Seed Polytopes for Incremental Approximation *

O. Aichholzer† F. Aurenhammer‡ T. Hackl† B. Kornberger† S. Plantinga§ G. Rote¶
A. Sturm¶ G. Vegter§

Abstract

Approximating a given three-dimensional object in order to simplify its handling is a classical topic in computational geometry and related fields. A typical approach is based on incremental approximation algorithms, which start with a small and topologically correct polytope representation (the seed polytope) of a given sample point cloud or input mesh. In addition, a correspondence between the faces of the polytope and the respective regions of the object boundary is needed to guarantee correctness.

We construct such a polytope by first computing a simplified though still homotopy equivalent medial axis transform of the input object. Then, we inflate this medial axis to a polytope of small size. Since our approximation maintains topology, the simplified medial axis transform is also useful for skin surfaces and envelope surfaces.

1 Introduction

Object simplification and surface reconstruction are fundamental tasks in several areas of computer science, like geometric modeling, computer graphics, and computational geometry. We refrain from a general discussion here and refer the reader e.g. to [3, 8, 9, 11] and references therein.

In this note we deal with the problem of computing a simple but topologically correct polytope for a given input object, which is typically presented as a point cloud or surface mesh. As this polytope will serve as a starting point for incremental approximation algorithms, we additionally provide a correspondence between the faces of the polytope and the regions of the object surface. Possible incremental algorithms we have in mind use e.g. elliptical and hyperbolic patches or are based on interpolating subdivision surfaces.

In a previous related approach, point clouds in *convex* position are approximated by spherical patches, using an incremental algorithm [7]. Starting with a very simple structure (a tetrahedron), the convex polytope is incrementally refined until the associated surface built from spherical patches approximates the convex point cloud within a given tolerance bound. The approximating surface consists of a 'bulgy' polytope, where the triangular faces of the polytope are replaced by spherical patches.

The same approach works for other classes of approximating surfaces based on polytopes, such as interpolating subdivision surfaces. For a correct approximation, the underlying polytope needs to have the same topology as the input object. Furthermore, one has to be able to find which part of the object is approximated by a given part of the polytope, as this is the area where we have to test for epsilon-closeness.

1.1 A new approach

Our construction of a small (in general, nonconvex) initial polytope for a given, sufficiently dense sample point cloud is based on a certified simplification of the medial axis transform (MAT). The goal is to represent the object with as few elements as possible. To this end, we use a modification of our previous work [1, 2] where the input object is approximated by a set of balls. This set is then pruned based on an approximation of the minimal set covering problem, thereby carefully choosing the parameters of the original algorithm in order to preserve topology, see Section 2. With slight modifications, this approach can also be used for simplification of skin surfaces [9] and envelope surfaces [11]. The exact medial axis of the pruned set of balls is then computed [5].

In a second step we 'inflate' the simplified medial axis (which, as being defined by a union of balls, is a piecewise-linear object) by replacing it with a combinatorial 2-manifold and moving its vertices back to the input surface, see Section 3.

From experimental results for the medial axis simplification we expect that our approach leads to incremental approximations with significantly fewer patches compared to results achievable when starting directly with the original input set.

*Partially supported by the FWF Joint Research Program 'Industrial Geometry' S9205-N12, and by the IST Programme of the EU as a Shared-cost RTD (FET Open) Project under Contract No IST-006413 (ACS – Algorithms for Complex Shapes)

†Institute for Software Technology, Graz University of Technology, Austria, {oaich,thackl,bkorn}@ist.tugraz.at

‡Institute for Theoretical Computer Science, Graz University of Technology, Austria, auren@igi.tugraz.at

§University of Groningen, Department of Mathematics and Computing Science {simon,gert}@cs.rug.nl

¶Institut für Informatik, Freie Universität Berlin {rote,sturm}@inf.fu-berlin.de

2 Medial axis extraction

Let O be a smooth and boundary-connected object in 3D. We allow O to have tunnels, but 'holes' (empty regions within the object, e.g. bubbles in a Swiss cheese, without connection to the exterior) are excluded. The medial axis of O is the set of centers of all maximal inscribed balls. The local feature size $f(x)$ of a point x on the boundary ∂O of O is the minimum distance from x to any point on the medial axis of O. A finite point set $S \subset \partial O$ is an r-sample [3] of ∂O if every point $x \in \partial O$ has at least one point of S within distance $r \cdot f(x)$. We are interested in a simplified version of the medial axis of O which, nevertheless, retains two important properties: inclusion in the object, and homotopy equivalence. For an example see Figure 1 which shows the discrete MAT of a cow model and its simplified version.

