the control points on the corresponding boundary of the control polyhedron.

General patch subdivision splits a patch into several patches that together have the same shape as the original one. Subdivision algorithms for Bézier triangles are given by [Goldman, 83]. In this paper we will use two ways of subdivision, into two, and into three triangles.

3. Internal energy

The internal energy of a surface S is the part of the total energy (1) that depends only on properties of the surface itself. This section discusses the internal energy functionals that are used in our prototype system.

In [Terzopoulos et al., 87] the following functionals are defined for stretch energy E_s and bend energy E_b:

$$E_s(S) = \int \|G\|^2 \, dS, \tag{4}$$

$$E_b(S) = \int \|B\|^2 \, dS, \tag{5}$$

where G and B are the first and second fundamental forms of the surface S (see [do Carmo, 76]), and $\|.\|$ is a suitably chosen matrix norm. In [Celniker and Gossard, 91], [Kallay, 93] and [Welch and Witkin, 92] the following approximations are given to the stretch energy (4) and the bend energy (5), sometimes with some additional parameters:

$$E_{stretch}(S) = \int_\Omega \left(\frac{\partial S}{\partial u}\right)^2 + \left(\frac{\partial S}{\partial v}\right)^2 \, dudv, \tag{6}$$

$$E_{bend}(S) = \int_\Omega \left(\frac{\partial^2 S}{\partial u^2}\right)^2 + 2\left(\frac{\partial^2 S}{\partial u \partial v}\right)^2 + \left(\frac{\partial^2 S}{\partial v^2}\right)^2 \, dudv, \tag{7}$$

where Ω is the parametric domain of the surface S. A drawback of the approximations is that they are parameterization dependent. It is only a good approximation if the Jacobian determinant $\left|\frac{\partial S}{\partial u} \times \frac{\partial S}{\partial v}\right|$ does not differ too much from one. A compromise between those two is the data dependent approximations described in [Greiner et al., 96], which is computationally more expensive than the approximations above, but is in general a better approximation. The internal energy E_{int} we use is a linear combination of $E_{stretch}$ and E_{bend}.

4. Constraints

The usual way to model variational surfaces is by means of point and normal interpolation constraints, and sometimes curve interpolation constraints. They can be used to sketch a rough outline of the surface. We also use point-on-line and point-in-plane constraints, as well as default continuity constraints.

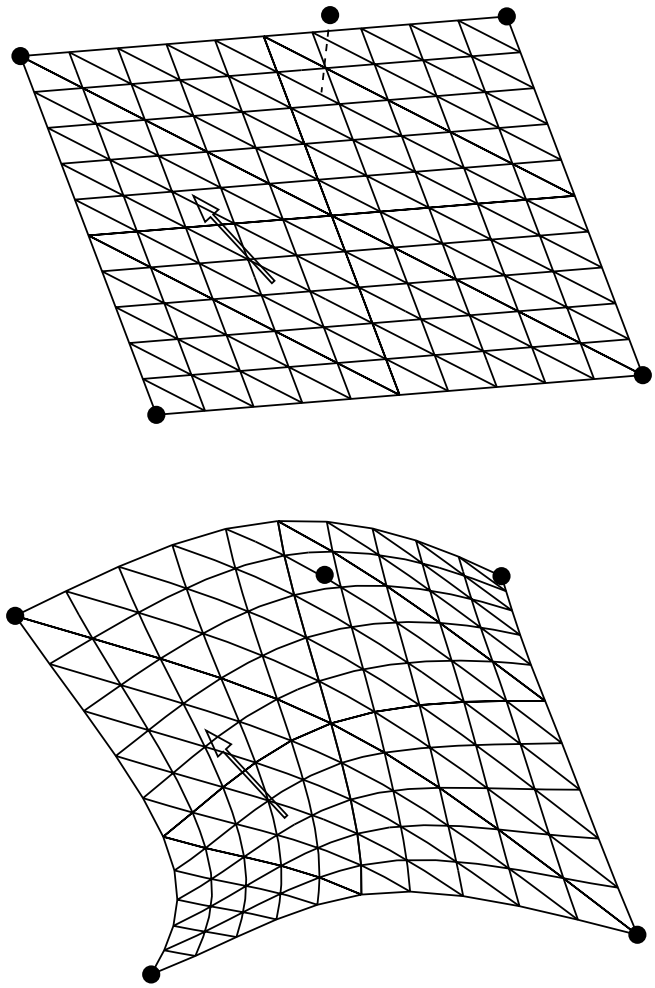

Figure 1: Top: a surface together with a point and a normal interpolation constraint. Bottom: resulting constrained variational surface.

4.1. Point and normal constraint

The point interpolation constraint forces some point of the surface to attain a prescribed position. In order to prevent the surface from collapsing to a single point in its strive to minimize its energy, at least a number of boundary points should be interpolated. The point interpolation constraint equation $T(u_0, v_0, w_0) = P$ in terms of the Bézier control points is

$$\sum_{i+j+k=n} P_{i,j,k} B^n_{i,j,k}(u_0, v_0, w_0) = P,$$

where $u_0 + v_0 + w_0 = 1$. To make this constraint linear in the control points, values have to be assigned to u_0, v_0 and w_0.

