

Sliver-suppressing tetrahedral mesh optimization with gradient-based shape matching energy



Saifeng Ni^a, Zichun Zhong^b, Yang Liu^c, Wenping Wang^d, Zhonggui Chen^e, Xiaohu Guo^{a,*}

^a Department of Computer Science, University of Texas at Dallas, Richardson, TX 75083, USA

^b Department of Computer Science, Wayne State University, Detroit, MI 48202, USA

^c Internet Graphics Group, Microsoft Research Asia, Beijing 100080, China

^d Department of Computer Science, The University of Hong Kong, Pok Fu Lam, Hong Kong

^e Department of Computer Science, Xiamen University, Xiamen 361005, China

ARTICLE INFO

Article history:

Received 23 January 2017

Accepted 16 February 2017

Available online 27 February 2017

Keywords:

Sliver-suppressing

Shape matching

Tetrahedral meshing

ABSTRACT

In this paper, a novel shape matching energy is proposed to suppress slivers for tetrahedral mesh generation. Given a volumetric domain with a user-specified template (regular) simplex, the tetrahedral meshing problem is transformed into a shape matching formulation with a gradient-based energy, i.e., the gradient of linear shape function. It effectively inhibits small heights and suppresses all the badly-shaped tetrahedrons in tetrahedral meshes. The proposed approach iteratively optimizes vertex positions and mesh connectivity, and makes the simplices in the computed mesh as close as possible to the template simplex. We compare our results qualitatively and quantitatively with the state-of-the-art algorithm in tetrahedral meshing on extensive models using the standard measurement criteria.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Triangular and tetrahedral meshes are most basic 2D and 3D elements in computer graphics and geometric modeling fields. The mesh generation is to discretize spatial domain into a set of connected but non-overlapped simplex elements. Mesh quality highly depends on the size and shape of each element. In this paper, we focus on the tetrahedral mesh generation. There are several measurements for tetrahedral mesh quality, and dihedral angle is one of the most important criteria (Alliez et al., 2005; Guo et al., 2016), since badly-shaped tetrahedrons with tiny dihedral angles (i.e., sliver) can severely affect numerical simulation (Shewchuk, 2002a).

Essentially, simplex meshes are used to form a piecewise linear approximation of function $u(\mathbf{x})$ to represent the given shapes. There are several ways to describe the approximation error. Optimal Delaunay Triangulation (ODT) (Chen and Xu, 2004) was proposed to minimize the L^p norm of the difference over the domain Ω between the target function $\hat{u}(\mathbf{x})$ and the interpolated function $u(\mathbf{x})$: $E(\mathbf{x}) = \int_{\Omega} \|\hat{u}(\mathbf{x}) - u(\mathbf{x})\|_{L^p} d\mathbf{x}$. Interpolation error is an important mesh quality measurement. However, the definition of ODT determines that it cannot avoid sliver in tetrahedral mesh. A sliver with close to zero volume still has small interpolation error. So minimizing interpolation error cannot avoid sliver.

* Corresponding author.

E-mail address: xguo@utdallas.edu (X. Guo).

Consider the L^p norm of the difference between the gradient of the target function $\nabla \hat{u}(\mathbf{x})$ and the gradient of the interpolated function $\nabla u(\mathbf{x})$: $E(\mathbf{x}) = \int_{\Omega} \|\nabla \hat{u}(\mathbf{x}) - \nabla u(\mathbf{x})\|_{L^p} d\mathbf{x}$, the gradient error can be strongly affected by the shape of the elements as well as their sizes. Once one dihedral angle approaches 0° or 180° in the tetrahedral mesh, the gradient error will grow dramatically large. In this paper, we propose a shape matching framework and design a gradient-based energy (i.e., the gradient of linear shape function), which heavily punish slivers in tetrahedral meshes. By specifying a template simplex, e.g. a regular simplex, the idea of the proposed method is to make the shape of the to-be-optimized simplex as close as possible to the shape of the template simplex. The experiment results show that our proposed energy has high effectiveness in sliver suppression compared with all other state-of-the-art methods in the tetrahedral meshing.

2. Related work

Tetrahedral mesh generation has been studied for several decades (Owen, 1998), and these approaches can be categorized as advancing front methods (Ito et al., 2004; Schöberl, 1997; Li et al., 2000), octree-based methods (Labelle and Shewchuk, 2007; Neil Molino et al., 2003), Delaunay-based tetrahedral methods (Cheng et al., 2012), Poisson-disk sampling methods (Guo et al., 2016), etc.

We will focus on Delaunay-based tetrahedral meshing approaches, since our proposed sliver-suppressing optimization method is most relevant to this category. Generally speaking, there are two groups of Delaunay-based methods: (1) Delaunay refinement-based methods (Chew, 1997; Jamin et al., 2015; Si, 2015), which gradually insert new vertices to improve the mesh quality until meeting certain user-specified stopping conditions, such as dihedral angle, edge length, tetrahedral quality, etc. They are good to provide the theoretical guarantees of mesh quality, but are difficult to control the number of mesh vertices explicitly, and do not optimize the tetrahedral shapes globally. And also the resulting mesh usually has large variance on the element volume (2) Variational-based optimization methods, including Centroidal Voronoi Tessellation (CVT) (Du et al., 1999), ODT (Chen and Xu, 2004), and Particle-based methods (Zhong et al., 2013). Each of these methods defines an energy function, which is minimized through numerical optimization to reach a good-quality mesh result. CVT-based methods (Du and Wang, 2003; Du and Wang, 2005) have been well studied in computing meshes. However, it may generate a large number of slivers in 3D tetrahedral mesh (Yan et al., 2010), it caused by that CVT calculates the dual meshes of Voronoi cells, instead of computing the shape of tetrahedrons. ODT-based methods (Alliez et al., 2005; Chen et al., 2014) perform better in terms of suppressing slivers comparing to CVT-based methods. However, it still cannot completely avoid slivers. Comparing to CVT and ODT methods, the recent Particle-based mesh optimization method (Zhong et al., 2013) does not require to optimize the mesh connectivity during the optimization. This property makes it easy and fast to converge, but the lack of considering mesh structure leads to many slivers in the final tetrahedral mesh. It is noted that all of three variational-based optimization methods cannot eliminate badly-shaped elements (e.g., slivers) completely, so a post process for improving the mesh quality is necessary, such as sliver exudation (Cheng et al., 2000), aggressive improvement (Klingner and Shewchuk, 2007), vertex perturbation (Tournois et al., 2009). In this paper, we present a gradient-based shape matching energy to effectively suppress slivers, either as a standalone optimization method, or as postprocessing for existing optimization techniques.

3. Shape matching triangulation energy

Our mesh optimization is illustrated in an algebraic framework, and we call it *shape matching*. Given a template d -simplex $\hat{\tau}_d$, our target is to make any d -simplex τ_d in the mesh as similar as possible to $\hat{\tau}_d$. Here “similar” means the same shape as well as the same sizing factor comparing with the template simplex. If the template is set as a regular d -simplex, any d -simplex in the mesh is expected to be a regular simplex with a constant sizing factor conforming to the defined template simplex, i.e., all simplices in the mesh are endowed with the same shape and the same size.

The shape matching idea is straightforward, but the tricky part is how to well represent the difference between the to-be-optimized simplex τ_d and the template simplex $\hat{\tau}_d$. As the difference is being minimized, the to-be-optimized simplex will become closer and closer to the template simplex. A good difference representation should be scale-sensitive, orientation-free, and also well encoding the shape information. Mathematically, a d -simplex embedded in d -dimension can be defined by a $d \times d$ matrix. Once the template simplex and the to-be-optimized simplex are both represented by $d \times d$ matrices, an affine mapping can be utilized to build the relationship between those two simplices with another $d \times d$ matrix, i.e., the Jacobian between those two simplices. If the affine mapping is an identity matrix, the to-be-optimized simplex is exactly equal to the template simplex. If the affine mapping is a rotation matrix, the to-be-optimized simplex is also equal to the template simplex. In this paper, the squared Frobenius norm of the affine transformation matrix is used to measure the difference between the to-be-optimized simplex and the template simplex, since Frobenius norm is invariant under rotations and also it keeps the sizing information. Then by minimizing the summation of all differences over the entire mesh, the optimal solution of isotropic mesh will be reached when all to-be-optimized simplices are regular and of the same size.

