Bilateral Blue Noise Sampling

Xiaoyin Ge* Li-Yi Wei† Yusu Wang*

*The Ohio State University †The University of Hong Kong

Abstract

Blue noise sampling is a core component for a variety of graphics applications including rendering, imaging, modeling, and animation. Prior blue noise methods consider mostly spatial domain properties such as density and anisotropy. This might not be sufficient for non-spatial properties such as image or geometry features.

We present bilateral blue noise sampling, a method that considers both spatial and feature properties. Our core idea is a sample distance measure that incorporates both sample positions and domain features. This idea, inspired by bilateral filtering, is very simple and can be easily combined with prior sampling algorithms. We describe corresponding synthesis methods for sample generation and analysis methods for distribution quality. We demonstrate the efficacy of bilateral sampling for a variety of applications including geometry sampling, image stippling, and dynamic domains.

Keywords: bilateral, blue noise, sampling, rendering, imaging, geometry, animation

1 Introduction

Sampling is a fundamental component for a variety of computational tasks. The basic goal is to represent a target domain in a certain best way under a given budget of samples. The notion of "best" is often scenario dependent, and can favor different properties. For example, some applications need to place enough samples at prominent regions to ensure accurate reproduction of image or geometric features [Lévy and Liu 2010; Li and Mould 2011], while others prefer samples distributed in a uniform random blue-noise fashion at smooth image regions for reasons perceptual (e.g. texturing [Li et al. 2010], animation [Yu et al. 2009], or stippling [Balzer et al. 2009; Fattal 2011]) or numerical (e.g. anti-aliasing [Cook 1986], reaction diffusion [Turk 1992], or crack simulation [Ebeida et al. 2011]). Some applications may favor both feature-preserving and blue-noise properties, e.g. half-toning [Pang et al. 2008; Chang et al. 2009] or geometry processing [Öztireli et al. 2010]. However, despite such needs, there is a lack of general methods that can simultaneously preserve both features and blue noise properties for different applications.

In particular, even though adaptive sampling allows samples to be distributed with density proportional to the importance of domain regions, this is not the same as feature preservation. For one, importance sampling is likely to miss a sharp or thin feature with near zero area, even if a very high importance value is given. Also, if the importance changes dramatically as can happen in images and geometry models, samples in less importance areas may squeeze out nearby samples over sharp features. It is desirable to have a feature-preserving blue-noise sampling method without these problems. See Figure 2, 9 & 11 for comparisons.

We present bilateral blue-noise sampling, a general method that can produce sample distributions preserving both features and blue noise properties within the same target domain. Our basic idea is a sample distance measure that incorporates both spatial positions and domain features. This idea is inspired by bilateral filtering [Tomasi and Manduchi 1998], a very simple and elegant technique that has found widespread applications in various graphics, vision, and image processing tasks (see e.g., [Paris et al. 2009] and references within). Our idea shares similar simplicity and generality and can be easily combined with prior sample analysis and synthesis algorithms. We describe both synthesis methods for sample generation and analysis methods for gauging distribution quality.

Figure 1: *Bilateral blue noise sampling. Our method preserves both features and blue noise properties, with applications in geometry sampling, dynamic domains, and image stippling.*

We demonstrate scenarios that can benefit from bilateral sampling versus traditional pure blue-noise or feature-preserving techniques, as well as applications in geometry sampling, image stippling, and dynamic domains (Figure 1).

2 Previous Work

Bilateral filtering Bilateral filtering is a simple and yet very effective method for feature-preserving smoothing, and has been applied over a variety of tasks for images and meshes [Tomasi and Manduchi 1998; Jones et al. 2003; Paris et al. 2009]. Our method is inspired by these methods but targets sampling instead of filtering, and is able to preserve features better than prior blue noise methods.

Blue noise sampling Blue noise sampling places samples in a random and yet uniform fashion and offers unique advantages such as visually pleasantries in the spatial domain, replacing aliasing by high frequency noise in the spectrum domain, and robustness for certain numerical computations [Lloyd 1983; Dippé and Wold 1985; Cook 1986; Mitchell 1987; Turk 1992; Glassner 1994; Alliez et al. 2002; Dutre et al. 2002; Pharr and Humphreys 2004; Ostromoukhov et al. 2004; Kopf et al. 2006; Ostromoukhov 2007; Fu and Zhou 2008; Balzer et al. 2009; Wei 2010; Öztireli et al. 2010; Fattal 2011; Ebeida et al. 2011; Kalantari and Sen 2011; Schlömer et al. 2011]. However, most blue noise sampling methods are not suitable for feature preservation, even under importance sampling (Figure 2). Some recent methods start to pay attention to feature preservation and can produce impressive results (e.g. [Pang et al. 2008; Chang et al. 2009; Li et al. 2010; Kalantari and Sen 2011]), but these are either restricted to specific domains and/or applications (e.g. regular grid for halftoning [Pang et al. 2008; Chang et al. 2009], thin image features through maximal point set + min-conflict metric [Kalantari and Sen 2011]), or feature preservations at the expense of blue noise properties [Li et al. 2010; Li and Mould 2011]).

Feature aware sampling Many sampling methods have been designed to preserve features for various applications such as stippling [Kim et al. 2008] and meshing [Lévy and Liu 2010]. However, they usually do not preserve blue noise properties. The spectral sampling in [Öztireli et al. 2010] is a notable exception in that it also attempts to keep some blue noise properties as a by-product

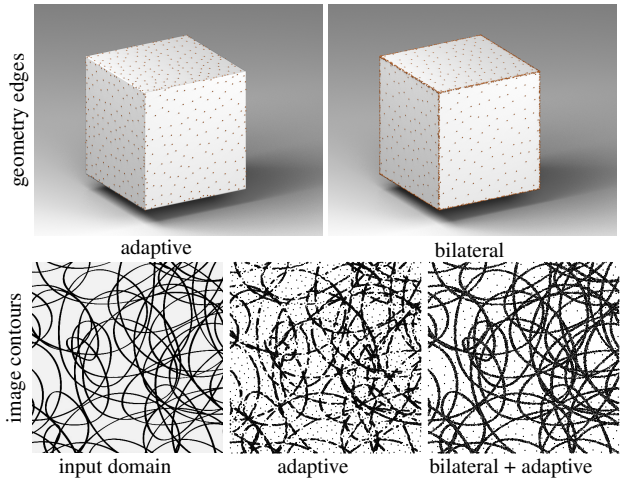


Figure 2: *Comparisons between adaptive and bilateral sampling. Shown here are sampling results for geometry and images. Results within each row are produced with similar number of samples (∼2800/17900 for geometry/image). Notice that our method preserves features better (e.g. geometry edges, image contours).*

of feature-preservation. However, as shown in their Figure 9 and our Section 6.2, the blue noise properties are not as well preserved. Furthermore, most of these methods are designed for one particular application instead of being a general methodology as ours.

3 Bilateral Sampling

Goal The goal is to sample a given target domain so that the sample set exhibits both blue noise properties and preserves application specific features (e.g. geometric ridges or image contours). The former can be analyzed through a unified approach (e.g. power spectrum or spatial uniformity [Wei and Wang 2011]) while the latter via application dependent measures, such as surface reconstruction for geometry sampling, visual perception for image stippling, or temporal coherence for dynamic motions.

Idea Our basic idea is to measure the distance d between any two samples s and s' via the following formula:

$$\begin{aligned}
\mathbf{d}_\mathbf{p}(s,s') &= \frac{\mathbf{p}(s)-\mathbf{p}(s')}{\sigma_p},\ d_p(s,s') = \left|\mathbf{d}_\mathbf{p}(s,s')\right| \\
\mathbf{d}_\mathbf{v}(s,s') &= \frac{\mathbf{v}(s)-\mathbf{v}(s')}{\sigma_v},\ d_v(s,s') = \left|\mathbf{d}_\mathbf{v}(s,s')\right| \\
\mathbf{d}(s,s') &= \left(\mathbf{d}_\mathbf{p}(s,s'), \mathbf{d}_\mathbf{v}(s,s')\right) \\
d(s,s') &= \left|\mathbf{d}(s,s')\right| = \sqrt{d_p^2(s,s') + d_v^2(s,s')} \qquad (1)
\end{aligned}$$

, where $\mathbf{p}$ indicates sample position, $\mathbf{v}$ the application dependent sample "feature", $d_p \backslash d_v$ the spatial\feature distance, and $\sigma_p \backslash \sigma_v$ the relative weights. The $\mathbf{p}$ and $\mathbf{v}$ parts are our sampling analogy to the *domain* and *range* parts of bilateral filtering. Specifically:

d_p This is essentially the spatial sample distance in traditional blue noise sampling, either Euclidean (e.g. image sampling) or Riemannian (e.g. surface sampling).

d_v This is the application-specific feature distance. For example, in geometry sampling, we can have $\mathbf{v} = \mathbf{n}$, the surface normal. In image stippling, we can have $\mathbf{v} = \mathbf{c}$, the pixel colors.

We will use Equation 1 for both synthesis and analysis. In particular, for synthesis, our goal is to distribute samples within the given domain Ω so that they are uniform under the distance measure in Equation 1. For analysis, we will use Equation 1 for various quality measures, including spatial measures like minimum spacing ρ [Lagae and Dutré 2008] and spectrum measures like differential domain analysis (DDA) [Wei and Wang 2011]. Note that Equation 1 is the only key difference between our method and prior sampling algorithms. Since the sample distance measure is mostly orthogonal to the specific analysis or synthesis algorithms, our method can be easily incorporated with these by simply replacing their position-only distance with our bilateral distance.

