DOI: 10.1111/cgf.13499
Eurographics Symposium on Geometry Processing 2018
T. Ju and A. Vaxman
(Guest Editors)

Volume 37 (2018), Number 5

Field-Aligned and Lattice-Guided Tetrahedral Meshing

Saifeng Ni[1], Zichun Zhong[2], Jin Huang[3], Wenping Wang[4], and Xiaohu Guo[1†]

[1] The University of Texas at Dallas
[2]Wayne State University
[3]State Key Lab of CAD&CG, Zhejiang University
[4] The University of Hong Kong

Abstract

We present a particle-based approach to generate field-aligned tetrahedral meshes, guided by cubic lattices, including BCC and FCC lattices. Given a volumetric domain with an input frame field and a user-specified edge length for the cubic lattice, we optimize a set of particles to form the desired lattice pattern. A Gaussian Hole Kernel *associated with each particle is constructed. Minimizing the sum of kernels of all particles encourages the particles to form a desired layout, e.g., field-aligned BCC and FCC. The resulting set of particles can be connected to yield a high quality field-aligned tetrahedral mesh. As demonstrated by experiments and comparisons, the field-aligned and lattice-guided approach can produce higher quality isotropic and anisotropic tetrahedral meshes than state-of-the-art meshing methods.*

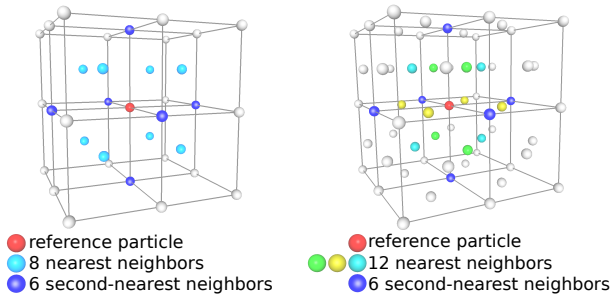
1. Introduction

Field-aligned quadrilateral and hexahedral meshing are active research topics in recent years [PPTSH14, SRUL16, GJTP17]. For quadrilateral and hexahedral meshes, field alignment is very natural because the edges of those meshes are expected to agree with the vectors defining the underlying frame fields. Field-alignment includes the alignments of both Riemannian *distances* and *directions*. In triangular meshing, there has been researches focusing on Riemannian metric (i.e. distance between nodes) control [ZGW*13, FLSG14, NPPZ12], direction control [JTPSH15, DLY*18] or even both [HZP*11]. However, in tetrahedral meshing, only the Riemannian metric has been considered [LS03, DW05a, FLSG14, BSTY15]. So far we have not found any tetrahedral meshing work that takes control of both Riemannian metric and direction into consideration.

The triangle and the tetrahedron are the simplest elements in 2D and 3D, respectively. The dihedral angle of a regular tetrahedron is 70.53°. Unlike tiling regular triangles for 2D Euclidean space, it is impossible to tile regular tetrahedra for 3D Euclidean space. For most of the existing variational tetrahedral meshing algorithms, e.g., either Centroidal Voronoi Tessellation (CVT) based [DW03, ACSYD05, LWL*09], or Optimal Delaunay Triangulation (ODT) based methods [CH11, CWL*14], the majority of their outputs are close to regular tetrahedra, accompanied by some badly shaped tetrahedra. This is one of the reasons that slivers are notoriously hard to remove in tetrahedral meshing [KS07, TSA09], and also one of the reasons that direction-aligned tetrahedral meshing has not been discussed.

Our motivation is to generate tetrahedral meshes with high quality elements, instead of regular tetrahedra, which can pack the 3D Euclidean space. Body-Centered-Cubic (BCC) and Face-Centered-Cubic (FCC) lattices are two close packing scheme of spheres in 3D. The corresponding tetrahedra formed by BCC and FCC lattices have high quality [DW05b], which has been confirmed and used in mesh generation [LS07] and applications [ATW13]. Besides that, the symmetric cubic structures of BCC and FCC also allow us to build field-aligned anisotropic tetrahedral meshes.

In this paper, we propose a particle-based variational method to generate field-aligned cubic lattice, which leads to anisotropic tetrahedral meshes. We design a *Gaussian Hole Kernel* to construct a potential energy of the particle system, and optimize the energy to effectively and efficiently distribute the particles for the desired lattice patterns. To the best of our knowledge, our particle-based cubic lattice (BCC and FCC) optimization method is the first approach that can generate field-aligned isotropic and anisotropic tetrahedral meshes. As illustrated by our experiments, the field-aligned and lattice-guided tetrahedral meshing provides two benefits: (1) for isotropic tetrahedral meshing, having a direction field to guide the mesh could potentially improve mesh quality, especially for models with rotational features; (2) for anisotropic tetrahedral meshing, having BCC/FCC to guide the mesh can generate higher quality meshes as compared to state-of-the-art methods.



(a) *BCC lattice* **(b)** *FCC lattice*

Figure 1: *BCC and FCC lattice*

2. Backgrounds and Related Works

2.1. BCC and FCC Lattices

A Body-Centered-Cubic (BCC) lattice is formed by vertices of cubic cells along with cell centers as shown in Fig. 1a. Their Voronoi cells are truncated octahedra, and each dual Delaunay tetrahedral element has dihedral angles [60°(4), 90°(2)], called *BCC tetrahedron*. BCC lattice is the optimal lattice quantizer in terms of the mean squared error [BS83]. Every point in the BCC lattice has identical one-ring neighbor structures, consisting of 8 nearest neighbors and 6 second-nearest neighbors. For a regular cubic lattice with unit edge length, the 14 one-ring neighbors of each vertex consist of the set:

$$\begin{aligned}\textbf{Onering}_{BCC} =& \{\pm 0.5, \pm 0.5, \pm 0.5\} \cup \\ & \{\{\pm 1, 0, 0\}, \{0, \pm 1, 0\}, \{0, 0, \pm 1\}\}.\end{aligned} \tag{1}$$

A Face-Centered-Cubic (FCC) lattice consists of vertices of cubic cells and their face centers as shown in Fig. 1b. Their Voronoi cells are rhombic dodecahedra, and the dual Delaunay tetrahedral elements include two kinds of tetrahedra with dihedral angles [54.735°(4), 90°, 109.47°] and [70.528°(6)]. Comparing to BCC, FCC is preferred as a finite-element mesh generation in terms of better approximation error bounds [RO00]. Each inner vertex in FCC lattice has the same one-ring neighbor structures, denoted as:

$$\begin{aligned}\textbf{Onering}_{FCC} =& \{\{\pm 0.5, \pm 0.5, 0\}, \{\pm 0.5, 0, \pm 0.5\}, \{0, \pm 0.5, \pm 0.5\}\} \\ & \cup \{\{\pm 1, 0, 0\}, \{0, \pm 1, 0\}, \{0, 0, \pm 1\}\}.\end{aligned} \tag{2}$$

The number of one-ring neighbors N_{or} for each vertex is: $N_{or} = 14$ for BCC and $N_{or} = 18$ for FCC.

2.2. Frame Field

A frame field specifies the desired direction and (possibly anisotropic) metric behavior of resulting mesh on a certain surface or in a volume domain. In 3D volume meshing, a *discrete frame field* is usually defined on a given tetrahedral mesh. Each vertex or tetrahedron i is associated with a 3×3 matrix $\mathbf{T}_i$, which is expanded by three vectors, i.e., $\mathbf{T}_i = [\mathbf{t}_{i_1}, \mathbf{t}_{i_2}, \mathbf{t}_{i_3}]$. We suppose every $\mathbf{T}_i$ is of full rank. Each vector may have non-unit length, and the three vectors may be non-orthogonal to each other. If every $\mathbf{T}_i$ is orthonormal, the frame field reduces to a cross field.

For every matrix $\mathbf{T}_i$, there is a corresponding linear transformation matrix $\mathbf{B}_i = \mathbf{T}_i^{-1}$. The matrix $\mathbf{B}_i$ locally maps a stretched/sheared cube with edges $\mathbf{t}_{i_1}, \mathbf{t}_{i_2}, \mathbf{t}_{i_3}$ to a regular cube aligned with the three axes of the 3D Cartesian coordinate system. $\mathbf{B}_i$ also defines a local mapping of any vector from the anisotropic space into isotropic space. The details are discussed in Sec. 3.1.

2.3. Related Works on Tetrahedral Meshing

Tetrahedral meshing has been studied for several decades in both engineering and computer science fields. The algorithms can be categorized into four types: advancing front methods, octree and lattice methods, Delaunay-based methods, and particle-based methods.

Advancing Front Methods [MH95] start from the domain boundary and gradually add vertices and tetrahedra until the domain is completely meshed. They preserve the domain boundary explicitly. However, the difficulty of this type of method is to resolve the intersected tetrahedra inside the domain.

Octree and Lattice Methods: Quadtree/Octree is a Cartesian grid structure in 2D/3D. Quadtree encoding an input curve in 2D was first introduced by Yerry and Shephard [YS83], then generalized to 3D [YS84] and refined by Shephard and Georges [SG91]. These methods start from encoding the surface as an adaptive grid structure, i.e., an octree, which converts the surface to a volumetric representation. The tetrahedral meshes are then constructed by local tetrahedralization of each octree cell, with special treatment for the cells intersecting the input surface [SG91, MV92].

Similar to octree methods, lattice methods utilize a space-filling tetrahedral lattice instead of Cartesian grid as volume representation, which omits the local tetrahedralization for the interior cells. The boundary tetrahedra are deformed to preserve smooth boundary. For a better quality, finer cells are generated by Fredenthal subdivision of a grid [VdMGT97] or adaptive BCC lattice [MBF03] along the surface boundary. The isosurface stuffing method [LS07] proposed an option to cut the boundary tetrahedra to resolve the input surface. The cutting rules yield theoretical bounds for the smallest/largest dihedral angles. Doran et al. [DCB13] extended the method with A15 lattice. However, these stuffing-based methods cannot generate either field-aligned isotropic tetrahedral meshes, or anisotropic tetrahedral meshes.

Delaunay-Based Methods [CDS12] can be further categorized into two groups: (1) Delaunay refinement-based methods [Che97, JAYB15, Si15] improve the mesh quality by inserting new vertices until certain user-specified conditions are met, e.g., the minimal dihedral angle. (2) Variational methods iteratively minimize an energy, e.g., CVT [DW03, ACSYD05, LWL*09], ODT [CH11, CWL*14], by optimizing positions of vertices and their connectivities.

