Bilateral Blue Noise Sampling: Additional Algorithms and Applications

Xiaoyin Ge, Li-Yi Wei, Yusu Wang, Huamin Wang

Bilateral Blue Noise Sampling: Additional Algorithms and Applications

Xiaoyin Ge[†] Li-Yi Wei[‡ §] Yusu Wang[†] Huamin Wang[†]

[†] Ohio State University [‡] University of Hong Kong [§] Microsoft Research

Abstract

Blue noise sampling is an important component in many graphics applications, but most of the prior techniques consider mainly the spatial positions of samples, making them less effective when handling features, such as image color and surface geometry. We apply the bilateral blue noise dart throwing method in [Chen et al. 2013] to graphics applications including image/video stippling and nonlinear filtering. We also propose a bilateral Lloyd relaxation method which works comparably with the kernel based bilateral relaxation method in [Chen et al. 2013].

Keywords: bilateral, feature, blue noise, sampling, filtering, stippling, image, video

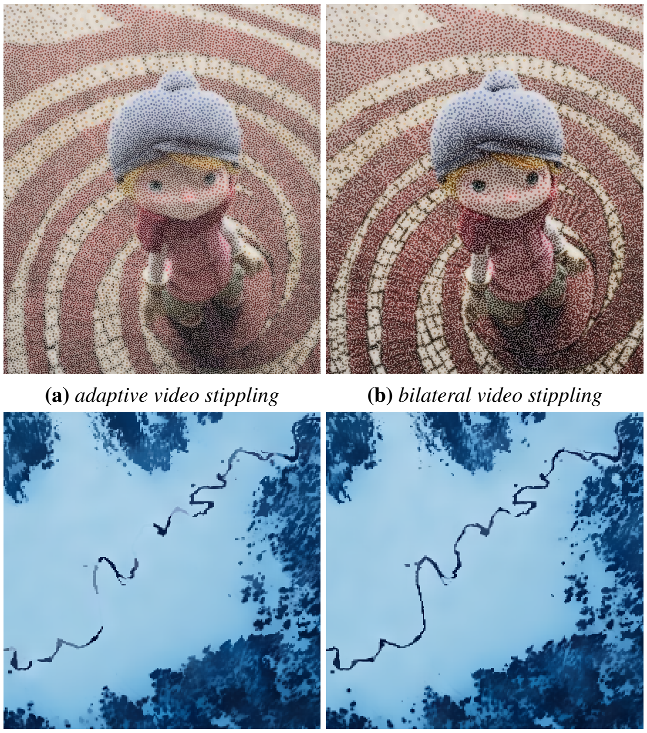
1 Introduction

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* Ω and its corresponding feature *range* Γ. Though 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 Ω is a 2D pixel grid, the range Γ is the pixel color, and the goal is to distribute the stipples as spatial blue noise and yet depicts the image colors well. As another example, in geometry sampling [Öztireli et al. 2010; Chen et al. 2013], 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.

Due to the fact that different applications have different range features, it is usually 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.[1]

Recently, [Chen et al. 2013] presented a general bilateral blue-noise sampling method, whose sampling results preserve both domain blue noise and range features. Their basic idea is a sample-distance measure that incorporates both sample positions and features preserving; and can be readily combined with prior blue noise sample analysis and synthesis algorithms.

In this technical report, we apply the bilateral dart throwing algorithm [Chen et al. 2013] for additional applications, including image stippling, dynamic stippling, and nonlinear filtering. Furthermore, we propose and demonstrate bilateral Lloyd relaxation method which can work comparably to the kernel based relaxation shown in [Chen et al. 2013].



(a) *adaptive video stippling* **(b)** *bilateral video stippling*

(c) *uniform image sampling* **(d)** *bilateral image sampling*

Figure 1: *Prior blue noise sampling versus bilateral blue noise sampling. (a) and (b) show the video stippling result of traditional adaptive blue noise sampling [Wei 2010] and bilateral blue noise sampling, respectively (Section 5.2) ; (c) and (d) show a comparison between the nonlinear filtering results with uniform blue noise sampling [Banterle et al. 2012] and bilateral blue noise sampling (Section 5.3). As shown, prior sampling methods may smooth out small yet important features (a) or miss them entirely (c).*

2 Previous Work

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 in numerous applications:

- rendering: [Cook 1986; Pharr and Humphreys 2004; Schlömer et al. 2011; Sun et al. 2013]
- geometry: [Turk 1992; Alliez et al. 2002; Öztireli et al. 2010; Bowers et al. 2010; Chen et al. 2013]
- image,visualization: [Balzer et al. 2009; Wei 2010; Li et al. 2010; Fattal 2011; Ebeida et al. 2011; de Goes et al. 2012]
- animation: [Schechter and Bridson 2012]

Importance adaptive sampling Non-uniform blue noise allows samples to be distributed conforming to some user defined density which is usually proportional to the importance of the domain regions. However a pure density-map-guided blue noise sampling may suffer from following problems: For one, a precise comput-

[1] Figure 1a and 1b are generated based on short film *Alma*; and Figure 1c and 1d are generated based on photo in *National Geographic*.

ing of such density map is usually not trivial. For two, importance sampling is likely to miss sharp or thin features with near zero area, even if some high importance value is assigned. Besides, 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.

Bilateral 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 work, in which it also attempts to keep blue noise properties as a by-product of feature-preservation.

Bilateral blue noise sampling introduced in [Chen et al. 2013] aims at preserving both blue noise properties and features. The basic idea behind bilateral blue noise sampling is to define a bilateral difference $\boldsymbol{\xi}(s_i, s_j)$ between pairs of samples s_i and s_j as a combination of the positional difference $\mathbf{d}(\mathbf{p}_i, \mathbf{p}_j) = \mathbf{p}_i - \mathbf{p}_j$ in Euclidean space Ω; and a (dis)similarity measure $\mathbf{f}(\mathbf{v}_i, \mathbf{v}_j)$ of non-spatial sample attributes(feature) $\mathbf{v}_i$ and $\mathbf{v}_j$ in range Γ:

$$\boldsymbol{\xi}(s_i, s_j) = \mathbf{d}(\mathbf{p}_i, \mathbf{p}_j) \circ \mathbf{f}(\mathbf{v}_i, \mathbf{v}_j), \tag{1}$$

where $\circ$ denotes a generic operator. [Chen et al. 2013] presented two bilateral distance metrics: augmentative form and multiplicative form. In the rest of this technical report, we will focus on the augmentative bilateral distance and its related applications.

Augmentative bilateral distance The augmentative bilateral distance was proposed in the form:

$$\boldsymbol{\xi}(s_i, s_j) = \left(\frac{\mathbf{d}(\mathbf{p}_i, \mathbf{p}_j)}{\sigma_p}, \frac{\mathbf{d}(\mathbf{v}_i, \mathbf{v}_j)}{\sigma_v} \right), \tag{2}$$

with following metric (L_2 norm):

$$\begin{aligned} \|\boldsymbol{\xi}(s_i, s_j)\|^2 &= d_p^2(s, s') + d_v^2(s, s') \\ &= \left\| \frac{\mathbf{p}(s) - \mathbf{p}(s')}{\sigma_p} \right\|^2 + \left\| \frac{\mathbf{v}(s) - \mathbf{v}(s')}{\sigma_v} \right\|^2 \end{aligned} \tag{3}$$

, in which $\mathbf{p}$ denotes sample's position vector, $\mathbf{v}$ denotes sample's feature vector, and σ_p and σ_v are the weights for position and feature, respectively. d_p is the spatial sample distance in traditional blue noise sampling, either Euclidean (e.g. image sampling, point cloud subsampling, or photon mapping) or Geodesic (e.g. surface sampling). d_v is the application-specific feature distance. For example, in geometry sampling, $\mathbf{v}$ can be set to be point cloud normal [Chen et al. 2013]. In image cases (Section 5), we have $\mathbf{v} = \mathbf{c}$, which is the pixel color (grayscale or RGB).

The combination of σ_p and σ_v provides a balance between the blue noise properties and feature preservation. When $\sigma_v \to \infty$, the bilateral blue noise sampling method is reduced to traditional blue noise sampling; and when σ_v gets smaller, more emphasis will be put onto features rather than blue noise.

Higher dimensional space interpretation If we regard $\boldsymbol{\xi}(s_i, s_j)$ in Equation 2 as a new vector in a higher-dimensional space Ψ formed by $\mathbf{p}$ and $\mathbf{v}$. Equation 3 basically calculates the distance between two samples in space Ψ. So that bilateral sampling with augmentative distance generates random samples over an embedded manifold M in Ψ, and the result is their projections to a lower-dimensional space, as Figure 2 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.