However, even though this basic idea is simple, it can bring significant benefits for many different applications such as geometry sampling, image stippling, and dynamic domains, as will be detailed in Section 6. In particular, all these can be done by simply using different features in our formulation, instead of having to design different algorithms for different applications. Using this bilateral distance for sampling does incur additional implementation and algorithm innovations, as will be described in the rest of the paper.

By tuning the relative values of σ_p and σ_v, we can control the relative degrees of blue noise and feature preservation. When $\sigma_v = \infty$, our method reduces to traditional blue noise sampling. With smaller $\frac{\sigma_v}{\sigma_p}$ ratio, our method puts more emphasis on features and less on blue noise. Since only the relative values of σ_v to σ_p matters, we usually set $\sigma_p = 1$. Note that Equation 1 bears strong similarity to bilateral image filtering [Tomasi and Manduchi 1998] where our $\mathbf{v}$ part encourages samples to be placed at dominant features.

Our method is not the first to consider both positions and features in sampling, much like [Tomasi and Manduchi 1998] is not the first filtering method to consider both. Arguably many sampling algorithms have done this in a certain form (e.g. considering radiance [Pharr and Humphreys 2004] or geometry [Öztireli et al. 2010] as features). However, just like [Tomasi and Manduchi 1998] that provides a very simple formulation for bilateral filtering, to our best knowledge ours is the first one that provides a simple and direct formulation for bilateral sampling, especially for blue noise.

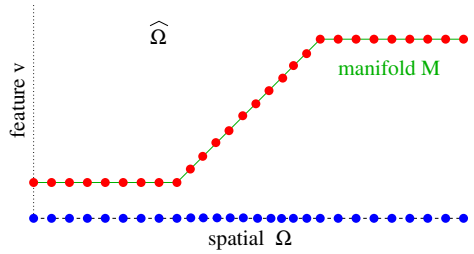


Figure 3: *Manifold interpretation. Given a spatial domain Ω (horizontal axis) with features $\mathbf{v}$ (vertical axis), if we perform a uniform sampling (red) in the corresponding manifold M (green) embedded in a higher dimensional space, the corresponding samples manifested in Ω (blue) can have non-uniform distributions induced by the features.*

Interpretation We can interpret Equation 1 as measuring distances in a higher dimensional space $\widehat{\Omega}$ with samples located within an embedded lower dimensional manifold M. (See Figure 3 for illustration. Similar interpretations have also appeared in prior bilateral filtering papers, e.g. [Chen et al. 2007].) For example, Equation 1 measures 6D distances for samples distributed over a 2-manifold with 3D surface normal directions as features, and 3D distances for samples distributed over a 2D image with 1D gray scale colors as features. Under the simplest case of uniform analysis/synthesis with $\mathbf{v}$ as the only source of non-uniformity, we are essentially performing a uniform sampling of M for synthesis, and similarly gauging the distribution uniformity over M during analysis. [1]

[1] This interpretation assumes a geodesic measure on M for both d_p and d_v. However, the latter is actually Euclidean in our current implementation. Thus the statements are only approximate but become more accurate under denser sampling rate.

Specifically, let φ^{-1} indicates the mapping from Ω to M (and its inverse, φ, can be considered as a projection that rids $\mathbf{v}$):

$$\varphi^{-1}(s) = (\frac{\mathbf{p}(s)}{\sigma_p}, \frac{\mathbf{v}(s)}{\sigma_v}) \tag{2}$$

. Without further non-uniformity, the distribution of $\varphi^{-1}(S)$ over M is uniform even though the sample set S might not be uniform over the original domain Ω due to $\mathbf{v}$.

Approximation Performing a direct sampling over a high dimensional manifold M, even though possible, can be daunting both algorithmically and computationally. We can avoid this and keep all computations in the original domain Ω through the anisotropic approximations described in [Li et al. 2010]. Intuitively, we absorb all anisotropy introduced by features $\mathbf{v}$ between Ω and M into local Jacobian matrices J, and use these as local linear approximations for conducting analysis or synthesis on Ω. We compute J following the approach of [Li et al. 2010], treating Equation 1 as a global map/warp φ from a high dimensional $(\frac{\mathbf{p}}{\sigma_p}, \frac{\mathbf{v}}{\sigma_v})$ (uniform) domain Ω' to the $\mathbf{p}$ only (non-uniform anisotropic) domain Ω (Equation 2).

For domain Ω , the corresponding Jacobian J can be derived as follows. For notational simplicity, let

$$\hat{\mathbf{p}} = \frac{\mathbf{p}}{\sigma_p},\ \hat{\mathbf{v}} = \frac{\mathbf{v}}{\sigma_v} \tag{3}$$

. We have

$$J(\varphi^{-1}) = \begin{pmatrix} J_{\mathbf{p}}(\hat{\mathbf{p}}) \\ J_{\mathbf{p}}(\hat{\mathbf{v}}) \end{pmatrix} \tag{4}$$

, where $J_{\mathbf{x}}(\mathbf{y})$ indicates the jacobian of $\mathbf{y}$ relative to $\mathbf{x}$. Note that $J_{\mathbf{p}}(\hat{\mathbf{p}}) = \frac{1}{\sigma_p} I$ (with I indicating the identity matrix) only if the domain Ω is (spatially) Euclidean.

For clarify, in the following we treat σ_p and σ_v as global constants; the derivations can be easily extended for varying σ_p and σ_v.

4 Synthesis

Our method can be easily applied to prior sampling methods; all one needs to do is to replace the traditional position-only distance measures in prior algorithms with our bilateral one described in Section 3. Below, we provide concrete examples for two popular algorithms, dart throwing and relaxation.

4.1 Dart Throwing

Our method can be easily applied for dart throwing [Dippé and Wold 1985; Cook 1986], where samples are produced stochastically subject to the constraint that no two samples s and s' can be closer to each other than a pre-determined distance threshold $r(s, s')$. All we need to do is to plug in our distance measure d from Equation 1 in lieu of the Euclidean distance in traditional dart throwing. By using different $r(s, s')$ representations, our method can be easily and orthogonally applied for various sampling scenarios, including uniform (r is a constant), isotropic ($r(s, s') = \frac{r(s)+r(s')}{2}$ where $r(s)$ indicates the diameter of a (hypothetical) disk centered at s), and anisotropic (an accurate directional function $r(s, s')$ or the Jacobian approximation in [Li et al. 2010]).

4.2 Relaxation

Lloyd relaxation [Lloyd 1983] is another classical method that has been applied to generating blue noise samples. Unlike dart throwing which generates samples from scratch, relaxation starts from a given sample distribution and gradually improves its uniformity. Let S be a set of samples (or "sites" in the jargon of [Balzer et al. 2009]) whose distribution we wish to optimize for. The uniformity of S can be measured by the following energy function:

$$\mathbf{E}(S, \mathcal{V}) = \sum_i \int_{s' \in V_i} d^2(s', s_i) ds' \tag{5}$$

, where $\mathcal{V}$ is the Voronoi tessellation generated from S, V_i the Voronoi region corresponding to site $s_i \in S$, s' a point in the domain Ω. The major difference here is that we are using our bilateral distance measure for d in Equation 1 instead of a pure spatial domain distance. Lloyd relaxation minimizes this energy function by iterating between the following two steps, Voronoi and centroid, until sufficient convergence:

Voronoi For each point $s' \in \Omega$, find the site $s(s')$ that is the closest to s' among all sites in S:

$$s(s') = \arg\min_{s \in S} d^2(s', s) \tag{6}$$

. Our bilateral distance can be directly plugged in here without changing the underlying search algorithm.

Centroid Move each site $s_i \in S$ to the centroid m_i of the corresponding Voronoi region $V_i \in \mathcal{V}$ to minimize the corresponding energy term:

$$\int_{s' \in V_i} d^2(s', s_i) ds' \tag{7}$$

. For traditional (Euclidean) spatial domain distance measure d, this is a quadratic energy term and can be minimized by setting m_i to the (Euclidean) centroid of V_i.

Unfortunately, due to the presence of features d_v in our d, we do not have a corresponding analytically minimizable solution. One possibility is to follow our manifold interpretation in Section 3, and compute the centroids over an embedded manifold M in a high dimensional space (e.g. a 2D manifold in a 6D space for 3D surfaces with 3D normal) following the methodology in [Du et al. 2002], but this can be a quite complex process. Fortunately, under the usual case of sufficient sampling rate, we can adopt the Jacobian approximation as described in [Li et al. 2010] as follows:

$$m_i = \left(\int_{V_i} J^{\mathrm{T}} J(s') ds' \right)^{-1} \int_{V_i} J^{\mathrm{T}} J(s') s' ds' \tag{8}$$

, where J is the Jacobian computed via Equation 4.

Note that even though our centroid step is only approximate, it works well when the sampling density is sufficient (relative to domain variations), as discussed in [Li et al. 2010]. Also, unlike [Li et al. 2010] whose Voronoi step is also approximate, our Voronoi step uses exact d without any approximation. This is usually a good tradeoff, as the Voronoi step tends to involve longer spaced site/point pairs than the centroid step, especially during the earlier stages of the iterations.

Non-uniformity The descriptions above assume uniform sampling aside from the feature $\mathbf{v}$ part. For additional non-uniformity, such as adaptive or anisotropic sampling for the $\mathbf{p}$ and/or $\mathbf{v}$ part, we can incorporate the additional $r(.)$ (for local isotropic adaptivity as in [Wei 2008]) or $\dot{J}$ (for local anisotropy as in [Li et al. 2010], a square matrix with dimension equals to the dimension of $\mathbf{p}$ + $\mathbf{v}$, different from J in Equation 4) information on top of our bilaterally derived d and J above. Specifically, for isotropic sampling over the $\mathbf{p}$ + $\mathbf{v}$ manifold (Section 3), we multiply both $d(s, s')$ in Equation 1 (for the Voronoi step) and $J(s)$ from Equation 4 (for the centroid step) by $\frac{1}{r(s)}$. Similarly, for anisotropic sampling, we multiply both $\mathbf{d}(s, s')$ and $J(s)$ by $\dot{J}(s)$.