Particle-Based Methods use repulsive particles to resample surfaces or volumes. It was first introduced by Turk [Tur92], and later extended by Witkin and Heckbert [WH05] for implicit surface meshing. They introduced Gaussian kernel to model the interaction between particles which sample an implicit surface. Researchers have tried different choices of kernels, such as a modified cotangent function with finite support [MGW05], or a bounded

cubic function [YS00], and packing ellipsoidal bubbles instead of spherical bubbles to get anisotropic tetrahedral meshes. Zhong et al. [ZGW*13] used the Gaussian kernel to model the inter-particle energy in an embedding space to solve anisotropic surface meshing. Note that the traditional Gaussian kernel is radially-symmetric. Even though it can be distorted to elliptically-symmetric under Riemannian metric, the interaction between particles still resembles packing of circles/spheres isotropically, or ellipses/ellipsoids anisotropically. This makes it impossible to explicitly control field-alignment of particles. In this paper, we propose *Gaussian Hole Kernel* as potential energy between particles to guide their distribution into either BCC or FCC pattern, which will be introduced in the next section.

3. Particle-Based Lattice Optimization

In the particle-based framework, each vertex in the resulting mesh is modeled as a particle with certain inter-particle potential energy, the derivative of which determines the inter-particle forces. The position of particles are optimized according to the forces from their neighbors until they reach the equilibrium. In the following subsection, we introduce how to design the potential energy which can guide particles to form the field-aligned lattice pattern, either BCC or FCC, when they reach the equilibrium.

3.1. Gaussian Hole Kernel

Gaussian kernel is radially-symmetric, thus defining the inter-particle energy using it resembles packing of circles/spheres in 2D/3D, as demonstrated for anisotropic triangular meshing of surfaces [ZGW*13]. Suppose two neighboring particles i and j are located at $\mathbf{p}_i$ and $\mathbf{p}_j$, respectively, their radially-symmetric energy can be defined as: $e^{-\frac{\|\mathbf{v}_{ij}\|^2}{2\sigma^2}}$, where $\mathbf{v}_{ij}=\mathbf{p}_i-\mathbf{p}_j$, and σ is the standard deviation of the Gaussian kernel. However, such radial-symmetry means that this potential energy does not have directional alignment property. In other words, given two different cross fields (of rotation only), their particle optimization results will be the same.

We need to construct a more specific potential energy to get the desired lattice structure. Once the frame field specifying the target edge direction and length of the lattice cube are given, the one-ring neighbors of a particle are fixed accordingly. Radial-symmetry is not enough to form the particular one-ring structures locally. Our goal is to force neighbor particles to fall into each others' desired one-ring neighbor positions exactly by minimizing the potential energy. To achieve such property, we place negative Gaussian kernels right at the desired one-ring neighbor positions, which is like digging a *hole* at those positions in the energy field: $-e^{-\frac{\|\mathbf{v}_{ij}-\mathbf{Onering}(k)\|^2}{2\sigma^2}}$, where $\mathbf{Onering}(k)$ is the desired offset from $\mathbf{p}_j$ to its k-th one-ring neighbor $\mathbf{p}_i$ for either BCC in Eq. (1) or FCC in Eq. (2), $k=1...N_{or}$. When we minimize such a potential energy, the neighboring particles will be pushed exactly to those *holes*. Besides that, we also include a positive Gaussian kernel at the position of the particle itself which will push its neighbors away to avoid particles being optimized to the same positions. We call this potential energy as *Gaussian Hole Kernel* (GHK):

$$E_{ij}=e^{-\frac{\|\mathbf{v}_{ij}\|^2}{2\sigma^2}}-\frac{1}{N_{or}}\sum_{k=1}^{N_{or}}e^{-\frac{\|\mathbf{v}_{ij}-\mathbf{Onering}(k)\|^2}{2\sigma^2}}. \tag{3}$$

To generate anisotropic field-aligned lattice pattern, we will transform the anisotropic lattice alignment problem to an isotropic one locally based on the given frame field. When particles form a regular cubic lattice aligned with the axes of Cartesian coordinate system, particles in the anisotropic space will exhibit a BCC/FCC pattern aligned with the desired frame field. Each particle i is associated with a matrix $\mathbf{T}_i$ expanded by three vectors $\{\mathbf{t}_{i1},\mathbf{t}_{i2},\mathbf{t}_{i3}\}$. Those three vectors define the local alignment of cubic lattice. Suppose there is no degenerate case, i.e., $|\mathbf{T}_i|\neq 0$, then the corresponding matrix $\mathbf{B}_i=\mathbf{T}_i^{-1}$ transforms the anisotropic space to an isotropic one locally: $\mathbf{B}_i\mathbf{T}_i=\mathbf{B}_i\{\mathbf{t}_{i1},\mathbf{t}_{i2},\mathbf{t}_{i3}\}=\mathbf{I}$. In other words, B_i transforms an anisotropic lattice to an isotropic one locally. If we take $\{\mathbf{t}_{i1},\mathbf{t}_{i2},\mathbf{t}_{i3}\}$ as basis of the anisotropic space, then any vector $\mathbf{v}=k_1\mathbf{t}_{i1}+k_2\mathbf{t}_{i2}+k_3\mathbf{t}_{i3}$ in the anisotropic space has a corresponding vector $\mathbf{v}'=\mathbf{B}_i\mathbf{v}=diag(k_1,k_2,k_3)\,\mathbf{I}$ in the mapped isotropic space, where the one-ring neighbors of each vertex in BCC and FCC are well defined as in Eq. (1) and Eq. (2).

Suppose there are N particles $\mathbb{V}=\{\mathbf{p}_i|i=1...N\}$. For two neighboring particles i and j, we use $\mathbf{T}_{ij}=\mathbf{T}\left(\frac{\mathbf{p}_i+\mathbf{p}_j}{2}\right)$ as the frame evaluated (T at the field sample closest to $\frac{\mathbf{p}_i+\mathbf{p}_j}{2}$) at the middle of two particles, and the correspondingly matrix $\mathbf{B}_{ij}=\mathbf{T}_{ij}^{-1}$ transforms $\mathbf{v}_{ij}$ from its anisotropic space to the isotropic one. The energy of Eq. (3) between two neighboring particles i and j can be modified as:

$$E'_{ij}=e^{-\frac{\|\mathbf{B}_{ij}\mathbf{v}_{ij}\|^2}{2\sigma^2}}-\frac{1}{N_{or}}\sum_{k=1}^{N_{or}}e^{-\frac{\|\mathbf{B}_{ij}\mathbf{v}_{ij}-\mathbf{Onering}(k)\|^2}{2\sigma^2}}. \tag{4}$$

Here σ should be proportional to the expected edge length l^* of cubic lattice. We discuss the choice of a proper value for σ in Sec. 4.

Note that we denote all the symbols in the isotropic space with a prime symbol $(')$. The energy E'_{ij} is defined in the isotropic space. The negative of first-order derivative of E'_{ij} with respect to $\mathbf{p}'_i$ is the force defined in the isotropic space: $\mathbf{f}'_{ij}=-\frac{\partial E'_{ij}}{\partial \mathbf{p}'_i}$. Since the particle positions $\mathbf{p}_i$ are optimized in the anisotropic space, we transform the force back to the anisotropic space: $\mathbf{f}_{ij}=\mathbf{T}_{ij}\mathbf{f}'_{ij}$, which is:

$$\mathbf{f}_{ij}=\frac{\mathbf{v}_{ij}}{\sigma^2}e^{-\frac{\|\mathbf{B}_{ij}\mathbf{v}_{ij}\|^2}{2\sigma^2}}-\frac{\mathbf{T}_{ij}}{N_{or}}\sum_{k=1}^{N_{or}}\frac{\mathbf{B}_{ij}\mathbf{v}_{ij}-\mathbf{Onering}(k)}{\sigma^2}e^{-\frac{\|\mathbf{B}_{ij}\mathbf{v}_{ij}-\mathbf{Onering}(k)\|^2}{2\sigma^2}}. \tag{5}$$

Our energy definition in Eq. (4) satisfies $E'_{ij}=E'_{ji}$, and the force defined in Eq. (5) satisfies $\mathbf{f}_{ij}=-\mathbf{f}_{ji}$.

3.2. Lattice Optimization

Once the inter-particle energy is defined, the particle optimization problem is modeled as an energy minimization problem. The variables are the particle positions $\mathbb{V}=\{\mathbf{p}_i|i=1...N\}$, which are constrained in domain Ω. The problem is formulated as follows:

$$\begin{aligned}\min\quad & E(\mathbb{V})=\sum_i\sum_{j\neq i}E_{ij}\approx\sum_i\sum_{j\in\mathbb{N}(i)}E_{ij}\\ s.t.\quad & \mathbf{p}_i\in\Omega,\quad \forall i=1...N\end{aligned} \tag{6}$$

where $\mathbb{N}(i)$ is the set of neighbors of particle i within distance R. Instead of considering the inter-particle energy between every pair of particles, we only consider the energy of two particles within distance R. We call R the neighbor radius. Gaussian energy is close to 0 when $R >= 5\sigma$. This approximation affect very little to the total energy while significantly reducing the number of items in the energy summation from $O(N^2)$ to $O(N)$. We use k-d tree to query the neighbors for each particle. When the frame field has large stretching ratio, it is also necessary to adjust the query radius accordingly since k-d tree is built based on Euclidean distance. The energy and force related to particle i is E_i and $\mathbf{f}_i$, which is the sum of inter-particle energies and forces from its neighbors $\mathbb{N}(i)$.

We use L-BFGS [LN89] to minimize the energy by loosening the constraint. The particles are labeled as four types: *fixed particles*, *sharp edge particles*, *boundary particles*, and *free particles*. Fixed particles are corner points of the domain boundary. In our implementation, we simply calculate the dihedral angles between neighboring triangles to detect all the sharp edges in the input surface mesh. A corner is identified if it is shared by more than two sharp edges. During the optimization, the gradient of the sharp edge particle will be projected onto the direction of its underlying sharp edge, and the gradient of boundary particles will be projected onto the tangent plane of its boundary surface. After each round of L-BFGS optimization, we will project particles to the domain boundary if it is either outside the domain or inside but close to the boundary. If a boundary particle is close to a sharp edge, then it is projected to the sharp edge and labeled as a sharp edge particle. This is used to maintain the constraint in Eq.(6).

