

Wasserstein Blue Noise Sampling

HONGXING QIN

Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Shandong University

YI CHEN and JINLONG HE

College of Computer Science and Technology, Chongqing University of Posts and Telecommunications and

BAOQUAN CHEN

Shandong University

In this article, we present a multi-class blue noise sampling algorithm by throwing samples as the constrained Wasserstein barycenter of multiple density distributions. Using an entropic regularization term, a constrained transport plan in the optimal transport problem is provided to break the partition required by the previous Capacity-Constrained Voronoi Tessellation method. The entropic regularization term cannot only control spatial regularity of blue noise sampling, but it also reduces conflicts between the desired centroids of Voronoi cells for multi-class sampling. Moreover, the adaptive blue noise property is guaranteed for each individual class, as well as their combined class. Our method can be easily extended to multi-class sampling on a point set surface. We also demonstrate applications in object distribution and color stippling.

Categories and Subject Descriptors: I.4.1 [Computer Graphics]: Digitization and Image Capture—*Sampling*

General Terms: Image and Geometry Processing

Additional Key Words and Phrases: Blue noise sampling, Wasserstein barycenter, adaptive sampling, surface sampling

ACM Reference Format:

Hongxing Qin, Yi Chen, Jinlong He, and Baoquan Chen. 2017. Wasserstein blue noise sampling. *ACM Trans. Graph.* 36, 5, Article 168 (October 2017), 13 pages.

DOI: <http://dx.doi.org/10.1145/3119910>

We acknowledge the National Natural Science Foundation of China (61772097, U1401252) and State Key Development Program of Basic Research of China (2015CB352501).

Authors' addresses: H. Qin, Y. Chen, J. He, and B. Chen (Corresponding author); emails; redstarq@gmail.com, 449624979@qq.com, jinlonghe_dorcy@foxmail.com, baoquan.chen@gmail.com.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2017 ACM 0730-0301/2017/10-ART168 \$15.00

DOI: <http://dx.doi.org/10.1145/3119910>

1. INTRODUCTION

Sampling is a ubiquitous problem for many graphics applications [Lagae and Dutré 2008], such as rendering [Mitchell 1987], imaging [Ulichney 1988], geometry processing [Öztireli et al. 2010], and visualization [Chen et al. 2014]. Since blue noise sampling distribution yields superior image quality over alternative distributions with the same number of samples, it has been researched extensively in recent decades. Prior research primarily focused on single-class sampling. In fact, multi-class sampling is involved in a variety of sampling problems, such as color stippling, visual abstraction of multi-class scatterplots, and so on. In these situations, the blue noise distribution property needs to be preserved for every individual class of samples and their union simultaneously. Although optimal sampling [Balzer et al. 2009; Chen et al. 2012; de Goes et al. 2012] generates a distribution with good blue noise property for single-class sampling, multi-class blue noise sampling remains challenging.

There has recently been work on multi-class sampling. Based on the analysis of the power spectrum characteristics of combined classes, digital filter techniques are applied to generate multi-class samples in Wang and Parker [1999]. In their method, $2^n - 1$ kinds of combined patterns need to be created, where n is the number of classes. The problem tends to become more complicated with an increasing number of classes. Dart throwing is extended to generate multi-class samples in Wei [2010]. The main idea is to create a conflict matrix on the accumulated two-class sampling for explicit control of sample spacing. When the density distribution of each class is non-uniform and different from the others, it is difficult to set off-diagonal elements of the conflict matrix to balance the blue noise property of individual classes of samples and their combination. Chen et al. [2012] extended blue noise sampling with Capacity-Constrained Voronoi Tessellation (CCVT) [Balzer et al. 2009] for multi-class sampling. However, it is nontrivial to preserve the uniformity of both individual classes and the combined classes simultaneously due to the conflict between their Voronoi diagrams. The reason is that the conflict may make multi-class relaxation get stuck in local minimums with insufficient sample uniformity, which is called the failure case in Wei [2010].

In this article, we present a Wasserstein blue noise sampling algorithm by throwing samples as a constrained Wasserstein barycenter, which is constrained to be a sum of Diracs, of multiple density distributions. The blue noise property of each class, either individual or combined, is preserved by assuming that their distributions are independent when computing the constrained Wasserstein barycenter. Our algorithm avoids the failure case of multi-class sampling on traditional Lloyd relaxation [Wei 2010] and on Voronoi

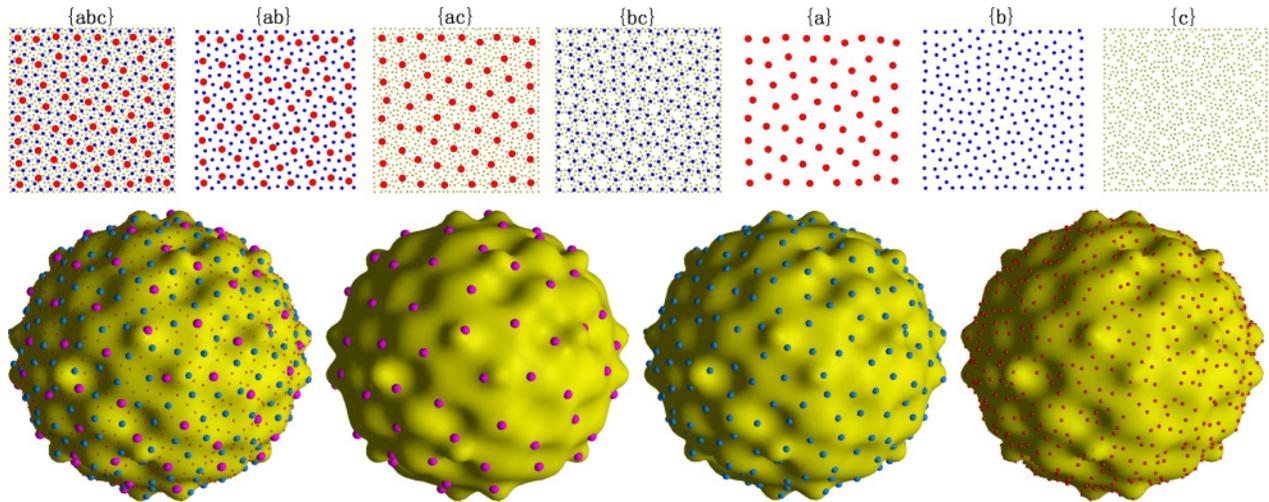


Fig. 1. Multi-class blue noise sampling with different sizes on three identical constant density functions. Sample distribution with the larger size remains the blue noise pattern, and the sample distribution with the smaller size might deviate from the blue noise pattern. The number of samples for each class is 62, 250, 1,000 (top) and 100, 300, 900 (bottom). The weighted parameters are set as $\lambda_{1,2,3,4} = (1, 1, 1, 1)/4$.

tessellation [Chen et al. 2012], since the partition required in these two methods is broken in our method. On the other hand, the efficiency of sampling is improved by applying Wasserstein distance with entropic constraints [Cuturi and Doucet 2014; Solomon et al. 2015]. Our method, furthermore, can be directly applied to multi-class sampling on a point set surface [Öztireli et al. 2010].

1.1 Motivation and Contributions

Although a series of CCVT-based algorithms can generate high-quality blue noise samples, they are not suitable to be directly extended to multi-class blue noise sampling. The main reason is that it is very difficult to build an adaptive equi-capacity partitioning for multi-class sampling, which is the key to ensuring equal visual importance for samples. Therefore, we aim at a general algorithm that is more feasible for multi-class sampling, adaptive sampling and surface sampling.

In this article, we present an adaptive multi-class blue noise sampling algorithm by throwing samples as a constrained Wasserstein barycenter of multiple density distribution. At the core of our algorithm, it generates point distributions by minimizing a Wasserstein distance with an entropic regularization. Experiments show that our approach can produce high-quality blue noise spectrums for either individual or combined classes. Our contributions are as follows:

- We formulate multi-class sampling as a constrained Wasserstein barycenter of multiple density functions, where constrained transport plans are used to replace the capacity constraints in the equal capacity partitioning.
- Via an entropic regularization term, the constrained transport plan breaks the partition required by the previous CCVT method. The entropic regularization term cannot only control spatial regularity of blue noise sampling, but it also avoids the failure case of multi-class sampling [Wei 2010].
- The proposed approach can be easily extended to support adaptive sampling and general domain sampling, such as sampling on a point set surface.

2. RELATED WORK

2.1 Blue Noise Sampling

The blue noise distribution was introduced to solve problems of image anti-aliasing [Dippé and Wold 1985]. Due to the huge applied values of its natural property (random and uniform distributions that match the distribution of the retina in human eyes), it has been widely used to solve diverse kinds of problems in computer graphics. There are three classical methods to generate blue noise distributions. The first one is stochastic sampling. The traditional method for stochastic sampling is called dart throwing and was first introduced by Cook [1986]. During the sampling, dart throwing positions samples one by one randomly and accepts a new sample, just when there are no other samples within a disk of a given radius surrounding it. The basic idea is simple and easy to implement but extremely time-consuming. It is possible that it might never terminate. Since then, many variations of dart throwing were proposed to solve the problem [Mitchell 1987; Jones 2006; Dunbar and Humphreys 2006; White et al. 2007; Wei 2008; Gamito and Maddock 2009; Ebeida et al. 2011, 2012]. However, it is still difficult to control the number of points generated with these methods, and the distribution that is produced also has a relatively high variance by variance standards. Recently, Yuksel described a sample elimination method that allows controlling the number of samples [Yuksel 2015].

