Euclidean Offset and Bisector Approximations of Curves over Freeform Surfaces

Gershon Elber[a,*], Myung-Soo Kim[b]

[a] *Computer Science Department, Technion, IIT, 32000 Haifa, Israel*
[b] *Department of Computer Science and Eng., Seoul National University, South Korea*

Abstract

The computation of offset and bisector curves/surfaces has always been considered a challenging problem in geometric modeling and processing. In this work, we investigate a related problem of approximating offsets of curves on surfaces (OCS) and bisectors of curves on surfaces (BCS). While at times the precise geodesic distance over the surface between the curve and its offset might be desired, herein we approximate the Euclidean distance between the two. The Euclidean distance OCS problem is reduced to a set of under-determined non-linear constraints, and solved to yield a univariate approximated offset curve on the surface. For the sake of thoroughness, we also establish a bound on the difference between the Euclidean offset and the geodesic offset on the surface and show that for a C^2 surface with bounded curvature, this difference vanishes as the offset distance is diminished. In a similar way, the Euclidean distance BCS problem is also solved to generate an approximated bisector curve on the surface. We complete this work with a set of examples that demonstrates the effectiveness of our approach to the Euclidean offset and bisector operations.

Keywords: Tool path design, Print path design, Paint paths, NC machining, additive manufacturing, offset, bisector, medial axis, Voronoi diagram

1. Introduction

Given a regular curve $C(t) = S(u(t), v(t))$ on a smooth parametric surface $S(u, v)$, the vector $C'(t)$ is in the tangent plane of the surface at $S(u(t), v(t))$. Rotating the normalized tangent vector $T(t) = C'(t)/\|C'(t)\|$ clockwise (or counter-clockwise) by angle $\pi/2$ in the tangent plane of the surface, we can get a unit vector $N(t)$ which is orthogonal to $T(t)$ in the same tangent plane. (See Figure 1.) Starting from each curve location $C(t)$, in the direction of $N(t)$, we can walk along the geodesic path on the surface $S(u, v)$ by a distance d. The locations we have reached in this process define the *geodesic offset* curve $O_d^g(t)$ of $C(t)$ on the same surface $S(u, v)$.

The geodesic offset curve has many important applications, for example, in the synthesis of rail curves of a fillet, given the intersection curve of the two surfaces to be blended [16]. Moreover, the geodesic offset curves can also be employed in the surface covering by creating a univariate path that visits every point on the surface up to a certain tolerance or gap. Such coverings can greatly simplify the planning of a paint process or inspection paths [1], tool path planning in NC machining [23], and even printing paths in additive manufacturing.

In the construction of geodesic offset curves, the most time-consuming step is in the process of walking along geodesic paths from all points of the given curve $C(t)$. Patrikalakis and Bardis [29] proposed a sampling technique that constructs $O_d^g(t_i)$ at some parameters t_i and approximate the geodesic offset curve $O_d^g(t)$ by interpolating these geodesic offset points. Wolter and Tuohy [38] revisited the geodesic offset problem in their general approach to the approximation of high-degree and procedural curves. In the conventional approaches, the discrete sample points $O_d^g(t_i)$ are computed only approximately. The geodesic paths on a regular parametric surface $S(u, v)$ are given as solutions of 2nd order ordinary differential equation, which can be approximated numerically [4, 29].

From an industrial perspective, Ulmet [36] suggested two simple alternatives to the geodesic path construction on the surface $S(u, v)$: (i) an orthogonal line to the plane curve $(u(t), v(t))$ in the parameter domain of $S(u, v)$, and (ii) the projection of an orthogonal tangent line $C(t)+sN(t)$, $(s \geq 0)$, to the surface $S(u, v)$. These simple solutions often produce desirable results for certain engineering applications [36], where the overall quality of offset curves is more important than the accuracy of the geodesic path approximation. (For example, the curve quality may include such features as small variations of curvature, no undesired oscillations, not too many segmentations into small pieces,

*Corresponding author
Email addresses: gershon@cs.technion.ac.il (Gershon Elber), mskim@snu.ac.kr (Myung-Soo Kim)

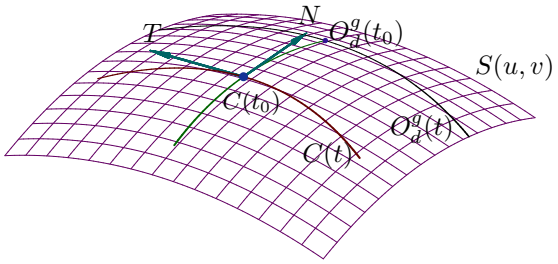

Figure 1: Geodesic offset $O_d^g(t)$ of $C(t)$, on a freeform surface $S(u,v)$.

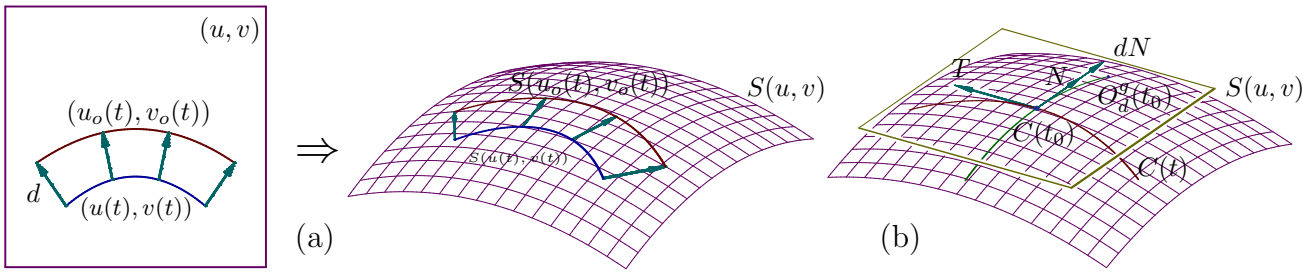

Figure 2: Geodesic offset approximation methods of Ulmet [36]: (a) the first approach maps a planar curve offset in the domain of the surface $S(u,v)$ to its image, and (b) the second approach is based on the curve projection onto $S(u,v)$.

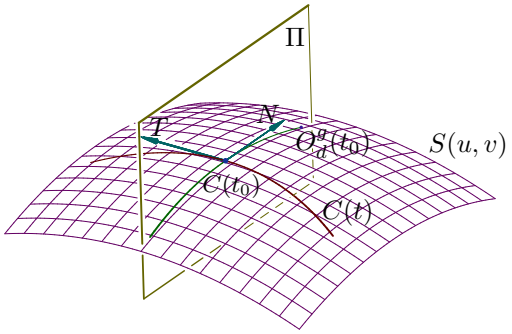

Figure 3: The method of Tam et al. [37] based on the cross-section of the surface $S(u,v)$ with the normal plane Π.

etc.) Depending on the shape variations of the given surface, the exact geodesic offset curves may not satisfy some of these important curve properties. Thus, it is worthwhile to explore and compare other alternatives to the exact geodesic offset. In this paper, we propose a new approach that can overcome some drawbacks of Ulmet [36], which will be discussed below.

The first approach of Ulmet [36] computes the planar offset curve $(u_o(t), v_o(t))$ to the curve $(u(t), v(t))$ in the parameter domain and constructs a space curve $S(u_o(t), v_o(t))$ on the surface $S(u,v)$. (See Figure 2(a).) Ulmet [36] discuss some obvious drawbacks of this simple approach due to the parametrization-dependent nature of this technique. That is, regions of the same size in the (u,v)-domain can be mapped to patches of considerably different sizes over the surface $S(u,v)$ and thus the gap between the curve $C(t)$ and its offset curve may not be uniform. An advantage of this approach is in that the offset curve $S(u_o(t), v_o(t))$ is generated as a continuous curve when the planar offset curve $(u_o(t), v_o(t))$ is continuous. Thus, no further curve interpolation is needed.

