



Applications of Geometry Processing

Blue noise sampling of surfaces

Yin Xu^a, Ruizhen Hu^a, Craig Gotsman^b, Ligang Liu^{a,*}^a Department of Mathematics, Zhejiang University, Hangzhou 310027, China^b Computer Science Faculty, Technion, Israel Institute of Technology, Haifa 32000, Israel

ARTICLE INFO

Article history:

Received 18 August 2011

Received in revised form

30 January 2012

Accepted 6 February 2012

Available online 21 February 2012

Keywords:

Blue noise

Capacity-Constrained Surface Triangulation (CCST)

Relaxation-based

Minimal area variance

ABSTRACT

We present an algorithm to generate point distributions with high-quality blue noise characteristics on discrete surfaces. It is based on the concept of Capacity-Constrained Surface Triangulation (CCST), which approximates the underlying continuous surface as a well-formed triangle mesh with uniform triangle areas. The algorithm takes a triangle mesh and the number of sample points as input, and iteratively alternates between optimization of the geometry (positions) of the points and optimization of their topology (connectivity) until convergence. Since the method is relaxation-based, it allows precise control over the number of sample points. Differential domain analysis shows that the point distribution of CCST exhibits typical blue noise characteristics, superior to other relaxation-based sampling methods and is very efficient compared to other traditional dart-throwing methods. We generalize CCST to non-uniform sampling by incorporating a density function. This can be useful in many geometry processing applications, such as curvature-aware remeshing.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

The problem of sampling, or point distribution, is ubiquitous in computer graphics [14]. It can be formulated as follows: given a domain S and the number of point samples n , position n points within the domain to form a pattern that satisfies some user-specified preferences. In practice, different applications require different patterns, but it seems that among numerous sampling patterns, the most useful are those which have so-called “blue noise” characteristic, which, in a nutshell, attest to high spatial uniformity and low regularity of the distribution. A good generator of blue noise distributions tends to replace low frequency aliasing with high frequency noise in order to be less visually objectionable [16], thus can be used in applications including rendering, sensing, imaging and geometry processing. One well-known class of sample patterns having the blue noise characteristic is the so-called *Poisson Disk* distribution [7], where the points are positioned such that the disk with an appropriate radius centered at each point is empty of other points.

Because of its importance, a large volume of work investigating the generation of blue noise sample patterns on the plane exists. However, the problem of efficiently generating such sample patterns on non-planar surfaces is more difficult and much less work focuses on this issue. Most of the existing methods use either surface

parameterization to reduce the surface case to the planar case, introducing distortions into the patterns, or extend the classical dart throwing algorithm [8] from the plane to the surface, thus are computationally expensive since geodesic distances must be used. Moreover, in some application domains such as LED displays, the number of “sample points” is required to be explicitly set in order to display an image using a given budget of LEDs, while the variants of the dart throwing algorithm cannot achieve this effect.

To meet the strong demand for an efficient and high-quality surface sampling method affording precise control over the number of points, we present a new relaxation-based approach for blue noise sampling on surfaces. It is an extension of the so-called *Capacity-Constrained Delaunay Triangulation* (CCDT) method [25], a Delaunay triangulation of a planar domain having as-uniform-as-possible triangle areas. The point distributions generated by CCDT have blue noise characteristics, and the algorithm runs much faster than all other state-of-the-art methods. We generalize the CCDT method to the surface case by restricting the movement of points to the surface, which is approximated by a triangular mesh, and minimizing the variance of triangle areas. We call this new method *Capacity-Constrained Surface Triangulation* (CCST) due to its uniform-area property. Taking a surface discretized as a piecewise-linear triangle mesh and the number of sample points as input, CCST generates a new triangle mesh lying on the input surface, having uniform area distributions. Similar to CCDT, the CCST algorithm consists of two alternating phases. The first phase optimizes the geometry (positions) of points to equalize the triangle areas with the current connectivity, and the second phase regenerates the topology

* Corresponding author.

E-mail addresses: yinxu.zju@gmail.com, jhxy2004@zju.edu.cn (Y. Xu), ruizhen.hu@gmail.com (R. Hu), gotsman@cs.technion.ac.il (C. Gotsman), ligangliu@zju.edu.cn (L. Liu).

(connectivity) while keeping the geometry fixed. Using frequency domain analysis, the sampling patterns obtained after convergence of the algorithm are shown to possess typical blue noise characteristics.

Although CCST can be considered a direct extension of the CCDT method, the fundamental difference between the flat two-dimensional plane and the curved 2-manifold surface in three dimensions implies that the most important and difficult part of the CCST algorithm is moving points along the surface. We solve this problem by approximating the local surface around each mesh vertex with an osculating torus, for which the approximation error is second order [18]. The objective energy which we try to optimize when moving the points on the osculating torus is non-quadratic, thus we resort to a second-order Taylor expansion to obtain a quadratic approximation of the energy, reducing the problem to a 2×2 linear system per vertex.

The CCST algorithm can also be extended to non-uniform sampling by incorporating a density function into the objective energy. This is useful for generating point distributions having different sampling densities on different regions of the surface, which are useful in many sampling-based applications, such as curvature-aware sampling.

The contribution of this work is a simple algorithm for generating sample patterns on surfaces with the following advantages:

- The resulting patterns possess superior blue noise characteristics.
- The algorithm allows precise control over the number of sample points generated.
- The algorithm is very fast compared to other relevant methods.

2. Related work

A blue noise sample pattern contains spatially uniform points while keeping the distribution irregular. In this section we briefly review previous work on both the planar and surface sampling problems, and the methods used to analyze and evaluate the sample quality.

2.1. Blue noise sampling in the plane

Poisson disk distributions were introduced to solve the aliasing problem in computer graphics [7]. The classical algorithm for generating these distributions is the so-called *Dart Throwing* algorithm [8] which randomly positions points one by one and accepts a new point only if there are no other points within a disk of given radius surrounding it. It generates Poisson disk distributions with blue noise characteristics, but is very expensive in runtime. Since then, numerous alternatives have been proposed, which may be roughly classified into two types of approaches: variants of the dart throwing algorithm and relaxation-based methods. The former focuses directly on accelerating the dart throwing approach [9,11,22,23], and has had some success. However, using this approach makes it hard to control the number of points generated and still performs quite slowly, unless some parallelization is introduced into the algorithm.