Optimal sampling is another type of method to generate higher quality sampling. A well-known approach in this class is Lloyd relaxation [Lloyd 1982]. The basic idea of this method is to relax the original distribution to make it obtain blue noise characteristics and improve the spectrum properties [McCool and Fiume 1992]. However, the samples produced suffer from too much regularity, which causes an aliasing problem. To improve the irregularity, Baler et al. [2009] proposed a variant of Lloyd’s method with capacity constraint that enforces that each point obtains equal importance in the distribution. Xu et al. [2011] proposed Capacity-Constrained Delaunay Triangulation (CCDT) by replacing the Voronoi cell of CCVT with Delaunay triangulation. Furthermore, CCDT is also extended to Capacity-constrained Surface Triangulation (CCST) [Xu

et al. 2012] for surface sampling. De Goes et al. [2012] considered CCVT as an optimal transport problem in the space of power diagrams (CCVT-PD). Furthermore, Xin et al. [2016] developed a super-linear convergence algorithm for computing optimal Power Diagram partition with capacity constraints. In addition to Lloyd's method, the kernel density model is also applied to optimal sampling. Fattal [2011] used a radially symmetric kernel function to produce an approximated density function. The difference between the approximation and the given target point density function is that it assigns an energy value to the points configuration. Jiang et al. [2015] used a kernel function to apply the smoothed particle hydrodynamics method (SPH) for a variety of controllable blue noise sample patterns. Compared with de Goes et al. [2012], our approach approximates the Wasserstein distance using a regularized transport plan instead of a partition. Our approach is, therefore, feasible for multi-class sampling.

2.2 Multi-class Sampling

Wang and Parker [1999] studied the power spectrum characteristics of combined blue noise patterns and proposed a multi-class blue noise sampling approach on digital filter techniques. However, the approach is constrained to uniform, regular, and discrete sampling. Wei [2010], Schmaltz et al. [2012], and Jiang et al. [2015] proposed multi-class sampling approaches by enforcing the blue noise property through the use of an interaction matrix that encodes the spacing between class pairs. However, the matrix can exhibit discontinuous changes in the off-diagonal entries, which represent the coupling between the different classes' distribution. Chen et al. [2012] proposed a multi-class sampling strategy based on capacity-constrained Voronoi tessellation. A two-stage algorithm is applied to the optimization of point distributions. The first stage is the optimizing procedure for each individual class, and the second stage is the optimizing procedure for the combined class. In our approach, the combined class is directly treated as a single class. Normalized parameters are used to balance the blue noise properties of different classes. Compared with Wang and Parker [1999], our approach preserves blue noise patterns in the density space rather than in the spectrum space. Compared with Chen et al. [2012], our approach optimizes the point distribution of each individual class and that of the combined class simultaneously.

2.3 Wasserstein Barycenter

For completeness, we briefly review the Wasserstein barycenter in this section. By analogy with the barycenter of points (x_1, \dots, x_n) in the Euclidean space, which is obtained as the minimizer of $x \leftarrow \sum_{i=1}^n \|x - x_i\|^2$, Wasserstein barycenter is defined as the barycenter of probability measures ν_1, \dots, ν_n in the Wasserstein space by replacing the Euclidean distance with the Wasserstein distance.

For an arbitrary space Ω , we use $d : \Omega \times \Omega \rightarrow \mathbb{R}_+$ to denote the distance metric, so $d(x, y)$ is the shortest distance from x to y along Ω . We use $P(\Omega)$ to indicate the set of probability measures on Ω , and $P(\Omega \times \Omega)$ to indicate that on the product space $\Omega \times \Omega$,

$$P(\Omega) = \{\mu | \mu(\emptyset) = 0, \mu(\Omega) = 1, 0 \leq \mu(U) \leq 1 (U \subseteq \Omega)\},$$

where $\mu(U) = \int_{U \subseteq \Omega} \rho(x) dx$ is a probability measure on Ω , and $\rho(x) \geq 0$ is a probability density function on Ω .

Given probability measures μ and ν in $P(\Omega)$ and $p \geq 1$, the p -Wasserstein distance [Villani 2008] between μ and ν is defined

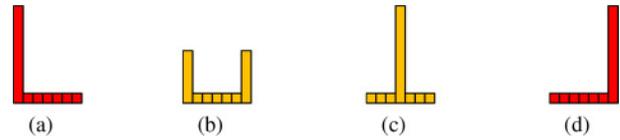


Fig. 2. Wasserstein barycenter. (a) and (d) are two probability densities ρ_0 and ρ_1 . (b) is the barycenter of ν_1 and ν_2 using l_2 distance, and (c) is the Wasserstein barycenter of ν_1 and ν_2 with $\lambda_1 = \lambda_2 = 0.5$.

as

$$W_p(\mu, \nu) = \left(\inf_{\pi \in \Pi(\mu, \nu)} \int_{\Omega \times \Omega} d(x, y)^p d\pi(x, y) \right)^{1/p},$$

where $d(x, y)$ is the cost of transporting one unit of mass from x to y , and $\Pi(\mu, \nu)$ is the subset of $P(\Omega \times \Omega)$ and meets mass conservation laws

$$\Pi(\mu, \nu) = \{\pi \in P(\Omega, \Omega) | \pi(\cdot, \Omega) = \mu, \pi(\Omega, \cdot) = \nu\}.$$

π is a transportation plan, which describes the amount of mass $\pi(x, y)$ to be placed from μ at x toward y to create ν in aggregation. Wasserstein distance, therefore, describes the minimum cost of transporting the source μ to the target ν .

Based on the Wasserstein distance, the Wasserstein barycenter [Aguet and Carlier 2011; Cuturi and Doucet 2014; Bonneel et al. 2016] of N probability measures ν_1, \dots, ν_N in $P(\Omega)$ is defined as

$$\mu = \arg \min_{\mu} \sum_{i=1}^N \lambda_i W_p^p(\mu, \nu_i) \quad s.t. \quad \sum_{i=1}^N \lambda_i = 1, \lambda_i \geq 0. \quad (1)$$

The Wasserstein barycenter is an optimal probability measure μ to approximate probability measures ν_1, \dots, ν_N with weights in the Wasserstein space. If the Wasserstein barycenter μ is a discrete probability measure, then μ represents sampling points. The Wasserstein barycenter is, therefore, a general approach for sampling in the space Ω . Figure 2 shows a simple example about Wasserstein barycenter.

3. ALGORITHM

In this article, we take single-class blue noise samples as the constrained Wasserstein barycenter of a probability measure. Multi-class blue noise samples are chosen as the combined Wasserstein barycenter of multiple probability measures.

3.1 Single-Class Blue Noise Sampling

Given an arbitrary space Ω and a density function $\rho(x)$ on the space Ω , a probability measure is defined as

$$\begin{aligned} \nu(U) &= \int_U \rho(x) dx, U \subseteq \Omega, \\ s.t. \quad \int_{\Omega} \rho(x) dx &= 1, \rho(x) \geq 0. \end{aligned} \quad (2)$$

Sampling the density function $\rho(x)$ consists of picking a few representative points x_i that capture ρ well. These representative points x_i represent a discrete probability measure μ ,

$$\mu = \sum_{i=1}^n \rho_i \delta_{x_i} \quad s.t. \quad \sum_{i=1}^n \rho_i = 1, \delta_{x_i} = \begin{cases} 1 & x_i \in \Omega \\ 0 & \text{others,} \end{cases}$$

where n is the number of samples and ρ_i is the probability at x_i . Sampling, therefore, means that the method aims to find μ that

captures ν well. We can use the constrained Wasserstein barycenter to model a single-class sampling as

$$\begin{aligned} \mu &= \arg \min_{\mu} W_p^p(\mu, \nu) = \arg \min_{\mu} \inf_{\pi} \int_{X \times \Omega} d(x_i, y)^p d\pi(x_i, y), \\ \text{s.t. } \int_{\Omega} \pi(x_i, y) &= \rho_i, \sum_{i=1}^n \int_{U \subseteq \Omega} \pi(x_i, y) dy = \nu(U), \end{aligned} \quad (3)$$

where $X = \{x_1, x_2, \dots, x_n\} \subseteq \Omega$.

Compared with previous methods on CCVT [Balzer et al. 2009; de Goes et al. 2012], the constrained Wasserstein barycenter provides a more general framework for sampling. Previous methods on CCVT are special cases of Equation (3). In the previous method, the transport plan π , which is the solution for the discrete transportation problem [Balzer et al. 2009] or semi-discrete optimal transportation problem [de Goes et al. 2012], satisfies the property

$$\pi(x_i, y) \cdot \pi(x_j, y) = 0 \quad (i \neq j). \quad (4)$$

Equation (4) means that the mass in a certain domain can only be transported to one sampling point. For example, the certain domain corresponding sampling points is a few discrete points in the Voronoi cell in Balzer et al. [2009], and the power diagram region in de Goes et al. [2012]. In fact, Equation (4) is a very strict condition for the variational problem in Equation (3). It can only be satisfied when ν is a continuous probability measure. It also introduces a “failure case” for multi-class sampling [Wei 2010]. We will discuss how to break the condition on the transport plan (Equation (4)) and the reason for the “failure case” in the next subsection.

3.2 Multi-class Blue Noise Sampling

For multi-class blue noise sampling, each individual class and combined classes exhibit blue noise characteristics [Wei 2010]. In this regard, we model multi-class sampling by accounting for individual probability measures and combined probability measures simultaneously.

Let $\{\varrho_1, \dots, \varrho_N\}$ be N individual probability density functions to be sampled on the space Ω . Let $\{\varrho_{i_1, \dots, i_k}\} (k \leq N, i \in \{N+1, N+2, \dots, K\}, i_j \in \{1, 2, \dots, N\}, N < K \leq 2^N - 1)$ be $K - N$ probability density functions of the combined classes, which is combined by the i_1 th, \dots , i_k th classes, and

$$\varrho_{i_1, i_2, \dots, i_k} = (\varrho_{i_1} + \varrho_{i_2} + \dots + \varrho_{i_k})/k.$$

Let X_i be sampling points of the probability density function ϱ_i ,

$$X_i = \{x_i^1, x_i^2, \dots, x_i^{n_i}\} \quad (1 \leq i \leq N), x_i^j \in \Omega (1 \leq j \leq n_i),$$

where n_i is the number of sampling points X_i .

In terms of Equation (2), there is a corresponding probability measure $\nu_i (1 \leq i \leq K)$ for every ϱ_i or $\varrho_{i_1, i_2, \dots, i_k}$. The probability measure $\nu_i (1 \leq i \leq N)$ corresponds to ϱ_i , and the probability measure $\nu_i (N < i \leq K)$ corresponds to $\varrho_{i_1, \dots, i_k}$. For sampling points X_i or $\{X_{i_1}, X_{i_2}, \dots, X_{i_k}\}$, there is a corresponding discrete probability measure $\mu_i (1 \leq i \leq K)$. For multi-class blue noise sampling, every class sampling point X_i should capture the probability function ϱ_i well, and the combined sampling points $\{X_{i_1}, X_{i_2}, \dots, X_{i_k}\}$ should also capture the probability density function $\varrho_{i_1, i_2, \dots, i_k}$ well. Thus, every discrete probability measure μ_i should approximate the corresponding probability measure ν_i well.