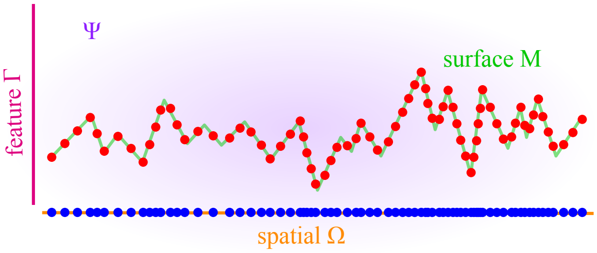


Figure 2: *Higher dimensional space interpretation. Taking height field as an example: given a spatial domain Ω (horizontal axis) with feature(height) range Γ (vertical axis), if we perform a uniform sampling (●) in the height field surface M embedded in a higher dimensional space Ψ, the corresponding samples (●) manifested in Ω can have non-uniform distributions induced by the features.*

3 Synthesis

The bilateral blue noise sampling can be applied to prior sampling methods by simply replacing the traditional position-only distance with augmentative distance Equation 3. Below we provide the bilateral Lloyd relaxation algorithm followed by a brief introduction of the bilateral dart throwing method in [Chen et al. 2013].

3.1 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 the bilateral distance d from Equation 3 in lieu of the spatial-only distance in traditional dart throwing. By using different $r(s, s')$ representations, bilateral blue noise sampling 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]).

3.2 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 the bilateral distance $d(s', s_i)^2 = \|\boldsymbol{\xi}(s', s_i)\|^2$ in Equation 3 instead of a pure spatial domain distance. Lloyd relaxation minimizes this energy 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:

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

. The augmentative bilateral distance can be directly plugged in here to substitute for d without changing the remaining 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 via the Jacobian approximation [Du and Wang 2005] or [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:

$$\hat{\mathbf{p}} = \frac{\mathbf{p}}{\sigma_p},\ \hat{\mathbf{v}} = \frac{\mathbf{v}}{\sigma_v} \tag{8}$$

$$\mathbf{J} = \begin{pmatrix} \mathbf{J}_{\mathbf{p}}(\hat{\mathbf{p}}) \\ \mathbf{J}_{\mathbf{p}}(\hat{\mathbf{v}}) \end{pmatrix} \tag{9}$$

, where $\mathbf{J}_{\mathbf{p}}(\hat{\mathbf{p}})$ and $\mathbf{J}_{\mathbf{p}}(\hat{\mathbf{v}})$ indicate the Jacobian of $\hat{\mathbf{p}}$ relative to $\mathbf{p}$, and Jacobian of $\hat{\mathbf{v}}$ relative to $\mathbf{p}$, respectively. 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.

Note that even though above centroid step is only approximate, it works well when the sampling density is sufficiently high (relative to domain variations), as discussed in [Li et al. 2010].

4 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.

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 Equation 9. However, the Jacobian in Equation 9 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. This can be addressed 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

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

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

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

, 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

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

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

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

, 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.

5 Results

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

5.1 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 best knowledge, prior methods 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]); and they might not offer enough flexibility in controlling the relative weights between the blue noise and feature preservation (e.g. [Chang et al. 2009; Li and Mould 2011]); besides, most of them operate in discrete grids (e.g. [Pang et al. 2008; Chang et al. 2009]) and might not be appropriate 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.)

Bilateral blue noise sampling (dart throwing and relaxation) can be applied for such feature-aware image stippling or halftoning by simply using gray-scale image color $\mathbf{c}$ as features $\mathbf{v}$ in Equation 3. It is applicable to both discrete and continuous domains, and 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 3. As shown in Figure 3, the bilateral method preserves features and blue-noise-properties better than prior techniques.