The shape matching framework can be generalized for any d -simplex mesh. A 0-simplex is a vertex $\mathbf{v}_i$; a 1-simplex is an edge $\mathbf{e}_{ij} = \mathbf{v}_j - \mathbf{v}_i$, $0 \leq i < j \leq d$; a 2-simplex is a triangle; and a 3-simplex is a tetrahedron, etc. In addition, the shape matching framework can be easily extended to solve adaptive and anisotropic meshing problems. By mapping the to-be-optimized simplex from Riemannian metric space to Euclidean space, then we still use the regular simplex as template. The computations after that are the same as the uniform isotropic case. The shape matching framework is extendable and

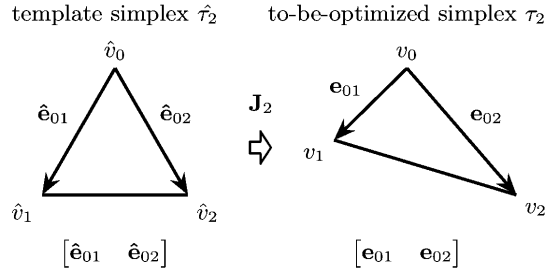


Fig. 1. The illustration of edge-based shape matching for a 2-simplex.

flexible, and it provides freedom to design different mesh element shapes based on different applications. Both isotropic and anisotropic meshing are demonstrated in the experiment section.

In this paper, our target is to remove badly-shaped elements, e.g., slivers, in the isotropic tetrahedral mesh. We will interpret our energy under the proposed shape matching framework. Compared with the traditional *Edge-based Shape Matching (ESM)* in Sec. 3.1, the gradient of the linear shape function is used as basis to represent a simplex, and we called it *Gradient-based Shape Matching (GSM)* in Sec. 3.2.

3.1. Edge-based Shape Matching (ESM)

The traditional methods (Knupp, 2001) used d edge vectors from one vertex as column vectors which form a $d \times d$ matrix $\mathbf{T}_d$ to represent a d -simplex τ_d . Given a matrix $\mathbf{T}_d$, the corresponding d -simplex is uniquely defined. Suppose $\hat{\mathbf{T}}_d$ represents the template d -simplex $\hat{\tau}_d$, then the affine transformation $\mathbf{J}_d$ satisfies $\mathbf{J}_d = \mathbf{T}_d \hat{\mathbf{T}}_d^{-1}$. Squared Frobenius norm of this affine transformation $\|\mathbf{J}_d\|_F^2$ is utilized to measure the difference between $\hat{\tau}_d$ and τ_d . This difference measurement is called shape matching energy E_{τ_d} . Summing up shape matching energies of all simplices, we obtain the energy for the entire d -simplex mesh $E_d = \sum_{\tau_d \in \mathcal{T}} E_{\tau_d}$, where $\mathcal{T}$ is the set of d -simplices in the mesh.

The shape matching framework works for any d -simplex. For the simplicity of illustration, we demonstrate the basic idea using a 2-simplex (i.e., a triangle) in Fig. 1.

Any 2-simplex τ_2 is represented as a matrix formed by two edge vectors $\mathbf{T}_2 = [\mathbf{e}_{01} \ \mathbf{e}_{02}]$, where $\mathbf{e}_{01} = \mathbf{v}_1 - \mathbf{v}_0$ and $\mathbf{e}_{02} = \mathbf{v}_2 - \mathbf{v}_0$. The determinant of $\mathbf{T}_2$ is proportional to the triangle area $|\tau_2|$. So matrix $\mathbf{T}_2$ encodes the shape as well as the size of the triangle.

The affine transformation $\mathbf{J}_2$ between $\hat{\tau}_2$ and τ_2 is expressed as:

$$\mathbf{J}_2 [\hat{\mathbf{e}}_{01} \ \hat{\mathbf{e}}_{02}] = [\mathbf{e}_{01} \ \mathbf{e}_{02}]. \quad (1)$$

If the template $\hat{\tau}_2$ is a regular triangle with edge length $\hat{a}$, the shape matching energy $E_{\tau_2}^{esm}$ of a simplex τ_2 can be simplified as

$$E_{\tau_2}^{esm} = \|\mathbf{J}_2\|_F^2 = \text{trace}(\mathbf{J}_2^T \mathbf{J}_2) = \frac{2}{3\hat{a}^2} \sum_{0 \leq i < j \leq 2} \mathbf{e}_{ij}^T \mathbf{e}_{ij}. \quad (2)$$

When $d = 3$, the shape matching energy $E_{\tau_3}^{esm}$ from a simplex τ_3 to a regular tetrahedron $\hat{\tau}_3$ is represented by the affine transformation $\mathbf{J}_3$ as

$$E_{\tau_3}^{esm} = \text{trace}(\mathbf{J}_3^T \mathbf{J}_3) = \frac{1}{2\hat{a}^2} \sum_{0 \leq i < j \leq 3} \mathbf{e}_{ij}^T \mathbf{e}_{ij}, \quad (3)$$

where $\hat{a}$ is the edge length of $\hat{\tau}_3$.

The above definitions (2-simplex and 3-simplex) are based on edge vectors, so we call them Edge-based Shape Matching (ESM). The main disadvantage of ESM is that it cannot avoid slivers in tetrahedral mesh. The key reason is that slivers may have large edge lengths but close-to-zero heights. Based on this observation, Gradient-based Shape Matching (GSM) energy is proposed, which can be used to effectively suppress slivers.

3.2. Gradient-based Shape Matching (GSM)

For a d -simplex τ_d , barycentric coordinate ω_i corresponding to each vertex i is used as the linear shape function. Any point $\mathbf{v}$ inside the simplex satisfies $\mathbf{v} = \sum_{i=0}^d \omega_i \mathbf{v}_i$, where $\sum_{i=0}^d \omega_i = 1$. If we use the mesh to approximate certain data function $u(\mathbf{x})$, then the data function value at the vertex is given. Suppose the data value at vertex $\mathbf{v}_i$ of τ_d is $u(\mathbf{v}_i)$, which is a constant. Then the data function value at $\mathbf{v}$ is $u(\mathbf{v}) = \sum_{i=0}^d \omega_i u(\mathbf{v}_i)$. The gradient of the data function can be written as $\nabla u(\mathbf{v}) = \sum_{i=0}^d \nabla \omega_i u(\mathbf{v}_i)$. $\nabla \omega_i$ corresponding to vertex $\mathbf{v}_i$ of a d -simplex is a constant vector. The direction of $\nabla \omega_i$ is

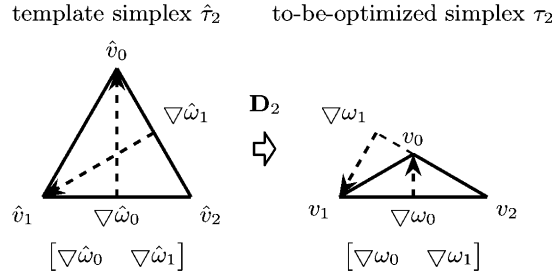


Fig. 2. The illustration of gradient-based shape matching for a 2-simplex.

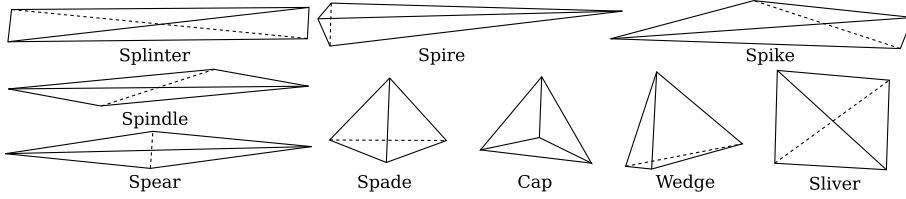


Fig. 3. Several typical badly-shaped tetrahedrons (Freitag and Knupp, 2002).

pointing perpendicularly from the opposite $(d-1)$ -simplex S_i (as the base) to the vertex v_i . The length of $\nabla \omega_i$ is equal to the inverse of height h_i , i.e., $|\nabla \omega_i| = \frac{1}{|h_i|}$ in 2-simplex case, and $|\nabla \omega_i| = \frac{|S_i|}{|\tau_d|}$ in 3-simplex case, where $|\tau_d|$ is the volume of the simplex τ_d and $|S_i|$ is the area of face S_i . Both its direction and length encode the shape information. Besides that $\nabla \omega_i, 0 \leq i < d$ are linearly independent for any non-degenerate simplex. So using the gradients of linear shape functions as the bases to represent simplex well describe the shape as well as size difference between a badly-shaped simplex and a regular simplex.

For the simplicity of illustration, we still use a 2-simplex in Fig. 2 to demonstrate the basic idea of GSM.

