Fast approximate G^1 surface blending to support interactive sculptured surface feature design

P.A. van Elsas and J.S.M. Vergeest

Delft University of Technology

*Faculty of Industrial Design Engineering,
Jaffalaan 9, NL-2628BX, The Netherlands.*

Telephone +31 015 2781196, Fax: +31 015 2787316.

Email : a.vanelas@io.tudelft.nl

Abstract

The support of designing features in free-form surfaces is a relatively new research topic. The designer needs tools to model objects in terms from his application domain, leaving the complex representation to the software. This is particularly crucial for a number of surface features (such as local depressions and protrusions), that involve the calculation of blending geometry between portions of surfaces in motion during modeling operations. Current surface blending techniques cannot support this adequately. Conceptual surface feature design requires the real-time solution of the blending problem in a semilocal way, which puts a constraint on algorithms for the derivation of good blends between two separated surface structures. On the other hand, some constraints conventionally imposed on surface blends, can be partly relaxed. This has enabled the development of a method suited for interactive surface protrusion/depression design, which was the driving application for this research. We have derived the customer requirements for a tool to support this. Based on these requirements a trade-off has been made between tangent continuity precision and the complexity of the algorithm to find blending geometry. The resulting method has been implemented in an existing conceptual shape modeling tool. This system has been subjected to tests. These demonstrate that protrusion/depression feature design is being supported with unprecedented flexibility and speed. Also, the approximations introduced into the method appear to be justified.

Keywords

Conceptual shape design, sculptured objects, surface features, blending algorithms, geometric continuity

1 INTRODUCTION

The problem of supporting conceptual design of mechanical/geometrical products is far from being solved. A recent test performed by professional designers yielded that none from a set of (believed to be the best suited) commercial design tools were able to support basic operations such as rearranging a product's structure (Hennessey (1993)). In the area of conceptual shape modeling, a generally acceptable tool is still nonexistent, despite some promising results on shape handles by Galyean and Hughes (1991) and methods to quickly create global shapes by van Dijk (1994a). Contemporary systems lack, in particular, the support of surface feature design, where a surface feature is some geometric aspect superimposed onto the current 3D object. An exception is formed by prismatic parts, for which feature modelers are available. However, many products of the nonprismatic type, from car bodies to hand-held devices, appear to contain surface features such as holes, ridges, protrusions and depressions.

This leads to dissatisfaction on the part of the designer since a significant part of the sculptured surface design process can be categorized as surface feature modeling. At present, modelers restrict the user to designing surface features in the same way as he would design macroscopic shapes. This implies a lot of detailing work, which must be redone every time the main object is changed. Especially during the conceptual phase of design, in which several shape proposals must be quickly created and evaluated, this is an unacceptable working procedure. Yet, professional users were able to specify concretely which properties a surface feature design tool should have. We have taken this set of specifications as the starting point in the search for a proper solution.

One technical obstacle to arrive at a satisfactory tool for conceptual surface feature design is caused by conflicting system requirements generally encountered for parametric design embedded in free-form objects. For one type of operation, displacement feature design, we have investigated this in detail. A displacement feature is defined as a local protrusion or depression in the shape of an object, resulting from repositioning some region of its surface. This region remains connected with the remainder of the object by some type of blending surface. An example of a displacement feature is shown in figure 1a). The protrusion in the car motorhood can be thought as been designed by specifying contour curves, a displacement height and rounding parameters for the transition surface near the edges of the feature. The basic technical issue is the fast and sufficiently accurate calculation of the blending geometry to construct this transition surface.

In section 2 we identify the requirements for conceptual displacement feature design as stated by potential customers. From these requirements a working procedure is derived for the creation and interactive modification of the features. In section 3 we briefly review existing surface blending techniques and in section 4 we focus on the specific mathematical issues related with the generation of transition geometry between separated surface structures. A construction method which effects this blending is presented in section 5. A discussion on the practically obtained results, on limitations of the method and on improvements is given in section 6.

2 CUSTOMER REQUIREMENTS ON THE DISPLACEMENT FEATURE

We have interviewed a number of professional designers in order to find what their expectations are of a useful tool to support displacement feature design. Some of the designers were used to computer tools, some were not. They all agreed that displacement features (as we defined them) are frequently occurring

Figure 1 Displacement features.

in objects they design. Although the features are often relatively small and local structures, they may strongly characterize the object's appearance. Therefore, displacement features cannot be regarded as details to be included at the end of the design process; they play an essential role already during the conceptual phase.

