Computer-Aided Design 129 (2020) 102903

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

CHoCC: Convex Hull of Cospherical Circles and Applications to Lattices[☆]

Yaohong Wu, Ashish Gupta[*], Kelsey Kurzeja, Jarek Rossignac

School of Interactive Computing, Georgia Institute of Technology, Atlanta, USA

ARTICLE INFO

Article history:
Received 10 August 2019
Received in revised form 21 February 2020
Accepted 9 June 2020

Keywords:
Convex hull
Cospherical circles
Convex decomposition
Lattice structures
Apollonius diagram
Developable surfaces

ABSTRACT

We discuss the properties and computation of the boundary B of a ***CHoCC*** (Convex Hulls of Cospherical Circles), which we define as the curved ***convex hull*** H($\mathcal{C}$) of a set $\mathcal{C}$ of n oriented and cospherical ***circles*** $\{C_i\}$ that bound disjoint spherical ***caps*** of possibly different radii. The faces of B comprise: n ***disks***, each bounded by an input circle, $t = 2n - 4$ ***triangles***, each having vertices on different circles, and $3t/2$ developable surfaces, which we call ***corridors***. The connectivity of B and the vertices of its triangles may be obtained by computing the Apollonius diagram of a flattening of the caps via a stereographic projection. As a more direct alternative, we propose a construction that works directly in 3D. The corridors are each a subset of an elliptic cone and their four vertices are coplanar. We define a ***beam*** as the convex hull of two balls (on which it is ***incident***) and a ***lattice*** as the union of beams that are incident each on a pair of balls of a given set. We say that a lattice is ***clean*** when its beams are disjoint, unless they are incident upon the same ball. To simplify the structure of a clean lattice, one may union it with copies of the balls that are each enlarged so that it includes all intersections of its incident beams. But doing so may increase the total volume of the lattice significantly. To reduce this side-effect, we propose to replace each enlarged ball by a CHoCC and to approximate the lattice by an ***ACHoCC***, which is an assembly of non-interfering CHoCCs for which the contact-faces are disks. We also discuss polyhedral approximations of CHoCCs and of ACHoCCs and advocate their use for processing and printing lattices.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

We propose a representation of the curved convex hull H($\mathcal{C}$) of a set $\mathcal{C} = \{C_i\}$ of n oriented circles of possibly different radii, that all lie on the same sphere S (Fig. 1—left). Throughout this paper, we assume that the spherical ***caps***, each defined by a chosen orientation of its bounding circle, are pairwise disjoint, which implies that the n circles are also disjoint.

The convex hull of two circles in general configuration (i.e., non-cospherical) has faces that are ruled surfaces of degree eight [1]. We show that, remarkably, when the two circles lie on a sphere (cospherical) their convex hull is bounded by faces that are either planar or are subsets of elliptical cones and are hence developable. Furthermore, we show that the Convex Hull of n Cospherical Circles (***CHoCC***) is bounded by geometrically simple faces, that are either disks, triangles or conical surfaces.

Although, the proposed CHoCC may have broader applications, we focus on accelerating the processing of complex lattices [2]. We propose to approximate a lattice by its ***ACHoCC*** decomposition (an assembly of convex beam- and junction-elements, which are all CHoCCs and are connected along shared disk-faces). Fig. 2 shows an example of this decomposition for a toy lattice. While the edges of the lattice include conic arcs [3], the edges of our ACHoCC approximation are either straight or circular. This property may be exploited to simplify geometric queries in some applications: for example, computing the distance between an interior point and the boundary of the lattice.

Instead of the proposed ACHoCC, one could union enlarged copies of the balls (Fig. 3) so as to cover intersection edges between beams. The result has only circular edges. But doing so may increase the volume of the lattice significantly (Fig. 21).

A coarse polyhedral approximation of the CHoCC may be obtained trivially by straightening the circular edges of the CHoCC (Fig. 3-d). We prove that the quad approximations of the green corridors are always planar.

The paper is organized as follows: In Section 2, we discuss related prior art. In Section 3, we describe the boundary decomposition of a CHoCC, and discuss efficient algorithms of boundary computation. We show that the boundary of the convex hull of $n > 2$ cospherical circles has $6n - 10$ faces, and that each face is either a ***disk***, a ***triangle***, or a developable surface, which we call a ***corridor*** (Fig. 1—center).

[☆] This paper has been recommended for acceptance by A. Pasko.

[*] Corresponding author.

E-mail addresses: wuyaohongdio@gmail.com (Y. Wu), ashish.gupta@gatech.edu (A. Gupta), kkurzeja3@gatech.edu (K. Kurzeja), jarek@cc.gatech.edu (J. Rossignac).

https://doi.org/10.1016/j.cad.2020.102903
0010-4485/© 2020 Elsevier Ltd. All rights reserved.

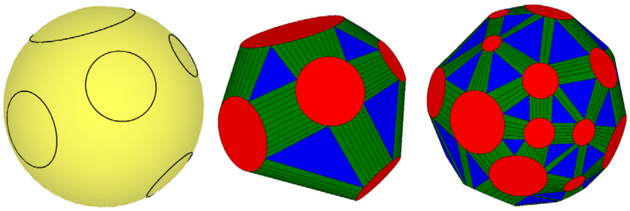


Fig. 1. Circles around disjoint caps on a sphere (left). The convex hull of these circles (center) is composed of disks (red), triangles (blue), and corridors (green), each bounded by two straight and two circular edges. We draw (black) generators on each corridor. We show (right) an example with more circles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

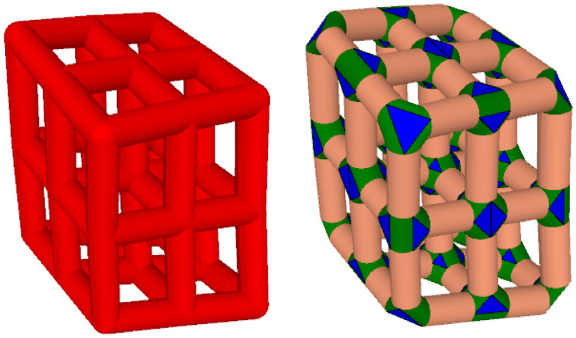


Fig. 2. Approximation of a lattice (left) by an assembly of convex junction-elements (blue–green) and convex beam-elements (pale brown). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