Metric comparison Figure 4 compares the bilateral distance in Equation 3 with other conflict metrics such as mean-conflict [Wei

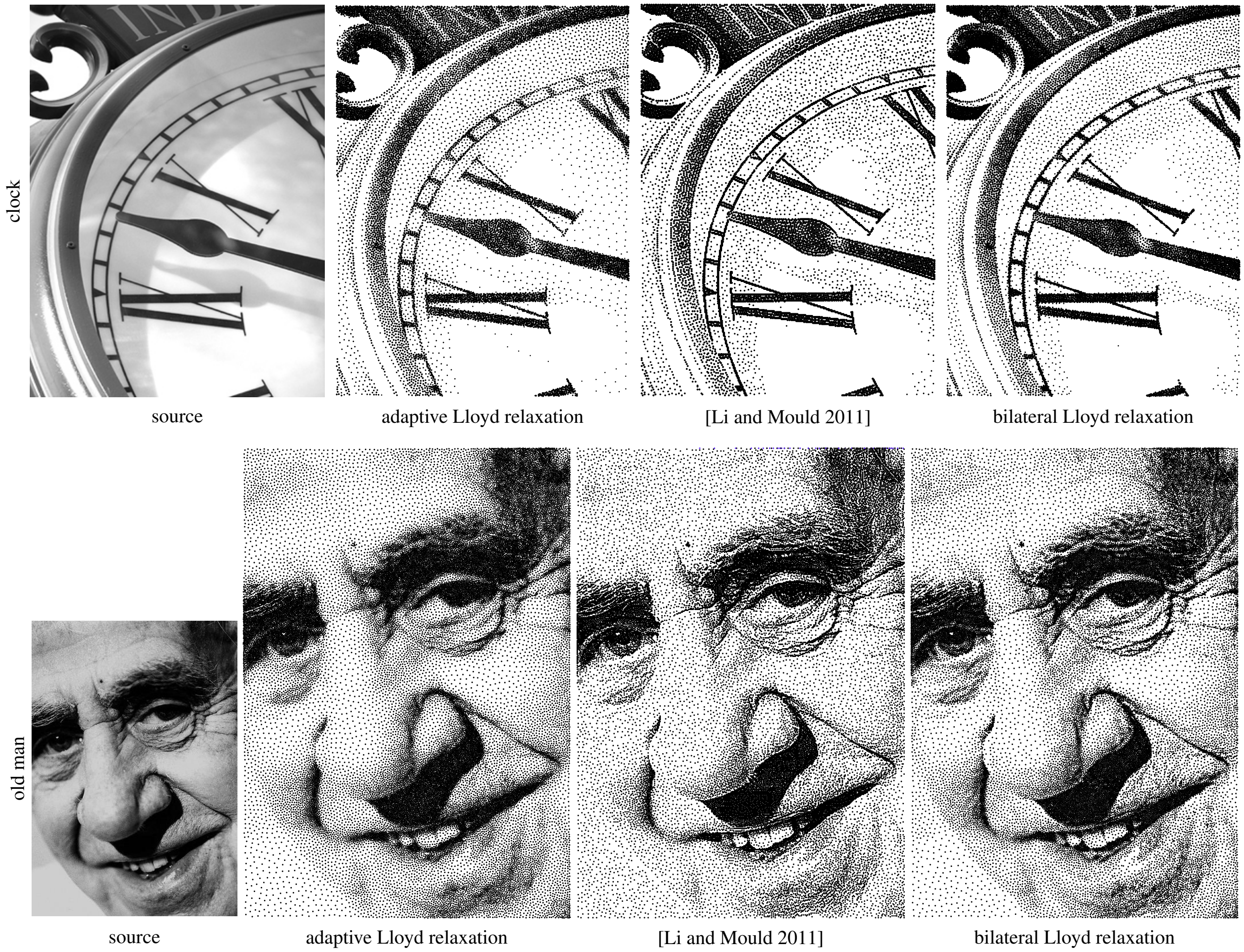

clock
source
adaptive Lloyd relaxation
[Li and Mould 2011]
bilateral Lloyd relaxation
old man
source
adaptive Lloyd relaxation
[Li and Mould 2011]
bilateral Lloyd relaxation

Figure 3: *Image stippling results with bilateral Lloyd relaxation. We compare bilateral Lloyd relaxation method (our method) against traditional adaptive Lloyd relaxation 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 Lloyd relaxation in feature preservation, and [Li and Mould 2011] in tones, 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. a more uniform sample distribution).*

2008] and min-conflict [Kalantari and Sen 2011] for dart throwing. As shown, comparing with these methods, the result generated by bilateral blue noise sampling has a better quality, especially around features. Notice, to keep the fairness in the comparisons, we generate all results in Figure 4 by dart throwing method (i.e., no relaxation is included).

