z component. Then, the tangent of the slope angle $\mathcal{S}$ is equal to:

$$tan(\mathcal{S}) = \frac{\sqrt{1 - n_z^2}}{n_z}. \tag{6.5}$$

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 Chapter 2 because of the need to determine the square root. However, the z component of the unnormalized normal surface, $\hat{n}$, is equal to:

$$\hat{n}_z(u, v) = \frac{\partial x(u, v)}{\partial u}\frac{\partial y(u, v)}{\partial v} - \frac{\partial y(u, v)}{\partial u}\frac{\partial x(u, v)}{\partial v}, \tag{6.6}$$

where $x(u, v)$ and $y(u, v)$ are the x and y components of surface $S(u, v)$, and $n_z(u, v) = \hat{n}_z(u, v)/\|\hat{n}(u, v)\|$, 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 can be squared and $n_z(u, v)^2$ can be represented as a rational function.

Given a slope $\mathcal{S}$ in degrees (or radians), binding $n_z^2(u, v)$ is straightforward using equation (6.5). Therefore, given a certain slope $\mathcal{S}$, one can compute n_z and n_z^2 using equation (6.5). Because n_z^2 is representable using (piecewise) rationals, one can contour this surface to find the specified n_z^2 levels. Figures 6.6 and 6.7 demonstrate this exact process.

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 6.8 is an example for this approach, for the same surface as in Figure 6.7.

The technique presented here has also been used to compute silhouette curves of surfaces [22], and is equivalent to the zero set of equation (6.6). $\hat{n}_z(u, v)$ is symbolically computed and its intersection (contouring) with the plane $z = 0$ provides

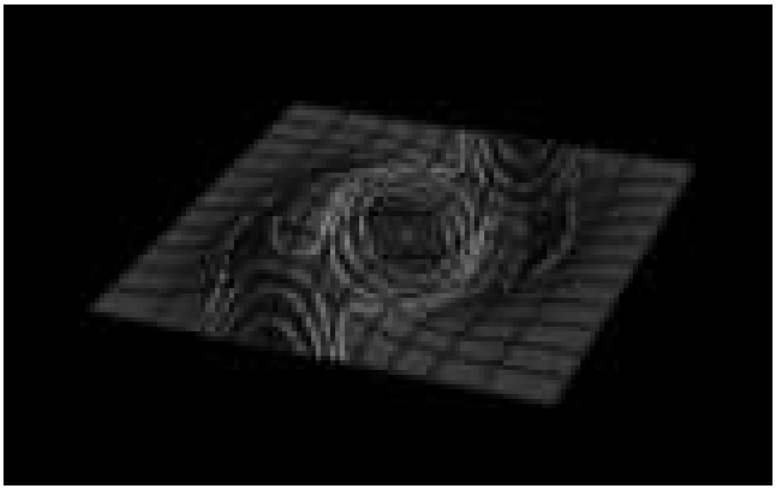

Figure 6.6. Different Steepness regions example

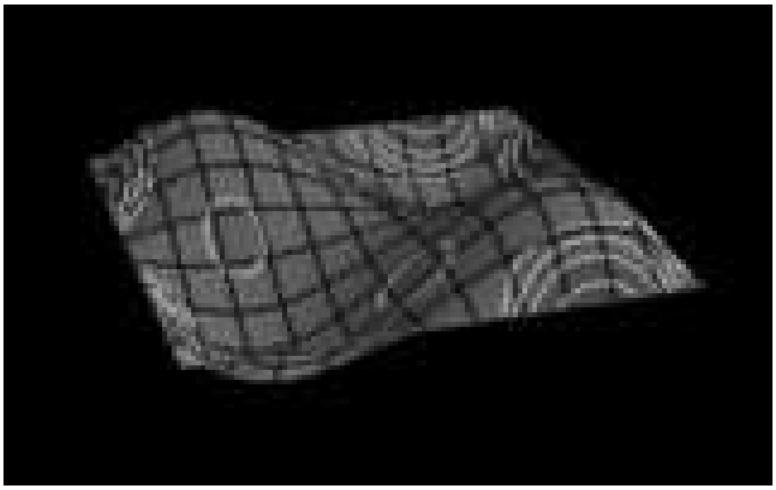

Figure 6.7. Different Slope or Steepness regions of the surface

the silhouette curves in parametric space for the specified speeds. Figure 6.9 shows one such example.

Unlike curvature, slope is not an intrinsic surface property. In fact, because it is orientation dependent, it provides the designer with a measure on the planarity of the surface in a specific orientation.

