Computer-Aided Design 38 (2006) 183–193

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Trimming local and global self-intersections in offset curves/surfaces using distance maps

Joon-Kyung Seong [a], Gershon Elber [b,*], Myung-Soo Kim [c]

[a] School of Computing, University of Utah, Salt Lake City, UT, USA
[b] Computer Science Dept., Technion–Israel Institute of Technology, Haifa 32000, Israel
[c] School of Computer Science and Engineering, Seoul National University, Seoul, South Korea

Received 24 March 2005; accepted 7 August 2005

Abstract

A robust and efficient algorithm for trimming both local and global self-intersections in offset curves and surfaces is presented. Our scheme is based on the derivation of a rational distance map between the original curve or surface and its offset. By solving a bivariate polynomial equation for an offset curve or a system of three polynomial equations for an offset surface, all local and global self-intersection regions in offset curves or surfaces can be detected. The zero-set of polynomial equation(s) corresponds to the self-intersection regions. These regions are trimmed by projecting the zero-set into an appropriate parameter space. The projection operation simplifies the analysis of the zero-set, which makes the proposed algorithm numerically stable and efficient. Furthermore, in a post-processing step, a numerical marching method is employed, which provides a highly precise scheme for self-intersection elimination in both offset curves and surfaces. The effectiveness of our approach is demonstrated using several experimental results.

Keywords: Local self-intersection; Global self-intersection; Offset curve; Offset surface; Distance map; Zero-set computation

1. Introduction

Offsetting of curves and surfaces is one of the most important geometric operations in CAD/CAM due to its immediate applications in geometric modeling, NC machining, and robot navigation [10]. Exact offset curves and surfaces usually have algebraic degrees considerably higher than their original curves and surfaces. Furthermore, the offsets of rational curves or surfaces are non-rational in general. Hence, offset curves or surfaces are often approximated using rational curves or surfaces of relatively lower degree [4,9,14,15,17].

When $C(t)$ is given as a rational freeform curve, the exact offset curve $O_d(t)$ (with respect to an offset distance d) is defined by

$$O_d(t) = C(t) + N(t)d,$$

where $N(t)$ is the unit normal of $C(t)$. Since the unit normal vector $N(t)$ is non-rational in general, the exact offset curve $O_d(t)$ is non-rational. Thus, the offset curve is usually approximated by a rational curve, denoted as $O_d^\varepsilon(t)$, within a tolerance ε. An offset approximation within a tolerance ε means that $O_d^\varepsilon(t)$ lies inbetween two nearby offset curves, $O_{d+\varepsilon}(t)$ and $O_{d-\varepsilon}(t)$. More precisely, $O_d^\varepsilon(t)$ is contained in the region swept by a disk of radius ε moving along the exact offset curve $O_d(t)$. In this paper, the offset approximation is denoted as $O_d^\varepsilon(r)$; namely, it is parameterized independently of the original curve $C(t)$. Given a rational surface $S(u,v)$, its offset approximation surface is also represented as $O_d^\varepsilon(r,t)$.

Even after an offset curve or surface is approximated by a rational curve or surface, the detection and elimination of self-intersections is a difficult problem. Fig. 1 shows that the offset approximation may have self-intersections even though the original curve or surface has no self-intersection—an offset curve or surface has self-intersections *locally* due to regions of high curvature in the original curve or surface and the *global* self-intersection results from two different points of the curve or surface that are offset to the same location. These self-intersections must be detected and trimmed away to obtain a proper offset.

Even for offset curves [3,14], it is a non-trivial task to detect and trim all local and global self-intersections. Lee et al. [14] applied a plane sweep algorithm to detect all self-intersections

[*] Corresponding author. Tel.: +972 4 829 4338; fax: +972 4 829 3900.
E-mail addresses: seong@cs.utah.edu (J.-K. Seong), gershon@cs.technion.ac.il (G. Elber), mskim@cse.snu.ac.kr (M.-S. Kim).

0010-4485//$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2005.08.002

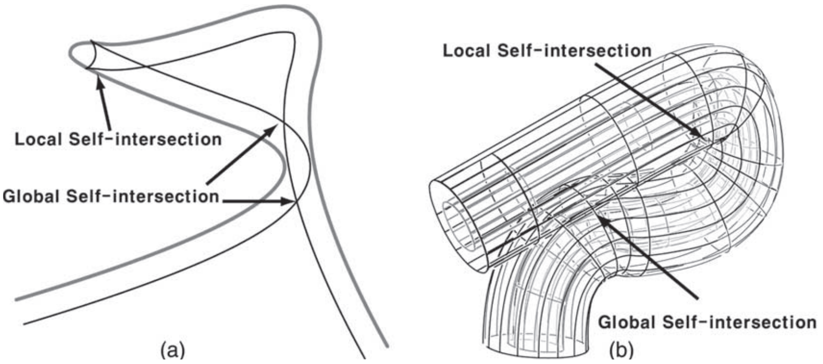

Fig. 1. A freeform curve (a) and a freeform surface (b) in gray and their offset approximations. Both local and global self-intersections occur in the offset curve and the offset surface.

of planar offset curves. Elber and Cohen [3] detected local self-intersections of offset curves by checking whether the tangent field of $C(t)$ and $O_d(t)$ have opposite directions. Whereas the plane sweep approach is difficult to implement, the tangent field approach of [3] is limited to detecting local self-intersections only. In a recent work, Elber [7] proposed an algorithm for trimming both local and global self-intersections of offset curves. In the current paper, the result of Elber [7] is extended to the general case of trimming local and global self-intersections of offset surfaces.

The problem of trimming self-intersections in offset surfaces is considerably more difficult to solve [1,2,13,16,21,22]. Cohen and Ho [2] introduced a trimming algorithm that is based on a necessary condition for a freeform surface to have self-intersections. This approach works for general surfaces. However, it is not an optimal solution for offset surfaces since no special consideration is taken into account for the relationship between the original surface and its offset. Wang [22] proposed an algorithm to compute the intersection curve between two offset surfaces, but his algorithm can deal with global self-intersections only. Aomura and Uehara [1] presented a similar approach based on the numerical integration starting from random initial points. Nevertheless, this method does not guarantee the detection of all the components of self-intersections. Maekawa et al. [16] presented a method for tracing self-intersection loops in the parameter domain. In their method, starting points are computed by solving a system of non-linear polynomial equations; but they are solving five equations in five variables and their algorithm requires special treatment for trivial solutions. Wallner et al. [21] considered the problem of computing the maximum offset distance that guarantees no local or global self-intersections. Elber and Cohen [3] detected local self-intersections of offset surfaces by considering the normal fields of a rational surface $S(u,v)$ and its offset surface $O_d(u,v)$ with respect to an offset distance d.

In this paper, a new method is proposed that simplifies the detection and trimming of offset self-intersections. Consider a disk (of radius d) moving along the original curve $C(t)$. If an offset curve point $O_d(r)$ is contained in an instance of the moving disk's interior, the point $O_d(r)$ belongs to a self-intersecting region (Fig. 2(a)). This geometric concept holds for both local and global self-intersections and it can be formulated algebraically as a rational distance map between the original curve or surface and its offset. If there is a curve point $C(t)$ within a distance d from the offset curve $O_d(r)$, the offset point $O_d(r)$ belongs to a self-intersecting region. In Fig. 2(b), an offset curve segment, which is inside the circle, belongs to a self-intersecting region since the distance between the circle center p and the offset curve segment is less than d. Thus, it is necessary to trim away such an offset curve segment $O_d(r)$. The same relation also holds for a surface point $S(u,v)$ and an offset surface $O_d(r,t)$.

