Computing Medial Axes of Generic 3D Regions Bounded by B-spline Surfaces

Suraj Musuvathy [a] Elaine Cohen [b] James Damon [c]

[a] School of Computing, University of Utah, 50 S. Central Campus Dr., Rm. 3190, Salt Lake City, Utah, USA 84112
[b] School of Computing, University of Utah, 50 S. Central Campus Dr., Rm. 3190, Salt Lake City, Utah, USA 84112
[c] Department of Mathematics, University of North Carolina at Chapel Hill, CB 3250 Phillips Hall, Chapel Hill, NC, USA 27599-3250

Abstract

A new approach is presented for computing the interior medial axes of generic regions in $\mathbb{R}^3$ bounded by $C^{(4)}$-smooth parametric B-spline surfaces. The generic structure of the 3D medial axis is a set of smooth surfaces along with a singular set consisting of edge curves, branch curves, fin points and six junction points. In this work, the medial axis singular set is first computed directly from the B-spline representation using a collection of robust higher order techniques. Medial axis surfaces are computed as a time trace of the evolving self-intersection set of the boundary under the the eikonal (grassfire) flow, where the bounding surfaces are dynamically offset along the inward normal direction. The eikonal flow results in special transition points that create, modify or annihilate evolving curve fronts of the (self-) intersection set. The transition points are explicitly identified using the B-spline representation. Evolution of the (self-) intersection set is computed by adapting a method for tracking intersection curves of two different surfaces deforming over generalized offset vector fields. The proposed algorithm accurately computes connected surfaces of the medial axis as well its singular set. This presents a complete solution along with accurate topological structure.

1 Introduction

The medial axis [7] is a fundamental tool for the analysis of three dimensional objects in a variety of geometric processing applications such as volumetric mesh generation [39,35], feature extraction [26,28], tool-path creation in computer-aided manufacturing [25] and shape search and retrieval [43]. The medial axis has also been used in other scientific applications including segmentation of medical images [34], statistical shape analysis of populations of objects [29], and motion planning for robotics [21,18]. A survey of medial axis applications is presented in [42].

There has been significant interest in automatically computing the medial axis of objects. The approaches presented in [37,36] compute partial solutions for a restricted set of objects bounded by B-spline surfaces. Most existing approaches require discrete approximations of smooth surfaces such as polygonal meshes or point sampled geometry. Accurate techniques that compute the medial axis and its topology from polygonal boundary representations have been demonstrated for low polygon count models [11,40]. Computationally fast techniques for point sampled geometry [14,2] compute approximations as a set of polygons on the medial axis without topological structure. Considerable human interaction is required in order to infer this information. Discrete approximations of smooth surfaces introduce artifacts that are not part of the medial axis of regions bounded by smooth surfaces and considerable effort is required to remove them.

This paper presents a new approach for computing the complete topologically correct interior medial axis of a three dimensional region directly from parametric B-spline representations of its boundary surfaces. Figure 1 shows a B-spline surface enclosing a region in $\mathbb{R}^3$ and its corresponding medial axis. The medial axis consists of surfaces (shown in orange, brown, bluish violet) bounded by edge curves (thick blue), and branch curves (thick yellow) where surfaces meet. The medial axis also contains fin points (purple spheres) where edge curves and branch curves meet, and a six junction point (green sphere) where six surfaces (and four branch curves) meet at a point. Precise mathematical definitions of each entity type are presented in Section 2.

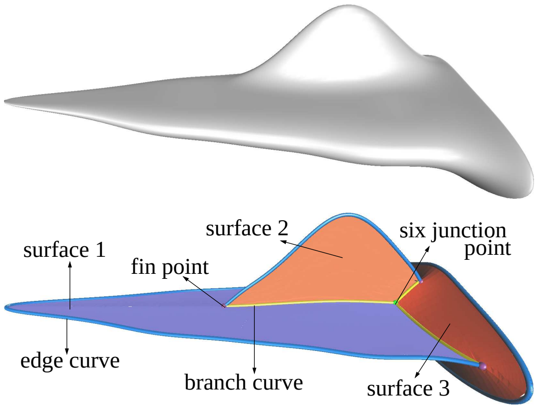

Fig. 1. Medial axis of a 3D region bounded by a B-spline surface.

The proposed approach computes all entities of the medial axis with arbitrary user specified accuracy along with correct topology. Parameter values of boundary surface points corresponding to every medial axis point and the respective distance is also computed, which gives the medial axis transform. The main contributions of this paper are:

- A new algorithm for computing medial axis surfaces.
- A new algorithm for tracing branch curves.
- Robust technique for computing edge curves.
- Robust techniques for computing fin and six junction points.

Email addresses: srm@cs.utah.edu (Suraj Musuvathy), cohen@cs.utah.edu (Elaine Cohen), jndamon@math.unc.edu (James Damon).

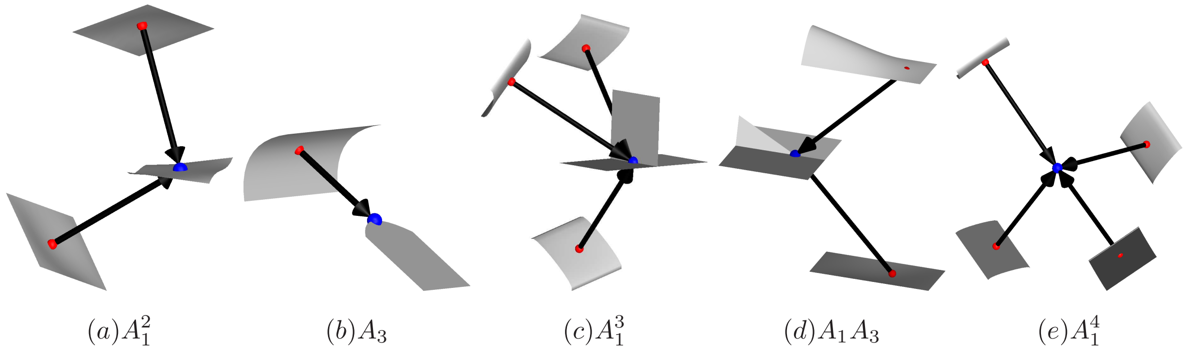

Fig. 2. Medial axis point types. Surface S shown in grey, surface points in red, medial axis points in blue. In (a)-(d), medial axis surfaces are also shown in grey. Arrows point in the corresponding surface normal direction.

The B-spline representation enables computation of the medial axis as a time trace of the evolving (self-) intersection set of the boundary surface under the eikonal flow (also called grassfire flow [42]). At each instant of the eikonal flow, every point on the surface is offset by a fixed distance along the corresponding inward surface normal direction. The eikonal flow method has been used to compute the medial axis of regions specified within volumetric scalar grids [41]. That method classifies the points of the discrete grid as whether or not they belong to the medial axis using computed properties of the eikonal flow at the grid points. The method proposed in this paper dynamically emulates the (self-) intersections of the eikonal flow using higher order methods to accurately compute the medial axis of regions bounded by B-spline surfaces.

The paper is organized as follows. Section 2 presents the definition and properties of the 3D medial axis. Section 3 presents related work. An overview of the proposed algorithm is presented in Section 4 with details in Sections 5, 6 and 7. Results are presented in Section 8 and the contributions of the paper are summarized in Section 9.

2 3D Medial Axis and its Local Structure

Mathematical properties of the medial axis are well documented in the existing literature. We follow the definition and notation given in [20,12] and present it here.

Definition 2.1 *The medial axis of a region in R^3 enclosed by a bounding surface S is the locus of centers of maximally inscribed spheres that are tangent to two points on S, together with the limit points of the locus that exhibit special configurations (explained below).*

A complete characterization of the local structure of the 3D medial axis for generic regions is presented in [20]. We briefly describe the properties of the 3D medial axis here. The assumption that the 3D region is generic includes "almost all" situations in the precise mathematical sense [12].

For a maximally inscribed sphere, a point of tangency on the boundary is of A_k type if the sphere has $k-th$ order contact with the boundary [20]. Only the values $k = 1$ or 3 occur in the generic situation for medial axis points. When $k = 1$, the sphere is tangent to the surface at the contact point. When $k = 3$, in addition to the tangency property, the radius of the sphere is a radius of principal curvature of the corresponding contact point on the surface and the contact point on the surface is a ridge point. The definition of a ridge point is presented in Section 6.1. A medial axis point with a maximal sphere with m points of tangency is denoted $A_{k_1}A_{k_2}\cdots A_{k_m}$, and the type for a sphere with m A_1 contact points will be abbreviated to A_1^m. For each medial axis point, our work maintains the set of parameter values corresponding to each contact point on the boundary and the distance to the boundary (radius of maximal sphere) in addition to the coordinate values of the center of the sphere.

The medial axis of a generic 3D object consists of the following surface, curve and point entities [20] (See Figure 2 for illustrations of each medial axis point types):

(a) A_1^2 *surfaces.* The locus of A_1^2 points are medial surfaces on which the maximal sphere at every point is tangent to two points on the surface.

