Bilateral Blue Noise Sampling

Xiaoyin Ge* Li-Yi Wei† Yusu Wang* Huamin Wang*

*The Ohio State University †The University of Hong Kong

Abstract

Blue noise sampling is an important component in many graphics applications, but existing techniques consider mainly the spatial positions of samples, making them less effective when handling problems with range features, such as color and geometry. Inspired by bilateral filtering, we propose a bilateral blue noise sampling strategy. Our key idea is a new sample-distance measure that considers not only samples' spatial domain locations but also their underlying range features. This measure is easy to implement and compatible with most previous sampling techniques. Based on this, we present bilateral blue noise synthesis algorithms and we perform a quality analysis on their results. We demonstrate the effectiveness of our method in a number of applications, including geometry sampling, dynamic stippling, and nonlinear filtering.

Keywords: bilateral, blue noise, sampling, filtering, stippling, domain, range, geometry, image, video

1 Introduction

Sampling is a fundamental component for a variety of computational tasks. Given a fixed number of samples, the goal is to best represent a given sample domain and its corresponding feature range. While the notion of "best representation" is application dependent, it often consists of two main components: spatial domain blue noise and range feature preservation. Here, blue noise refers to sample distributions that are uniform and yet random, resulting in reduced noise and aliasing [Cook 1986; Ulichney 1987]. For example, in image stippling [Balzer et al. 2009; Li et al. 2010; Fattal 2011], the domain is a 2D pixel grid, the range is the pixel color, and the goal is to distribute the stipples as spatial blue noise and yet depicts the underlying image colors. As another example, in geometry sampling [Öztireli et al. 2010], the domain is a 3D point cloud, the range is the surface normal, and the goal is to sub-sample the points so that the output can reproduce smooth and sharp regions of the underlying surface. While different applications have different range features, it is difficult to preserve both domain blue noise and range feature, since they conflict with each other in many cases. Although researchers have carried extensive research on preserving either of them, no existing techniques can handle both well as far as we know. Without considering range features, the results will lose interesting details as the left column in Figure 1 shows.

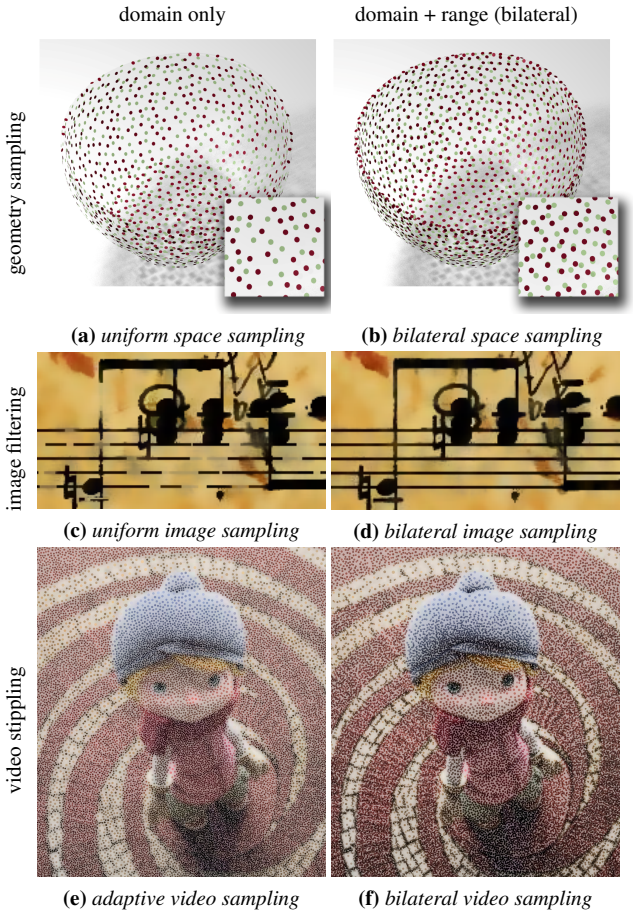


Figure 1: *Domain versus bilateral sampling. We propose a new sample-distance measure that considers both spatial domain blue noise and range feature, for uniform and non-uniform sampling problems. Our result demonstrates the effectiveness of this method in multiple cases, including uniformity of the sub-sampled point clouds, fidelity of the filtered image colors, or clarity of the stippled video contours. Sample point colors in (a) and (b) indicate whether the points are on front or back surfaces.*

We present bilateral blue-noise sampling, a general method whose sampling results preserve both domain blue noise and range features. Our basic idea is a sample-distance measure that incorporates both sample domain positions and range features. This idea is inspired by bilateral filtering [Tomasi and Manduchi 1998], a simple yet elegant technique that has been widely in graphics, vision,

and image processing applications (e.g., [Paris et al. 2009] and references within). Our idea shares similar simplicity and generality, and it can be readily combined with prior blue noise sample analysis and synthesis algorithms.

In this paper, we demonstrate scenarios that can benefit from our method, including geometry sampling, dynamic stippling, and nonlinear filtering. For the uniform *geometry* point-cloud sampling case, traditional blue noise that considers only Euclidean distances can easily mix up samples that are spatially close but with opposite normal directions, as Figure 1a shows. By considering both Euclidean distance and point normals, our method can produce desired uniform spatial sampling as shown in Figure 1b. For the bilateral *image* filtering case, placing filter taps via uniform spatial blue noise [Banterle et al. 2012] can increase filtering efficiency but miss image content potentially, such as broken lines on the music sheet in Figure 1c. Our method, by considering both spatial positions and image colors, can preserve such content while retaining benefits of spatial blue noise as in Figure 1d. For the adaptive *video* stippling case, our method preserves contours better than traditional dynamic blue noise [Vanderhaeghe et al. 2007], as shown in Figure 1f and 1e.

2 Previous Work

Bilateral filtering Bilateral filtering is a simple yet effective feature-preserving smoothing algorithm. It has been applied to a variety of image and mesh problems [Tomasi and Manduchi 1998; Jones et al. 2003; Paris et al. 2009; Huang et al. 2012]. Similar to bilateral filtering, our bilateral blue noise sampling method can preserve range features well, especially when compared with previous blue noise techniques.

Blue noise sampling Blue noise sampling generates random samples with uniform distribution. It is known for its uniformity in the spatial domain, low noise/aliasing in the spectrum domain, and its robustness when used in numerical applications [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; de Goes et al. 2012]. Unfortunately, many existing blue noise sampling techniques were developed without taking feature preservation into consideration. While some recent ones [Pang et al. 2008; Chang et al. 2009; Li et al. 2010; Kalantari and Sen 2011] incorporate features into their formulae, they are limited to specific problems, such as regular grids for halftoning [Pang et al. 2008; Chang et al. 2009], or thin image features through maximal point set and min-conflict metric [Kalantari and Sen 2011]. In some cases, it may become a tradeoff between blue noise properties and feature preservation, as Li and collaborators showed [Li and Mould 2010; Li and Mould 2011].

Importance sampling Even though non-uniform blue-noise allows samples to be distributed with density proportional to the importance of domain regions (see e.g. [Balzer et al. 2009; Li et al. 2010; Fattal 2011; de Goes et al. 2012]), 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. For two, if the importance changes dramatically as it often does 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 1 and 6 for comparisons. In addition, importance sampling cannot help uniform sampling that requires to consider both domain and range properties, since it is not intuitive to specify "importance" as shown in Figure 1. That being said, importance, and general non-uniform sampling, is orthogonal to our bilateral method and they can combined for better sampling results.

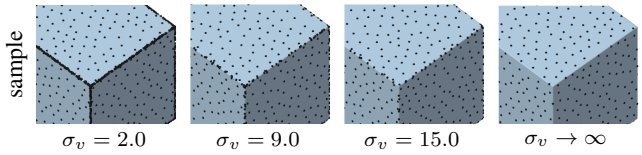


Figure 2: *The effect of σ_v. When $\sigma_p = 1$, a smaller σ_v can cause more samples to be placed near features, such as ridges and corners. Setting $\sigma_v = \infty$ reduces our method to traditional blue noise sampling. We use normal direction as feature $\mathbf{v}$ in this example.*

Feature-aware sampling Feature-aware sampling methods have been designed for various applications, such as stippling [Kim et al. 2008; Li and Mould 2011], half-toning [Li and Mould 2010], and meshing [Lévy and Liu 2010]. However, they usually do not preserve blue noise properties, which are desired by the corresponding applications. 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 of feature-preservation. Our method can better preserve both blue noise characteristics and surface features. Furthermore, most of these methods are designed for one particular application and they are not generic enough as ours.

3 Bilateral Sampling

Given a target domain, such as an image, video, or mesh, our goal is to generate a set of blue noise samples that are representative to the underlying range features. The blue noise property can be evaluated by a unified approach, such as power spectrum and spatial uniformity [Wei and Wang 2011]). How representative the samples are to the features is a more application-dependent problem and we will study it using different measures, including surface reconstruction for geometry sampling, visual perception for image filtering, or temporal coherence for dynamic stippling.

The basic idea behind bilateral blue noise sampling is to define a distance d between any two samples s and s':

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

, in which $\mathbf{p}$ is the sample's position vector, $\mathbf{v}$ is the sample's feature vector, and σ_p and σ_v are the weights. Intuitively, $\mathbf{p}$ and $\mathbf{v}$ correspond to the *domain* and *range* parts of bilateral filtering. d_p is the spatial sample distance in traditional blue noise sampling, either Euclidean (e.g. image sampling) or Riemannian (e.g. surface sampling). d_v 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. The coefficients σ_p and σ_v provide a balance between blue noise and feature preservation. When $\sigma_v = \infty$, our method is reduced to conventional blue noise sampling, and when σ_v gets smaller, our method puts more emphasis on features than blue noise, as Figure 2 shows. In our experiment, we set $\sigma_p = 1$ and $\sigma_v \in [7, 11]$.

Equation 1 will be used in both synthesis and analysis. For synthesis, we will generate random samples in a given domain Ω, such that the samples are uniformly distributed according to the distance measure in Equation 1. For analysis, we will evaluate Equation 1

using quality measures, including spatial measures (such as minimum spacing ρ [Lagae and Dutré 2008]) and spectrum measures (such as differential domain analysis [Wei and Wang 2011]).

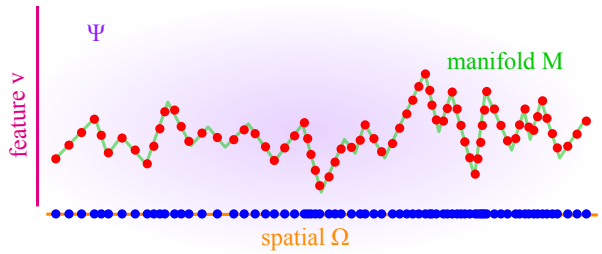


Figure 3: *Manifold interpretation. Given a spatial domain Ω (horizontal axis) with features $\mathbf{v}$ (vertical axis), if we perform a uniform sampling (•) in the corresponding manifold M embedded in a higher dimensional space Ψ, the corresponding samples (•) manifested in Ω can have non-uniform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

If we assume that $\mathbf{p}$ and $\mathbf{v}$ form a large vector in a higher-dimensional space Ψ, Equation 1 basically calculates the distance between two samples in Ψ. Bilateral sampling generates random samples over an embedded manifold M in Ψ, and the result is their projections to a lower-dimensional space, as Figure 3 shows. 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 gauging the distribution uniformity over M during analysis.

Approximation 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 $\mathbf{J}$, and use these as local linear approximations for conducting analysis or synthesis on Ω. We compute $\mathbf{J}$ 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.

For domain Ω , the corresponding Jacobian $\mathbf{J}$ can be derived as follows. For notational simplicity, let

$$\hat{\mathbf{p}} = \frac{\mathbf{p}}{\sigma_p}, \ \hat{\mathbf{v}} = \frac{\mathbf{v}}{\sigma_v} \tag{2}$$

. We have

$$\mathbf{J}(\varphi^{-1}) = \begin{pmatrix} \mathbf{J}_{\mathbf{p}}(\hat{\mathbf{p}}) \\ \mathbf{J}_{\mathbf{p}}(\hat{\mathbf{v}}) \end{pmatrix} \tag{3}$$

, where $\mathbf{J}_{\mathbf{x}}(\mathbf{y})$ indicates the jacobian of $\mathbf{y}$ relative to $\mathbf{x}$. Note that $\mathbf{J}_{\mathbf{p}}(\hat{\mathbf{p}}) = \frac{1}{\sigma_p} I$ (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 applied to prior sampling methods by simply replacing the traditional position-only distance with our bilateral distance as described in Section 3. Below, we provide concrete examples for two popular algorithms, dart throwing and relaxation.