Compared with a single-class sampling, we cannot compute an optimal μ_i for every ν_i in terms of Equation (3) independently for

multi-class sampling. This is because sampling points X_i are influenced by individual probability measures ν_i and combined probability measures $\nu_i (i'_j = i)$ simultaneously. We describe multi-class sampling as a combined Wasserstein barycenter,

$$\bar{\mu} = \arg \min_{\bar{\mu}} \sum_{i=1}^K \lambda_i W_p^p(\mu_i, \nu_i) \quad \text{s.t.} \quad \sum_{i=1}^K \lambda_i = 1, (\lambda_i \geq 0), \quad (5)$$

where $\bar{\mu} = \{\mu_1, \dots, \mu_N, \mu_{N+1}, \dots, \mu_K\}$, $\mu_i (1 \leq i \leq N)$ is a discrete probability measure corresponding to X_i , $\mu_i (N < i \leq K)$ is a discrete probability measure corresponding to combined sampling points.

In an extended version of Wei [2010], Wei pointed out that multi-class sampling on a traditional relaxation method [Balzer et al. 2009] might result in insufficient sample uniformity for the reason below: for a given sample s , different class combinations may have different opinions about the desired centroid location to which s should move. In essence, conflicts of desired centroid locations are introduced by the hard condition on transport plans π (Equation (4)). In the traditional relaxation method [Balzer et al. 2009], the mass in a certain domain can only be transported to the nearest sampling point. Thus, the transport plan is local, and the minimum unit of transported mass is constrained by the resolution of the domain. The two limits make relaxing conflicts of desired centroid locations more difficult.

To preserve sufficient sample uniformity for multi-class sampling, we introduce the regularized Wasserstein distance instead of Wasserstein distance in Equation (5) via entropic regular terms as Cuturi [2013]. Regularized Wasserstein distance is defined as

$$W_p(\mu, \nu) = \left(\inf_{\pi \in \Gamma(\mu, \nu)} \int_{\Omega \times \Omega} d(x, y)^p d\pi(x, y) + \epsilon H(\pi) \right)^{1/p}, \quad (6)$$

where ϵ is a positive regularization parameter, and $H(\pi) = -\int_{\Omega \times \Omega} \pi(x, y) \ln \pi(x, y) dx dy$ is the entropy of π . Compared with traditional Wasserstein distance, the optimal transport plan corresponding to regularized Wasserstein distance is required with sufficient smoothness [Cuturi 2013]. Entropic regular terms, therefore, relax transport plans and break the condition on the transport plan (Equation (4)). The optimal transport plan π is a separable transport plan in Equation (6), which means the mass, located in a small domain, can be transported to different samples other than only one sampling point. It extends the minimum unit of transported mass to computational accuracy. Furthermore, Entropic regularization terms expand the transport range to the whole domain rather than the nearest transport in the traditional relaxation method [Balzer et al. 2009]. It provides more flexibility for reducing conflicts of the desired centroid. Compared with multi-class sampling [Wei 2010], our approach is an efficient relaxation method for multi-class blue noise sampling.

3.3 Numerical Optimization

Equation (5) is a unified variational model for single-class and multi-class sampling. We only consider numerical optimization of Equation (5) for multi-class sampling. For every discrete probability measure $\mu_i \in \bar{\mu}$, we represent it as

$$\mu_i = \sum_{j=1}^{n_i} \rho_i^j \delta_{x_i^j} \quad \text{s.t.} \quad \sum_{j=1}^{n_i} \rho_i^j = 1, \quad (7)$$

where ρ_i^j is probability at sampling point x_i^j for the i th-class sampling and n_i is the number of sampling points for the i th-class sampling. For convenience of representation and computation, we

```

01: //Wasserstein blue noise sampling
02: Input: domain  $\Omega$ , densities  $\varrho_1, \dots, \varrho_N$ , coordinates of discrete points
 $\mathbf{Y}$ , number of sampling points  $n_1, \dots, n_N$ , number of combined classes
 $K - N$  and combined classes  $\{i_1, \dots, i_k\}$ 
03: Output: coordinates  $\mathbf{X} = \{x_1^1 \dots x_1^{n_1} \dots x_n^1 \dots x_n^{n_N}\}$  of sampled
points
04: Initialize  $\mathbf{X}$  with random points inside  $\Omega$  conforming to  $\rho_1, \dots, \rho_N$ 
05: Initialize parameters  $\epsilon$  for the entropic regularization
06: Initialize weight parameters  $\lambda_i (1 \leq i \leq K)$ 
07: Initialize transport cost matrices  $\Pi_k = \frac{1}{nm} \mathbf{1}$ ,  $\mathbf{1}$  is an all-1 matrix
08: Compute the distance matrix  $\mathbf{D}$ 
09: Compute the energy  $E_0 = \sum_{i=1}^K \lambda_i < \mathbf{D}, \Pi_i >$ 
10: Repeat
11: // Iterative Bregman projections for  $\Pi_k$ 
12: For  $i=1:K$ 
13:   Iterative-Bregman-projection() (lines 24-32)
14: // Newton iterative method for  $\mathbf{X}$ 
15: Generate  $K$  random numbers  $r_1, \dots, r_K \in [0, 1]$ 
16: Normalize  $r_1, \dots, r_K, \sum_{i=1}^K r_i = 1$ 
17: Update  $\mathbf{X}$ ,  $\mathbf{X} = (1 - \theta)\mathbf{X} + \theta \sum_{i=1}^K \lambda_i r_i \mathbf{X} \Pi_i^T \text{diag}(\rho_i^{-1})$ 
18: Update  $D$ 
19:  $E_1 = \sum_{i=1}^K \lambda_i < \mathbf{D}, \Pi_i >$ 
20:  $dE = E_1 - E_0, E_0 = E_1$ 
21: Compute the distance matrix  $\mathbf{D}$ 
22: Update  $\epsilon$ 
23: Until  $dE$  is convergent

24: Subroutine Iterative Bregman projections()
25: Input: the distance matrix  $D$ , the parameter  $\epsilon$ , the density function  $\mathbf{p} =$ 
 $\rho_k, \mathbf{q} = \varrho_k$ 
26: Initiate  $i = 0, \mathbf{v}^i = \mathbf{1}$  (all-1 vector)
27: Initiate  $\xi = e^{-\mathbf{D}/\epsilon}$ 
28: Repeat
29:  $\mathbf{u}^i = \frac{\mathbf{p}}{\xi_k \mathbf{v}^i}$ 
30:  $i = i + 1, \mathbf{v}^i = \frac{\mathbf{q}}{\xi_k^T \mathbf{u}^i}$ 
31: Until  $\mathbf{u}^i$  is convergent
32:  $\Pi_k = \text{diag}(\mathbf{u}^i) \xi_k \text{diag}(\mathbf{v}^i)$ 
    
```

Fig. 3. Pseudocode of multi-class blue noise algorithm.

extend the discrete probability measure μ_i to all class sampling points. We define all sampling points as

$$X = (X_1, X_2, \dots, X_N)^T = (x_1^1, \dots, x_1^{n_1}, \dots, x_N^1, \dots, x_N^{n_N})^T, \quad (8)$$

and the probability vector at all sampling points is

$$\mathbf{g}_i = \underbrace{(0, 0, \dots, 0, \rho_i^1, \rho_i^2, \dots, \rho_i^{n_i}, 0, 0, \dots, 0)^T}_{\sum_{j=1}^{i-1} n_j}, \quad (9)$$

for the i th-class probability measure. Thus, we can represent the discrete probability measure μ_i as

$$\mu_i = \sum_{j=1}^n g_i^j \delta_{x_i^j}, \quad (10)$$

where $n = \sum_{i=1}^N n_i$. For blue noise sampling, ρ_i^j is a constant and $\rho_i^j = 1/n_i$. Thus, we need only to compute variables X in Equation (5).

Every probability measure v_i in Equation (5) may be a continuous probability measure or a discrete probability measure. If v_i is a continuous probability measure, then we discretize v_i as a discrete probability measure to compute the integrations required in

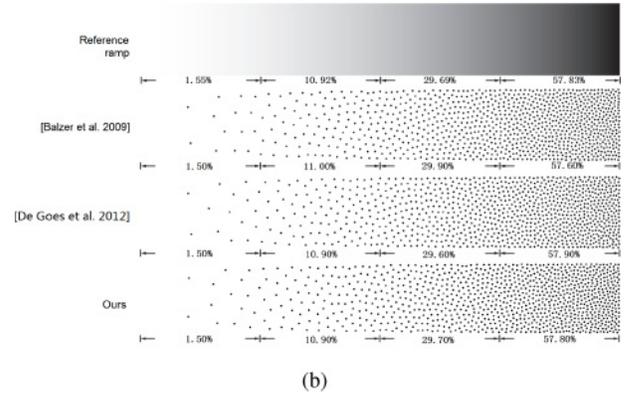
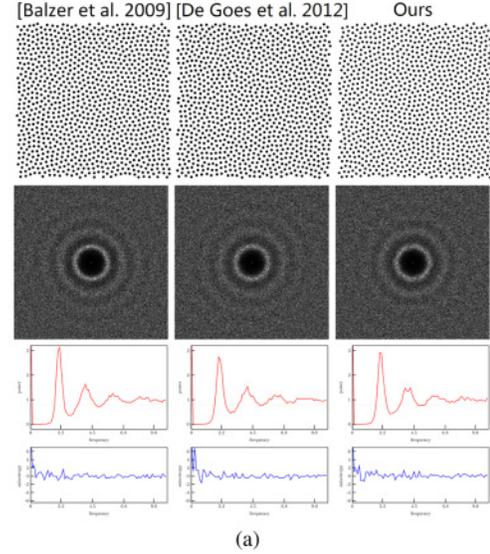


Fig. 4. Comparisons of CCVT, CCVT-PD, and our algorithm for a single-class blue noise sampling. (a) shows the comparison for the case of constant density. (b) shows the comparison for the case of a quadratic density function.