The second approach of Ulmet [36] essentially projects a space curve $C(t) + d \cdot N(t)$ to the surface $S(u,v)$, which is a non-trivial task. (See Figure 2(b).) Tam et al. [37] suggested a slightly different alternative, using a two-step construction: (i) intersecting the surface $S(u,v)$ with the normal plane of $C(t)$, and (ii) moving along the cross-sectional curve by the offset distance d. (See Figure 3.) Though it may look conceptually easier than the projection to the surface, the method of Tam et al. [37] is also not suitable for generating continuous offset curves on surfaces. As in the case of the conventional methods [5, 29, 38], the projection method of Ulmet [36] and the intersection method of Tam et al. [37] have to generate some samples on offset curves and then interpolate these sample points on the surface $S(u,v)$. In this paper, we suggest a convenient one-step construction method that directly computes continuous offset curves on surfaces.

Replacing the normal plane of Tam et al. [37] by a circle of radius d (i.e., the offset distance) contained in the normal plane, we generate intersection points on the surface $S(u,v)$ (instead of a cross-sectional intersection curve).

By continuously moving the normal plane along the curve $C(t)$, the circle of radius d will generate a pipe surface [28]. The intersection of the surface $S(u,v)$ with this pipe surface will generate continuous curves on $S(u,v)$. Each point of the intersection curve is (locally) at distance d from $C(t)$ under the Euclidean metric in $\mathbb{R}^3$. Nevertheless, the geodesic distance can be larger than d when measured over the surface $S(u,v)$. This property of over-offsetting is the main drawback of our approach. On the other hand, the Euclidean metric greatly simplifies our approach in the remaining post-processing stages of the offset construction. We discuss the main advantages of these simplification features below.

The above computation is local and ensures all intersection curves are locally at distance d from $C(t)$. Yet, some curve segments of the offset curves can be globally closer to the given curve $C(t)$ than the offset distance d and thus should be eliminated by trimming the offset curve at certain self-intersection points. Our approach deals with this trimming problem by measuring the Euclidean distance in $\mathbb{R}^3$. In the majority of other methods, the offset trimming is done in the arrangement of curve segments in the (u,v)-parameter domain of the surface $S(u,v)$ [37] or in a flattened planar domain for mesh surfaces [40, 41]. Due to the numerical errors in offset curve sampling and the computational difficulty measuring the geodesic distance on the surface, the offset trimming can be highly unstable, in particular, when the offset curves have tangential intersections. In this paper, we greatly simplify the offset curve trimming using the pipe surface of $C(t)$. All segments of the offset curve $O(t)$ that are contained in the interior of the pipe surface are trimmed away to generate a trimmed offset curve on the surface $S(u,v)$. (Note that Seong et al. [34] proposed this approach for the trimming of offset curve/surface self-intersections in the conventional Euclidean plane and space.)

The Euclidean metric also simplifies the construction of bisector curves, medial axis, and Voronoi diagram for curves on surfaces, which will be shown in Section 7 of this paper. Given two curves, $C(t)$ and $D(s)$, on the surface $S(u,v)$, the bisector surface $B(s,t)$ between $C(t)$ and $D(s)$ can be generated as a rational surface, in general [10]. By intersecting the two rational freeform surfaces $S(u,v)$ and $B(s,t)$ (based on the algorithm of Bartoň et al. [3]), we can construct a Euclidean approximation to the bisector curve, on the given surface $S(u,v)$. As in the case of offset curve trimming, we also need to eliminate certain segments of the bisector curves on surfaces since the bisector segments (locally at equal minimum distance from $C(t)$ and $D(s)$) may be globally closer to some other parts of $C(t)$ or $D(s)$. Our approach can handle the bisector curve trimming in the same way as the problem in the Euclidean plane or space. Nevertheless, the geodesic bisector curve trimming on surfaces is a highly non-trivial problem due to the difficulty in bounding the minimum and maximum geodesic distances between two curves on surfaces [32]. We discuss some more details below.

Numerically solving a system of differential equations, Rausch et al. [32] presented a local curve tracing technique for the geodesic bisector (medial axis) of two curves on a surface, while assuming a starting point is given on the bisector. Some curve segments thus constructed on the surface may be globally closer to other parts of the two curves than the local equi-geodesic-distance from the two given curves. Nevertheless, the trimming of bisector curves has not been solved in this approach due to the difficulty in computing the globally minimum geodesic distance on surfaces. For a set of discrete points on surfaces, under an assumption on the non-singularity of offset circles, Kunze et al. [22] presented an algorithm (of a divide-and-conquer style) for constructing the geodesic Voronoi diagram of these points on surfaces. Because of the computational difficulty in measuring and upper/lower bounding the global geodesic minimum and maximum distances between two curves on surfaces, the majority of previous results have been limited to the construction of geodesic Voronoi diagrams on triangular meshes [21, 26]. Compared with the mesh-based approaches to geodesic distance-related problems on surfaces, the existence of rational representations for pipe and bisector surfaces greatly simplifies the associated geometric operations by reducing the given problem to a special case of intersecting two rational surfaces (e.g., using the construction algorithm of Bartoň et al. [3]) and removing certain redundant solutions using simple (often squared) Euclidean distance fields.

Each point on the offset curve (constructed by our proposed approach on surface $S(u,v)$) is at distance d from the space curve $C(t) = S(u(t), v(t))$, in the Euclidean space $\mathbb{R}^3$. When measured using the surface (geodesic) metric, the distance can be larger, but never be less than the Euclidean distance d. This means that our approximation technique can be used for the construction of a certain upper bound of the exact geodesic offset curve.

The main contributions of this work can be summarized as follows:

- We propose an algorithm to approximate the Euclidean offsets of curves over surfaces.
- We present an adaptation of a planar offset scheme, to trim self-intersections in offsets of curves on surfaces.
- We extend this approach and show it can be also used to other applications, such as the Euclidean approximation of bisectors on surfaces.

The rest of this paper is organized as follows. In Section 2, related previous work is discussed. Section 3 presents the algorithmic approach we use for the Euclidean approximation of offset curves on surfaces, and in Section 4, we

establish bounds on the maximal possible deviation of the presented Euclidean offset compared to the geodesic offset. In Section 5, we briefly consider the question of eliminating self-intersections in OCS. In Section 6, some results are presented, whereas in Section 7, we extend this Euclidean approximation approach to handle the BCS problem, and approximate bisectors of curves over freeform surfaces. Finally, we discuss future work and conclude, in Section 8.

2. Previous Work

The problem of computing offset and bisector curves/surfaces has a long history of development in the area of geometric modeling and processing [2, 11, 12, 27, 30], due to the essential role of offsets, medial axis, and Voronoi diagrams in many important applications, including the toolpath generation for CNC machining [18], geometric tolerancing, etc. The majority of previous results are, however, mainly focused on the development of approximation techniques for non-rational offsets and bisectors using rational curves and surfaces. To make the offset representation exact, there has also been considerable research work on the special cases where the exact offset curves and surfaces can be represented in rational forms [14]. Compared with the rich set of research activities on the offset and bisector computation in Euclidean spaces, there are only a few previous results on the geodesic offset and bisector computation on rational surfaces [5, 29, 32, 38]. The majority of geodesic offset curves and geodesic Voronoi diagrams have been constructed for polygonal surfaces such as triangular meshes [19, 21, 26, 39].

Compared with the complexity of mesh-based algorithms and the associated data structures, the geometric reasoning based on the (u, v)-domain of a surface $S(u, v)$ is simpler (than one might have expected). Once the geodesic offset curves on $S(u, v)$ are mapped back to the (u, v)-domain, the offset curve trimming (that removes redundant curve segments) can be done in the same way as the conventional planar curve offset trimming algorithms [8, 20, 24, 25, 34]. In this section, we focus on the previous results that can deal with geodesic-distance related problems directly on regular parametric surfaces, without using their polygonal or mesh approximations (that may introduce additional computing time, inaccuracy, etc).

