A Survey of Blue-Noise Sampling and Its Applications

Dong-Ming Yan [1,2] (严冬明), *Member, CCF, ACM*, Jian-Wei Guo [2] (郭建伟), Bin Wang [3] (王 斌)
Xiao-Peng Zhang [2] (张晓鹏), and Peter Wonka [1,4]

[1] *Visual Computing Center, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia*

[2] *National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*

[3] *School of Software, Tsinghua University, Beijing 100084, China*

[4] *Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287, U.S.A.*

E-mail: yandongming@gmail.com; jianwei.guo@nlpr.ia.ac.cn; wangbins@tsinghua.edu.cn
xiaopeng.zhang@ia.ac.cn; pwonka@gmail.com

Abstract In this paper, we survey recent approaches to blue-noise sampling and discuss their beneficial applications. We discuss the sampling algorithms that use points as sampling primitives and classify the sampling algorithms based on various aspects, e.g., the sampling domain and the type of algorithm. We demonstrate several well-known applications that can be improved by recent blue-noise sampling techniques, as well as some new applications such as dynamic sampling and blue-noise remeshing.

Keywords blue-noise sampling, Poisson-disk sampling, Lloyd relaxation, rendering, remeshing

1 Introduction

Sampling is an essential technique in computer science. Sampling translates a continuous signal into its discrete counterpart or selects a subset from a discrete set of signals, such that the signal can be represented and processed by computers efficiently. For example, one-dimensional (1D) sound waves, two-dimensional (2D) images, and three-dimensional (3D) polygonal meshes are captured by discrete sampling from continues signals.

In computer graphics, sampling plays an important role in many applications, such as rendering[1], stippling[2], texture synthesis[3], object distribution[4], and simulation[5]. Among all the sampling techniques, *blue-noise sampling* is the most popular method in recent papers. The term "blue-noise" refers to any noise with minimal low-frequency components and no concentrated spikes in energy. Intuitively, blue-noise sampling generates randomized uniform distributions. Fig.1 shows a typical point set with blue-noise properties generated by Poisson-disk sampling.

In this paper, we first discuss the color of noise in Section 2, and then summarize recent techniques for blue-noise sampling in Section 3. We focus on recent approaches in our survey as a complement to the comprehensive survey by Lagae and Dutré[6]. Furthermore, we discuss several existing and new applications that benefit from the blue-noise sampling techniques in Section 4.

2 Color of Noise

We can classify point distributions by looking at the Fourier spectrum. Different spectra are associated with different colors①. For example, white noise refers

---
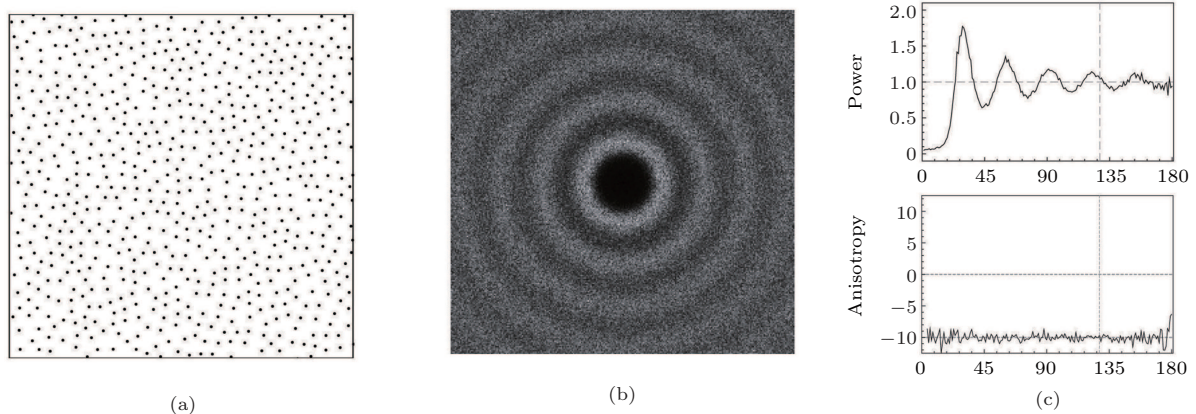
① http://en.wikipedia.org/wiki/Colors_of_noise, Mar. 2015.

Fig.1. Example of Poisson-disk sampling and its spectral analysis. (a) A sampled point set. (b) Power spectrum from this point set. (c) Radial means and normal anisotropy.

to noise with a flat spectrum, which contains an equal amount of energy in all frequency bands. It is usually used in random number generators[7]. Blue noise refers to point distributions with weak low-frequency energy rather than strong high-frequency energy. Pink noise is the complement of blue noise and its spectral energy is concentrated in the low-frequency bands. Pink noise occurs very frequently in nature and thus is used for physical simulation and biological distributions[8-9]. Green noise is an uncommon term that can refer to the mid-frequencies of white noise. It characterizes the distributions of a variety of natural phenomena and has been used for digital halftoning[10].

There are also many other colors used for noise, with or without precise definitions. In this paper, we focus only on techniques that generate blue-noise sampling patterns.

3 Blue-Noise Sampling

There are various ways to characterize existing blue-noise sampling techniques. For example, the sampling algorithms can be classified by the type of sampling domain (2D, 3D, or surfaces), the metric used in the domain (geodesic or Euclidean), the shape of the sampling primitives (point, line, ball, etc.), the properties of the sampling results (isotropic or anisotropic), the style of the algorithm (dart throwing, relaxation, or tiling), and so on.

3.1 Sampling Domain

Here, we briefly describe the common input domains.

Euclidean Domain. Most previous blue-noise sampling algorithms were first developed to handle the 2D or 3D Euclidean space. The traditional domain is the unit torus (i.e., the unit square and cube in 2D and 3D, respectively, with periodic boundary conditions)[11], in which the distance between two points is measured using the Euclidean metric. Some work also addresses more complicated domains, such as non-convex polygons with holes[12-13].

High Dimensions. Since high-dimensional point distributions have special applications, some approaches are able to generalize blue-noise sampling to high dimensions[13-16]. However, many high-dimensional sampling methods do not scale well with high dimensions because they typically suffer from the curse-of-dimensionality, which means that the effectiveness deteriorates very rapidly as the dimensions increase. Ideas to overcome this problem have been suggested by Ebeida *et al.*[17]

3D Surface. Blue-noise sampling has also been extended to mesh surfaces. In this case, the input is usually a two-manifold triangular mesh surface, which consists of a set of triangles. The sampled points should be located exactly on the surface.

3.2 Sampling Algorithm

Blue-noise sampling methods can be roughly classified into three types according to the techniques they use: 1) Poisson-disk sampling and its variations; 2) relaxation-based sampling; and 3) patch/tile-based sampling. In this subsection, we give an overview of these methods and discuss several classic algorithms in detail. Each type of algorithm is further classified by the sampling domain. We emphasize methods that use

points as primitives for isotropic blue-noise sampling in lower dimensions.

3.2.1 Poisson-Disk Sampling

Poisson-disk sampling is a classic technique that generates uniformly randomly distributed point sets. An ideal Poisson-disk sampled point set, $X = \{(\boldsymbol{x}_i, r_i)\}_{i=1}^{n}$, in sampling domain Ω should satisfy the following three properties: 1) minimal distance property, which requires that the distance between any two disk centers should be larger than the sampling radius, i.e., $\forall \boldsymbol{x}_i, \boldsymbol{x}_j \in P, \|\boldsymbol{x}_i, \boldsymbol{x}_j\| \geqslant \min(r_i, r_j)$; 2) unbiased sampling property, which requires that each point in the domain has a probability that is proportional to the sizing at this point to receive a sampling point; and 3) maximal sampling property, which requires that the union of the disks covers the entire sampling domain, i.e., $\bigcup (\boldsymbol{x}_i, r_i) \supseteq \Omega$. The sampling is uniform if the sampling radius, r_i, is constant. Otherwise, it becomes adaptive sampling.

Euclidean Domain. The traditional method for Poisson-disk sampling is called dart-throwing and was first proposed by Cook[18]. Given a sampling domain and a sampling radius, the algorithm generates disks in the sampling domain randomly. If the current generated disk conflicts with any previous sampled disk, then it is rejected; otherwise, it is accepted. This process is repeated until a continuous number of rejections are observed. The algorithm complexity of the original dart-throwing algorithm is $O(n^2)$. However, this approach is inefficient to achieve the maximal property. Therefore, a lot of work has been expended on generalizing and accelerating this algorithm.

Most recent work aiming for efficient Poisson-disk sampling maintains a data structure to track and sample the empty regions (also called gap primitives). Dunbar and Humphreys[19] described an efficient implementation of the dart-throwing algorithm for maximal Poisson-disk sampling (Fig.2(a)). They applied a data structure called scalloped sectors to record the active front of the sampled disk set. Their algorithm runs in $O(n \log(n))$ time, but the sampling is biased.

