7o4

Volume xxx, (1995) number yyy pp. 000–000

Interactive Sculpting with Deformable Nonuniform B-splines

Tzvetomir Ivanov Vassiliev

Department of Computer Science, Queen Mary and Westfield College, Mile End road, London E1 4NS, UK

Abstract

This paper describes an efficient method for manipulating deformable B-spline surfaces, based on minimizing an energy functional. The major benefit of the proposed new fairness norm is that it preserves the natural representation of the B-spline surface control points (a two dimensional array) which has an efficiency advantage over other methods. The designer uses forces as a main sculpting tool and is free to specify a single force, a set of isolated forces, forces situated on a line or curve or area of the deformable surface. The user is allowed to modify several parameters and in this way to change the physical properties of the object.

Keywords: Nonuniform B-splines, Curve and Surface Energy Minimization, Interactive Sculpting.

1. Introduction

A necessary feature of a modern CAD system is the facility for interactive free-form design, i.e. to supply the user with tools and techniques for interactive manipulation of the shape of an object. The simplest approach is to allow the user to manipulate the control points of the designed object (curve, surface or volume) and in this way to change its shape. Although this is the easiest and fastest way from a mathematical point of view, it does not prove to be very appealing for designers. The main reason is that when the portion of the object that is to be deformed includes many control points, the task of the user becomes tedious and in some cases even impossible. For a more detailed description of other drawbacks see Hsu et al.[1]

The main objective of the work presented in this paper is to develop a method for interactive sculpting of a B-spline surface (or curve), using direct manipulation on the object, rather than its controls. The technique has to preserve the two dimensionality of the array of the control points and to allow a real time sculpting even when a large portion of the object is deformed.

One of the approaches for direct manipulation, described in the literature[1, 2, 3, 4, 5], is to use mathematical means to compute the new position of the control points. Hsu, Hughes and Kaufman[1] give the user the freedom to move a single point or a set of points on the object to a new position. They use a least squares method to solve the under-determined system for the new control points. The method is local because for cubic B-spline curves when a single point is moved only the closest four control points are affected. This drawback was removed by Finkelstein and Salesin[2] applying multiresolution analysis. The method of Fowler and Bartels[3, 4] concerns not only points but also tangents and the twist vector. A length-minimum solution is applied to the under-determined system. The scheme proposed by Rappoport et al.[5] uses Kalman filtering to compute the new control points. Its main advantage is that the constraints are not exact but probabilistic.

A completely different approach to these methods is so called "physical modelling". Here the designed object is enriched with a behaviour governed by physical laws. Due to this fact physics-based techniques possess a much higher degree of flexibility and freedom than the pure mathematical solutions. A lot of research has been done recently in this field. Bloor and Wilson[6] obtained a B-spline surface as an approximate numerical solution to partial differential equations. Thingvold and Cohen[7] suggested a deformable

B-spline surface, whose control points were represented as mass points connected with elastic springs and hinges. Celniker and Gossard[8] suggested an interesting energy functional for free-form shape design, based on minimizing the finite-element stretching and bending energy terms. They used a Hermite polynomial basis for representing curves and triangular patches for surfaces. Later Celniker and Welch[9] extended the idea for point and curve constraints on a B-spline surface. The method of minimization of curvature[10] or of the thin plate and the membrane energy[11, 12, 13, 14] has also been widely used.

A common drawback of almost all existing methods for minimizing an energy functional in the context of B-spline surfaces[8, 9, 11, 12, 13] is that the natural representation of the control points (two dimensional array) is not preserved. They are rearranged in a column vector which, as proved by Farin[15], is more time consuming. In general, if we have to compute a subarray of $(n \times n)$ control points and if they are rearranged in a vector, the number of computations is $O(n^6)$. It can be reduced to $O(n^4)$ if we keep the points organized in a two dimensional array[15].

This paper describes a method for interactive sculpting of a nonuniform B-spline surface using a physics-based modelling approach. The energy functional is similar to those described in the literature[8, 9, 12, 13] but it preserves the two dimensional array representation of the control points. Due to this fact and other details in the norm, the number of the computations is reduced to $O(n^3)$. Since the application of forces rather that new positions is the main sculpting tool, the technique possesses the softness typical of the method of Rappoport et al.[5]

The rest of the paper is organized as follows. In section 2 we describe a deformable B-spline curve, including evaluation of the fairness norm and the forcing vector, solving for a single force and set of forces and moving a single point of the curve to a new position. Section 3 extends the scheme for B-spline surfaces. In section 4 an approach for solving the integrals for the nonuniform case is described. Section 5 presents the results and section 6 concludes the paper.

