Simple and branched skins of systems of circles and convex shapes

Bohumír Bastl[a,b], Jiří Kosinka[c], Miroslav Lávička[a,b]

[a] Department of Mathematics, Faculty of Applied Sciences, University of West Bohemia Univerzitní 8, 301 00 Plzeň, Czech Republic
[b] NTIS – New Technologies for the Information Society, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, 301 00 Plzeň, Czech Republic
[c] Computer Laboratory, University of Cambridge, 15 JJ Thomson Avenue, Cambridge CB3 0FD, United Kingdom

Abstract

Recently, there has been considerable interest in skinning circles and spheres. In this paper we present a simple algorithm for skinning circles in the plane. Our novel approach allows the skin to touch a particular circle not only at a point, but also along a whole circular arc. This results in naturally looking skins. Due to the simplicity of our algorithm, it can be generalised to branched skins, to skinning simple convex shapes in the plane, and to sphere skinning in 3D. The functionality of the designed algorithm is presented and discussed on several examples.

Keywords: Circle, convex shape, skinning, interpolation, spline.

1. Introduction

This paper is devoted to an interesting geometric operation, the operation of skinning. Skinning is a construction of a G^k (C^k) continuous interpolation curve/surface of an ordered sequence of planar or spatial shapes (most often of the same type such as circles and spheres). This operation can be viewed as a particular analogy to the well-known interpolation of point data sets, which is one of the most crucial techniques in Computer Aided Geometric Design, cf. [1] and the references therein. Due to its technical importance, skinning has attracted the geometric modelling community in recent years and one can find several papers on this topic, see e.g. [2–4]. This type of interpolatory skinning is not to be confused with skinning in graphics applications such as [5].

Circles and spheres belong among the most used input shapes for skinning, as these objects and their skins play an important role in various applications, e.g. computational chemistry, molecular biology, computer animation, and modelling of tubular surfaces, cf. [6, 7]. In a discrete sense, skinning can be regarded as a part of the problem of computing envelopes of families of circles/spheres using the cyclographic mapping [8, 9]. Skinning is also closely related to representing shapes with the help of the associated medial axis/surface transforms [10–15] and the theory of canal and pipe surfaces [16–18].

The problem of skinning is defined and solved ambiguously in the literature: for a given configuration of input shapes, there exist infinitely many potential solutions. For instance, several techniques exist that skin a molecular model, producing a C^1 continuous surface of high quality. These methods take a cloud of circles/spheres as input and their connectivity is decided automatically based on Voronoi and Delaunay complexes, see [19–21]. In contrast, our input is a *connected* system of input shapes and the resulting skin respects the input connectivity. Some authors impose additional conditions on skin constructions, e.g. minimal surface area or minimal mean curvature [3]. Others prefer simplicity of skinning algorithms; our approach belongs to this category. For instance, an elegant and purely geometric method based on constructing Apollonius circles [22] has been used recently for finding skin touching points on input circles/spheres [4]. Furthermore, different algorithms use different admissible input data sets [23].

All the techniques mentioned above assume that the skin touches each of the input planar (spatial) shapes at two points (along a curve). Our main motivation is to design a simple and efficient algorithm which allows skins to consist also of parts of the boundaries of the input shapes. In our opinion, this introduces an intuitive physical interpretation of skinning which simulates enclosing a sequence of planar shapes by a rubber band (i.e., a generalisation of a spline in the original sense). On the other hand, the modelling adaptivity of our method (i.e., the use of weights) enables the user to shrink the arcs to points (or spherical parts to curves) and thus to obtain a skin in the usual sense. In addition, the simplicity of our algorithm enables us to generalise it directly to the construction of branched skins, to skinning simple convex shapes in the plane, and to sphere skinning in 3D.

The rest of the paper is organised as follows. After introducing our approach to skinning and basic definitions in Section 2, we introduce our skinning algorithm in Section 3. Examples of the basic algorithm are presented in Section 4, examples of its generalisations are shown in Section 5. In Section 6, we discuss the novelty of our approach, and we conclude the paper in Section 7.

Email addresses: bastl@kma.zcu.cz (Bohumír Bastl), Jiri.Kosinka@cl.cam.ac.uk (Jiří Kosinka), lavicka@kma.zcu.cz (Miroslav Lávička)

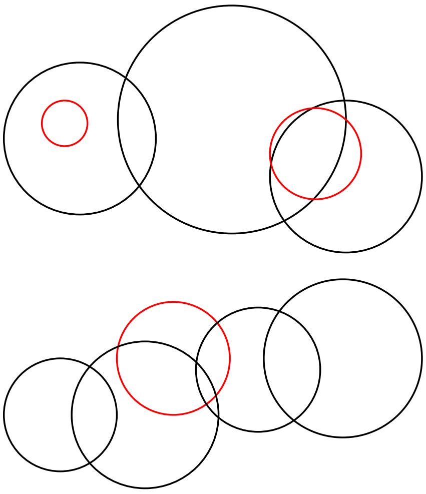

Figure 1: Non-admissible sequences of circles. Top: The circles in red violate condition (i). Bottom: The circle in red violates condition (ii) in (1).

2. Preliminaries

We recall some elementary notions and address the circle skinning problem. In the following sections this problem will be generalised to other convex shapes in the plane and to spheres in space.

We start by defining our notation. We denote c a positively oriented circle in the plane parametrised such that when travelling along the circle one always has its interior to the left, and d the corresponding disk whose boundary is c. Analogously, in what follows, positive orientation is always considered in the anticlockwise sense. Furthermore, when speaking about tangents and tangent vectors of circles, we assume that their orientation follows the anticlockwise orientation of the circles.

Following [4], we call a sequence $C = \{c_1, c_2, \dots, c_n\}$, $n \in \mathbb{N}$, of (oriented) circles *admissible* (see Fig. 1) if the corresponding sequence of disks $D = \{d_1, d_2, \dots, d_n\}$ satisfies the conditions:

$$\begin{aligned} &\text{(i)}\quad d_i \not\subset \textstyle\bigcup_{j=1, j\neq i}^{n} d_j,\ i \in \{1, 2, \dots, n\},\\ &\text{(ii)}\quad d_{i-1} \cap d_{i+1} \neq \emptyset \Rightarrow d_{i-1} \cap d_{i+1} \subset d_i. \end{aligned} \tag{1}$$

The relation $X \prec_c Y$ between two points X and Y on c reflects the direction of the parameter along c (considered wrapped around when necessary). The symbol $\widehat{XY}$ denotes the oriented arc on c delimited by the points X, Y such that $X \prec_c Y$. The *midpoint* Z of the arc $\widehat{XY} \subset c$ is defined by the property $Z \in \widehat{XY}$, $\|X - Z\| = \|Z - Y\|$, and denoted $Z = X \overset{\bullet}{\frown}_c Y$.