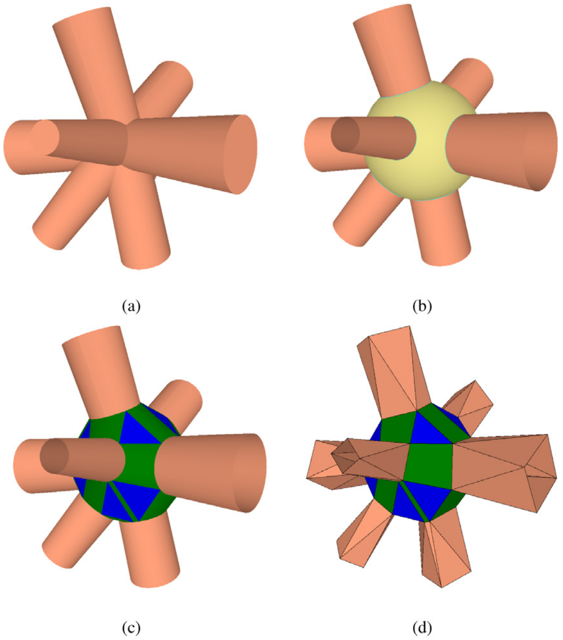


Fig. 3. (a) The boundary of the union U of 8 beams incident on the same ball S has conic edges. (b) Ball approximation: The union of U with a copy S′ of S enlarged so as to contain all beam/beam intersections has only circular edges, but increased volume. (c) CHoCC approximation: Obtained by replacing S′ by the convex hull of these cospherical circles. Its faces are conic frusta (beams), triangles (blue) and corridors (green), which each lie on an elliptic cone. Its edges are line segments or circular arcs. (d) Polygonal approximation: Obtained by straightening the curved edges of the CHoCC. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We explain how this face-graph may be computed efficiently (in expected $O(n \log n)$ time) by constructing the Apollonius diagram [4] of a set of circles in a plane or by using a "plane-rolling" strategy (complexity $O(n^2)$) which works directly in 3D. In Section 4, we observe that each corridor is a subset of an elliptical cone. This result was known in Classical Geometry [5]. We explain how to compute its axis and apex. In Section 5, we discuss an application for approximating a lattice by an ***assembly of convex solid elements*** (one per junction and one per beam), each being a CHoCC. The interiors of these elements are disjoint. Each ***junction-element*** touches incident ***beam-elements*** at ***disk-interfaces*** (Fig. 2). We assume that the length/diameter ratios of the beams are sufficiently large to ensure that the junction-elements are disjoint. We conjecture that most lattices respect this assumption, since prior art [6,7] recommends a ratio superior to 5 (which ensures accuracy of structural analysis). Finally, in Section 6, we discuss the generation of tessellations of these CHoCCs and their use for generating coarse polygonal-mesh approximations of any selected portion of a lattice. The crude tessellation that we propose decomposes a lattice into ***convex polyhedra*** [8] with low face-count. Such a decomposition may be useful in some applications [9], e.g., volume meshing of lattices. We also discuss tessellation refinements that approximate the exact lattice geometry more closely. These may be useful in computing mass properties, such as surface area or volume.

2. Related prior art

The problem of computing the convex hull of a set of points has been well studied [10–12]. In $\mathbb{R}^2$, simple or efficient algorithms are commonly taught in Computational Geometry. Some of these can be extended to $\mathbb{R}^3$ or higher dimensions. Boissonnat et al. [13] presented an algorithm to compute the convex hull of a set of spheres and Geismann et al. [14] presented an algorithm to compute the convex hull of ellipsoids. Rappaport [15] proposed a $O(n \log n)$ solution for computing the convex hull of a set of circles in a plane. Nash et al. [1] proposed a solution for two circles in $\mathbb{R}^3$.

A stereographic projection [16] of cospherical caps transforms them into coplanar disks. Each vertex of their Voronoi diagram is equidistant from three disks. The pre-images of the vertex's closest projections on these three disks are the vertices of the corresponding triangle (blue in Fig. 1) of the CHoCC. The Voronoi diagram of coplanar disks may be computed directly [17] or incrementally [4], with an expected time-complexity of $O(n \log n)$.

On the tessellations of lattices, we draw inspiration from the following references. Xiong et al. [18] proposed to compute structured quad-mesh tessellations of a network of tubular surfaces by following their natural parameterization. Their formulation produces good quality quad-meshes, but is globally constrained over all branches of the tubular network. Stasuik and Piker [19] proposed a Grasshopper plugin called "Exoskeleton" to construct the triangle-mesh of the surface of a solid obtained by the thickening of a center-line model. Their approach follows that of Srinivasan et al. [20], which uses convex hulls of polygonal faces of beams connected to a joint to approximate joints in a network of cylindrical pipes.

3. Boundary of a CHoCC

The boundary B of a CHoCC is homeomorphic to a triangle mesh, i.e. its disks, triangles and corridors correspond to vertices, edges and faces of a triangle mesh. Consequently, we use the Euler characteristic $v - e + f = 2$ (v, e and f being the number of vertices, edges and faces of the triangle mesh) and the fact that for a watertight triangle mesh $e = 3f/2$ (3 edges per triangle,

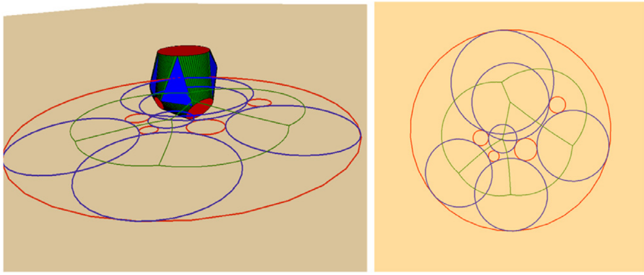


Fig. 4. 3D view (left) of 5 disks (red) bounded by cospherical circles and of their planar map (red circles) by a stereographic projection onto a plane parallel to the top red disk. The edges of the Apollonius diagram of these circles are shown in green. Each blue circle corresponds to a vertex of the Apollonius diagram and is tangent to three red circles. A rectified view of that projection (right). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

each edge belongs to two triangles) to solve for $f = 2v - 4$ and $e = 3f/2$. Hence, the boundary B of the convex hull of n cospherical circles has the following face-types (Fig. 1—center) and counts:

- n ***disks***, each being the convex hull of an input circle C_i
- $t = 2n - 4$ ***triangles***, each connecting 3 of the disks
- $3t/2$ ***corridors***, each adjacent to 2 disks and 2 triangles

Consequently, the total face-count is $n+(2n-4)+3(2n-4)/2$, which is $6n - 10$. Since each triangle has three vertices, B has $3t = 6n - 12$ vertices. Since each edge bounds one corridor and each corridor has 4 edges, B has $6t = 12n - 24$ edges (of which $6n - 12$ are line segments and $6n - 12$ are circular arcs).

We discuss below three different approaches for computing the boundary of a CHoCC.

3.1. Naïve boundary evaluation

The computation of the boundary of a CHoCC requires identifying the triangles of the CHoCC and their vertices. The simplest algorithm uses a "Generate&Test" paradigm:

For each candidate triplet of caps:

- Compute two candidate triangles and
- Reject the invalid ones

It has $O(n^4)$ time complexity, as testing a candidate has $O(n)$ cost.

3.2. Projection-based boundary evaluation

The vertices of the CHoCC may be computed in expected time complexity $O(n \log n)$ by a reduction to Apollonius diagram problem in 2D, which consists of the following steps: (1) mapping the input to the plane (via a stereographic projection of the caps), (2) computing the vertices of the Apollonius diagram of the images of these caps, (3) computing the closest projections of the vertices of that diagram onto the closest 3 disks, and (4) mapping these triplets of vertices back onto the sphere to obtain the vertices of the triangles of the CHoCC. An example is shown in Fig. 4.

The edges of the CHoCC comprise the edges of these triangles and the circular segments obtained by splitting each circle at the vertices that it contains. The faces of the CHoCC comprise the disks, the triangles, and the corridors (discussed in Section 4).

In our target application of approximating a lattice by an ACHoCC (described in Section 5), n is small. Hence, asymptotic complexity may matter less than absolute speed and numeric accuracy of the geometric constructions and tests. We propose below a third solution, which has $O(n^2)$ complexity, but avoids the stereographic projection and its round-off errors (see Section 3.5.)

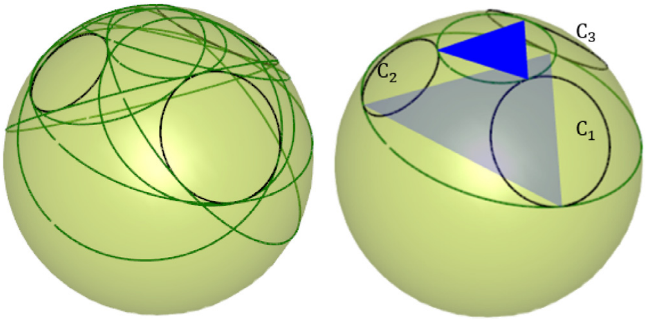


Fig. 5. (Left) Eight circles (dark green) tangent to three given circles $\{C_1, C_2,$ and $C_3\}$ (black) on a sphere. (Right) The two candidate circles that each lie on a supporting plane of the three black circles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.3. Plane-rolling boundary evaluation

We use a "plane-rolling" approach (version of Ball-Pivoting [21] in which we use a plane, i.e., a ball with infinite radius).

