Optimal Power Diagrams via Function Approximation

Yanyang Xiao[a], Zhonggui Chen[a,c], Juan Cao[b,c,*], Yongjie Jessica Zhang[c], Cheng Wang[a]

[a]*Fujian Key Laboratory of Sensing and Computing for Smart City, School of Information Science and Engineering, Xiamen University, Xiamen 361000, China*
[b]*School of Mathematical Sciences, Xiamen University, Xiamen 361000, China*
[c]*Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA*

Abstract

In this paper, we present a novel method for generating cell complexes with anisotropy conforming to the Hessian of an arbitrary given function. This is done by variationally optimizing the discontinuous piecewise linear approximation of the given functions over power diagrams. The resulting cell complexes corresponding to the approximations are referred to as Optimal Power Diagram (OPD). A hybrid optimization technique, coupling a modified Monte Carlo method with a local search strategy, is tailored for effectively solving the specific optimization task. In contrast to the Optimal Voronoi Tessellation (OVT) method [1], our OPD method does not restrict the target functions to be convex, providing more diverse classes of tessellations of the domain. Furthermore, our OPD method generally yields smaller approximation errors than the OVT method, which uses underlaid approximants. We conduct several experiments to demonstrate the efficacy of our optimization algorithm in finding good local minima and generating high-quality anisotropic polytopal meshes.

Keywords: anisotropic meshing, optimal power diagram, optimal Voronoi tessellation, function approximation

1. Introduction

Meshes are important representations used in a wide range of applications, including geometric modeling, computer graphics, mechanical engineering and simulations. The complex geometric objects are represented as an assembly of discrete elements, e.g., triangles, quadrilaterals and polygons in 2D as well as their counterparts in higher dimensions. Numerous algorithms have been developed to produce high-quality isotropic meshes, and some of them are commercially available. While the problem of isotropic meshing has been well studied, the research on converting complex objects into an anisotropic mesh is relatively behind.

Anisotropic meshes are often advantageous in terms of computational cost and solution accuracy in finite element simulations for resolving physical problems with solutions changing more rapidly in one direction than others. In shape/functional approximation, anisotropic meshes also provide better interpolations/approximations of geometries/functions having strong directionality with fewer elements. However, many recent attempts have been exclusively focused on extending isotropic meshing to anisotropic simplicial meshing and quadrilateral/hexahedral meshing, due to their popularity. In this paper, we propose a novel Optimal Power Diagram (OPD) method for generating anisotropic meshes formed by convex polygons/polyhedrons. The specific contributions of this paper are as follows:

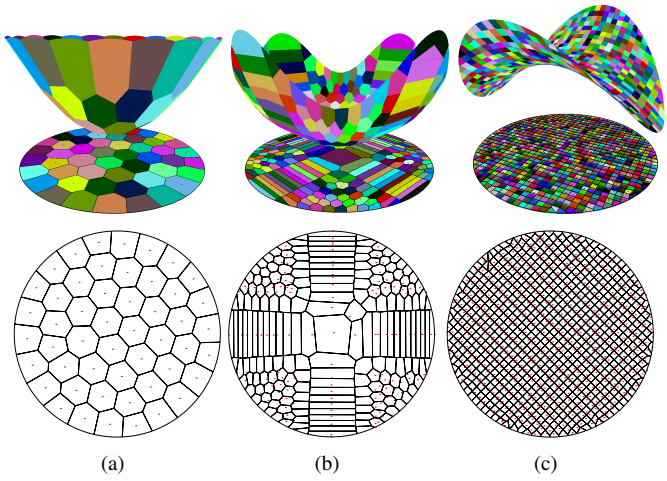


Figure 1: Optimal power diagrams based on discontinuous piecewise linear approximations of different target functions: (a) $f(x, y) = x^2 + y^2$; (b) $f(x, y) = x^4 + y^4$; and (c) $f(x, y) = xy$.

1. We extend the Optimal Voronoi Tessellation (OVT) energy function that was previously used for generating anisotropic polygonal/polyhedral meshes. From the point view of functional approximation, the OVT energy function describes the error between a convex function and its underlaid polytope in the L_1 norm. We relax the constraints for both the approximant and the target function of the OVT energy function. More precisely, the target function is not necessarily convex, which is approximated by linear functions individually defined over sub-regions in the L^2 norm.

2. We use power diagrams to represent the tessellations of

* Corresponding author. Email address: Juancao@xmu.edu.cn (Juan Cao)
Part of this work was done while Zhonggui Chen and Juan Cao were visiting the Department of Mechanical Engineering, Carnegie Mellon University.

a given domain, so that the energy minimization problem can be tackled efficiently. Minimizing the generalized energy boils down to finding the optimal partition of the domain, which is difficult in general cases. As a matter of fact, it is NP-hard to decide if a bivariate function can be approximated by a piecewise linear function with a prespecified number of facets and within an approximation error threshold, as has been shown in [2]. To make the problem solvable, we restrict the partitions to power diagrams, which have many important applications in various fields. Actually, this is not even a restriction for cell complex generation of volumes, as an arbitrarily simple cell complex in 3D can be represented by a power diagram [3].

3. A hybrid optimization algorithm is presented to efficiently solve the specific minimization problem in this paper. It is very challenging to minimize the modified energy function, due to its highly non-linear and non-convex nature. Our hybrid optimization method, combining a local search with a global optimization technique, reduces the chance of getting stuck at poor local minima and converges fast to a deep enough local minimum. Moreover, a comprehensive initialization method is proposed to tremendously improve the results.

The remainder of this paper is organized as follows. We review the related works in Section 2 and introduce some preliminary concepts and notations in Section 3. In Section 4, we extend the OVT energy function and derive closed-form expressions for its derivatives. In Section 5, we provide the overview and technical details of our hybrid optimization method for the minimization of the proposed energy function. Finally, after giving some experimental results and comparisons with the OVT method in Section 6, we conclude this paper with discussions and limitations in Section 7.

2. Related Work

In this section, we give a brief review of references that are most related to this work, with a focus on methods for anisotropic mesh generation.

Anisotropic simplicial meshing. Anisotropic simplicial meshes are triangulations of a given domain whose elements stretch along the desired directions. The anisotropy at each point of the domain is usually given as a symmetric matrix. The eigenvectors and eigenvalues of the matrix describe the desired stretching directions and magnitudes of mesh elements. Delaunay refinement is a widely used technique for generating high-quality isotropic simplicial meshes [4]. By taking anisotropy into account, various extensions to Delaunay refinement have been proposed to generate anisotropic Delaunay meshes [5, 6, 7, 8]. Anisotropic mesh quality can be measured in different ways and optimized by combining the operations of vertex redistribution and retriangulation [9, 10]. By embedding the anisotropic space into a higher dimensional isotropic one, anisotropic meshes can be obtained via constructing isotropic meshes in the embedded space [11, 12, 13].

Anisotropic Voronoi. The concept of Centroidal Voronoi Tessellation (CVT) has been successfully applied to high-quality isotropic mesh generation [14]. CVTs can be generalized to anisotropic Voronoi tessellations, by using different distance definitions [15, 16, 17, 18]. Generally, the bisectors between sites are no longer straight under the generalized distance measures, making the precise construction of anisotropic Voronoi cells computationally cumbersome. Approximation methods are thus adopted to compute anisotropic Voronoi diagrams, by clustering discrete elements of a given mesh [19, 18, 20] or constructing restricted Voronoi diagrams instead [16, 21]. In our method, power diagrams are used as the representation of convex tessellations of the domain, which allow more flexibility than Voronoi diagrams in optimization [22] and can be efficiently computed by off-the-shelf computational geometry libraries.

