Capacity Constrained Blue-Noise Sampling on Surfaces

Sen Zhang[a,b], Jianwei Guo[c], Hui Zhang[b,*], Xiaohong Jia[d], Dong-Ming Yan[c,e,*], Jun-Hai Yong[b], Peter Wonka[e]

[a] *Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China*
[b] *School of Software, Tsinghua University, Beijing, 100084, China*
[c] *NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China*
[d] *KLMM, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, 100190, China*
[e] *Visual Computing Center, King Abdullah University of Science and Technology, Thuwal, 23955-6900, Saudi Arabia*

Abstract

We present a novel method for high-quality blue-noise sampling on mesh surfaces under capacity constraints. Unlike the previous surface sampling approach that only uses capacity constraints as a regularizer of the Centroidal Voronoi Tessellation (CVT) energy, our approach enforces an exact capacity constraint using the restricted power tessellation on surfaces. Our approach is a generalization of the previous 2D blue noise sampling technique using an interleaving optimization framework. We further extend this framework to handle multi-capacity constraints. We compare our approach with several state-of-the-art methods and demonstrate that our results are superior to previous work in terms of preserving the capacity constraints.

Keywords:
blue noise sampling, capacity constraints, centroidal Voronoi tessellation, power diagram

1. Introduction

Sampling is an essential technique in computer graphics, and it is a building block of various applications. One of the most important sampling techniques, generates so-called blue-noise patterns. The term "blue-noise" refers to any kind of noise with minimal low frequency components and no concentrated spikes in energy [1]. The quality of a blue noise sampling can be evaluated by two one-dimensional functions that are derived from the power spectrum analysis [2]. One is the *radially averaged power spectrum*, and the second one is *anisotropy*. From a geometric point of view, blue-noise sampling aims to generate uniformly randomly distributed point sets in a given domain.

Blue-noise sampling in the Euclidean domain has been extensively studied [3] over the years. More recently, many approaches focus on generating point sets on mesh surfaces with blue-noise properties. Such sampling has many applications in practice, e.g., rendering [4], solving some PDEs (e.g., water animation [5]), stippling [6], and object distribution [7].

The classical way of generating blue-noise point sets are Poisson-disk sampling and relaxation based methods, e.g., Lloyd iteration [8]. Although Poisson-disk sampling is fast and is able to generate point sets with good blue-noise properties, it cannot explicitly control the number of sampling points, which

Figure 1: Results of multi-capacity constrained sampling. An earthen dragon and a ceramic Bunny. Both use 3k samples.

is important for many applications. While Lloyd relaxation always result in more regular patterns which reduces the blue-noise characteristics. This iterative algorithm has to be terminated before reaching the local minima to avoid regular patterns [9].

Balzer et al. [10] proposed a variant of the Lloyd iteration, called capacity-constrained Voronoi tessellation (CCVT), where "capacity" means that the size of the cells of the power diagram of weighted points should have the same size. This algorithm introduces more irregularity patterns and improves the randomness of the point set as well. However, the CCVT method needs a descritization of the sampling domain and uses a discrete optimizer to compute the final solution which is inefficient. Chen et al. [7] proposed CapCVT, which combines Centroidal Voronoi Tessellation (CVT) and the capacity constrained Voronoi tessellation to improve the efficiency of the CCVT algorithm. However, the CapCVT is not able to en-

*Corresponding authors.
Email addresses: senzhang0815@163.com (Sen Zhang), jianwei.guo@nlpr.ia.ac.cn (Jianwei Guo), huizhang@tsinghua.edu.cn (Hui Zhang), xhjia@amss.ac.cn (Xiaohong Jia), yandongming@gmail.com (Dong-Ming Yan), yongjh@tsinghua.edu.cn (Jun-Hai Yong), pwonka@gmail.com (Peter Wonka)

force the exact capacity constraints. More recently, de Goes et al. [11] proposed a practical algorithm for blue noise sampling based on the theory proved by Aurenhammer et al. [12], which could enforce exact capacity constraints using an interleaving optimization framework that iteratively optimizes the point positions and their associated weights (more details are given in Sec. 3.2). Such equal capacity tessellations also have general interests in many research filed, such as computational geometry [13] and architectural geometry [14].

In this paper, we generalize the above mentioned interleaving optimization framework for blue-noise sampling [11] to 3D mesh surfaces. We formulate the new objective function on mesh surfaces, and provide rigorous mathematic proofs of the gradient derivation. We demonstrate that our results exhibit the best quality in terms of the capacity constraints among all the state-of-the-art blue noise sampling techniques. Figure 1 shows two examples of our multi-capacity constrained sampling on surfaces. The contributions of this paper include:

- A new approach for computing blue-noise sampling on mesh surfaces under capacity constraints.
- A novel extension to handle multi-capacity constraints.
- The derivation of the gradient of the new formulation on mesh surfaces.

2. Related Work

We briefly review the previous work on blue-noise sampling focusing on the approaches for surface sampling and their corresponding 2D approaches. For more details, please refer to recent survey papers [3, 15].

Surface Poisson-disk Sampling. Inspired by the technique of dart-throwing, Cline et al. [16] first propose to generate Poisson-disk samples on surfaces by utilizing a hierarchical data structure. Corsini et al. [17] present a new constrained Poisson-disk sampling method, which carefully selects samples from a dense point set pre-generated by Monte-Carlo sampling. The work of Bowers et al. [18] proposes a parallel dart throwing algorithm for sampling arbitrary surfaces. Geng et al. [19] generate approximate Poisson disk distributions directly on surfaces based on the tensor voting method. Ying et al. [20] propose another GPU-based approach by using the geodesic distance as metric. Then they further improve the maximal property of the Poisson disk sampling in a parallel manner [21]. Peyrot et al. [22] propose a feature sensitive dart-throwing method with more focus on the complex shapes and sharp features. Medeiros et al. [6] propose a hierarchical Poisson-disk sampling algorithm on polygonal models, which is used for surface stippling and non-photo realistic rendering. Yan and Wonka [23] propose a gap analysis framework to achieve *Maximal Poisson-disk Sampling* (MPS) on surfaces, and they also generalize MPS to adaptive sampling. Based on this, Guo et al. [24] use a subdivided mesh, instead of the common uniform 3D grid, to improve both the sampling quality and the efficiency.

Relaxation-based Sampling. Relaxation-based methods iteratively reposition the samples in a random point set, where the mostly used optimization technique is Lloyd relaxation [8]. Fu and Zhou [25] extend the 2D dart-throwing approach of [26] to surfaces sampling, and then the Lloyd relaxation is applied for high quality remeshing. Yan et al. [27] present an efficient algorithm to compute the CVT for isotropic surface sampling and remeshing. However, CVT tends to generate point distributions with regular patterns that lack some blue-noise properties. Xu et al. [28] generalize the concept of CCVT [10] to surfaces, which generates point sets exhibiting blue-noise properties. To improve the performance of CCVT, Chen et al. [7] combine CCVT with the CVT framework for blue-noise surface sampling. de Goes et al. [11] generate the blue-noise point sets using optimal transport. Apart from Lloyd-based methods, there are some other iterative approaches on surfaces. Chen et al. [4] introduce bilateral blue-noise sampling which integrates the non-spatial features/properties into the sample distance measures. Yan et al. [29] use the *Farthest Point Optimization* (FPO) [30] to generate point sets with high quality of blue-noise properties while avoiding regular structures.

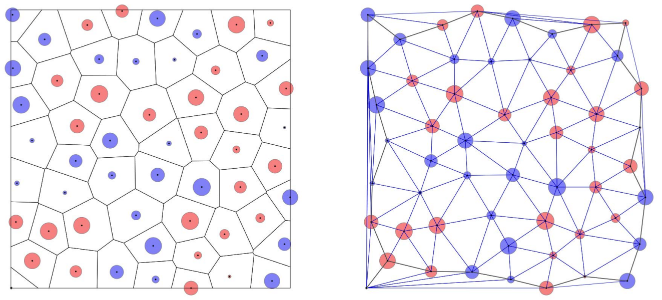

Figure 2: Illustration of the power diagram (left) and the regular triangulation (right) in 2D. The positive weights are shown in red and negative weights are shown in blue. The radius of each point $\mathbf{x}_i$ equals to $\sqrt{|w_i|}$.

3. Problem Statement

In this section, we first give the definitions of the power diagram and the restricted power diagram on surfaces, and the main theory that connects the power diagram and the capacity constraint. Then, we generalize the formulation of 2D capacity constrained blue-noise sampling to mesh surfaces. Finally, we propose a novel extension for multi-capacity constrained sampling.

