Published in final edited form as:
Comput Aided Geom Des. 2012 December 1; 29(9): 707–721. doi:10.1016/j.cagd.2012.07.001.

Quality Tetrahedral Mesh Smoothing via Boundary-Optimized Delaunay Triangulation

Zhanheng Gao[a,b], **Zeyun Yu**[a,*], and **Michael Holst**[c]

[a]Department of Computer Science, University of Wisconsin-Milwaukee, Milwaukee, WI 53211, USA

[b]College of Computer Science and Technology, Jilin University, Changchun, Jilin 130012, China

[c]Department of Mathematics, University of California, San Diego, USA

Abstract

Despite its great success in improving the quality of a tetrahedral mesh, the original optimal Delaunay triangulation (ODT) is designed to move only inner vertices and thus cannot handle input meshes containing "bad" triangles on boundaries. In the current work, we present an integrated approach called boundary-optimized Delaunay triangulation (B-ODT) to smooth (improve) a tetrahedral mesh. In our method, both inner and boundary vertices are repositioned by analytically minimizing the $\mathcal{L}^1$ error between a paraboloid function and its piecewise linear interpolation over the neighborhood of each vertex. In addition to the guaranteed volume-preserving property, the proposed algorithm can be readily adapted to preserve sharp features in the original mesh. A number of experiments are included to demonstrate the performance of our method.

Keywords

Tetrahedral Mesh Smoothing; Optimal Delaunay Triangulation; Mesh Quality; Feature-Preserving; Volume-Preserving

1. Introduction

In scientific and engineering applications, partial differential equations (PDEs) are often used for modeling the development and evolution of the underlying phenomena. In most cases, however, it is difficult and sometimes impossible to find the exact analytic solutions of the PDEs so that numerical approaches have to be employed to approximate the desired solutions. The finite element analysis (FEA) is one of the most useful tools for this purpose. In the FEA, the domain over which the PDEs are defined is partitioned into a mesh containing a large number of simple elements, such as triangles and quadrilaterals in 2D cases and tetrahedra and hexahedra in 3D cases (Djidjev, 2000; Phillippe and Baker, 2001; Ohtake et al., 2001; Knupp, 2002, 2003; Brewer et al., 2003). The quality of the mesh, typically measured by the minimum and maximum angles, can significantly affect the interpolation accuracy and solution stability of the FEA (Babuska and Aziz, 1976;

*Corresponding author: yuz@uwm.edu.

Shewchuk, 2002). Therefore, improving the mesh quality has been an active research area in computational mathematics and computer science. Due to the great popularity in the FEA, 3D tetrahedral meshes will be the focus of our present work.

The methods of mesh quality improvement can be classified into three categories as follows. (1) topology optimization, which modifies the connectivity between mesh vertices while keeping vertex positions unchanged. The edge- or face-swapping methods are commonly used in topology optimization (Freitag and Ollivier-Gooch, 1997; Klingner and Shewchuk, 2008). (2) vertex insertion/deletion, which inserts/deletes vertices to/from the mesh (Chew, 1997; Nave et al., 2004; Escobar et al., 2005; Klingner and Shewchuk, 2008). (3) vertex smoothing, which repositions the coordinates of the vertices while keeping the connectivity unchanged (Bank and Smith, 1997; Freitag, 1997; Canann et al., 1998). Generally speaking, mesh quality improvement is best achieved when all the three methods are properly combined in the mesh smoothing scheme (Klingner and Shewchuk, 2008). In our method described below, we shall focus on the vertex repositioning strategy, i.e. vertex smoothing.

One of the most popular vertex smoothing method is *Laplacian* smoothing, which moves a mesh vertex to the weighted average of its incident vertices (Herrmann, 1976; Field, 1988; Hansbo, 1995). If the neighborhood of the vertex is not a convex polyhedron, the Laplacian smoothing may not lead to a well-positioned mesh. Some angle-based methods were proposed for smoothing 2D triangular and 3D surface meshes (Zhou and Shimada, 2000; Xu and Newman, 2006; Yu et al., 2008). However, these methods are difficult to extend to 3D tetrahedral meshes. Du and Wang (2003) presented a method based on the Centroid Voronoi Tessellation (CVT) concept that is restricted to inner vertices of a mesh. A peeling off operation has to be taken to improve bad tetrahedra on boundaries. Freitag and Plassmann (2001) proposed a method of smoothing planar quadrilateral meshes. Some researchers presented methods for smoothing hexahedral mesh (Li et al., 1999; Knupp, 2001, 2000b; Delanaye et al., 2003; Menédez-Díaz et al., 2005). More recently, some new techniques of vertex smoothing were proposed. Vartziotis et al. (2008, 2009) presented methods of stretching the vertices of a tetrahedron at one time. The methods were extended by Vartziotis and Wipper (2010) to hexahedral mesh. Xu et al. (2009) assigned a quality coordinate for every vertex and calculated the new position by maximizing the combined quality of tetrahedra incident to it. Sirois et al. (2010) used a metric non-conformity driven method to smooth hybrid meshes such as a mesh with hexahedral and tetrahedral elements.

In addition to the above methods, approaches using numerical optimization to compute the new position of a vertex has been an important branch of the vertex smoothing category. The new position of a vertex is computed by optimizing a function that measures the local or global quality of the mesh (Parthasarathy and Kodiyalam, 1991; Canann et al., 1993; Chen et al., 1995; Zavattieri et al., 1996; Freitag Diachin and Knupp, 1999; Knupp, 2000a; Freitag and Plassmann, 2000; Freitag and Knupp, 2002; Escobar et al., 2003; Mezentsev, 2004). In particular, the optimal Delaunay triangulation (ODT) approach (Chen and Xu, 2004) tries to minimize the $\mathcal{L}^1$ error between a paraboloid function and its piecewise linear interpolation over the neighborhood of a vertex. This idea has been extended to 3D tetrahedral mesh smoothing in Tournois et al. (2009). Despite its great success in mesh quality improvement, the original ODT method was derived to optimize the positions of inner vertices only. In other words, the tetrahedral mesh to be smoothed must possess quality triangles on boundaries. In many real mesh models, however, “bad” tetrahedra often occur near or on the boundaries of a domain (Labelle and Shewchuk, 2007; Zhang et al., 2010). Therefore, how to handle the boundary vertex smoothing is an important yet unsolved problem in the original ODT method.

In the present work, we shall provide an analytical method named boundary-optimized Delaunay triangulation (B-ODT) to find the optimal positions of all mesh vertices, including those on boundaries, by minimizing an $\mathcal{L}^1$ error function that is defined in the incident neighborhood of each vertex. The minimization is an unconstrained quadratic optimization problem and has an exact analytic solution when the coefficient matrix of the problem is positive definite. As mentioned above, the quality improvement is often limited if we only perform one method of the three categories (Klingner and Shewchuk, 2008). For this reason, the vertex insertion operation is utilized prior to the vertex repositioning technique.

The remainder of the present paper is organized as follows. Section 2 is focused on the details of the B-ODT mesh smoothing algorithms. The original ODT as well as the vertex insertion schemes are also discussed in this section. We present some experimental results and quality analysis in Section 3, followed by our conclusions in Section 4.

2. Boundary-Optimized Delaunay Triangulation

The framework of our mesh quality improvement method is shown in Fig. 1. The vertex insertion operation is performed prior to the vertex repositioning. We try to insert as few vertices as possible in order to maintain the size of the original mesh. The detail of vertex insertion is described in Subsection 2.3. As for vertex smoothing, three algorithms are summarized below and the details will follow.

Algorithm 1. The original ODT algorithm for smoothing inner vertices

Algorithm 2. The basic B-ODT algorithm for smoothing boundary vertices. This algorithm is guaranteed to preserve the volume of the mesh.

Algorithm 3. The advanced B-ODT algorithm for smoothing boundary vertices. This algorithm preserves both the volume and sharp features of the mesh.

Since the advanced B-ODT algorithm can preserve both the volume and sharp features of the original mesh, in the rest of this paper, we will refer to B-ODT algorithm as the advanced B-ODT algorithm, unless otherwise specified.

2.1. Original ODT algorithm

For an inner vertex x_0 in a tetrahedral mesh $\mathcal{T}$, suppose the neighborhood of x_0 is Ω_0 consisting of a set of tetrahedra $\{\tau\}$. Let x_* be the smoothing result of x_0 and Ω_* the neighborhood of x_* (or the union of tetrahedra incident to x_*) in $\mathcal{T}$, then x_* can be computed by the following original ODT formula (Chen and Xu, 2004):

$$x_* = x_0 - \frac{1}{2|\Omega_0|} \sum_{\tau \in \Omega_0} \left(\nabla |\tau| \sum_{i=1}^{3} \left\| x_{\tau,i} - x_0 \right\|^2 \right) \quad (1)$$

where $x_{\tau,i}$ are the (three) vertices except x_0 of the incident tetrahedron τ. Theoretically, (1) is the unique solution of an optimization problem in which the objective function is the $\mathcal{L}^1$ interpolation error between the paraboloid function $f(x) = \|x - x_0\|^2$ and its linear approximation $f_I(x)$ over Ω_*. $f_I(x)$ is constructed by lifting the the vertices of Ω_* onto $f(x)$. The $\mathcal{L}^1$ error between $f(x)$ and $f_I(x)$ is:

$$Error_* = \left\| f - f_I \right\|_{L^1} = \int_{x \in \Omega_*} |f(x) - f_I(x)| \mathrm{d}x \quad (2)$$

We would point out that for an **inner** vertex x_0, the neighborhoods Ω_0 and Ω_* are identical but it is not true when x_0 is a **boundary** vertex, which explains why the original ODT approach cannot be directly used to optimize boundary vertices. Another important property of (2) is that, if all vertices are fixed, the solution of (2) is proved to be the Delaunay triangulation of these vertices (Chen and Xu, 2004). However, Delaunay triangulation does not guarantee high quality – the positions of the vertices are sometimes more important. In (2) we restrict the optimization problem to the neighborhood of x_* with the assumption that all other vertices are fixed. The optimization problem hence becomes finding the optimal position of x_* to minimize the error in (2).