Let $\mathcal{D}(r,t)$ be the distance function between the original curve $C(t)$ and its offset curve $O_d(r)$. Then, the solution set for the inequality condition $\mathcal{D}(r,t)-d<0$ corresponds to the self-intersection regions in the offset curve. By projecting the solution set onto the r-axis, i.e. the parameter space of the offset curve $O_d(r)$, all self-intersections in the offset curve $O_d(r)$ can be detected. The projected region of the solution set is a union of open intervals on the r-axis; when end points to these open intervals are added, the resulting union of closed intervals is exactly the same as the projection of the zero-set of $\mathcal{D}(r,t)-d=0$. Therefore, the problem of trimming offset self-intersections can be reduced to that of finding the zero-set of a single polynomial equation in an rt parameter space. A variety of geometric problems involving freeform curves

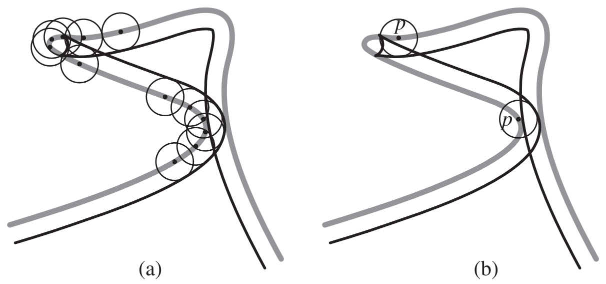

Fig. 2. (a) The self-intersecting region in an offset curve is contained in a swept region of a circle. (b) It can be represented by using the distance between an original curve (shown in gray color) and its offset curve (shown in bold lines). An offset curve segment, which is inside the circle, belongs to a self-intersecting region since the distance between the circle center p and the offset curve segment is less than the offset distance or the radius of the circle.

or surfaces can be reduced to the single question of finding the zero-set of a system of non-linear polynomial equations in the parameter space of the original curves or surfaces. In other work, we presented several algorithms that are based on the similar reduction schemes to parameter space [8,18,19]. These include computing the convex hull of freeform curves or surfaces [8,18], computing bisector curves or surfaces [5], constructing sweep envelopes, and intersecting a freeform surface with a sweep surface [19]. The trimming algorithm employed in this paper also operates on the same premise as taken in [12].

The trimming algorithm can easily be extended to offset surfaces. Similar to the curve case, the distance function $\mathcal{D}(u, v, r, t)$ between a freeform surface $S(u,v)$ and its offset surface $O_d(r,t)$ can be derived as a four-variate rational function. In this case, the solution set for $\mathcal{D}(u, v, r, t) - d < 0$ corresponds to the self-intersection of an offset surface. Let R denote the projected region of the solution set onto the rt-plane. Then, the offset surface patches $O_d(r,t)$ corresponding to the parameter domain R are self-intersecting regions. The boundary of this region R can be determined by solving three polynomial equations: $\mathcal{D}(u, v, r, t) - d = 0$, $\mathcal{D}_u(u, v, r, t) = 0$ and $\mathcal{D}_v(u, v, r, t) = 0$, where $\mathcal{D}_u$ and $\mathcal{D}_v$ are the u- and v-partial derivatives of $\mathcal{D}$.

The topological configuration of the zero-set may be arbitrarily complex depending on the shape of the original curve or surface and the offset distance. However, the projection operation of the trimming algorithm considerably simplifies the topological analysis of the zero-set since it reduces the dimensionality of the problem. Consider the curve case with a parameter interval I in the r-axis, which is the projection of the zero-set of $\mathcal{D}(r, t) - d = 0$. Viewed along the t-direction, there may be multiple zero-set curve segments appearing over the interval I. Nevertheless, no matter how complex the zero-set curves are, the interval I is characterized by its end points. Moreover, the algorithm for trimming offset surfaces reduces a four-dimensional problem into that of contouring curves in the rt-plane. This step greatly simplifies the trimming procedure. Hence, it is easy to implement our algorithm in a numerically stable and efficient way. Furthermore, to get highly precise self-intersection points and curves, a post-processing step is applied, where numeric marching methods are employed. A few steps of the Newton–Raphson iterations are sufficient to achieve precise self-intersections both in offset curves and surfaces. Several experimental results show the effectiveness of this approach.

The main contribution of our work can be summarized as follows:

- The problem of trimming self-intersections is simplified by using a rational distance function and a projection operation;
- The difficult problem of trimming self-intersections in offset curves and surfaces is reduced to the relatively easier problem of finding the zero-set of a single polynomial equation or a system of three polynomial equations, respectively;
- A highly precise self-intersection trimming is achieved by applying a post-process based on the numeric marching steps.

The rest of this paper is organized as follows. In Section 2, the trimming algorithm is discussed for the self-intersection of an offset curve, an approach that is based on a distance function computation. Section 3 presents its extension to the offset surface. In Section 4, the topological issue of the proposed trimming approach based on the characteristics of the zero-set of the constraint equations is considered. In Section 5, a method for numerical improvement is addressed. Some examples are presented in Section 6, and finally, in Section 7, this paper is concluded.

2. Trimming self-intersections in offset curves

In this section, the process of trimming self-intersections in the offset approximation of a freeform rational curve in the plane is considered. The algorithm of Elber [7] is briefly summarized and is supplemented with some new results.

Given a rational offset approximation of a rational curve $C(t)$ by a distance d, $O_d^\varepsilon(r)$, where $\varepsilon > 0$ denotes the accuracy of approximation, consider a squared distance function,

$$\Delta_d^\varepsilon(r, t) = <C(t) - O_d^\varepsilon(r), C(t) - O_d^\varepsilon(r)> .$$

If no self-intersection occurs in $O_d^\varepsilon(r)$, then $\Delta_d^\varepsilon(r, t) \geq (d - \varepsilon)^2$, $\forall (r, t)$. In contrast, if $O_d^\varepsilon(r)$ is self-intersecting, then there exist points in $O_d^\varepsilon(r)$ that are closer than $d - \varepsilon$ to $C(t)$. Therefore, any pair of points $O_d^\varepsilon(r)$ and $C(t)$ such that $\Delta_d^2(r, t) < (d - \varepsilon)^2$ implies that there is a self-intersection.

Let $\rho > 0$ be another small positive real value and let

$$F(r, t) = \Delta_d^\varepsilon(r, t) - (d - \varepsilon - \rho)^2. \qquad (1)$$

Any point (r_0, t_0) in the zero-set of $F(r,t) = 0$ represents two points, $C(t_0)$ and $O_d^\varepsilon(r_0)$, that are $(d - \varepsilon - \rho)$ apart. Every such point $O_d^\varepsilon(r_0)$ must be purged away as a self-intersecting point. Therefore, the set of offset curve points that are free from self-intersections is contained in

$$\mathcal{K} = \{O_d^\varepsilon(r) | F(r, t) > 0, \forall t\}.$$

In other words, if a point r on the r-axis is on the projection of the zero-set of $F(r,t) = 0$, then the corresponding offset curve point $O_d^\varepsilon(r)$ belongs to the self-intersecting region. This trimming process is denoted (using a small positive trimming distance ρ below the offset approximation) a ρ-accurate trimming or ρ-trimming, for short. Below, the algorithm is presented that detects and eliminates the self-intersection regions:

Algorithm 1.

```
Input:
    $C(t)$, a rational curve;
    $O_d^\varepsilon(r)$, a rational approximation offset of $C(t)$ by distance $d$ and tolerance $\varepsilon$;
    $\rho$, a trimming distance for self-intersections.
```

```
Output
    A piecewise rational approximation offset of $C(t)$ by distance $d$, tolerance $\varepsilon$, and $\rho$-trimming;
Begin
    $\Delta_d^\varepsilon(r,t) \Leftarrow \langle C(t)-O_d^\varepsilon(r), C(t)-O_d^\varepsilon(r)\rangle$;
    $F(r,t) \Leftarrow \Delta_d^\varepsilon(r,t)-(d-\varepsilon-\rho)^2$;
    $\mathcal{Z} \Leftarrow$ the zero-set of $F(r,t)=0$;
    $\mathcal{Z}_r \Leftarrow$ the projection of $\mathcal{Z}$ onto the $r$-axis;
    return the $r$ domain(s) of $O_d^\varepsilon(r)$ not included in $\mathcal{Z}_r$;
End.
```

Since $F(r,t)$ is a piecewise rational bivariate function, the zero-set, $\mathcal{Z}$, could be constructed by exploiting the convex hull and subdivision properties of NURBS, yielding a highly robust divide-and-conquer zero-set computation that is reasonably efficient [6]. The parameter value r_0, such that $C(t)$ is closer to $O_d^\varepsilon(r_0)$ than the distance $(d-\varepsilon-\rho)$, for some t, implies that the curve offset point $O_d^\varepsilon(r_0)$ belongs to a self-intersecting region. Hence, $\mathcal{Z}$ is projected onto the r-axis, as $\mathcal{Z}_r$. The domain of the r-axis covered by this projection prescribes the regions of $O_d^\varepsilon(r)$ that present self-intersections and thus must be purged away.

Fig. 3 presents the example from Fig. 1(a) again. In Fig. 4, the log of the squared distance function, $\Delta_d^\varepsilon(r,t)$, is presented for the curve of Fig. 3. Also shown in Fig. 4 is the zero-set, $\mathcal{Z}$, of $F(r,t)=\Delta_d^\varepsilon(r,t)-(d-\varepsilon-\rho)^2$ and its projection, $\mathcal{Z}_r$, on the r-axis. In this case, the r-axis is divided into four valid intervals, which characterize three sub-regions of the offset curve that are self-intersecting. The first and third bold intervals along the r-axis are due to the global self-intersection of the curve in Fig. 3, whereas the middle large thick interval corresponds to the local self-intersection in the curve. Fig. 3(b) shows the ρ-trimmed curve segments that are computed by extracting four valid intervals.

The projection of the zero-set, $\mathcal{Z}$, should be onto the r-axis. If the zero-set is projected onto the t-axis, which is the parameter space of the original curve $C(t)$, then invalid trimming of a valid parameter interval may occur on the curve $C(t)$. Considering the trimming process as a continuous sweeping of a disk (see Fig. 2(a)), it is necessary to sweep the disk along the original curve. Moving a disk along the offset curve $O_d^\varepsilon(r)$ may cut away some original curve regions that present no self-intersections. A projection of the zero-set onto the t-axis will find all locations in $C(t)$ that are at a distance $(d-\varepsilon-\rho)$ from a point $O_d^\varepsilon(r)$. In symmetry, however, the projection onto the r-axis finds all locations in $O_d^\varepsilon(r)$ that are within the distance $(d-\varepsilon-\rho)$ from a point $C(t)$. Fig. 5 shows such an example. After the offset operation, there are two valid curve segments in $O_d^\varepsilon(r)$ (Fig. 5(b)). Fig. 6 presents the distance function, $\Delta_d^\varepsilon(r,t)$ in gray, and the zero-set, $\mathcal{Z}$, in bold lines. The projection of $\mathcal{Z}$ onto the t-axis generates only a single valid interval, which means that one of the valid regions of $C(t)$ is trimmed away. On the other hand, the projection of the zero-set onto the r-axis produces three valid intervals. Two of them are connected since the curve is a closed one, and thus, the valid intervals under this projection yield two curve segments as shown in Fig. 5(b).

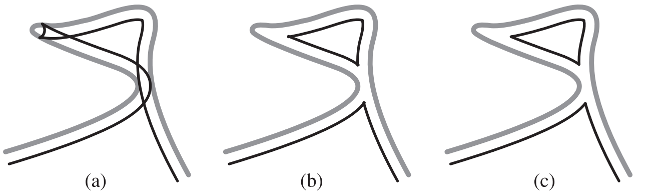

Fig. 3. (a) The simple curve and its offset, from Fig. 1(a). The curve and its offset are presented in (a). (b) is the result of ρ-trimming the curve using $\Delta_d^\varepsilon(r,t)$ whereas, in (c), the result is improved using numerical marching.

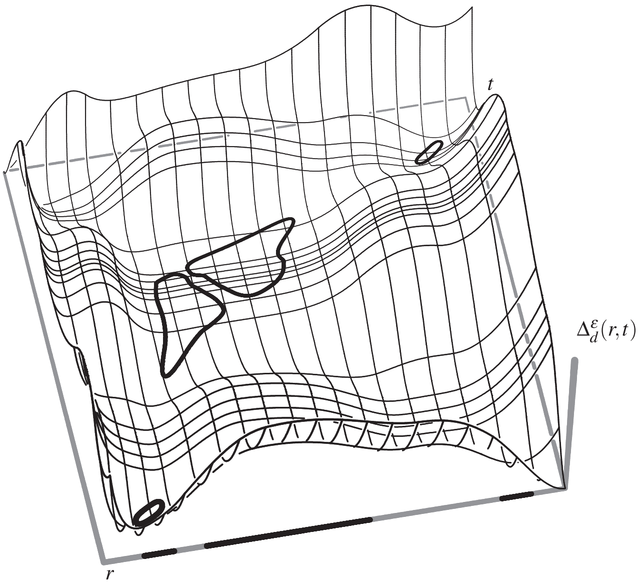

Fig. 4. The distance function, $\Delta_d^\varepsilon(r,t)$, of the curve in Fig. 3, on a logarithmic scale. Also shown, in bold lines, are the zero-set, $\mathcal{Z}$ of $F(r,t)=\Delta_d^\varepsilon(r,t)-(d-\varepsilon-\rho)^2$ as well as the projection of $\mathcal{Z}$ on the r-axis, $\mathcal{Z}_r$.

A point $O_d^\varepsilon(r_0)$ is called a *match to a point* $C(t_0)$ if it is the point offset from $C(t_0)$. If we use $\rho=0$ in the definition of the function $F(r,t)$, then the trimming process becomes numerically unstable as the matched point can be a trivial solution of the equation $F(r,t)=0$. Thus, a positive value of ρ needs to be exploited to compute the self-intersections somewhat conservatively. For numerical stability, ρ should be as large as possible, reducing the chance of detecting the matched points as self-intersections. Nonetheless, and while ρ should always be positive, as ρ approaches zero, it is going to be more difficult to robustly evaluate the zero-set of $F(r,t)=0$. While describing the details of a zero-set finding is beyond the scope of this work, it is clear that, the better the zero-set funding is, the higher the quality of the offset trimming will be. In contrast, the

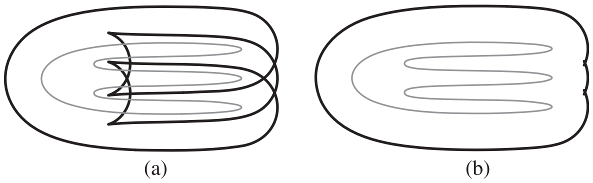

Fig. 5. An example that justifies the projection of the zero-set onto the r-axis.