Relaxation-based methods introduce some additional topological information into the point pattern. Starting with some inferior pattern, the algorithm progressively adjusts the point positions so that the distribution is as uniform as possible. The most obvious advantage of this class of methods is that it is easy to control the size of the point set. The simplest approach is *Lloyd's Relaxation* [17], which iteratively moves each point to the centroid of its Voronoi cell, thus converging to the well-known *Centroidal Voronoi Tessellation* (CVT). Lloyd's relaxation is quite

fast, but the limit sample exhibits significant regularity artifacts without blue noise characteristics. Balzer et al. [4] proposed a variant of Lloyd's relaxation called *Capacity-Constrained Voronoi Tessellation* (CCVT), forcing all the cells of the generalized centroidal Voronoi diagram of the points to have uniform areas. When extending CCVT to non-uniform sampling, the triangle area is replaced by "capacity", which is the integral of the density function per cell. The CCVT method produces distributions with good blue noise characteristics, but is extremely slow due to the fact that it approximates capacity numerically by densely subsampling the domain.

More recently, a new kind of relaxation-based method called *Capacity-Constrained Delaunay Triangulation* (CCDT) [25] was proposed, aiming to efficiently generate blue noise samples. Contrary to the CCVT algorithm, the CCDT method operates on the dual of the Voronoi diagram, namely the Delaunay triangulation, and tries to minimize the variance of triangle capacities. The algorithm works directly in the continuous plane, thus performs much faster than other methods, with no compromise in the quality of the blue noise characteristics.

Apart from the above two categories, a new kernel-based method [12] was proposed most recently, and the results are shown to possess excellent blue noise characteristics while having linear time complexity. However, it also cannot control the number of sampling points in precise.

2.2. Blue noise sampling on surfaces

The problem of blue noise sampling on surfaces has become popular in recent years in graphics applications focusing on rendering, remeshing and texturing. The existing approaches may be classified into three types. The first one relies on parameterizing the surface to the plane, thus reducing the problem to two-dimensional blue noise sampling, and mapping the 2D sample patterns back to the surface [2]. However, this approach might introduce distortion into the distribution when mapping from 3D to 2D and back. The second set of methods generates Poisson disk distributions directly on the surface [5,6]. These are mostly variants of the dart throwing algorithm, which, as mentioned above, require computation of geodesic distances, making for a relatively slow algorithm. Also, as in the planar case, it is hard to control the sample size in dart throwing algorithms. The third set of methods is relaxation-based methods [20,15], which overcome the problems of efficiency and controllability of sampling size but may fail to provide high-quality blue noise characteristics. Our CCST algorithm also belongs to the class of relaxation-based methods, while, as we will show, is able to provide much better blue noise properties than other algorithms in this class.

2.3. Evaluation of blue noise sampling

The methodology for analysis of blue noise sampling in the plane was first developed by Ulichney [21]. It is based on the Barlett's method [3] to estimate the power spectrum through averaging the periodograms of distributions, determined by Fourier transforms. The radially symmetric periodograms result in two one-dimensional measures. One is the so-called *Radially Averaged Power Spectrum*, in which the typical blue noise characteristic should start with a sharp transition region, a low-frequency cutoff and a flatter high-frequency region. The second one is called *Anisotropy*, which measures the radial symmetry of the power spectrum and indicates high symmetry when the curves are low and flat [14].

The evaluation of blue noise sampling on surfaces is difficult since the typical Fourier analysis cannot be extended simply to surfaces. Bowers et al. [5] addressed this by replacing the Fourier

basis on manifold surfaces with spectral mesh analysis [13], reducing the problem of power spectrum analysis to eigen-decomposition of the mesh Laplacian matrix. The radially averaged power spectrum and anisotropy of a point distribution on the surface then proceeds similarly to the planar case. This is the first method that addresses spectral analysis of surface samples, but is significantly slow and hard to use due to the fact that the eigen-decomposition must be performed on a very dense mesh approximating the true underlying surface.

Most recently, Wei and Wang [24] proposed an approach to analyzing point distributions that generalizes easily to non-uniform distributions on both the planar and surfaces. It replaces the standard Fourier analysis by *differential domain analysis*, which depends primarily on the distribution of the pairwise distances. Here too the statistics are averaged radially and tangentially around the origin to provide two one-dimensional curves.

3. Capacity-Constrained Surface Triangulations (CCST)

In this section, we first give an overview of the CCST algorithm and the notations, and then we elaborate on each step in detail.

3.1. CCST algorithm overview

Let S be a 2-manifold surface in three-dimensional space, discretized to a piecewise-linear triangle mesh M , and n the number of desired sample points. The CCST algorithm starts by randomly generating an initial point set X_0 on M and introducing triangular topology (connectivity) information T_0 to X_0 to form a *sample mesh*: $M_0=(X_0,T_0)$. It then alternates between moving the vertices of the sample mesh on S and rebuilding the sample mesh connectivity in order to achieve a sample mesh with well-shaped triangles having uniform areas.

Algorithm 1. CCST Algorithm

1. Generate initial point distribution X_0 on M .
2. Build initial topology T_0 from X_0 . $n \leftarrow 0$. $E_0 \leftarrow \sum_t A_t^2(X_0, T_0)$,
where $\sum_t A_t^2(X_0, T_0)$ is the sum of squared triangle areas formed by (X_0, T_0) .
3. Densely discretize M and compute its principal curvature information.
4. Geometry optimization on S : $X_{n+1} = \text{argmin}_t \sum_t A_t^2(X_n, T_n)$.
5. Rebuild topology T_{n+1} from X_{n+1} by edge flipping.
6. $M_{n+1} = (X_{n+1}, T_{n+1})$. $E_{n+1} \leftarrow \sum_t A_t^2(X_{n+1}, T_{n+1})$.
7. If $E_{n+1} = E_n$, output X_{n+1} and stop.
8. Else $n \leftarrow n + 1$, Go to Step 4.

3.2. Initialization of point distribution

Since S is given as a piecewise-linear triangular mesh M , all the n points of X_0 should be distributed uniformly on the surface of M , meaning smaller triangles should contain fewer points. A single point may be generated by first choosing a triangle with probability proportional to its area, and then generating a point randomly with uniform distribution within that triangle. This process is repeated until the size of X_0 reaches n .

After the generation of X_0 , a triangular topology T_0 is built to form an initial mesh $M_0=(X_0,T_0)$ in order to continue the iterative algorithm. We use the Crust algorithm [1] to build an initial

triangulation on X_0 and then improve its quality using the edge-flipping algorithm proposed by Dyer et al. [10] which produces a mesh with minimal surface area. T_0 is thus uniquely determined by X_0 and is closely related to the two-dimensional Delaunay triangulation due to the fact that it provides quite uniformly-shaped triangles.

3.3. Restricting movement on the surface

Unlike the CCDT algorithm, which restricts X to the plane, the CCST algorithm has to restrict X to remain on the surface mesh M , thus introducing additional constraints on the coordinates of the points of X . Following Pottman et al. [18], it is possible to approximate the surface at an arbitrary point \mathbf{s} up to second order using an osculating torus, which is obtained by rotating the first principal curvature circle c_1 centering \mathbf{k}_1 with signed radius r_1) around the axis of the other principal curvature circle c_2 , defined analogously to c_1 . Swapping the roles of c_1 and c_2 leads to two different tori, and usually the circle with larger radius is selected to be the mother circle. Here we assume $|r_1| > |r_2|$ such that c_2 is rotation around the axis of c_1 (see Fig. 1).

