Published in final edited form as:

Med Biol Eng Comput. 2009 September ; 47(9): 989–999. doi:10.1007/s11517-009-0501-9.

Automatic Identification and Truncation of Boundary Outlets in Complex Imaging-Derived Biomedical Geometries

Xiangmin Jiao,

Department of Applied Mathematics & Statistics, Stony Brook University, Stony Brook, NY. E-mail: jiao@ams.sunysb.edu

Daniel R. Einstein,

Biological Monitoring & Modeling, Pacific Northwest National Laboratory, Richland, WA. E-mail: daniel.einstein@pnl.gov

Vladimir Dyedov, and

Department of Applied Mathematics & Statistics, Stony Brook University, Stony Brook, NY. E-mail: Vladimir@ams.sunysb.edu

James P. Carson

Biological Monitoring & Modeling, Pacific Northwest National Laboratory, Richland, WA. E-mail: james.carson@pnl.gov

Abstract

Efficient and accurate reconstruction of imaging-derived geometries and subsequent quality mesh generation are enabling technologies for both clinical and research simulations. A challenging part of this process is the introduction of computable, orthogonal boundary patches, namely the *outlets*, into treed structures, such as vasculature, arterial or airway trees. We present efficient and robust algorithms for automatically identifying and truncating the outlets for complex geometries. Our approach is based on a conceptual decomposition of objects into tips, segments, and branches, where the tips determine the outlets. We define the tips by introducing a novel concept called the *average interior center of curvature (AICC)* and identify the tips that are stable and noise resistant. We compute well-defined orthogonal planes, which truncate the tips into outlets. The rims of the outlets are connected into curves, and the outlets are then closed using Delaunay triangulation. We illustrate the effectiveness and robustness of our approach with a variety of complex lung and coronary artery geometries.

Keywords

medical imaging; outlets; automation; mesh generation

1 Introduction

Fast and accurate reconstruction of imaging-derived geometries and subsequent quality mesh generation for biomedical computation are enabling technologies for both clinical and research simulations. Often, the transformation of image to mesh is the rate-limiting step, requiring arduous manual manipulations. A challenging part of this process is the introduction of computable, orthogonal boundary patches, namely the *outlets*, into treed structures, such as vasculature, arterial or airway trees. The identification of the outlets are important for imposing proper outlet boundary conditions for accurate simulations (see e.g. [9]). Traditionally, this process is accomplished interactively using some surface mesh editor. With recent advances in image acquisition and processing, it is now common to resolve tens of thousands of dense

branches in these geometries; thus, manual manipulation has become impractical, if not impossible.

The problem of introducing outlets into these geometries is related to the computation of the curve skeleton of the object. This latter problem has received much attention in the biomedical literature [23,19,25,17,20,15,3,6]. In contrast, few studies have addressed the former [21].

We propose an efficient and robust approach to identifying and truncating the outlets for complex biomedical geometries. Our approach is based on a conceptual decomposition of objects into three different types of local structures, namely tips, segments, and junctions. The tips determine the outlets. We define the tips by introducing a novel geometric concept called the *average interior center of curvature (AICC)*, and identify those that are physically stable and numerically noise resistant through an inflation and deflation process. We compute well-defined orthogonal planes, which truncate the tips into outlets. The rims of the outlets are then connected into curves, and the outlets are closed using Delaunay triangulation with the curves as the boundary constraints. Compared to existing methods, our method does not require explicitly computing the skeleton, although the decomposition used in our method is closely related to the skeleton. As a result, our method is computationally more efficient, with near linear time complexity. We illustrate the effectiveness and robustness of our approach with a variety of complex lung and coronary artery geometries.

2 Methods

The essence of our strategy to introduce proper orthogonal boundaries into a tree-like surface is to conceptually decompose the surface into *tips*, *segments*, and *junctions*, and then to use the tips to identify the outlets. We begin by defining a number of basic mathematical concepts that are the building blocks of our algorithm. Implementation of the overall algorithm is then presented in two parts: 1) identification of stable tips, and 2) truncation and adaption of outlets.

2.1 Mathematical Preliminaries

We consider volumetric objects embedded in $\mathbb{R}^3$. The *surface* is the boundary of an object, which we assume to be given by a triangulation. By convention, we consider surface normals to point outwards, such that the curvature is positive if the object is locally concave and is negative if it is locally convex.

Local Dimensionality of an Object—Given an object Ω embedded in $\mathbb{R}^3$, let $\mathbf{p}$ be a point in Ω . The surface of Ω , denoted by Γ , is a 2-manifold without boundary. If we grow the neighborhood of $\mathbf{p} \in \Omega$ to some finite size that is greater than the local diameter of the object, then this expanded neighborhood of $\mathbf{p}$ may have different topologies depending on its location in Ω , in that it may appear to be more like a tube, a disk, or a ball.

Mathematically, let $N(\mathbf{p})$ denote the neighborhood of $\mathbf{p}$ in Ω , and $\mathbf{c}$ denote its centroid. Consider the 3×3 covariance matrix

$$\mathbf{M} = \int_{N(\mathbf{p})} (\mathbf{x} - \mathbf{c})(\mathbf{x} - \mathbf{c})^T d\mathbf{u}. \quad (1)$$

In general, $\mathbf{M}$ is symmetric and positive semi-definite (i.e., its eigenvalues are non-negative). The *local dimensionality* of the neighborhood, denoted by d , is the number of relatively large eigenvalues of $\mathbf{M}$, i.e., $\lambda_1 \approx \dots \approx \lambda_d \gg \lambda_{d+1}$. We also refer to d as the *numerical rank* of $\mathbf{M}$.