2. Deformable B-spline Curve

Let $\mathbf{w}(u) = [x(u), y(u), z(u)]$ be a space curve parameterized by u and let $\mathbf{f}(u)$ denote the applied sculpting forces on it. Celniker and Gossard[8] suggested the following energy functional for curves

$$E = \int_{curve} [\alpha \dot{\mathbf{w}}(u) + \beta \ddot{\mathbf{w}}(u) - 2\mathbf{f}(u)\mathbf{w}(u)]\, du, \quad (1)$$

where $\dot{\mathbf{w}}(u)$ and $\ddot{\mathbf{w}}(u)$ are the first and second derivatives in respect to the parameter u. The energy E in (1) can be subdivided in two parts $E = E_1 \text{-} E_2$:

$$E_1 = \int_{curve} [\alpha \dot{\mathbf{w}}(u) + \beta \ddot{\mathbf{w}}(u)]\, du \quad (2)$$

$$E_2 = 2 \int_{curve} \mathbf{f}(u)\mathbf{w}(u)\, du. \quad (3)$$

E_1 represents the energy of the curve itself and its natural resistance to deformations while E_2 is the energy due to the applied forces. The curve energy is expressed as a weighted sum of its stretching and bending terms, where α and β are freely selected coefficients. First we shall try to evaluate (2) and (3) for a single B-spline segment and then by application of the finite element method the global energy can be found through summation.

2.1. Evaluating the fairness norm

A single cubic B-spline segment can be written as $\mathbf{w}_s(u) = \mathbf{V}_s^T \mathbf{B}_s$, where $\mathbf{V}_s$ is the column vector of control points and $\mathbf{B}_s$ is a column vector of the B-spline basis functions. Evaluating the integral (2) for the function $\mathbf{w}_s$ will result in the following quadratic form

$$E_{1s} = \mathbf{V}_s^T \mathbf{K}_s \mathbf{V}_s \quad (4)$$

where $\mathbf{K}_s$ is a (4×4) symmetric matrix, whose entries are calculated by solving the integral (2). E_{1s} is called a "local fairness norm" and $\mathbf{K}_s$ is the "local stiffness matrix" for the segment $\mathbf{w}_s(u)$. $\mathbf{K}_s$ can be represented as a weighted sum of its stretching and bending terms, i.e.

$$\mathbf{K}_s = \alpha \mathbf{K}_{1s} + \beta \mathbf{K}_{2s}. \quad (5)$$

In case of a uniform knot distribution and by using de Boor's formula[16] one can easily find the entries of the matrices in (5) directly solving

$$\mathbf{K}_{1s} = \int_{u_0}^{u_1} \dot{\mathbf{B}}_s \dot{\mathbf{B}}_s^T du, \quad \mathbf{K}_{2s} = \int_{u_0}^{u_1} \ddot{\mathbf{B}}_s \ddot{\mathbf{B}}_s^T du. \quad (6)$$

The exact values of $\mathbf{K}_{1s}$ and $\mathbf{K}_{2s}$ are given by Vassilev[17].

Now, after we have expressed the local fairness norm E_{1s} for a single segment, the global fairness norm E_1 for a B-spline curve can be found as a sum of the local fairness norms over each of the B-spline polynomial segments. The expression for E_1 can be written as

$$E_1 = \mathbf{V}^T \mathbf{K} \mathbf{V} \quad (\mathbf{K} = \alpha \mathbf{K}_1 + \beta \mathbf{K}_2), \quad (7)$$

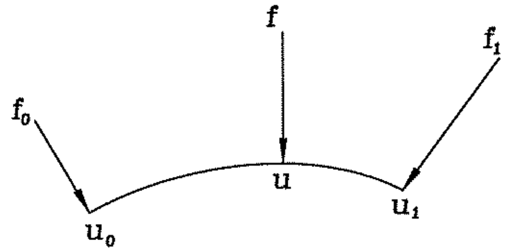


Figure 1: *A force acting on a single B-spline segment is a linear interpolation of forces applied at the ends.*

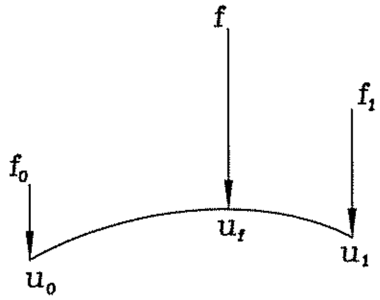


Figure 2: *A force acting on a B-spline segment is replaced by two other applied at the ends.*

where $\mathbf{V}$ is the vector of control points for the spline curve and $\mathbf{K}$ is a quadratic symmetric band matrix obtained from the local stiffness matrices.

2.2. Evaluating the forcing vector

In order to calculate (3) first an analytical expression for the force vector $\mathbf{f}(u)$ should be available. To simplify the model we assume that $\mathbf{f}$ is a linear interpolation of the forces applied at the ends of the polynomial segment (see Figure 1)

$$\mathbf{f}(u) = \frac{u_1 - u}{u_1 - u_0}\mathbf{f}_0 + \frac{u - u_0}{u_1 - u_0}\mathbf{f}_1 = \mathbf{S}^T\mathbf{F}_s, \qquad (8)$$

where $\mathbf{S}$ is a vector of interpolating scaling functions and $\mathbf{F}_s^T = [\mathbf{f}_0\ \mathbf{f}_1]$.