The position of an arbitrary point \mathbf{s}' on this osculating torus is uniquely determined by first rotating \mathbf{s} by θ along the axis of c_1 and then rotating by φ along the axis of c_2 . Since the rotation axes are orthogonal, the position is invariant to the order of the rotations.

In practice, since the underlying surface S is approximated by a piecewise-linear triangle mesh M , we use the Rusinkiewicz's method [19] to compute the two principal curvatures of M : (T_1, ρ_1) and (T_2, ρ_2) where $\rho_1 = 1/r_1$, $\rho_2 = 1/r_2$. For an arbitrary point \mathbf{s} on a smooth surface S , there always exists a three-dimensional rotation matrix R_s that rotates the local Darboux frame such that the two principal-curvature directions T_1 and T_2 coincide with the x and y axis respectively, and the normal direction with the z axis. Thus, without loss of generality, we will always assume that the local Darboux frame of \mathbf{s} is the standard coordinate frame and \mathbf{s} is the origin $(0,0,0)$. Using Euclidean geometry, Appendix A derives the coordinates of \mathbf{s}'

$$\mathbf{s}' = \begin{pmatrix} r_2 \sin \theta \cos \varphi + (r_1 - r_2) \sin \theta \\ -r_2 \sin \varphi \\ r_2 \cos \theta \cos \varphi + (r_1 - r_2) \cos \theta - r_1 \end{pmatrix} = \begin{pmatrix} x(\theta, \varphi) \\ y(\theta, \varphi) \\ z(\theta, \varphi) \end{pmatrix}$$

in terms of θ and φ . Since \mathbf{s}' is a second order approximation to the points on the underlying surface S , when θ and φ are close to zero, \mathbf{s}' can be regarded as lying on S .

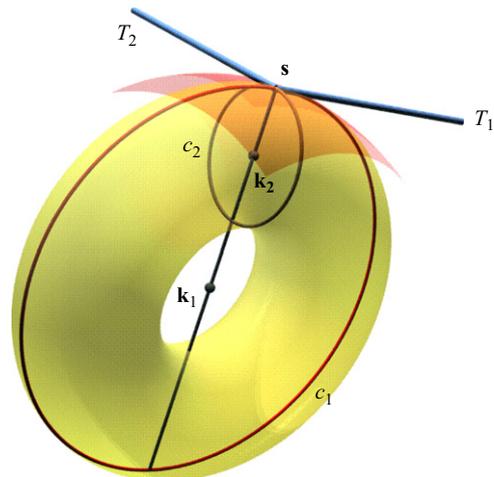


Fig. 1. Second order approximation of a point \mathbf{s} on surface using an osculating torus.

3.4. Geometry optimization on the surface

The CCVT and CCDT algorithms rely on the fact that area uniformity can overcome regularity artifacts, resulting in high-quality blue noise characteristics for point distributions in the plane. We extend this idea to a 2-manifold in 3D and try to equalize the triangle areas of the sample mesh M_n in order to generate a sample of the input surface mesh M with blue noise characteristics. In Step 4 of Algorithm 1, the geometry optimization scheme moves the vertices of M_n to their optimal position X_n on M , where the triangle areas $\{A_t\}$ should be as uniform as possible under the current topology T_n . By definition, the uniform area distribution minimizes its variance

$$\text{var}\{A(X,T)\} = \frac{1}{m} \sum_{t \in T} \left(A_t - \frac{1}{m} \sum_{t \in T} A_t \right)^2$$

where m is the number of triangles in T . The following theorem states that minimizing $\text{var}\{A(X,T)\}$ on a surface is equivalent to minimizing the sum of squared areas $\sum_t A_t^2$.

Theorem 1. For a triangular mesh $M = (X,T)$, minimizing the variance of triangle areas $\text{var}\{A(X,T)\}$ is equivalent to minimizing the sum of squared areas $\sum_t A_t^2$ when restricting the movement of X on M .

Proof. Since X is allowed to move only on M , the sum of all triangle areas is the constant surface area of M , denoted by $A = \sum_t A_t$. Then

$$\begin{aligned} \text{var}\{A(X,T)\} &= \frac{1}{m} \sum_{t \in T} \left(A_t - \frac{1}{m} \sum_{t \in T} A_t \right)^2 = \frac{1}{m} \sum_{t \in T} \left(A_t - \frac{1}{m} A \right)^2 \\ &= \frac{1}{m} \sum_{t \in T} A_t^2 - \frac{2A}{m} \sum_{t \in T} A_t + \frac{A^2}{m^2} = \frac{1}{m} \sum_{t \in T} A_t^2 - \frac{A^2}{m^2} \end{aligned}$$

Since both m and A are constants, minimizing $\text{var}\{A(X,T)\}$ is equivalent to minimization of $\sum_t A_t^2$. \square

By Theorem 1, the geometry optimization scheme should aim to minimize the sum of squared triangle areas $\sum_t A_t^2$. It does this locally, meaning it examines each vertex \mathbf{s}_i in turn, keeps its one-ring neighboring vertices fixed, and then moves \mathbf{s}_i to its optimal position \mathbf{s}_i^* on S such that the neighboring triangles $\{N(i)\}$ have minimal sum of squared areas

$$\sum_{t \in N(i)} A_t^2$$

This routine is performed in several passes over all the vertices of the sample mesh until the energy $\sum_t A_t^2$ converges.

Since \mathbf{s}_i^* is a function of θ and φ , the sum of squared areas of its neighboring triangles can also be represented as a function $f(\theta, \varphi)$, which is the special case of a more general representation (see next section) for the non-uniform sampling problem. The optimization of f is a non-linear system

$$\begin{cases} \frac{\partial f}{\partial \theta} = 0 \\ \frac{\partial f}{\partial \varphi} = 0 \end{cases}$$

which is numerically difficult to solve. Therefore, we use a second order Taylor expansion to approximate f as a quadratic function of θ and φ at $(0,0)$

$$\begin{aligned} g(\theta, \varphi) &= f(0,0) + \frac{\partial f}{\partial \theta}(0,0)\theta + \frac{\partial f}{\partial \varphi}(0,0)\varphi \\ &\quad + \frac{1}{2} \frac{\partial^2 f}{\partial \theta^2}(0,0)\theta^2 + \frac{1}{2} \frac{\partial^2 f}{\partial \varphi^2}(0,0)\varphi^2 + \frac{\partial^2 f}{\partial \theta \partial \varphi}(0,0)\theta\varphi \end{aligned}$$