A more direct way to compute the optimal position of x_* is (Chen, 2004):

$$x_* = x_0 - \frac{1}{2|\Omega_0|}\sum_{\tau\in\Omega_0}\left(\frac{1}{3}S_\tau \mathbf{n}_\tau \sum_{i=1}^{3}\left\|x_{\tau,i} - x_0\right\|^2\right) \quad (3)$$

Here S_τ and $\mathbf{n}_\tau$ are the area and unit normal vector of t_τ, which is the opposite triangle of x_0 in τ, $\mathbf{n}_\tau$ points to the inside of τ.

The following is the algorithm of smoothing inner vertices based on the original ODT method:

2.2. B-ODT algorithms

Let x_* be the smoothing result of a boundary vertex x_0 using the method described below. Ω_0 and Ω_* are the neighborhoods of x_0 and x_* respectively (since x_0 is a boundary vertex, the two neighborhoods are not identical any more). The basic idea of our B-ODT algorithms is to minimize the error in (2). According to (Chen and Xu, 2004), (2) can be rewritten in the following form:

$$Error_* = \frac{1}{4}\sum_{x_k\in\Omega_*}(\left\|x_k - x_0\right\|^2|\omega_k|) - \int_{x\in\Omega_*}\left\|x - x_0\right\|^2 \mathrm{d}x \quad (4)$$

where x_k denotes one of the vertices of Ω_*, ω_k is x_k's neighborhood restricted in Ω_*, and $|\omega_k|$ is its volume.

Note that x_* is also a vertex in Ω_*, we rewrite (4) into the following equation:

$$Error_* = \frac{1}{4}\left(\left\|x_* - x_0\right\|^2|\Omega_*| + \sum_{\tau\in\Omega_*}\left(|\tau|\sum_{i=1}^{3}\left\|x_{\tau,i} - x_0\right\|^2\right)\right) - \int_{\Omega_*}\left\|x - x_0\right\|^2 \mathrm{d}x \quad (5)$$

Since x_0 is a boundary vertex, we let x_* move on the tangent plane of the boundary surface at x_0. Specifically, let $x_* - x_0 = u\mathbf{s} + v\mathbf{t}$, where $\mathbf{s}$ and $\mathbf{t}$ are two orthogonal vectors on the tangent plane and u and v are the corresponding shifting distances. Furthermore, we prove in Appendix B that the tangent plane constraint guarantees that the volumes of Ω_* and Ω_0 are equal.

By using the constraint $x_* = x_0 + u\mathbf{s} + v\mathbf{t}$, we have:

$$\left\|x_* - x_0\right\|^2|\Omega_*| = (u^2 + v^2)|\Omega_*| \quad (6)$$

$$|\tau|\sum_{i=1}^{3}\left\|x_{\tau,i}-x_0\right\|^2=\frac{1}{3}S_\tau<\mathbf{n}_\tau,u\mathbf{s}+v\mathbf{t}+x_0-c_\tau>\sum_{i=1}^{3}\left\|x_{\tau,i}-x_0\right\|^2 \quad (7)$$

where S_τ and $\mathbf{n}_\tau$ are defined the same as in (3). c_τ is any vertex in the triangle t_τ. Here we take c_τ as the barycenter of t_τ. $<\cdot,\cdot>$ is the inner product operation.

Now we represent the integral $\int_{\Omega*}\|x-x_0\|^2\mathrm{d}x$ in (5) by x_* in order to compute the gradient of the objective function. The details are given in Appendix A. Suppose $\{y_i\}_{i=1}^{m}$ are the neighboring vertices of x_0 on the boundary of the tetrahedral mesh $\mathcal{T}$. The order of y_i is determined in the following way: for any $i = 1, \cdots, m$, the cross product between $\overrightarrow{x_0y_i}$ and $\overrightarrow{x_0y_{i+1}}$ points to the outside of Ω_0 (let $y_{m+1} = y_1$). Thus the integral $\int_{\Omega*}\|x-x_0\|^2\mathrm{d}x$ has the following form:

$$\int_{x\in\Omega_*}\left\|x-x_0\right\|^2\mathrm{d}x=\int_{x\in\Omega 0}\left\|x-x_0\right\|^2\mathrm{d}x+\frac{1}{60}\sum_{i=1}^{m}(X_*^2+Y_i^2+Y_{i+1}^2+X_*Y_i+X_*Y_{i+1}+Y_iY_{i+1})\det(X_*,Y_i,Y_{i+1}) \quad (8)$$

Here $X_* = x_* - x_0$, $Y_i = y_i - x_0$, $\det(\cdot)$ is the determinant operation. Note that x_* is limited on the tangent plane at x_0, (8) can be rewritten as:

$$\begin{aligned}\int_{x\in\Omega_*}&\left\|x-x_0\right\|^2\mathrm{d}x\\&=\int_{x\in\Omega_0}\left\|x-x_0\right\|^2\mathrm{d}x\\&+\frac{1}{60}\sum_{i=1}^{m}[u^2+v^2+Y_i^2+Y_{i+1}^2+<u\mathbf{s}+v\mathbf{t},Y_i>+<u\mathbf{s}+v\mathbf{t},Y_{i+1}>+<Y_i,Y_{i+1}>]\det(u\mathbf{s}+v\mathbf{t},Y_i,Y_{i+1})\end{aligned} \quad (9)$$

(9) appears as a cubic function of u, v. However, we prove in Appendix C that the coefficients of all cubic terms are actually zero. Therefore, (9) reduces to a quadratic function of u, v.

Now we can safely say that the objective function in (5) is in fact a quadratic function of u, v. By setting the gradient of (5) as zero, we get a linear system and the solution of this system gives rise to the minimization of (5). In Appendix C, we prove that the coefficient matrix of this linear system is identical to the Hessian matrix of (5). According to the optimization theory, (5) has a unique solution as long as the Hessian matrix is positive definite.

The algorithm of smoothing boundary vertices based on the above discussion is given below.

Besides keeping x_* on the tangent plane at x_0, we further restrict x_* moving along the features of the mesh to preserve the sharp features. Here, we refer to the feature direction at x_0 as the line that passes through x_0 and has the minimal curvature value among all the directions. This line is on the tangent plane; thus the volume is still preserved when x_* moves along this feature line. The direction of the feature line is found by computing the eigenvalues of the following tensor voting matrix at x_0:

$$M=\sum_{i=1}^{m}S_i\mathbf{n}_i\mathbf{n}_i^T \quad (11)$$

Here S_i is the area of surface triangle $x_0 y_i y_{i+1}$ and $\mathbf{n}_i = (n_{ix}, n_{iy}, n_{iz})^T$ is the unit normal vector of $x_0 y_i y_{i+1}$. The matrix M is a positive definite matrix and has three orthogonal eigenvectors. The feature line is determined in the following way. Suppose that the three eigenvalues of M are μ_0, μ_1, μ_2 with $\mu_0 \geq \mu_1 \geq \mu_2$ and $\mathbf{e}_0$, $\mathbf{e}_1$, $\mathbf{e}_2$ are the corresponding eigenvectors. If $\mu_0 \gg \mu_1 \approx \mu_2 \approx 0$, then the neighborhood of x_0 corresponds to a planar feature. In this case, the above Algorithm 2 is used to smooth x_0. If $\mu_0 \approx \mu_1 \gg \mu_2 \approx 0$, then x_0 lies on an edge (linear) feature and the direction of the edge is $\mathbf{e}_2$. In this case, the following Algorithm 3 is used to smooth x_0. If $\mu_0 \approx \mu_1 \approx \mu_2 \gg 0$, then x_0 is at a corner which should not be changed during the vertex smoothing process.

2.3. Removing bad tetrahedra by vertex insertion

There are several types of tetrahedra that can cause very large and/or small dihedral angles (Cheng et al., 1999) (see Fig. 2). One of the most important properties of ODT and B-ODT is the *circumsphere property* (Tournois et al., 2009), i.e., the lengths of edges incident to vertex x_0 tend to be equal after ODT or B-ODT smoothing. Therefore, the ODT and B-ODT algorithms can automatically improve the quality of spire, spear, spindle, spike, splinter and wedge, as the bad angles in these tetrahedra are caused by one or more short edges. For the spade, cap and sliver types of tetrahedra, we shall insert one or two vertices in order to make some short edges, which are then improved by the ODT or B-ODT algorithms, thereby improving the quality of the entire mesh. The details are given below.

For a spade in Fig. 3(a), we first compute the projection of **A** onto the edge **BC**, i.e. **E**. Thus the edge **BC** is split into two new edges **BE**, **CE** and the tetrahedron **ABCD** is split into two new tetrahedra **ABDE**, **ACDE** (Fig. 3(b)). Because **AE** is a short edge, we smooth **A** using the ODT or B-ODT method according to the type of **A** (Fig. 3(c)).

For a cap in Fig. 4(a), we first compute the projection of **A** on face **BCD**, i.e. **E**. Then we split the face **BCD** into three new faces **BCE**, **CDE** and **DBE**, and split the original tetrahedron **ABCD** into three new tetrahedra **ABCE**, **ACDE** and **ADBE** (Fig. 4(b)). Finally, the ODT and B-ODT methods are applied to the new tetrahedra to improve the quality of the mesh (Fig. 4(c)).

