Computer-Aided Design 46 (2014) 25–36

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Improving spatial coverage while preserving the blue noise of point sets

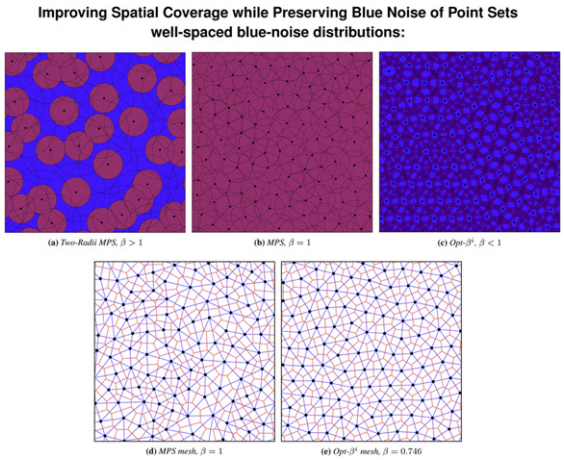
Mohamed S. Ebeida [a,*], Muhammad A. Awad [b], Xiaoyin Ge [c], Ahmed H. Mahmoud [b], Scott A. Mitchell [a], Patrick M. Knupp [a], Li-Yi Wei [d]

[a] Computing Research, Sandia National Laboratories, United States
[b] Alexandria University, Egypt
[c] Computer Science and Engineering, Ohio State University, United States
[d] Computer Science, The University of Hong Kong, Hong Kong

HIGHLIGHTS

- Local smoothing to optimize Voronoi cell aspect ratios.
- Simultaneously achieve random and well-spaced points.
- Image filtering applications.
- Meshing applications.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Optimization
Voronoi aspect ratio
Disk
Triangulation

ABSTRACT

We explore the notion of a Well-spaced Blue-noise Distribution (WBD) of points, which combines two desirable properties. First, the point distribution is random, as measured by its spectrum having blue noise. Second, it is well-spaced in the sense that the minimum separation distance between samples is large compared to the maximum coverage distance between a domain point and a sample, i.e. its Voronoi cell aspect ratios $2\beta^i$ are small. It is well known that maximizing one of these properties destroys the other: uniform random points have no aspect ratio bound, and the vertices of an equilateral triangular tiling have no randomness. However, we show that there is a lot of room in the middle to get good values for both. Maximal Poisson-disk sampling provides $\beta = 1$ and blue noise. We show that a standard optimization technique can improve the well-spacedness while preserving randomness.

Given a random point set, our Opt-β^i algorithm iterates over the points, and for each point locally optimizes its Voronoi cell aspect ratio $2\beta^i$. It can improve β^i to a large fraction of the theoretical bound given by a structured tiling: improving from 1.0 to around 0.8, about half-way to 0.58, while

* Corresponding author.
E-mail address: msebeid@sandia.gov (M.S. Ebeida).
URL: http://msebeida.net/ (M.S. Ebeida).

0010-4485/$ – see front matter. Published by Elsevier Ltd
http://dx.doi.org/10.1016/j.cad.2013.08.015

preserving most of the randomness of the original set. In terms of both β and randomness, the output of Opt-β^i compares favorably to alternative point improvement techniques, such as centroidal Voronoi tessellation with a constant density function, which do not target β directly. We demonstrate the usefulness of our output through meshing and filtering applications. An open problem is constructing from scratch a WBD distribution with a guarantee of $\beta < 1$.

Published by Elsevier Ltd

1. Introduction

Many applications desire a distribution of points that are not too close to one another, yet are evenly distributed throughout the domain. Such points are *well-spaced*, or *separated yet dense*. The separation distance improves efficiency; close points often add little information, yet they consume time and memory. Well-spaced points reduce the interpolation and simulation error in scientific applications. Points spread uniformly can help reduce noise or variance. Graphics applications often desire randomness to help reduce aliasing or bias. Blue noise refers to distributions that are roughly uniform random with no preferred inter-point directions or distances. Note that well-spacedness is a local measure of nearby points. In contrast, blue noise is a global measure dependent on the distances between far points. Some methods for well-spaced points often do not produce blue noise: e.g. periodic tilings such as lattices and structured meshes; raw low discrepancy sequences; and Delaunay Refinement (DR).

In the 1990s, the fracture mechanics community studied the effects of mesh structure in finite element fracture simulations where the crack directions are limited to mesh edges [1]. They concluded that uniform random points lead to uniform random edge orientations, and more physically realistic simulations. Both Delaunay and dual Voronoi elements are of interest. The initial techniques for generating such meshes included simple jittering, perturbing the position of points. Interest in the topic recently revived, and researchers have developed more sophisticated techniques, including random sampling [2–6]. Computer graphics has a long-standing interest in techniques for blue-noise distributions, for applications such as rendering [7,8], animation [9], modeling [10,11], and imaging [12,13]. Graphics provides standardized methods for determining if a distribution has blue noise [14].

The well-spacedness of a distribution can be characterized by two measures: *conflict*, which is about the spacing between samples, and *coverage*, which is about how samples collectively cover the underlying domain. The conflict distance r_f is the minimum distance between two samples, the coverage distance r_c is the maximum distance between a domain point and a sample, and $\beta = r_c/r_f$ is our spacing measure. The allowed (or target) values in an algorithm and the achieved values in its output are different concepts. In typical algorithms, these distances are enforced a priori by placing disks around sample points. On output, sample points might be farther apart than the enforced minimum, and domain points might be closer to a sample than the enforced maximum. See Figs. 1–3 for illustrations and Section 4 for formulations.

Different methods achieve various degrees of conflict and coverage. Dart throwing [7] enforces r_f, and attempts to have output r_c close to it, but usually falls short. Maximal Poisson-disk Sampling (MPS) [16] enforces both and has $r_f = r_c$. Two-radii MPS [15] also enforces both, but encourages more randomness (and less uniformity) by using $r_c > r_f$. In practice, for even a few thousand points, MPS output r_f is just slightly larger than the enforced r_f, and output r_c is just slightly smaller than the enforced r_f, because a local case similar to Fig. 3(b) or (c) is very likely. That is, MPS guarantees output $\beta \leq 1$, and usually achieves β very close to 1. Two-radii MPS guarantees output $\beta \leq$ target β, and typically achieves something close to it.

There is some correlation between coverage/conflict and randomness, but the two are not proportional. At one extreme, uniform random points (also known as white noise or Poisson Sampling PS without disks) has unbounded coverage and conflict. Spectrum plots of MPS and two-radii distributions [15] and Opt-β^i (Fig. 9) show a strong correlation between larger achieved β and more randomness.

However, this is not the full story. In two dimensions, the vertices of equilateral triangle tiling achieve the minimum possible β, with $r_c = r_f/\sqrt{3}$. It is the only distribution achieving this value, and distributions coming close to this value bear some resemblance to it. The vertices of its dual, hexagonal tiling, have $r_c = r_f$. Square tiling gives a value of $1/\sqrt{2}$, somewhere in between. By adding or moving a single point of any of these, one can generate a periodic tiling with any larger β desired, $\infty > \beta > 1/\sqrt{3}$. Delaunay Refinement DR [17,18] is a deterministic process for generating a triangulation. Its output vertices achieve $r_c \leq r_f$, and its spectra show features of both regularity and MPS. None of the tilings or algorithms mentioned in this paragraph are random. However, a random algorithm such as MPS or even PS could in principle produce one of these, albeit with very small probability. The possible β achieved by different distribution types are shown in Fig. 2.

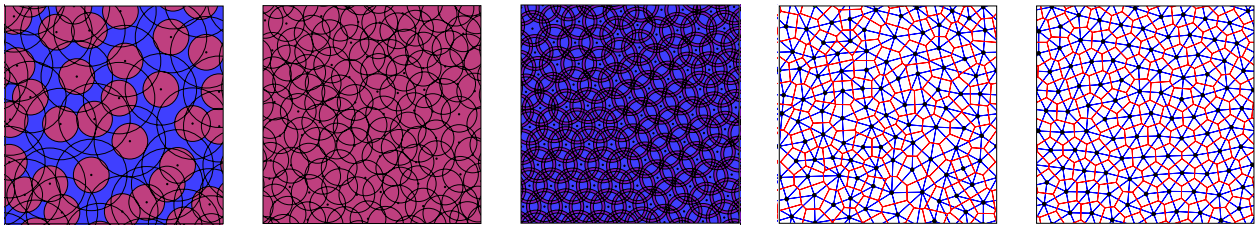
Any random sampling method that only considers local criteria when placing points, and never moves them, cannot be guaranteed to achieve any target value of $\beta_t < 1$. The alternatives are to consider global constraints, or to adjust point locations, as Opt-β^i does in this paper. The reason that local decisions and fixed locations are insufficient is that it is easy to "paint yourself into a corner", meaning that selecting points independently can create a subregion that is impossible to cover without placing points too close together. This can easily happen for any value of $\beta < 1$; see Fig. 3. Achieving $\beta < 1$ is actually more challenging than achieving $\beta = 1$ [16] or $\beta > 1$ [15].