2.1 Ball generation

In a first step we follow well known paths [4] in that we compute the Voronoi diagram of a given sample S of ∂O and extract all inner polar balls. Each point $s \in S$ defines an inner polar ball $b_{c,\rho}$ whose center c is a vertex of the Voronoi cell for s farthest away from s and inside O, and whose radius is $\rho = \delta(c, s)$ (the distance between c and s). Let B be the set of all inner polar balls. As has been shown in [4], the medial axis of the union, $U(B)$, of the balls in B is homotopy equivalent to O as long as S satisfies the sampling condition, that is, S constitutes an r-sample of ∂O for sufficiently small r.

In certain applications we are given not only an unorganized point set S but a triangular mesh representing ∂O and having S as its vertices. This form of input will allow the ball generation algorithm in [1] to work well even if S does not satisfy any sampling condition. Guarantees on the topology are then, of course, lost.

2.2 Pruning

The sampling density of S may cause the set, B, of balls to be quite large, so the medial axis of $U(B)$ is likely to contain many detailed and unwanted features. Therefore we do not directly compute the medial axis of $U(B)$ but perform a pruning of B first. Several pruning criteria based on proximity and angles have been proposed, e.g., in [10, 12]. In the work [1] a method is described that is capable of discarding balls belonging to unstable parts of the medial axis without any geometric criteria. This method can be adapted to keep control over the topology of the medial axis [2]. Loosely speaking, we enlarge all the balls in B and treat them together with S as an instance of the well-known set covering problem, as is briefly described below.

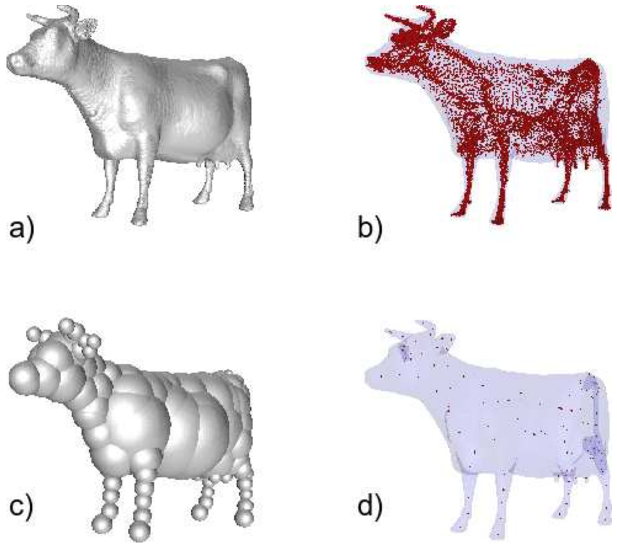

Figure 1: a,b) MAT and its 20108 medial ball centers
c,d) Pruned MAT with 116 ball centers

2.2.1 Ball enlargement

By construction, each ball $b \in B$ contains 4 points of S on its boundary and has no points of S in its interior. From B we now generate a set, B', of co-centric but enlarged balls, each typically covering tens or even hundreds of points of S. Thereby, a requirement important for later purposes is that $U(B')$ and $U(B)$ are topologically equivalent. We use the power diagram $PD(B)$ of B (see Figure 2) to control the proper enlargement of the ball radii. For each ball $b \in B$, its power cell $C(b)$ contains exactly those parts of b's boundary which contribute to $\partial U(B)$, see e.g. [6]. So, if we choose maximal radii such that (1) $PD(B') = PD(B)$ holds, and (2) each $b' \in B'$ intersects the same facets, edges, and vertices of $C(b)$ as does its original b, then the topology of the union of balls does not change. Such radii exist and can be found in time linear in the size of $PD(B)$, by exploiting the well-known polytope lifting of $PD(B)$ in 4D.

2.2.2 Set covering

Now we want to keep an (ideally) minimal subset of the set B' of enlarged balls, such that all points of S are still covered by at least one ball in this subset. This is an instance of the NP-hard set covering problem. In [1] we use a combination of exact and heuristic methods in order to get an almost minimal subset $B_o \subset B'$.

Concerning the topology of the union $U(B_o)$, the set covering step only removes balls from the set B' and thus it will never close tunnels that are present in $U(B')$. (Holes do not exist in $U(B')$ by construction.) However, this step might create holes and tun-

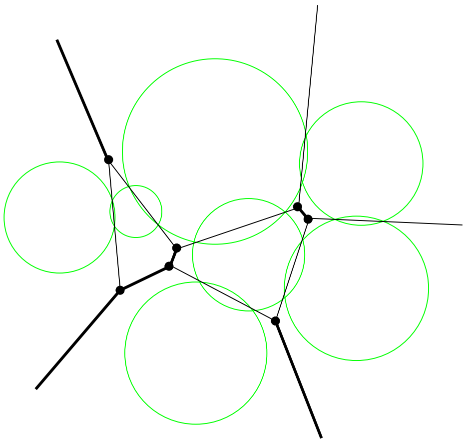

Figure 2: Power diagram and its external graph

nels in $U(B_o)$, and may even make it break apart. We therefore apply a postprocessing where such events are detected and repaired. Again, we make use of a power diagram, $PD(B_o)$, in this case. Note that disconnectedness of $U(B_o)$ can be checked from the dual graph of $PD(B_o)$ right away.