4.3 Discussion

As described above, the application of our bilateral distance in Equation 1 to prior sampling algorithms can benefit from a Jacobian approximation as in Equation 4. A similar Jacobian approximation is also adopted in [Li et al. 2010].

However, despite this similarity, the two methods differ fundamentally in several important aspects:

Concept The basic idea of [Li et al. 2010], as illustrated in their Equation 1, is to transform everything into spatial anisotropy (encoded in their square $J^{\mathrm{T}}J$ matrix) and subsequently consider only the resulting anisotropic spatial domain during sampling. In contrast, our Equation 1 directly considers both spatial and feature domains without any such intermediation.

Computation The fundamental conceptual difference above has important consequences in the application to various sampling algorithms: [Li et al. 2010] has to rely on the Jacobian approximation everywhere, whereas our method can use a direct and accurate distance measure. In particular, their distance measure needs to be computed approximately both in dart-throwing and relaxation. In contrast, we used Jacobian *only* in the centroid step of Lloyd relaxation for computational convenience (to avoid high dimensionality). Both our dart throwing and the Voronoi step of relaxation do not use any approximation.

In sum, our method uses [Li et al. 2010] as part of the machinery, but is not an extension of it.

5 Analysis

We analyze sample distributions through a variety of criteria, including both qualitative visual comparisons as well as quantitative measures, including spatial uniformity ρ [Lagae and Dutré 2008] and differential-domain spectrums (DDA) [Wei and Wang 2011] for blue noise, and Hausdorff distance H for geometry feature preservation. Some of these methods can be applied directly (e.g. H), while others (e.g. ρ and DDA) need to incorporate our bilateral distance measure (Equation 1). Below we provide a concrete example for [Wei and Wang 2011], a general method for analyzing non-uniform distributions using spatial statistics with direct connection to Fourier spectrums.

Exact computation In general, sample distributions produced by our methods in Section 4 can be anisotropic due to the presence of the feature term in Equation 1, even though the domain Ω itself is uniform without considering the features. Conceptually, we can follow the manifold interpretation in Section 3 and perform exact analysis by *warping* every sample s in the original domain Ω into the higher dimensional M (Equation 2), and perform all kinds of spatial and spectrum analysis there. For example, we can use the following formula for the differential domain analysis in [Wei and Wang 2011]

$$\chi(\mathbf{d} = s - s') = \varphi^{-1}(s) - \varphi^{-1}(s') \tag{9}$$

, where $\mathbf{d}$ is the differential between samples s and s' in Ω, χ the differential transformation, and $\chi(\mathbf{d})$ the transformed $\mathbf{d}$ in $\widehat{\Omega}$.

However, in order to do the above, we will need to define a global orientation field over M (analogous to the analysis of surface sampling in [Wei and Wang 2011]). This is doable for lower dimensional M (e.g. for stippling gray scale images for which M is a 2-manifold embedded in a 3D $\widehat{\Omega}$), it is not clear to us how to compute such orientation fields for higher dimensional cases (e.g. a 2-manifold M embedded in a 6D $\widehat{\Omega}$ for sampling 3D surfaces). For such cases, we resort to approximations described below.

Approximation We can avoid directly dealing with higher dimensional manifolds M and keep all analysis computations in the original domain Ω using the anisotropic analysis method in [Wei and Wang 2011] with the Jacobian approximation described in Section 3. However, the Jacobian in Equation 4 above might not be square due to the presence of $\mathbf{v}$. This can cause issues for analysis methods that require square Jacobians such as [Wei and Wang 2011] which needs to preserve the dimension of $\mathbf{d}$ after χ. We can address this following the approach described in the extended version of [Wei and Wang 2011] based on the simple observation: since the distance measure in [Li et al. 2010] depends on only $J^{\mathrm{T}}J$, not J itself, all we need is to derive a square J' so that

$$J^{\mathrm{T}}J = J'^{\mathrm{T}}J' \tag{10}$$

. This can be achieved by the standard matrix square root method:

$$J^{\mathrm{T}}J = V^{\mathrm{T}}DV \tag{11}$$

, where V is an orthonormal matrix and D a diagonal matrix. Note that since $J^{\mathrm{T}}J$ is positive definite, D will contain only non-negative diagonal elements. Thus, we have

$$J' = V^{\mathrm{T}}\sqrt{D}V \tag{12}$$

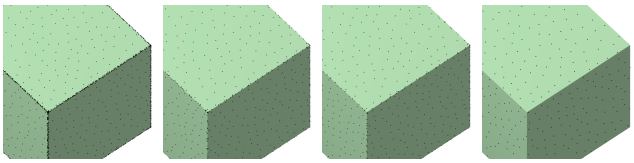
. Following Equation 15 of [Wei and Wang 2011] for anisotropic sampling, we have

$$\chi(\mathbf{d} = s - s') = \frac{1}{E(\lambda)}\left(\frac{J'^{-1}(s) + J'^{-1}(s')}{2}\right)^{-1}(s - s')^{\mathrm{T}} \tag{13}$$

, where $E(\lambda)$ is the mean of the eigenvalues of $J'(.)$ over Ω. Notice the use of J' instead of J allows us to compute different domain spectrum with the same dimensionality as the sample space Ω.

6 Results

Here, we present our results with applications in geometry sampling (Section 6.2), image stippling (Section 6.3), and dynamic domains (Section 6.4).

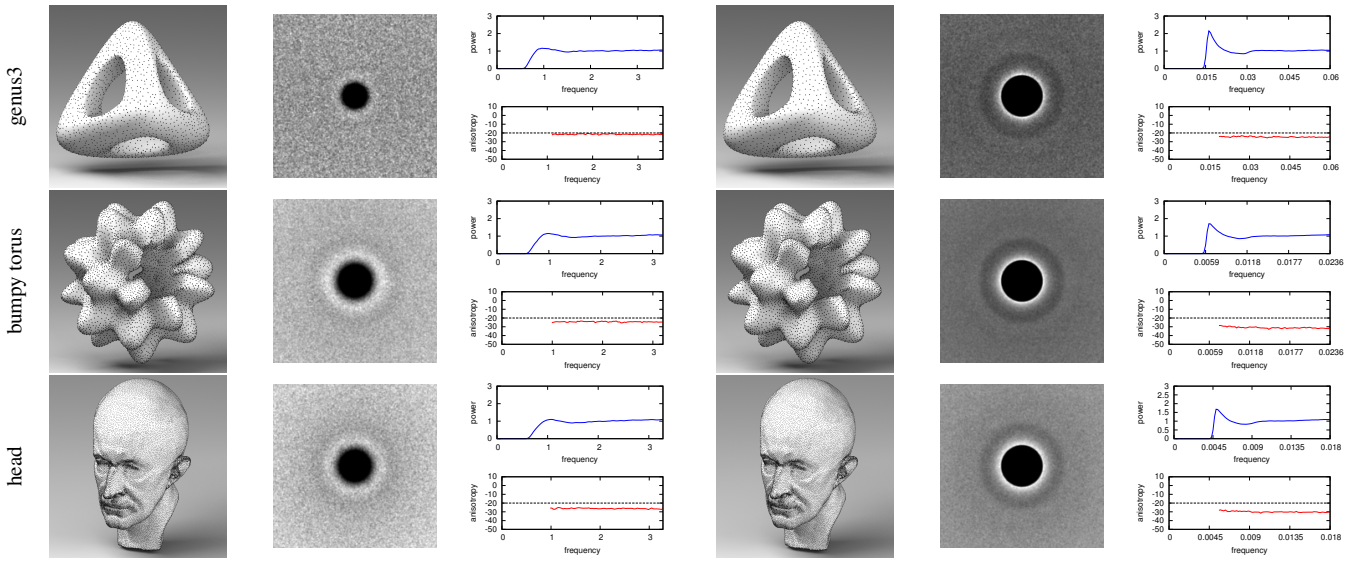


$\sigma_v = 2.0$	$\sigma_v = 9.0$	$\sigma_v = 15.0$	$\sigma_v \to \infty$
$H = 0.93$	$H = 2.01$	$H = 2.51$	$H = 13.14$

Figure 4: *The effect of σ_v. The smaller the σ_v relative to σ_p (which is 1 by default), the more samples are placed near features (such as cube edges and corners). Setting $\sigma_v = \infty$ will reduce our method to traditional blue noise sampling. We also provide Hausdorff distances H (in units of 10^{-4}) of the reconstructed surfaces to the original cube for additional feature measurements (details in Section 6.2).*

