Guaranteed Quality Tetrahedral Delaunay Meshing for Medical Images

Panagiotis A. Foteinos, Andrey N. Chernikov, and Nikos P. Chrisochoides
Department of Computer Science
College of William and Mary
Williamsburg, Virginia, USA, 23185
{*pfot,ancher,nikos*}*@cs.wm.edu*

***Abstract*—In this paper, we present a Delaunay refinement algorithm for meshing 3D medical images. We prove that (a) all the tetrahedra of the output mesh have radius-edge ratio less than 2, (b) all the boundary facets have planar angles larger than 30 degrees, (c) the symmetric (2-sided) Hausdorff distance between the object surface and mesh boundary is bounded from above by a user-specified parameter, and (d) the mesh boundary is ambient isotopic to the object surface. The first two guarantees assure that our algorithm removes most of the poorly shaped elements, making the mesh suitable for subsequent finite element analysis. The last two guarantees assure that the mesh boundary is a good geometrical and topological approximation of the object surface. Our long term goal is to develop a real time image-to-mesh conversion algorithm; towards that direction, our algorithm recovers the object surface and meshes the interior volume at the same time without sampling the object surface as a preprocessing step, unlike other Delaunay meshing techniques. Experimental evaluation of our algorithm on real medical data corroborates the theory.**

***Keywords*-Delaunay mesh generation; medical images; quality; fidelity;**

I. Introduction

Delaunay meshing is a popular technique for generating tetrahedral meshes, since it is amenable to rigorous mathematical analysis [1–5]; it has also been shown that Delaunay refinement behaves quite well in practice [6, 7].

In the literature, Delaunay refinement techniques have been employed to mesh objects whose surface consists of piecewise linear features and is, actually, given [1, 2, 5, 8]. In this paper, we deal with objects whose surface is a smooth 2-manifold and is not explicitly given (see Section II); it is the algorithm's responsibility to mesh the interior of the object such that the mesh boundary describes the object surface well.

The quality of an element is measured in terms of its *circumradius-to-shortest-edge* ratio or *radius-edge* ratio for short. An upper bound for the radius-edge ratio of all the elements in the output mesh would guarantee that all the poorly-shaped elements are removed from the mesh except for the so called *slivers*. Slivers can be removed by a post-processing method, e.g., the perturbation [9] or the exudation method [8].

Inspired by the popular Delaunay surface meshing algorithms described in [10, 11], Delaunay volume meshing algorithms offer quality and fidelity guarantees [3, 4] under the assumption that the surface of the object is smooth [3, 11] or does not form input angles less than $90°$ [4]. However, the quality achieved by these algorithms is somewhat weak: the upper bound for the elements' radius-edge ratio is larger than 4. On the contrary, the upper bound achieved by our algorithm is 2.

Moreover, Delaunay meshing algorithms, for example those described in [3] and [11], often start with an initial dense surface point set. The construction of such a set is a non-trivial task (see Boissonnat et al. [12] for more details) that underpins the total speed of the mesher. We, on the other hand, start with a *bounding box* (consisting of eight points), and then we proceed by meshing the interior volume and by sampling the object surface at the same time as needed, generating a surface point set in a simple, intuitive, and fast way (see Section III).

In the literature, there are also non-Delaunay surface and volume meshing algorithms for 3D images. *Marching Cubes* is a very popular technique for surface meshing [13]; it guarantees, however, neither good quality triangular facets nor good surface approximation. Among the non-Delaunay volume meshing techniques, we distinguish the *Red-Green Mesh* (RGM) [14] and the *Isosurface Stuffing* method [15]. Both of them start with a *body-centered cubic* lattice (BCC) and therefore output a large number of elements. Also, RGM does not give any quality or fidelity guarantees.

Our algorithm guarantees that the radius-edge ratio of all output tetrahedra is less than 2, the boundary facets have planar angles more than $30°$, the Hausdorff distance, both from the mesh boundary to the object surface and from the object surface to the mesh boundary, is bounded from above by a user-specified parameter, and that the mesh boundary is ambient isotopic to the object surface. The real surface of the biological object is smooth, i.e., does not contain sharp angles. To our knowledge, our algorithm is the first volume Delaunay mesher for smooth objects achieving such a small radius-edge ratio with these fidelity guarantees.

Another advantage of our algorithm is that it creates few big elements in the interior of the object and more elements on the surface where higher resolution is required.

In this paper, we focus on generating tetrahedral meshes for medical images. Such meshes are suitable for finite element simulations involved in non-rigid registration [16–18]. Our contribution can be summarized as follows:

- The elements of the resulting mesh are proved to have radius-edge ratio less than 2,
- the boundary facets have angles larger than 30°,
- the mesh boundary is proved to be a good topological and geometrical approximation of the object surface,
- our algorithm meshes the volume and samples the object surface at the same time, starting from an initial set of 8 points.

The rest of this paper is organized as follows: Section II provides the necessary definitions and Section III outlines our algorithm. Section IV proves the quality and Section V proves the fidelity guarantees. Finally, Section VI assesses the practical value of our work on both synthetic and real medical data and Section VII concludes our paper.

II. Preliminaries

Let $\mathcal{I} \subseteq \mathcal{R}^3$ be the (spatial) domain of a segmented image. $\mathcal{I}$ is the input of our algorithm. We assume that the following function f is available: $f : \mathcal{I} \rightarrow \{-1, 1, 0\}$, such that for every point $p \in \mathcal{I}$, $f(p) = 1$ if p lies inside the object $\Omega \subseteq \mathcal{I}$ to be meshed, $f(p) = -1$ if p lies outside, and $f(p) = 0$ if p lies on the surface of the object, denoted as $\partial\Omega$. The existence of such a function is a quite reasonable assumption: f can be constructed (or approximated from the image voxels quite well) for any segmented image.

As is generally the case in the literature [3, 11, 12], we also assume that $\partial\Omega$ is a smooth 2-manifold without boundary.

Definition 1. *The medial axis of $\partial\Omega$ is the closure of the set of those points having more than one closest point on $\partial\Omega$.*

Definition 2. *The local feature size of a point $p \in \partial\Omega$, denoted as lfs (p), is the distance from p to the medial axis of $\partial\Omega$.*

We denote with $\mathrm{lfs}_{\min}$ the minimum of the local feature sizes of all the points on $\partial\Omega$, that is: $\mathrm{lfs}_{\min} = \min\{\mathrm{lfs}\,(p) : p \in \partial\Omega\}$. Note that if $\partial\Omega$ does not contain sharp angles, then $\mathrm{lfs}_{\min}$ is bounded from below by a real positive constant.