The details of our L-BFGS particle optimization algorithm are illustrated in Alg. 1.

Algorithm 1: L-BFGS Particle Optimization Algorithm

Input: l^*, Ω, $\mathbf{T}$, $\mathbb{V}$
Output: Optimized $\mathbb{V}$

while stopping criteria not satisfied **do**
  Build k-d tree for $\mathbb{V}$;
  $E \leftarrow 0$;
  foreach $\mathbf{p}_i \in \{\mathbb{V} - fixedparticles\}$ **do**
    Query the neighbors $\mathbb{N}(i)$ from k-d tree;
    Calculate E_i and $\mathbf{f}_i$;
    if $\mathbf{p}_i$ is a sharp edge particle or a boundary particle **then**
      Update $\mathbf{f}_i$;
    end
    $E \leftarrow E + E_i$;
  end
  Run L-BFGS with E and $\{\mathbf{f}_i | i = 1...N\}$ to update $\mathbb{V}$;
end
foreach $\mathbf{p}_i \in \mathbb{V}$ **do**
  if $\mathbf{p}_i$ is outside of domain **or** its distance to boundary $\leq 0.3l^*$ **then**
    Project and mark it as a boundary particle;
  end
  if $\mathbf{p}_i$ is a boundary particle **and** its distance to sharp edge is $\leq 0.3l^*$ **then**
    Project and mark it as a sharp edge particle;
  end
end

3.2.1. Particle Insertion and Deletion

Minimizing the GHK energy encourages each particle to fall into a nearby hole. If there is no initial particles near a hole, then that hole will be left empty. If more than one particles are close to a hole, then those particles will compete for that hole. So the random initialization of particles results in some regions missing particles and some regions packing with extra particles. Hence we need a *particle insertion and deletion* algorithm to overcome this problem and obtain the desired BCC and FCC lattice patterns.

The existing mesh refinement schemes are designed based on the mesh structure, e.g., inserting a vertex at the center of an edge or the centroid of a face. In the particle optimization stage, we do not build the mesh, which provides efficency especially for anisotropic cases. Inspired by the existing mesh refinement scheme, we design the following "mesh-free" insertion and deletion schemes.

Particle Deletion Scheme: Without connectivity, each particle does not have a well-defined one-ring neighbor. But we can query the neighbors $\mathbb{N}(i)$ using k-d tree for any particle i. After that we calculate the anisotropic distance to its neighbors and sort the distance in ascending order. Suppose we store the sorted distance in array D_i, the particle i is deleted if any of the following condition holds:

- $D_i[0] < 0.5 * l_{closest}$;
- $\frac{1}{2}\sum_{k=0}^{1} D_i[k] < 0.75 l_{closest}$;
- $\frac{1}{4}\sum_{k=0}^{3} D_i[k] < 0.85 l_{closest}$;
- $\frac{1}{6}\sum_{k=0}^{5} D_i[k] < 0.9 l_{closest}$;
- $\frac{1}{8}\sum_{k=0}^{7} D_i[k] < 0.95 l_{closest}$;

where l_{cloest} is the closest one-ring neighbor distance in the lattice, e.g., $l_{closest} = \sqrt{3}/2 l^*$ for BCC, and $l_{closest} = \sqrt{2}/2 l^*$ for FCC. We denote the set of particles to be deleted as $\mathbb{S}_D$. The general rule of setting the coefficients is stricter constrain for the average distance of more neighbors. The coefficients given above are experimentally chosen based on our observation in the experiments, and used in all results in the paper.

Particle Insertion Scheme: Unlike the particle deletion scheme, the first step of particle insertion is to get the insertion candidates. Inspired by one-ring structure of the desired lattice, we collect the candidate set by going through each particle and add all expected positions of its one-ring neighbors to the candidate set $\mathbb{S}_I = \cup_{i=1}^{N} \cup_{k=1}^{N_{or}} \{p_i + \mathbf{T}_i * \mathbf{Onering}(k)\}$, where $\mathbf{T}_i$ is the frame field at particle i. If a candidate is outside the domain, we will project it to the domain boundary. The coinciding duplicates will be removed from the set $\mathbb{S}_I$ and also candidates coinciding with any particle will also be removed. We define two positions as coincidence if their distance is less than $0.1l^*$. After filtering the candidate set based on coincidence, we will calculate GHK energy for each remaining candidate and sort them in ascending order. Then we examine candidates one by one and pick a candidate if the nearest particle, including previously picked candidates, is at least $0.75 l_{cloest}$ away. Too small insertion threshold will cause unnecessary vertices being

Algorithm 2: Particle-Based Lattice Optimization Algorithm

Input: l^*, Ω, **T**
Output: particle set $\mathbb{V}$

Estimate vertex number N;
Randomly initialize $\mathbb{V}$; Optimize $\mathbb{V}$ by Alg. 1;
for $i \leftarrow 0$ **to** MaxRoundNum **do**
  Apply particle deletion scheme, $\mathbb{S}_D \leftarrow$ deleted particles;
  $\mathbb{V} \leftarrow \mathbb{V} \setminus \mathbb{S}_D$;
  Apply particle insertion scheme, $\mathbb{S}_I \leftarrow$ inserted particles;
  $\mathbb{V} \leftarrow \mathbb{V} \cup \mathbb{S}_I$;
  if $|\mathbb{S}_D| + |\mathbb{S}_I| == 0$ **then**
    break;
  end
  Optimize $\mathbb{V}$ by Alg. 1;
end
Optimize $\mathbb{V}$ by Alg. 1;

inserted and may slow the convergence. Too large insertion threshold will insert less vertices than required. The coefficient 0.75 is set according to our observation in the experiments.

The particle deletion and insertion schemes are performed after each round of L-BFGS optimization. In an L-BFGS optimization round, the particle number is fixed. After particle deletion and insertion, another round of L-BFGS optimization is performed. As the optimization processed, less particles are deleted and inserted. The overall lattice optimization process is given in Alg. 2.

The complexity of our particle optimization algorithm is related to the number of particles. A small target edge length indicates large number of particles, which is time-consuming. To speed up the optimization, we can start with the particle optimization (Alg. 2) by setting the target edge length as $2l^*$. After the optimization is completed, we get a particle set $\mathbb{V}$ with edge length $2l^*$. Using a similar strategy as the particle insertion scheme, we collect the candidates to refine the particle set. For each particle, the point which is l^* distance away along the frame field vectors is added to the refined candidates $\mathbb{V}_r$. After removing the coincidence candidates in $\mathbb{V}_r$, we take the particles in $\mathbb{V}$ and $\mathbb{V}_r$ as initial particles, and start another round of optimization by setting the target edge length as l^*. With a better initialization, the optimization converges much faster. If l^* is too small, we can start with $2^k l^*$ and do the above trick for k iterations. Such refinement strategy not only accelerates the optimization, but also helps converge to a better result. Both gradient norm $EpsG$ and max iteration number $MaxIts$ are set as the stopping criteria in Alg. 1. $EpsG$ is set to $1e{-}2$ for each round, except the last round, where $EpsG = 1e{-}4$. If $k = 0$, $MaxRoundNum$ is set to 16. If $k = 1$, $MaxRoundNum = 8$. If round number is less than 5, $MaxIts$ is set tp 13, otherwise it is set to 8. The last round is without the $MaxIts$ as stopping criteria. We use the kitten Model as an example to show the energy decreasing with respect to computation time.

3.3. Tetrahedral Mesh Generation

After particle optimization, we will connect the particles to build a tetrahedral mesh. Restricted Voronoi Diagram (RVD) [YLL*09] is used to build the surface boundary using the boundary particles. We use the RVD class provided by GEOGRAM [Lév15]. Once we get the boundary triangle mesh, we perform the restricted Delaunay tetrahedralization by TetGen [Si15], which does not consider the anisotropic frame field. To get a tetrahedral mesh with respect to the frame field, we perform a set of topological operations [She02b], by using the Gradient-Based Shape Matching Energy [NZL*17] as guidance to flip the tetrahedral mesh.

4. Experiments

We compare our methods with the state-of-the-art methods [JAYB15, FLSG14, ZGW*13]. The implementation of our algorithms are based in C++. The experiments are conducted on a workstation with Intel(R) Xeon E5645 2.40GHz CPU and 32GB DDR3 RAM. The calcuation of energy and force for each particle is independent, so we speed up the program with OpenMP. The input of our program includes a volume domain, its associated frame field, and also the target edge length l^* of the cubic lattice defined in the isotropic space.

Frame Field: Frame fields are given as an input. Several existing state-of-the-art algorithms can be used to generate a high-quality cross field for any arbitrary volumetric domain [HTWB11, RSL16, GJTP17, SVB17]. For the convenience, we denote the discrete cross field as **D**. We also test our methods with some user-designed frame fields, e.g., rotation along y-axis on torus and highly anisotropic frame fields on cubes.

Quality Metrics: To calculate the quality of anisotropic tetrahedral meshes, we first transform the elements τ from anisotropic space to isotropic space τ'. Many anisotropic mesh quality metrics are discussed in [She02a]. We measure the quality by *dihedral angles* θ, *edge-radius ratio* $\rho = \sqrt{6} e_{min}/4r_{circ}$, and *condition* $\kappa = 3\sqrt{6} v_{\tau'} / (2 l_{rms} * A_{rms})$, where r_{circ} is the circumradius, e_{min} is the shortest edge length, $v_{\tau'}$ is the volume, l_{rms} and A_{rms} are the root mean square of edge lengths and face areas of a tetrahedron. When τ' is a regular tetrahedron, $\rho_{opt} = 1$, $\kappa_{opt} = 1$. We report histograms and minimum, average and standard deviation for θ_{min}, ρ and κ, denoted as θ_{min}, $\overline{\theta_{min}}$, $\sigma(\theta min)$, ρ_{min}, $\overline{\rho}$, $\sigma(\rho)$, κ_{min}, $\overline{\kappa}$, and $\sigma(\kappa)$. θ_{min} and θ_{max} are the smallest and the largest dihedral angles of a tetrahedron, respectively.

