Hybrid Symbolic and Numeric Operators as Tools for Analysis of Freeform Surfaces*

Gershon Elber† and Elaine Cohen
Computer Science Department,
University of Utah

November 5, 1992

Abstract

Freeform surfaces are commonly used in Computer Aided Geometric Design, so accurate analysis of surface properties is becoming increasingly important. In this paper we define *surface slope* and *surface speed*, develop visualization tools, and demonstrate that they can be useful in the design process. Generally, surface properties such as curvature and twist are evaluated at a finite set of predetermined samples on the surface. This paper takes a different approach. A small set of tools is used to symbolically compute surfaces representing curvature, twist and other properties. These surfaces are then analyzed using numeric techniques.

The combination of symbolic computation to provide an exact property representation (up to machine accuracy) and numerical methods to extract data is demonstrated to be powerful and robust. This approach supports a uniform treatment once the surfaces are computed and also provides global information, so questions such as 'is a surface developable?' or 'what are the hyperbolic regions of a surface?' can be answered robustly.

1 Introduction

Sculptured surface representations are fundamental forms in computer graphics and in computer aided geometric design. During different stages of modeling with sculptured

*This work was supported in part by DARPA (N00014-91-J-4123) . All opinions, findings, conclusions or recommendations expressed in this document are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

†Appreciation is expressed to IBM for partial fellowship support of the first author.

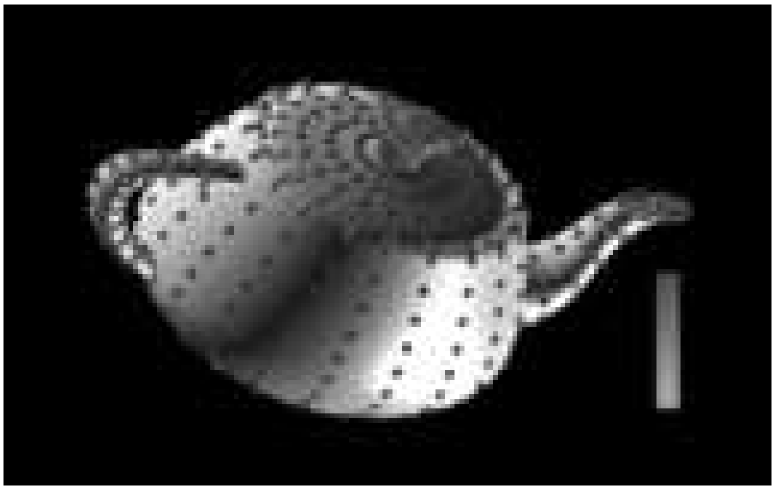

Figure 1: Surface curvature bounds computed at a predefined grid (See also figure 15).

surfaces, quite a few properties of the surfaces may be of interest to the designer or required for a proper design. The designer may need to isolate regions with surface slopes, defined in this paper, which are too high or too low, to detect all regions with twists larger than prespecified values, to have a visual bound on the distance traveled in the Euclidean domain while moving in the parametric domain (which we refer to as speed bound), or even to isolate all the hyperbolic (saddle) regions in the model.

Previous work directed at computing first and second order surface properties evaluated them over a discrete grid. Normals were computed and visualized by drawing them as arrows, called "hedgehogs" [20], over the grid. There have been attempts [1, 2, 3, 11] to understand and compute second order surface properties such as mean and Gaussian curvatures, as well as twist, by evaluating them over the predefined grid (figure 1).

Given a surface $S(u, v)$, there is no common method to accurately subdivide S into convex, concave, and saddle regions. Using symbolic tools developed in section 2 this trichotomy becomes feasible [13], as is demonstrated in section 3.

In section 2 we describe the required symbolic computation tools so properties such as Gaussian curvature, surface normal, surface slope, surface twist, and surface speed bound may be computed and represented as freeform surfaces. We call such derived surfaces *property surfaces*. We emphasize the NURBs and Bézier representations although other representations could be used, including any (piecewise) polynomial or (piecewise) rational representations. In section 3 we apply these tools to some examples and demonstrate their effectiveness. Visualization is used extensively in the section to communicate the relationship these properties have with the shape of the surface.

2 Background

Surprisingly enough the set of symbolic tools one needs for the analysis treated here is small. One needs to have representations for the derivative, sum, difference, and product of scalar curves and surfaces. Any manipulation of curves or surfaces using these tools will result in a curve or a surface of the same type. The resulting curve or surface is exact to within the accuracy of the numerical computation, since these operation have closed forms and are, in fact, symbol manipulators. Therefore, we refer to the usage of these tools as *symbolic* computation.

Contouring will also be used as a tool to extract information from the symbolically computed property surfaces.

2.1 Symbolic Tools

Given a Bézier or NURBs curve, the form of the derivative as a curve in vector space is well known [8],

$$\frac{dC(t)}{dt} = \frac{d\sum_{i=0}^{m-1} P_i B_i^k(t)}{dt} = (k-1)\sum_{i=0}^{m-2} \frac{(P_{i+1} - P_i)}{t_{i+k} - t_i} B_i^{k-1}(t), \tag{1}$$

and this result easily extends to tensor product surfaces.

The symbolic computation of sum and/or difference of two scalar Bézier or NURBs curves is achieved by computing the sum and/or difference of their respective control points [8, 9, 14], once the two curves are in the same space. This requirement can be met by representing them as curves with the same order (using degree raising [5, 6] on the lower order one, if necessary) and the same continuity (using refinement [7] of knot vectors for NURBs).

$$\begin{aligned} C_1(t) \pm C_2(t) &= \sum_{i=0}^{k} P_i B_{i,\tau}^k(t) \pm \sum_{i=0}^{k} Q_i B_{i,\tau}^k(t) \\ &= \sum_{i=0}^{k} \left(P_i B_{i,\tau}^k(u) \pm Q_i B_{i,\tau}^k(u) \right) \\ &= \sum_{i=0}^{k} (P_i \pm Q_i)\, B_{i,\tau}^k(u). \end{aligned} \tag{2}$$

This result easily extends to tensor product surfaces as well.