Definition 3 (Amenta et al. [11]). *A point set $P \subset \partial\Omega$ is called an ε-sample of $\partial\Omega$, if for every point $p \in \partial\Omega$ there is a sample point $q \in P$, such that $|p - q| \leq \varepsilon \cdot \mathit{lfs}\,(p)$.*

Next, we define a special *restriction*:

Definition 4 (Amenta et al. [11]). *Let $\mathcal{D}\,(P)$ be the Delaunay triangulation of the point set P. The restriction of $\mathcal{D}\,(P)$ to $\partial\Omega$, denoted as $\mathcal{D}_{|\partial\Omega}\,(P)$, contains the facets in $\mathcal{D}\,(P)$ whose dual Voronoi edges intersect $\partial\Omega$.*

We shall refer to a facet whose dual Voronoi edge intersects $\partial\Omega$ as a *restricted* facet.

In [12], the following useful theorem and lemma are proved:

Theorem 1 (Boissonnat et al. [12]). *If P is an ε-sample of $\partial\Omega$ with $\varepsilon < 0.09$, then:*

- *$\mathcal{D}_{|\partial\Omega}\,(P)$ is a 2-manifold ambient isotopic to $\partial\Omega$ and*
- *the 2-sided Hausdorff distance between $\mathcal{D}_{|\partial\Omega}\,(P)$ and $\partial\Omega$ is $O(\varepsilon^2)$.*

Lemma 1 (Boissonnat et al. [12]). *If P is an ε-sample of $\partial\Omega$ with $\varepsilon < 0.09$, then every Voronoi edge of the Voronoi diagram of P intersects $\partial\Omega$ at most once and transversally.*

We next define the *surface ball* of a restricted facet:

Definition 5 (Oudot et al. [3]). *Suppose that f is a restricted facet and e is f's dual Voronoi edge. Every point $p \in e \cap \partial\Omega$ is the center of an open ball B containing no points of P, and whose boundary sphere passes through the vertices of f. The surface ball of f is an arbitrary ball B.*

A real point p is called a *vertex*, if it has been already inserted into the mesh. An *element* t is a tetrahedron, a (triangular) facet, or an edge. The *diametral ball* $B(t)$ of t is the set of points that lie strictly inside t's smallest circumscribing sphere. The smallest circumscribing sphere of an element t will be sometimes called its *diametral sphere* and denoted by $S(t)$. The radius and the center of t's diametral ball/sphere are denoted by $r(t)$ and $c\,(t)$ respectively. Point p is called a *feature* point (or a feature vertex, if p is inserted into the mesh), if it is a surface point, i.e., $p \in \partial\Omega$. The shortest edge of element t is denoted by $l_{\min}\,(t)$. Finally, the radius-edge ratio of a tetrahedron or facet t is denoted as $\rho\,(t)$, i.e., $\rho\,(t) = \frac{|r(t)|}{|l_{\min}(t)|}$.

III. Algorithm

Our algorithm initially creates a box that contains the object Ω, such that no point on the box is closer than $2\sqrt{2}\delta$ units to the object surface $\partial\Omega$. Parameter δ is chosen by the user and it can be assigned to any positive value. (It will be clear in Section V that the lower δ is, the better the mesh boundary will approximate $\partial\Omega$.) Next, the Delaunay triangulation of this box (8 vertices totally) is computed. This triangulation is the initial mesh $\mathcal{M}$ where the refinement starts from.

During the refinement, some vertices are inserted exactly on the box; these vertices are called *box vertices*. The edges that lie precisely on one of the 12 edges of the bounding box are called *box edges*. We further divide the box vertices into two categories: *box-edge vertices* and *non-box-edge vertices*. The former vertices lie precisely on a box edge, while the later do not. The facets that lie precisely on one of the 6 faces of the box are called *box facets*. For example, the initial mesh $\mathcal{M}$ contains just 8 box vertices (which are also box-edge vertices) and 12 box edges (among other edges). Note that the endpoints of a box edge are always box edge vertices, but the opposite is not always true. We shall refer to the vertices that are neither box vertices nor feature vertices as *free vertices*.

Next, we define two types of tetrahedra:

- *intersecting tetrahedra*: tetrahedra whose circumsphere intersects $\partial\Omega$ (i.e., there is at least one feature point on or inside their circumsphere), and
- *skinny tetrahedra*: non-intersecting tetrahedra whose circumcenter lies (strictly) inside Ω and radius-edge ratio is larger than or equal to a user-specified parameter $\bar{\rho}$.

Observe that a tetrahedron $t \in \mathcal{M}$ may be neither an intersecting nor a skinny element or may be both.

The algorithm inserts new vertices for two reasons: to guarantee that the mesh boundary is close to the object surface and to remove tetrahedra or facets with large radius-edge ratio. Specifically, let t be a tetrahedron in $\mathcal{M}$; the following four rules are checked in this order:

- **R1**: If t is an intersecting tetrahedron and the closest feature point —say p —to $c(t)$ is at a distance not closer than δ to any other feature vertex, then p is inserted.
- **R2**: If t is an intersecting tetrahedron with radius larger than or equal to 2δ, then $c(t)$ is inserted.
- **R3**: If t is adjacent to a restricted facet f whose surface ball has radius larger than or equal to δ, then insert the center of the surface ball.
- **R4**: If t is a skinny tetrahedron, then $c(t)$ is inserted.

Observe that R1 and R3 insert only feature vertices and that R2 and R4 insert only free vertices.

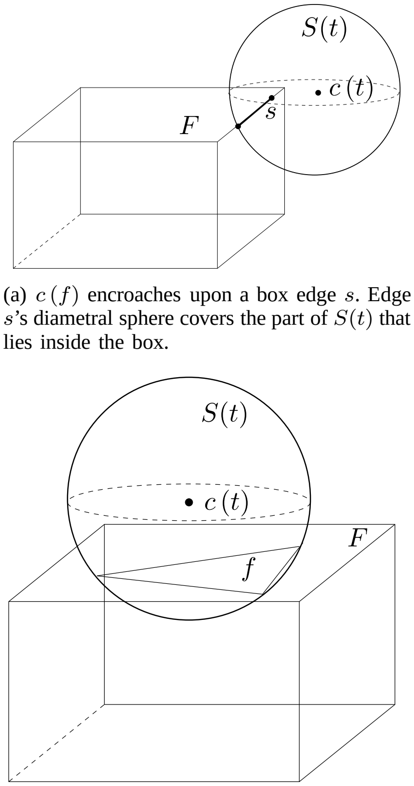
Every time a steiner vertex p, inserted into the mesh as dictated by the four rules above, happens to be a feature vertex, then all the non-feature vertices closer than 2δ to p are deleted. Notice that no box vertices are deleted as a result, since they are separated from the object surface by a distance of at least $2\sqrt{2}\delta$ units. Also, observe that no feature vertices are deleted. Therefore, our algorithm deletes only free vertices so far.