Those of the interviewed designers that had experience with 3D modeling tools confirmed that displacement feature design tends to be a time consuming process. The creation of such a feature involves a series of trimming and filleting operations, usually only partly automated. In addition, if the geometry in the neighbourhood of the feature is modified, part of or the entire feature has to be remade. For final design this could be acceptable economically. In conceptual design however, the demands for the support of surface features are quite different. This early stage of design is a creative process, in which a designer tries out several ideas, before choosing a few concepts to continue the design process with. A slow modeling tool disturbs these creative processes in an unacceptable way as shown by van Dijk (1994b).

From the suggestions that we obtained from the potential users we found that for conceptual displacement feature design 1) the creation and modification operations must take place by direct manipulation with immediate graphical feedback, 2) the controls must be intuitive to the designer. This came down to a procedural specification involving the following variables (See figure 1b):

1. A contour sketchable on the object's surface to specify the region.
2. The amount of displacement (height or depth).
3. The width of the transition geometry (which then determines the slope of the blend).
4. The sharpness of the roundings (near the displaced region and near the base surface).

Apart from these requirements, a tilting function (to obtain a variable height along the contour) was mentioned as a valuable extension. Examples of such features are common for car motorhoods, where height or depth typically vary between zero and ≈ 20 mm.

3 PREVIOUS WORK

Research on displacement features and the related problem of blending surface construction has progressed recently. Cavendish and Marin (1992a and 1992b) define a procedural method for the design of feature based functional surfaces. They note that explicit representation of feature based surfaces is often highly inefficient, since this would require large number of surface patches. Any design modifications would lead to redefinition of many patch borders and cross boundary derivatives.

Shimada, Balents and Gossard (1993) define an automatic triangular mesh generator, used to design functional free-form surfaces that contain arbitrarily shaped details, e.g bumps, grooves and holes. However, this approach is physics-based and requires an underlying FEM structure.

A different approach in the construction of surface features is the use of surface deformations (Sederberg and Parry (1986), Gascuel (1989), Coquillart (1990), Hsu, Hughes and Kaufman (1992)). These methods mimic traditional design tools such as sculpturing and moulding. A surface model is enclosed by a 3D lattice, which is then deformed. This deformation of the lattice is passed upon the model. Subdivision techniques are then used to create an acceptable resolution of the surface. In general, deformation techniques are only capable of creating bumps, although the welding techniques described by Gascuel can also create holes.

Another method to create surface protrusions or depressions is based on local refinement of the NURBS representation of the surface (Forsey and Bartels (1988)). In the region where the B-spline control points are made denser the user can be given control over the shape within this regions without affecting the remainder of the surface.

Finally, a number of methods is based on the notion that a displacement feature can be regarded as a structure consisting of three parts; a region moved away from the base surface, a transition surface connecting this displaced region with the base surface, and the base surface minus the removed region. The main issue of these methods is the construction of a smooth transition surface bounded by two given curves, one laying on the base surface, one on the displaced region. One way to obtain the required smoothness is to first construct a positional continuous transition surface, e.g. a ruled surface (which will in general produce sharp connections), and then create fillets to round the edges. Procedures to compute fillets are described by Klass and Kuhn (1992) and Chuang and Lien (1995). Another way to obtain a smoothly connecting transition surface is to base its shape directly on the positional and tangential (or even higher order) geometry of the surfaces at the aforementioned two given curves. Filip (1989) defines a procedure for the construction of such a blending surface is presented for general parametric surfaces, and Kim and Elber(1995) define one specialized to NURBS parameterizations. To avoid the high polynomial degree of the transition surface, several approximations are proposed. Bardis and Patrikalakis (1989) and Filkins, Tuohy and Patrikalakis (1993) approximate the curves on the surfaces by integer cubic B-spline curves, which in general will not be exactly on the surface. Similarly, G^1 continuity across the boundary of transition surface is approximated within a user-specified precision. Woodward (1986) and Vida, Martin and Varady (1994) have written excellent surveys on blends.

4 ON THE TRANSITION GEOMETRY FOR DISPLACEMENT FEATURES

For the construction of the displacement feature we follow the blending approach discussed in the previous section. This seems the most promising approach since, for the displacement feature, the two edges for

Figure 2 Derivation of a transition surface.

the transition surface are explicitly available. Furthermore we need, as the end result, common NURBS surfaces, which favors the application of nonprocedural blending.