Alignment Error: The alignment quality ε is evaluated on the resulting meshes. For each edge $\mathbf{v}_{ij}$ of the resulting tetrahedral mesh, we first transform it to the isotropic space $\mathbf{v}'_{ij} = \mathbf{B}_{ij}\mathbf{v}_{ij}$, then the smallest angle between $\mathbf{v}'_{ij}$ and vectors in **Onering** is used to measure the alignment error of edge $\mathbf{v}_{ij}$, i.e., $\varepsilon = \min_{k=1}^{N_{or}} \arccos\left(\frac{\mathbf{v}'^{\top}_{ij} \cdot \mathbf{Onering}(k)}{\|\mathbf{v}'_{ij}\| \cdot \|\mathbf{Onering}(k)\|}\right)$. Histograms, mean $\overline{\varepsilon}$ and $\sigma(\varepsilon)$ are reported for the result meshes.

Experiment Parameters: Our experiments show that setting σ in the range $[0.25 l_{closest}, 0.35 l_{closest}]$ has the similar performance, so we use $0.3 l_{closest}$ for all the experiments. The neighbor radius R is set as $1.3 l^*$, which includes all the one-ring neighbors. The

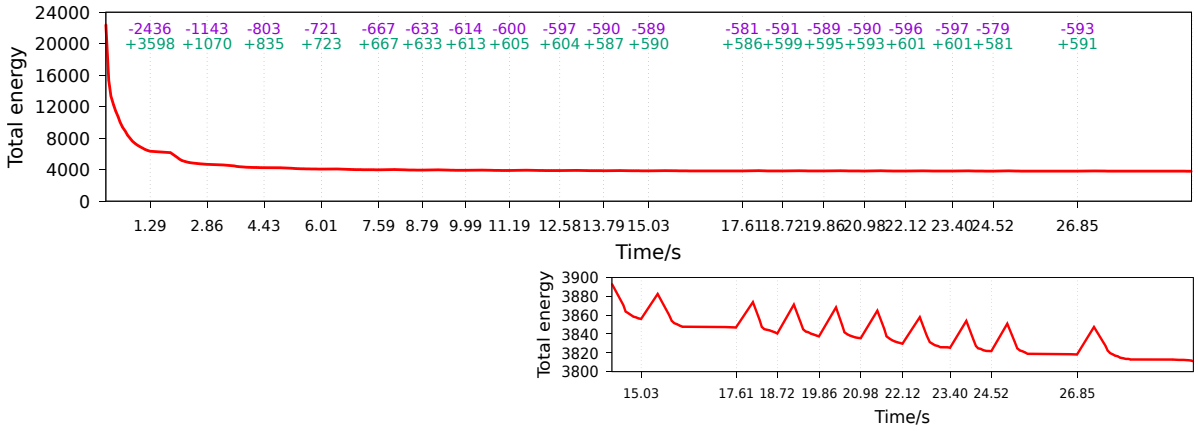


Figure 2: *The energy curve of Kitten Model about BCC optimization with 20 rounds of LBFGS optimization. The purple and green numbers are the number of being deleted and inserted at the end of each round respectively. The zoom-in view of the last 10 rounds is also provided.*

Table 1: *The mesh quality in comparison with Particle2013 [ZGW*13]. #V and #T are the numbers of vertices and tetrahedra in the result meshes. $\#T_{<20^\circ}$ and $\#T_{<40^\circ}$ are the numbers of tetrahedra with $\theta_{min} < 20^\circ$ and $\theta_{min} < 40^\circ$, respectively. The minimum/maximum, mean, and standard deviation of smallest dihedral angle θ_{min}, largest dihedral angle θ_{max}, edge radius ratio ρ, condition κ are provided. The mean and standard deviation of alignment error ε are also listed. Note that the best values are highlighted in bold for each group.*

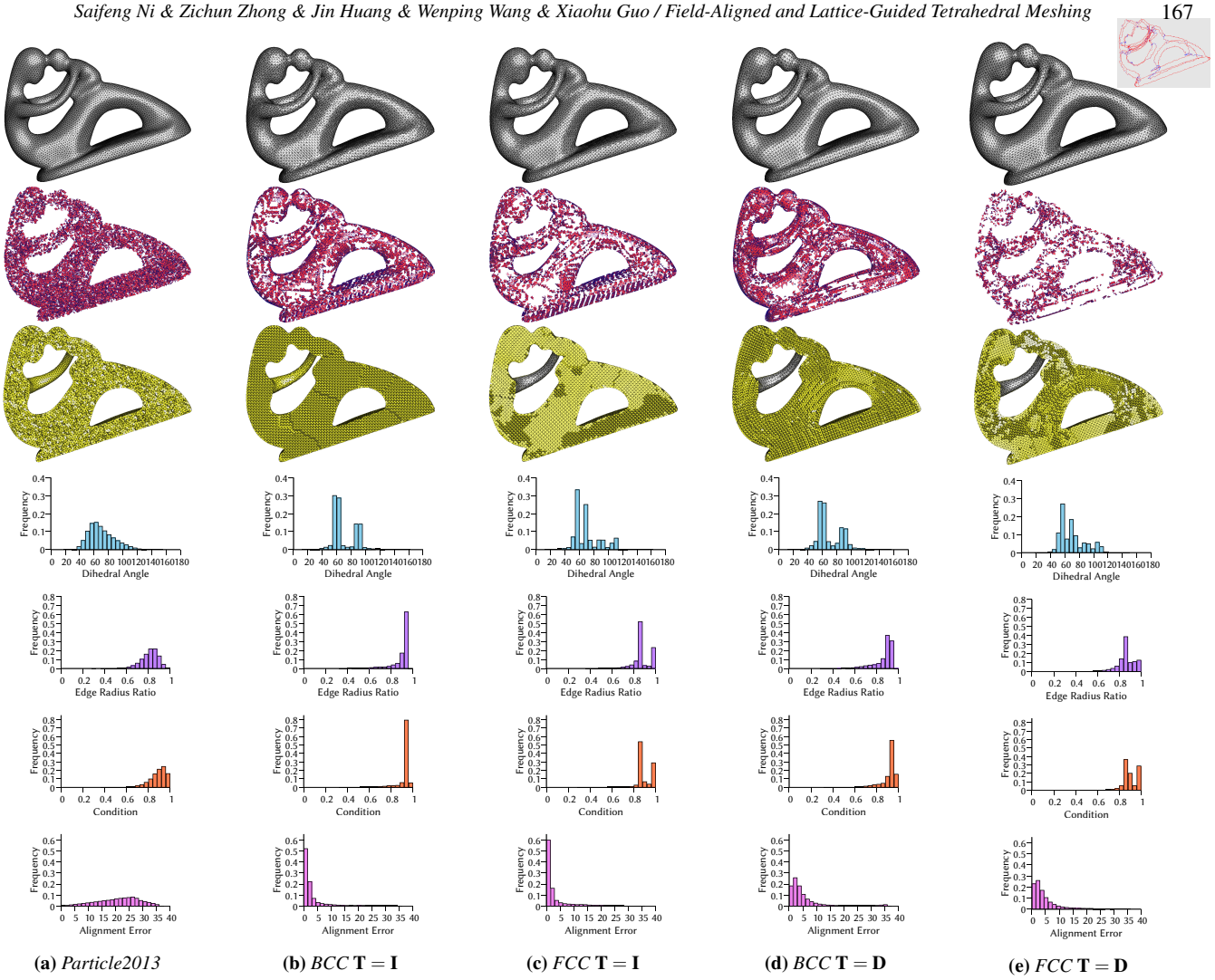
Model	**T**	Alg.	#V	$\theta_{min}/\overline{\theta_{min}}/\sigma(\theta_{min})$	$\theta_{max}/\overline{\theta_{max}}/\sigma(\theta_{max})$	$\rho_{min}/\overline{\rho}/\sigma(\rho)$	$\kappa_{min}/\overline{\kappa}/\sigma(\kappa)$	$\overline{\varepsilon}/\sigma(\varepsilon)$	$\#T_{<20^\circ}$	$\#T_{<40^\circ}$	#T	Time(s)
Bunny	n/a	Particle2013	60,000	15.73/52.93/7.00	156.62/94.90/11.65	**0.39**/0.82/0.07	0.31/0.89/0.07	20.88/7.77	3	13,359	319,562	143.60
Bunny	**I**	BCC	59,504	16.29/57.55/4.61	159.50/**91.93**/6.35	0.30/**0.91**/0.07	0.30/**0.94**/0.05	2.61/4.86	18	5,944	331,845	**55.52**
Bunny	**I**	FCC	57,533	17.98/**58.06**/7.83	157.45/96.64/16.68	0.35/0.89/0.08	0.33/0.90/0.07	**2.25**/4.35	2	5,143	318,470	94.84
Bunny	**D**	BCC	61,836	**18.72**/55.83/5.21	**148.07**/93.32/7.79	0.33/0.88/0.08	**0.42**/0.92/0.06	5.04/6.45	**1**	7,933	348,522	65.70
Bunny	**D**	FCC	60,324	17.50/56.36/7.19	153.16/96.27/14.75	0.31/0.87/0.07	0.35/0.91/0.06	4.20/4.43	2	**4,510**	336,753	137.79
Fertility	n/a	Particle2013	40,002	18.88/52.58/7.12	152.62/95.20/11.71	0.31/0.81/0.08	0.38/0.89/0.07	20.94/7.71	1	10,020	202,173	**33.77**
Fertility	**I**	BCC	41,188	16.86/56.59/5.88	158.44/**93.02**/8.25	0.35/**0.89**/0.09	0.32/**0.93**/0.07	3.70/6.07	26	7,286	218,319	40.60
Fertility	**I**	FCC	39,594	15.30/**57.09**/8.24	157.75/96.94/16.37	0.37/0.87/0.08	0.32/0.90/0.08	**3.29**/5.25	5	6,241	207,638	61.02
Fertility	**D**	BCC	42,594	17.91/55.23/5.97	154.92/94.30/9.18	0.30/0.87/0.09	0.35/0.92/0.07	6.05/7.52	6	7,664	229,902	51.88
Fertility	**D**	FCC	42,015	**22.17**/56.13/7.28	**144.17**/96.27/14.52	**0.38**/0.86/0.08	**0.45**/0.90/0.07	4.72/4.95	**0**	**3,697**	223,298	75.16
Kitten	n/a	Particle2013	60,000	15.63/52.47/7.90	158.25/95.15/11.65	0.32/0.80/0.10	0.31/0.88/0.08	20.83/7.78	28	23,988	305,286	92.65
Kitten	**I**	BCC	59,913	14.53/57.31/4.75	164.59/**92.02**/6.55	0.27/**0.91**/0.07	0.24/**0.93**/0.05	2.82/4.83	40	6,335	335,237	**71.28**
Kitten	**I**	FCC	56,499	16.46/**58.30**/8.13	156.81/96.84/17.37	0.35/0.89/0.08	0.32/0.90/0.07	**1.90**/4.37	8	5,430	313,779	93.27
Kitten	**D**	BCC	62,124	20.86/55.65/5.21	150.06/93.38/7.85	0.31/0.88/0.07	0.41/0.92/0.06	5.08/6.40	**0**	7,815	351,849	71.80
Kitten	**D**	FCC	59,998	**22.47**/56.31/7.24	**147.43**/96.33/14.76	0.30/0.87/0.07	**0.43**/0.91/0.06	4.21/4.40	**0**	**4,491**	336,382	135.46

Gaussian Hole Kernel definition in Eq. (4) takes the inverse of N_{or} as the weight for the negative Gaussian kernels. This is to balance the force used to push particles away and the forces to drag particles to holes. Larger weight will result in more coinciding particles, while with smaller weight, particles are more evenly pushed away but may be more off the desired one-ring structure. However, after a few rounds of optimization with deletion and insertion scheme, the performance of different weights are similar. The other parameters for deletion, insertion and projection of particles are given in the Sec. 3.2. All the experiments are conducted with the same parameters.