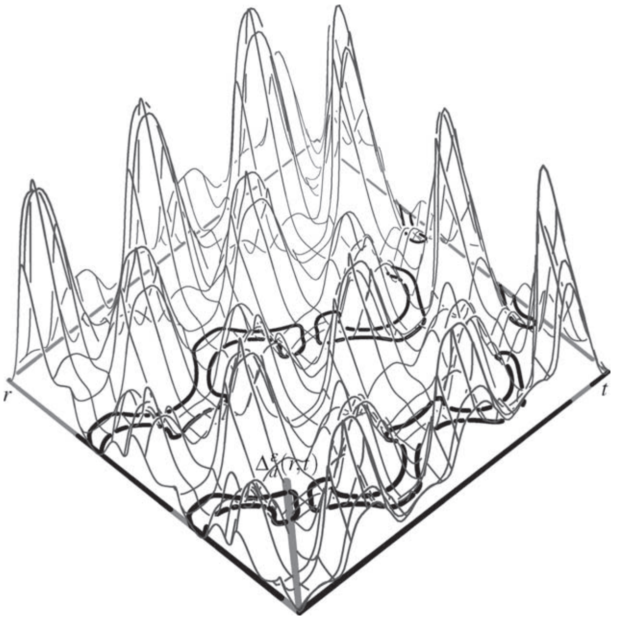

Fig. 6. The distance function, $\Delta_d^\varepsilon(r,t)$, of the curve in Fig. 5, on a logarithmic scale. Also shown, in bold lines, are the zero-set, $\mathcal{Z}$ of $F(r,t)=\Delta_d^\varepsilon(r,t)-(d-\varepsilon-\rho)^2=0$, as well as the projection of $\mathcal{Z}$ on the r-axis and t-axis.

larger ρ is, the higher the likelihood that small self-intersections will be missed. In practice, ρ was selected to be between 1% and 5% of the offset distance d. Another crucial characteristic of ρ is that using the function $F(r,t)$ all self-intersections at distance greater than ρ from the trimmed offset curve can be detected, where the function $F(r,t)$ yields negative values. The self-intersecting offset curve segments are extracted by the zero-set finding routine.

The outcome of Algorithm 1 is a subset of $O_d^\varepsilon(r)$. The latter comprises curve segments that have no point closer than $(d-\varepsilon-\rho)$ to $C(t)$. By selecting $\rho>0$, the curve segments that result from Algorithm 1 are not exactly connected but slightly cross each other. A sequence of numeric Newton–Raphson marching steps at each such pair of trimmed end points could very quickly converge to the exact self-intersection location. Only a few steps are required to converge to the highly precise self-intersection location. Fig. 3(c) shows the result of the numerical improving step. In Section 6, several other examples are presented that demonstrate this entire procedure, including the aforementioned numerical marching step.

3. Trimming self-intersections in offset surfaces

The trimming algorithm for offset curves is now extended to the case of offset surfaces. A squared distance function between a freeform surface $S(u,v)$ and its offset approximation, $O_d^\varepsilon(r,t)$, is derived. The function becomes a four-variate one and hence, it ends up dealing with a three-dimensional zero-set in a four-dimensional parameter space. The projection of the zero-set onto the rt-space prescribes the self-intersection regions in the offset surfaces. The main difference of this extension from ρ-trimming of a planar curve is that the boundary of the projected zero-set is now directly extracted by considering two more constraints. The same thing can be done in the curve case as well.

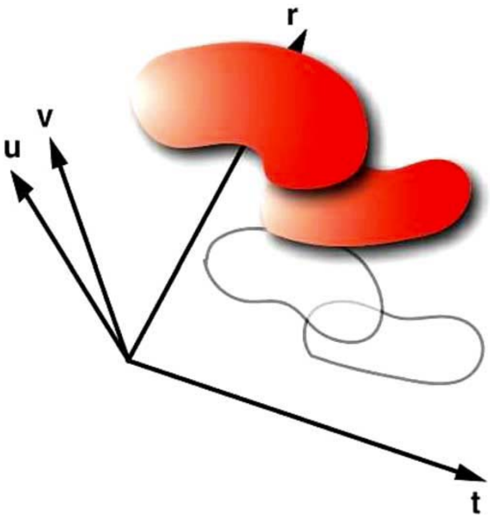

Fig. 7. A zero-set of Eq. (2) in the $uvrt$-space and its projected uv-silhouettes onto the rt-space. The zero-set components overlap along the uv-direction and cause the intersection of its projected silhouettes.

Consider the squared distance function:

$$\Delta_d^\varepsilon(u,v,r,t) = <S(u,v)-O_d^\varepsilon(r,t), S(u,v)-O_d^\varepsilon(r,t)> .$$

Assume that ρ represents a trimming distance and let

$$F(u,v,r,t) = \Delta_d^\varepsilon(u,v,r,t)-(d-\varepsilon-\rho)^2.$$

An offset surface point $O_d^\varepsilon(r_0,t_0)$ belongs to a self-intersecting region if there exists another point $S(u_1,v_1)$ satisfying the following inequality condition:

$$F(u_1,v_1,r_0,t_0)\le 0.$$

Similar to the curve case, the set of offset surface points that are free of self-intersections is contained in

$$\mathcal{T} = \{O_d^\varepsilon(r,t)|F(u,v,r,t)>0,\forall(u,v)\}.$$

As mentioned above, the solution set of the condition $F(u,v,r,t)\le 0$ in the $uvrt$-domain is involved in the self-intersections and thus, the corresponding offset points $O_d^\varepsilon(r,t)$ should be trimmed away. In other words, if the point (r,t) in the parameter domain falls into the projection of this solution set, then the corresponding offset surface point $O_d^\varepsilon(r,t)$ belongs to the self-intersecting region. The boundary of the 'uncovered' region of the rt-plane (under this projection) is characterized as the projection of the uv-silhouette curves (along the uv-direction) of the zero-set of the equation $F(u,v,r,t)=0$. This means that the u-partial derivative and v-partial derivative must simultaneously vanish along the silhouette curve, which can be characterized as the simultaneous satisfaction of the following three constraints in the $uvrt$-space:

$$F(u,v,r,t) = 0, \quad (2)$$

$$F_u(u,v,r,t) = 0, \quad (3)$$

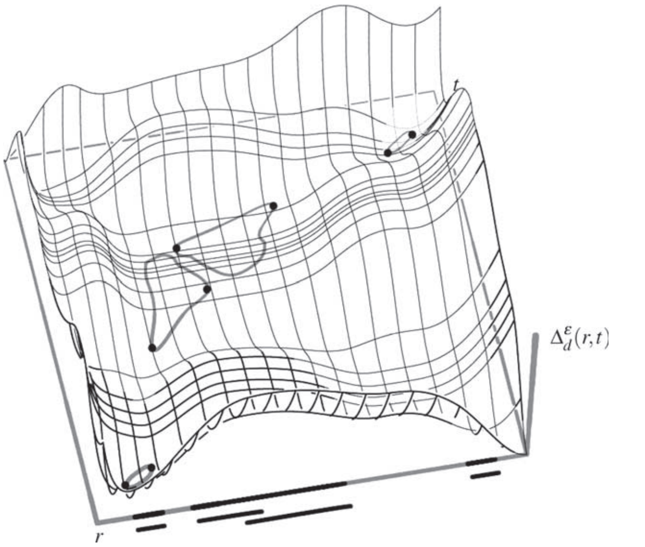

Fig. 8. The same example as in Fig. 4. The zero-set $\mathscr{Z}$ is shown in thick gray and the t-silhouette points of the zero-set curves are presented as thick dots.

$$F_v(u,v,r,t) = 0. \qquad (4)$$

Having three equations (2)–(4) in four variables, the solution is a univariate curve that is a simultaneous solution in the *uvrt*-space. In this work, we exploit a spline subdivision-based method for solving the set of equations represented as scalar spline functions [6]. The zero-set in arbitrary dimension could be computed by extending the Dual Contouring method into an arbitrary dimension [20].

