

Medial Spheres for Shape Approximation

Svetlana Stolpner, Paul Kry and Kaleem Siddiqi

Abstract We study the problem of approximating a solid with a union of overlapping spheres. We introduce a method based on medial spheres which, when compared to a state-of-the-art approach, offers more than an order of magnitude speed-up and achieves a tighter volumetric approximation of the original mesh, while using fewer spheres. The spheres generated by our method are internal to the object, which permits an exact error analysis and comparison with other sphere approximations. We demonstrate that a tight bounding volume hierarchy of our set of spheres may be constructed using rectangle-swept spheres as bounding volumes. Further, once our spheres are dilated, we show that this hierarchy generally offers superior performance in approximate separation distance tests.

1 Introduction

The choice of representation of 3D volumetric data is an important question in motion planning, solid modeling, computer vision, computer graphics, medical imaging and computer aided design. Whereas representations that approximate shape boundaries with triangles are popular, representation of solids as a union of spheres are a valuable alternative. In 1994, Ranjan and Fournier advocated the use of a union of spheres for representing volumetric data and argued that this representation captures the significant features of a shape robustly with respect to noise and resolution [1]. In recent years, a number of applications have made use of sphere-based representations. Particularly, hierarchies of spheres are used for collision detection in [2] because sphere-sphere intersection tests are fast and simple. Sphere approximations are also used for fast soft shadow generation [3] where a low-frequency object representation is sufficient. Additionally, sphere-based representations are used

School of Computer Science and Centre for Intelligent Machines, 3480 University Street, Room 410, Montréal, Québec, Canada, H3A 2A7, e-mail: {sveta,siddiqi}@cim.mcgill.ca, kry@cs.mcgill.ca

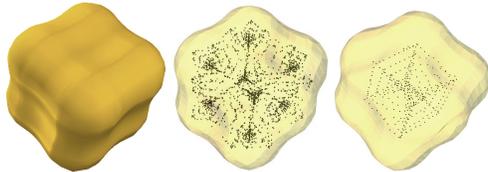


Fig. 1 A mesh (left), sphere centers given by Voronoi vertices (center), sphere centers computed with our method (right). Note that the Voronoi vertices are highly clustered at the centers of the osculating spheres of the 8 corners of the object, while the sphere centers produced with our method are distributed nearly uniformly.

for efficient level-of-detail rendering [4], shape matching [5], and shape deformation [6]. Typically, the boundary of the object is first represented as a triangle mesh, and next the interior of the mesh is approximated using a union of spheres.

In the above applications, it is desirable to generate a sphere-based approximation of a solid that 1) has a small number of spheres, 2) provides a tight fit to the original shape, and 3) is fast to compute. To find the minimum number of spheres that cover a set of points on the object boundary is a hard computational problem. In fact, it can be shown to be NP-hard by reduction from Set Cover [7]. Therefore, to find a small set of approximating spheres to an object boundary heuristics are used.

A popular strategy for approximating an object with a small number of well-fitting spheres is to start by computing the Voronoi spheres of a set of boundary points [2, 7, 8]. As we shall see in our subsequent discussion, the centers of these spheres, the Voronoi vertices, are inherently clustered at rounded corners of objects (*cf.* Fig. 1(center)). Subsequently, to generate a small number of well-fitting spheres, such methods require an optimization step to remove and redistribute spheres, making them computationally expensive.

Recently, Stolpner *et al.* [9] proposed a method to compute an alternative representation of a solid as a union of spheres such that the sphere centers are “well-distributed” on the medial surface of the object. The distribution of the sphere centers generated with this method is shown in Fig. 1 (right). In the current article, we observe that by working directly with this distribution, we avoid the expensive post-processing steps of adding, deleting, or redistributing spheres. In comparison with the state-of-the-art, we show that this sphere-generation method works significantly faster, generates fewer spheres, and provides a tighter fit to the original object in terms of volumetric error.

The spheres that we generate have the special property that they are *internal to the bounding surface* of the object. This fact allows us to quickly compute the exact volumetric error of our approximation and to compare the error with that of a sphere approximation that is not necessarily internal. Further, we demonstrate how our spheres may be quickly dilated in order to provide improved boundary coverage for approximate separation distance computation.