Any 2-simplex τ_2 is represented by its gradient matrix $[\nabla \omega_0 \quad \nabla \omega_1]$. Once $\nabla \omega_0$ and $\nabla \omega_1$ are given, $\mathbf{e}_{20}, \mathbf{e}_{21}$ is uniquely defined, so triangle is uniquely determined by $\nabla \omega_0$ and $\nabla \omega_1$. Affine transformation $\mathbf{D}_2$ between $\hat{\tau}_2$ and τ_2 can be expressed as:

$$\mathbf{D}_2 [\nabla \hat{\omega}_0 \quad \nabla \hat{\omega}_1] = [\nabla \omega_0 \quad \nabla \omega_1]. \quad (4)$$

Then, GSM energy between the template $\hat{\tau}_2$ and any 2-simplex τ_2 is defined as

$$E_{\tau_2}^{gsm} = \text{trace}(\mathbf{D}_2^T \mathbf{D}_2). \quad (5)$$

Inherited from Frobenius norm properties, GSM energy is also scale-sensitive and orientation-free. By expanding Eq. (5) with the height definition and Eq. (4), GSM energy is simplified as:

$$E_{\tau_2}^{gsm} = \frac{\hat{a}^2}{8} \frac{\sum_{0 \leq i < j \leq 2} \mathbf{e}_{ij}^T \mathbf{e}_{ij}}{|\tau_2|^2} = \frac{\hat{a}^2}{2} \sum_{i=0}^2 \frac{1}{|h_i|^2}, \quad (6)$$

where $\hat{a}$ is the edge length of the regular template $\hat{\tau}_2$, $|\tau_2|$ is the area of simplex τ_2 . For each 2-simplex, GSM energy is the summation of inverse of squared heights. When there is one internal dihedral angle approaching 0° or 180° , one or more heights will be also close to 0. Thus minimizing GSM energy inhibits small heights and also uneven height. So it suppresses all badly-shaped simplices. The minimal energy is reached when all heights are equal to each other, i.e., an equilateral 2-simplex. Besides that, the energy for each simplex also encodes its size information (height h_i). So when minimizing the total energy, the result mesh will converge to the optimal solution.

3.2.1. Tetrahedralization $d = 3$

Tetrahedral mesh optimization is more challenging than triangular mesh optimization, especially to remove slivers completely. Fig. 3 shows several typical badly-shaped tetrahedrons (Freitag and Knupp, 2002). Most of them either have one or more dihedral angles approaching $0^\circ/180^\circ$ or have uneven heights. Traditional ESM energy cannot avoid slivers as discussed in Sec. 3.1, while the proposed GSM energy significantly suppress all of them.

Affine transformation $\mathbf{D}_3$ between $\hat{\tau}_3$ and τ_3 is represented as:

$$\mathbf{D}_3 [\nabla \hat{\omega}_0 \quad \nabla \hat{\omega}_1 \quad \nabla \hat{\omega}_2] = [\nabla \omega_0 \quad \nabla \omega_1 \quad \nabla \omega_2]. \quad (7)$$

For any 3-simplex, $\nabla \omega_i$ has the same direction as the normal of face S_i , where S_i is the face opposite to vertex $\mathbf{v}_i$. If the template tetrahedron is regular and with edge length $\hat{a}$, then GSM energy $E_{\tau_3}^{gsm}$ between $\hat{\tau}_3$ and τ_3 can be simplified as:

$$E_{\tau_3}^{gsm} = \text{trace}(\mathbf{D}_3^T \mathbf{D}_3) = \frac{\hat{a}^2}{18} \frac{\sum_{i=0}^3 |S_i|^2}{|\tau_3|^2} = \frac{\hat{a}^2}{2} \sum_{i=0}^3 \frac{1}{|h_i|^2}, \quad (8)$$

where $|S_i|$ is the area of face S_i , $|\tau_3|$ is the volume of simplex τ_3 .

GSM energy of a tetrahedron is the summation of the inverse of the squared heights. The optimal solution is to have the same heights, i.e. regular tetrahedron. The total GSM energy for the entire tetrahedral mesh is:

$$E_{total}^{gsm} = \frac{\hat{a}^2}{18} \frac{1}{|\mathcal{T}|} \sum_{\tau_3 \in \mathcal{T}} \frac{\sum_{i=0}^3 |S_i|^2}{|\tau_3|^2}, \quad (9)$$

where $\mathcal{T}$ is the set of tetrahedrons in the volume mesh and $|\mathcal{T}|$ is the total number of tetrahedrons in the volume mesh. When minimizing the GSM energy, the optimal solution is to have all the heights to be the same. GSM energy has great punishment on a small height. So minimizing the energy effectively suppressing all the badly-shaped tetrahedrons.

3.2.2. Curve discretization $d = 1$

A curve can be discretized to be a set of 1-simplices. Inheriting from the idea of GSM, the mapping between the template $\hat{\tau}_1$ and the to-be-optimized τ_1 is the ratio between their lengths $\mathbf{D}_1 = \frac{|\hat{\tau}_1|}{|\tau_1|}$. So GSM energy function of the line segment τ_1 is:

$$E_{\tau_1}^{gsm} = \frac{|\hat{\tau}_1|^2}{|\tau_1|^2}. \quad (10)$$

GSM energy of the entire curve is:

$$E_{\tau_1} = \frac{|\hat{\tau}_1|^2}{|\mathcal{L}|} \sum_{\tau_1 \in \mathcal{L}} \frac{1}{|\tau_1|^2}, \quad (11)$$

where $\mathcal{L}$ is the set of 1-simplices and $|\mathcal{L}|$ is the total number of 1-simplices in the corresponding curve $\mathcal{L}$.

3.3. The importance of GSM energy in tetrahedral meshing

The difference between ESM and GSM lies in the representation of a simplex. ESM uses edge vectors, while GSM utilizes the gradients of linear shape functions. When $d = 3$, we have $\nabla\omega_0 = \frac{\mathbf{e}_{12} \times \mathbf{e}_{13}}{6|\tau_3|}$, $\nabla\omega_1 = \frac{\mathbf{e}_{02} \times \mathbf{e}_{03}}{6|\tau_3|}$, $\nabla\omega_2 = \frac{\mathbf{e}_{03} \times \mathbf{e}_{01}}{6|\tau_3|}$, and $\nabla\omega_1 \times \nabla\omega_2 = \frac{\mathbf{e}_{01}}{6|\tau_3|}$. Via the cross product property in matrix transformation, the affine transformation $\mathbf{J}_3$ in ESM satisfies:

$$(\mathbf{J}_3 \mathbf{e}_{ij}) \times (\mathbf{J}_3 \mathbf{e}_{ik}) = |\mathbf{J}_3| \mathbf{J}_3^{-T} \mathbf{e}_{ij} \times \mathbf{e}_{ik}, \quad (12)$$

where $|\mathbf{J}_3|$ is the determinant of matrix $\mathbf{J}_3$.

Then the relation between affine transformations of ESM and GSM can be expressed as

$$\begin{aligned} \mathbf{D}_3 &= [\nabla\omega_0 \quad \nabla\omega_1 \quad \nabla\omega_2] [\nabla\hat{\omega}_0 \quad \nabla\hat{\omega}_1 \quad \nabla\hat{\omega}_2]^{-1} \\ &= \frac{|\mathbf{J}_3| \mathbf{J}_3^{-T} |\hat{\tau}_3|}{|\tau_3|} [\hat{\mathbf{e}}_{12} \times \hat{\mathbf{e}}_{13} \quad \hat{\mathbf{e}}_{02} \times \hat{\mathbf{e}}_{03} \quad \hat{\mathbf{e}}_{03} \times \hat{\mathbf{e}}_{01}] [\hat{\mathbf{e}}_{12} \times \hat{\mathbf{e}}_{13} \quad \hat{\mathbf{e}}_{02} \times \hat{\mathbf{e}}_{03} \quad \hat{\mathbf{e}}_{03} \times \hat{\mathbf{e}}_{01}]^{-1} \\ &= \mathbf{J}_3^{-T}. \end{aligned} \quad (13)$$

Suppose the eigenvalues of $\mathbf{J}_3$ are $\lambda_1, \lambda_2, \lambda_3$, then ESM energy of one tetrahedron is

$$E_{\tau_3}^{esm} = \text{trace}(\mathbf{J}_3^T \mathbf{J}_3) = \lambda_1^2 + \lambda_2^2 + \lambda_3^2, \quad (14)$$

while GSM energy of the tetrahedron is

$$E_{\tau_3}^{gsm} = \text{trace}(\mathbf{J}_3^{-1} \mathbf{J}_3^{-T}) = \frac{1}{\lambda_1^2} + \frac{1}{\lambda_2^2} + \frac{1}{\lambda_3^2}. \quad (15)$$

Although both ESM and GSM reach their minimums when $\lambda_1 = \lambda_2 = \lambda_3$, GSM is more sensitive to a small λ_i .

4. GSM energy optimization

The proposed GSM energy optimization is to build a high-quality regular d -simplex mesh in a domain Ω_d based on a given regular template and the user-specified number of vertices N . Both the positions of N vertices and their connectivities are required to be optimized. Our mesh optimization process involves two operations: the numerical nonlinear optimization for vertex smoothing and the combinatorial optimization for connectivity update. These two operations are carried out iteratively to minimize the proposed GSM energy.

4.1. Vertex smoothing

With fixed mesh connectivity, the optimization computation reduces to a nonlinear numerical optimization problem. The connectivity is fixed and inverted simplices should be avoided during the vertex smoothing, so the possible solution of a vertex should be inside its one-ring domain. Newton's method converges quickly in the local region, so it is chosen to do vertex smoothing.

The vertex updating rule of Newton's method is:

$$\mathbf{v}^* = \mathbf{v} - \Delta \mathbf{v} = \mathbf{v} - \alpha \mathbf{h}^{-1} \mathbf{g}, \quad (16)$$

where $\mathbf{h}$ and $\mathbf{g}$ are the Hessian and gradient at vertex position $\mathbf{v}$, $\mathbf{v}^*$ is the updated position of vertex $\mathbf{v}$, and α is the step size. The backtracking line search is utilized to determine the step size.