2.1.1 Object Decomposition—Given the definition of local dimensionality above, we may now formally state that the tree-like structures we wish to truncate are mostly comprised of 1-

D, tube-like neighborhoods. Those neighborhoods, are conceptually bound by two well-defined cutting planes that are locally orthogonal or nearly orthogonal to the surface. Furthermore, by the eigen-value decomposition of those bounding neighborhoods, each cutting plane should be perpendicular to the eigenvector of $\mathbf{M}$ corresponding to λ_1 . In other neighborhoods, where the local dimension is not 1-D, the orientation of such a cutting plane is not well-defined.

We refer to the cross-section of a cutting plane with the object as a *cut*. We conceptually decompose a mostly 1-D object into three classes of local structures: a *segment* is a continuous point set bound by two non-overlapping, well-defined cuts; *junctions* are bound by more than two well-defined cuts; and *tips* are bound by a single well-defined cut. Note that the tips define the outlets, and therefore they will be our focus in this paper.

2.1.2 Geometric Measures and Topological Invariance—The definitions above are based on semi-quantitative arguments of dimensionality and orthogonality for smooth surfaces. For discrete, noisy surfaces, both of these concepts may be ambiguous. To obtain a robust algorithm, we need additional quantitative measures to help resolve such ambiguities, so that we can classify the tips more accurately. We introduce a concept, called the *interior center of curvature (ICC)* at a point on a surface, which is based on an extension of the well-known concept of *center of curvature* for curves in differential geometry. To improve noise resistance, we further extend this concept and introduce an *average interior center of curvature (AICC)* of a surface patch. In addition, we utilize the topological invariance, known as the *Euler characteristic*.

Interior Curvature and Center of Curvature: Given a curve γ in $\mathbb{R}^2$, the center of curvature corresponding to a point $\mathbf{p} \in \gamma$ is at $\mathbf{c} = \mathbf{p} + (1/\kappa_p)\hat{\mathbf{n}}_p$, where $\hat{\mathbf{n}}_p$ and κ_p are the unit normal and curvature at $\mathbf{p}$. The center of curvature does not directly apply to surfaces, because a surface in general has different curvatures in different directions at a point. We define the *interior curvature (IC)* to be the most negative curvature at a point, and define the reciprocal of the absolute value of IC to be the *interior radius of curvature (IRC)*. We define the point on the inward surface normal of $\mathbf{p}$ with a distance equal to IRC as the *interior center of curvature (ICC)*. More precisely, given a point $\mathbf{p}$ on a surface Γ , let $\hat{\mathbf{n}}_p$ denote the unit outward surface normal to Γ at $\mathbf{p}$, κ_{p1} and κ_{p2} denote the principal curvatures at $\mathbf{p}$. Then the *interior curvature*, denoted by κ_p^- , is the more negative value of κ_{p1} and κ_{p2} , i.e.,

$$\kappa_p^- = \begin{cases} \kappa_{p1} & \kappa_{p1} < 0 \text{ and } \kappa_{p1} \leq \kappa_{p2} \\ \kappa_{p2} & \kappa_{p2} < 0 \text{ and } \kappa_{p2} < \kappa_{p1} \\ -\varepsilon & \kappa_{p1} \geq 0 \text{ and } \kappa_{p2} \geq 0, \end{cases} \quad (2)$$

where ε is an infinitesimal positive number (e.g., 10^{-100} for double-precision floating-point arithmetic) and is introduced to avoid division by zero. The *interior center of curvature (ICC)* at $\mathbf{p}$ is then

$$\mathbf{c}_c = \mathbf{p} + \frac{1}{\kappa_p^-} \hat{\mathbf{n}}_p. \quad (3)$$

The ICC is the center of the *osculating sphere*, which is tangent to the surface at $\mathbf{p}$ and has a radius equal to the IRC. For the canonical shapes of a sphere, cylinder, and torus, the interior centers of curvature coincide with their medial axis.

Average Interior Center of Curvature: The curvatures are second-order differential quantities, so the ICC computed based on point-wise curvatures are inherently sensitive to noise. To improve noise resistance, we define the *average interior of center of curvature* (AICC) of a surface patch sampled by a discrete point set $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m\}$ to be a weighted average of the ICCs at these points, i.e.,

$$\bar{\mathbf{c}} = \frac{\sum_{i=1}^m w_i \mathbf{c}_i}{\sum_{i=1}^m w_i}, \quad (4)$$

where $\mathbf{c}_i$ denote the ICC at point $\mathbf{p}_i$. We choose the weight w_i to be $(\kappa_i^-)^2$ times a control area a_i of the vertex i , where the control area is chosen to be one third of the sum of the areas of the incident triangles of $\mathbf{p}_i$. This weighting scheme gives higher priorities to more curved areas and lower priorities to flatter areas. Therefore, it tends to overestimate the curvature and underestimate the average radius of curvature. It in turn stabilizes the AICC and at the same time captures small features. For better robustness, we sometimes set the weights to zeros for the points that are “invisible” from a particular viewpoint, as we will explain in Section 2.2.2. AICC preserves the main features of ICC while delivers better noise resistance. In particular, the AICC of a sphere is exactly the center of the sphere, and the AICC of a small segment of a cylinder or torus is close to the medial axis of the object.