After computing the integral (3) using the formula (8) one can derive the following result for the local segment forcing energy

$$E_{2s} = 2\mathbf{V}_s^T\mathbf{C}_s\mathbf{F}_s, \qquad (9)$$

where $\mathbf{V}_s$ is the local vector of control points and $\mathbf{C}_s$ is a (4×2) coefficient matrix whose entries are calculated from

$$\mathbf{C}_s = \int_{u_0}^{u_1} \mathbf{B}_s\mathbf{S}^T du. \qquad (10)$$

Solving (10) for the uniform case gives

$$\mathbf{C}_s = \frac{1}{120}\begin{bmatrix} 4 & 1 \\ 33 & 22 \\ 22 & 33 \\ 1 & 4 \end{bmatrix}. \qquad (11)$$

The product $\mathbf{C}_s\mathbf{F}_s$ is also called a "local forcing vector".

After we have an expression for the local forcing vector, applying the finite element principal, the global forcing energy becomes

$$E_2 = 2\mathbf{V}^T\mathbf{CF}, \qquad (12)$$

where $\mathbf{V}$ is the vector of control points with length n, $\mathbf{F}$ is the global forces vector with length (n-2) and $\mathbf{C}$ is a $[n\times(n\text{-}2)]$ matrix derived through summation from the local coefficient matrices.

Now we are ready to rewrite once again (1) but this time in matrix form:

$$E = \mathbf{V}^T\mathbf{KV} - 2\mathbf{CF}. \qquad (13)$$

Minimizing (13) will ensure the smoothness and the fairness of the resulting curve. The minimum can be found simply by setting the gradient equal to zero

$$\mathbf{KV} = \mathbf{CF}. \qquad (14)$$

2.3. Solving for a single force

Suppose we have a static state of the curve with control vector $\mathbf{V}_0$ and forces $\mathbf{F}_0$, i.e. $\mathbf{KV}_0 = \mathbf{CF}_0$. The physical meaning of the vector $\mathbf{F}_0$ is the set of forces that makes the curve occupy a particular shape in the space, i.e. it represents the geometric constraints. If no forces are applied ($\mathbf{F}_0 \equiv 0$) then the curve will contract to a single point, the origin of the coordinate system (0, 0, 0). Let us now change the force vector to a new value $\mathbf{F}_1$. Then the control points will also change to $\mathbf{V}_1$, i.e. $\mathbf{KV}_1 = \mathbf{CF}_1$. Subtracting the equalities corresponding to the two states results in the following system:

$$\mathbf{K}\Delta\mathbf{V} = \mathbf{C}\Delta\mathbf{F}, \qquad (15)$$

where $\Delta\mathbf{F}$ represents the change in the applied forces and $\Delta\mathbf{V}$ is the change in the control points due to these forces. The number r of the control points that are to be changed, and in this way the breadth of the change, can be controlled by the user. Let us now assume that a single force $\mathbf{f}$ is acting on the curve point $\mathbf{w}_f = \mathbf{w}(u_f)$. Then $\mathbf{f}$ can be divided into two forces ($\mathbf{f} = \mathbf{f}_0 + \mathbf{f}_1$) applied at the ends of the segment (see Figure 2), where

$$\mathbf{f}_0 = \frac{u_1 - u_f}{u_1 - u_0}\mathbf{f} \quad \mathbf{f}_1 = \frac{u_f - u_0}{u_1 - u_0}\mathbf{f}. \qquad (16)$$

Now the right hand side of (15) changes to

$$\mathbf{C}\Delta\mathbf{F} = \mathbf{C}\mathbf{f}_0 + \mathbf{C}\mathbf{f}_1 = \mathbf{C}_0\mathbf{f} + \mathbf{C}_1\mathbf{f} = \mathbf{C}_f\mathbf{f}$$

and the system is expressed as

$$\mathbf{K}_r\Delta\mathbf{V}_r = \mathbf{C}_f\mathbf{f}, \qquad (17)$$

where $\mathbf{K}_r$ is the system matrix with dimension $(r\times r)$, retrieved from the global stiffness matrix, $\Delta\mathbf{V}_r$ is the unknown vector of the displacement in the control points and $\mathbf{C}_f$ is a coefficient vector corresponding to the force $\mathbf{f}$. The systems (17) does not need to be solved for each spatial component x, y, z. It is sufficient to find the solution $\mathbf{V}_c$ of the system $\mathbf{K}_r\mathbf{V}=\mathbf{C}_f$ and then

$$\Delta\mathbf{V}_r = \mathbf{V}_c\mathbf{f}. \qquad (18)$$

Once the displacement vector $\Delta\mathbf{V}$ has been found the new vector of control points is $\mathbf{V}_1=\mathbf{V}_0+\Delta\mathbf{V}$.

2.4. Moving a curve point to a new position

Applying a force on a curve point will move the point in a desired direction but the exact new position cannot be predicted. However the user might want to move the point to a specified new target. Then we have to compute the magnitude of a force that will do the desired change.