(b) A_3 *edge curves.* The maximally inscribed sphere at an A_3 point is in contact with an elliptic ridge point on the surface (See Section 6.1). The locus of A_3 points are curve segments or loops that partially or completely bound A_1^2 surfaces.

(c) A_1^3 *junction curves.* The maximal sphere at an A_1^3 point is tangent to three points on the surface. The locus of A_1^3 points are curve segments or loops that partially or completely bound A_1^2 surfaces. Three A_1^2 surfaces meet along an A_1^3 curve.

(d) A_1A_3 *fin points.* At such a point, an A_3 curve segment meets an A_1^3 curve segment and it marks the beginning/end of each curve. The maximal sphere at an A_1A_3 point is in contact with two surface points - one with A_1 type and the other with A_3 type contact.

(e) A_1^4 *six junction points.* The maximal sphere at an A_1^4 point is tangent to four points on the surface. Such points occur when six A_1^2 surfaces meet. This can also be viewed as points where four A_1^3 curves meet.

3 Related Work

We briefly review existing techniques for computing the 3D medial axis. The approaches can be broadly classified into Voronoi, distance field, eikonal and tracing based methods. First, we present techniques used for piecewise smooth surface representations. Since many approaches suggest discretizing smooth surfaces, we summarize existing techniques used for discrete surface representations including point clouds, polygonal meshes and volumetric images (scalar grid data). This is by no means an exhaustive review, but presents representative works in each area. Extensive surveys are presented in [42,6].

Piecewise Smooth Representations. The tracing approach of [37] accurately computes partial medial axes of objects bounded by freeform surfaces. The method assumes that the object has at least one convex vertex (corner point). A_1^3 curves (termed seams in their paper) are numerically traced from convex vertices. A_1^4 points are detected during seam tracing using distance to surface checks, at which three new A_1^3 curves are spawned. Seam tracing ends either at a convex vertex or at an A_3 point which is detected using curvature and distance checks at seam points. The paper does not present results on computing A_1^2 surfaces. Since the method assumes that there is a convex edge on the surface that is a part of a bounding loop for an A_1^2 surface, it is not suitable for medial axes with A_3 boundary curves that may end at A_1A_3 points such as models with fins.

Bisectors of pairs of freeform surfaces are considered as building blocks of the medial axis. Accurate techniques for computing bisector surfaces of rational parametric surfaces are presented in [16]. However, there is no technique in the existing literature to identify all medial entities (such as fin points, six junction points) and their topology from the bisector surfaces. Related methods include bisectors of CSG objects [27] and Voronoi diagrams of collections of planes, sphere and cylinders [24].

A method for computing the 3D medial axis of extruded and revolved objects bounded by freeform surfaces is presented in [36]. The method computes the 2D medial axis of a planar profile face which is then transformed (extruded or revolved) to obtain a 3D medial axis.

Point Clouds. Algorithms for computing medial axes from point sampled surfaces typically begin with computing Voronoi diagrams of the point sets and then identify medial skeletons as subsets of the Voronoi graphs [1,2,13,14,9]. The shock scaffold method [31] presents a combinatorial algorithm to identify boundary points corresponding to medial axis points.

Polygonal Meshes. Foskey et al. [19] analyze the gradient of the distance field to the surface to identify medial skeletons. A polygonal approximation that has the same homotopy type as the medial skeleton is computed as a subset of the Voronoi diagram in [44]. Methods based on tracing seam curves and computing junctions of seam curves are presented in [11,40]. An offset based approach for computing approximations of the 3D medial axis of polyhedra is presented in [22].

Volumetric Images. A variety of techniques based on thinning of voxel data grids in an erosion like process starting from the object surface have been presented as surveyed in [30,45]. Siddiqi et al. [41] simulate eikonal flow on discrete grids using partial differential equations to detect singularities of the flow based on the average outward flux of the flow. Medial axis voxels are detected by combining the flux measurement with a homotopy preserving thinning process. Methods that rely on the computation of height ridges of distance fields are presented in [8,3].

Although mathematical properties of the medial axis are well documented in the existing literature, to date state of the art techniques have been able to compute pieces of medial representations only for discrete approximations of objects. In addition, approximate or simplified solutions using discrete techniques and partial solutions using higher order methods have been computed due to the complexity in structure and inherent nonlinearity of the medial axis [38]. This paper proposes a new higher order method that, in conjunction with results from singularity theory, automatically computes medial axes of three dimensional objects accurately along with correct structural information and does not generate non-medial artifacts.

4 Algorithm Overview

Let $\mathcal{B}$ be the boundary of a region in $\mathbb{R}^3$ represented by a closed tensor product parametric B-spline surface $S(u,v) \in C^{(4)}$. The surface normal, $n(u,v) = \frac{S_u \times S_v}{\|S_u \times S_v\|}$ (assumed oriented inward for a closed surface, with $\| S_u \times S_v \| \neq 0$) where, subscripts indicate the partial derivatives with respect to the corresponding parameter variable. The offset surface resulting from the eikonal flow at a time t is given by $\sigma(u,v,t) = S(u,v) + t\ n(u,v), t \geq 0$. The variable t is exactly the offset distance and is also referred to as *time* to emphasize the dynamic aspect of the algorithm. As the surface evolves under the eikonal flow in the increasing t direction, different regions start intersecting with each other creating (self-) intersection curves that grow and interact with each other until they collapse to single points and die. The approach presented in this paper exactly models this behavior to compute the medial axis. The medial axis consists of only the first intersection points of evolving offsets of any two given points on the surface.

During the course of the eikonal flow, special changes to the structure of intersection curves occur at certain *transition points* (including A_1A_3 and A_1^4 points), where intersection curves are created, interact with each other to undergo intermediate transitions or get annihilated. Away from transition points, the intersection curves evolve smoothly to sweep out A_1^2 surfaces of the medial axis. An intersection curve consists of a connected set of A_1^2 points, and A_3, A_1^3, A_1A_3 or A_1^4 points at curve ends where necessary, all sharing the same offset distance.

Algorithm 1 summarizes the steps involved in computing the medial axis. The proposed approach first computes transition points as well as A_3 and A_1^3 curves using properties of the B-spline representation. A_1^2 surfaces are then computed by evolving intersection curves over time using theoretically derived evolution vector fields. Connectivity between intersection curves at consecutive time instants is maintained as they evolve. The topology of intersection curves are appropriately modified during transition events.

This computational approach is equivalent to following the level sets of the distance field of $S(u, v)$. Since the set of all local normal forms of the distance field given in [32] are considered, our approach presents the complete topological structure of the medial axis.

```
Algorithm 1 Computing medial axis
  (i) Compute transition points.
 (ii) Compute $A_3$ curves and $A_1A_3$ points.
(iii) Compute $A_1^3$ curves.
 (iv) Classify transition points as creation, intermediate or
      annihilation.
  (v) Compute medial axis surfaces.
```

5 Transition Points and Transition Events

This section presents techniques to compute all types of transition points of intersection curves for the 3D medial axis. A summary of all types of transition events is also presented. In addition to A_1A_3 and A_1^4 points, the transition points include those points on A_3 curves, A_1^3 curves and A_1^2 surfaces where the distance to the boundary attains a critical value. Transitions on A_3 and A_1^3 curves depend on the direction of increasing distance to boundary along the curves, which we term as the curve's *flow direction.*

Let $S_i(u_i, v_i)$, $i = 1, 2, 3, 4$, denote representations of different regions of the boundary surface, and let $N_i = \frac{\partial S_i}{\partial u_i} \times \frac{\partial S_i}{\partial v_i}$ denote the unnormalized normals of S_i respectively. By treating the single boundary surface as if it were different regions, we present geometric equations for computing transition points. Roots of nonlinear geometric equations are computed using adaptations of robust subdivision based techniques [17,15]. Extensions of these methods to improve efficiency of computing A_1^3 critical points and A_1^4 points are presented.

5.1 A_3 critical points

Although every point on $S(u, v)$ has two principal curvatures $\kappa_1 \geq \kappa_2$, for the purposes of medial axis computation, it is necessary to consider only the larger principal curvature κ_1 [5]. The transition points of the medial axis related to κ_1 correspond to points on $S(u, v)$ where κ_1 attains a critical value. These points are computed by solving for simultaneous roots of Equation 5.1.

$$\kappa_{1u}(u, v) = 0 \quad \kappa_{1v}(u, v) = 0 \tag{5.1}$$

Further, the sphere centered at the offset point $S(u, v) + \frac{1}{\kappa_1(u, v)} n(u, v)$ having radius $\frac{1}{\kappa_1(u, v)}$ must be maximal, by definition of the medial axis. The procedure for the maximal condition check is presented in Appendix A. The transition points of A_3 type are the centers of such maximal spheres. An A_3 critical point is either a creation point or an intermediate transition point, depending on whether the two A_3 curve segments on either side of the critical point flow outward or inward.

