DOI: 10.1111/cgf.12725

COMPUTER GRAPHICS *forum*
Volume 00 (2015), number 0 pp. 1–12

Projective Blue-Noise Sampling

Bernhard Reinert[1], Tobias Ritschel[1,2], Hans-Peter Seidel[1] and Iliyan Georgiev[3]

[1]MPI Informatik, Saarbrücken, Germany
{breinert, ritschel, hpseidel}@mpi-inf.mpg.de
[2]Saarland University, Saarbrücken, Germany
[3]Solid Angle, London, United Kingdom
iliyan@solidangle.com

Abstract

We propose projective blue-noise patterns that retain their blue-noise characteristics when undergoing one or multiple projections onto lower dimensional subspaces. These patterns are produced by extending existing methods, such as dart throwing and Lloyd relaxation, and have a range of applications. For numerical integration, our patterns often outperform state-of-the-art stochastic and low-discrepancy patterns, which have been specifically designed only for this purpose. For image reconstruction, our method outperforms traditional blue-noise sampling when the variation in the signal is concentrated along one dimension. Finally, we use our patterns to distribute primitives uniformly in 3D space such that their 2D projections retain a blue-noise distribution.

Keywords: sampling, blue-noise, Monte Carlo rendering, image reconstruction, primitive placement

ACM CCS: I.3.3 [Computer Graphics]: Picture/Image Generation—Antialiasing; I.4.1 [Image Processing and Computer Vision]: Digitization and Image Capture—Sampling

1. Introduction

Producing 'good' sampling patterns is an important task in many computer graphics applications, including simulation, rendering, image reconstruction and primitive placement. But what makes a good pattern depends on the application. For Monte Carlo (MC) rendering, low integration error is important [Coo86], [Shi91], which is usually achieved by stratified Latin hypercube (LH) or low-discrepancy point sets that have well-distributed low-dimensional projections. For digital half-toning [Uli87], stippling [KCODL06] and object placement [HHD03], blue-noise point sets with maximized minimum distance are preferred, as their structure closely resembles the photoreceptor arrangement in the eye retina [Yel83]. Such patterns are also desirable for image reconstruction [DW85] where low-discrepancy patterns can lead to spurious aliasing artefacts [Mit87]. At the same time, it has been shown that blue-noise patterns are not as competitive for numerical integration [Shi91]. All these applications call for point sets that are uniformly distributed in the sampling domain, though research in each area has focused on optimizing the distribution for its slightly different definition of uniformity. It has remained an open question whether there exist distributions that meet the requirements of a wide range of applications.

In this paper, we propose *projective blue-noise* point distributions, in an attempt to give a positive answer to the above question. A key property of these distributions is that they retain their blue-noise characteristics when undergoing one or multiple projections to lower dimensional subspaces (Figure 1a). We show how the classic dart throwing and Lloyd relaxation algorithms can be extended to produce such point sets, and demonstrate the usefulness of their projective blue-noise properties in various applications. For MC rendering (Figure 1b) and image reconstruction (Figure 1c), our patterns often outperform existing distributions for functions with variations concentrated along one dimension, where the resulting sampling quality is dominated by a projection of the pattern. Furthermore, while common blue-noise patterns are useful for placing primitives in 3D space, our patterns preserve the good visual distribution when the arrangement is viewed from different angles (Figure 1d).

The rest of this paper is organized as follows. After reviewing relevant prior work in the next section, in Section 3 we describe

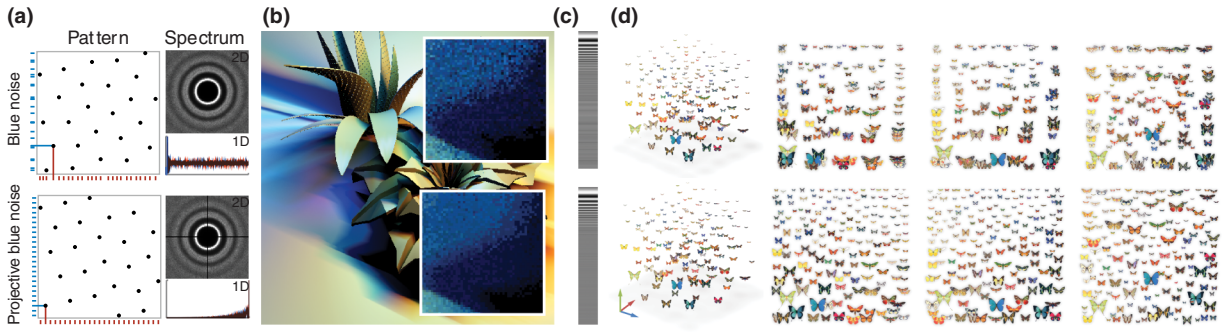

Figure 1: *Our projective blue-noise distributions (a, bottom) have blue-noise spectra in 2D as well as in their 1D projections, while classic 2D blue noise (a, top) has an almost white-noise spectrum in 1D. Applications include Monte Carlo rendering (b), reconstruction (c), as well as placement of primitives (d) such that they are well distributed both in 3D and when projected to 2D.*

the construction of projective blue-noise patterns and discuss some practical considerations. In Section 4, we analyse the properties of our point sets and the performance of our construction algorithms. In Section 5, we demonstrate the versatility of our method in practical applications, followed by final conclusions in Section 6.

2. Previous Work

Sampling is a key component in many computer graphics problems that arise in rendering, imaging, geometry processing and object distribution. Coined by Ulichney [Uli87], the term *blue noise* refers to the spectral properties of uniform and isotropic, yet structureless point distributions. Originally developed for the purposes of digital half-toning [Uli87], blue-noise distributions have been found useful in many other applications due to their resemblance of the photoreceptor arrangement in the eye retina [Yel83].

One way to generate a blue-noise point set is to insert the points one by one, maintaining a minimum interpoint distance along the way. The most common approach for doing this is dart throwing [Coo86], which ensures that points are tightly packed but no closer than a specified minimum distance, producing a so-called Poisson-disk distribution. This algorithm can be made progressive by adaptively shrinking the disk radius after a certain number of failed insertion attempts [MF92]. Alternatively, a new point can be inserted at the location farthest from the existing set [Mit91, ELPZ97].

Another approach to produce high-quality blue-noise distributions is to take an initial, e. g. random, point set and maximize the minimum interpoint distance using an iterative optimization scheme. The popular Lloyd relaxation algorithm [Llo82] is based on centroidal Voronoi tessellation (CVT). The quality of the resulting patterns has been recently improved by Balzer *et al.* [BSD09], and some further developments have addressed anisotropic sampling [LWSF10] and improving efficiency [dGBOD12, CYC*12].

Using uniform but irregular sample patterns can also reduce aliasing [DW85] and noise [Coo86] in MC rendering. Such applications have traditionally relied on *stratified* pseudo-random sampling, which places one sample in every stratum of the uniformly subdivided sampling domain [PH10]. One instance of this idea is *jittering*, where each sample on a regular grid is displaced randomly. Stratification can often increase the error convergence rate of the MC estimator over pure random sampling [Mit92]. LH, or N-rooks, sampling enforces stratification along each axis of a high-dimensional point set. Stratified and LH sampling have been combined to produce high-quality numerical integration patterns [CSW94, Ken13].