Suppose we want to move a curve point $\mathbf{p}_0 = \mathbf{w}(u_f)$ to a new position $\mathbf{p}_1$. Then from the B-spline curve formula one can write

$$\mathbf{B}^T\Delta\mathbf{V} = \Delta\mathbf{p}, \qquad (19)$$

where $\Delta\mathbf{p} = \mathbf{p}_1 - \mathbf{p}_0$. Using (18), (19) changes to $\mathbf{B}^T\mathbf{V}_c\mathbf{f} = \Delta\mathbf{p}$ and the expression for computing the necessary force becomes

$$\mathbf{f} = \frac{\Delta\mathbf{p}}{\mathbf{B}^T\mathbf{V}_c}. \qquad (20)$$

Note that the value in the denominator is not a vector but a scalar product of two vectors.

2.5. Solving for a set of forces

If the user specifies not only one but a set of forces $\mathbf{f}_i$, i=1,...,s, two different approaches are possible. The first is to compute each displacement vector $\Delta\mathbf{V}_i$, due to the force $\mathbf{f}_i$, and then the global displacement $\Delta\mathbf{V}$ is, due to the linearity, the sum of the separate $\Delta\mathbf{V}_i$. However, if the number of forces is large (s>4) it is more efficient to keep the equations of the type (15) and solve three systems for the different spatial components.

3. Deformable B-spline Surface

Let $\mathbf{w}(u,v) = [x(u,v),\ y(u,v),\ z(u,v)]$ be a surface parameterized by u and v, and let $\mathbf{f}(u,v)$ denote the applied sculpting forces on it. The following energy functional for surfaces is considered in this paper

$$E = \iint\limits_{surface} [c_1\mathbf{w}_u^2 + c_2\mathbf{w}_v^2 + c_3\mathbf{w}_{uu}^2 + c_4\mathbf{w}_{uv}^2 + c_5\mathbf{w}_{vv}^2 +$$

$$c_6\mathbf{w}_{u^2v}^2 + c_7\mathbf{w}_{uv^2}^2 + c_8\mathbf{w}_{u^2v^2}^2 - 2\mathbf{f}\mathbf{w}]\,du\,dv, \qquad (21)$$

where the suffixes mean partial derivatives in respect to the parameters u and v.

Analogously to the curve case the surface energy can be split in two parts $E = E_1$-E_2, E_1 representing the natural surface resistance to deformations and E_2 the forcing energy.

3.1. Evaluating the fairness norm

As mentioned in the introduction our main goal when calculating the fairness norm will be to preserve the two dimensionality of the array of control points. The equation of a tensor product B-spline surface is

$$\mathbf{w}(u,v) = \mathbf{B}_u^T\mathbf{V}\mathbf{B}_v, \qquad (22)$$

where $\mathbf{V}$ is a $(m\times n)$ matrix of the controls points and $\mathbf{B}_u$ and $\mathbf{B}_v$ are vectors containing the B-spline basis functions for the parameters u and v respectively. Now solving the integral (21) for E_1 and then finding the gradient in respect to $\mathbf{V}$ leads to

$$\nabla E_1 = 2(c_1\mathbf{K}_{1u}\mathbf{V}\mathbf{K}_{0v} + c_2\mathbf{K}_{0u}\mathbf{V}\mathbf{K}_{1v} + c_3\mathbf{K}_{2u}\mathbf{V}\mathbf{K}_{0v} +$$

$$c_4\mathbf{K}_{1u}\mathbf{V}\mathbf{K}_{1v} + c_5\mathbf{K}_{0u}\mathbf{V}\mathbf{K}_{2v} + c_6\mathbf{K}_{2u}\mathbf{V}\mathbf{K}_{1v} +$$

$$c_7\mathbf{K}_{1u}\mathbf{V}\mathbf{K}_{2v} + c_8\mathbf{K}_{2u}\mathbf{V}\mathbf{K}_{2v}). \qquad (23)$$

Here the matrices $\mathbf{K}_{1u}$, $\mathbf{K}_{1v}$, $\mathbf{K}_{2u}$, $\mathbf{K}_{2v}$ are the same as for a B-spline curve, see (7). The suffix shows the knot sequence (parameter) for which they are calculated. The matrices $\mathbf{K}_{0u}$ and $\mathbf{K}_{0v}$ are new and appear only for a B-spline surface. Their entries are computed solving

$$\mathbf{K}_{0u} = \int_{u_0}^{u_1} \mathbf{B}_u\mathbf{B}_u^T\,du, \quad \mathbf{K}_{0v} = \int_{v_0}^{v_1} \mathbf{B}_v\mathbf{B}_v^T\,dv. \qquad (24)$$

It is quite obvious that the expression (23) is too complicated to be useful for any practical application. The following approximation to (23) is suggested:

$$\nabla E_1 = 2\mathbf{K}_u\mathbf{V}\mathbf{K}_v, \qquad (25)$$

where $\mathbf{K}_u = \gamma_u\mathbf{K}_{0u} + \alpha_u\mathbf{K}_{1u} + \beta_u\mathbf{K}_{2u}$
and $\mathbf{K}_v = \gamma_v\mathbf{K}_{0v} + \alpha_v\mathbf{K}_{1v} + \beta_v\mathbf{K}_{2v}$.