6.4 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,

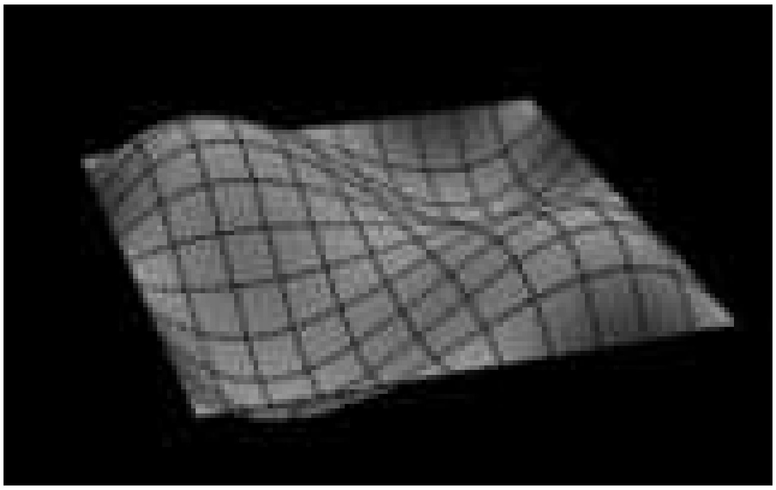

Figure 6.8. Continuous steepness of the surface in Figure 6.7

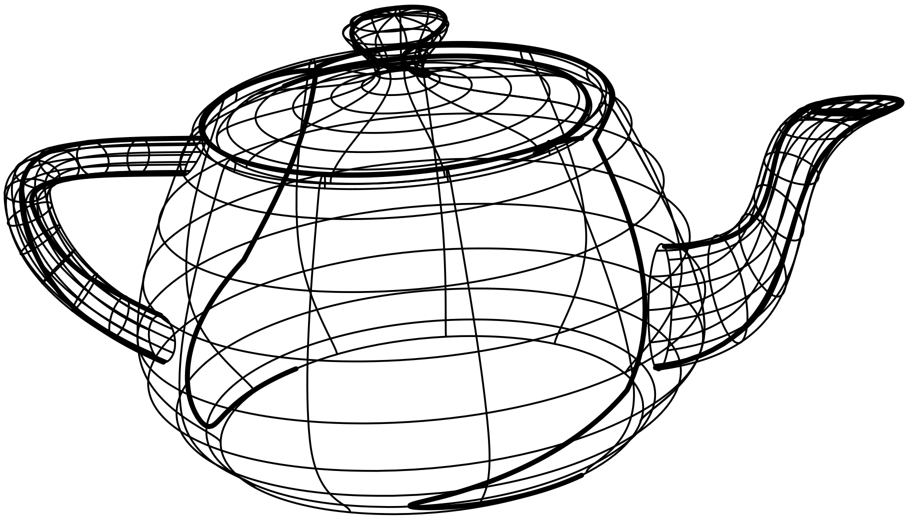

Figure 6.9. Silhouettes are equivalent to the zero set of equation (6.6) (rotated).

$$\begin{aligned}\mathcal{S}(t) &= \left\|\frac{dC(t)}{dt}\right\| \\ &= \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2}. \end{aligned} \tag{6.7}$$

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 $S(u, v)$, i.e., $\alpha(t) = (u(t), v(t))$. 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

$$\begin{aligned}
&\left\|\frac{dS(u(t), v(t))}{dt}\right\|^2 \\
= &\left\|\frac{\partial S}{\partial u}\frac{du}{dt} + \frac{\partial S}{\partial v}\frac{dv}{dt}\right\|^2 \\
= &\left(\frac{\partial x}{\partial u}\frac{du}{dt} + \frac{\partial x}{\partial v}\frac{dv}{dt}\right)^2 + \left(\frac{\partial y}{\partial u}\frac{du}{dt} + \frac{\partial y}{\partial v}\frac{dv}{dt}\right)^2 + \left(\frac{\partial z}{\partial u}\frac{du}{dt} + \frac{\partial z}{\partial v}\frac{dv}{dt}\right)^2 \\
\leq &\left(\frac{\partial x}{\partial u}\right)^2 + \left(\frac{\partial x}{\partial v}\right)^2 + \left(\frac{\partial y}{\partial u}\right)^2 + \left(\frac{\partial y}{\partial v}\right)^2 + \left(\frac{\partial z}{\partial u}\right)^2 + \left(\frac{\partial z}{\partial v}\right)^2,
\end{aligned} \tag{6.8}$$

because

$$\begin{aligned}
\left(\frac{\partial x}{\partial u}\frac{du}{dt} + \frac{\partial x}{\partial v}\frac{dv}{dt}\right)^2 &= \left(\left(\frac{\partial x}{\partial u}, \ \frac{\partial x}{\partial v}\right) \cdot \left(\frac{du}{dt}, \ \frac{dv}{dt}\right)\right)^2 \\
&= \left\|\left(\frac{\partial x}{\partial u}, \ \frac{\partial x}{\partial v}\right) \cdot \left(\frac{du}{dt}, \ \frac{dv}{dt}\right)\right\|^2 \\
&\leq \left\|\left(\frac{\partial x}{\partial u}, \ \frac{\partial x}{\partial v}\right)\right\|^2 \left\|\left(\frac{du}{dt}, \ \frac{dv}{dt}\right)\right\|^2 \\
&= \left\|\left(\frac{\partial x}{\partial u}, \ \frac{\partial x}{\partial v}\right)\right\|^2 \\
&= \left(\frac{\partial x}{\partial u}\right)^2 + \left(\frac{\partial x}{\partial v}\right)^2
\end{aligned} \tag{6.9}$$

If $\alpha\frac{\partial S}{\partial u} = \frac{\partial S}{\partial v}$ (see Figure 6.10 with collinear partials along the surface boundary, which implies the surface is not regular there) and $\alpha\frac{du}{dt} = \frac{dv}{dt}$, then