The initialization step picks a random circle, C_i. Let Π_i denote the plane that contains it. Then, we consider all possible pairs from the remaining circles, C_j and C_k, forming a triplet of circles with C_i. On each such triplet of circles, we can rest two ***supporting planes***, which we denote by Π_{ijk} (Fig. 5). We describe in the next section a closed-form solution to compute these supporting planes. We pick the pair (C_j, C_k) that minimizes the angle between the normals of Π_i and of Π_{ijk}. Intuitively, we want to minimize how much Π_i must "roll" before it rests on three circles. The three points on which Π_i rests are the vertices of the initial triangle, which we add to the boundary B. We also add the three edges of the triangle to a list of ***exposed borders***.

Then, as long as the list of exposed borders is not empty, we pop one of the exposed borders from the list, say between circles C_i and C_j, use Π_{ijk}, the plane of the triangle adjacent to that border, and find a circle C_l such that the angle between the normals of plane Π_{ijk} and the supporting plane Π_{jil} is minimum. Then, we add the triangle corresponding to plane Π_{jil} to the boundary B, and check whether one or both of its two edges that touch C_l match edges already in the list of exposed borders. If an edge is already in the list, we remove it. Otherwise, we add it.

To ensure a constant-time check, we orient the exposed edges consistently (with the outer normals of the triangles) and store, with each cap, the set of exposed borders that start on the circle of that cap.

Observe that, during a continuous "plane-rolling", the rolling-plane has tangential contacts with circles C_i and C_j. Consider the edge that joins these contact points. The corridor that corresponds to the rolling step is swept by this edge and is a developable surface (Section 4).

3.4. Solving apollonius's problem on a sphere

Without loss of generality, given circles $\{C_1, C_2, C_3\}$ on sphere S, we want to compute the two supporting-plane candidates. We provide below a system of equations that defines eight possible solutions (Fig. 5—left). We retain the two (Fig. 5—right) that yield a supporting plane (which has all three circles on the same side).

Note that the intersection of sphere S with Π_{123} is the circumcircle of T_{123} that has tangential contacts with $\{C_1, C_2, C_3\}$. Hence, we are solving a version of the Apollonius construction, but on a sphere.

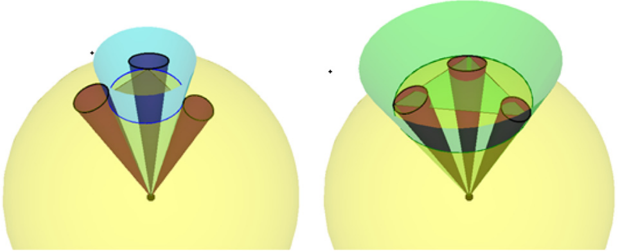


Fig. 6. Three circles (black) on the sphere and their corresponding cones (brown). In solution 1 (left) the three brown cones are outside of the cyan candidate-cone K(C). In solution 2 (right) the three brown cones are inside the green candidate-cone K(C). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

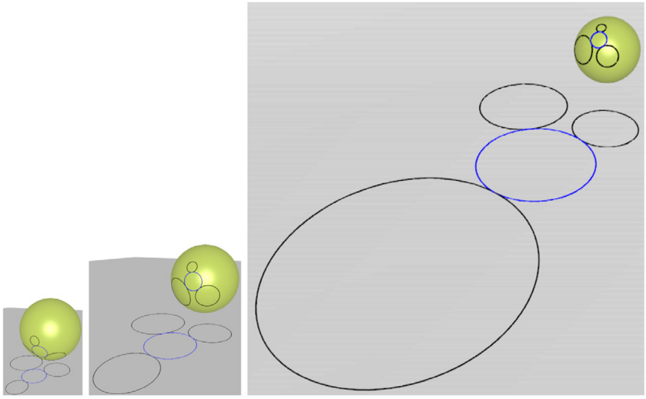


Fig. 7. We draw 3 black circles on a sphere and their stereographic projections on a plane, with respect to the north pole. The projections grow as the circle-configuration is rotated towards the north pole. We also show an Apollonius circle (blue) on the sphere and its counterpart on the plane. The left (resp. right) image corresponds to angles of 90 (resp. 20) degrees, as plotted in Fig. 8. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

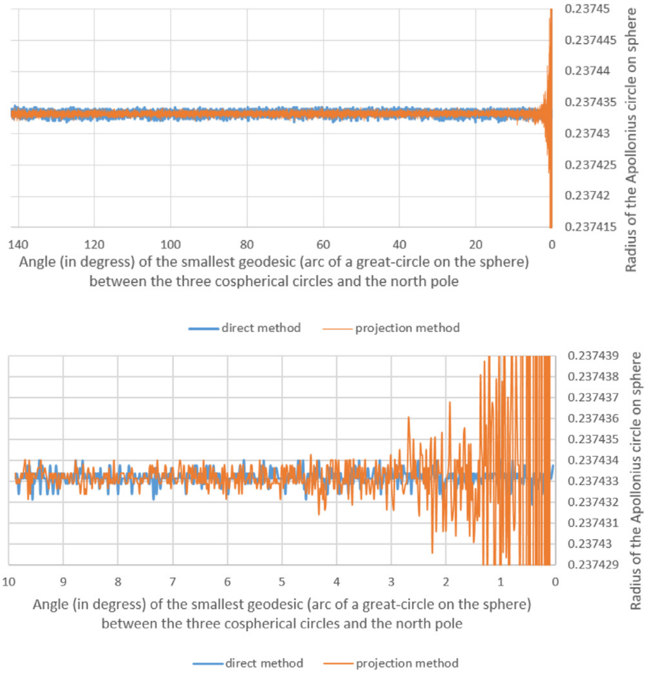


Fig. 8. The graph (top) shows the radius of the Apollonius circle as the three given cospherical circles are rotated closer to the north pole. The radius computed by our direct method (blue) remains precise throughout while that computed via stereographic projection method (orange) fluctuates increasingly as the three circles come closer to the north pole. The graph (bottom) shows a portion of the top graph for values closer to the north pole. At around 4.5 degrees the numerical round-off error in the projection method becomes greater than 1e-6 and continues to grow thereafter. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

For each one of the three circles, $\{C_1, C_2, C_3\}$, on a sphere S with center O and radius r, we consider an unbounded circular half-cone $K(C_i)$ that has apex at O and passes through C_i. Now, if a circle C on S is tangent to the three circles $\{C_1, C_2, C_3\}$, then the cone K(C) with apex at O is in tangential contact (along a line through O) with each of the three cones $\{K(C_1), K(C_1), K(C_3)\}$. As stated above, there are 8 such circles. We pick the two for which $\{K(C_1), K(C_1), K(C_3)\}$ are either all inside or all outside of K(C) (Fig. 6).

We compute K(C) as follows. Let N_1, N_2, and N_3 be the normals of the planes of the circles C_1, C_2, and C_3 and let $\alpha_1, \alpha_2, \alpha_3$ be the half-angles of the corresponding cones $K(C_1)$, $K(C_1)$, and $K(C_3)$ respectively. Then the normal N of the plane of the circle C must satisfy the following set of equations:

$$\begin{aligned} \|N\| &= 1 \\ \cos(\alpha + \alpha_1) &= N \cdot N_1 \\ \cos(\alpha + \alpha_2) &= N \cdot N_2 \\ \cos(\alpha + \alpha_3) &= N \cdot N_3 \end{aligned} \tag{1}$$

The derivation of a closed-form solution (for α and N) to the above system of equations is provided in Appendix A.