Similarly, the point-in-plane constraint $n \cdot T(u_0, v_0, w_0) = a$, forces the point $T(u_0, v_0, w_0)$ of the surface to lie in the plane $n \cdot x = a$, where n is the surface normal and '$\cdot$' denotes the

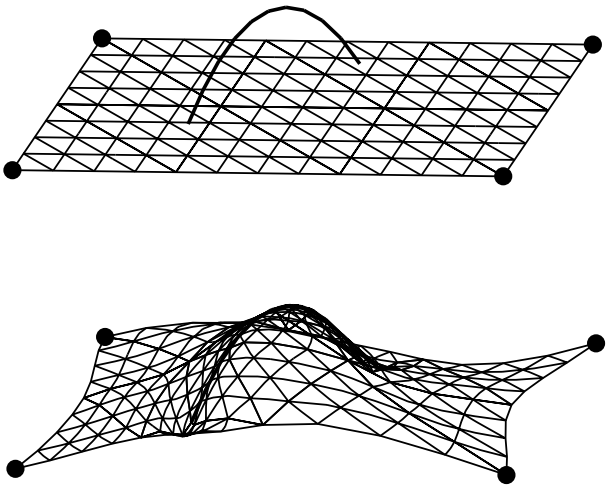

Figure 2: Top: a surface and a curve interpolation constraint. Bottom: resulting constrained variational surface.

inner product. A point-on-line constraint is the combination of two point-in-plane constraints.

A normal interpolation constraint forces the surface to attain a shape to have the prescribed normal at some point of the surface. Suppose that the surface normal N must be interpolated at the point $T(u_0, v_0, w_0)$. This means that the partial derivatives must be perpendicular to N:

$$N \cdot \frac{\partial T}{\partial u}(u_0, v_0, w_0) = 0, \qquad N \cdot \frac{\partial T}{\partial v}(u_0, v_0, w_0) = 0. \tag{8}$$

For prescribed N, these equations are also linear in the control points $P_{i,j,k}$.

Fig. 1 shows a surface together with point and normal interpolation constraints, and the resulting surface after constrained optimization.

4.2. Curve interpolation

A curve interpolation constraint forces the surface to pass through a given curve, as illustrated in Fig. 2. Let $C(t)$ be a single polynomial curve of degree k, and assume without loss of generality that this curve is in Bézier form (see Eq. 3). We need to determine which surface patches must interpolate the curve. (In our prototype implementation, this is currently done by the user.)

We let patch edges interpolate the curve. To do so, some patches may need to be subdivided. Then we optimize the surface with respect to the energy functional while still satisfying the constraints. Three cases can be distinguished: the two endpoints of $C(t)$ lie in one patch, in two adjacent patches, or in two non-adjacent patches. If the two endpoints of $C(t)$ lie in one patch, we split the patch into five triangles by successive subdivision as illustrated at the top left of Fig. 3. The parameter values at which to split the patch are currently determined by the user. The control points associated with the common boundary of the triangles T_1 and T_2 are now set equal to those of $C(t)$.

If the two endpoints $C(0)$ and $C(1)$ lie in two adjacent patches, we subdivide both patches at parameter values $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ into three triangles as illustrated at the top right of Fig. 3. The

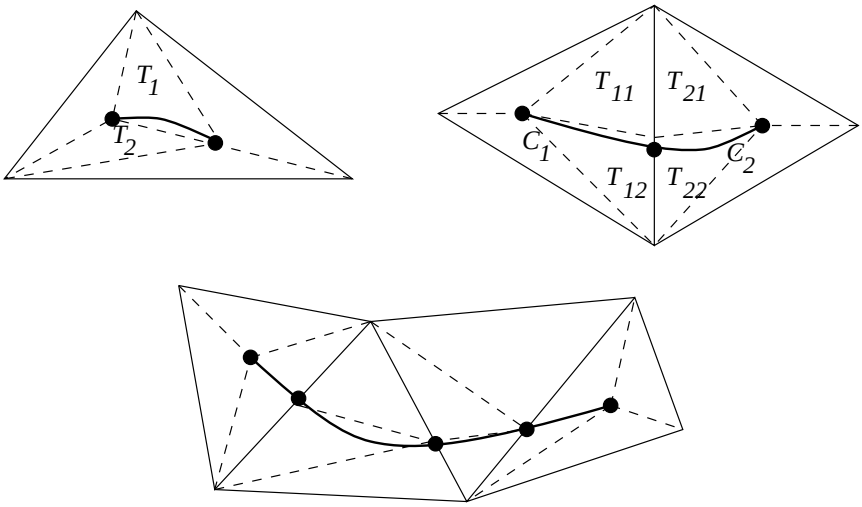

Figure 3: Interpolation curve endpoints in one patch, in two adjacent patches, and in two non-adjacent patches.

curve $C(t)$ is subdivided at $t = \frac{1}{2}$ into two curves C_1 and C_2, and the triangles T_1 and T_2 that are incident to the original edge are subdivided at the middle of that edge into the triangles T_{k1} and T_{k2}, as shown in the figure. The control points of the common boundary of T_{k1} and T_{k2} are now set to the control points of C_k, $k = 1, 2$.