3.1. Definitions

Power Diagram. A power diagram [31] tessellates the Euclidean space Ω into a set of convex polytopes (e.g., polygons in 2D, and polyhedra in 3D), by a set of n weighted points $\{\mathbf{x}_i, w_i\}$, where each $\mathbf{x}_i \in \mathbb{R}^n$, called *site*, is associated with a scalar value w_i called *weight* of site $\mathbf{x}_i$. Each polytope (or power cell) V_i of $\mathbf{x}_i$ contains the points that have smaller weighted distance to the site $\mathbf{x}_i$ than to others:

$$V_i = \{\mathbf{x} \in \Omega \mid d_w(\mathbf{x}_i, \mathbf{x}) < d_w(\mathbf{x}_j, \mathbf{x}), \forall j \neq i\}.$$

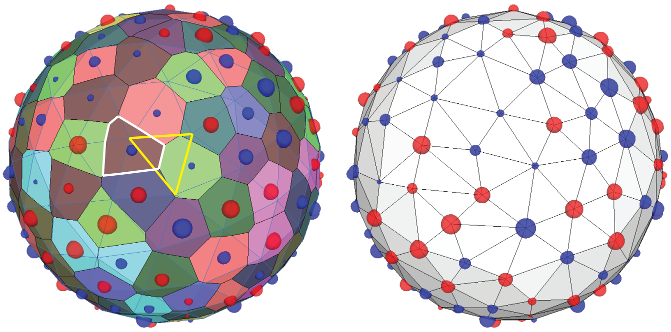

Figure 3: Illustration of the RPD and RRT on a sphere. The restricted power cells corresponding to each point is shown in random color. The boundary of RPC $V_{i|S}$ is marked with white color. A triangle in the input mesh (highlighted in yellow) is split into convex polygons and assigned to its incident cells.

To compute the weighted distance $d_w(\mathbf{x}_i, \mathbf{x})$, we adopt the power product $d_w(\mathbf{x}_i, \mathbf{x}) = \|\mathbf{x}_i - \mathbf{x}\|^2 - w_i$, here $\|\cdot\|$ denote the Euclidean norm.

Then the dual of the power diagram is called the regular triangulation. Figure 2 shows an example of the power diagram and regular triangulation in a 2D square. Note that when the weights of all the sites are the same, then the power diagram is equivalent to the Voronoi diagram.

Restricted Power Diagram. If the input domain is a 3D surface S, and the set of the weighted points are sampled on S, the intersection between the power diagram and the surface S is called the restricted power diagram (RPD), each intersected cell $V_{i|S}$ is called a restricted power cell on S, defined as

$$V_{i|S} = \{\mathbf{x} \in S \mid \Pi(\mathbf{x}_i, w_i; \mathbf{x}, 0) < \Pi(\mathbf{x}_j, w_j; \mathbf{x}, 0), \forall j \neq i\}.$$

The dual structure is called restricted regular triangulation (R-RT) on surfaces. Figure 3 illustrates the concept of RPD and RRT on a sphere.

Optimal Transport. The relation between the power diagram and the capacity constraint has been proven by Aurenhammer, Hoffman and Aranov [12]: Given a point set $\mathbf{X} = \{\mathbf{x}_i\}$ and a set of corresponding positive numbers $\{m_i\}$, and a probability measure μ such that $\sum m_i = \int d\mu$, it is possible to find the weights w_i of a power diagram such that $\mu(V_i) = m_i$ and the optimal weights are obtained as the maximum of a concave function.

Note that Aurenhammer, Hoffman and Aranov make the remark that the map defined by $\forall \mathbf{x} \in V_i, T(\mathbf{x}) = \mathbf{x}_i$ is an optimal transport map with respect to the L_2 cost. The equivalence can be also directly shown using Brenier's polar factorization theorem [32]. The proof of convergence and an implementation based on [12] is given by Mérigot [33]. A similar algorithm was proposed by Gu et al. [34] recently. This remark has been used in several works in optimal transport [11, 35, 36, 37, 38]. We refer the readers to the textbook [39] for more details on this topic.

3.2. Formulation on Surfaces

In our setting, the goal is to compute a point set $\mathbf{X} = \{\mathbf{x}_i\}$ on a give 3D surface that fulfills the capacity constraint, i.e., for each point $\mathbf{x}_i$, we want to constrain the (weighted) area of the restricted power cell associated with $\mathbf{x}_i$.

Our target is to minimize the following objective function subject to the equal capacity constraints on surfaces, i.e.,

$$\begin{aligned} \mathcal{E}(X, W) &= \sum_{i=1}^{n} \int_{V_{i|S}} \rho(\mathbf{x}) \|\mathbf{x} - \mathbf{x}_i\|^2 d\mathbf{x} \\ \text{s.t.} \quad m_i &= \int_{V_{i|S}} \rho(\mathbf{x}) d\sigma = m = \frac{m_\gamma}{n}, \end{aligned} \tag{1}$$

where $m_\gamma = \int_S \rho(\mathbf{x}) d\sigma$ is a given constant. This optimization problem is usually solved by introducing Lagrange multipliers $\Lambda = \{\lambda_i\}_{i=1}^n$, and the objective function becomes

$$\text{Minimize } \ \mathcal{E}(X, W) + \sum_{i=1}^{n} \lambda_i (m_i - m) \tag{2}$$

with respect to $\mathbf{x}_i, w_i, \lambda_i$. However, since an additional n variables λ_i add complexity to the optimization problem, it can be reformulated into a simple scalar function [11]:

$$\mathcal{F}(X, W) = \mathcal{E}(X, W) - \sum_{i=1}^{n} w_i (m_i - m), \tag{3}$$

with respect to $\mathbf{x}_i, w_i$. By our appendix and [11], the optimization of (2) is equivalent to finding a stationary point of (3).

Note that the difference between our formulation and [11] is that we use the restricted power diagram on surfaces instead of the ordinary power diagram. We derive the gradient on surfaces for variables X and W. Surprisingly, we found that the gradients have the similar forms as their Euclidean formulation. The gradients of the energy $\mathcal{F}(X, W)$ are

$$\begin{aligned} \nabla_{w_i} \mathcal{F}(X, W) &= m - m_i, \\ \nabla_{\mathbf{x}_i} \mathcal{F}(X, W) &= 2m_i(\mathbf{x}_i - \mathbf{b}_i). \end{aligned}$$

where $\mathbf{b}_i = \frac{1}{m_i} \int_{V_{i|S}} \mathbf{x} \rho(\mathbf{x}) d\mathbf{x}$ is the corresponding weighted barycenter. However, the derivation on surfaces is more involved. Similar to [11], the objective function $\mathcal{F}$ is a concave maximization problem when $\mathbf{X}$ is fixed, and it can be considered as a minimization problem of the centroidal power diagram when W is fixed. The formal proof and derivations are given in Appendix B. Note that an alternative elegant proof was independently derived by Bruno Lévy in a recent paper [38].

3.3. Multi-Capacity Extension

The formulation discussed above considers only a single capacity value. In this paper, we further extend the sampling problem to multiple capacity constraints. Given a ratio θ_i for $\mathbf{x}_i$, the customized capacity can be given as $m_i^c = \theta_i m$. In order to keep the total capacity requirement, we require $\sum_{i=1}^n m_i^c = m_\gamma$. Thus the new energy can be written as

$$\mathcal{F}^c(X, W) = \mathcal{E}(X, W) - \sum_{i=1}^{n} w_i (m_i - m_i^c).$$

The gradient w.r.t. w_i is changed to be

$$\nabla_{w_i} \mathcal{F}^c(X, W) = m_i^c - m_i,$$

and the gradient $\nabla_{\mathbf{x}_i} \mathcal{F}^c(X, W)$ remains unchanged.

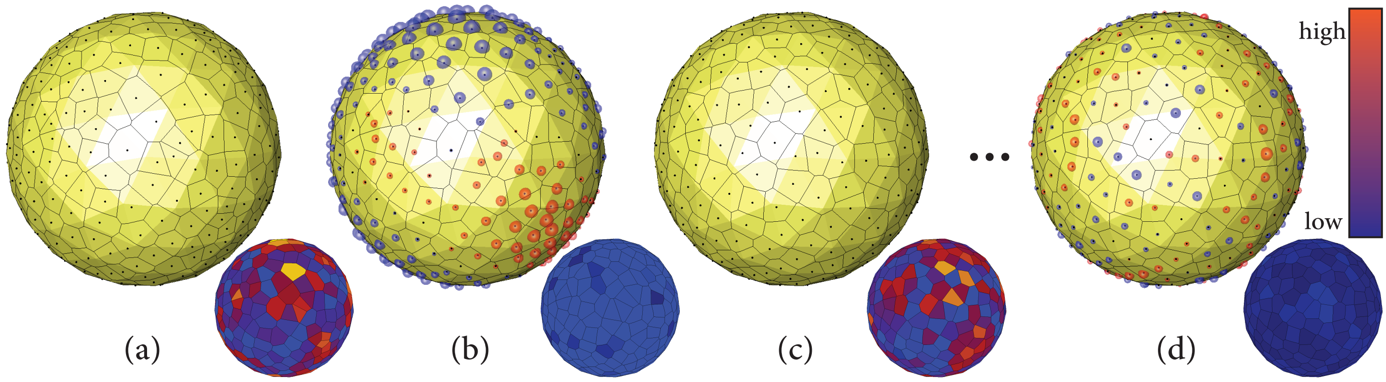

Figure 4: The main steps of our algorithm. The top row shows the Restricted power diagram of each step and the bottom row shows the corresponding quadratic errors respect to the prescribed capacities $\|m_i - m\|^2$. The colder color means small error and the warmer color means high error. (a) Initial sampling after 3 steps of Lloyd iteration (for better visualization), (b) after weight optimization, (c) after vertex optimization, and (d) final result.