The univariate solution curve can be parameterized by some variable α:

$$(u(\alpha), v(\alpha), r(\alpha), t(\alpha)).$$

Then, this set of curves forms the trimming curves for the offset surfaces that are self-intersection-free up to ε and ρ. Algorithm 2 summarizes the overall procedure:

Algorithm 2.

```
Input:
    $S(u,v)$, a rational freeform surface;
    $O_d^{\varepsilon}(r,t)$, a rational approximation offset of $S(u,v)$ by distance $d$ and tolerance $\varepsilon$;
    $\rho$, a trimming distance for the self-intersections.
Output:
    A piecewise rational approximation offset of $S(u,v)$ by distance $d$, tolerance $\varepsilon$, and $\rho$-trimming;
Begin
    $\Delta_d^{\varepsilon}(u,v,r,t) \Leftarrow \langle S(u,v) - O_d^{\varepsilon}(r,t), S(u,v) - O_d^{\varepsilon}(r,t) \rangle$;
    $F(u,v,r,t) \Leftarrow \Delta_d^{\varepsilon}(u,v,r,t) - (d - \varepsilon - \rho)^2$;
    $G(u,v,r,t) \Leftarrow F_u(u,v,r,t)$;   $H(u,v,r,t) \Leftarrow F_v(u,v,r,t)$;
  (1) $\mathscr{Z} \Leftarrow$ the simultaneous zero-set of $F$, $G$, and $H$;
```

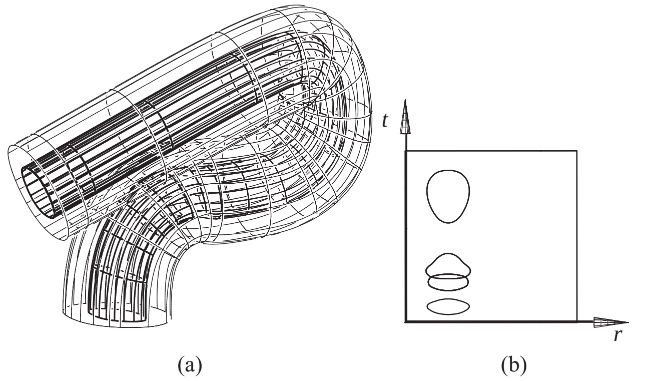

Fig. 9. (a) A rational freeform surface (shown in bold lines) and its untrimmed offset surface (shown in light lines). (b) The corresponding *rt*-projection of the zero-set of distance square function is shown as a set of curve segments. One can see the overlapping trimming curve segments in local self-intersections.

```
    return offset approximation trimmed by $\mathscr{Z}$;
End.
```

Step (1) in Algorithm 2 can also employ the original trimming curves of $S(u,v)$, if to begin with, $S(u,v)$ is a trimmed surface. The outcome of Algorithm 2 is a trimmed offset surface approximation $O_d^{\varepsilon}(r,t)$. It comprises surface patches that have no point closer than $(d - \varepsilon - \rho)$ to $S(u,v)$.

4. Topology of trimming curves in offset surfaces

In Section 3, we presented an algorithm for determining a boundary of the projected zero-set of Eq. (2) in the *rt*-plane by computing *uv*-silhouette curves (along the *uv*-direction) of the zero-set. The *uv*-silhouette curves in the *rt*-plane may intersect each other; namely, a connected component of the zero-set of Eq. (2) is partially blocked (along the *uv*-direction) by another one in the *uvrt*-space in the projection along the *uv*-direction (see Fig. 7). The intersections among *uv*-silhouette curves should be detected and resolved for a proper trimming. In the current problem, it is necessary to extract only the outmost boundary of the projected region, which makes the overall procedure numerically stable.

For the clarity of presentation, we first consider the possible topological problem using a simple low-dimensional case of

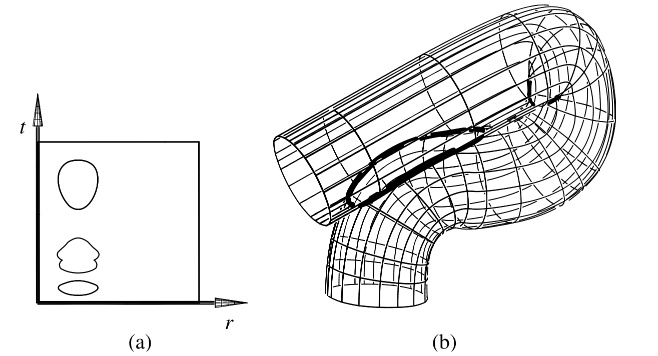

Fig. 10. Overlapping in local self-intersections is resolved in (a) and the resulting trimmed offset surface is shown in (b). Compare with Fig. 9.

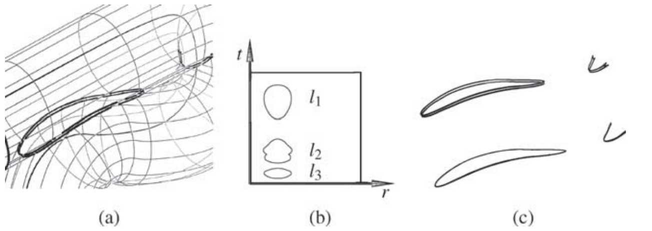

Fig. 11. (a) A zoom-in in Fig. 10 that shows the trimmed self-intersection curves in detail. (b) shows the three trimming curves in the *rt*-plane. (c) The trimmed self-intersection curves (upper ones) can be improved to the highly precise intersection curves (lower ones) after a post-process of numerical marching step.

trimming offset curves. Fig. 8 shows the same squared distance function of Fig. 4; the zero-set of Eq. (1) is drawn in gray. The projection of the zero-set onto the *r*-axis is determined by the *t*-silhouette points at which the *t*-partial derivatives of $F(r,t)$ vanish. The *t*-silhouette points are represented in bold dots in Fig. 8. Viewed along the *t*-direction, the two zero-set components overlap. They are located in the middle part of the *rt*-space. The intervals resulting from these two zero-set components correspond to local self-intersections in the offset curve. Since local self-intersections are due to the regions of high curvature in $C(t)$, there always exists an offset curve point $O_d^{\varepsilon}(r)$ in the local self-intersection region that has more than two points of $C(t)$ that are closer than $(d-\varepsilon-\rho)$ to $O_d^{\varepsilon}(r)$. This means that sub-regions made from the zero-sets of Eq. (1) always overlap in the local self-intersection region (see two intervals that are the projections of two zero-set curves in the middle part of Fig. 8).

In contrast to the local self-intersections, global self-intersections do not always introduce overlaps in the zero-set. Two zero-set components that correspond to the same global self-intersection region in $O_d^{\varepsilon}(r)$ come as a pair. They are called a matching pair. In Fig. 8, two zero-set components that are located in the corner parts of the *rt*-space correspond to the global self-intersection. Two intervals resulting from the matching pair do not overlap in this case.

In the case of offset surfaces, the situation is far more complex since the overlapping of the zero-set components in four-dimensional space needs to be considered. The topological structure of the problem, however, is almost identical to the case of offset curves. The *uv*-silhouette curves in the *rt*-space, which are boundary curves of the projected region of the zero-set, are intersecting each other when they correspond to the local self-intersection region. Fig. 9(a) shows a freeform rational surface and its offset approximation. Fig. 9(b) shows the projection of *uv*-silhouette curves of the zero-set of Eq. (2) onto the *rt*-space. In the middle part, there are two intersecting loops that correspond to the local self-intersection. For a proper trimming, we should form a union of the domains bounded by two overlapping loops into a single loop by applying a planar Boolean union operation in the *rt*-parameter domain. Fig. 10(a) shows the resulting trimming loop.