Orthogonality Revisited: The decomposition of the surface into segments, junctions and tips is reliant on a stable definition of cutting planes that are mostly orthogonal to the surface. A plane is uniquely determined by a point and a normal direction. The AICC of local neighborhoods provides a stable point. However, additional measures are needed to stably determine the orientation of the most orthogonal plane. We evaluate two complimentary measures of orthogonality: an *angle-based* measure and an *eigenvalue-based* measure.

The angle-based measure is quite intuitive. Given a plane P , let $\hat{\mathbf{n}}_p$ denote the unit normal to P , and $\hat{\mathbf{n}}_\tau$ denote the unit face normal of a triangle τ . The dihedral angle between P and τ is $\arccos |\hat{\mathbf{n}}_p^T \hat{\mathbf{n}}_\tau|$, which ideally should be 90° but can be as small as 0° . We therefore define this measure based on the minimum dihedral angle of P with all of the connected triangles that intersect P .

This angle-based measure is useful for determining the quality of a given plane, but it does not facilitate the adjustment of the plane to improve orthogonality. Thus, we introduce a complimentary eigenvalue-based measure. Consider the matrix

$$\mathbf{N} = \sum_{\tau} w_{\tau} \hat{\mathbf{n}}_{\tau} \hat{\mathbf{n}}_{\tau}^T, \quad (5)$$

where τ denotes a triangle that is intersected by the plane, $\hat{\mathbf{n}}_{\tau}$ denotes its unit normal to τ , and w_{τ} is the length of the line-segment of the intersection of the plane with τ . We refer to $\mathbf{N}$ as the *normal covariance matrix*. Note that $\mathbf{N}$ shares some similarity with the covariance matrix in (1) but it behaves differently. If the plane P intersects a cylinder nearly orthogonally, then all the normals $\hat{\mathbf{n}}_{\tau}$ are coplanar, and $\mathbf{N}$ should be a rank-two matrix. Let σ_2 and σ_3 denote the two smaller eigenvalues of $\mathbf{N}$; $\rho = \sigma_3/\sigma_2$ is then a measure of how close the surface is to a cylinder locally. In general, ρ is between 0 and 1, and the smaller the better. If ρ is sufficiently small, then the eigenvector of $\mathbf{N}$ corresponding to σ_3 provides a more accurate normal for the plane.

Minimum Containing Sphere and Euler Characteristic: Consider a sphere centered at the ICC with radius equal to the interior radius of curvature (IRC), i.e., the osculating sphere at a point. This sphere is tangent to the surface at the point. If the object is a cylinder or a torus, the osculating sphere intersects the object at a circle. If the object is a sphere, then the osculating sphere intersects the object at the whole sphere. However, these intersections are unstable topologically in the presence of numerical errors. To overcome this instability, we enlarge the osculating sphere by a small factor, then the enlarged sphere would intersect the object at a surface patch P , which would be topologically more stable. We refer to this grown sphere as the *minimum containing sphere* (MCS) of P .

Useful information about a local structure can be obtained by analyzing the topology of P . The most important topological quantity of a surface is the *Euler characteristic*. Given a polygonal surface (such as a triangulated surface), the *Euler characteristic* is the alternating sum:

$$\chi \equiv \# \text{vertices} - \# \text{edges} + \# \text{faces}. \quad (6)$$

For a tip, a segment, or a junction, its patch on Γ is an open surface. The Euler characteristic is 1 for the surface of a tip, 0 for a segment, and a negative number for a junction. Therefore, it gives a convenient means for classifying segment, tip, or junction. In the next section, we present an algorithm for decomposing an object using the AICC, orthogonality, and Euler characteristic.

2.2 Algorithm for Identifying Outlets

Herein we present our algorithm for identifying the tips of the surface, while paying special attention to stability and efficiency. At a high-level, this algorithm identifies the tips by going through an inflation-deflation process to locate their adjacent cuts that satisfy some orthogonality conditions. The algorithm takes a surface mesh as input, specified simply as a list of vertex coordinates and the element connectivity of the triangles. We split the algorithm into three main steps, as detailed in the following three subsections. First, we compute the AICC for the entire surface. From these potential plane centers, we select a subset as seeds for growing spheres, and thus identifying the stable cuts of the surface. Lastly, we adjust the position and orientation of the cuts at the stable tips.

2.2.1 Computing AICCs—Identification of the position of the candidate cuts is based on AICC. Thus, we begin by computing the geometric differential quantities of the surface mesh, including surface normals and principal curvatures at the vertices. The surfaces in our applications are inherently noisy, as they are typically derived from medical images. For stability and noise resistance, we apply a quadratic fitting over the two-ring neighborhood of each vertex and solve the fitting using the weighted least squares framework in [11]. From the surface normal and principal curvatures, we obtain the interior curvature (IC) and the interior center of curvature (ICC) using Eq. (2) and Eq. (3), and then compute the AICC of the two-ring neighbor of each vertex using Eq. (4) for improved noise resistance. To support these computations, we construct a half-edge data structure [5, §2.2], and in particular an array-based version in [2]. This data structure will also be used in the later steps of our algorithm.