4. Implementation Details

The input of our algorithm is a triangular mesh surface S, and the number of desired sampling points n. A density function $\rho(\mathbf{x})$ is defined on mesh vertices and piecewise linearly interpolated over the triangles. In our implementation, we use the local feature size introduced in [40] as the density function, i.e., $lfs^2(\mathbf{x})$. But other density can also be used. There are three main steps in our framework, i.e., initialization and interleaving weight/vertex optimization. Figure 4 shows the main steps of our pipeline.

4.1. Initial Sampling

The sampling points X are initialized randomly according to the density function. The initial power weights W are initialized to be 0. Before starting into optimization, we perform 3 $\sim$ 5 steps of Lloyd iteration to get a better initial distribution. Otherwise, the optimization might get stuck in undesirable local minima quickly and it becomes difficult to find optimal weights. In the case of multi-capacity sampling, we initialize each type of capacity separately to ensure a better distribution. Figure 4(a) shows the initialization result on a sphere model.

4.2. Weight Optimization

Before starting the weight optimization, all weights are reset to 0. Weight optimization makes every sampling point share a common capacity as much as possible when the positions of sampling points remain fixed. The Hessian matrix w.r.t. weight $H_{\mathcal{F}} = \nabla_w^2 \mathcal{F}(X, W)$ can be explicitly derived as (see Theorem 6 in Appendix):

$$[H_{\mathcal{F}}]_{ij} = \frac{\bar{\rho}_{ij}}{2} \sum_{l \in \mathcal{T}_{ij}} \frac{|e^*_{ij} \cap \tau_l|}{|e_{ij}|_{\tau_l}},$$

$$[H_{\mathcal{F}}]_{ii} = \sum_{j \in \Omega_i} [H_{\mathcal{F}}]_{ij},$$

where $|e_{ij}|_\tau$ is the length of projection of e_{ij} onto the triangular plane τ, $\mathcal{T}_{ij}$ is the index set of the triangles in the mesh that intersect with the bisecting plane e^*_{ij}, and $\bar{\rho}_{ij}$ is the average value of ρ over $e^*_{ij} \cap \mathcal{T}$. Newton iterations are used to optimize weights. Note that the Hessian on surfaces is different from the 2D case, the edges of the restricted power diagram is not a single segment but a set of connected segments.

The derivation of the multi-capacity sampling is similar. The only difference is that the righthand side of the linear system is changed to be $\nabla_{w_i} \mathcal{F}^c(X, W)$ instead of $\nabla_{w_i} \mathcal{F}(X, W)$.

During the iterations, the step size is adapted by a line search with Armijo condition [41]. The weight optimization stops when the threshold is met. The threshold for weight optimization is defined as $\sqrt{\sum_{i=1}^n (\nabla_{w_i} \mathcal{F}(X, W))^2} \leq \frac{\alpha_1}{n} m_\gamma^{\theta_1}$, where α_1 is a scaling coefficient accounting for the number of sampling points and the density function ($\alpha_1 = 0.1$, $\theta_1 = 1.0$ in our experiments). Typically, 5 $\sim$ 7 iterations can reduce the δ'_w within the threshold.

4.3. Vertex Optimization

Vertex optimization, which reduces the objective function $\mathcal{F}$ when the weight remains unchanged, can be seen as the process of finding a "centroidal power diagram" of the weighed sampling points, which could be achieved by using either Lloyd iteration [8] or quasi-Newton solvers [42].

During the optimization, the positions of the sampling points will be updated to their weighted barycenters, and then projecting $\mathbf{b}_i$ to the input mesh S if Lloyd iteration is used. Otherwise, if a quasi-Newton solver is used, the gradient $\nabla_{\mathbf{x}_i} \mathcal{F}(X, W)$ should be constrained within the tangent plane of $\mathbf{x}_i$, i.e.,

$$\begin{aligned} \nabla_{\mathbf{x}_i|S} \mathcal{F}(X, W) \quad &= \nabla_{\mathbf{x}_i} \mathcal{F}(X, W) \\ &- [\nabla_{\mathbf{x}_i} \mathcal{F}(X, W) \cdot \mathbf{N}(\mathbf{x}_i)] \mathbf{N}(\mathbf{x}_i). \end{aligned}$$

After each step of update, the vertices are then projected back to the input surface. Optimizing vertices only reduces the energy $\mathcal{F}(X, W)$, but might increase of capacity variance (see Figure 6 in Section 5). Typically after 3 $\sim$ 5 iterations, the requirement of the threshold will be satisfied. We set the condition for vertex optimization to $\sqrt{\sum_{i=1}^n \|\nabla_{\mathbf{x}_i} \mathcal{F}(X, W)\|^2} \leq \frac{\alpha_2}{n} m_\gamma^{\theta_2}$ ($\alpha_2 = 0.1$, $\theta_2 = 1.2$ in our experiments).

4.4. Randomness Improvement

Since our optimization framework has the same shortcoming as most relaxation based methods, i.e., the restricted power cells form a regular hexagonal pattern after optimization. To overcome this problem, Gaussian noise is used to add randomness in such regions to break regular patterns.

It is worth to point out that the local regular patterns of the point distributions are detected and are broken up in a way that is similar to [11]: we first measure the regularity for every point, and then disturb the point and its one-ring neighbors in the regular regions. The main difference of our implementation is that the disturbances occur in the corresponding containing triangles on the surface instead of resampling randomly. Our procedure ensures that the perturbed points still lie on the mesh.

Algorithm 1: Optimization algorithm

Initialize sampling point set $\mathbf{X}$ with n points;
Run 3 $\sim$ 5 times Lloyd iterations;
Compute the threshold for weight optimization $\delta_w = \frac{\alpha_1}{n} m_\gamma^{\theta_1}$;
Compute the threshold for vertex optimization $\delta_{\mathbf{x}} = \frac{\alpha_2}{n} m_\gamma^{\theta_2}$;
repeat
    Set all power weights to be 0;
    Call WEIGHT-OPTIMIZATION;
    Optimize vertices and update RVD;
    Compute $\delta'_{\mathbf{x}} = \sqrt{\sum_{i=1}^{n} \|\nabla_{\mathbf{x}_i} \mathcal{F}(X, W)\|^2}$;
until ($\delta'_{\mathbf{x}} \leq \delta_{\mathbf{x}}$);
Call WEIGHT-OPTIMIZATION;
Randomness improvement;
Function WEIGHT-OPTIMIZATION
repeat
    Solve the concave problem of weight optimization;
    Update power weights and RVD;
    Compute $\delta'_w = \sqrt{\sum_{i=1}^{n} (\nabla_{w_i} \mathcal{F}(X, W))^2}$;
until ($\delta'_w \leq \delta_w$);

5. Experimental Results

In this section, we demonstrate some results of the proposed method and compare our approach with several state-of-the-art surface sampling algorithms in various aspects. In our implementation, we use CGAL [43] for computing the 3D regular triangulation. We use the implementation of [27] for RPD computation. Note that more recently, Bruno Lévy has released a new open-source package, called *Geogram* [44], which contains an improved version of the RVD computation libraray. Our experiments are conducted on a PC with i5-2320, 3.00GHz CPU, 16GB memory and a 64-bit Ubuntu operating system.

Performance Analysis. Our framework is able to generate a high quality blue-noise point set efficiently. We test our method on a complicated Pegaso model as shown in Figure 5. The convergence behavior of the optimization procedure run on the Pegaso model is shown in Figure 6. In our implementation, we set the number of iterations of weight optimization and vertex optimization to 10 and 20 times, respectively. The optimization usually converges after 3-5 iterations. The total running times are 89.2 and 182.5 seconds for uniform and adaptive sampling, respectively. More results are shown in Fig. 7.

Figure 5: Uniform (top) and adaptive (bottom) sampling on the Pegaso model. The number of sampling points is 10K in both tests. Left: sampled points, middle: quadratic error with respect to the prescribed capacities, and right: restricted power diagram. Different colors indicate different valences of each vertex in the dual restricted regular triangulation. Light green is valence 6 (v_6), orange is v_7, blue is v_5, dark blue is v_4 and brown is v_7.