For a sliver in Fig. 5(a), we insert two vertices **E** and **F** on the edge **AC** and **BD** respectively. The two new vertices are selected such that the distance between **E** and **F** is the minimum between **AC** and **BD**. Then **AC** and **BD** are both split into two edges, and the tetrahedron **ABCD** is split into four new tetrahedra (Fig. 5(b)). By performing the ODT or B-ODT methods on **E** and **F**, the quality of the new tetrahedra can be improved (Fig. 5(c)).

3. Results

The proposed B-ODT algorithms were tested on several tetrahedral meshes generated from triangular surface meshes that serve as the boundaries of the domains. For every mesh, the smoothing process shown in Fig. 1 is repeated for 20 times. The mesh smoothing results are summarized in Table 1. The comparisons between the B-ODT algorithm (Algorithm 3) and several other approaches, including the original ODT algorithm, topology optimization and the Natural ODT algorithm (Tournois et al., 2009), are also provided in Tab. 1. In Fig. 6–12, the original and smoothed meshes are compared and from the histograms we can see significant improvement of dihedral angles in these meshes.

The main motivation of extending the original ODT method to the B-ODT algorithms is to find the optimal positions for boundary vertices such that the quality of the entire tetrahedral mesh is improved. To illustrate the quality improvement, we compare the smoothing results by using the B-ODT and ODT algorithms. In Table 1, all the minimum and maximum dihedral angles by using the B-ODT algorithm are better than those by the original ODT algorithm, especially on the Retinal model. Note that the minimum dihedral angle in Retinal model is very small and likely occurs on the boundary of the model. Therefore, the proposed B-ODT algorithm can perform much better than the original ODT method.

Although the topology optimization is utilized in many mesh smoothing algorithms, this technique alone may not always improve the quality of a mesh. To show this, we smooth all the meshes in Tab. 1 using only the topology optimization and compare the results with those obtained by using our B-ODT algorithm. From Tab. 1, we can see that the ability of improving mesh quality by using topology optimization alone is limited, compared to the B-ODT algorithm.

The tetrahedral mesh in Fig. 6 is generated by tetrahedralizing randomly-sampled point set on a unit sphere (Si et al., 2010). There are 642 points on the sphere and 87 inner vertices are inserted by the tetrahedralization algorithm. The minimum and maximum dihedral angles of this Random Sphere model are 5.86° and 164.70° respectively. After 20 times of running the B-ODT algorithm, the minimum and maximum dihedral angles are improved to 15.20° and 150.25° respectively. Note that the distribution of the boundary vertices of the smoothed mesh is much more uniform than that of the original mesh, demonstrating that the B-ODT algorithm can smooth both inner and boundary vertices in a tetrahedral mesh.

The B-ODT algorithm is also tested on tetrahedral meshes generated from several biomedical molecules: 2CMP molecule in Fig. 7, Retinal molecule in Fig 8 and Ryanodine receptor (RyR) in Fig. 9. The quality of 2CMP and RyR meshes reaches the best after only 3 B-ODT iterations although all the models in Tab. 1 are processed 20 times. We can also see from Tab. 1 that there are no new vertices introduced in 2CMP and RyR models and only 27 new vertices are inserted in the Retinal mesh. In Fig. 10, we demonstrate the convergence of minimum and maximum dihedral angles with respect to the number of iterations on the Retinal model using the B-ODT algorithm.

The 2Torus (Fig. 11) and FanDisk (Fig. 12) models show the feature-preserving property of the B-ODT algorithm. In order to measure the difference between the original and smoothed meshes, we compute the relative Hausdorff distances between the surface meshes of the original and smoothed models, as shown in Tab. 2. Here, the Hausdorff distance is first computed using the standard definition and then scaled as follows. Let h be the absolute Hausdorff distance between the original and smoothed meshes, and L be the largest side length of the bounding box of the original mesh. The relative Hausdorff distances is defined by $\frac{h}{L}$, which measures the difference of the original and smoothed models relative to the size of the original model. From Tab. 2 we can see that the relative Hausdorff distances between the original and smoothed models are very small showing that our B-ODT algorithm preserves the shape of the original models quite well.

The original ODT has also been extended by Tournois et al. (2009) to 3D tetrahedral mesh smoothing and the method is called Natural ODT (NODT). The NODT method computes the new position of a boundary vertex x_0 in a tetrahedral mesh $\mathcal{T}$ by adding a certain amount of compensation to the weighted centroid of the neighborhood of x_0. The compensation is a weighted sum of the normal vectors of the boundary triangles around x_0. Although boundary vertices are considered in the NODT method, the new positions calculated have to be projected onto the boundary of $\mathcal{T}$ to preserve the volume and shape of the original mesh. Therefore, the NODT method does not optimize the positions for boundary vertices. The smoothing results by using the afore-mentioned NODT method are shown in Table 1, where we can see that our B-ODT algorithm significantly outperforms the NODT method.

Sometimes the results obtained by the NODT method are even worse than the original meshes. The running time of B-ODT and NODT is compared in Tab. 3.

4. Conclusions

We described a method of simultaneously smoothing both inner and boundary vertices of a tetrahedral mesh under a unified optimization framework. The B-ODT algorithm presented can preserve sharp features very well and is guaranteed to preserve the volume of the original mesh. For every boundary vertex, the optimal position is computed by solving a linear system. The algorithm is numerically robust and easy to implement because the order of the linear equation system is only degree 2. Although the vertex insertion operation is integrated into the B-ODT approach to further improve the quality of the mesh, the number of new vertices added is very small compared to the size of the original mesh. The experimental results have shown the effectiveness of the proposed method.

Acknowledgments

The work described was supported in part by an NIH Award (Number R15HL103497) from the National Heart, Lung, and Blood Institute (NHLBI) and by a subcontract from the National Biomedical Computation Resource (NIH Award Number P41 RR08605). The content is solely the responsibility of the authors and does not necessarily represent the official views of the sponsors.

References

Babuska I, Aziz A. On the angle condition in the finite element method. SIAM Journal on Numerical Analysis. 1976; 13:214–226.

Bank R, Smith R. Mesh smoothing using a posteriori error estimates. SIAM Journal on Numerical Analysis. 1997; 34:979–997.

Brewer, M.; Freitag Diachin, L.; Knupp, P.; Leurent, T.; Melander, D. The mesquite mesh quality improvement toolkit. 12th International Meshing Roundtable; 2003. p. 239-250.

Canann S, Stephenson M, Blacker T. Optismoothing: An optimization-driven approach to mesh smoothing. Finite Elements in Analysis and Design. 1993; 13:185–190.

Canann, SA.; Tristano, JR.; Staten, ML. An approach to combined laplacian and optimization-based smoothing for triangular, quadrilateral and quad-dominant meshes. 7th International Meshing Roundtable; 1998. p. 419-494.

Chen, C.; Szema, K.; Chakravarthy, S. Optimization of unstructured grid. 33rd Aerospace Sciences Meeting and Exhibit; Reno, NV. January; 1995. p. 1-10.AIAA 95-0217

Chen, L. Mesh smoothing schemes based on optimal Delaunay triangulations. 13th International Meshing Roundtable; 2004. p. 109-120.

Chen L, Xu J. Optimal delaunay triangulations. Journal of Computational Mathematics. 2004; 22:299–308.

Cheng, SW.; Dey, TK.; Edelsbrunner, H.; Facello, MA.; Teng, SH. Sliver Exudation. Proceedings of the fifteenth annual symposium on Computational geometry; ACM. 1999. p. 1-13.

Chew, LP. Guaranteed-quality Delaunay meshing in 3D. The 13th Annual Symposium on Computational Geometry; 1997. p. 391-393.

Delanaye M, Hirsch C, Kovalev K. Untangling and optimization of unstructured hexahedral meshes. Computational mathematics and mathematical physics. 2003; 43:807–814.

Djidjev, H. Force-directed methods for smoothing unstructured triangular and tetrahedral meshes. 9th International Meshing Roundtable; 2000. p. 395-406.

Du Q, Wang D. Tetrahedral mesh generation and optimization based on centroidal Voronoi tessellations. Int J Numer Meth Engng. 2003; 56:1355–1373.

Escobar J, Montenegro R, Montero G, Rodríguez E, Gonzáez-Yuste J. Smoothing and local refinement techniques for improving tetrahedral mesh quality. Computers & Structures. 2005; 83:2423–2430.

Escobar JM, Rodríguez E, Montenegro R, Montero G, González-Yuste JM. Simultaneous untangling and smoothing of tetrahedral meshes. Computer Methods in Applied Mechanics and Engineering. 2003; 192:2775–2787.

Field D. Laplacian smoothing and Delaunay triangulations. Communications in Applied Numerical Methods. 1988; 4:709–712.

Freitag L, Knupp P. Tetrahedral mesh improvement via optimization of the element condition number. International Journal for Numerical Methods in Engineering. 2002; 53:1377–1391.

Freitag L, Ollivier-Gooch C. Tetrahedral mesh improvement using swapping and smoothing. International Journal for Numerical Methods in Engineering. 1997; 40:3979–4002.

Freitag L, Plassmann P. Local optimization-based simplicial mesh untangling and improvement. International Journal for Numerical Methods in Engineering. 2000; 49:109–125.

Freitag, L.; Plassmann, P. Local optimization-based untangling algorithms for quadrilateral meshes. 10th International Meshing Roundtable; 2001. p. 397-406.

Freitag LA. On combining Laplacian and optimization-based mesh smoothing techniques. Trends in unstructured mesh generation AMD. 1997; 220:37–43.