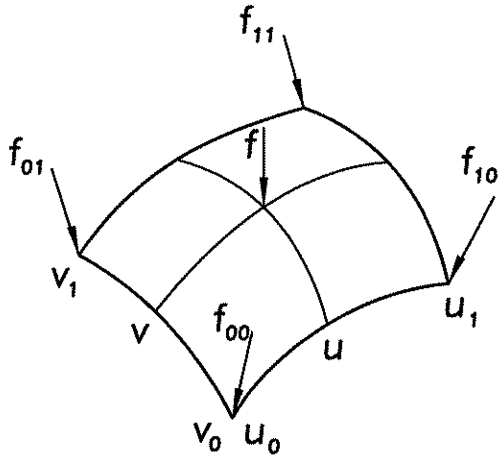


Figure 3: *A force acting on a single B-spline patch is a linear interpolation of forces applied at the ends.*

The fairness norm (25) has the following interesting properties:

- It is a good approximation of (23) (if $\gamma \ll \alpha, \beta$) and contains all its terms.
- While the highest order of the derivatives in the literature[8, 9, 12, 13] is only two, in (25) it is four ($\beta_u\beta_v\mathbf{K}_{2u}\mathbf{V}\mathbf{K}_{2v}$). This supposes a smoother and fairer resulting surface.
- Most of the described functionals[8, 9, 12, 13] have two parameters to control the physical properties of the surface: α represents resistance to stretching and β to bending. In (25) another control parameter appears, γ. As the various experiments the author conducted showed, it has the meaning of hardness of the material. The bigger the value of γ the harder the surface and consequently the narrower the deformation.
- The new proposed fairness norm has two stiffness matrices, one in direction of the parameter u and another one for v. This gives users the opportunity to design nonisotropic materials, i.e. with different properties in the different directions.
- The expression is quite simple and hence suitable for practical applications. As we shall see later it leads to a simple system which can be solved very fast.

3.2. Evaluating the forcing matrix.

When calculating the forcing matrix we shall follow the same approach as we did for curves: first evaluating for a single B-spline patch and then uniting all the elements in a global expression. As shown in Figure 3 a force applied anywhere on the patch can be represented as a bilinear interpolation of the forces applied at the four ends

$$\mathbf{f}(u,v) = \mathbf{S}_u^T\mathbf{F}_p\mathbf{S}_v, \tag{26}$$

where $\mathbf{S}_u$ and $\mathbf{S}_v$ are vectors with the interpolation scaling functions in direction u and v respectively and $\mathbf{F}_p$ is a (2×2) matrix of the forces. Solving the forcing part of the integral (21), substituting $\mathbf{f}$ with (26) and finding the gradient in respect to the control points gives the following matrix expression

$$\nabla E_{2p} = 2\mathbf{C}_{su}\mathbf{F}_p\mathbf{C}_{sv}^T. \tag{27}$$

Here $\mathbf{C}_{su}$ and $\mathbf{C}_{sv}$ are the same as for a curve segment, see (9) and (10). Applying the finite element principle and summating the energies of the local patches one can obtain

$$\nabla E_2 = 2\mathbf{C}_u\mathbf{F}\mathbf{C}_v^T. \tag{28}$$

where $\mathbf{F}$ is the $[(m-2)\times(n-2)]$ forces matrix, $\mathbf{C}_u[m\times(m-2)]$ and $\mathbf{C}_v[n\times(n-2)]$ are the global coefficient matrices. Using (25) and (28), we are ready to write a system of linear equations that minimizes the energy functional (21) and preserves the two dimensionality of the B-spline surface control points

$$\mathbf{K}_u\mathbf{V}\mathbf{K}_v = \mathbf{C}_u\mathbf{F}\mathbf{C}_v^T. \tag{29}$$

3.3. Solving for a single force

Exploiting the same ideas as for curves one can easily arrive at the following system describing the state of the surface when a single force is applied

$$\mathbf{K}_{ur}\Delta\mathbf{V}_r\mathbf{K}_{vr} = \mathbf{C}_{uf}\mathbf{C}_{vf}^T\mathbf{f}, \tag{30}$$

where $\Delta\mathbf{V}_r$ is an unknown $(r_u\times r_v)$ displacement matrix, $\mathbf{C}_{uf}$ and $\mathbf{C}_{vf}$ are coefficient vectors and $\mathbf{f}$ is the applied force. The number of unknowns $(r_u\times r_v)$ is given by the user. For simplicity when evaluating the efficiency of the algorithm we shall assume that $r_u=r_v=r$. The system (30) is of the type $\mathbf{AXB}=\mathbf{D}$ described by Farin[15]. It can be solved in two passes:

1. $\mathbf{AY}=\mathbf{D}$ which is in fact solving r linear systems with r unknowns;
2. $\mathbf{XB}=\mathbf{Y}$, that is solving another set of r linear systems.