Representation for product of scalar curves is the last requirement. For Bézier curves [9, 8],

$$
\begin{aligned}
C_1(t)C_2(t) &= \sum_{i=0}^{m} P_i B_i^m(t) \sum_{j=0}^{n} Q_j B_j^n(t) \\
&= \sum_{i=0}^{m} \sum_{j=0}^{n} P_i Q_j B_i^m(t) B_j^n(t) \\
&= \sum_{i=0}^{m} \sum_{j=0}^{n} P_i Q_j \frac{\binom{m}{i}\binom{n}{j}}{\binom{m+n}{i+j}} B_{i+j}^{m+n}(t) \\
&= \sum_{k=0}^{m+n} R_k B_k^{m+n}(t), \qquad (3)
\end{aligned}
$$

where

$$
R_k = \sum_{\substack{i,\, j \\ i+j=k}} P_i Q_j \frac{\binom{m}{i}\binom{n}{j}}{\binom{m+n}{k}}.
$$

This result can also be extended to tensor product surfaces. It is also necessary to represent scalar products as part of representing sums and differences of rational curves and surfaces, as well as derivatives of rationals.

Finding a representation for the product of NURBs is far more difficult. A direct algorithmic approach has recently been developed [17] which supports symbolic computation of the coefficients of the product after finding the knot vector of the product curve. However, since it is computationally expensive and complex to implement, one might choose to exploit the B-spline representation uniqueness property and compute the coefficients of the product by solving an equivalent interpolation problem [14].

2.2 Contouring operator

It is frequently useful to know the zero set of a property surface or to have all regions in which the values of the property is larger than some threshold, either for itself or to use in further analysis. Contours in the parameter space of the property surface can be used as trimming curves for the original surface [15], so the trimmed surface will consist of all regions of the original surface with property values larger (or smaller) than the contouring level. The problem of computing the contours is closely related to finding surface-surface intersections and ray-surface intersections [18], problems with inherent numerical complexities and instabilities.

Let $F(u,v) = (\frac{x(u,v)}{w(u,v)}, \frac{y(u,v)}{w(u,v)}, \frac{z(u,v)}{w(u,v)})$ and $P = Ax + By + Cz + D = 0$ be the property surface and the contouring plane, respectively. By substituting the components of $F(u,v)$ into P one can solve for all values of u and v in the parametric domain for which $F(u,v) \cap P \neq \emptyset$.

$$\begin{aligned} S(u,v) &= A\frac{x(u,v)}{w(u,v)} + B\frac{y(u,v)}{w(u,v)} + C\frac{z(u,v)}{w(u,v)} + D \\ &= \frac{Ax(u,v) + By(u,v) + Cz(u,v) + Dw(u,v)}{w(u,v)} \end{aligned} \tag{4}$$

A single NURBs surface representation for equation 4 can be found using the operations defined in section 2.1, namely surface addition and surface multiplication. The zero set of the surface $S(u,v)$, in equation 4, is the set of parametric values for the required intersection. Since both $F(u,v)$ and $S(u,v)$ share the same parametric domain, mapping the parametric domain information back to $F(u,v)$ is trivial. $S(u,v)$ is a *scalar* surface, which leads to a simpler and faster computation. Assuming $w(u,v) \neq 0$, the zero set of $S(u,v)$ can be computed using only the numerator of $S(u,v)$. Thus, even if $F(u,v)$ is a rational surface, contouring computations can be performed on scalar polynomial surfaces.

In the following section, the tools defined in this section will be used. The four basic operations for surfaces: addition, subtraction, multiplication, and division will be combined with differentiation to define or approximate property surfaces, as necessary. Then the contouring algorithm will be used to analyze and extract useful information from them.

3 Examples

3.1 Surface slopes

The slope of a planar curve at a given point is equal to the angle between the tangent to the curve and a reference line, usually the horizontal axis. In an analogous way we define the *surface slope* at a given point, p, as the angle between the plane tangent to the surface at p and a reference plane. Without loss of generality, in the discussion below we assume that the reference plane is the xy plane.

Since the angle between two planes, is equal to the angle between their two normals, to compute surface slope, one need only compute the angle between the surface normal and the z axis. Let n be the surface unit normal and let n_z be its z component. Then, the tangent of the slope angle $\mathcal{P}$ is equal to:

$$tan(\mathcal{P}) = \frac{\sqrt{1 - n_z^2}}{n_z}. \tag{5}$$

When $n_z = +1$ the surface orientation is horizontal. If $n_z = 0$ the surface is vertical, and finally if $n_z = -1$ that surface is horizontal again, but this time facing down.

Inspection of the surface unit normal equation shows that $n(u,v)$ cannot be computed directly using the symbolic tools of section 2.1 because of the need to determine the square root. However, the z component of the unnormalized normal surface, $\hat{n}$, is equal to:

$$\hat{n}_z(u,v) = \frac{\partial x(u,v)}{\partial u}\frac{\partial y(u,v)}{\partial v} - \frac{\partial y(u,v)}{\partial u}\frac{\partial x(u,v)}{\partial v}, \tag{6}$$

where $x(u,v)$ and $y(u,v)$ are the x and y components of surface $S(u,v)$.

Then, $n_z(u,v) = \hat{n}_z(u,v)/\|\hat{n}(u,v)\|$, where $\|\hat{n}(u,v)\|$ is the magnitude of $\hat{n}(u,v)$

Even though $n_z(u,v)$ contains a square root factor, it is a scalar function, and can be squared so that $n_z(u,v)^2$ can be represented.