Freitag Diachin, L.; Knupp, P. Tetrahedral element shape optimization via the jacobian determinant and condition number. 8th International Meshing Roundtable; 1999. p. 247-258.

Hansbo P. Generalized Laplacian smoothing of unstructured grids. Communications in Numerical Methods in Engineering. 1995; 11:455–464.

Herrmann L. Laplacian-isoparametric grid generation scheme. Journal of the Engineering Mechanics Division. 1976; 102:749–907.

Klingner, BM.; Shewchuk, JR. Aggressive tetrahedral mesh improvement. Proceedings of the 16th international meshing roundtable; Springer. 2008. p. 3-23.

Knupp P. Algebraic mesh quality metrics. SIAM journal on scientific computing. 2002; 23:193–218.

Knupp P. Algebraic mesh quality metrics for unstructured initial meshes. Finite Elements in Analysis and Design. 2003; 39:217–241.

Knupp PM. Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I-a framework for surface mesh optimization. International Journal for Numerical Methods in Engineering. 2000a; 48:401–420.

Knupp, PM. Hexahedral mesh untangling & algebraic mesh quality metrics. 9th International Meshing Roundtable; 2000b. p. 173-183.

Knupp PM. Hexahedral and tetrahedral mesh untangling. Engineering with Computers. 2001; 17:261–268.

Labelle F, Shewchuk JR. Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. ACM Transactions on Graphics. 2007; 26:57:1–57:10.

Li, X.; Freitag, LA.; Freitag, LA. Technical Report. Argonne National Laboratory; 1999. Optimization-Based Quadrilateral and Hexahedral Mesh Untangling and Smoothing Techniques.

Menédez-Díaz A, González-Nicieza C, Álvarez Vigil AE. Hexahedral mesh smoothing using a direct method. Computers & Geosciences. 2005; 31:453–463.

Mezentsev, A. A generalized graph-theoretic mesh optimization model. 13th International Meshing Roundtable; 2004. p. 255-264.

Nave D, Chrisochoides N, Chew LP. Guaranteed-quality parallel Delaunay refinement for restricted polyhedral domains. Computational Geometry: Theory and Applications. 2004; 28:191–215.

Ohtake Y, Belyaev A, Bogaevski I. Mesh regularization and adaptive smoothing. Computer-Aided Design. 2001; 33:789–800.

Parthasarathy V, Kodiyalam S. A constrained optimization approach to finite element mesh smoothing. Finite Elements in Analysis and Design. 1991; 9:309–320.

Phillippe, P.; Baker, T. A comparison of triangle quality measures. 10th International Meshing Roundtable; 2001. p. 327-340.

Shewchuk, JR. What is a good linear element? interpolation, conditioning, and quality measures. 11th International Meshing Roundtable; 2002. p. 115-126.

Si H, Gärtner K, Fuhrmann J. Boundary conforming Delaunay mesh generation. Computational Mathematics and Mathematical Physics. 2010; 50:38–53.

Sirois Y, Dompierre J, Vallet MG, Guibault F. Hybrid mesh smoothing based on Riemannian metric non-conformity minimization. Finite Elements in Analysis and Design. 2010; 46:47–60.

Tournois J, Wormser C, Alliez P, Desbrun M. Interleaving delaunay refinement and optimization for practical isotropic tetrahedron mesh generation. ACM Transactions on Graphics. 2009; 28:75:1–75:9.

Vartziotis D, Athanasiadis T, Goudas I, Wipper J. Mesh smoothing using the geometric element transformation method. Computer Methods in Applied Mechanics and Engineering. 2008; 197:3760–3767.

Vartziotis D, Wipper J. A dual element based geometric element transformation method for all-hexahedral mesh smoothing. Computer Methods in Applied Mechanics and Engineering. 2010 In Press, Corrected Proof, –.

Vartziotis D, Wipper J, Schwald B. The geometric element transformation method for tetrahedral mesh smoothing. Computer Methods in Applied Mechanics and Engineering. 2009; 199:169–182.

Xu H, Newman TS. An angle-based optimization approach for 2d finite element mesh smoothing. Finite Elements in Analysis and Design. 2006; 42:1150–1164.

Xu K, Cheng ZQ, Wang Y, Xiong Y, Zhang H. Quality encoding for tetrahedral mesh optimization. Computers & Graphics. 2009; 33:250–261.IEEE International Conference on Shape Modelling and Applications; 2009.

Yu Z, Holst MJ, McCammon JA. High-fidelity geometric modeling for biomedical applications. Finite Elements in Analysis and Design. 2008; 44:715–723.

Zavattieri PD, Dari EA, Buscaglia GC. Optimization strategies in unstructured mesh generation. International Journal for Numerical Methods in Engineering. 1996; 39:2055–2071.

Zhang Y, Hughes TJ, Bajaj CL. An automatic 3D mesh generation method for domains with multiple materials. Computer methods in applied mechanics and engineering. 2010; 199:405–415. [PubMed: 20161555]

Zhou, T.; Shimada, K. An angle-based approach to two-dimensional mesh smoothing. Proceedings of the 9th International Meshing Roundtable; Citeseer. 2000. p. 373-384.

Appendix A. Integral over Ω*

An important observation is that Ω_0 and Ω_* contain almost the same set of vertices except that $x_0 \in \Omega_0$ and $x_* \in \Omega_*$. There is an exact relationship between Ω_0 and Ω_*:

$$\Omega_* = \Omega_0 + \sum_{i=1}^{m} \mathrm{sign}(((x_* - x_0) \times \mathbf{n}_i)_z)\tau_i, \quad \text{(A.1)}$$

where τ_i is a tetrahedron with vertices x_0, x_*, y_i, y_{i+1} and the definitions of $\{y_i, i = 1, \cdots, m\}$ are given in Section 2.2. $\mathbf{n}_i$ is the normal vector of triangle $\Delta x_0 y_i y_{i+1}$ pointing outside of Ω_0. $(\cdot)_z$ is the z coordinate of a vector. We call τ_i a *boundary tetrahedron*. Thus, the integral $\int_{x\in\Omega_*} \|x - x_0\|^2 \mathrm{d}x$ can be rewritten as:

$$\int_{x\in\Omega_*} \|x - x_0\|^2 \mathrm{d}x = \int_{x\in\Omega_0} \|x - x_0\|^2 \mathrm{d}x + \sum_{i=1}^{m} \left(\mathrm{sign}(((x_* - x_0) \times \mathbf{n}_i)_z) \int_{x\in\tau_i} \|x - x_0\|^2 \mathrm{d}x \right) \quad \text{(A.2)}$$

Without loss of generality, we only consider computing the integral of $\|x - x_0\|^2$ over the first boundary tetrahedron, τ_1. For any point x in τ_1, we represent x with respect to the four vertices of τ_1 using the barycentric transformation:

$$x = \lambda_0 x_* + \lambda_1 y_1 + \lambda_2 y_2 + (1 - \lambda_0 - \lambda_1 - \lambda_2) x_0 \quad \text{(A.3)}$$

with $\lambda_i \geq 0$ and $\lambda_0 + \lambda_1 + \lambda_2 \leq 1$. Then we have:

$$\int_{x\in\tau_1}\left\|x-x_0\right\|^2\mathrm{d}x=\iiint\limits_{\substack{0\le\lambda_0\le1\\0\le\lambda_1\le1-\lambda_0\\0\le\lambda_2\le1-\lambda_0-\lambda_1}}\left\|\lambda_0(x_*-x_0)+\lambda_1(y_1-x_0)+\lambda_2(y_2-x_0)\right\|^2\mathrm{d}\lambda_2\mathrm{d}\lambda_1\mathrm{d}\lambda_0 \quad \text{(A.4)}$$

The integrated function is a quadratic polynomial on λ_i and thus can be exactly computed.

Let $X_* = x_* - x_0,\ Y_1 = y_1 - x_0,\ Y_2 = y_2 - x_0$ and the integral becomes:

$$\int_{x\in\tau_1}\left\|x-x_0\right\|^2\mathrm{d}x=\frac{1}{60}((X_*)^2+(Y_1)^2+(Y_2)^2+X_*Y_1+X_*Y_2+Y_1Y_2)\cdot|\det(X_*,Y_1,Y_2)| \quad \text{(A.5)}$$

Note that the $\det(X_*,\ Y_1,\ Y_2)$ has the same sign as $((x_* - x_0)\times\mathbf{n}_1)_z$, yielding the final formula as follows:

$$\int_{x\in\Omega_*}\left\|x-x_0\right\|^2\mathrm{d}x=\int_{x\in\Omega_0}\left\|x-x_0\right\|^2\mathrm{d}x+\frac{1}{60}\sum_{i=1}^{m}(X_*^2+Y_i^2+Y_{i+1}^2+X_*Y_i+X_*Y_{i+1}+Y_iY_{i+1})\det(X_*,Y_i,Y_{i+1}) \quad \text{(A.6)}$$

Appendix B. Volume preservation by tangent plane constraint

We shall prove that the constraint of limiting x_* on the tangent plane can guarantee the volume preservation of the neighborhood of x_0, i.e. $|\Omega_*| \equiv |\Omega_0|$. We take the following commonly-used formulae to compute the normal of the tangent plane at x_0:

$$\mathbf{n}=\sum_{i=1}^{m}S_i\mathbf{n}_i, \quad \text{(B.1)}$$

where $\mathbf{n}$ is the normal vector of the tangent plane at x_0, S_i and $\mathbf{n}_i$ are the area and unit normal vector of the boundary triangle $\Delta x_0y_iy_{i+1}$, respectively. Again, refer to Section 2.2 for the definitions of $\{y_i,\ i = 1, \cdots, m\}$.