Equation (5) by quadrature,

$$v_i = \sum_{j=1}^{m_i} \varrho_i^j \delta_{y_i^j} \quad s.t. \quad \sum_{j=1}^{m_i} \varrho_i^j = 1, \quad (11)$$

where $\varrho_i^j = \varrho_i(y_i^j) V_{y_i^j}$ and $V_{y_i^j}$ is the area with density $\varrho_i(y_i^j)$ at the point y_i^j when the density function ϱ_i is approximated with a positive piecewise constant function, ϱ_i is the probability density function of the probability measure v_i , and m_i is the number of discrete points of the i th-class probability measure v_i . In this article, we apply a regular discretization with the same area for all sampling points y_i^j . We also apply the same sampling points for all probability measures v_i , which is

$$m_i = m_k = m, \quad y_i^j = y_k^j = y^j \quad i \neq k.$$

Thus, we use $\mathbf{Y} = (y^1, y^2, \dots, y^m)^T$ to represent all sampling points for probability measures v_i . Based on the discrete representation of probability measures in Equation (11), the variational problem

Table I. Comparison of Several Frequencies and Spatial Statistics of Sampling Patterns

Methods	v_{eff}	Ω	δ_{min}	δ_{avg}	Q_6	
[Balzer et al. 2009]	0.88	2.395	0.765	0.886	0.488	
[De Goes et al. 2012]	0.855	2.014	0.739	0.872	0.417	
Our algorithm	$\epsilon = 1/150$	0.865	2.136	0.76	0.877	0.42
	$\epsilon = 1/40$	0.275	0.211	0.012	0.508	0.367
	$\epsilon = 1/80$	0.665	0.572	0.342	0.755	0.347
	$\epsilon = 1/120$	0.815	1.686	0.548	0.846	0.397
	$\epsilon = 1/160$	0.840	1.857	0.673	0.856	0.415
	$\epsilon = 1/200$	0.910	2.998	0.561	0.894	0.557
	$\epsilon = 1/240$	0.905	2.905	0.568	0.893	0.537
	$\epsilon = 1/280$	0.910	2.994	0.564	0.894	0.592
$\epsilon = 1/320$	0.910	2.969	0.526	0.894	0.595	

Notes: v_{eff} and Ω are the effective Nyquist and oscillation measures. The value of δ_{min} corresponds to the Poisson disk radius, and δ_{avg} roughly measures how uniformly the points are distributed [Schlömer et al. 2011]. Q_6 measures the similarity of a point distribution to a hexagonal arrangement [Kansal et al. 2000].

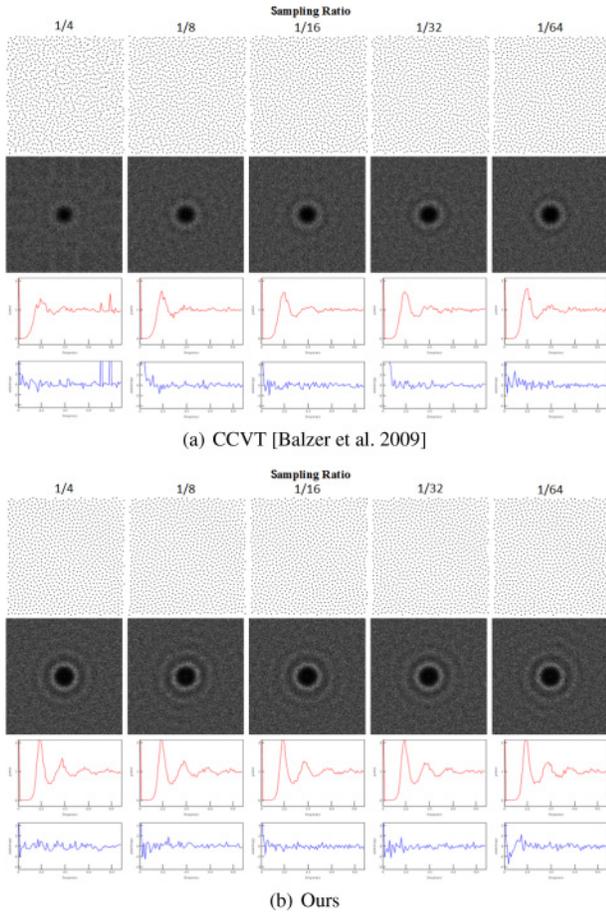


Fig. 5. Evaluation of sampling ratio. The number of sampling points is 1,024. When the sampling ratio is less than or equal to 1/4, the blue noise property is preserved well in our approach.

(Equation (5)) can be represented as

$$\mathbf{X} = \arg \min_{\mathbf{X}} \sum_{i=1}^K \lambda_i \langle \mathbf{D}_i, \mathbf{\Pi}_i \rangle, \quad (12)$$

$$s.t. \quad \mathbf{\Pi}_i \mathbf{1} = \mathbf{q}_i, \quad \mathbf{\Pi}_i^T \mathbf{1} = \mathbf{g}_i, \quad \sum_{i=1}^K \lambda_i = 1, \lambda_i \geq 0,$$

where $\mathbf{D}_i, \mathbf{\Pi}_i \in \mathbb{R}_+^{n \times m}$, \mathbf{D}_i is the distance matrix and $D_i^{(j,k)} = d(x_i^j, y^k)^p$, $\mathbf{\Pi}_i$ is the i th-class transport cost matrix and $\Pi_i^{(j,k)} = \Pi_i(x_i^j, y^k)$ represents the amount of mass transported from x_i^j to y^k for the i th-class, \mathbf{q}_i is the probability density vector and $\mathbf{q}_i = (q_i^1, q_i^2, \dots, q_i^m)$, λ_i is a weighted parameter, $\mathbf{1}$ is an all-1 vector, and $\langle \cdot, \cdot \rangle$ is the Frobenius product of two matrices.

We proceed with sample generation by computing an optimal solution of the variational problem (Equation (12)). We apply a loop iteration algorithm to extremize the variational problem by repeatedly performing a minimization step over positions \mathbf{X} followed by a projection step over transport plans $\mathbf{\Pi}_k$.

For a fixed set of points \mathbf{X} , we compute relaxed transport plans through a regularized Wasserstein distance (Equation (6)). Equation (6) can be discretized as

$$W_p^p(\mathbf{D}_i, \mathbf{\Pi}_i) = \arg \min_{\mathbf{\Pi}_i} \langle \mathbf{D}_i, \mathbf{\Pi}_i \rangle - \epsilon H(\mathbf{\Pi}_i), \quad (13)$$

$$s.t. \quad \mathbf{\Pi}_i \mathbf{1} = \mathbf{q}_i, \quad \mathbf{\Pi}_i^T \mathbf{1} = \mathbf{g}_i.$$

Furthermore, Equation (13) can be re-written as the smallest Kulback-Leibler (KL) divergence,

$$W_p^p(\mathbf{D}_i, \mathbf{\Pi}_i) = \epsilon \min_{\mathbf{\Pi}_i} \text{KL}(\mathbf{\Pi}_i | \xi_i),$$

from $\mathbf{\Pi}_i$ to the distance-based kernel $\xi_i = e^{-\frac{D_i}{\epsilon}}$ (here, the exponential is computed component-wise). KL divergence is defined as

$$\text{KL}(\mathbf{\Pi}_i | \xi_i) = \sum_j \sum_k \Pi_i^{(j,k)} \left(\log \left(\frac{\Pi_i^{(j,k)}}{\xi_i^{(j,k)}} \right) - 1 \right).$$

This minimization is convex. The optimal transport plan $\mathbf{\Pi}_i$ is the projection of the distance-based kernel ξ_i onto $\mathbf{\Pi}_i$, and can be obtained by iterative Bregman projection [Cuturi 2013; Benamou et al. 2015],

$$\mathbf{\Pi}_i = \text{diag}(\mathbf{u}) \xi_i \text{diag}(\mathbf{v}), \quad (14)$$

where $\mathbf{u} \in \mathbb{R}_+^n$ and $\mathbf{v} \in \mathbb{R}_+^m$. More details can be found in Cuturi [2013], Cuturi and Doucet [2014], and Benamou et al. [2015].

Suppose $\Pi_i (i = 1, \dots, K)$ is fixed and the variational formula Equation (12) is the sum of a convex quadratic function of \mathbf{X} when $p = 2$. If we only update \mathbf{X}_i at every iteration, then \mathbf{X} can be obtained by the Newton iterative method,

$$\mathbf{X} \leftarrow (1 - \theta)\mathbf{X} + \theta \mathbf{Y} \mathbf{\Pi}_i^T \text{diag}(\rho_i^{-1}) \quad (0 \leq \theta \leq 1). \quad (15)$$

We also apply a random Newton iterative method to update all sampling points \mathbf{X} simultaneously,

$$\mathbf{X} \leftarrow (1 - \theta)\mathbf{X} + \theta \sum_{i=1}^K \lambda_i r_i \mathbf{Y} \mathbf{\Pi}_i^T \text{diag}(\rho_i^{-1}), \quad (16)$$

$$s.t. \quad \sum_{i=1}^K r_i = 1, r_i \geq 0,$$

where r_i is a random number. Figure 3 shows the pseudocode of this schedule. Figure 6 compares the results of two class sampling