To summarize,

- $\beta > 1$ is produced by two-radii MPS [15], and the user may select β. Uniform sampling without disks produces white noise, which may have β approaching ∞.
- $\beta = 1$ is produced by traditional maximal Poisson-disk sampling [16]. Stopping short of maximality produces $\beta > 1$.
- $\beta < 1$ is produced by Opt-β^i and it tends to make the distribution more uniform. Incremental insertion rarely achieves β significantly below 1.
- The smaller the β, the more uniform a point set tends to be, but this is a feature of both the random process generating the points and β. In two dimensions, $\beta = 1/\sqrt{3}$ indicates a periodic equilateral triangle tiling. A periodic structure can be tuned to any larger value of β. The minimum achievable β is dimension dependent, but $\beta \geq 1/\sqrt{3}$ in two dimensions, and $\beta \geq 0.5$ by definition.

2. Spacing measures in other contexts

There are prior formulations quantifying both conflict and coverage, such as minimal spacing r_f in dart throwing [7], normalized spacing ρ [19], and coverage radius r_c [15]. Discrepancy measures spatial uniformity by comparing the fraction of points in any box



(a) Two-radii MPS, $\beta > 1$. (b) MPS, $\beta = 1$. (c) Opt-β^i, $\beta < 1$. (d) MPS mesh, $\beta = 1$. (e) Opt-β^i mesh, $\beta = 0.746$.

Fig. 1. Maximal Poisson-disk Sampling (MPS) with different uniformity as measured by $\beta = \frac{r_c}{r_f}$. In Poisson-disk sampling, each black sample point must be outside all other points' red conflict disks with radii r_f. A distribution is maximal if the blue coverage disks with radii r_c cover the entire domain. Differentiating the two types of disk results in three sampling scenarios. (a) Two-radii MPS [15], where $r_f < r_c$. The distribution is less uniform and more random than MPS. (b) Maximal Poisson-disk Sampling (MPS) [16], where $r_f = r_c$. (c) Opt-β^i, with $r_f > r_c$; this work. Opt-β^i is more uniform and less random than MPS, improving the well-spacedness while retaining most of the blue-noise properties, as seen in the meshes in (d) and (e). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

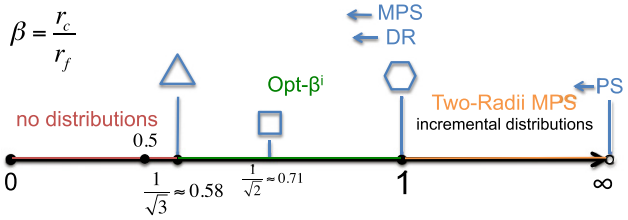
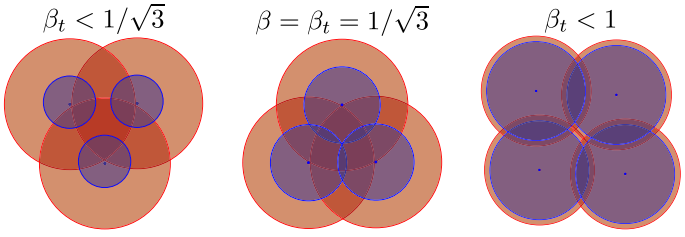


Fig. 2. Distributions by aspect ratio β, in two dimensions. Here, △, □, and ⬡ are the vertices of the equilateral triangle, square, and regular hexagonal tilings of the plane. PS is Poisson sampling without conflict disks. DR is Delaunay Refinement. MPS is Maximal Poisson-disk Sampling. No point distribution can achieve $\beta < 1/\sqrt{3}$ in two dimensions, or $\beta < 0.5$ in any dimension.

to the fraction of the domain (volume) in that box [20]. In spatial statistics, in the hard-core Strauss disk process, r_f is known as the inhibition radius and r_c is known as the coverage radius. In biological system analysis, e.g. forestry image analysis [21], one usually observes a (non-maximal) point distribution that is assumed to come from this process, then seeks to estimate the hidden generating parameters.

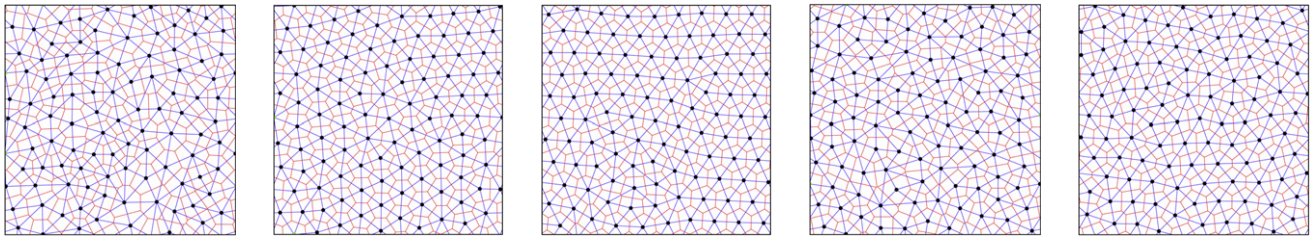
In computational geometry, r_f^i is twice the inradius of the ith Voronoi cell. And r_c^i is known as the outradius of the Voronoi cell, the distance from a seed point to the furthest Voronoi vertex of its cell. The outradius/inradius ratio is the Voronoi cell aspect ratio, $A = 2\beta^i$. Since outradius $\geq$ inradius, $A \geq 1$ and $\beta^i \geq 0.5$ in any dimension. A point set with maximum aspect ratio A is called an A-well-spaced set [22]. Much effort [23,24] has been devoted to generating A-well-spaced sets, for both uniform and spatially varying densities. Usually the upper bound on A is greater than 2, but we strive for $A < 2$.

In graphics, a common measure of point distributions is the relative radius ρ [19]. Its definition starts with non-overlapping disks with constant $r_f/2$ radius. The packing density η is the fraction of the domain area covered by these disks; η is itself a popular measure in physics. Given a unit-area periodic domain containing N disks, $r_{\max}$ is the radius achieved by the densest (in terms of η) known arrangement, assuming that N divides evenly into that tiling. In two dimensions, $r_{\max} = \text{const}/\sqrt{N}$. We define $\rho = r_f/(2r_{\max})$, where $\rho \in [0, 1]$. Thus ρ is directly related to r_f, but only indirectly related to r_c through N. For example, if we create



(a) No solution. (b) One solution. (c) Painted into a corner.

Fig. 3. Trying to achieve target ratio β_t using fixed r_c and r_f disks. In a solution the red r_f conflict disks must be free of other samples, and the blue r_c coverage disks must cover the entire domain. In (a), any new blue disk centered outside the red disks would be too small to cover the point equidistant from the three extant samples. In (b), there is only one solution: vertices forming equilateral triangles. In (c), algorithms that incrementally add points and never move them can easily get stuck even if β_t is very close to 1. The red center is too isolated to be coverable by blue disks, just as in (a). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) MPS, $\beta = 1.0$. (b) CVT, $\beta = 0.827$. (c) DistMesh, $\beta = 0.873$. (d) Far-Point, $\beta = 0.965$. (e) Opt-β^i, $\beta = 0.746$.

Fig. 4. Final mesh for a periodic unit box after applying various methods to the input MPS in (a). While CVT and DistMesh improve the quality of the majority of the Voronoi cells, they tend to lose randomness at larger values of β. Far-Point on the other hand tends to violate the coverage condition. None of the four methods were able to achieve $\beta < 0.7$ in general.

a gap by removing one vertex (disk) from an equilateral triangle tiling, then ρ decreases by a factor of $\sqrt{N-1}/\sqrt{N}$, a negligible change if N is large, but r_c and β almost double from $1/\sqrt{3}$ to 1. In our context, the number of points N is fixed, so we have $\beta \propto r_c/\rho$. Area coverage and $r_{\max}$ are global averages over the disks, whereas r_c^i is a maximum achieved in a local neighborhood of a point, its Voronoi cell. We think that β^i is a better measure of local spacing than ρ, because β measures local gaps and ρ does not.

3. Comparison methods

We compare our method against three alternative sampling techniques. While they all strive to improve the spatial quality of a distribution, none optimize β^i directly as we do.

3.1. Centroidal Voronoi tessellation (CVT)

A CVT [25] is defined as a distribution in which each point is at the center of mass of its Voronoi cell. The earliest technique to achieve a CVT is Lloyd's iteration [26], iteratively moving each point to the center of its Voronoi cell. Recently, Liu et al. [27] improved the speed of converging to a CVT by recognizing the second-order smoothness of its energy function, and optimizing it using a quasi-Newton method. But the converged solution is the same. The CVT converges to an equilateral triangle tiling, absent boundary effects. For this reason, many applications stop short of convergence.

Based on the iterative method definition, one might expect CVT to improve r_c^i, and not degrade r_f^i, at least in the average sense. Our experience is that CVT tends to decrease the maximum coverage radius r_c, but does not provide any control over the minimum disk-free radius r_f.

3.2. DistMesh

DistMesh is a popular point relocation technique that aims to make the Delaunay edge length distribution match a user-specified sizing function [28]. This technique simply treats each Delaunay edge as a spring that can expand or contract; but if its length is less than the user-specified sizing function it can only expand. For the uniform case, the sizing function is constant and is usually chosen to be 20% more than the desired edge length. While this technique is good for improving the quality of the majority of the elements, it is not as good for improving the worst-quality elements.