In the following, we will discuss how to calculate the gradient and Hessian of GSM energy. GSM energy in any d -simplex has the similar formulation, which can be written as:

$$E_d^{gsm} = c_d \frac{p(\mathbf{v})}{q(\mathbf{v})}, \quad (17)$$

where c_d is a constant. When $d = 1$, $p(\mathbf{v})$ is 1, $q(\mathbf{v})$ is the squared length of the simplex τ_1 ; when $d = 2$, $p(\mathbf{v})$ is the summation of the squared edge lengths, $q(\mathbf{v})$ is the squared area of the simplex τ_2 ; when $d = 3$, $p(\mathbf{v})$ is the summation of the squared face areas, $q(\mathbf{v})$ is the squared volume of the simplex τ_3 .

The gradient and Hessian of GSM energy in d -simplex are:

$$\mathbf{g}_{\tau_d} = c_d \frac{\mathbf{p}'(\mathbf{v})q(\mathbf{v}) - p(\mathbf{v})\mathbf{q}'(\mathbf{v})}{q(\mathbf{v})^2}, \quad (18)$$

$$\mathbf{h}_{\tau_d} = c_d \frac{\mathbf{h}_p q(\mathbf{v}) + \mathbf{p}'(\mathbf{v})\mathbf{q}'(\mathbf{v})^T - \mathbf{q}'(\mathbf{v})\mathbf{p}'(\mathbf{v})^T - \mathbf{h}_q p(\mathbf{v}) - 2q(\mathbf{v})\mathbf{g}_{\tau_d}\mathbf{q}'(\mathbf{v})^T}{q(\mathbf{v})^2}, \quad (19)$$

where $\mathbf{p}'(\mathbf{v})$, $\mathbf{q}'(\mathbf{v})$, $\mathbf{h}_p$ and $\mathbf{h}_q$ are the first order and second order derivatives of $p(\mathbf{v})$, $q(\mathbf{v})$. For arbitrary vertex $\mathbf{v}$, its one-ring simplices set $\mathcal{T}_v$ are used for gradient and Hessian computations.

Based on the gradient and Hessian of GSM energy at any vertex, the search direction $\Psi = \mathbf{h}^{-1}\mathbf{g}$ is obtained. Since GSM is not convex in the one-ring domain, the backtracking line search is used in both Ψ and $-\Psi$ directions to obtain the optimal step size α . The initial value of α is set as the maximal possible movement inside its one-ring domain, i.e., $\alpha_{init} = \frac{\max_{\mathbf{v}_i \in N_v} |\mathbf{v} - \mathbf{v}_i|}{\|\mathbf{h}^{-1}\mathbf{g}\|}$, where the set of vertices in $\mathcal{T}_v$ is N_v .

It is noted that Newton's method is defined for each vertex. The vertices are updated one by one following the descending order of the norm of vertex update vector, i.e., $\|\alpha \mathbf{h}^{-1}\mathbf{g}\|$. The worst positioned vertex is optimized first. The backtracking line search and vertex update avoid the increase of GSM energy as well as inverted simplex.

4.2. Connectivity update

Traditional CVT and ODT energy functions for meshing are based on Delaunay triangulation, while our GSM energy is not congruent with Delaunay triangulation. Given an initial connectivity, a set of flip operations are employed to optimize the connectivity. The details of connectivity updating in triangular and tetrahedral meshes are different.

For a 3D triangular mesh, the initial mesh is obtained using surface constrained mesh generation (Yan et al., 2009). The 2–2 edge-flipping operation is used to decrease the energy. A flip operation is performed if the energy will be decreased after the flip operation. Besides that, we also need to make sure the new triangles are still restricted to the original surface and avoid to generate non-manifold edges, i.e. one edge shared by more than two triangles. The edges along the sharp features and boundary edges are never flipped. During the connectivity optimization, we traverse through all the edges until the energy does not decrease anymore.

For a 3D tetrahedral volume mesh, the initial mesh is built by TetGen (Si, 2015). There are several flip operations available in tetrahedral meshing, including 2–3 flip, 3–2 flip, 4–4 flip. Besides that, edge removal and multi-face removal proposed in Shewchuk (2002b) are utilized to further improve the connectivity. Comparing to edge flip in triangle mesh, flip operation in tetrahedral mesh may change the number of tetrahedrons. The flip operation will be performed if average GSM energy to each simplex decreases after the operation. During the connectivity optimization, we will traverse through all the edge and faces until the energy cannot decrease anymore.

4.3. Boundary and feature

Sharp features and boundary edges of 2-simplex meshes are usually defined by a set of curves (1-simplex meshes). The boundary of a 3-simplex mesh is a 2-simplex mesh.

In 2-simplex mesh optimization, we first extract the boundary and feature curves and estimate vertex numbers on those curves. Then the vertices along those curves are optimized. After that, we randomly sample the remaining vertices on surface domain and optimize them by fixing those boundary and feature points. The vertex numbers on one surface boundary curves or sharp features are estimated in the following way. Suppose A_Ω is surface area and L_Ω is the length of boundary curves, according to Euler's polyhedron formula, we have:

$$N_s - \frac{F}{2} - \frac{L_\Omega}{2\sqrt{4A_\Omega/\sqrt{3}F}} = 2 - 2g, \quad (20)$$

where N_s is vertex number on boundary, F is face number of boundary, g is the genus of surface. Only F is unknown in Eq. (20), so we can compute F . Then edge length l_{est} is estimated by $l_{est} = \sqrt{4A_\Omega/\sqrt{3}F}$. Finally, the vertex number on each curve is obtained by dividing its length by l_{est} .

In tetrahedral meshing, the vertex number on boundary is estimated at the beginning based on the domain boundary area A_Ω and the domain volume V_Ω . Body-centered cubic (BCC) lattice is used for the estimation. Voronoi cell of each vertex in BCC lattice is a truncated octahedron. By using truncated octahedron as Voronoi cell, the edge length l_{est} of an equilateral tetrahedron satisfies

$$V_\Omega = \frac{N}{\sqrt{2}} l_{est}^3 - \frac{A_\Omega}{\sqrt{6}} l_{est} + 2 - 2g, \quad (21)$$

where N is total vertex number. l_{est} can be calculated from Eq. (21). Then the boundary vertex number is $N_b = \frac{2A_\Omega}{\sqrt{3}l_{est}^2} + 2 - 2g$. The 2-simplex mesh optimization is applied to the surface boundary with N_b vertices. After that, we randomly sample the remaining vertices inside the volume and optimize them with fixed boundary surface vertices.

In the following subsections, the vertex update rules for 1-simplex curves and 2-simplex surfaces are introduced.

4.3.1. Feature curve

1-simplex GSM optimizations are performed on the corresponding curves, which has been discussed in Sec. 3.2.2. During 1-simplex mesh optimizations, all the vertices are restricted on the corresponding curves. The update vector is projected along the tangent direction of the curve at $\mathbf{v}$. Suppose the normalized tangent direction of the curve l at vertex $\mathbf{v}$ is $\mathbf{d}_{lv}$, then the vertex update rule of Newton's method based on Eq. (16) is:

$$\mathbf{v}^* = \mathbf{v} - \left(\alpha \left(\mathbf{h}_v^{-1} \mathbf{g}_v \right)^T \mathbf{d}_{lv} \right) \mathbf{d}_{lv}. \quad (22)$$

After that, $\mathbf{v}^*$ is projected to the closest point on the corresponding curve.

4.3.2. Boundary surface

2-simplex GSM optimization is performed on the corresponding 3D surface, which has been discussed at the beginning of Sec. 3.2. During 3D surface optimization, vertices should be restricted to the given surface. When updating the vertex positions, vertex movement is only allowed on its tangent plane. Suppose the normal of vertex $\mathbf{v}$ is $\mathbf{n}_v$, then:

$$\mathbf{v}^* = \mathbf{v} - \left(\alpha \mathbf{h}_v \mathbf{g}_v - \left(\alpha \mathbf{h}_v \mathbf{g}_v \right)^T \mathbf{n}_v \right) \mathbf{n}_v. \quad (23)$$

After that, $\mathbf{v}^*$ is projected to the closest point on the given surface.

The GSM optimization framework is given in Algorithm 1.