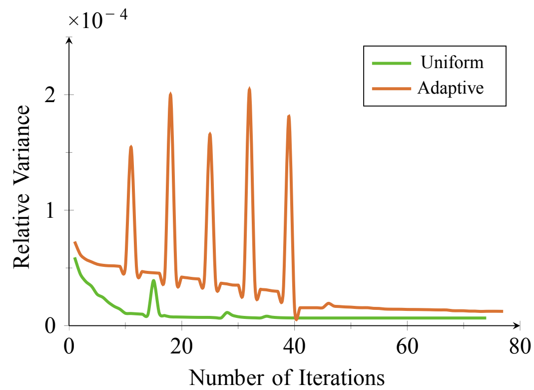

Figure 6: Illustration of the convergence of the capacity variance against the number of iterations. Each peak corresponds to a switch from the weight optimization to vertex position optimization.

Figure 8 compares the timing statistics of different approaches. The time cost of CVT and CapCVT are evaluated by apply-

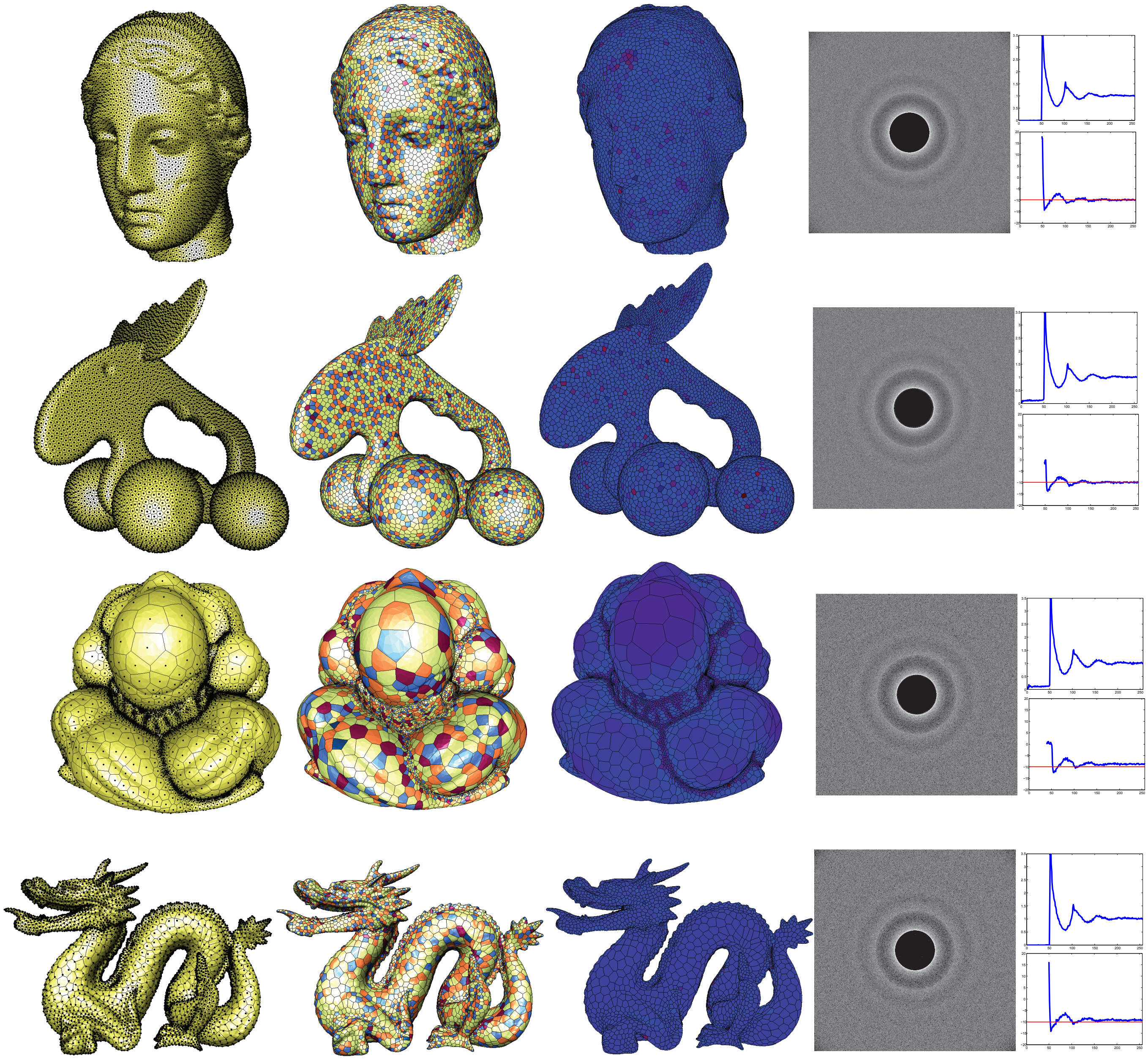

Figure 7: More sampling results. From top to bottom: uniform sampling of Venus and Elk, and adaptive sampling of Omotondo and Dragon. We use 10K samples for all the models. The time costs are 92.34s, 94.07s, 123.23s, and 125.45s, respectively. From left to right: sampled points and their corresponding RPDs; color-coded RPDs, where the color indicates different valences of each vertex in the dual restricted regular triangulation; quadratic error with respect to the prescribed capacities; and the power spectrum, the radial power and the normal anisotropy.

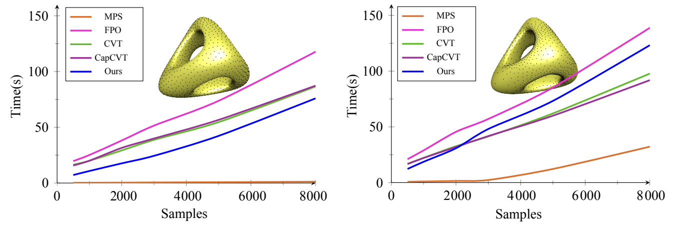

Figure 8: Comparison of the time cost of different methods using the Genus3 model. Left: uniform sampling. Right: adaptive sampling.

ing 100 L-BFGS iterations. Since MPS does not need iterative optimization, it is the most efficient approach compared to the other methods, while FPO is the most time consuming since it optimizes each individual point once during each step of iteration. From this comparison, we can see that the performance of our method is comparative to the other optimization-based approaches, while we can generate results with minimum capacity variances.

Randomness Improvement. We further analyze the effect of the Gaussian noise introduced in Sec. 4.4 for randomness improvement. We show two examples in Fig. 9 and Fig. 10 for both uniform and adaptive sampling, respectively. In each example, we first run our interleaving optimization framework until convergence. As we can see in the left column, both results contains many hexagonal cells. Then we apply Gaussian noise to break the regular patterns and run the optimization again. The right column in each Figure shows the final results with more irregular patterns while keeping small capacity variances. In the first example, the percentage of valence-6 points is reduced from 80.55% to 54.95% after adding Gaussian noise. In the second example, the percentage of valence-6 points is reduced from 75.51% to 50.53% after adding Gaussian noise.

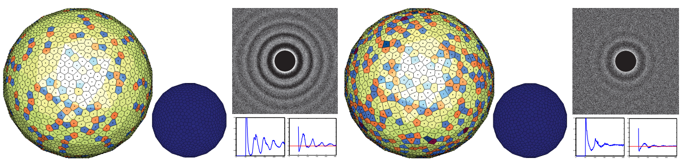

Figure 9: Randomness improvement of the uniform sampling on the Sphere model. Left: results without adding Gaussian noise; right: results of adding Gaussian noise and further optimization.

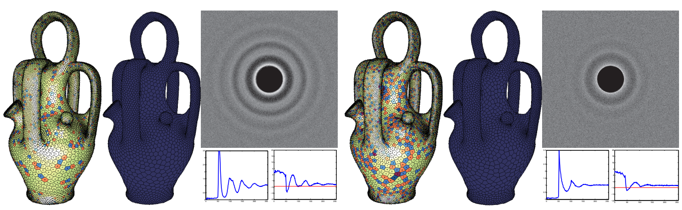

Figure 10: Randomness improvement of the adaptive sampling on the Botijo model. Left: results without adding Gaussian noise; right: results of adding Gaussian noise and further optimization.

Evaluation and Comparison. We then evaluate our results in terms of sampling irregularity, quadratic error with respect to the prescribed capacities and the spectral property. The last column of Figure 11 and Figure 12 demonstrate the visual qualities of these criteria of uniform sampling and adaptive sampling, respectively. It is easy to see that our results present high irregularity and low capacity variation, as well as good blue-noise property.

Next, we compare the above criteria with several state-of-the-art techniques in Figure 11 and Figure 12, including maximal Poisson-disk sampling (MPS) [23], farthest point optimization (FPO) [29], centoridal Voronoi tessellation (CVT) [27] and capacity-constrained centroidal Voronoi tessellation (CapCVT) [7]. To make a precise comparison, we use the same density function $\rho(\mathbf{x}) = 1/lfs^2(\mathbf{x})$ for all methods. The results of CVT and CapCVT are generated after 100 LBFGS iterations. The balance coefficient λ used in CapCVT is set to 50 to enforce better capacity constraints. Usually MPS has the maximal variance, and FPO and CVT also have large values since these methods do not have explicit control of the capacity constraints. CapCVT is better since it tends to equalize the capacity values using a penalty term in addition to CVT energy, which controls the regularity of the point distribution. Our result exhibits the lowest capacity variance among all the methods thanks to the exact capacity formulation.