Comparison with Particle2013 [ZGW*13]: Zhong et al.'s method [ZGW*13] used the traditional Gaussian kernel for generating anisotropic triangular mesh, and can be trivially extended for tetrahedral mesh generation. We compare our Gaussian hole kernel methods, named *BCC* and *FCC*, with their method, named *Particle2013*, on three models as shown in Fig. 3 and Table. 1. Particle2013 cannot achieve field-aligned meshing results for discrete cross fields. The experiments of our method use both $\mathbf{T} = \mathbf{I}$ and $\mathbf{T} = \mathbf{D}$ (discrete cross fields). Under either rotation field, our method achieves higher quality, e.g., about 3°to 5°growth on $\overline{\theta_{min}}$, 0.05 to 0.11 gain on $\overline{rho}$ and thousands less of tetrahedra in $\#T_{<40^\circ}$. Those gains are coming from the lattice-guided alignment, producing high quality BCC and FCC tetrahedra.

BCC alignment provides higher $\overline{\rho}$ and $\overline{\kappa}$ than FCC alignment. FCC alignment creates higher $\overline{\theta_{min}}$ but also higher $\overline{\theta_{max}}$ than BCC alignment. FCC alignment has smaller $\#T_{<40^\circ}$ and it also has smaller alignment error. Unlike identity field **I**, a discrete cross field usually contains singularities inside the volume domain. The upper right corner of Fig. 3 shows the singularities of **D** inside the Fertility model. Alignment to **D** is harder than alignment to **I**, which explains the slight decline in quality. The advantage of rotation alignment will be illustrated in the later part.

Comparisons with CVT and ODT: We also compare with the



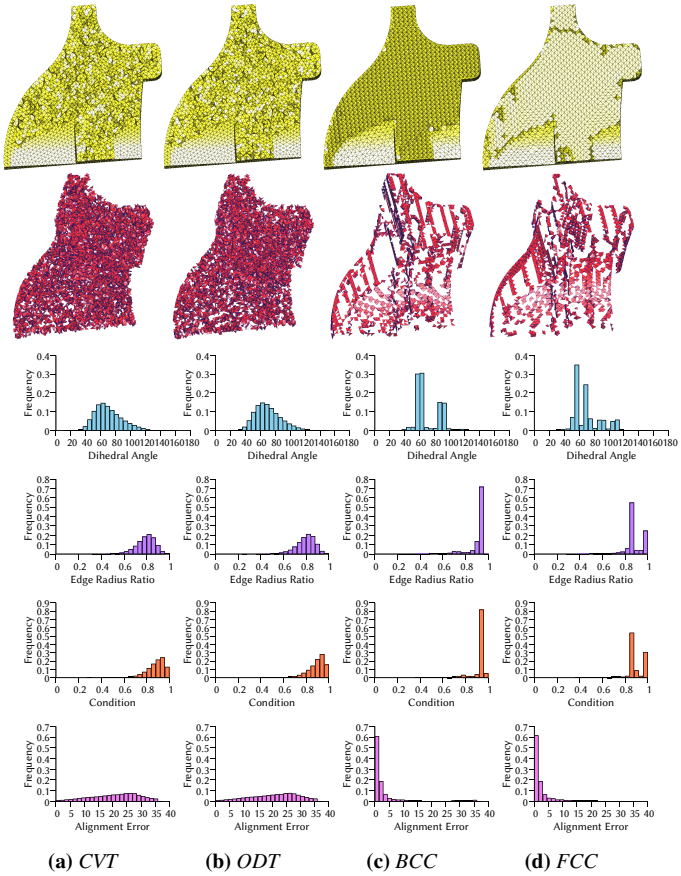
(a) *Particle2013* **(b)** *BCC* $\mathbf{T} = \mathbf{I}$ **(c)** *FCC* $\mathbf{T} = \mathbf{I}$ **(d)** *BCC* $\mathbf{T} = \mathbf{D}$ **(e)** *FCC* $\mathbf{T} = \mathbf{D}$

Figure 3: *Comparison with the traditional Gaussian kernel method proposed in Particle2013 [ZGW*13]. The first row is the result on surface. The second shows the tetrahedra with* $\theta_{min} < 40°$. *The third row shows the clipping views. The following rows are histograms of dihedral angles, edge radius ratio, condition, and alignment error.*

CVT and ODT methods to see whether BCC and FCC alignment will improve the mesh qualities. We use their implementations in CGAL [CGA17,JAYB15] for the comparison. Since both CVT and ODT energies do not support field-alignment in tetrahedral meshing, we set the frame field as $\mathbf{I}$ for our method. Our particle-based optimization well generates the BCC and FCC patterns as shown in Fig. 4 and Fig. 5. The dihedral angle histograms of BCC results have two peeks around 60°and 90°, and one peak in the histograms of ρ and κ. In FCC results, the histograms of ρ and κ have two peeks: the right ones are caused by regular tetrahedra in the meshes and the left ones are the other tetrahedra with dihedral angle $[54.735°(4), 90°, 109.47°]$, which also explains the peeks in the dihedral angle histogram. The tetrahedra with $\theta_{min} < 40°$ are on boundary surfaces. The more detailed quality statistics are given in Table. 2. Our methods have about 6°to 10°growth on $\overline{\theta_{min}}$, 0.05 to 0.12 increment on $\overline{\rho}$ and much less elements with $\theta_{min} < 40°$. Besides that, our optimization is faster since there is no computation of Voronoi diagram or connectivity in each iteration.

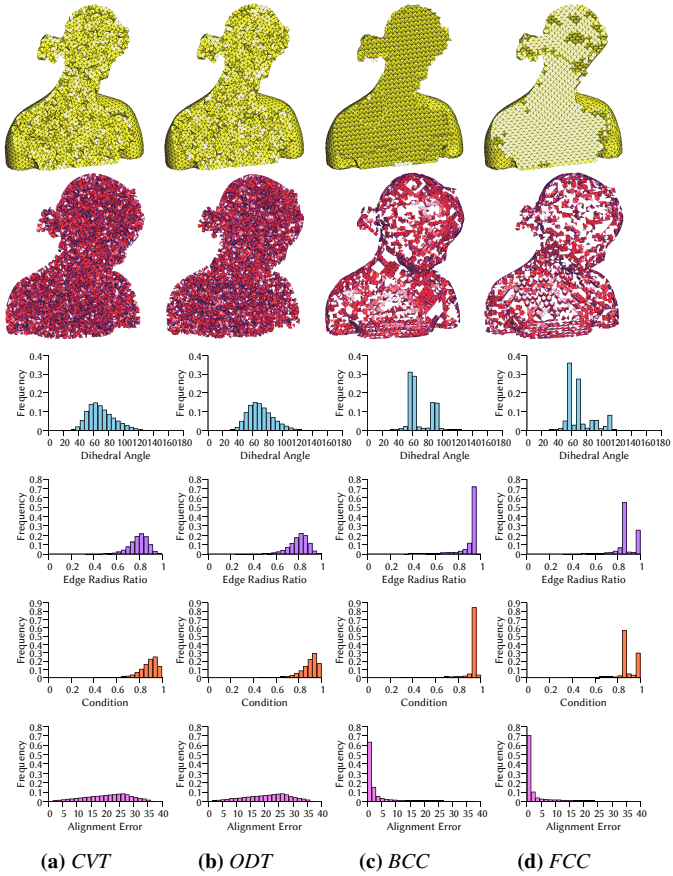
Rotation Field Alignment for Improving Mesh Quality: After showing our better mesh quality results as compared to Particle2013, CVT, and ODT methods, we would like to show that for some models with rotational features, the alignment with its rotation field can produce meshes with better quality. We use two models for such illustration: Torus (Fig. 6) and Fancyring (Fig. 7). The frame field we tried on Torus is the rotation along y-axis $\mathbf{R}_y$, and the frame field for Fancyring is a discrete cross field generated by [HTWB11], denoted as $\mathbf{D}$. We compare them with the results generated by identity field $\mathbf{I}$. It can be seen from these two experiments: if the rotation field aligns very well with the shape or the features of the geometry, we can get better shape approximation as well as higher tetrahedral qualities as shown in Table. 3.

Field Alignment for Anisotropic Tetrahedral Meshing:



(a) *CVT* (b) *ODT* (c) *BCC* (d) *FCC*

Figure 4: *Comparisons with CVT and ODT on Fandisk model. The first row shows the clipping views. The second row shows the tetrahedra with $\theta_{min} < 40°$. The following rows are histograms of dihedral angles, edge radius ratio, condition, and alignment error.*



(a) *CVT* (b) *ODT* (c) *BCC* (d) *FCC*

Figure 5: *Comparisons with CVT and ODT on Bimba model. The red tetrahedra shown in the second row have $\theta_{min} < 40°$.*

To further explore the performance of our field-aligned and lattice-guided methods, we conduct experiments using the highly anisotropic fields on Cube, and compare with LCT method [FLSG14] as shown in Fig. 8 and Fig. 9. Compared with LCT, our BCC and FCC results show higher quality due to the strong directional control and the advantage of lattice-alignment. The detailed quality statistics are given in Table. 4.