We now define the operation of *skinning*. Somewhat unconventionally, we first define only one branch of the skin of C, the one on the right; see the green curve in Fig. 2. This branch is denoted by $\mathcal{B}(C)$ and called the *skin branch* of C. The reason for this approach becomes apparent in Section 5.2, where we generalise the linear sequence C of input circles to a branched system of circles.

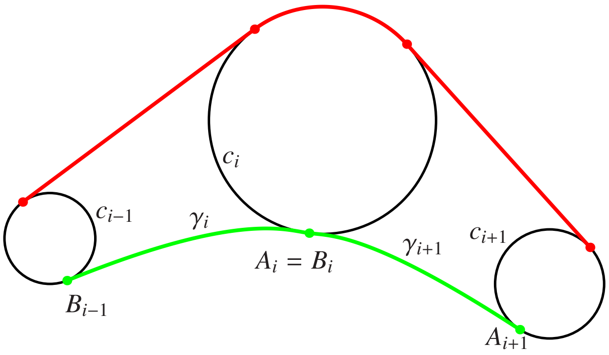

Figure 2: Three consecutive circles c_{i-1}, c_i, c_{i+1} of C, the contact points on them, and the interpolating splines between them.

For each circle $c_i \in C$, we define its right *contact arc* $a_i = \widehat{A_iB_i}$, $A_i \prec_{c_i} B_i$, as an arc on c_i along which the skin branch is to touch it; see Fig. 2. These arcs are allowed to degenerate into points (i.e., $A_i = B_i$ for some or all i). The computation of these points is discussed in detail in Section 3.1. Our approach generalises previous definitions of skins [2, 4].

A G^1 skin branch $\mathcal{B}(C)$ of an admissible sequence C of circles is a G^1 spline such that

$$\mathcal{B}(C) = B_1\gamma_2a_2 \dots a_{n-1}\gamma_nA_n, \tag{2}$$

where γ_i interpolates B_{i-1} and A_i. The other (left) branch of the skin of C is obtained by reversing the sequence and denoted $\mathcal{B}(-C)$. Note that the orientation of the input circles remains unchanged. The union of both branches, i.e., $\mathcal{B}(C) \cup \mathcal{B}(-C)$, is then called the *skin* of C and denoted $\mathcal{S}(C)$. Note that our definition of G^1 should be understood as *at least* G^1.

In the case when a contact arc degenerates to a point, the spline is required to be tangent to its corresponding circle at that point; see Fig. 2. Thus, the construction of a skin is reduced to the construction of contact arcs (or points) and the selection of a desired spline continuity and type (e.g., cubics, bi-arcs, G-code, rational quadratics).

To cover the majority of various input scenarios we adopt the following additional conventions. If condition (i) is not satisfied, we can simply omit the corresponding c_i in C; see Fig. 3, top. In the case that condition (ii) is violated, one can still construct a *generalised skin*. We skip each circle c_i on one side of the skin when only one of the intersection points of c_{i-1} and c_{i+1} lies outside of c_i; see Fig. 3, bottom. This is equivalent to the condition that the radical centre of the three consecutive circles c_{i-1}, c_i, c_{i+1} lies inside of the middle one, used in [4]. However, we prefer to avoid this description as we aim to generalise our approach to more general convex shapes later.

3. Skinning algorithm

In this section we present our approach to skinning admissible sequences of circles in the plane. For the sake of simplicity and to allow generalisations, the algorithm does not rely on properties and methods which are special for circles only (such as Apollonius circles).

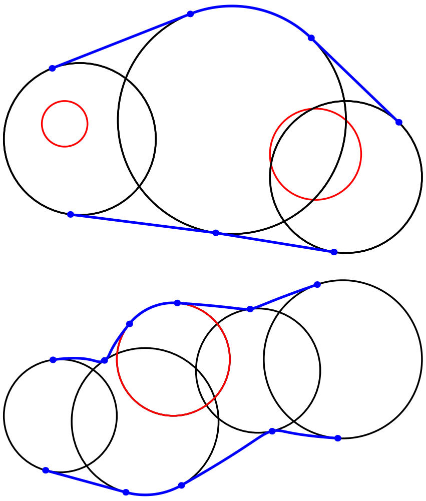

Figure 3: Generalised skins: the situations from Fig. 1 when certain additional conventions on conditions (i) and (ii) in (1) are adopted.

3.1. *Points or arcs of contact*

Since our method determines points or arcs of contact locally, we only consider three consecutive circles c_{i-1}, c_i, c_{i+1}, $i = 2, \ldots, n-1$, at a time. Moreover, in what follows the symbols $\prec$ and $\overset{\bullet}{\frown}$ correspond to c_i.

Our algorithm makes use of right outer bitangents t_{i-1} with touching points $P_{i-1} \in c_{i-1}$ and $Q_i \in c_i$, and t_i with touching points $P_i \in c_i$ and $Q_{i+1} \in c_{i+1}$; see Fig. 4. The ordering $P_{i-1}Q_i$, P_iQ_{i+1} reflects the oriented contact of the circles and the right bitangents. We want to find the maximal part of the arc on c_i delimited by the points Q_i and P_i not contained in the interiors of d_{i-1} or d_{i+1}. For this reason, we consider the intersection points (if they exist as real points) of the circles c_{i-1}, c_i and c_i, c_{i+1}. The intersections corresponding to the right branch are $N_i \in c_{i-1} \cap c_i$ and $N_{i+1} \in c_i \cap c_{i+1}$; see Fig. 4.

Now we compute A_i, B_i on c_i, the end points of touching arcs, as follows:

- if $P_i \in d_{i-1}$ then $V_i = N_i$
  else $V_i = P_i$,
- if $Q_i \in d_{i+1}$ then $U_i = N_{i+1}$
  else $U_i = Q_i$,
- if $U_i \preceq V_i$ then $A_i = U_i$, $B_i = V_i$
  else $A_i = U_i \overset{\bullet}{\frown} V_i = B_i$.

