Shape Simplification Based on the Medial Axis Transform

Roger Tam* Wolfgang Heidrich †

Imager Computer Graphics Laboratory
Department of Computer Science
University of British Columbia

Surface Reconstructed from Boundary Points | Original Medial Axis | Simplified Medial Axis | Strongly Simplified Medial Axis | Surface Reconstructed from Strongly Simplified Axis

Abstract

We present a new algorithm for simplifying the shape of 3D objects by manipulating their medial axis transform (MAT). From an unorganized set of boundary points, our algorithm computes the MAT, decomposes the axis into parts, then selectively removes a subset of these parts in order to reduce the complexity of the overall shape. The result is a simplified MAT that can be used for a variety of shape operations. In addition, a polygonal surface of the resulting shape can be directly generated from the filtered MAT using a robust surface reconstruction method. The algorithm presented is shown to have a number of advantages over other existing approaches.

CR Categories: I.3.5 [Computing Methodologies]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

Keywords: medial axis transform, shape simplification, topology preservation

1 Background and Introduction

The *medial axis transform* (*MAT*) is a shape model that represents an object by the set of maximal balls that are completely contained within the object. For a continuous object this set is infinite. The medial axis (often called the medial *surface* in 3D) consists of the centres of the balls, and can be intuitively thought of as the *skeleton* of the object. The MAT has numerous applications in visualization (*e.g.*, [Pizer et al. 1996; Näf et al. 1997; Paik et al. 1998]), computer graphics (*e.g.*, [Hubbard 1996; Bloomenthal 2002]) and computer vision (*e.g.*, [Ogniewicz 1994; Siddiqi et al. 1999]).

The usefulness of the MAT has inspired many methods for its computation. In most cases, the algorithm operates on a discrete approximation of the object, such as a set of sample boundary points, and outputs a polygonal approximation of the medial axis. It is well known that the medial axis is very sensitive to small perturbations of the object's boundary, and many *regularization* methods have been proposed to remove spurious components associated with noise or other artifacts. Many of these methods also aim to preserve the topology of the axis during the pruning process.

In 2D, the preservation of topology during processing of the medial axis is relatively easy to achieve because the axis is one-dimensional and structurally hierarchical, thereby providing a natural processing sequence when progressing from the outer branches toward the inside. In addition, any cycles in the axis indicate the presence of real loops in the shape of the object and can be easily preserved. As a result, a number of very effective algorithms exist (*e.g.*, [Attali et al. 1995; Ogniewicz 1995; Shaked and Bruckstein 1998; Tam and Heidrich 2002]), and the problem is more or less considered solved. Most of these methods utilize a *pruning* technique in which certain branches are shortened or removed according to their significance.

In 3D, the situation is more complicated. The relationship between components of the medial axis is much more complex. There are many cycles that do not represent loops in the object. There is no natural processing order and there are usually many different deletion sequences possible. In addition, there is often a mutual dependency between skeletal components where the removal of one component can change the topological relationship between others. The result is that the simplification algorithm must impose a processing order and perform explicit topology checks as components are removed. A number of regularization schemes have been proposed (*e.g.*, [Attali and Montanvert 1997; Näf et al. 1997; Amenta et al. 2001a; Dey and Zhao 2002]), each with their own advantages and limitations.

*e-mail: rtam@cs.ubc.ca
†e-mail: heidrich@cs.ubc.ca

IEEE Visualization 2003,
October 19-24, 2003, Seattle, Washington, USA
0-7803-8120-3/03/$17.00 ©2003 IEEE

We present a new algorithm that has a number of advantages over other existing approaches. Our parts-based approach, described in Section 4, allows the MAT to be simplified to a much greater degree without certain undesirable effects such as the disintegration caused by some methods that operate on lower order primitives. This makes the approach suitable for a variety of applications ranging from noise removal to manual modelling. In addition, we can use the connectivity of the parts to efficiently preserve the topology of the axis during simplification, a goal unmet by most other 3D medial axis techniques. Also, our method allows the user to control the degree of simplification using simple, visually intuitive parameters. Finally, we have designed our algorithm to fit very well into an existing surface reconstruction framework, so that the filtered MAT can be used directly to generate an accurate polygonal representation. The reconstruction algorithm, called the *power crust* ([Amenta et al. 2001a]), uses an approximate MAT to compute an interpolating surface from boundary point samples.

It is important to note that the goals of our algorithm differ significantly from the many *mesh simplification* algorithms (see [Heckbert and Garland 1997; Luebke 2001] for examples) whose primary aim is to minimize the number of polygons in a model given certain constraints such as storage size and image resolution. Such algorithms do not necessarily simplify the shape of the object. In contrast, our algorithm focuses on the reduction of shape features in the object by removing parts of the underlying shape model (*i.e.*, the medial axis). The primary goal of our work is to produce a *feature-based* simplification algorithm that does not adversely affect the integrity of the remaining components.

2 Related Work

An approach used by many researchers uses the Voronoi diagram (and/or the dual Delaunay tetrahedralization) of a set of sample points on the object's boundary to approximate the medial axis. This method is suitable for many applications as sample points are typically readily available (*e.g.*, laser scans) or easily derived. Our algorithm follows this approach.