4.1 Dart Throwing

Dart throwing [Dippé and Wold 1985; Cook 1986] produces individual samples 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 bilateral distance d from Equation 1 in lieu of the spatial-only distance in traditional dart throwing. By using different $r(s, s')$ representations, our method can be orthogonally applied for various sampling scenarios, including uniform (r is a constant), isotropic ($r(s, s')$ depends on the distance between s and s' but not their relative direction), 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:

$$\mathbf{E}(S, \mathcal{V}) = \sum_i \int_{s' \in V_i} d^2(s', s_i) ds' \tag{4}$$

, 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:

$$s(s') = \arg\min_{s \in S} d^2(s', s) \tag{5}$$

. 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:

$$\int_{s' \in V_i} d^2(s', s_i) ds' \tag{6}$$

. This can be achieved over the embedded manifold M (see our manifold interpretation in Section 3 and [Du et al. 2002]), or in the original domain Ω via the Jacobian approximation [Li et al. 2010] as follows:

$$m_i = \left(\int_{V_i} \mathbf{J}^{\mathrm{T}} \mathbf{J}(s') ds' \right)^{-1} \int_{V_i} \mathbf{J}^{\mathrm{T}} \mathbf{J}(s') s' ds' \tag{7}$$

, where $\mathbf{J}$ is the Jacobian computed via Equation 3.

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.

Our method can also be directly extended for non-uniform relaxation [Li et al. 2010] and various forms of capacity constraints [Balzer et al. 2009; Chen et al. 2012; de Goes et al. 2012]. Details can be found in the supplementary materials.

5 Analysis

Sample distributions can be analyzed 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 over M 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, and perform spatial and spectrum analysis there. For example, we can use the following formula for the differential domain analysis in [Wei and Wang 2011]

$$\chi(\mathbf{d}) = \varphi^{-1}(s) - \varphi^{-1}(s') \tag{8}$$

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

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 Ψ), 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 Ψ for sampling 3D surfaces). For such cases, we resort to approximations described below.

Approximation in Ω 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 3 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 $\mathbf{J}^{\mathrm{T}}\mathbf{J}$, not $\mathbf{J}$ itself, all we need is to derive a square $\mathbf{J}'$ so that

$$\mathbf{J}^{\mathrm{T}}\mathbf{J} = \mathbf{J}'^{\mathrm{T}}\mathbf{J}' \tag{9}$$

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

$$\mathbf{J}^{\mathrm{T}}\mathbf{J} = \mathbf{V}^{\mathrm{T}}\mathbf{D}\mathbf{V} \tag{10}$$

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

$$\mathbf{J}' = \mathbf{V}^{\mathrm{T}}\sqrt{\mathbf{D}}\mathbf{V} \tag{11}$$

. Following Equation 15 of [Wei and Wang 2011] for anisotropic sampling, we have $\mathbf{d} = s - s'$, and

$$\chi(\mathbf{d}) = \frac{1}{E(\lambda)}\left(\frac{\mathbf{J}'^{-1}(s) + \mathbf{J}'^{-1}(s')}{2}\right)^{-1}(s - s')^{\mathrm{T}} \tag{12}$$

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

6 Results

In this section, we present our results in three applications: geometry sampling, dynamic stippling, and nonlinear filtering.

6.1 Geometry Sampling

Geometry sampling is important for graphics and simulation, and can benefit from sample distributions that preserve both features and blue noise properties. In particular, allocating samples at features such as tips and creases are critical in preserving 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 geometry information (e.g. surface normal $\mathbf{n}$) as the feature $\mathbf{v}$ in Equation 1.

Here, we use point cloud down-sampling as a specific example [Öztireli et al. 2010]. Similar to [Öztireli et al. 2010], we assume the input consists of a point cloud with per vertex position and normal direction (which can be estimated through various methods as discussed in [Öztireli et al. 2010; Huang et al. 2012]). Our method can use geometry measures other than surface normal $\mathbf{n}$ (which tends to be noisy) as the feature $\mathbf{v}$, but we will focus on $\mathbf{n}$ for easy presentation and comparison with [Öztireli et al. 2010]. Because the underlying surface of the input geometry is unknown, it is impossible to use accurate geodesic distances among samples. Thus, Euclidean distance is commonly used as an approximation, which is usually adequate with sufficiently dense input samples.

Traditional blue noise sampling, by considering only sample positions, might incorrectly handle samples that have close spatial proximity but differ sufficiently in normal directions, such as samples lying on the opposite sides of a thin feature. The spectrum method in [Öztireli et al. 2010] considers both position and normal and thus provides better results, but our method is much simpler and produces even better results. We illustrate these in Figure 4 & 5, comparing the two main sampling methods for dart throwing and relaxation, which correspond to the sub-sampling and gradient-ascent methods in [Öztireli et al. 2010], respectively. As shown, bilateral dart-throwing/relaxation can produce more effective distributions (more uniform and better feature preservation) than sub-sampling/gradient-ascent, resulting in better surface reconstruction quality. Notice the thicker ring on the top of the bow lip and the hole on the apex of the moose horn.

6.2 Dynamic Stippling

Bilateral blue noise sampling can be applied to a variety of dynamic effects, 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.

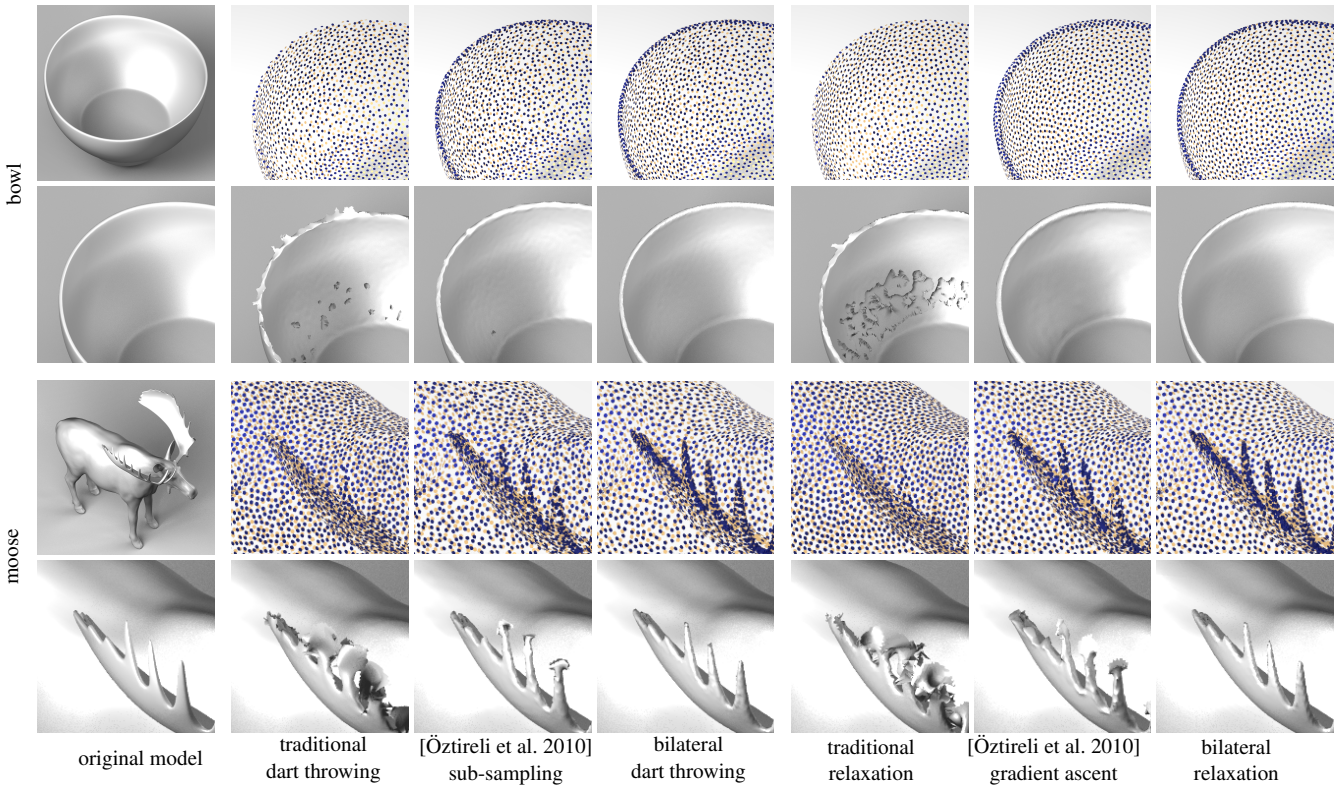


Figure 4: *Point cloud resampling results. Here, we compare traditional dart-throwing/relaxation, [Öztireli et al. 2010] sub-sampling/gradient-ascent, and our bilateral dart-throwing/relaxation. (Each relaxation/gradient-ascent result is produced from the corresponding dart-throwing/sub-sampling result.) Within each bowl/moose case are the point clouds (top row) and reconstructed surfaces (bottom row), except for the first column which shows the original models in full (top) or in the demonstrated subset (bottom). As shown, bilateral sampling methods achieve a more uniform sample distribution and hence more accurate surface reconstruction (using RIMLS in [Öztireli et al. 2009]). For the bowl/moose case, the input point cloud has $\sim 400K$/$800K$ samples and each output result has $\sim 6K$/$20K$ samples.*

2007], is to maintain the balance between 2D spatial blue noise, 1D temporal coherence, and 2D (video) or 3D (object) motion depiction. This is a challenging problem due to multiple spatial and temporal constraints. Here, we describe two particular applications: cross-dimensional 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 bilateral 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/adding samples from/to crowded/sparse regions. (Please refer to the accompany video of Figure 6 & 7 for animation effects.)

Feature $\mathbf{v}$ For video stylization, we simply use colors $\mathbf{c}$ as $\mathbf{v}$. For cross 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 $\mathbf{n}(s')$:

$$s' = raycast(s),\ \mathbf{v}(s) = \big(\alpha \mathbf{c}(s), \mathbf{n}(s')\big) \tag{13}$$

, 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 geometry features better than considering only 3D geometry or 2D image features. Some examples include interior and exterior silhouettes (e.g. the genus 3 model in Figure 7) 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 7).

Video stylization Figure 6 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. Here, the motions are computed through video optical flow, and the feature $\mathbf{v}$ is the underlying RGB video pixel color $\mathbf{c}$. As shown in Figure 6, our method preserves features better than non-bilateral blue noise [Vanderhaeghe et al. 2007] while maintains its other advantages including motion depiction, temporal coherence, and screen space blue noise.

Cross-dimensional sampling Figure 7 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 bilateral cross sampling method against the original cross dimensional (non-bilateral) blue noise sampling in [Vanderhaeghe et al. 2007]. As shown, our method provides better quality in describing features in both image and object spaces. For tone reproduction, we set the local adaptive-sampling radius $r(s)$ according to the shading value.