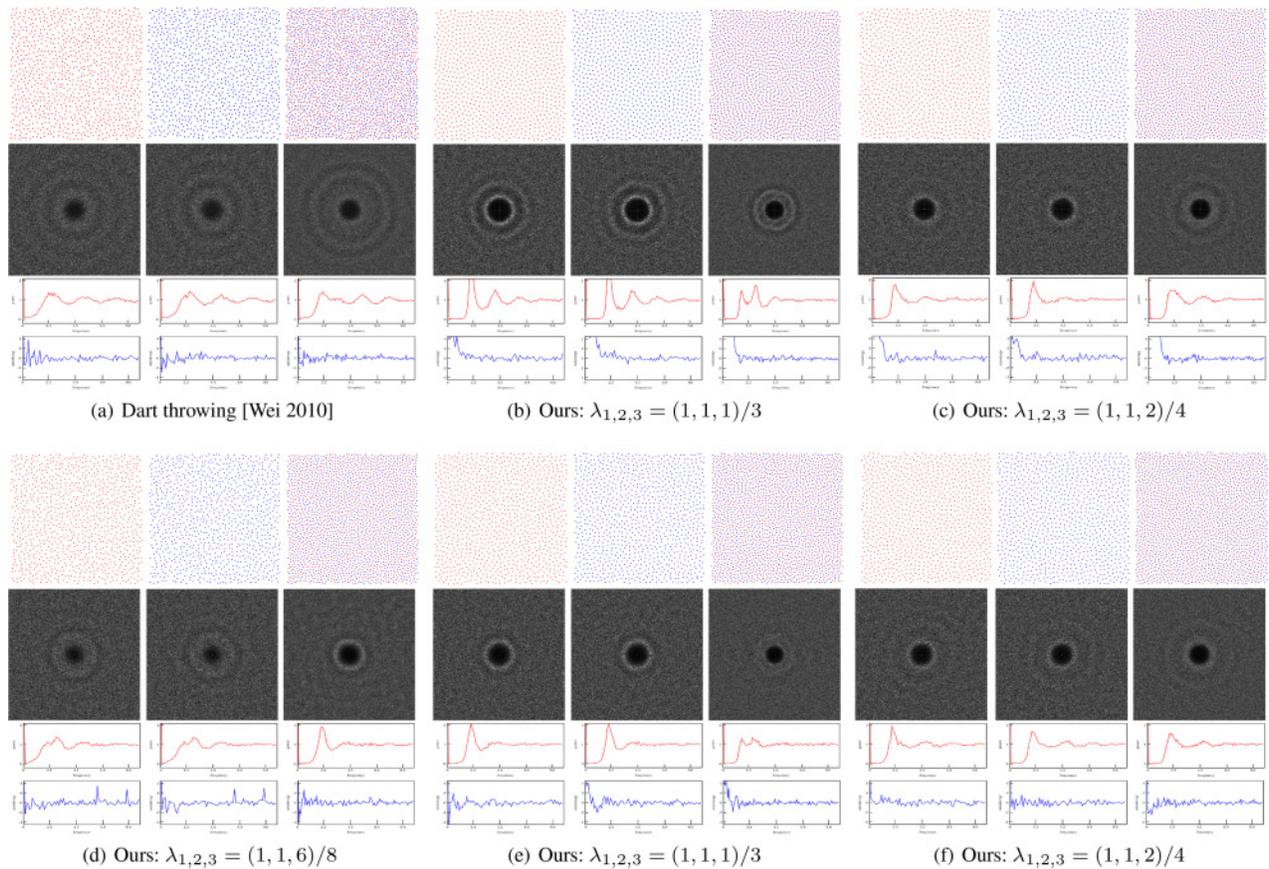


Fig. 6. The comparison of Wei’s algorithm and our algorithm for two-class blue noise sampling. In our algorithm, $\lambda_1, \lambda_2, \lambda_3$ are the weighted parameters for red samples, blue samples, and the combined samples. The number of samples for each class is 1000. (b) and (c) are the results without random parameters r_i . (d), (e), and (f) are the results with random parameters r_i . The weighted parameter λ_i has an influence on the trade-off between a good distribution of the samples belonging to each individual class and a good distribution of total samples. Large λ_i preserves the blue noise property of the i th well.

with and without random parameters r_i . The results show that there is no significant difference between the results. The reason is that the entropy regularization term has already played an important role to improve randomness of samples. The random parameters r_i only play a minor role for randomness of samples.

4. RESULTS

We ran our algorithm on a variety of inputs: from constant density to stripe and a point set surface. Various illustrations based on regularity and spectral analysis are used throughout the article to allow for an easy evaluation of our algorithm and to demonstrate how it compares to previous works. We use methods in Schlömer and Deussen [2011] and Wei and Wang [2011] to analyze the spectrum properties of sample distributions in two-dimensional space and two-dimensional manifold separately. The relative radius [Lagae and Dutré 2008] is used to analyze the spatial uniformity. In all experiments, we set $\theta = 0.8$.

Single class sampling. Blue noise point distribution for a constant density shows a characteristic blue noise profile in spectral space and a typical spatial arrangement. In Figure 4, we compare our algorithm with the CCVT [Balzer et al. 2009] and CCVT-PD [de Goes et al. 2012] methods for the case of constant density in

spectral space. We also provide evaluations of the spatial properties in Table I. We also evaluate blue noise sampling for an intensity ramp in Figure 4 by counting the number of points for each quarter of the ramp. It can be shown that our algorithm is comparable to CCVT [Balzer et al. 2009] and CCVT-PD [de Goes et al. 2012] for both the cases of constant density and the intensity ramp.

Since the numerical optimization that is involved is based on the discrete representation of sampled density functions in our approach, we evaluate the effect of the sampling ratio (n/m) on blue noise property by comparing our approach with CCVT [Balzer et al. 2009] in Figure 5. When the sampling ratio is equal to $1/4$, sampling on CCVT [Balzer et al. 2009] does not show the blue noise property well. With the decreasing sampling ratio, the blue noise property is shown more and more effectively. However, the sampling ratio hasn’t an important effect in our approach. This is because the relaxed transport plan makes more original points exert influence on one sampling point in our approach, while only the nearest original points have a role in CCVT [Balzer et al. 2009].

Multi-class sampling on constant density. To evaluate the blue noise distribution for multi-class sampling, we first analyze the simplest case where all classes have a constant density function and the same number of samples. Figures 6–8 illustrate the result generated by our method, compared to Wei’s [2010] and SPH

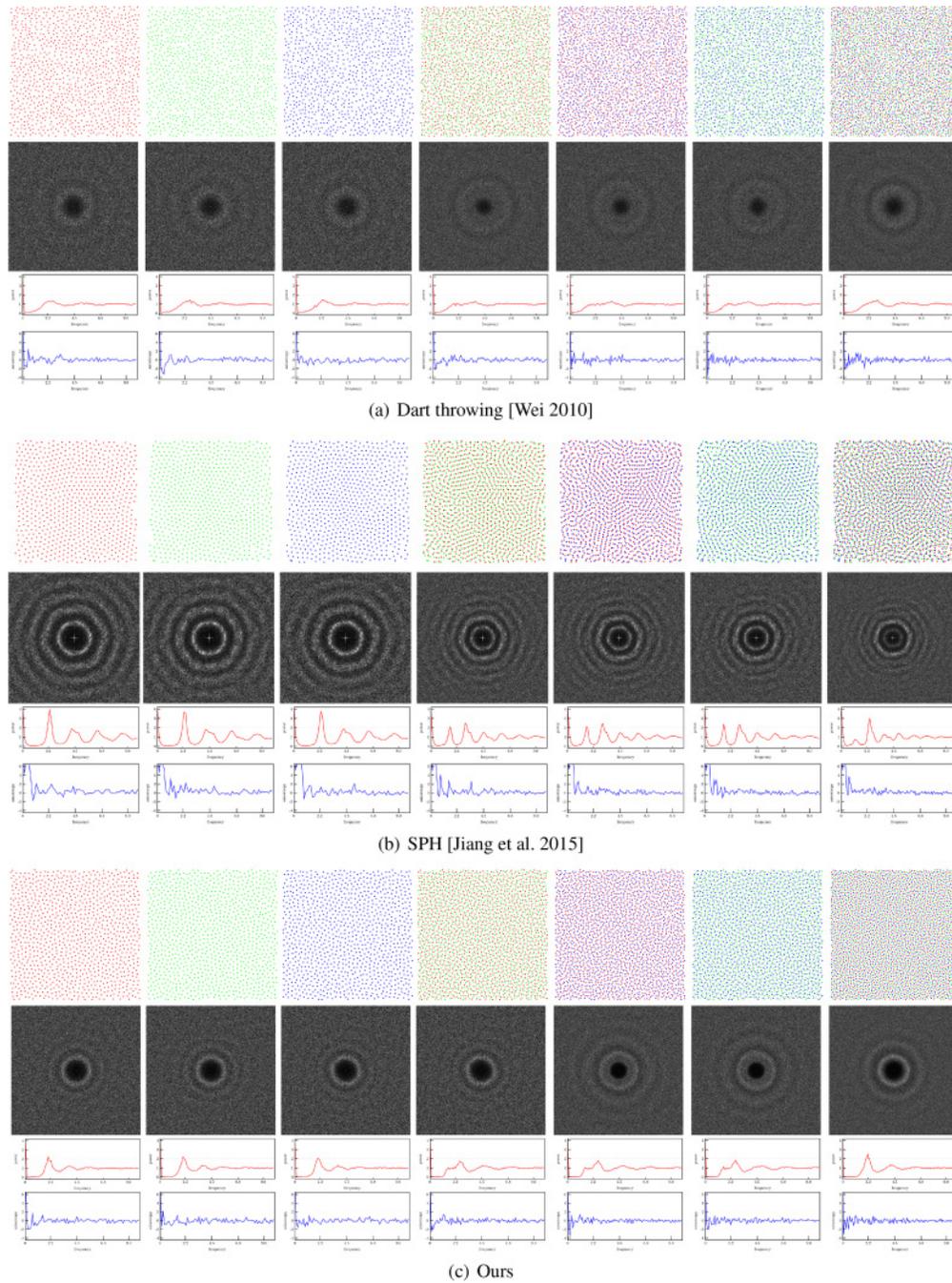


Fig. 7. The comparison of Wei’s algorithm, SPH, and our algorithm for three-class blue noise sampling. $\lambda_{1,2,3,4,5,6,7} = (1, 1, 1, 2, 2, 2, 9)/18$. λ_1, λ_2 , and λ_3 are weighted parameters for each individual class. λ_4, λ_5 , and λ_6 are weighted parameters for combined classes with two individual classes. λ_7 is the weighted parameters for the total samples. The number of samples of each individual class is 1,024.