We emphasise that the arcs given by U_i and V_i give all admissible touching points for the skin. Our choice $U_i \overset{\bullet}{\frown} V_i$ is only one possible and natural solution. This choice can be replaced by any other inner point of the arc when needed by additional modelling (or aesthetic) requirements or by an optimisation algorithm (see also the discussion about introducing weights at the end of Section 4).

Moreover, the first and last circles in the input need to be treated separately. The skin end points on these circles are set to the touching points of the constructed outer bitangents.

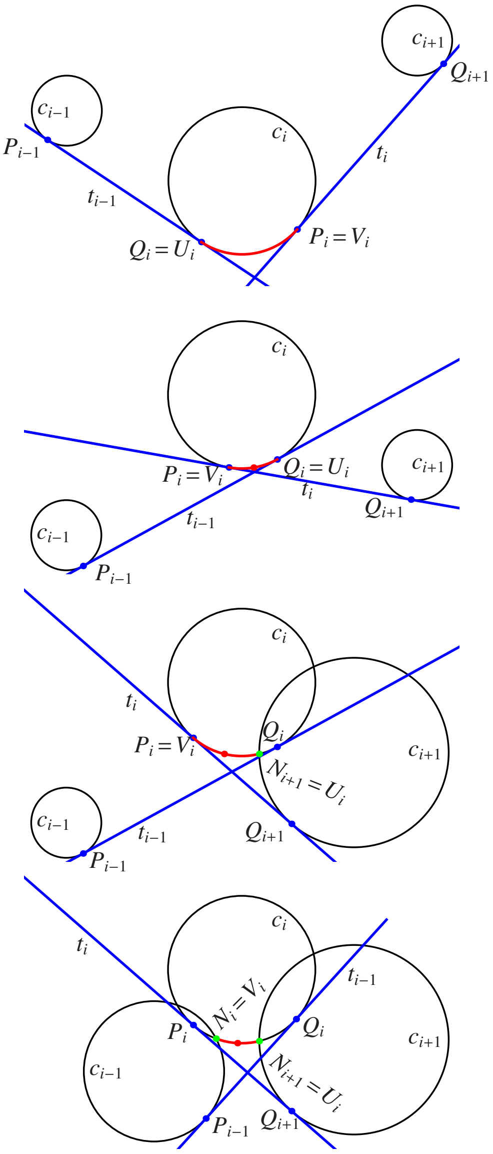

Figure 4: The circles c_{i-1}, c_i, c_{i+1} and the right outer bitangents t_{i-1}, t_i. Various configurations are shown.

3.2. *Skin construction*

With the contact points or arcs a_i from the previous section assigned to each c_i, we are now ready to construct both branches of the skin. To this end we construct interpolants γ_i, $i = 1, \ldots, n-1$ of (2). For the sake of simplicity, we chose to use Bézier/Fergusson cubics as interpolants γ_i. Their endpoints are B_{i-1} and A_i and their corresponding tangent vectors are f_{i-1} (tangent to c_{i-1}) and e_i (tangent to c_i). Their magnitudes are given by

$$\begin{aligned} &\text{if } A_i = B_i \text{ then } \|e_i\| = 2\|A_i - U_i\| = \|f_i\| = 2\|B_i - V_i\|, \\ &\qquad \text{else } \|e_i\| = 2\|A_i - U_{i-1}\|,\ \|f_i\| = 2\|B_i - V_{i+1}\|. \end{aligned} \tag{3}$$

These magnitudes reflect the mutual position of the input circles, their radii, and admissible range for contact on each circle; see Fig. 2. Also, the interpolant γ_i degenerates to a line

Algorithm 1 Skin branch computation

```
Input: $C = \{c_1, c_2, \ldots, c_n\}$ (an admissible sequence)
Output: $\mathcal{B}(C) = B_1\gamma_2 a_2 \ldots a_{n-1}\gamma_n A_n$ (the right skin branch)
 1: Construct outer right bitangents $t_i$ of the circles $c_i$ and $c_{i+1}$
    with the touching points $P_i \in c_i$ and $Q_{i+1} \in c_{i+1}$. All evaluations
    and assignments are done for all $i$.
 2: if $c_{i-1} \cap c_i \neq \emptyset$ then
 3:     $N_i = c_{i-1} \cap c_i \cap \widehat{P_i Q_i}$
 4: end if
 5: if $P_i \in d_{i-1}$ then
 6:     $V_i = N_i$
 7: else
 8:     $V_i = P_i$
 9: end if
10: if $Q_i \in d_{i+1}$ then
11:     $U_i = N_{i+1}$
12: else
13:     $U_i = Q_i$
14: end if
15: if $U_i \preceq V_i$ then
16:     $A_i = U_i, B_i = V_i$
17: else
18:     $A_i = U_i \overset{\bullet}{\frown} V_i = B_i$
19: end if
20: if $A_i = B_i$ (with $e_i$, $f_i$ the tangent vectors of $c_i$ at $A_i$, $B_i$,
    respectively) then
21:     $\|e_i\| = 2\|A_i - U_i\| = \|f_i\| = 2\|B_i - V_i\|$
22: else
23:     $\|e_i\| = 2\|A_i - U_{i-1}\|, \|f_i\| = 2\|B_i - V_{i+1}\|$
24: end if
25: Construct $\gamma_i$ as the Fergusson cubic given by $G^1$ data
    $\{B_i, A_{i+1}; f_i, e_{i+1}\}$
```

segment where appropriate, i.e., when $B_i \neq A_i$ and $B_{i+1} \neq A_{i+1}$. The whole construction is summarised in Algorithm 1.

An important observation here is the fact that all the operations in Algorithm 1 are continuous with respect to the input data. Consequently, the constructed skin depends continuously on the input sequence of circles. More precisely, a small/infinitesimal change of the input circles (positions and radii) causes a small/infinitesimal change of the constructed skin. This is important in modelling and other applications.

4. Examples

In this section, we present several examples of our algorithm applied to various input configurations. To facilitate easy comparison, we also include some of the data sets used in [4].

In Fig. 5, the second example from the top demonstrates that our approach outputs straight skins in appropriate scenarios.