Patrikalakis and Bardis [29] introduced the problem of geodesic offset curve construction to geometric modeling. Wolter and Tuohy [38] revisited the geodesic offset curve problem as a special case of approximating procedurally defined curves on surfaces. Brunnet [5] considered the geodesic offset problem on the surfaces of revolution, where the problem is reduced to that of solving the zero of a rather expensive integral function. Consequently, in these conventional approaches, the Runge-Kutta scheme was the method of choice for an efficient approximation of the geodesic offset curves. The variational approach of Pottmann and Hofer [31] can be used instead for more accurate approximate solutions to the geodesic path construction and the curve interpolation on a surface, again using numerical algorithms based on discretization and thus producing dense polygons with vertices on the surface. From an industrial perspective, Ulmet [36] suggested two simple alternatives to the geodesic offset curve construction. Tam et al. [37] proposed a slightly different alternative.

The generation of optimal NC tool paths often requires the maintenance of constant scallop-height [15, 33, 35], for which the offset distance is computed as a function $d(t)$. Depending on the specific way of each method in approximating the local geometry of $S(u, v)$ and the swept volume of the cutter tool, the variable distance function $d(t)$ is formulated differently. Nevertheless, the basic construction scheme for trimmed geodesic offset curves is about the same as in the case of constant-radius geodesic offsets.

Rausch et al. [32] presented a curve tracing technique for the geodesic bisector of two curves on surfaces. Kunze et al. [22] presented an algorithm for constructing the geodesic Voronoi diagram for a set of discrete points on surfaces. The majority of previous results are on the construction of geodesic Voronoi diagrams on triangular meshes [21, 26]. To the best of our knowledge, the current work is the first result that approximates the geodesic bisectors on surfaces from the industrial perspective, as suggested by Ulmet [36] for the geodesic offsetting problem.

3. Algorithm

Let $S(u, v)$ be a regular C^1 parametric surface, and let $c(t) = (u(t), v(t))$ be a regular C^1 parametric curve embedded in S, as $C(t) = S(c(t))$.

Definition 1. *The Euclidean offset curve by amount d, $O_d(t) \subset S(u, v)$, to $C(t)$ on $S(u, v)$ satisfies $||O_d(t) - C(t)|| = d$,*

where $||\cdot||$ denotes the L_2 norm, and d being the Euclidean offset distance. $O_d(t) \subset S(u, v)$ and hence the sought locus of point in $S(u, v)$ that are at a Euclidean distance d from $C(t)$ can be reduced to the following set of non linear constraints:

$$\begin{aligned} \|C(t) - S(u, v)\| &= d, \\ \langle C(t) - S(u, v), C'(t) \rangle &= 0, \end{aligned} \tag{1}$$

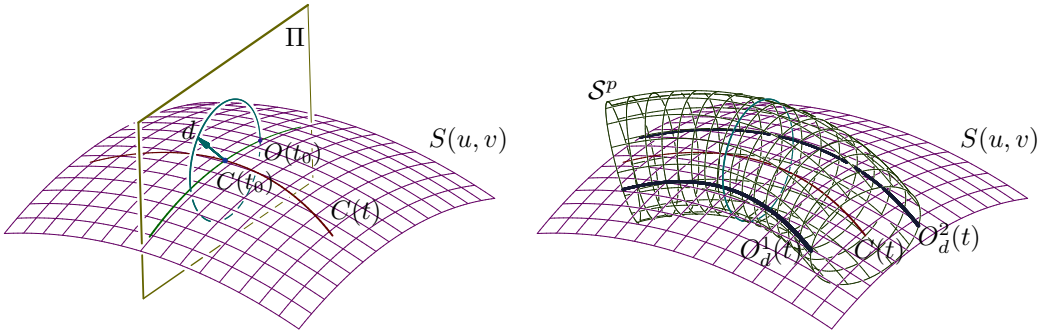

Figure 4: In (a), the geodesic offset approximation is computed using a swept circle of radius d contained in the normal plane, Π, of the curve $C(t)$. This circle is swept along $C(t)$ to yield the pipe surface $\mathcal{S}^p$ shown in (b). The intersection of $\mathcal{S}^p \cap S$ yields $O_d^i(t)$, $i = 1, 2$, two offset curves on S, on both sides of $C(t)$.

having two constraints in three variables (u, v, t) resulting in a univariate solution - the Euclidean offset curve.

The Constraints in Equations (1) find all locations on S that are locally at Euclidean distance d from $C(t)$, or in essence, define a pipe of radius d around $C(t)$. These constraints, in essence, define the locus of points of distance d from $C(t)$ as a pipe of radius d around $C(t)$, or a sweep of a circle of radius d along $C(t)$. See Figure 4 (b), for an example. The first constraint in Equation (1), makes sure the distance is d, while the second Constraint in (1) ensures this distance is locally minimal due to the imposed orthogonality, and together they define $O_d(t)$. A few comments:

1. One should note that nothing in Constraints (1) necessitates for $C(t)$ to be precisely on $S(u, v)$. This generalization allows one, if so desired, to use Constraints (1) and find the locus of points (i.e. curve(s)) on some surface S that are at distance d from some general curve $C(t)$, not on S. While, for simplicity, in the ensuing discussion we will assume $C(t)$ is on $S(u, v)$, examples where $C(t)$ is not on $S(u, v)$ will also be presented.
2. If $C(t)$ and $S(u, v)$ are (piecewise) polynomial or rational, the first constraint in Constraints (1) is not rational. This, due to the computed norm (that has a square root). Hence, in an effort to be algebraic, we use the square of the first constraint.
3. $C(t)$ can have several disjoint offset curves as a result of solving Constraints (1). If surface S wraps upon itself, there can be more than two solutions and if $C(t)$ is on the boundary of S, there might be only one solution.
4. Typically however, $C(t)$ will have two, left and right, solutions for this offset problem, as posed in Constrains (1). See, for example, $O_d^1(t)$ and $O_d^2(t)$ in Figure 4 (b). Further, we might be interested in only one of these solutions. Hence, we can also pose a third, inequality or semi-algebraic, constraint to filter out the undesired solution, that can be either:

$$\begin{aligned} \langle C'(t) \times (C(t) - S(u,v)), N(u,v) \rangle &> 0, \\ \text{or} & \\ \langle C'(t) \times (C(t) - S(u,v)), N(u,v) \rangle &< 0, \end{aligned} \tag{2}$$

   where $N(u, v)$ is a normal field (that always exists due to the regularity requirement) of S.
   Note this semi-algebraic constraint does not affect the dimension of the solution space (a univariate).

Hence, the final used set of constraints is:

$$\begin{aligned} \langle C(t) - S(u,v), C(t) - S(u,v) \rangle &= d^2, \\ \langle C(t) - S(u,v), C'(t) \rangle &= 0, \\ \langle C'(t) \times (C(t) - S(u,v)), N(u,v) \rangle &\lessgtr 0, \end{aligned} \tag{3}$$

4. Approximation Bounds

Denote by $O_d^g(t)$, the geodesic offset curve to $C(t)$, by amount d, on $S(u, v)$. Clearly, $\|O_d^g(t) - C(t)\| \leq \|O_d(t) - C(t)\|$. However, one can establish a bound on the deviation between $O_d(t)$ and $O_d^g(t)$.