The simplest data structure for uniform Poisson-disk sampling is the quad-tree. White *et al.*[20] first proposed to use such a data structure for acceleration. The cell size of the base grid equals $\frac{r}{\sqrt{2}}$, such that each grid cell can at most receive one disk with radius r. During the sampling process, the partially covered cells are subdivided into smaller fragments in a quad-tree manner. Gamito and Maddock[16] extended White *et*

al.'s algorithm to higher dimensions. Later, Jones and Karger[21] reported that they reduced the time complexity of Poisson-disk sampling to linear time. The follow-up work of Ebeida *et al.*[13] further accelerated the sampling process by sampling a flat fragment array instead of using the hierarchical quad-tree, as shown in Fig.2(b). The grid-based sampling algorithm has been implemented on recent graphics processing units (GPUs)[15,22-23]. But the GPU extensions cannot guarantee the unbiased sampling property.
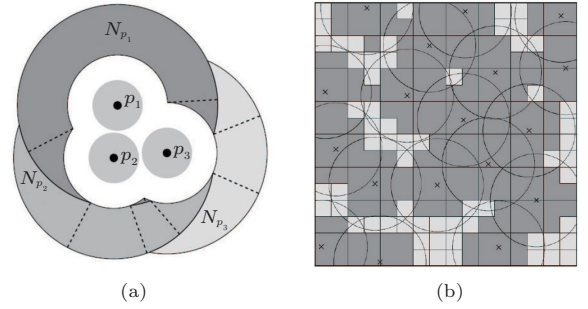

Fig.2. Data structures used for accelerating Poisson-disk sampling. (a) Scalloped sectors[19]. (b) Quad-tree[13].

Jones[24] first proposed an algorithm for unbiased maximal Poisson-disk sampling. A Voronoi diagram is used to extract the uncovered regions, called gaps in the sampling domain. These uncovered regions are further resampled to achieve the maximal sampling property in an unbiased manner. The core idea is that maximal sampling can be obtained if and only if the Voronoi cell of each vertex is fully covered by the disk centered at the vertex. In the sampling process, disks are repeatedly generated and inserted in a global Voronoi diagram one by one, until the sampling becomes maximal. Each vertex in the Voronoi diagram records a value that indicates the area of the empty region of the corresponding Voronoi cell. A new sample is generated by first selecting a Voronoi cell based on the empty area. Once a new sample is generated, the Voronoi diagram and the value of each vertex are updated.

Ebeida *et al.*[12] proposed a hybrid approach that first uses squares and later convex polygons bounding the intersections of a square and multiple circles as gap primitives. They developed a two-step unbiased maximal sampling framework. They first performed classic dart-throwing on a uniform grid and then switched to computing and filling empty regions by clipping the grid cells against the neighboring disks.

However, most algorithms mentioned above can only handle uniform sampling. Yan and Wonka[25] presented an algorithm for the generation of maximal

442

J. Comput. Sci. & Technol., May 2015, Vol.30, No.3

Poisson-disk sets with varying radii. Built on the regular triangulation and the power diagram, they conducted a theoretical analysis of gaps in such disk sets. Then they designed efficient algorithms and data structures for gap detection and gap updates when the disks were changed. Their method works well both in Euclidean space and on manifolds. Fig.3 compares the gap extraction algorithms of [12, 24-25].
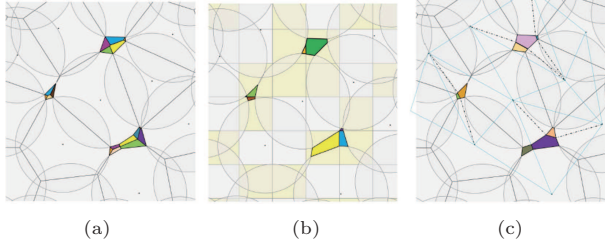

(a)   (b)   (c)

Fig.3. Comparison of three representative algorithms for gap computation. (a) Voronoi diagram[24]. (b) Uniform grid[12]. (c) Regular triangulation and power diagram[25].

Mitchell *et al.*[26] studied 2D Poisson-disk sampling with various radii. They analyzed the conflicting condition under which a maximal sampling can be achieved. Ebeida *et al.*[27] introduced a sifted disk technique for locally resampling a point cloud to reduce the number of points. The essence of this algorithm is still maximal Poisson-disk sampling. More recently, Yuksel[28] proposed a point set resampling approach for fast Poisson-disk sampling; however, this approach is biased and not maximal.

Surface Sampling. The classic dart-throwing algorithm has been extended to mesh surfaces[25,28-36] as well as to isosurfaces[37]. Euclidean distance is used in these approaches. Besides the Euclidean metric, the geodesic metric is also used for Poisson-disk sampling[38-41]. Surface sampling techniques can be also used for surface remeshing by computing the restricted Delaunay triangulation[42]. We address this issue as an application of blue-noise sampling in Subsection 4.4.

3.2.2 Relaxation-Based Sampling

Iterative relaxation is another important technique for generating point distributions. This type of method usually consists of two steps: 1) generating an initial point set, and 2) optimizing the point positions using Lloyd iterations[43] until convergence. In this type of approach, the points $X = \{\boldsymbol{x}_i\}_{i=1}^n$ to be optimized are called sites.

The methods are differentiated by different objective functions. For example, the original Lloyd algorithm minimizes the quantization error in signal processing. It is known as the centroidal Voronoi tessellation (CVT) in computer graphics[44]. The energy function of CVT can be formulated as

$$E_{\mathrm{CVT}}(X) = \sum_{i=1}^n \int_{V_i} \rho(\boldsymbol{x})\|\boldsymbol{x} - \boldsymbol{x}_i\|^2 \mathrm{d}\boldsymbol{x}, \qquad (1)$$

where $\{V_i\}_{i=1}^n$ is the Voronoi diagram of the points in the sampling domain Ω and $\rho(\boldsymbol{x})$ is a density function defined over Ω.

The key ingredient of computing a CVT is to construct the Voronoi diagram. The Voronoi diagram in Euclidean spaces is well studied. It has also been generalized on manifold surfaces. There are multiple ways to compute CVT on mesh surfaces, e.g., parameterization-based approaches[45-46], discrete clustering[47], and the exact computation of the Voronoi diagram on surfaces[34,42]. We compare these different approaches in Subsection 4.4.

However, CVT tends to generate a point distribution with regular patterns (i.e., hexagonal arrangements) that lacks some blue-noise properties. Several methods have been proposed to modify CVT to obtain better blue-noise properties.

Balzer *et al.*[48] introduced a capacity constrained Voronoi tessellation (CCVT) to generate point sets with excellent blue-noise properties. In this method, Voronoi cells should satisfy the constraints $\|V_i\| = \int_{V_i} \rho(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = c_i$, where c_i are capacity constraints with $c_i > 0$. Intuitively, the capacity of a site can be understood as the area of its corresponding Voronoi region weighted by the density function. However, since this method relies on a discretization of the capacities, it suffers from quadratic complexity and converges slowly. Even when implemented on a GPU[49], it is still inefficient for large-scale optimization problems. Three variants were further proposed to improve the algorithm performance.

Xu *et al.*[50] proposed capacity-constrained Delaunay triangulation (CCDT) for blue-noise sampling and generalized the concept of CCDT to mesh surfaces[51].

Chen *et al.*[52] combined CCVT[48] with the CVT framework in [42] for surface blue-noise sampling, which they called capacity-constrained centroidal Voronoi tessellation (CapCVT). Their energy function is defined as:

$$E_{\mathrm{CapCVT}}(X) = E_{\mathrm{CVT}}(X) + \lambda E_{\mathrm{CapVT}}(X),$$

where E_{CVT} is the same as (1) and $E_{\text{CapVT}}(X) = \sum_{i=1}^{n}(\int_{V_i} \rho(\boldsymbol{x})\mathrm{d}\boldsymbol{x})^2$. Then, using an efficient optimization framework based on the L-BFGS method[53], they achieved significant performance improvement.

de Goes *et al.*[54] reformulated CCVT as a continuous constrained minimization problem based on optimal transport, instead of the discretized approximation suggested in [48]. In addition, they used a power diagram, as opposed to a Voronoi diagram, to perform density-adapted sampling. They formulated their constrained minimization as:

$$E_{\text{BNOT}}(X, W) = \varepsilon(X, W) + \sum_{i=1} \lambda_i(c_i - c),$$

where $\varepsilon(X, W) = \sum_{i=1}^{n} \int_{V_i} \rho(\boldsymbol{x})\|\boldsymbol{x} - \boldsymbol{x}_i\|^2 \mathrm{d}\boldsymbol{x}$, $W = \{w_i\}_{i=1}^{n}$ is the weight defined at each site.

