Global Error Bounds and Amelioration of Sweep Surfaces *

Gershon Elber
Department of Computer Science
Technion, Israel Institute of Technology
Haifa 32000, Israel

June 15, 2001

Abstract

With the prescription of a (piecewise) polynomial or rational cross section and axis curves, contemporary sweep or generalized cylinder constructors are incapable of creating an exact or even an approximation with a bounded error of the actual sweep surface that is represented in the same functional space, in general. An approach is presented to bound the maximal error of the sweep approximation. This bound is automatically exploited to adaptively refine and improve the sweep approximation to match a prescribed tolerance. Finally, methods are considered to eliminate the self intersecting regions in the sweep surface resulting from an axis curve with large curvature.

Keywords: Freeform Surfaces, Sweep Approximation, Self Intersection Elimination.

1 Introduction

The sweep operation is a general freeform surface construction tool that can be found in virtually any solid modeling system. A sweep surface, also known as a generalized cylinder, is defined as the envelope formed by the transformation of a prescribed cross section curve along a second curve called the axis curve. Typically, this transformation is affine. As a powerful constructor, the sweep operation encompass various other freeform surface constructors as special cases. The extrusion is a sweep along a straight line. The surface of revolution is a sweep along a circular curve. It is unfortunate that sweep surfaces cannot be exactly represented in the space of NURB or piecewise rational, in general. Closely associated with offsets of freeform curves and surfaces [1], the inability to represent the exact sweep or offset surfaces lays in the necessity

*This work was supported in part by grant No. 92-00223 from the United States-Israel Binational Science Foundation (BSF), Jerusalem, Israel. However, opinions, conclusions or recommendations arising out of supported research activities are those of the author or the grantee and should not be presented as implying that they are the views of the BSF.

to represent square roots in unit vector fields. Various methods were derived [1, 2, 3, 4, 5] to compute approximations of sweep surfaces for solid modeling systems that support (piecewise) polynomial and rational parametric curves and surfaces. Others, circumventing the inability to represent the exact sweep surface as a (piecewise) polynomial or rational, have provided an implicit procedural representation [6, 7].

The sweep surface $S(u, v)$ can be defined as,

$$S(u, v) = A(v) + \mathcal{R}(v)[r(v)C(u)], \tag{1}$$

where $\mathcal{R}(v)$ is the rotational orientation frame that transforms the cross section curve, $C(u)$, according to the axis curve, $A(v)$, that is also known as trajectory or skeleton curve (See Figure 1). $r(v)$ is an optional scaling function, where the case of $r(v) \equiv 1$ is quite common, the case of constant scale sweep. $r(v)$ is also known as a profiling curve [1]. Let $\mathcal{F}$ be the space of (piecewise) polynomial or rational functions. Hereafter, and unless stated otherwise, assume $C(u)$, $A(v)$, and $r(v) \in \mathcal{F}$. Exact representation in $\mathcal{F}$ of the sweep surface exists if the axis curve $A(v)$ is linear or circular, as is noted in [1].

Linear and circular sweep operations were used to simulate the volume that is traced by a machining tool, in multi-axis motion. The swept volumes are then subtracted from the stock, simulating and verifying the milling process. A two dimensional piecewise linear approximation of a sweep operation that provides an upper bound on the error can be found in [8]. In [1], the sweep is approximated as a set of offset curves of $A(v)$ in space.

In the methods of [1, 2, 3, 4, 5], the resulting surface is only an approximation of the real sweep, in general, and no real error bound is established. In [2], the axis or trajectory curve is approximated using a piecewise linear representation that is interpolated in subsequent stages into the freeform sweep surface approximation, hindering any attempt to provide an error bound estimate. In [4], a sweep surface is constructed by interpolating a discrete set of position and orientation specifications. While the orientation is represented using unit quaternions, the interpolated shape is represented as a rational degree 6 Bézier achieving C^2 continuity. An error estimate for the sweep of two dimensional curved objects can be found in [8] with a result that is also a piecewise linear approximation. The two dimensional self intersection problem of

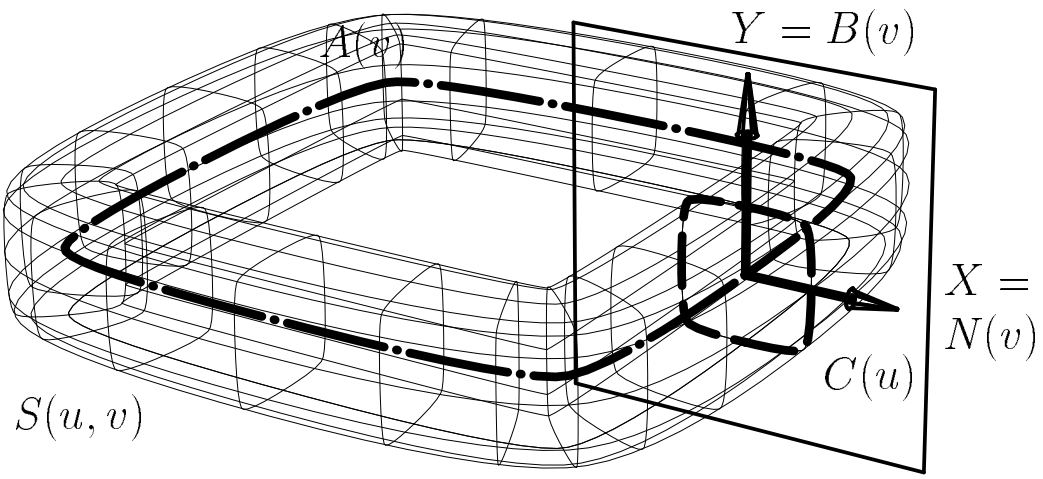

Figure 1: A sweep surface $S(u, v)$ is defined as the volume that the cross section $C(u)$ (dashed lines) traverses through while moving along the axis curve $A(v)$ (dotted-dashed lines).

the resulting shape is also considered in [8].

Many sweep surface constructors exploit the Frenet frame [9] of the axis curve to orient the cross section,

$$\begin{aligned} T(v) &= \frac{\frac{dA(v)}{dv}}{\left\| \frac{dA(v)}{dv} \right\|}, \\ \kappa(v)B(v) &= \frac{\frac{dA(v)}{dv} \times \frac{d^2A(v)}{dv^2}}{\left\| \frac{dA(v)}{dv} \right\|^3}, \\ N(v) &= B(v) \times T(v), \end{aligned} \tag{2}$$

where $T(v)$, $N(v)$, $B(v)$, and $\kappa(v)$ are the tangent, normal, binormal and curvature fields of curve $A(v)$, respectively. It can be easily verified that the trihedron T, N, B form an orthogonal basis.