Whenever there is no tetrahedron for which R1, R2, R3, or R4 apply, the refinement process terminates. *The final mesh reported is the set of tetrahedra whose circumcenter lies inside* Ω.

For reasons that will become obvious in Section IV, no vertices should be inserted outside the bounding box. Notice, however, that vertices inserted due to R2 may lie outside the bounding box. To deal with such cases, our algorithm observes special *encroachment* rules similar to the encroachment rules described in [1, 5, 19].

More precisely, assume that R2 is triggered for a tetrahedron t and $c(t)$ lies outside or on the box. In this case, there will always exist a box edge s or a box facet f whose diametral sphere covers the part of $S(t)$ lying inside the box. See Figure 1 for an illustration. We say that $c(t)$ *encroaches upon* s or f.

Every time the circumcenter $c(t)$ of an element t (for which R2 is activated) does not lie strictly inside the box, then $c(t)$ is rejected for insertion. Instead, we locally traverse the triangulation and find an encroached box edge s, if one exists. If such a box edge s is found (see Figure 1(a)), then its diametral ball $B(s)$ is emptied of free and non-box-edge vertices and $c(s)$ is inserted. If there is no an encroached box edge s, then $c(t)$ must encroach upon a box facet f (see Figure 1(b)). In this case, the diametral ball $B(f)$ of f is emptied of free vertices and $c(f)$ is inserted.



(a) $c(f)$ encroaches upon a box edge s. Edge s's diametral sphere covers the part of $S(t)$ that lies inside the box.

(b) $c(f)$ does not encroach upon a box edge but it encroaches upon a box facet f. Facet f's diametral sphere covers the part of $S(t)$ that lies inside the box.

Figure 1. The circumcenter $c(t)$ of a tetrahedron t (not shown) does not lie inside the box. Its sphere $S(t)$ is empty of vertices due to the Delaunay property.

There might be the case, however, the circumcenter $c(f)$ of an encroached box facet f lies outside the box. To prevent the insertion of such vertices, we protect the box edges with their diametral spheres. Specifically, suppose that the circumcenter $c(f)$ of an encroached box facet is considered for insertion. We check whether or nor $c(f)$ lies in a box edge s''s diametral ball $B(s')$. If not, then $c(f)$ is inserted. Otherwise, $c(f)$ is not inserted. In this case, we say that $c(f)$ *encroaches upon* s. Every time $c(f)$ encroaches upon a box edge s', $B(s')$ is emptied of free and non-box-edge vertices and $c(s')$ is inserted. Since after the insertion of any box vertex, the diametral ball of any box edge will be empty of box vertices, Lemma 1 of Shewchuk [19] assures us that

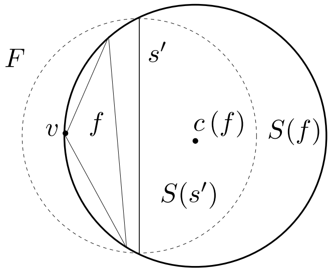


Figure 2. For the sake of contradiction, suppose that the center $c\,(f)$ of an encroached box facet f's diametral sphere $S(f)$ lies outside the box, as illustrated. There should be a box edge s' that lies between the interior of f and $c\,(f)$. Observe that s''s diametral sphere $S(s')$ contains f's vertices. But this is a contradiction: no box edge's diametral sphere contains another box vertex.

whenever the circumcenter of an encroached box facet f is inserted, it will always lie on the box; see Figure 2 for an explanation.

In summary, suppose that the circumcenter $c\,(t)$ of an intersecting tetrahedron t lies outside or on the box ($c\,(t)$ is not actually inserted into the mesh). The encroachment rules are enumerated below in descending priority:

- **E1**: If $c\,(t)$ encroaches upon a box edge s, then $B(s)$ is emptied of free and non-box-edge vertices and $c\,(s)$ (i.e., the midpoint of s) is inserted.
- **E2**: If $c\,(t)$ encroaches upon a box facet f and $c\,(f)$ does not encroach upon a box edge s', then $B(f)$ is emptied of free vertices and $c\,(f)$ is inserted.
- **E3**: If $c\,(t)$ encroaches upon a box facet f, but $c\,(f)$ encroaches upon a box edge s', then $B(s')$ is emptied of free and non-box-edge vertices and $c\,(s')$ is inserted.

Note that none of the encroachment rules deletes feature vertices; the only vertices that might be deleted are free or non-box-edge vertices.

IV. Proof of Quality

In this section, we prove that if the quality parameter is not less than 1, i.e., $\bar{\rho} \geq 1$, then our algorithm terminates outputting tetrahedra with radius-edge ratio less than $\bar{\rho}$ and boundary facets with planar angles larger than 30° (see Theorem 2).

Note that termination and quality are not compromised by any positive value of δ. Parameter δ affects only the fidelity guarantees (see Section V). Sadly, $\bar{\rho}$ will deteriorate in the next section; the fidelity guarantees we give (see Section V) require that $\bar{\rho}$ be not less than 2.

Lemma 2 (Shortest edge)**.** *Suppose that $\bar{\rho} \geq 1$. Let v be an inserted vertex and e an edge incident to v. One of the following holds:*

- *if v is a feature vertex, then $|e| \geq \delta$,*
- *if v a free vertex, then $|e| \geq \bar{\rho}\delta$, and*
- *if v is a box vertex, then $|e| \geq 2\delta$.*

Proof: We will prove this lemma's statement using induction on the number of the vertices inserted.

Initially, only the 8 box vertices of the bounding box are triangulated. Since any point on the box is not closer than $2\sqrt{2}\delta$ to any feature point, the initial edges are definitely larger than 2δ (actually they are larger than $4\sqrt{2}\delta$) and the statement holds.