There are only a few methods for 3D medial axis regularization that preserve the topology of the object. Because of the lack of a hierarchical structure, a *peeling* approach is usually employed in which the outermost components are removed one layer at a time. Two of the most notable techniques are by Attali and Montanvert [1997] and Näf *et al.* [1997]. Both of these methods begin with the entire set of interior Delaunay tetrahedra, and delete them one layer at a time according to some criteria for maintaining topological consistency. The main problem with such an approach is that unlike in 2D, the Voronoi vertices (circumcentres of the tetrahedra) in 3D do *not* converge to the medial axis as the sampling density approaches infinity [Amenta et al. 2001b]. Therefore, regardless of sampling density, there are many tetrahedra that are not even close to the medial axis that are being used for enforcing topological constraints. This can often hinder the regularization process.

A number of recent approaches are designed to guarantee convergence. For example, Dey and Zhao [2002] have a method for computing approximations that converge to the medial axis by applying certain filter conditions to the Voronoi diagram. These filters are also used to eliminate noisy components. The strongest advantage of their technique is that the filter parameters are independent of scale and density. Another approach is by Foskey *et al.* [2003], who present an efficient method for computing a simplified medial axis using a spatial subdivision scheme and graphics hardware. The primary advantage of their approach is speed. The greatest drawback of most of these techniques is that topology is ignored, and in many cases a disintegration effect is seen in which holes appear in the axis (*e.g.*, Figure 1). Such errors are particularly prevalent when simplification is being aggressively applied. In addition to being visually distracting, these artifacts make subsequent use of the axis more difficult.

Fandisk Medial Axis (Dey and Zhao) | Fandisk Medial Axis (Our Algorithm)

Figure 1: Example of the holes that are prevented by our algorithm

Another approach that guarantees convergence to the medial axis uses a relatively small subset of the Voronoi vertices called the *poles*. The poles of a boundary sample s are defined as the farthest Voronoi vertex from s in the interior of the object and the farthest Voronoi vertex from s in the exterior of the object. Thus, each sample normally has an *inner pole* and an *outer pole*. The maximal balls centred at the poles are called *polar balls*. The inner poles have been proven to converge to the interior medial axis [Amenta et al. 2001b]. The pioneers of this approach are Amenta *et al.*, who use the power diagram of the poles (using the polar ball radii as weights) to compute an approximate medial axis, called the *power shape* [Amenta et al. 2001a]. The power shape method is robust and gives visually reasonable results. Unfortunately, for most objects the power shape is largely composed of very flat tetrahedra instead of 2D faces, and this geometry complicates tasks such as parts decomposition. In addition, their proposed method of simplification can result in an approximation that can diverge quite dramatically from the true medial axis as the level of detail decreases.

Amenta and Kolluri [2000] use the power shape to produce a more accurate axis composed only of 2D components. Our experimentation with this algorithm reveals that it produces many duplicate vertices, causing cracks in the resulting medial surface. This again makes the axis difficult to work with.

Our algorithm for computing the medial axis builds on the work by Amenta and Kolluri. We make improvements to eliminate the degeneracies and add a simplification method that preserves topology. We choose to work with Amenta and Kolluri's method because of its convergence guarantees and because of the existence of the power crust algorithm, which can take a set of filtered polar balls and reconstruct a polygonal surface. Figure 2 shows how our algorithm complements the power crust pipeline.

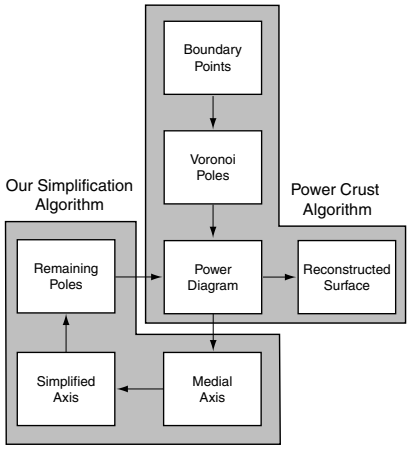


Figure 2: Processing pipeline for medial axis simplification and surface reconstruction

2.1 Summary of Amenta and Kolluri's Algorithm

As mentioned above, our method for computing the medial axis builds on the work by Amenta and Kolluri. We briefly summarize their algorithm here.

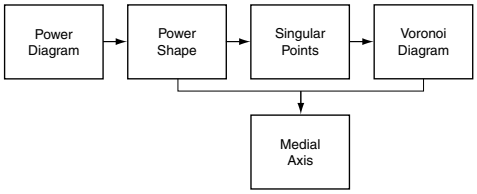


Figure 3: Amenta and Kolluri's algorithm for computing the medial axis from the power shape

As shown in Figure 2, Amenta and Kolluri's algorithm computes the medial axis from the power diagram of the Voronoi poles. This computation can be broken down into several steps, as shown in Figure 3:

1. The power shape of the object, a subcomplex of the power diagram, is computed by keeping only those simplices whose vertices are all inner poles.
2. For each regular triangle in the power shape, a *singular point* is computed. The three polar balls centred at the vertices of the triangle form two intersection points. One of these points lies on the surface of the union of polar balls, and the other lies in the interior. The surface point is the singular point.
3. A Voronoi diagram of the singular points is computed. The subcomplex of the Voronoi diagram that intersects the power shape is computed as the medial axis.