2.2.2 Identifying Tips—To identify the local structures of the surface, and in particular the tips of the surface, we first select a subset of the mesh vertices based on the AICC. At each of these seed vertices, we iteratively grow a surface patch starting from the seed's two-ring neighborhood until the patch has reached a steady state topologically and geometrically. For better robustness and efficiency, we choose a proper ordering of the seed vertices and filter out duplicate seeds during the growing process. The algorithm has four key aspects, including the

selection of the seed vertices, the growth of spheres, adjustment of orientation of the cut, and the stopping criteria of growth, which we describe in more detail as follows.

Selection and Ordering of Seed Vertices: We first choose the seed vertices as those vertices whose principal curvatures are both negative. To reduce the number of candidate seeds, we further impose an upper bound on the IRC of a seed vertex. In practice, it suffices to use a global radius of the object as the upper bound, approximated efficiently using half of the minimum dimension of the bounding box of the connected component of the object that contains the vertex.

To avoid missing smaller tips, it is important to order the seed vertices properly. With an arbitrary ordering, some larger structures might “swallow” smaller structures and in turn cause some small but important tips to be missed. Thus, seed vertices are sorted in ascending order of IRC, which approximates the local radius of the structure. This ensures that smaller features are processed first. We omit a seed vertex that falls within a surface patch that was grown from another seed.

Growth of Minimum Containing Sphere: Starting from a seed vertex, we grow a MCS into a local structure that is stable in terms of the persistency of the topology of the structure. For each seed vertex, we start that growth from the MCS from its two-ring neighbors, with the goal of growing it to a sphere that fully contains a tip. The MCS is grown iteratively by alternating between computing the AICC and computing the intersection of a sphere with the surface. Specifically, from the initial MCS, we compute its intersection with the surface that is connected with the seed vertex. We then compute the AICC and MCS of this new surface patch, and continue until the stopping criteria are reached, as we will describe shortly.

To guard against interference of nonconvex surfaces, we introduce a “visibility condition” during the computation of AICC to consider only those vertices for which the previous AICC was on the inside of their tangent planes. More specifically, let $\mathbf{c}$ denote the AICC of the previous MCS. Given a vertex $\mathbf{p}$ in the surface patch contained in the previous MCS, let $\hat{\mathbf{n}}$ denote the outward unit normal at $\mathbf{p}$. We set the weight for $\mathbf{p}$ to zero in (4) if $\hat{\mathbf{n}}^T(\mathbf{c} - \mathbf{p}) \geq 0$.

Adjustment of the Cut Orientation: Tips are defined by their bounding cuts. We obtain initial positions of the cuts based on the MCS and iteratively refine their positions and orientations. To define a cut from a MCS, we require that the cut passes through the AICC of its surface patch and is nearly orthogonal to the surface.

First, we obtain an initial orientation based on a principle component analysis of the intersection of the sphere with the surface. Let γ denote the intersection curve of the sphere and the surface; let $\mathbf{p}$ denote a vertex on the curve; let $\hat{\mathbf{n}}$ denote the unit normal to a vertex; let $\mathbf{N}$ denote the normal covariance matrix defined in (5). We obtain the initial normal to the cutting plane as the basis vector of the null space of $\mathbf{N}$, if the two smaller eigenvalues of $\mathbf{N}$ are well separated (in practice, we require $\rho = \sigma_3/\sigma_2 \leq 0.6$). For the cut to be valid, the whole curve γ must be on the same side of the cutting plane. We choose the sign of the normal such that $\int_{\gamma} \hat{\mathbf{n}}^T(\mathbf{p} - \mathbf{c})ds > 0$, so that γ must be above the cutting plane for the cutting plane to be valid.

Starting from a plane, let $\tilde{\gamma}$ be the intersection curve between the current plane with the surface Γ . Let $\hat{\mathbf{n}}$ denote the basis vector of the null space of the surface normal on $\tilde{\gamma}$, and let $\tilde{\mathbf{P}} = (\mathbf{c}, \hat{\mathbf{n}})$ denote a new plane. If γ is completely above $\tilde{\mathbf{P}}$, and the minimum angle between $\tilde{\mathbf{P}}$ and the normals are greater than some tolerance and is greater than that of $\mathbf{P}$, we then take $\tilde{\mathbf{P}}$ as the new plane. We repeat this process for up to four iterations. This tolerance is a free-parameter. In practice, we found that approximately 50° works well for a variety of smooth and noisy geometries.

Stopping Criteria: In general, we would like the growth of the MCS to stop if it has reached some *steady state* geometrically and topologically. The concept of steady state is closely related to the stability of the algorithm, in the sense that a small change to the geometry should not affect the qualitative behavior. Geometrically, we measure the stability based on three criteria:

- 1 the dimensionality of the intersection curve, γ , based on an eigenvalue analysis
- 2 the orthogonality of the intersection curve with the surface
- 3 the change of the radius of the sphere

These criteria are relatively easy to understand. The first condition gives a quantitative measure of whether the intersection curve is nearly planar. The second condition checks the maximum angle between an approximating plane of the curve with the surface. The third condition checks whether the growth of sphere has stopped. Topologically, we consider one additional criterion:

- 4 the persistency of Euler characteristic

As explained in Section 2.1.2, the Euler characteristic is a topological invariance and is easy to compute in principle. However, a MCS may cut out a spike in the mesh caused by the erroneous position of a single vertex, or accidentally cut out a small bump due to a small noisy neighborhood. To address the first issue, we check the holes in the surface patch and fill in those that correspond to a single face, two incident faces of an edge, or the one-ring neighborhood of a vertex. To resolve the second issue and further enhance robustness, we consider the Euler characteristic to be trustworthy only if it remains unchanged for at least two consecutive growth iterations of the MCS.