In contrast to CVT, DistMesh tends to increase the disk-free radius r_f of an MPS; however, it does not provide any control over the coverage radius r_c.

3.3. Farthest neighbor (Far-Point)

Farthest neighbor [29,30] is an iterative technique that moves each point to the Voronoi vertex farthest from its immediate neighbors. As such, it locally optimizes r_f^i, and ignores r_c^i. The algorithm works by finding the Delaunay neighbors $\mathcal{N} = \{x^j\}$ of sample x^i, computing the Voronoi vertices of $\mathcal{N}$ (without x^i), then reinserting x^i at the Voronoi vertex with the maximum minimum distance to $\mathcal{N}$. If the points are not in convex position, there will be some Voronoi vertices far from the convex hull of $\mathcal{N}$; these vertices are ignored.

This technique tends to increase all Delaunay edge lengths, not only the short ones.

3.4. Our method, Opt-β^i

We propose Opt-β^i, local optimization of the β^i of the initial MPS sample positions, stopping when a user-specified ratio $\beta_t = r_c/r_f$ is achieved. By tuning β, our method can produce sample distributions with a level of uniformity and randomness as desired by the users. There is no guarantee that our method can achieve any value of $\beta < 1$, or even that β will not increase, but in practice we can obtain $\beta \approx 0.75$.

Table 1
Statistics of meshes in Fig. 10 mesh; α denotes triangle angles.

Algorithm	β	$\frac{r_c}{r_{\text{MPS}}}$	$\frac{r_f}{r_{\text{MPS}}}$	$\min\alpha$	$\max\alpha$
(a) Coarse mesh, $r_{\text{MPS}} = 0.0314$, Fig. 10 left column.					
MPS	1.0	1.0	1.0	31	115
CVT	1.226	0.931	0.759	24	103
DistMesh	1.089	1.012	0.929	31	114
Far-Point	1.048	1.043	0.996	32	106
Opt-β^i	0.988	0.995	1.007	32	110
(b) Fine mesh, $r_{\text{MPS}} = 0.0157$, Fig. 10 right column.					
MPS	1.0	1.0	1.0	30	117
CVT	1.02	0.989	0.852	33	96
DistMesh	1.07	0.869	0.925	34	107
Far-Point	1.06	1.106	1.047	31	113
Opt-β^i	0.932	0.939	1.008	34	99

In the sampling methods of Section 1, at least one of r_c or r_f is a fixed enforced value. Opt-β^i is different, in that these are both measured quantities of the output. We simultaneously adjust both r_c and r_f in pursuit of optimizing their ratio β. Moving points might increase r_f or decrease r_c. Experiments show that, despite this possibility, we usually simultaneously achieve both an increased r_f and a reduced r_c, while maintaining good spectra; see Table 1 and Fig. 5. Fig. 4 compares the methods over a periodic square.

Our paper has the following main contributions.

- Introducing and formalizing the notion of Well-spaced Blue-noise Distributions (WBDs).
- Proposing an algorithm, Opt-β^i, to generate a WBD with controllable uniformity and randomness through β_t.
- Demonstrating the benefits of a WBD and Opt-β^i through applications in geometry meshing and image filtering.

4. Formal definitions

Conflict radius r_f. The minimum achieved spacing between any pair of samples in the set $\mathcal{S}$. For example, for MPS, at least the enforced Poisson-disk radius.

Coverage radius r_c. Any point in the domain Ω will be at most r_c away from the nearest sample in $\mathcal{S}$. Equivalently, the set of spheres with radius r_c will cover the entire domain.

Ratio β. The ratio between conflict and coverage is measured by $\beta = r_c/r_f$. This is the ultimate objective of our method.

Initial radii. The input (MPS) point set has $r_{\text{MPS}} = r_f = r_c$.

During the local optimization procedure, we only consider the local values of these quantities, i.e.

Adjusted point x^i. We move one point at a time, $x^i \in \mathcal{S}$.

Conflict radius r_f^i. Distance from x^i to its nearest $x^j \in \mathcal{S}$.

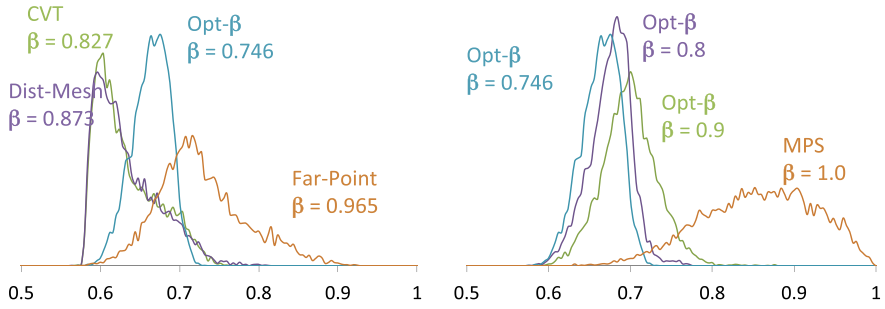
Coverage radius r_c^i. Distance from x^i to the farthest vertex of its Voronoi cell, i.e. the distance to the domain point farthest from x^i that is not closer to some other sample x^j.

Local ratio β^i. This is r_c^i/r_f^i. Note that

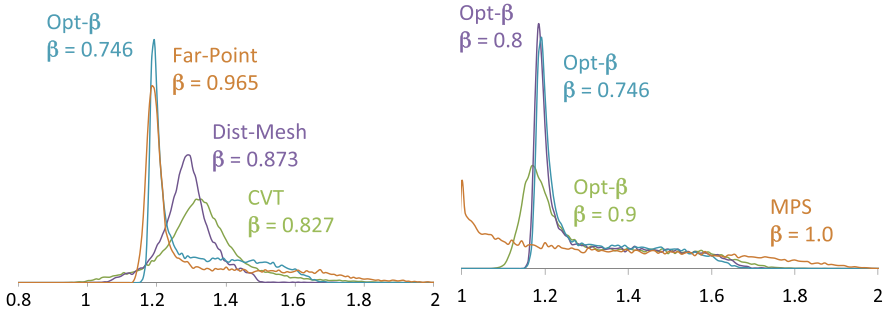
$$\beta = \frac{\max_i r_c^i}{\min_j r_f^j} \geq \max_i(\beta^i) = \max_i\left(\frac{r_c^i}{r_f^i}\right). \tag{1}$$

5. Opt-β^i algorithm

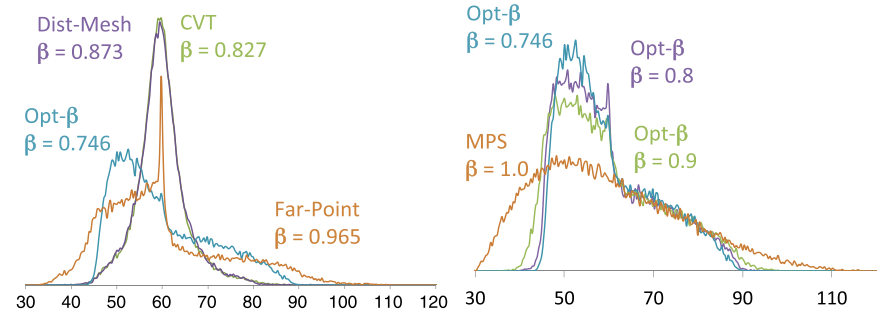
We now describe our algorithm in detail. We start with an MPS, or some other distribution with blue noise, with $\beta = 1$. (Recall that



(a) Voronoi cell aspect ratio.



(b) Delaunay edge lengths.



(c) Delaunay angles.

Fig. 5. Distribution of quality measures. The left column compares the different methods, and the right column compares Opt-β^i for different β.

the MPS output might have $\beta < 1$, but in practice β is so close to 1 that we just say that $\beta = 1$.) It is possible to start with other distributions, such as the output of DR, but, since our method removes noise, the output will be limited by the noise in the input. In principle it is also possible to start with a distribution with $\beta > 1$; in this paper we assume that such distributions are enriched to maximality with an MPS algorithm so that $\beta = 1$.

The user specifies the target $\beta = \beta_t$, the desired trade-off between spacing and randomness. For two dimensions, $\beta_t \in [\frac{1}{\sqrt{3}}, 1]$.

Let $\beta^i = r_c^i / r_f^i$ denote the local coverage/conflict ratio. Ideally, one might wish to simultaneously adjust the location of all points to find a global minimum for β, or at least to minimize the maximum of β^i over all x^i. A more practical method is to iteratively move each x^i towards its local optimum [31].

We iteratively reposition each sample point x^i to locally minimize its β^i, until globally $\beta < \beta_t$.