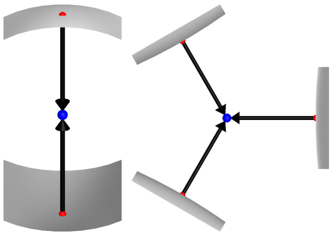

Fig. 3. Critical point of type (a) A_1^2, (b) A_1^3, shown in blue. Different regions of the surface S are shown in grey and arrows point in the corresponding surface normal directions.

5.2 A_1^2 critical points

The A_1^2 transition points correspond to A_1^2 points for which the distance to the boundary attains a critical value. An A_1^2 critical point at which the distance function has a local minimum corresponds to a creation event. When the distance function has a local saddle, the critical point corresponds to an intermediate transition event. Finally, the critical point corresponds to an annihilation event when the distance function attains a local maximum. This characterization follows the behavior of the transition points for the intersection of two surfaces under generalized offset flows [10].

$$\mathcal{D}(u_1, v_1, u_2, v_2) = \| S_1 - S_2 \|^2 = \langle S_1 - S_2, S_1 - S_2 \rangle \tag{5.2}$$

Taking the partial derivatives of $\mathcal{D}$ with respect to u_1, v_1, u_2, v_2 and solving Equation 5.3, we obtain critical points of $\mathcal{D}$.

$$\langle S_1 - S_2, \frac{\partial S_i}{\partial u_i} \rangle = 0, \quad \langle S_1 - S_2, \frac{\partial S_i}{\partial v_i} \rangle = 0, \quad i = 1, 2 \tag{5.3}$$

Since some solutions of Equation 5.3 may not correspond to maximal spheres, the maximality condition must be checked at all solutions of Equation 5.3 using the procedure in Appendix A. Trivial solutions at which $(u_1, v_1) = (u_2, v_2)$ are ignored. Equation 5.3 implies that the surface normals at $S_1(u_1, v_1)$ and $S_2(u_2, v_2)$ must point exactly in opposite directions as illustrated in Figure 3(a).

5.3 A_1A_3 points

A_1A_3 points are computed as part of the algorithm for computing A_3 curves. See Section 6.1. A_1A_3 points correspond to intermediate transition events based on whether the A_3 curve and the A_1^3 curve flow inward or outward at the A_1A_3 point. Techniques for computing the flow directions (called the shock structure) and characterization of their combinations are presented in [20].

5.4 A_1^3 critical points

The A_1^3 critical points correspond to A_1^3 points at which the distance to the boundary attains a critical value. At such points, the normals at the three surface points are coplanar as illustrated in Figure 3(b) [31]. A_1^3 critical points can be creation, annihilation or intermediate transition types based on the flow direction of the A_1^3 curve segments at the critical point. For a point $P \in \mathbb{R}^3$ to be an A_1^3 critical point, the following equations must be satisfied.

$$\langle P - S_i, \frac{\partial S_i}{\partial u_i} \rangle = 0,\ \langle P - S_i, \frac{\partial S_i}{\partial v_i} \rangle = 0,\ i = 1, 2, 3 \quad (5.4a)$$

$$\parallel P - S_1 \parallel \quad = \quad \parallel P - S_j \parallel,\ j = 2, 3 \quad (5.4b)$$

$$\langle N_1 \times N_2, N_3 \rangle = 0 \quad (5.4c)$$

Since $P = (x, y, z)$ is unknown, Equations 5.4(a)-(c) form a system of nine non-linear equations in nine unknowns. This system is reduced by adapting the simplification techniques presented for surface-surface bisectors in [17]. Suppose $P = S_1 + \alpha N_1$, and substitute for P in Equation 5.4(b) with $j = 2$ to obtain

$$\alpha = \frac{-\langle S_1 - S_2, S_1 - S_2 \rangle}{2\langle S_1 - S_2, N_1 \rangle} \quad (5.5)$$

Substituting for P and α in Equations 5.4(a)-(b) with $i = 2, 3$, $j = 3$, and simplifying yields a system of six equations in six variables. Denote $S_1 - S_i$ by $S_{1\mathcal{M}i}, i = 2, 3$.

$$2\langle S_{1\mathcal{M}2}, N_1 \rangle \langle S_{1\mathcal{M}i}, \frac{\partial S_i}{\partial u_i} \rangle - \parallel S_{1\mathcal{M}2} \parallel^2 \langle N_1, \frac{\partial S_i}{\partial u_i} \rangle = 0, i = 2, 3 \quad (5.6a)$$

$$2\langle S_{1\mathcal{M}2}, N_1 \rangle \langle S_{1\mathcal{M}i}, \frac{\partial S_i}{\partial v_i} \rangle - \parallel S_{1\mathcal{M}2} \parallel^2 \langle N_1, \frac{\partial S_i}{\partial v_i} \rangle = 0, i = 2, 3 \quad (5.6b)$$

$$\langle S_{1\mathcal{M}2}, N_1 \rangle \parallel S_{1\mathcal{M}3} \parallel^2 - \parallel S_{1\mathcal{M}2} \parallel^2 \langle N_1, S_{1\mathcal{M}3} \rangle = 0 \quad (5.6c)$$

$$\langle N_1 \times N_2, N_3 \rangle = 0 \quad (5.6d)$$

Note that Equations 5.4(a) for $i = 1$ are automatically satisfied. It is possible to solve the system of equations in 5.6(a)-(d) by symbolically representing the left hand sides of the equations and using subdivision based methods directly on the 6-variate functions [17], but the large memory requirements rendered it infeasible on an 8GB RAM machine. The expression tree based method presented in [15] significantly reduced the memory requirements but was still computationally infeasible due to the large number of dimensions.

To make this approach practical, we have improved the computation time for this problem using hierarchical and parallel computation strategies. Equations 5.6(a)-(b) for $i = 2$ are dependent only upon u_1, v_1, u_2, v_2. Therefore, the problem size is reduced by first finding subdomains that satisfy these two equations, and then finding subdomains that also satisfy Equations 5.6(a)-(d) for $i = 3$. Intuitively, Equations 5.6(a)-(b) $i = 2$ require two points on the surface that are equidistant from an offset point P in $\mathbb{R}^3$. The remaining equations find a subset of those points for which there is a third surface point that is also equidistant to P.

The expression tree based approach is used to subdivide u_1, v_1, u_2, v_2 parametric domains until a user specified threshold is reached while using interval arithmetic to reject subdomains that do not satisfy Equations 5.6(a)-(b) with $i = 2$. Also, subdomains that do not correspond to regions that satisfy the maximal condition are pruned. The maximal condition is checked by computing ranges of $\rho = \alpha \parallel N \parallel$ with interval arithmetic and testing whether 1) the range contains a part of the positive real line, and 2) is bounded by the size of the region enclosed within S. Negative values of ρ correspond to points outside the region since the offset would be in the outward normal direction and thus cannot contribute to the solution. Each of the remaining subdomains along with variables u_3, v_3 is then tested with Equations 5.6(a)-(d) using subdivision along u_3, v_3 parametric directions, interval arithmetic and pruning. Moreover, this step is performed in parallel since the subdomains are independent. Trivial solutions are ignored. The centers of remaining subdomains of u_i, v_i, $i = 1, 2, 3$ are used in a Newton-Raphson refinement step [17] to obtain accurate solutions and non-maximal solutions are ignored. This step is also performed using parallel processing techniques.

5.5 A_1^4 points

A_1^4 points are equidistant to four different points on S and the corresponding sphere is maximal. A_1^4 points can be annihilation or intermediate transition based on the direction of the flow of the four incident A_1^3 curves as presented in [20]. For a point $P \in \mathbb{R}^3$ to be an A_1^4 point, the following equations must be satisfied.

$$\langle P - S_i, \frac{\partial S_i}{\partial u_i} \rangle = 0, \quad \langle P - S_i, \frac{\partial S_i}{\partial v_i} \rangle = 0, \text{ i = 1,2,3,4} \quad (5.7a)$$

$$\parallel P - S_1 \parallel \quad = \quad \parallel P - S_j \parallel \text{, j = 2,3,4} \quad (5.7b)$$

Just as in the case for A_1^3 critical points, the system in Equation 5.7 of eleven equations is reduced using Equation 5.5 to a system in eight equations in eight variables u_i, v_i, $i = 1, 2, 3, 4$ shown in Equation 5.8(a)-(c).

$$2\langle S_{1\mathcal{M}2}, N_1 \rangle \langle S_{1\mathcal{M}i}, \frac{\partial S_i}{\partial u_i} \rangle - \parallel S_{1\mathcal{M}2} \parallel^2 \langle N_1, \frac{\partial S_i}{\partial u_i} \rangle = 0, i = 2, 3, 4 \quad (5.8a)$$

$$2\langle S_{1\mathcal{M}2}, N_1 \rangle \langle S_{1\mathcal{M}i}, \frac{\partial S_i}{\partial v_i} \rangle - \parallel S_{1\mathcal{M}2} \parallel^2 \langle N_1, \frac{\partial S_i}{\partial v_i} \rangle = 0, i = 2, 3, 4 \quad (5.8b)$$

$$\parallel S_{1\mathcal{M}j} \parallel^2 \langle S_{1\mathcal{M}2}, N_1 \rangle - \parallel S_{1\mathcal{M}2} \parallel^2 \langle N_1, S_{1\mathcal{M}j} \rangle = 0, j = 3, 4 \quad (5.8c)$$

Intuitively, Equations 5.8(a)-(b) with $i = 2$ require two points on the surface that are equidistant from an offset point P in $\mathbb{R}^3$. Equations 5.8(a)-(c) with $i = 3, j = 3$ find a third surface point that is also equidistant to P with the other two points. And finally, Equations 5.8(a)-(c) with $i = 4, j = 4$ find a fourth surface point that is equidistant to P with the other three points.