The sweep surface is typically constructed by associating $N(v)$ and $B(v)$ with the X and Y axes of the plane containing the cross section, $C(u)$ (see Figure 1), and reorienting the plane containing $C(u)$ according to $N(v)$ and $B(v)$ as we move along $A(v)$. While common, $C(u)$ need not be planar, and the Z coefficient of $C(u)$ may be associated with $T(v)$.

In [2], the Frenet frame is employed, while special care is taken for cases where the curvature of the axis curve is vanishing. In [1, 5] and due to the unstable behavior of the Frenet frame near inflection points, modified Frenet frame schemes that minimize the axis curve's twist motion, are introduced. In [1, 10], a rotation minimizing frame that is stable around inflection points

is defined. In [10], a discussion that compares the Frenet frame to the rotation minimizing frame is presented. In [5], a projection normal orientation is described that produces the next axis curve's pseudo normal using a Gramm-Schmidt process, $N_i = N_{i-1} - \langle N_{i-1}, T_i \rangle T_i$, where $\langle \cdot, \cdot \rangle$ denotes the inner product. This projection is similar to the rotation minimizing frame presented in [1].

Unfortunately, given $C(u)$, $A(v)$, $r(v) \in \mathcal{F}$, $\mathcal{R}(v)$ in Equations (1) can only be approximated in $\mathcal{F}$, in general. All sweep construction methods of [1, 2, 3, 4, 5] provide means to neither estimate nor bound the error of the resulting approximated sweep surface in $\mathcal{F}$. None of the vector fields of the Frenet frame in Equation (2) is representable in $\mathcal{F}$, in general, even when $A(v) \in \mathcal{F}$. Thus, sweep surfaces are approximated. In this paper, we characterize the error of the approximation of the sweep operator. Moreover, we employ this error upper bound to automatically determine the regions in $A(v)$ that needs further refinement in order to improve the quality of the approximation of the sweep surface. We exploit a NURBs based representation to construct a sweep operator that automatically guarantee a global upper bound on the maximal error of the approximation.

This paper is organized as follows. Section 2 presents computational considerations on the error bound and the sweep surface amelioration algorithm. Section 3 shows several examples of the algorithm's use and discusses the convergence rate of the algorithm. In Section 4, we consider another difficult problem that must be dealt with by a sweep surface constructor. A sweep surface can self intersect, if the axis curve has regions with a sufficiently large curvature, an intersection we denote local. In Section 4, we develop techniques to detect and eliminate these local self intersections from the resulting representation of the sweep surface. Finally, in Section 5, we conclude and raise some related open questions for future research. All examples in this paper were created with the aid of the Alpha_1 NURBs based solid modeling system, that is being developed at the University of Utah.

2 Algorithm

Let $S(u, v)$ be a sweep surface as in Equation (1). The distance between $S(u, v)$ and $A(v)$, $\|S(u, v) - A(v)\|$, is equal to $\|C(u)\| r(v)$ if the sweep representation is exact. Thus, we charac-

terize an error norm of,

$$\epsilon(u,v) \quad = \quad \|S(u,v) - A(v)\| - \|C(u)\|r(v). \tag{3}$$

Unfortunately, the result of Equation (3) cannot be represented in $\mathcal{F}$ due to the square root terms, in general. Nonetheless, using symbolic computation over the (piecewise) rationals, one can represent in the space of $\mathcal{F}$ the modified error bound of,

$$\begin{aligned}
\sigma(u,v) &= \|S(u,v) - A(v)\|^2 - \|C(u)\|^2 r^2(v) \\
&= \Big(\|S(u,v) - A(v)\| - \|C(u)\|r(v)\Big)\Big(\|S(u,v) - A(v)\| + \|C(u)\|r(v)\Big) \\
&= \epsilon(u,v)\Big(\epsilon(u,v) + 2\|C(u)\|r(v)\Big),
\end{aligned} \tag{4}$$

in a similar approach to that taken in [11].

Hence, by computing $\sigma(u,v)$, one can bound the global error $\epsilon(u,v)$ from above up to a factor of $2\|C(u)\|r(v) = 2\sqrt{\langle C(u), C(u)\rangle}r(v)$, a factor that depends on the shape of $C(u)$ and the profile, $r(v)$. Prescribe an upper bound tolerance τ such that $\epsilon(u,v) < \tau$ must hold. Then,

$$\begin{aligned}
\sigma(u,v) &\underset{\epsilon \to 0}{=} 2\epsilon(u,v)\|C(u)\|r(v) \\
&< 2\tau\|C(u)\|r(v).
\end{aligned} \tag{5}$$

Assume $r(v) \equiv 1$. Then, satisfying,

$$\sigma(u,v) < 2\tau\underline{C}, \tag{6}$$

where $\underline{C} = \inf_u \sqrt{\langle C(u), C(u)\rangle}$ will guarantee the satisfaction of Equation (5). Provided that $\sigma(u,v)$ is less than $2\tau\underline{C}$, in no place on $S(u,v)$ will $S(u,v)$ be more than τ away from the exact sweep surface.

For arbitrary $r(v)$, one gets,

$$\sigma(u,v) < 2\tau\underline{C}\,\underline{R}, \tag{7}$$

where $\underline{R} = \inf_v r(v)$ and $\underline{C}$ is as in Equation (6), will guarantee the satisfaction of Equation (7). Alternatively and since $r(v)$ is a scalar curve, one can establish a tighter bound that considers the affect that $r(v)$ has on the error, satisfying,

$$\mu(u,v) = \sigma(u,v) - 2\tau\underline{C}r(v) < 0. \tag{8}$$

Equations (6), (7) and (8) not only characterize the error that is introduced by the sweep operator but they are all computable and representable as scalar surface fields in $\mathcal{F}$. Equations (6), (7) and (8) exploit the lower bound of cross section curve $\underline{C} = \inf_u \sqrt{\langle C(u), C(u) \rangle}$. If $\inf_u \|C(u)\| << \sup_u \|C(u)\|$ the bound that is established will not be tight, when a single $\underline{C}$ is exploited. One can adaptively subdivide the domain of the error function in u until $\inf_u \|C(u)\| << \sup_u \|C(u)\|$ no longer holds, employing several $\underline{C_i}$ for several different regions in u.