2 Background and Previous Work

We begin by providing some necessary definitions. The complete (generically, infinite) set of spheres whose union corresponds to a solid is given by the Medial Surface Transform. Given a solid $\Omega \subset \mathbb{R}^3$ with boundary \mathcal{B} , the *medial surface* \mathcal{MS} of Ω is the locus of centers of maximal inscribed spheres in Ω . A maximal inscribed sphere in Ω is called a *medial sphere*; the set of all medial spheres is the *Medial Surface Transform* of Ω .

When approximating an object with spheres, a popular strategy is to start with a subset of the spheres of the medial surface transform of the object. A useful tool for doing this is the following discrete analogue of the medial surface transform.

Given a set of point sites $S = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n\}$, $\mathbf{s}_i \in \mathbb{R}^3$, the *Voronoi diagram* of S is a partition of \mathbb{R}^3 into Voronoi cells $V(\mathbf{s}_i)$ with respect to the Euclidean distance d_E , such that

$$V(s_i) = \{\mathbf{x} \in \mathbb{R}^3 \mid d_E^2(\mathbf{s}_i, \mathbf{x}) \leq d_E^2(\mathbf{s}_j, \mathbf{x}), \forall j \neq i\}. \quad (1)$$

A Voronoi cell of a site \mathbf{s}_i is, hence, the set of points that are closer to the site \mathbf{s}_i than to any other site in S . A vertex of $V(\mathbf{s}_i)$ is called a *Voronoi vertex* and is the center of a *Voronoi sphere* that touches 4 or more sites of S , but does not contain any sites in its interior.

When the points S are sampled on the boundary \mathcal{B} of an object Ω , Amenta et al. [10] show that a subset of the Voronoi vertices of S , the *poles*, converges to the medial surface of Ω as the sampling density of S approaches infinity, and hence, the union of a subset of Voronoi spheres converges to Ω . This property is the basis for several methods that approximate a solid with spheres.

We now overview methods that approximate a solid with spheres with the goal of providing a tight approximation to the object using a small number of spheres. An early method for the approximation of an object with spheres [11], bounds each mesh triangle with a sphere but does not necessarily offer a tight fit. Also, a number of methods have been proposed recently for the simplification of sphere representations [7, 12], but these methods do not explicitly share the same goal of ensuring tightness of fit.

There are two methods which approximate an object Ω with tight-fitting spheres, starting with a set of Voronoi spheres. Hubbard [8] greedily selects adjacent Voronoi spheres for merging when their bounding sphere has the best tightness of fit. Bradshaw and O'Sullivan [2] improve the tightness of fit of Hubbard's algorithm by using an adaptive greedy strategy that adds sample points to the boundary of Ω to generate new Voronoi spheres as needed. Tightness of fit is evaluated as the maximum distance between each sphere and Ω . As this quantity is difficult to compute exactly, an approximation is used. Methods that initialize sphere centers to Voronoi vertices of a set of boundary points face the inherent problem that some of these vertices may be clustered; for example, rounded parts of the object boundary generate many Voronoi vertices near the center of the osculating sphere of such regions, as these are points which are equidistant from 4 boundary points.

In more recent work, Wang *et al.* [3] propose a variational optimization method that achieves tighter fits than the method of [2], but is only feasible for approximations having up to a small number of spheres (around 128 spheres). The error measure used is an approximation to total sphere volume outside the object Ω . Approximate volumetric error is also used in [13] to compare the performance of the methods of [8], [2], and an octree-based method for approximating deforming objects with spheres. In the present article we use also use a volume-based error measure.

3 Computation of Spheres

In this section, we describe how we approximate the object Ω with spheres such that the centers of the spheres are “well-distributed” and lie near a subset of the medial surface.

We will compute a set of *approximate* medial point locations using a method based on the analysis of the gradient of the Euclidean distance transform ∇D of Ω described in [9]. This gradient, $\nabla D : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, is a vector field that assigns each point in Ω the direction to its closest point on \mathcal{B} . ∇D is a smooth vector field everywhere inside Ω , except on the medial surface where it is multi-valued. In order to generate a well-distributed set of approximate medial points, we find those voxels of a regular grid that are internal to Ω , analyze the values of the ∇D vector field in each grid cell, and, if the presence of the medial surface is likely, we look for approximate medial point locations in these grid cells. At most one approximate medial point is output per grid cell. This property ensures the “well-distributed” nature of our approximate medial points. To find approximate medial points, we use the property shown in [14], that states that the medial surface intersects a line segment (p, q) interior to Ω , where $q = p + \gamma \cdot \nabla D(p)$, iff $\nabla D(p) \neq \nabla D(q)$, for any scalar value of γ . Refer to Fig. 2, left, for an illustration. This property allows us to find the location of medial points within a user-chosen distance of the medial surface by performing binary search on line segments (p, q) intersected by the medial surface. When the radius of the spheres at the approximate medial points is chosen to be the distance from these points to their nearest boundary points, such spheres are internal and tangent to the boundary \mathcal{B} . We compute nearest points on the mesh boundary to query points

