Anisotropic Blue Noise Sampling

Hongwei Li, Hong Kong UST
Li-Yi Wei, Microsoft Research
Pedro V. Sander, Hong Kong UST
Chi-Wing Fu, Nanyang Tech. Univ.

Abstract

Blue noise sampling is widely employed for a variety of imaging, geometry, and rendering applications. However, existing research so far has focused mainly on isotropic sampling, and challenges remain for the anisotropic scenario both in sample generation and quality verification. We present *anisotropic blue noise sampling* to address these issues. On the generation side, we extend dart throwing and relaxation, the two classical methods for isotropic blue noise sampling, for the anisotropic setting, while ensuring both high-quality results and efficient computation. On the verification side, although Fourier spectrum analysis has been one of the most powerful and widely adopted tools, so far it has been applied only to uniform isotropic samples. We introduce approaches based on warping and sphere sampling that allow us to extend Fourier spectrum analysis for adaptive and/or anisotropic samples; thus, we can detect problems in alternative anisotropic sampling techniques that were not yet found via prior verification. We present several applications of our technique, including stippling, visualization, surface texturing, and object distribution.

Keywords: blue noise sampling, anisotropy, dart throwing, relaxation, spectrum analysis, stippling, texturing, rendering

1 Introduction

Sampling is important for a wide range of graphics applications including imaging, rendering, and geometry processing. Although different applications may favor different sampling methods, blue noise sampling remains one of the most popular due to its unique spatial and spectrum properties [Cook 1986; Mitchell 1987; McCool and Fiume 1992; Dunbar and Humphreys 2006; Kopf et al. 2006; Ostromoukhov 2007; Wei 2008; Cline et al. 2009; Balzer et al. 2009; Wei 2010]. In the spatial domain, blue noise samples are randomly and uniformly distributed; this corresponds well to a variety of natural phenomena (e.g., retina cell distribution [Yellott 1983]) and has certain aesthetic appeal (e.g., stippling [Secord 2002; Hiller et al. 2003]). In the frequency domain, a blue noise spectrum has a signature lack of low frequency energy and absence of structural bias (hence the name blue noise). In essence, a blue noise sampling produces better results by replacing low frequency aliasing with high frequency noise, which tends to be less visually objectionable. Due to its importance, blue noise sampling has been widely studied with a variety of proposed methods, many of which are certain variations or combinations of two classical algorithms: dart throwing [Cook 1986] and relaxation [Lloyd 1983].

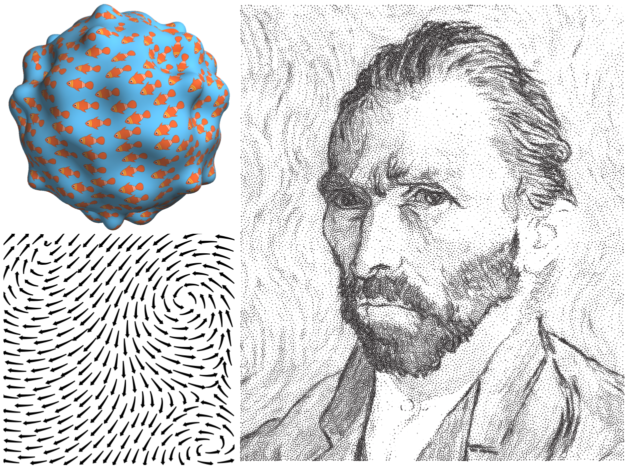


Figure 1: Anisotropic blue noise sampling. Applications include stippling, texturing, vector field visualization, and object distribution.

Despite the advances of blue noise sampling, except for rare exceptions (e.g., the pioneering work in [Feng et al. 2008]) most of these methods produce only isotropic sample distributions. This could be insufficient for application scenarios that benefit from anisotropic sampling. For example, in vector field visualization [Turk and Banks 1996], since the glyphs (e.g., arrows) are often elongated, it is more informative and visually pleasing to pack glyphs anisotropically instead of isotropically. In surface sampling, anisotropic surface elements have been shown to be more suitable for certain geometry processing applications [Alliez et al. 2003; Lévy and Liu 2010]. Furthermore, even for isotropic uniform surface sampling, the process may still become anisotropic if conducted via a parameterization domain instead of the original surface domain (e.g., as in [Alliez et al. 2002]) due to geometric distortion. Our anisotropic blue noise sampling methods can be applied to the aforementioned application scenarios, and provide better quality than prior techniques (Figure 1). In addition to visualization and surface sampling, other applications of our methods include stippling, surface texturing, and 3D objects placement with anisotropic shapes and distribution.

Since dart throwing and relaxation are the two commonly used approaches for blue noise sampling, we extend both of them for anisotropic sampling to provide a more general solution. We have developed a clean and elegant method that can be applied to both: *replace the Euclidean distance metric with a Riemannian/geodesic one*. For example, in dart throwing, instead of measuring the distance between two samples via Euclidean distance during conflict check as in isotropic sampling, we use a geodesic distance metric. The precise nature of the geodesic distance is application dependent, e.g., surface geodesics for manifold sampling or tensors for image stippling, but otherwise the algorithm remains identical across different applications. We have found this methodology to work better than alternative formulations, such as extending the conflict metric for anisotropic sampling. Similarly, for relaxation, we have found that the use of geodesic distance metrics allows us to easily extend different flavors of methods, including basic Lloyd relaxation [Lloyd 1983] and CCVT [Balzer et al. 2009].

Once anisotropic sample sets have been generated, we need a method to properly evaluate their quality. For uniform isotropic sampling, Fourier power spectrum has been demonstrated to be a very effective evaluation method, especially for detecting biases that are difficult to find by other approaches [Lagae and Dutré 2008]. However, a direct application of Fourier power spectrum for a set of adaptive or anisotropic samples is not going to produce very meaningful results, as the spatial or angular variations could clutter vital information in the spectrum. To be able to apply Fourier spectrum analysis for anisotropic samples, we introduce two approaches based on 2D warping and sphere surface sampling to produce samples in an anisotropic domain while verifying samples in a known uniform isotropic one. These shed light on problems in previous anisotropic sampling techniques (e.g., [Feng et al. 2008]) that are not yet detected by existing verification methods.

2 Distance Metric

To carry on our discussion, we first describe our distance metric d, a shared component between our anisotropic sampling algorithms to be presented in Section 3.

2.1 Representation

To generate a sampling pattern, the user usually needs to specify two kinds of metrics: the control metric r, and the distance metric d. The control metric r reflects the user preferences of the sampling pattern, such as density, adaptivity, or anisotropy. For the isotropic case, it usually takes the form of minimum sample spacing in dart throwing, or sample density in relaxation. The distance metric d reflects properties of the underlying sample space Ω, essentially defining the distance between any pairs of samples within Ω.

Here, we describe the duality between r and d, and suggest that it is better to absorb r into d for both algorithm design and implementation. This results in a geodesic d formulation, caused by not only the geometry of the underlying sample space but also the properties of r. For clarity of presentation, we focus on dart throwing below, as analogous arguments could be made for relaxation as well.

For uniform isotropic dart throwing, the user specifies a parameter r, a constant dictating the minimum distance between pairs of samples. For adaptive isotropic sampling, the user specifies a scalar spatially varying function $r(s)$, which can be interpreted as the diameter of the topological disk centered at sample s. Following this line of classical formulation, we can deduce that for adaptive anisotropic sampling, since the conflict metric r could in general depend on both the relative distance and orientation between any two samples s_1 and s_2, its most general form is $r(s_1, s_2)$, specifying the minimum required distance between any two samples.

Geodesic d. In the classical formulation via r above, the distance metric $d(s_1, s_2)$ is the usual Euclidean distance. Even though it is totally legitimate to just use Euclidean distance and let r absorb all the application specific geometric complexities, we have found a general form of $r(s_1, s_2)$ more difficult to specify, analyze and compute. Taking uniform surface sampling as an example, how are we going to define $r(s_1, s_2)$ between any two samples s_1 and s_2 if we measure their distance $d(s_1, s_2)$ via the Euclidean definition? A more natural formulation would be to define $d(s_1, s_2)$ via the surface geodesic and let r be a global constant. For surface sampling, this geodesic formulation may seem quite straightforward, but it might not be so for other scenarios; we will give more examples on formulations in Section 5 when presenting different applications.

Wrapping r into d. Furthermore, since we already incorporate anisotropy into d rather than r, we might as well include the isotropic information for the adaptivity of r into d as well, and simply use a constant for r. We can achieve this by replacing the original distance by $\frac{d}{r}$ and replacing r with 1. This allows us to wrap both the angular and distance information into d, resulting in a cleaner algorithm formulation and code implementation. Furthermore, a unified representation of d facilitates an approximation for faster computation, as we shall see shortly in Section 2.2. One consequence of this unified representation is that our new d might become non-Euclidean if either the original distance is Riemannian or the original r is non-uniform. Another consequence, as stated above, is that r will no longer show up explicitly in our dart throwing and relaxation algorithms. For example, to determine whether two samples s and s' are in conflict in dart throwing, we will use the criterion $d(s, s') < 1$ instead of $d(s, s') < r$.

2.2 Approximation

Although it is possible to generate samples via our rigorous d definition (see, e.g., [Sethian 1996; Weber et al. 2008]), it could be overly complex and computationally demanding, as the problem could be as challenging as calculating the geodesic distance between pairs of samples. Fortunately, we have found it sufficient to approximate the geodesic distance from sample s_1 to s_2 with the following distance metric (see [Labelle and Shewchuk 2003]):

$$d(s_1, s_2) = \sqrt{(J(s_1)(s_2 - s_1))^{\mathrm{T}}(J(s_1)(s_2 - s_1))} \qquad (1)$$

where the superscript T indicates the transpose operator and J the Jacobian (applied locally to the domain-specific function used to determine the desired sample distance and anisotropy, e.g., the surface for mesh sampling). Note that by considering the local Jacobian, as s_1 and s_2 get closer, Equation 1 approaches the true geodesic distance between the two samples. In general, as long as J is smooth, or inversely, the desired sampling density is high, this approximation is accurate. One crucial advantage of this approximation is that it directly leads us to treat anisotropic samples as ellipses, analogous to the use of geometric disks for isotropic dart throwing. Although it is just an approximation, we have found this sufficient for our anisotropic applications. As we will soon demonstrate below, the approximated distance metric allows us to design simple and clean anisotropic extensions for both dart throwing and relaxation, which would have been much more complex if we had to rely on a strict geodesic distance. While we have found Equation 1 more than adequate under most circumstances, one has to keep in mind that this is only an approximation. We provide a detailed justification and analysis in Section 4.3.

3 Anisotropic Sampling

Here we describe our anisotropic sampling algorithms: white noise, dart throwing, and relaxation.

3.1 Anisotropic White Noise

A common component of our anisotropic blue noise sampling methods is an anisotropic white noise generator. To our best knowledge we are not aware of suitable prior methods. We take the following approach to generate each anisotropic white noise sample s: (1) uniformly draw a random sample s' from the domain Ω; (2) compute its Jacobian $J(s')$ and the inverse $J^{-1}(s')$; (3) draw a small sphere with radius r_δ centered on s', and warp it into an ellipse via $J(s')$; (4) draw another random sample s uniformly from the sphere centered at s' with radius r_{max}, the maximum radius of all $J(p)$ ellipses for $p \in \Omega$; and (5) if s is inside the $J(s')$ ellipse, warp it back to the original r_δ sphere via J^{-1} and accept it; otherwise, reject s and repeat the whole process.

The intuition behind our method is as follows. If we know how the anisotropic domain Ω is warped from a uniform domain Ω', we could first generate uniform white noise on the latter and warp the samples to the former (Figure 2). In reality, this knowledge is not available for most anisotropic domains, so our method essentially approximates this hypothetical process via local warps through the Jacobian J. Specifically, each (uniformly random) s' selects a local domain (red sphere on the right), which, after warping through J, has the correct size and shape in the (hypothetical) uniform domain

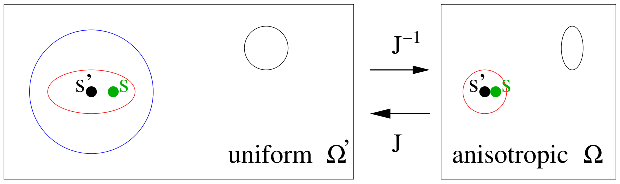


Figure 2: Anisotropic white noise. The blue/red sphere on the left/right has radius r_{max}/r_δ.