The point of contact P_i of the circle C with each of the circles C_i, $i = 1, 2, 3$, is then obtained by:

$$P_i = O + \frac{r(\sin(\alpha_i)N + \sin(\alpha)N_i)}{\sin(\alpha + \alpha_i)} \tag{2}$$

3.5. Comparison of numerical robustness

As discussed above, one of the ways to solve the Apollonius's problem on a sphere is to *project* the three cospherical circles onto a plane (using a stereographic projection), solve the Apollonius's problem in a plane, and then project the solution circle back on to the sphere. The other way is to solve it *directly* on the sphere using the closed form solution that we proposed above.

The proposed ***direct method***, used as a subroutine in the plane-rolling algorithm (Section 3.3), avoids the computational cost of the projection and its inverse. Furthermore, we conjecture that it is numerically more robust than the ***projection method***, which is a critical step in the projection-based algorithm (Section 3.2). The rationale is that the stereographic projection of a circle that is close to the projection pole, is far more enlarged (Fig. 7) than the projection of a circle farther from the pole. To test this conjecture, we consider 3 cospherical circles of different radii on a unit sphere. These circles are initially far from the projection point (north pole) and we incrementally rotate them together (Fig. 7) by a small angle around the sphere so that they come closer to the north pole. To prevent accumulated error in our tests, we always compute the rotated circles from their initial positions. We track how the radius of one of the 8 solutions of the Apollonius's problem varies, when computed by the projection method and when computed by our direct method.

For the direct method, we use the closed-form computation that we provide in Appendix A. Similarly, for the projection method, we project the three circles from the sphere to the plane

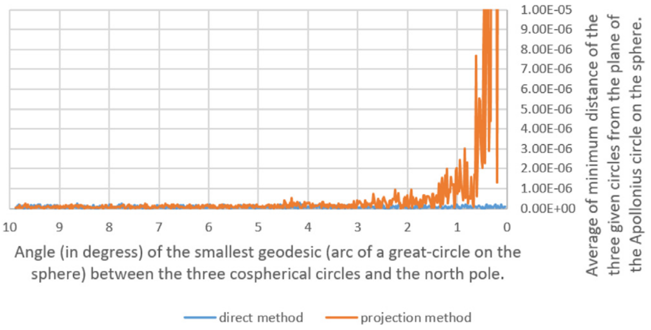


Fig. 9. The graph shows the average of minimum distance (absolute) of the three given circles from the plane of the Apollonius circle on the sphere. For each given circle the distance is computed in a plane that is perpendicular to the plane of that circle and to the plane of the Apollonius circle. The distance computed by directed method (blue) remains closer to zero, while the numerical round-off errors in the distance computed by the projection method (orange) fluctuates increasingly as the given circles come closer to the north pole. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

using the closed-form expressions that we provide in Appendix B. We compute the Apollonius circle in the plane using the standard algebraic method, and then project the solution back on to the sphere using the expressions provided in Appendix B. All computations are performed with single-precision 32-bit IEEE 754 floating-point numbers.

We observe (Fig. 8) that the radius value of the solution-circle obtained by our direct method seems less sensitive to round-off errors than that obtained by the projection method, which as expected, starts fluctuating as the three circles come nearer to the north pole.

Furthermore, in Fig. 9 we plot the average of minimum distance of the three given circles from the plane of the Apollonius circle computed via the two methods. As expected, this distance remains close to zero for the direct method, while it fluctuates increasingly as the three circles come close to the north pole.

Though one may argue that a judicious choice of pole, i.e. a pole that is away from all the cospherical circles, may improve numerical precision, but searching for such a pole would increase the computational cost and may not be feasible in some cases, such as when the input comprises a large number of small circles that are uniformly distributed over the sphere.

3.6. Comparison of performance

We also compare the average run time (from 100k instances) (0.00044 ms) of solving the Apollonius's problem on a sphere using the direct method, against the average run time (0.00030 (projection) + 0.00007 (Apollonius on plane) + 0.00012 (reverse projection) = 0.00049 ms) using the projection method. Of course, when computing the CHoCC using the projection-based algorithm, part of the overhead may be amortized by projecting each input circle only once.

3.7. Our recommendations

We presented above two useful approaches for computing the boundary of the convex hull of n cospherical circles: the projection-based approach (which has expected $O(n \log n)$ time complexity) and the plane-rolling approach (which has $O(n^2)$ time complexity). Although the projection-based method has lower expected complexity, this advantage may not be significant in cases where n is small and may be offset by associated computational costs. Furthermore, the projection-based approach appears more sensitive to numerical round-off errors. We therefore recommend using the plane-rolling approach for which we provide closed-form solutions.

We also recommend exploring the following process of converting a computationally efficient algorithm that computes the Apollonius diagram of coplanar circles into an algorithm that computes it on a sphere. We suggest, to replace the Apollonius circle construction (computing the non-separating circles that are tangent to three given circles) and the circle-interference test (testing whether the disks bounded by two given circles are disjoint) by their 3D counterparts on the sphere. We provide the closed-form solution to the former in Section 3.4. The latter can be implemented trivially since two caps on a sphere overlap if and only if the sum of the half-angles of their cones exceeds the angle between the normals to the oriented planes through their borders.

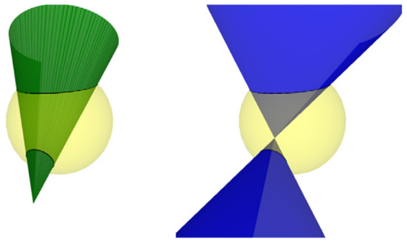


Fig. 10. There exist two elliptic cones passing through two disjoint circles on a sphere S. One cone with its apex outside of S (left) and the other cone with its apex inside of S (right).

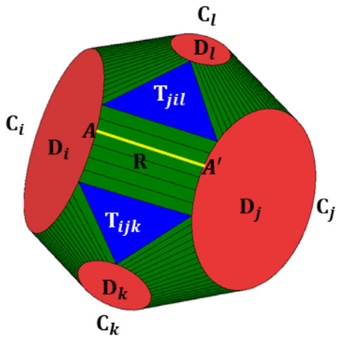


Fig. 11. Each corridor of a CHoCC is a developable surface. Line segment AA' is one of the generators of the corridor R.

4. The surface of a corridor

The convex hull in three dimensions of two circles in general configuration has faces that are ruled surfaces of degree eight [1]. Remarkably, when the two circles lie on a sphere, their convex hull is bounded by faces that are either planar (the two disks in situations where the caps are disjoint) or are subsets of elliptical cones. Indeed, each circle is the intersection of a plane with the sphere. Spain [5] proved that in the special case of intersection between two quadric surfaces, wherein one of the quadrics degenerates into a pair of planes, there exist two cones that pass through their intersection. Hence, there exist two elliptic cones that pass through two cospherical circles (Fig. 10).

Hence, our corridors are each a subset of an elliptical cone (with its apex outside the sphere (Fig. 10)) through two input circles, and the two straight edges bounding a corridor are generators of that cone. As shown in Fig. 11, a corridor, R, is adjacent to two red disks, D_i and D_j, and two blue triangles, T_{ijk} and T_{jil}.

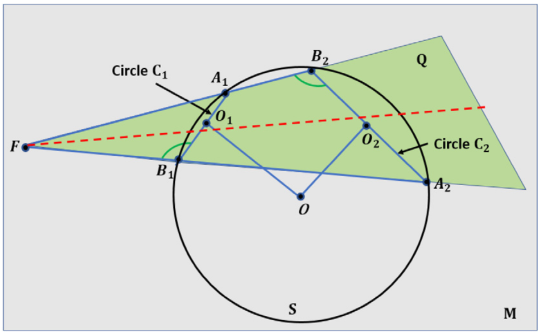


Fig. 12. The apex F of the elliptic cone (green) is the intersection of line $L(A_1, B_2)$ with line $L(B_1, A_2)$. The red dotted line is the axis of the elliptic cone, which bisects $\angle A_1FB_1$.