5.2 Dynamic Stippling

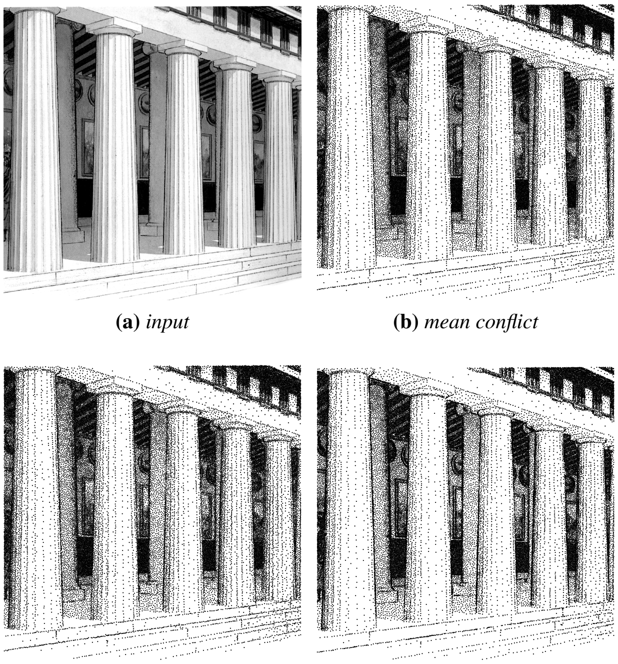
Bilateral blue noise sampling can be applied to numerous dynamic effects, such as sprite-based animation [Yu et al. 2009], video stippling [Chen et al. 2012], and hybrid point distribution [Vanderhaeghe et al. 2007]. The key, as stated in [Vanderhaeghe et al. 2007], is to maintain the balance between 2D spatial blue noise, 1D temporal coherence, and 2D (video) or 3D (object) motion depiction. This is a challenging problem due to both 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, meanwhile maintains the feature preserving. (Please refer to the accompany video of Figure 5 & 6 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 $\mathbf{n}(s')$:

$$s' = raycast(s),\ \mathbf{v}(s) = \big(\alpha\mathbf{c}(s), \mathbf{n}(s')\big) \tag{14}$$

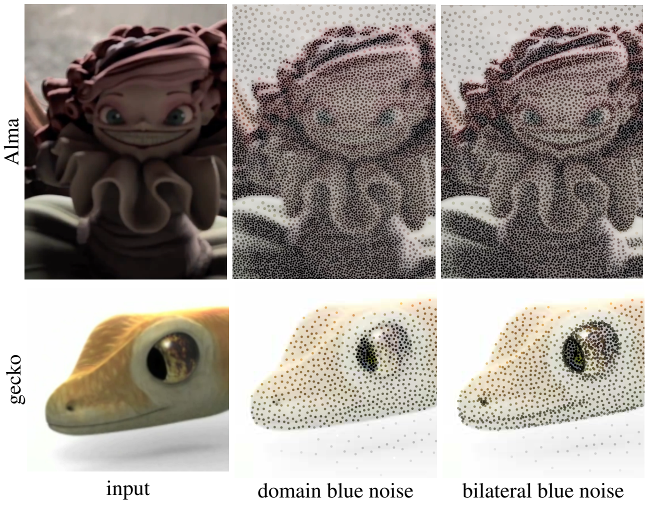
, where α is the relative weight between $\mathbf{c}$ and $\mathbf{n}$, s' the 3D object surface point corresponding to s (obtained through ray casting from the eye point), and $\mathbf{n}$ is the 3D object normal in the eye coordinate system. (If $raycast(s)$ does not hit the object surface, we set $\mathbf{n}(s')$ to $\mathbf{0}$.) This choice allows us to emphasize both shading and projected geometry features better than considering only 3D geometry