If the endpoints of C lie in two non-adjacent patches and C must pass through a total of m patches, we subdivide $C(t)$ into m curves C_i, $i = 0, \ldots, m-1$, at $t = \frac{i}{m}$, $i = 1, \ldots, m-1$. The patches are subdivided as indicated at the bottom of Fig. 3. The control points of the appropriate triangle edges are now set to the control points of C_i.

In all three cases the patches must be of sufficiently high degree to set the control points of patch boundaries equal to those of the interpolation curve. Specifically, the degree of the patches may not be lower than the curve's degree. (In our current prototype the degree is three.) The situations that the endpoints fall on patch edges are special cases of the above ones. The parameter values where the patches are split do not really matter, because the positions of the newly introduced patches will eventually be determined by the subsequent energy minimization.

4.3. Continuity constraints

Continuity constraints are constraints to ensure continuity between adjacent patches. Two patches can be joined C^0-continuously by letting the control points along their common boundary coincide. The tangent plane of the surface varies continuously inside the patches, but not necessarily from patch to patch. Two patches can be joined G^1-continuously by imposing some nonlinear relations on their control points. Since nonlinear constraints make the minimization problem much more difficult, the following restricted form of tangent plane continuity is chosen. The patch corner points (e.g. M_0 in Fig 4) and the incident control points around it (R_0, M_1, L_0, U_0, ..., U_k) are constrained to lie in a plane. This yields equations like:

$$e_0(M_1 - M_0) + f_0(L_0 - M_0) + g_0(R_0 - M_0) \quad = \quad 0, \tag{9}$$

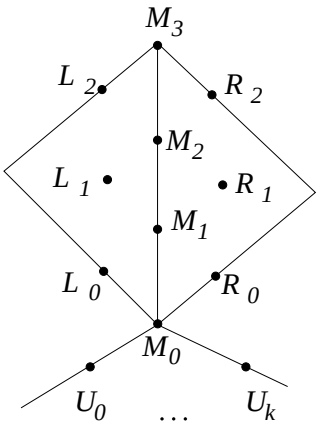

Figure 4: Control point involved in continuity constraints.

$$e_1(M_3 - M_2) + f_1(L_2 - M_2) + g_1(R_2 - M_2) \quad = \quad 0. \tag{10}$$

When these constraints are met, the surface is G^1-continuous at the vertices. G^1-continuity across the edges can be obtained, if additionally the following constraints are satisfied:

$$\begin{aligned}(e_0 + e_1)(M_1 - M_0) + 2e_0(M_2 - M_1) \quad &+ \quad (f_0 + f_1)(L_0 - M_0) + 2f_0(L_1 - M_1) + \\ &\quad (g_0 + g_1)(R_0 - M_0) + 2g_0(R_1 - M_1) = 0,\end{aligned} \tag{11}$$

$$\begin{aligned}2e_1(M_2 - M_1) + (e_0 + e_1)(M_3 - M_2) \quad &+ \quad 2f_1(L_1 - M_1) + (f_0 + f_1)(L_2 - M_2) + \\ &\quad 2g_1(R_1 - M_1) + (g_0 + g_1)(R_2 - M_2) = 0.\end{aligned} \tag{12}$$

These equations are linear in the control points if the coefficients are constant. The values of the coefficients are set as follows. If M_0 is in the interior of the surface, and there are three patches around it, then $e_0 = f_0 = g_0 = 1$. If M_0 is in the interior of the surface, and there are four patches around it, then $e_0 = 0$, and $f_0 = g_0 = 1$. If M_0 is in the interior of the surface, and there are k patches around it, then $e_0 = 1$, and $f_0 = g_0 = -1/(2\cos(2\pi/k))$. If M_0 is on the border of the surface, the surface should make an angle of ϕ at M_0, and there are k incident patches, then $e_0 = 1$, and $f_0 = g_0 = -1/(2\cos(\phi/k))$. The user can set the value of ϕ, the default value we assign is π. The values of e_1, f_1, and g_1 are set analogously.

These constraints take away many degrees of freedom. Alternatively, equations (11) and (12) can be replaced by

$$e_2(M_2 - M_1) + f_2(L_1 - M_1) + g_2(R_1 - M_1) = 0,$$

which is less restrictive but gives only approximate G^1-continuity across the edges.

The continuity constraints are default constraints, i.e. they apply without being explicitly specified by the user. However, they are overruled by interpolation constraints if there would be a conflict otherwise.

5. External energy operators

Now we introduce a new way of modeling variational surfaces using external energy operators, so called because they produce the external part of the total energy (1) of a surface. The