Function approximation based methods. Anisotropic meshes can be obtained by minimizing the approximation error between a target function f and its piecewise linear approximation. The optimal simplices with a minimal approximation error have been shown to be stretched along the two principal directions of f, with an aspect ratio equal to $\sqrt{|k_{\max}/k_{\min}|}$, where $k_{\max}$ and $k_{\min}$ are the principal curvatures [23, 24]. Chen *et al*. [25, 26] proposed the concept of optimal Delaunay triangulation which minimizes the linear interpolation error for a given convex function. Fu *et al*. [27] generalized optimal Delaunay triangulation to anisotropic metrics by constructing convex functions that locally match a given anisotropic metric. The anisotropy of the resulting mesh elements conforms to the Hessian of an input target function. A CVT can also be defined from a variational point of view, which minimizes the error between a paraboloid $f(\mathbf{x}) = |\mathbf{x}|^2$ and its underlaid piecewise linear approximation. Budninskiy *et al*. [1] gave an anisotropic extension of CVTs, called Optimal Voronoi Tessellations (OVTs), which optimizes the piecewise linear approximation of convex functions over anisotropic cell complexes. Several other methods for generating function-dependent Voronoi diagrams have been studied in [28, 29]. In this paper, we present an extension of OVTs based on the approximation of functions that are not necessarily convex.

Global optimization. Due to the highly non-linear and non-convex nature of the CVT energy, the commonly used local search methods such as the Lloyds iteration and quasi-Newton methods tend to get stuck at shallow local minima. Several global techniques were developed to search a deep enough local minimizer and even a global minimizer. For example, a Monte Carlo-minimization (MCM) based framework [30] was adopted to compute Euclidean CVT in [31], in which the Monte Carlo method was used to pass from one local minimum to the next local minimum obtained by the L-BFGS method, and ultimately to the global minimum. Similar methods were also used for generating constrained centroidal Delaunay meshes [32] on surfaces and computing the optimal Delaunay triangulation in 3D space [33]. A differential evolution based method was developed to compute the globally optimal geodesic CVT energy on triangle meshes [34]. In this paper, we tailor the MCM method to our specific problem and provide a comprehensive initializa-

tion technique to speed up the convergence.

3. Background

In this section, we introduce preliminary concepts and notation that will be used throughout the entire paper.

Power diagram. Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ be a set of n distinct sites in a compact domain $\Omega \subset \mathbb{R}^d$ and $W = \{w_i\}_{i=1}^n$ be a set of real numbers. Each site $\mathbf{x}_i$ is associated with a real number w_i, called the weight of $\mathbf{x}_i$. Then the power cell Ω_i of $\mathbf{x}_i$ is

$$\Omega_i = \{\mathbf{x} \in \Omega \mid \|\mathbf{x} - \mathbf{x}_i\|^2 - w_i \leq \|\mathbf{x} - \mathbf{x}_j\|^2 - w_j, \forall i \neq j\},$$

where $\|\cdot\|$ denotes the Euclidean norm. The power diagram of the weighted point set $(\mathbf{X}, W) = \{(\mathbf{x}_1, w_1), ..., (\mathbf{x}_n, w_n)\}$ is the cell complex formed by the collection of power cells $\{\Omega_i\}_{i=1}^n$. The power diagram is a generalization of the Voronoi diagram, which coincides with the Voronoi diagram of the sites $\mathbf{X}$ when all the weights are the same [35].

Optimal Voronoi tessellations. Let $f : \Omega \to \mathbb{R}$ be a convex function in the compact domain $\Omega \subset \mathbb{R}^d$, and $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ be a set of n distinct sites in Ω. For each site $\mathbf{x}_i$, we denote $T_i(\mathbf{x})$ the tangent hyperplane of $f(\mathbf{x})$ at $\mathbf{x}_i$. Then optimal Voronoi tessellations [1] are minimizers of

$$\mathcal{E}_{OVT}(\mathbf{X}, \mathcal{V}) = \|f - f_T\|_{L^1} = \sum_{i=1}^n \int_{V_i} (f(\mathbf{x}) - T_i(\mathbf{x}))\mathrm{d}\mathbf{x}, \qquad (1)$$

where f_T is a piecewise linear approximation of the target function f, formed by the set of tangent hyperplanes $\{T_i(\mathbf{x})\}_{i=1}^n$, and $\mathcal{V} = \{V_i\}_{i=1}^n$ is a tessellation of the domain. Since the function f is convex, tangent hyperplanes $T_i(\mathbf{x})$ are always below it, that is, $f(\mathbf{x}) \geq T_i(\mathbf{x})$ for all i. It has been shown in [1] that for a given set of sites $\mathbf{X}$, the optimal tessellation $\mathcal{V}$ is the projection of the upper envelop of $\{T_i(\mathbf{x})\}_{i=1}^n$, which can be efficiently obtained by constructing a power diagram of a set of shifted and weighted sites $\{(\mathbf{p}_i, w_i)\}_{i=1}^n$. We have

$$\mathbf{p}_i = \frac{1}{2}\nabla f(\mathbf{x}_i), \quad w_i = \frac{1}{4}|\nabla f(\mathbf{x}_i)|^2 + f(\mathbf{x}_i) - \nabla f(\mathbf{x}_i) \cdot \mathbf{x}_i.$$

The OVT energy function in Equation (1) can be minimized by a Lloyd-based approach which alternately optimizes the locations of sites and updates the tessellation of the domain. Note that for the case of $f(\mathbf{x}) = |\mathbf{x}|^2$, $\mathcal{V}$ is simply the Voronoi tessellation of X, and the OVT energy function coincides with the centroidal Voronoi tessellation energy function [14].

4. Optimal Power Diagrams

According to [23, 24], the optimal aspect ratio of linear elements is dictated by the local Hessian of a target function f, regardless whether f is convex or not. In this section, we describe how to modify the OVT function in Equation (1) to compute anisotropic cell complexes which are cast as the optimal piecewise linear approximations of given target functions that are not necessarily convex.

4.1. Formulation

Suppose $f : \Omega \to \mathbb{R}$ is a real-valued function and is continuous in the compact domain $\Omega \subset \mathbb{R}^d$. Given a tessellation $\mathcal{V}$ of the domain Ω, we can construct the best L^2 fitting hyperplane of $f(\mathbf{x})$ for each cell V_i, denoted by $P_i(\mathbf{x})$. Then we approximate f with a discontinuous piecewise linear function $f_P(\mathbf{x}) = \sum_{i=1}^n P_i(\mathbf{x})\mathbb{1}_{V_i}(\mathbf{x})$, where $\mathbb{1}_{V_i}$ is the indicator function of a cell V_i. We formulate our energy function as follows:

$$\mathcal{E}(\mathcal{V}, \mathcal{P}) = \|f - f_P\|_{L^2}^2 = \sum_{i=1}^n \int_{V_i} (f(\mathbf{x}) - P_i(\mathbf{x}))^2 \mathrm{d}\mathbf{x},$$

where $\mathcal{P} = \{P_i(\mathbf{x})\}_{i=1}^n$ is a set of linear functions. As the function f is not convex and the fitting hyperplanes cannot guarantee to be always below or above the function f, we measure the approximation error between f and f_P using the L^2 norm instead of the L^1 norm in the OVT function.

The determination of the best tessellation $\mathcal{V}$ and the corresponding best-fit hyperplanes is generally NP-hard [2]. To simplify the computation, we restrict the tessellation to a power diagram, which is uniquely determined by the site positions X and the weights W. Note that, the L^2-norm best-fit hyperplane on each cell is determined once the tessellation is given. We denote by $P_i^*(x)$ the best linear fit on a cell V_i. Hence, the above energy function essentially depends on the site positions and weights as follows:

$$\mathcal{E}_{OPD}(X, W) = \|f - f_P\|_{L^2}^2 = \sum_{i=1}^n \int_{V_i} (f(\mathbf{x}) - P_i^*(\mathbf{x}))^2 \mathrm{d}\mathbf{x}. \qquad (2)$$

Thus we refer to minimizers of the above energy function as Optimal Power Diagrams (OPD).