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

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

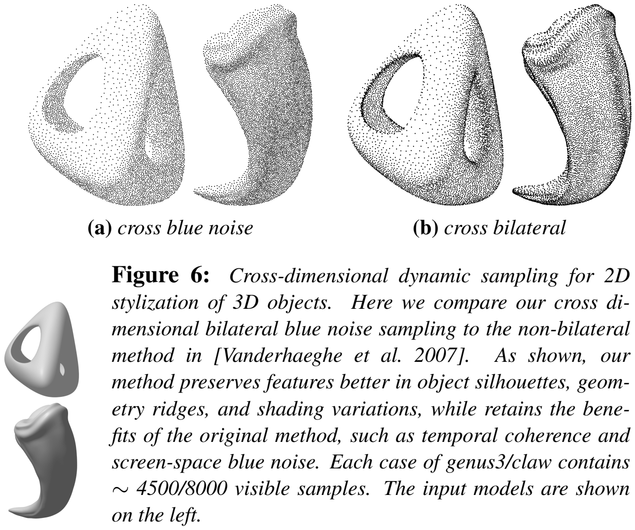
Figure 4: *Distance metric comparison. Here, we compare our bilateral distance metric with alternative measures including (b) mean conflict [Wei 2008] and (c) min conflict [Kalantari and Sen 2011]. Each case contains about 35K samples produced by dart throwing. Notice that mean conflict metric can produce 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, with the cost of a less uniform distribution in non-feature area.*



input domain blue noise bilateral blue noise

Figure 5: *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.*

or 2D image features. Some examples include interior and exterior silhouettes (e.g. the genus 3 model in Figure 6) 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 6).



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

Figure 6: *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 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.*

5.2.1 Video stylization

Figure 5 demonstrates our spatial-temporal sampling application for video stylization[2]. 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 5, 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.

5.2.2 Cross-dimensional sampling

Figure 6 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. For tone reproduction, we set the local adaptive-sampling radius $r(s)$ according to the shading value.

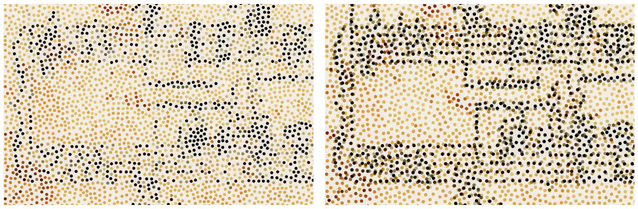
5.3 Nonlinear Filtering

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

[2] The original *Alma* video clip comes from the short movie *Alma* in: `http://almashortfilm.com`; the original *Gecko* video clip comes from `Blender.org`.

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

Figure 7: *Filter sample distributions around the upper-left corner of the music sheet in Figure 8. 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 Equation 3, 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 independently 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 7), from which samples falling within individual output kernels are used during filtering.

In the following, we compare uniform/bilateral × local/global sampling methods for bilateral filtering via quality (Figure 8 & Figure 9) and performance (Table 1). Based on these, we have found that the bilateral + global method offers the best tradeoff.

Quality Bilateral sampling tends to produce better quality than uniform sampling, especially under sparse sample distributions. This is illustrated in Figure 8 for perceptual image quality, and Figure 9 for numerical error measurements under a variety of images and parameter settings[3]. 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 8e), 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