In this section we mention some geometrical issues concerning the construction of the transition surface. For this purpose, in this section, we slightly simplify the definition of the displacement feature by assuming that the object is represented by its volume boundary surface S . For the actual method (described later) we allow that this base surface S consists of multiple surfaces forming a patch work. In this section, we further assume that S is bounded, arcwise connected, not selfintersecting and that it is mappable to a rectangle. Two curves on S are given, c and d . Each curve forms a loop and does not selfintersect except at its end points. They can be regarded as being defined by the sketched contour and a user-defined width, although we do not assume this in the following. In the topological space S , c encloses d , where parts of or the entire curve d may coincide with c . Thus S is partitioned into three parts: S_1 is the exterior of c ; S_2 is the region bounded by both c and d ; S_3 is the region enclosed by d . If c and d are identical, the region S_2 degenerates into a curve. For this discussion we assume that each surface is identical to its closure.

Two operations are now performed on the structure. First the region S_2 is deleted. Then region S_3 is displaced using an arbitrary rigid body transformation T . This defines the surface $S'_3 = TS_3$ bounded by $d' = Td$ (Figure 1b).

We now wish to construct a transition or blending surface S_T that connects c with d' . We also need control over the roundness of S_T ; the smoothness of its contact with S_1 and with S'_3 must be independently variable. We introduce two nonnegative control parameters for the roundness of the blending, R_1 and R_2 . The condition represented by R_1 is as follows (the condition due to R_2 is similar).

For each point p on the curve c we imagine the intersection curve formed by S_T and the plane through

$\mathbf{p}$ perpendicular to $\mathbf{c}$. We consider only the part of this intersection curve that lies inside the sphere with centre $\mathbf{p}$ and radius R_1 . Following this planar curve, starting at $\mathbf{p}$, there may be no inflection point until the curve reaches the sphere; in addition the integrated curvature up to there must be between zero and $\pi/2$. Less formally, R_1 means that $\mathbf{S}_T$ extends roughly over a distance R_1 in which no bending of more than $\pi/2$ occurs.

How the bending actually develops depends on the placement of $\mathbf{S}'_3$ relative to $\mathbf{S}_1$. We want that in between the two contours the curvature of $\mathbf{S}_T$ is smoothly interpolated; in the case that for each $\mathbf{p}$ on $\mathbf{c}$ an appropriate corresponding point on $\mathbf{d}'$ can be defined it could be demanded that the shortest curve on $\mathbf{S}_T$ between these two points has at most one inflection point. Also we want that no folds emerge in the interior of $\mathbf{S}_T$. Informally this can be defined as an upper limit on the curvature of $\mathbf{S}_T$ perpendicular to the shortest curve just mentioned. Finally $\mathbf{S}_T$ must not be selfintersecting.

4.1 Conditions for the starting situation

It is easily possible to specify a base surface, boundaries and a rigid body transformation as a starting situation for which *no* blending surface $\mathbf{S}_T$ exists with the above mentioned properties. We therefore need additional conditions for the starting situation, for which a solution can be found. It can be shown that, if T is the identity operator, even the surface $\mathbf{S}_2$ sometimes fails to meet the requirements.

We proceed with reducing the complexity of the starting situation. First, we assume that both $\mathbf{c}$ and $\mathbf{d}'$ are specified as mappings from the finite interval $[s_0, s_1]$ into 3D. The parameterizations of the two curves are made consistent by imposing that the integrated distance $|\mathbf{c}(s) - \mathbf{d}'(s)|$ is minimal, where s runs over $[s_0, s_1]$. Of each curve, their end points meet with tangent direction continuity.

Let us first look at the case $R_1 = R_2 = 0$. The curvature of $\mathbf{S}_T$ near its boundaries is then allowed to be infinitely large, which means that there is in general no tangent plane continuity between $\mathbf{S}_T$ and the two surfaces $\mathbf{S}_1$ and $\mathbf{S}'_3$. Then $\mathbf{S}_T$ can be simply constructed as the ruled surface between the two contours, parameterized e.g. as :

$$\mathbf{S}_T(s, t) = (1 - t)\mathbf{c}(s) + t\mathbf{d}'(s), \quad (1)$$

on the domain $[s_0, s_1] \times [0, 1]$.

Depending on the shapes of the contour curves and on the position of $\mathbf{S}'_3$ relative to $\mathbf{S}_1$ this ruled surface may be selfintersecting. To avoid this we require that there exists a plane onto which the neighbourhoods of and the interiors of $\mathbf{c}$, $\mathbf{d}$ and $\mathbf{d}'$ can be projected without introducing intersections between the projected curves. If, still in this projection, for each $a \in [s_0, s_1]$ the point $\mathbf{c}(a)$ is visible from both $\mathbf{d}(a)$ and $\mathbf{d}'(a)$ then $\mathbf{S}_T(s, t)$ of equation (1) will not selfintersect. $\mathbf{S}_T$ will contain sharp edges along the t -direction for those parameter values s where either $\mathbf{c}$ or $\mathbf{d}$ have sharp corners. However, these folds can be regarded as "natural" and as not invalidating $\mathbf{S}_T$.