Define the *external power graph*, $G(B_o)$, of B_o as the set of all edges and vertices of $PD(B_o)$ that are completely avoided by $U(B_o)$. (All objects are considered to be topologically closed.) Consult Figure 2, where $G(B_o)$ is drawn with bold lines. Each hole in $U(B_o)$ can be detected by recognizing that $G(B_o)$ contains a respective connected component that is bounded.

To deal with tunnels, two strategies can be applied. One is to avoid tunnels altogether, by a modification of the pruning strategy for the set B' of enlarged balls: Exploiting that the input point cloud S is an r-sample, we (conceptually) shrink each ball $b \in B'$ by $r \cdot f$, where $f = \max_{x \in S \cap b} f(x)$, and execute the set covering as if such balls were present. This may lead to a (moderate) increase of the size of the pruned set B_o. On the other hand, if a mesh on S is present, then we can check for tunnels with its aid, because for each tunnel of $U(B_o)$ there exists at least one edge in $G(B_o)$ that intersects some triangle of the mesh. Starting from each such triangle, we trace $G(B_o)$ inside the mesh until we run out of edges or intersect the mesh again, in which case a tunnel has been detected. Note that most mesh triangles can be excluded from consideration; e.g. all those being covered by a single ball.

3 Construction of the polytope

The main use of a simplified polytope is to supply a good starting configuration for incremental surface approximation algorithms. To this end, we base the construction of this polytope on the pruned,

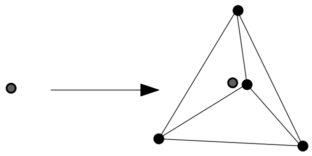

Figure 3: Constructing a pyramid from a vertex

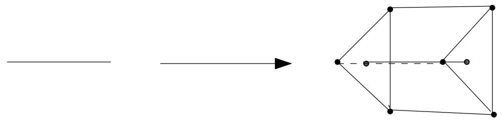

Figure 4: Constructing a tube from a segment

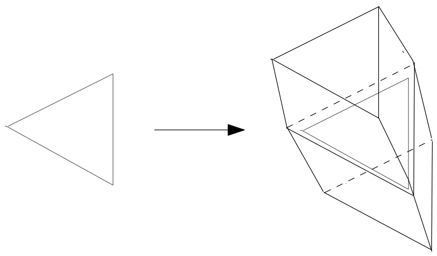

Figure 5: Constructing a polytope from a facet

piecewise-linear medial axis, $M(B_o)$, obtained in the previous section. The basic idea is to blow up $M(B_o)$ to a polytope, P_M, which uses only vertices of the original point cloud S. A main advantage of our construction is that the power cells of B_o give a decomposition of the space into cells, which define for each facet of P_M the neighborhood in which points from S have to be checked for epsilon-closeness to the approximating surface. This is especially important for point clouds not in convex position, since points can be very close to a surface patch then, but lie on the 'opposite' side of the medial axis, so they have to be handled by a different part of the polytope.

Note that a main condition for the constructed polytope is that $M(B_o)$ lies inside it. Therefore no polytope facet intersects the medial axis and no bound on the distance of the original vertices to the facets of the new polytope exists.

To start with, we wrap $M(B_o)$ with a combinatorial 2-manifold mesh. This wrapping will result in a mesh that is topologically equivalent to the boundary of the original input object. As $M(B_o)$ is a piece-wise linear structure, it consist of vertices, segments, and facets with boundary-segments. For each of these features, we construct a polytope feature:

- For a *vertex* we construct a pyramid (Figure 3).
- For a *segment* we construct a tube (Figure 4).

- We double a *facet* and connect it using the features of the boundary segments (Figure 5).

We obtain a combinatorial 2-manifold mesh with its vertices still coinciding with the vertices of $M(B_o)$. The next step is to select a point of S for every vertex of the inflated medial axis P_M to be constructed. We build a cone of size γ which depends on the local feature size of the r-sampling S. Each vertex of P_M is an apex of a cone pointing in the direction of the normal in this vertex. This cone gives the direction in which we move the vertex of the wrapped medial axis towards the object boundary. The cones are chosen in such a way that they do not intersect $M(B_o)$. Moreover, the way we define the cone size (namely, depending on the local feature size) assures that each cone includes at least one point of S. If more than one point of S is included, we choose an arbitrary one. This results in a polytope containing $M(B_o)$ and with vertices chosen from the set S. The facets of this polytope are similar to the *supertriangles* as defined in [7], which can be used as starting facets for any incremental approximation algorithm.