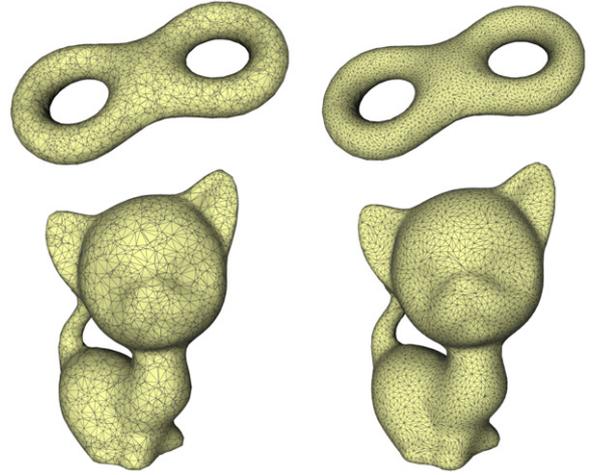


Fig. 2. Geometry optimization (right) of some initial sample meshes (left) containing 5000 samples. The input triangle meshes contain 7402 (eight) and 9756 (kitten) vertices.

As shown in Appendix A, f is quadratic in some trigonometric functions. The second-order Taylor expansion g is sufficient to approximate f precisely for small θ and φ . Thus, the minimization of f reduces to solving a linear system.

In order to preserve the shape of the sample set during the geometry optimization scheme, the vertex should be projected back to the input surface mesh M after each movement. We use the ANN [26] library to densely discretize M in a preprocessing step and then, given a query point, find the nearest point on M to estimate its projection. Fig. 2 shows some results of the geometry optimization scheme for a sample size of 5000 points. It is evident that the triangles of the resulting meshes have more uniform areas than the initial ones, while preserving the shape.

3.5. Rebuild topology from geometry

As seen in Fig. 2, the triangles of the sample mesh might become long and skinny during geometry optimization scheme due to the uniform area distribution. Hence the topology needs to be regenerated to form a more well-shaped triangulation. This is achieved by again applying the edge-flipping algorithm to the sample mesh with its new geometry (X_{n+1}, T_n) . This will, of course, require the geometry to be optimized again. Thus, the two steps are alternated until convergence.

Fig. 3 shows the evolution of the sample point set over the iterations of CCST, displaying the triangle meshes and point distributions as the iterations proceed. The energy curve is also shown, typically it will increase after rebuilding the topology (marked with red stars). Usually the energy converges in less than 10 iterations after rebuilding the topology.

4. CCST for non-uniform surface sampling

The CCST algorithm described in Section 3 generates uniform distributions on surfaces having good blue noise properties. However, in many cases, non-uniform distributions which conform to a given density are required. For instance, surface regions with high curvature may require denser point distributions to achieve a more precise sampling accuracy. In this section we extend the uniform CCST algorithm to the non-uniform case.

Given a piecewise-constant density function $\rho(T)$ defined on triangles of a mesh M , we generalize the concept of *area* to assign

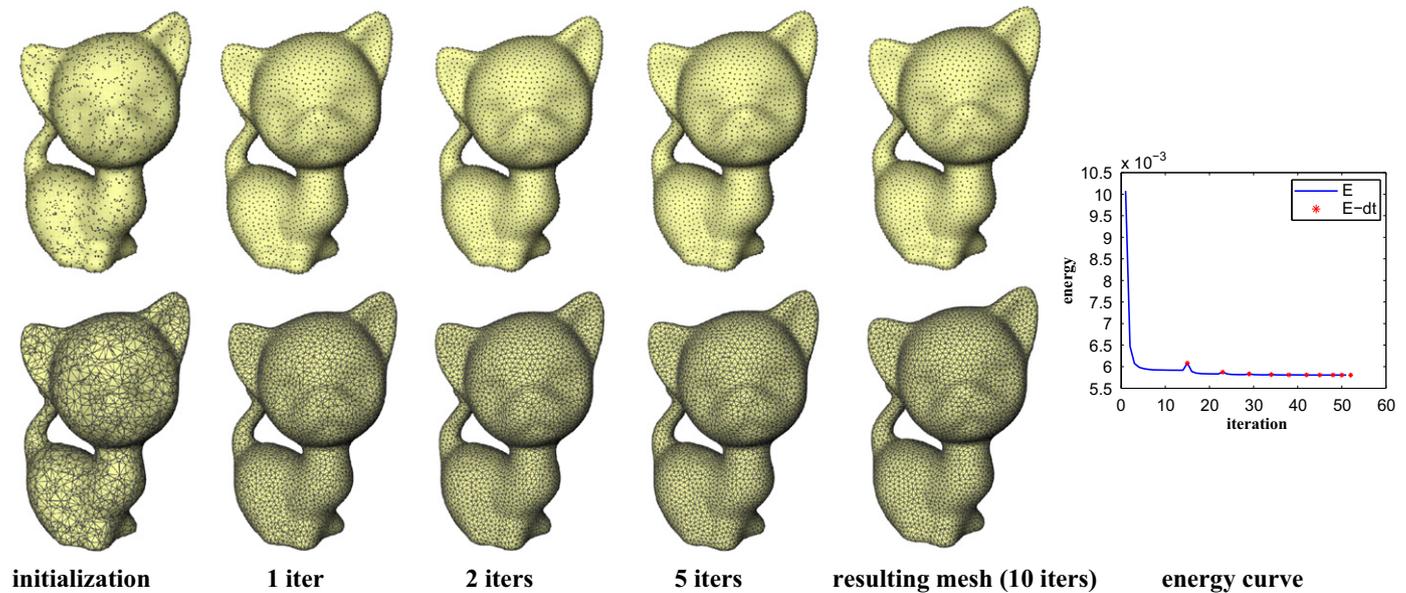


Fig. 3. Evolution of 5000 sample points on the kitten model during iterations of the CCST algorithm. Red stars mark a topology optimization phase (end of each iteration). Top: point distributions; bottom: the corresponding sample meshes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

to each triangle a *capacity* that integrates $\rho(T)$ over the triangle areas $C_t = A_t \rho_t$. The non-uniform CCST algorithm then aims at minimizing the sum of squared triangle capacities of the surface

$$X_{n+1} = \arg \min_X \sum_t C_t^2(X, T_n) \quad (1)$$

Appendix B shows the closed-form solution of each vertex when updating it in turn, assuming that $\rho(T)$ and the positions of all other vertices are kept fixed. In this case, Algorithm 1 can be directly applied by replacing the objective energy with the sum of squared capacities. Since $\rho(T)$ is outdated after several passes over all vertices, the geometry optimization scheme is terminated. $\rho(T)$ is updated after rebuilding the topology of M_n .