We now consider the case that R_1 is finite (for finite R_2 we obtain similar results). If $R_1 > 0$ then the designer expects tangent plane continuity between $\mathbf{S}_T$ and $\mathbf{S}_1$ everywhere along $\mathbf{c}$. If $\mathbf{S}_1$ is not strongly curved near $\mathbf{c}$ and if also the curvature radius of $\mathbf{c}$ is everywhere larger than R_1 then it is possible to construct $\mathbf{S}_T$ in such a way that it meets the conditions imposed by the value R_1 (figure 3). If, however, $\mathbf{c}$ has a smaller curvature radius than R_1 at some point, then a local selfintersection of $\mathbf{S}_T$ could be introduced in the neighbourhood of that point. As an extreme case consider a sharp corner in $\mathbf{c}$. In the neighbourhood of the corner, $\mathbf{S}_T$ must be almost flat to fulfill the tangent plane continuity condition. This condition could put a stronger limit on the permitted curvature of $\mathbf{S}_T$ than the $\pi/2$ specified above. Finally, we need to take care for macroscopic selfintersections of $\mathbf{S}_T$. This could happen if the sizes of

R_1 and R_2 are comparable to or larger than the diameter of $\mathbf{d}'$; then there is the risk that $\mathbf{S}_T$ intersects itself at the "opposite side".

4.2 How to construct $\mathbf{S}_T$

We first consider those situations in which the above mentioned problems do not occur. Like in equation (1) we want to find a parameterization $\mathbf{S}_T(s, t)$ on $[0, 1] \times [s_0, s_1]$. We know that then $\mathbf{S}_T(s, 0) = \mathbf{c}(s)$ and $\mathbf{S}_T(s, 1) = \mathbf{d}'(s)$. In principle we have now reduced our problem to a classical situation of G^1 blending of the two surfaces $\mathbf{S}_1$ and $\mathbf{S}_2'$. What we still need is boundary derivative information along the curves $\mathbf{c}$ and $\mathbf{d}'$ in order to achieve tangent plane continuity. Since $\mathbf{S}$ is smooth, a surface normal vector is defined everywhere on it, and hence on the boundaries $\mathbf{c}$ and $\mathbf{d}'$. We assume that unit normal vectors are, like the curves themselves, available as parameterizations from $[s_0, s_1]$ into 3D. For each $a \in [s_0, s_1]$ we can construct a derivative vector $\mathbf{b}_1(a)$ starting at $\mathbf{c}(a)$ in the tangent plane of $\mathbf{S}_1$ at that location and pointing away from $\mathbf{S}_1$. There is some freedom in the orientation that $\mathbf{b}_1(a)$ has in the tangent plane; a reasonable choice seems to make it locally perpendicular to $\mathbf{c}$. Also the length of the derivative vector is still not determined. There is some relation between this length (the parameterization speed in the t -direction) and the risk of a selfintersection in $\mathbf{S}_T$. In the same way as we did for $\mathbf{b}_1(a)$ we can construct the derivative vector $\mathbf{b}_2(a)$ starting at $\mathbf{d}'(a)$. With these four vector valued functions the standard Hermite interpolant can be made :

$$\mathbf{S}_T(s, t) = H_0^3(t)\mathbf{c}(s) + H_1^3(t)\mathbf{b}_1(s) + H_2^3(t)\mathbf{d}'(s) + H_3^3(t)\mathbf{b}_2(s), \quad (2)$$

on the domain $[s_0, s_1] \times [0, 1]$, where the $H_i^3(t)$ are the cubic Hermite interpolation functions.

4.3 Complexity of the parameterization

For the construction of $\mathbf{S}_T$ there is the problem that the parameterizations of the contour curves and of the derivative curves will get complex. For example, suppose that $\mathbf{S}$ is represented as a NURBS surface of degree $m \times m$ and both contours are NURBS curves of degree n in the 2D parameter domain of $\mathbf{S}$. Then $\mathbf{c}$ and $\mathbf{d}'$ are NURBS curves of degree $2nm$. Even if these curves are geometrically smooth their parameterizations exhibit discontinuities from two sources. If the parameter domain curve crosses a knot line of multiplicity k ($k > 0$) then the surface curve is discontinuous in its $(m - k)$ th derivative, in other words it is C^{m-k-1} continuous at that point. In the special case that the surface curve is an isoparameter curve on $\mathbf{S}_T$, it is a NURBS curve of degree m with a knot of multiplicity k . In general the surface curve is representable as a NURBS curve of degree $2nm$ with a knot of multiplicity k . The second source of discontinuity in the surface curve is any discontinuity in the 2D NURBS curve itself. A knot multiplicity of k' generates a discontinuity in the $(n - k' + 1)$ th derivative of the parameter domain curve and hence, in general, also for the surface curve. This leads to a knot of multiplicity $2nm - (n - k' + 1) + 1 = 2nm - n + k'$ for its NURBS representation. If both types of discontinuity occur at the same place (which can well happen in practice), then the knot multiplicity is $\max(k, 2nm - n + k')$.