More details on the algorithms by Amenta *et al.*, including the theoretical derivation, sampling assumptions, and convergence guarantees can be found in [Amenta et al. 2001b].

3 Medial Axis Computation

For use in geometric processing, the most significant limitation of Amenta and Kolluri's algorithm is that it produces many duplicate vertices in the medial axis. These vertices cause double edges that show up in the form of cracks in the medial surface. By solving this problem we can produce a medial axis that has much cleaner geometry for further processing. We take a combinatorial approach because removing duplicate vertices numerically is computationally expensive and subject to errors in precision.

From our analysis, there are two primary causes of duplicate vertices:

1. Many duplicate singular points are generated. This results in duplicate vertices in the medial axis because the vertices of the axis are Voronoi vertices computed from the singular points.
2. Many of the singular points are cospherical. The Voronoi vertices are the circumcentres of the dual Delaunay tetrahedralization of the singular points, so when more than four points are cospherical, duplicate circumcentres are produced, resulting in multiple identical medial vertices.

To address the first problem, we note that Amenta and Kolluri's algorithm computes a singular point for *every* regular triangle in the power shape. However, we observe that two or more neighbouring triangles can often produce identical singular points. We use Figure 4 to illustrate the 2D analogy of a situation that happens frequently in real data sets. In this case, A–E are five boundary points of the object, the dotted lines represent the Delaunay triangulation of the boundary points, P1–P3 are the polar balls computed from the triangles, and S1, S2 are two simplices of the power shape (edges in 2D, triangles in 3D). The construction leads to S1 and S2 producing the same singular point, located at boundary point B, because P2 intersects both P1 and P3 at that point. We can efficiently identify and remove duplicate singular points produced in this manner by checking which polar balls have corresponding tetrahedra that share the same boundary point.

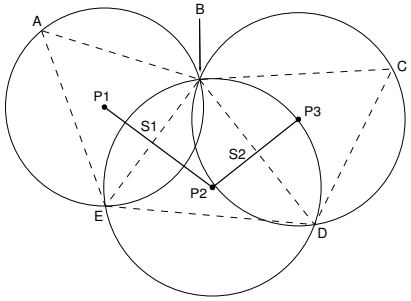


Figure 4: Simplices S1 and S2 produce two identical singular points, co-located at B

The second problem, cospherical singular points, can be attributed to the simple fact that the points are computed by intersecting balls. So it should not be surprising to find cases in which more than four singular points lie on the surface of a polar ball. We can quickly identify which singular points are cospherical by keeping track of which polar balls are intersected to form which singular points. We can thus find and eliminate duplicate medial vertices very efficiently.

4 Medial Axis Simplification

As with most 3D medial axis regularization methods that preserve topology, we utilize a peeling approach in which the outer layers of components are removed over a number of iterations. At the beginning of each iteration, we decompose the medial axis into parts. We then assign a significance value to each part that is a candidate for removal. An ordered pruning process then removes all parts that have a significance value in a given range and can be deleted while maintaining topological consistency. The number of iterations performed depends on the complexity of the shape and the degree of simplification required.

4.1 Parts Decomposition of the Medial Axis

After generating the MAT, we decompose the axis into parts before further processing. We begin by triangulating all faces to make implementation easier. We then form parts by grouping triangles. Our decomposition scheme takes advantage of the fact that the 3D medial axis can be viewed as a composition of 2D surfaces (sometimes called *boundary sheets*). The result of the decomposition is a set of connected 2-manifold parts embedded in 3D space. As shown in Section 4.3, this is very useful for simplifying the topological representation used during pruning.

The parts creation process begins at the boundaries of the medial axis and works inwardly on a level-by-level basis. The first level parts are at the boundaries of the axis, the second level parts are inward neighbours of the first level, and so on.

Each first level part starts with a randomly chosen seed triangle that has an edge with no neighbours (*i.e.*, it is at a boundary of the axis) and does not yet belong to any part. The part begins to grow by gathering neighbouring triangles of the seed triangle. For each of the triangle's edges, if that edge is shared with only one other triangle, then the part grows into that neighbour. This growth

process proceeds recursively until no more triangles can be added to this part. The resulting part is a 2D surface. Another boundary triangle is then selected to begin another first level part.

After all parts of level 1 have been formed (*i.e.*, there are no more unused triangles at the axis boundaries), parts of level 2 are computed. For a part of level i, where $i > 1$, a randomly chosen triangle that neighbours a part of level $i - 1$ is used to begin the part. The growth process is the same as for level 1. In this manner a number of levels of parts are created. Figure 5 illustrates an example using a simple medial axis, in this case computed from the boundary points of a rectangular box. The numbers in the figure indicate the levels of the respective parts.

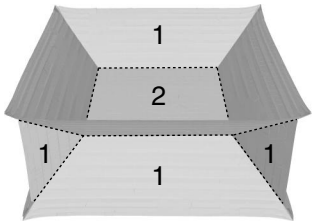


Figure 5: Medial axis showing level 1 and level 2 parts (the dotted lines show the boundaries between parts)

The user can limit the number of levels created depending on the application. During each pruning iteration, only parts of level 1 are removed, and parts of level > 1 are only used for enforcing topological constraints. Therefore, if topological preservation is considered unimportant for the current application, only the first level parts need to be created.