The quality of a pattern for numerical integration can be measured by its *discrepancy* [Shi91, Mit92]. Low-discrepancy point sets have seen wide adoption in physically based rendering, as they are relatively simple to implement and possess excellent stratification and LH properties [KK02, PH10, KPR12]. However, most low-discrepancy sampling methods achieve good stratification only for a restricted number of points (e. g. powers of two), depending on the construction method [KPR12]. Our approach does not have this limitation and performs on par with or better than such methods.

Even though blue-noise point sets have very uniform distribution and lack regularity, their application in image synthesis has so far been mostly restricted to image anti-aliasing [Mit87, PH10]. Reports on their performance for estimating illumination integrals have been controversial [Shi91, SHD11, MBR*13]. We hypothesize that the main reason for their suboptimal performance is the poor uniformity in their low-dimensional projections, and aim to produce point sets with both blue-noise and LH properties. Research in numerical integration has shown that the quality of a pattern can be improved by 'latinizing' either the initial values for Lloyd relaxation or the final result [RBGP06, SGB07]. However, to our knowledge no attempt has been made to achieve both LH and blue-noise properties simultaneously.

Some of the aforementioned works have demonstrated the utility of blue-noise sampling for procedural primitive placement. This application also benefits from the real-time performance of tile-based sampling [ODJ04, KCODL06]. Computational placement of extended primitives in 2D [HHD03, RRS13] has applications in automated generation of layouts in print, on screens and for fabrication.

While producing images using projections, e. g. shadows [MP09], has been addressed in computer graphics, no prior work has considered the *spectral* properties of projections of primitive layouts. This

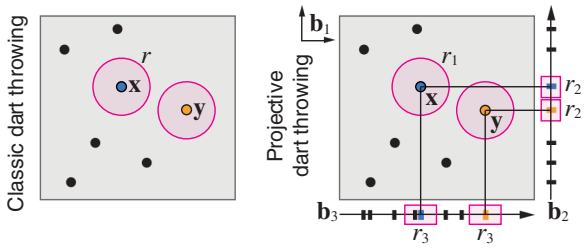

Figure 2: *Classic dart throwing (left) accepts both the orange and the blue candidate points, as no existing point in the set is within the rejection radius r (pink circle). Projective dart throwing (right) accepts the orange candidate, but not the blue one, as its projection onto* $\mathbf{b}_3$*, with radius* r_3 *(pink box), conflicts with an existing point.*

becomes important when placing (fabricated) objects in 3D space, i. e. in a physical exhibition or a collaborative virtual environment. Typically used patterns do not produce layouts with blue-noise distribution when observed from different viewpoints. Our approach can reduce clutter and occlusion by optimizing both the spatial and the projected arrangement.

3. Our Approach

The basic idea of projective blue noise is to extend existing sampling methods to not only operate in the full d-dimensional sample space, but also in multiple lower dimensional projective subspaces simultaneously. While classic dart throwing and Lloyd relaxation test candidates or move points only in the full-dimensional space, our extensions additionally check candidates, respectively, move points, in the projection subspaces. We specify all spaces via a set of m projection vectors

$$\mathcal{B} = \left\{\mathbf{b}_j \in \{0,1\}^d\right\}_{j=1}^{m}. \tag{1}$$

The vectors $\mathbf{b}_j$ can be used with the Hadamard product (i. e. the element-wise vector multiplication) to project points and vectors onto the (sub)spaces specified by those vectors. Classic, non-projective blue noise is the special case with $\mathcal{B} = \{\{1\}^d\}$. In our projective extension, for 2D point sets we use $\mathcal{B} = \{(1,1),(1,0),(0,1)\}$.

In the following subsections, we extend the dart throwing and Lloyd relaxation algorithms to projective blue-noise sampling. For each method, we first review its classic, non-projective variant, before presenting our projective extension.

3.1. Dart throwing

Classic. Dart throwing (Figure 2, left) starts with an empty point set and iteratively generates random candidate points $\mathbf{x}$ that are added to the set only if their distance to every other point $\mathbf{x}_i$ is larger than a certain threshold r, called Poisson-disk radius [Coo86]:

$$\min_{i=1,\dots,n'} \|\mathbf{x} - \mathbf{x}_i\| > r, \tag{2}$$

where $n' < n$ is the number of the already accepted points. For tiled patterns, the distance is computed on a toroidally wrapped domain. In its most basic form, the algorithm terminates if no new points can be added after a certain number of successive failed attempts. Alternatively, instead of terminating, the radius can be shrunk by a constant factor, which makes the sampling method progressive [MF92]. Our implementation operates on a toroidal domain and also incorporates this shrinkage.

Projective. Our dart throwing extension (Figure 2, right) accepts a candidate point $\mathbf{x}$ only if its distance to every other point $\mathbf{x}_i$ in the full d-dimensional space and in every projection space is larger than a certain threshold:

$$\min_{i=1,\dots,n'} \left\|\mathbf{b}_j \circ (\mathbf{x} - \mathbf{x}_i)\right\| > r_j \quad \forall j \in 1,\dots,m, \tag{3}$$

where r_j is the desired radius (i. e. minimum distance) in the jth space, and $\circ$ is the aforementioned Hadamard product.

Radii. Since the distances between points are smaller in lower dimensional subspaces, the radii for these spaces should be smaller than the ones for higher dimensional spaces. We derive the radius for a space from the radius of the tightest known lattice sphere packing in the corresponding dimension. For 1D through 4D, these maximum radii are given, respectively, by [LD06, KZ77]

$$r_1^{\max} = \frac{1}{2n}, \quad r_2^{\max} = \sqrt{\frac{1}{2\sqrt{3}n}}, \quad r_3^{\max} = \sqrt[3]{\frac{1}{4\sqrt{2}n}}, \quad r_4^{\max} = \sqrt[4]{\frac{1}{8n}}.$$

For higher dimensions d, the maximum radius is found by solving $\mathrm{V}_d = \eta_d / n$ for the radius $r_d^{\max}$ of a d-dimensional sphere, where V_d is the sphere volume and η_d is the best lattice disk packing density.

Given the maximum packing radii, we compute the Poisson-disk radii as

$$r_j = r \cdot \frac{r_{d_j}^{\max}}{r_d^{\max}}, \tag{4}$$

where $d_j = \|\mathbf{b}_j\|_1$ is the dimension of space j and d is the dimension of the full space. The algorithm is now controlled via a single parameter $r \in [0,1]$, which we set to $r = 0.15$.

3.2. Lloyd relaxation

Classic The Lloyd optimization algorithm (Figure 3, left) constructs a point arrangement that is a CVT. In a CVT, each point, or site, $\mathbf{x}_i$ is also the centre of its associated Voronoi cell—the subset of the domain that is closer to $\mathbf{x}_i$ than to any other site. To obtain such a set $X = \{\mathbf{x}_1,\dots,\mathbf{x}_n\}$, Lloyd relaxation minimizes the cost

$$c(X) = \sum_{i=1}^{n} \int_{\Omega_i} \|\mathbf{x}_i - \mathbf{x}\|^2 \mathrm{d}\mathbf{x}, \tag{5}$$

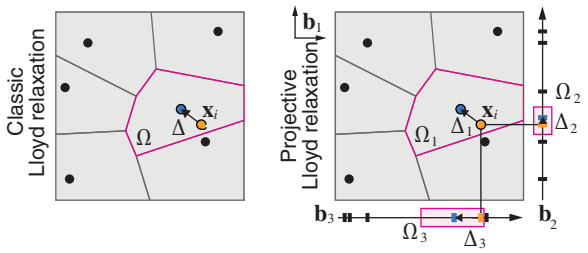

Figure 3: *Classic Lloyd relaxation (left) moves each point* $\mathbf{x}_i$ *(orange) to the centre (blue) of its Voronoi cell (pink). Our projective extension (right) tries to move each point* $\mathbf{x}_i$ *to the centre of its Voronoi cell in the full space* $\mathbf{b}_1$ *as well as in the two subspaces specified by* $\mathbf{b}_2$ *and* $\mathbf{b}_3$.

where Ω_i is the ith cell in the Voronoi tessellation of X. Conceptually, this cost measures how far the sites are from the centre of mass of their Voronoi cells. As with dart throwing, for tiled patterns the distances are computed on a toroidally wrapped domain.