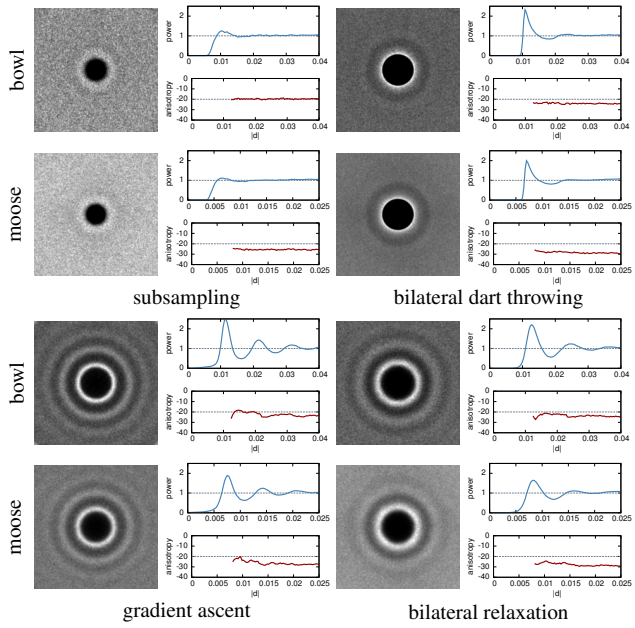


Figure 5: *Differential domain analysis for surface sampling. Here we show the DDA [Wei and Wang 2011] spectrum, radial mean, and radial anisotropy for the point sets in Figure 4. Compared to the methods in [Öztireli et al. 2010] (left), our methods (right) have better blue noise properties, such as more uniform distributions in dart throwing versus subsampling (upper group) and the lower amount of anisotropy in general.*

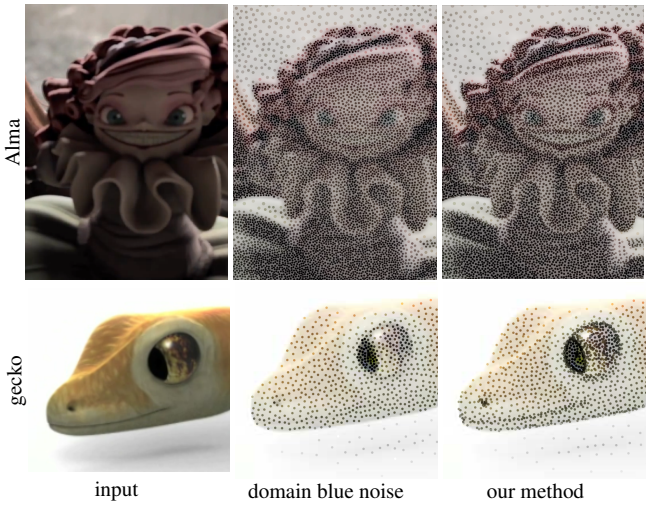
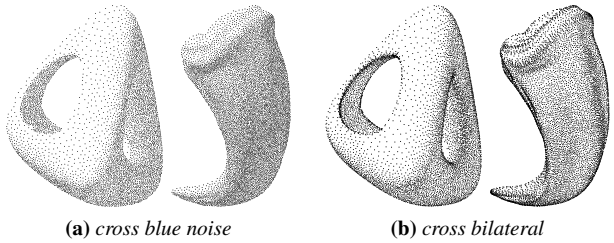


Figure 6: *Spatial-temporal sampling for video stylization. Notice the better feature preservation of our method, such as the gecko mouth and eye, and the doll face and clothing. Each frame of gecko/Alma contains ∼ 2200/19000 samples.*

6.3 Nonlinear Filtering

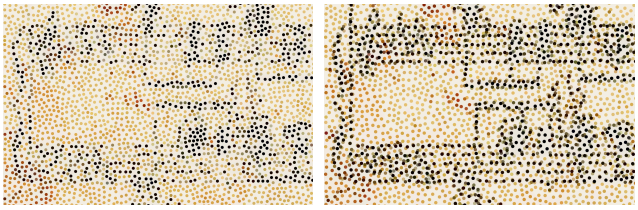
Nonlinear filtering, such as bilateral and medial filtering, has a variety of important applications. However, it tends to be relatively slow compared to linear filtering. Various acceleration methods have been proposed (see e.g. [Weiss 2006; Chen et al. 2007; Adams et al. 2009; Gastal and Oliveira 2012]).

Among these accelerations, sub-sampling has shown to be a viable approach for filters that can function well with a sub-set of taps, such as bilateral filtering [Banterle et al. 2012]. Banterle et al. [2012] further demonstrated that, among various sub-sampling schemes, blue noise offers unique advantages such as reduced noise and absence of aliasing. However, their method uses a global constant sampling rate for each image. This content-oblivious approach might not adapt well to the underlying image content, where complex areas may require more samples whereas simple areas may suffice with less. This naturally leads to a content-aware subsampling method, whose sampling rate varies according to the underlying image region complexity.



(a) *cross blue noise* **(b)** *cross bilateral*

Figure 7: *Cross-dimensional dynamic sampling for 2D stylization of 3D objects. Here we compare our cross dimensional bilateral blue noise sampling to the non-bilateral method in [Vanderhaeghe et al. 2007]. As shown, our method preserves features better in both object and image spaces (such as object silhouettes, geometry ridges, and shading variations), while retains the benefits of the original method, such as temporal coherence and screen-space blue noise. Each case of genus3/claw contains ∼ 4500/8000 visible samples. The input models are shown on the left.*

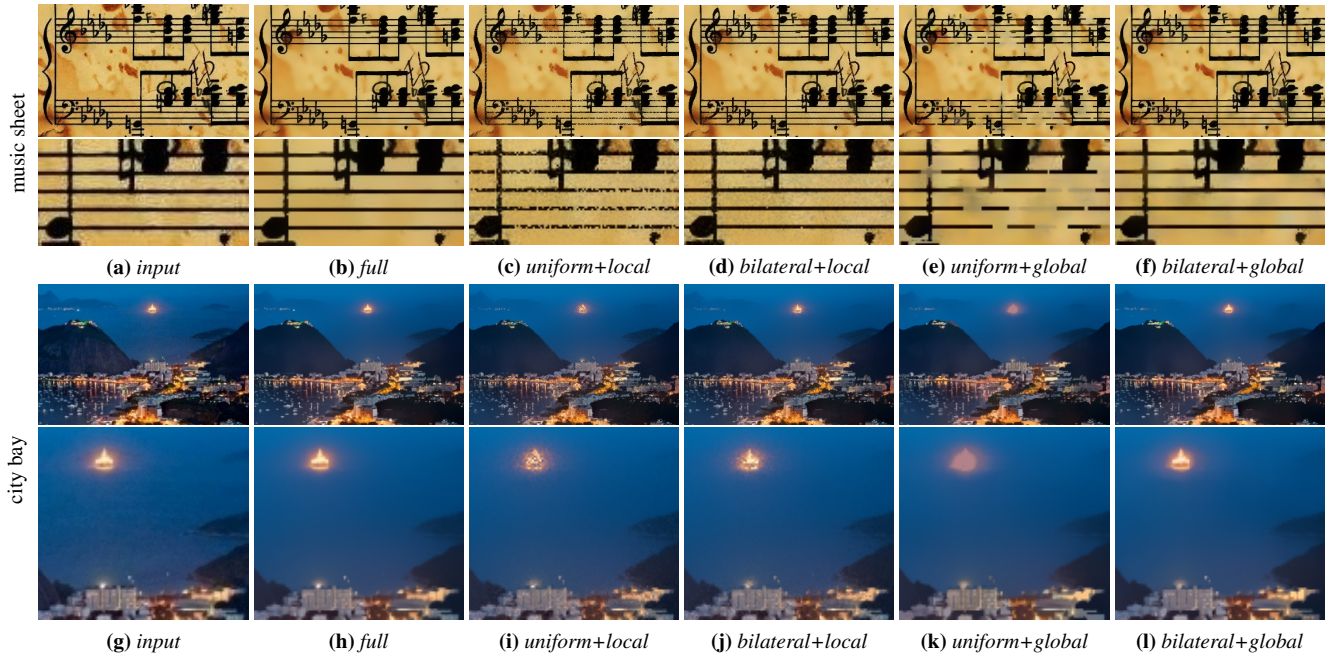


(a) *uniform + global* **(b)** *bilateral + global*

Figure 8: *Filter sample distributions around the upper-left corner of the music sheet in Figure 9. Notice the better preservation of features for the bilateral method. Here, all samples lie on a discrete grid and are colored through the underlying pixels.*

Our approach offers a potential content-aware sub-sampling method for nonlinear filtering. In a nutshell, we perform bilateral blue noise sampling based on the underlying image content, by treating image color $\mathbf{c}$ as the feature $\mathbf{v}$ in our Equation 1, and use these samples for filtering. We propose two flavors of our method to meet different speed/memory considerations: *local*, in which sample pattern within a filter kernel is generated independent for each output pixel (as originally proposed by [Banterle et al. 2012]); and *global*, in which sample pattern of the entire output image is produced as a pre-process (Figure 8), from which samples falling within individual output kernels are used during filtering.

In the following, we compare uniform/bilateral $\times$ local/global sampling methods for bilateral filtering via quality (Figure 9 & Figure 10) and performance (Table 1). Based on these, we have found that the bilateral + global method offers the best tradeoff.



(a) *input* **(b)** *full* **(c)** *uniform+local* **(d)** *bilateral+local* **(e)** *uniform+global* **(f)** *bilateral+global*

(g) *input* **(h)** *full* **(i)** *uniform+local* **(j)** *bilateral+local* **(k)** *uniform+global* **(l)** *bilateral+global*

Figure 9: *Subsampling-accelerated bilateral filtering. Here we compare uniform/bilateral × local/global subsampling schemes with full bilateral filtering. With each case are the full image (top) and the zoom-in (bottom). Notice the better quality of bilateral sampling for both the local and global settings, i.e. less noise for the former and less bias (in the form of large missing chunks) in the latter. In the music-sheet/city-bay case, the sub-sampling results are generated with kernel size K=15/30 pixels and average samples-per-kernel $N_s = 0.5K$.*

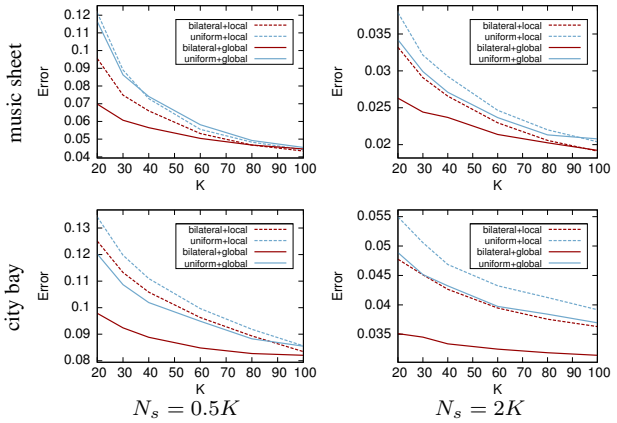


Figure 10: *Accuracy evaluation of different sub-sampling methods versus different kernel sizes K. Shown here are the RMSE relative to the ground truth full bilateral filtering of uniform/bilateral × local/global , respectively, over the music sheet/city bay input in Figure 9. The left/right columns are computed with different samples-per-kernel N_s=$0.5K$/$2K$. As shown, under identical sample counts, the bilateral method tends to have a lower error rate than the uniform method, whereas the global method tends to have a lower error rate than the local method.*

Quality Bilateral sampling tends to produce better quality than uniform sampling, especially under sparse sample distributions. This is illustrated in Figure 9 for perceptual image quality, and Figure 10 for numerical error measurements under a variety of images and parameter settings. As a rule of thumb, bilateral sampling can achieve similar quality to uniform sampling with fewer samples, usually $75/60\%$ for local/global sampling.

Global sampling tends to be less noisy than local sampling, both visually and numerically. This is because in a global method, two adjacent output pixels can have overlaps in their filter tap/sample sets, providing extra coherence than the case where the filter sets are produced independently. We have observed that such sample set coherence may cause bias in a very sparse sampling setting (e.g. large missing chunks in Figure 9e), but in most cases, it outperforms a local sampling method without such coherence.

Performance Table 1 shows the timing information of various methods under equal quality settings, with major steps separated for clarity, such as sampling, filtering, and Voronoi cell area computation which is needed for unbiased filtering under bilateral sampling. Compared to uniform sampling, bilateral sampling tends to take more time in sampling and less time in filtering. Global sampling tends to be faster than local sampling due to amortized sampling workload among output pixels. Note that global-bilateral sampling is faster than uniform sampling (both local and global), even assuming zero sampling time for the latter (e.g. using pre-computed tiles as in [Banterle et al. 2012]). So far, we did all of our measurements on a single CPU (Intel Core i7 machine with 4GB memory). Due to the parallel nature of blue noise sampling [Wei 2008] and filtering, we believe the entire process can be further accelerated for multi-/many-core CPUs/GPUs.

Parameters For the kernel size K and per-kernel sample count N_s, we recommand to use $\frac{K}{N_s} = 0.5 \sim 2.0$ in general by considering both quality and speed. For fair comparison between local and global methods, we use following rule to compute the total sample count for the global method: $M_s = \frac{|\Omega|}{K^2} \cdot N_s$, where $|\Omega|$ is the domain size and K the filter kernel size.