Considering that the surface of a corridor is conical and therefore is the union of a pencil of ***generators*** (line segments), we discuss below a closed-form approach to render this surface by a quad strip, wherein the quad edges that connect the points on one circular edge of the corridor to the corresponding points on the opposite circular edge of the corridor, are all generators, i.e., they lie on the surface of the corridor.

To render a corridor as a quad mesh or to compute its silhouette, we compute the correspondence between the two circular edges of that corridor, i.e., given a point A (one end of a generator) on C_i we compute the corresponding point A' on C_j (Fig. 11). To establish this correspondence, we first compute the ***apex*** F of the cone through C_i and C_j and then compute the corresponding point A' (the other end of that generator) on C_j as the intersection of plane containing C_j with the line through F and A. We compute the apex F of an elliptic cone corresponding to a corridor as follows.

Consider a sphere S with center O and radius r (Fig. 12). Let C_1 (center O_1, radius r_1) and C_2 (center O_2, radius r_2) be two circles on S. Consider the plane M passing through O_1, O_2, and O. Circle C_1 intersects plane M at two points A_1 and B_1. Similarly, circle C_2 intersects plane M at A_2 and B_2. The lines $L(A_1, B_2)$ and $L(A_2, B_1)$ are two generators of the elliptic cone through C_1 and C_2. The apex F of this elliptic cone is then the intersection of these two lines. We compute this apex as follows, wherein we use simple vector algebra and avoid computing the intersection between two lines in 3D or between a circle and a plane in 3D.

We first compute the points A_1, B_1 and A_2, B_2 using the following equations. The underlined vectors are normalized.

$$\begin{gathered} N_1 = \underline{OO_1},\ N_2 = \underline{OO_2},\ N = \underline{N_1 \times N_2} \\ A_1 = O_1 - r_1(N_1 \times N),\ B_1 = O_1 + r_1(N_1 \times N) \\ A_2 = O_2 - r_2(N_2 \times N),\ B_2 = O_2 + r_2(N_2 \times N) \end{gathered} \tag{3}$$

As a cyclic (i.e., inscribed) quadrilateral, opposite angles are supplementary, i.e., $\angle A_1B_1A_2 + \angle A_1B_2A_2 = 180°$, thus $\angle FB_1A_1 = \angle FB_2A_2$ (Fig. 12). Therefore triangles ΔFA_1B_1 and ΔFA_2B_2 are similar. Thus:

$$\frac{|FA_1|}{|FA_2|} = \frac{|FB_1|}{|FB_2|} = \frac{|A_1B_1|}{|A_2B_2|} \tag{4}$$

Let $d_1 = 2r_1,\ d_2 = 2r_2,\ d_3 = |A_1B_2|,\ d_4 = |A_2B_1|$, then using Eq. (4), we first compute $|FA_1| = d_1(d_1d_3 + d_2d_4)/(d_2^2 - d_1^2)$, then the apex F is computed as:

$$F = A_1 + |FA_1|\underline{B_2A_1} \tag{5}$$

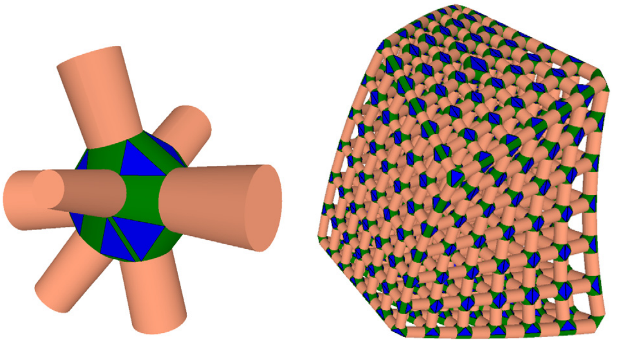


Fig. 13. The CHoCC (left) of an inflated chopped-ball. The ACHoCC approximation of a lattice (right).

5. Approximating a lattice by an ACHoCC

One motivation for this work is to simplify and accelerate the processing of extremely complex lattices represented by parameterized programs [2]. Such a lattice may contain billions of beams and it may not be possible to fully evaluate its boundary or even the parameters of all its elements. Hence, we rely on lazy (on demand) evaluation of a selected subset of the model to support various queries for analysis or printing.

Gupta et al. [2] define a ***lattice*** in terms of the union of ***balls*** and cone ***beams***. A beam F is a solid conical-frustum defined to smoothly connect two balls, B_1 and B_2, such that the union $F \cup B_1 \cup B_2$ is the convex hull of B_1 and B_2.

We assume that two beams that are not incident upon a ball of the lattice are disjoint.

A decomposition of the lattice into an assembly of ***solids*** (***chopped-balls*** and ***chopped-beams***) is discussed in [3]. Each solid is the convex intersection of a primitive (ball or beam) with a set of planar half-spaces. Consequently, the edges of each solid are each a conic section.

We propose below an approximation of these solids for which all edges are either straight lines or circular arcs, a fact that may simplify geometric queries, such as distances from interior points to the boundary of the lattice.

The key idea is to associate an inflated sphere S^+ of radius r^+, with each ball. S^+ has the same center as the ball. r^+ is computed as the smallest radius for which the beams incident on the ball do not interfere outside of S^+. To compute r^+, we consider all pairs of incident beams and for each pair compute a candidate radius using line–line intersection between the profiles of both beams in the plane that contains their axes. Radius r^+ is the maximum of these candidates. For simplicity, we set r^+ to be slightly larger than this maximum, so as to ensure that the caps (or disks) are disjoint.

Replacing each ball by a ball of the same center but of inflated radius adds volume to the junction (Fig. 3), but simplifies the solids involved: the chopped-ball is bounded by a spherical face with holes and by a set of perfect disks. Each chopped-beam is bounded by a cone and by two disks. Each chopped-ball has disk contacts with its incident chopped-beams.

To reduce the added volume introduced by the inflated ball, we replace each of the chopped-balls by the convex hull of its contact-disks. Observe that, now, each element of this new model is a CHoCC. Hence, the whole lattice is an Assembly of CHoCCs (ACHoCCs) (Fig. 13).

To simplify further processing, we assume that all CHoCCs in a lattice are disjoint.

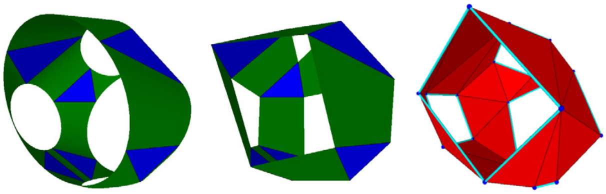


Fig. 14. The CHoCC (shown without its disks, left) has 18 vertices, 6 triangles, and 9 corridors. Its Crude Natural Tessellation (CNT) (center) has 18 vertices, 6 triangles, and 9 quads. The convex hull of uniformly spaced (4 on each circle) samples (right) has 20 vertices and 26 triangles.

6. ACHoCC tessellation

For some applications, it may be useful to produce a watertight, polyhedral approximation of the boundary of a selected portion of an ACHoCC (i.e., of the union of a selected set of its solids). We propose below a direct tessellation that does not require computing the boundary outside of the desired portion of the ACHoCC.

The first approach that we have considered starts by sampling all of the circular edges of the selected portion of the ACHoCC. In this version, the sampling of the two circular edges of each chopped-beam may be coordinated so that the beam will be tessellated by a quad-loop. But the samplings of the circular edges of a chopped-ball are not coordinated. Then, for each solid, it computes the convex hull of the sample points on all the circular edges of that solid. This is trivial for the beams, but requires computing the convex hull of sn cospherical points for a ball with n incident beams, assuming an average of s samples per circle. This approach has higher computational cost and more faces on the polyhedral approximation of the CHoCC of a ball than the approach proposed next.