Given a slope $\mathcal{P}$ in degrees (or radians) the conversion to the $n_z^2(u,v)$ value required is straightforward using equation 5. Therefore, given a certain slope $\mathcal{P}$, one can compute the required n_z and n_z^2 using equation 5. Since n_z^2 is representable using (piecewise) rationals, one can contour this surface at the required n_z^2 level. Figure 2 demonstrates this exact process for several levels.

Alternatively, one can use the symbolically computed property $n_z^2(u,v)$ as a scalar map designating the color of the surface at each location, much like a texture map. Figure 3 is an example for this approach, for the same surface as in figure 2.

The technique presented here has also been used to compute silhouette curves of surfaces [12], and is equivalent to the zero set of equation 6. $\hat{n}_z(u,v)$ is symbolically computed and its intersection (contouring) with the plane $Z = 0$ provides the required silhouette curves in parametric space. Figure 4 shows one such example.

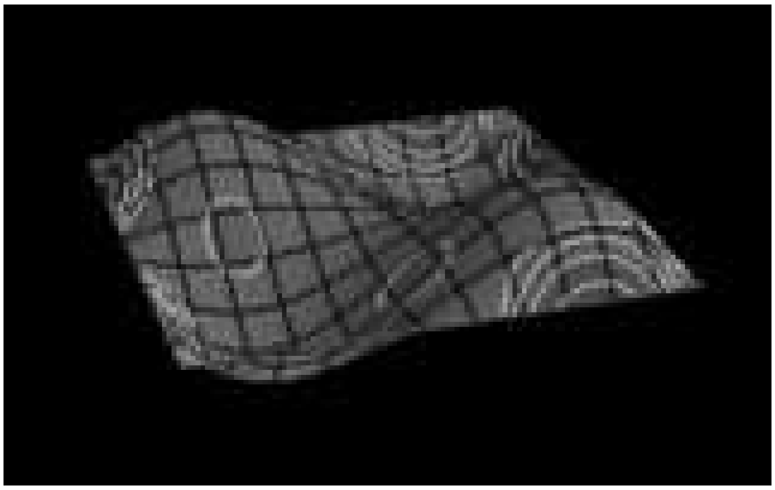

Figure 2: Different Slope or Steepness regions of the surface

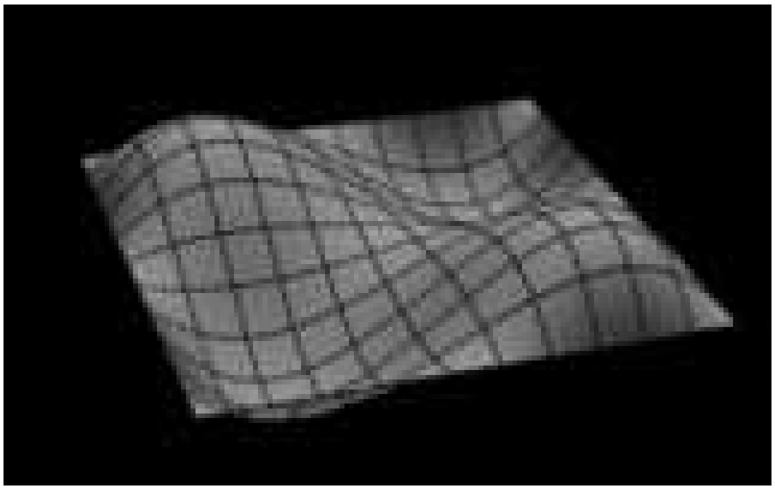

Figure 3: Continuous steepness of the surface in figure 2

Unlike curvature, slope is not an intrinsic surface property. In fact, since it is orientation dependent, it provides the designer with a measure on the planarity of the surface as well as on its orientation.

3.2 Surface Speed

The speed of a curve is defined as the distance moved in Euclidean space per unit of movement in parameter space. For a curve,

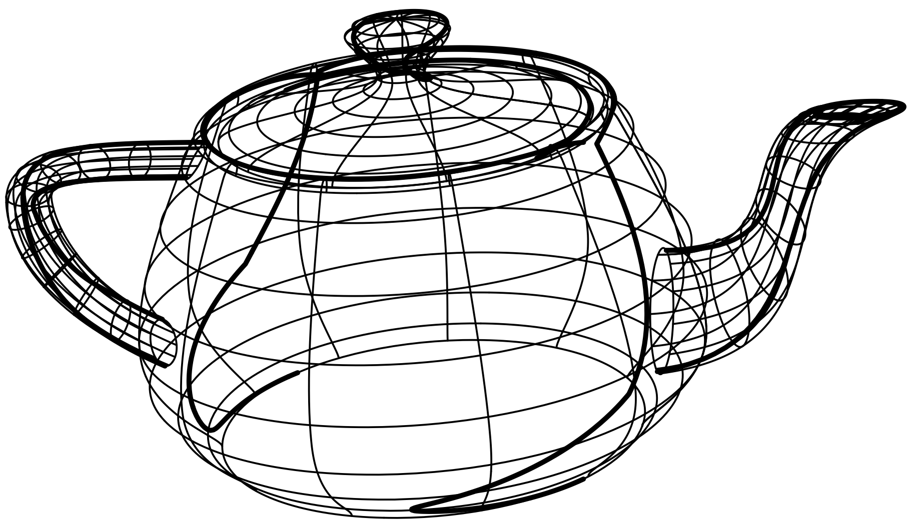

Figure 4: Silhouettes are equivalent to the zero set of equation 6 (rotated).

$$\begin{aligned}\mathcal{S}(t) &= \left\|\frac{dC(t)}{dt}\right\| \\ &= \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2}. \end{aligned} \tag{7}$$

We define the *speed bound* of surface $S(u, v)$ as the supremum of the speeds of all curves on the unit circle of the tangent plane using the first partials as a basis.

Let $\alpha(t)$ be a curve in the parametric domain of $S(u, v)$, that is $\alpha(t) = (u(t), v(t))$. By providing this speed bound of the surface parametrization, one can compute certain properties on $\alpha(t)$ and use the speed bound to extrapolate and provide bounds on the properties on the composed curve $S \circ \alpha = S(u(t), v(t))$.