Assume $S(u, v)$ is a C^2 continuous and let κ_n^{max} be the maximal normal curvature of $S(u, v)$ in the region between $C(t)$ and $O_d(t)$. Because the Gaussian curvature, $K(u, v)$, and the square of the mean curvature $H^2(u, v)$ can be

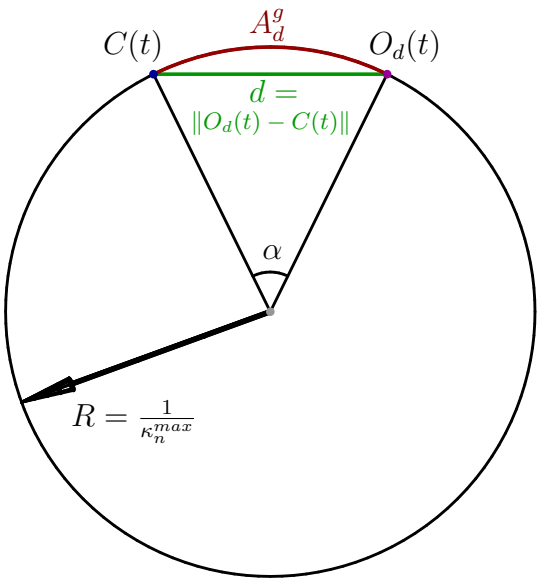

Figure 5: A cross section, orthogonal to curve $C(t)$, of a surface which is a sphere of radius $R = \frac{1}{\kappa_n^{max}}$.

computed as rational fields, if $S(u, v)$ is rational [9], one can use K and H^2 to establish bounds on the maximal normal curvature. A similar bound on the normal curvatures is also established in [13] using bounds over the coefficients of the first and second fundamental form fields that are also computed over the entire domain of surface $S(u, v)$.

Recall $\|C(t) - O_d(t)\| = d$. Given a maximal normal curvature, κ_n^{max}, in the region between $O_d(t)$ and $O_d^g(t)$, in $S(u, v)$, the following holds (See Figure 5):

Lemma 1. *The length of the geodesic arc, A_d^g, on $S(u, v)$ between $C(t)$ and $O_d(t)$ cannot exceed:*

$$A_d^g \leq \frac{2}{\kappa_n^{max}} \arcsin\left(\frac{d\kappa_n^{max}}{2}\right).$$

Proof: The normal curvature in the surface region of $S(u, v)$, between $C(t)$ and $O_d(t)$, is bounded by κ_n^{max}. Therefore, the worst deviation of $O_d(t)$ from $O_d^g(t)$ will occur if S assumes a normal curvature of κ_n^{max} everywhere (and in all tangential directions) on S, or S is a sphere of radius $\frac{1}{\kappa_n^{max}}$.

Now reconsider Figure 5. The spherical angle α between $C(t)$ and $O_d(t)$, in a maximal curvature sphere of radius $R = \frac{1}{\kappa_n^{max}}$ equals

$$\alpha = 2 \arcsin\left(\frac{d/2}{R}\right). \tag{4}$$

Then, A_d^g equals:

$$\begin{aligned} A_d^g &= 2\pi R \frac{\alpha}{2\pi} \\ &= R\alpha \\ &= 2R \arcsin\left(\frac{d/2}{R}\right) \\ &= \frac{2}{\kappa_n^{max}} \arcsin\left(\frac{d\kappa_n^{max}}{2}\right). \end{aligned} \tag{5}$$

■

Stated differently, the error of the presented approximation is the difference between the arc-length of a circular arc (A_d^g in Figure 5) and the length of a linear chord (d in Figure 5) connecting the end points of the arc. Interestingly and while $A_d^g \geq d$ always holds, the Taylor series of $\arcsin(x)$ is $x + \frac{x^3}{6} + O(x^5)$. Hence, for small offset or $d \to 0$, $\arcsin(x) \approx x$ and

$$A_d^g \approx \frac{2}{\kappa_n^{max}} \left(\frac{d\kappa_n^{max}}{2}\right) = d, \tag{6}$$

or the Euclidean offset approximation over a C^2 surface converges (from one side) to the geodesic offset, as $d \to 0$.

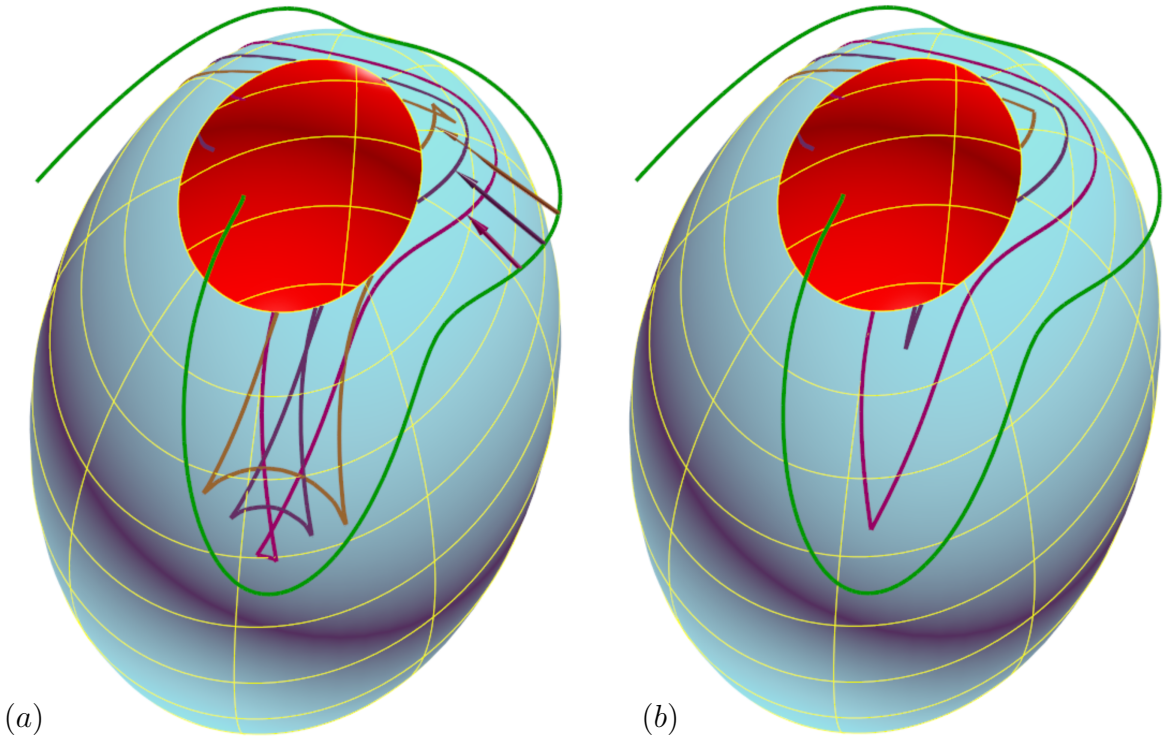

Figure 6: In (a), an example is presented of several offsets in increasing offset amount (in magenta, blue, and brown, in order), of a high degree 3-space curve (in green) *that is not on* $S(u, v)$, a bi-cubic with a control mesh of size (13×6). Herein, however, the results self-intersect. Note in (a), on top right, the three arrows depicting the respective offset amounts. In (b), the same example is presented after trimming the self-intersections in the offset curves. The brown offset, while completely eliminated in the front, has some small residue on the top of (b).

5. Handling Self-intersections

Much like planar offsets, self-intersections can occur in Euclidean (or geodesic) offsets over surfaces. Figure 6 (a) shows an example where several offsets self-intersect in the highly curved regions of $C(t)$ *that in this example is not on* $S(u, v)$, while mostly within distance d to S.