The number k of passes over all vertices in the geometry optimization phase depends on the density function. We found that a value of 5–10 is sufficient to produce good results. Some curvature-aware sampling results are shown in Fig. 4, where the density function is taken to be the square-root of the total Gaussian curvature on each triangle.

5. Experimental results

In this section, we show results of the CCST algorithm in both the uniform and non-uniform cases. We compare them with other competing algorithms and measure their quality using differential domain analysis. The efficiency of CCST is also analyzed.

5.1. Blue noise characteristics

We use the method of Wei and Wang [24], mentioned in Section 2.2, to perform differential domain analysis of surface samples. We compare CCST with a typical relaxation-based sampling method: Lloyd's relaxation on surfaces [15], and standard dart-throwing Poisson disk distribution on surfaces [5]. Each case is estimated by 8 sets with ~ 1700 samples. The results are shown in Fig. 5, where it is evident that the spectra of CCST algorithm reveals typical blue noise characteristics, similar to

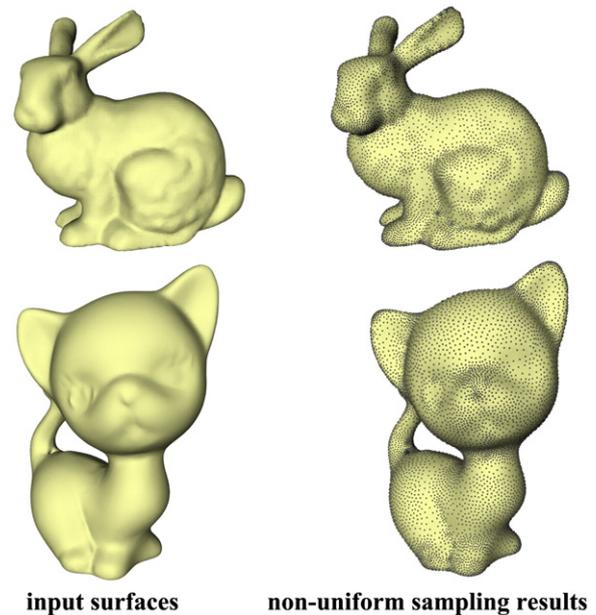


Fig. 4. Results of non-uniform placement of 10,000 points using CCST algorithm with Gaussian curvature-based density function. The input bunny model contains 6289 vertices and the kitten 9756 vertices.

those of the Poisson disk distribution and superior to those of Lloyd's relaxation.

5.2. Relationship with CCDT

The CCST algorithm can also be regarded as the generalization of the CCDT method for planar surfaces to curved surfaces. To verify this, we apply uniform CCST to a developable surface and map the points back to the plane. We then compare the result with the planar CCDT sampling result using Ulichney's planar spectral analysis [21], shown in Fig. 6. As expected, it turns out that the radially averaged power spectra and anisotropies of these two methods are quite similar.

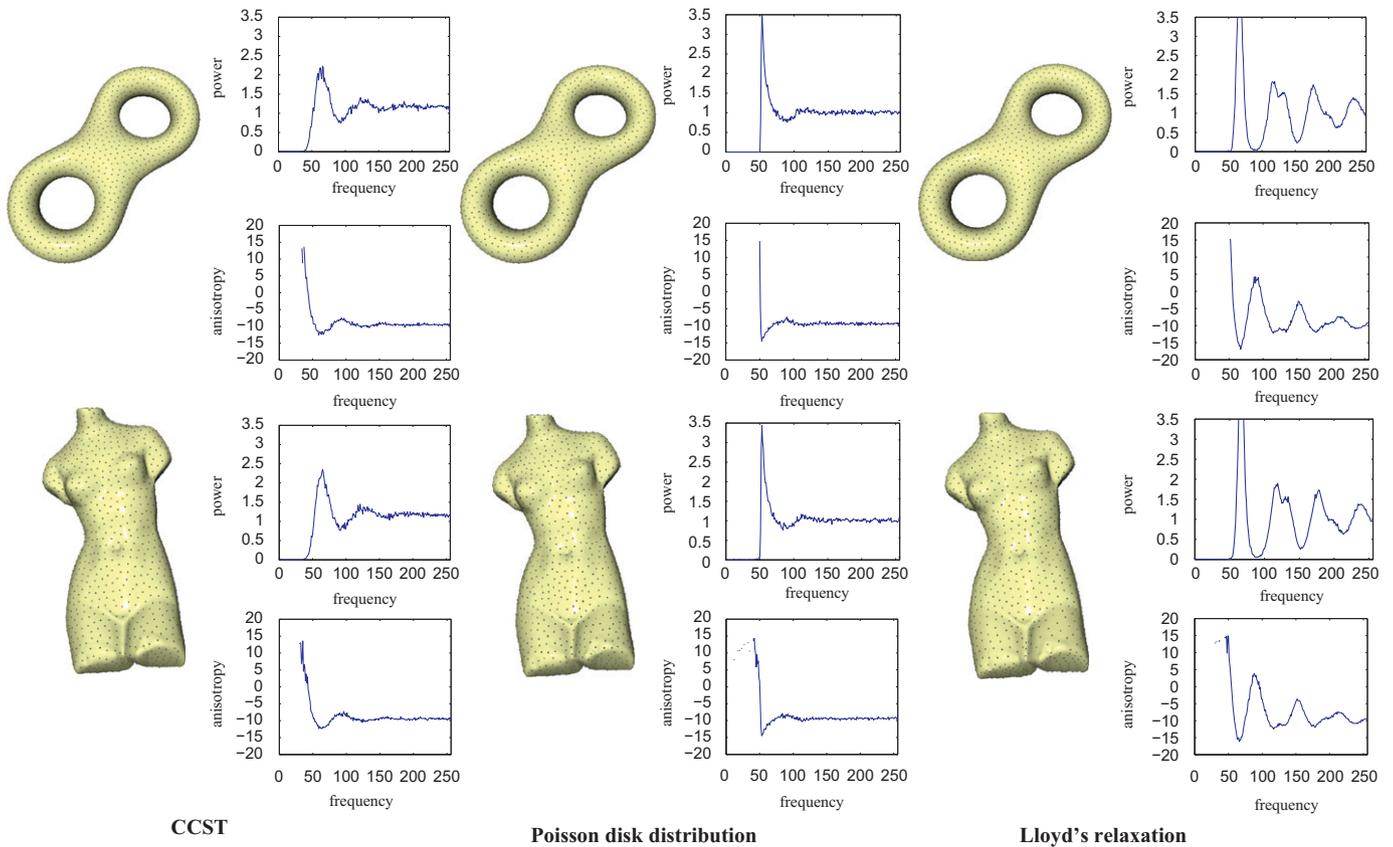


Fig. 5. Differential analysis of surface sampling distributions generated by different algorithms. Each graph is averaged over 8 sets with ~1700 samples. For each model, top: radial mean; bottom: anisotropy.