Given x_*, the volume of Ω_* can be computed by adding all the volume of tetrahedra in Ω_*. For any tetrahedron τ in Ω_*, $|\tau|=\frac{1}{3}S_\tau<\mathbf{n}_\tau,x_*-x_\tau>$, $x_* - x_\tau>$, where S_τ and $\mathbf{n}_\tau$ are the area and unit normal vector of the triangle τ_τ opposite to x_* in τ and x_τ is the barycenter of t_τ. Thus, we have the following formula for the volume of Ω_*:

$$|\Omega_*|=\sum_{\tau\in\Omega_*}\left(\frac{1}{3}S_\tau<\mathbf{n}_\tau,x_*-x_\tau>\right)=<\sum_{\tau\in\Omega_*}\frac{1}{3}S_\tau\mathbf{n}_\tau,x_*-x_0>-\mathscr{C}, \quad \text{(B.2)}$$

where $\mathscr{C}$ is a constant independent of x_*. According to (B.2), if $|\Omega_*|$ is constant (independent of x_*), then we must have

$$<\sum_{\tau\in\Omega_*}\frac{1}{3}S_\tau\mathbf{n}_\tau,x_*-x_0>=\overline{\mathscr{C}} \quad \text{(B.3)}$$

That means x_* must be on a plane that passes through x_0 with $\sum_{\tau\in\Omega_*}\frac{1}{3}S_\tau\mathbf{n}_\tau$ being the normal vector. Therefore, we only need to prove that the normal vector of this plane is identical to that of the tangent plane at x_0 given in (B.1). Note that S_τ and $\mathbf{n}_\tau$ are independent of x_*, we only need to prove the follow equation (the coefficient $\frac{1}{3}$ is omitted):

$$\sum_{i=1}^{m}S_i\mathbf{n}_i=\sum_{\tau\in\Omega_0}S_\tau\mathbf{n}_\tau \quad \text{(B.4)}$$

For any tetrahedron τ, suppose s_i and $\mathbf{m}_i$ are the areas and unit normal vectors of the four triangles of τ, we have

$$\sum_{i=1}^{4}s_i\mathbf{m}_i=0 \quad \text{(B.5)}$$

Putting together all the tetrahedra in Ω_0, we have

$$\sum_{\tau\in\Omega_0}\left(\sum_{i=1}^{4}s_{\tau,i}\mathbf{m}_{\tau,i}\right)=0. \quad \text{(B.6)}$$

Note that the triangles shared by two adjacent tetrahedra in Ω_0 are cancelled in (B.6) because of the same area but opposite normal vectors in the two tetrahedra. The remaining triangles in (B.6) exactly give rise to (B.4).

Appendix C. Simplification of the optimization problem

We have the following coefficients for the cubic items in (9):

$$u^3=\langle\mathbf{s},\frac{1}{60}\sum_{i=1}^{m}Y_i\times Y_{i+1}\rangle$$
$$v^3=\langle\mathbf{t},\frac{1}{60}\sum_{i=1}^{m}Y_i\times Y_{i+1}\rangle$$
$$u^2v=uv^2=u^3+v^3$$

Note that $Y_i = y_i - x_0$ and $Y_i\times Y_{i+1} = 2S_i\mathbf{n}_i$, hence $\sum_{i=1}^{m}Y_i\times Y_{i+1}$ is orthogonal to the tangent plane at x_0. Therefore, the inner products of $\sum_{i=1}^{m}Y_i\times Y_{i+1}$ and $\mathbf{s}$, $\mathbf{t}$ are zeros, which means that u^3, v^3, u^2v and uv^2 are all zeros.

We rewrite the objective function in (5) as:

$$Error_*=Eu^2+Fv^2+Guv+Hu+Iv,$$

where the coefficients E, F, G, H, I have the forms as given in Algorithm 2. The gradient of $Error_*$ is $(2Eu+Gv+H, 2Fv+Gu+I)$ and the Hessian matrix of $Error_*$ is $\begin{pmatrix}2E & G\\ G & 2F\end{pmatrix}$.

Thus we can see that the optimization problem has a unique solution as long as $4EF > G^2$. In all the examples we have experimented so far, this condition is always satisfied. But a more theoretical analysis of whether or not this condition is guaranteed is part of our future work.

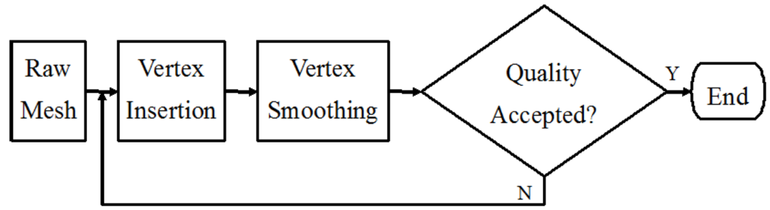

Figure 1.
The framework of our mesh quality improvement method

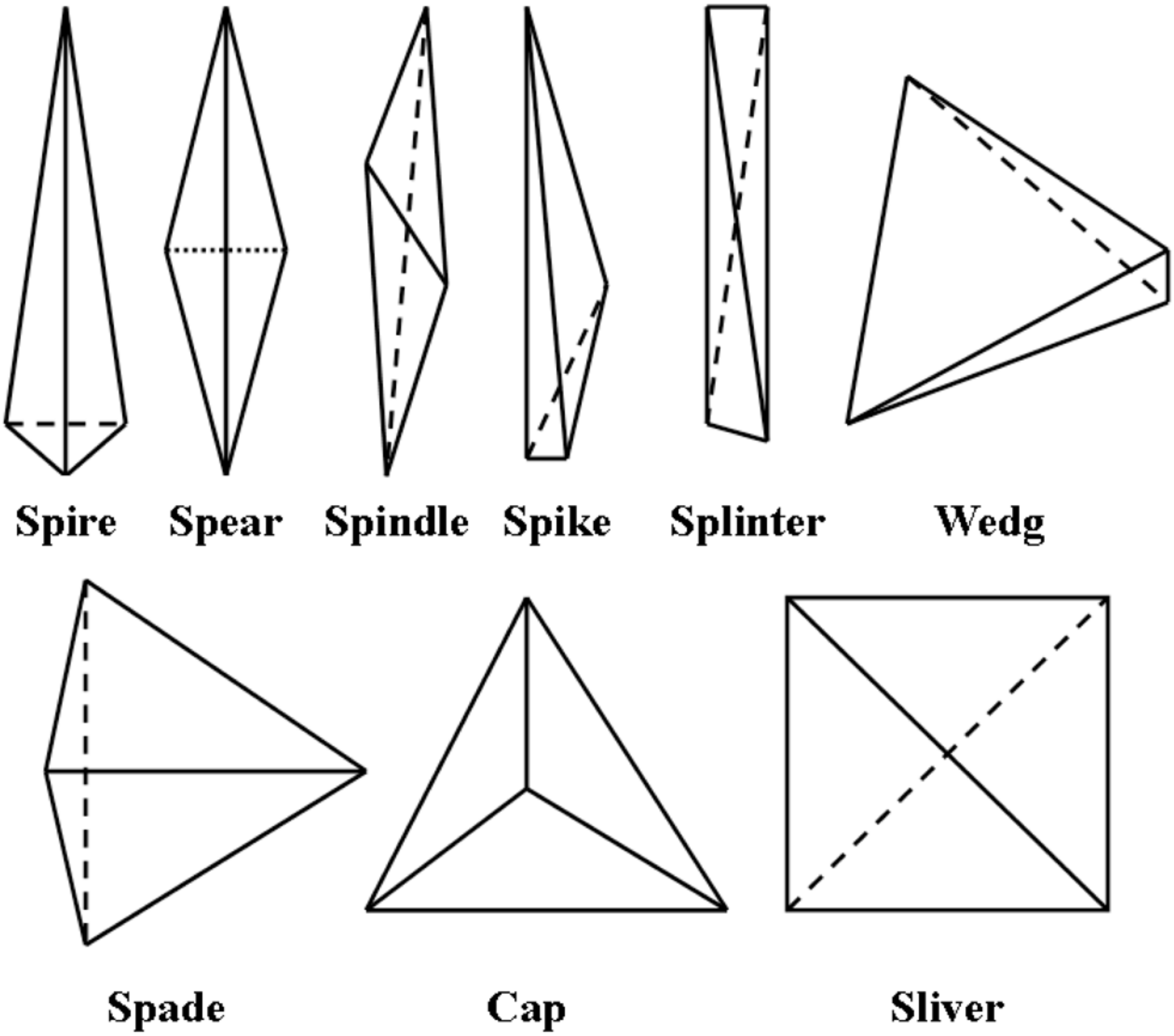

Figure 2.
Types of bad tetrahedron with too large and/or too small dihedral angles (Cheng et al., 1999)