4.2 Medial Axis Pruning

The general simplification strategy is to remove one layer of the outermost (*i.e.*, level 1) parts at a time. This allows us to check for and prevent undesirable changes in topology. In order to determine which parts are to be removed, we apply two significance measures to evaluate the importance of any given part. In each iteration, the user selects a significance measure and a threshold value. Every level 1 part that falls below the threshold and satisfies all topological constraints is removed.

We have designed our significance measures to be efficient, versatile and intuitive to the average user. The first measure is simply the number of triangles in the part. We use triangle count because because it allows us to filter out a large number of insignificant parts with very little computation. The other measure that we use is the volume of the feature of the object that would be removed as a result of pruning the part. We use the volume because it is an intuitive and visually meaningful property of each component. Useful thresholds can be easily determined by examining the size of features that the user wants to remove. To make this measure independent of scale, we divide the volume of each part by the total volume of the object to give a relative value.

To estimate the volume for each part, we note that the set of Delaunay tetrahedra computed from the object's boundary points makes a visually reasonable approximation of the object's interior. Since these tetrahedra are available from earlier Voronoi computations, it would be efficient to reuse them for volume estimation. To do so we need to associate a appropriate set of tetrahedra with each part. We make use of the following two observations:

Observation 1 ([Amenta and Kolluri 2000]) *Every vertex of the power shape (*i.e.*, an inner pole) must lie on an edge or a vertex of the medial axis.*

Observation 2 *Since the medial axis is the intersection between the power shape and the Voronoi diagram of the singular points, every part of the axis must contain a subset of the inner poles.*

Thus, each part of the axis has a number of inner poles lying on its edges and/or vertices. We can combinatorially determine the set of poles that lie in a given part by keeping track of which polar balls intersect to produce which singular points, and tracking which singular points are subsequently used to produce which medial vertices and edges. Having the set of inner poles immediately gives us a set of tetrahedra, because each pole is computed as a circumcentre from the Delaunay tetrahedralization. We use the sum of the volumes of the tetrahedra associated with a part to estimate the volume of the object feature corresponding to that part. Figure 6 shows a feature of an object (the top of the Tweety model in Figure 11a) and the group of tetrahedra used to estimate its volume.

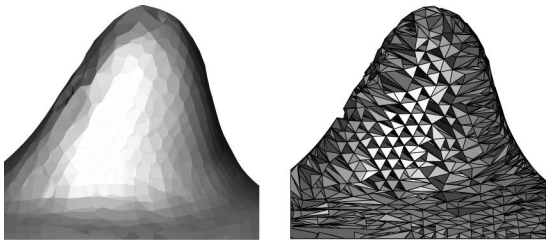


Figure 6: Feature of an object and the tetrahedra used to estimate its volume

Although the two significance measures can theoretically be used in any order or combination, we have found the process generally more effective and easier to control if the triangle count measure is used first to remove the very small outer layer parts, followed by using the volume measure to remove the more significant parts underneath. The reasoning is that for most data sets, practically all of the very small parts in the first few layers are discretization or noise artifacts. After these are removed, the volume measure can be applied iteratively in a manner appropriate for the data and application. Indeed, as discussed in Section 6, our experiments have shown this to be a useful strategy.

4.3 Topology Constraints and Pruning Order

Our algorithm is able to preserve the topology of the medial axis during pruning. The topological properties that concern us are the number of connected components and the number of loops. Each loop in the axis represents a tunnel going through the object. Given a set of connected parts representing the medial axis, we need to determine whether any given level 1 part can be safely removed without disconnecting the remaining parts and without changing the number of existing loops.

We deal first with the loops. The part removal method described in the previous section guarantees that new loops are never created, because only parts at the borders of the axis are candidates for pruning. In order to save existing loops, we first determine which parts join together to form loops in the axis. We can then preserve these parts during pruning. In 3D, finding loops is not just a simple matter of detecting cycles, because there are many cycles that do not form loops. For example, the three connected parts on the left in Figure 7 form a cycle (A→B→C→A) but do not make a loop. Removal of any of the three parts would not change the topology of the object. In contrast, the three parts on the right in Figure 7 do make a loop.

To detect loops efficiently, we take advantage of the simplicity of our parts-based representation to build a topology graph that concisely captures the connectivity information between the parts. In

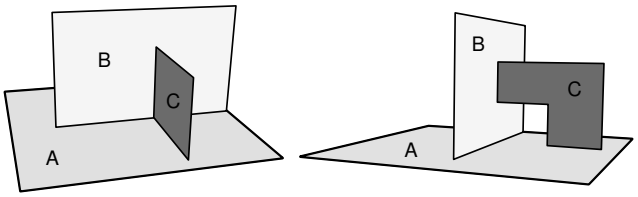


Figure 7: The three parts on the left do not form a loop, while the three parts on the right do.

this graph, each node represents a "point" of contact between two neighbouring parts. This "point", called a *contact curve*, takes the form of an unbroken polyline shared by the touching parts. The edges of the topology graph represent the parts themselves. In order to eliminate cycles that are not loops, two nodes are collapsed into one if the corresponding contact curves are connected. For example, Figure 8 shows the graphs representing the topology of the two sets of parts in Figure 7. In the first graph, there is only one