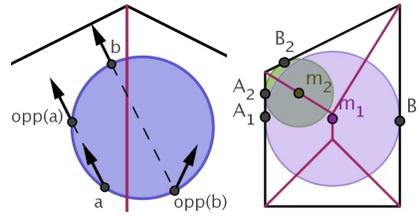


Fig. 2 The boundary of an object is shown in black and its medial surface is in red. Left, arrows show ∇D , the directions to nearest locations on the boundary of the object. In this example, the medial surface intersects the line segment $(b, opp(b))$ because $\nabla D(b) \neq \nabla D(opp(b))$. Right, the object angle of circle m_1 is greater than that of circle m_2 since $\angle A_1 m_1 B_1 > \angle A_2 m_2 B_2$.

analyze the values of the ∇D vector field in each grid cell, and, if the presence of the medial surface is likely, we look for approximate medial point locations in these grid cells. At most one approximate medial point is output per grid cell. This property ensures the “well-distributed” nature of our approximate medial points. To find approximate medial points, we use the property shown in [14], that states that the medial surface intersects a line segment (p, q) interior to Ω , where $q = p + \gamma \cdot \nabla D(p)$, iff $\nabla D(p) \neq \nabla D(q)$, for any scalar value of γ . Refer to Fig. 2, left, for an illustration. This property allows us to find the location of medial points within a user-chosen distance of the medial surface by performing binary search on line segments (p, q) intersected by the medial surface. When the radius of the spheres at the approximate medial points is chosen to be the distance from these points to their nearest boundary points, such spheres are internal and tangent to the boundary \mathcal{B} . We compute nearest points on the mesh boundary to query points

quickly using a version of the PQP software [15] modified for efficient point-to-mesh distance queries. Additional details of the algorithm can be found in [9].

Suppose that a medial point $m \in \mathcal{MS}$ is equidistant from two points $A, B \in \mathcal{B}$. Angle $\angle AmB$ (see Fig. 2, right) is the *object angle*. The complete medial surface of a polyhedron can be a complex structure; for example, it includes bisectors of each pair of mesh faces that meet convexly. Our goal is to locate a few medial spheres such that the volume of their union approximates the volume of Ω well. As proven in [16], removal of spheres having a small object angle has a small impact on the volume of the reconstructed object (refer to Fig. 2, right). As we will be looking for a set of spheres whose volume is close to that of the original solid, we will look for those medial points having a large object angle. For an approximate medial point computed, only one point of tangency is known exactly (the nearest point on \mathcal{B} to the approximate medial point), while the other point is approximated. In our experiments, we use a threshold of 0.6 radians for the object angle.

Figure 1 shows a distribution of approximate medial points computed with our method and that computed with a Voronoi-based method of [10]. Voronoi-based methods approximate the medial surface using a set of boundary samples as input and the density of the medial surface approximation depends on the distribution of the boundary points. In contrast, the distribution of medial points produced by our distance-based method depends on the sampling strategy of the interior of the object and allows one to control the density of the medial points produced.

The larger the grid resolution, the larger the radius of the smallest sphere computed. As small spheres contribute less to the volume of the reconstructed object than large spheres, by choosing a large grid resolution and a large object angle threshold, we generate a small number of spheres that provide good volumetric object coverage, as shown experimentally in Section 4.3. A possible limitation of our approach is that to approximate objects with long narrow parts well we would have to use a fine grid, resulting in a large number of spheres. Such objects may be better approximated with primitives other than spheres, such as oriented bounding boxes or rectangle-swept spheres.