This approach requires $O(r^4)$ computations. Note that if we rearrange the unknowns in a column vector the number of computations will be $O(r^6)$. However, due to its special right hand side, (30) can be solved in a much more efficient way. It is sufficient to find only two solutions: $\mathbf{V}_u$ of the system $\mathbf{K}_{ur}\mathbf{X}=\mathbf{C}_{uf}$ and $\mathbf{V}_v$ of $\mathbf{K}_{vr}\mathbf{Y}=\mathbf{C}_{vf}$. Then $\Delta\mathbf{V}_r$ is expressed

$$\Delta\mathbf{V}_r = \mathbf{V}_u\mathbf{V}_v^T\mathbf{f}. \tag{31}$$

Due to this fact the number of required computations is reduced to $O(r^3)$.

3.4. Moving a surface point to a new position

Exploiting the idea for curves and using equation (31), the formula for computing the force which is required to move a surface point to a desired new target becomes

$$\mathbf{f} = \frac{\Delta \mathbf{p}}{\mathbf{B}_u^T \mathbf{V}_u \mathbf{V}_v^T \mathbf{B}_v}. \tag{32}$$

3.5. Solving for a set of forces

The user is free to specify a set of forces acting on a particular area of the surface. Unlike the curve case for surfaces only the first approach is possible. First the displacements due to every single force should be found and then they are to be summed to find the global displacement $\Delta\ \mathbf{V}$. The second approach is applicable only if some of the forces are applied on points with the same parameter value (either u or v), which is an unlikely situation.

3.6. Solving for forces on an embedded curve of the surface

Let $\mathbf{c}(t) = [u(t), v(t)]$ be a curve on the surface parameterized by t and let $\mathbf{f}(t)$ denote the forces acting on it. The curve $\mathbf{c}(t)$ can be obtained by specifying a set of points on the surface and then, for example, applying a spline interpolation. Since we have mathematical expressions for the curve and the force, we can compute a set of forces acting on the curve over equal parameter distances Δt. Then the problem is the same as for the previous section. Note that since we have formula (32) for computing the necessary displacement force, it is possible to move a surface curve $\mathbf{c}_0(t)$ to a new target curve in the space $\mathbf{c}_1(t)$.

3.7. Solving for forces on a specified area of the surface

Suppose we have an area on the surface specified by a closed curve. Exploiting the same ideas we can compute a set of forces acting in the specified area over equal parameter distances Δu=Δv. Then the problem can be solved as in section 3.5.

4. Evaluating the Fairness Norm and the Forcing Vector for the Nonuniform Case

It was already mentioned that for the uniform case the integrals (6), (10) and (24) can be computed exactly, which will speed up the algorithm. However, when performing the interactive sculpting, the user might decide that for some intervals the resolution is not high enough and might want to insert new knots in these areas. Then the parameterization will no longer be uniform and numerical methods must be used to solve (6), (10) and (24).

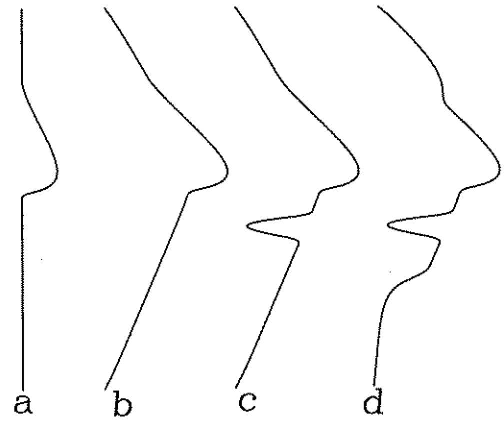


Figure 4: *Sculpting a face profile*

The approach using Gaussian quadrature[18] is very efficient for polynomial integrands. Its main idea is that