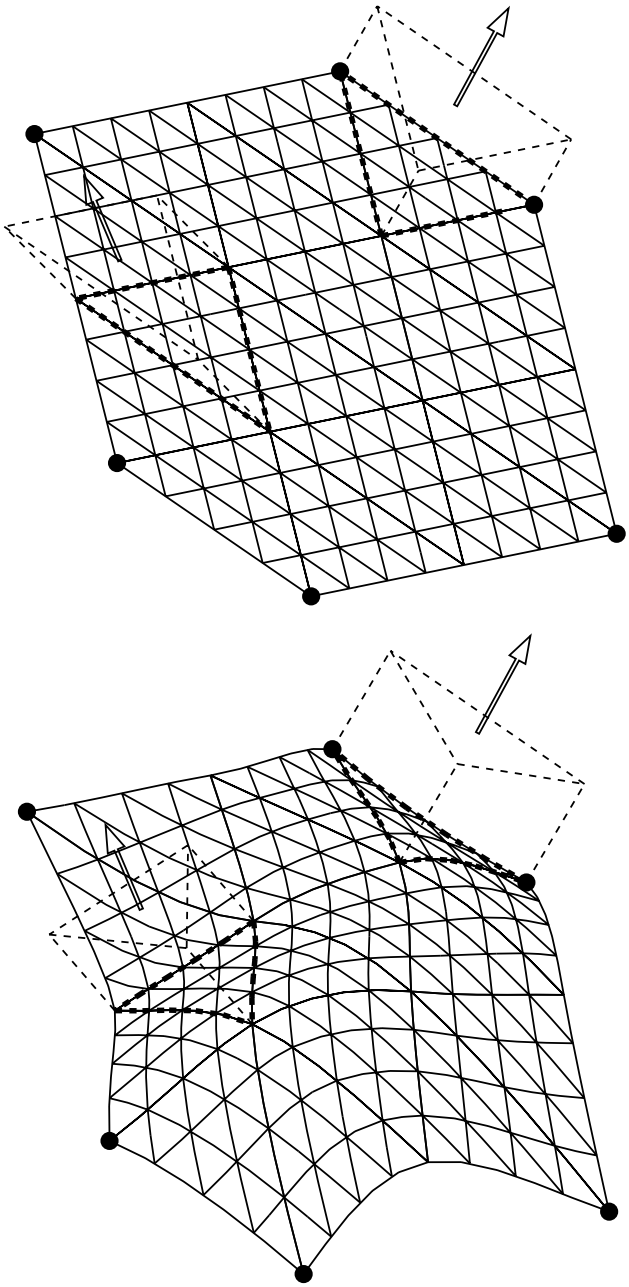

Figure 5: Top: a surface and two director operators. Bottom: resulting constrained variational surface.

goal of these operators is to offer flexible ways of deforming a surface locally. Each operator has a strength that can be set by the user. The energy terms are weighted accordingly when added to the total energy.

5.1. Director

The director is a design operator that tries to push the normal along a convex region of the surface into a specified direction. This is illustrated in Fig. 5, where the surface region is triangular, and the director is represented by a triangle. The correspondence between the region and triangle corner points is denoted by dashed lines. The normal of the director triangle is the direction imposed on the surface region to which the operator applies.

Let N be a unit vector normal to the director triangle. In any point on the surface S the quantities $(\frac{\partial S}{\partial u} \cdot N)^2$ and $(\frac{\partial S}{\partial v} \cdot N)^2$ are a measure for the difference between the directions of the surface normal and the vector N. So, the larger the inner products, the larger the deviation.

The external energy term for this operator has the following form:

$$E_{region-dir}(S) = \int_{\Gamma} \left(\frac{\partial S}{\partial u} \cdot N \right)^2 + \left(\frac{\partial S}{\partial v} \cdot N \right)^2 \, dudv, \tag{13}$$

where , is the parametric region of the surface to which the operator is applied. Minimizing this expression amounts to minimizing the total deviation between the prescribed normal and the normal along the surface region. The computation of the integral over , is described in Sect. 6.

For a single surface point $S(u_0, v_0)$, the energy term is simply

$$E_{point-dir}(S) = \left(\frac{\partial S(u_0, v_0)}{\partial u} \cdot N \right)^2 + \left(\frac{\partial S(u_0, v_0)}{\partial v} \cdot N \right)^2. \tag{14}$$

5.2. Point attractor

The point attractor is a design operator that pulls the surface towards a point, as illustrated in Fig. 6. The grey dot is the point that attracts the surface. The point on the surface to which the attractor applies is indicated by the dashed line.

Suppose we want the point on the surface S with parameters u_0, v_0 to be attracted by the point P. An external energy term that has this effect is:

$$E_{point-to-point}(S) = \|S(u_0, v_0) - P\|^2. \tag{15}$$

The minimal value of this expression is achieved when the surface point $S(u_0, v_0)$ coincides with P. However, other energy terms will often prevent this. For example, if the point attractor will bend the surface, then the bend energy will increase. The bend energy component in the total energy functional will then compete with $E_{point-to-point}$ to obtain a minimum, and will restrain the surface from interpolating P.