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

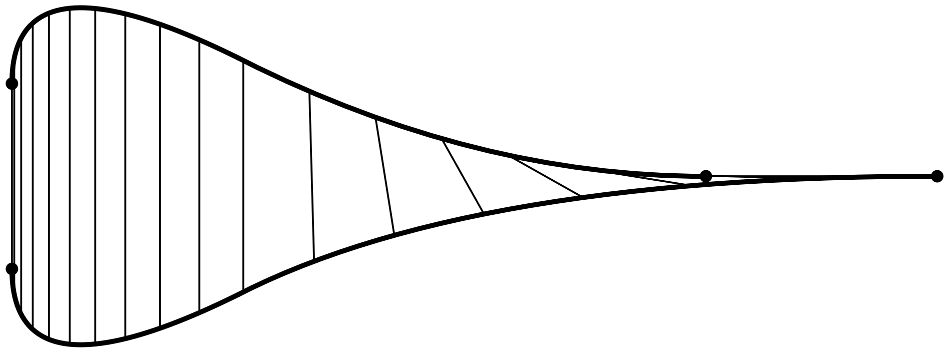

Figure 6.10. Degenerated boundary provides the two extremes on speed bound.

$$= \left(\frac{\partial x}{\partial u}\right)^2 + \left(\frac{\partial x}{\partial v}\right)^2, \qquad (6.10)$$

and the upper bound established in equation (6.8) is reached. Therefore this bound is minimal.

Because it is not possible to represent the square root of equation (6.8) as a (piecewise) rational surface, in general, we compute instead

$$\hat{\mathcal{S}}(u,v) = \left(\left(\frac{\partial x}{\partial u}\right)^2 + \left(\frac{\partial y}{\partial u}\right)^2 + \left(\frac{\partial z}{\partial u}\right)^2 + \left(\frac{\partial x}{\partial v}\right)^2 + \left(\frac{\partial y}{\partial v}\right)^2 + \left(\frac{\partial z}{\partial v}\right)^2\right) \qquad (6.11)$$

Figures 6.11 and 6.12 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.

6.5 Variations on Surface Twist

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

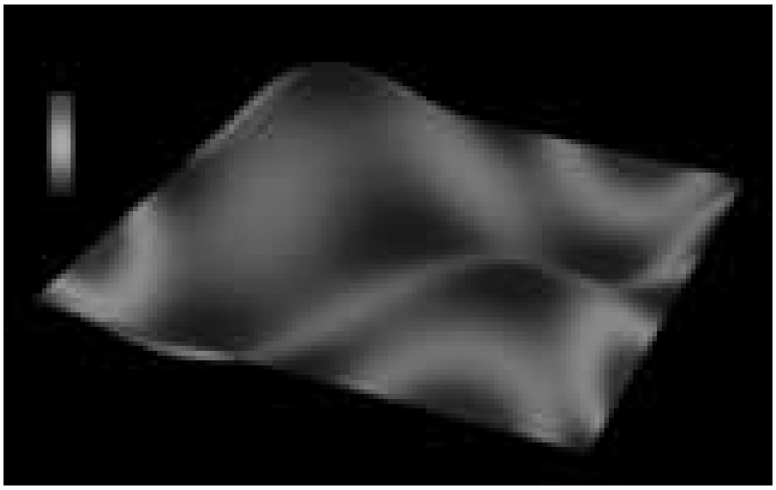

Figure 6.11. Parametrization speed estimate (same surface as Figure 6.7).

Figure 6.12. Parametrization speed estimate for the teapot model.

$$\mathcal{T}(u,v) = \frac{\partial^2 S(u,v)}{\partial u \partial v}. \tag{6.12}$$

This equation is representable and can always be computed symbolically for (piecewise) rationals. Figures 6.13, 6.14, and 6.15 shows this property as a texture mapped on the surfaces.

Using equation (6.12) as a twist measure has a major drawback as can be seen in Figure 6.14. Even though the surface is flat, the twist component is not zero because the speed of the parametrization is changing. In other words, the mapping

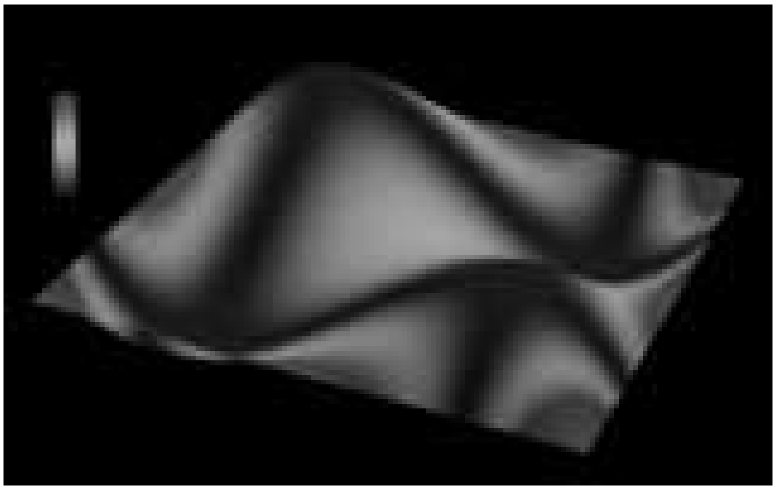

Figure 6.13. Twist component of a surface (same surface as Figure 6.7).