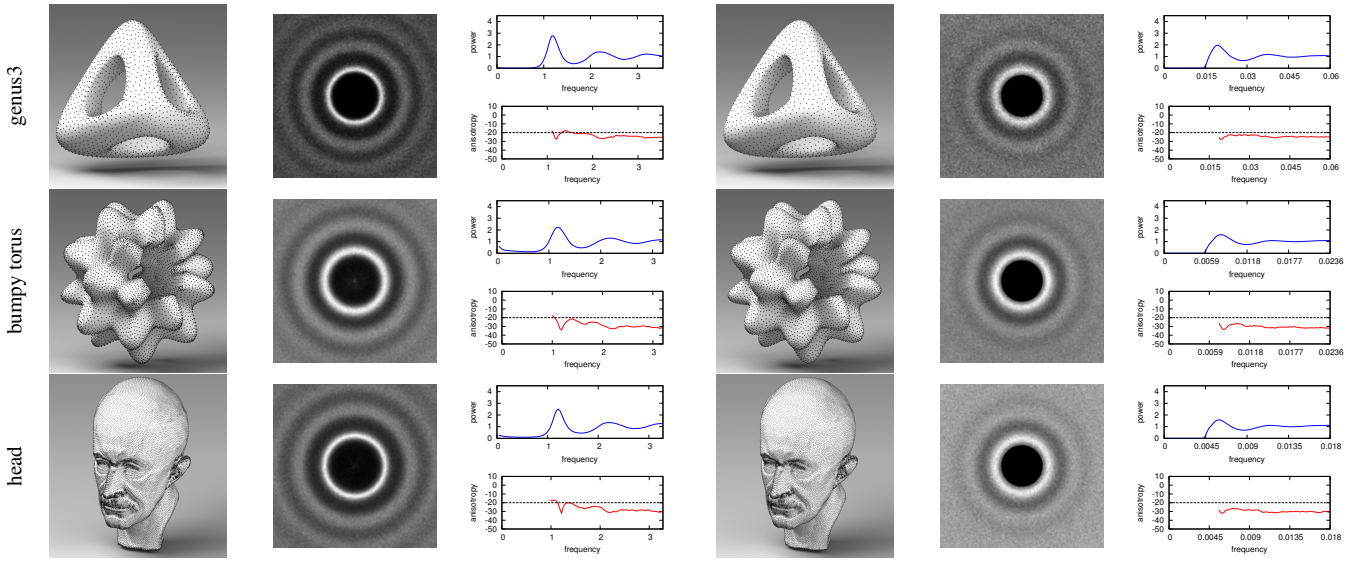
6.1 Parameters

σ_v and σ_p The main parameters our method are σ_p and σ_v. We usually set $\sigma_p = 1$ so that our method would reduce to traditional (non-bilateral) blue noise sampling with $\sigma_v = \infty$. σ_v needs to be low enough to preserve features while high enough to maintain blue noise properties (see Figure 4). Even though it is entirely possible to pick custom values depending on particular application needs, through extensive experiments with different domains (both analytical and discrete), we have found that a good range is [7 11], with a good default value 9 which we have used for all our results (unless stated otherwise).



[Öztireli et al. 2010] subsampling dart throwing

Figure 5: *Blue noise properties of geometry sampling via dart throwing. Shown here are sampling results of different domains with varying topological and geometrical properties. Detailed statistics for each case can be found in Table 1. Shown in each group are the spatial samples, DDA spectrum, and its radial mean and anisotropy [Wei and Wang 2011]. Notice the more uniform distributions of our results (larger inner ring of the DDA spectrums).*



[Öztireli et al. 2010] gradient ascent relaxation

Figure 6: *Blue noise properties of geometry sampling via relaxation/gradient-ascent following the results from Figure 5. We run 10 iterations of CCVT [Balzer et al. 2009] for relaxation. Notice the lower amount of anisotropy in our results.*

Complexity Our method simply increases the dimension of samples from n_p to $n_p + n_v$, where n_p and n_v are the dimensions for spatial positions and features. This information can be easily plugged into various analysis and synthesis algorithms to derive speed and storage complexity. For example, the speed of dart throwing and the Voronoi step of relaxation will be slowed by a rough constant factor of $1 + \frac{n_v}{n_p}$, and the centroid step of relaxation roughly $(1 + \frac{n_v}{n_p})^2$ (squared due to the Jacobian). The exact performance depends on particular applications, but as a ballpark number, our current implementation generates about tens of thousands samples per second on a single commodity PC core.

6.2 Geometry Sampling

Geometry sampling is important for graphics and simulation, and can benefit from sample distributions that preserve both features and blue noise properties [Öztireli et al. 2010]. In particular, allocating samples at features such as tips and creases are needed to preserve the original geometry, while distributing samples with blue noise properties can avoid undesirable aliasing or bias in computations involving surface samples [Turk 1991; Turk 1992; Fu and Zhou 2008]. Our method can be of help here by simply using surface normal $\mathbf{n}$ as the feature $\mathbf{v}$ in our basic formulation in Section 3.

We have applied our method to sample a variety of models with different geometric and topological properties, and analyzed both their blue noise and feature preservation properties. For the former

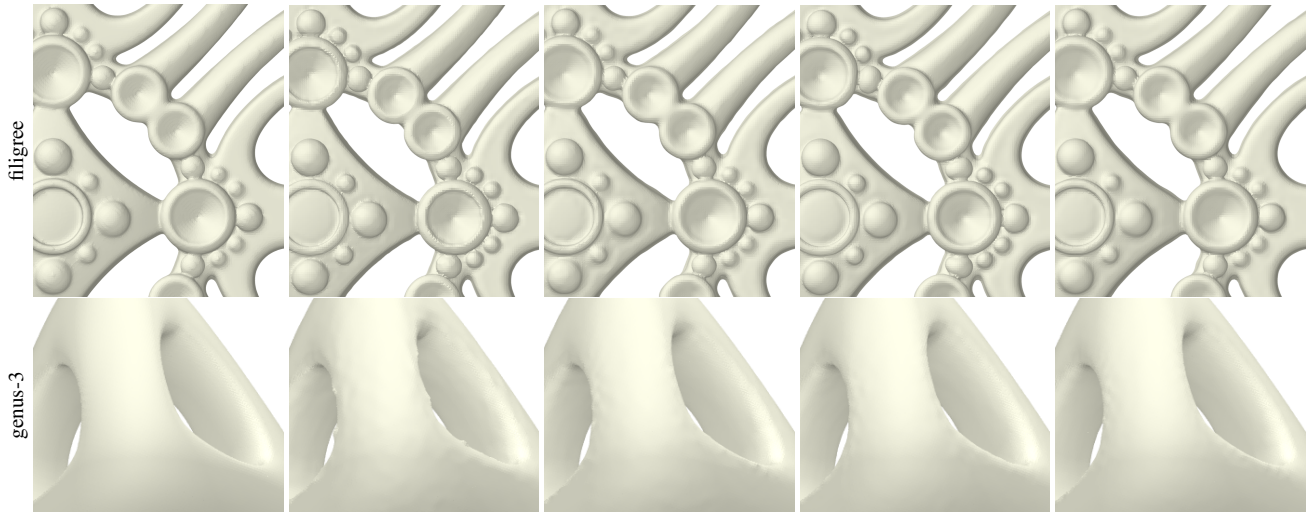


Figure 7: *Visual quality comparison. Here, we display the surface reconstruction results from down-sample sets generated by different methods (surfaces are reconstructed using the same PSS algorithm as in [Öztireli et al. 2010]). The filigree results (top row) are produced with about 84K samples each, where the input contains roughly 514K points. (These results are produced under the same setting as Figure 11 of [Öztireli et al. 2010]).) The genus-3 results are from Figure 5 and 6, where the input contains 440K points while our algorithm subsamples it to about 2.89K points. Notice the better reconstruction quality of our results (dart throwing and relaxation) versus [Öztireli et al. 2010] (subsampling and gradient ascent). The filigree contains regularly distributed samples while genus-3 less regular white noise samples, providing different stress tests.*

case	# samp	ρ				$H(10^{-5})$			
		d	s	r	g	d	s	r	g
genus-3	2.89K	0.75	0.50	0.75	0.43	59.39	64.42	58.27	66.01
bumpy	18.65K	0.75	0.48	0.75	0.003	17.54	22.52	17.30	22.77
head	32.08K	0.75	0.54	0.75	0.01	8.41	9.50	8.7	8.19

Table 1: *Statistics for geometry sampling results in Figure 5 & 6. The measures include ρ (the larger the better up to ~ 0.85) for spatial uniformity and Hausdorff distance H (the smaller the better) for feature preservation. (All surface areas are normalized to 1 for comparable H computation.) d\s\r\g represents dart-throwing\subsampling\relaxation\gradient-ascent, respectively. Each case is computed by averaging 10 sets except for "head" for which only 3 sets are used. σ_n/σ_p = 14 for head, and 9 otherwise. Note that ρ for [Öztireli et al. 2010] can be very small.*

we measure the differential domain spectrum [Wei and Wang 2011] and ρ [Lagae and Dutré 2008] directly over the original surfaces, while for the latter we first reconstruct surfaces (via Tight-Cocone [Dey and Goswami 2003]) from the samples followed by Hausdorff distance H and qualitative comparison. As shown in Figure 5, Figure 6, Table 1 and Figure 7, our method can preserve both features and blue noise properties. Note that in these figures and tables, we are comparing our method with a state-of-the-art feature-preserving geometry sampling method proposed in [Öztireli et al. 2010], which we will describe in more detail next.

Spectral manifold sampling Here we compare our method with [Öztireli et al. 2010], a state of art geometry sampling method that aims primarily at feature preservation but also retains some blue noise properties. [Öztireli et al. 2010] performs manifolds sampling based on a feature measure $\tilde{\mu}$ derived from the current output sample set $S = \{s_i\}$:

$$\tilde{\mu}(s) = 1 - \mathbf{k}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{k}/k(s,s) \tag{14}$$

, where $k(s, s')$ is a Gaussian kernel measuring the distances between two samples s and s' considering both position and normal similar to our bilateral distance, $\mathbf{k}$ a vector with component $\mathbf{k}_i = k(s, s_i)$, and $\mathbf{K}$ a matrix with component $\mathbf{K}_{ij} = k(s_i, s_j)$. $\tilde{\mu}(s)$ is a quantity between 0 and 1; intuitively, the higher the value, the more "feature" s represents relative to S.