We advocate an approach in which the vertices on each chopped ball are those of the convex hull of its circular edges. This solution produces the ***Crude Natural Tessellation*** (CNT) of each CHoCC. Fig. 14 compares it to the curved boundary of the CHoCC and to the first approach (above).

The Crude Natural Tessellation (CNT) of a CHoCC is special in many ways. Its vertices are the vertices of the CHoCC, half of its edges (2 of the 4 edges of each corridor) are the edges of the CHoCC, and almost one-third of its faces (number of triangles/number of total faces $= (2n - 4)/(6n - 10) \approx 1/3$ for $n \geq 4$) are the faces of the CHoCC. Furthermore, the quad that approximates each corridor is always planar. Arguably, for a given number of vertices, the CNT has fewer faces than the triangle mesh that bounds the convex hull of samples on the circles. In the sampling approach, the sampling of the two edges of a beam are not coordinated, so the beam tessellation must be computed by a simple algorithm that traces a triangle strip around the beam, choosing for each triangle whether to use the next sample on one edge or on the other, based on angle around the beam axis. In Fig. 15, we show the ACHoCC of a ball with its incident beams, the corresponding CNT, and the individual solids. In Fig. 16, we show the CNT of a steady lattice [2].

In some configuration, CNT may produce unacceptable approximation errors. The CNT of a ball and of its incident beams may differ from the original in two significant ways: 1) its volume may be increased when the CHoCC of the inflated chopped-ball, or the CNT of the CHoCC, is much larger than the original ball, and 2) a significant portion of the original ball may lie outside of the corresponding CHoCC or its CNT. Examples of extreme situations are shown in Fig. 17.

We propose below a sequence of optional post-processing steps that may be used to increase the fidelity and quality of the tessellation.

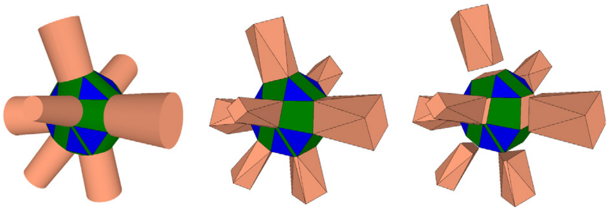


Fig. 15. The ACHoCC (left) of a ball and its incident beams. Its CNT (center) in which each corridor is approximated by a single planar quad. An exploded view of the solids of this ACHoCC.

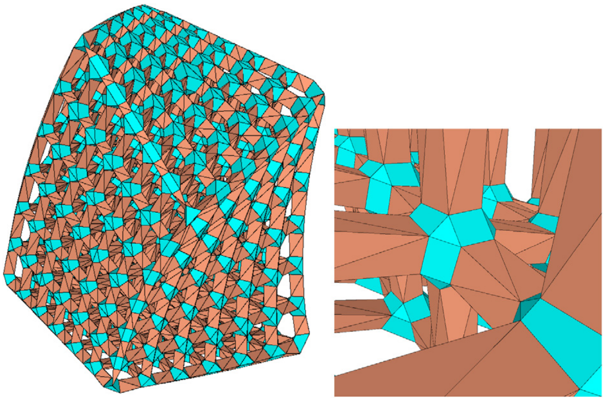


Fig. 16. CNT (left) of steady lattice and a detail (right).

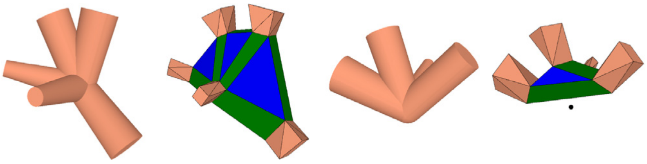


Fig. 17. Original ball and its beams, followed by its CNT, which may have significantly larger volume (left) or may fail to contain a significant portion of the ball (right).

The first step is to refine the tessellation. To do so, we first insert a vertex at the midpoint of each edge of the CNT of CHoCC of each chopped-ball. Then, for each triangle of that CNT, we refine it into four new triangles, where one of the new triangles is incident on the three inserted vertices. Similarly, for each quad (approximating a corridor), we also insert one new vertex at its centroid and refine the quad into eight triangles (as shown in Fig. 18—left).

This refinement step does not change the shape of the mesh, and hence does not improve its approximation accuracy. To improve accuracy, a vertex that was inserted on an edge of a polygonal approximation of a circle is snapped radially onto its closest projection on that circle (see Fig. 18—right). Note that after this process, some flat edges may become concave.

Next, we adjust the positions of the refined and snapped vertices of the CNT associated with the ball. To do this, our central-projection step moves each one of these vertices towards or away from the center of the ball, so as to bring it to the boundary of the exact lattice. A typical result is shown in Fig. 19—left.

The central-projection works properly and is easy to implement when the center of the ball is contained in the CNT of the CHoCC. When this is not the case, we refine the CHoCC by introducing an additional circle of infinitesimal radius on the inflated sphere. To choose where to introduce the circle, we find a face of the natural tessellation of the CHoCC that is visible

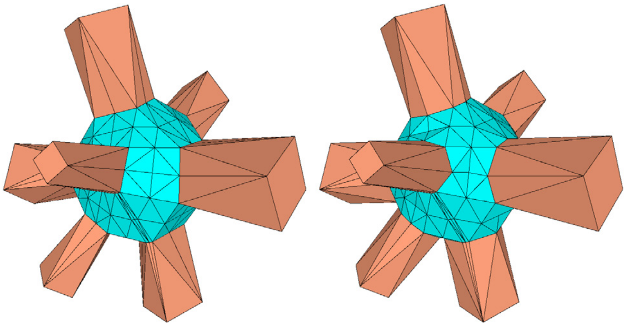


Fig. 18. A refinement (left) of the mesh of Fig. 15—center. The result (right) of snapping each new vertex onto its circle.

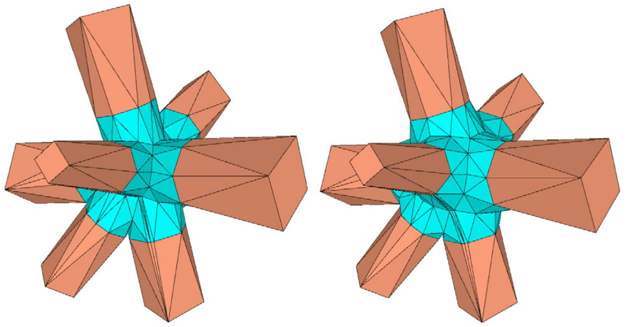


Fig. 19. Result of adjusting the new vertices to the boundary of the lattice (left) or to the fillet surface (right).