[Jiang et al. 2015] on two-class, three-class, and seven-class sampling separately. In two-class sampling (Figure 6), we illustrate the influence of the weighted parameter λ_i on the trade-off between a good distribution of the samples belonging to each individual class and a good distribution of the total samples. A Large λ_i preserves the blue noise property of the i th-class well. Depending on which

distribution is more important, λ_i can be chosen for different applications. When $\lambda_1 + \lambda_2 = \lambda_3$, the balance between a good distribution of the samples belonging to each individual class and a good distribution of the total samples is achieved in Figure 6(f). It shows that our results are comparable to that of Wei’s. In three-class sampling (Figure 7), we evaluate the blue noise distribution of samples of

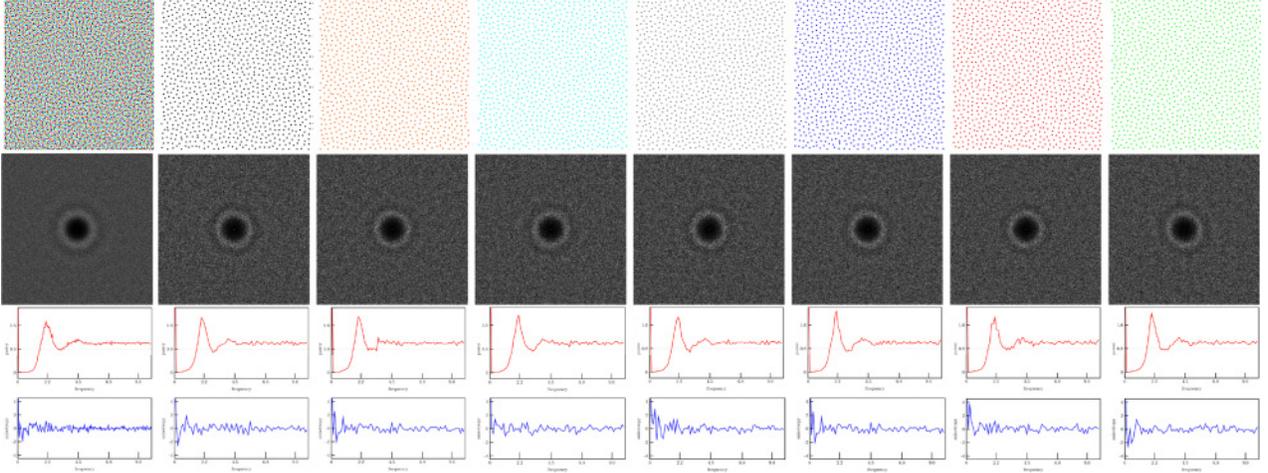


Fig. 8. Seven-class blue noise sampling on our algorithm. $\lambda_{1,2,3,4,5,6,7,8} = (1, 1, 1, 1, 1, 1, 1, 7)/14$. $\lambda_1, \dots, \lambda_7$ are weighted parameters for each individual class, λ_8 is the weighted parameter for the total set. The number of samples of each individual class is 1,024.

different combined classes. Compared with Wei’s method [Wei 2010] and SPH [Jiang et al. 2015], the blue noise profile of the combined class with two individual classes is improved in our method while the blue noise distribution of each individual class is preserved. In seven-class sampling (Figure 8), the blue noise profile is still kept for samples of each individual class and total class. It shows that our method is feasible for more numbers of class sampling.

To evaluate the efficiency of relaxing conflicts of desired centroid locations in our approach, we compare our approach with Wei’s method [2010] on the difference between centroid locations of Voronoi diagrams for single-class samples and multi-class samples in Figure 9. We apply a conflicts ratio to evaluate the difference between desired centroid locations. The conflicts ratio is defined as

$$R = \left(\sum_{i=1}^N \sum_{j=1}^{n_i} \|C_{x_i^j}^I - C_{x_i^j}^C\| \right) / \left(\sum_{i=1}^N \sum_{j=1}^{n_i} D(x_i^j) \right),$$

where $C_{x_i^j}^I$ and $C_{x_i^j}^C$ are centroid locations of Voronoi diagrams of the sample x_i^j for individual class samples and combined class samples, and $D(x_i^j)$ is the distance between the sample x_i^j and its nearest neighbor sample and n is the number of samples. Compared with Wei’s approach [2010], the conflicts ratio in our approach is decreased by approximately 50%. It shows that our approach avoids the failure case of multi-class sampling in Wei [2010], and is an efficient relaxation method for multi-class sampling.

Adaptive multi-class sampling on a ramp. To evaluate multi-class blue noise sampling on a non-uniform density function, we applied two intensity ramps and counted the number of points for each quarter of the ramp. Figure 10 shows samples generated by our method and Wei [2010]. While samples of every individual class and the combined class represent approximately the proper counting of points per quarter, it can be seen that our results show noticeably less noise.

Multi-class sampling on a point set surface. Our method can be directly applied to multi-class sampling on a point set surface. We assume that the original points represent a discrete probability measure $\nu = \sum_{k=1}^m \varrho_k \delta_{y^k}$ as Equation (11) on the surface. ϱ_k is set as the

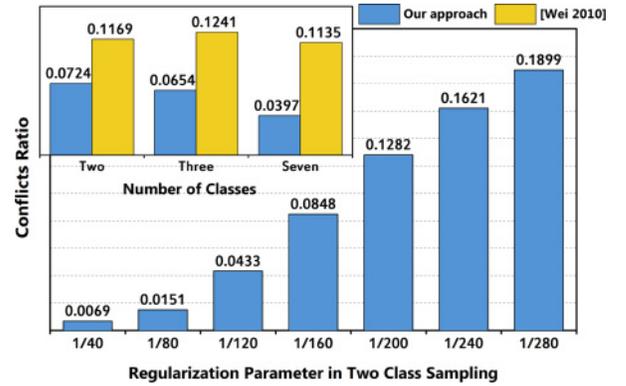


Fig. 9. The evaluation for conflicts of desired Voronoi centroid locations of individual classes and combined class.

normalized area element at the point y^k for multi-class blue noise sampling with a constant density function. $D_i^{(j,k)}$ is the geodesic distance. For convenience, we applied the Euclidean distance instead of the geodesic distance in this article. Samples are initialized on the surface. After each iteration, every sample is mapped back onto the surface by a moving least-square projection [Alexa et al. 2001]. In Figure 11, we make a two-class blue noise sampling on a bunny model and show the Differential Domain Function [Wei and Wang 2011] to demonstrate the blue noise profile of different class sample distributions.

Multi-class blue noise sampling with different sizes. Analogous to setting a different radius of Poisson disks for different classes in dart throwing, different sizes of samples can be set using a discrete probability measure μ_i for different classes in our method. Suppose that the same constant density function $\varrho_i (1 \leq i \leq N)$ is used for different classes, we set the discrete probability density as $\rho_i^j = 1/n_i$ for every individual class and $\rho_i^j = 1/Nn_i$ for every combined class, where N is the number of individual classes of the combined class. When two-class sampling is done with N_1 and N_2 samples ($N_1 > N_2$), every sample belonging to the second class is given more measure than every sample belonging to the first class

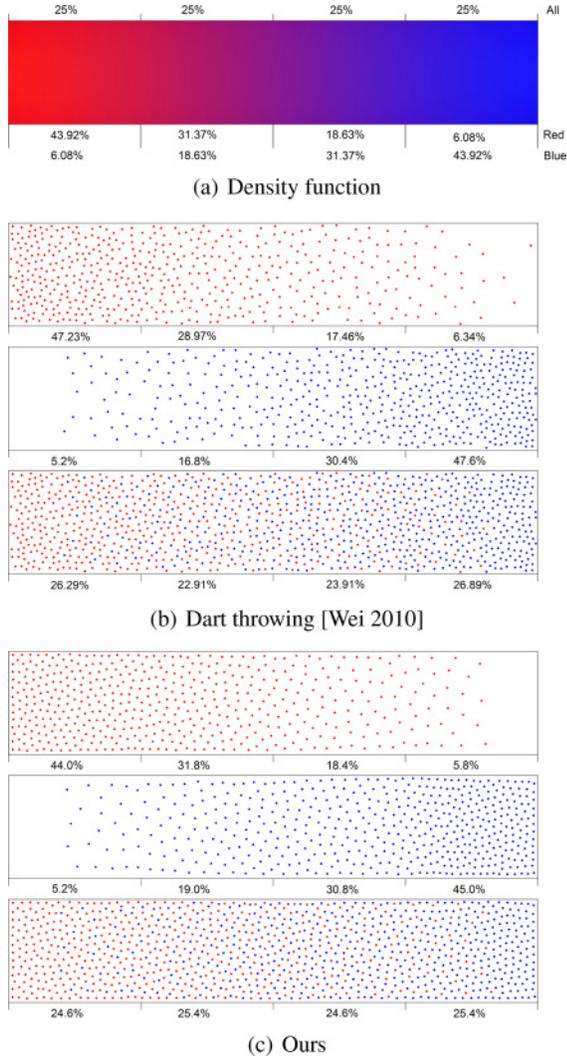


Fig. 10. Adaptive two-class sampling of a non-uniform density function with 1,000 points for each class. The percentages in each quarter indicate the ink density of different colors in the image. By contrast, our results show precise adaptation for every single sampling and total sampling.

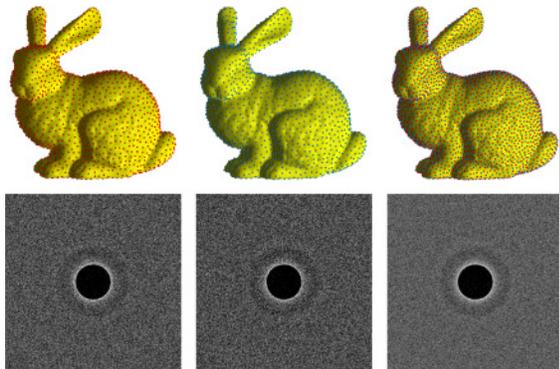


Fig. 11. Two-class blue noise sampling on a point set surface. $\lambda_{1,2,3} = (1, 1, 2)/4$, and the number of samples belonging to each class is 2,000.