We set $\frac{\sigma_v}{\sigma_p}$ differently for local and global methods. For the local-bilateral method, our experiments indicate that $\frac{\sigma_v}{\sigma_p} = 2.0 \cdot |\Omega|^{-\frac{1}{2}} \sim$

$8.0 \cdot |\Omega|^{-\frac{1}{2}}$ will give optimal outcomes in both visual and accuracy evaluation, across different kernel sizes K and different sample counts N_s. For the global bilateral method, our experiments suggest to use $\frac{\sigma_v}{\sigma_p} = 3.0 \cdot |\Omega|^{-\frac{1}{2}} \sim 10.0 \cdot |\Omega|^{-\frac{1}{2}}$. Among this range, we further suggest a lower $\frac{\sigma_v}{\sigma_p}$ for a smaller kernel size (e.g. $K < 40$) and a relatively higher $\frac{\sigma_v}{\sigma_p}$ for a larger kernel size.

kernel size $K = 10/20/40$ pixels			
uniform (local* and global†)			
	$N_s = 1K$	$N_s = 2K$	$N_s = 3K$
sampling*	0.83/*1.52*/*2.26*	1.15/*2.56*/*4.93*	1.74/*2.96*/*7.89*
sampling†	0.94/*0.30*/*0.07*	2.00/*0.67*/*0.26*	4.37/*1.19*/*0.41*
filtering	3.96/*8.33*/*16.45*	8.37/*16.00*/*33.20*	12.18/*25.26*/*49.54*
total*	4.79/*9.85*/*18.71*	9.52/*18.56* /*38.13*	13.93/*28.23*/*57.43*
total†	4.90/*8.63*/*16.52*	10.37/*16.67* /*33.46*	16.55/*26.45*/*49.95*
local bilateral			
	$N_s = 0.75K$	$N_s = 1.5K$	$N_s = 2.25K$
sampling	4.59/*8.59* /*14.11*	8.82/*19.49*/*36.94*	13.56/*34.68*/*61.99*
Voronoi	5.19/*11.55*/*23.86*	11.10/*23.00*/*46.91*	15.22/*34.24* /*71.70*
filtering	3.11/*6.48*/*12.78*	6.22/*12.41*/*24.00*	9.59/*18.37*/*36.43*
total	12.90/*26.62*/*50.74*	26.14/*54.90*/*107.8*	38.37/*87.30*/*170.1*
global bilateral			
	$N_s = 0.6K$	$N_s = 1.2K$	$N_s = 1.8K$
sampling	0.30 /*0.15*/*0.04*	0.93/*0.33*/*0.11*	1.74 /*0.59*/*0.19*
Voronoi	0.22/*0.15*/*0.04*	0.44/*0.26*/*0.10*	0.74/*0.37*/*0.18*
filtering	2.74/*5.30*/*9.48*	5.26/*9.85*/*19.34*	7.74/*14.96*/*28.53*
total	3.26/*5.60*/*9.56*	6.63/*10.44*/*19.55*	10.23/*15.93*/*28.90*

Table 1: *Timing information for sub-sampling accelerated bilateral filtering. Here we show the timing information of various methods, with kernel size K = 10, 20, and 40 pixels, and various sample counts under equal-quality settings (75/60% of samples-per-kernel for bilateral local/global sampling relative to uniform sampling as described in the main text). The measurement units is 10^{-3}sec/Kpixel. For uniform sampling, the local and global methods differ only in sampling time and share very similar filtering time.*

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], which shares distance measure and approximations for some of the synthesis and analysis parts. A potential future work is to examine alternative analysis methods that are standalone and independent of any sample synthesis algorithms. In our formulation we have been using either Euclidean or Riemannian 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]. Further refinements in acceleration methods will help the performance of bilateral sampling.

We believe for geometry sampling, in addition to surface normal as the feature, our method can also be directly extended for other geometric measures, which may offer more robustness. For non-linear filtering, in addition to bilateral filtering, our method could be well applied to other methods such as median filtering [Weiss 2006].

Finally, we would like to extend our idea to other sampling patterns [Zhou et al. 2012; Öztireli and Gross 2012] and applications such as rendering [Spencer and Jones 2009] and fluids [Ando et al. 2012; Schechter and Bridson 2012].

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

ADAMS, A., GELFAND, N., DOLSON, J., AND LEVOY, M. 2009. Gaussian kd-trees for fast high-dimensional filtering. In *SIGGRAPH '09*, 21:1–12.

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

ANDO, R., THUREY, N., AND TSURUNO, R. 2012. Preserving fluid sheets with adaptively sampled anisotropic particles. *TVCG 99*.

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

BANTERLE, F., CORSINI, M., CIGNONI, P., AND SCOPIGNO, R. 2012. A low-memory, straightforward and fast bilateral filter through subsampling in spatial domain. *Computer Graphics Forum 31*, 1, 19–32.

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.

DE GOES, F., BREEDEN, K., OSTROMOUKHOV, V., AND DESBRUN, M. 2012. Blue noise through optimal transport. In *SIGGRAPH Asia '12*, 171:1–171:11.

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 multidimensional poisson-disk sampling. *ACM Trans. Graph. 29*, 1, 1–19.

GASTAL, E. S. L., AND OLIVEIRA, M. M. 2012. Adaptive manifolds for real-time high-dimensional filtering. *ACM Trans. Graph. 31*, 4 (July), 33:1–33:13.

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

HUANG, H., WU, S., GONG, M., COHEN-OR, D., ASCHER, U., AND ZHANG, H. 2012. Edge-aware point set resampling. *TOG*.

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., AND GROSS, M. 2012. Analysis and synthesis of point distributions based on pair correlation. In *SIGGRAPH Asia '12*, 170:1–170:10.

ÖZTIRELI, C., GUENNEBAUD, G., AND GROSS, M. 2009. Feature preserving point set surfaces based on non-linear kernel regression. In *Proceedings of Eurographics 2009*, 493–501.

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.

SCHECHTER, H., AND BRIDSON, R. 2012. Ghost sph for animating water. *ACM Trans. Graph. 31*, 4, 61:1–61:8.

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.

SPENCER, B., AND JONES, M. W. 2009. Into the blue: Better caustics through photon relaxation. *Comput. Graph. Forum 28*, 2, 319–328.

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.

ULICHNEY, R. 1987. *Digital halftoning*. MIT Press, Cambridge, MA.

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.

WEISS, B. 2006. Fast median and bilateral filtering. In *SIGGRAPH '06*, 519–526.

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.

ZHOU, Y., HUANG, H., WEI, L.-Y., AND WANG, R. 2012. Point sampling with general noise spectrum. In *SIGGRAPH '12*.

Supplementary Materials

A Method

Here we provide further details about our method.

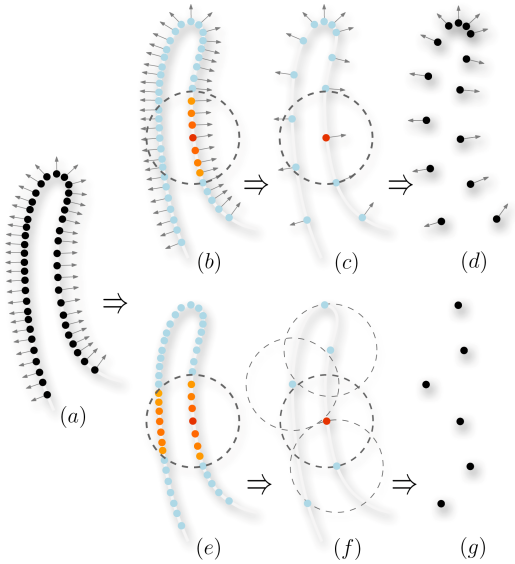


Figure 11: *Proximal surface illustration. We show the difference between bilateral blue noise sampling (b)(c)(d) and the traditional blue noise sampling (e)(f)(g) in dart throwing. (a) is the input point cloud, where thick light gray curve indicates the hidden surface; dashed circle in (b),(c),(e) and (f) indicates the Poisson disk of the red center sample. Orange dots in (b) and (e) are samples inside the Poisson disk of the red center sample. In traditional blue noise, all of them will be removed, causing an empty area in the left hidden surface. Bilateral blue noise sampling avoids this issue by considering also features such as surface normal.*

A.1 Relaxation

Non-uniformity The descriptions in Section 4.2 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 $\dot{\mathbf{J}}$ (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 $\mathbf{J}$ in Equation 3) information on top of our bilaterally derived d and $\mathbf{J}$ 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 $\mathbf{J}(s)$ from Equation 3 (for the centroid step) by $\frac{1}{r(s)}$. Similarly, for anisotropic sampling, we multiply both $\mathbf{d}(s, s')$ and $\mathbf{J}(s)$ by $\dot{\mathbf{J}}(s)$.

Capacity constraint We can also incorporate capacity constraints CCVT [Balzer et al. 2009] into our method above to avoid the potential semi-regular distributions commonly seen in Lloyd relaxation. CCVT modifies the Voronoi step of classical Lloyd relaxation, by ensuring that the number of points affiliated with each site remains unchanged throughout the iterative process. Since CCVT is orthogonal to our bilateral d, they can be naturally combined.

A.2 Jacobian

Here we show that the approximation in Equation 3 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:

$$\delta\mathbf{p}(s) = \mathbf{p}(s') - \mathbf{p}(s) \tag{14}$$

$$\delta\mathbf{v}(s) = \mathbf{v}(s') - \mathbf{v}(s) \tag{15}$$

. First, following Equation 1, we have:

$$\delta\mathbf{d}(s) = \left(\frac{\delta\mathbf{p}(s)}{\sigma_p} \; \frac{\delta\mathbf{v}(s)}{\sigma_v}\right) \tag{16}$$

. Then, following Equation 3, we have

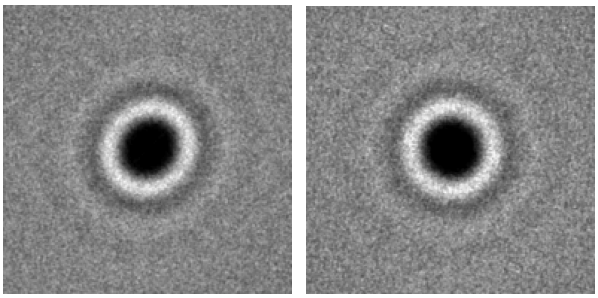
$$\mathbf{J}\delta\mathbf{p}(s) = \left(\frac{\delta\mathbf{p}(s)}{\sigma_p} \; \frac{\delta\mathbf{v}(s)}{\sigma_v}\right) \tag{17}$$

. Note that Equation 16 and Equation 17 produce the same results under the assumption that $\delta\mathbf{p}$ is small, thus:

$$\delta\mathbf{d} = \mathbf{J}\delta\mathbf{p} \tag{18}$$

.

A.3 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.4 Relationship to [Li et al. 2010]

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 3. A similar Jacobian approximation is also adopted in [Li et al. 2010].

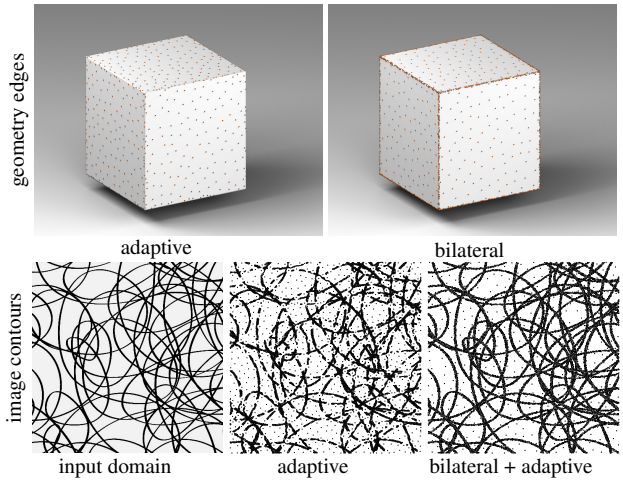


Figure 13: *Comparisons between adaptive and bilateral sampling. Shown here are sampling results for geometry and images. Results within each row are produced with similar number of samples (∼2800/17900 for geometry/image). Notice that our method preserves features better (e.g. geometry edges, image contours).*

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 $\mathbf{J}^{\mathrm{T}}\mathbf{J}$ 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. We use 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.