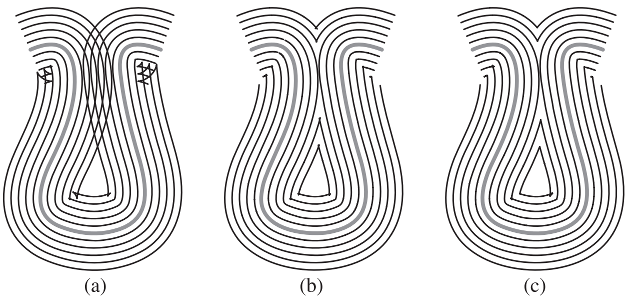

Fig. 12. An example of a curve (in gray) and its untrimmed offsets are shown in (a). (b) is the result of ρ-trimming the offset curves using $\Delta_d^{\varepsilon}(r,t)$ and (c) is the result of applying the post-process numerical marching steps.

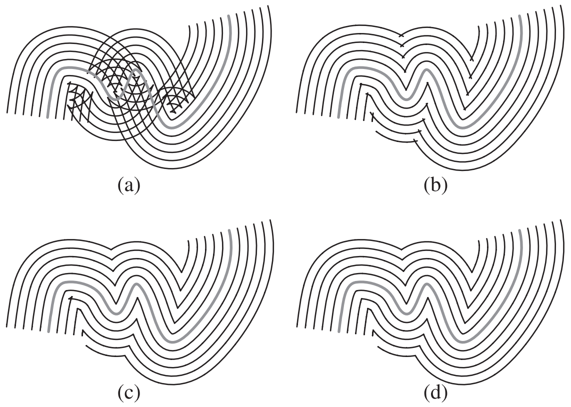

Fig. 13. (a) presents the original curve (in gray) and its untrimmed offsets, while (b) is the result of ρ-trimming the self-intersections with the aid of the $\Delta_d^{\varepsilon}(r,t)$ function. (b) and (c) are the results of ρ-trimming of the offset with two different ρ-trimming percentages of 5% and 1%, respectively. (d) is the result after the post-process of numerical marching steps.

In summary, since our approach to trimming self-intersections is based on the distance function, the topological

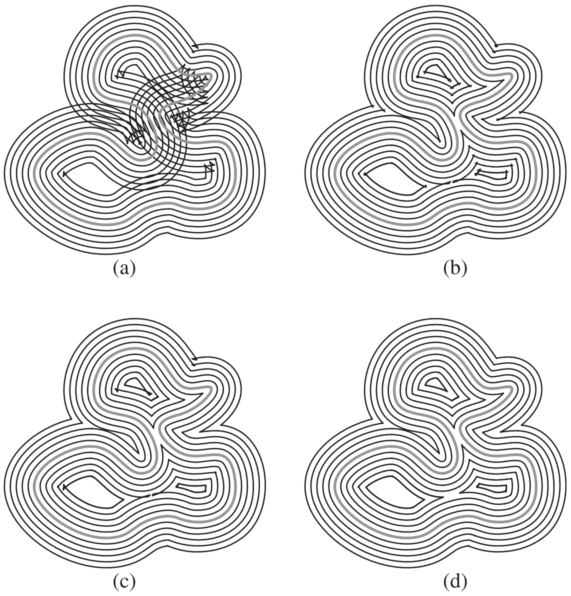

Fig. 14. (a) presents the original curve (in gray) and its offsets, while (b) is the result of ρ-trimming the self-intersections using the function $F(r,t)$ of Eq. (1). (b) and (c) are the result of ρ-trimming of the offset with two different ρ-trimming percentages of 5 and 1%, respectively. (d) is the result after the post-process of numerical marching steps.

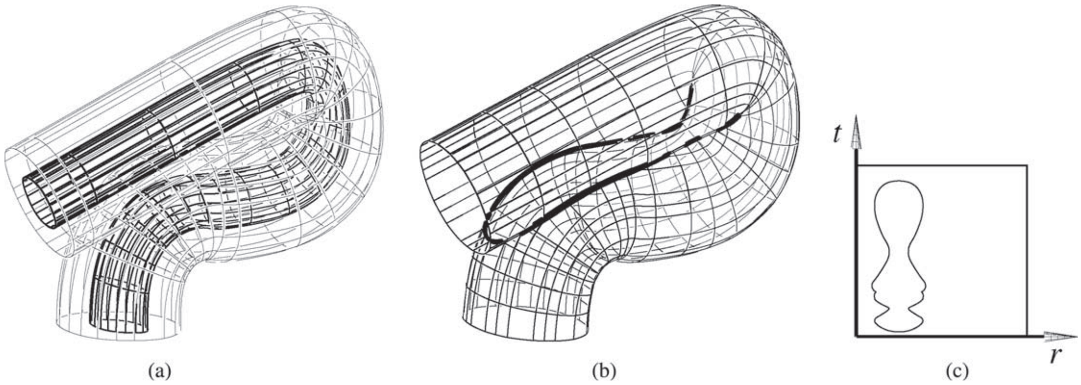

Fig. 15. The topology of trimming curves can change as an offset distance changes. Here, the global and local self-intersections of Fig. 10 are merged to form a single component.

complexity can be handled in a simpler and more flexible way. Although the structure of the zero-set in two-dimensional space (offset curves) or four-dimensional space (offset surfaces) may be arbitrarily complex, the presented scheme only requires forming a union of the projected regions of the zero-set if they overlap. The topological problem is far more complex in offset surfaces than in offset curves. Nevertheless, it can be efficiently resolved using a Boolean union operation in the *rt*-parameter space. Furthermore, the main strength of our approach is that the topological complexity in both local and global self-intersections is handled in a unified procedure. The local and global self-intersections are resolved simultaneously using the same union procedure. Fig. 15 shows such a case. Even in this situation, our method can deal with self-intersections properly by performing a planar Boolean operation.

5. Post-process of numerical improvement

A ρ-trimming of offset surfaces is computed rather conservatively by selecting $\rho > 0$. The surface patches resulting from Algorithm 2 are not exactly connected but cross each other slightly (see Fig. 11(a)). Therefore, a postprocess of numeric marching steps needs to be applied to get a highly precise self-intersection curve of the offset surface. A numeric marching between two offset surface patches is essentially a Surface–Surface Intersection (SSI) problem.

In our implementation, a standard numerical marching SSI technique is adapted exploiting the Newton–Raphson method. Finding a good initial solution and analyzing the topology of the intersection curves are the most difficult parts in such a numerical tracing algorithm. A ρ-trimming, however, resolves both tasks and leads to a highly precise self-intersection curve. Two matching trimming curves are first found, which are close to each other in the Euclidean space after evaluation. In the example of Fig. 10, there exist three trimming curves, l_1, l_2, l_3 (see also Fig. 11(b)). Since l_1 and l_3 correspond to the global self-intersection, they produce a single connected component of the self-intersection curves. l_2 is matched to itself since it corresponds to the local self-intersection. Consequently, there are two connected components of the intersection curves. Starting from these matchings, a few steps of the standard Newton–Raphson technique produce a highly precise self-intersection curve. Fig. 11(c) shows two self-intersection curves. The upper one presents self-intersection curves before the numerical improvement stage and the lower one is the result of numerical marching step.

6. Experimental results

Several examples of trimming both local and global self-intersections in offset curves and surfaces are now presented. First, some examples for trimming offset curves are shown. Fig. 12 presents an example of a curve with several offsets in both directions. In Fig. 12(a), the original curve is shown (in gray) with the untrimmed offset approximations. With the aid of the distance square function, the self-intersections are ρ-trimmed in Fig. 12(b), and the result of applying the numerical marching step is shown in Fig. 12(c).