Another kind of relaxation technique, called farthest point optimization (FPO), maximizes the minimal distance of a given point set. The energy function of FPO is discrete and can be optimized only by discrete optimization method. The original FPO algorithm was proposed by Schlömer *et al.*[11] An equivalent algorithm was proposed by Kanamori *et al.*[55], which is based on Delaunay triangulations. They successively moved each point to the farthest point (i.e., the Voronoi vertex farthest from its immediate neighbors), by removing it and reinserting it at the farthest point. Chen and Gotsman[56] parallelized the FPO framework of [11] via local Delaunay triangulation. However, these two FPO approaches could handle only 2D uniform sampling. By introducing the regular triangulation and the power diagram, Yan *et al.*[57] proposed two important generalizations of the original FPO framework: adaptive sampling and sampling on surfaces. Fig.4 shows an example of applying FPO to a mesh surface.

Apart from the above two categories, there are many other approaches that aim at high-quality sampling based on relaxation techniques. Öztireli *et al.*[58] solved the problem of finding optimal sampling conditions based on the spectral analysis of manifolds. Fattal[2] presented an adaptive sampling algorithm based on kernel density estimation. Chen *et al.*[59] introduced bilateral blue-noise sampling that is suitable for dense point set sub-sampling. Ebeida *et al.*[60] proposed an iterative optimization method to improve blue-noise properties starting from a Poisson-disk sampled point set.

3.2.3 Patch/Tile-Based Sampling

Possion-disk sampling and relaxation-based methods can generate high-quality point sets, but the computational overhead can become an issue for real-time applications. Patch/tile-based sampling is able to generate large point sets in real time while sacrificing the sampling quality.

The core of a tile-based sampling method is that one or more tiles are pre-computed and then placed next to each other to form point sets of arbitrary sizes. Hiller *et al.*[61] first utilized Wang tiles[62-63] to generate non-periodic point sets with blue-noise properties. Lagae and Dutré[64] extended Wang tiles to Poisson-disk tiles aiming at the rapid generation of Poisson-disk distributed point sets. A recursive tile subdivision proposed by Kopf *et al.*[65] also uses Wang tiles to produce a higher level of noise. However, Wang-tile approaches usually generate sampling artifacts because the low count of prototiles and their placement on a square lattice induce a grid of peaks in Fourier spectra.

Ostromoukhov *et al.*[66] proposed to utilize non-periodic Penrose tiling for generating blue-noise patterns on 2D domains. They hierarchically subdivided
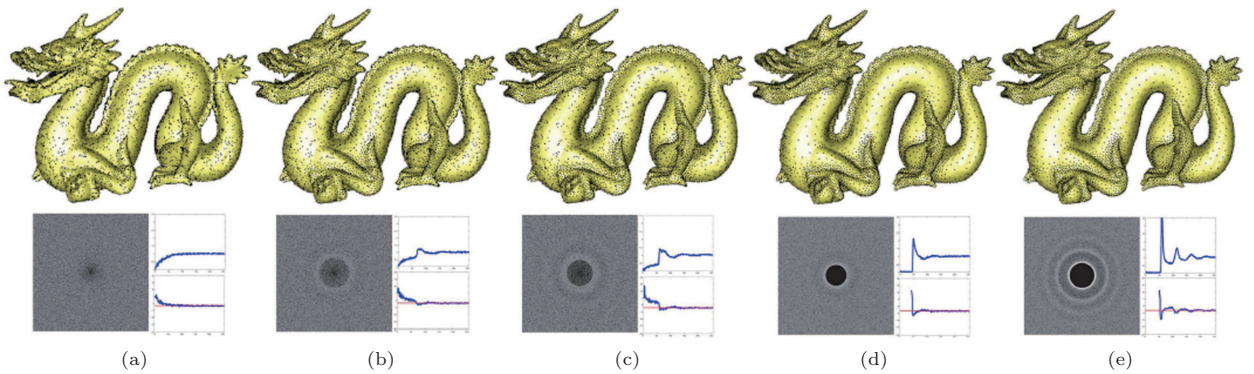

Fig.4. Example of relaxation-based optimization[57]. (a)∼(e) Results of increasing the number of iterations. The top row: the sampling results, the bottom row: the results of spectral analysis.

a Penrose tiling and used the Fibonacci number system to label the sampled point. This technique is well suited for the generation of non-uniform sampling patterns. However, this method yields rather strong artifacts in the spectral domain. Ostromoukhov then improved the previous work[66] for general fast hierarchical importance sampling[67]. This approach is built on self-similar tiling of the plane or the surface of a sphere with rectifiable polyominoes, as opposed to Penrose tiling. Each polyomino contains just one sample and it is recursively subdivided until the desired local density of samples is reached. The exact position of the sampling point within the polyomino is determined by a pre-computed structural index. However, these two approaches result in poor Fourier spectra, as the single-sample tiles they use exacerbate the presence of tiling structures that are very simple or regular. Considering the drawbacks of Wang tiles and single-sample tiles, Wachtel *et al.*[68] proposed a new fast tile-based method for adaptive 2D sampling. At its heart is an adaptive non-periodic tiling with a deterministic, hierarchical construction of self-similar, equal-areal, tri-hex tiles. Then an offline computation of a lookup table of optimized (spectrally controlled) point sets is used to populate the tiles.

Kalantari and Sen[69-70] proposed other efficient methods for fast generation of a large number of blue-noise samples. Their main idea is to generate an initial set of Poisson-disk samples first using any existing approach, and then replicate this set at various locations in the final space using the convolution theorem. This method is very fast, but its blue-noise properties are not so good as the previously mentioned Poisson-disk sampling methods since it is approximate.

3.2.4 Other Approaches

There are several other approaches that cannot be classified by the categories discussed above. For example, Zhou *et al.*[71] and Heck *et al.*[72] studied the relationships between spatial statistics and spectral properties of point distributions, and they proposed efficient methods to generate point sets that match the given spectra. Mitchell *et al.*[73] generated a blue-noise quad-mesh using two-color Poisson-disk sampling. Beyond the point dart, high-dimensional darts were also investigated for blue-noise sampling[17,74-76]. Other sampling methods focus on anisotropic sampling instead of isotropic sampling[77-79]. We do not discuss the details of these approaches here because they are out of the scope of this paper.

3.3 Evaluation

Different sampling algorithms result in point distributions that have different characteristics. Hence, how to choose the right method for a given application is very important. So far, there are mainly two methodologies for evaluating the quality of samples: spectral properties and geometric analysis.

Spectral Evaluation. Spectral analysis is a common method for evaluating the quality of point distributions and has been demonstrated to be effective in detecting sampling artifacts. The first technique was introduced by Ulichney[80] to study dither patterns. The power spectrum is estimated by averaging the periodogram of distributions, determined by Fourier transforms. Then, two useful one-dimensional statistics from the power spectrum are derived from the power spectrum. The first is the *radially averaged power spectrum*, in which the typical blue-noise characteristic should start in a sharp transition region, with a low-frequency cutoff and a flatter, high-frequency region. The second one is *anisotropy* that measures the radial symmetry of the power spectrum. This tool was used by [6] to compare different methods for generating Poisson-disk distributions. Schlömer and Deussen[81] extended the work of [80] and [6] to investigate accuracy issues regarding the spectral analysis of 2D point sets. In addition, Öztireli and Gross[82] and Subar and Kautz[83] proposed approaches to analyze the quality of samples. The former is based on the statistical measure pair correlation function (PCF), while the later utilizes the amplitude and variance of the sampling spectrum.

The evaluation of blue-noise sampling on surfaces is difficult since a typical Fourier analysis cannot be directly used. Bowers *et al.*[30] first proposed a spectral analysis method for surface sampling, but it could only be used for analyzing uniform sampling. Wei and Wang[84] introduced the differential domain analysis (DDA) technique for analyzing the spectral properties of non-uniformly sampled point sets, as well as for surface sampling. With this tool, it is possible to analyze the blue-noise properties of various methods on surfaces.

Fig.5 compares selected blue-noise sampling algorithms by using the spectral analysis tool PSA provided by Schlömer and Deussen[81]. Fig.6 does the comparison on mesh surfaces by applying the differential domain analysis of Wei and Wang[84].