(red ellipse on the left). Thus, when r_δ is sufficiently small so that J is locally constant, our approach can draw s without bias. Also, r_{max} does not need to be very accurate as long as it is large enough to contain all J-transformed ellipses, even though a smaller/more-accurate r_{max} will make the process more efficient. This is analogous to the estimation of maximum density value in rejection-based importance sampling.

3.2 Anisotropic Dart Throwing

Dart throwing [Cook 1986] is a classical algorithm for generating isotropic blue noise samples. The output samples are generated sequentially one by one. To determine the location of the next sample, a trial sample s is randomly drawn from the domain. If s is at least a distance r away from all existing samples, it is accepted. Otherwise, it is rejected. This process is repeated until a sufficient number of output samples are produced, or a maximum number of iterations is reached. To extend traditional isotropic dart throwing for anisotropic sampling, we replace the Euclidean distance metric with our geodesic one in Equation 1, which also incorporates the r information as discussed in Section 2.

```
function $S \leftarrow$ AnisotropicDartThrowing($\Omega, r, J, N, MT$)
  // $\Omega$: sampling domain
  // $J$: Jacobian field
  // $N$: number of samples to be generated
  // $MT$: maximum number of trials
  define distance metric $d$ from $\Omega$ and $r$ // Section 2
  $S \leftarrow \emptyset$
  $trials \leftarrow 0$
  while $trials < MT$ and $|S| < N$
    $s \leftarrow$ white noise sample drawn from $\Omega$
    // conflict checking
    $conflicted \leftarrow$ false
    forall $s' \in S$
      if $d(s, s') < 1$ or $d(s', s) < 1$
        $conflicted \leftarrow$ true
      endif
    end
    // if no conflict then accept this sample
    if not $conflicted$
      add $s$ to $S$
      $trials \leftarrow 0$
    else
      $trials \leftarrow trials + 1$
    endif
  end
  return $S$
```

Program 1: Anisotropic dart throwing. See Equation 1 for the definition of d. Notice that our new distance metric d absorbs the functionality of the conflict metric r in classical formulation.

Conflict check metric. The centerpiece of our algorithm, as summarized in Program 1, is the conflict check. Since our distance metric (Equation 1) is not symmetric, namely $d(s_1, s_2) \neq d(s_2, s_1)$ in general, there are two potential ways to conduct the conflict check, referred to as mean and max conflict metric [Wei 2008]. In the mean conflict metric, we check whether the geometric shapes of s_1 and s_2 intersect each other, while in the max conflict metric, we check whether s_1 or s_2 is inside the "territory" of one another. Note that when d is symmetric, the mean and max conflict metrics are equivalent, e.g., for uniform isotropic sampling. We chose the max metric in our anisotropic sampling algorithm, because the mean conflict metric requires solving a quartic polynomial for ellipse intersection, which might suffer from numerical instability [Feng et al. 2008]. Also, the mean conflict metric cannot be easily extended to higher dimensions. We have found the spectral quality of the two metrics quite similar and chose the max metric due to its superior quality/performance tradeoff.

3.3 Anisotropic Relaxation

Lloyd relaxation [Lloyd 1983] is another classical method that has been applied to generating isotropic blue noise samples. Unlike dart throwing which draws samples one by one, 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:

$$\mathbf{E}(S, \mathcal{V}) = \sum_i \int_{p \in V_i} \rho(p) \left|p - s_i\right|^2 dp \tag{2}$$

where $\mathcal{V}$ is the Voronoi tessellation generated from S, V_i the Voronoi region corresponding to site $s_i \in S$, p a point in the domain Ω, and ρ a density function defined over Ω. Lloyd relaxation minimizes this energy function by iterating between the following two steps until meeting some termination criterion:

Voronoi generate the Voronoi tessellation $\mathcal{V}$ from the sample set S

Centroid move each site $s_i \in S$ to the centroid m_i of the corresponding Voronoi region $V_i \in \mathcal{V}$, i.e.,

$$m_i = \left(\int_{V_i} \rho(p) dp\right)^{-1} \int_{V_i} \rho(p) p dp \, . \tag{3}$$

Lloyd relaxation can be implemented in either a discrete or a continuous sample domain Ω, where the former represents the sample space Ω via a collection of discrete points P, while the latter is without such discretization. In this paper, we focus on the discrete formulation, as it is conceptually simpler and easier to formulate, especially for high dimensional Ω for which a continuous method could be tricky to implement. A discrete formulation can also achieve very similar quality to a continuous method via sufficient point density. In a discrete setting, Lloyd relaxation performs the **Voronoi** step by finding, for each discrete point $p \in \Omega$, the site $s(p)$ that is the closest to p among all sites in S:

$$s(p) = \arg\min_{s \in S} \left|p - s\right|^2 \, . \tag{4}$$

The Voronoi region V_i for site s_i is then defined as the collection of all points p whose site affiliation is s_i:

$$V_i = \bigcup \left\{p \in \Omega, s(p) = s_i\right\} . \tag{5}$$

Anisotropy. We can extend discrete Lloyd relaxation for anisotropic sampling as follows. First, we use our geodesic distance approximation (Equation 1) instead of the Euclidean one whenever measuring distance between two points (e.g., during the Voronoi step as in Equation 4). Second, following the discussion in [Feng et al. 2008], we replace density ρ by Jacobian $J^{\mathrm{T}} J$ when computing the centroid:

$$m_i = \left(\int_{V_i} J^{\mathrm{T}} J(p) dp\right)^{-1} \int_{V_i} J^{\mathrm{T}} J(p) p dp \, . \tag{6}$$

Note that since $J^{\mathrm{T}}J$ is positive definite, the first term is invertible. Third, under the discrete formulation above, we generate the point set P via our anisotropic white noise method, and pick the initial site locations S by uniform random sampling from P.

Capacity constraint. Lloyd relaxation has the potential quality issue of settling into a semi-regular sample distribution, causing bias in the power spectrum. This problem has recently been addressed by a technique termed capacity-constrained Voronoi tessellation (CCVT) [Balzer et al. 2009]. 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 the anisotropic extension for Lloyd relaxation, they can be naturally combined. Furthermore, since CCVT could be computationally slow, we also incorporate the accelerations in [Li et al. 2010].

4 Analysis

One of the standard methods to evaluate sample distribution quality is Fourier power spectrum (see, e.g., [Lagae and Dutré 2008]). Specifically, given a set of N samples $\{\mathbf{p}^k\}_{k=0 \text{ to } N-1}$, its Fourier spectrum $F(\mathbf{f})$ can be computed as follows:

$$F(\mathbf{f}) = \frac{1}{N}\sum_{k=0}^{N-1} e^{-2\pi i(\mathbf{f}.\mathbf{p}^k)} \qquad (7)$$

where $\mathbf{f}$ is the frequency. For the purpose of analyzing sample patterns, we are usually interested in the power spectrum/periodogram $P(\mathbf{f})$, which is essentially $|F(\mathbf{f})|^2$, as well as its radial mean and radial variance/anisotropy.

Unfortunately, the above formulation is suitable only to uniform isotropic sampling, as its direct application over an adaptive and/or anisotropic sample set might not produce very informative results. Specifically, for anisotropic sample sets the power spectrum will exhibit anisotropy as well, making it difficult to evaluate the true radial anisotropy as in traditional power spectrum for uniform isotropic sampling (see, e.g. Figure 13 in [Feng et al. 2008]). Consequently, prior publications often completely skip rigorous evaluations or rely on alternative metrics, such as the density and spatial uniformity.

In order to evaluate and compare different anisotropic sampling algorithms via Fourier spectrum analysis just like in traditional isotropic uniform sampling, we propose two alternative methods: (1) uniform-isotropic reversible warping and (2) spherical harmonics transform on uniform isotropic spherical sampling. Both methods evaluate anisotropic samples by Fourier analysis in an alternative isotropic domain, either on a plane (metric 1) or on a sphere (metric 2). Thus, metric 1 facilitates the comparison of our technique against sampling methods on planes [Feng et al. 2008] while metric 2 on manifold surfaces [Turk 1992; Alliez et al. 2002].

4.1 Uniform-isotropic reversible warping

Equation 7 can be modified for evaluating adaptive and/or anisotropic samples over a plane by utilizing planar warpings [Wolberg 1994] that are reversible to a uniform isotropic domain. Specifically, we draw samples from a warped domain that is anisotropic and/or adaptive, and (un)warp the generated samples to the uniform isotropic domain so that we can directly apply traditional spectrum analysis. Intuitively, drawing adaptive/anisotropic samples in the warped domain has an equivalent mirrored process in the uniform/isotropic domain for which distribution quality can be analyzed rigorously. This basic concept is simple, and we have found it very useful for analyzing spectrum property of anisotropic samples. Details are as follows.

Let φ be a warp from a uniform domain (denoted via spatial variable $\mathbf{p}$) to a deformed domain (denoted with $\mathbf{q}$):

$$\begin{aligned} \mathbf{q} &= \varphi(\mathbf{p}) \\ J_q(\mathbf{q}) &= I \\ J_q(\mathbf{p}) &= J_q(\varphi^{-1}(\mathbf{q}))\ . \end{aligned} \qquad (8)$$

For a set of samples $\{\mathbf{q}^k\}$ generated from Jacobian field $J_q(\mathbf{p})$, we can evaluate their power spectrum in the uniform domain by replacing $\mathbf{p}^k$ with $\varphi^{-1}(\mathbf{q}^k)$ in Equation 7:

$$F(\mathbf{f}) = \frac{1}{N}\sum_{k=0}^{N-1} e^{-2\pi i\left(\mathbf{f}.\varphi^{-1}(\mathbf{q}^k)\right)}\ . \qquad (9)$$

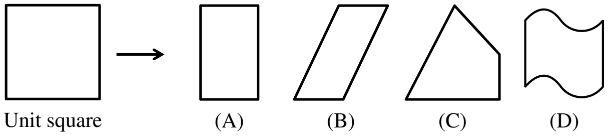


Figure 3: Four basic warps from [Wolberg 1994]. (A) a scaling projection, (B) a shear projection, (C) a perspective projection, and (D) a terrain relief.

Even though not all Jacobian fields have a reversible warp back to a uniform isotropic domain, we can still find a repertoire of warpings that are very useful for evaluation purposes.

Example. As a concrete example, let us consider a very simple scaling transform from $\mathbf{p} = (x, y)$ to $\mathbf{q} = (u, v)$ (see Figure 3 (A)).

$$\varphi : (u, v) = (x/2, y)$$

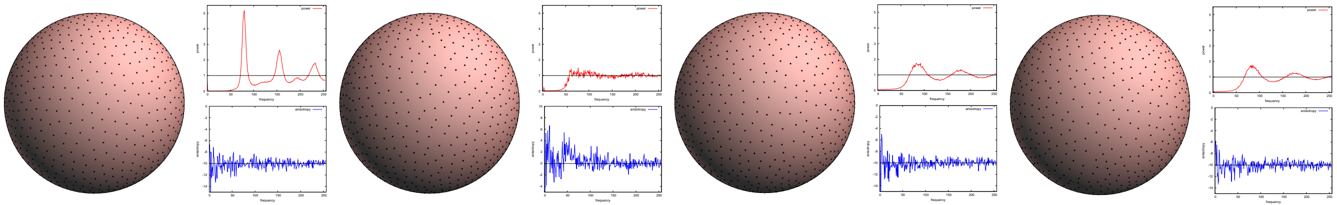
where we have

$$J_q(\varphi^{-1}(\mathbf{q})) = \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}\ .$$

Results. To verify our adaptive and/or anisotropic sampling method and compare it with previous works, we apply the reversible warping verification to results of ours and [Feng et al. 2008]. We choose several different warps as shown in Figure 3 that have representative characteristics: scaling, shearing, perspective projection, and a more complex aggregate warp. The first 3 warps are chosen because they represent basic warps with distinctive characteristics from which more complex warps can be aggregated from [Wolberg 1994], while the last warp has more complex aggregate properties. We then derive Jacobian fields from these warps and generate samples accordingly (Figure 15). We verify the sample quality by mapping the generated samples back to the uniform domain, and computing the Fourier power spectrum and the corresponding radial mean/variance analysis. As indicated in Figure 15, both our anisotropic dart throwing and relaxation methods produce more stable results with good spatial and spectral quality. Meanwhile, as indicated in Figure 4, the results by [Feng et al. 2008] may exhibit bias due to the use of jittered grid in their initialization stage as well as regular grid in their relaxation stage.