5. Experiment and comparisons

We implement the algorithms using C++. The experiments are done on a workstation with Intel® Xeon E5645 CPU 2.40 GHz, and 32G DDR3 RAM. To demonstrate the performance of the proposed GSM method, we compare it with four mesh optimization approaches provided by The Computational Geometry Algorithms Library (CGAL) (Jamin et al., 2015). The optimizations of CGAL mesher have two categories. One is the local optimization, including vertex perturbation (Tournois et al., 2009) and sliver exudation (Cheng et al., 2000). The other one is the global optimization, including Lloyd smoother (Du et al., 1999; Du and Wang, 2003) and ODT smoother (Alliez et al., 2005; Chen and Xu, 2004). In the following, the tetrahedral mesh quality criteria are introduced in Sec. 5.1. The experimental results on isotropic meshing are presented in Sec. 5.2 (smooth surface) and Sec. 5.3 (surface with sharp features), and the experimental results of adaptive and anisotropic meshings are provided in Sec. 5.4. Finally, the running time and robustness analysis are given in Sec. 5.5 and 5.6. Due to the page limit, we only present seven models in the following, more meshing results are given in a supplementary document. Table 1 gives detailed quality statistics of all volume meshing models in the experiment. The best result in each model is shown in bold font.

Algorithm 1: GSM optimization.

Input: vertex number N , boundary domain Ω , anisotropic metric $\mathbf{M}$
Output: tetrahedral mesh (V, T) with N vertices

- 1 estimate boundary vertex number N_b and edge length l_{est} ;
- 2 **if** Ω contains sharp features **then**
- 3 estimate vertex number N_s for sharp features;
- 4 optimize vertices on sharp features V_s according to Eq. (22);
- 5 **end**
- 6 optimize vertices V_b by fix V_s according to Eq. (23);
- 7 randomly sample $N - N_b - N_s$ vertices inside Ω as V_f ;
- 8 build tetrahedral mesh T of all N vertices;
- 9 **for** $k \leftarrow 0$ to 50 **do**
- 10 calculate update vector $\Delta \mathbf{v}$ of all vertices;
- 11 **for** $i \leftarrow 0$ to $(N - N_b - N_s)/4$ **do**
- 12 $\mathbf{v}_i \leftarrow \arg \max_{\mathbf{v} \in V_f} \|\Delta \mathbf{v}\|$ with Newton's Method Eq. (16);
- 13 update position of $\mathbf{v}_i$;
- 14 **end**
- 15 optimize connectivity T ;
- 16 **end**

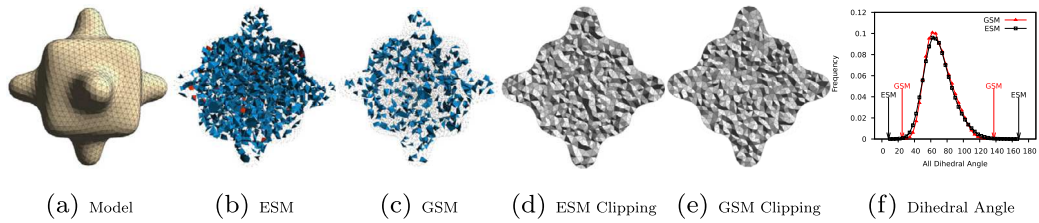


Fig. 4. Comparison of ESM and GSM on the Bumpycube volume. The red ones are tetrahedrons with the smallest dihedral angles less than 18° , while the blue ones are tetrahedrons with the smallest dihedral angles less than 36° . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.1. Quality measurement

The quality criteria used in all our experiments for the isotropic meshing are dihedral angle θ and radius ratio $\gamma = 3 \frac{r_{in}}{r_{circ}}$, where r_{in} is inradius and r_{circ} is circumradius. θ_{min} is the smallest dihedral angle. θ_{max} is the largest dihedral angles. $\bar{\theta}_{min}$ is the average value of the smallest dihedral angle of each tetrahedron. γ_{min} is the smallest radius ratios among all tetrahedrons. γ_{mean} is the average radius ratios of all tetrahedrons. The distribution of dihedral angles and radius ratios of all tetrahedrons are provided. Since the sliver is measured by the dihedral angles, we evaluate our experiments extensively by the number of tetrahedrons with different degrees of smallest dihedral angles 10° , 20° , 30° , and 40° as thresholds. For anisotropic tetrahedral meshes, each tetrahedron is transformed to the isotropic space, then its quality is measured based on the above isotropic criteria.

5.2. Isotropic tetrahedral meshing

Fig. 4 shows the isotropic tetrahedral meshing results on the Bumpycube volume for comparison between GSM and ESM. GSM produces better θ_{min} and θ_{max} comparing to ESM. It is also observed that GSM has better performance on sizing control. The tetrahedral meshing result of ESM optimization has larger variance on tetrahedral volumes.

Fig. 5 shows the isotropic tetrahedral meshing results on the Duck volume. With random initialization, GSM method produces meshes with better θ_{min} , θ_{max} , as well as radius ratios, which outperforms all other methods provided by CGAL (i.e., both local and global optimizations) as shown in Table 1. In order to generate high-quality tetrahedral meshes, we use Particle, Lloyd, and ODT results as the initializations, and then apply the proposed GSM, vertex perturbation, and sliver exudation to further suppress slivers. The results demonstrate that GSM method is an effective sliver remover. Fig. 5 also provides the distributions of dihedral angles and radius ratios in Duck volume meshes of different methods.

5.3. With sharp features

Fig. 6 shows the isotropic tetrahedral meshing results as well as distributions of dihedral angles and radius ratios on the Fandisk volume with sharp features in different methods. We reach the same conclusion as the previous example that no matter with random initialization or initialized by Particle, Lloyd, and ODT results, our proposed GSM method obtains better θ_{min} , θ_{max} , as well as radius ratios, outperforming all other methods provided by CGAL. More detailed quality statistics comparison is given in Table 1.

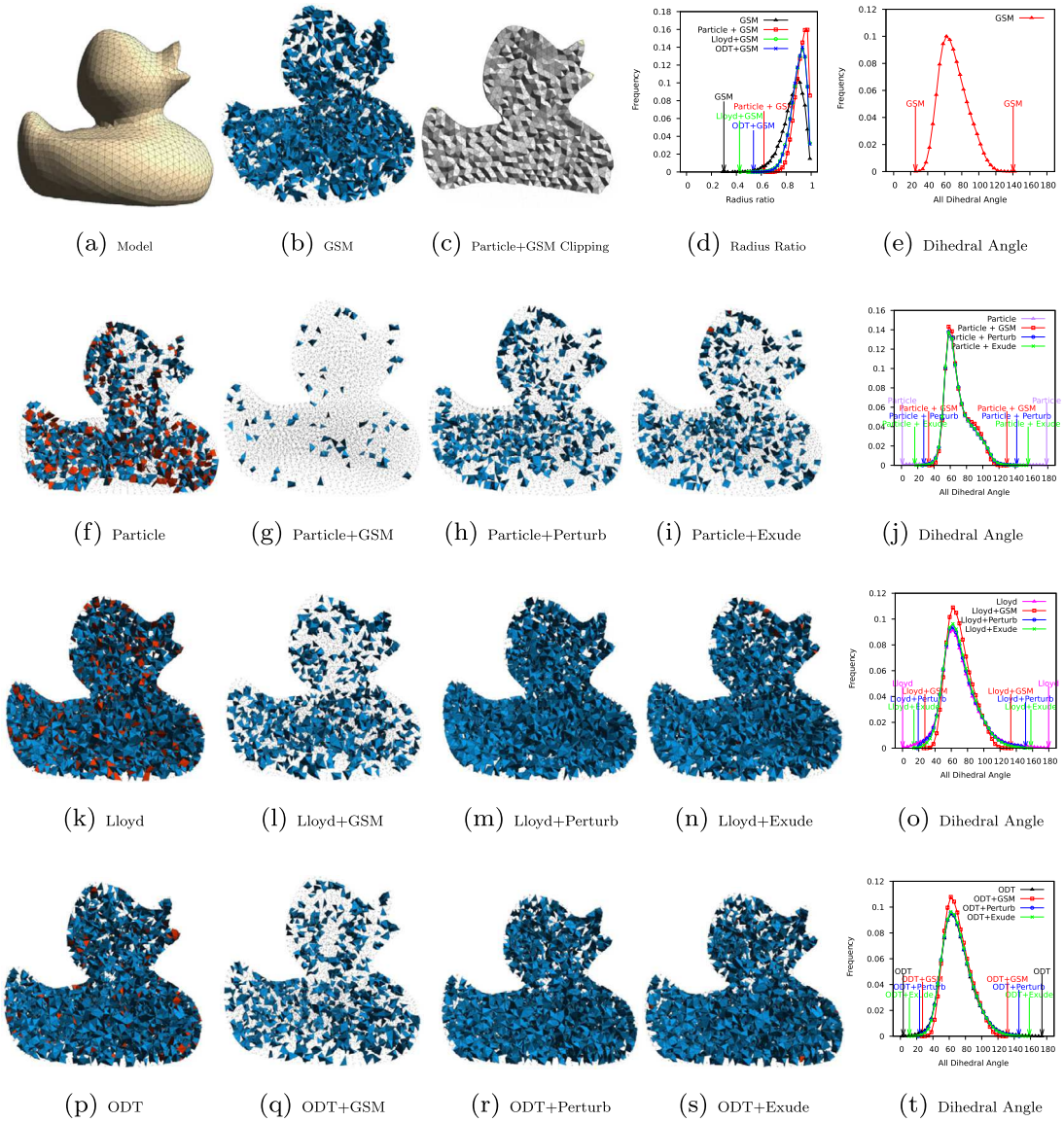


Fig. 5. Duck volume meshing with 10,000 vertices. The red ones are tetrahedrons with the smallest dihedral angles less than 20° , while the blue ones are tetrahedrons with smallest dihedral angles less than 40° . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.4. Adaptive and anisotropic tetrahedral meshing

In order to show the generalization of the proposed GSM method, we also work on tetrahedral meshes in adaptive and anisotropic cases.