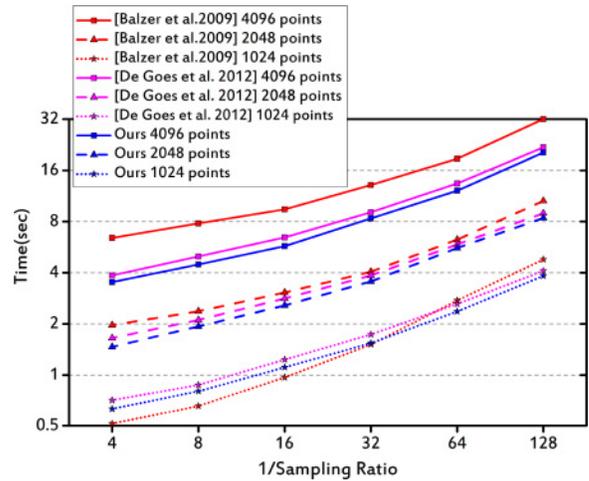


Fig. 12. Running time in log scale. Comparison of sampling times of single-class blue noise sampling on different sampling ratios.

in the combined class. When the density function of the combined class is a constant function, the second-class sample has a larger size than the first-class sample. The radius of every sample can be approximated as

$$r_i = \lambda r_{i_{max}} / n,$$

where $r_{i_{max}}$ is the maximum possible disk radius of the i th class and λ is a relative radius parameter. In our experiments, we found that there is an overlap between any pair of samples when $0 \leq \lambda \leq 0.5$. In Figure 15, our method is applied for three-class blue noise sampling with different sizes on a square and a two-dimensional manifold.

The entropic regularization parameter ϵ . Compared with the previous method on CCVT, the regularized Wasserstein distance (Equation (6)) is used in the proposed approach instead of the Wasserstein distance in CCVT [Balzer et al. 2009; de Goes et al. 2012]. When the regularization parameter ϵ is set as 0, the regularized Wasserstein distance degenerates to Wasserstein distance. We evaluate the importance of regularized Wasserstein distance by analyzing the role of the regularization parameter ϵ . In Figure 13, we compare single blue noise samplings on different ϵ for the case of a constant density in spectral space. We also provide evaluations of the spatial properties in Table I. It shows that various blue noise profiles can be generated on different ϵ . We also have found that spatial regularity of sampling becomes more and more prominent with the decreasing of ϵ . In fact, the regularity term in Equation (6) reduces the solution space of Equation (3) without the regularity term. Hexagonal lattices are the optimal solutions of the problem in Equation (3) without the regularity term. The regularity term breaks the optimal solutions. Therefore, we can set ϵ to control spatial regularity of sampling, other than using a stochastically scrambled configuration in the numeral procedure [de Goes et al. 2012]. We also evaluate the role of the parameter ϵ for multi-class sampling in Figure 9. We compare conflicts ratios on different ϵ for two class sampling. The conflicts ratio is becoming less and less with the increasing of ϵ , since the regularization terms expand the transport range. Therefore, it also interprets the reason why the failure case happens when Wasserstein distance is used in Wei's [2010]. For single sampling and multi-class sampling, blue noise property is preserved well when $1/120 < \epsilon < 1/160$. We set $\epsilon = 1/150$ by default in other experiments.

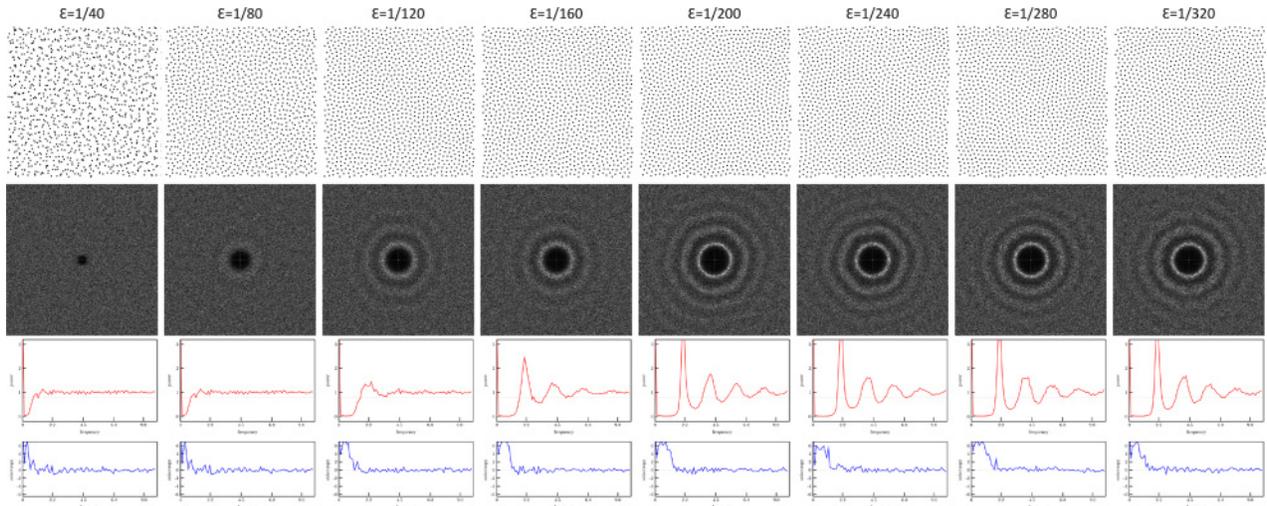


Fig. 13. Comparison of single blue noise samplings on different ϵ for the case of a constant density. The number of sampling points is 1,024. Various blue noise profiles can be generated on different ϵ . Spatial regularity of samples becomes more and more prominent with the decreasing of ϵ .

Complexity. One iteration of our algorithm involves an iterative Bregman projection. The time complexity depends on what classes are involved in the framework. For a single-class sampling, the time complexity is $O(MN)$. Although this is worse than $O(N \log N + M)$, an iterative Bregman projection can be easily parallelized on the GPU. When only a total set is used as a combined class, the time complexity is $O(2MN)$. If all $2^n - 1$ classes are involved, then the complexity is $O(2^{n-1}MN)$. In most applications, it is enough for only every single class and a combined class of the total set to be with the blue noise property, such as in Figures 16 and 17. Therefore, the time complexity is $O(2MN)$ in general. While our algorithm performs well for small datasets, our method is limited by its memory requirements $O(2MN)$ for large datasets, such as the experiment in Figure 16.

Performance. All of our performance is measured on a workstation with Intel Xeon 3.50GHz dual-core CPUs and 32GB memory, and NVIDIA Quadro K5000 GPU with 2GB memory. For single blue noise sampling, we compare our method with the traditional CCVT [Balzer et al. 2009] and power diagrams method [de Goes et al. 2012] on running time in Figure 12. Our approach exhibits comparable performance and even outperforms the power diagrams method [de Goes et al. 2012] on running time. Since the most time is spent on integral computation in the power diagrams method [de Goes et al. 2012] and distance matrix computation in our approach, it is possible that CCVT [Balzer et al. 2009] performs better in running time than these two methods when the number of sampling points is small. With the increasing of the sampling points, our approach outperforms CCVT [Balzer et al. 2009]. When the number of sampling points is larger than 4,096, our approach is almost two times faster than CCVT [Balzer et al. 2009]. For multi-class blue noise sampling, we compare our method with Wei’s method [Wei 2010] on running time in Table II. Our approach is at least two times faster than Wei’s method [Wei 2010]. In Table II, we also show the memory usage in our approach for single-class and multi-class sampling. Since we have used a global distance matrix in our algorithm, more memory space is needed to store the matrix in our algorithm than in the other algorithm. In every iteration, updating the matrix accounts for most of the running time. Our approach, therefore, faces a challenge when there

Table II. Memory Usage and Comparison of Sampling Times of Single Blue Noise Sampling

Number of Class	Times(s)		Memory(M)
	[Wei 2010]	Ours	Ours
2	34.769	14.670	128.11
3	151.727	37.553	288.11
7	414.031	212.133	448.31

The number of sampling points of every individual class is 512. The sampling ratio is 1/16 for every individual class sampling.

are large numbers of original points and sampling points. However, we also notice that it is, in general, a sparse matrix. It is possible to further improve the performance of our approach by making use of a sparse distance matrix.

5. APPLICATIONS

Surface reconstruction of noisy point clouds. Surface reconstruction of noisy point clouds is a challenging problem. In this article, we take original point clouds with noise as a discrete probability measure of a two-dimensional surface, and we take samples on the surface as another discrete probability measure. Samples on the surface are the constrained Wasserstein barycenter of the original point clouds. We can apply a single-class blue noise sampling to generate samples on the surface. Triangulation is then implemented by the open source code provided by Rahmani et al. [2014]. Figure 14 shows an example for surface reconstruction of point clouds with heavy noise. Since the entropic regular term in the Wasserstein barycenter plays a role in the smoothing of the transport plan with hard constraint (Equation (4)), our approach can generate a smooth surface.

Object distribution. Object placement without regularity artifacts plays an important role in texture synthesis. Blue noise sampling provides a feasible approach for such a distribution. However, placing multi-class objects with different sizes on the surface is a difficult problem. Our approach can be directly applied to this problem. Figure 1 shows an example of placing three class objects with different sizes on a square. Figures 1 and 15 show two examples for

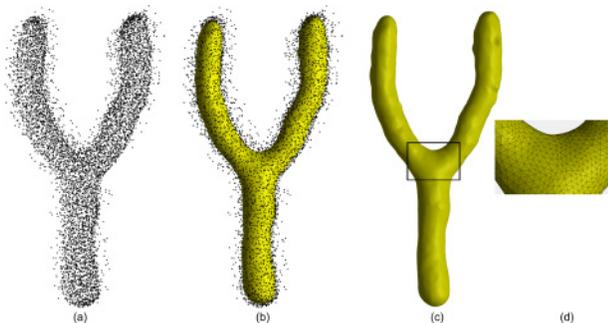


Fig. 14. Surface reconstruction of noisy point clouds. (a) is the original point clouds with 8,856 points. (c) is the mesh reconstruction result on 3,946 sampling points. (b) is the rendering result with point clouds (a) and mesh model (c). (d) is the close-up view of (c) at the square mark.



Fig. 15. Object placement on a complex geometry model. The number of samples of each class is 200, 1,000, and 1,000.

placing three class objects with different sizes on two-dimensional surfaces.

Color stippling. Color stippling is another very important application related to blue noise sampling. We can apply our method to multi-class color stippling on a CMY-RGB-K color model using the input image as a weighted density field. The weighted value represents the importance of each color channel and is used to set the number of each color dot. More dots are placed in the area where the corresponding color is dominant. The discrete probability value ρ_i^j is set as a constant $1/n_i$ for every class. As shown in Figure 16, our method can achieve reasonable color stippling. Note that the visual quality depends on the color decomposition and blending method.