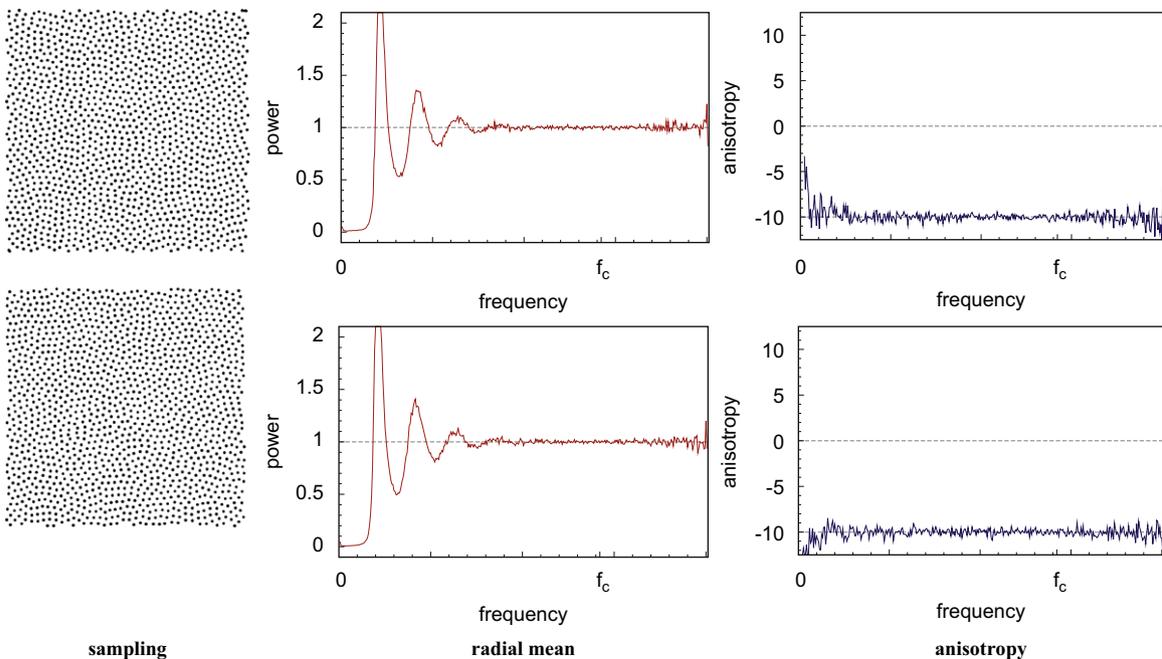


Fig. 6. Comparison of uniform planar sample distributions and their spectra. Top: CCDT, bottom: CCST on a developable surface.

5.3. Relationship with CCVT

As relaxation-based sampling methods, both CCST and CCDT could be regarded as the dual versions of CCVT method. CCVT operates on Voronoi diagrams while CCST/CCDT generates the dual Delaunay triangulations. All of these three methods start

with some random initializations and will converge to some local minimums of area variance that possess spatial uniformity of points after several alternations between geometry and topology phases. These local minimums also avoid from regularity artifacts that are caused by global minimums, namely all the triangles/cells are equilateral, thus achieving blue noise characteristics.

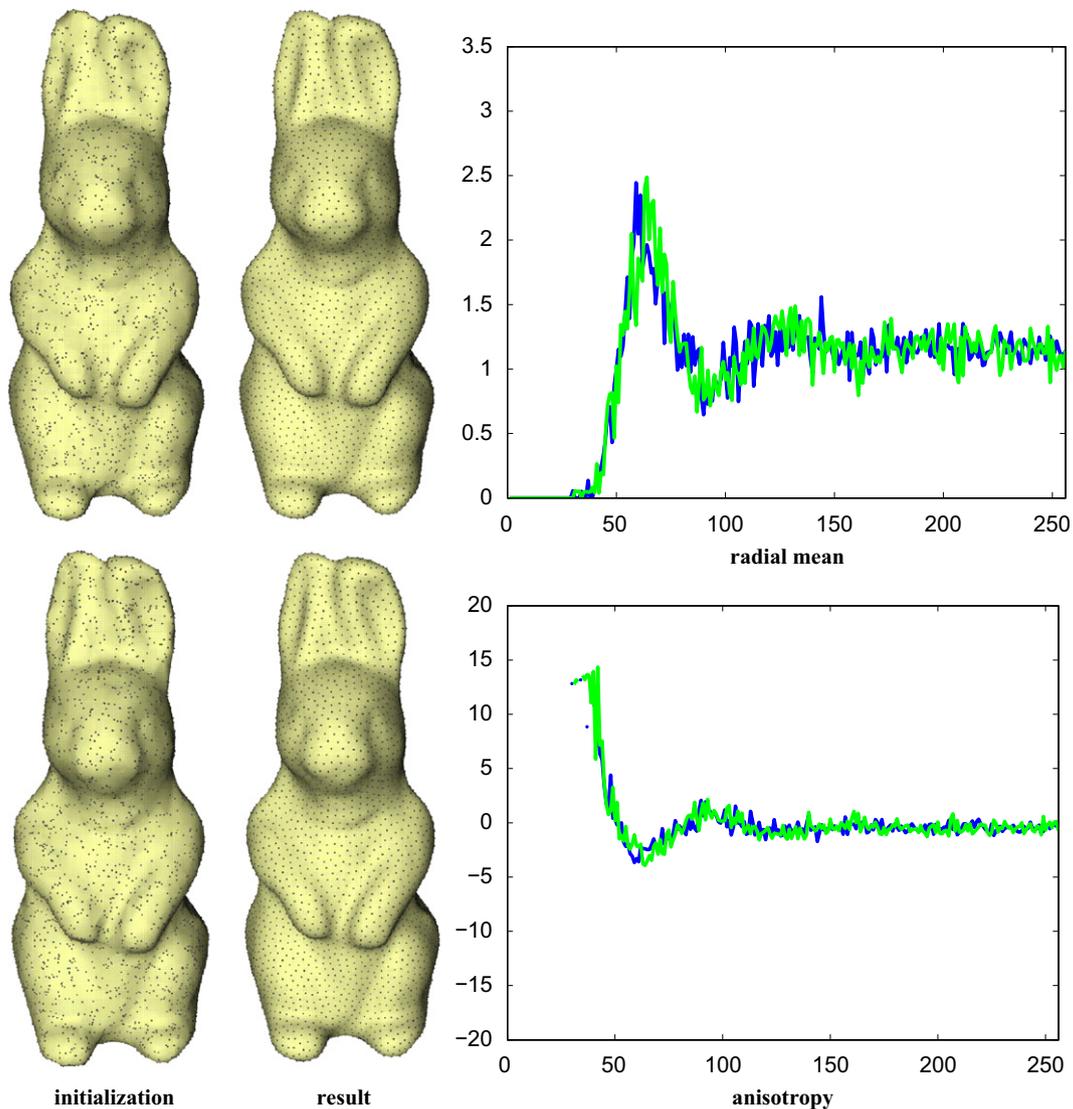


Fig. 7. Sensitivity to initial distribution. Outputs of the CCST algorithm starting from different initial distributions have similar patterns and similar radial mean and anisotropy plots.