$$\int_{u_0}^{u_1} f(u)\, du = \sum_{i=1}^{N} w_i f(u_i), \tag{33}$$

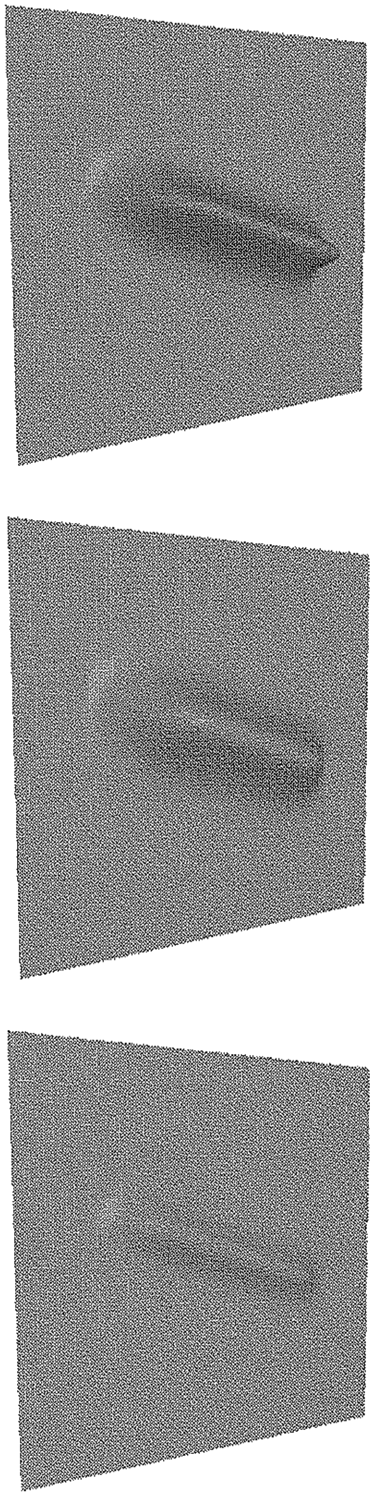
where w_i is a sequence of weights and u_i is a sequence of abscissas, whose computation is described by Press et al.[18] The expression (33) is exact when $f(u)$ is a polynomial of degree $(2N\text{-}1)$ or less. In our case the sufficient values for N are 2, 3 and 4, which makes the evaluation fast enough. Note also that the integrals need to be computed only if either of the knot sequences is modified, i.e. a new knot is inserted.

5. Implementation and Results

The algorithms were implemented on a Silicon Graphics Indy workstation using the OpenGL library to render the surfaces. The results are shown in Figures 4-6.

Figure 4 shows the different stages of sculpting a face profile starting from a straight line. The change in Figure 4a is medium broad and the force is not applied in the centre of the modified interval. Figure 4b exhibits a broad change that affects the whole curve. The change in Figure 4c is narrow and the force is applied in the middle of the modified area.

Figure 5 demonstrates the influence of the parameters α, β and γ. As it was mentioned before a bigger value for α means a surface that resists more to stretching, a bigger value for β means a greater resist-



a.

b.

c.

Figure 5: *Influence of the parameters:* **a.** α*=1*, β*=2*, γ*=0.1;* **b.** α*=1*, β*=20*, γ*=0.1;* **c.** α*=1*, β*=20*, γ*=0.5.*

ance to bending and a bigger value for γ represents a harder surface.

Figure 6 shows the stages of sculpting a human face starting from a plane and using different types of manipulations on a B-spline surface:

- Figure 6a: a broad deformation that affects the whole surface with forces applied on an area (convex shape);
- Figure 6b: forces on a line, which magnitude changes from zero to a certain value (nose);
- Figure 6c: forces applied on a curve (eyebrows);
- Figure 6d: narrow deformation with forces applied on a line (mouth);
- Figure 6e: forces on an area (forehead) and trimming the final shape.

The response time for a particular deformation depends on the breadth of the change (number of the affected control points) and on the length of the deformed curve (number of the applied forces). When up to five forces are applied it is in real time. The response for each operation performed in Figure 6 took less than five seconds.

6. Conclusions and Future Research

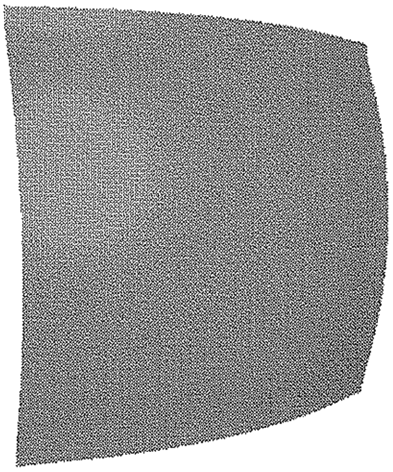
An efficient method for interactive sculpting of B-spline curves and surfaces was presented. Its main advantage is that the proposed energy functional for surfaces preserves the two dimensionality of the arrays of control and data points which speeds up the algorithm and allows a real time sculpting. The user applies forces as a main sculpting tool but it is also possible to move a point or a curve of the object to a new position. The user interface is based on a mouse controlled 3D cursor. An interesting further step of research would be to allow the user to interactively deform the surface in a virtual environment using different sculpting tools.

7. Acknowledgements

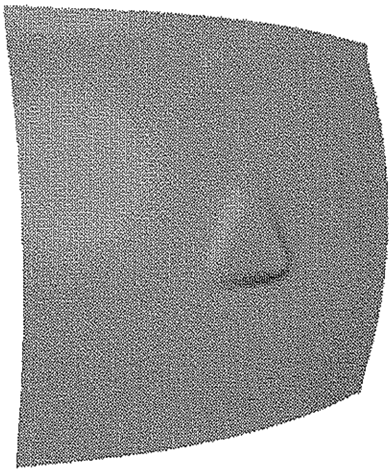
The author would like to thank the British Council and Microsoft who sponsored his work. He is also grateful to his supervisor Mel Slater for his support.