Quite a few methods to try and trim self-intersections in planar offsets are known. In [34], a self-intersecting trimming and elimination approach based on the Euclidean distance between $C(t)$ and $O_d(t)$ has been proposed. Because the approach of [34] to trim the self-intersections is Euclidean distance based, we can adapt it here. Reparameterize $O_d(t)$ using an independent parameter r as $O_d(r)$ and compute the bivariate function of $\delta(t, r) = \|O_d(r) - C(t)\|^2 - d^2$. If $O_d(t)$ was the precise offset of $C(t)$, by d, then any locations $\delta(t, r) < 0$ is a location that identifies $O_d(r)$ as too close to some $C(t)$ location and should be pruned. Given an offset approximation that is accurate to within ϵ, by computing all r parameters such that for an $O_d(r)$ location, there exists some $C(t)$ location that satisfies

$$\Delta(t, r) = \delta(t, r) + \epsilon < 0, \tag{7}$$

all offset self-intersections, local and global [34], can be identified, up to ϵ. In other words, the approach of [34] detects all self-intersection in Euclidean space, both local self-intersections due to the high curvature of $C(t)$ and global self-intersections that may result from independent different regions of $C(t)$ to intersect. One should note that using Equation (7) we are able to remove all regions (global and local) closer to $C(t)$ than d, up to some tolerance. The self-intersection double points are not identified precisely by Equation (7), but only clip the curve very close to them. One can apply a post process, if so desired, to numerically converge onto the precise double point of the intersection, while this post process has not been done in the results presented in this work. Figure 6 (b) shows the result from Figure 6 (a), after the proper trimming of the self-intersections.

6. Results

In this section, we present several examples of computing Euclidean offset curves over freeform surfaces (OCS).

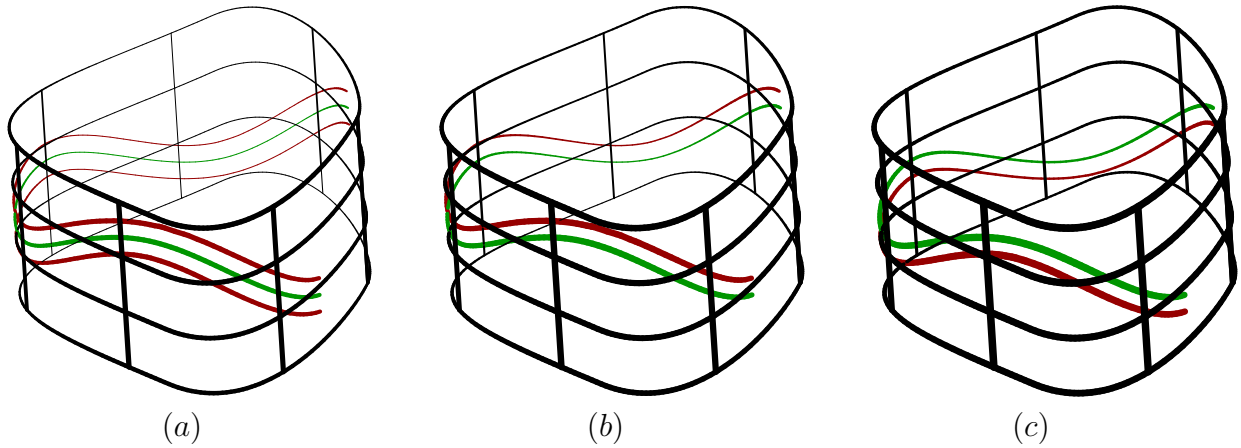

Figure 7: Simple Euclidean offsets (in red) of a curve (in green) on a freeform surface. In (a), offsets to both, left and right, directions are presented whereas in (b) and (c), inequality constraints (See Equation (3)) are used to filter only the left or only the right offset.

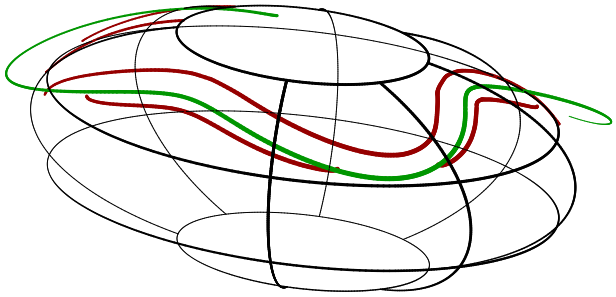

Figure 8: The Euclidean offset (in red) of a 3-space curve (in green) that is not on $S(u, v)$ is presented.

One advantage of the presented algorithm is its robustness. In Figure 6 (a), $C(t)$ is a degree 18 curve with 145 control points. $S(u, v)$, a bi-cubic. The degree of all the offset curves is 12, and the offset curves have between several dozens to several hundred control points. While the solution to Constraints (3) is low order in (u, v, t) space, the result must be composed with S to yield the Euclidean representation of the offset approximation. Because we precisely compute this (solution) curve-surface composition [6, 7], the result is of a relatively high degree. In spite of these high degrees, the end result is precise and stable.

Figure 7 shows a simple Euclidean offset of a curve on a freeform surface, of orders (2×4) and a control mesh of size (10×2). In Figure 7 (a), the Euclidean offsets to both, left and right, directions are shown using Constraints (1). In (b) and (c), inequality constraints (See Constraints (3)) are used to filter only the left or only the right offset. The degrees of the offset curves are 8 and 12 with hundreds of control points.

As already stated, the input curve, $C(t)$, need not be on the surface $S(u, v)$ and as long as the distance between $C(t)$ and $S(u, v)$ is less than the offset distance d, a valid Euclidean offset curve solution can be attained, on $S(u, v)$. In Figure 8, the Euclidean offset on S of a 3-space curve that is not on $S(u, v)$ (in green) is presented. In this example S is a bi-cubic surface with a control mesh of size (13×6) and the input and offset curves are of degrees 18 and 12, respectively.

These offset curves can serve as covering curves for $S(u, v)$ toward, for example, painting, inspection, CNC, or AM applications. Figure 9 shows an example of an initial small circle that is offset eleven times to yield a complete path that covers the surface, within distance d. In this example S is a bi-cubic surface with a control mesh of size (7×6) and the input and offset curves are of degrees 3 and 12, respectively.

By carefully tailoring the initial curve $C(t)$, a covering path for $S(u, v)$ might be made as one continuous path by spiraling through the (domain of the) surface. Figure 10 shows several examples of an initial helical curve $C(t)$ that is offset several times to yield a complete spiraling path that covers the respective surface, within distance d. In these examples, the surfaces are either bi-cubic or linear$\times$cubic and the input curves are of degrees 4 to 8 and the output offset curves are of degrees 8 and 12.

Our last examples in Figures 11 and Figures 12, show a curve $C(t)$ that is precisely on surface S with its offset(s). In Figure 11, the offset curve on the surface is shown before and after trimming, whereas in Figure 12, six consecutive offsets on S with self-intersections are presented. These self-intersections are then also shown after trimming. Both

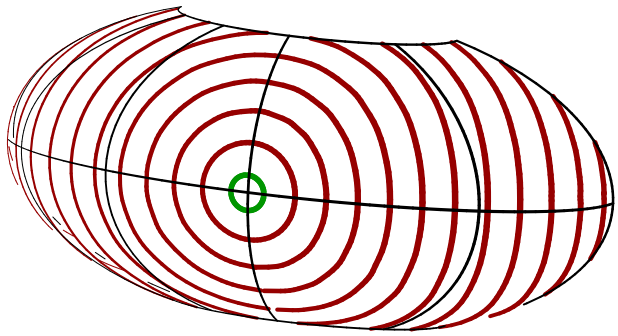

Figure 9: A covering univariate path for the surface is computed as a set of eleven parallel Euclidean offsets, within distance d.