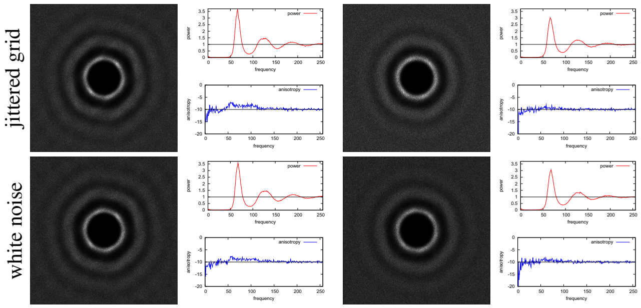
4.2 Sphere surface sampling

The warping method proposed in the previous subsection allows us to perform Fourier analysis of anisotropic samples in a planar uniform isotropic domain. Here, we follow a similar philosophy, but via a slightly different uniform isotropic domain: a 2-sphere (embedded in a 3D Euclidean space). We achieve this by a planar parameterization of the sphere, which by nature is anisotropic (and thus analogous to warping). We perform planar anisotropic



[Turk 1992] [Alliez et al. 2002] ground truth Our method

Figure 5: Sphere surface sampling and spectral analysis via spherical harmonics. The radial mean and anisotropy are obtained by averaging SH coefficients of 10 runs with around 1800 samples each, except [Alliez et al. 2002], which is a deterministic algorithm so only 1 run is used. All results are generated with the same sphere mesh data from [Praun and Hoppe 2003]. For [Turk 1992], 40 times of relaxation is used and for [Alliez et al. 2002], the resolution of area-distortion map is 512 × 512. As shown, our sampling quality is much closer to the ground truth produced by direct dart throwing on the sphere. In comparison, [Turk 1992] has a less smooth radial mean profile with excessive near-DC energy, and [Alliez et al. 2002] has high anisotropy.



square grid space discretization adaptive space discretization

Figure 4: Spectral analysis of [Feng et al. 2008] with different combinations of initial sample distribution (jittered grid vs. white noise) and sample space discretization for relaxation (squared grid vs. adaptive Jacobian). We use warp (B) in Figure 3 for clearest effects and use the same number of relaxation iterations. The original algorithm in [Feng et al. 2008] uses jittered grid + square grid discretization (i.e., the upper-left case). As shown, jittered grid initialization will produce more biased results than white noise; this is not surprising, as similar effects can be observed for vanilla isotropic dart throwing. Furthermore, square grid discretization also tends to produce worse results than an adaptive discretization following the underlying Jacobian field. This is because a square grid is not a uniform sampling in an anisotropic domain, and thus may cause additional anisotropy.

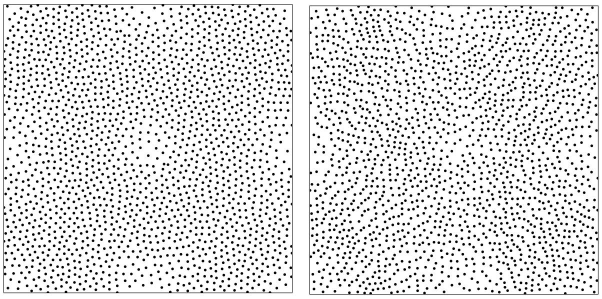
sampling on the parameterization domain and evaluate the uniform isotropic samples on the corresponding sphere surface.

However, here comes the question: since Fourier transform is for 2D planes, how are we going to apply Equation 7 to a sphere surface? Fortunately, there is a natural correspondence of planar Fourier transforms on spheres, which are spherical harmonics (SH) [Groemer 1996], the most commonly used tool for signal analysis on the 2-sphere [Roy 1976]. The basis functions of SH are the *associated Legendre polynomials*, defined over the range $[-1, 1]$, as shown in Equation 10:

$$Y_\ell^m(\theta,\varphi) = \begin{cases} \sqrt{(2\ell+1)\frac{(\ell-m)!}{(\ell+m)!}}\, P_\ell^m(\cos\theta)e^{im\varphi} & \text{if } m \geq 0 \\ (-1)^m Y_\ell^{*(-m)}(\theta,\varphi) & \text{if } m < 0 \end{cases} \tag{10}$$

where θ and φ are the spherical/polar coordinates; ℓ is the band index of the frequency (the larger the ℓ the higher the frequency); m, constrained in $[-\ell, \ell]$, can be regarded as the rotation variable within the same band and for different m on the same band, the polynomials are orthogonal; P_ℓ^m is the Legendre polynomials, defined as a recursive function on variables ℓ and m; and the superscript * denotes complex conjugation.

By replacing the basis function with the associated Legendre polynomials in Equation 7, we can obtain the frequency function of the sample distribution on the sphere surface:



[Alliez et al. 2002] Our method

Figure 6: The samples in parameter space. Here are the intermediate results of [Alliez et al. 2002] and our method for Figure 5.

$$\begin{aligned} SH(\ell,m) &= \frac{1}{N}\int_\Omega Y_\ell^m(\theta,\varphi)\sum_{k=0}^{N-1}\delta(\mathbf{s}-\mathbf{p}^k)d\mathbf{s} \\ &= \frac{1}{N}\sum_{k=0}^{N-1} Y_\ell^m(\mathbf{p}^k) \end{aligned} \tag{11}$$

where Ω is the sampling space (the 2-sphere) and δ the Dirac delta function. Different from Equation 7 where the frequency $\mathbf{f}$ are lattice integer points, Equation 11 has different numbers of ms for different ℓ, and thus does not result in a square-shaped power spectrum image. Fortunately, we are still able to examine the sampling quality through the radial mean and anisotropy on different frequency bands. Specifically, the radial mean for frequency ℓ is computed as the average of $\{|SH(\ell,m)|^2\}_{m=-\ell:\ell}$ and the radial anisotropy as their variance.

Results. Figures 5 and 6 compare our method with previous surface sampling algorithms [Turk 1992; Alliez et al. 2002]. As shown, using the ground truth produced by direct dart throwing over surfaces as reference, our approach outperforms alternative methods based on either relaxation [Turk 1992] or error diffusion [Alliez et al. 2002]. In particular, for [Turk 1992], we have found it difficult to choose the proper number of iterations to achieve the right balance between uniformity and randomness, as too few iterations might produce insufficient uniformity (the cause of the near-DC energy) whereas too many iterations might produce a highly regular (and thus biased) result. (In Figures 5, we have carefully chosen the number of iterations to produce the best result for [Turk 1992].) For [Alliez et al. 2002], as demonstrated in Figure 6, a more uniform sampling on the parameterization domain does not necessarily translate to a better sampling on the sphere surface sample domain.

4.3 Approximation

Here, we provide a more detailed analysis for the approximate distance metric d in Equation 1 as well as the consequent elliptical approximation of anisotropic samples. Borrowing our warping ver-

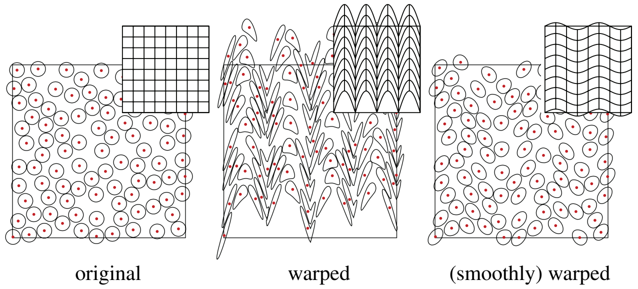


Figure 7: Why an anisotropic sample is not a little ellipse. The left is an original uniform isotropic field, while the middle is a warped field [Wolberg 1994] upon which sampling becomes anisotropic. However, when the distortion field varies smoothly enough, the warped shape can be well approximated by ellipse as shown on the right.

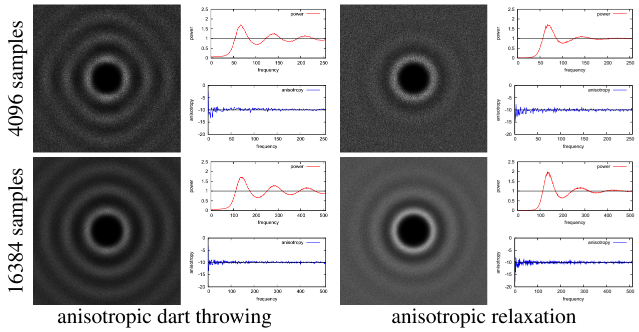


Figure 8: Quality of approximate geodesic distance metric. Here we apply our anisotropic dart throwing and anisotropic relaxation algorithms to the warping domain shown in the middle of Figure 7 and use reversible warping verification to examine the spectral quality of output samples. Note that when the sample number increases, the spectral quality of our anisotropic sampling improves, which is due to Equation 1 approaching the true geodesic distance.

ification method presented earlier, we can easily construct a scenario showing that anisotropic samples are not little ellipses. As shown in the left two images of Figure 7, suppose that we have an original uniform isotropic sampling domain that is warped to another domain. If we were to perform dart throwing on the warped domain *exactly*, we would have to treat the samples with shapes warped from the geometric disks from the original domain via strict geodesic distance computation. Since this warping can be arbitrary, so can the sample shapes in the output domain. Obviously, it is going to be computationally difficult to perform anisotropic sampling using these complex and spatially varying geometric constructs, as it would involve many geodesic distance computations.

Fortunately we have found the elliptical approximation to be more than adequate and even exact under special circumstances. For example, if we have a constant Jacobian field, then the amount of anisotropy will not vary and the samples will be shaped as identical ellipses, whose parameters are given by the singular values of the Jacobian. In general, if the Jacobian field is smooth or the sampling is sufficiently dense, then locally the size and orientation of our samples will not vary significantly, and the approximation in Equation 1 suffices. Refer to Figure 7 right, where this assumption holds, and Figure 7 middle, where this assumption is violated as well as Figure 8, where we show the quality of sampling approaching the ground truth when the sampling density increases.

4.4 Performance

We have measured the performance of our current implementation in Tables 1 and 2. We would like to emphasize that we currently implement our anisotropic dart throwing without any further performance optimizations. We envision its speed would go up significantly by borrowing ideas from previous acceleration techniques [Dunbar and Humphreys 2006; White et al. 2007; Wei 2008]. We leave this as a potential future work.

dimension \ $\|S\|$	512	1024	2048	4096	8192	16384
2D	0.07	0.43	1.86	4.67	20.6	81.4
3D	0.14	0.81	2.448	8.92	30.0	142.7
4D	0.15	0.83	2.65	13.6	41.7	187.3

Table 1: Performance of anisotropic dart throwing with different dimensions and number of samples $|S|$. All timings are in seconds, measured on an HP xw4400 workstation with an Intel Core(TM)2 CPU 6400 at 2.13GHz and 2GB Memory.

$\frac{\|P\|}{\|S\|}$ \ $\|S\|$	1024	4096	16384	32768
64	0.83	6.4	78.5	910.3
256	2.6	23.4	284.3	3980.1
1024	8.7	125.5	1911.0	N/A

Table 2: Performance of anisotropic relaxation with different number of samples $|S|$ and number of points per sample $\frac{|P|}{|S|}$. All timings (in seconds) are measured on the same workstation as in Table 1. (See Section 3.3 for the definition of S and P.)

5 Applications

Here, we present several applications of our technique, beyond plain vanilla sampling that we have shown earlier in Section 4. Depending on the particular application characteristics, either anisotropic dart throwing or anisotropic relaxation could be more suitable, as detailed in the individual applications.