4.2. Derivatives of energy function

We now derive the gradient of our energy function for later use. First, we consider the derivative of energy function with respect to site position $\mathbf{x}_i$. Applying the general Leibniz rule [36] to Equation (2), we have

$$\frac{\partial \mathcal{E}_{OPD}(X, W)}{\partial \mathbf{x}_i} = \sum_{j \in J_i \cup \{i\}} \int_{V_j} \frac{\partial}{\partial \mathbf{x}_i} |f(\mathbf{x}) - P_j^*(\mathbf{x})|^2 \mathrm{d}\mathbf{x} + \sum_{j \in J_i} \int_{V_{ij}} (|f(\mathbf{x}) - P_i^*(\mathbf{x})|^2 - |f(\mathbf{x}) - P_j^*(\mathbf{x})|^2) \frac{\partial \mathbf{x}}{\partial \mathbf{x}_i} \mathbf{n} \mathrm{d}s,$$

where J_i is the indexes of sites with cells adjacent to V_i, $V_{ij} = \partial V_i \cap \partial V_j$ is the common boundary of V_i and V_j, and $\mathbf{n}$ is the outward unit normal vector on the boundary of V_i. According to the envelope theorem, the first term in the above equation vanishes. Now, let us consider the evaluation of $\partial \mathbf{x} / \partial \mathbf{x}_i \mathbf{n}$ in the second term. Note that,

$$(\mathbf{x} - \frac{\mathbf{x}_i + \mathbf{x}_j}{2}) \cdot (\mathbf{x}_j - \mathbf{x}_i) = w_i - w_j, \quad \text{for} \quad \mathbf{x} \in V_{ij}.$$

Differentiating both sides of the equation with respect to $\mathbf{x}_i$, we obtain

$$\frac{\partial \mathbf{x}}{\partial \mathbf{x}_i}(\mathbf{x}_j - \mathbf{x}_i) = \mathbf{x} - \mathbf{x}_i.$$

Thus, we have $\partial \mathbf{x}/\partial \mathbf{x}_i \mathbf{n} = (\mathbf{x} - \mathbf{x}_i)/|\mathbf{x}_j - \mathbf{x}_i|$. Then the derivative of $\mathcal{E}_{OPD}(X, W)$ with respect to $\mathbf{x}_i$ can be simplified to

$$\frac{\partial \mathcal{E}_{OPD}(X, W)}{\partial \mathbf{x}_i} = \sum_{j \in J_i} \int_{V_{ij}} (|f(\mathbf{x}) - P_i^*(\mathbf{x})|^2 - |f(\mathbf{x}) - P_j^*(\mathbf{x})|^2) \frac{\mathbf{x} - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|} \mathrm{d}s. \tag{3}$$

The derivative of $\mathcal{E}_{OPD}(X, W)$ with respect to w_i can be found in the same fashion:

$$\frac{\partial \mathcal{E}_{OPD}(X, W)}{\partial w_i} = \sum_{j \in J_i} \int_{V_{ij}} (|f(\mathbf{x}) - P_i^*(\mathbf{x})|^2 - |f(\mathbf{x}) - P_j^*(\mathbf{x})|^2) \frac{1}{2|\mathbf{x}_j - \mathbf{x}_i|} \mathrm{d}s. \tag{4}$$

5. Optimization Algorithm

Minimizing the OPD energy function in Equation (2) is difficult, due to its highly non-linear and non-convex essence. As shown in Figure 2, the energy function presents many shallow local minima when we hold all variables except one site position or weight constant. Hence, directly applying a traditional local search method suffers from a common problem, i.e., trapping in local solutions. Whereas, conventional global optimization methods are usually very time-consuming. In this section, a hybrid local and global optimization method is adopted to efficiently solve our OPD energy function. Before getting into the details of our algorithm, we first provide some observations on the optimization of the OPD energy function, which relives the influence of variables to be optimized on the objective function and gives rise to an efficient optimization method based on block coordinate update.

Note that, the OPD energy function in Equation (2) contains two types of variables (site positions and weights). One of the most straightforward methods for its optimization is using a gradient-based local search with a random initialization and without discriminating these two types of variables. As shown in Figure 3(a), starting from a set of $1,000$ sites randomly sampled from the given domain and equal weights, a gradient-based local search method (the L-BFGS method in this paper) gets trapped in a shallow local minimum when we optimize all variables simultaneously, yielding a low-quality tessellation; see Figure 3(b). Meanwhile, we find that optimizing site positions before optimizing all variables usually produces much better results, as shown in Figure 3(c). Starting from the same random initialization in Figure 3(a) and fixing the weights in the optimization, the L-BFGS method converges to a deeper local minimum, leading to a tessellation with much higher quality than the result in Figure 3(b). The intermediate result in Figure 3(c) can be further improved by applying local search again, with all variables involved in the optimization; see Figure 3(d). More long and thin cells are introduced which better convey the expected anisotropy. The plot of approximation error versus optimization iteration number also illustrates that this block-coordinate-descent-like method converges to a deeper local minimum. In other words, site positions are the most influential variables in the optimization process, while weights exert

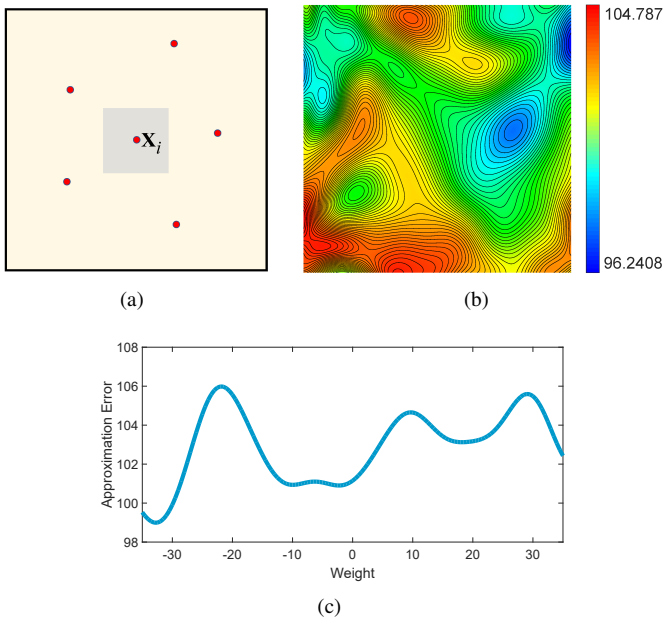


Figure 2: Energy landscape near $\mathbf{x}_i$, with an input function $f(x, y) = \sin(\pi(x + 0.5)) \cos(\pi y), -10 \leq x, y \leq 10$. (a) All sites and weights are held constant except the position or weight of site x_i; (b) variations of the energy function with respect to site $\mathbf{x}_i$; and (c) plot of approximation error with respect to weight w_i varying from -35 to 35.

a smaller influence on the minimization of the objective function. Another benefit of optimizing site positions before weight optimization is that it reduces the chance of introducing large differences on weights. A power diagram coupled with weights varying over a wide range usually has sites locating outside the corresponding cells, as shown in Figure 3(b), leading to a low-quality dual regular triangulation.