To reposition x^i, we perform ten iterations of Nelder–Mead [32]. Nelder–Mead is a downhill-traveling optimization heuristic that requires only function values, not derivatives. Its state consists of three candidate points $c_{\{1,2,3\}}$, and the β^i function values if x^i was repositioned at those locations. To get started, we select the first candidate point c_1 at x^i, the second c_2 at x^i shifted in the x-axis direction by $r_{\text{MPS}}/10$, and the third c_3 at x^i shifted in the y-axis direction by $r_{\text{MPS}}/10$. These $c_{\{1,2,3\}}$ define a triangle, tilted in three dimensions by assigning the respective height β^i to its corners. Nelder–Mead replaces the high corner by some point on the other

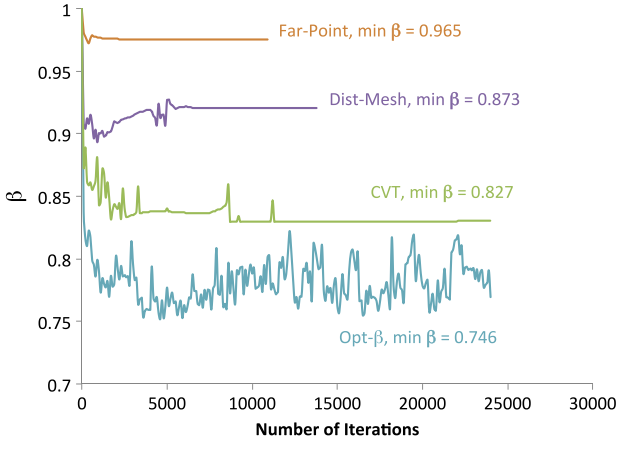


Fig. 6. Variation of β as the iteration proceeds.

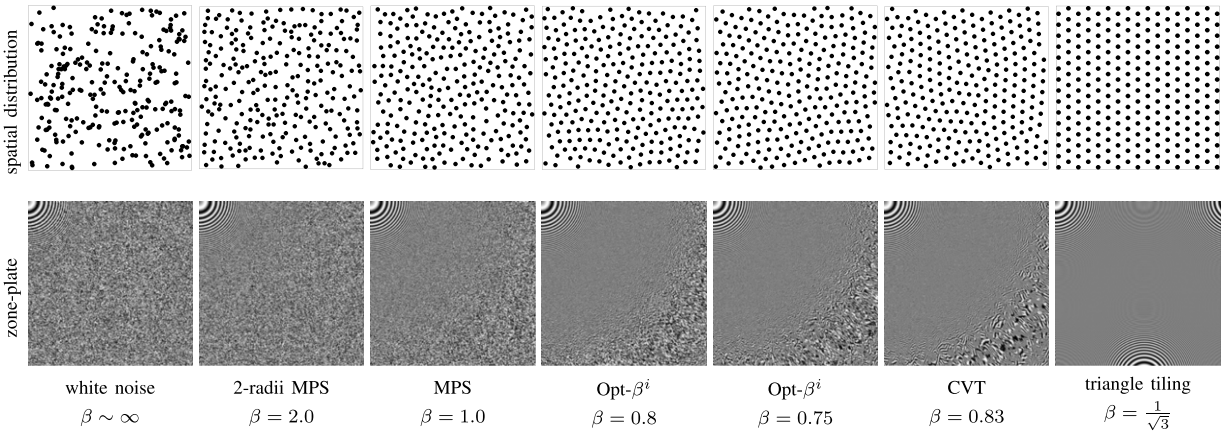


Fig. 7. Spatial analysis via zone-plate sampling. Each uses about 122,500 samples (about 1 sample per pixel) and a 3-pixel-wide Gaussian kernel.

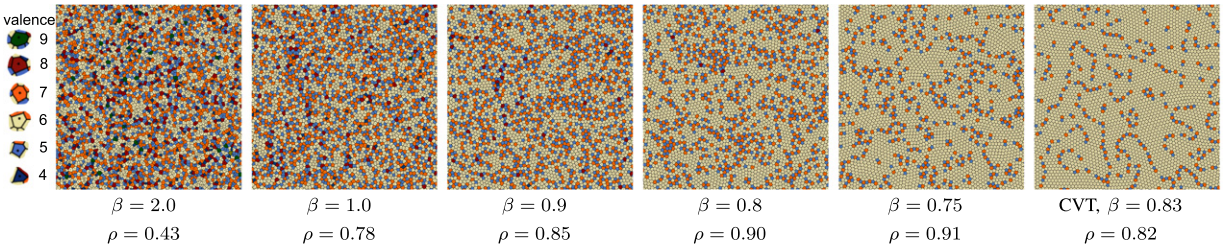
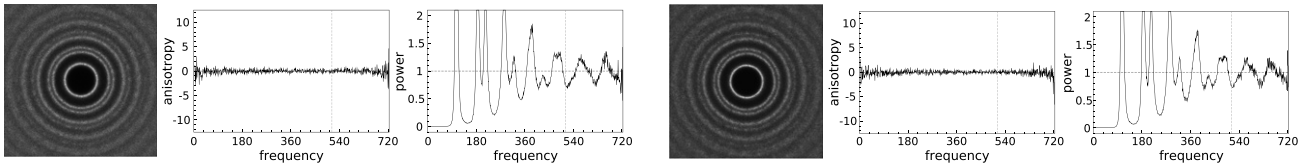
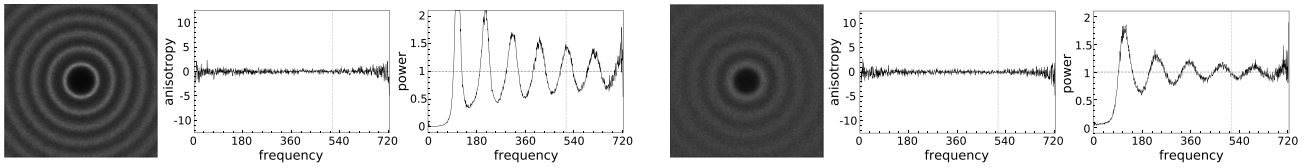


Fig. 8. Voronoi cell valence and ρ for different β achieved by Opt-β^i and CVT. Recall that $\beta \propto r_c/\rho$. See also Fig. 8 in "Blue Noise…" [38].



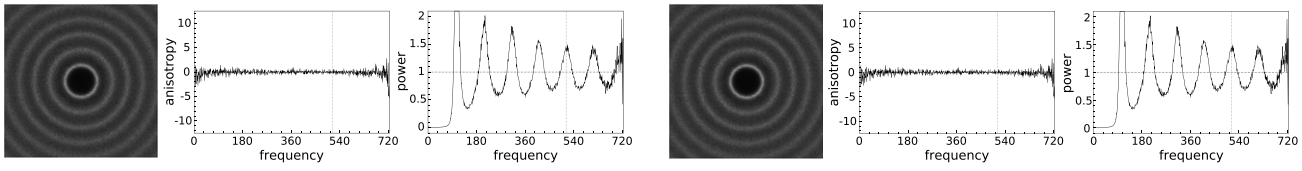
(a) CVT, $\beta = 0.827$.

(b) DistMesh, $\beta = 0.873$.



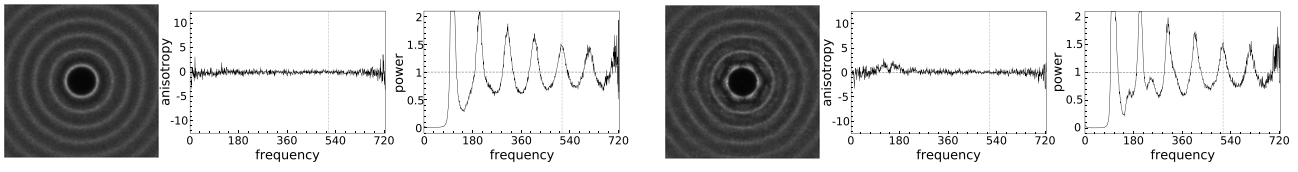
(c) Far-Point, $\beta = 0.965$.

(d) MPS, $\beta = 1.0$.



(e) Opt-β^i, $\beta = 0.85$.

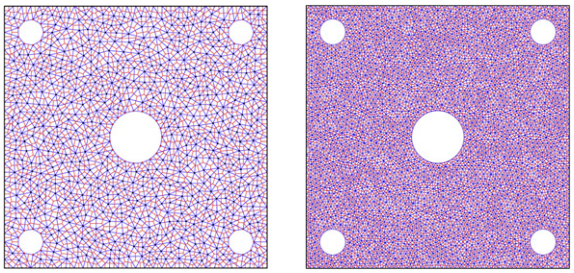
(f) Opt-β^i, $\beta = 0.80$.



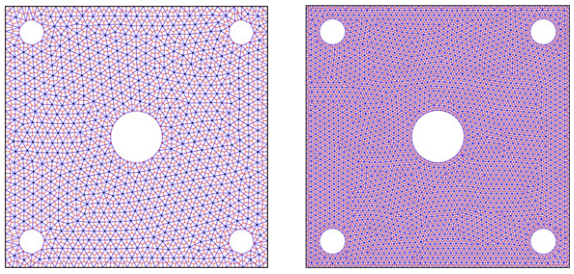
(g) Opt-β^i, $\beta = 0.75$.

(h) Opt-β^i, $\beta = 0.70$.

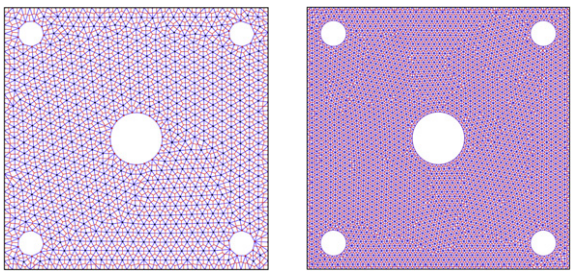
Fig. 9. Spectral analysis. Shown here are the Fourier spectrum analyses of various sample patterns. Each case contains ∼10k samples in a unit box.