Although fast approximate GPU-based methods for distance field computation exist [17, 18], our use of exact point-to-mesh distance allows us to compute spheres that are guaranteed to be interior to the object boundary. This property is essential to allow efficient comparison of volumetric error in Section 4.3.

4 Volumetric Error for Unions of Spheres

In this section, we evaluate the quality of our sphere-based approximation. In Section 4.1, we compute lower bounds on the volumetric error of the approximation of an object with a union of spheres. We discuss the combinatorial structure of a union of spheres in Section 4.2 and review a formula for the exact computation of the volume of a union of spheres. We use the error bounds and this formula to compare volumetric error of our approximation and that of a leading method in Section 4.3.

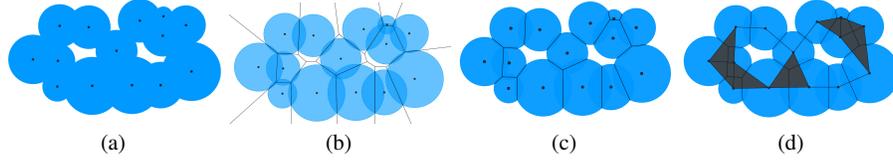


Fig. 3 (a) A union of a set of disks; (b) its power diagram overlaid; (c) the union of the set of disks decomposed using the power diagram; (d) the dual of the decomposition.

4.1 Volumetric Error: Exact or Lower Bound

We evaluate tightness of fit of an approximation of Ω with a set of spheres S as the volume of S outside Ω plus the volume of Ω outside S . Let $\text{vol}(\cdot)$ denote volume and \bar{A} be the complement of a set A . Then the error of the approximation of Ω with S is given by:

$$\text{Err}_\Omega(S) = \text{vol}(S \cap \bar{\Omega}) + \text{vol}(\Omega \cap \bar{S})$$

It is easy to show that

Lemma 1. $\max\{\text{vol}(S) - \text{vol}(\Omega), \text{vol}(\Omega) - \text{vol}(S)\} \leq \text{Err}_\Omega(S)$ and when $\text{vol}(S \cap \bar{\Omega}) = 0$, $\text{vol}(\Omega) - \text{vol}(S) = \text{Err}_\Omega(S)$.

4.2 Unions of Spheres: Tools

The combinatorial structure of a union of spheres has been the subject of much recent research. In this brief overview, let us define a special space filling diagram for a set of sphere sites, called the *power diagram* [19]:

Given a set of spheres, $S = \{(\mathbf{c}_1, r_1), (\mathbf{c}_2, r_2), \dots, (\mathbf{c}_n, r_n)\}$, with centers $\mathbf{c}_i \in \mathbb{R}^3$ and radii $r_i \in \mathbb{R}$, the *power diagram* of S , denoted $PD(S)$, is a partition of \mathbb{R}^3 into convex *power cells* $P(s_i)$, such that

$$P(s_i) = \{\mathbf{x} \in \mathbb{R}^3 \mid d_E^2(\mathbf{c}_i, \mathbf{x}) - r_i^2 \leq d_E^2(\mathbf{c}_j, \mathbf{x}) - r_j^2, \forall j \neq i\}. \quad (2)$$

Power cells, eq.(2), are identical to Voronoi cells, eq.(1), when all the sphere radii are the same. Figure 3 shows a 2D example of the power diagram of a set of disks.

Consider the decomposition of the union of spheres by intersecting each sphere with its power cell. The dual of this decomposition, $DC(S)$, is a simplicial complex that captures the topology of S . As shown in [20], to find the total volume of the union of spheres s_i , $\text{vol}(\cup_i s_i)$, one need only consider the spheres corresponding to vertices, edges, triangles and tetrahedra of $DC(S)$: $\text{vol}(\cup_i s_i) = \sum_{i \in DC(S)} \text{vol}(s_i) - \sum_{(i,j) \in DC(S)} \text{vol}(s_i \cap s_j) + \sum_{(i,j,k) \in DC(S)} \text{vol}(s_i \cap s_j \cap s_k) - \sum_{(i,j,k,l) \in DC(S)} \text{vol}(s_i \cap s_j \cap s_k \cap s_l)$ We use this formula to compute the exact volume of a given union of spheres S , $\text{vol}(S)$. Since computing $\text{vol}(\Omega)$ is easy, using the result of Lemma 1,

we can compute either the exact volumetric error of an approximation of a solid with spheres or a lower bound on the error.