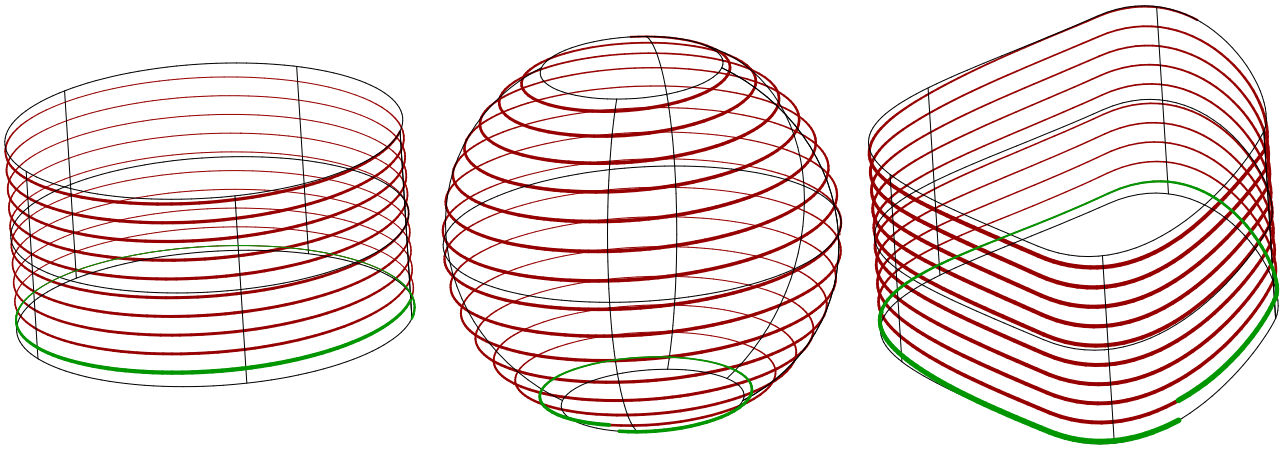

Figure 10: Three examples of spiraling paths that cover the surfaces. The initial helical curve $C(t)$ is shown in green whereas Euclidean offsets are successively computed throughout the surface to create these continuous spiraling covering curves.

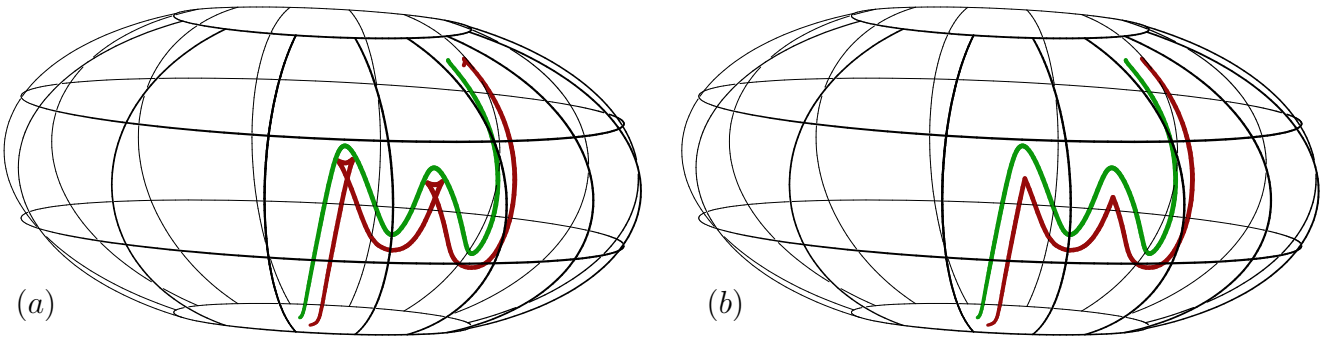

Figure 11: Another example of an offset (in red) of a 3-space curve (in green) on surface $S(u, v)$. Before self-intersections' elimination, in (a), and after the elimination of self-intersections, in (b).

examples employ the approach adapted from [34], as in Equation (7).

Finally, all presented examples, in this work, were computed in a few seconds on a modern Windows 10 computer. In fact, all the images in this paper were generated, accumulatively, in about one minute of computing time.

7. Extending the approach to Bisectors of Curves on Surfaces

Much like the way we compute the Euclidean offset approximation of curves over freeform surfaces, one can compute the Euclidean bisector approximation between curves that are all on some freeform surface, S. In [10], it was shown that the bisector surface of two rational space curves in $\mathbb{R}^3$ is, in general, rational as is the case for a point and a space curve in $\mathbb{R}^3$. Hence, herein, we can exploit these rational bisector surfaces and intersect them

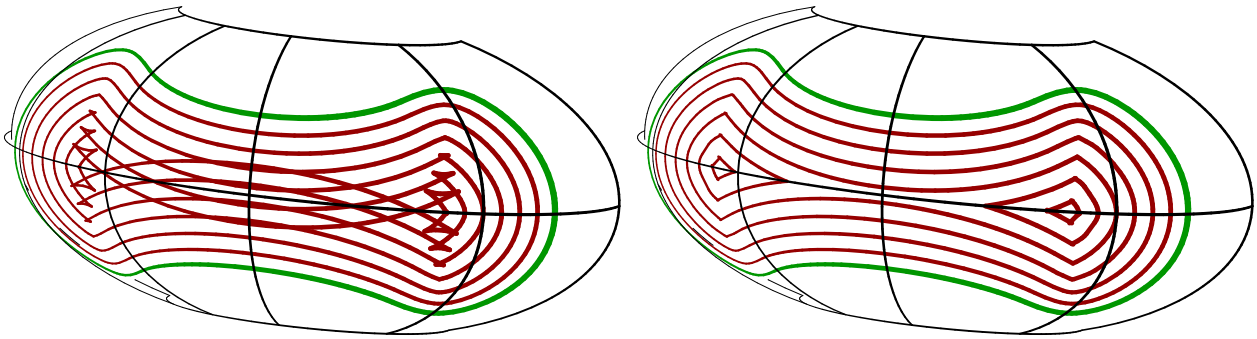

Figure 12: An example of trimming the self-intersections in the offsets (in red), of an input curve $C(t)$ (in green). (a) shows the untrimmed offset geometry whereas (b) shows the same geometry after trimming the self-intersections.

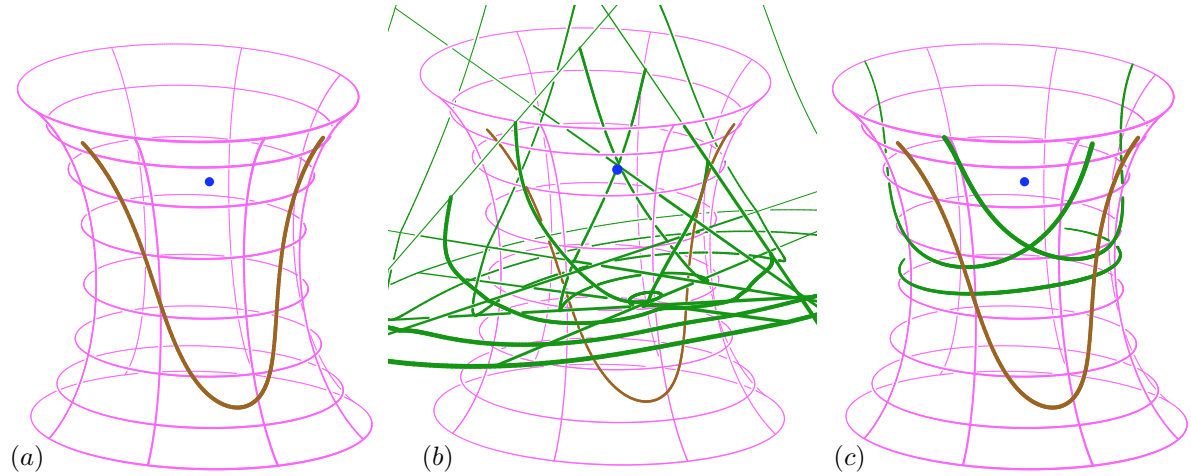

Figure 13: An example of a point-curve (in blue and yellow, respectively, in (a)) bisector (in green) in (c) over a freeform surface, S, computed by intersecting the 3-space bisector of the point and the curve (in green in (b), clipped to fit) with the input surface S (in magenta).