One can derive a global lower bound for $\underline{C}$ and hence a global bound for σ or μ, provided $C(u)$, $A(v)$ and $r(v)$ are represented as Bézier or NURBs curves. The square root of the minimal coefficient of $\langle C(u), C(u) \rangle$ can be employed as the global lower bound, $\underline{C}$, exploiting the convex hull property of the Bézier and NURBs representations [12]. Moreover, the maximal coefficient of σ or μ can be employed as a global upper bound on the error. Finally, one can associate parametric node values with each of the coefficients of the error function and apply refinement to the axis curve, $A(v)$, at the locations where the error is larger than the prescribed tolerance, τ. These node values are also known as Greville abscissae [13].

The Schoenberg variation diminishing property of splines [14] guarantees that the error of a spline approximation is in the order of $O\left(\left(\max_i \left(t_{i+1} - t_i\right)\right)^2\right)$, over a knot vector $\{t_i\}$. By orienting a cross section at each node value of the refined axis curve, the refinement of $A(v)$ in regions with large approximation error, will introduce more cross sections in these locations and allow the convergence to the exact sweep surface. Algorithm 1 presents the process of computing the global upper bound on the error of the sweep representation and automatically refining $A(v)$ and improving the approximation.

Algorithm 1 in line (1) assumes the availability of a sweep surface constructor that is able to approximate a sweep surface, given $C(u)$, $A(v)$, and optionally $r(v)$. Furthermore, the convergence of Algorithm 1 to the exact sweep surface is guaranteed via a sweep surface constructor in line (1) that *converge* to the exact sweep surface, as more cross sections are introduced. Such a sweep constructor that is able to exploits the refined versions of $A(v)$ to improve the quality and accuracy of the sweep surface approximation can be found in [1, 3, 5]. The error norm we provide in this framework cannot guarantee convergence by itself and it is a necessary but

Algorithm 1

Input:

$C(u)$, cross section curve;
$A(v)$, axis or skeleton curve;
$r(v)$, optional scale or profile curve;
τ, tolerance of sweep approximation;

Output:

$S(u,v)$, an approximation to a sweep surface prescribed by $C(u)$, $A(v)$, $r(v)$, to within τ.

Algorithm:

$i \Leftarrow 0$;
$A_0(v) \Leftarrow A(v)$;
$\psi(u,v) \Leftarrow \langle C(u), C(u) \rangle r^2(v)$;
$\chi(u,v) \Leftarrow 2\tau \underline{C} r(v)$;
do
(1) $S_i(u,v) \Leftarrow$ A sweep approximation using $C(u)$, $A_i(v)$, and $r(v)$;
$\mu_i(u,v) \Leftarrow \langle S_i(u,v) - A_i(v), S_i(u,v) - A_i(v) \rangle - \psi(u,v) - \chi(u,v)$;
$\{\mathcal{N}\} \Leftarrow$ Set of parametric node values associated with $\mu_i(u,v)$'s coefficients;
$A_{i+1}(v) \Leftarrow A_i(v)$ refined at all nodal locations v_k, such that $\mu_i(u_j, v_k) > 0$, $(u_j, v_k) \in \{\mathcal{N}\}$;
$i \Leftarrow i + 1$;
while ($A_{i+1}(v)$ is a refined version of $A_i(v)$);

insufficient condition for constructing a sweep surface that is accurate to within a prescribed tolerance. Hence, regions in $A(v)$ that are highly refined will closely approximate the exact sweep surface. Algorithm 1 exploits this sweep constructor in a "construct, measure the error, and refine" set of iterations. The $\psi(u,v)$ and $\chi(u,v)$ terms in Algorithm 1 are invariant under the refinement that is applied to $A(v)$ and are therefore precomputed once. Every location along $A(v)$ that is found inaccurate in Algorithm 1 becomes a candidate for refinement in the next iteration. In practice, two knots are inserted, one to the left and one to the right of every node point that violates the prescribed tolerance τ, in each iteration of the algorithm. The introduced knots are all single, minimizing continuity degradation. In Section 3, we present some examples that use Algorithm 1.

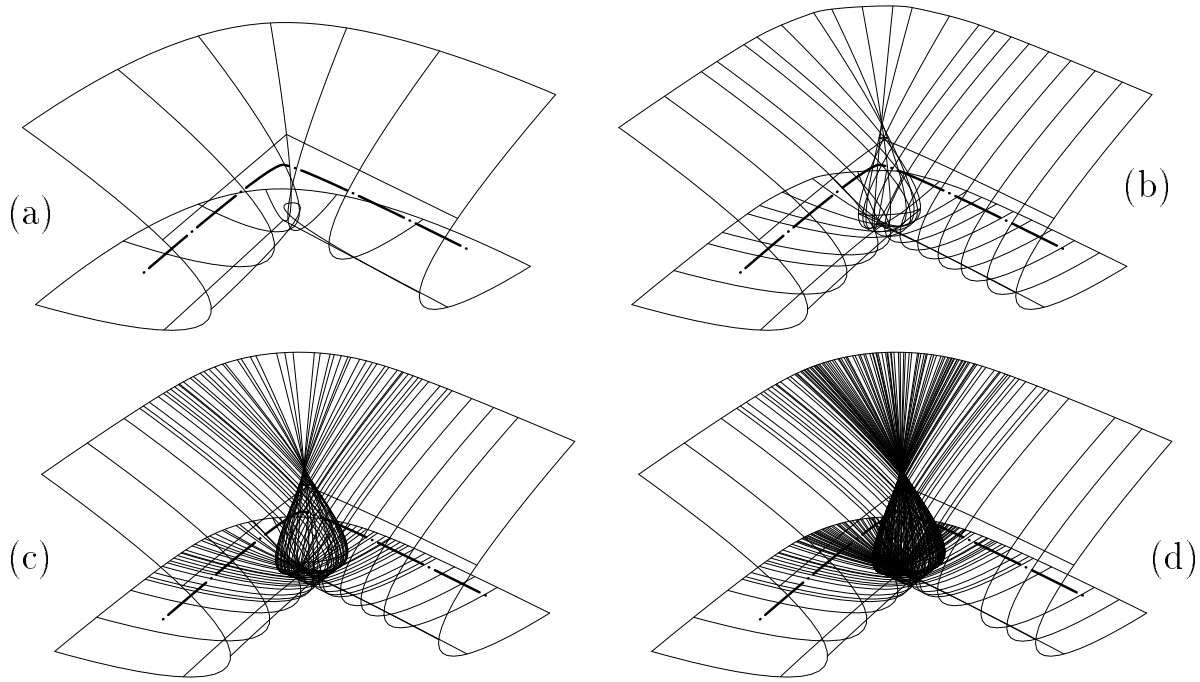

Figure 2: Four stages of improvements of a sweep approximation using Algorithm 1.