5.1 Anisotropic stippling

Stippling is a technique to represent an image with small single-color primitives (e.g., dots or sticks), and isotropic blue noise sampling has been proved to be a potential method for reproducing the image intensity without noticeable graininess or structural artifacts [Secord 2002; Kopf et al. 2006; Ostromoukhov 2007; Balzer et al. 2009]. However, as shown in Figure 16, isotropic blue noise might not properly reproduce anisotropic image features. Our method, by explicitly considering anisotropy, can produce better results with a similar number of samples.

Since spatial uniformity is more important than strict spectrum quality for stippling applications, here we choose anisotropic relaxation over anisotropic dart throwing as the generation algorithm. Given an input gray-scale image I, we first compute the Jacobian field J by treating I as a surface M with an intrinsic parameterization, $M(x, y) = (x, y, I(x, y))$, with the intensity $I(x, y)$ representing the third coordinate. After that, a Jacobian field J can be easily computed as the parameterization Jacobian of the surface M to the 2D domain of I. J is further scaled by the distance field D derived from I (see [Wei 2008]) to reflect the local image density. We then feed J into our anisotropic relaxation algorithm for distributing the stipples. To depict anisotropy, we use sticks as the rendering primitive as they have shown to be capable of revealing image gradients and thus more faithfully reproduce local features than dots [Hiller et al. 2003; Fritzsche et al. 2005]. (We have not found it necessary to use more complex primitives like ellipses or curved strokes due to the high sample density of typical stippling applications.) The directions of the sticks are orthogonal to the principal axis of the local J and their lengths are inversely proportional to the local D.

Figure 16 shows anisotropic stippling performs better than alternative methods such as isotropic stippling [Balzer 2009] and structure-aware half-toning [Pang et al. 2008]. We would like to note that both anisotropic primitives and anisotropic distributions contribute to the effectiveness of our approach, as isotropic primitives over anisotropic distributions fall short in depicting anisotropic structures, while anisotropic primitives over isotropic distributions produce less uniform (and thus noisier) results (see our supplementary materials).

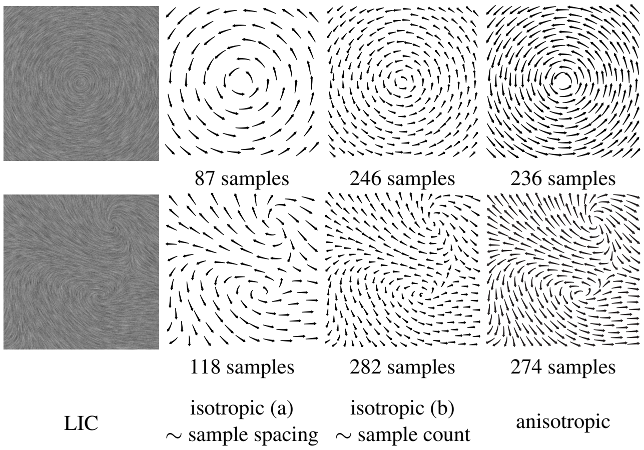


Figure 9: Vector field visualization via blue noise sampling. The input vector field (left-most column) is visualized via LIC [Cabral and Leedom 1993]. We produce two sets of isotropic sampling results, isotropic (a) (second column) with similar spacing parameter and another isotropic (b) (third column) with similar sample count to our method (right-most column). For the glyph rendering, we bend the glyph arrows to align with the underlying flow. Compared to isotropic blue noise sampling in (a, b), our method can pack the glyphs anisotropically, allowing more desirable inter-glyph spacing in directions both parallel and orthogonal to the underlying flows.

5.2 Vector field visualization

Another potential application for blue noise sampling is to place glyphs for vector field visualization. Compared to other types of vector field visualization, a uniform and random distribution of glyphs [Turk and Banks 1996] could maximize the visual quality (such as the description of critical points) while clearly presenting the direction of vector flows [Laidlaw et al. 2005]. Such uniform and random distributions are exactly in line with the property of blue noise sampling. Since glyphs often take elongated, anisotropic shapes such as long arrows, simply using isotropic blue noise sampling would not be effective because it would result in having too much/little spacing in the direction orthogonal/parallel to the flows, as shown in Figure 9.

Our anisotropic blue noise sampling method, in contrast, can naturally be applied to place such anisotropic glyphs, producing better visualization than isotropic blue noise sampling. For this application, we opt for our anisotropic dart throwing over relaxation to ensure that the glyphs never intersect each other. To apply our method, we derive the Jacobian from the input vector field so that the corresponding ellipse at each location will have its major axis aligned and scaled in proportional to the underlying flow vector, while having its minor axis lined up with the orthogonal direction according to a user specified constant width. Therefore, such an anisotropic packing can better balance the inter-glyph spacing in the directions parallel and orthogonal to the flows. Compared to prior vector field visualization algorithms such as [Turk and Banks 1996], the main advantage of our approach is its simplicity; in a sense, the random descent solver in [Turk and Banks 1996] already hints on an anisotropic blue noise distribution, but it could be potentially more complex and slower than our approach. However, we wish to point out that our method is suitable only for placing short segment glyphs; for longer and more global contours such as those shown in the latter half of [Turk and Banks 1996], a more specialized (and possibly more complex) algorithm might still be required to produce the best effects.

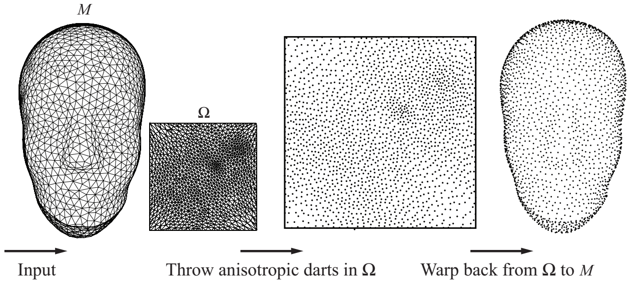


Figure 10: Surface sampling via anisotropic dart throwing.

5.3 Sampling manifold surfaces

Distributing samples uniformly on a manifold surface has long been a challenging and well studied problem, e.g., [Turk 1992; Fu and Zhou 2008]. However, most of these methods perform sampling directly on surfaces; this might require geodesics and other surface related data structures, which are usually nontrivial to compute and implement. Some other approaches draw samples on the 2D parameter domain and thus avoid implementing surface geodesics [Alliez et al. 2002], but as we have shown in Section 4.2, these approaches might suffer from quality issues. Our goal is to present a method that combines the benefits of both methodologies: on one hand, we would like to perform sampling directly on a 2D parameter domain in order to avoid the complex implementation of surface geodesics and data structures, and on the other hand, we would like our sampling results to exhibit excellent blue noise spectrums. We can achieve this goal by performing our anisotropic blue noise sampling algorithm directly in the 2D parameter space instead of the input surfaces. In essence, we avoid the direct computation of surface geodesics by converting the problem of *isotropic surface* sampling into *anisotropic planar* sampling.

For this application, we opt for anisotropic dart throwing over relaxation since the former can be more easily adapted for different situations, e.g., multi-chart parameterization. Surface sampling also does not seem to require exact sample count specification, and even though in theory relaxation will produce spatially more uniform samples than dart throwing, we have not found significant differences in the surface setting.

Figure 10 shows the procedure of our surface sampling approach. On the first step, given a parameterized surface M, we compute the Jacobian J of the parameterization $f : \Omega \to M$. A nice property of our algorithm is that it generalizes to any existing surface parameterization. For the specific case of triangle meshes, we can choose from one of several suitable existing methods. For surface sampling application the Jacobian has to take into account not only surface geometry but also the sample spacing requirement (see Section 2). To achieve this, we scale the J computed above (from parameterization only) to account for the desired minimum distance between two samples on the surface. The minimum distance is computed with the required number of samples N on M, with upper bound given by

$$r_{max} = 2\sqrt{\frac{S}{2\sqrt{3}N}} \tag{12}$$

where S is the area of the surface M [Li et al. 2008]. In practice, $r = \rho r_{max}$, where ρ is a scalar over $[0, 1]$ [Lagae and Dutré 2006a]. In our case, we let ρ be 0.75. We multiply J by $\frac{1}{\rho r_{max}}$ to obtain the input Jacobian field for our algorithm. Then, the algorithm is carried out to produce anisotropic samples in the 2D parameter

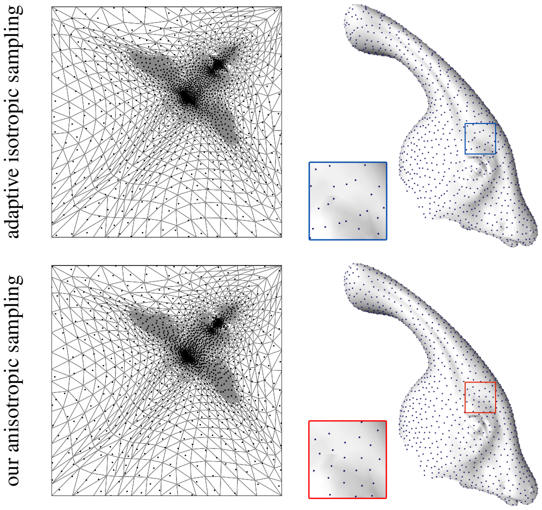


Figure 12: Comparison of adaptive isotropic sampling and our anisotropic sampling to generate isotropic samples over a mesh surface. The parameterization used is mostly effective at preserving angles [Desbrun et al. 2002], thus achieving a nearly isotropic mapping.

space, which are finally warped back to the 3D surface space using barycentric coordinates to form the uniform isotropic samples.

We present results for two meshes using the parameterization metric from [Desbrun et al. 2002] in Figure 11 . Note that both meshes are hard to parameterize with low stretch, yet our algorithm still produces satisfactory results. Figure 12 demonstrates that, even when using such a parameterization that gives importance to angle preservation, the use of anisotropic sampling produces superior results over adaptive isotropic sampling.

By altering the input Jacobian field to our algorithm, we can also perform other kinds of samplings. For example, we can multiply it with an importance field (e.g., mean curvatures) in order to perform *adaptive isotropic* surface sampling, as shown in Figure 13. We could also produce an *anisotropic surface* sampling by generating a Jacobian that combines the parameterization Jacobian with a user specified tensor field defined on the surface. Such capability allows us to distribute *anisotropic* texture patterns on surfaces, whereas prior methods (see [Lagae and Dutré 2005]) use isotropic blue noise and thus are applicable only for isotropic texture patterns. As shown in Figure 14, our method is able to produce uniform placement for both isotropic and anisotropic texture patterns, whereas prior isotropic methods might not work well for the latter. None of these applications require any changes to our anisotropic sampling machinery.

6 Limitations and Future Work

We currently place anisotropic samples subject to a given Jacobian field without additional constraints, such as mesh features [Vorsatz et al. 2001] and image contours [Kim et al. 2008]. One potential future work is to impose additional constraints to our current algorithms, so that the sample placements satisfy not only anisotropic blue noise properties but also specific application needs.

Even though we have shown that ellipses can be a reasonable approximation to true anisotropic samples under many circumstances, we have also shown that this is really only an approximation. One interesting future work is to investigate more advanced metaphors to perform a more accurate approximation, as well as strategies to gracefully fall back to such more accurate measures when our approximation is detected to be inadequate.

The main purpose of our proposed analysis methods is to compare against different anisotropic sampling methods, and we have shown that both warping and sphere surface analysis are capable of detecting issues in prior methods. However, we would like to point out that our analysis methods are for evaluating anisotropic sampling methods, not the generated anisotropic sample distributions. One potential future work is to devise a more general methodology to perform Fourier spectrum analysis, e.g., for sample sets distributed over general surfaces [Bowers et al. 2010].