5.1. Overview of OPD optimization algorithm

Global minimization of a nonconvex objective function with many local minima is generally difficult, many efforts have been made to speed up the computation. Among them, the Monte Carlo-minimization (MCM) method [30], which combines the power of an efficient local search method to find local minima and that of the Monte Carlo method in global combinatorial optimization, is simple yet efficient to overcome the multiple-minima problem. Roughly speaking, the MCM method considers all local minimizers obtained by interleaving perturbation and local searching. A new local minimum is accepted if either it is better than the current local minimum, or it satisfies a pre-specified probability condition. Based on the above observations in Figure 3, we modify the standard MCM method to efficiently minimize the OPD energy function. As shown in Figure 4, taking a desired site number n and a target function $f(\mathbf{x})$ as input, our algorithm consists of three stages: initialization, position optimization and position-weight optimization. During position optimization and position-weight optimization, the following three steps are carried out to find the minimizer of the OPD energy function:

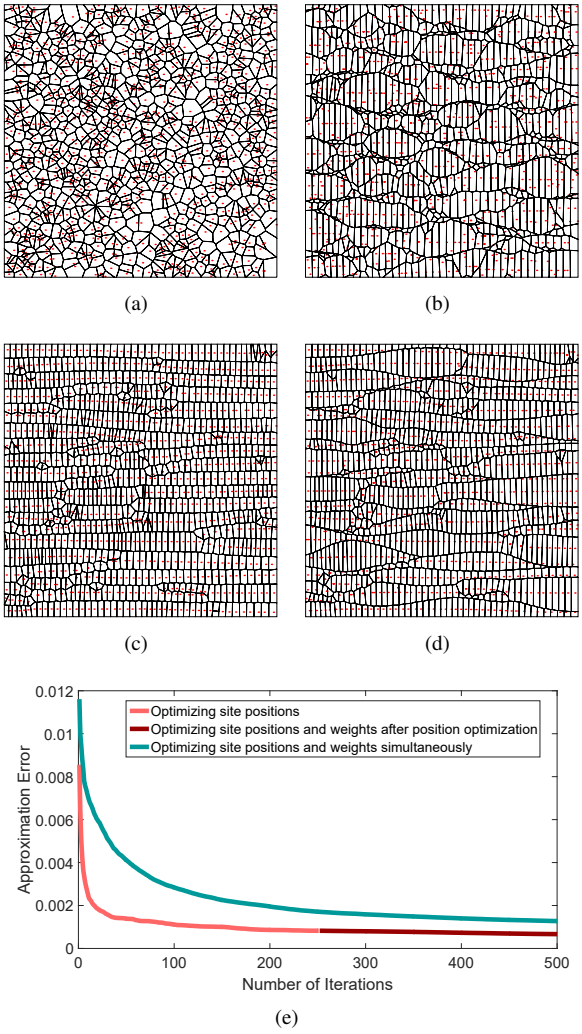


Figure 3: Influence of variables on the objective function, where $f(x, y) = 100x^2 + y^2$, $-1 \leq x, y \leq 1$. (a) Random initialization of 1,000 sites; (b) result obtained by directly optimizing all site positions and weights, with $\mathcal{E}_{OPD} = 1.273 \times 10^{-3}$; (c) result obtained by only optimizing site positions from the random initialization in (a), with $\mathcal{E}_{OPD} = 8.239 \times 10^{-4}$; (d) result obtained by optimizing all site positions and weights from the intermediate result in (c), with $\mathcal{E}_{OPD} = 6.607 \times 10^{-4}$; and (e) plot of approximation error versus iteration number, obtained by updating variables in different orders.

Step 1. Perturb the site positions or/and weights by using specific perturbation strategies. Denote the perturbed sites and weights by X' and W', respectively.

Step 2. Optimize the site positions X' or/and weights W' simultaneously using the L-BFGS method, upon which the approximation error $\mathcal{E}_{OPD}(X', W')$ can be computed.

Step 3. If $\mathcal{E}_{OPD}(X', W') < \mathcal{E}_{OPD}(X, W)$, we accept new site positions and weights, i.e., $X \leftarrow X'$ and $W \leftarrow W'$. Otherwise, if $\mathcal{E}_{OPD}(X', W') > \mathcal{E}_{OPD}(X, W)$ in successive three times of perturbation and optimization, stop.

Note that, site positions are more important in the optimization of the OPD energy function, as we have observed previously. We can have an expectation that optimizing site positions thoroughly before optimizing all variables together will significantly speed up the convergence. Hence, our optimization algorithm is applied twice: the first time we only optimize site positions (position optimization) and the second time we optimize all variables to achieve a deeper local minimum (position-weight optimization). Details of the algorithm will be given in the rest of this section.

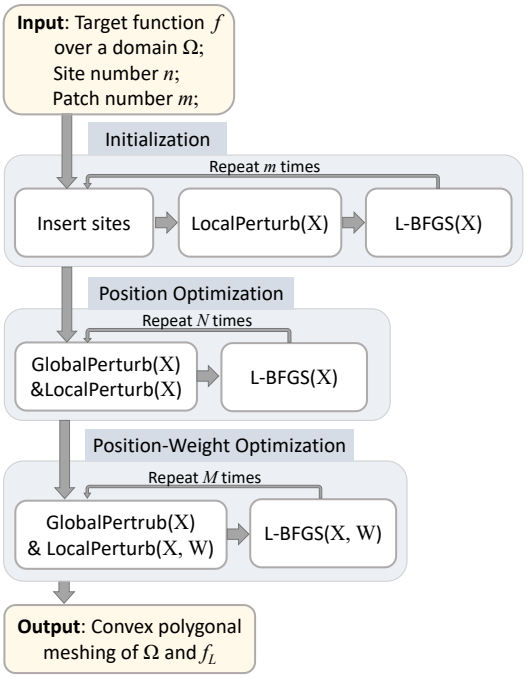


Figure 4: Overview of our OPD optimization algorithm.

5.2. Initialization

Providing a good initial guess is critical for optimization to find a deep local minimum. Here, we generate an initial guess that is close to an optimal solution, which will considerably speed up the iteration progress.

Our initialization algorithm holds all the weights constant and takes a target $f(\mathbf{x})$, the desired site number n and a batch number m as input. We start with a small number of sites, and then add more sites adaptively in each subsequent iteration, guided by approximation errors, so that the approximation quality is improved progressively. The initialization phase consists of the following steps:

Step 1. The first batch of n/m sites, randomly sampled for the given domain Ω, are optimized using the L-BFGS method. Then, the approximation error on each cell of the tessellation can be computed.

Step 2. The n/m cells with the largest approximation errors are selected, with one new site inserted into each of them. In particular, we compute the principal axes of each selected cell using principal component analysis (PCA), and a new site is added next to the current site along the direction of the short axis. This new site placement operation is called *site insertion* in the following sections.

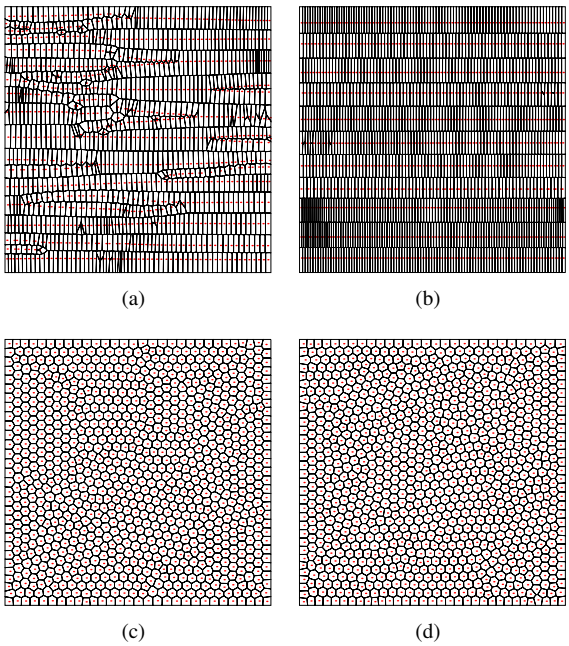


Figure 5: Comparison of new site insertion methods with different initializations. (a) and (c) are results using a random insertion method for new sites; and (b) and (d) are results using the PCA based new site insertion method. Top row: $f(x, y) = 100x^2 + y^2, -1 \le x, y \le 1$, with $\mathcal{E}_{OPD} = 4.124 \times 10^{-4}$ (a) and $\mathcal{E}_{OPD} = 8.492 \times 10^{-5}$ (b). Bottom row: $f(x, y) = x^2 + y^2, -1 \le x, y \le 1$, with $\mathcal{E}_{OPD} = 6.044 \times 10^{-7}$ (c) and $\mathcal{E}_{OPD} = 6.095 \times 10^{-7}$ (d).