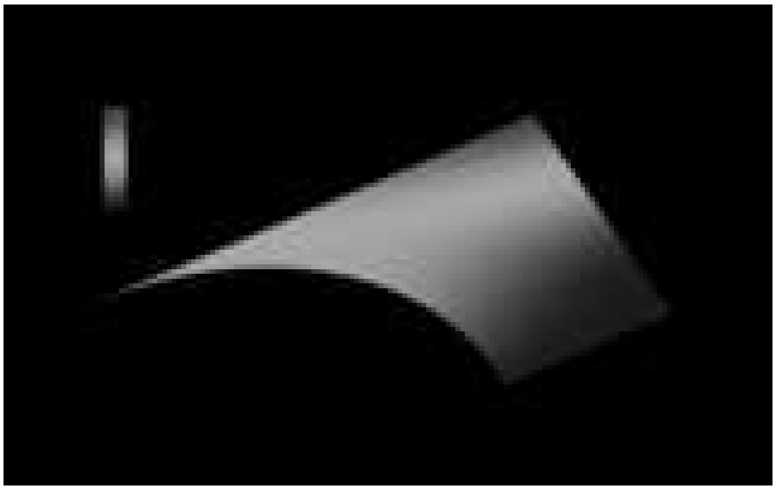

Figure 6.14. Twist component of a flat surface.

from the parametric space to the Euclidean space is not isometric. It would be more helpful to use the twist component in only the surface normal direction (see [3]) to eliminate the twist as a result of a nonisometric mapping.

$$l_{12} = l_{21} = \left(n, \frac{\partial^2 F}{\partial u \partial v} \right) \tag{6.13}$$

where l_{12}, and $= l_{21}$ are two of the components of second fundamental form, L (see Chapter 2).

Figure 6.15. Twist component of the teapot model.

Obviously, this time the l_{12} component in the flat surface in Figure 6.14 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 6.15 was a result of the nonisometric mapping. Figure 6.16 shows a nonplanar surface, similar to the one in Figure 6.14 using l_{12} as property surface mapping colors onto the surface, as texture.

Because now one can compute both the total twist (equation (6.12)), and the twist in the normal direction (equation (6.13)), one can consider computing the twist in the tangent 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.

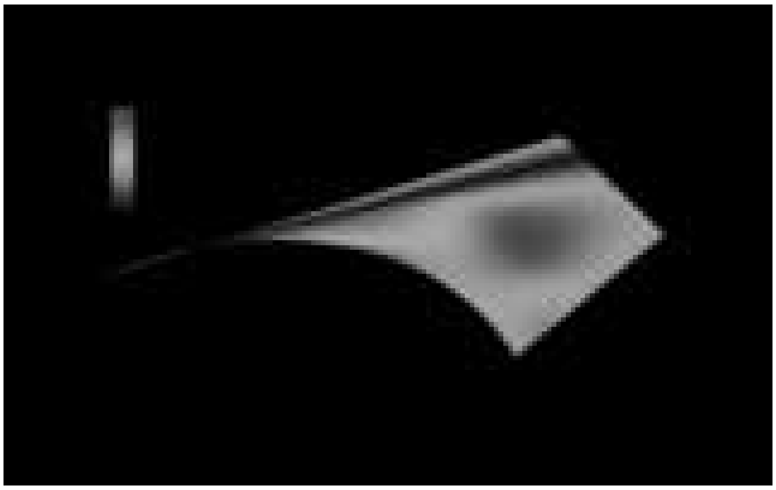

Figure 6.16. Twist component of a nonplanar twisted surface.

CHAPTER 7

CONCLUSIONS

Computers are useless; they can only give answers.

Picasso

It is our hope that symbolic computation will find its way as a useful tool in computer aided geometric design and in computer graphics. We hope that the symbolic approach developed throughout this work has demonstrated the capability and usefulness of this representation.

Several questions were left unanswered in this research. Detecting and isolating self-intersection in offsets of curves and surfaces is a difficult unsolved problem [51]. This thesis introduced a new robust method to isolate self intersections occurring in curve offsets. Extending the self-intersection isolation to surface offsets is difficult and is still a future research topic.

The work presented in Chapter 4 makes it practical to use second order surface analysis as a tool to support the development of robust, accurate, optimal algorithms for design and NC toolpath generation and to support alternative criteria for surface subdivision based on the second order properties of the shape. Consideration of Figures 4.3 and 4.5 shows another area of use. Users of NURBs are frequently unaware of the implications on the *shape* of the surface from using different orders. Manipulating the same control mesh can give different, unexpected, shapes depending on the order. The ability to accurately visualize second order properties in a reasonable time will enable better inspection and understanding of the effect of order, and potentially knot vector, changes. Furthermore, while NC verifications frequently simulate the tool path moving over the surface geometry, they do not check that a tool path for a convex region is actually cutting a convex region. The work presented here can be used in implementing that larger *visual process validation*. The viewer can use the understanding gained from exhibiting second order properties to take effective action.