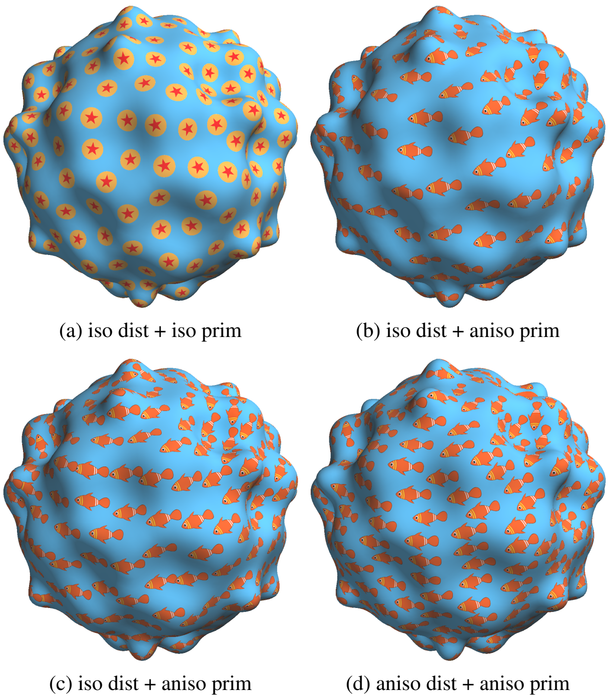

(a) iso dist + iso prim
(b) iso dist + aniso prim
(c) iso dist + aniso prim
(d) aniso dist + aniso prim

Figure 14: Texture distribution on surfaces. Here we show a variety of combinations of isotropic/anisotropic distributions with isotropic/anisotropic texture primitives. Case (a) is the traditional iso/iso combination. Case (d) is our aniso/aniso result. Both case (b) and (c) have a iso/aniso combination, with (b) having the same spacing parameter while (c) the same sample count to (d). As shown, anisotropic distribution (d) provides better quality than isotropic distribution (b) (c) for anisotropic primitives. (Notice that both (b) (c) are either too sparse or too dense in certain directions of the distributions.) Cases (a) (c) (d) have 500 samples while case (b) has 350 samples.

Acknowledgements We would like to thank Eugene Zhang for sharing his code on surface tensor field design, Yingge Qu + other authors of [Pang et al. 2008] for providing the van Gogh image + their structure-aware half-toning results, and the anonymous reviewers for their comments and suggestions. Hongwei Li and Pedro V. Sander were partly supported by Hong Kong RGC grants 619008 and 619509.

References

Alliez, P., Meyer, M., and Desbrun, M. 2002. Interactive geometry remeshing. In *SIGGRAPH '02*, 347–354.

Alliez, P., Cohen-Steiner, D., Devillers, O., Lévy, B., and Desbrun, M. 2003. Anisotropic polygonal remeshing. In *SIGGRAPH '03*, 485–493.

Balzer, M., Schlomer, T., and Deussen, O. 2009. Capacity-constrained point distributions: A variant of Lloyd's method. In *SIGGRAPH '09*, 86:1–8.

Balzer, M. 2009. Capacity-constrained Voronoi diagrams in

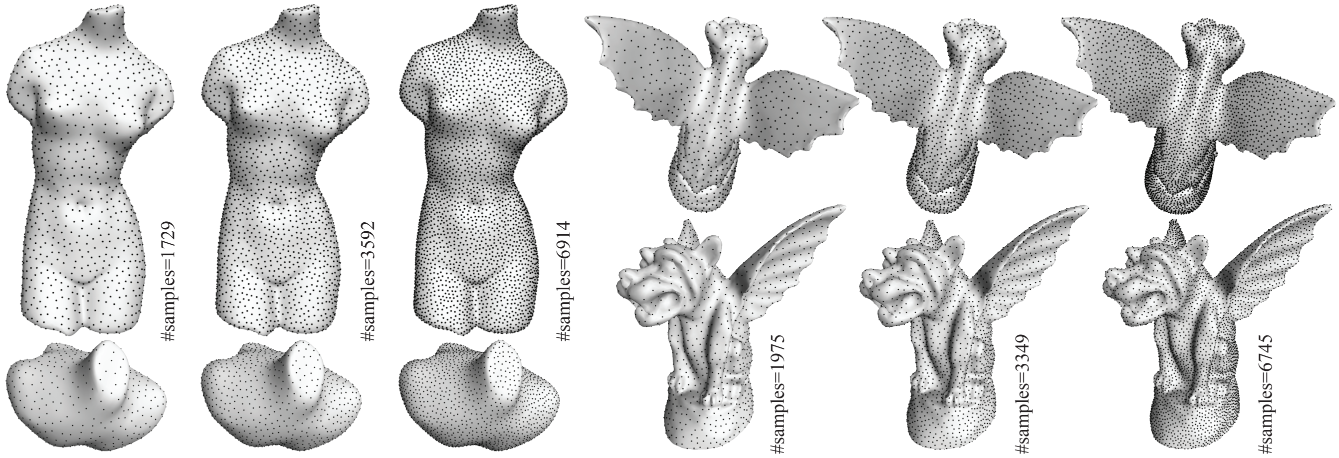


Figure 11: Uniformly sampling a manifold surface with anisotropic Poisson disk sampling. We show two models, Venus and Gargoyle, at different sampling densities. Both meshes are genus-0 models and parameterized with the same method [Desbrun et al. 2002]. We highlight the neck of Venus and wings of Gargoyle as these two regions are the most stretched areas in the parameter domain.

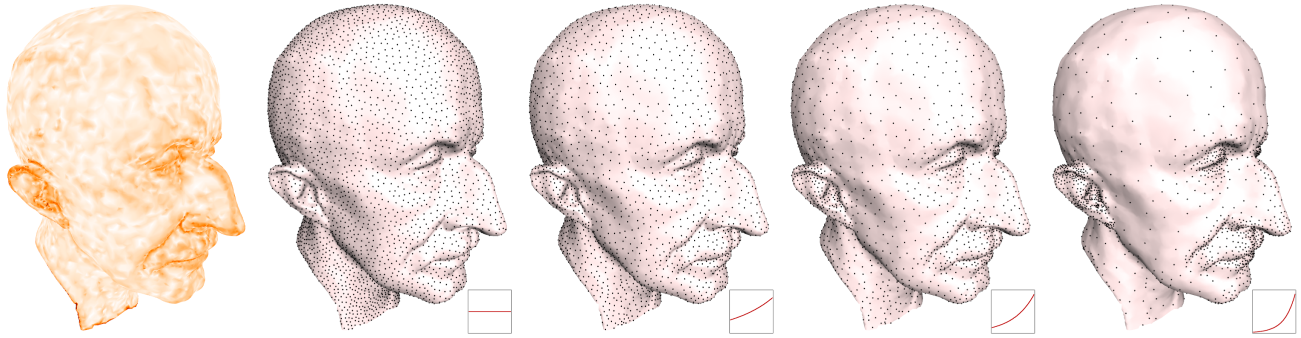


Figure 13: Adaptive sampling MaxPlanck model based on mean curvature. The method of adaptive sampling is similar to uniformly sampling within our anisotropic dart throwing framework. The only modification is that the input Jacobian is scaled with a function of the mean curvature of surface. Here we present the mean curvature visualization and the resulting samples with different 1-D scaling functions of the curvature (small plots at the lower right of each figure).

continuous spaces. In *International Symposium on Voronoi Diagrams in Science and Engineering*, 79–88.

Bowers, J., Wang, R., Wei, L.-Y., and Maletz, D. 2010. Parallel Poisson disk sampling with spectrum analysis on surfaces. In *SIGGRAPH Asia '10*, to appear.

Cabral, B., and Leedom, L. C. 1993. Imaging vector fields using line integral convolution. In *SIGGRAPH '93*, 263–270.

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 Transactions on Graphics 5*, 1, 51–72.

Desbrun, M., Meyer, M., and Alliez, P. 2002. Intrinsic parameterizations of surface meshes. *Computer Graphics Forum 21*, 3, 209–218.

Dunbar, D., and Humphreys, G. 2006. A spatial data structure for fast Poisson-disk sample generation. In *SIGGRAPH '06*, 503–508.

Feng, L., Hotz, I., Hamann, B., and Joy, K. 2008. Anisotropic noise samples. *IEEE Transactions on Visualization and Computer Graphics 14*, 2, 342–354.

Fritzsche, L.-P., Hellwig, H., Hiller, S., and Deussen, O. 2005. Interactive design of authentic looking mosaics using Voronoi structures. In *In Proc. 2nd International Symposium on Voronoi Diagrams in Science and Engineering*, 1–11.

Fu, Y., and Zhou, B. 2008. Direct sampling on surfaces for high quality remeshing. In *SPM '08: Proceedings of the 2008 ACM symposium on Solid and physical modeling*, 115–124.

Groemer, H. 1996. *Geometric Applications of Fourier Series and Spherical Harmonics*. Cambridge University Press.

Hiller, S., Hellwig, H., and Deussen, O. 2003. Beyond stippling – methods for distributing objects on the plane. In *Proceedings of Eurographics 2003*, vol. 22, 515–522.

Kim, D., Son, M., Lee, Y., Kang, H., and Lee, S. 2008. Feature-guided image stippling. *Computer Graphics Forum 27*, 4, 1209–1216.

Kopf, J., Cohen-Or, D., Deussen, O., and Lischinski, D. 2006. Recursive Wang tiles for real-time blue noise. In *SIGGRAPH '06*, 509–518.

Labelle, F., and Shewchuk, J. R. 2003. Anisotropic Voronoi diagrams and guaranteed-quality anisotropic mesh generation. In *SCG '03: Proceedings of the nineteenth annual symposium on Computational geometry*, 191–200.

Lagae, A., and Dutré, P. 2005. A procedural object distribution function. *ACM Transactions on Graphics 24*, 4, 1442–1461.

Lagae, A., and Dutré, P. 2006. An alternative for Wang tiles: colored edges versus colored corners. *ACM Transactions on Graphics 25*, 4, 1442–1459.

Lagae, A., and Dutré, P. 2006. Poisson sphere distributions. In *Vision, Modeling, and Visualization 2006*, IEEE Computer Society, 373–379.

Lagae, A., and Dutré, P. 2008. A comparison of methods for generating Poisson disk distributions. *Computer Graphics Forum 27*, 1, 114–129.

Laidlaw, D. H., Kirby, R. M., Jackson, C. D., Davidson, J. S., Miller, T. S., da Silva, M., Warren, W. H., and Tarr, M. J. 2005. Comparing 2D vector field visualization methods: A user study. *IEEE Transactions on Visualization and Computer Graphics 11*, 1, 59–70.

Lévy, B., and Liu, Y. 2010. Lp Centroidal Voronoi Tessellation and its applications. In *SIGGRAPH '10*, 119:1–11.

Li, H., Lo, K.-Y., Leung, M.-K., and Fu, C.-W. 2008. Dual Poisson-disk tiling: An efficient method for distributing features on arbitrary surfaces. *IEEE Transactions on Visualization and Computer Graphics 14*, 5, 982–998.

Li, H., Nehab, D., Wei, L.-Y., Sander, P. V., and Fu, C.-W. 2010. Fast capacity constrained Voronoi tessellation. In *I3D '10: SIGGRAPH symposium on Interactive 3D Graphics and Games Posters*, 13:1–1.

Lloyd, S. 1983. An optimization approach to relaxation labeling algorithms. *Image and Vision Computing 1*, 2, 85–91.

McCool, M., and Fiume, E. 1992. Hierarchical Poisson disk sampling distributions. In *Proceedings of the conference on Graphics interface '92*, 94–105.

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

Ostromoukhov, V. 2007. Sampling with Polyominoes. In *SIGGRAPH '07*, 78:1–6.

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.

Praun, E., and Hoppe, H. 2003. Spherical parametrization and remeshing. In *SIGGRAPH '03*, 340–349.

Roy, R. 1976. Spectral analysis for a random process on the sphere. *Annals of the Institute of Statistical Mathematics 28*, 1 (December), 91–97.

Secord, A. 2002. Weighted Voronoi stippling. In *NPAR '02: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, 37–43.

Sethian, J. A. 1996. A fast marching level set method for monotonically advancing fronts. In *Proc. Nat. Acad. Sci*, 1591–1595.

Turk, G., and Banks, D. 1996. Image-guided streamline placement. In *SIGGRAPH '96*, 453–460.