Step 3. To escape from the current local minimum and find another possibly better local minimum, each site $\mathbf{x}_i$ is perturbed by a small amount in an arbitrary direction:

$$\mathbf{x}_i' = \mathbf{x}_i + \lambda l_i \mathbf{v}$$

where l_i is the length of the short axis of the corresponding cell V_i, λ is a factor used to adjust the magnitude of the perturbation, and $\mathbf{v}$ is a random vector within a unit 2D disk or 3D sphere. Hereinafter, this perturbation operation is referred to as *local perturbation*. Then, all the site positions are optimized by the L-BFGS method.

Step 4. If all batches of sites are added, stop; otherwise, update the approximation error on each cell and go back to Step 2.

Note that, as an initialization for a new round of local search in Step 3, the positions of newly added sites in Step 2 is of critical importance. Similar to previous observations, a random placement of newly added site at each selected cell would also lead to unsatisfying results. Examples are shown in Figure 5. By randomly placing a new site at each selected cell, our initialization algorithm results in shallow local minimizers, especially when the input target function shows strong anisotropic property, see Figure 5(a). Whereas, adding new sites at short axes gives a better anisotropic approximation to the input function, and leads to a much deeper local minimizer, see Figure 5(b). This site insertion method does no harm for the isotropic case, see Figures 5(d), which generates similar results to the random insertion method in Figure 5(c).

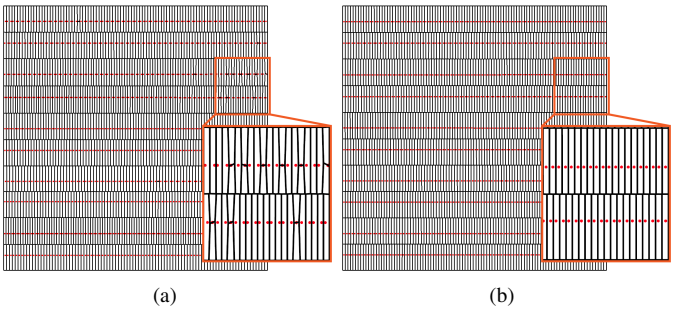


Figure 6: Influence of global perturbation on site position optimization. (a) Result using only local perturbation, $\mathcal{E}_{OPD} = 7.396 \times 10^{-5}$; and (b) result using both global and local perturbations, $\mathcal{E}_{OPD} = 7.115 \times 10^{-5}$.

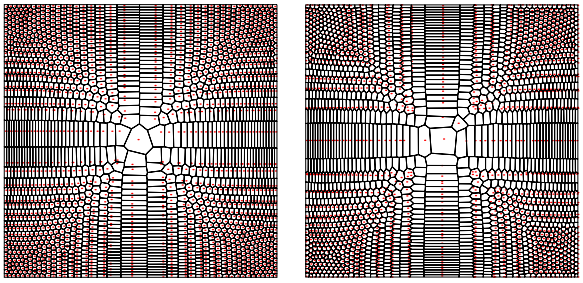
It is also worth pointing out that it is hard to determine the right number of batches such that the approximation can reach the best quality. Generally speaking, having fewer sites inserted in each step needs more iteration steps, leading to slower approximation and smaller approximation error. While having more newly-inserted sites in each approximation step needs fewer iteration steps, but may result in larger approximation error. The worst case is adding all sites at once, as shown in Figure 3(c), which results in a much shallower local minimum than the result obtained by adding new sites in batches in Figure 5(b). Empirically, we insert n sites by 20 batches, which reaches a good balance between computation time and approximation quality.

5.3. Perturbation

The results of our comprehensive initialization can be further improved by the aforementioned MCM global optimization which involves perturbation and local searching.

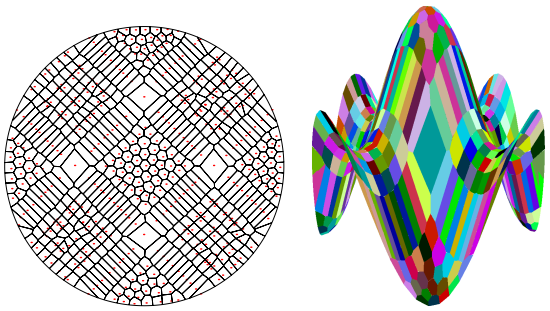
Position perturbation. The proposed perturbation strategy sequentially applies two types of perturbations: *global perturbation* and *local perturbation*. In global perturbation, 1% of the sites whose cells have the smallest approximation errors are removed. In compensation, each of 1% of the cells with the largest approximation errors is split by adding a new site, using the site insertion operation in Step 2 of Section 5.2. In local perturbation, each site is perturbed as Step 3 of Section 5.2. Note that, the factor λ plays an important role in adjusting the perturbation magnitude. Too small λ may lead to a failure of escaping from the current local minimizer. Whereas, too large λ would amount to re-starting the optimization from a random initialization. Empirically, λ is set to be 0.2 in all our experiments. As can be observed in Figure 6(b), our global optimization method helps find a deeper local minimum, benefiting from the perturbations. It is also worth pointing out that, it is the cooperative effect between global and local perturbations that allows our algorithm to reach deep local minima. As shown in Figure 6, the local minimum obtained by solely using local perturbation is shallower than the local minimum obtained by using both global and local perturbations in position optimization.

Weight perturbation. In the position-weight optimization, weights provide additional degrees of freedoms and allow a better approximation to the given function. After adding perturba-



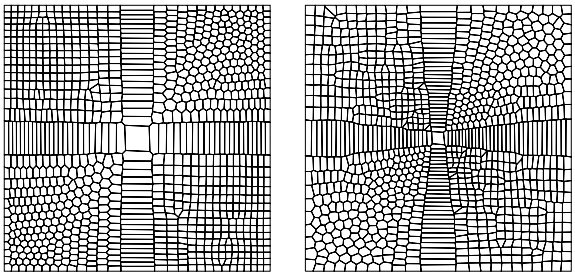
(a) (b)

Figure 7: $f(x, y) = x^4 + y^4 + (x-3)^2 + (y-3)^2, -10 \le x, y \le 10$ with 2,000 sites. (a) Result after position optimization; and (b) result of position-weight optimization, taking the tessellation in (a) as input.



(a) (b)

Figure 8: Resulting tessellation (a) and piecewise linear fit (b) of a non-convex target function $f(x, y) = \sin(\pi(x + 0.5)) \cos(\pi y), x^2 + y^2 \le 1$ with 500 sites.



(a) (b)

Figure 9: Density control. Resulting tessellations for a non-convex target function $f(x, y) = x^3 + y^3, -1 \le x, y \le 1$ with a constant density in (a) and a nonuniform density function $\rho(x, y) = 1.0/\left((x^2 + y^2)^2 + 0.001\right)$ in (b).

tion to sites, a local perturbation is also added to each weight. In particular, each weight w_i is randomly perturbed by a magnitude less than βl_i, where l_i is defined in Step 3 of Section 5.2 and β is an adjustable factor empirically chosen as $\beta = 0.1$. Results in Figure 7 indicate that the approximation result obtained from position optimization is further improved in the following position-weight optimization, as more degrees of freedom are exploited to control the local anisotropy.