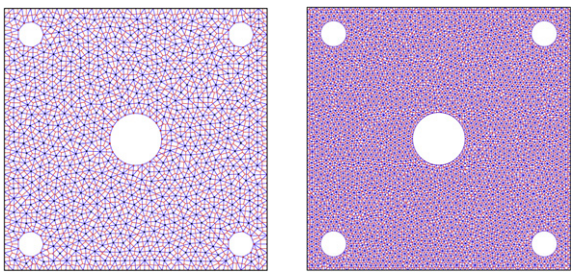
(a) MPS, $\beta = 1.0$. (b) MPS, $\beta = 1.0$.



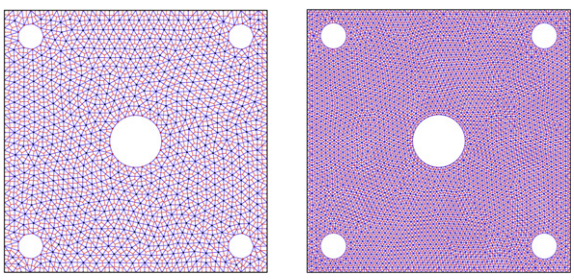
(c) CVT, best $\beta = 1.226$. (d) CVT, best $\beta = 1.02$.



(e) DistMesh, $\beta = 1.089$. (f) DistMesh, $\beta = 1.07$.



(g) Far-Point, $\beta = 1.048$. (h) Far-Point, $\beta = 1.06$.



(i) Opt-β^i, $\beta = 0.988$. (j) Opt-β^i, $\beta = 0.932$.

Fig. 10. Applying various methods to a coarse (left) and fine (right) mesh of a non-convex two-dimensional domain. For the finer mesh, the boundary effects are less constraining, and a smaller β was achieved.

side of its opposite triangle edge, and recomputes a new triangle and its tilt. We flip-flop through ten triangles and stop.

Each candidate may have different Delaunay neighbors than the original point; we calculate β^i using the candidate's Delaunay neighbors, rather than the original points' neighbors. Sometimes, especially in initial iterations, x^i moves outside of the convex hull of its initial Delaunay neighbors: r_c^i decreases as x^i moves towards the convex hull, but it may happen that r_f^i increases more rapidly, yielding a local improvement in β^i. We experimented with preventing this from happening, but found it was faster to allow this to happen.

Often, improving β^i causes a nearby point's β^j to get worse. This is a common phenomenon in mesh smoothing algorithms, and the common approach is to allow this to happen, and attempt to remedy it later during the optimization of x^j.

We sweep over all x^i, locally optimizing and updating their positions. Each update happens immediately. That is, for a neighbor x^j of x^i, we use the updated position of x^i rather than its initial position at the start of the sweep. Our input comes from Simple MPS [33]. The output of that algorithm provides a nice ordering to the points, by scan lines. Simple MPS divides the domain into boxes of side length r_{MPS}, and each box has at most one point. On output the points are lexicographically ordered by the box they lie in, first by row then by column. This is the order the points are visited during our Opt-β^i sweep, regardless of where they later migrate to. This is better than considering points in random order.

In each sweep, we optimize the position of all x^i, even if their local β^i is already small. We tried iterating over just the large-β^i points, but this tended to get us stuck in a local minimum.

For large targets, e.g., $\beta_t > 0.9$, we often get far below it in one or two iterations. To prevent that, we use a damping factor, moving x^i only about half-way (0.6–0.8) of the distance from its initial to its Nelder–Mead optimized position.

Local patch smoothing is common for unstructured meshes [34]. Unlike patch smoothing, the connectivity of the implied mesh, the set of Delaunay neighbors of each sample, changes throughout our algorithm, and in our comparison methods.

6. Analysis

6.1. Angle and edge bounds

Here, we recall the relationship between $\beta = r_c/r_f$ and the edge lengths $|e|$, empty circumcircle radius R, and angles α in a Delaunay Triangulation (DT) of the point set. The relationships are well known, and they form the basis for the Delaunay refinement [17] family of algorithms. We restate the succinct summary from Mitchell et al. [15]:

Proposition 6.1. $|e| \in [r_f, 2R]$ *and* $R \leq r_c$.

Proposition 6.2. $\sin\alpha \geq |e|/2R$.

Normalizing by r_f, and noting that the largest angle in a triangle is the supplement of the two smaller ones, we get the following.

Proposition 6.3. $|e|/r_f \in [1, 2\beta]$.

Proposition 6.4. $\alpha \in [\arcsin 1/(2\beta),\ 180^\circ - 2\arcsin 1/(2\beta)]$.

For example, $\beta < 0.75$ gives $|e|/r_f \in [1, 1.5]$ and $\alpha \in [41.8^\circ, 96.4^\circ]$. A value of $\beta < 1/\sqrt{2} \approx 0.71$ provides a non-obtuse triangulation.

These hold locally; i.e., if the local value of β^i for x^i is smaller, it provides tighter bounds for the edges and triangles of x^i.

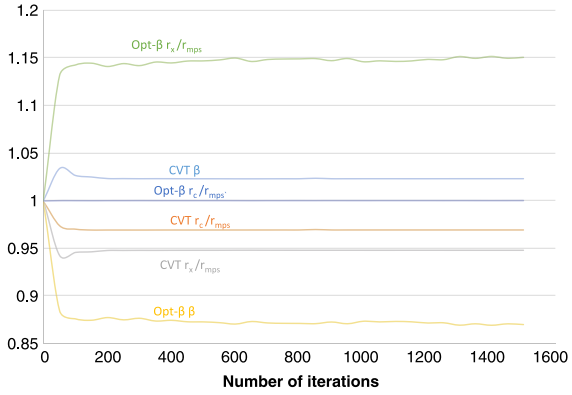


Fig. 11. Typical progress in β, r_c/r_{MPS} and r_f/r_{MPS} for CVT and Opt-β^i.

6.2. β ratio

The main result is that Opt-β^i generates the smallest β values and the best blue noise (the most random spectra). We get a smaller β simultaneously with a better spectrum. Opt-β^i is the best, followed by CVT, DistMesh, and then Far-Point.

Unfortunately, the speed of *one iteration* roughly reverses this order: DistMesh is the fastest, followed by Lloyd's iteration (CVT without Liu et al.'s faster optimization [27]), Far-Point, and then Opt-β^i. *One iteration* of Opt-β^i is 15× slower than one iteration of standard CVT. For all methods, the overall runtime is roughly linear in the number of iterations. However, Opt-β^i is competitive with respect to the overall runtime. For 10,000 points in the periodic cube, to reach $\beta = 0.85$, Opt-β^i took 3.3 s and CVT took 10 s, because CVT required more iterations.

Since we only locally optimize the local $\beta^i = r_c^i/r_f^i$, a somewhat surprising result is that the global $r_f = \min_i r_f^i$ is actually improved above r_{MPS}. Since we improved β, this implies that we have also reduced the maximum r_c^i. These features can be seen from the third and fourth columns of Table 1.

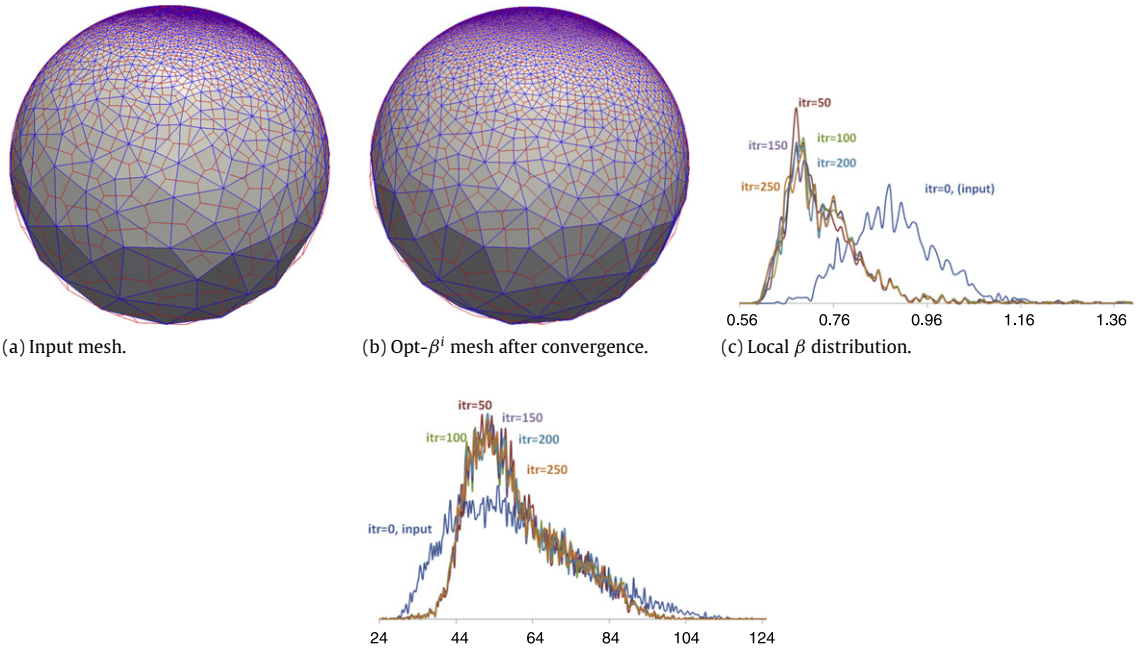
Each β^i is smaller than β, so how small β can be and still have randomness depends on the sample size. A larger sample size is more likely to have a local case stuck at a larger value of β. One alternative would be to consider average β^i.