Figure 13 compares the capacity variances against the increasing number of points for all approaches. The relative capacity variance is computed as $\frac{1}{m_\gamma}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(m_i - m)^2}$. We use the logarithmic coordinates for better visualization. From this figure, we can see that capacity variances converge when increasing the number of sampling points for all sampling methods. The magnitude of our method is several orders smaller than other approaches.

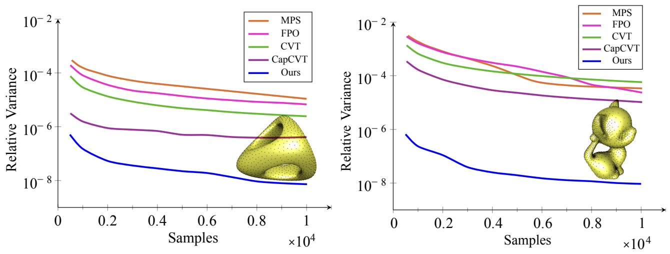

Figure 13: Comparison of the capacity variance against the increasing number of sample points. Left: uniform sampling. Right: adaptive sampling.

Feature Preserving. Our framework is able to handle sharp features easily. We assume that the sharp features are given as input. During the optimization, the points whose restricted power cells are clipped with feature curves are project back to the feature skeletons. Figure 14 shows an example of feature preserving sampling and its spectral analysis. This simple extension does not spoil the blue-noise property.

Multi-Capacity Constraints. Two examples of multi-capacity constraints are shown in Figure 1. Figure 14 shows the quadratic error with respect to the prescribed capacities and the spectral analysis results of a two-capacity example on a sphere model. This new extension keeps the variances small and maintains high blue-noise quality.

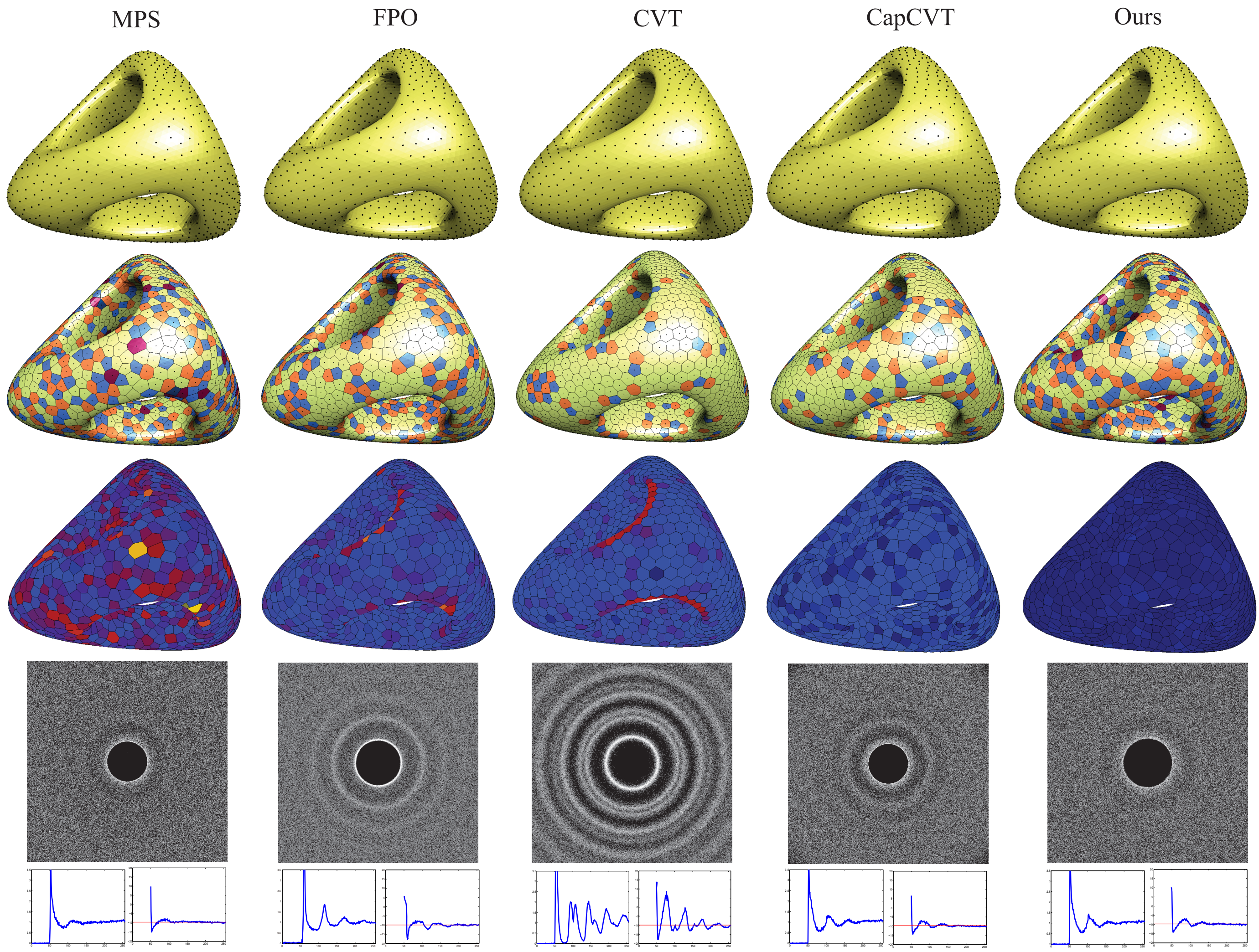

Figure 11: Comparison of the uniform sampling results. From left to right: results of MPS, FPO, CVT, CapCVT and ours. The top row shows the sampling results of each method. The second row shows the restricted Power diagram of the sampling points. The third row shows quadratic errors with respect to the prescribed capacities. The colors from blue to red indicate the errors from low to high. The fourth row is the power spectrum of the differential domain analysis [45] and the last row shows the radial power and the normal anisotropy of each method.

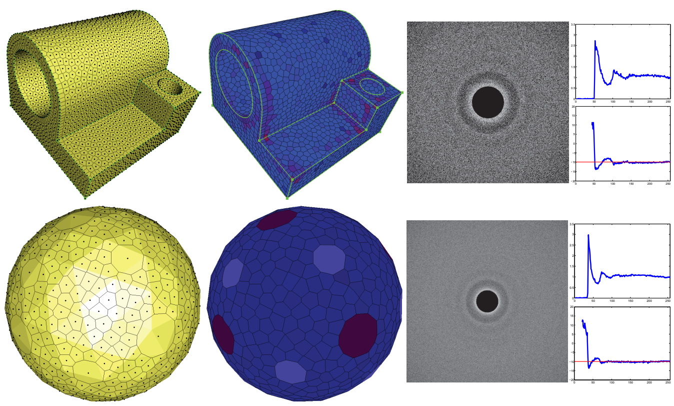

Figure 14: Spectral analysis of examples of feature preserving (top) and multi-capacity sampling (bottom). The feature curves of the joint model are shown in green. Left: results of RPDs; middle: quadratic error with respect to the prescribed capacities; and right results of spectral analysis.

Limitations. One limitation of our algorithm is that we cannot guarantee the maximal sampling property as [23]. Gaps can be detected if we draw a sphere at each vertex using the shortest edge length as radius in uniform sampling case and using the shortest incident edge length as radius in adaptive sampling case. Although our algorithm works well in practice, the connection between the capacity constraint and the blue-noise property is still not well explained. We would like to address these issues as future works.

6. Conclusions

We present a new method for blue noise sampling on mesh surfaces under exact capacity constraints. The problem is formulated as an optimization problem on mesh surfaces. A closed-form formula for gradient computation on surfaces has been derived and it has been proved that the gradient of the new formulation coincide with its Euclidean counterpart, thus can be