The user has the option to control which branch of the skin is to touch which circle. The example in Fig. 5, second from bottom, shows six input circles $\{c_1, c_2, c_3, c_4, c_5, c_6\}$ (ordered from left). The bottom example uses as input only the sequences $\{c_1, c_2, c_3, c_4, c_6\}$ and $\{c_6, c_5, c_3, c_2, c_1\}$. This is user's choice.

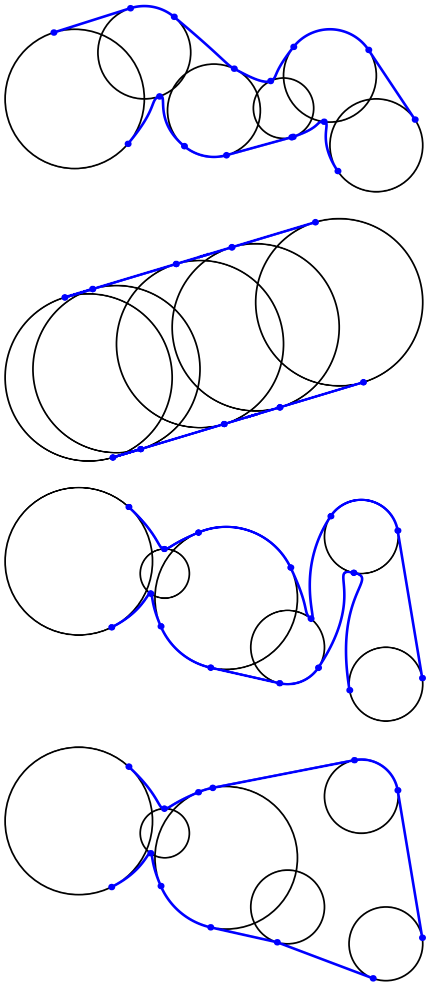

Figure 5: Several examples. Note that our algorithm reproduces straight skins where appropriate. The same sequence of circles with a standard skin (second from bottom) and a generalised skin (bottom) where the user has the possibility to skip a chosen circle.

If a particular application so requires, we can let all contact arcs degenerate to points to obtain skins in the usual sense. The simplest and natural choice is to choose midpoints of contact arcs. An example is shown in Fig. 6: On the left, we show the default output of Algorithm 1; on the right, arcs are reduced to contact points by setting $A_i = U_i \overset{\bullet}{\frown} V_i = B_i$ on line 16 of our algorithm. This brings us to the idea of using weights to specify the contact of the skin with the input circles. These weights can be specified locally, but for the sake of simplicity, we only consider a single global weight w. Then, $w = 1$ gives the default solution with contact arcs (provided that $U_i \prec V_i$), $w = 0$ yields contact points, and any other value of $w \in (0, 1)$ gives intermediate results by linear angular interpolation between U_i and $U_i \overset{\bullet}{\frown} V_i$, and similarly for V_i.

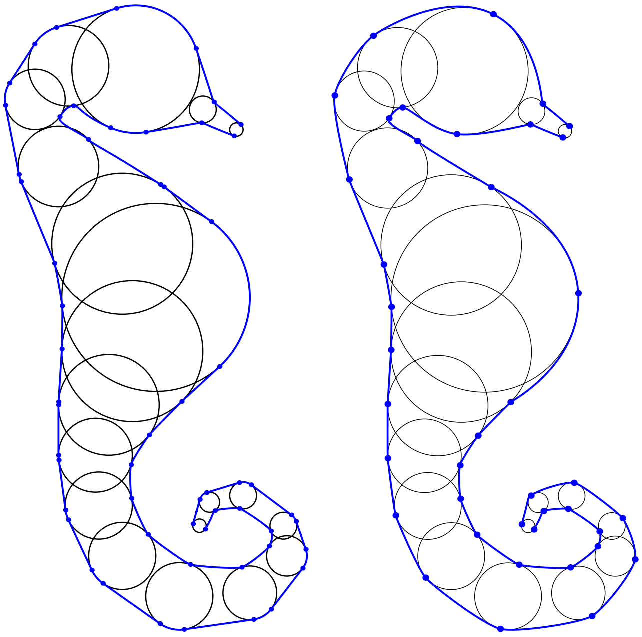

Figure 6: A skin with touching arcs (left, $w = 1$) and touching points (right, $w = 0$).

5. Extensions

Due to the simplicity of our skinning algorithm, we can generalise it to cover also other input shapes both in the plane and 3D space.

5.1. *Convex planar shapes*

Inspecting our algorithm more closely, we see that most of the steps can be used unchanged also for a broader class of shapes than circles. Hence, in this section we generalise the presented approach to an admissible sequence C of closed convex parametric curves $c_1, \ldots, c_n$ in the plane (and associated planar domains d_i determined by the boundaries c_i) fulfilling conditions (1). Again, we assume that when travelling along c_i in the direction given by its parametrisation, its interior lies to the left.

Analogously to the circle case, the relation $X \prec_c Y$ is related to the direction of the parameter along c, the symbol $\widehat{XY}$ denotes the oriented arc on c delimited by the points X, Y such that $X \prec_c Y$, and $Z = X \overset{\bullet}{\frown}_c Y$ is given by the property that $X \prec_c Z \prec_c Y$ and $\|X - Z\| = \|Z - Y\|$. Introducing $\overset{\bullet}{\frown}_c$ in this way is not only a direct generalisation of the definition of the midpoint on a circular arc but it also ensures C^1 continuity of the skin branch at that point; cf. the magnitudes of the tangent vectors in (3).

In Algorithm 1, line 1, we need to construct outer right bitangents t_i of the convex curves c_i, c_{i+1}. For this, one can use e.g. the method of [24, 25] for processing bitangents between c_i, c_{i+1} with the touching points $P_i \in c_i$ and $Q_{i+1} \in c_{i+1}$. The remaining steps of Algorithm 1 do not require any modification. We emphasise that all steps in the construction, and thus also the result, are independent of the input parametrisation of the input convex curves c_i preserving the orientation. An example of this generalisation is presented in Fig. 7.