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

Vorsatz, J., Rössl, C., Kobbelt, L., and Seidel, H.-P. 2001. Feature sensitive remeshing. In *Computer Graphics Forum, Proceedings of Eurographics 2001*, vol. 20, 393–401.

Weber, O., Devir, Y. S., Bronstein, A. M., Bronstein, M. M., and Kimmel, R. 2008. Parallel algorithms for approximation of distance maps on parametric surfaces. *ACM Trans. Graph. 27*, 4, 1–16.

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.

White, K., Cline, D., and Egbert, P. 2007. Poisson disk point sets by hierarchical dart throwing. In *Symposium on Interactive Ray Tracing*, 129–132.

Wolberg, G. 1994. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, CA, USA.

Yellott, J. I. J. 1983. Spectral consequences of photoreceptor sampling in the rhesus retina. *Science 221*, 382–385.

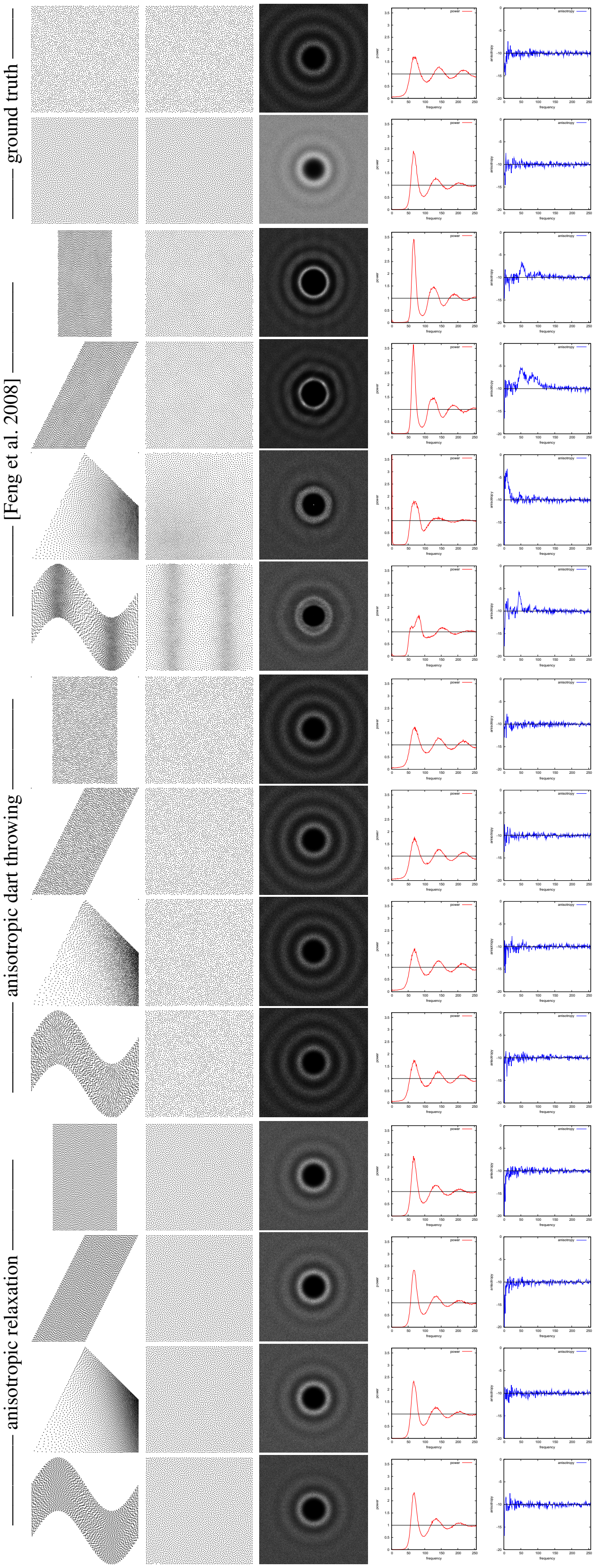
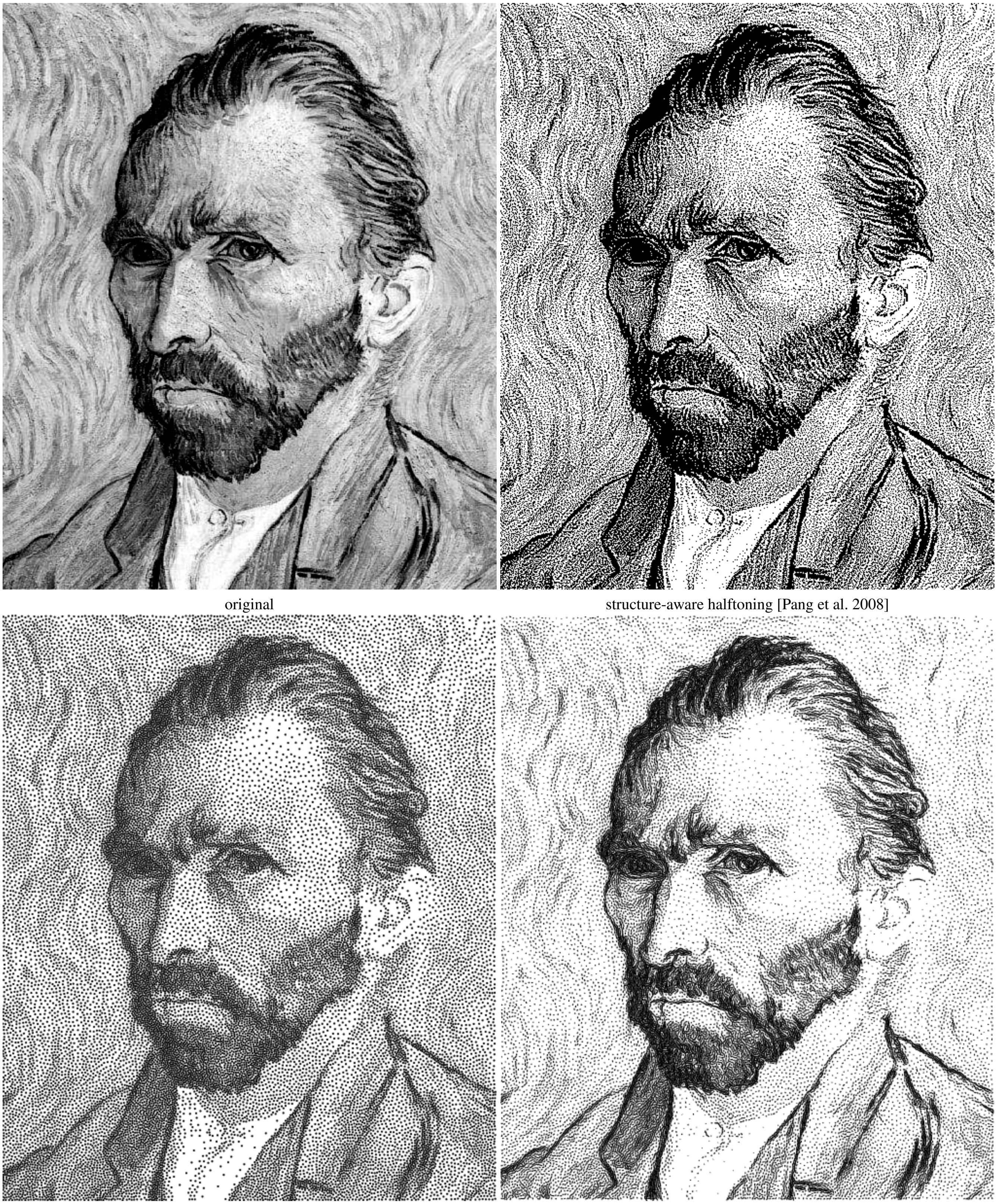


Figure 15: Verification via warping. Except for the ground truth produced by dart throwing (1st row) and relaxation [Balzer et al. 2009] (2nd row), each method (marked vertically on the left) is associated with 4 different warpings as in Figure 3. From left to right: original/anisotropic samples, warped/isotropic samples, power spectrum averaged over 10 runs, and the corresponding radial mean and anisotropy plots. Each case contains ∼3800 samples. Notice the obvious anisotropic bias exhibited by [Feng et al. 2008].



original

structure-aware halftoning [Pang et al. 2008]

isotropic stippling [Balzer et al. 2009]

anisotropic stippling [our method]

Figure 16: Anisotropic stippling. Here we compare results generated by structure-aware halftoning [Pang et al. 2008], isotropic stippling [Balzer et al. 2009], and our anisotropic method. The top 2 bitmap images have resolution 508 × 603, while the bottom 2 stippling results contain 30,000 samples. Compared to (discrete) halftoning, (continuous) stippling provide a more natural and organic effect. Compared to isotropic stippling, our method depicts anisotropic features better, such as the hair, the strokes, and the region contours. Note: to minimize the quality differences caused by the bitmap and vector graphics renderers, we have rasterized the two stippling images into PNG format with the same resolution of the two bitmap images. The vector graphics versions of stippling images are shown in the supplementary materials.

A Supplementary Materials

A.1 Jacobian Computation

Here, we provide more details on how to compute Jacobians for various applications.

Math formulas for the 4 warps in Figure 3 :

Scale:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{13}$$

Shear:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{14}$$

Perspective:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} 3 & 1 & 0 \\ 0 & 2 & 0 \\ 2 & 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{15}$$

Terrain:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} x \\ \frac{1}{4}\left(1 + 2y + sin(2\pi x)\right) \\ 1 \end{pmatrix} \tag{16}$$

Deriving Jacobian from warp Let (x, y, z) denote the uniform space and (u, v, w) the anisotropic space in homogeneous coordinates. Then in general we have

$$J = \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{pmatrix} \tag{17}$$

If the warp is linear, i.e.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & k \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{18}$$

it can be shown that the corresponding Jacobian is

$$J = \frac{1}{z} \begin{pmatrix} a & b \\ c & d \end{pmatrix} - \frac{1}{z^2} \begin{pmatrix} x \\ y \end{pmatrix} \begin{pmatrix} g & h \end{pmatrix} \tag{19}$$

Deriving Jacobian from image Given an importance field I (e.g. a bitmap image as in Figure 16), we can treat it as a height field (x, y, z):

$$\begin{aligned} x &= u \\ y &= v \\ z &= I(u, v) \end{aligned} \tag{20}$$

We can compute its Jacobian as follows:

$$J = \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial v} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \frac{\partial I}{\partial u} & \frac{\partial I}{\partial v} \end{pmatrix} \tag{21}$$

where $\frac{\partial I}{\partial u}$ and $\frac{\partial I}{\partial v}$ are image gradients (computed by e.g. Sobel filter).

We then have

$$J^{\mathrm{T}} J = \begin{pmatrix} 1 + \left(\frac{\partial I}{\partial u}\right)^2 & \frac{\partial I}{\partial u}\frac{\partial I}{\partial v} \\ \frac{\partial I}{\partial u}\frac{\partial I}{\partial v} & 1 + \left(\frac{\partial I}{\partial v}\right)^2 \end{pmatrix} \tag{22}$$

from which we will compute the eigenvalues. (Notice that $J^{\mathrm{T}} J$ becomes an identity matrix for zero gradient, reducing our anisotropic stippling into isotropic stippling.)

To take into account importance $I(.)$, we will have to derive the corresponding $r(.)$ field from $I(.)$ (see e.g. [Wei 2008]) and multiply $r(.)/r(.)^{-1}$ for J^{-1}/J.

A.2 3D object distribution

The applications we have presented so far could be considered as placing 2D objects. Here, we present a potential application for placing 3D objects, i.e., a planetary ring consisting of rocks and small asteroids around an imaginary planet. Due to gravity, a blue noise distribution is more suitable than a white noise distribution, where the objects could collide and smash, or an isotropic Poisson sphere distribution [Lagae and Dutré 2006b], where the objects have to be of the same shape, size and mass. We place the objects by first creating a 3D Jacobian field and then feeding it into the anisotropic dart throwing algorithm. This is accomplished by first discretizing the field into a 3D grid, then setting a user-specified number of random Js into random positions in the grid array, and finally filling the remaining empty cells by linear interpolation. Once we obtain the Jacobian field, we can place the objects in 3D space and produce the final renderings shown in Figure 17. In this example, we use a random J distribution, but our method could be applied to other scenarios by taking a different initial Jacobian field.