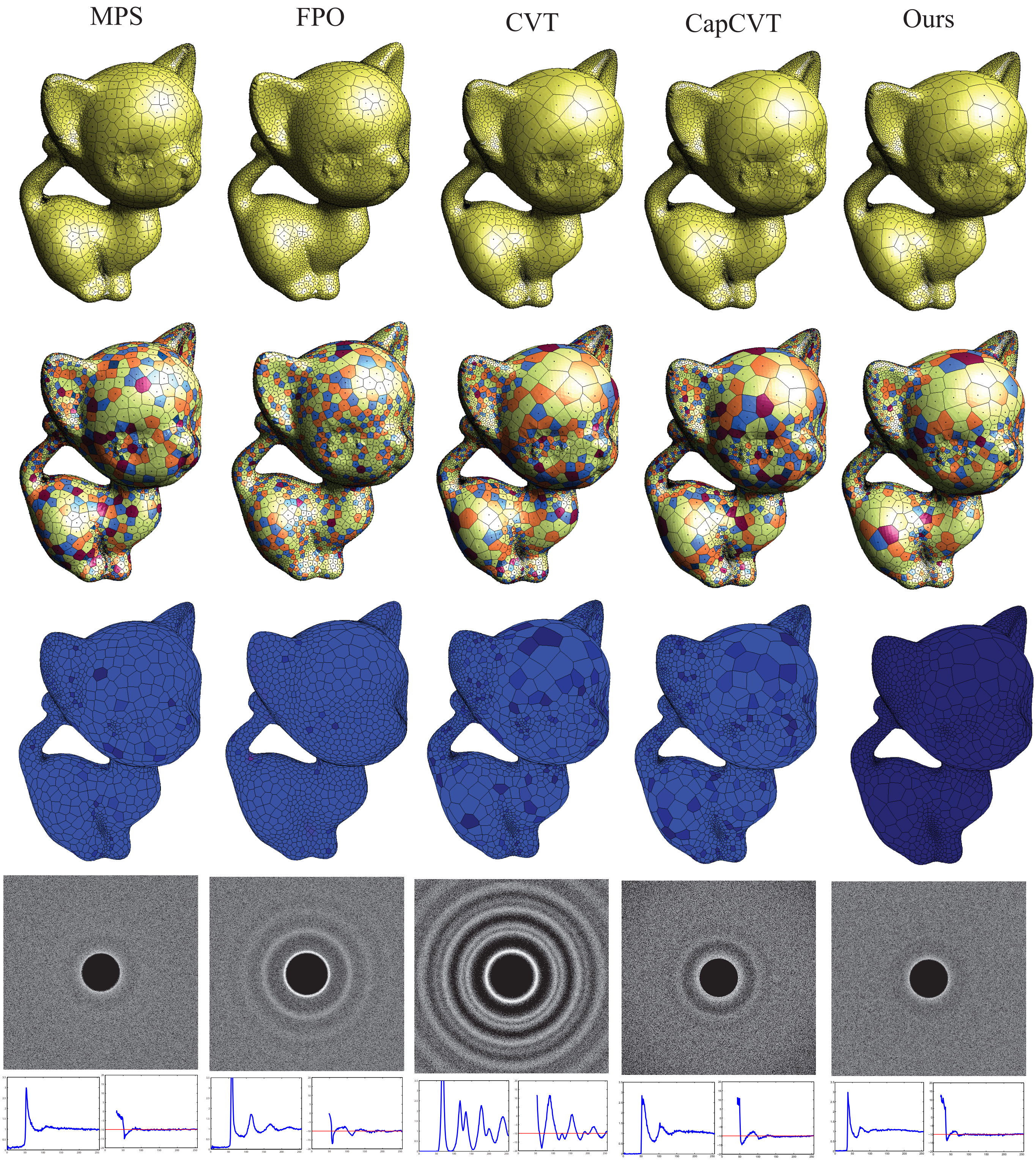

Figure 12: Comparison of the adaptive sampling results.

minimized efficiently using modern solvers. We also extend the presented sampling framework to handle multi-capacity constraints. We make a complete comparison of various criteria between the state-of-the-art surface sampling approaches, and we show that our results perform better than others when preserving capacity constraints. In the future, we would like to investigate more properties of this sampling framework, and apply it for more applications, such as remeshing.

Acknowledgements

We would like to thank anonymous reviewers for their insightful comments and suggestions which greatly improve the quality of the paper. We also thank Zhonggui Chen for providing the executable of CapCVT, and Bruno Lévy for sharing the *G*eex open-source platform with us. This work was partially supported by the National Key Technologies R&D Program of China (2015BAF23B03), the National Nature Science Foundation of China (61373070, 61372168, 11201463, and 61572502), Tsinghua University Initiative Scientific Research Program (2012Z02170), the Scientific Research Foundation for the Returned Overseas Chinese Scholars of State Education Ministry of China, and the Open Funding Project of the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (BUAA-VR-15KF-06).

References

[1] Wiki . https://en.wikipedia.org/wiki/colors_of_noise#blue_noise. 2015.

[2] Ulichney R. Digital Halftoning. MIT Press; 1987.

[3] Lagae A, Dutré P. A comparison of methods for generating Poisson disk distributions. Computer Graphics Forum 2008;27(1):114–29.

[4] Chen J, Ge X, Wei LY, Wang B, Wang Y, Wang H, et al. Bilateral blue noise sampling. ACM Trans on Graphics (Proc SIGGRAPH Asia) 2013;32(6):216:1–216:11.

[5] Schechter H, Bridson R. Ghost SPH for animating water. ACM Trans on Graphics (Proc SIGGRAPH) 2012;31(4):1–8.

[6] Medeiros E, Ingrid L, Pesco S, Silva C. Fast adaptive blue noise on polygonal surfaces. Graphical Models 2014;76(1):17–29.

[7] Chen Z, Yuan Z, Choi YK, Liu L, Wang W. Variational blue noise sampling. IEEE Trans on Vis and Comp Graphics 2012;18(10):1784–96.

[8] Lloyd SA. Least squares quantization in PCM. IEEE Transactions on Information Theory 1982;28(2):129–37.

[9] Mitchell DP. Spectrally optimal sampling for distribution ray tracing. In: Proc. ACM SIGGRAPH. 1991, p. 157–64.

[10] Balzer M, Schlömer T, Deussen O. Capacity-constrained point distributions: A variant of Lloyd's method. ACM Trans on Graphics (Proc SIGGRAPH) 2009;28:1–8.

[11] de Goes F, Breeden K, Ostromoukhov V, Desbrun M. Blue noise through optimal transport. ACM Trans on Graphics (Proc SIGGRAPH Asia) 2012;31:171:1–171:12.

[12] Aurenhammer F, Hoffmann F, Aronov B. Minkowski-type theorems and least-squares partitioning. In: Symposium on Computational Geometry. 1992, p. 350–7.

[13] Hales TC. The honeycomb conjecture. Discrete & Computational Geometry 2001;25(1):1–22.

[14] Pottmann H, Asperl A, Hofer M, Kilian: A. Architectural Geometry. Bentley Institute Press; 2007.

[15] Yan DM, Guo J, Wang B, Zhang X, Wonka P. A survey of blue-noise sampling and its applications. Journal of Computer Science and Technology 2015;30(3):439–52.

[16] Cline D, Jeschke S, Razdan A, White K, Wonka P. Dart throwing on surfaces. Computer Graphics Forum (Proc EGSR) 2009;28(4):1217–26.

[17] Corsini M, Cignoni P, Scopigno R. Efficient and flexible sampling with blue noise properties of triangular meshes. IEEE Trans on Vis and Comp Graphics 2012;18(6):914–24.

[18] Bowers J, Wang R, Wei LY, Maletz D. Parallel Poisson disk sampling with spectrum analysis on surfaces. ACM Trans on Graphics (Proc SIGGRAPH Asia) 2010;29.

[19] Geng B, Zhang H, Wang H, Wang G. Approximate Poisson disk sampling on mesh. SCIENCE CHINA Information Sciences 2013;56(9):1–12.

[20] Ying X, Xin SQ, Sun Q, He Y. An intrinsic algorithm for parallel Poisson disk sampling on arbitrary surfaces. IEEE Trans on Vis and Comp Graphics 2013;19(9):1425–37.

[21] Ying X, Li Z, He Y. A parallel algorithm for improving the maximal property of Poisson disk sampling. Computer-Aided Design 2014;46(9):37–44.

[22] Peyrot JL, Payan F, Antonini M. Direct blue noise resampling of meshes of arbitrary topology. The Visual Computer 2014;Accepted.

[23] Yan DM, Wonka P. Gap processing for adaptive maximal Poisson-disk sampling. ACM Trans on Graphics 2013;32(5):148:1–148:15.

[24] Guo J, Yan DM, Jia X, Zhang X. Efficient maximal Poisson-disk sampling and remeshing on surfaces. Computers & Graphics 2015;46(6-8):72–9.

[25] Fu Y, Zhou B. Direct sampling on surfaces for high quality remeshing. In: ACM symposium on Solid and physical modeling. 2008, p. 115–24.

[26] Dunbar D, Humphreys G. A spatial data structure for fast Poisson-disk sample generation. ACM Trans on Graphics (Proc SIGGRAPH) 2006;25(3):503–8.

[27] Yan DM, Lévy B, Liu Y, Sun F, Wang W. Isotropic remeshing with fast and exact computation of restricted Voronoi diagram. Computer Graphics Forum 2009;28(5):1445–54.

[28] Xu Y, Hu R, Gotsman C, Liu L. Blue noise sampling of surfaces. Computers & Graphics 2012;36(4):232–40.

[29] Yan DM, Guo J, Jia X, Zhang X, Wonka P. Blue-noise remeshing with farthest point optimization. Computer Graphics Forum (Proc SGP) 2014;33(5):167–76.

[30] Schlömer T, Heck D, Deussen O. Farthest-point optimized point sets with maximized minimum distance. In: High Performance Graphics Proceedings. 2011, p. 135–42.