Equations 5.8(a)-(b) with $i = 2$ are dependent only upon u_1, v_1, u_2, v_2; Equations 5.8(a)-(c) with $i = 3, j = 3$ are dependent on $u_1, v_1, u_2, v_2, u_3, v_3$, and Equations 5.8(a)-(c) with $i = 4, j = 4$ are dependent on $u_1, v_1, u_2, v_2, u_4, v_4$. So this structure lends itself to a hierarchical technique similar to the A_1^3 critical point case. Equations 5.8(a)-(c) will provide exactly the same solutions for both $i = 3, j = 3$ and $i = 4, j = 4$, so this step is performed once and subdomains in the u_3, v_3 parametric directions are used for u_4, v_4 parametric directions, thereby effectively reducing the number of dimensions in the subdivision search stage from eight to six. Moreover, Equations 5.8(a)-(c) with $i = 2, 3, j = 3$ are identical to Equations 5.6(a)-(c). So the subdivision stage for A_1^3 critical point and A_1^4 point computations are combined. The refinement stage for Equations 5.8(a)-(c) are performed in parallel.

5.6 Catalog of Transition Events

At transition points, corresponding evolution curves are either created, annihilated or undergo intermediate transitions. We present a complete list of all generic transitions for the medial axis.

1. *Creation Events:*
   - i) At an A_3 critical point where κ_1 has a local maximum. An intersection curve segment is created with the two end points on an A_3 curve.
   - ii) At an A_1^2 critical point where the distance to S has a local minimum. An intersection curve loop of the evolving offset surfaces is created.
2. *Annihilation Events:*
   - i) At an A_1^2 critical point where the distance to S has a local maximum. An intersection curve loop ends.
   - ii) At an A_1^3 critical point where two A_1^3 curves flow inward, and the corresponding intersection curves disappear.
   - iii) At an A_1^4 point where four A_1^3 curves flow inward, and the corresponding intersection curves end.
3. *Intermediate Transition Events:*
   - i) At an A_3 critical point where κ_1 has a local saddle point, at which two intersection curves join.
   - ii) At an A_1^2 critical point where the distance to S has a local saddle so two intersection curves meet and exchange branches.
   - iii) At an A_1A_3 point where the intersection curve from a smooth sheet passes and creates an A_1^3 point which evolves along an A_1^3 curve away from the A_1A_3 point.
   - iv) At an A_1A_3 point where an A_1^3 curve flows into it and the intersection curves meeting at the transition point transform into a smooth curve on a smooth medial sheet.
   - v) At an A_1^3 critical point where three intersection curves on different A_1^2 surfaces meet and split into two triples of intersection curve segments which meet at the end points on A_1^3 curves flowing outward.
   - vi) At an A_1^3 critical point where two intersection curves on different A_1^2 surfaces meet and split into two pairs of intersection curve segments meeting at the end points on A_1^3 curves flowing outward. Also, a new intersection curve on a third A_1^2 surface is created with the same end points on the A_1^3 curve.
   - vii) Two intersection curves on distinct A_1^2 surfaces meet at two regular A_1^3 points, where two intersection curves on a third A_1^2 surface also end. The two A_1^3 curves flow into the critical point, the first two intersection curves on different A_1^2 surfaces collapse to the critical point, and the other two intersection curves on the third A_1^2 surface merge into a single intersection curve.
   - viii) Three A_1^3 curves meet at an A_1^4 point with a transition to one A_1^3 curve flowing outward from the A_1^4 point. Intersection curves are modified accordingly.
   - ix) Two A_1^3 curves meet at an A_1^4 point with a transition to the other two A_1^3 curves. Intersection curves are modified accordingly.

6 Curve Elements

6.1 A_3 Curves

Ridges are loci of points on a surface at which one of the principal curvatures attains a critical value along its principal direction [23]. A ridge point of κ_1 satisfies $\langle \nabla\kappa_1, t_1\rangle = 0$, where principal direction t_1 is a 2D column vector with the two elements of the vector denoting coefficients of S_u and S_v at $S(u, v)$. A κ_1 ridge point is called *elliptic* if κ_1 attains a local maximum along t_1 [33].

A_3 curves on the medial axis correspond to loci of elliptic ridge points of κ_1 on the surface such that the sphere centered at the center of curvature corresponding to κ_1 is maximally inscribed within the region of interest. Ridges on $S(u, v)$ are computed using techniques presented in [33] that guarantee robust and accurate extraction of all ridges on B-spline surfaces. Ridge curves are output as polylines in [33]. Elliptic ridges of κ_1 are identified by testing the extremum type condition at each ridge point vertex of the polylines. Only those points that pass the maximal condition are retained resulting in segments or closed curve loops (See Appendix A). The end points of each non-loop segment occur where the maximal condition first fails and corresponds to an A_1A_3 point. A_3 curves on the medial axis are then computed as offsets of elliptic ridges at distances $\frac{1}{\kappa_1}$ (radius of curvature) along the inward normal of the surface at each point. In Equation 6.1, $\mathcal{R}$ is a collection of m ridge curves $\mathcal{R}_j$ that correspond to A_3 curves. Each $\mathcal{R}_j$ is a connected set of elliptic ridge points on S.

$$\mathcal{R} = \{\bigcup_{j=1}^{m} \mathcal{R}_j\},\ \mathcal{R}_j = \{(u,v) : \langle \nabla\kappa_1(u,v), t_1(u,v)\rangle \ = \ 0,$$
$$t_1^T \begin{bmatrix} \kappa_{1uu} & \kappa_{1uv} \\ \kappa_{1uv} & \kappa_{1vv} \end{bmatrix} t_1 < 0, \quad b_{max}(u,v) = (C(u,v), \frac{1}{\kappa_1(u,v)})\}$$
$$C(u,v) = S(u,v) + \frac{1}{\kappa_1(u,v)} n(u,v) \tag{6.1}$$

where $b_{max}(u, v)$ is a maximally inscribed sphere with center $C(u, v)$ and radius $\frac{1}{\kappa_1(u,v)}$. The A_3 curves on the medial axis are the loci of the sphere centers $C(u, v)$ for all $\mathcal{R}_j$.

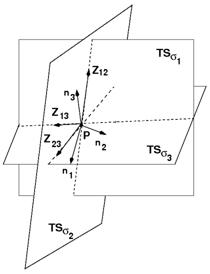

Fig. 4. Tangent planes of offset surfaces σ_i with normals n_i, $i = 1, 2, 3$ at a point P. Pairwise intersections of offset surfaces are along Z_{12}, Z_{23}, Z_{13}.

Identifying A_1A_3 points on A_3 curves. By definition, A_1A_3 points are locations where the maximal sphere is tangent to the corresponding ridge point on the surface as well as to another point in a different region of the surface. For each end point of a non-loop segment where the maximal condition fails, the other surface point corresponding to the A_1 condition is determined by finding closest points on the surface with distance equal to the radius of curvature ($\frac{1}{\kappa_1}$) at the ridge point using Equation A.1 in Appendix A.

6.2 A_1^3 Curves

This section presents a tracing algorithm for computing A_1^3 curves. A_1^3 critical points, A_1A_3 points and A_1^4 points, the computation of which is presented in Section 5, are source and sink points for the proposed algorithm. The regular points of A_1^3 curves correspond to points where three different regions of the of the deforming boundary intersect transversely (i.e., the three tangent planes of the offset surface at the intersection point are different). Using this characterization, we construct evolution vector fields based on methods and ideas from [12] to follow the time trace of A_1^3 points. When these vector fields are integrated beginning at an A_1^3 point, we obtain the A_1^3 curve from that point.