The iterative algorithm of CCVT could be categorized into Lloyd's relaxation method, typically requiring about 100 alternations between geometry and topology phases. In contrast, CCST/CCDT method aims at directly minimizing variance of Delaunay triangle areas, leading the algorithm converges in less than 10 alternations between geometry and topology phases, thus is much faster than CCVT method [25].

5.4. Sensitivity to initialization

The CCST algorithm starts with a random initial distribution of n points. Fig. 7 shows two results of different initial sampling patterns and their corresponding differential analyses, which are quite similar.

5.5. Non-uniform sampling on surfaces

Algorithm 1, described in Section 4, can be applied to numerous types of density functions to non-uniformly sample points on surfaces, such as curvature-aware sampling. Another useful application is to sample a texture-mapped mesh to produce an image halftone effect on the surface. In this case, the grayscale value of the texture pixel is taken to be the density function. Fig. 8



Fig. 8. Non-uniform distributions generated by CCST algorithm using texture grayscale intensity as the density function.

shows an example generated using the non-uniform CCST algorithm.

5.6. Algorithm efficiency

The CCST algorithm alternates between modifying the geometry and the topology of the surface triangle mesh until convergence.

Since the geometry optimization computes the new position of a vertex by explicitly solving a 2×2 linear system and the edge-flipping re-triangulation routine is very efficient, the speed of algorithm is comparable with other relaxation-based methods such as retiling [20] and Lloyd’s relaxation on surfaces [15]. Apart from the preprocessing routine from Step 1 to Step 3 in Algorithm 1, the theoretical time complexity of CCST consists of two parts. Although the topology modification phase of CCST has time complexity $O(N \log N)$ where N is the number of sampling points, its run time is dominated by the geometry optimization phase, whose time complexity is $O(N)$, as opposed to the time complexity of $O(N \log N)$ of most dart throwing-based methods which need to compute geodesic distances. CCST can be accelerated significantly using GPGPU parallelization for the geometry optimization.

6. Conclusion

We have presented the CCST algorithm—a novel and efficient approach to generate point distributions on a surface. The algorithm alternates between a geometry optimization phase, which minimizes the variance of triangle capacities, and an edge-flipping based re-triangulation phase, until convergence. When applied to a uniform distribution, the result has been shown to possess superior blue noise characteristics. Our method is the first relaxation-based approach to generate blue noise samples on surfaces, thus easier to control the number of points generated, compared with other variants of the dart throwing algorithm. As an application, the density function can use surface curvature information to generate curvature-aware samples. The CCST algorithm is very efficient compared to other dart throwing-based methods. It can also be accelerated significantly using the GPGPU parallelization.

A limitation of our work is that, ironically, in order to achieve blue noise characteristics, the objective energy has to converge to a local minimum. A global minimum, where all the triangles are equilateral with identical areas, introduces significant regularity artifacts into the sampling patterns. Thus a sufficiently random initialization is required for the CCST algorithm.

Acknowledgments

We thank Rui Wang for providing source code for differential domain analysis of point sets, and Dongming Yan for providing an executable for generating sample patterns on surfaces. We also thank Zhonggui Chen and Renjie Chen for helpful comments during our work. The work of L. Liu was partially supported by the National Natural Science Foundation of China (61070071), the 973 National Key Basic Research Foundation of China (2009CB320801), and the Fundamental Research Funds for the Central Universities (2010QNA3039).

Appendix A. Moving on a surface

For an arbitrary point \mathbf{s} on a smooth surface S , we assume that the local Darboux frame of \mathbf{s} is the standard coordinate frame with \mathbf{s} at the origin $(0,0,0)$. We derive the coordinates of \mathbf{s}' , an arbitrary point on the osculating torus of \mathbf{s} . As mentioned in Section 3.2, \mathbf{s}' is uniquely determined by two rotation angles θ

and φ around orthogonal axes c_1 and c_2 . For simplicity, we first rotate \mathbf{s} by φ around c_2 and then rotate by θ around c_1 .

After rotating \mathbf{s} by φ around c_2 , we get $\mathbf{s}_\varphi = T_2^{-1} \cdot R_\varphi \cdot T_2(\mathbf{s})$, where $R_\varphi = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{pmatrix}$, $T_2(\mathbf{s}) = \mathbf{s} + (0,0,r_2)'$, $T_2^{-1}(\mathbf{s}) = \mathbf{s} + (0,0,-r_2)'$.

Then we rotate \mathbf{s}_φ by θ around c_1 , resulting in $\mathbf{s}' = T_1^{-1} \cdot R_\theta \cdot T_1(\mathbf{s}_\varphi)$, where $R_\theta = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}$, $T_1(\mathbf{s}) = \mathbf{s} + (0,0,r_1)'$, $T_1^{-1}(\mathbf{s}) = \mathbf{s} + (0,0,-r_1)'$. Therefore

$$\begin{aligned} \mathbf{s}' &= T_1^{-1} \cdot R_\theta \cdot T_1 \cdot T_2^{-1} \cdot R_\varphi \cdot T_2(\mathbf{s}) \\ &= \begin{pmatrix} r_2 \sin \theta \cos \varphi + (r_1 - r_2) \sin \theta \\ -r_2 \sin \varphi \\ r_2 \cos \theta \cos \varphi + (r_1 - r_2) \cos \theta - r_1 \end{pmatrix} \end{aligned}$$

Appendix B. Closed-form solution of Eq. (1)

Consider a point \mathbf{s} and its k one-ring neighbors, denoting by A_i and d_i the area and density of the triangle whose vertices are \mathbf{s} , \mathbf{s}_i and \mathbf{s}_{i+1} (see embedded (Fig. B1)).

$$A_i^2 = |(\mathbf{s}_{i+1} - \mathbf{s}) \times (\mathbf{s}_i - \mathbf{s})|^2$$