We wish to explicitly parameterize $\mathbf{S}_T(s, t)$ as a NURBS surface. Then, in the situation that $R_1 = R_2 = 0$ we can apply equation (1) provided that both $\mathbf{c}(s)$ and $\mathbf{d}'(s)$ are parameterized consistently, i.e. they have the same polynomial degree and the same number of knots. Then $\mathbf{S}_T$ will be of degree $2nm$ in the s -direction and of degree 1 in the t -direction. The structure of the s -knots for $\mathbf{S}_T$ can be obtained by first scaling the parameter range of one of the contours such that it gets equal to the range

of the other contour. Then the knots from $\mathbf{c}$ must be merged with those of $\mathbf{d}'$ in the following way. For each $a \in [s_0, s_1]$ a knot must be placed with multiplicity $\max(k, k')$, where k and k' are nonnegative and represent the multiplicities of the knots of $\mathbf{c}$ and $\mathbf{d}'$ at a , if any, respectively. It should be noticed that not only the values of the knots can be different for the two contour curves, but also their amount. The latter situations happens when the two contour curves intersect a different number of knot lines in the parameter domain of $\mathbf{S}_T$.

In the case that the R_i are nonzero the surface $\mathbf{S}_T(s, t)$ gets much more complex (see equation (2)) due to the derivative vector $\mathbf{b}_1(s)$, and $\mathbf{b}_2(s)$. First these functions must be parameterized on the same interval that we constructed above for the two contour curves. Second, the polynomial degree of the derivative vectors can become $(2nm - 1)^3$ as this vector is constructed by an outer product of a surface normal vector with a tangent vector. This would increase the polynomial degree of $\mathbf{S}_T$ as well. Lower degree parameterizations are only possible for some special situations.

5 METHOD FOR INTERACTIVE DISPLACEMENT FEATURE CREATION AND MODIFICATION

Due to a number of specific requirements for the displacement feature, the application of a blending technique should take account of the following issues:

1. The edges of the transition surface must be exactly on their associated surface (within graphical display accuracy). No gaps or excess geometry are permitted since this would be immediately distracting to the designer, as has been observed in evaluation experiments (van Dijk (1995)). Therefore we decided to compute the contour curve through the exact conversion from parameter space curve to geometrical space curve.
2. If the contour curve in parameter space exhibits G^1 discontinuities, so will, in general, the associated geometric contour curve. Since the user expects a smooth contour curve its parametric origin must also be smooth. This can be accomplished by a polyline interpolating very densely spaced points in parameter space or by smooth curve in parameter space. The polyline would result in a NURBS contour curve of degree $2m$ if the base surface is a NURBS surface of degree $m \times m$. If in parameter space the polyline is replaced by a NURBS curve of degree m then the contour curve will be of degree $2nm$. From the conclusions drawn in the previous section, and given the fact that current implementation platforms maximally support polynomial spline degrees of 20, this requirement limits the NURBS degree for base surface and contour curve (in parameter space) to 3; then $2nm = 18$ according to equation (1).
3. If no tangent plane continuity between $\mathbf{S}_T$ and the two adjacent surfaces is required (i.e. $R_1 = R_2 = 0$) then an exact NURBS solution with polynomial degree $2nm \times 1$ is feasible using equation (1).
4. Approximate tangent plane continuity across the borders of $\mathbf{S}_T$ can be obtained by choosing appropriate non-boundary control points for $\mathbf{S}_T$.
5. If the roundings R_i are larger than the curvature radius of contour $\mathbf{c}$ or $\mathbf{d}'$ at any of its points then a fundamental inconsistency arises; the R_i need to be reduced locally to resolve this.

In the next section we describe an implementable method that takes account of each of these.

We have chosen to present the designer an image of the displacement features directly after he sketches the contour. The height, width and roundings are set to some default values, which can then immediately be modified by the user.