[Öztireli et al. 2010] provides two sampling algorithms based on $\tilde{\mu}$: (1) randomized linear scan (referred to as subsampling in the figures and tables), which sub-samples an input point set by sequentially picking samples with $\tilde{\mu}$ greater than a certain threshold ϵ, and (2) iterative gradient ascent, which optimizes the location of a sample set by maximizing the individual $\tilde{\mu}$ values locally. The sub-sampling and gradient ascent parts are analogous to our bilateral versions of dart throwing and relaxation, respectively.

Compared to [Öztireli et al. 2010], our method is much simpler and easier to implement. Our method also provides explicit controls for both feature preservation and blue noise, and can be easily hooked up with prior sampling methods (e.g. dart throwing and relaxation). Mostly importantly, our method preserves both features and blue noise properties better.

The corresponding results are shown in Table 1 and Figure 5 & 6. Notice that our method has comparable numerical measures for feature preservation (H in Table 1) and better blue noise properties (higher ρ in general, more uniform spectrum profiles in Figure 5, and less anisotropy in Figure 6). We have set the parameters for both their sub-sampling and gradient ascent methods to be compatible with our dart throwing and relaxation parts. In particular, we pick their kernel sizes according to our r, and number of gradient ascent iterations to be identical to our relaxation.

In Figure 7 we provide a qualitative visual comparison for reconstruction quality from sample sets produced by [Öztireli et al. 2010] and our method. As shown, for models with regular initial sampling (e.g. filigree) our dart throwing slightly outperforms their subsampling while our relaxation performs similarly to their gradient ascent; however for models with less regular initial sampling (e.g. genus-3) both our methods perform noticeably better.

6.3 Image Stippling

Stippling refers to techniques that use small primitives (e.g. dots) to illustrate images [Secord 2002; Balzer et al. 2009; Li et al. 2010; Fattal 2011]. The primitives are usually of the same color (e.g. black) or from a small palette of colors [Wei 2010]. Since human visual systems tend to blend multiple dots in local spatial regions, stippling with limited colors can still faithfully reproduce continuous image tones. Such trick for trading off spatial for color resolu-

Figure 8: *Image stippling results. We compare our method against [Li and Mould 2011], a state of art stippling method that considers structure, blue noise, and tone reproduction. As shown, our results have competitive tones and structures (e.g. better boundary between the chin and background in the lower left corner of old man face) but better blue noise properties (e.g. more uniform sample distributions).*

tions has also been taken advantage of in image halftoning where samples lie on discrete regular pixel grids [Pang et al. 2008; Chang et al. 2009; Li and Mould 2011].

For both stippling (continuous domain sample location) and halftoning (discrete domain sample location) applications, it is well known that sample sets with blue noise properties are more visually pleasing. In addition to blue noise, it could also be desirable to maintain image structures or features [Pang et al. 2008; Chang et al. 2009; Li and Mould 2011].

However, to our knowledge, prior methods that consider both blue noise and features have certain limitations. They may sacrifice blue noise properties to preserve features (e.g. contrast aware halftoning [Li and Mould 2010; Li and Mould 2011]), they might not offer flexible enough controls to tune the relative weights of blue noise and feature preservation (e.g. [Chang et al. 2009; Li and Mould 2011]), and most of them operate in discrete grids (e.g. [Pang et al. 2008; Chang et al. 2009]) and might not be suitable for continuous domain applications such as stippling. ([Li and Mould 2011] demonstrated impressive stippling results via error diffusion; however, sufficiently large neighborhoods have to be used. See [Wei 2010] for relevant discussions about other potential issues for producing continuous domain effects via discrete domain sampling.)

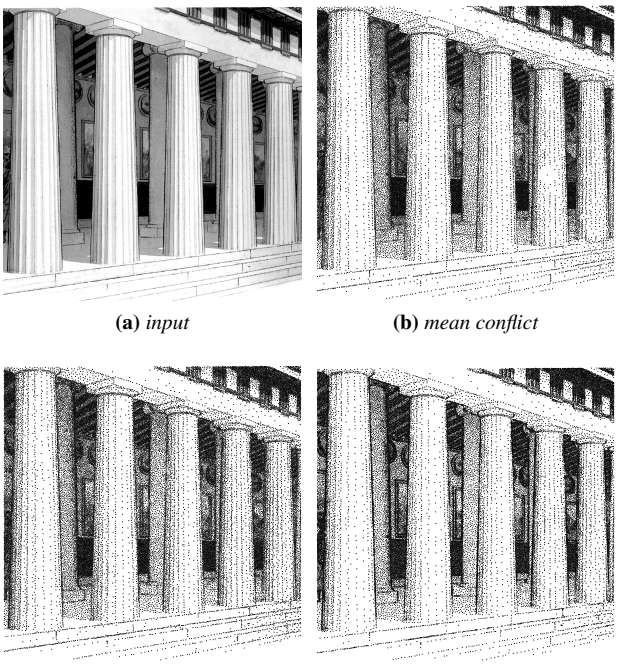
Our method can be applied for such feature-aware blue noise image stippling or halftoning by simply using gray-scale image color $\mathbf{c}$ as features $\mathbf{v}$ in Equation 1. It is applicable to both discrete and continuous domains, and very easy to combine with prior blue noise stippling algorithms [Balzer et al. 2009; Li et al. 2010; Fattal 2011] by simply plugging the distance measure in Equation 1. As shown in Figure 2 and Figure 8, our method preserves features and blue-noise-properties better than prior techniques.

Note that the only difference between traditional stippling and our method is the presence of the feature term in Equation 1, where the former can be considered as a special case with $\sigma_v = \infty$. In particular, our bilateral distance measure in Equation 1 is orthogonal with other aspects of blue noise sampling, such as using intensity as the importance field to define local $r(.)$ fields.

To preserve tones, we set our stipple radius proportional to $\frac{d_p(s)}{d(s)}$, i.e. the ratio of the local average distance at sample s computed by traditional position-only method and our bilateral distance in Equation 1. (We compute this through the ratio of the eigen-values of the Jacobian matrices from d and d_p.) Users can also optionally cap the ratio (and thus the corresponding d) of the smallest to largest possible stipple sizes for additional control (e.g. artistic reasons). We have found that uniform stipple size adequate for many cases, but variable stipple size beneficial for sufficiently complex images.

Figure 9 compares our bilateral distance with other conflict metrics such as mean-conflict [Wei 2008] and min-conflict [Kalantari and Sen 2011] for dart throwing. As shown, our result has the best

quality, especially around features. This is a further confirmation (beyond Figure 2) that our method is better than traditional adaptive sampling for simultaneous blue noise and feature preservations.



(a) *input* **(b)** *mean conflict*

(c) *min conflict* **(d)** *bilateral*

Figure 9: *Distance metric comparison. Here, we compare our bilateral distance metric with alternative measures including mean [Wei 2008] and min [Kalantari and Sen 2011] conflicts. Each case contains about 35K samples produced by dart throwing. Notice that mean conflict metric can produce noticeable knock-out effects (e.g. lower-middle of the second right column) and a max conflict metric can produce even worse results (not shown here but see [Kalantari and Sen 2011] Figure 8). Min-conflict metric produces less knock-out effects, but still preserves features less well than our bilateral method (e.g. compare the boundaries of the rectangular paintings and circular objects on the wall behind the columns).*

6.4 Dynamic Distribution

Blue noise sampling has been applied for a variety of applications involving dynamic phenomena, such as sprite-based animation [Yu et al. 2009], video stippling [Chen et al. 2012], and hybrid point distribution [Vanderhaeghe et al. 2007]. The key, as stated in [Vanderhaeghe et al. 2007], is to maintain the balance between 2D blue noise uniformity, 2D temporal coherence, and 2D (video) or 3D (object) motion depiction. This is a challenging problem due to multiple spatial and temporal constraints. To our best knowledge, prior dynamic blue noise sampling methods have yet to consider domain features when placing samples. Our method can benefit these applications to help bring out the features. And similar to other applications introduced earlier, our method can be easily and orthogonally combined with these prior dynamic blue noise sampling algorithms by simply plugging our bilateral distance measure (Equation 1).

Here, we describe two particular dynamic point distribution applications: cross sampling for 2D stylization of dynamic 3D objects and spatial-temporal sampling for video stylization. Both are described in [Vanderhaeghe et al. 2007]. The basic idea is to perform blue noise sampling for the first frame, advect the samples according to the scene motions (e.g. 3D object motions or 2D video optical flows), and maintain blue noise properties by removing and adding samples from crowded and sparse regions, respectively.

Feature $\mathbf{v}$ For video stylization, we simply use colors $\mathbf{c}$ as $\mathbf{v}$. For cross-dimensional sampling (2D stylization of 3D objects), we define the $\mathbf{v}(s)$ for each 2D screen space sample s as a combination of the 2D screen space shading $\mathbf{c}(s)$ and the 3D object space normal $\mathbf{n}(s')$:

$$s' = raycast(s),\ \mathbf{v}(s) = \big(\alpha\mathbf{c}(s), \mathbf{n}(s')\big) \tag{15}$$

, where α is the relative weight between $\mathbf{c}$ and $\mathbf{n}$, s' the 3D object surface point corresponding to s (obtained through ray casting from the eye point), and $\mathbf{n}$ the 3D object normal in the eye coordinate system. (If $raycast(s)$ does not hit the object surface, we set $\mathbf{n}(s')$ to $\mathbf{0}$.) This choice allows us to emphasize both shading and projected geometric features better than only 3D geometry (Section 6.2) or 2D image (Section 6.3) features. Some common examples include interior and exterior silhouettes (e.g. the genus 3 model in Figure 10) and shallow ridges that might not be very prominent on the original 3D geometry but can be enhanced due to projection and shading (e.g. the claw model in Figure 10).