The adaptive isocurve generation algorithm developed in Chapter 5 was extensively and successfully used for 3 axis machining. Adopting it to 4 or 5 axis toolpath generation is straightforward. However, this multi axis toolpath generation raises difficult questions regarding accessibility that must be addressed first. In 3 axis milling the accessibility problem is equivalent to an orthographic projection the hidden surface problem. "What you see is what you can mill". Although not simple, the hidden surface problem is well understood by the computer graphics community. Unfortunately, this does not work in 5 axis any more. The tool axis ("view direction") is not constant and in fact may vary as the tool moves. This area is under current research.

During the presentation of the new layout fabrication method in Chapter 5, it was implicitly assumed that the material thickness is negligible. Unfortunately, this is not always the case and compensating for the distortion that can result should be further investigated. In addition, extending this methodology to support stretching and tearing, should be investigated as well. Not only will that enable dealing with arbitrary surfaces (which cannot be decomposed into piecewise developable surfaces) but this algorithm may then support the ability to handle fabric and other anisotropic materials.

Chapter 6 introduces several small applications that can benefit from symbolic computation. The high order curve approximation using lower order Bézier curves method should be qualitatively and quantitatively compared to currently known approaches. Furthermore, it should be investigated whether a combined approach can yield an even better result.

The composition tool was also introduced in Chapter 6. In [42] the composition tools developed in this work were used to derive new and exact methods for fillet construction. The full potential of the composition operator combined with the symbolic tools derived in Chapter 2 should be further investigated.

Several shape measures, namely surface steepness, surface speed, and surface twist, where defined and shown to be computable and representable symbolically, in Chapter 6. The advantages these shape measures can provide the designer or the manufacturing engineer should be explored.

Undoubtedly, other applications in computer aided geometric design and in computer graphics can benefit from these tools. These fields matured enough to a level in which robustness is becoming an increasingly important issue. Symbolic computation is one such tool that can help alleviating the numerical problems we are facing today.

APPENDIX

CUSP EXISTENCE PROOF

This appendix shows that a cusp is formed in the offset curve $\mathcal{C}_d(t)$ any time the curve, $C(t)$, has curvature $\kappa(t)$ equal to $\frac{1}{d}$ where d is the offset distance and the mathematical curve normal $N(t)$, coincides with offset normal $N_o(t)$. Conditions for detecting curvature higher than $\frac{1}{d}$ are also derived.

Let $C(t)$ be a regular planar parametric curve that may not be arc length parameterized. Without loss of generality assume $C(t)$ is in the $x-y$ plane. Let $\mathcal{C}_d(t)$ be the offset curve of $C(t)$ by amount d. Let T, N and $\mathcal{T}$, $\mathcal{N}$ be their unit tangents and normals respectively. A nonunit length vector will be tagged with a hat, i.e., $\hat{T}$.

The tangent, T, of the planar curve, C, is equal to

$$\begin{aligned} T(t) &= \frac{\hat{T}(t)}{\|\hat{T}(t)\|} \\ &= \frac{(x'(t), y'(t))}{\sqrt{x'(t)^2 + y'(t)^2}}. \end{aligned} \tag{7.1}$$

From differential geometry theory [48, 63]:

$$\begin{aligned} \kappa(t)B(t) &= \frac{C'(t) \times C''(t)}{\|C'(t)\|^3} \\ &= \frac{(x'(t), y'(t), 0) \times (x''(t), y''(t), 0)}{\sqrt{x'(t)^2 + y'(t)^2}^3} \\ &= \frac{(0, 0, x'(t)y''(t) - y'(t)x''(t))}{\|\hat{T}\|^3} \end{aligned}$$

$$= \frac{(0, 0, \Psi)}{\|\hat{T}\|^3}. \tag{7.2}$$

Because $B_o(t)$ has been selected in $+z$ direction (see equations (3.1) and (3.2)), $N_o(t)$ is equal to

$$\begin{aligned} N_o(t) &= B_o(t) \times T(t) \\ &= \frac{(-y'(t), x'(t))}{\sqrt{x'(t)^2 + y'(t)^2}} \\ &= \frac{(-y'(t), x'(t))}{\|\hat{T}\|}. \end{aligned} \tag{7.3}$$

The offset curve $\mathcal{C}_d(t)$ of the planar curve $C(t)$ by amount d is defined as (equation (3.2)):

$$\begin{aligned} \mathcal{C}_d(t) &= C(t) + N_o(t)d \\ &= (x(t), y(t)) + \frac{(-y'(t), x'(t))}{\|\hat{T}\|}d \\ &= \frac{(x(t)\|\hat{T}\| - y'(t)d, y(t)\|\hat{T}\| + x'(t)d)}{\|\hat{T}\|}. \end{aligned} \tag{7.4}$$

The first derivative $\hat{\mathcal{T}}(t)$ of the offset curve $\mathcal{C}_d(t)$ is:

$$\begin{aligned} &\hat{\mathcal{T}}(t) \\ &= \mathcal{C}_d'(t) \\ &= \left(\frac{(x'(t)\|\hat{T}\| + x(t)\|\hat{T}\|' - y''(t)d)\|\hat{T}\| - (x(t)\|\hat{T}\| - y'(t)d)\|\hat{T}\|'}{\|\hat{T}\|^2}, \right. \\ &\quad \left. \frac{(y'(t)\|\hat{T}\| + y(t)\|\hat{T}\|' + x''(t)d)\|\hat{T}\| - (y(t)\|\hat{T}\| + x'(t)d)\|\hat{T}\|'}{\|\hat{T}\|^2} \right) \\ &= \left(\frac{x'(t)\|\hat{T}\|^2 - y''(t)\|\hat{T}\|d + y'(t)\|\hat{T}\|'d, y'(t)\|\hat{T}\|^2 + x''(t)\|\hat{T}\|d - x'(t)\|\hat{T}\|'d}{\|\hat{T}\|^2} \right). \end{aligned} \tag{7.5}$$