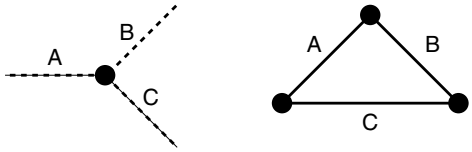
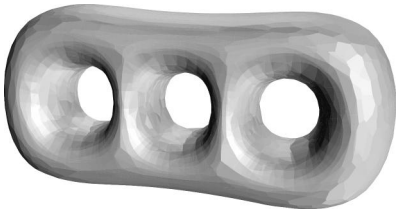
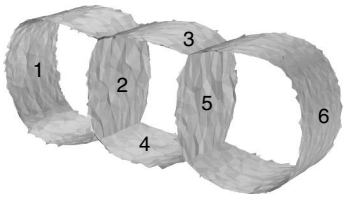


Figure 8: Graphs representing the topology of the parts in Figure 7

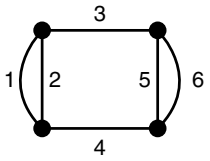
node because the contact curves between parts A and B, B and C, and A and C are all connected. The edges of this graph do not connect to any other nodes. The second graph has a loop in the configuration because the contact curves do not touch each other. Figure 9 shows a less trivial example computed by our algorithm. The topology graph has four nodes interconnected by six parts.



(a)



(b)



(c)

Figure 9: (a) Object (b) Medial axis (c) Topology graph

With the loops preserved, maintaining topology then becomes a matter of ensuring that the number of connected components stays the same. We satisfy this constraint by checking that the removal of a part does not disconnect any of the neighbouring parts.

The lack of a natural processing order and the fact that there are no theoretical results known about the effect of the deletion sequence on the skeleton mean that we need to impose a pruning order. Our algorithm takes the simple approach of sorting the parts in order from lowest to highest value, using the significance measures defined in the previous section. The reasoning is that in general, we want to remove the less important parts first whenever possible so that they do not prevent the more significant parts that typically represent more salient features from being pruned.

5 Surface Reconstruction

After pruning, we use the power crust algorithm [Amenta et al. 2001a] to reconstruct a surface of the simplified object. The algorithm works by computing a piecewise linear boundary between the inner and outer polar balls. Before simplification, each boundary point sample has an inner and outer pole. Afterwards, only a subset of the inner poles remain. We take our simplified axis and use the method described in Section 4.2 to determine which inner poles lie on the remaining parts. For most data sets, we only need to remove the inner poles. However, as mentioned in [Amenta et al. 2001a], for objects with sharp corners, more accurate reconstructions can be had by removing the corresponding outer poles as well. The polar balls associated with the remaining poles are used to compute the surface.

6 Results

This section describes the results of testing our algorithm with a number of data sets. In all of the examples shown, it is easily noticed that our approach can greatly reduce the complexity of the axis without creating holes or breaks in the remaining components. Table 1 lists the examples presented, along with the processing times for computing and simplifying their medial axis. A Pentium 4 processor running at 2.0 Ghz is used. Our implementation makes extensive use of the CGAL and LEDA libraries.

Model	Points	Axis Generation (sec.)	Pruning (sec.)
Tweety	48668	186	205
Max Planck	25044	57	92
Hip bone	70688	348	354
Bunny	34835	103	136

Table 1: Processing times for our examples

The typically pruning scheme that we use is to first apply the triangle count significance measure for one to three iterations with a very low value (≤ 5 triangles) and without topology checks. This usually removes many very small and visually unimportant parts with little computation. Then we apply the volume measure with topology checks for one or more iterations as required to achieve the desired level of detail. Figure 10 shows a plot of the number of parts of the medial axis of the Tweety model (Figure 11) as it undergoes several iterations of pruning. The first three iterations use the triangle count measure, and the fourth iteration uses the volume measure. A dramatic drop in the number of parts removed while using the triangle count measure, such as that observed in the third iteration, is typically a good indication to switch to the more discerning volume measure.

The Tweety model is shown in Figure 11a. Its medial axis is in Figure 11b. This example is provided to illustrate how our algorithm can be used to remove small noise-type artifacts. Figure 11d shows the medial axis after five iterations of pruning ($3 \times t_t = 5$,

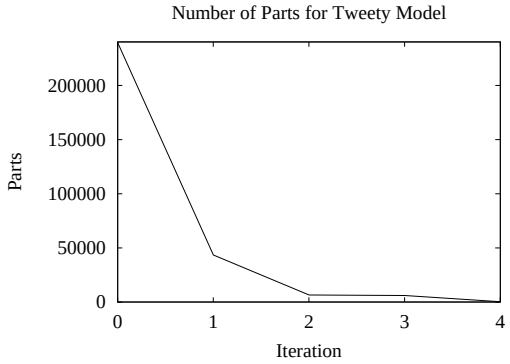


Figure 10: Number of axis parts at the end of each pruning iteration for the Tweety model

$2\times t_v = 1.0\%$, where t_t is the triangle count threshold, and t_v is the volume threshold). The simplified axis is clearly much cleaner, and potentially much more useful for applications such as shape matching. Figures 11c and 11e show a closeup of the original model and the simplified model, respectively. The simplified model is clean of the small bumps seen on the back and leg of the original model.