Cross dimensional sampling Figure 10 provides examples for applying our method for cross dimensional sampling, i.e. placing stipples on a 2D plane to render dynamic 3D objects. There, we compare our cross sampling method against our bilateral sampling in either the screen space or object space only, as well as cross dimensional (non-bilateral) blue noise sampling [Vanderhaeghe et al. 2007]. As shown, our cross-sampling method provides better quality than these alternatives, including tone reproduction, features in both image and object spaces, flatness (screen space blue noise uniformity), motion depiction, and temporal coherence. (The latter three are desiderata from dynamic stylization [Bénard et al. 2011] while the first two from static image stippling.) For tone reproduction, we set the local radius $r(s)$ according to the shading value.

This application can be considered as a hybrid of our other two applications, geometry sampling (Section 6.2) and image stippling (Section 6.3) in that we place samples on a 2D plane similar to the latter while preserving features considering the former, but with two main differences: (1) the 3D features are sampled after projection; thus, we can produce more uniform screen-space blue noise samples (uniform blue noise samples over a 3D object surface can become highly non-uniform after projection, e.g. near the silhouettes of the genus-3 model in Figure 10) and better preserve features that might not be very prominent in the 3D geometry but can get enhanced after projection (e.g. the shallow ridge in the claw model in Figure 10), and (2) we need to take into account dynamic effects such as object motions, and thus provide better motion illustrations than 2D sampling which can produce temporal incoherence or shower-door artifacts (as demonstrated in [Vanderhaeghe et al. 2007] and our video).

Video stylization Figure 11 demonstrates our spatial-temporal sampling application for video stylization. This can be considered as a generalization of stippling static images to dynamic videos with the need to consider motion depiction and temporal coherence similar to the cross sampling application. However, unlike cross sampling 3D objects, the motions are computed through video optical flow instead of projection tracking (as in Equation 15). Furthermore, in addition to gray scale color or shading as used in our previous applications, here we also demonstrate the application of our method for color videos. This can be easily achieved in our method by using 3-channel RGB instead of 1-channel intensity in our $\mathbf{c}$ feature. As shown in Figure 11, our method preserves features better than non-bilateral blue noise sampling [Vanderhaeghe et al. 2007] while maintains its other advantages including motion depiction, temporal coherence, and screen space blue noise.

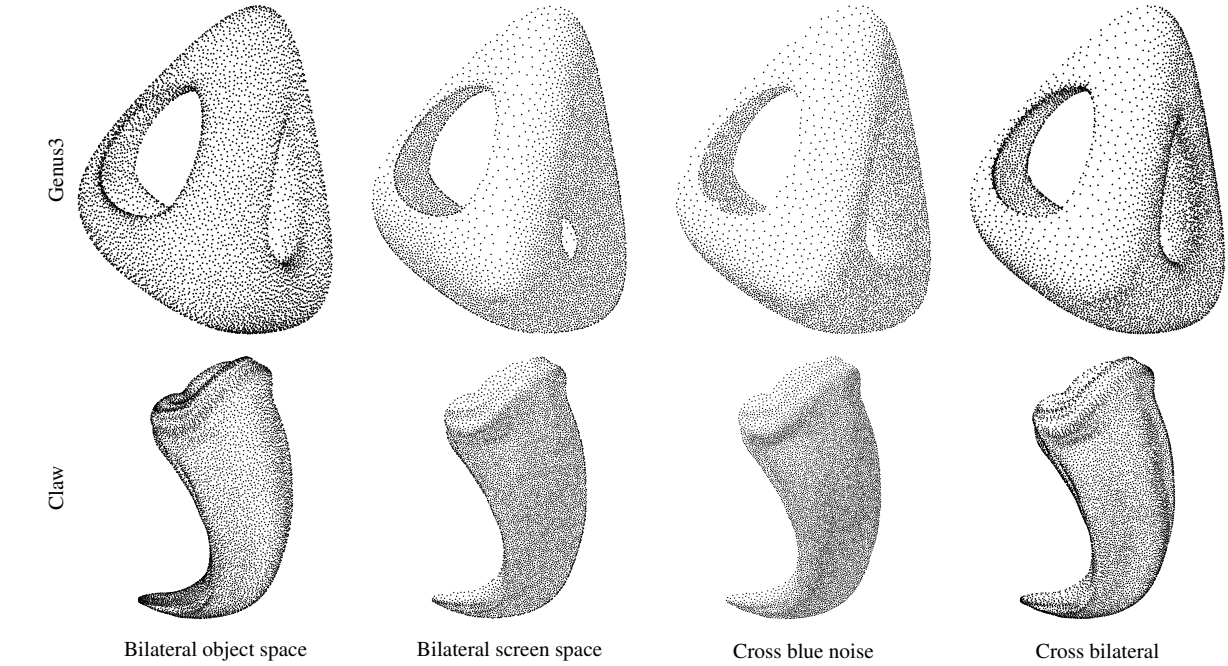


Figure 10: *Cross dimensional dynamic sampling for 2D stylization of 3D objects. Here we compare our cross dimensional bilateral blue noise sampling against alternative methods, including bilateral sampling in either object- or screen-space only (as described in Section 6.2 & 6.3, respectively), as well as cross dimensional (non-bilateral) blue noise sampling [Vanderhaeghe et al. 2007]. Notice that our method provides better effects, including shading/tone reproduction, features in image and object spaces, as well as temporal coherence. Compared to object space sampling, our cross bilateral sampling has more uniform screen space distribution (notice highly distorted and non-uniform sample distributions near the silhouettes of the genus-3 model for pure object space sampling) and illustrates clearer those geometric features that are not very prominent on the original 3D model but can get enhanced due to screen projection, such as the shallow ridge in the middle right of the claw model. Compared to screen space sampling, our method does not suffer from temporal coherence issues such as shower door or temporal discontinuity (see video). Compared to cross dimensional (non-bilateral) blue noise sampling, our method preserves features better. Each case of genus3\claw contains ∼ 4500\8000 visible samples. The input models are shown on the left. Please refer to the accompany video for animation effects.*

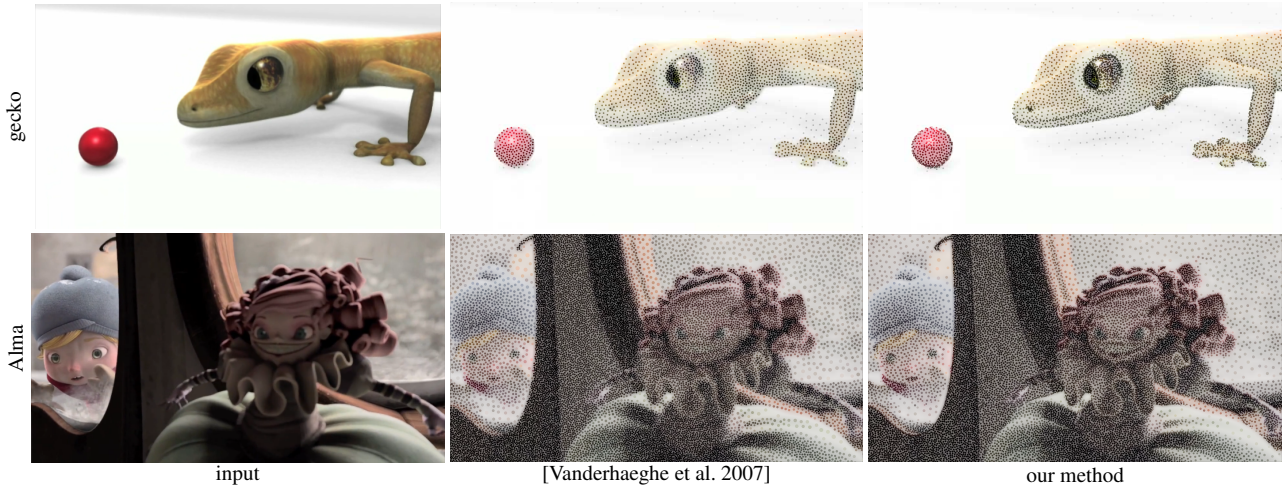


Figure 11: *Spatial-temporal sampling for video stylization. Notice the better feature preservation of our method, such as the gecko mouth and eye, and the doll face and clothing. Each frame of gecko\Alma contains ∼ 2200\19000 samples. Please refer to the accompany video for animation effects.*

7 Limitations and Future Work

Currently we determine the parameters σ_p and σ_v empirically and would like to pursue more rigorous analytical methods. In addition, all our current results are produced with uniform σ_p and σ_v, and we believe varying them spatially may benefit certain applications.

We still rely on Jacobian approximations for various synthesis (e.g. centroid step for relaxation) and analysis (e.g. DDA [Wei and Wang 2011]) algorithms. It would be great to devise precise solutions to cleanly use our bilateral formulation in Equation 1.

Our current analysis method is derived from [Wei and Wang 2011]. We are not aware of better alternatives for analyzing non-uniform sample distributions, but [Wei and Wang 2011] is not perfect either. In particular, we will have to share distance measure and approximations for some of the synthesis and analysis parts. A potential future work is to develop analysis methods that are standalone and independent of any sample synthesis algorithms.

In our formulation we have been using either Euclidean or Rieman-

nian positional distance d_p, but only Euclidean feature distance d_v. Even though this suffices so far, we wonder if there are scenarios that can benefit from a non-Euclidean d_v.