[31] Aurenhammer F. Power diagrams: Properties, algorithms and applications. SIAM J Comput 1987;16:78–96.

[32] Brenier Y. Polar factorization and monotone rearrangement of vector-valued functions. Communications on Pure and Applied Mathematics 1991;44:375–417.

[33] Mérigot Q. A multiscale approach to optimal transport. Comput Graph Forum 2011;30(5):1583–92.

[34] Gu X, Luo F, Sun J, Yau ST. Variational principles for minkowski type problems, discrete optimal transport, and discrete monge-ampere equations. arXiv preprint arXiv:13025472 2013;.

[35] Su Z, Sun J, Gu X, Luo F, Yau ST. Optimal mass transport for geometric modeling based on variational principles in convex geometry. Engineering with Computers 2014;30(4):475–86.

[36] Mérigot Q, Oudet E. Discrete optimal transport: complexity, geometry and applications. Tech. Rep.; 2014. URL: `http://hal.univ-grenoble-alpes.fr/hal-00980195`.

[37] David Bourne SR. Centroidal power diagrams, Lloyd's algorithm and applications to optimal location problems. arXiv preprint arXiv:14092786 2014;URL: `http://arxiv.org/abs/1409.2786`.

[38] Lévy B. A numerical algorithm for L2 semi-discrete optimal transport in 3D. 2014. URL: `https://hal.inria.fr/hal-01105021`; no.

[39] Villani C. Optimal transport, old and new. Springer; 2009.

[40] Amenta N, Bern M, Kamvysselis M. A new Voronoi-based surface reconstruction algorithm. In: Proc. ACM SIGGRAPH. 1998, p. 415–21.

[41] Nocedal J, Wright SJ. Numerical Optimization. Springer; 2006.

[42] Liu Y, Wang W, Lévy B, Sun F, Yan DM, Lu L, et al. On centroidal Voronoi tessellation - energy smoothness and fast computation. ACM Trans on Graphics 2009;28(4):101:1–17.

[43] CGAL, Computational Geometry Algorithms Library. 2015. Http://www.cgal.org.

[44] Lévy B. Restricted voronoi diagrams for (re)-meshing surfaces and volumes. In: Curves and Surfaces. 2014,Code download: `http://gforge.inria.fr/projects/geogram/`.

[45] Wei LY, Wang R. Differential domain analysis for non-uniform sampling. ACM Trans on Graphics (Proc SIGGRAPH) 2011;30:50:1–8.

Appendix A. Reynolds Transport Theorem

The derivation of an integral function $\mathbf{f} = \mathbf{f}(\mathbf{x}, t)$ over the time-dependent region $\Omega(t)$ that has boundary $\partial\Omega(t)$ with respect to time t is in the following form:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega(t)} \mathbf{f}\ \mathrm{dV} = \int_{\Omega(t)} \frac{\partial \mathbf{f}}{\partial t}\ \mathrm{dV} + \int_{\partial\Omega(t)} (\mathbf{v}^b \cdot \mathbf{n})\mathbf{f}\mathrm{dA},$$

where $\mathbf{n}(\mathbf{x}, t)$ is the outward-pointing unit-normal, $\mathbf{x}$ is a point in the region and is the variable of integration, dV and dA are volume and surface elements at $\mathbf{x}$, and $\mathbf{v}^b(\mathbf{x}, t)$ is the velocity of the area element.

Appendix B. Gradient Derivation on Surfaces

In this appendix, we derive the gradient ∇_{w_i} and $\nabla_{\mathbf{x}_i}$ of the objective function. We assume that when applying a sufficiently small perturbation to the weight w_i or the location of $\mathbf{x}_i$, only the shapes of the Voronoi regions $\{V_j | j \in \Omega_i\}$ will change.

We denote by e_{ij} the edge connecting the sites $\mathbf{x}_i$ and $\mathbf{x}_j$, e^*_{ij} the bisecting plane of the weighted sites $\mathbf{x}_i$ and $\mathbf{x}_j$, $|\cdot|$ the length of an edge, $|e_{ij}|_\tau$ the length of the projection of e_{ij} onto the triangle τ, $\mathcal{T}_{ij}$ the index set of the triangles in the mesh that intersect with the Voronoi face e^*_{ij}, and $\bar{\rho}_{ij}$ the average value of ρ over $e^*_{ij} \cap \mathcal{S}$.

Let $m_i = \int_{V_{i|S}} \rho(\mathbf{x}) d\sigma$. Since for a fixed domain, the partition of the density function $\rho(\mathbf{x})$ into cells $V_{i|S}$ sums up to a constant, i.e.,

$$\sum_i m_i = m_\gamma, \tag{B.1}$$

we take derivative of (B.1) w.r.p to w_i and $\mathbf{x}_i$:

$$\begin{aligned} \nabla_{w_i} m_i + \sum_{j\in\Omega_i} \nabla_{w_i} m_j = 0 \\ \nabla_{\mathbf{x}_i} m_i + \sum_{j\in\Omega_i} \nabla_{\mathbf{x}_i} m_j = 0 \end{aligned} \tag{B.2}$$

Figure B.15 illustrates the notations of the RVD used in the following proof.

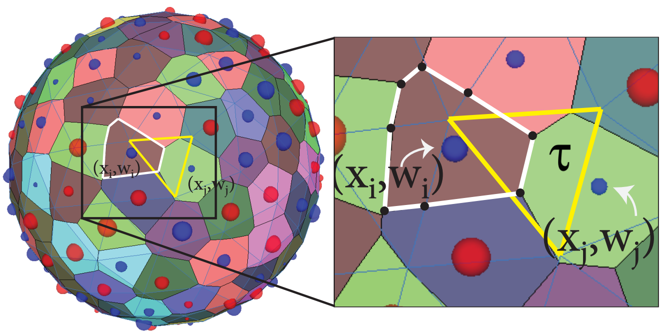

Figure B.15: Illustration of the notations of restricted power diagram. A triangle of input mesh is denoted as τ. The intersection of the triangle with a bisecting plane of two neighboring cells i, j is shown in white.

Lemma 1.

$$\nabla_{w_i} m_j = -\frac{\bar{\rho}_{ij}}{2} \sum_{l\in\mathcal{T}_{ij}} \frac{|e^*_{ij} \cap \tau_l|}{|e_{ij}|_{\tau_l}}.$$

Proof: By Reynolds' theorem, noticing that $\rho(\mathbf{x})$ is independent of $(\mathbf{x}_i, w_i)$, we have

$$\nabla_{w_i} m_j = \sum_{k\in\Omega_j} \sum_{l\in\mathcal{T}_{jk}} \int_{e^*_{jk}\cap\tau_l} \rho(\mathbf{x})\mathbf{v}_{w_i} \cdot \mathbf{b} ds = -\sum_{l\in\mathcal{T}_{ji}} \int_{e^*_{ij}\cap\tau_l} \rho(\mathbf{x})\mathbf{v}_{w_i} \cdot \mathbf{b} ds, \tag{B.3}$$

where Ω_j is the index set of the cells that are adjacent with $V_{j|S}$, $\mathbf{v}_{w_i} = \nabla_{w_i}\mathbf{x}$ for those intersection points $\mathbf{x}$ of the bisecting plane e^*_{jk} and a mesh triangular τ_l (with normal $\mathbf{n}_{\tau_l}$ and a vertex $\mathbf{p}_{\tau_l}$), $\mathbf{b}$ is the outpointing normal at the boundary points.

Now we formulate $\mathbf{v}_{w_i}$ by writing out the explicit representation of the intersection point $\mathbf{x}$:

$$\begin{aligned} (\mathbf{x}_j - \mathbf{x}_i) \cdot (\mathbf{x} - \mathbf{c}_{ij}) = 0 \\ (\mathbf{x} - \mathbf{p}_{\tau_l}) \cdot \mathbf{n}_{\tau_l} = 0, \end{aligned} \tag{B.4}$$

where

$$\mathbf{c}_{ij} = \mathbf{x}_i + \frac{d_{ij}}{|e_{ij}|}(\mathbf{x}_j - \mathbf{x}_i), \quad d_{ij} = \frac{|e_{ij}|^2 + w_i - w_j}{2|e_{ij}|}$$

Taking the derivative ∇_{w_i} of (B.4) yields:

$$\begin{aligned} \nabla_{w_i}\mathbf{x} \cdot (\mathbf{x}_j - \mathbf{x}_i) = \frac{1}{2} \\ \nabla_{w_i}\mathbf{x} \cdot \mathbf{n}_{\tau_l} = 0 \end{aligned} \tag{B.5}$$

Noticing that the unit normal $\mathbf{b}$ is given by

$$\mathbf{b} = \frac{(\mathbf{x}_j - \mathbf{x}_i) - ((\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbf{n}_{\tau_l})\mathbf{n}_{\tau_l}}{\|(\mathbf{x}_j - \mathbf{x}_i) - ((\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbf{n}_{\tau_l})\mathbf{n}_{\tau_l}\|} \tag{B.6}$$