A.5 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 2). 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).

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.

(a) *full bilateral* **(b)** *uniform local* **(c)** *bilateral local*

Figure 14: *Subsampling bilateral filtering, local bilateral vs. local uniform. We compare our local bilateral blue noise sampling method with local uniform blue noise sampling method [Banterle et al. 2012] in a sparse sampling case, $K = 10$ pixel, $N_s = 0.3K$. The bilateral-local method (c) can preserve features better with less noise and blurs than the uniform-local method (b).*

B 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 resolutions 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]

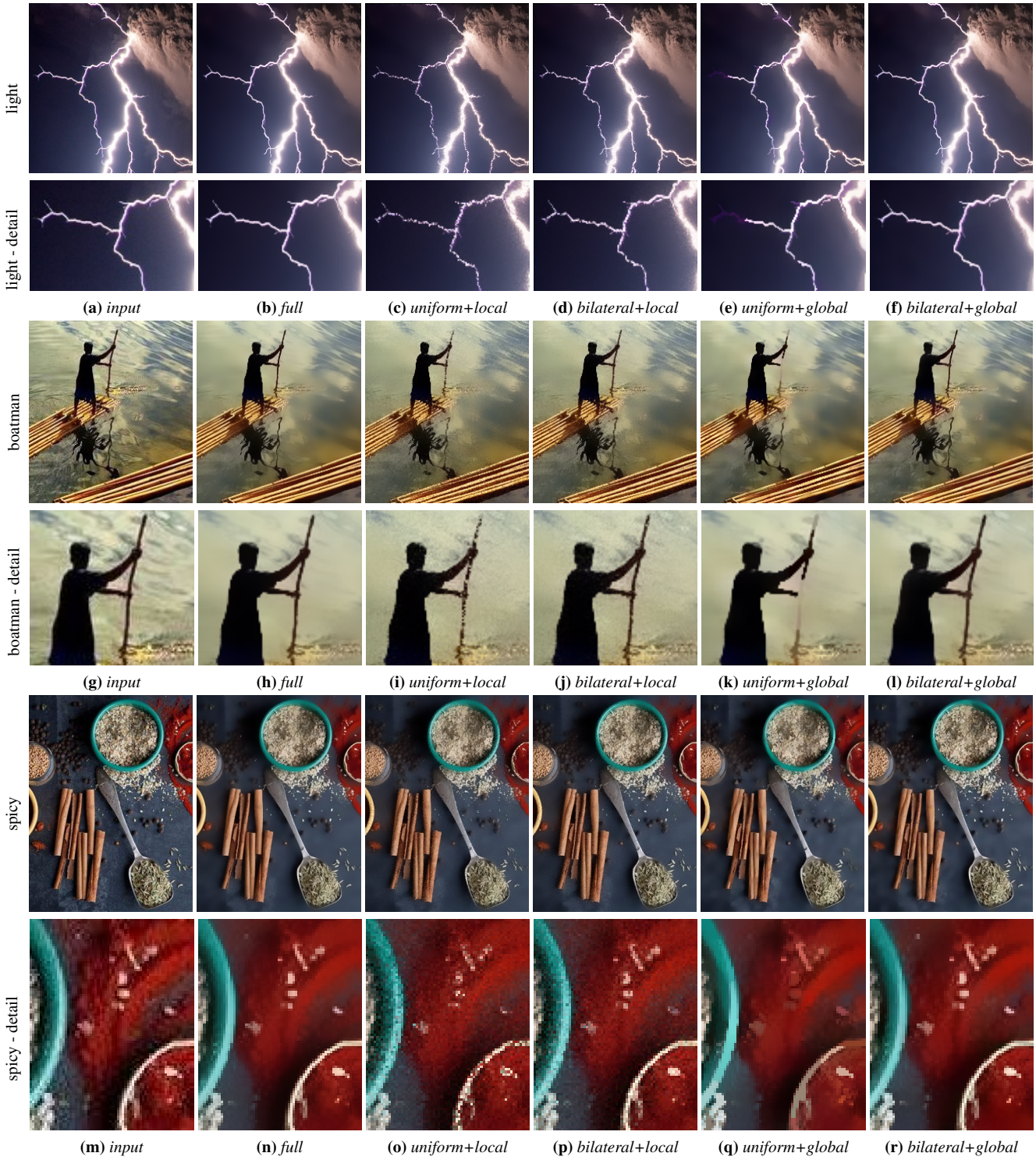


Figure 15: *Subsampling-accelerated bilateral filtering. Here we compare uniform/bilateral × local/global subsampling schemes with full bilateral filtering. Notice the better quality of bilateral filtering for both the local and global settings, i.e. less noise for the former and less bias (in the form of large missing chunks) in the latter. In the light/boatman/spicy, the sub-sampling results are generated with kernel size K=20/20/20 pixels and average samples-per-kernel $N_s = 0.5K$.*

by simply plugging the distance measure in Equation 1. As shown in Figure 1 and Figure 19, 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

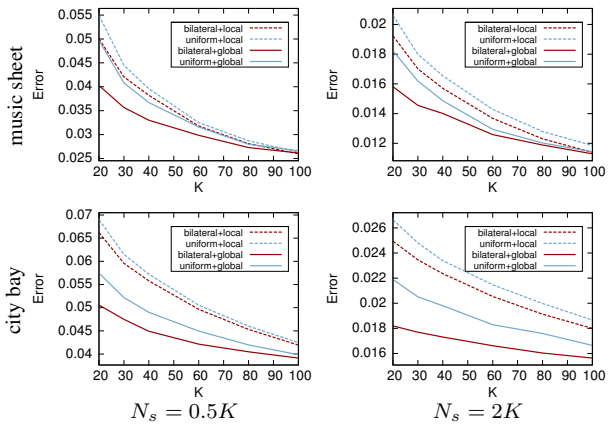
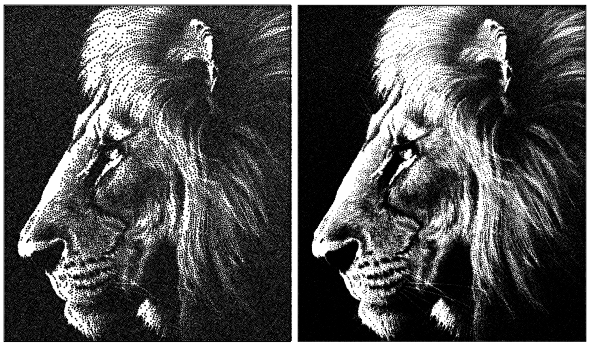


Figure 16: *Accuracy evaluation of different sub-sampling methods versus different kernel sizes K. Shown here are the mean error relative to the ground truth full bilateral filtering of the local-uniform [Banterle et al. 2012], local-bilateral, global-uniform, and global-bilateral blue noise sub-sampling methods, respectively, over the music sheet/city bay input in Figure 9. The left/right columns are computed with different samples-per-kernel N_s=$0.5K$/$2K$. As shown, under identical sample counts, the bilateral method tends to have a lower error rate than the uniform method, whereas the global method tends to have a lower error rate than the local method.*

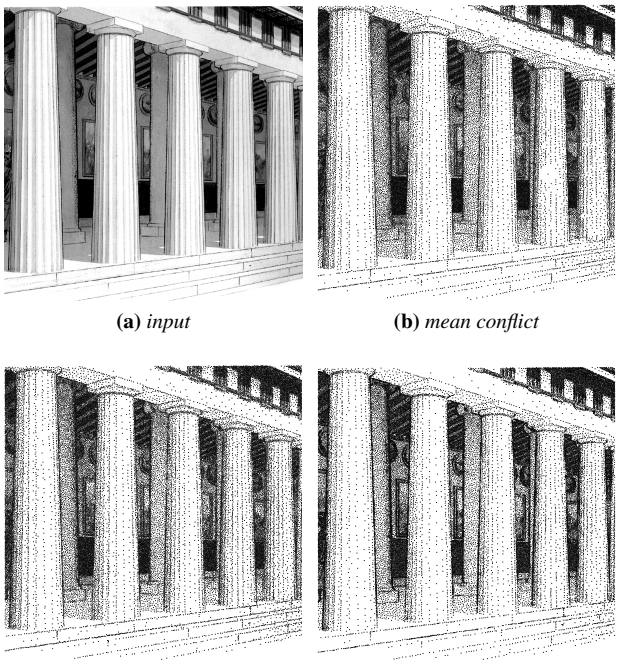
with other aspects of blue noise sampling, such as using intensity as the importance field to define local $r(.)$ fields.

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.



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

Figure 17: *Stippling with variable dot sizes. Notice the better structure and tone reproduction of (b) versus (a). The source image is shown on the left.*



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

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

Figure 18: *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).*

B.1 Metric Comparison

Figure 18 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 1) that our method is better than traditional adaptive sampling for simultaneous blue noise and feature preservations.

C Surface Sampling

Our method introduced in Section 6.1 can also be applied for direct surface sampling. Here we demonstrate sampling results over a variety of models with different geometric and topological properties, and analyzed both their blue noise and feature preservation properties. For the former 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 25, Figure 26, Table 2 and Figure 27, 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.

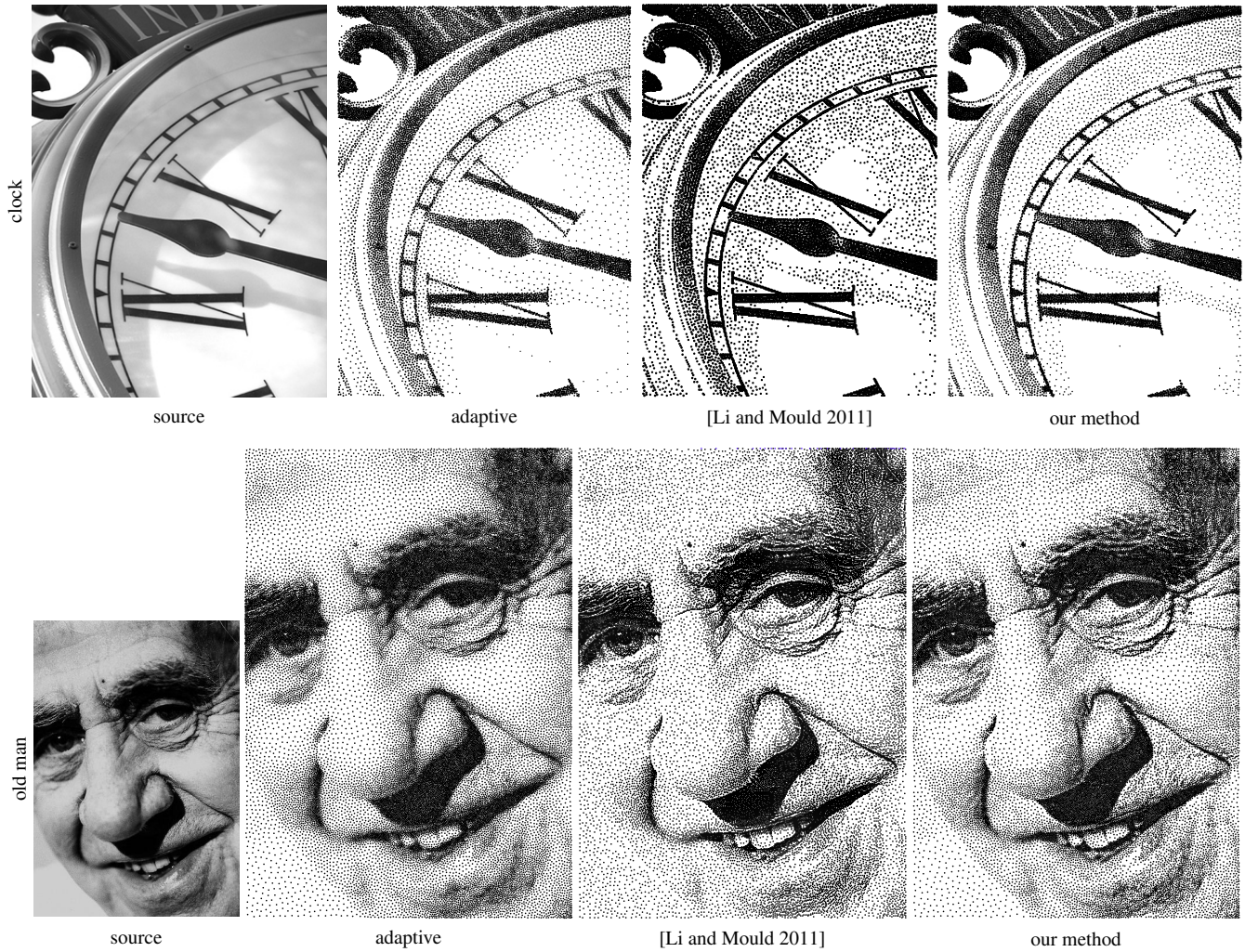


Figure 19: *Image stippling results. We compare our method against traditional adaptive importance blue noise sampling and [Li and Mould 2011], a state of art stippling method that considers structure, blue noise, and tone reproduction. As shown, our method outperforms adaptive importance sampling in feature preservation, and [Li and Mould 2011] in tones and structures (e.g. better boundary between the chin and background in the lower left corner of old man face) and blue noise properties (e.g. more uniform sample distributions).*

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 2: *Statistics for for geometry sampling results in Figure 25 & 26. 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.*

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 $S = \{s_i\}$:

$$\tilde{\mu}(s) = 1 - \mathbf{k}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{k}/k(s,s) \tag{19}$$

, 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.