Assume that the statement holds after the insertion of the $(i-1)^{\text{th}}$ vertex, $i = 10, 11, \dots$ (induction hypothesis), and that the i^{th} vertex —say v —is inserted. The lemma will be proved, if we show that the lemma's statement holds for the edges incident to v. We separate cases.

Assume that v is a feature vertex. Only R1 or R3 should have been responsible for v's insertion. In this case, v is no closer than $2\sqrt{2}\delta$ to any box vertex by construction. We also claim that v is not closer than δ to any other feature vertex. If R1 applies then our claim is obviously true. Suppose that R3 applies. Let f be the restricted facet. From Definition 5, the center v of any surface ball B of f is empty of vertices. But R3 says that B has radius at least δ and therefore no vertex is closer to v than δ. Moreover, notice that the algorithm deletes all the free vertices closer than 2δ to v which implies that the feature vertex v cannot be closer than δ to any vertex and the statement holds.

Now, assume that v is a free vertex. Only R2 or R4 should have been responsible for v's insertion. If R2 applies, then from the Delaunay property, v is separated from any vertex by a distance at least equal to $2\delta > \delta$. If R4 applies, then let t be the skinny tetrahedron. Since $\rho\,(t) \geq \bar{\rho}$, we get that the circumradius of t is at least $|r(t)| \geq \bar{\rho}\,|l_{\min}\,(t)|$; but $l_{\min}\,(t)$ is an edge already in the mesh and by the induction hypothesis is longer than δ, yielding that $|r(t)| \geq \bar{\rho}\delta$. Since v is equal to $c\,(t)$, no edge incident to v is closer than $\bar{\rho}\delta$ to v, and the statement holds.

So far, we have not covered the case v is a box vertex. Let t be the intersecting tetrahedron whose center lies on or outside the box.

Assume that E1 applies. Let s be the encroached edge. Recall that v is the center $c\,(s)$ of s's diametral ball. As we have already mentioned (see Figure 1(a)), $S(s)$ covers the part of $S(t)$ that lies inside the box. Therefore, $S(s)$ contains a feature point, simply because $S(t)$ contains a feature point due to R2. Hence, $|r(s)|$ is at least $2\sqrt{2}\delta$ by the way the bounding box was constructed. Hence, after the free and non-box-edge vertices are removed from $B(s)$ (as E1 suggests), $c\,(s)$ will not be closer than $2\sqrt{2}\delta$ to any vertex and the statement holds.

Now, assume that E2 applies. Let f be the encroached facet. Recall that v is the center $c\,(f)$ of f's diametral ball. As we have already mentioned (see Figure 1(b)), $S(f)$ covers the part of $S(t)$ that lies inside the box. The same reasoning as above yields that $|r(f)|$ is at least $2\sqrt{2}\delta$ by the way the bounding box was constructed. Also, note that since f belongs to the triangulation, $B(f)$ does not contain any

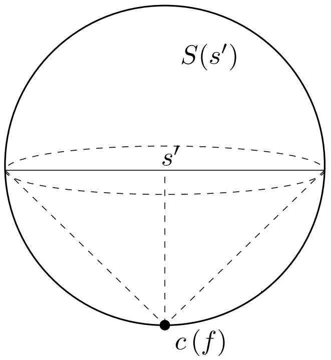


Figure 3. The center $c\,(f)$ of the diametral ball of an encroached box facet f (not shown) is considered for insertion, but it lies inside the diametral sphere $S(s')$ of a box edge s'. The smaller value that $|s'|$ can take is when $c\,(f)$ lies on $S(s')$ and on the bisector of s' as shown. From the isosceles right triangle and from the fact that $c\,(f)$ cannot be closer than $2\sqrt{2}\delta$ to any of s''s endpoints, we get that $\frac{|s'|}{2} \geq 2\delta$; therefore, the midpoint $c\,(s')$ of s' cannot be closer than 2δ to any vertex that lies outside $B(s')$.

box vertices. Therefore, after the free vertices are removed from $B(f)$ (as E2 suggests), $c\,(f)$ will not be closer than $2\sqrt{2}\delta$ to any vertex and the statement holds.

Lastly, assume that v was inserted due to E3. Let s' be the box edge upon which $c\,(f)$ encroaches. Recall that v is the center $c\,(s')$ of s''s diametral ball. As proved in the previous paragraph, $c\,(f)$ is not closer than $2\sqrt{2}\delta$ to any other box vertex. Since $c\,(f)$ lies in the diametral ball $B(s')$ of s', $|r(s')|$ is at least $\frac{2\sqrt{2}\delta}{\sqrt{2}} = 2\delta$ (see Figure 3 for an illustration). Therefore, after the deletion of the free and non-box-edge vertices from $B(s')$, $c\,(s')$ will not be closer than 2δ to any other vertex, and the statement holds. ■

The next Lemma shows that the boundary facets of the output mesh are in fact restricted facets.

Lemma 3. *Let V be the set of vertices of the output mesh $\mathcal{M}$. The set of the boundary facets is a subset of $\mathcal{D}_{|\partial\Omega}\,(V)$.*

Proof: Recall that the tetrahedra reported as part of the mesh have circumcenters that lie inside the object Ω. Therefore, a facet f is a facet of the mesh boundary if it is incident upon a tetrahedron t_i whose circumcenter lies inside Ω and upon a tetrahedron t_j whose circumcenter lies outside Ω or on its surface $\partial\Omega$. But this means that the dual Voronoi edge of f intersects $\partial\Omega$, and therefore, f belongs to $\mathcal{D}_{|\partial\Omega}\,(V)$. ■

Theorem 2. *If $\bar{\rho} \geq 1$, then the algorithm terminates. Also, all the mesh tetrahedra have radius-edge ratio less than $\bar{\rho}$ and all the boundary facets have planar angles larger than* 30°.

Proof: Note that we never insert vertices outside the bounding box which means that the algorithm inserts vertices inside a finite volume. Lemma 2 suggests that any edge introduced into the mesh will have length larger than δ, and therefore, termination is guaranteed.

Upon termination, the tetrahedra reported as part of the mesh have circumcenters that lie inside Ω and therefore they cannot be skinny, because otherwise R4 would apply. This implies that any mesh tetrahedron has radius-edge ratio less than $\bar{\rho}$.