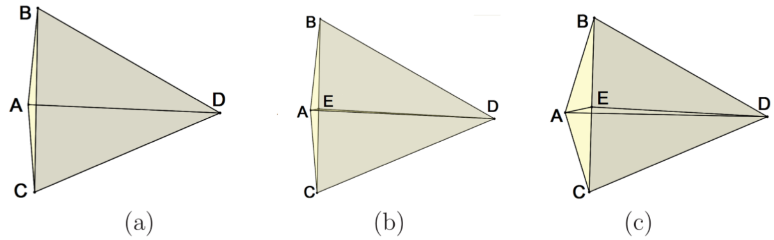

Figure 3.
Vertex insertion and smoothing for a spade

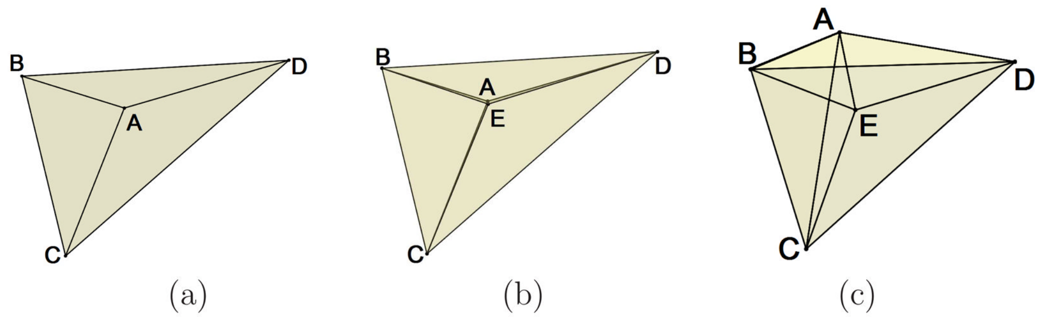

Figure 4.
Vertex insertion and smoothing for a cap

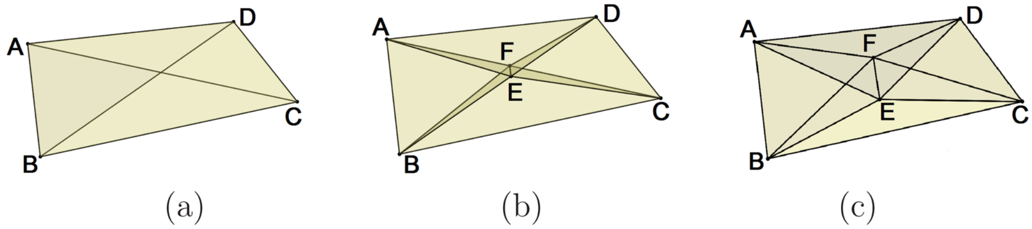

Figure 5.
Vertex insertion and smoothing for a sliver

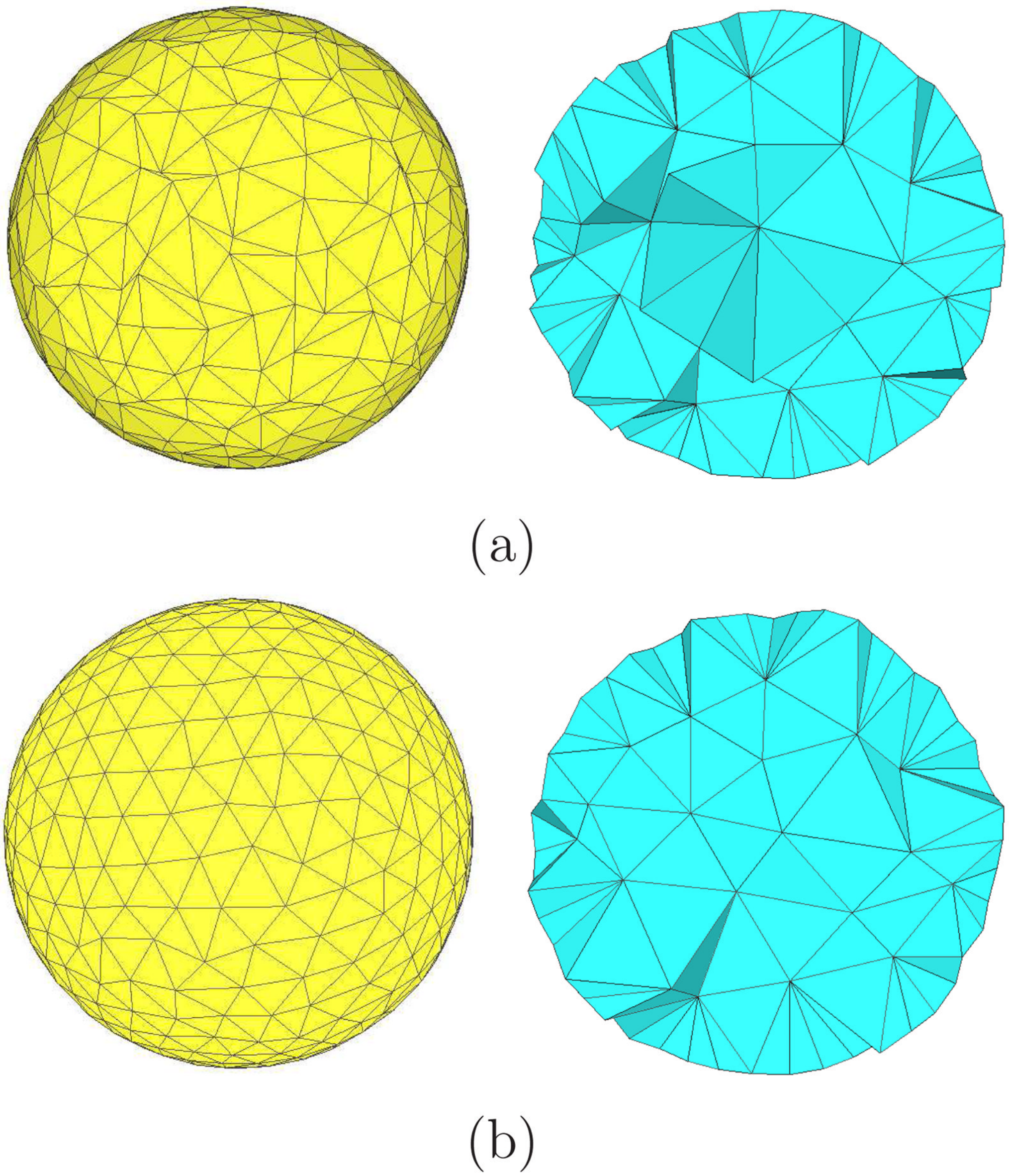

Figure 6.
The original mesh model (a) and the smoothed result (b). In both meshes, the outer and cross-section views are shown. The minimum dihedral angles of these two meshes are 5.86° and 15.20° respectively, and the maximum dihedral angles are 164.70° and 150.25° respectively.

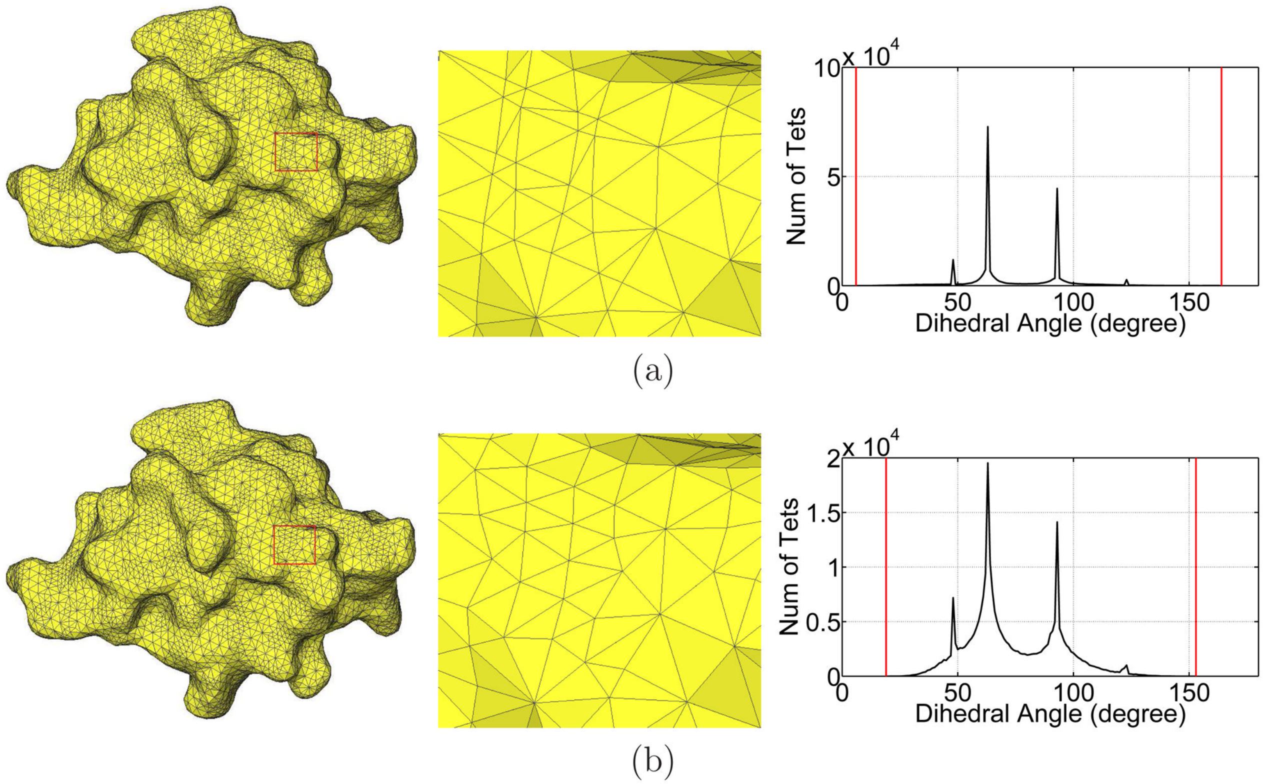

Figure 7.
Original and smoothed 2CMP models. The minimum dihedral angles of these two meshes are 5.57° and 18.10° respectively, and the maximum dihedral angles are 163.24° and 152.66° respectively.