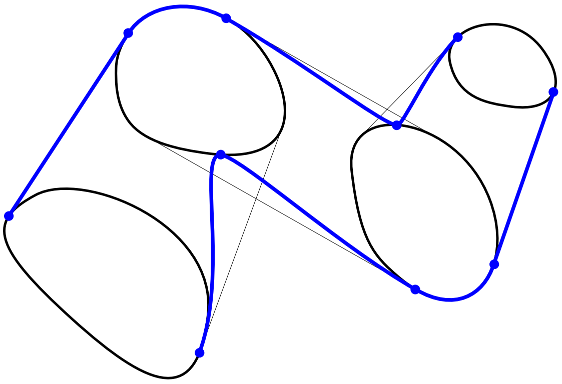

Figure 7: A skin of a sequence of convex shapes. Outer bitangents used in the computation are also shown.

5.2. *Branching*

We have so far allowed only a linear sequence of input shapes. However, this restriction can be easily lifted. In what follows we also admit more complicated configurations. We present only some simple examples which illustrate the main idea.

First, we extend our approach so that a certain circle in the sequence can be repeated. Consider e.g. a system consisting of five circles $c_1, \ldots, c_5$; see Fig. 8. Then we can construct the standard skin as the union $\mathcal{B}(c_1, c_2, \ldots, c_5) \cup \mathcal{B}(c_5, c_4 \ldots, c_1)$, see Fig. 8 (top), or the skin as the union $\mathcal{B}(c_1, c_2, \ldots, c_5, c_1, c_2) \cup \mathcal{B}(c_5, c_4 \ldots, c_1, c_5, c_4)$, see Fig. 8 (bottom).

Furthermore, consider another system of six circles $c_1, \ldots, c_6$ in Fig. 9. We can again construct the standard skin as the union $\mathcal{B}(c_1, c_2, \ldots, c_6) \cup \mathcal{B}(c_6, c_5 \ldots, c_1)$, see Fig. 9 (top), or a branched skin understood as the union $\mathcal{B}(c_1, c_2, c_3, c_6, c_5, c_4, c_3, c_2, c_1) \cup \mathcal{B}(c_3, c_4, c_5, c_6, c_3, c_4)$, see Fig. 9 (bottom). Skins consisting of more than two branches are also possible.

The aforementioned approach offers enough flexibility for modelling purposes as the user can simply control the shape (and type) of the skin by prescribing different ordered systems of circles, as shown above. Note that Algorithm 1 does not require any modification. In addition, we are now able to describe simple skins by one branch only; e.g. the skins in Fig. 5 can be considered as $\mathcal{S}(C) = \mathcal{B}(c_1, c_2, \ldots, c_5, c_6, c_5, \ldots, c_2, c_1)$. Finally, we emphasise that branched skins can be easily constructed also for systems of arbitrary convex shapes (see Fig. 10) using the approach introduced in Section 5.1.

5.3. *Spheres in 3D space*

Following the approach of [4], we can extend our algorithm to 3D space and construct a skin of an admissible sequence of spheres $\Sigma = \{S_1, S_2, \ldots, S_n\}$ satisfying analogous conditions as the circles in (1), disks are simply replaced by balls. We want to construct a G^1 skin $\mathcal{S}(\Sigma)$ of Σ as a G^1 spline surface

$$\mathcal{S}(\Sigma) = \nu_1 \Gamma_2 \Delta_2 \ldots \Delta_{n-1} \Gamma_n \mu_n, \tag{4}$$

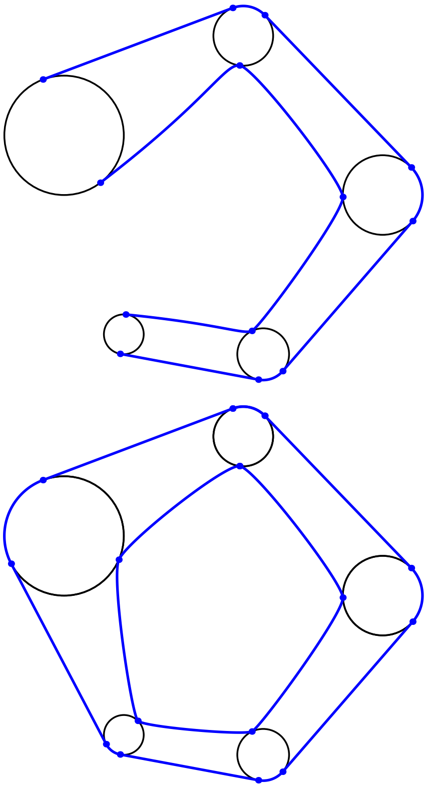

Figure 8: Two examples of possible skins for the same input configuration of circles (up to connectivity).

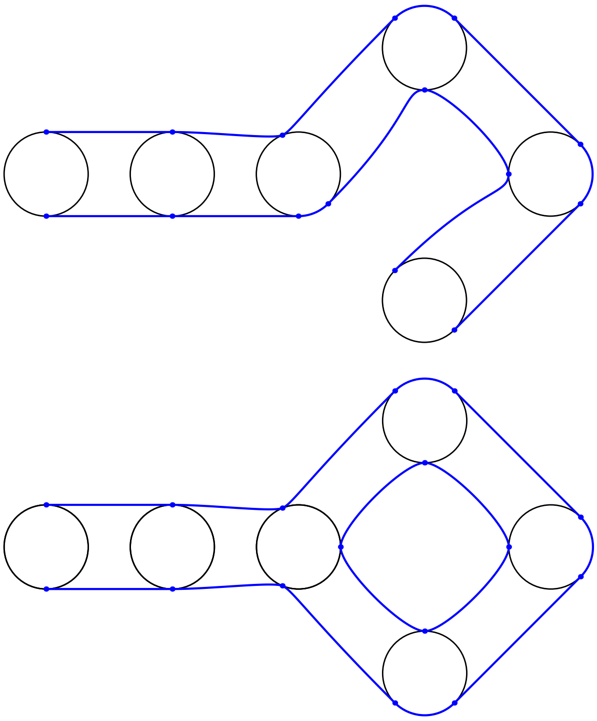

Figure 9: A simple and a branched skin constructed for the same input system of circles (up to connectivity).

where μ_i, ν_i are the (nonintersecting, possibly touching) contact circles on each sphere S_i, Δ_i is the difference of S_i and the spherical caps determined by the circles μ_i, ν_i. Γ_i is a surface joining smoothly S_{i-1} along ν_{i-1} with S_i along μ_i (e.g. parts of canal or ringed blending surfaces).