An external energy term that will attract a whole surface region to point P is:

$$E_{region-to-point}(S) = \int_{\Gamma} \|S(u, v) - P\|^2 \, dudv. \tag{16}$$

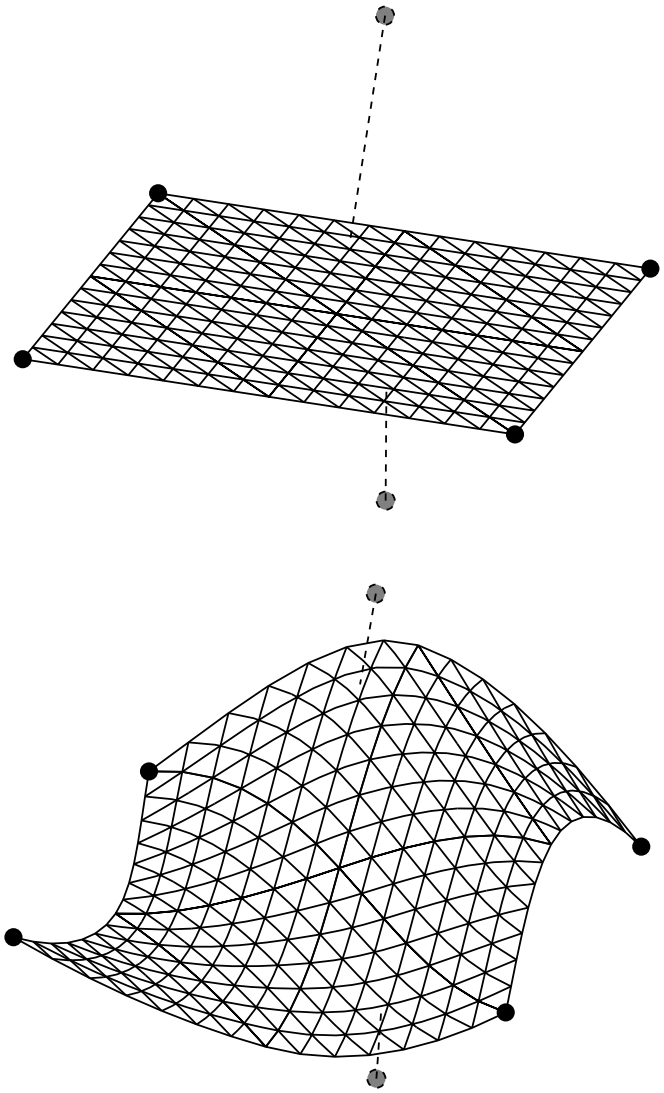

Figure 6: Top: a surface and two point attractors. Bottom: resulting constrained variational surface.

5.3. Surface attractor

The surface attractor is a design operator that pulls a part of the design surface (a single point or a whole region) towards another surface. This is illustrated in Fig. 7, where the surface region is triangular and the attractor surface is a triangle, drawn in grey. The correspondence between the region and triangle corners are designated by dashed lines.

A point-to-plane operator pulls a point of a surface towards a given plane $n \cdot x = a$:

$$E_{point-to-plane}(S) = \frac{\|n \cdot S(u_0, v_0) - a\|^2}{n \cdot n}. \tag{17}$$

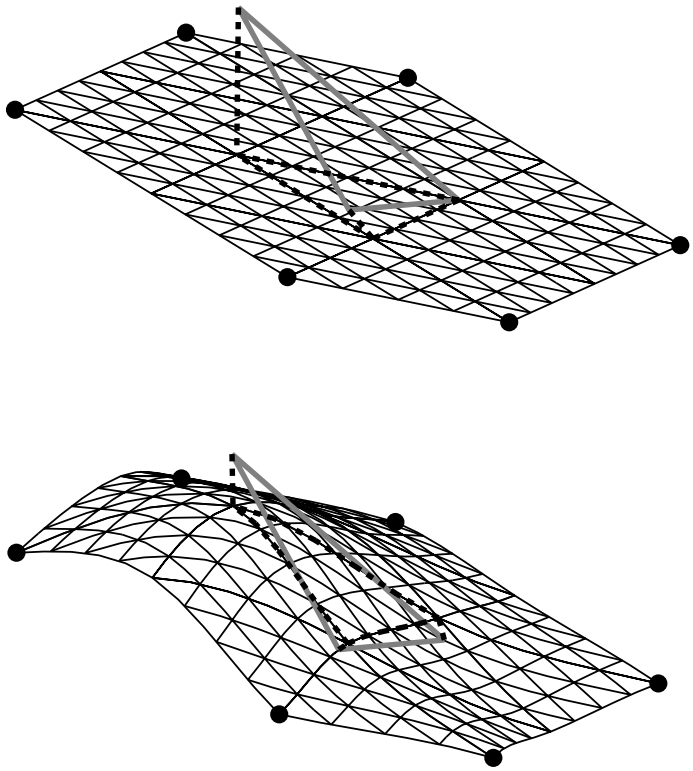

Figure 7: Top: original surface and a surface attractor. Bottom: resulting constrained variational surface.

where n is the plane normal. The corresponding energy term that attract a whole region to a plane is:

$$E_{region-to-plane}(S) = \int_{\Gamma} \frac{\|n \cdot S(u,v) - a\|^2}{n \cdot n} \, dudv. \tag{18}$$

An external energy term that will attract a region to an arbitrary surface T is:

$$E_{region-to-surface}(S) = \int_{\Gamma} \|S(u,v) - T(u,v)\|^2 \, dudv. \tag{19}$$

Like for the director and the point attractor, the surface attractor energy terms have to compete with other terms, which may restrain the design surface from exactly interpolating the attractor surface.