Let $\gamma(t)$ be an auxiliary arc length parametrized curve with its image in the parametric space of $S(u, v)$, i.e. $\gamma(t) = (u(t), v(t))$, with $\sqrt{\left(\frac{du}{dt}\right)^2 + \left(\frac{dv}{dt}\right)^2} = 1$, for all t. Then

$$\begin{aligned}
& \left\| \frac{dS(u(t), v(t))}{dt} \right\|^2 \\
= & \left\| \frac{\partial S}{\partial u}\frac{du}{dt} + \frac{\partial S}{\partial v}\frac{dv}{dt} \right\|^2 \\
= & \left(\frac{\partial x}{\partial u}\frac{du}{dt} + \frac{\partial x}{\partial v}\frac{dv}{dt} \right)^2 + \left(\frac{\partial y}{\partial u}\frac{du}{dt} + \frac{\partial y}{\partial v}\frac{dv}{dt} \right)^2 + \left(\frac{\partial z}{\partial u}\frac{du}{dt} + \frac{\partial z}{\partial v}\frac{dv}{dt} \right)^2 \\
\leq & \left(\frac{\partial x}{\partial u} \right)^2 + \left(\frac{\partial x}{\partial v} \right)^2 + \left(\frac{\partial y}{\partial u} \right)^2 + \left(\frac{\partial y}{\partial v} \right)^2 + \left(\frac{\partial z}{\partial u} \right)^2 + \left(\frac{\partial z}{\partial v} \right)^2, \qquad (8)
\end{aligned}$$

since

$$\begin{aligned}
\left(\frac{\partial x}{\partial u}\frac{du}{dt} + \frac{\partial x}{\partial v}\frac{dv}{dt} \right)^2 &= \left(\left(\frac{\partial x}{\partial u}, \ \frac{\partial x}{\partial v} \right) \cdot \left(\frac{du}{dt}, \ \frac{dv}{dt} \right) \right)^2 \\
&= \left\| \left(\frac{\partial x}{\partial u}, \ \frac{\partial x}{\partial v} \right) \cdot \left(\frac{du}{dt}, \ \frac{dv}{dt} \right) \right\|^2 \\
&\leq \left\| \left(\frac{\partial x}{\partial u}, \ \frac{\partial x}{\partial v} \right) \right\|^2 \left\| \left(\frac{du}{dt}, \ \frac{dv}{dt} \right) \right\|^2 \\
&= \left\| \left(\frac{\partial x}{\partial u}, \ \frac{\partial x}{\partial v} \right) \right\|^2 \\
&= \left(\frac{\partial x}{\partial u} \right)^2 + \left(\frac{\partial x}{\partial v} \right)^2 \qquad (9)
\end{aligned}$$

If $\alpha \frac{\partial S}{\partial u} = \frac{\partial S}{\partial v}$ (see figure 5 with collinear partials along the surface boundary, which implies the surface is not regular there) and $\alpha \frac{du}{dt} = \frac{dv}{dt}$, then

$$\begin{aligned}
\left\| \left(\frac{\partial x}{\partial u}, \ \frac{\partial x}{\partial v} \right) \cdot \left(\frac{du}{dt}, \ \frac{dv}{dt} \right) \right\|^2 &= \left\| \left(\frac{\partial x}{\partial u}, \ \alpha \frac{\partial x}{\partial u} \right) \cdot \left(\frac{du}{dt}, \ \frac{dv}{dt} \right) \right\|^2 \\
&= \left\| \frac{\partial x}{\partial u}(1, \ \alpha) \left(\frac{du}{dt}, \ \frac{dv}{dt} \right) \right\|^2 \\
&= \left| \frac{\partial x}{\partial u} \right|^2 (1 + \alpha^2)
\end{aligned}$$

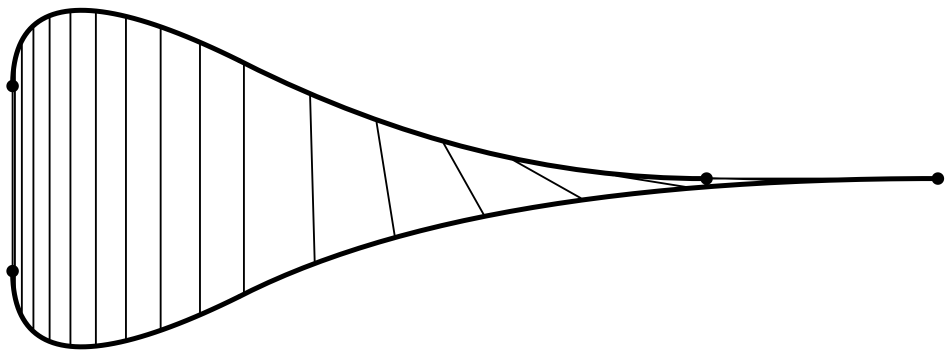

Figure 5: Degenerated boundary provides the two extremes on speed bound.

$$= \left(\frac{\partial x}{\partial u}\right)^2 + \left(\frac{\partial x}{\partial v}\right)^2, \tag{10}$$

and the upper bound established in equation 8 is reached. Therefore, this bound is minimal.

Since it is not possible to represent the square root of equation 8 as a (piecewise) rational surface, in general, we compute instead

$$\hat{\mathcal{S}}(u,v) = \left(\left(\frac{\partial x}{\partial u}\right)^2 + \left(\frac{\partial y}{\partial u}\right)^2 + \left(\frac{\partial z}{\partial u}\right)^2 + \left(\frac{\partial x}{\partial v}\right)^2 + \left(\frac{\partial y}{\partial v}\right)^2 + \left(\frac{\partial z}{\partial v}\right)^2\right). \tag{11}$$

Figures 6 and 7 are two examples of using $\hat{\mathcal{S}}(u,v)$ to compute a speed bound on the surface.