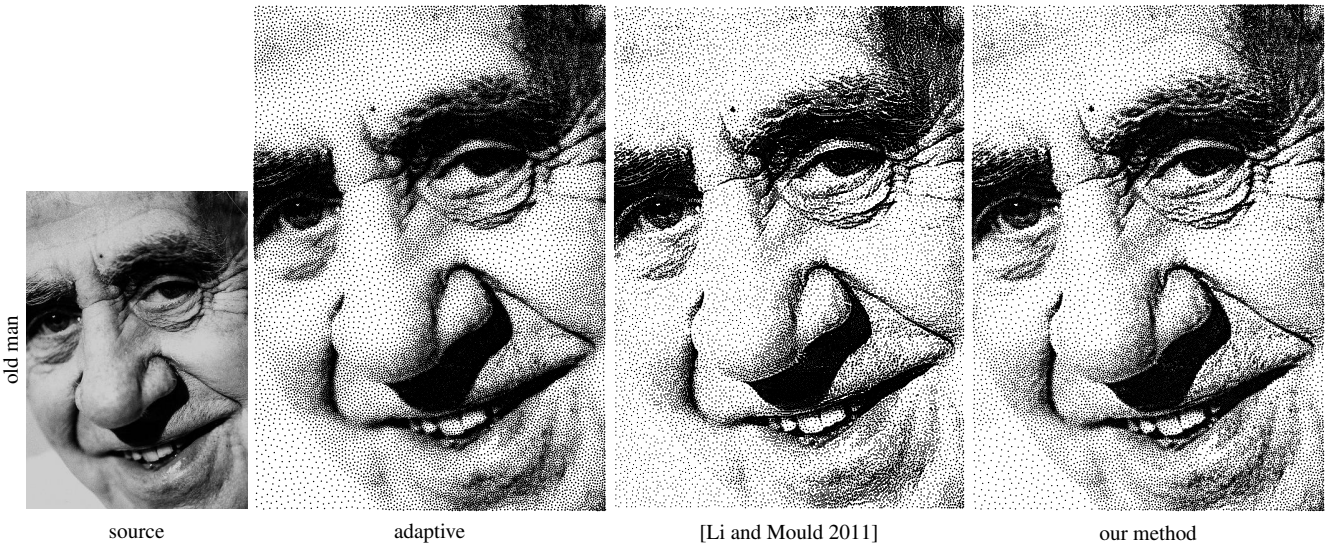


Figure 20: *More image stippling results with variable stipple size. Similar to Figure19, but using variable stipple sizes.*

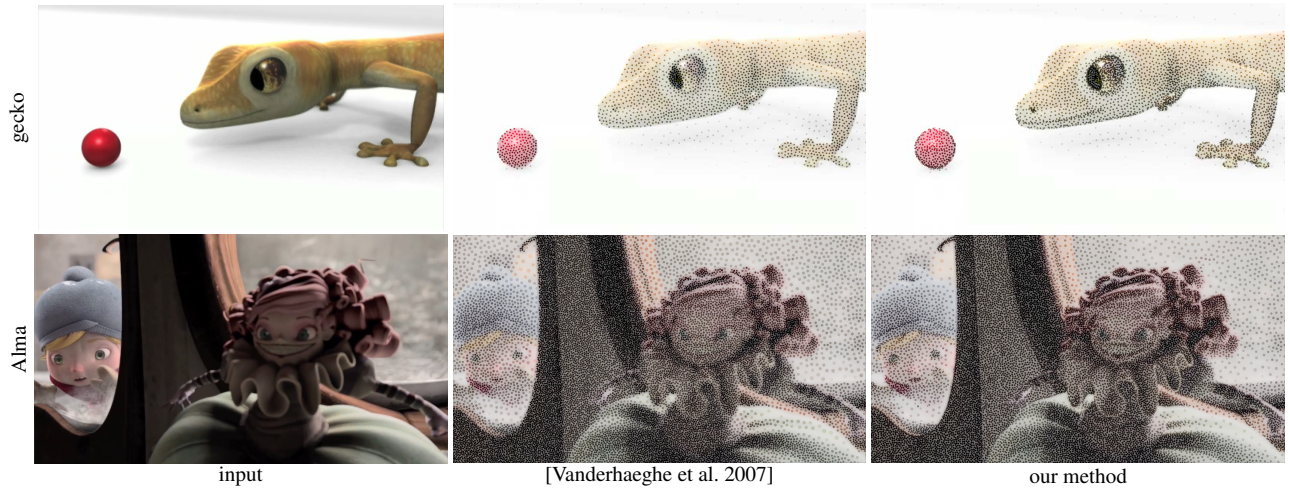


Figure 21: *Spatial-temporal sampling for video stylization. Notice the better feature preservation of our method, such as the gecko mouth and eye, and the doll face and clothing. Each frame of gecko\Alma contains $\sim$ 2200\19000 samples. Please refer to the accompany video for animation effects.*

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 2 and Figure 25 & 26. Notice that our method has comparable numerical measures for feature preservation (H in Table 2) and better blue noise properties (higher ρ in general, more uniform spectrum profiles in Figure 25, and less anisotropy in Figure 26). 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 27 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.

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 3: *Statistics for for geometry sampling results in Figure 30 & 31. See Table 2 for more details.*

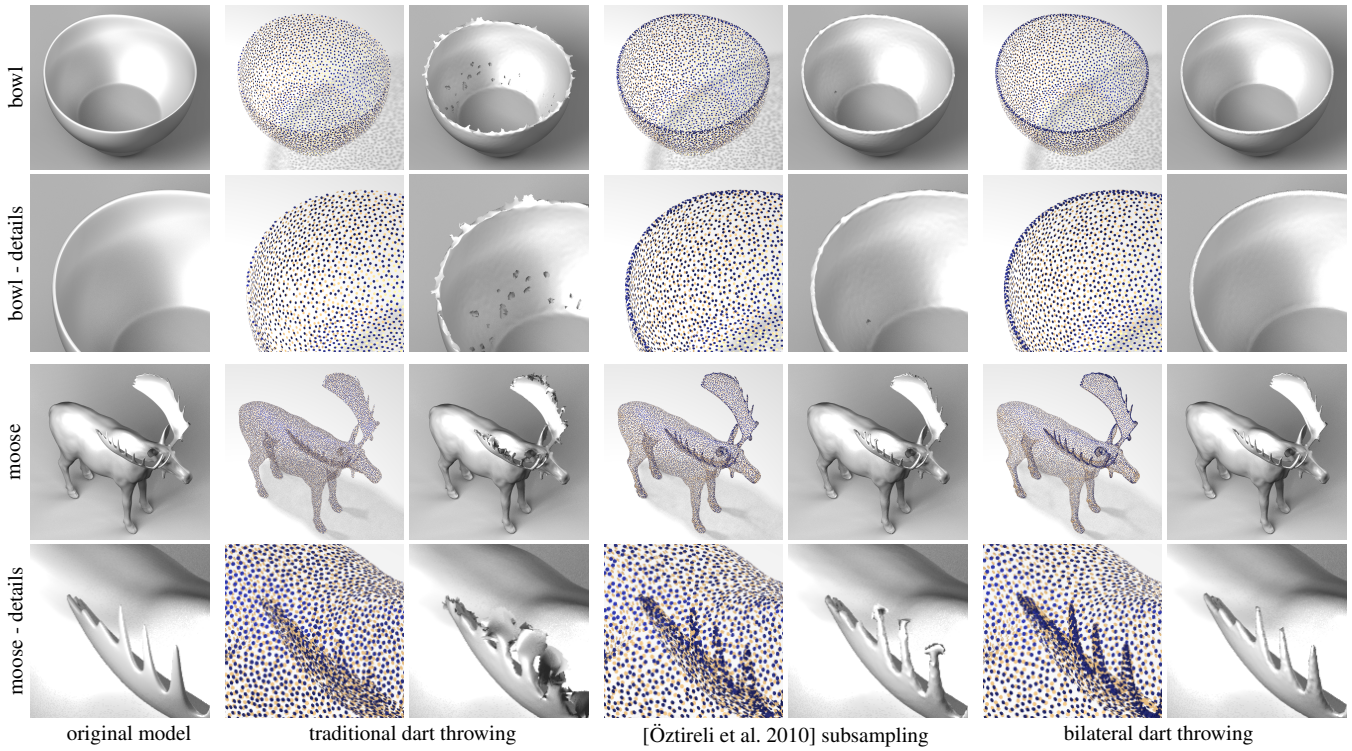


Figure 22: *Resampling point clouds through dart throwing. Our bilateral blue noise sampling method can achieve a uniform sample distribution even in the case of proximal surfaces. We compare our results with traditional blue noise (only considers spatial information) and the subsampling method discussed in [Öztireli et al. 2010]. In both the bowl and moose cases, bilateral blue noise sampling achieves a more uniform sample distribution and hence a more accurate surface reconstruction (using RIMLS in [Öztireli et al. 2009]). For the bowl/moose case, the input point cloud has* $\sim 400K/\sim 800K$ *samples and each output result has* $\sim 6K/\sim 20K$ *samples.*

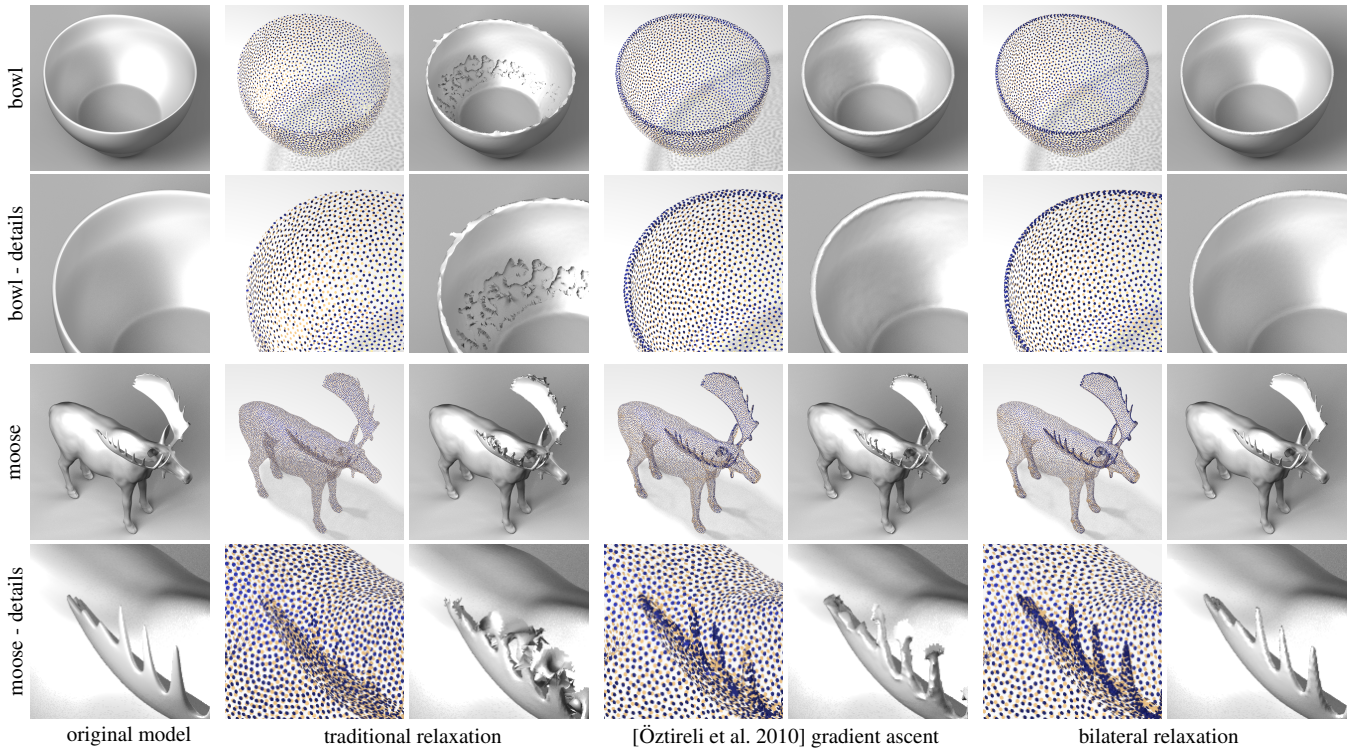


Figure 23: *Resampling point clouds through relaxation following the dart throwing results in Figure 22.*