Since a boundary facet f is a restricted facet (by Lemma 3), R3 guarantees that the radius $r(f)$ of f's diametral ball cannot be larger than or equal to δ. From Lemma 2, we also know that the shortest possible edge is at least δ units long. Therefore, $\rho\,(f) = \frac{|r(f)|}{|l_{\min}(f)|} < \frac{\delta}{|l_{\min}(f)|} \leq \frac{\delta}{\delta} = 1$. It is well known that a facet f has radius-edge ratio less than 1 if and only if its smallest planar angle is larger than 30°. ■

V. Proof of Fidelity

In this section, we further restrict $\bar{\rho}$ and derive an upper bound for δ, such that the boundary of the final mesh is a provably good topological and geometrical approximation of $\partial\Omega$. Our goal is to prove that the mesh boundary is equal to $\mathcal{D}_{|\partial\Omega}\,(E)$ for E a 0.09-sample of $\partial\Omega$ (see Theorem 3 of this section). To see why this is enough, recall that from Theorem 1, the restriction of a 0.09-sample of $\partial\Omega$ to $\partial\Omega$ is a good topological and geometrical approximation of $\partial\Omega$.

Let V be the set of vertices appeared in the final mesh and E be equal to $V \cap \partial\Omega$.

Lemma 4. *Upon termination, for any point $p \in \partial\Omega$, there is a vertex $v \in E$, such that $|p - v| < 5\delta$.*

Proof: Recall that upon termination, there is no tetrahedron for which R1, R2, R3, or R4 apply.

Let p be an arbitrary point on $\partial\Omega$. Point p has to lie on or inside the circumsphere of a tetrahedron t. Hence, t is an intersecting tetrahedron. Suppose that point p' is the feature point closest to $c\,(t)$. Note that $|c\,(t) - p| \geq |c\,(t) - p'|$ and therefore p' does not lie outside t's circumsphere. There must exist a feature vertex v closer than δ to p', since otherwise R1 would apply for t. We also know that t's circumradius has to be less than 2δ, since otherwise R2 would apply for t. From the triangle inequality, we get that $|p - v| \leq |p' - v| + |p' - p| \leq |p' - v| + 2\,|r(t)| < \delta + 4\delta = 5\delta$. ■

From Lemma 4 and Definition 3, the following corollary follows:

Corollary 1. *If $\delta \leq \frac{0.09}{5} \cdot lfs_{\min}$, then E is a 0.09-sample of $\partial\Omega$.*

Corollary 1 implies it is enough to prove that the mesh boundary is equal to $\mathcal{D}_{|\partial\Omega}\,(E)$, with E being equal to $V \cap \partial\Omega$.

Lemma 5. *If $\bar{\rho} \geq 2$, then any facet $f \in \mathcal{D}_{|\partial\Omega}\,(V)$ has its vertices on $\partial\Omega$.*

Proof: Suppose that f has a vertex v which does not lie on $\partial\Omega$, i.e., v is not a feature vertex. From Lemma 2, we get that all the edges incident to v are longer than or equal to 2δ. Hence, the radius of any sphere circumscribing f has to be at least δ which raises a contradiction: the diametral ball of any restricted facet f has radius less than δ due to R3. ■

The following Theorem proves the fidelity guarantees achieved by our algorithm:

Theorem 3. *If $\delta \leq 0.09 \cdot lfs_{\min}$ and $\bar{\rho} \geq 2$, then the mesh boundary is a 2-manifold ambient isotopic to $\partial\Omega$ and the 2-sided Hausdorff distance between the mesh boundary and $\partial\Omega$ is $O(\delta^2)$.*

Proof: By Theorem 1, it is enough to prove that the mesh boundary is the restriction to $\partial\Omega$ of a 0.09-sample of $\partial\Omega$. We will, in fact, show that the mesh boundary is equal to $\mathcal{D}_{|\partial\Omega}(E)$ which is the restriction to $\partial\Omega$ of a 0.09-sample of $\partial\Omega$, by Corollary 1.

We will first show that $\mathcal{D}_{|\partial\Omega}(V) \subseteq \mathcal{D}_{|\partial\Omega}(E)$. Let f belong to $\mathcal{D}_{|\partial\Omega}(V)$. Lemma 5 implies that f's vertices lie on $\partial\Omega$. Also, any surface ball B of f does not contain any vertex of V, by Definition 5. Therefore, B does not contain any vertex of E, since E is a subset of V. But that means that f is contained in $\mathcal{D}_{|\partial\Omega}(E)$ as well.

We next prove that that the mesh boundary is equal to $\mathcal{D}_{|\partial\Omega}(V)$. We have already proved in Lemma 3 that the mesh boundary is a subset of $\mathcal{D}_{|\partial\Omega}(V)$. For the other direction, let f belong to $\mathcal{D}_{|\partial\Omega}(V)$. Observe that since $\mathcal{D}_{|\partial\Omega}(\subseteq)\mathcal{D}_{|\partial\Omega}(E)$, Lemma 1 suggests that the dual Voronoi edge of f intersects $\partial\Omega$ exactly once and transversally. Hence, there are two tetrahedra t_i and t_j sharing facet f, such that one tetrahedron has its circumcenter inside Ω and the other has its circumcenter outside or on Ω. But that means that f is also a member of the mesh boundary.

Now, the only fact we need to prove is that $\mathcal{D}_{|\partial\Omega}(E)$ is a subset of $\mathcal{D}_{|\partial\Omega}(V)$, since that would imply that $\mathcal{D}_{|\partial\Omega}(E)$ is equal to $\mathcal{D}_{|\partial\Omega}(V)$. But this is, in essence, proved in Lemma 4.5 of Oudot et al. [3], and the desired result follows. ■

VI. Experimental Evaluation

This section presents the final meshes generated by our algorithm on synthetic and real medical data. All the experiments were conducted on a 64 bit machine equipped with a 2.5 GHz Intel Core 2 Duo CPU and 4 GB of main memory.

We used the *Insight Toolkit* (ITK) for image processing [20]. ITK provides, among others, the implicit function f that describes the object Ω to be meshed (see Section II). More precisely, given a real point p, f can tell whether the voxel to which p belongs is inside, outside, or exactly on Ω. The actual mesh generator was built on top of the *Computational Geometry Algorithms Library* (CGAL) [21]. CGAL offers flexible data structures for Delaunay point insertions and removals. For the 3D visualization of the final mesh, we used the *Visualization Toolkit* (VTK) [22]; after the termination of the mesher, we write the output mesh in a VTK file format which is read by Paraview [23], an open source visualization application.