Stage	Error (τ)
1	1.201154
2	0.072710
3	0.001808
4	0.000055

Table 1: Convergence of global upper bound. Maximum error of the sweep surfaces in Figure 2.

3 Amelioration of the Sweep Surface

Figure 2 shows a U shaped cross section swept along an L shaped axis curve. Four iterations of Algorithm 1 were necessary to guarantee convergence below the prescribed tolerance. Table 1 shows the maximal error computed in each iteration. Stage four was found below the prescribed tolerance of $\tau = 0.0001$.

Figure 3 shows a cross section of a rounded square shape swept along a rounded square axis curve. Three iterations of Algorithm 1 were necessary to guarantee convergence below the prescribed tolerance. Table 2 shows the maximal error computed in each iteration. Stage three was below the prescribed tolerance of $\tau = 0.0001$. Figure 4 shows $\sigma(u, v)$, the scalar field that represents the error of the sweep construction (Equation (4)), for the three iterations

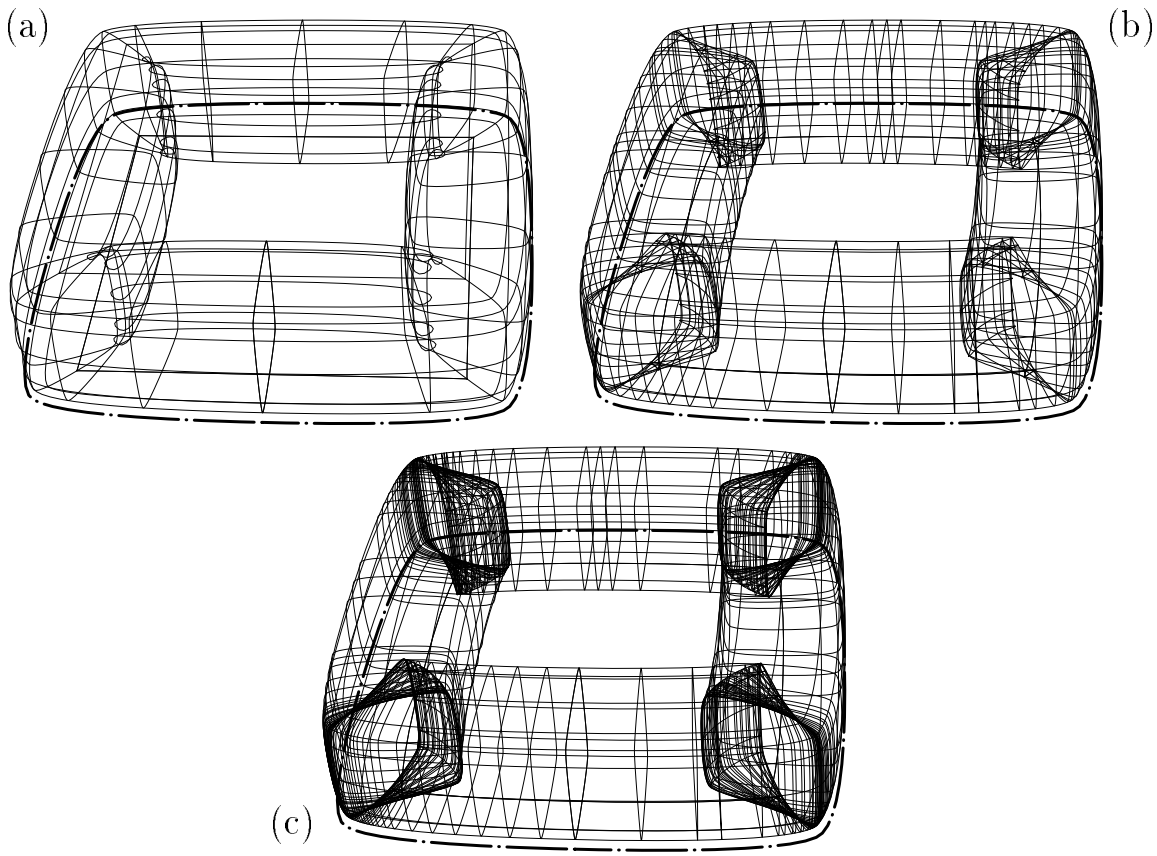

Figure 3: Three stages of improvements of a sweep approximation using Algorithm 1. Note the axis curve that is not inside the sweep surface.

Stage	Error (τ)
1	0.018209
2	0.001017
3	0.000016

Table 2: Convergence of global upper bound. Maximum error of the sweep surfaces in Figure 3.

that are shown in Figure 3, with *different* scales. The four distinct regions with large errors in Figures 4 (a)-(c) are related to the four self intersecting regions at the four inner corners of the surface. These areas get most of the refinement's attention as most of the new knots are assigned to these regions.

In Section 4, we consider methods to eliminate the self intersecting regions, due to large curvature in $A(v)$.