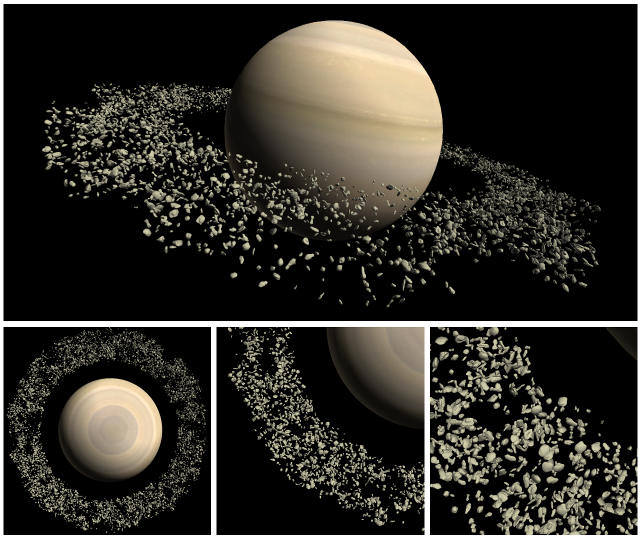


Figure 17: Asteroids in the orbit of an imaginary planet.

A.3 Conflict metrics

As stated in Section 3.2, we chose the $\max$ rather than the mean conflict metric for its efficiency and extensibility to higher dimensions. In theory, since the mean metric corresponds to true geometric shapes, it ought to provide better quality. Even though this is true in local spatial regions where the Jacobian changes significantly (and thus causing large sample ellipses to be adjacent to small ones), we have found the overall spectrum quality of these two methods quite similar (Figure 19). Thus, due to its superior

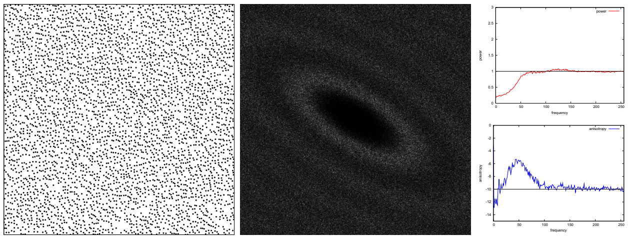


Figure 18: Power spectrum on anisotropic samples. Direct application of traditional spectral analysis on an anisotropic sample set results in anisotropic power spectrum, making it hard to judge if the anisotropy is really expected or as a consequence of some undesirable bias.

quality/performance tradeoff, we opt for max instead of mean as the conflict metric for anisotropic dart throwing.

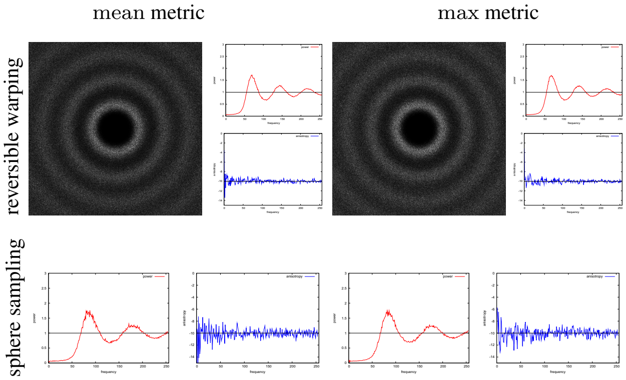


Figure 19: Comparison between mean and max conflict metrics. We use our two evaluation methods, reversible warping (top row) and sphere sampling (bottom row) to compare the qualities of distributions generated by the mean and max metrics. For reversible warping, we use case (D) in Figure 15. Other parameters used in the evaluation are the same as in Figure 15 and 5. Note the similarity between results produced by the two metrics.

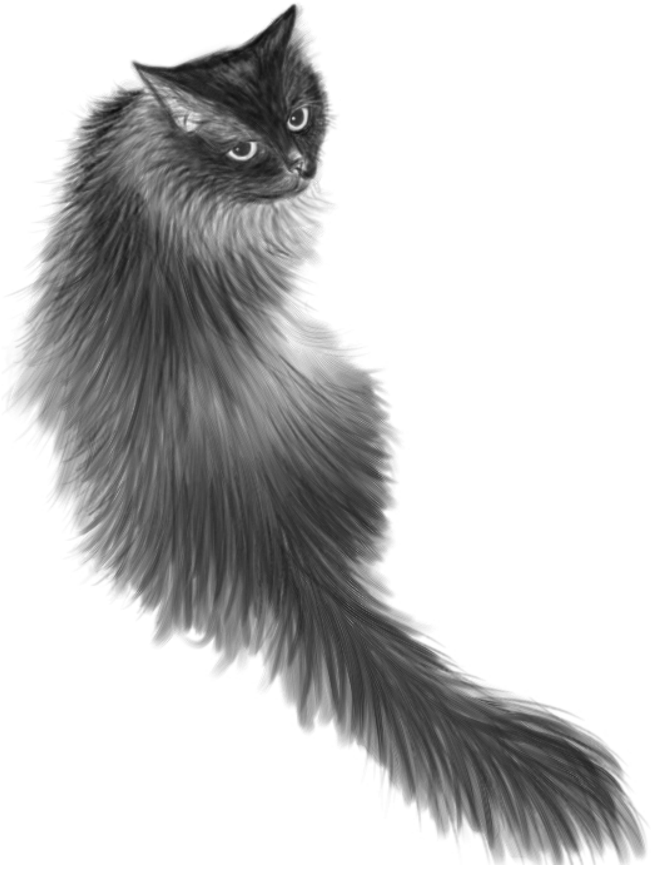
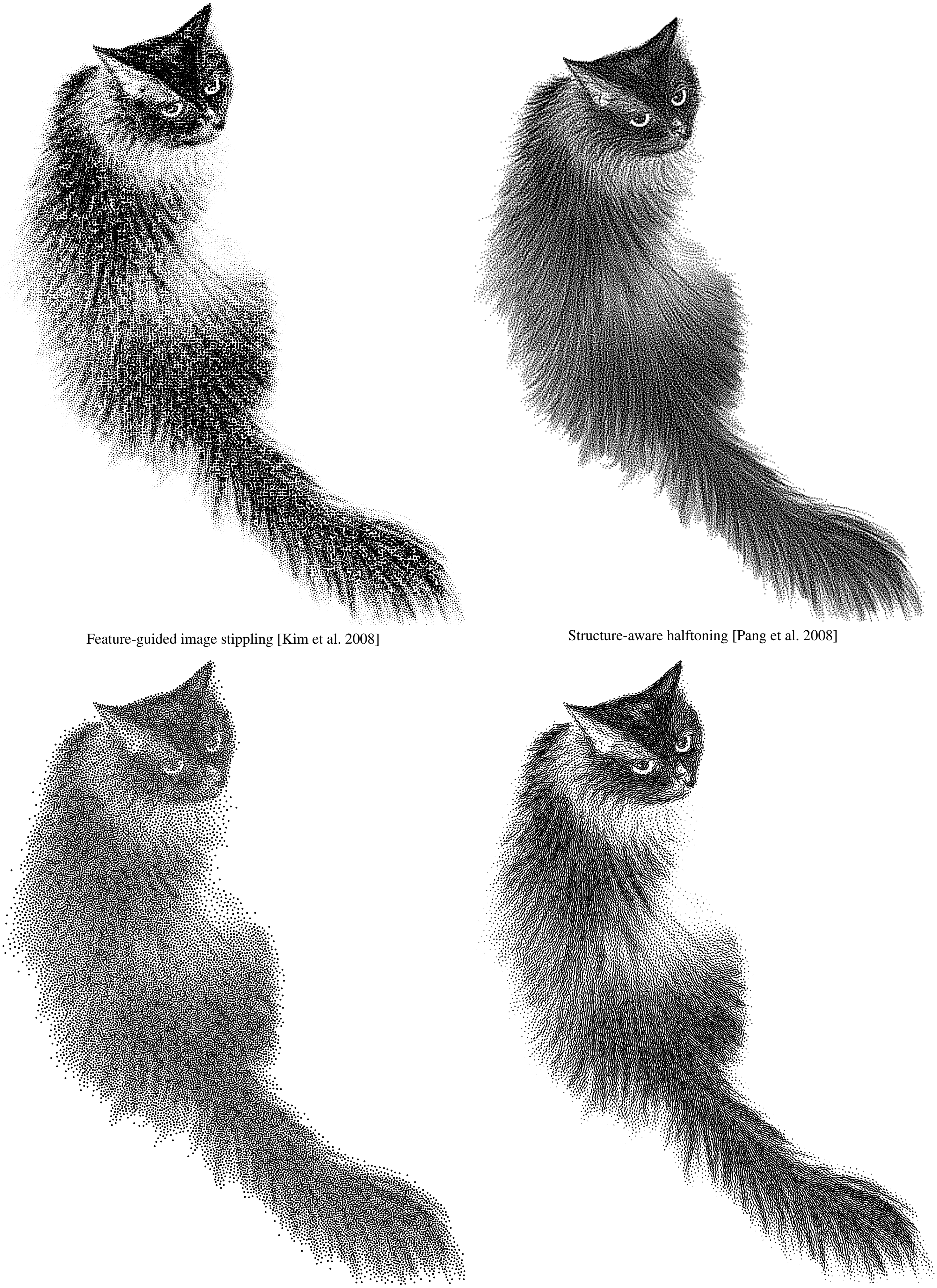


Figure 20: The original cat image. Courtesy of mMark (`http://mMark.deviantart.com/`).



Feature-guided image stippling [Kim et al. 2008]

Structure-aware halftoning [Pang et al. 2008]

Isotropic stippling [Balzer et al. 2009]

Anisotropic stippling [our method]

Figure 21: More anisotropic stippling comparisons. The original image is shown in Figure 20. Each stippling result contains ∼20,000 samples.