4.3 Experimental Results

												
$ S_V =$	512	482	506	348	496	497	493	498	484	431	474	503
$ S_D =$	344	367	249	163	393	406	468	436	233	108	179	396
$\text{nerr}(S_V) \geq$	0.035	0.015	0.17	0.021	0.266	0.054	0.284	0.051	0.018	0.016	0.017	0.050
$\text{nerr}(S_D) =$	0.034	0.010	0.133	0.009	0.195	0.052	0.178	0.051	0.016	0.014	0.016	0.046
$\text{Time}(S_V) =$	2342	7428	1890	25698	1930	2672	1515	1581	3078	3421	4049	9479
$\text{Time}(S_D) =$	45	184	15	98	30	29	14	107	75	77	134	51

Table 1 Sphere sets S_D are generated with our distance-based method, while sphere sets S_V are generated with a Voronoi-based method of [4]. Timings are given in seconds. S_D provides a tighter approximation using fewer spheres significantly faster than S_V . Volumetric error evaluates the global quality of fit of the sphere approximation. Shown is the smallest computed set of spheres S_D such that the volumetric error of S_D is smaller than S_V .

Given a representation of an object using its triangulated boundary, applications such as soft shadow generation can benefit from an alternative representation of this object using a small number of well-fitting spheres. We use our method described in Section 3 to compute a set of internal spheres whose centers are “well-distributed” on the medial surface of the object and then use tools developed in Section 4.3 to evaluate the exact volumetric error of our approximation with respect to the original boundary. Recall that approximate volumetric error has been used in the literature [3, 13] to evaluate the quality of sphere approximations and provides a global measure of fit. Wang *et al.* [3] provides a tighter fit than [2], but is only feasible for generating small sphere sets (approximately 128 spheres). Reference [2] is a state-of-the-art method for approximating objects tightly with a large number of spheres. As our method is able to generate a large number of spheres quickly, we compare it to that of [2].

Let S_D be the approximation to Ω computed with our distance-based method described in Section 3 and let S_V be the approximation to Ω computed with the Voronoi-based AMAA method of [2]. As explained in Section 3, the sphere set S_D is completely contained inside Ω , *i.e.*, $\text{vol}(S_D \cap \overline{\Omega}) = 0$. By Lemma 1, $\text{vol}(\Omega) - \text{vol}(S_D)$ gives the exact error of the approximation of Ω with S_D , while $\text{vol}(S_V) - \text{vol}(\Omega)$ is a lower bound on the error of the approximation of Ω with S_V . We will define a normalized error $\text{nerr}(S) = \text{Err}_\Omega(S)/\text{vol}(\Omega)$. Thus, $\text{nerr}(S_D) = (\text{vol}(\Omega) - \text{vol}(S_D))/\text{vol}(\Omega)$ and $\text{nerr}(S_V) \geq (\text{vol}(S_V) - \text{vol}(\Omega))/\text{vol}(\Omega)$.

We tabulate the exact error of S_D given by $\text{nerr}(S_D)$, and a lower bound on the error of S_V , $\text{nerr}(S_V)$, in Table 1. Note that the set of spheres S_V does not necessar-

ily contain the object Ω being approximated, so the computation of exact error in this case is a challenging computational problem. In finding the appropriate set of spheres S_D , the voxel resolution was iteratively increased until the error of the approximation given by our method was smaller than that of AMAA. Table 1 compares the performance of the two methods for a number of models for a fixed maximum number of AMAA spheres. Timings are shown on a 3.6 GHz Pentium IV processor with 3 GB of RAM. As can be seen from Table 1, our method generates a set of tighter fitting spheres significantly faster than AMAA.

AMAA construction proceeds top-down by building a hierarchy of a fixed branching factor and depth. For the timings in Table 1, S_V are the leaves of an 8-ary hierarchy of depth 4. Fast construction of a tight binary hierarchy for the spheres S_D is discussed in Section 5.2.

Note that our “well-spaced” distribution of sphere centers is suboptimal in certain cases; for example, for the ellipsoid model in Table 1, a distribution with fewer small spheres and more large spheres would provide better volumetric coverage than our distribution. However, we observe that for this model as well, our method performs better than that of [2].