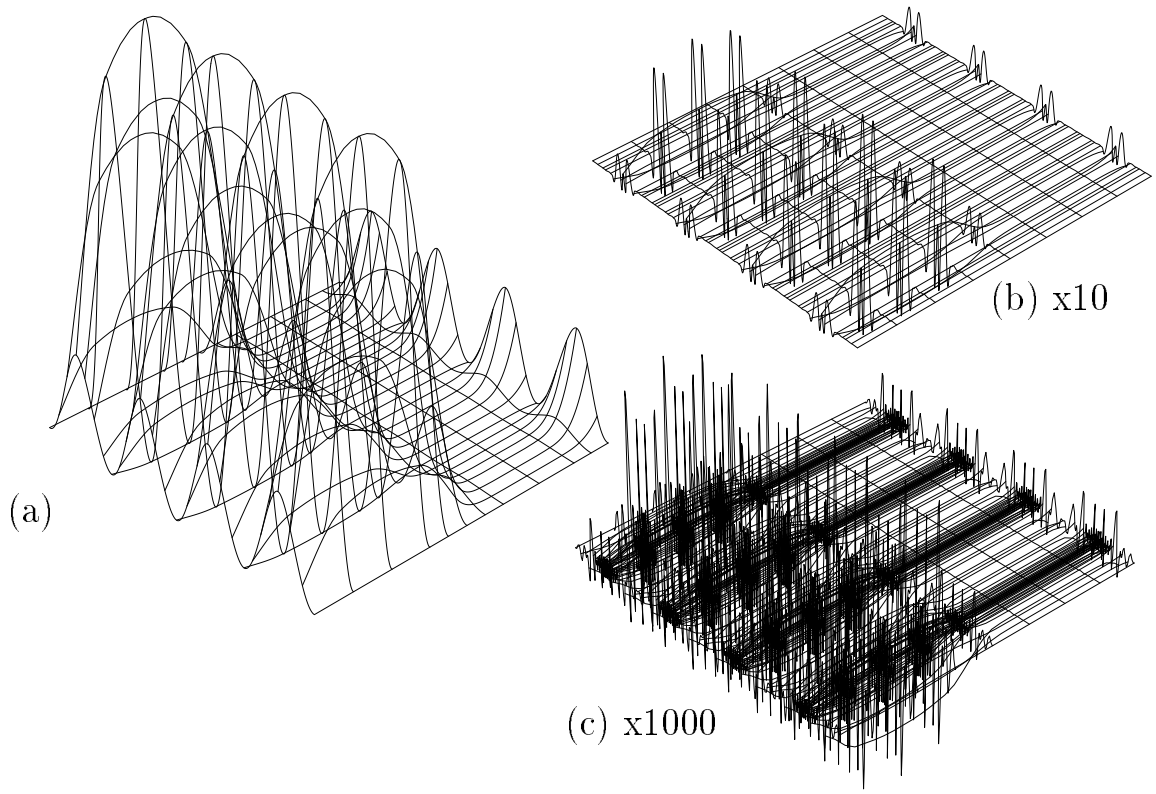

Figure 4: $\sigma(u, v)$ for the three sweep approximations in Figure 3. (b) is scaled by 10 and (c) is scaled by 1000, compared to (a). See also table 2.

4 Detection and Elimination of Local Self Intersections

Inspect Figure 5. This surface is a result of a sweep of a circular cross section $C(u)$ along a rounded rectangular axis curve, $A(v)$, that was computed using Algorithm 1. $A(v)$ has large curvature at the four corners, causing the sweep surface, $S(u, v)$, to self intersect. In this section, we consider methods to detect and eliminate the local self intersecting regions.

The self intersection problem in sweep surfaces is closely associated with the self intersection problem in offset curves [11] and similar techniques to eliminate self intersections can be exploited. Assume $A(v)$ and $C(u)$ do not self intersect, and let $\mathcal{S}(u, v)$ and $S(u, v)$ be the exact sweep surface as defined in Equation (1) and its approximation as derived in Section 2, respectively. Let $\kappa(v)$ be the curvature scalar field of $A(v)$ (Equation (2)). In the exact sweep surface, the isoparametric curve $\mathcal{S}(u, v_0)$ is an affine transformation of $C(u)$. In contrast, the isoparametric curve $\mathcal{A}_0(v) = \mathcal{S}(u_0, v)$ is a *generalized offset* of the axis curve,

$$\mathcal{A}_0(v) = \mathcal{S}(u_0, v) = A(v) + x(u_0)r(v)N(v) + y(u_0)r(v)B(v) + z(u_0)r(v)T(v), \qquad (9)$$

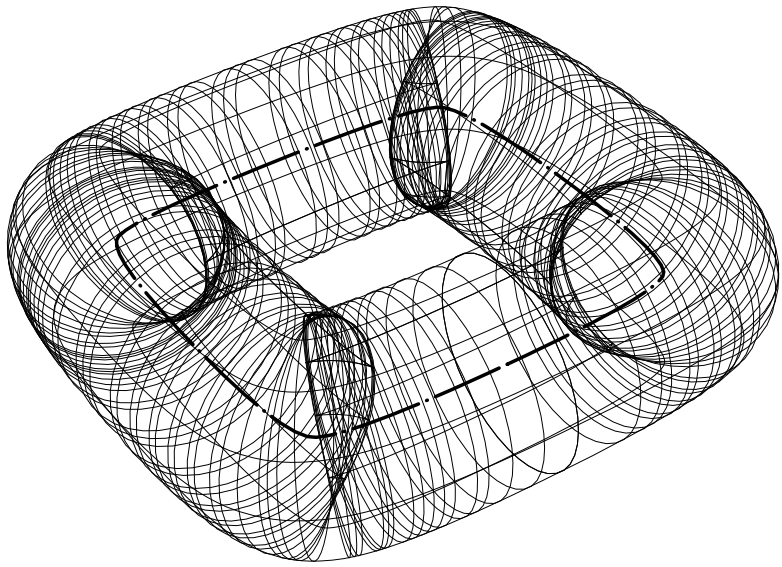

Figure 5: A NURBs representation of a sweep surface of a circular cross section along a rounded rectangular axis curve, $A(v)$. The surface locally self intersects in the regions where $A(v)$ has large curvature.

where $C(u) = (x(u), y(u), z(u))$ is the cross section curve, $r(v)$ is the profiling curve and $T(v)$, $N(v)$ and $B(v)$ are the tangent, normal and binormal fields of $A(v)$ (see Equation (2)). For a regular offset of a planar curve by an amount d, $x(u) \equiv d$, $y(u) \equiv z(u) \equiv 0$, and $r(v) \equiv 1$. Assume both $A(v)$ and $\mathcal{A}_0(v)$ are sufficiently differentiable. Without loss of generality assume v is the arc length parameterization [9] of $A(v)$. Then,

$$\begin{aligned}
\frac{d\mathcal{A}_0(v)}{dv} &= \frac{dA(v)}{dv} + x(u_0)\frac{dr(v)}{dv}N(v) + x(u_0)r(v)\frac{dN(v)}{dv} \\
&\quad + y(u_0)\frac{dr(v)}{dv}B(v) + y(u_0)r(v)\frac{dB(v)}{dv} + z(u_0)\frac{dr(v)}{dv}T(v) + z(u_0)r(v)\frac{dT(v)}{dv} \\
&= T(v) + x(u_0)\frac{dr(v)}{dv}N(v) + x(u_0)r(v)(-\kappa(v)T(v) - \tau(v)B(v)) \\
&\quad y(u_0)\frac{dr(v)}{dv}B(v) + y(u_0)r(v)\tau(v)N(v) + z(u_0)\frac{dr(v)}{dv}T(v) + z(u_0)r(v)\kappa(v)N(v) \\
&= \left(1 - x(u_0)r(v)\kappa(v) + z(u_0)\frac{dr(v)}{dv}\right)T(v) \\
&\quad + \left(x(u_0)\frac{dr(v)}{dv} + y(u_0)r(v)\tau(v) + z(u_0)r(v)\kappa(v)\right)N(v) \\
&\quad + \left(y(u_0)\frac{dr(v)}{dv} - x(u_0)r(v)\tau(v)\right)B(v), \qquad (10)
\end{aligned}$$

since $\frac{dT(v)}{dv} = \kappa(v)N(v)$, $\frac{dN(v)}{dv} = -\kappa(v)T(v) - \tau(v)B(v)$ and $\frac{dB(v)}{dv} = \tau(v)N(v)$ [9].