5.4. Curve attractors

For a line $\ell : p + \lambda q,\ p, q \in \mathbb{R}^3,\ \lambda \in \mathbb{R}$, the distance between a point x and ℓ is given by $\|Ax + b\|$, $A = I_3 - q\,q^T/(p \cdot p)$, $b = (p \cdot q)q/(q \cdot q) - p$. Energy terms that attract a point or

region to line ℓ are:

$$E_{point-to-line}(S) = \|A\,S(u_0, v_0) - b\|^2\,dudv \tag{20}$$

and

$$E_{region-to-line}(S) = \int_\Gamma \|A\,S(u, v) - b\|^2\,dudv. \tag{21}$$

Another type of curve attractor is one that pulls a patch edge toward a given curve. Let $E(u)$ be a patch edge, i.e. a Bézier curve as in Eq. (3), and $C(u)$ a given Bézier curve. An energy term that attracts E towards C is

$$E_{edge-to-curve}(S) = \int \|E(u) - C(u)\|^2\,du. \tag{22}$$

5.5. Repellors

In principle, the weight factor of the attractor operators could be made negative in order to simulate repelling behavior. However, if the weight factor is too negative, the total energy is dominated by the negative term, and the surface tries to minimize its energy by shooting to infinity. We have implemented the repellor operator as the reciprocal of the attractor energy term. In particular, the repellor operator that pushes a surface point $S(u_0, v_0)$ away from a given point P has energy term

$$E_{point-from-point}(S) = \frac{1}{\|S(u_0, v_0) - P\|^2}. \tag{23}$$

6. Computation

This section discusses how the surface with minimal energy can be computed. First, expressions are derived for the various energy functionals and constraints in terms of control points. Then we discuss how the problem can be solved as a minimization problem.

6.1. Energy and constraint expressions

Suppose that the surface S consists of m patches S_k: $S_0 \ldots S_{m-1}$. The stretch energy of S is simply the sum of the stretch energies of the patches:

$$E_{stretch}(S) = \sum_{k=0}^{m-1} E_{stretch}(S_k).$$

The stretch energy of a single patch S_k can be found by substitution of the Bézier triangle representation[1] $S_k(u, v, w) = \sum_{i=0}^{q-1} P_i B_i^n(u, v, w)$ into $E_{stretch}$ (Eq. (6)). This yields the following quadratic expression in the control points P_i:

$$E_{stretch}(S_k) = \sum_{i=0}^{q-1}\sum_{j=0}^{q-1} a_{ij} P_i^T P_j,$$

[1] We label control points and basis functions with numbers, instead of index triples as in Eq. (2).

where the coefficients a_{ij} are given by

$$a_{ij} = \int_\Gamma \frac{\partial B_i}{\partial u}\frac{\partial B_j}{\partial u} + \frac{\partial B_i}{\partial v}\frac{\partial B_j}{\partial v}\, dudv. \tag{24}$$

There is still a problem that has to be solved: the Bernstein basis functions B_i in Eq. (2) are defined in terms of barycentric coordinates (u, v, w), whereas for the integrals in Eq. (24) a parameterization in two coordinates is needed. This parameterization cannot be arbitrarily chosen. It should at least satisfy some symmetry properties, for instance that the matrix A_k does not depend on the orientation of the corner points. More formally: if $(k_0, k_1, k_2) = \sigma(i_0, i_1, i_2)$ and $(\ell_0, \ell_1, \ell_2) = \sigma(j_0, j_1, j_2)$ for some permutation σ, then the entries a_{ij} and $a_{k\ell}$ should be the same. The most trivial parameterization, namely the first two barycentric coordinates u and v, does not satisfy this property!

Therefore we have constructed a different parameterization with parameters x and y. Let the domain of the patch be an equilateral triangle $V_1V_2V_3$ in $\mathbb{R}^2$. The point on the patch with barycentric coordinates (u, v, w) has parameters x and y such that the point (x, y) has barycentric coordinates (u, v, w) with respect to the triangle $V_1V_2V_3$. In a formula:

$$(x, y) = uV_1 + vV_2 + wV_3. \tag{25}$$

This parameterization satisfies the above symmetry property. So the coefficients a_{ij} are of the form

$$a_{ij} = \int_\Gamma \frac{\partial B_i}{\partial x}\frac{\partial B_j}{\partial x} + \frac{\partial B_i}{\partial y}\frac{\partial B_j}{\partial y}\, dxdy. \tag{26}$$

Using the linear Eq. (25), the integral (26) can be transformed to the barycentric coordinates u and v. The result of this coordinate transformation is an expression of the following form:

$$a_{ij} = \int_\Gamma \left(\alpha_0 \frac{\partial B_i}{\partial u}\frac{\partial B_j}{\partial u} + \alpha_1 \frac{\partial B_i}{\partial u}\frac{\partial B_j}{\partial v} + \alpha_2 \frac{\partial B_i}{\partial v}\frac{\partial B_j}{\partial v}\right) J\, dudv, \tag{27}$$

where $\alpha_i, i = 0 \ldots 2$ and the Jacobian J are constants. For each patch, the region , is an arbitrary convex polygon, subset of $\{(u, v) \in \mathbb{R}^2 \,|\, u, v \geq 0, u + v \leq 1\}$. This region is split into trapezoids of the following form:

$$\{(u, v) \in \mathbb{R}^2 \,|\, a \leq u \leq b \wedge a_0 + a_1 u \leq v \leq b_0 + b_1 u\}. \tag{28}$$

Now, the integrand in (27) is a polynomial in u and v, which can be easily integrated over the ranges of u and v.