6.3. Spatial and spectral analysis

Figs. 7 and 9 analyze the Opt-β^i output for different β values, over the periodic unit box domain. We use common spatial and spectral measures [35,19,36,13,37,38]. For spatial analysis in Fig. 7, the zone-plate patterns are input, and we produce Gaussian-filtered output via our sample sets. Since the pattern contains a variety of spatial frequencies in different directions, it is a good stress test to detect any sampling pattern anomalies. We also plot the spatial samples directly for visual inspection. Opt-β^i exhibits the classic noise/alias trade-off: the smaller the β, the more uniform the distribution, and thus there is less noise but more aliasing. Fig. 9 shows the Fourier spectra, radial mean, and anisotropy. We start to lose blue noise between $\beta = 0.75$ and $\beta = 0.7$. The noise/alias trade-off does not apply across methods: for example, even though CVT has a higher β than our Opt-β^i $\beta = 0.8$ case, it still produces more aliasing.

6.4. Convergence effect of small β_t

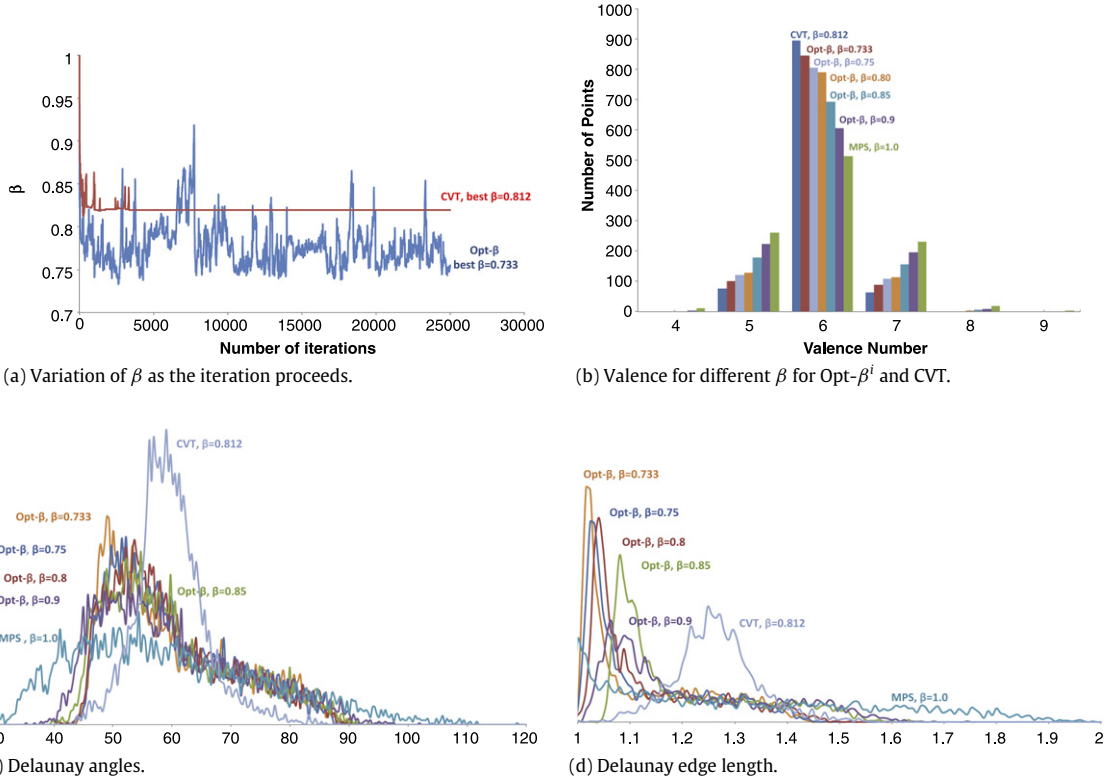
Our experience is that, for values of β_t above about $1/\sqrt{2} \approx 0.71$, the repositioning problem is mostly a geometry problem. For most local configurations, changing point positions to achieve this



(a) Input mesh. (b) Opt-β^i mesh after convergence. (c) Local β distribution.

(d) Delaunay angle distribution.

Fig. 12. Applying Opt-β^i on non-uniform surface mesh with linear sizing function in the z-direction. The angles distribution and local β distribution are shown for number of iterations till convergence (or stagnation) occurs.



(a) Variation of β as the iteration proceeds. (b) Valence for different β for Opt-β^i and CVT.

(c) Delaunay angles. (d) Delaunay edge length.

Fig. 13. β by iteration, and qualities by β, for Opt-β^i and CVT. The input was 1000 points on a sphere.

β_t is eventually possible. Below about 0.71, we see oscillatory behavior in Fig. 6, and the problem becomes a discrete configuration problem. To achieve $\beta_t = 1/\sqrt{3}$, each point must have exactly six neighbors, evenly spaced on a circle around it. The six-neighbor pattern appears at about $\beta = 0.7$ in the hexagonal shape of the rings in the spectrum, bottom left of Fig. 9. Even with five or seven neighbors, it is difficult to obtain a β close to $1/\sqrt{3}$. In the rightmost Opt-β^i example in Fig. 8, with $\beta = 0.75$, all vertices have valences between 5 and 7.

Besides this local discrete configuration constraint, small β_t introduces global geometric constraints. The equilateral triangle tiling has points aligned along three sets of parallel lines. These lines constrain points that are far from one another. This is another reason for slow (or no) convergence for small β_t. Fig. 11

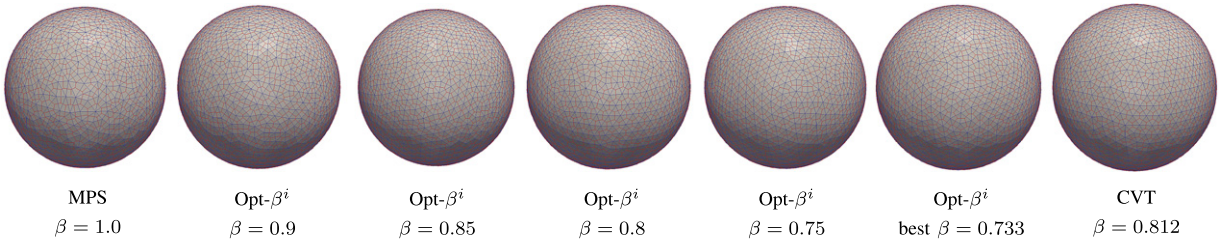
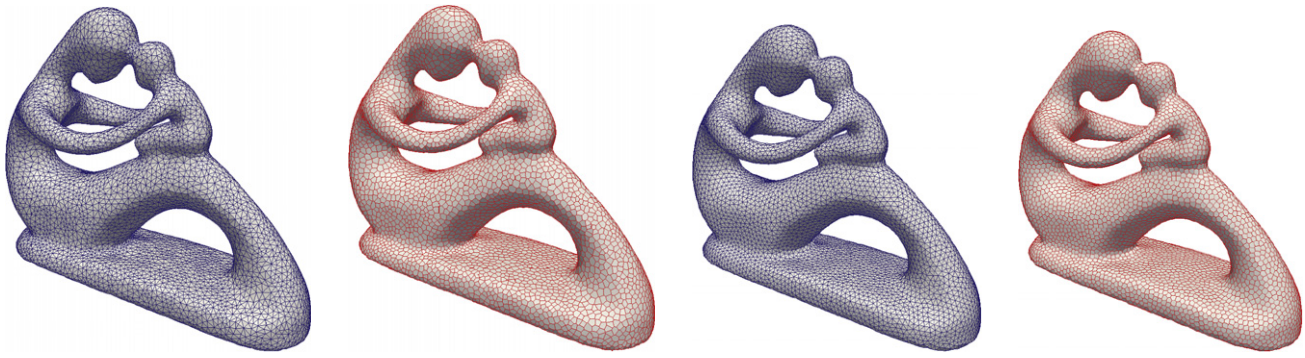
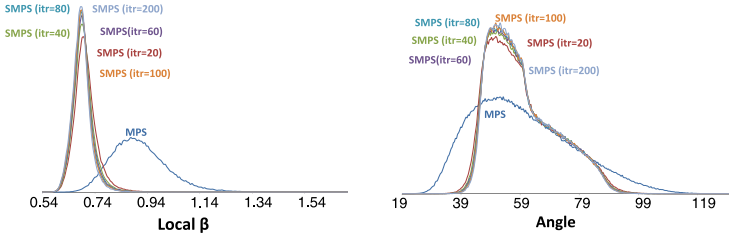


Fig. 14. Meshing for various β values over a sphere surface using 1000 samples.