Figs. 13 and 14 present two more complex examples of offset trimming for curves. Here, (a) shows the original curve

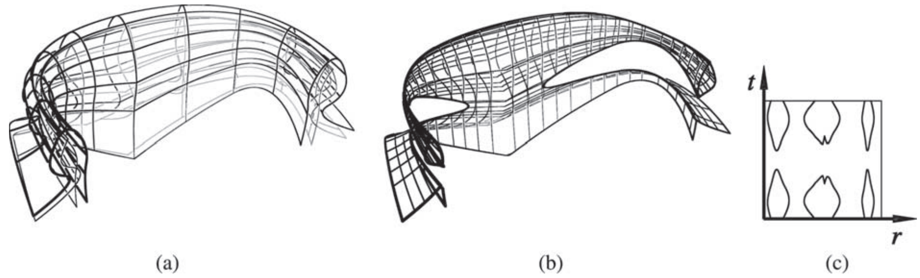

Fig. 16. (a) presents a sweep surface with scale change of cross-sections (bold lines) and its untrimmed ε-offset approximation (light lines). (b) shows the result of ρ-trimming of the ε-offset surface. The trimming curves are shown in (c), in the *rt*-parameter domain.

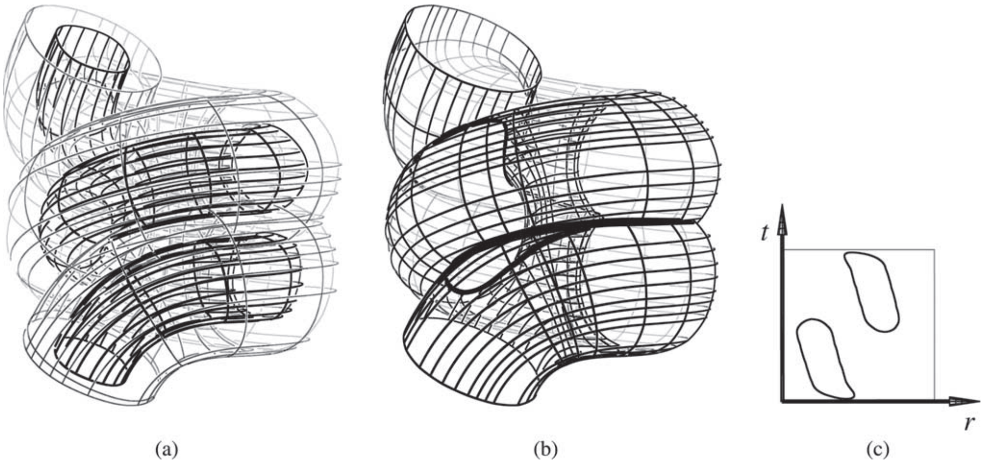

Fig. 17. (a) presents a sweep of a circular section (bold lines) following a helix trajectory and its untrimmed ε-offset approximation (light lines) and (b) shows the result of ρ-trimming of the ε-offset surface. The trimming curves are shown in (c), in the tr-parameter domain.

(in gray) and its untrimmed offset approximations, and (b) is the result of ρ-trimming the self-intersections using the function $F(r,t)$ of Eq. (1). In all the examples presented in this work, the trimming distance ρ was taken from 1% to 5% of the offset distance d, while an offset tolerance, ε, was 0.1%–1% of the offset distance. Figs. 13(b) and (c) and 14(b) and (c) present the results of trimming with ρ at 5% and 1% of the offset distance, respectively. Small local self-intersections escape the ρ-trimming step at 5% and are trimmed better at 1%. Figs. 13(d) and 14(d) show the results after the post-process of numerical marching steps. The computation time for these results are about the same, all within 1 or 2 seconds on a 2 GHz Pentium IV machine.

These small local self-intersections could clearly appear at any percentage of the ρ-trimming distance, when $\rho > 0$. In many applications, such as robotics and CNC machining, local and arbitrarily small self-intersections will induce large accelerations along the derived path, and hence are highly undesirable. If both $S(u,v)$ and its offset surface $O_d(u,v)$ are parameterized so that $O_d(u_0,v_0)$ is along the normal of S at $S(u_0, v_0)$, a typical result in many offset approximation schemes, one could employ the local self-intersection test presented in [3] as another filtering step that could completely resolve these small local self-intersections.

Continuing to examples of trimming offset surfaces, Fig. 15 shows the same example as in Fig. 10 but with a different offset distance. The local self-intersection and global self-intersection now merge to form a single component. Even in this case, our approach works well and Fig. 15(b) presents the result of trimming. Fig. 15(c) shows the trimming curve in the rt-parameter space.

Figs. 16 and 17 present two more complex examples. Here, (a) is the original surface and its untrimmed ε-offset approximation, and (b) is the result of ρ-trimming of the self-intersections using the function $F(u,v,r,t)$ and the numerical marching step. The trimming curves in the parameter space are shown in Fig. 16(c) and 17(c). In all the examples presented in this work, the trimming distance ρ was taken from 1% to 5% of the offset distance. Since the offset surface becomes quite complex after a rational approximation, it takes about 4–5 min for ρ-trimming the offset surface approximations on a 2 GHz Pentium IV machine. The original surfaces presented in these experimental examples are represented by bicubic NURBS having about 50–70 control points. Their rational bicubic NURBS offset approximation with a tolerance of $\varepsilon = 0.02$ (original objects' dimensions span about a unit length) turned out to have about 5000–6000 control points.

Fig. 18 shows one interesting result of the ρ-trimming. The offset surface of a simple concave surface is shown in Fig. 18(a). Fig. 18(b) presents the simultaneous zero-set of Eqs. (2)–(4) in the rt-domain. There exist self-intersections in two loops, with one curve segment completely contained in the other one. According to the topological structure of the zero-set components as in Section 4, the self-intersections in the trimming curves need to be resolved. Furthermore, the inner loop should be removed since its corresponding zero-set component is totally blocked by the other one. The result of ρ-trimming by using the resolved trimming curves in Fig. 18(c) is shown in Fig. 18(d). Although the ρ-trimming can be accomplished robustly, a numerical improvement step in this kind of singular example is still difficult.

7. Conclusions

A robust and efficient scheme for trimming both local and global self-intersections in offset curves and surfaces have been presented. The presented approach is based on the derivation of a rational distance map between the original curve or surface and its offset. By simultaneously solving one polynomial equation for an offset curve or three polynomial equations for an offset surface in the parameter space, all the local and global self-intersection regions in offset curves or surfaces can be detected. The zero-set of the polynomial equation(s) prescribes the self-intersection regions and these regions are trimmed by projecting the zero-set into an appropriate parameter space. The projection operation simplifies the topological complexity of the zero-set and makes the algorithm numerically stable and efficient. Furthermore, numerical marching post-processing