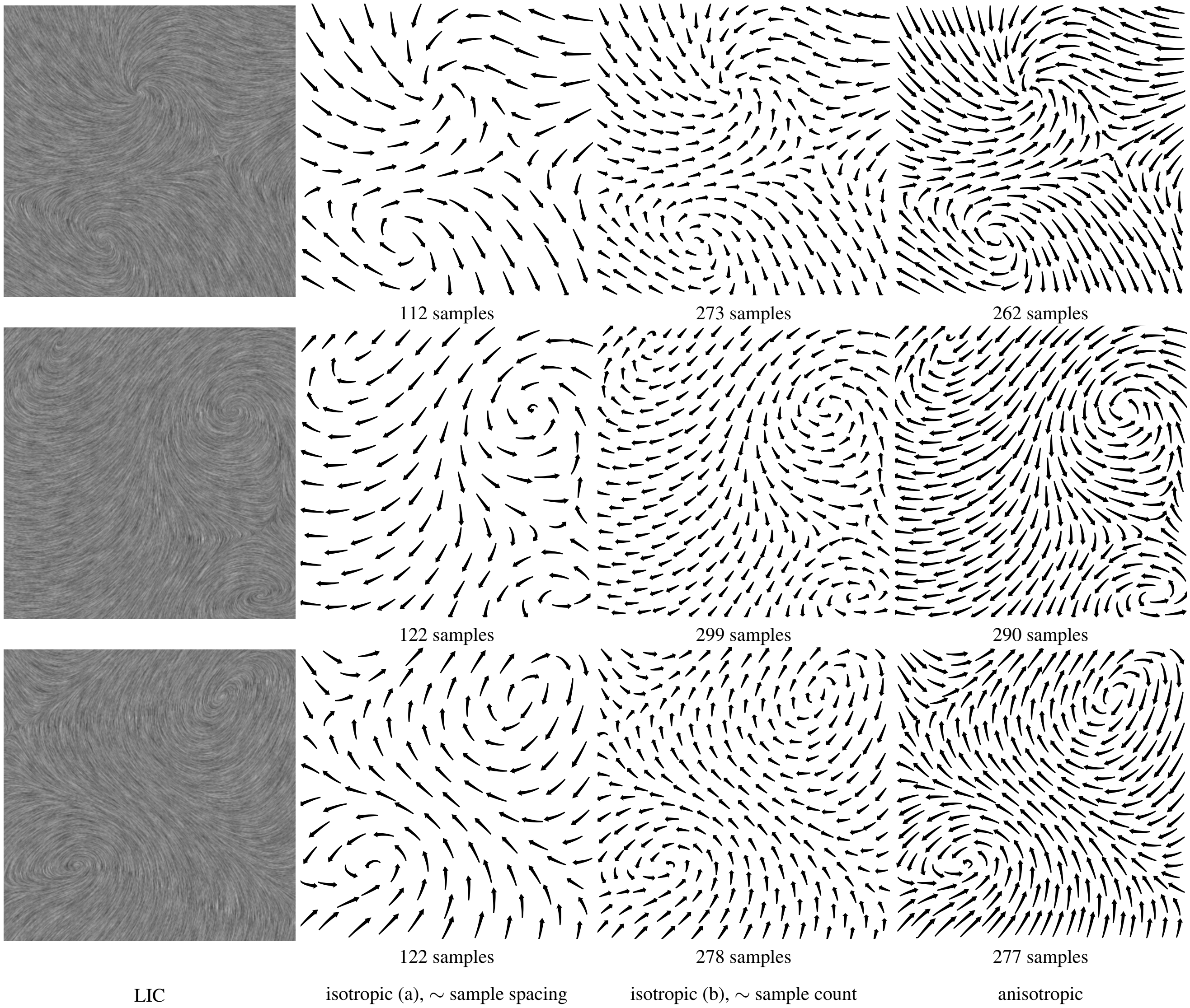


Figure 22: Vector field visualization via blue noise sampling. Here are more results for Figure 9.

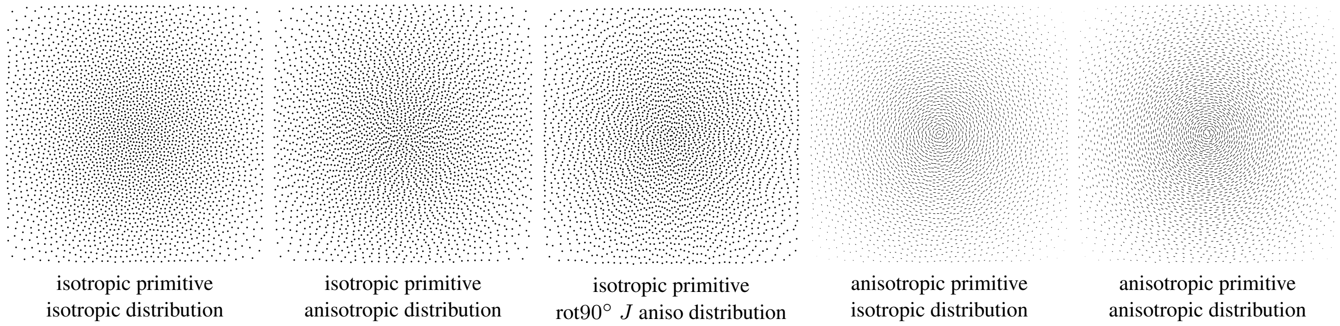
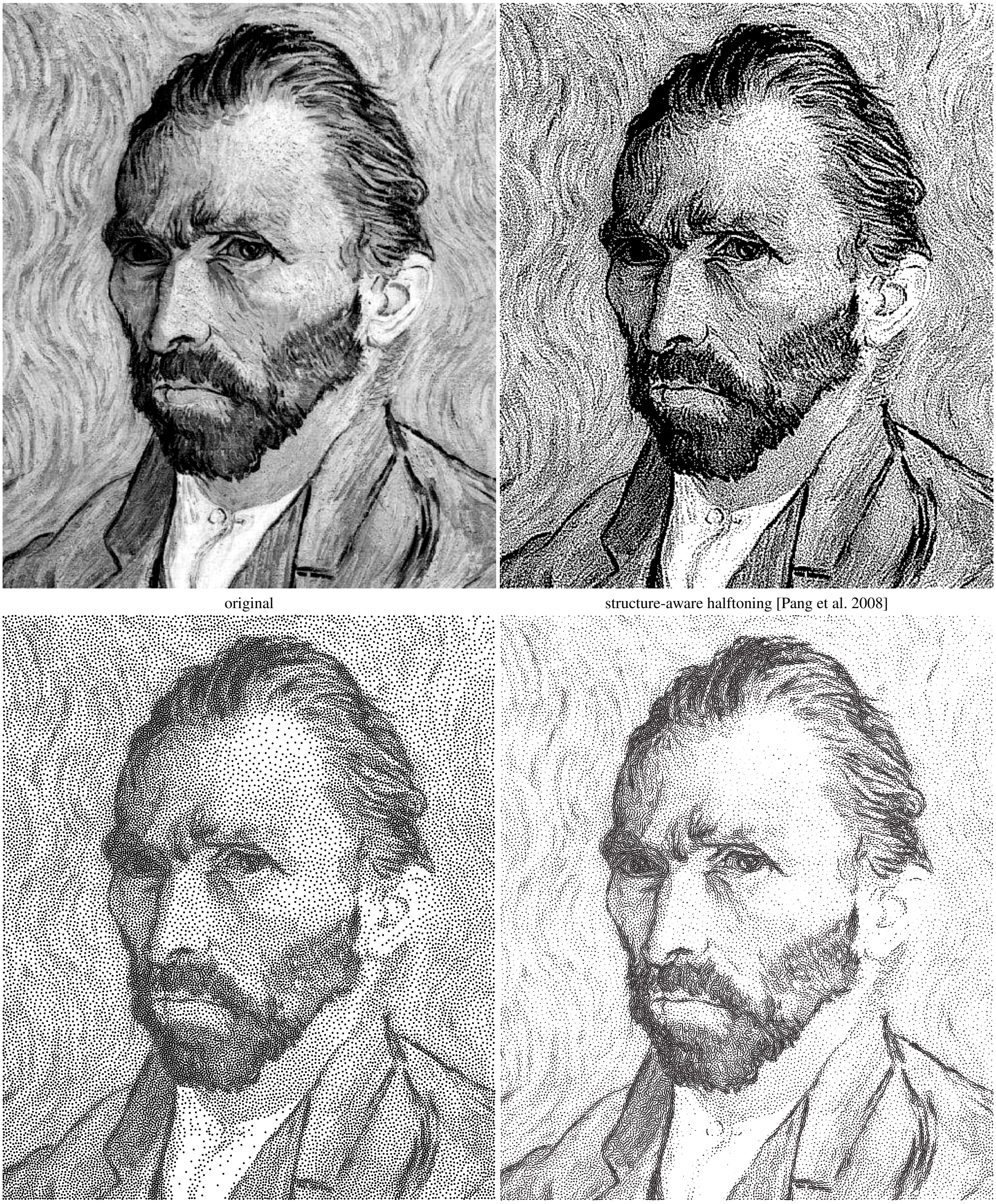


Figure 23: Stippling effects with different combinations of isotropic/anisotropic primitives and distributions. Here, we use the concentric circular Jacobian field in Figure 9 for didactic illustration. As shown, both anisotropic primitive and anisotropic distribution help achieve the best effect. Note that the first and fourth results share the same isotropic distribution, while the second and the fifth results share the same anisotropic distribution. For further comparison, we also produce another result (third case) with isotropic primitive + anisotropic distribution (from second case) with the Jacobian field rotated 90 degrees. Due to the nature of blue noise, samples usually will not align along flow directions as in other methods (e.g., [Kim et al. 2008]) where explicit alignment is performed.



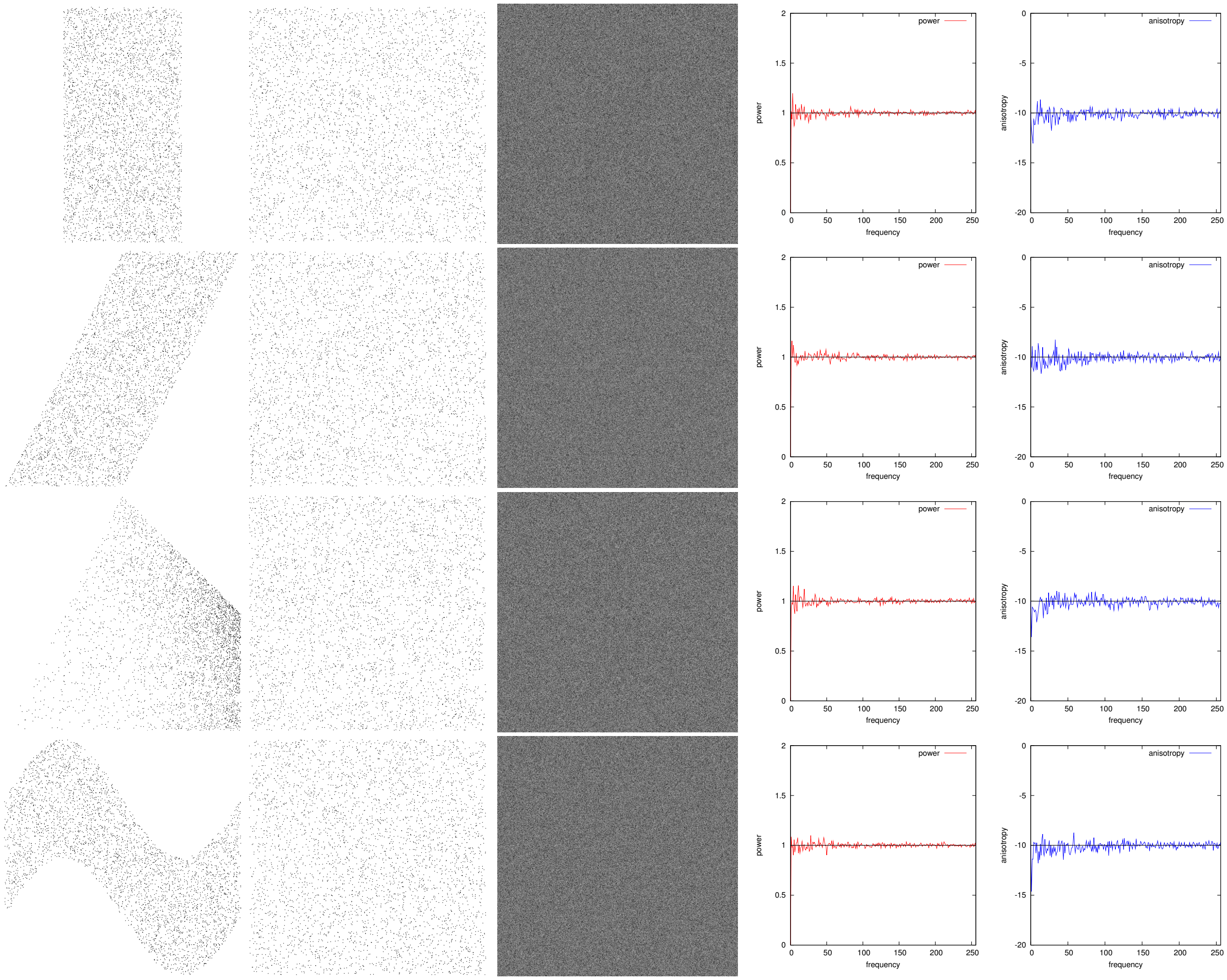
original

structure-aware halftoning [Pang et al. 2008]

isotropic stippling [Balzer et al. 2009]

anisotropic stippling [our method]

Figure 24: Anisotropic stippling. Same as Figure 16, except that we use vector graphics pdf files for [Balzer et al. 2009] and our method.



anisotropic samples (warped) uniform samples power spectrum radial mean radial anisotropy

Figure 25: Anisotropic white noise sampling under 4 different warp domains (Figure 3). Here, we apply our verification method in Section 4.1 to anisotropic white noise. Each case contains ∼4000 samples generated by our algorithm in Section 3.1.