For determining the contact circles μ_i, ν_i on each sphere S_i we use the method of [4] which transforms this to a planar problem. Let us briefly recall it. Consider the spheres S_{i-1}, S_i, S_{i+1} and intersect them with the plane Π_i given by their centres. If the centres are collinear, we choose Π_i as any plane from the pencil. Intersecting the spheres with the plane Π_i, we obtain three circles. Applying the approach from Section 3 we can construct the points $A_i^{\pm}, B_i^{\pm}$ on the middle circle ($^{\pm}$ corresponds to the left and right branch of the skin, respectively). There exist unique planes Φ_i, Ψ_i passing through the points A_i^{+}, B_i^{-} and A_i^{-}, B_i^{+}, respectively, and perpendicular to Π_i. Finally, $\mu_i = S_i \cap \Phi_i$ and $\nu_i = S_i \cap \Psi_i$.

For constructing the blends Γ_i between input data $\{S_{i-1}, \nu_{i-1};\ S_i, \mu_i\}$ one can use any of the well-established blending methods; for overview of several blending techniques see e.g. [1, 15, 26] and the references therein. In our constructions, we have used the algorithm of [27], which uses rational contour curves of canal surfaces guaranteeing not only the rationality of the resulting blends but also the rationality of their offsets. Fig. 11 illustrates our 3D skinning technique. Fig. 12 shows the output of our skinning algorithm in 3D for another configuration of spheres. We would like to stress out that using canal surfaces for constructing skins between spheres enables one to apply many existing techniques such as offsetting, distance computation, computation of perspective and parallel silhouettes, computing intersections, computing isophotes – see e.g. [28–30].

In analogy to the construction of branched skins in the plane, we can also consider branching in 3D and construct branched skins of systems of spheres. However, this construction is more complicated than in the planar case. Two types of situations can arise. Either, branching occurs on a particular sphere. In this case, the skin consists of branches of type (4) and parts of spheres; see Fig. 13. Or, a blend between more than two spheres is needed. This is beyond the scope of this paper. We only recall that the method from [15] can be employed

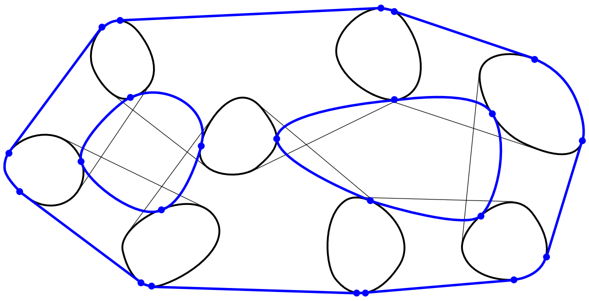

Figure 10: A branched skin of a system of convex shapes. Outer bitangents used in the computation are also shown.

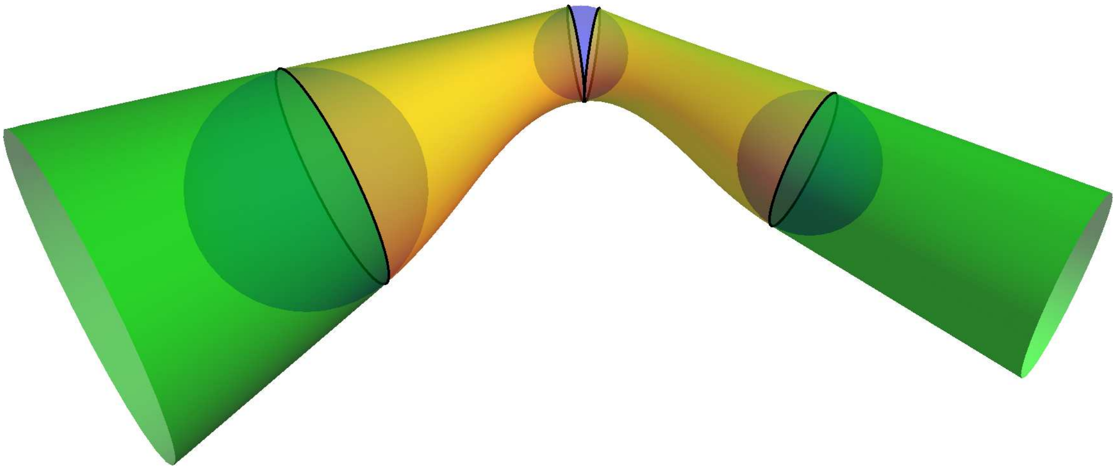

Figure 11: The main principle of the skinning algorithm in 3D is illustrated on three spheres S_1, S_2, S_3. The skin consists of two canal surfaces Γ_1, Γ_2 (yellow) blending two touching cones (green) along the end circles ν_1, μ_2 and ν_2, μ_3 (black), respectively, and the spherical part $\Delta_2 \subset S_2$ (blue).

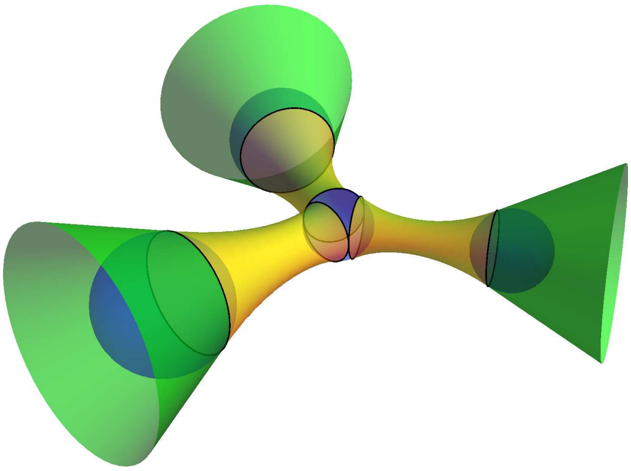

Figure 13: A branched skin of a system of spheres in 3D. Branching occurs on the central sphere.

and blends of canal surfaces from polyhedral medial transform representations used; see Fig. 14 in which the red surface is the envelope of a suitable two-parametric system of spheres.

6. Discussion

We first address the relation of our skinning method to known algorithms available in the literature and emphasise the novelty of our approach.

- Our approach requires only simple primitives to perform skinning, whereas other methods often require understanding of several advanced areas such as algebraic and differential topology, convex geometry. The implementation of our method is therefore very simple and accessible.

- Existing methods work solely for discs/balls. We work with more general convex shapes.

- Using our approach, it is easy to compute offsets of constructed skins which, when necessary, can be expressed in NURBS form: it is enough to use Pythagorean-hodograph interpolants.