5.4. Terminating condition

In our modified MCM method, a deep local minimum is found in a more aggressive way, i.e., a worse local minimum is always rejected, instead of accepting it according to a probability in the original MCM method. Our algorithm terminates when it fails to decrease the energy after successive three times of perturbation or reaches a pre-defined maximum iteration number. Benefiting from the delicate initialization, our method usually achieves satisfying results within 20 iterations of position optimization and 5 iterations of position-weight optimization. Thus, in all our experiments (except the experiment shown in Figure 3), the maximum iteration numbers of position optimization and position-weight optimization are set to 20 and 5, respectively.

6. Results

In this section, we present several experimental results to demonstrate the effectiveness of our OPD method and compare it with the OVT method. Voronoi cells restricted to a given domain are computed using an efficient clipping method proposed in [37]. All the experiments were performed on a machine with a 3.3 GHz Intel Xeon processor and 12 GB RAM.

Non-convex function approximation. We first conduct our algorithm on several non-convex functions in the 2D case to demonstrate the capability of our algorithm in generating anisotropic tessellations. Anisotropy described by the Hessian of the given non-convex functions may change greatly from region to region. The experimental results shown in Figures 1(c), 8, 9 and 12 indicate that the generated tessellations well capture the expected anisotropic variations.

Density control. By adding a non-negative scalar function to modulate the L^2 norm in the OPD energy function (2), our method is capable of generating anisotropic mesh with density adapting to the scalar function. In particular, we compute the minimizer of the modified energy function

$$\mathcal{E}_{OPD}(X, W) = \sum_{i=1}^{n} \int_{V_i} \rho(x)(f(\mathbf{x}) - P_i^*(\mathbf{x}))^2 \mathrm{d}\mathbf{x}, \qquad (5)$$

whose derivative can be derived in the same manner as Equation (2). Figure 9 shows that the generated meshes adapt to both the predicted anisotropy and density.

3D results. Our algorithm is naturally applicable for 3D mesh generation and provides the same control over anisotropy and density. Figure 10 shows an example of anisotropic meshing on a simple spherical domain, and the cutaway views show that the cells precisely get aligned to the expected anisotropy. In Figure 11, our method generates a sequence of meshes from isotropic mesh to anisotropic meshes (with aspect ratios being 1, 2 and 8, respectively) on a geometric model. The cutaway views show that the aspect ratio of cells are effectively controlled by the input function. In Figure 12, a cube model is decomposed to cells with spatially-varying anisotropy and density. Once again, the predicted anisotropy and density are well captured, as can be observed in exterior and cutaway views.

Comparison with the OVT method. We evaluate our OPD method via a comparison with the OVT method [1]. Three measures of the cells V_i, including the Hessian variation, shape ratio and modified area, are adopted here to evaluate the mesh quality as follows:

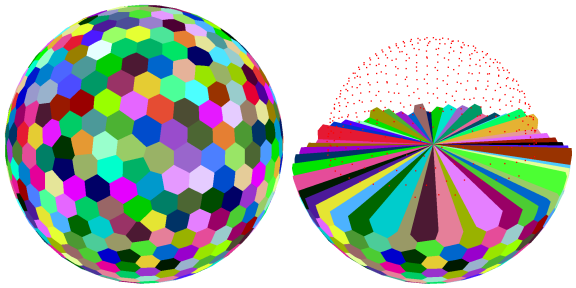


Figure 10: Tessellation of sphere for a non-smooth target function $f(x, y, z) = \sqrt{x^2 + y^2 + z^2}$ with 800 sites.

(a) (b) (c)

Figure 11: 3D optimal power diagrams with increasing anisotropy. (a) Isotropic; (b) 2:1:1; and (c) 8:1:1.

(a) Hessian variation: $\max_{\mathbf{x},\mathbf{y}\in V_i} \|Hess[f](\mathbf{x}) - Hess[f](\mathbf{y})\|_F$, where $Hess[f](\mathbf{x})$ is the Hessian of function $f(\mathbf{x})$ and $\|.\|_F$ is the Frobenius norm.

(b) Shape ratio: $\max_{\mathbf{x},\mathbf{y}\in V_i} \left[\sqrt{(\mathbf{x}-\mathbf{y})^t \overline{H}_{V_i} (\mathbf{x}-\mathbf{y})} \right] \left[|V_i| \sqrt{\det \overline{H}_{V_i}} \right]^{-\frac{1}{d}}$, where $\overline{H}_{V_i}$ is the averaged Hessian of f within a cell V_i.

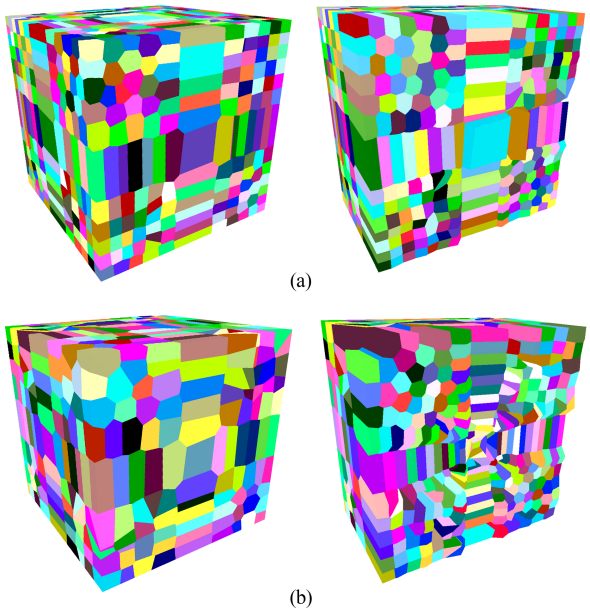
(c) Modified area: $\left(\overline{\rho}^d_{V_i} \det \overline{H}_{V_i}\right)^{\frac{1}{d+2}} |V_i|$, where $d = 2, 3$, and $\overline{\rho}_{V_i}$ is the average density within a cell V_i.

Generally speaking, the cells of a high-quality tessellation have similar Hessian variations, shape ratios and areas. Figures 13-15 show a comparison of our OPD method with the OVT method on generating meshes conforming to anisotropic metrics induced by 2D/3D convex functions. From the tessellation results, we can observe that our OPD method produces meshes with visually more regular cells. Histograms display the distributions of the three aforementioned measures on each resulting mesh, which indicate that our OPD method achieves a better performance than the OVT method. Histograms of Hessian variations in Figures 13 and 15 are not given as the Hessian is constant on each cell. The color-coded approximation errors in both L^1 and L^2 norms also reveal that our OPD method provides better approximation to the given functions.

7. Conclusion

We present a cell complex generation framework using OPD, achieving tessellations with well controlled local anisotropy and density. Our OPD method extends the existing OVT method by removing the convex restriction on the approximated function and replacing tangent planes with general planes as approximants. A combined local-global optimization technique, coupled with comprehensive initialization and perturbation schemes, is tailored for solving the specific optimization problem. Experimental results demonstrate that our method is capable of generating more diverse tessellations with expected density and anisotropy and providing a better approximation to the given function than the OVT method.

Limitation and future work. In practical applications, one may desire to generate meshes adapted to a given tensor field. However, an arbitrarily prescribed tensor field is not the Hessian of any function in general. Hence, the aforementioned anisotropic technique cannot be directly used for this situation. One promising solution is to find a function whose Hessian pro-



(a) (b)

Figure 12: Exterior and cutaway views of tessellations for a non-convex target function $f(x, y, z) = x^3 + y^3 + z^3, -1 \le x, y, z \le 1$ with a constant density in (a) and a nonuniform density function $\rho(x, y, z) = 1.0/\left((x^2 + y^2 + z^2)^2 + 0.001\right)$ in (b).

Table 1: Statistics of Running Time.