6.2.1 Evolution Vector Field for the A_1^3 Curves

Consider a triple intersection point $P \in \mathbb{R}^3$ at which the three offset surfaces intersect transversely. Denote the three different surface regions as S_i, with unit surface normal vectors n_i, and the corresponding offset surfaces as $\sigma_i(u_i, v_i, t)$, $i = 1, 2, 3$ respectively. Let $\phi_i = \sigma_{i_{u_i}} \times \sigma_{i_{v_i}}$ (Note $\phi_i \parallel n_i$). Since the tangent planes at P are all different, n_i, $i = 1, 2, 3$ span $\mathbb{R}^3$. $Z_{ij} = \phi_i \times \phi_j$, $i = 1, 2, 3$, $j = (i + 1)$ *mod* 3, are tangent vectors to the intersection curves of the pair of offset surfaces σ_i and σ_j. A linear algebra argument shows that $\{Z_{12}, Z_{23}, Z_{13}\}$ also forms a basis for $\mathbb{R}^3$ at P. Also, because both Z_{ij} and Z_{jk} are orthogonal to n_j, $i = 1, 2, 3$, $j = (i+1)$ *mod* 3, $k = (i+2)$ *mod* 3, they span the tangent plane of σ_j at P. (See Figure 4).

At each point in the neighborhood of P, we write

$$\begin{aligned} n_1 - n_2 &= a_1 Z_{12} + b_1 Z_{13} + c_1 Z_{23} \\ n_3 - n_2 &= a_2 Z_{12} + b_2 Z_{13} + c_2 Z_{23} \end{aligned} \tag{6.2}$$

From Equation 6.2, the evolution vector field η in a neighborhood of P in $\mathbb{R}^3$ is given by Equation 6.3, which is given in a representation in terms of the normal vector field and tangent vector fields to each offset surface.

$$\begin{aligned} \eta &= n_1 - \big((a_1 - a_2)Z_{12} + b_1 Z_{13}\big) \\ &= n_2 + \big(c_1 Z_{23} + a_2 Z_{12}\big) \\ &= n_3 - \big(b_2 Z_{13} + (c_2 - c_1)Z_{23}\big) \end{aligned} \tag{6.3}$$

The integral curve of η through P is a curve of triple intersection points and hence follows the evolution of the A_1^3 curve until a sink point is reached.

$$\frac{d\chi}{dt} = \eta(\chi), \quad \chi(0) = P, \quad \chi(t) \in \mathbb{R}^3 \tag{6.4}$$

Furthermore, define vector fields on the parameter-time space as

$$\nu_i = e_t + \alpha_i e_{u_i} + \beta_i e_{v_i}, \; i = 1, 2, 3 \tag{6.5}$$

$$\begin{aligned} \alpha_1 &= (a_1 - a_2)\langle -\phi_2, \sigma_{1_{v_1}} \rangle + b_1 \langle \phi_3, \sigma_{1_{v_1}} \rangle \\ \beta_1 &= (a_1 - a_2)\langle \phi_2, \sigma_{1_{u_1}} \rangle - b_1 \langle \phi_3, \sigma_{1_{u_1}} \rangle \\ \alpha_2 &= -a_2 \langle \phi_1, \sigma_{2_{v_2}} \rangle + c_1 \langle \phi_3, \sigma_{2_{v_2}} \rangle \\ \beta_2 &= a_2 \langle \phi_1, \sigma_{2_{u_2}} \rangle - c_1 \langle \phi_3, \sigma_{2_{u_2}} \rangle \\ \alpha_3 &= b_2 \langle -\phi_1, \sigma_{3_{v_3}} \rangle + (c_1 - c_2)\langle \phi_2, \sigma_{3_{v_3}} \rangle \\ \beta_3 &= b_2 \langle \phi_1, \sigma_{3_{u_3}} \rangle - (c_1 - c_2)\langle \phi_2, \sigma_{3_{u_3}} \rangle \end{aligned} \tag{6.6}$$

and e_l denotes the unit vector in the parameter-time space direction l, $l = t, u_1, v_1, u_2, v_2, u_3, v_3$. Then, $\eta = d\sigma_i(\nu_i)$, $i = 1$ or 2 or 3. This implies that the integral curves of ν_i are mapped by σ_i to integral curves of η. The corresponding integral curves of ν_i will trace the evolution of the intersection curves in the parameter space.

6.2.2 Tracing Algorithm for A_1^3 Curves

We first classify the critical points and end points as source and sink points. A_1A_3 source points each provide a start A_1^3 point that is computed using local geometric properties of the medial axis [20]. Source A_1^3 critical points provide two start A_1^3 points computed on either side of the normal to the plane containing the three surface points. A_1^4 source points provide one or two start A_1^3 points as determined using local geometric properties of the medial axis presented in [20]. Algorithm 2 is used to trace all A_1^3 curves.

α_i, β_i are computed by first solving for a_j, b_j, c_j, $j = 1, 2$ from Equation 6.2 and substituting in Equation 6.6. In order to avoid numerical errors accumulating over time, a refinement step for the system of Equations 5.4 (a), (b) along with $\parallel P - S_1 \parallel = t + dt$ to project points accurately onto A_1^3 curves [17].

7 A_1^2 Surfaces

This section presents evolution vector fields to sweep out intersection curves of offset surfaces under the eikonal flow. An algorithm to compute the surfaces of the medial axis using the evolution method is then presented. Between transition points, the evolution vector fields are integrated to

Algorithm 2 Tracing A_1^3 curves

```
INPUT: SRC, SINK
OUTPUT: $C_{all}$, the set of all $A_1^3$ curves
begin
  $C_{all} := \emptyset$
  for $p := (u_1, v_1, u_2, v_2, u_3, v_3, t) \in SRC$
    $C := \{p\}$
    while $p \notin SINK$ do
      $u_i := u_i + \alpha_i dt;\ v_i := v_i + \beta_i dt, i = 1, 2, 3$
      $p := refine(u_1, v_1, u_2, v_2, u_3, v_3, t + dt)$
      $C := C \bigcup \{p\}$
    od
    $C_{all} := C_{all} \bigcup C$
end
```

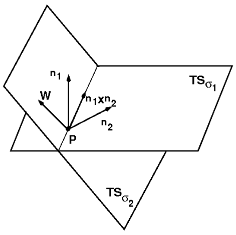

Fig. 5. Tangent planes of offset surfaces σ_i with normals n_i, $i = 1, 2$, intersecting along $n_1 \times n_2$ at a point P. $W \in TS_{\sigma_2}$.

compute a time trace of the evolving intersection curves that, together with transition points, forms the medial axis A_1^2 surfaces. The construction of the evolution vector fields will follow the same ideas used in the preceding section for A_1^3 curves. It is an adaptation of that given in [10] for the evolution of intersection curves of two different surfaces under generalized offset flows. In our case, it is applied to the self-intersections of the offsets of a single surface under the eikonal flow.

7.1 Evolution Vector Field for Intersection Curves

Consider two separate surface regions S_i with unit normals n_i and the offset surfaces under the eikonal flow by $\sigma_i(u_i, v_i, t)$, $i = 1, 2$, respectively at a point P on an intersection curve. If P is not a transition point, then σ_1 and σ_2 are not tangent and hence, n_1 and n_2 are independent. $n_1 \times n_2$ is tangent to the intersection curve. $W = n_2 \times (n_1 \times n_2)$ is in the tangent plane of σ_2. W is also orthogonal to the intersection curve (Figure 5). Since n_1 and n_2 are independent, $\{\sigma_{1_{u_1}}, \sigma_{1_{v_1}}, W\}$ are independent vector fields, and hence a basis for $\mathbb{R}^3$. Thus, in the the neighborhood of P, we write

$$n_2 - n_1 = a_1\sigma_{1_{u_1}} + b_1\sigma_{1_{v_1}} - W \tag{7.1}$$

Since W lies in the tangent plane of σ_2,

$$W = a_2\sigma_{2_{u_2}} + b_2\sigma_{2_{v_2}} \tag{7.2}$$

Combining Equations 7.1 and 7.2, we obtain two equivalent representations of the evolution vector field in the neighborhood of P in $\mathbb{R}^3$ given by

$$\xi = n_1 + a_1\sigma_{1_{u_1}} + b_1\sigma_{1_{v_1}} = n_2 + a_2\sigma_{2_{u_2}} + b_2\sigma_{2_{v_2}} \tag{7.3}$$

By arguments analogous to those used in [10], ξ is tangent to the surface formed from the union of the evolving intersection curves of σ_1 and σ_2, which is the A_1^2 surface. Thus, we can follow the evolution of the intersection curve (where the evolving curves remain transverse) by integrating the vector field ξ with initial conditions as the points on the intersection curve.

$$\frac{d\chi}{dt} = \xi(\chi), \quad \chi(0) = P, \quad \chi(t) \in \mathbb{R}^3 \tag{7.4}$$

Furthermore, define vector fields on the parameter-time space as $\zeta_i = e_t + a_i e_{u_i} + b_i e_{v_i}$, where e_l denotes the unit vector in the parameter-time space direction l, $l = t, u_1, v_1, u_2, v_2$. Then, $\xi = d\sigma_1(\zeta_1) = d\sigma_2(\zeta_2)$. This implies that the integral curves of the ζ_i are mapped by σ_i to integral curves of ξ. The corresponding integral curves of ζ_i will trace the evolution of the intersections curves in the parameter space.