5.1 Calculation of the contour curves from the sketch

The sketched contour is transformed into the two contour curves c and d' , using inverse parameterization techniques for the sampled screen locations and by polynomial basis conversion. If S is converted to power basis form, and the power basis representation of the sketched contour (in 2D parameter space) is evaluated in S , we obtain two sequences c_i and d'_i of (possibly rational) Bézier curves of degree $2nm$. A curve splitting algorithm generates a new piece wherever a knot line is crossed in parameter space.

5.2 Construction of the transition surface

Figure 3 Derivation of a transition surface.

The transition surface S_T is constructed as the union of pieces S_{T_i} , each connecting Bézier curve c_i with its displaced counterpart d'_i . Each S_{T_i} is represented as a Bézier patch of degree $2nm \times 3$ with control points e_{ij} , $i = 0, \dots, 3$, $j = 0, \dots, n$. The two exterior rows e_{0j} and e_{3j} are copied from the two Bézier curves. The two interior rows of control points are calculated as follows. For each point e_{1j} a point p_j is sampled on c_i (for the row e_{2j} a similar method is used). At p_j a direction vector v_{1j} is calculated using the cross product of the surface normal of S and the curve tangent of c_i defined locally. The resulting vector v_{1j} is perpendicular to c_i , and lies in the tangent plane of S defined at p_j .

The corresponding control point $\mathbf{e}_{1j}$ can now be chosen using $\mathbf{v}_{1j}$. If $\|\mathbf{v}_{1j}\| = 0$ then $\mathbf{e}_{1j} = \mathbf{p}_j$. If $\|\mathbf{v}_{1j}\| > 0$ then $\mathbf{e}_{1j} = \mathbf{p}_j + r_1 \mathbf{v}_{1j}$, where r_1 represents the user defined length of $\mathbf{v}_{1j}$. Note that if a user demands an explicit value of R_i , the parameter r_1 has to be chosen in such a way that S_T geometrically extends as far as R_i .

In section 4 we found that contour radii smaller than the rounding parameters may cause a selfintersecting blend surface. We therefore cannot apply equation (2) locally to each pair of curve pieces. Instead we decrease the r_1 and/or r_2 value wherever necessary using a separation plane. Consider figure 3a), where $\mathbf{c}_{i-1}(t)$ and $\mathbf{c}_i(t)$ meet at a point $\mathbf{p}_i$. The separation plane at $\mathbf{p}_i$ contains the surface normal $\mathbf{n}_s$, and the average of two curve tangents $\mathbf{t}_{i-1}$ and $\mathbf{t}_i$. A similar separation plane is constructed at point $\mathbf{q}_i$ on curve $\mathbf{d}'_i$.

The control points $\mathbf{e}_{10}$ and $\mathbf{e}_{1n}$, and $\mathbf{e}_{20}$ and $\mathbf{e}_{2n}$, shared between S_{T_i} and its two adjacent transition surface pieces are chosen such that they lie in their respective surface tangent planes, and also in their respective separation planes (See figure 3b).

The remaining (interior) control points of each S_{T_i} are determined by first linearly interpolating the vectors $\mathbf{e}_{20} - \mathbf{e}_{30}$ and $\mathbf{e}_{2n} - \mathbf{e}_{3n}$ along $\mathbf{c}_i(t)$. Then $n - 2$ points are located on the interpolation line. The projection of such a point on a surface tangent plane calculated at a location near the point becomes an interior control point. The control points $\mathbf{e}_{1j}$ are found in a similar way. In this way approximate tangent plane continuity across the feature boundaries is achieved.

6 RESULTS, EVALUATION AND RECOMMENDATIONS

(a) A default displacement feature.

(b) Comparing surface normals on either side of the feature boundary.

Figure 4 Effective G^1 blending approximation

All examples have been produced on a Silicon Graphics Crimson/RE workstation. The software was written in C++ using the Open Inventor graphics library from Silicon Graphics, and has been integrated into the Fast Shape Designer system (van Dijk (1994)). We have tested the method with parameter curve degree $n = 1$ and $n = 3$. Both cases give visually smooth results. For $n = 3$ the parameter curves need to

(a) Alteration of height, rounding and scaling parameters.

(b) A complex displacement feature on the well known Utah teapot.

Figure 5 Examples of displacement features.

be split into much fewer pieces, making the calculation faster. The protrusion in figure 4a) was produced in approximately 0.5 seconds after the contour was sketched.

The appearance of this feature can be altered interactively using any of the controls mentioned earlier. In figure 5a) it is shown what effect the different parameters have. The two roundness parameters R_1 and R_2 have been chosen relatively large, but the transition surface remains smooth. Controlling the rounding takes approximately 0.1 seconds, while alteration of the height takes only 0.01 seconds. Alteration of the transition width takes approximately 0.3 seconds, as a large part of the feature needs to be recalculated. The tests indicate that conceptual displacement feature design is possible at interactive speed, producing visually pleasing results.