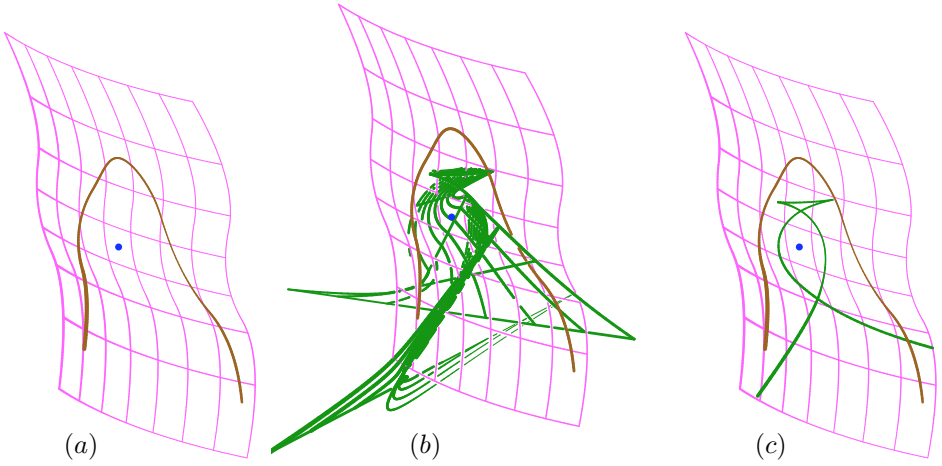

Figure 14: An example of a point-curve (in blue and yellow, respectively, in (a)) bisector (in green) in (c) over a freeform surface, S, computed by intersecting the 3-space bisector of the point and the curve (in green in (b)) with the surface S (in magenta).

against the surface S that the two input curves (or a curve and a point) are laying on, approximating the bisectors on S of the input, as Euclidean bisectors.

Figures 13 to 16 show a few examples. One should note that in all these examples, the full bisector arrangement is presented and a filtering post-process is likely to be required to extract the desired subset of the arrangement, as will be the case, for example, if Voronoi cells are sought.

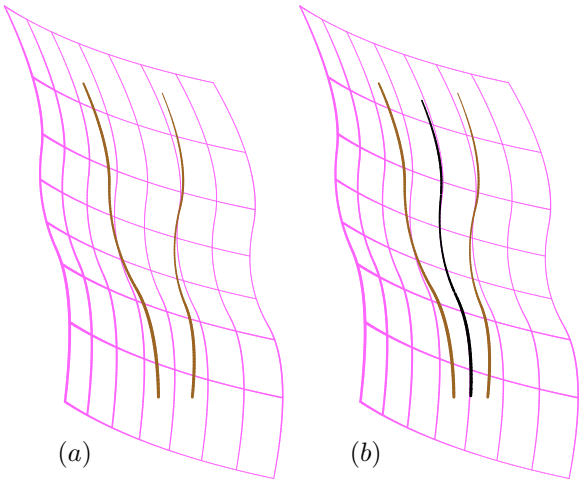

Figure 15: An example of a curve-curve bisector (in blue in (b)) over a freeform surface, S, computed by intersecting the 3-space bisector (that is not shown as it is too convoluted, having negative weights), with the surface S (in magenta). Input curves are shown in yellow.

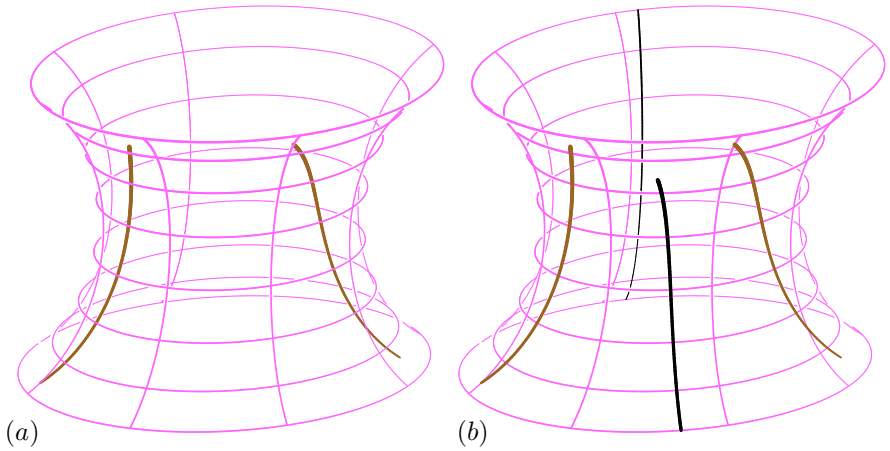

Figure 16: An example of a curve-curve bisector (in blue in (b)) over a freeform surface, S, computed by intersecting the 3-space bisector (that is not shown as it is too convoluted, having negative weights), with the surface S (in magenta). Note also the second bisector curve on S, on the back side of the surface. Input curves are shown in yellow.

Finally, the rational bisector surfaces, resulting from the bisector reconstruction presented in [10], frequently have negative weights and hence often present poles. This is not surprising considering the fact that these bisector sheets are unbounded at times. That said, the intersection of such unbounded rational bisector sheets with negative weights, with surface S, should be handled with care.

8. Future Work and Conclusions

An algorithm to compute the Euclidean offsets of freeform curves over freeform surfaces (OCS) has been presented. Bounds on the deviation from the precise geodesic offsets has been established based on curvature estimates. Further, an adaption of self-intersections elimination, of planar offsets, to the problem in hand, of offsets over freeform surfaces, was shown to be plausible. Finally, the BCS problem, or the problem of approximating bisector curves on freeform surfaces was also shown to be a possible extention of the same approach.

The proposed Euclidean approximation to offsets over freeform surfaces will suffer from low approximation quality, in highly curved surface regions or large offset distances. In the extreme cases, it might fail to find any approximation at all. Consider a cylinder S of diameter one and consider $C(t)$ on the cylinder, parallel to the axis of the cylinder. Any attempt to find a Euclidean offset approximation to $C(t)$ for $d > 1$ will yield no answer on S, while clearly the precise offset of $C(t)$ on S exists for all $d > 0$ (or at least $d < \pi$, if no periodic wrapping-around the cylinder is permissible). That said, this approximation error can be reduced (and controlled) by iteratively computing the offset curves, in several steps, and considering a fraction of the distance in each step.

Not every surface $S(u, v)$ and not every initial curve $C(t)$ (i.e. Figure 9) can yield a covering spiral path. Creating spiraling paths and detecting when spiraling paths are possible, toward a covering of the surface, up to a distance d,

using a single continuous curve, is left for further investigations.

Variable distance offsets [8] might also be desired in OCS. For example in CNC, locally adjusting the offset d based on the curvature of the machined surfaces and/or the cutting tool.

Trimming self-intersections in offsets is considered a highly challenging problem. More so herein, were we deal with offsets over freeform surfaces. The approach of [34], that was adapted in this work, needs to be further investigated due to the interplay between the offset approximation tolerance and the epsilon, ϵ, used in the $\Delta(r,t)$ function (Equation (7)). Too small of an ϵ and undesired, real offsets, might get trimmed. Too large of an ϵ and tiny self-intersections might be left intact. Similarly, self-intersection elimination in the constructed bisectors over surfaces is also desired, possibly following the lower envelope approach presented in [17].

In this work, we have established an upper bound on the relation between the Euclidean offset approximation computed herein and the surface geodesic offset. When combined with the sampling technique of the conventional methods [29, 38], we may be able to develop an approximation technique for the lower bound of the exact geodesic offset, which is the topic of our future work. Compared with the geodesic offset problem, the issue of upper and lower bounding for the geodesic bisector construction is considerably more difficult. Nevertheless, the upper and lower bounds for the geodesic offset may also provide a good solution to the geodesic bisector bounds.