7.2 Algorithm for Computing A_1^2 Surfaces

Given a connected set of samples representing an intersection curve at time t, a discrete marching algorithm is used to trace each point onto a new intersection curve at time $t{+}dt$ using Equation 7.4. Suppose, $P = \sigma_i(u_i^p, v_i^p, t)$ on an intersection curve evolves to a point $Q = \sigma_i(u_i^q, v_i^q, t{+}dt)$ after a small time dt, then $(u_i^q, v_i^q) = (u_i^p + a_i dt, v_i^p + b_i dt)$. a_i and b_i can be obtained by solving Equations 7.1 and 7.2. In order to avoid numerical errors accumulating over time, the middle point algorithm [4] is used to project points accurately onto intersection curves and refine parameter values simultaneously. Points on an intersection curve are resampled at every time step by adaptively inserting or removing points such that they are uniformly spaced in $\mathbb{R}^3$.

Starting from $t = 0$, Algorithm 3 is used to compute A_1^2 surfaces. Intersection curves can have A_3 or A_1^3 points at curve ends after certain transitions. Since A_3 and A_1^3 curves are computed in a prior step, the evolution of such end points is performed by tracking points on corresponding A_3 and A_1^3 curves. A_1A_3 and A_1^4 points occur at transition events only. Such points are added to intersection curve ends during the transition to maintain topological structure of the medial axis. Recall from Section 2 that every medial axis point is associated with a set of parameter values, one for each contact point on S. Each parameter designates a distinct region of S, where the intersection of the offsets at time given by the radius of the maximal sphere results in the medial axis point. During transitions, the correspondences between parameter values of two interacting intersection curves are obtained using distance in parameter space to consistently identify distinct regions of S.

8 Results and Discussion

This section presents examples of 3D medial axis computation using the proposed approach. In each example, the region of interest is bounded by a single tensor product parametric biquintic B-spline surface. As explained in

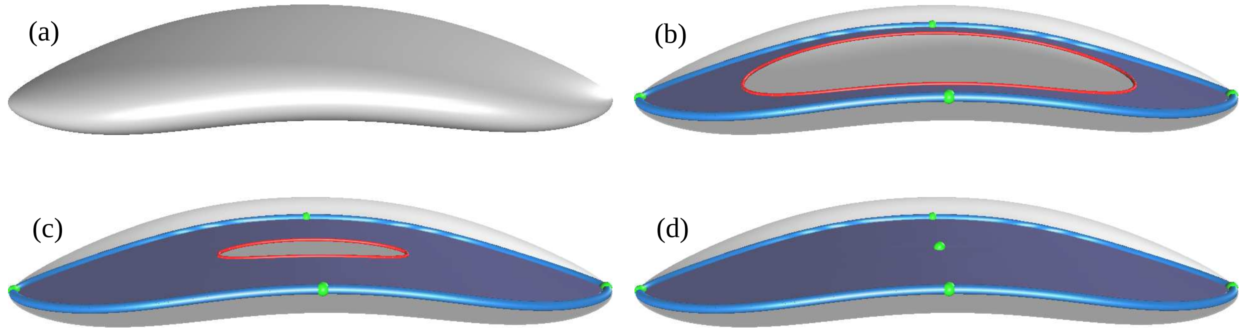

Fig. 6. Several stages of evolution of medial axis of a deformed ellipsoid. Transition points are shown in green, evolving intersection curves are shown in red.

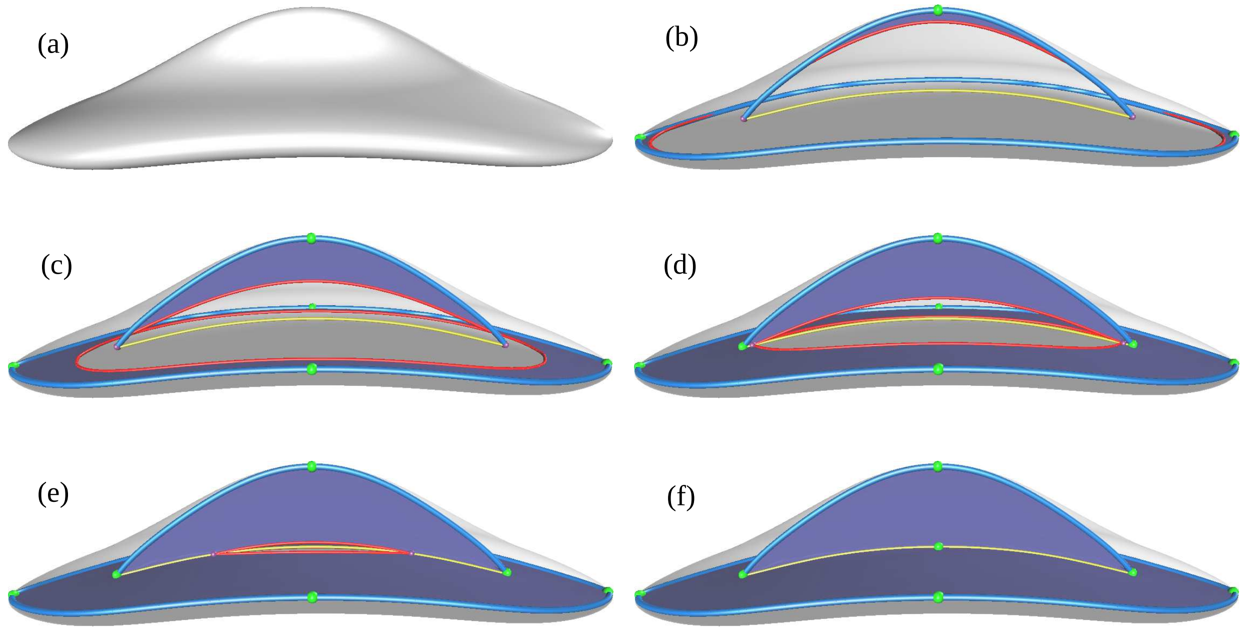

Fig. 7. Several stages of evolution of medial axis of a deformed ellipsoid with a fin. Transition points are shown in green, evolving intersection curves are shown in red.

Algorithm 3 Computing surface sheets

```
(i) Sort transition points in order of increasing time.
(ii) Increment $t$ by small timestep $dt$.
(iii) If no transition points are encountered, evolve all current intersection curves to $t + dt$.
(iv) Otherwise, perform transition for each transition point encountered in increasing order of $t$ and evolve intersection curves not involved in the transition event.
(v) If all transitions have been completed, then stop. Otherwise, repeat Step (ii).
```

Section 4, the medial axis singular set and transition points are computed prior to computing surface sheets of a medial axis. Figures 6, 7 and 8 show several steps of the evolution of the (self-)intersecting set of the eikonal flow for various shapes. In these figures, A_1A_3 points are indicated by purple spheres, A_1^4 points by dark blue spheres, A_3 curves by thick blue curves, A_1^3 curves by thick yellow curves and A_1^2 surface sheets in dull violet. Transition points are shown as green spheres and evolving intersection curve fronts are shown as thick red curves.

Figure 6 shows several steps of the A_1^2 surface computation for a deformed ellipsoid shape. The medial axis singular set for this example is a single A_3 curve. The evolution starts at the left and right extremities of the shape that correspond to curvature maximum creation points. The intersection curves merge into a single loop at two simultaneous curvature saddle points that then annihilates at a distance maximum point. The medial axis therefore consists of a single A_1^2 surface bounded by a closed A_3 curve.

Figure 7 shows several steps of the A_1^2 surface computation for a deformed ellipsoid shape with a fin. The medial axis singular set for this example consists of a closed A_3 curve, another A_3 curve segment with an A_1A_3 point at both ends and an A_1^3 curve connecting the two A_1A_3 points. Intersection curves are created at curvature maximum points at the left and right extremities of the shape, as well as at the top of the model corresponding to the fin. The intersection curves corresponding to the bottom sur-