Robustness: We demonstrate the robustness of our method by experiments on Teddy with different numbers of vertices on the same discrete frame field. The minimal dihedral angle $\overline{\theta_{min}}$, edge radius ratio $\overline{\rho}$, condition $\overline{\kappa}$, alignment error $\overline{\varepsilon}$ of the resulting meshes, and their optimization time along with the different vertex numbers are shown in Fig. 10. The results of BCC and FCC are shown in red and blue curves, respectively.

5. Discussion and Future Work

It should be noted that our method can be easily extended to solve field-aligned anisotropic triangular meshing for surfaces. This can be achieved by defining six *holes* in the GHK of Eq. (3) on the tangent plane of surface. As shown in Fig. 11, our method can obtain better mesh quality of anisotropic triangular meshes compared to Particle2013 [ZGW*13] and LCT [FLSG14]. Here r_6 is the ratio

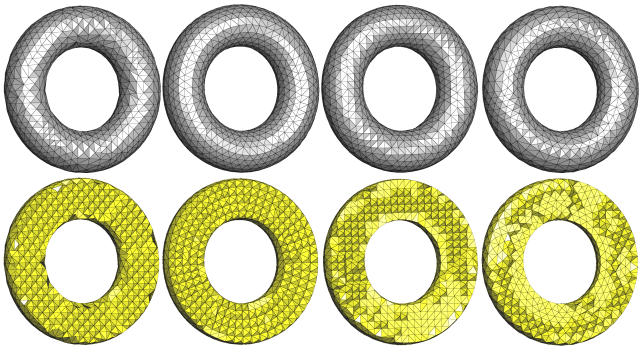
Table 2: *Comparisons with CVT and ODT. #V and #T are the numbers of vertices and tetrahedra in the output meshes. The mean value of smallest dihedral angle θ_{min}, largest dihedral angle θ_{max}, edge radius ratio ρ, condition κ, and alignment error ε are provided. The computation time is provided, which is the total time including particle optimization and mesh generation.*

Note that the best values are highlighted in bold for each group.

Model	Alg.	#V	$\theta_{min}/\overline{\theta_{min}}$	$\theta_{max}/\overline{\theta_{max}}$	$\overline{\rho}$	$\overline{\kappa}$	$\overline{\varepsilon}$	$\#T_{<20°}$	$\#T_{<40°}$	#T	Time(s)
Bimba	CVT	20,000	0.41/50.20	179.34/98.12	0.81	0.86	20.75	2,002	14,273	109,806	58.67
Bimba	ODT	20,000	2.69/51.36	175.47/93.97	0.81	0.89	20.78	543	9,757	107,400	39.34
Bimba	BCC	20,042	16.48/57.22	159.03/**92.39**	**0.90**	**0.93**	3.10	15	2,789	108,168	**20.66**
Bimba	FCC	19,477	**18.02/57.56**	**153.06**/96.94	0.88	0.90	**2.75**	**4**	**2,715**	104,340	48.03
Cube	CVT	10,000	0.50/50.05	179.22/98.32	0.81	0.86	21.01	1,037	7,311	54,221	28.34
Cube	ODT	10,000	2.06/51.19	176.65/94.21	0.81	0.89	21.02	315	5,326	52,975	18.32
Cube	BCC	9,009	44.48/59.01	120.61/**91.86**	**0.93**	**0.94**	1.00	**0**	**0**	49,152	**10.06**
Cube	FCC	9,842	**50.79/59.89**	**111.29**/96.49	0.91	0.91	**0.10**	**0**	**0**	52,728	28.61
Elephant	CVT	10,000	0.80/50.85	178.41/97.43	0.81	0.87	20.88	823	5,932	49,596	27.11
Elephant	ODT	10,000	0.80/50.99	178.41/94.53	0.80	0.88	20.85	359	5,003	48,801	16.24
Elephant	BCC	10,389	16.27/55.87	158.97/**93.46**	**0.88**	**0.92**	4.45	14	2,209	51,579	**11.23**
Elephant	FCC	10,050	**16.88/56.30**	**158.46**/97.31	0.86	0.89	**4.17**	**5**	**1,929**	49,299	21.86
Fandisk	CVT	18,000	0.49/49.92	179.22/98.54	0.81	0.86	20.92	1,876	12,939	94,977	51.24
Fandisk	ODT	18,000	3.72/50.99	173.48/94.60	0.81	0.88	20.89	639	9,172	92,939	34.00
Fandisk	BCC	18,518	**18.09**/57.25	155.71/**92.80**	**0.90**	**0.93**	3.20	**2**	**1,809**	99,005	**16.39**
Fandisk	FCC	17,920	18.08/**57.47**	**151.07**/96.70	0.88	0.90	**2.59**	3	2,077	94,509	25.50

Table 3: *The quality statistics of rotation alignment experiments on Torus and Fancyring. #V and #T are the numbers of vertices and tetrahedra of the result meshes. $\#T_{<20^\circ}$ and $\#T_{<40^\circ}$ are the numbers of tetrahedra with $\theta_{min} < 20^\circ$ and $\theta_{min} < 40^\circ$, respectively. The minimum, mean, and standard deviation of smallest dihedral angle θ_{min}, edge radius ratio ρ, condition κ are provided. The mean and standard deviation of ε are also listed. Dist is the Hausdorff distance between the boundary surfaces of the result and input meshes. Note that the best values are highlighted in bold for each group.*

Model	T	Alg.	#V	$\theta_{min}/\overline{\theta_{min}}/\sigma(\theta_{min})$	$\rho_{min}/\overline{\rho}/\sigma(\rho)$	$\kappa_{min}/\overline{\kappa}/\sigma(\kappa)$	$\overline{\varepsilon}/\sigma(\varepsilon)$	$\#T_{<20^\circ}$	$\#T_{<40^\circ}$	#T	Dist	Time(s)
Torus	**I**	BCC	2,274	18.70/54.06/7.78	0.37/0.85/0.11	0.34/0.90/0.09	6.02/7.65	5	798	10,605	0.500	6.98
Torus	**I**	FCC	2,299	15.62/**55.37**/8.70	0.34/0.85/0.09	0.30/0.89/0.08	**4.98**/6.37	1	541	10,486	0.341	**6.66**
Torus	$\mathbf{R}_y$	BCC	2,440	**28.10**/55.05/6.55	**0.43/0.86**/0.09	**0.55/0.91**/0.07	5.69/6.97	**0**	**520**	11,389	0.377	7.91
Torus	$\mathbf{R}_y$	FCC	2,347	20.87/54.38/8.34	0.35/0.85/0.10	0.42/0.89/0.09	5.60/6.19	**0**	677	11,106	**0.318**	11.76
Fancyring	**I**	BCC	4,775	17.22/51.76/8.68	0.23/0.81/0.13	0.37/0.88/0.10	9.66/9.72	14	2,587	21,420	0.542	**8.42**
Fancyring	**I**	FCC	4,810	13.79/52.05/9.56	0.19/0.80/0.12	0.29/0.87/0.10	8.15/7.70	15	2,493	21,094	0.480	14.44
Fancyring	**D**	BCC	4,187	**22.93/53.98**/6.40	0.29/**0.85**/0.08	0.37/**0.90**/0.07	8.55/8.84	**0**	**417**	18,992	**0.195**	21.56
Fancyring	**D**	FCC	3,979	22.14/53.57/6.96	**0.33**/0.84/0.08	**0.48/0.90**/0.07	**6.76**/5.76	**0**	527	17,256	0.203	14.93



(a) *BCC* ($\mathbf{B} = \mathbf{I}$) **(b)** *BCC* ($\mathbf{B} = \mathbf{R}_y$) **(c)** *FCC* ($\mathbf{B} = \mathbf{I}$) **(d)** *FCC* ($\mathbf{B} = \mathbf{R}_y$)

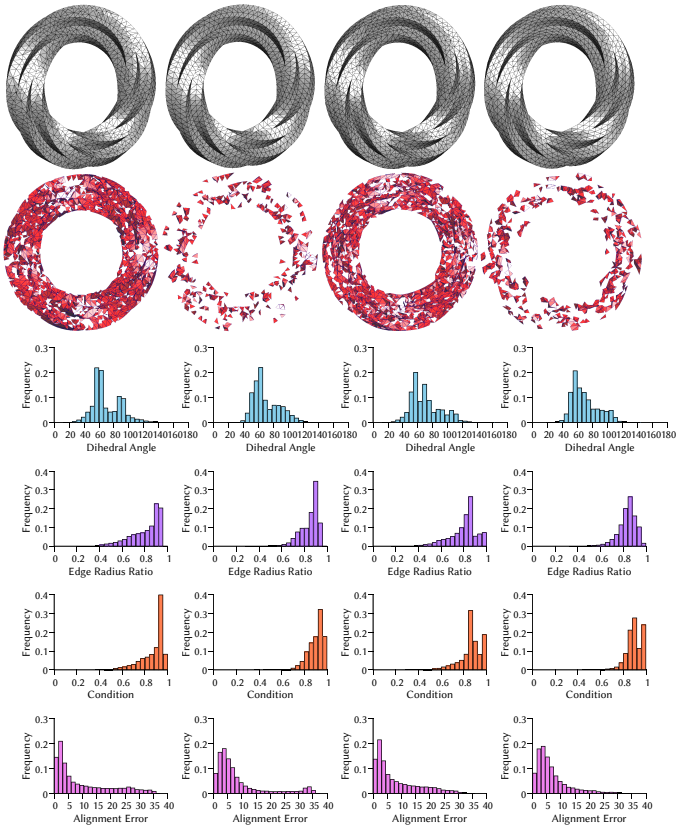
Figure 6: *BCC and FCC experiments on Torus with* $\mathbf{I}$ *and* $\mathbf{R}_y$ *rotation fields. The yellow ones are the clipping views.*

Table 4: *Statistics of mesh quality and time consumption compared with LCT [FLSG14]. #V and #T are the numbers of vertices and tetrahedra of the result meshes. $\#T_{<20^\circ}$ and $\#T_{<40^\circ}$ are the numbers of tetrahedra with $\theta_{min} < 20^\circ$ and $\theta_{min} < 40^\circ$, respectively. The mean of smallest dihedral angle θ_{min}, edge radius ratio ρ, condition κ, and computation time are provided. Note that the best values are highlighted in bold for each group.*

Model	Alg.	#V	θ_{min}	$\overline{\theta_{min}}$	$\overline{\rho}$	$\overline{\kappa}$	$\overline{\varepsilon}$	$\#T_{<20^\circ}$	$\#T_{<40^\circ}$	#T	Time(s)
	LCT	1,869	24.84	51.71	0.81	0.89	21.59	**0**	585	8,117	**8.3**
Fig. 8	BCC	2,476	36.88	56.95	**0.90**	**0.92**	3.09	**0**	**0**	11,988	16.28
	FCC	2,471	**52.85**	**58.75**	**0.90**	0.91	**0.78**	**0**	**0**	11,520	31.13
	LCT	6,338	15.51	49.87	0.78	0.88	20.94	88	4,220	31,840	72.6
Fig. 9	BCC	6,410	14.26	53.73	**0.85**	0.89	5.37	8	2,297	33,099	**56.05**
	FCC	6,498	**23.44**	**55.27**	**0.85**	**0.90**	**3.84**	**0**	**632**	32,993	87.59

of vertices with degree-6. The quality of a triangle is measured by $\xi = 4\sqrt{3}ap/h$, where a is its area, p is its perimeter and h is its longest edge length in its mapped isotropic space.