2.2.3 Fine-Tuning Cuts at Tips—The previous steps identify the stable local structures of tips. However, the minimum containing spheres might have been over-inflated. After determining an initial cut, we make additional adjustment to the positions of the cut that bounds the tip. Our objective in this step is to move the cut as close to the tip as possible. In particular, given the current plane, if the AICC of the surface patch is below the plane, we move the plane to pass through the AICC, and adjust its orientation. After adjusting the cuts, we then label the triangles that intersect or are below the cutting plane within the MCS by a unique integer ID for the tip. These cutting planes and labeled triangles will be used in truncating the outlets, described in Section 2.3.

2.3 Truncation and Mesh Generation of Outlets

The result of the foregoing analysis is the identification of stable tips and their associated orthogonal planes. In order to truncate the tips and adapt the border triangles to produce a smooth, planar curve that can be closed by triangulation, we first compute the triangle chains intersected by the cutting plane for each tip. The triangles that are in the tip but do not intersect the plane are removed. We traverse through the triangle chain from neighbor to neighbor, so that the triangles are sorted in some consistent order, either clockwise or counterclockwise. We then carry out the truncation and rim adaption operations simultaneously.

After the tips have been truncated, the vertices of an ordered triangle chain $T_j^{\text{start} \dots \text{end}}$ will be either above or below the cutting plane P_j , where above and below are defined according to the inward-pointing normal to P_j . The strategy for planarizing the open edges of $T_j^{\text{start} \dots \text{end}}$ consists of: 1) moving the vertices that are below the plane up to the plane along triangle edges; and 2) splitting the vertices to create two triangles where needed. Figure 1 illustrates the basic algorithm and notation. Triangles with two vertices below the plane are referred to as *2-start* triangles; those with only one vertex below the plane are referred to as *1-start* triangles.

Typically, the vertices of 2-start triangles are moved to new positions along the edges transected by plane P_j (dotted line in the figure). With 1-start triangles, the vertex below the plane is typically split to create a new vertex (asterisk in the figure), and thus a new triangle. Therefore, 2-start triangles spawn one triangle and 1-start triangles spawn two triangles. The new vertex is called the *child*, which is split from the parent. Vertices above the plane are denoted as *anchor* and *corner*, according to the traveling direction of the walking algorithm, the corner being on the leading edge.

Obviously, the specific moves for a given triangle are dependent on whether the preceding triangle was a 1-start or a 2-start triangle. Once all of the open edges are planarized, they are closed with constrained Delaunay triangulation by calling Shewchuk's 2D Delaunay code Triangle [22]. After generating consistent surface triangulations, we optionally refine and optimize surface meshes globally and generate volume meshes that are suitable for numerical simulations.

3 Results

We report some results with our proposed method for imaging data sets. We omit any presentation of our segmentation or isosurface extraction methods, except to say that a fuzzy-connected threshold approach was used for segmentation and a variant of the well-known Marching Cubes algorithm [14,18] was used for isosurface extraction. Where isosurfaces were smoothed, we optionally applied a volume-conserving smoothing algorithm [12]. This is simply a Laplacian smoothing, with a corrector step to conserve volume. In practice, this corrector step is very useful for working with voxelated data since the corrector step preserves volume and reduces the possibility of self-intersections.

3.1 Mouse Coronary Arteries

We first apply our algorithm to a segmented coronary artery dataset, obtained from the computed tomography (CT) data of a mouse coronary courtesy of Dr. Ghassan Kassab. The initial surface mesh had 49,881 vertices and 99,758 triangles. Figure 2 shows the results of our algorithm. Our algorithm identified 99 stable tips (Panel A). Panels B & C show the stable tips and truncated outlets, respectively, for the region near the middle of Panel A. Visual inspection revealed that our method delivered 100% accuracy, compared to the results of a manual identification. No outlets were missed, nor were any spurious outlets introduced. Figure 3 shows a section of a volume mesh, generated from the truncated surface mesh. The methods for generating this volume mesh are described elsewhere [13,7].

3.2 Monkey Lung Airways

The monkey lung cast data is courtesy Dr. Kevin Minard [16]. Figure 4 shows the unprocessed monkey lung surface mesh (Subpanel A), which consists of 895,915 vertices and 1,791,826 triangles. Our algorithm identified 6,802 outlets (sub panels B–D).

To measure the accuracy for this large datasets, visual inspection of all the thousands of outlets was impossible. Thus, we performed a statistical sampling. Four sets of random x, y, z coordinates were generated. These coordinates each defined the center of $50 \times 50 \times 50$ voxel sampling zones in the volumetric dataset. For each of these sampling zones, we first employed a surface editor to identify and truncate the outlets manually. Independently, we ran our algorithm to identify and generate the outlets automatically for the same sampling zones. Finally, an independent judge visually inspected both the manually and automatically identified outlets and voted for the outlets that are most trustworthy. We label these outlets as *true* and the others as *false*. The total number of true outlets in the sampling zones was 98. The automated approach identified 92 of these outlets, indicating an accuracy of $93.9 \pm 2.4\%$. Accuracy metrics

of sensitivity and precision were also calculated and reported in Table 1. Sensitivity was defined as the number of true outlets found divided by the total number of true outlets. Precision was defined as the number of true outlets found divided by the total number of outlets produced by a process [1,24].