(a) Delaunay input mesh. (b) Voronoi input mesh. (c) Delaunay after convergence. (d) Voronoi after convergence.



(e) Local β distribution. (f) Delaunay angle distribution.

Fig. 15. Applying Opt-β^i on the Fertility model with non-uniform sizing function.

shows that little progress is made beyond a certain number of iterations.

We perform ten Nelder–Mead steps for each local optimization. Using seven or fewer steps causes the overall algorithm to get stuck at a larger value of β. Using more steps requires more runtime; ten steps ensures both convergence and efficiency.

7. Applications

7.0.1. Meshing planar non-convex domains

We generated Well-spaced Blue-noise Distributions (WBDs) for the bounded non-convex domain in Fig. 10. This is more challenging than for a periodic square, and all four methods suffered from boundary constraints. All methods tended to improve the quality of most of the Voronoi cells. However, CVT and DistMesh violated the disk-free condition. On the other hand, Far-Point violated the coverage condition. Only Opt-β^i was able to reduce the value of β while preserving both properties. The other methods increased β over unity within their first few iterations. The problem is harder for the coarse mesh, because the boundary has an increased effect. Statistics after convergence (or more precisely, after stagnation) are given in Table 1. For Opt-β^i we solved a one-dimensional version of Nelder–Mead optimization to keep points on the boundary.

7.0.2. Meshing curved surfaces

We applied Opt-β^i to a sphere and the Fertility sculpture. The disk-free and coverage distances are geodesic. For the uniform sphere, Fig. 14, Opt-β^i reduced β significantly, from 1.0 to 0.73. This is reflected in more regularity in the Voronoi and Delaunay meshes. Using CVT, the minimum β achieved was 0.81. Fig. 13 shows the quality measure distributions for valence, Delaunay angle, and Delaunay edge length.

For the non-uniform sphere, Fig. 12, we used a linear sizing function in the z-direction, varying from 0.21 to 0.01. We define the conflicts based on the *smaller-disk* criteria [15]. Fig. 12 shows that neither the Delaunay angle bounds nor the maximum global β have changed much; instead, their distributions have changed within about the same intervals.

We took the same approach with the Fertility model, a more complex shape. Fig. 15 shows that we reach convergence after a few iterations, and achieve a better angle and local β distribution.

7.1. Image filtering

Bilateral filtering is a core filtering method with a variety of applications in graphics, vision, and image processing [39]. Similar to traditional linear filtering, e.g. Gaussian blur, bilateral filtering

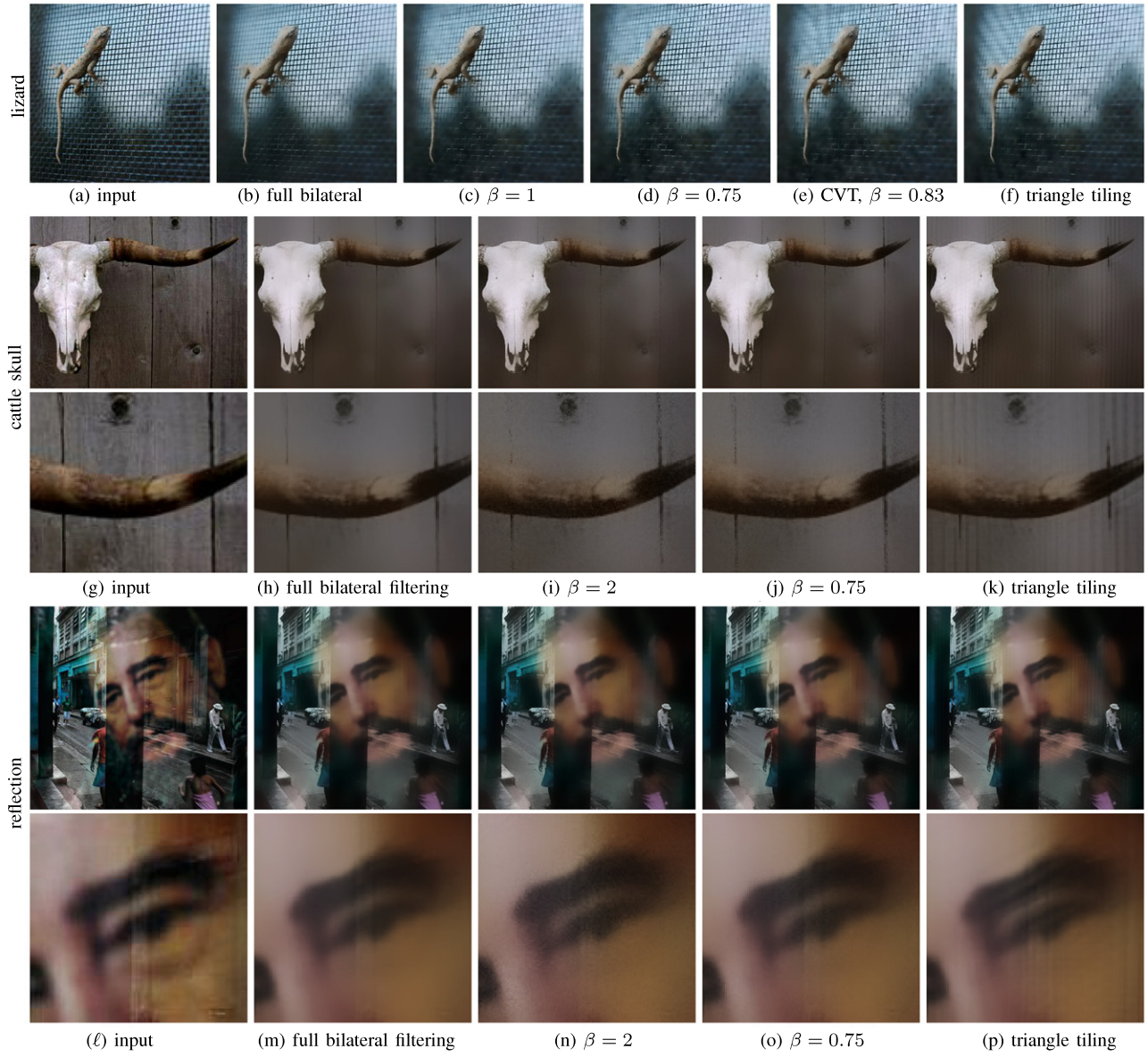


Fig. 16. Subsampling-accelerated bilateral filtering. $\beta = 0.75$ achieves the right balance between uniformity (reducing noise) and randomness (avoiding aliasing). Notice the noisier results with less uniform sampling ($\beta = 2.0$) and more aliasing with more regular sampling (CVT and triangle tiling). For the skull and reflection cases, we show both the whole images and partial zoom-ins. Table 2 lists the accuracy by β. All results are produced with a kernel width of 40 pixels and 30 samples per kernel, the recommended setting in [41].

strives to reduce noise through blending nearby pixels. However, unlike linear filtering, which assigns filter weights based on domain information (spatial pixel locations) only, bilateral filtering also considers range information, such as pixel colors. This allows better preservations of image features such as region boundaries.

The downside is that bilateral filtering is slower and more difficult to accelerate than linear filtering. Many methods have been proposed to accelerate it [40]. Among these, subsampling [41] is simple and effective. Instead of using all pixels, Banterle et al. [41] select only a subset of the kernel pixels. A properly chosen subset yields faster computation without a noticeable loss in output quality. The authors of [41] experimented with various subsampling methods, and concluded that Poisson-disk sampling can be an excellent choice, due to its uniformity (reducing filtering noise) and randomness (low aliasing). Their paper suggests that *maximal* Poisson-disk sampling could be even better.

Here, we explore maximal Poisson-disk sampling with different β for subsampling-accelerated bilateral filtering. Intuitively, since a smaller β will produce more uniform and more regular distributions, it has the potential to reduce noise but increase aliasing in the filtering output. This is confirmed in Fig. 16, in which we experiment through a range of β, using Opt-β^i ($\beta < 1$) and other methods ($\beta \geq 1$) [15]. In our experiments it seems that $\beta \in [0.85, 0.75]$ strikes the right balance between uniformity (reducing noise) and randomness (avoiding bias).

8. Conclusions

This paper introduced a Well-spaced Blue-noise Distribution (WBD), with $\beta = r_c/r_f$ measuring coverage uniformity or well-spacedness. We proposed the Opt-β^i algorithm to change a random point set to a WBD; blue noise is preserved up to $\beta \approx 0.75$. We demonstrated Opt-β^i's efficacy in geometry meshing and image filtering applications. We believe that the main contribution of this paper is on the introduction and measurement side, and we envision fruitful future directions for both algorithm and application

Table 2
Mean error and Root Mean Squared Error (RMSE) for various schemes over models from Fig. 16.

Measure	White noise	Opt-β^i, by β				Triangle tiling
		2	1	0.85	0.75	
Cattle skull						
Mean error	0.013	0.010	0.0093	0.0089	0.0087	0.012
RMSE	0.020	0.014	0.013	0.013	0.012	0.017
Reflection						
Mean error	0.020	0.013	0.012	0.011	0.010	0.016
RMSE	0.029	0.020	0.019	0.017	0.017	0.020

development. For example, generating a WBD with $\beta < 1$ from scratch is an open problem.