The relaxation algorithm starts with a random set of sites X which is refined iteratively in three steps. First, a Voronoi tessellation of X is built, mapping every location in the domain to its closest site. Secondly, the centroid of every Voronoi cell is computed by averaging the locations in the cell. Finally, every site is moved to the centroid of its associated cell. Practical implementations often discretize the domain into a finite set of locations, turning the integral in Eq. (5) into a sum.

Projective. In our case, we want to distribute the sites uniformly not only in the full-dimensional space, but also in multiple lower dimensional projection subspaces (Figure 3, right). Similarly to our dart throwing extension, we specify all spaces via a set of m projection vectors $\mathbf{b}_j$. We build m Voronoi tessellations, one for each space, and aim to optimize the sites such that they coincide with centroids of their corresponding Voronoi cells in all tessellations. Note that for any site, its associated cells are different in the different spaces. The cost of each site is computed by summing m weighted norms, and the total cost reads

$$c_{\mathrm{p}}(X) = \sum_{j=1}^{m} w_j \sum_{i=1}^{n} \int_{\Omega_{i,j}} \left\| \mathbf{b}_j \circ \mathbf{x}_i - \mathbf{x} \right\|^2 \mathrm{d}\mathbf{x}, \tag{6}$$

where $\circ$ is the Hadamard product, $\{w_j\}_{j=1}^m$ is a set of scalar projection weights that sum up to one, and $\Omega_{i,j}$ is the Voronoi cell of the ith site after projection onto the jth space. Note that the dimension of the points $\mathbf{x}$ above depends on j.

Our relaxation scheme seeks to minimize Eq. (6). We start with the same initial point set as the classic Lloyd method, but we perform the optimization steps in m spaces simultaneously as follows. For each site $\mathbf{x}_i$, we compute m correction vectors $\Delta_{i,j}$, one from the Voronoi tessellation in each space. Each vector $\Delta_{i,j}$ would move $\mathbf{x}_i$ to the centre of Voronoi cell $\Omega_{i,j}$. These vectors are generally different for the different projections j, and each individual site can be moved to exactly fulfil only one constraint locally. We instead try to partially fulfil all constraints by applying all correction vectors $\Delta_{i,j}$ to site $\mathbf{x}_i$, each scaled by a corresponding weight w_j. After moving all sites, the m Voronoi tessellations are recomputed and the process is iterated.

Note that the projective correction vectors are heuristically chosen and not proven to be optimal as the non-projective vectors are, but work well in practice as shown by our analysis.

Weights. Ideally, we want all spaces to have equal importance in the total cost in Eq. (6). However, since distances between points in lower dimensional spaces are shorter, the relative contribution of such spaces is smaller than that of higher dimensional spaces. We equalize all contributions by making each weight w_j inversely proportional to the tightest sphere packing radius in dimension d_j:

$$w_j = \frac{1/r_{d_j}^{\max}}{\sum_{k=1}^{m} 1/r_{d_k}^{\max}}. \tag{7}$$

Additional optimizations. Due to the increased number of constraints, the straightforward implementation of the above scheme may converge much slower than the classic Lloyd relaxation algorithm. We propose two enhancements to improve both the speed and the quality of the resulting patterns.

First, in early iterations, the different correction vectors contradict heavily, making the process susceptible to local minima and slow convergence. To remedy this, the weights w_j for the lower dimensional spaces are faded in linearly from 0 for 50 iterations. This is done for each dimension successively, i. e. first the weights of the $d-1$-dimensional subspaces are faded in, then the $d-2$-dimensional ones, etc.

Secondly, the convergence speed can be further increased by exploiting the fact that in 1D the best point arrangement that maximizes the mutual minimum distance is the regular distribution [RAMN12], which is the global minimum of the Lloyd cost in 1D (Eq. 5). Thus, to satisfy a single 1D projection, the pattern can simply be 'snapped' to a regular grid along the corresponding axis [SGB07]. So instead of building 1D Voronoi tessellations, we directly compute the correction vectors as the differences between the regular grid $\{(0.5+i)/n\}_{i=0}^{n-1}$ and the sorted point coordinates along each axis. Note that this closed-form solution works only for 1D projections, and minimizing the cost in multiple dimensions simultaneously still requires iterative optimization.

4. Analysis

In this section, we compare the quality of our patterns to existing methods in terms of their spectral and projective properties, Poisson-disk radii and discrepancy. We also compare our method to latinization [SGB07] and analyse the convergence and the computational performance of our projective Lloyd relaxation.

4.1. Projective analysis

We begin with an analysis of the spectral and spatial properties of our projective patterns. We follow the recommendations of Schlömer