$watermark-text
$watermark-text

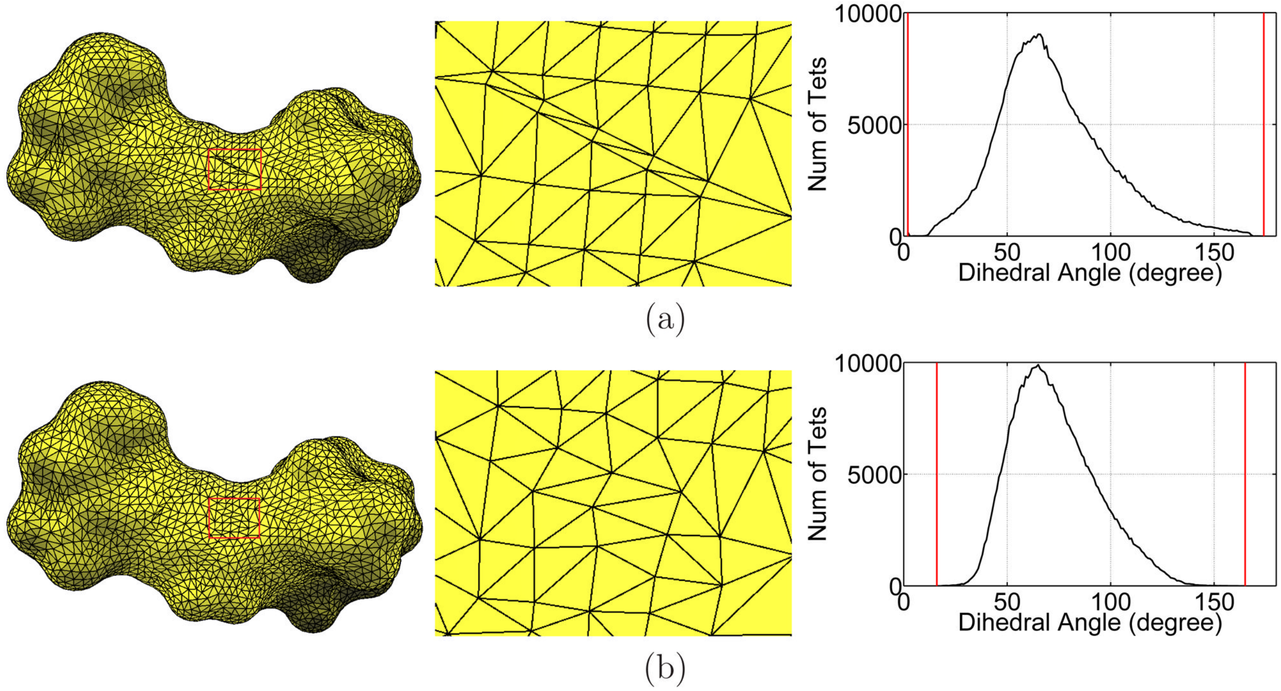

Figure 8.
Original and smoothed Retinal models. The minimum dihedral angles of these two meshes are 1.25° and 15.10° respectively, and the maximum dihedral angles are 173.85° and 164.58° respectively.

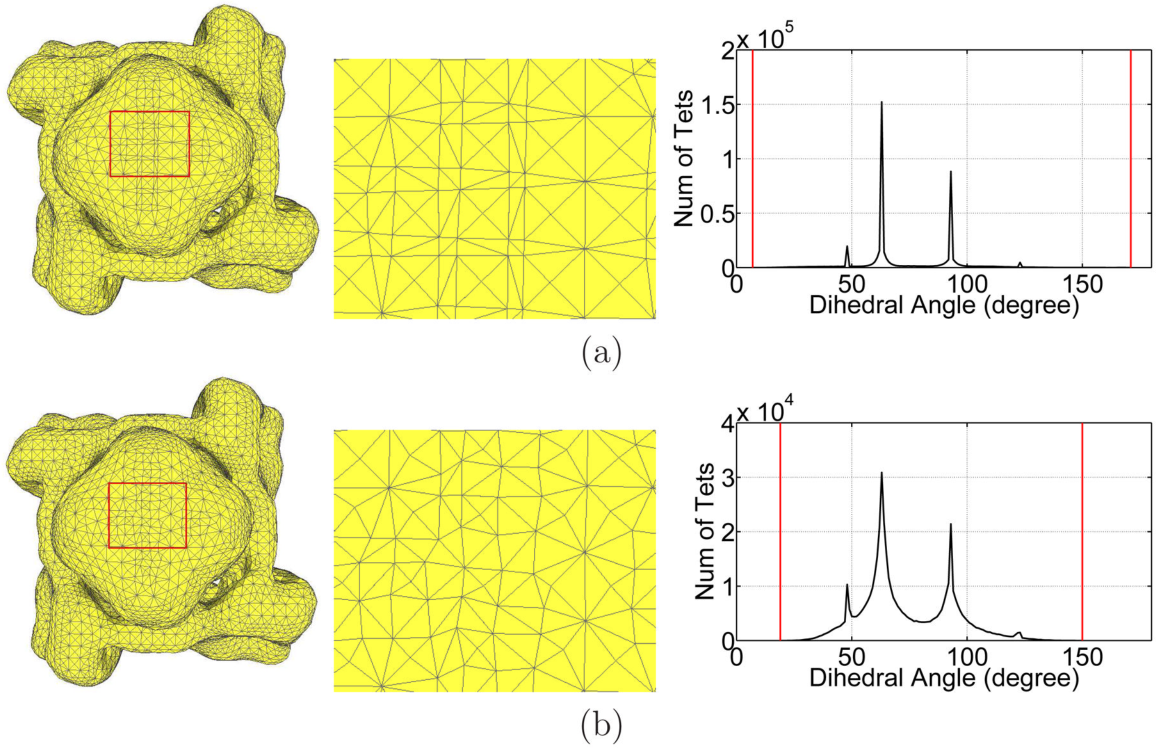

Figure 9.
Original and smoothed RyR models. The minimum dihedral angles of these two meshes are 6.19° and 18.52° respectively, and the maximum dihedral angles are 170.74° and 149.25° respectively.

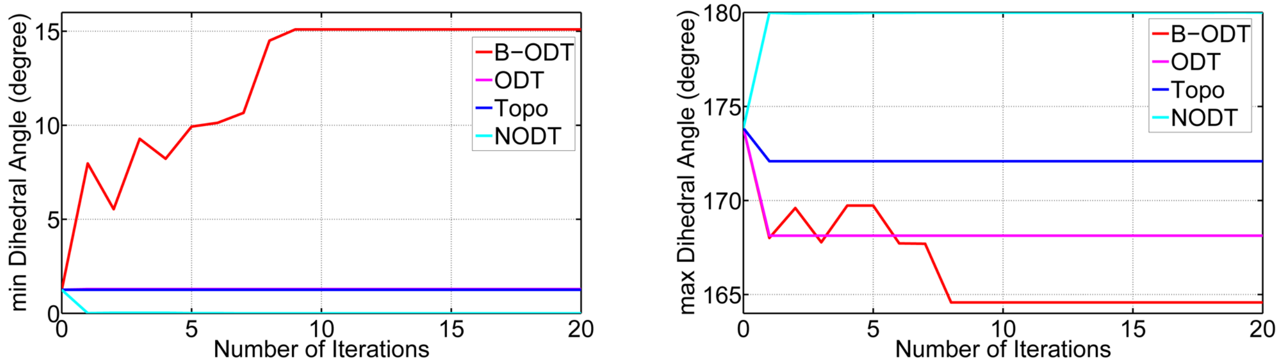

Figure 10.
The convergence of minimum and maximum dihedral angles with respect to the number of iterations on the Retinal model using the B-ODT algorithm. Note that on the left the curves of ODT and topology optimization are almost identical.

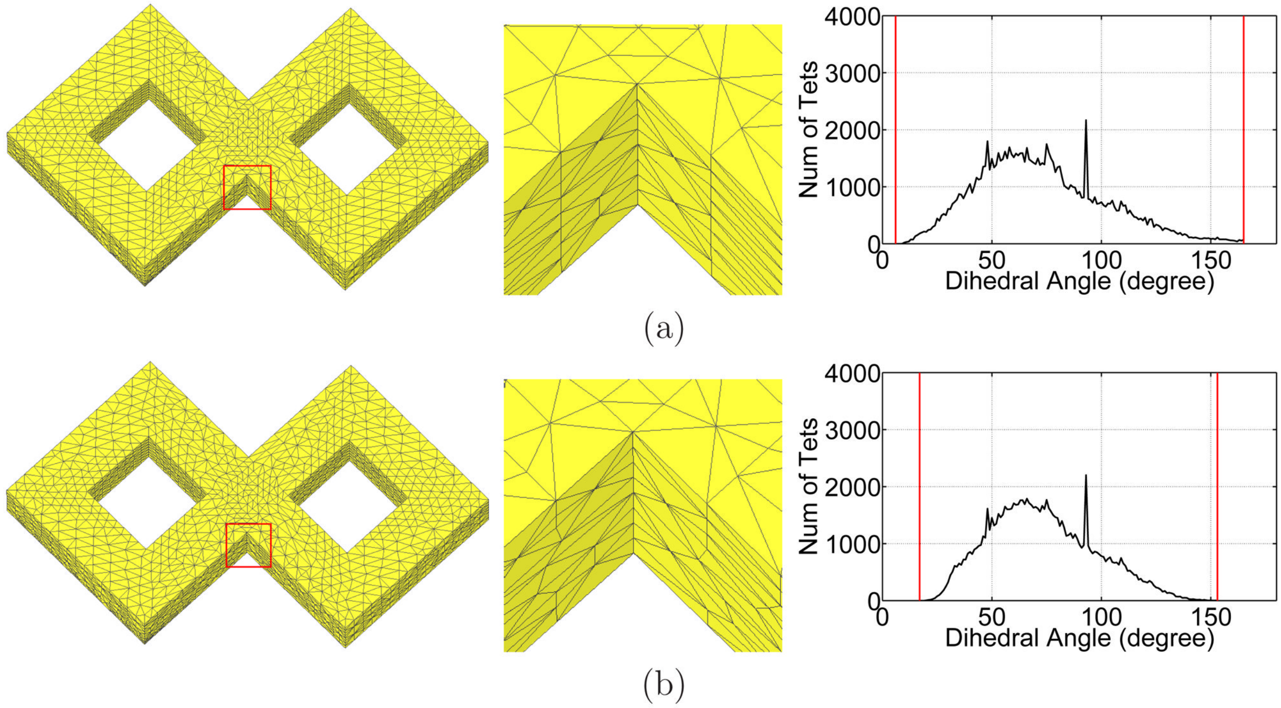

Figure 11.
Original and smoothed 2Torus models. The minimum dihedral angles of these two meshes are 5.96° and 16.92° respectively, and the maximum dihedral angles are 164.92° and 152.05° respectively.