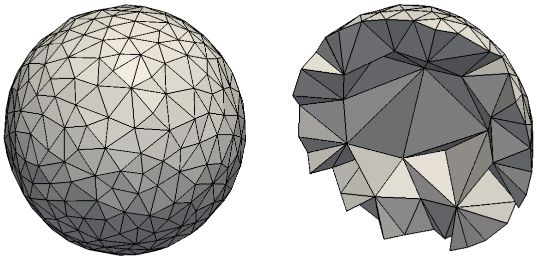


Figure 4. The isosurface and a cross section of a sphere meshed by our algorithm. The surface consists of 750 triangles whose planar angles are larger than $30°$. The whole mesh consists of 1285 tetrahedra whose radius-edge ratio is less than 2. The refinement process lasted for 4 seconds.

This section is divided into two parts. Subsection VI-A illustrates how our algorithm (described in Section III) behaves in practice, while Subsection VI-B shows how a small modification of our algorithm yields a mesh consisting of tetrahedra with dihedral angles larger than $10°$.

A. Theory verification

For all the experiments of this subsection, we set $\bar{\rho}$ to 2, and therefore (from Theorem 2) all the output tetrahedra are guaranteed to have radius-edge ratio less than 2 and all the boundary facets are guaranteed to have planar angles larger than $30°$. Recall that quality is not affected by any value of δ.

Although the fidelity guarantees we give hold for a very small value of δ (see Theorem 3), we wanted to see if our algorithm works well for much larger values of δ. A larger value of δ also implies that the size of the output mesh is smaller. Small-size meshes are desirable for two reason: first, because the mesh generation execution time is considerably less and second, because finite element simulations [24, 25] on them run faster. We observed that even though the fidelity guarantees proved in Section V do not hold for large δ, the results in fact are pretty good. A study of the impact of δ on the accuracy of non-rigid registration will appear elsewhere.

We first tested our algorithm on synthetic data. We chose to mesh a sphere, because we know how the output mesh should look like. Parameter δ is set to a value 50 times larger than the value Theorem 3 suggests. Figure 4 shows the mesh generated by our algorithm. The refinement process lasted for 4 seconds.

We next tested our algorithm on a real human brain image obtained from Huashan Hospital[1]. This 3D image has been acquired by a *Magnetic Resonance Imaging* (MRI) scanner.

[1] Huashan Hospital, 12 Wulumuqi Zhong Lu, Shanghai, China.

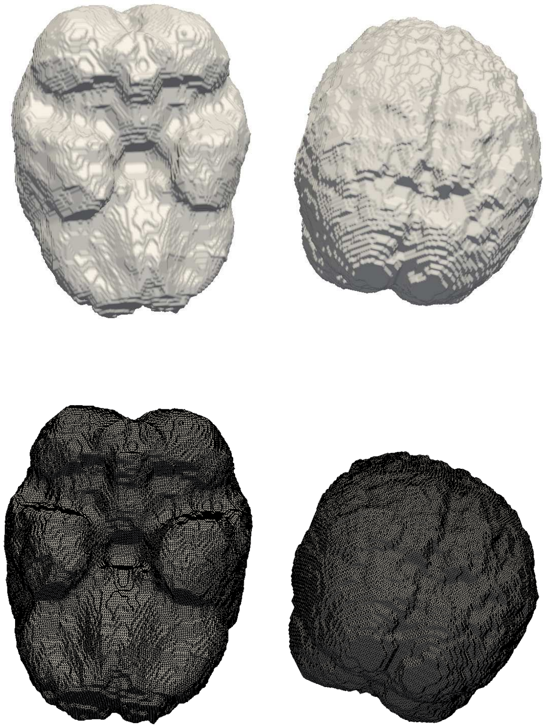


Figure 5. **(Top row)** Two views of the isosurface (generated by Paraview) of the brain we used in our experiments. **(Bottom row)** The boundary triangles (of the isosurfaces of the top row) are shown. The isosurface itself consists of 90,742 triangles.

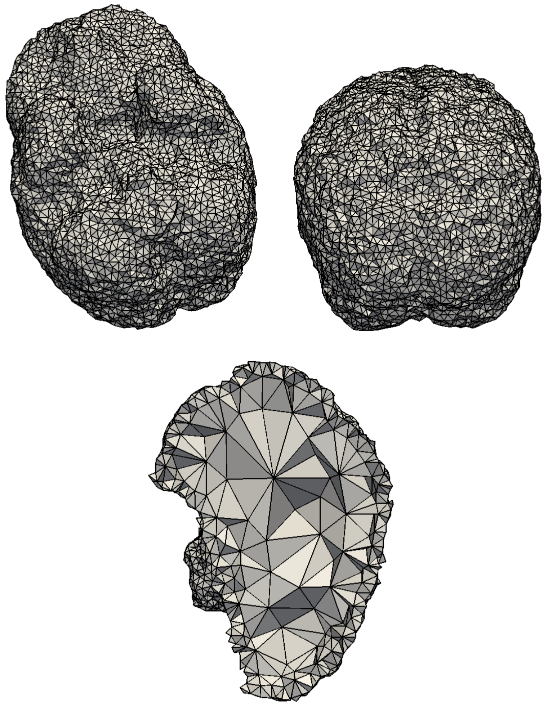
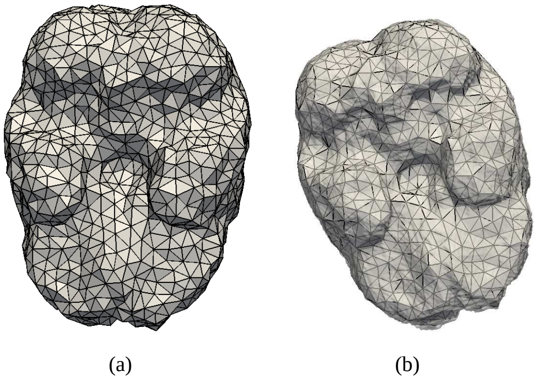


Figure 6. Two views of the isosurface and a cross section of the interior of the brain generated by our algorithm. The isosurface itself consists of 12,340 triangles. All these (boundary) triangles have planar angles larger than 30°. The total number of tetrahedra is only 27,333. All the tetrahedra have radius-edge ratio less than 2. The refinement took 53 seconds.