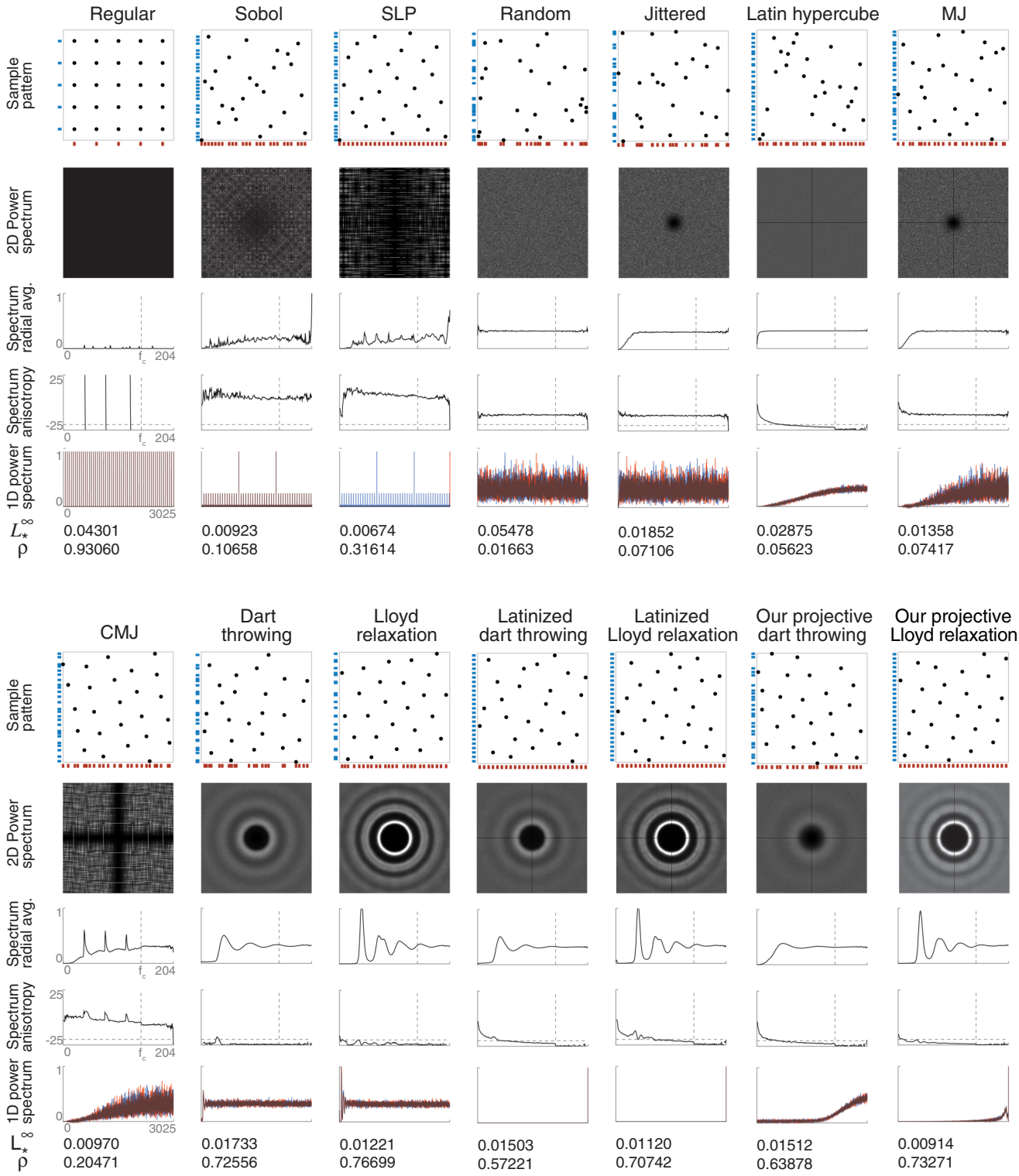

Figure 4: *Spectral analysis of 14 different 2D point sets and their 1D axis projections (shown as red and blue bars and graphs, respectively), discussed in Section 4. Note that the horizontal (frequency) scale is different for the 1D power spectrum plots and the radial average and anisotropy plots. The last rows report the star discrepancy L_{*}^{∞} and Poisson-disk radius ρ of each point set.*

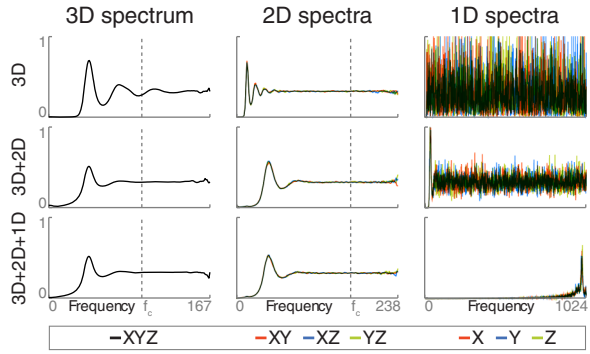

Figure 5: *The 3D, 2D and 1D power spectra (columns) of 1024-sample patterns, averaged from 10 periodograms, produced by our projective Lloyd relaxation with different projections enabled (rows). The spectra of the different same-dimensional projections are colour-coded.*

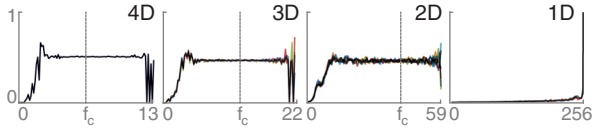

Figure 6: *The power spectra of a 256-sample 4D projective blue-noise point distribution, averaged from 10 periodograms. Each plot shows the radial averages for all subspaces of the same dimensionality: 1 in 4D, 4 in 3D, 6 in 2D and 4 in 1D.*

et al. [SHD11] on reporting spectral data for point sets, including the critical frequency f_c and anisotropy reference levels in Figures 1, 4, 5 and 6. For the 1D, 3D and 4D power spectrum plots, we generalize the relevant frequency ranges using the respective $r_d^{\max}$ (see Section 3.1).

2D analysis. In Figure 4, we compare various 2D patterns in terms of their Fourier power spectra as well as their star discrepancy and Poisson-disk radii. The patterns we consider are: (1) regular, (2) Sobol, (3) scrambled Larcher-Pillichshammer (SLP) [KK02], (4) uniform random, (5) jittered, (6) LH, (7) multi-jittered (MJ) [CSW94], (8) correlated multi-jittered (CMJ) [Ken13], (9) dart throwing, (10) Lloyd relaxation, (11) latinized dart throwing [SGB07], (12) latinized Lloyd relaxation [SGB07], (13) our projective dart throwing and (14) our projective Lloyd relaxation. All plotted patterns consist of 25 samples. The spectral power plots have been computed as the average of the periodograms of 100 random instances of each pattern with 3025 samples. The star discrepancy L_*^∞ and Poisson-disk radii ρ (i. e. the normalized global minimum intersample distance [LD08]) have been computed for sample patterns of size 529.

A characteristic feature of 2D blue-noise distributions is their isotropic Fourier power spectrum with a black disk in the centre indicating the absence of low-frequency content, surrounded by an energy-peak ring around the principal frequency [LD08]. Such distributions are produced by maximizing the minimum distance between the points in the set. As discussed in Section 3.2, in 1D the regular distribution achieves the maximum point separation, and its Fourier power spectrum is zero except at frequencies that are multiples of n (the number of points). As a consequence, in Figure 4 patterns with good 1D axis projections, such as LH, MJ as well as the latinized and our projective patterns, have black crosses in their 2D power spectra. This is because the interesting power spectrum features of such distributions occur at different scales in 1D and 2D. For the sake of visual clarity, we clamp the 2D spectrum plots to the frequency range $[0; 204]$, and show the 1D-projection slices of these spectra as separate plots in the range $[0; 3025]$. Note also that the black crosses in the 2D spectra cause a slight artificial increase in 2D anisotropy.

The Sobol and SLP low-discrepancy patterns achieve regular 1D axis projections for sample counts that are powers of two. For other sample counts however, gaps remain along at least one axis. In our 25-sample example in Figure 4, only the x-axis of the SLP pattern has a uniform distribution. Furthermore, both methods fail to produce blue-noise 2D spectra.

Uniform random, regular and jittered sampling all fail to produce high-quality 2D blue-noise distribution and 1D projections. The regular 2D pattern has very poor projection distributions with multiple points sharing the same coordinates on each axis, causing spikes in the 1D power spectra. LH sampling has good projection properties but poor 2D uniformity.

MJ sampling [CSW94], as a combination of jittered and LH sampling, produces a 2D power spectrum that shares the features of both methods. MJ patterns have acceptable projections, but like other jittered patterns do not guarantee a minimal intersample distance, which is reflected in the smooth low-frequency ramps in both the 2D and the projected 1D power spectra. CMJ [Ken13] results in identical 1D spectra but a much less uniform 2D spectrum.

Classic Lloyd relaxation produces excellent blue-noise distributions but only in 2D. Latinizing the pattern improves the 1D projections but at the cost of increasing the 2D anisotropy and decreasing the Poisson-disk radius. Our projective Lloyd distribution combines a good 2D spectrum with well-distributed 1D projections, and has a larger Poisson-disk radius than the latinized variant. Improving the 1D projections of a blue-noise pattern also decreases its discrepancy [SGB07], where our projective method once again comes on top of latinization.