Adaptive and anisotropic sampling. We kept r_c and r_f constant throughout the domain, except for the sphere. In principle, we could extend both to spatially varying functions of sample positions, for adaptive or anisotropic sampling [42]. The spatial and spectral properties of such adaptive/anisotropic distributions can be analyzed by the differential domain method [37] through local warping. It remains to formulate an optimization objective function.

Tiling. We currently compute entire sample sets. A potential extension is generating sample tiles [43,44,35] for acceleration.

Higher dimensions and meshing. We would like to address $d > 2$. Recall that MPS [33] can produce $\beta = 1$ in any dimension, but the lower limit achieved by the densest packing is dimension dependent. In this paper we focused on planar meshing. Volumetric and general curved surface meshing would be interesting and important extensions. Lower β provides better angles in $d = 2$, and better radius-edge condition in $d \geq 2$, which are helpful for a variety of scientific and engineering applications.

Lower β. Our current algorithm can reach $\beta \approx 0.75$. We would like to investigate the theoretical potential of reaching even lower values, even though these distributions may have limited utility due to excessive regularity, as is evident from the hexagonal spectrum bias of the $\beta = 0.70$ case in Fig. 9. Also, the local optimization approach may require too much computation for small β.

Acknowledgments

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

[1] Bolander JE, Saito S. Fracture analyses using spring networks with random geometry. Engineering Fracture Mechanics 1998;61(5–6):569–91.
[2] Bishop Joseph. Simulating the pervasive fracture of materials and structures using randomly close packed Voronoi tessellations. Computational Mechanics 2009;44:455–71.
[3] Bishop Joseph E, Martinez Mario J, Newell Pania. A finite-element method for modeling fluid-pressure induced discrete-fracture propagation using random meshes. In: Proceedings of the 46th US rock mechanics/geomechanics symposium. American Rock Mechanics Association; 2012. Document ID 2012-190.
[4] Ebeida Mohamed S, Mitchell Scott A. Uniform random Voronoi meshes. In: 20th Int. meshing roundtable. 2011. p. 258–75.
[5] Ebeida MohamedS, Mitchell ScottA, Davidson AndrewA, Patney Anjul, Knupp PatrickM, Owens JohnD. Efficient and good Delaunay meshes from random points. Computer-Aided Design 2011;43(11):1506–15. Solid and Physical Modeling 2011.
[6] Rimoli Julian J, Rojas Juan J. Meshing strategies for the alleviation of mesh-induced effects in cohesive element models. Feb 2013. arXiv:1302.1161 [physics.comp-ph]. p. 1–19.
[7] Cook Robert L. Stochastic sampling in computer graphics. ACM Transactions on Graphics 1986;5(1):51–72.
[8] Ostromoukhov Victor, Donohue Charles, Jodoin Pierre-Marc. Fast hierarchical importance sampling with blue noise properties. In: SIGGRAPH'04. 2004. p. 488–95.
[9] Schechter Hagit, Bridson Robert. Ghost SPH for animating water. ACM Transactions on Graphics 2012;31(4): 61:1–8.
[10] Turk Greg. Re-tiling polygonal surfaces. In: SIGGRAPH'92. 1992. p. 55–64.
[11] Öztireli A Cengiz, Alexa Marc, Gross Markus. Spectral sampling of manifolds. In: SIGGRAPH ASIA'10. 2010. p. 168:1–8.
[12] Balzer Michael, Schlömer Thomas, Deussen Oliver. Capacity-constrained point distributions: a variant of Lloyd's method. In: SIGGRAPH'09. 2009. p. 86:1–8.
[13] Fattal Raanan. Blue-noise point sampling using kernel density model. In: SIGGRAPH'11. 2011. p. 48:1–12.
[14] Schlömer Thomas. PSA point set analysis. 2011. http://code.google.com/p/psa/.
[15] Mitchell Scott A, Rand Alexander, Ebeida Mohamed S, Bajaj Chadrajit. Variable radii Poisson-disk sampling, extended version. In: A Canadian conference on computational geometry. 2012. p. 1–9. URL: http://2012.cccg.ca/papers/paper15-extended.pdf.
[16] Ebeida Mohamed S, Patney Anjul, Mitchell Scott A, Davidson Andrew, Knupp Patrick M, Owens John D. Efficient maximal Poisson-disk sampling. In: SIGGRAPH'11. 2011. p. 49:1–12.
[17] Chew L Paul. Guaranteed-quality triangular meshes. Technical report 89-983. Department of Computer Science, Cornell University. 1989.
[18] Shewchuk J. Triangle: engineering a 2D quality mesh generator and Delaunay triangulator. In: Applied computational geometry towards geometric engineering. 1996. p. 203–22.
[19] Lagae Ares, Dutré Philip. A comparison of methods for generating Poisson disk distributions. Computer Graphics Forum 2008;21(1):114–29.
[20] Shirley Peter. Discrepancy as a quality measure for sample distributions. In: Eurographics'91. 1991. p. 183–94.
[21] Bondesson Lennart, Fahlén Jessica. Mean and variance of vacancy for hard-core disc processes and applications. Scandinavian Journal of Statistics 2003;30(4):797–816.
[22] Talmor Dafna. Well-spaced points for numerical methods. Ph.D. thesis. Tech report CMU-CS-97-164. Carnegie Mellon University. 1997.
[23] Jampani Ravi, Üngör Alper. Construction of sparse well-spaced point sets for quality tetrahedralizations. In: Int. meshing roundtable. 2008. p. 63–80.
[24] Miller Gary L, Sheehy Donald R, Velingker Ameya. A fast algorithm for well-spaced points and approximate Delaunay graphs. In: Symp. on computational geometry. 2013. p. 289–98.
[25] Du Q, Faber V, Gunzburger M. Centroidal Voronoi tessellations: applications and algorithms. SIAM Review 1999;41(4):637–76.
[26] Lloyd S. Least squares quantization in PCM. IEEE Transactions on Information Theory 1982;28(2):129–37.
[27] Liu Yang, Wang Wenping, Lévy Bruno, Sun Feng, Yan Dong-Ming, Lu Lin, et al. On centroidal Voronoi tessellation—energy smoothness and fast computation. ACM Transactions on Graphics 2009.
[28] Persson PO, Strang G. A simple mesh generator in MATLAB. SIAM Review 2004;46(2):329–45.
[29] Schlömer Thomas, Heck Daniel, Deussen Oliver. Farthest-point optimized point sets with maximized minimum distance. In: HPG'11. 2011. p. 135–42.
[30] Chen Renjie, Gotsman Craig. Parallel blue-noise sampling by constrained farthest point optimization. Computer Graphics Forum 2012;31(5):1775–85.
[31] Knupp PM. Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I: a framework for surface mesh optimization. International Journal for Numerical Methods in Engineering 2000;48:401–20.
[32] Nelder JA, Mead R. A simplex method for function minimization. The Computer Journal 1965;7(4):308–13.
[33] Ebeida Mohamed S, Mitchell Scott A, Patney Anjul, Davidson Andrew A, Owens John D. A simple algorithm for maximal Poisson-disk sampling in high dimensions. Computer Graphics Forum 2012;31(2pt4):785–94.
[34] Hansbo Peter. Generalized Laplacian smoothing of unstructured grids. Communications in Numerical Methods in Engineering 1995;11(5):455–64.
[35] Ostromoukhov Victor. Sampling with polyominoes. In: SIGGRAPH'07. 2007. p. 78.
[36] Wei Li-Yi. Parallel Poisson disk sampling. In: SIGGRAPH'08. 2008. p. 20:1–9.
[37] Wei Li-Yi, Wang Rui. Differential domain analysis for non-uniform sampling. In: SIGGRAPH'11. 2011. p. 50:1–10.
[38] de Goes Fernando, Breeden Katherine, Ostromoukhov Victor, Desbrun Mathieu. Blue noise through optimal transport. In: SIGGRAPH Asia'12. 2012. p. 171:1–11.
[39] Tomasi Carlo, Manduchi Roberto. Bilateral filtering for gray and color images. In: Int. conf. on computer vision. 1998. p. 839–46.
[40] Paris Sylvain, Kornprobst Pierre, Tumblin Jack, Durand Frédo. Bilateral filtering: theory and applications. Foundations and Trends in Computer Graphics and Vision 2009.
[41] Banterle Francesco, Corsini Massimiliano, Cignoni Paolo, Scopigno R. A low-memory, straightforward and fast bilateral filter through subsampling in spatial domain. Computer Graphics Forum 2012;31(1):19–32.
[42] Li Hongwei, Wei Li-Yi, Sander Pedro, Fu Chi-Wing. Anisotropic blue noise sampling. In: SIGGRAPH Asia'10. 2010. p. 167:1–12.
[43] Cohen Michael F, Shade Jonathan, Hiller Stefan, Deussen Oliver. Wang tiles for image and texture generation. In: SIGGRAPH'03. 2003. p. 287–94.
[44] Kopf Johannes, Cohen-Or Daniel, Deussen Oliver, Lischinski Dani. Recursive Wang tiles for real-time blue noise. In: SIGGRAPH'06. 2006. p. 509–18.