We are now ready to inspect the value of $\hat{\mathcal{T}}(t)$ in a case where d is equal to $\frac{1}{\kappa(t)}$. Using equation (7.2):

$$d = \frac{1}{\kappa(t)}$$

$$= \frac{\|\hat{T}\|^3}{\mid \Psi \mid}. \tag{7.6}$$

Substituting d in the x component of $\hat{\mathcal{T}}(t)$ we have:

$$\begin{aligned}
\hat{\mathcal{T}}_x(t) &= \frac{x'(t)\|\hat{T}\|^2 - y''(t)\|\hat{T}\|d + y'(t)\|\hat{T}\|'d}{\|\hat{T}\|^2} \\
&= x'(t) + \frac{-y''(t)\|\hat{T}\|^2 + y'(t)\|\hat{T}\|\|\hat{T}\|'}{\mid \Psi \mid} \\
&= x'(t) + \frac{-y''(t)x'(t)^2 - y''(t)y'(t)^2 + y'(t)x'(t)x''(t) + y'(t)^2y''(t)}{\mid \Psi \mid} \\
&= x'(t) + \frac{-y''(t)x'(t)^2 + y'(t)x'(t)x''(t)}{\mid \Psi \mid} \\
&= x'(t) + \frac{x'(t)(-y''(t)x'(t) + y'(t)x''(t))}{\mid \Psi \mid} \\
&= x'(t) - \frac{x'(t)\Psi}{\mid \Psi \mid} \\
&= \begin{cases} \equiv 0, & \Psi > 0 \\ = 2x'(t) & \Psi < 0 \end{cases} \qquad (7.7)
\end{aligned}$$

because

$$\begin{aligned}
\|\hat{T}\|\|\hat{T}\|' &= \sqrt{x'(t)^2 + y'(t)^2}\frac{1}{2\sqrt{x'(t)^2 + y'(t)^2}}(2x'(t)x''(t) + 2y'(t)y''(t)) \\
&= x'(t)x''(t) + y'(t)y''(t)
\end{aligned}$$

and

$$\|\hat{T}\|^2 = x'(t)^2 + y'(t)^2.$$

From equation 7.6, d may be substituted into the y component of $\hat{\mathcal{T}}(t)$, $\hat{\mathcal{T}}_y(t)$, in a similar way for the same result. Therefore, $\hat{\mathcal{T}}(t) \equiv 0$ in this situation or $\mathcal{C}(t)$ has a cusp if $\Psi = x'(t)y''(t) - x''(t)y'(t) > 0$ or the binormal $B(t)$ is positive and coincides with the definition of $B_o(t)$.

Moreover, if $d > \frac{1}{\kappa(t)}$, then the tangent vector $\hat{\mathcal{T}}$ flips direction as can be shown by its dot product with $\hat{T}$. Rewriting equation (7.5) as:

$$\hat{\mathcal{T}}(t) = (x'(t), y'(t)) + \frac{(-y''(t), x''(t))d}{\|\hat{T}\|} + \frac{(y'(t), -x'(t))\|\hat{T}\|'d}{\|\hat{T}\|^2}$$

and substituting it into

$$\begin{aligned}
&\left\langle \hat{\mathcal{T}}(t), \hat{T}(t) \right\rangle \\
&= \left\langle \left((x'(t), y'(t)) + \frac{(-y''(t), x''(t))d}{\|\hat{T}\|} + \frac{(y'(t), -x'(t))\|\hat{T}\|'d}{\|\hat{T}\|^2} \right), (x'(t), y'(t)) \right\rangle \\
&= (x'(t)^2 + y'(t)^2) + \frac{(-y''(t)x'(t) + x''(t)y'(t))d}{\|\hat{T}\|} \\
&= (x'(t)^2 + y'(t)^2) - \frac{\Psi d}{\|\hat{T}\|},
\end{aligned}$$

because the last term of $\hat{\mathcal{T}}(t)$ is perpendicular to $\hat{T}(t)$. Using equation (7.2):

$$\begin{aligned}
&\left\langle \hat{\mathcal{T}}(t), \hat{T}(t) \right\rangle \\
&= (x'(t)^2 + y'(t)^2) - \frac{\Psi d}{\|\hat{T}\|} \\
&= \begin{cases} (x'(t)^2 + y'(t)^2) - \frac{\kappa(t)(x'(t)^2+y'(t)^2)^{\frac{3}{2}} d}{\sqrt{x'(t)^2+y'(t)^2}} = (x'(t)^2 + y'(t)^2)(1 - \kappa(t)d), & \Psi > 0 \\ (x'(t)^2 + y'(t)^2) + \frac{\kappa(t)(x'(t)^2+y'(t)^2)^{\frac{3}{2}} d}{\sqrt{x'(t)^2+y'(t)^2}} = (x'(t)^2 + y'(t)^2)(1 + \kappa(t)d), & \Psi < 0. \end{cases}
\end{aligned}$$