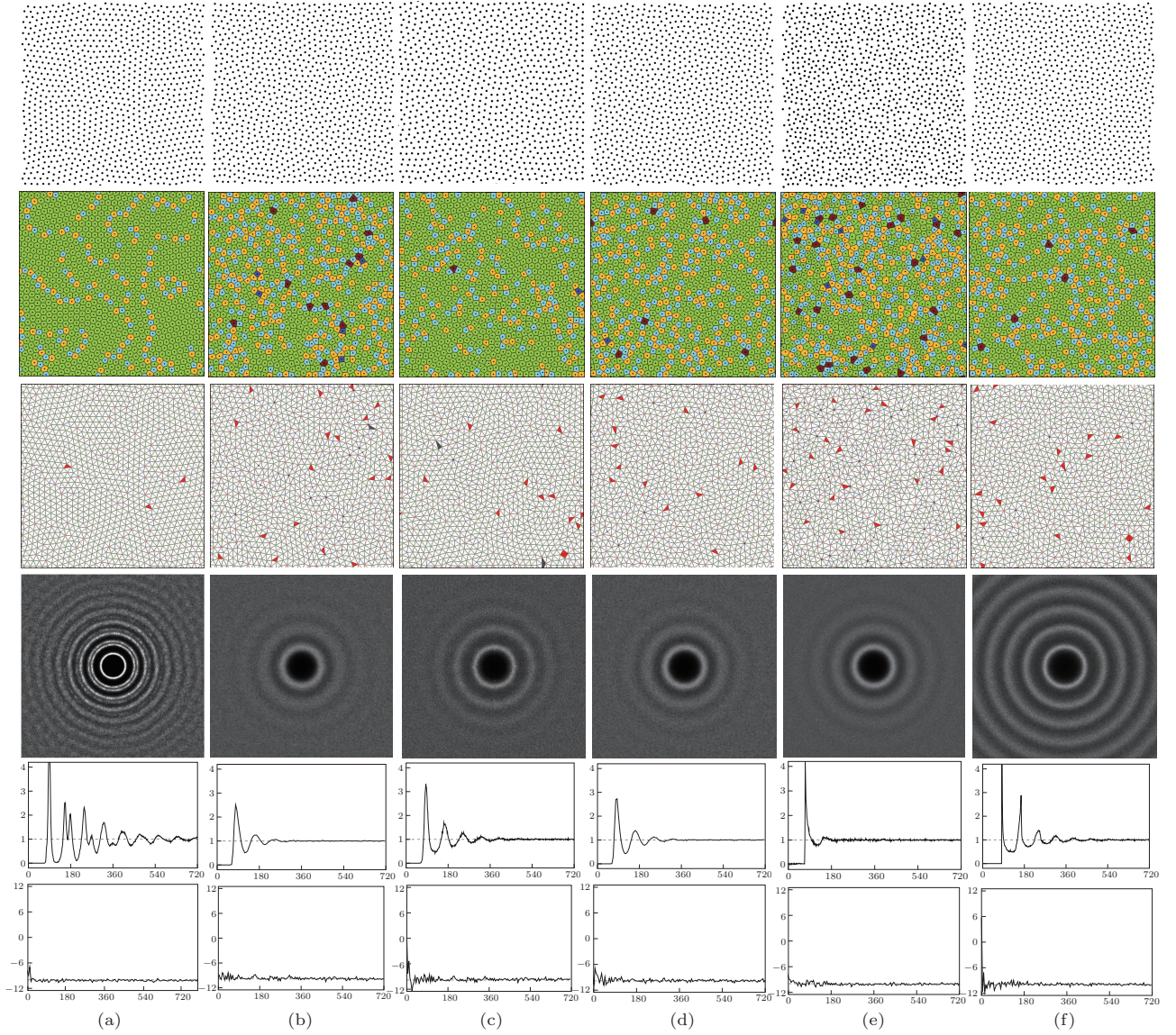*Geometric Evaluation.* Various spatial quantities have been proposed to measure the spatial distribution

Fig.5. Comparison of recent 2D blue-noise sampling algorithms, including (a) CVT[44,53], (b) CCVT[48], (c) CapCVT[52], (d) BNOT[54], (e) MPS[25], and (f) FPO[11]. From top to bottom: distributions of 1 024 points in a periodic square, Voronoi cells (each cell is color-coded by its degree: green is valence 6, orange is valence 7, light blue is valence 5, dark blue is valence 4 and brown is valence larger than 7), Delaunay triangulation (triangles with the minimal angle less than 30° are shown in dark gray and the obtuse triangles are shown in red), the power spectrum, radial means and anisotropy.

properties of samples. One common choice is the relative radius, $\delta_X = d_{\min}/d_{\max}$, defined in [6, 81], where $d_{\min}$ is the global minimum distance for any pair of points in point set X and $d_{\max}$ is the theoretically largest minimum distance between any two points (i.e., $d_{\max} = \sqrt{(2/\sqrt{3}\,n)}$). Other spatial measures used in recent meshing/remeshing papers are listed in Table 1 and Table 2. The quality of a triangle is measured by $Q_t = \frac{6}{\sqrt{3}}\frac{s_t}{p_t h_t}$, where s_t is the area of t, p_t is the half-perimeter of t and h_t is the longest edge length of t[85]. Here, $Q_{\min}$ and Q_{avg} are the minimal and the average triangle quality respectively; $\theta_{\min}$ and $\theta_{\max}$ are the minimal and the maximal angle respectively, and $\bar{\theta}_{\min}$ is the average of the minimal angles of all triangles; $\theta_{<30°}$ and $\theta_{>90°}$ are the ratios of the triangles with $\theta_{\min}$ smaller than 30° and with $\theta_{\max}$ larger than 90°; V_{567} is the percentage of vertices with valences 5, 6 and 7; d_{H} and d_{RMS} are the Hausdorff distance and the root mean squared distance between the input mesh and the remeshing result (divided by the diagonal length of the input mesh bounding box), measured with the Metro tool[86].
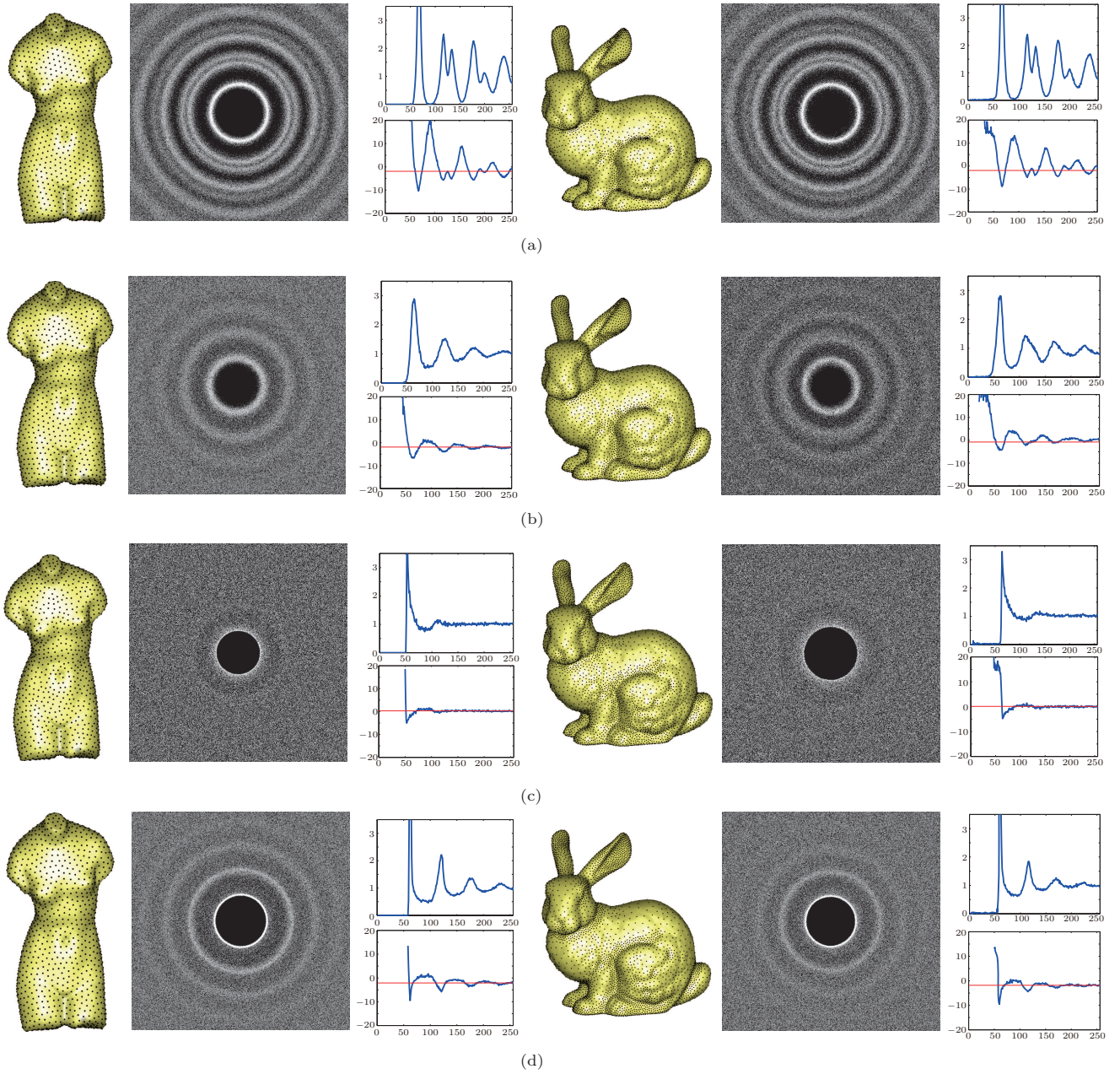
Fig. 6. Comparison of surface sampling and remeshing. (a) CVT[42]. (b) CapCVT[52]. (c) MPS[25]. (d) FPO[57]. From left to right: sampling points, the power spectrum, radial means and anisotropy. For each method, the first image shows uniform sampling on the Venus mode with 1 800 samples, and the fourth image from the left shows adaptive sampling on the Bunny model with about 6 300 samples.