References

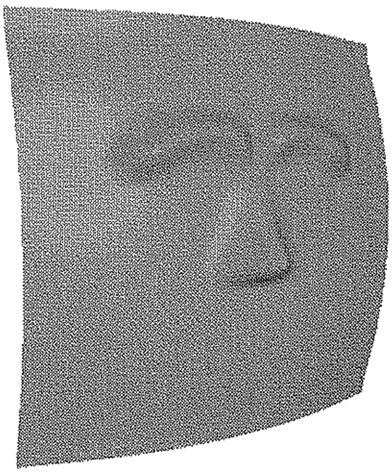
1. Hsu, W.M., Hughes J.F. and Kaufman H. Direct manipulation of free-form deformations. Proceedings of SIGGRAPH, Vol. 26, No 2, July, 1992, pp 177-184.
2. Finkelstein, A. and Salesin, D.H. Multiresolution curves. Proceedings of SIGGRAPH, Vol. 28, No 2, July, 1994, pp 261-268.



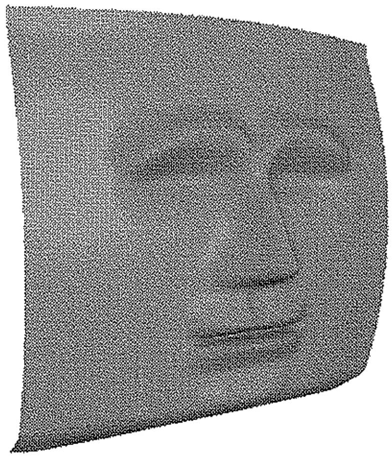
a.



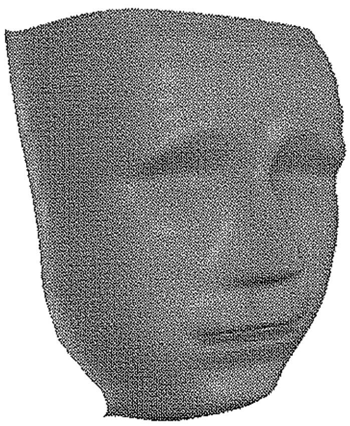
b.



c.



d.



e.

Figure 6: *Sculpting a face:* **a.** *Bending the whole surface;* **b.** *Nose;* **c.** *Eyebrows;* **d.** *Mouth;* **e.** *Face.*

3. Fowler, B. and Bartels, R.H. Constrained based curve manipulation. IEEE Computer Graphics and Applications, Vol. 13, No 5, 1993, pp 43-49.

4. Fowler, B. Geometric manipulation on tensor product surfaces. Proceedings of the 1992 Symposium on Interactive 3D Graphics, March 1992, pp 101-108.

5. Rappoport, A., Hel-Or, Y. and Werman, M. Interactive design of smooth objects with probabilistic point constraints. ACM Transactions on Graphics. Vol. 13, No 2, April 1994, pp 156-176.

6. Bloor, M. and Wilson, M.J. Representing PDE surfaces in terms of B-splines. Computer-Aided Design, Vol. 22, No 6, 1990, pp 324-331.

7. Thingvold, J. and Cohen, E. Physical modeling with B-spline surfaces for interactive design and animation. Proceedings of the 1990 Symposium on Interactive 3D Graphics, pp 129-137.

8. Celniker, G. and Gossard D. Deformable curve and surface finite elements for free-form shape design. Proceedings of SIGGRAPH, Vol. 25, No 4, July, 1991, pp 257-266.

9. Celniker, G. and Welch, W. Linear constaints for deformable B-spline surfaces. Proceedings of the 1992 Symposium on Interactive 3D Graphics, March 1992, pp 165-170.

10. Moreton, H.P. and Sequin, C. Functional optimization for fair surface design. Proceedings of SIGGRAPH, Vol. 26, No 2, July 1992, pp 167-176.

11. Welch, W. and Witkin, A. Variational surface modeling. Proceedings of SIGGRAPH, Vol. 26, No 2, July 1992, pp 157-166.

12. Halstead, M., Kass, M. and DeRose, T. Efficient, fair interpolation using Catmull-Clark Surfaces. Proceedings of SIGGRAPH, Annual series, 1993, pp 35-44.

13. Terzopoulos, D. and Qin, H. Dynamic NURBS with geometric constraints for interactive sculpting. ACM Transactions on Graphics, Vol. 13, No 2, April 1994, pp 103-136.

14. Qin, H. and Terzopoulos, D. Dynamic NURBS swung surfaces for physics-based shape design. Computer-Aided Design, Vol. 27, No 2, February 1995, pp 111-127.

15. Farin, G. Curves and Surfaces for Computer Aided Geometric Design. Academic Press, Inc. 1993.

16. de Boor, C. A Practical Guide to Splines. Springer-Verlag, New York, 1978.

17. Vassilev, T. A Method for Fair B-spline Interpolation and Approximation by Insertion of Additional Data Points. Submitted for publication.

18. Press, W. H., Flannery, B. P., Teukolsky, S. A. and Vetterling W. T. Numerical Recipes, The Art of Scientific Computations. Cambridge University Press, New York, 1989, pp 121-126.