The Max Planck model, shown in Figure 12a, is given as an example of how an object can be edited by manually specifying parts of the medial axis to be removed. After four pruning iterations ($2 \times t_t = 5$, $2 \times t_v = 1.0\%$), we have a relatively clean axis to work with (Figure 12c). We can manually select the left ear and its stump by simply specifying a single triangle in each part. The medial axis without the left ear is shown in Figure 12d and the resulting surface is shown in Figure 12e. The ear is removed without appreciable distortion to the surrounding area.

The hip bone model (Figure 12f) is an example of where topology preservation becomes very useful. Figure 12g shows the original medial axis, which is full of small details. Figure 12h shows the medial axis after six pruning iterations ($3\times t_t = 5$, $3\times t_v = 5.0\%$). The algorithm greatly reduces the amount of small details, but is able to preserve the narrow arch in the axis. Many other existing pruning methods would break or disintegrate this loop, particular where it is thin. The reconstructed simplified object with the narrow loop clearly intact is shown in Figure 12i.

We use the Bunny model (title image) to demonstrate how our algorithm can be used to automatically remove large features for a more radical simplification. After four iterations ($2 \times t_t = 5$, $1 \times t_v = 1.0\%$, $1 \times t_v = 10.0\%$), the resulting object has lost all of its small details, such as the eyes, as well as most of the large features such as the ears and the tail. The reconstructed surface is a considerably simpler version of the original. However, this example also shows a limitation of the approach: our simple decomposition method does not always completely divide the parts in the way a human or a more complex medial axis analysis technique (*e.g.*, *shocks* approach [Giblin and Kimia 2002]) would. In this case, the feet of the bunny are not separated from the rest of the body, and cannot be thresholded out. Consequently, the feet are only mildly simplified compared to the rest of the object. In such cases, more sophisticated *mesh segmentation* techniques (*e.g.*, [Zhang et al. 2002]) should prove useful for further decomposition of the axis.

7 Future Work

There are a number of avenues we would like to pursue in order to further enhance our current algorithm. First, so far we have assumed that the input data is adequately sampled. While this is usually the case for data sources such as laser scanners, other sources of data may result in undersampled point sets. More research in the detection of undersampling, based on previous work (*e.g.*, [Amenta and Bern 1999; Dey and Giesen 2001; Boissonnat and Flötotto 2002]), should be done to improve the robustness of the algorithm.

As mentioned in the Results section, even though our method of simply using the number of triangle neighbours to divide the axis into parts works well in general, there are cases that warrant a more advanced parts decomposition technique. We would also like to find a more automatic method for estimating the parameter values. Even though simply estimating the parameter values by visual inspection gives reasonable results, an automatic or semiautomatic method based on an analysis of data characteristics is preferable. In addition, we would like to further investigate the effect of processing order (both during decomposition and pruning), other significance measures, and different types of data, especially those with complex topology. Lastly, we would like to improve the efficiency and speed of our method to facilitate the processing of much larger data sets.

8 Summary and Conclusion

This paper presents a new algorithm for simplifying 3D shapes by pruning their medial axes. The approach is particularly novel in that the axis is decomposed into parts before simplification. This feature-based approach is to shown to have a number of advantages over existing techniques. Our results demonstrate that the algorithm is able to greatly reduce the amount of in detail in the medial axis without negatively affecting the remaining components. We use the power crust method to reconstruct a polygonal surface from the filtered MAT.

References

AMENTA, N., AND BERN, M. 1999. Surface reconstruction by voronoi filtering. *Discrete & Computational Geometry 22*, 4, 481–504.

AMENTA, N., AND KOLLURI, R. 2000. The medial axis of a union of balls. In *Proceedings of the Canadian Conference on Computational Geometry*, 111–114.

AMENTA, N., BERN, M., AND KAMVYSSELIS, M. 1998. A new Voronoi-based surface reconstruction algorithm. In *Proceedings of ACM SIGGRAPH*, 415–421.

AMENTA, N., CHOI, S., AND KOLLURI, R. 2001. The power crust. In *Proceedings of the ACM Symposium on Solid Modeling and Applications*, 249–260.

AMENTA, N., CHOI, S., AND KOLLURI, R. 2001. The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications 19*, 2-3, 127–153.

ATTALI, D., AND MONTANVERT, A. 1997. Computing and simplifying 2D and 3D continuous skeletons. *Computer Vision and Image Understanding 67*, 3, 161–273.

ATTALI, D., SANNITI DI BAJA, G., AND THIEL, E. 1995. Pruning discrete and semicontinuous skeletons. In *Lecture Notes in Computer Science, Image Analysis and Processing*, C. De Floriani, C. Braccini, and G. Vernazza, Eds., vol. 974. Springer-Verlag, 488–493.

BLOOMENTHAL, J. 2002. Medial-based vertex deformation. In *Proceedings of the ACM SIGGRAPH Symposium on Computer Animation*.

BOISSONNAT, J.-D., AND FLÖTOTTO, J. 2002. A local coordinate system on a surface. In *Proceedings of the ACM Symposium on Solid Modeling and Applications*, 116–126.

DEY, T. K., AND GIESEN, J. 2001. Detecting undersampling in surface reconstruction. In *Proceedings of the 17th Annual Symposium on Computation Geometry*, 257–263.