4   Applications

In this section, we discuss several applications that can benefit from blue-noise sampling.

4.1   Rendering

Rendering algorithms typically face the challenge of numerically computing high-dimensional integrals.

Most of the time, sampling algorithms are used and therefore sample generation is a core problem in rendering. Due to its low discrepancy and randomness, blue-noise sampling has been exploited to improve rendering quality and efficiency. Spencer and Jones[87] applied blue-noise sampling to caustics rendering powered by photon mapping. They presented a method that progressively removes noise from photon maps, which

Table 1. Statistics of 2D Sampling and Meshing Qualities

Method	δ_X	$Q_{\min}$	Q_{avg}	$\theta_{\min}$	$\bar{\theta}_{\min}$	$\theta_{\max}$	$\theta_{<30°}\%$	$\theta_{>90°}\%$	$V_{567}\%$
CVT	0.793	**0.643**	**0.932**	**38.97**	**54.37**	**98.28**	**0.00**	**0.19**	**100.00**
CCVT	0.778	0.518	0.832	28.82	47.13	113.26	0.05	8.74	98.54
CapCVT	0.741	0.512	0.847	28.67	48.44	116.83	0.17	6.81	99.91
BNOT	0.766	0.570	0.848	31.67	48.23	107.08	**0.00**	6.34	99.02
MPS	0.781	0.487	0.806	30.19	45.30	117.11	**0.00**	15.07	96.53
FPO	**0.925**	0.567	0.856	35.12	50.90	107.51	**0.00**	6.50	99.61

Table 2. Statistics of Remeshing Qualities

Model	Method	$\|X\|(\times 10^3)$	$Q_{\min}$	Q_{avg}	$\theta_{\min}$	$\bar{\theta}_{\min}$	$\theta_{\max}$	$\theta_{<30°}\%$	$\theta_{>90°}\%$	$V_{567}\%$	$d_{\text{RMS}}(\times 10^{-3})$	$d_{\text{H}}(\times 10^{-2})$
Venus	MPS	3.0	**0.67**	0.85	32.7	48.6	**90.0**	**0.00**	**0.00**	**100.0**	0.67	0.68
	FPO	3.0	0.57	0.85	34.2	50.8	107.1	**0.00**	6.29	99.7	**0.64**	0.71
	CapCVT	3.0	0.39	0.78	20.5	43.3	128.9	4.41	17.7	98.8	0.68	0.61
	CVT	3.0	0.65	**0.93**	**39.5**	**54.5**	97.3	**0.00**	0.25	**100.0**	0.76	**0.59**
Bunny	MPS	8.3	0.40	0.83	33.9	53.9	103.0	**0.00**	0.29	**100.0**	0.47	0.34
	FPO	8.0	0.39	0.84	22.6	48.7	128.5	0.56	0.29	98.0	0.45	0.34
	CapCVT	8.0	0.39	0.84	15.6	47.3	125.1	1.04	8.46	98.7	**0.43**	**0.24**
	CVT	8.0	**0.64**	**0.93**	34.8	54.2	**98.8**	**0.00**	**0.06**	99.9	0.53	0.37

are view-independent. The resulting photon distribution holds blue-noise properties, which improve the rendering quality while avoiding a huge amount of photons. Chen *et al.*[59] further improved the blue-noise distribution of the photons. The so-called "bilateral blue-noise sampling" method[59] considers not only the photon positions, but also the photon properties during relaxation, resulting in a better quality both in smooth regions and sharp features. A comparison of the rendering results is shown in Fig.7.

4.2 Image/Video Stippling

Stippling is a kind of art form using points to represent a drawing/painting. The contrast of the image is controlled by using different densities of point distribution in different regions. The distribution of the points cannot be regular or there would be visual artifacts. Blue-noise sampling is well suited for this application. Secord[88] used a weighted Voronoi diagram for image stippling. Then, various techniques were proposed to improve the quality of stippling[2,48,52,54,89]. More recently, Ge *et al.*[90] used bilateral blue-noise sampling[59] for video stippling. Fig.8 shows two recent examples of image and video stippling.

4.3 Dynamic Sampling

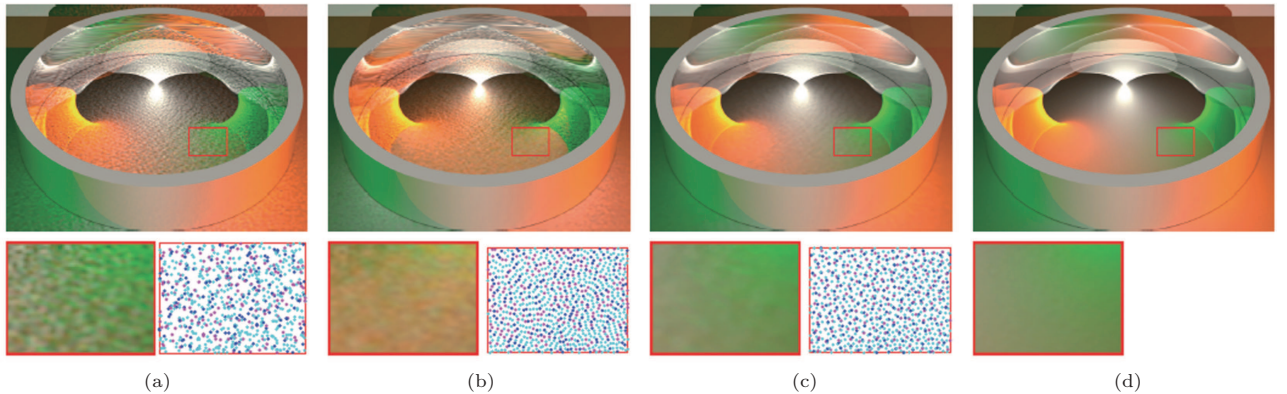We propose two new applications for dynamic sampling based on maximal Poisson-disk sampling[25].

Fig. 7. Comparison of photon mapping results of different methods. (a) Unrelaxed result. (b) Result of [87]. (c) Result of [59]. (d) Reference 40X photons.
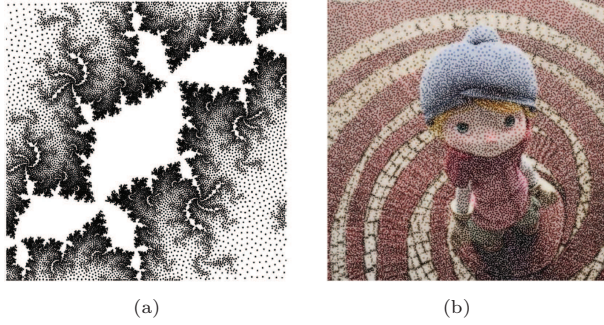
Fig.8. Blue-noise sampling for image and video stippling. (a) Image stippling[54]. (b) Video stippling[90].

Plant Growth Simulation. We adapt a plant growth simulation[4] to formulate it as an adaptive Poisson-disk sampling problem[25]. Starting with a set of seeds (disks with minimal radii), the radius of each plant (disk) increases over time. Different species have varying growth speeds. The plants whose centers are covered by others are removed due to insufficient light. New gaps are created and filled by new seeds repeatedly. Fig.9 shows the simulation in different stages.
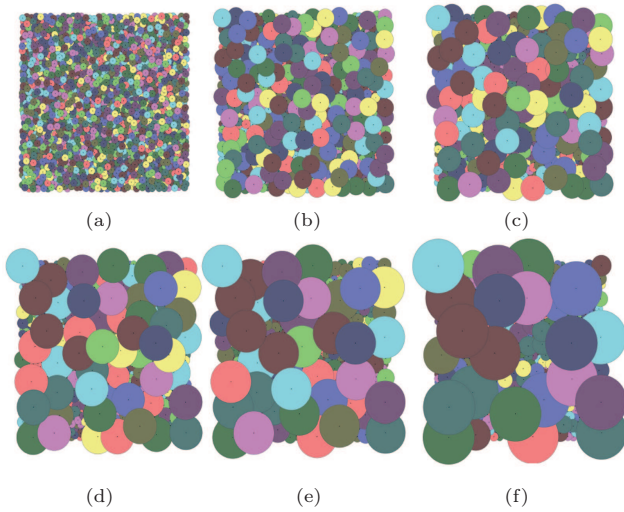

Fig.9. Plant simulation with Poisson-disk sampling. (a) Time = 0. (b) Time = 300. (c) Time = 600. (d) Time = 1 200. (e) Time = 1 800. (f) Time = 3 000.