We have explicitly determined how well G^1 continuity is achieved. The following test was performed on the displacement feature of figure 4a). While moving along a feature boundary, pairs of surface normals on both sides of the boundary were calculated, one surface normal on the base surface and the other directly opposite on the transition surface. If exact tangent plane continuity exists along the feature boundary, these pairs of surface normals must be identical.

In figure 4b), the results of this test are shown for two cases. In the first case the rounding parameter was set to 1 mm, which is approximately a factor 100 smaller than the entire feature, while in the second case, this rounding was decreased by a factor of two. We compared 1200 sampled normals on both sides to each other. The angle between each pair of surface normals was calculated. These results were then used to create the histogram showing the distribution of the found angle values.

In both cases the deviation was found to be considerably less than 1 degree. The results of test 1 are better due to the fact that the rounding parameter in test 2 was a factor two smaller. A shift of the distribution to higher angle values for smaller R_i could thus be expected.

Finally, we show a displacement feature created on the well known Utah teapot in figure 5b). Even on a more complex model like this teapot, a smooth displacement feature is easily created. The test results indicate that it is possible to design complex displacement features at interactive speed.

Several improvements are possible with regard to the offered functionality and with regard to the current implementation.

The functionality could be extended by allowing the designer to specify a tilt of the displaced region (again, by direct manipulation). This is required, for example, to create the surface feature on the motorhood shown in figure 1a). In some cases the designer may desire to modify the shape of the displaced region while (or immediately after) creating the feature. This would imply a tight integration of general shape design with parametric shape design. Finally, extension of the function toward other types of surface features such as holes (which can be considered as a special case of displacement features) and ridges could be useful.

Also the current implementation has some shortcomings. Although we specified the properties of the required roundings near the edges of the transition surface using the quantities R_i we did not implement them as such. The user control has been implemented by the variable length of the vectors $\mathbf{e}_{20} - \mathbf{e}_{30}$ etc., which is usually sufficient for concept generation. In some cases, however, it could be needed to specify the values R_i explicitly

In some applications continuities of higher order than G^1 are necessary. This could be obtained with approximation, e.g., starting from the method proposed by Filip (1989).

We have shown that we approximate tangent plane continuity quite effectively. However, it is not clear if we have reached an optimal solution yet. Instead of interpolating the two tangent vectors at the corners of each piece of the transition surface, we could calculate intermediate tangent vectors and interpolate those, as is done, e.g., in (Bardis and Patrikalakis (1989)). G^1 continuity can then be obtained within any tolerance by taking a sufficiently many intermediate tangent vectors. However, in tests preformed so far, we did not observe the need for this precision.

It becomes difficult to obtain smooth results if the specified R_i are extremely large in the neighbourhood of sharp corners. Although the use of separation planes will prevent any self intersections locally, in this case the resulting transition geometry might show a sharp decrease in its rounding. Also, there are some drawbacks with the correspondence of points on the two contour curves. In some extreme situations it is necessary to further break up the contour curves into pieces. The reduction of the length of the tangent vectors depends on the sharpness of two adjacent pieces of the contour curve. A more general method would be to determine this reduction locally from the curvature at each point on the contour curve.

Theoretically, overlapping features are possible. The current implementation would produce a very high polynomial degree for two overlapping transition surfaces.

Important is the behaviour of the feature after a global change to the underlying surface model. The feature might naturally adapt itself to the changed surface model, or it could remain fixed, if possible. Currently, any alteration of the surface model would lead to a displacement feature that has a different shape, which might not always be what is required.

Future work will deal with these topics, and both the method and implementation will be tested and evaluated by professional industrial designers.

REFERENCES

- Bardis, L. and Patrikalakis, N.M. (1989) Blending Rational B-Spline Surfaces. *Proceedings of Eurographics 1989*, Hamburg FRG., 4-8 september 1989, Eds. Hansmann W., Hopgood, F.R.A., Strasster, W., Elsevier Science Publishers, Amsterdam, 453-462.
- Cavendish, J.C. and Marin S.P. (1992) Feature Based Surface Design and Machining. *IEEE Computer and Graphics & Applications*, 61-68.