To get a more uniform approach these energies can be conveniently expressed in the concatenation vector $\mathcal{P}_k$ of the control points of the patch S_k:

$$\mathcal{P}_k = \left[\; P_0^T P_1^T \ldots P_{q-1}^T \;\right]^T.$$

Apart from the repellor terms, the internal and external energy contribution E to the patch S_k is always a quadratic expression in the vector $\mathcal{P}_k$:

$$E(S_k) = \mathcal{P}_k^T A_k \mathcal{P}_k + B_k^T \mathcal{P}_k + c_k.$$

The repellor is a special case: its energy functional is not quadratic, but the reciprocal of a quadratic function. Substitution of the Bezier representation into $E_{point-from-point}$ gives an expression of the form

$$\frac{1}{\mathcal{P}_k^T A_k \mathcal{P} + 2B_k^T \mathcal{P} + c_k}.$$

The coefficients a_{ij} of the different matrices A_k can again be written as (integrals over) polynomials in the barycentric coordinates u and v. The parameter transformation (25) applied to the internal energy is also applied here. The transformed integrands are linear combinations of B_i, uB_i, vB_i, B_iB_j, $\frac{\partial B_i}{\partial u}\frac{\partial B_j}{\partial u}$, $\frac{\partial B_i}{\partial u}\frac{\partial B_j}{\partial v}$, $\frac{\partial B_i}{\partial v}\frac{\partial B_j}{\partial v}$, $\frac{\partial^2 B_i}{\partial u^2}\frac{\partial^2 B_j}{\partial u^2}$, $\frac{\partial^2 B_i}{\partial u^2}\frac{\partial^2 B_j}{\partial u \partial v}$, etc. The integration domains are again split into trapezoids (28).

The total energy of the whole surface is the sum of the energies over the patches:

$$E(S) = \sum_k \mathcal{P}_k^T A_k \mathcal{P}_k + B_k^T \mathcal{P}_k + c_k + \sum_i \sum_k \frac{1}{\mathcal{P}_k^T C_{k,i} \mathcal{P}_k + 2D_{k,i}^T \mathcal{P}_k + c_{k,i}}, \tag{29}$$

where k ranges over the number of patches, and i ranges over the number of repellors. Since patches that are adjacent in S have control points in common, the energy is expressed in the concatenation vector $\mathcal{P}$ of all distinct control points of the whole surface S. It is an expression in $\mathcal{P}$ of the following form:

$$E(S) = \mathcal{P}^T A \mathcal{P} + B^T \mathcal{P} + c + \sum_i \frac{1}{\mathcal{P}^T C_i \mathcal{P} + 2D_i^T \mathcal{P} + c_i}. \tag{30}$$

In Sect. 4 we have seen that point and continuity constraints consist of three linear equations in the components of the control points of a patch (one for the x, y and z coordinate) and normal constraints consist of two linear equations in these components. This means that these constraints are also linear in the vector of all control points $\mathcal{P}$. Therefore, the point and normal constraints can be gathered in one matrix equation

$$F\mathcal{P} = G, \tag{31}$$

with F a suitably chosen matrix. Since each constraint equation depends on a small number of control points, the matrix F is sparse. Finally, the curve constraint is constructed in such a way that a number of control points are fixed. These constraints can be incorporated by eliminating these fixed control points from the minimization problem.

We have now arrived at a minimization problem with object function (30) and linear constraints (31):

$$\text{Minimize} \quad E_{tot}(\mathcal{P}), \tag{32}$$

$$\text{subject to} \quad F\mathcal{P} = G. \tag{33}$$

We solve this system in two steps. First the constraint equations (33) are used to eliminate some variables from the minimization problem (32). This results in an unconstrained minimization problem with an object function of the same form as (30). This unconstrained minimization problem is simply solved using the standard conjugate gradient method (see e.g. [Luenberger, 89]).