We consider a local self intersection in the sweep surface construction, an intersection that occurs due to a highly curved axis curve, in the neighborhood where the curvature of $A(v)$ is

too large. This, in contrast with global self intersection that might occur between two unrelated regions of the sweep surface. If $\left\langle \frac{d\mathcal{A}_0(v)}{dv}, \frac{dA(v)}{dv} \right\rangle > 0$ the sweep surface is monotone with respect to $A(v)$ and therefore will never self intersect in the local. Moreover, if $\left\langle \frac{d\mathcal{A}_0(v)}{dv}, \frac{dA(v)}{dv} \right\rangle < 0$ for some (u, v) location, then there exists an isoparametric curve $\mathcal{A}_0(v) = \mathcal{S}(u_0, v)$ that flips its tangent direction with respect to the tangent direction of $A(v)$. This isoparametric curve is likely to cause a local self intersection.

$$\begin{aligned}
\left\langle \frac{d\mathcal{A}_0(v)}{dv}, \frac{dA(v)}{dv} \right\rangle &= \left\langle \left(1 - x(u_0)r(v)\kappa(v) + z(u_0)\frac{dr(v)}{dv}\right) T(v) \right. \\
&\quad + \left(x(u_0)\frac{dr(v)}{dv} + y(u_0)r(v)\tau(v) + z(u_0)r(v)\kappa(v) \right) N(v) \\
&\quad \left. + \left(y(u_0)\frac{dr(v)}{dv} - x(u_0)r(v)\tau(v) \right) B(v) \ , \ \ T(v) \right\rangle \\
&= \left(1 - x(u_0)r(v)\kappa(v) + z(u_0)\frac{dr(v)}{dv}\right). \qquad (11)
\end{aligned}$$

In many cases the cross section, $C(u)$, is planar or alternatively no profiling or scaling function, $r(v)$, is employed ($\frac{dr(v)}{dv} = 0$). In both cases, Equation (11) is reduced to $\left\langle \frac{d\mathcal{A}_0(v)}{dv}, \frac{dA(v)}{dv} \right\rangle = (1 - x(u_0)r(v)\kappa(v))$. The non-negativity of Equation (11) guarantees that there is no local self intersection the sweep surface approximation. Nevertheless, and while Equation (11) provides tight bounds in practice, the detection of a negative region using Equation (11) does guarantee the existence of a self intersection only in sweeps of constant scale and either zero twist in the orientation frame or a circular cross section. One example of a false detection of a self intersection is presented in Figure 6. A profile curve that scales down the sweep surface near a highly curved region of the axis curve actually prevents the self intersection.

If there exists a surface location $(u_0,\ v_0)$ such that $\frac{1+z(u_0)\frac{dr(v)}{dv}}{\kappa(v_0)} < x(u_0)r(v_0)$ $(\frac{1}{\kappa(v_0)} < x(u_0)r(v_0))$ for a planar cross section or constant scale sweep), then the tangent of $\mathcal{A}_0(v) = \mathcal{S}(u_0, v)$ will be *reversed* with respect to the tangent of $A(v)$ at v_0. In an analogous approach to that taken in [11], one can consider the symbolic computation over the (piecewise) rationals of $\delta(v) = \left\langle \frac{d\mathcal{A}_0(v)}{dv}, \frac{dA(v)}{dv} \right\rangle$ to detect the locations where the tangent direction of $\mathcal{A}_0(v)$ is being reversed. From the continuity of $\delta(v)$, the zero set of $\delta(v)$ separates the domains of $\delta(v) > 0$ from the domains of $\delta(v) < 0$. If $\delta(v)$ is negative, the tangent direction of $\mathcal{A}_0(v)$ is reversed.

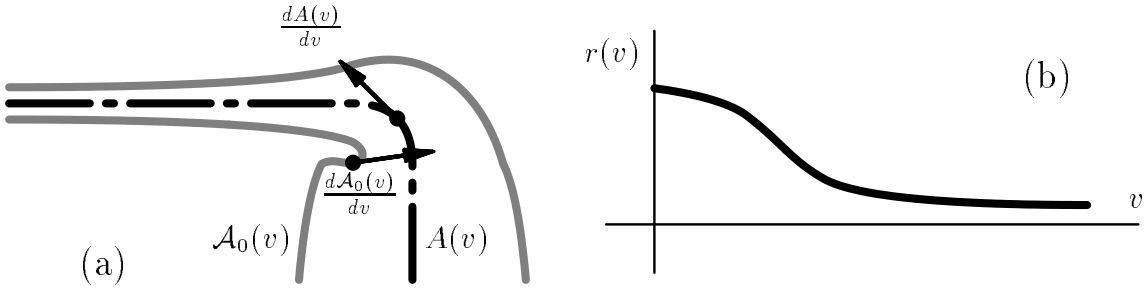

Figure 6: While tight in practice, the computation of $\left\langle \frac{d\mathcal{A}_0(v)}{dv}, \frac{dA(v)}{dv} \right\rangle$ might be too strict for the detection of local self intersections in case of the approximation of variable radius sweep surfaces. In (a), the change in the magnitude of the scaling function at the area of high curvature in the axis curve actually prevents the self intersection. The inner product of the tangent of the axis curve, $\frac{dA(v)}{dv}$, and the tangent of the sweep curve, $\frac{d\mathcal{A}_0(v)}{dv}$ is clearly negative in (a) while the sweep surface does not self intersect. In (b), the scale function is shown, parameterized along the arc length parameter, v, of $A(v)$.