Finally, dart throwing produces inferior blue-noise distributions compared to Lloyd relaxation, as confirmed by both the 2D and the projected 1D spectral plots. This is not unexpected, as this method is highly sensitive to the order in which the samples are inserted. The additional constraints introduced by our projective extension further reduce the probability of successfully inserting a candidate point. As a result, the 2D blue-noise spectrum of projective dart throwing suffers more from the imposed constraints than that of Lloyd relaxation, and some low-frequency components appear—the dark circle in the centre of the spectrum is grey, not black. On the other hand, latinized dart throwing produces good projective and non-projective spectra but also higher anisotropy.

3D analysis. In Figure 5, we compare the power spectra of classic 3D blue-noise patterns to the projective distributions produced by

our extended Lloyd relaxation. We see that classic, non-projective patterns (top row) have inferior 2D spectral properties and almost white-noise 1D distributions. Our approach (middle and bottom rows) retains good quality in all specified projection subspaces, although this comes at the cost of slight quality degradation in the 3D spectrum.

4D analysis. Finally, in Figure 6 we plot the power spectra of the 4D patterns produced by our extended Lloyd relaxation with projective blue-noise properties imposed on all lower dimensional subspaces. The characteristic blue-noise spectrum shape in all projections demonstrates that our method generalizes to higher dimensions. Note the different frequency scale of each plot, indicating a quality improvement in the corresponding dimension over classic 4D blue noise, as also seen in Figure 5. However, imposing this many constraints leads to a more noticeable overall deterioration in blue-noise quality compared to the 3D case. The noise in the plots is due to the relatively low number of periodograms averaged (10).

4.2. Comparison to latinization

Having analysed the spectral properties of our projective Lloyd relaxation and the latinization method of Saka *et al.* [SGB07] in Section 4.1, we now conduct a closer comparison between these two approaches. On a high level, our notion is a non-greedy, high-dimensional generalization of latinization: Instead of first creating a blue-noise pattern and latinizing it as a post-process, we interleave these two steps. Note that while our method handles arbitrary dimensions, latinization only considers 2D patterns and their 1D axis projections. We look the average projective Poisson-disk radius with respext to vector $\mathbf{b}$:

$$\tilde{\rho}_{\mathbf{b}} = \frac{1}{n} \sum_{\mathbf{x} \in X} \min_{\mathbf{y} \in X} \|\mathbf{b} \circ (\mathbf{x} - \mathbf{y})\| / d_{\mathbf{b}}^{\max}, \tag{8}$$

which is the projective generalization of the average Poisson-disk radius measure [SHD11]. The minimal intersample distance is normalized by the optimal packing distance $d_{\mathbf{b}}^{\max}$ in subspace $\mathbf{b}$, i. e. $d_{\mathbf{b}}^{\max} = 2r_{\|\mathbf{b}\|_1}^{\max}$. In the following, we will use $\tilde{\rho}$, $\tilde{\rho}_{\mathbf{x}}$ and $\tilde{\rho}_{\mathbf{y}}$ to denote the average radii in 2D and in the x- and y-axis projections, respectively, corresponding to projection the vectors (1, 1), (1, 0) and (0, 1).

For patterns with 512 samples, latinized dart throwing and our projective variant achieve $(\tilde{\rho}, \tilde{\rho}_{\mathbf{x}}, \tilde{\rho}_{\mathbf{y}}) = (0.801, 1, 1)$ and $(\tilde{\rho}, \tilde{\rho}_{\mathbf{x}}, \tilde{\rho}_{\mathbf{y}}) = (0.806, 0.917, 0.921)$, respectively. Latinized Lloyd relaxation produces $(\tilde{\rho}, \tilde{\rho}_{\mathbf{x}}, \tilde{\rho}_{\mathbf{y}}) = (0.873, 1, 1)$, whereas our projective variant gives the best balance with $(\tilde{\rho}, \tilde{\rho}_{\mathbf{x}}, \tilde{\rho}_{\mathbf{y}}) = (0.909, 0.998, 0.999)$. In summary, latinization produces perfect 1D stratification, but at the cost of decreasing the uniformity in 2D. Our approach optimizes for both measures simultaneously.

The above measures indicate that in 2D latinized dart throwing outperforms our projective variant. However, latinization has a negative effect on the progressiveness of the algorithm, which is one of its strengths. We analyse this in Figure 7 by comparing the average Poisson-disk radii for classical, latinized and our projective dart throwing and the Sobol low-discrepancy sequence [Sob94]. We

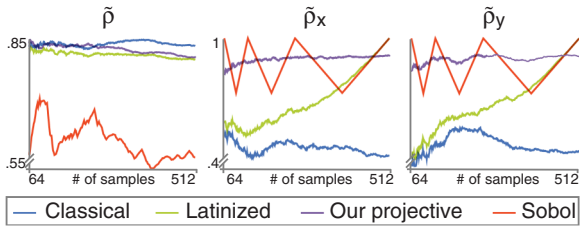

Figure 7: *Average Poisson-disk radii in 2D and along the x- and y-axis projections (left to right) as functions of increasing sample subset size for three dart throwing variants and the Sobol sequence. Please see Section 4.2 for details.*

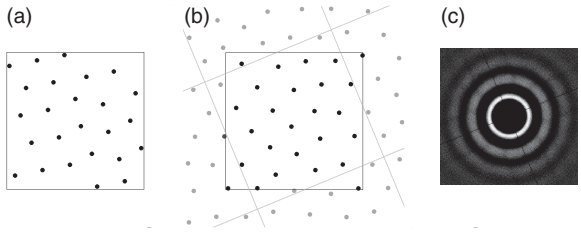

Figure 8: *Projection axis rotation. (a) Original pattern. (b) Tiled and rotated pattern. (c) Resulting power spectrum.*

plot the radii as functions of the first n points in 512-sample sets, where the latinized pattern is computed from the classical one after generating all samples. The plots reveal that our projective dart throwing has the most consistent progressive performance in 1D and 2D, while the latinized variant has good properties only for sample counts close to the maximum. The classical dart throwing performs well in 2D, but its 1D projections are consistently poor. In contrast, the Sobol pattern is perfectly latinized for power-of-two sample counts, but its 2D radius is significantly worse than those of the dart throwing patterns.

4.3. Rotation

Our approach supports rotating the canonical coordinate axes, as long as they remain orthogonal. Since our methods operate on a toroidal domain, this can be trivially achieved by optimizing the pattern for the canonical axes (Figure 8a) and then rotating it by the desired amount (Figure 8b). The black cross in the resulting power spectrum is oriented according to the rotation angle (Figure 8c). The same approach can also be used in higher dimensions, as the patterns tile in all directions.