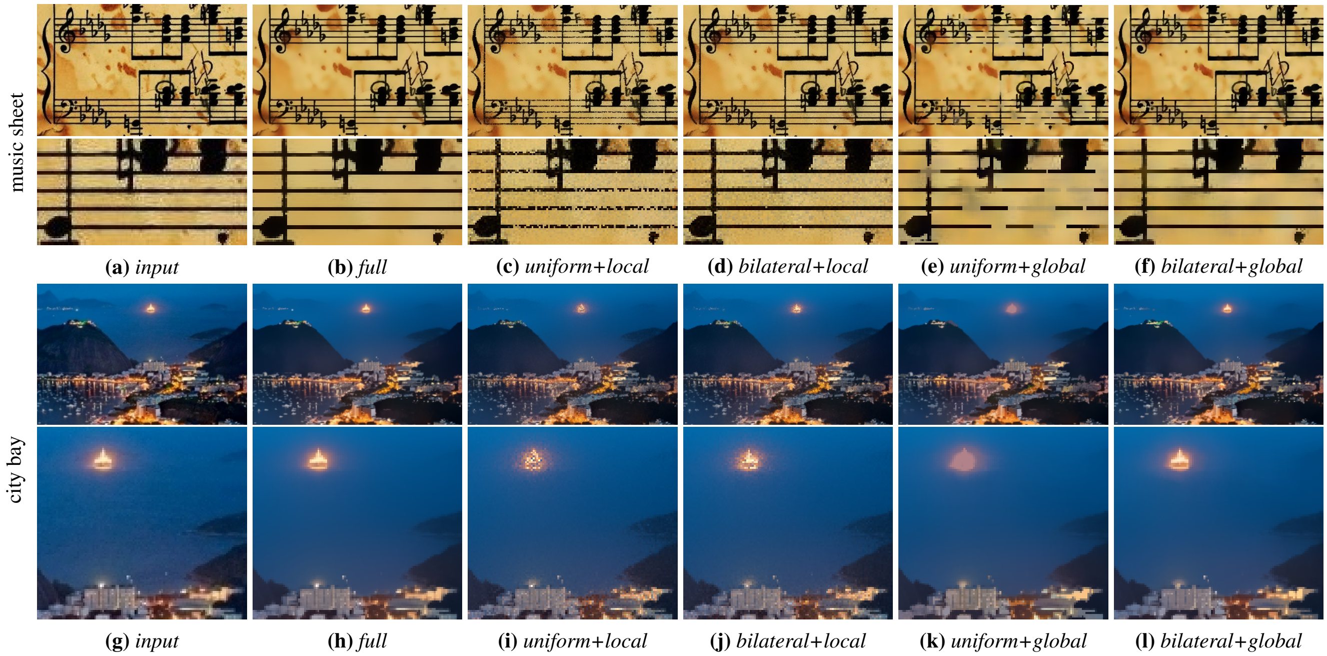
global bilateral			
	$N_s = 0.6K$	$N_s = 1.2K$	$N_s = 1.8K$
sampling	0.30 /0.15/0.04	0.93/0.33/0.11	1.74 /0.59/0.19
Voronoi	0.22/0.15/0.04	0.44/0.26/0.10	0.74/0.37/0.18
filtering	2.74/5.30/9.48	5.26/9.85/19.34	7.74/14.96/28.53
total	3.26/5.60/9.56	6.63/10.44/19.55	10.23/15.93/28.90

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.*

6 Limitations and Future Work

We have not yet attempted to combine 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].

[3] The input *music sheet* picture comes from `http://s729.photobucket.com/user/coryodonnell/library/`; and the *city bay* picture comes from *National Geographic*.



(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 8: *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* $N_s = 0.5K$.

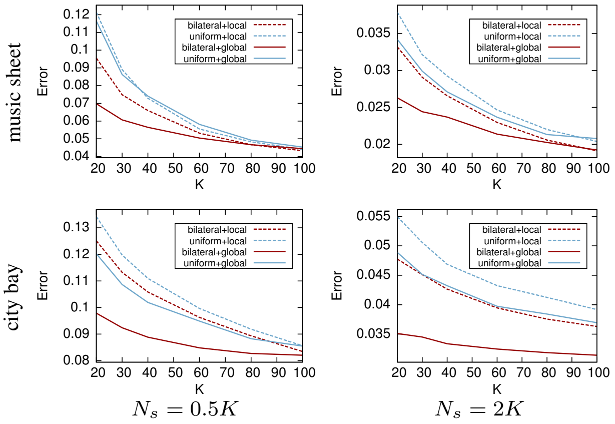


Figure 9: *Accuracy evaluation of different sub-sampling methods versus different kernel sizes* K*. Shown here are the RMSE relative to the ground truth full bilateral filtering of uniform/bilateral × local/global , respectively, over the music sheet/city bay input in Figure 8. The left/right columns 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.*

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.

In our current dynamic stippling application, to keep a better temporal coherence and speed up the process, we avoid to use any relaxation process in our current framework. It would be an interesting future work to improve the temporal coherence with Lloyd relaxation or kernel based relaxation in [Chen et al. 2013].

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].

References

ADAMS, A., GELFAND, N., DOLSON, J., AND LEVOY, M. 2009. Gaussian KD-trees for fast high-dimensional filtering. *SIGGRAPH '09, ACM Trans. Graph. 28*, 3 (July), 21:1–21:12.

ALLIEZ, P., MEYER, M., AND DESBRUN, M. 2002. Interactive geometry remeshing. *SIGGRAPH '02, ACM Trans. Graph. 21*, 3 (July), 347–354.

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

BALZER, M., SCHLÖMER, T., AND DEUSSEN, O. 2009. Capacity-constrained point distributions: a variant of Lloyd's method. *SIGGRAPH '09, ACM Trans. Graph. 28*, 3 (July), 86:1–86:8.

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.