The speed surface can be used to provide a measure on the quality of the parametrization. This can becomes especially important if the surface is to be evaluated (for any purpose, including rendering) at a predefined set of parameter values.

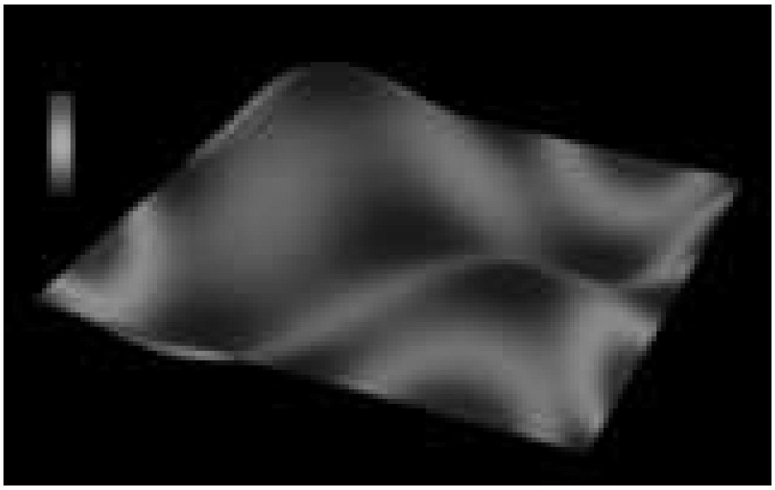

Figure 6: Parametrization speed estimate for a surface (same as figure 2).

Figure 7: Parametrization speed estimate for the teapot model.

3.3 Variations on Surface Twist

Also interesting is the ability to visualize surface twist. Basically, the twist is defined as the cross derivative component:

$$\mathcal{T}(u,v) = \frac{\partial^2 S(u,v)}{\partial u \partial v}. \tag{12}$$

This equation is representable and can always be computed symbolically for (piece-

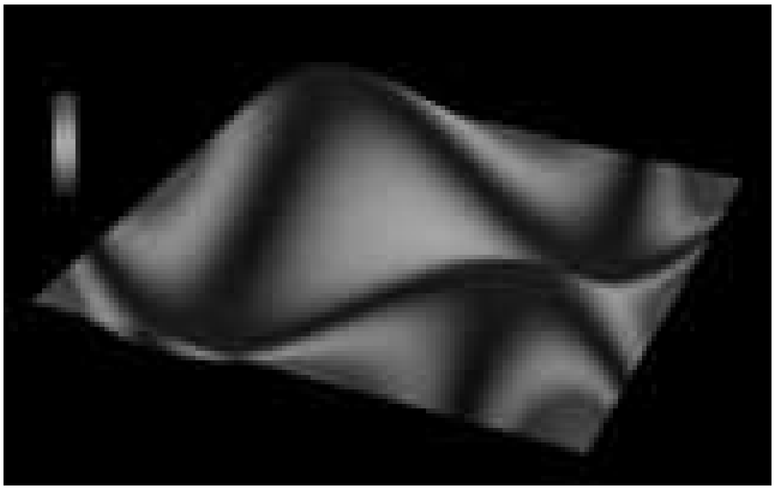

Figure 8: Twist component of a surface (same as figure 2).

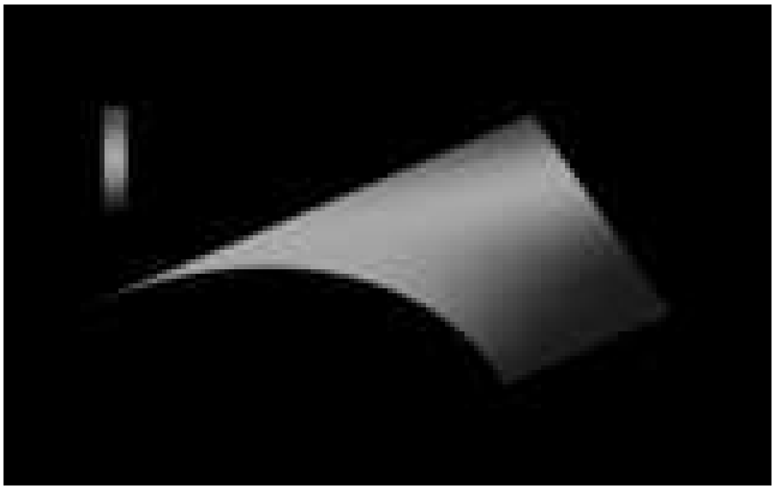

Figure 9: Twist component of a flat surface.

wise) rationals. Figures 8, 9 and 10 shows this property as a texture mapped on the surfaces.

Using equation 12 as a twist measure has a major drawback as can be seen in figure 9. Even though the surface is flat, the twist component is not zero since the speed of the parametrization is changing. In other words, the mapping from the parametric space to the Euclidean space is not isometric. It would be more helpful to use the twist component

Figure 10: Twist component of the teapot model.

in only the surface normal direction (see [1]) to eliminate the twist as a result of a non isometric mapping.

$$l_{12} = l_{21} = \left(n, \frac{\partial^2 S(u,v)}{\partial u \partial v} \right) \tag{13}$$

where l_{12}, and $= l_{21}$ are two of the components of second fundamental matrix form, L (see [4, 16, 19]).

Obviously, this time the l_{12} component in the flat surface in figure 9 is zero showing no twist in the normal direction. Furthermore, the use of this property showed that the teapot has virtually no twist in the normal direction as well. All the twist in figure 10 was a result of the nonisometric mapping. Figure 11 shows a nonplanar surface, similar to the one in figure 9 using l_{12} as property surface mapping colors onto the surface, as texture.