4.4. Sample warping

MC rendering and primitive placement often require warping samples according to a specified importance function. In Figure 9, we compare the warping quality of our projective Lloyd patterns against other patterns on a thin 2D curve, which can represent, e.g. an environment light source for rendering or a path on a plane for primitive placement. To produce the warped patterns, we first create samples as described before, i. e. using a uniform importance. We then warp

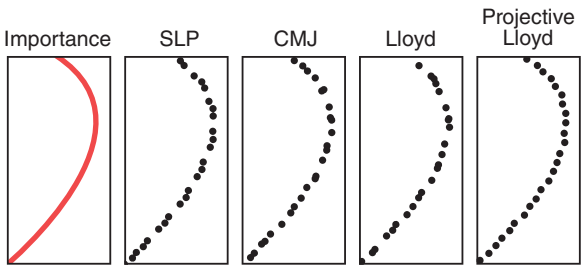

Figure 9: *Importance-driven warping of different 25-sample 2D patterns. Our projective Lloyd pattern produces the best distribution, thanks to its good projections along both axes.*

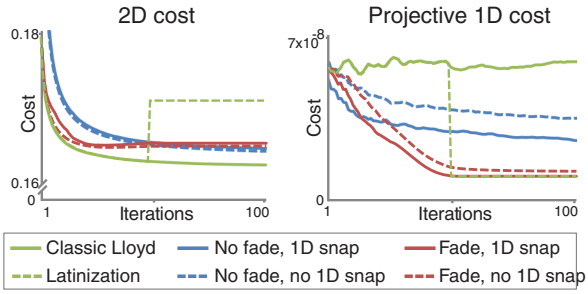

Figure 10: *2D and projective 1D cost plots at different iterations of classic and latinized Lloyd relaxation as well as four different variants of our projective extension for a pattern with 2048 samples (see Section 4.5). Different weight fading strategies are shown in different colours, while solid and dashed lines denote the use of 1D grid snapping (Section 3.2).*

these samples to follow a new importance function (Figure 9, left) using the common approach of tabulating conditional and marginal distributions along the x- and y-axis, respectively [Dev86, p. 555]. Our pattern achieves the most uniform warped distribution, which in rendering translates to a low integration error. This is due to its well-distributed projections along both axes, a property that no other pattern in the comparison has.

4.5. Lloyd convergence

We now analyse the convergence of different variants of our projective Lloyd optimization on a 2D point set X. In Figure 10, we plot the classic 2D cost $c(X)$ (Eq. (5)) and the projective 1D part $c_p(X) - w_1 \cdot c(X)$ of the generalized cost (Eq. (6)), where w_1 is the full (2D) space weight, over an increasing number of iterations. We consider five variants of our algorithm: classic non-projective as well as variants with and without the weight fading and 1D grid snapping optimizations from Section 3.2. A sixth graph shows the effect of latinizing the pattern after 50 optimization iterations.

We see that all non-latinized variants minimize the classic 2D cost, though with slightly different speeds. Our projective variants achieve a final 2D cost only 0.76% worse than that of the classical method (note the small $[0.16; 0.18]$ vertical range in the left plot), while also minimizing the 1D costs. On the other hand, latinization brings excellent projective properties but at the expense of increasing the total cost.

Unsurprisingly, the projective 1D cost is not minimized by the classic Lloyd relaxation. The weight fading increases the projective cost initially, but ultimately achieves a substantially faster convergence. The 1D grid snapping adds a further constant runtime improvement.

4.6. Performance

The GPU implementation of our projective Lloyd relaxation performs slower than classic Lloyd relaxation by a factor slightly smaller than m (the number of projection spaces). While the performance of the 1D relaxation is improved by the grid snapping optimization, its computational cost is insignificant compared to the higher dimensional Voronoi tessellations. Our naïve implementation of projective dart throwing is about $3\times$ slower than the classic method.

5. Applications

In this section, we demonstrate the utility of our projective blue-noise patterns in rendering, image reconstruction and primitive placement—applications that have traditionally relied on specialized distributions.

5.1. Rendering

In Figure 11, we compare several patterns for MC rendering of direct illumination from three different types of extended light sources. The light sources used to produce the renderings are (a) an anisotropic area light with a colour gradient, 21 samples per pixel (spp); (b) an importance-sampled environment light (see inset in Figure 11b, Section 4.4), 25 spp and (c) a square area light, 25 spp. We decorrelate the pixel estimators for all but the SLP and CMJ patterns via Cranley-Patterson rotation [CP76]. We use (1) Hammersley [KK02], (2) CMJ [Ken13], (3) SLP [KK02], (4) classic Lloyd relaxation (which was slightly better than dart throwing) and (5) our projective Lloyd relaxation (again, slightly better than our projective dart throwing) to sample the rectangluar domains of the area and environment light sources.

The visual and numerical results in Figure 11 indicate that our projective blue-noise patterns outperform all others on thin area lights. SLP is the closest competitor on all scenes and even slightly outperforms our patterns on the environment map scene. However, the pattern exhibits pixel correlation which is clearly visible in both the first and the second scenes, as also observed by Kensler [Ken13]. The environment map in the second scene features thin curved lights that are importance sampled as described in Section 4.4. In Figure 11(c), we see that the blue-noise patterns perform slightly better than the classical optimal methods on a square area light source.

To analyse the effect of light source anisotropy, in Figure 12 top-left we plot the Root Mean Square (RMS) reference error of three patterns on a simple scene for varying numbers of samples. Classic Lloyd relaxation (green) and SLP (blue) exhibit strong variation in quality. For power-of-two sample counts, SLP is on par with our projective Lloyd pattern (red) which otherwise outperforms consis-

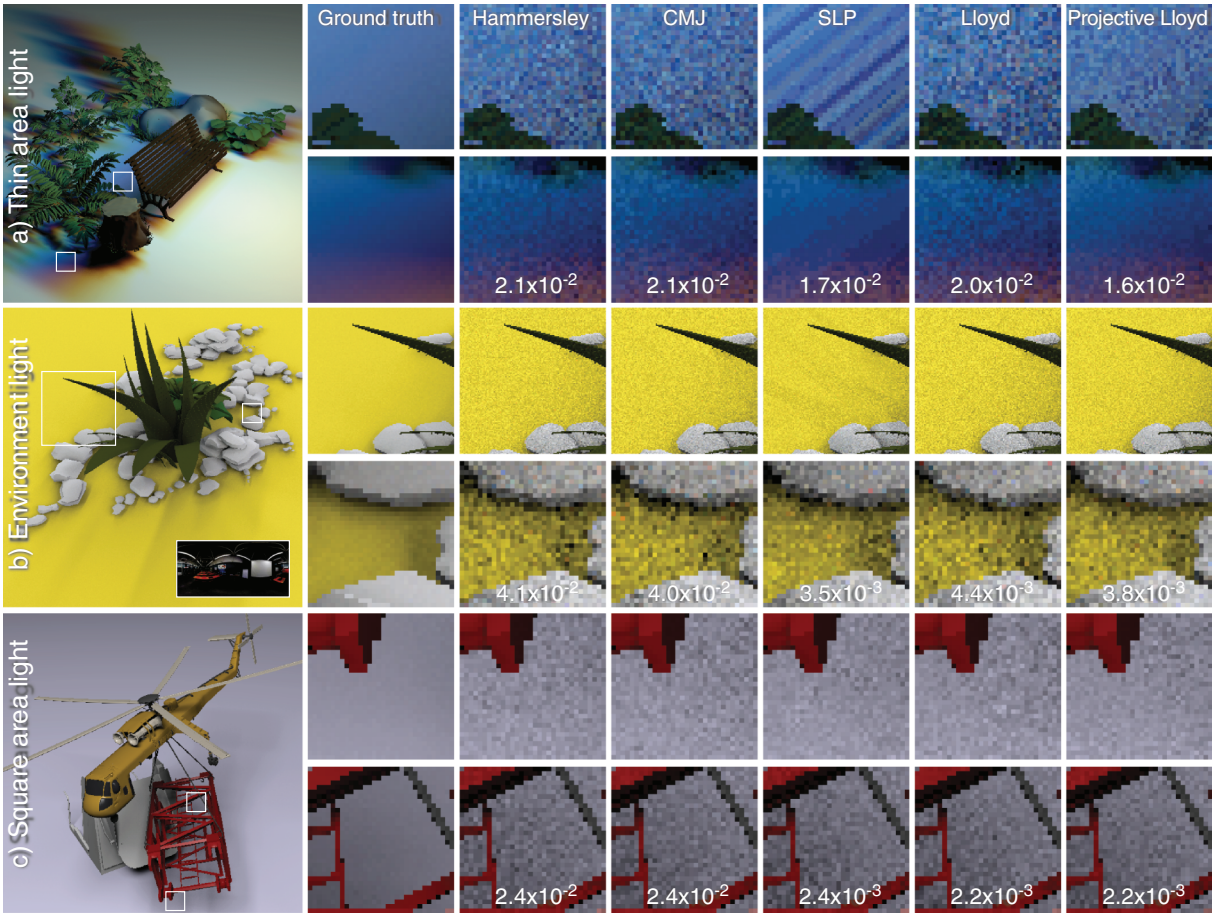

Figure 11: *Rendering comparison of five sample patterns on three scenes with different light sources, discussed in Section 5.1.*