3.3 Smooth and Noisy Surfaces of Rat Lungs

The rat lung cast data is also courtesy Dr. Kevin Minard [16]. To demonstrate the robustness of our method, we illustrate the application of our algorithms to a truncated version of the rat lung cast at five different levels of surface adaption (see [13]). At one of these levels (*rat1*), we compared the results of a smoothed and a noisy surface, where the smoothed one was applied the conservative surface smoothing while the noisy was not. In other words, the unsmoothed version is simply the triangulation of boundary voxels on a Cartesian mesh. Finally, we applied our algorithms to the full rat lung cast data set (see Table 2).

Figure 5 (A & B) shows the unsmoothed surface and smoothed surface, respectively. Panels A & C of the same figure are smoothed data before and after truncation, respectively. Panels B & D are unsmoothed data before and after truncation. In both cases, the surface mesh consists of 139,080 vertices and 278,156 triangles. For both datasets, our algorithm identified 255 stable tips. The numbers of vertices, triangles, outlets and execution times for all cases is reported in Table 2.

3.4 Comparison of Efficiency

Computational efficiency is a critical consideration for large datasets. We report the execution times of our method in Table 2. Our method was implemented in MATLAB and then converted into C using the Agility MCS software (www.agilityds.com). We compiled the resulting C codes using gcc 4.2.4 with optimization enabled, and performed our tests on a Linux computer with a 3GHz Intel Duo Core Pentium 4 processor and 2GB of RAM. For the largest mesh with about 1.8 million triangles, our method required about 5 seconds.

To get a better understanding of the performance of our algorithm, we compared our method to an available skeleton-based method for three meshes. Other published methods for computing the skeleton were not compared since they were not available to us. Specifically, we computed the skeleton using the method of Dey and Sun [6], which is competitive with, if not better than, most other existing methods. As can be seen from the table, the skeleton-based method was more than 10,000 times slower than our method for moderate sized meshes. In addition, the curve-skeleton code never terminated for the monkey pulmonary geometry and failed abruptly for the noisy rat pulmonary tree.

4 Discussion

We presented a new method for identifying and truncating the boundary outlets for complex tree-like biological geometries. In general, there have two types of approaches for automatic identification of the outlets for surfaces extracted from medical images. In the first approach, one identifies the outlets from the voxels of the medial images, and extracts the surface mesh also from the voxels using marching cubes. Such an approach is taken by Pennecot et al. in [21], who identified the outlets by first constructing the curve skeleton of the objects using the algorithm in [26], where the curve skeleton is a one-dimensional representation to the volumetric object. A major problem of this type of approach is that the constructed skeleton and the surface mesh are not guaranteed to be consistent, because they are obtained independently of each other using two separate algorithms. This inconsistency can be especially pronounced if the surface mesh must undergo topological surgeries, as is often required for complex geometries. In addition, the accuracies of the skeletons extracted from

the voxels are limited by the voxel sizes, and the identified outlets are not sufficiently orthogonal to the surface and may require additional post-processing.

The second type of approach is to extract the outlets from a triangulated surface, which may be extracted using marching cubes and may have undergone additional mesh manipulation. This is a preferred approach in practice, as it avoids the aforementioned inconsistencies. In principle, one may compute the skeleton from the surface mesh using an algorithm surveyed in [4], and then use the skeleton to identify and truncate the outlets in the surface mesh. However, the construction of curve skeletons from the surface meshes is far less mature than their constructions from voxels, and they are algorithmically more complex and often very expensive. In addition, constructing the whole skeleton may be overkill in some cases, because identifying the outlets often requires only the end-points (i.e., tips) of the skeleton instead of the internal branches, especially if one wants to preserve the complete tree structure in the simulations.

In this paper we introduced a third way based on the triangulated surface, which is related to the curve skeleton but does not require explicit construction of it. Instead, we locate the outlets by identifying the stable tips in the surface, which would correspond to the tips of a stable skeleton. We introduced a novel concept called the average interior center of curvature to aid the identification of these tips. We then computed well-defined orthogonal planes and used them to truncate the tips into outlets.

The description of our algorithm assumes the surface is smooth. This assumption is for conceptual clarity because of the use of curvatures, which are meaningless quantitatively at sharp features of a surface. However, our algorithm uses curvatures for computing starting points for our iterative approach, and the other measures used in our algorithm do not depend on smoothness of a surface. For example, our algorithm can reliably process surface meshes that have been partially processed manually with some outlets and sharp features introduced.

Our method is robust and noise resistant. This was demonstrated on several large and complex smooth geometries as well as raw, unsmoothed surface meshes from biomedical images. For the mouse coronary geometry (Figure 2), visual inspection revealed that our method delivered 100% accuracy compared to the results of a manual process, without any missing an outlet or introducing extraneous outlets. The total execution time, for this surface was 0.22 seconds (Table 2), illustrating the efficiency of the approach. For this geometry, we further demonstrated that the resultant surface was suitable for volume mesh generation, by constructing a CFD-ready, layered-tetrahedral mesh [13] from the truncated, and labeled surface (Figure 3).

We next applied the method to a dense, large scale monkey lung surface (Figure 4), consisting of nearly two million triangles, with the result that nearly seven thousand outlets were introduced and labeled in about five seconds (Table 2). Together with the results from other geometries (Table 2), it is clear that the execution times of our algorithm depend both on the size of the surface meshes, as well as on the number of outlets, and that the time complexity of the overall algorithm is roughly linear, making it highly competitive for automated mesh generation in both research and clinical settings. Visual inspection of the complete monkey lung dataset was impossible. However, we did perform a statistical sampling (Table 1) of a subregion of the mesh and found that our method is comparable with manual identification while being vastly more efficient.