BOWERS, J., WANG, R., WEI, L.-Y., AND MALETZ, D. 2010. Parallel Poisson disk sampling with spectrum analysis on surfaces. *SIGGRAPH Asia '10, ACM Trans. Graph. 29*, 6 (Dec.), 166:1–166:10.

CHANG, J., ALAIN, B., AND OSTROMOUKHOV, V. 2009. Structure-aware error diffusion. *SIGGRAPH Asia '09, ACM Trans. Graph. 28*, 5 (Dec.), 162:1–162:8.

CHEN, J., PARIS, S., AND DURAND, F. 2007. Real-time edge-aware image processing with the bilateral grid. *SIGGRAPH '07, ACM Trans. Graph. 26*, 3 (July).

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

CHEN, J., GE, X., WEI, L.-Y., WANG, B., WANG, Y., WANG, H., FEI, Y., QIAN, K.-L., YONG, J.-H., AND WANG, W. 2013. Bilateral blue noise sampling. In *SIGGRAPH Asia '13*, to appear.

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. *SIGGRAPH Asia '12, ACM Trans. Graph. 31*, 6 (Nov.), 171:1–171:11.

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

DU, Q., AND WANG, D. 2005. Anisotropic centroidal Voronoi tessellations and their applications. *SIAM J. Sci. Comput. 26*, 737–761.

EBEIDA, M. S., DAVIDSON, A. A., PATNEY, A., KNUPP, P. M., MITCHELL, S. A., AND OWENS, J. D. 2011. Efficient maximal Poisson-disk sampling. *SIGGRAPH '11, ACM Trans. Graph. 30*, 4 (July), 49:1–49:12.

FATTAL, R. 2011. Blue-noise point sampling using kernel density model. *SIGGRAPH '11, ACM Trans. Graph. 30*, 4 (July), 48:1–48:12.

GAMITO, M. N., AND MADDOCK, S. C. 2009. Accurate multi-dimensional Poisson-disk sampling. *ACM Trans. Graph. 29*, 1 (Dec.), 8:1–8:19.

GASTAL, E. S. L., AND OLIVEIRA, M. M. 2012. Adaptive manifolds for real-time high-dimensional filtering. *SIGGRAPH '12, ACM Trans. Graph. 31*, 4 (July), 33:1–33:13.

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*.

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. L_p centroidal Voronoi tessellation and its applications. *SIGGRAPH '10, ACM Trans. Graph. 29*, 4 (July), 119:1–119:11.

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. V., AND FU, C.-W. 2010. Anisotropic blue noise sampling. *SIGGRAPH Asia '10, ACM Trans. Graph. 29*, 6 (Dec.), 167:1–167:12.

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

ÖZTIRELI, A. C., AND GROSS, M. 2012. Analysis and synthesis of point distributions based on pair correlation. *SIGGRAPH Asia '12, ACM Trans. Graph. 31*, 6 (Nov.), 170:1–170:10.

ÖZTIRELI, A. C., ALEXA, M., AND GROSS, M. 2010. Spectral sampling of manifolds. *SIGGRAPH Asia '10, ACM Trans. Graph. 29*, 6 (Dec.), 168:1–168:8.

PANG, W.-M., QU, Y., WONG, T.-T., COHEN-OR, D., AND HENG, P.-A. 2008. Structure-aware halftoning. *SIGGRAPH '08, ACM Trans. Graph. 27*, 3 (Aug.), 89:1–89:8.

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. *SIGGRAPH '12, 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.

SUN, X., ZHOU, K., GUO, J., XIE, G., PAN, J., WANG, W., AND GUO, B. 2013. Line segment sampling with blue-noise properties. *ACM Trans. Graph. 32*, 4, 127:1–127:14.

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. *SIGGRAPH '11, ACM Trans. Graph. 30*, 4 (July), 50:1–50:10.

WEI, L.-Y. 2008. Parallel Poisson disk sampling. *SIGGRAPH '08, ACM Trans. Graph. 27*, 3 (Aug.), 20:1–20:9.

WEI, L.-Y. 2010. Multi-class blue noise sampling. *SIGGRAPH '10, ACM Trans. Graph. 29*, 4 (July), 79:1–79:8.

WEISS, B. 2006. Fast median and bilateral filtering. *SIGGRAPH '06, ACM Trans. Graph. 25*, 3 (July), 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. *SIGGRAPH '12, ACM Trans. Graph. 31*, 4 (July), 76:1–76:11.