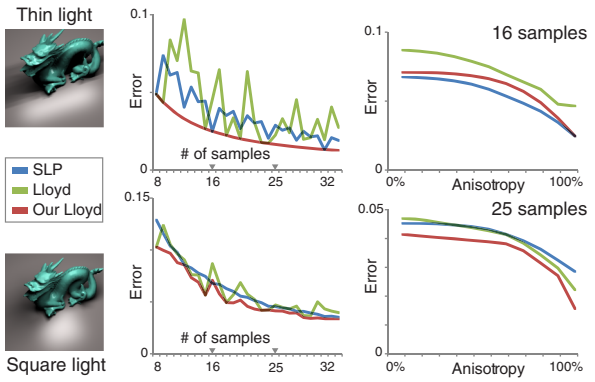

Figure 12: *RMS error plots of the renderings of a scene using three different sample patterns. We plot the error as a function of sample count (left) and area light source anisotropy (right).*

tently better. We speculate that classic Lloyd spuriously achieves the same quality as ours when it randomly produces patterns with good 1D projections. For a square area light source (Figure 12, bottom-left), the differences are smaller, though our approach again consistently outperforms classic Lloyd relaxation. Finally, in Figure 12 right we plot the RMS error as a function of the light source anisotropy.

5.2. Image reconstruction

In Figure 13, we compare the performance of classic Lloyd relaxation and our projective extension for reconstructing 2D images. We test a 2D zone plate function with a 4-pixel-wide Lanczos filter (512×512 pixels = 262 144 samples) as well as a modified anisotropic variant (512×16 pixels = 16 384 samples). On the isotropic zone plate (left), the two patterns perform similarly. However, our pattern significantly outperforms classic blue noise on the anisotropic variant (right), demonstrating the importance of having well-distributed 1D projections.

5.3. Primitive placement

We finally demonstrate the utility of our three-dimensional projective blue-noise patterns for primitive placement. Our approach enables arranging objects in 3D in a way that is artistically pleasant [HHD03], has semantic structure along its axes [RRS13] and fills multiple 2D projection subspaces uniformly. Figure 14 shows one such arrangement where primitives are sorted by size, brightness and orientation (direction of the primitive's first principal

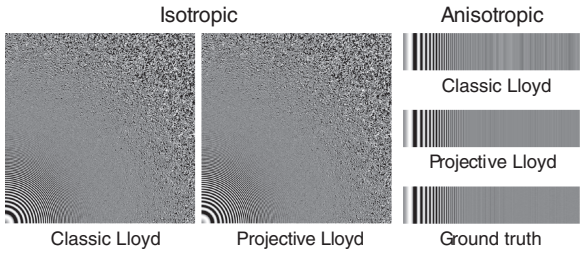

Figure 13: *Reconstructions of an isotropic (left) and an anisotropic (right) 2D zone plate functions using a classic blue-noise pattern and our projective blue-noise pattern.*

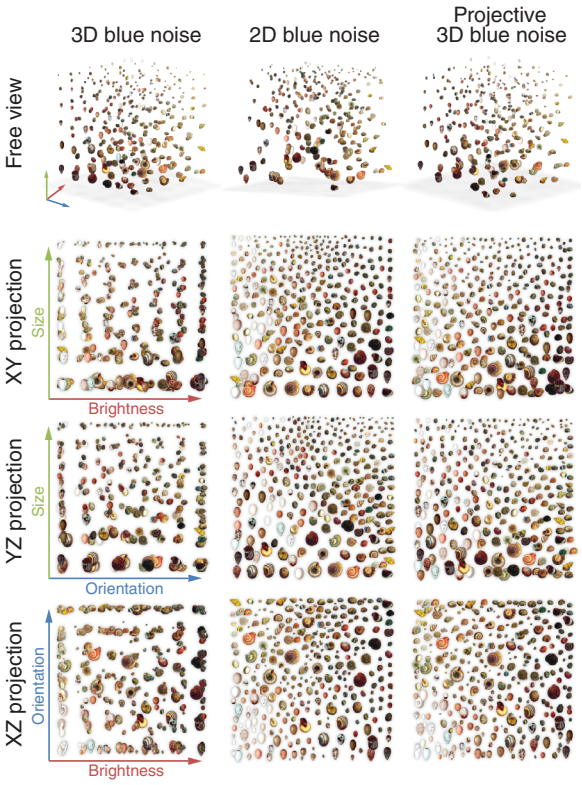

Figure 14: *Using 3D, 2D and our projective 3D blue-noise patterns for distributing primitives ordered by brightness, size and orientation along the x-, y- and z-axis, respectively.*

component) along the x-, y- and z-axis, respectively. The distances in the projective Lloyd cost (Eq. (6)) have been computed on a non-toroidal domain by taking the spatial extent of the objects into account [RRS13]. When looking at the 2D projection along an axis, the distribution remains uniform and shows one aspect of the data, e. g. size and brightness when looking down the z-axis.

6. Conclusion

We proposed a simple extension to blue-noise sampling that produces patterns with good spectral properties when projected onto lower dimensional subspaces. Our extension is easy to implement and likely orthogonal to many ways of generating patterns, as we demonstrated for two popular methods—dart throwing and Lloyd relaxation. For MC rendering, we showed that the resulting patterns perform better than classic blue-noise patterns and on par with or better than low-discrepancy patterns which have been specifically designed for the purpose of numerical integration. For primitive placement, we showed arrangements with uniform distribution in both 3D space and multiple 2D viewing projections. We believe our approach is a step towards a single universal multi-dimensional pattern with a wide range of applications.

Compared to low-discrepancy sequences, our patterns are much more costly to construct. Compared to the cost of common blue-noise sampling approaches, however, the overhead is small and the produced patterns are of almost the same quality in the full-dimensional space. For numerical integration, we have found our projective 2D Lloyd relaxation to perform consistently better than most other patterns in all our experiments. Nevertheless, our patterns can be slightly outperformed by low-discrepancy sequences that achieve perfect 1D stratification for certain sample counts. We have only demonstrated our method for up to four dimensions; however our results indicate that it generalizes to higher dimensions, which are required in full global illumination rendering.