5 Approximate Separation Distance

For those objects whose surface can be tightly approximated as a surface of a union of a small number of spheres relative to the number of surface triangles, separation distance computations can be accelerated by working with the sphere representation instead of the boundary mesh representation. Further, pairwise distance tests are significantly faster between spheres than triangles. In this section, we show how to quickly grow our set of spheres so as to improve boundary coverage, and how to construct a tight bounding volume hierarchy of the set of spheres. We then evaluate the performance of our sphere approximation against that of the state-of-the-art method AMAA of [2] for approximate separation distance computation.

5.1 Improving Boundary Coverage by Conservative Dilation

In Section 4, we aimed to fill the volume of the object being approximated tightly. However, in this section, we require our set of spheres to provide a tight approximation to the boundary of the object. We show a simple heuristic for conservatively growing and adding spheres that allows us to achieve a more favourable approximation to the boundary.

The surface of the union of spheres S is a non-differentiable object. As explained in Section 4.2, sphere adjacency in the union of spheres S is described by the dual complex $DC(S)$ of the power diagram $PD(S)$. In $DC(S)$, edges correspond to pairs of spheres intersecting along circular arcs on the surface of the union of spheres, while

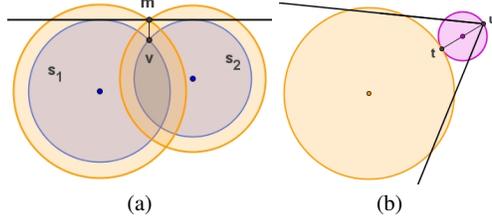


Fig. 4 (a) The blue circles s_1 and s_2 intersect at point v , whose nearest point on the boundary is m . The radii of both blue circles are increased to create orange circles that contain m . (b) To cover the corner of the object at u , we add a new purple circle with center along the line segment (u, t) .

triangles correspond to triplets of spheres intersecting at vertices on the surface of the union of spheres. We can solve for the locations of these vertices, called *v-points*, by considering all triangles (s_i, s_j, s_k) in $DC(S)$, finding the 2 intersection points of spheres s_i, s_j and s_k , and retaining those intersection points that lie on the boundary of the union of spheres. We also note which triplet of spheres contributed to the creation of a *v-point*. For a *v-point* v , we consider the *nearest location on the mesh boundary* to v . The radius of each sphere s is increased sufficiently so as to cover the nearest mesh point to each of the *v-points* that is created by s . Figure 4(a) presents a 2D example. For those edges of $DC(S)$ that are not part of any triangles of $DC(S)$, we sample 2 random points on the circle of intersection of the two corresponding spheres and proceed similarly as with *v-points*.

In order to further improve the fit of the union of spheres S , we introduce new spheres at those vertices of the mesh boundary whose *nearest location on the set of spheres* is more than a tolerance away—see Fig. 4(b). Let \mathbf{u} be a vertex of the mesh boundary \mathcal{B} and \mathbf{t} be the nearest point on S to \mathbf{u} . We initially place a new sphere $s = (\mathbf{c}, r)$ with radius $r = \|\mathbf{t} - \mathbf{u}\|_2 / 2$ centered at $\mathbf{c} = \mathbf{u} + r(\mathbf{t} - \mathbf{u}) / \|\mathbf{t} - \mathbf{u}\|_2$.

5.2 Hierarchy Construction using Rectangle-Swept Spheres

In order to use our set of spheres S to perform fast proximity queries, we fit a bounding volume hierarchy to the set of spheres S . We observe that the medial surface simplified by object angle is often composed of relatively flat sheets, along which the radius of the medial spheres varies smoothly. A Rectangle-Swept Sphere (RSS) (introduced in [15], see Fig. 5) is therefore a suitable bounding volume for medial spheres. The hierarchy is constructed top-down, each parent has 2 children, and each RSS is fitted to the set of leaf spheres it bounds. We find the orientation of the RSS rectangle and the partitioning plane using a linear



Fig. 5 Rectangle-Swept Sphere.

least square fit. The radius is found by using a procedure similar to that in [15].

Alternatively, one may compute a bounding volume hierarchy using spheres as the bounding volume. Given our sphere set approximation, we compute the hierarchy using a similar top-down procedure as when using RSSs as the bounding volume. We compute the minimum bounding sphere of a set of spheres using the algorithm presented in [21].