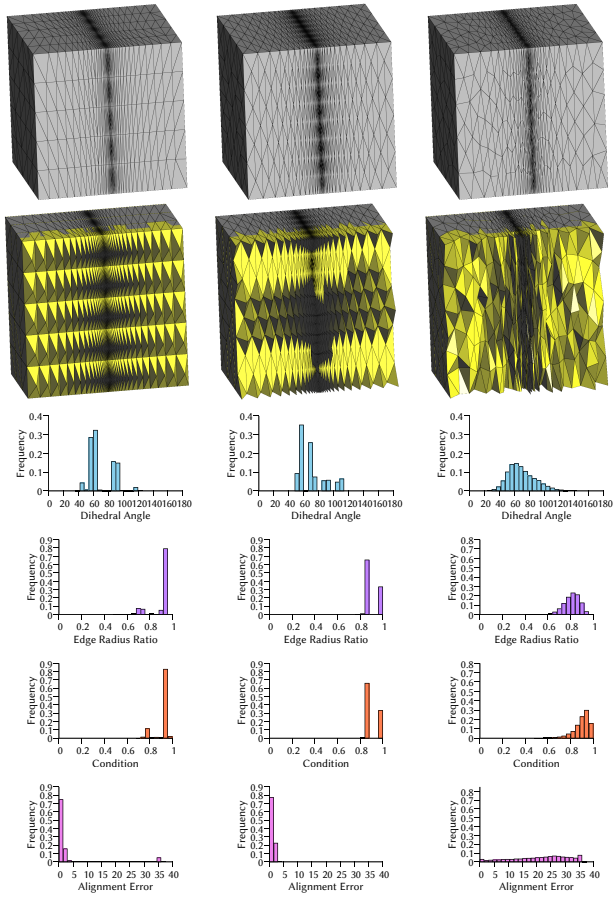
Fig. 6 and Fig. 7 show two examples that our rotation-field-aligned BCC and FCC methods might improve mesh quality for models having rotational shapes and features. In the future we would like to investigate in depth the relationship between the generation of frame-fields and the quality of field-aligned BCC and FCC meshes, in order to come up with some better field genera-



(a) *BCC* ($\mathbf{T} = \mathbf{I}$) **(b)** *BCC* ($\mathbf{T} = \mathbf{D}$) **(c)** *FCC* ($\mathbf{T} = \mathbf{I}$) **(d)** *FCC* ($\mathbf{T} = \mathbf{D}$)

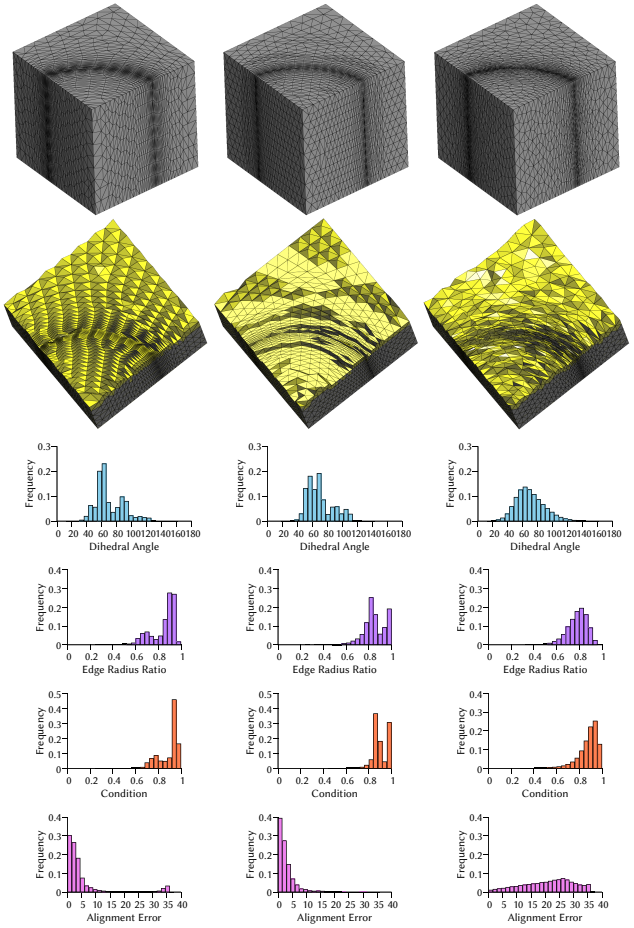
Figure 7: *BCC and FCC experiments on Fancyring with different rotation fields. The red ones are the tetrahedra with* $\theta_{min} < 40^\circ$

.



(a) *BCC* (b) *FCC* (c) *LCT*

Figure 8: *Anisotropy variation along a single direction on Cube* $[0.1,1.1]^3$. *The inverse of frame field is defined as* $\mathbf{B} = diag\left(\left(1.0125 - e^{-|x-0.6|}\right)^{-1}, 1, 1\right)$. *The second row shows the clipping views of the result tetrahedral meshes. The last four rows show the histograms of dihedral angle, edge radius ratio, condition, and alignment error.*



(a) *BCC* (b) *FCC* (c) *LCT*

Figure 9: *Cylindrical anisotropy on Cube* $[1,11]^3$. *The inverse of frame field is defined as* $\mathbf{B} = \mathbf{S} * \mathbf{R}$, *where* $\mathbf{S} = diag\left(\left(1.05 - e^{-0.01|x^2+y^2-49|}\right)^{-1}, 1, 1\right)$, *and the three columns of* $\mathbf{R}$ *are* $\left(-x/\sqrt{x^2+y^2}, y/\sqrt{x^2+y^2}, 0\right)$, $\left(-y/\sqrt{x^2+y^2}, x/\sqrt{x^2+y^2}, 0\right)$, *and* $(0,0,1)$, *respectively. The second row is the clipping view of the result tetrahedral meshes. The last four rows show the histograms of dihedral angle, edge radius ratio, condition, and alignment error.*

tion methods that are specifically tailored for such lattice meshes. In addition, we would like to investigate the possibility of other lattices [DW05b], such as A15 and Z-type configurations, etc.

6. 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 and the Wayne State University Startup Grant. Jin Huang is partially supported by NSFC (No. 61522209, 6121000).

References

[ACSYD05] ALLIEZ P., COHEN-STEINER D., YVINEC M., DESBRUN M.: Variational tetrahedral meshing. *ACM Transactions on Graphics 24*, 3 (2005), 617–625. 1, 2

[ATW13] ANDO R., THÜREY N., WOJTAN C.: Highly adaptive liquid simulations on tetrahedral meshes. *ACM Transactions on Graphics 32*, 4 (2013), 103. 1

[BS83] BARNES E., SLOANE N.: The optimal lattice quantizer in three dimensions. *SIAM Journal on Algebraic Discrete Methods 4*, 1 (1983), 30–41. 2

[BSTY15] BOISSONNAT J.-D., SHI K.-L., TOURNOIS J., YVINEC M.: Anisotropic Delaunay Meshes of Surfaces. *ACM Transactions on Graphics 34*, 2 (2015), 1–11. 1

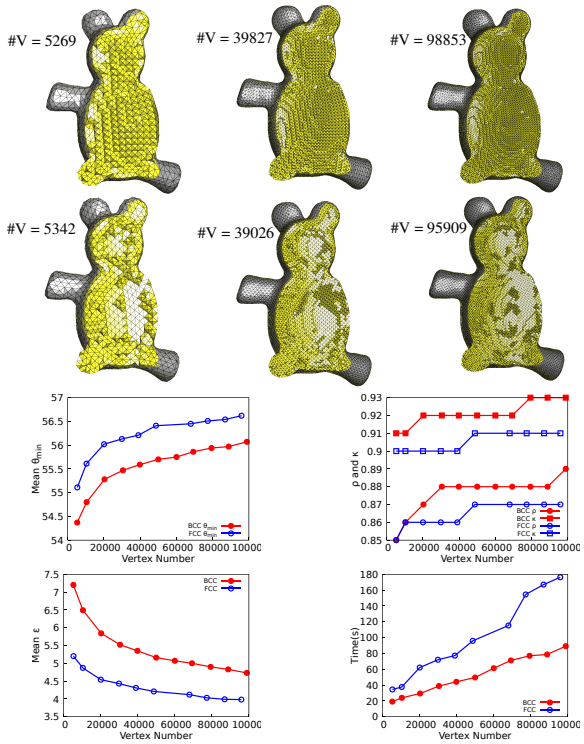
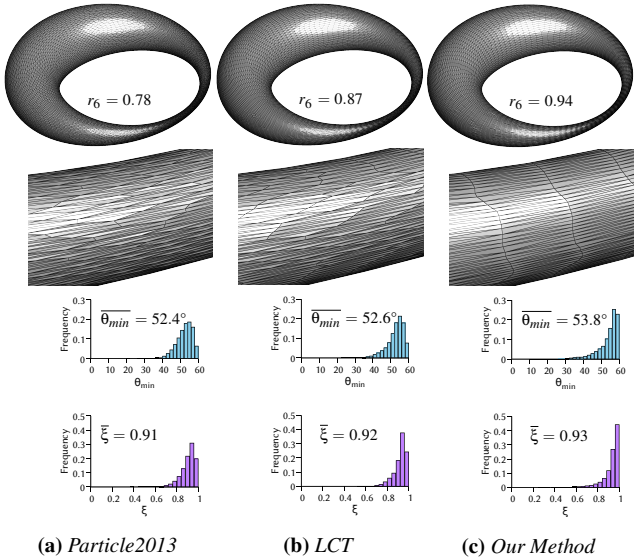


Figure 10: *The first two rows shows BCC and FCC results in different vertex numbers. The next two rows show the quality statistics and time consumption with different numbers of vertices on Teddy.*



(a) *Particle2013* **(b)** *LCT* **(c)** *Our Method*

Figure 11: *Field-aligned anisotropic triangular meshing on Cyclide, compared with Particle2013 [ZGW*13] and LCT [FLSG14]. The three columns are the results of Particle2013 method, LCT method, and our method, respectively. The first two rows show the resulting surface meshes and the zoom-in views of the narrow part. r_6 shown in the first row is the ratio of vertices with degree 6 in the result mesh. The last two rows are the histograms of $\overline{\theta_{min}}$ and $\overline{\xi}$.*

.