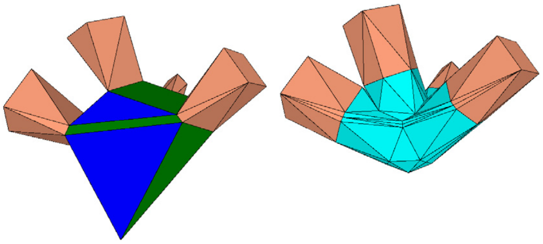
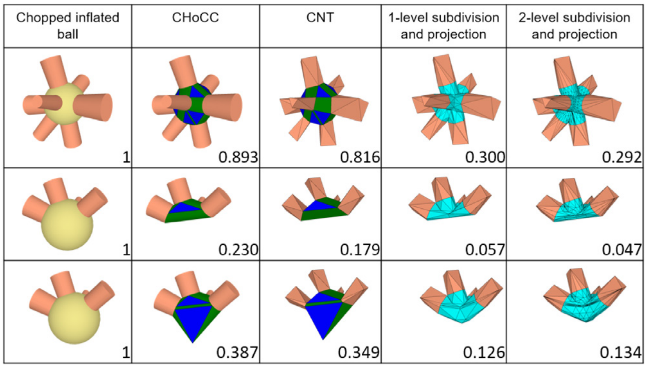


Fig. 20. Adjusted CNT (left) of the configuration in Fig. 17—right and the corresponding result after refinement and central-projection (right).

from the ball's center by checking if the ball's center is in the positive half-space of that face. This face is either a triangle or a planar convex quad. We compute the centroid of this face, and from that centroid, shoot a ray towards the ball's center. The infinitesimal radius circle is then created at the intersection of this ray with the inflated sphere. The rationale justifying this approach is that, given a polygon with its n vertices on a sphere, we can pick any point in the spherical polygon formed by the n diametrically opposite points corresponding to these vertices [22] and the sphere's center will be contained in the convex hull of the given n vertices and this picked point. The result is illustrated in Fig. 20.

In Fig. 21 we compare the volume of 5 different approximations of a junction-element. Notice that the volume of the CHoCC approximation of the junction-element is a fraction of the volume of the chopped ball, thereby reducing the error in volume computation, specially for the configuration in bottom two rows. The CNT then reduces the volume approximation further but the subdivision and projection steps bring it closer to the actual value.

Finally, to reduce stress concentrations, a fillet may be desired, in place of sharp concave edges at the junctions between beams.



Chopped inflated ball	CHoCC	CNT	1-level subdivision and projection	2-level subdivision and projection
1	0.893	0.816	0.300	0.292
1	0.230	0.179	0.057	0.047
1	0.387	0.349	0.126	0.134

Fig. 21. A comparison of five different approximations of a junction-element and its incident beams. The top row shows a configuration with 8 beams wherein the beams are distributed evenly around the junction. The middle row shows a configuration with 4 beams, wherein beams are oriented such that the CNT does not contain the center of the yellow sphere. The bottom row shows the same configuration as the second row but with the CNT adjusted to contain the center of the yellow sphere. Each cell also shows the volume fraction of the junction-element (excluding beam-elements) for that approximation with respect to its chopped-ball approximation in the first column. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

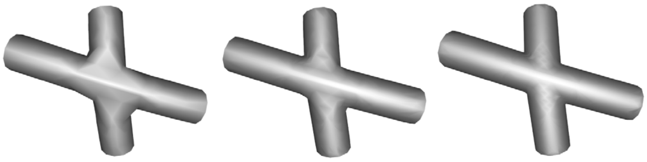


Fig. 22. Results of central-projection after one (left), two (center), and three (right) refinements of the CNT.

The filleted portion of the lattice near one ball may be represented as an implicit ***fillet-surface***, in a form based on Blinn's blobby molecules [23]. The equation for a filleted surface is

$$d(P) = -\frac{\ln \sum_{i=1}^{n} \exp\left(-bd_i(P)\right)}{b} \qquad (6)$$

where $d(P) = 0$ when point P is on the filleted surface, $d_i(P)$ is the signed distance from P to the ith beam (in union with the two balls it connects), and $b > 0$ is the blobbiness parameter where smaller values yield a larger fillet. We implement $d_i(P)$ as the cone-sphere distance computation from Barbier et al. [24].

In the central-projection discussed above, instead of using the boundary of the original lattice, we use the fillet-surface. We use sphere-tracing to solve the ray intersection against the fillet-surface [25] (Fig. 19—right).

We can also subdivide the tessellation further to better approximate the lattice geometry (Fig. 22) or the fillet.

7. Conclusions

We show that the topology and geometry of the convex hull of $n > 2$ disjoint cospherical circles are surprisingly simple. It has $6n - 10$ faces that are each either a disk, a triangle, or a corridor that is a portion of an elliptic cone and is bounded by two line segments and two circular arcs. We suggest a simple and efficient approach for computing the connectivity of that convex hull, its vertices, and the apex of the cone of each corridor. We discuss a possible application for computing an assembly of simple solids that approximates a given lattice by the union of Convex Hulls of Cospherical Circles (CHoCCs). Each CHoCC corresponds to a beam or a junction of the lattice. The CHoCC of each junction touches the CHoCC of each of its incident beams along a disk. But the interiors of these CHoCCs are disjoint. We discuss tessellations of the ACHoCC representation of a lattice that offer different degrees of accuracy.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

We thank Alexander Pasko (Bournemouth University, UK) and his lab for providing the blending function used to generate the filleted version of the junctions. We also thank Helmut Pottmann (TU Wien, Austria) for his comments on how results from Classical Geometry may be used to prove that the convex hull of two cospherical circles is bounded by an elliptic cone.

Appendix A. Supporting plane of three cospherical circles

We want to solve the following system:

$$\cos(\alpha + \alpha_i) = N \cdot N_i, \;\; i = 1, 2, 3$$
$$\|N\| = 1$$

We can write vector triple product of the following term in two different ways:

$$N_1 \times (N \times (N_2 \times N_3)) = (N_1 \cdot (N_2 \times N_3))N - (N_1 \cdot N)(N_2 \times N_3) \tag{A.1}$$
$$N_1 \times (N \times (N_2 \times N_3)) = (N \cdot N_3)(N_1 \times N_2) + (N \cdot N_2)(N_3 \times N_1) \tag{A.2}$$

Equating the RHS of Eqs. (A.1) and (A.2), we get:

$$\begin{aligned} N &= \frac{(N \cdot N_1)(N_2 \times N_3) + (N \cdot N_2)(N_3 \times N_1) + (N \cdot N_3)(N_1 \times N_2)}{N_1 \cdot (N_2 \times N_3)} \\ &= \frac{\cos(\alpha + \alpha_1)(N_2 \times N_3) + \cos(\alpha + \alpha_2)(N_3 \times N_1) + \cos(\alpha + \alpha_3)(N_1 \times N_2)}{N_1 \cdot (N_2 \times N_3)} \\ &= \cos\alpha A - \sin\alpha B \end{aligned}$$

where,

$$\begin{aligned} A &= \frac{\cos\alpha_1(N_2 \times N_3) + \cos\alpha_2(N_3 \times N_1) + \cos\alpha_3(N_1 \times N_2)}{N_1 \cdot (N_2 \times N_3)} \\ B &= \frac{\sin\alpha_1(N_2 \times N_3) + \sin\alpha_2(N_3 \times N_1) + \sin\alpha_3(N_1 \times N_2)}{N_1 \cdot (N_2 \times N_3)} \end{aligned}$$

Let $t = \tan\alpha$, and substitute $\cos\alpha = \frac{1}{\sqrt{1+t^2}}$ and $\sin\alpha = \frac{t}{\sqrt{1+t^2}}$ into expression for N to get:

$$N = \frac{1}{\sqrt{1+t^2}}(A - tB)$$

As $N \cdot N = 1$, we get:

$$(B \cdot B - 1)t^2 - 2A \cdot Bt + A \cdot A - 1 = 0$$

Solving the above equation gives:

$$t = \frac{A \cdot B \pm \sqrt{(A \cdot B)^2 - (A \cdot A - 1)(B \cdot B - 1)}}{B \cdot B - 1}$$

We then back substitute t, to compute $\cos\alpha$, $\sin\alpha$ and N.