5.3 Experimental Results

	Size	Timings		Error	
		SH	RSSH	Ave.	Max.
 S_D^+	520	0.524	0.085	0.012	0.074
S_V	498	0.268	0.088	0.024	0.063
 S_D^+	237	0.152	0.061	0.082	0.332
S_V	397	0.502	0.084	0.104	0.454
 S_D^+	830	0.582	0.139	0.033	0.114
S_V	831	0.239	0.143	0.053	0.226
 S_D^+	379	0.255	0.067	0.009	0.104
S_V	379	0.302	0.064	0.067	0.278
 S_D^+	293	1.024	0.171	0.019	0.056
S_V	420	0.327	0.208	0.036	0.113
 S_D^+	296	0.385	0.066	0.009	0.024
S_V	439	0.133	0.123	0.020	0.075
 S_D^+	772	0.405	0.109	0.040	0.188
S_V	695	0.279	0.145	0.106	0.587

Table 2 Timing and error results for inter-object distance tests for our updated sphere sets S_D^+ and sphere sets S_V of [2]. Average timings per frame are recorded in milliseconds. SH uses a sphere hierarchy fitted to the sphere set, while RSSH uses a Rectangle-Swept Sphere hierarchy. The maximum dimension of the bounding box for all objects is 10.

We evaluate the usefulness of our sphere approximation for approximate separation distance computation in terms of computation time and accuracy. In our experiments, we perform a physics simulation by allowing pairs of the same object to collide in a box with random gravity and measure inter-object distance both approximately using sphere approximation and exactly using the meshes at each frame. Such a simulation considers pairs of non-intersecting objects at a variety of separation distances and relative orientations.

We compare the performance of approximating separation distance for the set of spheres computed with our method and that computed with the AMAA method of [2]. We grow and add spheres to our initial sphere set S_D using the strategy outlined in Section 5.1 to create a new sphere set S_D^+ . This step takes one to five seconds for the models in Table 2. We compute both an RSS hierarchy and a sphere

hierarchy of our set of spheres using the method described in Section 5.2. AMAA constructs a bounding volume hierarchy where bounding elements are spheres. Table 1 lists timings for building a hierarchy with branching factor eight. In proximity query experiments, we consider hierarchies with branching factor two. Computation of a binary AMAA hierarchy for the models shown in Tables 2 takes significantly longer: from one hour (for the knot model) to 11 hours (for the peanut model). In addition to a binary sphere hierarchy, we also construct an RSS hierarchy of the AMAA spheres. Constructing an RSS hierarchy for the sphere sets considered takes a fraction of a second on average.

We compare average per-frame time for performing approximate separation distance tests using our sphere approximations and those produced by AMAA, where both sphere hierarchies (SH) and RSS hierarchies (RSSH) are considered. These results are summarized in Table 2. Timings are shown for a 3.6 GHz Pentium 4 CPU with 3 GB of RAM. In our experiments, we find that building an RSS hierarchy of spheres significantly improves distance query time compared to using a sphere hierarchy for both our and AMAA sphere sets, as the RSS offers greater tightness of fit. For most models, we see that the fastest performance is achieved by using our sphere sets and the RSS hierarchy. We believe that because our spheres are nearly medial, RSS bounding volumes provide them with a particularly tight fit.

We evaluate the inter-object distance error as the difference between the exact distance between the meshes and the distance computed using the sphere set approximation. Table 2 presents error statistics for the different sphere sets. Our sphere sets display a smaller average error and generally smaller maximum error than those of AMAA. When exact separation distance is not needed, for those objects that may be approximated well using a small number of spheres relative to the number of boundary triangles, sphere approximations can offer significant speed up at a small cost in quality.

6 Conclusions

We have described a method to compute a tight-fitting union of spheres approximation to a solid without an expensive optimization step that improves the sphere distribution. As such, our method is significantly faster than existing methods and can be used to generate sphere set approximations with a larger number of spheres than previously possible. We have shown experimentally that in comparison with the state-of-the-art AMAA method, our method generates fewer spheres, has a smaller volumetric error, and is significantly faster. Because our spheres are internal to the object, it is possible to evaluate the volumetric error of our approximation exactly and compare the quality of the approximation with a method that generates a set of spheres that are not necessarily internal and do not cover the object boundary. We experimentally demonstrate the benefit of using the rectangle-swept sphere bounding volume for building a hierarchy of approximately medial spheres. Using this hierarchy, we have shown that our sphere sets generally perform faster and more

accurately than those of the AMAA method for approximate separation distance computation. There is a wide variety of additional applications where our tight fitting sphere approximations can be valuable, including path planning, point location, shadow generation, shape matching, mesh deformation, shape morphing, shape segmentation, and approximate Minkowski sums.