(a) (b)

Figure 7. **(a)** The final mesh reported by our algorithm. The quality parameter $\bar{\rho}$ has been set to 2 and δ to 6mm. The mesh consists of 5589 tetrahedra and 2856 boundary facets. All the tetrahedra have radius-edge ratio less than 2 and the boundary facets have planar angles larger than $30°$, as it is proved. The refinement process terminated in 14 seconds. **(b)** The drawn edges are the edges of the boundary facets of our mesh shown at the left. The gray surface is the "true" surface (shown alone in Figure 5).

The segmented MR brain has been already stored in ITK format. This (input) image consists of $316{\times}316{\times}168$ voxels. The size of each voxel is $0.9375 \times 0.9375 \times 1.5$ mm^3.

Meshing a brain is a quite challenging task, since it is a complex geometry. We compare the fidelity of our mesh with that obtained by the isosurface contour filter of Paraview. This filter uses a Marching Cubes [13] variation. Figure 5 shows how Paraview extracted the isosurface of the brain; observe that some parts are of high curvature.

Figure 6 shows the output of our algorithm. The quality parameter $\bar{\rho}$ has been set to 2, and δ has been set to 3mm. Observe the grading from the boundary towards the interior. The refinement process took 53 seconds.

Figure 7 shows the output mesh generated by our algorithm for the same brain image, but with an even larger δ (i.e., 6mm). We also overlaid the output mesh on the isosurface obtained by Paraview (shown in Figure 5) which we trust as an "acceptable" approximation. Observe that they are still very close to each other.

B. Sliver removal

Although our algorithm guarantees that all the mesh tetrahedra will have a small radius-edge ratio, some tetrahedra, referred to as *slivers*, may have very small dihedral angles. Figure 8 presents the histogram of the dihedral angles of the mesh shown in Figure 7(a). Every bar has a width of 2°. The height of every bar represents the number of tetrahedra whose smallest dihedral angle falls into the range of the bar. The histogram shows that few tetrahedra with very small

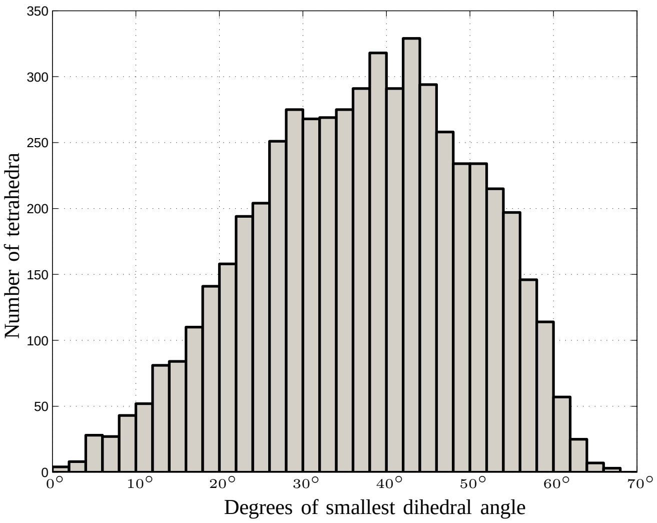


Figure 8. The distribution of the dihedral angles of the mesh shown in Figure 7(a). Although the majority of the output tetrahedra have good angles, there are a few with dihedral angles less than 2°.

dihedral angles may survive.

It would be interesting to see how our algorithm would perform if we split tetrahedra with small dihedral angles rather than splitting tetrahedra with large radius-edge ratio. We, therefore, replaced rule R4 (the rule that splits tetrahedra with large radius-edge ratio) of our original algorithm with the following rule:

- **R4′:** If the circumcenter $c(t)$ of a tetrahedron t lies inside the object Ω and its smallest dihedral angle is less than α degrees, then $c(t)$ is inserted.

Parameter α is controlled by the user. When the algorithm terminates, all the dihedral angles of the output mesh will be larger than α degrees. Figure 9 shows the output mesh of our modified algorithm with parameter α being set to $10°$. Although we offer no guarantees in this case, the final mesh describes fairly well the object surface $\partial\Omega$ and does not contain any slivers: all the output tetrahedra have dihedral angles larger than $10°$. We also plot the histogram of the dihedral angles in this case; see Figure 10.

VII. Conclusions and Future work

We have implemented a 3D Delaunay refinement algorithm which guarantees that the output tetrahedra have radius-edge ratio less than 2 and the boundary facets have planar angles larger than $30°$ (see Theorem 2). We also prove that if δ is sufficiently small, then the mesh boundary is a 2-manifold ambient isotopic to the object surface and the 2-sided Hausdorff distance between the mesh boundary and the object surface is $O(\delta^2)$ (see Theorem 3).

Experimental evaluation of our algorithm on real medical data (see Subsection VI-A) confirmed the theory and also showed that we can generate quite coarse meshes very fast.

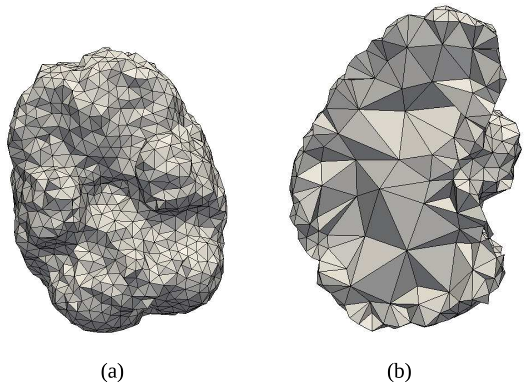

(a) (b)

Figure 9. The final mesh and a cross section generated by our modified algorithm with α equal to $10°$. Parameter δ is set to the same value with the value we used for the mesh shown in Figure 7(a). The mesh consists of 6491 tetrahedra and 2936 boundary facets. All the tetrahedra have dihedral angles larger than $10°$ and the boundary facets have planar angles larger than $30°$. The refinement process terminated in 15 seconds.

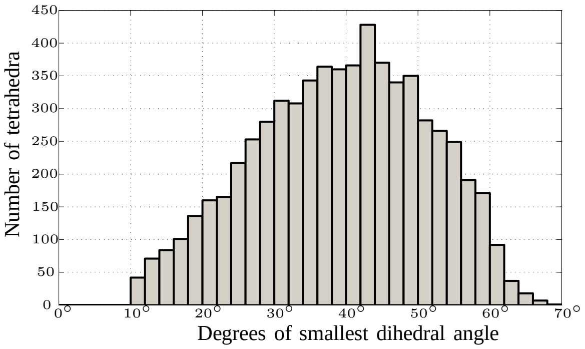


Figure 10. The distribution of the dihedral angles of the mesh shown in Figure 9. All the dihedral angles are larger than $10°$.