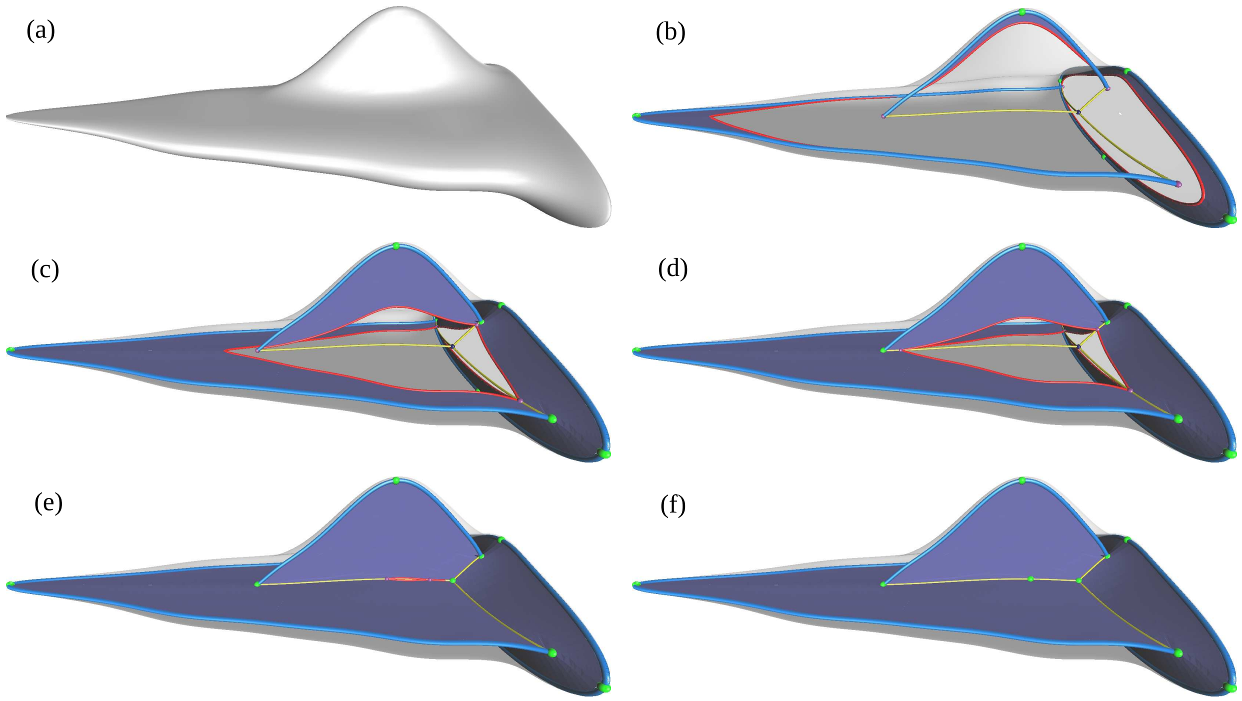

Fig. 8. Several stages of evolution of medial axis of an object with multiple interconnected medial surfaces. Transition points are shown in green, evolving intersection curves are shown in red.

face merge into a single loop at two simultaneous curvature saddle points. The intersection curves then transition at the two A_1A_3 points and annihilate at a critical point on the A_1^3 curve.

Figure 8 shows several steps of the A_1^2 surface computation for a more complicated shape. Intersection curves are created at curvature maximum points (Figure 8(b)). An A_3 curve loop is formed after corresponding intersection curve segments merge at curvature saddle points (right most A_3 curve shown in Figure 8(b)). The intersection curves undergo further transitions at fin points (Figure 8(c) and (d)) and then evolve through the A_1^4 point where three A_1^3 curves flow into the A_1^4 point and a fourth one flows outward (Figure 8(e)). The intersection curves finally annihilate at a critical point on the A_1^3 curve (Figure 8(f)). The medial axis for this shape exhibits all the generic structural elements presented in Section 2. There are two A_3 curve segments each having an A_1A_3 fin point at both ends. There is also one A_3 curve loop. Four A_1^3 curve segments meet at an A_1^4 point shown as a dark blue sphere (Figure 8(b)). The other ends of the four A_1^3 curve segments correspond to the A_1A_3 fin points where they meet A_3 curve segments.

In addition to accurate points on the medial axis, the proposed approach also accurately captures the topological structure of the medial axis. Figure 1 shows all the surfaces as well as entities of the singular set colored differently. In addition, since the evolution tracing is performed in parametric space, the boundary locations corresponding to each medial axis point is also known.

Although the approach is presented for the generic situation, the examples presented above do have certain non-genericities. For example, in the examples shown in Figure 6 and Figure 7, there are two simultaneously occurring curvature saddle point transitions that modify the same set of evolving intersection curves. Further, in the example shown in Figure 7, there are two simultaneously occurring A_1A_3 points. These situations are addressed as special cases in the implementation. The examples shown in Figure 6 and Figure 7 also have degenerate points that are sharp corners at the left and right extremities. These points are computed as curvature maximum points having a very small radius. Critical points with more than four tangency points will be identified as A_1^4 points with the four tangency points being a subset of the actual number of points. Symmetrical regions (e.g., spherical or cylindrical regions) also result in non-generic situations.

Table 1
Computation Times (TP = Transition Point, C = Curves)

Biquintic surfaces	A_3 TP	A_1^2 TP	$A_1^3 + A_1^4$ TP	A_3 C, A_1A_3	A_1^3 C	Surfaces
(control points)	(min)	(sec)	(min)	(min)	(sec)	(sec)
Fig. 6: 13 × 9	18	35	40	4.4	-	2
Fig. 7: 15 × 9	23	36	40	5.3	1	3
Fig. 8: 17 × 15	55	59	90	40	1	7

The computational complexity of the proposed algorithm is directly proportional to the number of transition points since it bounds the total number of intersection curves over the entire evolution for a given model. Likewise, the number of critical points, A_1A_3 points and A_1^4 points bound the number and hence computational complexity of A_3 and A_1^3 curve tracing. Table 1 presents running times on an Intel x64 machine with four cores and 8GB RAM for the examples shown in this paper. It is evident from the table that most of the time was spent in computing the transition points since that involves B-spline root finding. We are currently investigating further enhancements to the root finding algorithms to reduce computation time. Although the total running time is longer than is desirable, each stage of the proposed approach is automatic, presenting accurate solutions with topology. Therefore, time consuming manual pruning and structure extraction steps are not required.

The techniques for computing transition points of A_3, A_1^2 and A_1A_3 types provide all solutions. In order to reduce computation time for A_1^3 and A_1^4 transition points, they were required to be at least further apart than 5% of the size of the parametric domain in the examples shown. Therefore, no two transition events could occur within a region of this size. The topology of the computed medial axis is accurate up to this specified accuracy. Between transition events, a much higher accuracy (10^{-4} of the size of the model in $\mathbb{R}^3$) was used to locate the medial axis points since the evolution of the intersection curves is smooth.

In the current implementation, it is assumed that the boundary surface is given by a single B-spline surface. However, the theoretical results are quite general and extend to regions bounded by multiple surface patches stitched together since we present results considering different regions of a single surface. For multi-surface models, computation of transition points will involve a combinatorial search with subsets of surfaces ($O(n^k)$ combinations for n-surface model for A_1^k point type). The implementation of algorithms for computing curve elements and medial surfaces will involve keeping track of the boundary surfaces involved along with corresponding parameter values. Further, the approach is not restricted to simply connected regions having genus zero even though the examples shown in the paper are such types of objects. We are also investigating efficient techniques to address multi-surface models that also includes objects with multiply connected surfaces having higher genus.

9 Conclusions

A new algorithm for computing the medial axis of regions in $\mathbb{R}^3$ bounded by tensor product parametric B-spline surfaces is presented. The generic structure of the 3D medial axis consists of A_1^2 surfaces along with a singular set of A_3 curves, A_1^3 curves, A_1A_3 points and A_1^4 points. The proposed approach is based on the eikonal or grassfire flow of the bounding surfaces along the inward surface normal direction. The eikonal flow results in special transition points that create, modify or annihilate evolving (self-) intersection curves of the corresponding offset surfaces. The transition points as well as A_1A_3 and A_1^4 points are computed by solving geometric equations using B-spline based root finding techniques. The geometric equations are of high degree and dimensions and several strategies for reducing the computation time are provided. A_3 curves are computed using the technique presented in [33]. A_1^3 curves are computed using a new evolution based tracing approach. A_1^2 surfaces of the medial axis are computed as a time trace of the evolving (self-) intersection set under the eikonal flow using an adaptation of the technique presented in [10] in conjunction with techniques for addressing special transition events of the eikonal flow.

Existing discrete techniques introduce artifacts that are not part of the medial representation and considerable effort is required to remove them. In addition, the output of existing techniques is typically a set of discrete elements without connectivity and structural information. Considerable manual effort is required in order to infer this information. Existing approaches for smooth surface representations compute only partial solution of the medial axis, specifically A_1^3 curves and A_1^4 points, for a restricted set of shapes. The proposed approach enables accurate computation of the complete medial axis as well as correct topological structure for regions bounded by freeform surfaces. The proposed approach does not generate non-medial artifacts and also computes parameter values of the bounding surface points corresponding to each medial axis point. Ongoing work includes further optimizations of the transition point computation strategies to improve computation time, and extension of the proposed work for multi-surface models and to address other non-generic situations.

10 Acknowledgments

This work was supported in part by the University of Utah Graduate Research Fellowship, NSF DMS-0706941 and DARPA grant HR0011-09-1-0055. All opinions, findings, conclusions or recommendations expressed in this document are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

References

[1] N. Amenta, S. Choi, and R. Kolluri. The power crust. In *Proceedings of 6th ACM Symposium on Solid Modeling*, pages 249–260, 2001.

[2] N. Amenta, S. Choi, and R. Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications*, 19(2-3):127–153, 2001.