References

1. Ranjan, V., Fournier, A.: Volume models for volumetric data. *Computer* **27**(7), 28–36 (1994)
2. Bradshaw, G., O’Sullivan, C.: Adaptive medial-axis approximation for sphere-tree construction. *ACM Transactions on Graphics* **23**(1), 1–26 (2004)
3. Wang, R., Zhou, K., Snyder, J., Liu, X., Bao, H., Peng, Q., Guo, B.: Variational sphere set approximation for solid objects. *Visual Computer* **22**(9), 612–621 (2006)
4. Rusinkiewicz, S., Levoy, M.: QSplat: A multiresolution point rendering system for large meshes. In: *ACM SIGGRAPH*, pp. 343–352 (2000)
5. Shamir, A., Sotzio, A., Cohen-Or, D.: Enhanced hierarchical shape matching for shape transformation. *International Journal for Shape Modeling* **9**(2), 203–222 (2003)
6. Yoshizawa, S., Belyaev, A., Seidel, H.P.: Skeleton-based variational mesh deformations. *Computer Graphics Forum* **26**(3), 255–264 (2007)
7. Aichholzer, O., Aurenhammer, F., Kornberger, B., Plantinga, S., Rote, G., Sturm, A., Vegter, G.: Recovering structure from r -sampled objects. *Computer Graphics Forum* **28**(5), 1349–1360 (2009)
8. Hubbard, P.M.: Approximating polyhedra with spheres for time-critical collision detection. *ACM Transactions on Graphics* **15**(3), 179–210 (1996)
9. Stolpner, S., Whitesides, S., Siddiqi, K.: Sampled medial loci and boundary differential geometry. In: *ICCV Workshop on 3-D Digital Imaging and Modeling*, pp. 1855–1862 (2009)
10. Amenta, N., Choi, S., Kolluri, R.: The Power Crust, Unions of Balls, and the Medial Axis Transform. *Computational Geometry: Theory and Applications* **19**(2-3), 127–153 (2001)
11. Quinlan, S.: Efficient distance computations between non-convex objects. In: *International Conference on Robotics and Automation*, pp. 3324–3329 (1994)
12. Giesen, J., Miklos, B., Pauly, M., Wormser, C.: The scale axis transform. In: *Symposium on Computational Geometry*, pp. 106–115 (2009)
13. Garcia, M., Bayona, S., Toharia, P., Mendoza, C.: Comparing sphere-tree generators and hierarchy updates for deformable objects collision detection. In: *International Symposium on Visual Computing*, pp. 167–174 (2005)
14. Stolpner, S., Whitesides, S.: Medial axis approximation with bounded error. In: *International Symposium on Voronoi Diagrams*, pp. 171–180 (2009)
15. Larsen, E., Gottschalk, S., Lin, M.C., Manocha, D.: Fast proximity queries with swept sphere volumes. Tech. rep., University of North Carolina at Chapel Hill (1999)
16. Foskey, M., Lin, M.C., Manocha, D.: Efficient computation of a simplified medial axis. In: *Solid Modeling and Applications*, pp. 96–107 (2003)
17. Sud, A., Govindaraju, N.K., Gayle, R., Manocha, D.: Interactive 3d distance field computation using linear factorization. In: *ACM Symposium on Interactive 3D Graphics and Games*, pp. 117–124 (2006)
18. Rong, G., Tan, T.S.: Jump flooding in GPU with applications to Voronoi diagram and distance transform. In: *ACM Symposium on Interactive 3D graphics and Games*, pp. 109–116 (2006)
19. Aurenhammer, F.: Power diagrams: properties, algorithms and applications. *SIAM Journal of Computing* **16**(1), 78–96 (1987)
20. Edelsbrunner, H.: The union of balls and its dual shape. In: *Symposium on Computational Geometry*, pp. 218–231 (1993)
21. Gärtner, B.: Fast and robust smallest enclosing balls. In: *European Symposium on Algorithms*, pp. 325–338 (1999)