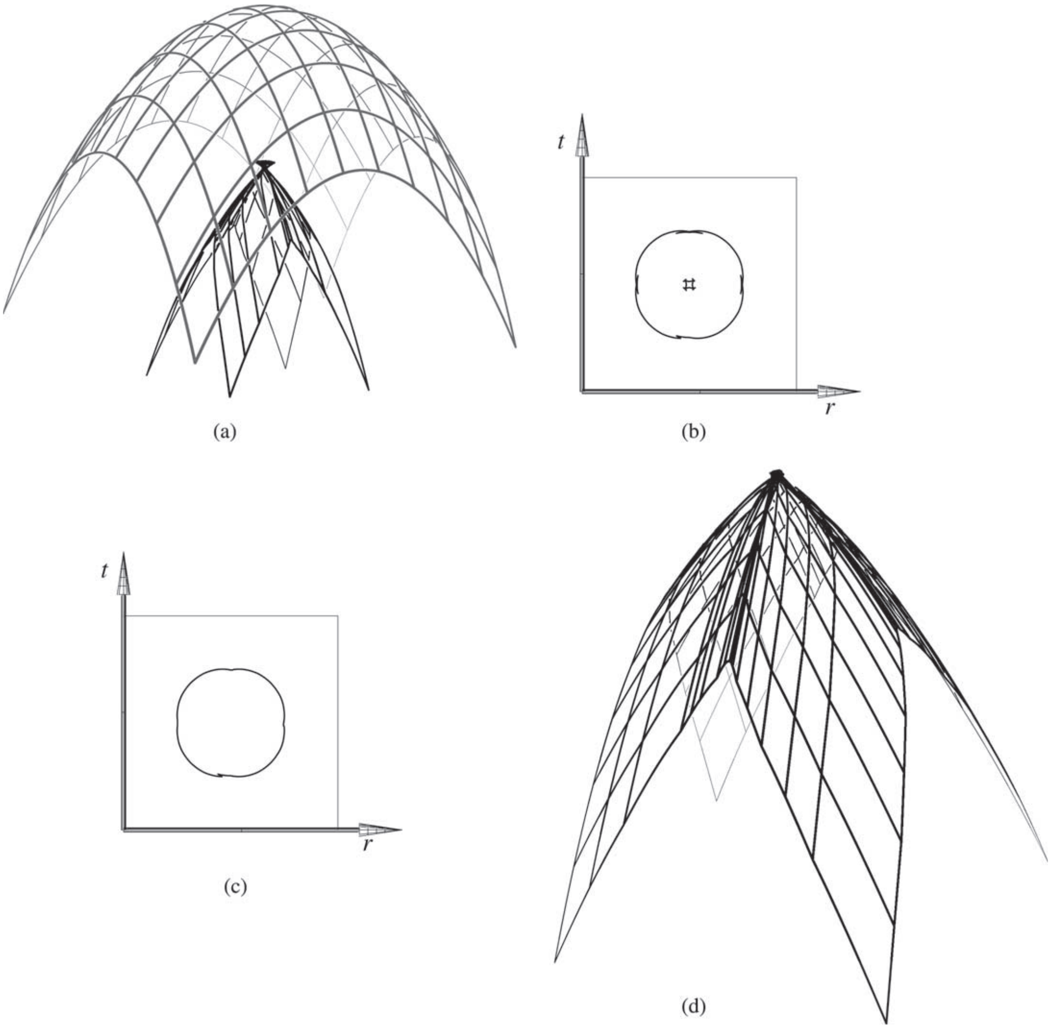

Fig. 18. A simple concave surface and its untrimmed ε-offset surface approximation is presented in (a) and the simultaneous zero-set of Eqs. (2)–(4) in the tr-domain is shown in (b). After considering the topological structure of the zero-set, self-intersections are resolved in (c). Also, the inner trimming curve has been removed since its corresponding zero-set component is totally blocked by the outer component. (d) shows the result of ρ-trimming of the ε-offset surface. The numeric marching technique of SSI encounters some difficulties in such a highly singular case.

steps provide highly precise self-intersection elimination in both offset curves and surfaces.

A limitation of the current ρ-trimming approach is that some tiny self-intersection loops may not be detected in the trimming procedure. Better ways of preventing all small self-intersections in the offset curves and surfaces should be sought. In cases where the offset is differently parameterized, by reparameterizing of the offset curves and surfaces so that they match the original curves and surfaces, the local self-intersection scheme of Elber and Cohen [3] can possibly be applied. Self-distance maps and discontinuities in the derivative of the maps near the offset curves and surfaces can also be useful in detecting tiny loops and global self-intersections.

Acknowledgements

We would like to thank the anonymous reviewers for their invaluable comments. All the algorithms and figures presented in this paper were implemented and generated using the IRIT solid modeling system [11] developed at the Technion, Israel. This work was supported in part by European FP6 NoE grant 506766 (AIM@SHAPE), in part by the Israel Science Foundation (grant No. 857/04), in part by the Korean Ministry of Information and Communication (MIC) under the Program of IT Research Center on CGVR and in part by grant No. R01-2002-000-00512-0 from the Basic Research Program of the Korea Science and Engineering Foundation (KOSEF).

References

[1] Aomura S, Uehara T. Self-intersection of an offset surface. Comput-Aided Des 1990;22(7):417–21.

[2] Cohen E, Ho CC. Surface self-intersection. In: Lyche T, Schumaker LL, editors. Mathematical methods for curves and surfaces, 2000. p. 183–94.

[3] Elber G, Cohen E. Error bounded variable distance offset operator for freeform curves and surfaces. Int J Comput Geometry Appl 1991;1(1): 67–78.

[4] Elber G, Lee IK, Kim MS. Comparing offset curve approximation methods. IEEE CG A 1997;17(3):62–71.

[5] Elber G, Kim MS. Computing rational bisectors. IEEE CG A 1999;19(6): 76–81.

[6] Elber G, Kim MS. Geometric constraint solver using multivariate rational spline functions. Proc. of ACM symposium on solid modeling and applications, Ann Arbor, MI, June 2001;1–10.

[7] Elber G. Trimming local and global self-intersections in offset curves using distance maps. Proc. of the 10th IMA conference on the mathematics of surfaces, Leeds, UK, pp. 213–222; September 2003.

[8] Elber G, Kim MS, Heo HS. The convex hull of rational plane curves. Graph Models 2001;63:151–62.

[9] Farouki RT, Tsai YF, Yuan GF. Contour machining of freeform surfaces with real-time PH curve CNC interpolators. Comput Aided Geometric Des 1999;16:61–76.

[10] Hoschek J, Lasser D. Fundamentals of computer aided geometric design. Wellesley, MA: AK Peters; 1993.

[11] IRIT 9.0 User's Manual, October (2000), Technion. http://www.cs.technion.ac.il/∼irit.

[12] Kim MS, Elber G. Problem reduction to parameter space. The mathematics of surface IX (Proc. of the ninth IMA conference), London; 2000. p. 82-98.

[13] Ravi Kumar CVV, Shastry KG, Prakash BG. Computing non-self-intersecting offsets of NURBS surfaces. Comput-Aided Des 2002;34(3):209–28.

[14] Lee IK, Kim MS, Elber G. Planar curve offset based on circle approximation. Comput-Aided Des 1996;28(8):617–30.

[15] Li YM, Hsu VY. Curve offsetting based on legendre series. Comput Aided Geometric Des 1998;15(7):711–20.

[16] Maekawa T, Cho W, Patrikalakis NM. Computation of self-intersections of offsets of bezier surface patches. J Mechanical Des: ASME Trans 1997; 119(2):275–83.

[17] Peternell M, Pottmann H. A Laguerre geometric approach to rational offsets. Comput Aided Geometric Des 1998;15(3):223–49.

[18] Seong JK, Elber G, Johnstone JK, Kim MS. The convex hull of freeform surfaces. Computing 2004;72(1):171–83.

[19] Seong JK, Kim KJ, Kim MS, Elber G, Martin R. Intersecting a freeform surface with a general swept surface. Comput-Aided Des 2005;37(5):473–83.

[20] Seong JK, Elber G, Kim MS. Contouring 1- and 2-manifolds in arbitrary dimensions. Int Conference Shape Model Appl 2005;216–25.

[21] Wallner J, Sakkalis T, Maekawa T, Pottmann H, Yu G. Self-intersections of offset curves and surfaces. Int J Shape Model 2001;7(1):1–21.

[22] Wang Y. Intersection of offsets of parametric surfaces. Comput Aided Geometric Des 1996;13(5):453–65.