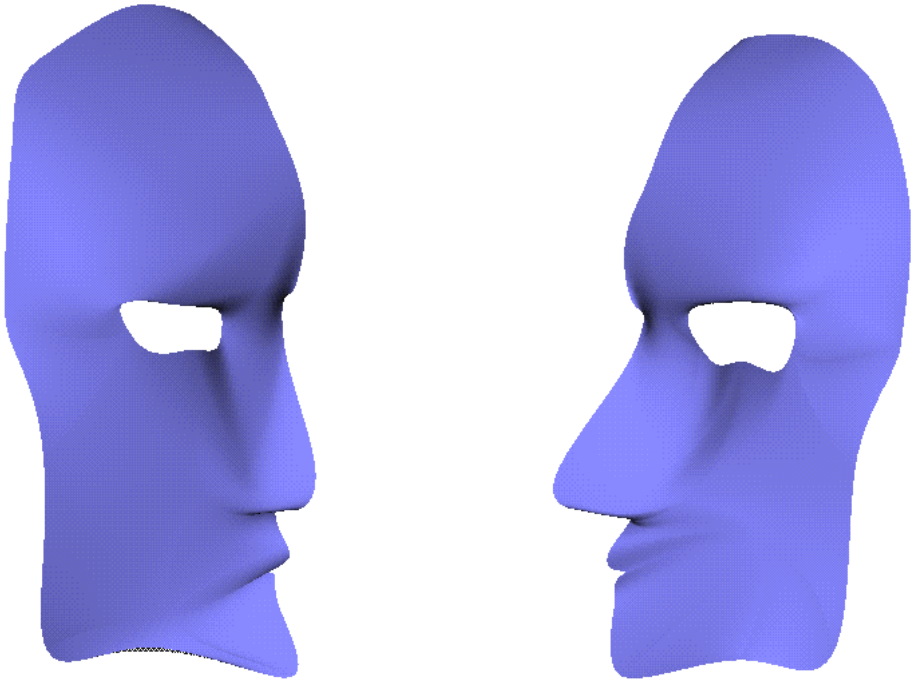

Figure 8: Surfaces modeled with external energy operators.

7. Conclusions

We have introduced a way to interactively model variational surfaces, by means of operators that affect the total energy of the surface. These operators effectively control groups of control points. In this way we can deform the surface locally, while the energy minimizing principles remain satisfied. This gives additional flexibility over the standard way of variational modeling (with constraints only). The computation involves standard minimization of a quadratic object function under linear constraints. We have tested our ideas in a prototype system for modeling surfaces of Bézier triangles. Experiments show that these operators offer an effective way of modeling. Figure 8 shows two surfaces modeled by changing only a few external energy operators.

The continuity constraints of Sect. 4.3 are inconvenient for two reasons. Firstly, since they must be linear in the control points they are more restrictive than the general tangent plane continuity constraints we actually want, which costs many degrees of freedom. Secondly they generate many constraint equations that must be eliminated. Triangular simplex splines, for example DMS-splines [Dahmen et al., 92], are inherently smooth, which eliminates the need for continuity constraints. We have also implemented the external operators for DMS-splines, but the energy terms are hard to evaluate exactly, and the integrands may differ from patch to patch [Wesselink, 96]. Also, fixing the boundary of the surface is more involved.

References

[Böhm and Farin, 83] W. Böhm and G. Farin. Letter to the editor. *Computer Aided Design*, 15(5), 1983, 260–261.

[Celniker and Gossard, 91] G. Celniker and D. Gossard. Deformable curve and surface finite-elements for free-form shape design. *Computer Graphics*, 25(4), 1991, 257–266.

[Dahmen et al., 92] W. Dahmen, C. A. Michelli, and H.-P. Seidel. Blossoming begets B-spline beses built better by B-patches. *Mathematics of Computation*, 59(199), 1992, 97–115.

[do Carmo, 76] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, isbn 0-13-212589-7, 1976.

[Farin, 93] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, 3rd edition, 1993.

[Goldman, 83] R. N. Goldman. Subdivision algorithms for Bézier triangles. *Computer Aided Design*, 15(3), 1983, 159–166. See also [Böhm and Farin, 83].

[Greiner et al., 96] G. Greiner, J. Loos, and W. Wesselink. Surface modeling with data dependent energy functionals. *Computer Graphics Forum, Proceedings EG'96*, , 1996, .

[Hagen and Schulze, 91] H. Hagen and G. Schulze. Variational principles in curve and surface design. In H. Hagen and D. Roller (editors), *Geometric Modelling–Methods and Applications*, Springer-Verlag, 1991.

[Kallay, 93] M. Kallay. Constrained optimization in surface design. In *Preliminary proceedings IFIP TC5/WG5.10 Working Conference on Modeling in Computer Graphics, Methods and Applications*, 1993.

[Luenberger, 89] D. G. Luenberger. *Linear and Non-Linear Programming*. Addison-Wesley, 2nd edition, 1989.

[Moreton and Séquin, 92] H. Moreton and C. H. Séquin. Functional optimization for fair surface design. *Computer Graphics*, 26(2), 1992, .

[Rando and Roulier, 91] T. Rando and J. A. Roulier. Designing faired parametric surfaces. *Computer Aided Design*, 23(7), 1991, 492–497.

[Terzopoulos et al., 87] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *ACM Computer Graphics*, 21(4), 1987, 205–214.

[Terzopoulos and Qin, 94] D. Terzopoulos and H. Qin. Dynamic NURBS with geometric constraints and for interactive sculpting. *Transactions on Graphics*, 13(2), 1994, 103–136.

[Veltkamp and Wesselink, 95] R. C. Veltkamp and W. Wesselink. Modeling 3d curves of minimal energy. *Computer Graphics Forum*, 14(3), 1995, 97–110. Proceedings Eurographics'95.

[Welch and Witkin, 92] W. Welch and A. Witkin. Variational surface modeling. *ACM Computer Graphics*, 26(2), 1992, 157–166.

[Wesselink, 96] J. W. Wesselink. *Variational Modeling of Curves and Surfaces*. PhD thesis, Technical University of Eindhoven, Department of Computing Science, Eindhoven, The Netherlands, 1996.

[Witkin, 94] A. Witkin (editor). *An Introduction to Physically Based Modeling*. SIGGRAPH Course#32, 1994.