DEY, T. K., AND ZHAO, W. 2002. Approximate medial axis as a voronoi subcomplex. In *Proceedings of the ACM Symposium on Solid Modeling and Applications*.

FOSKEY, M., LIN, M., AND MANOCHA, D. 2003. Efficient computation of a simplified medial axis. In *Proceedings of the ACM Symposium on Solid Modeling and Applications*.

GIBLIN, P., AND KIMIA, B. 2002. Transitions of the 3D medial axis under a one-parameter family of deformations. *Lecture Notes in Computer Science (Proceedings of the European Conference on Computer Vision) 2351*, 719–734.

HECKBERT, P., AND GARLAND, M. 1997. Survey of polygonal surface simplification algorithms. Tech. rep., Department of Computer Science, Carnegie Mellon University.

HUBBARD, P. 1996. Approximating polyhedra with spheres for time-critical collision detection. *ACM Transactions on Graphics 15*, 3, 179–210.

LUEBKE, D. 2001. A developer's survey of polygonal simplification algorithms. *IEEE Computer Graphics and Applications 21*, 3, 24–35.

NÄF, M., SZÉKELY, G., KIKINIS, R., SHENTON, M., AND KÜBLER, O. 1997. 3D Voronoi skeletons and their usage for the characterization and recognition of 3D organ shape. *Computer Vision and Image Understanding 66*, 2 (May), 147–161.

OGNIEWICZ, R. 1994. Skeleton-space: A multiscale shape description combining region and boundary information. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 746–751.

OGNIEWICZ, R. 1995. Automatic medial axis pruning by mapping characteristics of boundaries evolving under the euclidean geometric heat flow onto voronoi skeletons. Tech. Rep. 95-4, Harvard Robotics Laboratory.

PAIK, D., BEAULIEU, C., JEFFREY, R., RUBIN, G., AND NAPEL, S. 1998. Automated flight path planning for virtual endoscopy. *Medical Physics 25*, 5, 629–37.

PIZER, S., FRITSCH, D., YUSHKEVICH, P., JOHNSON, V., AND CHANEY, E. 1996. Segmentation, registration, and measurement of shape variation via image object shape. *IEEE Transactions on Medical Imaging 18*, 10, 851–865.

SHAKED, D., AND BRUCKSTEIN, A. 1998. Pruning medial axes. *Computer Vision and Image Understanding 69*, 2 (Feb.), 156–169.

SIDDIQI, K., SHOKOUFANDEH, A., DICKINSON, S., AND ZUCKER, S. 1999. Shock graphs and shape matching. *International Journal of Computer Vision 35*, 1, 13–32.

TAM, R., AND HEIDRICH, W. 2002. Feature-preserving medial axis noise removal. *Lecture Notes in Computer Science (Proceedings of the European Conference on Computer Vision) 2351*, 672–686.

ZHANG, Y., PAIK, J., KOSCHAN, A., ABIDI, M., AND D., G. 2002. A simple and efficient algorithm for part decomposition of 3D triangulated models based on curvature analysis. In *Proceedings of the International Conference on Image Processing (ICIP)*, vol. 3, 273–276.

9 Acknowledgments

We would like to thank Nina Amenta, Sunghee Choi and Ravi Kolluri for providing their power crust code and a template for our medial axis code. We would also like to thank Tamal Dey for providing the tight cocone code for comparison. Finally, we would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Advanced Systems Institute of British Columbia (ASI) for their funding support.