Because $C(t)$ is a regular curve, $T(t)$ is never zero and $(x'(t)^2 + y'(t)^2)$ is positive everywhere. Therefore, for cases where the mathematical normal, $N(t)$, coincides with the offset normal, $N_o(t)$, or $\Psi > 0$, we get:

$$\begin{aligned}
sign(\left\langle \hat{\mathcal{T}}(t), \hat{T}(t)) \right\rangle &= sign((x'(t)^2 + y'(t)^2)(1 - \kappa(t)d)) \\
&= sign(1 - \kappa(t)d). \qquad (7.8)
\end{aligned}$$

Now for small $\kappa(t)$ or a relatively straight curve, $(1 - \kappa(t)d)$ is positive. When $\kappa(t)$ reaches $\frac{1}{d}$, the expression becomes zero, or $\hat{\mathcal{T}}(t) = 0$ because $\hat{T}(t)$ is never zero. If $\kappa(t)$ is larger than $\frac{1}{d}$, the expression is negative, that is $\hat{\mathcal{T}}(t)$ has flipped its direction.

If $\Psi < 0$ the expression is never zero because both d and $\kappa(t)$ are positive. This is not a surprising result because such offset only *increases* the radius of the osculating circle and hence can never make it vanish.

REFERENCES

[1] A. V. Aho, J. E. Hopcroft and J. D. Ullman. Data Structures and Algorithms. Addison-Wesley, June 1983.

[2] S. Aomura and T. Uehara. Self-Intersection of an Offset Surface. Computer Aided Design, vol. 22, no. 7, pp 417-422, september 1990

[3] RE 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.

[4] 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.

[5] C. Bennis, J. M. Vezien, and G. Iglesias. Piecewise Surface Flattening for Non-Distorted Texture Mapping. SIGGRAPH 1991.

[6] J. Bertrand. La Theorie Des Courbes a Double Courbure. Journal de Mathematique Pures et Appliquees, 15 (1850), 332-350.

[7] W. Boehm. Inserting New Knots into B-spline Curves. Computer Aided Design, vol. 12, no. 4, pp 199-201, July 1980.

[8] J. E. Bobrow. NC Machine Toll Path Generation From CSG Part Representations. Computer Aided Design, vol. 17, no. 2, pp 69-76, March 1985.

[9] M. S. Casale. Free-Form Solid Modeling with Trimmed Surface Patches. IEEE Computer Graphics and Applications, vol. 7, no. 1, pp 33-43, January 1987.

[10] B. K. Choi and C. S. Jun. Ball-End Cutter Interference Avoidance in NC Machining of Sculptured Surfaces. Computer Aided Design, vol. 21, no. 6, pp 371-378, July/August 1989.

[11] B. K. Choi and S. Y. Ju. Constant-Radius Blending in Surface Modelling. Computer Aided Design, vol. 21, no. 4, pp 213-220, May 1989.

[12] J. J. Chou. Numerical Control Milling Machine Toolpath Generation for Regions Bounded by Free Form Curves and Surfaces. Ph.D. Thesis, University of Utah, Computer Science Department, June 1989.

[13] B. Cobb. Design of Sculptured Surfaces Using The B-spline Representation. Ph.D. Thesis, University of Utah, Computer Science Department, June 1984.

[14] E. Cohen. Some Mathematical Tools for a Modeler's Workbench IEEE Computer Graphics and Applications, vol. 3, pp 63-66, October 1983.

[15] 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).

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

[17] 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.

[18] S. Coquillart. Computing Offset of Bspline Curves. Computer Aided Design, vol. 19, no. 6, pp 305-309, July/August 1987.

[19] S. Coquillart. Animated Free-Form Deformation: An Interactive Animation Technique. SIGGRAPH 1991, pp 23-26.

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

[21] M. P. DoCarmo. Differential Geometry of Curves and Surfaces. Prentice-Hall 1976.

[22] G. Elber and E. Cohen. Hidden Curve Removal for Untrimmed and Trimmed NURB Surfaces. University of Utah, Computer Science Dept., Technical Report UUCS-89-019, May 1989.

[23] G. Elber and E. Cohen. Hidden Curve Removal for Free Form Surfaces. SIGGRAPH 1990, pp 95-104.

[24] G. Elber and E. Cohen. Second Order Surface Analysis Using Hybrid Symbolic and Numeric Operators. To appear in Transaction on Graphics.

[25] G. Elber and E. Cohen. Error Bounded Variable Distance Offset Operator for Free Form Curves and Surfaces. International Journal of Computational Geometry and Applications, vol. 1., no. 1, pp 67-78, March 1991.

[26] G. Elber and E. Cohen. Adaptive Isocurves Based Rendering for Freeform Surfaces. Submitted for Publication.

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

[28] R. T. Farouki and C. A. Neff. Some Analytic and Algebraic Properties of Plane Offset Curves. Research Report 14364 (#64329) 1/25/89, IBM Research Division.

[29] R. T. Farouki. The Approximation of Non-Degenerate Offset Surfaces Computer Aided Geometric Design, vol. 3, no. 1, pp 15-43, May 1986.