Dynamic Point Set. We consider time varying blue-noise triangulations as another application of dynamic sampling. We choose to represent the changes over time as a dynamical system where the vertex positions are integrated according to a vector field. Fig.10(a) shows an example. This application uses efficient operations for inserting, deleting, and moving points as proposed in [25]. The spectral analysis of the first and the last

frame shows that the blue-noise properties of the triangulation are maintained over time (Fig.10(b)).
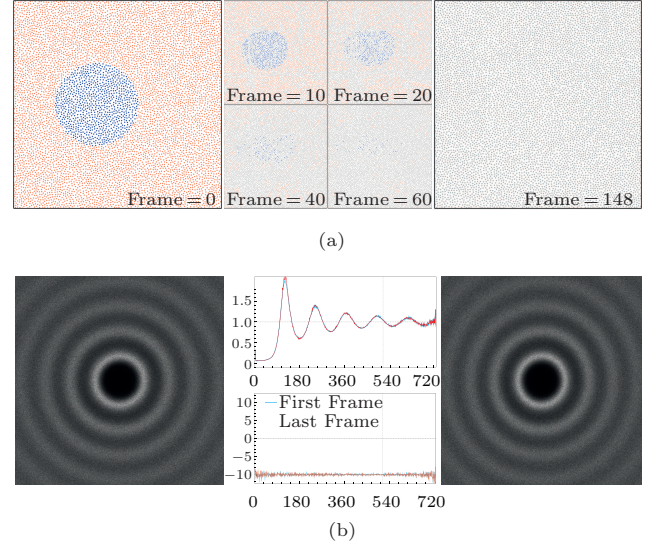

Fig.10. Dynamic point set with Poisson-disk sampling.

4.4 (Re)Meshing

As another new application, we show here that blue-noise sampling can be used directly for high-quality surface remeshing and 2D/3D mesh generation.

Surface Remeshing. Recent work of Ebeida *et al.*[91] and Guo *et al.*[92] showed that the triangulation of a maximal Poisson-disk set has many elegant geometric properties, such as the edge-length bound, the angle bound, which coincide with the theoretical analysis of [93]. Yan and Wonka proposed to use MPS for blue-noise surface remeshing[25]. Yan *et al.*[57] further improved the blue-noise properties by generalizing farthest point optimization for surface remeshing. In this subsection, we compare the remeshing quality of different blue-noise sampling approaches, including CVT[42], CapCVT[52], MPS[25], and FPO[57]. Fig.11 illustrates the remeshing results of the different approaches. The recent papers [25, 57] present more details on the quality comparison between the different methods.

2D/3D Meshing. We slightly modify the sampling framework for mesh generation. The boundary of the input 2D polygon or the 3D mesh is first sampled, and then the interior of the domain is further sampled. Finally, the samples are triangulated with respect to the boundaries. The triangulation can be further optimized using the randomized optimization operators proposed by [25]. Fig.12 and Fig.13 show two examples of 2D and 3D blue-noise mesh generation, respectively.
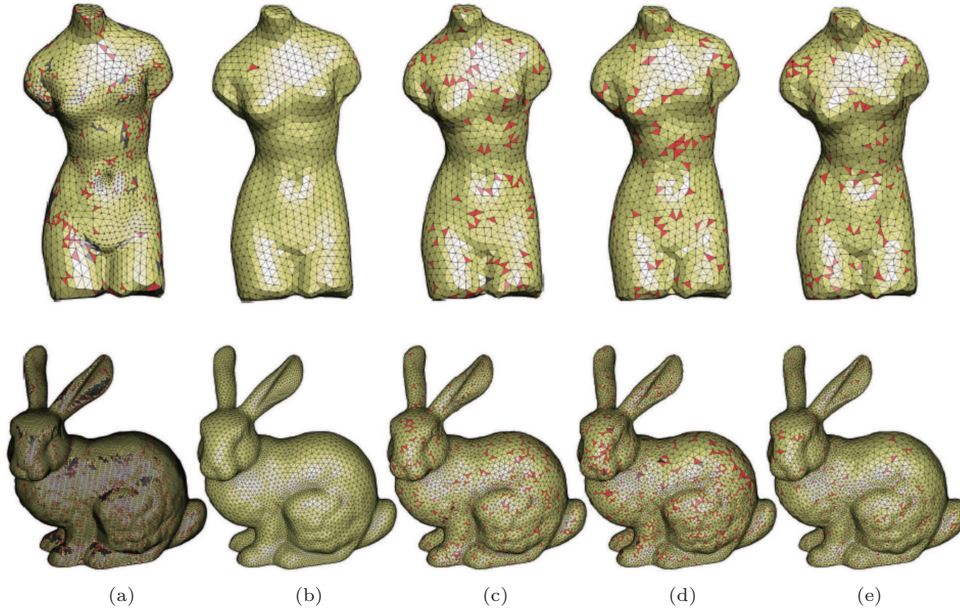
Fig.11. Blue-noise surface remeshing. Top row: uniform remeshing. Bottom row: adaptive remeshing. The result of CVT always has the best meshing quality but lacks blue-noise features, while the other blue-noise remeshing methods are able to generate competitive results. The red triangles have angles larger than $90°$, and the gray triangles have angles smaller than $30°$. (a) Input. (b) CVT. (c) CapCVT. (d) FPO. (e) MPS.

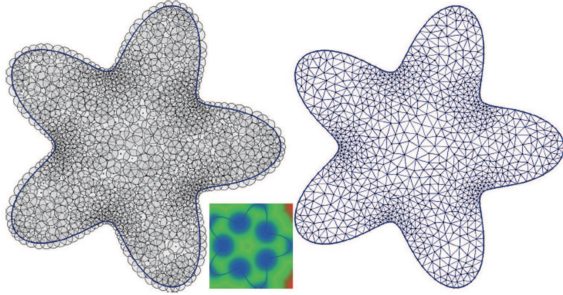Fig.12. 2D meshing with blue-noise sampling[25]. The angle bounds of the triangles are $[30°, 120°]$.

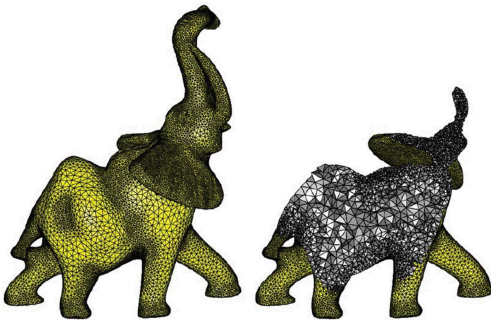
Fig.13. 3D tetrahedral meshing with blue-noise sampling. The dihedral angle bounds of the tetrahedra are $[15.5°, 156.8°]$.

5 Conclusions

This paper reviewed recent work on blue-noise sampling and its related applications. In this section, we briefly summarize these methods and discuss several future research topics.

Poisson-disk sampling is the traditional algorithm for blue-noise sampling and has been studied extensively. Most previous approaches cannot guarantee all three sampling criteria, especially the maximal sampling property, except for the recent studies of [12, 24-25]. It has been shown that the lack of maximal sampling has a drastic influence on the meshing quality[12,25], an important aspect for applications like physical simulations.

Relaxation-based methods are able to generate high-quality point distributions. These methods are most suitable for applications like stippling and remeshing. However, relaxation-based methods are time consuming and not suitable for real-time applications.

Tile-based approaches can generate large-scale point sets in real time, but they sacrifice blue-noise properties, and it is not clear whether this type of method can be used for mesh generation. This is an interesting topic for further study.

As shown in Subsection 4.4, CVT-based remeshing generates meshes with the best meshing quality, whereas other approaches result in meshes with better blue-noise characteristics. The question of which applications need the meshes with such blue-noise properties is worth exploring in the future.

References

[1] Mitchell D P. Generating antialiased images at low sampling densities. In *Proc. the 14th ACM SIGGRAPH*, July 1987, pp.65-72.

[2] Fattal R. Blue-noise point sampling using kernel density model. *ACM Trans. Graphics*, 2011, 30(4): 48:1-48:12

[3] Wu F, Dong W, Kong Y *et al*. Featureaware natural texture synthesis. *The Visual Computer*. (to be appeared)

[4] Deussen O, Hanrahan P, Lintermann B *et al*. Realistic modeling and rendering of plant ecosystems. In *Proc. the 25th ACM SIGGRAPH*, July 1998, pp.275-286.

[5] Schechter H, Bridson R. Ghost SPH for animating water. *ACM Trans. Graphics*, 2012, 31(4): 61:1-61:8.

[6] Lagae A, Dutré P. A comparison of methods for generating Poisson disk distributions. *Computer Graphics Forum*, 2008, 27(1): 114-129.

[7] Tzeng S, Wei L Y. Parallel white noise generation on a GPU via cryptographic hash. In *Proc. the 2008 Symp. Interactive 3D Graphics and Games*, February 2008, pp.79-87.