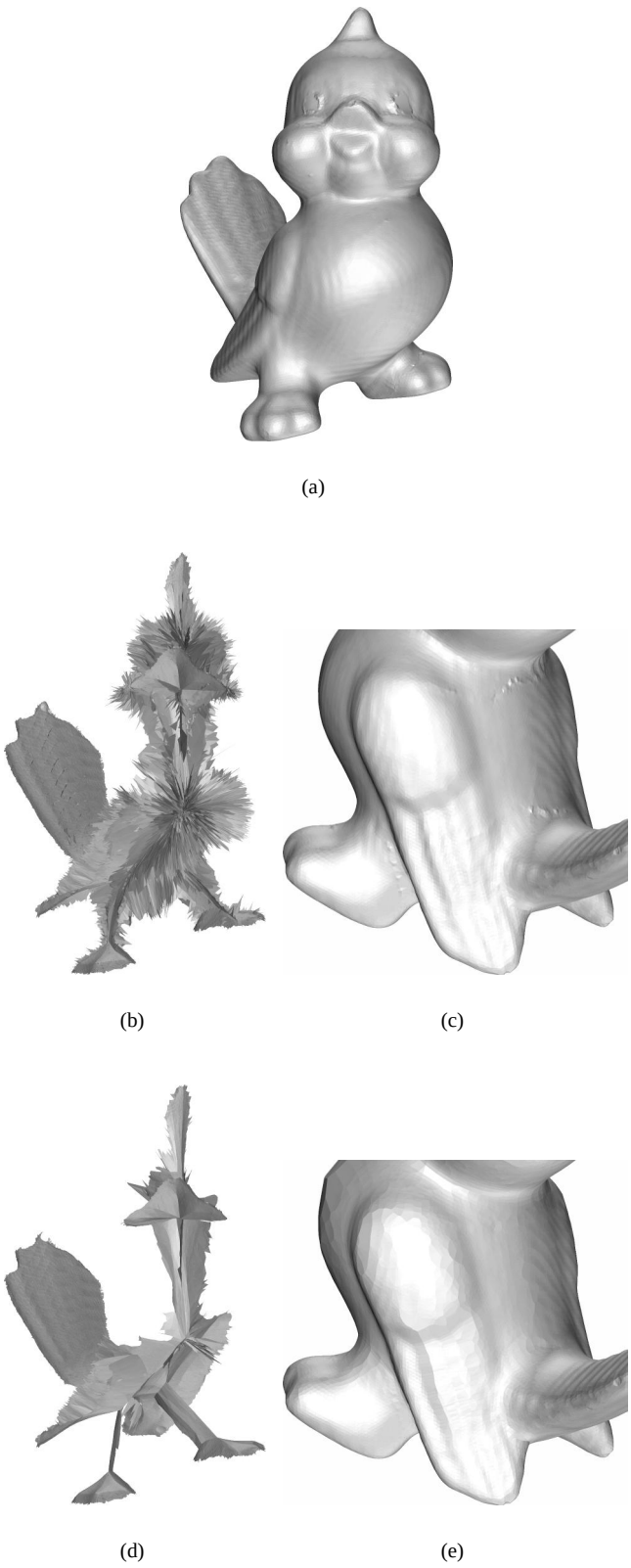
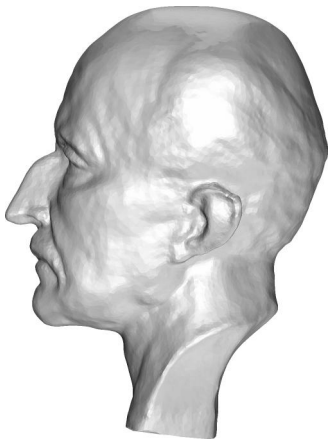
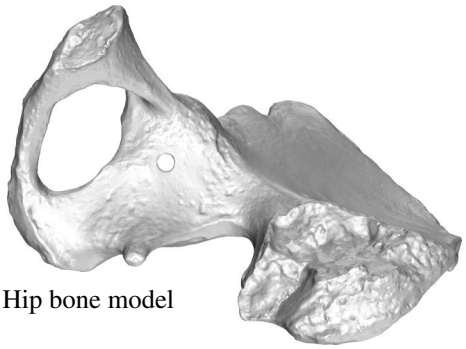


Figure 11: (a) Tweety model (b) Medial axis (c) Back of Tweety model, with obvious noise artifacts on the back and leg (d) Simplified medial axis (e) Back of simplified model, with noise clearly reduced



(a) Max Planck model

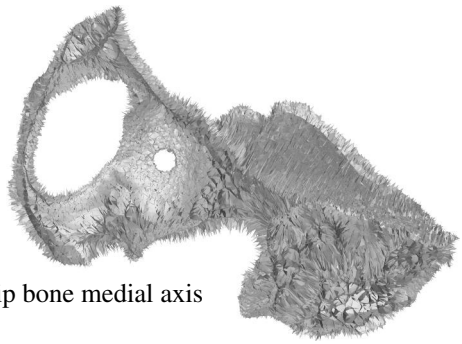
(b) Medial axis of Planck model



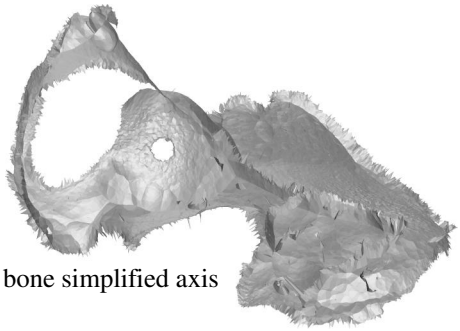
(f) Hip bone model

(c) Simplified MA of Planck model

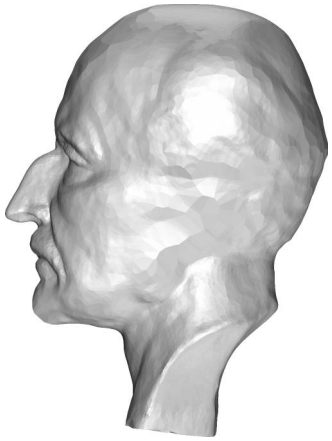
(d) Simplified MA of Planck model (left ear removed)



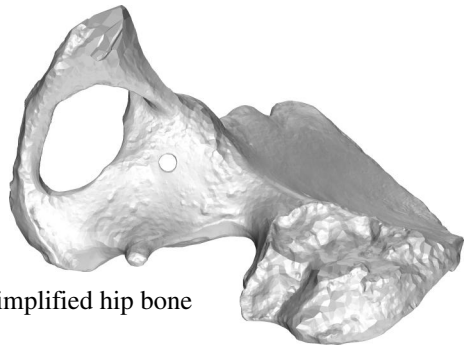
(g) Hip bone medial axis



(h) Hip bone simplified axis



(e) Simplified Planck model (left ear removed)



(i) Simplified hip bone

Figure 12: Results of applying our simplification algorithm to the Max Planck and hip bone models