Figure	Site Number	Initialization (second)	Position Optimization (second)	Position-Weight Optimization (second)	Total Time (second)
1(a)	50	0.1	1.6	2.4	4.1
1(b)	200	49	65	72	186
1(c)	800	32	18.8	17.6	68.4
7	2,000	270	213	109	592
8	500	56	27.8	44.8	128.6
9	1,000	57	59.7	43.3	160
10	800	780	246	1,245	2,271
11	500	860	414	283	1,557
12(a)	2,000	1,731	3,615	615	5,961
12(b)	2,000	1,694	3,521	354	5,579
13	1,000	63	67	23	153
14	2,000	98	200	36	334
15	1,000	598	1,028	286	1,912

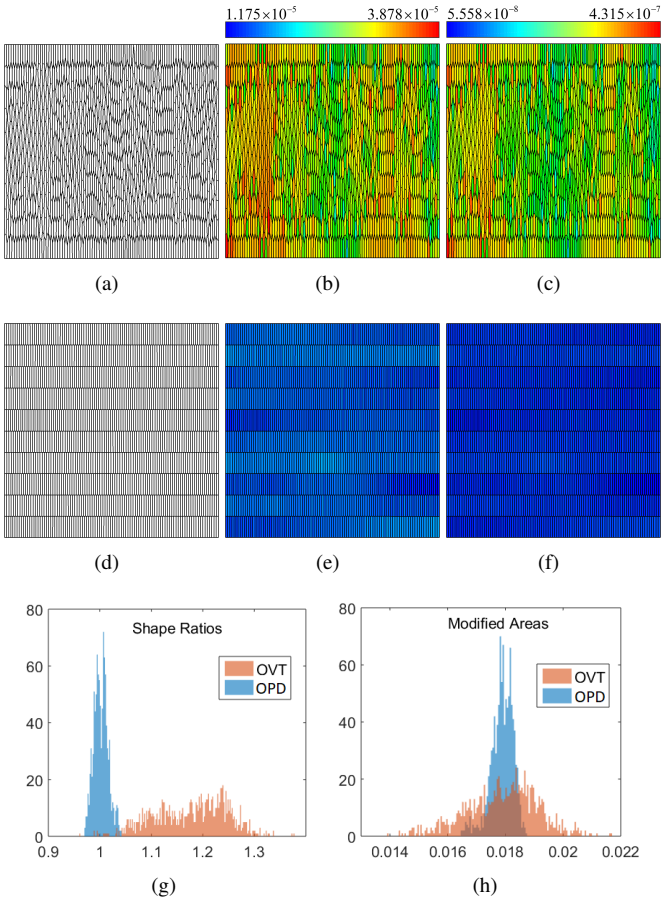


Figure 13: Comparison of our OPD method with the OVT method on a convex function with a constant Hessian $f(x, y) = 100x^2 + y^2, -1 \le x, y \le 1$. (a-c) Results from the OVT method; (d-f)results from our OPD method; and (g-h) histograms of shape ratios and modified areas. (a, d) Resulting tessellations; (b, e) color-coded errors in the L^1 norm; and (c, f) color-coded errors in the L^2 norm.

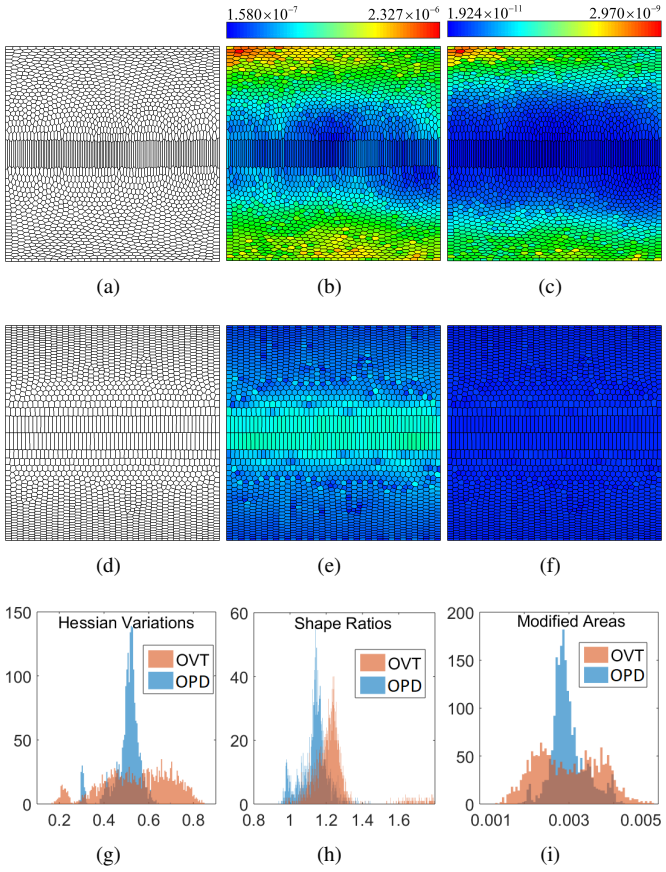


Figure 14: Comparison of our OPD method with the OVT method on a convex function with varying Hessian $f(x, y) = x^2 + 10^{-5}y^2 + y^4, -1 \le x, y \le 1$ with 2,000 sites. (a-c) Results from the OVT method; (d-f) results from our OPD method; and (g-h) histograms of Hessian variations, shape ratios and modified areas. (a, d) Resulting tessellations; (b, e) color-coded errors in the L^1 norm; and (c, f) color-coded errors in the L^2 norm.

vides the best fit to the prescribed anisotropy using a fitting method. Compared with the optimization method of computing OVT, our hybrid optimization often suffers from the high computational cost, despite of successes in finding optimal local minima. The running time of our OPD method for each example is given in Table 1. As part of our future work, we will focus on speeding up the optimization.

Although in this paper we only focus our research endeavors on generating anisotropic cell complexes for function approximation, potential applications of cell complex generation are much broader. For example, shape functions defined over polygonal/polyhedral elements are often desirable in finite element analysis and isogeometry analysis. We plan to further refine our algorithm by integrating more constraints on the geometry shape of cells and apply it to analysis and simulation tasks.

Acknowledgement

The research of Zhonggui Chen and Juan Cao was supported by the National Natural Science Foundation of China (No. 61472332, 61572020, 61728206), the Natural Science Foundation of Fujian Province of China (No. 2018J01104), and the grant of China Scholarship Council. The research of Cheng Wang was supported in part by the National Natural Science Foundation of China (No. U1605254). The research of Yongjie Jessica Zhang was supported in part by the PECASE Award N00014-16-1-2254 and NSF CAREER Award OCI-1149591.

References

[1] M. Budninskiy, B. Liu, F. De Goes, Y. Tong, P. Alliez, M. Desbrun, Optimal Voronoi tessellations with Hessian-based anisotropy, ACM Transactions on Graphics (TOG) 35 (6) (2016) 242:1–242:12.

[2] P. K. Agarwal, S. Suri, Surface approximation and geometric partitions, in: Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, 1994, pp. 24–33.

[3] F. Aurenhammer, A criterion for the affine equivalence of cell complexes in R^d and convex polyhedra in R^{d+1}, Discrete & Computational Geometry 2 (1) (1987) 49–64.

[4] J. R. Shewchuk, Delaunay refinement algorithms for triangular mesh generation, Computational Geometry 22 (1-3) (2002) 21–74.

[5] H. Borouchaki, P. L. George, F. Hecht, P. Laug, E. Saltel, Delaunay mesh generation governed by metric specifications. Part I. Algorithms, Finite Elements in Analysis and Design 25 (1-2) (1997) 61–83.