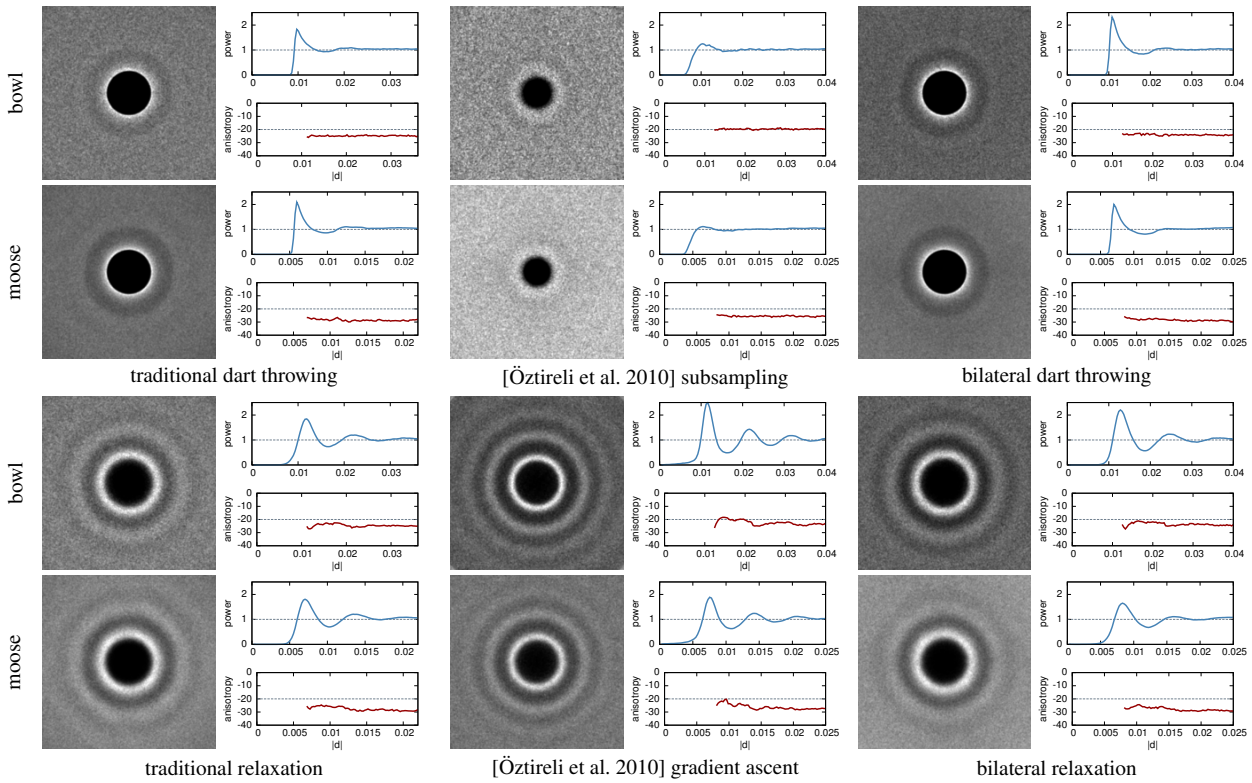


Figure 24: *Differential domain analysis for surface sampling. Here we show the DDA [Wei and Wang 2011] spectrum, radial mean, and radial anisotropy for the point sets in Figure 22 and Figure 23. Compared to the methods in [Öztireli et al. 2010], our methods have better blue noise properties, such as more uniform distributions in dart throwing versus sub-sampling (upper group) and the lower amount of anisotropy in general.*

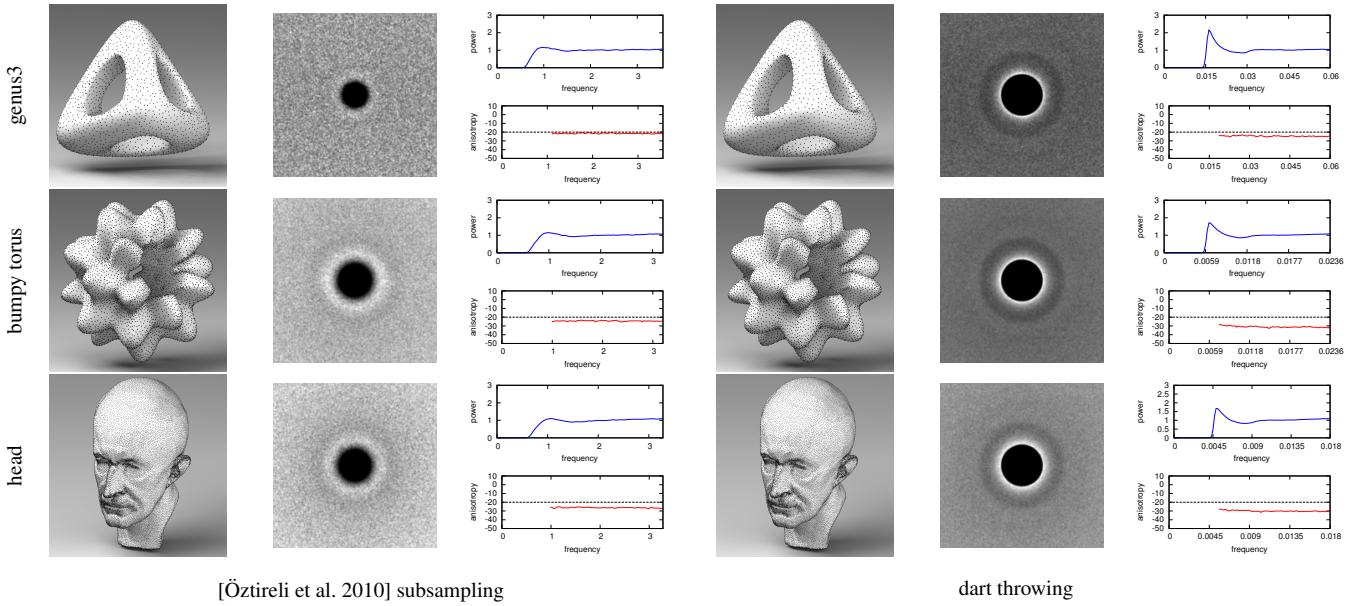


Figure 25: *Blue noise properties of geometry sampling via dart throwing. Shown here are sampling results of different domains with varying topological and geometrical properties. Detailed statistics for each case can be found in Table 2. Shown in each group are the spatial samples, DDA spectrum, and its radial mean and anisotropy [Wei and Wang 2011]. Notice the more uniform distributions of our results (larger inner ring of the DDA spectrums).*

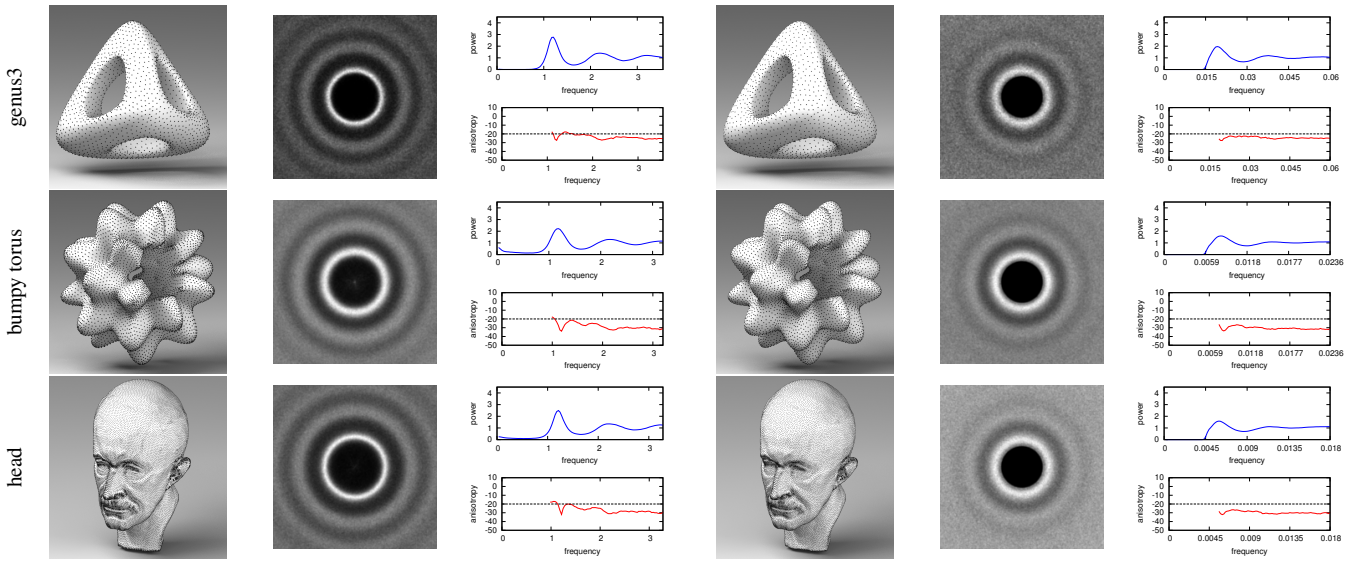


Figure 26: *Blue noise properties of geometry sampling via relaxation/gradient-ascent following the results from Figure 25. We run 10 iterations of CCVT [Balzer et al. 2009] for relaxation. Notice the lower amount of anisotropy in our results.*

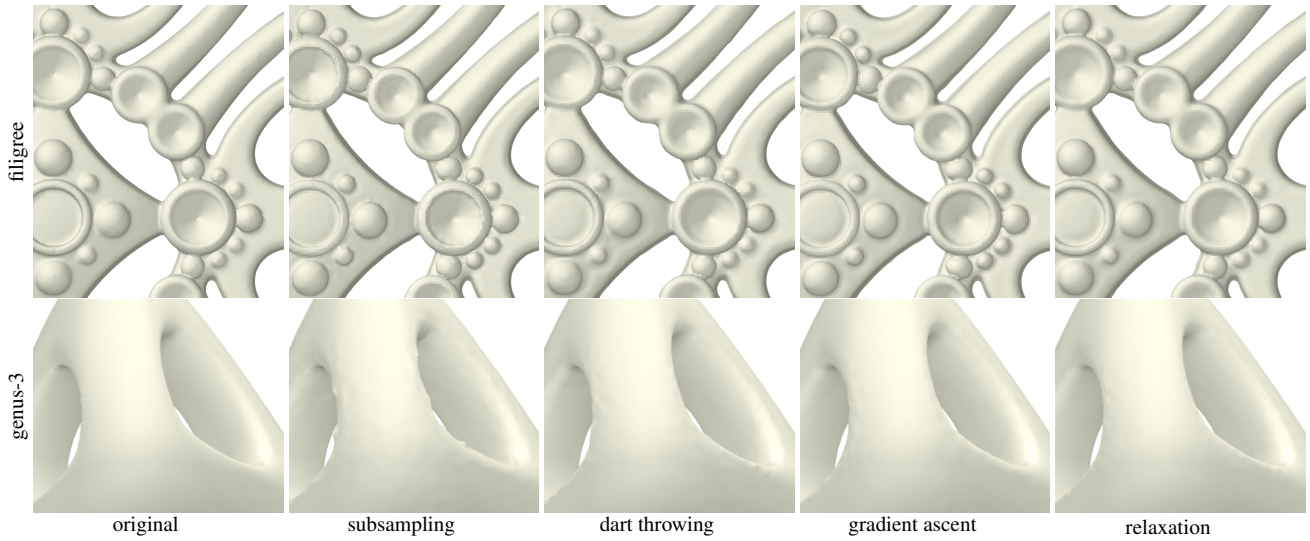


Figure 27: *Visual quality comparison. Here, we display the surface reconstruction results from down-sample sets generated by different methods (surfaces are reconstructed using the same PSS algorithm as in [Öztireli et al. 2010]). The filigree results (top row) are produced with about 84K samples each, 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 25 and 26, 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.*