Fig. 7 shows the adaptive tetrahedral meshing results on Sphere volume with scaling field, i.e. $\mathbf{M}(\mathbf{x}) = \Lambda^2$, where $s = (0.025 + 0.2 \sqrt{x^2 + y^2 + z^2} - 0.5)^{-1}$, $\Lambda = \text{diag}\{s, s, s\}$. It is noted that the proposed GSM method leads to better dihedral angles and radius ratios among other methods provided by CGAL.

Fig. 8 shows the anisotropic tetrahedral meshing results on the Sphere volume with metric field $\mathbf{M}(\mathbf{x}) = \mathbf{Q}^T(\mathbf{x})\Lambda\mathbf{Q}(\mathbf{x})$, where $\Lambda = \text{diag}(100, 10, 10)$, and $\mathbf{Q}$'s three columns are $(2 \cos 6x, 1, 0)^T$ and two orthogonal unit vectors. Table 1 provides quality statistics of Sphere volume meshes in the specified anisotropic field with both random and Particle initializations. Here, only GSM and GSM with Particle initialization are compared, since according to our observations, the approach in GSM with Particle initialization can obtain best mesh quality among all other methods.

Fig. 9 shows anisotropic tetrahedral meshing results inside a Cube with metric field defined as $\mathbf{M}(\mathbf{x}) = \Lambda^2$, where $\Lambda = \text{diag}((0.025 + 0.2(1 - e^{-x}))^{-1}, 5, 5)$. The proposed GSM method works well with both random and Particle initializations as shown in Table 1.

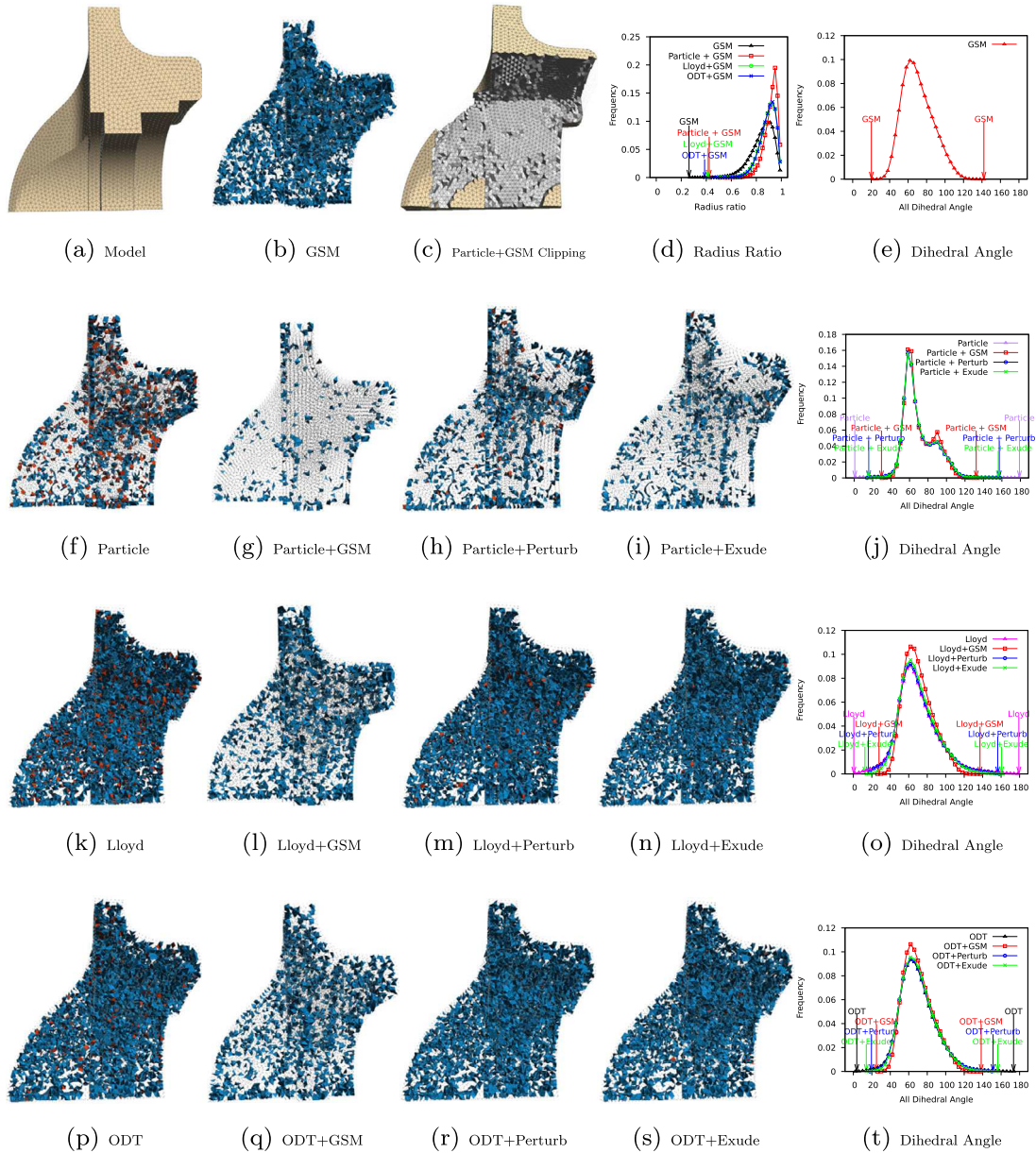
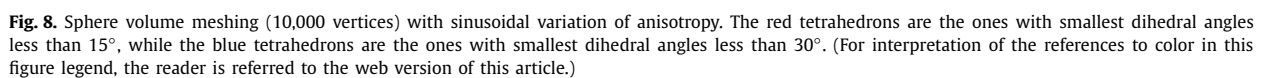
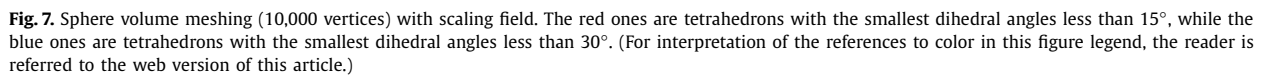


Fig. 6. Fandisk volume meshing with 18,000 vertices. The red ones are tetrahedrons with the smallest dihedral angles less than 20° , while the blue ones are tetrahedrons with the smallest dihedral angles less than 40° . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.5. Running time and convergence analysis

GSM is defined as a global optimization energy. Our GSM optimization iteratively updates vertex positions and vertex connectivities. There are several options to set the stop condition, i.e., maximum round number, energy decrease threshold, or any other mesh quality criteria. In our implementation, we set maximum round number as the criterion. In all of our experiments, maximum round number is set to be 50, which is large enough for all our experiments to converge. The time consumption is given in Table 1.

Duck volume is utilized as an example to show the relationship between θ_{min} , θ_{max} , $\bar{\theta}_{min}$, γ_{mean} , and computational time in Fig. 10. The figure draws the first 40 rounds. The result shows that the optimizations converge fast so it provides good enough result after the first few rounds. θ_{min} is not strictly increasing along with the computational time and θ_{max} is not strictly decreasing along with the computational time, because we are not directly optimizing the minimal dihedral angle and the maximal dihedral angle. However, $\bar{\theta}_{min}$ and γ_{mean} keep increasing along with the computational time until reaching some local optimal solutions. Comparing to Perturb and Exude methods provided by CGAL, our implementation is not as



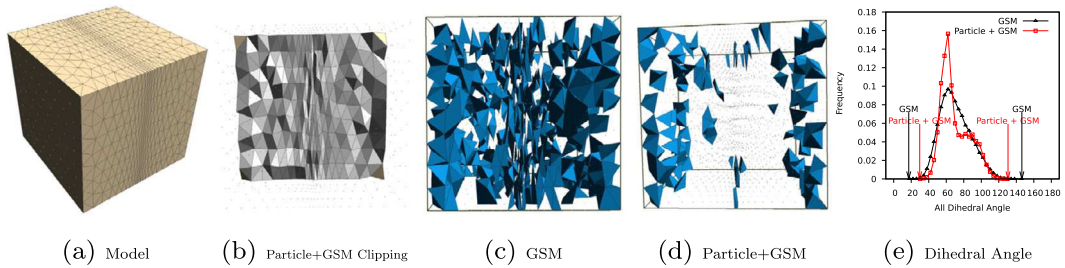


Fig. 9. Cube volume meshing (2000 vertices) with anisotropic variation in a single direction. The red ones are tetrahedron with smallest dihedral angles less than 20° , while the blue ones are tetrahedron with smallest dihedral angles less than 40° . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1

Quality statistics of all volume meshing models. Note: best results of each model are in bold font.