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

[31] R. T. Farouki and V. T. Rajan. On The Numerical Condition of Polynomial in Bernstein Form. Computer Aided Geometric Design 4, pp 191-216, 1987.

[32] I. D. Faux and M. J. Pratt. Computational Geometry for Design and Manufacturing. John Wiley & Sons, 1979.

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

[34] C. M. Hoffmann. Geometric & Solid Modeling, an Introduction. Morgan Kaufmann Publisher, Inc..

[35] J. Hoschek. Offset Curves in the Plane. Computer Aided Design, vol. 17, no. 2, pp 77-82, March 1985.

[36] J. Hoschek. Spline Approximation of Offset Curves. Computer Aided Geometric Design 5, pp 33-40, 1988.

[37] J. Hoschek and N. Wissel. Optimal Approximate Conversion of Spline Curves and Spline Approximation of Offset Curves. Computer Aided Design, vol. 20, no. 8, pp 475-483, October 1988.

[38] J. Hoschek. Approximate Conversion of Spline Curves. Computer Aided Geometric Design 4, pp 59-66, 1987.

[39] J. Hoschek, F. J. Schneider, and P. Wassum. Optmial Approximate Conversion of Spline Surfaces. Computer Aided Geometric Design 6, pp 293-306, 1989.

[40] R. B. Jerard, J. M. Angleton and R. L. Drysdale. Sculptured Surface Tool path Generation with Global Interference Checking. Design Productivity Conference, Feb. 6-8, 1991, Honolulu, Hawaii.

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

[42] K. Kim. Blending Parametric Surfaces. M.S. Thesis, University of Utah, Computer Science Department, August 1992.

[43] J. Lane and R. Riesenfeld. A Theoretical Development for the Computer Generation and Display of Piecewise Polynomial Surfaces. IEEE Transaction on pattern analysis and machine intelligence, vol. PAMI-2, no. 1, January 1980.

[44] J. Lane and R. Riesenfeld. Bounds on a Polynomial BIT 21, pp 112-117, 1981.

[45] R. T. Lee and D. A. Fredericks. Intersection of Parametric Surfaces and a Plane. IEEE Computer Graphics and Applications, vol. 4, no. 8, pp 48-51, August 1984.

[46] G. C. Loney and T. M. Ozsoy. NC Machining of Free Form Surfaces. Computer Aided Design, vol. 19, no. 2, pp 85-90, March 1987.

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

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

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

[50] B. Pham. Offset Approximation of Uniform B-splines. Computer Aided Design, vol. 20, no. 8, pp 471-474, October 1988.

[51] B. Pham. Offset Curves and Surfaces: a Brief Survey. Computer Aided Design, vol. 24, no. 4, pp 223-229, April 1992.

[52] H Persson. NC Machining of Arbitrarily Shaped Pockets. Computer Aided Design, vol. 10, no. 3, pp 169-174, May 1978.

[53] PostScript Language Reference Manual. Adobe Systems Incorporated. Addison Wesley Publishing Company, Inc., 1987.

[54] A. A. G. Requicha. Toward a Theory of Geometric Tolerancing. International Journal of Robotics Research, vol. 2, no. 4, pp 45-49, Winter 1983.

[55] R. Riesenfeld. Applications of B-spline Approximation to Geometric Problems of Computer-Aided Design. Ph.D. Thesis. University of Utah, Technical Report UTEC-CSc-73-126, March 1973.

[56] D. F. Rogers and J. A. Adams. Mathematical Elements for Computer Graphics, second edition. McGraw-Hill Publishing Company, 1990.

[57] J. R. Rossignac and A. A. G. Requicha. Constant Radius blending In Solid Modeling. Computers in Mechanical Engineering, pp 65-73, July 1984.

[58] F. Salkowski. Zur Transformation von Raumkurven. Mathematische Annalen, 66 (1909), 517-557.

[59] S. G. Satterfield and D. F. Rogers. A Procedure for Generating Contour Lines From a Bspline Surface. IEEE Computer Graphics and Applications, vol. 5, no. 4, pp 71-75, April 1985.

[60] T. W. Sederberg and S. Parry. Comparison of Three Curve Intersection Algorithms. Computer Aided Design, vol. 18, no. 1, pp 58-63, January/February 1986.

[61] T. W. Sederberg and S. Parry. Free-Form Deformation of Solid Geometric Models. SIGGRAPH 1986, pp 151-160.

[62] T. W. Sederberg and A. K. Zundel. Scan Line Display of Algebraic Surfaces. SIGGRAPH 1989, pp 147-156.

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

[64] S. W. Thomas. Scanline Rendering for 3-Axis NC Toolpath Generation, Simulation and Verification. Dept. of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109-2122, Technical Report CSE-TR-43-90, January 1990.

[65] A. Voss. Uber Kurvenpaare in Raume. Sitzungsberichte, Akadamie der Wissenschaften zu Munchen. 39 (1909), 106.

[66] D. J. Walton abd D. S. Meek. Curvature Bounds For Planar B-spline Curve Segments. Computer Aided Design, vol. 20, no. 3, pp 146-150, April 1988.

[67] D. Zhang and A. Bowyer. CSG Set-Theoretical Solid Modelling and NC Machining of Blend Surfaces. The Second Computation Geometry Conference, ACM 1986.