To further assess the robustness of our method, we compared the results of the application of our algorithm to both an unsmoothed and a smoothed a rat lung cast isosurface (Figure 5), following the application of Marching Cubes. Since the unsmoothed isosurface is simply the triangulation of boundary voxels on a Cartesian mesh, it represents the extreme of noise (Figure

5 B&D). Nevertheless, our method was able to identify the same number of outlets in the two geometries. This result demonstrates the remarkable stability and noise resistance of our method. While it is necessary to caution that the algorithm is not guaranteed to work on all voxelated isosurfaces, no computational mesh ever displays the level of noise present in a voxelated isosurface. Therefore, we can assert with a high level of confidence that our method will work robustly and accurately for the vast majority of surface meshes destined for computational analysis.

Since we focus on surfaces rather than voxelated volumes, we contrast our approach with approaches for obtaining skeletons from surfaces. Few robust alternatives exist. One of these is publicly available [6] and was contrasted with our method. Table 3 shows a comparison of the execution times of our method against this method for three triangulated surfaces. We found that the skeleton-based method was more than 10,000 times slower than our method for moderate-sized meshes. This is not surprising because the skeleton-based algorithm has very high time complexities, in contrast to our nearly linear time complexity. Worse, the curve-skeleton based code never terminated for the monkey pulmonary geometry and failed abruptly for the noisy rat pulmonary tree. For the application of introducing boundary outlets into an imaging derived surface mesh, we conclude that our method delivers a better combination of efficiency, robustness, noise resistance and accuracy compared to existing alternatives that we are aware of.

As a future research direction, we plan to employ the truncated mesh from this work in biomedical applications, to better understand pulmonary and coronary flows, aerosol deposition, and fluid-structure interactions. A limitation of our current method is that the information about the branch generations cannot be inferred from the tips. However, our decomposition of the object into tips, segments, and junctions can in fact be used as a preprocessing step for efficient computing the curve skeletons of surface meshes. We plan to extend our method to obtain efficient and robust algorithms for computing the curve skeletons to enhance the generality of our method. In addition, the computed curve skeletons will also benefit other biomedical engineering applications, such as the definition of multi-scale boundary conditions [8,10] and morphometry [8,25,19,23].

Acknowledgements

Research was supported by the National Heart and Blood Institute Award 1R01HL073598-01A1; by the National Institute of Environmental Health Sciences Award P01 ES011617 and by the National Science Foundation award DMS-0809285. The authors would also like to acknowledge Dr. Ghassan Kassab for graciously providing the coronary CT data, and Dr. Kevin Minard for the lung MR data.

References

1. Altman DG, Bland JM. Diagnostic tests. 1: Sensitivity and specificity. *BMJ* 1994;308:6943.
2. Alumbaugh, T.; Jiao, X. Compact array-based mesh data structures. *Proc. 14th Int. Meshing Roundtable*; 2005. p. 485-504.
3. Cornea N, Silver D, Yuan X, Balasubramanian R. Computing hierarchical curve-skeletons of 3d objects. *Vis Comput (Germany)* 2005;21(11):945–955.
4. Cornea ND, Silver D, Min P. Curve-skeleton properties, applications, and algorithms. *IEEE Trans Vis Comput Graph* 2007;13:530–548. [PubMed: 17356219]
5. de Berg, M.; Cheong, O.; van Kreveld, M.; Overmars, M. Vol. 3rd edn.. Springer; 2008. *Computational Geometry: Algorithms and Applications*.
6. Dey, T.; Sun, J. Defining and computing curve-skeletons with medial geodesic function. *Proc. Eurographics Symp. Geometry Proc.*; 2006. p. 143-152.
7. Dyedov V, Einstein DR, Jiao X, Kuprat AP, Carson JP, del Pin F. Variational generation of prismatic boundary-layer meshes for biomedical computing. *Int J Numer Meth Engrg*. 2009in press