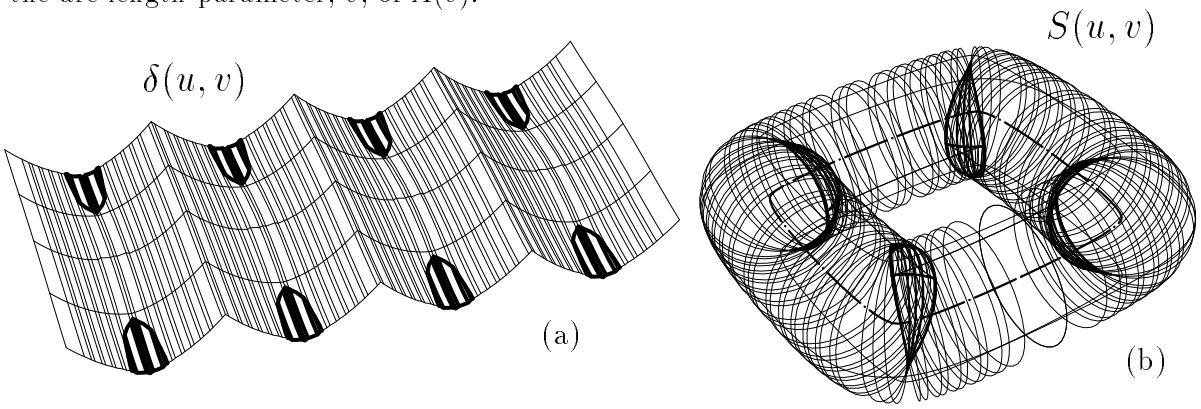

Figure 7: $\delta(u,v) = \left\langle \frac{\partial S(u,v)}{\partial v}, \frac{dA(v)}{dv} \right\rangle$ is used in (a) to detect the locally self intersecting regions in the sweep surface, $S(u,v)$, shown in (b). Shown in thick lines are the respective regions for which $\delta(u,v) < 0$.

Considering the sweep approximation, $S(u,v)$ might self intersects if the following holds,

$$\delta(v) = \left\langle \frac{d\mathcal{A}_0(v)}{dv}, \frac{dA(v)}{dv} \right\rangle = \left\langle \frac{\partial S(u_0,v)}{\partial v}, \frac{dA(v)}{dv} \right\rangle < 0, \quad \forall u_0 \in [u_{min}, u_{max}], \tag{12}$$

or,

$$\delta(u,v) = \left\langle \frac{\partial S(u,v)}{\partial v}, \frac{dA(v)}{dv} \right\rangle < 0, \tag{13}$$

for some (u,v) in the domain of $S(u,v)$.

Thus, one can symbolically compute the scalar field of $\delta(u,v)$ and determine its negative regions, if exists. In Figure 7, we see $\delta(u,v)$ computed for the surface in Figure 5 along with the respective regions in the sweep surface, for which $\delta(u,v) < 0$, highlighted in thick lines.

While the manifestation of the local self intersections as a zero set computation on $\delta(u,v)$ alleviates the difficulties in detecting local self intersections, eliminating these self intersecting

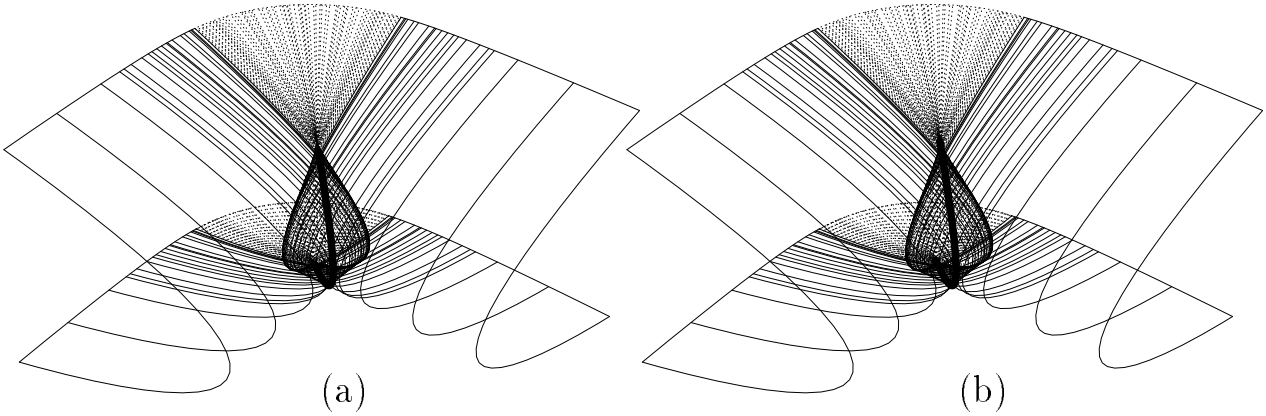

Figure 8: $S(u, v)$ can be subdivided in v into the regions for which $\delta(u, v) > 0$ (full lines) and regions for which $\exists u$ such that $\delta(u, v) < 0$ (dotted lines). The intersection curves of the regions for which $\delta(u, v) > 0$ can then be computed using a surface surface intersection (SSI) algorithm. In (a), a subdivision based SSI algorithm was used to derive the seed to the self intersection curve that is extrapolated using surface marching in (b). See also Figures 2 and 9.

regions is more strenuous. In [11], the representation of the offset curve was split into two regions before and after the region in which the tangent is reversed. The two regions were then intersected against each other to find the exact self intersection location. Unfortunately, we are unable to directly extend this approach to the problem at hand, mainly because the self intersecting regions in the sweep surface S can assume an arbitrary shape in the parametric space of S.

One can isolate the v domain along the axis curve $A(v)$ for which $\exists u$ such that $\delta(u, v) < 0$. Denote such a domain by (v_1, v_2). S can be subdivided into three regions, $v < v_1$, $v_2 < v$, and $v_1 < v < v_2$. The first two regions can be intersected against each other using a surface surface intersection (SSI) algorithm [15] to yield a *subset* of the real self intersecting curve (See Figure 8). Surface marching and tracing methods [15] can then be employed to march on S and extrapolate the solution until the two curve segments of the self intersection meet in the parametric space of S (See Figures 8 and 9).

Once the self intersecting regions are completely determined, these regions can be eliminated from $S(u, v)$. Figure 10 shows the two sweep surface examples used in this paper, after the self intersecting regions were properly trimmed away, represented as a trimmed NURBs surfaces.