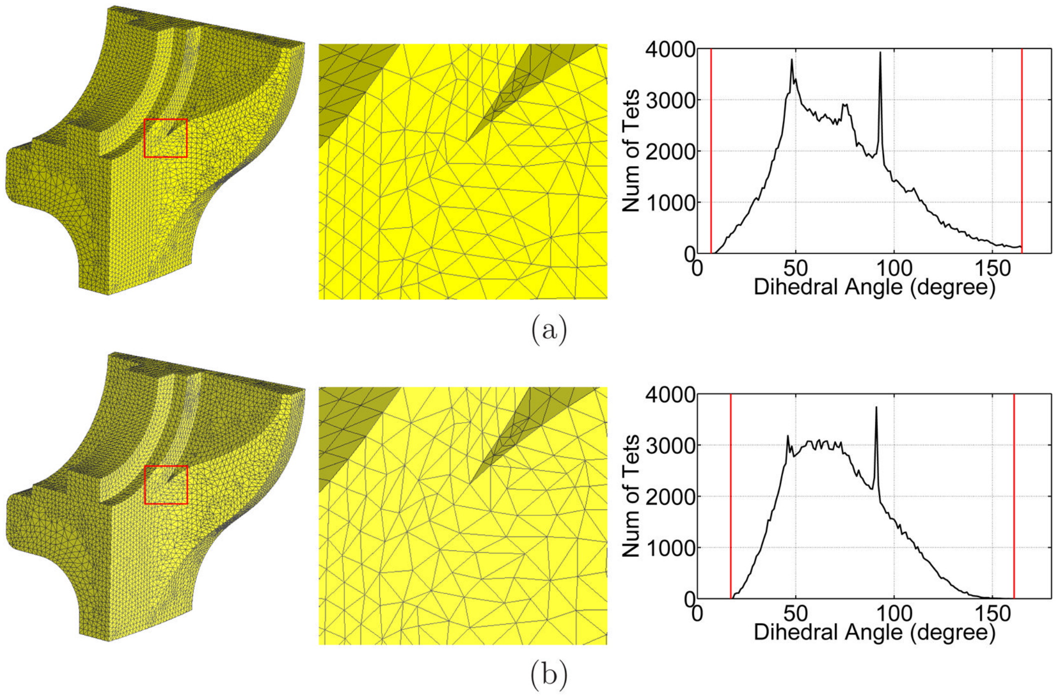

Figure 12.
Original and smoothed FanDisk models. The minimum dihedral angles of these two meshes are 6.04° and 16.80° respectively, and the maximum dihedral angles are 164.98° and 160.53° respectively.

Table 1

Comparisons of Dihedral Angles using Different Methods

Model		Vertex Number	min Angle	max Angle
Random Sphere	Original Mesh	729	5.86°	164.70°
	B-ODT	731	15.20°	150.25°
	ODT	729	6.28°	162.46°
	Topology Optimization	729	5.86°	164.70°
	NODT	729	6.21°	173.64°
2Cmp	Original Mesh	10415	5.57°	163.24°
	B-ODT	10415	18.10°	152.66°
	ODT	10415	11.64°	158.06°
	Topology Optimization	10415	5.57°	163.24°
	NODT	10415	10.70°	157.19°
Retinal	Original Mesh	14921	1.25°	173.85°
	B-ODT	14948	15.10°	164.58°
	ODT	14921	1.29°	168.13°
	Topology Optimization	14921	1.25°	172.09°
	NODT	14921	0.00°	179.99°
RyR	Original Mesh	18585	6.19°	170.74°
	B-ODT	18585	18.52°	149.25°
	ODT	18585	10.34°	158.32°
	Topology Optimization	18585	6.19°	170.74°
	NODT	18585	7.78°	162.74°
2Torus	Original Mesh	4635	5.96°	164.92°
	B-ODT	4656	16.92°	152.05°
	ODT	4635	9.46°	157.53°
	Topology Optimization	4635	6.85°	164.75°
	NODT	4635	0.01°	179.98°
FanDisk	Original Mesh	9131	6.04°	164.98°
	B-ODT	9162	16.80°	160.53°
	ODT	9131	9.59°	163.53°
	Topology Optimization	9131	6.78°	164.98°
	NODT	9131	0.08°	179.86°

Table 2

Relative Hausdorff Distance between Original and Smoothed Meshes

Models	Random Sphere	2Cmp	Retinal	RyR	2Torus	FanDisk
Rel. Hausdorff Distance	0.37%	1.00%	0.63%	0.88%	0.15%	0.28%

Table 3

Comparison of Running Time (20 iterations)

	Random Sphere	2Cmp	Retinal	RyR	2Torus	FanDisk
B-ODT	13.94s	203.11s	382.53s	369.70s	92.14s	168.57s
NODT	11.85s	159.80s	323.12s	291.24s	64.08s	124.24s

Algorithm 1

Original ODT for Smoothing Inner Vertices

for every inner vertex x_0 **do**

1. **For every** adjacent tetrahedron τ **do**

   Compute S_τ, $\mathbf{n}_\tau$ and $\|x_{\tau,i} - x_0\|^2$

2. Sum up all the values of $\frac{1}{3} S_\tau \mathbf{n}_\tau \sum_{k=1}^{3} \left\| x_{\tau,k} - x_0 \right\|^2$

3. Compute the volume of Ω_0, i.e. $|\Omega_0|$

4. Compute x_* using (3)

Algorithm 2

Basic B-ODT Smoothing for Boundary Vertices with Volume Perserving

---

for every boundary vertex x_0 **do**

1 Compute the the normal vector of the tangent plane at x_0, then select two orthogonal unit vectors **s**, **t** on the tangent plane.

2 Compute the following coefficients:

i. $$E=\frac{1}{4}|\Omega_0|-\frac{1}{60}\sum_{i=1}^{m}(<\mathbf{s}, Y_i+Y_{i+1}><\mathbf{s}, Y_i \times Y_{i+1}>)$$

ii. $$F=\frac{1}{4}|\Omega_0|-\frac{1}{60}\sum_{i=1}^{m}(<\mathbf{t}, Y_i+Y_{i+1}><\mathbf{t}, Y_1 \times Y_2>)$$

iii. $$G=-\frac{1}{60}\sum_{i=1}^{m}(<\mathbf{s}, Y_i+Y_{i+1}><\mathbf{t}, Y_i \times Y_{i+1}>+<\mathbf{t}, Y_i+Y_{i+1}><\mathbf{s}, Y_i \times Y_{i+1}>)$$

iv. $$H=\frac{1}{12}<\mathbf{s}, \sum_{\tau\in\Omega_*} S_\tau \mathbf{n}_\tau L_\tau>-\frac{1}{60}\sum_{i=1}^{m}(Y_i^2+Y_{i+1}^2+Y_iY_{i+1})<\mathbf{s}, Y_i \times Y_{i+1}>$$

v. $$I=\frac{1}{12}<\mathbf{t}, \sum_{\tau\in\Omega_*} S_\tau \mathbf{n}_\tau L_\tau>-\frac{1}{60}\sum_{i=1}^{m}(Y_i^2+Y_{i+1}^2+Y_iY_{i+1})<\mathbf{t}, Y_i \times Y_{i+1}>$$

where $L_\tau=\sum_{j=1}^{3}\left\|x_{\tau,j}-x_0\right\|^2$, $(\cdot \times \cdot)$ is the cross product operation.

3 Solve the following degree-2 linear equation system:

$$\begin{bmatrix} 2E & G \\ G & 2F \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}=\begin{bmatrix} -H \\ -I \end{bmatrix} \quad (10)$$

4 The solution of (10) gives rise to the optimal solution of x_* as $x_* = x_0 + u\mathbf{s} + v\mathbf{t}$.

Algorithm 3

Advanced B-ODT Smoothing for Boundary Vertices with Feature Preserving

for every boundary vertex x_0 **do**

1. Compute the feature line of x_0, suppose the unit vector of this line is **d**.
2. Compute the following coefficients:
   - i. $$A=\frac{1}{4}|\Omega_0|-\frac{1}{60}\sum_{i=1}^{m}<\mathbf{d}, Y_i+Y_{i+1}><\mathbf{d}, Y_i \times Y_{i+1}>$$
   - ii. $$B=\frac{1}{12}<\mathbf{d}, \sum_{\tau\in\Omega_*} S_\tau \mathbf{n}_\tau L_\tau>-\frac{1}{60}\sum_{i=1}^{m}(Y_i^2+Y_{i+1}^2+Y_iY_{i+1})<\mathbf{d}, Y_i \times Y_{i+1}>$$
3. Compute x_* as $x_* = x_0 + f\mathbf{d}$ with $f=-\frac{B}{2A}$.