An issue we do not completely address in this paper is the elimination of slivers. In fact, we observed that elements with dihedral angles less than $2°$ may survive (see Figure 8). As mentioned in the Introduction, slivers can be eliminated using techniques well described in the literature [8, 9]. Although we leave *sliver elimination* as future work, preliminary results (see Subsection VI-B) have shown that in practice, if we just split slivers at their circumcenter, then the algorithm terminates generating a mesh that describes fairly well the object surface (see Figure 9) and does not contain any slivers (see Figure 10).

As shown in Subsection VI-A, our algorithm generates fewer elements as we travel away from the object surface. We would also like to achieve good grading along the object surface, that is, fewer boundary facets on the parts of the surface far from the medial axis. The fully-graded version of our algorithm is left as future work.

ACKNOWLEDGMENTS

We would like to thank Yixun Liu for providing us the brain image. This work is supported in part by NSF grants: CCF-0916526, CCF-0833081, and CSI-719929 and by the John Simon Guggenheim Foundation.

REFERENCES

[1] J. R. Shewchuk, "Tetrahedral mesh generation by Delaunay refinement," in *Proceedings of the* 14*th* *ACM Symposium on Computational Geometry*, Minneapolis, MN, 1998, pp. 86–95.

[2] L. P. Chew, "Guaranteed-quality Delaunay meshing in 3D," in *Proceedings of the* 13*th* *ACM Symposium on Computational Geometry*, Nice, France, 1997, pp. 391–393.

[3] S. Oudot, L. Rineau, and M. Yvinec, "Meshing Volumes Bounded by Smooth Surfaces," in *International Meshing Roundtable*. San Diego, California, USA: Springer-Verlag, September 2005, pp. 203–220.

[4] L. Rineau and M. Yvinec, "Meshing 3D Domains Bounded by Piecewise Smooth Surfaces," in *International Meshing Roundtable*, 2007, pp. 443–460.

[5] A. Chernikov and N. Chrisochoides, "Three-Dimensional Semi-Generalized Point Placement Method for Delaunay Mesh Refinement," in *Proceedings of the* 16*th* *International Meshing Roundtable*. Seattle, WA: Elsevier, October 2007, pp. 25–44.

[6] Dobrina Boltcheva, Mariette Yvinec, and Jean-Daniel Boissonnat, "Mesh Generation from 3D Multi-material Images," in *Medical Image Computing and Computer-Assisted Intervention*. Springer, September 2009, pp. 283–290.

[7] J.-P. Pons, F. Ségonne, J.-D. Boissonnat, L. Rineau, M. Yvinec, and R. Keriven, "High-Quality Consistent Meshing of Multi-label Datasets," in *Information Processing in Medical Imaging*, 2007, pp. 198–210.

[8] S.-W. Cheng, T. K. Dey, H. Edelsbrunner, M. A. Facello, and S.-H. Teng, "Sliver exudation," *Journal of the ACM*, vol. 47, no. 5, pp. 883–904, 2000.

[9] Jane Tournois, Rahul Srinivasan, and Pierre Alliez, "Perturbing Slivers in 3D Delaunay Meshes," in *Proceedings of the* 18*th* *International Meshing Roundtable*. Salt Lake City, Utah, USA: Sandia Labs, October 2009, pp. 157–173.

[10] L. P. Chew, "Guaranteed-quality mesh generation for curved surfaces," in *Proceedings of the* 9*th* *ACM Symposium on Computational Geometry*, San Diego, CA, 1993, pp. 274–280.

[11] N. Amenta, S. Choi, T. K. Dey, and N. Leekha, "A Simple Algorithm for Homeomorphic Surface Reconstruction," *International Journal of Computational Geometry and Applications*, vol. 12, no. 1-2, pp. 125–141, 2002.

[12] J.-D. Boissonnat and S. Oudot, "Provably good sampling and meshing of surfaces," *Graphical Models*, vol. 67, no. 5, pp. 405–451, 2005.

[13] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[14] N. Molino, R. Bridson, J. Teran, and R. Fedkiw, "A crystalline, red green strategy for meshing highly deformable objects with tetrahedra," in *Proceedings of the* 12*th* *International Meshing Roundtable*. Sandia National Laboratories, September 2003, pp. 103–114.

[15] F. Labelle and J. R. Shewchuk, "Isosurface stuffing: fast tetrahedral meshes with good dihedral angles," *ACM Transactions on Graphics*, vol. 26, no. 3, p. 57, 2007.

[16] Y. Liu, C. Yao, L. Zhou, and N. Chrisochoides, "A Point based Non-Rigid Registration for Tumor Resection using iMRI," in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2010.

[17] A. Fedorov and N. Chrisochoides, "Tetrahedral Mesh Generation for Non-rigid Registration of Brain MRI: Analysis of the Requirements and Evaluation of Solutions," in *International Meshing Roundtable*. Springer Verlag, October 2008.

[18] O. Clatz, H. Delingette, I.-F. Talos, A. J. Golby, R. Kikinis, F. Jolesz, N. Ayache, and S. Warfield, "Robust non-rigid registration to capture brain shift from intra-operative MRI," *IEEE Transactions on Medical Imaging*, vol. 24, no. 11, pp. 1417–1427, November 2005.

[19] J. R. Shewchuk, "Delaunay refinement algorithms for triangular mesh generation," *Computational Geometry: Theory and Applications*, vol. 22, no. 1-3, pp. 21–74, May 2002.

[20] "ITK, Insight Segmentation and Registration Toolkit," http://www.itk.org.

[21] "CGAL, Computational Geometry Algorithms Library," http://www.cgal.org.

[22] "VTK, Visualization Toolkit," http://www.vtk.org.

[23] A. H. Squillacote, *ParaView Guide, A Parallel Visualization Application*. Kitware Inc., 2008.

[24] M. A. Audette, H. Delingette, A. Fuchs, O. Burgert, and K. Chinzei, "A topologically faithful, tissue-guided, spatially varying meshing strategy for computing patient-specific head models for endoscopic pituitary surgery simulation," *Computer Aided Surgery*, vol. 12, no. 1, pp. 43–52, 2007.

[25] M. Audette, M. Miga, J. Nemes, K. Chinzei, and T. Peters, "A Review of Biomechanical Modeling of the Brain for Intrasurgical Displacement Estimation and Medical Simulation," *Biomechanical Systems, General Anatomy*, pp. 83–112, 2007.