Denoting $a_i = x_i - x_{i+1}$, $b_i = y_i - y_{i+1}$, $c_i = z_i - z_{i+1}$, $e_i = x_i y_{i+1} - x_{i+1} y_i$, $f_i = x_i z_{i+1} - x_{i+1} z$, $g_i = y_i z_{i+1} - y_{i+1} z_i$, we have

$$\begin{aligned} A_i^2 &= (b_i^2 + c_i^2)x^2 + (a_i^2 + c_i^2)y^2 + (a_i^2 + b_i^2)z^2 \\ &\quad - 2b_i c_i y z - 2a_i c_i x z - 2a_i b_i + 2(c_i f_i + b_i e_i)x \\ &\quad + 2(c_i g_i - a_i e_i)y + 2(-a_i f_i - b_i g_i)z + e_i^2 + f_i^2 + g_i^2 \end{aligned}$$

Now replace \mathbf{s} by \mathbf{s}' , thus A_i^2 becomes a function of θ and φ . After that, we use a second order Taylor expansion to approximate A_i^2 as a quadratic function of θ and φ at $(0,0)$

$$\begin{aligned} \tilde{A}_i^2 &= \frac{1}{2} [(b_i^2 + c_i^2)r_1^2 - (a_i f_i + b_i g_i)r_1] \theta^2 \\ &\quad + \frac{1}{2} [(a_i^2 + c_i^2)r_2^2 + (a_i f_i + b_i g_i)r_2] \varphi^2 + a_i b_i r_1 r_2 \theta \varphi \\ &\quad + (c_i f_i + b_i e_i)r_1 \theta + (a_i e_i - c_i g_i)r_2 \varphi + e_i^2 + f_i^2 + g_i^2 \end{aligned}$$

For the sum of neighboring squared capacities $\sum_{i=1}^k C_i^2$, there also exists a second order Taylor expansion to approximate it by

$$\sum_{i=1}^k C_i^2 \approx \sum_{i=1}^k \tilde{C}_i^2 = \sum_{i=1}^k d_i \tilde{A}_i^2$$

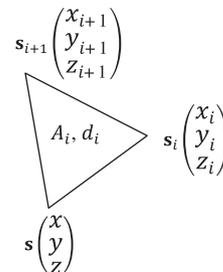


Fig. B1

The second equation is a simple 2×2 linear system for the angles θ and φ that satisfy $C \begin{pmatrix} \theta \\ \varphi \end{pmatrix} = b$, where

$$C(1,1) = \sum_{i=1}^k d_i [(b_i^2 + c_i^2)r_1^2 + (a_i f_i + b_i g_i)r_1]$$

$$C(1,2) = C(2,1) = \sum_{i=1}^k d_i a_i b_i r_1 r_2$$

$$C(2,2) = \sum_{i=1}^k d_i [(a_i^2 + c_i^2)r_2^2 + (a_i f_i + b_i g_i)r_2]$$

$$b(1) = \sum_{i=1}^k -d_i (c_i f_i + b_i e_i) r_1$$

$$b(2) = \sum_{i=1}^k -d_i (a_i e_i - c_i g_i) r_2$$

References

- [1] Amenta N, Bern M, Kamvysselis M. A new Voronoi-based surface reconstruction algorithm. In: Proceedings of the SIGGRAPH; 1998. p. 415–421.
- [2] Alliez P, Meyer M, Desbrun M. Interactive geometry remeshing. *ACM Trans Graphics* 2002;21(3):347–54.
- [3] Bartlett M. An introduction to stochastic processes with special reference to methods and applications. Cambridge University Press; 1955.
- [4] Balzer M, Schlomer T, Deussen O. Capacity-constrained point distributions: a variant of Lloyd's method. *ACM Trans Graphics* 2009;28(3):861–8.
- [5] Bowers J, Wang R, Wei LY, Maletz D. Parallel Poisson disk sampling with spectrum analysis on surfaces. *ACM Trans Graphics* 2010;29(6) (pp. 166:1–166:12).
- [6] Cline D, Jeschke S, Razdan KW, Wonka P. Dart throwing on surfaces. *Comput Graphics Forum* 2009;28(4):1217–26.
- [7] Crow FC. The aliasing problem in computer-generated shaded images. *Commun ACM* 1977;20(11):799–805.
- [8] Cook RL. Stochastic sampling in computer graphics. *ACM Trans Graphics* 1986;5(1):51–72.
- [9] Dunbar D, Humphreys G. A spatial data structure for fast Poisson-disk sample generation. *ACM Trans Graphics* 2006;25(3):503–8.
- [10] Dyer R, Zhang H, Möller T. Delaunay mesh construction. *Comput Graphics Forum* 2007:273–82.
- [11] Ebeida MS, Patney A, Mitchell SA, Davidson AA, Knupp PM, Owens JD. Efficient maximal Poisson-disk sampling. *ACM Trans Graphics* 2011;30:4.
- [12] Fattal R. Blue-noise point sampling using Kernel Density Model. *ACM Trans Graphics* 2011;28(3):1–10.
- [13] Karni Z, Gotsman C. Spectral compression of mesh geometry. *Comput Graphics* 2000:279–86.
- [14] Lagae A, Dutre P. A comparison of methods for generating Poisson disk distributions. *Comput Graphics Forum* 2008;27(1):114–29.
- [15] Liu Y, Wang W, Levy B, Sun F, Yan DM, Lu L, et al. On centroidal Voronoi tessellation—energy smoothness and fast computation. *ACM Trans Graphics* 2009;28(4):1–17.
- [16] Li H, Wei LY, Sander PV, Fu CW. Anisotropic blue noise sampling. *ACM Trans Graphics* 2010;29:6.
- [17] McCool M, Fiume E. Hierarchical Poisson disk sampling distributions. *Proc Graphics Interface* 1992:94–105.
- [18] Pottman H, Hofer M. Geometry of the squared distance function to curves and surfaces. *Visualization Math III* 2003:223–44.
- [19] Rusinkiewicz S. Estimating curvatures and their derivatives on triangle meshes. In: *Symposium on 3D Data Processing, Visualization, and Transmission*; 2004. p. 486–493.
- [20] Turk G. Re-tiling polygonal surfaces. In: *Proceedings of the SIGGRAPH*; 1992. p. 55–64.
- [21] Ulichney R. *Digital halftoning*. MIT Press; 1987.
- [22] White KB, Cline D, Egbert PK. Poisson disk point sets by hierarchical dart throwing. In: *Proceedings of the IEEE Symposium on Interactive Ray Tracing*; 2007. p. 129–132.
- [23] Wei LY. Parallel Poisson disk sampling. *ACM Trans Graphics* 2008;27(3) (pp. 20:1–20:9).
- [24] Wei LY, Wang R. Differential domain analysis for non-uniform sampling. *ACM Trans Graphics* 2011.
- [25] Xu Y, Liu L, Gotsman C, Gortler SJ. Capacity-constrained Delaunay triangulation for point distributions. *Comput Graphics* 2011;35(3):510–6.
- [26] <<http://www.cs.umd.edu/~mount/ANN>>.