Since now one can compute both the total twist (equation 12), and the twist in the normal direction (equation 13), one can consider computing the twist in the tangent

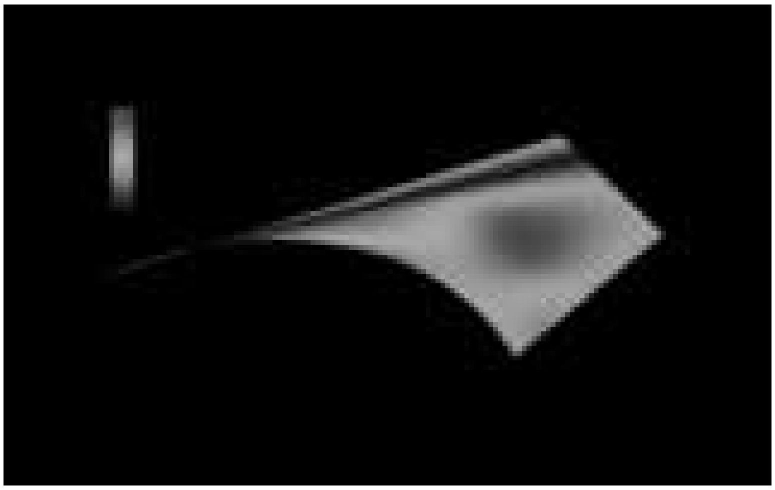

Figure 11: Twist component of a nonplanar twisted surface.

plane to the surface as the difference of the two quantities. This difference would provide another measure as to the quality of the surface parametrization.

3.4 Surface Trichotomy

It is frequently desired to provide a bound on the angularity of a surface. It is also desired in some cases to detect and isolate concave or convex regions. In 5-axis NC milling, a flat end cutter is usable only for the convex part of the surface.

In [4, 13] it is shown that one of the principal curvatures must be zero along the boundaries of convex, concave, or saddlelike regions and that this immediately necessitates that $\|L\| = 0$ where $\|L\|$ is the determinant of the second fundamental matrix form. It is also shown in [13] that the zero set of $\|\hat{L}\|$ can be used instead where

$$\hat{L} = (l_{ij}) = \begin{bmatrix} \left(\hat{n}, \frac{\partial^2 S}{\partial u^2}\right) & \left(\hat{n}, \frac{\partial^2 S}{\partial u \partial v}\right) \\ \left(\hat{n}, \frac{\partial^2 S}{\partial u \partial v}\right) & \left(\hat{n}, \frac{\partial^2 S}{\partial v^2}\right) \end{bmatrix}, \tag{14}$$

and $\hat{n}$ is the unnormalized normal $\hat{n}(u, v) = \frac{\partial S}{\partial u} \times \frac{\partial S}{\partial v}$ to the surface.

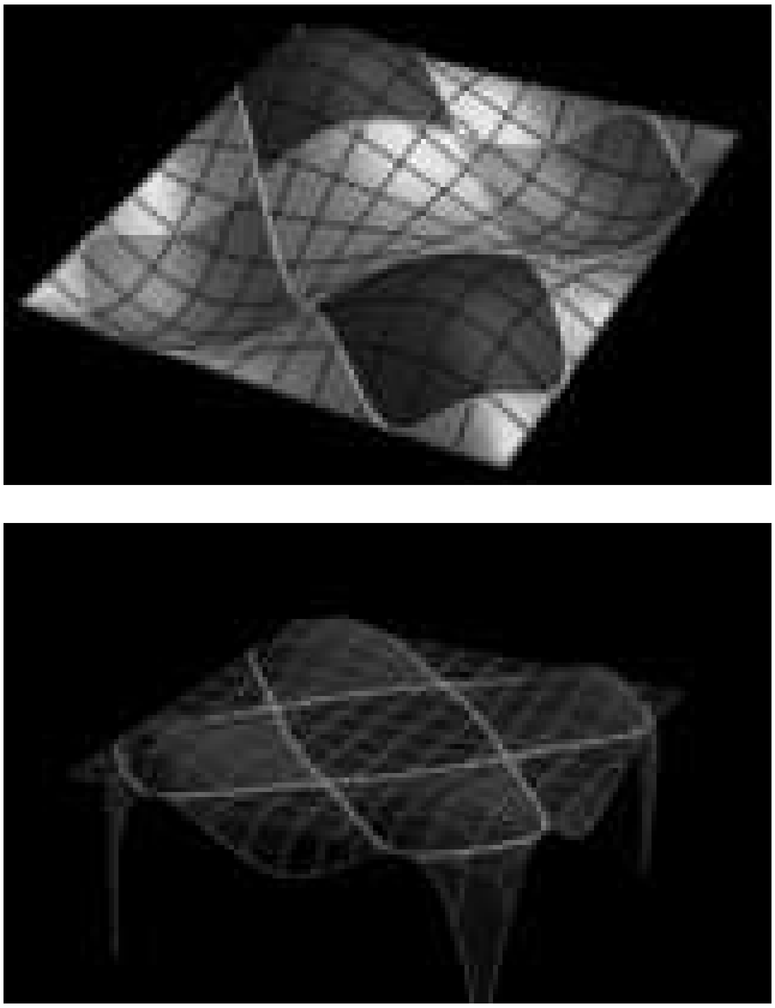

Figure 12: Bicubic surface with convex and concave regions meet at a single point (top). The surface second fundamental form property surface and its zero set (bottom). This surface is the same as in figure 2.

Each element of $\hat{L}$ is representable as a NURBs, using the tools developed in section 2. The bottom of figure 12 shows the scalar surface $\|\hat{L}\|$ with the zero plane and their intersection. The top of figure 12 uses these intersection curves to form the surface trichotomy into convex (red), concave (green), and saddle (yellow) trimmed regions. Figure 13 demonstrates this method on a more realistic object. The teapot trichotomy degenerates into a dichotomy since no concave regions exist in the teapot model.

Figure 13: Teapot trichotomy is degenerated into a ditochomy - no concave regions exist.