Hence

$$\nabla_{w_i}\mathbf{x} \cdot \mathbf{b} = \frac{1}{2\|(\mathbf{x}_j - \mathbf{x}_i) - ((\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbf{n}_{\tau_l})\mathbf{n}_{\tau_l}\|} = \frac{1}{2|e_{ij}|_{\tau_l}}. \tag{B.7}$$

Substituting (B.7) back to (B.3) gives

$$\nabla_{w_i} m_j = -\sum_{l\in\mathcal{T}_{ij}} \frac{1}{2|e_{ij}|_{\tau_l}} \int_{e^*_{ij}\cap\tau_l} \rho(\mathbf{x}) ds = -\frac{\bar{\rho}_{ij}}{2} \sum_{l\in\mathcal{T}_{ij}} \frac{|e^*_{ij} \cap \tau_l|}{|e_{ij}|_{\tau_l}}. \tag{B.8}$$

Lemma 2.

$$\nabla_{\mathbf{x}_i} m_j = \sum_{l\in\mathcal{T}_{ij}} \frac{-\int_{e^*_{ij}\cap\tau_l} \rho(\mathbf{x})\mathbf{x} ds}{|e^*_{ij}|_{\tau_l}} - \sum_{l\in\mathcal{T}_{ij}} \frac{|e^*_{ij} \cap \tau_l|}{|e^*_{ij}|_{\tau_l}} \bar{\rho}_{ij}\mathbf{m}_{ij}, \tag{B.9}$$

where

$$\mathbf{m}_{ij} = -\mathbf{x}_i + (1 - \frac{2d_{ij}}{|e_{ij}|})(\mathbf{x}_j - \mathbf{x}_i).$$

Proof. The derivation is similar to [1] the previous proof, hence we directly write out

$$\nabla_{\mathbf{x}_i} m_j = \sum_{l\in\mathcal{T}_{ij}} \int_{e^*_{ji}\cap\tau_l} \rho(\mathbf{x})\mathbf{b}\mathbf{v}_{\mathbf{x}_i} ds = -\sum_{l\in\mathcal{T}_{ij}} \int_{e^*_{ij}\cap\tau_l} \rho(\mathbf{x})\mathbf{b}\mathbf{v}_{\mathbf{x}_i} ds, \tag{B.10}$$

[1] A slight difference here is that $\mathbf{x}_i$ is now a vector. Taking the derivative of any vector $\mathbf{f} = (f_1, f_2, f_3)$ w.r.p. to $\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3})$ gives a matrix, i.e., $\nabla_{\mathbf{x}_i}\mathbf{f} = (f_{jk})_{3\times3}$, whose element $f_{jk} = \nabla_{x_{ik}} f_j$. Correspondingly, the vector dot-product in (B.5) now becomes the matrix production

where $\mathbf{v}_{\mathbf{x}_i}$ now represents $\nabla_{\mathbf{x}_i}\mathbf{x}$ for those boundary point $\mathbf{x}$. The formulation of these boundary point $\mathbf{x}$ has already been provided by equation (B.4). So we now take the derivative for (B.4):

$$\begin{aligned}(\mathbf{x}_j - \mathbf{x}_i)\nabla_{\mathbf{x}_i}\mathbf{x} &= (\mathbf{x} - \mathbf{x}_i) + (1 - \frac{2d_{ij}}{|e_{ij}|})(\mathbf{x}_j - \mathbf{x}_i) \\ \mathbf{n}_{\tau_l}\nabla_{\mathbf{x}_i}\mathbf{x} &= 0.\end{aligned} \tag{B.11}$$

The outpoint normal $\mathbf{b}$ still preserves the representation in (B.6). Hence

$$\mathbf{b}\nabla_{\mathbf{x}_i}\mathbf{x} = \frac{(\mathbf{x} - \mathbf{x}_i) + (1 - \frac{2d_{ij}}{|e_{ij}|})(\mathbf{x}_j - \mathbf{x}_i)}{|e^*_{ij}|_{\tau_l}}. \tag{B.12}$$

Substituting (B.12) back to (B.10) gives

$$\begin{aligned}\nabla_{\mathbf{x}_i}m_j &= \sum_{l\in\mathcal{T}_{ij}} \frac{-\int_{e^*_{ij}\cap\tau_l}\rho(\mathbf{x})\mathbf{x}ds - \mathbf{m}_{ij}\int_{e^*_{ij}\cap\tau_l}\rho(\mathbf{x})ds}{|e^*_{ij}|_{\tau_l}} \\ &= \sum_{l\in\mathcal{T}_{ij}} \frac{-\int_{e^*_{ij}\cap\tau_l}\rho(\mathbf{x})\mathbf{x}ds}{|e^*_{ij}|_{\tau_l}} - \sum_{l\in\mathcal{T}_{ij}} \frac{|e^*_{ij}\cap\tau_l|}{|e^*_{ij}|_{\tau}}\bar{\rho}_{ij}\mathbf{m}_{ij},\end{aligned} \tag{B.13}$$

where

$$\mathbf{m}_{ij} = -\mathbf{x}_i + (1 - \frac{2d_{ij}}{|e_{ij}|})(\mathbf{x}_j - \mathbf{x}_i).$$

492 *Appendix B.1. Total Cost Change Rate*

The total cost is defined by

$$\mathcal{E}(X, W) = \sum_i \int_{V_{i|S}} \rho(\mathbf{x})\|\mathbf{x} - \mathbf{x}_i\|^2 d\mathbf{x} \tag{B.14}$$

Theorem 3.

$$\nabla_{\mathbf{x}_i}\mathcal{E} = 2m_i(\mathbf{x}_i - \mathbf{b}_i) + \sum_{j\in\Omega_i}(w_j - w_i)\nabla_{\mathbf{x}_i}m_j, \tag{B.15}$$

where

$$\mathbf{b}_i = \frac{\int_{V_{i|S}} \mathbf{x}\rho(\mathbf{x})d\mathbf{x}}{m_i}.$$

Proof. By B.12,B.13,

$$\begin{aligned}\nabla_{\mathbf{x}_i}\mathcal{E} &= \int_{V_{i|S}} \nabla_{\mathbf{x}_i}(\rho(\mathbf{x})\|\mathbf{x} - \mathbf{x}_i\|^2)d\mathbf{x} \\ &+ \sum_{j\in i\cup\Omega_i}\int_{\partial V_{j|S}} \rho(\mathbf{x})\|\mathbf{x} - \mathbf{x}_i\|^2(\nabla_{\mathbf{x}_i}\mathbf{x}\cdot\mathbf{b})ds \\ &= 2m_i(\mathbf{x}_i - \mathbf{b}_i) + \sum_{j\in\Omega_i}(w_j - w_i)\nabla_{\mathbf{x}_i}m_j\end{aligned} \tag{B.16}$$

Theorem 4.

$$\nabla_{w_i}\mathcal{E} = \sum_{j\in\Omega_i}(w_j - w_i)\nabla_{w_i}m_j, \tag{B.17}$$

493 **Proof.** The proof is similar to above using Lemma 1.

494 *Appendix B.2. New Functional*

We use the new energy functional

$$\mathcal{F}(X, W) = \mathcal{E}(X, W) - \sum_i w_i(m_i - m)$$

Theorem 5.

$$\begin{aligned}\nabla_{w_i}\mathcal{F}(X, W) &= m - m_i \\ \nabla_{\mathbf{x}_i}\mathcal{F}(X, W) &= 2m_i(\mathbf{x}_i - \mathbf{b}_i)\end{aligned} \tag{B.18}$$

Proof. By Theorem 4 and by equation (B.2), we have

$$\begin{aligned}\nabla_{w_i}\mathcal{F}(X, W) &= \nabla_{w_i}\mathcal{E}(X, W) - (m_i - m) - \sum_{j\in\Omega_i}(w_j - w_i)\nabla_{w_i}m_j \\ &= m - m_i.\end{aligned} \tag{B.19}$$

By Theorem 3 and by equation (B.2), we have

$$\begin{aligned}\nabla_{\mathbf{x}_i}\mathcal{F}(X, W) &= \nabla_{\mathbf{x}_i}\mathcal{E}(X, W) - \sum_{j\in\Omega_i}(w_j - w_i)\nabla_{\mathbf{x}_i}m_j \\ &= 2m_i(\mathbf{x}_i - \mathbf{b}_i)\end{aligned} \tag{B.20}$$

495 By (2), Lemma 1 and Theorem 5 we directly have

Theorem 6.

$$\begin{aligned}[H_{\mathcal{F}}]_{ij} &= \frac{\bar{\rho}_{ij}}{2}\sum_{l\in\mathcal{T}_{ij}}\frac{|e^*_{ij}\cap\tau_l|}{|e_{ij}|_{\tau_l}} \\ [H_{\mathcal{F}}]_{ii} &= \sum_{j\in\Omega_i}[H_{\mathcal{F}}]_{ij}.\end{aligned} \tag{B.21}$$