Inspecting our skin algorithm, we see that the skin is fully determined once the points or arcs of contact have been found. These are determined by the outer bitangents of the input circles. Conditions (i) and (ii) in (1) guarantee that these bitangents can always be obtained. Therefore, we can conclude that a skin exists for all admissible configurations of circles.

We now discuss our admissibility conditions (1) in more detail. These condition were carefully designed and generalise those of existing methods in several ways:

- We allow our input to branch, whereas traditional techniques are limited to a linear sequence of input shapes.

- We support arbitrary convex shapes, which is in contrast with standard algorithms where only circles and spheres are allowed.

- Regarding our conditions (i) and (ii): Note that (i) cannot be lifted as it would automatically lead to nested shapes and inevitably to self-intersections in the resulting skins. In the case of (ii), we emphasise that we allow the user to skip a shape on one of its branches (see Fig. 5), thus effectively lifting any restrictions adopted in existing techniques.

Self-intersections of skins and unwanted intersections of skins with the input shapes are important issues. In the case of circles, there are several possible approaches to avoid self-intersections in our results. Once a cubic interpolant γ has been constructed, one can check that γ does not intersect its two adjacent circles apart from the touching points. This, however, leads to a quartic equation and gives only a posteriori information. An alternative modification could make use of the convex

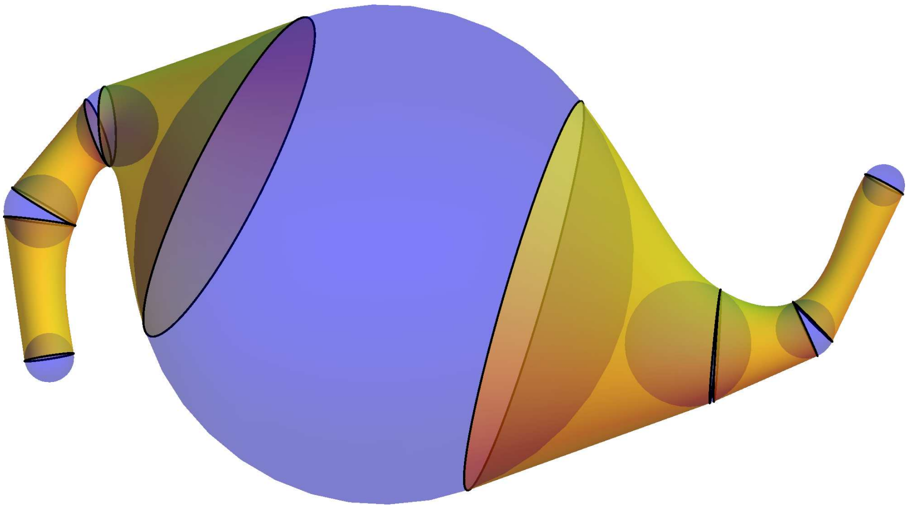

Figure 12: A skin of a sequence of given spheres in 3D. The skin consists of parts of canal surfaces (yellow), touching the spheres along the end circles (black), and spherical parts (blue).

hull of γ, considered as a cubic Bézier curve. Its two endpoints can be moved along their respective circles to ensure that the control polygon gives a C-shaped γ. This can be achieved by simultaneously modifying the weights (Section 4) corresponding to the touching points until a C-shaped control polygon is obtained; see Fig. 15.

The other challenge has to do with the fact that in 3D, the medial axis (surface) transform is, in general, a two-parametric object. Thus, even branched skins of spheres do not guarantee full modelling flexibility (contrary to the planar case). Therefore, one would need to allow two-parametric branched systems (in a discrete sense) of spheres as input to achieve full generality.

Finally, note that we worked with G^1 contact only. However, by employing e.g. quintic splines as interpolants, our method can be extended to higher-order smoothness.

7. Conclusion

In this paper we studied the operation of skinning on a sequence of planar and also spatial shapes. We started with circles (which are the most used input shapes for skinning) and designed a simple algorithm producing naturally looking skins. This was achieved by allowing the skin to touch the input circles along circular arcs.

The algorithm was formulated intentionally so that it does not rely on properties and methods which are special only for circles and thus it can be used also for other planar convex shapes. Further, we discussed more problematic configurations when the skin touches a certain circle more than once. This led to a generalisation with branched skins. We also shortly addressed simple and branched sphere skinning in 3D. Nevertheless, branched sphere skinning needs additional modelling techniques based on envelopes of two-parametric systems of spheres and canal blends if input connectivity is to be respected.

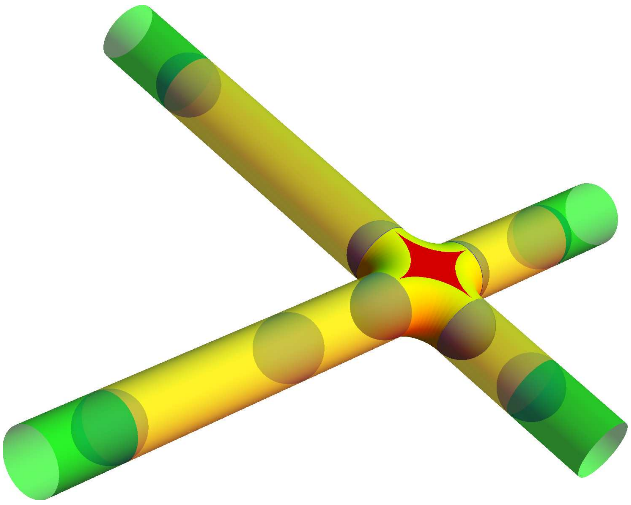

Figure 14: A four-way branched skin of a system of spheres in 3D.

Acknowledgements

The first and the third author were supported by the European Regional Development Fund (ERDF), project “NTIS – New Technologies for the Information Society”, European Centre of Excellence, CZ.1.05/1.1.00/02.0090. The second author thanks EPSRC for supporting this work through Grant EP/H030115/1. We thank the anonymous referees for their helpful insights.

References

[1] Farin G. Curves and surfaces for CAGD: A practical guide. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 2002. ISBN 1-55860-737-4.