Finally it is interesting to note that a sufficient condition for a surface to be developable [10] is that its Gaussian curvature is zero everywhere, i.e. $K(u,v) \equiv 0$. Since $K(u,v) = \frac{\|L\|}{\|G\|}$, where G is the first fundamental form, this condition is equivalent to the condition that $\|L\| \equiv 0$, for regular surfaces when $\|G\| \neq 0$. A simple practical test that can answer whether a surface is developable or not is to symbolically compute and compare each of $\|L\|$ coefficients to zero. Figure 14 shows two developable NURBs surfaces, one ruled along an isoparametric direction while the other is not.

3.5 Bounding the Curvature

In [13] it is suggested that the sum of the squares of the principal curvatures may be a relevant measure of shape and can be represented as

$$\begin{aligned} \xi &= \left(\kappa_n^1\right)^2 + \left(\kappa_n^2\right)^2 \\ &= \frac{(g_{11}\hat{l}_{22} + \hat{l}_{11}g_{22} - 2g_{12}\hat{l}_{12})^2 - 2\|G\|\|\hat{L}\|}{\|G\|^2\|\hat{n}\|^2}, \end{aligned} \tag{15}$$

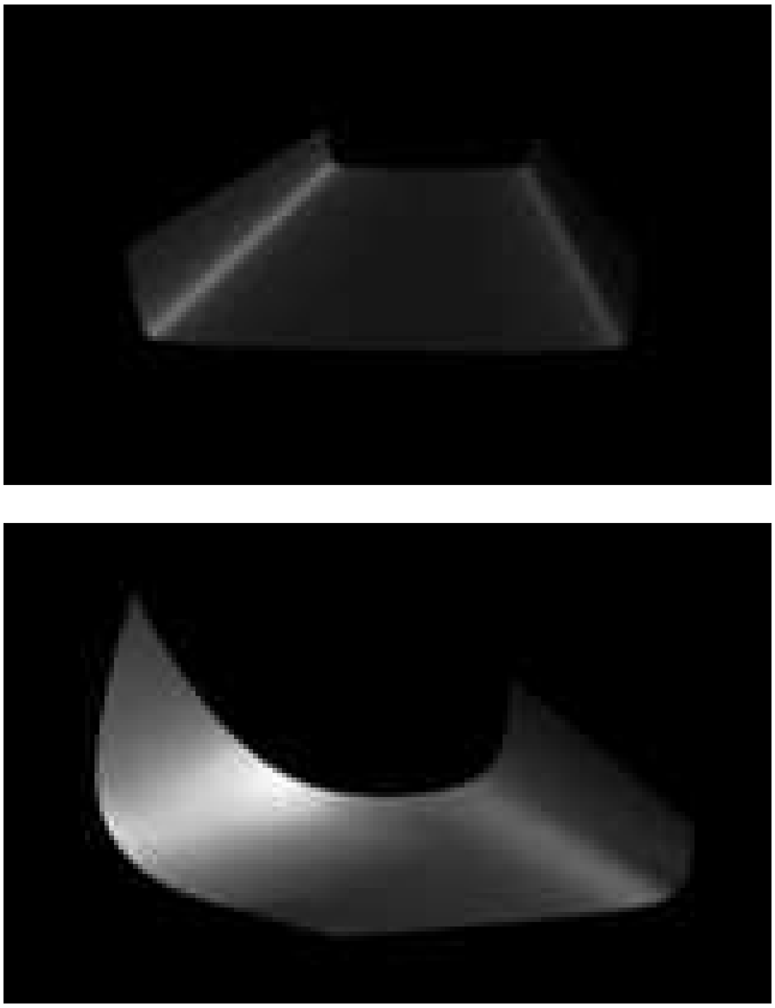

Figure 14: Two developable surfaces, the top is ruled along isoparametric direction and the bottom is not.

where G is the first fundamental matrix form [4, 16, 19].

ξ is bounded to be at most $\sqrt{2}$ larger than the larger absolute value of the two principal curvatures. Furthermore, ξ can be represented using the tools described in section 2. In figures 14 and 15 the ξ property has been computed for developable surfaces and for the teapot model respectively and used as a texture mapped through a color map table.

Figure 16 shows a surface subdivided into regions based on ξ. The property surface $\xi(u, v)$ of the surface in figure 16 is contoured in figure 17 and regions with different

Figure 15: Teapot curvature estimation using curvature property surface computation.

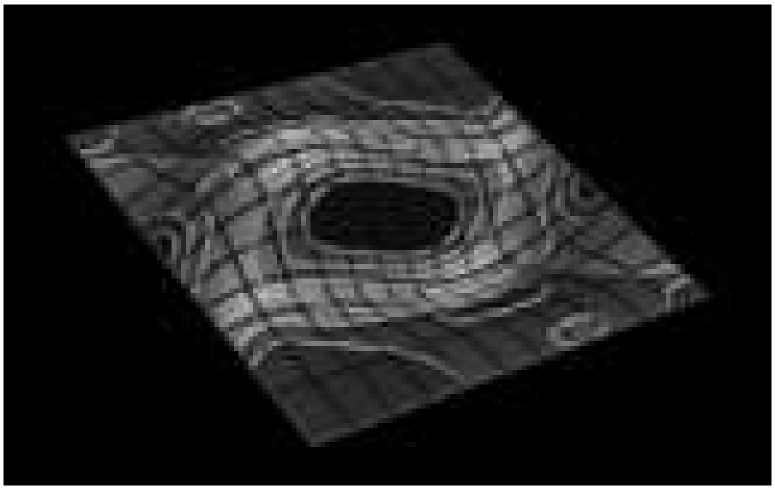

Figure 16: The surface is subdivided into regions with different curvature bounds.

curvature bounds are formed.