[3] G. Baja and S. Svensson. Surface Skeletons Detected on the D6 Distance Transform. In *Proceedings of the Joint IAPR Int. Workshops on Advances in Pattern Recognition*, pages 387–396. Springer-Verlag, 2000.

[4] R. Barnhill, G. Farin, M. Jordan, and B. Piper. Surface/surface intersection. *Computer Aided Geometric Design*, 4(1-2):3–16, 1987.

[5] A. Belyaev, I. Bogaevski, and T. Kunii. Principal direction ridges. Technical report, Technical Report 96-4-001, Center for Mathematical Sciences, The University of Aizu, Japan, 1996.

[6] S. Biasotti, D. Attali, J. Boissonnat, H. Edelsbrunner, G. Elber, M. Mortara, G. Baja, M. Spagnuolo, M. Tanase, and R. Veltkamp. Skeletal structures. *Shape Analysis and Structuring*, pages 145–183, 2008.

[7] H. Blum and R. Nagel. Shape description using weighted symmetric axis features. *Pattern recognition*, 10(3):167–180, 1978.

[8] G. Borgefors, I. Nyström, and G. S. di Baja. Computing skeletons in three dimensions. *Pattern Recognition*, 32(7):1225–1236, 1999.

[9] F. Chazal and A. Lieutier. Stability and homotopy of a subset of the medial axis. In *SM '04: Proceedings of the ninth ACM symposium on Solid modeling and applications*, pages 243–248, Aire-la-Ville, Switzerland, 2004. Eurographics Association.

[10] X. Chen, R. F. Riesenfeld, E. Cohen, and J. Damon. Theoretically-based algorithms for robustly tracking intersection curves of deforming surfaces. *Compuer-Aided Design*, 39(5):389–397, 2007.

[11] T. Culver, J. Keyser, and D. Manocha. Accurate computation of the medial axis of a polyhedron. In *Proceedings of the fifth ACM symposium on Solid modeling and applications*, pages 179–190. ACM New York, NY, USA, 1999.

[12] J. Damon. Smoothness and geometry of boundaries associated to skeletal structures I: Sufficient conditions for smoothness. *Annales de l'Institut Fourier*, 53(6):1941–1985, 2003.

[13] T. Dey and W. Zhao. Approximating the medial axis from the Voronoi diagram with a convergence guarantee. *Algorithmica*, 38(1):179–200, 2003.

[14] T. Dey and W. Zhao. Approximate medial axis as a voronoi subcomplex. *Computer-Aided Design*, 36(2):195–202, 2004.

[15] G. Elber and T. Grandine. Efficient solution to systems of multivariate polynomials using expression trees. In *IEEE Int. Conference on Shape Modeling and Applications, 2008*, pages 163–169, 2008.

[16] G. Elber and M. Kim. Computing rational bisectors. *IEEE Computer Graphics and Applications*, pages 76–81, 1999.

[17] G. Elber and M. Kim. Geometric constraint solver using multivariate rational spline functions. In *Proceedings of the sixth ACM symposium on Solid modeling and applications*, pages 1–10. ACM New York, NY, USA, 2001.

[18] M. Foskey, M. Garber, M. Lin, and D. Manocha. A voronoi-based hybrid planner. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 1, pages 55–60, 2001.

[19] M. Foskey, M. Lin, and D. Manocha. Efficient computation of a simplified medial axis. *Journal of Computing and Information Science in Engineering*, 3:274, 2003.

[20] P. Giblin and B. Kimia. A formal classification of 3D medial axis points and their local geometry. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(2):238–251, 2004.

[21] L. Guibas, C. Holleman, and L. Kavraki. A probabilistic roadmap planner for flexible objects with a workspace medial-axis-based sampling approach. In *Proceedings of 1999 IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, volume 1, pages 254–259. IEEE, 1999.

[22] H. Gürsoy. *Shape interrogation by medial axis transform for automated analysis.* PhD thesis, M.I.T., U.S.A, 1990.

[23] P. Hallinan, G. Gordon, A. Yuille, P. Giblin, and D. Mumford. *Two-and three-dimensional patterns of the face.* AK Peters, Ltd. Natick, MA, USA, 1999.

[24] I. Hanniel and G. Elber. Computing the voronoi cells of planes, spheres and cylinders in r3. *Computer Aided Geometric Design*, 26(6):695–710, 2009.

[25] M. Held. VRONI: An engineering approach to the reliable and efficient computation of Voronoi diagrams of points and line segments. *Computational Geometry*, 18(2):95–123, 2001.

[26] M. Hisada, A. Belyaev, and T. Kunii. A Skeleton-based Approach for Detection of Perceptually Salient Features on Polygonal Surfaces. In *Computer Graphics Forum*, volume 21, pages 689–700, 2002.

[27] C. Hoffman. How to construct the skeleton of CSG objects. The Mathematics of Surfaces, IVA, Bowyer and Davenport Eds, 1990.

[28] C. Hoffmann. *Geometric and solid modeling: an introduction*, volume 146. Morgan Kaufmann, 1989.

[29] S. Joshi and et al. Multiscale deformable model segmentation and statistical shape analysis using medial descriptions. *IEEE Trans. on Medical Imaging*, 21(5):538–550, 2002.

[30] L. Lam, S. Lee, and C. Suen. Thinning methodologies-a comprehensive survey. *IEEE Trans. on pattern analysis and machine intelligence*, 14(9):869–885, 1992.

[31] F. Leymarie and B. Kimia. The medial scaffold of 3d unorganized point clouds. *IEEE trans. on pattern analysis and machine intelligence*, pages 313–330, 2007.

[32] J. Mather. Distance to a manifold in euclidean space. In *Proceedings of the Symposium on Pure Mathematics 40 (part II)*, pages 199–216. American Mathematical Society, 1983.

[33] S. Musuvathy, E. Cohen, J. Seong, and J. Damon. Tracing ridges on B-spline surfaces. In *2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling*, pages 55–66. ACM, 2009.

[34] S. Pizer and et al. Deformable m-reps for 3 D medical image segmentation. *Int. Journal of Computer Vision*, 55(2):85–106, 2003.

[35] R. Quadros, K. Ramaswami, F. Prinz, and B. Gurumoorthy. LayTracks: a new approach to automated geometry adaptive quadrilateral mesh generation using medial axis transform. *Int. Journal for Numerical Methods in Engineering*, 61:209–237, 2004.

[36] M. Ramanathan and B. Gurumoorthy. Constructing medial axis transform of extruded and revolved 3D objects with free-form boundaries. *Computer-Aided Design*, 37(13):1370–1387, 2005.

[37] M. Ramanathan and B. Gurumoorthy. Interior medial axis transform computation of 3 d objects bound by free-form surfaces. *Computer-Aided Design*, 42(12):1217–1231, 2010.

[38] D. Sheehy, C. Armstrong, and D. Robinson. Shape Description By Medial Surface Construction. *IEEE Trans. on Visualization and Computer Graphics*, 2(1):62–72, 1996.

[39] A. Sheffer, M. Etzion, A. Rappoport, and M. Bercovier. Hexahedral mesh generation using the embedded Voronoi graph. *Engineering with Computers*, 15(3):248–262, 1999.

[40] E. Sherbrooke, N. Patrikalakis, and E. Brisson. An algorithm for the medial axis transform of 3d polyhedral solids. *IEEE trans. on visualization and computer graphics*, pages 44–61, 1996.

[41] K. Siddiqi, S. Bouix, A. Tannenbaum, and S. Zucker. Hamilton-jacobi skeletons. *Int. Journal of Computer Vision*, 48(3):215–231, 2002.

[42] K. Siddiqi and S. Pizer. *Medial representations: mathematics, algorithms and applications.* Springer Verlag, 2008.

[43] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, and S. Dickinson. Retrieving articulated 3-D models using medial surfaces. *Machine Vision and Applications*, 19(4):261–275, 2008.

[44] A. Sud, M. Foskey, and D. Manocha. Homotopy-preserving medial axis simplification. In *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pages 39–50, 2005.

[45] Y. ZHANG and P. WANG. Analytical comparison of thinning algorithms. *Int. journal of pattern recognition and artificial intelligence*, 7(5):1227–1246, 1993.

A Maximal Condition Check

In several steps of the algorithms presented in this paper, it is necessary to check whether a given point P corresponds to the center of a sphere with radius d that is maximally inscribed within the region bounded by $\mathcal{B}$. In this work, this condition is checked by first computing the closest point on $S(u,v)$ to P. Closest points are computed by first computing points on the surface where the distance to P, given by $\parallel S(u,v) - P \parallel$, attains a critical value. Such points are obtained by finding simultaneous roots of Equation A.1 using robust B-spline equation solvers [17,15].

$$\langle S - P, S_u \rangle \; = \; 0, \quad \langle S - P, S_v \rangle \; = \; 0 \tag{A.1}$$

All roots are then inspected to select the one with the smallest distance. For P to correspond to a maximally inscribed sphere, the smallest distance must equal d.