- Cavendish, J.C. and Marin S.P. (1992) A Procedural Feature-Based Approach for Designing Functional Surfaces. *Topics in Surface Modeling*, Ed. Hagen, H., SIAM, Philadelphia 1992, 145-168.
- Chuang, J.H. and Lien, F.L. (1995) One and Two-Parameter Blending for Parametric Surfaces. *Proceedings of the Third Pacific Conference on Computer Graphics and Applications*, Seoul, Korea, 21-24 august 1995, Eds. Shin, S.Y., Kunii, T.L., World Scientific, 333-346.
- Coquillart, S. (1990) Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling. *Proceedings of SIGGRAPH 1990*, august 6-10, Dallas, Texas, **24** no. 4, Ed. Beach, R.J., ACM, New York, 1990, 187-196.
- Filip, D.J. (1989) Blending parametric surfaces. *Transactions on Graphics*, **8**, 164-173.
- Filkins, P.C., Tuohy, S.T. and Patrikalakis, N.M. (1993) Computational Methods for Blending Surface Approximation. *Engineering with Computers*, **9** no. 9, Springer-Verlag, London Limited, 49-62.
- Forsey, D.R. and Bartels, R.H. (1988) Hierarchical B-Spline Refinement. *Proceedings of SIGGRAPH 1988*, august 1-5, Atlanta, Georgia, ACM, New York, **22** no. 4, 205-212.
- Galyean, T.A. and Hughes, J.F. (1991) Sculpting: an interactive volumetric modeling technique. *Proceedings of SIGGRAPH 1991*, july 28-august 2 1991, Las Vegas, Nevada, Ed. Sederberg, T.W., ACM, New York, **25** no. 4, 205-212.
- Gascuel, M.P. (1989) Welding and Pinching Spline Surfaces: New Methods for Interactive Creation of Complex Objects and Automatic Fleshing of Skeletons. *Proceedings of Graphics Interface*, London, Ontario, 19-23 june, Canadian Information Processing Society, Toronto, 20-27.
- Hennessey, J.M. (1993) Computerized conceptualizing in industrial design. *Proceedings of Interface*, may 1993, Raleigh, NC. USA, Human Factors and Ergonomics Society, Santa Monica, 112-117.
- Kim, K. and Elber, G. (1995) A Symbolic Approach to Freeform Surface Blends. *Proceedings of the Third Pacific Conference on Computer Graphics and Applications*, Seoul, Korea, 21-24 august 1995, Eds. Shin, S.Y., Kunii, T.L., World Scientific, 348-360.
- Hsu, W.M., Hughes, J.F. and Kaufman, H. (1992) Direct Manipulation of Free-Form Deformations. *Proceedings of SIGGRAPH 1992*, 26-31 july, Chicago, Ed. Catmull, E.E., ACM, New York, **26** no. 2, 177-184.
- Klass, R. and Kuhn, B. (1992) Fillet and Surface Intersection defined by Rolling Balls. *Computer Aided Geometric Design*, Eds. Barnhill, R.E., Boehm, W. Elsevier Science Publishers, North Holland, **9**, 185-193.
- Sederberg, T.W. and Parry, S.R. (1986) Free-form Deformation of Solid Geometric Models. *Proceedings of SIGGRAPH 1986*, august 18-22, Dallas, **20** no. 4, Eds. Evans, D.C., Athay, R.J., ACM New York, 151-160.
- Shimada, K., Balents, B. and Gossard, D.C. (1993) Automatic Mesh Reconstruction for Feature Based Sculpting of Deformable Surfaces. *Proceedings of Geometric Modeling for Product Realization*, 165-188.
- van Dijk, C.G.C. (1994) Evaluation of a surface modeler for conceptual design. *Proceedings of the 1994 Lancaster International Workshop on Engineering Design, CACD 1994*, 11-13 april, Lancaster, Eds. Sharpe, J., Oh, V., Lancaster University Engineering Design Centre, 149-169.
- van Dijk, C.G.C. (1994) Interactive Modeling of Transfinite Surfaces with Sketched Design Curves. *Thesis, Delft University of Technology*, ISBN 90-407-1037-6.
- van Dijk, C.G.C. (1995) New insights in computer-aided conceptual design. *Design Studies*, **16** no. 1, Ed. Cross, N., Elsevier Science Ltd., Great Britain, 62-80.
- Vida, J., Martin, R.R. and Varady, T. (1994) A survey of blending methods that use parametric surfaces. *Computer Aided Design*, Eds. Woodward, J., Piegl, L., Butterworth-Heinemann Ltd., **26** no. 5, 341-365.
- Woodward, J.R. (1986) Blends in Geometric Modeling. *Proceedings of The Mathematics of Surfaces II*, september 1986, University College, Cardiff, Ed. Martin, R.R., Clarendon, Oxford, 1987, 255-297.