We have not yet attempted to combine our bilateral sample distance with a maximal sampling method such as [Cline et al. 2009; Gamito and Maddock 2009; Ebeida et al. 2011; Kalantari and Sen 2011]. Such combination should be doable, and could preserve features even better.

Our current implementation adopted prior acceleration methods based on spatial measures only, such as grids for dart throwing [Wei 2008]. This is too conservative, as our bilateral distance is never smaller than its spatial component. Further refinements in acceleration methods will help the performance of bilateral sampling.

Finally, even though we have applied bilateral sampling only to blue noise, the idea could be extended for more general sampling applications.

Acknowledgements We would like to thank Hua Li for answering questions about [Li and Mould 2011], and anonymous SIGGRAPH reviewers for their valuable comments.

References

ALLIEZ, P., MEYER, M., AND DESBRUN, M. 2002. Interactive geometry remeshing. In *SIGGRAPH '02*, 347–354.

BALZER, M., SCHLOMER, T., AND DEUSSEN, O. 2009. Capacity-constrained point distributions: A variant of lloyd's method. In *SIGGRAPH '09*, 86:1–8.

BÉNARD, P., BOUSSEAU, A., AND THOLLOT, J. 2011. State-of-the-Art Report on Temporal Coherence for Stylized Animations. *Computer Graphics Forum 30*, 8, 2367–2386.

CHANG, J., ALAIN, B., AND OSTROMOUKHOV, V. 2009. Structure-aware error diffusion. In *SIGGRAPH Asia '09*, 162:1–8.

CHEN, J., PARIS, S., AND DURAND, F. 2007. Real-time edge-aware image processing with the bilateral grid. In *SIGGRAPH '07*.

CHEN, Z., YUAN, Z., CHOI, Y.-K., LIU, L., AND WANG, W. 2012. Variational blue noise sampling. *TVCG*.

CLINE, D., JESCHKE, S., RAZDAN, A., WHITE, K., AND WONKA, P. 2009. Dart throwing on surfaces. In *EGSR '09*, 1217–1226.

COOK, R. L. 1986. Stochastic sampling in computer graphics. *ACM Trans. Graph. 5*, 1, 51–72.

DEY, T. K., AND GOSWAMI, S. 2003. Tight Cocone: A water tight surface reconstructor. In *Proc. 8th ACM Sympos. Solid Modeling Appl.*, 127–134.

DIPPÉ, M. A. Z., AND WOLD, E. H. 1985. Antialiasing through stochastic sampling. In *SIGGRAPH '85*, 69–78.

DU, Q., GUNZBURGER, M. D., AND JU, L. 2002. Constrained centroidal voronoi tessellations for surfaces. *SIAM J. Sci. Comput. 24* (May), 1488–1506.

DUTRE, P., BALA, K., AND BEKAERT, P. 2002. *Advanced Global Illumination*. A. K. Peters, Ltd., Natick, MA, USA.

EBEIDA, M. S., PATNEY, A., MITCHELL, S. A., DAVIDSON, A., KNUPP, P. M., AND OWENS, J. D. 2011. Efficient maximal poisson-disk sampling. In *SIGGRAPH '11*, 49:1–12.

FATTAL, R. 2011. Blue-noise point sampling using kernel density model. In *SIGGRAPH '11*, 48:1–12.

FU, Y., AND ZHOU, B. 2008. Direct sampling on surfaces for high quality remeshing. In *SPM '08*, 115–124.

GAMITO, M. N., AND MADDOCK, S. C. 2009. Accurate multi-dimensional poisson-disk sampling. *ACM Trans. Graph. 29*, 1, 1–19.

GLASSNER, A. S. 1994. *Principles of Digital Image Synthesis*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

JONES, T. R., DURAND, F., AND DESBRUN, M. 2003. Non-iterative, feature-preserving mesh smoothing. In *SIGGRAPH '03*, 943–949.

KALANTARI, N. K., AND SEN, P. 2011. Efficient computation of blue noise point sets through importance sampling. *Computer Graphics Forum (EGSR '11) 30*, 4, 1215–1221.

KIM, D., SON, M., LEE, Y., KANG, H., AND LEE, S. 2008. Feature-guided image stippling. In *EGSR'08*.

KOPF, J., COHEN-OR, D., DEUSSEN, O., AND LISCHINSKI, D. 2006. Recursive wang tiles for real-time blue noise. In *SIGGRAPH '06*, 509–518.

LAGAE, A., AND DUTRÉ, P. 2008. A comparison of methods for generating Poisson disk distributions. *Computer Graphics Forum 21*, 1, 114–129.

LÉVY, B., AND LIU, Y. 2010. Lp centroidal voronoi tessellation and its applications. In *SIGGRAPH '10*, 119:1–11.

LI, H., AND MOULD, D. 2010. Contrast-aware halftoning. *Comput. Graph. Forum*, 273–280.

LI, H., AND MOULD, D. 2011. Structure-preserving stippling by priority-based error diffusion. In *GI '11*, 127–134.

LI, H., WEI, L.-Y., SANDER, P., AND FU, C.-W. 2010. Anisotropic blue noise sampling. In *SIGGRAPH Asia '10*, 167:1–12.

LLOYD, S. 1983. An optimization approach to relaxation labeling algorithms. *Image and Vision Computing 1*, 2.

MITCHELL, D. P. 1987. Generating antialiased images at low sampling densities. In *SIGGRAPH '87*, 65–72.

OSTROMOUKHOV, V., DONOHUE, C., AND JODOIN, P.-M. 2004. Fast hierarchical importance sampling with blue noise properties. In *SIGGRAPH '04*, 488–495.

OSTROMOUKHOV, V. 2007. Sampling with polyominoes. In *SIGGRAPH '07*, 78.

ÖZTIRELI, A. C., ALEXA, M., AND GROSS, M. 2010. Spectral sampling of manifolds. In *SIGGRAPH ASIA '10*, 168:1–8.

PANG, W.-M., QU, Y., WONG, T.-T., COHEN-OR, D., AND HENG, P.-A. 2008. Structure-aware halftoning. In *SIGGRAPH '08*, 89:1–8.

PARIS, S., KORNPROBST, P., TUMBLIN, J., AND DURAND, F. 2009. Bilateral filtering: Theory and applications. In *Foundations and Trends in Computer Graphics and Vision*.

PHARR, M., AND HUMPHREYS, G. 2004. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc.

SCHLÖMER, T., HECK, D., AND DEUSSEN, O. 2011. Farthest-point optimized point sets with maximized minimum distance. In *HPG '11*, 135–142.

SECORD, A. 2002. Weighted Voronoi stippling. In *NPAR '02*, 37–43.

TOMASI, C., AND MANDUCHI, R. 1998. Bilateral filtering for gray and color images. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, 839–.

TURK, G. 1991. Generating textures on arbitrary surfaces using reaction-diffusion. In *SIGGRAPH '91*, 289–298.

TURK, G. 1992. Re-tiling polygonal surfaces. In *SIGGRAPH '92*, 55–64.

VANDERHAEGHE, D., BARLA, P., THOLLOT, J., AND SILLION, F. 2007. Dynamic point distribution for stroke-based rendering. In *EGSR '07*, 139–146.

WEI, L.-Y., AND WANG, R. 2011. Differential domain analysis for non-uniform sampling. In *SIGGRAPH '11*, 50:1–10.

WEI, L.-Y. 2008. Parallel Poisson disk sampling. In *SIGGRAPH*

'08, 20:1–9.

WEI, L.-Y. 2010. Multi-class blue noise sampling. In *SIGGRAPH '10*, 79:1–8.

YU, Q., NEYRET, F., BRUNETON, E., AND HOLZSCHUCH, N. 2009. Scalable real-time animation of rivers. *Computer Graphics Forum (Proceedings of Eurographics 2009) 28*, 2.

Supplementary Materials

A Jacobian

Here we show that the approximation in Equation 4 will be infinitely closer to Equation 1 under sufficient sampling rates and accurate computation of all derivatives.

Let s and s' are nearby samples, denote:

$$\delta\mathbf{p}(s) = \mathbf{p}(s') - \mathbf{p}(s) \tag{16}$$

$$\delta\mathbf{v}(s) = \mathbf{v}(s') - \mathbf{v}(s) \tag{17}$$

. First, following Equation 1, we have:

$$\delta\mathbf{d}(s) = \left(\frac{\delta\mathbf{p}(s)}{\sigma_p} \; \frac{\delta\mathbf{v}(s)}{\sigma_v}\right) \tag{18}$$

. Then, following Equation 4, we have

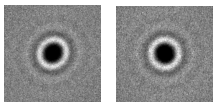
$$J\delta\mathbf{p}(s) = \left(\frac{\delta\mathbf{p}(s)}{\sigma_p} \; \frac{\delta\mathbf{v}(s)}{\sigma_v}\right) \tag{19}$$

. Note that Equation 18 and Equation 19 produce the same results under the assumption that $\delta\mathbf{p}$ is small, thus:

$$\delta\mathbf{d} = J\delta\mathbf{p} \tag{20}$$

.

B Derivatives

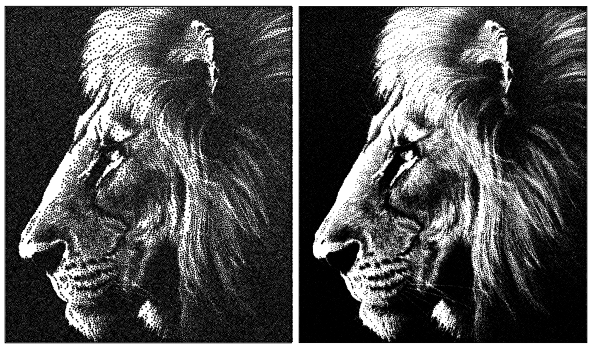


(a) *wrong* **(b)** *right*

Figure 12: *Derivative comparison.*

A core component of our algorithms is computing derivatives. Care has to be taken to ensure they are accurately estimated to avoid artificial distortions such as anisotropically warped different domain spectrum results (see Figure 12).