Our implementation is currently limited to rectangular domains and allows consequently to (optionally rotated) axis-aligned projections, although this is not a limitation of our theoretical formulation. Extensions to non-orthogonal and non-linear projections is an interesting avenue for further research, which requires a more in-depth analysis of how such projections and the toroidal tiling influence each other. Other pertinent future work includes the extension of our and other sampling algorithms, e.g. capacity-constrained Lloyd relaxation [BSD09] or farthest point optimization [ELPZ97, SHD11], to projective blue noise and non-linear projections.

References

[BSD09] Balzer M., Schlömer T., Deussen O.: Capacity-constrained point distributions: A variant of Lloyd's method. *ACM Transactions on Graphics 28*, 3 (2009), 86:1–86:8.

[Coo86] Cook R. L.: Stochastic sampling in computer graphics. *ACM Transactions on Graphics 5*, 1 (1986), 51–72.

[CP76] Cranley R., Patterson T.: Randomization of number theoretic methods for multiple integration. *SIAM Journal of Numerical Analysis 13*, 6 (1976), 904–914.

[CSW94] Chiu K., Shirley P., Wang C.: Multi-jittered sampling. In *Graphics Gems IV* (San Diego, CA, USA, 1994), Academic Press, Professional, Inc., pp. 370–374.

[CYC*12] Chen Z., Yuan Z., Choi Y.-K., Liu L., Wang W.: Variational blue noise sampling. *IEEE Transactions on Visualization and Computer Graphics 18*, 10 (2012), 1784–1796.

[Dev86] Devroye L.: *Non-Uniform Random Variate Generation*. Springer-Verlag, New York, 1986.

[dGBOD12] DE GOES F., BREEDEN K., OSTROMOUKHOV V., DESBRUN M.: Blue noise through optimal transport. *ACM Transactions on Graphics 31*, 6 (2012), 171:1–171:11.

[DW85] DIPPÉ M. A., WOLD E. H.: Antialiasing through stochastic sampling. *ACM SIGGRAPH Computer Graphics 19*, 3 (1985), 69–78.

[ELPZ97] ELDAR Y., LINDENBAUM M., PORAT M., ZEEVI Y. Y.: The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing 6*, 9 (1997), 1305–1315.

[HHD03] HILLER S., HELLWIG H., DEUSSEN O.: Beyond stippling—Methods for distributing objects in the plane. In *Proceedings of Eurographics* (Granada, 2003), vol. 22, pp. 515–522.

[KCODL06] KOPF J., COHEN-OR D., DEUSSEN O., LISCHINSKI D.: Recursive Wang tiles for real-time blue noise (New York, NY, USA, 2006), *SIGGRAPH '06, ACM 25*, 509–518.

[Ken13] KENSLER A.: *Correlated multi-jittered sampling*. Pixar Technical Memo 13-01, 2013.

[KK02] KOLLIG T., KELLER A.: Efficient multidimensional sampling. In *Proceedings of Eurographics* (Saarbrücken, 2002), vol. 21, pp. 557–563.

[KPR12] KELLER A., PREMOZE S., RAAB M.: Advanced (quasi) Monte Carlo methods for image synthesis. In *ACM SIGGRAPH 2012 Courses* (2012), pp. 21:1–21:46.

[KZ77] KORKIN A., ZOLOTAREV G.: Sur les formes quadratiques positives. *Mathematische Annalen 11* (1877), 242–292.

[LD06] LAGAE A., DUTRÉ P.: Poisson sphere distributions. In *Proceedings of Vision, Modeling, and Visualization 2006* (Aachen, November 2006), pp. 373–379.

[LD08] LAGAE A., DUTRÉ P.: A comparison of methods for generating Poisson disk distributions. *Computer Graphics Forum 27*, 1 (2008), 114–129.

[Llo82] LLOYD S.: Least squares quantization in PCM. *IEEE Transactions on Information Theory 28*, 2 (1982), 129–137.

[LWSF10] LI H., WEI L.-Y., SANDER P. V., FU C.-W.: Anisotropic blue noise sampling. In *Proceedings of SIGGRAPH* (Los Angeles, 2010), vol. 29, ACM, p. 167.

[MBR*13] MARQUES R., BOUVILLE C., RIBARDIERE M., SANTOS L. P., BOUATOUCH K.: Spherical Fibonacci point sets for illumination integrals. *Computer Graphics Forum 32*, 8 (2013), 134–143.

[MF92] MCCOOL M., FIUME E.: Hierarchical Poisson disk sampling distributions. In *Proceedings of GI* (Vancouver, 1992).

[Mit87] MITCHELL D. P.: Generating antialiased images at low sampling densities. *SIGGRAPH Computer Graphics 21*, 4 (1987), 65–72.

[Mit91] MITCHELL D. P.: Spectrally optimal sampling for distribution ray tracing. *ACM SIGGRAPH Computer Graphics 25*, 4 (1991), 157–164.

[Mit92] MITCHELL D.: Ray tracing and irregularities of distribution. In *Proceedings of EGWR* (Pisa, 1992), pp. 61–69.

[MP09] MITRA N. J., PAULY M.: Shadow art. In *Proceedings of SIGGRAPH* (New Orleans, 2009), vol. 28, pp. 156:1–156:7.

[ODJ04] OSTROMOUKHOV V., DONOHUE C., JODOIN P.-M.: Fast hierarchical importance sampling with blue noise properties. In *Proceedings of SIGGRAPH* (Los Angeles, 2004), vol. 23, pp. 488–495.

[PH10] PHARR M., HUMPHREYS G.: *Physically Based Rendering: From Theory to Implementation* (San Francisco, CA, USA, 2010), Morgan Kaufmann.

[RAMN12] RAMAMOORTHI R., ANDERSON J., MEYER M., NOWROUZEZAHRAI D.: A theory of Monte Carlo visibility sampling. *ACM Transactions on Graphics 31*, 5 (2012), 121:1–121:16.

[RBGP06] ROMERO V. J., BURKARDT J. V., GUNZBURGER M. D., PETERSON J. S.: Initial evaluation of pure and Latinized centroidal Voronoi tessellation for non-uniform statistical sampling. *Rel. Eng. & Sys. Safety 91*, 10 (2006), 1266–1280.

[RRS13] REINERT B., RITSCHEL T., SEIDEL H.-P.: Interactive by-example design of artistic packing layouts. In *Proceedings of SIGGRAPH Asia* (Hong Kong, 2013), vol. 32, p. 218.

[SGB07] SAKA Y., GUNZBURGER M., BURKHARDT J.: Latinized, improved LHS, and CVT point sets in hypercubes. *International Journal of Numerical Analysis and Modeling 4*, 3–4 (2007), 729–743.

[SHD11] SCHLÖMER T., HECK D., DEUSSEN O.: Farthest-point optimized point sets with maximized minimum distance. In *Proceedings of HPG* (Vancouver, 2011), pp. 135–142.

[Shi91] SHIRLEY P.: Discrepancy as a quality measure for sample distributions. In *Proceedings of Eurographics* (Vienna, 1991), vol. 91, pp. 183–194.

[Sob94] SOBOL I. M.: *A Primer for the Monte Carlo Method* (1st edition). Boca Raton, CRC Press, Florida, May 1994.

[Uli87] ULICHNEY R.: *Digital Halftoning*. MIT Press, Cambridge, MA, 1987.

[Yel83] YELLOTT J.: Spectral consequences of photoreceptor sampling in the Rhesus retina. *Science 221*, 4608 (1983), 382–385.

Supporting Information

Additional Supporting Information may be found in the online version of this article at the publisher's web site:

Images S1
Video S1