[8] Condit R, Ashton P S, Baker P *et al*. Spatial patterns in the distribution of tropical tree species. *Science*, 2000, 288(5470): 1414-1418.

[9] Ostling A, Harte J, Green J. Self-similarity and clustering in the spatial distribution of species. *Science*, 2000, 290(5492): 671-671.

[10] Lau D L, Ulichney R, Arce G R. Blue and green noise halftoning models. *IEEE Signal Processing Magazine*, 2003, 20(4): 28-38

[11] Schlömer T, Heck D, Deussen O. Farthest-point optimized point sets with maximized minimum distance. In *Proc. ACM SIGGRAPH Symposium on High Performance Graphics*, August 2011, pp.135-142.

[12] Ebeida M S, Patney A, Mitchell S A *et al*. Efficient maximal Poisson-disk sampling. *ACM Trans. Graphics*, 2011, 30(4): 49:1-49:12.

[13] Ebeida M S, Mitchell S A, Patney A *et al*. A simple algorithm for maximal Poisson-disk sampling in high dimensions. *Computer Graphics Forum*, 2012 31(2): 785-794

[14] Bridson R. Fast Poisson disk sampling in arbitrary dimensions. In *Proc. ACM SIGGRAPH 2007 Sketches*, August 2007, Article No. 22.

[15] Wei L Y. Parallel Poisson disk sampling. *ACM Trans. Graphics*, 2008, 27(3): 20:1-20:9.

[16] Gamito M N, Maddock S C. Accurate multidimensional Poisson-disk sampling. *ACM Trans. Graphics*, 2009, 29(1): 8:1-8:19.

[17] Ebeida M S, Mitchell S A, Patney A *et al*. Exercises in high-dimensional sampling: Maximal Poisson-disk sampling and k-d Darts. In *Topological and Statistical Methods for Complex Data*, Bennett J, Vivodtzev F, Pascucci V(eds.), Springer-Verlag, 2015, pp.221-238.

[18] Cook R L. Stochastic sampling in computer graphics. *ACM Trans. Graphics*, 1986, 5(1): 51-72.

[19] Dunbar D, Humphreys G. A spatial data structure for fast Poisson-disk sample generation. *ACM Trans. Graphics*, 2006, 25(3): 503-508.

[20] White K B, Cline D, Egbert P K. Poisson disk point sets by hierarchical dart throwing. In *Proc. IEEE Symposium on Interactive Ray Tracing*, Sept. 2007, pp.129-132.

[21] Jones T R, Karger D R. Linear-time Poisson-disk patterns. *Journal of Graphics, GPU, and Game Tools*, 2011, 15(3): 177-182.

[22] Wei L Y. Multi-class blue noise sampling *ACM Trans. Graphics*, 2010, 29(4): 79:1-79:8

[23] Ip C Y, Yalçin M A, Luebke D, Varshney A. Pixelpie: Maximal Poisson-disk sampling with rasterization. In *Proc. the 5th High-Performance Graphics Conference*, July 2013, pp.17-26.

[24] Jones T R. Efficient generation of Poisson-disk sampling patterns. *Journal of Graphics Tools*, 2006, 11(2): 27-36.

[25] Yan D M, Wonka P. Gap processing for adaptive maximal Poisson-disk sampling. *ACM Trans. Graphics*, 2013, 32(5): 148:1-148:15.

[26] Mitchell S A, Rand A, Ebeida M S, Bajaj C L. Variable radii Poisson disk sampling. In *Proc. the 24th Canadian Conference on Computational Geometry* (*CCCG*), August 2012, pp.185-190.

[27] Ebeida M S, Mahmoud A H, Awad M A *et al*. Sifted disks. *Computer Graphics Forum*, 2013, 32(2): 509-518.

[28] Yuksel C. Sample elimination for generating Poisson disk sample sets. *Computer Graphics Forum*, 2015, 37(2). (to be appeared)

[29] Cline D, Jeschke S, White K, Razdan A, Wonka P. Dart throwing on surfaces. *Computer Graphics Forum*, 2009, 28(4): 1217-1226

[30] Bowers J, Wang R, Wei L Y, Maletz D. Parallel Poisson disk sampling with spectrum analysis on surfaces. *ACM Trans. Graphics*, 2010, 29(6): 166:1-166:10

[31] Corsini M, Cignoni P, Scopigno R. Efficient and flexible sampling with blue noise properties of triangular meshes. *IEEE Trans. Vis. and Comp. Graphics*, 2012, 18(6): 914-924.

[32] Geng B, Zhang H, Wang H, Wang G. Approximate Poisson disk sampling on mesh. *Science China Information Sciences*, 2013, 56(9): 1-12.

[33] Yan D M, Wonka P. Adaptive maximal Poisson-disk sampling on surfaces. In *Proc. ACM SIGGRAPH Asia Technical Briefs*, Nov. 28–Dec. 1, 2012: 21:1-21:4,.

[34] Yan D M, Bao G B, Zhang X, Wonka P. Low-resolution remeshing using the localized restricted Voronoi diagram. *IEEE Trans. Vis. and Comp. Graphics*, 2014, 20(10): 1418-1427.

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

[36] Guo J, Yan D M, Jia X, Zhang X. Efficient maximal Poisson-disk sampling and remeshing on surfaces. *Computers & Graphics*, 2015, 46: 72-79.

[37] Yan D M, Wallner J, Wonka P. Unbiased sampling and meshing of isosurfaces. *IEEE Trans. Vis. and Comp. Graphics*, 2014, 20(11): 1579-1589.

[38] Fu Y, Zhou B. Direct sampling on surfaces for high quality remeshing. In *Proc. ACM Symposium on Solid and Physical Modeling*, June 2008, pp.115-124.

[39] Ying X, Xin S Q, Sun Q, He Y. An intrinsic algorithm for parallel Poisson disk sampling on arbitrary surfaces. *IEEE Trans. Vis. and Comp. Graphics*, 2013, 19(9): 1425-1437.

[40] Ying X, Li Z, He Y. A parallel algorithm for improving the maximal property of Poisson disk sampling. *Computer-Aided Design*, 2014, 46: 37-44.

[41] Peyrot J L, Payan F, Antonini M. Direct blue noise resampling of meshes of arbitrary topology. *The Visual Computer*, 2014. (to be appeared)

[42] Yan D M, Lévy B, Liu Y *et al.* Isotropic remeshing with fast and exact computation of restricted Voronoi diagram. *Computer Graphics Forum*, 2009, 28(5): 1445-1454.

[43] Lloyd S P. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 1982, 28(2): 129-137.

[44] Du Q, Faber V, Gunzburger M. Centroidal Voronoi tessellations: Applications and algorithms. *SIAM Review*, 1999, 41(4): 637-676.

[45] Alliez P, de Verdière E C, Devillers O, Isenburg M. Centroidal Voronoi diagrams for isotropic surface remeshing. *Graphical Models*, 2005, 67(3): 204-231.

[46] Rong G, Jin M, Shuai L, Guo X. Centroidal Voronoi tessellation in universal covering space of manifold surfaces. *Comp. Aided Geom. Design*, 2011, 28(8): 475-496.

[47] Valette S, Chassery J M, Prost R. Generic remeshing of 3D triangular meshes with metric-dependent discrete Voronoi diagrams. *IEEE Trans. Vis. and Comp. Graphics*, 2008, 14(2): 369-381.

[48] Balzer M Schlömer T, Deussen O. Capacity-constrained point distributions: A variant of Lloyd's method. *ACM Trans. Graphics*, 2009, 28(6): 86:1-86:8.

[49] Li D, Nehab D, Wei L Y, Sander P V, Fu C W. Fast capacity constrained Voronoi tessellation. In *Proc. the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, Feb. 2010, Article No. 13.

[50] Xu Y, Liu L, Gotsman C, Gortler S J. Capacity-constrained Delaunay triangulation for point distributions. *Computers & Graphics*, 2011, 35(3): 510-516.

[51] Xu Y, Hu R, Gotsman C, Liu L. Blue noise sampling of surfaces. *Computers & Graphics*, 2012, 36(4): 232-240.

[52] Chen Z, Yuan Z, Choi Y K, Liu L, Wang W. Variational blue noise sampling. *IEEE Trans. Vis. and Comp. Graphics*, 2012, 18(10): 1784-1796.

[53] Liu Y, Wang W, Lévy B *et al.* On centroidal Voronoi tessellation — Energy smoothness and fast computation. *ACM Trans. Graphics*, 2009, 28(4): 101:1-101:17.

[54] de Goes F, Breeden K, Ostromoukhov V, Desbrun M. Blue noise through optimal transport. *ACM Trans. Graphic*, 2012, 31(6): 171:1-171:11

[55] Kanamori Y, Szego Z, Nishita T. Deterministic blue noise sampling by solving largest empty circle problems. *The Journal of the Institute of Image Electronics Engineers of Japan*, 2011, 40(1): 6-13.