8. Einstein DR, Neradilak B, Pollisar N, Minard KR, Wallis C, Fanucchi M, Carson JP, Kuprat, Kabilan S, Jacob RE, Corley RA. An automated self-similarity analysis of the pulmonary tree of the sprague-dawley rat. *Anat Rec* 2008;291:1628–1648.
9. Grinberg L, Karniadakis GE. Outflow boundary conditions for arterial networks with multiple outlets. *Annals of Biomedical Engineering* 2008;36:1496–1514. [PubMed: 18612828]
10. Huo Y, Kassab GS. A hybrid one-dimensional/Womersley model of pulsatile blood flow in the entire coronary arterial tree. *Am J Physiol Heart Circ Physiol* 2007;292(6):H2623–H2633. [PubMed: 17208998]
11. Jiao, X.; Zha, H. Consistent computation of first-and second-order differential quantities for surface meshes. *Prof. ACM Solid and Physical Modeling Symposium*; 2008. p. 159-170.
12. Kuprat A, Khamayseh A, George D, Larkey L. Volume conserving smoothing for piecewise linear curves, surfaces, and triple lines. *J Comput Phys* 2001;172:99–118.
13. Kuprat AP, Einstein DR. An anisotropic scale-invariant unstructured mesh generator suitable for volumetric imaging data. *J Comput Phys* 2009;228:619–640.
14. Lorensen, W.; Cline, H. Marching cubes: A high resolution 3D surface construction algorithm. *Proc. SIGGRAPH* 87; 1987. p. 163-169.
15. Maddah M, Soltanian-Zadeh H, Afzali-Kusha A. Snake modeling and distance transform approach to vascular centerline extraction and quantification. *Comput Med Imaging Graph (UK)* 2003;27(6): 503–512.
16. Minard KR, Einstein DR, Jacob RE, Kabilan S, Kuprat AP, Timchalk CA, Trease LL, Corley RA. Application of magnetic resonance (MR) imaging for the development and validation of computational fluid dynamic (CFD) models of the rat respiratory system. *Inhal Toxicol* 2006;18(10): 787–794. [PubMed: 16774868]
17. Molthen RC, Karau KL, A DC. Quantitative models of the rat pulmonary arterial tree morphometry applied to hypoxia-induced arterial remodeling. *J Appl Physiol* 2004;97(6):2372–2384. [PubMed: 15333611]
18. Newman T, Yi H. A survey of the marching cubes algorithm. *Computers & Graphics* 2006;30:854–879.
19. Nordsletten DA, Blackett S, Bentley MD, Ritman EL, Smith NP. Structural morphology of renal vasculature. *Am J Physiol Heart Circ Physiol* 2006;291(1):H296–H309. [PubMed: 16399870]
20. Palagyi K, Kuba A. A parallel 3d 12-subiteration thinning algorithm. *Graph Models Image Process (USA)* 1999;61(4):199–221.
21. Pennecot J, Krenkel L, Wagner C. Lung automatic reconstruction algorithm for CFD simulations. *GiD 2008 4th Conference on Advances and Applications of GiD*. 2008
22. Shewchuk JR. Triangle: Engineering a 2d quality mesh generator and Delaunay triangulator. *Lecture Notes in Computer Science* 1996;vol 1148:203–222.
23. Spaan JA, ter Wee R, van Teeffelen JW, Streekstra G, Siebes M, Kolyva C, Vink H, Fokkema DS, VanBavel E. CT, MRI and cryomicrotome: Visualisation of intramural coronary vasculature by an imaging cryomicrotome suggests compartmentalisation of myocardial perfusion areas. *Med Biol Eng Comput* 2005;43(4):431–435. [PubMed: 16255423]
24. Taylor R. *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*. University Science Books. 1999
25. Wischgoll T, Choy JS, Ritman EL, Kassab GS. Validation of image-based method for extraction of coronary morphometry. *Ann Biomed Eng* 2008;36(3):356–368. [PubMed: 18228141]
26. Zhou Y, Toga AW. Efficient skeletonization of volumetric objects. *IEEE Trans Vis Comput Graph* 1999;5:196–209.

Fig. 1.

Notation for fast walking adaption algorithm. In general vertices belonging to *2-start* triangles (blue) are moved (dark circles) and vertices for *1-start* triangles (red) are created (dark asterisk). The specific reconnection depends on whether the previous triangle in the triangle chain was a *1-* or a *2-start* triangle. *Anchor* and *corner* nodes are only labeled for Triangle 3. *Child* nodes are nodes that are created from previously moved vertices. The *startnode* (red) is the trailing node with respect to the direction of travel of the first triangle of the chain. In the case where a *1-start* triangle is the first triangle in the chain, the startnode is created.

Fig. 2.
Example of coronary arteries.

Fig. 3.

Elaboration of closed surface mesh. Truncation produces valid triangulations and optimally orthogonal planes (A). Those triangulations may then be adapted to the physics of the problem (B) in order to produce a quality volume mesh (C). Methods for B & C are described in [13, 7].

Fig. 4. Truncation of a monkey lung cast dataset (A). 6802 stable tips were identified and truncated to produce the same number of outlets (B–D). Panel C is a detail of panel B and panel D is a detail of panel C.

Fig. 5. Outlets identified on smoothed (A & C) and “noisy” (B & D) versions of the same dataset. The noisy data are simply the unsmoothed result of the application of Marching Cubes to the imaging data. In both cases 255 stable tips were identified and truncated.

Table 1

Sensitivity and precision of identified outlets.

	True	False	Missed	Sensitivity	Precision
Manual	95	4	3	.97	.96
Automated	92	0	6	.94	1

Table 2

Execution times of our method.

case	input mesh		output mesh		times (in seconds)	
	#triangles		#triangles	#outlets	finding tips	pruning
tee-junction	4302		5218	3	0.01	0.001
coronary	99758		102288	98	0.19	0.028
rat1 lung	278156		283498	255	0.62	0.074
rat2 lung	238940		243412	209	0.53	0.065
rat3 lung	272320		278666	258	0.6	0.073
rat4 lung	248740		253428	208	0.6	0.069
rat5 lung	243936		248784	222	0.54	0.066
monkey lung	1791826		1990514	6802	4.73	0.52
rat5 full	983496		1072594	3405	2.6	0.28
noisy rat 1 lung	278156		283826	255	0.69	0.13

Table 3

Comparison of execution times.

case	input mesh		times (in seconds)	
	#vertices	#triangles	ours	skeleton
tee-junction	2153	4302	0.01	12.42
coronary artery	49881	99758	0.22	1202.9
rat 2 lung	119472	238940	0.60	7509.25