4 Conclusions

Surfaces derived from both first and second order analysis of sculptured surfaces are represented as NURBs surfaces using a small set of operators. We show that a combination of symbolic and numeric operators can be used to globally represent, approximate, or analyze these property surfaces. Other properties that cannot be represented as piece-

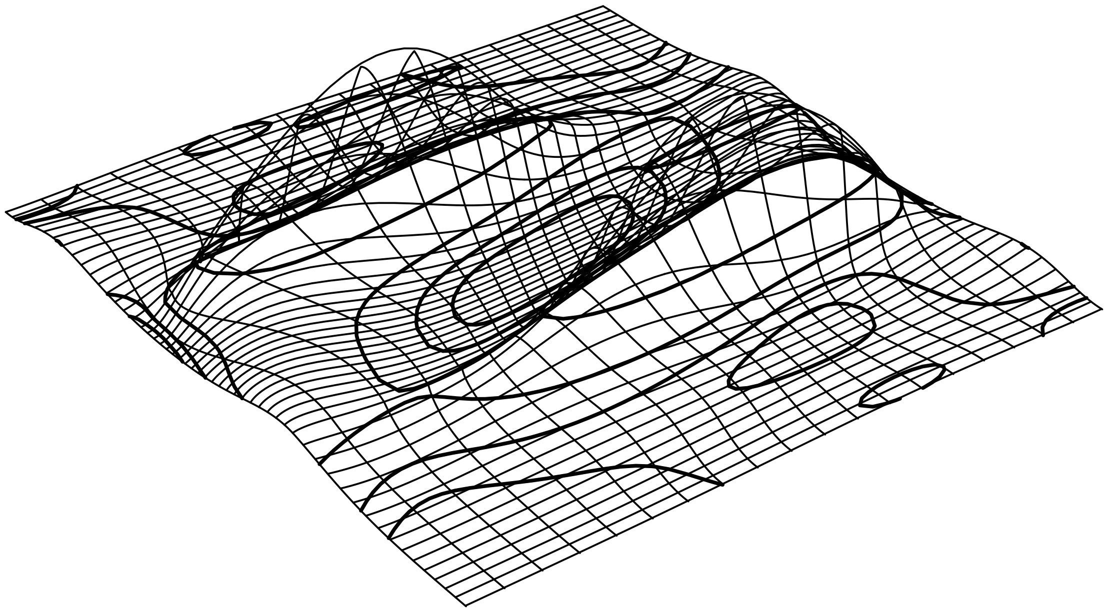

Figure 17: Curvature surface bound, ξ, of the surface in figure 16.

wise rationals have approximations that bound these properties and are representable. Further, we introduce two new derived surfaces to help visualizing and understanding surface shapes - speed and slope.

The full power of the NURBs representation can be used to analyze and to globally determine characteristics of these derived surfaces, which can then be used to visualize results or for feedback into design. For the first time the designer can guarantee that the steepness of the whole surface will be less than a specified slope or that a whole surface will have speed bound smaller than a specified value.

We show that symbolic computation supports robust computation and simplifies visualization of surface properties. Its usefulness is demonstrated in [14] for applications from error bound for offset approximation to adaptive and almost optimal toolpaths for machining purposes, as well as the surface analysis discussed in this paper.

References

[1] R. E. Barnhill, G. Farin, L. Fayard and H. Hagen. Twists, Curvatures and Surface Interrogation. Computer Aided Design, vol. 20, no. 6, pp 341-346, July/August 1988.

[2] J. M. Beck, R. T. Farouki, and J. K. Hinds. Surface Analysis Methods. IEEE Computer Graphics and Applications, Vol. 6, No. 12, pp 18-36, December 1986.

[3] J. C. Dill. An Application of Color Graphics to the Display of Surface Curvature. Siggraph 1981, pp 153-161.

[4] M. D. Carmo. Differential Geometry of Curves and Surfaces. Prentice-Hall 1976.

[5] E. Cohen, T. Lyche, and L. Schumaker. Degree Raising for Splines. Journal of Approximation Theory, Vol 46, Feb. 1986.

[6] E. Cohen, T. Lyche, and L. Schumaker. Algorithms for Degree Raising for Splines. ACM Transactions on Graphics, Vol 4, No 3, pp.171-181, July 1986.

[7] E. Cohen, T. Lyche, and R. Riesenfeld. Discrete B-splines and subdivision Techniques in Computer-Aided Geometric Design and Computer Graphics. Computer Graphics and Image Processing, 14, 87-111 (1980).

[8] G. Farin. Curves and Surfaces for Computer Aided Geometric Design. Academic Press, Inc. Second Edition 1990.

[9] R. T. Farouki and V. T. Rajan. Algorithms For Polynomials In Bernstein Form. Computer Aided Geometric Design 5, pp 1-26, 1988.

[10] I. D. Faux and M. J. Pratt. Computational Geometry for Design and Manufacturing.

[11] A. R. Forrest. On the Rendering of Surfaces. Siggraph 1979, pp 253-259.

[12] G. Elber and E. Cohen. Hidden Curve Removal for Free Form Surfaces. Siggraph 90, pp 95-104.

[13] G. Elber and E. Cohen. Second Order Surface Analysis Using Hybrid Symbolic and Numeric Operators. Submitted for Publication.

[14] G. Elber. Free Form Surface Analysis using a Hybrid of Symbolic and Numeric Computation. Ph.D. thesis, University of Utah, Computer Science Department, 1992.

[15] T. McCollough. Support for Trimmed Surfaces. M.S. thesis, University of Utah, Computer Science Department, 1988.

[16] Millman and Parker. Elements of Differential Geometry. Prentice Hill Inc., 1977.

[17] K. Morken. Some Identities for Products and Degree Raising of Splines. To appear in the journal of Constructive Approximation.

[18] J. T. Kajiya. Ray Tracing Parametric Patches. Siggraph 1982, pp 245-256.

[19] J. J. Stoker. Differential Geometry. Wiley-Interscience 1969.

[20] D. L. Schwitzer. Interactive Surface Visualization Using Raster Graphics. Ph.D. dissertation, University of Utah, August 1983.