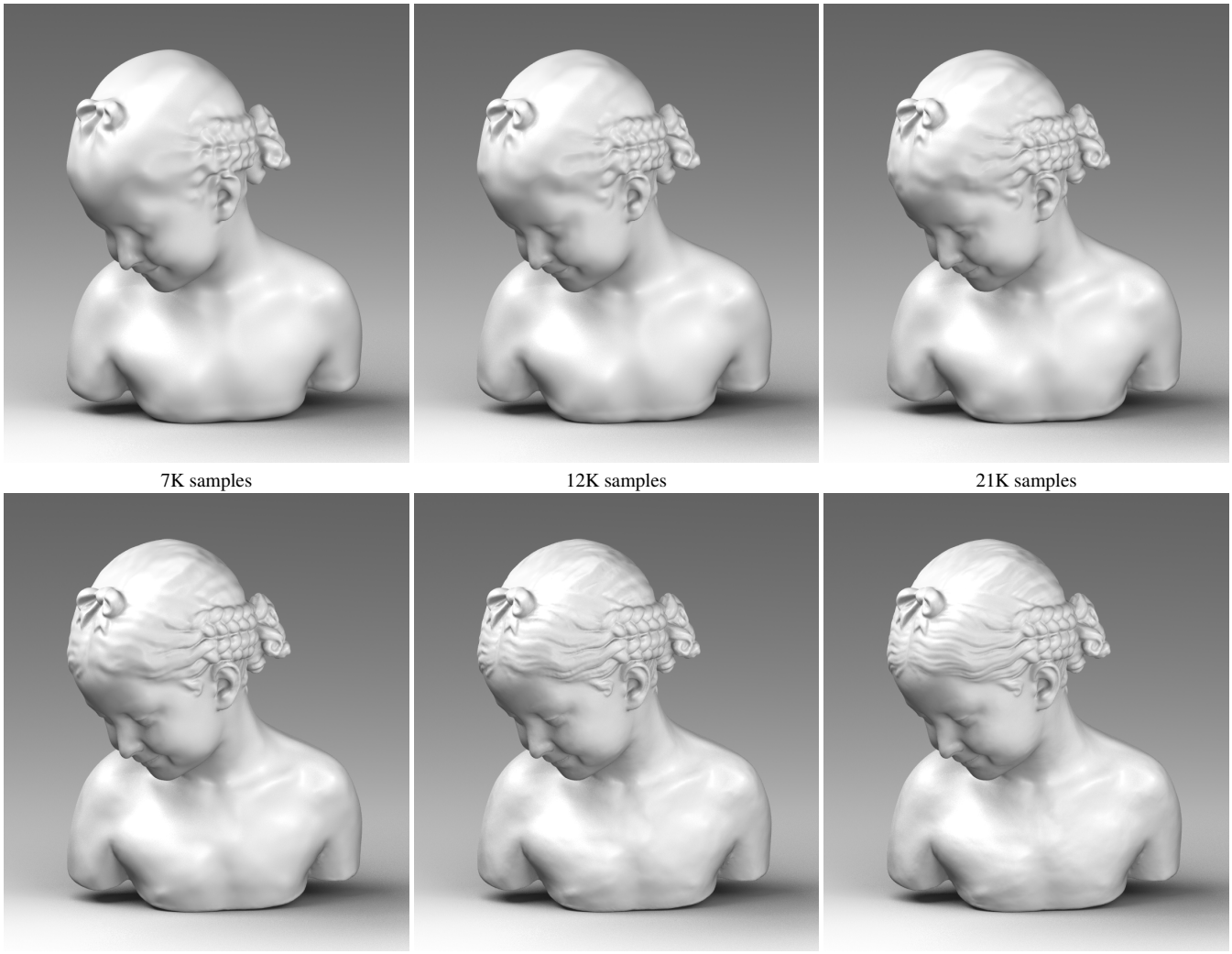
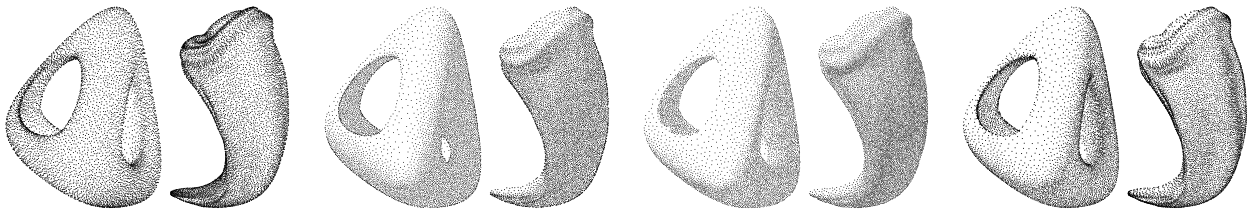


Figure 28: *Surface reconstruction of our results with varying sample rates. All results are produced by our bilateral dart throwing algorithm. As shown, our method can well preserve features under a variety of sampling rates.*



(a) *bilateral object* **(b)** *bilateral screen* **(c)** *cross blue noise* **(d)** *cross bilateral*

Figure 29: *Cross-dimensional dynamic sampling for 2D stylization of 3D objects. Here we compare our cross dimensional bilateral blue noise sampling (d) against alternative methods, including our bilateral sampling in either (a) object- or (b) screen-space only, as well as (c) cross dimensional (non-bilateral) blue noise sampling [Vanderhaeghe et al. 2007].* ***Object*** *space sampling (a) has non-uniform screen space distribution (notice highly distorted and non-uniform sample distributions near the silhouettes of the genus-3 model) and loses 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).* ***Screen*** *space sampling (b) can suffer from temporal coherence issues such as shower door or temporal discontinuity (see [Vanderhaeghe et al. 2007]). Cross dimensional (**non-bilateral**) blue noise sampling (c) does not consider or preserve well features. In comparison, our method (d) provides better effects, including shading/tone reproduction, features in image and object spaces, as well as temporal coherence. (Please refer to the accompany video for animation effects.) Each case of genus3/ claw contains* $\sim$ *4500/ 8000 visible samples. The input models are shown on the left.*

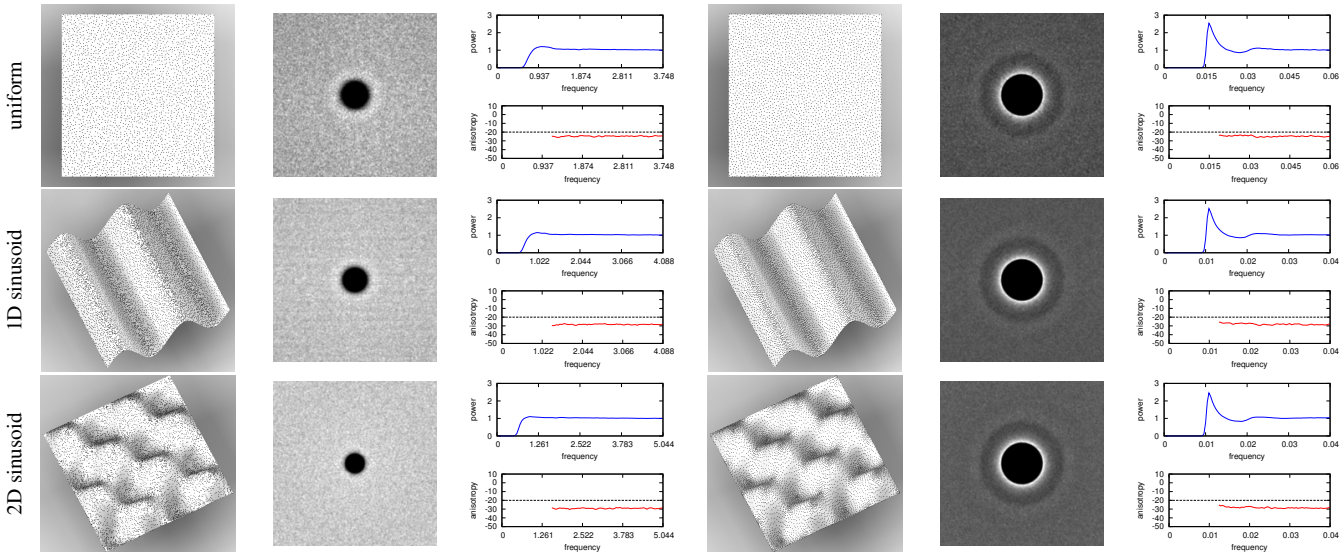


Figure 30: *Blue noise properties of geometry sampling via dart throwing. Shown here are more sampling results of different domains from Figure 25, including uniform, 1D sinusoidal* $0.15\sin(2\pi fx)$ *with* $f = 2$*, 2D sinusoidal* $0.15\sin(2\pi fx)\sin(2\pi fy)$ *with* $f = 2$*. Detailed statistics for each case can be found in Table 3.*

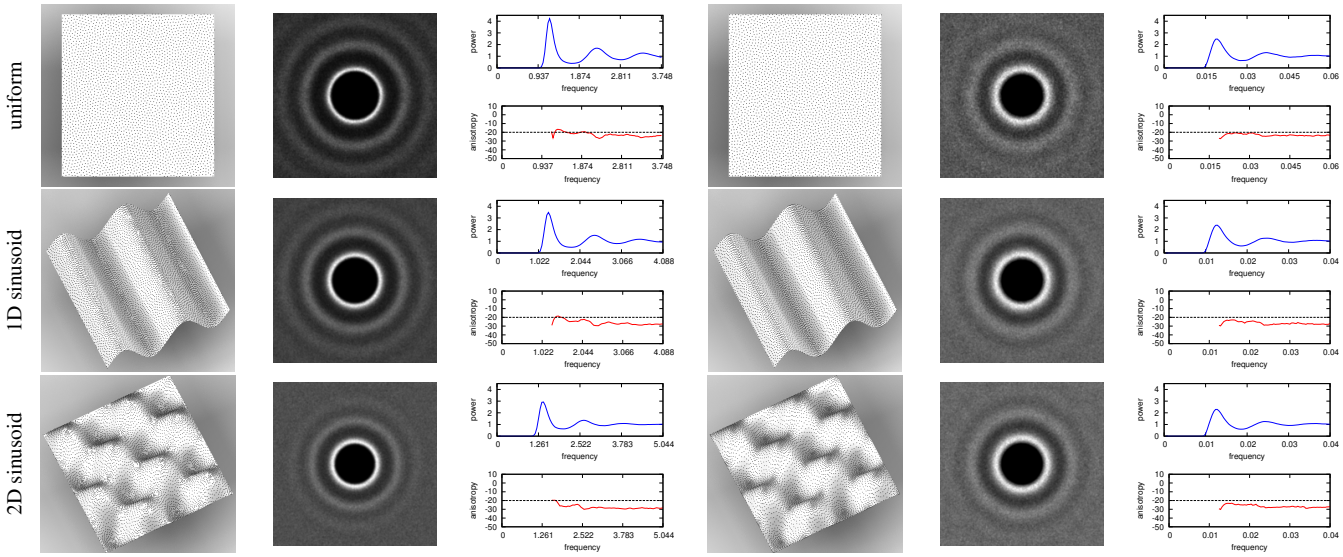


Figure 31: *Blue noise properties of geometry sampling via relaxation/gradient-ascent following the results from Figure 30.*