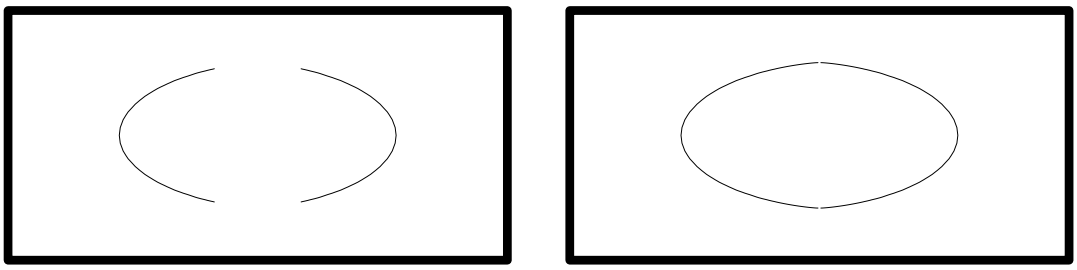

Figure 9: The parametric domain of the sweep surface in Figure 8. In (a), the result of the subdivision based SSI is presented. In (b), this result has been numerically extrapolated using surface marching.

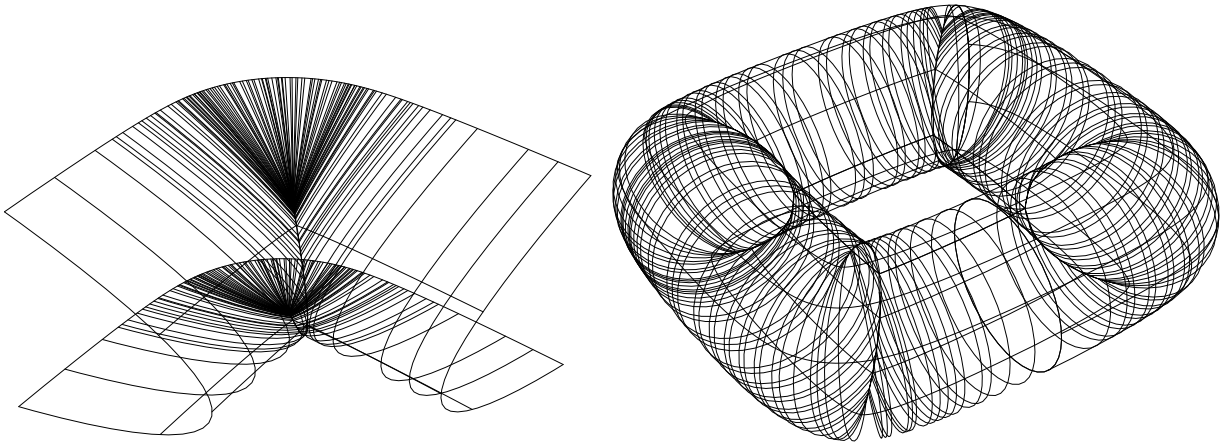

Figure 10: The surfaces in Figures 5, and 8, are shown as trimmed surfaces after the self intersecting regions were eliminated.

5 Conclusion

We have presented methods to bound the error of a sweep surface approximation, to improve the accuracy of the representation, and to detect and eliminate self intersections that can result. One can consider a *bivariate* scaling or profiling surface, $r(u, v)$, as part of the sweep constructor, an extension that trivially affects Equation (1). The respective error norm $\mu(u, v)$ in Equation (8) should also be modified to reflect the bivariate profiling. $r(u, v)$ provides the ability to change the shape of the cross section $C(u)$ as well as its size, while swept along $A(v)$.

The presented approach for the elimination of self intersections in sweep surfaces has room for improvements. The exploited SSI solver needs to properly handle surfaces that are (almost) tangent, a condition that may arise. A numerical surface marching method was employed to extrapolate a seed established via a subdivision based SSI. Cases of tangent surfaces are difficult

to handle for all SSI algorithms but unfortunately can arise in the self intersecting regions, in sweep surface constructions.

Herein, as well as in [11], curve refinement was exploited as the tool to converge to the exact representation. In [16], an approach to improve an offset approximation was proposed using perturbation of control points of the offset approximation, minimizing the error function that has been symbolically computed. A similar perturbation of control points might be exploited for the improvement of sweep surface approximations represented as Bézier or NURBs surfaces. This approach should be further investigated.

References

[1] M. Bloomenthal. Approximation of Sweep Surfaces by Tensor Product B-splines. Tech Reports UUCS-88-008, University of Utah, 1988.

[2] W. Bronsvoort and J. J. Waarts. A Method for Converting the Surface of a Generalized Cylinder into a Bspline Surface. Computer & Graphics, Vol. 16, No. 2, pp. 175-178, 1992.

[3] S. Coquillart. A control Point Based Sweeping Technique. IEEE Computer Graphics and Applications, Vol. 7, No. 11, pp 36-44, November 1987.

[4] J. K. Johnstone and J. P. Williams. A rational model of the surface swept by a curve. Computer Graphics Forum, Vol. 14, No. 3, pp 77-88, Eurographics 1995.

[5] P. Siltanen and C. Woodward. Normal Orientation Methods for 3D Offset Curves, Sweep Surfaces and Skinning. Computer Graphics Forum, Vol. 11, No. 3, pp 449-457, Eurographics 1992.

[6] W. Bronsvoort, P. Van Nieuwenhuizen, F. Post. Display of profiled Sweep Objects. The Visual Computer, Vol. 5, No. 3, pp. 147-157, 1989.

[7] W. Bronsvoort. A surface-scanning Algorithm for Displaying Generalized Cylinders. The Visual Computer, Vol. 8, No. 3, pp. 162-170, 1992.

[8] J. W. Ahn, M. S. Kim, and S. B. Lim. Approximate General Sweep Boundary of a 2D Curved Object. Comput. Vision, Graphics and Image Processing, Vol 55, No. 2, pp 98-128, March 1993.

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

[10] F. Klok. Two Moving Coordinate Frames for Sweeping along a 3D trajectory. Computer Aided Geometric Design, Vol. 3, No. 3, pp 217-229. 1986.

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

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

[13] G. Farin. NURB Curves and Surfaces from Projective Geometry to Practice Use. A. K. Peters. Wellesley, Massachusetts, 1995.

[14] M. Marsden and I. J. Schoenberg. On Variation Diminishing Spline Approximation Methods. *Mathematica*, Vol. 8(31), No. 1, pp 61-82, 1966.

[15] G. A. Kriezis, N. M. Patrikalakis, and F. E. Wolter. Topological and differential equation methods for surface intersections. Computer Aided Design, Vol. 24, No. 1, pp 41-55, January 1992.

[16] G. Elber and E. Cohen. Offset Approximation Improvement by Control Points Perturbation. Mathematical methods in Computer Aided Geometric Design II. Tom Lyche and Larry L. Schumaker, pp 229-237. Academic Press, Inc., 1992.