The basic idea is to fit a smooth surface over the feature field before taking derivatives, for robustness reasons. In our experiments, we have found it suffice to fit a locally quadratic function $\tilde{\mathbf{v}}(\mathbf{p})$ of sample positions $\mathbf{p}$. A first order (linear) function may have discontinuous derivatives, while higher order functions (3 or more) are more expensive to compute without visible quality improvements. To make sure the fitting is robust (over instead of under constrained), we use a local neighborhood of size 3-5 rings for regularly sampled Euclidean domains (e.g. height fields).



(a) *uniform stipple size* **(b)** *variable stipple size*

Figure 13: *Stippling with variable dot sizes. Notice the better structure and tone reproduction of (b) versus (a). The source image is shown on the left.*

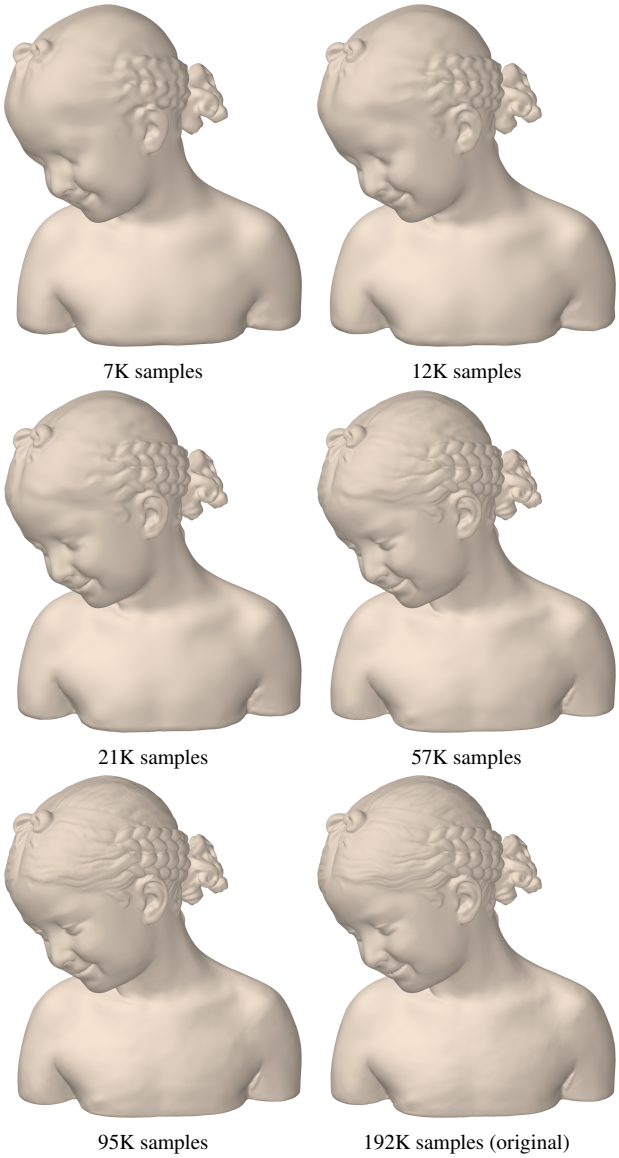
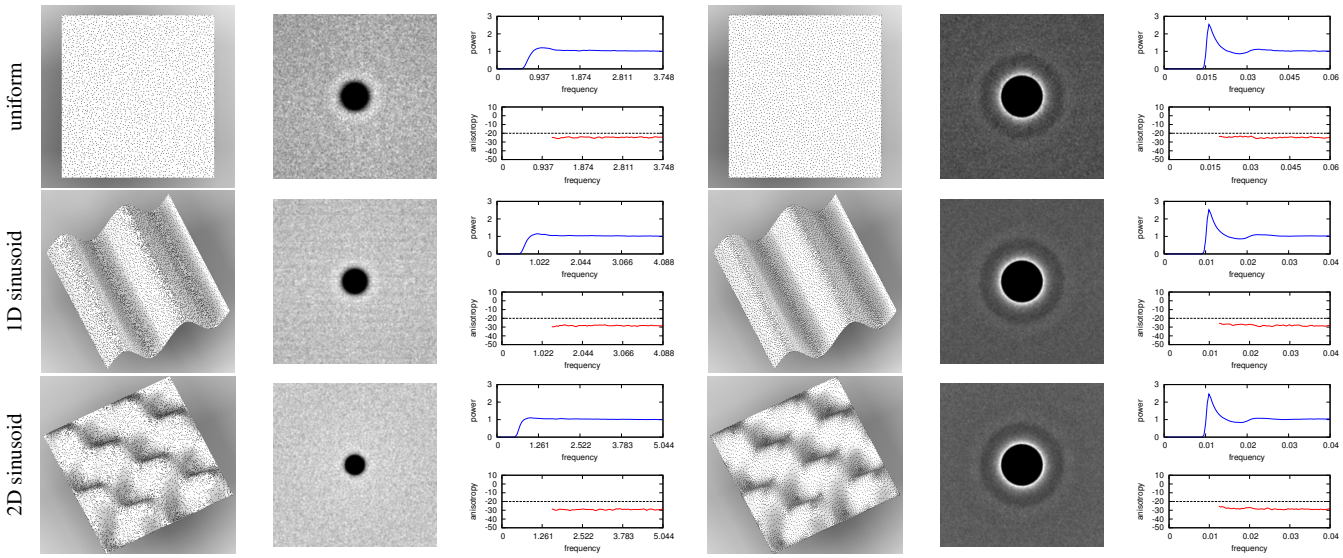


Figure 14: *Surface reconstruction of our results with varying sample rates. All results are produced by our bilateral dart throwing algorithm. As shown, our method can well preserve features under a variety of sampling rates.*

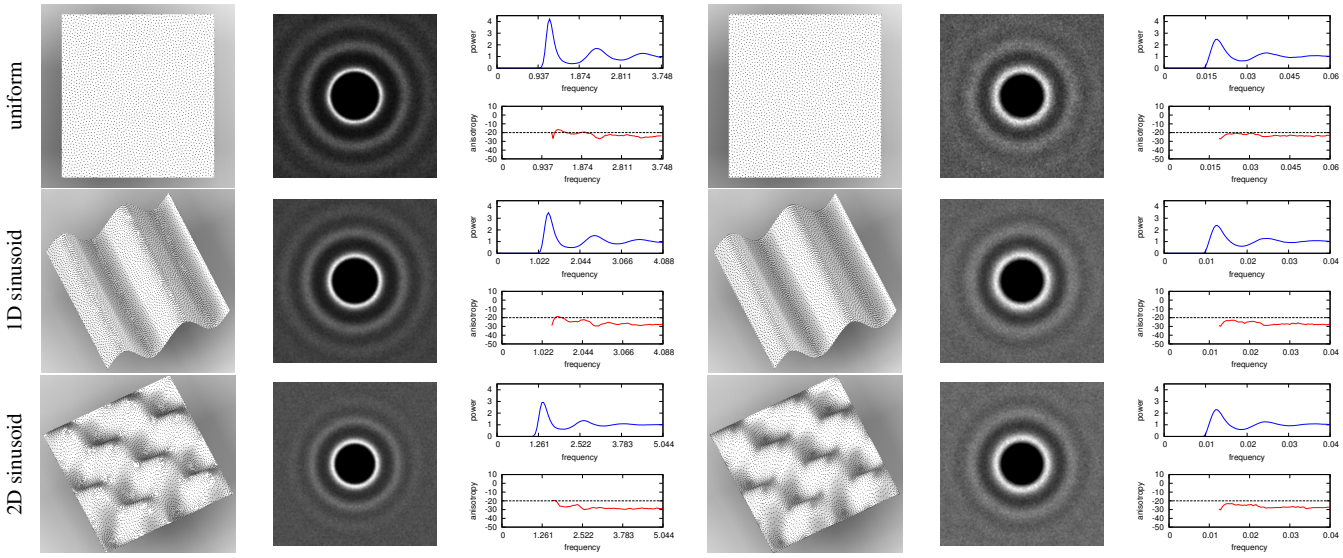
case	# samp	ρ				$H(10^{-5})$			
		d	s	r	g	d	s	r	g
uniform	2.89K	0.75	0.47	0.75	0.79	0	0	0	0
1d-sine	6.50K	0.75	0.44	0.75	0.76	13.26	17.06	12.83	16.60
2d-sine	6.50K	0.75	0.36	0.75	0.67	44.09	58.15	41.36	61.62

Table 2: *Statistics for for geometry sampling results in Figure 15 & 16. See Table 1 for more details.*



[Öztireli et al. 2010] subsampling dart throwing

Figure 15: *Blue noise properties of geometry sampling via dart throwing. Shown here are more sampling results of different domains from Figure 5, including uniform, 1D sinusoidal* $0.15\sin(2\pi fx)$ *with* $f = 2$*, 2D sinusoidal* $0.15\sin(2\pi fx)\sin(2\pi fy)$ *with* $f = 2$*. Detailed statistics for each case can be found in Table 2.*



[Öztireli et al. 2010] gradient ascent relaxation

Figure 16: *Blue noise properties of geometry sampling via relaxation/gradient-ascent following the results from Figure 15.*