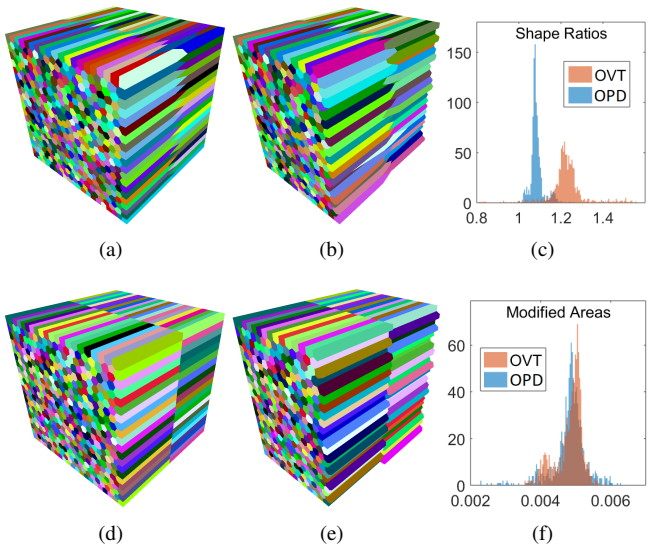


Figure 15: Comparison of our OPD method with the OVT method in the 3D case $f(x, y, z) = x^2 + y^2 + 0.01z^2, -1 \leq x, y \leq 1$. (a, b) Result and cutaway view from the OVT method; (d, e) result and cutaway view from our OPD method; and (c, f) histograms of shape ratios and modified areas, respectively

[6] C. Dobrzynski, P. Frey, Anisotropic Delaunay mesh adaptation for unsteady simulations, in: Proceedings of the 17th International Meshing Roundtable, 2008, pp. 177–194.

[7] J.-D. Boissonnat, C. Wormser, M. Yvinec, Anisotropic diagrams: Labelle Shewchuk approach revisited, Theoretical Computer Science 408 (2-3) (2008) 163–173.

[8] J.-D. Boissonnat, C. Wormser, M. Yvinec, Anisotropic Delaunay mesh generation, SIAM Journal on Computing 44 (2) (2015) 467–512.

[9] F. J. Bossen, P. S. Heckbert, A pliant method for anisotropic mesh generation, in: Proceedings of the 5th International. Meshing Roundtable, 1996, pp. 63–74.

[10] X. Jiao, A. Colombi, X. Ni, J. Hart, Anisotropic mesh adaptation for evolving triangulated surfaces, Engineering with Computers 26 (4) (2010) 363–376.

[11] Z. Zhong, X. Guo, W. Wang, B. Lévy, F. Sun, Y. Liu, W. Mao, et al., Particle-based anisotropic surface meshing., ACM Transactions on Graphics 32 (4) (2013) 99–1.

[12] F. Dassi, H. Si, S. Perotto, T. Streckenbach, Anisotropic finite element mesh adaptation via higher dimensional embedding, Procedia Engineering 124 (2015) 265–277.

[13] F. Dassi, S. Perotto, H. Si, T. Streckenbach, A priori anisotropic mesh adaptation driven by a higher dimensional embedding, Computer-Aided Design 85 (Supplement C) (2017) 111 – 122, 24th International Meshing Roundtable Special Issue: Advances in Mesh Generation.

[14] Q. Du, V. Faber, M. Gunzburger, Centroidal Voronoi tessellations: applications and algorithms, SIAM Review 41 (1999) 637–676.

[15] Q. Du, D. Wang, Anisotropic centroidal Voronoi tessellations and their applications, SIAM Journal on Scientific Computing 26 (3) (2005) 737–761.

[16] B. Lévy, Y. Liu, L_p centroidal Voronoi tessellation and its applications, ACM Transactions on Graphics 29 (4) (2010) 1–11.

[17] F. Sun, Y.-K. Choi, W. Wang, D.-M. Yan, Y. Liu, B. Lévy, Obtuse triangle suppression in anisotropic meshes, Computer Aided Geometric Design 28 (9) (2011) 537–548.

[18] R. Richter, M. Alexa, Mahalanobis centroidal Voronoi tessellations, Computers & Graphics 46 (Supplement C) (2015) 48 – 54, Shape Modeling International 2014.

[19] S. Valette, J. M. Chassery, R. Prost, Generic remeshing of 3D triangular meshes with metric-dependent discrete Voronoi diagrams, IEEE Transactions on Visualization and Computer Graphics 14 (2) (2008) 369–381.

[20] Y. Cai, X. Guo, Y. Liu, W. Wang, W. Mao, Z. Zhong, Surface approximation via asymptotic optimal geometric partition, IEEE Transactions on Visualization and Computer Graphics 23 (12) (2017) 2613–2626.

[21] B. Lévy, N. Bonneel, Variational anisotropic surface meshing with Voronoi parallel linear enumeration, in: Proceedings of the 21st International Meshing Roundtable, 2013, pp. 349–366.

[22] S.-Q. Xin, B. Lévy, Z. Chen, L. Chu, Y. Yu, C. Tu, W. Wang, Centroidal power diagrams with capacity constraints: computation, applications, and extension, ACM Transactions on Graphics (TOG) 35 (6) (2016) 244.

[23] E. Nadler, Piecewise linear best L^2 approximation on triangulations, Approximation Theory V (1986) 499–502.

[24] J.-M. Mirebeau, A. Cohen, Greedy bisection generates optimally adapted triangulations, Mathematics of Computation 81 (278) (2012) 811–837.

[25] L. Chen, J.-C. Xu, Optimal Delaunay triangulations, Journal of Computational Mathematics (2004) 299–308.

[26] L. Chen, Mesh smoothing schemes based on optimal Delaunay triangulations, in: Proceedings of 13th International Meshing Roundtable, 2004, pp. 109–120.

[27] X.-M. Fu, Y. Liu, J. Snyder, B. Guo, Anisotropic simplicial meshing using local convex functions, ACM Transactions on Graphics (TOG) 33 (6) (2014) 182.

[28] V. Nivoliers, B. Lévy, Approximating functions on a mesh with restricted Voronoï diagrams, Computer Graphics Forum 32 (5) (2013) 83–92.

[29] Z. Chen, Y. Xiao, J. Cao, Approximation by piecewise polynomials on Voronoi tessellation, Graphical Models 76 (5) (2014) 522–531.

[30] Z. Li, H. A. Scheraga, Monte Carlo-minimization approach to the multiple-minima problem in protein folding, Proceedings of the National Academy of Sciences of the United States of America 84 (19) (1987) 6611–6615.

[31] L. Lu, F. Sun, H. Pan, W. Wang, Global optimization of centroidal Voronoi tessellation with Monte Carlo approach, IEEE Transactions on Visualization and Computer Graphics 18 (11) (2012) 1880–1890.

[32] Z. Chen, J. Cao, W. Wang, Isotropic surface remeshing using constrained centroidal Delaunay mesh, Computer Graphics Forum 31 (7) (2012) 2077–2085.

[33] Z. Chen, W. Wang, B. Lévy, L. Liu, F. Sun, Revisiting optimal Delaunay triangulation for 3D graded mesh generation, SIAM Journal on Scientific Computing 36 (3) (2014) A930–A954.

[34] Y.-J. Liu, C.-X. Xu, R. Yi, D. Fan, Y. He, Manifold differential evolution (MDE): a global optimization method for geodesic centroidal Voronoi tessellations on meshes, ACM Transactions on Graphics (TOG) 35 (6) (2016) 243:1–243:10.

[35] F. Aurenhammer, Power diagrams: properties, algorithms and applications, SIAM Journal on Computing 16 (1) (1987) 78–96.

[36] H. Flanders, Differentiation under the integral sign, The American Mathematical Monthly 80 (6) (1973) 615–627.

[37] D.-M. Yan, W. Wang, B. Lévy, Y. Liu, Efficient computation of 3D clipped Voronoi diagram, in: International Conference on Geometric Modeling and Processing, Springer, 2010, pp. 269–282.