Note that in the special case, when the three normals N_1, N_2 and N_3 are coplanar, $N_1 \cdot (N_2 \times N_3) = 0$, we equate the RHS of Eqs. (A.1) and (A.2) and obtain:

$$\cos\alpha A' - \sin\alpha B' = 0 \tag{A.3}$$

where,

$$\begin{aligned} A' &= \cos\alpha_1(N_2 \times N_3) + \cos\alpha_2(N_3 \times N_1) + \cos\alpha_3(N_1 \times N_2) \\ B' &= \sin\alpha_1(N_2 \times N_3) + \sin\alpha_2(N_3 \times N_1) + \sin\alpha_3(N_1 \times N_2) \end{aligned}$$

We solve Eq. (A.3) for α and compute N by solving the linear system of equations $\cos(\alpha + \alpha_i) = N \cdot N_i, \;\; i = 1, 2, 3$.

Appendix B. Stereographic projection and its reverse

Consider a unit radius sphere S centered at the origin. The point of stereographic projection is the north pole $(0, 0, 1)$ and the plane of projection is the plane L through the south pole $(0, 0, -1)$ and perpendicular to the north direction. Let P be a point on the sphere and has coordinates (x_p, y_p, z_p). Let P′ be the stereographic projection of P on L. The coordinates (x'_p, y'_p, z'_p) of P′ are:

$$x'_p = \frac{2x_p}{1 - z_p}, y'_p = \frac{2y_p}{1 - z_p}, z'_p = -1 \tag{B.1}$$

And the coordinates of the reverse projection of P′ on to S are:

$$x_p = \frac{4x'_p}{4 + x'^2_p + y'^2_p}, y_p = \frac{4y'_p}{4 + x'^2_p + y'^2_p}, z_p = \frac{-4 + x'^2_p + y'^2_p}{4 + x'^2_p + y'^2_p} \tag{B.2}$$

For stenographic projection of a circle consider a circle C on sphere S. The center of C is d distance from the origin along the unit direction (a, b, c), hence this circle lies in a plane given by $ax + by + cz - d = 0$ and its radius is $r = \sqrt{1 - d^2}$. Let the circle C′ be the stereographic projection of C on L. If (x_p, y_p, z_p) are the coordinates of a point on C′ then the reverse projection of this point is obtained by Eq. (B.2). We substitute the coordinates of this reverse projection into the equation of the plane and rearrange terms to obtain the following expression:

$$(x_p + \frac{2a}{c - d})^2 + (y_p + \frac{2b}{c - d})^2 = 4(\frac{a^2 + b^2}{(c - d)^2} + \frac{c + d}{c - d}) \tag{B.3}$$

Then, the coordinates of the center (x'_c, y'_c, z'_c) and the radius r'_c of C′ are:

$$\begin{aligned} x'_c = \frac{-2a}{c - d}, y'_c = \frac{-2b}{c - d}, z'_c = -1, \text{ and} \\ r'_c = 2\sqrt{\frac{a^2 + b^2}{(c - d)^2} + \frac{c + d}{c - d}} \end{aligned} \tag{B.4}$$

Now for reverse stereographic projection of a circle, let circle C be the reverse projection of C′ on to S such that C lies in a plane $ax + by + cz - d = 0$. We solve the equations in Eq. (B.1) and that $a^2 + b^2 + c^2 = 1$ to compute a, b, c and d as follows:

$$\begin{aligned} k &= \frac{-4 - r'^2_c + x'^2_c + y'^2_c}{4 - r'^2_c + x'^2_c + y'^2_c} \\ m &= \sqrt{x'^2_c(k - 1)^2 + y'^2_c(k - 1)^2 + 4k^2} \\ a &= \frac{x'_c(1 - k)}{m}, b = \frac{y'_c(1 - k)}{m}, c = \frac{2k}{m}, d = \frac{2}{m} \end{aligned} \tag{B.5}$$

The center of C is d distance from the origin along the unit direction (a, b, c) and its radius is $r = \sqrt{1 - d^2}$.

References

[1] Nash ED, Pir AF, Sottile F, Ying L. Convex hull of two circles in r3. Combin Algebr Geom 2016;297.

[2] Gupta A, Kurzeja K, Rossignac J, Allen G, Kumar PS, Musuvathy S. Programmed-lattice editor and accelerated processing of parametric program-representations of steady lattices. Comput Aided Des 2019.

[3] Gupta A, Allen G, Rossignac J. Exact representations and geometric queries for lattice structures with quador beams. Comput-Aided Des 2019.

[4] Karavelas MI, Yvinec M. Dynamic additively weighted voronoi diagrams in 2d. In: European symposium on algorithms. Springer; 2002, p. 586–98.

[5] Spain B. Analytic quadrics. Pergamon Press; 1960, p. 102–4.

[6] Wang C, Berhan L, Sastry A. Structure, mechanics and failure of stochastic fibrous networks: Part i—Microscale considerations. J Eng Mater Technol 2000;122(4):450–9.

[7] Valdevit L, Godfrey SW, Schaedler TA, Jacobsen AJ, Carter WB. Compressive strength of hollow microlattices: Experimental characterization, modeling, and optimal design. J Mater Res 2013;28(17):2461–73.

[8] Lien J-M, Amato NM. Approximate convex decomposition of polyhedra. In: Proceedings of the 2007 ACM symposium on solid and physical modeling. ACM; 2007, p. 121–31.

[9] Lien J-M. Approximate convex decomposition and its applications, Vol. 69 (Ph.D.-thesis), 2006.

[10] o'Rourke J. Computational geometry in C. Cambridge university press; 1998.

[11] Convex hulls. In: Computational geometry: Algorithms and applications. Berlin, Heidelberg: Springer Berlin Heidelberg; 2008, p. 243–58. http://dx.doi.org/10.1007/978-3-540-77974-2_11.

[12] Hert S, Schirra S. 3D convex hulls. In: CGAL user and reference manual. 4.13 ed.. CGAL Editorial Board; 2018, URL https://doc.cgal.org/4.13/Manual/packages.html#PkgConvexHull3Summary.

[13] Boissonnat J-D, Cérézo A, Devillers O, Duquesne J, Yvinec M. An algorithm for constructing the convex hull of a set of spheres in dimension d. Comput Geom 1996;6(2):123–30.

[14] Geismann N, Hemmer M, Schömer E. The convex hull of ellipsoids. In: Proceedings of the seventeenth annual symposium on computational geometry. ACM; 2001, p. 321–2.

[15] Rappaport D. A convex hull algorithm for discs, and applications. Comput Geom 1992;1(3):171–87.

[16] Coexter HSM. Introduction to geometry. Newyork Wiley; 1969, (pp. 93 and 289-290).

[17] Fortune S. A sweepline algorithm for voronoi diagrams. Algorithmica 1987;2(1–4):153.

[18] Xiong G, Musuvathy S, Fang T. Automated structured all-quadrilateral and hexahedral meshing of tubular surfaces. In: Proceedings of the 21st international meshing roundtable. Springer; 2013, p. 103–20.

[19] Stasiuk D, Piker D. Exoskeleton - bespoke geometry. 2019, http://www.bespokegeometry.com/2014/05/17/exoskeleton/ (Accessed: 01-05-2019).

[20] Srinivasan V, Mandal E, Akleman E, et al. Solidifying wireframes. In *Proceedings of the 2004 bridges conference on mathematical connections in art, music, and science*, 2005.

[21] Bernardini F, Mittleman J, Rushmeier H, Silva C, Taubin G. The ball-pivoting algorithm for surface reconstruction. IEEE Trans Vis Comput Graphics 1999;5(4):349–59. http://dx.doi.org/10.1109/2945.817351.

[22] Howard R, Sisson P. Capturing the origin with random points: Generalizations of a putnam problem. College Math J 1996;27(3):186–92.

[23] Blinn JF. A generalization of algebraic surface drawing. ACM Trans Graph (TOG) 1982;1(3):235–56.

[24] Barbier A, Galin E. Fast distance computation between a point and cylinders, cones, line-swept spheres and cone-spheres. J Graph Tools 2004;9(2):11–9.

[25] Hart JC. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. Vis Comput 1996;12(10):527–45.