Visual abstraction. Scatterplots are widely used to visualize a scatter dataset. When multi-class scattered points are shown within a single scatterplot view, heavy overdraw of different class points makes it inefficient for analysis (Figure 17). We apply our algorithm to reduce the overdraw of original scattered points and to preserve the point distributions in Figure 17. Compared with the dart throwing method [Wei 2010; Chen et al. 2014], it is not necessary to compute a continuous density function for each class in our method.

6. CONCLUSION AND FUTURE WORK

We have presented a new multi-class blue noise sampling method that takes initial samples and optimizes them to be constrained Wasserstein barycenters of multi-class density function. Our approach introduces weighted parameters to explicitly adjust point distributions of each individual class and the combined classes. By adjusting the discrete density value at samples, we are able to generate multi-class samples with different sizes. Our approach is also easy to be implemented in parallel, which ensures the efficiency of computations. We have also performed applications on object placement and color stippling.

ACM Transactions on Graphics, Vol. 36, No. 5, Article 168, Publication date: October 2017.



Fig. 16. Color stippling by using the RGB-CMYK color model. The number of samples is 48k.

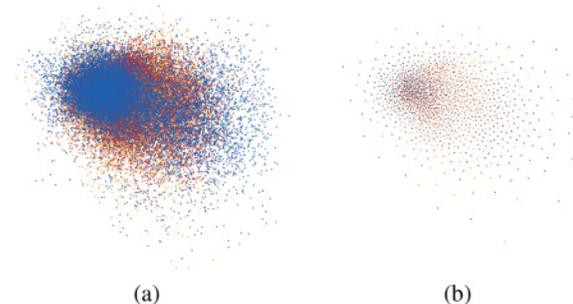


Fig. 17. Visual abstraction of multi-class scatterplots. The numbers of original points in (a) are 14,880, 14,130, and 12,600 for yellow, red, and blue points, respectively. The numbers of samples in (b) are 496, 471, and 420.

In the future, an interesting problem would be to study how to extend our approach to the synthesis of point distributions. Another possible direction is to study how to improve the efficiency for large databases.

ACKNOWLEDGMENTS

We thank Liyi Wei, Yunhai Wang, and Lin Lv for discussion and suggestions, and Chen Zhonggui for providing the data in Figure 15. We are also grateful to the anonymous reviewers for their helpful comments.

REFERENCES

- Martial Agueh and Guillaume Carlier. 2011. Barycenters in the Wasserstein space. *SIAM J. Math. Anal.* 43, 2 (2011), 904–924.
- Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T. Silva. 2001. Point set surfaces. In *Proceedings of the Conference on IEEE Visualization*. 21–28.

- Michael Balzer, Thomas Schlömer, and Oliver Deussen. 2009. Capacity-constrained point distributions: A variant of Lloyd's method. *ACM Trans. Graph. (TOG)* 28, 3 (2009), 86:1–86:8.
- Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. 2015. Iterative Bregman projections for regularized transportation problems. *SIAM J. Sci. Comput.* 37, 2 (2015), A1111–A1138.
- Nicolas Bonneel, Gabriel Peyré, and Marco Cuturi. 2016. Wasserstein barycentric coordinates: Histogram regression using optimal transport. *ACM Trans. Graph. (TOG)* 35, 4 (2016), 71:1–71:10.
- Haidong Chen, Wei Chen, Honghui Mei, Zhiqi Liu, Kun Zhou, Weifeng Chen, Wentao Gu, and Kwan-Liu Ma. 2014. Visual abstraction and exploration of multi-class scatterplots. *IEEE Trans. Visual. Comput. Graph.* 20, 12 (2014), 1683–1692.
- Zhonggui Chen, Zhan Yuan, Yi-King Choi, Ligang Liu, and Wenping Wang. 2012. Variational blue noise sampling. *IEEE Trans. Visual. Comput. Graph.* 18, 10 (2012), 1784–1796.
- Robert L. Cook. 1986. Stochastic sampling in computer graphics. *ACM Trans. Graph. (TOG)* 5, 1 (1986), 51–72.
- Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Adv. Neural Info. Process. Syst.* 2292–2300.
- Marco Cuturi and Arnaud Doucet. 2014. Fast computation of Wasserstein barycenters. *Proceedings of The 31st International Conference on Machine Learning* (2014), 685–693.
- Fernando de Goes, Katherine Breeden, Victor Ostromoukhov, and Mathieu Desbrun. 2012. Blue noise through optimal transport. *ACM Trans. Graph. (TOG)* 31, 6 (2012), 171:1–171:11.
- Mark A. Z. Dippé and Erling Henry Wold. 1985. Antialiasing through stochastic sampling. *ACM Siggraph Comput. Graph. (TOG)* 19, 3 (1985), 69–78.
- Daniel Dunbar and Greg Humphreys. 2006. A spatial data structure for fast Poisson-disk sample generation. *ACM Trans. Graph. (TOG)* 25, 3 (2006), 503–508.
- Mohamed S. Ebeida, Andrew A. Davidson, Anjul Patney, Patrick M. Knupp, Scott A. Mitchell, and John D. Owens. 2011. Efficient maximal Poisson-disk sampling. *ACM Trans. Graph. (TOG)* 30, 4 (2011), 49:1–49:12.
- Mohamed S. Ebeida, Scott A. Mitchell, Anjul Patney, Andrew A. Davidson, and John D. Owens. 2012. A simple algorithm for maximal poisson-disk sampling in high dimensions. *Computer Graphics Forum* 31, 24 (2012), 785–794.
- Raanan Fattal. 2011. Blue-noise point sampling using kernel density model. 30, 4 (2011), 48:1–48:10.
- Manuel N. Gamito and Steve C. Maddock. 2009. Accurate multidimensional Poisson-Disk sampling. *ACM Trans. Graph. (TOG)* 29, 1 (2009), 81–8:19.
- Min Jiang, Yahan Zhou, Rui Wang, Richard Southern, and Jian Jun Zhang. 2015. Blue noise sampling using an SPH-based method. *ACM Trans. Graph. (TOG)* 34, 6 (2015), 211:1–211:11.
- Thouis R. Jones. 2006. Efficient generation of Poisson-Disk sampling patterns. *J. Graph. GPU Game Tools* 11, 2 (2006), 27–36.
- Anuraag R. Kansal, Thomas M. Truskett, and Salvatore Torquato. 2000. Nonequilibrium hard-disk packings with controlled orientational order. *J. Chem. Phys.* 113, 12 (2000), 4844–4851.
- A. Lagae and P. Dutré. 2008. A comparison of methods for generating Poisson disk distributions. *Comput. Graph. Forum* 27, 1 (2008), 114–129.
- Stuart P. Lloyd. 1982. Least squares quantization in PCM. *IEEE Trans. Info. Theory* 28, 2 (1982), 129–137.
- Michael McCool and Eugene Fiume. 1992. Hierarchical Poisson disk sampling distributions. In *Proceedings of the Conference on Graphics Interface*, Vol. 92. 94–105.
- D. P. Mitchell. 1987. Generating antialiased images at low sampling densities. In *Proceedings of the 14th ACM SIGGRAPH*. ACM, 65–72.
- A. Cengiz Öztireli, Marc Alexa, and Markus Gross. 2010. Spectral sampling of manifolds. *ACM Trans. Graph. (TOG)* 29, 6 (2010), 168:1–168:8.
- Hossein Rahmani, Arif Mahmood, Du Q. Huynh, and Ajmal Mian. 2014. HOPC: Histogram of oriented principal components of 3D pointclouds for action recognition. In *Proceedings of the European Conference on Computer Vision*. 742–757.
- Thomas Schlömer and Oliver Deussen. 2011. Accurate spectral analysis of two-dimensional point sets. *J. Graph. GPU. Game Tools* 15, 3 (2011), 152–160.
- Thomas Schlömer, Daniel Heck, and Oliver Deussen. 2011. Farthest-point optimized point sets with maximized minimum distance. In *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics*. ACM, 135–142.
- Christian Schmalz, Pascal Gwosdek, and Joachim Weickert. 2012. Multi-class anisotropic electrostatic halftoning. *Comput. Graph. Forum* 31, 6 (2012), 1924–1935.
- Justin Solomon, Fernando De Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. 2015. Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Trans. Graph. (TOG)* 34, 4 (2015), 66.
- Robert A. Ulrichney. 1988. Dithering with blue noise. In *Proceedings of the IEEE* 76, 1, 56–79.
- Cédric Villani. 2008. *Optimal Transport: Old and New*. Vol. 338. Springer Science & Business Media.
- Muge Wang and Kevin J. Parker. 1999. Properties of combined blue noise patterns. In *Proceedings of the International Conference on Image Processing*. 328–332.
- Li-Yi Wei. 2008. Parallel Poisson disk sampling. *ACM Trans. Graph. (TOG)* 27, 3 (2008), 20.
- Li-Yi Wei. 2010. Multi-class blue noise sampling. *ACM Trans. Graph. (TOG)* 29, 4 (2010), 79:1–79:8.
- Li-Yi Wei and Rui Wang. 2011. Differential domain analysis for non-uniform sampling. *ACM Trans. Graph. (TOG)* 30, 4 (2011), 50:1–50:10.
- Kenric B. White, David Cline, and Parris K. Egbert. 2007. Poisson disk point sets by hierarchical dart throwing. In *Proceedings of the IEEE Symposium on Interactive Ray Tracing, 2007*. 129–132.
- Shi-Qing Xin, Bruno Lévy, Zhonggui Chen, Lei Chu, Yaohui Yu, Changhe Tu, and Wenping Wang. 2016. Centroidal power diagrams with capacity constraints: Computation, applications, and extension. *ACM Trans. Graph. (TOG)* 35, 6 (2016), 244:1–244:12.
- Yin Xu, Ruizhen Hu, Craig Gotsman, and Ligang Liu. 2012. Blue noise sampling of surfaces. *Comput. Graph.* 36, 4 (2012), 232–240.
- Yin Xu, Ligang Liu, Craig Gotsman, and Steven J. Gortler. 2011. Capacity-constrained Delaunay triangulation for point distributions. *Comput. Graph.* 35, 3 (2011), 510–516.
- Cem Yuksel. 2015. Sample elimination for generating poisson disk sample sets. *Comput. Graph. Forum* 34, 2 (May 2015), 25–32. DOI: <http://dx.doi.org/10.1111/cgf.12538>

Received October 2016; revised March 2017; accepted May 2017