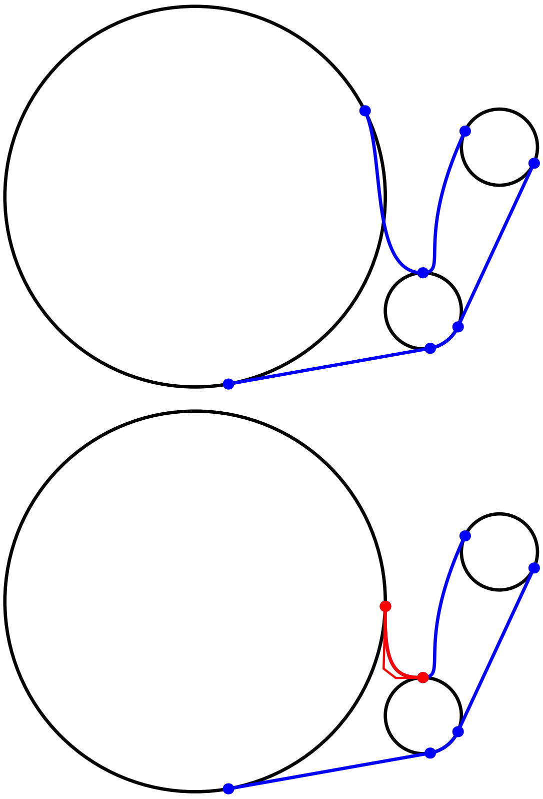

Figure 15: Removing unwanted intersections. Top: The blue skin intersects the big input circle. Bottom: Moving the contact points along their respective circles yields a control polygon (red polyline) giving a C-shaped cubic γ (red) with no intersections.

[2] Slabaugh G, Unal G, Fang T, Rossignac J, Whited B. Variational skinning of an ordered set of discrete 2D balls. Lecture Notes in Computer Science 2008;4975 LNCS:450–61.

[3] Slabaugh G, Whited B, Rossignac J, Fang T, Unal G. 3D ball skinning using PDEs for generation of smooth tubular surfaces. Computer-Aided Design 2010;42(1):18–26.

[4] Kunkli R, Hoffmann M. Skinning of circles and spheres. Computer Aided Geometric Design 2010;27(8):611–21.

[5] Kavan L, Collins S, Žára J, O'Sullivan C. Geometric skinning with approximate dual quaternion blending. ACM Trans Graph 2008;27(4):1–23.

[6] Cheng Hl, Shi X. Quality mesh generation for molecular skin surfaces using restricted union of balls. In: in IEEE Visualization. 2005, p. 399–405.

[7] Rossignac J, Whited B, Slabaugh G, Fang T, Unal G. Pearling: 3D interactive extraction of tubular structures from volumetric images. In: MICCAI workshop on Interaction in Medical Image Analysis and Visualization. 2007,.

[8] Peternell M, Pottmann H. A Laguerre geometric approach to rational offsets. Computer Aided Geometric Design 1998;15:223–49.

[9] Pottmann H, Peternell M. Applications of Laguerre geometry in CAGD. Computer Aided Geometric Design 1998;15:165–86.

[10] Blum H. A transformation for extracting new descriptors of shape. In: Wathen-Dunn W, editor. Models for the perception of speech and visual form. MIT Press; 1967, p. 362–80.

[11] Hoffmann CM, Vermeer PJ. Validity determination for MAT surface representation. In: Proceedings of the 6th IMA Conference on the Mathematics of Surfaces. New York, NY, USA: Clarendon Press. ISBN 0-19-851198-1; 1996, p. 249–65.

[12] Choi H, Han C, Moon H, Roh K, Wee NS. Medial axis transform and offset curves by Minkowski Pythagorean hodograph curves. Computer-Aided Design 1999;31(1):59–72.

[13] Moon H. Minkowski Pythagorean hodographs. Computer Aided Geometric Design 1999;16:739–53.

[14] Yushkevich P, Fletcher PT, Joshi S, Thall A, Pizer SM. Continuous medial representations for geometric object modeling in 2D and 3D. Image and Vision Computing 2003;21(1):17–27.

[15] Bastl B, Jüttler B, Lávička M, Schulz T. Blends of canal surfaces from polyhedral medial transform representations. Computer-Aided Design 2011;43(11):1477–84.

[16] Landsmann G, Schicho J, Winkler F. The parametrization of canal surfaces and the decomposition of polynomials into a sum of two squares. Journal of Symbolic Computation 2001;32(1-2):119–32.

[17] Xu Z, Feng R, Sun J. Analytic and algebraic properties of canal surfaces. Journal of Computational and Applied Mathematics 2006;195(1-2):220–8. The International Symposium on Computing and Information (ISCI2004).

[18] Dohm M, Zube S. The implicit equation of a canal surface. J Symb Comput 2009;44:111–30.

[19] Edelsbrunner H. Deformable smooth surface design. Discrete & Computational Geometry 1999;21(1):87–115.

[20] Cheng HL, Dey T, Edelsbrunner H, Sullivan J. Dynamic skin triangulation. Discrete and Computanional Geometry 2001;25(4):525–68.

[21] Cheng HL, Edelsbrunner H. Area, perimeter and derivatives of a skin curve. Computational Geometry: Theory and Applications 2003;26(2):173–92.

[22] Knight R. The Apollonius contact problem and Lie contact geometry. Journal of Geometry 2005;83(1-2):137–52.

[23] Yang X, Zheng J. Approximate T-spline surface skinning. CAD Computer Aided Design 2012;44(12):1269–76.

[24] Bharath RS, Ramanathan M. The shortest path in a simply-connected domain having a curved boundary. Computer-Aided Design 2011;43(8):923 –33.

[25] Bharath RS, Ramanathan M. Shortest path in a multiply-connected domain having curved boundaries. Computer-Aided Design 2013;45(3):723–32.

[26] Hoschek J, Lasser D. Fundamentals of computer-aided geometric design. AK Peters; 1993. ISBN 1568810075.

[27] Bizzarri M, Lávička M. Parameterizing rational offset canal surfaces via rational contour curves. Computer-Aided Design 2013;45(2):342–50.

[28] Kim KJ, Lee IK. Computing isophotos of surface of revolution and canal surface. Computer Aided Design 2003;35(3):215–23.

[29] Kim KJ, Lee IK. The perspective silhouette of a canal surface. Computer Graphics Forum 2003;22(1):15–22.

[30] Ma Y, Tu C, Wang W. Distance computation for canal surfaces using cone-sphere bounding volumes. Comput Aided Geom Des 2012;29(5):255–64.