[CDS12] CHENG S.-W., DEY T. K., SHEWCHUK J.: *Delaunay mesh generation*. CRC Press, 2012. 2

[CGA17] CGAL, Computational Geometry Algorithms Library, 2017. http://www.cgal.org. 7

[CH11] CHEN L., HOLST M.: Efficient mesh optimization schemes based on Optimal Delaunay Triangulations. *Computer Methods in Applied Mechanics and Engineering 200*, 9-12 (2011), 967–984. 1, 2

[Che97] CHEW L. P.: Guaranteed-quality Delaunay meshing in 3D. In *Proceedings of the 13th Annual Symposium on Computational Geometry* (1997), pp. 391–393. 2

[CWL*14] CHEN Z., WANG W., LÉVY B., LIU L., SUN F.: Revisiting Optimal Delaunay Triangulation for 3D graded mesh generation. *SIAM Journal on Scientific Computing 36*, 3 (2014), A930–A954. 1, 2

[DCB13] DORAN C., CHANG A., BRIDSON R.: Isosurface stuffing improved: acute lattices and feature matching. In *ACM SIGGRAPH 2013 Talks* (2013), p. 38. 2

[DLY*18] DU X., LIU X., YAN D.-M., JIANG C., YE J., ZHANG H.: Field-aligned isotropic surface remeshing. *Computer Graphics Forum* (2018). 1

[DW03] DU Q., WANG D.: Tetrahedral mesh generation and optimization based on Centroidal Voronoi Tessellations. *International Journal for Numerical Methods in Engineering 56*, 9 (2003), 1355–1373. 1, 2

[DW05a] DU Q., WANG D.: Anisotropic Centroidal Voronoi Tessellations and their applications. *SIAM Journal on Scientific Computing 26*, 3 (2005), 737–761. 1

[DW05b] DU Q., WANG D.: The optimal Centroidal Voronoi Tessellations and the Gersho's conjecture in the three-dimensional space. *Computers & Mathematics with Applications 49*, 9-10 (2005), 1355–1373. 1, 10

[FLSG14] FU X.-M., LIU Y., SNYDER J., GUO B.: Anisotropic simplicial meshing using local convex functions. *ACM Transactions on Graphics 33*, 6 (2014), 182. 1, 5, 8, 9, 11

[GJTP17] GAO X., JAKOB W., TARINI M., PANOZZO D.: Robust hex-dominant mesh generation using field-guided polyhedral agglomeration. *ACM Transactions on Graphics 36*, 4 (2017), 114. 1, 5

[HTWB11] HUANG J., TONG Y., WEI H., BAO H.: Boundary aligned smooth 3D cross-frame field. In *ACM Transactions on Graphics* (2011), vol. 30, p. 143. 5, 7

[HZP*11] HUANG J., ZHANG M., PEI W., HUA W., BAO H.: Controllable highly regular triangulation. *Science China Information Sciences 54*, 6 (Jun 2011), 1172–1183. 1

[JAYB15] JAMIN C., ALLIEZ P., YVINEC M., BOISSONNAT J.-D.: CGALmesh: a generic framework for Delaunay mesh generation. *ACM Transactions on Mathematical Software 41*, 4 (2015), 23. 2, 5, 7

[JTPSH15] JAKOB W., TARINI M., PANOZZO D., SORKINE-HORNUNG O.: Instant field-aligned meshes. *ACM Transactions on Graphics 34*, 6 (2015), 189. 1

[KS07] KLINGNER B. M., SHEWCHUK J. R.: Aggressive tetrahedral mesh improvement. In *Proceedings of 16th International Meshing Roundtable* (2007), pp. 3–23. 1

[Lév15] LÉVY B.: GEOGRAM, a programing library of geometric algorithm, 2015. http://alice.loria.fr/software/geogram/doc/html/index.html. 5

[LN89] LIU D. C., NOCEDAL J.: On the limited memory BFGS method for large scale optimization. *Mathematical Programming 45*, 3 (1989), 503–528. 4

[LS03] LABELLE F., SHEWCHUK J. R.: Anisotropic Voronoi diagrams and guaranteed-quality anisotropic mesh generation. In *Proceedings of the nineteenth conference on Computational geometry - SCG'03* (2003), p. 191. 1

[LS07] LABELLE F., SHEWCHUK J. R.: Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. In *ACM Transactions on Graphics* (2007), vol. 26, p. 57. 1, 2

[LWL*09] LIU Y., WANG W., LÉVY B., SUN F., YAN D.-M., LU L., YANG C.: On Centroidal Voronoi Tessellation: energy smoothness and fast computation. *ACM Transactions on Graphics 28*, 4 (2009), 101. 1, 2

[MBF03] MOLINO N., BRIDSON R., FEDKIW R.: Tetrahedral mesh generation for deformable bodies. In *Proc. Symposium on Computer Animation* (2003). 2

[MGW05] MEYER M. D., GEORGEL P., WHITAKER R. T.: Robust particle systems for curvature dependent sampling of implicit surfaces. In *International Conference on Shape Modeling and Applications* (2005), pp. 124–133. 2

[MH95] MÖLLER P., HANSBO P.: On advancing front mesh generation in three dimensions. *International Journal for Numerical Methods in Engineering 38*, 21 (1995), 3551–3569. 2

[MV92] MITCHELL S. A., VAVASIS S. A.: Quality mesh generation in three dimensions. In *Proceedings of the eighth annual symposium on Computational geometry* (1992), pp. 212–221. 2

[NPPZ12] NIESER M., PALACIOS J., POLTHIER K., ZHANG E.: Hexagonal global parameterization of arbitrary surfaces. *IEEE Transactions on Visualization and Computer Graphics 18*, 6 (June 2012), 865–878. 1

[NZL*17] NI S., ZHONG Z., LIU Y., WANG W., CHEN Z., GUO X.: Sliver-suppressing tetrahedral mesh optimization with gradient-based shape matching energy. *Computer Aided Geometric Design 52* (2017), 247–261. 5

[PPTSH14] PANOZZO D., PUPPO E., TARINI M., SORKINE-HORNUNG O.: Frame fields: anisotropic and non-orthogonal cross fields. *ACM Transactions on Graphics 33*, 4 (2014), 134. 1

[RO00] RADOVITZKY R., ORTIZ M.: Tetrahedral mesh generation based on node insertion in crystal lattice arrangements and advancing-front-delaunay triangulation. *Computer Methods in Applied Mechanics and Engineering 187*, 3-4 (2000), 543–569. 2

[RSL16] RAY N., SOKOLOV D., LÉVY B.: Practical frame field generation. *ACM Transactions on Graphics 35*, 233 (2016). 5

[SG91] SHEPHARD M. S., GEORGES M. K.: Automatic three-dimensional mesh generation by the finite octree technique. *International Journal for Numerical methods in engineering 32*, 4 (1991), 709–749. 2

[She02a] SHEWCHUK J.: What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures (preprint). *University of California at Berkeley 73* (2002), 137. 5

[She02b] SHEWCHUK J. R.: Two discrete optimization algorithms for the topological improvement of tetrahedral meshes. *Unpublished manuscript 65* (2002). 5

[Si15] SI H.: TetGen, a Delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software 41*, 2 (2015), 11:1–11:36. 2, 5

[SRUL16] SOKOLOV D., RAY N., UNTEREINER L., LÉVY B.: Hexahedral-dominant meshing. *ACM Transactions on Graphics 35*, 5 (2016), 157. 1

[SVB17] SOLOMON J., VAXMAN A., BOMMES D.: Boundary element octahedral fields in volumes. *ACM Transactions on Graphics 36*, 3 (2017), 28. 5

[TSA09] TOURNOIS J., SRINIVASAN R., ALLIEZ P.: Perturbing slivers in 3D delaunay meshes. In *Proceedings of 18th International Meshing Roundtable* (2009), pp. 157–173. 1

[Tur92] TURK G.: Re-tiling polygonal surfaces. In *ACM SIGGRAPH Computer Graphics* (1992), vol. 26, pp. 55–64. 2

[VdMGT97] VELHO L., DE MIRANDA GOMES J., TERZOPOULOS D.: Implicit manifolds, triangulations and dynamics. *Neural Parallel and Scientific Computations 5* (1997), 103–120. 2

[WH05] WITKIN A. P., HECKBERT P. S.: Using particles to sample and control implicit surfaces. In *ACM SIGGRAPH 2005 Courses* (2005), p. 260. 2

[YLL*09] YAN D.-M., LÉVY B., LIU Y., SUN F., WANG W.: Isotropic remeshing with fast and exact computation of restricted Voronoi diagram. In *Computer Graphics Forum* (2009), vol. 28, pp. 1445–1454. 5

[YS83] YERRY M. A., SHEPHARD M. S.: A modified quadtree approach to finite element mesh generation. *IEEE Computer Graphics and Applications 3*, 1 (1983), 39–46. 2

[YS84] YERRY M. A., SHEPHARD M. S.: Automatic three-dimensional mesh generation by the modified-octree technique. *International Journal for Numerical Methods in Engineering 20*, 11 (1984), 1965–1990. 2

[YS00] YAMAKAWA S., SHIMADA K.: High quality anisotropic tetrahedral mesh generation via ellipsoidal bubble packing. In *Proceedings of International Meshing Roundtable* (2000), pp. 263–274. 3

[ZGW*13] ZHONG Z., GUO X., WANG W., LÉVY B., SUN F., LIU Y., MAO W., ET AL.: Particle-based anisotropic surface meshing. *ACM Transactions on Graphics 32*, 4 (2013), 99–1. 1, 3, 5, 6, 7, 8, 11