Acknowledgments

We would like to thank our anonymous reviewers for their comments. This work was funded in part by the ISRAEL SCIENCE FOUNDATION (grant No. 597/18) and in part by the National Research Foundation of Korea (No. NRF-2019R1A2C1003490, NRF-2019K1A3A1A78112596).

References

[1] P.N. Atkar, A. Greenfield, D.C. Conner, H. Choset, and A.A. Rizzi. Uniform coverage of automotive surface patches. *The Int'l J. of Robotics Research*, 24 (11), 883–898, 2005.

[2] F. Aurenhammer: Voronoi diagrams – a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23 (3), 345–405, 1991.

[3] M. Bartoň, G. Elber, and I. Hanniel. Topologically guaranteed univariate solutions of underconstrained polynomial systems via no-loop and single-component tests. *Computer-Aided Design*, 43 (8), 1035–1044, 2011.

[4] J. Beck, R. Farouki, and J. Hinds. Surface analysis methods. *IEEE Computer Graphics & Applications*, 6 (12), 18–36, 1986.

[5] G. Brunnett. Geometric modeling of parallel curves on surfaces. *Computing*, Supplement **14**, pp. 37–53, 2001.

[6] T. DeRose, R. Goldman, H. Hagen, and S. Mann. Functional Composition via Blossoming. *ACM Trans. on Graphics*, 12 (2), 113–135, 1993.

[7] G. Elber. *Free Form Surface Analysis using a Hybrid of Symbolic and Numeric Computation.* PhD Thesis, University of Utah, 1992.

[8] G. Elber, E. Cohen. Error bounded variable distance offset operator for freeform curves and surfaces. *Int'l J. of Computational Geometry & Applications*, 1 (1), 67–78, 1991.

[9] G. Elber, and E. Cohen. Second-order surface analysis using hybrid symbolic and numeric operators. *ACM Trans. on Graphics*, 12 (2), 160–178, 1993.

[10] G. Elber, and M.-S. Kim. The bisector surface of rational space curves. *ACM Trans. on Graphics*, 17 (1), 32–49. 1998.

[11] G. Elber, I.-K. Lee, and M.-S. Kim. Comparing offset curve approximation methods. *IEEE Computer Graphics & Applications*, 17 (3), 62–71, 1997.

[12] G. Elber, and M.-S. Kim. Computing rational bisectors. *IEEE Computer Graphics and Applications*, 19 (6), 76–81. 1999.

[13] B. Ezair, and G. Elber. Using curvature bounds towards collision free 5-axis tool-paths. *Graphical Models*, Vol. 103, Article 101022, 2019.

[14] R. Farouki. *Pythagorean-Hodograph Curves*, Springer, Berlin, 2008.

[15] H.-Y. Feng and H. Li. Constant scallop-height tool path generation for three-axis sculptured surface machining. *Computer-Aided Design*, 34 (9), 647–654, 2002.

[16] D. Filip. Blending parametric surfaces. *ACM Trans. in Graphics*, 8 (3), 164–173, 1989.

[17] I. Hanniel, R. Muthuganapathy, G. Elber, and M. -S. Kim. Precise Voronoi Cell Extraction of Free-form Rational Planar Closed Curves. *The International Journal of Computational Geometry and Applications*, 17 (5), 453–496, 2007.

[18] M. Held. *On the Computational Geometry of Pocket Machining*, LNCS, Springer-Verlag, Vol. 500, 1991.

[19] V.D. Holla, K.G. Shastry, and B.G. Prakash. Offset of curves on tessellated surfaces. *Computer-Aided Design*, 35 (12), 1099–1108, 2003.

[20] Y.-J. Kim, J. Lee, M.-S. Kim, and G. Elber. Efficient offset trimming for planar rational curves using biarc trees. *Computer Aided Geometric Design*, 29 (7), 555–564, 2012.

[21] R. Kimmel, J. A. Sethian. Computing geodesic paths on manifolds. *Proceedings of the National Academy of Sciences of the United States of America, Applied Mathematics*, 95, 8431–8435, 1998.

[22] R. Kunze, F.-E. Wolter, and T. Rausch. Geodesic Voronoi diagrams on parametric surfaces. *Proc. Computer Graphics International 1997*, pp. 230–237, IEEE Press, Hasselt, Belgium, June 24–28, 1997.

[23] E. Lee. Contour offset approach to spiral toolpath generation with constant scallop height. *Computer-Aided Design*, 35 (6), 511–518, 2003.

[24] I.-K. Lee, M.-S. Kim, and G. Elber. Planar curve offset based on circle approximation. *Computer-Aided Design*, 28 (8), 617–630, 1996.

[25] J. Lee, Y-J. Kim, M.-S. Kim, and G. Elber. Efficient offset trimming for deformable planar curves using a dynamic hierarchy of bounding circular arcs. *Computer-Aided Design*, 58 (1), 248–255, 2015.

[26] Y.-J. Liu. Semi-continuity of skeletons in two-manifold and discrete Voronoi approximation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 37 (9), 1938–1944, 2015.
[27] T. Maekawa, N.M. Patrikalakis, T. Sakkalis, and G. Yu. Analysis and applications of pipe surfaces. *Computer Aided Geometric Design*, 15 (5), 437–458, 1998.
[28] T. Maekawa. An overview of offset curves and surfaces. *Computer-Aided Design*, 31 (3), 165–173, 1999.
[29] N.M. Patrikalakis, and L. Bardis. Offsets of curves on rational B-spline surfaces. *Engineering with Computers*, 5 (1), 39–46, 1989.
[30] B. Pham. Offset curves and surfaces: a brief survey. *Computer-Aided Design*, 24 (4), 223–229, 1992.
[31] H. Pottmann and M. Hofer. A variational approach to spline curves on surfaces. *Computer Aided Geometric Design*, 22 (7), 693–709, 2005.
[32] T. Rausch, F.-E. Wolter, and O. Sniehotta. Computation of medial curves on surfaces. *The Mathematics of Surfaces VII*, T. Goodman and R. Martin (Eds.), Information Geometers, pp. 43–68, 1997.
[33] R. Sarma and D. Dutta. The geometry and generation of NC tool path. *Trans. of the ASME: J. of Mechanical Design*, 119 (2), 253–258, 1997.
[34] J.-K. Seong, G. Elber, and M.-S. Kim. Trimming local and global self-intersections in offset curves/surfaces using distance maps. *Computer-Aided Design*, 38 (3), 183–193, 2006.
[35] K. Sureh and D.C.H. Yang. Constant scallop-height machining of free-form surfaces. *J. of Engineering for Industry*, 116 (2), 253–259, 1994.
[36] D.-E. Ulmet. Geodesic offsets of spline curves on spline surfaces: an industrial perspective. *Proc. of 24th National Conf. of Geometry and Topology*, Timisoara, Romania, July 5–9, 1994, pp. 263–274.
[37] H.-Y. Tam, H.-W. Law, abd H. Xu. A geometric approach to the offsetting of profiles on three-dimensional surfaces. *Computer-Aided Design*, 36 (10), 887–902, 2004.
[38] F.-E. Wolter and S. Tuohy. Approximation of high-degree and procedural curves. *Engineering with Computers*, 8 (2), 61–80, 1992.
[39] S.-Q. Xin, X. Ying, Y. He. Efficiently computing geodesic offsets on triangle meshes by the extended XinWang algorithm. *Computer-Aided Design*, 43 (11), 1468–1476, 2011.
[40] J. Xu, Y. Sun, and L. Zhang. A mapping-based approach to eliminating self-intersection of offset paths on mesh surfaces for CNC machining. *Computer-Aided Design*, 62, 131–142, 2015.
[41] J. Xu, Y. Wang, X. Zhang. and S. Chang. Contour-parallel tool path generation for three-axis mesh surface machining based on one-step inverse forming. *Proc. IMechE Part B: J. of Engineering Manufacture*, 227 (12), 1800–1807, 2013.