Figure 6 summarizes our polytope construction for a (two-dimensional) point sample.

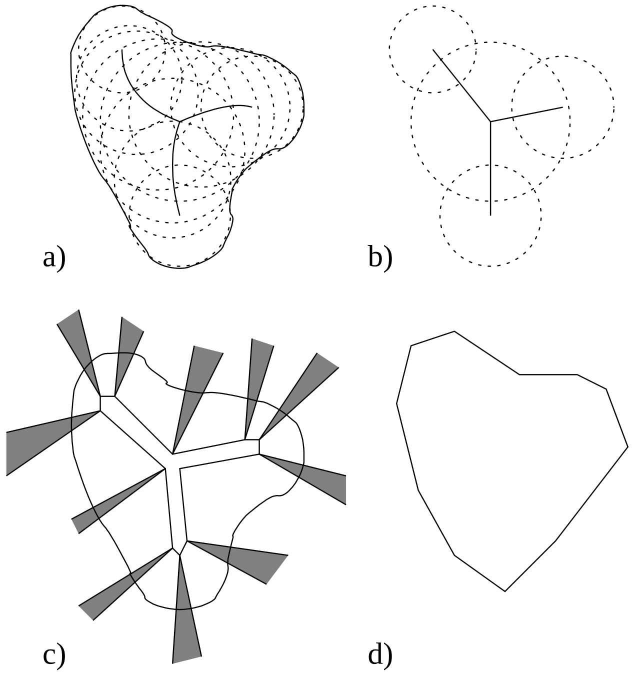

Figure 6: a) Medial axis transform, b) Pruned MAT, c) wrapped MA with cones, d) inflated MA

4 Future Work

For convex objects, the spherical patch algorithm described in [7] can be used together with our setting. For the more general case of non-convex inputs we plan to extend [7] to use a combination of e.g. elliptical and hyperbolic patches, based on the presented framework. Adapting the growing strategy, the first part of our algorithm can also be useful for the skin surface algorithm as well as envelope surfaces.

References

[1] O. Aichholzer, F. Aurenhammer, T. Hackl, B. Kornberger, M. Peternell, H. Pottmann. *Approximating boundary-triangulated objects with balls.* Proc. 23rd European Workshop on Computational Geometry, EuroCG'07, 2007, 130-133.

[2] O. Aichholzer, F. Aurenhammer, T. Hackl, B. Kornberger. *Scaleable piecewise linear approximations of 3D medial axes.* Manuscript.

[3] N. Amenta, M. Bern. *Surface reconstruction by Voronoi filtering.* Discrete & Computational Geometry 22 (1999), 481-504.

[4] N. Amenta, R. Kolluri *Accurate and efficient unions of balls.* Proc. 16th Ann. ACM Symp. Computational Geometry, 2000, 119-128.

[5] D. Attali, A. Montanvert. *Computing and simplifying 2D and 3D continuous skeletons.* Computer Vision and Image Understanding 67 (1997), 261-273.

[6] F. Aurenhammer. *Improved algorithms for discs and balls using power diagrams.* Journal of Algorithms 9 (1988), 151-161.

[7] K. Buchin, S. Plantinga, G. Rote, A. Sturm, G. Vegter. *Convex approximation by spherical patches.* Proc. 23rd European Workshop on Computational Geometry, EuroCG'07, 2007, 26-29.

[8] T.K. Dey. *Curve and surface reconstruction.* In: Handbook of Discrete and Computational Geometry, (J.E. Goodman and J.O'Rourke, eds.), CRC Press, Vol. 2, 2004.

[9] H. Edelsbrunner. *Deformable smooth surface design.* Discrete and Computational Geometry 21 (1999), 87-115.

[10] M. Foskey, M.C. Lin, D. Manocha. *Efficient computation of a simplified medial axis.* Proc. 8th ACM Symp. Solid Modeling and Applications, 2003, 96-107.

[11] N. Kruithof, G. Vegter. *Envelope surfaces* Proc. 22nd Ann. ACM Symp. Computational Geometry, 2006, 411-420.

[12] M. Samozino, M. Alexa, P. Alliez, M. Yvinec. *Reconstruction with Voronoi centered radial basis functions.* Proc. 4th Eurographics Symp. on Geometry Processing, 2006, 51-60.