[56] Chen R, Gotsman C. Parallel bluenoise sampling by constrained farthest point optimization. *Computer Graphics Forum*, 2012, 31(5): 1775-1785.

[57] Yan D M, Guo J, Jia X, Zhang X, Wonka P. Blue-noise remeshing with farthest point optimization. *Computer Graphics Forum*, 2014, 33(5): 167-176.

[58] Öztireli A C, Alexa M, Gross M. Spectral sampling of manifolds. *ACM Trans. Graphics*, 2010, 29(6): 168:1-168:8.

[59] Chen J, Ge X, Wei L W *et al.* Bilateral blue noise sampling. *ACM Trans. Graphics*, 2013, 32(6): 216:1-216:11.

[60] Ebeida M S, Awad M A, Ge X *et al.* Improving spatial coverage while preserving the blue noise of point sets. *Computer-Aided Design*, 2014, 46: 25-36.

[61] Hiller S, Deussen O, Keller A. Tiled blue noise samples. In *Proc. the 16th Vision Modeling and Visualization Conference*, October 2001, pp.265-272.

[62] Wang H. Proving theorems by pattern recognition — II. *Bell Systems Technical Journal*, 1961, 40(1): 1-42.

[63] Wang H. Games, logic and computers. *Scientific American*, 1965, 213(5): 98-106.

[64] Lagae A, Dutré P. A procedural object distribution function. *ACM Trans. Graphics*, 2005, 24(4): 1442-1461.

[65] Kopf J, Cohen-Or D, Deussen O, Lischinski D. Recursive Wang tiles for real-time blue noise. *ACM Trans. Graphics*, 2006, 25(3): 509-518.

[66] Ostromoukhov V, Donohue C, Jodoin P M. Fast hierarchical importance sampling with blue noise properties. *ACM Trans. Graphics*, 2004, 23(3): 488-495.

[67] Ostromoukhov V. Sampling with polyominoes. *ACM Trans. Graphics*, 2007, 26(3): 78:1-78:6.

[68] Wachtel F, Pilleboue A, Coeurjolly D *et al.* Fast tile-based adaptive sampling with user-specified fourier spectra. *ACM Trans. Graphics*, 2014, 33(4): 56:1-56:11.

[69] Kalantari N K, Sen P. Fast generation of approximate blue noise point sets. *Computer Graphics Forum*, 2012, 31(4): 1529-1535.

[70] Kalantari N K, Sen P. Efficient computation of blue noise point sets through importance sampling. *Computer Graphics Forum*, 2011, 30(4): 1215-1221.

[71] Zhou Y, Huang H, Wei L Y, Wang R. Point sampling with general noise spectrum. *ACM Trans. Graph.*, 2012, 31(4): 76:1-76:11.

[72] Heck D, Schlömer T, Deussen O. Blue noise sampling with controlled aliasing. *ACM Trans. Graphics*, 2013, 32(3): 25:1-25:12.

[73] Mitchell S A, Mohammed M A, Mahmoud A H, Ebeida M S. Delaunay quadrangulations by two-coloring vertices. In *Proc. the 23rd International Meshing Roundtable*, October 2014, pp.364-376.

[74] Tzeng S, Patney A, Davidson A *et al.* High-quality parallel depth-of-field using line samples. In *Proc. the 4th ACM SIGGRAPH/Eurographics Conference on High-Performance Graphics*, June 2012, pp.23-31.

[75] Sun X, Zhou K, Guo J *et al.* Line segment sampling with blue-noise properties. *ACM Trans. Graphics*, 2013, 32(4): 127:1-127:14.

[76] Ebeida M S, Patney A, Mitchell S A *et al.* k-d Darts: Sampling by k-dimensional flat searches. *ACM Trans. Graphics*, 2014, 33(1): 3:1-3:16.

[77] Feng L, Hotz I, Hamann B, Joy K I. Anisotropic noise samples. *IEEE Trans. Vis. and Comp. Graphics*, 14(2): 342-354, 2008.

[78] Li H, Wei L Y, Sander P, Fu C W. Anisotropic blue noise sampling. *ACM Trans. Graphics*, 2010, 29(6): 167:1-167:12.

[79] Quinn J A, Langbein F C, Lai Y K, Martin R R. Generalized anisotropic stratified surface sampling. *IEEE Trans. Vis. and Comp. Graphics*, 2013, 19(7): 1143-1157.

[80] Ulichney R. Digital Halftoning. Cambridge, USA: MIT Press, 1987.

[81] Schlömer T, Deussen O. Accurate spectral analysis of two-dimensional point sets. *Journal of Graphics, GPU, and Game Tools*, 2011, 15(3): 152-160.

[82] Öztireli A C, Gross M. Analysis and synthesis of point distributions based on pair correlation. *ACM Trans. Graphics*, 2012, 31(6): 170:1-170:10.

[83] Subr K, Kautz J. Fourier analysis of stochastic sampling strategies for assessing bias and variance in integration. *ACM Trans. Graphics*, 2013, 32(4): 128:1-128:12.

[84] Wei L Y, Wang R. Differential domain analysis for non-uniform sampling. *ACM Trans. Graphics*, 2011, 30(4): Article No. 50.

[85] Frey P, Borouchaki H. Surface mesh evaluation. In *Proc. the 6th Int. Meshing Roundtable*, October 1997, pp.363-374.

[86] Cignoni P, Rocchini C, Scopigno R. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, 1998, 17(2): 167-174.

[87] Spencer B, Jones M W. Progressive photon relaxation. *ACM Trans. Graphics*, 2013, 32(1): 7:1-7:11.

[88] Secord A. Weighted Voronoi stippling. In *Proc. the 2nd International Symposium on Non-Photorealistic Animation and Rendering*, June 2002, pp.37-43.

[89] Ascencio-Lopez I, Meruvia-Pastor O, Hidalgo-Silva H. Adaptive incremental stippling using the Poisson-disk distribution. *Journal of Graphics, GPU, and Game Tools*, 2010, 15(1): 29-47.

[90] Ge X, Wei L Y, Wang Y, Wang H. Bilateral blue noise sampling: Additional algorithms and applications. Technical Report, OSU-CISRC-8/13-TR17, The Ohio State University, 2013.

[91] Ebeida M S, Mitchell S A, Davidson A A *et al.* Efficient and good Delaunay meshes from random points. *Computer-Aided Design*, 2011, 43(11): 1506-1515.

[92] Guo J, Yan D M, Bao G, Dong W, Zhang X, Wonka P. Efficient triangulation of Poisson-disk sampled point sets. *The Visual Computer*, 2014, 30(6/7/8): 773-785.

[93] Chew L P. Guaranteed-quality triangular meshes. Technical Report, 89-983, Department of Computer Science, Cornell University, April 1989.
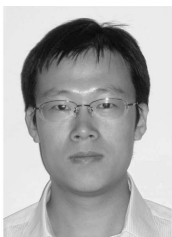
Dong-Ming Yan is a research scientist at King Abdullah University of Science and Technology (KAUST), and he is also an associate professor at National Laboratory of Pattern Recognition (NLPR), the Institute of Automation of the Chinese Academy of Sciences (CASIA), Beijing. He received his Ph.D. degree in computer science from Hong Kong University in 2010, and M.S. and B.S. degrees in computer science and technology from Tsinghua University in 2005 and 2002, respectively. His research interests include computer graphics, geometric processing and visualization.
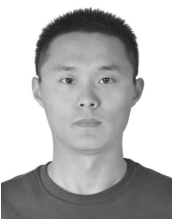
Jian-Wei Guo received his B.S. degree from Shandong University, Jinan, in 2011, and he is currently a Ph.D. candidate in NLPR, CASIA. His research interests include 3D shape analysis and geometry processing.

Bin Wang is an associate professor of School of Software, Tsinghua University, Beijing. He received his B.S. degree in chemistry in 1999, and Ph.D. degree in computer science from Tsinghua University in 2005. He was a research assistant at Department of Computer Science, Hong Kong University, and had postdoctoral research training at ISA/ALICE Research Group, INRIA-LORIA, France. His research interests include photorealistic rendering, non-photorealistic rendering, and image and video processing.

Xiao-Peng Zhang is a professor in National Laboratory of Pattern Recognition at Institute of Automation, Chinese Academic of Sciences (CAS), Beijing. He received his Ph.D. degree in computer science from Institute of Software, CAS, in 1999. He received the National Scientific and Technological Progress Prize of China (second class) in 2004. His main research interests include computer graphics and image processing.

Peter Wonka received his Ph.D. degree in computer science and M.S. degree in urban planning from the Technical University of Vienna, Vienna, Austria, in 2001 and 2002, respectively. He was a postdoctoral researcher with the Georgia Institute of Technology, Atlanta, GA, USA, for two years. He is currently a professor with the Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia, and also an associate professor with Arizona State University, Tempe, USA. His research interests include topics in computer graphics, visualization, computer vision, remote sensing, image processing, and machine learning.