Model	Method	$\theta_{min}/\theta_{max}$	$\gamma_{min}/\gamma_{mean}$	# < 10°	# < 20°	# < 30°	# < 40°	#tet	Time (sec.)
Duck (Iso.)	Init	0.222/179	0.008/0.588	3422	12,830	25,445	38,473	57,898	n/a
	GSM	25.93/137.6	0.357/0.852	0	0	33	2211	49,782	1768.76
	Particle	1/179	0.02/0.898	190	413	737	1580	52,322	82.89
	Particle+GSM	32/123.8	0.632/0.919	0	0	0	99	51,522	1830.03
	Particle+Perturb	27.4/142	0.469/0.907	0	0	77	747	51,750	5.99
	Particle+Exude	16.3/156	0.305/0.906	0	3	85	625	51,736	3.66
	Lloyd	0.164/180	0.003/0.846	226	996	2592	6867	53,620	26.16
	Lloyd+GSM	30.05/132.7	0.411/0.893	0	0	0	923	50,800	1876.72
	Lloyd+Perturb	19.3/151	0.349/0.859	0	1	1480	5413	52,697	33.09
	Lloyd+Exude	14.1/158	0.273/0.866	0	52	704	4116	52,093	28.16
	ODT	4.18/174	0.084/0.875	36	325	1192	5036	52,571	17.28
	ODT+GSM	26.75/134.6	0.456/0.894	0	0	2	981	50,837	1089.84
Fandisk (Feature)	Init	0.09656/179	0.0035/0.589	6039	21934	44258	67,110	102,135	n/a
	GSM	20.36/142.8	0.275/0.845	0	0	91	4550	88,015	4158.5
	Particle	0.6863/179	0.013/0.898	272	701	1324	3038	92,620	301.71
	Particle+GSM	29.6/132.2	0.423/0.916	0	0	1	471	91,241	4458.5
	Particle+Perturb	15.73/157.1	0.266/0.903	0	192	783	2379	91,545	6.79
	Particle+Exude	15.5/156.1	0.303/0.906	0	14	185	1290	90,442	6.59
	Lloyd	0.486/179.2	0.01/0.842	409	1865	4869	12,850	93,683	51.24
	Lloyd+GSM	27.51/136.9	0.413/0.889	0	0	6	1927	89,790	4315.6
	Lloyd+Perturb	16.15/156.3	0.292/0.852	0	535	3355	10,903	92,420	55.53
	Lloyd+Exude	12.56/160.5	0.219/0.863	0	84	1353	7802	90,329	52.4
	ODT	3.724/173.5	0.081/0.873	70	626	2218	9079	91,633	34.00
	ODT+GSM	25.06/138.4	0.386/0.889	0	0	12	2174	90,082	4345.86
Sphere (Adap.)	Init	0.23/179	0.005/0.558	7903	29,612	58,635	87,717	121,650	n/a
	GSM	21.1/138	0.355/0.837	0	0	118	5887	103,485	7840.78
	Particle	1.03/178	0.0203/0.856	372	1718	4655	12,282	111,505	410.37
	Particle+GSM	30.2/130	0.531/0.900	0	0	0	1092	106,106	8787.22
	Particle+Perturb	23.7/147	0.384/0.872	0	0	1801	8444	109,445	21.16
	Particle+Exude	12.3/162	0.210/0.874	0	53	1127	6794	108,789	6.79
	Lloyd	0.292/180	0.006/0.852	547	2234	6349	17,809	147,734	81.18
	Lloyd+GSM	28.9/128	0.496/0.895	0	0	4	2134	140,757	9636.53
	Lloyd+Perturb	23.2/147	0.368/0.867	0	0	2563	13,051	144,745	81.46
	Lloyd+Exude	12.8/161	0.245/0.869	0	108	1775	11,106	144,018	73.28
	ODT	1.47/178	0.0297/0.874	86	742	3284	15,068	144,176	54.9
	ODT+GSM	26.1/129	0.439/0.893	0	0	6	2854	140,776	8888.4
Sphere (Aniso.)	GSM	16.5/150.5	0.147/0.812	0	7	500	6359	51,405	4659.15
	Particle+GSM	21.5/142.5	0.289/0.864	0	0	174	3398	52,206	4464.69
Cube (Aniso.)	GSM	17.2/146	0.271/0.832	0	2	33	737	8783	404.55
	Particle+GSM	26.5/136	0.373/0.903	0	0	4	155	25,253	453.9

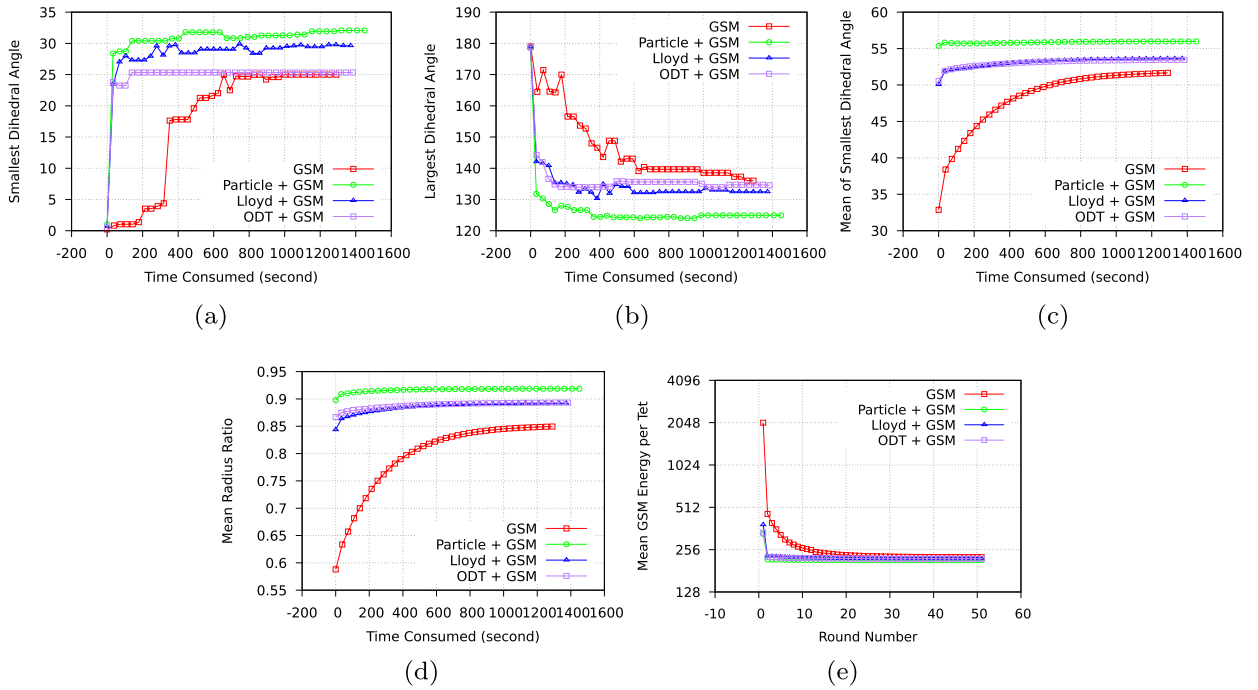


Fig. 10. (a), (b), (c), (d) Dihedral angle and radius ratio quality changing along with computational time of the Duck model. (e) GSM energies changing along with the round numbers in different initializations.

Table 2

Quality statistics of Teddy volume meshes with different vertex numbers.

Method	Vertex number	2000	5000	10,000	20,000	30,000
GSM	$\theta_{\min}/\theta_{\max}$	25.6/138.9	24.09/139.1	23/136.9	23.74/136.6	23.39/140.2
Particle+GSM	$\theta_{\min}/\theta_{\max}$	32.9/126.4	30.88/132.4	34.28/124	33.63/128.9	34.54/127.1
GSM	$\gamma_{\min}/\gamma_{\max}$	0.309/0.842	0.347/0.848	0.402/0.853	0.36/0.853	0.353/0.853
Particle+GSM	$\gamma_{\min}/\gamma_{\max}$	0.623/0.907	0.595/0.914	0.581/0.918	0.556/0.92	0.571/0.921

efficient currently, but could be improved. For instance, if we compute vertex and connectivity updates smarter instead of a global strategy or use some parallel strategies, the implementation will be much more efficient. This will be one of our future work.

Fig. 10e shows GSM energy changing along with the increasing of optimization round number in different initializations. In our GSM optimization, monotonic decrease of energy is guaranteed, so our optimization always converge with different initializations, including random, Particle, Lloyd, and ODT initializations. Although different initializations may lead to different final energies, Particle+GSM scheme reaches the lowest convergence energy comparing to other initializations in this example (i.e., Duck volume model).

5.6. Robustness

To show the robustness of our GSM energy, different vertex numbers on the Teddy volume are demonstrated. The volume clipping results are shown in Fig. 11 and quality statistics e.g. $\theta_{\min}$, $\theta_{\max}$, $\gamma_{\min}$, $\gamma_{\max}$ of GSM and Particle+GSM methods are shown in Table 2. The result shows that GSM has stable performance under different vertex numbers both as a standalone optimization method and as post-processing after Particle optimization.

6. Conclusion

In this paper, we introduce an effective sliver suppression method based on shape matching idea. It generates high-quality tetrahedral meshes in isotropic, adaptive, and anisotropic cases. The proposed GSM method is evaluated on extensive volume models and compared with state-of-the-art approaches. The results of proposed GSM method show much better performance than all other current methods. In the future, we would like to improve the computation speed by using GPU parallel techniques.

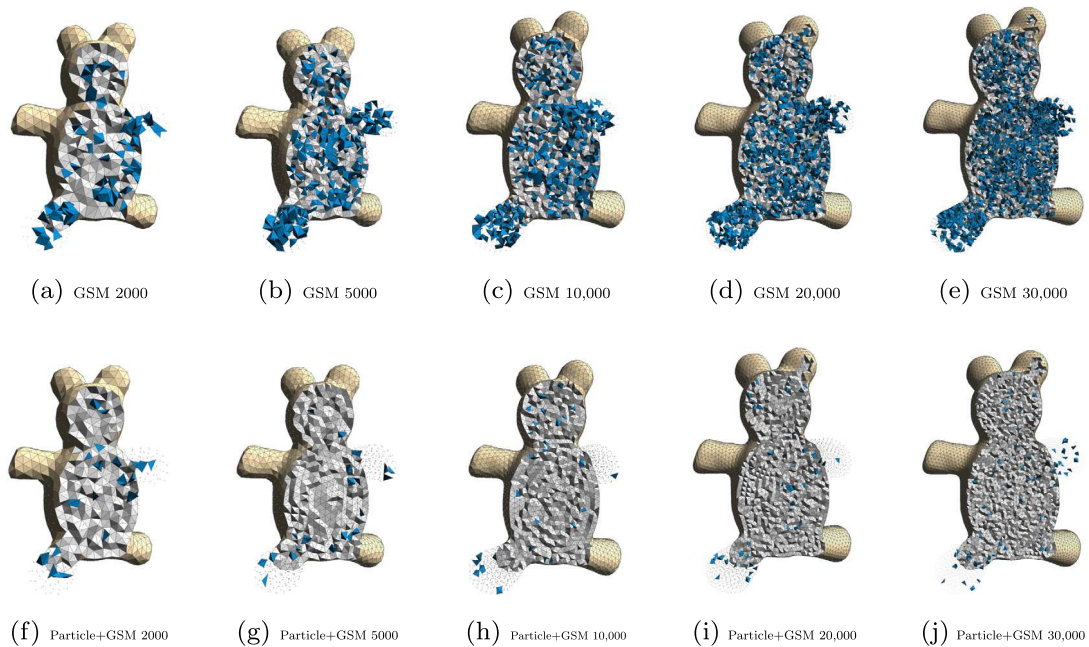


Fig. 11. Teddy volume meshes with different vertex numbers. The blue ones are tetrahedrons with the smallest dihedral angle less than 40° . Note: there's no tetrahedron with the smallest dihedral angle less than 20° . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the paper. Saifeng Ni and Xiaohu Guo are partially supported by National Science Foundation (NSF) under Grant No. IIS-1149737. Zichun Zhong is partially supported by NSF ACI-1657364. Wenping Wang is partially supported by National Natural Science Foundation of China (No. 61272019). Zhonggui Chen is partially supported by National Natural Science Foundation of China (No. 61472332).

Appendix A. Supplementary material

Supplementary material related to this article can be found online at <http://dx.doi.org/10.1016/j.cagd.2017.02.004>.

References

- Alliez, P., Cohen-Steiner, D., Yvinec, M., Desbrun, M., 2005. Variational tetrahedral meshing. *ACM Trans. Graph.* 24 (3), 617–625.
- Chen, L., Xu, J.-C., 2004. Optimal Delaunay triangulations. *J. Comput. Math.*, 299–308.
- Chen, Z., Wang, W., Lévy, B., Liu, L., Sun, F., 2014. Revisiting optimal Delaunay triangulation for 3D graded mesh generation. *SIAM J. Sci. Comput.* 36 (3), A930–A954.
- Cheng, S.-W., Dey, T., Edelsbrunner, H., Facello, M., Teng, S.-H., 2000. Sliver exudation. *J. ACM* 47 (5), 883–904.
- Cheng, S.-W., Dey, T., Shewchuk, J., 2012. *Delaunay Mesh Generation*. Chapman and Hall/CRC.
- Chew, L.P., 1997. Guaranteed-quality Delaunay meshing in 3D. In: *Proceedings of the 13th Annual Symposium on Computational Geometry*, pp. 391–393.
- Du, Q., Wang, D., 2003. Tetrahedral mesh generation and optimization based on centroidal Voronoi tessellations. *Int. J. Numer. Methods Biomed. Eng.* 56 (9), 1355–1373.
- Du, Q., Wang, D., 2005. Anisotropic centroidal Voronoi tessellations and their applications. *SIAM J. Sci. Comput.* 26 (3), 737–761.
- Du, Q., Faber, V., Gunzburger, M., 1999. Centroidal Voronoi tessellations: applications and algorithms. *SIAM Rev.* 41 (4), 637–676.
- Freitag, L., Knupp, P., 2002. Tetrahedral mesh improvement via optimization of the element condition number. *Int. J. Numer. Methods Biomed. Eng.* 53 (6), 1377–1391.
- Guo, J., Yan, D.-M., Chen, L., Zhang, X., Deussen, O., Wonka, P., 2016. Tetrahedral meshing via maximal Poisson-disk sampling. *Comput. Aided Geom. Des.* 43, 186–199.
- Ito, Y., Shih, A., Soni, B., 2004. Reliable isotropic tetrahedral mesh generation based on an advancing front method. In: *Proceedings of 13th International Meshing Roundtable*, pp. 95–106.
- Jamin, C., Alliez, P., Yvinec, M., Boissonnat, J.-D., 2015. CGALmesh: a generic framework for Delaunay mesh generation. *ACM Trans. Math. Softw.* 41 (4), 23:1–23:24.
- Klingner, B.M., Shewchuk, J.R., 2007. Aggressive tetrahedral mesh improvement. In: *Proceedings of 16th International Meshing Roundtable*, pp. 3–23.
- Knupp, P., 2001. Algebraic mesh quality metrics. *SIAM J. Sci. Comput.* 23 (1), 193–218.
- Labelle, F., Shewchuk, J.R., 2007. Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. *ACM Trans. Graph.* 26 (3), 57.1–57.10.
- Li, X.-Y., Teng, S.-H., Üngör, A., 2000. Biting: advancing front meets sphere packing. *Int. J. Numer. Methods Biomed. Eng.* 49 (1–2), 61–81.

- Neil Molino, J.T., Bridson, Robert, Fedkiw, R., 2003. A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. In: *Proceedings of 12th International Meshing Roundtable*, pp. 103–114.
- Owen, S.J., 1998. A survey of unstructured mesh generation technology. In: *Proceedings of 7th International Meshing Roundtable*, pp. 239–267.
- Schöberl, J., 1997. NETGEN An advancing front 2D/3D-mesh generator based on abstract rules. *Comput. Vis. Sci.* 1 (1), 41–52.
- Shewchuk, J.R., 2002a. What is a good linear element? interpolation, conditioning, and quality measures. In: *11th International Meshing Roundtable*, pp. 115–126.
- Shewchuk, J.R., 2002b. Two discrete optimization algorithms for the topological improvement of tetrahedral meshes. Unpublished Manuscript 65.
- Si, H., 2015. Tetgen, a Delaunay-based quality tetrahedral mesh generator. *ACM Trans. Math. Softw.* 41 (2), 11:1–11:36.
- Tournois, J., Srinivasan, R., Alliez, P., 2009. Perturbing slivers in 3D Delaunay meshes. In: *Proceedings of 18th International Meshing Roundtable*, pp. 157–173.
- Yan, D.-M., Lévy, B., Liu, Y., Sun, F., Wang, W., 2009. Isotropic remeshing with fast and exact computation of restricted Voronoi diagram. In: *Proceedings of the Symposium on Geometry Processing*, pp. 1445–1454.
- Yan, D.-M., Wang, W., Lévy, B., Liu, Y., 2010. Efficient computation of 3D clipped Voronoi diagram. In: *Proceedings of the 6th International Conference on Advances in Geometric Modeling and Processing*, pp. 269–282.
- Zhong, Z., Guo, X., Wang, W., Lévy, B., Sun, F., Liu, Y., Mao, W., 2013. Particle-based anisotropic surface meshing. *ACM Trans. Graph.* 32 (4), 1–14.